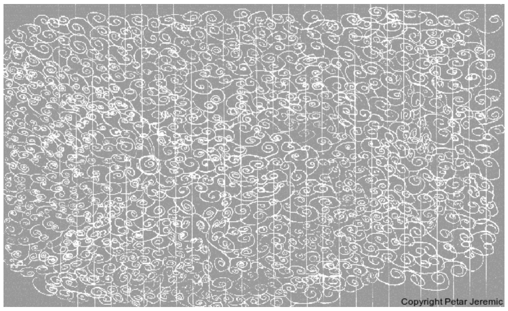
Nonlinear Finite Elements: Modeling and Simulation of Earthquakes, Soils, Structures and their Interaction

---

Prof. Dr. Boris Jeremić; Борис Јеремић
University of California, Davis, USA

with significant contributions, as noted in Chapters, by:

Prof. Dr. Zhaohui Yang 杨朝晖 University of Alaska, USA	Dr. Zhao Cheng 程 昭 Itasca International Inc., USA	Dr. Guanzhou Jie 揭冠周 Wells Fargo Securities, USA
Dr. Nima Tafazzoli نیما تفضلی Tetra Tech EBA, CAN	Dr. Matthias Preisig GeoMod, CHE	Dr. Panagiota Tasiopoulou Παναγιώτα Τασιοπούλου Fugro, GRC
Dr. Federico Pisanò NGI, USA	Prof. Dr. José Abell University of the Andes, CHL	Mr. Kohei Watanabe 渡邊 航平 Shimizu Corporation, JPN
Dr. Yuan Feng 冯源 TuSimple, USA	Prof. Dr. Sumeet Kumar Sinha सुमित कुमार सिन्हा University of New Delhi, IND	Ms. Fatemah Behbehani فاطمة بهبهاني UC, San Diego, USA
Prof. Dr. Han Yang 杨涵 University of Tianjin, CHN	Dr. Hexiang Wang 王和祥 Berkshire Hathaway Inc., USA	Dr. Katarzyna Staszewska Gdańsk UT, POL



Version: 3Jul2025, 10:19

ISBN: 978-0-692-19875-9
Typeset in LaTeX
http://sokocalo.engr.ucdavis.edu/~jeremic/LectureNotes/

Motivation and Inspiration

The main motivation for development of the FEI system, comprised of these lecture notes and accompanying modeling tools, computational libraries and visualization tools, is to help research and teach modeling and simulation for civil engineering mechanics problems. Focus is on development and use of methods that reduce Kolmogorov complexity and modeling uncertainty. In other words, focus is on development and use of methods that predict and inform rather than fit. These lecture notes, in particular, are being developed to document research, teaching and practical problem solving work for Real-ESSI problems (Realistic modeling and simulation of Earthquakes, Soils, Structures and their Interaction). Almost all of the theories, formulations and algorithms described here can be directly analyzed using Real-ESSI Simulator system (`http://real-essi.info/` ; `http://real-essi.us/`) on local computers or on Cloud Computers (Amazon Web Services, AWS, on Marketplace, search for ESSI). A number of theories and formulations, related to Real-ESSI problems, developed by us and others, as referenced, are collected within these Lecture Notes in order to have one location, one write-up, with all/most necessary material for the analysis of ESSI problems. These Lecture Notes are in perpetual development, and chapters and sections are being edited and added as you read this. In that sense, these Lecture Notes are not "polished" and there are some rough edges, however improvement work is underway.

Work on these lecture notes was inspired by a number of books and lecture notes that I have enjoyed over many years, (Bathe, 1982), (Bathe and Wilson, 1976), (Felippa, 1992b, 1989, 1993; Felippa and Park, 1995), (Willam, 1993), (Sture, 1993), (Lubliner, 1990), (Crisfield, 1991), (Chen and Han, 1988a), (Zienkiewicz and Taylor, 1991a,b), (Argyris and Mlejnek, 1991), (Malvern, 1969), (Saouma, 1992-2013), (Dunica and Kolundžija, 1986), (Kojić, 1997), (Hjelmstad, 1997), (Oberkampf et al., 2002). I particularly enjoyed book by Bathe (1982), the only one I had partial access to in the late '80s, with all the examples that could be worked out on paper. In '89 I managed to purchase a used book by Zienkiewicz (1977) for US$50, from my first salary as a young engineer in Energoprojekt Company, from a colleague. In the early '90s, I was lucky to get exposed to early, draft versions of books by Kojić (1993); Kojić (1997). Few years later, I enjoyed lectures and lecture notes by Felippa (1992b, 1989, 1993); Felippa and Park (1995), Willam (1993), and Sture (1993).

Current version of lecture notes, the one in front of you, aims to extend concepts described by my Teachers and Professors. Presented formulations and implementations are available within the Real-ESSI Simulator `http://real-essi.info/`, `http://real-essi.us/`. A number of available, provided models, some very simple, some more sophisticated, and some very sophisticated, can be analyzed and results visualized using Real-ESSI Simulator on local computers or on Cloud Computers (Amazon Web Services, AWS).

Dedication

This book is dedicated to friends, colleagues, supporters, promoters that I had a privilege to know and work with over last many years, and that are not with us anymore.
May they rest in peace!

Robert P. Kennedy, 1939-2018

"Response of a soil structure system is nonlinear, and I would really like to know what that response is!"
"There are engineers and then there are Engineers!"

Nebojša Orbović, Небојша Орбовић, 1962-2021

"As an engineer, I have to know, with good accuracy, what will happen to the structure during loading, and will use modeling and analysis to find out, hence verification and validation for modeling and analysis is really important"
"As an engineer, I have to know what are response sensitivities to modeling parameters."

Contributions

Useful contributions were also made by the following students, colleagues and collaborators (other than those listed on the front page): Mr. Babak Kamranimoghadam (بابک کامرانی), Mr. Chang-Gyun Jeong (정창균), Mr. Chao Luo (罗超), Mr. Max Sieber, Mr. Antonio Felipe Salazar, and Mr. Borko Miladinović (Борко Миладиновић).

Comments

Comments, corrections, edits &c. are much appreciated! Special thanks to (in chronological order): Miroslav Živković (Мирослав Живковић), Dmitry J. Nicolsky, Andrzej Niemunis, Robbie Jaeger, Yiorgos Perikleous (Γιώργος Περικλέους), Robert Roche, Viktor Vlaski, Edison Lam, Dr. Sukumar Baishya, Marco Andreini, Francisco "Paco" Beltran.

The best way to send a comment on these lecture notes is by email, however please read the following NOTE about sending an email to me. It would be great if you can place the following in the subject line of your email: `Draft CompMech Lecture Notes`. This will be much appreciated as it will help me filter your email and place it in Draft_CompMech_Lecture_Notes email-box that I regularly read.

Acknowledgement

Many developments described in these lecture notes, developed over many years, were made possible in collaborators with, and with financial support from the: US-DOE, US-NRC, CNSC/CCSN, US-NSF, CalTrans, CH-ENSI/IFSN, ATC/US-FEMA, UN-IAEA, US-ACE, and NASA. Their support and collaboration is much appreciated!

Software

Theoretical and computational developments described in these lecture notes are implemented in a program Real-ESSI Simulator (`http://real-essi.info/`; (`http://real-essi.us/`; `http://sokocalo.engr.ucdavis.edu/~jeremic/Real_ESSI_Simulator/`). The Real ESSI Simulator (Realistic Modeling and Simulation of Earthquakes, Soils, Structures and their Interaction), (pronunciation of ESSI is similar to easy, as in "as easy as pie") is a software, hardware and documentation system for select level of fidelity (high, low, intermediate), high performance, time domain, nonlinear/inelastic, deterministic or probabilistic, 3D, finite element modeling and simulation of

- statics and dynamics of soil,
- statics and dynamics of rock,
- statics and dynamics of structures,
- statics of soil-structure systems, and
- dynamics of earthquake-soil-structure system interaction.

The Real-ESSI Simulator systems is used for the analysis, design and assessment of static and dynamic behavior of infrastructure objects, including buildings, bridges, dams, nuclear energy installations, tunnels, etc. The Real-ESSI Simulator develops modeling and simulations that inform and predict.

The Real-ESSI Simulator program is available in source, in executable form and as debian package at `http://real-essi.info/`, `http://real-essi.us/`. The Real ESSI Simulator system is also available for use on Amazon Web Services Market Place, search for "Real ESSI" or "MS ESSI".

The name Real-ESSI, is explained in some detail in section 201.2.6 on page 710.

Distribution?

These Lecture Notes can be downloaded and distributed worldwide. If you use these Lecture Notes, please cite them as:

Boris Jeremić, Zhaohui Yang, Zhao Cheng, Guanzhou Jie, Nima Tafazzoli, Matthias Preisig, Panagiota Tasiopoulou, Federico Pisanò, José Abell, Kohei Watanabe, Yuan Feng, Sumeet Kumar Sinha, Fatemah Behbehani, Han Yang, and Hexiang Wang. Nonlinear Finite Elements: Modeling and Simulation of Earthquakes, Soils, Structures and their Interaction. University of California, Davis, CA, USA; 2021. ISBN: 978-0-692-19875-9

or

Jeremić et al. Nonlinear Finite Elements: Modeling and Simulation of Earthquakes, Soils, Structures and their Interaction. UCD, CA, USA, 2021. ISBN: 978-0-692-19875-9

Overview Table of Contents

200 Software and Hardware Platform: Design, Development, Procurement and Use

300 Verification and Validation

600 References

700 Appendix

Contents

702 The nDarray Programming Tool

703 Closed Form Gradients to the Plastic Potential Function

704 Hyperelasticity, Detailed Derivations

Part 100

Theoretical and Computational Formulations

Chapter 101

Introduction

(1996-2003-2016-2017-2018-2019-2020-2021-)

101.1 Chapter Summary and Highlights

101.2 On Modeling

Modeling of mechanical behaviour of civil engineering problems is performed using models. It is important to note that everything we do for design or assessment is based on models. Models can range from very simple to very sophisticated.

Simple Model Example. For example, simple model for the strength of a beam, as developed by Leonardo da Vinci and noted by Timoshenko (1953), states: "In every article that is supported, but is free to bend, and is of uniform cross section and material, the part that is farthest from the support will bend the most." Leonardo da Vinci suggested a number of tests with variation in beam length and keeping the same cross section, and record what loads could these beams carry. He concluded that strength of beams supported on both ends, simple beams, varies inversely proportional to teh beam length and directly to the beam width.

This simple model for beam bending, as stated, is based directly on observations of experiments. These experiments represent the physics discovery experiments.

Sophisticated Model Example. The same simple beam problem can be analyzed using a sophisticated Bernoulli or Timoshenko beam finite elements, or even using 3D solid brick elements.

The importance of computer analysis for design and performance assessment of civil engineering systems has recently dramatically increased. With availability of fast, inexpensive computers, and availability of numerical analysis programs, computer analysis of civil engineering systems for design and for assessment of performance is commonly used. It is important to note that failure of civil engineering systems has high consequences. For safe and economical design and for assessment of performance (reliability, robustness and safety) of civil engineering systems, engineers have to perform proper analysis, usually a numerical, computer analysis.

Developers of computer analysis software, computational analysts/engineers and users of numerical analysis face a critical question: How should confidence in analysis results be critically assessed?

Verification and Validation (V&V) of computational analysis is the major process for assessing and quantifying this confidence. Verification is the assessment of the software correctness and numerical accuracy of the solution to a given computational model. Validation is the assessment of physical accuracy of a computational model based on comparisons between computation simulation and experimental data. For verification, the association or relationship of the simulation to the real world is not an issue while for validation the relationship between computation and real world external data is the issue. Verification and validation are covered in much more detail in part 300 of this book, on pages after page 1436.

101.2.1 The Performance Challenge

Computational science challenges (Post, 2004):

- The Performance Challenge
- The Programming Challenge
- The Prediction Challenge

Parallel computing, development of portable Distributed Memory Parallel (DMP) applications, spedifically Architecture Aware Plastic Domain Decomposition (AAPDD) Method (Feng et al., 2024) that addresses the challenge of multiple CPUs, multiple cores, multiple GPUs, multiple networks, etc.

The Programming Challenge.

Quality controlled, managed program development...

The Prediction Challenge

Epistemic, modeling uncertainties

Aliatory, parametric uncertainties

Development of numerical modeling tools for engineers to develop a hierarchy of models, to explore, try, different levels of modeling sophistication in order to understand, gain insight into the influence of epistemic/modeling uncertainty on analysis and to propagate aleatory/parametric uncertainty through models...

101.2.2 Analysis Governance

Analysis governance, sometimes also called simulations governance[1] (Szabó and Actis, 2011, 2012), is a very important components of numerical analysis in civil engineering.

Analysis governance is a process of increasing confidence in numerical analysis results. Analysis governance covers both

- Numerical analysis tools, the program that is used for analysis

---

[1] Analysis consists of modeling, the physics component, and simulation, the numerics component, hence analysis covers both modeling and simulation components (Analysis = Modeling + Simulation).

- Numerical analyst, expert engineer that is performing the analysis

The numerical analysis program has to be extensively verified and validated.

Numerical analyst, expert engineer has to have required level of expertise, knowledge, experience and, if possible, has to be certified to perform required analysis.

101.3 Specialization to Computational Mechanics

In this section we start from general mechanics and specialize our interest toward the field of computational mechanics (this is based on great lecture notes by Prof. Carlos Felippa (Felippa, 1993)):

101.3.1 Mechanics

- Theoretical
- Applied
- Computational
    - Nanomechanics
    - Micromechanics
    - Continuum Mechanics
        - Solids and Structures
        - Fluids
        - Multiphysics
    - Systems
- Experimental

101.3.2 Continuum Mechanics

- Statics
    - Time invariant
    - Transient (quasi-statics)
- Dynamics

101.3.3 Statics and Dynamics

- Linear
- Nonlinear
    - Elastic
    - Inelastic

101.3.4 Discretization Methods

- Finite Element Method (FEM)
  - FEM Formulation
    - Displacement
    - Equilibrium
    - Mixed
    - Hybrid
  - FEM Solution
    - Stiffness
    - Flexibility
    - Mixed
- Boundary Element Method
- Finite Difference Method
- Finite Volume Method
- Spectral Method
- Mesh-Free Method

101.3.5 The Solution Morass

A system of 1000 linear equations has one solution.

A system of 1000 cubic equations has $3^{1000} \approx 10^{477}$ solutions.

It is worth putting this number in prospective: number of atoms in the earth is about 10^{50}, and a number of atoms in the universe is about 10^{78} (Niemunis, 2015 –).

Solution: Continuation or Incremental analysis!

101.3.6 Smooth Nonlinearities

- Finite deflections
- Nonlinear elasticity
- Follower forces

101.3.7 Rough Nonlinearities

- Elasto-plasticity
- Contact/Interface/Joint
- Interface/joint Friction

101.4 Tour of Computational Mechanics

In this section we describe various examples of equilibrium path and set up basic terminology.

101.4.1 Equilibrium Path

101.4.2 Special Equilibrium Points

101.4.2.1 Critical Points

Limit Points

Bifurcation Points

101.4.2.2 Turning Points

101.4.2.3 Failure Points

101.4.3 Generalized Response

101.4.4 Sources of Nonlinearities

Tonti Diagrams

101.4.5 Simulation Process: Loading Stages, Increments and Iterations

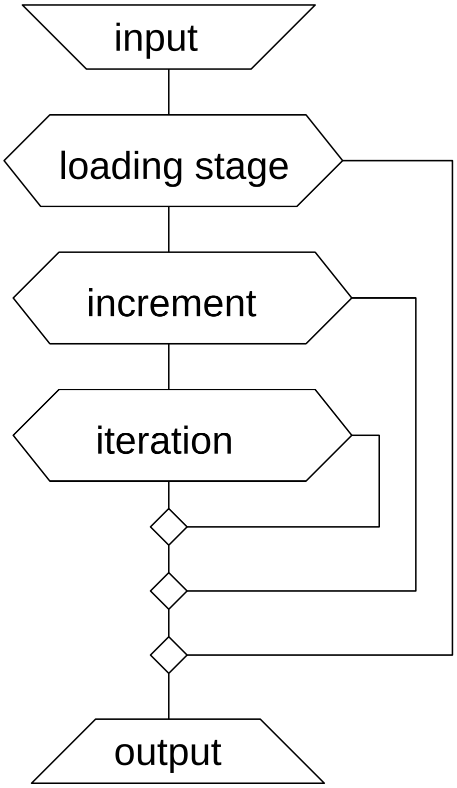


Figure 101.1: Nonlinear analysis loading stages, loading increments, equilibrium iterations.

Chapter 102

Finite Elements Formulation

(1989-1994-1999-2005-2010-2011-2012-2013-2015-2016-2017-2018-2019-2020-2021-)

(In collaboration with Dr. Zhao Cheng, Dr. Nima Tafazzoli, Prof. José Abell, Dr. Yuan Feng. Prof. Han Yang)

102.1 Chapter Summary and Highlights

This section uses basic principles of mechanics to derive finite element equations. We start with general setup, assuming large deformation in section 102.2, and then proceed to develop finite element formulation for small deformations in section 102.3 on page 104 . Further investigation of large deformation formulation is given in chapter 106 on page 482.

102.2 Formulation of the Continuum Mechanics Incremental Equations of Motion

This section follows Bathe (1982), Felippa (1989) and Felippa (1993).

Assume that a 3D solid is analyzed in a fixed Cartesian coordinate system, Figure (102.1). Also, assume that the solid can undergo large displacements and rotations, large strains, and nonlinear or inelastic constitutive response. The main aim is to evaluate the equilibrium of solid at discrete times $0, \Delta t, 2\Delta t, \ldots$, where Δt is an increment in time. To do that, a continuation strategy is used. That is, assume that the solution for all the variables (generalized displacements, strain, stress, forces, etc.) is available, was solved for, for all time steps from 0 to time t. Solution for the next time step $t + \Delta t$ will be based on already obtained solution from the previous time step, at time t, (Felippa, 1993). This approach will be applied for each time step, repetitively until the solution for all time steps is obtained.

In following all parts of the solid, as they undergo displacements and rotations, from the original configuration to the final configuration, adopted is a Lagrangian (or material) formulation of the problem. This approach is contrasting Eulerian (or spatial) formulation , usually used in the analysis of fluid mechanics problems.

In the Lagrangian incremental analysis approach the equilibrium of the solid at time $t+\Delta t$ is expressed using the principle of virtual displacements. Using tensorial notation[1] this principle requires that:

$$\int_{{}^{t+\Delta t}V} {}^{t+\Delta t}\sigma_{ij} \; \delta \; {}_{t+\Delta t}\epsilon_{ij} \; {}^{t+\Delta t}dV = {}^{t+\Delta t}\mathcal{R} \tag{102.1}$$

where the ${}^{t+\Delta t}\sigma_{ij}$ are Cartesian components of the Cauchy stress tensor, see section 701.2.1 on page 2860, the ${}_{t+\Delta t}\epsilon_{ij}$ are the Cartesian components of an infinitesimal strain tensor, see section 701.2.2 on page 2863, and the δ means "variation in" i.e.:

---

[1] Einstein's summation rule is implied unless stated differently, all lower case indices $(i, j, p, q, m, n, o, r, s, t, \ldots\ )$ can have values of 1, 2, 3, and values for capital letter indices will be specified where need be.

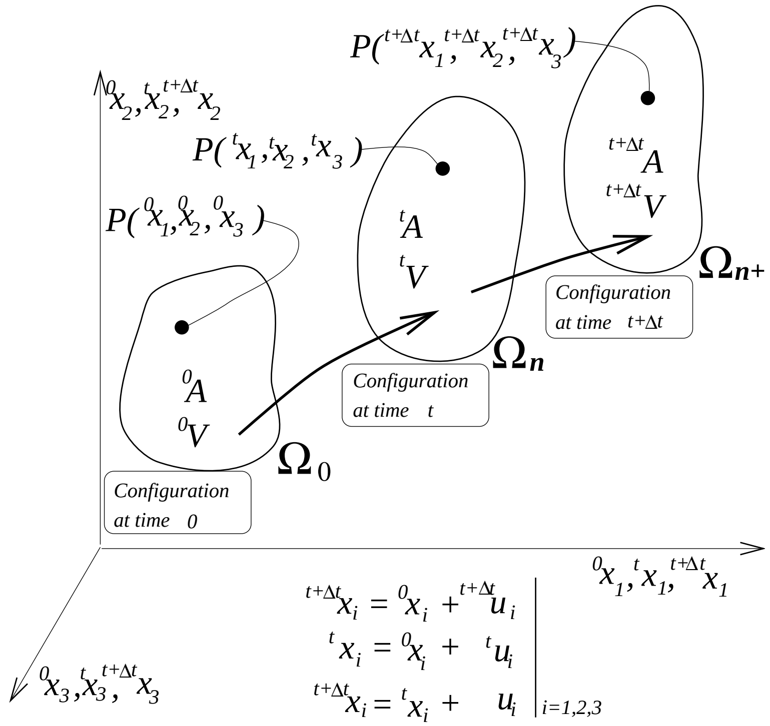


Figure 102.1: Motion of solid in a Cartesian coordinate system.

$$\delta\ {}_{t+\Delta t}\epsilon_{ij} = \delta\frac{1}{2}\left(\frac{\partial u_i}{\partial\ {}^{t+\Delta t}x_j}+\frac{\partial u_j}{\partial\ {}^{t+\Delta t}x_i}\right) = \frac{1}{2}\left(\frac{\partial\delta u_i}{\partial\ {}^{t+\Delta t}x_j}+\frac{\partial\delta u_j}{\partial\ {}^{t+\Delta t}x_i}\right) \qquad (102.2)$$

It should be noted that Cauchy stresses are "body forces per unit area" in the configuration at time $t+\Delta t$, and the infinitesimal strain components are also referred to this as yet unknown configuration. The right hand side of equation (102.1), i.e. ${}^{t+\Delta t}\mathcal{R}$ is the virtual work performed when the solid is subjected to a virtual displacement at time $t+\Delta t$:

$${}^{t+\Delta t}\mathcal{R} = \int_{{}^{t+\Delta t}V}\left({}^{t+\Delta t}f_i^B - \rho\ddot{u}_i^{t+\Delta t}\right)\ \delta u_i^{t+\Delta t}\ {}^{t+\Delta t}dV + \int_{{}^{t+\Delta t}S} {}^{t+\Delta t}f_i^S\ \delta u_i^{t+\Delta t}\ {}^{t+\Delta t}dS$$

where ${}^{t+\Delta t}f_i^B$ and ${}^{t+\Delta t}f_i^S$ are the components of the externally applied body and surface force vectors, respectively, and $-\rho\ddot{u}_i{}^{t+\Delta t}$ is the inertial body force that is present if accelerations are present[2], δu_i is the ith component of the virtual displacement vector.

The main problem in applying equation (102.1) is that the configuration of the solid at a time $t+\Delta t$ is unknown. The continuous change in the configuration, deformation of the solid needs to be properly

---

[2] This is based on D'Alembert's principle (D'Alembert, 1758).

modeled. As an example, consider, for example, Cauchy stress at time $t + \Delta t$. This stress cannot be obtained by adding to the Cauchy stresses at time t, a stress increment that is due only to material deformation. The reason is that material might rotate, and stress state is a function of tractions (loads) and size and orientation of differentially small faces on which stress components act. For material only nonlinear analysis, the of large displacements, large rotations and large strain will be neglected. Large displacements, rotations and large strains will be addressed in more detail in Chapter 106 on page 482.

The continuous change in the configuration of the solid is dealt with by using appropriate stress and strain measures and constitutive relations. When solving the general problem[3] one possible approach is given in Simo (1988). The previous discussion was oriented toward small deformation, small-displacement analysis leading to the use of Cauchy stress tensor σ_{ij} and small strain tensor ϵ_{ij}.

In the following, covered briefly are other stress and strain measures particularly useful in large strain and large displacement analysis. More detailed description of large displacements, large rotations and large strains problems is addressed in Chapter 106 on page 482.

The solution is sought for equation 102.1, which expresses the equilibrium and compatibility requirements of the general solid considered in the configuration corresponding to time $t + \Delta t$. The nonlinear or inelastic behavior of material enters equation 102.1 through the stress-strain constitutive equations. In general, the solid can undergo large displacements, large rotations large strains, and since constitutive relations are nonlinear, the relation in equation 102.1 cannot be solved directly. However, an approximate solution can be obtained by referring all variables to a previously calculated known equilibrium configuration, and linearizing the resulting equations. Iterations can then improve this solution.

To develop the governing equations for the approximate solution obtained by linearization, recall that the solutions for time $0, \Delta t, 2\Delta t, \ldots, t$ have already been calculated and that the Piola–Kirchhoff stress tensor is energy conjugate to the Green–Lagrange strain tensor:

$$\int_{{}^0V} {}^t_0S_{ij}\; \delta^t_0\epsilon_{ij}\; {}^0dV = \int_{{}^0V} \left(\frac{{}^0\rho}{{}^t\rho}\; {}^0_t x_{i,m}\; {}^t\sigma_{mn}\; {}^0_t x_{j,n} \right) \left({}^t_0 x_{k,i}\; \delta^t_t\epsilon_{kl}\; {}^t_0\, x_{l,j} \right)\; {}^0dV = \int_{{}^0V} \frac{{}^0\rho}{{}^t\rho}\; {}^t\sigma_{mn}\; \delta^t_0\epsilon_{mn}\; {}^0dV \tag{102.3}$$

since

$${}^t_0x_{k,l}\,{}^0_t x_{l,m} = \delta_{km}$$

and

---

[3] That is, large displacements, large rotations, large deformations and material nonlinear and/or inelastic.

$$ {}_{0}\rho \; {}^{0}dV = {}^{t}\rho \, {}^{t}dV $$

one obtains

$$ \int_{^{0}V} {}^{t}_{0}S_{ij} \; \delta^{t}_{0}\epsilon_{ij} \; {}^{0}dV = \int_{^{0}V} {}^{t}\sigma_{mn} \; \delta^{t}_{t}\epsilon_{mn} \; {}^{t}dV \tag{102.4} $$

where 2nd Piola–Kirchhoff stress tensor is defined as:

$$ {}^{t}_{0}S_{ij} = \frac{{}^{0}\rho}{{}^{t}\rho} \; {}^{0}_{t}x_{i,m} \; {}^{t}\sigma_{mn} \; {}^{0}_{t}x_{j,n} \tag{102.5} $$

and ${}^{0}_{t}x_{j,n} = \frac{\partial^{0}x_i}{\partial^{t}x_m}$, and $\frac{{}^{0}\rho}{{}^{t}\rho}$ represents the ratio of the mass density at time 0 and time t, and the Green–Lagrange strain is defined as:

$$ {}^{t}_{0}\epsilon_{ij} = \frac{1}{2}\left({}^{t}_{0}u_{i,j} + {}^{t}_{0}u_{j,i} + {}^{t}_{0}u_{k,i} \; {}^{t}_{0}u_{k,j} \right) \tag{102.6} $$

By employing equation 102.4 stresses and strains are referenced to the known equilibrium configuration. The choice lies between two formulations, named (a)total Lagrangian and (b) updated Lagrangian formulations.

For the total Lagrangian formulations, all static and kinematic variables are referenced to the initial configuration at time 0. On the other hand, for the updated Lagrangian formulation, all static and kinematic variables are referenced to the previous step equilibrium configuration at time t. Both the total Lagrangian and updated Lagrangian formulations include all kinematic nonlinear effects due to large displacement, large rotations, and large strains. Whether the large strain behavior is modeled appropriately depends on the constitutive relations specified. The only advantage of using one over the other formulation lies in numerical efficiency.

Using equation 102.4 in the total Lagrangian formulation, considered is this equation:

$$ \int_{^{0}V} {}^{t+\Delta t}_{0}S_{ij} \; \delta^{t+\Delta t}_{0}\epsilon_{ij} \; {}^{0}dV = {}^{t+\Delta t}\mathcal{R} \tag{102.7} $$

while in the updated Lagrangian formulation considered is this equation:

$$ \int_{^{t}V} {}^{t+\Delta t}_{t}S_{ij} \; \delta^{t+\Delta t}_{t}\epsilon_{ij} \; {}^{t}dV = {}^{t+\Delta t}\mathcal{R} \tag{102.8} $$

where ${}^{t+\Delta t}\mathcal{R}$ is the external virtual work as defined in equation ??. Approximate solution to the equation 102.7 and equation 102.8 can be obtained by linearization. Comparison of the total Lagrangian and updated Lagrangian formulations reveal that they are quite similar, with the difference in the choice of different reference configurations for kinematic and static variables. If in the numerical solution the appropriate constitutive tensors are employed, identical results should be obtained.

102.3 Finite Element Discretization

Consider the equilibrium of a general three–dimensional solid such as in Figure (102.2) (Bathe, 1996). The external forces acting on a solid are surface tractions f_i^S and body forces f_i^B. Displacements are u_i and strain tensor[4] is ϵ_{ij} and the stress tensor corresponding to strain tensor is σ_{ij}.

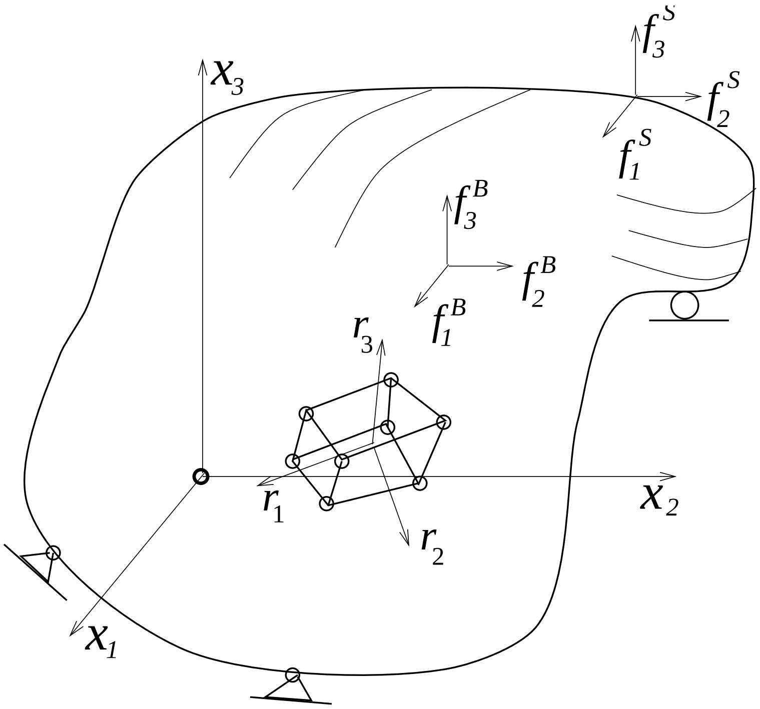


Figure 102.2: General three dimensional solid.

Dynamic equilibrium equation is given as

$$\sigma_{ij,j} = f_i - \rho \ddot{u}_i \tag{102.9}$$

where $\sigma_{ij,j}$ is a small deformation (Cauchy) stress tensor, f_i are external (body (f_i^B) and surface (f_i^S)) forces, ρ is material density and $\ddot{u}_i$ are accelerations. Inertial forces $\rho \ddot{u}_i$ follow from D'Alembert's principle

---

[4] small strain tensor as defined in equation: $\epsilon_{ij} = \frac{1}{2}\left(u_{i,j} + u_{j,i}\right)$.

(D'Alembert, 1758). The above equation can be premultiplied with virtual displacements δu_i and then integrated by parts to obtain the weak form, as further elaborated below.

For a given solid, loaded with external forces, with prescribed supports solution for displacements, strains and stresses are sought. The principle of virtual displacements (PVD) can be used to find a solution. Using PVD, equilibrium is achieved if the total internal virtual work is equal to the total external virtual work, for any compatible, small virtual displacements, that satisfy the essential boundary conditions.

Mathematically this is expressed using equation 102.10 for the solid at time $t + \Delta t$. Since the incremental approach is used, a time dimension is dropped so that all the equations are applied for the given increment[5], at time $t + \Delta t$. The equation is now, using tensorial notation[6]:

$$\int_V \sigma_{ij} \; \delta\epsilon_{ij} \; dV = \int_V \left(f_i^B - \rho \ddot{u}_i \right) \; \delta u_i \; dV + \int_S f_i^S \; \delta u_i \; dS \tag{102.10}$$

The internal work given on the left side of (102.10) is equal to the actual stresses σ_{ij} going through the virtual strains $\delta\epsilon_{ij}$ that corresponds to the imposed virtual displacements. The external work is on the right side of (102.10) and is equal to the actual surface forces f_i^S and body forces $f_i^B - \rho\ddot{u}_i$ going through the virtual displacements δu_i.

It is noted virtual strains used in equationi 102.10 correspond to the imposed virtual displacements that can be represented by any compatible set of displacements that satisfy the geometric boundary conditions. The equation 102.10 is an expression of equilibrium, and for different virtual displacements, correspondingly, different equations of equilibrium are obtained. Equation 102.10 also contains the compatibility and constitutive requirements. Displacements should be continuous and compatible and should satisfy the displacement boundary conditions, and the stresses should be evaluated from the strains using appropriate constitutive relations. Thus, the principle of virtual displacements contains all requirements that need to be fulfilled to analyze a problem in solid and structural mechanics. The principle of virtual displacements can be directly related to the principle that the total potential Π of the system must be stationary.

In the finite element analysis, approximation for the solid in Figure 102.2 is done by creating an assemblage of discrete finite elements with the elements connected at nodal points. The displacements measured in a local coordinate system r_1, r_2 and r_3 within each element are assumed to be a function of the displacements at the N finite element nodal points:

---

[5] $t + \Delta t$ will be dropped from now one in this chapter.

[6] Einstein's summation rule is implied unless stated differently, all lower case indices $(i, j, p, q, m, n, o, r, s, t, \dots)$ can have values of 1, 2, 3, and values for capital letter indices will be specified where need be.

$$u_i \approx \hat{u}_a = H_I \bar{u}_{Ia} \tag{102.11}$$

where $I = 1, 2, 3, \ldots, n$ and n is number of nodes in a specific element, $a = 1, 2, 3$ represents a number of dimensions (can be 1 or 2 or 3). Real displacement field u_i is approximated with approximate displacement field $\hat{u}_a$, and H_I represent displacement interpolation functions, $\bar{u}_{Ia}$ is the tensor of global approximate generalized displacement components at all element nodes. The term generalized displacements mean that both translations, rotations, or any other nodal unknown are modeled independently. Here specifically, only translational degrees of freedom are considered. The strain tensor is defined as:

$$\epsilon_{ab} = \frac{1}{2}\left(u_{a,b} + u_{b,a}\right) \tag{102.12}$$

and by using equation 102.11, approximate strain tensor can be defined as:

$$\begin{aligned} \epsilon_{ab} \;\approx\; \hat{e}_{ab} \;&=\; \frac{1}{2}\left(\hat{u}_{a,b} + \hat{u}_{b,a}\right) = \\ &=\; \frac{1}{2}\left((H_I\ \bar{u}_{Ia})_{,b} + (H_I\ \bar{u}_{Ib})_{,a}\right) = \\ &=\; \frac{1}{2}\left(\left(H_{I,b}\ \bar{u}_{Ia}\right) + \left(H_{I,a}\ \bar{u}_{Ib}\right)\right) \end{aligned} \tag{102.13}$$

The most general stress-strain relationship[7] for an isotropic material is:

$$\hat{\sigma}_{ab} = E_{abcd}\left(\hat{e}_{cd} - \epsilon^0_{cd}\right) + \sigma^0_{ab} \tag{102.14}$$

where $\hat{\sigma}_{ab}$ is the approximate Cauchy stress tensor, E_{abcd} is the constitutive tensor[8], $\hat{e}_{cd}$ is the infinitesimal approximate strain tensor, ϵ^0_{cd} is the infinitesimal initial strain tensor and σ^0_{ab} is the initial Cauchy stress tensor.

Using the assumption of the displacements within each finite element, as expressed in equation 102.11, equilibrium equations that correspond to nodal point displacements of the assemblage of finite elements can be derived. Equation 102.10 can be rewritten as a sum[9] of integrations over the volume and areas of all finite elements:

$$\bigcup_m \int_{V^m} \hat{\sigma}_{ab}\ \delta\hat{e}_{ab}\ dV^m = \bigcup_m \int_{V^m} \left(f^B_a - \rho\ddot{u}_a\right)\ \delta\hat{u}_a\ dV^m + \bigcup_m \int_{S^m} f^S_a\ \delta\hat{u}^S_a\ dS^m$$

---

[7] in terms of exact stress and strain fields, but it holds for approximate fields as well.

[8] This tensor can be elastic or elastoplastic constitutive tensor.

[9] Or, more correctly as a union $\bigcup_m$ since we are integrating over the union of elements.

where $m = 1, 2, 3, \ldots, k$ and k is the number of elements. It is important to note that the integrations in (??) are performed over the element volumes and surfaces, and that for convenience different element coordinate systems may be used in the calculations. If we substitute equations 102.11, 102.12, 102.13 and 102.14 in ??, it follows:

$$\bigcup_m \int_{V^m} \left(E_{abcd} \left(\hat{e}_{cd} - \epsilon^0_{cd} \right) + \sigma^0_{ab} \right) \; \delta \left(\frac{1}{2} \left(H_{I,b} \; \bar{u}_{Ia} + H_{I,a} \; \bar{u}_{Ib} \right) \right) \; dV^m = \\ \bigcup_m \int_{V^m} f^B_a \; \delta \left(H_I \bar{u}_{Ia} \right) \; dV^m - \bigcup_m \int_{V^m} H_J \; \ddot{u}_{Ja} \; \rho \; \delta \left(H_I \bar{u}_{Ia} \right) \; dV^m + \bigcup_m \int_{S^m} f^S_a \; \delta \left(H_I \bar{u}_{Ia} \right) \; dS^m \tag{102.15}$$

or:

$$\bigcup_m \int_{V^m} \left(E_{abcd} \left(\left(\frac{1}{2} \left(H_{J,d} \; \bar{u}_{Jc} + H_{J,c} \; \bar{u}_{Jd} \right) \right) - \epsilon^0_{cd} \right) + \sigma^0_{ab} \right) \; \delta \left(\frac{1}{2} \left(H_{I,b} \; \bar{u}_{Ia} + H_{I,a} \; \bar{u}_{Ib} \right) \right) \; dV^m = \\ = \bigcup_m \int_{V^m} f^B_a \; \delta \left(H_I \bar{u}_{Ia} \right) \; dV^m - \bigcup_m \int_{V^m} H_J \; \ddot{u}_{Ja} \; \rho \; \delta \left(H_I \bar{u}_{Ia} \right) \; dV^m + \bigcup_m \int_{S^m} f^S_a \; \delta \left(H_I \bar{u}_{Ia} \right) \; dS^m \tag{102.16}$$

We can observe that δ in the previous equations represents a virtual quantity, but the rules for δ are quite similar to regular differentiation so that δ can enter the brackets and "virtualize" the nodal displacement[10]. It thus follows:

$$\bigcup_m \int_{V^m} \left(E_{abcd} \left(\left(\frac{1}{2} \left(H_{J,d} \; \bar{u}_{Jc} + H_{J,c} \; \bar{u}_{Jd} \right) \right) - \epsilon^0_{cd} \right) + \sigma^0_{ab} \right) \; \left(\frac{1}{2} \left(H_{I,b} \; \delta\bar{u}_{Ia} + H_{I,a} \; \delta\bar{u}_{Ib} \right) \right) \; dV^m = \\ = \bigcup_m \int_{V^m} f^B_a \; \left(H_I \delta\bar{u}_{Ia} \right) \; dV^m - \bigcup_m \int_{V^m} H_J \; \ddot{u}_{Ja} \; \rho \; \left(H_I \delta\bar{u}_{Ia} \right) \; dV^m + \bigcup_m \int_{S^m} f^S_a \; \left(H_I \delta\bar{u}_{Ia} \right) \; dS^m \tag{102.17}$$

Let us now work out some algebra on the left-hand side of the equation (102.17):

$$\bigcup_m \int_{V^m} \left(E_{abcd} \left(\frac{\left(H_{J,d} \; \bar{u}_{Jc} + H_{J,c} \; \bar{u}_{Jd} \right)}{2} \right) - E_{abcd} \epsilon^0_{cd} + \sigma^0_{ab} \right) \; \left(\frac{\left(H_{I,b} \; \delta\bar{u}_{Ia} + H_{I,a} \; \delta\bar{u}_{Ib} \right)}{2} \right) \; dV^m = \\ = \bigcup_m \int_{V^m} f^B_a \; \left(H_I \delta\bar{u}_{Ia} \right) \; dV^m - \bigcup_m \int_{V^m} H_J \; \ddot{u}_{Ja} \; \rho \; H_I \delta\bar{u}_{Ia} \; dV^m + \bigcup_m \int_{S^m} f^S_a \; \left(H_I \delta\bar{u}_{Ia} \right) \; dS^m \tag{102.18}$$

and further:

---

[10] since they are driving variables that define the overall displacement field through interpolation functions

$$
\begin{aligned}
\bigcup_m \int_{V^m} \left(\left(\frac{1}{2} \left(H_{J,d}\; \bar{u}_{Jc} + H_{J,c}\; \bar{u}_{Jd} \right) \right)\; E_{abcd}\; \left(\frac{1}{2} \left(H_{I,b}\; \delta\bar{u}_{Ia} + H_{I,a}\; \delta\bar{u}_{Ib} \right) \right) \right)\; dV^m + \\
+ \bigcup_m \int_{V^m} \left(-E_{abcd}\; \epsilon^0_{cd}\; \left(\frac{1}{2} \left(H_{I,b}\; \delta\bar{u}_{Ia} + H_{I,a}\; \delta\bar{u}_{Ib} \right) \right) \right)\; dV^m + \\
+ \bigcup_m \int_{V^m} \left(\sigma^0_{ab} \right)\; \left(\frac{1}{2} \left(H_{I,b}\; \delta\bar{u}_{Ia} + H_{I,a}\; \delta\bar{u}_{Ib} \right) \right)\; dV^m = \\
\bigcup_m \int_{V^m} f^B_a\; \left(H_I \delta\bar{u}_{Ia} \right)\; dV^m \\
- \bigcup_m \int_{V^m} H_J\; \ddot{\bar{u}}_{Ja}\; \rho\; H_I \delta\bar{u}_{Ia}\; dV^m \\
+ \bigcup_m \int_{S^m} f^S_a\; \left(H_I \delta\bar{u}_{Ia} \right)\; dS^m
\end{aligned}
\qquad (102.19)
$$

Several things should be observed in the equation (102.19). Namely, the first three lines in the equation can be simplified if one takes into account symmetries of E_{ijkl} and σ_{ij}. In the case of the elastic stiffness tensor E_{ijkl} major and both minor symmetries exist. In the case of the elastoplastic stiffness tensor, such symmetries exist if a flow rule is associated. If the flow rule is non–associated, only minor symmetries exist while major symmetry is destroyed[11]. As a matter of fact, both minor symmetries in E_{ijkl} are the only symmetries needed, and the first line of (102.19) can be rewritten as:

$$
\begin{aligned}
& \bigcup_m \int_{V^m} \left(\left(\frac{1}{2} \left(H_{J,d}\; \bar{u}_{Jc} + H_{J,c}\; \bar{u}_{jd} \right) \right)\; E_{abcd}\; \left(\frac{1}{2} \left(H_{I,b}\; \delta\bar{u}_{Ia} + H_{I,a}\; \delta\bar{u}_{Ib} \right) \right) \right)\; dV^m = \\
= & \bigcup_m \int_{V^m} \left(H_{J,d}\; \bar{u}_{Jc} \right)\; E_{abcd}\; \left(H_{I,b}\; \delta\bar{u}_{Ia} \right)\; dV^m = \\
= & \bigcup_m \int_{V^m} \left(H_{I,b}\; \delta\bar{u}_{Ia} \right)\; E_{abcd}\; \left(H_{J,d}\; \bar{u}_{Jc} \right)\; dV^m
\end{aligned}
\qquad (102.20)
$$

Similar simplifications are possible in the second and third line of the equation (102.19). Namely, in the second line both minor symmetries of E_{ijkl} can be used so that:

---

[11] for more on stiffness tensor symmetries, see sections (104.6.1, 104.3 and 104.4)

$$\begin{aligned} & \bigcup_m \int_{V^m} \left(-E_{abcd}\;\epsilon^0_{cd}\;\left(\frac{1}{2}\left(H_{I,b}\;\delta\bar{u}_{Ia} + H_{I,a}\;\delta\bar{u}_{Ib}\right)\right)\right)\;dV^m = \\ = \;& \bigcup_m \int_{V^m} \left(-E_{abcd}\;\epsilon^0_{cd}\;\left(H_{I,b}\;\delta\bar{u}_{Ia}\right)\right)\;dV^m \end{aligned} \tag{102.21}$$

and the third line can be simplified due to the symmetry in Cauchy stress tensor σ_{ij} as:

$$\begin{aligned} & \bigcup_m \int_{V^m} \left(\sigma^0_{ab}\right)\;\left(\frac{1}{2}\left(H_{I,b}\;\delta\bar{u}_{Ia} + H_{I,a}\;\delta\bar{u}_{Ib}\right)\right)\;dV^m = \\ = \;& \bigcup_m \int_{V^m} \left(\sigma^0_{ab}\right)\;\left(H_{I,b}\;\delta\bar{u}_{Ia}\right)\;dV^m \end{aligned} \tag{102.22}$$

After these simplifications, equation (102.19) looks like this:

$$\begin{aligned} & \bigcup_m \int_{V^m} \left(H_{I,b}\;\delta\bar{u}_{Ia}\right)\;E_{abcd}\;\left(H_{J,d}\;\bar{u}_{Jc}\right)\;dV^m + \\ + \;& \bigcup_m \int_{V^m} \left(-E_{abcd}\;\epsilon^0_{cd}\;\left(H_{I,b}\;\delta\bar{u}_{Ia}\right)\right)\;dV^m + \bigcup_m \int_{V^m} \left(\sigma^0_{ab}\right)\;\left(H_{I,b}\;\delta\bar{u}_{Ia}\right)\;dV^m = \\ = \;& \bigcup_m \int_{V^m} f^B_a\;\left(H_I\delta\bar{u}_{Ia}\right)\;dV^m - \bigcup_m \int_{V^m} H_J\;\ddot{\bar{u}}_{Ja}\;\rho\;H_I\delta\bar{u}_{Ia}\;dV^m + \bigcup_m \int_{S^m} f^S_a\;\left(H_I\delta\bar{u}_{Ia}\right)\;dS^m \end{aligned} \tag{102.23}$$

or if unknown nodal accelerations[12] $\ddot{\bar{u}}_{Jc}$ and displacements $\bar{u}_{Jc}$ are left on the left hand side and all known quantities are moved to the right hand side:

$$\begin{aligned} & \bigcup_m \int_{V^m} H_J\;\delta_{ac}\;\ddot{\bar{u}}_{Jc}\;\rho\;H_I\delta\bar{u}_{Ia}\;dV^m + \bigcup_m \int_{V^m} \left(H_{I,b}\;\delta\bar{u}_{Ia}\right)\;E_{abcd}\;\left(H_{J,d}\;\bar{u}_{Jc}\right)\;dV^m = \\ = \;& \bigcup_m \int_{V^m} f^B_a\;\left(H_I\delta\bar{u}_{Ia}\right)\;dV^m + \bigcup_m \int_{S^m} f^S_a\;\left(H_I\delta\bar{u}_{Ia}\right)\;dS^m + \\ + \;& \bigcup_m \int_{V^m} \left(E_{abcd}\;\epsilon^0_{cd}\;\left(H_{I,b}\;\delta\bar{u}_{Ia}\right)\right)\;dV^m - \bigcup_m \int_{V^m} \left(\sigma^0_{ab}\right)\;\left(H_{I,b}\;\delta\bar{u}_{Ia}\right)\;dV^m \end{aligned} \tag{102.24}$$

To obtain the equation for the unknown nodal generalized displacements from equation 102.24, invoke the virtual displacement theorem. This theorem states that virtual displacements are any, non

---

[12] It is noted that $\ddot{\bar{u}}_{Jc} = \delta_{ac}\ddot{\bar{u}}_{Ja}$ relationship was used here, where δ_{ac} is the Kronecker delta.

zero, kinematically admissible displacements. In that case, we can factor out nodal virtual displacements $\delta\bar{u}_{Ia}$ so that equation 102.24 becomes:

$$\begin{aligned}
&\left[\bigcup_m \int_{V^m} H_J\ \delta_{ac}\ \ddot{\bar{u}}_{Jc}\ \rho\ H_I\ dV^m + \bigcup_m \int_{V^m} \left(H_{I,b}\right)\ E_{abcd}\ \left(H_{J,d}\ \bar{u}_{Jc}\right)\ dV^m\right]\ \delta\bar{u}_{Ia} = \\
= &\bigcup_m \left[\int_{V^m} f_a^B\ H_I\ dV^m\right] \delta\bar{u}_{Ia} + \bigcup_m \left[\int_{S^m} f_a^S\ H_I\ dS^m\right] \delta\bar{u}_{Ia} + \\
+ &\bigcup_m \left[\int_{V^m} \left(E_{abcd}\ \epsilon_{cd}^0\ H_{I,b}\right)\ dV^m\right] \delta\bar{u}_{Ia} - \bigcup_m \left[\int_{V^m} \left(\sigma_{ab}^0\right)\ H_{I,b}\ dV^m\right] \delta\bar{u}_{Ia}
\end{aligned} \tag{102.25}$$

and now just cancel $\delta\bar{u}_{Ia}$ on both sides:

$$\begin{aligned}
&\bigcup_m \int_{V^m} H_J\ \ \delta_{ac}\ \rho\ H_I\ \ddot{\bar{u}}_{Jc} dV^m + \\
&\bigcup_m \int_{V^m} \left(H_{I,b}\right)\ E_{abcd}\ \left(H_{J,d}\ \bar{u}_{Jc}\right)\ dV^m = \\
= &\bigcup_m \int_{V^m} f_a^B\ H_I\ dV^m + \bigcup_m \int_{S^m} f_a^S\ H_I\ dS^m + \\
+ &\bigcup_m \int_{V^m} \left(E_{abcd}\ \epsilon_{cd}^0\ H_{I,b}\right)\ dV^m - \bigcup_m \int_{V^m} \left(\sigma_{ab}^0\right)\ H_{I,b}\ dV^m
\end{aligned} \tag{102.26}$$

One should also observe that in the first line of equation (102.26) generalized nodal accelerations $\ddot{\bar{u}}_{Jc}$ and generalized nodal displacements $\bar{u}_{Jc}$ are unknowns that are not subjected to integration so they can be factored out of the integral:

$$\begin{aligned}
&\bigcup_m \int_{V^m} H_J\ \delta_{ac}\ \rho\ H_I\ dV^m\ \ddot{\bar{u}}_{Jc} \\
+ &\bigcup_m \int_{V^m} H_{I,b}\ E_{abcd}\ H_{J,d}\ dV^m\ \bar{u}_{Jc} \\
= &\bigcup_m \int_{V^m} f_a^B\ H_I\ dV^m + \bigcup_m \int_{S^m} f_a^S\ H_I\ dS^m + \\
+ &\bigcup_m \int_{V^m} \left(E_{abcd}\ \epsilon_{cd}^0\ H_{I,b}\right)\ dV^m - \bigcup_m \int_{V^m} \left(\sigma_{ab}^0\right)\ H_{I,b}\ dV^m
\end{aligned} \tag{102.27}$$

We can now define several tensors from equation (102.27):

$$^{(m)}M_{IacJ} = \int_{V^m} H_J \; \delta_{ac} \; \rho \; H_I \; dV^m \tag{102.28}$$

$$^{(m)}K_{IacJ} = \int_{V^m} H_{I,b} \; E_{abcd} \; H_{J,d} \; dV^m \tag{102.29}$$

$$^{(m)}F^B_{Ia} = \int_{V^m} f^B_a \; H_I \; dV^m \tag{102.30}$$

$$^{(m)}F^S_{Ia} = \int_{S^m} f^S_a \; H_I \; dS^m \tag{102.31}$$

$$^{(m)}F^{\epsilon^0_{mn}}_{Ia} = \int_{V^m} E_{abcd} \; \epsilon^0_{cd} \; H_{I,b} \; dV^m \tag{102.32}$$

$$^{(m)}F^{\sigma^0_{mn}}_{Ia} = \int_{V^m} \sigma^0_{ab} \; H_{I,b} \; dV^m \tag{102.33}$$

where $^{(m)}M_{IacJ}$ is the element mass tensor, $^{(m)}K_{IacJ}$ is the element stiffness tensor, $^{(m)}F^B_{Ia}$ is the tensor of element body forces, $^{(m)}F^S_{Ia}$ is the tensor of element surface forces, $^{(m)}F^{\epsilon^0_{mn}}_{Ia}$ is the tensor of element initial strain effects, $^{(m)}F^{\sigma^0_{mn}}_{Ia}$ is the tensor of element initial stress effects. Now equation (102.27) becomes:

$$\bigcup_{(m)} {}^{(m)}M_{IacJ} \; \ddot{\bar{u}}_{Jc} + \bigcup_{(m)} {}^{(m)}K_{IacJ} \; \bar{u}_{Jc} = \bigcup_{m} {}^{(m)}F^B_{Ia} + \bigcup_{m} {}^{(m)}F^S_{Ia} + \bigcup_{m} {}^{(m)}F^{\epsilon^0_{mn}}_{Ia} - \bigcup_{m} {}^{(m)}F^{\sigma^0_{mn}}_{Ia} \tag{102.34}$$

By summing[13] all the relevant tensors, a well known equation is obtained:

$$\begin{aligned} M_{AacB} \; \ddot{\bar{u}}_{Bc} + K_{AacB} \; \bar{u}_{Bc} &= F_{Aa} \\ A, B &= 1, 2, \ldots, \# \text{ of nodes} \\ a, c &= 1, \ldots, \# \text{ of dimensions (1, 2 or 3)} \end{aligned} \tag{102.35}$$

where:

---

[13] Summation of the element volume integrals expresses the direct addition of the element tensors to obtain global, system tensors. This method of direct addition is usually referred to as the direct stiffness method.

$$M_{AacB} = \bigcup_m {}^{(m)}M_{IacJ} \quad ; \quad K_{AacB} = \bigcup_m {}^{(m)}K_{IacJ} \tag{102.36}$$

are the system mass and stiffness tensors, respectively, $\ddot{u}_{Bc}$ is the tensor of unknown nodal accelerations, and $\bar{u}_{Bc}$ is the tensor of unknown generalized nodal displacements, while the load tensor is given as:

$$F_{Aa} = \bigcup_m {}^{(m)}F^{B}_{Ia} + \bigcup_m {}^{(m)}F^{S}_{Ia} + \bigcup_m {}^{(m)}F^{\epsilon^0_{mn}}_{Ia} - \bigcup_m {}^{(m)}F^{\sigma^0_{mn}}_{Ia} \tag{102.37}$$

After assembling the system of equations in (102.36), it is relatively easy to solve for the unknown displacements $\bar{u}_{Lc}$ either for static or fully dynamic case. It is also very important to note that in all previous equations, omissions of inertial force term (all terms with ρ) will yield static equilibrium equations. Description of solutions procedures for static linear and nonlinear problems are described in some detail in chapter 107. In addition to that, solution procedures for dynamic, linear and nonlinear problems are described in some detail in chapter 108.

A note on the final form of the tensors used is in order. In order to use readily available system of equation solvers equation (102.36) will be rewritten in the following form:

$$M_{PQ}\,\ddot{u}_P + K_{PQ}\,\bar{u}_P = F_Q \quad P, Q = 1, 2, \ldots, (\#ofDOFs)N \tag{102.38}$$

where M_{PQ} is system mass matrix, K_{PQ} is system stiffness matrix and F_Q is the loading vector. Matrix form of equation 102.36, presented as equation 102.38 is obtained flattening the system mass tensor M_{AacB}, system stiffness tensor K_{AacB}, unknown acceleration tensor $\ddot{u}_{Bc}$, unknown displacement tensor $\bar{u}_{Bc}$ and the system loading tensor F_{Aa}. Flattening from the fourth order mass/stiffness tensors to two-dimensional mass/stiffness matrix is done by simply performing appropriate (re–) numbering of nodal DOFs in each dimension. A similar approach is used for unknown accelerations/displacements and loadings.

102.3.1 Static Analysis: Internal and External Loads.

Internal and external loading tensors is defined as:

$$(f_{Ia})_{int} = \bigcup_{(m)} {}^{(m)}K_{IacJ}\ \bar{u}_{Jc} = \bigcup_{m} \int_{V^m} \sigma_{ab}\ H_{I,b}\ dV^m \tag{102.39}$$

$$(f_{Ia})_{ext} = \bigcup_{m} {}^{(m)}F^{B}_{Ia} + \bigcup_{m} {}^{(m)}F^{S}_{Ia} + \bigcup_{m} {}^{(m)}F^{\epsilon^0_{mn}}_{Ia} - \bigcup_{m} {}^{(m)}F^{\sigma^0_{mn}}_{Ia} \tag{102.40}$$

where $(f_{Ia})_{int}$ is the internal force tensor and $(f_{Ia})_{ext}$ is the external force tensor. Equilibrium is obtained when residual:

$$r_{Ia}(\bar{u}_{Jc}, \lambda) = (f_{Ia}\,(\bar{u}_{Jc}))_{int} - \lambda\,(f_{Ia})_{ext} \tag{102.41}$$

is equal to zero, $\mathbf{r}(\mathbf{u}, \lambda) = 0$. The same equation in flattened form yields:

$$\mathbf{r}(\mathbf{u}, \lambda) = \mathbf{f}_{int}(\mathbf{u}) - \lambda \mathbf{f}_{ext} = 0 \tag{102.42}$$

102.4 Isoparametric Solid Finite Elements

102.4.1 8 Node Brick

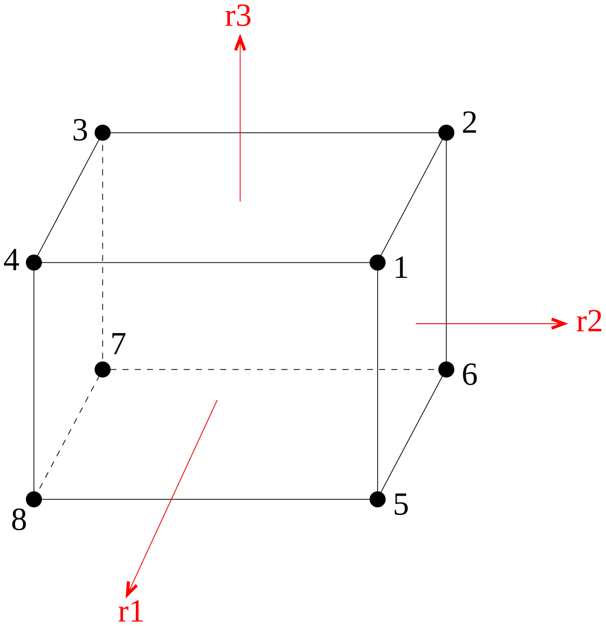


Figure 102.3: 8 node brick element

Table 102.1: Values of r_1, r_2, and r_3 at each of the eight nodes

Node	r_1	r_2	r_3
1	+1	+1	+1
2	-1	+1	+1
3	-1	-1	+1
4	+1	-1	+1
5	+1	+1	-1
6	-1	+1	-1
7	-1	-1	-1
8	+1	-1	-1

Shape function of the nodes which i indicates the node number:

$$N_i^{(e)} = \frac{1}{8}(1 + r_1(r_1)_i)(1 + r_2(r_2)_i)(1 + r_3(r_3)_i) \tag{102.43}$$

102.4.2 Collapsed 8 Node Brick

It is sometimes required to mesh finite element models using collapsed brick elements. Collapsed brick elements are finite elements that do not feature all 8 nodes, rather some nodes are merged. This is done to help generate meshes for complicated geometries where it is impossible to rely on solid bricks with eight (8) nodes only. For example, SASSI2000 (System for Analysis of Soil-Structure Interaction) program (Ostadan, 2007) uses such elements. For example, solid elements with 7, 6, and 5 nodes are used extensively and are created by collapsing/combining nodes of 8 node brick into solid elements with 7, 6 or 5 nodes. There are three types of collapsed SASSI 8 node brick element, as shown in Figure 102.4.

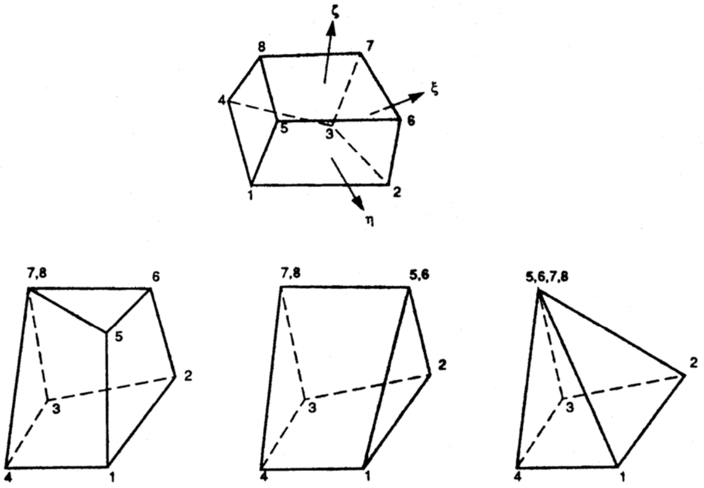


Figure 102.4: Three types of collapsed brick elements in SASSI (Ostadan, 2007).

Verification tests for collapsed brick finite elements are provided in verification section ?? on page 1547.

102.4.3 20 Node Brick

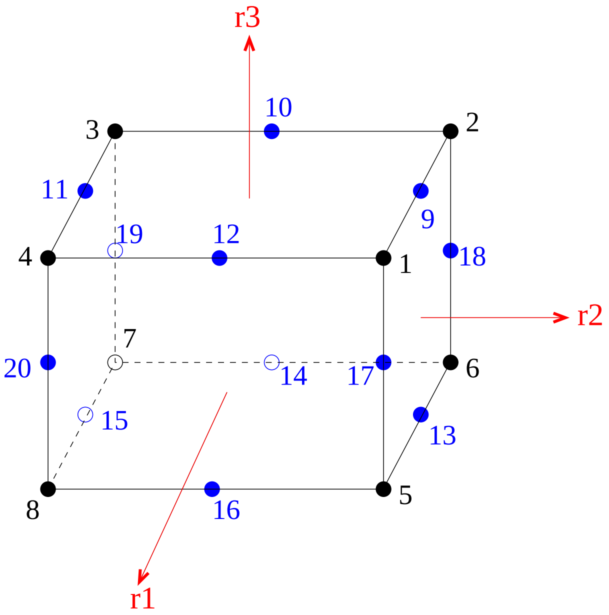


Figure 102.5: 20 node brick element

Table 102.2: Values of r_1, r_2, and r_3 at each of the 9^{th} to 20^{th} nodes

Node	r_1	r_2	r_3
9	0	+1	+1
10	-1	0	+1
11	0	-1	+1
12	+1	0	+1
13	0	+1	-1
14	-1	0	-1
15	0	-1	-1
16	+1	0	-1
17	+1	+1	0
18	-1	+1	0
19	-1	-1	0
20	+1	-1	0

Shape function of the 8 corner nodes (1 to 8) which i indicates the node number:

$$N_i^{(e)} = \frac{1}{8}(1 + r_1(r_1)_i)(1 + r_2(r_2)_i)(1 + r_3(r_3)_i)(r_1(r_1)_i + r_2(r_2)_i + r_3(r_3)_i - 2) \tag{102.44}$$

Shape function of the node numbers 9, 11, 13, and 15 which i indicates the node number:

$$N_i^{(e)} = \frac{1}{4}(1 - r_1^2)(1 + r_2(r_2)_i)(1 + r_3(r_3)_i) \tag{102.45}$$

Shape function of the node numbers 10, 12, 14, and 16 which i indicates the node number:

$$N_i^{(e)} = \frac{1}{4}(1 - r_2^2)(1 + r_1(r_1)_i)(1 + r_3(r_3)_i) \tag{102.46}$$

Shape function of the node numbers 17, 18, 19, and 20 which i indicates the node number:

$$N_i^{(e)} = \frac{1}{4}(1 - r_3^2)(1 + r_1(r_1)_i)(1 + r_2(r_2)_i) \tag{102.47}$$

102.4.4 27 Node Brick

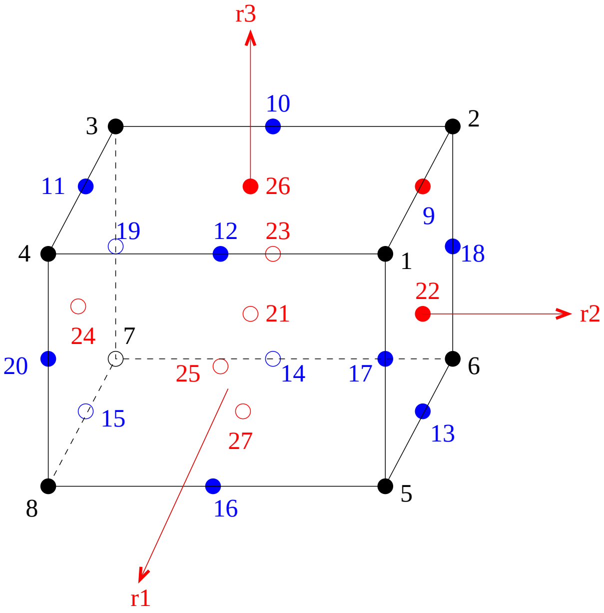


Figure 102.6: 27 node brick element

Table 102.3: Values of r_1, r_2, and r_3 at each of the 21^{th} to 27^{th} nodes

Node	r_1	r_2	r_3
21	0	0	0
22	0	$+1$	0
23	-1	0	0
24	0	-1	0
25	$+1$	0	0
26	0	0	$+1$
27	0	0	-1

Shape function of the 8 corner nodes (1 to 8) which i indicates the node number:

$$N_i^{(e)} = \frac{1}{8}(1 + r_1(r_1)_i)(1 + r_2(r_2)_i)(1 + r_3(r_3)_i)(r_1(r_1)_i)(r_2(r_2)_i)(r_3(r_3)_i) \tag{102.48}$$

Shape function of the node numbers 9, 11, 13, and 15 which i indicates the node number:

$$N_i^{(e)} = \frac{1}{4}(1 - r_1^2)(1 + r_2(r_2)_i)(1 + r_3(r_3)_i)(r_2(r_2)_i)(r_3(r_3)_i) \tag{102.49}$$

Shape function of the node numbers 10, 12, 14, and 16 which i indicates the node number:

$$N_i^{(e)} = \frac{1}{4}(1 + r_1(r_1)_i)(1 - r_2^2)(1 + r_3(r_3)_i)(r_1(r_1)_i)(r_3(r_3)_i) \tag{102.50}$$

Shape function of the node numbers 17, 18, 19, and 20 which i indicates the node number:

$$N_i^{(e)} = \frac{1}{4}(1 + r_1(r_1)_i)(1 + r_2(r_2)_i)(1 - r_3^2)(r_1(r_1)_i)(r_2(r_2)_i) \tag{102.51}$$

Shape function of the node number 21:

$$N_{21}^{(e)} = (1 - r_1^2)(1 - r_2^2)(1 - r_3^2) \tag{102.52}$$

Shape function of the node numbers 22 and 24 which i indicates the node number:

$$N_i^{(e)} = \frac{1}{2}(1 - r_1^2)(1 + r_2(r_2)_i)(1 - r_3^2)(r_2(r_2)_i) \tag{102.53}$$

Shape function of the node numbers 23 and 25 which i indicates the node number:

$$N_i^{(e)} = \frac{1}{2}(1 + r_1(r_1)_i)(1 - r_2^2)(1 - r_3^2)(r_1(r_1)_i) \tag{102.54}$$

Shape function of the node numbers 26 and 27 which i indicates the node number:

$$N_i^{(e)} = \frac{1}{2}(1 - r_1^2)(1 - r_2^2)(1 + r_3(r_3)_i)(r_3(r_3)_i) \tag{102.55}$$

102.4.5 Isoparametric 8 – 20 Node Finite Element

The basic procedure in the isoparametric[14] finite element formulation is to express the element coordinates and element displacements in the form of interpolations using the local three dimensional[15] coordinate system of the element. Considering the general 3D element, the coordinate interpolations, using indicial notation[16] are:

---

[14] name isoparametric comes from the fact that both displacements and coordinates are defined in terms of nodal values. Superparametric and subparametric finite elements exists also.

[15] in the case of element presented here, that is isoparametric 8 – 20 node finite element.

[16] Einstein's summation rule is implied unless stated differently, all lower case indices $(i, j, p, q, m, n, o, r, s, t, \ldots)$ can have values of 1, 2, 3, and values for capital letter indices will be specified where need be.

$$x_i = H_A\left(r_k\right)\,\bar{x}_{Ai} \tag{102.56}$$

where $A = 1, 2, \ldots, n$ and n is the total number of nodes associated with that specific element, $\bar{x}_{Ai}$ is the i-th coordinate of node A, $i = 1, 2, 3$, $k = 1, 2, 3$ and H_A are the interpolation functions defined in local coordinate system of the element, with variables r_1, r_2 and r_3 varying from -1 to $+1$.

The interpolation functions H_A for the isoparametric 8–20 node are the so called serendipity interpolation functions mainly because they were derived by inspection. For the finite element with nodes numbered as in Figure (102.7) they are given[17] in the following set of formulae:

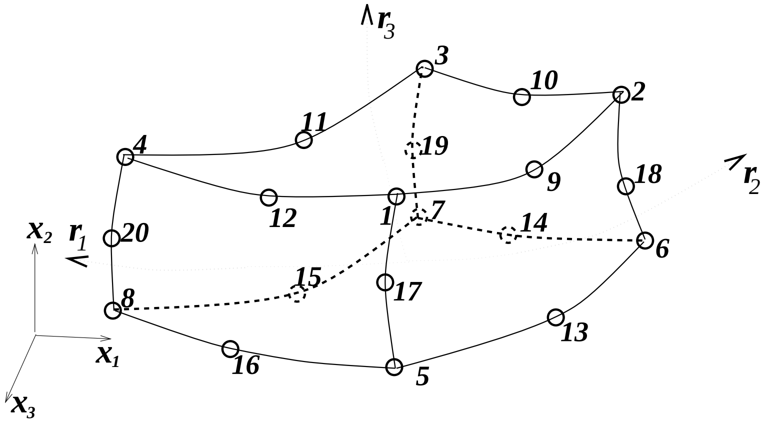


Figure 102.7: Isoparametric 8–20 node brick element in global and local coordinate systems

$$H_{20} = \frac{isp\,(20)\,\left(1+r_1\right)\left(1-r_2\right)\left(1-r_3^2\right)}{4} \qquad H_{19} = \frac{isp\,(19)\,\left(1-r_1\right)\left(1-r_2\right)\left(1-r_3^2\right)}{4}$$

$$H_{18} = \frac{isp\,(18)\,\left(1-r_1\right)\left(1+r_2\right)\left(1-r_3^2\right)}{4} \qquad H_{17} = \frac{isp\,(17)\,\left(1+r_1\right)\left(1+r_2\right)\left(1-r_3^2\right)}{4}$$

$$H_{16} = \frac{isp\,(16)\,\left(1+r_1\right)\left(1-r_2^2\right)\left(1-r_3\right)}{4} \qquad H_{15} = \frac{isp\,(15)\,\left(1-r_1^2\right)\left(1-r_2\right)\left(1-r_3\right)}{4}$$

$$H_{14} = \frac{isp\,(14)\,\left(1-r_1\right)\left(1-r_2^2\right)\left(1-r_3\right)}{4} \qquad H_{13} = \frac{isp\,(13)\,\left(1-r_1^2\right)\left(1+r_2\right)\left(1-r_3\right)}{4}$$

$$H_{12} = \frac{isp\,(12)\,\left(1+r_1\right)\left(1-r_2^2\right)\left(1+r_3\right)}{4} \qquad H_{11} = \frac{isp\,(11)\,\left(1-r_1^2\right)\left(1-r_2\right)\left(1+r_3\right)}{4}$$

$$H_{10} = \frac{isp\,(10)\,\left(1-r_1\right)\left(1-r_2^2\right)\left(1+r_3\right)}{4} \qquad H_{9} = \frac{isp\,(9)\,\left(1-r_1^2\right)\left(1+r_2\right)\left(1+r_3\right)}{4}$$

---

[17] for more details see Bathe (1982).

$$
\begin{aligned}
H_8 &= \frac{(1+r_1)\,(1-r_2)\,(1-r_3)}{8} + \frac{-H_{15}-H_{16}-H_{20}}{2} \\
H_7 &= \frac{(1-r_1)\,(1-r_2)\,(1-r_3)}{8} + \frac{-H_{14}-H_{15}-H_{19}}{2} \\
H_6 &= \frac{(1-r_1)\,(1+r_2)\,(1-r_3)}{8} + \frac{-H_{13}-H_{14}-H_{18}}{2} \\
H_5 &= \frac{(1+r_1)\,(1+r_2)\,(1-r_3)}{8} + \frac{-H_{13}-H_{16}-H_{17}}{2} \\
H_4 &= \frac{(1+r_1)\,(1-r_2)\,(1+r_3)}{8} + \frac{-H_{11}-H_{12}-H_{20}}{2} \\
H_3 &= \frac{(1-r_1)\,(1-r_2)\,(1+r_3)}{8} + \frac{-H_{10}-H_{11}-H_{19}}{2} \\
H_2 &= \frac{(1-r_1)\,(1+r_2)\,(1+r_3)}{8} + \frac{-H_{10}-H_{18}-H_{9}}{2} \\
H_1 &= \frac{(1+r_1)\,(1+r_2)\,(1+r_3)}{8} + \frac{-H_{12}-H_{17}-H_{9}}{2}
\end{aligned}
$$

where r_1, r_2 and r_3 are the axes of natural, local, curvilinear coordinate system and $isp\,(nod\_num)$ is boolean function that returns +1 if node number (nod_num) is present and 0 if node number (nod_num) is not present.

To be able to evaluate various important element tensors[18], to calculate the strain–displacement transformation tensor[19] is needed. The element strains are obtained in terms of derivatives of element displacements with respect to the local coordinate system. Because the element displacements are defined in the local coordinate system, there is a need to relate global x_1, x_2 and x_3 derivatives to the r_1, r_2 and r_3 derivatives. In order to obtain derivatives with respect to global coordinate system, i.e. $\frac{\partial}{\partial x_a}$, use chain rule for differentiation in the following form:

$$
\frac{\partial}{\partial x_k} = \frac{\partial r_a}{\partial x_k}\frac{\partial}{\partial r_a} = J^{-1}_{ak}\,\frac{\partial}{\partial r_a} \tag{102.57}
$$

while the inverse relation is:

$$
\frac{\partial}{\partial r_k} = \frac{\partial x_a}{\partial r_k}\frac{\partial}{\partial x_a} = J_{ak}\,\frac{\partial}{\partial x_a} \tag{102.58}
$$

where J_{ak} is the Jacobian operator relating local coordinate derivatives to the global coordinate derivatives:

---

[18] i.e. ${}^{(m)}K_{IacJ}$, ${}^{(m)}F^{B}_{Ia}$, ${}^{(m)}F^{S}_{Ia}$, ${}^{(m)}F^{\epsilon^0_{mn}}_{Ia}$, ${}^{(m)}F^{\sigma^0_{mn}}_{Ia}$, that are defined in chapter (102.3).

[19] from the equation $\hat{\epsilon}_{ab} = \frac{1}{2}\left(\left(H_{I,b}\,\bar{u}_{Ia}\right) + \left(H_{I,a}\,\bar{u}_{Ib}\right)\right)$

$$J_{ak} = \frac{\partial x_a}{\partial r_k} = \begin{bmatrix} \frac{\partial x_1}{\partial r_1} & \frac{\partial x_2}{\partial r_1} & \frac{\partial x_3}{\partial r_1} \\ \\ \frac{\partial x_1}{\partial r_2} & \frac{\partial x_2}{\partial r_2} & \frac{\partial x_3}{\partial r_2} \\ \\ \frac{\partial x_1}{\partial r_3} & \frac{\partial x_2}{\partial r_3} & \frac{\partial x_3}{\partial r_3} \end{bmatrix} \tag{102.59}$$

The existence of equation (102.57) requires that the inverse of J_{ak} exists and that inverse exists provided that there is a one–to–one[20] correspondence between the local and the global coordinates of element.

It should be pointed out that except for the very simple cases, volume and surface element tensor[21] integrals are evaluated by means of numerical integration[22] Numerical integration rules is quite a broad subject and will not be covered here[23].

102.4.6 Isoparametric 8 - 27 Node Finite Element

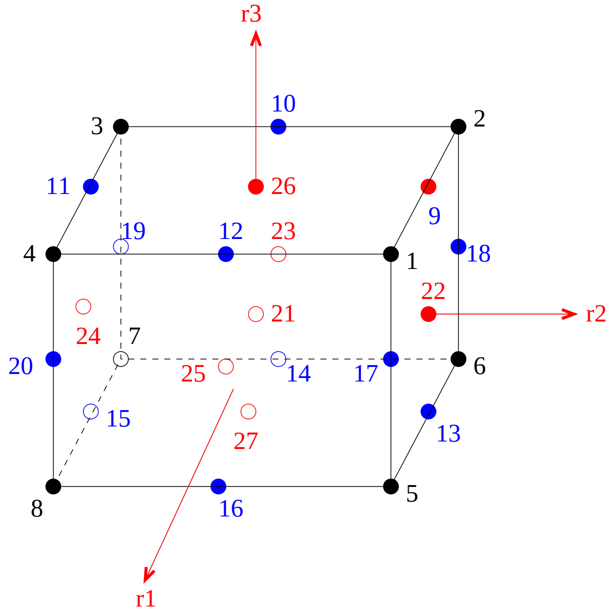


Figure 102.8: 8-27 variable node brick element

---

[20] unique.

[21] as defined in chapter (102.3) by equations (102.29), (102.30), (102.31), (102.32) and (102.33).

[22] Gauss–Legendre, Newton–Coates, Lobatto are among the most used integration rules.

[23] nice explanation with examples is given in Bathe (1982).

$$
\begin{aligned}
H_1 &= \frac{(1+r_1)(1+r_2)(1+r_3)}{8} - \frac{H_9+H_{12}+H_{17}}{2} - \frac{H_{22}+H_{25}+H_{26}}{4} - \frac{H_{21}}{8}\\
H_2 &= \frac{(1-r_1)(1+r_2)(1+r_3)}{8} - \frac{H_9+H_{10}+H_{18}}{2} - \frac{H_{22}+H_{23}+H_{26}}{4} - \frac{H_{21}}{8}\\
H_3 &= \frac{(1-r_1)(1-r_2)(1+r_3)}{8} - \frac{H_{10}+H_{11}+H_{19}}{2} - \frac{H_{23}+H_{24}+H_{26}}{4} - \frac{H_{21}}{8}\\
H_4 &= \frac{(1+r_1)(1-r_2)(1+r_3)}{8} - \frac{H_{11}+H_{12}+H_{20}}{2} - \frac{H_{24}+H_{25}+H_{26}}{4} - \frac{H_{21}}{8}\\
H_5 &= \frac{(1+r_1)(1+r_2)(1-r_3)}{8} - \frac{H_{13}+H_{16}+H_{17}}{2} - \frac{H_{22}+H_{25}+H_{27}}{4} - \frac{H_{21}}{8}\\
H_6 &= \frac{(1-r_1)(1+r_2)(1-r_3)}{8} - \frac{H_{13}+H_{14}+H_{18}}{2} - \frac{H_{22}+H_{23}+H_{27}}{4} - \frac{H_{21}}{8}\\
H_7 &= \frac{(1-r_1)(1-r_2)(1-r_3)}{8} - \frac{H_{14}+H_{15}+H_{19}}{2} - \frac{H_{23}+H_{24}+H_{27}}{4} - \frac{H_{21}}{8}\\
H_8 &= \frac{(1+r_1)(1-r_2)(1-r_3)}{8} - \frac{H_{15}+H_{16}+H_{20}}{2} - \frac{H_{24}+H_{25}+H_{27}}{4} - \frac{H_{21}}{8}
\end{aligned}
$$

$$
\begin{aligned}
H_9 &= \frac{1}{4}(1-r_1^2)(1+r_2)(1+r_3) - \frac{H_{22}+H_{26}}{2} - \frac{H_{21}}{4}\\
H_{10} &= \frac{1}{4}(1-r_2^2)(1-r_1)(1+r_3) - \frac{H_{23}+H_{26}}{2} - \frac{H_{21}}{4}\\
H_{11} &= \frac{1}{4}(1-r_1^2)(1-r_2)(1+r_3) - \frac{H_{24}+H_{26}}{2} - \frac{H_{21}}{4}\\
H_{12} &= \frac{1}{4}(1-r_2^2)(1+r_1)(1+r_3) - \frac{H_{25}+H_{26}}{2} - \frac{H_{21}}{4}\\
H_{13} &= \frac{1}{4}(1-r_1^2)(1+r_2)(1-r_3) - \frac{H_{22}+H_{27}}{2} - \frac{H_{21}}{4}\\
H_{14} &= \frac{1}{4}(1-r_2^2)(1-r_1)(1-r_3) - \frac{H_{23}+H_{27}}{2} - \frac{H_{21}}{4}\\
H_{15} &= \frac{1}{4}(1-r_1^2)(1-r_2)(1-r_3) - \frac{H_{24}+H_{27}}{2} - \frac{H_{21}}{4}\\
H_{16} &= \frac{1}{4}(1-r_2^2)(1+r_1)(1-r_3) - \frac{H_{25}+H_{27}}{2} - \frac{H_{21}}{4}
\end{aligned}
$$

$$
\begin{aligned}
H_{17} &= \frac{1}{4}(1-r_3^2)(1+r_1)(1+r_2) - \frac{H_{22}+H_{25}}{2} - \frac{H_{21}}{4}\\
H_{18} &= \frac{1}{4}(1-r_3^2)(1-r_1)(1+r_2) - \frac{H_{22}+H_{23}}{2} - \frac{H_{21}}{4}\\
H_{19} &= \frac{1}{4}(1-r_3^2)(1-r_1)(1-r_2) - \frac{H_{23}+H_{24}}{2} - \frac{H_{21}}{4}\\
H_{20} &= \frac{1}{4}(1-r_3^2)(1+r_1)(1-r_2) - \frac{H_{24}+H_{25}}{2} - \frac{H_{21}}{4}
\end{aligned}
$$

$$H_{21} = (1-r_1^2)(1-r_2^2)(1-r_3^2)$$

$$
\begin{aligned}
H_{22} &= \frac{1}{2}(1-r_1^2)(1+r_2)(1-r_3^2)r_2 \\
H_{23} &= -\frac{1}{2}(1-r_1)(1-r_2^2)(1-r_3^2)r_1 \\
H_{24} &= -\frac{1}{2}(1-r_1^2)(1-r_2)(1-r_3^2)r_2 \\
H_{25} &= \frac{1}{2}(1+r_1)(1-r_2^2)(1-r_3^2)r_1 \\
H_{26} &= \frac{1}{2}(1-r_1^2)(1-r_2^2)(1+r_3)r_3 \\
H_{27} &= -\frac{1}{2}(1-r_1^2)(1-r_2^2)(1-r_3)r_3
\end{aligned}
$$

102.4.7 Surface Loads for Solid Bricks

To apply surface load on brick elements, equivalent nodal forces have to be applied instead of the surface load. The equivalent force of the i-th node F_i is given by the following equation with shape function H_i and load distribution function f .

$$F_i = \int_S f H_i ds \tag{102.60}$$

Assuming that the load distribution is uniform

$$F_i = f \int_S H_i ds \tag{102.61}$$

Furthermore, when the magnitude of the load per unit area is 1, and the size of the element is $1 \times 1 \times 1$, equivalent nodal forces are given as shown in Figure 102.9 for 8 node brick element, 20 brick element, and 27 nodes brick element.

Figure 102.9 shows cases of normal loads on vertical upper surface (with nodes: 1, 2, 3, 4 for 8 node brick; 1, 2, 3, 4, 9, 10, 11, and 12 for 20 node brick; and 1, 2, 3, 4, 9, 10, 11, 12 and 26 for the 27 node brick).

Nodal loads from uniform surface loads for 27 node brick are obtained as:

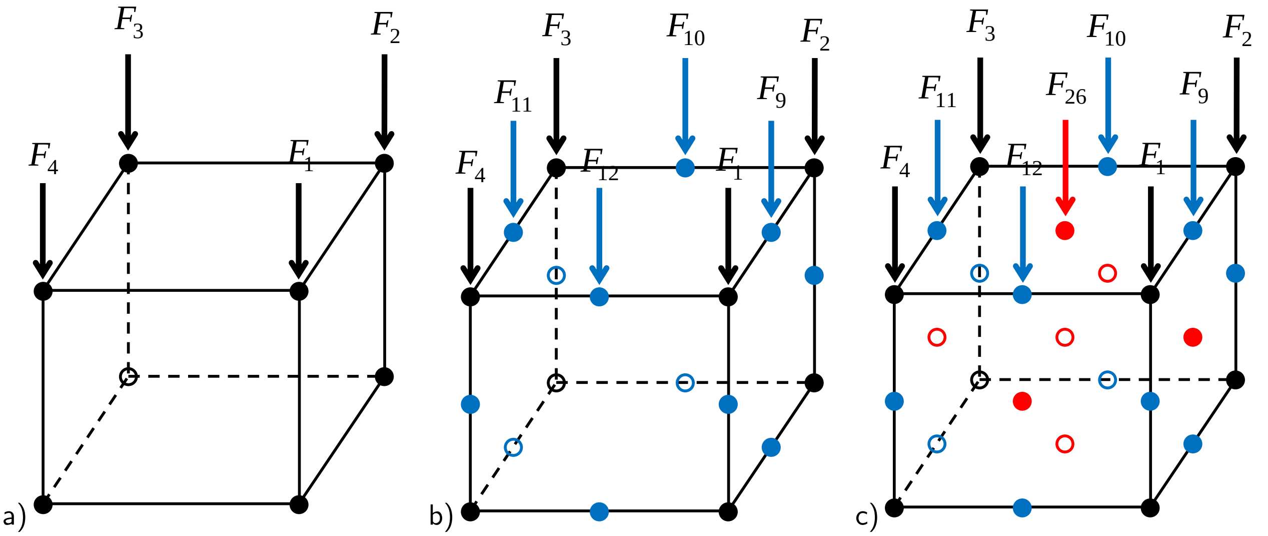


Figure 102.9: Nodal loads for brick elements: (a) $F_1 = F_2 = F_3 = F_4 = +1/4$; (b) $F_1 = F_2 = F_3 = F_4 = -\frac{1}{12}$, $F_9 = F_{10} = F_{11} = F_{12} == +\frac{1}{3}$; (c) $F_1 = F_2 = F_3 = F_4 = +\frac{1}{36}$, $F_9 = F_{10} = F_{11} = F_{12} == +\frac{1}{9}, F_{36} = \frac{4}{9}$.

- for nodes 1, 2, 3, and 4, $N_i^{(e)} = \frac{1}{8}(1 + r_1(r_1)_i)(1 + r_2(r_2)_i)(1 + r_3(r_3)_i)(r_1(r_1)_i)(r_2(r_2)_i)(r_3(r_3)_i)$

$$\begin{aligned}
\int_{-1}^{+1} H_i dS &= \\
&= \frac{1}{8}(1 + r_3(r_3)_i)(r_3(r_3)_i) \int_{-1}^{+1}\int_{-1}^{+1} (1 + r_1(r_1)_i)(1 + r_2(r_2)_i)(r_1(r_1)_i)(r_2(r_2)_i)(r_2(r_2)_i) dr_1 dr_2 \\
&= \frac{1}{8}(1 + r_3(r_3)_i)(r_3(r_3)_i)((r_1)_i)^2((r_2)_i)^2(\frac{2}{3})^2 \\
&= \frac{1}{18}(1 + r_3(r_3)_i)(r_3(r_3)_i)((r_1)_i)^2((r_2)_i)^2 \qquad (102.62)
\end{aligned}$$

- for nodes 9, 10, 11 and 12, $N_i^{(e)} = \frac{1}{4}(1 - r_1^2)(1 + r_2(r_2)_i)(1 + r_3(r_3)_i)(r_2(r_2)_i)(r_3(r_3)_i)$

$$\begin{aligned}
\int_{-1}^{+1} H_i dS &= \\
&= \frac{1}{4}(1 + r_3(r_3)_i)(r_3(r_3)_i) \int_{-1}^{+1} (1 - r_1^2)(1 + r_2(r_2)_i)(r_2(r_2)_i) dr_1 dr_2 \\
&= \frac{1}{8}(1 + r_3(r_3)_i)(r_3(r_3)_i)((r_2)_i)^2(\frac{4}{3})(\frac{2}{3}) \\
&= \frac{2}{9}(1 + r_3(r_3)_i)(r_3(r_3)_i)((r_2)_i)^2 \qquad (102.63)
\end{aligned}$$

- for nodes 26 $N_i^{(e)} = \frac{1}{2}(1-r_1^2)(1-r_2^2)(1+r_3(r_3)_i)(r_3(r_3)_i)$

$$
\begin{aligned}
\int_{-1}^{+1} H_i dS &= \\
&= \frac{1}{4}(1+r_3(r_3)_i)(r_3(r_3)_i)\int_{-1}^{+1}\int_{-1}^{+1}(1-r_1^2)(1-r_2^2)dr_1dr_2 \\
&= \frac{1}{8}(1+r_3(r_3)_i)(r_3(r_3)_i)(\frac{4}{3})(\frac{4}{3}) \\
&= \frac{8}{9}(1+r_3(r_3)_i)(r_3(r_3)_i) \qquad (102.64)
\end{aligned}
$$

102.5 Numerical Integration for Solid Brick Elements

Gauss integration rule, see Bathe (1996), section 5.5.3. While using the regular Newton-Coates integration formula, one uses $(n+1)$ equally spaced points to integrate exactly polynomial of order n. On the other hand, while using the Gauss integration formula, one uses n unequally spaced points to integrate exactly polynomial of order $(2n-1)$.

102.6 Two Node, 3D Truss Finite Element

Bathe and Wilson (1976); Bathe (1982)

102.7 3D Beam-Column Finite Element, 12 Degrees of Freedom

Bathe and Wilson (1976); Bathe (1982); Przemieniecki (1985)

Stiffness Matrix: Equation 102.65

Mass Matrix: Equation 102.66

$$
[K] = \begin{bmatrix}
\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} & 0 & -\frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} \\
0 & 0 & \frac{12EI_y}{L^3} & 0 & -\frac{6EI_y}{L^2} & 0 & 0 & 0 & -\frac{12EI_y}{L^3} & 0 & -\frac{6EI_y}{L^2} & 0 \\
0 & 0 & 0 & \frac{GJ_x}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{GJ_x}{L} & 0 & 0 \\
0 & 0 & -\frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 \\
0 & \frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L} & 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} \\
-\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\
0 & -\frac{12EI_z}{L^3} & 0 & 0 & 0 & -\frac{6EI_z}{L^2} & 0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & -\frac{6EI_z}{L^2} \\
0 & 0 & -\frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 & 0 & 0 & \frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 \\
0 & 0 & 0 & -\frac{GJ_x}{L} & 0 & 0 & 0 & 0 & 0 & \frac{GJ_x}{L} & 0 & 0 \\
0 & 0 & -\frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 \\
0 & \frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} & 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L}
\end{bmatrix} \qquad (102.65)
$$

$$
[M] = \rho A L
\begin{bmatrix}
\frac{1}{3} & 0 & 0 & 0 & 0 & 0 & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{13}{35} + \frac{6I_z}{5Al^2} & 0 & 0 & 0 & \frac{11l}{210} + \frac{I_z}{10Al} & 0 & \frac{9}{70} - \frac{6I_z}{5Al^2} & 0 & 0 & 0 & -\frac{13l}{420} + \frac{I_z}{10Al} \\
0 & 0 & \frac{13}{35} + \frac{6I_y}{5Al^2} & 0 & -\frac{11l}{210} - \frac{I_y}{10Al} & 0 & 0 & 0 & \frac{9}{70} - \frac{6I_y}{5Al^2} & 0 & \frac{13l}{420} - \frac{I_y}{10Al} & 0 \\
0 & 0 & 0 & \frac{J_x}{3A} & 0 & 0 & 0 & 0 & 0 & \frac{J_x}{6A} & 0 & 0 \\
0 & 0 & -\frac{11l}{210} - \frac{I_y}{10Al} & 0 & \frac{l^2}{105} + \frac{2I_y}{15A} & 0 & 0 & 0 & -\frac{13l}{420} + \frac{I_y}{10Al} & 0 & -\frac{l^2}{140} - \frac{I_y}{30A} & 0 \\
0 & \frac{11l}{210} + \frac{I_z}{10Al} & 0 & 0 & 0 & \frac{l^2}{105} + \frac{2I_z}{15A} & 0 & \frac{13l}{420} - \frac{I_z}{10Al} & 0 & 0 & 0 & -\frac{l^2}{140} - \frac{I_z}{30A} \\
\frac{1}{6} & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{9}{70} - \frac{6I_z}{5Al^2} & 0 & 0 & 0 & \frac{13l}{420} - \frac{I_z}{10Al} & 0 & \frac{13}{35} + \frac{6I_z}{5Al^2} & 0 & 0 & 0 & -\frac{11l}{210} - \frac{I_z}{10Al} \\
0 & 0 & \frac{9}{70} - \frac{6I_y}{5Al^2} & 0 & -\frac{13l}{420} + \frac{I_y}{10Al} & 0 & 0 & 0 & \frac{13}{35} + \frac{6I_y}{5Al^2} & 0 & \frac{11l}{210} + \frac{I_y}{10Al} & 0 \\
0 & 0 & 0 & \frac{J_x}{6A} & 0 & 0 & 0 & 0 & 0 & \frac{J_x}{3A} & 0 & 0 \\
0 & 0 & \frac{13l}{420} - \frac{I_y}{10Al} & 0 & -\frac{l^2}{140} - \frac{I_y}{30A} & 0 & 0 & 0 & \frac{11l}{210} + \frac{I_y}{10Al} & 0 & \frac{l^2}{105} + \frac{2I_y}{15A} & 0 \\
0 & -\frac{13l}{420} + \frac{I_z}{10Al} & 0 & 0 & 0 & -\frac{l^2}{140} - \frac{I_z}{30A} & 0 & -\frac{11l}{210} - \frac{I_z}{10Al} & 0 & 0 & 0 & \frac{l^2}{105} + \frac{2I_z}{15A}
\end{bmatrix}
\qquad (102.66)
$$

102.8 3D Beam-Column Finite Element, 9 Degrees of Freedom

Przemieniecki (1985)

Condensation Formulation: Equations 102.67 to 102.73

Rearranged 12dof Stiffness Matrix: Equation 102.74

K_{rr} part of stiffness matrix: Equation 102.75

K_{rc} part of stiffness matrix: Equation 102.76

K_{cr} part of stiffness matrix: Equation 102.77

K_{cc} part of stiffness matrix: Equation 102.78

Stiffness Matrix: Equation 102.79

T Matrix: Equation 102.80

Rearranged Mass Matrix: Equation 102.81

Mass Matrix: Equation 102.82

$$\begin{bmatrix} k_{rr} & k_{rc} \\ k_{cr} & k_{cc} \end{bmatrix} \begin{Bmatrix} d_r \\ d_c \end{Bmatrix} = \begin{Bmatrix} r_r \\ r_c \end{Bmatrix} \tag{102.67}$$

$$\left(\left[\, k_{rr} \,\right] - \left[\, K_{rc} \,\right] \left[\, K_{cc} \,\right]^{-1} \left[\, K_{cr} \,\right] \right) \left\{\, d_r \,\right\} = \left\{\, r_r \,\right\} - \left[\, K_{rc} \,\right] \left[\, K_{cc} \,\right]^{-1} \left\{\, r_c \,\right\} \tag{102.68}$$

$$[k_{condensed}] = \left[\, k_{rr} \,\right] - \left[\, K_{rc} \,\right] \left[\, K_{cc} \,\right]^{-1} \left[\, K_{cr} \,\right] \tag{102.69}$$

$$r_{condensed} = \left\{\, r_r \,\right\} - \left[\, K_{rc} \,\right] \left[\, K_{cc} \,\right]^{-1} \left\{\, r_c \,\right\} \tag{102.70}$$

$$[T] = \begin{bmatrix} I \\ -\left[\, K_{cc} \,\right]^{-1} \left[\, K_{cr} \,\right] \end{bmatrix} \tag{102.71}$$

$$[K_{condensed}] = [T]^T [K][T] \tag{102.72}$$

$K_{condensed}$ should give the same results using either method.

$$[M_{condensed}] = [T]^T [M][T] \tag{102.73}$$

$$
[K_{rearranged}] \;=\;
\begin{bmatrix}
\frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{12EI_z}{L^3} & 0 & 0 & -\frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} & 0 & 0 & \frac{6EI_z}{L^2} \\
0 & 0 & \frac{12EI_y}{L^3} & 0 & 0 & -\frac{12EI_y}{L^3} & 0 & -\frac{6EI_y}{L^2} & 0 & 0 & -\frac{6EI_y}{L^2} & 0 \\
-\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\frac{12EI_z}{L^3} & 0 & 0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & -\frac{6EI_z}{L^2} & 0 & 0 & -\frac{6EI_z}{L^2} \\
0 & 0 & -\frac{12EI_y}{L^3} & 0 & 0 & \frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 & 0 & \frac{6EI_y}{L^2} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{GJ_x}{L} & 0 & 0 & -\frac{GJ_x}{L} & 0 & 0 \\
0 & 0 & -\frac{6EI_y}{L^2} & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 & 0 & \frac{2EI_y}{L} & 0 \\
0 & \frac{6EI_z}{L^2} & 0 & 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L} & 0 & 0 & \frac{2EI_z}{L} \\
0 & 0 & 0 & 0 & 0 & 0 & -\frac{GJ_x}{L} & 0 & 0 & \frac{GJ_x}{L} & 0 & 0 \\
0 & 0 & -\frac{6EI_y}{L^2} & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 & 0 & \frac{4EI_y}{L} & 0 \\
0 & \frac{6EI_z}{L^2} & 0 & 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} & 0 & 0 & \frac{4EI_z}{L}
\end{bmatrix}
\qquad (102.74)
$$

$$[K_{rr}] = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12EI_z}{L^3} & 0 & 0 & -\frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} \\ 0 & 0 & \frac{12EI_y}{L^3} & 0 & 0 & -\frac{12EI_y}{L^3} & 0 & -\frac{6EI_y}{L^2} & 0 \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12EI_z}{L^3} & 0 & 0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & -\frac{6EI_z}{L^2} \\ 0 & 0 & -\frac{12EI_y}{L^3} & 0 & 0 & \frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{GJ_x}{L} & 0 & 0 \\ 0 & 0 & -\frac{6EI_y}{L^2} & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 \\ 0 & \frac{6EI_z}{L^2} & 0 & 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L} \end{bmatrix} \tag{102.75}$$

$$[K_{rc}] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \frac{6EI_z}{L^2} \\ 0 & -\frac{6EI_y}{L^2} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -\frac{6EI_z}{L^2} \\ 0 & \frac{6EI_y}{L^2} & 0 \\ -\frac{GJ_x}{L} & 0 & 0 \\ 0 & \frac{2EI_y}{L} & 0 \\ 0 & 0 & \frac{2EI_z}{L} \end{bmatrix} \tag{102.76}$$

$$[K_{cr}] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -\frac{GJ_x}{L} & 0 & 0 \\ 0 & 0 & -\frac{6EI_y}{L^2} & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 \\ 0 & \frac{6EI_z}{L^2} & 0 & 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} \end{bmatrix} \tag{102.77}$$

$$[K_{cc}] = \begin{bmatrix} \frac{GJ_x}{L} & 0 & 0 \\ 0 & \frac{4EI_y}{L} & 0 \\ 0 & 0 & \frac{4EI_z}{L} \end{bmatrix} \tag{102.78}$$

$$
[K_{condensed}] =
\begin{bmatrix}
\frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{3EI_z}{L^3} & 0 & 0 & -\frac{3EI_z}{L^3} & 0 & 0 & 0 & \frac{3EI_z}{L^2} \\
0 & 0 & \frac{3EIy}{L^3} & 0 & 0 & -\frac{3EI_y}{L^3} & 0 & -\frac{3EIy}{L^2} & 0 \\
-\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\
0 & -\frac{3EI_z}{L^3} & 0 & 0 & \frac{3EI_z}{L^3} & 0 & 0 & 0 & -\frac{3EI_z}{L^2} \\
0 & 0 & -\frac{3EI_y}{L^3} & 0 & 0 & \frac{3EI_y}{L^3} & 0 & \frac{3EI_y}{L^2} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -\frac{3EI_y}{L^2} & 0 & 0 & \frac{3EI_y}{L^2} & 0 & \frac{3EI_y}{L} & 0 \\
0 & \frac{3EI_z}{L^2} & 0 & 0 & -\frac{3EI_z}{L^2} & 0 & 0 & 0 & \frac{3EI_z}{L}
\end{bmatrix}
\tag{102.79}
$$

$$
[T] =
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & \frac{3}{2L} & 0 & 0 & -\frac{3}{2L} & 0 & -\frac{1}{2} & 0 \\
0 & -\frac{3}{2L} & 0 & 0 & \frac{3}{2L} & 0 & 0 & 0 & -\frac{1}{2}
\end{bmatrix}
\tag{102.80}
$$

$$[M_{rearranged}] = \rho A L \begin{bmatrix}
\frac{1}{3} & 0 & 0 & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{13}{35} + \frac{6I_z}{5Al^2} & 0 & 0 & \frac{9}{70} - \frac{6I_z}{5Al^2} & 0 & 0 & 0 & -\frac{13l}{420} + \frac{I_z}{10Al} & 0 & 0 & \frac{11l}{210} + \frac{I_z}{10Al} \\
0 & 0 & \frac{13}{35} + \frac{6I_y}{5Al^2} & 0 & 0 & \frac{9}{70} - \frac{6I_y}{5Al^2} & 0 & \frac{13l}{420} - \frac{I_y}{10Al} & 0 & 0 & -\frac{11l}{210} - \frac{I_y}{10Al} & 0 \\
\frac{1}{6} & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{9}{70} - \frac{6I_z}{5Al^2} & 0 & 0 & \frac{13}{35} + \frac{6I_z}{5Al^2} & 0 & 0 & 0 & -\frac{11l}{210} - \frac{I_z}{10Al} & 0 & 0 & \frac{13l}{420} - \frac{I_z}{10Al} \\
0 & 0 & \frac{9}{70} - \frac{6I_y}{5Al^2} & 0 & 0 & \frac{13}{35} + \frac{6I_y}{5Al^2} & 0 & \frac{11l}{210} + \frac{I_y}{10Al} & 0 & 0 & -\frac{13l}{420} + \frac{I_y}{10Al} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{J_x}{3A} & 0 & 0 & \frac{J_x}{6A} & 0 & 0 \\
0 & 0 & \frac{13l}{420} - \frac{I_y}{10Al} & 0 & 0 & \frac{11l}{210} + \frac{I_y}{10Al} & 0 & \frac{l^2}{105} + \frac{2I_y}{15A} & 0 & 0 & -\frac{l^2}{140} - \frac{I_y}{30A} & 0 \\
0 & -\frac{13l}{420} + \frac{I_z}{10Al} & 0 & 0 & -\frac{11l}{210} - \frac{I_z}{10Al} & 0 & 0 & 0 & \frac{l^2}{105} + \frac{2I_z}{15A} & 0 & 0 & -\frac{l^2}{140} - \frac{I_z}{30A} \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{J_x}{6A} & 0 & 0 & \frac{J_x}{3A} & 0 & 0 \\
0 & 0 & -\frac{11l}{210} - \frac{I_y}{10Al} & 0 & 0 & -\frac{13l}{420} + \frac{I_y}{10Al} & 0 & -\frac{l^2}{140} - \frac{I_y}{30A} & 0 & 0 & \frac{l^2}{105} + \frac{2I_y}{15A} & 0 \\
0 & \frac{11l}{210} + \frac{I_z}{10Al} & 0 & 0 & \frac{13l}{420} - \frac{I_z}{10Al} & 0 & 0 & 0 & -\frac{l^2}{140} - \frac{I_z}{30A} & 0 & 0 & \frac{l^2}{105} + \frac{2I_z}{15A}
\end{bmatrix} \tag{102.81}$$

$$
[M_{condensed}] = \rho A L \begin{bmatrix}
\frac{1}{3} & 0 & 0 & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{33}{140} + \frac{6I_z}{5AL^2} & 0 & 0 & \frac{39}{280} - \frac{6I_z}{5AL^2} & 0 & 0 & 0 & -\frac{11L}{280} + \frac{I_z}{5AL} \\
0 & 0 & \frac{33}{140} + \frac{6I_y}{5AL^2} & 0 & 0 & \frac{39}{280} - \frac{6I_y}{5AL^2} & 0 & \frac{11L}{280} - \frac{I_y}{5AL} & 0 \\
\frac{1}{6} & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{39}{280} - \frac{6I_z}{5AL^2} & 0 & 0 & \frac{17}{35} + \frac{6I_z}{5AL^2} & 0 & 0 & 0 & -\frac{3L}{35} - \frac{I_z}{5AL} \\
0 & 0 & \frac{39}{280} - \frac{6I_y}{5AL^2} & 0 & 0 & \frac{17}{35} + \frac{6I_y}{5AL^2} & 0 & \frac{3L}{35} + \frac{I_y}{5AL} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{J_x}{A} & 0 & 0 \\
0 & 0 & \frac{11L}{280} - \frac{I_y}{5AL} & 0 & 0 & \frac{3L}{35} + \frac{I_y}{5AL} & 0 & \frac{2L^2}{105} + \frac{I_y}{5A} & 0 \\
0 & -\frac{11L}{280} + \frac{I_z}{5AL} & 0 & 0 & -\frac{3L}{35} - \frac{I_z}{5AL} & 0 & 0 & 0 & \frac{2L^2}{105} + \frac{I_z}{5A}
\end{bmatrix} \tag{102.82}
$$

102.9 Shear Beam Finite Element

102.10 Quadrilateral Shell Finite Element with 6DOFs per Node

Based on works by Bergan and Felippa (1985); Alvin et al. (1992); Felippa and Militello (1992); Felippa and Alexander (1992); Militello and Felippa (1991). The stiffness matrix for this element is obtained by averaging two quad shells made up of two ANDES triangular shells (with an alternating orientation of diagonals, Stošić (1984-2022))

102.11 Seismic Isolator and Dissipator Finite Elements

Base isolation system are used to change dynamic characteristics of seismic motions that excite structure and also to dissipate seismic energy before it excites structure. Therefore there are two main types of devices:

- Base Isolators (Kelly, 1991a,b; Toopchi-Nezhad et al., 2008; Huang et al., 2010; Vassiliou et al., 2013) are usually made of low damping (energy dissipation) elastomers and are primarily meant to change (reduce) frequencies of input motions. They are not designed nor modeled as energy dissipators.
- Base Dissipators Kelly and Hodder (1982); Fadi and Constantinou (2010); Kumar et al. (2014) are developed to dissipate seismic energy before it excites the structure. There two main types of such dissipators:
  - Elastomers made of high dissipation rubber, and
  - Frictional pendulum dissipators

Both isolators and dissipators are usually developed to work in two horizontal dimensions, while motions in vertical direction are not isolated or dissipated. This can create potential problems and need to be carefully modeled.

Modeling of base isolation and dissipation system is done using two-node finite elements of relatively short length.

102.11.1 Base Isolation Systems

Base isolation systems are modeled using linear or nonlinear elastic elements. Stiffness is provided from either tests on a full-sized base isolators, or from material characterization of rubber (and steel plates if used in a sandwich isolator construction). Depending on rubber used, a number of models can be used to develop stiffness of the device Ogden (1984); Simo and Miehe (1992); Simo and Pister (1984).

Particularly important is to properly account for vertical stiffness as vertical motions can be amplified depending on characteristics of seismic motions, structure and stiffness of the isolators Hijikata et al. (2012); Araki et al. (2009). It is also important to note that assumption of small deformation is used in most cases. In other words, the stability of the isolator, for example, overturning or rolling is not modeled. It is assumed that elastic stiffness will not suddenly change if the isolator becomes unstable (rolls or overturns).

102.11.2 Base Dissipator Systems

Base dissipator systems are modeled using inelastic (nonlinear) two node elements. There are three basic types of dissipator models used:

- High damping rubber dissipators
- Rubber dissipators with lead core
- Frictional pendulum (double or triple) dissipators

Each one is calibrated using tests done on a full dissipator. It is important to be able to take into account the influence of (an increase in) temperature on resulting behavior. Energy dissipation results in heating of devices, and an increase in temperature influences material properties of dissipators.

102.11.3 Two Node, 3D, Rubber Isolator Finite Element

Kelly (1991a,b)

Behavior of rubber (Ogden, 1984; Simo and Miehe, 1992; Simo and Pister, 1984)

102.11.4 Two Node, 3D, Frictional Pendulum Finite Element

102.12 Fully Coupled, Porous Solid – Pore Fluid Finite Elements

102.12.1 u-p-U Formulation

102.12.1.1 Background

This section follows developments by Zienkiewicz and Shiomi (1984).

The relationship between effective stress, total stress and pore pressure is given as:

$$\sigma''_{ij} = \sigma_{ij} - \alpha\delta_{ij}p \tag{102.83}$$

where σ''_{ij} is effective stress tensor, σ_{ij} is total stress tensor, δ_{ij} is Kronecker delta. $\delta_{ij} = 1$, when i=j, and $\delta_{ij} = 0$, when $i \neq j$. It is assumed that tensile components of effective and total stress are positive, and the pore fluid pressure p is also positive in tension, hence for compressions (usual case) pore fluid pressure is negative ($p < 0$) (Zienkiewicz et al., 1999a). For isotropic materials, $\alpha = 1 - K_T/K_S$. K_T is the total bulk modulus of the solid matrix, K_S is the bulk modulus of the solid particle. For most of the soil mechanics problems, as the bulk modulus K_S of the solid particles is much larger than that of the whole material, $\alpha \approx 1$ can be assumed. Equation (102.83) becomes

$$\sigma''_{ij} = \sigma_{ij} - \delta_{ij}p \tag{102.84}$$

In the next sections, a detailed derivation of formulation and numerical implementation for a fully coupled (pore fluid and porous soil) solid mechanics problem is given. Derivations are based in part on earlier work by Zienkiewicz et al. (1999a).

102.12.1.2 Governing Equations of Porous Media

The following notation is used:

- σ_{ij}, the total Cauchy stress in the mixture,
- u_i, the displacement of the solid skeleton,
- w_i, the displacement of the fluid phase relative to the skeleton of solid,
- p, the pore water pressure,
- $\varepsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i})$, the strain increment of the solid phase,
- $\omega_{ij} = \frac{1}{2}(u_{i,j} - u_{j,i})$, the rotation increment of the solid phase,
- ρ, ρ_s, ρ_f, the densities of the mixture, solid phase and water respectively,
- n the porosity,
- $\theta = -\dot{w}_{i,i}$, the rate of change of volume of water per unit total volume of mixture.

The Equilibrium Equation of the Mixture. The overall equilibrium or momentum balance equation for the soil-fluid mixture is written as

$$\sigma_{ij,j} - \rho \ddot{u}_i - \rho_f [\ddot{w}_i + \underline{\dot{w}_j \dot{w}_{i,j}}] + \rho b_i = 0 \tag{102.85}$$

here $\ddot{u}_i$ is the acceleration of the solid part, b_i is the body force per unit mass, $\ddot{w}_i + \underline{\dot{w}_j \dot{w}_{i,j}}$ is the fluid acceleration relative to the solid part, $\ddot{w}_i$ is local acceleration, $\underline{\dot{w}_j \dot{w}_{i,j}}$ is convective acceleration.

The underlined terms in the above equation represent the fluid acceleration relative to the solid and convective terms of this acceleration. Generally, this acceleration is so small that it is frequently omitted. For static problems, equation 102.85 only consists of the first and last terms.

For fully saturated porous media (no air inside), from definition

$$
\begin{aligned}
\rho &= \frac{M_t}{V_t} \\
&= \frac{M_s + M_f}{V_t} \\
&= \frac{V_s\rho_s + V_f\rho_f}{V_t} \\
&= \frac{V_f}{V_t}\rho_f + \frac{V_t - V_f}{V_t}\rho_s \\
&= n\rho_f + (1-n)\rho_s \\
\rho &= n\rho_f + (1-n)\rho_s \qquad (102.86)
\end{aligned}
$$

where M_t, M_s and M_f are the mass of total, solid part and fluid part respectively. V_t, V_s and V_f are the volume of total, solid part and fluid part respectively.

The Equilibrium Equation of the Fluid. For the pore fluid, the equation of momentum balance is written as

$$
-p_{,i} - R_i - \rho_f\ddot{u}_i - \rho_f[\ddot{w}_i + \underline{\dot{w}_j\dot{w}_{i,j}}]/n + \rho_f b_i = 0 \qquad (102.87)
$$

where R is the viscous drag forces. It is noted that the underlined terms in equation 102.87 represent the convective fluid acceleration again and are generally small. Also note that the permeability $\mathbf{k}$ is used with dimensions of $[length]^3[time]/[mass]$, which is different from the usual soil mechanics convention, where the permeability has the dimension of velocity, i.e., $[length]/[time]$. Their values are related by $k = K/\rho_f g$, where g is the gravitational acceleration at which the permeability is measured. Assuming the Darcy seepage law: $n\dot{w} = Ki$, here i is the head gradient. Seepage force is then $R = \rho_f gi$. R can be written as

$$
R_i = k_{ij}^{-1}\dot{w}_j \qquad \text{or} \qquad R_i = k^{-1}\dot{w}_i \qquad (102.88)
$$

where k_{ij} or k are Darcy permeability coefficients for anisotropic and isotropic conditions respectively.

Flow Conservation Equation. The final equation is supplied by the mass conservation of the fluid flow

$$
\dot{w}_{i,i} + \alpha\dot{\varepsilon}_{ii} + \frac{\dot{p}}{Q} + \underline{n\frac{\dot{\rho}_f}{\rho_f}} + \dot{s}_0 = 0 \qquad (102.89)
$$

The first term of equation(102.89) is the flow divergence of a unit volume of mixture. The second term is the volume change of the mixture. In the third term, Q is relative to the compressibility of the solid and fluid. The underlined terms represent change of density and rate of volume expansion of the solid in case of thermal changes. They are generally negligible.

$$\frac{1}{Q} \equiv \frac{n}{K_f} + \frac{\alpha - n}{K_s} \cong \frac{n}{K_f} + \frac{1-n}{K_s} \tag{102.90}$$

where K_s and K_f are the bulk moduli of the solid and fluid phases respectively. Note that the bulk modulus of the solid phase K_s is the actual bulk modulus of the solid particle.

Obtained are the total mixture equilibrium equation (102.85), fluid equilibrium equation (102.87) and the flow conservation equation (102.89) for saturated soil. By omitting the convective acceleration (the underline terms in (102.85) and (102.87)), density variation and the volume expansion due to the thermal change (the underline terms in (102.89)), the equations of coupled system can be further simplified, as summarized below

$$\sigma_{ij,j} - \rho \ddot{u}_i - \rho_f \ddot{w}_i + \rho b_i = 0 \tag{102.91}$$

$$-p_{,i} - R_i - \rho_f \ddot{u}_i - \frac{\rho_f \ddot{w}_i}{n} + \rho_f b_i = 0 \tag{102.92}$$

$$\dot{w}_{i,i} + \alpha \dot{\varepsilon}_{ii} + \frac{\dot{p}}{Q} = 0 \tag{102.93}$$

Bulk Modulus of Fluid (see Verruijt (2012) page 97... compressibility of water and with air bubbles)...

102.12.1.3 Modified Governing Equations.

Solid Part Equilibrium Equation. A new variable U_i is introduced in place of the relative pseudo-displacement w_i

$$U_i = u_i + U_i^R = u_i + \frac{w_i}{n} \tag{102.94}$$

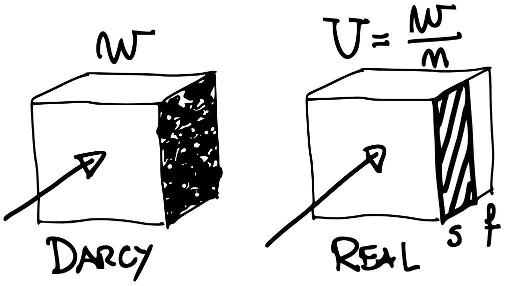


Figure 102.10: Fluid mechanics of Darcy's flow (w_i) versus real flow ($U_i = w_i/n$).

Change of variables is finalized by insertion of equation 102.94 into equations 102.91 and 102.92, and then by subtraction of term ($[n \times \mathit{equation}$ 102.92]) from equation 102.91, which leads to the equation of skeleton equilibrium

$$\sigma_{ij,j} - \rho\ddot{u}_i + \rho b_i + np_{,i} + nR_i + n\rho_f\ddot{u}_i - n\rho_f b_i = 0 \tag{102.95}$$

By substituting $\rho = (1-n)\rho_s + n\rho_f$

$$\sigma_{ij,j} - (1-n)\rho_s\ddot{u}_i - n\rho_f\ddot{u}_i + (1-n)\rho_s b_i + n\rho_f b_i + np_{,i} + nR_i + n\rho_f\ddot{u}_i - n\rho_f b_i = 0$$

$$\sigma_{ij,j} + np_{,i} + (1-n)\rho_s b_i - (1-n)\rho_s\ddot{u}_i + nR_i = 0 \tag{102.96}$$

By using the definition of effective stress, equation 102.83, equation 102.95 becomes

$$\sigma''_{ij,j} - (\alpha - n)p_{,i} + (1-n)\rho_s b_i - (1-n)\rho_s\ddot{u}_i + nR_i = 0 \tag{102.97}$$

Fluid Part Equilibrium Equation. The fluid part equilibrium equation can be obtained simply by $[n \times$ (102.92],i.e.

$$-np_{,i} - nR_i - n\rho_f\ddot{u}_i - \rho_f\ddot{w}_i + n\rho_f b_i = 0$$

$$-np_{,i} - nR_i - n\rho_f(\ddot{u}_i + \frac{\ddot{w}_i}{n}) + n\rho_f b_i = 0 \tag{102.98}$$

From equation (102.94),

$$\ddot{U}_i = \ddot{u}_i + \frac{\ddot{w}_i}{n} \tag{102.99}$$

so that equation (102.98) becomes:

$$-np_{,i} + n\rho_f b_i - n\rho_f\ddot{U}_i - nR_i = 0 \tag{102.100}$$

Mixture Balance of Mass. By differentiating equation (102.94) in time and space

$$\dot{w}_{i,i} = n\dot{U}_{i,i} - n\dot{u}_{i,i} \tag{102.101}$$

Notice that $\dot{\varepsilon}_{ii} = \dot{u}_{i,i}$, so that equation (102.101) becomes

$$\dot{w}_{i,i} = n\dot{U}_{i,i} - n\dot{\varepsilon}_{ii} \tag{102.102}$$

By substituting (102.102) to (102.93)

$$n\dot{U}_{i,i} - n\dot{\varepsilon}_{ii} + \alpha\dot{\varepsilon}_{ii} + \frac{\dot{p}}{Q} = 0 \tag{102.103}$$

or:

$$-n\dot{U}_{i,i} = (\alpha - n)\dot{\varepsilon}_{ii} + \frac{1}{Q}\dot{p} \tag{102.104}$$

Developed is a set of modified governing equations (102.97), (102.100) and (102.104). They are summarized below

$$\sigma^{''}_{ij,j} - (\alpha - n)p_{,i} + (1-n)\rho_s b_i - (1-n)\rho_s \ddot{u}_i + nR_i = 0 \tag{102.105}$$

$$-np_{,i} + n\rho_f b_i - n\rho_f \ddot{U}_i - nR_i = 0 \tag{102.106}$$

$$-n\dot{U}_{i,i} = (\alpha - n)\dot{\varepsilon}_{ii} + \frac{1}{Q}\dot{p} \tag{102.107}$$

From the modified equation set (102.105), (102.106) and (102.107), it is noted that only $\ddot{u}_i$ occurs in the first equation, and only $\ddot{U}_i$ in the second, thus leading to a convenient diagonal form in discretization.

Obtained is a a complete equation system given by (102.105), (102.106) and (102.107). With the basic definitions introduced earlier, there are three essential unknowns:

1. three solid displacement u_i
2. pore pressure p
3. three fluid displacement U_i

The boundary conditions imposed on these variables will complete the problem. These boundary conditions are:

- For the momentum balance part,
  - on boundary Γ_t, traction $t_i(t)$(or $\sigma_{ij}n_j$), where n_i is the i-th component of the normal to the boundary.
  - On boundary Γ_u, the displacement u_i is given.
- For the fluid part, again the boundary is divided into two parts:
  - On Γ_p, the pressure p is specified,
  - on Γ_w, the normal outflow $\dot{w}_n$ is specified. For impermeable boundary a zero value for the outflow should be specified.

The boundary conditions can be summarized below

$$\begin{aligned}
\Gamma &= \Gamma_t \cup \Gamma_u \\
t_i &= \sigma_{ij} n_j = \bar{t}_i \qquad \text{on} \qquad \Gamma = \Gamma_t \\
u_i &= \bar{u}_i \qquad \text{on} \qquad \Gamma = \Gamma_u
\end{aligned} \tag{102.108}$$

and

$$\begin{aligned}
\Gamma &= \Gamma_p \cup \Gamma_w \\
p &= \bar{p} \qquad \text{on} \qquad \Gamma = \Gamma_p \\
n^T w &= w_n \qquad \text{on} \qquad \Gamma = \Gamma_w
\end{aligned} \tag{102.109}$$

102.12.1.4 Numerical Solution of the u-p-U Governing Equations

The solutions to the problems governed by the modified governing equation set (102.105), (102.106) and (102.107) can be found by solving partial differential equations, which can be written as

$$A\ddot{\Phi} + B\dot{\Phi} + L(\Phi) = 0 \tag{102.110}$$

where A, B are constant matrices, and L is an operator involving spatial differentials. The dot notation represents the time differentiation. Vector of dependent variables, Φ represents the displacement u or the pore fluid pressure p.

The finite element solution of a problem proceeds as follows.

1. Discretize or approximate the unknown functions Φ by a finite set of parameters $\overline{\Phi}_k$ and shape function H_k. They are specified in space dimensions. Thus

$$\Phi \cong \Phi^h = \sum_{k=1}^{n} H_k \overline{\Phi}_k \tag{102.111}$$

2. Insert the value of the approximating function $\hat{\Phi}$ into the differential equations to obtain a residual, then a set of weighted residual equations can be written in the form

$$\int^{\Omega} W_j^T (A\ddot{\Phi}^h + B\dot{\Phi}^h + L(\Phi^h)) d\Omega = 0 \tag{102.112}$$

In the finite element method, the weighting functions W_j are usually identical to the shape functions.

The solid displacement u_i, the pore pressure p, and the absolute fluid displacement U_i can be approximated using shape functions and nodal values.

$$\begin{aligned} u_i &= H^u_K \bar{u}_{Ki} \\ p &= H^p_K \bar{p}_K \\ U_i &= H^U_K \bar{U}_{Ki} \end{aligned} \tag{102.113}$$

where H^u_K, H^p_K, and H^U_K are shape functions for solid displacement, pore pressure and fluid displacement, respectively, $\bar{u}_{Ki}$,$\bar{p}_K$,$\bar{U}_{Ki}$ are nodal values of solid displacement, pore pressure and fluid displacement, respectively.

Numerical Solution of solid part equilibrium equation. To obtain the numerical solution of the first equation, premultiply equation 102.105 by H^u_K and integrate over the domain.

First term of (102.105) becomes

$$\begin{aligned} \int_\Omega H^u_K \sigma''_{ij,j} d\Omega &= \int_{\Gamma_t} H^u_K n_j \sigma''_{ij} d\Gamma - \int_\Omega H^u_{K,j} \sigma''_{ij} d\Omega \\ &= \int_{\Gamma_t} H^u_K (\bar{t}_i + n_i \alpha p) d\Gamma - \int_\Omega H^u_{K,j} \sigma''_{ij} d\Omega \\ &= (f^u_1)_{Ki} - \int_\Omega H^u_{K,j} D_{ijml} \varepsilon_{ml} d\Omega \\ &= (f^u_1)_{Ki} - [\int_\Omega H^u_{K,j} D_{ijml} H^u_{P,l} d\Omega] \bar{u}_{Pm} \\ &= (f^u_1)_{Ki} - K^{EP}_{KimP} \bar{u}_{Pm} \\ &= (f^u_1)_{Ki} - K^{EP}_{KijL} \bar{u}_{Lj} \\ &= (\boldsymbol{f^u_1}) - \boldsymbol{K^{EP}} \bar{\boldsymbol{u}} \end{aligned} \tag{102.114}$$

where K^{EP} is the stiffness matrix of the solid part,n_i is the direction of the normal on the boundary.

Second term of (102.105) becomes

$$\begin{aligned} -\int_\Omega H^u_K (\alpha - n) p_{,i} d\Omega &= -\int_{\Gamma_p} H^u_K (\alpha - n) n_i p d\Gamma + \int_\Omega H^u_{K,i} (\alpha - n) p d\Omega \\ &= -\int_{\Gamma_p} H^u_K (\alpha - n) n_i p d\Gamma + [\int_\Omega H^u_{K,i} (\alpha - n) H^p_M d\Omega] \bar{p}_M \\ &= -(f^u_4)_{Ki} + (G_1)_{KiM} \bar{p}_M \\ &= -\boldsymbol{f^u_4} + (\boldsymbol{G_1}) \bar{\boldsymbol{p}} \end{aligned} \tag{102.115}$$

Third term of (102.105) (solid body force) is then

$$\int_\Omega H_K^u (1-n)\rho_s b_i d\Omega = (f_5^u)_{Ki} \tag{102.116}$$

Fourth term of (102.105) can be written as

$$\begin{aligned}
-\int_\Omega H_K^u (1-n)\rho_s \delta_{ij}\ddot{u}_j d\Omega &= -[\int_\Omega H_K^u (1-n)\rho_s \delta_{ij} H_L^u d\Omega]\ddot{\bar{u}}_{Lj} \\
&= -(M_s)_{KijL}\ddot{\bar{u}}_{Lj} \\
&= \boldsymbol{-M_s\ddot{\bar{u}}}
\end{aligned} \tag{102.117}$$

where M_s is the mass matrix of solid part. By substituting equations (102.88) and (102.94), last term of (102.105) (Damping Matrix) becomes

$$\begin{aligned}
\int_\Omega H_K^u n R_i d\Omega &= \int_\Omega H_K^u n k_{ij}^{-1}\dot{w}_j d\Omega \\
&= \int_\Omega H_K^u n^2 k_{ij}^{-1}\dot{U}_j d\Omega - \int_\Omega H_K^u n^2 k_{ij}^{-1}\dot{u}_j d\Omega \\
&= [\int_\Omega H_K^u n^2 k_{ij}^{-1} H_L^U d\Omega]\dot{\bar{U}}_{Lj} - [\int_\Omega H_K^u n^2 k_{ij}^{-1} H_L^U d\Omega]\dot{\bar{u}}_{Lj} \\
&= (C_2)_{KijL}\dot{\bar{U}}_{Lj} - (C_1)_{KijL}\dot{\bar{u}}_{Lj} \\
&= \boldsymbol{C_2\dot{\bar{U}} - C_1\dot{\bar{u}}}
\end{aligned} \tag{102.118}$$

Equation (102.105) becomes

$$\boldsymbol{-K^{EP}\bar{u} + f_1^u - f_4^u + G_1\bar{p} + f_5^u + M_s\ddot{\bar{u}} + C_2\dot{\bar{U}} - C_1\dot{\bar{u}}} = 0 \tag{102.119}$$

or

$$\boldsymbol{K^{EP}\bar{u} - G_1\bar{p} - C_2\dot{\bar{U}} + C_1\dot{\bar{u}} + M_s\ddot{\bar{u}} = \bar{f}_s} \tag{102.120}$$

where

$$\bar{f}_s = f_1^u - f_4^u + f_5^u \tag{102.121}$$

and in index form

$$K_{KijL}^{EP} - (G_1)_{KiL}\bar{p}_L + (C_2)_{KijL}\dot{\bar{U}}_{Lj} - (C_1)_{KijL}\dot{\bar{u}}_{Lj} + (M_s)_{KijL}\ddot{\bar{u}}_{Ki} = (\bar{f}_s)_{Ki} \tag{102.122}$$

where

$$\begin{aligned}
\boldsymbol{K^{EP}} = (K^{EP})_{KimP} &= \int_{\Omega} H^u_{K,j} D_{ijml} H^u_{P,l} d\Omega \\
\boldsymbol{G_1} = (G_1)_{KiM} &= \int_{\Omega} H^u_{K,i} (\alpha - n) H^p_M d\Omega \\
\boldsymbol{C_2} = (C_2)_{KijL} &= \int_{\Omega} H^u_K n^2 k^{-1}_{ij} H^U_L d\Omega \\
\boldsymbol{C_1} = (C_1)_{KijL} &= \int_{\Omega} H^u_K n^2 k^{-1}_{ij} H^u_L d\Omega \\
\boldsymbol{M_s} = (M_s)_{KijL} &= \int_{\Omega} H^u_K (1-n) \rho_s \delta_{ij} H^u_L d\Omega \\
\boldsymbol{\bar{f}} = (\bar{f}_s)_{Ki} &= (f^u_1)_{Ki} - (f^u_4)_{Ki} + (f^u_5)_{Ki}
\end{aligned} \tag{102.123}$$

Numerical Solution of fluid part equilibrium equation. From equations (102.88) and (102.94),

$$R_i = n k^{-1}_{ij} (\dot{U}_j - \dot{u}_j) \tag{102.124}$$

By substituting (102.124) into equation (102.106),

$$-n p_{,i} + n \rho_f b_i - n \rho_f \ddot{U}_i - n^2 k^{-1}_{ij} (\dot{U}_j - \dot{u}_j) = 0 \tag{102.125}$$

By premultiplying (102.125) by H^U_K and integrating over the domain, first term of (102.125) becomes

$$\begin{aligned}
-\int_{\Omega} n H^U_K p_{,i} d\Omega &= -\int_{\Gamma_p} n H^U_K n_i p d\Gamma + \int_{\Omega} n H^U_{K,i} p d\Omega \\
&= -(f_1)_{Ki} + [\int_{\Omega} n H^U_{K,i} H^p_M d\Omega] \bar{p}_M \\
&= -(f_1)_{Ki} + (G_2)_{KiM} \bar{p}_M \\
&= \boldsymbol{-(f_1)_{Ki} + (G_2)\bar{p}}
\end{aligned} \tag{102.126}$$

Second term of (102.125) is then

$$\int_{\Omega} H^U_K \rho_f b_i d\Omega = (f_2)_{Ki} \tag{102.127}$$

Third term of (102.125) (Lumped mass matrix obtained by multiplying δ_{ij}) becomes

$$\begin{aligned}
-\int_{\Omega} H^U_K n \rho_f \delta_{ij} \ddot{U}_j d\Omega &= -[\int_{\Omega} H^U_K n \rho_f \delta_{ij} H^U_L d\Omega] \ddot{\bar{U}}_{Lj} \\
&= -(M_f)_{KijL} \ddot{\bar{U}}_{Lj} \\
&= \boldsymbol{-M_f \ddot{\bar{U}}}
\end{aligned} \tag{102.128}$$

Forth term of (102.125) becomes

$$\begin{aligned}
-\int_\Omega H^U_K n^2 k^{-1}_{ij} \dot{U}_j d\Omega + \int_\Omega H^U_K n^2 k^{-1}_{ij} \dot{u}_j d\Omega &= -[\int_\Omega H^U_K n^2 k^{-1}_{ij} H^U_L d\Omega]\dot{\overline{U}}_{Lj} \qquad (102.129)\\
+[\int_\Omega H^U_K n^2 k^{-1}_{ij} H^u_L d\Omega]\dot{\overline{u}}_{Lj} & \\
&= -(C_3)_{KijL}\dot{\overline{U}}_{Lj} + (C_2)^T_{LjiK}\dot{\overline{u}}_{Lj} \\
&= \boldsymbol{C_3}\dot{\overline{\boldsymbol{U}}} + \boldsymbol{C_2^T}\dot{\overline{\boldsymbol{u}}} \qquad (102.130)
\end{aligned}$$

Equation (102.125) becomes

$$-\boldsymbol{f_1} + \boldsymbol{G_2}\overline{\boldsymbol{p}} + \boldsymbol{f_2} - \boldsymbol{M_f}\ddot{\overline{\boldsymbol{U}}} - \boldsymbol{C_3}\dot{\overline{\boldsymbol{U}}} + \boldsymbol{C_2^T}\dot{\overline{\boldsymbol{u}}} = \boldsymbol{0} \qquad (102.131)$$

or

$$-\boldsymbol{G_2}\overline{\boldsymbol{p}} - \boldsymbol{C_2^T}\dot{\overline{\boldsymbol{u}}} + \boldsymbol{C_3}\dot{\overline{\boldsymbol{U}}} + \boldsymbol{M_f}\ddot{\overline{\boldsymbol{U}}} = \overline{\boldsymbol{f}}_{\boldsymbol{f}} \qquad (102.132)$$

where

$$\overline{f}_f = f_2 - f_1 \qquad (102.133)$$

and in index form

$$-(G_2)_{KiM}\overline{p}_M - (C_2)^T_{LjiK}\dot{\overline{u}}_{Lj} + (C_3)_{KijL}\dot{\overline{U}}_{Lj} + (M_f)_{KijL}\ddot{U}_{Lj} = (f_f)_{Ki} \qquad (102.134)$$

where

$$\begin{aligned}
\boldsymbol{(f_f)_{Ki}} &= (f_1)_{Ki} - (f_2)_{Ki} \\
\boldsymbol{G_2} &= (G_2)_{KiN} = \int_\Omega n H^U_{K,i} H^p_M d\Omega \\
\boldsymbol{C_2^T} &= (C_2^T)_{KijL} = \int_\Omega H^U_K n^2 k^{-1}_{ij} H^u_L d\Omega \\
\boldsymbol{C_3} &= (C_3)_{KijL} = \int_\Omega H^U_K n^2 k^{-1}_{ij} H^U_L d\Omega \\
\boldsymbol{M_f} &= (M_f)_{KijL} = \int_\Omega H^U_K n \rho_f \delta_{ij} H^U_L d\Omega \qquad (102.135)
\end{aligned}$$

Numerical Solution of flow conservation equation. By integrating (102.107) in time and noticing that $\varepsilon_{ii} = u_{i,i}$,

$$-nU_{i,i} = (\alpha - n)\varepsilon_{ii} + \frac{1}{Q}p \qquad (102.136)$$

By multiplying (102.136) by H^p_M and integrating over domain, first term of (102.136) becomes

$$-[\int_\Omega H^p_M n H^U_{L,j} d\Omega]\overline{U}_{Lj} = -(G_2)_{MLj}\overline{U}_{Li} = -\boldsymbol{G_2^T}\overline{\boldsymbol{U}} \qquad (102.137)$$

Second term of (102.136) is

$$\begin{aligned}\int_{\Omega} H^p_M(\alpha - n)u_{i,i}d\Omega &= [\int_{\Omega} H^p_M(\alpha - n)H^u_{L,j}d\Omega]\bar{u}_{Lj} \\ &= (G_1)_{LjM}\bar{u}_{Lj} \\ &= \boldsymbol{G_1^T}\bar{\boldsymbol{u}} \end{aligned} \tag{102.138}$$

Third term of (102.136) becomes

$$[\int_{\Omega} H^p_N \frac{1}{Q} H^p_M d\Omega]p_N = P_{NM}p_M = \boldsymbol{P}\bar{\boldsymbol{p}} \tag{102.139}$$

The equation (102.136) becomes

$$\boldsymbol{G_2^T}\bar{\boldsymbol{U}} + \boldsymbol{G_1^T}\bar{\boldsymbol{u}} + \boldsymbol{P}\bar{\boldsymbol{p}} = \boldsymbol{0} \tag{102.140}$$

in index form

$$(G_2)_{LiK}\bar{U}_{Li} + (G_1)_{LiK}\bar{u}_{Li} + P_{KL}\bar{p}_L = 0 \tag{102.141}$$

102.12.1.5 Matrix form of the governing equations.

The numerical forms of governing equations (102.120),(102.132) and (102.140) can be written together in the matrix form as

$$\begin{bmatrix} M_s & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & M_f \end{bmatrix} \begin{bmatrix} \ddot{\bar{u}} \\ \ddot{\bar{p}} \\ \ddot{\bar{U}} \end{bmatrix} + \begin{bmatrix} C_1 & 0 & -C_2 \\ 0 & 0 & 0 \\ -C_2^T & 0 & C_3 \end{bmatrix} \begin{bmatrix} \dot{\bar{u}} \\ \dot{\bar{p}} \\ \dot{\bar{U}} \end{bmatrix} + \begin{bmatrix} K^{EP} & -G_1 & 0 \\ -G_1^T & -P & -G_2^T \\ 0 & -G_2 & 0 \end{bmatrix} \begin{bmatrix} \bar{u} \\ \bar{p} \\ \bar{U} \end{bmatrix} = \begin{bmatrix} \bar{f}_s \\ 0 \\ \bar{f}_f \end{bmatrix} \tag{102.142}$$

or in index form

$$\begin{bmatrix} (M_s)_{KijL} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & (M_f)_{KijL} \end{bmatrix} \begin{bmatrix} \ddot{\bar{u}}_{Lj} \\ \ddot{\bar{p}}_N \\ \ddot{\bar{U}}_{Lj} \end{bmatrix} + \begin{bmatrix} (C_1)_{KijL} & 0 & -(C_2)_{KijL} \\ 0 & 0 & 0 \\ -(C_2)_{LjiK} & 0 & (C_3)_{KijL} \end{bmatrix} \begin{bmatrix} \dot{\bar{u}}_{Lj} \\ \dot{\bar{p}}_N \\ \dot{\bar{U}}_{Lj} \end{bmatrix}$$
$$+ \begin{bmatrix} (K^{EP})_{KijL} & -(G_1)_{KiM} & 0 \\ -(G_1)_{LjM} & -P_{MN} & -(G_2)_{LjM} \\ 0 & -(G_2)_{KiL} & 0 \end{bmatrix} \begin{bmatrix} \bar{u}_{Lj} \\ \bar{p}_M \\ \bar{U}_{Lj} \end{bmatrix} = \begin{bmatrix} \bar{f}^{solid}_{Ki} \\ 0 \\ \bar{f}^{fluid}_{Ki} \end{bmatrix} \tag{102.143}$$

where

$$
\begin{aligned}
\boldsymbol{M_s} &= (M_s)_{KijL} = \int_\Omega H_K^u (1-n)\rho_s \delta_{ij} H_L^u d\Omega \\
\boldsymbol{M_f} &= (M_f)_{KijL} = \int_\Omega H_K^U n\rho_f \delta_{ij} H_L^U d\Omega \\
\boldsymbol{C_1} &= (C_1)_{KijL} = \int_\Omega H_K^u n^2 k_{ij}^{-1} H_L^u d\Omega \\
\boldsymbol{C_2} &= (C_2)_{KijL} = \int_\Omega H_K^u n^2 k_{ij}^{-1} H_L^U d\Omega \\
\boldsymbol{C_3} &= (C_3)_{KijL} = \int_\Omega H_K^U n^2 k_{ij}^{-1} H_L^U d\Omega \\
\boldsymbol{K^{EP}} &= (K^{EP})_{KijL} = \int_\Omega H_{K,m}^u D_{imjn} H_{L,n}^u d\Omega \\
\boldsymbol{G_1} &= (G_1)_{KiM} = \int_\Omega H_{K,i}^u (\alpha - n) H_M^p d\Omega \\
\boldsymbol{G_2} &= (G_2)_{KiM} = \int_\Omega n H_{K,i}^U H_M^p d\Omega \\
\boldsymbol{P} &= P_{NM} = \int_\Omega H_N^p \frac{1}{Q} H_M^p d\Omega
\end{aligned}
\tag{102.144}
$$

$$
\begin{aligned}
\bar{f}_{Ki}^{solid} &= (f_1^u)_{Ki} - (f_4^u)_{Ki} + (f_5^u)_{Ki} \\
\bar{f}_{Ki}^{fluid} &= -(f_1^U)_{Ki} + (f_2^U)_{Ki} \\
(f_1^u)_{Ki} &= \int_{\Gamma_t} H_K^u n_j \sigma_{ij}^{''} d\Gamma \\
(f_4^u)_{Ki} &= \int_{\Gamma_p} H_K^u (\alpha - n) n_i p d\Gamma \\
(f_5^u)_{Ki} &= \int_\Omega H_K^u (1-n)\rho_s b_i d\Omega \\
(f_1^U)_{Ki} &= \int_{\Gamma_p} n H_K^U n_i p d\Gamma \\
(f_2^U)_{Ki} &= \int_\Omega n H_K^U \rho_f b_i d\Omega
\end{aligned}
\tag{102.145}
$$

Functions N^u,N^p,N^U are shape functions for unknown field of skeleton dispalacements, pore fluid pressures and fluid displacements, respectively, while ρ, ρ_s, ρ_f are the density of the total, solid and fluid phases, respectively, n is the porosity, and by its definition $\rho = (1-n)\rho_s + n\rho_f$, the symbol n_i is the direction of the normal on the boundary, u_i is the displacement of the solid part, p is pore fluid pressure and U_i is the absolute displacement of the fluid part. Equation (102.142) represents the general form $(u-p-U)$ for coupled system which can be written in a familiar form as

$$
\boldsymbol{M\ddot{x} + C\dot{x} + Kx = f}
\tag{102.146}
$$

where x represents the generalized unknown variable. The solution of this equation for each time step will render an unknown field for given initial and boundary conditions.

102.12.1.6 Choice of shape functions

Isoparametric elements are used in previous sections, where the coordinates are interpolated using the same shape functions as for the unknown. This mapping allows using elements of more arbitrary shape than simple forms such as rectangles and triangles. But in static or dynamic undrained analysis the permeability (and compressibility) matrices are zero, i.e.($Q \rightarrow \infty$,and $P \rightarrow 0$),resulting in a zero-matrix diagonal term in the equation(102.143).

The matrix to be solved is the same as that in the solutions of problems of incompressible elasticity or fluid mechanics. Actually, a wide choice of shape functions is available if the limiting(undrained) condition is never imposed. Due to the presence of first derivatives in space in all the equations, it is necessary to use "C_0-continuous" interpolation functions and the suitable element forms are shown in Fig.102.11.

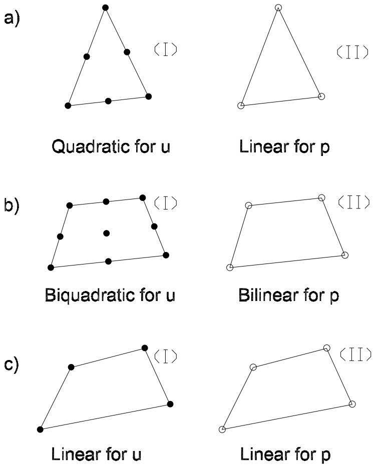


Figure 102.11: Shape functions used for coupled analysis, displacement u and pore pressure p formulation

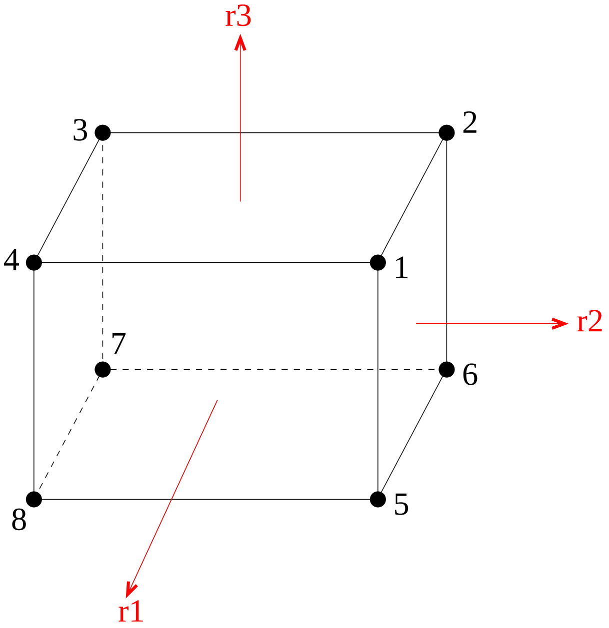


Figure 102.12: 8 node u-p-U brick element. Note that all seven DOFs (three porous solid displacements u_i, pore fluid pressure p and pore fluid displacements U_i are defined at each node.

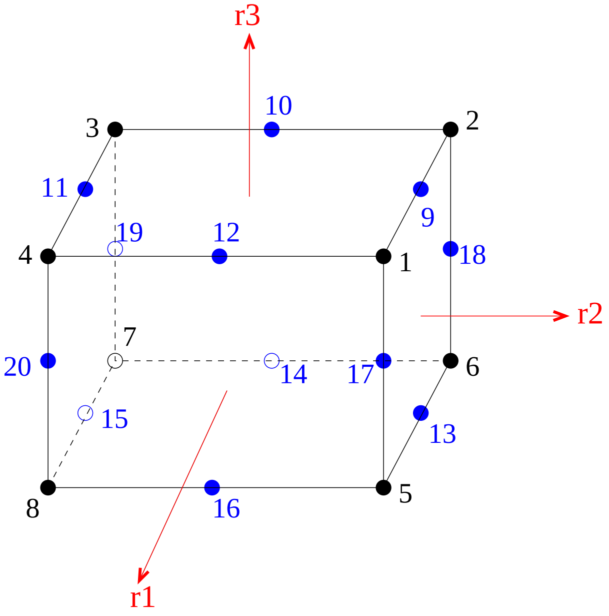


Figure 102.13: 20 node brick element. Note that all seven DOFs (three porous solid displacements u_i, pore fluid pressure p and pore fluid displacements U_i are defined at each node.

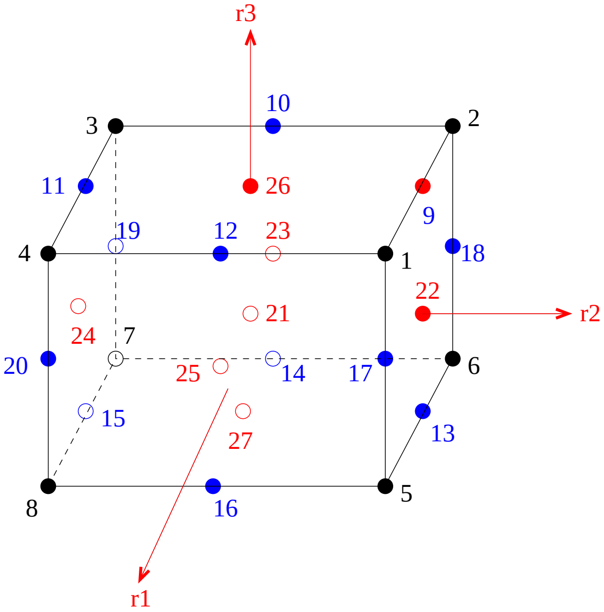


Figure 102.14: 27 node brick element. Note that all seven DOFs (three porous solid displacements u_i, pore fluid pressure p and pore fluid displacements U_i are defined at each node.

102.12.1.7 8 Node $u-p-U$ Brick

102.12.1.8 20 Node $u-p-U$ Brick

102.12.1.9 27 Node $u-p-U$ Brick

102.12.2 u-p-U Formulation for Partially Saturated, Unsaturated Material

Coming SOON, by end of Winter 2020

102.12.3 u-p Formulation

102.12.3.1 Governing Equations of Porous Media

The formulation given here is based on Zienkiewicz et al. (1999b).

The first governing equation of porous media is total momentum balance equation:

$$\sigma_{ij,j} - \rho \ddot{u}_i + \rho b_i = 0 \tag{102.147}$$

where $\sigma_{ij} = \sigma''_{ij} - \alpha p \delta_{ij}$ and $\rho = (1-n)\rho_s + n\rho_f$.

The second governing equation is the fluid mass balance equation:

$$\left(k_{ij}(-p_{,j} + \rho_f b_j)\right)_{,i} + \alpha \dot{u}_{i,i} + \frac{\dot{p}}{Q_{sf}} = 0 \tag{102.148}$$

where

$$k_{ij} = \frac{k'_{ij}}{g\rho_f} = \frac{k'_{ij}}{\gamma_f} \tag{102.149}$$

and k'_{ij} is the permeability in Darcy's law with the same unit as velocity.

$$Q_{sf} = \frac{K_s K_f}{K_s + K_f} \tag{102.150}$$

is the total compression modulus, K_s and K_f are solid and fluid compression modulus, respectively.

The boundary conditions are

$$\sigma_{ij} n_j = \bar{t}_i \quad \text{on} \quad \Gamma = \Gamma_t \tag{102.151}$$

$$u_i = \bar{u}_i \quad \text{on} \quad \Gamma = \Gamma_u \tag{102.152}$$

$$n_i w_i = n_i k_{ij}(-p_{,j} + \rho_f b_j) = \bar{w} = -\bar{q} \quad \text{on} \quad \Gamma = \Gamma_w \tag{102.153}$$

$$p = \bar{p} \quad \text{on} \quad \Gamma = \Gamma_p \tag{102.154}$$

where $\bar{w}$ is the outflow and $\bar{q}$ is the influx.

102.12.3.2 Numerical Solutions of the Governing Equations

The solid displacement u_i and the pore pressure p can be approximated using shape functions and nodal values:

$$u_i = N^u_K \bar{u}_{Ki} \tag{102.155}$$

$$p = N^p_L \bar{p}_L \tag{102.156}$$

Similar approximations are applied to $\dot{u}_i$, $\ddot{u}_i$, $\dot{p}$ and $\ddot{p}$.

Numerical solution of the total momentum balance The numerical solution of the total momentum balance is

$$\int_\Omega N^u_K \left(\sigma_{ij,j} - \rho \ddot{u}_i + \rho b_i \right) \mathrm{d}\Omega = 0 \tag{102.157}$$

First-term of (102.157) becomes

$$
\begin{aligned}
\int_{\Omega} N^u_K \sigma_{ij,j} \mathrm{d}\Omega &= \int_{\Gamma_t} N^u_K \sigma_{ij,j} n_j \mathrm{d}\Omega - \int_{\Omega} N^u_{K,j} \sigma_{ij} \mathrm{d}\Omega \\
&= \int_{\Gamma_t} N^u_K \bar{t}_i \mathrm{d}\Omega - \int_{\Omega} N^u_{K,j} (\sigma''_{ij} - \alpha p \delta_{ij}) \mathrm{d}\Omega \\
&= (f^u_1)_{Ki} - \int_{\Omega} N^u_{K,j} \sigma''_{ij} \mathrm{d}\Omega + \int_{\Omega} N^u_{K,i} \alpha p \mathrm{d}\Omega \\
&= (f^u_1)_{Ki} - \int_{\Omega} N^u_{K,j} D_{ijml} \varepsilon_{ml} \mathrm{d}\Omega + [\int_{\Omega} \alpha N^u_{K,i} N^p_N \mathrm{d}\Omega] \bar{p}_N \\
&= (f^u_1)_{Ki} - [\int_{\Omega} N^u_{K,j} D_{ijml} N^u_{P,m} \mathrm{d}\Omega] \bar{u}_{Pm} + [\int_{\Omega} \alpha N^u_{K,i} N^p_N \mathrm{d}\Omega] \bar{p}_N \\
&= (f^u_1)_{Ki} - (K^{ep}_{KimP}) \bar{u}_{Pm} + (Q_{KiN}) \bar{p}_N \\
&= \boldsymbol{f^u_1 - (K^{ep}) \bar{u} + Q \bar{p}}
\end{aligned}
\tag{102.158}
$$

Second term of (102.157) becomes

$$
\begin{aligned}
-\int_{\Omega} N^u_K \rho \ddot{u}_i \mathrm{d}\Omega &= -\int_{\Omega} N^u_K \rho N^u_L \mathrm{d}\Omega \ddot{u}_{Li} \\
&= -[\int_{\Omega} N^u_K \rho N^u_L \mathrm{d}\Omega] \ddot{\bar{u}}_{Li} \\
&= -[\delta_{ij} \int_{\Omega} N^u_K \rho N^u_L \mathrm{d}\Omega] \ddot{\bar{u}}_{Lj} \\
&= -(M_{KijL}) \ddot{\bar{u}}_{Lj} \\
&= \boldsymbol{-M \ddot{\bar{u}}}
\end{aligned}
\tag{102.159}
$$

The third term of (102.157) becomes

$$
\begin{aligned}
\int_{\Omega} N^u_K \rho b_i \mathrm{d}\Omega &= (f^u_2)_{Ki} \\
&= \boldsymbol{f^u_2}
\end{aligned}
\tag{102.160}
$$

The equation (102.157) thus becomes

$$
(M_{KijL}) \ddot{\bar{u}}_{Lj} - (Q_{KiN}) \bar{p}_N + (K^{ep}_{KijL}) \bar{u}_{Lj} = (f^u_1)_{Ki} + (f^u_2)_{Ki} = (f^u)_{Ki}
\tag{102.161}
$$

or

$$
\boldsymbol{M \ddot{\bar{u}} - Q \bar{p} + (K^{ep}) \bar{u} = f^u_1 + f^u_2 = f^u}
\tag{102.162}
$$

Numerical solution of the fluid mass balance The numerical solution of the fluid mass balance is

$$
\int_{\Omega} N^p_M \left(k_{ij} (-p_{,j} + \rho_f b_j)_{,j} + \alpha \dot{u}_{i,i} + \frac{\dot{p}}{Q_{sf}} \right) \mathrm{d}\Omega = 0
\tag{102.163}
$$

First term of (102.163) becomes

$$\begin{aligned}
& \int_{\Omega} N_M^p \left(k_{ij}(-p_{,j} + \rho_f b_j)\right)_{,i} \mathrm{d}\Omega \qquad (102.164)\\
= & \int_{\gamma_w} N_M^p w_i n_i \mathrm{d}\Omega - \int_{\Omega} N_{M,i}^p k_{ij}(-p_{,j} + \rho_f b_j)\mathrm{d}\Omega \\
= & \int_{\gamma_w} N_M^p \bar{w}\mathrm{d}\Omega + \int_{\Omega} N_{M,i}^p k_{ij} p_{,j}\mathrm{d}\Omega - \int_{\Omega} N_{M,i}^p k_{ij}\rho_f b_j \mathrm{d}\Omega \\
= & (f_1^p)_M + \int_{\Omega} N_{M,i}^p k_{ij} p_{,j}\mathrm{d}\Omega - \int_{\Omega} N_{M,i}^p k_{ij}\rho_f b_j \mathrm{d}\Omega \\
= & (f_1^p)_M + [\int_{\Omega} N_{M,i}^p k_{ij} N_{N,j}^p \mathrm{d}\Omega]\bar{p}_N - (f_2^p)_M \\
= & (f_1^p)_M + (H_{MN})\bar{p}_N - (f_2^p)_M \\
= & \boldsymbol{f_1^p} + \boldsymbol{H}\bar{\boldsymbol{p}} - \boldsymbol{f_2^p} \qquad (102.165)
\end{aligned}$$

Second term of (102.163) becomes

$$\begin{aligned}
\int_{\Omega} N_M^p \alpha \dot{u}_{i,i}\mathrm{d}\Omega &= [\int_{\Omega} N_M^p \alpha N_{L,j}^u \mathrm{d}\Omega]\dot{\bar{u}}_{Lj} \\
&= (Q_{LjM})\dot{\bar{u}}_{Lj} \\
&= \boldsymbol{Q}^{\boldsymbol{T}}\dot{\bar{\boldsymbol{u}}} \qquad (102.166)
\end{aligned}$$

The third term of (102.163) becomes

$$\begin{aligned}
\int_{\Omega} N_M^p \frac{\dot{p}}{Q_{sf}}\mathrm{d}\Omega &= [\int_{\Omega} N_M^p \frac{1}{Q_{sf}} N_N^p \mathrm{d}\Omega]\dot{\bar{p}}_N \\
&= (S_{MN})\dot{\bar{p}}_N \\
&= \boldsymbol{S}\dot{\bar{\boldsymbol{p}}} \qquad (102.167)
\end{aligned}$$

The equation (102.163) thus becomes

$$(H_{MN})\bar{p}_N + (Q_{LjM})\dot{\bar{u}}_{Lj} + (S_{MN})\dot{\bar{p}}_N = -(f_1^p)_M + (f_2^p)_M = (f^p)_M \tag{102.168}$$

or

$$\boldsymbol{H}\bar{\boldsymbol{p}} + \boldsymbol{Q}^{\boldsymbol{T}}\dot{\bar{\boldsymbol{u}}} + \boldsymbol{S}\dot{\bar{\boldsymbol{p}}} = -\boldsymbol{f_1^p} + \boldsymbol{f_2^p} = \boldsymbol{f^p} \tag{102.169}$$

Matrix form of the governing equations Combine equation (102.161) and (102.168), to obtain

$$\begin{aligned}
&\begin{bmatrix} M_{KiLj} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{u}_{Lj} \\ \ddot{p}_N \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ Q_{LjM} & S_{MN} \end{bmatrix} \begin{bmatrix} \dot{u}_{Lj} \\ \dot{p}_N \end{bmatrix} \\
&\quad + \begin{bmatrix} (K^{ep})_{KiLj} & -Q_{KiN} \\ 0 & H_{MN} \end{bmatrix} \begin{bmatrix} u_{Lj} \\ p_N \end{bmatrix} = \begin{bmatrix} f_{Ki}^u \\ f_M^p \end{bmatrix}
\end{aligned} \tag{102.170}$$

or, by combining equations (102.162) and (102.169), obtain

$$\begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}} \\ \ddot{\boldsymbol{p}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{Q}^T & \boldsymbol{S} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{u}} \\ \dot{\boldsymbol{p}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}^{ep} & \boldsymbol{Q} \\ \boldsymbol{0} & \boldsymbol{H} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}^u \\ \boldsymbol{f}^p \end{bmatrix} \tag{102.171}$$

where

$$\boldsymbol{f}^u \quad \leftrightarrow \quad f^u_{Ki} = (f^u_1)_{Ki} + (f^u_2)_{Ki} \tag{102.172}$$

$$\boldsymbol{f}^p \quad \leftrightarrow \quad f^p_M = -(f^p_1)_M + (f^p_2)_M \tag{102.173}$$

and

$$\boldsymbol{f}^u_1 \quad \leftrightarrow \quad (f^u_1)_{Ki} = \int_{\Gamma_t} N^u_K \bar{t}_i \mathrm{d}\Gamma \tag{102.174}$$

$$\boldsymbol{f}^u_2 \quad \leftrightarrow \quad (f^u_2)_{Ki} = \int_{\Omega} N^u_K \rho b_i \mathrm{d}\Omega \tag{102.175}$$

$$\boldsymbol{f}^p_1 \quad \leftrightarrow \quad (f^p_1)_M = \int_{\Gamma_w} N^p_M \bar{w} \mathrm{d}\Gamma \tag{102.176}$$

$$\boldsymbol{f}^p_2 \quad \leftrightarrow \quad (f^p_2)_M = \int_{\Omega} N^p_{M,i} k_{ij} \rho_f b_j \mathrm{d}\Omega \tag{102.177}$$

$$\boldsymbol{M} \quad \leftrightarrow \quad M_{KiLj} = \delta_{ij} \int_{\Omega} N^u_K \rho N^u_L \mathrm{d}\Omega \tag{102.178}$$

$$\boldsymbol{Q} \quad \leftrightarrow \quad Q_{KiN} = \int_{\Omega} \alpha N^u_{K,i} N^p_N \mathrm{d}\Omega \tag{102.179}$$

$$\boldsymbol{S} \quad \leftrightarrow \quad S_{MN} = \int_{\Omega} N^p_M \frac{1}{Q_{sf}} N^p_N \mathrm{d}\Omega \tag{102.180}$$

$$\boldsymbol{H} \quad \leftrightarrow \quad H_{MN} = \int_{\Omega} N^p_{M,i} k_{ij} N^p_{N,j} \mathrm{d}\Omega \tag{102.181}$$

102.12.3.3 8 Node $u-p$ Brick

102.13 Material and Geometric Non–Linear Finite Element Formulation

102.13.1 Introduction

Presented here is a detailed formulation of material and geometric non–linear static finite element system of equations. The configuration of choice is material or Lagrangian. Eulerian and mixed Eulerian-Lagrangian configuration will be mentioned as need be.

102.13.2 Equilibrium Equations

The local form of equilibrium equations in material format (Lagrangian) for static case can be written as:

$$P_{iJ,J} - \rho_0 b_i = 0 \tag{102.182}$$

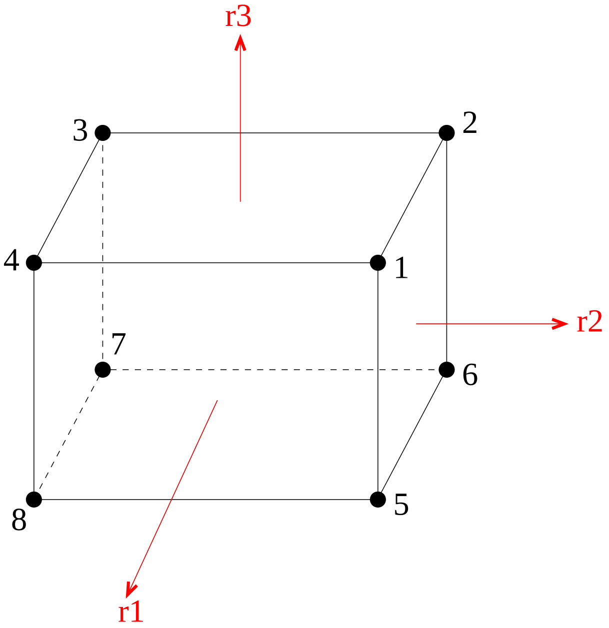


Figure 102.15: 8 node $u-p$ brick element. Note that all four DOFs (three porous solid displacements u_i and a pore fluid pressure p are defined at each node.

where $P_{iJ} = S_{IJ}(F_{iI})^t$ and S_{IJ} are first and second Piola–Kirchhoff stress tensors, respectively and b_I are body forces.

Weak form of equilibrium equations is obtained by premultiplying 102.182 with virtual displacements δu_i and integrating by parts on the initial configuration B_0 (initial volume V_0):

$$\int_{V_0} \delta u_{i,j} P_{ij} dV = \int_{V_0} \rho_0 \delta u_i b_i dV - \int_{S_0} \delta u_i \bar{t}_i dS \tag{102.183}$$

It proves beneficial to rewrite Lagrangian format of weak form of equilibrium equilibrium by using symmetric second Piola–Kirchhoff stress tensor S_{ij}:

$$\begin{aligned}
\int_{V_0} \delta u_{i,j} F_{jl} S_{il} dV &= \\
\int_{V_0} \frac{1}{2} \left(\delta u_{i,j} F_{jl} + F_{lj} \delta u_{j,i} + \right) S_{il} dV &= \\
\int_{V_0} \frac{1}{2} \left(\delta u_{i,j} \left(\delta_{jl} + u_{j,l} \right) + \left(\delta_{lj} + u_{l,j} \right) \delta u_{j,i} \right) S_{il} dV &= \\
\int_{V_0} \frac{1}{2} \left(\delta u_{i,l} + \delta ui, ju_{j,l} \right) + \left(\delta u_{l,i} + u_{l,j} \delta u_{j,i} \right) S_{il} dV &= \\
\int_{V_0} \frac{1}{2} \left(\left(\delta u_{i,l} + \delta u_{l,i} \right) + \left(\delta u_{i,j} u_{j,l} + u_{l,j} \delta u_{j,i} \right) \right) S_{il} dV &=
\end{aligned} \tag{102.184}$$

where symmetry of S_{il} was used, definition for deformation gradient $F_{ki} = \delta_{ki} + u_{k,i}$. In addition, conveniently defined was differential operator $\hat{E}_{il}(\delta u_i, u_i)$ as

$$\hat{E}_{il}(\delta u_i, u_i) = \frac{1}{2}\left(\delta u_{l,i} + \delta u_{i,l}\right) + \frac{1}{2}\left(u_{l,j}\delta u_{j,i} + \delta u_{i,j}u_{j,l}\right) \tag{102.185}$$

102.13.3 Formulation of Non–Linear Finite Element Equations

Consider the motion of a general solid in a fixed, non-moving Cartesian coordinate system, as shown in Figure (102.16), and assume that the solid can experience large displacements, large strains, and nonlinear constitutive response. The aim is to evaluate the equilibrium positions of the complete solid at discrete time points $0, \Delta t, 2\Delta t, \ldots,$ where Δt is an increment in time. To develop the solution strategy, assume that the solutions for the static and kinematic variables for all time steps from 0 to time t inclusive, have been obtained. The solution process for the next required equilibrium position corresponding to time $t + \Delta t$ is typical and would be applied repetitively until a complete solution path has been found. Hence, in the analysis one follows all particles of the solid in their motion, from the original to the final configuration of the solid. In so doing, a Lagrangian (or material) formulation of the problem was adopted.

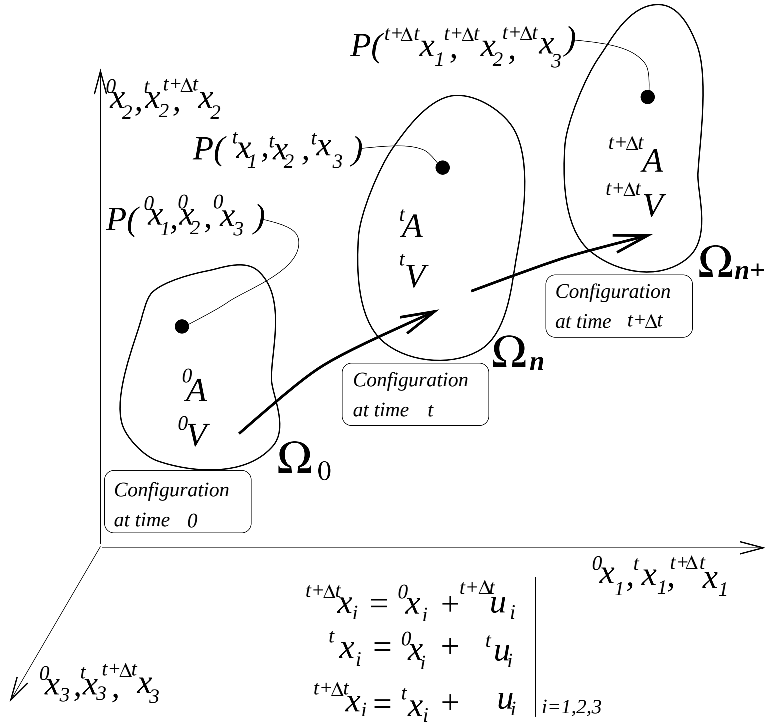


Figure 102.16: Motion of a solid in non-moving Cartesian coordinate system.

Weak format of the equilibrium equations can be obtained by premultiplying 102.182 with virtual displacements δu_i and integrating by parts. We obtain the virtual work equations in the Lagrangian format:

$$\int_{V_0} \delta u_{i,j} P_{ij} dV = \int_{V_0} \rho_0 \delta u_i b_i dV - \int_{S_0} \delta u_i \bar{t}_i dV \tag{102.186}$$

Virtual work equations can also be written in terms of second Piola–Kirchhoff stress tensor S_{IJ} as:

$$\int_{V_0} \delta u_{i,j} F_{jl} S_{il} dV = \int_{V_0} \rho_0 \delta u_i b_i dV + \int_{S_0} \delta u_i \bar{t}_i dV \tag{102.187}$$

which after some algebraic manipulations, and after observing that $S_{IJ} = S_{JI}$ yields (SEE ABOVE!) By introducing a differential operator $\hat{E}(u_1, u_2)$ as:

$$\hat{E}_{il}({}^1u_i, {}^2u_i) = \frac{1}{2}\,\left({}^1u_{i,l} + {}^1u_{l,i}\right) + \frac{1}{2}\,\left({}^1u_{l,j}\ {}^2u_{j,i} + {}^2u_{i,j}\ {}^1u_{j.l}\right) \tag{102.188}$$

virtual work equation 102.185 can be written as:

$$\int_{V_0} \hat{E}_{il}(\delta u_i, u_i) S_{il} dV \quad = \quad \int_{V_0} \rho_0 \delta u_i b_i dV + \int_{S_0} \delta u_i \bar{t}_i dV \tag{102.189}$$

or as:

$$W(\delta u_i, u_i^{(k)})^{int} + W^{ext}(\delta u_i) = 0 \tag{102.190}$$

with:

$$\begin{aligned} W^{int}(\delta u_i, {}^{n+1}_{\ \ 0}u_i^{(k)}) \quad &= \quad \int_{\Omega_c} \hat{E}_{ij}(\delta u_i, {}^{n+1}_{\ \ 0}u_i^{(k)})\ {}^{n+1}_{\ \ 0}S_{ij}^{(k)}\ dV \\ &= \quad \int_{\Omega_c} \left(\left(\delta u_{j,i} + \delta u_{i,j}\right) + \left(u_{j,r}\delta u_{r,i} + \delta u_{i,r} u_{r,j}\right)\right)\ S_{ij}^{(k)}\ dV \end{aligned} \tag{102.191}$$

$$W^{ext}(\delta u_i) \quad = \quad -\int_{\Omega_c} \rho_0\ \delta u_i\ {}^{n+1}_{\ \ 0}b_i\ dV - \int_{\partial\Omega_c} \delta u_i\ {}^{n+1}_{\ \ 0}t_i\ dS \tag{102.192}$$

102.13.4 Computational Domain in Incremental Analysis

This chapter elaborates on the choice of Total Lagrangian (TL) formulations as a computational domain. In addition, a Newton-type procedure is chosen for satisfying equilibrium, i.e., virtual work for a given computational domain. Given the displacement field $u_i^{(k)}(X_j)$, in iteration k, the iterative change δu_i

$$u_i^{(k+1)} = u_i^{(k)} + \Delta u_i \tag{102.193}$$

is obtained from the linearized virtual work expression

$$W(\delta u_i, u_i^{(k+1)}) \simeq W(\delta u_i, u_i^{(k)}) + \Delta W(\delta u_i, \Delta u_i; u_i^{(k)}) \tag{102.194}$$

Here, $W(\delta u_i, u_i^{(k)})$ is the virtual work expression

$$W(\delta u_i, u_i^{(k)}) = W(\delta u_i, u_i^{(k)})^{int} + W^{ext}(\delta u_i) \tag{102.195}$$

with

$$W^{int}(\delta u_i, {}^{n+1}_{\;\;0}u_i^{(k)}) = \int_{\Omega_c} \hat{E}_{ij}(\delta u_i, {}^{n+1}_{\;\;0}u_i^{(k)})\; {}^{n+1}_{\;\;0}S_{ij}^{(k)}\; dV \tag{102.196}$$

$$W^{ext}(\delta u_i) = -\int_{\Omega_c} \rho_0\; \delta u_i\; {}^{n+1}_{\;\;0}b_i\; dV - \int_{\partial\Omega_c} \delta u_i\; {}^{n+1}_{\;\;0}t_i\; dS \tag{102.197}$$

and the $\Delta W(\delta u_i, \Delta u_i; u_i^{(k)})$ is the linearization of virtual work

$$\begin{aligned} \Delta W(\delta u_i, \Delta u_i; u_i^{(k)}) &= \lim_{\epsilon \to 0} \frac{\partial W(\delta u_i, u_i + \epsilon \Delta u_i)}{\partial \epsilon} \\ &= \int_{\Omega_c} \hat{E}_{ij}(\delta u_i, u_i)\; dS_{ij}\; dV + \int_{\Omega_c} \Delta\hat{E}_{ij}(\delta u_i, u_i)\; S_{ij} dV \\ &= \int_{\Omega_c} \hat{E}_{ij}(\delta u_i, u_i)\; \mathcal{L}_{ijkl}\; \hat{E}_{kl}(\Delta u_i, u_i)\; dV + \int_{\Omega_c} \Delta\hat{E}_{ij}(\delta u_i, u_i)\; S_{ij}\; dV \end{aligned} \tag{102.198}$$

Here it was used that $dS_{ij} = 1/2\; \mathcal{L}_{ijkl} dC_{kl} = \mathcal{L}_{ijkl}\hat{E}_{kl}(\Delta u_i, u_i)$.

In order to obtain expressions for stiffness matrix, work is done on equation 102.198 in some more details. To this end, equation 102.198 can be rewritten by expanding definitions for $\hat{E}$ as

$$\begin{aligned} &\Delta W(\delta u_i, \Delta u_i; u_i^{(k)}) = \\ &\frac{1}{4}\int_{\Omega_c} \left(\left(\delta u_{j,i} + \delta u_{i,j}\right) + \left(u_{j,r}\delta u_{r,i} + \delta u_{i,r}u_{r,j}\right)\right)\; \mathcal{L}_{ijkl}\; \left(\left(\Delta u_{k,l} + \Delta u_{l,k}\right) + \left(u_{k,s}\Delta u_{s,l} + \Delta u_{l,s}u_{s,k}\right)\right)\; dV + \\ &+ \int_{\Omega_c} \frac{1}{2}\left(\Delta u_{j,l}\delta u_{l,i} + \delta u_{i,l}\Delta u_{l,j}\right) S_{ij}\; dV \end{aligned} \tag{102.199}$$

Or, by conveniently splitting the above equation

$$\begin{aligned} &\Delta^1 W(\delta u_i, \Delta u_i; u_i^{(k)}) = \\ &\frac{1}{4}\int_{\Omega_c} \left(\left(\delta u_{j,i} + \delta u_{i,j}\right) + \left(u_{j,r}\delta u_{r,i} + \delta u_{i,r}u_{r,j}\right)\right)\; \mathcal{L}_{ijkl}\; \left(\left(\Delta u_{k,l} + \Delta u_{l,k}\right) + \left(u_{k,s}\Delta u_{s,l} + \Delta u_{l,s}u_{s,k}\right)\right)\; dV \end{aligned} \tag{102.200}$$

$$\Delta^2 W(\delta u_i, \Delta u_i; u_i^{(k)}) = \int_{\Omega_c} \frac{1}{2}\left(\Delta u_{j,l}\delta u_{l,i} + \delta u_{i,l}\Delta u_{l,j}\right) S_{ij}\; dV \tag{102.201}$$

By further working on equation 102.200

$$\begin{aligned} \Delta^1 W(\delta u_i, \Delta u_i; u_i^{(k)}) = \quad & \int_{\Omega_c} \left(\frac{1}{2}\left(\delta u_{j,i} + \delta u_{i,j}\right)\right)\; \mathcal{L}_{ijkl}\; \left(\frac{1}{2}\left(\Delta u_{k,l} + \Delta u_{l,k}\right)\right)\; dV \\ + \quad & \int_{\Omega_c} \left(\frac{1}{2}\left(\delta u_{j,i} + \delta u_{i,j}\right)\right)\; \mathcal{L}_{ijkl}\; \left(\frac{1}{2}\left(u_{k,s}\Delta u_{s,l} + \Delta u_{l,s}u_{s,k}\right)\right) dV \\ + \quad & \int_{\Omega_c} \frac{1}{2}\left(u_{j,r}\delta u_{r,i} + \delta u_{i,r}u_{r,j}\right) \mathcal{L}_{ijkl}\; \frac{1}{2}\left(u_{k,s}\Delta u_{s,l} + \Delta u_{l,s}u_{s,k}\right) dV \\ + \quad & \int_{\Omega_c} \frac{1}{2}\left(u_{j,r}\delta u_{r,i} + \delta u_{i,r}u_{r,j}\right) \mathcal{L}_{ijkl}\; \left(\frac{1}{2}\left(\Delta u_{k,l} + \Delta u_{l,k}\right)\right)\; dV \end{aligned} \tag{102.202}$$

It should be noted that the Algorithmic Tangent Stiffness (ATS) tensor $\mathcal{L}_{ijkl}$ poses both minor symmetries ($\mathcal{L}_{ijkl} = \mathcal{L}_{jikl} = \mathcal{L}_{ijlk}$). However, Major symmetry cannot be guaranteed. Non–associated flow rules in elastoplasticity lead to the loss of major symmetry ($\mathcal{L}_{ijkl} \neq \mathcal{L}_{klij}$). Moreover, it can be shown (i.e., Jeremić and Sture (1997)) that there is algorithmic induced symmetry loss even for associated flow rules.

With the minor symmetry of $\mathcal{L}_{ijkl}$ one can write (102.202) as:

$$\begin{aligned}\Delta^1 W(\delta u_i, \Delta u_i; u_i^{(k)}) = & \int_{\Omega_c} \delta u_{i,j} \; \mathcal{L}_{ijkl} \; \Delta u_{l,k} dV \\ + & \int_{\Omega_c} \delta u_{i,j} \; \mathcal{L}_{ijkl} \; u_{k,s} \Delta u_{l,s} dV \\ + & \int_{\Omega_c} \delta u_{i,r} u_{r,j} \; \mathcal{L}_{ijkl} \; u_{k,s} \Delta u_{l,s} dV \\ + & \int_{\Omega_c} \delta u_{i,r} u_{r,j} \; \mathcal{L}_{ijkl} \; \Delta u_{l,k} dV \end{aligned} \tag{102.203}$$

Similarly, by observing symmetry of second Piola–Kirchhoff stress tensor S_{ij}

$$\Delta^2 W(\delta u_i, \Delta u_i; u_i^{(k)}) = \int_{\Omega_c} \delta u_{i,l} \Delta u_{l,j} \; S_{ij} dV \tag{102.204}$$

Weak form of equilibrium expressions (i.e. (102.192) and (102.192)) for internal (W^{int}) and external (W^{ext}) virtual work, with the above mentioned symmetry of S_{ij} can be written as

$$W^{int}(\delta u_i, {}^{n+1}_{\;\;0}u_i^{(k)}) = \int_{\Omega_c} \delta u_{i,j} \; S_{ij} dV + \int_{\Omega_c} \delta u_{i,r} u_{r,j} \; S_{ij} dV \tag{102.205}$$

$$W^{ext}(\delta u_i) \;\; = \;\; - \int_{\Omega_c} \rho_0 \; \delta u_i b_i \; dV - \int_{\partial\Omega_c} \delta u_i \; t_i \; dS \tag{102.206}$$

Standard finite element discretization of displacement field yields:

$$u_i \approx \hat{u}_i = H_I \bar{u}_{Ii} \tag{102.207}$$

where $\hat{u}_i$ is the approximation to exact, analytic (if it exists) displacement field u_i, H_I are standard FEM shape functions and $\bar{u}_{Ii}$ are nodal displacements. With this approximation

$$\begin{aligned}\Delta^1 W(\delta u_i, \Delta u_i; u_i^{(k)}) = & \int_{\Omega_c} \left(H_{I,j} \delta \bar{u}_{Ii}\right) \; \mathcal{L}_{ijkl} \; \left(H_{Q,k} \Delta \bar{u}_{Ql}\right) dV \\ + & \int_{\Omega_c} \left(H_{I,j} \delta \bar{u}_{Ii}\right) \; \mathcal{L}_{ijkl} \; \left(H_{J,k} \bar{u}_{Js}\right) \left(H_{Q,s} \Delta \bar{u}_{Ql}\right) dV \\ + & \int_{\Omega_c} \left(H_{I,r} \delta \bar{u}_{Ii}\right) \left(H_{J,j} \bar{u}_{Jr}\right) \mathcal{L}_{ijkl} \; \left(H_{J,k} \bar{u}_{Js}\right) \left(H_{Q,s} \Delta \bar{u}_{Ql}\right) dV \\ + & \int_{\Omega_c} \left(H_{I,r} \delta \bar{u}_{Ii}\right) \left(H_{J,j} \bar{u}_{Jr}\right) \mathcal{L}_{ijkl} \; \left(H_{Q,k} \Delta \bar{u}_{Ql}\right) dV \end{aligned} \tag{102.208}$$

$$\Delta^2 W(\delta u_i, \Delta u_i; u_i^{(k)}) = \int_{\Omega_c} \left(H_{I,l}\delta \bar{u}_{Ii}\right) \left(H_{Q,j}\Delta \bar{u}_{Ql}\right) S_{ij} dV \tag{102.209}$$

$$W^{int}(\delta u_i, {}^{n+1}_{\ \ 0}u_i^{(k)}) = \int_{\Omega_c} \left(H_{I,j}\delta \bar{u}_{Ii}\right)\ S_{ij} dV + \int_{\Omega_c} \left(H_{I,r}\delta \bar{u}_{Ii}\right)\ \left(H_{J,j}\bar{u}_{Jr}\right)\ S_{ij} dV \tag{102.210}$$

$$W^{ext}(\delta u_i) \ = \ -\int_{\Omega_c} \rho_0\ \left(H_I \delta \bar{u}_{Ii}\right)\ b_i\ dV - \int_{\partial\Omega_c} \left(H_I \delta \bar{u}_{Ii}\right)\ t_i\ dS \tag{102.211}$$

Upon noting that virtual nodal displacements δu_{Ii} are any non–zero, continuous displacements, and since they occur in all expressions for linearized virtual work (from Equations (102.194), (102.195), (102.196), (102.197) and (102.198)) they can be factored out so that (while remembering that $\Delta W^1 + \Delta W^2 + W^{ext} + W^{int} = 0$):

$$\begin{aligned}
& \int_{\Omega_c} \left(H_{I,j}\right)\ \mathcal{L}_{ijkl}\ \left(H_{Q,k}\Delta \bar{u}_{Ql}\right) dV \\
+ & \int_{\Omega_c} \left(H_{I,j}\right)\ \mathcal{L}_{ijkl}\ \left(H_{J,k}\bar{u}_{Js}\right) \left(H_{Q,s}\Delta \bar{u}_{Ql}\right) dV \\
+ & \int_{\Omega_c} \left(H_{I,r}\right) \left(H_{J,j}\bar{u}_{Jr}\right) \mathcal{L}_{ijkl}\ \left(H_{J,k}\bar{u}_{Js}\right) \left(H_{Q,s}\Delta \bar{u}_{Ql}\right) dV \\
+ & \int_{\Omega_c} \left(H_{I,r}\right) \left(H_{J,j}\bar{u}_{Jr}\right) \mathcal{L}_{ijkl}\ \left(H_{Q,k}\Delta \bar{u}_{Ql}\right) dV \\
+ & \int_{\Omega_c} \left(H_{I,l}\right) \left(H_{Q,j}\Delta \bar{u}_{Ql}\right) S_{ij} dV \\
+ & \int_{\Omega_c} \left(H_{I,j}\right)\ S_{ij} dV + \int_{\Omega_c} \left(H_{I,r}\right)\ \left(H_{J,j}\bar{u}_{Jr}\right)\ S_{ij} dV \\
= & \int_{\Omega_c} \rho_0\ \left(H_I\right)\ b_i\ dV + \int_{\partial\Omega_c} \left(H_I\right)\ t_i\ dS
\end{aligned} \tag{102.212}$$

By rearranging previous equations, one can write:

$$\begin{aligned}
& \left(\int_{\Omega_c} H_{I,j}\mathcal{L}_{ijkl}H_{Q,k} dV + \int_{\Omega_c} H_{I,j}\mathcal{L}_{ijkl}H_{J,k}\bar{u}_{Js}H_{Q,s} dV + \int_{\Omega_c} H_{I,r}H_{J,j}\bar{u}_{Jr}\mathcal{L}_{ijkl}H_{J,k}\bar{u}_{Js}H_{Q,s} dV\right. \\
+ & \left.\int_{\Omega_c} H_{I,r}H_{J,j}\bar{u}_{Jr}\mathcal{L}_{ijkl}H_{Q,k} dV + \int_{\Omega_c} H_{I,l}H_{Q,j}S_{ij} dV\right) \Delta \bar{u}_{Ql} \\
+ & \int_{\Omega_c} \left(H_{I,j}\right)\ S_{ij} dV + \int_{\Omega_c} \left(H_{I,r}\right)\ \left(H_{J,j}\bar{u}_{Jr}\right)\ S_{ij} dV \\
= & \int_{\Omega_c} \rho_0\ \left(H_I\right)\ b_i\ dV + \int_{\partial\Omega_c} \left(H_I\right)\ t_i\ dS
\end{aligned} \tag{102.213}$$

The vectors of external and internal forces are

$$\mathbf{f}_{int} \ = \ \frac{\partial(W^{int}(\delta u_i, {}^{n+1}_{\ \ 0}u_i^{(k)}))}{\partial(\delta u_i)} \tag{102.214}$$

$$\mathbf{f}_{ext} \ = \ \frac{\partial(W^{ext}(\delta u_i))}{\partial(\delta u_i)} \tag{102.215}$$

The Algorithmic Tangent Stiffness (ATS) tensor $\mathcal{L}^{ATS}_{ijkl}$ is defined as a linearization of second Piola–Kirchhoff stress tensor S_{ij} with respect to the right deformation tensor C_{kl}

$$dS_{ij} = \frac{1}{2}\,\mathcal{L}_{ijkl}\,dC_{kl} \quad \text{with} \quad dC_{kl} = 2\,\hat{E}_{kl}(du_i, u_i) \tag{102.216}$$

Then, the global algorithmic tangent stiffness matrix (tensor) is given as

$$\mathbf{K}_t = \frac{\partial(\Delta W(\delta u_i, \Delta u_i; u_i^{(k)}))}{\partial(\delta u_i)} \tag{102.217}$$

The iterative change in displacement vector Δu_i is obtained by setting a linearized virtual work to zero

$$W(\delta u_i, u_i^{(k+1)}) = 0 \quad \Rightarrow \quad W(\delta u_i, u_i^{(k)}) = -\Delta W(\delta u_i, \Delta u_i; u_i^{(k)}) \tag{102.218}$$

102.13.4.1 Total Lagrangian Format

The undeformed configuration Ω_0 is chosen as the computational domain ($\Omega_c = \Omega_0$). The iterative displacement Δu_i is obtained from the equation

$$W(\delta u_i, {}^{n+1}u_i^{(k)}) = -\Delta W(\delta u_i, \Delta u_i; {}^{n+1}u_i^{(k)}) \tag{102.219}$$

where

$$\begin{aligned} W(\delta u_i, {}^{n+1}u_i^{(k)}) \;\; &= \;\; \int_{\Omega_c} \hat{E}_{ij}(\delta u_i, {}^{n+1}u_i^{(k)})\; {}^{n+1}S_{ij}^{(k)}\; dV \\ &\quad - \int_{\Omega_c} \rho_0\; \delta u_i\; {}^{n+1}b_i\; dV - \int_{\partial\Omega_c} \delta u_i\; {}^{n+1}t_i\; dS \end{aligned} \tag{102.220}$$

and

$$\begin{aligned} \Delta W(\delta u_i, \Delta u_i; {}^{n+1}u_i^{(k)}) \;\; &= \;\; \int_{\Omega_c} \hat{E}_{ij}(\delta u_i, {}^{n+1}u_i^{(k)})\; {}^{n+1}\mathcal{L}_{ijkl}^{(k)}\; \hat{E}_{kl}(\Delta u_i, {}^{n+1}u_i^{(k)})\; dV \\ &\quad + \int_{\Omega_c} d\hat{E}_{ij}(\delta u_i, \Delta u_i)\; {}^{n+1}S_{ij}^{(k)}\; dV \end{aligned} \tag{102.221}$$

In the case of hyperelastic–plastic response, second Piola–Kirchhoff stress ${}^{n+1}S_{ij}^{(k)}$ is obtained by integrating the constitutive law, described in Chapter 106.4. It should be noted that by integrating in the intermediate configuration, obtained is Mandel stress ${}^{n+1}\bar{T}_{ij}$ and subsequently[24] the second Piola–Kirchhoff stress $\bar{S}_{kj}$. The ATS tensor $\bar{\mathcal{L}}_{ijkl}$ is then obtained based on $\bar{S}_{kj}$. In order to obtain second Piola–Kirchhoff stress S_{kj} and ATS tensor in initial configuration, a pull-back from the intermediate configuration to the initial configuration is performed

$${}^{n+1}S_{ij} = {}^{n+1}F^p_{ip}\; {}^{n+1}F^p_{jq}\; {}^{n+1}\bar{S}_{pq} \tag{102.222}$$

$${}^{n+1}\mathcal{L}_{ijkl} = {}^{n+1}F^p_{im}\; {}^{n+1}F^p_{jn}\; {}^{n+1}F^p_{kr}\; {}^{n+1}F^p_{ls}\; {}^{n+1}\bar{\mathcal{L}}_{mnrs} \tag{102.223}$$

---

[24] $\bar{S}_{kj} = \left(\bar{C}_{ik}\right)^{-1} \bar{T}_{ij}$

102.13.5 Finite Element Formulations

Presented here is a slightly different approach to developing large deformation FEM in total Lagrangian form. Lower case indices are used for variables in current configuration, while the capital case indices are used for the reference configuration.

102.13.5.1 Strong Form

The static, strong form of momentum balance in the current configuration is

$$\frac{\partial \sigma_{ij}}{\partial x_j} + \rho b_i = 0 \tag{102.224}$$

where σ_{ij} is the Cauchy stress, rho is the material density, b_i is the material body force.

Used here was the so-called Total Lagrangian formulation that is based on the reference configuration. The strong form of momentum balance can be expressed in the reference configuration

$$\frac{\partial P_{iJ}}{\partial X_J} + \rho_0 b_i = 0 \tag{102.225}$$

where P_{iJ} is the first Pialo-Kirchhoff stress, ρ_0 is the material Lagrangian density and $\rho_0 = J\rho$.

102.13.5.2 Weak Form

The corresponding weak form of Equation 102.225 can be expressed as

$$\int_{\Omega_0} \delta u_i \left(\frac{\partial P_{iJ}}{\partial X_J} + \rho_0 b_i \right) dV = 0 \tag{102.226}$$

where δu_i is some arbitrary virtual displacement, Ω_0 is the concerned domain of the reference configuration. Using the partial integration rule, the above equation can be alternatively expressed as

$$\int_{\Omega_0} \frac{\partial}{\partial X_J} \left(P_{iJ} \delta u_i \right) dV - \int_{\Omega_0} P_{iJ} \frac{\partial \delta u_i}{\partial X_J} dV + \int_{\Omega_0} \rho_0 b_i \delta u_i dV = 0 \tag{102.227}$$

The first term of Equation 102.227 can be rewritten in terms of the surface traction

$$\int_{\Omega_0} \frac{\partial}{\partial X_J} \left(P_{iJ} \delta u_i \right) dV = \int_{\partial\Omega_0} \delta u_i P_{iJ} H_J dA = \int_{\partial\Omega_0} t_i \delta u_i dA \tag{102.228}$$

with the traction $t_i = P_{iJ} H_J$, where H_J the unit surface normal vector in the reference configuration, and $\partial\Omega_0$ is the boundary of the reference domain Ω_0.

The second term of Equation 102.227 can be rewritten as

$$\int_{\Omega_0} P_{iJ} \frac{\partial \delta u_i}{\partial X_J} dV = \int_{\Omega_0} P_{iJ} \delta F_{iJ} dV = \int_{\Omega_0} S_{IJ} \delta E_{IJ} dV \tag{102.229}$$

where S_{IJ} is the second Pialo-Kirchhoff stress and E_{IJ} is the Lagrangian-Green strain. $\delta F_{iJ} = \partial \delta u_i / \partial X_J$ is used.

The overall weak form in the reference configuration is now

$$\int_{\Omega_0} S_{IJ} \delta E_{IJ} dV = \int_{\partial \Omega_0} t_i \delta u_i dA + \int_{\Omega_0} \rho_0 b_i \delta u_i dV \tag{102.230}$$

102.13.5.3 Linearized Form

To utilize the iterative algorithm for incremental strategy, needed is the linearized form the governing equation 102.230.

The first term linearization of Equation 102.230 is

$$\begin{aligned} \Delta \int_{\Omega_0} S_{IJ} \delta E_{IJ} &= \int_{\Omega_0} \left[\Delta S_{IJ} \delta E_{IJ} + S_{IJ} \delta(\Delta E_{IJ}) \right] dV \\ &= \int_{\Omega_0} \left[\mathcal{L}_{IJKL} \Delta E_{KL} \delta E_{IJ} + S_{IJ} \delta(\Delta E_{IJ}) \right] dV \end{aligned} \tag{102.231}$$

where $\mathcal{L}_{IJKL}$ is the Lagrangian stiffness linked the second Pialo-Kirchhoff stress S_{IJ} and the Lagrangian-Green strain E_{KL} by the relation

$$S_{IJ} = \mathcal{L}_{IJKL} E_{KL} \tag{102.232}$$

The linearization of E_{KL} is

$$\Delta E_{KL} = Sym \left(F_K \frac{\partial \Delta u_b}{\partial X_L} \right) \tag{102.233}$$

where Sym is the operator of tensor symmetry, defined as $Sym(A_{ij}) := (1/2)(A_{ij} + A_{ji})$. Similarly,

$$\delta E_{IJ} = Sym \left(F_{aI} \frac{\partial \delta u_a}{\partial X_J} \right) \tag{102.234}$$

and

$$\delta(\Delta E_{IJ}) = Sym \left(\frac{\partial \delta u_c}{\partial X_I} \frac{\partial \Delta u_c}{\partial X_J} \right) \tag{102.235}$$

Note that S_{IJ} is a symmetric tensor, and L_{IJKL} is a tensor with major and minor symmetries, Equation 102.231 can be expressed as

$$\begin{aligned} \Delta \int_{\Omega_0} S_{IJ} \delta E_{IJ} &= \int_{\Omega_0} \left[\mathcal{L}_{IJKL} \left(F_{bK} \frac{\partial \Delta u_b}{\partial X_L} \right) \left(F_{aI} \frac{\partial \delta u_a}{\partial X_J} \right) + S_{IJ} \left(\frac{\partial \delta u_c}{\partial X_I} \frac{\partial \Delta u_c}{\partial X_J} \right) \right] dV \\ &= \int_{\Omega_0} \left[\frac{\partial \delta u_a}{\partial X_J} \left(F_{aI} F_{bK} \mathcal{L}_{IJKL} + \delta_{ab} S_{JL} \right) \frac{\partial \Delta u_b}{\partial X_L} \right] dV \end{aligned} \tag{102.236}$$

The overall linearization form is thus

$$\int_{\Omega_0} \left[\frac{\partial \delta u_a}{\partial X_J} \left(F_{aI} F_{bK} \mathcal{L}_{IJKL} + \delta_{ab} S_{JL} \right) \frac{\partial \Delta u_b}{\partial X_L} \right] dV = \int_{\partial \Omega_0} \Delta t_i \delta u_i dA + \int_{\Omega_0} \rho_0 \Delta b_i \delta u_i dV \tag{102.237}$$

102.13.5.4 Finite Element Form

In finite element form, the displacements u_i are interpolated from the element nodal displacements $\bar{u}_{Ai}$:

$$u_i = H_A \bar{u}_{Ai} \tag{102.238}$$

where H_A is the element shape function of the node A, $\bar{u}_{Ai}$ is the node A displacements, and

$$\frac{\partial \delta u_a}{\partial X_J} = \frac{\partial H_A}{\partial X_J}\delta \bar{u}_{Aa}, \quad \frac{\partial \Delta u_b}{\partial X_L} = \frac{\partial H_B}{\partial X_L}\Delta \bar{u}_{Bb} \tag{102.239}$$

Equation 102.237 can be expressed as

$$\left(K_{AaBb}\ \Delta \bar{u}_{Bb}\ - \Delta f^{ex}_{Aa}\right)\delta \bar{u}_{Aa} = 0 \tag{102.240}$$

where

$$K_{AaBb} = \int_{\Omega_0} \frac{\partial H_A}{\partial X_J}\left(F_{aI}F_{bK}\mathcal{L}_{IJKL} + \delta_{ab}S_{JL}\right)\frac{\partial H_B}{\partial X_L} dV \tag{102.241}$$

$$\Delta f^{ex}_{Aa} = \int_{\partial\Omega_0} H_A \Delta t_i dA + \int_{\Omega_0} \rho_0 H_A \Delta b_i dV \tag{102.242}$$

Due to the arbitrariness of the virtual nodal displacements, the expression in the parentheses should be zero in Equation 102.240, which gives the incremental finite element form:

$$K_{AaBb}\ \Delta \bar{u}_{Bb}\ = \Delta f^{ex}_{Aa} \tag{102.243}$$

102.14 Cosserat Continuum Finite Element Formulation

102.14.1 Introduction

The classical theory of elasticity describes well the behavior of metals, like steel and aluminum. But when the micro-structure of the material becomes significant, like soil and sand, the classical elasticity lacks the ability to represent the granular media properties. In 1909, Cosserat Cosserat (1909) brothers published their prominent work on the Cosserat continua to remove the shortcomings of the classical elasticity. Compared to the classical continua, Cosserat continua has the additional couple stress to reflect the free rotations on the particles.

102.14.1.1 3D Finite Element Formulation for Cosserat Continua

Pothier Pothier and Rencis (1994) proposed the three-dimensional finite element formulation for micro-polar elasticity with both classical strain and the micro-polar strain components. Riahi Dehkordi (2008) developed the finite element Cosserat formulation with the application in layered structures. This article contributes the three-dimensional finite element formulation with the micro-polar strain, which simplifies further development in the elastoplasticity algorithm. In addition, the formulation for the isoparametric brick element is provided.

102.14.1.2 Cosserat Elastoplastic Algorithm

Vardoulakis et al. Vardoulakis (1989); Papamichos et al. (1990) developed the 2D Cosserat plasticity to predict the thickness of shear bands in granular materials. De Borst de Borst (1993, 1991) applied the 2D Cosserat elastoplasticity to analyze the strain localization. Li Li and Bing Chen (2005) applied the 2D pressure-dependent Cosserat elastoplasticity to simulate the strain localization. Grammenoudis Grammenoudis and Tsakmakis (2005) implemented the micro-polar plasticity in the finite deformation framework. This article developed the implicit algorithms for 3D Cosserat plasticity. Rotational kinematic hardening is introduced to the Cosserat elastoplasticity.

102.14.2 Cosserat Elasticity

The Cosserat (micro-polar) elasticity has 6 parameters Pothier and Rencis (1994) , including λ, μ, χ, π_1, π_2, π_3.

The units of λ, μ, and χ are $Newton/(meter^2)$. The units of π_1, π_2, and π_3 are $Newton$.

102.14.2.1 Stress

The stress is the same to the classical elastic stress Riahi and Curran (2009); Dehkordi (2008),

$$\sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{pmatrix} \tag{102.244}$$

where the unit of σ is force per unit area ($Newton/meter^2$).

102.14.2.2 Couple Stress

The couple-stress is the couple stress of the classical elastic stress.

$$\boldsymbol{t} = \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{pmatrix} \tag{102.245}$$

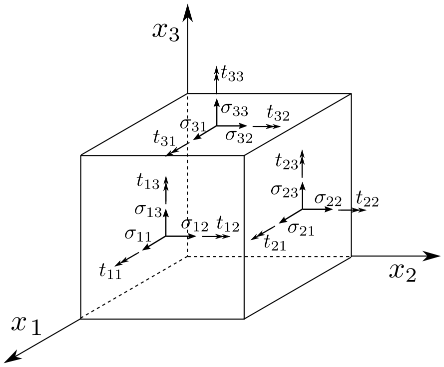


Figure 102.17: Illustration of Stress and Couple Stress

where the unit of $\boldsymbol{t}$ is torque per unit area ($Newton/meter$).

In the classical elasticity, the stress is symmetric: $\sigma_{12} = \sigma_{21}$. This is no longer true in the Cosserat materials. Namely, $\sigma_{12} \neq \sigma_{21}$ due to the couple stress.

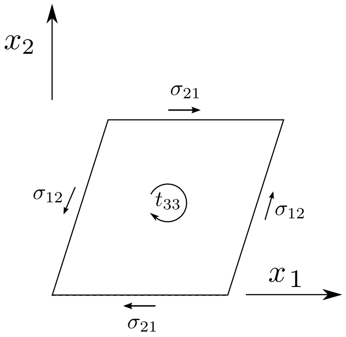


Figure 102.18: Illustration of Asymmetric Stress due to Couple Stress

102.14.2.3 Generalized Stress

In the calculation, for the purpose of simplification, the generalized stress are defined by :

$$
\begin{aligned}
&\boldsymbol{\sigma} = [\sigma \vdots t] \\
&\boldsymbol{\sigma} = [\sigma_{11}\ \sigma_{12}\ \sigma_{13}\ \sigma_{21}\ \sigma_{22}\ \sigma_{23}\ \sigma_{31}\ \sigma_{32}\ \sigma_{33} \vdots t_{11}\ t_{12}\ t_{13}\ t_{21}\ t_{22}\ t_{23}\ t_{31}\ t_{32}\ t_{33}]
\end{aligned}
\qquad (102.246)
$$

Namely, the generalized stress $\boldsymbol{\sigma}$ (18×1) is a horizontal combination of σ and t.

102.14.2.4 Cosserat Strain

$$\epsilon = u_{j,i} - e_{ijk}\phi_k = \begin{pmatrix} u_{1,1} & u_{2,1} - \phi_3 & u_{3,1} + \phi_2 \\ u_{1,2} + \phi_3 & u_{2,2} & u_{3,2} - \phi_1 \\ u_{1,3} - \phi_2 & u_{2,3} + \phi_1 & u_{3,3} \end{pmatrix} \tag{102.247}$$

where e_{ijk} is the permutation symbol.

where u and ϕ are the displacement and micro-rotation respectively.

Note that generally $\epsilon_{12} \neq \epsilon_{21}$ even when $\boldsymbol{\phi} = 0$.

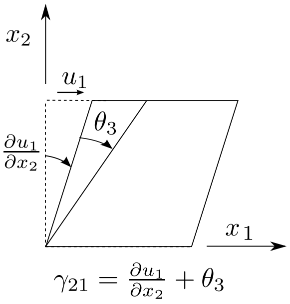


Figure 102.19: Illustration of Asymmetric Strain due to Rotations

102.14.2.5 Curvature

$$\boldsymbol{\omega} = \phi_{i,j} = \begin{pmatrix} \phi_{1,1} & \phi_{1,2} & \phi_{1,3} \\ \phi_{2,1} & \phi_{2,2} & \phi_{2,3} \\ \phi_{3,1} & \phi_{3,2} & \phi_{3,3} \end{pmatrix} \tag{102.248}$$

where ϕ is called the micro-rotation and ω is called the Cosserat rotation gradient (curvature).

102.14.2.6 Generalized Strain

In the calculation, for the purpose of simplification, the Generalized Strain are defined by :

$$\begin{aligned} \xi &= [\epsilon \vdots \omega] \\ \xi &= [\epsilon_{11}\ \epsilon_{12}\ \epsilon_{13}\ \epsilon_{21}\ \epsilon_{22}\ \epsilon_{23}\ \epsilon_{31}\ \epsilon_{32}\ \epsilon_{33} \vdots \omega_{11}\ \omega_{12}\ \omega_{13}\ \omega_{21}\ \omega_{22}\ \omega_{23}\ \omega_{31}\ \omega_{32}\ \omega_{33}] \end{aligned} \tag{102.249}$$

The generalized strain ξ (18×1) is a horizontal combination of ϵ and κ.

102.14.2.7 Constitutive Equations

Force Tangent: The relationship between the stress and strain Pothier and Rencis (1994) is

$$\sigma_{ij} = \lambda\epsilon_{kk}\delta_{ij} + \mu\epsilon_{ij} + (\mu + \chi)\epsilon_{ji} \tag{102.250}$$

Therefore, the tangent between stress and strain is

$$E_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu \delta_{ik} \delta_{jl} + (\mu + \chi) \delta_{jk} \delta_{il} \tag{102.251}$$

Curvature Tangent: The relationship between the couple-stress and curvature is

$$t_{ij} = \pi_1 \omega_{kk} \delta_{ij} + \pi_2 \omega_{ij} + \pi_3 \omega_{ji} \tag{102.252}$$

Therefore, the tangent between couple-stress and curvature is

$$C_{ijkl} = \pi_1 \delta_{ij} \delta_{kl} + \pi_2 \delta_{ik} \delta_{jl} + \pi_3 \delta_{jk} \delta_{il} \tag{102.253}$$

102.14.2.8 Relation to the classical elasticity tangent

When $\chi = \pi_1 = \pi_2 = \pi_3 = 0$, the curvature tangent is zero, and the force tangent is the same to the classical elasticity tangent.

Full tangent matrix for one Cosserat point:

$$\mathcal{D}^{full}_{18\times 18} = \begin{pmatrix} D^{force}_{9\times 9} & [0]_{9\times 9} \\ [0]_{9\times 9} & D^{curvature}_{9\times 9} \end{pmatrix}_{18\times 18} \tag{102.254}$$

102.14.2.9 Flatten the tensor of force tangent and curvature tangent

Corresponding to the defined generalized stress and generalized strain, flatten the $3 \times 3 \times 3 \times 3$ tensor C_{ijkl} to 9×9 matrix D.

$$D_{9\times 9} = \begin{pmatrix} C_{1111} & C_{1112} & C_{1113} & C_{1121} & C_{1122} & C_{1123} & C_{1131} & C_{1132} & C_{1133} \\ C_{1211} & C_{1212} & C_{1213} & C_{1221} & C_{1222} & C_{1223} & C_{1231} & C_{1232} & C_{1233} \\ C_{1311} & C_{1312} & C_{1313} & C_{1321} & C_{1322} & C_{1323} & C_{1331} & C_{1332} & C_{1333} \\ C_{2111} & C_{2112} & C_{2113} & C_{2121} & C_{2122} & C_{2123} & C_{2131} & C_{2132} & C_{2133} \\ C_{2211} & C_{2212} & C_{2213} & C_{2221} & C_{2222} & C_{2223} & C_{2231} & C_{2232} & C_{2233} \\ C_{2311} & C_{2312} & C_{2313} & C_{2321} & C_{2322} & C_{2323} & C_{2331} & C_{2332} & C_{2333} \\ C_{3111} & C_{3112} & C_{3113} & C_{3121} & C_{3122} & C_{3123} & C_{3131} & C_{3132} & C_{3133} \\ C_{3211} & C_{3212} & C_{3213} & C_{3221} & C_{3222} & C_{3223} & C_{3231} & C_{3232} & C_{3233} \\ C_{3311} & C_{3312} & C_{3313} & C_{3321} & C_{3322} & C_{3323} & C_{3331} & C_{3332} & C_{3333} \end{pmatrix}_{9\times 9} \tag{102.255}$$

102.14.3 3D Finite Element Formulation

102.14.3.1 Force Equilibrium

$\sigma_{ji,j} + b_i = \rho \ddot{u}_i$ in dynamic problems.

$$\begin{aligned}
\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{yx}}{\partial y} + \frac{\partial \sigma_{zx}}{\partial z} + b_x = \rho \ddot{u}_x \\
\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{zy}}{\partial z} + b_y = \rho \ddot{u}_y \\
\frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{yz}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} + b_z = \rho \ddot{u}_z
\end{aligned} \tag{102.256}$$

where b_i is the body force, and $\ddot{u}_i$ is the acceleration of the point.

102.14.4 Momentum Equilibrium

$t_{ji,j} + e_{ijk}\sigma_{jk} + M_i = J\ddot{\theta}_i$ in dynamic problems.

$$\begin{aligned}
\frac{\partial t_{xx}}{\partial x} + \frac{\partial t_{yx}}{\partial y} + \frac{\partial t_{zx}}{\partial z} + \sigma_{xy} - \sigma_{yx} + M_x = J_x \theta_x \\
\frac{\partial t_{xy}}{\partial x} + \frac{\partial t_{yy}}{\partial y} + \frac{\partial t_{zy}}{\partial z} + \sigma_{zx} - \sigma_{xz} + M_y = J_y \theta_y \\
\frac{\partial t_{xz}}{\partial x} + \frac{\partial t_{yz}}{\partial y} + \frac{\partial t_{zz}}{\partial z} + \sigma_{yz} - \sigma_{zy} + M_z = J_z \theta_z
\end{aligned} \tag{102.257}$$

where M_i is the body rotation force per unit area ($Newton/meter^2$).

102.14.4.1 Illustration in 2D

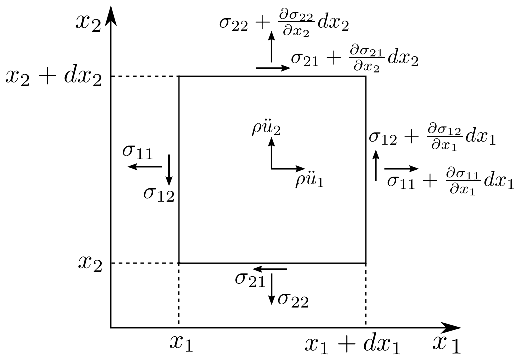


Figure 102.20: Illustration of Force Equilibrium in Cosserat Continua

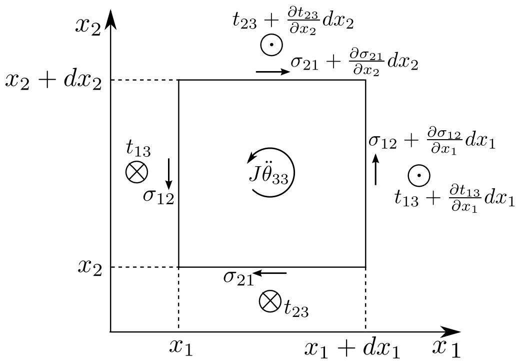


Figure 102.21: Illustration of Moment Equilibrium in Cosserat Continua

102.14.4.2 Strain-Displacement Relationship

Define the DOFs and the generalized strain. For the 8-node brick element. Each node has 6 DOFs.

The vector of nodal DOFs is defined by :

$$\mathcal{U} = [\boldsymbol{u} \;\; \phi]^T = [u_x \; u_y \; u_z \; \phi_x \; \phi_y \; \phi_z]^T_{1\times 6} \tag{102.258}$$

The vector of generalized strain ξ, (18×1) is a horizontal combination of ϵ and κ.

$$\xi = [\epsilon_{11} \; \epsilon_{12} \; \epsilon_{13} \; \epsilon_{21} \; \epsilon_{22} \; \epsilon_{23} \; \epsilon_{31} \; \epsilon_{32} \; \epsilon_{33} \vdots \omega_{11} \; \omega_{12} \; \omega_{13} \; \omega_{21} \; \omega_{22} \; \omega_{23} \; \omega_{31} \; \omega_{32} \; \omega_{33}] \tag{102.259}$$

Strain-displacement for one node. Therefore,

$$\xi_{18\times 1} = \mathcal{B}_{18\times 6} \mathcal{U}_{6\times 1} \tag{102.260}$$

So the dimension of matrix $\mathcal{B}$ (strain-displacement matrix) for 1 node is 18×6.

According to the definition of force strain (Eq 102.247) and curvature (Eq 102.248), express the matrix $\mathcal{B}$ for 1 node Riahi and Curran (2009); Padovan (1978) by

$$\mathcal{B}_N = \begin{pmatrix} \mathcal{B}_{Na} & \mathcal{B}_{Nb} \\ [0]_{9\times 3} & \mathcal{B}_{Nc} \end{pmatrix}_{18\times 6} \tag{102.261}$$

where

$$\mathcal{B}_{Na} = \begin{pmatrix} H_{N,x} & 0 & 0 \\ 0 & H_{N,x} & 0 \\ 0 & 0 & H_{N,x} \\ H_{N,y} & 0 & 0 \\ 0 & H_{N,y} & 0 \\ 0 & 0 & H_{N,y} \\ H_{N,z} & 0 & 0 \\ 0 & H_{N,z} & 0 \\ 0 & 0 & H_{N,z} \end{pmatrix}_{9\times 3}, \quad \mathcal{B}_{Nb} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -H_N \\ 0 & +H_N & 0 \\ 0 & 0 & +H_N \\ 0 & 0 & 0 \\ -H_N & 0 & 0 \\ 0 & -H_N & 0 \\ +H_N & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}_{9\times 3}, \tag{102.262}$$

$$\mathcal{B}_{Nc} = \begin{pmatrix} -H_{N,x} & 0 & 0 \\ -H_{N,y} & 0 & 0 \\ -H_{N,z} & 0 & 0 \\ 0 & -H_{N,x} & 0 \\ 0 & -H_{N,y} & 0 \\ 0 & -H_{N,z} & 0 \\ 0 & 0 & -H_{N,x} \\ 0 & 0 & -H_{N,y} \\ 0 & 0 & -H_{N,z} \end{pmatrix}_{9\times 3} \tag{102.263}$$

where H is the shape function. N is the node number, which can be 1, 2, ..., 8.

Strain-displacement for 8 node brick. So the dimension of matrix $\mathcal{B}$ (strain-displacement matrix) for 8 node brick is 18×48.

$$\mathscr{B} = [\ \mathcal{B}_1\ \ \mathcal{B}_2\ \ \mathcal{B}_3\ \ \mathcal{B}_4\ \ \mathcal{B}_5\ \ \mathcal{B}_6\ \ \mathcal{B}_7\ \ \mathcal{B}_8\ ]_{18\times 48} \tag{102.264}$$

Express the 8 node displacement by

$$\mathscr{U} = [\ \mathcal{U}_1\ \ \mathcal{U}_2\ \ \mathcal{U}_3\ \ \mathcal{U}_4\ \ \mathcal{U}_5\ \ \mathcal{U}_6\ \ \mathcal{U}_7\ \ \mathcal{U}_8\ ]^T_{48\times 1} \tag{102.265}$$

Note that matrix $\mathscr{B}$ is a field function obtained by the interpolation of the displacements at 8 nodes.

For a specified Gauss point, the Gauss coordinates should be substituted to calculate the corresponding generalized strain.

So the generalized strain at a specific Gauss coordinates is

$$\mathscr{E}|_{x_i,y_i,z_i} = \mathscr{B}|_{x_i,y_i,z_i}\mathscr{U} \tag{102.266}$$

102.14.4.3 Isoparametric 8 Node Brick

For the purpose of completeness, the shape function Bathe (1996) for the isoparametric 8 node brick is given by

$$H_i = \frac{1}{8}(1 + \xi_i \xi)(1 + \eta_i \eta)(1 + \zeta_i \zeta) \tag{102.267}$$

where the values of ξ_i, η_i, ζ_i for eight nodes are listed.

Node	ξ_i	η_i	ζ_i
1	+1	+1	+1
2	-1	+1	+1
3	-1	-1	+1
4	+1	-1	+1
5	+1	-1	+1
6	-1	+1	-1
7	-1	-1	-1
8	+1	-1	-1

In addition, the shape function is used for both the displacement interpolation in Eq (102.268) and coordinate transformation formula in Eq (102.269) from the global coordinate to the natural coordinate system.

$$\begin{aligned} u_x &= \sum_{i=1}^{8} H_i(\xi,\eta,\zeta) u_x^i & u_y &= \sum_{i=1}^{8} H_i(\xi,\eta,\zeta) u_y^i & u_z &= \sum_{i=1}^{8} H_i(\xi,\eta,\zeta) u_z^i \\ \phi_x &= \sum_{i=1}^{8} H_i(\xi,\eta,\zeta) \phi_x^i & \phi_y &= \sum_{i=1}^{8} H_i(\xi,\eta,\zeta) \phi_y^i & \phi_z &= \sum_{i=1}^{8} H_i(\xi,\eta,\zeta) \phi_z^i \end{aligned} \tag{102.268}$$

The coordinate transformation formula from the global coordinate to the natural coordinate system is

$$x = \sum_{i=1}^{8} H_i(\xi,\eta,\zeta) x_i \qquad y = \sum_{i=1}^{8} H_i(\xi,\eta,\zeta) y_i \qquad z = \sum_{i=1}^{8} H_i(\xi,\eta,\zeta) z_i \tag{102.269}$$

where the x_i, y_i, z_i are the nodal coordinates. Therefore, with the help of Eq (102.269), the global coordinate (x, y, z) become a function of (ξ, η, ζ), which is the natural coordinate in the isoparametric element.

To construct the strain-displacement matrix, the derivative with respect to the global coordinate (x, y, z) is required. Jacobian transformation is used to calculate the derivatives in Eq (102.262, 102.263).

$$\begin{pmatrix} \frac{\partial H_i}{\partial x} \\ \\ \frac{\partial H_i}{\partial y} \\ \\ \frac{\partial H_i}{\partial z} \end{pmatrix} = \boldsymbol{J}^{-1} \begin{pmatrix} \frac{\partial H_i}{\partial \xi} \\ \\ \frac{\partial H_i}{\partial \eta} \\ \\ \frac{\partial H_i}{\partial \zeta} \end{pmatrix} \tag{102.270}$$

where

$$\boldsymbol{J} = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{8} \frac{\partial H_i}{\partial \xi} x_i & \sum_{i=1}^{8} \frac{\partial H_i}{\partial \xi} y_i & \sum_{i=1}^{8} \frac{\partial H_i}{\partial \xi} z_i \\ \\ \sum_{i=1}^{8} \frac{\partial H_i}{\partial \eta} x_i & \sum_{i=1}^{8} \frac{\partial H_i}{\partial \eta} y_i & \sum_{i=1}^{8} \frac{\partial H_i}{\partial \eta} z_i \\ \\ \sum_{i=1}^{8} \frac{\partial H_i}{\partial \zeta} x_i & \sum_{i=1}^{8} \frac{\partial H_i}{\partial \zeta} y_i & \sum_{i=1}^{8} \frac{\partial H_i}{\partial \zeta} z_i \end{pmatrix} \tag{102.271}$$

Note that $\mathcal{B}_{Nb}$ in Eq (102.262) has the original shape functions. Jacobian transformation is not required on $\mathcal{B}_{Nb}$ because the interpolated rotation is used directly to construct the strain in Eq (102.247)
.

Chapter 103

Micromechanical Origins of Elasto-Plasticity

(1994-2002-2010-2019-2021-)

103.1 Chapter Summary and Highlights

This chapter is based in large part on lecture notes by Prof. Stein Sture (Sture, 1993).

103.2 Friction

103.2.1 Early Works

- Leonardo da Vinci, (1452-1519), worked and wrote about friction in 1493... (https://en.wikipedia.org/wiki/Leonardo_da_Vinci)
- Guillaume Amontons (1663-1705), Law of friction, rediscovered in 1699 (since da Vinci's notes were lost), (https://en.wikipedia.org/wiki/Guillaume_Amontons)
- Charles-Augistine de Coulomb (1736-1806), verified laws of friction in 1791 (https://en.wikipedia.org/wiki/Charles-Augustin_de_Coulomb)

Surface with asperities
grain assemblies, loose and dense
Saw-Teeth model analog
Frictional response of solids, friction angle for the polished mineral and dilatancy angle
Particle shapes
Particle rotations
types of packaging of particles

103.3 Particle Contact Mechanics

103.3.1 Particle Contact Mechanics, Axial Behavior

- Hertz contact theory, elastic (1885)
- Cattaneo and Midnlin theories, 1938 elastic and plastic contact

Two equal particles in normal contact
equations for a, Δ, σ_N, σ_N^{max}
Two unequal particles in normal contact
average normal stress and average normal strain
tangent stiffness, function of $\sqrt[3]{\sigma_{avg}}$

Bulk modulus for loose and dense packaging

Mindlin and Deresiewicz (1953)

Hashin (1983)

Rowe (1962)

Cosserat (1909)

103.3.2 Particle Contact Mechanics, Shear Behavior

Shear Behavior

closed form solution for no-slip behavior

equilibrium, integral of τ

slip ring

SLIP ring

There is NO elastic behavior of particles in contact if there is ANY small amount of shear!

Example values for typical contact stress parameters

typical stiffness parameters for sands

103.4 Dilatancy

Chapter 104

Small Deformation Elasto-Plasticity

(1991-1994-2002-2006-2010-2016-2017-2018-2019-2020-2021-)

(In collaboration with Prof. Zhaohui Yang, Dr. Zhao Cheng, Dr. Nima Tafazzoli, Dr. Federico Pisanò, and Prof. Han Yang)

104.1 Chapter Summary and Highlights

104.2 Elasto–plasticity

104.2.1 Constitutive Relations for Infinitesimal Plasticity

A wide range of elasto–plastic materials can be characterized by means of a set of incremental constitutive relations of the general form:

$$d\epsilon_{ij} = d\epsilon^{e}_{ij} + d\epsilon^{p}_{ij} \tag{104.1}$$

$$d\sigma_{ij} = E_{ijkl} d\epsilon^{e}_{kl} \tag{104.2}$$

$$d\epsilon^{p}_{ij} = d\lambda \, \frac{\partial Q}{\partial \sigma_{ij}} = d\lambda \, m_{ij}(\sigma_{ij}, q_{*}) \tag{104.3}$$

$$dq_{*} = d\lambda \, h_{*}(\tau_{ij}, q_{*}) \tag{104.4}$$

where, following standard notation ϵ_{ij}, ϵ^{e}_{ij} and ϵ^{p}_{ij} denotes the total, elastic and plastic strain tensor, (and $d\epsilon_{ij}$ is an increment of a strain tensor ϵ_{ij}), σ_{ij} is the Cauchy stress tensor, and q_{*} signifies some suitable set of internal variables[1]. The asterisk in the place of indices in q_{*} replaces n indices[2]. Equation (104.1) expresses the commonly assumed additive decomposition of the infinitesimal strain tensor into elastic and plastic parts. Equation (104.2) represents the generalized Hooke's law[3] which linearly relates stresses and elastic strains through a stiffness modulus tensor E_{ijkl}. Equation (104.3) expresses a generally associated or non-associated flow rule for the plastic strain and (104.4) describes a suitable set of hardening laws, which govern the evolution of the plastic variables. In these equations, m_{ij} is the plastic flow direction, h_{*} the plastic moduli and $d\lambda$ is a plastic parameter to be determined with the aid of the loading—unloading criterion, which can be expressed in terms of the Karush–Kuhn–Tucker condition (Karush, 1939; Kuhn and Tucker, 1951) as:

$$F(\sigma_{ij}, q_{*}) \leq 0 \tag{104.5}$$

---

[1] In the simplest models of plasticity the internal variables are taken as either plastic strain components ϵ^{p}_{ij} or the hardening variables κ defined, for example as a function of inelastic (plastic) work, i.e. $\kappa = f\left(W^{p}\right)$. See Lubliner (1990) page 115.

[2] for example $_{ij}$ if the variable is ϵ^{p}_{ij}, or nothing if the variable is a scalar value, i.e. κ .

[3] also Eq. 104.157

$$d\lambda \geq 0 \tag{104.6}$$

$$F\ d\lambda = 0 \tag{104.7}$$

In the previous equations $F(\sigma_{ij}, q_*)$ denotes the yield function of the material and (104.5) characterizes the corresponding elastic domain, which is presumably convex. Along any process of loading, conditions (104.5), (104.6) and (104.7) must hold simultaneously. For $F < 0$, equation (104.7) yields $d\lambda = 0$, i.e. elastic behavior, while plastic flow is characterized by $d\lambda > 0$, which with (104.7) is possible only if the yield criterion is satisfied, i.e. $F = 0$. From the latter constraint, in the process of plastic loading the plastic consistency conditions[4] is obtained in the form:

$$dF = \frac{\partial F}{\partial \sigma_{ij}} d\sigma_{ij} + \frac{\partial F}{\partial q_*} dq_* = n_{ij} d\sigma_{ij} + \xi_* dq_* = 0 \tag{104.8}$$

where :

$$n_{ij} = \frac{\partial F}{\partial \sigma_{ij}} \tag{104.9}$$

$$\xi_* = \frac{\partial F}{\partial q_*} \tag{104.10}$$

Equation (104.8) has the effect of confining the stress trajectory to the yield surface[5]. It is worthwhile noting that n_{ij} and ξ_* are normals to the yield surface in stress space and the plastic variable space respectively.

An interesting alternative way of representing non–associated flow rules can be found in Runesson (1987). A fictitious plastic strain derived from associated flow rule, e^p_{ij} is introduced. This fictitious plastic strain is assumed to be related to the real plastic strain ϵ^p_{ij}, which is derived from a non–associated flow rule[6] through the linear transformation:

$$e^p_{ij} = A_{ijkl} \epsilon^p_{kl} \tag{104.11}$$

Linear transformation tensor A_{ijkl} may be state dependent in general case, and it reduces to the symmetric part of the fourth order identity tensor[7] for the case of associated plasticity.

---

[4] first order accuracy condition.

[5] Since it is only linear expansion stress trajectory is confined to the tangential plane only.

[6] as in equation 104.3.

[7] $A_{ijkl} \equiv I^{sym}_{ijkl} \equiv \frac{1}{2}\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right)$.

It is often of interest to model deviatoric strains by an associated flow rule while the volumetric part is non–associated. For this case, A_{ijkl} can be formulated as:

$$A_{ijkl} = \left(\beta \frac{1}{3} \left(\delta_{ij} \delta_{kl} \right) + \frac{1}{2} \left(\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk} \right) \right) \tag{104.12}$$

$$A^{-1}_{ijkl} = \left(- \frac{\beta}{1+\beta} \frac{1}{3} \left(\delta_{ij} \delta_{kl} \right) + \frac{1}{2} \left(\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk} \right) \right) \tag{104.13}$$

and it is obvious that the non–associated flow rule is obtained with $\beta \neq 0$ and the associated flow rule with $\beta = 0$. It is useful to choose $\beta \geq 0$ and retain nice, positive definite properties of adjusted constitutive tensors later.

Let the $\| \cdot \|$ norm, be the complementary energy norm[8]:

$$\| \sigma_{ij} \|^2 = \sigma_{ij} D_{ijkl} \sigma_{kl} \tag{104.14}$$

where D_{ijkl} is the elastic compliance tensor ($D_{ijkl} = E^{-1}_{ijkl}$), and let us introduce the adjusted complementary energy norm as:

$$^{A}\| \sigma_{ij} \|^2 = \sigma_{ij} \left(A_{ijkl} D_{klmn} \right) \sigma_{mn} = \sigma_{ij} \left({}^{A}D_{ijmn} \right) \sigma_{mn} \tag{104.15}$$

where ${}^{A}D_{ijmn}$ is the elastic compliance tensor transformed with respect to the non–associativity involved. It is clear that when $A_{ijkl} \equiv I^{sym}_{ijkl} \Longrightarrow {}^{A}\| \sigma_{ij} \|^2 \equiv \| \sigma_{ij} \|^2$

104.2.2 On Integration Algorithms

In the section Constitutive Relations for Infinitesimal Plasticity we have summarized constitutive equations that are capable of representing a wide variety of elasto–plastic materials. The problem in Computational Elasto–plasticity is to devise accurate and efficient algorithms for the integration of such constitutive relations. In the context of finite element analysis using isoparametric elements, the integration of constitutive equations is carried out at Gauss points. In each step the deformation increments are given or known, and the unknowns to be found are updated stresses and plastic variables. According to Ortiz and Popov (1985) an acceptable algorithm should satisfy:

- consistency with the constitutive relations to be integrated or first order accuracy,
- Numerical stability,

---

[8] This norm will be reintroduced later on!

- incremental plastic consistency

A non—required but desirable feature to be added to the above list is:

- higher[9] order accuracy

First two conditions are needed for attaining convergence for the numerical solution as the step or increment becomes vanishingly small. The third condition is the algorithmic counterpart of the plastic consistency condition and requires that the state of stress computed from the algorithm be contained within the elastic domain.

104.2.3 Midpoint Rule Algorithm

A class of algorithms for integrating constitutive equations with potential to satisfy the above mentioned conditions are the Generalized Midpoint rule algorithms. They are given in the following form:

$$d({}^{n+1}\sigma_{ij}) = E_{ijkl}\left(d({}^{n+1}\epsilon_{kl}) - d({}^{n+1}\epsilon^{p}_{kl})\right) \tag{104.16}$$

$$d({}^{n+1}\epsilon^{p}_{ij}) = d({}^{n}\epsilon^{p}_{ij}) + d\lambda\ {}^{n+\alpha}m_{ij} \tag{104.17}$$

$$d({}^{n+1}q_{*}) = d({}^{n}q_{*}) + d\lambda\ {}^{n+\alpha}h_{*} \tag{104.18}$$

$$F_{n+1} = 0 \tag{104.19}$$

where:

$${}^{n+\alpha}m_{ij} = m_{ij}\left((1-\alpha)\ {}^{n}\sigma_{ij} + \alpha\ \left({}^{n+1}\sigma_{ij}\right), (1-\alpha)\ {}^{n}q_{*} + \alpha\ \left({}^{n+1}q_{*}\right)\right) \tag{104.20}$$

$${}^{n+\alpha}h_{*} = h_{*}\left((1-\alpha)\ {}^{n}\sigma_{ij} + \alpha\ \left({}^{n+1}\sigma_{ij}\right), (1-\alpha)\ {}^{n}q_{*} + \alpha\ \left({}^{n+1}q_{*}\right)\right) \tag{104.21}$$

It is quite clear that the case $\alpha = 0$ corresponds to the Forward Euler approach[10], the case $\alpha = 1$ corresponds to the Backward Euler approach[11], and the case $\alpha = 1/2$ to the Crank – Nicholson scheme. Equations (104.16), (104.17), (104.18), (104.19), (104.20) and (104.21) are the nonlinear algebraic

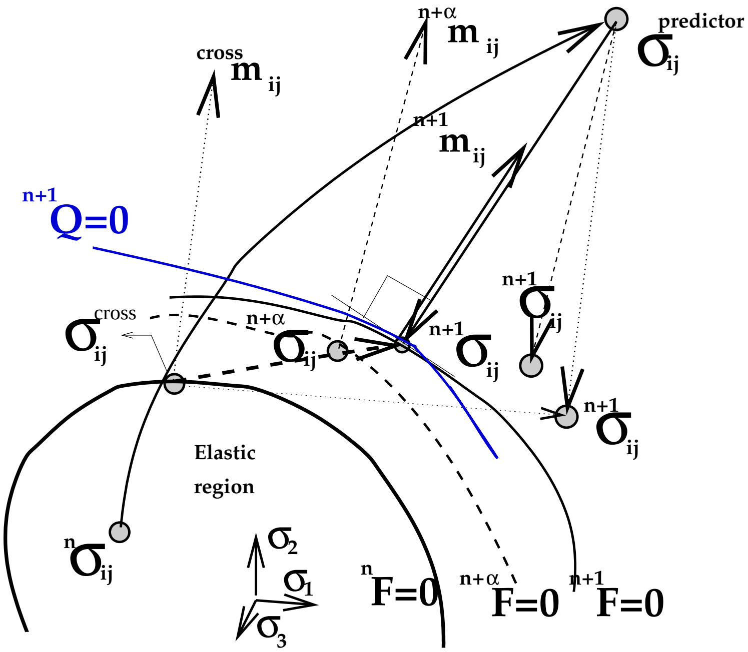


Figure 104.1: integration algorithms in elasto–plasticity

equations to be solved for the unknowns $d(^{n+1}\sigma_{ij})$, $d(^{n+1}\epsilon^p_{ij})$, $d(^{n+1}q_*)$ and $d\lambda$. From the Figure (104.1)[12] it can be seen that the Generalized Midpoint rule may be regarded as a returning mapping algorithm in which the elastic predictor $^{pred}\sigma_{ij}$ is projected on the updated yield surface along the flow direction evaluated at the midpoint $\left(^{n+\alpha}\sigma_{ij}, {}^{n+\alpha}q_*\right)$.

104.2.3.1 Accuracy Analysis

Bearing in mind the context of the displacement based finite element analysis the integration of constitutive equations is performed for the given strain increment. The updated strains $^{n+1}\epsilon_{ij} = \epsilon_{ij}\,(t_n + \Delta t)$ may be viewed as the known function of the step size Δt. The remaining updated variables $^{n+1}\sigma_{ij}$, $^{n+1}\epsilon^p_{ij}$, $^{n+1}q_*$, as well as the incremental plastic parameter λ become functions of Δt implicitly defined through equations (104.16), (104.17), (104.18) and (104.19). It should be clear from (104.16), (104.17), (104.18) and (104.19) that as $\Delta t \to 0$ than $^{n+1}\epsilon_{ij} \to {}^{n}\epsilon_{ij}$, and thus the limiting values of $^{n+1}\sigma_{ij}$, $^{n+1}\epsilon^p_{ij}$, $^{n+1}q_*$ and λ are obtained:

---

[9] at least second order accuracy.

[10] explicit scheme.

[11] implicit scheme.

[12] it should be pointed out that the vectors, as drawn on this figure, are pointing in the right direction only if we assume that $E_{ijkl} \equiv I_{ijkl}$. For any general elasticity tensor E_{ijkl} all vectors are defined in the E_{ijkl} metric, so the term "normal", as we are used to it, does not apply here.

$$
\begin{aligned}
\lim_{\Delta t \to 0} \left({}^{n+1}\sigma_{ij} \right) &= {}^{n}\sigma_{ij} \\
\lim_{\Delta t \to 0} \left({}^{n+1}\epsilon^{p}_{ij} \right) &= {}^{n}\epsilon^{p}_{ij} \\
\lim_{\Delta t \to 0} \left({}^{n+1}q_{*} \right) &= {}^{n}q_{*} \\
\lim_{\Delta t \to 0} \lambda &= 0
\end{aligned}
\tag{104.22}
$$

It can also be argued that, by virtue of the implicit function theorem ((Abraham et al., 1988) Chapter 2.5), ${}^{n+1}\sigma_{ij}$, ${}^{n+1}\epsilon^{p}_{ij}$, ${}^{n+1}q_{*}$ and λ are differentiable functions of Δt, if the functions ${}^{n+\alpha}m_{ij}$, ${}^{n+\alpha}h_{*}$ and F are sufficiently smooth. Sufficient smoothness will be assumed as needed.

First Order Accuracy. First order accuracy[13] of the algorithm, defined by the equations (104.16), (104.17), (104.18) and (104.19) with the constitutive equations given by (104.1), (104.2), (104.3) and (104.4) necessitates that the numerically integrated variables ${}^{n+1}\sigma_{ij}$, ${}^{n+1}\epsilon^{p}_{ij}$ and ${}^{n+1}q_{*}$ agree with their exact values $\sigma_{ij}(t+\Delta t)$, $\epsilon^{p}_{ij}(t+\Delta t)$ and $q_{*}(t+\Delta t)$ to within second order terms in the Taylor's expansion around the initial state ${}^{n}\sigma_{ij} = \sigma_{ij}(t)$, ${}^{n}\epsilon^{p}_{ij} = \epsilon^{p}_{ij}(t)$ and ${}^{n}q_{*} = q_{*}(t)$ in Δt. First order accuracy can be written in the following form:

$$
\lim_{\Delta t \to 0} \frac{d\left({}^{n+1}\sigma_{ij}\right)}{d(\Delta t)} = \frac{d\left({}^{n}\sigma_{ij}\right)}{d(\Delta t)} = E_{ijkl}\left(\frac{d\left({}^{n}\epsilon_{ij}\right)}{d(\Delta t)} - \frac{d\left({}^{n}\epsilon^{p}_{ij}\right)}{d(\Delta t)} \right)
\tag{104.23}
$$

$$
\lim_{\Delta t \to 0} \frac{d\left({}^{n+1}\epsilon^{p}_{ij}\right)}{d(\Delta t)} = \frac{d\left({}^{n}\epsilon^{p}_{ij}\right)}{d(\Delta t)} = \frac{d\left({}^{n}\lambda\right)}{d(\Delta t)} {}^{n}m_{ij}
\tag{104.24}
$$

$$
\lim_{\Delta t \to 0} \frac{d\left({}^{n+1}q_{*}\right)}{d(\Delta t)} = \frac{d\left({}^{n}q_{*}\right)}{d(\Delta t)} = \frac{d\left({}^{n}\lambda\right)}{d(\Delta t)} {}^{n}h_{*}
\tag{104.25}
$$

$$
\lim_{\Delta t \to 0} \frac{d(\lambda)}{d(\Delta t)} = \frac{d\left({}^{n}\lambda\right)}{d(\Delta t)}
\tag{104.26}
$$

and the plastic parameter $d\left({}^{n}\lambda\right)/d(\Delta t)$ is determined with the aid of the plastic consistency condition at t:

---

[13] first order consistency.

$$\frac{d\ \left({}^{n}F\right)}{d\left(\Delta t\right)} = \frac{\partial\left({}^{n}F\right)}{\partial\sigma_{ij}}\frac{d\sigma_{ij}}{d\left(\Delta t\right)} + \frac{\partial\left({}^{n}F\right)}{\partial q_{*}}\frac{dq_{*}}{d\left(\Delta t\right)} = {}^{n}n_{ij}\frac{d\sigma_{ij}}{d\left(\Delta t\right)} + {}^{n}\xi_{*}\frac{dq_{*}}{d\left(\Delta t\right)} = 0 \tag{104.27}$$

It is now rather straightforward to check whether the Generalized Midpoint rule satisfies the consistency conditions as given by (104.23), (104.24), (104.25) and (104.26). We can proceed further on by differentiating (104.16), (104.17), (104.18) and (104.19) with respect to Δt[14]:

$$\frac{d\left({}^{n+1}\sigma_{ij}\right)}{d\left(\Delta t\right)} = E_{ijkl}\left(\frac{d\left({}^{n+1}\epsilon_{kl}\right)}{d\left(\Delta t\right)} - \frac{d\left({}^{n+1}\epsilon^{p}_{kl}\right)}{d\left(\Delta t\right)}\right) \tag{104.28}$$

$$\begin{aligned}\frac{d\left({}^{n+1}\epsilon^{p}_{ij}\right)}{d\left(\Delta t\right)} &= \frac{d\lambda}{d\left(\Delta t\right)}\left({}^{n+\alpha}m_{ij}\right) + \lambda\,\frac{d\left({}^{n+\alpha}m_{ij}\right)}{d\left(\Delta t\right)} = \\ \frac{d\lambda}{d\left(\Delta t\right)}\left({}^{n+\alpha}m_{ij}\right) &+ \lambda\alpha\left(\left.\frac{\partial m_{ij}}{\partial\sigma_{ij}}\right|_{n+1}\frac{d\left({}^{n+1}\sigma_{ij}\right)}{d\left(\Delta t\right)} + \left.\frac{\partial m_{ij}}{\partial q_{*}}\right|_{n+1}\frac{d\left({}^{n+1}q_{*}\right)}{d\left(\Delta t\right)}\right)\end{aligned} \tag{104.29}$$

$$\begin{aligned}\frac{d\left({}^{n+1}q^{p}_{*}\right)}{d\left(\Delta t\right)} &= \frac{d\lambda}{d\left(\Delta t\right)}\left({}^{n+\alpha}h_{*}\right) + \lambda\frac{d\left({}^{n+\alpha}h_{*}\right)}{d\left(\Delta t\right)} = \\ \frac{d\lambda}{d\left(\Delta t\right)}\left({}^{n+\alpha}h_{*}\right) &+ \lambda\alpha\left(\left.\frac{\partial h_{*}}{\partial\sigma_{ij}}\right|_{n+1}\frac{d\left({}^{n+1}\sigma_{ij}\right)}{d\left(\Delta t\right)} + \left.\frac{\partial h_{*}}{\partial q_{*}}\right|_{n+1}\frac{d\left({}^{n+1}q_{*}\right)}{d\left(\Delta t\right)}\right)\end{aligned} \tag{104.30}$$

$$\frac{d\ \left({}^{n+1}F\right)}{d\left(\Delta t\right)} = \frac{\partial\left({}^{n+1}F\right)}{\partial\left({}^{n+1}\sigma_{ij}\right)}\frac{d\left({}^{n+1}\sigma_{ij}\right)}{d\left(\Delta t\right)} + \frac{\partial\left({}^{n+1}F\right)}{\partial\left({}^{n+1}q_{*}\right)}\frac{d\left({}^{n+1n+1}q_{*}\right)}{d\left(\Delta t\right)} = 0 \tag{104.31}$$

where ${}^{n+\alpha}m_{ij}$ and ${}^{n+\alpha}h_{*}$ are defined by the equations (104.20) and (104.21).

By taking Δt to the limit value, $\Delta t \rightarrow 0$, in the (104.28), (104.29), (104.30) and (104.31) and using the relations from (104.22) one finds:

---

[14] bearing in mind that values at t are constants and that only variables at $t + \Delta t$ are changing with respect to Δt.

$$\lim_{\Delta t \to 0} \frac{d\left({}^{n+1}\sigma_{ij}\right)}{d\left(\Delta t\right)} = E_{ijkl}\left(\frac{d\left({}^{n}\epsilon_{kl}\right)}{d\left(\Delta t\right)} - \frac{d\left({}^{n+1}\epsilon^{p}_{kl}\right)_{\Delta t=0}}{d\left(\Delta t\right)}\right) \tag{104.32}$$

$$\lim_{\Delta t \to 0} \frac{d\left({}^{n+1}\epsilon^{p}_{ij}\right)}{d\left(\Delta t\right)} = \frac{d\lambda}{d\left(\Delta t\right)}\left({}^{n}m_{ij}\right) \tag{104.33}$$

$$\lim_{\Delta t \to 0} \frac{d\left({}^{n+1}q_{*}\right)}{d\left(\Delta t\right)} = \frac{d\lambda}{d\left(\Delta t\right)}\left({}^{n}h_{*}\right) \tag{104.34}$$

$$\lim_{\Delta t \to 0} \frac{d\ \left({}^{n+1}F\right)}{d\left(\Delta t\right)} = \frac{\partial\left({}^{n}F\right)}{\partial \sigma_{ij}}\left(\lim_{\Delta t \to 0} \frac{d\left({}^{n+1}\sigma_{ij}\right)}{d\left(\Delta t\right)}\right) + \frac{\partial\left({}^{n}F\right)}{\partial q_{*}}\left(\lim_{\Delta t \to 0} \frac{d\left({}^{n+1}q_{*}\right)}{d\left(\Delta t\right)}\right) = 0 \tag{104.35}$$

In the previous equations it is quite clear that since $\Delta t = 0$, then equations (104.22) hold and since the variables ${}^{n}\sigma_{ij}$, ${}^{n}\epsilon^{p}_{ij}$ and ${}^{n}q_{*}$ are constant with respect to the change in Δt, the result follows readily, i.e. the Midpoint rule satisfies first order accuracy.

Second Order Accuracy To investigate second order accuracy of the algorithm given by (104.16), (104.17), (104.18) and (104.19) together with the constitutive equations given by (104.1), (104.2), (104.3) and (104.4) we shall proceed in the following manner. Second order accuracy actually means that the numerically integrated variables ${}^{n+1}\sigma_{ij}$, ${}^{n+1}\epsilon^{p}_{ij}$ and ${}^{n+1}q_{*}$ agree with their "exact" values $\sigma_{ij}(t + \Delta t)$, $\epsilon^{p}_{ij}(t + \Delta t)$ and $q_{*}(t + \Delta t)$ to within third order terms in the Taylor's expansion around the initial state ${}^{n}\sigma_{ij} = \sigma_{ij}(t)$, ${}^{n}\epsilon^{p}_{ij} = \epsilon^{p}_{ij}(t)$ and ${}^{n}q_{*} = q_{*}(t)$ in Δt. This verbal statement can be written in the following mathematical form:

$$
\begin{gathered}
\lim_{\Delta t\to 0}\frac{d^2\left({}^{n+1}\sigma_{ij}\right)}{d\left(\Delta t\right)^2} = \\
E_{ijkl}\left(\lim_{\Delta t\to 0}\frac{d^2\left({}^{n+1}\epsilon_{kl}\right)}{d\left(\Delta t\right)^2} - \lim_{\Delta t\to 0}\frac{d^2\left({}^{n}\epsilon^p_{kl}\right)}{d\left(\Delta t\right)^2}\right) = E_{ijkl}\left(\frac{d^2\left({}^{n}\epsilon_{kl}\right)}{d\left(\Delta t\right)^2} - \frac{d^2\left({}^{n}\epsilon^p_{kl}\right)}{d\left(\Delta t\right)^2}\right)
\end{gathered}
\tag{104.36}
$$

$$
\begin{gathered}
\lim_{\Delta t\to 0}\frac{d^2\left({}^{n+1}\epsilon^p_{ij}\right)}{d\left(\Delta t\right)^2} = \\
\frac{d^2\lambda}{d\left(\Delta t\right)^2}\lim_{\Delta t\to 0}\left({}^{n+1}m_{ij}\right) + \lim_{\Delta t\to 0}\frac{d\left({}^{n+1}\lambda\right)}{d\left(\Delta t\right)}\frac{d\left({}^{n+\alpha}m_{ij}\right)}{d\left(\Delta t\right)} = \\
\frac{d^2\lambda}{d\left(\Delta t\right)^2}\left({}^{n}m_{ij}\right) + \frac{d\left({}^{n}\lambda\right)}{d\left(\Delta t\right)}\frac{d\left({}^{n+\alpha}m_{ij}\right)}{d\left(\Delta t\right)} = \\
\frac{d^2\lambda}{d\left(\Delta t\right)^2}\left({}^{n}m_{ij}\right) + \frac{d\left({}^{n}\lambda\right)}{d\left(\Delta t\right)}\left(\frac{\partial m_{ij}}{\partial\sigma_{ij}\,|_n}\frac{d\left({}^{n}\sigma_{ij}\right)}{d\left(\Delta t\right)} + \frac{\partial m_{ij}}{\partial q_*\,|_n}\frac{d\left({}^{n}q_*\right)}{d\left(\Delta t\right)}\right)
\end{gathered}
\tag{104.37}
$$

$$
\begin{gathered}
\lim_{\Delta t\to 0}\frac{d^2\left({}^{n+1}q^p_*\right)}{d\left(\Delta t\right)^2} = \\
\frac{d^2\lambda}{d\left(\Delta t\right)^2}\lim_{\Delta t\to 0}\left({}^{n+1}h_*\right) + \lim_{\Delta t\to 0}\frac{d\left({}^{n+1}\lambda\right)}{d\Delta t}\frac{d\left({}^{n+1}h_*\right)}{d\left(\Delta t\right)} = \\
\frac{d^2\lambda}{d\left(\Delta t\right)^2}\left({}^{n}h_*\right) + \frac{d\left({}^{n+1}\lambda\right)}{d\Delta t}\frac{d\left({}^{n}h_*\right)}{d\left(\Delta t\right)} = \\
\frac{d^2\lambda}{d\left(\Delta t\right)^2}\left({}^{n}h_*\right) + \frac{d\left({}^{n+1}\lambda\right)}{d\Delta t}\left(\frac{\partial h_*}{\partial\sigma_{ij}\,|_n}\frac{d\left({}^{n}\sigma_{ij}\right)}{d\left(\Delta t\right)} + \frac{\partial h_*}{\partial q_*\,|_n}\frac{d\left({}^{n}q_*\right)}{d\left(\Delta t\right)}\right)
\end{gathered}
\tag{104.38}
$$

$$
\lim_{\Delta t\to 0}\frac{d^2\left(\lambda\right)}{d\left(\Delta t\right)^2} = \frac{d^2\left({}^{n}\lambda\right)}{d\left(\Delta t\right)^2}
\tag{104.39}
$$

and the plastic parameter $d^2\left({}^{n}\lambda\right)/d\left(\Delta t\right)^2$ is determined with the aid of the second order oscillatory satisfaction of the plastic consistency condition:

$$
\frac{d^2\ \left({}^{n}F\right)}{d\left(\Delta t\right)^2} = \left.\frac{dn_{ij}}{d\Delta t}\right|_n\frac{d\sigma_{ij}}{d\left(\Delta t\right)} + {}^{n}n_{ij}\left.\frac{d^2\left(\sigma_{ij}\right)}{d\left(\Delta t\right)^2}\right|_n + \left.\frac{d\left(\xi_*\right)}{d\Delta t}\right|_n\frac{d^n q_*}{d\left(\Delta t\right)} + {}^{n}\xi_*\frac{d^2\left({}^{n}q_*\right)}{d\left(\Delta t\right)^2} = 0
\tag{104.40}
$$

Now we can proceed by taking the second derivative of the equations (104.16), (104.17), (104.18) and (104.19) or use the already derived first derivatives from equations (104.28), (104.29), (104.30) and (104.31), and then differentiate them again so that we get:

$$\frac{d^2\left({}^{n+1}\sigma_{ij}\right)}{d\left(\Delta t\right)^2} = E_{ijkl}\left(\frac{d^2\left({}^{n+1}\epsilon_{kl}\right)}{d\left(\Delta t\right)^2} - \frac{d^2\left({}^{n+1}\epsilon^p_{kl}\right)}{d\left(\Delta t\right)^2}\right) \tag{104.41}$$

$$\begin{aligned}
\frac{d^2\left({}^{n+1}\epsilon^p_{ij}\right)}{d\left(\Delta t\right)^2} &= \\
\frac{d^2\lambda}{d\left(\Delta t\right)^2}\left({}^{n+\alpha}m_{ij}\right) &+ \\
2\frac{d\lambda}{d\left(\Delta t\right)}\alpha\left(\left.\frac{\partial m_{ij}}{\partial \sigma_{ij}}\right|_{n+1}\frac{d\left({}^{n+1}\sigma_{ij}\right)}{d\left(\Delta t\right)} + \left.\frac{\partial m_{ij}}{\partial q_*}\right|_{n+1}\frac{d\left({}^{n+1}q_*\right)}{d\left(\Delta t\right)}\right) &+ \\
\lambda\alpha\frac{d}{d\left(\Delta t\right)}\left(\left.\frac{\partial m_{ij}}{\partial \sigma_{ij}}\right|_{n+1}\frac{d\left({}^{n+1}\sigma_{ij}\right)}{d\left(\Delta t\right)} + \left.\frac{\partial m_{ij}}{\partial q_*}\right|_{n+1}\frac{d\left({}^{n+1}q_*\right)}{d\left(\Delta t\right)}\right) &
\end{aligned} \tag{104.42}$$

$$\begin{aligned}
\frac{d^2\left({}^{n+1}q^p_*\right)}{d\left(\Delta t\right)^2} &= \\
\frac{d^2\lambda}{d\left(\Delta t\right)^2}\left({}^{n+\alpha}h_*\right) &+ \\
2\frac{d\lambda}{d\left(\Delta t\right)}\alpha\left(\left.\frac{\partial h_*}{\partial \sigma_{ij}}\right|_{n+1}\frac{d\left({}^{n+1}\sigma_{ij}\right)}{d\left(\Delta t\right)} + \left.\frac{\partial h_*}{\partial q_*}\right|_{n+1}\frac{d\left({}^{n+1}q_*\right)}{d\left(\Delta t\right)}\right) &+ \\
\lambda\alpha\frac{d}{d\left(\Delta t\right)}\left(\left.\frac{\partial h_*}{\partial \sigma_{ij}}\right|_{n+1}\frac{d\left({}^{n+1}\sigma_{ij}\right)}{d\left(\Delta t\right)} + \left.\frac{\partial h_*}{\partial q_*}\right|_{n+1}\frac{d\left({}^{n+1}q_*\right)}{d\left(\Delta t\right)}\right) &
\end{aligned} \tag{104.43}$$

$$\begin{aligned}
\frac{d^2\left({}^{n+1}F\right)}{d\left(\Delta t\right)^2} &= \\
\frac{d\left({}^{n+1}n_{ij}\right)}{d\sigma_{ij}}\frac{d\left({}^{n+1}\sigma_{ij}\right)}{d\left(\Delta t\right)} + {}^{n+1}n_{ij}\frac{d^2\left({}^{n+1}\sigma_{ij}\right)}{d\left(\Delta t\right)^2} &+ \\
+\frac{d\left({}^{n+1}\xi_*\right)}{d\sigma_{ij}}\frac{d\left({}^{n+1}q_*\right)}{d\left(\Delta t\right)} + {}^{n+1}\xi\frac{d^2\left({}^{n+1}q_*\right)}{d\left(\Delta t\right)^2} &= 0
\end{aligned} \tag{104.44}$$

If we drive Δt to the limit, namely by taking $\lim_{\Delta t \to 0}$ and keeping in mind equations (104.22) and the assumed consistency of the algorithm[15] as given by the equations (104.23), (104.24), (104.25) and (104.26) one finds:

---

[15] actually the first order accuracy that is already proven.

$$\lim_{\Delta t \to 0} \frac{d^2 \left({}^{n+1}\sigma_{ij}\right)}{d\left(\Delta t\right)^2} = E_{ijkl} \left(\frac{d^2 \left({}^{n}\epsilon_{kl}\right)}{d\left(\Delta t\right)^2} - \lim_{\Delta t \to 0} \frac{d^2 \left({}^{n+1}\epsilon^{p}_{kl}\right)}{d\left(\Delta t\right)^2} \right) \tag{104.45}$$

$$\lim_{\Delta t \to 0} \frac{d^2 \left({}^{n+1}\epsilon^{p}_{ij}\right)}{d\left(\Delta t\right)^2} = \lim_{\Delta t \to 0} \frac{d^2 \left({}^{n+1}\lambda\right)}{d\left(\Delta t\right)^2} \left({}^{n+\alpha}m_{ij}\right) + 2 \frac{d\left({}^{n}\lambda\right)}{d\left(\Delta t\right)} \alpha \left(\left.\frac{\partial m_{ij}}{\partial \sigma_{ij}}\right|_{n} \frac{d\left({}^{n}\sigma_{ij}\right)}{d\left(\Delta t\right)} + \left.\frac{\partial m_{ij}}{\partial q_{*}}\right|_{n} \frac{d\left({}^{n}q_{*}\right)}{d\left(\Delta t\right)} \right) \tag{104.46}$$

$$\lim_{\Delta t \to 0} \frac{d^2 \left({}^{n+1}q^{p}_{*}\right)}{d\left(\Delta t\right)^2} = \lim_{\Delta t \to 0} \frac{d^2 \left({}^{n+1}\lambda\right)}{d\left(\Delta t\right)^2} \left({}^{n+\alpha}h_{*}\right) + 2 \frac{d\left({}^{n}\lambda\right)}{d\left(\Delta t\right)} \alpha \left(\left.\frac{\partial h_{*}}{\partial \sigma_{ij}}\right|_{n} \frac{d\left({}^{n}\sigma_{ij}\right)}{d\left(\Delta t\right)} + \left.\frac{\partial h_{*}}{\partial q_{*}}\right|_{n} \frac{d\left({}^{n}q_{*}\right)}{d\left(\Delta t\right)} \right) \tag{104.47}$$

$$\begin{aligned} &\lim_{\Delta t \to 0} \frac{d^2 \left({}^{n+1}F\right)}{d\left(\Delta t\right)^2} = \\ &\frac{d\left({}^{n}n_{ij}\right)}{d\sigma_{ij}} \frac{d\left({}^{n}\sigma_{ij}\right)}{d\left(\Delta t\right)} + {}^{n}n_{ij} \lim_{\Delta t \to 0} \frac{d^2 \left({}^{n+1}\sigma_{ij}\right)}{d\left(\Delta t\right)^2} + \\ &+ \frac{d\left({}^{n}\xi_{*}\right)}{d\sigma_{ij}} \frac{d\left({}^{n}q_{*}\right)}{d\left(\Delta t\right)} + {}^{n}\xi \lim_{\Delta t \to 0} \frac{d^2 \left({}^{n+1}q_{*}\right)}{d\left(\Delta t\right)^2} = 0 \end{aligned} \tag{104.48}$$

By comparing equations (104.45), (104.46), (104.47) and (104.48) with the second order accuracy condition stated in equations (104.36), (104.37), (104.38) and (104.39) it is quite clear that the second order accuracy is obtained iff[16] $\alpha = 1/2$!

The conclusion is that the Midpoint–rule algorithm is consistent[17] for all $\alpha \in [0, 1]$ and it is second order accurate for $\alpha = 1/2$. However, one should not forget that these results are obtained for the limiting case $\Delta t \to 0$, i.e. the strain increments are small and tend to zero.

---

[16] if and only if ($\iff$).

[17] first order accurate.

104.2.3.2 Numerical Stability Analysis

Numerical stability of an algorithm plays a central role in approximation theory for initial value problems. In fact, it can be stated that consistency and stability are necessary and sufficient conditions for convergence of an algorithm as the time step tends to zero. In the approach presented by Ortiz and Popov (1985) a new methodology is proposed by which the stability properties of an integration algorithm for elasto–plastic constitutive relations can be established. Our attention is confined to perfect plasticity and a smooth yield surface.

The purpose of the following stability analysis is to determine under what conditions a finite perturbation in the initial stresses is diluted by the algorithm. In other words:

$$d\left({}^{n+1}\sigma_{ij}^{(2)}, {}^{n+1}\sigma_{ij}^{(1)}\right) \leq d\left({}^{n}\sigma_{ij}^{(2)}, {}^{n}\sigma_{ij}^{(1)}\right) \tag{104.49}$$

where $d(\cdot, \cdot)$ is some suitable distance on the yield surface and ${}^{n+1}\sigma_{ij}^{(1)}$ and ${}^{n+1}\sigma_{ij}^{(2)}$ are two sets of updated stresses corresponding to arbitrary initial stress values ${}^{n}\sigma_{ij}^{(1)}$ and ${}^{n}\sigma_{ij}^{(2)}$, respectively, and all of the previous stress values are assumed to lie on the yield surface. Stability in the sense of equation (104.49) is referred to as large scale stability. It is shown in Helgason (1978)[18] that for nonlinear initial value problems defined on Banach manifolds, consistency and large scale stability with respect to a complete metric are sufficient for convergence.

The task of directly establishing estimates of the type expressed in (104.49) is rather difficult, and so despite the conceptual appeal of large scale stability, simplified solutions are sought. It should be recognized that attention can be restricted to infinitesimal perturbation in the initial conditions of the type ${}^{n}\sigma_{ij} \rightarrow {}^{n}\sigma_{ij} + d\left({}^{n}\sigma_{ij}\right)$. This simplification is founded on the fact that the dilution or attenuation, by the algorithm of infinitesimal perturbations:

$$\|d\ {}^{n+1}\sigma_{ij}\| \leq \|d\ {}^{n}\sigma_{ij}\| \tag{104.50}$$

with respect to some suitable norm $\|\cdot\|$, of small scale stability, implies large scale stability in the sense of equation (104.49).

Let the $\|\cdot\|$ norm, be the energy norm:

$$\|\sigma_{ij}\|^2 = \sigma_{ij} D_{ijkl} \sigma_{kl} \tag{104.51}$$

---

[18] the first Chapter of Helgason's book.

where D_{ijkl} is the elastic compliance tensor $(D_{ijkl} = E^{-1}_{ijkl})$, and let the distance $d(\cdot, \cdot)$ on the yield surface be defined as

$$d\left(\sigma_{ij}^{(1)}, \sigma_{ij}^{(2)}\right) = \inf_{\gamma} \int_{\gamma} \|\sigma_{ij}^{'}(s)\| ds \tag{104.52}$$

where the infinum is taken over all possible stress paths γ on the yield surface that are joining two stress states, namely $\sigma_{ij}^{(1)}$ and $\sigma_{ij}^{(2)}$. It can be found in Helgason (1978) that for a smooth yield surface, equation (104.52) defines the geodesic distance which endows the yield surface with a complete metric structure.

Suppose that we have any two initial states of stress ${}^{n}\sigma_{ij}^{(1)}$ and ${}^{n}\sigma_{ij}^{(2)}$ and let ${}^{n+1}\sigma_{ij}^{(1)}$ and ${}^{n+1}\sigma_{ij}^{(2)}$ be the corresponding updated values, respectively, and all the previous stress states are assumed to lie on the yield surface. Then, according to Helgason (1978), there exists a unique geodesic curve that joins ${}^{n}\sigma_{ij}^{(1)}$ and ${}^{n}\sigma_{ij}^{(2)}$ for which the infinum in equation (104.52) is attained. If γ_n is such a curve, then by definition:

$$d\left({}^{n}\sigma_{ij}^{(1)}, {}^{n}\sigma_{ij}^{(2)}\right) = \int_{\gamma_n} \|\sigma_{ij}^{'}(s)\| ds \tag{104.53}$$

Let the new curve γ_{n+1} be the transform of curve γ_n by the algorithm. By definition γ_{n+1} lies on the yield surface and joins two stress states ${}^{n+1}\sigma_{ij}^{(1)}$ and ${}^{n+1}\sigma_{ij}^{(2)}$. By the definition given in (104.52), it follows that:

$$d\left({}^{n+1}\sigma_{ij}^{(1)}, {}^{n+1}\sigma_{ij}^{(2)}\right) = \int_{\gamma_{n+1}} \|\sigma_{ij}^{'}(s)\| ds \tag{104.54}$$

Under the assumption of small scale stability of the algorithm one can write:

$$\|\sigma_{ij}^{'}(s_{n+1})\| ds = \|d\sigma_{ij}(s_{n+1})\| \leq \|d\sigma_{ij}(s_n)\| = \|\sigma_{ij}^{'}(s_n)\| ds \tag{104.55}$$

for every pair of corresponding points s_n and s_{n+1} on γ_n and γ_{n+1} respectively, so it follows:

$$\int_{\gamma_{n+1}} \|\sigma_{ij}^{'}(s_{n+1})\| ds \leq \int_{\gamma_n} \|\sigma_{ij}^{'}(s_n)\| ds \tag{104.56}$$

By combining equations (104.54), (104.55) and (104.56) it is concluded that:

$$d\left({}^{n+1}\sigma_{ij}^{(1)}, {}^{n+1}\sigma_{ij}^{(2)}\right) \leq d\left({}^{n}\sigma_{ij}^{(1)}, {}^{n}\sigma_{ij}^{(2)}\right) \tag{104.57}$$

which proves large scale stability. The main conclusion of the above argument may be stated as follows: small scale stability in the energy norm is equivalent to large scale stability in the associated geodesic distance.

The previous result is of practical importance, since it shows that the stability analysis for the integration algorithm in elasto–plasticity can be carried out by the assessment of small scale stability. The small scale stability analysis of the Generalized Midpoint rule is necessary to determine how the algorithm propagates infinitesimal perturbations in the initial conditions. By differentiating equations (104.16), (104.17), (104.18) and (104.19)and considering that we are dealing with perfectly plastic case here so that $\left({}^{n+1}q_*\right) = \left({}^{n}q_*\right)$ = constants, it follows:

$$d\left({}^{n+1}\sigma_{ij}\right) = -E_{ijkl}\, d\left({}^{n+1}\epsilon^{p}_{kl}\right) \tag{104.58}$$

$$d\left({}^{n}\sigma_{ij}\right) = -E_{ijkl}\, d\left({}^{n}\epsilon^{p}_{kl}\right) \tag{104.59}$$

$$d\left({}^{n+1}\epsilon^{p}_{ij}\right) - d\left({}^{n}\epsilon^{p}_{ij}\right) = d\,\lambda\ \left({}^{n+\alpha}m_{ij}\right) + \lambda\, d\left({}^{n+\alpha}m_{ij}\right) \tag{104.60}$$

$$d\ \left({}^{n+1}F\right) = \left.\frac{\partial F}{\partial \sigma_{ij}}\right|_{n+1} d\left({}^{n+1}\sigma_{ij}\right) = {}^{n+1}n_{ij}\, d\left({}^{n+1}\sigma_{ij}\right) = 0 \tag{104.61}$$

Let us now examine the shape of $d\left({}^{n+\alpha}m_{ij}\right)$ having in mind the original definition[19] given in equation (104.20):

$${}^{n+\alpha}m_{ij} = m_{ij}\left((1-\alpha)\ {}^{n}\sigma_{ij} + \alpha\ \left({}^{n+1}\sigma_{ij}\right), (1-\alpha)\ {}^{n}q_* + \alpha\ \left({}^{n+1}q_*\right)\right)$$

and the differential of the previous equation is:

$$d\left({}^{n+\alpha}m_{ij}\right) = (1-\alpha)\left.\frac{\partial m_{ij}}{\partial \sigma_{kl}}\right|_{n+\alpha} d\left({}^{n}\sigma_{kl}\right) + \alpha\ \left.\frac{\partial m_{ij}}{\partial \sigma_{kl}}\right|_{n+\alpha} d\left({}^{n+1}\sigma_{kl}\right)$$

To ease writing let us introduce the following fourth order tensor:

---

[19] the remark about restraining analysis to perfectly plastic case still holds, so that $\left({}^{n+1}q_*\right)$ and $\left({}^{n}q_*\right)$ are constant.

$$M_{ijkl} = \frac{\partial m_{ij}}{\partial \sigma_{kl}}$$

The equation (104.60) now reads:

$$\begin{aligned} & d\left({}^{n+1}\epsilon^{p}_{ij}\right) - d\left({}^{n}\epsilon^{p}_{ij}\right) = \\ & d\lambda\ \left({}^{n+\alpha}m_{ij}\right) + \lambda\left((1-\alpha)\left({}^{n+\alpha}M_{ijkl}\right)\ d\left({}^{n}\sigma_{kl}\right) + \alpha\ \left({}^{n+\alpha}M_{ijkl}\right)\ d\left({}^{n+1}\sigma_{kl}\right)\right) \end{aligned} \tag{104.62}$$

By using equations (104.58) and (104.59) and knowing that $E^{-1}_{ijkl} = D_{ijkl}$ one can write:

$$d\left({}^{n+1}\epsilon^{p}_{ij}\right) = -D_{ijkl}\ d\left({}^{n+1}\sigma_{kl}\right)$$

$$d\left({}^{n}\epsilon^{p}_{ij}\right) = -D_{ijkl}\ d\left({}^{n}\sigma_{kl}\right)$$

so that the equation (104.62) now reads:

$$\begin{aligned} & -D_{ijkl}\ d\left({}^{n+1}\sigma_{kl}\right) + D_{ijkl}\ d\left({}^{n}\sigma_{kl}\right) = \\ & d\lambda\ \left({}^{n+\alpha}m_{ij}\right) + \lambda\left((1-\alpha)\left({}^{n+\alpha}M_{ijkl}\right) d\left({}^{n}\sigma_{kl}\right) + \alpha\ \left({}^{n+\alpha}M_{ijkl}\right) d\left({}^{n+1}\sigma_{kl}\right)\right) \end{aligned}$$

Now we are proceeding by solving the previous equation for $d\left({}^{n+1}\sigma_{kl}\right)$:

$$\begin{aligned} & \left(D_{ijkl} + \lambda\ \alpha\ \left({}^{n+\alpha}M_{ijkl}\right)\right)\ d\left({}^{n+1}\sigma_{kl}\right) = \\ & \left(D_{ijkl} - \lambda\ (1-\alpha)\left({}^{n+\alpha}M_{ijkl}\right)\right) d\left({}^{n}\sigma_{kl}\right) - d\lambda\ \left({}^{n+\alpha}m_{ij}\right) \end{aligned}$$

and by denoting :

$$\Psi_{ijkl} = D_{ijkl} - \lambda\ (1-\alpha)\left({}^{n+\alpha}M_{ijkl}\right)$$

$$\Gamma_{ijkl} = D_{ijkl} + \lambda\ \alpha\left({}^{n+\alpha}M_{ijkl}\right)$$

it follows:

$$d\left({}^{n+1}\sigma_{kl}\right) = \Gamma^{-1}_{ijkl}\left(\Psi_{ijkl}\, d\left({}^{n}\sigma_{kl}\right) - d\lambda\ \left({}^{n+\alpha}m_{ij}\right)\right) \tag{104.63}$$

Then by inserting the solution for $d\left({}^{n+1}\sigma_{kl}\right)$ in the consistency condition (104.61):

$$d\ \left({}^{n+1}F\right) = {}^{n+1}n_{kl}\ d\left({}^{n+1}\sigma_{kl}\right) = 0$$

one gets:

$$\begin{gathered} d\ \left({}^{n+1}F\right) = \\ {}^{n+1}n_{kl}\ \Gamma^{-1}_{ijkl}\left(\Psi_{ijkl}\, d\left({}^{n}\sigma_{kl}\right) - d\lambda\ \left({}^{n+\alpha}m_{ij}\right)\right) = 0 \end{gathered} \tag{104.64}$$

then if we solve for $d\lambda$:

$$d\lambda\ {}^{n+\alpha}m_{ij}{}^{n+1}n_{kl}\ \Gamma^{-1}_{ijkl} = {}^{n+1}n_{kl}\ \Gamma^{-1}_{ijkl}\Psi_{ijkl}\, d\left({}^{n}\sigma_{kl}\right) \tag{104.65}$$

or[20]:

$$d\lambda = \frac{{}^{n+1}n_{rs}\ \Gamma^{-1}_{pqrs}\Psi_{pqrs}\ d\left({}^{n}\sigma_{rs}\right)}{\left({}^{n+\alpha}m_{pq}\right)\left({}^{n+1}n_{rs}\right)\Gamma^{-1}_{pqrs}} \tag{104.66}$$

then by using the solution for $d\left({}^{n+1}\sigma_{kl}\right)$ from (104.63) and the solution for $d\lambda$ from (104.66) one can find:

$$d\left({}^{n+1}\sigma_{kl}\right) = \Gamma^{-1}_{ijkl}\Psi_{ijkl}\, d\left({}^{n}\sigma_{kl}\right) - \Gamma^{-1}_{pqrs}\ \Psi_{pqrs}\frac{{}^{n+1}n_{rs}\ \Gamma^{-1}_{ijkl}\ \left({}^{n+\alpha}m_{ij}\right)}{{}^{n+\alpha}m_{pq}{}^{n+1}n_{rs}\ \Gamma^{-1}_{pqrs}} d\left({}^{n}\sigma_{rs}\right) \tag{104.67}$$

$$d\left({}^{n+1}\sigma_{kl}\right) = \Gamma^{-1}_{ijkl}\Psi_{ijkl}\left(\delta_{ks}\delta_{rl} - \frac{{}^{n+1}n_{rs}\ \Gamma^{-1}_{ijkl}\ \left({}^{n+\alpha}m_{ij}\right)}{{}^{n+\alpha}m_{pq}{}^{n+1}n_{rs}\ \Gamma^{-1}_{pqrs}}\right) d\left({}^{n}\sigma_{rs}\right) \tag{104.68}$$

to ease the writing we can define the following notation:

---

[20] where the change in dummy indices is possible because $d\lambda$ is scalar.

$$\Phi_{klrs} = \delta_{ks}\delta_{rl} - \frac{{}^{n+1}n_{rs}\ \Gamma^{-1}_{ijkl}\ \left({}^{n+\alpha}m_{ij}\right)}{{}^{n+\alpha}m_{pq}{}^{n+1}n_{rs}\ \Gamma^{-1}_{pqrs}} \tag{104.69}$$

so that the equation (104.68) now reads:

$$d\left({}^{n+1}\sigma_{kl}\right) = \Gamma^{-1}_{ijkl}\Psi_{ijkl}\ \Phi_{klrs}\ d\left({}^{n}\sigma_{rs}\right) \tag{104.70}$$

In order to derive the estimate of the type (104.50) from (104.70) we shall proceed in the following way. The norm of a tensor is defined as:

$$\|A_{ijkl}\| = \sup_{\sigma} \frac{\|A_{ijkl}\sigma_{kl}\|}{\|\sigma_{kl}\|} \tag{104.71}$$

If we take the norm of (104.70), while recalling the inequalities:

$$\|A_{ijkl}\sigma_{kl}\| \leq \|A_{ijkl}\|\ \|\sigma_{kl}\| \quad ; \quad \|A_{ijkl}B_{ijkl}\| \leq \|A_{ijkl}\|\ \|B_{ijkl}\| \tag{104.72}$$

it follows:

$$\|d\left({}^{n+1}\sigma_{kl}\right)\| = \|\Gamma^{-1}_{ijkl}\Psi_{ijkl}\ \Phi_{klrs}\ d\left({}^{n}\sigma_{rs}\right)\| \tag{104.73}$$

then by using equations (104.72), we are able to write:

$$\|d\left({}^{n+1}\sigma_{kl}\right)\| \leq \|\Gamma^{-1}_{ijkl}\ \Psi_{ijkl}\|\ \|\Phi_{klrs}\|\ \|d\left({}^{n}\sigma_{rs}\right)\| \tag{104.74}$$

Considering the norm of $\|\Phi_{klrs}\|$ it should be noted that Φ_{klrs} defines a projection along the direction of $\Gamma^{-1}_{ijkl}\ {}^{n+\alpha}m_{ij}$ onto the hyperplane that is orthogonal to ${}^{n+1}n_{rs}$, so that the following properties hold:

$$\left(\Phi_{klrs}\right)\left(\Gamma^{-1}_{ijkl}\ {}^{n+\alpha}m_{ij}\right) = \emptyset \tag{104.75}$$

$$\left(\Phi_{klrs}\right)\left(\sigma_{rs}\right) = \sigma_{rs} \tag{104.76}$$

for every σ_{rs} that is orthogonal to ${}^{n+1}n_{rs}$. From these properties and the definition in equation (104.71) it follows that:

$$\|\Phi_{klrs}\| \equiv 1 \tag{104.77}$$

In what follows it is assumed that the fourth order tensor field

$$M_{ijkl} = \partial m_{ij} / \partial \sigma_{kl}$$

is symmetric and positive definite everywhere on the yield surface. The assumption is valid, if the flow direction m_{ij} is derived from the convex potential function, which is a rather common feature among yield criteria. It is now clear that :

$$\|\Gamma^{-1}_{ijkl}\ \Psi_{ijkl}\| = \left| \frac{{}^{max}\gamma_{ij}\ \Psi_{ijkl}\ {}^{max}\gamma_{kl}}{{}^{max}\gamma_{ij}\ \Gamma_{ijkl}\ {}^{max}\gamma_{kl}} \right| \tag{104.78}$$

where ${}^{max}\gamma_{ij}$ is the eigentensor corresponding to the maximum eigenvalue of the eigenproblem:

$$\left(\Psi_{ijkl} - \mu\ \Gamma_{ijkl}\right) \gamma_{kl} = 0 \tag{104.79}$$

which is normalized to satisfy:

$$\|{}^{max}\gamma_{ij}\|\|{}^{max}\gamma_{ij}\| = {}^{max}\gamma_{ij}\ D_{ijkl}\ {}^{max}\gamma_{kl} = 1 \tag{104.80}$$

If we denote:

$${}^{n+\alpha}\beta = {}^{max}\gamma_{ij}\ {}^{n+\alpha}M_{ijkl}\ {}^{max}\gamma_{kl} \tag{104.81}$$

as the maximum eigenvalue of the fourth order tensor ${}^{n+\alpha}M_{ijkl}$ and that value is a positive real number[21], then from equations (104.78), (104.80), (104.81) and from the definition[22] of Ψ_{ijkl} and Γ_{ijkl}, it follows:

$$\|\Gamma^{-1}_{ijkl}\ \Psi_{ijkl}\| = \left| \frac{1 - (1-\alpha)\ \lambda\ \left({}^{n+\alpha}\beta\right)}{1 + \alpha\ \lambda\ \left({}^{n+\alpha}\beta\right)} \right| \tag{104.82}$$

which, when inserted in the equation (104.74) yields:

---

[21] because ${}^{n+\alpha}M_{ijkl}$ is derived from a convex potential function.

[22] $\Psi_{ijkl} = D_{ijkl} - \lambda\ (1-\alpha)\left({}^{n+\alpha}M_{ijkl}\right)$ and $\Gamma_{ijkl} = D_{ijkl} + \lambda\ \alpha\left({}^{n+\alpha}M_{ijkl}\right)$

$$\|d\left({}^{n+1}\sigma_{kl}\right)\| \leq \left|\frac{1-(1-\alpha)\ \lambda\ \left({}^{n+\alpha}\beta\right)}{1+\alpha\ \lambda\ \left({}^{n+\alpha}\beta\right)}\right|\ \|d\left({}^{n}\sigma_{rs}\right)\| \tag{104.83}$$

Since it is said that ${}^{n+\alpha}\beta$ is a positive real number it follows that:

$$\left|\frac{1-(1-\alpha)\ \lambda\ \left({}^{n+\alpha}\beta\right)}{1+\alpha\ \lambda\ \left({}^{n+\alpha}\beta\right)}\right| \leq \left|\frac{1-\alpha}{\alpha}\right| \frac{{}^{n+\alpha}\beta}{{}^{n+\alpha}\beta} = \left|\frac{1-\alpha}{\alpha}\right| \tag{104.84}$$

and $\alpha \in [0,1]$. The new form of equation (104.83) is now:

$$\|d\left({}^{n+1}\sigma_{kl}\right)\| \leq \left|\frac{1-\alpha}{\alpha}\right|\ \|d\left({}^{n}\sigma_{rs}\right)\| \tag{104.85}$$

which in conjunction with the requirement for unconditional stability[23] yields:

$$\left|\frac{1-\alpha}{\alpha}\right| \leq 1 \tag{104.86}$$

and so it is necessary that:

$$\alpha \geq {}^{min}\alpha = \frac{1}{2} \tag{104.87}$$

The conclusion is that the Generalized Midpoint rule is unconditionally stable for $\alpha \geq 1/2$. In the case when $\alpha < 1/2$ the Generalized Midpoint rule is only conditionally stable. To obtain a stability condition for $\alpha \leq 1/2$ one has to go back to equation (104.83), and we conclude that:

$$\left|\frac{1-(1-\alpha)\ \lambda\ \left({}^{n+\alpha}\beta\right)}{1+\alpha\ \lambda\ \left({}^{n+\alpha}\beta\right)}\right| \leq 1 \quad \Rightarrow \quad \lambda \leq \frac{2}{{}^{max}\beta\ (1-2\alpha)} \quad \text{for} \quad \alpha \leq \frac{1}{2} \tag{104.88}$$

and when $\alpha = 1/2$, then ${}^{critical}\lambda \rightarrow \infty$, and thus the unconditional stability is recovered.

---

[23] that is $\|d\ {}^{n+1}\sigma_{ij}\| \leq \|d\ {}^{n}\sigma_{ij}\|$

104.2.4 Crossing the Yield Surface

Midpoint rule algorithms in computational elasto–plasticity require[24] the evaluation of the intersection[25] stress. Despite the appeal of the closed form solution, as found in Bićanić (1989), and numerical iterative procedures as found in Marques (1984) and Nayak and Zienkiewicz (1972), for some yield criteria[26] the solution is not that simple to find. Special problems arises, even with the numerical iterative methods in the area of a apex. The apex area problems are connected to the derivatives of yield a function.

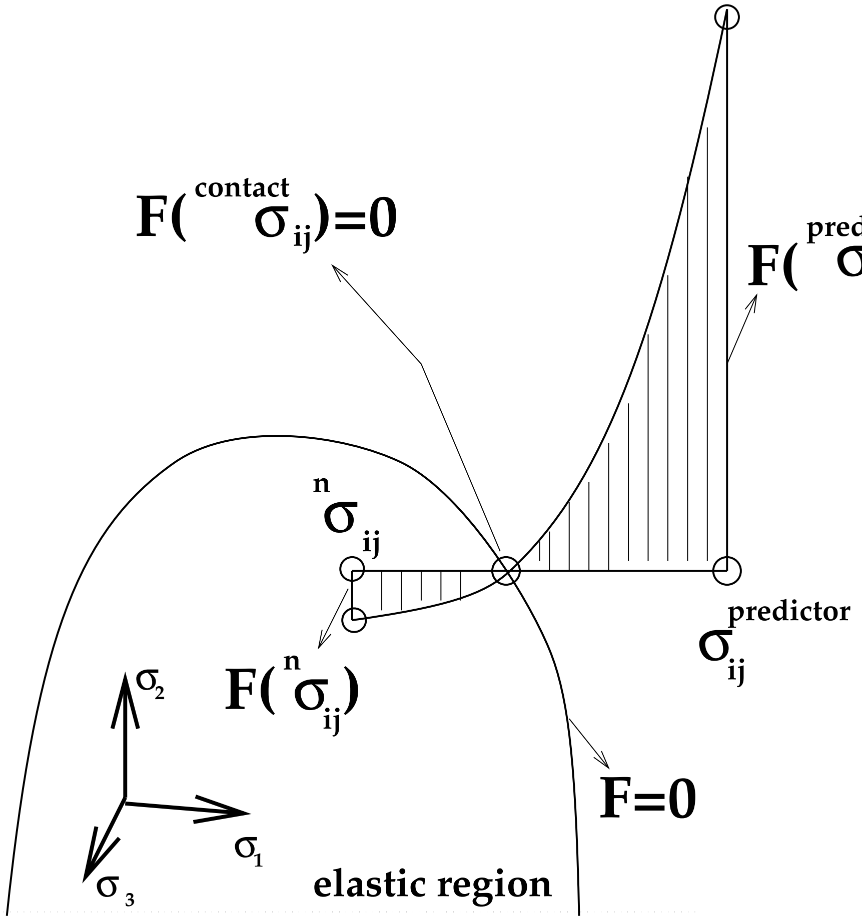


Figure 104.2: The pictorial representation of the intersection point problem in computational elasto–plasticity: which must be resolved for the Forward and Midpoint schemes

Having in mind the before mentioned problems, a different numerical scheme, that does not need derivatives, was sought for solving this problem. One possible solution was found in Press et al. (1988b)

---

[24] except for the fully implicit Backward Euler algorithm.

[25] contact, penetration point, i.e the point along the stress path where $F = 0$ or the point where stress state crosses from the elastic to the plastic region.

[26] namely for the MRS-Lade elasto–plastic model.

in the form of an excellent algorithm that combines root bracketing, bisection, and inverse quadratic interpolation to converge from a neighborhood of a zero crossing. The algorithm was developed in the 1960s by van Wijngaarden, Dekker and others at the Mathematical Center in Amsterdam. The algorithm was later improved by Brent, and so it is better known as Brent's method. The method is guaranteed to converge, so long as the function[27] can be evaluated within the initial interval known to contain a root. While the other iterative methods that do not require derivatives[28] assume approximately linear behavior between two prior estimates, inverse quadratic interpolation uses three prior points to fit an inverse quadratic function[29], whose value at $y = 0$ is taken as the next estimate of the root x. Lagrange's classical formula for interpolating the polynomial of degree $N-1$ through N points $y_1 = f(x_1)$, $y_2 = f(x_2)$, $\ldots y_3 = f(x_3)$ is given by:

$$\begin{aligned} P(x) = & \frac{(x-x_2)(x-x_3)\cdots(x-x_N)}{(x_1-x_2)(x_1-x_3)\cdots(x_1-x_N)}\, y_1 + \\ + & \frac{(x-x_1)(x-x_3)\cdots(x-x_N)}{(x_2-x_1)(x_2-x_3)\cdots(x_2-x_N)}\, y_2 + \cdots \\ \cdots + & \frac{(x-x_1)(x-x_2)\cdots(x-x_N)}{(x_N-x_1)(x_N-x_3)\cdots(x_N-x_{N-1})}\, y_N \end{aligned} \tag{104.89}$$

If the three point pairs are $[a, f(a)], [b, f(b)], [c, f(c)]$, then the interpolating formula (104.89) yields:

$$\begin{aligned} x = & \frac{(y-f(a))(y-f(b))}{(f(c)-f(a))(f(c)-f(b))}\, c + \\ + & \frac{(y-f(b))(y-f(c))}{(f(a)-f(b))(f(a)-f(c))}\, a + \\ + & \frac{(y-f(c))(y-f(a))}{(f(b)-f(a))(f(b)-f(a))}\, b \end{aligned} \tag{104.90}$$

By setting $y = 0$, we obtain a result for the next root estimate, which can be written as:

$$x = b + \frac{\frac{f(b)}{f(a)}\left(\frac{f(a)}{f(c)}\left(\frac{f(b)}{f(c)} - \frac{f(a)}{f(c)}\right)(c-b) - \left(1 - \frac{f(b)}{f(c)}\right)(b-a)\right)}{\left(\frac{f(a)}{f(c)} - 1\right)\left(\frac{f(b)}{f(c)} - 1\right)\left(\frac{f(b)}{f(a)} - 1\right)} \tag{104.91}$$

In practice b is the current best estimate of the root and the term:

---

[27] in our case yield function $F(\sigma_{ij})$.

[28] false position and secant method.

[29] x as a quadratic function of y.

$$\frac{\frac{f(b)}{f(a)}\left(\frac{f(a)}{f(c)}\left(\frac{f(b)}{f(c)}-\frac{f(a)}{f(c)}\right)(c-b)-\left(1-\frac{f(b)}{f(c)}\right)(b-a)\right)}{\left(\frac{f(a)}{f(c)}-1\right)\left(\frac{f(b)}{f(c)}-1\right)\left(\frac{f(b)}{f(a)}-1\right)}$$

is a correction. Quadratic methods[30] work well only when the function behaves smoothly. However, they run serious risk of giving bad estimates of the next root or causing floating point overflows, if divided by a small number

$$\left(\frac{f\left(a\right)}{f\left(c\right)}-1\right)\left(\frac{f\left(b\right)}{f\left(c\right)}-1\right)\left(\frac{f\left(b\right)}{f\left(a\right)}-1\right)\approx 0$$

Brent's method prevents against this problem by maintaining brackets on the root and checking where the interpolation would land before carrying out the division. When the correction of type (104.92) would not land within bounds, or when the bounds are not collapsing rapidly enough, the algorithm takes a bisection step. Thus, Brent's method combines the sureness of bisection with the speed of a higher order method when appropriate.

104.2.5 Singularities in the Yield Surface

104.2.5.1 Corner Problem

Some yield criteria are defined with more that one yield surface[31]. We will restrict our attention to a two–surface yield criterion[32]. Koiter has shown in Koiter (1960) and Koiter (1953) that in the case when two yield surfaces are active, the plastic strain rate from equation (104.3) can be derived as follows:

$$d\epsilon_{ij}^{p} = d\lambda_{cone}\ {}^{cone}m_{ij}\left(\sigma_{ij}, q_{*}\right) + d\lambda_{cap}\ {}^{cap}m_{ij}\left(\sigma_{ij}, q_{*}\right) \qquad (104.92)$$

where ${}^{cone}m_{ij}\left(\sigma_{ij}, q_{*}\right)$ and ${}^{cap}m_{ij}\left(\sigma_{ij}, q_{*}\right)$ are normals to the potential functions at a corner, which belongs to the yield functions that are active, i.e. F_{cone} and F_{cap}. We now observe that we have two non–negative plastic multipliers $d\lambda_{cone}$ and $d\lambda_{cap}$ instead of one. We must require that at the end of the loading step[33], neither of the two yield functions is violated. These multipliers $d\lambda_{cone}$ and $d\lambda_{cap}$ can be determined from the conditions:

---

[30] Newton's method for example.

[31] for example MRS-Lade yield criterion has two surfaces.

[32] having in mind MRS-Lade cone-cap yield criterion.

[33] after stress correction, i.e. return to the yield surface(s).

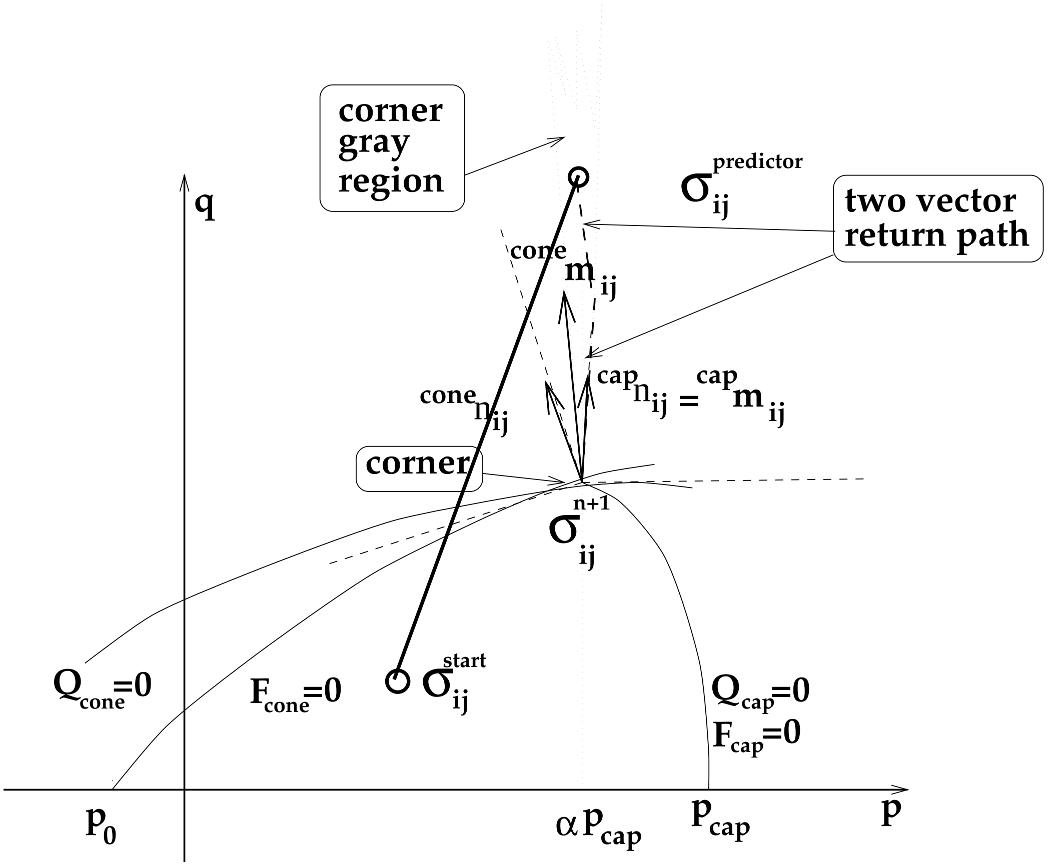


Figure 104.3: Pictorial representation of the corner point problem in computational elasto–plasticity: Yield surfaces with singular points

$$F_{cone}\left({}^{n+1}\sigma_{ij}, {}^{n+1}q_*\right) = 0 \tag{104.93}$$

$$F_{cap}\left({}^{n+1}\sigma_{ij}, {}^{n+1}q_*\right) = 0 \tag{104.94}$$

noting that by virtue of equation (104.92) we have at the corner singular point:

$${}^{n+1}\sigma_{ij} = {}^{pred}\sigma_{ij} - d\lambda_{cone}\, E_{ijkl}\, {}^{cone}m_{kl} - d\lambda_{cap}\, E_{ijkl}\, {}^{cap}m_{kl} \tag{104.95}$$

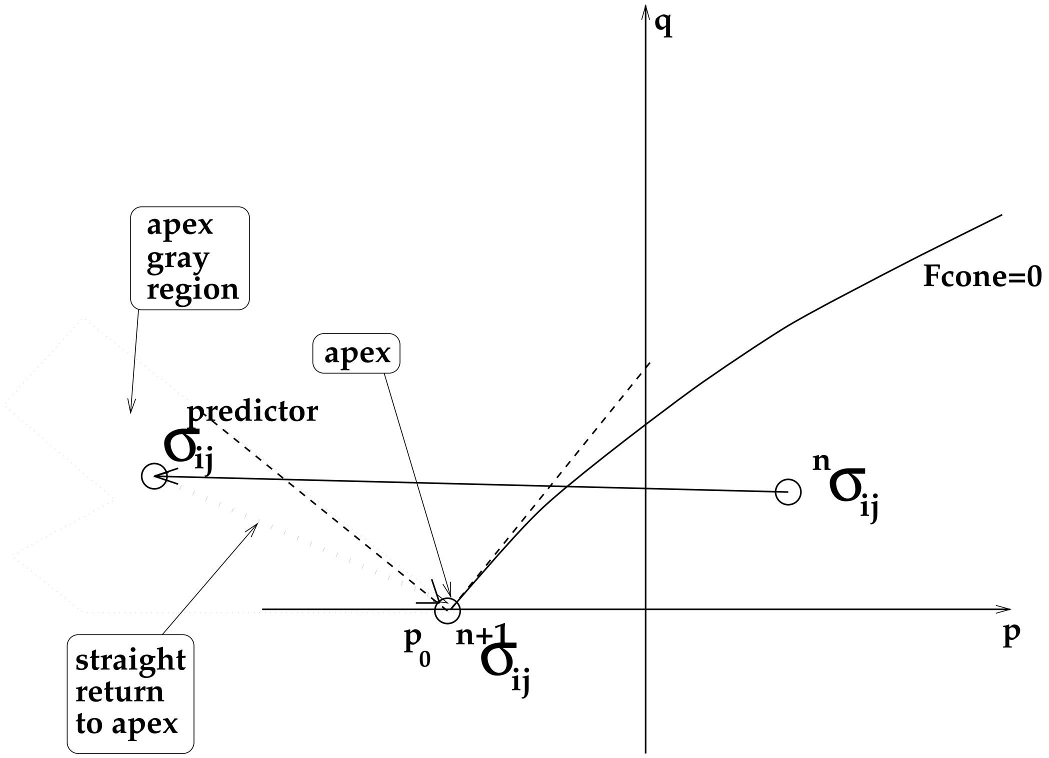


Figure 104.4: The pictorial representation of the apex point problem in computational elasto–plasticity: Yield surfaces with singular points

104.2.5.2 Apex Problem

The apex problem, as depicted in Figure (104.4) is solved in an empirical fashion. Rather than facing the complexity of solving a complex differential geometry problem[34] the stress point that is situated in the gray apex region is immediately returned to the apex point.

In the case when the hardening rule for the cone portion has developed to the stage that it affects the size of that cone portion of the yield criterion and not the position of intersection with the hydrostatic axis, then all stress returns from any part of apex gray region will be to the apex point itself. This strategy was used by Crisfield (1987). Nevertheless, the problem of integrating the rate equations in the apex gray region is readily solvable for the piecewise flat yield criteria[35] by using Koiter's conditions as found in Koiter (1960) and Koiter (1953). The apex problem for yield criteria that are smooth and differentiable everywhere except at the apex point, is solvable by means of differential geometry. Further

---

[34] using Koiter's work described in Koiter (1960) and Koiter (1953) and the fact that the sum $d\epsilon^p_{ij} = \sum_k d\lambda_k \left(\partial F_k / \partial\sigma_{ij}\right)$ can be transformed into the integral equation $d\epsilon^p_{ij} = \int d\lambda \left(\partial F / \partial\sigma_{ij}\right)\big|_{aroundapex}$ where the integration should be carried out infinitesimally close to, but in the vicinity of the apex point.

[35] Mohr - Coulomb for example.

work is needed for solving the problem, when the yield surface is not piecewise flat in the apex vicinity.

104.2.5.3 Influence Regions in Meridian Plane

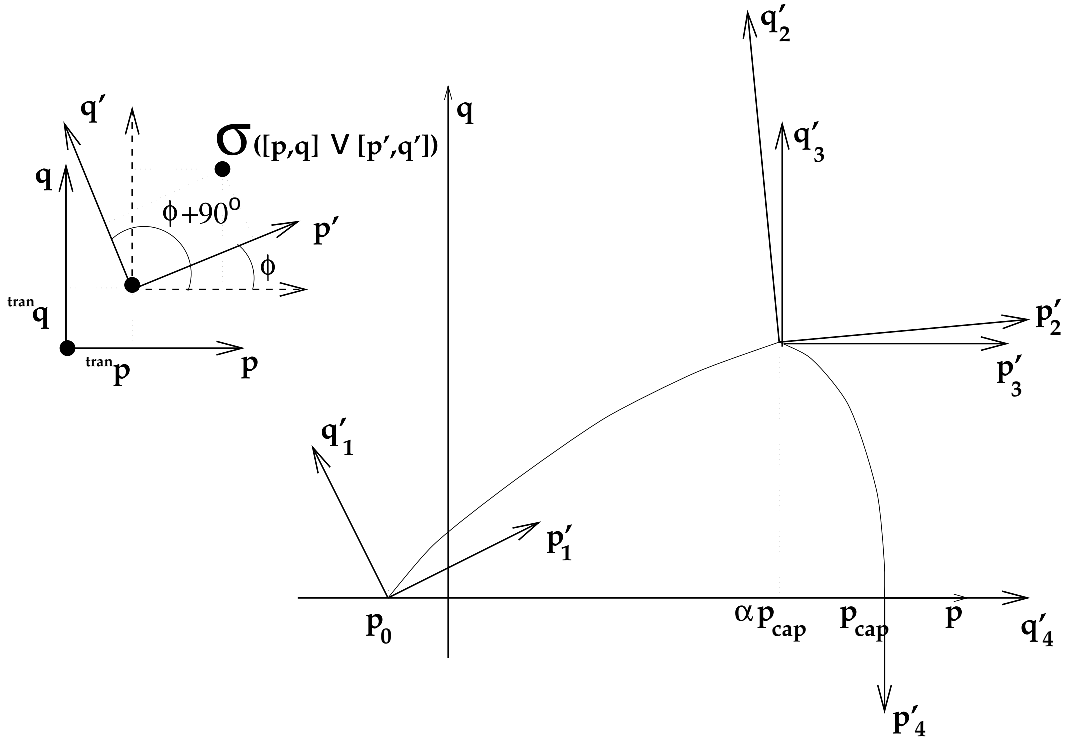


Figure 104.5: Influence regions in the meridian plane for the cone/cap surface of the MRS-Lade material model.

In order to define which surface is active and which is not for the current state of stress, a simple two dimensional analysis will be conducted. The fortunate fact for the MRS-Lade material model is that such an analysis can be conducted in the $p - q$ meridian plane, only, i.e. the value for θ can be "frozen". The concept is to calculate the stress invariants p, q and θ for the current state of stress[36], calculate the position of the apex and corner points in $p - q$ space for given the θ, calculate the two dimensional gradients at these points, perform linear transformation of the current stress state[37] to the new coordinate systems, and then check for the values of ${p'}_i$, $i = 1, 2, 3, 4$, where ${p'}_i$ is the transformed p_i axis.

The angle ψ is defined as the angle between the p axis and the tangent to the potential function.

The gradients to the cone portion of the potential surface are defined as:

---

[36] by using equations (104.138) and (104.139) as defined in section (104.4.4).

[37] now in p, q and θ space.

$$\frac{\partial Q_{cone}}{\partial p} = -n \; \eta_{cone}$$

$$\frac{\partial Q_{cone}}{\partial q} = g(\theta) \left(1 + \frac{q}{q_a}\right)^m + \frac{g(\theta) \, m \, q \left(1 + \frac{q}{q_a}\right)^{-1+m}}{q_a}$$

The gradients of the cap portion of the yield/potential surface is defined as:

$$\frac{\partial Q_{cap}}{\partial p} = \frac{2 \; (p - p_m)}{p_r^2}$$

$$\frac{\partial Q_{cap}}{\partial q} = \frac{2 \, g(\theta)^2 \, q \left(1 + \frac{q}{q_a}\right)^{2\,m}}{{f_r}^2} + \frac{2 \, g(\theta)^2 \, m \, q^2 \left(1 + \frac{q}{q_a}\right)^{-1+2\,m}}{{f_r}^2 \, q_a}$$

The vector of gradients in $p-q$ space is defined as:

$$\begin{bmatrix} \frac{\partial Q}{\partial p} \\ \frac{\partial Q}{\partial q} \end{bmatrix} \tag{104.96}$$

and the angle ϕ is calculated as:

$$\phi = \arctan \left(\frac{\left(\frac{\partial Q}{\partial p}\right)}{\left(\frac{\partial Q}{\partial q}\right)} \right) - 90^\circ \tag{104.97}$$

Care must be exercised with regard to which potential function is to be used in angle calculations. It should be mentioned that for the cap portion, the angle at the corner is $\phi = 0^\circ$, while at the tip of the cap, the angle is $\phi = -90^\circ$. If a new definition, as found in Ferrer (1992), is used for the cone potential function, where n is variable and $n \rightarrow 0$ as $p \rightarrow \alpha p_{cap}$, then the corner gray region is empty.

The linear transformation[38] between coordinate systems $p' - q'$ and $p - q$ is defined as:

$$\begin{bmatrix} p' \\ q' \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} p - {}^{tran}p \\ q - {}^{tran}q \end{bmatrix} \tag{104.98}$$

and by using that linear transformation, one can check the region where our current stress state, in p, q and θ space, belongs. Figure (104.5) depicts the transformation scheme and the new coordinate systems at three important points[39].

---

[38] translation and rotation.

[39] at the apex point, corner point and the cap tip point.

104.3 A Forward Euler (Explicit) Algorithm

The explicit algorithm (Forward Euler) is based on using the starting point (the state stress σ_{ij}^{n} and internal variable space q_*^n on the yield surface) for finding all the relevant derivatives and variables.

The Explicit algorithm can be derived by starting from a first order Taylor expansion about starting point (σ_{ij}^{n}, q_*^n):

$$\begin{aligned} F^{new} &= F^{old} + \left.\frac{\partial F}{\partial \sigma_{mn}}\right|_n d\left({}^{n}\sigma_{mn}\right) + \left.\frac{\partial F}{\partial q_*}\right|_n dq_* = \\ &= {}^{n}n_{mn}\, d\sigma_{mn} + \xi_* h_* d\lambda = 0 \end{aligned} \tag{104.99}$$

From the differential form of equation (104.16) it follows:

$$d\left({}^{fE}\sigma_{mn}\right) = E_{mnpq}\left(d\left(\epsilon_{pq}\right) - d\left(\epsilon_{pq}^{p}\right)\right) =$$
$$= E_{mnpq} d\left(\epsilon_{pq}\right) - E_{mnpq}\; d\left(\epsilon_{pq}^{p}\right) = E_{mnpq} d\left(\epsilon_{pq}\right) - E_{mnpq}\; d\lambda\left({}^{cross}m_{pq}\right)$$

so that equation (104.99) becomes:

$${}^{n}n_{mn}\; E_{mnpq}\; d\epsilon_{pq} - {}^{n}n_{mn}\; E_{mnpq}\; d\lambda^{n}m_{pq} + \xi_* h_* d\lambda = 0$$

and it follows, after solving for $d\lambda$

$$d\lambda = \frac{{}^{n}n_{mn}\; E_{mnpq}\; d\epsilon_{pq}}{{}^{cros}n_{ab}\; E_{abcd}\; {}^{cros}m_{cd} - \xi_* h_*}$$

With this solution for $d\lambda$ one can obtain the increments in stress tensor and internal variables as

$$d\sigma_{mn} = E_{mnpq}\; d\epsilon_{pq} - E_{mnpq}\; \frac{{}^{n}n_{rs}\; E_{rstu}\; d\epsilon_{tu}}{{}^{n}n_{ab}\; E_{abcd}\; {}^{n}m_{cd} - \xi_A h_A}\; {}^{n}m_{pq} \tag{104.100}$$

$$dq_A = \left(\frac{{}^{n}n_{mn}\; E_{mnpq}\; d\epsilon_{pq}}{{}^{cros}n_{ab}\; E_{abcd}\; {}^{cros}m_{cd} - \xi_B h_B}\right) h_A \tag{104.101}$$

where ${}^{n}()$ denotes the starting elastic–plastic point for that increment. It should be noted that the explicit algorithm performs only one step of the computation and does not check on the equilibrium of the obtained solutions. This usually results in the slow drift of the stress–internal variable point from the yield surface for monotonic loading. It also results in spurious plastic deformations during elastic unloading for cyclic loading–unloading.

104.3.1 Continuum Tangent Stiffness Tensor.

The continuum tangent stiffness tensor (${}^{cont}E_{pqmn}^{ep}$) is obtained from the explicit (forward Euler) integration procedure (Jeremić and Sture, 1997):

$${}^{cont}E_{pqmn}^{ep} = E_{pqmn} - \frac{E_{pqkl}{}^{n}m_{kl}{}^{n}n_{ij}E_{ijmn}}{{}^{n}n_{ot}E_{otrs}\; {}^{n}m_{rs} - {}^{n}\xi_A\; h_A} \tag{104.102}$$

It is important to note that continuum tangent stiffness (${}^{cont}E_{pqmn}^{ep}$) posses minor symmetries (${}^{cont}E_{pqmn}^{ep} =$

${}^{cont}E^{ep}_{qpmn} = {}^{cont}E^{ep}_{pqnm}$), while major symmetry (${}^{cont}E^{ep}_{pqmn} = {}^{cont}E^{ep}_{mnpq}$), is only retained for associated elastic–plastic materials, when $n_{ij} \equiv m_{ij}$.

104.4 A Backward Euler (Implicit) Algorithm

In previous sections, the general theory of elasto–plasticity was presented. The accuracy and stability for the general Midpoint rule algorithm has been shown. In this chapter, the focus is on the Backward Euler algorithm, which is derived from the general Midpoint algorithm by setting $\alpha = 1$. The advantage of the Backward Euler scheme over other midpoint schemes is that the solution is sought by using the normal[40] at the final stress state. By implicitly assuming that such a stress state exists, the Backward Euler scheme is guaranteed to provide a solution, despite the size of the strain step[41]. However, it was shown in section (104.2.3.1) that the Backward Euler algorithm is only accurate to the first order.

The full implicit Backward Euler algorithm is based on the equation:

$$ {}^{n+1}\sigma_{ij} = {}^{pred}\sigma_{ij} - \Delta\lambda\; E_{ijkl}\; {}^{n+1}m_{kl} \qquad (104.103)$$

where ${}^{pred}\sigma_{ij} = E_{ijkl}\;\epsilon_{kl}$ is the elastic trial stress state, Q is the plastic potential function and ${}^{n+1}m_{kl} = \left.\frac{\partial Q}{\partial \sigma_{kl}}\right|_{n+1}$ is the gradient to the plastic potential function in the stress space at the final stress position, and

$$ {}^{pred}\sigma_{ij} = {}^{n}\sigma_{ij} + E_{ijkl}\; {}^{pred}\Delta\epsilon_{kl} \qquad (104.104)$$

is the elastic predicted (trial) stress state.

An initial estimate for the stress ${}^{n+1}\sigma_{ij}$ can be obtained using various other methods. This estimate generally does not satisfy the yield condition, so some kind of iterative scheme is necessary to return the stress to the yield surface.

104.4.1 Single Vector Return Algorithm.

If the predictor stress ${}^{pred}\sigma_{ij}$ is not in a corner or apex gray regions, a single vector return to the yield surface is possible. In order to derive such a scheme for a single vector return algorithm, a tensor of residuals r_{ij} will be defined as [42] :

$$ r_{ij} = \sigma_{ij} - \left({}^{pred}\sigma_{ij} - \Delta\lambda\; E_{ijkl}\; m_{kl} \right) \qquad (104.105)$$

---

[40] $m_{ij} = \partial Q / \partial \sigma_{ij}$

[41] large strain step increments were tested, the scheme converged to the solution even for deviatoric strain steps of 20% in magnitude.

[42] By default at increment $n+1$, and ${}^{n+1}()$ is omitted for simplicity.

This tensor represents the difference between the current stress state σ_{ij} and the Backward Euler stress state ${}^{pred}\sigma_{ij} - \Delta\lambda\ E_{ijkl}\ m_{kl}$.

The trial stress state ${}^{pred}\sigma_{ij}$ is kept fixed during the iteration process. The first order Taylor series expansion can be applied to Equation 104.105 to obtain the new residual ${}^{new}r_{ij}$ from the old one ${}^{old}r_{ij}$

$$ {}^{new}r_{ij} = {}^{old}r_{ij} + \mathrm{d}\sigma_{ij} + \mathrm{d}(\Delta\lambda)\ E_{ijkl}\ m_{kl} + \Delta\lambda\ E_{ijkl}\ \left(\frac{\partial m_{kl}}{\partial \sigma_{mn}} \mathrm{d}\sigma_{mn} + \frac{\partial m_{kl}}{\partial q_A} \mathrm{d}q_A \right) \tag{104.106} $$

where $\mathrm{d}\sigma_{ij}$ is the change in σ_{ij}, $\mathrm{d}(\Delta\lambda)$ is the change in $\Delta\lambda$, and $\frac{\partial m_{kl}}{\partial \sigma_{mn}} \mathrm{d}\sigma_{mn} + \frac{\partial m_{kl}}{\partial q_A} \mathrm{d}q_A$ is the change in m_{kl}. The goal is let ${}^{new}r_{ij} = \emptyset$, so one can write

$$ \emptyset = {}^{old}r_{ij} + \mathrm{d}\sigma_{ij} + \mathrm{d}(\Delta\lambda)\ E_{ijkl}\ m_{kl} + \Delta\lambda\ E_{ijkl}\ \left(\frac{\partial m_{kl}}{\partial \sigma_{mn}} \mathrm{d}\sigma_{mn} + \frac{\partial m_{kl}}{\partial q_A} \mathrm{d}q_A \right) \tag{104.107} $$

Similarly,

$$ q_A = {}^{n}q_A + \Delta\lambda\ h_A \tag{104.108} $$

r_A will be defined as:

$$ r_A = q_A - \left({}^{n}q_A + \Delta\lambda\ h_A \right) \tag{104.109} $$

and ${}^{n}q_A$ is kept fixed during iteration, that

$$ \emptyset = {}^{old}r_A + \mathrm{d}q_A - \mathrm{d}(\Delta\lambda)\ h_A - \Delta\lambda\ \left(\frac{\partial h_A}{\partial \sigma_{ij}} \mathrm{d}\sigma_{ij} + \frac{\partial h_A}{\partial q_B} \mathrm{d}q_B \right) \tag{104.110} $$

From equation 104.107 and 104.110, one obtains

$$ \begin{bmatrix} I^s_{ijmn} + \Delta\lambda E_{ijkl} \frac{\partial m_{kl}}{\partial \sigma_{mn}} & \Delta\lambda E_{ijkl} \frac{\partial m_{kl}}{\partial q_A} \\ -\Delta\lambda \frac{\partial h_A}{\partial \sigma_{ij}} & \delta_{AB} - \Delta\lambda \frac{\partial h_A}{\partial q_B} \end{bmatrix} \begin{Bmatrix} \mathrm{d}\sigma_{mn} \\ \mathrm{d}q_B \end{Bmatrix} + \mathrm{d}(\Delta\lambda) \begin{Bmatrix} E_{ijkl} m_{kl} \\ -h_A \end{Bmatrix} + \begin{Bmatrix} {}^{old}r_{ij} \\ {}^{old}r_A \end{Bmatrix} = \emptyset \tag{104.111} $$

Since $f(\sigma_{ij}, q_A) = 0$, one obtains

$$ \emptyset = {}^{old}f + n_{mn} \mathrm{d}\sigma_{mn} + \xi_B \mathrm{d}q_B \tag{104.112} $$

From equations 104.111 and 104.112,

$$ \mathrm{d}(\Delta\lambda) = \frac{ {}^{old}f - \begin{Bmatrix} n_{mn} & \xi_B \end{Bmatrix} \begin{bmatrix} I^s_{ijmn} + \Delta\lambda E_{ijkl} \frac{\partial m_{kl}}{\partial \sigma_{mn}} & \Delta\lambda E_{ijkl} \frac{\partial m_{kl}}{\partial q_A} \\ -\Delta\lambda \frac{\partial h_A}{\partial \sigma_{ij}} & \delta_{AB} - \Delta\lambda \frac{\partial h_A}{\partial q_B} \end{bmatrix}^{-1} \begin{Bmatrix} {}^{old}r_{ij} \\ {}^{old}r_A \end{Bmatrix} }{ \begin{Bmatrix} n_{mn} & \xi_B \end{Bmatrix} \begin{bmatrix} I^s_{ijmn} + \Delta\lambda E_{ijkl} \frac{\partial m_{kl}}{\partial \sigma_{mn}} & \Delta\lambda E_{ijkl} \frac{\partial m_{kl}}{\partial q_A} \\ -\Delta\lambda \frac{\partial h_A}{\partial \sigma_{ij}} & \delta_{AB} - \Delta\lambda \frac{\partial h_A}{\partial q_B} \end{bmatrix}^{-1} \begin{Bmatrix} E_{ijkl} m_{kl} \\ -h_A \end{Bmatrix} } \tag{104.113} $$

The iteration of $\Delta\lambda$ is then

$$\Delta\lambda^{k+1} = \Delta\lambda^{k} + \mathrm{d}(\Delta\lambda)^{k} \tag{104.114}$$

The iterative procedure is continued until the yield criterion $f = 0$, $\|r_{ij}\| = \emptyset$, and $\|r_A\| = \emptyset$ are satisfied within some tolerances at the final stress state [43].

In Equation 104.113, the generalized matrix $\mathbb{C}$, which is defined by

$$\mathbb{C} = \begin{bmatrix} I^s_{ijmn} + \Delta\lambda E_{ijkl}\frac{\partial m_{kl}}{\partial\sigma_{mn}} & \Delta\lambda E_{ijkl}\frac{\partial m_{kl}}{\partial q_A} \\ -\Delta\lambda\frac{\partial h_A}{\partial\sigma_{ij}} & \delta_{AB} - \Delta\lambda\frac{\partial h_A}{\partial q_B} \end{bmatrix}^{-1} \tag{104.115}$$

plays an important role in the implicit algorithm. It should be mentioned here that the above definition is a simplified expression for very general model with various isotropic and kinematic hardening. Specifically, if there is no hardening,

$$\mathbb{C} = \left[\; I^s_{ijmn} + \Delta\lambda E_{ijkl}\frac{\partial m_{kl}}{\partial\sigma_{mn}} \; \right]^{-1} \tag{104.116}$$

If there is only one isotropic internal variable q,

$$\mathbb{C} = \begin{bmatrix} I^s_{ijmn} + \Delta\lambda E_{ijkl}\frac{\partial m_{kl}}{\partial\sigma_{mn}} & \Delta\lambda E_{ijkl}\frac{\partial m_{kl}}{\partial q} \\ -\Delta\lambda\frac{\partial h}{\partial\sigma_{ij}} & 1 - \Delta\lambda\frac{\partial h}{\partial q} \end{bmatrix}^{-1} \tag{104.117}$$

For only one kinematic internal variable α_{ij},

$$\mathbb{C} = \begin{bmatrix} I^s_{ijmn} + \Delta\lambda E_{ijkl}\frac{\partial m_{kl}}{\partial\sigma_{mn}} & \Delta\lambda E_{ijkl}\frac{\partial m_{kl}}{\partial\alpha_{mn}} \\ -\Delta\lambda\frac{\partial h_{mn}}{\partial\sigma_{ij}} & I^s_{ijmn} - \Delta\lambda\frac{\partial h_{mn}}{\partial\alpha_{ij}} \end{bmatrix}^{-1} \tag{104.118}$$

For one isotropic variable q and one kinematic variable α_{ij},

$$\mathbb{C} = \begin{bmatrix} I^s_{ijmn} + \Delta\lambda E_{ijkl}\frac{\partial m_{kl}}{\partial\sigma_{mn}} & \Delta\lambda E_{ijkl}\frac{\partial m_{kl}}{\partial q} & \Delta\lambda E_{ijkl}\frac{\partial m_{kl}}{\partial\alpha_{mn}} \\ -\Delta\lambda\frac{\partial h}{\partial\sigma_{ij}} & 1 - \Delta\lambda\frac{\partial h}{\partial q} & -\Delta\lambda\frac{\partial h}{\partial\alpha_{ij}} \\ -\Delta\lambda\frac{\partial h_{mn}}{\partial\sigma_{ij}} & -\Delta\lambda\frac{\partial h_{mn}}{\partial q} & I^s_{ijmn} - \Delta\lambda\frac{\partial h_{mn}}{\partial\alpha_{ij}} \end{bmatrix}^{-1} \tag{104.119}$$

or for two kinematic variables z_{ij} and α_{ij},

$$\mathbb{C} = \begin{bmatrix} I^s_{ijmn} + \Delta\lambda E_{ijkl}\frac{\partial m_{kl}}{\partial\sigma_{mn}} & \Delta\lambda E_{ijkl}\frac{\partial m_{kl}}{\partial z_{mn}} & \Delta\lambda E_{ijkl}\frac{\partial m_{kl}}{\partial\alpha_{mn}} \\ -\Delta\lambda\frac{\partial h^z}{\partial\sigma_{ij}} & I^s_{ijmn} - \Delta\lambda\frac{\partial h^z}{\partial z_{mn}} & -\Delta\lambda\frac{\partial h^z}{\partial\alpha_{ij}} \\ -\Delta\lambda\frac{\partial h^\alpha_{mn}}{\partial\sigma_{ij}} & -\Delta\lambda\frac{\partial h^\alpha_{mn}}{\partial z_{mn}} & I^s_{ijmn} - \Delta\lambda\frac{\partial h^\alpha_{mn}}{\partial\alpha_{ij}} \end{bmatrix}^{-1} \tag{104.120}$$

If we define

$$\mathbf{n} = \begin{Bmatrix} n_{mn} \\ \xi_B \end{Bmatrix} \tag{104.121}$$

---

[43] $\|\|$ is some normal of the tensor

$$\mathbf{m} = \left\{ \begin{array}{c} E_{ijkl} m_{kl} \\ -h_A \end{array} \right\} \tag{104.122}$$

$$^{old}\mathbf{r} = \left\{ \begin{array}{c} ^{old}\sigma_{ij} \\ ^{old}r_A \end{array} \right\} \tag{104.123}$$

Equation 104.114 can be simplified as

$$\mathrm{d}(\Delta\lambda) = \frac{^{old}f - \mathbf{n}^T \ \mathbb{C} \ ^{old}\mathbf{r}}{\mathbf{n}^T \ \mathbb{C} \ \mathbf{M}} \tag{104.124}$$

and

$$\left\{ \begin{array}{c} \mathrm{d}\sigma_{mn} \\ \mathrm{d}q_B \end{array} \right\} = -\mathbb{C} \left(^{old}\mathbf{r} + \mathrm{d}(\Delta\lambda)\mathbf{m} \right) \tag{104.125}$$

104.4.2 Backward Euler Algorithms: Starting Points

Some remarks are necessary in order to clarify the Backward Euler Algorithm. It is a well known fact that the rate of convergence of the Newton - Raphson Method , or even obtaining convergence at all, is closely tied to the starting point for the iterative procedure. Bad initial or starting points might lead our algorithm to an oscillating solution, i.e. the algorithm does not converge. In the following, starting points for the Newton - Raphson iterative procedure will be established for one– and two–vector return algorithms.

104.4.2.1 Single Vector Return Algorithm Starting Point.

One of the proposed starting points (Crisfield, 1991) uses the normal at the elastic trial point[44] ${}^{pred}\sigma_{ij}$. A first order Taylor expansion about point ${}^{pred}\sigma_{ij}$ yields:

$$\begin{aligned}
{}^{pred}F^{new} &= {}^{pred}F^{old} + \left.\frac{\partial F}{\partial \sigma_{mn}}\right|_{pred} d\left({}^{pred}\sigma_{mn}\right) + \left.\frac{\partial F}{\partial q_A}\right|_{pred} dq_A = \\
&= {}^{pred}F^{old} + {}^{pred}n_{mn}\, d\sigma_{mn} + \xi_A h_A d\lambda = 0
\end{aligned} \tag{104.126}$$

It is assumed that the total incremental strain ϵ_{kl} is applied in order to reach the point ${}^{pred}\sigma_{ij}$, i.e. ${}^{pred}\sigma_{ij} = E_{ijkl}\ \epsilon_{kl}$ so that any further stress "relaxation" toward the yield surface takes place under zero total strain condition $\epsilon_{kl} = \emptyset$. From the differential form of equation (104.16) it follows:

$$\begin{aligned}
d\left({}^{pred}\sigma_{mn}\right) = E_{mnpq}\left(d\left({}^{pred}\epsilon_{pq}\right) - d\left({}^{pred}\epsilon^{p}_{pq}\right)\right) = \\
= -E_{mnpq}\ d\left({}^{pred}\epsilon^{p}_{pq}\right) = -E_{mnpq}\ d\lambda\left({}^{pred}m_{pq}\right)
\end{aligned}$$

and equation (104.126) becomes:

$${}^{pred}F^{old} - {}^{pred}n_{mn}\ E_{mnpq}\ d\lambda\, {}^{pred}m_{pq} + \xi_A h_A d\lambda = 0$$

and it follows:

$$d\lambda = \frac{{}^{pred}F^{old}}{{}^{pred}n_{mn}\ E_{mnpq}\ {}^{pred}m_{pq} - \xi_A h_A}$$

With this solution for $d\lambda$ we can obtain the starting point for the Newton-Raphson iterative procedure

$${}^{start}\sigma_{mn} = E_{mnpq}\ {}^{pred}\epsilon_{pq} - E_{mnpq}\ \frac{{}^{pred}F^{old}}{{}^{pred}n_{mn}\ E_{mnpq}\ {}^{pred}m_{pq} - \xi_A h_A}\ {}^{pred}m_{pq} \tag{104.127}$$

This starting point in six dimensional stress space will in general not satisfy the yield condition $F = 0$, but it will provide a good initial guess for the upcoming Newton-Raphson iterative procedure.

It should be mentioned, however, that this scheme for returning to the yield surface is the well known Radial Return Algorithm , if the yield criterion under consideration is of the von Mises type. In the special case the normal at the elastic trial point ${}^{pred}\sigma_{ij}$ coincides with the normal at the final stress state ${}^{n+1}\sigma_{ij}$, the return is exact, i.e. the yield condition is satisfied in one step.

Another possible and readily available starting point can be obtained by applying one Forward Euler step[45]. To be able to use the Forward Euler integration scheme, an intersection point has to be found. The procedure for calculating intersection points is given in section (104.2.4).

---

[44] I have named this scheme as semi Backward Euler scheme.

[45] or more steps for really large strain increments, for example over 10% in deviatoric direction. What has actually been done is to divide the θ region into several parts and depending on the curvature of the yield surface in deviatoric plane, use different schemes and different number of subincrements (the more curved, the more subincrements) to get the first, good initial guess. In the region around $\theta = 0$, one step of the semi Backward Euler scheme is appropriate, but close to $\theta = \pi/3$ the Forward Euler subincrementation works better.

A first order Taylor expansion about intersection point ${}^{cross}\sigma_{ij}$ yields:

$$\begin{aligned} F^{new} &= F^{old} + \left.\frac{\partial F}{\partial \sigma_{mn}}\right|_{cross} d\left({}^{cross}\sigma_{mn}\right) + \left.\frac{\partial F}{\partial q_A}\right|_{cross} dq_A = \\ &= {}^{cross}n_{mn}\; d\sigma_{mn} + \xi_A h_A d\lambda = 0 \end{aligned} \tag{104.128}$$

From the differential form of equation (104.16) it follows:

$$d\left({}^{fE}\sigma_{mn}\right) = E_{mnpq}\left(d\left(\epsilon_{pq}\right) - d\left(\epsilon^{p}_{pq}\right)\right) =$$

$$= E_{mnpq} d\left(\epsilon_{pq}\right) - E_{mnpq}\; d\left(\epsilon^{p}_{pq}\right) = E_{mnpq} d\left(\epsilon_{pq}\right) - E_{mnpq}\; d\lambda\left({}^{cross}m_{pq}\right)$$

and equation (104.128) becomes:

$$-{}^{cross}n_{mn}\; E_{mnpq}\; d\epsilon_{pq} - {}^{cross}n_{mn}\; E_{mnpq}\; d\lambda\, {}^{cross}m_{pq} + \xi_A h_A d\lambda = 0$$

and it follows

$$d\lambda = \frac{{}^{cross}n_{mn}\; E_{mnpq}\; d\epsilon_{pq}}{{}^{cros}n_{mn}\; E_{mnpq}\; {}^{cros}m_{pq} - \xi_A h_A}$$

With this solution for $d\lambda$ we can obtain the starting point for the Newton-Raphson iterative procedure

$${}^{start}\sigma_{mn} = E_{mnpq}\; d\epsilon_{pq} - E_{mnpq}\; \frac{{}^{cross}n_{rs}\; E_{rstu}\; d\epsilon_{tu}}{{}^{cross}n_{ab}\; E_{abcd}\; {}^{cross}m_{cd} - \xi_A h_A}\; {}^{cross}m_{pq} \tag{104.129}$$

This starting point in six–dimensional stress space will again not satisfy the yield condition[46] $F = 0$, but will provide a good initial estimate for the upcoming Newton-Raphson iterative procedure.

104.4.3 Consistent Tangent Stiffness Tensor

The final goal in deriving the Backward Euler scheme for integration of elasto–plastic constitutive equations is to use that scheme in finite element computations. If the Newton – Raphson iterative scheme is used at the global equilibrium level then the use of the so called traditional tangent stiffness tensor[47] E^{ep}_{ijkl} destroys the quadratic rate of asymptotic convergence of the iterative scheme. In order to preserve such a quadratic rate, a consistent, also called algorithmic, tangent stiffness tensor is derived. The consistent tangent stiffness tensor make use of derivatives of direction[48] normal to the potential function, and they are derived at the final, final at each iteration, that converges to the final stress point on the yield surface, stress point. The traditional forward scheme has a constant derivative, m_{ij} that is evaluated at the intersection point.

---

[46] except for the yield criteria that have flat yield surfaces (in the stress invariant space) so that the first order Taylor linear expansion, is exact.

[47] the one obtained with the Forward Euler method, i.e. where parameter $\alpha = 0$.

[48] $m_{ij} = \partial Q / \partial \sigma_{ij}$, i.e. $\partial m_{ij} / \partial \sigma_{kl} = \partial^2 Q / \partial \sigma_{ij} \partial \sigma_{kl}$.

It appears that Simo and Taylor (1985) and Runesson and Samuelsson (1985) have first derived the consistent tangent stiffness tensor. Other interesting articles on the subject can be found in Simo and Taylor (1986), Simo and Govindjee (1988), Jetteur (1986), Braudel et al. (1986), Crisfield (1987), Ramm and Matzenmiller (1988) and Mitchell and Owen (1988). As a consequence of consistency, the use of the consistent tangent stiffness tensor significantly improves the convergence characteristics of the overall equilibrium iterations, if a Newton - Raphson scheme is used for the latter. Use of the consistent tangent stiffness tensor yields a quadratic convergence rate of Newton - Raphson equilibrium iterations. In what follows, two derivations are given, namely the consistent tangent stiffness tensor for single– and two–vector return algorithms.

The concept of consistent linearization was introduced by Hughes and Pister (1978), while detailed explanation is given by Simo and Hughes (1998). The consistent tangent stiffness leads to quadratic convergence rates at global level.

It should be mentioned that there are various 'equivalent' forms of consistent tangent stiffness depending on the specific implicit algorithm equations. For instance, Simo and Hughes (1998), and Belytschko et al. (2001) derived the consistent tangent stiffness by taking current plastic strain as unknown and seeking its derivatives in the stress space; Pérez-Foguet and Huerta (1997) and Pérez-Foguet et al. (2000) used the numerical differentiation to calculate the consistent tangent stiffness in a compact matrix-vector form; Choi (2004) adopted the compact matrix-vector form by Pérez-Foguet and Huerta (1997) and Pérez-Foguet et al. (2000) but taking current plastic strain as unknown and seeking its derivatives in the elastic strain space. Slightly different from the above strategies, in this work the implicit algorithm is adopting the traditional form but taking current stress as unknown and seeking its derivatives in the stress space. Provided these differences, the consistent tangent stiffness in this work is slightly different from those in the above work.

104.4.3.1 Single Vector Return Algorithm.

In implicit algorithm, a very important advantage is that it may lead to consistent (algorithmic) tangent stiffness (Equation 104.137). The concept of consistent linearization was introduced in Hughes and Pister (1978), more details on consistent tangent stiffness were explained in Simo and Hughes (1998). The consistent tangent stiffness leads to quadratic convergence rates at global level.

It should be mentioned that there are various 'equivalent' forms of consistent tangent stiffness depending on the specific implicit algorithm equations. For instance, Simo and Hughes (1998), and Belytschko et al. (2001) derived the consistent tangent stiffness by taking current plastic strain as unknown and seeking its derivatives in the stress space; Pérez-Foguet and Huerta (1997) and Pérez-Foguet et al. (2000) used the numerical differentiation to calculate the consistent tangent stiffness in a compact matrix-vector form; Choi (2004) adopted the compact matrix-vector form by Pérez-Foguet and Huerta (1997) and Pérez-Foguet et al. (2000) but taking current plastic strain as unknown and seeking its derivatives in the elastic strain space. Slightly different from the above strategies, in this work (section 104.4) the implicit algorithm is adopting the traditional form but taking current stress as unknown and seeking its derivatives in the stress space. Provided these differences, the consistent tangent stiffness in this work is slightly different from those in the above work. The detail derivation will be followed.

When seeking the algorithmic tangent stiffness, we look into the explicit expression of $\mathrm{d}\sigma_{ij}/\mathrm{d}\epsilon_{mn}^{pred}$. At the same time, the internal variables are initialized the values at the previous time step, in other words, they are fixed within the time step when seeking the algorithmic tangent stiffness.

Linearize Equation 104.103, one obtains

$$\mathrm{d}\sigma_{ij} = E_{ijkl}\ \mathrm{d}\epsilon_{kl}^{pred} - \mathrm{d}(\Delta\lambda)\ E_{ijkl}\ m_{kl} - \Delta\lambda\ E_{ijkl}\ \left(\frac{\partial m_{kl}}{\partial \sigma_{mn}}\mathrm{d}\sigma_{mn} + \frac{\partial m_{kl}}{\partial q_A}\mathrm{d}q_A\right) \tag{104.130}$$

Similarly, linearize Equation 104.108, one obtains

$$\mathrm{d}q_A = \mathrm{d}(\Delta\lambda)\ h_A + \Delta\lambda\ \left(\frac{\partial h_A}{\partial \sigma_{ij}}\mathrm{d}\sigma_{ij} + \frac{\partial h_A}{\partial q_B}\mathrm{d}q_B\right) \tag{104.131}$$

From equation 104.130 and 104.131, one obtains

$$\begin{bmatrix} I^s_{ijmn} + \Delta\lambda E_{ijkl}\frac{\partial m_{kl}}{\partial \sigma_{mn}} & \Delta\lambda E_{ijkl}\frac{\partial m_{kl}}{\partial q_A} \\ -\Delta\lambda\frac{\partial h_A}{\partial \sigma_{ij}} & \delta_{AB} - \Delta\lambda\frac{\partial h_A}{\partial q_B} \end{bmatrix} \begin{Bmatrix} \mathrm{d}\sigma_{mn} \\ \mathrm{d}q_B \end{Bmatrix} + \mathrm{d}(\Delta\lambda) \begin{Bmatrix} E_{ijkl} m_{kl} \\ -h_A \end{Bmatrix} = \begin{Bmatrix} E_{ijkl}\ \mathrm{d}\epsilon_{kl}^{pred} \\ 0 \end{Bmatrix} \tag{104.132}$$

If one use the definitions of 104.115, 104.122 and 104.121, Equation 104.132 can be simplified to

$$\mathbb{C}^{-1} \begin{Bmatrix} \mathrm{d}\sigma_{mn} \\ \mathrm{d}q_B \end{Bmatrix} + \mathrm{d}(\Delta\lambda)\mathbf{m} = \begin{Bmatrix} E_{ijkl}\ \mathrm{d}\epsilon_{kl}^{pred} \\ 0 \end{Bmatrix} \tag{104.133}$$

Linearize the yield function $f(\sigma_{ij}, q_A) = 0$, one obtains

$$n_{mn}\mathrm{d}\sigma_{mn} + \xi_B \mathrm{d}q_B = 0 \tag{104.134}$$

or in a simplified form

$$\mathbf{n}^{\mathbf{T}} \begin{Bmatrix} \mathrm{d}\sigma_{mn} \\ \mathrm{d}q_B \end{Bmatrix} = \mathbf{0} \tag{104.135}$$

From Equations 104.133 and 104.135, one obtain

$$\mathrm{d}(\Delta\lambda) = \frac{\mathbf{n}^{\mathbf{T}}\mathbb{C}}{\mathbf{n}^{\mathbf{T}}\mathbb{C}\mathbf{m}} \begin{Bmatrix} E_{ijmn}\ \mathrm{d}\epsilon_{mn}^{pred} \\ 0 \end{Bmatrix} \tag{104.136}$$

Substitute expression 104.136 into 104.133, one obtains

$$\left\{ \begin{array}{c} \mathrm{d}\sigma_{ij} \\ \mathrm{d}q_A \end{array} \right\} = \left\{ \mathbb{C} - \frac{\mathbb{C}\mathbf{m}\mathbf{n}^T\mathbb{C}}{\mathbf{n}^T\ \mathbb{C}\ \mathbf{m}} \right\} \left\{ \begin{array}{c} E_{ijmn}\ \mathrm{d}\epsilon^{pred}_{mn} \\ 0 \end{array} \right\} \tag{104.137}$$

This equation gives the explicit expression of the consistent tangent stiffness $\mathrm{d}\sigma_{ij}/\mathrm{d}\epsilon^{pred}_{mn}$ for the implicit algorithm.

From section 104.4, if there are interactions between internal variables, the implicit algorithm will become very complicated. Simple models (e.g. von Mises model, or sometimes termed as J_2 model) have been proved efficient and good performance by the implicit algorithm (Simo and Hughes, 1998). Evidently, the implicit algorithm is mathematically based on the Newton-Raphson nonlinear equation solving method as well as the Eulerian backward integration method. Theoretically, the Newton-Raphson method may have quadratic convergence rate. However, Newton-Raphson method is not unconditional stable, and sometimes the iteration will diverge (Press et al., 1988a). Any bad starting point, non-continuous derivatives around solution, high nonlinearity, and interactions between internal variables, will deteriorate the implicit algorithm performance. A complicated model cannot guarantee good performance or quadratic convergence by the implicit algorithm Crisfield (1997a). The task to obtain the analytical expressions (Equations 104.116 to 104.120) may prove exceeding laborious for complicated plasticity models Simo and Hughes (1998).

104.4.4 Gradients to the Potential Function

In the derivation of the Backward Euler algorithm and the Consistent Tangent Matrix it is necessary to derive the first and the second derivatives of the potential function. The function Q is the function of the stress tensor σ_{ij} and the plastic variable tensor q_A. Derivatives with respect to the stress tensor σ_{ij} and plastic variable tensor q_A are given here. It is assumed that any stress state can be represented with the three stress invariants p, q and θ given in the following form:

$$p = -\frac{1}{3} I_1 \quad q = \sqrt{3 J_{2D}} \quad \cos 3\theta = \frac{3\sqrt{3}}{2} \frac{J_{3D}}{\sqrt{(J_{2D})^3}} \tag{104.138}$$

$$I_1 = \sigma_{kk} \quad J_{2D} = \frac{1}{2} s_{ij} s_{ij} \quad J_{3D} = \frac{1}{3} s_{ij} s_{jk} s_{ki} \quad s_{ij} = \sigma_{ij} - \frac{1}{3}\sigma_{kk}\delta_{ij} \tag{104.139}$$

Stresses are here chosen as positive in tension. The definition of Lode's angle θ in equation (104.138) implies that $\theta = 0$ defines the meridian of conventional triaxial extension (CTE), while $\theta = \pi/3$ denotes the meridian of conventional triaxial compression (CTC).

The Potential Function is given in the following form:

$$Q = Q(p, q, \theta) \tag{104.140}$$

The complete derivation of the closed form gradients is given in Appendix 703.

104.4.4.1 Analytical Gradients

The first derivative of the function Q in stress space is:

$$\frac{\partial Q}{\partial \sigma_{ij}} = \frac{\partial Q}{\partial p}\frac{\partial p}{\partial \sigma_{ij}} + \frac{\partial Q}{\partial q}\frac{\partial q}{\partial \sigma_{ij}} + \frac{\partial Q}{\partial \theta}\frac{\partial \theta}{\partial \sigma_{ij}} \tag{104.141}$$

and subsequently the first derivatives of the chosen stress invariants are

$$\frac{\partial p}{\partial \sigma_{ij}} = -\frac{1}{3}\,\delta_{ij} \tag{104.142}$$

$$\frac{\partial q}{\partial \sigma_{ij}} = \frac{3}{2}\,\frac{1}{q} s_{ij} \tag{104.143}$$

$$\frac{\partial \theta}{\partial \sigma_{ij}} = \frac{3}{2}\,\frac{\cos(3\theta)}{q^2 \sin(3\theta)} s_{ij} - \frac{9}{2}\,\frac{1}{q^3 \sin(3\theta)} t_{ij} \tag{104.144}$$

where:

$$t_{ij} = \frac{\partial J_{3D}}{\partial \sigma_{ij}}$$

The second derivative of the function Q in stress space is

$$\begin{aligned}
\frac{\partial^2 Q}{\partial \sigma_{pq}\partial \sigma_{mn}} &= \\
\left(\frac{\partial^2 Q}{\partial p^2}\frac{\partial p}{\partial \sigma_{mn}} + \frac{\partial^2 Q}{\partial p \partial q}\frac{\partial q}{\partial \sigma_{mn}} + \frac{\partial^2 Q}{\partial p \partial \theta}\frac{\partial \theta}{\partial \sigma_{mn}} \right) \frac{\partial p}{\partial \sigma_{pq}} + \frac{\partial Q}{\partial p}\frac{\partial^2 p}{\partial \sigma_{pq}\partial \sigma_{mn}} &+ \\
+ \left(\frac{\partial^2 Q}{\partial q \partial p}\frac{\partial p}{\partial \sigma_{mn}} + \frac{\partial^2 Q}{\partial q^2}\frac{\partial q}{\partial \sigma_{mn}} + \frac{\partial^2 Q}{\partial q \partial \theta}\frac{\partial \theta}{\partial \sigma_{mn}} \right) \frac{\partial q}{\partial \sigma_{pq}} + \frac{\partial Q}{\partial q}\frac{\partial^2 q}{\partial \sigma_{pq}\partial \sigma_{mn}} &+ \\
+ \left(\frac{\partial^2 Q}{\partial \theta \partial p}\frac{\partial p}{\partial \sigma_{mn}} + \frac{\partial^2 Q}{\partial \theta \partial q}\frac{\partial q}{\partial \sigma_{mn}} + \frac{\partial^2 Q}{\partial \theta^2}\frac{\partial \theta}{\partial \sigma_{mn}} \right) \frac{\partial \theta}{\partial \sigma_{pq}} + \frac{\partial Q}{\partial \theta}\frac{\partial^2 \theta}{\partial \sigma_{pq}\partial \sigma_{mn}} &
\end{aligned} \tag{104.145}$$

and the second derivatives of the stress invariants are

$$\frac{\partial^2 p}{\partial \sigma_{pq} \partial \sigma_{mn}} = \emptyset \tag{104.146}$$

$$\frac{\partial^2 q}{\partial \sigma_{pq} \partial \sigma_{mn}} = \frac{3}{2}\frac{1}{q}\left(\delta_{pm}\delta_{nq} - \frac{1}{3}\delta_{pq}\delta_{nm}\right) - \frac{9}{4}\frac{1}{q^3}\; s_{mn} s_{pq} \tag{104.147}$$

$$\begin{aligned} &\frac{\partial^2 \theta}{\partial \sigma_{pq} \partial \sigma_{mn}} = \\ &-\left(\frac{9}{2}\frac{\cos 3\theta}{q^4 \sin(3\theta)} + \frac{27}{4}\frac{\cos 3\theta}{q^4 \sin^3 3\theta}\right)\; s_{pq}\; s_{mn} + \frac{81}{4}\frac{1}{q^5 \sin^3 3\theta}\; s_{pq}\; t_{mn} + \\ &+\left(\frac{81}{4}\frac{1}{q^5 \sin 3\theta} + \frac{81}{4}\frac{\cos^2 3\theta}{q^5 \sin^3 3\theta}\right)\; t_{pq}\; s_{mn} - \frac{243}{4}\frac{\cos 3\theta}{q^6 \sin^3 3\theta}\; t_{pq}\; t_{mn} + \\ &+\frac{3}{2}\frac{\cos(3\theta)}{q^2 \sin(3\theta)}\; p_{pqmn} - \frac{9}{2}\frac{1}{q^3 \sin(3\theta)}\; w_{pqmn} \end{aligned} \tag{104.148}$$

where:

$$w_{pqmn} = \frac{\partial t_{pq}}{\partial \sigma_{mn}} = s_{np}\delta_{qm} + s_{qm}\delta_{np} - \frac{2}{3}s_{qp}\delta_{nm} - \frac{2}{3}\delta_{pq}s_{mn}$$

and:

$$p_{pqmn} = \frac{\partial s_{pq}}{\partial \sigma_{mn}} = \left(\delta_{mp}\delta_{nq} - \frac{1}{3}\delta_{pq}\delta_{mn}\right)$$

Another important gradient is:

$$\begin{aligned} &\frac{\partial^2 Q}{\partial \sigma_{ij} \partial q_A} = \frac{\partial m_{ij}}{\partial q_A} = \\ &= \frac{\partial \frac{\partial Q}{\partial p}}{\partial q_A}\frac{\partial p}{\partial \sigma_{ij}} + \frac{\partial \frac{\partial Q}{\partial q}}{\partial q_A}\frac{\partial q}{\partial \sigma_{ij}} + \frac{\partial \frac{\partial Q}{\partial \theta}}{\partial q_A}\frac{\partial \theta}{\partial \sigma_{ij}} = \\ &= \frac{\partial^2 Q}{\partial p \partial q_A}\frac{\partial p}{\partial \sigma_{ij}} + \frac{\partial^2 Q}{\partial q \partial q_A}\frac{\partial q}{\partial \sigma_{ij}} + \frac{\partial^2 Q}{\partial \theta \partial q_A}\frac{\partial \theta}{\partial \sigma_{ij}} \end{aligned} \tag{104.149}$$

104.4.4.2 Finite Difference Gradients

After having developed the closed form, analytical derivatives[49] the author of this thesis asked himself: "is there a simpler way of finding these derivatives?" One of the proposed ways to check the analytical solution is found in Dennis and Schnabel (1983). Dennis and Schnabel proposes the finite difference method for approximating derivatives if these derivatives are not analytically available and as a tool to check your analytical derivatives if they are derived.

Another good reason for developing alternative gradients is that for $\theta = 0, \pi/3$ gradients are not defined, i.e. indefinite terms as $0/0$ are appearing. One possible solution is the use of l'Hospital's rule. This has been done in Perić (1991). The solution to the problem in this work went in a different direction, i.e. instead of aiming for the analytical form, numerical derivatives are derived.

We should recall that for a function f of a single variable, the finite difference approximation to $f'(x)$, by using forward finite difference approach, is given by:

$$a = \frac{f(x+h) - f(x)}{h} \tag{104.150}$$

where h is a vanishingly small quantity. The same definition was used in deriving the finite difference approximation for the first derivative of the yield function F and potential function Q. The first derivative of F (or Q) with respect to the stress tensor σ_{ij} for diagonal elements is[50] :

$$^{approx.}F_{,ii} = \frac{F(\sigma_{ii} + h_{ii}) - F(\sigma_{ii})}{h_{ii}} \tag{104.151}$$

and for non-diagonal elements[51]:

$$^{approx.}F_{,ij} = \frac{F(\sigma_{ij} + h_{ij} + h_{ji}) - F(\sigma_{ij})}{2h_{ij}} \tag{104.152}$$

where h_{ij} is the step size which, because of finite precision arithmetic, is a variable[52].

The accuracy of the finite difference approximation to the analytical derivatives is closely bound to the step size h_{ij}. It is suggested in Dennis and Schnabel (1983)[section 5.4.] that for functions given by the simple formula, the number h should be $h = \sqrt{macheps}$, while for more complicated functions

---

[49] see Appendix (703).

[50] no sum convention implied, just the position of the element.

[51] since the stress tensor σ_{ij} is symmetric, change in one non-diagonal element triggers the other to be changed as well.

[52] it is actually one small number, h, that is multiplied with the current stress value so that the relative order of magnitude is retained.

that number should be larger. Here $macheps$ is the so called machine epsilon. It is defined as the smallest distinguishable positive number[53], such that $1.0 + macheps > 1.0$ on the given platform. For example[54], on the Intel x86 platform[55] $macheps = 1.08E-19$ while on the SUNSparc and DEC platforms $macheps = 2.22E-16$. It has been found that in the case of yield or potential functions the best approximation of analytical gradients is obtained by using $h = \sqrt{macheps}\ 10^3$. The three order of magnitude increase in the finite difference step is due to a rather complicated[56] formula for yield and potential functions. The error in the approximation, ${}^{approx.}F_{,ij}$ is found to be after the N^{th} decimal place, where N is the order of $macheps$, i.e. $macheps = O(N)$.

Second derivative approximations for one variable function are given in the form:

$$a = \frac{\big(f(x + h_i e_i + h_j e_j) - f(x + h_i e_i)\big) - \big(f(x + h_j e_j) - f(x)\big)}{h_i h_j} \tag{104.153}$$

If the first derivatives are available in closed form, one could use equations (104.151) and (104.152) just by replacing the function values with tensor values for analytical derivatives[57].

However, if the analytic derivatives are not available, one has to devise a formula that will create a fourth order tensor from the changes in two dimensional stress tensors, σ_{ij} and σ_{kl}. Using the scheme employed in equation (104.153) the following scheme has been devised:

$$\begin{aligned} {}^{approx.}Q_{,ijkl} = \\ \frac{\big(Q(\sigma_{mn} + h_{ij} + h_{kl}) - Q(\sigma_{mn} + h_{ij})\big) - \big(Q(\sigma_{mn} + h_{kl}) - Q(\sigma_{mn})\big)}{h_{ij} h_{kl}} \end{aligned} \tag{104.154}$$

Special considerations are necessary in order to retain symmetry of the fourth order tensor. At the moment it has not been possible to figure out how to build the finite difference approximation to the second derivatives of yield/potential functions for a general stress state. The only finite difference approximation of the second derivatives that appears to have worked was the one devised in principal stress space. Namely, diagonal elements of the analytical and the approximate gradients matched exactly, but development of non-diagonal elements, and the whole scheme of symmetrizing the fourth order approximation, still remain a mystery. However, some pattern was observed in non–diagonal elements, and the work on symmetrizing it is in progress.

---

[53] in a given precision, i.e. float (real*4), double (real*8) or long double (real*10).

[54] the precision sought was double (real*8).

[55] PC computers.

[56] One should not forget that we work with six dimensional tensor formulae directly.

[57] see Dennis and Schnabel (1983), section 5.6.

For many different potential functions (or yield functions) the only task left would be the derivation of the first derivatives of F and Q and the second derivatives of Q with respect to p, q and θ, namely the first derivatives $\frac{\partial Q}{\partial p}$, $\frac{\partial Q}{\partial q}$ and $\frac{\partial Q}{\partial \theta}$ and $\frac{\partial Q}{\partial p}$, $\frac{\partial Q}{\partial q}$ and $\frac{\partial Q}{\partial \theta}$ and the second derivatives $\frac{\partial^2 Q}{\partial p^2}$, $\frac{\partial^2 Q}{\partial p \partial q}$, $\frac{\partial^2 Q}{\partial p \partial \theta}$, $\frac{\partial^2 Q}{\partial q \partial p}$, $\frac{\partial^2 Q}{\partial q^2}$, $\frac{\partial^2 Q}{\partial q \partial \theta}$, $\frac{\partial^2 Q}{\partial \theta \partial p}$, $\frac{\partial^2 Q}{\partial \theta \partial q}$ and $\frac{\partial^2 Q}{\partial \theta^2}$. If the potential function is twice differentiable with respect to the stress tensor σ_{ij}, and if it is continuous then the Hessian matrix is symmetric.

104.5 Line Search Technique for Constitutive Elastic-Plastic Integration

This section is entirely based on Jeremić (2001). There exist a repetition of some previously defined equations...

104.6 Elastic and Elastic–Plastic Material Models for Solids

In this section we present elements of general elastic and elastic–plastic material models for engineering materials. We describe various forms of the yield functions, plastic flow directions and hardening and softening laws.

104.6.1 Elasticity

DSL COMMANDS for the elastic material models are given in section 205.3 on page 827.

In linear elasticity the relationship between the stress tensor σ_{ij} and the strain tensor ϵ_{kl} can be represented in the following form:

$$\sigma_{ij} = \sigma \left(\epsilon_{ij} \right) \tag{104.155}$$

If we assume the existence of a strain energy function[58] $W \left(\epsilon_{ij} \right)$ then the stress strain relation is:

$$\sigma_{ij} = \frac{\partial W \left(\epsilon_{ij} \right)}{\partial \epsilon_{ij}} \tag{104.156}$$

The introduction of the strain energy density function into elasticity is due to Green, and elastic solids for which such a function is assumed to exist are called Green elastic or hyperelastic solids.

---

[58] per unit volume.

Linearization of an elastic continuum is carried out with respect to a reference configuration which is stress free at temperature T_0, so that ${}^0\sigma_{ij} = 0$. If we denote as E_{ijkl} an isothermal modulus tensor, then under isothermal conditions, we obtain the generalized Hooke's law:

$$\sigma_{ij} = E_{ijkl}\epsilon_{kl} \tag{104.157}$$

where E_{ijkl} is the fourth order elastic stiffness tensor with 81 independent components in total. The elastic stiffness tensor features both minor symmetry $E_{ijkl} = E_{jikl} = E_{ijlk}$ and major symmetry $E_{ijkl} = E_{klij}$ (Jeremić and Sture, 1997). The number of independent components for such elastic stiffness tensor is 21 (Spencer, 1980).

$$E_{ijkl} = \left| \frac{\partial^2 W}{\partial\epsilon_{ij}\,\partial\epsilon_{kl}} \right|_{\epsilon=0} = \left| \frac{\partial^2 W}{\partial\epsilon_{kl}\,\partial\epsilon_{ij}} \right|_{\epsilon=0} \tag{104.158}$$

We will restrain our considerations to the isotropic case. The most general form of the isotropic tensor of rank 4 has the following representation:

$$I_4 = \lambda\delta_{ij}\delta_{kl} + \mu\delta_{ik}\delta_{jl} + \nu\delta_{il}\delta_{jk} \tag{104.159}$$

If E_{ijkl} has this form then in order to satisfy the symmetry condition[59] $E_{ijkl} = E_{jikl}$ we must have $\nu = \mu$. The symmetry condition[60] $E_{ijkl} = E_{klji}$ is then automatically satisfied. The elastic constant tensor has the following form:

$$E_{ijkl} = \lambda\delta_{ij}\delta_{kl} + \mu\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right) \tag{104.160}$$

where λ and μ are the Lamé coefficients:

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \quad ; \quad \mu = \frac{E}{2(1+\nu)} \tag{104.161}$$

and E and ν are Young's Modulus and Poisson's ratio respectively. The symmetric part of the fourth order unit tensor is :

$$I^{sym}_{ijkl} = \frac{1}{2}\left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jl}\right) \tag{104.162}$$

and can be found as multiplier of μ in equation (104.160). Equation (104.160) can be written in terms of E and ν as:

$$E_{ijkl} = \frac{E}{2(1+\nu)}\left(\frac{2\nu}{1-2\nu}\,\delta_{ij}\delta_{kl} + \delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right) \tag{104.163}$$

The same relation in terms of bulk modulus K and shear modulus G is:

---

[59] symmetry in stress tensor.

[60] existence of strain energy function.

$$E_{ijkl} = K\delta_{ij}\delta_{kl} + G\left(-\frac{2}{3}\delta_{ij}\delta_{kl} + \delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right) \tag{104.164}$$

where K and G are given as:

$$K = \lambda + \frac{2}{3}\mu \quad ; \quad G = \mu \tag{104.165}$$

The relation between the strain tensor, ϵ_{kl} and the stress tensor, σ_{ij} is:

$$\epsilon_{kl} = D_{klpq}\sigma_{pq} \tag{104.166}$$

where D_{klpq} is the elastic compliance fourth order tensor, defined as:

$$D_{klpq} = \frac{-\lambda}{2\mu\left(3\lambda + 2\mu\right)}\ \delta_{kl}\delta_{pq} + \frac{1}{4\mu}\left(\delta_{kp}\delta_{lq} + \delta_{kq}\delta_{lp}\right) \tag{104.167}$$

or in terms of E and ν:

$$D_{klpq} = \frac{1+\nu}{2E}\left(\frac{-2\nu}{1+\nu}\ \delta_{kl}\delta_{pq} + \delta_{kp}\delta_{lq} + \delta_{kq}\delta_{lp}\right) \tag{104.168}$$

of in terms of K and G:

$$D_{klpq} = \frac{1}{9K}\ \left(\delta_{kl}\delta_{pq}\right) + \frac{1}{2G}\left(-\frac{1}{3}\delta_{kl}\delta_{pq} + \frac{1}{2}\left(\delta_{kp}\delta_{lq} + \delta_{kq}\delta_{lp}\right)\right) \tag{104.169}$$

It is worthwhile noting that the part adjacent to the inverse of the bulk modulus K:

$$\left(\delta_{kl}\delta_{pq}\right)$$

controls the volumetric response and that the part adjacent to the inverse of the shear modulus G:

$$\left(-\frac{1}{3}\delta_{kl}\delta_{pq} + \frac{1}{2}\left(\delta_{kp}\delta_{lq} + \delta_{kq}\delta_{lp}\right)\right)$$

controls the shear response! This note will prove useful later on. Linear transformation of the stress tensor σ_{pq} into itself, i.e. σ_{ij} is defined as:

$$\sigma_{ij} = E_{ijkl}\epsilon_{kl} = E_{ijkl}D_{klpq}\sigma_{pq} \tag{104.170}$$

where

$$E_{ijkl}D_{klpq} = \frac{1}{2}\left(\delta_{ip}\delta_{jq} + \delta_{iq}\delta_{jp}\right) = I^{sym}_{ijpq} \tag{104.171}$$

104.6.1.1 Elastic Model

Linear elastic law is the simplest one and assumes constant Young's modulus E and constant Poisson's Ration ν.

104.6.1.2 Non–linear Elastic Model #1

This nonlinear model (Janbu, 1963), (Duncan and Chang, 1970) assumes dependence of the Young's modulus on the minor principal stress $\sigma_3 = \sigma_{min}$ in the form

$$E = K p_a \left(\frac{\sigma_3}{p_a} \right)^n \tag{104.172}$$

Here, p_a is the atmospheric pressure in the same units as E and stress. The two material constants K and n are constant for a given void ratio.

104.6.1.3 Non–linear Elastic Model #2

If Young's modulus and Poisson's ratio are replaced by the shear modulus G and bulk modulus K the non–linear elastic relationship can be expressed in terms of the normal effective mean stress p as

$$G \quad \text{and/or} \quad K = AF(e, OCR)p^n \tag{104.173}$$

where e is the void ratio, OCR is the overconsolidation ratio and $p = \sigma_{ii}/3$ is the mean effective stress (Hardin, 1978).

104.6.1.4 Lade's Non–linear Elastic Model

Lade and Nelson (1987) and Lade (1988a) proposed a nonlinear elastic model based on Hooke's law in which Poisson ratio ν is kept constant. According to this model, Young's modulus can be expressed in terms of a power law as:

$$E = M \; p_a \left(\left(\frac{I_1}{p_a} \right)^2 + \left(6 \frac{1+\nu}{1-2\nu} \right) \frac{J_{2D}}{p_a^2} \right)^\lambda \tag{104.174}$$

where $I_1 = \sigma_{ii}$ is the first invariant of the stress tensor and $J_{2D} = (s_{ij} s_{ij})/2$ is the second invariant of the deviatoric stress tensor $s_{ij} = \sigma_{ij} - \sigma_{kk}\delta_{ij}/3$. The parameter p_a is atmospheric pressure expressed in the same unit as E, I_1 and $\sqrt{J_{2D}}$ and the modulus number M and the exponent λ are constant, dimensionless numbers.

104.6.1.5 Cross Anisotropic Linear Elastic Model

104.6.2 Yield Functions

The typical plastic behavior of frictional materials is influenced by both normal and shear stresses. It is usually assumed that there exists a yield surface F in the stress space that encompasses the elastic region. States of stress inside the yield surface are assumed to be elastic (linear or non–linear). Stress states on the surface are assumed to produce plastic deformations. Yield surfaces for geomaterials are usually shaped as asymmetric tar drops with smoothly rounded triangular cross sections. In addition to that, simpler yield surfaces, based on the Drucker–Prager cone or Mohr–Coulomb hexagon can also be successfully used if matched with appropriate hardening laws. Yield surface shown in Figure 104.6 Lade (1988b) represent typical meridian plane trace for an isotropic granular material. Line BC represents stress path for conventional triaxial compression test. Figure 104.7 represents the view of the yield surface traces in the deviatoric plane.

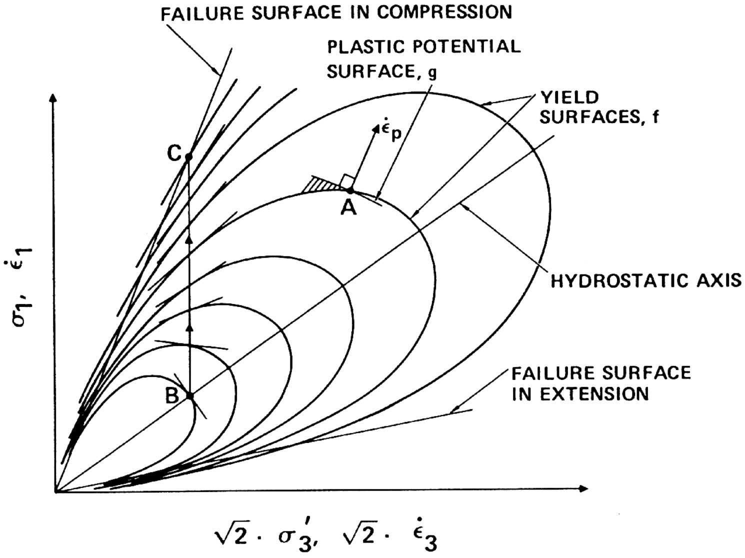


Figure 104.6: Yield surface patterns in the meridian plane for isotropic granular materials (from Lade (1988b))

104.6.3 Plastic Flow Directions

Plastic flow directions are traditionally derived from a potential surface which to some extent reassembles the yield surface. Potential surfaces for metals are the same as their yield surfaces but experimental evidence suggests that it is not the case for geomaterials. The non–associated flow rules, used in geomechanics, rely on the potential surface, which is different from the yield surface, to provide the plastic flow directions. It should be noted that the potential surface is used for convenience and there is no physical reason to assume that the plastic strain rates are related to a potential surface Q (Vardoulakis and Sulem, 1995). Instead of defining a plastic potential, one may assume that the plastic flow direction is derived from an tensor function which does not have to possess a potential function.

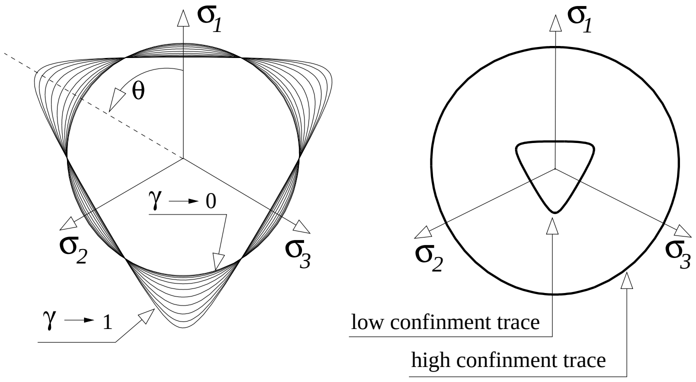


Figure 104.7: Deviatoric trace of typical yield surface for pressure sensitive materials.

104.6.4 Hardening–Softening Evolution Laws

The change in size and/or shape of the yield and potential surfaces is controlled by the hardening–softening evolution laws. Physically, these laws control the hardening and/or softening process during loading. Depending on the evolution type they control, these laws can be in general separated into isotropic and kinematic (also called anisotropic). The isotropic evolution laws control the size of the yield surface through a single scalar variable. This is usually related to the Coulomb friction or to the mean stress values at isotropic yielding. The non–isotropic evolution laws can be further specialized to rotational, translational kinematic and distortional. It should be noted that all of the kinematic evolution laws can be treated as special case of the general, distortional laws (Baltov and Sawczuk, 1965). Figure 104.8 depicts various types of evolution laws (for the control of hardening–softening) in the meridian plane[61].

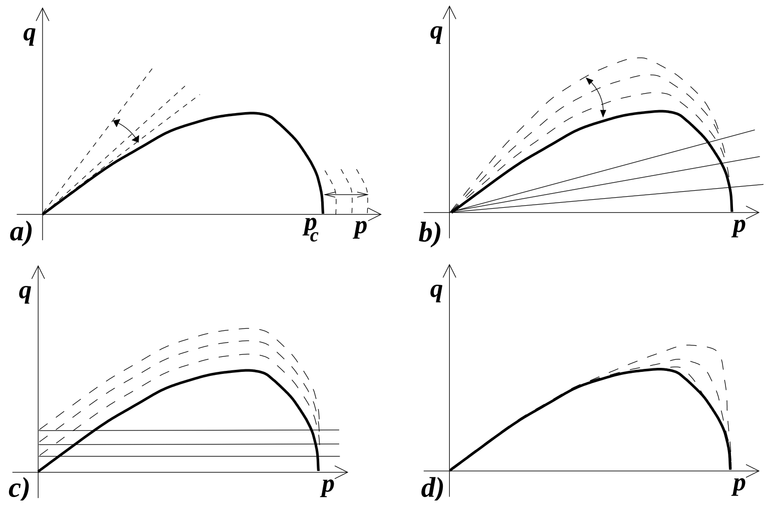


Figure 104.8: Various types of evolution laws that control hardening and/or softening of elastic–plastic material models: (a) Isotropic (scalar) controlling equivalent friction angle and isotropic yield stress. (b) Rotational kinematic hardening (second order tensor) controlling pivoting around fixed point (usually stress origin) of the yield surface. (c) Translational kinematic hardening (second order tensor) controlling translation of the yield surface. (d) Distortional (fourth order tensor) controlling the shape of the yield surface.

104.6.5 Tresca Model

The first yield criteria in the metal plasticity is Tresca yield criteria. Tresca yield criteria states that when the maximum shear stress or, the half difference of the maximum and minimum principal stresses, reaches the shear strength, τ_s, the material will begin yielding. It is can be expressed by the yield function

$$f = |\tau_{max}| - \tau_s = \frac{1}{2}|\sigma_1 - \sigma_3| - \tau_s = 0 \qquad (104.175)$$

Tresca yield surface in the principal stress space is a regular hexagonal cylinder. It is implied that the intermediate principal stress plays no role in the yielding for Tresca yield criteria.

104.6.6 von Mises Model

DSL COMMANDS for the von Mises material models are given in section 205.3 on page 827.

---

[61] The meridian plane is chosen just for illustration purposes, similar sketch can be produced in deviatoric plane as well.

Experimental data showed that for most metals, von Mises yield criteria is more accurate than Tresca criteria. von Mises yield function can be expressed by

$$f = 3J_2 - k^2 = 0 \tag{104.176}$$

or if extended to include the kinematic hardening,

$$f = \frac{3}{2}(s_{ij} - \alpha_{ij})(s_{ij} - \alpha_{ij}) - k^2 = 0 \tag{104.177}$$

where k is the scalar internal variable; its initial value is the uniaxial tension strength. α_{ij} is the tensor internal variable called the back stress. Similar to s_{ij}, α_{ij} is also a deviatoric symmetric tensor.

Although von Mises model is mainly for the metal plasticity analysis, for undrained analysis in geomechanics, von Mises model can be approximately used to simulate the undrained behaviors, (Yang and Jeremić, 2002), (Yang and Jeremić, 2003).

The stress derivative of the yield function is

$$\frac{\partial f}{\partial \sigma_{ij}} = 3(s_{ij} - \alpha_{ij}) \tag{104.178}$$

From Equation 104.178, it is easily to derive that

$$\frac{\partial f}{\partial \alpha_{ij}} = -3(s_{ij} - \alpha_{ij}) \tag{104.179}$$

and

$$\frac{\partial f}{\partial k} = -2k \tag{104.180}$$

If the associated plastic flow rule $g = f$ is assumed, then

$$m_{ij} = \frac{\partial g}{\partial \sigma_{ij}} = 3(s_{ij} - \alpha_{ij}) \tag{104.181}$$

$$\frac{\partial m_{ij}}{\partial \sigma_{mn}} = 3I^s_{ijmn} - \delta_{ij}\delta_{mn} \tag{104.182}$$

$$\frac{\partial m_{ij}}{\partial \alpha_{mn}} = -3I^s_{ijmn} \tag{104.183}$$

where I^s_{ijmn} is the symmetric unit rank-4 tensor.

It is interesting that from the Equation 104.181, von Mises model gives

$$\dot\epsilon^p_v = \dot\epsilon^p_{ii} = \dot\lambda m_{ii} = 3\dot\lambda(s_{ii} - \alpha_{ii}) = 0 \tag{104.184}$$

which accords with the phenomena that no plastic volumetric strain occurs for metals. It is implied that the isotropic stress (hydrostatic pressure) can never make the metal yield for this yield criteria. von Mises model is therefore pressure-independent.

If k is assumed a linear relation to the equivalent plastic strain ϵ_q^p, or by the equation

$$\dot{k} = H_s \dot{\epsilon}_{eq}^p = \dot{\lambda} H_s \left(\frac{2}{3} m_{ij}^{dev} m_{ij}^{dev} \right)^{0.5} \tag{104.185}$$

where H_s is the linear hardening/softening modulus to the equivalent plastic strain, the corresponding h_A is then

$$h = H_s \left(\frac{2}{3} m_{ij}^{dev} m_{ij}^{dev} \right)^{0.5} \tag{104.186}$$

where m_{ij}^{dev} is the 'deviatoric' plastic flow, and if it is associated plasticity,

$$h = 2 H_s k \tag{104.187}$$

If α_{ij} is assumed a linear relation to the plastic strain tensor ϵ_{ij}^p, or by the equation

$$\dot{\alpha}_{ij} = H_t \dot{\epsilon}_{ij}^p = \dot{\lambda} H_t m_{ij} \tag{104.188}$$

if it is associated plasticity,

$$\dot{\alpha}_{ij} = 3 \dot{\lambda} H_t (s_{ij} - \alpha_{ij}) \tag{104.189}$$

where H_t is the linear hardening/softening modulus to plastic strain tensor, the corresponding h_A is then

$$h_{ij} = H_t m_{ij} \tag{104.190}$$

if it is associated plasticity,

$$h_{ij} = 3 H_t (s_{ij} - \alpha_{ij}) \tag{104.191}$$

A saturation-type kinematic hardening rule is the Armstrong-Frederick hardening (Armstrong and Frederick, 1966),

$$\dot{\alpha}_{ij} = \frac{2}{3} h_a \dot{\epsilon}_{ij}^p - c_r \dot{\epsilon}_{eq}^p \alpha_{ij} \tag{104.192}$$

if it is associated plasticity,

$$\dot{\alpha}_{ij} = \dot{\lambda} [h_a (s_{ij} - \alpha_{ij}) - 2 c_r k \alpha_{ij}] \tag{104.193}$$

where h_a and c_r are material constants. The corresponding h_A is then

$$h_{ij} = \frac{2}{3} h_a m_{ij} - c_r m_{eq} \alpha_{ij} \tag{104.194}$$

where m_{eq} is the 'equivalent' plastic flow, and if it is associated plasticity,

$$h_{ij} = 2 h_a s_{ij} - 2 (h_a + c_r k) \alpha_{ij} \tag{104.195}$$

104.6.6.1 Yield and Plastic Potential Functions: von Mises Model (form I)

Yield function and related derivatives

$$f = \frac{3}{2}\left[\left(s_{ij}-\alpha_{ij}\right)\left(s_{ij}-\alpha_{ij}\right)\right] - k^2 = 0 \tag{104.196}$$

$$\begin{aligned}\frac{\partial f}{\partial \sigma_{ij}} &= 3\frac{\partial s_{kl}}{\partial \sigma_{ij}}\left(s_{kl}-\alpha_{kl}\right)\\ &= 3\left(\delta_{ki}\delta_{lj}-\frac{1}{3}\delta_{kl}\delta_{ij}\right)\left(s_{kl}-\alpha_{kl}\right)\\ &= 3\left(s_{ij}-\alpha_{ij}\right)\end{aligned} \tag{104.197}$$

$$\begin{aligned}\frac{\partial f}{\partial \alpha_{ij}} &= -3\frac{\partial \alpha_{kl}}{\partial \alpha_{ij}}\left(s_{kl}-\alpha_{kl}\right)\\ &= -3\delta_{ki}\delta_{lj}\left(s_{kl}-\alpha_{kl}\right)\\ &= -3\left(s_{ij}-\alpha_{ij}\right)\end{aligned} \tag{104.198}$$

$$\frac{\partial f}{\partial k} = -2k \tag{104.199}$$

Plastic flow (associated plasticity) and related derivatives

$$m_{ij} = \frac{\partial f}{\partial \sigma_{ij}} = 3\left(s_{ij}-\alpha_{ij}\right) \tag{104.200}$$

$$\frac{\partial m_{ij}}{\partial \sigma_{mn}} = 3\delta_{im}\delta_{jn} - \delta_{ij}\delta_{mn} \tag{104.201}$$

$$\frac{\partial m_{ij}}{\partial k} = 0 \tag{104.202}$$

$$\frac{\partial m_{ij}}{\partial \alpha_{mn}} = -3\delta_{im}\delta_{jn} \tag{104.203}$$

104.6.6.2 Yield and Plastic Potential Functions: von Mises Model (form II)

Yield function and related derivatives

$$f = \left[\left(s_{ij} - \alpha_{ij}\right)\left(s_{ij} - \alpha_{ij}\right)\right]^{0.5} - \sqrt{\frac{2}{3}}k = 0 \tag{104.204}$$

$$\begin{aligned}\frac{\partial f}{\partial \sigma_{ij}} &= \frac{\partial s_{kl}}{\partial \sigma_{ij}}\left(s_{kl} - \alpha_{kl}\right)\left[\left(s_{mn} - \alpha_{mn}\right)\left(s_{mn} - \alpha_{mn}\right)\right]^{-0.5} \\ &= \left(\delta_{ki}\delta_{lj} - \frac{1}{3}\delta_{kl}\delta_{ij}\right)\left(s_{kl} - \alpha_{kl}\right)\left[\left(s_{mn} - \alpha_{mn}\right)\left(s_{mn} - \alpha_{mn}\right)\right]^{-0.5} \\ &= \left(s_{ij} - \alpha_{ij}\right)\left[\left(s_{mn} - \alpha_{mn}\right)\left(s_{mn} - \alpha_{mn}\right)\right]^{-0.5}\end{aligned} \tag{104.205}$$

$$\frac{\partial f}{\partial \alpha_{ij}} = -\left(s_{ij} - \alpha_{ij}\right)\left[\left(s_{mn} - \alpha_{mn}\right)\left(s_{mn} - \alpha_{mn}\right)\right]^{-0.5} \tag{104.206}$$

$$\frac{\partial f}{\partial k} = -\sqrt{\frac{2}{3}} \tag{104.207}$$

Plastic flow (associated plasticity) and related derivatives

$$m_{ij} = \frac{\partial f}{\partial \sigma_{ij}} = \left(s_{ij} - \alpha_{ij}\right)\left[\left(s_{mn} - \alpha_{mn}\right)\left(s_{mn} - \alpha_{mn}\right)\right]^{-0.5} \tag{104.208}$$

$$\frac{\partial m_{ij}}{\partial \sigma_{mn}} = \left(\delta_{im}\delta_{jn} - \frac{1}{3}\delta_{ij}\delta_{mn}\right)\left[\left(s_{rs} - \alpha_{rs}\right)\left(s_{rs} - \alpha_{rs}\right)\right]^{-0.5} - \left(s_{ij} - \alpha_{ij}\right)\left(s_{mn} - \alpha_{mn}\right)\left[\left(s_{rs} - \alpha_{rs}\right)\left(s_{rs} - \alpha_{rs}\right)\right]^{-1.5} \tag{104.209}$$

$$\frac{\partial m_{ij}}{\partial k} = 0 \tag{104.210}$$

$$\frac{\partial m_{ij}}{\partial \alpha_{mn}} = -\delta_{im}\delta_{jn}\left[\left(s_{rs} - \alpha_{rs}\right)\left(s_{rs} - \alpha_{rs}\right)\right]^{-0.5} + \left(s_{ij} - \alpha_{ij}\right)\left(s_{mn} - \alpha_{mn}\right)\left[\left(s_{rs} - \alpha_{rs}\right)\left(s_{rs} - \alpha_{rs}\right)\right]^{-1.5} \tag{104.211}$$

104.6.6.3 Hardening and Softening Functions: von Mises Model

Linear isotropic hardening and related derivatives

$$\bar{k} = Hm^{equivalent} = H \left(\frac{2}{3} m_{ij} m_{ij} \right)^{0.5} \tag{104.212}$$

$$\frac{\partial \bar{k}}{\partial \sigma_{ij}} = \frac{2}{3} H m_{pq} \frac{\partial m_{pq}}{\partial \sigma_{ij}} \left(\frac{2}{3} m_{mn} m_{mn} \right)^{-0.5} \tag{104.213}$$

$$\frac{\partial \bar{k}}{\partial k} = \frac{2}{3} H m_{pq} \frac{\partial m_{pq}}{\partial k} \left(\frac{2}{3} m_{mn} m_{mn} \right)^{-0.5} \tag{104.214}$$

$$\frac{\partial \bar{k}}{\partial \alpha_{ij}} = \frac{2}{3} H m_{pq} \frac{\partial m_{pq}}{\partial \alpha_{ij}} \left(\frac{2}{3} m_{mn} m_{mn} \right)^{-0.5} \tag{104.215}$$

Linear kinematic hardening and related derivatives

$$\bar{\alpha}_{ij} = Hm^{dev}_{ij} = H \left(m_{ij} - \frac{1}{3} m_{kl} \delta_{kl} \delta_{ij} \right) \tag{104.216}$$

$$\frac{\partial \bar{\alpha}_{ij}}{\partial \sigma_{mn}} = H \left(\frac{\partial m_{ij}}{\partial \sigma_{mn}} - \frac{1}{3} \frac{\partial m_{kl}}{\partial \sigma_{mn}} \delta_{kl} \delta_{ij} \right) \tag{104.217}$$

$$\frac{\partial \bar{\alpha}_{ij}}{\partial k} = H \left(\frac{\partial m_{ij}}{\partial k} - \frac{1}{3} \frac{\partial m_{kl}}{\partial k} \delta_{kl} \delta_{ij} \right) \tag{104.218}$$

$$\frac{\partial \bar{\alpha}_{ij}}{\partial \alpha_{mn}} = H \left(\frac{\partial m_{ij}}{\partial \alpha_{mn}} - \frac{1}{3} \frac{\partial m_{kl}}{\partial \alpha_{mn}} \delta_{kl} \delta_{ij} \right) \tag{104.219}$$

Armstrong-Frederick kinematic hardening for von Mises

$$\bar{\alpha}_{ij} = \frac{2}{3} h_a m^{dev}_{ij} - c_r \left(\frac{2}{3} m^{dev}_{rs} m^{dev}_{rs} \right)^{0.5} \alpha_{ij} \tag{104.220}$$

The unit of parameter h_a is Pascal. The parameter c_r is unitless. The unit of α_{ij} is Pascal.

The deviatoric component of m is employed because the backstress α_{ij} is the center of yield surface in the deviatoric stress space.

When the derivative of backstress $\bar{\alpha}_{ij} = 0$, the tensor α_{ij} reaches the tensor limit.

$$\alpha_{ij}^{lim} = \sqrt{\frac{2}{3}}\frac{h_a}{c_r}\frac{m_{ij}^{dev}}{\sqrt{m_{rs}^{dev}m_{rs}^{dev}}} \tag{104.221}$$

Some useful tensor derivatives for von Mises $\bar{\alpha}$.

$$m_{ij}^{dev} = m_{ij} - \frac{1}{3}m_{kl}\delta_{kl}\delta_{ij} \tag{104.222}$$

- Useful tensor derivatives for von Mises $\bar{\alpha}$ with respect to σ.

$$\frac{\partial\bar{\alpha}_{ij}}{\partial\sigma_{mn}} = \frac{2}{3}h_a\frac{\partial m_{ij}^{dev}}{\partial\sigma_{mn}} - \frac{2}{3}c_r m_{rs}^{dev}\frac{\partial m_{rs}^{dev}}{\partial\sigma_{mn}}\left(\frac{2}{3}m_{kl}^{dev}m_{kl}^{dev}\right)^{-0.5}\alpha_{ij} \tag{104.223}$$

where

$$\frac{\partial m_{ij}^{dev}}{\partial\sigma_{mn}} = \frac{\partial m_{ij}}{\partial\sigma_{mn}} - \frac{1}{3}\frac{\partial m_{ot}}{\partial\sigma_{mn}}\delta_{ot}\delta_{ij} \tag{104.224}$$

- Useful tensor derivatives for von Mises $\bar{\alpha}$ with respect to α.

$$\frac{\partial\bar{\alpha}_{ij}}{\partial\alpha_{mn}} = \frac{2}{3}h_a\frac{\partial m_{ij}^{dev}}{\partial\alpha_{mn}} - \frac{2}{3}c_r m_{rs}^{dev}\frac{\partial m_{rs}^{dev}}{\partial\alpha_{mn}}\left(\frac{2}{3}m_{kl}^{dev}m_{kl}^{dev}\right)^{-0.5}\alpha_{ij} - c_r\left(\frac{2}{3}m_{pq}^{dev}m_{pq}^{dev}\right)^{0.5}\delta_{im}\delta_{jn} \tag{104.225}$$

where

$$\frac{\partial m_{ij}^{dev}}{\partial\alpha_{mn}} = \frac{\partial m_{ij}}{\partial\alpha_{mn}} - \frac{1}{3}\frac{\partial m_{ot}}{\partial\alpha_{mn}}\delta_{ot}\delta_{ij} \tag{104.226}$$

- Useful tensor derivatives for von Mises $\bar{\alpha}$ with respect to k.

$$\frac{\partial\bar{\alpha}_{ij}}{\partial k} = \frac{2}{3}h_a\frac{\partial m_{ij}^{dev}}{\partial k} - \sqrt{\frac{2}{3}}c_r\alpha_{ij}(m_{rs}^{dev}m_{rs}^{dev})^{-0.5}\frac{\partial m_{pq}^{dev}}{\partial k}m_{pq}^{dev} \tag{104.227}$$

where

$$\frac{\partial m_{ij}^{dev}}{\partial k} = \frac{\partial m_{ij}}{\partial k} - \frac{1}{3}\frac{\partial m_{ot}}{\partial k}\delta_{ot}\delta_{ij} \tag{104.228}$$

104.6.7 Drucker-Prager Model

DSL COMMANDS for the Drucker Prager material models are given in section 205.3 on page 827.

Drucker and Prager (1952) proposed a right circle cone to match with the Mohr-Coulomb irregular hexagonal pyramid, which can be expressed by

$$f = \alpha I_1 + \sqrt{J_2} - \beta = 0 \tag{104.229}$$

or if considering the kinematic hardening,

$$f = \alpha I_1 + [\frac{1}{2}(s_{ij} - p\alpha_{ij})(s_{ij} - p\alpha_{ij})]^{\frac{1}{2}} - \beta = 0 \tag{104.230}$$

where α and β are material constants.

By coinciding Drucker-Prager cone with the outer apexes of the Mohr-Coulomb hexagon locus, we get the compressive cone of Drucker-Prager model, with the constants as

$$\alpha = \frac{2\sin\phi}{\sqrt{3}(3-\sin\phi)}, \quad \beta = \frac{6\cos\phi}{\sqrt{3}(3-\sin\phi)}c \tag{104.231}$$

By coinciding Drucker-Prager cone with the inner apexes of the Mohr-Coulomb hexagon locus, we get the tensile cone of Drucker-Prager model, with the constants as

$$\alpha = \frac{2\sin\phi}{\sqrt{3}(3+\sin\phi)}, \quad \beta = \frac{6\cos\phi}{\sqrt{3}(3+\sin\phi)}c \tag{104.232}$$

We can also get the mean cone of the compressive and tensile cone, with the constants as

$$\alpha = \frac{\sqrt{3}\sin\phi}{9-\sin^2\phi}, \quad \beta = \frac{2\sqrt{3}\cos\phi}{9-\sin^2\phi}c \tag{104.233}$$

Another inner-tangent cone to the the Mohr-Coulomb pyramid, with the constants as

$$\alpha = \frac{\tan\phi}{\sqrt{9+12\tan^2\phi}}, \quad \beta = \frac{3c}{\sqrt{9+12\tan^2\phi}} \tag{104.234}$$

Obviously, in practice α and β are not directly obtained from experiments. They are functions of Mohr-Coulomb parameters, the cohesion c and the friction angle ϕ, which can be determined by experiments. The shape of Drucker-Prager yield surface has different types. They only partially satisfy the above requirements for locus in the π plane: they do not coincide with both compressive and tensile experimental points.

A useful formulation on Equation 104.229 is

$$\frac{\partial f}{\partial \sigma_{ij}} = \alpha\delta_{ij} + \frac{s_{ij}}{2\sqrt{J_2}} \tag{104.235}$$

For cohesionless sands, $k = 0$, Drucker-Prager yield function can thus be simplified as

$$f = \alpha I_1 + \sqrt{J_2} = 0 \tag{104.236}$$

or in terms of p and q,

$$f = q - Mp = 0 \tag{104.237}$$

If Equation 104.231 is adopted, then M can be easily derived as

$$M = \frac{6\sin\phi}{3-\sin\phi} \tag{104.238}$$

If the kinematic hardening is taken account, Equation 104.237 can be extended into

$$f = \frac{3}{2}[(s_{ij} - p\alpha_{ij})(s_{ij} - p\alpha_{ij})] - M^2p^2 = 0 \tag{104.239}$$

Useful formulations for this yield function are

$$\frac{\partial f}{\partial \sigma_{ij}} = 3\bar{s}_{ij} + \left(\bar{s}_{mn}\alpha_{mn} + \frac{2}{3}M^2 p\right)\delta_{ij} \tag{104.240}$$

$$\frac{\partial f}{\partial \alpha_{ij}} = -3p\bar{s}_{ij} \tag{104.241}$$

where $\bar{s}_{ij} = s_{ij} - p\alpha_{ij}$.

If the plastic flow is assumed associated, $g = f$, then

$$m_{ij} = \frac{\partial g}{\partial \sigma_{ij}} = 3\bar{s}_{ij} + \left(\bar{s}_{mn}\alpha_{mn} + \frac{2}{3}M^2 p\right)\delta_{ij} \tag{104.242}$$

the 'deviatoric' plastic flow is therefore

$$m_{eq} = 2Mp \tag{104.243}$$

104.6.7.1 Yield and Plastic Potential Functions: Drucker-Prager Model (form I)

Yield function and related derivatives

$$f = \frac{3}{2}\left[\left(s_{ij} - p\alpha_{ij}\right)\left(s_{ij} - p\alpha_{ij}\right)\right] - k^2p^2 = 0 \tag{104.244}$$

$$\begin{aligned}
\frac{\partial f}{\partial \sigma_{ij}} &= \frac{3}{2}\left[2\frac{\partial s_{mn}}{\partial \sigma_{ij}}\left(s_{mn} - p\alpha_{mn}\right)\right] + \frac{3}{2}\left[-2\alpha_{mn}\frac{\partial p}{\partial \sigma_{ij}}\left(s_{mn} - p\alpha_{mn}\right)\right] - 2k^2p\frac{\partial p}{\partial \sigma_{ij}} \\
&= 3\left(\delta_{mi}\delta_{nj} - \frac{1}{3}\delta_{mn}\delta_{ij}\right)\left(s_{mn} - p\alpha_{mn}\right) + 3\left[\alpha_{mn}\frac{1}{3}\delta_{ij}\left(s_{mn} - p\alpha_{mn}\right)\right] + \frac{2}{3}k^2p\delta_{ij} \\
&= 3\left(s_{ij} - p\alpha_{ij}\right) + \alpha_{mn}\left(s_{mn} - p\alpha_{mn}\right)\delta_{ij} + \frac{2}{3}k^2p\delta_{ij}
\end{aligned} \tag{104.245}$$

$$\frac{\partial f}{\partial \alpha_{ij}} = -3p\left(s_{ij} - p\alpha_{ij}\right) \tag{104.246}$$

$$\frac{\partial f}{\partial k} = -2kp^2 \tag{104.247}$$

Plastic flow (associated plasticity) and related derivatives

$$m_{ij} = \frac{\partial f}{\partial \sigma_{ij}} = 3\left(s_{ij} - p\alpha_{ij}\right) + \alpha_{rs}\left(s_{rs} - p\alpha_{rs}\right)\delta_{ij} + \frac{2}{3}k^2 p\delta_{ij} \tag{104.248}$$

$$\begin{aligned}
\frac{\partial m_{ij}}{\partial \sigma_{mn}} &= 3\left(\left(\delta_{im}\delta_{jn} - \frac{1}{3}\delta_{ij}\delta_{mn}\right) - \frac{1}{3}\delta_{mn}\alpha_{ij}\right) + \alpha_{rs}\frac{\partial\left(s_{rs} - p\alpha_{rs}\right)}{\partial \sigma_{mn}}\delta_{ij} + \frac{2}{3}k^2\frac{\partial p}{\partial \sigma_{mn}}\delta_{ij} \\
&= 3\delta_{im}\delta_{jn} - \delta_{ij}\delta_{mn} - \delta_{mn}\alpha_{ij} + \alpha_{rs}\left(\delta_{rm}\delta_{sn} - \frac{1}{3}\delta_{rs}\delta_{mn} + \frac{1}{3}\delta_{mn}\alpha_{rs}\right)\delta_{ij} + \frac{2}{3}k^2\frac{\partial p}{\partial \sigma_{mn}}\delta_{ij} \\
&= 3\delta_{im}\delta_{jn} - \delta_{ij}\delta_{mn} - \delta_{mn}\alpha_{ij} + \alpha_{mn}\delta_{ij} + \frac{1}{3}\delta_{mn}\alpha_{rs}\alpha_{rs}\delta_{ij} - \frac{2}{9}k^2\delta_{mn}\delta_{ij} \\
&= 3\delta_{im}\delta_{jn} + \left(-1 + \frac{1}{3}\alpha_{rs}\alpha_{rs} - \frac{2}{9}k^2\right)\delta_{ij}\delta_{mn} - \delta_{mn}\alpha_{ij} + \alpha_{mn}\delta_{ij}
\end{aligned} \tag{104.249}$$

$$\frac{\partial m_{ij}}{\partial k} = \frac{4}{3}kp\delta_{ij} \tag{104.250}$$

$$\begin{aligned}
\frac{\partial m_{ij}}{\partial \alpha_{mn}} &= -3p\delta_{im}\delta_{jn} + \delta_{rm}\delta_{sn}\left(s_{rs} - p\alpha_{rs}\right)\delta_{ij} - \alpha_{rs}p\delta_{rm}\delta_{sn}\delta_{ij} \\
&= -3p\delta_{im}\delta_{jn} + \left(s_{mn} - p\alpha_{mn}\right)\delta_{ij} - \alpha_{mn}p\delta_{ij} \\
&= -3p\delta_{im}\delta_{jn} + s_{mn}\delta_{ij} - 2p\alpha_{mn}\delta_{ij}
\end{aligned} \tag{104.251}$$

104.6.7.2 Yield and Plastic Potential Functions: Drucker-Prager Model (form II)

Yield function and related derivatives

$$f = \left[\left(s_{ij} - p\alpha_{ij}\right)\left(s_{ij} - p\alpha_{ij}\right)\right]^{0.5} - \sqrt{\frac{2}{3}}kp = 0 \tag{104.252}$$

$$\begin{aligned}
\frac{\partial f}{\partial \sigma_{ij}} &= \left(\frac{\partial s_{mn}}{\partial \sigma_{ij}} - \alpha_{mn}\frac{\partial p}{\partial \sigma_{ij}}\right)\left(s_{mn} - p\alpha_{mn}\right)\left[\left(s_{rs} - p\alpha_{rs}\right)\left(s_{rs} - p\alpha_{rs}\right)\right]^{-0.5} - \sqrt{\frac{2}{3}}k\frac{\partial p}{\partial \sigma_{ij}} \\
&= \left(\delta_{mi}\delta_{nj} - \frac{1}{3}\delta_{mn}\delta_{ij} + \frac{1}{3}\alpha_{mn}\delta_{ij}\right)\left(s_{mn} - p\alpha_{mn}\right)\left[\left(s_{rs} - p\alpha_{rs}\right)\left(s_{rs} - p\alpha_{rs}\right)\right]^{-0.5} + \sqrt{\frac{2}{27}}k\delta_{ij} \\
&= \left[\left(s_{ij} - p\alpha_{ij}\right) + \frac{1}{3}\alpha_{mn}\delta_{ij}\left(s_{mn} - p\alpha_{mn}\right)\right]\left[\left(s_{rs} - p\alpha_{rs}\right)\left(s_{rs} - p\alpha_{rs}\right)\right]^{-0.5} + \sqrt{\frac{2}{27}}k\delta_{ij}
\end{aligned} \tag{104.253}$$

$$\begin{aligned}
\frac{\partial f}{\partial \alpha_{ij}} &= -p\delta_{mi}\delta_{nj}\left(s_{mn} - p\alpha_{mn}\right)\left[\left(s_{rs} - p\alpha_{rs}\right)\left(s_{rs} - p\alpha_{rs}\right)\right]^{-0.5} \\
&= -p\left(s_{ij} - p\alpha_{ij}\right)\left[\left(s_{rs} - p\alpha_{rs}\right)\left(s_{rs} - p\alpha_{rs}\right)\right]^{-0.5}
\end{aligned} \tag{104.254}$$

$$\frac{\partial f}{\partial k} = -\sqrt{\frac{2}{3}}p \tag{104.255}$$

Plastic flow (associated plasticity) and related derivatives

$$m_{ij} = \frac{\partial f}{\partial \sigma_{ij}} = \left[\left(s_{ij} - p\alpha_{ij}\right) + \frac{1}{3}\alpha_{pq}\delta_{ij}\left(s_{pq} - p\alpha_{pq}\right)\right]\left[\left(s_{rs} - p\alpha_{rs}\right)\left(s_{rs} - p\alpha_{rs}\right)\right]^{-0.5} + \sqrt{\frac{2}{27}}k\delta_{ij} \tag{104.256}$$

$$\begin{aligned}\frac{\partial m_{ij}}{\partial \sigma_{mn}} = & \left[\left(\delta_{mi}\delta_{nj} - \frac{1}{3}\delta_{mn}\delta_{ij} + \frac{1}{3}\delta_{mn}\alpha_{ij}\right) + \frac{1}{3}\alpha_{pq}\delta_{ij}\left(\delta_{mp}\delta_{nq} - \frac{1}{3}\delta_{mn}\delta_{pq} + \frac{1}{3}\delta_{mn}\alpha_{pq}\right)\right] \\ & \left[\left(s_{rs} - p\alpha_{rs}\right)\left(s_{rs} - p\alpha_{rs}\right)\right]^{-0.5} \\ - & \left[\left(s_{ij} - p\alpha_{ij}\right) + \frac{1}{3}\alpha_{pq}\delta_{ij}\left(s_{pq} - p\alpha_{pq}\right)\right]\left(\delta_{mr}\delta_{ns} - \frac{1}{3}\delta_{mn}\delta_{rs} + \frac{1}{3}\delta_{mn}\alpha_{rs}\right) \\ & \left(s_{rs} - p\alpha_{rs}\right)\left[\left(s_{tu} - p\alpha_{tu}\right)\left(s_{tu} - p\alpha_{tu}\right)\right]^{-1.5}\end{aligned} \tag{104.257}$$

$$\frac{\partial m_{ij}}{\partial k} = \sqrt{\frac{2}{27}}\delta_{ij} \tag{104.258}$$

$$\begin{aligned}\frac{\partial m_{ij}}{\partial \alpha_{mn}} = & \left[-p\delta_{mi}\delta_{nj} + \frac{1}{3}\delta_{mp}\delta_{nq}\delta_{ij}\left(s_{pq} - p\alpha_{pq}\right) - \frac{1}{3}p\alpha_{pq}\delta_{ij}\delta_{mp}\delta_{nq}\right]\left[\left(s_{rs} - p\alpha_{rs}\right)\left(s_{rs} - p\alpha_{rs}\right)\right]^{-0.5} \\ - & \left[\left(s_{ij} - p\alpha_{ij}\right) + \frac{1}{3}\alpha_{pq}\delta_{ij}\left(s_{pq} - p\alpha_{pq}\right)\right]\left[-p\delta_{rm}\delta_{sn}\left(s_{rs} - p\alpha_{rs}\right)\right]\left[\left(s_{tu} - p\alpha_{tu}\right)\left(s_{tu} - p\alpha_{tu}\right)\right]^{-1.5}\end{aligned} \tag{104.259}$$

Plastic flow (non-associated plasticity) and related derivatives

$$m_{ij} = \left(\frac{\partial f}{\partial \sigma_{ij}}\right)^{dev} - \frac{1}{3}D\delta_{ij} = \left(s_{ij} - p\alpha_{ij}\right)\left[\left(s_{rs} - p\alpha_{rs}\right)\left(s_{rs} - p\alpha_{rs}\right)\right]^{-0.5} - \frac{1}{3}D\delta_{ij} \tag{104.260}$$

where

$$D = \xi\left(\sqrt{\frac{2}{3}}k_d - \frac{\sqrt{s_{mn}s_{mn}}}{p}\right) \tag{104.261}$$

$$\frac{\partial D}{\partial \sigma_{mn}} = -p^{-1}s_{mn}(s_{kl}s_{kl})^{-0.5} - \frac{1}{3}p^{-2}\delta_{mn}(s_{ot}s_{ot})^{0.5} \tag{104.262}$$

Therefore,

$$\begin{aligned}\frac{\partial m_{ij}}{\partial \sigma_{mn}} = & \left(\delta_{mi}\delta_{nj} - \frac{1}{3}\delta_{mn}\delta_{ij} + \frac{1}{3}\delta_{mn}\alpha_{ij}\right)\left[\left(s_{rs} - p\alpha_{rs}\right)\left(s_{rs} - p\alpha_{rs}\right)\right]^{-0.5} \\ - & \left(s_{ij} - p\alpha_{ij}\right)\left(\delta_{mr}\delta_{ns} - \frac{1}{3}\delta_{mn}\delta_{rs} + \frac{1}{3}\delta_{mn}\alpha_{rs}\right) \\ & \left(s_{rs} - p\alpha_{rs}\right)\left[\left(s_{tu} - p\alpha_{tu}\right)\left(s_{tu} - p\alpha_{tu}\right)\right]^{-1.5} \\ + & \frac{1}{3}\delta_{ij}s_{mn}p^{-1}(s_{kl}s_{kl})^{-0.5} + \frac{1}{9}\delta_{ij}\delta_{mn}p^{-2}(s_{ot}s_{ot})^{0.5}\end{aligned} \tag{104.263}$$

$$\frac{\partial m_{ij}}{\partial k} = 0 \tag{104.264}$$

$$\frac{\partial m_{ij}}{\partial \alpha_{mn}} = -p\delta_{mi}\delta_{nj}\left[(s_{rs} - p\alpha_{rs})(s_{rs} - p\alpha_{rs})\right]^{-0.5} - \left(s_{ij} - p\alpha_{ij}\right)\left[-p\delta_{rm}\delta_{sn}(s_{rs} - p\alpha_{rs})\right]\left[(s_{tu} - p\alpha_{tu})(s_{tu} - p\alpha_{tu})\right]^{-1.5} \tag{104.265}$$

104.6.7.3 Hardening and Softening Functions: Drucker-Prager Model

Note that the linear isotropic and linear kinematic hardening equations for Drucker-Prager model are the same as the ones for von Mises model. Here they are shown again for completeness.

Linear isotropic hardening and related derivatives

$$\bar{k} = Hm^{equivalent} = H\left(\frac{2}{3}m_{ij}m_{ij}\right)^{0.5} \tag{104.266}$$

$$\frac{\partial \bar{k}}{\partial \sigma_{ij}} = \frac{2}{3}Hm_{pq}\frac{\partial m_{pq}}{\partial \sigma_{ij}}\left(\frac{2}{3}m_{mn}m_{mn}\right)^{-0.5} \tag{104.267}$$

$$\frac{\partial \bar{k}}{\partial k} = \frac{2}{3}Hm_{pq}\frac{\partial m_{pq}}{\partial k}\left(\frac{2}{3}m_{mn}m_{mn}\right)^{-0.5} \tag{104.268}$$

$$\frac{\partial \bar{k}}{\partial \alpha_{ij}} = \frac{2}{3}Hm_{pq}\frac{\partial m_{pq}}{\partial \alpha_{ij}}\left(\frac{2}{3}m_{mn}m_{mn}\right)^{-0.5} \tag{104.269}$$

Linear kinematic hardening and related derivatives

$$\bar{\alpha}_{ij} = Hm^{dev}_{ij} = H\left(m_{ij} - \frac{1}{3}m_{kl}\delta_{kl}\delta_{ij}\right) \tag{104.270}$$

$$\frac{\partial \bar{\alpha}_{ij}}{\partial \sigma_{mn}} = H\left(\frac{\partial m_{ij}}{\partial \sigma_{mn}} - \frac{1}{3}\frac{\partial m_{kl}}{\partial \sigma_{mn}}\delta_{kl}\delta_{ij}\right) \tag{104.271}$$

$$\frac{\partial \bar{\alpha}_{ij}}{\partial k} = H\left(\frac{\partial m_{ij}}{\partial k} - \frac{1}{3}\frac{\partial m_{kl}}{\partial k}\delta_{kl}\delta_{ij}\right) \tag{104.272}$$

$$\frac{\partial \bar{\alpha}_{ij}}{\partial \alpha_{mn}} = H\left(\frac{\partial m_{ij}}{\partial \alpha_{mn}} - \frac{1}{3}\frac{\partial m_{kl}}{\partial \alpha_{mn}}\delta_{kl}\delta_{ij}\right) \tag{104.273}$$

Armstrong-Frederick Kinematic Hardening for Drucker-Prager

$$\bar{\alpha}_{ij} = \frac{2}{3}\frac{h_a}{p} m_{ij}^{dev} - c_r \left(\frac{2}{3} m_{rs}^{dev} m_{rs}^{dev}\right)^{0.5} \alpha_{ij} \tag{104.274}$$

where p is pressure of the current stress state. The unit of parameter h_a is Pascal. The parameter c_r is unitless. The α_{ij} in Drucker-Prager is unitless.

The pressure p is introduced for two reasons.

- The center of the yield surface is $p\alpha_{ij}$, not α_{ij}. The kinematic hardening rule is to control the center of the yield surface.
- The unit in equation (104.303) matches after the pressure p is introduced.

When the derivative of backstress $\bar{\alpha}_{ij} = 0$, the tensor α_{ij} reaches the tensor limit.

$$\alpha_{ij}^{lim} = \sqrt{\frac{2}{3}} \frac{h_a}{p c_r} \frac{m_{ij}^{dev}}{\sqrt{m_{rs}^{dev} m_{rs}^{dev}}} \tag{104.275}$$

Some useful tensor derivatives for Drucker-Prager $\bar{\alpha}$.

$$m_{ij}^{dev} = m_{ij} - \frac{1}{3} m_{kl} \delta_{kl} \delta_{ij} \tag{104.276}$$

- Useful tensor derivatives for Drucker-Prager $\bar{\alpha}$ with respect to σ.

$$\frac{\partial \bar{\alpha}_{ij}}{\partial \sigma_{mn}} = \frac{2}{3}\frac{h_a}{p}\frac{\partial m_{ij}^{dev}}{\partial \sigma_{mn}} + \frac{2}{9p^2} h_a m_{ij}^{dev} \delta_{mn} - \frac{2}{3} c_r m_{rs}^{dev} \frac{\partial m_{rs}^{dev}}{\partial \sigma_{mn}} \left(\frac{2}{3} m_{kl}^{dev} m_{kl}^{dev}\right)^{-0.5} \alpha_{ij} \tag{104.277}$$

where

$$\frac{\partial m_{ij}^{dev}}{\partial \sigma_{mn}} = \frac{\partial m_{ij}}{\partial \sigma_{mn}} - \frac{1}{3}\frac{\partial m_{ot}}{\partial \sigma_{mn}} \delta_{ot} \delta_{ij} \tag{104.278}$$

- Useful tensor derivatives for Drucker-Prager $\bar{\alpha}$ with respect to α.

$$\frac{\partial \bar{\alpha}_{ij}}{\partial \alpha_{mn}} = \frac{2}{3}\frac{h_a}{p}\frac{\partial m_{ij}^{dev}}{\partial \alpha_{mn}} - \frac{2}{3} c_r m_{rs}^{dev} \frac{\partial m_{rs}^{dev}}{\partial \alpha_{mn}} \left(\frac{2}{3} m_{kl}^{dev} m_{kl}^{dev}\right)^{-0.5} \alpha_{ij} - c_r \left(\frac{2}{3} m_{pq}^{dev} m_{pq}^{dev}\right)^{0.5} \delta_{im} \delta_{jn} \tag{104.279}$$

where

$$\frac{\partial m_{ij}^{dev}}{\partial \alpha_{mn}} = \frac{\partial m_{ij}}{\partial \alpha_{mn}} - \frac{1}{3}\frac{\partial m_{ot}}{\partial \alpha_{mn}} \delta_{ot} \delta_{ij} \tag{104.280}$$

- Useful tensor derivatives for Drucker-Prager $\bar{\alpha}$ with respect to k.

$$\frac{\partial \bar{\alpha}_{ij}}{\partial k} = \frac{2}{3}\frac{h_a}{p}\frac{\partial m_{ij}^{dev}}{\partial k} - \sqrt{\frac{2}{3}} c_r \alpha_{ij} (m_{rs}^{dev} m_{rs}^{dev})^{-0.5} \frac{\partial m_{pq}^{dev}}{\partial k} m_{pq}^{dev} \tag{104.281}$$

where

$$\frac{\partial m_{ij}^{dev}}{\partial k} = \frac{\partial m_{ij}}{\partial k} - \frac{1}{3}\frac{\partial m_{ot}}{\partial k} \delta_{ot} \delta_{ij} \tag{104.282}$$

104.6.7.4 Federico's Description of a Drucker–Prager Kinematic Hardenig Model

Presented is a concise description of nonlinear rotational kinematic hardening (Armstrong-Frederick) Drucker-Prager model (Lemaitre and Chaboche, 1990).

Elastic behavior The standard Hooke's law has been assumed for the sake of simplicity.

Yield function The yield locus is of the same kind described by Prevost (1985a) and Manzari and Dafalias (1997), i.e. conical and allowed to rotate around its apex (the centre of rotation coincides – for cohesionless materials – with the origin of the principal stress space):

$$f = \sqrt{\left(s_{ij} - p\alpha_{ij}\right)\left(s_{ij} - p\alpha_{ij}\right)} - \sqrt{\frac{2}{3}}kp = 0 \tag{104.283}$$

in which p is the isotropic mean pressure, α_{ij} the back–stress ratio tensor and k a constitutive surface parameter[62]. While this latter governs the opening angle of the cone, α_{ij} is a second–rank deviatoric tensor determining the yield locus rotation.

Plastic flow rule As usual, the incremental plastic strain tensor can be expressed as:

$$\epsilon_{ij}^{p} = \dot{\lambda} m_{ij} \tag{104.284}$$

where $\dot{\lambda}$ is a scalar plastic multiplier and m_{ij} assigning the direction of the plastic flow. While in the associated version of the model m_{ij} would coincide with the stress gradient of the yield function (104.283), here the following non–associated flow rule has been adopted (Prevost, 1985a; Manzari and Dafalias, 1997):

$$m_{ij} = \left(\frac{\partial f}{\partial \sigma_{ij}}\right)^{dev} - \frac{1}{3}D\delta_{ij} \tag{104.285}$$

where the superscript $^{\text{dev}}$ and δ_{ij} denote the "deviatoric" tensor operator and the Kronecker hydrostatic tensor, respectively, and D is a dilatancy coefficient defined as:

$$D = \xi\left(\sqrt{\frac{2}{3}}k_d - \sqrt{r_{mn}r_{mn}}\right) \tag{104.286}$$

Here, we defined normalized deviatoric stress tensor $r_{ij} = s_{ij}/p$.

---

[62] stresses are meant here to be effective

Apparently, the flow rule (104.285) implies the deviatoric plastic strain increment to be associated, while non-associativeness holds for the volumetric component. The definition (104.286) requires two consitutive parameters to be identified, namely k_d and ξ: the former represents the stress obliquity for the transition from contractive to dilative response; the latter quantitatively governs the volumetric plastic strain rate. Specifically, k_d denotes the existence of a "dilatancy surface", the soil response being contractive for inner stress states and dilative otherwise; this surface – characterized by no volumetric plastic strain – is still a Drucker–Prager conical locus, fixed in the principal stress space and with an opening angle given by k_d.

Armstrong–Frederick kinematic hardening rule The last ingredient in the model formulation is represented by the hardening rule for the internal variable α_{ij}. Here, an Armstrong–Frederick hardening (Armstrong and Frederick, 1966; Lemaitre and Chaboche, 1990) has been introduced:

$$\dot{\alpha}_{ij} = \frac{2}{3} h_a \left(\epsilon^p_{ij}\right)^{dev} - c_r \alpha_{ij} \sqrt{\frac{2}{3} \left(\epsilon^p_{rs}\right)^{dev} \left(\epsilon^p_{rs}\right)^{dev}} \tag{104.287}$$

where h_a and c_r are two hardening constitutive parameters. Equation (104.287) yields a saturation–type evolution under deviatoric plastic straining, up to the achievement of a limit back–stress ratio α^{lim}_{ij}. Starting from (104.287), it could be easily proven that:

$$\left\|\alpha^{lim}_{ij}\right\| = \sqrt{\frac{2}{3}} \frac{h_a}{c_r} \tag{104.288}$$

i.e. the norm of the limit tensor α^{lim}_{ij} exclusively depends on the h_a/c_r ratio for any value of the Lode angle. The existence of an outer bound for α_{ij} implies all feasible stress states to lie within a so–called bounding surface, determined both by α^{lim}_{ij} and the opening of the yield locus $f = 0$ (104.283). The bounding surface, governing the shear strength of the material, is in this case a Lode–angle insensitive Drucker–Prager cone: in the simplest case of triaxial loading, the limit stress obliquity M can be shown to equal:

$$M = \left(\frac{q}{p'}\right)_{failure} = k + \frac{h_a}{c_r} \tag{104.289}$$

both in compression and extension. From the above analytical relationships it can be inferred that the material shear strength is ruled by the ratio h_a/c_r, whereas c_r determines the evolution rate of the back–stress tensor α_{ij}.

104.6.7.5 Han's Description of Drucker–Prager Model with Armstrong–Frederick Kinematic Hardening

Presented is a new description of non-associated Drucker-Prager model with nonlinear Armstrong-Frederick kinematic hardening. Compared with the traditional description shown in earlier sections, the main difference in this description is taking into consideration of the symmetry nature of stress and strain tensors. As a result, some of the derivatives of plastic flow and internal variable are different. More importantly, this change leads to a consistent stiffness tensor that has minor symmetry.

Elastic Behavior The elastic behavior is modeled as linear elastic, following classic generalized Hook's law. The elastic stiffness tensor is that for isotropic, linear elastic material (Equation 104.163).

$$E_{ijkl} = \frac{E}{2(1+\nu)} \left(\frac{2\nu}{1-2\nu} \delta_{ij}\delta_{kl} + \delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk} \right) \tag{104.290}$$

Yield Function The yield function and its derivatives remain unchanged.

$$f = \left[\left(s_{ij} - p\alpha_{ij}\right) \left(s_{ij} - p\alpha_{ij}\right) \right]^{0.5} - \sqrt{\frac{2}{3}} kp = 0 \tag{104.291}$$

$$\frac{\partial f}{\partial \sigma_{ij}} = \left[\left(s_{ij} - p\alpha_{ij}\right) + \frac{1}{3}\alpha_{mn}\delta_{ij} \left(s_{mn} - p\alpha_{mn}\right) \right] \left[\left(s_{rs} - p\alpha_{rs}\right) \left(s_{rs} - p\alpha_{rs}\right) \right]^{-0.5} + \sqrt{\frac{2}{27}} k\delta_{ij} \tag{104.292}$$

$$\frac{\partial f}{\partial \alpha_{ij}} = -p \left(s_{ij} - p\alpha_{ij}\right) \left[\left(s_{rs} - p\alpha_{rs}\right) \left(s_{rs} - p\alpha_{rs}\right) \right]^{-0.5} \tag{104.293}$$

$$\frac{\partial f}{\partial k} = -\sqrt{\frac{2}{3}} p \tag{104.294}$$

Plastic Flow The changes in derivatives of plastic flow, and later in hardening, comes from the clarification of derivative of the stress tensor with respect to itself. According to Park (2018), since the stress tensor is intrinsically symmetric, its derivative with respect to itself should be:

$$\frac{\partial \sigma_{ij}}{\partial \sigma_{mn}} = \frac{1}{2} \left(\delta_{im}\delta_{jn} + \delta_{in}\delta_{jm} \right) \tag{104.295}$$

Then, the derivative of deviatoric stress with respect to stress becomes:

$$\frac{\partial s_{ij}}{\partial \sigma_{mn}} = \frac{1}{2}\delta_{im}\delta_{jn} + \frac{1}{2}\delta_{in}\delta_{jm} - \frac{1}{3}\delta_{mn}\delta_{ij} \tag{104.296}$$

The non-associated plastic flow still has the same form:

$$m_{ij} = \left(\frac{\partial f}{\partial \sigma_{ij}} \right)^{dev} - \frac{1}{3} D\delta_{ij} = \left(s_{ij} - p\alpha_{ij}\right) \left[\left(s_{rs} - p\alpha_{rs}\right) \left(s_{rs} - p\alpha_{rs}\right) \right]^{-0.5} - \frac{1}{3} D\delta_{ij} \tag{104.297}$$

where

$$D = \xi \left(\sqrt{\frac{2}{3}} k_d - \frac{\sqrt{s_{mn} s_{mn}}}{p} \right) \tag{104.298}$$

and

$$\frac{\partial D}{\partial \sigma_{mn}} = -\xi \left[p^{-1} s_{mn} (s_{kl} s_{kl})^{-0.5} + \frac{1}{3} p^{-2} \delta_{mn} (s_{ot} s_{ot})^{0.5} \right] \tag{104.299}$$

Therefore,

$$\begin{aligned}\frac{\partial m_{ij}}{\partial \sigma_{mn}} =& \left(\frac{1}{2} \delta_{im} \delta_{jn} + \frac{1}{2} \delta_{in} \delta_{jm} - \frac{1}{3} \delta_{mn} \delta_{ij} + \frac{1}{3} \delta_{mn} \alpha_{ij} \right) \left[(s_{rs} - p\alpha_{rs}) (s_{rs} - p\alpha_{rs}) \right]^{-0.5} \\ & - \left(s_{ij} - p\alpha_{ij} \right) \left[(s_{mn} - p\alpha_{mn}) + \frac{1}{3} \alpha_{rs} (s_{rs} - p\alpha_{rs}) \delta_{mn} \right] \left[(s_{tu} - p\alpha_{tu}) (s_{tu} - p\alpha_{tu}) \right]^{-1.5} \\ & + \frac{1}{3} \xi \left[s_{mn} p^{-1} (s_{kl} s_{kl})^{-0.5} + \frac{1}{3} \delta_{mn} p^{-2} (s_{ot} s_{ot})^{0.5} \right] \delta_{ij}\end{aligned} \tag{104.300}$$

$$\frac{\partial m_{ij}}{\partial k} = 0 \tag{104.301}$$

$$\begin{aligned}\frac{\partial m_{ij}}{\partial \alpha_{mn}} =& -p \left(\frac{1}{2} \delta_{mi} \delta_{nj} + \frac{1}{2} \delta_{mj} \delta_{ni} \right) \left[(s_{rs} - p\alpha_{rs}) (s_{rs} - p\alpha_{rs}) \right]^{-0.5} \\ & + p \left(s_{ij} - p\alpha_{ij} \right) (s_{mn} - p\alpha_{mn}) \left[(s_{tu} - p\alpha_{tu}) (s_{tu} - p\alpha_{tu}) \right]^{-1.5}\end{aligned} \tag{104.302}$$

Armstrong-Frederick kinematic hardening for Drucker-Prager

$$\bar{\alpha}_{ij} = \frac{2}{3} \frac{h_a}{p_{atm}} m_{ij}^{dev} - c_r \left(\frac{2}{3} m_{rs}^{dev} m_{rs}^{dev} \right)^{0.5} \alpha_{ij} \tag{104.303}$$

where p_{atm} is the atmospheric pressure of 101.325 kPa. The unit of parameter h_a is Pascal. The parameter c_r is unitless. The α_{ij} in Drucker-Prager is unitless. The atmospheric pressure p_{atm} is introduced so that the unit in equation (104.303) matches.

Notice that

$$m_{rs}^{dev} m_{rs}^{dev} = (s_{rs} - p\alpha_{rs}) (s_{rs} - p\alpha_{rs}) \left[(s_{tu} - p\alpha_{tu}) (s_{tu} - p\alpha_{tu}) \right]^{-1} = 1 \tag{104.304}$$

This means the hardening equation and related derivatives can be simplified:

$$\bar{\alpha}_{ij} = \frac{2}{3} \frac{h_a}{p_{atm}} m_{ij}^{dev} - \sqrt{\frac{2}{3}} c_r \alpha_{ij} \tag{104.305}$$

When the derivative of backstress $\bar{\alpha}_{ij} = 0$, the tensor α_{ij} reaches the tensor limit:

$$\alpha_{ij}^{lim} = \sqrt{\frac{2}{3}} \frac{h_a}{p_{atm} c_r} m_{ij}^{dev} \tag{104.306}$$

Some useful tensor derivatives for Drucker-Prager $\bar{\alpha}$.

$$m_{ij}^{dev} = \left(s_{ij} - p\alpha_{ij}\right) \left[\left(s_{rs} - p\alpha_{rs}\right)\left(s_{rs} - p\alpha_{rs}\right)\right]^{-0.5} \tag{104.307}$$

$$\begin{aligned}\frac{\partial m_{ij}^{dev}}{\partial \sigma_{mn}} = & \left(\frac{1}{2}\delta_{im}\delta_{jn} + \frac{1}{2}\delta_{in}\delta_{jm} - \frac{1}{3}\delta_{mn}\delta_{ij} + \frac{1}{3}\delta_{mn}\alpha_{ij}\right) \left[\left(s_{rs} - p\alpha_{rs}\right)\left(s_{rs} - p\alpha_{rs}\right)\right]^{-0.5} \\ & - \left(s_{ij} - p\alpha_{ij}\right) \left[\left(s_{mn} - p\alpha_{mn}\right) + \frac{1}{3}\alpha_{rs}\left(s_{rs} - p\alpha_{rs}\right)\delta_{mn}\right] \left[\left(s_{tu} - p\alpha_{tu}\right)\left(s_{tu} - p\alpha_{tu}\right)\right]^{-1.5}\end{aligned} \tag{104.308}$$

$$\frac{\partial m_{ij}^{dev}}{\partial k} = 0 \tag{104.309}$$

$$\begin{aligned}\frac{\partial m_{ij}^{dev}}{\partial \alpha_{mn}} = & -p\left(\frac{1}{2}\delta_{im}\delta_{jn} + \frac{1}{2}\delta_{in}\delta_{jm}\right) \left[\left(s_{rs} - p\alpha_{rs}\right)\left(s_{rs} - p\alpha_{rs}\right)\right]^{-0.5} \\ & - p\left(s_{ij} - p\alpha_{ij}\right)\left(s_{mn} - p\alpha_{mn}\right) \left[\left(s_{tu} - p\alpha_{tu}\right)\left(s_{tu} - p\alpha_{tu}\right)\right]^{-1.5}\end{aligned} \tag{104.310}$$

- Tensor derivative of Drucker-Prager $\bar{\alpha}_{ij}$ with respect to σ_{ij}.

$$\frac{\partial \bar{\alpha}_{ij}}{\partial \sigma_{mn}} = \frac{2h_a}{3p_{atm}} \frac{\partial m_{ij}^{dev}}{\partial \sigma_{mn}} \tag{104.311}$$

- Tensor derivative of Drucker-Prager $\bar{\alpha}_{ij}$ with respect to k.

$$\frac{\partial \bar{\alpha}_{ij}}{\partial k} = 0 \tag{104.312}$$

- Tensor derivative of Drucker-Prager $\bar{\alpha}_{ij}$ with respect to α_{ij}.

$$\frac{\partial \bar{\alpha}_{ij}}{\partial \alpha_{mn}} = \frac{2h_a}{3p_{atm}} \frac{\partial m_{ij}^{dev}}{\partial \alpha_{mn}} - \sqrt{\frac{2}{3}} c_r \left(\frac{1}{2}\delta_{im}\delta_{jn} + \frac{1}{2}\delta_{in}\delta_{jm}\right) \tag{104.313}$$

104.6.8 Hyperbolic Drucker Prager Model

104.6.8.1 Original Yield Function and Hyperbolic Function

The original Drucker-Prager yield function is

$$\Phi(\sigma) = \sqrt{J_2} - \eta p - \xi \tag{104.314}$$

where

$$\eta = \frac{3tan\phi}{\sqrt{9 + 12tan^2\phi}} \tag{104.315}$$

$$\xi = \frac{3c}{\sqrt{9 + 12tan^2\phi}} \tag{104.316}$$

where ϕ is the friction ratio, and c is the cohesion. Besides, the constant η is unitless. The unit of constant ξ is $Pascal$. In addition, here we define that p is compression-positive. Namely, $p = -\frac{\sigma_{ii}}{3}$.

Rewrite the yield function (104.314) with p, q such that

$$\Phi(\sigma) = \frac{q}{\sqrt{3}} - \eta p - \xi \tag{104.317}$$

The slope of the yield surface is $\sqrt{3}\eta$, and the hydrostatic cutoff point is at $(-\frac{\xi}{\eta}, 0)$.

104.6.8.2 Hyperbolic Drucker Prager Model

Assume a standard hyperbolic function is

$$\left(\frac{x-d}{a}\right)^2 - \left(\frac{y}{b}\right)^2 = 1 \tag{104.318}$$

In the hyperbolic function, as showed in the Figure 104.9, the distance between the original apex to the rounded hydrostatic cut-off is a, and the corresponding y value is b. The hydrostatic cut off of the asymptotic line is d.

104.6.8.3 Modified Hyperbolic Drucker Prager

Mapping the Hyperbolic Equation. In the modified hyperbolic Drucker Prager yield surface, the asymptotic line is the original yield surface.

The slope of the asymptotic line is

$$\frac{b}{a} = \sqrt{3}\eta \tag{104.319}$$

The hydrostatic cut-off of the asymptotic line is

$$d = \frac{\xi}{\eta} \tag{104.320}$$

The rounded distance a of the hyperbolic line is a new parameter.

The hyperbolic function is

$$\left(\frac{p + \frac{\xi}{\eta}}{a}\right)^2 - \left(\frac{q}{\sqrt{3}\ a\ \eta}\right)^2 = 1 \tag{104.321}$$

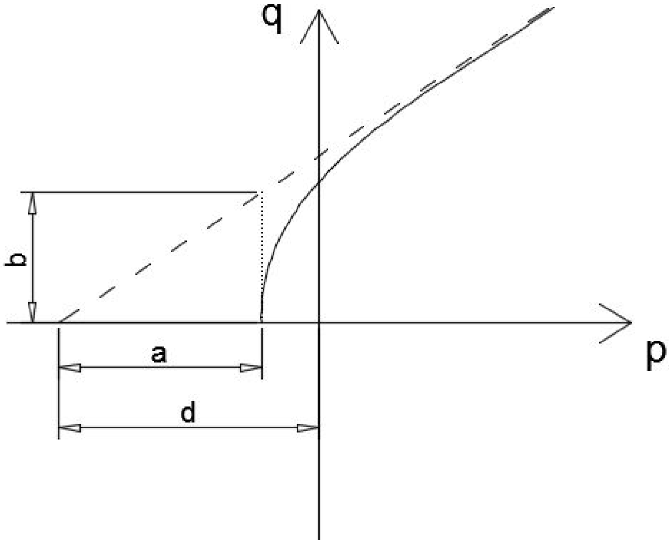


Figure 104.9: The standard hyperbolic function parameters.

Removing the Negative Branch. To remove the negative branch, we need the pressure $(p + \frac{\xi}{\eta})$ always be positive. Therefore,

$$\left(\frac{p + \frac{\xi}{\eta}}{a} \right)^2 = \left(\frac{q}{\sqrt{3}\ a\ \eta} \right)^2 + 1 \tag{104.322}$$

Take the root on both sides,

$$\frac{p + \frac{\xi}{\eta}}{a} = \sqrt{\left(\frac{q}{\sqrt{3}\ a\ \eta} \right)^2 + 1} \tag{104.323}$$

Inside the hyperbolic To make the yield surface value smaller than zero when the stress state is inside the modified cone, the yield surface is

$$0 = \sqrt{\left(\frac{q}{\sqrt{3}\ a\ \eta} \right)^2 + 1} - \frac{p + \frac{\xi}{\eta}}{a} \tag{104.324}$$

Avoiding Zero Denominator. To avoid the zero denominator situation, multiply the equation on both sides by η and a.

$$0 = \sqrt{\left(\frac{q}{\sqrt{3}} \right)^2 + a^2\eta^2} - \eta p - \xi \tag{104.325}$$

The basic yield function The yield function now has one internal variable η for isotropic hardening.

$$\Phi(\sigma,\eta) = \sqrt{\left(\frac{q}{\sqrt{3}}\right)^2 + a^2\eta^2} - \eta p - \xi \tag{104.326}$$

Simplify the yield function and substitute $q = \sqrt{\frac{3}{2}s_{ij}s_{ij}}$, we have

$$\Phi(\sigma,\eta) = \sqrt{\frac{1}{2}s_{ij}s_{ij} + a^2\eta^2} - \eta p - \xi \tag{104.327}$$

Introduce the capability of rotational kinematic hardening by α.

$$\Phi(\sigma,\eta,\alpha) = \sqrt{\frac{1}{2}(s_{ij} - p\alpha_{ij})(s_{ij} - p\alpha_{ij}) + a^2\eta^2} - \eta p - \xi \tag{104.328}$$

104.6.8.4 The Non-Associative Plastic Potential Function.

The non-associative plastic flow direction is

$$m_{ij} = \left(\frac{\partial f}{\partial \sigma_{ij}}\right)^{dev} + \frac{1}{3}\bar{\eta}\delta_{ij} \tag{104.329}$$

where $\bar{\eta}$ controls the plastic flow direction. When $\bar{\eta} = 0$, the material has the deviatoric plastic flow only.

The relation between the dilatancy angle and $\bar{\eta}$ is similar to the relation between the friction angle and η.

$$\bar{\eta} = \frac{3tan\psi}{\sqrt{9 + 12tan^2\psi}} \tag{104.330}$$

where ψ is the dilatancy angle.

Numerical Issues in the Non-Associative Plastic Potential Function. The non-associative plastic flow rule will have asymmetric stiffness matrix, which requires a asymmetric solver. The non-associative requires smaller subincrements for convergence. One important non-associative plastic potential is the purely deviatoric plastic flow.

The purely deviatoric plastic flow may have the numerical issues for the convergence. For example, in Fig 104.10, the purely deviatoric plastic flow direction can never return back to the yield surface. Subincrements can solve this problem.

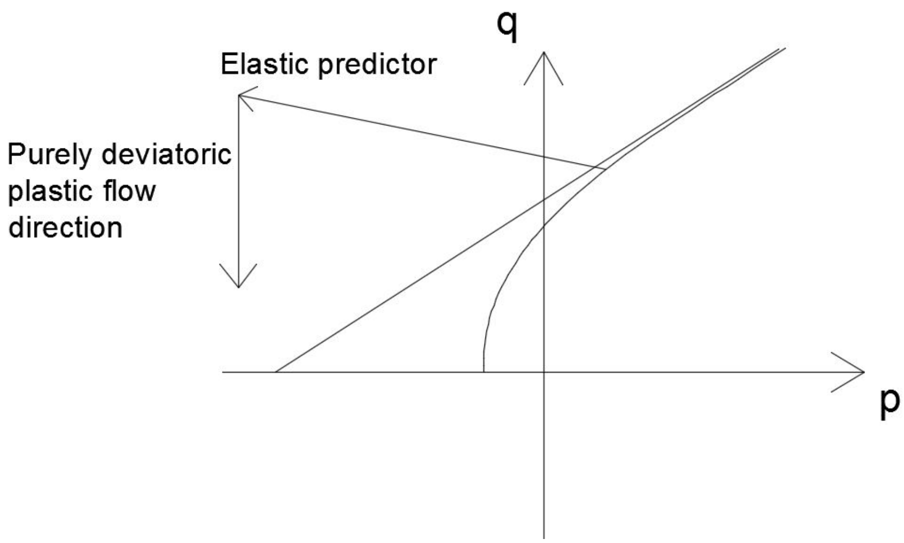


Figure 104.10: The inconvergence situation for purely deviatoric plastic flow.

104.6.8.5 Han's Description of Hyperbolic Drucker–Prager Model with Armstrong–Frederick Kinematic Hardening

Presented is a new description of non-associated hyperbolic Drucker-Prager model with nonlinear Armstrong-Frederick kinematic hardening. It's intended to keep the same theoretical framework and the same set of material parameters as the Drucker-Prager material model shown in section 104.6.7.5.

Classic Drucker–Prager Yield Function with Cohesion This is the yield function shown in Equation 104.237, extended to considering cohesion

$$f = q - Mp - \sqrt{3}\beta \tag{104.331}$$

where

$$M = \frac{6\sin\phi}{3-\sin\phi} \quad \text{and} \quad \beta = \frac{6\cos\phi_0}{\sqrt{3}\,(3-\sin\phi_0)}c \tag{104.332}$$

Note that ϕ is the friction angle, which can evolve if hardening is present in the model, and ϕ_0 is the initial friction angle.

Rewrite Equation 104.333 using the parameters in Real-ESSI implementation

$$f = \sqrt{\left(s_{ij} - p\alpha_{ij}\right)\left(s_{ij} - p\alpha_{ij}\right)} - \sqrt{\frac{2}{3}}kp - \sqrt{2}\beta \tag{104.333}$$

Hyperbolic Drucker–Prager Yield Function Assume the same generic hyperbolic function shown in Equation 104.318, the hyperbolic Drucker-Prager yield function considering isotropic and kinematic hardening is given as

$$f = \sqrt{\left(s_{ij} - p\alpha_{ij}\right)\left(s_{ij} - p\alpha_{ij}\right) + \frac{2}{3}k^2a^2} - \sqrt{\frac{2}{3}}kp - \sqrt{2}\beta \tag{104.334}$$

where a is the rounded distance shown in Figure 104.9.

Derivatives of the yield function Equation 104.334 are shown below.

$$\frac{\partial f}{\partial \sigma_{ij}} = \left[\left(s_{ij} - p\alpha_{ij}\right) + \frac{1}{3}\alpha_{kl}\left(s_{kl} - p\alpha_{kl}\right)\delta_{ij}\right]\left[\left(s_{mn} - p\alpha_{mn}\right)\left(s_{mn} - p\alpha_{mn}\right) + \frac{2}{3}k^2a^2\right]^{-0.5} + \sqrt{\frac{2}{27}}k\delta_{ij} \tag{104.335}$$

$$\frac{\partial f}{\partial \alpha_{ij}} = -p\left(s_{ij} - p\alpha_{ij}\right)\left[\left(s_{mn} - p\alpha_{mn}\right)\left(s_{mn} - p\alpha_{mn}\right) + \frac{2}{3}k^2a^2\right]^{-0.5} \tag{104.336}$$

$$\frac{\partial f}{\partial k} = \frac{2}{3}ka^2\left[\left(s_{mn} - p\alpha_{mn}\right)\left(s_{mn} - p\alpha_{mn}\right) + \frac{2}{3}k^2a^2\right]^{-0.5} - \sqrt{\frac{2}{3}}p \tag{104.337}$$

Plastic Flow The non-associated plastic flow is defined as

$$m_{ij} = \left(\frac{\partial f}{\partial \sigma_{ij}}\right)^{dev} - \frac{1}{3}D\delta_{ij} = \left(s_{ij} - p\alpha_{ij}\right)\left[\left(s_{rs} - p\alpha_{rs}\right)\left(s_{rs} - p\alpha_{rs}\right) + \frac{2}{3}k^2a^2\right]^{-0.5} - \frac{1}{3}D\delta_{ij} \tag{104.338}$$

where

$$D = \xi\left(\sqrt{\frac{2}{3}}k_d - \frac{\sqrt{s_{mn}s_{mn}}}{p}\right) \tag{104.339}$$

The derivatives of the plastic flow are shown below.

$$\begin{aligned}\frac{\partial m_{ij}}{\partial \sigma_{mn}} = &\left(\frac{1}{2}\delta_{im}\delta_{jn} + \frac{1}{2}\delta_{in}\delta_{jm} - \frac{1}{3}\delta_{mn}\delta_{ij} + \frac{1}{3}\delta_{mn}\alpha_{ij}\right)\left[\left(s_{rs} - p\alpha_{rs}\right)\left(s_{rs} - p\alpha_{rs}\right) + \frac{2}{3}k^2a^2\right]^{-0.5} \\ &- \left(s_{ij} - p\alpha_{ij}\right)\left[\left(s_{mn} - p\alpha_{mn}\right) + \frac{1}{3}\alpha_{rs}\left(s_{rs} - p\alpha_{rs}\right)\delta_{mn}\right]\left[\left(s_{tu} - p\alpha_{tu}\right)\left(s_{tu} - p\alpha_{tu}\right) + \frac{2}{3}k^2a^2\right]^{-1.5} \\ &+ \frac{1}{3}\xi\left[s_{mn}p^{-1}(s_{kl}s_{kl})^{-0.5} + \frac{1}{3}\delta_{mn}p^{-2}(s_{ot}s_{ot})^{0.5}\right]\delta_{ij}\end{aligned} \tag{104.340}$$

$$\begin{aligned}\frac{\partial m_{ij}}{\partial \alpha_{mn}} = &-p\left(\frac{1}{2}\delta_{mi}\delta_{nj} + \frac{1}{2}\delta_{mj}\delta_{ni}\right)\left[\left(s_{rs} - p\alpha_{rs}\right)\left(s_{rs} - p\alpha_{rs}\right) + \frac{2}{3}k^2a^2\right]^{-0.5} \\ &+ p\left(s_{ij} - p\alpha_{ij}\right)\left(s_{mn} - p\alpha_{mn}\right)\left[\left(s_{tu} - p\alpha_{tu}\right)\left(s_{tu} - p\alpha_{tu}\right) + \frac{2}{3}k^2a^2\right]^{-1.5}\end{aligned} \tag{104.341}$$

$$\frac{\partial m_{ij}}{\partial k} = -\frac{2}{3}ka^2\left(s_{ij} - p\alpha_{ij}\right)\left(s_{mn} - p\alpha_{mn}\right)\left[\left(s_{tu} - p\alpha_{tu}\right)\left(s_{tu} - p\alpha_{tu}\right) + \frac{2}{3}k^2a^2\right]^{-1.5} \tag{104.342}$$

Linear Isotropic Hardening Linear isotropic hardening is used for this material model. The evolution of the internal variable k is defined as

$$\bar{k} = H m^{equi} \quad , \quad m^{equi} = \left(\frac{2}{3} m_{ij} m_{ij}\right)^{0.5} \tag{104.343}$$

where H is a material constant.

The derivatives of the internal variable k are shown below.

$$\frac{\partial \bar{k}}{\partial \sigma_{ij}} = \frac{2H}{3} m_{kl} \frac{\partial m_{kl}}{\partial \sigma_{ij}} \left(m^{equi}\right)^{-1} \tag{104.344}$$

$$\frac{\partial \bar{k}}{\partial \alpha_{ij}} = \frac{2H}{3} m_{kl} \frac{\partial m_{kl}}{\partial \alpha_{ij}} \left(m^{equi}\right)^{-1} \tag{104.345}$$

$$\frac{\partial \bar{k}}{\partial k} = \frac{2H}{3} m_{kl} \frac{\partial m_{kl}}{\partial k} \left(m^{equi}\right)^{-1} \tag{104.346}$$

Armstrong-Frederick kinematic hardening for Hyperbolic Drucker-Prager

$$\bar{\alpha}_{ij} = \frac{2}{3} \frac{h_a}{p_{atm}} m_{ij}^{dev} - c_r \left(\frac{2}{3} m_{rs}^{dev} m_{rs}^{dev}\right)^{0.5} \alpha_{ij} \tag{104.347}$$

where p_{atm} is the atmospheric pressure of 101.325 kPa. The unit of parameter h_a is Pascal. The parameter c_r is unitless. The α_{ij} in Drucker-Prager is unitless.

Some useful tensor derivatives for Drucker-Prager $\bar{\alpha}$.

$$m_{ij}^{dev} = \left(s_{ij} - p\alpha_{ij}\right) \left[\left(s_{rs} - p\alpha_{rs}\right)\left(s_{rs} - p\alpha_{rs}\right) + \frac{2}{3} k^2 a^2\right]^{-0.5} \tag{104.348}$$

$$\begin{aligned}\frac{\partial m_{ij}^{dev}}{\partial \sigma_{mn}} = & \left(\frac{1}{2}\delta_{im}\delta_{jn} + \frac{1}{2}\delta_{in}\delta_{jm} - \frac{1}{3}\delta_{mn}\delta_{ij} + \frac{1}{3}\delta_{mn}\alpha_{ij}\right) \left[\left(s_{rs} - p\alpha_{rs}\right)\left(s_{rs} - p\alpha_{rs}\right) + \frac{2}{3} k^2 a^2\right]^{-0.5} \\ & - \left(s_{ij} - p\alpha_{ij}\right) \left[\left(s_{mn} - p\alpha_{mn}\right) + \frac{1}{3}\alpha_{rs}\left(s_{rs} - p\alpha_{rs}\right)\delta_{mn}\right] \left[\left(s_{tu} - p\alpha_{tu}\right)\left(s_{tu} - p\alpha_{tu}\right) + \frac{2}{3} k^2 a^2\right]^{-1.5}\end{aligned} \tag{104.349}$$

$$\begin{aligned}\frac{\partial m_{ij}^{dev}}{\partial \alpha_{mn}} = & -p\left(\frac{1}{2}\delta_{im}\delta_{jn} + \frac{1}{2}\delta_{in}\delta_{jm}\right) \left[\left(s_{rs} - p\alpha_{rs}\right)\left(s_{rs} - p\alpha_{rs}\right) + \frac{2}{3} k^2 a^2\right]^{-0.5} \\ & - p\left(s_{ij} - p\alpha_{ij}\right)\left(s_{mn} - p\alpha_{mn}\right) \left[\left(s_{tu} - p\alpha_{tu}\right)\left(s_{tu} - p\alpha_{tu}\right) + \frac{2}{3} k^2 a^2\right]^{-1.5}\end{aligned} \tag{104.350}$$

$$\frac{\partial m_{ij}^{dev}}{\partial k} = -\frac{2}{3}ka^2\left(s_{ij}-p\alpha_{ij}\right)\left(s_{mn}-p\alpha_{mn}\right)\left[\left(s_{tu}-p\alpha_{tu}\right)\left(s_{tu}-p\alpha_{tu}\right)+\frac{2}{3}k^2a^2\right]^{-1.5} \tag{104.351}$$

Tensor derivative of Drucker-Prager $\bar{\alpha}_{ij}$ with respect to σ_{ij}.

$$\frac{\partial \bar{\alpha}_{ij}}{\partial \sigma_{mn}} = \frac{2h_a}{3p_{atm}}\frac{\partial m_{ij}^{dev}}{\partial \sigma_{mn}} - \frac{2}{3}c_r m_{kl}^{dev}\frac{\partial m_{kl}^{dev}}{\partial \sigma_{mn}}\left(\frac{2}{3}m_{rs}^{dev}m_{rs}^{dev}\right)^{-0.5}\alpha_{ij} \tag{104.352}$$

Tensor derivative of Drucker-Prager $\bar{\alpha}_{ij}$ with respect to α_{ij}.

$$\begin{aligned}\frac{\partial \bar{\alpha}_{ij}}{\partial \alpha_{mn}} &= \frac{2h_a}{3p_{atm}}\frac{\partial m_{ij}^{dev}}{\partial \alpha_{mn}} - \frac{2}{3}c_r m_{kl}^{dev}\frac{\partial m_{kl}^{dev}}{\partial \alpha_{mn}}\left(\frac{2}{3}m_{rs}^{dev}m_{rs}^{dev}\right)^{-0.5}\alpha_{ij} \\ &\quad - c_r\left(\frac{2}{3}m_{rs}^{dev}m_{rs}^{dev}\right)^{0.5}\left(\frac{1}{2}\delta_{mi}\delta_{nj}+\frac{1}{2}\delta_{mj}\delta_{ni}\right)\end{aligned} \tag{104.353}$$

Tensor derivative of Drucker-Prager $\bar{\alpha}_{ij}$ with respect to k.

$$\frac{\partial \bar{\alpha}_{ij}}{\partial k} = \frac{2h_a}{3p_{atm}}\frac{\partial m_{ij}^{dev}}{\partial k} - \frac{2}{3}c_r m_{kl}^{dev}\frac{\partial m_{kl}^{dev}}{\partial k}\left(\frac{2}{3}m_{rs}^{dev}m_{rs}^{dev}\right)^{-0.5}\alpha_{ij} \tag{104.354}$$

104.6.9 Rounded Mohr-Coulomb Model

DSL COMMANDS for the rounded Mohr-Coulomb material models are given in section 205.3 on page 827.

The model consists of a smooth conical yield surface. This conical yield surface has an apex that is located at the zero of stress coordinate system or to the extension side, depending on the cohesion characteristic of material in question. The yield surface is defined by:

$$F\left(p,q,\theta,\kappa_{cone}\right) = q\left(1+\frac{q}{q_a}\right)^m g\left(\theta\right) - \eta_{cone}\left(\kappa_{cone}\right)\ \left(p-p_c\right) \tag{104.355}$$

where q_a is a (positive) reference deviator stress, m is a material constant such that $0 \le m \le 1$, controlling the curvature of the cone in the meridian $(p,\ q)$ planes, η_{cone} represents the angle of internal friction, and p_c represents the cohesion. η_{cone} is a function of the hardening variable κ_{cone} which, in turn, is a function of the plastic work. Moreover, an asymmetric trace of the yield surface in the deviatoric plane is generated through the introduction of $g(\theta)$ according to the expression by Willam and Warnke (1974):

$$g\left(\theta\right) = \frac{4\left(1-e^2\right)\cos^2\theta + (2e-1)^2}{2\left(1-e^2\right)\cos\theta + (2e-1)\sqrt{4\left(1-e^2\right)\cos^2\theta + 5e^2 - 4e}} \tag{104.356}$$

and e is an eccentricity parameter that satisfies the condition $1/2 < e \le 1$. From eqn. (104.356) we conclude that $g\left(0\right) = 1/e$ and $g\left(\pi/3\right) = 1$. For $e = 1 \quad \Rightarrow \quad g\left(\theta\right) = 1$, so the influence of the third stress invariant via θ is dropped and the conical surface represents a curved cone[63]. For $e \to 1/2$ the triangular cone[64] is obtained. A nice pictorial representation of function $1/g\,(\theta)$ is presented in the Figure (104.6.9), showing material model traces in the deviatoric plane.

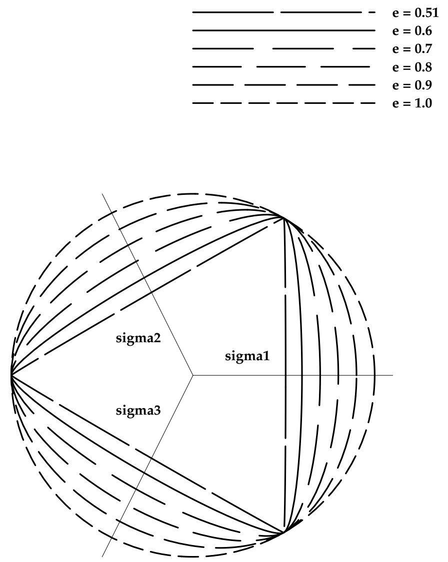


Figure 104.11: Willam Warnke function 1/g for different values of e (e = 0.51, 0.6, 0.7, 0.8, 0.9, 1.0) representing traces of rounded Mohr-Coulomb model in deviatoric space.

104.6.10 Modified Cam-Clay Model

DSL COMMANDS for the Modified Cam Clay material models are given in section 205.3 on page 827.

The pioneering research work on the critical state soil mechanics by the researchers at Cambridge University (Roscoe et al., 1958), (Roscoe and Burland, 1968), (Muir Wood, 1990)) has made great contribution on the modern soil elastoplastic models. The original Cam Clay model (Roscoe et al., 1963), and later the modified Cam Clay model (Schofield and Wroth, 1968) were within the critical state soil mechanics framework. We focus on only the modified Cam Clay model and herein the word 'modified' is omitted to shorten writing.

---

[63] extended Drucker Prager cone.

[64] extended Rankine yield criteria.

104.6.10.1 Critical State

The critical state line (CSL) takes the form

$$e_c = e_{c,r} - \lambda_c \ln p_c \tag{104.357}$$

where e_c is the critical void ratio at the critical mean effective stress p_c [65], $e_{c,r}$ is the reference critical void ratio, λ_c is the normal consolidation slope.

The critical state soil mechanics assumes that the normal consolidation line (NRL) is parallel to the CSL, which is expressed by

$$e = e_\lambda - \lambda \ln p \tag{104.358}$$

where e_λ is the intercept on the NRL at $p = 1$. λ is the normal consolidation slope or the elastoplastic slope of $e - \ln p$ relation, and $\lambda_c = \lambda$.

The unloading-reloading line (URL) take the similar form but with different slope by

$$e = e_\kappa - \kappa \ln p \tag{104.359}$$

where e_κ is the intercept on the URL at $p = 1$. λ_c is the normal consolidation slope or the elastoplastic slope of $e - \ln p$ relation.

104.6.10.2 Elasticity

The elastic bulk modulus K can be directly derived from the Equation 104.359 and takes the form

$$K = \frac{(1+e)\,p}{\kappa} \tag{104.360}$$

If a constant Poisson's ratio ν is assumed, since the isotropic elasticity needs only two material constants, the shear elastic modulus can be obtained in terms of K and ν by

$$G = \frac{3\,(1-2\nu)}{2\,(1+\nu)}K = \frac{3\,(1-2\nu)\,(1+e)}{2\,(1+\nu)\,\kappa}p \tag{104.361}$$

Alternatively, a constant shear elastic modulus G can be assumed and then the Poisson's ratio ν is expressed in terms of K and G as

$$\nu = \frac{3K - 2G}{2\,(G+3K)} \tag{104.362}$$

---

[65] In this chapter, only single-phase (dry phase) is studied, the total and effective stresses are thus identical, e.g. $p_c' = p_c$.

104.6.10.3 Yield Function

The yield function of the Cam Clay model is defined by

$$f = q^2 - M_c^2[p\,(p_0 - p)] = 0 \tag{104.363}$$

where M_c is the critical state stress ration in the $q-p$ plane, and the p_0 is the initial internal scalar variable, which is controlled by the change of the plastic volumetric strain.

The gradient of the yield surface to the stress can be obtained as

$$\frac{\partial f}{\partial \sigma_{ij}} = 2q\frac{\partial q}{\partial \sigma_{ij}} - M_c^2\,(2p - p_0)\,\frac{\partial p}{\partial \sigma_{ij}} = 3s_{ij} + \frac{1}{3}M_c^2\,(p_0 - 2p)\,\delta_{ij} \tag{104.364}$$

where $\partial q/\partial \sigma_{ij}$ and $\partial p/\partial \sigma_{ij}$ are independent of the yield function.

The gradient of the yield surface to p_0 will be used in the integration algorithm, and can be expressed by

$$\frac{\partial f}{\partial p_0} = M_c^2 p \tag{104.365}$$

104.6.10.4 Plastic Flow

The plastic flow of the Cam Clay model is associated with its yield function, in other words, the plastic flow is defined by the potential function, g, which is assumed the same as the yield function, f.

$$g = f = q^2 - M_c^2[p\,(p_0 - p)] = 0 \tag{104.366}$$

The stress gradient to the yield surface can be obtained as

$$m_{ij} = \frac{\partial g}{\partial \sigma_{ij}} = 2q\frac{\partial q}{\partial \sigma_{ij}} + M_c^2\,(2p - p_0)\,\frac{\partial p}{\partial \sigma_{ij}} = 3s_{ij} + \frac{1}{3}M_c^2\,(p_0 - 2p)\,\delta_{ij} \tag{104.367}$$

It can define the plastic dilation angle β, which is related to the ratio of plastic volumetric and deviatoric strain (Muir Wood, 1990), by

$$\tan\beta = -\frac{\Delta\epsilon_v^p}{\Delta\epsilon_q^p} = \frac{M_c^2\,(p_0 - 2p)}{2q} \tag{104.368}$$

It is interesting to find that from Equation 104.368, when $p < p_0/2$, the plastic dilation angle is positive; when $p > p_0/2$, the plastic dilation angle is negative. If $p = p_0/2$, the plastic dilation angle is zero, which is corresponding to the critical state. This is evidently more realistic than Drucker-Prager model, whose associated plastic flow always gives positive plastic dilation angle.

104.6.10.5 Evolution Law

The evolution law of the Cam Clay model is a scalar one, which can be expressed by

$$\dot{p}_0 = \frac{(1+e)\, p_0}{\lambda - \kappa} \dot{\epsilon}_v^p \tag{104.369}$$

With this scalar evolution law, the change of p_0 is decided by the change of plastic volumetric strain. When it reaches the critical state, or when there is no plastic volumetric strain, the evolution of p_0 will cease. From Equation 104.369, one gets

$$\dot{p}_0 = \dot{\lambda} \frac{(1+e)\, p_0}{\lambda - \kappa} m_{ii} \tag{104.370}$$

so if using Equation 104.367 further, one obtains

$$h = \frac{(1+e)\, p_0}{\lambda - \kappa} M_c^2 \,(2p - p_0) \tag{104.371}$$

or using dilation angle,

$$h = \frac{2\,(1+e)\, p_0 q}{\lambda - \kappa} \tan\beta \tag{104.372}$$

104.6.10.6 Yield and Plastic Potential Functions: Cam-Clay Model

Yield function and related derivatives

$$f = q^2 - M_c^2 [p\,(p_0 - p)] = 0 \tag{104.373}$$

$$\begin{aligned} \frac{\partial f}{\partial \sigma_{ij}} &= 2q \frac{\partial q}{\partial \sigma_{ij}} - M_c^2 \,(2p - p_0)\, \frac{\partial p}{\partial \sigma_{ij}} \\ &= 3 s_{ij} + \frac{1}{3} M_c^2 \,(p_0 - 2p)\, \delta_{ij} \end{aligned} \tag{104.374}$$

$$\frac{\partial f}{\partial p_0} = -M_c^2 p \tag{104.375}$$

Plastic flow (associated plasticity) and related derivatives

$$m_{ij} = \frac{\partial f}{\partial \sigma_{ij}} = 3 s_{ij} + \frac{1}{3} M_c^2 \,(p_0 - 2p)\, \delta_{ij} \tag{104.376}$$

$$\begin{aligned} \frac{\partial m_{ij}}{\partial \sigma_{mn}} &= 3 \frac{\partial s_{ij}}{\partial \sigma_{mn}} - \frac{2}{3} M_c^2 \delta_{ij} \frac{\partial p}{\partial \sigma_{mn}} \\ &= 3 \delta_{im} \delta_{jn} - \delta_{ij} \delta_{mn} + \frac{2}{9} M_c^2 \delta_{ij} \delta_{mn} \\ &= 3 \delta_{im} \delta_{jn} + \left(\frac{2}{9} M_c^2 - 1 \right) \delta_{ij} \delta_{mn} \end{aligned} \tag{104.377}$$

$$\frac{\partial m_{ij}}{\partial p_0} = \frac{1}{3} M_c^2 \delta_{ij} \tag{104.378}$$

Isotropic Hardening and related derivatives (CC_Ev) Note the due to the current definition of p (i.e. $p = -\frac{1}{3}\sigma_{ii}$), a minus sign appears in from of the evolution of p_0 as follows:

$$\bar{p}_0 = -\frac{(1+e)\, p_0}{\lambda - \kappa} m_{ii} \tag{104.379}$$

$$= \frac{(1+e)\, p_0}{\lambda - \kappa} M_c^2 \left(2p - p_0\right) \tag{104.380}$$

$$\frac{\partial \bar{p}_0}{\partial \sigma_{ij}} = \frac{(1+e)\, p_0}{\lambda - \kappa} M_c^2 \left(-\frac{2}{3}\delta_{ij}\right) \tag{104.381}$$

$$\frac{\partial \bar{p}_0}{\partial p_0} = \frac{2\,(1+e)}{\lambda - \kappa} M_c^2 \left(p - p_0\right) \tag{104.382}$$

104.6.11 SaniSand2004 (aka Dafalias-Manzari) Model

DSL COMMANDS for the Dafalias-Manzari material models are given in section 205.3 on page 827.

Within the critical state soil mechanics framework, Manzari and Dafalias (1997) proposed a two-surface sand model. This model considered the effects of the state parameter on the behaviors of the dense or loose sands. The features of this model include successfully predicting the softening at the dense state in drained loading, and also softening at the loose state but in the undrained loading. Dafalias and Manzari (2004a) later presented an improved version. This version introduced the fabric dilatancy tensor which has a significant effect on the contraction unloading response. It is also considered the Lode's angle effect on the bounding surface, which produces more realistic responses in non-triaxial conditions. Here only the new version is summarized. The compression stress is assumed negative here, which is different from the original reference by Dafalias and Manzari (2004a).

104.6.11.1 Critical State

Instead of using the most common linear line of critical void ration vs. logarithmic critical mean effective stress, the power relation recently suggested by Li and Wang (1998) was used:

$$e_c = e_{c,r} - \lambda_c \left(\frac{p_c}{P_{at}}\right)^{\xi} \tag{104.383}$$

where e_c is the critical void ratio at the critical man effective stress p'_c, $e_{c,r}$ is the reference critical void ratio, λ_c and ξ (for most sands, $\xi = 0.7$) are material constants, and P_{at} is the atmospheric pressure for normalization.

104.6.11.2 Elasticity

The elastic incremental moduli of shear and bulk, are following Richart et al. (1970):

$$G = G_0 \frac{(2.97 - e)^2}{(1+e)} \left(\frac{p}{P_{at}} \right)^{0.5} P_{at} \;\; ; \;\; K = \frac{2\,(1+\nu)}{3\,(1 - 2\nu)} G \tag{104.384}$$

where G_0 is a material constant, e is the void ratio, and ν is the Poisson's ratio.

The isotropic hypoelasticity is then defined by

$$\dot{e}^e_{ij} = \frac{\dot{s}_{ij}}{2G}, \;\; \dot{\epsilon}^e_v = \frac{\dot{p}}{K} \tag{104.385}$$

104.6.11.3 Yield Function

The yield function is defined by

$$f = |\Lambda| - \sqrt{\frac{2}{3}} mp = 0 \tag{104.386}$$

where s_{ij} is the deviatoric stress tensor, α_{ij} is the deviatoric back stress-ratio tensor, m is a material constant, and

$$|\Lambda| = \left\| s_{ij} - p\alpha_{ij} \right\| = [(s_{ij} - p\alpha_{ij})(s_{ij} - p\alpha_{ij})]^{0.5} \tag{104.387}$$

The gradient of the yield surface to the stress can be obtained as

$$\frac{\partial f}{\partial \sigma_{ij}} = n_{ij} + \frac{1}{3}(\alpha_{pq} n_{pq} + \sqrt{\frac{2}{3}} m)\delta_{ij} \tag{104.388}$$

where $r_{ij} = s_{ij}/p$ is the normalized deviatoric stress tensor, and n_{ij} is the unit gradient tensor to the yield surface defined by

$$n_{ij} = \frac{s_{ij} - p\alpha_{ij}}{|\Lambda|} \tag{104.389}$$

It is evident that $n_{ii} \equiv 0$ and $n_{ij} n_{ij} \equiv 1$.

The gradient of the yield surface to α_{ij} can be easily obtained as

$$\frac{\partial f}{\partial \alpha_{ij}} = -p n_{ij} \tag{104.390}$$

The tensor of n_{ij} is to defined θ_n, the Lode's angle of the yield gradient, by the equation

$$\cos 3\theta_n = -\sqrt{6} n_{ij} n_{jk} n_{ki} \tag{104.391}$$

where $0 \leq \theta_n \leq \pi/6$ and $\theta_n = 0$ at triaxial compression and $\theta_n = \pi/6$ at triaxial extension.

The critical stress ratio M at any stress state can be interpolated between M_c, the triaxial compression critical stress ratio, and M_e, the triaxial extension critical stress ratio.

$$M = M_c g(\theta_n, c), \quad g(\theta_n, c) = \frac{2c}{(1+c) - (1-c)\cos 3\theta_n}, \quad c = \frac{M_e}{M_c} \tag{104.392}$$

The line from the origin of the π plane parallel to n_{ij} will intersect the bounding, critical and dilation surfaces at three 'image' back-stress ratio tensor α^b_{ij}, α^c_{ij}, and α^d_{ij} respectively (Figure 104.12), which are expressed as

$$\alpha^b_{ij} = \sqrt{\frac{2}{3}}[M \exp(-n^b \psi) - m] n_{ij} = \left(\sqrt{\frac{2}{3}} \alpha^b_\theta\right) n_{ij} \tag{104.393}$$

$$\alpha^c_{ij} = \sqrt{\frac{2}{3}}[M - m] n_{ij} = \left(\sqrt{\frac{2}{3}} \alpha^c_\theta\right) n_{ij} \tag{104.394}$$

$$\alpha^d_{ij} = \sqrt{\frac{2}{3}}[M \exp(n^d \psi) - m] n_{ij} = \left(\sqrt{\frac{2}{3}} \alpha^d_\theta\right) n_{ij} \tag{104.395}$$

where $\psi = e - e_c$ is the state parameter; n^b and n^d are material constants.

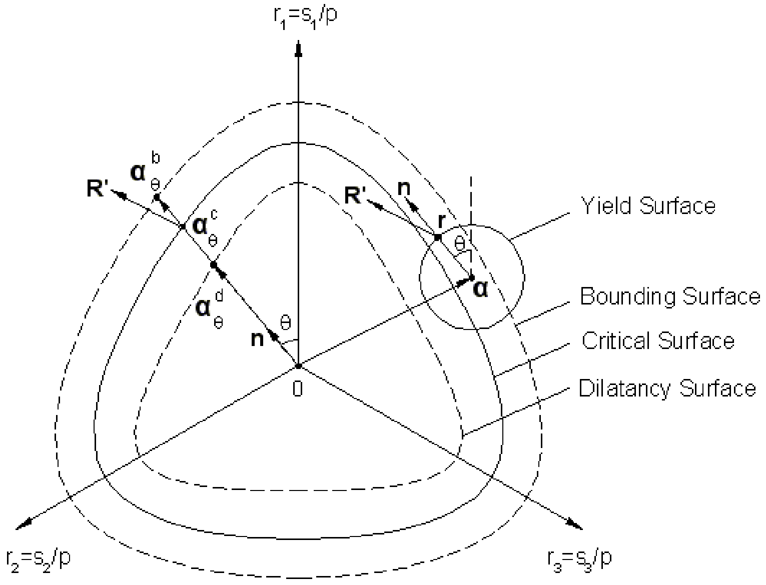


Figure 104.12: Schematic illustration of the yield, critical, dilatancy, and bounding surfaces in the π-plane of deviatoric stress ratio space (after Dafalias and Manzari 2004).

104.6.11.4 Plastic Flow

The plastic strain is given by

$$\dot{\epsilon}^p_{ij} = \dot{\lambda} R_{ij} = \dot{\lambda}(R'_{ij} + \frac{1}{3}D\delta_{ij}) \tag{104.396}$$

The deviatoric plastic flow tensor is

$$R'_{ij} = Bn_{ij} + C(n_{ik}n_{kj} - \frac{1}{3}\delta_{ij}) \tag{104.397}$$

where

$$B = 1 + \frac{3}{2}\frac{1-c}{c}g\cos 3\theta_n, \quad C = 3\sqrt{\frac{3}{2}}\frac{1-c}{c}g \tag{104.398}$$

The volumetric plastic flow part is

$$D = -A_d(\alpha^d_{ij} - \alpha_{ij})n_{ij} = -A_d\left(\sqrt{\frac{2}{3}}\alpha^d_\theta - \alpha_{ij}n_{ij}\right) \tag{104.399}$$

where

$$A_d = A_0(1 + \langle z_{ij}n_{ij}\rangle) \tag{104.400}$$

A_0 is a material constant, and z_{ij} is the fabric dilation tensor. The Macauley brackets $\langle\rangle$ is defined that $\langle x\rangle = x$, if $x > 0$ and $\langle x\rangle = 0$, if $x \leq 0$.

104.6.11.5 Evolution Laws

This model has two tensorial evolution internal variable, namely, the back stress-ratio tensor α_{ij} and the fabric dilation tensor z_{ij}.

The evolution law for the back stress-ratio tensor α_{ij} is

$$\dot{\alpha}_{ij} = \dot{\lambda}[\frac{2}{3}h(\alpha^b_{ij} - \alpha_{ij})] \tag{104.401}$$

with

$$h = \frac{b_0}{(\alpha_{ij} - \tilde{\alpha}_{ij})n_{ij}} \tag{104.402}$$

where $\tilde{\alpha}_{ij}$ is the initial value of α_{ij} at initiation of a new loading process and is updated to the new value when the denominator of Equation 104.402 becomes negative. b_0 is expressed by

$$b_0 = G_0h_0(1 - c_he)\left(\frac{p}{P_{at}}\right)^{-0.5} \tag{104.403}$$

where h_0 and c_h are material constants.

The evolution law for the fabric dilation tensor z_{ij} is

$$\dot{z}_{ij} = -c_z\left\langle\dot{D}\right\rangle(z_{max}n_{ij} + z_{ij}) \tag{104.404}$$

where c_z and z_{max} are material constants.

104.6.11.6 Analytical Derivatives for the Implicit Algorithm

When implemented into an implicit algorithm for the Dafalias-Manzari model, some complicated additional analytical derivatives are needed. This section gives the analytical derivatives expressions based on the tensor calculus.

Analytical expression of $\dfrac{\partial m_{ij}}{\partial \sigma_{kl}}$:

$$\begin{aligned}\frac{\partial m_{ij}}{\partial \sigma_{mn}} &= B\frac{\partial n_{ij}}{\partial \sigma_{mn}} + n_{ij}\frac{\partial B}{\partial \sigma_{mn}} + C\frac{\partial n_{ik}}{\partial \sigma_{mn}} n_{kj} + (n_{ik}n_{kj} - \frac{1}{3}\delta_{ij})\frac{\partial C}{\partial \sigma_{mn}} \\ &\quad + \frac{1}{3}\delta_{ij}\frac{\partial D}{\partial \sigma_{mn}}\end{aligned} \tag{104.405}$$

where

$$\frac{\partial n_{ij}}{\partial \sigma_{mn}} = \frac{1}{|\Lambda|}\left[I^{s}_{ijmn} - \frac{1}{3}\delta_{ij}\delta_{mn} + \frac{1}{3}\alpha_{ij}\delta_{mn} - n_{ij}n_{mn} - \frac{1}{3}(\alpha_{ab}n_{ab})n_{ij}\delta_{mn}\right] \tag{104.406}$$

$$\frac{\partial D}{\partial \sigma_{mn}} = -\frac{\partial A_d}{\partial \sigma_{mn}}\left(\sqrt{\frac{2}{3}}\alpha^{d}_{\theta} - \alpha_{ab}n_{ab}\right) - A_d\left(\sqrt{\frac{2}{3}}\frac{\partial \alpha^{d}_{\theta}}{\partial \sigma_{mn}} - \alpha_{ab}\frac{\partial n_{ab}}{\partial \sigma_{mn}}\right) \tag{104.407}$$

and

$$\frac{\partial B}{\partial \sigma_{mn}} = \frac{3}{2}\left(\frac{1-c}{c}\right)\left(\frac{\partial g}{\partial \sigma_{mn}}\cos 3\theta + g\frac{\partial \cos 3\theta}{\partial \sigma_{mn}}\right) \tag{104.408}$$

$$\frac{\partial C}{\partial \sigma_{mn}} = 3\sqrt{\frac{3}{2}}\left(\frac{1-c}{c}\right)\frac{\partial \alpha^{d}_{\theta}}{\partial \sigma_{mn}} \tag{104.409}$$

$$\frac{\partial \alpha^{d}_{\theta}}{\partial \sigma_{mn}} = M_c \exp\left(n^{d}\psi\right)\left(g n^{d}\frac{\partial \psi}{\partial \sigma_{mn}} + \frac{\partial g}{\partial \sigma_{mn}}\right) \tag{104.410}$$

$$\frac{\partial \psi}{\partial \sigma_{mn}} = -\frac{\xi\lambda_c}{3P_{at}}\left(\frac{p}{P_{at}}\right)^{(\xi-1)}\delta_{mn} \tag{104.411}$$

$$\frac{\partial g}{\partial \sigma_{mn}} = g^{2}\left(\frac{1-c}{2c}\right)\frac{\partial \cos 3\theta}{\partial \sigma_{mn}} \tag{104.412}$$

$$\frac{\partial \cos 3\theta}{\partial \sigma_{mn}} = -3\sqrt{6}\frac{\partial n_{ij}}{\partial \sigma_{mn}}(n_{jk}n_{ki}) \tag{104.413}$$

$$\frac{\partial A_d}{\partial \sigma_{mn}} = A_0 z_{ab}\frac{\partial n_{ab}}{\partial \sigma_{mn}}\left\{\overline{z_{ab}n_{ab}}\right\} \tag{104.414}$$

and define $\left\{\overline{X}\right\} = 1$ if $X > 0$, and $\left\{\overline{X}\right\} = 0$ if $X \leq 0$.

Analytical expression of $\dfrac{\partial m_{ij}}{\partial \alpha_{kl}}$:

$$\begin{aligned}\frac{\partial m_{ij}}{\partial \alpha_{mn}} &= B\frac{\partial n_{ij}}{\partial \alpha_{mn}} + n_{ij}\frac{\partial B}{\partial \alpha_{mn}} + C\frac{\partial n_{ik}}{\partial \alpha_{mn}}n_{kj} + (n_{ik}n_{kj} - \frac{1}{3}\delta_{ij})\frac{\partial C}{\partial \alpha_{mn}} \\ &+\frac{1}{3}\delta_{ij}\frac{\partial D}{\partial \alpha_{mn}}\end{aligned} \tag{104.415}$$

where

$$\frac{\partial n_{ij}}{\partial \alpha_{mn}} = \frac{p}{|\Lambda|}\left(n_{ij}n_{mn} - I^{s}_{ijmn}\right) \tag{104.416}$$

$$\frac{\partial D}{\partial \alpha_{mn}} = -\frac{\partial A_d}{\partial \alpha_{mn}}\left(\sqrt{\frac{2}{3}}\alpha^{d}_{\theta} - \alpha_{ab}n_{ab}\right) - A_d\left(\sqrt{\frac{2}{3}}\frac{\partial \alpha^{d}_{\theta}}{\partial \alpha_{mn}} - n_{mn} - \alpha_{ab}\frac{\partial n_{ab}}{\partial \alpha_{mn}}\right) \tag{104.417}$$

and

$$\frac{\partial B}{\partial \alpha_{mn}} = \frac{3}{2}\left(\frac{1-c}{c}\right)\left(\frac{\partial g}{\partial \alpha_{mn}}\cos 3\theta + g\frac{\partial \cos 3\theta}{\partial \alpha_{mn}}\right) \tag{104.418}$$

$$\frac{\partial C}{\partial \alpha_{mn}} = 3\sqrt{\frac{3}{2}}\left(\frac{1-c}{c}\right)\frac{\partial \alpha^{d}_{\theta}}{\partial \alpha_{mn}} \tag{104.419}$$

$$\frac{\partial \alpha^{d}_{\theta}}{\partial \alpha_{mn}} = M_c \exp\left(n^{d}\psi\right)\frac{\partial g}{\partial \alpha_{mn}} \tag{104.420}$$

$$\frac{\partial g}{\partial \alpha_{mn}} = g^{2}\left(\frac{1-c}{2c}\right)\frac{\partial \cos 3\theta}{\partial \alpha_{mn}} \tag{104.421}$$

$$\frac{\partial \cos 3\theta}{\partial \alpha_{mn}} = -3\sqrt{6}\frac{\partial n_{ij}}{\partial \alpha_{mn}}(n_{jk}n_{ki}) \tag{104.422}$$

$$\frac{\partial A_d}{\partial \alpha_{mn}} = A_0 z_{ab}\frac{\partial n_{ab}}{\partial \alpha_{mn}}\{\overline{z_{ab}n_{ab}}\} \tag{104.423}$$

Analytical expression of $\dfrac{\partial m_{ij}}{\partial z_{mn}}$:

$$\frac{\partial m_{ij}}{\partial z_{mn}} = \frac{1}{3}\delta_{ij}\frac{\partial D}{\partial z_{mn}} \tag{104.424}$$

where

$$\frac{\partial D}{\partial z_{mn}} = -\frac{\partial A_d}{\partial z_{mn}}\left(\sqrt{\frac{2}{3}}\alpha^{d}_{\theta} - \alpha_{ab}n_{ab}\right) \tag{104.425}$$

and

$$\frac{\partial A_d}{\partial z_{mn}} = A_0 n_{mn} \{\overline{z_{ab} n_{ab}}\} \tag{104.426}$$

Analytical expression of $\frac{\partial A_{ij}}{\partial \sigma_{mn}}$:

$$\frac{\partial A_{ij}}{\partial \sigma_{mn}} = \frac{2}{3}\left[\frac{\partial h}{\partial \sigma_{mn}}\left(\sqrt{\frac{2}{3}}\alpha^b_\theta n_{ij} - \alpha_{ij}\right) + \sqrt{\frac{2}{3}}h\left(n_{ij}\frac{\partial \alpha^b_\theta}{\partial \sigma_{mn}} + \alpha^b_\theta \frac{\partial n_{ij}}{\partial \sigma_{mn}}\right)\right] \tag{104.427}$$

where

$$\frac{\partial \alpha^b_\theta}{\partial \sigma_{mn}} = M_c \exp\left(-n^b \psi\right)\left(\frac{\partial g}{\partial \sigma_{mn}} - n^b g \frac{\partial \psi}{\partial \sigma_{mn}}\right) \tag{104.428}$$

$$\frac{\partial h}{\partial \sigma_{mn}} = \frac{1}{(\alpha_{ab} - \alpha^{in}_{ab})n_{ab}}\left[\frac{\partial b_0}{\partial \sigma_{mn}} - h(\alpha_{pq} - \alpha^{in}_{pq})\frac{\partial n_{pq}}{\partial \sigma_{mn}}\right] \tag{104.429}$$

and

$$\frac{\partial b_0}{\partial \sigma_{mn}} = \frac{b_0}{6p}\delta_{mn} \tag{104.430}$$

Analytical expression of $\frac{\partial A_{ij}}{\partial \alpha_{mn}}$:

$$\begin{aligned}\frac{\partial A_{ij}}{\partial \alpha_{mn}} = \frac{2}{3}\Bigg[&\left(\sqrt{\frac{2}{3}}\alpha^b_\theta n_{ij} - \alpha_{ij}\right)\frac{\partial h}{\partial \alpha_{mn}} \\ &+ \sqrt{\frac{2}{3}}h\left(n_{ij}\frac{\partial \alpha^b_\theta}{\partial \alpha_{mn}} + \alpha^b_\theta\frac{\partial n_{ij}}{\partial \alpha_{mn}} - I^s_{ijmn}\right)\Bigg]\end{aligned} \tag{104.431}$$

where

$$\frac{\partial \alpha^b_\theta}{\partial \alpha_{mn}} = M_c \exp\left(-n^b \psi\right)\frac{\partial g}{\partial \alpha_{mn}} \tag{104.432}$$

$$\frac{\partial h}{\partial \alpha_{mn}} = -\frac{h}{(\alpha_{ab} - \alpha^{in}_{ab})n_{ab}}\left[n_{mn} + (\alpha_{pq} - \alpha^{in}_{pq})\frac{\partial n_{pq}}{\partial \alpha_{mn}}\right] \tag{104.433}$$

Analytical expression of $\frac{\partial A_{ij}}{\partial z_{mn}}$:

$$\frac{\partial A_{ij}}{\partial z_{mn}} = \emptyset \tag{104.434}$$

Analytical expression of $\frac{\partial Z_{ij}}{\partial \sigma_{mn}}$:

$$\frac{\partial Z_{ij}}{\partial \sigma_{mn}} = -c_z \left[(z_{max} n_{ij} + z_{ij}) \frac{\partial D}{\partial \sigma_{mn}} + z_{max} D \frac{\partial n_{ij}}{\partial \sigma_{mn}} \right] \{\overline{D}\} \tag{104.435}$$

Analytical expression of $\frac{\partial Z_{ij}}{\partial \alpha_{mn}}$:

$$\frac{\partial Z_{ij}}{\partial \alpha_{mn}} = -c_z \left[(z_{max} n_{ij} + z_{ij}) \frac{\partial D}{\partial \alpha_{mn}} + z_{max} D \frac{\partial n_{ij}}{\partial \alpha_{mn}} \right] \{\overline{D}\} \tag{104.436}$$

Analytical expression of $\frac{\partial Z_{ij}}{\partial z_{mn}}$:

$$\frac{\partial Z_{ij}}{\partial z_{mn}} = -c_z \left(D I^s_{ijmn} + z_{max} n_{ij} \frac{\partial D}{\partial z_{mn}} \right) \{\overline{D}\} \tag{104.437}$$

104.6.12 SaniSand2008 (aka SANISAND) Model

Taiebat and Dafalias (2008)

104.6.13 SANICLAY Model

Dafalias et al. (2006)

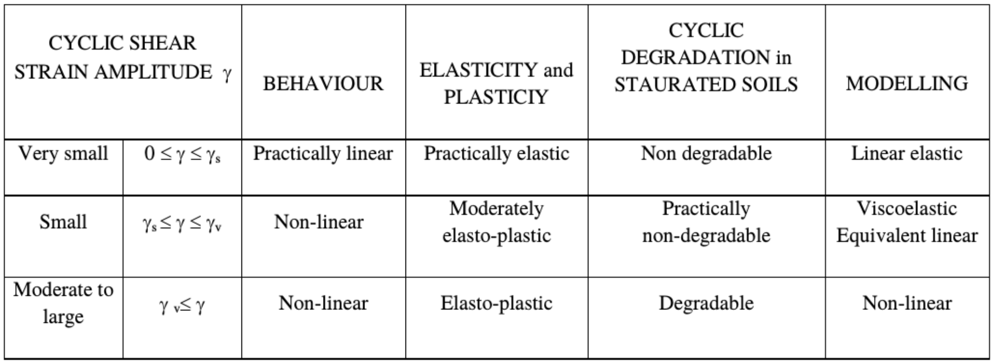
104.6.14 G/G_{max} Modeling

Modeling of stiffness reduction using G/G_{max} curves is frequently used. It is important to note that such modeling is essentially using linear elastic stiffness, reduced stiffness, secant stiffness to model an inelastic, nonlinear process. As such, some important simplifying assumptions are made. As noted by Pecker et al. (2022), using G/G_{max} models is appropriate if certain conditions are met, as shown in Figure 104.13.

104.6.15 Pisanò Elastic-Plastic Model with Vanishing Elastic Region (for G/G_{max} Modeling)

A more recent description of this model is available by Pisanò and Jeremić (2014).

Modeling the mechanical behavior of soils under cyclic/dynamic loading is crucial in most Geotechnical Earthquake Engineering (GEE) applications, including site response analysis and soil structure interaction (SSI) problems. In the last decades, a number of experimental studies (Ishihara, 1996; di Prisco and Wood, 2012) pointed out the complexity of such behavior – especially in the presence of pore



CYCLIC SHEAR STRAIN AMPLITUDE γ		BEHAVIOUR	ELASTICITY and PLASTICIY	CYCLIC DEGRADATION in STAURATED SOILS	MODELLING
Very small	$0 \leq \gamma \leq \gamma_s$	Practically linear	Practically elastic	Non degradable	Linear elastic
Small	$\gamma_s \leq \gamma \leq \gamma_v$	Non-linear	Moderately elasto-plastic	Practically non-degradable	Viscoelastic Equivalent linear
Moderate to large	$\gamma_v \leq \gamma$	Non-linear	Elasto-plastic	Degradable	Non-linear

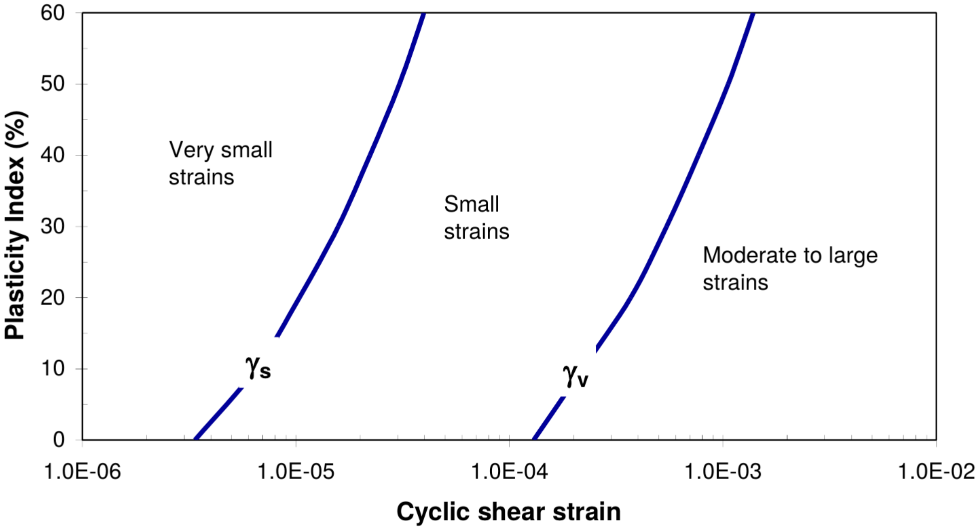


Figure 104.13: Pecker thresholds for G/G_{max} modeling.

fluid(s) – characterized by non-linearity, irreversibility, anisotropy, barotropy, picnotropy, rate-sensitivity, etc. In principle, a comprehensive soil model should be capable of reproducing all the aspects of the mechanical response for any loading condition, as well as predicting the occurrence of liquefaction and cyclic mobility, distinguishing the conditions for shakedown or ratcheting under repeated loads and so forth. However, such an ideal model would probably require too many data for calibration, along with a cumbersome numerical treatment.

Traditionally, many GEE problems are still tackled in the frequency domain through 1D (equivalent) linear models, mainly because of their computational convenience and straightforward calibration. In the light of a Kelvin-Voigt visco-elastic idealization, the dynamic soil behavior is fully described in terms of strain-dependent stiffness degradation (G/G_{max}) and damping (ζ) ratios (Kramer, 1996a). As it holds in the linear regime, the shear and the volumetric responses are assumed to be decoupled, so that G/G_{max} and ζ curves are derived from the experimental cyclic shear tests (triaxial, simple shear or torsional) as a function of the cyclic shear strain amplitude.

Owing to the availability of computer programs for 1D site response analysis (SHAKE (Schnabel et al., 1972), EERA (Bardet et al., 2000), DEEPSOIL (Hashash and Park, 2001)) and SSI problems (SASSI (Lysmer, 1988)), the visco-elastic approach has become more and more popular among practitioners, regardless of drawbacks:

- despite a non-negligible rate-sensitiveness, most energy dissipation in soils derives from frictional inter-granular mechanisms rather than viscous flow (as it is implicitly assumed by using $\mathrm{G/G^{max}}$ and damping approaches);
- G/G_{max} and ζ curves do not allow to evaluate irreversible deformations, nor the influence of pore fluid(s);
- adopting 1D shear constitutive relationships has poor mechanical soundness, since soil behavior exhibits a pronounced deviatoric-volumetric multiaxial coupling;
- the meaning of cyclic shear strain amplitude for the choice of G/G_{max} and ζ values is not evident in the presence of irregular seismic loads.

The above observations justify the need for alternative approaches and more physically consistent soil models. From this standpoint, the incremental elastic-plastic theory represents the main modeling framework, within which significant efforts have been spent in the last decades to simulate the response of cyclically loaded soils. Several approaches have been explored and gradually refined, including e.g. "multi-surface plasticity", "bounding surface plasticity", "generalized plasticity" and "hypoplasticity". Comprehensive overviews on cyclic elasto-plasticity modeling can be found in Lemaitre and Chaboche (1990) and, with specific reference to soils, Prevost and Popescu (1996), Zienkiewicz et al. (1999b) and di Prisco and Wood (2012). In most cases, rotational kinematic hardening formulations have been adopted in conjunction with increasingly accurate flow rules and hardening laws; a number of valuable contributions are worth citing, such as – to mention only a few – Mróz et al. (1978); Prevost (1985a); Borja and Amies (1994); Manzari and Dafalias (1997); Papadimitriou and Bouckovalas (2002); Elgamal et al. (2002); Taiebat and Dafalias (2008); recently, it has been also shown how a good simulation of dynamic properties can be achieved by means of even elastic-perfectly plastic models, as long as formulated in a probabilistic elastic-plastic framework (Sett et al., 2011b). The major issues about the practical use of elastic-plastic models concern the complexity of the mathematical formulations, the computer implementation and the possible high number of material parameters. For a model to appeal to practicing engineers, a tradeoff is needed between the overall accuracy and the number of parameters to be calibrated, particularly provided the frequent lack of detailed *in situ* or laboratory data.

Among the aforementioned models, the one by Borja and Amies (1994) is here taken into special consideration. These authors proposed a total-stress von Mises-type model in the framework of kinematic-hardening bounding surface plasticity, then successfully applied to the seismic simulation of fine-grained deposits at Lotung site in Taiwan (Borja et al., 1999, 2000). Based on work by Dafalias and Popov (1977) and Dafalias (1979), the multiaxial model is characterized by the assumption of vanishing elastic domain, thus implying soil plastification under any load level and a redefinition of the standard loading/unloading criterion. Apart from the mathematical aspects, the model possesses sufficient flexibility to reproduce the undrained dynamic properties of clayey/silty soils, while keeping a minimum number of physically-based parameters.

In this paper similar bounding surface approach with vanishing elastic region is adopted to derive a Drucker-Prager effective-stress model, incorporating pressure sensitivity and non-associativeness, essential ingredients for material modeling of granular materials. As a result, the following constitutive relationship is suitable for the effective-stress time-domain analysis of even liquefiable soils. In addition, the dissipative model performance is here explored in combination with a further viscous mechanism, which can be wisely exploited to improve the simulation of the experimental damping. Although numerical convenience often motivates the embedment of viscous dissipation into elastic-plastic computations, it has a *de facto* physical origin, coming from rate-dependent processes occurring at both inter-granular contacts and grain/pore fluid interfaces.

104.6.15.1 Frictional and viscous dissipative mechanisms

The time-domain finite element (FE) solution of dynamic problems is usually carried out by solving an incremental discrete system of the following form (Bathe, 1982; Zienkiewicz and Taylor, 1991a):

$$\mathbf{M}\Delta\ddot{\mathbf{U}} + \mathbf{C}\Delta\dot{\mathbf{U}} + \mathbf{K}^t\Delta\mathbf{U} = \Delta\mathbf{F}^{ext} \tag{104.438}$$

where Δ and dots stand respectively for step increment and time derivative, $\mathbf{U}$ is the generalized DOF vector (nodal displacement for example), $\mathbf{F}^{ext}$ the nodal external force vector and $\mathbf{M}$, $\mathbf{C}$, $\mathbf{K}^t$ are the mass, damping and (tangent) stiffness matrices, respectively.

In system (104.438) two dissipative sources are readily recognizable, namely the viscous (velocity-proportional) and the frictional (displacement-proportional) terms (Argyris and Mlejnek, 1991). While the latter is given by the variation of the elastic-plastic tangent stiffness $\mathbf{K}^t$, the viscous term related to the damping matrix $\mathbf{C}$ can represent interaction of solid skeleton and pore fluid, and constitutive time-sensitiveness of the soil skeleton. The above combination of frictional and viscous dissipation can be interpreted in terms of two distinct effective stress components acting on the soil skeleton:

$$\sigma_{ij} = \sigma^f_{ij} + \sigma^v_{ij} \tag{104.439}$$

where the effective stress tensor σ_{ij} has been split into frictional (elastic-plastic) and viscous stresses[66]. From a rheological point of view, the resulting scheme can be defined as visco-elastic-plastic – not elastic-viscoplastic – as the elastic-plastic response is rate-independent and accompanied by a parallel viscous resisting mechanism. In what follows, the frictional component is first specified via the formulation of the bounding surface model with vanishing elastic domain; then, the role and the calibration of the linear viscous term is discussed.

Index tensor notation is used, along with the standard Einstein convention for repeated indices; the norm of any second-order tensor x_{ij} is defined as $\|x_{ij}\| = \sqrt{x_{ij}x_{ij}}$, whereas the deviatoric component can be extracted as $x^{dev}_{ij} = x_{ij} - x_{kk}\delta_{ij}/3$ (δ_{ij} is the Kronecker delta). In accordance with usual Solid Mechanics conventions, positive tensile stresses/strains are considered, whereas – as is done in Fluid Mechanics – only the isotropic mean pressure is positive if compressive.

104.6.15.2 Bounding surface frictional model with vanishing elastic domain

The formulated constitutive model represents the frictional effective-stress version of the previous work by Borja and Amies (1994); for the sake of clarity, the presentation sequence of the former publication is here maintained, highlighting both differences and similarities. As was expected, the introduction of pressure-dependence into the constitutive equations implies somewhat more involved derivations, so that the analytical details skipped in this section are reported toward the end in section 104.6.15.7; the superscript f referring to the frictional component of the global effective stress (Equation (104.439)) is avoided for the sake of brevity.

Elastic relationship Provided the usual additive (incremental) strain split into elastic and plastic components $d\epsilon_{ij} = d\epsilon^e_{ij} + d\epsilon^p_{ij}$, the incremental linear elastic Hooke's law is expressed as follows:

$$d\sigma_{ij} = D^e_{ijhk}\left(d\epsilon_{hk} - d\epsilon^p_{hk}\right) \tag{104.440}$$

where d stands for a differentially small increment of strain and D^e_{ijhk} is the fourth-order elastic stiffness tensor. Under the elastic deviatoric/volumetric decoupling, the deviatoric and volumetric counterparts of Equation (104.440) can be also given:

$$ds_{ij} = 2G_{max}\left(de_{hk} - de^p_{hk}\right) \tag{104.441}$$

---

[66] Henceforth, effective stresses are exclusively accounted for

$$dp = -K \left(d\epsilon_{vol} - d\epsilon^{p}_{vol}\right) \tag{104.442}$$

in which $p = -\sigma_{kk}/3$ is the mean stress, $\epsilon_{vol} = \epsilon_{kk}$ is the volumetric strain, $s_{ij} = \sigma^{dev}_{ij}$ is the stress deviator, and $e_{ij} = \epsilon^{dev}_{ij}$ is the strain deviator. The shear modulus $G_{max} = E/2\left(1+\nu\right)$ and the bulk modulus $K = E/3\left(1-2\nu\right)$ are derived from the Young modulus E and the Poisson's ratio ν. Henceforth, G_{max} will be always used for the elastic small-strain shear modulus, whereas the secant cyclic shear stiffness will be referred to as G.

Drucker-Prager yield and bounding loci A conical Drucker-Prager type yield locus is first introduced, similar to what is used by Prevost (1985a) and Manzari and Dafalias (1997):

$$f_y = \frac{3}{2}\left(s_{ij} - p\alpha_{ij}\right)\left(s_{ij} - p\alpha_{ij}\right) - k^2p^2 = 0 \tag{104.443}$$

where α_{ij} is the so called deviatoric back-stress ratio ($\alpha_{kk} = 0$) governing the kinematic hardening of the yield surface; k is a parameter determining the opening angle of the cone. It is also important to note that the variation of the back-stress ratio α_{ij} in (104.443) determines a rigid rotation of the yield locus and, therefore, a rotational kinematic hardening.

The stress derivative of the yield function is also reported for the following developments:

$$\frac{\partial f_y}{\partial \sigma_{ij}} = \left(\frac{\partial f_y}{\partial \sigma_{ij}}\right)^{dev} + \left(\frac{\partial f_y}{\partial \sigma_{ij}}\right)^{vol} = 3\left(s_{ij} - p\alpha_{ij}\right) + \left[\alpha_{hk}\left(s_{hk} - p\alpha_{hk}\right) + \frac{2}{3}k^2p\right]\delta_{ij} \tag{104.444}$$

The yield locus must always reside within the so called bounding surface, here assumed to be a further Drucker-Prager cone (non kinematically hardening, fixed in size):

$$f_B = \frac{3}{2}s_{ij}s_{ij} - M^2p^2 = 0 \tag{104.445}$$

where M provides the bounding cone opening and, as a consequence, the material shear strength (as a function of the mean effective pressure p).

Plastic flow and translation rule When dealing with granular materials, a non-associated plastic flow rule is needed (Nova and Wood, 1979), allowing for plastic contractancy or dilatancy depending on whether loose or dense materials are analyzed. Here, the plastic flow rule is borrowed from Manzari and Dafalias (1997):

$$d\epsilon^p_{hk} = d\lambda \left(n^{dev}_{ij} - \frac{1}{3} D\delta_{ij} \right) \tag{104.446}$$

where $d\lambda$ is the plastic multiplier, n^{dev}_{ij} is a deviatoric unit tensor ($\|n^{dev}_{ij}\| = 1$) and D is a dilatancy coefficient defined as (Manzari and Dafalias, 1997):

$$D = \xi \left(\alpha^d_{ij} - \alpha_{ij} \right) n^{dev}_{ij} = \xi \left(\sqrt{\frac{2}{3}} k_d n^{dev}_{ij} - \alpha_{ij} \right) n^{dev}_{ij} \tag{104.447}$$

in which ξ and k_d are two positive constitutive parameters. While the former controls the amount of volumetric plastic strain, the latter determines the position of the so called "dilatancy surface" and rules the transition from contractive ($D > 0$) to dilative ($D < 0$) behavior.

The kinematic hardening evolution of the yield locus is imposed via the standard Prager translation rule for the (deviatoric) back-stress ratio (Borja and Amies, 1994):

$$d\alpha_{ij} = \|d\alpha_{ij}\| n^{dev}_{ij} \tag{104.448}$$

with both n^{dev}_{ij} and the norm $\|d\alpha_{ij}\|$ to be determined.

Vanishing elastic region and consistency condition As previously mentioned, the most notable feature of the present model concerns the vanishing elastic domain, corresponding with the limit $k \to 0$ in Equation (104.443). Accordingly, the Drucker-Prager cone reduces to its symmetry axis, so that:

$$\lim_{k\to 0} f_y = 0 \Rightarrow \lim_{k\to 0} s_{ij} = p\alpha_{ij} \Rightarrow ds_{ij} = d\alpha_{ij} p + \alpha_{ij} dp \tag{104.449}$$

and, after substituting the Prager rule (104.448) (for more detailed derivation, see section 104.6.15.7):

$$n^{dev}_{ij} = \frac{ds_{ij} - \alpha_{ij} dp}{\|ds_{ij} - \alpha_{ij} dp\|} \tag{104.450}$$

The direction of the deviatoric plastic strain increment n^{dev}_{ij} depends on the variation of both the stress deviator and the mean pressure, which differs from the cohesive version by Borja and Amies (1994). It is also worth noting that purely hydrostatic stress increments ($ds_{ij} = 0$) from initial hydrostatic states ($\alpha_{ij} = 0$) yield $n^{dev}_{ij} = 0$ and thus generates no deviatoric plastic strains.

From a theoretical standpoint, since the direction of the deviatoric plastic strain increment n^{dev}_{ij} depends on $d\sigma_{ij}$, the resulting constitutive formulation can be properly defined as "hypoplastic", this being a spontaneous outcome of the limit operation applied on the elastic region (Dafalias, 1986).

The norm of the back-stress increment in Equation (104.448) is obtained by imposing the standard consistency condition, that is the fulfillment of $df_y = 0$ during plastic loading (section 104.6.15.7):

$$df_y = 0 \Leftrightarrow \frac{\partial f}{\partial \sigma_{ij}} d\sigma_{ij} + \frac{\partial f}{\partial \alpha_{ij}} d\alpha_{ij} = 0 \tag{104.451}$$

whence:

$$\|d\alpha_{ij}\| = \frac{1}{pN^{dev}} \frac{\partial f}{\partial \sigma_{ij}} d\sigma_{ij} \tag{104.452}$$

and $N^{dev} = \| \left(\partial f_y / \partial \sigma_{ij}\right)^{dev} \| = 3\|s_{ij} - p\alpha_{ij}\|$. From Equation (104.452), the norm of $d\alpha_{ij}$ can be further specified for the case of radial loading paths in the deviatoric plane, characterized by the nullity of dp and the coaxiality of s_{ij}, α_{ij} and their increments. After simple manipulations (see section 104.6.15.7) this results in:

$$\|d\alpha_{ij}\| = \sqrt{\frac{2}{3}} \frac{dq}{p} \tag{104.453}$$

where $q = \sqrt{3/2}\|s_{ij}\|$ stands for the usual deviatoric stress invariant.

Hardening relationship and plastic multiplier An incremental hardening relationship is directly established (Borja and Amies, 1994):

$$dq = \sqrt{\frac{2}{3}} H \|de^p_{ij}\| \tag{104.454}$$

where H is the hardening modulus. Then, the substitution of both the flow rule (104.446) and the hardening relationship (104.454) into (104.453) leads to:

$$\|d\alpha_{ij}\| = \frac{2}{3} \frac{H d\lambda}{p} \tag{104.455}$$

By equaling the right–hand sides of Equations (104.441) and (104.449) the following relationship is obtained:

$$2G_{max} \left(de_{ij} - d\lambda n^{dev}_{ij}\right) = \|d\alpha_{ij}\| n^{dev}_{ij} p + \alpha_{ij} dp = \frac{2}{3} \frac{H d\lambda}{p} n^{dev}_{ij} p - \alpha_{ij} K \left(d\epsilon_{vol} + d\lambda D\right) \tag{104.456}$$

whence:

$$d\lambda = \frac{2G_{max}\|de_{ij}\| + Kd\epsilon_{vol}\alpha_{ij}n_{ij}^{dev}}{2G + \frac{2}{3}H - KD\alpha_{ij}n_{ij}^{dev}} \tag{104.457}$$

Equation (104.457) represents the consistent "frictional" generalization of Equation (18) in Borja and Amies (1994), as well as the limit of Equation (12) in Manzari and Dafalias (1997)[67] for a vanishing yield locus size.

Stress projection, hardening modulus and unloading criterion The bounding surface plasticity theory relies on the basic concept that the plastic modulus explicitly depends on the distance between the current stress state and an *ad hoc* stress projection onto the bounding surface. While Borja and Amies (1994) defined a purely deviatoric projection operator, here the whole stress state is involved:

$$\bar{\sigma}_{ij} = \sigma_{ij} + \beta\left(\sigma_{ij} - \sigma_{ij}^{0}\right) \tag{104.458}$$

where β is a scalar distance coefficient and σ_{ij}^{0} embodies the stress state at the last stress reversal (Figure 104.14). The coefficient β must be such that the projected stress $\bar{\sigma}_{ij}$ lies on the bounding surface (Equation (104.445)):

$$\frac{3}{2}\bar{s}_{ij}\bar{s}_{ij} = M^2\bar{p}^2 \tag{104.459}$$

whence, after substituting (104.458) into (104.459), β can be obtained as the positive root of the following second-order algebraic equation:

$$\begin{aligned} &\left[\left\|s_{ij} - s_{ij}^{0}\right\|^2 - \frac{2}{3}M^2\left(p - p_0\right)^2\right]\beta^2 + \\ 2&\left[\left(s_{ij} - s_{ij}^{0}\right)s_{ij} - \frac{2}{3}M^2 p\left(p - p_0\right)\right]\beta + \\ &\left[\|s_{ij}\|^2 - \frac{2}{3}M^2p^2\right] = 0 \end{aligned} \tag{104.460}$$

Apparently, the limit situations $\beta = 0$ and $\beta \to \infty$ correspond with the current stress state being right on the bounding locus or at instantaneous unloading (stress reversal).

In principle, any analytical relationship can be adopted to relate H and β, as long as two fundamental requirements are satisfied, i.e. $H(\beta = 0) = 0$ and $H(\beta \to \infty) \to \infty$: the former ensures the material shear strength to be fully mobilized when the bounding surface is attained; the latter guarantees an

---

[67] Different signs result because of the opposite sign conventions adopted by these authors

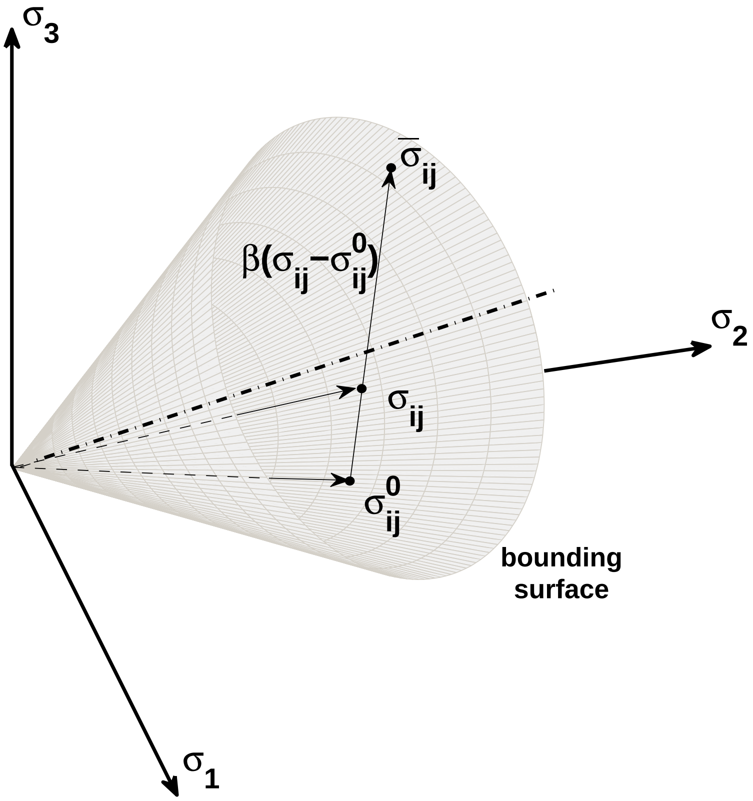


Figure 104.14: Representation of the stress projection onto the bounding surface.

instantaneous elastic stiffness upon any stress reversal, as is explained next. In this case, the expression by Borja and Amies (1994) has been extended to frictional media by accounting for the influence of the mean pressure:

$$H = ph\beta^m \tag{104.461}$$

in which h and m are two additional constitutive parameters.

The last element of the model formulation is the unloading criterion, which in this case is ill-defined due to the lack of the elastic region and the yield surface. The same multiaxial unloading criterion suggested by Borja and Amies (1994) is employed, based on the observation that the hardening modulus H increases at the onset of unloading. Accordingly, as long as $H(\beta)$ is a monotonically increasing function, instantaneous unloading is assumed to occur whenever $dH > 0$, i.e. $d\beta > 0$. The variation of β (and thus its sign) can be readily derived by substituting (104.458) into (104.459), and then differentiating the latter with respect to β:

$$d\beta = -(1+\beta)\,\frac{\bar{s}_{ij} ds_{ij} - \dfrac{2}{3} M^2 \bar{p} dp}{\bar{s}_{ij}\left(s_{ij} - s^0_{ij}\right) - \dfrac{2}{3} M^2 \bar{p}\left(p - p^0\right)} > 0 \tag{104.462}$$

It is worth noting that the variation of β, $d\beta$, plays here the same role of the scalar product $\left(\partial f/\sigma_{ij}\right) d\sigma_{ij}$ in standard elastic-plastic models, i.e. it defines the alternatives of elastic-plastic loading ($d\beta < 0$), neutral loading ($d\beta = 0$) or elastic unloading ($d\beta > 0$). The last key point concerns the update of the stress σ^0_{ij} in Equation (104.458), which must be set equal to the current stress state when $d\beta > 0$ is instantaneously found.

Possible refinements The frictional model has been developed trying to keep the number of material parameters as low as possible, even with a non-associated flow rule. However, it is worth mentioning which kind of improvements might be introduced if required by the problem under examination.

It should be first noted that, as a Drucker-Prager type bounding surface has been adopted, the material shear strength is unaffected by the Lode angle, so that for instance the same failure obliquity is predicted for triaxial compression and extension. This drawback could be easily remedied by modifying the deviatoric cross-section of the bounding surface itself, e.g. by adopting the well known Mohr-Coulomb deviatoric locus or other smooth loci (Matsuoka and Nakai, 1974; Willam and Warnke, 1974; Lade, 1977). A change in the deviatoric cross-section would negligibly influence the overall formulation, as just the evaluation of the projection distance β and of its increment should be modified (Equations (104.459)-(104.460) and (104.462)).

Secondly, the present version of the model cannot predict a possible brittle behavior of the soil, usually taking place in the case of dense materials. Constitutive brittleness could be accounted for by incorporating a further isotropic hardening mechanism at the bounding surface level, allowing for a gradual shrinkage of the outer surface during plastifications.

Another relevant point is about the fact that different parameters must be calibrated for different relative densities of the same granular material, as if distinct materials were indeed considered. As a matter of fact, continuous transitions from loose to dense states (and vice versa) spontaneously take place during straining: this aspect has been successfully addressed and reproduced via the concept of “state parameter” (Been and Jefferies, 1985; Wood et al., 1994; Manzari and Dafalias, 1997), which could be also introduced into a critical-state version of the proposed model.

The above and further refinements – related for instance to non-linear elastic behavior, anisotropy, fabric effects, delayed plastic response, etc. – might result in a more accurate soil model, implying though higher difficulties in terms of calibration, implementation and, as a consequence, practical employment.

104.6.15.3 The role of linear viscous damping

An additional viscous mechanism (Equation (104.439)) can be usually exploited in finite element (FE) analysis, even though it is not directly included in the constitutive model. Indeed, many numerical codes solve discrete systems with a viscous damping term (Equation (104.438)), usually assembled as a linear combination of the mass and the (elastic) stiffness matrices (Rayleigh formulation (Argyris and Mlejnek, 1991; Chopra, 2000)):

$$\mathbf{C} = a_0\mathbf{M} + a_1\mathbf{K}^e \tag{104.463}$$

where a_0 and a_1 are two constant parameters, related to the n^{th} modal damping ratio ζ_n of the discrete structural system.

It could be easily shown that a constitutive viscosity of the form:

$$\sigma^v_{ij} = D^v_{ijhk}\dot{\epsilon}_{hk} \tag{104.464}$$

gives rise to a stiffness-proportional damping matrix, which can be equivalently reproduced through the following calibration of the Rayleigh damping parameters (Borja et al., 2000; Hashash and Park, 2002):

$$a_0 = 0 \qquad a_1 = \frac{2\zeta_0}{\omega} \tag{104.465}$$

The calibration (104.465) establishes the same ratio between tangential/bulk elastic and the viscous moduli, that is $G^e_{max}/G^v_{max} = K^e/K^v$. More importantly, a damping ratio ζ_0 is ensured for a given circular frequency ω, as long as the parallel resisting mechanism (σ^f_{ij}) is purely elastic; as a consequence, provided the a_1 value at the beginning of the analysis, modal frequencies and the corresponding damping ratios are linearly related.

It is also worth remarking some further points about the implications of linear viscous damping in conjunction with non-linear soil models. If a soil element undergoes an imposed shear strain history, the overall shear stress/strain cycles $\tau - \gamma$ differ from the purely frictional component $\tau^f - \gamma$, this difference being due to the viscous shear stress τ^v. As will be shown in next section, the viscous component implies smoother cycles and avoid the sharp transitions at stress reversal usually exhibited by purely elastic-plastic responses (Borja et al., 2000). However, the overall G/G_{max} ratio between the average cyclic stiffness and the elastic shear modulus is unaffected by viscosity.

As far as the damping ratio is concerned, its standard definition (Kramer, 1996a) can be easily adapted to point out the frictional/viscous splitting of the energy ΔW dissipated in a loading cycle:

$$\zeta = \frac{\Delta W}{2\pi G \gamma_{max}^2} = \frac{\Delta W^f + \Delta W^v}{2\pi G \gamma_{max}^2} = \zeta^f + \zeta^v \tag{104.466}$$

where γ_{max} is the imposed cyclic shear strain amplitude and G the corresponding (secant) cyclic shear stiffness. As γ_{max} approaches zero, the plastic dissipation tends to zero as well, so that $\zeta = \zeta^v$; therefore, the Rayleigh parameter a_1 can be calibrated to obtain $\zeta(\gamma_{max} \to 0) = \zeta_0$ for a given circular frequency ω (see Equation (104.465)). This is a desirable feature of the model, as natural soils are well known to dissipate energy at even very small strain amplitudes.

At progressively larger strains, both the frictional and viscous components contribute to the global damping, although the relative quantitative significance is hard to assess *a priori*. In addition, the viscous component of the ζ–γ_{max} curve is not constant, since ζ^v depends on the strain-dependent secant modulus $G(\gamma_{max})$ and, implicitly, on the strain rate. This is different to what has been argued by Borja et al. (2000).

As an example, consider the response of an elastic-perfectly plastic model with additional viscosity under a sinusoidal shear excitation $\gamma(t) = \gamma_{max} \sin(\omega t)$. The simplicity of the elastic-perfectly plastic response allows derivation of instructive analytical formulas for the G/G_{max} and the damping ratios, even in the presence of viscous dissipation. While $\gamma_{max} < \gamma_y$ (yielding shear strain), the material behavior is linear elastic, so that $G/G_{max} = 1$ and ζ equals the purely viscous contribution at $\gamma_{max} = 0$, i.e. $\zeta = \zeta_0$; γ_y depends on the elastic stiffness and the shear strength of the material, $\gamma_y = \tau_{lim}/G_{max}$, where τ_{lim} is the limit (frictional) shear stress for a given confining pressure. For $\gamma_{max} > \gamma_y$ plastifications take place with a flat elastic-perfectly plastic $\tau^f - \gamma$ branch, and the following expressions can be easily derived:

$$\frac{G}{G_{max}} = \frac{\tau_{lim}}{G_{max}\gamma_{max}} \tag{104.467}$$

$$\zeta = \frac{\Delta W^f + \Delta W^v}{2\pi G \gamma_{max}^2} = \underbrace{\frac{2}{\pi}\left(1 - \frac{\tau_{lim}}{G_{max}\gamma_{max}}\right)}_{\zeta^f} + \underbrace{\zeta_0 \frac{G_{max}\gamma_{max}}{\tau_{lim}}}_{\zeta^v} \tag{104.468}$$

In Figure 104.15 the G/G_{max} and ζ ratios are plotted for increasing ζ_0 values. As γ_{max} increases, the frictional damping tends to $2/\pi \approx 0.63$, while the viscous one keeps increasing because of the reduction in the secant stiffness and the increase in the shear strain rate (depending on the strain amplitude). Hence, the value of ζ_0 is to be carefully chosen, in order to avoid excessive dissipation when medium/large strains are induced by the loading process.

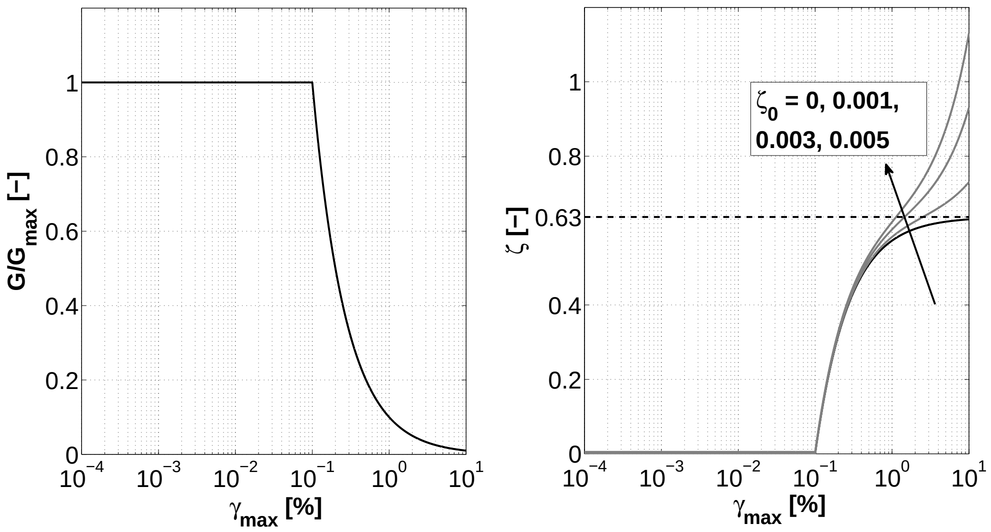


Figure 104.15: G/G_{max} and damping curves for a elastic-perfectly plastic model with linear viscous damping at varying ζ_0 (τ_{lim}=100 kPa, G_{max}=100 MPa)

The fact that the viscous mechanism can modify the purely frictional $\zeta-\gamma_{max}$ curve without altering the cyclic stiffness degradation can be fruitfully exploited to remedy the (frequent) cases in which the experimental-numerical agreement is not satisfactory in terms of energy dissipation properties.

104.6.15.4 Model performance and calibration

The frictional mechanism of the above model is characterized by a rather low number of material parameters, namely the following seven:

- two elastic parameters, the Young modulus E (or the shear modulus G_{max}) and the Poisson's ratio ν;
- the shear strength parameter M for the definition of the bounding surface (Equation (104.445));
- the flow rule parameters, ξ and k_d, governing the increment of the volumetric plastic strain under shearing and the size of the dilatancy surface, respectively (Equation (104.446));
- the hardening parameters h and m for the dependence of the hardening modulus on the distance coefficient β (Equation (104.461)), affecting the pre-failure deformational behavior and, in overall, the resulting dynamic properties (G/G_{max} and damping curves).

Provided a reasonable value for the Poisson's ratio (usually in the range $0.25-0.3$), the small-strain elastic stiffness can be evaluated from dynamic laboratory (RC tests) or *in situ* (seismic geophysical surveys) tests. As far as the shear strength is concerned, the parameter M can be related to the friction angle ϕ as follows:

$$M = \frac{6 \sin \phi}{3 \pm \sin \phi} \tag{104.469}$$

depending on whether triaxial compression (sign – in (104.469)) or extension (sign + in (104.469)) failure conditions are to be reproduced (a change in the deviatoric section of the bounding surface would allow to capture both compressive and extensive limits).

The calibration of the flow rule parameters, ξ and k_d, requires at least a triaxial test to be performed, in order to obtain some information about the volumetric behavior. Figure 104.16 shows the predicted triaxial response for three different values of k_d (and fixed ξ), that is by varying the opening angle of the dilatancy surface (the employed parameters are reported in the figure caption, where p_0 stands for the initial mean pressure).

While the limit stress deviator q is exclusively given by M, the pre-failure behavior is influenced by the plastic deformability and therefore by k_d. The model possesses sufficient flexibility to reproduce contractive, dilative or contractive/dilative behavior; also, such a feature is necessary to reproduce undrained conditions (liquefying and non-liquefying responses), this being a further motivation for non-associativeness when dealing with sandy materials.

Figure 104.17 exemplifies the response predicted under pure shear (PS) cyclic loading, applied as a sinusoidal shear strain history (γ_{max} = $0.2\%, 20\%$, period T=2π s) at constant normal stresses (and thus constant mean pressure p_0 as well. This corresponds with a radial loading path on the deviatoric plane); for the sake of clarity, the volumetric plastic response has been inhibited (ξ=0), in order to evaluate the deviatoric mechanism exclusively. Both purely frictional (solid line) and frictional/viscous (dashed line) responses are plotted.

Owing to the kinematic hardening of the vanished yield locus, the model is capable of reproducing both the Bauschinger and the Masing effects, the latter implying the stabilization of the cyclic response to take place after more than one loading cycle. As expected, the additional viscous damping increases the area of the cyclic loop and therefore the overall dissipated energy; however, the effect of the viscous dissipation becomes significant only at medium-high shear strains, corresponding – for a given loading frequency – with higher strain rates. Further, viscosity causes the aforementioned "smoothing" of stress reversals, as it can be readily noticed in Figure 104.18 by comparing the purely frictional and the frictional/viscous responses.

Given the elastic stiffness and the strength of the soil, the shape of the resulting loading cycles is totally governed by the hardening properties, by h and m in Equation (104.461): this directly affects the simulation of experimental G/G_{max} and damping curves, which can be therefore exploited for the

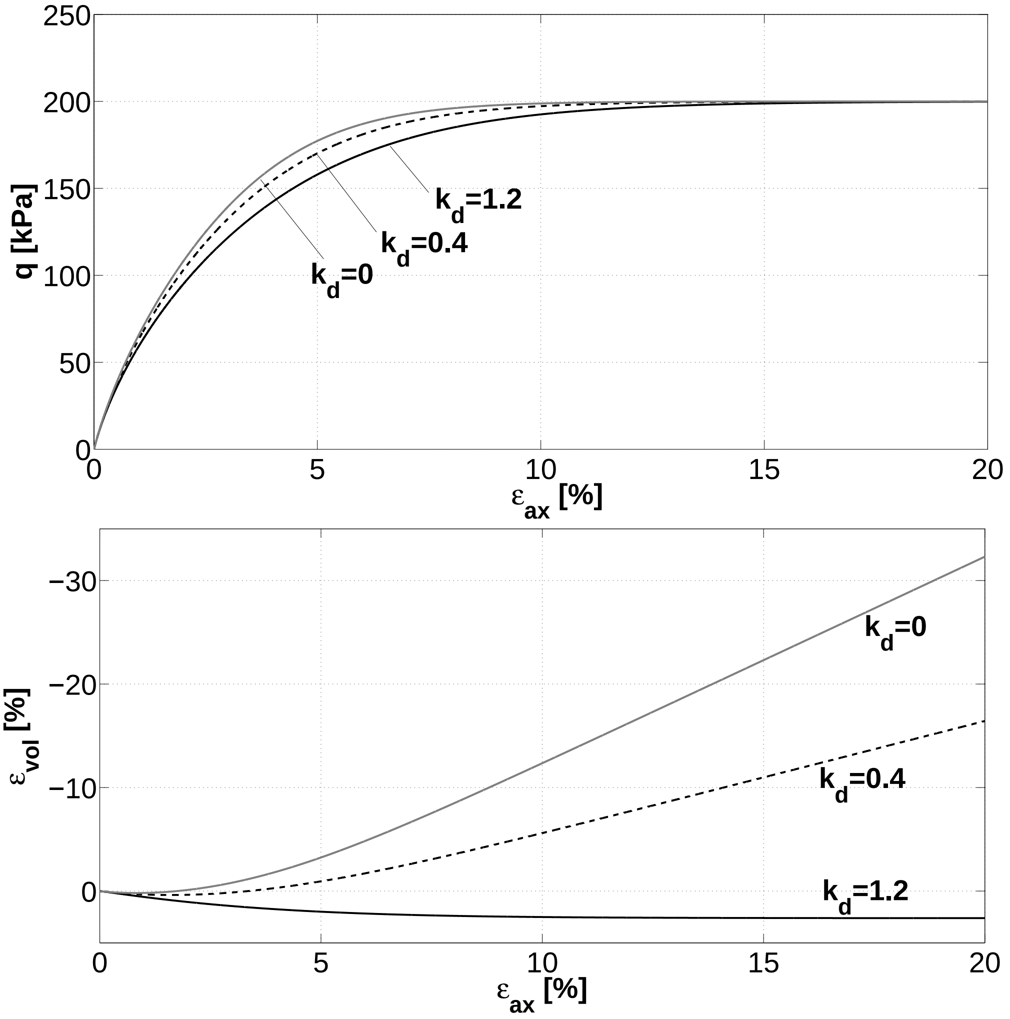


Figure 104.16: Predicted triaxial responses for different dilatancy surfaces (p_0=100 kPa, G_{max} = 4 MPa, ν=0.25, M=1.2, ξ=1, h=$G/(1.5p_0)$, m=1)

calibration of both h and m. As can be easily demonstrated (the proof is given in section 104.6.15.7), the following equality holds under PS loading conditions, i.e. under constant pressure shearing:

$$1 = \frac{G}{G_{max}} \left[1 + \frac{6G_{max}}{hp_0\gamma_{max}} \int_0^{\gamma_{max}} \left(\frac{\gamma}{\tau_{lim}/G - 2\gamma + \gamma_{max}} \right)^m d\gamma \right] \tag{104.470}$$

where $\tau_{lim} = Mp_0/\sqrt{3}$. Relationship (104.470) has been obtained by integrating the constitutive equations over the first loading cycle, and represents the frictional counterpart of Equation (6) in Borja et al. (2000) – as is testified by the explicit influence of the confining pressure p_0. The proper use of Equation (104.470) requires first the choice of two meaningful points on the G/G_{max} experimental curve, i.e. two $(\gamma_{max}, G/G_{max})$ couples; then, the unknowns h and m are obtained by solving the integral system arising

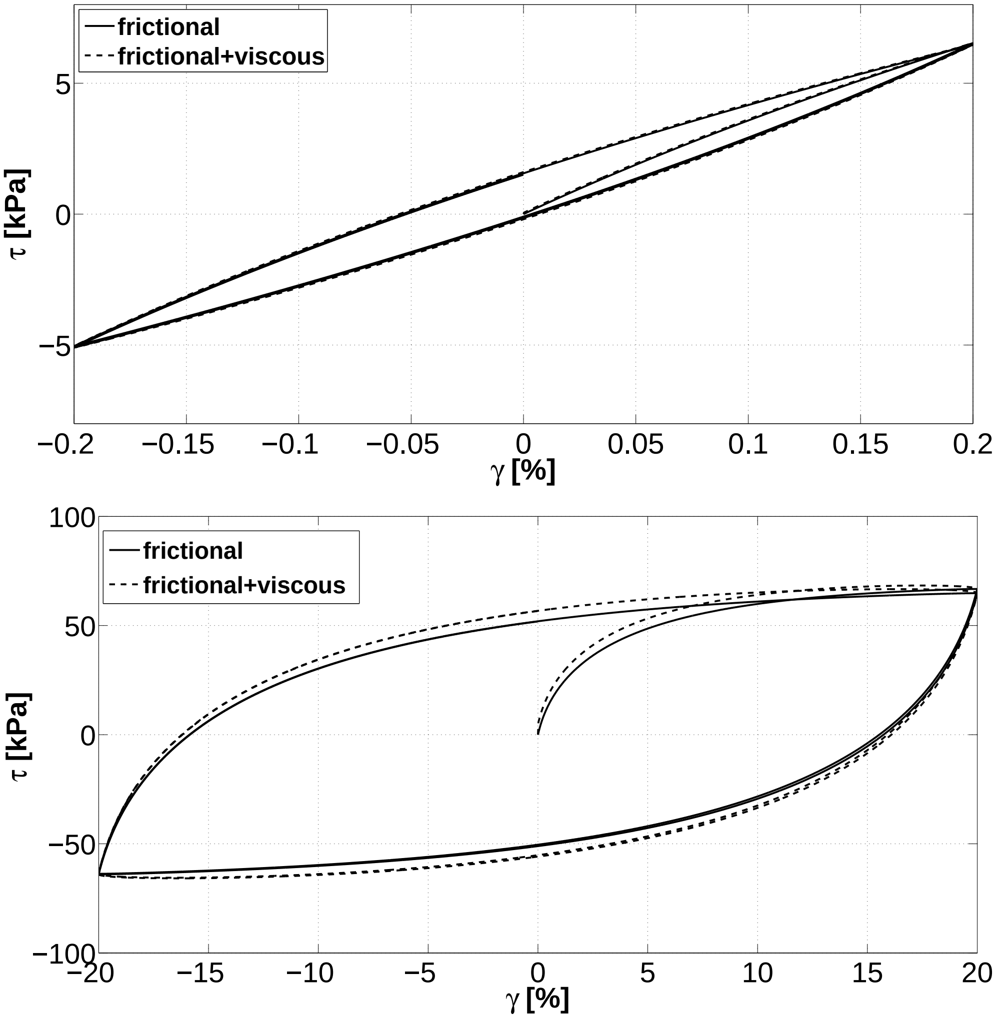


Figure 104.17: Predicted pure shear response at two different shear strain amplitudes (p_0=100 kPa, T=2π s, ζ_0 = 0.003, G_{max}= 4 MPa, ν=0.25, M=1.2, k_d=ξ=0, h=$G/(1.5p_0)$, m=1)

from the specification of Equation (104.470) for both selected $(\gamma_{max}, G/G_{max})$ couples.

Figure 104.19 illustrates the result of the above calibration procedure, applied on the G/G_{max} and ζ curves for sands implemented into the code EERA (Bardet et al., 2000) and formerly obtained by Seed and Idriss (1970b).

Since Equation (104.470) exclusively accounts for the G/G_{max} curve, the very satisfactory agreement in terms of stiffness degradation (viscosity has no effect on it) should not surprise. On the other hand, once h and m are set, the predicted damping curve may or may not match the experimental outcome irrespective of the calibration procedure. In this respect, Figure 104.19 also presents the comparison between the damping curve by Seed and Idriss and the model prediction. The frictional ζ curve lies in the same experimental range, even though the accuracy at γ_{max} = $0.03-1\%$ is not as good as for the

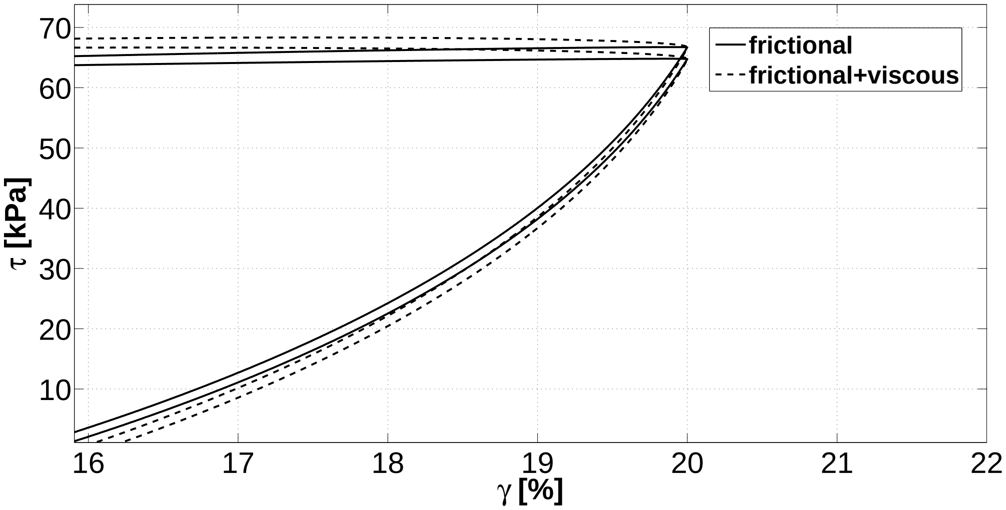


Figure 104.18: Detail of stress reversals for the pure shear response in Figure 104.17 (γ_{max} = 20%)

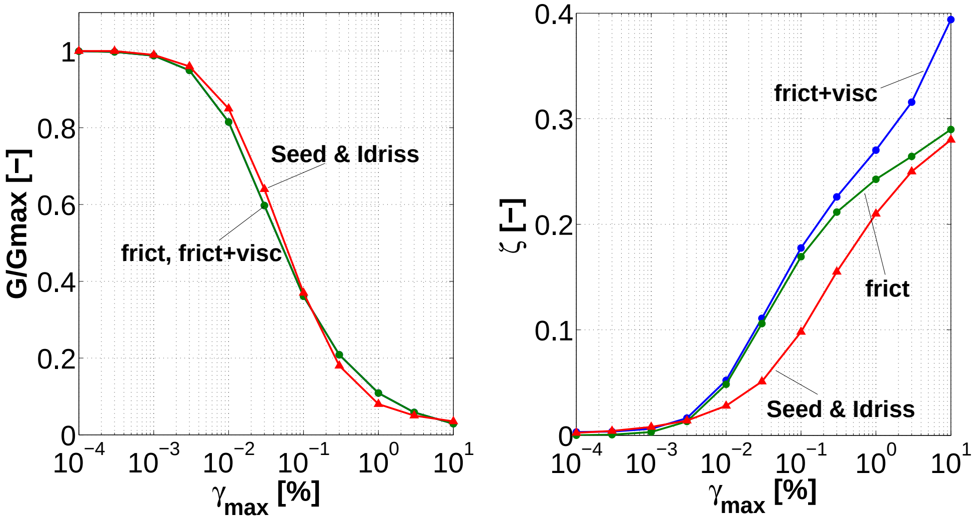


Figure 104.19: Comparison between experimental and simulated G/G_{max} and damping curves (p_0=100 kPa, T=2π s, ζ = 0.003, G_{max} = 4 MPa, ν=0.25, M=1.2, k_d=ξ=0, h=$G/(112p_0)$, m=1.38)

G/G_{max} ratio. In this case, the contribution of the viscous mechanism is practically non-existent, as it only increases the total ζ ratio for $\gamma_{max} > 0.1\%$.

Depending on the specific application, a "trial and error" calibration might be preferable, sacrificing some of the accuracy in terms of G/G_{max} ratio to improve the damping performance. A possible outcome of a manual calibration is plotted in Figure 104.20: apparently, while the simulation of the stiffness curve is still acceptable, the damping curve appears to be much better than the previous one. The use of the viscous mechanism seems to be highly beneficial, since it remedies the lack of accuracy in the frictional curve at medium/large cyclic strains.

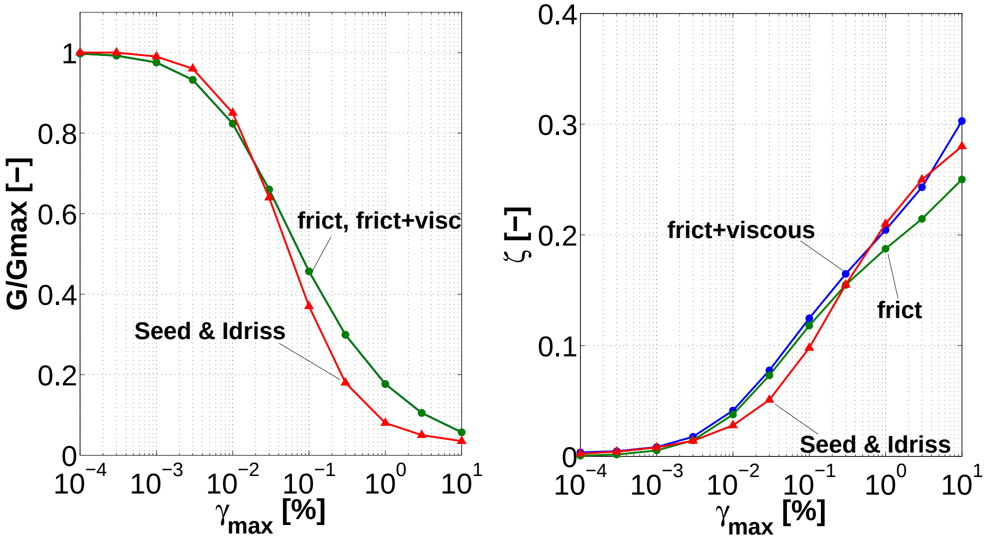


Figure 104.20: Comparison between experimental and simulated G/G_{max} and damping curves (p_0=100 kPa, T=2π s, $\zeta = 0.003$, $G_{max} = 4$ MPa, ν=0.25, M=1.2, k_d=ξ=0, h=$G_{max}/(15 p_0)$, m=1)

It is also worth noting that the experimental/numerical agreement is good up to γ_{max} = 10%, this being a rather high cyclic strain level, for equivalent elastic modeling of soil. In fact, even though the interpretation of experimental cyclic tests is questionable when substantial plasticity occurs, the proposed model produces, within a different framework, the same mechanical response incorporated into traditional equivalent-linear approaches. Besides, if the experimental data under examination are unsatisfactorily reproduced for any h and m combination, the user still has the chance of substituting the interpolation function (104.461) with no further changes in the model formulation.

104.6.15.5 Parametric analysis

In this section the sensitivity of the model predictions to some relevant input parameters is parametrically investigated.

Influence of the confining pressure Figure 104.21 illustrates the sensitivity, under PS loading, of both G/G_{max} and damping frictional curves to the initial confining pressure. As can be noticed, increasing p_0 does enlarge the "pseudo-elastic" range, that is the strain interval within which the deviation by the elastic behavior is negligible even with a vanishing yield locus. It is also noted that the variations in the confining pressure do not imply appreciable changes in the shape of the curves.

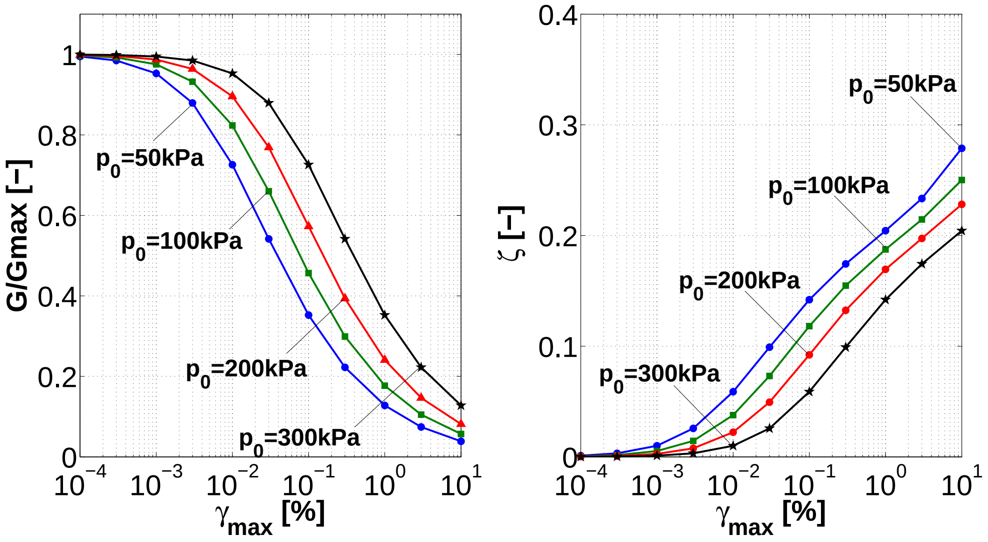


Figure 104.21: Simulated G/G_{max} and damping curves at varying confining pressure (T=2π s, G_{max} = 4 MPa, ν=0.25, M=1.2, k_d=ξ=0, h=$G/(15p_0)$, m=1)

Influence of the hardening parameters Figures 104.22 and 104.23 show the influence of the hardening parameters h and m on the predicted cyclic properties. As the material strain-hardening is accelerated by decreasing either h or m, the pseudo-elastic range tends to disappear, so that $G/G_{max} < 1$ and $\zeta > 0$ at even $\gamma_{max} = 10^{-4}\%$; conversely, an extended pseudo-elastic behavior can be obtained over a large strain range by increasing the hardening parameters. Apparently, the model ensures high flexibility in terms of cyclic curve shapes, so that the response of standard elastic-plastic models (i.e. with non-vanishing elastic region) can be smoothly approximated (compare for instance the $m = 3$ curves in Figure 104.23 and the analytical elastic-perfectly plastic frictional curves in Figure 104.15).

Influence of the viscous mechanism The influence of the viscous parameter ζ_0 on the resulting frictional/viscous damping curve is illustrated in Figure 104.24 (the G/G_{max} is not affected by the parallel viscous mechanism). As was expected, an increase in ζ_0 induce larger values of $\zeta\,(\gamma_{max} \rightarrow 0)$, as well as a faster increase of the ζ curve at medium/high cyclic strains. Figure 104.24 confirms the suitability of the viscous mechanism, as an additional degree of freedom for reproducing the cyclic dissipative soil behavior.

Influence of the volumetric behavior in constrained problems So far, all the simulations have been performed by inhibiting the elastic-plastic soil dilatancy ($\xi = 0$), which in most cases cannot be done to represent real soil behavior. As previously shown for triaxial loading conditions (Figure 104.16), in the absence of kinematic boundary constraints, a variation in the volumetric behavior slightly affects only the

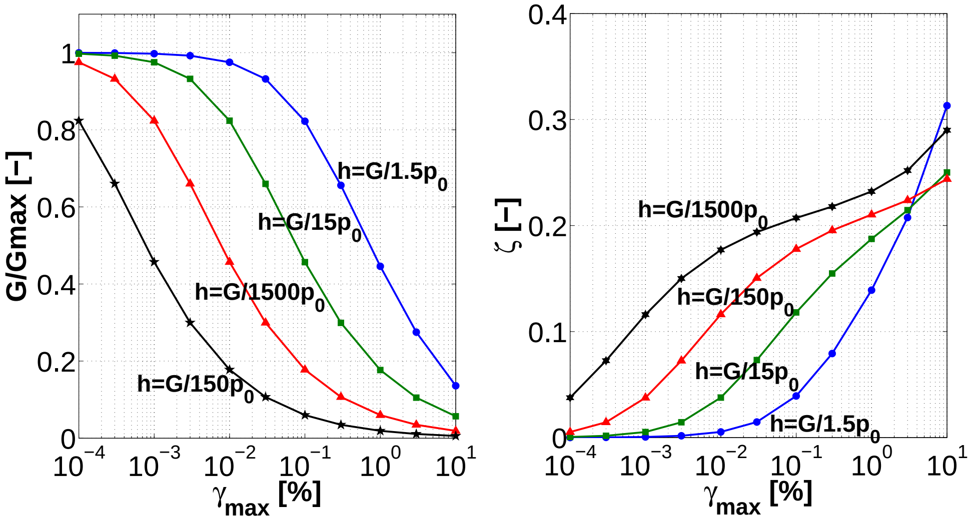


Figure 104.22: Simulated G/G_{max} and damping curves at varying h (p_0=100 kPa, T=2π s, G_{max} = 4 MPa, ν=0.25, M=1.2, k_d=ξ=0, m=1)

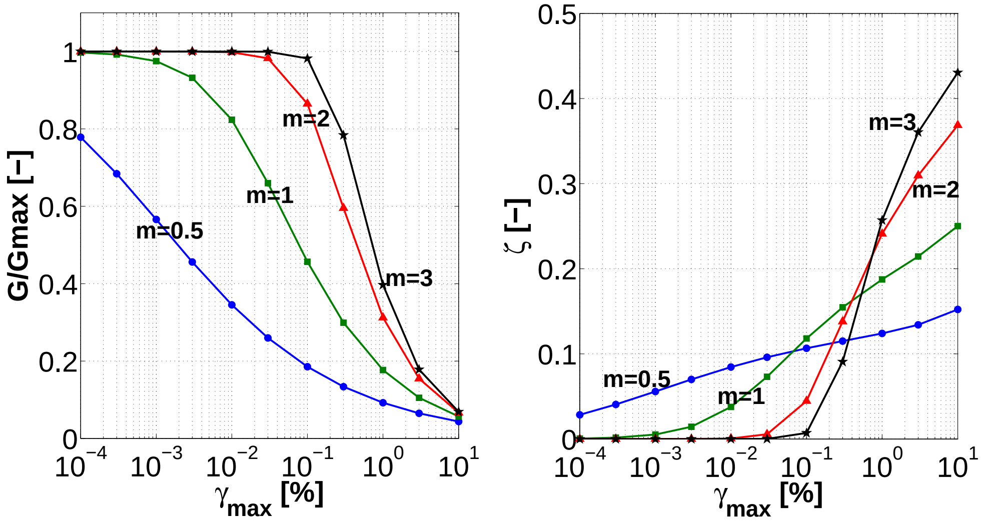


Figure 104.23: Simulated G/G_{max} and damping curves at varying m (p_0=100 kPa, T=2π s, G_{max} = 4 MPa, ν=0.25, M=1.2, k_d=ξ=0, h=$G_{max}/(15p_0)$)

hardening evolution of the stress-strain response toward the limit shear strength; a similar consideration applies to PS loading conditions, since even in this case the normal confinement is statically determined.

However, computational (FE) models contain kinematic constraints arising from certain symmetries (consider plane strain or one-dimensional schemes) (Prevost, 1989; Borja et al., 1999; di Prisco et al., 2012). In addition, for SSI problems, where soil interacts with a (stiff) structural foundations and wall, the soil volume change plays an important role. The presence of kinematic constraints implies that some

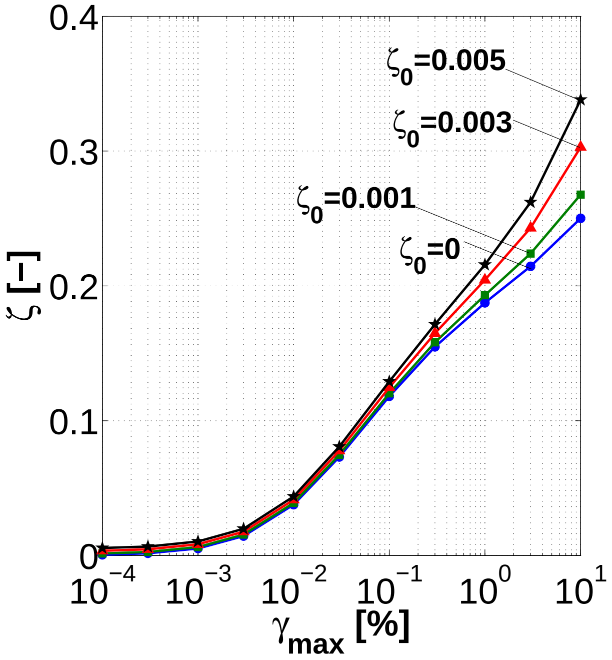


Figure 104.24: Damping curves simulated at varying ζ_0 (p_0=100 kPa, T=2π s, G_{max} = 4 MPa, ν=0.25, M=1.2, k_d=ξ=0, h=$G_{max}/(15p_0)$, m=1)

stress components are to be derived through compatibility conditions (e.g. prevented lateral expansion). That means that the local mean confinement is directly affected by the tendency of the material to dilate or contract. In particular, dilative frictional materials will increase the limit shear stress (with respect to unconfined conditions), while compactive frictional materials will decrease the limit shear stress. Further, not only the limit shear stress, but also the whole pre-failure response depends on the plastic flow rule whenever kinematic constraints are imposed (di Prisco and Pisanò, 2011; di Prisco et al., 2012).

The above considerations suggest that both experimental and numerical results are certainly affected by the kinematics of the system, even though this effect is not easy to be *a priori* quantified in terms of, for instance, G/G_{max} and damping curves. The kinematic conditions of an infinite soil layer during 1D shear wave propagation are experimentally approximated through the well known "simple shear (SS) apparatus" (Wood, 2004), in which the soil specimen is cyclically sheared with no lateral expansion allowed. In order to assess how the kinematic confinement influences the cyclic response, stiffness degradation and damping curves are hereafter simulated under SS conditions by varying the volumetric response of the soil; in particular, three different calibrations of the plastic flow rule (104.446) are considered, namely (i) isochoric ($k_d = \xi = 0$), (ii) compactive ($k_d = M$, $\xi = 1$) and (iii) dilative ($k_d = 0.4$, $\xi = 1$)

The results reported in Figure 104.25 provide an insight into the possible effect of the volumetric response in combination with constrained loading conditions. In the isochoric case, the PS and the SS curves perfectly match (compare e.g. with the p_0 =100 kPa curves in Figure 104.21), as, with no plastic expansion (or contraction), the lateral constraints do not affect the mean pressure during the shear loading; conversely, non-negligible SS-PS differences arise when dilative or contractive materials are considered. As is evident in Figure 104.25, the discrepancy between isochoric and non-isochoric curves becomes evident at medium/high cyclic strains, i.e. at the onset of significant plastifications. Indeed, while the mechanical response is barely inelastic, the deviatoric and the volumetric responses are practically decoupled, so that no variation of the normal confinement takes place.

Apparently, the quantitative significance of the above effects is strictly related to the actual dilational properties and confinement conditions. It can be in general concluded that the cyclic properties are expected to vary depending on specific loading conditions (triaxial, biaxial, simple shear, torsional shear, etc), so that, when numerical models are calibrated, this aspect should be always explicitly considered.

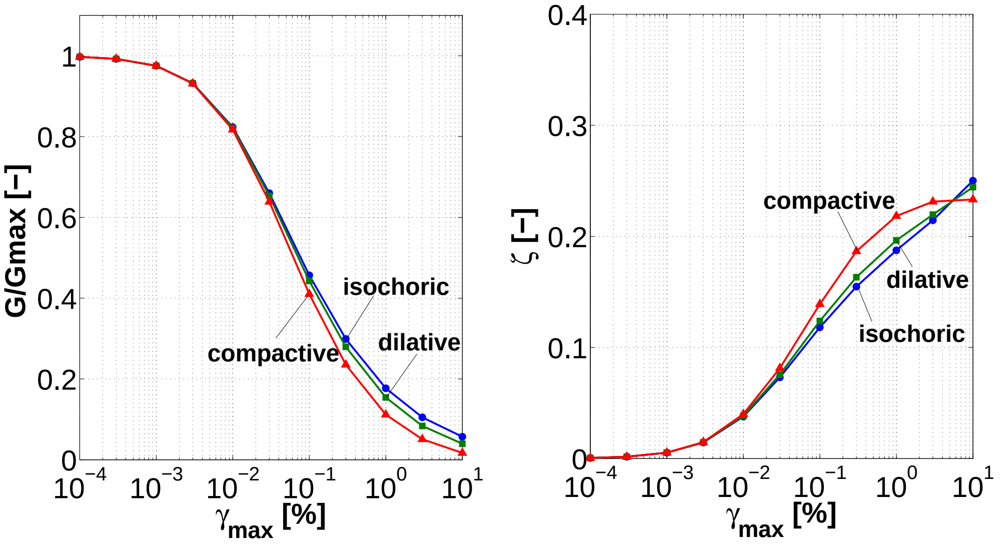


Figure 104.25: G/G_{max} and damping curves simulated under SS conditions and different volumetric responses (p_0=100 kPa, T=2π s, G_{max} = 4 MPa, ν=0.25, M=1.2, k_d=[1.2, 0.4], ξ=[0,1], h=$G_{max}/(15p_0)$, m=1)

104.6.15.6 Concluding remarks

An incremental 3D elastic-plastic constitutive model was developed to reproduce the mechanical response of soils under cyclic/dynamic loading. The model is based on an effective-stress formulation with two parallel dissipative mechanisms, purely frictional (elastic-plastic) and viscous.

As far as the frictional mechanism is concerned, a bounding surface formulation with vanishing elastic region was adopted, extending to the case of pressure-sensitive non-associative soils the previous cohesive model by Borja and Amies (1994) for total-stress analysis. Notable features of the frictional model are: (i) the vanishing yield locus implies an elastic-plastic response at any load levels, as is observed in real experiments; (ii) a minimum number of physically meaningful parameters, which can be easily calibrated on the basis of a few experimental data; (iii) excellent performance and flexibility in reproducing in the elasto-plastic framework the standard stiffness degradation and damping curves. With reference to these latter, the parallel viscous mechanism – easy to be introduced in FE computations – was shown to provide an additional degree of freedom to improve the simulation of the cyclic energy dissipation, as long as the viscous parameter is properly calibrated. As a matter of fact, the viscous mechanism, used here, does physically exist in the form of viscous interaction between the soil solid skeleton and the pore fluid(s), and needs to be taken into account (as for example done here).

Future work will concern the investigation of the model performance in dynamic problems with pronounced hydro-mechanical coupling (cyclic mobility and liquefaction), as well as the comparison of the present model and traditional equivalent-linear approaches in seismic site response and SSI analysis. Further research is also needed to evaluate the accuracy of the model under non-symmetric loading conditions, these being particularly important in seismic slope stability applications.

104.6.15.7 Derivations of Various Equations

Derivation for Equation (104.449) The vanishing size of the yield locus implies:

$$\lim_{k\to 0} s_{ij} = p\alpha_{ij} \Rightarrow ds_{ij} = d\alpha_{ij} p + \alpha_{ij} dp = \|d\alpha_{ij}\| n_{ij}^{dev} p + \alpha_{ij} dp \tag{104.471}$$

and, after substituting the Prager translation rule (104.448):

$$n_{ij}^{dev} = \frac{ds_{ij} - \alpha_{ij} dp}{\|d\alpha_{ij}\| p} = \frac{ds_{ij} - \alpha_{ij} dp}{\|ds_{ij} - \alpha_{ij} dp\|} \tag{104.472}$$

In this last equality the property $\|n_{ij}^{dev}\| = 1$ has been exploited.

Derivations for Equations (104.452) – (104.453) The consistency condition:

$$df_y = 0 \Leftrightarrow \frac{\partial f}{\partial \sigma_{ij}} d\sigma_{ij} + \frac{\partial f}{\partial \alpha_{ij}} d\alpha_{ij} = 0 \tag{104.473}$$

results in the following equality chain

$$\frac{\partial f}{\partial \sigma_{ij}} d\sigma_{ij} = 3p\left(s_{ij} - p\alpha_{ij}\right) \|d\alpha_{ij}\| n_{ij}^{dev} = 3p\left(s_{ij} - p\alpha_{ij}\right) \|d\alpha_{ij}\| \frac{3\left(s_{ij} - p\alpha_{ij}\right)}{N^{dev}} = 0 \tag{104.474}$$

leading to:

$$\|d\alpha_{ij}\| = \frac{\frac{\partial f}{\partial \sigma_{ij}} d\sigma_{ij}}{pN^{dev}} \tag{104.475}$$

where $N^{dev} = \| \left(\partial f_y / \partial \sigma_{ij}\right)^{dev} \| = 3\|s_{ij} - p\alpha_{ij}\|$. The above equation can be further simplified for the case of radial loading paths in the deviatoric plane, characterized by $d\sigma_{ij} = ds_{ij}$ and coaxiality between the current stress state and its increment:

$$\|d\alpha_{ij}\| = \frac{\frac{\partial f}{\partial \sigma_{ij}} d\sigma_{ij}}{pN^{dev}} = \frac{\frac{\partial f}{\partial \sigma_{ij}} ds_{ij}}{pN^{dev}} = \frac{\frac{\partial f}{\partial \sigma_{ij}}^{dev} ds_{ij}}{pN^{dev}} = \frac{n_{ij}^{dev} ds_{ij}}{p} = \frac{s_{ij} ds_{ij}}{p\|s_{ij}\|} \tag{104.476}$$

The final relationship can be re-expressed in terms of standards invariants $q = \sqrt{3/2}\|s_{ij}\|$ and p:

$$\|d\alpha_{ij}\| = \sqrt{\frac{2}{3}} \frac{dq}{p} \tag{104.477}$$

Derivation for Equation (104.470) Under PS loading conditions (constant mean pressure), Equation (104.458) can be reduced to a simpler scalar form:

$$\beta = \frac{\tau_{lim} - \tau}{\tau - \tau_0} \tag{104.478}$$

where $\tau_{lim} = Mp_0/\sqrt{3}$. By exploiting the previous definitions of deviatoric stress and strain invariants, the elastic-plastic response can be expressed as:

$$d\epsilon_d = \frac{dq}{3G_{max}} + \left(\frac{\tau - \tau_0}{\tau_{lim} - \tau}\right)^m \frac{dq}{hp_0} \tag{104.479}$$

and specialized to the case of PS loading:

$$\frac{d\gamma}{\sqrt{3}} = \frac{\sqrt{3} d\tau}{3G_{max}} + \left(\frac{\tau - \tau_0}{\tau_{lim} - \tau}\right)^m \frac{\sqrt{3} d\tau}{hp_0} \tag{104.480}$$

Integration over a strain interval between two stress reversals ($\gamma \in [-\gamma_{max}; \gamma_{max}]$) yields:

$$2\gamma_{max} = \frac{2\tau}{G_{max}} + \frac{3}{hp_0} \int_{-\tau}^{\tau} \left(\frac{\tau' + \tau}{\tau_{lim} - \tau'}\right)^m d\tau' \tag{104.481}$$

where $\tau_0 = -\tau$ has been set. Straightforward variable changes lead to:

$$1 = \frac{G}{G_{max}} + \frac{3}{2hp_0\gamma_{max}} \int_0^{2G\gamma_{max}} \left(\frac{\tau''}{\tau_{lim} - \tau'' + G\gamma_{max}}\right)^m d\tau'' \tag{104.482}$$

$$1 = \frac{G}{G_{max}} \left[1 + \frac{6G_{max}}{hp_0\gamma_{max}} \int_0^{\gamma_{max}} \left(\frac{\gamma}{\tau_{lim}/G - 2\gamma + \gamma_{max}} \right)^m d\gamma \right] \tag{104.483}$$

It is worth highlighting that two approximations are implicitly contained in Equation (104.483): (i) the integration over the first loading cycle does not exactly reproduce the stabilized cyclic response (because of the aforementioned Masing effect); (ii) a symmetric loading cycle in terms of shear strain does not in general ensure the symmetry of the corresponding shear stress range (as it is assumed in Equation (104.481)). However, such approximations do not prevent reasonable values for the hardening parameters h and m to be obtained.

104.6.16 Cosserat Elastoplasticity

Four components of the classical elastoplasticity.

- Elasticity law, E_{ijkl} and C_{ijkl}.
- Yield surface, $f(\sigma, t, \kappa)$
- Plastic flow direction, $m_{ij}^{force} = \frac{\partial Q(\sigma,t,\kappa)}{\partial \sigma_{ij}}$ and $m_{ij}^{curvature} = \frac{\partial Q(\sigma,t,\kappa)}{\partial t}$.
- Hardening law for internal variables, $\kappa = d\lambda h(\sigma, t)$.

where σ is the stress, and t is the couple-stress.

104.6.16.1 Elasticity Law

$$\begin{aligned} E_{ijkl} &= \lambda\delta_{ij}\delta_{kl} + \mu\delta_{ik}\delta_{jl} + (\mu + \chi)\delta_{jk}\delta_{il} \\ C_{ijkl} &= \pi_1\delta_{ij}\delta_{kl} + \pi_2\delta_{ik}\delta_{jl} + \pi_3\delta_{jk}\delta_{il} \end{aligned} \tag{104.484}$$

104.6.16.2 Yield Criterion

Hencky (1924) provided a physical interpretation of von Mises criterion suggesting that yielding begins when the elastic energy of distortion reaches a critical value.[4] So von Mises is called maximum distortion strain energy yield criterion.

In the von-Mises plasticity for the classical elasticity, the distortion extent is measured by

$$J_2 = \frac{1}{2} s_{ij} s_{ij} \tag{104.485}$$

where,

$$s_{ij} = \sigma_{ij} + p\delta_{ij} \tag{104.486}$$

In the von-Mises plasticity for the Cosserat elasticity, the distortion extent de Borst (1993) can be measured by

$$J_2 = \frac{1}{2}(h_1 s_{ij} s_{ij} + h_2 s_{ij} s_{ji} + h_3 t_{kl} t_{kl}/l^2) \tag{104.487}$$

where, $h_1 = 3/4$, $h_2 = -1/4$, and $h_3 = 1/8$. The length l is the characteristic length. The yield strength is

$$\sigma_y = \sqrt{3J_2} \tag{104.488}$$

such that the yield criterion is

$$f(\sigma, t) = \sqrt{3J_2} - k = \sqrt{\frac{3}{2}(h_1 s_{ij} s_{ij} + h_2 s_{ij} s_{ji} + h_3 t_{kl} t_{kl}/l^2)} - k \tag{104.489}$$

where k is a constant for perfectly plasticity, and k is a function of stress and couple-stress for hardening materials.

104.6.16.3 Plastic Flow

The plastic flow direction is

$$\begin{aligned} m_{ij}^{force} &= \frac{\partial Q(\sigma, t, \kappa)}{\partial \sigma_{ij}} \quad , \quad d\sigma_{ij} = d\lambda m_{ij}^{force} \\ m_{ij}^{curvature} &= \frac{\partial Q(\sigma, t, \kappa)}{\partial t} \quad , \quad dt_{ij} = d\lambda m_{ij}^{curvature} \end{aligned} \tag{104.490}$$

In the conventional elastoplasticity, the plastic potential function Q is a function of stress σ and the internal variables κ.

In the Cosserat elastoplasticity, the plastic potential function Q is a function of stress σ, couple-stress t, and the internal variables κ.

104.6.16.4 Hardening Rule

The hardening rule describes how to update the internal variables with the plastic multiplier.

$$dq = d\lambda h(\sigma, t, \kappa) \tag{104.491}$$

In the conventional elastoplasticity, the hardening rule is a function of stress σ and the internal variables κ.

In the Cosserat elastoplasticity, the hardening rule is a function for stress σ, couple-stress t, and the internal variables κ.

104.6.16.5 Forward Euler Algorithm

Forward Euler Algorithm uses the cross point on the yield surface as the starting point to calculate the plastic flow direction and the normal to the yield surface.

104.6.16.6 Explicit Formulation

The governing equation of the forward Euler, explicit algorithm is

$$\begin{aligned}
{}^{n+1}\boldsymbol{\sigma} + \lambda \boldsymbol{C}^{force} \boldsymbol{m}({}^{cross}\boldsymbol{\sigma}, {}^{cross}\boldsymbol{t}, {}^{cross}\boldsymbol{\kappa}) &= {}^{n}\boldsymbol{\sigma} + \boldsymbol{C}^{force} \triangle\boldsymbol{\epsilon} \\
{}^{n+1}\boldsymbol{t} + \lambda \boldsymbol{C}^{curvature} \boldsymbol{m}({}^{cross}\boldsymbol{\sigma}, {}^{cross}\boldsymbol{t}, {}^{cross}\boldsymbol{\kappa}) &= {}^{n}\boldsymbol{t} + \boldsymbol{C}^{curvature} \triangle\boldsymbol{\omega} \\
{}^{n+1}\boldsymbol{\kappa} - \lambda \boldsymbol{h}({}^{cross}\boldsymbol{\sigma}, {}^{cross}\boldsymbol{t}, {}^{cross}\boldsymbol{\kappa}) &= {}^{n}\boldsymbol{\kappa} \\
F({}^{cross}\boldsymbol{\sigma}, {}^{cross}\boldsymbol{t}, {}^{cross}\boldsymbol{\kappa}) &= 0
\end{aligned} \tag{104.492}$$

where ${}^{n}\boldsymbol{\sigma}$, ${}^{n}\boldsymbol{t}$, $\triangle\boldsymbol{\epsilon}$, $\triangle\boldsymbol{\omega}$, and ${}^{n}\boldsymbol{\kappa}$ are known, which represents the stress state, couple-stress state, total strain increment, and internal variables state in the previous step. The four unknowns are ${}^{n+1}\boldsymbol{\sigma}$, ${}^{n+1}\boldsymbol{t}$, ${}^{n+1}\boldsymbol{\kappa}$ and λ, which represents the stress state, internal variables state, and plastic multiplier in the current step.

Continuum Stiffness Tensor The explicit algorithm is derived by starting from the first order Taylor expansion about the starting point (σ_{ij}, t, κ).

$$f^{new} = f^{old} + \frac{\partial f}{\partial \sigma_{ij}} d\sigma_{ij} + \frac{\partial f}{\partial t_{kl}} dt_{kl} + \frac{\partial f}{\partial \kappa} d\kappa \tag{104.493}$$

Since both yield surface values should be zeroes, the Taylor expansion is simplified to

$$\begin{aligned}
&\frac{\partial f}{\partial \sigma_{ij}} d\sigma_{ij} + \frac{\partial f}{\partial t_{kl}} dt_{kl} + \frac{\partial f}{\partial \kappa} d\kappa = 0 \\
&\frac{\partial f}{\partial \sigma_{ij}} (E_{ijkl} d\epsilon - E_{ijkl} d\lambda m^{s}_{kl}) + \frac{\partial f}{\partial t_{kl}} (C_{ijkl} d\omega - C_{ijkl} d\lambda m^{c}_{kl}) + \xi_{*} h_{*} d\lambda = 0 \\
&d\lambda = \frac{\frac{\partial f}{\partial \sigma_{ij}} E_{ijkl} d\epsilon + \frac{\partial f}{\partial t_{kl}} C_{ijkl} d\omega}{\frac{\partial f}{\partial \sigma_{ij}} E_{ijkl} m^{s}_{kl} + \frac{\partial f}{\partial t_{kl}} C_{ijkl} m^{c}_{kl} - \xi_{*} h_{*}}
\end{aligned} \tag{104.494}$$

Simplify the $d\lambda$ expression by $n^{s}_{ij} = \frac{\partial f}{\partial \sigma_{ij}}$ and $n^{c}_{ij} = \frac{\partial f}{\partial t_{ij}}$. So we have

$$d\lambda = \frac{n^{s}_{ij} E_{ijkl} d\epsilon + n^{c}_{ij} C_{ijkl} d\omega}{n^{s}_{ij} E_{ijkl} m^{s}_{kl} + n^{c}_{ij} C_{ijkl} m^{c}_{kl} - \xi_{*} h_{*}} \tag{104.495}$$

104.6.16.7 Backward Euler, Implicit Algorithm

Forward Euler Algorithm uses the final stress state to calculate the plastic flow direction and the normal to the yield surface.

Iterations are required to find the final stress state.

Implicit Formulation The governing equations of the backward Euler, implicit algorithm

$$\begin{aligned} {}^{n+1}\boldsymbol{\sigma} + \lambda \boldsymbol{E}\boldsymbol{m}^s({}^{n+1}\boldsymbol{\sigma}, {}^{n+1}\boldsymbol{t}, {}^{n+1}\boldsymbol{\kappa}) &= {}^{n}\boldsymbol{\sigma} + \boldsymbol{E}\triangle\boldsymbol{\epsilon} \\ {}^{n+1}\boldsymbol{t} + \lambda \boldsymbol{C}\boldsymbol{m}^c({}^{n+1}\boldsymbol{\sigma}, {}^{n+1}\boldsymbol{t}, {}^{n+1}\boldsymbol{\kappa}) &= {}^{n}\boldsymbol{t} + \boldsymbol{C}\triangle\boldsymbol{\omega} \\ {}^{n+1}\boldsymbol{\kappa} - \lambda \boldsymbol{h}({}^{n+1}\boldsymbol{\sigma}, {}^{n+1}\boldsymbol{t}, {}^{n+1}\boldsymbol{\kappa}) &= {}^{n}\boldsymbol{\kappa} \\ F({}^{n+1}\boldsymbol{\sigma}, {}^{n+1}\boldsymbol{t}, {}^{n+1}\boldsymbol{\kappa}) &= 0 \end{aligned} \tag{104.496}$$

where ${}^{n}\boldsymbol{\sigma}$, ${}^{n}\boldsymbol{t}$, $\triangle\boldsymbol{\epsilon}$, $\triangle\boldsymbol{\omega}$, and ${}^{n}\boldsymbol{\kappa}$ are known, which represents the stress state, couple-stress state, total strain increment, and internal variables state in the previous step. The four unknowns are ${}^{n+1}\boldsymbol{\sigma}$, ${}^{n+1}\boldsymbol{t}$, ${}^{n+1}\boldsymbol{\kappa}$ and λ, which represents the stress state, internal variables state, and plastic multiplier in the current step.

Consistent Stiffness Tensor The consistent stiffness tensor for a Cosserat elastoplastic algorithm is extended from the classic elastoplastic algorithmPérez-Foguet et al. (2001). The Jacobian of the backward algorithm can be written as

$$ {}^{n+1}\boldsymbol{J} = \begin{pmatrix} (\boldsymbol{I} + \lambda \boldsymbol{E}\frac{\partial \boldsymbol{m}^s}{\partial \boldsymbol{\sigma}}) & (\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{t}} + \lambda \boldsymbol{E}\frac{\partial \boldsymbol{m}^s}{\partial \boldsymbol{t}}) & \lambda \boldsymbol{E}\frac{\partial \boldsymbol{m}^s}{\partial \boldsymbol{\kappa}} & \boldsymbol{E}\boldsymbol{m}^s \\ (\frac{\partial \boldsymbol{t}}{\partial \boldsymbol{\sigma}} + \lambda \boldsymbol{C}\frac{\partial \boldsymbol{m}^c}{\partial \boldsymbol{\sigma}}) & (\boldsymbol{I} + \lambda \boldsymbol{C}\frac{\partial \boldsymbol{m}^c}{\partial \boldsymbol{t}}) & \lambda \boldsymbol{C}\frac{\partial \boldsymbol{m}^c}{\partial \boldsymbol{\kappa}} & \boldsymbol{C}\boldsymbol{m}^c \\ -\lambda\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\sigma}} & -\lambda\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{t}} & (\boldsymbol{I} - \lambda\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\kappa}}) & -\boldsymbol{h} \\ \boldsymbol{n}_s^T & \boldsymbol{n}_c^T & \boldsymbol{\xi}^T & 0 \end{pmatrix} \tag{104.497}$$

104.6.17 Cosserat von-Mises Elastoplastic Model

Following the fundamental Cosserat elastoplastic material model defined above, an full Cosserat von-Mises plastic model is defined below.

104.6.17.1 Cosserat Plastic Model

Cosserat von-Mises Yield Surface :

$$f(\sigma, t) = \left(\frac{1}{2}s_{ij}s_{ij} + \frac{1}{2}s_{ij}s_{ji} + \frac{1}{2l_p^2}(t_{ij}t_{ij} + t_{ij}t_{ji})\right)^{1/2} - \sqrt{\frac{2}{3}}k \tag{104.498}$$

The normal to the yield surface with respect to the force

$$\begin{aligned} n^f &= \frac{\partial f}{\partial \sigma} \\ &= \frac{1}{2}(s_{ij} + s_{ji})\left(\frac{1}{2}s_{ij}s_{ij} + \frac{1}{2}s_{ij}s_{ji} + \frac{1}{2l_p^2}(t_{ij}t_{ij} + t_{ij}t_{ji})\right)^{-1/2} \end{aligned} \tag{104.499}$$

The normal to the yield surface with respect to the curvature

$$\begin{aligned} n^c =& \frac{\partial f}{\partial t} \\ =& \frac{1}{2l_p^2}(t_{ij} + t_{ji})\left(\frac{1}{2}s_{ij}s_{ij} + \frac{1}{2}s_{ij}s_{ji} + \frac{1}{2l_p^2}(t_{ij}t_{ij} + t_{ij}t_{ji})\right)^{-1/2} \end{aligned} \tag{104.500}$$

Plastic Flow :

$$\begin{aligned} m^f &= n^f \\ m^c &= n^c \end{aligned} \tag{104.501}$$

Hardening Law:

$$\bar{k} = Hm_{equiv} = H\left(\frac{1}{3}m_{ij}^f m_{ij}^f + \frac{1}{3}m_{ij}^f m_{ji}^f + \frac{1}{3}l_p^2(m_{ij}^c m_{ij}^c + m_{ij}^c m_{ji}^c)\right)^{0.5} \tag{104.502}$$

Calculation of dλ : The algorithm is derived by starting from the first order Taylor expansion about the starting point (σ_{ij}, t, κ).

$$f^{new} = f^{old} + \frac{\partial f}{\partial \sigma_{ij}}d\sigma_{ij} + \frac{\partial f}{\partial t_{kl}}dt_{kl} + \frac{\partial f}{\partial \kappa}d\kappa \tag{104.503}$$

Since both yield surface values should be zeros, the Taylor expansion is simplified to

$$\begin{aligned} &\frac{\partial f}{\partial \sigma_{ij}}d\sigma_{ij} + \frac{\partial f}{\partial t_{kl}}dt_{kl} + \frac{\partial f}{\partial \kappa}d\kappa = 0 \\ &\frac{\partial f}{\partial \sigma_{ij}}(E_{ijkl}d\epsilon - E_{ijkl}d\lambda m_{kl}^s) + \frac{\partial f}{\partial t_{kl}}(C_{ijkl}d\omega - C_{ijkl}d\lambda m_{kl}^c) + \xi_* h_* d\lambda = 0 \\ &d\lambda = \frac{\frac{\partial f}{\partial \sigma_{ij}}E_{ijkl}d\epsilon + \frac{\partial f}{\partial t_{kl}}C_{ijkl}d\omega}{\frac{\partial f}{\partial \sigma_{ij}}E_{ijkl}m_{kl}^s + \frac{\partial f}{\partial t_{kl}}C_{ijkl}m_{kl}^c - \xi_* h_*} \end{aligned} \tag{104.504}$$

Simplify the $d\lambda$ expression by $n^s_{ij} = \frac{\partial f}{\partial \sigma_{ij}}$ and $n^c_{ij} = \frac{\partial f}{\partial t_{ij}}$. So we have

$$d\lambda = \frac{n^s_{ij} E_{ijkl} d\epsilon + n^c_{ij} C_{ijkl} d\omega}{n^s_{ij} E_{ijkl} m^s_{kl} + n^c_{ij} C_{ijkl} m^c_{kl} - \xi_* h_*} \tag{104.505}$$

For this Cosserat von-Mises isotropic hardening case,

$$\xi_* = \frac{\partial f}{\partial k} = -\sqrt{\frac{2}{3}} \tag{104.506}$$

$$h_* = \bar{k} = H m_{equiv} \tag{104.507}$$

104.6.17.2 Analysis of Cosserat Elastoplastic Solids

For the elastoplastic analysis, a rectangular plate of 10 m width and 20 m height is horizontally fixed at one end while the other end is subjected to a given horizontal displacement. The geometry and boundary conditions used for the plate are shown in Fig.104.26. In addition, the upper left corner of the plate is vertically fixed. The elastic material constants are $\lambda = 20MPa$, $\mu = 10MPa$, $\chi = 0$, $\pi_1 = 10MN$, $\pi_2 = 0$, $\pi_3 = 1kN$, and the plastic constants are internal length $l_p = 1E-6m$, von-Mises radius $k = 50kPa$, and hardening (softening) rate $H = -500kPa$. For the sake of comparison, the geometry, material properties and loadings used in this problem are the same as the ones used where classical elasticity is employed.

Fig.104.27 illustrates the plastic zone obtained for the problem when the classical and Cosserat theories with different discretization are used. As it is expected, the plastic zone obtained for the classical theory becomes narrowed as the discretization refines. In fact, the plastic zone is two elements wide for all discretization which shows the mesh-dependency of the results based on the classical theory. One the other hand, the plastic zone obtained for the Cosserat theory approximately remains the unchanged when discretization refines.

104.6.18 Accelerated Constitutive Models

One of the concerns of elastic-plastic analysis is the computational effort. In this section developed are closed form elastic-plastic stress increments and elastic-plastic tangent stiffness matrices for commonly used elastic-plastic models. This is achieved by multiplying, in closed form, explicit, forward Euler elastic-plastic stress incremental solution and explicit, tangent elastic-plastic stiffness tensor, as developed in section 104.3 on page 206.

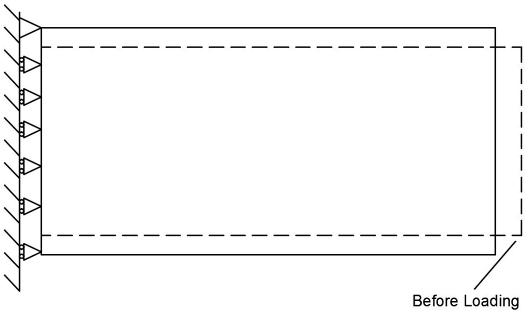


Figure 104.26: Loading Condition and Boundary Conditions of the Cosserat Elastoplastic Model.

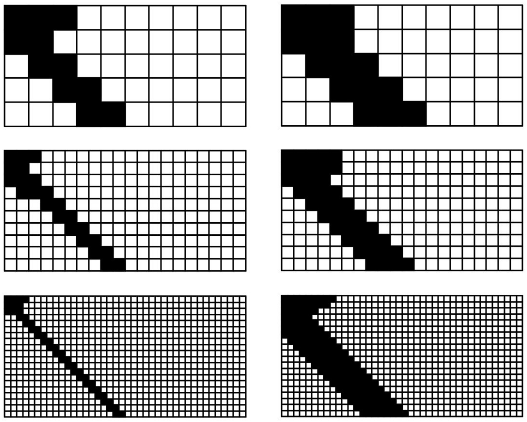


Figure 104.27: Comparison of the Plastic Zone of the Cosserat Elastoplastic Model. Left side is the classical elastoplastic material model, where the mesh refinement leads to localization of the plastic zone. Right side is the Cosserat elastoplastic material model, where the plastic zone is independent of the mesh sizes.

Relation between stress increment $d\sigma_{ij}$ and strain increments $d\epsilon_{kl}$ can be written for linear isotropic elasticity as:

$$d\sigma_{ij} = E_{ijkl} d\epsilon_{kl} \tag{104.508}$$

where E_{ijkl} is the fourth order elastic stiffness tensor. The elastic stiffness tensor features both minor symmetry $E_{ijkl} = E_{jikl} = E_{ijlk}$ and major symmetry $E_{ijkl} = E_{klij}$ (Jeremić and Sture (1997)). The elastic stiffness tensor for isotropic material can be written as:

$$E_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu \left(\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk} \right) \tag{104.509}$$

where λ and μ are the Lamé constants and E and ν are Young's Modulus and Poisson's ratio, respectively:

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \quad ; \quad \mu = \frac{E}{2(1+\nu)} \tag{104.510}$$

The relation between the stress tensor increment $d\sigma_{ij}$ and strain tensor increment $d\epsilon_{kl}$ is:

$$d\epsilon_{kl} = D_{klpq} d\sigma_{pq} \tag{104.511}$$

where D_{klpq} is the elastic compliance fourth order tensor, defined as:

$$D_{klpq} = \frac{-\lambda}{2\mu\,(3\lambda + 2\mu)}\,\delta_{kl}\delta_{pq} + \frac{1}{4\mu}\left(\delta_{kp}\delta_{lq} + \delta_{kq}\delta_{lp}\right) \tag{104.512}$$

104.6.18.1 Elasto-plasticity

Using developments from section 104.3 on page 206, one can write

$$d\sigma_{ij} = E_{ijkl} d\epsilon^{e}_{kl} \tag{104.513}$$

$$d\epsilon^{p}_{ij} = d\lambda\,\frac{\partial Q}{\partial \sigma_{ij}} = d\lambda\; m_{ij}(\sigma_{ij}, q) \tag{104.514}$$

$$d\epsilon_{ij} = d\epsilon^{e}_{ij} + d\epsilon^{p}_{ij} \tag{104.515}$$

$$dq_{*} = d\lambda\; h(\tau_{ij}, q) \tag{104.516}$$

where, ϵ_{ij}, ϵ^{e}_{ij}, and ϵ^{p}_{ij} are total, elastic, and plastic strain tensors respectively, σ_{ij} is stress tensor, and q_{*} represents internal variables. Moreover, m_{ij} is the plastic flow direction, h is the plastic moduli, and $d\lambda$ is a plastic constistency parameter that is to be determined. Equation (104.513) is the Hooke's law that relates stress to elastic strain using the stiffness tensor E_{ijkl}. Equation(104.514) shows relations of plastic strains to the associated or non-associated plastic flow rule. Equation (104.515) represents the additive decomposition of strain increment into elastic strain increment and plastic strain increments, that is applicable for small deformation analysis. Equation (104.516) represents evolution law for internal variables.

104.6.18.2 Elastic-Plastic Constitutive Models

For each elastic-plastic constitutive model, there are four components to be specified. Those components are (a) elasticity relation, (b) yield function, (c) plastic flow function, and (d) hardening and/or softening laws.

Once those four components are chosen, specified, elastic-plastic stiffness tensor, developed earlier in section 104.3 on page 206, can be developed for each model, by analytically multiplying functions and tensor equations for each component. In following sections this is done for von Mises, Drucker-Prager, and Cam-Clay material models.

The tanget elastic-plastic tensor is written as

$$E^{el-pl}_{pqmn} = E_{pqmn} - \frac{E_{pqkl} \, {}^{n}m_{kl} \, {}^{n}n_{ij} E_{ijmn}}{{}^{n}n_{ot} E_{otrs} \, {}^{n}m_{rs} - \xi_* \; h_*} \tag{104.517}$$

For perfectly plastic materials ξ_* and h_* would be zero while for the cases with evolution laws, the appropriate evolution laws should be used in derivation of tangent stiffness E^{el-pl}_{pqmn}. Elastic modulus tensor is written as:

$$E_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu(\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}) \tag{104.518}$$

where $\lambda = \nu E/(1 + \nu)(1 - 2\nu)$, $\mu = E/2(1 + \nu)$.

104.6.18.3 von Mises Model

von Mises yield criteria can be written as

$$f = \left[\left(s_{ij} - \alpha_{ij} \right) \left(s_{ij} - \alpha_{ij} \right) \right]^{0.5} - \sqrt{\frac{3}{2}} k = 0 \tag{104.519}$$

where k is the scalar internal variable where initial value is related to the uniaxial tension strength, α_{ij} is the tensorial internal variable, the so called back stress that controls translational kinematic hardening.

The stress derivative of the yield function is

$$n_{ij} = \frac{\partial f}{\partial \sigma_{ij}} = \frac{1}{\sqrt{(s_{ij} - \alpha_{ij})(s_{ij} - \alpha_{ij})}} (s_{ij} - \alpha_{ij}) \tag{104.520}$$

In case of associated plastic flow rule where the plastic potential function is considered to be the same of yield function, stress derivative of plastic flow function is:

$$m_{ij} = \frac{\partial g}{\partial \sigma_{ij}} = \frac{1}{\sqrt{(s_{ij} - \alpha_{ij})(s_{ij} - \alpha_{ij})}} (s_{ij} - \alpha_{ij}) \tag{104.521}$$

In order to find the closed form equation for elastic-plastic modulus for case of perfectly-plastic, Equation (104.519) is used for both yield surface and plastic potential and no evolution law is considered. Relevant stress derivatives are:

$$n_{ij} = \frac{1}{\sqrt{s_{ij}s_{ij}}} s_{ij} \quad ; \quad m_{ij} = \frac{1}{\sqrt{s_{ij}s_{ij}}} s_{ij} \tag{104.522}$$

Base on obtained stress derivatives, plastic consistency parameter is derived as

$$d\lambda = \frac{n_{ij} E_{ijpq} d\epsilon_{pq}}{n_{ij} E_{ijkl} m_{kl}} = \frac{n_{ij} d\sigma_{ij}}{2\mu} \tag{104.523}$$

The elastic-plastic tangent tensor is then

$$\begin{aligned} E^{pl}_{pqmn} &= \frac{E_{pqkl} n_{kl} m_{ij} E_{ijmn}}{n_{ab} E_{abcd} m_{cd}} = 2\mu n_{pq} n_{mn} \\ &= 2\mu \frac{1}{\sqrt{s_{kl}s_{kl}}} s_{pq} \frac{1}{\sqrt{s_{ij}s_{ij}}} s_{mn} = 2\mu \frac{1}{s_{kl}s_{kl}} s_{pq} s_{mn} \end{aligned} \tag{104.524}$$

In case isotropic hardening, ξ and h are not zero anymore, since k is scalar internal variable that is updated in each increment. Harderning/softenig functions, scalar functions in this case. ξ and h can be calculated as

$$\xi = -\sqrt{\frac{2}{3}} \tag{104.525}$$

$$h = \sqrt{\frac{2}{3} m_{ij} m_{ij}} k = \sqrt{\frac{2}{3}} k \tag{104.526}$$

Therefore, plastic parameter $(d\lambda)$ and tangnet elastic-plastic tensor (E^{pl}_{pqmn}) can be written as:

$$d\lambda = \frac{n_{ij} E_{ijpq} d\epsilon_{pq}}{n_{ij} E_{ijkl} m_{kl} - \xi\ h} = \frac{n_{ij} d\sigma_{ij}}{2\mu + \frac{2}{3}k} \tag{104.527}$$

$$\begin{aligned} E^{pl}_{pqmn} &= \frac{E_{pqkl} n_{kl} m_{ij} E_{ijmn}}{n_{ab} E_{abcd} m_{cd} - \xi\ h} = \frac{4\mu^2 n_{pq} n_{mn}}{2\mu + \frac{2}{3}k} \\ &= \frac{4\mu^2 \frac{1}{\sqrt{s_{kl}s_{kl}}} s_{pq} \frac{1}{\sqrt{s_{ij}s_{ij}}} s_{mn}}{2\mu - \frac{2}{3}k} = \frac{4\mu^2 \frac{1}{s_{kl}s_{kl}} s_{pq} s_{mn}}{2\mu + \frac{2}{3}k} \end{aligned} \tag{104.528}$$

The value of k should be updated at each step using the following equation:

$$k^{updated} = k + dk = k + hd\lambda = k + \sqrt{\frac{2}{3}}kd\lambda = k(1 + \sqrt{\frac{2}{3}}d\lambda) \tag{104.529}$$

For case the of von Mises with kinematic hardening, α_{ij} is the tensorial internal variable to be updated at each step of analysis, while tensor functions ξ_{ab} and h_{ab} can be calculated from the following equations using Armstrong-Frederick saturation-type kinematic hardening rule (Armstrong and Frederick (1966); Lemaitre and Chaboche (1990)) :

$$\xi_{ab} = \frac{\partial F}{\partial \alpha_{ab}} = -\frac{1}{\sqrt{(s_{ab} - \alpha_{ab})(s_{ab} - \alpha_{ab})}}(s_{ab} - \alpha_{ab}) \tag{104.530}$$

Recall that $dq_{ij} = d\alpha_{ij} = d\lambda h_{ij}$, and $d\epsilon^{pl}_{ij} = d\lambda m_{ij}$ and that the Armstrong-Frederic kinematic hardening rule for the internal variable α_{ij} is given as:

$$d\alpha_{ij} = \frac{2}{3}h_a \left(d\epsilon^p_{ij}\right)^{dev} - c_r\alpha_{ij}\sqrt{\frac{2}{3}\left(d\epsilon^p_{st}\right)^{dev}\left(d\epsilon^p_{st}\right)^{dev}} \tag{104.531}$$

so that

$$h_{st} = \frac{2}{3}h_a m_{st} - \alpha_{st}c_r\sqrt{\frac{2}{3}} \tag{104.532}$$

since $\sqrt{m_{ij}m_{ij}} = 1$.

Plastic consistency parameter $(d\lambda)$ and tanget elastic-plastic tensor (E^{pl}_{pqmn}) can be expressed as:

$$d\lambda = \frac{n_{ij}E_{ijpq}d\epsilon_{pq}}{n_{ij}E_{ijkl}m_{kl} - \xi\ h} = \frac{n_{ij}d\sigma_{ij}}{2\mu - \xi_{ab}\ h_{ab}} \tag{104.533}$$

$$E^{pl}_{pqmn} = \frac{E_{pqkl}n_{kl}m_{ij}E_{ijmn}}{n_{ab}E_{abcd}m_{cd} - \xi\ h} = \frac{4\mu^2 n_{pq}n_{mn}}{2\mu - \xi_{ij}\ h_{ij}} \tag{104.534}$$

Full, developed form of tangent elastic-plastic stiffness can be written as

$$
E^{pl}_{pqmn} = \frac{E_{pqkl}\left(\frac{1}{\sqrt{(s_{ab}-\alpha_{ab})(s_{ab}-\alpha_{ab})}}\left(s_{kl}-\alpha_{kl}\right)\right)\left(\frac{1}{\sqrt{(s_{ab}-\alpha_{ab})(s_{ab}-\alpha_{ab})}}\left(s_{ij}-\alpha_{ij}\right)\right)E_{ijmn}}{\left(\frac{1}{\sqrt{(s_{mn}-\alpha_{mn})(s_{mn}-\alpha_{mn})}}\left(s_{ab}-\alpha_{ab}\right)\right)E_{abcd}\left(\frac{1}{\sqrt{(s_{mn}-\alpha_{mn})(s_{mn}-\alpha_{mn})}}\left(s_{cd}-\alpha_{cd}\right)\right)-\left(-\frac{1}{\sqrt{(s_{ab}-\alpha_{ab})(s_{ab}-\alpha_{ab})}}(s_{rs}-\alpha_{rs})\right)\left(\frac{2}{3}h_a m_{ab}-\frac{2}{3}\alpha_{rs}c_r\right)} \tag{104.535}
$$

Above equation can be simplified, written in a shorter form by using a substitution

$$\mathbb{A} = \frac{1}{\sqrt{(s_{ab} - \alpha_{ab})(s_{ab} - \alpha_{ab})}} \tag{104.536}$$

it can be written as:

$$E^{pl}_{pqmn} = \frac{E_{pqkl}\left(\mathbb{A}\left(s_{kl} - \alpha_{kl}\right)\right)\left(\mathbb{A}\left(s_{ij} - \alpha_{ij}\right)\right)E_{ijmn}}{\left(\mathbb{A}\left(s_{ab} - \alpha_{ab}\right)\right)E_{abcd}\left(\mathbb{A}\left(s_{ab} - \alpha_{ab}\right)\right) - (-\mathbb{A})\left(\frac{2}{3}h_a\left(\mathbb{A}\left(s_{cd} - \alpha_{cd}\right)\right) - \sqrt{\frac{2}{3}}\alpha_{cd}c_r\right)\left(s_{cd} - \alpha_{cd}\right)} \tag{104.537}$$

Finally the full elastic-plastic tangent stiffness tensor can be written as:

$$\begin{aligned}
&E^{el\text{–}pl}_{pqmn} = \\
&E^{el}_{pqmn} - E^{pl}_{pqmn} = \\
&E^{el}_{pqmn} - \frac{E_{pqkl}\left(\mathbb{A}\left(s_{kl} - \alpha_{kl}\right)\right)\left(\mathbb{A}\left(s_{ij} - \alpha_{ij}\right)\right)E_{ijmn}}{\left(\mathbb{A}\left(s_{ab} - \alpha_{ab}\right)\right)E_{abcd}\left(\mathbb{A}\left(s_{cd} - \alpha_{cd}\right)\right) - (-\mathbb{A})\left(\frac{2}{3}h_a\ \left(\mathbb{A}\left(s_{cd} - \alpha_{cd}\right)\right) - \sqrt{\frac{2}{3}}\alpha_{cd}c_r\right)\left(s_{cd} - \alpha_{cd}\right)}
\end{aligned} \tag{104.538}$$

104.6.18.4 Nonlinear Elastic Model in 1D based on Armstrong Frederick Equation

$$d\sigma_{ij} = d\alpha_{ij} = \frac{2}{3}h_a\left(d\epsilon^{p}_{ij}\right)^{dev} - c_r\sigma_{ij}\sqrt{\frac{2}{3}\left(d\epsilon^{p}_{st}\right)^{dev}\left(d\epsilon^{p}_{st}\right)^{dev}} \tag{104.539}$$

104.6.18.5 Drucker-Prager Model

Drucker-Prager model yield surface, including rotational kinematic hardening can be written as

$$f = \alpha I_1 + [\frac{1}{2}(s_{ij} - p\alpha_{ij})(s_{ij} - p\alpha_{ij})]^{\frac{1}{2}} - \beta = 0 \tag{104.540}$$

where α and β are material constants.

By coinciding Drucker-Prager cone with the outer apexes of the Mohr-Coulomb hexagon locus, the constants for compressive cone of Drucker-Prager can be evaluated as shown in Equation (104.541) by knowing the soil strength parameters of cohesion (c) and friction angle (ϕ):

$$\alpha = \frac{2\sin\phi}{\sqrt{3}(3 - \sin\phi)}, \quad \beta = \frac{6\cos\phi}{\sqrt{3}(3 - \sin\phi)}c \tag{104.541}$$

Drucker-Prager yield function for cohesionless sands ($k = 0$) can be obtained as:

$$f = \alpha I_1 + \sqrt{J_2} = 0 \tag{104.542}$$

or in $p-q$ space:

$$f = q - Mp = 0 \tag{104.543}$$

which then M can be obtained as:

$$M = \frac{6\sin\phi}{3-\sin\phi} \tag{104.544}$$

By considering kinematic hardening, Equation (104.543) can be expressed as:

$$f = \left[\left(s_{ij}-p\alpha_{ij}\right)\left(s_{ij}-p\alpha_{ij}\right)\right]^{0.5} - \sqrt{\frac{2}{3}}kp = 0 \tag{104.545}$$

which then stress derivative of yield function and plastic potential function in case of associated plasticity can be defined as:

$$\begin{aligned} n_{ij} &= m_{ij} = \frac{\partial f}{\partial\sigma_{ij}} = \frac{\partial g}{\partial\sigma_{ij}} \\ &= \left[\left(s_{ij}-p\alpha_{ij}\right) + \frac{1}{3}\alpha_{pq}\delta_{ij}\left(s_{pq}-p\alpha_{pq}\right)\right]\left[\left(s_{rs}-p\alpha_{rs}\right)\left(s_{rs}-p\alpha_{rs}\right)\right]^{-0.5} \\ &+ \sqrt{\frac{2}{27}}k\delta_{ij} \end{aligned} \tag{104.546}$$

To find the closed form equation of elastic-plastic modulus for case of perfectly-plastic, Equation (104.545) is used for associated plasticity rule with no hardening. By these assumptions stress derivative is calculated as:

$$n_{ij} = m_{ij} = \frac{1}{\sqrt{s_{ij}s_{ij}}}s_{ij} + \sqrt{\frac{2}{27}}k\delta_{ij} \tag{104.547}$$

Splitting plastic modulus to different parts:

$$\begin{aligned} Hq_{ij} &= E_{ijkl}m_{kl} = \{\lambda\delta_{ij}\delta_{kkl} + \mu(\delta_{ik}\delta_{jl}+\delta_{il}\delta_{jk})\}\{\frac{1}{\sqrt{s_{mn}s_{mn}}}s_{kl} + \sqrt{\frac{2}{27}}k\delta_{kl}\} \\ &= 2\mu\frac{1}{\sqrt{s_{mn}s_{mn}}}s_{ij} + \delta_{ij}\{\frac{1}{\sqrt{s_{mn}s_{mn}}}\lambda(s_{pq}\delta_{pq}) + \sqrt{\frac{2}{27}}k(3\lambda+2\mu)\} \\ &= 2\mu\frac{1}{\sqrt{s_{mn}s_{mn}}}s_{ij} + B\delta_{ij} \end{aligned} \tag{104.548}$$

Assuming:

$$B = \frac{1}{\sqrt{s_{mn}s_{mn}}}\lambda(s_{pq}\delta_{pq}) + \sqrt{\frac{2}{27}}k(3\lambda + 2\mu) \tag{104.549}$$

Plastic parameter can be written as:

$$d\lambda = \frac{n_{ij}E_{ijpq}d\epsilon_{pq}}{n_{ij}E_{ijpq}m_{kl}} = \frac{\lambda_{nominator}}{\lambda_{denominator}} \tag{104.550}$$

$$\lambda_{nominator} = n_{ij}E_{ijpq}d\epsilon_{pq} = n_{ij}d\sigma_{ij} \tag{104.551}$$

$$\begin{aligned}\lambda_{denominator} &= n_{ij}E_{ijpq}m_{kl} = n_{ij}Hq_{ij}\\ &= \{\frac{1}{\sqrt{s_{ij}s_{ij}}}s_{ij} + \sqrt{\frac{2}{27}}k\delta_{ij}\}\{2\mu\frac{1}{\sqrt{s_{mn}s_{mn}}}s_{ij} + B\delta_{ij}\}\\ &= 2\mu + \frac{1}{\sqrt{s_{mn}s_{mn}}}(B + 2\mu\sqrt{\frac{2}{27}}k)(s_{ij}\delta_{ij}) + 3B\sqrt{\frac{2}{27}}k\end{aligned} \tag{104.552}$$

Finally plastic tensor can be expressed as:

$$E^{pl}_{pqmn} = \frac{E_{pqkl}n_{kl}m_{ij}E_{ijmn}}{n_{ab}E_{abcd}m_{cd}} = \frac{Hq_{pq}Hq_{mn}}{\lambda_{denominator}} \tag{104.553}$$

where:

$$\begin{aligned}Hq_{pq}Hq_{mn} &= \{\frac{2\mu}{\sqrt{s_{ij}s_{ij}}}s_{pq} + B\delta_{pq}\}\{\frac{2\mu}{\sqrt{s_{ij}s_{ij}}}s_{mn} + B\delta_{mn}\}\\ &= 4\mu^2\frac{1}{s_{ij}s_{ij}}s_{pq}s_{mn} + 4\mu\frac{1}{\sqrt{s_{ij}s_{ij}}}Bs_{pq}\delta_{mn} + B^2\delta_{pq}\delta_{mn}\end{aligned} \tag{104.554}$$

When isotropic hardening is considered for Drucker-Prager model, ξ and h are not zero. These parameters can be obtained using following equations:

$$\xi = -p\sqrt{\frac{2}{3}} \tag{104.555}$$

$$h = \sqrt{\frac{2}{3} m_{ij} m_{ij}} k = \sqrt{\frac{2}{3}(1 + \frac{2}{9}k^2)} k \tag{104.556}$$

Plastic parameter ($d\lambda$) and plastic modulus (E^{pl}_{pqmn}) can be written as:

$$\lambda_{denominator} \;\; = \;\; n_{ij} E_{ijpq} m_{kl} - \xi\ h = n_{ij} H q_{ij} - \xi\ h \tag{104.557}$$

$$E^{pl}_{pqmn} = \frac{E_{pqkl} n_{kl} m_{ij} E_{ijmn}}{n_{ab} E_{abcd} m_{cd}} = \frac{H q_{pq} H q_{mn}}{\lambda_{denominator}} \tag{104.558}$$

k is the scalar internal variable to be updated at each step of analysis using following equation:

$$\begin{aligned} k^{updated} &= k + dk = k + h d\lambda \\ &= k + \sqrt{\frac{2}{3}(1 + \frac{2}{9}k^2)} k d\lambda = k(1\sqrt{\frac{2}{3}(1 + \frac{2}{9}k^2)} d\lambda) \end{aligned} \tag{104.559}$$

By considering kinematic hardening for Drucker-Prager material model, tensorial internal variable (α_{ij}) is introduced and has to be updated at each step of analysis. The stress derivatives of yield function and plastic potential function can be written as:

$$n_{ij} = m_{ij} = \frac{1}{\sqrt{(s_{ij} - p\alpha_{ij})(s_{ij} - p\alpha_{ij})}} (s_{ij} - p\alpha_{ij}) + \sqrt{\frac{2}{27}} k \delta_{ij} \tag{104.560}$$

Tensorial parameters of $\xi)ij$ and h_{ij} can be calculated using Armstrong-Frederick saturation-type kinematic hardening rule (Armstrong and Frederick (1966)):

$$\xi_{ij} = -\frac{p}{\sqrt{(s_{ij} - p\alpha_{ij})(s_{ij} - p\alpha_{ij})}} (s_{ij} - p\alpha_{ij}) \tag{104.561}$$

$$h_{ij} = \frac{2}{3} h_a m_{ij} - \alpha_{ij} c_r \sqrt{\frac{2}{3} m_{ij} m_{ij}} \tag{104.562}$$

Plastic parameter ($d\lambda$) and plastic modulus (E^{pl}_{pqmn}) are obtained from following equations:

$$d\lambda = \frac{n_{ij} E_{ijpq} d\epsilon_{pq}}{n_{ij} E_{ijkl} m_{kl} - \xi_{ij}\ h_{ij}} = \frac{n_{ij} d\sigma_{ij}}{2\mu - \xi_{ij}\ h_{ij}} \tag{104.563}$$

$$E^{pl}_{pqmn} = \frac{E_{pqkl} n_{kl} m_{ij} E_{ijmn}}{n_{ab} E_{abcd} m_{cd} - \xi_{ij}\ h_{ij}} \tag{104.564}$$

Tensorial internal variable (α_{ij}) can be updated using following equation:

$$\begin{aligned} {\alpha_{ij}}^{updated} &= \alpha_{ij} + d\alpha_{ij} = \alpha_{ij} + h_{ij} d\lambda \\ &= \alpha_{ij} + (\frac{2}{3} h_a m_{ij} - \alpha_{ij} c_r \sqrt{\frac{2}{3} m_{ij} m_{ij}}) d\lambda \end{aligned} \tag{104.565}$$

104.6.18.6 Modified Cam-Clay Model

The critical state line for Cam-Clay can be written as

$$e_c = e_{c,r} - \lambda_c \ln p_c \tag{104.566}$$

where e_c is the critical void ratio at critical mean stress (p_c), $e_{c,r}$ is the reference critical void ratio, and λ_c is the normal consolidation slope. In general it is assumed that the normal consolidation line (NCL) is parallel to CSL, which is defined as:

$$e = e_\lambda - \lambda \ln p \tag{104.567}$$

where e_λ is the intercept on the NRL at $p = 1$. λ is the normal consolidation slope or the elasto-plastic slope of $e - \ln p$ relation. The same relation is used for unloading-reloading line (URL) with different slope as:

$$e = e_\kappa - \kappa \ln p \tag{104.568}$$

where e_κ is the intercept on the URL at $p = 1$. The yield function for Cam-Clay model can be defined as

$$f = q^2 - M^2[p(p_0 - p)] = 0 \tag{104.569}$$

where M is the critical state stress ration in $q - p$ space and p_0 is the initial internal scalar variable which will be changed by change in plastic volumetric strain.

Cam-Clay model is one of the associated flow rule models which means the same function is used for both yield and plastic potential surfaces ($f = g$). The plastic flow of the Cam-Clay model is associated with its yield function, in other words, the plastic flow is defined by the potential function (g), which is assumed the same as the yield function (f).

The evolution law of for Cam-Clay model is a scalar one which can be expressed by:

$$\dot{p}_0 = \frac{(1+e)p_0}{\lambda - \kappa}\dot{\epsilon}_v^p \tag{104.570}$$

where e is the void ratio, λ is the normal consolidation slope or the elasto-plastic slope of $e-\ln p$ relation, and κ is the slope of unloading-reloading line. This equation proves that the change of p_0 is controlled by change of plastic volumetric strain. By considering the yield function expressed in Equation (104.569) and considering the associated flow rule, stress derivatives can be evaluated as:

$$n_{ij} = m_{ij} = 3s_{ij} + \frac{1}{3}M^2(p_0 - 2p)\delta_{ij} \tag{104.571}$$

Elastic modulus in terms of shear modulus (G) and bulk modulus (K) can be expressed as:

$$E_{ijkl} = (K - \frac{2}{3}G)\delta_{ij}\delta_{kl} + G(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) \tag{104.572}$$

$$\begin{aligned}
Hq_{ij} &= E_{ijkl}m_{kl} \\
&= \{(K - \frac{2}{3}G)\delta_{ij}\delta_{kl} + G(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk})\}\{3s_{ij} + \frac{1}{3}M^2(p_0 - 2p)\delta_{ij}\} \\
&= 6Gs_{ij} + \{(K - \frac{2}{3}G)M^2(p_0 - 2p) + \frac{2}{3}M^2(p_0 - 2p)G\}\delta_{ij}
\end{aligned} \tag{104.573}$$

Defining D as:

$$D = (K - \frac{2}{3}G)M^2(p_0 - 2p) + \frac{2}{3}M^2(p_0 - 2p)G \tag{104.574}$$

ξ and h in case of using Cam-Clay constitutive model can be expressed as following equations:

$$\xi = M^2 p \tag{104.575}$$

$$h = (1 + e_0)p_0\frac{d\epsilon_v}{\lambda - \kappa} \tag{104.576}$$

where $d\epsilon_v = M^2(2p - p_0)$. Plastic parameter can be written as:

$$d\lambda = \frac{n_{ij} E_{ijpq} d\epsilon_{pq}}{n_{ij} E_{ijpq} m_{kl} - \xi\, h} = \frac{\lambda_{nominator}}{\lambda_{denominator}} \tag{104.577}$$

$$\lambda_{nominator} = n_{ij} E_{ijpq} d\epsilon_{pq} = n_{ij} d\sigma_{ij} \tag{104.578}$$

$$\begin{aligned} \lambda_{denominator} &= n_{ij} E_{ijpq} m_{kl} - \xi\, h = n_{ij} H q_{ij} - \xi\, h \\ &= \{3 s_{ij} + \frac{1}{3} M^2 (p_0 - 2p) \delta_{ij}\}\{6 G s_{ij} + D \delta_{ij}\} - \xi\, h \\ &= 18 G s_{ij} s_{ij} + D M^2 (p_0 - 2p) - \xi\, h \end{aligned} \tag{104.579}$$

Then the plastic tensor can be expressed as:

$$\begin{aligned} E^{pl}_{pqmn} &= \frac{E_{pqkl} n_{kl} m_{ij} E_{ijmn}}{n_{ab} E_{abcd} m_{cd} - \xi\, h} \\ &= \frac{H q_{pq} H q_{mn}}{\lambda_{denominator}} \end{aligned} \tag{104.580}$$

where:

$$\begin{aligned} H q_{pq} H q_{mn} &= \{6 G s_{ij} + D \delta_{ij}\}\{6 G s_{ij} + D \delta_{ij}\} \\ &= D^2 \delta_{pq} \delta_{mn} + 6 G D \delta_{pq} s_{mn} + 6 G D s_{pq} \delta_{mn} + 36 G^2 s_{pq} s_{mn} \end{aligned} \tag{104.581}$$

p_0 is the scalar internal variable to be updated at each step of analysis using the following equation:

$$p_0{}^{updated} = p_0 + dp_0 = p_0 + h d\lambda \tag{104.582}$$

104.6.18.7 Comparison of Computational Time of Accelerated Constitutive Models with NewTemplate3Dep

In order to compare the computational time of accelerated constitutive models with the ones available in NewTemplate3Dep, cyclic simulations are done in constitutive level using explicit integration method. Simulations are done for all the mentioned cases of von Mises, Drucker-Prager, and Cam-Clay constitutive models. The ratios of computational time of NewTemplate3Dep (t_N) to accelerated constitutive models (t_{Acc}) are summarized in Table (104.1). As it is observed, there are improvements in range of 2 to 3 times in computational time which can lead to reducing the computational time of soil-structure systems.

Table 104.1: Comparison of computational time of accelerated constitutive models with NewTemplate3Dep

Constitutive Model	t_N/t_{Acc}	Constitutive Model	t_N/t_{Acc}
von Mises Perfectly Plastic	3.1	Drucker-Prager Perfectly Plastic	2.9
von Mises Isotropic Hardening	2.6	Drucker-Prager Isotropic Hardening	2.5
von Mises Kinematic Hardening	2.3	Drucker-Prager Kinematic Hardening	1.9
Cam-Clay	2.2	————	

104.7 Elastic-Plastic Models for Contacts, Joints and Interfaces

This section is based on Sinha and Jeremić (2017).

104.7.1 Experimental Data

The response of the interface plays a very important role on the behavior of deep and shallow foundations, retaining walls, geo-membranes, submerged structures and soil-structure interaction. The load transfer mechanism from structure to the soil acts at the interface. The soil-structure interface comprise of a very thin small shearing band. The initial investigation by Yoshimi and Kishida (1981) indicated the thickness of the shear band as nine times the mean grain size diameter D_{50}. Tejchman and Wu (1995) conducted several tests on sand-steel interface and concluded that the thickness of the interface for rough interface is 30-40 D_{50} and for smooth interface is 6-10 D_{50}. Dejong et al. (2006); DeJong and Westgate (2009) investigated the shear-zone thickness to be 5-10 times the mean particle diameter D_{50}. Martinez et al. (2015) conducted several axial and torsional shear experiments on sand-steel interface to understand the micro mechanics involved at interface. The micro-mechanical hypothesis proposed by Martinez et al. (2015) is shown in Figure 104.28. The thickness of the interface zone formed by the particle in axial shearing migrates along the interface whereas for torsional shearing it migrates away from the interface.Similar mechanical hypothesis was observed by Dejong et al. (2006) It was also observed that for torsional shear the shear band zone was 2-3 times larger than the purely axial shearing. It also depends upon the characteristics of the soil beneath and the surface structural material and its roughness.

Since 1960's, researchers have been carrying out experiments to understand the interface behavior. The initial works have been contributed by Potyondy (1961); Brumund and Leonards (1973); Littleton

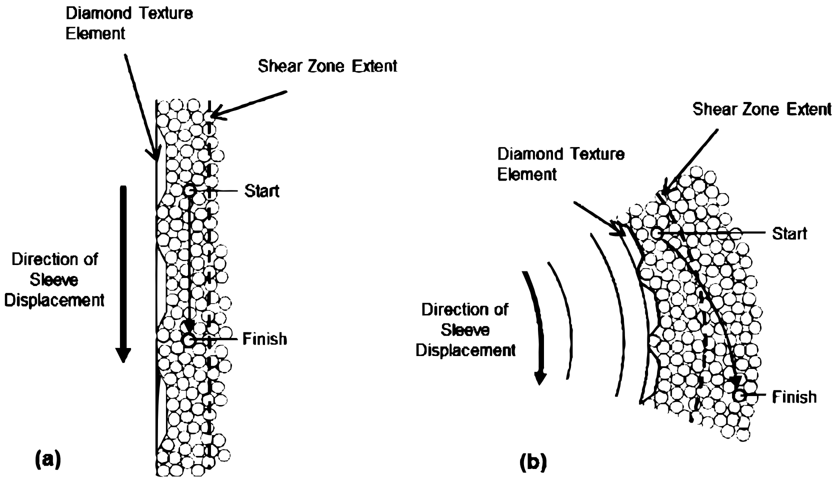


Figure 104.28: Hypothesis for a particle movement under (a) axial and (b) torsional loading (Martinez et al. (2015))

(1976). Potyondy (1961) studied the effect of soil-moisture content, structural surface roughness, soil-composition and normal load intensity on the skin-friction of the soil-structure interface. Brumund and Leonards (1973) investigated the static and dynamic friction angle between sand steel interface. Littleton (1976) performed drained and undrained tests on clay-steel interface and found that the shear stress response was steeper than the usual clay-clay interface. Based on the normal confinement σ_n, an initial hardening was observed until the peak shear strength τ_p is reached. After that softening to residual τ_r was observed. Later Desai (1981) emphasized on the importance of modeling of interface behavior for real soil-structure interactions. He also pointed out the lack of existing experimental data which could be used to develop constitutive models defining the interface behavior. Yoshimi and Kishida (1981) used a ring torsion apparatus to find the friction angle between dry sand and steel surface over a wide variation of surface roughness and sand density. Uesugi and Kishida (1986a,b); Kishida and Uesugi (1987) carried a series of laboratory experiments between steel and air-dried sands using simple shear apparatus shown in Figure 104.29 and Figure 104.30. It was found that the interface behavior is highly influenced by the sand type and surface roughness R_{max} while the effect of normal stress σ_n and mean grain size D_{50} are of poor significance. Thus, Kishida and Uesugi (1987) proposed a normalized roughness R_n to evaluate the relative coefficient of friction μ_y of sand-steel interface as shown in Figure ??.

$$R_{max}(L = D_{50})/D_{50} \tag{104.583}$$

where, $R_{max}(L = D_{50})$ is the R_{max} value of the steel surface with gauge length $L = D_{50}$.

Desai and Nagaraj (1988) performed a cyclic normal and shear tests on dry sand and concrete interface in translational shear box. Monotonic and cyclic normal loads along with cyclic shear loads

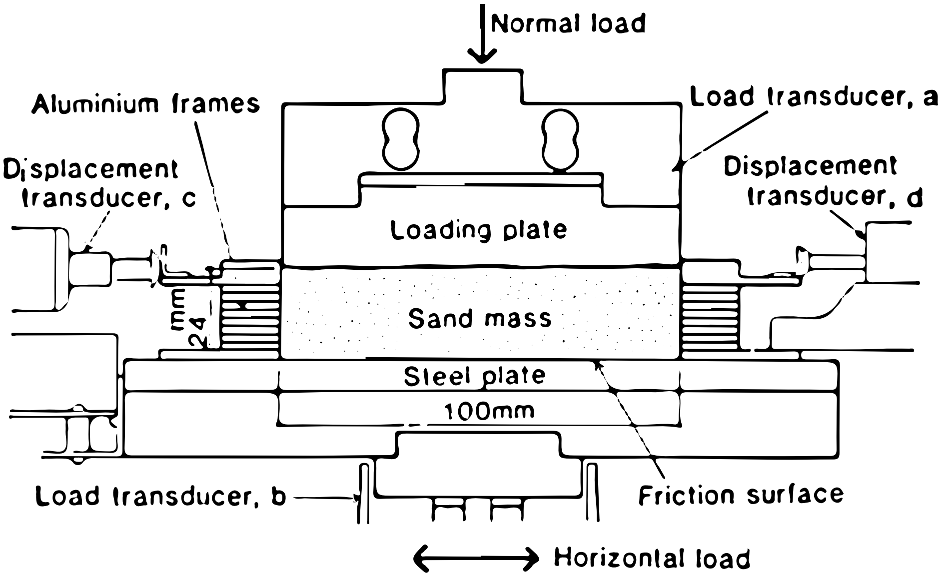


Figure 104.29: Section of friction test apparatus (Uesugi and Kishida (1986b)).

were applied. The shear behavior was modeled with a modified form of Ramberg-Osgood (R-O) model. Although Desai and Nagaraj (1988) did not show any experimental results, he idealized the normal interface behavior to be composed of (1) Virgin loading; (2) unloading; (3) reloading; 4 tensile condition; (5) partial debonding; and 6 rebonding as shown in Figure 104.32. Uesugi et al. (1989, 1990) studied the frictional behavior of sand-steel interface subject to repeated shearing under one-way or two-way loadings. It was found that under repeated loading conditions the coefficient of friction μ converged close to the residual coefficient of friction μ_r as could be observed in Figure ?? Boulon (1989) performed a lot of experiments on piles in sand. Direct simple shear tests were carried out to investigate the shear behavior between the granular soil and rough construction material. Based on the experimental results obtained later Boulon and Nova (1990) proposed a mathematical model and constitutive integration to model the interface behavior in finite element method (FEM). Aubry et al. (1990) proposed a dilatancy based cyclic elastic-plastic constitutive model for the interface. Cyclic loading functions with memory of last loading reversal was used to model subsequent loadings and unloading. The yield function was defined using simple Mohr Coulomb with additional parameter F as a function of normal stress σ_n and plastic compressibility β to account for curvature and dilation of the yield surface.

Fakharian and Evgin (1995) developed a 3-D apparatus capable of performing direct and simple shear type testing of interfaces between soil and structure. The developed apparatus was subsequently used to perform numerous experiments Fakharian and Evgin (1996); Evgin and Fakharian (1997); Fakharian and Evgin (1997); Fakharian (1996); Fakharian et al. (2002) over sand-steel interface for different stress paths for different relative densities D_r of sand. The 3-D apparatus made it possible to conduct 2-D

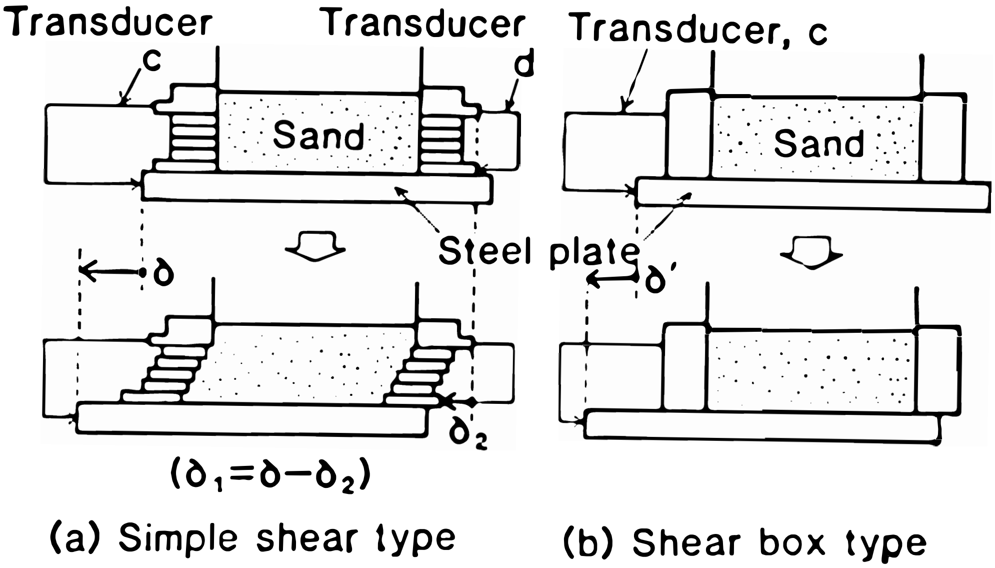


Figure 104.30: Measurement of tangential displacement (Uesugi and Kishida (1986b)).

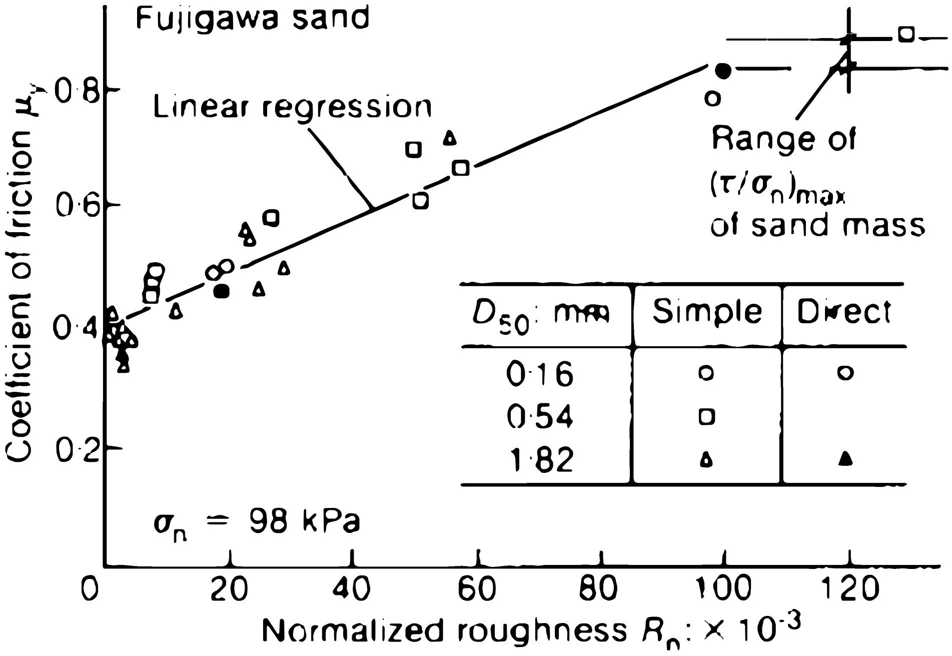


Figure 104.31: Coefficient of friction at yield μ_y and normalized roughness (after Kishida and Uesugi (1987)).

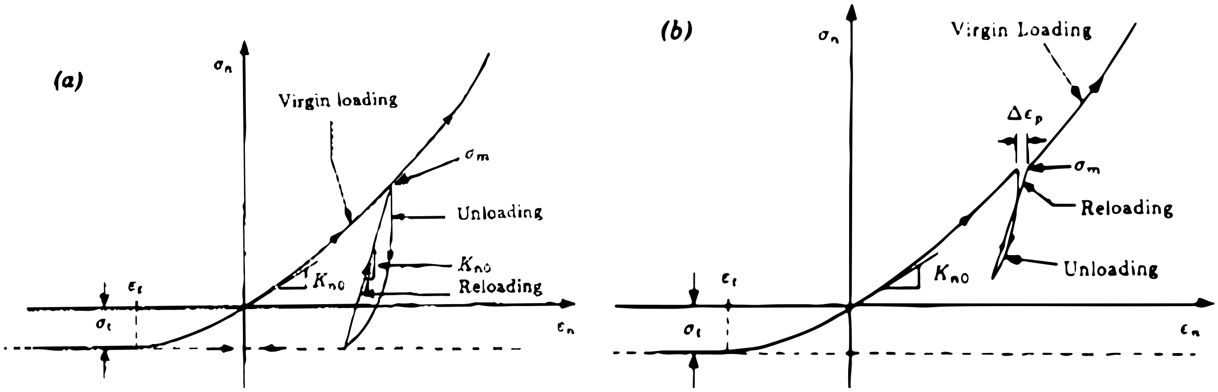


Figure 104.32: Schematic of Stress-Strain Response for Normal Behavior (Desai and Nagaraj (1988)) : (a) Virgin Loading and unloading with Tensile stress Condition; (b) Partial Loading.

shear test with constant normal stress σ_n. Monotonic and cyclic test results are shown in Figure 104.34 and Figure 104.34. The experiment results in Figure 104.34 clearly shows a peak shear stress ratio $(\tau/\sigma_n)_p$ and a residual stress ratio $(\tau/\sigma_n)_r$. Initial hardening and then softening depends upon the relative density D_r of the sand was observed. A higher relative density $D_r = 80\%$ sand shows dilation for lower confinement and thus a peak behavior whereas a low relative density $D_r = 25\%$ sand shows no dilation. For cyclic shear tests, the loose soil $D_r = 25\%$ shown in Figure 104.35(b) showed gain in strength due to densification resulting from particle breakage. While for soil with high relative density $D_r = 80\%$ almost no gain in shear strength was observed during cyclic shearing. The 3-D tests performed showed that the shear stress τ is almost isotropic for different shear stress paths.

Shahrour and Rezaie (1997) performed a series of monotonic and cyclic tests on Hostun Sand with rough and smooth surface with constant normal load condition. The results obtained were used to propose an elasto-plastic bounding surface based constitutive model for the interface behavior. The monotonic and cyclic test are shown in Figure 104.36 for rough and smooth interface surface. From Figure 104.36, it could be observed that for smooth interface, the shear stress τ increases only upto the critical shear stress τ_c. Whereas for rough soil, the shear stress τ hardens to a peak strength τ_p and then softens to the critical shear strength τ_c. The behavior observed for rough and smooth interface is similar to dense and loose soil as observed in tests by Fakharian and Evgin (1996). The cyclic test shown in Figure 104.36(c) showed similar response as was observed by Fakharian and Evgin (1996).

Dejong et al. (2006) performed laboratory soil-structure investigation under constant normal stiffness using particle image velocimetry (PIV)method. Cyclic shearing was carried out to simulate and understand the the shear behavior at pile interface. Cyclic degradation exponential model was proposed of model the change in void ratio. Later Mortara et al. (2007) also performed cyclic shear tests on sand

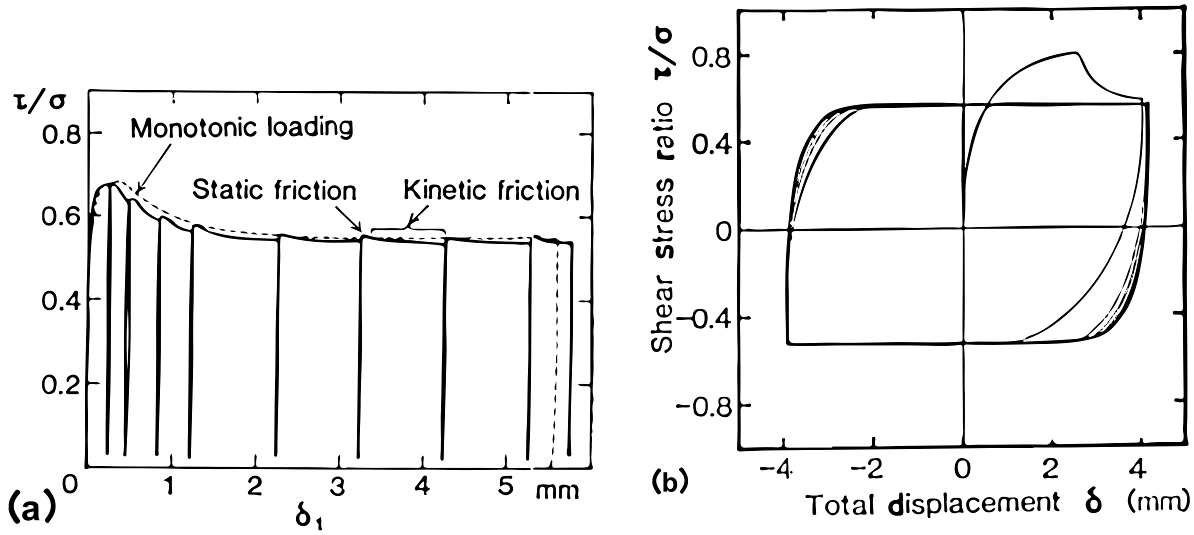


Figure 104.33: (a) Monotonic and (b) cyclic response of Toyora sand with steel interface (Uesugi et al. (1989)) ($D_r = 90\%, \sigma_n = 98kPa, R_n = 150e^{-3}$).

steel interfaces. DeJong and Westgate (2009) quantified the soil-structure interface behavior to the shearing on the factor like relative density D_r, particle angularity, particle hardness, surface roughness, normal stress and normal stiffness. Local as well as global load displacement response was recorded to understand the load-transfer mechanism.

104.7.2 Axial Contact, Joint, Interface

The contact/joint/interface behavior in the normal direction is modeled as penalty stiffness function as described in the Section 104.7.2.1. The penalty function can be chosen as linear with fixed stiffness also known as Hard Contact/Joint/Interface (Section 104.7.2.2), or it can be assumed to be a non-linear function with stiffness increasing exponentially with penetration. This type of normal behavior is called as Soft Contact/Joint/Interface (Section 104.7.2.3). Soft Contact/Joint/Interface represents more realistic soil-structure interface behavior. The soil becomes stiff as the penetration increases and gets relaxed upon unloading.

104.7.2.1 Penalty Method

At the interface of the soil-foundation system, an impenetrability constraint exists as shown in Equation ?? to ??. The contacting/interfacing bodies cannot penetrate into each other. The impenetrability leads to an inequality constraint, which requires special methods such as penalty method, Lagrange, barrier,

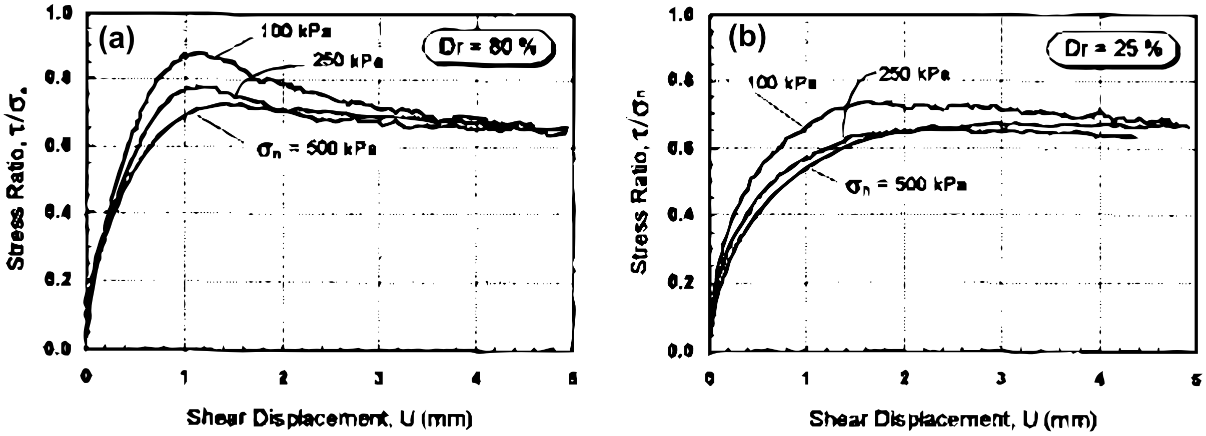


Figure 104.34: Stress ratio versus shear displacement for $\sigma_n = 100, 200, 500 kPa$: (a) rough surface-dense sand ($D_r = 80\%$); (b) rough surface loose sand ($D_r = 25\%$) (Fakharian and Evgin (1996)).

augmented Lagrangian, etc. as described in (Wriggers, 2002). Penalty method is a common approach used for solving constrained minimization (or maximization) problems involving inequalities as described in section ?? and Section ??. In this approach, a large penalty term is added to the minimizing functional to prevent the solution from escaping the constrained space.

Figure 104.37 shows a two contact/interface node pairs initially separated by a small distance of g in the contact/interface normal direction. During pure contact/interface/joint, the two node penetrates against each other by Δ_n. The instantaneous relative distance between the two contact/interface surfaces is u. Thus, if $u < g$ there is no contact/interface and normal force $N = 0$ otherwise there is contact/interface and a normal force N will act.

In the penalty stiffness formulation, a small penetration Δ_n is allowed between the mass and the floor having stiffness k_n such that during contact the normal force N is defined as

$$N = k_n \Delta_n \;\; if \;\; \Delta_n \geq 0 \tag{104.584}$$

where k_n can be thought of the normal contact/interface stiffness and Δ_n is the relative displacement between the two contact/interface surfaces with respect to the initial gap g in contact/interface normal direction. It is defined as the following

$$\Delta_n = u_n^r - g_n = u - g \tag{104.585}$$

where u_n^r is the relative displacement in the contact/interface normal direction and g_n is the initial gap in contact/interface normal direction. Theoretically, for rigid contact/interface case, the penalty stiffness k_n is assumed to infinite resulting in $\Delta_n = 0$. However, for numerical reasons, infinity is not possible, and

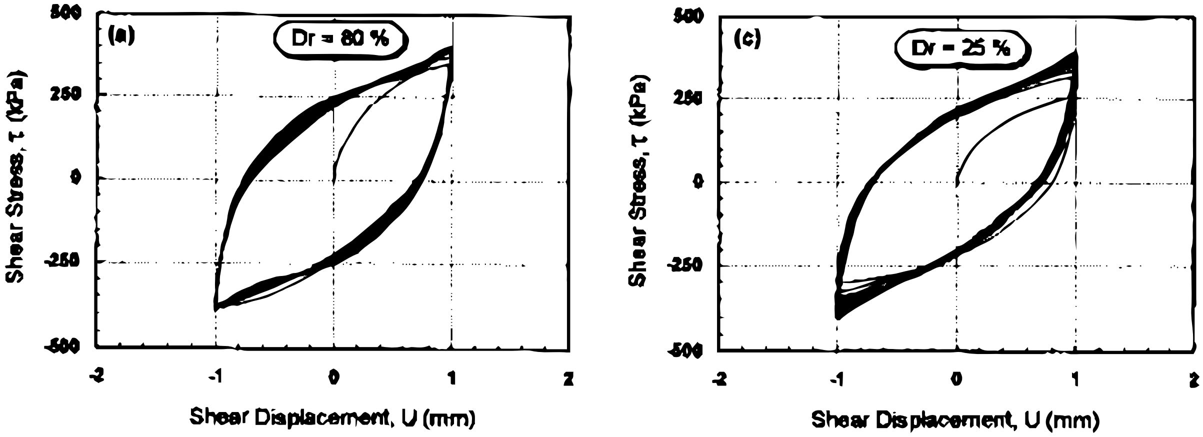


Figure 104.35: Cyclic test results, shear stress versus shear displacement for $\sigma_n = 500kPa$: (a) rough surface, dense sand ($D_r = 80\%$); (b) rough surface, loose sand ($D_r = 25\%$) (Fakharian and Evgin (1996)).

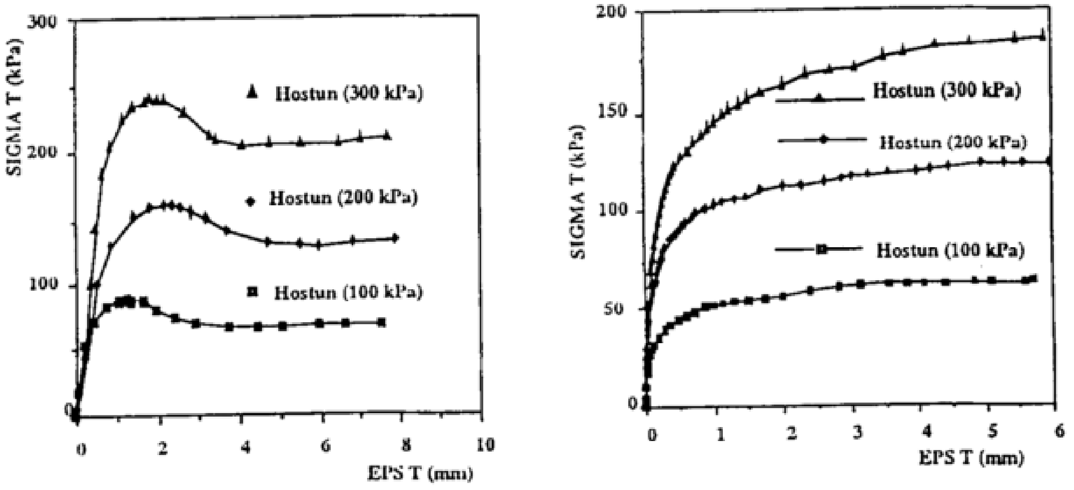
thus $\Delta_n = 0$ is never enforced. This results in small penetration at contact/interface surfaces resulting in $\Delta_n < 0$ during contact. For penalty method, the term penetration is thus normally referred to Δ_n defining the two possible states as:

- *No Contact/Joint/Interface* (Penetration $\Delta_n > 0$)
- *Contact/Joint/Interface State* (Penetration $\Delta_n \leq 0$)

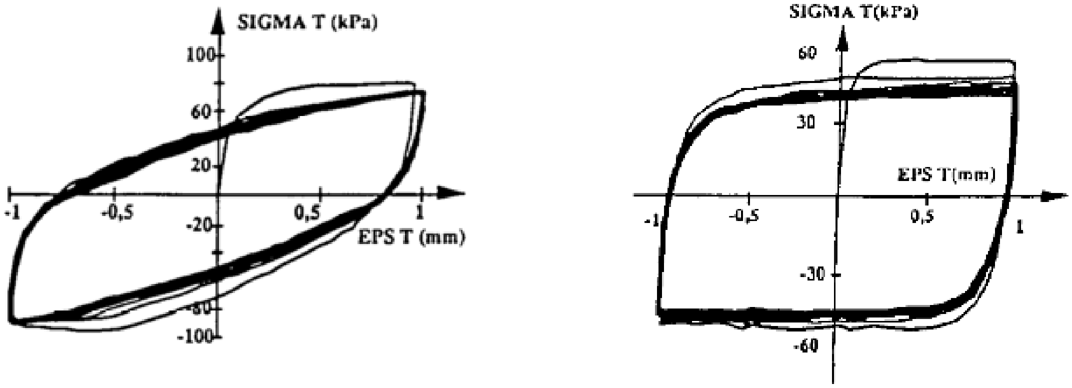
In equation 104.584, if the penalty stiffness parameter k_n is assumed to be constant and independent of penetration (Δ_n), it is referred as a hard contact. This type of contact/interface is more physical for interactions between two rigid surfaces or bodies. However, to model interaction between soft-soil and rigid foundation, a soft contact/interface with penalty stiffness increasing with penetration is preferred. The following Section 104.7.2.2 and 104.7.2.3 describes hard and soft contact/interface respectively.

For coupled contact/interface problems described in Section ??, in order to enforce the no-drainage condition in contact/interface normal direction between U (soil) and u (foundation) degrees of freedom, an additional penalty stiffness parameter k_p is required. Section ?? describes how penalty stiffness k_p is used to enforce the undrained condition.

It must be noted that in the penalty or any other method (Lagrange, barrier .. etc.) as described in (Wriggers, 2002), to get to the solution it has to take into account of whether the contact/interface is active or not. Thus, the inequality constraint has to be changed to the active (closed gap) or inactive (open gap) based on the state of contact. As a result, the topology of the structure changes due to



(a) Rough Interface (b) Smooth Interface



(c) Cyclic test on rough Interface (d) Cyclic test on smooth interface

Figure 104.36: (a) and (b) Monotonic and (c) and (d) cyclic test on dense Hostun sand and steel interface(Shahrour and Rezaie (1997)).

contact. This points out one of the difficulties while solving the contact/interface problem i.e. the stiffness matrix changes with active or inactive constraint equations.

As compared to one of the popular Lagrange method, the penalty method leads to non-physical penetration but does not create any additional variables. However, the non-physical penetration could be utilized to model more complicated normal contact/interface force function as such for soft contact/interface shown in Figure ?? and non-linear shear interface models as described in section ??.

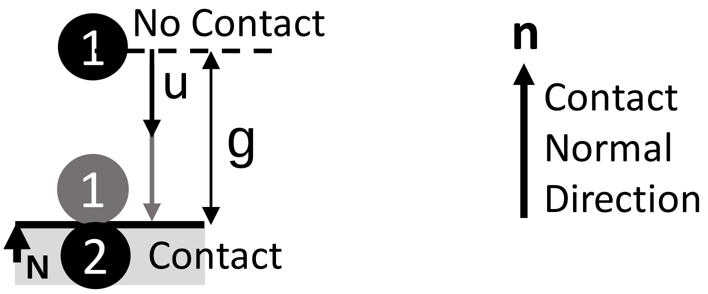


Figure 104.37: Contact/Interface/Joint Node Pairs.

104.7.2.2 Hard Contact/Joint/Interface

In hard contact, the normal penalty stiffness k_n is assumed to be constant with penetration Δ_n. As a result, the normal contact/interface force F_n or stress σ_n varies linearly with penetration.

$$\begin{aligned} F_n &= k_n \Delta_n \\ \sigma_n &= k_n \epsilon \end{aligned} \tag{104.586}$$

where k_n represents the normal stiffness between soil-structure interface and Δ_n is the penetration in contact/interface normal direction.

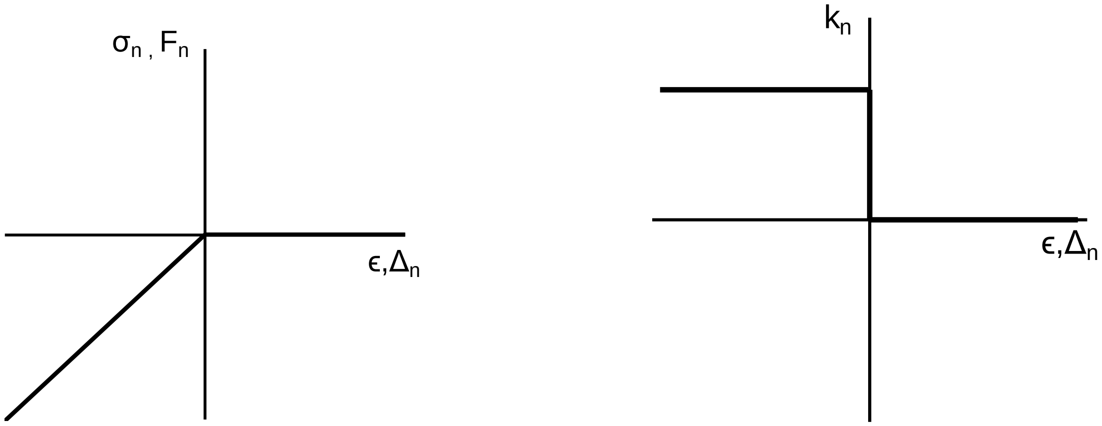


Figure 104.38: Hard contact/interface normal (a) force and (b) stiffness function with penetration Δ_n

Figure 104.38 shows the normal force F_n or stress σ_n and stiffness k_n as a function of penetration Δ_n or normal strain ϵ respectively. The normal stiffness k_n is assumed to be constant and thus has an abrupt jump or discontinuity at $\Delta_n = 0$ leading to C^0 continuity. The abrupt change of stiffness could often lead to numerical convergence problems.

104.7.2.3 Soft Contact/Joint/Interface

For rocks, (Gens et al., 1990) presented a nonlinear (hyperbolic) function of elastic normal stress with penetration. The function had different stiffness for loading and unloading up-to a permanent deformation u_{mc}. The hyperbolic function $u/(u - u_{mc})$ has a singularity at $u = u_{mc}$. It does not increase monotonically and does not poses continuous derivatives for $u \geq 0$. In FEM since, the stiffness cannot be infinite at $u = u_{mc}$ and the displacement u can be greater than $u \geq u_{mc}$, which can lead to numerical instability and convergence problems. As per authors knowledge, there has not been enough experimental investigation to understand the normal contact/interface behavior of the soil-structure interface. Desai and Nagaraj (1988) claim to have performed cyclic normal tests on a concrete-soil interface on a shear box but did not show any experimental results. Desai and Nagaraj (1988) idealized the normal behavior based on the critical state soil mechanics as shown in Figure 104.32. (Bandis et al., 1983) investigated the response of fresh and weathered rock. It could be observed that after some $(2-4)$ cycles, the loading and unloading curve fairly overlaps and could be approximated by the same function. More recently, Lei and Barton (2022) presented a very nice set of experiments with data for proper choice of interface parameters.

The normal behavior at the interface being modeled here using penalty stiffness approach, a non-linear exponential elastic function is assumed for the soft contact. The parameters include an initial stiffness k_i and a stiffening rate S_r to control the normal stress σ_n function with penetration Δ_n. In comparison to hard contact, soft contact/interface thus has a smooth exponential, normal contact/interface force function with penetration as shown in Figure ??. The exponential variation is expected to match the realistic increasing contact force with penetration as shown in Figure 104.32. As stated earlier, it must

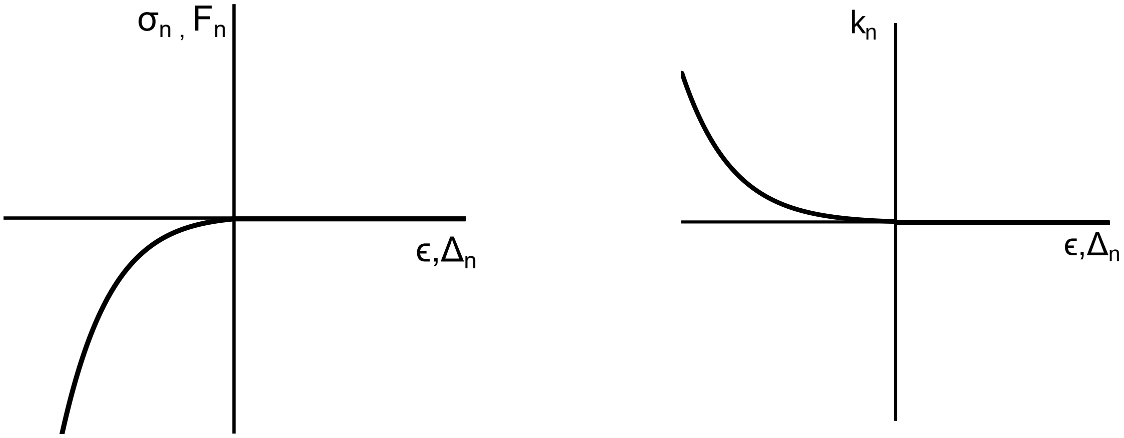


Figure 104.39: Soft contact/interface normal (a) force and (b) stiffness function with penetration Δ_n

be noted that in this model the response is assumed to be fully non-linear elastic with no tensile region. As a result, the loading and unloading stiffness and response is the same.

The non-linear normal force F_n or stress σ_n is defined as:

$$\begin{aligned} F_n &= k_i exp(-S_r \Delta_n) \Delta_n \\ \sigma_n &= k_i exp(-S_r \epsilon) \epsilon \end{aligned} \tag{104.587}$$

where k_i represents the initial normal stiffness between soil-structure interface and S_r represents the stiffening (or hardening) rate with penetration Δ_n or normal strain ϵ. Equation 104.587 represents the normal force F_n and stress σ_n for force based and stress based contact respectively. At soil-foundation interface, as the foundation penetrates more, the soil becomes harder resulting in an increase of interface normal stiffness k_n and normal stress σ_n. The stiffening rate leads to an exponential increment of contact/interface stress per unit of penetration Δ_n. From the above formulation it must be noted that for the stress based contact, the penetration Δ_n is replaced with normal strain ϵ. Subsequently, the parameters initial normal stiffness k_i and stiffening rate S_r should also be calibrated accordingly. Similarly, the stiffness and other derivatives could be obtained by replacing penetration Δ_n with normal strain ϵ. The normal stiffness k_n has unit of n/m for the force based contact/interface and Pa for the stress based contact.

Equation 104.587 could be differentiated to get the stiffness k_n as:

$$\begin{aligned} k_n &= k_i exp(-S_r \Delta_n)(1 - S_r \Delta_n) \\ k_n &= k_i exp(-S_r \epsilon)(1 - S_r \epsilon) \end{aligned} \tag{104.588}$$

It can be observed from equation 104.588 that putting $\Delta_n = 0$, the normal stiffness k_n becomes equal to initial normal stiffness k_i i.e. $k_n = k_i$. When $\Delta_n \geq 0$, the stiffness grows exponentially. Extending Equation 104.587 and 104.588 to uplift (no-contact) ($\Delta_n \geq 0$), with the assumption of small initial stiffness k_i, the force and stiffness function would lead to C^1 continuity. The C^1 continuity thus would lead to a smooth stiffness function even at the border of contact/interface and non-contact region resulting in quadratic convergence at global FEM level for the Newton-Raphson method. However the non-linear behavior would lead to comparatively larger number f iteration than hard contact,

It is also interesting to note that by setting the stiffening rate $S_r = 0$, hard contact/interface can be recovered i.e. $k_n = k_i$ and $\sigma_n = k_n \epsilon$. This demonstrates the generality of soft contact/interface formulation.

The exponential growth of stiffness in finite element method (FEM) can lead to numerical instability for large values. To avoid this, a maximum normal stiffness k_n^{max} is applied to restrict its further growth. Figure 104.40 shows the stiffness k_n function with and without a cap. The stiffness function thus can

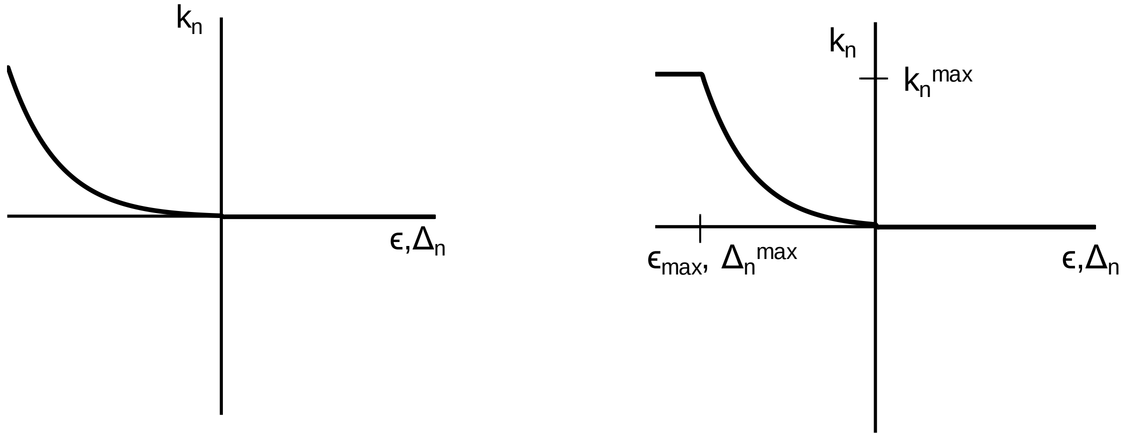


Figure 104.40: Soft Contact Stiffness function (a) without any limit and (b) with limit

be written as:

$$\begin{aligned} k_n &= max(k_i exp(-S_r \Delta_n)(1 - S_r \Delta_n), k_n^{max}) \\ k_n^{max} &= k_i exp(-S_r \Delta_n^{max})(1 - S_r \Delta_n^{max}) \end{aligned} \tag{104.589}$$

For implementing the above Equation 104.589 in FEM, there would be a need to find Δ_n^{max} corresponding to maximum allowable normal stiffness k_n^{max} in order to integrate the stiffness function k_n. An efficient method to get Δ_n^{max} is described in Section 104.7.2.3 below.

Iterative Method To Find Δ_n^{max} . For soft contact/interface implementation in FEM, it would be required to find out Δ_n^{max} for a given initial normal stiffness k_i, stiffening rate S_r and maximum normal stiffness k_n^{max}. Since Equation 104.589 is a nonlinear function, an iterative method is needed to get to the solution. One of the best solution search methods is the bisection method, which repeatedly bisects an assumed solution interval, choosing only one of the branches where the solution might exist. In this method, an initial guess of solution space is required.

Theoretically, a large solution space could be given but would not be computationally feasible and optimal. For the given problem, it is often desired to predict solution space to get in as fewer iterations as possible. To reach the solution optimally and efficiently, an initial guess of solution range for penetration Δ_n was found to be $[k_n^{max}/k_i, 0.5 k_n^{max}/k_i]$.

104.7.3 Shear Contact/Joint/Interface

104.7.3.1 Interface Shear Zone

At the soil-structure interface, there exists a thin shearing zone of 5-20 times the D_{50} Yoshimi and Kishida (1981); Martinez et al. (2015); Dejong et al. (2006); DeJong and Westgate (2009) as shown in Figure 104.41. Since the interface constitutive models are defined in stress-strain space, the applied displacements must be converted to strains. Based on the shear zone thickness SZ_h, the total shear strain γ and incremental shear strain $\Delta\gamma$ can be calculated as

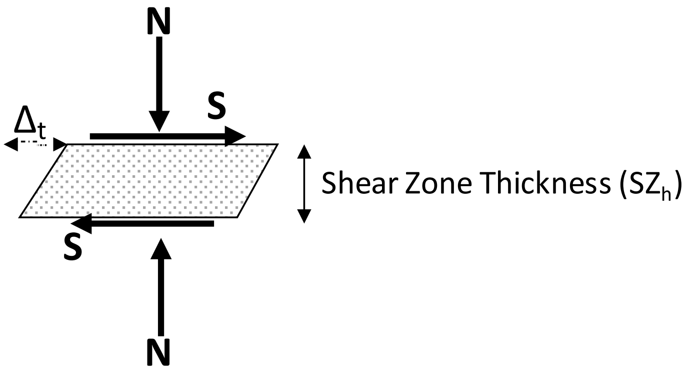


Figure 104.41: Thin shear zone at soil-structure interface

$$\begin{aligned} \gamma &= \Delta_t / SZ_h \\ \Delta\gamma &= \delta\Delta_t / SZ_h \end{aligned} \tag{104.590}$$

where Δ_t and $\delta\Delta_t$ are the total and incremental shear displacement at the interface between the two soil-structure contact/interface surface.

Similarly, the normal strain ϵ and incremental normal strain $\Delta\epsilon$ can be calculated as

$$\begin{aligned} \epsilon &= \Delta_n / SZ_h \\ \Delta\epsilon &= \delta\Delta_n / SZ_h \end{aligned} \tag{104.591}$$

where Δ_n and $\delta\Delta_n$ are the total and incremental penetration in contact/interface normal direction. In the interface constitutive models, the normal strain ϵ is generally also referred as volumetric strain ϵ_v Stutz (2016). The normal stress is assumed to offer confinement to the interface shear band. It must be noted that although a shear band of thickness SZ_h is assumed, the interface element itself has zero thickness. The shear zone thickness SZ_h is a material parameter for the interface models. The shear zone thickness can vary based on the roughness of the soil-structure interface but could be generally assumed to be around 5-20 mean particle size diameter D_{50}.

Shear Contact/Joint/Interface Models. This section describes three models to describe the non-linear shear interface behavior which is intended to capture some of the actual soil-structure interface response.

Apart from the traditional Mohr-Coulomb i.e. Elastic-Perfectly Plastic Shear (EPPS) model, two additional non-linear models have been proposed. The Non-Linear Hardening Shear (NLHS) is a non-linear Armstrong-Frederick type hardening model where the normalized shear stress parameter $\mu = \tau/\sigma_n$ increases non-linearly from 0 to residual normalized shear stress $\mu_r = \tau_r/\sigma_n$. Non-Linear Hardening Softening Shear (NLHSS) adds one more level of sophistication. It can model the softening of normalized shear stress parameter μ. Once the peak normalized shear stress $\mu_p = \tau_p/\sigma_n$ is attained, it starts to decrease to the residual normalized shear stress μ_r. Figure 104.42 shows a typical monotonic response of the three models for a constant normal stress σ_n;

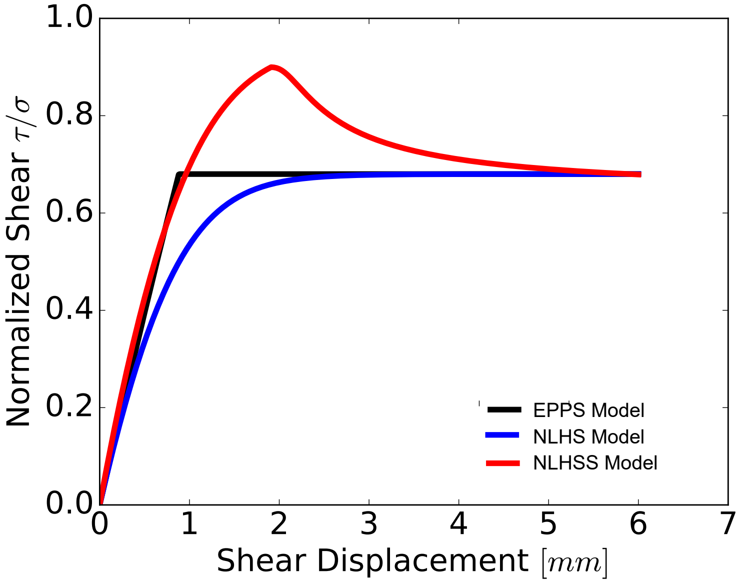


Figure 104.42: Comparison of the interface models with monotonic response

From the Figure 104.42, it can be observed that the models behave quite differently. The EPPS model reaches the residual state at very small shear strain $\gamma = \Delta_u/SZ_h$ level where as because of non-linear hardening both NLHS and NLHSS reach at larger shear strains γ. EPPS and NLHS do not show any peak behavior whereas NLHSS shows a peak followed by softening to residual strength. The models are explained in detail in the next section. NLHS and NLHSS model assume to have negligible to no elastic region and follow the elasto-plasticity theory. And since, the linear Armstrong-Frederick hardening parameter H_a is assumed to be equal to the elastic stiffness k_t resulting in the overall elasto-plastic stiffness equal to $0.5k_t$. The thin shear zone at the soil-structure interface starts to develop plastic deformation as soon as a small shear stress τ is applied. It must also be noted in Figure 104.42 that the elastic shear stiffness k_t depends upon the normal stress σ_n. Thus, for a given normal stress σ_n, the shear stiffness k_t is defined as

$$k_t = k_t^{\sigma_n} = (k_t)^{\sigma_{p0}} \frac{\sigma_n}{\sigma_{p0}} \tag{104.592}$$

where σ_{p0} is the constant stress of $101.3kPa$ and $(k_t)^{\sigma_{p0}}$ is the shear stiffness for a normal stress of $\sigma_n = \sigma_{p0} = 100kPa$. The models are thus developed using kinematic hardening plasticity with initial kinematic hardening stiffness H_a equal to the elastic stiffness k_t. As stated above, this leads to initial elastic-plastic stiffness equal to $0.5k_t$ and also results in incremental elastic strain energy equal to the incremental plastic free energy as described in Section ??.

104.7.3.2 Elastic Perfectly Plastic Shear (EPPS) Model

The simplest shear interface model is the Mohr Coulomb interface model with an elastic stiffness under no slippage and zero stiffness when it slips. The material behavior is of type elastic-perfectly plastic type. The yield function (f) is thus given as

$$f := \tau - \mu\sigma_n \leq 0 \tag{104.593}$$

where μ is a constant coefficient of friction, τ is the shear stress and σ_n is the normal stress.

Figure 104.43 shows the performance of EPPS model for different loading conditions. Since it is a elastic perfectly-plastic model, the shear stiffness k_t is constant with shear strain γ or displacement Δ_t and becomes zero (perfectly-plastic state) when it reaches its residual friction coefficient μ_r. Figure 104.43(a) and 104.43(c) shows the monotonic and full cyclic response with elastic perfectly-plastic behavior respectively. Figure 104.43(b) and 104.43(d) shows the monotonic cyclic behavior before and after reaching the residual friction coefficient μ_r. This kind of interface behavior is mostly observed between rigid surfaces in contact. For more realistic soil-structure interface non-linear yield function should be used as described in the coming sections.

104.7.3.3 Nonlinear Hardening Shear (NLHS) Model

In this model, the normalized shear stress hardening variable μ increases from 0 to critical or residual normalized shear stress μ_r using the non-linear Armstrong Frederick type hardening law. The evolution rule for frictional hardening variable μ is given as

$$\Delta\mu = k_t \Delta\gamma^p - \frac{k_t}{\mu_r} |\Delta\gamma^p| \mu \tag{104.594}$$

where k_t is the non-linear elastic hardening variable and $\Delta\gamma^p$ is the plastic part of the shear strain $\Delta\gamma$. The material behavior is thus of type non-linear hardening type.

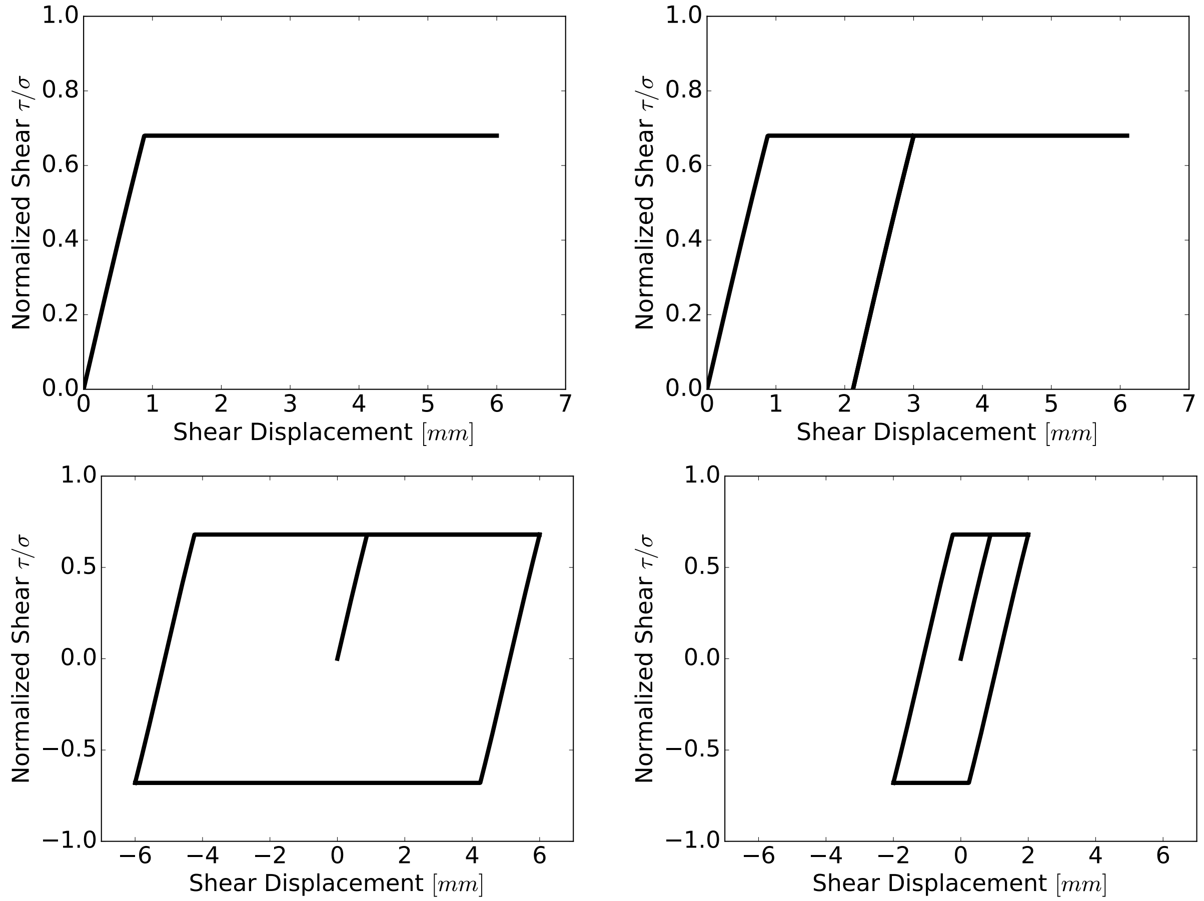


Figure 104.43: Response of Linear Elastic Perfectly Plastic Shear (EPPS) Model with normal stress of $100kPa$, residual coefficient of friction $\mu_r = 0.68$, shear stiffness $k_t = 200kPa$ and shear zone length $SZ_h = 5mm$.

The yield function (f) is still given as

$$f := \tau - \mu\sigma_n \leq 0 \tag{104.595}$$

where the normalized shear stress hardening variable μ evolves by Equation 104.594, τ is the shear stress and σ_n is the normal stress.

Figure 104.44 shows the performance of NLHS model for different loading conditions. It can be observed that the hardening variable μ increases non-linearly from 0 to residual normalized shear stress ratio μ_r at large shear displacements. As compared to the EPPS models, it is more realistic as the soil-structure interface develops the shear strength gradually with increments of shear strain $\Delta\gamma$ or shear displacements Δ_t.

For loose or low relative density D_r soil at soil-structure interface Fakharian and Evgin (1996);

Shahrour and Rezaie (1997) as shown in Figure 104.34(b) and 104.36(b), this model could be calibrated to model the non-linear hardening response. The monotonic and full cyclic response of this model shown in Figure 104.44(b) can be seen to match the interface behavior investigated by Uesugi et al. (1989) and is shown in Figure ??. Figure 104.44(d) shows the response of the model subject to cyclic shearing before reaching the residual strength. It could be observed that it is able to model the non-linear interface behavior as investigated by Fakharian and Evgin (1996); Shahrour and Rezaie (1997) which is shown in Figure 104.35 and Figure 104.36(c) respectively.

For dense soil with higher relative density D_r, it is important to model the peak normalized shear stress μ_p, followed by the softening behavior until the residual shear stress μ_r is reached. The NLHSS model proposed in next section can be used to model both hardening and softening.

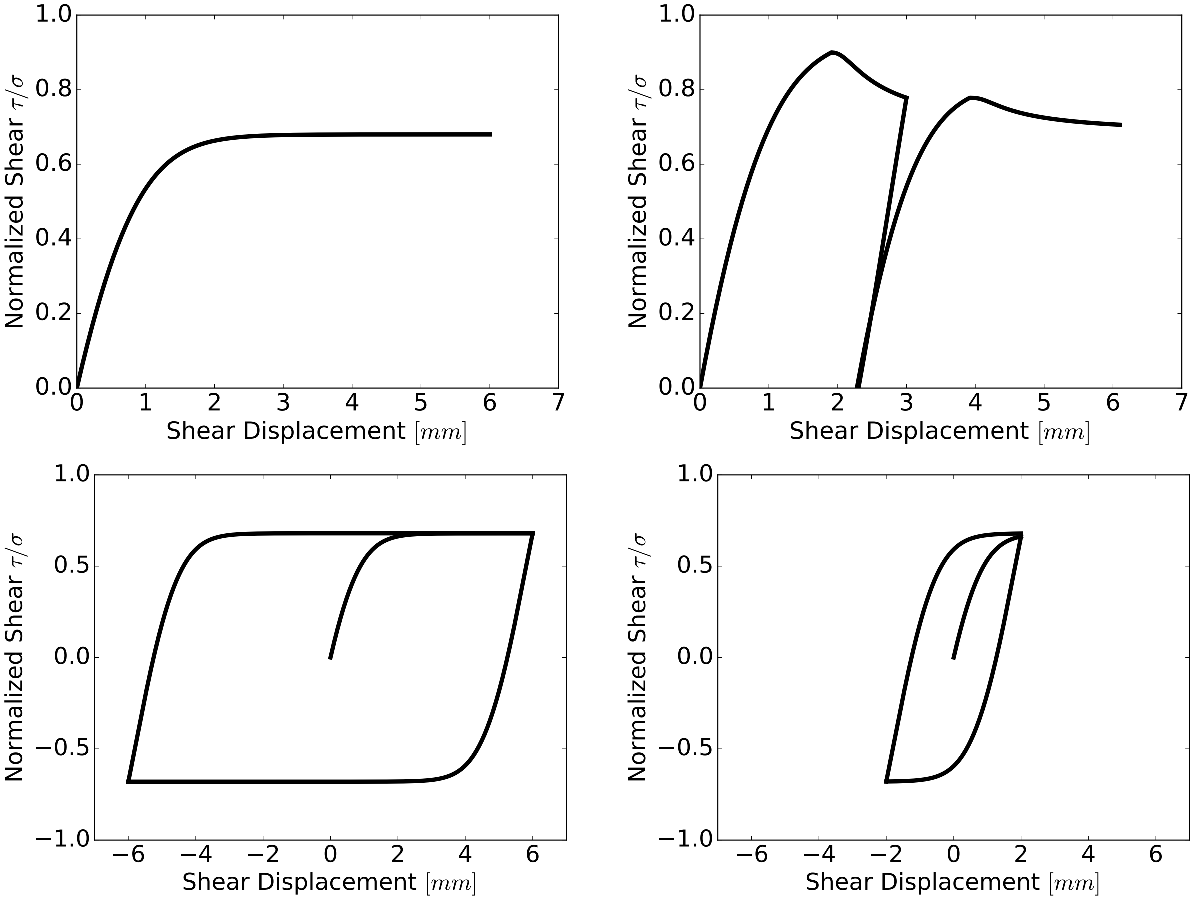


Figure 104.44: Response of Non-Linear Hardening Plastic Contact/Joint/Interface (NLHS) model with normal stress of $100kPa$, residual normalized shear stress of $\mu_r = 0.68$, shear stiffness $k_t = 400kPa$ and shear zone length $SZ_h = 5mm$.

104.7.3.4 Nonlinear Hardening Softening Shear (NLHSS) Model

In this model, the normalized shear stress hardening/softening variable μ increases from 0 to its peak limit μ_p and then with more shear displacement reaches to the residual normalized shear stress of μ_r using a non-linear Armstrong Frederick type hardening/softening law. The evolution of the hardening/softening variable μ during hardening phase is given as

$$\Delta\mu = k_t \Delta\gamma^p - \frac{k_t}{\mu_p}|\Delta\gamma^p|\mu \tag{104.596}$$

where k_t is the non-linear hardening variable and $\Delta\gamma^p$ is the the plastic part of incremental shear strain $\Delta\gamma$. Once the peak normalized shear stress is attained, the material starts to soften. The softening behavior is modeled as reduction of normalized shear stress ratio μ as

$$\Delta\mu = -\frac{n * b(\mu_p - \mu_r)}{(\pi/2)^n \theta^{1/n-1}} * cos^2\theta\Delta\gamma^p \tag{104.597}$$

$$\theta = \frac{\mu_p - \mu}{\mu_p - \mu_r}(\pi/2)^n \tag{104.598}$$

where b is the softening rate, $\Delta\gamma^p$ is the incremental plastic shear strain and n represents the size of the peak plateau as shown in Figure 104.45. This incremental form of softening phase is derived from the inverse tangent function raised to power n as

$$f = a * (arctan(b * \gamma^P))^n \tag{104.599}$$

where a is a constant parameter of the function and in Equation 104.598 is equal to $(\mu_p - \mu)/(\mu_p - \mu_r)$. The softening rate parameter b represents the rate at which the normalized shear stress μ_p decreases with further application of shear displacement Δ_t as shown in Figure 104.45. A larger value of b would result in faster decay. The size of peak plateau parameter n determines the size of the plateau formed at the peak as shown in Figure 104.45. A larger value of n would result in a larger plateau. It must be noted that the peak plateau size parameter n also influences the overall rate of softening as could be seen from the derived incremental Equation 104.598. The parameters n and b can be calibrated from monotonic shearing tests. Figure 104.46 shows the response of the model with the peak plateau size parameter $n = 4$ and softening rate parameter $b = 40$.

The yield function (f) is again given as

$$f := \tau - \mu\sigma_n \leq 0 \tag{104.600}$$

where μ is a normalized shear stress hardening/softening variable evolved by Equation 205.17 and 104.598, τ is the shear stress and σ_n is the normal stress.

Since in the model, the hardening law is defined as Armstrong-Frederick type, the peak shear stress ratio μ_p is defined as only 95% of the asymptotic limit of the Equation 205.17. Thus, the asymptotic

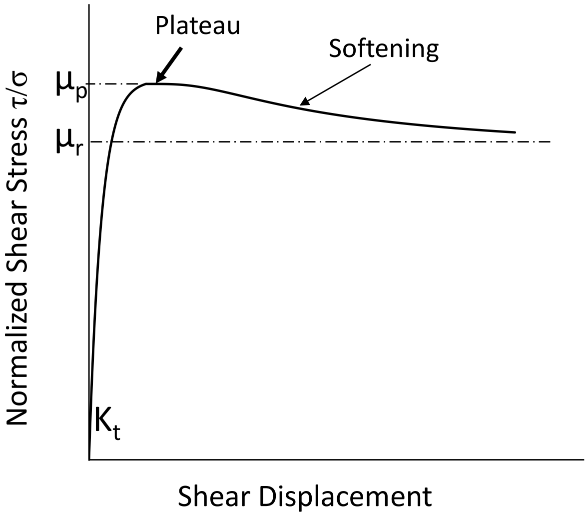


Figure 104.45: Non-Linear Hardening Softening Shear model parameters

limit of Armstrong-Frederick type hardening is raised by a factor of $1/0.95$ times the peak shear stress ratio μ_p.

In this model during cyclic shearing, it is assumed that once the peak strength is passed, the material would not be able to again attain the peak strength during cyclic loading. It is based on the assumption that as the material passes the peak strength, particle breakage and smoothening of the surface takes place which cannot be recovered back by any process. Thus as the material softens, the peak coefficient of friction μ_p is iteratively redefined to the coefficient of friction μ in the softening phase. This effect can also be observed from the tests performed by Uesugi et al. (1989) as shown in Figure ??(b). As soon as the peak shear stress ratio μ_p degrades to residual strength in the first cycle, other cycles follow the residual shear stress ratio μ_r.

Figure 104.44 shows the performance of NLHSS model for different loading conditions. The response of the model is very close to the realistic interface behavior observed by Uesugi et al. (1989) as shown in Figure ??(b) . During cyclic shearing shown in Figure 104.44(c), the model predicts the peak behavior only in the first cycle. After that, the response is governed by the residual normalized shear stress μ_r. Figure 104.44(c) shows the cyclic shearing behavior when the residual normalized shear stress μ_r is not attained in the first cycle. As a result of unloading and reloading, it again attains the last peak normalized shear stress μ_p that it had attained during the softening phase.

NLHSM can be extended further to model the variation of peak normalized shear stress μ_p for different normal stress σ_n conditions as observed in tests by Fakharian and Evgin (1996); Shahrour and Rezaie (1997); Evgin and Fakharian (1997); Fakharian and Evgin (1997); Fakharian (1996); Fakharian

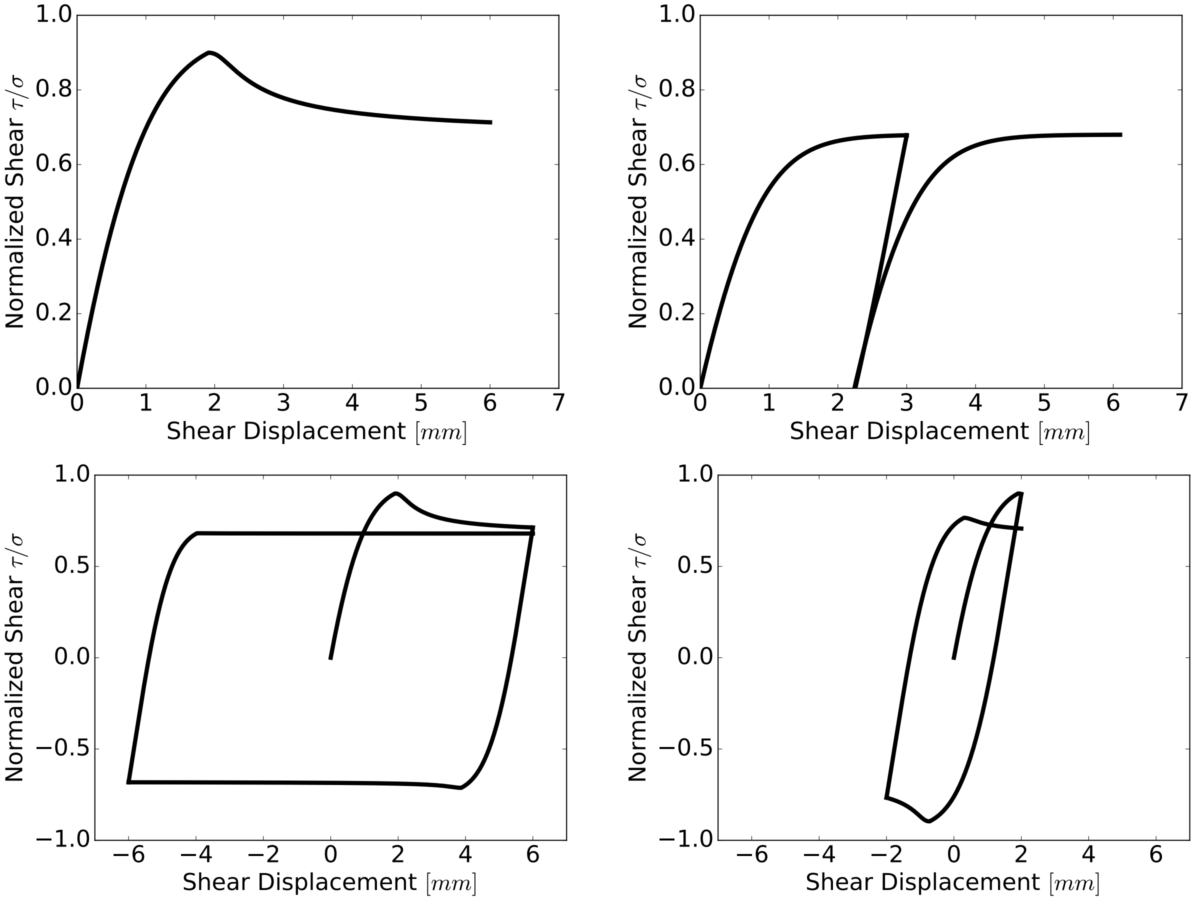


Figure 104.46: Response of Non-Linear Hardening Softening Shear (NLHSS) model with normal stress of $100kPa$, residual normalized shear stress $\mu_r = 0.68$, peak normalized shear stress $\mu_p = 0.9$, shear stiffness $k_t = 800kPa$, peak-plateau parameter $n = 4$, softening rate parameter $b = 40$ and shear zone length $SZ_h = 5mm$.

et al. (2002). For a given relative density D_r of soil in the sheared zone, the peak normalized shear stress μ_p can be generalized to be a logarithmic function of normal stress σ_n as

$$\mu_p = \mu_{p_0} - k * log(\sigma_n/P_0) \tag{104.601}$$

where μ_{p_0} is the peak normalized shear stress at normal stress of $\sigma_n = P_0$, k is the peak normalized shear stress rate of decrease and P_0 is the reference stress of $P_0 = 100kPa$. This is similar to the Bolton (1986) stress-dilatancy relationship observed in sands. In the above Equation 104.601, for the normal stress of $\sigma_n \leq P_0$, the peak normalized shear stress μ_p would become greater than μ_{p_0} as the term $log(\sigma_n/P_0)$ becomes negative. This would result in high peak normalized shear stress μ_p for low confining stress σ_n. As a result, the peak normalized shear stress μ_p needs to be restricted to a value. In this model, it is assumed that μ_{p_0} would also act as the maximum possible peak normalized shear stress μ_p for low normal stresses or confinement. Thus the above Equation 104.601 can be re-written with the limit on peak normalized shear stress μ_p as

$$\mu_p = max(\mu_{p_0}, \mu_{p_0} - k * log(\sigma_n/P_0)) \tag{104.602}$$

where μ_{p_0} also represents the maximum peak normalized shear stress μ_p^{max}, that the sheared zone soil could attain. Thus, with experiments conduction for different normal stress σ_n, the peak normalized shear stress μ_p can be calibrated as the function of normal stress σ_n with peak normalized shear stress limit μ_{p0} and peak normalized shear stress rate of decrease k. Figure ?? shows the response of the model for different normal loads of $100kPa$, $250kPa$ and $500kPa$ and also validates the model with the experimental results from Fakharian and Evgin (1996) as shown in Figure 104.34.

Extending The Models to 3D . Section ?? described the model in 2-D in $\tau - \sigma_n$ space, to make the reader understand the basics of the model. The model can be easily extended to 3-D using the back stress variable α instead of the normalized shear stress μ for NLHS and NLHSS models. In 3-D, there would be normal stress component σ_n in contact/interface normal direction n and two tangential stress components τ_1 and τ_2 in tangential contact/interface plane in directions 1 and 2 respectively. Similarly, the shear strain γ has two components as γ_1 and γ_2 in the two tangential directions 1 and 2 respectively. As stated earlier, since normal interface behavior is assumed to be non-linear elastic, the plastic strains are only developed in shear. Thus the incremental plastic strain $\Delta\gamma^p$ and its magnitude $|\Delta\gamma^p|$ in these models is defined as

$$|\Delta\gamma^p| = \sqrt{\Delta\gamma_1^p \Delta\gamma_2^p} \tag{104.603}$$

$$\Delta\gamma^p = [\Delta\gamma_1^p, \Delta\gamma_2^p] \tag{104.604}$$

where $\Delta\gamma_1^p$ and $\Delta\gamma_2^p$ represents the incremental plastic shear strain components in contact/interface tangential (shear) direction 1 and 2 respectively. The plastic flow direction m is defined as the direction of incremental plastic shear strain $\delta\gamma^p$ and thus is defined as $m = \delta\gamma^p/|\Delta\gamma^p|$

104.7.3.5 EPPS Model

EPPS Model described in Section 104.7.3.2, is the simplest Mohr Coulomb type elastic perfectly-plastic model. The yield function (f) is defined as

$$f := (\tau_1/\sigma_n - \mu_r)^2 + (\tau_2/\sigma_n - \mu_r)^2 = 0 \tag{104.605}$$

where σ_n is the normal stress, τ_1, τ_2 are the shear stress and μ_r is the residual friction coefficient. Being an elastic-perfectly plastic model with no internal variables, there is no hardening/softening evolution rule.

104.7.3.6 NLHS Model

The non-linear hardening shear model described in Section 104.7.3.3 is modeled in 3-D using the back stress internal variable α. The yield function (f), is defined as

$$f := (\tau_1/\sigma_n - \alpha_1)^2 + (\tau_2/\sigma_n - \alpha_2)^2 = 0 \tag{104.606}$$

where σ_n is the normal stress, τ_1, τ_2 are the shear stress and α_1, α_2 are the back stress components in contact/interface tangential direction 1 and 2 respectively on the contact-interface plane. The hardening law would be then defined as

$$\Delta\alpha = k_t \Delta\gamma^p - \frac{k_t}{\mu_p}|\Delta\gamma^p|\alpha \tag{104.607}$$

$$\Delta\alpha = H_m|\Delta\gamma^p| \tag{104.608}$$

$$H_m = k_t m - \frac{k_t}{\mu_p}\alpha \tag{104.609}$$

where μ_p is the peak normalized shear stress, which depends upon the normal stress σ_n as stated in Equation 104.601 and k_t represents the initial elastic shear stiffness of soil-structure interface. H_m represents the non-linear Armstrong-Frederick type hardening modulus and m represents the plastic flow direction.

104.7.3.7 NLHSS Model

The non-linear hardening softening shear model described in Section 104.7.3.4 is also modeled in 3D using the back stress internal variable α. The yield function (f) is defined as Equation 104.606. The hardening evolution law for the back stress α is defined by Equation 104.609. The softening law is defined as Von-Mises type as

$$\Delta\alpha = -\frac{n * b(\mu_p - \mu_r)}{(\pi/2)^n \theta^{1/n-1}} * cos^2\theta \Delta\gamma^p \tag{104.610}$$

$$\theta = \frac{\mu_p - |\alpha|}{\mu_p - \mu_r}(\pi/2)^n \tag{104.611}$$

$$\Delta\alpha = S_m \Delta\gamma^p \tag{104.612}$$

$$S_m = -\frac{n * b(\mu_p - \mu_r)}{(\pi/2)^n \theta^{1/n-1}} * cos^2\theta \tag{104.613}$$

where μ_r is the residual normalized shear stress that is constant and depends upon the soil and structure material, μ_p is the peak normalized shear stress at the start of softening phase and S_m represents the softening modulus.

As stated in Section 104.7.3.4, for NLHSS model, the peak normalized shear stress μ_p of the material is iteratively defined to the back stress $\mu_p = |\alpha|$ in softening phase. This means that the dilatancy surface squeezes towards the critical surface as the sheared zone soil at interface continues to shear.

Using the incremental Equation 104.609 Equation 104.613 presented for both hardening and softening phase respectively, the 3-D model can be integrated. using the elastic-plastic theory Hill (1950); Temam (1985); Wu Tai (1966); Lubliner (1990); de Borst and Feenstra (1990); de Borst (1987). The elastic-plastic stiffness or consistent tangent stiffness Jeremić (1994); Crisfield (1987) can then be computed easily at each loading increment or iteration. Section ?? presents the elastic-plastic integration for the interface models.

104.8 Inelastic Behavior and Models for Rock

104.8.1 Overview of Intact Rock Behavior

(Stavrogin et al., 2001), (Fairhurst, 2003), (Mogi, 2006)

Rock, and other geomaterials, feature a distinct set of material behavior that separates them from other natural and/or man made materials. Rock material subjected to shock waves in particular shows a variety of response regimes and behavior that warrant further discussion. Of particular interest are the following specifics of rock behavior: pressure sensitivity, dilative and compactive response, inherent and induced anisotropy, full coupling of porous rock solid with pore fluid and temperature fields and bifurcation response, resulting in shear and compaction bands. Each of these aspects of rock behavior will be described in sections below.

It is also very important to note that this study will focus on behavior of intact rock material, while main focus of research in rock mechanics is on behavior of jointed rock masses. While behavior of discontinuous, jointed rock mass is very important for construction in rock (tunnels, dams, foundations...), behavior of intact rock mass becomes very important for any modeling and simulation of deeper rock structures, particularly where strong shock waves are involved. This is emphasized by the fact that rock blocks do behave like solid for high pressures. For example, during shock loading blocks might fuse at contact/interface and behave like a solid, in which case the inelastic (elastic-plastic) behavior of intact rock mass has major impact on overall rock mass response.

This section uses selection of published results to emphasize distinct features of mechanical response of rock that are considered important for proper modeling of shock wave propagation. While there exist a significant body of published work on behavior of intact rock (which is still much smaller than body of published work on behavior of rock as discrete media), chosen here are publications and results that provide important results used to emphasize distinct features of mechanical behavior of rock. Emphasizing these distinct feature of rock behavior is important for a number of reasons. Two main reasons are that while the shock wave theory was developed over last two centuries (see brief overview in section 104.8.3) main focus was on ideal domains (linear elastic solids or fluids), rock features many distinct modes of mechanical response that demand use of high fidelity numerical modeling. In addition to that, even when shock wave theories were developed for inelastic (elastic-plastic) solids, this was done for metals, elastic-plastic response of which lacks many features found in rock (geomaterial) response (pressure sensitivity, anisotropy, dilative and compactive response...)

104.8.1.1 Pressure Sensitivity

Rock material response shows strong pressure sensitivity. Both initial yielding response, inelastic response and ultimate strength strongly depend on confinement pressure experienced by rock material. Confinement pressure acting on rock can be inherent, coming from location of rock mass (depth) and from geologic factors (tectonics). Fairhurst (2003) details initial stress determination procedures. Initial stress determination (stress level from previous, historical loading stages, tectonics, erosion) is very important in view of pressure sensitivity and elastic-plastic response of rock. Depending on the type of rock material, the effects of confinement pressure on response vary. Figures 104.47 – 104.55) show rest data for a full elastic-plastic response of various rock specimens for varying confinement pressures. Both axial (ϵ_1) and radial (ϵ_2) strains are shown versus axial stress ($\Delta\sigma_1$). It is very important to recognize a number of distinct features of rock behavior that these test show:

- Varying confinement pressure influences yield stress, that is an increase of yield stress is observed with the increase in confinement pressure,
- Elastic stiffness increases with increase in confinement pressure,
- Ductility increases with increase in confinement pressure,
- Increase in confinement pressure significantly influences final strain level achieved
- Volume change (dilation, as observed from radial strain results (ϵ_2) is significant, however, in some cases it increases and in some decreases with confinement pressure increase (as observed in Figures 104.47 – 104.52)
- Elastic degradation/damage is increasing with cycles of loading (Fig. 104.54 for example, reduction of loading-unloading-reloading slope signifies elastic damage)
- Presence of energy dissipation in rock for various confinement pressures (area of hysteresis loops, see for example Fig. 104.53 – 104.55)

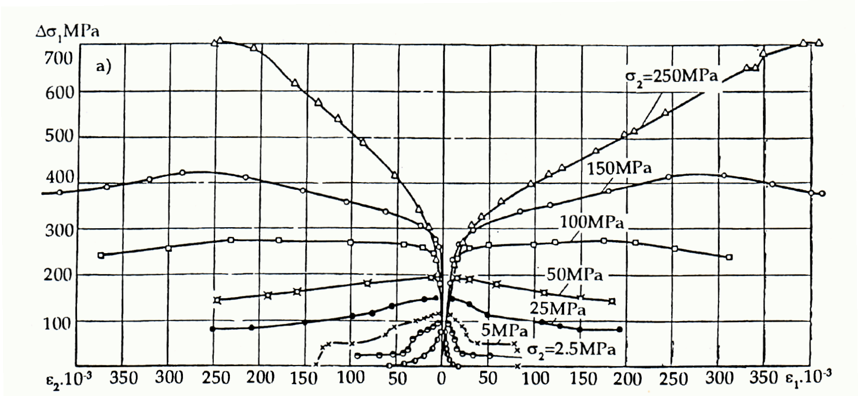


Figure 104.47: Full elastic-plastic response of marble specimens for varying confinement pressures. Shown are axial (ϵ_1) and radial (ϵ_2) strains versus axial stress ($\Delta\sigma_1$) for triaxial loading of 3D samples (Stavrogin et al., 2001).

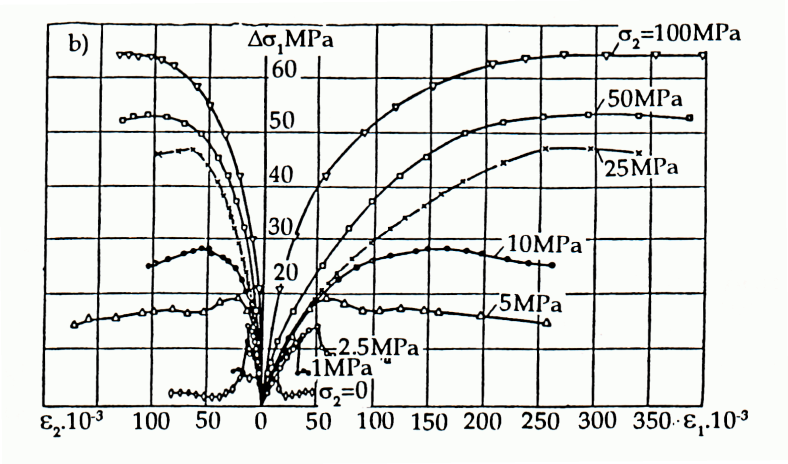


Figure 104.48: Full elastic-plastic response of lignite specimens for varying confinement pressures. Shown are axial (ϵ_1) and radial (ϵ_2) strains versus axial stress ($\Delta\sigma_1$) for triaxial loading of 3D samples (Stavrogin et al., 2001).

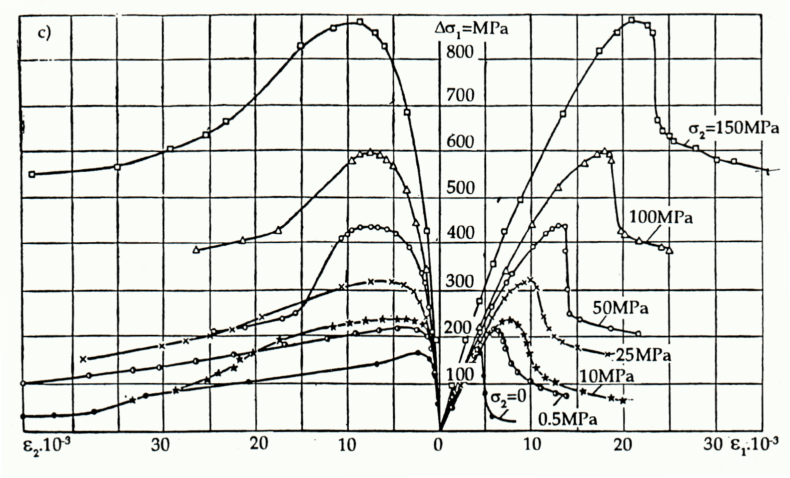


Figure 104.49: Full elastic-plastic response of granite specimens for varying confinement pressures. Shown are axial (ϵ_1) and radial (ϵ_2) strains versus axial stress ($\Delta\sigma_1$) for triaxial loading of 3D samples (Stavrogin et al., 2001).

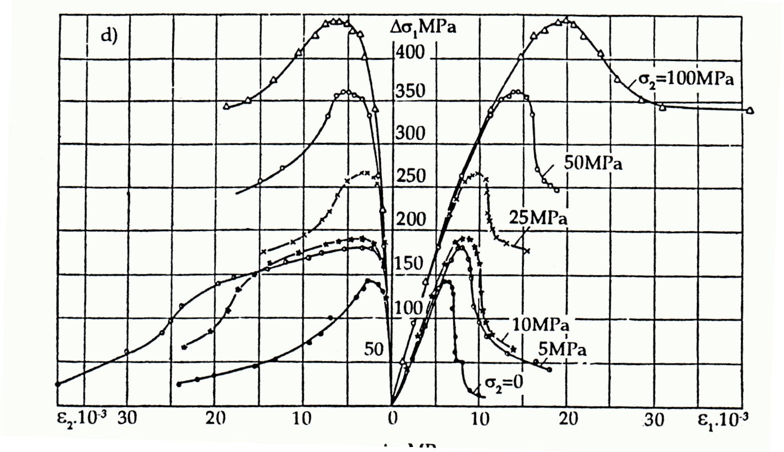


Figure 104.50: Full elastic-plastic response of sandstone (non-burst prone) specimens for varying confinement pressures. Shown are axial (ϵ_1) and radial (ϵ_2) strains versus axial stress ($\Delta\sigma_1$) for triaxial loading of 3D samples (Stavrogin et al., 2001).

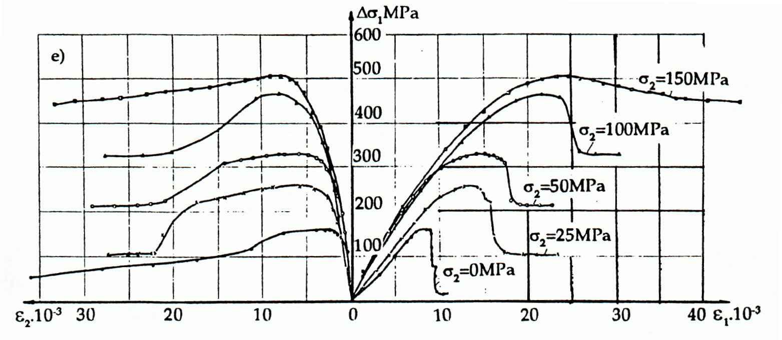


Figure 104.51: Full elastic-plastic response of sandstone (burst prone) specimens for varying confinement pressures. Shown are axial (ϵ_1) and radial (ϵ_2) strains versus axial stress ($\Delta\sigma_1$) for triaxial loading of 3D samples (Stavrogin et al., 2001).

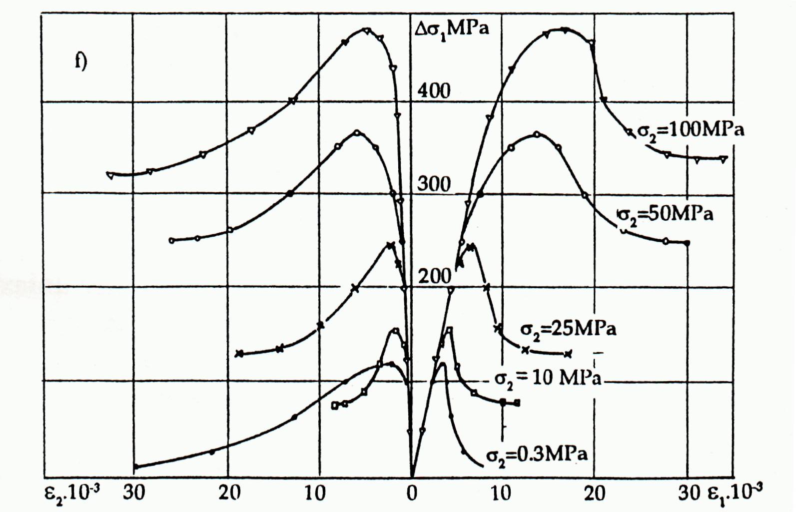


Figure 104.52: Full elastic-plastic response of sulphidic ore specimens for varying confinement pressures. Shown are axial (ϵ_1) and radial (ϵ_2) strains versus axial stress ($\Delta\sigma_1$) for triaxial loading of 3D samples (Stavrogin et al., 2001).

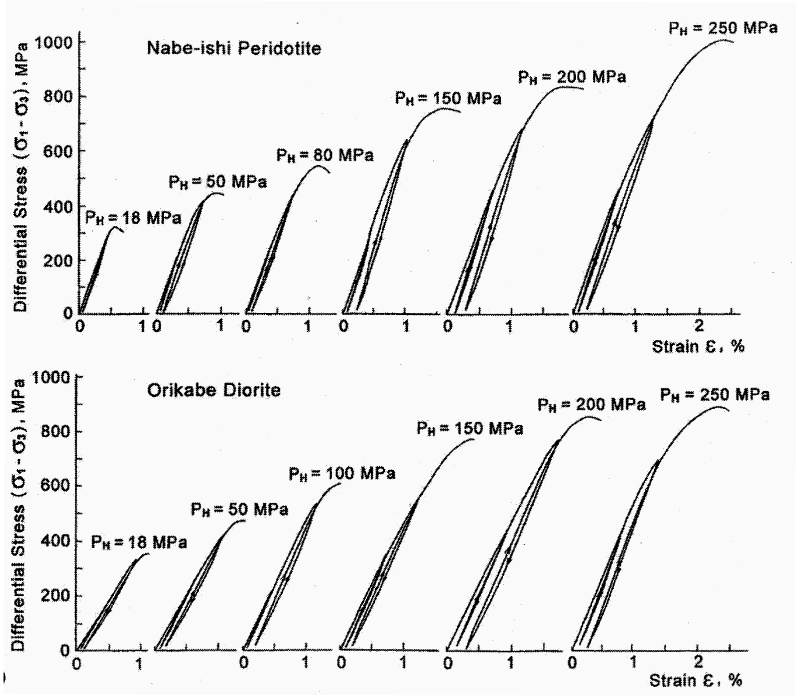


Figure 104.53: Full elastic-plastic response, with loading-reloading cycles of Periodite (upper) and Diorite (lower) specimens for varying confinement pressures. Shown are axial (ϵ_1) strains versus differential stresses ($\sigma_1 - \sigma_2$) for triaxial loading of 3D samples (Mogi, 2006).

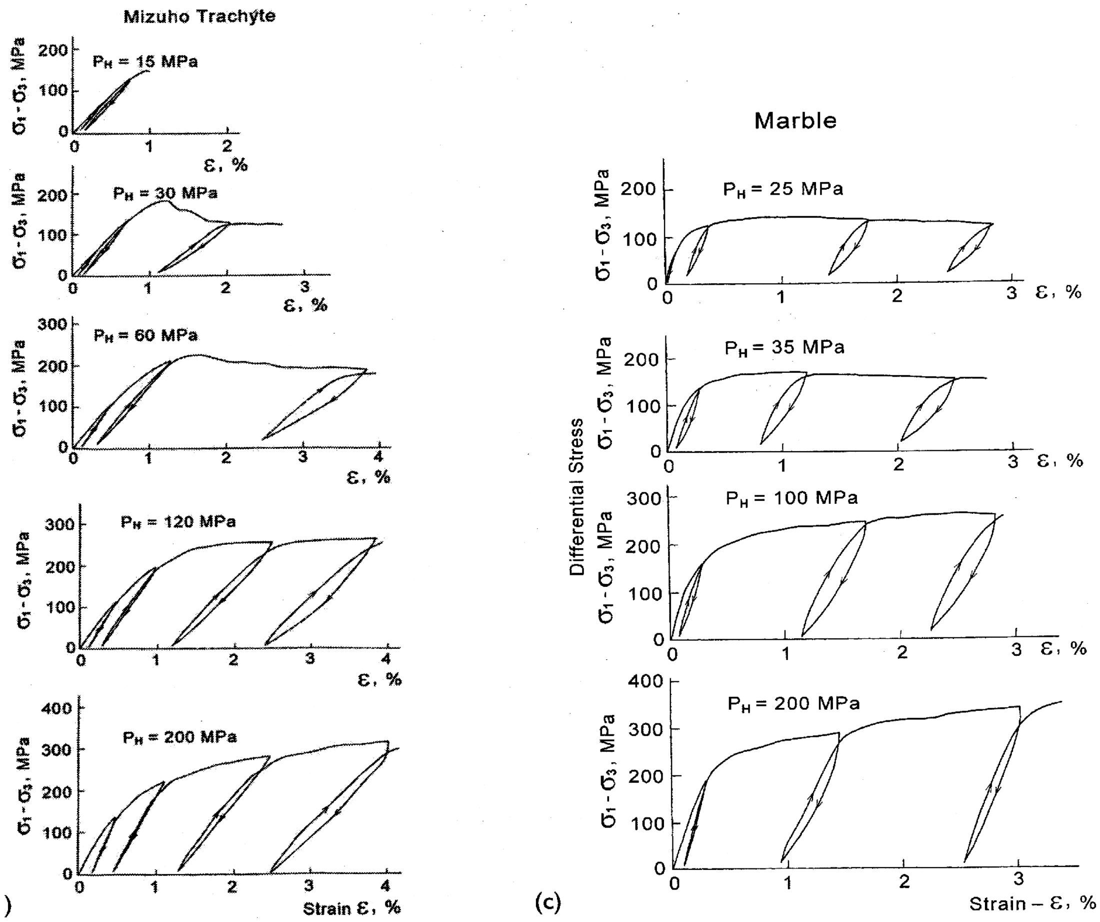


Figure 104.54: Full elastic-plastic response, with loading-reloading cycles of Trachite (left) and Marble (right) specimens for varying confinement pressures. Shown are axial (ϵ_1) strains versus differential stresses ($\sigma_1 - \sigma_2$) for triaxial loading of 3D samples (Mogi, 2006).

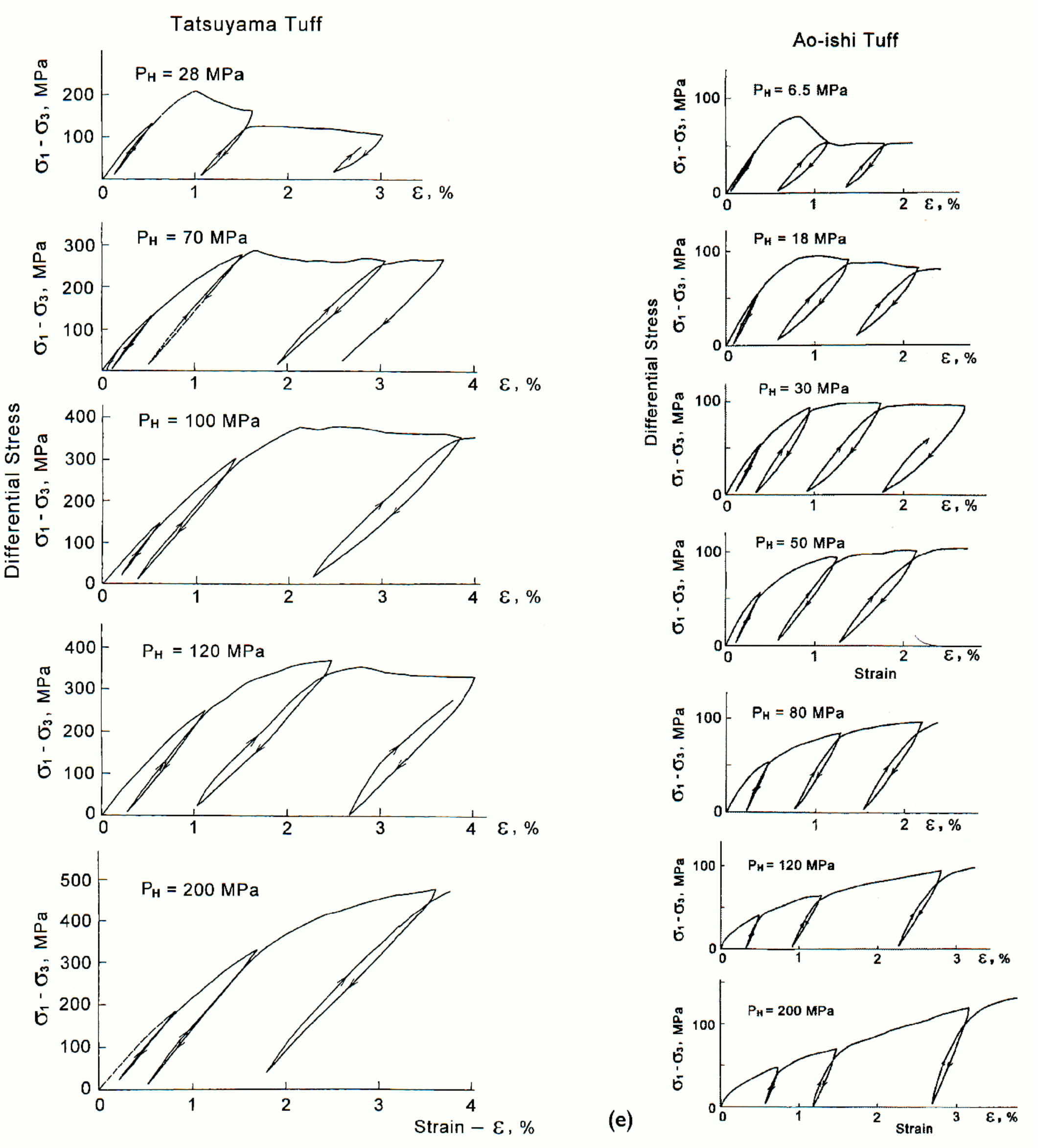


Figure 104.55: Full elastic-plastic response, with loading-reloading cycles of Tuff specimens (two different rock sources) for varying confinement pressures. Shown are axial (ϵ_1) strains versus differential stresses ($\sigma_1 - \sigma_2$) for triaxial loading of 3D samples (Mogi, 2006).

104.8.1.2 Dilative and Compactive Response

Rock (similar to other geomaterials) feature dilative (increase in volume) and compactive (decrease in volume) response for both hydrostatic confinement as well as for deviatoric loading (shear stresses). Lockner and Stanchits (2002) performed a number of tests on (initially isotropic) sandstone and measured the undrained poroelastic response for changes in mean (hydrostatic, normal) and deviatoric (shear) stress. While change in pore pressure was found to result from changes in mean (hydrostatic) stress, it was also resulting from changes in deviatoric stress. This dependence of pore pressure, and consequentially pore volume, on deviatoric stress is called dilatancy and is a feature found in most geomaterials. Figure 104.56 shows measured dependence of a coefficient η, defined as a ratio of change in pore pressure due to change in deviatoric stress ($\Delta p = -\eta \Delta \sigma_{deviatoric}$).

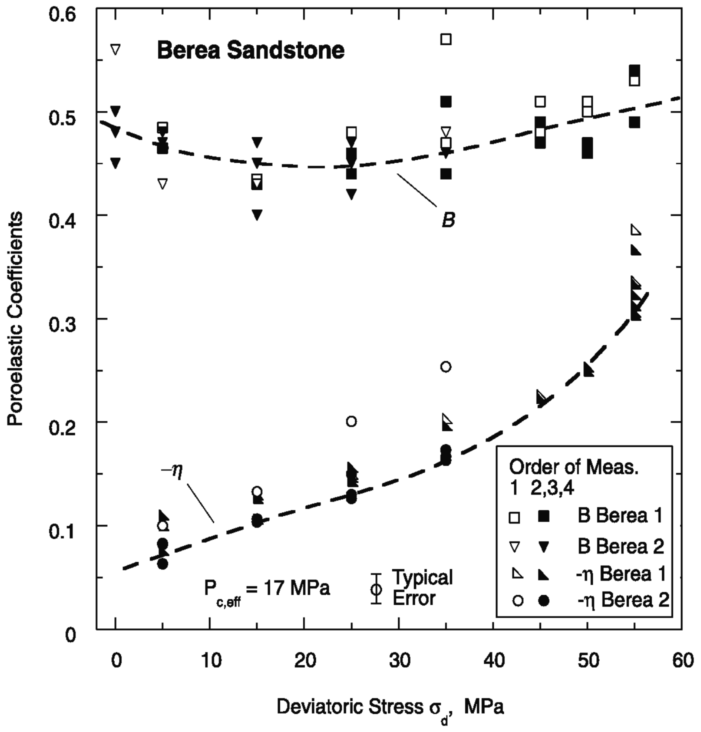


Figure 104.56: Measured poroelastic (Skempton's) coefficient B and η for Berea Sandstone. (Lockner and Stanchits, 2002).

Figure 104.56 also shows measured values for coefficient B, which defines a ratio of change in pore pressures due to change in mean stress ($\Delta p = -\eta \Delta \sigma_{mean}$). Coefficient B and η are also known as Skempton's coefficients (Skempton, 1954). Hamiel et al. (2005) developed a model based on data provided by Lockner and Stanchits (2002) that captures dilatancy in poroelastic regime. In addition to that, one of their models allows for variability in Skempton's coefficients and seems to capture test data quite well. While presented development focuses on seismic pore fluid pressure development (positive and negative) it has modeling and simulation implications for other dynamic, rock events where pores are filled with fluid and dilatancy is involved (which is always the case).

Dilatant response in elastic-plastic regime can be observed in Figures 104.47 – 104.52. In particular, volumetric strain calculations ($\epsilon_{volumetric} = \epsilon_{ii} = \epsilon_1 + 2\epsilon_2$) reveal that dilatant response is present during elastic phase of loading (before yielding), significant dilation occurs after plastic limit. Thus we can conclude that while dilatancy is present in elastic phase of loading (as concluded by Lockner and Stanchits (2002)) dilatancy is even more pronounced in elastic-plastic loading regime.

In the limit of compactive and dilative response is the localized compactive/explosive deformation. Olsson (1999, 2001) details recent findings of compaction bands. They are thin zones of pure compressional deformation with very low permeability and porosity. porosity drop for such zones is on the order of ten times $(10\times)$ when compared to porosity of surrounding rock (Olsson, 1999). In addition to that, compaction zones (bands) growth is described in terms of shock wave analysis. Issen and Rudnicki (2001) developed a more general theory of compaction band formation, particularly in view of plasticity models with cap. Plasticity models with cap are necessary if realistic behavior of rock is to be modeled, covering a wide range of stress states, from tension, shear to compression. Borja (2006) reviews analytical conditions for appearance of volume implosion/explosion (diffuse process) and compaction bands (localized process). He shows that general conditions for localization of deformation can be used for both shear as well as for compaction and dilative localization of deformation.

104.8.1.3 Anisotropy

Various rock types feature directional features (bedding, foliation, flow structures) which are reflected in anisotropic elastic and plastic properties. Anisotropy of rock behavior can be inherent and/or induced. Inherent anisotropy is present before the current loading is applied and is most likely resulting from past geological processes in rock (past loading). Induced anisotropy results from current loading processes and can significantly change orientation and value of elastic constants from inherent values (Amadei and Goodman, 1982; Amadei, 1983). Anisotropy can also be apparent, when bedding planes obvious and hidden when bedding planes not directly observable. Measurements of anisotropy are not readily available in literature as such tests are quite involved and complex. For example Pariseau (2006) reports that measurements indicate that Young's modulus is parallel to bedding plane is often about twice as large as Young's modulus perpendicular to bedding plane. Figure 104.57 shows usual test setup when rock cores are extracted with different orientations to the test loading. For such anisotropic test setups, variations of axial Young's modulus are shown in Figure 104.58.

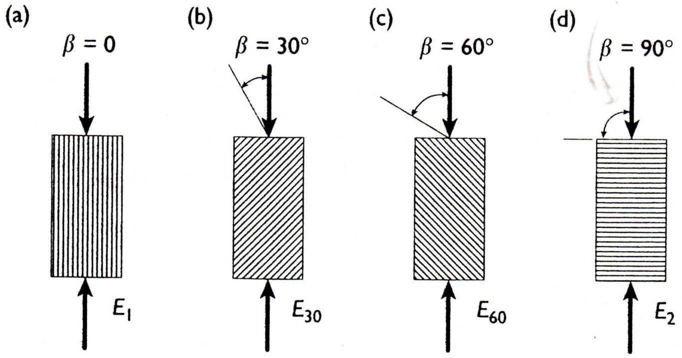


Figure 104.57: Test setup for measuring elastic anisotropy for rock cores extracted at different orientations (Pariseau, 2006).

In addition to influence of anisotropy on elastic constants, permeability is significant influenced by anisotropy of rock. A.Angabini (2003) shows significant influence of rock anisotropy on both elastic properties and on anisotropic permeability. He used 438 samples with distinct orientations to measure isotropic and anisotropic material properties out of research wells in The Netherlands along of 2.6 km test section and going to depths of up to 1.6 km. His measurements indicate that in addition elastic anisotropy, permeability anisotropy is significant. Figure 104.59 shows differences (variation) in measured specific permeabilities at different depths for wells in The Netherlands.

In addition to significant anisotropy in permeability (for example at depth 345m permeability in

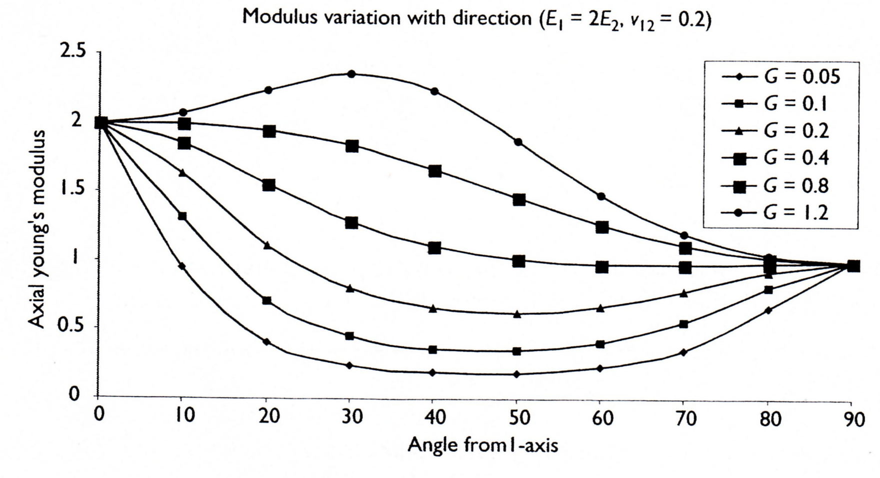


Figure 104.58: Anisotropic variations of axial Young's moduli for different orientation of core samples (Pariseau, 2006).

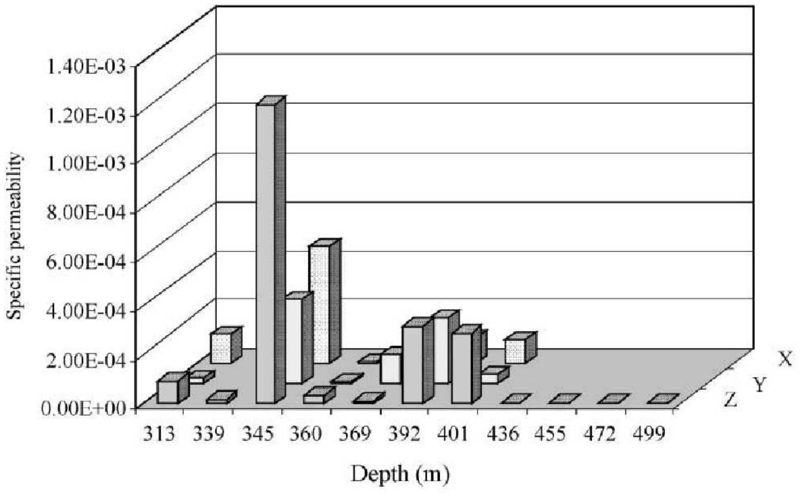


Figure 104.59: Measured specific permeabilities at different depths for wells in The Netherlands, parallel to bedding (x, y) and perpendicular to bedding (z) (A.Angabini, 2003).

vertical direction (Z) is three times higher than the one in X and 5 times higher than permeability in Y direction), significant spatial variation of permeability is present. For example, within 6 meters (between depth of 339m and 345m difference is two orders of magnitude!

Measured anisotropy of elastic and elastic-plastic properties is also significant. For example, Figure 104.60 shows directional elastic moduli (obtained using either static or dynamic tests, for dry and saturated conditions) for mudstone and sandstone.

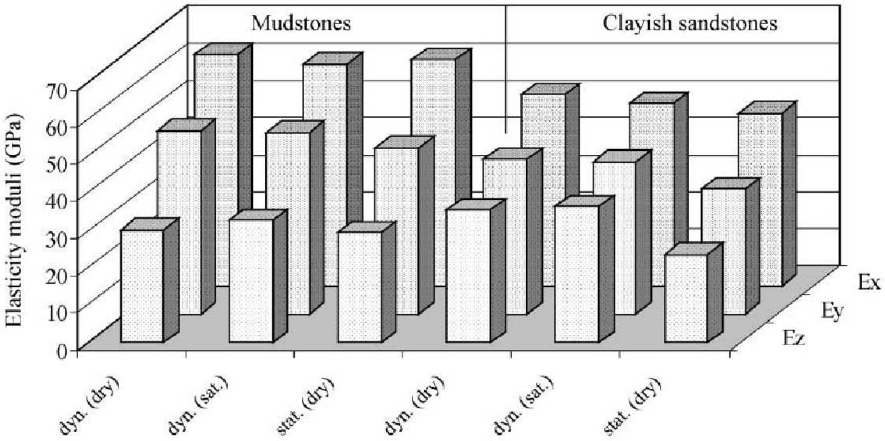


Figure 104.60: Measured anisotropic elastic moduli for different rock samples (Mudstone and Clayish sandstone) for wells in The Netherlands. Young's modulus parallel to bedding (x, y) and perpendicular to bedding (z) (A.Angabini, 2003).

Differences between elastic moduli in different directions of two or more are obvious. In addition to that, since elastic moduli in all three directions are different, rock exhibits general anisotropy (as opposed to cross anisotropy, where two elastic moduli would be the same).

Similar to elastic moduli (Young's moduli), Poisson's ratios do show significant anisotropy as shown in Figure 104.61.

In addition to elastic anisotropy (as shown in Figures 104.60 and 104.61) uniaxial strength also shows very strong directional dependence as shown in Figure 104.62.

Anisotropy of rock is not localized to one region of the domain, rather it is present in every level. Figure 104.63 shows elastic anisotropic moduli for wells in the Netherlands test area for a full depth of 1.7km. It is important to note that in addition to anisotropy that extends throughout depth. inherent uncertainty on measured moduli is present as well. This uncertainty (spread of data points for same depth) will be discussed in some more details in section 104.8.2.

In view of importance of anisotropy on rock response, determination of anisotropy is very important if simulations are to be used in predicting rock behavior. This is particularly true for modeling and simulation of shock wave propagation as anisotropy (both in elasticity and permeability) can significantly influence results.

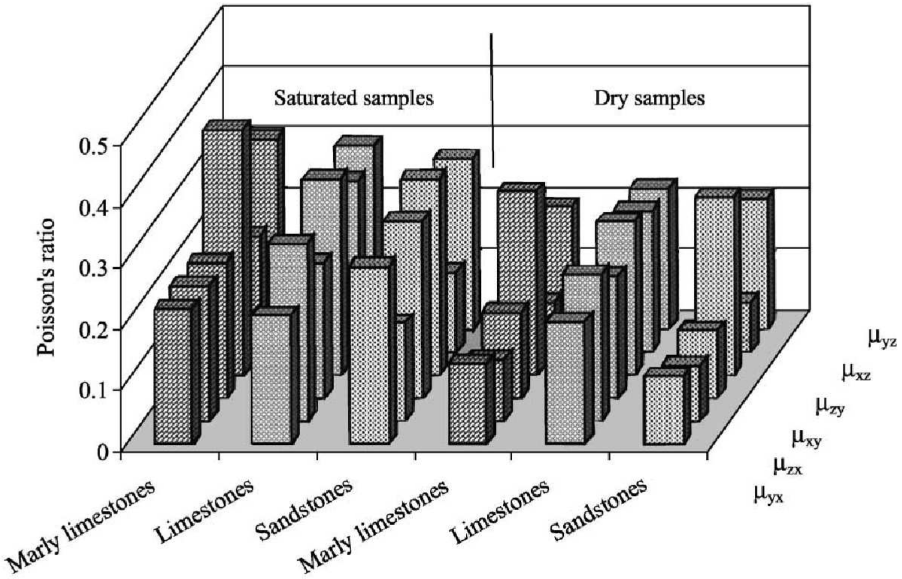


Figure 104.61: Dynamic Poisson's ratios for different rock samples (limestone and sandstone) in dry and saturated conditions, for wells in The Netherlands, (A.Angabini, 2003).

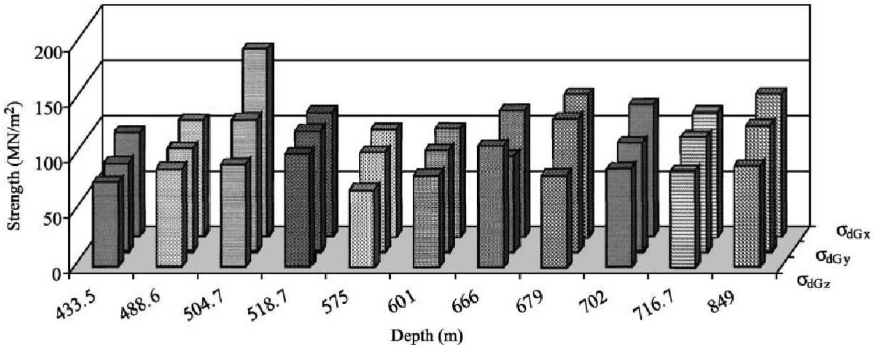


Figure 104.62: Uniaxial strength of mudstone for three different orientations of core sample, for various sampling depths, for wells in The Netherlands, (A.Angabini, 2003).

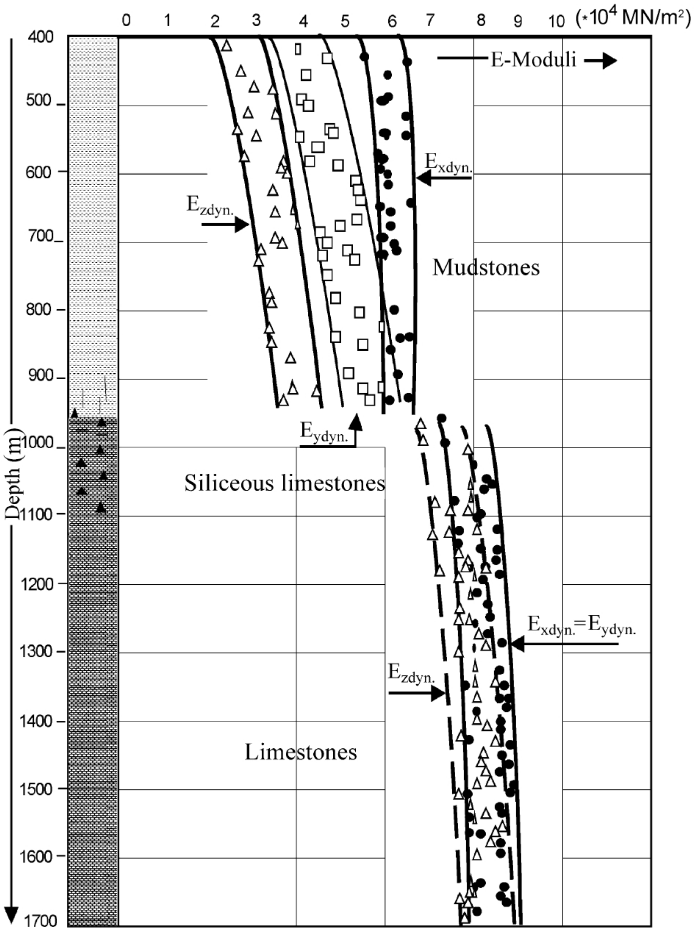


Figure 104.63: Anisotropic properties of mudstone and limestone an three directions at different depths (full depth profile), for wells in The Netherlands, (A.Angabini, 2003).

104.8.1.4 High Rate Elastic-Plastic Loading

Experimental data presenting (controlled) high rate elastic-plastic loading is fairly limited compared to the abundance of low rate data. It is very important to note that high rate controlled tests are of much value as they allow separation of high rate constitutive response from shock loading tests that are accomplished using high energy shock application to rock samples, which then initiate many other facets of multi-physics of rock behavior (described in more detail in section 104.8.3 on page 354). One of the best controlled high loading rate data sets were published by Stavrogin and Protosenya (1983); Stavrogin and Pevzner (1983); Stavrogin et al. (2001). A wide range of loading rates (strain rates, ranging over 12 orders of magnitude), presented (Stavrogin and Protosenya, 1983; Stavrogin and Pevzner, 1983; Stavrogin et al., 2001), allows development of qualitative understanding and development of quantitative model parameters for rock material. Figures 104.64, 104.65 and 104.66 show the influence of rate of loading on peak strength and elastic limit (yield stress) for rock samples (marble, diabase and sandstone respectively) confined to different pressures,

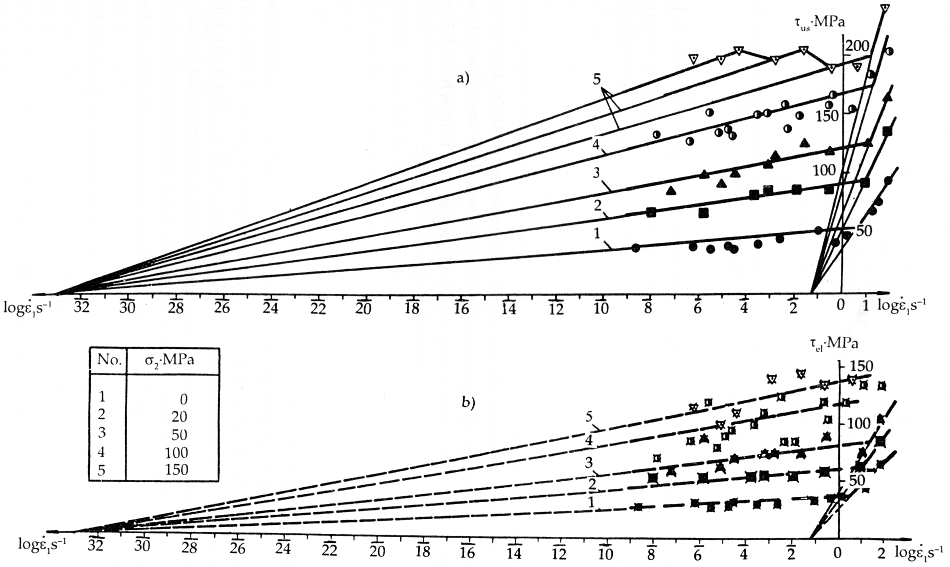


Figure 104.64: Marble: dependence of peak strength (a) upper) and elastic limit (yield stress) (b) lower) on loading rate ($\dot{\epsilon}$) for different confining stresses ($1 - \sigma_2 = 0$ Mpa; $2 - \sigma_2 = 20$ Mpa; $3 - \sigma_2 = 50$ Mpa; $4 - \sigma_2 = 100$ Mpa; $5 - \sigma_2 = 150$ Mpa) (Stavrogin and Protosenya, 1983).

While highest loading (strain) rates are on the order of $\dot{\epsilon} = 10^2$ 1/s which is lower than shock loading rates (estimated to be over $\dot{\epsilon} = 10^5$ 1/s) this data is still very useful as it can be used to calibrated models where strength and elastic limits are dependent or loading rate, and then use those models

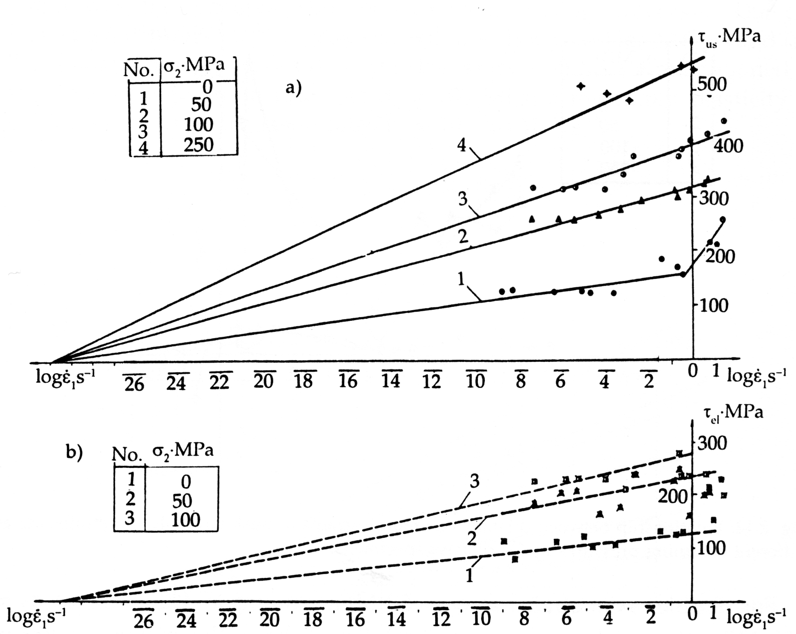


Figure 104.65: Diabase: dependence of peak strength (a) upper) and elastic limit (yield stress) (b) lower) on loading rate ($\dot{\epsilon}$) for different confining stresses ($1 - \sigma_2 = 0$ Mpa; $2 - \sigma_2 = 50$ Mpa; $3 - \sigma_2 = 100$ Mpa; $4 - \sigma_2 = 150$ Mpa) (Stavrogin et al., 2001).

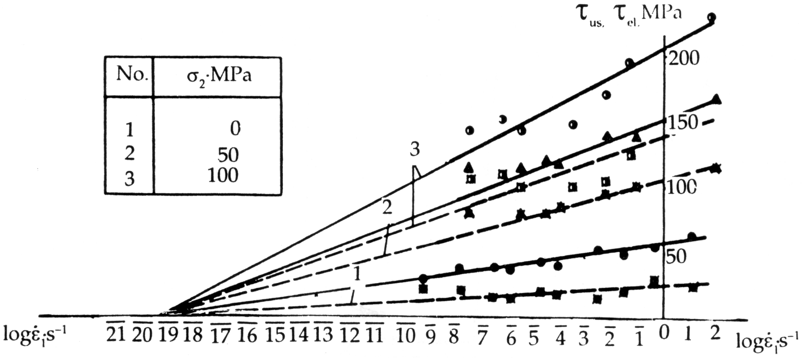


Figure 104.66: Sandstone: dependence of peak strength (full lines) and elastic limit (dashed lines) on loading rate ($\dot{\epsilon}$) for different confining stresses ($1 - \sigma_2 = 0$ Mpa; $2 - \sigma_2 = 50$ Mpa; $3 - \sigma_2 = 100$ Mpa) (Stavrogin et al., 2001).

to predict (in the sense of true prediction (Oberkampf et al., 2002)) shock wave propagation. Data presented in above Figures suggests that both peak strength and elastic limit increase with confinement pressures (pressure sensitive material, as discussed in section 104.8.1.1 on page 331). Presented data also suggest that both peak strength and elastic limit increase (linearly!) with the increase of rate of loading. Dependence of peak strength and elastic limits is proportional to loading rate increase, although such factor of proportionality is not very high. This is move evident from Figures 104.67 and 104.68 where full stress strain curves are shown for marble and granite respectively.

While data shown in Figures 104.67 and 104.68 does not cover such wide range of loading strain rates, it still leads to similar conclusion, that is increase in strain rate will lead to proportional increase in elastic limit and peak strength. Similar conclusion was drawn by Zhao et al. (1999), however, their results show much larger variation and weaker factor of proportionality.

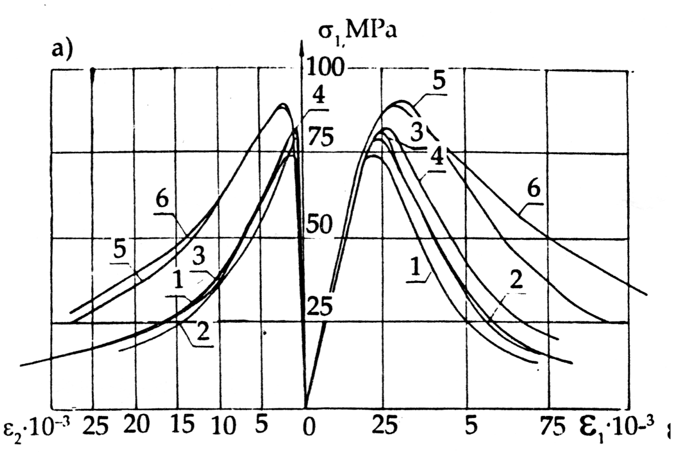


Figure 104.67: Marble: stress-strain response for different strain rates ($\dot{\epsilon}_1$) 1 – $\dot{\epsilon}_1 = 2 \times 10^{-6} 1/s$; 2 – $\dot{\epsilon}_1 = 2 \times 10^{-5} 1/s$; 3 – $\dot{\epsilon}_1 = 2 \times 10^{-4} 1/s$; 4 – $\dot{\epsilon}_1 = 2 \times 10^{-3} 1/s$; 5 – $\dot{\epsilon}_1 = 2 \times 10^{-2} 1/s$; 6 – $\dot{\epsilon}_1 = 2 \times 10^{-1} 1/s$; (Stavrogin et al., 2001).

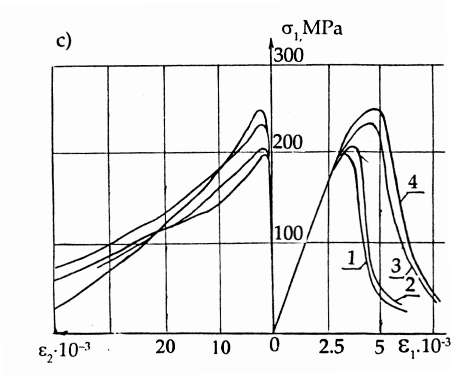


Figure 104.68: Granite: stress-strain response for different strain rates ($\dot{\epsilon}_1$) 1 – $\dot{\epsilon}_1 = 10^{-5} 1/s$; 2 – $\dot{\epsilon}_1 = 2 \times 10^{-4} 1/s$; 3 – $\dot{\epsilon}_1 = 5 \times 10^{-2} 1/s$; 4 – $\dot{\epsilon}_1 = 2 \times 10^{-1} 1/s$; (Stavrogin et al., 2001).

104.8.1.5 Coupling with Pore Fluid Pressure and Temperature

In addition to constitutive response of elastic, plastic and damaged matrix (solid porous skeleton) pore fluid and temperature fields have large influence on behavior of rock. Pore pressure directly influences mechanical response of solid skeleton through effective stress principle (see section 104.8.6). In addition to that, changes in temperature field will affect both pore fluid and elastic-plastic-damage characteristics of the porous solid (rock skeleton).

For example, Figure 104.69 shows dependence of stress at failure for sandstone. In particular, Figure 104.69(a) (left) shows clear dependence of failure stress on pore pressure, where increase in pore fluid pressure reduces the failure limit (stress). This is the case for different confining stresses, and the dependence is linear, which nicely follows equations that will be given in section 104.8.6. If the dependence is plotted in somewhat different form, as a function of effective stress, as shown in Figure 104.69(b) (right) all the points fall into (almost) same line. Present variation is inherent to all geomaterials and is discussed in some detail in section 104.8.2.

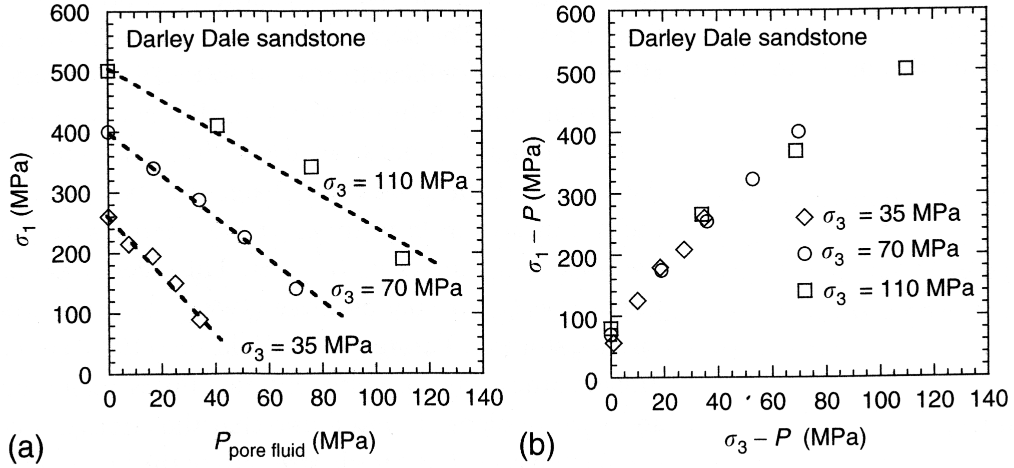


Figure 104.69: Sandstone: ultimate stresses (full capacity mobilized) as a function of (a) left) pore fluid pressure; (b) right) effective principal stress (Jaeger et al., 2007).

In addition to a full saturation, where effective stress principle is fully valid, partial saturation plays a very important role in behavior of porous rock. Partial saturation will affect response of porous rock matrix through increase in pore fluid (mix or water and air) pressures as loading is applied and rock undergoes compactive or dilatant response. Such interaction of pore fluid (mix or air and water) will also be affected by the rate of load application as the fluid viscosity starts having effects on pore pressure advection and diffusion. For example, Figures 104.70 and 104.71 shows effects of moisture content[68] on peak strength. elastic limit and coefficient of cohesion for sandstone and limestone (cohesion only). Full saturation of sandstone samples was achieved at moisture content of 3 % while for limestone it

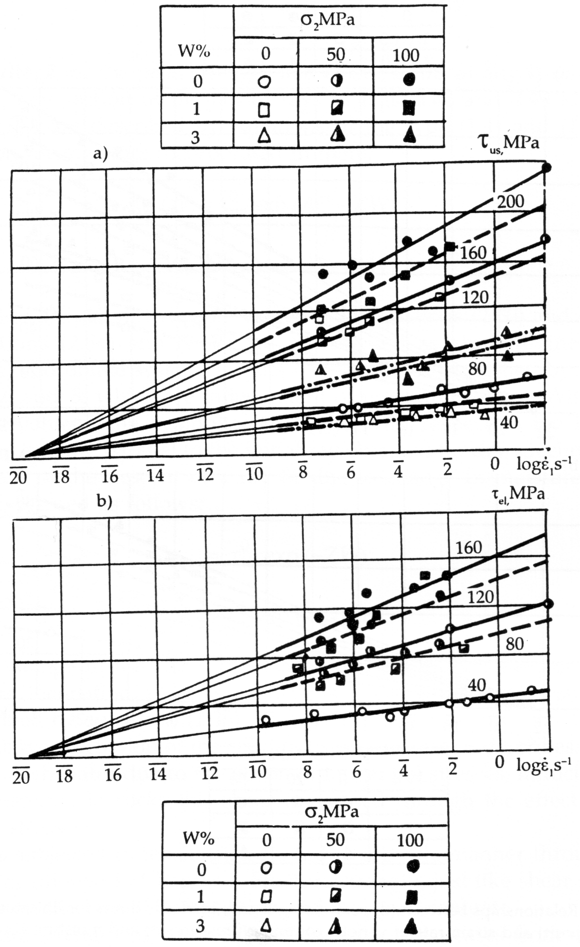


Figure 104.70: Sandstone: dependence of peak strength (a) upper) and elastic limit (yield stress) (b) lower) on loading rate ($\dot{\epsilon}$) for different confining stresses ($1 - \sigma_2 = 0$ Mpa; $2 - \sigma_2 = 50$ Mpa; $3 - \sigma_2 = 100$ Mpa; $4 - \sigma_2 = 150$ Mpa) and moisture content (W) (Stavrogin et al., 2001).

was 8 %. Results show that steady decrease of peak strength, elastic limit and coefficient of cohesion with increase in moisture content. It is very interesting to note that an increase in loading rate still increases the peak strength, elastic limit and coefficient of cohesion (as concluded in section 104.8.1.4 for dry samples), even with increase in influence of pore fluid (water and air) pressures, that now has to advect and diffuse. This type of interaction of pore fluid with loading rate and response of rock skeleton is somewhat counter-intuitive as it was expected that increase in saturation with increase in loading rate

---

[68] Moisture content is here defined as weight percentage of the water content compared to the weight of the sample.

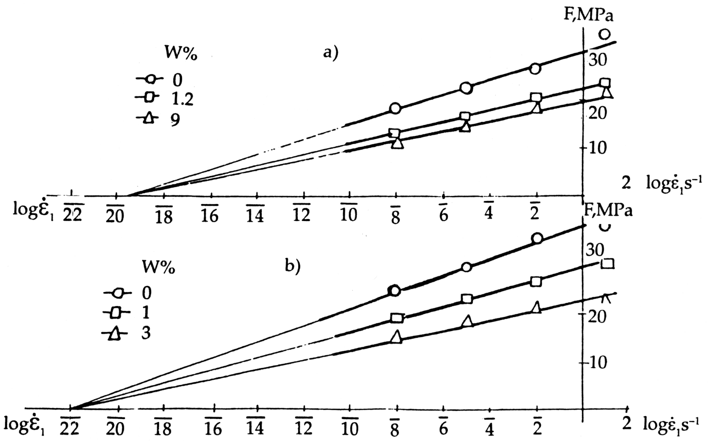


Figure 104.71: Dependence of the coefficient of cohesion for sandstone (a, upper) and limestone (b, lower) on strain rate ($\dot{\epsilon}$) for different levels of moisture content (W) (Stavrogin et al., 2001).

will lead to an increase in pore fluid pressures during high loading rate events, and thus reduce peak strength, elastic limit and coefficient of cohesion.

The influence of pore fluid on response can also be followed on full stress strain curves, as shown in Figure 104.72 for limestone. While increase in water (moisture) content leads to a decrease in stiffness and elastic-plastic strength, increase in strain (loading) rate leads to an increase in stiffness and elastic-plastic strength (stress-strain curves are "higher" for faster loading). This seemingly counter-intuitive response might need to be explored in more depth.

While previously shown results assume that pore space within rock is mostly connected, Curran (1994) uses simple micromechanical models to predict that at high confinement, the behavior of fully water-saturated rock changes from a classical effective stress response (see more about effective stresses in section 104.8.6) to a much stronger and stiffer material. This transition happens at different confining pressures, depending primarily on solids bulk modulus and the pore morphology/fabric. For example, Curran (1994) note that hard rock with small ratio of crack volume to pore volume, the transition begins at 0.2 GPa, while for soft rock (or rock with high ratios of crack volume to pore volume), the transition happens at much higher confinements, of $1-2$ GPa. This change in rock skeleton needs to be taken into account for any modeling and simulations where such threshold compressive values are reached or exceeded. In addition to that, a very important observation was made by (Larson and Anderson, 1979), in that under high confining pressures, liquid water will turn into ice VI [69] thus significantly changing the nature of coupling of pore fluid with porous solid.

Changes in temperature will affect both the pore fluid (by changing the volume and reactivity with

---

[69] Ice VI is a tetragonal crystalline form of ice formed by cooling water to 270 K at 1.1 GPa.

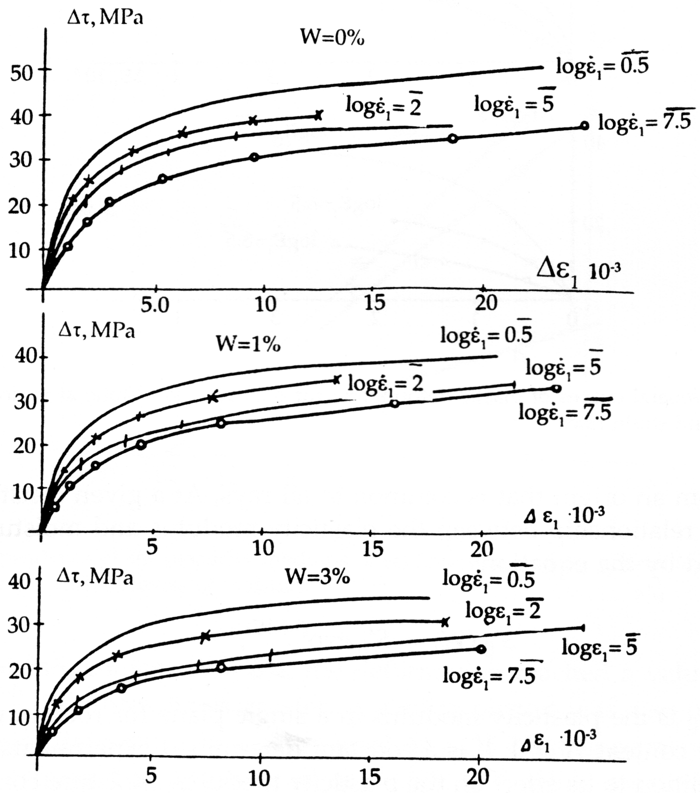
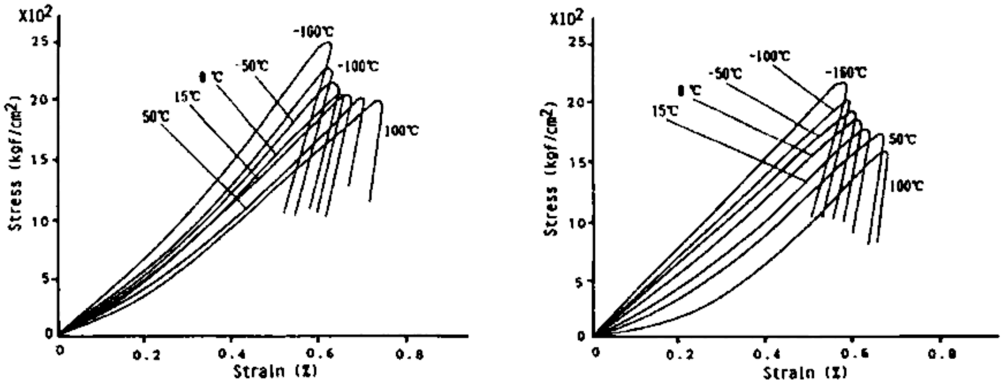


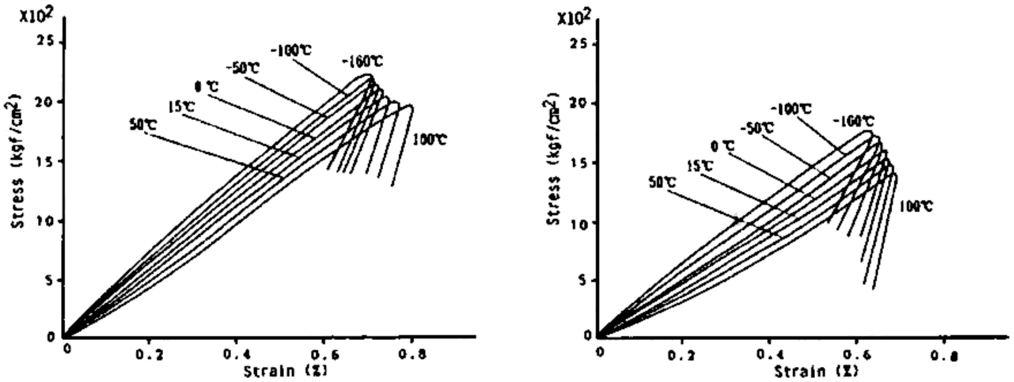
Figure 104.72: Limestone: stress-strain ($\tau - \epsilon_1$) curves for different strain rates ($\dot{\epsilon}$) and different levels of moisture content (W) (Stavrogin et al., 2001).

rock minerals), as well as the behavior of the solid (porous) matrix. With an increase in temperature, rock becomes softer but much more ductile. Inada et al. (1992) performed a number of tests on granite, andesite and sandstone, with changes of temperature from –160^{o} C (–256^{o} F) to +100.00^{o} C (212^{o} F) for both dry and wet (saturated) rock samples. Figure 104.73 shows stress-strain curves resulting from those tests. While the influence of saturation and temperature varies for different types of rock, general trend is that with increased temperature response becomes softer (lower stiffness, lower peak strength) with higher ductility. Temperature range presented does not cover completely application area for shock loading (where temperatures might reach and exceed melting point for rock), however softening and increase in ductility trend will continue as the temperatures increase until close to melting point. At such high temperatures, rock behavior will gradually change from solid to heavy fluid with significant changes in viscosity. Holyoke and Rushmer (2002) did a number of tests on muscovite-biotite metapelite and a biotite gneiss with very high temperatures (from 650^{o} C (1202^{o} F) to 950^{o} C (1742^{o} F)) and reported dilatancy effects and highly ductile response, with strains extending over 15 % and reaching peak strength at 5 – essentially represents a new material that has to be properly modeled.



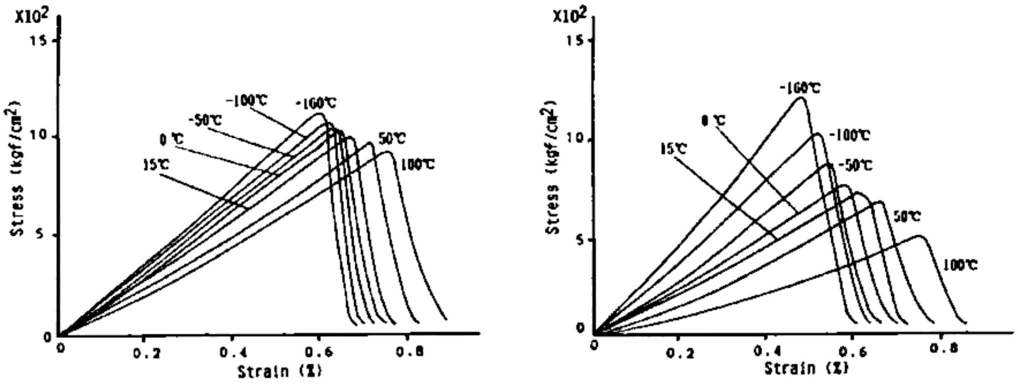
(a) granite
(normal to the rift plane, Dry)

(b) granite
(normal to the rift plane, Wet)



(c) andesite (Dry)

(d) andesite (Wet)



(e) sandstone
(parallel to the layer, Dry)

(f) sandstone
(parallel to the layer, Wet)

Figure 104.73: Stress-strain curves for dry and wet samples of granite, andesite and sandstone with varying temperature (Inada et al., 1992).

104.8.2 Uncertainty and Variability of Rock Behavior

Rock material behavior is characterized by point wise uncertainty and spatial variability. While this topic is covered in much more detail in section 506.6, given here is a brief account of experimental data that supports above statement. Test data shown in previous sections exhibits variation. This variation is present in any set of test data were more than one sample was used to determine material properties. For example, Figure 104.74 from Pariseau (2006), shows values for shear modulus for a variety of rock types. It is noted that the variability is quite large, yet this is nothing unusual and is rather characteristic of rock material. In addition to that, variability of rock material parameters is apparent in any other set of test data. For example larger and/or smaller variability is present in data presented in Figures 104.56, 104.59–104.63, 104.64–104.66, 104.69–104.71. This variability is always present (see discussion in section 506.6) and need to be taken into account in order to have higher confidence in modeling and simulation of rock behavior. One possible approach to fully incorporating uncertainty into modeling and simulation is given in section 506.6 on page 2649.

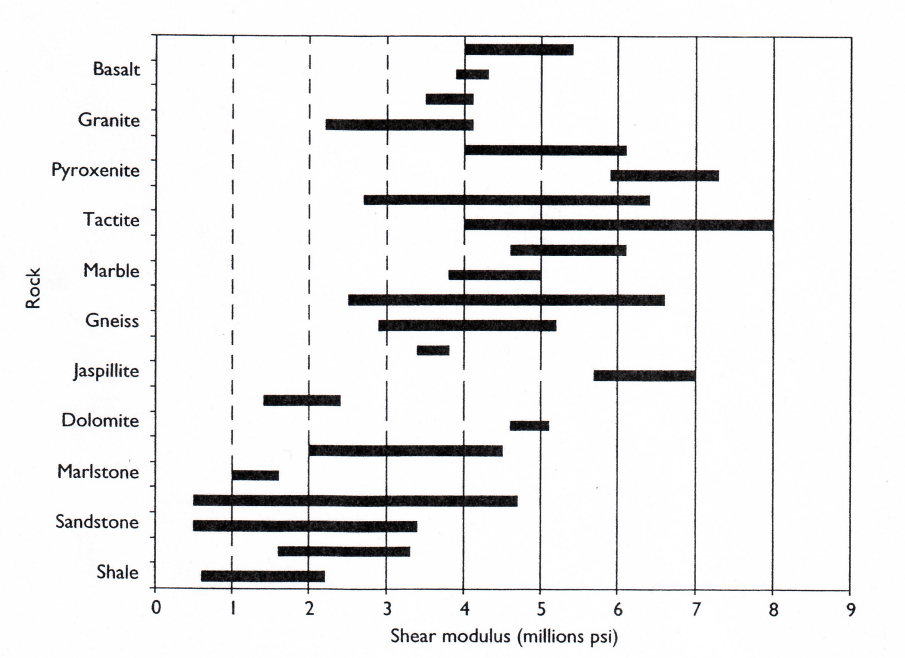


Figure 104.74: Variation of shear modulus for different rock types (Pariseau, 2006).

104.8.3 Effects of Shock Loading on Intact Rock Behavior

In previous sections I have attempted to present main features of intact rock behavior for a variety of loading conditions, loading rates, temperatures... Select set of publications were used to describe specifics of rock behavior, including pressure sensitivity, dilative and compactive response, anisotropy, effects of high rate of loading and effects of pore fluid pressures and temperatures and finally the uncertainty of such data. In this section, select publications on shock loading in rock are reviewed with the main aim of synthesizing previous findings and showing that all of the previous specifics of intact rock behavior influence response to shock loading.

It is very important to note that the history of development of modern theory of shock waves is long and quite interesting. Salas (2006) describes many (unsuccessful) attempts by the greatest mechanics minds of last two centuries to develop a consistent theory of shock waves. Such historic are invaluable in having the right prospective in developing consistent approaches for multi-physics modeling of shock waves in intact rock.

A very interesting set of experimental results and findings for rocks of both high porosity (tuff and limestone) (Larson and Anderson, 1979) and low porosity (granite and sandstone) (Larson and Anderson, 1980) are available in literature. For low porosity tests a number of important observation are made. Relationship of shock wave velocity versus particle velocity shows a discontinuity. It is suggested that this is due to the shear initiated pore collapse, which relates to dilatancy and compactive/shear localized deformation bands. This is indeed more appropriate, in Authors opinion, and in view of recent research on compressive localization (Olsson, 1999, 2001; Issen and Rudnicki, 2001; Borja, 2006) that such pore collapse is due to the initiation of diffusive, implosive and localized, compaction bands. In addition to that Larson and Anderson (1980) show that uniaxial strain data for low strain rates can be used to model high strain rate events, for dry rocks. This is a very important findings as it allows calibration of material models using low rate experiments for prediction of shock wave events. In addition to that, it was found that for very low porosity rock where pore are not in contact, the effective stress principle does indeed need to be used in its original form (Biot and not Terzaghi form) as given in equation 104.616 in section 104.8.6 on page 363, with α having a very low value. This is consistent with findings from micromodels mentioned in section 104.8.1.5 (Curran, 1994). Test result date for highly porous rock (tuff and limestone) (Larson and Anderson, 1979) show that porosity plays a major role in behavior of rock. For example, the irreversible process of compaction and elastic unloading leads to a large hysteresis, dissipating significant amounts of wave energy, effectively damping the wave out. In addition to that, an increase of strength with strain rate is observed, reinforcing experimental observations be Stavrogin and Protosenya (1983); Stavrogin and Pevzner (1983); Stavrogin et al. (2001), described in section 104.8.1.4

on page 344.

Dynamic compaction behavior of intact rock material was and still is of considerable interest. Lysne (1970) did a number of experiments on dry and water saturated tuff. The main conclusion he was able to draw was that the compaction of porous rock material is a process that is slower than the shock wave propagation and that it takes longer time (than for the compressive wave to pass) to complete such compaction process. In addition to that, Lysne (1970) was able to show, that at least for stresses in water below 2.5 GPa, the influence of heating on volume change can be neglected in porous rock saturated with water. D. Erskine and Weir (1994) presents data on dry and wet tuff, which exhibits quite complex behavior. Both pore crushing and phase change (liquefaction) are observed and are responsible for complex compression behavior. Heterogeneity of rock also plays a major role in the observed response. Another very important conclusion is that Gruneisen model (aka Debye-Grüneisen model) does not perform well, indicating that there a likely phase transition is happening and needs to be accounted for.

Hiltl et al. (1999) present interesting set of shock-recovery tests results on dry and fully saturated sandstone. Principle of effective stress is again playing important role as it is observed that water saturated samples had much smaller compaction due to distribution of confining pressures between pore water and porous solid. In addition to that, reduction of porosity due to high pressure of shock waves was much smaller for saturated samples, again proving that pore fluid caries quite a bit of load due to slow drainage, as present during shock wave loading. It is also observed that as the shock pulse duration increases, so does the damage, implying that as the pore fluid gains time to drain, effective stresses in porous solid increase and causes the damage and compaction.

104.8.3.1 Shock Waves of First and Second Kind

The dynamic behavior of saturated porous media was studied at length by M.A.Biot (1956); Biot (1962, 1972). In one of his studies (M.A.Biot, 1956) he concluded that there exist three kinds of coupling between pore fluid and porous solid (inertial, viscous and mechanical). He also concluded that the viscous coupling plays a key role and determines response of a coupled system to dynamic excitation, while making wave propagation dispersive. He demonstrated (analytically) the existence of two kinds of compressional waves corresponding to the mechanical and inertial coupling, while the viscous coupling is responsible for a pure wave. For high viscous coupling, the relative movements between the two phases are prevented, so there is only one compressional wave and the total mass behaves as a single-phase medium.

Recent paper by Lomov et al. (2001), actually shows an experimental proof of such coupling. Figure 104.8.3.1 shows two test results for dry and wet samples. It is important to recognize that these test results can also be used to validate shock wave propagation modeling and simulation as described in section 104.8.6 on page ??.

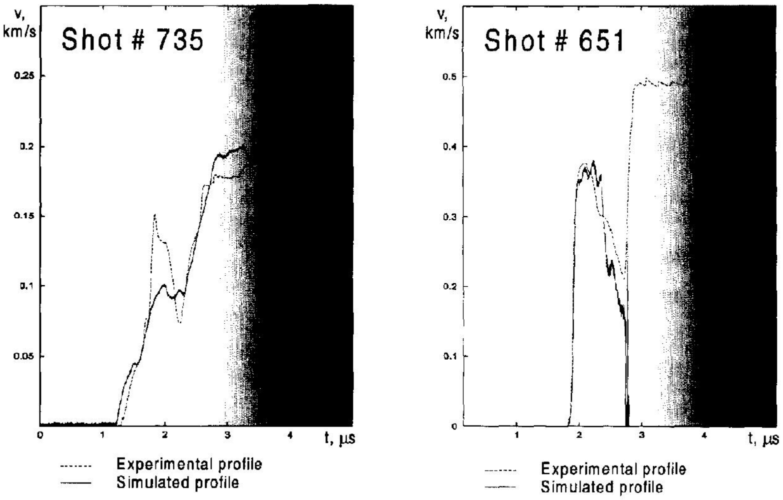


Figure 104.75: Test results for a 1D wave propagation in dry (high coupling) and wet (low coupling) sandstone (Lomov et al., 2001).

For example, for high coupling case (dry sandstone, pore fluid is air), the velocity profile shown in Figure 104.8.3.1 is increasing and when integrated (to get displacements), profile corresponds (at least qualitatively) to high coupling case ($K = 10^{-6}$ cm/s) presented in Figure 310.14. Similarly, low coupling case (wet sandstone, pore fluid is water) the velocity profile increases, then drops to almost zero and then increases again. Upon integration to get displacement, this will correspond (qualitatively) to low coupling case ($K = 10^{-2}$ cm/s) presented in Figure 310.14. Permeabilities are here used qualitatively for both water and air and signify ease with which fluid (water or air) moves past porous solid. While, results presented in Figure 310.14 correspond to a linear elastic case, at least qualitatively, they follow test results, which of course correspond to elastic–plastic behavior of rock material. Such elastic-plastic behavior will affect results in many ways (slow the wave propagation speed for one), but at least it is reassuring that an elastic solution can be used to help gain understanding of the basic mechanics.

Analytical solutions for shock wave propagation, even with many simplifying assumptions, are valuable as candidates for verification and validation. Two such solutions are mentioned below. Vasilev et al. (1980) discusses interaction of gas, liquid and porous medium during and after an underground explosion. A complex interaction is described which eventually leads to the implosion of explosion cavity. Although the analysis presented is based on elastic behavior of the porous medium, it provides excellent basis for understanding phenomena involved as well as for verification. Nikolaevskiy et al. (2006) presents more sophisticated analysis of similar phenomena, using finite difference method (Wilkins, 1999), and an elastic-plastic material model. However, Authors neglect the dilatancy of the material in order to simplify their solutions. Neglecting dilatancy certainly affects results Compaction related to coupling of pore fluid and porous solid. Moreover, influence of temperature fields is also neglected (isothermal process) which might hinder fidelity of modeling where large temperature changes are present and where temperature change influences behavior of pore fluid and porous solid.

104.8.3.2 Hugoniots

The Hugoniot curves (also known as Rankine-Hugoniot) for material present important data about material state. Material compression state defined by initial pressure, density and energy, can be used, to determine new state upon applying shock loads. Such curves prove important in material modeling for rock subjected to shock loading. Early on Afanasenkov et al. (1969) showed that a it is possible to predict shock Hugoniots of any substance up to compression ratio of two with the knowledge of initial density and initial compressibility. Shipman et al. (1971) used a number of experiments to determine Hugoniots for Sandstone. In addition to that they used measured data and developed curves to show that phase boundaries do shift significantly compared to those determined using static means. This important conclusion affects development of elastic-plastic modeling for intact rock where high temperature effects cannot be neglected.

104.8.4 Material Modeling of Rock

104.8.4.1 Lawrence Livermore National Laboratory Models

A number of models originating from researchers from Lawrence Livermore National Laboratory have been developed over the years. While they do not represent a single line of development (and might have been produced by different research groups from different departments) they are summarized in this LLNL section.

Glenn (1995) presents a simple, yet effective total stress model that depends on mean confinement (pressure), temperature and on a damage parameter that serves as an internal variable and depends on degradation due to tensile and shear failure.

$$\sqrt{3 s_{ij} s_{ij}} \leq Y = (1-D)\bar{Y} + \beta D \bar{Y} \tag{104.614}$$

where s_{ij} is the deviatoric stress, β is a constant, and D is a scalar function of the volumetric components of void and equivalent plastic strain tensors. The generalized compressive strength $\bar{Y}$ is a function of unconfined compressive strength, the ultimate compressive strength, the melting temperature, the mean pressure, the cohesion and an material additional constant.

More recently Lomov et al. (2001); Antoun et al. (2003), presented an elastic viscoplastic material model that takes into the account various influences on rock yielding behavior. For example, taken into the account are the effects of scaling, hardening, damage and melting. In addition to that, compaction is modeled using analytic porous compaction model, while also included are the effects of dilatancy. Model is set in a proper thermodynamic framework. However, it should be noted that strictly following thermodynamics for geomaterial behavior, can have negative effects on modeling proper volumetric response (dilatancy and compaction), with alternative material model formulation spaces being suggested by Collins and Houlsby (1997). This is an area which certainly deserves much attention, namely the apparent small disconnect between sound thermodynamic framework for modeling (which nicely applies to metal plasticity) and observed behavior of geomaterials (rock included).

104.8.4.2 Hoek and Brown Model

One of the most often used material models for rock is Hoek and Brown. It is important to note that this model is actually a failure criterion, delineating elastic and failure states of stress, lacking usual plasticity features, such as hardening and/or softening. The most recent edition of the model (Hoek et al., 2002) fixes some earlier observed problems with friction angle determination. This is a valuable model for practical work for rock with low confinement stresses, where behavior is brittle and failure indeed occurs as soon as the failure state of stress is reached. However, in view of shock loading modeling, this model does not hold much promise, as it lacks, as mentioned above, basic elastic-plastic features.

104.8.4.3 Other Models

A number of other material models have been developed for modeling of elastic-plastic-damage behavior of rock. Small selection is presented below. It should be noted that most of those developed models inherit most of their features from models described in previous sections.

Benz and Schwabb (2008) provide comparison for six most commonly used failure criteria for rock. While failure is emphasized, as opposed to full elastic-plastic behavior, the data presented is very telling in view of uncertainty of rock behavior. For example, six deterministic models are calibrated using statistical fitting techniques, and deterministic parameters are developed for a deterministic elastic-plastic (failure) models. The information about the uncertainty of response is thus completely lost.

Das and Basudhar (2009) perform similar exercise with four deterministic models. Moreover, they label some of the test data as outliers thus negatively influencing date regression (removal of statistical moments).

G. W. Ma and Zhou (1998) present an isotropic elastic-plastic model that includes rate dependence, damage development and plasticity of rock. Isotropic damage was used where the elastic constitutive tensor was related to scalar damage parameter D

$$E_{ijkl} = E_{ijkl}(1 - D) \tag{104.615}$$

This is a standard way of incorporating scalar damage (Carol et al., 1995). Of course, anisotropic damage (Rizzi, 1993; Carol et al., 2001a,b; Loret and Rizzi, 1997) is more accurate in modeling realistic materials and should be used whenever possible, however in this case, scalar damage was identified as sufficiently accurate for modeling. In this model, yield strength is controlled by accumulated damage, through a simple linear, isotropic relationship. In addition to that, a non-associated plastic flow is employed. However, plastic flow is limited to deviatoric plane (there is no volumetric component) which reduces accuracy of modeling, since rock material does undergo plastic volumetric change upon plastification (Borja, 2006). Comparison of simulation and experimental data shows somewhat satisfactory similarity, however, bias is present in attenuation plots (similar to results obtained by Wei et al. (2009)).

Bart et al. (2000) presents an interesting approach where an anisotropic poroelastic damage model is used to model slow behavior of rock samples (sandstone). Model is able to predict effects of damage induced by micro-cracking, such as deterioration of elastic and poroelastic properties, induced anisotropy and dilatancy. Model does not feature any plastic deformation, rather inelasticity is completely managed through damage.

Chen et al. (2010) develop an elastic-plastic-damage model which can handle inherent and induced anisotropy. Calibration and application to shale is presented, with main focus on behavior of strongly anisotropic samples. Fabric tensor (not unlike fabric tensor developed for SANISAND family of models described in section 104.6.12) presents an effective modeling tool for modeling anisotropy. However, in their model, Chen and Phoon (2009) simplified modeling of induced anisotropy to isotropic damage (citing complexity of doing it otherwise). This might be unfortunate as induced anisotropy (resulting from anisotropic damage) might be more important than inherent anisotropic, particularly for cases where reversal of loading plays an important role, for example in modeling of shock wave propagation.

104.8.5 Model Calibration / Testing Devices

While a number of material models have been developed over years to model rock behavior, calibration of such models has to be done with great care. Rock is fairly stiff material and as such, stiffness of testing equipment can have significant effects on test results. For example, Labuz and Biolzi (2007) discusses such influence in great detail. Figure 104.8.5 shows how inappropriate stiffness of testing equipment can affect (mask) the real rock response.

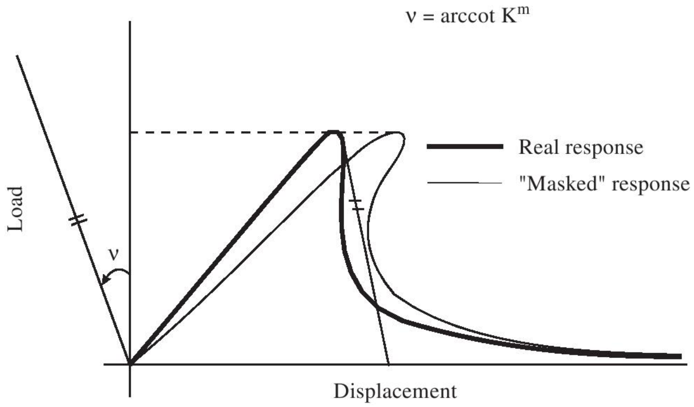


Figure 104.76: Influence of testing machine stiffness on observed and real test specimen behavior (Labuz and Biolzi, 2007).

In addition to that, elastic-plastic models assume intact rock, so that any influence of discontinuities is removed from test results. This is where scaling of samples plays a very important role. For example Lo et al. (1987) show how variation in test specimen volume (see Figure 104.8.5) can affects (bias) measurements of dynamic elastic modulus, by simply including, within the tested volume, discontinuities and not accounting for them.

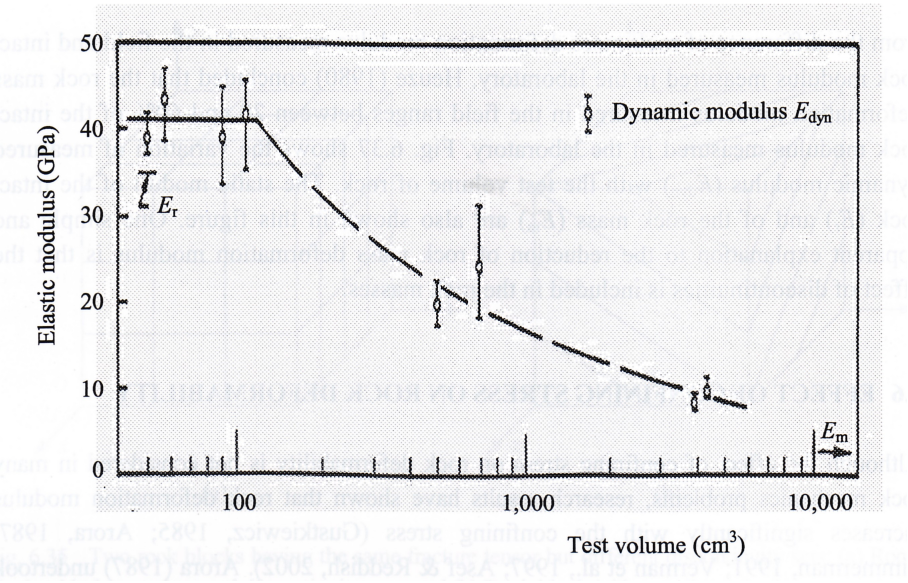


Figure 104.77: Influence of volume of test specimen elastic modulus of rock ((Lo et al., 1987)).

104.8.6 Influence of Pore Fluid Pressure and Temperature on Rock Response

One of the main features of geomaterials is the coupling of pore fluid with the porous solid. Such coupling is taken into account through the effective stress principle. The relationship between effective stress, total stress and pore pressure is (assuming tensile components of stress as positive and compressive pressure, p is positive) (Zienkiewicz et al., 1999a)

$$\sigma_{ij}^{''} = \sigma_{ij} + \alpha \delta_{ij} p \tag{104.616}$$

where $\sigma_{ij}^{''}$ is effective stress tensor, σ_{ij} is total stress tensor, δ_{ij} is Kronecker delta. $\delta_{ij} = 1$, when i=j, and $\delta_{ij} = 0$, when $i \neq j$. For isotropic materials,$\alpha = 1 - K_T/K_S$ (Bouteca and Gueguen, 1999), and K_T is the total bulk modulus of the solid matrix, K_S is the bulk modulus of the solid particle/grains. For most of the geomechanics problems, as the bulk modulus K_S of the solid particles is much larger that that of the whole material, $\alpha \approx 1$ can be assumed. However, in case of rock (as well as for concrete, bone material...), such assumption does not hold all the time so α needs to be kept throughout derivations (Bouteca and Gueguen, 1999).

104.9 Inelastic Behavior and Models for Concrete Beams, Walls and Shells

104.9.1 Uniaxial Material Model for Steel

The uniaxial steel material model used in this study was developed by Menegotto and Pinto (1973) and extended by Filippou et al. (1983). Model is capable of capturing the nonlinear hysteretic behavior and isotropic strain-hardening effect of steel for uniaxial state of stress and strain (1D). The stress–strain response of rebar steel material is shown in Figure 104.93. The model, as presented in Menegotto and Pinto (1973), takes on the form:

$$\sigma^* = b\epsilon^* + \frac{(1-b)\epsilon^*}{(1+\epsilon^{*R})^{1/R}} \tag{104.617}$$

with

$$\epsilon^* = \frac{\epsilon - \epsilon_r}{\epsilon_0 - \epsilon_r}; \quad \sigma^* = \frac{\sigma - \sigma_r}{\sigma_0 - \sigma_r} \tag{104.618}$$

where b is the strain-hardening ratio, ϵ_r and σ_r are the strain and stress at the point of strain reversal, ϵ_0 and σ_0 are the strain and stress at the point of intersection of the two asymptotes, R is the curvature parameter that governs the shape of the transition curve between the two asymptotes. It is noted that this model is for uniaxial material behavior, in which the stresses and strains are scalars instead of tensors.

The expression for the curvature parameter R is suggested by Menegotto and Pinto (1973) as:

$$R = R_0 - \frac{c_{R_1}\xi}{c_{R_2} + \xi} \tag{104.619}$$

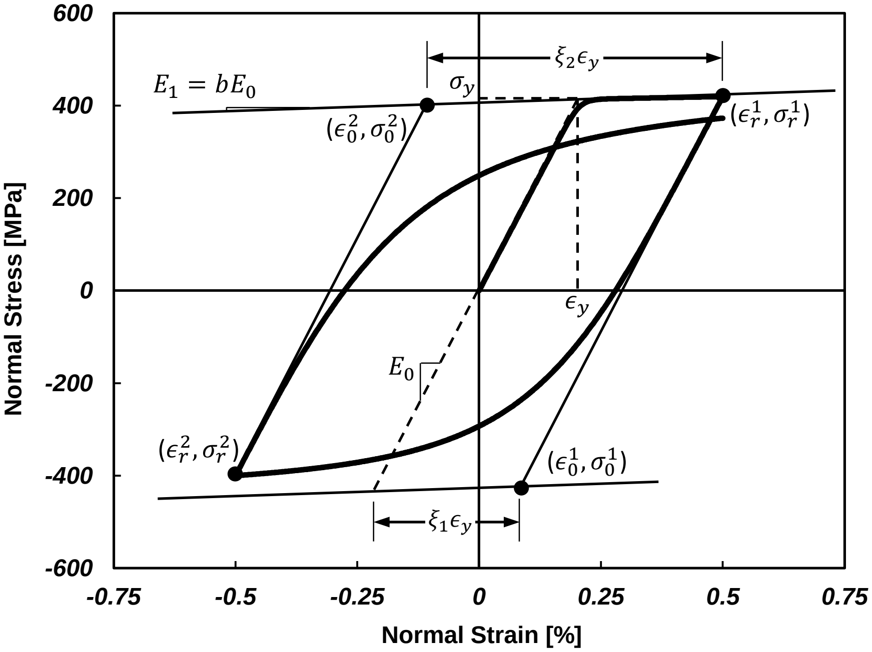


Figure 104.78: Constitutive model for uniaxial steel fiber (Menegotto and Pinto (1973)).

where R_0 is the value of the curvature parameter R during initial loading, c_{R_1} and c_{R_2} are degradation parameters that need to be experimentally determined. The parameter ξ, which is updated after strain reversal, is defined as:

$$\xi = \left| \frac{(\epsilon_m - \epsilon_0)}{\epsilon_y} \right| \tag{104.620}$$

where ϵ_m is the maximum (or minimum) strain at the previous strain reversal point, depending on the loading direction of the material. If the current incremental strain is positive, the parameter ϵ_m takes the value of the maximum reversal strain. Parameter ϵ_y is the monotonic yield strain.

In order to capture isotropic hardening behavior, Filippou et al. (1983) introduced stress shift mechanism into the original model by Menegotto and Pinto (1973). Note that the hardening rate in compression and tension can be different by choosing different hardening parameters for compression and tension. The proposed relation takes the form:

$$\frac{\sigma_{st}}{\sigma_y} = a_1 \left(\frac{\epsilon_{max}}{\epsilon_y} - a_2 \right) \tag{104.621}$$

where σ_{st} is the shift stress that determines the shift of yield asymptote, ϵ_{max} is the absolute maximum strain at strain reversal, and a_1 and a_2 are hardening parameters in compression which are experimentally determined. In the case of tension, the hardening parameters a_1 and a_2 in Equation 104.621 are changed to a_3 and a_4, that are also determined experimentally or from previous studies for given steel.

104.9.2 3D Plastic Damage Concrete Material Model, Faria-Oliver-Cervera

The concrete material model used in this study was developed by Faria et al. (1998). Model features:

- distinct stress-strain envelopes obtained under compression or under tension
- stiffness recovery after loading reversal
- higher concrete strength under 2D or 3D compression test, compared to 1D loading
- plastic deformations discernible after some compressive stress limit is reached

The material model, as presented in Faria et al. (1998), takes on the form:

$$\sigma_{ij} = (1-d^+)\bar{\sigma}^+_{ij} + (1-d^-)\bar{\sigma}^-_{ij} \tag{104.622}$$

where, d^+ and d^- are scalar damage variables corresponding to tensile and compressive degradation. Cauchy stress tensor σ_{ij} involves effective stress components $\bar{\sigma}^+_{ij}$ and $\bar{\sigma}^-_{ij}$, that are related to the total effective stress ($\bar{\sigma}_{ij} = \bar{\sigma}^+_{ij} + \bar{\sigma}^-_{ij}$), defined as follows:

$$\bar{\sigma}_{ij} = D_{ijkl}(\varepsilon_{kl} - \varepsilon^p_{kl}) \tag{104.623}$$

In the previous equation, D_{ijkl} is the fourth order isotropic linear elastic constitutive tensor, ε_{kl} the small strain tensor and ε^p_{kl} is the plastic strain tensor. Damage variables together with the plastic strain constitute the internal variable set. Tensile part of the effective stress tensor can be written using principal stresses ($\bar{\sigma}_i$) and principal directions (p_i):

$$\bar{\boldsymbol{\sigma}}^+ = \sum_i \langle \bar{\sigma}_i \rangle p_i \otimes p_i \tag{104.624}$$

Compressive components of the effective stress can be written as:

$$\bar{\sigma}^-_{ij} = \bar{\sigma}_{ij} - \bar{\sigma}^+_{ij} \tag{104.625}$$

Following adopted stress split, a tensile equivalent stress $\bar{\tau}^+$ and a compressive equivalent stress $\bar{\tau}^-$ are considered. According to Simo and Ju (1987):

$$\bar{\tau}^+ = (\bar{\sigma}^+_{ij} D_{ijkl} \bar{\sigma}^+_{kl})^{1/2} \tag{104.626}$$

$$\bar{\tau}^- = (\sqrt{3}(K\bar{\sigma}^-_{oct} + \bar{\tau}^-_{oct}))^{1/2} \tag{104.627}$$

In the last equation, $\bar{\sigma}^-_{oct}$ and $\bar{\tau}^-_{oct}$ are the octahedral normal and shear stress, respectively, obtained from $\bar{\sigma}^-$. K is a material characteristic, adjusted so that 2D and 1D compressive strength ratio can match ratio of 1.16-1.2 (Kupfer et al., 1969). Two separate damage criteria, functions, g^+ for tension and g^- for compression, are introduced:

$$g^+(\bar{\tau}^+, r^+) = \tau^+ - r^+ \leq 0 \tag{104.628}$$

$$g^-(\bar{\tau}^-, r^-) = \tau^- - r^- \leq 0 \tag{104.629}$$

Variables r^+ and r^- represent current damage thresholds. Their role is to control the size of expanding damage surfaces. Quadrant ($\bar{\sigma}_2 = 0, \bar{\sigma}_1, \bar{\sigma}_3 \geq 0$) shows 2D representation for this surface, when $\bar{\tau}^+ = r^+$, Fig 104.79. The bounding surface associated to the principal effective compressive stresses resembles Drucker-Prager cone. It is obvious that the elastic domain under 2D compression is bounded by stresses greater than the 1D elastic compressive stress, denoted by f_0^-.

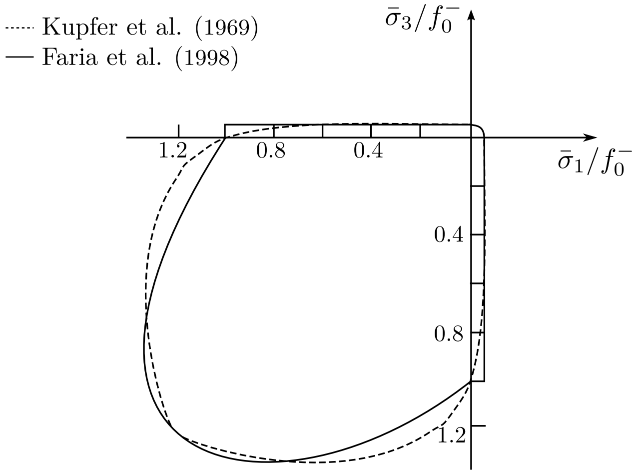


Figure 104.79: Initial 2D elastic domain

For the plastic flow of the tensor ε^p_{ij} the following is proposed in Faria et al. (1998):

$$\dot{\varepsilon}^p_{ij} = \beta\, E\, H(\dot{d}^-) \frac{\langle \bar{\sigma}_{ij} \dot{\varepsilon}_{ij} \rangle}{\bar{\sigma}_{ij} \bar{\sigma}_{ij}}\, D^{-1}_{ijkl}\, \bar{\sigma}_{kl} \tag{104.630}$$

where β is the coefficient which controls the rate of intensity of plastic deformation, E is Young's modulus, $H(\dot{d}^-)$ denotes the Heaviside step function for the compressive damage rate. Proposed model assumes that plastic strain has the direction of the elastic strain tensor $D^{-1}_{ijkl} \bar{\sigma}_{kl}$. It should be noted that the model cannot account for plastic strains for a pure tension test.

Kinematics of damage, internal variables is defined based on the following rate equations:

$$\dot{d}^+ = \dot{\theta}^+ \frac{\partial G^+(r^+)}{\partial r^+} \tag{104.631}$$

$$\dot{r}^+ = \dot{\theta}^+ (\geq 0) \tag{104.632}$$

where G^+ and G^- are monotonically increasing functions for tension and compression, that are experimentally determined, and $\dot{\theta}^+$ and $\dot{\theta}^-$ are damage consistency parameters. Karush-Kuhn-Tucker conditions implies the following (Jeremić et al., 1989-2025):

$$g^+ \leq 0 \quad \text{damage/plastic function} \tag{104.633}$$

$$\dot{\theta}^+ \geq 0 \quad \text{consistency parameter} \tag{104.634}$$

$$g^+ \, \dot{\theta}^+ = 0 \tag{104.635}$$

which leads to the following conclusions:

- when damage/plasticity function is smaller then zero, $g^+ < 0$, consistency parameter must be zero, $\dot{\theta}^+ = 0$, implying that no damage occurs,
- when consistency parameter is greater than zero, $\dot{\theta}^+ > 0$, damage/plastic function is zero, $g^+ = 0$ implying the presence of damage.

It is now possible to define consistency parameter $\dot{\theta}^+$ from the damage consistency condition:

$$\dot{g}^+(\bar{\tau}^+, r^+) = 0 \implies \dot{r}^+ = \dot{\tau}^+ = \dot{\theta}^+, \qquad \dot{\theta}^+ \geq 0 \tag{104.636}$$

Introducing equation (104.636) into (104.631) the flow rule for the tensile damage variable can be expressed as:

$$\dot{d}^+ = \frac{\partial G^+(r^+)}{\partial r^+} \dot{r}^+ = \dot{G}^+(r^+) \geq 0 \tag{104.637}$$

The compression damage variable is then:

$$\dot{d}^- = \frac{\partial G^-(r^-)}{\partial r^-} \dot{r}^- = \dot{G}^-(r^-) \geq 0 \tag{104.638}$$

104.10 Calibration of Elastic-Plastic Material Models

104.10.1 Calibration of Elastic-Plastic Material Models, Soil

104.10.2 Calibration of Elastic-Plastic Material Models, Rock

104.10.3 Calibration of Elastic-Plastic Material Models, Contact/Joint/Interface

104.10.4 Calibration of Elastic-Plastic Material Models, Concrete

104.10.5 Calibration of Elastic-Plastic Material Models, Steel

104.11 Energy Dissipation Calculations for Solids

This section is based on Yang et al. (2018)

104.11.1 Introduction

Energy dissipation in elastic plastic solids and structures is the result of an irreversible dissipative process in which energy is transformed from one form to another and entropy is produced. The transformation and dissipation of energy is related to permanent deformation and damage within an elastic-plastic material. Of particular interest here is the dissipation of mechanical energy that is input into elastic-plastic solids by static or dynamic excitations.

Early work on plastic dissipation was done by Farren and Taylor (1925) and Taylor and Quinney (1934). They performed experiments on metals and proved that a large part, but not all, of the input mechanical energy is converted into heat. The remaining part of the non-recoverable plastic work is known as the stored energy of cold work. The ratio of plastic work converted into heating (Quinney–Taylor coefficient), usually denoted as β, has been used in most later work on this topic. Based on large amount of experimental data, this ratio was determined to be a constant between 0.6 to 1.0 (Clifton et al., 1984; Belytschko et al., 1991; Zhou et al., 1996; Dolinski et al., 2010; Ren and Li, 2010; Osovski et al., 2013).

More recently Rittel (Rittel, 2000; Rittel and Rabin, 2000; Rittel et al., 2003) published several insightful papers on the energy dissipation (heat generation) of polymers during cyclic loading, presenting both experimental and theoretical works. Rosakis et al. Rosakis et al. (2000) presented a constitutive model for metals based on thermoplasticity that is able to calculate the evolution of energy dissipation. Follow up papers (Hodowany et al., 2000; Ravichandran et al., 2002) present assumptions to simplify the problem. One direct application of plastic dissipation to geotechnical engineering is presented by Veveakis et al. (Veveakis et al., 2007, 2012), using thermoporomechanics to model the heating and pore pressure increase in large landslides, like the 1963 Vajont slide in Italy.

In the past few decades, extensive studies have been conducted on energy dissipation in structures and foundations. Work by Uang and Bertero (1990) has been considered a source and a reference for many recent publications dealing with energy as a measure of structural demand. Uang and Bertero (1990) developed an energy analysis methodology based on absolute input energy (or energy demand). Numerical analysis results were compared with experiments on a multi-story building. In work by Uang and Bertero (1990) , hysteretic energy is calculated indirectly by taking the difference of absorbed energy and elastic strain energy. The term absorbed energy of each time step is simply defined as restoring force times incremental displacement. It is also stated that hysteretic energy is irrecoverable, which indicates that this parameter was considered the same as hysteretic dissipation or plastic dissipation. An equation for energy balance, is given by (Uang and Bertero (1990)) as:

$$E_i = E_k + E_\xi + E_a = E_k + E_\xi + E_s + E_h \tag{104.639}$$

where E_i is the (absolute) input energy, E_k is the (absolute) kinetic energy, E_ξ is the viscous damping energy, E_a is the absorbed energy, which is composed of elastic strain energy E_s and hysteretic energy E_h.

The problem with this approach is the absence of plastic free energy, which is necessary to correctly evaluate energy dissipation of elastic-plastic materials and to uphold the second law of thermodynamics. While there is no direct plot of plastic dissipation (hysteretic energy) in Uang and Bertero (1990), since it was not defined directly, there are plots of other energy components. Plastic dissipation can be easily calculated from these plots. After doing this, indications of negative incremental energy dissipation, which violates the basic principles of thermodynamics, were found in various sections of the paper.

This misconception could be clarified by renaming hysteretic energy as plastic work, a sum of plastic dissipation and plastic free energy. Both plastic work and plastic free energy can be incrementally negative, but plastic dissipation (defined as the difference of plastic work and plastic free energy) must be incrementally non-negative during any time period. Unfortunately, this misconception has been inherited (if not magnified) by many following studies on energy analysis of earthquake soils and structures (hundreds of papers).

Even Chopra (2000) used similar set of equations in section 7.9, and equation 7.9.6 is clearly wrong!

The basic principles of thermodynamics are frequently used to derive new constitutive models, for example by Dafalias and Popov (1975), Ziegler and Wehrli (1987), Collins and Houlsby (1997), Houlsby and Puzrin (2000), Collins (2002), Collins and Kelly (2002), Collins (2003) and Feigenbaum and Dafalias (2007). The concept of plastic free energy is introduced to enforce the second law of thermodynamics for developed constitutive models. It is important to distinguish between energy dissipation due to plasticity and plastic work, which is often a source of a confusion. Plastic work is the combination of plastic free energy and plastic energy dissipation, which is defined as the amount of heat (and other forms of energy) transformed from mechanical energy during an irreversible dissipative process. The physical nature of plastic free energy is illustrated later in this section through a conceptual example that is analyzed on particle scale. Essentially, development of plastic free energy is caused by particle rearrangement in granular assembly under external loading.

Specific formulation of free energy depends on whether the elastic and plastic behavior of the material is coupled. According to Collins et al. Collins and Houlsby (1997), Collins (2002), Collins (2003), material coupling behavior can be divided into modulus coupling, where the instantaneous elastic stiffness (or compliance) moduli depend on the plastic strain, and dissipative coupling, where the rate of dissipation function depends not only on the plastic strains and their rates of change but also on the stresses (or equivalently the elastic strains). The modulus coupling describes the degradation of stiffness as in for rock and concrete, and is usually modeled by employing a coupled elastic-plastic constitutive model or by introducing damage variables. The dissipative coupling is considered to be one of the main reasons for non-associative behavior in geomaterials Collins and Houlsby (1997), Ziegler (1981).

A number of stability postulates are commonly used to prevent violation of principles of thermodynamics. Stability postulates include Drucker's stability condition Drucker (1956), Drucker (1957), Hill's stability condition Bishop and Hill (1951), Hill (1958), and Il'Iushin's stability postulate Il'Iushin (1961), Lubliner (1990). As summarized in a paper by Lade Lade (2002), theoretical considerations by Nemat–Nasser (1983) and Runesson and Mrǒz (1989) have suggested that they are sufficient but not necessary conditions for stability. These stability postulates can indeed ensure the admissibility of the constitutive models by assuming certain restrictions on incremental plastic work. As demonstrated by Collins (2002), if the plastic strain rate is replaced by the irreversible stain rate in Drucker's postulate, then all the standard interpretations of the classical theory still apply for coupled materials. Dafalias (1977) also modified Il'Iushin's postulate in a similar way and applied it to both coupled and uncoupled materials.

It is important to note that development of inelastic deformation in geomaterials involves large changes in entropy, and significant energy dissipation. It is thus useful to perform energy dissipation (balance) analysis for all models with inelastic deformation. In section we focus on energy dissipation on material level. Focus is on proper modeling that follows thermodynamics. Comparison is made between accumulated plastic dissipation and accumulated plastic work, since these quantities can be quite different in most cases. As a way of verification, the input work, which is introduced by applying external forces, is compared with the stored energy and dissipation in the entire system. Finally, conclusions on plastic energy dissipation are drawn from the verified results.

104.11.2 Theoretical and Computational Formulations

104.11.2.1 Thermo-Mechanical Theory

For energy analysis of elastic-plastic materials undergoing isothermal process, it is beneficial to start from the statement of the first and second laws of thermodynamics:

$$\hat{W} = \dot{\Psi} + \Phi \tag{104.640}$$

where $\Phi \geq 0$ and $\hat{W} \equiv \sigma : \dot{\epsilon} = \sigma_{ij}\dot{\epsilon}_{ij}$ is the rate of work per unit volume. The function Ψ is the Helmholtz free energy, and Φ is the rate of dissipation; both defined per volume. The free energy Ψ is a function of the state variables (also known as internal variables), but Φ and $\hat{W}$ are not the time derivatives of the state functions. The choice of state variables depends on the complexity of constitutive model that is being used, as cyclic loading with certain hardening behaviors usually requires more state variables. This will be elaborated in the following sections as we discuss specific elastic-plastic material models.

For general elastic-plastic materials, the free energy depends on both the elastic and plastic strains. In most material models, it can be assumed that the free energy Ψ can be decomposed into elastic and plastic parts:

$$\Psi = \Psi_{el} + \Psi_{pl} \tag{104.641}$$

The total rate of work associated with the effective stress can be written as the sum of an elastic and plastic component:

$$\hat{W}^{el} \equiv \sigma_{ij}\dot{\epsilon}^{el}_{ij} = \dot{\Psi}_{el} \tag{104.642}$$

and

$$\hat{W}^{pl} \equiv \sigma_{ij}\dot{\epsilon}^{pl}_{ij} = \dot{\Psi}_{pl} + \Phi \tag{104.643}$$

Note that the focus of this section is the energy dissipation caused by material plasticity, which should be distinguished from viscous coupling and other sources of energy dissipation. So the effects of solid-fluid interaction are neglected and all stresses are defined as effective stresses in further derivations. In order to avoid confusion, the common notation $(\sigma^{'}_{ij})$ will not be used. Standard definition of stress from mechanics of materials, i.e. positive in tension, is used.

In the case of a decoupled material, the elastic free energy Ψ_{el} depends only on the elastic strains, and the plastic free energy Ψ_{pl} depends only on the plastic strains, as shown by Collins and Houlsby (1997):

$$\Psi = \Psi_{el}(\epsilon^{el}_{ij}) + \Psi_{pl}(\epsilon^{pl}_{ij}) \tag{104.644}$$

The effective stress can also be decomposed into two parts:

$$\sigma_{ij} = \alpha_{ij} + \chi_{ij} \tag{104.645}$$

lwhere χ_{ij} is a stress-like variable that is related to the dissipative behavior of elastic-plastic material. The difference between the actual stress σ_{ij} and the stress-like variable χ_{ij} is another stress-like term α_{ij}, which is defined from the plastic free energy function Ψ_{pl}. In simple kinematic hardening models, this variable α_{ij} controls the shift behavior of stress under cyclic loading, and thus usually referred to as shift or back stress.

Ziegler's orthogonal postulate Ziegler and Wehrli (1987) ensures the validity of Equation 104.645. It is equivalent to the maximum entropy production criterion, which is necessary to obtain unique formulation. Also, this is a weak assumption so that all the major continuum models of thermo-mechanics are included. Equation 104.643 of plastic work rate can hence be rewritten as:

$$\hat{W}^{pl} \equiv \sigma_{ij}\dot{\epsilon}_{ij}^{pl} = \dot{\Psi}_{pl} + \Phi = \alpha_{ij}\dot{\epsilon}_{ij}^{pl} + \chi_{ij}\dot{\epsilon}_{ij}^{pl} \tag{104.646}$$

The plastic work $\hat{W}^{pl}$ is the product of the actual Cauchy stress σ_{ij} with the plastic strain rate $\dot{\epsilon}_{ij}^{pl}$, while the dissipation rate Φ is the product of the stress variable χ_{ij} with the plastic strain rate $\dot{\epsilon}_{ij}^{pl}$. They are only equal if the rate of plastic free energy $\dot{\Psi}_{pl}$ is zero, or equivalently, if the free energy depends only on the elastic strains.

In kinematic hardening models, where the back stress describes the translation (or rotation) of the yield surface, the decomposition of the true stress (sum of back stress and dissipative stress) is a default assumption. Although such a shift stress is important for anisotropic material models, Collins and Kelly (2002) have pointed out that it is also necessary in isotropic models of geomaterials with different strength in tension and compression.

104.11.2.2 Plastic Free Energy

A popular conceptual model, which focused on particulate materials and demonstrated the physical occurrence of shift stresses, was described by Besseling and Van Der Giessen (1994) and Collins and Kelly (2002). On macro (continuum) scale, every point in a given element is at yield state and deforms plastically. But on meso-scale, only part of this element is undergoing plastic deformations, the remaining part is still within yield surface and respond elastically. The elastic strain energy stored in the elastic part of a plastically deformed macro-continuum element is considered to be locked into the macro-deformation, giving rise to the plastic free energy function Ψ_{pl} and its associated back stress α_{ij}. This energy can be released only when the plastic strains are reversed.

For better explanation, the nature of plastic free energy in particulate materials is illustrated through a finite element simulation combined with considerations of particle rearrangement on mesoscopic scale. Figure 104.80 shows stress-strain response of Drucker-Prager with nonlinear Armstrong-Frederick kinematic hardening, a typical elastic-plastic model for metals and geomaterials. Six states during shear are chosen to represent evolution of micro fabric of the numerical sample. Correspondingly, Figure 104.81 shows the process of particle rearrangement of the 2D granular assembly under cyclic shearing from microscopic level. The square window can be roughly considered as a representative volume (a constitutive level or a finite element) in FEM.

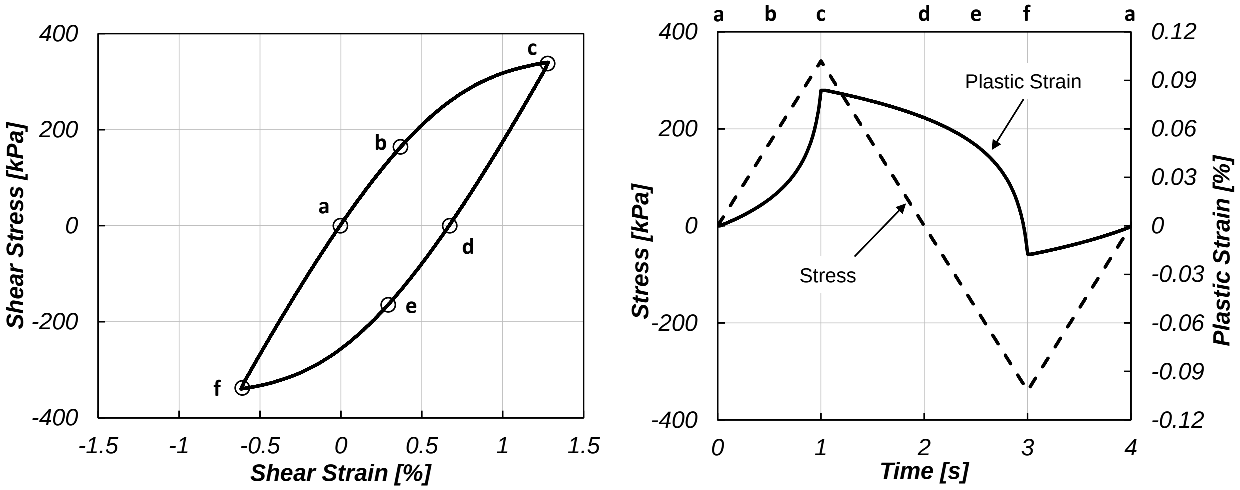


Figure 104.80: Elastic-plastic material modeled with Drucker-Prager yield function and Armstrong-Frederick kinematic hardening under cyclic shear loading: (a) Stress-strain curve; (b) stress and plastic strain versus time.

By discussing movement and energy of particle A in Figure 104.81, the physical nature of plastic free energy is illustrated. At state (a), which is the beginning of deformation, particle A does not bear any load other than its self weight. State (b) is in middle of loading, when particle B pushes downwards to particle A until it makes contact with particle D and E. Load reaches peak at state (c), and there's no space for particle A to move. Then the sample is unloaded to state (d). Particle A is now stuck between particles C, D, and F, which means that certain amount of elastic energy is stored due to particle elastic deformation. Compared with state (a), this part of elastic energy is not released when the sample is unloaded, which indicates that it's not classic strain energy. This part of elastic energy on particle level which can't be released by unloading is defined as the plastic free energy in granular materials. Reverse loading starts at state (e), where particle D pushes particle A upwards, making it squeeze through particle C and F. Elastic energy on particle level, which is now defined as plastic free energy, is released during

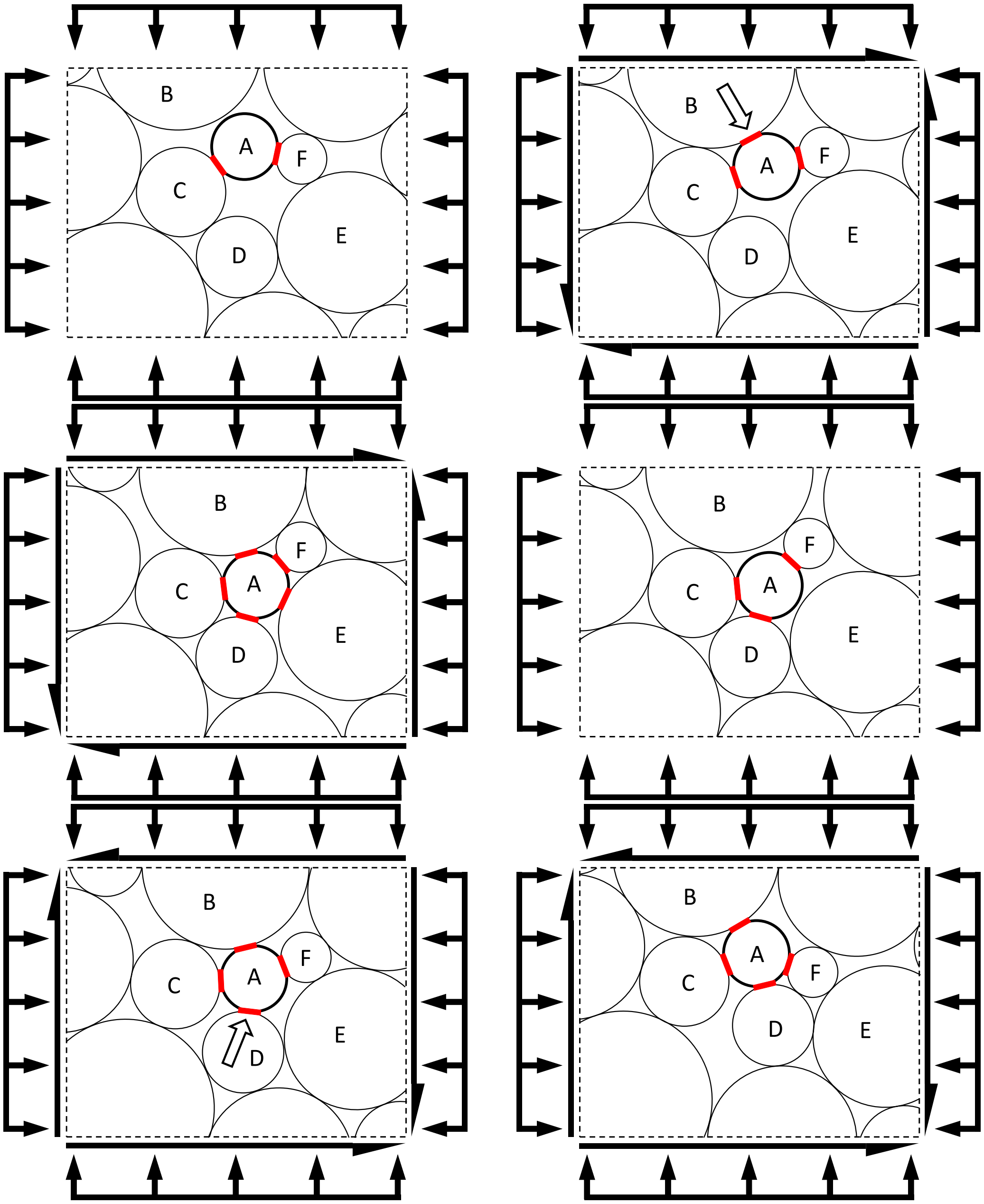


Figure 104.81: Particle rearrangement of a 2D granular assembly under cyclic shearing: (a) Initial state; (b) Loading (accumulating plastic free energy); (c) End of loading (maximum plastic free energy); (d) Unloading (plastic free energy unchanged); (e) Reverse loading (releasing plastic free energy); (f) End of reverse loading (plastic free energy released).

reverse loading.

By analyzing this example, an explanation on particle scale is provided for the origin of plastic free energy in granular materials. It is important to note that the concept of plastic free energy also exists in metals and other materials, as studied by Dafalias et al. (2002) and Feigenbaum and Dafalias (2007). The physical nature of plastic free energy in these materials can be different and probably should be studied on molecular and/or crystalline level.

Collins Collins and Kelly (2002), Collins (2003) suggested that in the case of granular materials, the particle-level plastic energy dissipation during normal compaction, arises from the plastic deformations occurring at the inter-granular contacts on the strong force chains, that are bearing the bulk of the applied loads. Collins also suggested that the locked-in elastic energy is produced in the weak force networks, where the local stresses are not large enough to produce plastic deformation at the grain contacts. The plastic strains can be associated with the irreversible rearrangement of the particles, whilst the elastic energy arises from the elastic compression of the particle contacts. Part of this elastic strain energy will be released during unloading, however other part of this energy will be trapped as a result of the irreversible changes in the particle configuration.

104.11.2.3 Plastic Dissipation

As pointed out, plastic work and energy dissipation are not the same physical quantity. The confusion of these two concepts often leads to incorrect results and conclusions, especially in seismic energy dissipation analysis. Of major concern in this section is the computation of plastic dissipation, as elaborated in this section.

With the decoupling assumption (Equation 104.644), the second law of thermodynamics (positive entropy production) directly leads to the dissipation inequality, which states that the energy dissipated due to the difference of the plastic work rate and the rate of the plastic part of the free energy must be non-negative:

$$\Phi = \sigma_{ij}\dot{\epsilon}_{ij}^{pl} - \dot{\Psi}_{pl} = \sigma_{ij}\dot{\epsilon}_{ij}^{pl} - \rho\dot{\psi}_{pl} \geq 0 \qquad (104.647)$$

where $\dot{\psi}_{pl}$ is the rate of plastic free energy, per unit mass, and ρ is the mass density. In addition, ψ_{pl} denotes plastic free energy density, which is generally not constant at different locations in a body. This expression is closer to physics and makes it convenient for further derivations.

Now we proceed to consider how to calculate plastic free energy, which can then be used to calculate dissipation. According to Feigenbaum and Dafalias (2007), plastic free energy density ψ_{pl} is assumed to be additively decomposed into parts which correspond to the isotropic, kinematic and distortional hardening mechanisms as follows:

$$\psi_{pl} = \psi_{pl}^{iso} + \psi_{pl}^{ani}; \quad \psi_{pl}^{ani} = \psi_{pl}^{kin} - \psi_{pl}^{dis} \tag{104.648}$$

where ψ_{pl}^{iso}, ψ_{pl}^{ani}, ψ_{pl}^{kin}, and ψ_{pl}^{dis} are the isotropic, anisotropic, kinematic, and distortional parts of the plastic free energy, respectively. The anisotropic part is assumed to decompose into kinematic and distortional parts, which correspond to different hardening models. The subtraction, instead of addition, of ψ_{pl}^{dis} from ψ_{pl}^{kin}, to obtain the overall anisotropic part ψ_{pl}^{ani} of the plastic free energy, is a new concept proposed by Feigenbaum and Dafalias (2007). This expression can better fit experimental data, as well as satisfy the plausible expectations for a limitation of anisotropy development. The distortional part of the plastic free energy ψ_{pl}^{dis} is related to the directional distortion of yield surface and will only be present if the material model incorporates distortional strain hardening, which is not considered in the formulations and examples of this study.

As pointed out by Dafalias et al. (2002), the thermodynamic conjugates to each of the internal variables exist and each part of the plastic free energy can be assumed to be only a function of these conjugates. The explicit expressions for the isotropic and kinematic components of the plastic free energy are:

$$\psi_{pl}^{iso} = \psi_{pl}^{iso}(\bar{k}) = \frac{\kappa_1}{2\rho}\bar{k}^2; \quad \psi_{pl}^{kin} = \psi_{pl}^{kin}(\bar{\alpha}_{ij}) = \frac{a_1}{2\rho}\bar{\alpha}_{ij}\bar{\alpha}_{ij} \tag{104.649}$$

where $\bar{k}$ and $\bar{\alpha}_{ij}$ are the thermodynamic conjugates to k (size of the yield surface) and α_{ij} (deviatoric back stress tensor representing the center of the yield surface), respectively. Material constants κ_1 and a_1 are non-negative material constants whose values depend on the choice of elastic-plastic material models.

According to definition, the thermodynamic conjugates are related to the corresponding internal variables by:

$$k = \rho\frac{\partial\psi_{pl}^{iso}}{\partial\bar{k}} = \kappa_1\bar{k}; \quad \alpha_{ij} = \rho\frac{\partial\psi_{pl}^{kin}}{\partial\bar{\alpha}_{ij}} = a_1\bar{\alpha}_{ij} \tag{104.650}$$

By substituting Equation 104.650 back into Equation 104.649, the plastic free energy can be expressed in terms of the internal variables:

$$\psi_{pl}^{iso} = \frac{1}{2\rho\kappa_1}k^2; \quad \psi_{pl}^{kin} = \frac{1}{2\rho a_1}\alpha_{ij}\alpha_{ij} \tag{104.651}$$

With Equation 104.651, the components of plastic free energy can be computed, as long as the internal variables are provided. Combining Equation 104.647 with 104.651, the plastic dissipation in a given elastic-plastic material can be accurately obtained at any location, at any time. This approach allows engineers and designers to correctly identify energy dissipation in time and space and make appropriate conclusions on material behavior.

104.11.2.4 Energy Computation in Finite Elements

Formulations from the previous section are applied to FEM analysis in order to follow energy dissipation. Energy density is chosen as the physical parameter for energy analysis. Energy density in this study is defined as the amount of energy stored in a given region of space per unit volume.

For FEM simulations, both external forces and displacements can be prescribed. The finite element program accepts either (or both) forces and/or displacements as input and solves for the other. Either way, the rate of input work can be calculated by simply multiplying force and displacement within a time step. Therefor input work of a finite element model is:

$$W_{Input}(t) = \int_0^t \dot{W}_{Input}(T)dT = \int_0^t \sum_i F_i^{ex}(\boldsymbol{x}, T)\dot{u}_i(\boldsymbol{x}, T)dT \tag{104.652}$$

where F_i^{ex} is the external force and u_i is the displacement computed at the location of the applied load, at given time step, for a load controlled analysis. The external load can have many forms, including nodal loads, surface loads, and body loads. All of them are ultimately transformed into nodal forces. As shown in Equation 104.652, input work is computed incrementally at each time step, in order to obtain the evolution of total input work at certain time.

As shown in Figure 104.82, when loads and/or displacements are introduced into a finite element model, the input energy will be converted in a number of different forms as it propagates through the system. Input energy will be converted into kinetic energy, free energy, and dissipation. As mentioned before, free energy can be further separated into elastic part, which is traditionally defined as strain energy, and plastic part, which is defined as the plastic free energy. Kinetic energy and strain energy can be considered as the recoverable portion of the total energy since they are transforming from one to another. Plastic free energy is more complicated in the sense that it is conditionally recoverable during reverse loading, as has been discussed in detail in previous sections. Other than kinetic energy and free energy, the rest of the input energy is dissipated, transformed into heat or other forms of energy that are irrecoverable.

Calculation of kinetic energy and strain energy is rather straight forward:

$$U_K(\boldsymbol{x}, t) = \frac{1}{2}\rho\dot{u}_{ij}(\boldsymbol{x}, t)\dot{u}_{ij}(\boldsymbol{x}, t) \tag{104.653}$$

$$U_S(\boldsymbol{x}, t) = \int_0^t \dot{U}_S(\boldsymbol{x}, T)dT = \int_0^t \sigma_{ij}(\boldsymbol{x}, T)\dot{\epsilon}_{ij}^{el}(\boldsymbol{x}, T)dT \tag{104.654}$$

where U_K and U_S are the kinetic energy density and strain energy density, respectively.

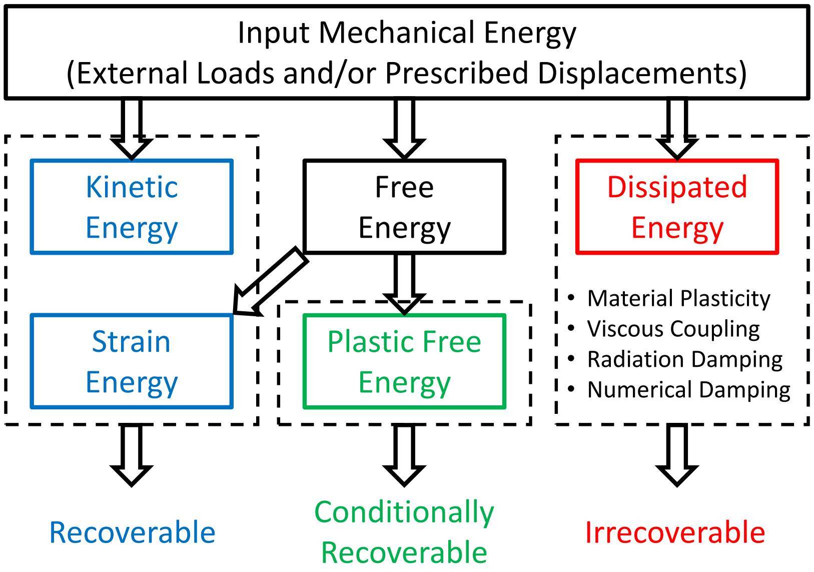


Figure 104.82: Different forms of energy in a dynamic soil-structure system.

Similar to the input energy, strain energy density and plastic free energy are also computed incrementally. Integrating energy density over the entire model, corresponding energy quantities are expressed as:

$$E_K(t) = \int_V U_K(\boldsymbol{x}, t) dV \tag{104.655}$$

$$E_S(t) = \int_V U_S(\boldsymbol{x}, t) dV \tag{104.656}$$

$$E_P(t) = \int_V \Psi_{pl}(\boldsymbol{x}, t) dV \tag{104.657}$$

where E_K, E_S, and E_P are the kinetic energy, strain energy, and plastic free energy of the entire model, respectively. Energy densities, defined in Equations 104.653 and 104.654 are functions of both time and space, while energy components, defined in the above equations (Equation 104.655, 104.656, and 104.657) are only functions of time, since they are integrated over the whole model.

Although the plastic free energy is conditionally recoverable, it is still considered to be stored in the system, rather than dissipated. Summing up all the stored energy E_{Stored}, one obtains:

$$E_{Stored} = E_K + E_S + E_P \tag{104.658}$$

Rate of plastic dissipation, given by Equation 104.647, can be integrated over time and space:

$$D_P(t) = \int_V \int_0^t \Phi(\boldsymbol{x}, T) dT dV \tag{104.659}$$

where D_P is the dissipation due to plasticity of the entire model at certain time.

Finally the energy balance of a finite element model is given by:

$$W_{Input} = E_{Stored} + D_P = E_K + E_S + E_P + D_P \tag{104.660}$$

104.11.3 Numerical Studies

Numerical simulation results presented in this section are performed using the Real-ESSI (Jeremić et al., 1988-2025). Examples in this section focus on constitutive behavior of elastic-plastic material from the perspective of energy dissipation.

All cases are assumed to be static problems. External loads are applied incrementally using load- or displacement-control scheme. System equations are solved using Newton-Raphson iteration algorithm and UMFPACK solver. Standard 8-node-brick elements are used in all cases, in order to eliminate the variation in energy computation caused by different element types.

104.11.3.1 Elastic Material

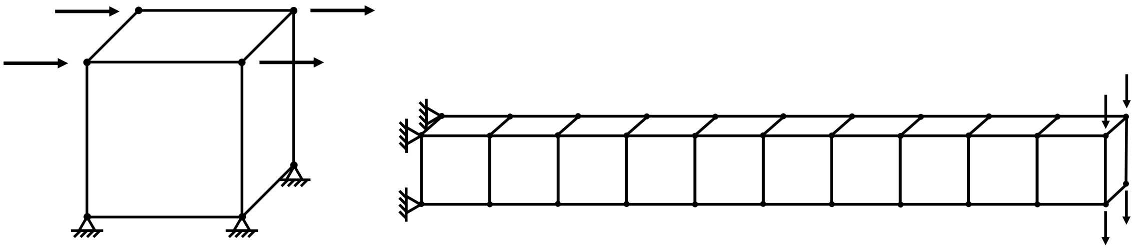


Figure 104.83: Numerical models used in this section: (a) Single brick element; (b) Cantilever with 10 brick elements.

Initial investigation of energy dissipation is focused on linear elastic material. It is noted that linear elastic material does not dissipate energy. Use of linear elastic material model is suitable for preliminary verification of the newly developed energy analysis methodology. In this section, energy balance in a single brick element and a cantilever beam is studied, as shown in Figure 104.83.

It should be mentioned that the bending deformations of cantilever are not accurate due to the use of a single layer of 8-node-brick elements in the direction of stress and strain variation. However, the focus of this example is energy transformation and balance, which are not affected by inaccurate deformations in this example.

The simplest case is a single element model under uniform shear load. The model is constrained appropriately to simulate simple shear test. In order to show the influence of different material parameters and loads, a set of simulations are performed and the results are presented in Table 104.2 and Figure 104.84.

Table 104.2: Energy analysis results for linear elastic materials (single element).

Material Property		Simulation Results						
E (GPa)	ν	u (m)	$\boldsymbol{W_{Input}}$ (J)	E_K (J)	E_S (J)	E_P (J)	$\boldsymbol{E_{Stored}}$ (J)	$\boldsymbol{D_P}$ (J)
100	0.30	2.60E-5	13.00	0.00	13.00	0.00	13.00	0.00
150	0.30	1.73E-5	8.67	0.00	8.67	0.00	8.67	0.00
200	0.30	1.30E-5	6.50	0.00	6.50	0.00	6.50	0.00
250	0.30	1.04E-5	5.20	0.00	5.20	0.00	5.20	0.00
300	0.30	8.67E-6	4.33	0.00	4.33	0.00	4.33	0.00
200	0.20	1.20E-5	6.00	0.00	6.00	0.00	6.00	0.00
200	0.25	1.25E-5	6.25	0.00	6.25	0.00	6.25	0.00
200	0.30	1.30E-5	6.50	0.00	6.50	0.00	6.50	0.00
200	0.35	1.35E-5	6.75	0.00	6.75	0.00	6.75	0.00
200	0.40	1.40E-5	7.00	0.00	7.00	0.00	7.00	0.00

Since linear elastic material is used with static algorithm, energy components related to dynamics (kinetic energy) and plasticity (plastic free energy and plastic dissipation) are equal to zero. This means that all input work is stored in the system, as observed in all cases.

Figure 104.84 shows that energy stored in the system is inversely proportional to Young's moduli E and proportional to one plus Poisson's ratio $(1+\nu)$. This is expected because of the following equations for strain energy under static shear loading:

$$E_S = \frac{1}{2}\tau\gamma = \frac{1}{2G}\tau^2 = \frac{1+\nu}{E}\tau^2 \tag{104.661}$$

Note that these relationships are only valid at constitutive level. For models with more finite elements, stress and strain are generally not uniform. The computation of energy depends on the distribution of energy density, and nonuniform stress/strain distribution will result in nonuniform energy density distribution.

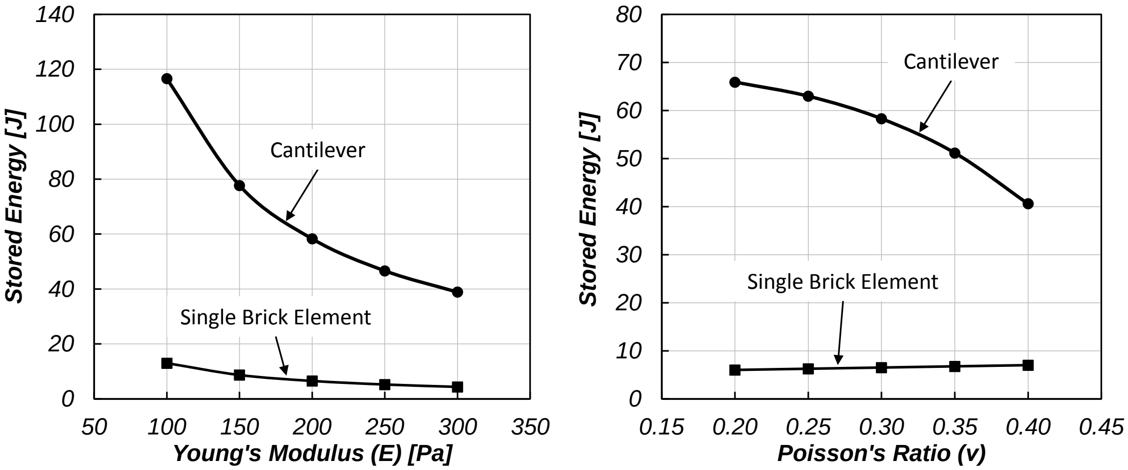


Figure 104.84: Relationships between energy storage and different simulation parameters (single element model): (a) Young's modulus; (b) Poisson's ratio.

In order to study the influence of simulation parameters in larger models, another set of simulations with cantilever model (Figure 104.83b) are performed. Vertical loads are applied to the nodes of the free end. In this case, both shearing and bending occurs, which means that in general a full 3D state of stress and strain is present. The results are presented in Table 104.3 and Figure 104.84. As expected, energy behavior of cantilever is different than the single-element/constitutive example.

For all cases, the energy balance between input and storage is maintained, which gives us confidence on the energy calculation methodology for elastic material. According to results in Figure 104.84, energy stored in the system is still inversely proportional to Young's modulus. This is because the general equation for elastic strain energy density is:

$$E_S = \frac{1}{2E}\left(\sigma_{xx}^2 + \sigma_{yy}^2 + \sigma_{zz}^2 + 2(1+\nu)(\sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{zx}^2)\right) \tag{104.662}$$

So as long as all the elements have the same Young's modulus, the relationship between stored energy and Young's modulus will remain valid.

104.11.3.2 von Mises Plasticity

Elastic-plastic modeling using von Mises material model has been proven to be effective in modeling pressure-independent materials like steel or other metals. In this section, the energy behavior of models using von Mises plasticity with various hardening rules are examined using the proposed method. The material model parameters used in this section are summarized in Table 104.4.

Note that associated plasticity is used in all models in this section, which means that the plastic flow

Table 104.3: Energy analysis results for linear elastic materials (cantilever model).

Material Property		Simulation Results						
E (GPa)	ν	u (m)	$\boldsymbol{W_{Input}}$ (J)	E_K (J)	E_S (J)	E_P (J)	$\boldsymbol{E_{Stored}}$ (J)	$\boldsymbol{D_P}$ (J)
100	0.30	2.33E-3	116.57	0.00	116.57	0.00	116.57	0.00
150	0.30	1.55E-3	77.71	0.00	77.71	0.00	77.71	0.00
200	0.30	1.17E-3	58.28	0.00	58.28	0.00	58.28	0.00
250	0.30	9.33E-4	46.63	0.00	46.63	0.00	46.63	0.00
300	0.30	7.77E-4	38.86	0.00	38.86	0.00	38.86	0.00
200	0.20	1.20E-5	65.89	0.00	65.89	0.00	65.89	0.00
200	0.25	1.26E-3	62.97	0.00	62.97	0.00	62.97	0.00
200	0.30	1.17E-3	58.28	0.00	58.28	0.00	58.28	0.00
200	0.35	1.02E-3	51.17	0.00	51.17	0.00	51.17	0.00
200	0.40	8.12E-4	40.60	0.00	40.60	0.00	40.60	0.00

direction m_{ij} is equal to the gradient of the yield surface $n_{ij}(= \partial f / \partial \sigma_{ij})$. Since the yield function is of von Mises type, associated plasticity leads to the result that only deviatoric plastic flow will appear in all cases.

No Hardening (Elastic-Perfectly Plastic). In this example, elastic-perfectly plastic material is used. Equations 104.647 and 104.651 indicate that in the case of no hardening the rate of plastic free energy is zero. Then the incremental plastic work is equal to incremental plastic dissipation. Note that this is one of the rare cases where plastic dissipation equals to plastic work.

Figure 104.85 shows stress–strain curve (left) and energy calculated for elastic-perfectly plastic constitutive model (right) used here.

In this case, the plastic dissipation is equal to the plastic work. This means that the plastic free energy does not develop at all during loading and unloading. Zero plastic free energy points out the absence of fabric evolution of a particulate, elastic-plastic material, as all the input work is dissipated through particle to particle friction. Since there is no plastic free energy E_P in this case, the stored energy equals to mechanical energy, which is the combination of strain energy E_S and kinetic energy E_K. Total stored energy E_{Stored} develops nonlinearly and always has the same value at the beginning of every loop after the first one. Plastic dissipation D_P increases linearly when the material yields. This can be

Table 104.4: Model parameters for cases using von Mises plasticity

Parameter	Unit	Hardening Type			
		No Hardening	Linear Isotropic	Linear Kinematic	A–F Kinematic
`mass_density`	kg/m^3	8050	8050	8050	8050
`elastic_modulus`	GPa	200	200	200	200
`poisson_ratio`		0.3	0.3	0.3	0.3
`von_mises_radius`	MPa	250	250	250	250
`isotropic_hardening_rate`	GPa		20	0	
`kinematic_hardening_rate`	GPa		0	50	
`armstrong_frederick_ha`	GPa				200
`armstrong_frederick_cr`					100

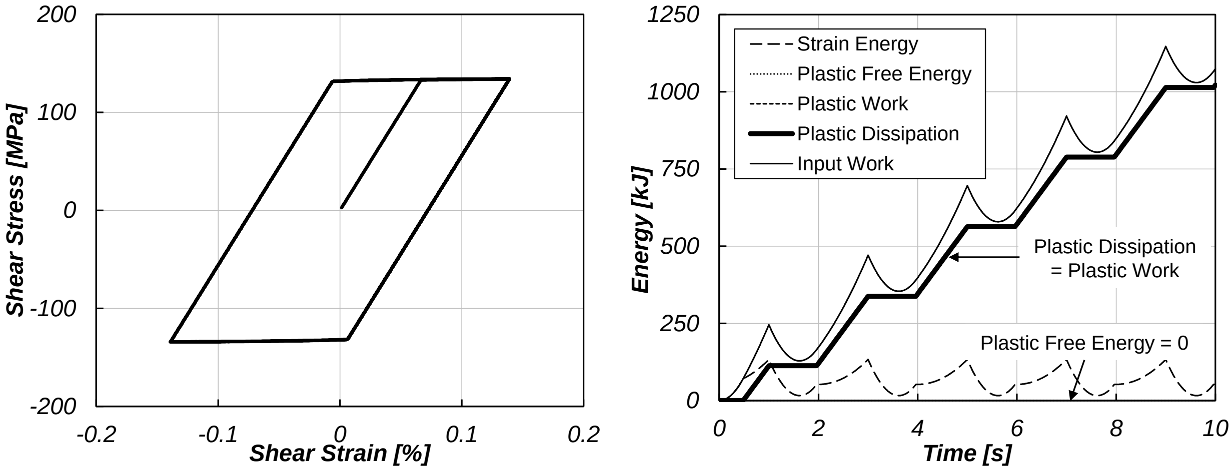


Figure 104.85: Energy analysis of elastic-plastic material modeled using von Mises plasticity with no hardening: (a) Stress–strain curve; (b) Input work, plastic dissipation, strain energy and plastic work.

explained by rewriting Equation 104.647 with $\Psi_{pl} = 0$:

$$\Phi = \sigma_{ij}\dot{\epsilon}_{ij}^{pl} \tag{104.663}$$

where stress σ_{ij} is constant after elastic perfectly plastic material yields, and rate plastic deformation $\dot{\epsilon}_{ij}^{pl}$ is also constant. Then the rate of plastic dissipation is constant which makes the plastic dissipation D_P increase linearly.

Linear Isotropic Hardening. Next material model used is von Mises plasticity with linear isotropic hardening. First used to model monotonic behavior of elastic-plastic materials, isotropic hardening assumes that the yield surface maintains shape, while isotropically (proportionally) changing its size. Figure 104.86 illustrates the stress–strain response as well as energy balance for elastic-plastic material with isotropic hardening.

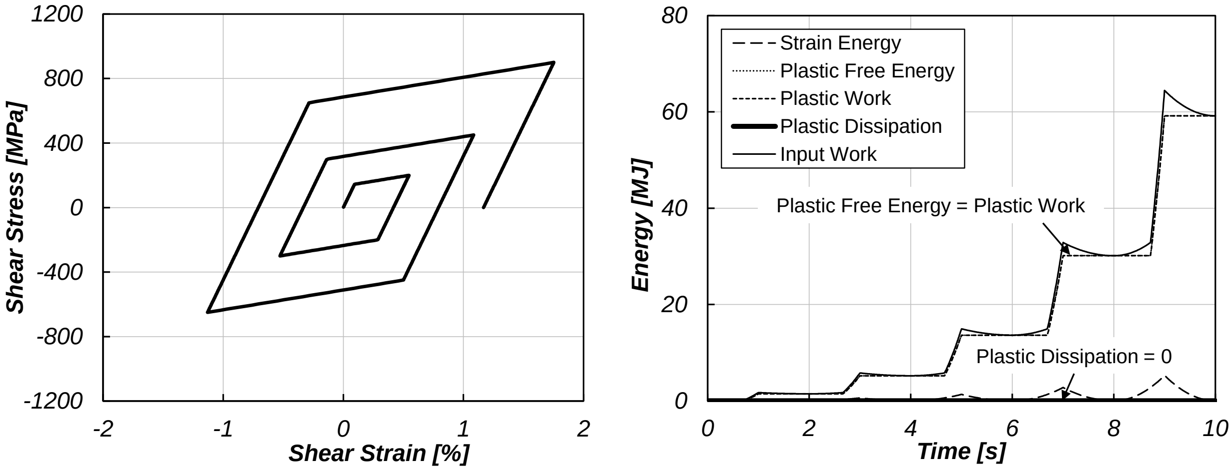


Figure 104.86: Energy analysis of elastic-plastic material modeled using von Mises plasticity with linear isotropic hardening: (a) Stress–strain curve; (b) Input work, plastic dissipation, strain energy, and plastic work.

As can be observed from Figure 104.86, plastic free energy is equal to the plastic work, which means that the plastic dissipation is zero during cycles of loading. Even though this might sound surprising, it can be explained using basic thermodynamics. Linear isotropic hardening, used in this case, can be described through a rate of the internal variable (size of the yield surface) $\dot{k}$ as:

$$\dot{k} = \kappa_1 |\dot{\epsilon}_{ij}^{pl}| \tag{104.664}$$

where $|\dot{\epsilon}_{ij}^{pl}|$ is the magnitude of the rate of plastic strain while κ_1 is a hardening constant. The hardening constant κ_1 is denoted as `isotropic_hardening_rate` in Table 104.4. Substituting previous equation into Equation 104.651 yields:

$$\psi_{pl} = \psi_{pl}^{iso} = \frac{\kappa_1}{2\rho} \epsilon_{ij}^{pl} \epsilon_{ij}^{pl} \tag{104.665}$$

Take the time derivative of the above equation:

$$\dot{\psi}_{pl} = \frac{\kappa_1}{\rho} \epsilon_{ij}^{pl} \dot{\epsilon}_{ij}^{pl} \tag{104.666}$$

Then the rate of dissipation due to plasticity can be expressed as:

$$\Phi = \sigma_{ij}\epsilon_{ij}^{pl} - \rho\dot{\psi}_{pl} = (\sigma_{ij} - \kappa_1\epsilon_{ij}^{pl})\dot{\epsilon}_{ij}^{pl} = (\sigma_{ij} - km_{ij})\dot{\epsilon}_{ij}^{pl} \tag{104.667}$$

where m_{ij} is the plastic flow direction. The plastic flow direction defines the direction of incremental plastic strain, which can be different from the direction of total plastic strain. But in the case of associated von Mises plasticity with only isotropic hardening, the plastic flow direction m_{ij} is the same as the direction of the total plastic strain ϵ_{ij}^{pl}. Thus we have $\kappa_1\epsilon_{ij}^{pl} = km_{ij}$ in the above equation.

Substitute the plastic flow direction m_{ij} with the gradient of yield surface n_{ij}, and also note that $\sigma_{ij}\dot{\epsilon}_{ij}^{pl} = s_{ij}\dot{\epsilon}_{ij}^{pl}$, where $s_{ij}(= \sigma_{ij} - 1/3\delta_{ij}\sigma_{kk})$ is the deviatoric part of the stress tensor, the rate of plastic dissipation can be rewritten as:

$$\Phi = (s_{ij} - kn_{ij})\dot{\epsilon}_{ij}^{pl} = \alpha_{ij}\dot{\epsilon}_{ij}^{pl} \tag{104.668}$$

Realizing that the back stress α_{ij} is always zero since we assume no kinematic hardening, then the rate of plastic dissipation becomes zero, which means there is no energy dissipation during cycles of loading for isotropically hardening material. Obviously, the observed response is not physical from the perspective of energy dissipation. Therefore, isotropic hardening material models cannot properly model energy dissipation, even for monotonic loading.

Prager Linear Kinematic Hardening. Compared with isotropic hardening, kinematic hardening can better describe the constitutive, stress-strain behavior of elastic-plastic materials, particularly for cyclic loading. Elastic-plastic material that relies on kinematic hardening is used to analyze energy dissipation. Both linear and nonlinear kinematic hardening rules are investigated in relation to energy dissipation.

Prager's linear kinematic hardening rule is given as:

$$\dot{\alpha}_{ij} = a_1\dot{\epsilon}_{ij}^{pl} \tag{104.669}$$

where a_1 is a hardening constant. The hardening constant a_1 is denoted as `kinematic_hardening_rate` in Table 104.4.

If only linear kinematic hardening (Equation 104.669) is assumed, the back stress α_{ij} is expressed explicitly, and can be substituted into Equation 104.651 yielding:

$$\psi_{pl} = \psi_{pl}^{kin} = \frac{a_1}{2\rho}\epsilon_{ij}^{pl}\epsilon_{ij}^{pl} \tag{104.670}$$

Take the time derivative of the above equation:

$$\dot{\psi}_{pl} = \frac{a_1}{\rho}\epsilon_{ij}^{pl}\dot{\epsilon}_{ij}^{pl} \tag{104.671}$$

Then the rate of dissipation due to plasticity can be rewritten as:

$$\Phi = \sigma_{ij}\dot{\epsilon}_{ij}^{pl} - \rho\dot{\psi}_{pl} = (s_{ij} - \alpha_{ij})\dot{\epsilon}_{ij}^{pl} = k m_{ij}\dot{\epsilon}_{ij}^{pl} \tag{104.672}$$

Notice that the term $m_{ij}\dot{\epsilon}_{ij}^{pl}$ denotes the magnitude of the rate of plastic strain. Since only linear kinematic hardening is assumed, the internal variable k will remain constant. So if loads are applied in such a way that the rate of plastic strain is constant, then the rate of dissipation will also remain constant. In other words, the accumulated dissipation will be linearly increasing under the assumption of linear kinematic hardening.

Figure 104.87 shows stress–strain response (left) and energy computation results (right) of an elastic-plastic material modeled using von Mises plasticity with linear kinematic hardening.

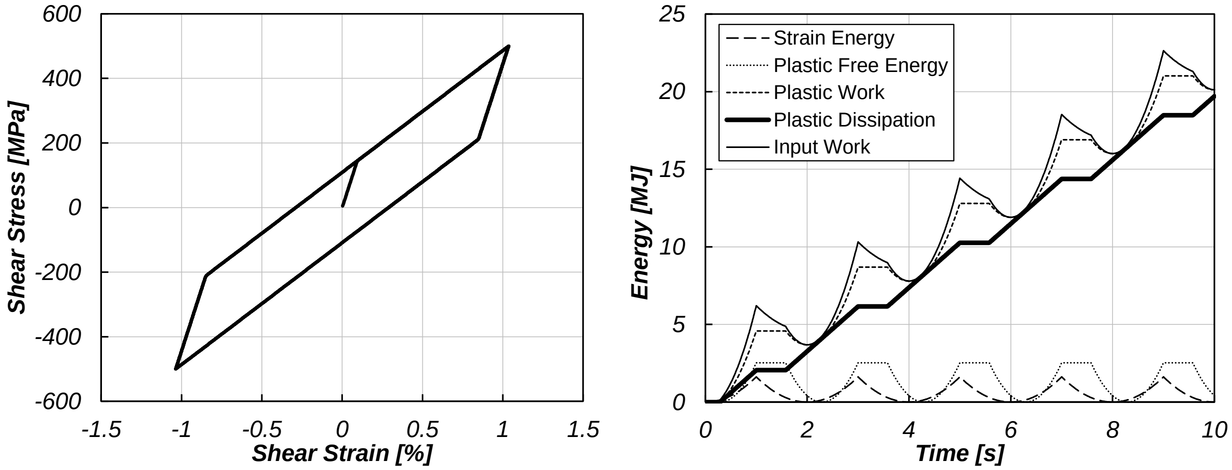


Figure 104.87: Energy analysis of elastic-plastic material modeled using von Mises plasticity with linear kinematic hardening: (a) Stress–strain curve; (b) Input work, plastic dissipation strain energy and plastic work.

As expected, the plastic dissipation increases linearly once the material yields. In contrast to the isotropic hardening case, a significant amount of the input work is dissipated due to material plasticity. The ratio of dissipated energy to input work is largely influenced by the material parameters. However, in general, energy dissipation will be observed if kinematic hardening model is used.

Another important observation is that the plastic work decreases during certain phases of reverse loading, while the actual rate of energy dissipation is always nonnegative. It is important to distinguish plastic work from plastic energy dissipation. Otherwise, one might argue that accumulated energy dissipation can increase or decrease, which is a common mistake observed in a number of publications that violates the second law of thermodynamics.

Armstrong-Frederick Kinematic Hardening. Armstrong-Frederick kinematic hardening model Armstrong and Frederick (1966) is often used to simulate elastic-plastic material behavior under cyclic loading. Material parameters of the Armstrong-Frederick kinematic hardening rule can be derived from basic thermodynamics. The following equation is a general expression for Armstrong-Frederick kinematic hardening rule:

$$\dot{\alpha}_{ij} = a_1 \dot{\epsilon}_{ij}^{pl} - a_2 \dot{\lambda} \alpha_{ij} \tag{104.673}$$

where $\dot{\lambda}$ is a non-negative scalar plastic multiplier and a_2 is a non-negative material hardening constant. It can be proven that a_1/a_2 is related to the limit of back stress magnitude $\left|\alpha_{ij}\right|$. In Table 104.4, the hardening constants a_1 and a_2 correspond to parameters `armstrong_frederick_ha` and `armstrong_frederick_cr`.

Taking the time derivative of the kinematic part of plastic free energy (Equation 104.671), and substituting the expression of back stress α_{ij} (Equation 104.673) gives:

$$\dot{\psi}_{pl}^{kin} = \frac{1}{\rho a_1} \alpha_{ij} \dot{\alpha}_{ij} = \frac{1}{\rho} \alpha_{ij} (\dot{\epsilon}_{ij}^{pl} - \frac{a_2}{a_1} \dot{\lambda} \alpha_{ij}) \tag{104.674}$$

Then the rate of plastic energy dissipation of an Armstrong-Frederick kinematic hardening elastic-plastic material is given by:

$$\Phi = \sigma_{ij} \dot{\epsilon}_{ij}^{pl} - \rho \dot{\psi}_{pl} = s_{ij} \dot{\epsilon}_{ij}^{pl} - \alpha_{ij} \dot{\epsilon}_{ij}^{pl} + \frac{a_2}{a_1} \dot{\lambda} \alpha_{ij} \alpha_{ij} = k m_{ij} \dot{\epsilon}_{ij}^{pl} + \frac{a_2}{a_1} \dot{\lambda} \alpha_{ij} \alpha_{ij} \tag{104.675}$$

Compared with Equation 104.672, the above expression has an additional term which makes the rate of plastic dissipation non-constant even if the rate of plastic strain is constant. As the back stress α_{ij} becomes larger when load increases, the rate of plastic dissipation also increases. This indicates a nonlinear result of total plastic dissipation, which is exactly what we have observed in our computations.

Figure 104.88 shows the energy computation results of an elastic-plastic material modeled using von Mises plasticity with Armstrong-Frederick kinematic hardening. Compared to all previous cases, the material response of this model is more sophisticated and more realistic. Decrease of plastic work is observed, again, while the plastic dissipation is always nonnegative during the entire simulation. For both linear and nonlinear kinematic hardening cases, the plastic free energy is relatively small compared to the plastic dissipation.

104.11.3.3 Drucker–Prager Plasticity

It has been proven that von Mises plasticity generally performs poorly in modeling pressure-sensitive materials like soils. In this section, the thermomechanical formulations presented in earlier sections are

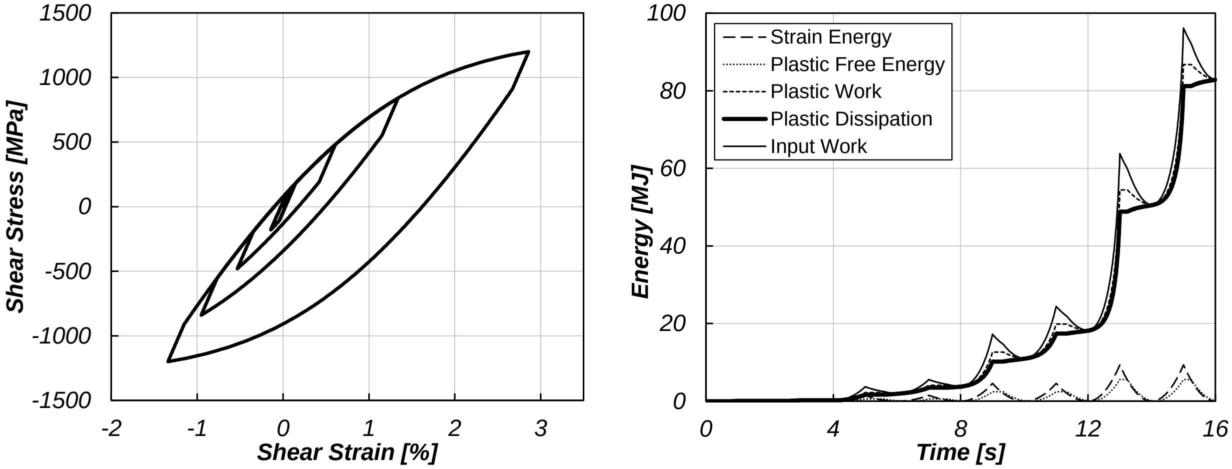


Figure 104.88: Energy analysis of elastic-plastic material modeled using von Mises plasticity with Armstrong–Frederick kinematic hardening: (a) Stress–strain curve; (b) Input work, plastic dissipation, strain energy, and plastic work.

Table 104.5: Model parameters for cases using Drucker–Prager plasticity

Parameter	Unit	Hardening Type				
		Linear Isotropic	Linear Kinematic	A–F Kinematic		
`mass_density`	kg/m^3	2000	2000	2000	2000	2000
`elastic_modulus`	MPa	150	150	200	200	200
`poisson_ratio`		0.3	0.3	0.3	0.3	0.3
`druckerprager_k`		0.25	0.5	0.1	0.1	0.1
`confining_stress`	kPa	100	100	100	300	500
`isotropic_hardening_rate`		50	0			
`kinematic_hardening_rate`		0	50			
`armstrong_frederick_ha`	MPa			20	20	20
`armstrong_frederick_cr`				100	100	100

applied to models using Drucker–Prager yield criteria with different hardening types. The material model parameters used in this section are summarized in Table 104.5.

The yield function of Drucker–Prager plasticity is:

$$f = [(s_{ij} - p\alpha_{ij})(s_{ij} - p\alpha_{ij})]^{0.5} - \sqrt{\frac{2}{3}}kp \tag{104.676}$$

where $p = (1/3)\delta_{ij}\sigma_{kk}$ is the mean stress (or hydrostatic pressure). Note that in this form of Drucker–Prager plasticity, the internal variables α_{ij} and k, as well as the hardening constants κ_1 and a_1, are dimensionless.

For the computation of plastic free energy in Drucker–Prager plasticity models, Equation 104.651 is modified:

$$\psi_{pl}^{iso} = \frac{1}{2\rho(\kappa_1 p)}(kp)^2 = \frac{1}{2\rho\kappa_1}k^2 p \tag{104.677}$$

$$\psi_{pl}^{kin} = \frac{1}{2\rho(a_1 p)}(\alpha_{ij}p)(\alpha_{ij}p) = \frac{1}{2\rho a_1}\alpha_{ij}\alpha_{ij}p \tag{104.678}$$

All examples presented in this section are using non-associated Drucker–Prager plasticity. The plastic potential function is of von Mises type so that only deviatoric plastic flow exists. In addition, all cases are loaded with constant hydrostatic pressure, which means that the plastic flow direction m_{ij} is the same as the direction of the total plastic strain ϵ_{ij}^{pl}. With the above conditions, Equation 104.663, 104.668, and 104.672 are all still valid for the following examples, as will be observed in their results.

It should be noted that the proposed energy computation approach can also be applied to associated Drucker–Prager plasticity and non-associated Drucker–Prager plasticity with different plastic potential functions. The loading condition can be arbitrary, even with evolving hydrostatic pressures.

Linear Isotropic Hardening. Figure 104.89 shows the stress–strain response and energy computation results of an elastic-plastic material modeled using Drucker–Prager plasticity with linear kinematic hardening.

No plastic dissipation is observed in this case, which have been theoretically proven in Equation 104.668. This example again indicates that isotropic hardening is not capable of proper modeling of energy dissipation in elastic-plastic materials.

Prager Linear Kinematic Hardening. Figure 104.90 shows the stress–strain response and energy computation results of an elastic-plastic material modeled using Drucker–Prager plasticity with linear kinematic hardening.

Plastic dissipation increases linearly when the material yields, while the plastic work decreases during certain phases of reverse loading. This observation is consistent with the theoretical conclusion drawn from Equation 104.672.

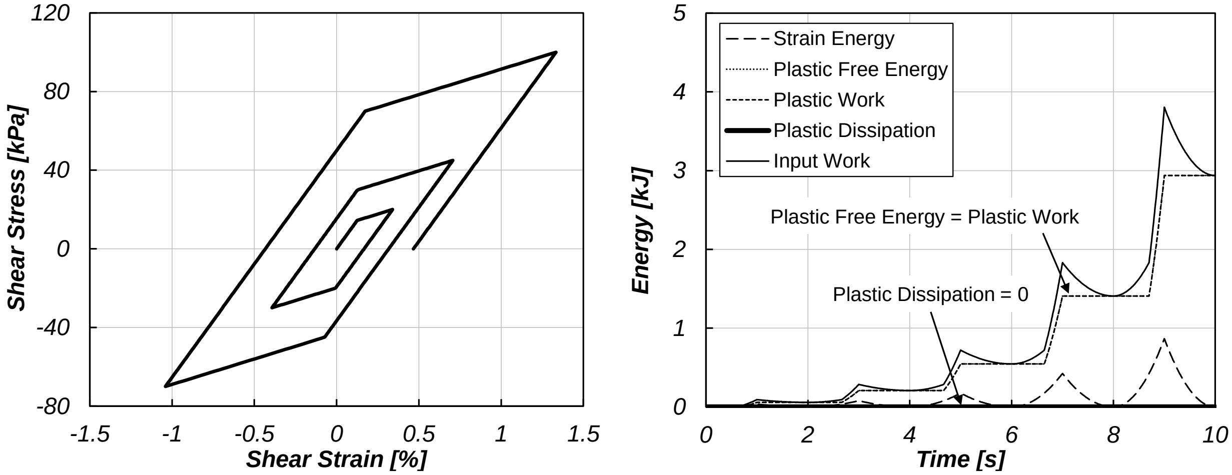


Figure 104.89: Energy analysis of elastic-plastic material modeled using Drucker–Prager plasticity with linear isotropic hardening: (a) Stress–strain curve; (b) Input work, plastic dissipation strain energy and plastic work.

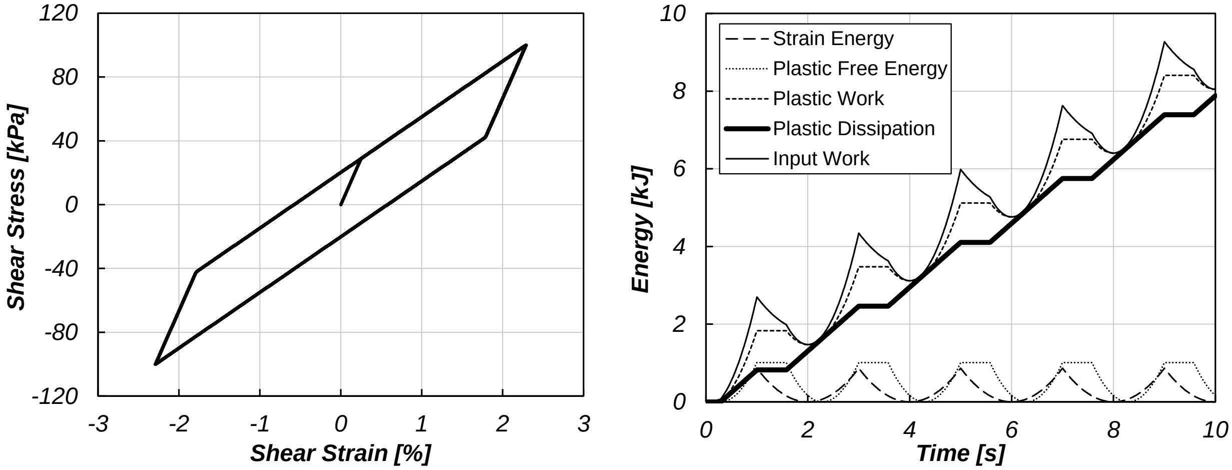


Figure 104.90: Energy analysis of elastic-plastic material modeled using Drucker–Prager plasticity with linear kinematic hardening: (a) Stress–strain curve; (b) Input work, plastic dissipation strain energy and plastic work.

Note that the results of the above two examples share high similarity with those of the cases modeled with associated von Mises plasticity. This is because the hydrostatic pressures were constant during shearing, which makes the pressure-dependent feature of Drucker–Prager plasticity not observed. The energy computation results are expected to be more complicated with different loading conditions. However, the difference between plastic work and plastic dissipation will be observed. And the incremental plastic dissipation should always be nonnegative.

Armstrong–Frederick Kinematic Hardening. In order to illustrate the influence of hydrostatic pressure to the energy dissipation in Drucker–Prager models, three cases with different confining pressures are studied. Armstrong–Frederick kinematic hardening is used here to model the nonlinear hardening response of pressure-dependent material, like soils. Figure 104.91 shows the stress–strain response and energy computation results of these three cases.

As can be observed from Figure 104.91, the slope of stress–strain loop increases, which means the material becomes stiffer, as the confining stress increases. Also, the size of elastic region becomes larger when the confining stress is bigger.

Plastic dissipation and plastic free energy starts to evolve as soon as the material yields. The pattern of evolution of energy components are the same for all three cases, while the value of plastic dissipation decreases as the confining stress increases. This is expected since the material becomes stiffer and harder to plastify with a higher confining stress.

104.11.4 Conclusions

Presented was a methodology for (correct) computation of energy dissipation in elastic-plastic materials based on the second law of thermodynamics. A very important role of plastic free energy was analyzed, with highlights on its physical nature and theoretical formulations. The proposed methodology has been illustrated using a number of elasto-plastic material models.

An analysis of a common misconception that equates plastic work and dissipation, which leads to the violation of the basic principles of thermodynamics, was addressed. A conceptual example, for granular materials, was used to explain the physical meaning of plastic free energy. It was also shown that plastic free energy is responsible for the evolution of internal variables.

It was shown that energy balance is ensured by taking into consideration all energy components, including kinetic and strain energy. Input work was balanced with the stored and dissipated energy, expressed as the summation of all possible components.

Presented approach was illustrated and tested using several elastic-plastic constitutive models with various hardening rules. Elastic materials showed no energy dissipation (as expected), leading to the

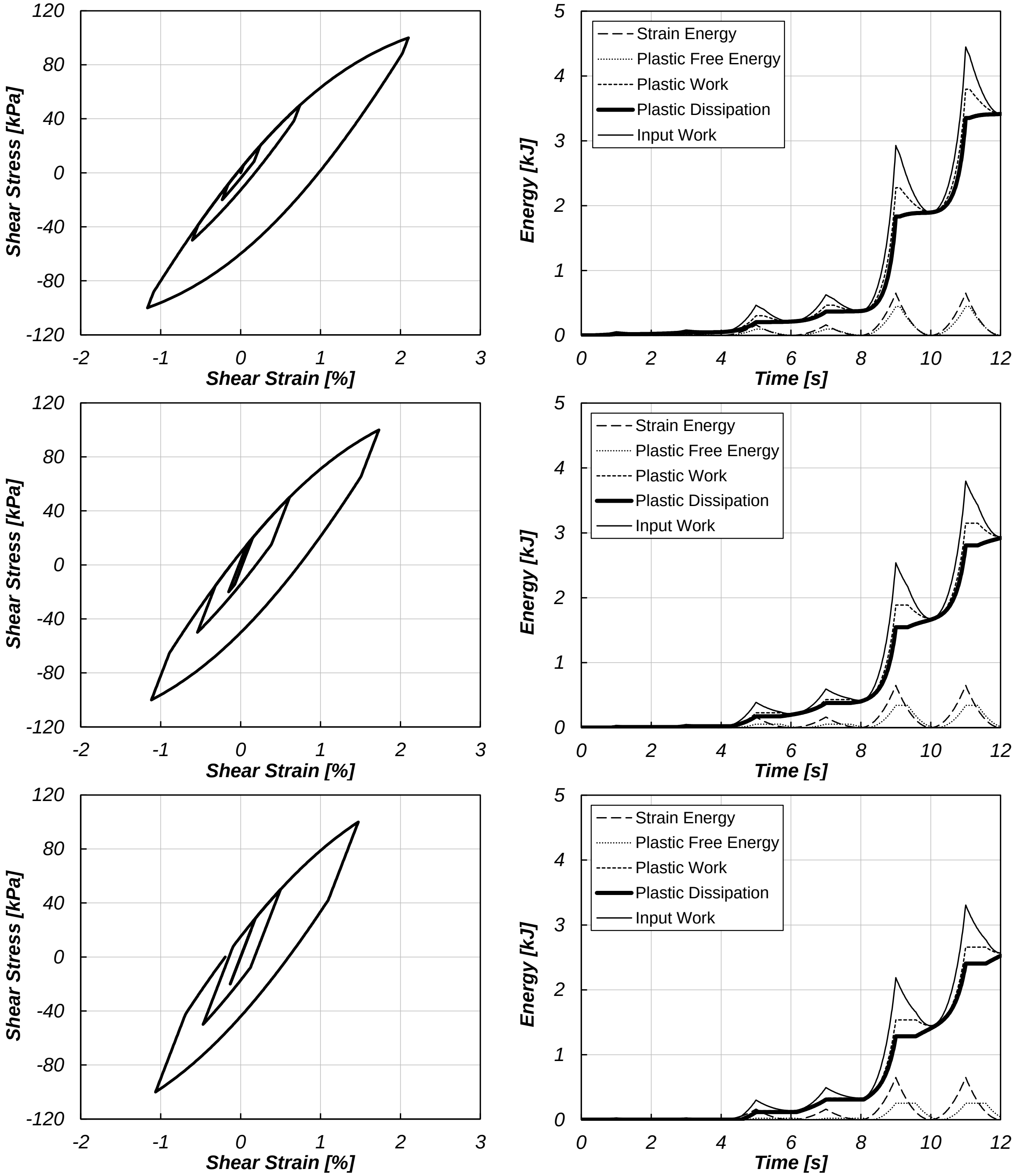


Figure 104.91: Energy analysis of elastic-plastic material modeled using Drucker–Prager plasticity with Armstrong–Frederick kinematic hardening: (a) Confining stress = 100 kPa; (b) Confining stress = 300 kPa; (c) Confining stress = 500 kPa;

input work being equal to the stored energy. Elastic-perfectly plastic materials had no change in plastic free energy, which led to the equality of plastic work and plastic dissipation and indicated no evolution of particle arrangements. The plastic dissipation, in that case, was observed to be increasing linearly. Isotropic hardening materials experienced zero dissipation even after yielding. This observation was surprising, but verified by further derivation of energy equations. This observation also serves as a reminder that the isotropic hardening rules can be used, but only with observed lack of energy dissipation. Prager's linear and Armstrong-Frederick nonlinear kinematic hardening materials both gave significant dissipations, with large fluctuation of plastic free energy as well. In the case with linear kinematic hardening, linear increase of dissipation was derived and observed, while energy was dissipated nonlinearly in the case of nonlinear kinematic hardening. Although the plastic free energy was not significant for some materials, it is noted that it should always be recognized and considered during energy analysis, so that the basic principles of thermodynamics are maintained.

104.12 Energy Dissipation Calculations for Structures

This section is based on Yang et al. (2019a)

104.12.1 Introduction

Mechanical energy in soil structure interaction (SSI) systems are dissipated during the irreversible dissipative process of energy transformation in which entropy of the system increases. Energy dissipation has been used, directly or indirectly, as a key parameter to evaluate damage in elastic-plastic materials. A common misconception of plastic work and energy dissipation due to plasticity has been noticed in a number of publications Uang and Bertero (1990), Léger and Dussault (1992), Symans and Constantinou (1998), Soong and Spencer (2002), Symans et al. (2008), Wong (2008), Nehdi et al. (2010) in which violations of the second law of thermodynamics were observed. As presented in an earlier section (and also by Yang et al. (2018), the correct formulation for energy analysis on elastic-plastic solids has been derived from the second law of thermodynamics. The theoretical and computational framework has been verified through system energy balance in a series of numerical studies on elastic and elastic-plastic material models. The purpose of this section is to present a methodology of correctly evaluating energy dissipation in structural elements, which is crucial in determining the safety and economy of a SSI system.

It has been shown Dafalias et al. (2002), Feigenbaum and Dafalias (2007), Yang et al. (2018) that the difference between plastic work and plastic dissipation is the plastic free energy, or cold work, which can be calculated from material internal variables (or state variables), like radius of yield surface or back stress. This computation can be easily performed on solids modeled with classic elasticity/plasticity constitutive relationships in which internal variables are computed and obtained at every time step. On the other hand, constitutive relationships used to model nonlinear structural elements were proposed mainly based on experimental results Spacone et al. (1996a), Spacone et al. (1996b), Lee and Fenves (1998), Popovics (1973), Mander et al. (1988), Chang and Mander (1994), Waugh (2009), Kolozvari et al. (2015). The internal variables used in these models are different than those used in classic constitutive models for solids. Therefore, a new methodology that can correctly evaluate energy storage and dissipation in structural elements is required.

During the recent few decades, a number of studies have been conducted with focus on energy analysis of SSI systems Uang and Bertero (1990), Léger and Dussault (1992), Kalkan and Kunnath (2007), Kalkan and Kunnath (2008), Symans et al. (2008), Gajan and Saravanathiiban (2011), Moustafa (2011), Moustafa and Mahmoud (2014), Mezgebo and Lui (2017), Deniz et al. (2017). Despite different formulations used, the calculations of energy dissipation due to hysteretic damping (material elasto-plasticity) in these publications were all performed without consideration of plastic free energy, which lead to violations of principles of thermodynamics. It is worth pointing out that such oversight is not rare, especially in literature of civil and geotechnical engineering.

Early work reported by Farren and Taylor (1925) and Taylor and Quinney (1934) showed that plastic free energy could be significant in metals, thus should not be neglected without reasoning. The ratio of plastic work converted into heat, usually referred to as the Quinney–Taylor coefficient, was measured to be between 0.6 to 1.0 Belytschko et al. (1991), Zhou et al. (1996), Dolinski et al. (2010), Osovski et al. (2013). Mason Mason et al. (1994) pointed out that the the Quinney–Taylor coefficient is both strain and strain rate dependent but could be assumed to be a constant in most cases. A constitutive model for metals was presented by Rosakis et al. Rosakis et al. (2000), Hodowany et al. (2000), Ravichandran et al. (2002) based on thermoplasticity, which can model the evolution of energy dissipation and has been validated through experiments. Semnani et al. (2016) presented a thermoplastic framework that could predict strain localization in transversely isotropic materials.

Despite of the existence of sophisticated theories that are capable of modeling the evolution of energy dissipation, including those mentioned earlier, most constitutive relationships used to model structural elements do not involve thermodynamics or thermoplasticity. One commonly used finite element (FE) technique of modeling frame structures is fiber section, in which beams and columns are divided into multiple uniaxial fibers with various constitutive models. This model have been proved to be able to capture nonlinear stress–strain behaviors of structural elements under axial loading and/or bending.

Problems arise when such elements are used to calculate energy dissipation. As observed in many publications Kwan and Billington (2001), Zhu et al. (2006), Gajan and Saravanathiiban (2011), Wang et al. (2012), Zhang et al. (2013), Nikbakht et al. (2014) energy analysis were performed based on the hysteretic stress–strain or force–displacement response of the elements. This indicates that plastic work was confused with plastic energy dissipation, which is the common misconception pointed out earlier. It is also important to point out that various damage indices derived from energy dissipation are used widely to evaluate damage in structures. Such parameters will not be valid if the fundamental formulation of energy dissipation is incorrect.

In order to correctly evaluate energy dissipation in structural elements modeled with fiber sections, the framework of thermo-mechanics must be enforced on the uniaxial constitutive models. Focus of this section is on proper modeling of different forms of energy (storage and dissipation) in uniaxial materials that follows the second law of thermodynamics. Theoretical and computational formulations of energy dissipation in uniaxial concrete and steel fibers are presented. A series of FE simulations are carried out using the Real-ESSI (Jeremić et al., 1988-2025) to illustrate the energy behavior of structural systems. The method is verified by comparing the input work and the energy storage and dissipation in the system. The difference between accumulated plastic work and accumulated plastic dissipation, which can be significant in many cases, is addressed. Finally, conclusions on plastic energy dissipation in structural elements are drawn from the verified results.

104.12.2 Theoretical and Computational Formulations

104.12.2.1 Thermomechanical Framework

The theories of continuum thermo-mechanics have been discussed in a number of earlier publications Lubliner (1972), Rosakis et al. (2000), from which the fundamental framework of this study is derived. General equations of elastoplasticity and thermodynamics are modified with a few plausible assumptions to accommodate the scope of this study. Small deformation theory is assumed, so that the small strain tensor ϵ_{ij} is used to describe deformation of the material body. All equations in this section are expressed in index notation.

The general thermomechanical process is governed by momentum balance and the first and second law of thermodynamics. The localized version of the first law of thermodynamics (energy balance equation) is given in the form:

$$\sigma_{ij}\dot{\epsilon}_{ij} + q_{i,i} + \rho r = \rho \dot{e} \tag{104.679}$$

where the term $\sigma_{ij}\epsilon_{ij}$ is called the stress power, q_i are the components of the heat flux vector, ρ is the mass density of the material, r is the heat supply per unit volume, and e is the internal energy per unit volume. Note that in this section all stresses are defined as effective stresses. In order to avoid confusion, the common notation $(\sigma_{ij}^{'})$ will not be used. Standard definition of stress from mechanics of materials, i.e. positive in tension, is used.

The localized version of the second law of thermodynamics (Clausius–Duhem inequality) is expressed as:

$$\rho\dot{\eta} - (\frac{q_i}{\theta})_{,i} - \frac{1}{\theta}\rho r \geq 0 \tag{104.680}$$

where η is the entropy per unit volume and θ is the absolute temperature.

Substituting the heat supply per unit volume r in Equation 104.680 with the expression from Equation 104.679, and introducing the rate of change of internal dissipation per unit volume Φ gives:

$$\rho\theta\dot{\eta} - \rho\dot{e} + \sigma_{ij}\dot{\epsilon}_{ij} + \frac{1}{\theta}q_i\theta_{,i} = \Phi + \frac{1}{\theta}q_i\theta_{,i} \geq 0 \tag{104.681}$$

Note that the internal dissipation can have many sources, including material plasticity, viscous coupling, radiation damping, and other forms of energy dissipation.

The Helmholtz free energy per unit volume ψ, which is referred to as free energy in this section, is defined as:

$$\psi = e - \theta\eta \tag{104.682}$$

The second law of thermodynamics can be expressed in terms of free energy ψ as:

$$\Phi + \frac{1}{\theta}q_i\theta_{,i} = -\rho\dot{\psi} - \rho\dot{\theta}\eta + \sigma_{ij}\dot{\epsilon}_{ij} + \frac{1}{\theta}q_i\theta_{,i} \geq 0 \tag{104.683}$$

The rate of internal dissipation per unit volume Φ can be written as:

$$\Phi = \sigma_{ij}\dot{\epsilon}_{ij} - \rho\dot{\psi} - \rho\dot{\theta}\eta \tag{104.684}$$

At this point, a few assumptions are introduced to simplify the governing equations. According to Feigenbaum and Dafalias (2007), Collins and Houlsby (1997), Collins (2002), Collins and Kelly (2002), it can be assumed that the deformation of soil and structural elements under earthquake loading is approximately isothermal, which indicates that the temperature field θ is constant and uniform. This approximation is reasonable considering the fact that seismic energy is mostly carried by the low-frequency components of earthquake ground motion, which allows the heat generated in the materials to be largely dissipated. With this assumption, the rate of internal dissipation Φ is simplified into the form:

$$\Phi = \sigma_{ij}\dot{\epsilon}_{ij} - \rho\dot{\psi} \geq 0 \tag{104.685}$$

Next, all material models studied in this section is assumed to be decoupled, which means that the (small) strain tensor can be additively decomposed into elastic and plastic parts:

$$\epsilon_{ij} = \epsilon_{ij}^{el} + \epsilon_{ij}^{pl} \tag{104.686}$$

Lubliner (1972) and Collins and Houlsby (1997) showed that this assumption can be deduced if the instantaneous elastic moduli of a material are independent of the internal variables. Under the assumption of decoupled material, the free energy ψ can also be decomposed into elastic and plastic parts as follows:

$$\psi = \psi_{el} + \psi_{pl} \tag{104.687}$$

where the elastic part of the free energy ψ_{el} is also known as the elastic strain energy, which is defined in incremental form as:

$$\dot{\psi}_{el} = \sigma_{ij}\dot{\epsilon}_{ij}^{el} \tag{104.688}$$

By substituting Equation 104.686, Equation 104.687, and Equation 104.688 into Equation 104.685, the rate of internal dissipation Φ can be expressed in terms of the rate of plastic free energy $\dot{\psi}_{pl}$:

$$\Phi = \sigma_{ij}\dot{\epsilon}_{ij} - \sigma_{ij}\dot{\epsilon}_{ij}^{el} - \rho\dot{\psi}_{pl} \geq 0 \tag{104.689}$$

Equation 504.8 represents two basic principles that should always be upheld in any energy analysis for decoupled material undergoing isothermal process:

- The stress power that is input into a material body by external loading is transformed into elastic strain energy, plastic free energy, and material internal dissipation. All forms of energy must be considered to maintain energy balance of the material body. This principle ensures the first law of thermodynamics.
- The rate of change of material internal dissipation (plastic dissipation) is nonnegative at any time. In other words, accumulated internal dissipation can not decrease during any time period. This principle ensures the second law of thermodynamics.

Note that material internal dissipation can have many sources. Our interest is the energy dissipation caused by material plasticity, so the term plastic dissipation will be used instead, which indicates no other source of energy dissipation is present in the examples that are being analyzed in the remaining part of this section.

104.12.2.2 Plastic Free Energy

The physical nature of plastic free energy is associated with the material micro-structure. For particulate material, like soil, plastic free energy will be accumulated or released if there is evolution of particle arrangement (micro-fabric), which generally happens as soon as the material body is loaded. For other structural and geotechnical materials, like metals, their micro-structures are represented by the shape and arrangement of the crystals, whose evolution will result in change in plastic free energy. Detailed explanations of the evolution of plastic free energy can be found in publications by Besseling and Van Der Giessen (1994), Collins and Kelly (2002), and Yang et al. (2018).

Using Equation 504.8, the energy dissipation of any elastic-plastic material under isothermal loading process can be calculated, if all the terms on the right hand side of the equation is known. For most elastic-plastic constitutive models, the stress tensor σ_{ij} and the elastic strain tensor ϵ_{ij}^{el} are being calculated as simulation progresses. The challenging task is to evaluate the plastic free energy term ψ_{pl}, whose formulation depends on the internal variables used in the constitutive model.

For a decoupled elastic-plastic material model that exhibits both isotropic and kinematic hardening, the plastic free energy is decomposed into isotropic and kinematic parts, which are calculated individually and then added up. The formulation of plastic free energy in this type of material was given by Feigenbaum and Dafalias (2007):

$$\psi_{pl} = \psi_{pl}^{iso} + \psi_{pl}^{kin} = \frac{1}{2\rho\kappa_1}k^2 + \frac{1}{2\rho a_1}\alpha_{ij}\alpha_{ij} \tag{104.690}$$

where ψ_{pl}^{iso} and ψ_{pl}^{kin} are the isotropic and kinematic parts of the plastic free energy, respectively, k is the radius of yield surface, α is the back stress, κ_1 and a_1 are non-negative material constants. Note that Equation 104.690 can be used for a wide range of constitutive models with various yield functions, including von Mises and Drucker-Prager yield criteria whose energy behaviors has been studied and presented by Yang et al. (2018). Such materials are usually used to model soils and parts of the structure that need to be modeled with solid elements in SSI system.

However, frame structures are generally modeled with beam-column elements in combination with fiber sections and uniaxial material models, where Equation 104.690 does not apply. It should be realized that most uniaxial constitutive relationships are capable of modeling the stress–strain behavior but not the energy dissipation features of the material. Therefore, an approach that follows our thermomechanical framework is presented to correctly evaluate energy storage and dissipation in these material models.

104.12.2.3 Energy Dissipation in Beam-Column Element

Beams and columns are modeled with nonlinear displacement-based beam element, which is implemented in the Real-ESSI Simulator. In order to incorporate confined/unconfined concrete and steel reinforcement bars into beam elements, fiber sections are constructed with corresponding uniaxial fibers. A bottom-fixed reinforced concrete column model is shown in Figure 104.92, along with the constitutive relationships used for the concrete and steel fibers.

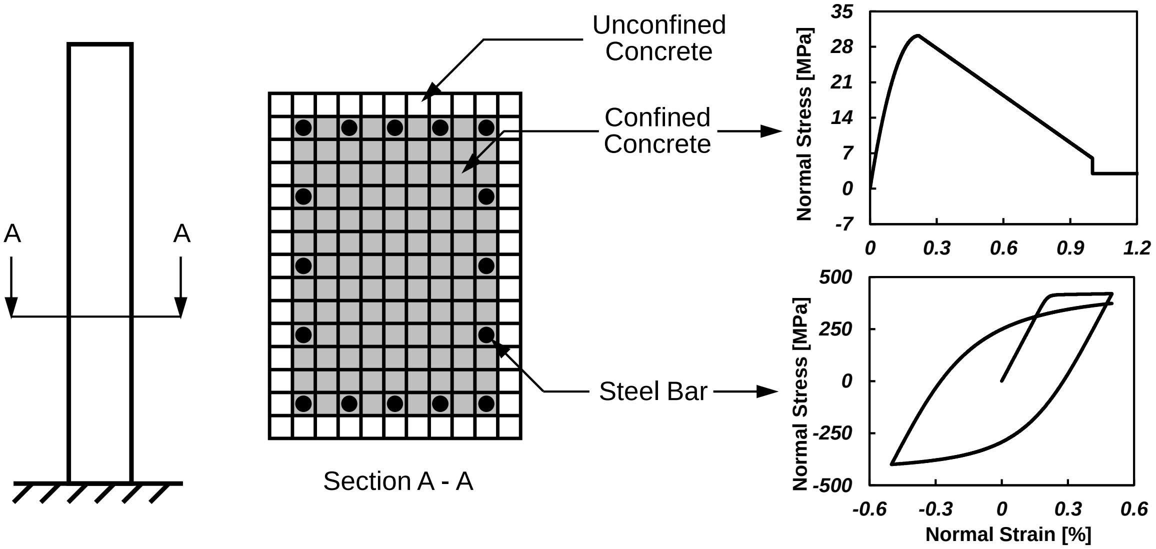


Figure 104.92: Schematic of a bottom-fixed column modeled with concrete and steel fibers.

Uniaxial Steel Fiber. The uniaxial steel material model examined in this study was developed by Menegotto and Pinto (1973) and extended by Filippou et al. (1983), and is capable of capturing the nonlinear hysteretic behavior and isotropic strain-hardening effect of steel. The stress–strain response of this material is shown in Figure 104.93, along with some of the material parameters. The model, as presented in Menegotto and Pinto (1973), takes on the form:

$$\sigma^* = b\epsilon^* + \frac{(1-b)\epsilon^*}{(1+\epsilon^{*R})^{1/R}} \tag{104.691}$$

with

$$\epsilon^* = \frac{\epsilon - \epsilon_r}{\epsilon_0 - \epsilon_r}; \quad \sigma^* = \frac{\sigma - \sigma_r}{\sigma_0 - \sigma_r} \tag{104.692}$$

where b is the strain-hardening ratio, ϵ_r and σ_r are the strain and stress at the point of strain reversal, ϵ_0 and σ_0 are the strain and stress at the point of intersection of the two asymptotes, R is the curvature parameter that governs the shape of the transition curve between the two asymptotes. Note that this model is for uniaxial materials, in which the stresses and strains are scalars instead of tensors.

The expression for the curvature parameter R is suggested by Menegotto and Pinto (1973):

$$R = R_0 - \frac{c_{R_1}\xi}{c_{R_2} + \xi} \tag{104.693}$$

where R_0 is the value of the curvature parameter R during initial loading, c_{R_1} and c_{R_2} are degradation parameters that need to be experimentally determined. The parameter ξ, that is updated after strain

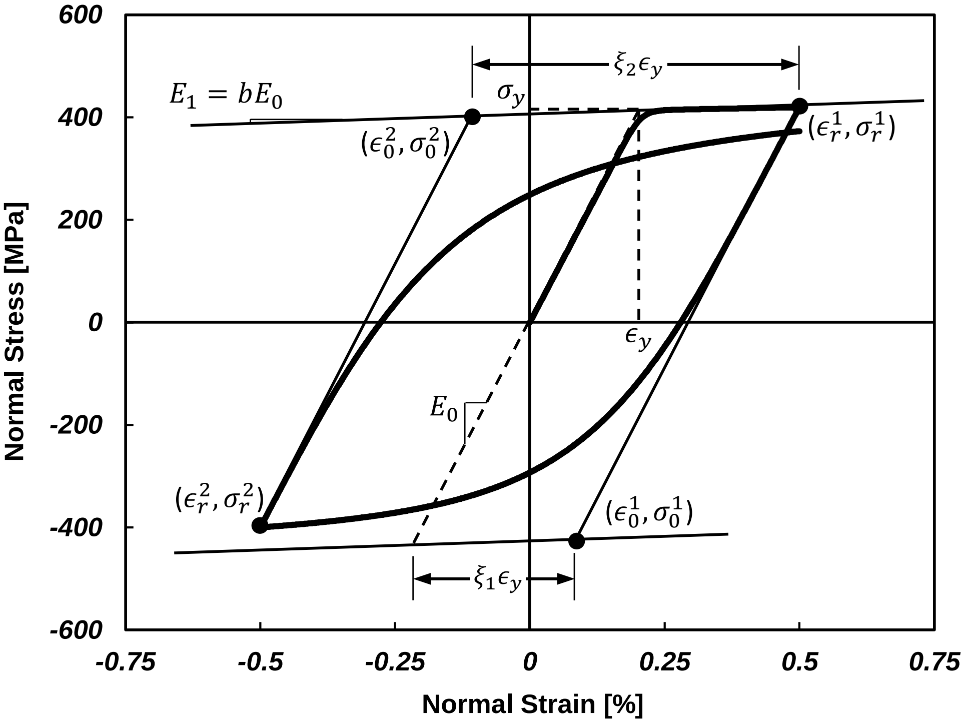


Figure 104.93: Constitutive model for uniaxial steel fiber (Menegotto and Pinto (1973)).

reversal, is defined as:

$$\xi = \left| \frac{(\epsilon_m - \epsilon_0)}{\epsilon_y} \right| \tag{104.694}$$

where ϵ_m is the maximum (or minimum) strain at the previous strain reversal point, depending on the loading direction of the material. If the current incremental strain is positive, the parameter ϵ_m takes the value of the maximum reversal strain. Parameter ϵ_y is the monotonic yield strain.

In order to capture isotropic hardening behavior, Filippou et al. (1983) introduced stress shift mechanism into the original model by Menegotto and Pinto (1973). Note that the hardening rate in compression and tension can be different by choosing different hardening parameters for compression and tension. The proposed relation takes the form:

$$\frac{\sigma_{st}}{\sigma_y} = a_1 \left(\frac{\epsilon_{max}}{\epsilon_y} - a_2 \right) \tag{104.695}$$

where σ_{st} is the shift stress that determines the shift of yield asymptote, ϵ_{max} is the absolute maximum strain at strain reversal, and a_1 and a_2 are hardening parameters in compression that are experimentally determined. In the case of tension, the hardening parameters a_1 and a_2 in Equation 104.695 are changed to a_3 and a_4 that are also determined by experiment or obtained from the literature.

The energy computation procedure for this uniaxial steel model is shown in Figure 507.4, and it follows the thermomechanical framework established earlier in this section.

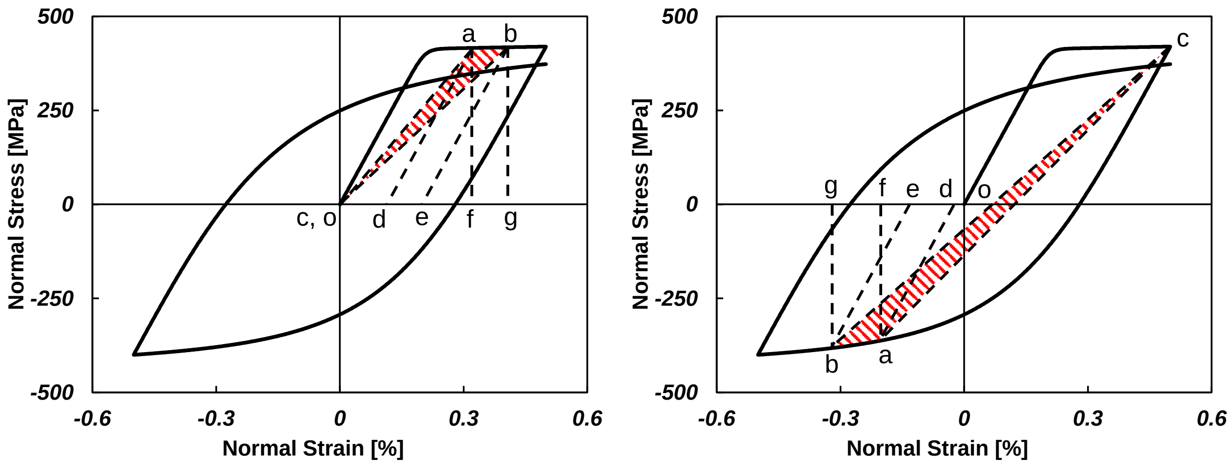


Figure 104.94: Energy computation of uniaxial steel fiber: (a) Monotonic loading branch; (b) Cyclic loading branch.

Note that the only difference between the monotonic loading branch (Figure 507.4(a)) and the cyclic loading branch (Figure 507.4(b)) is that the strain reversal point c is at the origin o in the monotonic case. So the following explanation of the proposed energy computation method applies to both monotonic and cyclic loading scenarios.

Firstly, the elastic strain energy density E_S is defined in accordance with the classic assumption that it is only a function of current stress state of the material, which yields:

$$E_S = E_S(\sigma) = \frac{1}{2E_0}\sigma^2 \tag{104.696}$$

Graphically, the elastic strain energy density of the material shown in Figure 507.4 at states a and b are the triangular areas afd and bge. Then the incremental form of Equation 507.9 is simply:

$$dE_S = \frac{1}{E_0}\sigma d\sigma \tag{104.697}$$

Next, the incremental plastic dissipation density D_P from state a to b is assumed to be the triangular area abc:

$$dD_P = \frac{1}{2}[(\sigma - \sigma_r)d\epsilon - (\epsilon - \epsilon_r)d\sigma] \tag{104.698}$$

This assumption ensures that the incremental plastic dissipation is nonnegative, which is one of the two basic principles of our thermomechanical framework. One special case is when the material exhibits no cyclic softening, which means that perfectly overlapping stress–strain loops will be observed, the energy dissipation calculated using Equation 507.11 for one cyclic is the area of the hysteresis loop. In the proposed thermomechanical framework, the area of hysteresis loop should be equal to the plastic work, rather than plastic dissipation, done in one loading cycle. But in this special case of no cyclic softening, which means no evolution of material state and thus no development of plastic free energy after a complete loading cycle, the plastic work equals to the plastic dissipation in the material in one loading cycle.

For general case where the material does exhibit cyclic softening, plastic free energy density E_P is graphically denoted by the areas $adoc$ and $beoc$ at states a and b, respectively. The formulation representing this assumption is given by:

$$E_P = \frac{1}{2}\left[\sigma\left(\epsilon - \frac{\sigma}{E_0} - \epsilon_r\right) + \sigma_r \epsilon\right] \tag{104.699}$$

The incremental form of Equation 507.12 is:

$$dE_P = \frac{1}{2}\left[(\sigma + \sigma_r)\, d\epsilon + \left(\epsilon - \frac{1}{E_0}\sigma - \epsilon_r\right) d\sigma\right] \tag{104.700}$$

Adding Equation 507.10, 507.11, and 507.13, the incremental form of energy balance is achieved:

$$dE_S + dE_P + dD_P = \sigma d\epsilon \tag{104.701}$$

where the increment of three energy components add up to the increment of stress power during any loading step.

Uniaxial Concrete Fiber. The uniaxial concrete material model used in this study is based on the model proposed by Yassin (1994), which is capable of modeling the nonlinear hysteretic behavior and damage effect of concrete. The material parameters and stress–strain response of this material are shown in Figure 104.95.

The monotonic envelope curve of this model in compression is based on the model of Kent and Park (1971) and later generalized by Scott et al. (1982). For a given strain ϵ_c, the compressive stress σ_c and corresponding tangent stiffness E are given by:

$$\epsilon_c \leq \epsilon_{cs} \qquad \sigma_c = f_{cs}\left[2\left(\frac{\epsilon_c}{\epsilon_{cs}}\right) - \left(\frac{\epsilon_c}{\epsilon_{cs}}\right)^2\right] \qquad E = E_c\left(1 - \frac{\epsilon_c}{\epsilon_{cs}}\right) \tag{104.702}$$

$$\epsilon_{cs} < \epsilon_c \leq \epsilon_{cu} \qquad \sigma_c = \frac{\epsilon_c - \epsilon_{cs}}{\epsilon_{cu} - \epsilon_{cs}}(f_{cu} - f_{cs}) + f_{cs} \qquad E = \frac{f_{cu} - f_{cs}}{\epsilon_{cu} - \epsilon_{cs}} \tag{104.703}$$

$$\epsilon_c > \epsilon_{cu} \qquad \sigma_c = f_{cu} \qquad E = 0 \tag{104.704}$$

where f_{cs} is the maximum compressive strength of the concrete material, ϵ_{cs} is the concrete strain at compressive strength, f_{cu} is the ultimate (crushing) strength of the concrete material, ϵ_{cu} is the concrete

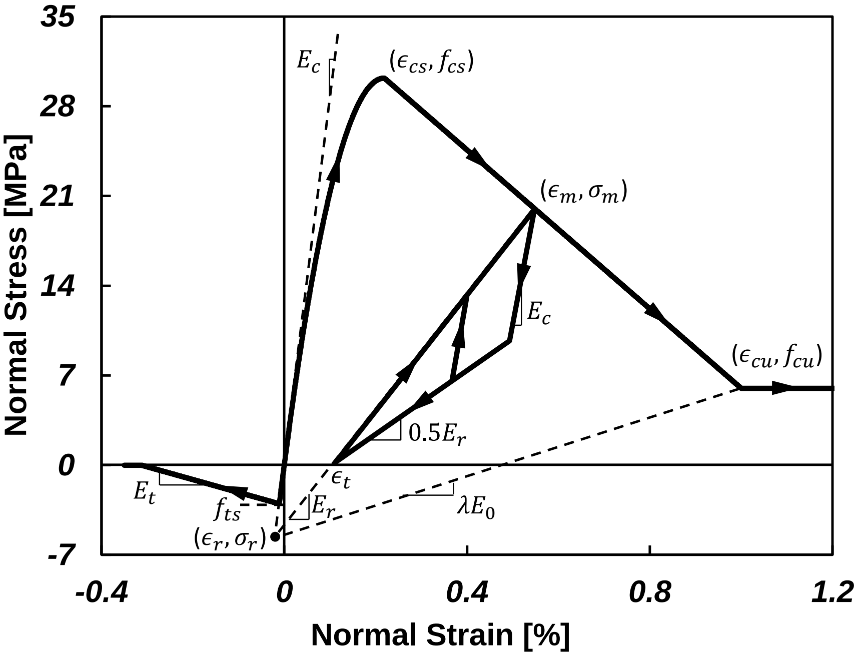


Figure 104.95: Constitutive model for uniaxial concrete fiber (Yassin (1994)).

strain at ultimate strength, and E_c is the initial concrete tangent stiffness that can be calculated using the equation:

$$E_c = \frac{2f_{cs}}{\epsilon_{cs}} \tag{104.705}$$

All material parameters should be determined by experiment or related literature data.

The cyclic behavior of this concrete model in compression is shown in Figure 104.95. One assumption of this model is that all reloading lines intersect at a common point, where the stress σ_r and strain ϵ_r are given by the following expressions:

$$\epsilon_r = \frac{f_{cu} - \lambda E_c \epsilon_{cu}}{E_c(1-\lambda)} \tag{104.706}$$

$$\sigma_r = E_c \epsilon_r \tag{104.707}$$

After unloading from a point on the compressive monotonic envelope, the model response is bounded by two lines that are defined by:

$$\sigma_{max} = \sigma_m + E_r(\epsilon_c - \epsilon_m) \tag{104.708}$$

$$\sigma_{min} = 0.5E_r(\epsilon_c - \epsilon_t) \tag{104.709}$$

where

$$E_r = \frac{\sigma_m - \sigma_r}{\epsilon_m - \epsilon_r} \tag{104.710}$$

$$\epsilon_t = \epsilon_m - \frac{\sigma_m}{E_r} \tag{104.711}$$

where σ_m and ϵ_m are the stress and strain at the unloading point on the compressive monotonic envelope, respectively. If the unloading–reloading cycle is incomplete, the material response will be a straight line with slope E_c, as shown in Figure 104.95.

The tensile behavior of this concrete model considers tension stiffening and the effects of initial cracking. Details of the monotonic and cyclic behavior of this model under tensile stress can be found in Yassin (1994).

Since there are different loading/unloading branches in this model, the energy computation needs to be considered separately for each branch. One energy component that remains the same in all loading cases is the elastic strain energy density E_S, which is only a function of current stress:

$$E_S = E_S(\sigma) = \frac{1}{2E_c}\sigma^2 \tag{104.712}$$

And the incremental form of Equation 104.712 is:

$$dE_S = \frac{1}{E_c}\sigma d\sigma \tag{104.713}$$

In order to calculate plastic dissipation, a few assumptions are made that ensures the energy behavior of the concrete material to follow the proposed thermomechanical framework:

- Majority of energy is dissipated during first monotonic load.
- Subsequent cycles of loading, on an already damaged concrete, do not dissipate much energy.
- No energy is dissipated during unloading in both compressive and tensile conditions.
- When the material is cyclically loaded under compression, energy dissipation only happens when the stress reaches the upper bound σ_{max}.
- No energy is dissipated during cyclic loading when the material is under tension.

For a single loading step from stress state a to b in each subplot of Figure 104.96, the energy dissipation is represented by the shaded area.

If the material is under compression (Figure 104.96 (a), (b), and (c)), the amount of energy dissipated in the concrete fiber D_P is calculated by taking the area $abcdef$, which is generally a hexagon formed by the two unloading paths originated from stress state a and b:

$$dD_P = \frac{1}{2}[(\sigma - \sigma_c)d\epsilon + (\epsilon_c - \epsilon)d\sigma + (\epsilon_c - \epsilon_f)\sigma + (\sigma_f - \sigma_c)(\epsilon - \epsilon_t) + \sigma_c d\epsilon_t] \tag{104.714}$$

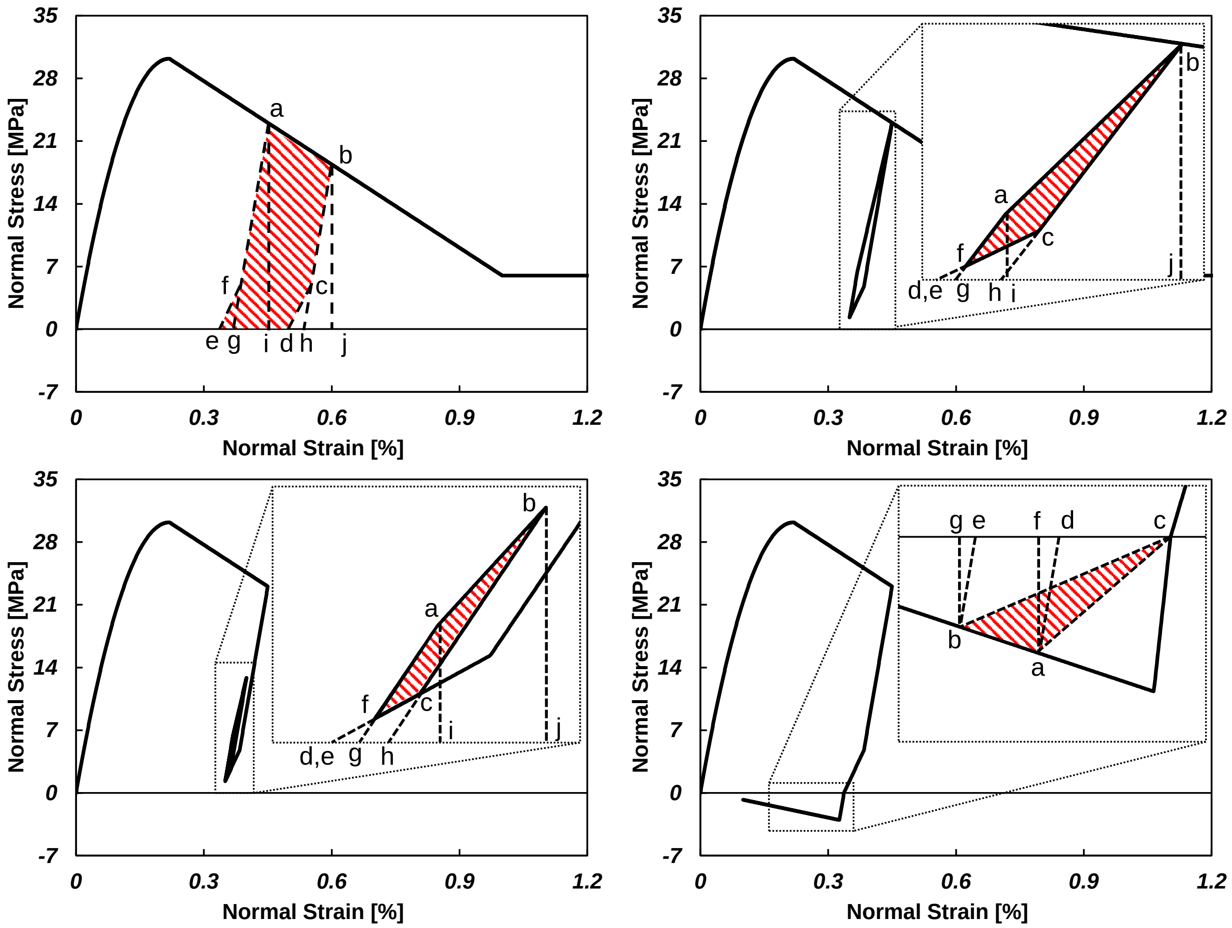


Figure 104.96: Energy computation of uniaxial concrete fiber: (a) Monotonic compression; (b) Single unloading-reloading cycle; (c) Unloading-reloading cycles within monotonic envelope; (d) Tension.

where the stress and strain at point f can be computed using the following expression:

$$\epsilon_f = \frac{\sigma + 0.5 E_r \epsilon_t - E_c \epsilon}{0.5 E_r - E_c} \qquad \sigma_f = 0.5 E_r (\epsilon_f - \epsilon_t) \tag{104.715}$$

Point c can be calculated using the same fashion, but with all variables evaluated at state b.

Note that the hexagon becomes quadrilateral in the cases of cyclic loading within the monotonic envelope, as can be observed in Figure 104.96 (b) and (c). But Equation 104.714 and 104.715 remains valid, obviously.

Plastic free energy E_P in this concrete material is calculated by taking the triangular area fge at state a:

$$E_P = \frac{1}{2}\left[\left(\epsilon - \frac{\sigma}{E_c} - \epsilon_t\right)\sigma_f\right] \tag{104.716}$$

The incremental form of Equation 104.716 is obtained by taking the difference between the plastic free energy at state a and b:

$$dE_P = \frac{1}{2}\left[\left(\sigma_c - \sigma_f - \frac{1}{E_c}\sigma\right)(\epsilon - \epsilon_t) - (d\epsilon - d\epsilon_t)\sigma_c - \frac{1}{E_c}\sigma c d\sigma\right] \tag{104.717}$$

Adding Equation 104.713, 104.714, and 104.717, the incremental form of energy balance is achieved:

$$dE_S + dE_P + dD_P = \sigma d\epsilon \tag{104.718}$$

where the increment of three energy components add up to the increment of stress power during any loading step.

104.12.3 Numerical Studies

Numerical examples presented in this section are performed using the Real-ESSI Simulator Jeremić et al. (1988-2025). Energy dissipation in numerical models consist of fiber section elements and uniaxial steel/concrete fibers are computed and analyzed.

First, numerical simulation of steel and plain concrete columns under various loading conditions are performed to study the energy behavior of uniaxial steel and concrete material models. Then, a model of reinforced concrete column, which consists of both concrete and steel fibers, is constructed and simulated to illustrate the energy dissipation in realistic structural elements. Finally, a bare steel frame structure is modeled with fiber section elements and loaded with seismic motion. Through these examples, it will be shown that the difference between plastic work and plastic energy dissipation can be significant.

External loads are applied incrementally using displacement-control scheme. System equations are solved using Newton-Raphson iteration algorithm and UMFPACK solver, which are available in Real-ESSI. Static integration algorithm is used for the column cases, while Newmark integration is used for the dynamic steel frame case. Note that viscous and numerical damping are excluded from all cases, in order to accurately evaluate energy dissipation due to material elastoplasticity.

104.12.3.1 Steel Column

In order to verify the proposed energy computation approach for uniaxial steel material model, examples of steel columns are studied in this section. As shown in Figure 104.97, the one-meter-long column model is fixed at the bottom, and loads are applied at the top. The size of the cross section is 100 $mm \times$ 100 mm. The parameters for uniaxial steel material used in this section are summarized in Table 104.6.

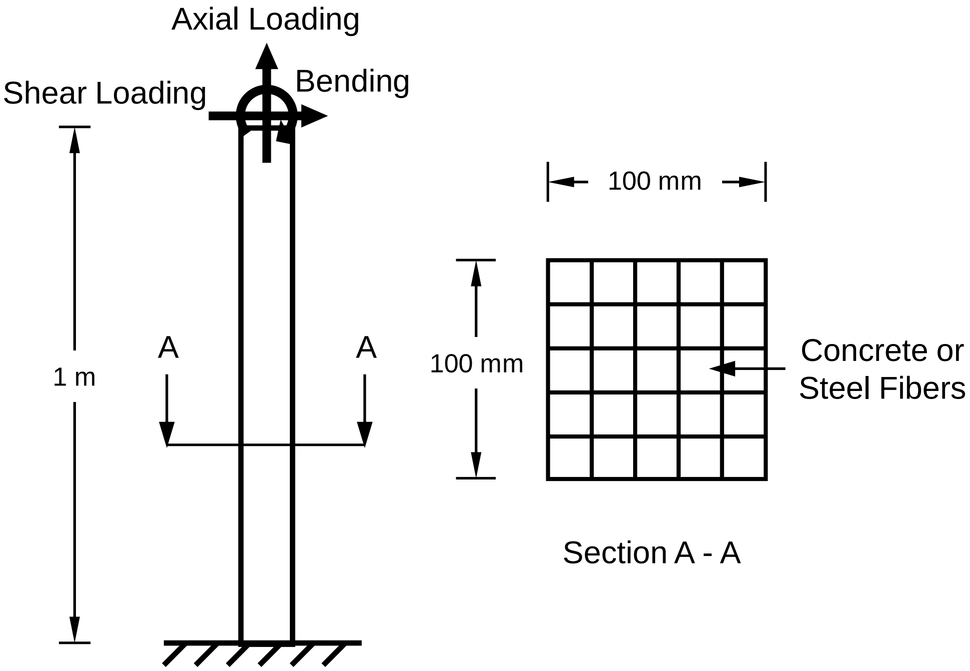


Figure 104.97: Schematic of the steel/plain-concrete column modeled with fiber sections and uniaxial steel/concrete materials.

Table 104.6: Material model parameters used in steel column examples.

σ_y (MPa)	E (GPa)	b	R_0	c_{R_1}	c_{R_2}	a_1	a_2	a_3	a_4
413.8	200.0	0.01	18.0	0.925	0.15	0.0	55.0	0.0	55.0

Cyclic Axial Loading. Since the fiber material model is uniaxial in nature, axial loading case is being investigated first. The evolution of energy parameters for uniaxial steel material are computed using Equation 507.10, 507.11, and 507.13. Figure 104.98 shows the stress–strain response as well as the energy results of the steel column under cyclic axial loading.

As expected, the stress–strain response shown in Figure 104.98 follows the constitutive model presented in Figure 104.93. Due to the choice of hardening parameters (a_1, a_2, a_3, and a_4), isotropic hardening after first loading reversal is relatively small in this case. The evolution of plastic free energy, which is related to the hardening behavior of the constitutive model, is also observed to be insignificant after the first loading reversal. Energy balance in the steel material (Equation 507.14) is maintained during entire simulation.

In this particular case, the difference between plastic dissipation and plastic work is significant during initial loading (or monotonic loading), but then becomes less obvious during cyclic loading, which is

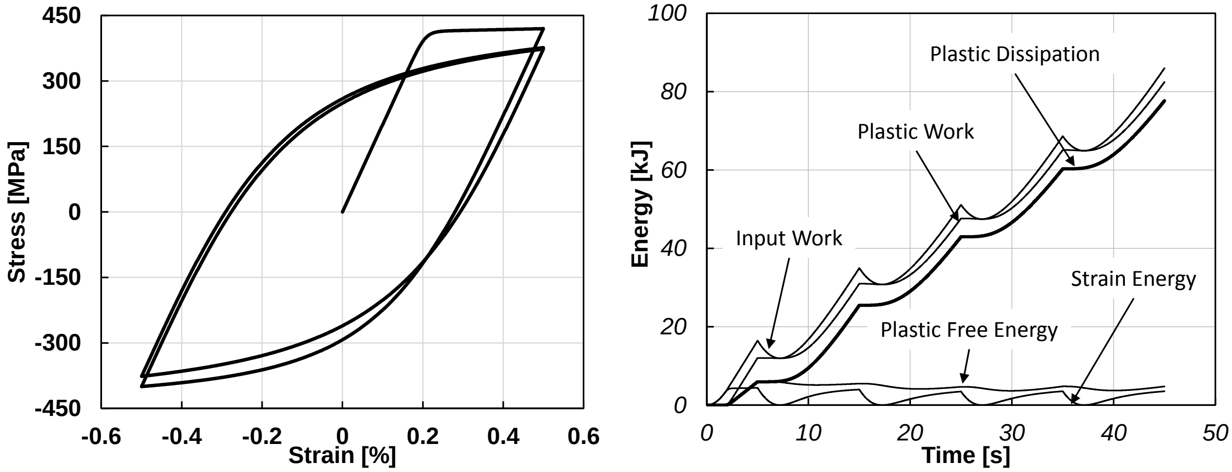


Figure 104.98: Energy analysis of steel column under cyclic axial loading: (a) Stress–strain response; (b) Plastic dissipation, plastic work, plastic free energy, strain energy, and input work.

probably the reason of ignorance of plastic free energy in many studies. It is important to point out that such difference could be significant if different hardening parameters are chosen or complex loading conditions (like seismic loading) are applied.

Another observation made in this example is that the ratio between plastic dissipation and plastic work (the Quinney–Taylor coefficient) changes from 0.5 to 0.9 in just a few loading cycles. Therefore, it is not accurate to prescribe a fixed number to be the Quinney–Taylor coefficient of a material during entire simulation, which is a common assumption made in a number of studies.

Cyclic Bending Loading. It has been proven that fiber section elements perform well under axial- and bending-dominant loading conditions. In this case, a cyclic bending moment is loaded on the top of the steel column. Figure 104.99 shows the moment–rotation response as well as the energy results of the steel column under cyclic bending loading.

Clearly, the moment–rotation response and energy results in this case are very similar to those in the axial loading case. When a beam element is under bending, half of the fibers will be under tension while the other half under compression, and the normal stress distribution on any cross section should be symmetric. Since the fiber material model used in this case has almost the same stress–strain response under tension and compression, the energy results in this bending case are expected to share the same pattern with those in the axial loading case.

Note that in both axial and bending cases, the strain energy accumulated in the material body is much smaller than the plastic dissipation. This means that most of the input work results in plastic

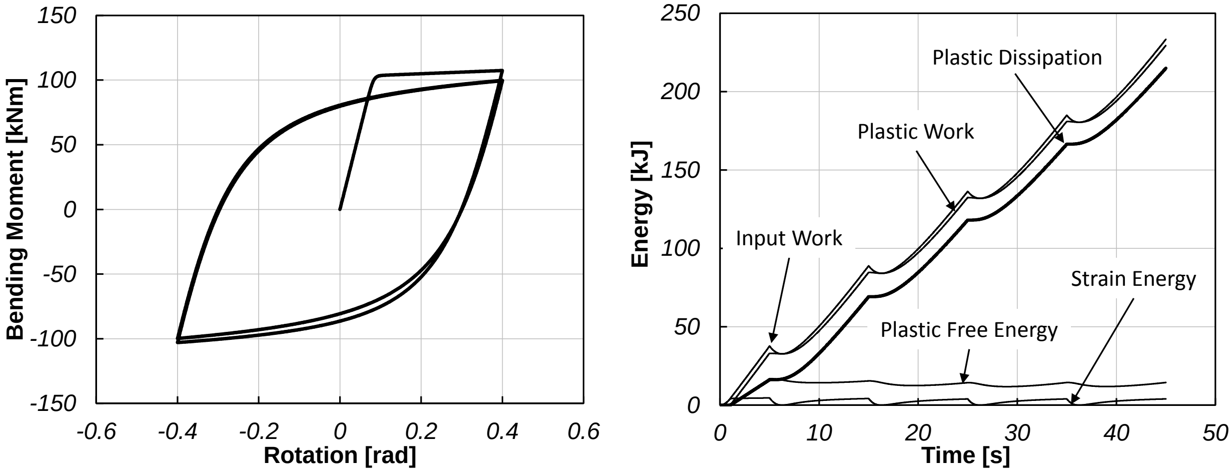


Figure 104.99: Energy analysis of steel column under cyclic bending loading: (a) Moment–rotation response; (b) Plastic dissipation, plastic work, plastic free energy, strain energy, and input work.

deformation of the material, which indicates high possibility of large deformation and material damage.

104.12.3.2 Plain Concrete Column

In order to verify the proposed energy computation approach for uniaxial concrete material model, examples of plain concrete columns are studied in this section. The size and setup of the model are the same as those of the steel column, which has been shown in Figure 104.97. The parameters for uniaxial concrete material used in this section are summarized in Table 104.7.

Table 104.7: Material model parameters used in plain concrete column examples.

f_{cs} (MPa)	ϵ_{cs}	f_{cu} (MPa)	ϵ_{cu}	λ	f_{ts} (MPa)	E_t (GPa)
-30.2	-0.00219	-6.0	-0.00696	0.5	3.02	5.0

Monotonic Axial Loading. As stated in the assumptions for energy dissipation in the uniaxial concrete model, the amount of energy dissipated during monotonic loading is much larger than that during unloading/reloading. Such assumption is made based on the brittle nature of concrete materials, in which fracture is the main source of energy dissipation. In this case, the stress–strain response as well as the energy results of the plain concrete column model under monotonic axial compression is investigated and presented in Figure 104.100.

The stress–strain response shown in Figure 104.100 follows the constitutive model presented in

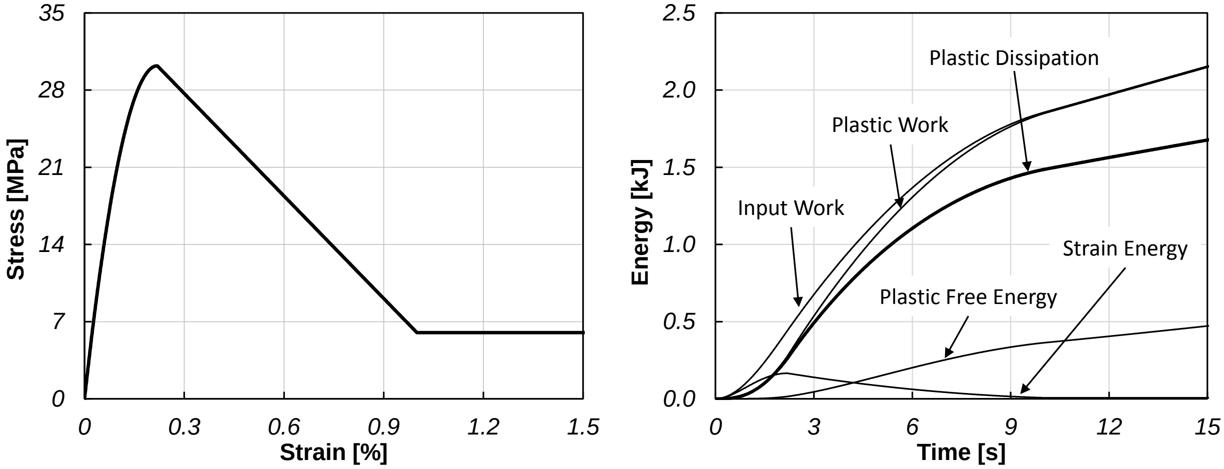


Figure 104.100: Energy analysis of plain concrete column under monotonic axial loading: (a) Stress–strain response; (b) Plastic dissipation, plastic work, plastic free energy, strain energy, and input work.

Figure 104.95, as expected. Energy balance in the model, which is expressed in Equation 507.8, is maintained during entire simulation.

As observed in Figure 104.100, large amount of the input work is dissipated during monotonic compression. It is important to point out that the difference between plastic dissipation and plastic work is significant. Plastic free energy starts to accumulate after maximum compressive strength is reached and continue to increase even after crushing. Such behavior can be explained by considering that the micro-structure of concrete continues to evolve as external loads continues to be applied on the material.

The strain energy starts to drop after maximum compressive strength and gradually decreases to almost zero after crushing. This observation is consistent with the fact that the micro-fractures expand rapidly after maximum strength is reached, which leads to the release of elastic strain energy and energy dissipation caused by fracture and crushing.

Cyclic Axial Loading. Due to the complex unloading–reloading rules of the model, the cyclic behavior of the uniaxial concrete material is much more complicated than that of the steel model. Figure 104.101 presents the stress–strain response as well as the energy results of the plain concrete column under cyclic axial loading.

As shown in Figure 104.101, the majority of plastic dissipation happens during monotonic loading branch. Notice that there are drops in plastic work during unloading, but plastic dissipation never decreases, which means that the second law of thermodynamics (Equation 504.8) is always obeyed.

It should be mentioned that there are certain amount of energy dissipation when the material is in

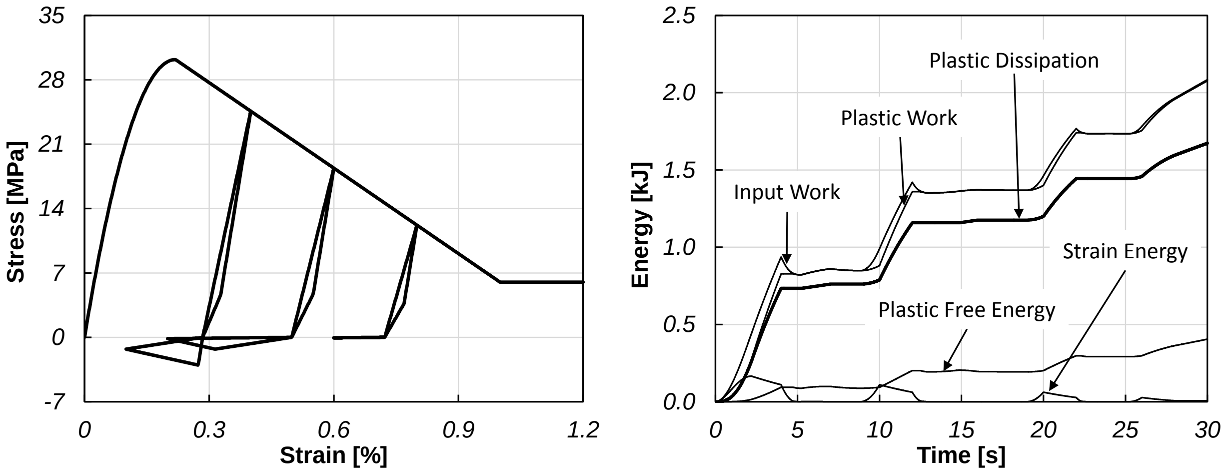


Figure 104.101: Energy analysis of plain concrete column under cyclic axial loading: (a) Stress–strain response; (b) Plastic dissipation, plastic work, plastic free energy, strain energy, and input work.

tension. But it is much smaller than that when the material is in monotonic compression, due to the low tensile strength of concrete material in general.

104.12.3.3 Reinforced Concrete Column

To study the combined influence of concrete and steel fibers, a reinforced concrete column is modeled and tested in this section. The schematic of the model is shown in Figure 104.101, and the material model parameters are summarized in Table 104.8. The cross section of the column is modeled with unconfined concrete, confined concrete, and steel fibers with uniaxial material models discussed in earlier sections.

Cyclic Axial Loading. Figure 104.103 shows the force–displacement response as well as the energy results of the reinforced concrete column under cyclic axial loading.

Since concrete fibers have much higher compressive strength than tensile strength, the stress–strain response of the column is controlled by the concrete part when it is under compression, and by the steel part when under tension. In this case, the initial loading curve clearly resembles the stress–strain response of concrete fiber under monotonic compression. Then the unloading–reloading cycles have the same pattern as those of the steel fiber under cyclic axial loading.

By comparing the energy results shown in Figure 104.103 and those shown in Figure 104.98, it can be seen that the energy dissipation patterns in both cases are very close. This indicates that the majority of input work is dissipated in the steel fibers once the maximum strength of the concrete is exceeded. Again, it can be observed that the difference between plastic work and plastic dissipation is significant

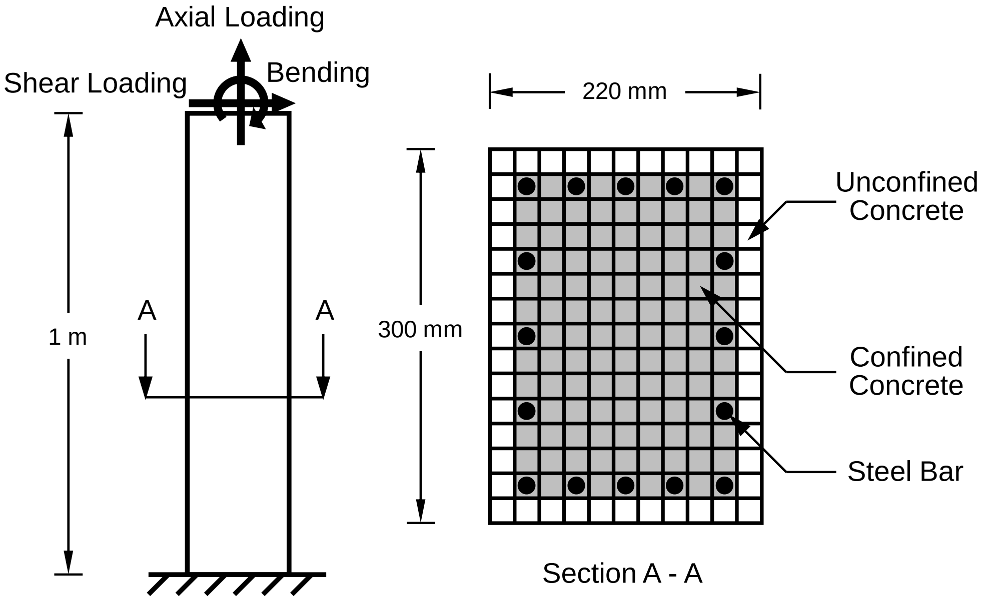


Figure 104.102: Schematic of the reinforced concrete column modeled with fiber sections and uniaxial steel/concrete materials.

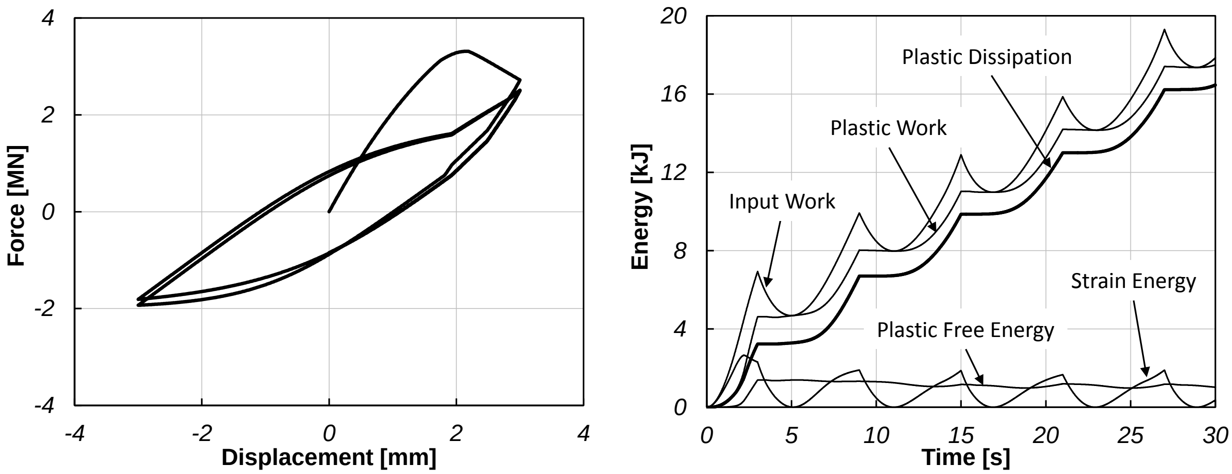


Figure 104.103: Energy analysis of reinforced concrete column under cyclic axial loading: (a) Force–displacement response; (b) Plastic dissipation, plastic work, plastic free energy, strain energy, and input work.

Table 104.8: Material model parameters used in reinforced concrete column examples.

Steel Fiber		Concrete Fiber		
			Confined	Unconfined
σ_y (MPa)	413.8	f_{cs} (MPa)	-30.2	-24.16
E (GPa)	200.0	ϵ_{cs}	-0.00219	-0.001752
b	0.01	f_{cu} (MPa)	-6.0	0.0
R_0	18.0	ϵ_{cu}	-0.00696	-0.005568
c_{R_1}	0.925	λ	0.5	0.5
c_{R_2}	0.15	f_{ts} (MPa)	3.02	0.0
a_1, a_3	0.0	E_t (GPa)	5.0	0.0
a_2, a_4	55.0			

in this case.

Cyclic Bending Loading. Figure 104.104 shows the moment–rotation response as well as the energy results of the reinforced concrete column under cyclic bending loading.

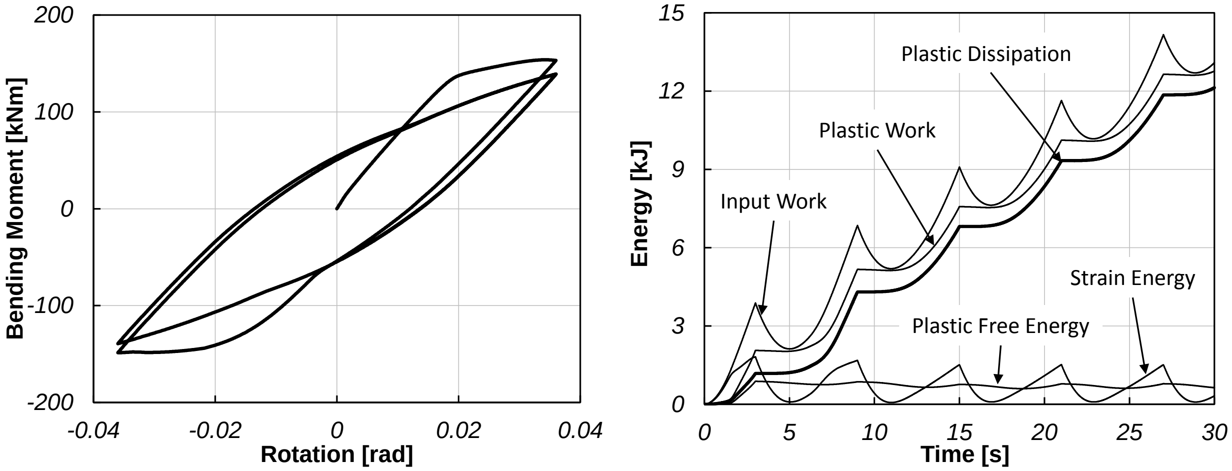


Figure 104.104: Energy analysis of reinforced concrete column under cyclic bending loading: (a) Moment–rotation response; (b) Plastic dissipation, plastic work, plastic free energy, strain energy, and input work.

During initial loading, the concrete fibers on the compressive side of the cross section take most of the compression, then during the first reverse loading, the concrete fibers on the other side of the cross section are compressed and damaged. This process is indicated in the moment–rotation curve where a bump caused by the compressive strength of the concrete fibers during the first reverse loading is observed. The energy computation result also shows that the concrete fibers dissipate large amount of energy and get damaged during the first loading cycle. After that, the response of the reinforced concrete column is controlled by the steel bars.

According to the two cases of reinforced concrete column under cyclic loading, the concrete part of the column can dissipate the majority of the input work if the loading is mainly monotonic compression. For cyclic loading cases, if the loading does not exceed the maximum compressive strength of the concrete, which should not be significantly damaged, energy dissipation would be observed in both the concrete and steel. However, if the cyclic loading does exceed the maximum strength of the concrete, the majority of energy dissipation would be in the steel reinforcing bars after the concrete is damaged. This conclusion is consistent with the engineering experience that reinforcements are crucial to the performance of concrete structure during seismic events, when the beams and columns suffer from cyclic loadings.

104.12.3.4 Steel Frame

All the previous cases are assumed to be static or quasi-static to investigate the energy dissipation on material level without the influence of dynamics. In other words, inertia and kinetic energy were not considered. In this example, a steel frame structure is model using fiber section element with uniaxial steel material, as shown in Figure 104.105, and loaded dynamically with a realistic seismic motion. The peak acceleration of the input motion is 0.76 g.

The energy computation results are shown in Figure 104.106. Strain energy, plastic free energy, and plastic dissipation in different stories are computed using Equation 507.10, 507.11, and 507.13. Input work is computed from the input motion and reaction forces at the base of model. Kinetic energy is computed indirectly by subtracting all other forms of energy from input work.

Note that the kinetic energy of the system is almost zero at certain times, and strain energy reaches a peak value. As can be observed from Figure 104.106, when kinetic energy becomes zero, the combination of strain energy, plastic free energy, and plastic dissipation of the system equals to the total input work. This observation proves that the energy balance of the system is maintained during entire simulation.

At the end of simulation, more than 80% of the total input work is dissipated due to material elasto-plasticity, while about 13% is transformed into plastic free energy which does not result in heating or material damage. In some cases, it might be reasonable to use input work (or energy demand in some literature) as a parameter to evaluate structure safety. However, as shown in this example, correctly computed energy dissipation is more appropriate for evaluation of material damage and structure performance in general.

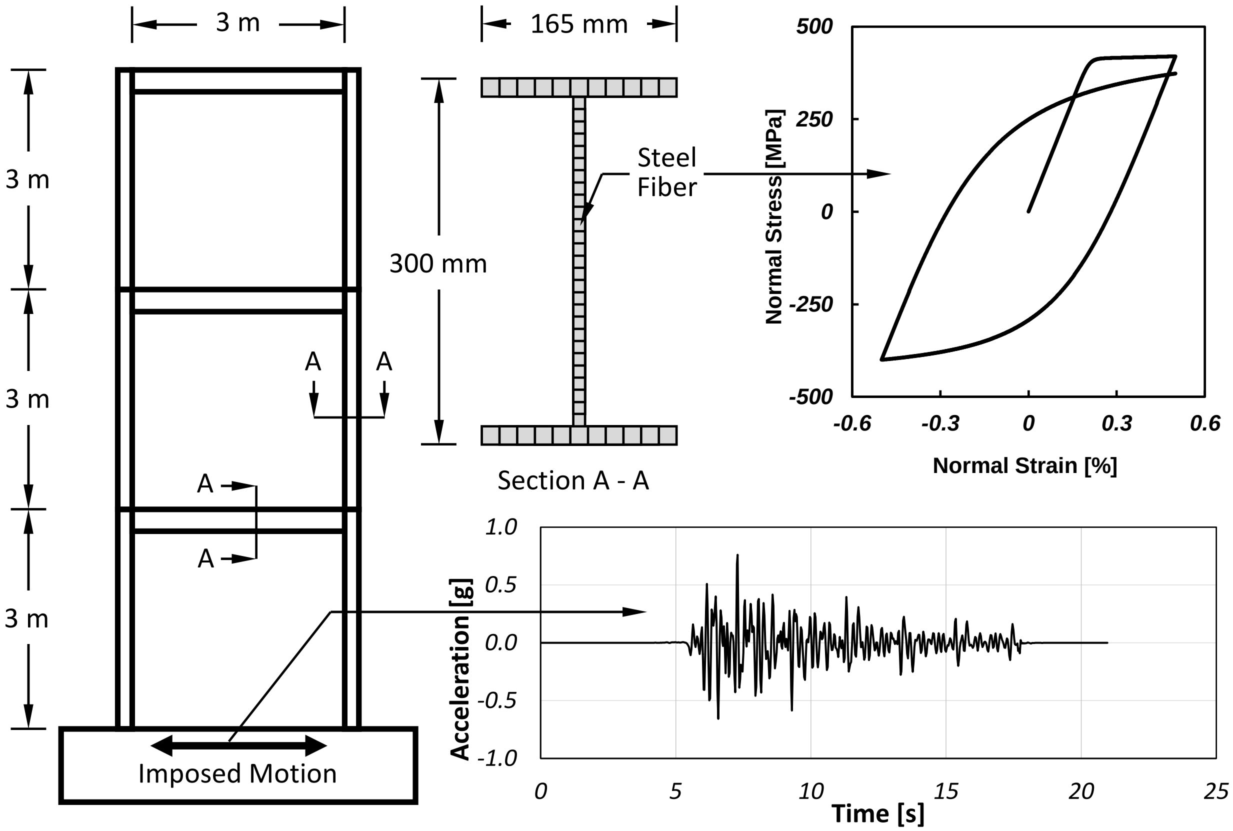


Figure 104.105: Schematic of the steel frame modeled with fiber section elements and uniaxial steel material.

104.12.4 Conclusions

Presented in this section was a thermodynamic-based methodology for (correct) computation of energy dissipation in nonlinear structural elements modeled with fiber section and uniaxial material models. Two popular material models for steel and concrete were examined with focus on their nonlinear cyclic behaviors. Formulations for the energy storage and dissipation in these two material models were derived from the basic principles of thermodynamics, in combination with a few reasonable assumptions. The proposed methodology has been illustrated using a series of numerical simulations on columns and frame modeled with fiber section elements.

The misconception between plastic work and plastic dissipation, which leads to the violation of principles of thermodynamics, was addressed. Theoretical derivation and experimental observation have both proven that plastic free energy is a basic form of energy that should not be neglected without proper reasoning. By taking into account of all possible energy forms, including kinetic energy, strain energy, plastic free energy, and plastic dissipation, the first law of thermodynamics (energy balance) was

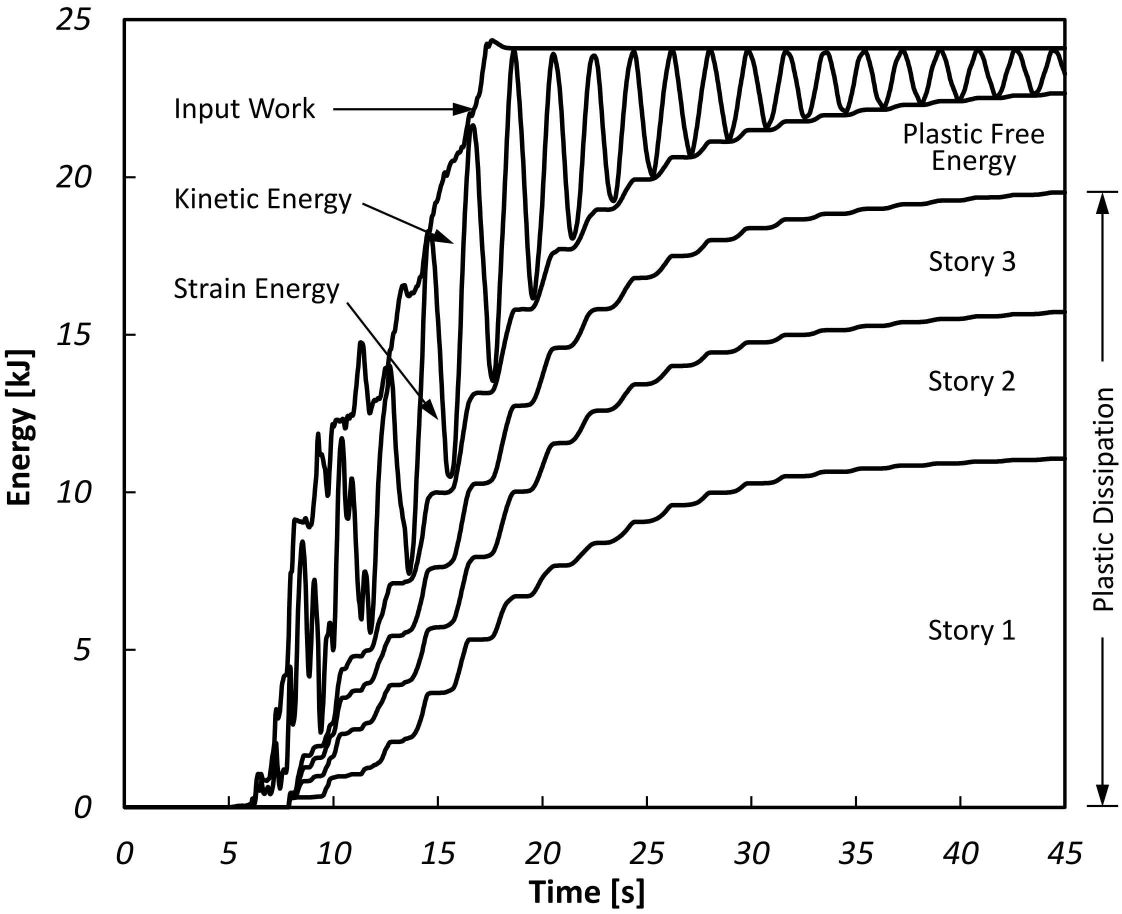


Figure 104.106: Energy analysis of steel frame model under imposed seismic motion.

ensured in the proposed methodology.

Physically, plastic free energy is related to the evolution of material micro-structure, which is not represented by specific parameters (like the internal variables in some elastoplasticity models). According to the experimental behavior of the material models, a few assumptions that ensures their energy behavior to follow the proposed thermomechanical framework were made. Equations for energy computation were derived and implemented in Real-ESSI, which was used to perform numerical simulations in this study.

Presented approach was illustrated and tested using several column and frame models with different loading conditions. As expected, energy balance was maintained during entire simulation in all tested cases. It was shown that plastic work could drop but plastic dissipation always maintained nonnegative during any time period, which is expressed in the second law of thermodynamics. It was also observed that the difference between plastic work and plastic dissipation could be significant in most cases. The ratio between them (Quinney–Taylor coefficient) evolved with time and thus should not be assumed to be constant in general.

104.13 Localization of Deformation

(Rudnicki and Rice, 1975), (Lu et al., 2009)

Chapter 105

Probabilistic Elasto-Plasticity and Stochastic Elastic-Plastic Finite Element Method

(2004-2006-2009-2014-2016-2017-2019-2021-)

(In collaboration with Prof. Kallol Sett, Dr. Hexiang Wang, and Dr. Katarzyna Staszewska)

105.1 Chapter Summary and Highlights

For more details on work in this area, please consult the following papers:

Jeremić et al. (2007a),

Sett et al. (2007a),

Sett et al. (2007b),

Jeremić and Sett (2007),

Jeremić and Sett (2006),

Sett and Jeremić (2007),

Jeremić and Sett (2009a),

Sett and Jeremić (2010),

Sett et al. (2011b),

Sett et al. (2011a),

Jeremić and Sett (2010).

Material from (some of) the above papers is presented below.

105.2 Probabailistic Elasto-Plasticity, 1D FPK Formulation

A second-order exact expression for evolution of Probability Density Function (PDF) of stress is derived for general, one dimensional (1-D) elastic-plastic constitutive rate equations with uncertain material parameters. The Eulerian–Lagrangian (EL) form of Fokker–Planck–Kolmogorov (FPK) equation is used for this purpose. It is also shown that by using EL form of FPK, the so called "closure problem" associated with regular perturbation methods used so far, is resolved too. The use of EL form of FPK also replaces repetitive and computationally expensive deterministic elastic-plastic computations associated with Monte Carlo technique.

The derived general expression are specialized to the particular cases of point location scale linear elastic and elastic–plastic constitutive equations, related to associated Drucker-Prager with linear hardening

In a companion paper, the solution of FPK equations for 1D is presented, discussed and illustrated through a number of examples.

105.2.1 Probabilistic Elasto-Plasticity: Introduction

Advanced elasto–plasticity constitutive models, when properly calibrated, are very accurate in capturing important aspects of material behavior. However, all materials', and in particular geomaterials' (soil, rock, concrete, powder, bone etc.) behavior is uncertain due to inherent spatial and point-wise uncertainties. These uncertainties in material properties could outweigh the advantages gained by using advanced constitutive models. For example, Fig. 105.1 shows a schematic of anticipated influence of material uncertainties on a bi-linear elastic-plastic stress-strain behavior. Depending on uncertainties in material properties and interaction between them, the behavior of the same material could be very different.

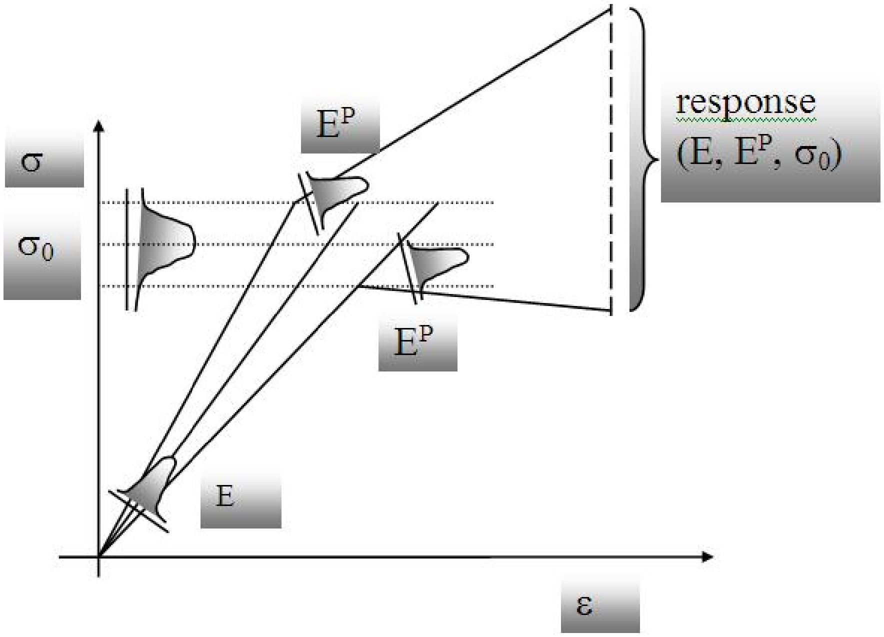


Figure 105.1: Anticipated Influence of Material Fluctuations on Stress-Strain Behavior

The uncertainties in material properties are inevitable in real materials and it is best to account for them in modeling and simulation. In traditional deterministic constitutive modeling, material models are calibrated against set of experimental data. Although those experimental data sets generally exhibit statistical distribution, the models are usually calibrated against the mean of the data and all the information about uncertainties is neglected.

The modeling and simulation of solids and structures with uncertain material properties involves two steps: (a) classification and quantification of uncertainties and (b) propagation of uncertainties through governing differential equations.

The uncertainties can be broadly classified into aleatory and epistemic types. Aleatory uncertainties are associated with the inherent variabilities of nature. This type of uncertainty can not be reduced. Highly developed mathematical theory is available for dealing with aleatory uncertainty. On the other hand, epistemic uncertainties arise due to our lack of knowledge. This type of uncertainty can be reduced by collecting more data but the mathematical tools to deal with them are not highly developed (e.g. fuzzy logic Zadeh (1983), convex models Ben-Haim and Elishakoff (1990), interval arithmetic Moore (1979) etc.). Hence, it proves useful to trade epistemic uncertainties for aleatory uncertainties in order to facilitate their propagation through the governing equations using advanced mathematical tools. It is important to note that in trading-off epistemic uncertainties for aleatory uncertainties, one doesn't reduce the total uncertainties in the system, but assumes that the uncertainties in the system are irreducible. Under the framework of probability theory, uncertain material parameters are modeled as random variables or random fields (Vanmarcke, 1983) depending on whether they are specialized to a fixed location in their continuum or a function of location in their continuum. We note recent works in quantifying the uncertainties in material (soil) properties for geotechnical engineering applications, Lumb (1966), Vanmarcke (1977), Mayerhoff (1993), DeGroot and Baecher (1993), Popescu (1995), Lacasse and Nadim (1996), Popescu et al. (1998), Phoon and Kulhawy (1999a,b), Fenton (1999a,b), Duncan (2000b), Rackwitz (2000), Marosi and Hiltunen (2004), and Stokoe II et al. (2004) The issue of uncertain material properties becomes very pronounced when one starts dealing with the boundary value problems with uncertain material properties (elastic or elastic–plastic).

In mechanics, the equilibrium equation,$A\sigma = \phi(t)$, together with the strain compatibility equation, $Bu = \epsilon$, and the constitutive equation, $\sigma = D\epsilon$, are sufficient[1] to describe the behavior of the solid. Rigorous mathematical theory has been developed for problems where the only random parameter is the external force $\phi(t)$. In this case, the probability distribution function (PDF) of the response variable will satisfy FPK partial differential equation (Soize, 1994) . With appropriate initial and boundary conditions the FPK PDE can be solved for PDF of response variable. The numerical solution method for FPK equation by finite element method (FEM) is described by number of researchers e.g. Langtangen (1991), Masud and Bergman (2005).

The other extreme case, which is of interest in this work, is when the stochasticity of the system is purely due to operator uncertainty. Exact solution of the problems with stochastic operator was attempted by Hopf (1952) using characteristic functional approach. Later, Lee (1974) applied the methodology to the problem of wave propagation in random media and derived a FPK equation satisfied by the characteristic functional of the random wave field. This characteristic functional approach is very complicated for linear problems and becomes even more intractable (and possibly unsolvable) for nonlinear problems and problems with irregular geometries and boundary conditions.

Monte Carlo simulation technique is an alternative to analytical solution of partial differential equation with stochastic coefficient. Nice descriptions of different aspects of formulation of Monte Carlo technique for stochastic mechanics problem is described by Schüeller (1997). Monte Carlo method is very popular

---

[1] Generalized stress is σ, $\phi(t)$ is generalized forces that can be time dependent, u is generalized displacements, ϵ is generalized strain and A, B, and D are operators which could be linear or non-linear.

tool with the advantage that accurate solution can be obtained for any problem whose deterministic solution (either analytical or numerical) is known. Monte Carlo technique has been used by a number of researchers in obtaining probabilistic solution of geotechnical boundary value problems, e.g. Paice et al. (1996); Griffiths et al. (2002); Fenton and Griffiths (2003, 2005). Popescu et al. (1997), Mellah et al. (2000), De Lima et al. (2001), Koutsourelakis et al. (2002), Nobahar (2003). The major disadvantage of Monte Carlo analysis is the repetitive use of the deterministic model until the solution variable become statistically significant. The computational cost associated with it could be very high especially for non–linear problems with multiple uncertain material properties.

Various difficulties in finding analytical solutions and the high computational cost associated with Monte Carlo technique instigated development of numerical method for the solution of stochastic differential equation with random coefficient. For stochastic boundary value problems Stochastic Finite Element Method (SFEM) is the most popular such method. There exist several formulations of SFEM, among which perturbation (Kleiber and Hien (1992); Der Kiureghian and Ke (1988); Mellah et al. (2000); Gutierrez and De Borst (1999)) and Spectral (Ghanem and Spanos (1991); Keese and Matthies (2002); Xiu and Karniadakis (2003); Debusschere et al. (2003); Anders and Hori (2000)) methods are very popular. A nice review on advantages and disadvantages of different formulations of SFEM was provided by Matthies et al. (1997). Mathematical issues regarding different formulations of SFEM was addressed by Deb et al. (2001) and Babuska and Chatzipantelidis (2002). It is important to note that most of the formulations described in the above mentioned references are for linear elastic problems.

A limited number of references is also available related to geometric non–linear problems, Liu and Der Kiureghian (1991); Keese and Matthies (2002) and Keese (2003)). Similarly, there exist only few published references related to material non–linear (elastic–plastic) problems with uncertain material parameters. The major difficulty in extending the available formulations of SFEM to general elastic–plastic problem is the high non–linear coupling in the elastic–plastic constitutive rate equation. First attempt to propagate uncertainties through elastic–plastic constitutive equations considering random Young's modulus was published only recently, e.g. Anders and Hori (1999, 2000). The perturbation expansion at the stochastic mean behavior (considering only the first term of the expansion) was used in the above mentioned references. In computing the mean behavior the Authors took the advantage of bounding media approximation. Although this method doesn't suffer from computational difficulty associated with Monte Carlo method for problems having no closed-form solution, it inherits "closure problem" and the "small coefficient of variation" requirements for the material parameters. Closure problem refers to the need for higher order statistical moments in order to calculate lower order statistical moments Kavvas (2003). The small COV requirement claims that the perturbation method can be used (with reasonable accuracy) for probabilistic simulations of solids and structures with uncertain properties only if their COV < 20 % (Sudret and Der Kiureghian, 2000). For soils and other natural materials, COVs are rarely below 20 % (Lacasse and Nadim (1996); Phoon and Kulhawy (1999a,b)). Furthermore, with bounding media approximation, difficulty arises in computing the mean behavior when one considers uncertainties in internal variable(s) and/or direction(s) of evolution of internal variable(s).

The focus of present work is on development of methodology for the probabilistic simulation of constitutive behavior of elastic–plastic materials with uncertain properties. Recently, Kavvas (2003) obtained a generic Eulerian–Lagrangian (EL) form of FPK equation, exact to second-order, corresponding to any non–linear ordinary differential equation with random coefficients and random forcing. The approach using EL form of the FPK equation doesn't suffer from the drawbacks of Monte Carlo method and perturbation technique. In this paper the authors applied developed EL form of the FPK equation to obtain probabilistic formulation for a general, one-dimensional incremental elastic–plastic constitutive equation with random coefficient. The solution methodology is designed with several applications in mind, namely to

- obtain probabilistic stress–strain behavior from spatial average form (upscaled form) of constitutive equation, when input uncertain material properties to the constitutive equation are random fields; and
- obtain probabilistic stress-strain behavior from point-location scale constitutive equation, when input uncertain material properties to the constitutive equation are random variables.

Application of the developed methodology is demonstrated on a particular point-location scale one-dimensional constitutive equation, namely Drucker–Prager associative linear hardening elastic–plastic material model. In this paper, derivation is made of the EL form of FPK equation that govern the 1D probabilistic elastic–plastic material models with uncertain material parameters. This general formulation is then specialized to a particular 1D Drucker–Prager associative linear hardening material model. In the companion paper the solution methodology of the FPK equation corresponding to Drucker–Prager associative linear hardening material model is described, along with illustrative examples. The methodology is general enough that it allows extension to three-dimensions and incorporation into a general stochastic finite element framework. This work is underway and will be reported in future publications.

105.2.2 Probabilistic Elasto-Plasticity: General Formulation

The incremental form of spatial-average elastic-plastic constitutive equation can be written as

$$\frac{d\sigma_{ij}(x_t,t)}{dt} = D_{ijkl}(x_t,t)\frac{d\epsilon_{kl}(x_t,t)}{dt} \tag{105.1}$$

where the continuum stiffness tensor $D_{ijkl}(x_t, t)$ can be either elastic or elastic-plastic

$$D_{ijkl} = \begin{cases} D^{el}_{ijkl} & ; f < 0 \vee (f = 0 \wedge df < 0) \\ \\ D^{el}_{ijkl} - \dfrac{D^{el}_{ijmn} \dfrac{\partial U}{\partial \sigma_{mn}} \dfrac{\partial f}{\partial \sigma_{pq}} D^{el}_{pqkl}}{\dfrac{\partial f}{\partial \sigma_{rs}} D^{el}_{rstu} \dfrac{\partial U}{\partial \sigma_{tu}} - \dfrac{\partial f}{\partial q_*} r_*} & ; f = 0 \vee df = 0 \end{cases} \tag{105.2}$$

and where D^{el}_{ijkl} is the elastic stiffness tensor, D^{ep}_{ijkl} is the elastic–plastic continuum stiffness tensor, f is the yield function, which is a function of stress (σ_{ij}) and internal variables (q_*), U is the plastic potential function (also a function of stress and internal variables). The internal variables (q_*) could be scalar(s) (for perfectly-plastic and isotropic hardening models), second-order tensor (for translational and rotational kinematic hardening) or fourth-order tensor (for distortional hardening). Therefore, the most general form of incremental constitutive equation in terms of its parameters can be written as

$$\frac{d\sigma_{ij}(x_t, t)}{dt} = \beta_{ijkl}(\sigma_{ij}, D_{ijkl}, q_*, r_*; x_t, t) \frac{d\epsilon_{kl}(x_t, t)}{dt} \tag{105.3}$$

Due to randomnesses in elastic constants (D^{el}_{ijkl}) and internal variables (q_*) and/or rate of evolution of internal variables (r_*) the material stiffness operator β_{ijkl} in Eq. (105.3) becomes stochastic. It follows that the Equation (105.1) becomes a linear/non-linear ordinary differential equations with stochastic coefficients. Similarly, randomness in he forcing term (ϵ_{kl}) of Equation (105.3) results in Equation (105.3) becoming linear/non-linear ordinary differential equations with stochastic forcing. This can be generalized, so that randomnesses in material properties and forcing function of Equation (105.3) results in Equation (105.3) becoming a linear/non-linear ordinary differential equation with stochastic coefficients and stochastic forcing.

In order to gain better understanding of the effects of random material parameters and forcing on response, focus is shifted from a general 3D case to a 1D case. In what follows, the probabilistic formulation for 1-D constitutive elastic–plastic incremental equation with stochastic coefficient and stochastic forcing is derived. In addition to that, the probabilistic formulation for 1-D elastic linear constitutive equation is obtained as a special case of non-linear general derivation.

Focusing on 1-D behavior, the Eq. (105.3) is written as

$$\frac{d\sigma(x_t, t)}{dt} = \beta(\sigma, D, q, r; x_t, t) \frac{d\epsilon(x_t, t)}{dt} \tag{105.4}$$

which is a non-linear ordinary differential equation with stochastic coefficient and stochastic forcing. The right hand side of Eq. (105.4) is replaced with the function η as

$$\eta(\sigma, D, q, r, \epsilon; x, t) = \beta(\sigma, D, q, r; x_t, t) \frac{d\epsilon(x_t, t)}{dt} \tag{105.5}$$

so that now Eq. (105.4) can be written as

$$\frac{\partial \sigma(x_t, t)}{\partial t} = \eta(\sigma, D, q, r, \epsilon; x, t) \tag{105.6}$$

with initial condition,

$$\sigma(x, 0) = \sigma_0 \tag{105.7}$$

In the above Eq. (105.6) σ can be considered to represent a point in the σ-space and hence, the Eq. (105.6) determines the velocity for the point in that σ-space. This may be visualized, from the initial point, and given initial condition σ_0, as a trajectory that describes the corresponding solution of the non-linear stochastic ordinary differential equation (ODE) (Eq. (105.6)). Considering now a cloud of initial points (refer to Fig. 105.2), described by a density $\rho(\sigma, 0)$ in the σ-space.

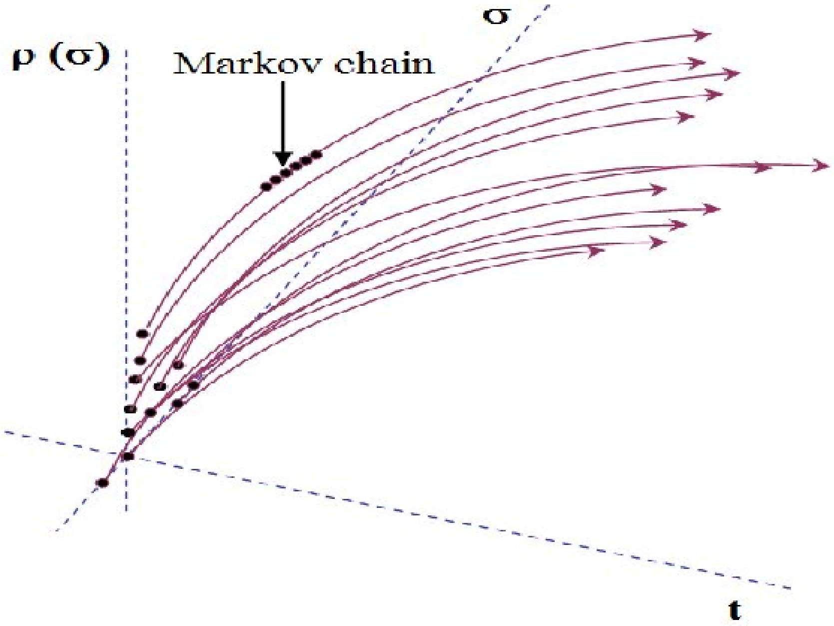


Figure 105.2: Movements of Cloud of Initial Points, described by density $\rho(\sigma, 0)$, in the σ-space

The phase density ρ of $\sigma(x, t)$ (movement of any point dictated by Eq. (105.6)) varies in time according to a continuity equation which expresses the conservation of all these points in the σ-space. This continuity equation can be expressed in mathematical terms, using Kubo's stochastic Liouville equation (Kubo, 1963):

$$\frac{\partial \rho(\sigma(x, t), t)}{\partial t} = -\frac{\partial}{\partial \sigma} \eta[\sigma(x, t), D(x), q(x), r(x), \epsilon(x, t)].\rho[\sigma(x, t), t] \tag{105.8}$$

with an initial condition,

$$\rho(\sigma, 0) = \delta(\sigma - \sigma_0) \tag{105.9}$$

where δ is the Dirac delta function and Eq. (105.9) is the probabilistic restatement in the σ-phase space of the original deterministic initial condition (Eq. (105.7)). Here it proves useful to recall Van Kampen's Lemma (Van Kampen, 1976), which states that the ensemble average of a phase density is the probability density

$$< \rho(\sigma, t) >= P(\sigma, t) \tag{105.10}$$

where, the symbol $< \cdot >$ denotes the expectation operation, and $P(\sigma, t)$ denotes evolutionary probability density of the state variable σ of the constitutive rate equation (Eq. (105.4)).

In order to obtain the deterministic probability density function (PDF) (σ, t) of the state variable, σ, it is necessary to obtain the deterministic partial differential equation (PDE) of the σ-space mean phase density $< \rho(\sigma, t) >$ from the linear stochastic PDE system (Eqs. (105.8) and (105.9)). This necessitates the derivation of the ensemble average form of Eq. (105.8) for $< \rho(\sigma, t) >$. This ensemble average was recently derived by (Kavvas and Karakas, 1996; Kavvas, 2003) as

$$\begin{aligned}
&\frac{\partial \left\langle \rho(\sigma(x_t, t), t) \right\rangle}{\partial t} = \\
&- \quad \frac{\partial}{\partial \sigma} \Bigg\{ \Bigg[\Bigg\langle \eta(\sigma(x_t, t), D(x_t), q(x_t), r(x_t), \epsilon(x_t, t)) \Bigg\rangle \\
&- \quad \int_0^t d\tau Cov_0 \Bigg[\eta(\sigma(x_t, t), D(x_t), q(x_t), r(x_t), \epsilon(x_t, t)); \\
&\qquad \frac{\partial \eta(\sigma(x_{t-\tau}, t-\tau), D(x_{t-\tau}), q(x_{t-\tau}), r(x_{t-\tau}) \epsilon(x_{t-\tau}, t-\tau)}{\partial \sigma} \Bigg] \Bigg] \left\langle \rho(\sigma(x_t, t), t) \right\rangle \Bigg\} \\
&+ \quad \frac{\partial}{\partial \sigma} \Bigg\{ \Bigg[\int_0^t d\tau Cov_0 \left[\eta(\sigma(x_t, t), D(x_t), q(x_t), r(x_t), \epsilon(x_t, t)); \right. \\
&\qquad \left. \eta(\sigma(x_{t-\tau}, t-\tau), D(x_{t-\tau}), q(x_{t-\tau}), r(x_{t-\tau}), \epsilon(x_{t-\tau}, t-\tau)) \right] \Bigg] \frac{\partial \left\langle \rho(\sigma(x_t, t), t) \right\rangle}{\partial \sigma} \Bigg\}
\end{aligned} \tag{105.11}$$

to exact second order (to the order of the covariance time of η). In Eq. (105.11), $Cov_0[\cdot]$ is the time ordered covariance function defined by

$$Cov_0 \left[\eta(x, t_1), \eta(x, t_2) \right] = \left\langle \eta(x, t_1) \eta(x, t_2) \right\rangle - \left\langle \eta(x, t_1) \right\rangle \cdot \left\langle \eta(x, t_2) \right\rangle \tag{105.12}$$

By combining Eqs. (105.11) and (105.10) and rearranging the terms yields the following Fokker-Planck equation (FPE, also known as Forward–Kolmogorov Equation or Fokker–Planck–Kolmogorov

FPK Equation) (Risken (1989), Gardiner (2004), Schüeller (1997)):

$$
\begin{aligned}
&\frac{\partial P(\sigma(x_t,t),t)}{\partial t} = \\
&-\quad \frac{\partial}{\partial \sigma}\Bigg[\Bigg\{\Bigg\langle \eta(\sigma(x_t,t), D(x_t), q(x_t), r(x_t)\epsilon(x_t,t)) \Bigg\rangle \\
&+\quad \int_0^t d\tau Cov_0 \Bigg[\frac{\partial \eta(\sigma(x_t,t), D(x_t), q(x_t), r(x_t)\epsilon(x_t,t))}{\partial \sigma}; \\
&\qquad \eta(\sigma(x_{t-\tau}, t-\tau), D(x_{t-\tau}), q(x_{t-\tau}), r(x_{t-\tau}), \epsilon(x_{t-\tau}, t-\tau)\Bigg]\Bigg\} P(\sigma(x_t,t),t)\Bigg] \\
&+\quad \frac{\partial^2}{\partial \sigma^2}\Bigg[\Bigg\{\int_0^t d\tau Cov_0 \Bigg[\eta(\sigma(x_t,t), D(x_t,t), q(x_t,t), r(x_t,t), \epsilon(x_t,t)); \\
&\qquad \eta_1(\sigma(x_{t-\tau}, t-\tau), D(x_{t-\tau}), q(x_{t-\tau}), r(x_{t-\tau}), \epsilon(x_{t-\tau}, t-\tau))\Bigg]\Bigg\} P(\sigma(x_t,t),t)\Bigg]
\end{aligned}
\tag{105.13}
$$

to exact second order. This is the most general relation for probabilistic behavior of inelastic (non–linear, elastic–plastic) 1-D stochastic incremental constitutive equation. The solution of this deterministic linear FPE (Eq. (105.13)), in terms of the probability density $P(\sigma, t)$, under appropriate initial and boundary conditions will yield the PDF of the state variable σ of the original 1-D non-linear stochastic constitutive rate equation (Eq. (105.4)). It is important to note that while the original equation (Eq. (105.4)) is non-linear, the FPE (Eq. (105.13)) is linear in terms of its unknown, the probability density $P(\sigma, t)$ of the state variable σ. This linearity, in turn, provides significant advantages in the solution of the probabilistic behavior of the incremental constitutive equation (Eq. (105.4)).

One should also note that Eq. (105.13) is a mixed Eulerian-Lagrangian equation. This stems from the fact that while the real space location x_t at time t is known, the location $x_{t-\tau}$ is an unknown. If one assumes small strain theory, one can relate the unknown location $x_{t-\tau}$ from the known location x_t by using the strain rate, $\dot{\epsilon}$ $(=d\epsilon/dt)$ as,

$$x_{t-\tau} = (1 - \dot{\epsilon}\tau)x_t \tag{105.14}$$

Once the probability density function $P(\sigma, t)$ is obtained it can be used to obtain the mean of state variable (σ) by usual expectation operation

$$< \sigma(t) >= \int \sigma(t) P(\sigma(t)) d\sigma(t) \tag{105.15}$$

Another possible way to obtain the mean of state variable is to use the equivalence between FPE and Itô stochastic differential equation (Gardiner, 2004). In this case Itô stochastic differential equation equivalent to Eq. (105.13) is

$$\begin{aligned}
d\sigma(x,t) &= \Bigg\{\Bigg\langle \eta(\sigma(x_t,t),D(x_t),q(x_t),r(x_t),\epsilon(x_t,t))\Bigg\rangle \\
&+ \int_0^t d\tau Cov_0\left[\frac{\partial\eta(\sigma(x_t,t),D(x_t),q(x_t),r(x_t),\epsilon(x_t,t))}{\partial\sigma};\right. \\
&\left.\eta(\sigma(x_{t-\tau},t-\tau),D(x_{t-\tau}),q(x_{t-\tau}),r(x_{t-\tau}),\epsilon(x_{t-\tau},t-\tau))\right]\Bigg\} dt \\
&+ b(\sigma,t)dW(t)
\end{aligned} \tag{105.16}$$

where,

$$\begin{aligned}
b^2(\sigma,t) &= 2\int_0^t d\tau Cov_0\Big[\eta(\sigma(x_t,t),D(x_t),q(x_t),r(x_t),\epsilon(x_t,t)); \\
&\eta(\sigma(x_{t-\tau},t-\tau),D(x_{t-\tau}),q(x_{t-\tau}),r(x_{t-\tau}),\epsilon(x_{t-\tau},t-\tau))\Big]
\end{aligned} \tag{105.17}$$

and, $dW(t)$ is an increment of Wiener process W with $< dW(t) >= 0$. It is also interesting to note that all the stochasticity of the original equation (Eq. (105.4)) are lumped together in the last term (Wiener increment term) of the right-hand-side of Eq. (105.16). By taking advantage of the independent increment property of the Wiener process ($< dW(t) >= 0$), one can derive the differential equation which describes the evolution of mean of state variable (σ) of the nonlinear constitutive rate equation in time and space as, (e.g. (Kavvas, 2003))

$$\begin{aligned}
\frac{< d\sigma(x,t) >}{dt} &= \Bigg\langle \eta(\sigma(x_t,t),D(x_t),q(x_t),r(x_t),\epsilon(x_t,t))\Bigg\rangle \\
&+ \int_0^t d\tau Cov_0\left[\frac{\partial\eta(\sigma(x_t,t),D(x_t),q(x_t),r(x_t),\epsilon(x_t,t))}{\partial\sigma};\right. \\
&\left.\eta(\sigma(x_{t-\tau},t-\tau),D(x_{t-\tau}),q(x_{t-\tau}),r(x_{t-\tau}),\epsilon(x_{t-\tau},t-\tau))\right]
\end{aligned} \tag{105.18}$$

Eq. (105.18) is a nonlocal integro-differential equation in Eulerian-Lagrangian form, since, although the location x_t at time t is known, the Lagrangian location $x_{t-\tau}$ is an unknown which is determined by Eq. (105.14). It is important to note that the state variable appearing within $\eta(\cdot)$ on the right-hand-side of Eq. (105.18) is random and needs to be treated accordingly.

This concludes the development of relation for probabilistic behavior of 1-D elastic–plastic constitutive incremental equation with stochastic coefficients and stochastic forcing in most general form. In the following section the developed general relation is specialized to two particular types of point-location scale constitutive modeling: a) 1-D (shear) linear elastic constitutive behavior, and b) 1-D (shear) elastic-plastic Drucker-Prager associative linear hardening constitutive behavior.

105.2.3 Probabilistic Elasto-Plasticity: Elastic–Plastic Probabilistic 1-D Constitutive Incremental Equation

For materials obeying Drucker-Prager yield criteria (without cohesion), the yield surface can be written as:

$$f = \sqrt{J_2} - \alpha I_1 \tag{105.19}$$

where $J_2 = \frac{1}{2} S_{ij} s_{ij}$ is the second invariant of the deviatoric stress tensor $s_{ij} = \sigma_{ij} - 1/3\delta_{ij}\sigma_{kk}$, and $I_1 = \sigma_{ii}$ is the first invariant of the stress tensor, and α, an internal variable, is a function of friction angle ($\alpha = 2\sin(\phi)/(\sqrt{(}3)(3 - sin\phi)$), where ϕ is the friction angle (e.g. (Chen and Han, 1988b)).

By assuming associative flow rule, so that the yield function has the same derivatives as the plastic flow function

$$\frac{\partial f}{\partial \sigma_{ij}} = \frac{\partial U}{\partial \sigma_{ij}} \tag{105.20}$$

one can expand parts of the tangent constitutive tensor given in Eq. (105.2) (from Page 424), to read[2]

$$\begin{aligned} A_{kl} = \frac{\partial f}{\partial \sigma_{pq}} D_{pqkl} = \quad & A_{kl} \frac{\partial f}{\partial I_1} \left(2G \left(\frac{\partial I_1}{\partial \sigma_{11}} \delta_{1l}\delta_{1k} + \frac{\partial I_1}{\partial \sigma_{22}} \delta_{2l}\delta_{2k} + \frac{\partial I_1}{\partial \sigma_{33}} \delta_{3l}\delta_{3k} \right) \right. \\ + & \left. \left(K - \frac{2}{3}G \right) \frac{\partial I_1}{\partial \sigma_{cd}} \delta_{cd}\delta_{kl} \right) \\ + & \frac{\partial f}{\partial \sqrt{J_2}} \left(2G \frac{\partial \sqrt{J_1}}{\partial \sigma_{ij}} \delta_{ik}\delta_{jl} + \left(K - \frac{2}{3}G \right) \frac{\partial \sqrt{J_2}}{\partial \sigma_{ab}} \delta_{ab}\delta_{kl} \right) \end{aligned} \tag{105.21}$$

and,

$$\begin{aligned} B \;=\; & \frac{\partial f}{\partial \sigma_{rs}} D_{rstu} \frac{\partial f}{\partial \sigma_{tu}} = \\ & \left(\frac{\partial f}{\partial I_1} \right)^2 \left(2G \left(\left(\frac{\partial I_1}{\partial \sigma_{11}} \right)^2 + \left(\frac{\partial I_1}{\partial \sigma_{22}} \right)^2 + \left(\frac{\partial I_1}{\partial \sigma_{33}} \right)^2 \right) \right. \\ + & \left. \left(K - \frac{2}{3}G \right) \left(\frac{\partial I_1}{\partial \sigma_{ij}} \delta_{ij} \right)^2 \right. \\ + & \left(\frac{\partial f}{\partial \sqrt{J_2}} \right)^2 \left(2G \frac{\partial \sqrt{J_2}}{\partial \sigma_{ij}} \frac{\partial \sqrt{J_2}}{\partial \sigma_{ij}} + \left(K - \frac{2}{3}G \right) \left(\frac{\partial \sqrt{J_2}}{\partial \sigma_{ij}} \delta_{ij} \right)^2 \right) \Bigg) \end{aligned} \tag{105.22}$$

where, K and G are the elastic bulk modulus and the elastic shear modulus respectively.

---

[2] A more detailed derivation of this probabilistic differentiation is given in the Appendix.

By further assuming that the evolution of internal variable is a function of equivalent plastic strain[3], $e^p_{eq} = 2/3 e^p_{ij} e^p_{ij}$ then one can write

$$K_P = -\frac{\partial f}{\partial q_n} r_n = -\frac{1}{\sqrt{3}} \frac{\partial f}{\partial \alpha} \frac{d\alpha}{de^p_{eq}} \frac{\partial f}{\partial \sqrt{J_2}} \tag{105.23}$$

It should be noted that since material properties are assumed to be random, the resulting stress tensor will also become random and hence the derivatives of the stress invariants with respect to stress tensor (σ_{ij}) will become random. Therefore, differentiations appearing in Eqs. (105.21), (105.22), and (105.23) can not be carried out in a deterministic sense.

The parameter tensor in Eq. (105.1) then becomes

$$D^{ep}_{ijkl} = \begin{cases} 2G\delta_{ik}\delta_{jl} + \left(K - \frac{2}{3}G\right)\delta_{ij}\delta_{kl} & ; f < 0 \vee (f = 0 \wedge df < 0) \\ \\ 2G\delta_{ik}\delta_{jl} + \left(K - \frac{2}{3}G\right)\delta_{ij}\delta_{kl} - \frac{A_{ij}A_{kl}}{B + K_P} & ; f = 0 \vee df = 0 \end{cases} \tag{105.24}$$

where tensor A_{ij} and scalars B and K_P are defined by Eqs. (105.21), (105.22), and (105.23) respectively. The above equation (Eq. 105.24) represents a probabilistic continuum stiffness tensor for an elastic–plastic material model, in this case Drucker-Prager isotropic linear hardening material with associated plasticity. By focusing our attention on one dimensional point-location scale shear constitutive relationship between σ_{12} and ϵ_{12} for Drucker-Prager material model, one can simplify the function $\eta(\sigma, D, q, r, \epsilon; x, t)$ as defined in Eq. (105.5) (on Page 424) to read

$$\eta = \begin{cases} 2G\frac{d\epsilon_{12}}{dt} & ; f < 0 \vee (f = 0 \wedge df < 0) \\ \left(2G - \frac{4G^2\left(\frac{\partial f}{\partial\sqrt{J_2}}\frac{\sqrt{J_2}}{\partial\sigma_{12}}\right)^2}{B + K_P}\right)\frac{d\epsilon_{12}}{dt} & ; f = 0 \vee df = 0 \end{cases} \tag{105.25}$$

By considering both the material properties (shear modulus G, bulk modulus K, friction angle α, and rate of change of friction angle (linear hardening) α') and the strain rate ($d\epsilon_{12}/dt(t)$) as random, one can substitute η as derived in Eq. (105.13) to obtain the particular FPK equation for the probabilistic behavior of Drucker-Prager associative linear hardening, 1-D point-location scale elastic-plastic shear constitutive rate equation. In particular, two cases are recognized, one for elastic (pre–yield) behavior of

---

[3] This is a fairly common assumption, e.g. (Chen and Han, 1988b)

material $(f < 0 \vee (f = 0 \wedge df < 0))$

$$
\begin{aligned}
&\frac{\partial P(\sigma_{12}(t),t)}{\partial t} = \\
&- \frac{\partial}{\partial \sigma_{12}} \left[\left\langle 2G \frac{d\epsilon_{12}}{dt}(t) \right\rangle P(\sigma_{12}(t),t) \right] \\
&+ \frac{\partial^2}{\partial \sigma_{12}^2} \left[\left\{ \int_0^t d\tau Cov_0 \left[2G \frac{d\epsilon_{12}}{dt}(t); 2G \frac{d\epsilon_{12}}{dt}(t-\tau) \right] \right\} P(\sigma_{12}(t),t) \right]
\end{aligned}
\tag{105.26}
$$

noting that this is the same equation as Eq. (??). In addition to that, the case of elastic–plastic behavior $(f = 0 \vee df = 0)$ is described by the following probabilistic equation

$$
\begin{aligned}
&\frac{\partial P(\sigma_{12}(t),t)}{\partial t} = -\frac{\partial}{\partial \sigma_{12}} \left[\left\{ \left\langle \left(G^{ep}(t) \right) \frac{d\epsilon_{12}}{dt}(t) \right\rangle \right.\right. \\
&+ \left.\left. \int_0^t d\tau Cov_0 \left[\frac{\partial}{\partial \sigma_{12}} \left(G^{ep}(t) \frac{d\epsilon_{12}}{dt}(t) \right); G^{ep}(t-\tau) \frac{d\epsilon_{12}}{dt}(t-\tau) \right] \right\} P(\sigma_{12}(t),t) \right] \\
&+ \frac{\partial^2}{\partial \sigma_{12}^2} \left[\left\{ \int_0^t d\tau Cov_0 \left[G^{ep}(t) \frac{d\epsilon_{12}}{dt}(t); G^{ep}(t-\tau) \frac{d\epsilon_{12}}{dt}(t-\tau) \right] \right\} P(\sigma_{12}(t),t) \right]
\end{aligned}
\tag{105.27}
$$

where $G^{ep}(a)$ is defined as probabilistic elastic–plastic kernel and is introduced to shorten the writing (but will also have other uses later)

$$
G^{ep}(a) = \left(2G - \frac{4G^2 \left(\frac{\partial f}{\partial \sqrt{J_2}(a)} \frac{\partial \sqrt{J_2}(a)}{\partial \sigma_{12}(a)} \right)^2}{B(a) + K_P(a)} \right)
\tag{105.28}
$$

and a assumes values t or $t - \tau$.

It is important to note that the differentiations appearing in the coefficient terms of the FPK PDE (Eq. (105.27)), within the probabilistic elastic–plastic kernel $G^{ep}(a)$ (i.e. Eq. (105.28)), are for fixed values of σ_{12} and hence those differentiations can be carried out in a deterministic sense. After carrying out the differentiations, the probabilistic elastic–plastic kernel becomes

$$
G^{ep}(a)|_{\sigma_{12} \rightarrow const.} = \left(2G - \frac{G^2}{G + 9K\alpha^2 + \frac{1}{\sqrt{3}} I_1(a) \alpha'} \right)
\tag{105.29}
$$

which, after substitution, result in simplification of the FPK equation (105.27). Further simplification is possible by noting that the first random process in the covariance term of the first coefficient on the r.h.s of the equation (105.27) is independent of σ_{12}. Furthermore, since the covariance of zero with any random process is zero, the FPK equation (105.27) is further simplified to read

$$\begin{aligned}
&\frac{\partial P(\sigma_{12}(t),t)}{\partial t} = \\
&- \frac{\partial}{\partial \sigma_{12}}\left[\left\langle G^{ep}(t)\frac{d\epsilon_{12}}{dt}(t)\right\rangle P(\sigma_{12}(t),t)\right] \\
&+ \frac{\partial^2}{\partial \sigma_{12}^2}\left[\left\{\int_0^t d\tau Cov_0\left[G^{ep}(t)\frac{d\epsilon_{12}}{dt}(t); G^{ep}(t-\tau)\frac{d\epsilon_{12}}{dt}(t-\tau)\right]\right\} P(\sigma_{12}(t),t)\right]
\end{aligned} \tag{105.30}$$

where the probabilistic elastic–plastic kernel $G^{ep}(a)$ is given by the Eq. (105.29).

The evolution of a mean value of shear stress σ_{12} is obtained by substituting η (derived for Drucker-Prager material in Eq. (105.18)) as

$$\begin{aligned}
&\frac{< d\sigma_{12}(t) >}{dt} = \left\langle G^{ep}(t)\frac{d\epsilon_{12}}{dt}(t)\right\rangle \\
&+ \int_0^t d\tau Cov_0\left[\frac{\partial}{\partial \sigma_{12}}\left(G^{ep}(t)\frac{d\epsilon_{12}}{dt}(t)\right); G^{ep}(t-\tau)\frac{d\epsilon_{12}}{dt}(t-\tau)\right]
\end{aligned} \tag{105.31}$$

It is important to note that the derivatives appearing in the mean and covariance term of the above Eulerian-Lagrangian integro-differential equation (Eq. (105.31) with the probabilistic elastic–plastic kernel defined through the Eq. (105.29)) are random differentiations and need to be treated accordingly. One possible approach to obtaining these differentiations could be perturbation with respect to mean (Anders and Hori, 2000) but the "closure problem" will appear. Hence, in this study the evolution of mean of σ_{12} will be obtained by the expectation operation on the PDF (Eq. (105.15)).

105.2.4 Probabilistic Elasto-Plasticity: Initial and Boundary Conditions for the Probabilistic Elastic–Plastic PDE

The PDE describing the probabilistic behavior of constitutive rate equations can be written in the following general form:

$$\begin{aligned}
\frac{\partial P(\sigma_{12},t)}{\partial t} &= -\frac{\partial}{\partial \sigma_{12}}\left\{P(\sigma_{12},t)N_{(1)}\right\} + \frac{\partial^2}{\partial \sigma_{12}^2}\left\{P(\sigma_{12},t)N_{(2)}\right\} \\
&= -\frac{\partial}{\partial \sigma_{12}}\left[P(\sigma_{12},t)N_{(1)} - \frac{\partial}{\partial \sigma_{12}}\left\{P(\sigma_{12},t)N_{(2)}\right\}\right] \\
&= -\frac{\partial \zeta}{\partial \sigma_{12}}
\end{aligned} \tag{105.32}$$

where, $N_{(1)}$ and $N_{(2)}$ are coefficients[4] of the PDE and represent the expressions within the curly braces of the first and second terms respectively on the right–hand–side of Eqs. (??), (105.26), and (105.27) These terms are called the advection ($N_{(1)}$) and diffusion ($N_{(2)}$) coefficients as the form of Eq. (105.32)

---

[4] Indices in brackets are not used in index summation convention.

closely resembles advection–diffusion equation (Gardiner, 2004). The symbol ζ in Eq. (105.32) can be considered to be the probability current. This follows from Eq. (105.32), which is a continuity equation and the state variable of the equation is probability density.

After introducing initial and boundary conditions, one can solve Eq. (105.32) for probability densities of σ_{12} with evolution of time. The initial condition could be deterministic or stochastic depending on the type of problem. For probabilistic behavior of linear elastic constitutive rate equation (Eq. (??)), one can assume that all the probability mass at time $t = 0$ is concentrated at $\sigma_{12} = 0$ or at some constant value of σ_{12} if there were some initial stresses to begin with (e.g. overburden pressure on a soil mass).

In mathematical term, this translates to,

$$P(\sigma_{12}, 0) = \delta(\sigma_{12}) \tag{105.33}$$

where, $\delta(\cdot)$ is the Dirac delta function.

For the post–yield behavior of probabilistic elastic-plastic constitutive rate Equation[5] (105.30), there will be a distribution of σ_{12}, corresponding to the solution of the pre-yield probabilistic behavior (Eq. (105.26)), to begin with. This probability mass ($P(\sigma_{12}(t), t)$), dictated by Eq. (105.13), will advect and diffuse into the domain (σ_{12}, t space) of the system throughout the evolution (in time/strain) of the simulation. Since it is required that the probability mass within the system is conserved i.e. no leaking is allowed at the boundaries, a reflecting barrier at the boundaries will be the preferred choice. In mathematical term, one can express this condition as (Gardiner, 2004)

$$\zeta(\sigma_{12}, t)\big|_{AtBoundaries} = 0 \tag{105.34}$$

In theory, the stress domain could extend from $-\infty$ to ∞ so that boundary conditions are then

$$\zeta(-\infty, t) = \zeta(\infty, t) = 0 \tag{105.35}$$

With these initial and boundary conditions, the probabilistic differential equation (with random material properties and random strain) for elasto–plasticity, specialized in this case to associated Drucker–Prager material model with linear hardening, and by using FPK transform described above, can be solved for probability densities of shear stress (σ_{12}) as it evolves with time/shear strain (ϵ_{12}).

105.2.5 Probabilistic Elasto-Plasticity: Fokker–Planck–Kolmogorov Equation for Probabilistic Elasticity and Elasto–Plasticity in 1-D

By focusing attention to the randomness of material properties only (i.e. assuming the forcing function (strain rate) as deterministic), partial differential equation (PDE) describing the evolution of probability

---

[5] Specialized to Drucker-Prager associated linear hardening model.

density function (PDF) of stress can be simplified. In particular, for 1D case, and for linear elastic material (but still with probabilistic material properties, in this case shear modulus G) one can write the following PDE

$$\begin{aligned}\frac{\partial P(\sigma_{12}(t))}{\partial t} &= - \left\langle 2G\frac{d\epsilon_{12}}{dt} \right\rangle \frac{\partial P(\sigma_{12}(t))}{\partial \sigma_{12}} \\ &+ \left\{ \int_0^t d\tau Cov_0 \left[2G\frac{d\epsilon_{12}}{dt}; 2G\frac{d\epsilon_{12}}{dt} \right] \right\} \frac{\partial^2 P(\sigma_{12}(t))}{\partial \sigma_{12}^2}\end{aligned} \tag{105.36}$$

Similarly, for elastic–plastic state, again by neglecting the randomness in strain rate, one can write the PDE for evolution of PDF of stress in 1D as

$$\begin{aligned}\frac{\partial P(\sigma_{12}(t))}{\partial t} &= - \left\langle \left(G^{ep}(t)\right) \frac{d\epsilon_{12}}{dt} \right\rangle \frac{\partial P(\sigma_{12}(t))}{\partial \sigma_{12}} \\ + \left\{ \int_0^t d\tau Cov_0 \left[G^{ep}(t)\frac{d\epsilon_{12}}{dt}; G^{ep}(t-\tau)\frac{d\epsilon_{12}}{dt} \right] \right\} &\frac{\partial^2 P(\sigma_{12}(t))}{\partial \sigma_{12}^2}\end{aligned} \tag{105.37}$$

where $G^{ep}(a)$ is the probabilistic elastic–plastic tangent stiffness, (given in Jeremić et al. (2007a))

$$G^{ep}(a) = 2G - \frac{G^2}{G + 9K\alpha^2 + \frac{1}{\sqrt{3}} I_1(a)\alpha'} \tag{105.38}$$

where in the previous equation (105.38), a assumes values t or $t - \tau$. With appropriate initial and boundary conditions as described in Jeremić et al. (2007a), one can solve Eqs. (105.36) and (105.37) for evolution of PDF of shear stress with shear strain.

105.2.6 Probabilistic Elasto-Plasticity: Example Problem Statements

The applicability of proposed FPK equations (Eqs. (105.36) and (105.37)) in describing probabilistic elasto-plastic behavior, is verified using the following three example problems.

Problem I. Assume the material is linear elastic, probabilistic, with probabilistic shear modulus (G) given by a normal distribution at a point–location scale with mean of 2.5 MPa and standard deviation of 0.707 MPa. The aim is to calculated the evolution of PDF of shear stress (σ_{12}) with shear strain (ϵ_{12}) for a displacement-controlled test with deterministic shear strain increment. The other parameters are considered deterministic and are as follows: Poisson's ratio ($\nu = 0.2$, and confining pressure $I_1 = 0.03$ MPa.

Problem II. Assume elastic–plastic material model, composed of linear elastic component and Drucker–Prager associative isotropic linear hardening elastic–plastic component. The probabilistic shear modulus

(G) is given through a normal distribution at a point–location scale with mean of 2.5 MPa and standard deviation of 0.707 MPa. The aim is to calculate the evolution of the PDF of shear stress (σ_{12}) with shear strain (ϵ_{12}) for a displacement-controlled test with deterministic shear strain increment. The other parameters are considered deterministic and are as follows: Poisson's ratio $\nu = 0.2$, confining pressure $I_1 = 0.03$ MPa, yield parameter[6] $\alpha = 0.071$, plastic slope[7] $\alpha' = 5.5$.

Problem III. Assume elastic–plastic material model, with linear elastic component and Drucker–Prager associative isotropic linear hardening elastic–plastic component. The probabilistic yield parameter (α) is given through a normal distribution at a point-location scale with mean of 0.52 and standard deviation of 0.1. The aim is to calculate the evolution of the PDF of shear stress (σ_{12}) with shear strain (ϵ_{12}) for a displacement-controlled test with deterministic shear strain increment. The other parameters are considered deterministic and are as follows: shear modulus $G = 2.5$ MPa, Poisson's ratio $\nu = 0.2$, confining pressure $I_1 = 0.03$ MPa, and the plastic slope $\alpha' = 5.5$.

The above three problems will be solved using the proposed FPK equation approach. In addition to that, the solution will verified using either variable transformation method, for linear elastic case or repetitive Monte Carlo type simulations for elastic-plastic case.

105.2.7 Probabilistic Elasto-Plasticity: Determination of Coefficients for Fokker–Planck–Kolmogorov Equation

To solve Problems I, II, and III, the advection and diffusion coefficients $N_{(1)}$ and $N_{(2)}$ must be determined for all three problems. For sake of simplicity, a constant strain rate is assumed and hence, terms containing $d\epsilon_{12}/dt$ in coefficients of Eqs. (105.36) and (105.37) can be substituted by a constant numerical value for the entire simulation of the evolution of PDF. It should be noted that the FPK equation (Eqs. (105.36) or (105.37)) describes the evolution of PDFs of stress with time, while, similarly, strain rate describes the evolution of strain with time. Combining the two, the evolution of PDF of stress with strain can be obtained. Time has been brought in this simulation as an intermediate dimension to help in solution process, and hence, the numerical value of strain rate could be any arbitrary value, which will cancel out once the time evolution of PDF of stress is converted to strain evolution of PDF of stress. For simulation of all the three example problems, an arbitrary value of strain rate of $d\epsilon_{12}/dt = 0.0541/s$ is assumed.

It should also be noted that since the material properties are assumed as random variables at a point-location scale, the covariance terms appearing within the advection and diffusion coefficients become variances of random variables. For estimations of means and variances of functions of random variables

---

[6] The yield parameter α is an internal variable and is a function of friction angle ϕ given by $(\alpha = 2\sin(\phi)/(\sqrt(3)(3 - sin\phi))$ (e.g. (Chen and Han, 1988b))

[7] The plastic slope α' is a rate of change of friction angle governing linear hardening.

(e.g. for Problems II and III) from basic random variables, commercially available statistical software mathStatica Rose and Smith (2002) was used.

Substituting the values of deterministic and random material properties and the strain rate, coefficients $N_{(1)}$ and $N_{(2)}$ of the FPK equations can be obtained for all problems:

Problem I

$$\begin{aligned} N_{(1)} &= \left\langle 2G\frac{d\epsilon_{12}}{dt} \right\rangle \\ &= 2\frac{d\epsilon_{12}}{dt}\langle G \rangle \\ &= 0.27 \quad \text{MPa/s} \end{aligned}$$

$$\begin{aligned} N_{(2)} &= \int_0^t d\tau Var\left[2G\frac{d\epsilon_{12}}{dt}\right] \\ &= 4t\left(\frac{d\epsilon_{12}}{dt}\right)^2 Var[G] \\ &= 0.0058t \quad (\text{MPa/s})^2 \end{aligned}$$

Problem II

For pre-yield linear elastic case, the coefficients $N_{(1)}$ and $N_{(2)}$ will be the same as those for Problem I. For post-yield elastic-plastic case the coefficients are

$$\begin{aligned} N_{(1)} &= \left\langle \left(2G - \frac{G^2}{G + 9K\alpha^2 + \frac{1}{\sqrt{3}}I_1\alpha'} \right) \frac{d\epsilon_{12}}{dt} \right\rangle \\ &= \frac{d\epsilon_{12}}{dt}\left\langle 2G - \frac{G^2}{G + 9K\alpha^2 + \frac{1}{\sqrt{3}}I_1\alpha'} \right\rangle \\ &= 0.147 \quad \text{MPa/s} \end{aligned}$$

$$\begin{aligned} N_{(2)} &= t\left(\frac{d\epsilon_{12}}{dt}\right)^2 Var\left[2G - \frac{G^2}{G + 9K\alpha^2 + \frac{1}{\sqrt{3}}I_1\alpha'}\right] \\ &= 0.00074t \quad (\text{MPa/s})^2 \end{aligned}$$

Problem III

For post-yield elastic-plastic simulation the coefficients $N_{(1)}$ and $N_{(2)}$ are

$$
\begin{aligned}
N_{(1)} &= \left\langle \left(2G - \frac{G^2}{G + 9K\alpha^2 + \frac{1}{\sqrt{3}} I_1 \alpha'} \right) \frac{d\epsilon_{12}}{dt} \right\rangle \\
&= \frac{d\epsilon_{12}}{dt} \left\langle 2G - \frac{G^2}{G + 9K\alpha^2 + \frac{1}{\sqrt{3}} I_1 \alpha'} \right\rangle \\
&= 0.2365 \quad \mathrm{MPa/s}
\end{aligned}
$$

$$
\begin{aligned}
N_{(2)} &= t \left(\frac{d\epsilon_{12}}{dt} \right)^2 Var \left[2G - \frac{G^2}{G + 9K\alpha^2 + \frac{1}{\sqrt{3}} I_1 \alpha'} \right] \\
&= 0.0001t \quad (\mathrm{MPa/s})^2
\end{aligned}
$$

It should be noted that for Problem III, since the shear modulus is deterministic, the pre-yield elastic case is deterministic.

105.2.8 Probabilistic Elasto-Plasticity: Results and Verifications of Example Problems

In this section results are presented for elastic and elastic–plastic probabilistic 1D problem. The results are obtained by using FPK equation approach described in previous sections and in the companion paper (Jeremić et al., 2007a). In addition to that, the Monte Carlo based verification of developed solutions (results) is presented. The effort to verify developed solutions (that are based on FPK approach) plays a crucial role in presented development of probabilistic elasto–plasticity as there are no previously published solutions which could have been used for verification. In addition to that, verification and validation efforts should always be included in any modeling and simulations work (Oberkampf et al., 2002).

For linear elastic constitutive rate equations (Problem-I and pre-yield case of Problem-II) the verification is performed by comparing solutions obtained through the use of FPK equation approach with high accuracy (exact) solution, using a transformation method of random variables (Montgomery and Runger, 2003). This method is applicable as for rate-independent linear elastic case the 1D shear constitutive equation simplify to a linear algebraic equation of the form,

$$\sigma_{12} = 2G\epsilon_{12} = u(G, \epsilon_{12}) \tag{105.39}$$

Using the definition of strain rate, the above equation can be written in terms of time t as,

$$\sigma_{12} = 2G(0.054t) = v(G, t) \tag{105.40}$$

where, 0.054 $1/\mathrm{s}$ is the arbitrary strain-rate assumed for this example problem. According to the transformation method of random variables (Montgomery and Runger, 2003), and, given the continuous random variable (shear modulus) G, with PDF $g(G)$ and Eqs. (105.39) or (105.40) as one-to-one transformations between the values of random variables of G and σ_{12}, one can obtain the PDF of shear stress (σ_{12}), $P(\sigma_{12})$ as,

$$P(\sigma_{12}) = g(u^{-1}(\sigma_{12}, \epsilon_{12}))\,|J| \tag{105.41}$$

which will allow for predicting the evolution of PDF of σ_{12} with ϵ_{12} or,

$$P(\sigma_{12}) = g(v^{-1}(\sigma_{12}, t))\,|J| \tag{105.42}$$

Eq. (105.42) will predict the evolution of PDF of σ_{12} with t. In Eqs. (105.41) and (105.42), functions $G = u^{-1}(\sigma_{12}, \epsilon_{12})$ or $G = u^{-1}(\sigma_{12}, t)$ are the inverse of functions $\sigma_{12} = u(G, \epsilon_{12})$ or $\sigma_{12} = v(G, t)$ respectively and $J = du^{-1}(\sigma_{12}, \epsilon_{12})/d\sigma_{12}$ and $J = dv^{-1}(\sigma_{12}, t)/d\sigma_{12}$ are their respective Jacobians of transformations.

For non-linear elastic-plastic constitutive rate equations (post-yield cases of Problems II and III) the verification is done using Monte-Carlo simulation technique by generating sample data for material properties from standard normal distribution and by repeating solution of the deterministic elastic-plastic constitutive rate equation for each data generated above. The probabilistic characteristics of resulting random stress variable for each time (or strain) step are then easily computed. A relatively large number of data points (1,000,000) were generated for each material constant random variable for this simulation purpose.

105.2.9 Problem I

The evolution of PDF of shear stress with time and shear strain is shown in Figures 105.3 and 105.4. Presented PDFs are for linear elastic material with random shear modulus, and were obtained using *FPE approach* (Fig. reffigure:ElasticPDF) and *transformation method* (Fig. 105.4).

The contours of evolution of PDFs are compared in Fig. 105.5. Similarly, comparison of the evolution of mean and standard deviations are shown in Fig. 105.6. It can be seen from the comparison figure that even-though the FPK approach predicted the mean behavior exactly, it slightly over-predicted the standard deviation. This is because of the approximation used to represent the Dirac delta function, which was used as the initial condition for the FPK. One may note that at $\epsilon_{12} = 0$, the probability of shear stress σ_{12} should theoretically be 1 i.e. all the probability mass should theoretically be concentrated at $\sigma_{12} = 0$. As such, it would be best described by the Dirac delta function. However, for numerical simulation of FPK, Dirac delta function as initial condition was approximated with a Gaussian function of

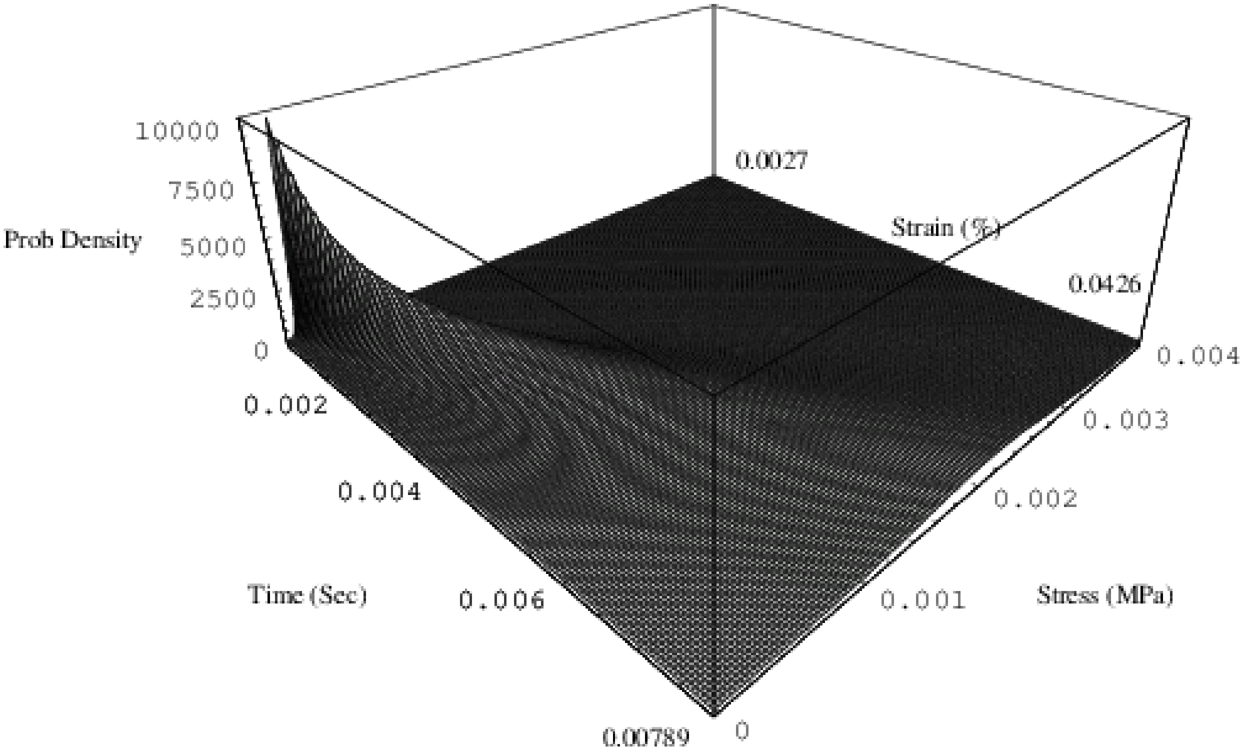


Figure 105.3: Evolution of PDF of shear stress versus strain (or time) for linear elastic material model with random shear modulus (Problem I) obtained using FPK equation approach.

mean zero and standard deviation of 0.00001 MPa, as shown in Fig. ??. This error in the initial condition advected and diffused into the domain with the simulation of the evolution process. This error could be minimized by better approximating the Dirac delta initial condition (but at higher computational cost). The effect of approximating the initial condition of the PDF of shear stress at $\epsilon_{12} = 0.0426$ % is shown in Fig. 105.7. In this figure the actual PDF at $\epsilon_{12} = 0.0426$ % obtained using the transformation method was compared with the PDFs at $\epsilon_{12} = 0.0426$ % obtained using the FPK approach with three different approximate initial conditions - all having zero mean but standard deviations of 0.01 MPa, 0.005 MPa and 0.00001 MPa.

One may also note that finer approximation of initial condition necessitates finer discretization of stress domain close to (or at) σ_{12} =0. The finite difference discretization scheme adopted here uses the same fine discretization uniformly all throughout the entire domain. It is noted that that fine, uniform discretization is not needed (and is quite expensive) in later stages of calculation of evolution of PDF, but is kept the same for simplicity sake. In presented examples, to properly capture the approximate initial condition (as shown in Fig. ??), the stress domain between –0.1 MPa and +0.1 MPa was discretized with a uniform step size of 0.000005 MPa and hence there is a total of $40,000$ nodes. This not only requires

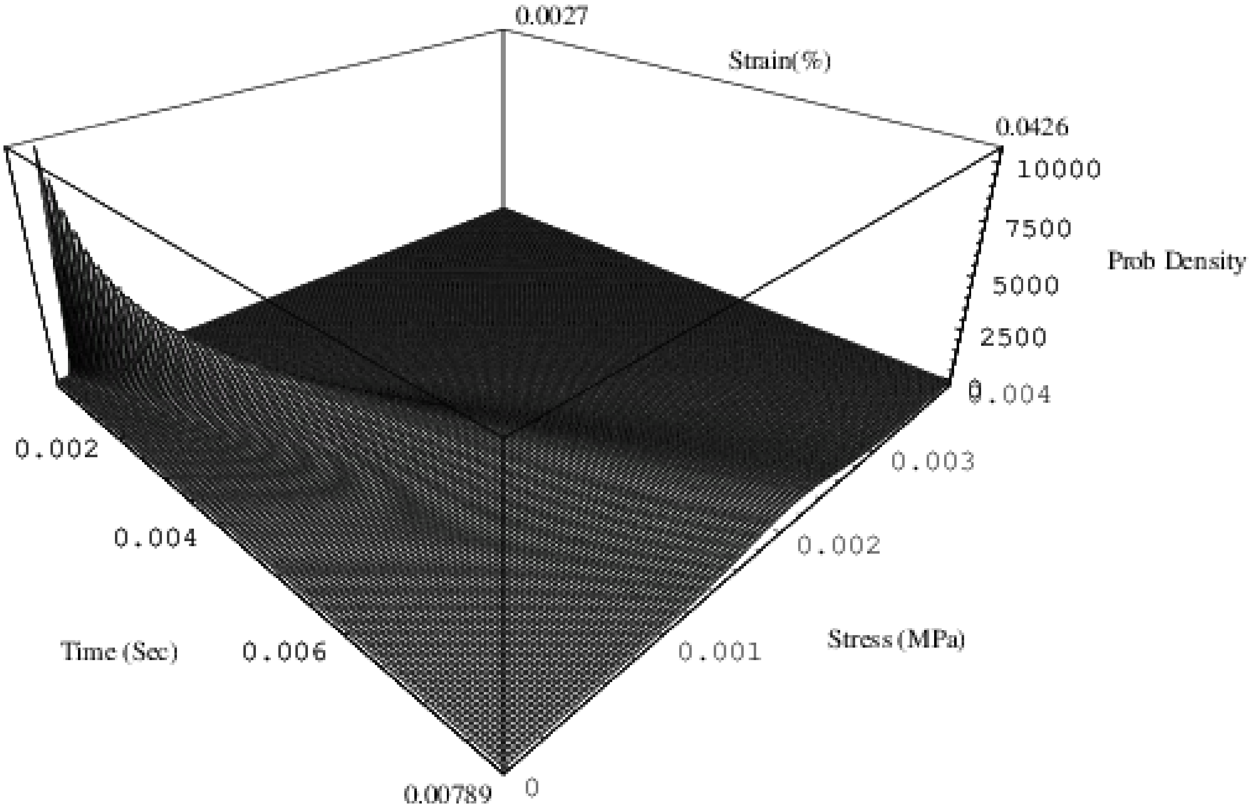


Figure 105.4: Evolution of PDF of shear stress versus strain (or time) for linear elastic material model with random shear modulus (Problem I) obtained using transformation method.

large computational effort but is also very memory sensitive. An adaptive discretization technique will be a much better approach to solving this problem. Current work is going on in formulating an adaptive algorithm for the solution of this type of problem.

105.2.10 Problem II

The solution to this problem involves the solving two FPK equations, one corresponding to the pre-yield elastic part and the other corresponding to the post-yield elastic-plastic part. The elastic part of this problem is identical to Problem–I. The initial condition for the post-yield elastic-plastic part of the problem is random and is shown in Fig. 105.8. It may be noted that this initial condition corresponds to the PDF of shear stress ($P(\sigma_{12})$) at yield obtained from the solution of FPK equation of the pre-yield elastic part. A view of the surface of evolution of the PDF of shear stress versus shear strain (time) is shown in Fig. 105.9. Another view to the PDF of stress–strain surface is shown in Fig. 105.10. It is noted that the yielding of this material occurred at t=0.00789 second (which is equivalent to ϵ_{12}=

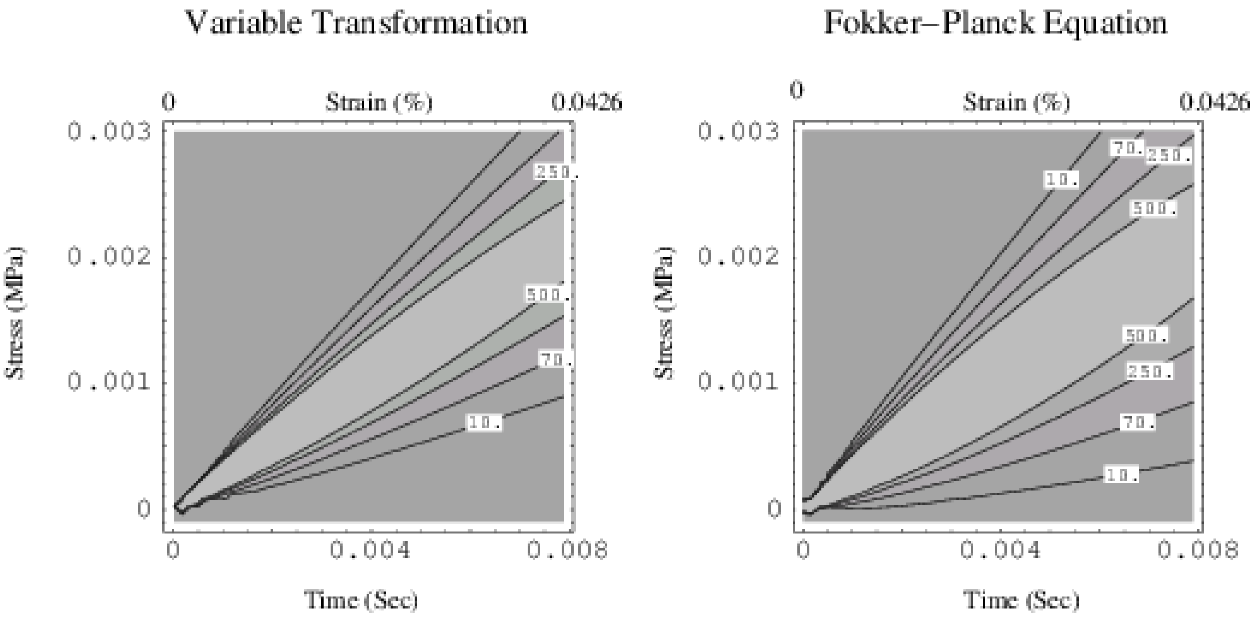


Figure 105.5: Comparison of Contours of Time (or Strain) Evolution of Probability Density Function for Shear Stress for Elastic Constitutive Rate Equation with Random Shear Modulus (Problem–I) for FPE Solution and Variable Transformation Method Solution.

0.0426 %). The evolution contours for PDF of shear stress versus strain (time) along with the mean and standard deviations are shown in Fig. 105.11. It can be seen from that figure that, as expected, the evolution of mean of shear stress changes slope after the material yielded. Another interesting aspect to note is the relative slope of the evolution of standard deviation with respect to the evolution of mean. The relative slope in the pre-yield elastic zone increases at a higher rate during the evolution process when compared with that in the post-yield elastic-plastic zone. In other words, in the evolution process the post-yield elastic–plastic constitutive rate equation did not amplify the initial uncertainty as much as the pre-yield elastic constitutive rate equation did. This can be easily viewed from Fig. 105.12 where the post-yield elastic-plastic evolution of PDF of shear stress was compared with fictitious extension of elastic evolution of PDF. Comparing the PDF of shear stress at $\epsilon_{12} = 0.0804\%$ (which is equivalent to $t = 0.01489s$), one can conclude that the variance of predicted elastic-plastic shear stress is much smaller (i.e. prediction is less uncertain) as compared to the same if the material were modeled as completely elastic.

Fig. 105.13 compares the evolution of means and standard deviations of predicted shear stress obtained using FPK equation approach and transformation method (pre-yield behavior)/Monte-Carlo approach (post-yield behavior). Although in the pre–yield response the FPK equation approach over-predicted the evolution of standard deviations because of reasons discussed earlier, in the post-yield

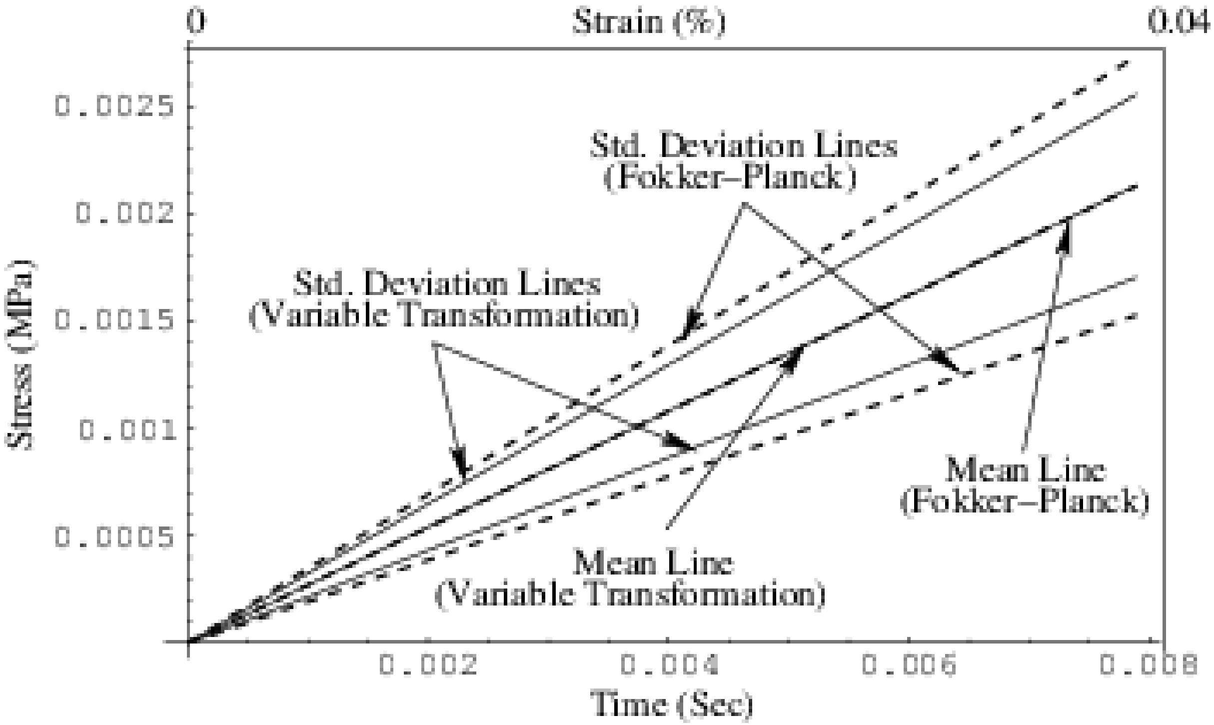


Figure 105.6: Comparison of Mean and Standard Deviation of Shear Stress for Elastic Constitutive Rate Equation with Random Shear Modulus (Problem–I) for FPE Solution and Variable Transformation Method Solution.

response it matched closely at regions further from the yielding region. The somewhat larger difference between FPK equation solution and the verification one (Monte Carlo solution) close to the yielding region is attributed to the fact that the initial condition for solution of post-yield elastic-plastic FPK equation was obtained from the solution of pre-yield elastic FPK equation. One way to better predict the overall probabilistic elastic-plastic behavior, would probably be to obtain the pre-yield elastic behavior through the transformation method and then use the FPK approach to predict post-yield elastic-plastic behavior.

105.2.11 Problem III

In this problem, the pre–yield linear elastic part is deterministic, however, at yield there is a distribution (with very small standard deviation) in shear stress due to assumed distribution in yield parameter α. The distribution in shear stress corresponds to the PDF of the random variable αI_1 (first invariant of the stress tensor or mean confining stress) and is assumed to be deterministic. This PDF of shear stress at yield was assumed to be the initial condition for the solution of post-yield elastic-plastic FPK equation

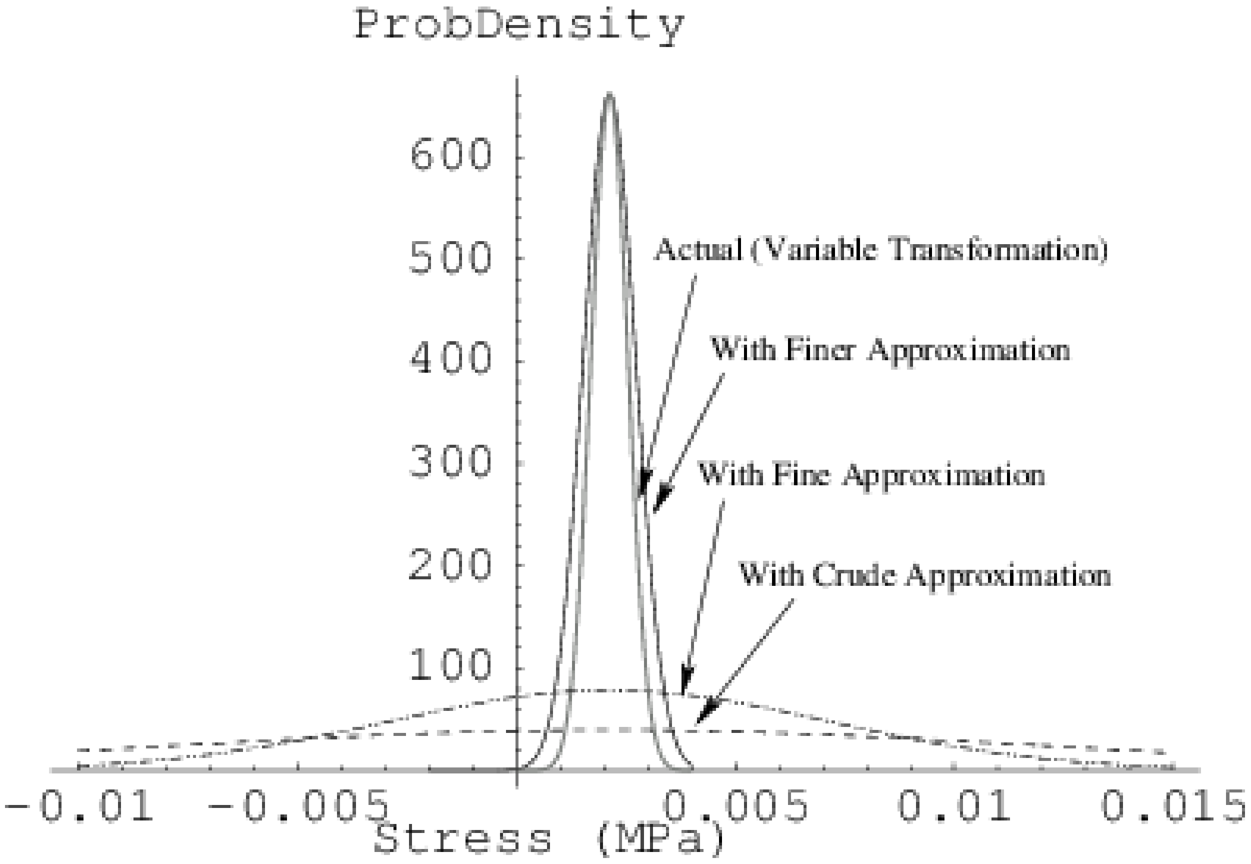


Figure 105.7: Effect of Approximating Function of Dirac Delta Initial Condition : PDF of Stress at Yield for Different Approximation of Initial Condition with Actual (Variable Transformation Method) Solution).

and is shown in Fig. 105.14.

The evolution of PDF for shear stress versus strain (time) is shown in Fig. 105.15. In addition to that the contours (including mean and standard deviation) of the evolution of PDF for shear stress versus strain (time) are shown in Fig. 105.16.

Looking at Fig. 105.16 and comparing the slopes of evolution of mean and standard deviation, one can conclude that the elastic-plastic evolution process didn't amplify the initial uncertainty in yield strength significantly. The initial (at yield) probability density function of shear stress just advected into the domain during the elastic–plastic evolution process without diffusing much. Fig. 105.15 clearly shows this advection process. The evolution of mean and standard deviations of shear stress obtained from the FPK equation approach was compared with those obtained from the Monte Carlo simulation and is shown in Fig. 105.17.

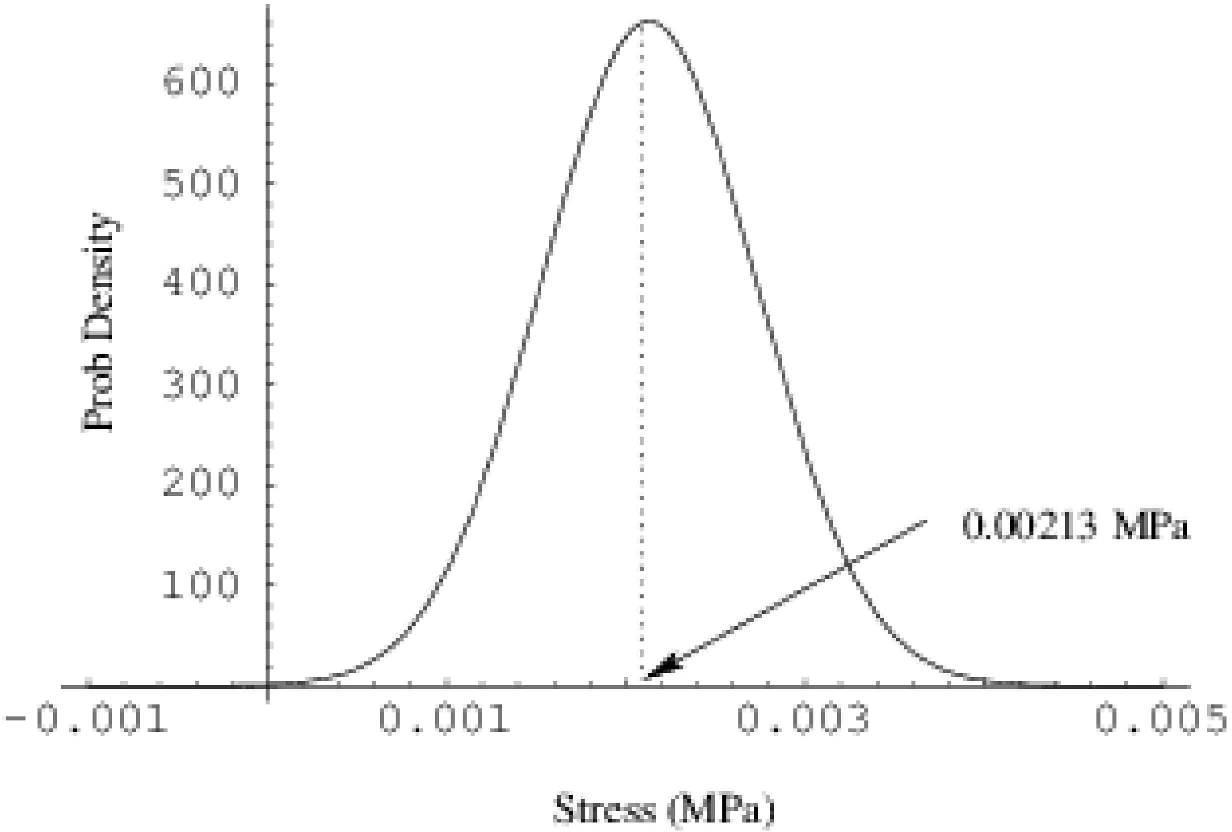


Figure 105.8: Initial condition for FPK equation for elastic–plastic zone (Problem–II).

105.3 Probabilistic Yielding and Cyclic Loading, 1D FPK Formulation

Modeling of geomaterials is inherently uncertain. These uncertainties stem from natural variability of geomaterials (spatial uncertainty), and testing and transformation errors (point uncertainty) (Lacasse and Nadim Lacasse and Nadim (1996), Phoon and Kulhawy Phoon and Kulhawy (1999a)). These uncertainties not only affect the failure characteristics of geomaterials, but also the behavior of geostructures, made with geomaterials. Traditionally, geotechnical engineering community deals with uncertainties in geomaterial by applying (large) factor of safety. However, use of large factors of safety results not only in over-expensive design, but also, sometimes, in unsafe structures (cf. Duncan Duncan (2000b)). Hence, in recent years, the geotechnical community has seen an increasing emphasis on probabilistic characterization of soil and subsequent reliability-based design.

One of the important aspects of probabilistic geomechanics simulation that has received less attention is the probabilistic constitutive problem. Among the few published papers were those by Fenton and Griffiths (Fenton and Griffiths (2002), Fenton and Griffiths (2003), Fenton and Griffiths (2005)) on

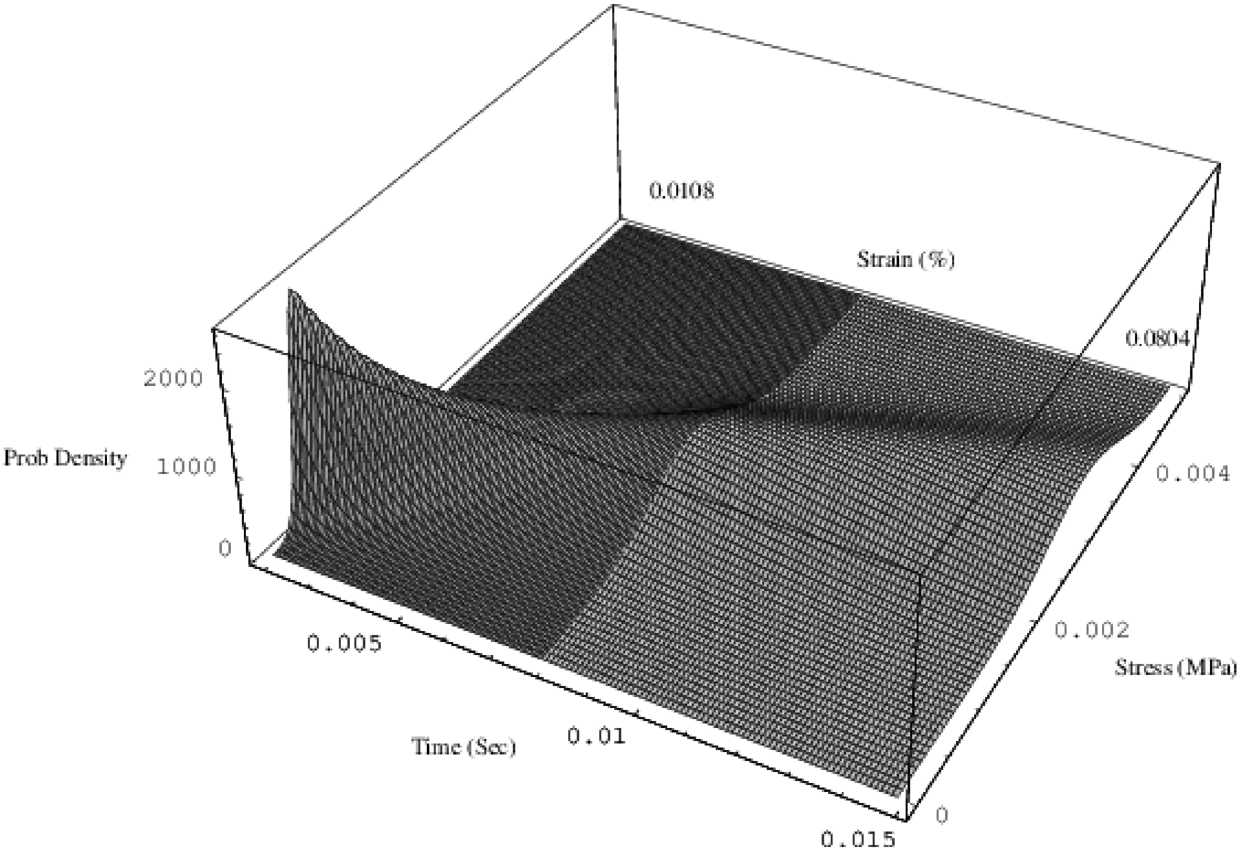


Figure 105.9: Evolution of PDF of shear stress versus strain (time) for elastic-plastic material with random shear modulus (Problem–II). View 1.

probabilistic simulation of spatially random c-ϕ soil using Monte Carlo technique, and those by Anders and Hori (Anders and Hori (1999), Anders and Hori (2000)) on probabilistic simulation of von Mises elastic-perfectly plastic material using perturbation technique. Both Monte Carlo and perturbation techniques have their inherent drawbacks (Matthies et al. Matthies et al. (1997), Keese Keese (2003)) and in dealing with those, recently, Jeremić et al. Jeremić et al. (2007b) proposed Eulerian–Lagrangian form of Fokker–Planck–Kolmogorov equation (FPKE) approach (cf. Kavvas Kavvas (2003)) to modeling and simulation for probabilistic elasto–plasticity. FPKE approach to probabilistic elasto–plasticity not only overcomes the drawbacks associated with other probabilistic simulation techniques, but also is fully compatible with the incremental theory of elasto–plasticity, and hence can easily be applied to probabilistic modeling and simulation of different elastic–plastic constitutive models. Solution strategies for FPK partial differential equation, corresponding to elastic–plastic constitutive rate equation and simulated probabilistic stress-strain responses under monotonic loading, assuming mean stress yielding, were discussed by Sett et al. (Sett et al. (2007c), Sett et al. (2007d)) for both linear and non-linear

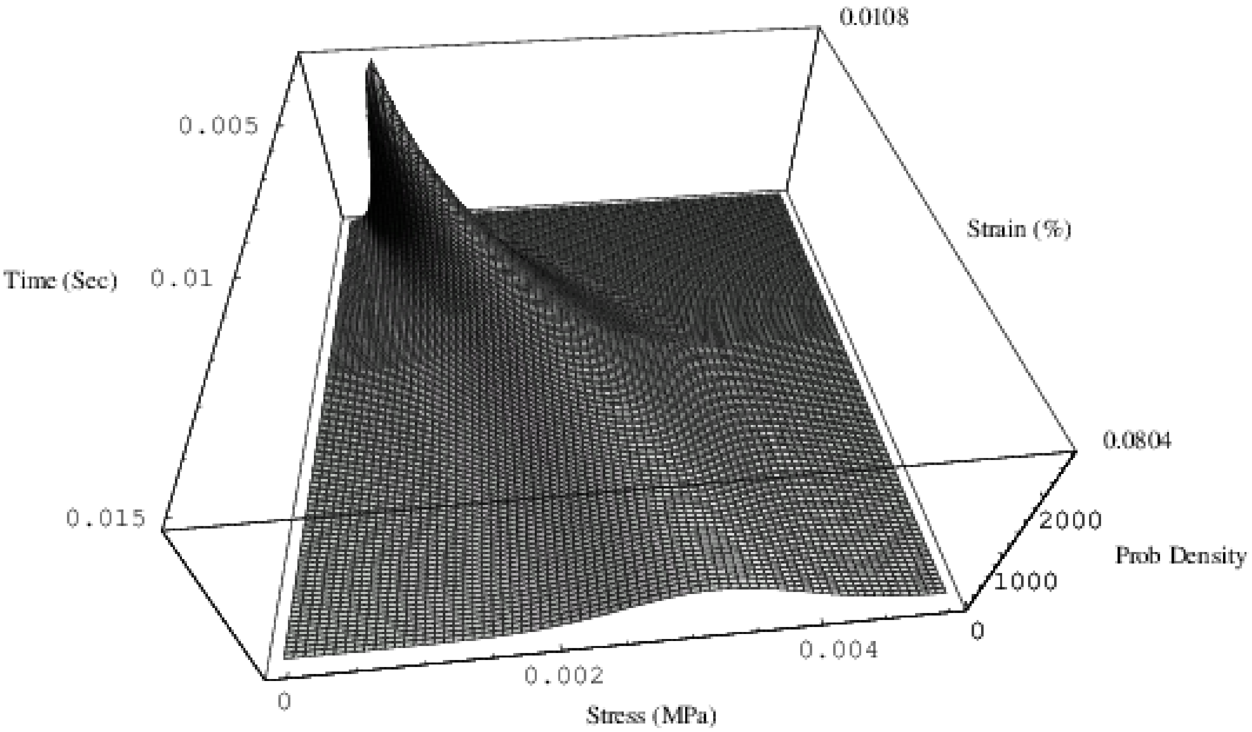


Figure 105.10: Evolution of PDF of shear stress versus strain (time) for elastic-plastic material with random shear modulus (Problem-II). View 2.

hardening models. The concept of probabilistic yielding was introduced and its effect on constitutive simulation under monotonic loading was discussed by Jeremić and Sett Jeremić and Sett (2009b). It was shown that due to uncertainty in yield function (stress), there is always a possibility, depending upon the magnitude of uncertainty, that plastic behavior starts at very very low strain and influence of elastic behavior continues far into plastic domain (at large strains) and hence, the ensemble average (mean) of all the possibilities or the most probable (mode) possibility differ from deterministic behavior. In addition to that, a very realistic, smooth transition between elastic and plastic domains was observed even for elastic perfectly plastic models. Further, nonlinear behavior was observed even for linear hardening models.

In this paper, the concept of probabilistic yielding is extended to 1–D cyclic simulations of geomaterials. Both elastic–perfectly plastic and hardening-type material model are considered. The numerical technique of solving FPKE cyclically with probabilistic yielding is discussed. Simulated responses were discussed in terms of probability density function (PDF) and its statistical moments.

Modeling of geomaterials is inherently uncertain. This uncertainty stems from natural variability of geomaterials (spatial uncertainty), and testing and transformation errors (point uncertainty) (Lacasse

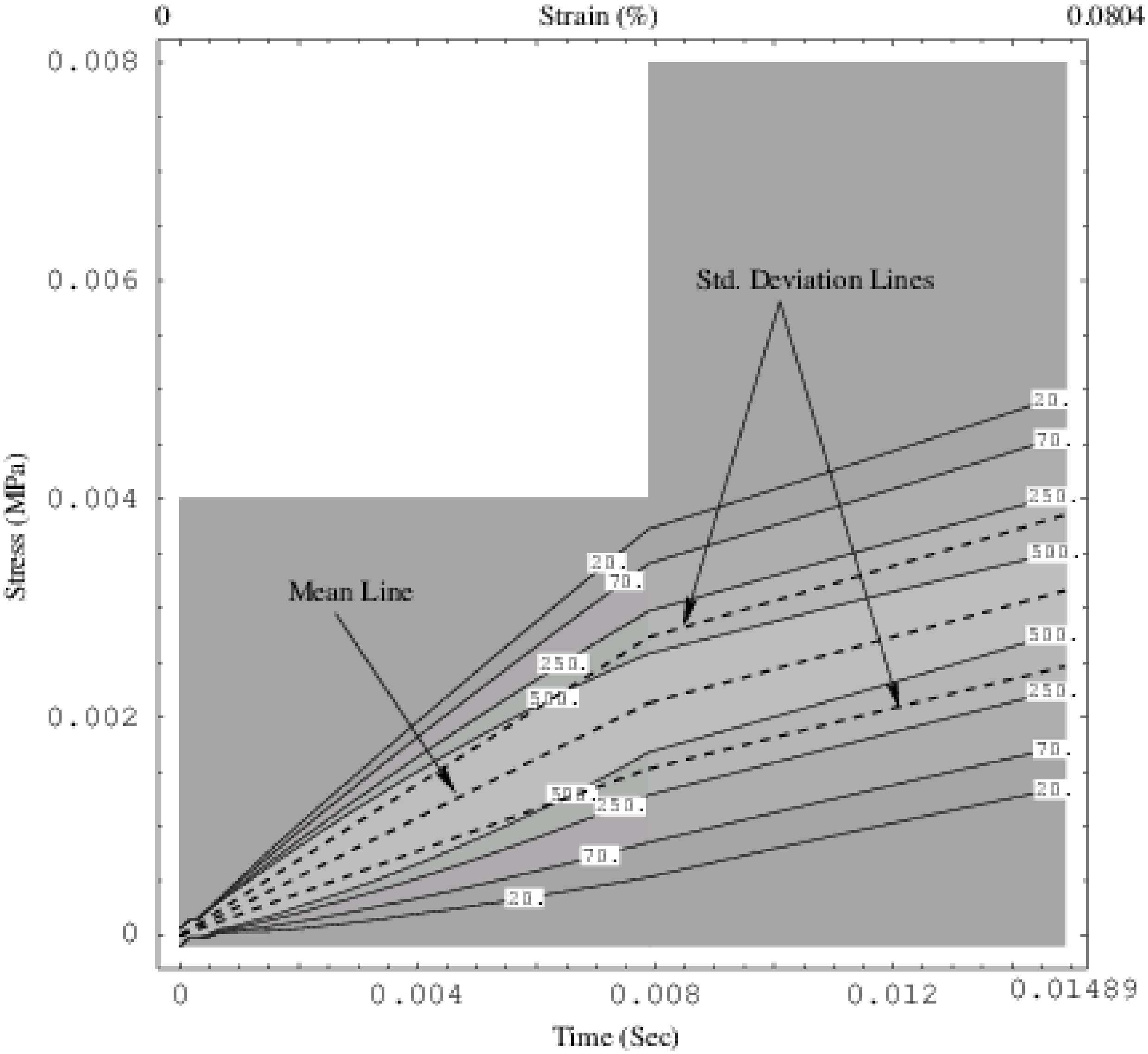


Figure 105.11: Contour of evolution of PDF for shear stress versus strain (time) for elastic-plastic material with random shear modulus (Problem–II).

and Nadim Lacasse and Nadim (1996), Phoon and Kulhawy Phoon and Kulhawy (1999a)). These uncertainties not only affect the failure characteristics of geomaterials, but also the behavior of geostructures, made with geomaterials. Traditionally, geotechnical engineering community deals with uncertainties in geomaterial by applying (large) factor of safety. However, use of large factors of safety results not only in over-expensive design, but also, sometimes, in unsafe structures (cf. Duncan Duncan (2000b)). Hence, in recent years, the geotechnical community has seen an increasing emphasis on probabilistic characterization of soil and subsequent reliability-based design.

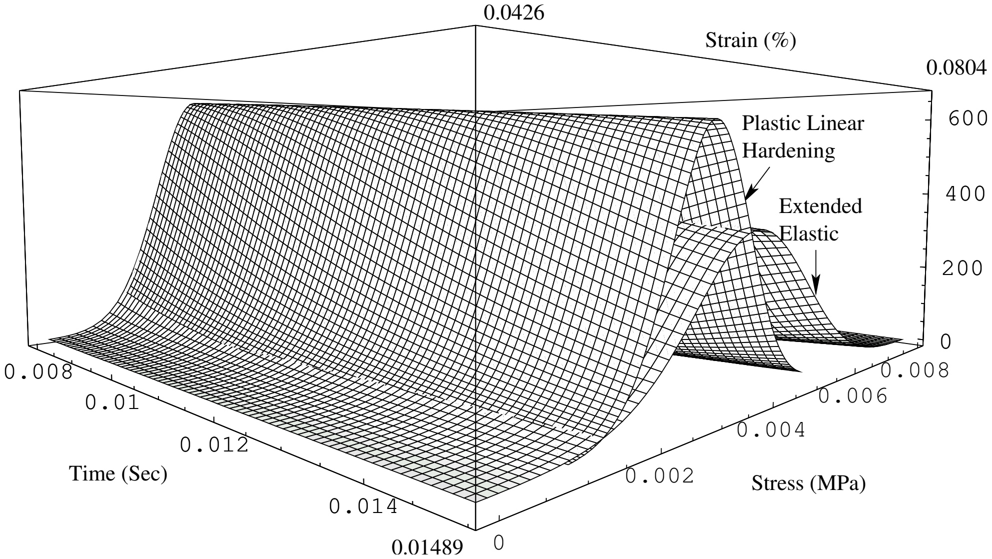


Figure 105.12: Comparison of evolution of PDF for elastic-plastic material and extended elastic material cases for random shear modulus.

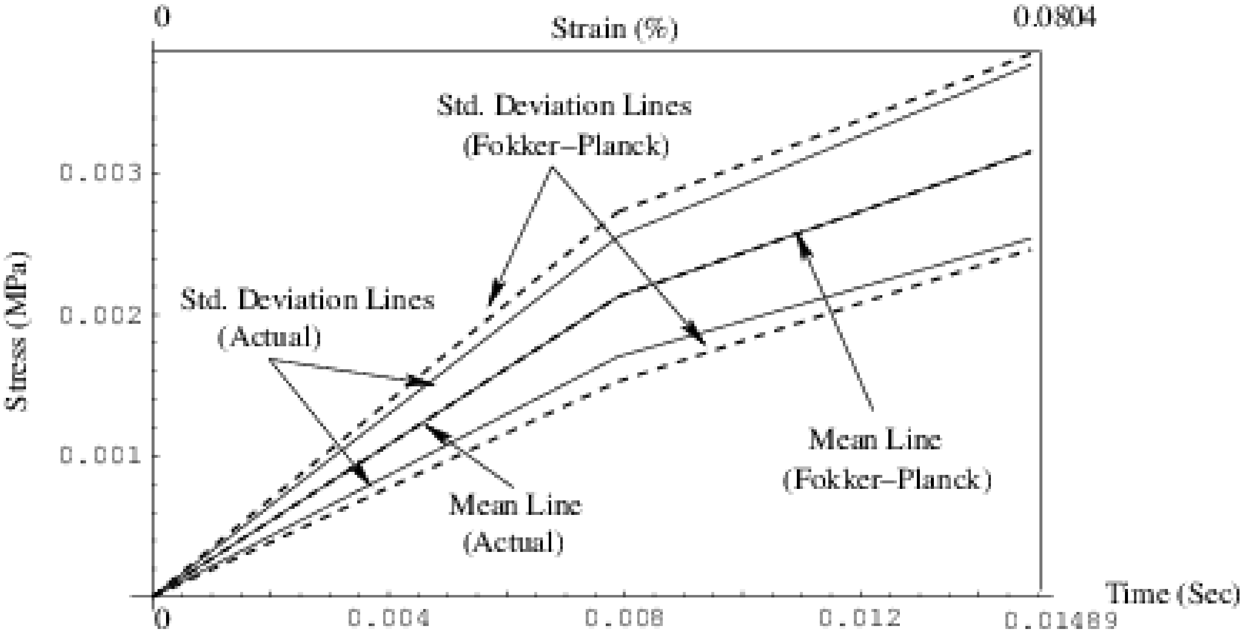


Figure 105.13: Comparison of mean and standard deviation of shear stress for plastic constitutive rate equation with random shear modulus (problem-ii) for FPK equation solution and Monte Carlo simulation solution.

One of the important aspects of probabilistic geomechanics simulation that has received less attention is the probabilistic constitutive problem. Among the few published papers were those by Fenton and Griffiths (Fenton and Griffiths (2002), Fenton and Griffiths (2003), Fenton and Griffiths (2005)) on probabilistic simulation of spatially random c-ϕ soil using Monte Carlo technique, and those by Anders and Hori (Anders and Hori (1999), Anders and Hori (2000)) on probabilistic simulation of von Mises elastic-perfectly plastic material using perturbation technique. Both Monte Carlo and perturbation techniques have their inherent drawbacks (Matthies et al. Matthies et al. (1997), Keese Keese (2003)) and in dealing with those, recently, Jeremić et al. Jeremić et al. (2007b) proposed Eulerian–Lagrangian form of Fokker–Planck–Kolmogorov equation (FPKE) approach (cf. Kavvas Kavvas (2003)) to modeling and simulation for probabilistic elasto–plasticity. FPKE approach to probabilistic elasto–plasticity not only overcomes the drawbacks associated with other probabilistic simulation techniques, but also is fully compatible with the incremental theory of elasto–plasticity, and hence can easily be applied to probabilistic modeling and simulation of different elastic–plastic constitutive models. Solution strategies for FPK partial differential equation, corresponding to elastic–plastic constitutive rate equation and simulated probabilistic stress-strain responses under monotonic loading, assuming mean stress yielding, were discussed by Sett et al. (Sett et al. (2007c), Sett et al. (2007d)) for both linear and non-linear

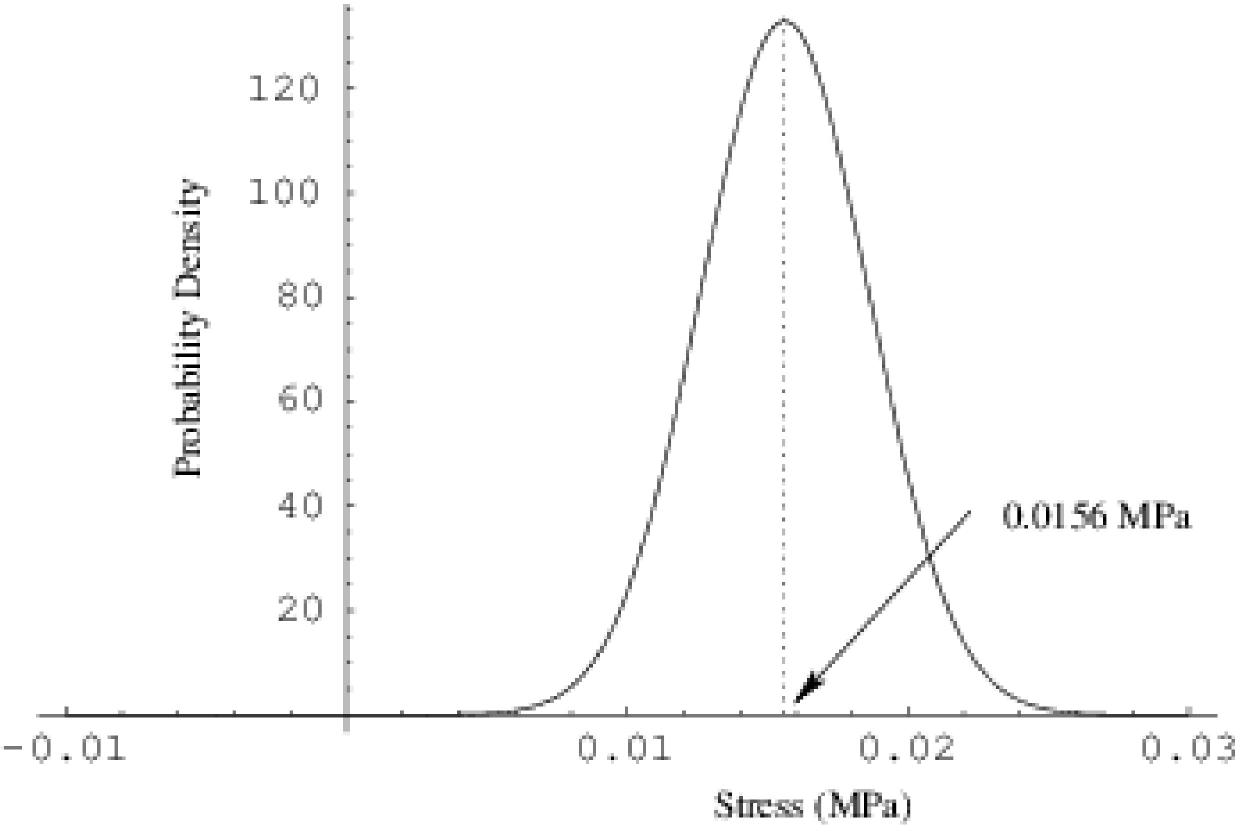


Figure 105.14: Initial condition for FPK equation for elastic–plastic material with random yield strength (Problem–III).

hardening models. The concept of probabilistic yielding was introduced and its effect on constitutive simulation under monotonic loading was discussed by Jeremić and Sett Jeremić and Sett (2009b). It was shown that due to uncertainty in yield function (stress), there is always a possibility, depending upon the magnitude of uncertainty, that plastic behavior starts at very very low strain and influence of elastic behavior continues far into plastic domain (at large strains) and hence, the ensemble average (mean) of all the possibilities or the most probable (mode) possibility differ from deterministic behavior. In addition to that, a very realistic, smooth transition between elastic and plastic domains was observed even for elastic perfectly plastic models. Further, nonlinear behavior was observed even for linear hardening models.

In this paper, the concept of probabilistic yielding is extended to 1–D cyclic simulations of geomaterials. Both elastic–perfectly plastic and hardening-type material model are considered. The numerical technique of solving FPKE cyclically with probabilistic yielding is discussed. Simulated responses were discussed in terms of probability density function (PDF) and its statistical moments.

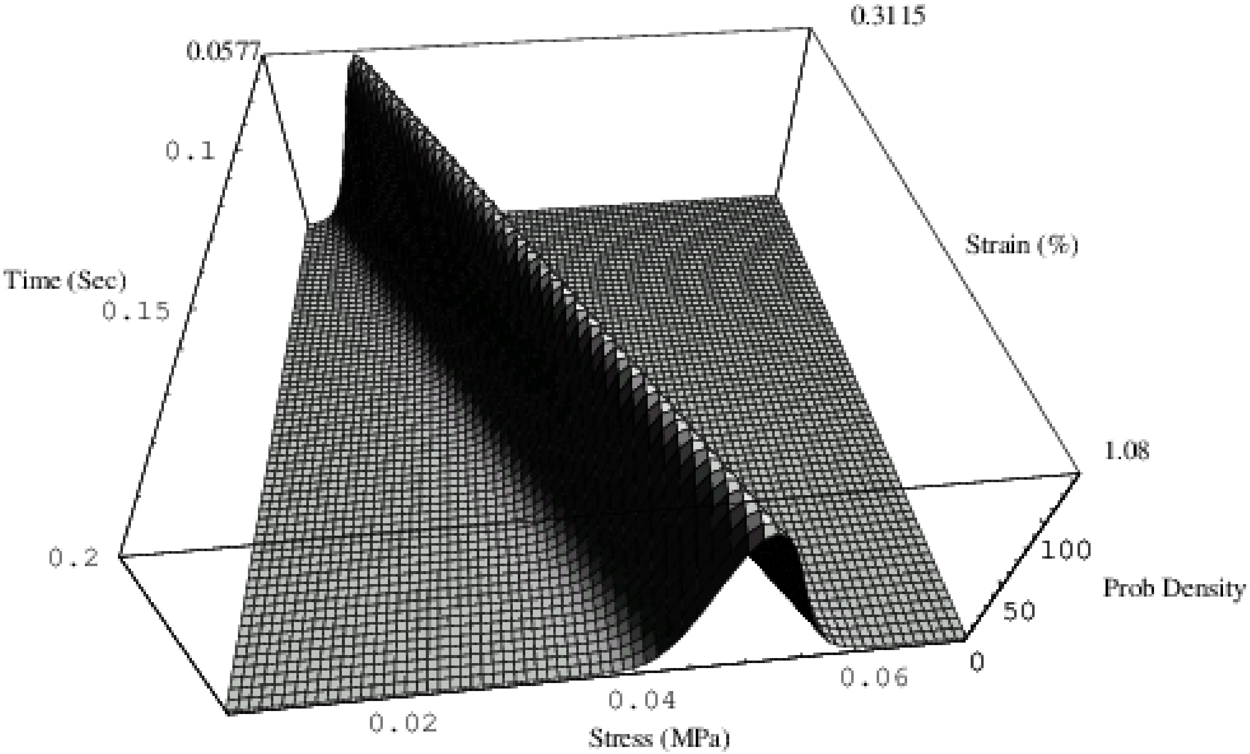


Figure 105.15: Evolution of PDF for shear stress for elastic–plastic material with random yield strength (Problem–III) (only plastic zone is shown).

105.3.1 Fokker–Planck–Kolmogorov Approach to Probabilistic Elasto–Plasticity

The Eulerian–Lagrangian form Fokker–Planck–Kolmogorov equation (cf. Kavvas Kavvas (2003)) corresponding to generalized 1–D constitutive rate equation can be written as (Jeremić et al. Jeremić et al. (2007b)):

$$
\begin{aligned}
&\frac{\partial P(\sigma(x_t,t),t)}{\partial t} = \\
&\quad \frac{\partial}{\partial \sigma}\left[\left\{\left\langle \eta(\sigma, D, \epsilon; x_t, t)\right\rangle + \int_0^t d\tau Cov_0\left[\frac{\partial \eta(\sigma, D, \epsilon; x_t, t)}{\partial \sigma}; \eta(\sigma, D, \epsilon; x_{t-\tau}, t-\tau)\right]\right\} P(\sigma(x_t,t),t)\right] \\
&\quad + \frac{\partial^2}{\partial \sigma^2}\left[\left\{\int_0^t d\tau Cov_0\left[\eta(\sigma, D, \epsilon; x_t, t); \eta(\sigma, D, \epsilon; x_{t-\tau}, t-\tau)\right]\right\} P(\sigma(x_t,t),t)\right]
\end{aligned}
\tag{105.43}
$$

where, $P(\sigma(x_t,t),t)$ is the probability density of stress (σ) at (pseudo) time t, and η is the operator variable, obtained by collecting together all the operators and variables on the r.h.s of the generalized constitutive rate equation:

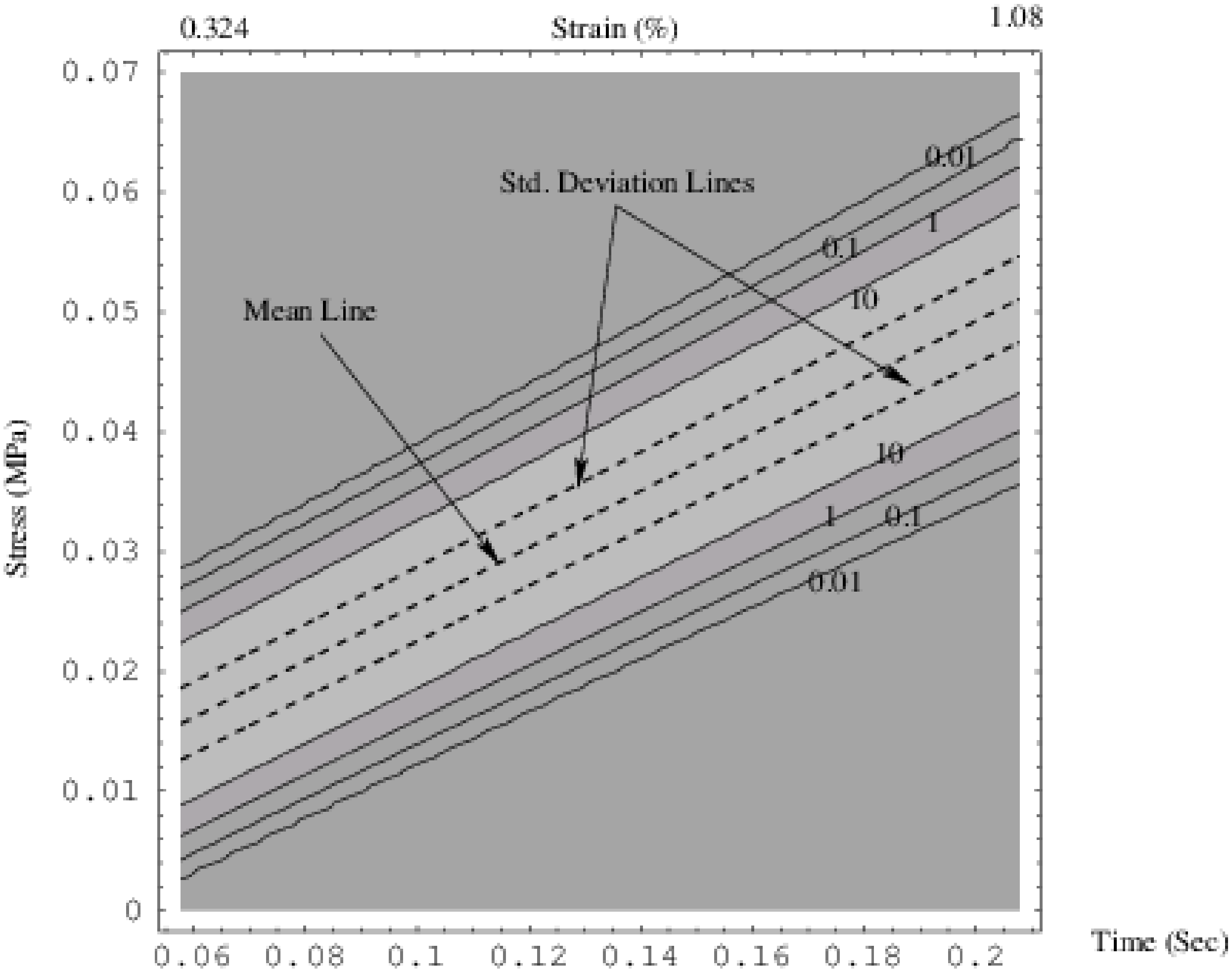


Figure 105.16: Contour PDF for shear stress versus strain (time) for elastic–plastic material with random yield strength (Problem–III).

$$\frac{d\sigma(x_t,t)}{dt} = \eta(\sigma, D, \epsilon; x_t, t) \tag{105.44}$$

In Eq. (105.66), ϵ is the strain, and D is the tangent modulus, which could be elastic or elastic–plastic:

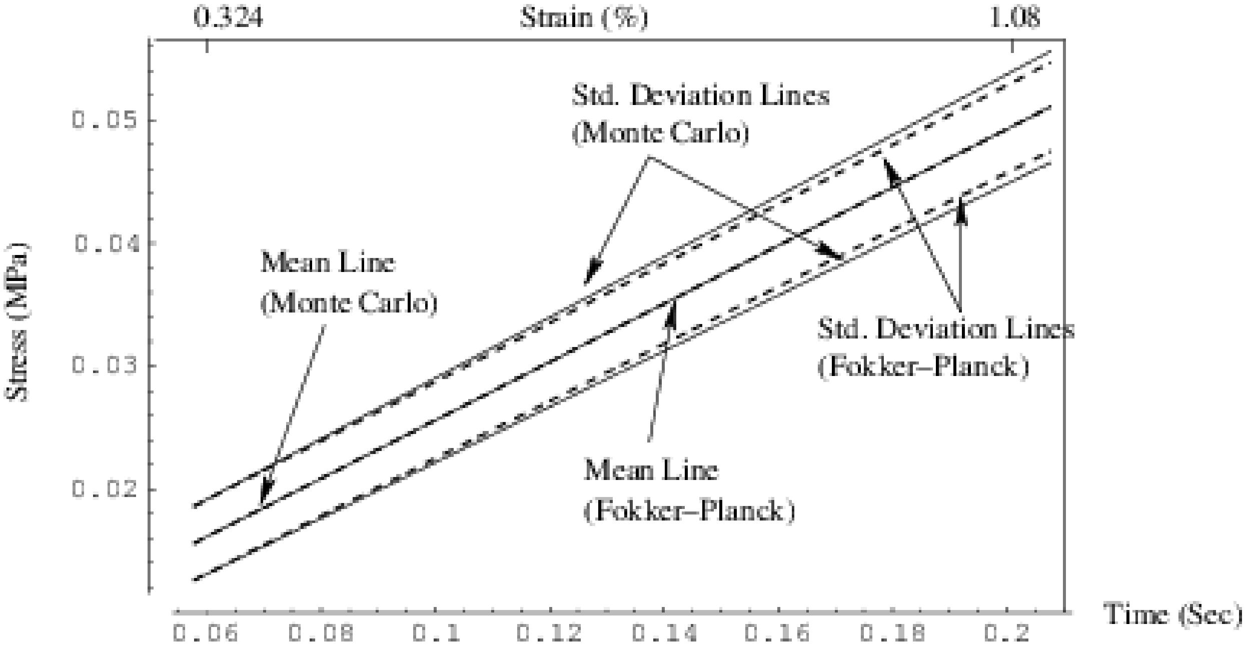


Figure 105.17: Comparison of Mean and Standard Deviation of Shear Stress for Elastic-Plastic Constitutive Rate Equation with Random Yield Strength (Problem-III) for FPE Solution and Monte Carlo Simulation Solution

$$D = \begin{cases} D^{el} & \text{elastic} \\ \\ D^{el} - \dfrac{D^{el} \dfrac{\partial U}{\partial \sigma} \dfrac{\partial f}{\partial \sigma} D^{el}}{\dfrac{\partial f}{\partial \sigma} D^{el} \dfrac{\partial U}{\partial \sigma} - \dfrac{\partial f}{\partial q_*} r_*} & \text{elastic-plastic} \end{cases} \tag{105.45}$$

where, D^{el}, f, U, q_*, and r_* are elastic modulus, yield surface, plastic potential surface, internal variable(s), and rate(s) of evolution of internal variable(s) respectively.

Eq. (105.65) is the most general form of elastic–plastic constitutive rate equation, written in probability density space. This equation (Eq. (105.65)) can be written in a more compact form:

$$\frac{\partial P(\sigma(x_t,t),t)}{\partial t} = \frac{\partial}{\partial \sigma} \left\{ N_{(1)} P(\sigma(x_t,t),t) \right\} + \frac{\partial^2}{\partial \sigma^2} \left\{ N_{(2)} P(\sigma(x_t,t),t) \right\} \tag{105.46}$$

where, $N_{(1)}$ and $N_{(2)}$ are advection and diffusion coefficients respectively, and are material model specific. By specializing Eq. (105.46) to (any) particular constitutive model, the resulting FPKE can be solved to obtain the probability density function of stress response, given uncertainties in material properties and driving strain. However, difference in material behavior in elastic and elastic-plastic regions necessities solution of FPKE twice - one corresponding to elastic constitutive equation (with $N^{el}_{(1)}$ and $N^{el}_{(2)}$, the advection and diffusion coefficients corresponding to elastic constitutive equation) and the other corresponding to elastic–plastic constitutive equation (with $N^{ep}_{(1)}$ and $N^{ep}_{(2)}$, the advection and diffusion coefficients corresponding to elastic–plastic constitutive equation). The switch from elastic to elastic–plastic region (solution) can be controlled using mean stress yielding:

$$\begin{array}{lll} \text{if} & \langle f \rangle < 0 \vee \left(\langle f \rangle = 0 \wedge d \langle f \rangle < 0 \right) & \text{use elastic FPKE} \\ \text{or, if} & \langle f \rangle = 0 \vee d \langle f \rangle = 0 & \text{use elastic–plastic FPKE} \end{array} \tag{105.47}$$

However, difficulty arises if the material yield parameter(s) are uncertain, as the mean yield criteria then does not account for the complete probabilistic yielding of material. For example, such mean yielding will neglect the possibilities of elastic–plastic behavior in the elastic region and vice versa. The concept of probabilistic yielding overcomes this limitation, as it solves Eq. (105.46) once, with equivalent advection and diffusion coefficients, $N^{eq}_{(1)}$ and $N^{eq}_{(2)}$ (Jeremić and Sett Jeremić and Sett (2009b)):

$$\begin{array}{l} N^{eq}_{(1)}(\sigma) = (1 - P[\Sigma_y \leq \sigma])N^{el}_{(1)} + P[\Sigma_y \leq \sigma]N^{ep}_{(1)} \\ N^{eq}_{(2)}(\sigma) = (1 - P[\Sigma_y \leq \sigma])N^{el}_{(2)} + P[\Sigma_y \leq \sigma]N^{ep}_{(2)} \end{array} \tag{105.48}$$

where $(1 - P[\Sigma_y \leq \sigma])$ represents the probability of material being elastic, while $P[\Sigma_y \leq \sigma]$ represents the probability of material being elastic–plastic. The probabilities of material being elastic and the probabilities of material being elastic–plastic can easily be calculated from the cumulative density function of yield function (stress).

It is worth noting that the probabilistic yield criterion (Eq. (105.48)) represents probabilistic restatement of the deterministic yield criteria. The probabilistic yield criteria is introduced (or, the deterministic yield criteria is written in probability space) in order to properly model uncertain (probabilistic) yield strength.

It is also very interesting to note that proposed approach for calculating equivalent advection and diffusion coefficients is similar to the solution strategy of famous Black–Scholes Black and Scholes (1973) equation in financial engineering modeling of European option, where probabilities of exercise of the (European) option, obtained from cumulative density functions, are multiplied with stock price and present value of option strike price to calculate the option price.

105.3.2 Elastic–Perfectly Plastic Material

In this section, the FPKE–approach, along with the concept of probabilistic yielding, is applied to simulate 1–D (shear stress–shear strain) cyclic behavior of elastic–perfectly plastic material. Only von Mises material model has been considered. It may, however, be noted that presented development is general enough to be used with any material model and that von Mises is just one such model we use for illustration purposes.

The von Mises yield criteria can be written as:

$$\sqrt{J_2} - k = 0 \tag{105.49}$$

where, k is a material parameter (yield strength like) and $J_2 = 3/2 s_{ij} s_{ij}$ is the second invariant of deviatoric stress tensor $s_{ij} = \sigma_{ij} - 1/3\sigma_{kk}\delta_{ij}$. For 1–D shear, Eq. (105.49) becomes:

$$|\sigma| - \sigma_y = 0 \quad \text{or} \quad \sigma = \pm\sigma_y \tag{105.50}$$

The yielding occurs at a yield stress of $\pm\sigma_y$. It, however, is important to note that both σ_y and σ are uncertain and are described by their respective probability density functions. For elastic–perfectly plastic material, the distribution of yield stress (σ_y) is given by its experimentally measured initial distribution, and remains constant. The stress (σ), however, evolves according to the governing FPKE (Eq. (105.46)) and its distribution is given by the solution of the governing FPKE (Eq. (105.46)). For 1–D von Mises elastic–perfectly plastic shear constitutive model, the elastic and the elastic–plastic advection and diffusion coefficients of the governing FPKE (Eq. (105.46)), becomes:

$$\begin{aligned} N^{el}_{(1)} &= \frac{d\epsilon_{xy}}{dt}\langle G\rangle \quad ; \quad N^{el}_{(2)} = t\left(\frac{d\epsilon_{xy}}{dt}\right)^2 Var[G] \\ N^{ep}_{(1)} &= 0 \quad ; \quad N^{ep}_{(1)} = 0 \end{aligned} \tag{105.51}$$

where, G is the shear modulus, $d\epsilon_{xy}$ is the (deterministic) incremental shear strain, t is the pseudo time, $\langle\cdot\rangle$ represents expectation operation and $Var[\cdot]$ represents variance operation. The equivalent advection and diffusion coefficients (refer Eq. (105.48)) for von Mises elastic–perfectly plastic material, then, becomes:

$$\begin{aligned} N^{eq}_{(1)}(\sigma) &= (1 - P[\Sigma_y \leq \sigma])\frac{d\epsilon_{xy}}{dt}\langle G\rangle \\ N^{eq}_{(2)}(\sigma) &= (1 - P[\Sigma_y \leq \sigma])t\left(\frac{d\epsilon_{xy}}{dt}\right)^2 Var[G] \end{aligned} \tag{105.52}$$

One may note that, in deriving the elastic and elastic–plastic advection and diffusion coefficients (Eq. (105.51)), it was assumed that spatial random field material properties (G, and σ_y) would be first discretized into random variables, for example at Gauss points, by appropriate tools, for example Karhunen–Loève expansion (Karhunen Karhunen (1947), Loève Loève (1948), Ghanem and Spanos Ghanem and Spanos (1991)). In other words, the solution of FPKE, with advection and diffusion coefficients given by Eq. (105.52), represents point–location scale von Mises elastic–perfectly plastic material behavior, and not the local–average material behavior. The local–average material behavior, if sought for, can then be assembled using polynomial chaos expansion (Wiener Wiener (1938), Ghanem and Spanos Ghanem and Spanos (1991)).

105.3.2.1 Probability Density Function

The FPKE (Eq. (105.46)), with advection and diffusion coefficients given by Eq. (105.52), was solved incrementally with pseudo time steps using method of lines. The stress domain of the Fokker–Planck–Kolmogorov PDE was discretized first on a uniform grid by central differences, and thereby obtaining a series of ODE. The series of ODEs was then solved, after incorporating boundary conditions, simultaneously and incrementally, with n pseudo time steps, using a standard open–source ODE solver, SUNDIALS Hindmarsh et al. (2005), which utilizes ADAMS method and functional iteration.

The yield shear strength (σ_y) of the material was assumed to have a mean value of 60 kPa with a COV of 30%, values typical for clay (Federal Highway Administration Federal Highway Administration (2002), Lacasse and Nadim Lacasse and Nadim (1996)). Also, the yield shear strength was assumed to be either normal or Weibull (with shape parameter of 3.31 and scale parameter of 0.067) distribution as shown in Fig. 105.18. The shear modulus (G) was also assumed to be either normal or Weibull distribution, but

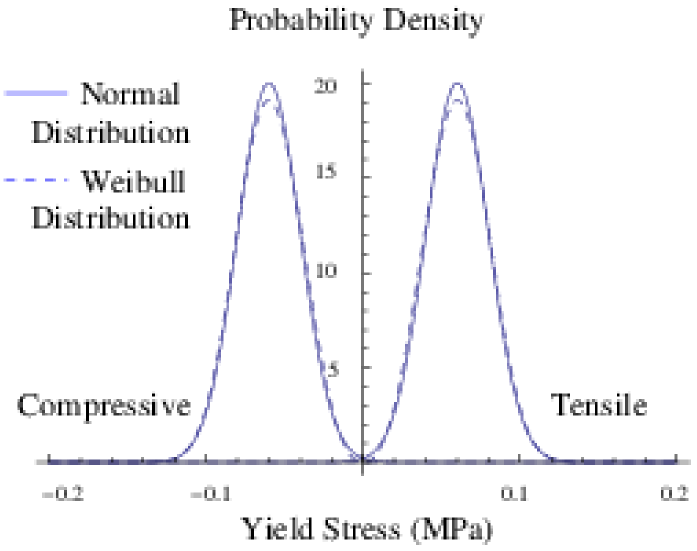


Figure 105.18: Elastic–perfectly plastic probabilistic model: PDF of yield stress

with a mean value of 100 MPa and a COV of 25%. The cyclic probabilistic von Mises, elastic–perfectly plastic shear stress–shear strain response (evolutionary probability density function (PDF) of shear stress), for the case where both yield shear strength (σ_y) and shear modulus (G) are normally distributed, is shown in Fig. 105.19. Two different views of the loading–unloading–reloading cycle are shown, focusing on the

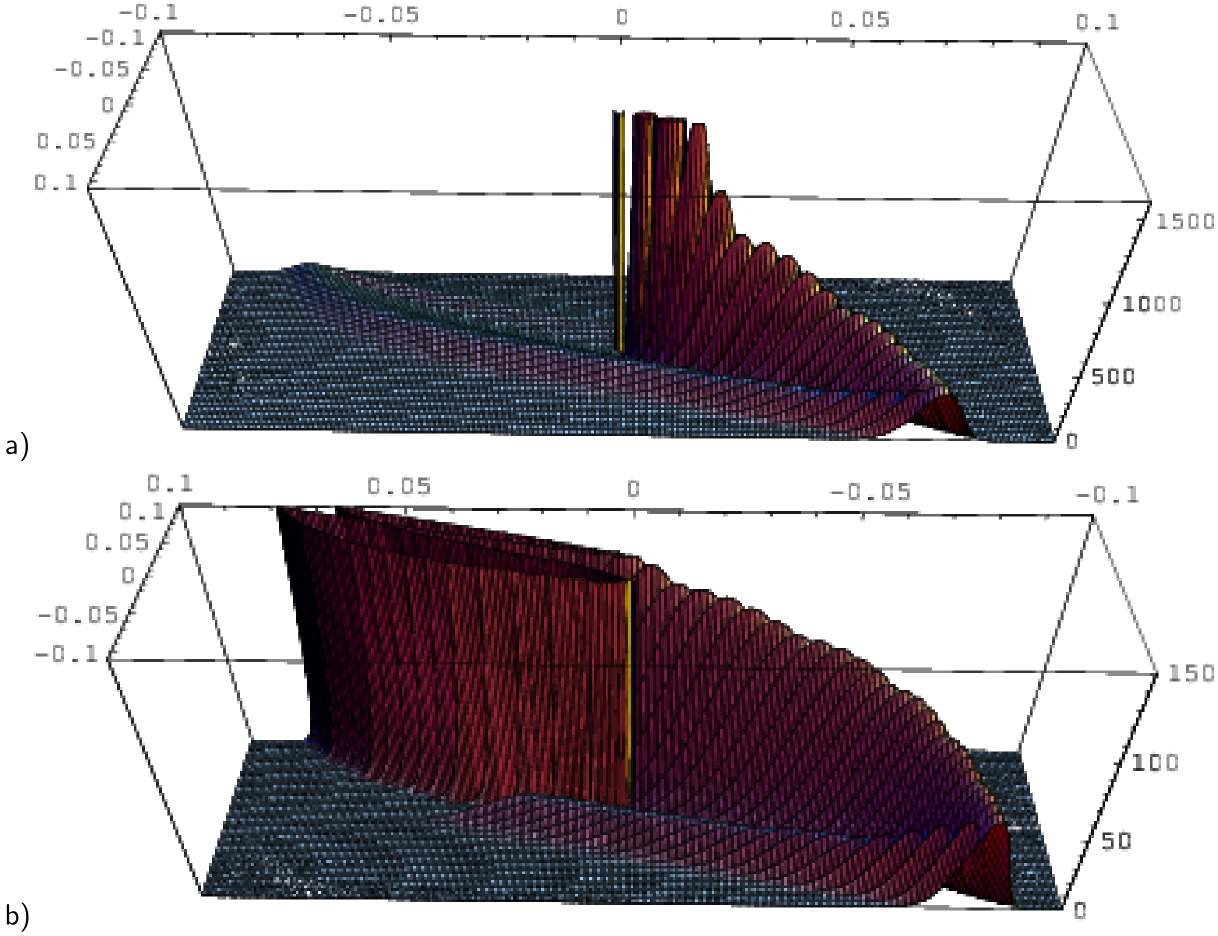


Figure 105.19: Elastic–perfectly plastic probabilistic model under cyclic loading: evolutionary PDF of shear stress (a) view from the junction of loading and unloading branches (probability densities of shear stress are truncated at a value 1500 for clarity of the plot) and (b) view from the junction of unloading and reloading branches (probability densities of shear stress are truncated at a value of 150 for clarity of the plot)

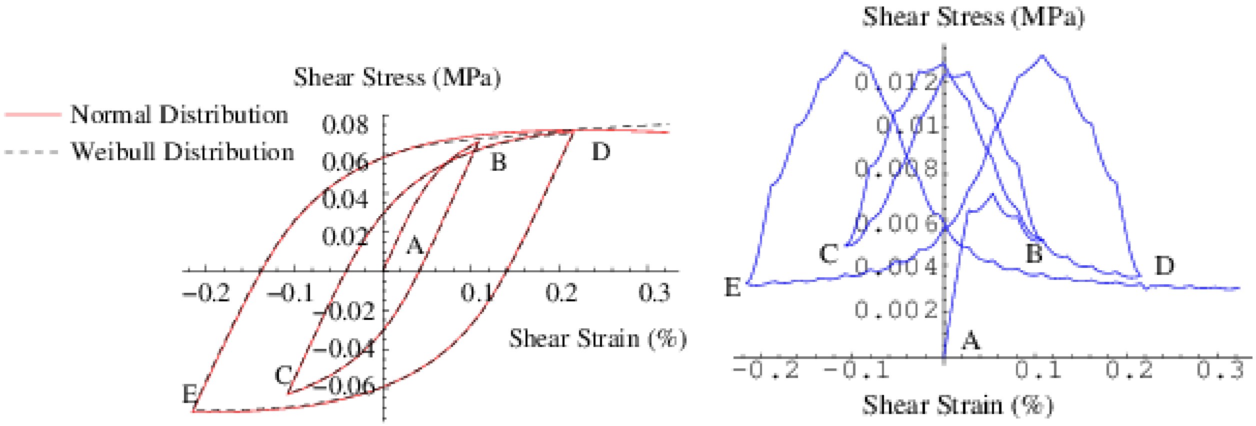
transition between loading and unloading, and unloading and reloading branches. As can be seen from Fig. 105.19, PDF for initial stress (a deterministic Dirac delta function at stress–strain origin) advected and diffused into the domain, governed by the advection and diffusion coefficients (Eq. (105.52)). It is very important to also note that, even–though the deterministic response for von Mises elastic–perfectly plastic material is bi–linear, due to introduced uncertainties in yielding, the probabilistic response is non–linear from the beginning. That is, due to uncertainty in yield strength, there is a (small) possibility that the material becomes elasto–plastic from the very beginning of loading. This possibility has been quantified from the PDF of the yield strength and taken into consideration implicitly during simulation using the equivalent advection and diffusion coefficients ($N^{eq}_{(1)}$ and $N^{eq}_{(2)}$, refer Eq. (105.52)). Those coefficients assigns probability weights to the realizations of stress response based on the probability of material being elastic or elastic–plastic. Initially, in the loading branch, at small strains, the probability of material being elastic–plastic is very small and hence, the initial probabilistic stress response (ensemble of all realizations) is closer (but not fully) to linear, elastic response. However, as strain increases, the probability of elastic–plastic material behaving increases and the probabilistic stress response gradually becomes more elastic–plastic (Fig. 105.19(a)).

Upon unloading, the material behaves as (mostly) elastic since elastic–plastic probability weights from the governing PDF of mirror image (negative) of shear strength (Fig. 105.18) are initially very small. During later stages of unloading (loading in the opposite direction), and similar to the loading branch, the elastic–plastic probability weights increase and gradually transition the response toward elasto–plasticity (Fig. 105.19(b)). Similar to this, in the subsequent reloading branch, the probability weights are again governed the PDF of (positive, loading branch of) shear strength (Fig. 105.18), and hence the probabilistic response is again initially more linear, elastic, while gradually it transitions to full elasto–plasticity.

105.3.2.2 Case of Increasing Strain Loops

In Fig. 105.20, the evolutionary PDF of shear stress for von Mises elastic–perfectly plastic material (refer Fig. 105.19) is plotted in terms of its statistical moments – the evolutionary mean (Fig. 105.20(a)), and standard deviation (Fig. 105.20(b)) of shear stress – for the first couple of cycles with increasing strain loops. The mean response, when both the yield shear strength (σ_y) and the shear modulus (G) are modeled as Weibull distribution, is also shown in Fig. 105.20(a). The oscillations in the evolution of standard deviation of shear stress with shear strain are due to step size issue, inherent to the forward Euler method that has been used in solving the FPKE. Work is underway to implement linearly implicit mid-point rule for solving the FPKE corresponding to elastic-plastic constitutive rate equation.

The very important observation that can be made using Fig. 105.20(a) is that, if one consider uncertainties in geomaterial properties, even the simplest elastic–perfectly model, captures some of the very important features of geomaterial behaviors. For example, reduction of (secant) modulus with cyclic strain, commonly observed in soil (cf. Vucetic and Dobry Vucetic and Dobry (1991)), is fairly nicely captured. If using deterministic models, this feature can only be somewhat successfully modeled with



(a) (b)

Figure 105.20: Elastic–perfectly plastic probabilistic model under cyclic loading with increasing strain loops: evolution of (a) mean and (b) standard deviation of shear stress

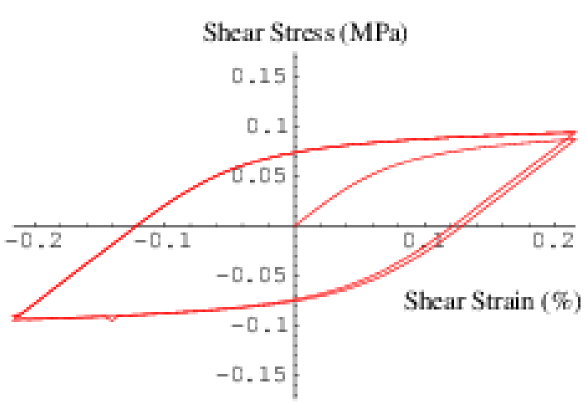
fairly complex models, which require many more parameters. It is important to remark that for our probabilistic modeling, (only) statistical distributions (probability density functions) of shear modulus (G) and shear strength (σ_y), are needed. Expansion of elastic–plastic modeling into probability space seems to have added significant new capabilities to modeling.

105.3.2.3 Case of Constant Strain Loops

This von Mises elastic–plastic material, however, didn't exhibit (secant) modulus degradation, commonly observed in clay (cf. Vucetic and Dobry Vucetic and Dobry (1988)), when the material is cycled repeatedly at the same strain. Fig. 105.21(a) shows such probabilistic response (mean of shear stress). The material was cycled repeatedly up to 0.2% strain. Only first three cycles are shown. It is important to note that the von Mises mean elastic–plastic material behavior is function of both the mean and standard deviation of both shear modulus (G) and yield shear strength (σ_y). The same von Mises elastic–perfectly plastic model with a different set of material properties could, however, be able to capture the degradation of mean (secant) shear modulus. For example, Japanese stiff clay, when modeled as von Mises elastic–perfectly plastic material, exhibited modulus degradation with number of cycles (Sett et al. Sett et al. (2008))

105.3.2.4 Monotonic Loading

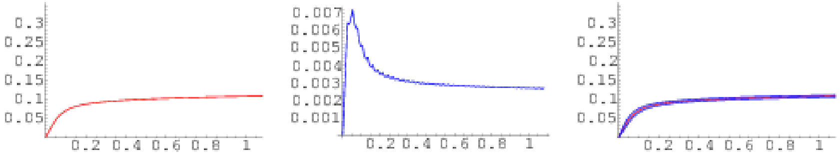
For completeness of comparison, the monotonic behavior of this probabilistic von Mises perfectly plastic material is also shown (refer Fig. 105.22). As can be observed from Fig. 105.22(a), the mean shear



(a) (b)

Figure 105.21: Elastic–perfectly plastic probabilistic model under cyclic loading with all equal loops: evolution of (a) mean and (b) standard deviation of shear stress

stress non–linearly increases with shear strain before reaching the perfectly plastic state.



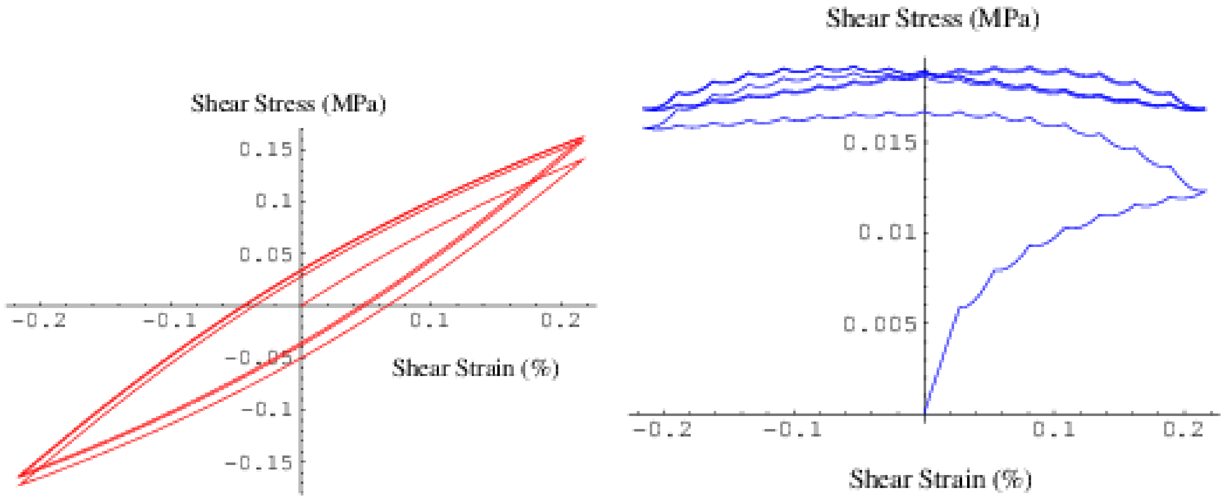
(a) (b)

Figure 105.22: Elastic–perfectly plastic probabilistic model under monotonic loading: evolution of (a) mean, (b) standard deviation, and (c) mean ± standard deviation of shear stress

Physically, one may visualize the probabilistic soil constitutive response as an ensemble of the behaviors of infinite number of soil particles in a representative volume element (RVE), for example, a laboratory soil specimen. The behavior of an individual soil particle in a RVE is governed, in case of elastic–perfectly plastic material, by its modulus and strength. However, if the modulus and strength of each particle are different, for example, governed by their respective PDF, then each particle would behave differently. The PDF of the response behavior then represents the ensemble of all such behaviors, with their respective probability weights. The mean, on the other hand, represents the ensemble average of all such behaviors. In this context, it is important to note that the behaviors presented in this paper do not take into account the correlation between soil particles (scale effect). The scale effect can be accounted for, among others, using stochastic elastic–plastic finite element technique. Sett Sett (2007) proposed one such finite element method by extending the spectral approach to stochastic finite element (cf. Ghanem and Spanos Ghanem and Spanos (1991)) to elastic–plastic problems by updating the material properties at Gauss integration points using the FPKE approach, as the material plastifies.

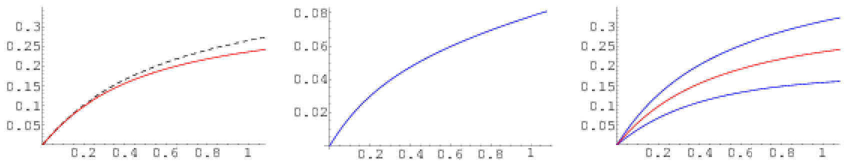
Further to the promise of an alternate approach to geomaterial modeling, probabilistic approach also quantifies our confidence in the simulated behavior of geomaterials. FPKE based probabilistic elasto–plasticity solves for second-order accurate evolutionary PDF of shear stress (Fig. 105.19). Ability to obtain the PDF of stress accurately is very important in failure simulation of geomaterials, as they often fail at low probabilities (tails of PDF). A full PDF contains enormous amount of information. From the PDF, other than the statistical moments, other useful engineering information, for example, the probability of exceedance, most probable solution, as well as some derivative application like sensitivity analysis can be easily obtained or derived. Figs. 105.20(b) and 105.21(b) show one of the important confidence measuring parameters, the evolutionary standard deviation of shear stress (square-root of second moment of the evolutionary PDF of shear stress (Fig. 105.19)), for cyclic responses with increasing loops and all equal loops, respectively. As can be observed from the above figures (Figs. 105.20(b) and 105.21(b)), inside any branch (loading, unloading, re–loading, re–unloading, ...), as well as in Fig. 105.22(b), where the monotonic response is shown, the standard deviation, first increases and then decreases. This is because, initially, when the material is mostly elastic, both the uncertainties in shear modulus (G) and yield strength (σ_y) are governing. As material becomes mostly elastic–plastic, the influence of uncertainty in shear modulus (G) decreases. However, it is important to note that this type of standard deviation response is not generic to all von Mises elastic–perfectly plastic material. The standard deviation response is very much dependent on the amount uncertainties present in both shear modulus (G) and yield strength (σ_y). For example, Fig. 105.23(b), shows probabilistic response of cyclic behavior of the same material model, except that COV of yield strength (σ_y), is now assumed to be 300%. The standard deviation response shown here is always increasing which is completely different from what was observed in previous case (Figs. 105.20(b), 105.21(b) and 105.22(b))). This is because, for this material, the COV of shear modulus (assumed 30%) is non–significant, compared to the COV of yield strength (assumed 300%), and hence, the standard deviation response (Fig. 105.23(b)) is predominantly influenced by the uncertainty in yield strength (σ_y). Similar standard deviation response can be observed in Fig. 105.24(b), where the material with large COV of yield strength was subjected to monotonic loading.

It is also interesting to compare Figs. 105.21(a) and 105.23(a). Both are mean responses of von Mises elastic–perfectly plastic material model with same material parameters, except with different COV of yield strength. COV of yield strength for simulation in Fig. 105.21(a) was 30% and that for simulation in Fig. 105.23(a) was 300%. It is observed that a completely different responses were obtained. The effect



(a) (b)

Figure 105.23: Elastic–perfectly plastic probabilistic model under cyclic loading with all equal loops (probabilistic model parameters are exactly the same as used for simulation in Fig. 105.21, but with very large yield uncertainty): evolution of (a) mean and (b) standard deviation of shear stress



(a) (b)

Figure 105.24: Elastic–perfectly plastic probabilistic model under monotonic loading (model parameters are exactly the same as used for simulation in Fig. 105.22, but with very large yield uncertainty): evolution of (a) mean, mode, (b) standard deviation, and (c) mean ± standard deviation of shear stress

of COV of yield strength on monotonic mean behavior can, similarly, be compared in Figs. 105.22(a) and 105.24(a).

105.3.2.5 Hardening Material

In this section, the influence of probabilistic yielding is evaluated on cyclic responses of isotropic and kinematic hardening materials. To this end, the same example, as discussed in the previous section (Section 105.3.2) is used but with appropriate hardening rule – isotropic or kinematic.

The main difference between the simulations shown in Section 105.3.2 for elastic–perfectly plastic material is that for a hardening material the internal variables (q_*, refer Eq. (105.45)) will evolve as the material plastifies. Such evolution (change) of internal variables is here assumed to be a function of plastic strain. The FPKE that govern the probabilistic evolution of internal variable (q) can be written, in most the general form, as:

$$\frac{\partial P(q(x_t,t),t)}{\partial t} = \frac{\partial}{\partial q}\left\{N^{eq}_{(1)_{IV}} P(q(x_t,t),t)\right\} + \frac{\partial^2}{\partial q^2}\left\{N^{eq}_{(2)_{IV}} P(q(x_t,t),t)\right\} \tag{105.53}$$

where, $N^{eq}_{(1)_{IV}}$ and $N^{eq}_{(2)_{IV}}$ are the equivalent advection and diffusion coefficients, respectively, for the internal variable. As explained for the case of probabilistic stress response for elastic–perfectly plastic material (refer Section 105.3.2), since point–location scale FPKE will be solved, the equivalent advection and diffusion coefficients for the internal variable, $N^{eq}_{(1)_{IV}}$ and $N^{eq}_{(2)_{IV}}$, can be written as:

$$\begin{aligned} N^{eq}_{(1)_{IV}}(q) &= P[\Sigma_y \leq \sigma(q)]\frac{d\epsilon_{xy}}{dt}\left\langle \frac{Gr}{G+\frac{1}{\sqrt{3}}r}\right\rangle \\ N^{eq}_{(2)_{IV}}(q) &= P[\Sigma_y \leq \sigma(q)]t\left(\frac{d\epsilon_{xy}}{dt}\right)^2 Var\left[\frac{Gr}{G+\frac{1}{\sqrt{3}}r}\right] \end{aligned} \tag{105.54}$$

where, r is the rate of evolution of internal variable (q) with plastic strain. One may note that in the above equivalent advection and diffusion coefficients (Eq. (105.54)), the contributions of probability weights that the material being elastic are absent. This is because the evolution rule of internal variable is governed by the plastic component of strain only. The equivalent advection and diffusion coefficients for shear stress ($N^{eq}_{(1)}$ and $N^{eq}_{(2)}$) for hardening–type materials, will have contributions from both elastic and plastic components, just like the elastic–perfectly plastic case. However, unlike the elastic–perfectly plastic case, those ($N^{eq}_{(1)}$ and $N^{eq}_{(2)}$) will contain the hardening terms:

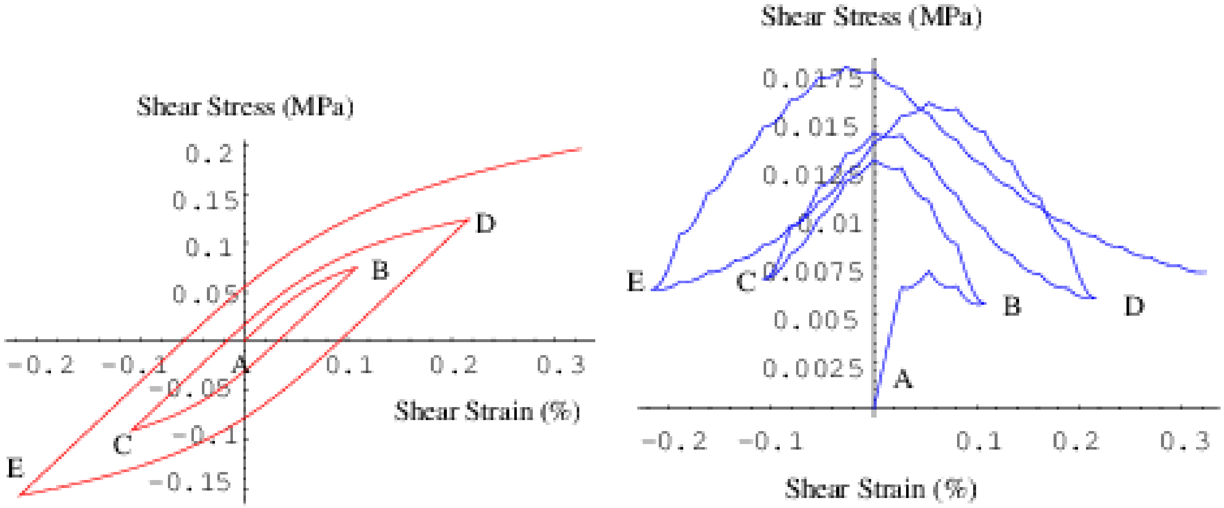
$$\begin{aligned} N^{eq}_{(1)}(\sigma) &= \frac{d\epsilon_{xy}}{dt}\left[(1-P[\Sigma_y \leq \sigma])\langle G\rangle + P[\Sigma_y \leq \sigma]\left\langle G - \frac{G^2}{G+\frac{1}{\sqrt{3}}r}\right\rangle\right] \\ N^{eq}_{(2)}(\sigma) &= t\left(\frac{d\epsilon_{xy}}{dt}\right)^2\left[(1-P[\Sigma_y \leq \sigma])Var[G] + P[\Sigma_y \leq \sigma]Var\left[G - \frac{G^2}{G+\frac{1}{\sqrt{3}}r}\right]\right] \end{aligned} \tag{105.55}$$

To obtain the probabilistic response of von Mises hardening material, the FPKE for probabilistic evolution of internal variable (Eq. (105.53), with advection and diffusion coefficients given by Eq. (105.54)) needs to be solved incrementally. This solution needs to be done simultaneously with the FPKE for probabilistic evolution of shear stress (Eq. (105.46), with advection and diffusion coefficients given by Eq. (105.55)). Those, in turn, need also to be solved incrementally, with the yield strength random variable (Σ_y) in Eqs. (105.54) and (105.55) being updated after each incremental step.

105.3.2.6 Isotropic Hardening

For von Mises isotropic hardening material, the yield strength (σ_y) is the internal variable. Yield strength will evolve probabilistically with plastic strain, following Eq. (105.53), with advection and diffusion coefficients given by Eq. (105.54). The shear stress, on the other hand, evolves in accordance with Eq. (105.46), with advection and diffusion coefficients given by Eq. (105.55).

Fig. 105.25 shows the evolutionary mean and standard deviation of shear stress during first couple of loading–unloading cycles for von Mises isotropic hardening material with a non–dimensional rate of evolution of internal variable (yield strength, in this case) of 10. All other material parameters are assumed to be the same as used for simulation of elastic–perfectly plastic material in the previous section (Section 105.3.2).



(a) (b)

Figure 105.25: Isotropic hardening probabilistic model under cyclic loading with increasing loops: evolution of (a) mean and (b) standard deviation of shear stress

The evolved PDFs of yield strength after each branch (loading, unloading, re–loading, and re–unloading) are shown in Fig. 105.26. The initial PDFs of yield strength (positive for loading branch and negative for unloading branch) are the same as assumed for elastic–perfectly plastic material in Section 105.3.2 (refer Fig. 105.18). As expected (and prescribed by the isotropic hardening model), the yield strength evolved (grew) isotropically. However, it is interesting to note the change in probability distributions of yield strength. The normally distributed initial PDFs of yield strength (Fig. 105.18) evolved into much dispersed non-Gaussian distributions having low kurtosis. In other words, when the material is cycled through loading–unloading cycles, the uncertainty in yield strength increases.

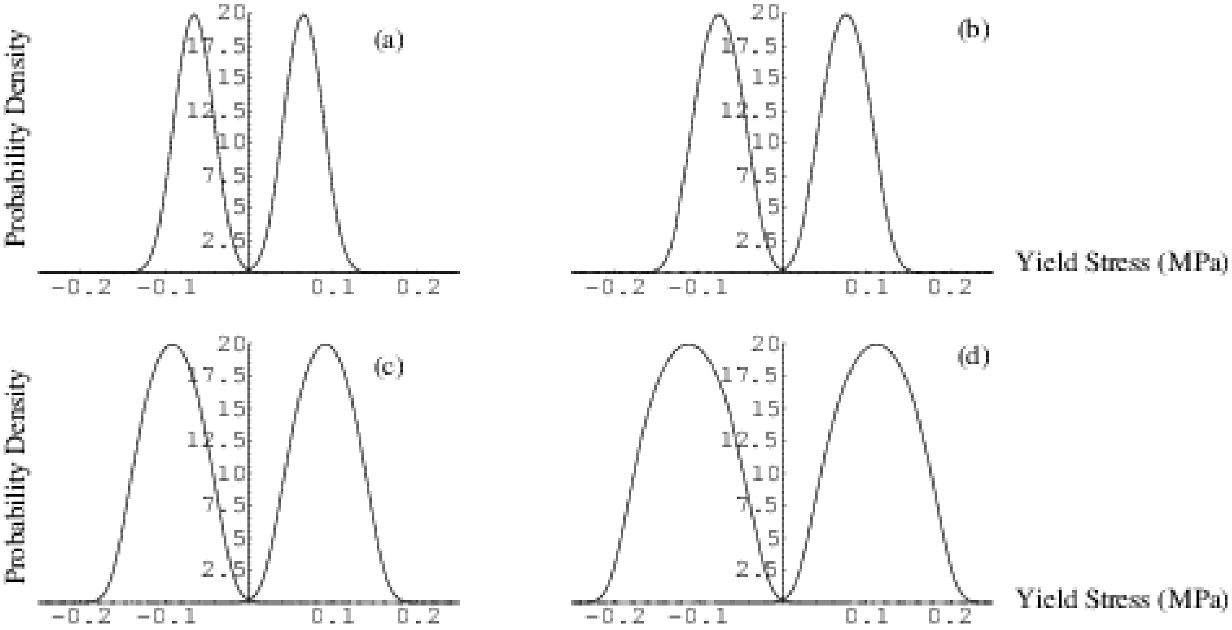
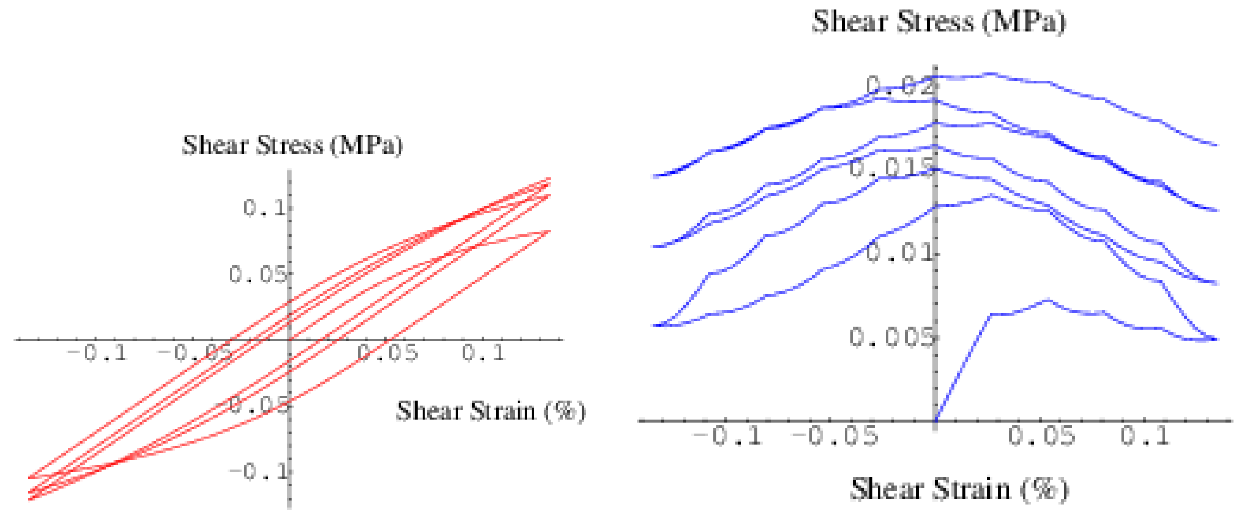


Figure 105.26: Isotropic hardening probabilistic model under cyclic loading with increasing loops: evolved PDF of yield stress after (a) loading branch, (b) unloading branch, (c) re–loading branch, and (d) re–unloading branch

Mathematically, increase in uncertainty of shear strength is due to the nonlinearity in formulation of probabilistic yielding , that is, the state variable q appears in both advection and diffusion equations (refer Eq. (105.54)), and in the evolution equation for internal variable (Eq. (105.53)).

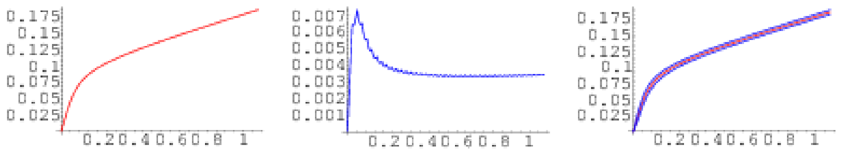
When comparison is made between Figs. 105.25 and 105.20, one can clearly see that, in simulating cyclic behaviors of geomaterials, isotropic hardening model (Fig. 105.25) performed, as expected, poorly. That is, the elastic–perfectly plastic probabilistic model (Fig. 105.20) captures (PDF of) stress–strain loops in a much more realistic way. However, for completeness of comparison, the behavior of isotropic hardening material, when it was cycled to same level (Fig. 105.27) and when loaded monotonically (Fig. 105.28) are also shown.

It is noted that monotonic loading curves for both perfectly plastic probabilistic model (Fig. 105.22) and linear isotropic hardening probabilistic model (Fig. 105.28) do look similar (with a noted difference of more pronounced hardening for a hardening model), but the real difference in stress–strain predictions with both probabilistic models becomes obvious in the case of cyclic loading.



(a) (b)

Figure 105.27: Isotropic hardening probabilistic model under cyclic loading with equal loops: evolution of (a) mean and (b) standard deviation of shear stress



(a) (b)

Figure 105.28: Isotropic hardening probabilistic model under monotonic loading: evolution of (a) mean, (b) standard deviation, and (c) mean ± standard deviation of shear stress

105.3.2.7 Kinematic Hardening

Expanding on elastic–plastic hardening probabilistic models, we now focus on a simple linear kinematic hardening rule based on evolution of back stress (α). By introducing back stress (α) to von Mises yield criteria, one can write:

$$\sqrt{J_\alpha} - k = 0 \tag{105.56}$$

where, k is again material parameter (yield strength like) and $J_\alpha = 3/2(s_{ij}-\alpha_{ij})(s_{ij}-\alpha_{ij})$ is the α–modified second invariant of deviatoric stress tensor (s_{ij}). For 1–D shear, Eq. (105.56) becomes:

$$|\sigma - \alpha| - \sigma_y = 0 \quad \text{or} \quad \sigma = \alpha \pm \sigma_y \tag{105.57}$$

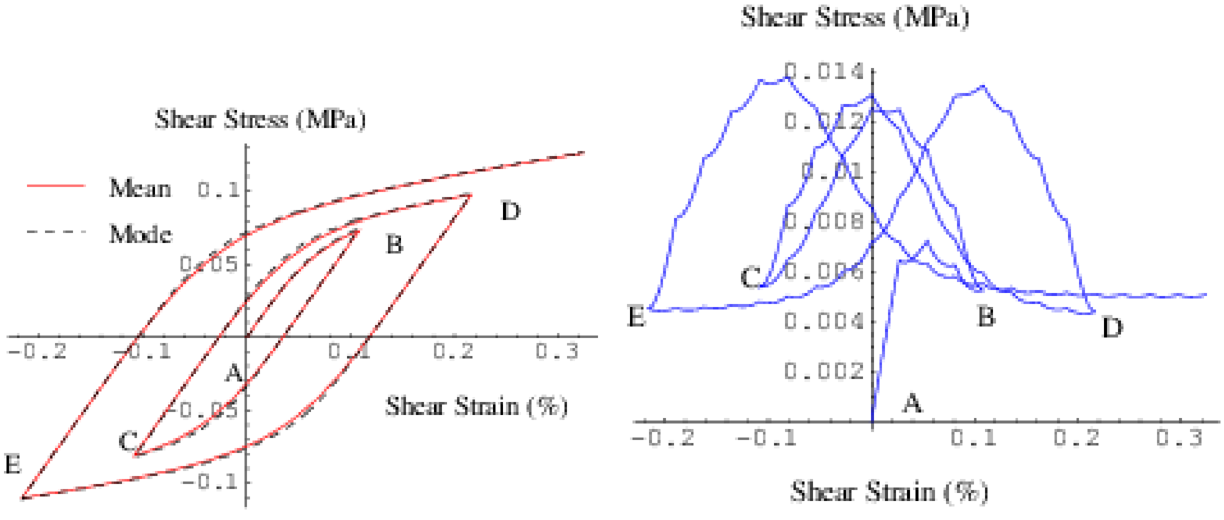
Hence, for kinematic hardening material, the yielding occurs at a stress of $\alpha \pm \sigma_y$, termed in the following as the equivalent yield stress. Initially, α is zero, and σ_y is assumed to have a mean value of 60 kPa with a standard deviation of 20 kPa, resulting in equivalent yield stress of 60 kPa with a COV of 30%, same as the assumed yield stress for the elastic–perfectly plastic material in Section 105.3.2 and isotropic hardening material in Section 105.3.2.6. However, the same distribution of equivalent yield stress will be obtained, if one transfers the initial uncertainty from σ_y to α. In other words, a deterministic σ_y of 60 kPa, and an uncertain α of zero mean and a standard deviation of 20 kPa will result in the same equivalent yield stress. The advantage of keeping σ_y deterministic is that it will simplify the probabilistic addition/subtraction in Eq. (105.57), while estimating the equivalent yield stress after each incremental step of the governing FPKEs, once the back stress (α), the internal variable for kinematic hardening material, starts evolving.

In this study, the back stress (α) is assumed to evolve with plastic strain and hence, it would evolve probabilistically similar to probabilistic evolution of the yield strength for isotropic hardening material. Probabilistic evolution of the back stress will occur according to Eq. (105.53), with advection and diffusion coefficients given by Eq. (105.54). Shear stress evolves according to Eq. (105.46), with advection and diffusion coefficients given by Eq. (105.55). One may note that the yield strength random variable (Σ_y), appearing in Eqs. (105.54) and (105.55), is the equivalent yield strength and is given by Eq. (105.57). Fig. 105.29 shows the probabilistic evolution of shear stress in terms of mean, mode, and standard deviation, when a kinematic hardening material[8], was cycled couple of times with increasing strain loops. All other material parameters are assumed to be the same as for the elastic–perfectly plastic material in Section 105.3.2. The evolved PDFs of the back stress (α) at the beginning and end of each branch (loading, unloading, re–loading, and re–unloading) are shown in Fig. 105.30. The evolved PDFs of equivalent yield stress (refer Eq. (105.57)) after each loading branch are shown in Fig. 105.31. Similar to the isotropic hardening case the uncertainty in (equivalent) yield strength increased as the material was cycled through, but unlike the isotropic hardening model, kinematic hardening model resulted in high kurtosis PDFs of (equivalent) yield strength. It is noted that the cyclic shear stress response of kinematic hardening material (Fig. 105.29), was more realistic than isotropic hardening material (Fig. 105.25), however, it didn't differ much from elastic–perfectly plastic material response (Fig. 105.20). Qualitatively, those, the elastic–perfectly plastic and the kinematic hardening responses, are similar. Like the elastic–perfectly plastic material, for kinematic hardening material, the mean and mode of the evolutionary shear stress (refer Fig. 105.29) are different, although not significantly.

Similarly, when one compares response (mean and standard deviation of shear stress) for loading cycles to the same strain level, for (i) elastic–perfectly plastic, (Fig. 105.21), (ii) isotropic linear hardening

---

[8] with non–dimensional rate of evolution of back stress with plastic strain of 10.



(a) (b)

Figure 105.29: Kinematic hardening probabilistic model under cyclic loading with increasing loops: evolution of (a) mean, mode and (b) standard deviation of shear stress

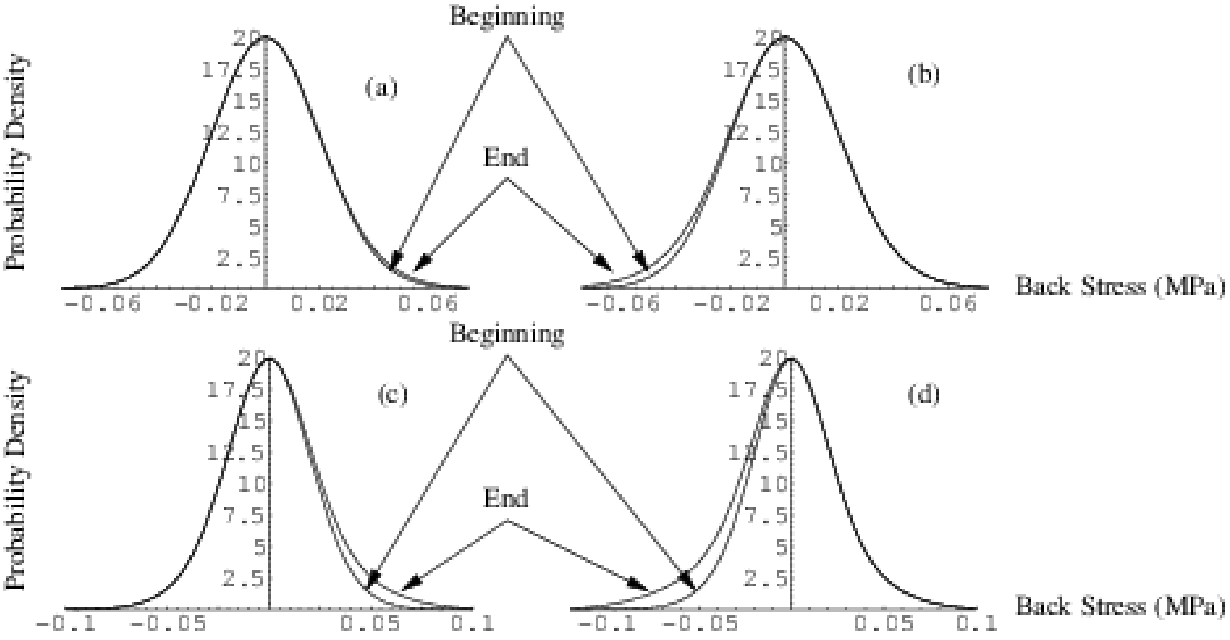


Figure 105.30: Kinematic hardening probabilistic model under cyclic loading with increasing loops: evolved PDF of back stress at the beginning and end of (a) loading branch, (b) unloading Branch, (c) re–loading branch, and (d) re–unloading branch

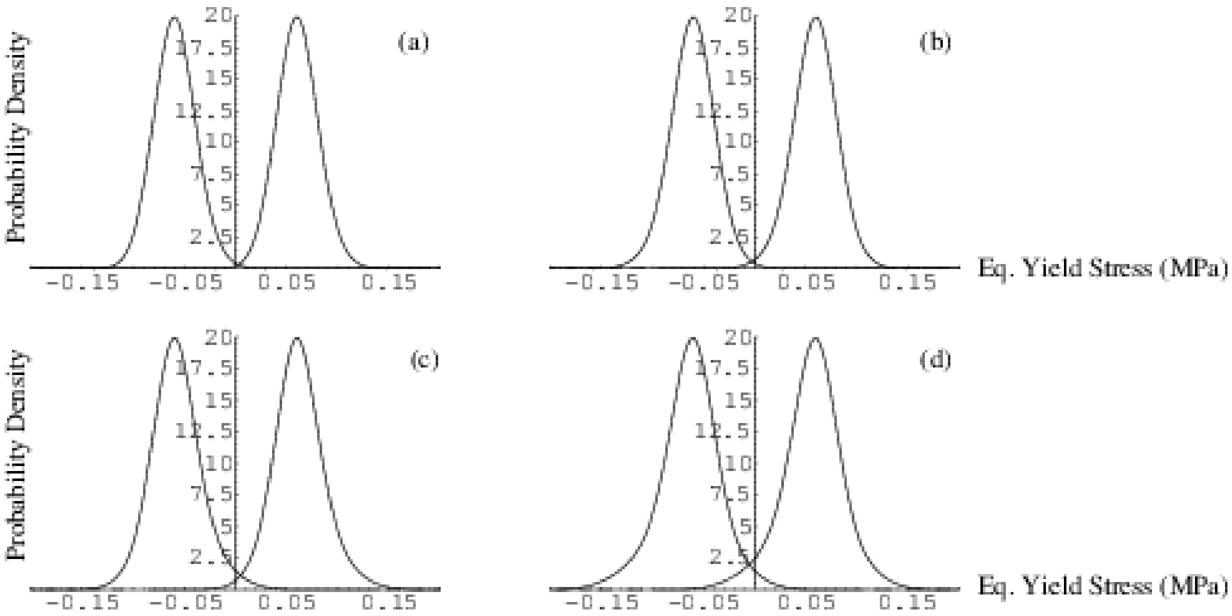


Figure 105.31: Kinematic hardening probabilistic model under cyclic loading with increasing loops: evolved PDF of equivalent yield stress after (a) loading branch, (b) unloading branch, (c) re–loading branch, and (d) re–unloading branch

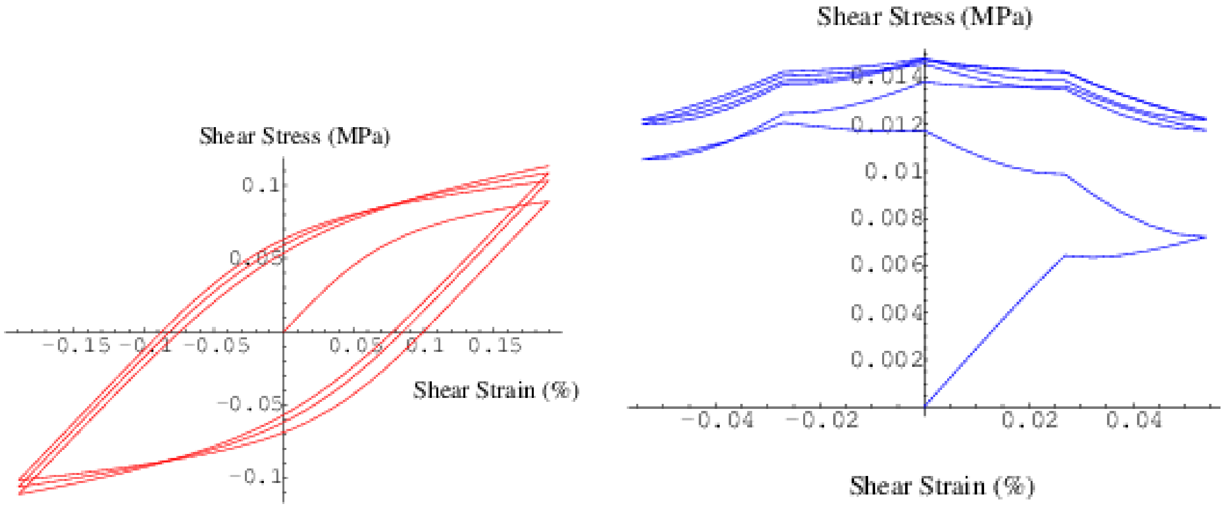
(Fig. 105.27), and (iii) linear kinematic hardening (Fig. 105.32), probabilistic material models, one can easily observe the qualitative similarity between elastic–perfectly plastic (i) and kinematic hardening responses (iii).

Monotonic loading cases, however, for all probabilistic material models ((i) elastic–perfectly plastic, (Fig. 105.22), (ii) isotropic linear hardening (Fig. 105.28), and (iii) linear kinematic hardening (Fig. 105.33)), are qualitatively similar, with expected differences in rate of hardening.

105.4 Hermite Polynomial Chaos Karhunen-Loève Expansion

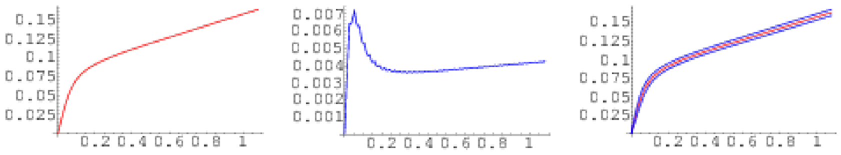
Hermite polynomial chaos Karhunen-Loève (PC-KL) expansion is formulated for the general heterogeneous random field $H(\boldsymbol{x}, \theta)$ of arbitrary marginal distributions. Here θ denotes the uncertainties and $\boldsymbol{x}$ is the general coordinate that can be either temporal as in the case of uncertain random process motions or spatial as in the case of random field material parameters.

Random field $H(\boldsymbol{x}, \theta)$ with any type of marginal distributions can be discretized with orthogonal Hermite polynomial chaos $\Omega_i(\gamma(\boldsymbol{x}, \theta))$ up to a certain order P Sakamoto and Ghanem (2002):



(a) (b)

Figure 105.32: Kinematic hardening probabilistic model under cyclic loading with equal loops: evolution of (a) mean and (b) standard deviation of shear stress



(a) (b)

Figure 105.33: Kinematic hardening probabilistic model under monotonic loading: evolution of (a) mean, (b) standard deviation, and (c) mean ± standard deviation of shear stress

$$H(\boldsymbol{x},\theta) = \sum_{i=0}^{P} H_i(\boldsymbol{x})\Omega_i(\gamma(\boldsymbol{x},\theta)) \tag{105.58}$$

where random functions $\{\Omega_i\}$ are Hermite polynomials constructed from a zero mean, unit variance kernel Gaussian random field $\gamma(\boldsymbol{x},\theta)$. Random functions $\{\Omega_i\}$ are determined by the initial condition $\Omega_0 = 1$ and the recursive relation:

$$\Omega_{i+1}(\gamma) = \gamma\Omega_i(\gamma) - \Omega_i^{'}(\gamma) \tag{105.59}$$

The deterministic PC coefficient field $H_i(\boldsymbol{x})$ can be calculated by projecting random field $H(\boldsymbol{x},\theta)$ onto PC basis Ω_i. Given the input covariance structure $Cov_H(x_1, x_2)$ of the original random field, the underlying

Gaussian covariance kernel $Cov_\gamma(x_1, x_2)$ can then be determined by solving:

$$Cov_H(x_1, x_2) = \sum_{i=1}^{P} H_i(x_1)\, H_i(x_2)\; i\,!\; Cov_\gamma(x_1, x_2)^i \tag{105.60}$$

Following Karhunen-Loève (KL) theorem Ghanem and Spanos (1991), the underlying kernel Gaussian random field $\gamma(\boldsymbol{x}, \theta)$ can be represented as:

$$\gamma(\boldsymbol{x}, \theta) = \sum_{i=1}^{M} \sqrt{\lambda_i} f_i(\boldsymbol{x}) \xi_i(\theta) \tag{105.61}$$

where eigenvalues λ_i and eigenvectors $f_i(\boldsymbol{x})$ of the covariance kernel $Cov_\gamma(x_1, x_2)$ have to satisfy Fredholm's integral equation of the second kind Sakamoto and Ghanem (2002). Zero mean and unit variance Gaussian random variables $\{\xi_i(\theta)\}$ are represented in M independent dimensions.

By combining Equations 105.58 to 105.61, the ultimate PC-KL expansion of the general random field $H(\boldsymbol{x}, \theta)$ into multi-dimensional orthogonal Hermite PC bases $\{\Psi_i\}$ of order P and dimension M can be obtained as:

$$H(\boldsymbol{x}, \theta) = \sum_{i=0}^{K} h_i(\boldsymbol{x}) \Psi_i(\{\xi_j(\theta)\}) \tag{105.62}$$

$$h_i(\boldsymbol{x}) = \frac{p!}{\langle \Psi_i^2 \rangle} H_p(\boldsymbol{x}) \prod_{j=1}^{p} \frac{\sqrt{\lambda_{k(j)}} f_{k(j)}(\boldsymbol{x})}{\sqrt{\sum_{m=1}^{M} (\sqrt{\lambda_m} f_m(\boldsymbol{x}))^2}} \tag{105.63}$$

where K is the total number of multidimensional Hermite PC bases $\{\Psi_i\}$ that depends on order P and dimension M as $K = 1 + \sum_{s=1}^{P} \frac{1}{s!} \prod_{j=0}^{s-1} (M + j)$. The upper product limit p, in equation 105.63 is the order of PC basis $\Psi_i(\{\xi_j(\theta)\})$. The marginal mean, marginal variance, correlation structure and any other statistics of the random field $H(\boldsymbol{x}, \theta)$ can be synthesized from the above Hermite PC-KL expansion. The goodness of the PC-KL expansion can be checked by comparing the PC-synthesized statistics with the input statistics of the random field $H(\boldsymbol{x}, \theta)$.

105.5 Galerkin Stochastic Elastoplastic Finite Element Formulations

105.5.1 Stochastic Elastoplastic Finite Element Method

Within the developed time domain stochastic Galerkin formulations, the uncertain material parameters and the uncertain forcing are modeled as heterogeneous random fields and non-stationary random process, respectively. As a unified probabilistic discretization scheme, Hermite PC-KL expansion is applied to both input uncertainties and output uncertainties, e.g., probabilistic displacement and acceleration responses.

Stochastic Galerkin projection is performed to minimize the error for estimating response PC coefficients. The resulting PC coefficients are used to develop statistics and distributions of the probabilistic response.

The weak form of deterministic, dynamic finite elements Jeremić et al. (1989-2025) can be written as:

$$\int_{D_e} N_m(\boldsymbol{x})\rho(\boldsymbol{x})N_n(\boldsymbol{x})dV\ \ddot{u}_n(t) + \int_{D_e} \nabla N_m(\boldsymbol{x})E(\boldsymbol{x})\nabla N_n(\boldsymbol{x})dV\ u_n(t) - f_m(t) = 0 \tag{105.64}$$

where N_m are the finite element shape functions, and $f_m(t)$ is the nodal force vector, while $\rho(\boldsymbol{x})$ and $E(\boldsymbol{x})$ are the deterministic density and deterministic stiffness fields in the deterministic FEM.

105.5.2 Stochastic Elastoplastic Finite Element Method, 1D Polynomial Chaos Formulation

Considering the tangential stiffness field, $E(\boldsymbol{x})$ to be a heterogeneous random field, and the forcing function, $f_m(t)$ to be a non-stationary random process, both are represented using multidimensional, Hermite PC expansions with known coefficients:

$$E(\boldsymbol{x},\theta) = \sum_{i=0}^{P_1} E_i(\boldsymbol{x})\Psi_i(\{\xi_r(\theta)\}) \tag{105.65}$$

$$f_m(t,\theta) = \sum_{j=0}^{P_2} f_{mj}(t)\psi_j(\{\xi_r(\theta)\}) \tag{105.66}$$

When the system with uncertain properties is excited by uncertain forces, the results, displacement and acceleration responses also become uncertain and can be represented using Hermite PCs with unknown coefficients expanded up to order P_3:

$$u_n(t,\theta) = \sum_{k=0}^{P_3} u_{nk}(t)\phi_k(\{\xi_l(\theta)\}) \tag{105.67}$$

$$\ddot{u}_n(t,\theta) = \sum_{k=0}^{P_3} \ddot{u}_{nk}(t)\phi_k(\{\xi_l(\theta)\}) \tag{105.68}$$

By substituting Eqs. 105.65, 105.66, 105.67, and 105.68 into Eq. 105.64, and denoting the shape function gradients $\nabla N_n(\boldsymbol{x})$ as $B(\boldsymbol{x})$, we obtain:

$$\begin{aligned} &\sum_{k=0}^{P_3} \int_{D_e} N_m(\boldsymbol{x})\rho(\boldsymbol{x})N_n(\boldsymbol{x})dV\ \phi_k\ddot{u}_{nk}(t)\ + \\ &\quad \sum_{k=0}^{P_3}\sum_{i=0}^{P_1} \int_{D_e} B_m(\boldsymbol{x})E_i(\boldsymbol{x})B_n(\boldsymbol{x})dV\ \Psi_i\phi_k u_{nk}(t) - \sum_{j=0}^{P_2} f_{mj}(t)\psi_j = 0 \end{aligned} \tag{105.69}$$

Performing stochastic Galerkin projection by multiplying both sides of Eq. 105.69 with PC basis ϕ_l and taking expectation Ghanem and Kruger (1996), we obtain the following system of ordinary differential equations (ODEs):

$$\sum_{k=0}^{P_3} \langle \phi_k \phi_l \rangle \int_{D_e} N_m(\boldsymbol{x}) \rho(\boldsymbol{x}) N_n(\boldsymbol{x}) dV \; \ddot{u}_{nk}(t) \; + \\ \sum_{k=0}^{P_3} \sum_{i=0}^{P_1} \langle \Psi_i \phi_k \phi_l \rangle \int_{D_e} B_m(\boldsymbol{x}) E_i(\boldsymbol{x}) B_n(\boldsymbol{x}) dV \; u_{nk}(t) = \sum_{j=0}^{P_2} \langle \psi_j \phi_l \rangle f_{mj}(t) \tag{105.70}$$

with $l = 0, 1, 2, ..., P_3$ where P_3 is the PC expansion order, while $m = 1, 2, ..., N$ where N is the number of finite element nodes, and the expectation operator is denoted as $\langle \cdot \rangle$. The expectations of double products $\langle \phi_k \phi_l \rangle$, $\langle \psi_j \phi_l \rangle$ and triple products $\langle \Psi_i \phi_k \phi_l \rangle$ of Hermite PC bases can be analytically computed beforehand and looked up during the stochastic FEM analysis.

Eq. 105.70 can be written into a matrix-vector form as:

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{F} \tag{105.71}$$

where $\boldsymbol{M}$, $\boldsymbol{K}$ and $\boldsymbol{F}$ are the stochastic expanded mass matrix, stiffness matrix and force vector, respectively. Vectors of unknown acceleration and displacement PC coefficients are denoted as $\ddot{\boldsymbol{u}}$ and $\boldsymbol{u}$, respectively.

Equation 105.71 is written in index notation, using Einstein summation convention, as:

$$M_{mlnk} \ddot{u}_{nk} + K_{mlnk} u_{nk} = F_{ml} \tag{105.72}$$

where

$$M_{mlnk} = \bigcup_e \langle \phi_k \phi_l \rangle \int_{D_e} N_m(\boldsymbol{x}) \rho(\boldsymbol{x}) N_n(\boldsymbol{x}) dV \tag{105.73}$$

$$K_{mlnk} = \bigcup_e \sum_{i=0}^{P_1} \langle \Psi_i \phi_k \phi_l \rangle \int_{D_e} B_m(\boldsymbol{x}) E_i(\boldsymbol{x}) B_n(\boldsymbol{x}) dV \tag{105.74}$$

$$F_{ml} = \bigcup_e \sum_{j=0}^{P_2} \langle \psi_j \phi_l \rangle f_{mj} \tag{105.75}$$

and $\bigcup_e$ is the assembly operator for elemental mass matrices, stiffness matrices and force vectors. Rayleigh damping can also be added into Eq. 105.71 as:

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{C}\dot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{F} \tag{105.76}$$

The stochastic expanded damping matrix $\boldsymbol{C}$ can be represented as:

$$\boldsymbol{C} = \alpha \boldsymbol{M} + \beta \boldsymbol{K} \tag{105.77}$$

where α and β are Rayleigh damping parameters. The Rayleigh damping would produce the damping ratio ξ dependent on the response frequency ω in a hyperbolic functional form as (Chopra Chopra (2000)):

$$\xi = \frac{1}{2}(\frac{\alpha}{\omega} + \beta\omega) \tag{105.78}$$

Due to the formulation of Rayleigh damping, its uncertainties are inherently considered and modeled through the uncertainties of the mass and stiffness matrix. In Eq. 105.77, both the matrix $\boldsymbol{M}$ and the matrix $\boldsymbol{K}$, are stochastic expanded matrices that reflect the uncertainties from the material density and stiffness. Additional uncertainties could be included through uncertain Rayleigh damping parameters α and β. However, these uncertainties are difficult to calibrate. Therefore, in this study, we consider both the damping parameters α and β to be deterministic. As such, the damping uncertainties come directly as the combination of the mass uncertainties and the stiffness uncertainties. The same Galerkin projection scheme is inherently applied to propagate the damping uncertainties to the probabilistic system response.

The ordinary differential system of equations 105.71 or 105.76 may be solved using any time integration method, for example Newmark method Newmark (1959). Note that the size of the stochastic finite element system of equations is much larger when compared to corresponding deterministic finite element system of equations, depending upon the number of PC terms used to represent the probabilistic displacement and acceleration response. After solving for the unknown PC coefficients for displacements u_{nk} and accelerations $\ddot{u}_{nk}$ from the stochastic FEM analysis, the complete probabilistic dynamic response of the system can be determined. Using the resulting PC coefficients for u_{nk} and $\ddot{u}_{nk}$, any probabilistic measure of the uncertain system response can be obtained. For example, the time-evolving mean and variance of the probabilistic displacement response at node n can be computed as:

$$\langle u_n(t,\theta) \rangle = u_{n0}(t) \tag{105.79}$$

$$\mathrm{Var}(u_n(t,\theta)) = \sum_{k=1}^{P_3} \langle \phi_k^2 \rangle u_{nk}^2(t) \tag{105.80}$$

105.5.3 Probabilistic Elastoplastic Constitutive Modeling, 1D Polynomial Chaos Formulation

In the above stochastic elastoplastic FEM formulation, the probabilistic tangential stiffness $E(\boldsymbol{x},\theta)$ needs to be updated at each incremental step. Probabilistic elastoplastic constitutive modeling is performed at each Gauss point to update the uncertain elastoplastic stiffness and stress. For the one dimensional (1D) stochastic site response analysis in this study, the constitutive behavior of the soil is represented by a 1D elastoplastic material model with a vanishing elastic region and Armstrong-Frederick nonlinear kinematic hardening Armstrong and Frederick (1966). Without considering uncertainties in material properties, the relationship between the stress increment $d\sigma$ and the strain increment $d\epsilon$, in 1D, can be written in the incremental form as:

$$d\sigma = H_a d\epsilon - C_r \sigma |d\epsilon| \tag{105.81}$$

where H_a and C_r are model parameters. The shear strength S_u given by the model becomes $S_u = H_a/C_r$. The elastoplastic tangential stiffness is a function of the stress σ as:

$$E(\sigma) = \frac{d\sigma}{d\epsilon} = H_a - C_r \sigma \; sgn(d\epsilon) \tag{105.82}$$

where $sgn(d\epsilon)$ is the sign function of the strain increment $d\epsilon$. This function returns $sgn(d\epsilon) = 1$ for the positive strain strain increment $d\epsilon > 0$ and $sgn(d\epsilon) = -1$ otherwise, for $d\epsilon < 0$.

Here model parameters H_a and C_r are considered to be uncertain and modeled as random fields $H_a(\boldsymbol{x}, \theta)$ and $C_r(\boldsymbol{x}, \theta)$. Representation of those two model parameters using Hermite PC bases $\varphi_i(\{\xi_r(\theta)\})$ can be written as:

$$H_a(\boldsymbol{x}, \theta) = \sum_{i=0}^{P} H_{ai}(\boldsymbol{x}) \varphi_i(\{\xi_r(\theta)\}) \tag{105.83}$$

$$C_r(\boldsymbol{x}, \theta) = \sum_{i=0}^{P} C_{ri}(\boldsymbol{x}) \varphi_i(\{\xi_r(\theta)\}) \tag{105.84}$$

The strain increments $d\epsilon(\boldsymbol{x}, \theta)$, that represent input to the constitutive driver (Equation 105.81), are also uncertain, since $\epsilon(\boldsymbol{x}, \theta) = B(\boldsymbol{x}) u_n(t, \theta)$:

$$d\epsilon(\boldsymbol{x}, \theta) = \sum_{i=0}^{P} d\epsilon_i(\boldsymbol{x}) \varphi_i(\{\xi_r(\theta)\}) \tag{105.85}$$

As a result, the probabilistic incremental stress $d\sigma(\boldsymbol{x}, \theta)$ and the tangential stiffness $E(\boldsymbol{x}, \theta)$ can be represented using unknown PC coefficients $\{d\sigma_i(\boldsymbol{x})\}$ and $\{E_i(\boldsymbol{x})\}$ as:

$$d\sigma(\boldsymbol{x}, \theta) = \sum_{i=0}^{P} d\sigma_i(\boldsymbol{x}) \varphi_i(\{\xi_r(\theta)\}) \tag{105.86}$$

$$E(\boldsymbol{x}, \theta) = \sum_{i=0}^{P} E_i(\boldsymbol{x}) \varphi_i(\{\xi_r(\theta)\}) \tag{105.87}$$

Substituting Equations 105.83 ∼ 105.87 into Equations 105.81 and 105.82, and applying Galerkin projection on PC basis $\varphi_i\{\xi_r(\theta)\}$ yields:

$$\sum_{m=0}^{P} d\sigma_m \langle \varphi_m \varphi_i \rangle = \sum_{j=0}^{P}\sum_{k=0}^{P} H_{aj} d\epsilon_k \langle \varphi_j \varphi_k \varphi_i \rangle \pm \sum_{l=0}^{P}\sum_{n=0}^{P}\sum_{s=0}^{P} C_{rl}\sigma_n d\epsilon_s \langle \varphi_l \varphi_n \varphi_s \varphi_i \rangle \tag{105.88}$$

$$\sum_{i=0}^{P} E_m \langle \varphi_m \varphi_i \rangle = \sum_{j=0}^{P} H_{aj} \langle \varphi_j \varphi_i \rangle \pm \sum_{l=0}^{P}\sum_{n=0}^{P} C_{rl}\sigma_n \langle \varphi_l \varphi_n \varphi_i \rangle \tag{105.89}$$

Solutions to the unknown PC coefficients of the incremental stress $d\sigma(\boldsymbol{x},\theta)$ and the elastoplastic stiffness $E(\boldsymbol{x},\theta)$ can be computed using the orthogonality of Hermite PC bases $\langle \varphi_i \varphi_j \rangle = \delta_{ij}$:

$$d\sigma_i = \frac{1}{\mathrm{Var}[\varphi_\mathrm{i}]}\left[H_{aj} d\epsilon_k \langle \varphi_j \varphi_k \varphi_i \rangle \pm C_{rl}\sigma_n d\epsilon_s \langle \varphi_l \varphi_n \varphi_s \varphi_i \rangle\right] \tag{105.90}$$

$$E_i = H_{ai} \pm \frac{1}{\mathrm{Var}[\varphi_\mathrm{i}]} C_{rl}\sigma_n \langle \varphi_l \varphi_n \varphi_i \rangle \tag{105.91}$$

where $\mathrm{Var}[\varphi_\mathrm{i}]$ is the scalar variance of PC basis $\varphi_i\{\xi_r(\theta)\}$, that equals to $\langle \varphi_i^2 \rangle$. The Einstein's summation convention is followed in equations 105.90 and 105.91 with index i as a free index. The above formulation for the probabilistic constitutive modeling is implemented in the context of the explicit, forward Euler algorithm Jeremić et al. (1989-2025).

To illustrate the above probabilistic constitutive model, Figure 105.34 shows the stress-strain behavior using uncertain material parameters Ha with mean 10MPa and coefficient of variation (CV) 25%, uncertain shear strength $S_u = H_a/C_r$ with mean 150kPa and coefficient of variation 25%. The material is driven by an uncertain cyclic strain with a mean strain increment 10^{-4} and coefficient of variation of 20%.

It is observed that the probabilistic material response obtained through the intrusive polynomial chaos modeling matches well with Monte Carlo analysis using $10,000$ samples. It is noted that the intrusive probabilistic modeling is around $2,000$ times more computationally efficient, faster than Monte Carlo simulations.

105.5.4 Stochastic Elastoplastic Finite Element Method, 3D Polynomial Chaos Formulation

105.5.5 Probabilistic Elastoplastic Constitutive Modeling, 3D Polynomial Chaos Formulation

105.6 Sobol' Indices Computation Using Polynomial Chaos Expansion

In global sensitivity analysis, the variance of model output is decomposed into a sum of contributions from individual random variable, or groups of random variables. Consider a general mathematical model

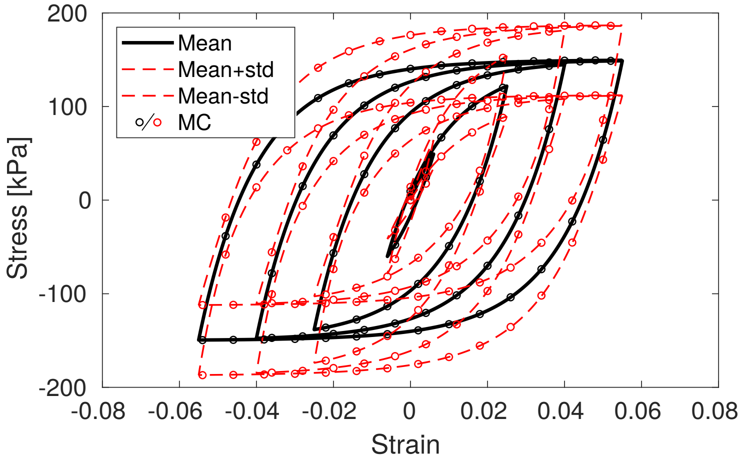


Figure 105.34: The hysteretic behavior of 1-D elastoplastic material with uncertain parameters H_a and H_a/C_r.

with n uncertain inputs represented by vector $\boldsymbol{x}$ and scalar output y as:

$$y = f(\boldsymbol{x}) \quad \boldsymbol{x} \in I^n \tag{105.92}$$

where the input parameters $\boldsymbol{x}$ are defined in n dimensional unit cube I^n without the loss of generality. The ANOVA (ANalysis Of VAriance) representation of $f(x)$ can be written as Sobol (2001):

$$f(x_1, x_2, ..., x_n) = f_0 + \sum_{i=1}^{n} f_i(x_i) + \sum_{1 \le i < j \le n} f_{ij}(x_i, x_j) + ... + f_{1,2,...,n}(x_1, ..., x_n) \tag{105.93}$$

There are 2^n summands in total. Constant f_0 is the mean value of the function:

$$f_0 = \int_{I^n} f(\boldsymbol{x}) d\boldsymbol{x} \tag{105.94}$$

The integral of each summand in Equation 105.93 over any of its independent variable is zero:

$$\int_0^1 f_{i_1,...,i_s}(x_{i_1}, x_{i_2}, ..., x_{i_s}) \; dx_{i_k} = 0 \quad \text{for} \quad 1 \le k \le s \tag{105.95}$$

From Equation 105.95, it can be seen that the summands are orthogonal to each other in the following sense:

$$\int_{I^n} f_{i_1,...,i_s}(x_{i_1}, x_{i_2}, ..., x_{i_s}) f_{j_1,...,j_t}(x_{j_1}, x_{j_2}, ..., x_{j_t}) \; d\boldsymbol{x} = 0 \quad \text{for} \quad \{i_1, ..., i_s\} \neq \{j_1, ..., j_t\} \tag{105.96}$$

For given mathematical model $f(x)$, the above ANOVA representation is unique and can be derived analytically. For example, the univariate terms can be solved as:

$$f_i(x_i) = \int_{I^{n-1}} f(\boldsymbol{x})\ d\boldsymbol{x}_{\sim i} - f_0 \tag{105.97}$$

where $\int_{I^{n-1}}(\cdot)\ d\boldsymbol{x}_{\sim i}$ denotes the integration over all dimensions except x_i. Similarly, the bivariate terms can be derived as follows:

$$f_{ij}(x_i, x_j) = \int_{I^{n-2}} f(\boldsymbol{x})\ d\boldsymbol{x}_{\sim[ij]} - f_i(x_i) - f_j(x_j) - f_0 \tag{105.98}$$

Following this procedure, any summand $f_{i_1,...,i_s}(x_{i_1}, x_{i_2}, ..., x_{i_s})$ can be constructed from some multidimensional integral of $f(\boldsymbol{x})$.

Considering uncertain input parameters $\boldsymbol{X}$ to be independent random variables following uniform distribution in $[0,1]$:

$$\boldsymbol{X} = [X_1, ..., X_n], \quad X_i \sim U(0,1), \quad \text{for} \quad i = 1, ..., n \tag{105.99}$$

Then the total variance of the probabilistic model response $y = f(\boldsymbol{X})$ can be computed as:

$$D = \mathrm{Var}[f(\boldsymbol{X})] = \int_{I^n} f^2(\boldsymbol{x})\ d\boldsymbol{x} - f_0^2 \tag{105.100}$$

Using Equations 105.93 and 105.96, the total variance D can be decomposed as follows:

$$D = \sum_{i=1}^{n} D_i + \sum_{1\le i<j\le n} D_{ij} + ... + D_{1,2,...,n} = \sum_{s=1}^{n} \sum_{i_1<...<i_s} D_{i_1...i_s} \tag{105.101}$$

where the variance contribution from individual summand is given as follows:

$$D_{i_1...i_s} = \int_{I^s} f^2_{i_1...i_s}(x_{i_1}, ..., x_{i_s})\ dx_{i_1}, ...dx_{i_s}, \text{with } 1 \le i_1 < ... < i_s \le n, s = 1, ..., n \tag{105.102}$$

The Sobol' indices $S_{i_1...i_s}$ are defined as:

$$S_{i_1...i_s} = D_{i_1...i_s}/D \tag{105.103}$$

The Sobol' indices $S_{i_1...i_s}$ quantifies the fractional contributions from random inputs $\{X_{i_1}, ..., X_{i_s}\}$ to the total variance D. The first order indices S_i gives the influence of each uncertain input parameter X_i when considered individually. The high order terms describe the mixed influence when a group of uncertain input parameters are considered collectively. Clearly, from Equation 105.101 we have:

$$\sum_{i=1}^{n} S_i + \sum_{1\le i<j\le n} S_{ij} + ... + S_{1,2,...,n} = 1 \tag{105.104}$$

The Sobol' indices for global sensitivity analysis can be used for the following purposes: (1) Ranking of input uncertain parameters $\boldsymbol{X} = [X_1, ..., X_n]$. (2) Neglect of inessential uncertain parameters and high order terms in Equation 105.93. The total sensitivity indices S_i^{total} is defined to evaluate the total influence of a certain input parameter X_i as:

$$S_i^{\text{total}} = \sum_{\mathscr{S}_i} D_{i_1 \ldots i_s} \tag{105.105}$$

where set $\mathscr{S}_i$ contains all the indexes involving index i defined as Equation 105.106 and Sobol' indices set $\{D_{i_1 \ldots i_s}\}$ is the collection of all partial sensitivity indices that are related to parameter X_i.

$$\mathscr{S}_i = \{(i_1, ..., i_s) : \exists k, 1 \leq k \leq s, i_k = i\} \tag{105.106}$$

Using Hermite PC expansion Ghanem and Spanos (1991), the probabilistic model response $y = f(\boldsymbol{X})$ can be represented as:

$$y = \sum_{j=0}^{P-1} y_j \Psi_j(\boldsymbol{\xi}), \quad \boldsymbol{\xi} = \{\xi_1, ..., \xi_M\} \tag{105.107}$$

where $\{\Psi_j\}$ are multi-dimensional, mutually orthogonal Hermite PC bases of order p constructed from M dimension, independent, standard Gaussian random vector $\boldsymbol{\xi}$. The total number of PC bases P is related to the dimension M and order p as:

$$P = \begin{pmatrix} M + p \\ p \end{pmatrix} = \frac{(M+p)!}{M!p!} \tag{105.108}$$

It is noted that the input random vector $\boldsymbol{X}$ of any prescribed joint PDF or any given marginal PDF and correlations can be approximately transformed to standard Gaussian random vector $\boldsymbol{\xi}$ using transformation techniques such as iso-probabilistic transform and Nataf transform Lebrun and Dutfoy (2009):

$$\boldsymbol{X} = T(\boldsymbol{\xi}) \tag{105.109}$$

Therefore, the probabilistic model response can be evaluated and represented with Hermite PC expansion as:

$$y = f(\boldsymbol{X}) = f \circ T(\boldsymbol{\xi}) = \sum_{j=0}^{P-1} y_j \Psi_j(\boldsymbol{\xi}), \quad \boldsymbol{\xi} = \{\xi_1, ..., \xi_M\} \tag{105.110}$$

Here we transform the input random vector $\boldsymbol{X}$ into standard Gaussian random vector $\boldsymbol{\xi}$ and perform Hermite polynomial chaos expansion. We can also transform the input random vector $\boldsymbol{X}$ into other types of basic random variables, e.g., uniform distribution, and represent model response y with associated generalized PC expansion Xiu and Karniadakis (2002).

Since the PC bases $\{\Psi_j(\boldsymbol{\xi})\}$ are zero-mean $(j \geq 1)$ and mutually orthogonal, the mean $\bar{y}$ and total variance of model response D_{PC} can be calculated from its PC representation:

$$\begin{aligned} \bar{y} &= \boldsymbol{E}[f(\boldsymbol{X})] = y_0 \\ D^{PC} &= \text{Var}\left[\sum_{j=0}^{P-1} y_j \Psi_j\right] = \sum_{j=1}^{P-1} y_j^2 \boldsymbol{E}\left[\Psi_j^2\right] \end{aligned} \tag{105.111}$$

To compute the Sobol' indices, the above PC expansion of y should be organized into the ANOVA form Sudret (2008). The multi-dimensional polynomial chaos bases $\{\Psi_j(\boldsymbol{\xi})\}$ can be decomposed into the multiplication of single dimension polynomial chaos bases of different orders:

$$\Psi_j(\boldsymbol{\xi}) = \prod_{i=1}^{n} \phi_{\alpha_i}(\xi_i) \tag{105.112}$$

where $\phi_{\alpha_i}(\xi_i)$ is the single dimensional, order α_i, polynomial function of underlying basic random variable ξ_i. The functional form of the generalized polynomial chaos function $\phi_{\alpha_i}(\xi_i)$ depends on the selected underlying basic random variable ξ_i Xiu and Karniadakis (2002). For example, when the basic random variable ξ_i follows standard Gaussian distribution, Hermite polynomial functions $\phi_{\alpha_i}(\xi_i)$ should be constructed as follows:

$$\phi_0 = 1; \qquad \phi_{k+1}(\xi_i) = \xi_i \phi_k(\xi_i) - \phi_k^{'}(\xi_i) \;\; \text{for all } k \geq 1 \tag{105.113}$$

Each multidimensional polynomial chaos basis can be uniquely characterized by vector $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_n)$. The connection between the PC expansion and the ANOVA representation of model response y can be established by defining set $\mathscr{S}$ from $\boldsymbol{\alpha}$ as Sudret (2008):

$$\mathscr{S}_{i_1,...,i_s} = \{\boldsymbol{\alpha} : \forall k = 1, ..., n, \text{when } k \in (i_1, ..., i_s),\; \alpha_k > 0,\; \text{otherwise, } \alpha_k = 0\} \tag{105.114}$$

For example, $\mathscr{S}_i$ would correspond to the PC bases depending only on dimension ξ_i. Following Equation 105.114, the PC expansion shown in Equation 105.110 could be written into ANOVA representation:

$$\begin{aligned} y = y_0 &+ \sum_{i=1}^{n} \sum_{\boldsymbol{\alpha} \in \mathscr{S}_i} y_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\xi_i) + \sum_{1 \leq i_1 < i_2 \leq n} \sum_{\boldsymbol{\alpha} \in \mathscr{S}_{i_1,i_2}} y_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\xi_{i_1}, \xi_{i_2}) + ... \\ &+ \sum_{1 \leq i_1 < ... < i_s \leq n} \sum_{\boldsymbol{\alpha} \in \mathscr{S}_{i_1,...,i_s}} y_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\xi_{i_1}, ..., \xi_{i_s}) + ... \\ &+ \sum_{\boldsymbol{\alpha} \in \mathscr{S}_{1,2,...,n}} y_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\xi_1, ..., \xi_n) \end{aligned} \tag{105.115}$$

where the term $\sum_{\boldsymbol{\alpha}\in\mathscr{S}_{i_1,...,i_s}} y_{\boldsymbol{\alpha}}\Psi_{\boldsymbol{\alpha}}(\xi_{i_1},...,\xi_{i_s})$ denotes the summation of polynomial chaos expansions depending on all the basic random variables $\{\xi_{i_1},...,\xi_{i_s}\}$ and only on them. From the above ANOVA representation of probabilistic model response, the PC-based Sobol' indices $S^{PC}_{i_1...i_s}$ can be derived as:

$$S^{PC}_{i_1...i_s} = \sum_{\boldsymbol{\alpha}\in\mathscr{S}_{i_1,...,i_s}} y^2_{\boldsymbol{\alpha}}\boldsymbol{E}\left[\Psi^2_{\boldsymbol{\alpha}}\right]/D^{PC} \tag{105.116}$$

The total Sobol' indices $S^{PC,\text{total}}_{j_1..j_t}$ for any group of parameters $\{\xi_{j_1},...,\xi_{j_t}\}$ can then be computed as:

$$S^{PC,\text{total}}_{j_1..j_t} = \sum_{(i_1,...,i_s)\in\mathscr{S}_{j_1,...,j_t}} S^{PC}_{i_1...i_s} \tag{105.117}$$

where set $\mathscr{S}_{j_1,...,j_t}$ is defined as:

$$\mathscr{S}_{j_1,...,j_t} = \{(i_1,...,i_s) : (j_1,...,j_t) \subset (i_1,...,i_s)\} \tag{105.118}$$

From the above formulations, it can be observed that once the PC representation of probabilistic model response is established, Sobol' sensitivity indices can be analytically evaluated with very small computational expense.

Chapter 106

Large Deformation Elasto-Plasticity

(1996-2004-)

(In collaboration with Dr. Zhao Cheng)

106.1 Chapter Summary and Highlights

106.2 Continuum Mechanics Preliminaries: Kinematics

106.2.1 Deformation

In modeling the material nonlinear behavior of solids, plasticity theory is applicable primarily to those bodies that can experience inelastic deformations considerably greater than the elastic deformation. If the resulting total deformation, including both translations and rotations, are small enough, we can apply small deformation theory in solving these problems. If, however strains and rotations are finite, one must resort to the theory of large deformations. In doing so, we will be using two sets of representations[1], namely:

- *Material coordinates* in the undeformed configuration, also called *Lagrangian coordinates*,
- *Spatial coordinates* in the deformed configuration, also called *Eulerian coordinates*.

Figure 106.1 shows the displacement of a particle from its initial position X_I to the current position x_i, defined by the deformation equation:

$$x_i = x_i\,(X_1, X_2, X_3, t) \tag{106.1}$$

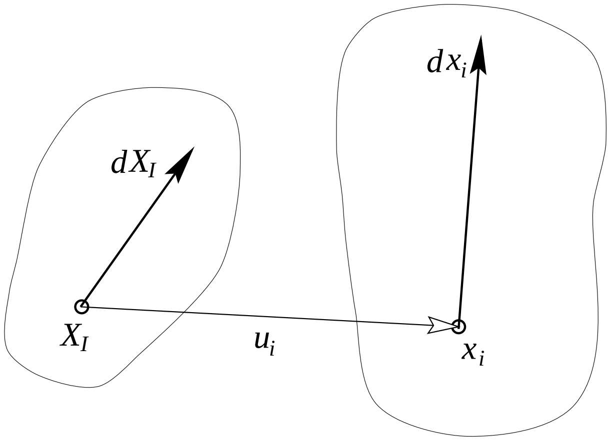


Figure 106.1: Displacement, stretch and rotation of material vector dX_I to new position dx_i.

The initial position X_I of the particle now occupying the position x_i is given by the *Eulerian equation*:

$$X_I = X_I\,(x_1, x_2, x_3, t) \tag{106.2}$$

---

[1] See Malvern (1969).

The two positions are connected by the displacement u_I:

$$x_i = X_I + u_i \qquad ; \qquad X_I = x_i - u_i \tag{106.3}$$

106.2.2 Deformation Gradient

The *deformation gradients* are the gradients of the functions on the right–hand side of equations (106.1) and (106.2). To emphasize the difference between the material, Lagrangian setting and the spatial, Eulerian setting, we will use capital letters for the material coordinate indices and lower case letters for the spatial coordinate indices. We limit our work to the rectangular Cartesian coordinates, thus simplifying the tensor notation to the covariant indices only.

The deformation gradient is defined as the *two–point tensor* whose rectangular Cartesian components are the partial derivatives:

$$F_{kK} = \frac{\partial x_k}{\partial X_K} = x_{k,K} \tag{106.4}$$

The deformation gradient F_{kK} transforms (convects) on an arbitrary infinitesimal material vector dX_I at X_I to associate it with a vector dx_i at x_i:

$$dx_k = F_{kK} dX_K = \frac{\partial x_k}{\partial X_K} dX_K = x_{k,K} dX_K \tag{106.5}$$

The *the spatial deformation gradients* are tensors referred to the deformed, Eulerian configuration:

$$(F_{Kk})^{-1} = \frac{\partial X_K}{\partial x_k} = X_{K,k} \tag{106.6}$$

Similarly to the deformation gradient F_{kK}, *spatial deformation gradient* $(F_{Kk})^{-1}$ operates on an arbitrary infinitesimal material vector dx_i at x_i to associate it with a vector dX_I at X_I:

$$dX_K = (F_{Kk})^{-1} dx_k = \frac{\partial X_K}{\partial x_k} dx_k = X_{K,k} dx_k \tag{106.7}$$

The *spatial deformation gradient* $(F_{Kk})^{-1}$ at x_i is the inverse to the two–point tensor F_{kK} at X_I:

$$F_{iJ} \left(F_{Jk}\right)^{-1} = \delta_{ik} \quad \text{and} \quad \left(F_{Ij}\right)^{-1} F_{jK} = \delta_{IK} \tag{106.8}$$

The Jacobian of the mapping (106.4) can be represented as:

$$J = det\left(F_{kK}\right) = \frac{1}{6} e_{ijk} e_{PQR} F_{iP} F_{jQ} F_{kR} \tag{106.9}$$

The relative deformation gradient f_{km} is the gradient for the relative motion function:

$$\xi = \chi_t \left(x_i , \tau \right) \tag{106.10}$$

and is defined as:

$$f_{km} = \xi_{k,m} \equiv \frac{\partial \xi_k}{\partial x_m} \tag{106.11}$$

If the fixed reference position X_I, the current position x_i and the variable position ξ_i are all referred to the rectangular Cartesian coordinate system, the chain rule of differentiation yields:

$$\frac{\partial \xi_k}{\partial X_I} = \frac{\partial \xi_k}{\partial x_m} \frac{\partial x_m}{\partial X_I} \quad \text{or} \quad F_{kI} = f_{km} \, F_{mI} \tag{106.12}$$

The polar decomposition theorem permits the unique representation[2]:

$$F_{ij} = R_{ik} U_{kj} = v_{ik} R_{kj} \tag{106.13}$$

where U_{kj}, v_{ik} are positive definite symmetric tensors, called *right stretch tensors* and *left stretch tensors*, respectively, and R_{kj} is an *orthogonal tensor* such that:

$$R_{ik} R_{jk} = \delta_{ij} \qquad \text{and also} \qquad R_{ki} R_{kj} = \delta_{ij} \tag{106.14}$$

Equation (106.13), as well as Figure 106.2.2 demonstrate that the motion and deformation of an infinitesimal volume element at X_i consist of consecutive applications of:

- a stretch by U_{kj},
- a rigid body rotation by R_{ik},
- a rigid body translation to x_i

or alternatively:

- a rigid body translation to x_i
- a rigid body rotation by R_{kj},
- a stretch by v_{ik},

---

[2] referring x_i and X_i to the same reference axes and using lower case indices for both. This reference to the same coordinate system will be applied only for the polar decomposition example presented here.

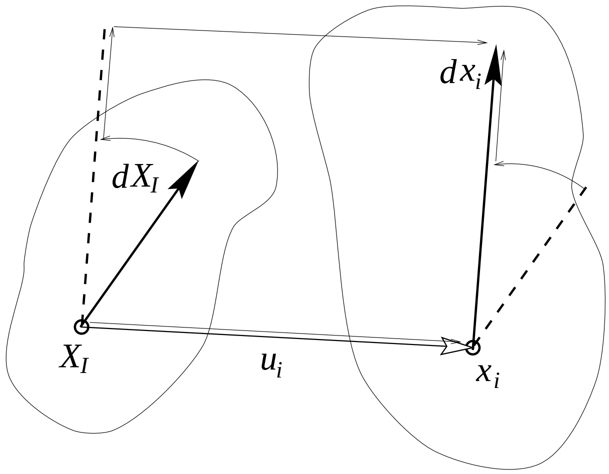


Figure 106.2: Illustration of the equation $F_{ij} = R_{ik}U_{kj} = v_{ik}R_{kj}$.

106.2.3 Strain Tensors, Deformation Tensors and Stretch

The strain tensors E_{IJ} and e_{ij} are defined so that they give the change in the square length of the material vector dX_I. For the Lagrangian formulation we write:

$$(ds)^2 - (dS)^2 = 2dX_I E_{IJ} dX_J \tag{106.15}$$

and for the Eulerian formulation:

$$(ds)^2 - (dS)^2 = 2dx_i e_{ij} dx_j \tag{106.16}$$

The deformation tensors C_{IJ} and c_{ij} are connecting the squared lengths in Lagrangian and Eulerian configurations. The Green deformation tensor[3] C_{IJ}, referred to the undeformed configuration, gives the new squared length $(ds)^2$ of the element into which the given element dX_I is deformed:

$$(ds)^2 = dX_I C_{IJ} dX_J \tag{106.17}$$

The Cauchy deformation tensor c_{ij}, sometimes also denoted as[4] $\left(b_{ij}\right)^{-1}$, gives the initial squared length $(dS)^2$ of an element dx_i identified in the deformed configuration:

$$(dS)^2 = dx_i c_{ij} dx_j \tag{106.18}$$

---

[3] Also called *right Cauchy–Green tensor*.

[4] Another name for b_{ij} is *Finger deformation tensor* or *left Cauchy–Green tensor*.

Substituting equation (106.17) into (106.15) yield:

$$2E_{IJ} = C_{IJ} - \delta_{IJ} \tag{106.19}$$

and similarly, substituting equation (106.18) into (106.16) we obtain:

$$2e_{ij} = \delta_{ij} - c_{ij} \tag{106.20}$$

By using equation (106.5) we can express $(ds)^2$ as:

$$\begin{gathered}(ds)^2 = dx_k dx_k = (F_{kI} dX_I)(F_{kJ} dX_J) = \\ (x_{k,I} dX_I)(x_{k,J} dX_J) = dX_I (F_{kI} F_{kJ}) dX_K = dX_I C_{IJ} dX_K\end{gathered} \tag{106.21}$$

so we have obtained the connection between the deformation tensor C_{IJ} and the deformation gradient F_{kI} in the form:

$$C_{IJ} = (F_{kI} F_{kJ}) = x_{k,I} dX_I x_{k,J} dX_J \tag{106.22}$$

Similarly, by using equation (106.7) and the expression for $(dS)^2$ we can establish the connection between the deformation tensor c_{ij} and the deformation gradient F_{Ki} as:

$$\begin{gathered}(dS)^2 = dS_K dX_K = (F_{Ki} dx_i)(F_{Kj} dx_j) = \\ (X_{K,i} dx_i)(X_{K,j} dx_J) = dx_i (F_{Ki} F_{Kj}) dx_k = dx_i c_{ij} dx_k \Rightarrow \\ c_{ij} = (F_{Ki})^{-1} F^{-1}_{Kj}\end{gathered} \tag{106.23}$$

The expressions for the strain tensors in Lagrangian and Eulerian description[5] is obtained from equations (106.19) and (106.20):

$$\text{L:}\ \ E_{IJ} = \frac{1}{2}\left((F_{kI} F_{kJ}) - \delta_{IJ}\right) \qquad ; \qquad \text{E:}\ \ e_{ij} = \frac{1}{2}\left(\delta_{ij} - (F_{Ki})^{-1} \left(F_{Kj}\right)^{-1}\right) \tag{106.24}$$

If one starts from the displacement equation (106.3), referenced to the same axes for both X_I and x_i

$$x_I = X_I + u_I \quad ; \quad X_I = x_I - u_I$$

the general expression for the Lagrangian strain tensor E_{IJ} in terms of displacements is:

$$\begin{gathered}E_{IJ} = \frac{1}{2}\left((F_{KI} F_{KJ}) - \delta_{IJ}\right) = \\ \frac{1}{2}\left(\left(\delta_{KI} + u_{K,I}\right)\left(\delta_{KJ} + u_{K,J}\right) - \delta_{IJ}\right) = \\ \frac{1}{2}\left(\delta_{KI}\delta_{KJ} + \delta_{KI} u_{K,J} + u_{K,I}\delta_{KJ} + u_{K,I} u_{K,J} - \delta_{IJ}\right) = \\ \frac{1}{2}\left(\delta_{IJ} + u_{I,J} + u_{J,I} + u_{K,I} u_{K,J} - \delta_{IJ}\right) = \\ \frac{1}{2}\left(u_{I,J} + u_{J,I} + u_{K,I} u_{K,J}\right)\end{gathered} \tag{106.25}$$

---

[5] Lagrangian format will be denoted by L: while Eulerian format by E:.

Similarly, the general expression for the Eulerian strain tensor e_{ij} in terms of displacements is:

$$\begin{aligned} e_{ij} = \frac{1}{2}\left(\delta_{ij} - (F_{ki})^{-1}\left(F_{kj}\right)^{-1}\right) = \\ \frac{1}{2}\left(\delta_{ij} - \left(\delta_{ki} - u_{k,i}\right)\left(\delta_{kj} - u_{k,j}\right)\right) = \\ \frac{1}{2}\left(\delta_{ij} - \delta_{ki}\delta_{kj} + \delta_{ki}u_{k,j} + u_{k,i}\delta_{kj} - u_{k,i}u_{k,j}\right) = \\ \frac{1}{2}\left(\delta_{ij} - \delta_{ij} + u_{i,j} + u_{j,i} - u_{k,i}u_{k,j}\right) = \\ \frac{1}{2}\left(u_{i,j} + u_{j,i} - u_{k,i}u_{k,j}\right) \end{aligned} \tag{106.26}$$

It is worthwhile noting that equations (106.25) and (106.26) represent the complete finite strain tensor. They involve only linear and quadratic terms in the components of displacement gradients.

The stretch is a measure of extension of an infinitesimal element and is a function of direction of an element, in either deformed or undeformed configuration. By denoting N_I a unit vector in the undeformed configuration and n_i a unit vector in the deformed configuration, we denote material stretch as $\Lambda_{(N)}$ of those elements whit initial direction N_I and spatial stretch $\lambda_{(n)}$ of those elements with initial direction n_i. By dividing equations (106.15) and (106.16) by $(ds)^2$ and $(dS)^2$ respectively and by using:

$$N_I = \frac{dX_I}{dS} \qquad \text{and} \qquad n_i = \frac{dx_i}{ds} \tag{106.27}$$

we obtain the Cartesian form of stretch in the Lagrangian and Eulerian descriptions:

$$\text{L:}\ \ \Lambda^2_{(N)} = \frac{dX_I}{dS} C_{IJ} \frac{dX_J}{dS} \qquad \text{and} \qquad \text{E:}\ \ \lambda^2_{(n)} = \frac{dx_i}{ds} c_{ij} \frac{dx_j}{ds} \tag{106.28}$$

General strain tensors can be defined by considering a scale function (Hill, 1978) for the stretch. Scale function is any smooth, monotonic function of stretch $f(\lambda)$ such that:

$$f(\lambda)\ \ ;\ \ \lambda \in [0, \infty)\ \ \text{subject to}\ \ f(1) = 0, f'(1) = 1 \tag{106.29}$$

Scale function is often taken in the form $(\lambda^{2m} - 1)/2m$, where m may have any value. If we choose m to be an integer, the corresponding strain tensor is:

$$E_{IJ} = \frac{\left(U^{2m}_{IJ} - \delta_{IJ}\right)}{2m} \quad \text{where} \quad F_{IJ} = R_{IK}U_{KJ} = v_{IK}R_{KJ} \tag{106.30}$$

Table 106.1 shows different Lagrangian strain measures obtained for a particular choice of parameter m.

In the Eulerian setting, generalized strain tensor is defined as

$$e_{ij} = \frac{\left(\delta_{ij} - v^{2m}_{ij}\right)}{2m} \quad ; \quad F_{IJ} = R_{IK}U_{KJ} = v_{IK}R_{KJ} \tag{106.31}$$

Table 106.1: Different Lagrangian strain measures.

Strain measure name	parameter m	expression for E^{m}_{IJ}
Green–Lagrange	1	$E^{GL}_{IJ} = \left(U^{2}_{IJ} - \delta_{IJ}\right)/2$
Almansi	-1	$E^{A}_{IJ} = \left(\delta_{IJ} - U^{-2}_{IJ}\right)/2$
Biot	1/2	$E^{B}_{IJ} = \left(U_{IJ} - \delta_{IJ}\right)$
Hencky	0	$E^{H}_{IJ} = \ln\left(U_{IJ}\right)$

106.2.4 Rate of Deformation Tensor

The *rate of deformation tensor*[6] describes the *tangent motion* in terms of velocity components $v_i = dx_i/dt$. The spatial coordinates are:

$$v_i = v_i\left(x_1, x_2, x_3, t\right) \tag{106.32}$$

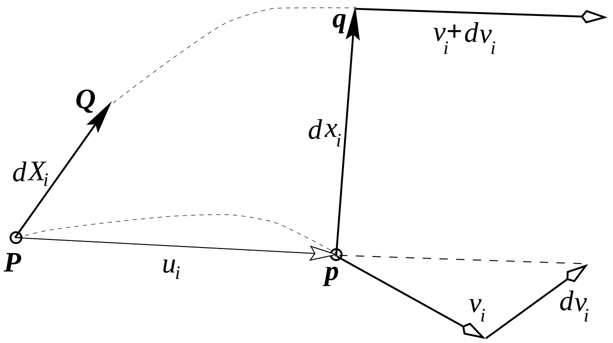


Figure 106.3: Relative velocity dv_i of particle *Q* at point *q* relative to particle *P* at point *p*.

In Figure 106.3 the dashed lines represents the trajectories of particles *P* and *Q*. The velocity vectors v_i at *p* and $v_i + dv_i$ at *q* are tangent to the two trajectories. The relative velocity components dv_i of particle at *q* relative to the particle at *p* are given by:

$$dv_k = \frac{\partial v_k}{\partial x_m} dx_m = v_{k,m} dx_m = L_{km} dx_m \tag{106.33}$$

---

[6] Also called *stretch tensor* or *velocity strain*.

The *spatial gradient of the velocity* L_{km} can be decomposed as the sum of the symmetric, *rate of deformation tensor* D_{km}, and a skew symmetric *spin tensor* W_{km} as follows:

$$L_{km} = \frac{1}{2}\left(L_{km} + L_{mk}\right) + \frac{1}{2}\left(L_{km} - L_{mk}\right) = D_{km} + W_{km} \tag{106.34}$$

where:

$$D_{km} = \frac{1}{2}\left(L_{km} + L_{mk}\right) = D_{mk} \qquad \text{and} \qquad W_{km} = \frac{1}{2}\left(L_{km} - L_{mk}\right) = -W_{mk} \tag{106.35}$$

An alternate way of deriving the rate of deformation tensor goes as follows. The rate of change of squared length $(ds)^2$ is given as:

$$\frac{d\,(ds)^2}{dt} = 2\frac{d\,(ds)}{dt}ds \tag{106.36}$$

since $(ds)^2 = dx_k dx_k$ it follows:

$$\frac{d\,(ds)^2}{dt} = 2\frac{d\,(dx_k)}{dt}dx_k \tag{106.37}$$

and with $dx_k = (\partial x_k / \partial X_m) dX_m$ it follows:

$$\frac{d\,(dx_k)}{dt} = \frac{d\left(\frac{\partial x_k}{\partial X_m}dX_m\right)}{dt} = \frac{d\left(\frac{\partial x_k}{\partial X_m}\right)}{dt}dX_m + \frac{d\,(dX_m)}{dt}\frac{\partial x_k}{\partial X_m} = \frac{d\left(\frac{\partial x_k}{\partial X_m}\right)}{dt}dX_m \tag{106.38}$$

since $d\,(dX_m)/dt \equiv 0$, because the initial relative position vector dX_m does not change with time. By interchanging the order of differentiation we get:

$$\frac{d\,(dx_k)}{dt} = dv_k = \frac{d\left(\frac{\partial x_k}{\partial X_M}\right)}{dt}dX_m = \frac{\partial v_k}{\partial X_m}dX_m \qquad \text{where} \qquad v_k = \frac{\partial x_k}{dt} \tag{106.39}$$

From equation (106.33) $dv_k = L_{km} dx_m$ and equation (106.39) it follows that:

$$\frac{\partial v_k}{\partial X_m}dX_m = L_{km}dx_m \qquad \Rightarrow \qquad \frac{d\,(dx_k)}{dt} = dv_k = L_{km}dx_m = v_{k,m}dx_m \tag{106.40}$$

and then the equation (106.37) becomes:

$$\begin{aligned}\frac{d\,(ds)^2}{dt} &= 2\frac{d\,(dx_k)}{dt}dx_k = 2dx_k v_{k,m} dx_m dx_k = 2dx_k L_{km} dx_m dx_k = \\ &= 2dx_k D_{km} dx_m dx_k + 2dx_k W_{km} dx_m dx_k = 2dx_k D_{km} dx_m dx_k\end{aligned} \tag{106.41}$$

since $dx_k dx_m \equiv dx_m dx_k$ and W_{km} is skew symmetric such that $W_{km} = -W_{mk}$. Finally we obtain:

$$\frac{d\,(ds)^2}{dt} = 2dx_k D_{km} dx_m \tag{106.42}$$

and thus it follows that the rate of change of the squared length $(ds)^2$ of the material instantaneously occupying any infinitesimal relative position dx_k at point *p* is determined by the tensor D_{km} at point *p*.

In order to compare the strain rate to the rate of deformation, we differentiate equation (106.15) with respect to time:

$$\frac{d\left((ds)^2-(dS)^2\right)}{dt} = 2\frac{d\left(dX_I E_{IJ} dX_J\right)}{dt} =$$

$$= \frac{d\left((ds)^2\right)}{dt} = 2dX_I \frac{d\left(E_{IJ}\right)}{dt} dX_J \qquad (106.43)$$

since $(dS)^2$ and dX_I are constant through time. From the equations (106.42) and (106.43) it follows that:

$$\frac{d\,(ds)^2}{dt} = 2dx_k D_{km} dx_m d = 2\left(dX_I F_{Ik}\right) D_{km}\left(F_{mJ} dX_J\right) = 2dX_I\left(F_{Ik} D_{km} F_{mJ}\right) dX_J \qquad (106.44)$$

and from equations (106.43) and (106.44) it follows that:

$$\frac{dE_{IJ}}{dt} = F_{Ik} D_{km} F_{mJ} \qquad (106.45)$$

or inversely:

$$D_{km} = \left(F_{Ik}\right)^{-1} \frac{dE_{IJ}}{dt}\left(F_{mJ}\right)^{-1} \qquad (106.46)$$

To obtain the rate of change of the deformation gradient we start from equations (106.4) and differentiate it with respect to time:

$$\frac{dF_{kK}}{dt} = \frac{d\left(\frac{\partial x_k}{\partial X_K}\right)}{dt} = \frac{\partial\left(\frac{dx_k}{dt}\right)}{\partial X_K} = \frac{\partial v_k}{\partial X_K} = \frac{\partial v_k}{\partial x_m}\frac{\partial x_m}{\partial X_K} = v_{k,m} x_{m,K} = \frac{dx_{k,K}}{dt} =$$

$$= L_{km} F_{mK} = \dot{F}_{kK} \qquad (106.47)$$

or inversely:

$$v_{k,m} = \frac{dx_{k,K}}{dt} X_{K,m} = \frac{dF_{kK}}{dt}\left(F_{Km}\right)^{-1} =$$

$$= \dot{F}_{kK}\left(F_{Km}\right)^{-1} = L_{km} \qquad (106.48)$$

106.3 Constitutive Relations: Hyperelasticity

106.3.1 Introduction

A material is called *hyperelastic* or *Green elastic*, if there exists an *elastic potential function* W, also called the *strain energy function per unit volume of the undeformed configuration*, which represents a scalar function of strain of deformation tensors, whose derivatives with respect to a strain component determines the corresponding stress component. The most general form of the elastic potential function, is described in equation 106.49, with restriction to pure mechanical theory, by using the *axiom of locality* and the *axiom of entropy production*[7]:

$$W = W\left(X_K, F_{kK}\right) \tag{106.49}$$

By using the *axiom of material frame indifference*[8], we conclude that W depends only on X_K and C_{IJ}, that is:

$$W = W\left(X_K, C_{IJ}\right) \quad \text{or:} \quad W = W\left(X_K, c_{ij}\right) \tag{106.50}$$

By assuming hyperelastic response, the following are the constitutive equations for the material stress tensors:

- 2. Piola–Kirchhoff stress tensor:

$$S_{IJ} = 2\frac{\partial W}{\partial C_{IJ}} \tag{106.51}$$

- Mandel stress tensor:

$$T_{IJ} = C_{IK} S_{KJ} = 2 C_{IK} \frac{\partial W}{\partial C_{KJ}} \tag{106.52}$$

- 1. Piola–Kirchhoff stress tensor

$$P_{iJ} = S_{IJ}(F_{iI})^t = 2\frac{\partial W}{\partial C_{IJ}}(F_{iI})^t \tag{106.53}$$

and the spatial, Kirchhoff stress tensor is defined as:

- Kirchhoff stress tensor

$$\tau_{ij} = 2\,\frac{\partial W}{\partial b_{ij}} = 2\,F_{iA}(F_{jB})^t \frac{\partial W}{\partial C_{AB}} = F_{iA}(F_{jB})^t S_{AB} \tag{106.54}$$

---

[7] See Marsden and Hughes (1983) pp. 190.

[8] See Marsden and Hughes (1983) pp. 194.

Material tangent stiffness relation is defined from:

$$dS_{IJ} = 2\,\frac{\partial^2 W}{\partial C_{IJ}\,\partial C_{KL}}\,dC_{KL} = \frac{1}{2}\,\mathcal{L}_{IJKL}\,dC_{KL} \tag{106.55}$$

where

$$\mathcal{L}_{IJKL} = 4\,\frac{\partial^2 W}{\partial C_{IJ}\,\partial C_{KL}} \tag{106.56}$$

The spatial tangent stiffness tensor $\mathcal{E}_{ijkl}$ is obtained by the following *push–forward* operation with the deformation gradient:

$$\mathcal{E}_{ijkl} = F_{iI}F_{jJ}(F_{kK})^t(F_{lL})^t\mathcal{L}_{IJKL} \tag{106.57}$$

106.3.2 Isotropic Hyperelasticity

In the case of material isotropy, the strain energy function $W\left(X_K, C_{IJ}\right)$ belongs to the class of isotropic, invariant scalar functions. It satisfies the relation:

$$W\left(X_K, C_{KL}\right) = W\left(X_K, Q_{KI}C_{IJ}\left(Q_{JL}\right)^t\right) \tag{106.58}$$

where Q_{KI} is the proper orthogonal transformation. If we choose $Q_{KI} = R_{KI}$, where R_{KI} is the orthogonal rotation transformation, defined by the polar decomposition theorem in equation (106.13), then:

$$W\left(X_K, C_{KL}\right) = W\left(X_K, U_{KL}\right) = W\left(X_K, v_{kl}\right) \tag{106.59}$$

Right and left stretch tensors, U_{KL}, v_{kl} have the same principal values[9] $\lambda_i \;\; ; \;\; i = \overline{1,3}$ so the strain energy function W can be represented in terms of principal stretches, or similarly in terms of principal invariants of deformation tensor:

$$W = W\left(X_K, \lambda_1, \lambda_2, \lambda_3, \right) = W\left(X_K, I_1, I_2, I_3\right) \tag{106.60}$$

where:

$$\begin{aligned} I_1 &\overset{\text{def}}{=} C_{II} \\ I_2 &\overset{\text{def}}{=} \frac{1}{2}\left(I_1^2 - C_{IJ}C_{JI}\right) \\ I_3 &\overset{\text{def}}{=} \det\left(C_{IJ}\right) = \frac{1}{6}e_{IJK}e_{PQR}C_{IP}C_{JQ}C_{KR} = J^2 \end{aligned} \tag{106.61}$$

Left and right Cauchy–Green tensors were defined by equations (106.22) and (106.23), respectively as:

$$C_{IJ} = (F_{kI})^t F_{kJ} \quad ; \quad (c^{-1})_{ij} = b_{ij} = F_{iK}(F_{jK})^t \tag{106.62}$$

---

[9] Principal stretches.

The spectral[10] decomposition theorem for symmetric positive definite tensors[11] states that:

$$C_{IJ} = \lambda_A^2 \left(N_I^{(A)} N_J^{(A)} \right)_A \qquad \text{where} \qquad A = \overline{1,3} \tag{106.63}$$

and N_I are the eigenvectors[12] of C_{IJ}. Values λ_A^2 are the roots of the characteristic polynomial

$$P(\lambda_A^2) \stackrel{\text{def}}{=} -\lambda_A^6 + I_1\ \lambda_A^6 - I_2\ \lambda_A^4 + I_3 = 0 \tag{106.64}$$

It should be noted that no summation is implied over indices in parenthesis[13].

The mapping of the eigenvectors can be deduced from equation (106.5) and is given by

$$\lambda_{(A)}\ n_i^{(A)} = F_{iJ}\ N_J^{(A)} \tag{106.65}$$

where $\|n_i^{(A)}\| \equiv 1$. The spectral decomposition of F_{iJ}, R_{iJ} and b_{ij} is then given by

$$F_{iJ} = \lambda_A \left(n_i^{(A)} N_J^{(A)} \right)_A \tag{106.66}$$

$$R_{iJ} = \sum_{A=1}^{3} n_i^{(A)} N_J^{(A)} \tag{106.67}$$

$$b_{ij} = \lambda_A^2 \left(n_i^{(A)} n_j^{(A)} \right)_A \tag{106.68}$$

Spectral decomposition from equation (106.63) is valid for the case of non–equal principal stretches, i.e. $\lambda_1 \neq \lambda_2 \neq \lambda_3$. If two or all three principal stretches are equal, we shall introduce a small perturbation to the numerical values for principal stretches in order to make them distinct. The case of two or all three values of principal stretches being equal is theoretically possible and results for example from standard triaxial tests or isotropic compression tests. However, we are never certain about equivalence of two numerical numbers, because of the finite precision arithmetics involved in calculation of these numbers. From the numerical point of view, two number are equal if the difference between them is smaller than the machine precision ($macheps$) specific to the computer platform on which computations are performed. Our perturbation will be a function of the $macheps$.

The characteristic polynomial $P(\lambda_A^2)$ from equation (106.64) can be solved[14] for λ_A:

$$\lambda_A = \frac{1}{\sqrt{3}} \sqrt{I_1 + 2\sqrt{I_1^2 - 3I_2} \cos\left(\frac{\Theta + 2\pi A}{3} \right)} \tag{106.69}$$

---

[10] See Simo and fTaylor (1991).

[11] Cauchy–Green tensor C_{IJ} for example.

[12] So that $\|N_I\| = 1$.

[13] For example, in the present case $N_I^{(A)}$ is the Ath eigenvector with members $N_1^{(A)}$, $N_2^{(A)}$ and $N_3^{(A)}$, so that the actual equation $C_{IJ} = \lambda_A^2 \left(N_I^{(A)} N_J^{(A)} \right)_A$ can also be written as $C_{IJ} = \sum_{A=1}^{A=3} \lambda_{(A)}^2 N_I^{(A)} N_J^{(A)}$. In order to follow the consistency of indicial notation in this work, we shall make an effort to represent all the tensorial equations in indicial form.

[14] See also Morman (1986) and Schellekens and Schellekens and Parisch (1994).

where

$$\Theta = \arccos \frac{2I_1^3 - 9I_1I_2 + 27I_3}{2\sqrt{\left(I_1^2 - 3I_2\right)^3}} \tag{106.70}$$

Recently, Ting (1985) and Morman (1986) have used Serrin's representation theorem in order to devise a useful representation for generalized strain tensors[15] E_{IJ} and e_{ij} through C^m_{IJ} and b^m_{ij}. Morman (1986) has shown that b^m_{ij} can be stated as

$$b^m_{ij} = \lambda_A^{2m} \left(\frac{(b^2)_{ij} - \Big((I_1 - \lambda^2_{(A)} \Big) b_{ij} + I_3 \lambda^{-2}_{(A)} \delta_{ij}}{2\lambda^4_{(A)} - I_1 \lambda^2_{(A)} + I_3 \lambda^{-2}_{(A)}} \right)_A \tag{106.71}$$

By comparing equations (106.71) and (106.68) it follows that the Eulerian eigendiad $n_i^{(A)} n_j^{(A)}$ can be written as

$$n_i^{(A)} n_j^{(A)} = \frac{(b^2)_{ij} - \left(I_1 - \lambda^2_{(A)} \right) b_{ij} + I_3 \lambda^{-2}_{(A)} \delta_{ij}}{2\lambda^4_{(A)} - I_1 \lambda^2_{(A)} + I_3 \lambda^{-2}_{(A)}} \tag{106.72}$$

The Lagrangian eigendiad $N_I^{(A)} N_J^{(A)}$, from equation (106.63), can be derived, if one substitutes mapping of the eigenvectors, (106.65), into equation (106.72) to get:

$$N_I^{(A)} N_J^{(A)} = \lambda^2_{(A)} \frac{C_{IJ} - \left(I_1 - \lambda^2_{(A)} \right) \delta_{IJ} + I_3 \lambda^{-2}_{(A)} (C^{-1})_{IJ}}{2\lambda^4_{(A)} - I_1 \lambda^2_{(A)} + I_3 \lambda^{-2}_{(A)}} \tag{106.73}$$

where it was used that:

$$C_{IJ} = (F_{iI})^{-1} \; (b^2)_{ij} \; (F_{jJ})^{-t} \tag{106.74}$$

$$\delta_{IJ} = (F_{iI})^{-1} \; b_{ij} \; (F_{jJ})^{-t} \tag{106.75}$$

$$(C^{-1})_{IJ} = (F_{iI})^{-1} \; \delta_{ij} \; (F_{jJ})^{-t} \tag{106.76}$$

It should be noted that the denominator in equations (106.72) and (106.73) can be written as:

$$2\lambda^4_{(A)} - I_1 \lambda^2_{(A)} + I_3 \lambda^{-2}_{(A)} = \left(\lambda^2_{(A)} - \lambda^2_{(B)} \right) \left(\lambda^2_{(A)} - \lambda^2_{(C)} \right) \overset{\text{def}}{=} D_{(A)} \tag{106.77}$$

where indices A, B, C are cyclic permutations of $1, 2, 3$. It follows directly from the definition of $D_{(A)}$ in equation (106.77) that $\lambda_1 \neq \lambda_2 \neq \lambda_3 \Rightarrow D_{(A)} \neq 0$ for equations (106.72) and (106.73) to be valid. Similarly to equations (106.63) and (106.68) we can obtain:

$$(C^{-1})_{IJ} = \lambda_A^{-2} \left(N_I^{(A)} N_J^{(A)} \right)_A \tag{106.78}$$

$$(b^{-1})_{ij} = \lambda_A^{-2} \left(n_i^{(A)} n_j^{(A)} \right)_A \tag{106.79}$$

---

[15] Defined by equations (106.30) and (106.31).

106.3.3 Volumetric–Isochoric Decomposition of Deformation

It proves useful to separate deformation in volumetric and isochoric parts by a multiplicative split of a deformation gradient as

$$F_{iI} = \tilde{F}_{i\beta} \, {}^{vol}F_{\beta I} \qquad \text{where} \qquad \tilde{F}_{i\beta} = F_{iI} J^{-\frac{1}{3}} \quad ; \quad {}^{vol}F_{\beta I} = J^{\frac{1}{3}} \delta_{\beta I} \tag{106.80}$$

where x_β represents an intermediate configuration such that deformation $X_I \rightarrow x_\beta$ is purely volumetric and $x_\beta \rightarrow x_i$ is purely isochoric. It also follows from equation (106.80) that $\tilde{F}_{\beta I}$ and F_{iI} have the same eigenvectors.

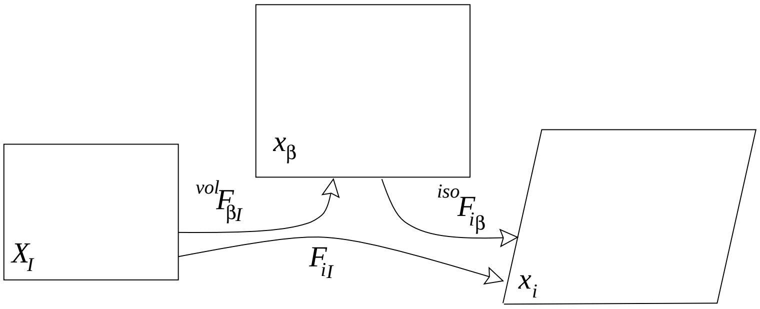


Figure 106.4: Volumetric isochoric decomposition of deformation.

The isochoric part of the Green deformation tensor C_{IJ}, defined in equation (106.63) can be defined as

$$\tilde{C}_{IJ} = J^{-\frac{2}{3}} C_{IJ} = \tilde{\lambda}_A^2 \left(N_I^{(A)} N_J^{(A)} \right)_A \tag{106.81}$$

while the isochoric part of the Finger deformation tensor b_{ij} can be defined similarly as

$$\tilde{b}_{ij} = J^{-\frac{2}{3}} b_{ij} = \tilde{\lambda}_A^2 \left(n_i^{(A)} n_j^{(A)} \right)_A \tag{106.82}$$

where the isochoric principal stretches are defined as

$$\tilde{\lambda}_A = J^{-\frac{1}{3}} \lambda_A = (\lambda_1 \lambda_2 \lambda_3)^{-\frac{1}{3}} \lambda_A \tag{106.83}$$

The free energy W is then decomposed additively as:

$$W \left(X_K, \lambda_{(A)} \right) = {}^{iso}W \left(X_K, \tilde{\lambda}_{(A)} \right) + {}^{vol}W \left(X_K, J \right) \tag{106.84}$$

106.3.4 Simo–Serrin's Formulation

In Section (106.3.2) we have presented the most general form of the isotropic strain energy function W in terms of of principal stretches:

$$W = W \left(X_K, \lambda_1, \lambda_2, \lambda_3, \right) \tag{106.85}$$

It was also shown in Section (106.3.1) that it is necessary to calculate the gradient $\partial W / \partial C_{IJ}$ in order to obtain 2. Piola–Kirchhoff stress tensor S_{IJ} and accordingly other stress measures. Likewise, it was shown that the material tangent stiffness tensor $\mathcal{L}_{IJKL}$ (as well as the spatial tangent stiffness tensor $\mathcal{E}_{ijkl}$) requires second order derivatives of strain energy function $\partial^2 W/(\partial C_{IJ}\,\partial C_{KL})$. In order to obtain these quantities we introduce[16] a second order tensor $M^{(A)}_{IJ}$

$$\begin{aligned} M^{(A)}_{IJ} &\overset{\text{def}}{=} \lambda^{-2}_{(A)}\, N^{(A)}_I\, N^{(A)}_J \\ &= (F_{iI})^{-1} \left(n^{(A)}_i n^{(A)}_j\right) (F_{jJ})^{-t} \\ &= \frac{1}{D_{(A)}} \left(C_{IJ} - \left(I_1 - \lambda^2_{(A)}\right)\delta_{IJ} + I_3\lambda^{-2}_{(A)}(C^{-1})_{IJ}\right) \quad \text{from (106.73)} \end{aligned} \tag{106.86}$$

where $D_{(A)}$ was defined by equation (106.77). With $M^{(A)}_{IJ}$ defined by equation (106.86), we get from equation (106.63) that:

$$C_{IJ} = \lambda^4_A \left(M^{(A)}_{IJ}\right)_A \tag{106.87}$$

and also from equation (106.78) it follows that:

$$(C^{-1})_{IJ} = M^{(1)}_{IJ} + M^{(2)}_{IJ} + M^{(3)}_{IJ} \tag{106.88}$$

It can also be concluded that:

$$\delta_{IJ} = \lambda^2_{(1)}M^{(1)}_{IJ} + \lambda^2_{(2)}M^{(2)}_{IJ} + \lambda^2_{(3)}M^{(3)}_{IJ} = \lambda^2_A \left(M^{(A)}_{IJ}\right)_A \tag{106.89}$$

since, from the orthogonal properties of eigenvectors

$$\delta_{IJ} = \sum_{A=1}^{3} N^{(A)}_I\, N^{(A)}_J = \left(N^{(A)}_I\right)_A \left(N^{(A)}_J\right)_A \tag{106.90}$$

We are now in a position to define the *Simo–Serrin* fourth order tensor $\mathcal{M}_{IJKL}$ as:

$$\begin{aligned} \mathcal{M}^{(A)}_{IJKL} \overset{\text{def}}{=} & \frac{\partial M^{(A)}_{IJ}}{\partial C_{KL}} = \\ & \frac{1}{D_{(A)}} \left(I_{IJKL} - \delta_{KL}\delta_{IJ} + \lambda^2_{(A)} \left(\delta_{IJ}\, M^{(A)}_{KL} + M^{(A)}_{IJ}\,\delta_{KL}\right) + \right. \\ + \; & I_3\lambda^{-2}_{(A)} \left((C^{-1})_{IJ}(C^{-1})_{KL} + \frac{1}{2}\left((C^{-1})_{IK}(C^{-1})_{JL} + (C^{-1})_{IL}(C^{-1})_{JK}\right)\right) - \\ - \; & \left. \lambda^{-2}_{(A)}\, I_3 \left((C^{-1})_{IJ}\, M^{(A)}_{KL} + M^{(A)}_{IJ}\,(C^{-1})_{KL}\right) - D'_{(A)}\, M^{(A)}_{IJ}\, M^{(A)}_{KL}\right) \end{aligned} \tag{106.91}$$

Complete derivation of $\mathcal{M}_{IJKL}$ is given in Appendix (704.2).

---

[16] See Runesson (1996).

106.3.5 Stress Measures

In Section (106.3.1) we have defined various stress measures in terms of derivatives of the free energy function W. With the free energy function decomposition, as defined in equation (106.84) we can appropriately decompose all the previously defined stress measures:

- 2. Piola–Kirchhoff stress tensor:

$$\begin{aligned} S_{IJ} &= 2\frac{\partial W}{\partial C_{IJ}} = 2\frac{\partial^{iso} W}{\partial C_{IJ}} + 2\frac{\partial^{vol} W}{\partial C_{IJ}} \\ &= {}^{iso}S_{IJ} + {}^{vol}S_{IJ} \end{aligned} \tag{106.92}$$

- Mandel stress tensor:

$$\begin{aligned} T_{IJ} &= C_{IK}S_{KJ} = 2C_{IK}\frac{\partial W}{\partial C_{KJ}} = 2C_{IK}\frac{\partial^{iso} W}{\partial C_{KJ}} + 2C_{IK}\frac{\partial^{vol} W}{\partial C_{KJ}} \\ &= {}^{iso}T_{IJ} + {}^{vol}T_{IJ} \end{aligned} \tag{106.93}$$

- 1. Piola–Kirchhoff stress tensor

$$\begin{aligned} P_{iJ} &= S_{IJ}(F_{iI})^t = 2\frac{\partial W}{\partial C_{IJ}}(F_{iI})^t = 2\frac{\partial^{iso} W}{\partial C_{IJ}}(F_{iI})^t + 2\frac{\partial^{vol} W}{\partial C_{IJ}}(F_{iI})^t \\ &= {}^{iso}P_{iJ} + {}^{vol}P_{iJ} \end{aligned} \tag{106.94}$$

- Kirchhoff stress tensor

$$\begin{aligned} \tau_{ab} &= 2\frac{\partial W}{\partial e_{ij}} = F_{aI}(F_{bJ})^t S_{IJ} = 2F_{aI}(F_{bJ})^t\frac{\partial^{iso} W}{\partial C_{IJ}} + 2F_{aI}(F_{bJ})^t\frac{\partial^{vol} W}{\partial C_{IJ}} \\ &= F_{aI}(F_{bJ})^{t\,iso}S_{IJ} + F_{aI}(F_{bJ})^{t\,vol}S_{IJ} \\ &= {}^{iso}\tau_{ab} + {}^{vol}\tau_{ab} \end{aligned} \tag{106.95}$$

The derivative of the volumetric part of the free energy function is

$$\frac{\partial^{vol}W(J)}{\partial C_{IJ}} = \frac{\partial^{vol}W(J)}{\partial J}\,\frac{\partial J}{\partial C_{IJ}} = \frac{1}{2}\,\frac{\partial^{vol}W(J)}{\partial J}\;J\;(C^{-1})_{IJ} \tag{106.96}$$

where equation (704.9) was used, while the derivative of the isochoric part of the free energy function yields

$$\frac{\partial^{iso}W(\tilde{\lambda}_{(A)})}{\partial C_{IJ}} = \frac{\partial^{iso}W(\tilde{\lambda}_{(A)})}{\partial \lambda_{(A)}}\,\frac{\partial \lambda_{(A)}}{\partial C_{IJ}} = \frac{1}{2}\frac{\partial^{iso}W(\lambda_{(A)})}{\partial \lambda_{(A)}}\lambda_{(A)}(M^{(A)}_{IJ})_A = \frac{1}{2}w_A(M^{(A)}_{IJ})_A \tag{106.97}$$

where equation (704.7) was used and w_A is derived in Appendix 704.5 as:

$$w_A = \frac{\partial^{iso}W(\lambda_{(A)})}{\partial \tilde{\lambda}_B}\,\frac{\partial \tilde{\lambda}_B}{\partial \lambda_{(A)}}\;\tilde{\lambda}_{(A)} = -\frac{1}{3}\,\frac{\partial^{iso}W(\tilde{\lambda}_{(A)})}{\partial \tilde{\lambda}_B}\;\tilde{\lambda}_B + \frac{\partial^{iso}W(\tilde{\lambda}_{(A)})}{\partial \tilde{\lambda}_{(A)}}\;\tilde{\lambda}_{(A)} \tag{106.98}$$

The decomposed 2. Piola–Kirchhoff stress tensor is

$$\begin{aligned} S_{IJ} &= {}^{vol}S_{IJ} + {}^{iso}S_{IJ} \\ &= \frac{\partial^{vol}W(J)}{\partial J}\;J\;(C^{-1})_{IJ} + w_A\;(M^{(A)}_{IJ})_A \end{aligned} \tag{106.99}$$

The derivative of the free energy is then:

$$\begin{aligned} \frac{\partial W(\lambda_{(A)})}{\partial C_{IJ}} &= \frac{\partial^{vol}W(\lambda_{(A)})}{\partial C_{IJ}} + \frac{\partial^{iso}W(\lambda_{(A)})}{\partial C_{IJ}} \\ &= \frac{1}{2}\,\frac{\partial^{vol}W(J)}{\partial J}\;J\;(C^{-1})_{IJ} + \frac{1}{2}\;w_A\;(M^{(A)}_{IJ})_A \end{aligned} \tag{106.100}$$

It is obvious that the only material dependent parts are derivatives in the form $\partial^{vol}W/\partial J$ and w_A, while the rest is independent of which hyperelastic material model we choose.

106.3.6 Tangent Stiffness Operator

The free energy function decomposition (106.84) is used together with the appropriate definitions made in section (106.3.1) toward the tangent stiffness operator decomposition

$$\mathcal{L}_{IJKL} = {}^{vol}\mathcal{L}_{IJKL} + {}^{iso}\mathcal{L}_{IJKL} = 4\,\frac{\partial^2\left({}^{vol}W\right)}{\partial C_{IJ}\;\partial C_{KL}} + 4\,\frac{\partial^2\left({}^{iso}W\right)}{\partial C_{IJ}\;\partial C_{KL}} \tag{106.101}$$

The volumetric part $\partial^2\left({}^{vol}W\right)/(\partial C_{IJ}\;\partial C_{KL})$ can be written as:

$$\frac{\partial^{2\,vol}W}{\partial C_{IJ}\;\partial C_{KL}} = \frac{1}{4}\left(J^2\frac{\partial^2\left({}^{vol}W\right)}{\partial J \partial J} + J\frac{\partial\left({}^{vol}W\right)}{\partial J}\right)(C^{-1})_{KL}(C^{-1})_{IJ} + \frac{1}{2}J\frac{\partial\left({}^{vol}W\right)}{\partial J}I^{(C^{-1})}_{IJKL} \tag{106.102}$$

and the complete derivation is again given in appendix 704.3.

The isochoric part $\partial^2 \left({}^{iso}W\right)/(\partial C_{IJ}\ \partial C_{KL})$ can be written in the following form:

$$\frac{\partial^{2\,iso}W(\lambda_{(A)})}{\partial C_{IJ}\partial C_{KL}} = \frac{1}{4}\ Y_{AB}\ (M^{(B)}_{KL})_B\ (M^{(A)}_{IJ})_A + \frac{1}{2}\ w_A\ (\mathcal{M}^{(A)}_{IJKL})_A \tag{106.103}$$

and the complete derivation is given in the appendix (704.4).

Finally, one can write the volumetric and isochoric parts of the tangent stiffness tensors as:

$${}^{vol}\mathcal{L}_{IJKL} =$$

$$J^2 \frac{\partial^{2\,vol}W(J)}{\partial J \partial J}(C^{-1})_{KL}(C^{-1})_{IJ} + J\frac{\partial^{vol}W(J)}{\partial J}(C^{-1})_{KL}(C^{-1})_{IJ} + 2J\frac{\partial^{vol}W(J)}{\partial J} I^{(C^{-1})}_{IJKL} \tag{106.104}$$

$$\mathcal{L}^{iso}_{IJKL} = Y_{AB}\ (M^{(B)}_{KL})_B\ (M^{(A)}_{IJ})_A + 2\ w_A\ (\mathcal{M}^{(A)}_{IJKL})_A \tag{106.105}$$

In a similar manner to the stress definitions it is clear that the only material model dependent parts are Y_{AB} and w_A. The remaining second and fourth order tensors $M^{(A)}_{IJ}$ and $\mathcal{M}^{(A)}_{IJKL}$ are independent of the choice of the material model. This observation has a practical consequence in that it is possible to create a *template derivations* for various hyperelastic isotropic material models. Only first and second derivatives of strain energy function with respect to isochoric principal stretches ($\tilde{\lambda}_A$) and Jacobian (J) are needed in addition to the independent tensors, for the determination of various stress and tangent stiffness tensors.

106.3.7 Isotropic Hyperelastic Models

The strain energy function for isotropic solid in terms of principal stretches is represented as:

$$W = W(\lambda_1, \lambda_2, \lambda_3) \tag{106.106}$$

The only restriction is that W is a symmetric function of $\lambda_1, \lambda_2, \lambda_3$, although an appropriate natural configuration condition requires that:

$$W(1,1,1) = 0 \qquad \text{and} \qquad \frac{\partial W(1,1,1)}{\partial \lambda_i} = 0 \tag{106.107}$$

The strain energy function W can either be regarded as a function of principal stretches or the principal invariants of stretches[17]:

$$\begin{aligned} I_1 &= \lambda_1^2 + \lambda_2^2 + \lambda_3^2 \\ I_2 &= \lambda_2^2\lambda_3^2 + \lambda_3^2\lambda_1^2 + \lambda_1^2\lambda_2^2 \\ I_3 &= \lambda_1^2\lambda_2^2\lambda_3^2 \end{aligned} \tag{106.108}$$

---

[17] See also equation (106.61).

A slightly more general formulation is obtained by using principal stretches in the strain energy function definition. A widely exploited family of compressible hyperelastic models[18] are defined (Ogden, 1984) as an infinite series in powers of $(I_1 - 3), (I_2 - 3)$ and $(I_3 - 1)$ as:

$$W = \sum_{p,q,r=0}^{N \to \infty} c_{pqr} \; (I_1 - 3)^p \, (I_2 - 3)^q \, (I_3 - 1)^r \tag{106.109}$$

The regularity condition that W is continuously differentiable an infinitely number of times is satisfied. The requirement that energy vanishes in the reference configuration is met provided $c_{000} = 0$. Reference configuration is stress free iff $c_{100} + 2c_{010} + c_{001} = 0$. Isochoric deviatoric decoupling is possible by setting $c_{pqr} = 0 \;\; (r = 1, 2, 3, ...)$ and $c_{pqr} = 0 \;\; (p, q = 1, 2, 3, ..)$ to obtain:

$$W = {}^{iso}W + {}^{vol}W \tag{106.110}$$

where:

$$\begin{aligned} {}^{iso}W &= \sum_{p,q=0}^{N \to \infty} c_{pq0} \; (I_1 - 3)^p \, (I_2 - 3)^q \\ {}^{vol}W &= \sum_{r=0}^{N \to \infty} c_{00r} \; (I_3 - 1)^r \end{aligned} \tag{106.111}$$

In what follows, we will present a number of widely used strain energy functions for isotropic elastic solids.

106.3.7.1 Ogden Model

A very general set of hyperelastic models was defined by Ogden (1984). The strain energy is expressed as a function of principal stretches as:

$$W = \sum_{r=1}^{N \to \infty} \frac{c_r}{\mu_r} \left(\lambda_1^{\mu_r} + \lambda_2^{\mu_r} + \lambda_3^{\mu_r} - 3 \right) \tag{106.112}$$

The isochoric strain energy function can be written as:

$${}^{iso}W = \sum_{r=1}^{N \to \infty} \frac{c_r}{\mu_r} \left(\tilde{\lambda}_1^{\mu_r} + \tilde{\lambda}_2^{\mu_r} + \tilde{\lambda}_3^{\mu_r} - 1 \right) \tag{106.113}$$

where the following was used $\tilde{\lambda}_i = J^{-\frac{1}{3}} \lambda_i$.

Derivatives needed for building tensors w_A and Y_{AB} are given by the following formulae:

$$\frac{\partial\, {}^{iso}W}{\partial \tilde{\lambda}_A} = \sum_{r=1}^{N \to \infty} c_r \left(\tilde{\lambda}_A \right)^{\mu_r - 1} \tag{106.114}$$

---

[18] Used mainly for rubber–like materials.

$$\frac{\partial^2 \left({}^{iso}W\right)}{\partial \tilde{\lambda}_A^2} = \sum_{r=1}^{N\to\infty} c_r \;\; (\mu_r - 1) \;\; \left(\tilde{\lambda}_A\right)^{\mu_r - 2} \tag{106.115}$$

$$\frac{\partial^2 \left({}^{iso}W\right)}{\partial \tilde{\lambda}_A \partial \tilde{\lambda}_B} = 0 \tag{106.116}$$

106.3.7.2 Neo–Hookean Model

The general isotropic hyperelastic model defined in terms of invariants of principal stretches contains the Neo–Hookean model as special cases. The isochoric part of Neo–Hookean isotropic elastic model can be obtained by selecting $N = 1$, $q = 0$, $c_{p00} = G/2$, to get:

$${}^{iso}W = \frac{G}{2} \left(\tilde{\lambda}_1^2 + \tilde{\lambda}_2^2 + \tilde{\lambda}_3^2 - 3\right) \tag{106.117}$$

while the volumetric part can be defined by choosing $N = 2$, $c_{001} = 0$, $c_{002} = K_b/2$, as:

$${}^{vol}W = \frac{K_b}{2} \left(\lambda_1^2 \lambda_2^2 \lambda_3^2 - 1\right)^2 = \frac{K_b}{2} \left(J^2 - 1\right)^2 \tag{106.118}$$

where G and K_b are the shear and bulk moduli respectively.

Derivatives needed for building tensors w_A and Y_{AB} are given by the following formulae:

$$\frac{\partial^{iso}W}{\partial \tilde{\lambda}_A} = G \; \tilde{\lambda}_A \tag{106.119}$$

$$\frac{\partial^2 \left({}^{iso}W\right)}{\partial \tilde{\lambda}_A^2} = G \tag{106.120}$$

$$\frac{\partial^2 \left({}^{iso}W\right)}{\partial \tilde{\lambda}_A \partial \tilde{\lambda}_B} = 0 \tag{106.121}$$

106.3.7.3 Mooney–Rivlin Model

Mooney proposed a strain energy function for isochoric behavior of the form:

$$\begin{aligned} {}^{iso}W &= \sum_{n=0}^{N\to\infty} \left(a_n \left(\tilde{\lambda}_1^{2n} + \tilde{\lambda}_2^{2n} + \tilde{\lambda}_3^{2n} - 3\right) + a_n \left(\tilde{\lambda}_1^{-2n} + \tilde{\lambda}_2^{-2n} + \tilde{\lambda}_3^{-2n} - 3\right)\right) \\ &= \sum_{n=0}^{N\to\infty} \left(a_n \left(\tilde{\lambda}_1^{2n} + \tilde{\lambda}_2^{2n} + \tilde{\lambda}_3^{2n} - 3\right)\right. \\ &\quad \left. + \, a_n \left(\left(\tilde{\lambda}_2 \tilde{\lambda}_3\right)^{-2n} + \left(\tilde{\lambda}_3 \tilde{\lambda}_1\right)^{-2n} + \left(\tilde{\lambda}_1 \tilde{\lambda}_2\right)^{-2n} - 3\right)\right) \end{aligned} \tag{106.122}$$

with a_n and b_n being the material parameters and a volume preserving constrain $\lambda_a = 1/(\lambda_b \lambda_c$, and a, b, c are cyclic permutations of $(1, 2, 3)$. A more general form was proposed by Rivlin:

$${}^{iso}W = \sum_{p,q=0}^{N\to\infty} c_{pq} \;\; (I_1 - 3)^p \, (I_2 - 3)^q \tag{106.123}$$

which is actually quite similar to the isochoric part of the general isotropic representation from equation (106.111). Both Mooney and Rivlin strain energy functions become similar, if one chooses to set $N = 1$ and $c_{10} = C_1$ and $c_{01} = C_2$ to obtain:

$$\begin{aligned} {}^{iso}W &= \left(C_1 \left(\tilde{\lambda}_1^2 + \tilde{\lambda}_2^2 + \tilde{\lambda}_3^2 - 3 \right) + C_2 \left(\tilde{\lambda}_1^{-2} + \tilde{\lambda}_2^{-2} + \tilde{\lambda}_3^{-2} - 3 \right) \right) \\ &= \left(C_1 \left(\tilde{I}_1 - 3 \right) + C_2 \left(\tilde{I}_2 - 3 \right) \right) \end{aligned} \tag{106.124}$$

Derivatives needed for building tensors w_A and Y_{AB} are given by the following formulae:

$$\frac{\partial {}^{iso}W}{\partial \tilde{\lambda}_A} = 2\, C_1\, \tilde{\lambda}_A - 2\, C_2\, \tilde{\lambda}_A^{-3} \tag{106.125}$$

$$\frac{\partial^2 \left({}^{iso}W \right)}{\partial \tilde{\lambda}_A^2} = 2\, C_1 + 6\, C_2\, \tilde{\lambda}_A^{-4} \tag{106.126}$$

$$\frac{\partial^2 \left({}^{iso}W \right)}{\partial \tilde{\lambda}_A \partial \tilde{\lambda}_B} = 0 \tag{106.127}$$

106.3.7.4 Logarithmic Model

By choosing an alternative set of isochoric principal stretch invariants in the form:

$$\begin{aligned} \tilde{I}_1^{ln} &= 2 \left(\ln \tilde{\lambda}_1 \right)^2 + 2 \left(\ln \tilde{\lambda}_2 \right)^2 + 2 \left(\ln \tilde{\lambda}_3 \right)^2 \\ &= \left(\tilde{\lambda}_1^{\ln} \right)^2 + \left(\tilde{\lambda}_2^{\ln} \right)^2 + \left(\tilde{\lambda}_3^{\ln} \right)^2 \\[1em] \tilde{I}_2^{ln} &= 4 \left(\ln \tilde{\lambda}_2 \right)^2 \left(\ln \tilde{\lambda}_3 \right)^2 + 4 \left(\ln \tilde{\lambda}_3 \right)^2 \left(\ln \tilde{\lambda}_1 \right)^2 + 4 \left(\ln \tilde{\lambda}_1 \right)^2 \left(\ln \tilde{\lambda}_2 \right)^2 \\ &= \left(\tilde{\lambda}_2^{\ln} \right)^2 \left(\tilde{\lambda}_3^{\ln} \right)^2 + \left(\tilde{\lambda}_3^{\ln} \right)^2 \left(\tilde{\lambda}_1^{\ln} \right)^2 + \left(\tilde{\lambda}_1^{\ln} \right)^2 \left(\tilde{\lambda}_2^{\ln} \right)^2 \end{aligned} \tag{106.128}$$

where the isochoric logarithmic stretch $\tilde{\lambda}_i^{\ln}$ was used:

$$\tilde{\lambda}_i^{\ln} = \sqrt{2} \ln \tilde{\lambda}_i = \frac{1}{\sqrt{2}} \ln \tilde{\lambda}_i^2 \tag{106.129}$$

The general representation of the isochoric part of the strain energy function in terms of $\tilde{I}_1^{ln}$ and $\tilde{I}_2^{ln}$ was proposed by Simo and Miehe (1992). A somewhat simpler isochoric strain energy function can be presented in the form:

$${}^{iso}W = G \left(\left(\ln \tilde{\lambda}_1 \right)^2 + \left(\ln \tilde{\lambda}_2 \right)^2 + \left(\ln \tilde{\lambda}_3 \right)^2 \right) \tag{106.130}$$

while the volumetric part is suggested in the form:

$${}^{vol}W = \frac{K_b}{2} \left(\ln J \right)^2 \tag{106.131}$$

Derivatives needed for building tensors w_A and Y_{AB} are given by the following formulae:

$$\frac{\partial^{iso} W}{\partial \tilde{\lambda}_A} = 2\ G\ \left(\tilde{\lambda}_A\right)^{-1} \tag{106.132}$$

$$\frac{\partial^2\left({}^{iso}W\right)}{\partial \tilde{\lambda}_A^2} = -2\ G\ \left(\tilde{\lambda}_A\right)^{-2} \tag{106.133}$$

$$\frac{\partial^2\left({}^{iso}W\right)}{\partial \tilde{\lambda}_A \partial \tilde{\lambda}_B} = 0 \tag{106.134}$$

$$\frac{d\left({}^{vol}W\right)}{dJ} = K_b\ J^{-1} \ln J \tag{106.135}$$

$$\frac{d^2\left({}^{vol}W\right)}{dJ^2} = K_b\ J^{-2} - K_b\ J^{-2} \ln J \tag{106.136}$$

106.3.7.5 Simo–Pister Model

Another form or a volumetric part of strain energy function was proposed by Simo and Pister (1984) in the form:

$$W_{vol}(J) = \frac{1}{4}\ K_b\ \left(J^2 - 1 - 2 \ln J\right) \tag{106.137}$$

The first and second derivatives with respect to J are then given as:

$$\frac{d^{vol}W(J)}{dJ} = \frac{\left(\frac{-2}{J} + 2\,J\right)\ K_b}{4} \tag{106.138}$$

$$\frac{d^{2\,vol}W(J)}{dJ^2} = \frac{\left(2 + \frac{2}{J^2}\right)\ K_b}{4} \tag{106.139}$$

106.4 Finite Deformation Hyperelasto–Plasticity

106.4.1 Introduction

The mathematical structure and numerical analysis of classical small deformation elasto–plasticity is generally well established. However, development of large deformation elastic–plastic algorithms for isotropic and anisotropic material models is still a research area. Here, we present a new integration algorithm, based on the multiplicative decomposition of the deformation gradient into elastic and plastic parts. The algorithm is novel in that it is designed to be used with isotropic as well as anisotropic material models. Consistent derivation is based on the idea from the infinitesimal strain algorithm developed earlier by Jeremić and Sture (1997). The algorithm is not an extension of earlier developments, but rather a novel development which consistently utilizes Newton's method for numerical solution scheme for integrating pertinent constitutive equations. It is also shown that in the limit, the proposed algorithm reduces to the small strain counterpart.

In what follows, we briefly introduce the multiplicative decomposition of the deformation gradient and pertinent constitutive relations. We then proceed to present the numerical algorithm and the algorithmic tangent stiffness tensor consistent with the presented algorithm.

106.4.2 Kinematics

An appropriate generalization of the additive strain decomposition is the multiplicative decomposition of displacement gradient. The motivation for the multiplicative decomposition can be traced back to the early works of Bilby et al. (1957), and Kröner (1960) on micromechanics of crystal dislocations and application to continuum modeling. In the context of large deformation elastoplastic computations, the work by Lee and Liu (1967), Fox (1968) and Lee (1969) stirred an early interest in multiplicative decomposition.

The appropriateness of multiplicative decomposition technique for soils may be justified from the particulate nature of the material. From the micromechanical point of view, plastic deformation in soils arises from slipping, crushing, yielding and plastic bending[19] of granules comprising the assembly[20]. It can certainly be argued that deformations in soils are predominantly plastic, however, reversible deformations could develop from the elasticity of soil grains, and could be relatively large when particles are locked in high density specimens.

The reasoning behind multiplicative decomposition is a rather simple one. If an infinitesimal neighborhood of a body $x_i, x_i + dx_i$ in an inelastically deformed body is cut–out and unloaded to an unstressed configuration, it would be mapped into $\hat{x}_i, \hat{x}_i + d\hat{x}_i$. The transformation would be comprised of a rigid body displacement[21] and purely elastic unloading. The elastic unloading is a fictitious one, since materials with a strong Baushinger's effect, unloading will lead to loading in an other stress direction, and, if there are residual stresses, the body must be cut–out in small pieces and then every piece relieved of stresses. The unstressed configuration is thus incompatible and discontinuous. The position $\hat{x}_i$ is arbitrary, and we may assume a linear relationship between dx_i and $d\hat{x}_i$, in the form[22]:

---

[19] For plate like clay particles.

[20] See also Lambe and Whitman (1979) and Sture (1993).

[21] Translation and rotation.

[22] referred to same Cartesian coordinate system.

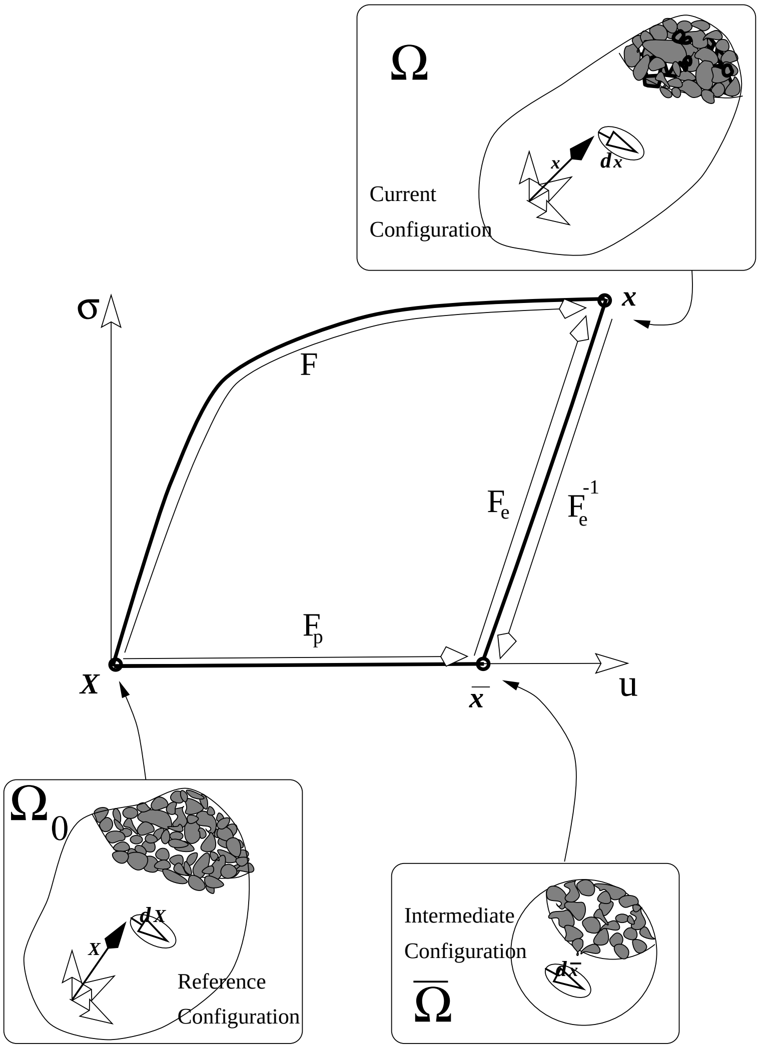


Figure 106.5: Multiplicative decomposition of deformation gradient: schematics.

$$d\hat{x}_k = \left(F^e_{ik}\right)^{-1} dx_i \tag{106.140}$$

where $\left(F^e_{ik}\right)^{-1}$ is not to be understood as a deformation gradient, since it may represent the incompatible, discontinuous deformation of a body. By considering the reference configuration of a body dX_i, then the connection to the current configuration is[23]:

$$dx_k = F_{ki} dX_i \Rightarrow d\hat{x}_k = \left(F^e_{ik}\right)^{-1} F_{ij} dX_j \tag{106.141}$$

---

[23] See section 106.2.2.

so that one can define:

$$F^p_{kj} \overset{def}{=} \left(F^e_{ik}\right)^{-1} F_{ij} \Rightarrow F_{ij} \overset{\text{def}}{=} F^e_{ki} F^p_{kj} \tag{106.142}$$

The plastic part of the deformation gradient, F^p_{kj} represents micro–mechanically, the irreversible process of slipping, crushing dislocation and macroscopically the irreversible plastic deformation of a body. The elastic part, F^e_{ki} represents micro–mechanically a pure elastic reversal of deformation for the particulate assembly, macroscopically a linear elastic unloading toward a stress free state of the body, not necessarily a compatible, continuous deformation but rather a fictitious elastic unloading of small cut outs of a deformed particulate assembly or continuum body.

106.4.3 Constitutive Relations

We propose the free energy density W, which is defined in $\bar{\Omega}$, as follows

$$\rho_0 W(\bar{C}^e_{ij}, \kappa_\alpha) = \rho_0 W^e(\bar{C}^e_{ij}) + \rho_0 W^p(\kappa_\alpha) \tag{106.143}$$

where $W^e(\bar{C}^e_{ij})$ represents a suitable hyperelastic model in terms of the elastic right deformation tensor $\bar{C}^e_{ij}$, whereas $W^p(\kappa_\alpha)$ represents the hardening. It has been shown elsewhere (Runesson, 1996), that the pertinent dissipation inequality becomes

$$D = \bar{T}_{ij}\ \bar{L}^p_{ij} + \sum_\alpha \bar{K}_\alpha\ \dot{\kappa}_\alpha \geq 0 \tag{106.144}$$

where $\bar{T}_{ij}$ is the Mandel stress[24] and $\bar{L}^p_{ij}$ is the plastic velocity gradient defined on $\bar{\Omega}$.

We now define elastic domain $\mathcal{B}$ as

$$\mathcal{B} = \{\bar{T}_{ij}, \bar{K}_\alpha \mid \Phi(\bar{T}_{ij}, \bar{K}_\alpha) \leq 0\} \tag{106.145}$$

When Φ is isotropic in $\bar{T}_{ij}$ (which is the case here) in conjunction with elastic isotropy, we can conclude that $\bar{T}_{ij}$ is symmetrical and we may replace $\bar{T}_{ij}$ by τ_{ij} in Φ.

As to the choice of elastic law, it is emphasized that this is largely a matter of convenience since we shall be dealing with small elastic deformations. Here, the Neo–Hookean elastic law is adopted. The generic situation is $\bar{T}_{ij} = \bar{T}_{ij}(\tilde{\bar{U}}^e_{kl}, \bar{J}^e)$, where we have used the isochoric/volumetric split of the elastic right stretch tensor as $\bar{U}^e_{kl} = \tilde{\bar{U}}^e_{kl}\ (\bar{J}^e)^{1/3}$.

The constitutive relations can now be written as

$$\bar{T}_{ij} = \bar{T}_{ij}(\tilde{\bar{U}}^e_{kl}, \bar{J}^e) \tag{106.146}$$

$$\bar{L}^p_{ij} := \dot{F}^p_{ik}\ \left(F^p_{jk}\right)^{-1} = \dot{\mu}\frac{\partial \Phi^*}{\partial \bar{T}_{ij}} = \dot{\mu}\bar{M}_{ij} \tag{106.147}$$

---

[24] See section 106.3.1.

$$\bar{K}_\alpha = \bar{K}_\alpha(\bar{\kappa_\beta}) \tag{106.148}$$

$$\dot{\bar{\kappa}}_\beta = \dot{\mu} \, \frac{\partial \Phi^*}{\partial K_\beta} \quad , \quad \kappa_\beta(0) = 0 \tag{106.149}$$

where $F^p_{ik} = (\bar{F}^e_{li})^{-1} F_{lk}$ is the plastic part of the deformation gradient.

106.4.4 Implicit Integration Algorithm

The incremental deformation and plastic flow are governed by the system of evolution equations (106.147) and (106.149):

$$\dot{F}^p_{ik} \, \left(F^p_{jk}\right)^{-1} = \dot{\mu}\bar{M}_{ij} \tag{106.150}$$

$$\dot{\bar{\kappa}}_\beta = \dot{\mu} \, \frac{\partial \Phi^*}{\partial K_\beta} \quad , \quad \kappa_\beta(0) = 0 \tag{106.151}$$

The flow rule from equation (106.150) can be integrated to give

$$^{n+1}F^p_{ij} = \exp\left(\Delta\mu^{n+1}\bar{M}_{ik}\right) {}^nF^p_{kj} \tag{106.152}$$

By using the multiplicative decomposition

$$F_{ij} = \bar{F}^e_{ik} \, F^p_{kj} \Rightarrow \bar{F}^e_{ik} = F_{ij} \, \left(F^p_{kj}\right)^{-1} \tag{106.153}$$

and equation (106.152) we obtain

$$\begin{aligned} {}^{n+1}\bar{F}^e_{ij} &= {}^{n+1}F_{im} \, \left({}^nF^p_{mk}\right)^{-1} \exp\left(-\Delta\mu^{n+1}\bar{M}_{kj}\right) \\ &= {}^{n+1}\bar{F}^{e,tr}_{ik} \exp\left(-\Delta\mu^{n+1}\bar{M}_{kj}\right) \end{aligned} \tag{106.154}$$

where we used that

$$^{n+1}\bar{F}^{e,tr}_{ik} = {}^{n+1}F_{im} \, \left({}^nF^p_{km}\right)^{-1} \tag{106.155}$$

The elastic deformation is then

$$\begin{aligned} {}^{n+1}\bar{C}^e_{ij} &\overset{\text{def}}{=} \left({}^{n+1}\bar{F}^e_{im}\right)^T \, {}^{n+1}\bar{F}^e_{mj} \\ &= \exp\left(-\Delta\mu^{n+1}\bar{M}^T_{ir}\right) \, \left({}^{n+1}\bar{F}^{e,tr}_{rk}\right)^T \, {}^{n+1}\bar{F}^{e,tr}_{kl} \, \exp\left(-\Delta\mu^{n+1}\bar{M}_{lj}\right) \\ &= \exp\left(-\Delta\mu^{n+1}\bar{M}^T_{ir}\right) \, {}^{n+1}\bar{C}^{e,tr}_{rl} \, \exp\left(-\Delta\mu^{n+1}\bar{M}_{lj}\right) \end{aligned} \tag{106.156}$$

By recognizing that the exponent of a tensor can be expanded in Taylor series[25]

$$\exp\left(-\Delta\mu^{n+1}\bar{M}_{lj}\right) = \delta_{lj} - \Delta\mu^{n+1}\bar{M}_{lj} + \frac{1}{2}\left(\Delta\mu^{n+1}\bar{M}_{ls}\right)\left(\Delta\mu^{n+1}\bar{M}_{sj}\right) + \cdots \tag{106.157}$$

---

[25] See for example Pearson (1974).

and by using the first order expansion in the equation (106.156), we obtain

$$\begin{aligned} {}^{n+1}\bar{C}^{e}_{ij} &= \left(\delta_{ir} - \Delta\mu^{n+1}\bar{M}_{ir}\right) {}^{n+1}\bar{C}^{e,tr}_{rl} \left(\delta_{lj} - \Delta\mu^{n+1}\bar{M}_{lj}\right) \\ &= \left({}^{n+1}\bar{C}^{e,tr}_{il} - \Delta\mu^{n+1}\bar{M}_{ir}\;{}^{n+1}\bar{C}^{e,tr}_{rl}\right)\left(\delta_{lj}-\Delta\mu^{n+1}\bar{M}_{lj}\right) \\ &= {}^{n+1}\bar{C}^{e,tr}_{ij} - \Delta\mu^{n+1}\bar{M}_{ir}\;{}^{n+1}\bar{C}^{e,tr}_{rj} - \Delta\mu\;{}^{n+1}\bar{C}^{e,tr}_{il}\;{}^{n+1}\bar{M}_{lj} \\ &\quad +\Delta\mu^2\;{}^{n+1}\bar{M}_{ir}\;{}^{n+1}\bar{C}^{e,tr}_{rl}\;{}^{n+1}\bar{M}_{lj} \end{aligned} \tag{106.158}$$

Remark 106.4.1 The Taylor's series expansion from equation (106.157) is a proper approximation for the general nonsymmetric tensor $\bar{M}_{lj}$. That is, the approximate solution given by equation (106.158) is valid for a general anisotropic solid. This contrasts with the spectral decomposition family of solutions[26] which are restricted to isotropic solids.

Remark 106.4.2 Taylor's series expansion[27] is proper for "small" values of plastic flow tensor $\Delta\mu^{n+1}\bar{M}_{lj}$. This is indeed the case for small increments, when $\Delta\mu \to 0$ which are required for following the equilibrium path for path–dependent solids.

Remark 106.4.3 In the limit, when the displacements are sufficiently small, the solution (106.158) collapses to

$$\begin{aligned} \lim_{F_{ij}\to\delta_{ij}} \delta_{ij} + 2^{n+1}\epsilon_{ij} = &+ \;\delta_{ij} + 2^{n+1}\epsilon^{e,tr}_{ij} \\ &- \;\Delta\mu^{n+1}\bar{M}_{ir}\left(\delta_{rj} + 2^{n+1}\epsilon^{e,tr}_{rj}\right) \\ &- \;\Delta\mu\left(\delta_{il} + 2^{n+1}\epsilon^{e,tr}_{il}\right)\;{}^{n+1}\bar{M}_{lj} \\ &+ \;\Delta\mu^2\;{}^{n+1}\bar{M}_{ir}\left(\delta_{rl} + 2^{n+1}\epsilon^{e,tr}_{rl}\right)\;{}^{n+1}\bar{M}_{lj} \\ = &+ \;\delta_{ij} + 2^{n+1}\epsilon^{e,tr}_{ij} \\ &- \;\Delta\mu^{n+1}\bar{M}_{ij} - 2\Delta\mu^{n+1}\bar{M}_{ir}{}^{n+1}\epsilon^{tr}_{rj} \\ &- \;\Delta\mu\;{}^{n+1}\bar{M}_{ij} - 2\Delta\mu\;{}^{n+1}\epsilon^{tr}_{il}\;{}^{n+1}\bar{M}_{lj} \\ &+ \;\Delta\mu^2\;{}^{n+1}\bar{M}_{il}\;{}^{n+1}\bar{M}_{lj} + 2\Delta\mu^2\;{}^{n+1}\bar{M}_{ir}\;{}^{n+1}\epsilon^{tr}_{rl}{}^{n+1}\bar{M}_{lj} \\ &= \;\delta_{ij} + 2^{n+1}\epsilon^{e,tr}_{ij} - 2\Delta\mu\;{}^{n+1}\bar{M}_{ij} \\ \Rightarrow {}^{n+1}\epsilon_{ij} &= \;{}^{n+1}\epsilon^{tr}_{ij} - \Delta\mu\;{}^{n+1}\bar{M}_{ij} \end{aligned} \tag{106.159}$$

which is a small deformation elastic predictor–plastic corrector equation in strain space. In working out

---

[26] See Simo (1992).

[27] It should be called MacLaurin's series expansion, since expansion is about zero plastic flow state (no incremental plastic deformation).

the small deformation counterpart (106.159) it was used that

$$\begin{aligned}
\lim_{F_{ij}\to\delta_{ij}} {}^{n+1}\bar{C}^{e}_{ij} &= \delta_{ij} + 2\,{}^{n+1}\epsilon_{ij} \\
2\Delta\mu\ {}^{n+1}\epsilon^{tr}_{il}\ {}^{n+1}\bar{M}_{lj} &\ll {}^{n+1}\bar{M}_{ij} \\
\Delta\mu &\ll 1
\end{aligned} \tag{106.160}$$

By neglecting the higher order term with $\Delta\mu^2$ in equation (106.158), the solution for the right elastic deformation tensor ${}^{n+1}\bar{C}^{e}_{ij}$ can be written as

$$ {}^{n+1}\bar{C}^{e}_{ij} = {}^{n+1}\bar{C}^{e,tr}_{ij} - \Delta\mu \left({}^{n+1}\bar{M}_{ir}\ {}^{n+1}\bar{C}^{e,tr}_{rj} + {}^{n+1}\bar{C}^{e,tr}_{il}\ {}^{n+1}\bar{M}_{lj} \right) \tag{106.161}$$

The hardening rule (106.151) can be integrated to give

$$ {}^{n+1}\kappa_{\alpha} = {}^{n}\kappa_{\alpha} + \Delta\mu \left. \frac{\partial\Phi^{*}}{\partial K_{\alpha}} \right|_{n+1} \tag{106.162}$$

Remark 106.4.4 It is interesting to note that equation (106.161) resembles the *elastic predictor–plastic corrector* equation for small deformation elastic–plastic incremental analysis. That resemblance will be used to build an iterative solution algorithm in the next section.

The incremental problem is defined by equations (106.161), (106.162), and the constitutive relations

$$ {}^{n+1}\bar{S}_{IJ} = 2 \left. \frac{\partial W}{\partial C_{IJ}} \right|_{n+1} \tag{106.163}$$

$$ {}^{n+1}K_{\alpha} = - \left. \frac{\partial W}{\partial \kappa_{\alpha}} \right|_{n+1} \tag{106.164}$$

and the Karush–Kuhn–Tucker (KKT) conditions

$$ \Delta\mu < 0 \quad ; \quad {}^{n+1}\Phi \leq 0 \quad ; \quad \Delta\mu\ {}^{n+1}\Phi = 0 \tag{106.165}$$

where

$$ \Phi = \Phi(\bar{T}_{ij}, K_{\alpha}) \tag{106.166}$$

Remark 106.4.5 The Mandel stress tensor $\bar{T}_{ij}$ can be obtained from the second Piola–Kirchhoff stress tensor $\bar{S}_{kj}$ and the right elastic deformation tensor $\bar{C}^{e}_{ik}$ as

$$ \bar{T}_{ij} = \bar{C}^{e}_{ik}\ \bar{S}_{kj} \tag{106.167}$$

This set of nonlinear equations will be solved with a Newton type procedure, described in the next section. For a given ${}^{n+1}F_{ij}$, or ${}^{n+1}\bar{C}^{e,tr}_{ij}$, the upgraded quantities ${}^{n+1}\bar{S}_{IJ}$ and ${}^{n+1}K_{\alpha}$ can be found, then the appropriate pull–back to B_0 or push–forward to B will give ${}^{n+1}S_{IJ}$ and ${}^{n+1}\tau_{ij}$

$$ {}^{n+1}S_{IJ} = \left({}^{n+1}F^{p}_{iI}\right)^{-1} \; {}^{n+1}\bar{S}_{IJ} \; \left({}^{n+1}F^{p}_{jJ}\right)^{-T} \tag{106.168}$$

$$ {}^{n+1}\tau_{ij} = {}^{n+1}\bar{F}^{e}_{iI} \; {}^{n+1}\bar{S}_{IJ} \; \left({}^{n+1}F^{e}_{jJ}\right)^{-1} \tag{106.169}$$

The elastic predictor, plastic corrector equation

$$\begin{aligned} {}^{n+1}\bar{C}^{e}_{ij} &= {}^{n+1}\bar{C}^{e,tr}_{ij} - \Delta\mu \left({}^{n+1}\bar{M}_{ir} \; {}^{n+1}\bar{C}^{e,tr}_{rj} + {}^{n+1}\bar{C}^{e,tr}_{il} \; {}^{n+1}\bar{M}_{lj}\right) \\ &= {}^{n+1}\bar{C}^{e,tr}_{ij} - \Delta\mu \; {}^{n+1}Z_{ij} \end{aligned} \tag{106.170}$$

is used as a starting point for a Newton iterative algorithm. In previous equation, we have introduced tensor Z_{ij} to shorten writing. The trial right elastic deformation tensor is defined as

$$\begin{aligned} {}^{n+1}\bar{C}^{e,tr}_{ij} = \; & \left({}^{n+1}\bar{F}^{e,tr}_{ri}\right)^{T} \left({}^{n+1}\bar{F}^{e,tr}_{rj}\right) \\ & \left({}^{n+1}F_{rM} \; \left({}^{n}F^{p}_{iM}\right)^{-1}\right)^{T} \left({}^{n+1}F_{rS} \; \left({}^{n}F^{p}_{jS}\right)^{-1}\right) \end{aligned} \tag{106.171}$$

We introduce a tensor of deformation residuals

$$ R_{ij} = \underbrace{\bar{C}^{e}_{ij}}_{current} - \underbrace{\left({}^{n+1}\bar{C}^{e,tr}_{ij} - \Delta\mu \; {}^{n+1}Z_{ij}\right)}_{BackwardEuler} \tag{106.172}$$

Tensor R_{ij} represents the difference between the current right elastic deformation tensor and the Backward Euler right elastic deformation tensor. The trial right elastic deformation tensor ${}^{n+1}\bar{C}^{e,tr}_{ij}$ is maintained fixed during the iteration process. The first order Taylor series expansion can be applied to the equation (106.172) in order to obtain the iterative change, the new residual R^{new}_{ij} from the old R^{old}_{ij}

$$ R^{new}_{ij} = R^{old}_{ij} + d\bar{C}^{e}_{ij} + d(\Delta\mu) \; {}^{n+1}Z_{ij} + \Delta\mu \frac{\partial \, {}^{n+1}Z_{ij}}{\partial \bar{T}_{mn}} \; d\bar{T}_{mn} + \Delta\mu \frac{\partial \, {}^{n+1}Z_{ij}}{\partial K_{\alpha}} \; dK_{\alpha} \tag{106.173}$$

By using that

$$ \bar{T}_{mn} = \bar{C}^{e}_{mk} \; \bar{S}_{kn} \qquad \Rightarrow \qquad \left(\bar{C}^{e}_{sk}\right)^{-1} \; \bar{T}_{sn} = \bar{S}_{kn} \tag{106.174}$$

we can write

$$\begin{aligned} d\bar{T}_{mn} &= d\bar{C}^{e}_{mk} \; \bar{S}_{kn} + \bar{C}^{e}_{mk} \; d\bar{S}_{kn} \\ &= d\bar{C}^{e}_{mk} \; \bar{S}_{kn} + \frac{1}{2} \; \bar{C}^{e}_{mk} \; \bar{\mathcal{L}}^{e}_{knpq} \; d\bar{C}^{e}_{pq} \qquad \text{from (106.55)} \\ &= d\bar{C}^{e}_{mk} \; \left(\bar{C}^{e}_{sk}\right)^{-1} \; \bar{T}_{sn} + \frac{1}{2} \; \bar{C}^{e}_{mk} \; \bar{\mathcal{L}}^{e}_{knpq} \; d\bar{C}^{e}_{pq} \qquad \text{from (106.174)} \end{aligned} \tag{106.175}$$

and the equation (106.173) can be rewritten as

$$
\begin{aligned}
R^{new}_{ij} = \quad & R^{old}_{ij} + d\bar{C}^e_{ij} + d(\Delta\mu)\ {}^{n+1}Z_{ij} + \\
+ \quad & \Delta\mu \frac{\partial\, {}^{n+1}Z_{ij}}{\partial \bar{T}_{mn}} \left(d\bar{C}^e_{mk}\ \left(\bar{C}^e_{sk}\right)^{-1}\ \bar{T}_{sn} + \frac{1}{2}\ \bar{C}^e_{mk}\ \bar{\mathcal{L}}^e_{knpq}\ d\bar{C}^e_{pq} \right) + \\
+ \quad & \Delta\mu \frac{\partial\, {}^{n+1}Z_{ij}}{\partial K_\alpha}\ dK_\alpha \\
\\
= \quad & R^{old}_{ij} + d\bar{C}^e_{ij} + d(\Delta\mu)\ {}^{n+1}Z_{ij} + \\
+ \quad & \Delta\mu \frac{\partial\, {}^{n+1}Z_{ij}}{\partial \bar{T}_{mn}}\ d\bar{C}^e_{mk}\ \left(\bar{C}^e_{sk}\right)^{-1}\ \bar{T}_{sn} + \\
+ \quad & \frac{1}{2}\ \Delta\mu \frac{\partial\, {}^{n+1}Z_{ij}}{\partial \bar{T}_{mn}}\ \bar{C}^e_{mk}\ \bar{\mathcal{L}}^e_{knpq}\ d\bar{C}^e_{pq} + \\
+ \quad & \Delta\mu \frac{\partial\, {}^{n+1}Z_{ij}}{\partial K_\alpha}\ dK_\alpha \\
\\
= \quad & R^{old}_{ij} + d\bar{C}^e_{ij} + d(\Delta\mu)\ {}^{n+1}Z_{ij} + \\
+ \quad & \Delta\mu \frac{\partial\, {}^{n+1}Z_{mn}}{\partial \bar{T}_{ik}} \left(\bar{C}^e_{sj}\right)^{-1}\ \bar{T}_{sk}\ d\bar{C}^e_{ij} + \qquad \text{dummy indices rearrangement} \\
+ \quad & \frac{1}{2}\ \Delta\mu \frac{\partial\, {}^{n+1}Z_{pq}}{\partial \bar{T}_{mn}}\ \bar{C}^e_{mk}\ \bar{\mathcal{L}}^e_{knij}\ d\bar{C}^e_{ij} + \qquad \text{dummy indices rearrangement} \\
+ \quad & \Delta\mu \frac{\partial\, {}^{n+1}Z_{ij}}{\partial K_\alpha}\ dK_\alpha
\end{aligned}
\tag{106.176}
$$

The goal is to have $R^{new}_{ij} = 0$ so one can write

$$
\begin{aligned}
0 \;=\;& R^{old}_{ij} + d\bar{C}^e_{ij} + d(\Delta\mu)\; {}^{n+1}Z_{ij} + \\
+\;& \Delta\mu \frac{\partial\, {}^{n+1}Z_{mn}}{\partial \bar{T}_{ik}} \left(\bar{C}^e_{sj}\right)^{-1} \; \bar{T}_{sk}\; d\bar{C}^e_{ij} + \\
+\;& \frac{1}{2}\,\Delta\mu \frac{\partial\, {}^{n+1}Z_{pq}}{\partial \bar{T}_{mn}}\; \bar{C}^e_{mk}\; \bar{\mathcal{L}}^e_{knij}\; d\bar{C}^e_{ij} + \\
+\;& \Delta\mu \frac{\partial\, {}^{n+1}Z_{ij}}{\partial K_\alpha}\; dK_\alpha \\
\\
=\;& R^{old}_{ij} + d(\Delta\mu)\; {}^{n+1}Z_{ij} \\
+\;& \Delta\mu \frac{\partial\, {}^{n+1}Z_{ij}}{\partial K_\alpha}\; dK_\alpha + \\
+\;& d\bar{C}^e_{ij} + \\
+\;& \Delta\mu \frac{\partial\, {}^{n+1}Z_{mn}}{\partial \bar{T}_{ik}} \left(\bar{C}^e_{sj}\right)^{-1} \; \bar{T}_{sk}\; d\bar{C}^e_{ij} + \\
+\;& \frac{1}{2}\,\Delta\mu \frac{\partial\, {}^{n+1}Z_{pq}}{\partial \bar{T}_{mn}}\; \bar{C}^e_{mk}\; \bar{\mathcal{L}}^e_{knij}\; d\bar{C}^e_{ij} \\
\\
=\;& R^{old}_{ij} + d(\Delta\mu)\; {}^{n+1}Z_{ij} + \Delta\mu \frac{\partial\, {}^{n+1}Z_{ij}}{\partial K_\alpha}\; dK_\alpha + \\
+\;& \left(\delta_{im}\delta_{nj} + \Delta\mu \frac{\partial\, {}^{n+1}Z_{mn}}{\partial \bar{T}_{ik}} \left(\bar{C}^e_{sj}\right)^{-1} \; \bar{T}_{sk} + \frac{1}{2}\,\Delta\mu \frac{\partial\, {}^{n+1}Z_{mn}}{\partial \bar{T}_{pq}}\; \bar{C}^e_{pk}\; \bar{\mathcal{L}}^e_{kqij}\;\right) d\bar{C}^e_{ij}
\end{aligned}
\tag{106.177}
$$

Upon introducing notation

$$
\mathcal{T}_{mnij} = \delta_{im}\delta_{nj} + \Delta\mu \frac{\partial\, {}^{n+1}Z_{mn}}{\partial \bar{T}_{ik}} \left(\bar{C}^e_{sj}\right)^{-1} \; \bar{T}_{sk} + \frac{1}{2}\,\Delta\mu \frac{\partial\, {}^{n+1}Z_{mn}}{\partial \bar{T}_{pq}}\; \bar{C}^e_{pk}\; \bar{\mathcal{L}}^e_{kqij}
\tag{106.178}
$$

we can solve (106.177) for $d\bar{C}^e_{ij}$

$$
d\bar{C}^e_{ij} = \left(\mathcal{T}_{mnij}\right)^{-1} \left(-R^{old}_{mn} - d(\Delta\mu)\; {}^{n+1}Z_{mn} - \Delta\mu \frac{\partial\, {}^{n+1}Z_{mn}}{\partial K_\alpha}\; dK_\alpha \right)
\tag{106.179}
$$

or, by rearranging indices

$$
d\bar{C}^e_{pq} = \left(\mathcal{T}_{mnpq}\right)^{-1} \left(-R^{old}_{mn} - d(\Delta\mu)\; {}^{n+1}Z_{mn} - \Delta\mu \frac{\partial\, {}^{n+1}Z_{mn}}{\partial K_\alpha}\; dK_\alpha \right)
\tag{106.180}
$$

By using that

$$
dK_\alpha = \frac{\partial K_\alpha}{\partial \kappa_\beta}\; d\kappa_\beta = -d(\Delta\mu)\; \frac{\partial K_\alpha}{\partial \kappa_\beta}\; \frac{\partial Q}{\partial K_\beta} = -d(\Delta\mu)\; H_{\alpha\beta}\; \frac{\partial Q}{\partial K_\beta}
\tag{106.181}
$$

it follows from (106.180)

$$d\bar{C}^e_{pq} = \left(\mathcal{T}_{mnpq}\right)^{-1} \left(-R^{old}_{mn} - d(\Delta\mu)\ {}^{n+1}Z_{mn} + \Delta\mu \frac{\partial\, {}^{n+1}Z_{mn}}{\partial K_\alpha}\ d(\Delta\mu)\ H_{\alpha\beta}\ \frac{\partial Q}{\partial K_\beta} \right) \tag{106.182}$$

A first order Taylor series expansion of a yield function yields

$$\begin{aligned}
{}^{new}\Phi(\bar{T}_{ij}, K_\alpha) = & \quad {}^{old}\Phi(\bar{T}_{ij}, K_\alpha) + \\
+ & \quad \frac{\partial\Phi(\bar{T}_{ij}, K_\alpha)}{\partial \bar{T}_{mn}}\ d\bar{T}_{mn} \\
+ & \quad \frac{\partial\Phi(\bar{T}_{ij}, K_\alpha)}{\partial K_\alpha}\ dK_\alpha \\
= & \quad {}^{old}\Phi(\bar{T}_{ij}, K_\alpha) + \\
+ & \quad \frac{\partial\Phi(\bar{T}_{ij}, K_\alpha)}{\partial \bar{T}_{mn}} \left(d\bar{C}^e_{mk}\ \left(\bar{C}^e_{sk}\right)^{-1}\ \bar{T}_{sn} + \frac{1}{2}\ \bar{C}^e_{mk}\ \bar{\mathcal{L}}^e_{knpq}\ d\bar{C}^e_{pq} \right) \\
+ & \quad \frac{\partial\Phi(\bar{T}_{ij}, K_\alpha)}{\partial K_\alpha}\ dK_\alpha \\
= & \quad {}^{old}\Phi(\bar{T}_{ij}, K_\alpha) + \\
+ & \quad \frac{\partial\Phi(\bar{T}_{ij}, K_\alpha)}{\partial \bar{T}_{pn}} \left(\bar{C}^e_{sq}\right)^{-1}\ \bar{T}_{sn}\ d\bar{C}^e_{pq} \qquad \text{dummy indices rearrangement} \\
+ & \quad \frac{1}{2}\ \frac{\partial\Phi(\bar{T}_{ij}, K_\alpha)}{\partial \bar{T}_{mn}}\ \bar{C}^e_{mk}\ \bar{\mathcal{L}}^e_{knpq}\ d\bar{C}^e_{pq} \\
+ & \quad \frac{\partial\Phi(\bar{T}_{ij}, K_\alpha)}{\partial K_\alpha}\ dK_\alpha \\
= & \quad {}^{old}\Phi(\bar{T}_{ij}, K_\alpha) + \\
+ & \quad \left(\frac{\partial\Phi(\bar{T}_{ij}, K_\alpha)}{\partial \bar{T}_{pn}} \left(\bar{C}^e_{sq}\right)^{-1}\ \bar{T}_{sn} + \frac{1}{2}\ \frac{\partial\Phi(\bar{T}_{ij}, K_\alpha)}{\partial \bar{T}_{mn}}\ \bar{C}^e_{mk}\ \bar{\mathcal{L}}^e_{knpq} \right)\ d\bar{C}^e_{pq} \\
+ & \quad \frac{\partial\Phi(\bar{T}_{ij}, K_\alpha)}{\partial K_\alpha}\ dK_\alpha
\end{aligned} \tag{106.183}$$

By using (106.181), equation (106.183) becomes

$$\begin{aligned}
{}^{new}\Phi(\bar{T}_{ij}, K_\alpha) = & \quad {}^{old}\Phi(\bar{T}_{ij}, K_\alpha) + \\
+ & \quad \left(\frac{\partial\Phi(\bar{T}_{ij}, K_\alpha)}{\partial \bar{T}_{pn}} \left(\bar{C}^e_{sq}\right)^{-1}\ \bar{T}_{sn} + \frac{1}{2}\ \frac{\partial\Phi(\bar{T}_{ij}, K_\alpha)}{\partial \bar{T}_{mn}}\ \bar{C}^e_{mk}\ \bar{\mathcal{L}}^e_{knpq} \right)\ d\bar{C}^e_{pq} \\
- & \quad d(\Delta\mu)\ \frac{\partial\Phi(\bar{T}_{ij}, K_\alpha)}{\partial K_\alpha}\ H_{\alpha\beta}\ \frac{\partial\Phi^*}{\partial K_\beta}
\end{aligned} \tag{106.184}$$

Upon introducing the following notation

$$\mathcal{F}_{pq} = \frac{\partial\Phi(\bar{T}_{ij}, K_\alpha)}{\partial \bar{T}_{pn}} \left(\bar{C}^e_{sq}\right)^{-1}\ \bar{T}_{sn} + \frac{1}{2}\ \frac{\partial\Phi(\bar{T}_{ij}, K_\alpha)}{\partial \bar{T}_{mn}}\ \bar{C}^e_{mk}\ \bar{\mathcal{L}}^e_{knpq} \tag{106.185}$$

and with the solution for $d\bar{C}^e_{pq}$ from (106.182), (106.184) becomes

$$\begin{aligned}
& {}^{new}\Phi(\bar{T}_{ij}, K_\alpha) = {}^{old}\Phi(\bar{T}_{ij}, K_\alpha) + \\
+ \quad & \mathcal{F}_{pq} \left(\left(\mathcal{T}_{mnpq}\right)^{-1} \left(-R^{old}_{mn} - d(\Delta\mu)\; {}^{n+1}Z_{mn} + d(\Delta\mu)\; \Delta\mu\; \frac{\partial\, {}^{n+1}Z_{mn}}{\partial K_\alpha}\; H_{\alpha\beta}\; \frac{\partial \Phi^*}{\partial K_\beta} \right) \right) \\
- \quad & d(\Delta\mu)\; \frac{\partial \Phi(\bar{T}_{ij}, K_\alpha)}{\partial K_\alpha}\; H_{\alpha\beta}\; \frac{\partial \Phi^*}{\partial K_\beta}
\end{aligned} \tag{106.186}$$

After setting ${}^{new}\Phi(\bar{T}_{ij}, K_\alpha) = 0$ we can solve for the incremental inconsistency parameter $d(\Delta\mu)$

$$\begin{aligned}
& d(\Delta\mu) = \\
& \frac{{}^{old}\Phi - \mathcal{F}_{pq}\; \left(\mathcal{T}_{mnpq}\right)^{-1}\; R^{old}_{mn}}{\mathcal{F}_{pq}\; \left(\mathcal{T}_{mnpq}\right)^{-1}\; {}^{n+1}Z_{mn} - \Delta\mu\; \mathcal{F}_{pq}\; \left(\mathcal{T}_{mnpq}\right)^{-1}\; \frac{\partial\, {}^{n+1}Z_{mn}}{\partial K_\alpha}\; H_{\alpha\beta}\; \frac{\partial \Phi^*}{\partial K_\beta} + \frac{\partial \Phi}{\partial K_\alpha}\; H_{\alpha\beta}\; \frac{\partial \Phi^*}{\partial K_\beta}}
\end{aligned} \tag{106.187}$$

Remark 106.4.6 In the perfectly plastic case, the increment inconsistency parameter $d(\Delta\mu)$ is

$$d(\Delta\mu) = \frac{{}^{old}\Phi - \mathcal{F}_{pq}\; \left(\mathcal{T}_{mnpq}\right)^{-1}\; R^{old}_{mn}}{\mathcal{F}_{pq}\; \left(\mathcal{T}_{mnpq}\right)^{-1}\; {}^{n+1}Z_{mn}} \tag{106.188}$$

Remark 106.4.7 In the limit, for small deformations, isotropic response, the increment inconsistency parameter $d(\Delta\mu)$ becomes

$$d(\Delta\mu) = \frac{{}^{old}\Phi - \left(n_{mn}\; E_{mnpq}\right)\; \left(\delta_{pm}\delta_{nq} + \Delta\mu\; \frac{\partial m_{mn}}{\partial \sigma_{ij}}\; E_{ijpq}\right)^{-1}\; R^{old}_{mn}}{n_{mn}\; E_{mnpq}\; \left(\delta_{mp}\delta_{qn} + \Delta\mu\; \frac{\partial m_{pq}}{\partial \sigma_{ij}}\; E_{ijmn}\right)^{-1}\; {}^{n+1}m_{mn} + \frac{\partial \Phi}{\partial K_\alpha}\; H_{\alpha\beta}\; \frac{\partial \Phi^*}{\partial K_\beta}} \tag{106.189}$$

since in the limit, as deformations are getting small

$$\begin{aligned}
\mathcal{T}_{mnpq} & \rightarrow \delta_{pm}\delta_{nq} + \Delta\mu\; \frac{\partial m_{mn}}{\partial \sigma_{ij}}\; E_{ijpq} \\
\mathcal{F}_{pq} & \rightarrow \frac{1}{2}\; \frac{\partial \Phi}{\partial \sigma_{mn}}\; E_{mnpq} \\
Z_{pq} & \rightarrow 2\; m_{pq} \\
R_{pq} & \rightarrow 2\; \epsilon_{pq}
\end{aligned} \tag{106.190}$$

Upon noting that residual R_{pq} is defined in strain space, the increment inconsistency parameter $d(\Delta\mu)$ compares exactly with it's small strain counterpart (Jeremić and Sture, 1997).

The procedure described below summarizes the implementation of the return algorithm.

Trial State Given the right elastic deformations tensor ${}^{n}\bar{C}^{e}_{pq}$ and a set of hardening variables ${}^{n}K_{\alpha}$ at a specific quadrature point in a finite element, compute the relative deformation gradient ${}^{n+1}f_{ij}$ for a given displacement increment $\Delta^{n+1}u_i$, and the right deformation tensor

$$ {}^{n+1}f_{ij} = \delta_{ij} + u_{i,j} \tag{106.191}$$

$$ {}^{n+1}\bar{C}^{e,tr}_{ij} = \left({}^{n+1}f_{ir}\ {}^{n}F^{e}_{rk}\right)^{T} \left({}^{n+1}f_{kl}\ {}^{n}F^{e}_{lj}\right) = \left({}^{n}F^{e}_{rk}\right)^{T} \left({}^{n+1}f_{ir}\right)^{T} \left({}^{n+1}f_{kl}\ {}^{n}F^{e}_{lj}\right) \tag{106.192}$$

Compute the trial elastic second Piola–Kirchhoff stress and the trial elastic Mandel stress tensor

$$ {}^{n+1}\bar{S}^{e,tr}_{ij} = 2\frac{\partial W}{\partial^{n+1}\bar{C}^{e,tr}_{ij}} \tag{106.193}$$

$$ {}^{n+1}\bar{T}^{e,tr}_{ij} = {}^{n+1}\bar{C}^{e,tr}_{il}\ {}^{n+1}\bar{S}^{e,tr}_{lj} \tag{106.194}$$

Evaluate the yield function ${}^{n+1}\Phi^{tr}(\bar{T}^{e,tr}_{ij}, K_{\alpha})$. If ${}^{n+1}\Phi^{tr} \leq 0$ there is no plastic flow in current increment

$$\begin{aligned} {}^{n+1}\bar{C}^{e}_{ij} &= {}^{n+1}\bar{C}^{e,tr}_{ij} \\ {}^{n+1}K_{\alpha} &= {}^{n}K_{\alpha} \\ {}^{n+1}T_{ij} &= {}^{n}T^{e,tr}_{ij} \end{aligned}$$

and exit constitutive integration procedure.

Return Algorithm If yield criteria has been violated $({}^{n+1}\Phi^{tr} > 0)$ proceed to step 1.

step 1. k^{th} iteration. Known variables

$$ {}^{n+1}\bar{C}^{e(k)}_{ij} \quad ; \quad {}^{n+1}\kappa^{(k)}_{\alpha} \quad ; \quad {}^{n+1}K^{(k)}_{\alpha} \quad ; \quad {}^{n+1}T^{(k)}_{ij} \quad ; \quad {}^{n+1}\Delta\mu^{(k)}$$

evaluate the yield function and the residual

$$\begin{aligned} \Phi^{(k)} &= \Phi({}^{n+1}\bar{T}^{e(k)}_{ij}, {}^{n+1}K^{(k)}_{\alpha}) \\ R^{(k)}_{ij} &= {}^{n+1}\bar{C}^{e,(k)}_{ij} - \left({}^{n+1}\bar{C}^{e,tr}_{ij} - {}^{n+1}\Delta\mu^{(k)n+1}Z^{(k)}_{ij}\right) \end{aligned}$$

step 2. Check for convergence, $\Phi^{(k)} \leq NTOL$ and $\|R^{(k)}_{ij}\| \leq NTOL$. If convergence criteria is satisfied set

$$\begin{aligned} {}^{n+1}\bar{C}^{e}_{ij} &= {}^{n+1}\bar{C}^{e(k)}_{ij} \\ {}^{n+1}\kappa_{\alpha} &= {}^{n+1}\kappa^{(k)}_{\alpha} \\ {}^{n+1}K_{\alpha} &= {}^{n+1}K^{(k)}_{\alpha} \\ {}^{n+1}T_{ij} &= {}^{n+1}T^{(k)}_{ij} \\ {}^{n+1}\Delta\mu &= {}^{n+1}\Delta\mu^{(k)} \end{aligned}$$

Exit constitutive integration procedure.

step 3.[28] If convergence is not achieved, i.e. $\Phi^{(k)} > NTOL$ or $\|R^{(k)}_{ij}\| > NTOL$ then compute the elastic stiffness tensor $\mathcal{L}_{ijkl}$

$$\bar{\mathcal{L}}^{(k)}_{ijkl} = 4 \; \frac{\partial^2 W}{\partial \bar{C}^{e(k)}_{ij} \; \partial \bar{C}^{e(k)}_{kl}} \tag{106.195}$$

step 4. Compute the incremental inconsistency parameter $d(\Delta\mu^{(k+1)})$

$$d(\Delta\mu^{(k+1)}) = \frac{\Phi^{(k)} - \bar{\mathcal{F}}^{(k)}_{mn} \; R_{mn}{}^{(k)}}{\bar{\mathcal{F}}^{(k)}_{mn} \; Z_{mn}{}^{(k)} - \Delta\mu^{(k)} \; \bar{\mathcal{F}}_{mn}{}^{(k)} \; \dfrac{\partial Z_{mn}{}^{(k)}}{\partial K_\alpha} \; \bar{H}_\alpha{}^{(k)} + \dfrac{\partial \Phi^{(k)}}{\partial K_\alpha} \; \bar{H}_\alpha{}^{(k)}} \tag{106.196}$$

where

$$\bar{H}_\alpha{}^{(k)} = H_{\alpha\beta}{}^{(k)} \; \frac{\partial \Phi^{*,(k)}}{\partial K_\beta} \quad ; \quad \bar{\mathcal{F}}_{mn}{}^{(k)} = \mathcal{F}_{pq}{}^{(k)} \; \left(\mathcal{T}_{mnpq}{}^{(k)}\right)^{-1}$$

$$\mathcal{F}_{pq} = \frac{\partial \Phi(\bar{T}^{(k)}_{ij}, K^{(k)}_\alpha)}{\partial \bar{T}_{pn}} \; \left(\bar{C}^{e,(k)}_{sq}\right)^{-1} \; \bar{T}^{(k)}_{sn} + \frac{1}{2} \; \frac{\partial \Phi(\bar{T}^{(k)}_{ij}, K^{(k)}_\alpha)}{\partial \bar{T}_{mn}} \; \bar{C}^{e,(k)}_{mk} \; \bar{\mathcal{L}}^{e,(k)}_{knpq}$$

$$\mathcal{T}_{mnij} = \delta_{im}\delta_{nj} + \Delta\mu^{(k)} \; \frac{\partial Z^{(k)}_{mn}}{\partial \bar{T}^{(k)}_{ik}} \; \left(\bar{C}^{e,(k)}_{sj}\right)^{-1} \; \bar{T}^{(k)}_{sk} + \frac{1}{2} \; \Delta\mu^{(k)} \; \frac{\partial Z^{(k)}_{mn}}{\partial \bar{T}_{pq}} \; \bar{C}^{e,(k)}_{pk} \; \bar{\mathcal{L}}^{e,(k)}_{kqij}$$

step 5. Updated the inconsistency parameter $\Delta\mu^{(k+1)}$

$$\Delta\mu^{(k+1)} = \Delta\mu^{(k)} + d(\Delta\mu^{(k+1)}) \tag{106.197}$$

step 6. Updated the right deformation tensor, the hardening variable and the Mandel stress

$$d\bar{C}^{e,(k+1)}_{pq} = \left(\mathcal{T}^{(k)}_{mnpq}\right)^{-1} \left(-R^{(k)}_{mn} - d(\Delta\mu^{(k+1)}) \; {}^{n+1}Z^{(k)}_{mn} + \Delta\mu^{(k)} \; \frac{\partial Z^{(k)}_{mn}}{\partial K_\alpha} \; d(\Delta\mu^{(k+1)}) \; \bar{H}^{(k)}_\alpha \right) \tag{106.198}$$

$$d\kappa^{(k+1)}_\alpha = d(\Delta\mu^{(k+1)}) \; \frac{\partial \Phi^{*,(k)}}{\partial K_\beta} \tag{106.199}$$

$$dK^{(k+1)}_\alpha = -d(\Delta\mu^{(k+1)}) \; H^{(k)}_{\alpha\beta} \; \frac{\partial \Phi^{*,(k)}}{\partial K_\beta} \tag{106.200}$$

---

[28] From step 3. to step 9. all of the variables are in intermediate $n+1$ configuration. For the sake of brevity we are omitting superscript $n+1$.

$$d\bar{T}_{mn}^{(k+1)} = d\bar{C}_{mk}^{e,(k+1)} \left(\bar{C}_{sk}^{e,(k)}\right)^{-1} \bar{T}_{sn}^{(k)} + \frac{1}{2}\,\bar{C}_{mk}^{e,(k)}\,\bar{\mathcal{L}}_{knpq}^{e,(k)}\,d\bar{C}_{pq}^{e,(k+1)} \tag{106.201}$$

step 7. Update right deformation tensor $\bar{C}_{pq}^{e,(k+1)}$, hardening variable $K_\alpha^{(k+1)}$ and Mandel stress $\bar{T}_{mn}^{(k+1)}$

$$\begin{aligned}
\bar{C}_{pq}^{e,(k+1)} &= \bar{C}_{pq}^{e,(k)} + d(\bar{C}_{pq}^{e,(k+1)})\\
\kappa_\alpha^{(k+1)} &= \kappa_\alpha^{(k)} + d(\kappa_\alpha^{(k+1)})\\
K_\alpha^{(k+1)} &= K_\alpha^{(k)} + d(K_\alpha^{(k+1)})\\
\bar{T}_{mn}^{(k+1)} &= \bar{T}_{mn}^{(k)} + d(\bar{T}_{mn}^{(k+1)})
\end{aligned} \tag{106.202}$$

step 8. evaluate the yield function and the residual

$$\Phi^{(k+1)} = \Phi(\bar{T}_{ij}^{e(k+1)}, K_\alpha^{(k+1)}) \;\; ; \;\; R_{ij}^{(k+1)} = \bar{C}_{ij}^{e,(k+1)} - \left(\bar{C}_{ij}^{e,tr} - \Delta\mu^{(k+1)} Z_{ij}^{(k+1)}\right) \tag{106.203}$$

step 9. Set $k = k + 1$

$$\begin{aligned}
\Delta\mu^{(k)} &= \Delta\mu^{(k+1)}\\
\bar{C}_{pq}^{e,(k)} &= \bar{C}_{pq}^{e,(k+1)}\\
\kappa_\alpha^{(k)} &= \kappa_\alpha^{(k+1)}\\
K_\alpha^{(k)} &= K_\alpha^{(k+1)}\\
\bar{T}_{mn}^{(k)} &= \bar{T}_{mn}^{(k+1)}
\end{aligned} \tag{106.204}$$

and return to step 2.

106.4.5 Algorithmic Tangent Stiffness Tensor

Starting from the elastic predictor–plastic corrector equation

$${}^{n+1}\bar{C}_{ij}^{e} = {}^{n+1}\bar{C}_{ij}^{e,tr} - \Delta\mu\;{}^{n+1}Z_{ij} \tag{106.205}$$

and taking the first order Taylor series expansion we obtain

$$\begin{aligned}
d\bar{C}_{ij}^{e} &= d\bar{C}_{ij}^{e,tr} - d(\Delta\mu)\,Z_{ij} - \Delta\mu\,\frac{\partial Z_{ij}}{\partial \bar{T}_{mn}}\,d\bar{T}_{mn} - \Delta\mu\,\frac{\partial Z_{ij}}{\partial K_\alpha}\,dK_\alpha\\
&= d\bar{C}_{ij}^{e,tr} - d(\Delta\mu)\,Z_{ij}\\
&\quad -\Delta\mu\,\frac{\partial Z_{ij}}{\partial \bar{T}_{mn}}\left(d\bar{C}_{mk}^{e}\,\left(\bar{C}_{sk}^{e}\right)^{-1}\,\bar{T}_{sn} + \frac{1}{2}\,\bar{C}_{mk}^{e}\,\bar{\mathcal{L}}_{knpq}^{e}\,d\bar{C}_{pq}^{e}\right) \quad \text{from (106.175)}\\
&\quad -\Delta\mu\,\frac{\partial Z_{ij}}{\partial K_\alpha}\,dK_\alpha
\end{aligned} \tag{106.206}$$

Previous equation can be written as

$$\begin{aligned}
&d\bar{C}_{ij}^{e} + \Delta\mu\,\frac{\partial Z_{ij}}{\partial \bar{T}_{mn}}\,\left(\bar{C}_{sk}^{e}\right)^{-1}\,\bar{T}_{sn}\,d\bar{C}_{mk}^{e} + \Delta\mu\,d(\Delta\mu)\,\frac{\partial Z_{ij}}{\partial \bar{T}_{mn}}\,\frac{1}{2}\,\bar{C}_{mk}^{e}\,\bar{\mathcal{L}}_{knpq}^{e}\,d\bar{C}_{pq}^{e}\\
&\qquad = d\bar{C}_{ij}^{e,tr} - d(\Delta\mu)\,Z_{ij} + \Delta\mu\,d(\Delta\mu)\,\frac{\partial Z_{ij}}{\partial K_\alpha}\,H_{\alpha\beta}\,\frac{\partial \Phi^*}{\partial K_\beta}
\end{aligned} \tag{106.207}$$

or as

$$d\bar{C}^{e}_{ij}\ \left(\mathcal{T}_{mnij}\right) = d\bar{C}^{e,tr}_{ij} - d(\Delta\mu)\, Z_{ij} + \Delta\mu\ d(\Delta\mu)\ \frac{\partial Z_{ij}}{\partial K_{\alpha}}\ H_{\alpha\beta}\ \frac{\partial \Phi^{*}}{\partial K_{\beta}} \tag{106.208}$$

where

$$\mathcal{T}_{mnij} = \delta_{im}\delta_{nj} + \Delta\mu^{(k)}\ \frac{\partial Z^{(k)}_{mn}}{\partial \bar{T}^{(k)}_{ik}}\ \left(\bar{C}^{e,(k)}_{sj}\right)^{-1}\ \bar{T}^{(k)}_{sk} + \frac{1}{2}\ \Delta\mu^{(k)}\ \frac{\partial Z^{(k)}_{mn}}{\partial \bar{T}_{pq}}\ \bar{C}^{e,(k)}_{pk}\ \bar{\mathcal{L}}^{e,(k)}_{kqij}$$

The solution for the increment in right elastic deformation tensor is then

$$d\bar{C}^{e}_{ij}\ = \left(\mathcal{T}_{mnij}\right)^{-1}\ \left(d\bar{C}^{e,tr}_{ij} - d(\Delta\mu)\, Z_{ij} + \Delta\mu\ d(\Delta\mu)\ \frac{\partial Z_{ij}}{\partial K_{\alpha}}\ H_{\alpha\beta}\ \frac{\partial \Phi^{*}}{\partial K_{\beta}}\right) \tag{106.209}$$

We next use the first order Taylor series expansion of yield function $d\Phi(\bar{T}_{ij}, K_{\alpha}) = 0$

$$\begin{aligned}
\frac{\partial \Phi}{\partial \bar{T}_{mn}}\ d\bar{T}_{mn} + \frac{\partial \Phi}{\partial K_{\alpha}}\ dK_{\alpha} &= \\
\frac{\partial \Phi}{\partial \bar{T}_{mn}}\ \left(d\bar{C}^{e}_{mk}\ \left(\bar{C}^{e}_{sk}\right)^{-1}\ \bar{T}_{sn} + \frac{1}{2}\ \bar{C}^{e}_{mk}\ \bar{\mathcal{L}}^{e}_{knpq}\ d\bar{C}^{e}_{pq}\right) + \frac{\partial \Phi}{\partial K_{\alpha}}\ dK_{\alpha} &= \\
\frac{\partial \Phi}{\partial \bar{T}_{pn}}\ \left(\bar{C}^{e}_{sq}\right)^{-1}\ \bar{T}_{sn}\ d\bar{C}^{e}_{pq} + \frac{1}{2}\ \frac{\partial \Phi}{\partial \bar{T}_{mn}}\ \bar{C}^{e}_{mk}\ \bar{\mathcal{L}}^{e}_{knpq}\ \ d\bar{C}^{e}_{pq} + \frac{\partial \Phi}{\partial K_{\alpha}}\ dK_{\alpha} &= \\
\left(\frac{\partial \Phi}{\partial \bar{T}_{pn}}\ \left(\bar{C}^{e}_{sq}\right)^{-1}\ \bar{T}_{sn} + \frac{1}{2}\ \frac{\partial \Phi}{\partial \bar{T}_{mn}}\ \bar{C}^{e}_{mk}\ \bar{\mathcal{L}}^{e}_{knpq}\right)\ d\bar{C}^{e}_{pq} - \frac{\partial \Phi}{\partial K_{\alpha}}\ d(\Delta\mu)\, H_{\alpha\beta}\ \frac{\partial \Phi^{*}}{\partial K_{\beta}} &= \\
\mathcal{F}_{pq}\ d\bar{C}^{e}_{pq} - \frac{\partial \Phi}{\partial K_{\alpha}}\ d(\Delta\mu)\, H_{\alpha\beta}\ \frac{\partial \Phi^{*}}{\partial K_{\beta}} &= \ \ 0
\end{aligned} \tag{106.210}$$

where

$$\mathcal{F}_{pq} = \frac{\partial \Phi}{\partial \bar{T}_{pn}}\ \left(\bar{C}^{e}_{sq}\right)^{-1}\ \bar{T}_{sn} + \frac{1}{2}\ \frac{\partial \Phi}{\partial \bar{T}_{mn}}\ \bar{C}^{e}_{mk}\ \bar{\mathcal{L}}^{e}_{knpq} \tag{106.211}$$

By using solution for $d\bar{C}^{e}_{ij}$ from 106.209 we can write

$$\begin{aligned}
\mathcal{F}_{pq}\ \left(\mathcal{T}_{mnpq}\right)^{-1}\ \left(d\bar{C}^{e,tr}_{mn} - d(\Delta\mu)\, Z_{mn} + \Delta\mu\ d(\Delta\mu)\ \frac{\partial Z_{mn}}{\partial K_{\alpha}}\ H_{\alpha\beta}\ \frac{\partial \Phi^{*}}{\partial K_{\beta}}\right)& \\
-\frac{\partial \Phi}{\partial K_{\alpha}}\ d(\Delta\mu)\, H_{\alpha\beta}\ \frac{\partial \Phi^{*}}{\partial K_{\beta}} &= \ \ 0
\end{aligned} \tag{106.212}$$

We are now in the position to solve for the incremental inconsistency parameter $d(\Delta\mu)$

$$d(\Delta\mu) = \frac{\mathcal{F}_{pq}\ \left(\mathcal{T}_{mnpq}\right)^{-1}\ d\bar{C}^{e,tr}_{mn}}{\Gamma} \tag{106.213}$$

where we have used Γ to shorten writing

$$\Gamma = \mathcal{F}_{pq}\left(\mathcal{T}_{mnpq}\right)^{-1}\, {}^{n+1}Z_{mn} - \Delta\mu\mathcal{F}_{pq}\left(\mathcal{T}_{mnpq}\right)^{-1}\ \frac{\partial^{n+1}Z_{mn}}{\partial K_{\alpha}} H_{\alpha\beta} \frac{\partial \Phi^{*}}{\partial K_{\beta}} + \frac{\partial \Phi}{\partial K_{\alpha}} H_{\alpha\beta} \frac{\partial \Phi^{*}}{\partial K_{\beta}} \tag{106.214}$$

Since

$$d\bar{S}_{kn} = \frac{1}{2}\,\bar{\mathcal{L}}^{e}_{knpq}\,d\bar{C}^{e}_{pq} \tag{106.215}$$

and by using 106.209 we can write

$$\begin{aligned}
&d\bar{C}^{e}_{pq} = \\
&\left(\mathcal{T}_{mnpq}\right)^{-1}\left(\delta_{mv}\,\delta_{nt} - \frac{\mathcal{F}_{op}\,\left(\mathcal{T}_{rsop}\right)^{-1}\,\delta_{rv}\,\delta_{st}}{\Gamma}\,Z_{mn} + \right. \\
&\left.\Delta\mu\,\frac{\mathcal{F}_{op}\,\left(\mathcal{T}_{rsop}\right)^{-1}\,\delta_{rv}\,\delta_{st}}{\Gamma}\,\frac{\partial Z_{ij}}{\partial K_{\alpha}}\,H_{\alpha\beta}\,\frac{\partial \Phi^{*}}{\partial K_{\beta}}\right)\,d\bar{C}^{e,tr}_{vt}
\end{aligned} \tag{106.216}$$

Then

$$d\bar{C}^{e}_{pq} = \bar{\mathcal{P}}_{pqvt}\,d\bar{C}^{e,tr}_{vt} \tag{106.217}$$

where

$$\begin{aligned}
\bar{\mathcal{P}}_{pqvt} \;&=\; \left(\mathcal{T}_{mnpq}\right)^{-1}\,\delta_{mv}\,\delta_{nt} - \frac{\mathcal{F}_{op}\,\left(\mathcal{T}_{rsop}\right)^{-1}\,\delta_{rv}\,\delta_{st}}{\Gamma}\,Z_{mn} \\
&+\;\Delta\mu\,\frac{\mathcal{F}_{op}\,\left(\mathcal{T}_{rsop}\right)^{-1}\,\delta_{rv}\,\delta_{st}}{\Gamma}\,\frac{\partial Z_{ij}}{\partial K_{\alpha}}\,H_{\alpha\beta}\,\frac{\partial \Phi^{*}}{\partial K_{\beta}} \\
&=\; (\mathcal{T}_{mnpq})^{-1}\left(\delta_{mv}\delta_{nt} - \frac{\mathcal{F}_{ab}(\mathcal{T}_{vtab})^{-1}}{\Gamma}\left(Z_{mn} - \Delta\mu\,\frac{\partial^{n+1}Z_{mn}}{\partial K_{\alpha}}H_{\alpha\beta}\frac{\partial \Phi^{*}}{\partial K_{\beta}}\right)\right)
\end{aligned} \tag{106.218}$$

Algorithmic tangent stiffness tensor $\bar{\mathcal{L}}_{ijkl}$ (in intermediate configuration $\bar{\Omega}$) is then defined as

$$\bar{\mathcal{L}}^{ATS}_{knvt} = \bar{\mathcal{L}}^{e}_{knpq}\,\bar{\mathcal{P}}_{pqvt} \tag{106.219}$$

Pull–back to the reference configuration Ω_0 yields the algorithmic tangent stiffness tensor $\mathcal{L}_{ijkl}$ in reference configuration Ω_0

$${}^{n+1}\mathcal{L}^{ATS}_{ijkl} = {}^{n+1}F^{p}_{im}\;{}^{n+1}F^{p}_{jn}\;{}^{n+1}F^{p}_{kr}\;{}^{n+1}F^{p}_{ls}\;{}^{n+1}\bar{\mathcal{L}}^{ATS}_{mnrs} \tag{106.220}$$

Remark 106.4.8 In the limit, for small deformations, isotropic response, the Algorithmic Tangent Stiffness tensor $\mathcal{L}^{ATS}_{ijkl}$ becomes

$$\begin{aligned}
\lim \bar{\mathcal{L}}^{ATS}_{vtpq} = E^{ATS}_{vtpq} = \;&\; E_{knpq}\left(\Upsilon^{-1}_{mnpq}\left(\delta_{mv}\delta_{nt} - \frac{n_{cd}E_{cdab}\Upsilon^{-1}_{vtab}\mathcal{H}_{mn}}{\Gamma}\right)\right) \\
=\;&\; E_{knpq}\left(\Upsilon^{-1}_{vtpq} - \frac{\Upsilon^{-1}_{mrpq}n_{cd}E_{cdab}\Upsilon^{-1}_{vtab}\mathcal{H}_{mr}}{\Gamma}\right) \\
=\;&\; E_{knpq}\Upsilon^{-1}_{vtpq} - \frac{E_{knpq}\Upsilon^{-1}_{mrpq}n_{cd}E_{cdab}\Upsilon^{-1}_{vtab}\mathcal{H}_{mr}}{\Gamma} \\
=\;&\; \mathcal{R}_{knvt} - \frac{n_{cd}\mathcal{R}_{cdvt}\mathcal{R}_{knmr}\mathcal{H}_{mr}}{\Gamma}
\end{aligned} \tag{106.221}$$

since

$$\begin{aligned} \lim \bar{\mathcal{T}}_{mnpq} = \Upsilon_{mnpq} = \quad & \delta_{pm}\delta_{nq} + \Delta\mu \frac{\partial Z_{mn}}{\partial \bar{T}_{pk}} \left(\bar{C}^e_{sq}\right)^{-1} \bar{T}_{sk} + \frac{1}{2}\Delta\mu \frac{\partial Z_{mn}}{\partial \bar{T}_{rs}} \bar{C}^e_{rk} \bar{\mathcal{L}}^e_{kspq} \\ = \quad & \delta_{pm}\delta_{nq} + \Delta\mu \frac{\partial m_{mn}}{\partial \sigma_{rs}} E^e_{kspq} \end{aligned} \tag{106.222}$$

$$\begin{aligned} \lim \mathcal{F}_{ab} = \quad & \lim \left(\frac{\partial \Phi}{\partial \bar{T}_{ad}} \left(\bar{C}^e_{sb}\right)^{-1} \bar{T}_{sd} + \frac{1}{2} \frac{\partial \Phi}{\partial \bar{T}_{cd}} \bar{C}^e_{ck} \bar{\mathcal{L}}^e_{kdab} \right) \\ = \quad & \frac{1}{2} n_{cd} E^e_{cdab} \end{aligned} \tag{106.223}$$

$$\mathcal{H}_{mn} = m_{mn} - \Delta\mu \frac{\partial m_{mn}}{\partial K_\alpha} H_{\alpha\beta} \frac{\partial \Phi^*}{\partial K_\beta} \tag{106.224}$$

$$\begin{aligned} \lim \Gamma = \quad & \lim \left(\mathcal{F}_{pq} \left(\mathcal{T}_{mnpq}\right)^{-1} {}^{n+1}Z_{mn} - \Delta\mu \mathcal{F}_{pq} \left(\mathcal{T}_{mnpq}\right)^{-1} \frac{\partial^{n+1} Z_{mn}}{\partial K_\alpha} H_{\alpha\beta} \frac{\partial \Phi^*}{\partial K_\beta} + \frac{\partial \Phi}{\partial K_\alpha} H_{\alpha\beta} \frac{\partial \Phi^*}{\partial K_\beta} \right) \\ = \quad & n_{ab} E_{abpq} \Upsilon^{-1}_{mnpq} m_{mn} - \Delta\mu n_{ab} E_{abpq} \Upsilon^{-1}_{mnpq} \frac{\partial m_{mn}}{\partial K_\alpha} H_{\alpha\beta} \frac{\partial \Phi^*}{\partial K_\beta} + \frac{\partial \Phi}{\partial K_\alpha} H_{\alpha\beta} \frac{\partial \Phi^*}{\partial K_\beta} \\ = \quad & n_{ab} E_{abpq} \Upsilon^{-1}_{mnpq} \left(m_{mn} - \Delta\mu \frac{\partial m_{mn}}{\partial K_\alpha} H_{\alpha\beta} \frac{\partial \Phi^*}{\partial K_\beta} \right) + \frac{\partial \Phi}{\partial K_\alpha} H_{\alpha\beta} \frac{\partial \Phi^*}{\partial K_\beta} \\ = \quad & n_{ab} \mathcal{R}_{abmn} \mathcal{H}_{mn} + \frac{\partial \Phi}{\partial K_\alpha} H_{\alpha\beta} \frac{\partial \Phi^*}{\partial K_\beta} \end{aligned} \tag{106.225}$$

It is noted that the Algorithmic Tangent Stiffness tensor given by 106.221 compares exactly with it's small strain counterpart (Jeremić and Sture, 1997).

Chapter 107

Solution of Static Equilibrium Equations

(1994-2016-)

(In collaboration with Dr. Yuan Feng)

107.1 Chapter Summary and Highlights

107.2 The Residual Force Equations

This chapter is based on Felippa (1993).

In previous Chapters we have derived the basic equations for (material and/or geometric) nonlinear analysis of solids. Discretization of such problems by finite element methods results in a set of nonlinear algebraic equations called *residual force equations*:

$$\mathbf{r}(\mathbf{u}, \lambda) = \mathbf{f}_{int}(\mathbf{u}) - \lambda \mathbf{f}_{ext} = 0 \tag{107.1}$$

where $\mathbf{f}_{int}(\mathbf{u})$ are the internal forces which are functions of the displacements, $\mathbf{u}$, the vector $\mathbf{f}_{ext}$ is a fixed external loading vector and the scalar λ is a load–level parameter that multiplies $\mathbf{f}_{ext}$. Equation (107.1) describes the case of proportional loading in which the loading pattern is kept fixed.

All solution procedures of practical importance are strongly rooted in the idea of "advancing the solution" by *continuation*. Except in very simple problems, the continuation process is *multilevel* and involves hierarchical breakdown into stages, incremental steps and iterative steps. Processing a complex nonlinear problem generally involves performing a series of analysis stages. Multiple control parameters are not varied independently in each stage and may therefore be characterized by a single stage control parameter λ. Stages are only weakly coupled in the sense that end solution of one may provide the starting point for another.

107.3 Constraining the Residual Force Equations

Various forms of path following methods[1] have stemmed from the original work of Riks (1972), Riks (1979) and Wempner (1971). They aimed at finding the intersection of equation (107.1) with $s = constant$ where s is the arc-length , defined as[2]:

$$s = \int ds \tag{107.2}$$

where:

$$ds = \sqrt{\frac{\psi_u^2}{u_{ref}^2} d\mathbf{u}^T \mathbf{S} d\mathbf{u} + d\lambda^2 \psi_f^2} \tag{107.3}$$

---

[1] also called arc-length methods with various methods of approximating the exact length of an arc.

[2] A bit different form in that it is scaled with scaling matrix $\mathbf{S}$, introduced by Felippa (1984).

Differential form (107.3) can be replaced with an incremental form:

$$a = (\Delta s)^2 - (\Delta l)^2 = \left(\frac{\psi_u^2}{u_{ref}^2} \Delta \mathbf{u}^T \mathbf{S} \Delta \mathbf{u} + \Delta\lambda^2 \psi_f^2 \right) - (\Delta l)^2 \tag{107.4}$$

where Δl is the radius of the desired intersection[3] and represents an approximation to the incremental arc length. Scaling matrix $\mathbf{S}$ is usually diagonal non-negative matrix that scales the state vector $\Delta\mathbf{u}$ and u_{ref} is a reference value with the dimension of $\sqrt{\Delta\mathbf{u}^T\mathbf{S}\Delta\mathbf{u}}$. It is important to note that the vector $\Delta\mathbf{u}$ and scalar $\Delta\lambda$ are incremental and not iterative values, and are starting from the last converged equilibrium state.

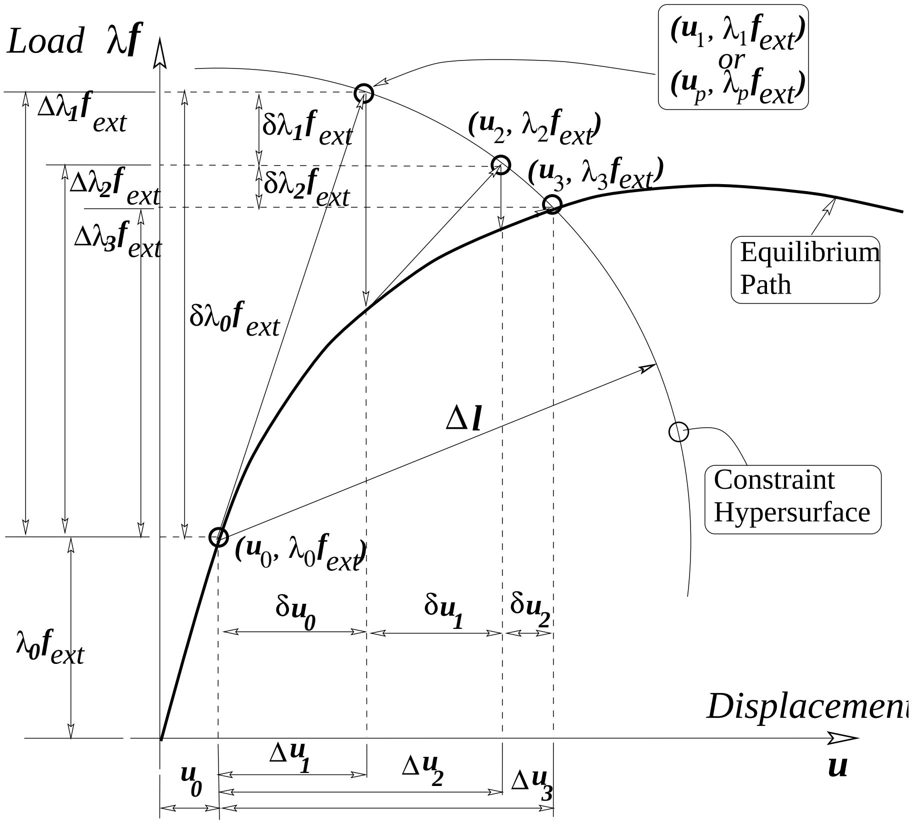


Figure 107.1: Spherical arc-length method and notation for one DOF system.

The main essence of the arc-length methods is that the load parameter λ becomes a variable. With load parameter λ variable we are dealing with $n+1$ unknowns[4]. In order to solve this problem we have n equilibrium equations (107.1) and the one constraint equation (107.4). We can solve the augmented

---

[3] See Figure (107.1).

[4] n unknown displacement variables and on extra unknown in the form of load parameter.

system of $n + 1$ equations by applying the Newton-Raphson[5] method to equations (107.1) and (107.4)

$$
\begin{aligned}
\mathbf{r}^{new}(\mathbf{u}, \lambda) &= \mathbf{r}^{old}(\mathbf{u}, \lambda) + \frac{\partial \mathbf{r}(\mathbf{u}, \lambda)}{\partial \mathbf{u}} \, \delta \mathbf{u} + \frac{\partial \mathbf{r}(\mathbf{u}, \lambda)}{\partial \lambda} \, \delta \lambda = \\
&= \mathbf{r}^{old}(\mathbf{u}, \lambda) + \mathbf{K}_t \, \delta \mathbf{u} - \mathbf{f}_{ext} \, \delta \lambda = \\
&= 0 \qquad (107.5)
\end{aligned}
$$

$$
a^{new} = a^{old} + 2 \frac{\psi_u^2}{u_{ref}^2} \Delta \mathbf{u}^T \mathbf{S} \delta \mathbf{u} + 2 \Delta \lambda \delta \lambda \, \psi_f^2 = 0 \qquad (107.6)
$$

where $\mathbf{K}_t = \frac{\partial \mathbf{r}(\mathbf{u},\lambda)}{\partial \mathbf{u}}$ is the tangent stiffness matrix. The aim is to have $\mathbf{r}^{new}(\mathbf{u}, \lambda) = 0$ and $a^{new} = 0$ so the previous system can be written as:

$$
\begin{bmatrix} \mathbf{K}_t & -\mathbf{f}_{ext} \\ 2 \frac{\psi_u^2}{u_{ref}^2} \Delta \mathbf{u}^T \mathbf{S} & 2 \Delta \lambda \, \psi_f^2 \end{bmatrix} \begin{bmatrix} \delta \mathbf{u} \\ \delta \lambda \end{bmatrix} = - \begin{bmatrix} \mathbf{r}^{old} \\ a^{old} \end{bmatrix} \qquad (107.7)
$$

One can solve previous system of two equations for $\delta \mathbf{u}$ and $\delta \lambda$:

$$
\begin{bmatrix} \delta \mathbf{u} \\ \delta \lambda \end{bmatrix} = - \begin{bmatrix} \mathbf{K}_t & -\mathbf{f}_{ext} \\ 2 \frac{\psi_u^2}{u_{ref}^2} \Delta \mathbf{u}^T \mathbf{S} & 2 \Delta \lambda \, \psi_f^2 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{r}^{old} \\ a^{old} \end{bmatrix} \qquad (107.8)
$$

or by defining the augmented stiffness matrix[6] $\mathcal{K}$ as:

$$
\mathcal{K} = \begin{bmatrix} \mathbf{K}_t & -\mathbf{f}_{ext} \\ 2 \frac{\psi_u^2}{u_{ref}^2} \Delta \mathbf{u}^T \mathbf{S} & 2 \Delta \lambda \, \psi_f^2 \end{bmatrix} \qquad (107.9)
$$

the equation (107.8) can be written as:

$$
\begin{bmatrix} \delta \mathbf{u} \\ \delta \lambda \end{bmatrix} = - \mathcal{K}^{-1} \begin{bmatrix} \mathbf{r}^{old} \\ a^{old} \end{bmatrix} \qquad (107.10)
$$

It should be mentioned that the augmented stiffness matrix remains non-singular even if $\mathbf{K}_t$ is singular.

---

[5] By using a truncated Taylor series expansion.

[6] Or augmented Jacobian.

107.4 Load Control

107.5 Displacement Control

107.6 Generalized, Hyper–Spherical Arc-Length Control

In section (107.3) we have introduced an constraining equation that is intended to reduce the so called drift error in the incremental nonlinear finite element procedure. The constraining equation was given in a rather general form. Some further comments and observations are in order. By assigning various numbers to parameters ψ_u, ψ_f, $\mathbf{S}$ and u_{ref} one can obtain different constraining schemes from (107.4).

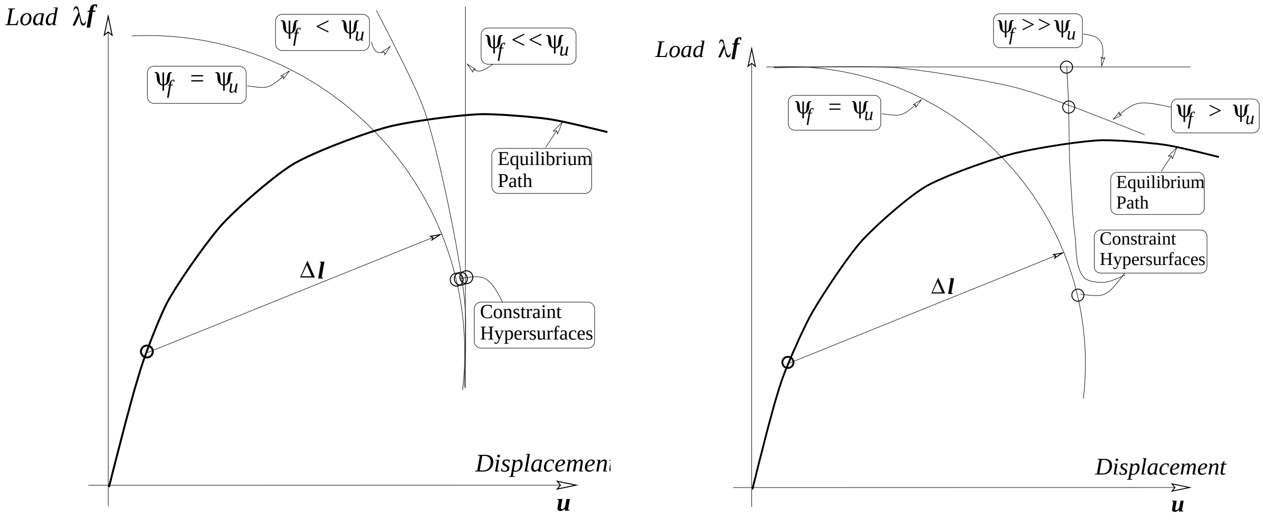


Figure 107.2: Influence of ψ_u and ψ_f on the constraint surface shape.

Coefficients ψ_u and ψ_f may not be simultaneously zero. Useful choices for $\mathbf{S}$ are $\mathbf{I}$, $\mathbf{K}_t$ and $diag\left(\mathbf{K}_t\right)$. If $\mathbf{S} = \mathbf{I}$ and $u_{ref} = 1$ the method is called the arclength method[7]. If we choose $\mathbf{S} = diag\left(\mathbf{K}_t\right)$ nice scaling is obtained[8] but otherwise no physical meaning can be attributed to this scaling type. With $\mathbf{S} = \mathbf{K}_t$ and $\psi_f \equiv 0$ one ends up with something very similar to the external work constraint of Bathe and Dvorkin (1983). A rather general equation (107.4) can be further specialized to load (λ) control with $\psi_u \equiv 0; \psi_f \equiv 1$ and state control[9] with $\psi_u \equiv 1; \psi_f \equiv 0$ and $\mathbf{S} = \mathbf{I}$. In the finite element literature, the term displacement control has been traditionally associated with the case in which only one of the components of the displacement vector $\mathbf{u}$[10] is specified. This may be regarded either as a variant of state

---

[7] It actually reduces to the original work of fRiks (1972), Riks (1979) and Wempner (1971).

[8] For example if FEM model includes both translational and rotational DOFs.

[9] That is the cylindrical constraint, or general displacement control.

[10] Say $\mathbf{u}_i$.

control, in which a norm that singles out the ith component is used, or as a variant of the λ control if the control parameter is taken as $\lambda \mathbf{u}_i$. It is, of course, possible to make the previous parameters variable, functions of different unknowns. For example if one defines $u_{ref} = \Delta\mathbf{u}^T\mathbf{S}\Delta\mathbf{u}$ then close to the limit point $\Delta\mathbf{u} \to 0 \Rightarrow \frac{\psi_u^2}{u_{ref}^2} \gg \psi_f^2$ that makes our constraint from equation (107.4) behave like state control. One important aspect of scaling constraint equations by using $\mathbf{S} = diag\left(\mathbf{K}_t\right)$ or $\mathbf{S} = \mathbf{K}_t$ is the possibility of non–positive definite stiffness matrix $\mathbf{K}_t$. It usually happens that after the limit point is passed, at least one of the eigenvalues of $\mathbf{K}_t$ is non–positive, thus rendering the constraint hypersurface non–convex.

In order to get better control of the solution to the system of equations (107.10) one may directly introduce the constraint from equation (107.6) by following the approach proposed by Batoz and Dhatt (1979), as described by Crisfield (1991) and Felippa (1993).

The iterative displacement $\delta\mathbf{u}$ is split into two parts, and with the Newton change at the new unknown load level:

$$\lambda^{new} = \lambda^{old} + \delta\lambda \tag{107.11}$$

becomes:

$$\begin{aligned}
\delta\mathbf{u} &= -\mathbf{K}_t^{-1}\mathbf{r}\left(\mathbf{u}^{old}, \lambda\right) = -\mathbf{K}_t^{-1}\left(\mathbf{f}_{int}(\mathbf{u}^{old}) - \lambda^{new}\mathbf{f}_{ext}\right) \\
&= -\mathbf{K}_t^{-1}\left(\mathbf{f}_{int}(\mathbf{u}^{old}) - \left(\lambda^{old} + \delta\lambda\right)\mathbf{f}_{ext}\right) = -\mathbf{K}_t^{-1}\left(\left(\mathbf{f}_{int}(\mathbf{u}^{old}) - \lambda^{old}\mathbf{f}_{ext}\right) - \delta\lambda\mathbf{f}_{ext}\right) \\
&= -\mathbf{K}_t^{-1}\left(\mathbf{r}\left(\mathbf{u}^{old}, \lambda^{old}\right) - \delta\lambda\mathbf{f}_{ext}\right) = -\mathbf{K}_t^{-1}\mathbf{r}^{old} + \delta\lambda\mathbf{K}_t^{-1}\mathbf{f}_{ext} = \delta\bar{\mathbf{u}} + \delta\lambda\delta\mathbf{u}_t
\end{aligned} \tag{107.12}$$

where $\delta\mathbf{u}_t = \mathbf{K}_t^{-1}\mathbf{f}_{ext}$ is the displacement vector corresponding to the fixed load vector $\mathbf{f}_{ext}$, and $\delta\bar{\mathbf{u}}$ is an iterative change that would stem from the standard load-controlled Newton-Raphson, at a fixed load level λ^{old}. With the solution[11] for the $\delta\mathbf{u}$ from (107.12), the new incremental displacements are:

$$\Delta\mathbf{u}^{new} = \Delta\mathbf{u}^{old} + \delta\mathbf{u} = \Delta\mathbf{u}^{old} + \delta\bar{\mathbf{u}} + \delta\lambda\delta\mathbf{u}_t \tag{107.13}$$

where $\delta\lambda$ is the only unknown. The constraint from equation (107.4) can be used here, and by rewriting it as:

$$\left(\frac{\psi_u^2}{u_{ref}^2}\left(\Delta\mathbf{u}^{new}\right)^T\mathbf{S}\left(\Delta\mathbf{u}^{new}\right) + \left(\Delta\lambda^{new}\right)^2\psi_f^2\right) = (\Delta l)^2 \tag{107.14}$$

---

[11] But having in mind that $\delta\lambda$ is still unknown!

then by substituting $\Delta \mathbf{u}^{new}$ from equation (107.13) into equation (107.14) and by recalling that $\lambda^{new} = \lambda^{old} + \delta\lambda$ one ends up with the following quadratic scalar equation:

$$\left(\frac{\psi_u^2}{u_{ref}^2} \left(\Delta \mathbf{u}^{old} + \delta \bar{\mathbf{u}} + \delta\lambda \delta \mathbf{u}_t \right)^T \mathbf{S} \left(\Delta \mathbf{u}^{old} + \delta \bar{\mathbf{u}} + \delta\lambda \delta \mathbf{u}_t \right) + \left(\Delta\lambda^{old} + \delta\lambda \right)^2 \psi_f^2 \right) = (\Delta l)^2 \tag{107.15}$$

or, by collecting terms:

$$\begin{array}{rcl} \left(\frac{\psi_u^2}{u_{ref}^2} \delta \mathbf{u}_t^T \mathbf{S} \delta \mathbf{u}_t + \psi_f^2 \right) & \delta\lambda^2 & + \\ +2 \left(\frac{\psi_u^2}{u_{ref}^2} \delta \mathbf{u}_t^T \mathbf{S} \left(\Delta \mathbf{u}^{old} + \delta \bar{\mathbf{u}} \right) + \Delta\lambda^{old} \psi_f^2 \right) & \delta\lambda & + \\ + \left(\frac{\psi_u^2}{u_{ref}^2} \left(\Delta \mathbf{u}^{old} + \delta \bar{\mathbf{u}} \right)^T \mathbf{S} \left(\Delta \mathbf{u}^{old} + \delta \bar{\mathbf{u}} \right) - \Delta l^2 + \left(\Delta\lambda^{old} \right)^2 \psi_f^2 \right) & & = 0 \end{array} \tag{107.16}$$

or:

$$a_1 \delta\lambda^2 + 2 a_2 \delta\lambda + a_3 = 0 \tag{107.17}$$

where:

$$\begin{array}{rcl} a_1 & = & \frac{\psi_u^2}{u_{ref}^2} \delta \mathbf{u}_t^T \mathbf{S} \delta \mathbf{u}_t + \psi_f^2 \\ 2a_2 & = & 2 \left(\frac{\psi_u^2}{u_{ref}^2} \delta \mathbf{u}_t^T \mathbf{S} \left(\Delta \mathbf{u}^{old} + \delta \bar{\mathbf{u}} \right) + \Delta\lambda^{old} \psi_f^2 \right) \\ a_3 & = & \frac{\psi_u^2}{u_{ref}^2} \left(\Delta \mathbf{u}^{old} + \delta \bar{\mathbf{u}} \right)^T \mathbf{S} \left(\Delta \mathbf{u}^{old} + \delta \bar{\mathbf{u}} \right) - \Delta l^2 + \left(\Delta\lambda^{old} \right)^2 \psi_f^2 \end{array}$$

The quadratic scalar equation (107.17) can be solved for $\delta\lambda$:

$$\delta\lambda = \delta\lambda_1 = \frac{-a_2 + \sqrt{a_2^2 - a_1 a_3}}{a_1} \quad ; \quad \delta\lambda = \delta\lambda_2 = \frac{-a_2 - \sqrt{a_2^2 - a_1 a_3}}{a_1} \tag{107.18}$$

or, if $a_1 = 0$, then:

$$\delta\lambda = -\frac{a_3}{2a_2} \tag{107.19}$$

and then the complete change is defined from equation (107.13):

$$\Delta \mathbf{u}^{new} = \Delta \mathbf{u}^{old} + \delta \bar{\mathbf{u}} + \delta\lambda \delta \mathbf{u}_t \tag{107.20}$$

An ambiguity is introduced in the solution for $\delta\lambda$ from (107.18). The tangent at the regular point on the equilibrium path has two possible directions, which generally intersect the constraint hypersurface at two points. However, some exceptions from that rule are possible, so the solutions from (107.18) can be categorized as:

- Real roots of opposite signs. This occurs when the iteration process converges normally and there is no limit or turning point enclosed by the constraint hypersurface. The root is chosen by applying one of the schemes proposed below.
- Real roots of equal sign opposite to that of $\Delta\lambda^{old}$. This usually happens when going over a "flat" limit point.
- Real roots of equal sign same as that of $\Delta\lambda^{old}$. This is an unusual case. It may signal a turning point or be triggered by erratic iteration behavior.
- Complex roots. This is an unusual case too. It may signal a sharp turning point, a bifurcation point, erratic or divergent iterates.

For the first two cases, the correct $\Delta\lambda$ can be chosen by applying one of the following schemes.

107.6.1 Traversing Equilibrium Path in Positive Sense

107.6.1.1 Positive External Work

The simplest rule requires that the external work expenditure over the predictor step be positive:

$$\Delta W = \mathbf{f}_{ext}^T \Delta \mathbf{u} = \mathbf{f}_{ext}^T \mathbf{K}_t^{-1} \mathbf{f}_{ext} \delta\lambda > 0 \tag{107.21}$$

The simple conclusion is that $\delta\lambda$ should have the sign of $\mathbf{f}_{ext}^T \mathbf{K}_t^{-1} \mathbf{f}_{ext}$. This condition is particularly effective at limit points. However, it fails when $\mathbf{f}_{ext}$ and $\mathbf{K}_t^{-1}\mathbf{f}_{ext}$ are orthogonal:

$$\mathbf{f}_{ext}^T \mathbf{K}_t^{-1} \mathbf{f}_{ext} = 0 \tag{107.22}$$

This can happen at:

- Bifurcation points,

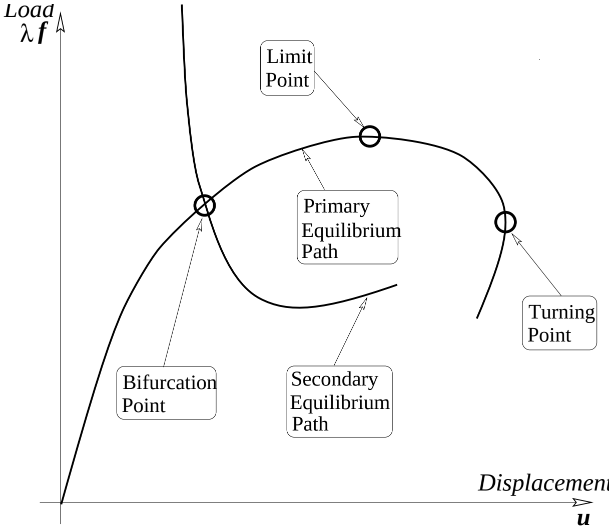


Figure 107.3: Simple illustration of Bifurcation and Turning point.

- Turning points,

The treatment of bifurcation points is of a rather special nature and is left for the near future. Turning points[12] can be traversed by a modification of a previous rule, as described in the next section.

107.6.1.2 Angle Criterion

Near a turning point application of the positive work rule (107.21) causes the path to double back upon itself. When it crosses the turning point it reverses so the turning point becomes impassable. Physically, a positive work rule is incorrect because in passing a turning point the structure releases external work until another turning point is encountered.

To pass a turning point imposing a condition on the angle of the prediction vector proves more effective. The idea is to compute both solutions $\delta\lambda_1$ and $\delta\lambda_2$ and then both $\Delta\mathbf{p}_1^{new}$ and $\Delta\mathbf{u}_1^{new}$:

$$\Delta\mathbf{u}_1^{new} = \Delta\mathbf{u}^{old} + \delta\bar{\mathbf{u}} + \delta\lambda_1\delta\mathbf{u}_t \tag{107.23}$$

$$\Delta\mathbf{u}_2^{new} = \Delta\mathbf{u}^{old} + \delta\bar{\mathbf{u}} + \delta\lambda_2\delta\mathbf{u}_t \tag{107.24}$$

---

[12] One might ask "why treating turning points in a material nonlinear analysis?". The answer is rather simple: "try to prevent all unnecessary surprises". For a good account of some of surprises in material nonlinear analysis one might take a look at some examples in Crisfield (1991) pp. 270.

The one that lies closest to the old incremental step direction $\Delta\mathbf{u}^{old}$ is the one sought. This should prevent the solution from double backing. The procedure can be implemented by finding the solution with the minimum angle between $\Delta\mathbf{u}^{old}$ and $\Delta\mathbf{p}^{new}$, and hence the maximum cosine of the angle:

$$\cos\phi = \frac{\left(\Delta\mathbf{u}^{old}\right)^T \left(\Delta\mathbf{u}^{new}\right)}{\|\Delta\mathbf{u}^{old}\| \, \|\Delta\mathbf{u}^{new}\|} = \frac{\left(\Delta\mathbf{u}^{old}\right)^T \left(\Delta\mathbf{u}^{old} + \delta\bar{\mathbf{u}} + \delta\lambda\delta\mathbf{u}_t\right)}{\|\Delta\mathbf{u}^{old}\| \, \|\Delta\mathbf{u}^{old} + \delta\bar{\mathbf{u}} + \delta\lambda\delta\mathbf{u}_t\|} \tag{107.25}$$

where $\delta\bar{\mathbf{u}} = -\mathbf{K}_t^{-1}\mathbf{r}^{old}$ and $\delta\mathbf{u}_t = \mathbf{K}_t^{-1}\mathbf{f}_{ext}$. Once the turning point has been crossed, the work criterion should be reversed so the external work is negative.

By directly introducing the constraint from equation (107.6) and following the method through equations (107.12) to (107.25) a limitation is introduced. Precisely at the limit point, $\mathbf{K}_t$ is singular and the equations cannot be solved. However, Batoz and Dhatt (1979) and Crisfield (1991) report that no such problem has occurred, because one appears never to arrive precisely at limit point.

107.6.2 Predictor step

The predictor solution is achieved by applying one forward Euler, explicit step from the last obtained equilibrium point:

$$\Delta\mathbf{u}_p = \mathbf{K}_t^{-1}\Delta\mathbf{q}_e = \Delta\lambda_p\mathbf{K}_t^{-1}\mathbf{f}_{ext} = \Delta\lambda_p\delta\mathbf{u}_t \tag{107.26}$$

where $\mathbf{K}_t$ is the tangent stiffness matrix at the beginning of increment. Substituting equation (107.26) into the constraint equation (107.14) one obtains:

$$\begin{aligned}
\left(\frac{\psi_u^2}{u_{ref}^2}\left(\Delta\mathbf{u}^{new}\right)^T \mathbf{S}\left(\Delta\mathbf{u}^{new}\right) + \left(\Delta\lambda^{new}\right)^2\psi_f^2\right) &= \\
\left(\frac{\psi_u^2}{u_{ref}^2}\Delta\lambda_p^2\delta\mathbf{u}_t^T\mathbf{S}\delta\mathbf{u}_t\right) + \left(\Delta\lambda_p\right)^2\psi_f^2 &= \\
\Delta\lambda_p^2\left(\frac{\psi_u^2}{u_{ref}^2}\delta\mathbf{u}_t^T\mathbf{S}\delta\mathbf{u}_t + \psi_f^2\right) &= (\Delta l)^2
\end{aligned} \tag{107.27}$$

The solution for $\Delta\lambda_p$ is readily found:

$$\Delta\lambda_p = \pm\frac{\Delta l}{\sqrt{\frac{\psi_u^2}{u_{ref}^2}\left|\delta\mathbf{u}_t^T\mathbf{S}\delta\mathbf{u}_t\right| + \psi_f^2}} \tag{107.28}$$

where $\Delta l > 0$ is the given increment length. The absolute value of $|\delta \mathbf{u}_t^T \mathbf{S} \delta \mathbf{u}_t|$ is needed if the stiffness matrix is chosen as a scaling matrix, i.e. $\mathbf{S} = \mathbf{K}_t$, since, after passing limit point, the stiffness matrix is non–positive definite so $\delta \mathbf{u}_t^T \mathbf{S} \delta \mathbf{u}_t \leq 0$. The question of choosing the right sign + or – in (107.28) is still a vigorous research topic. In the simplified procedure[13] the negative sign – is chosen with respect to the occurrence of one negative pivot during factorization of the tangent stiffness matrix $\mathbf{K}_t$. If more than one pivot happens to be negative, one is advised[14] to stop the analysis and try to restart from previously converged solution with smaller step size.

107.6.3 Automatic Increments

A number of workers have advocated different strategies for controlling the step length size. In this work we will follow the strategy advocated by Crisfield (1991). The idea is to find the new incremental length by applying:

$$\Delta l^{new} = \Delta l^{old} \left(\frac{I_{desired}}{I_{old}} \right)^n \tag{107.29}$$

where Δl^{old} is the old incremental factor for which I_{old} iterations were required, $I_{desired}$ is the input, desired number of iterations[15] and the parameter n is set to $\frac{1}{2}$ as suggested by Ramm (1982) Ramm (1981).

107.6.4 Convergence Criteria

Introduction of an iterative scheme calls for the introduction of an iteration termination test. There are several convergence criteria that can be applied.

- Absolute Displacement Convergence Criteria. The change in the last correction $\delta \mathbf{u}$ of the state vector $\mathbf{u}$, measured in an appropriate norm, should not exceed a given tolerance ϵ_u. For example, if we use Euclidean norm[16] the termination criteria can be written as:

  $$||\delta \mathbf{u}||_{absolute} = \sqrt{(\delta \mathbf{u})^T \mathbf{S} (\delta \mathbf{u})} \leq \epsilon_u \tag{107.30}$$

  Scaling matrix $\mathbf{S}$ is used in order to ensure that for a problem involving mixed units[17], all parameters have the same unit. Here, an obvious choice for the scaling matrix is $\mathbf{S} = diag(\mathbf{K}_t^{-1})$. If, on the

---

[13] Which is not guaranteed to work if one takes into account bifurcation phenomena.

[14] For more details see Crisfield (1991).

[15] Say $I_{desired} \approx 3$.

[16] The so called 2–norm.

[17] For example, if rotations and displacements are involved.

other hand we don't have mixed variables in state vector $\mathbf{u}$ then the simplest choice for scaling matrix is $\mathbf{S} = \mathbf{I}$.

Currently used within FEI and the Real-ESSI program is an absolute convergence criteria with unit scaling matrix $\mathbf{S} = \mathbf{I}$. This means that absolute tolerance criteria mixes units for different Degrees of Freedom (DoFs). Supplied tolerance is converted to basic units of the system (meter and Newton) and that is used for comparison and convergence decisions.

- Relative Displacement Convergence Criteria. It is beneficial to use the relative convergence criteria in order to relax convergence criteria for problems where one or few displacements dominate. In order to to that, use of ratio of Euclidean norm of iterative displacement $||\delta\mathbf{u}||_{scaled}$ and Euclidean norm of total displacement $||\mathbf{u}||_{scaled}$ is recommended:

$$\frac{||\delta\mathbf{u}||_{relative}}{||\mathbf{u}||_{scaled}} \leq \epsilon_u \tag{107.31}$$

Again it is important to note that currently used within FEI and the Real-ESSI program is an absolute convergence criteria with unit scaling matrix $\mathbf{S} = \mathbf{I}$. This means that relative tolerance criteria mixes units for different Degrees of Freedom (DoFs). Supplied tolerance is converted to basic units of the system (meter and Newton) and that is used for comparison and convergence decisions.

- Average Displacement Convergence Criteria. The change in the last correction $\delta\mathbf{u}$ of the state vector $\mathbf{u}$, measured in an appropriate norm, should not exceed a given tolerance ϵ_u, divided by the total number of DoFs:

$$||\delta\mathbf{u}||_{average} = \frac{\sqrt{(\delta\mathbf{u})^T\,\mathbf{S}\,(\delta\mathbf{u})}}{nDoFs} \leq \epsilon_u \tag{107.32}$$

This is important in order to preserve objectivity of displacement convergence criteria for similar (same) models that are discretized with a different number of finite elements (and therefor feature different number of nodes and DoFs). For example, a cantilever (simplest model) can be discretized using 5 DoFs and 5,000,000 DoFs. Since equation 107.32 essentially sums up absolute values of all displacements, it is expected that in the case of larger number of DoFs, larger norm will be calculated. This will create a problem since specified tolerance will then be a function of a number of DoFs a model features. Hence, a norm of all iterative displacements is divided by the total number of DoFs.

One more time, it is important to note that currently used within FEI and the Real-ESSI program is an absolute convergence criteria with unit scaling matrix $\mathbf{S} = \mathbf{I}$. Therefor average tolerance criteria mixes units. Supplied tolerance is converted to basic units of the system (meter and Newton) and that is used for comparison and convergence decisions.

- Absolute Residual Force Convergence Criteria. Since the residual $\mathbf{r}(\mathbf{u}, \lambda)$ measures the departure from the equilibrium path, an appropriate convergence test would be to compare Euclidean norm of residual with some predefined tolerance:

  $$\|\mathbf{r}(\mathbf{u}, \lambda)\|_{scaled} = \sqrt{(\mathbf{r})^T \, \mathbf{S} \, (\mathbf{r})} \leq \epsilon_r \tag{107.33}$$

  Here, an obvious choice for scaling matrix is $\mathbf{S} = diag(\mathbf{K}_t)$

  Much the same as for displacement based convergence criteria, currently used within FEI and the Real-ESSI program is a convergence criteria with unit scaling matrix $\mathbf{S} = \mathbf{I}$. This means that absolute tolerance criteria mixes units for different Degrees of Freedom (DoFs). Supplied tolerance is converted to basic units of the system (meter and Newton) and that is used for comparison and convergence decisions.

- Relative Residual Force Convergence Criteria. In order to provide scaling of residual forces that are used to tolerance criteria, (previously defined) absolute residual force norm is divided (scaled) by a a norm of residual forces at the beginning of the iterative step:

  $$\|\mathbf{r}(\mathbf{u}, \lambda)\|_{relative} = \frac{\sqrt{(\mathbf{r})^T \, \mathbf{S} \, (\mathbf{r})}}{\sqrt{(\mathbf{r}_0)^T \, \mathbf{S} \, (\mathbf{r}_0)}} \leq \epsilon_r \tag{107.34}$$

  Here, an obvious choice for scaling matrix is $\mathbf{S} = diag(\mathbf{K}_t)$, however, within FEI and the Real-ESSI program, for this convergence criteria, a unit scaling matrix $\mathbf{S} = \mathbf{I}$ is used, therefor relative tolerance criteria will feature mixed units.

- Average Residual Force Convergence Criteria. With the residual $\mathbf{r}(\mathbf{u}, \lambda)$, which measures the departure from the equilibrium path, there is a need to take into the account number of DoFs, in order to reduce the influence of a model discretization (number of DoFs) on norm of the residual force vector:

  $$\|\mathbf{r}(\mathbf{u}, \lambda)\|_{scaled} = \frac{\sqrt{(\mathbf{r})^T \, \mathbf{S} \, (\mathbf{r})}}{nDoFs} \leq \epsilon_{average} \tag{107.35}$$

  Here, an obvious choice for scaling matrix is $\mathbf{S} = diag(\mathbf{K}_t)$ however, again, as noted avove, within FEI and the Real-ESSI program, for this convergence criteria, a unit scaling matrix $\mathbf{S} = \mathbf{I}$ is used, therefor relative tolerance criteria will feature mixed units.

- Energy Based Convergence Criteria. The previous convergence criteria can be combined in a single work change criterion:

  $$\| \, (\delta \mathbf{u})^T \, (\mathbf{r}) \, \| = \sqrt{(\delta \mathbf{u})^T \, (\mathbf{r})} \leq \epsilon_u \epsilon_r \tag{107.36}$$

  A word of caution is appropriate at this point. As pointed out by Crisfield (1991), it follows that:

  $$\| \, (\delta \mathbf{u})^T \, (\mathbf{r}) \, \| = \| \, (\delta \mathbf{u})^T \left(\mathbf{K}_t^{-1} \mathbf{K}_t \right) (\mathbf{r}) \, \| = \| - (\delta \mathbf{u})^T \, \mathbf{K}_t \, (\delta \mathbf{u}) \, \| \leq \epsilon_u \epsilon_r \tag{107.37}$$

  where the iterative change was $(\delta \mathbf{u}) = -\mathbf{K}_t^{-1} \mathbf{r}$. It should be noted that equations (107.37) give a measure of the "stiffness" of $\mathbf{K}_t$. This merely implies that a stationary energy position has been reached in the current iterative direction, $\delta \mathbf{u}$. This can occur when the solution is still far away from equilibrium.

Since $\mathbf{u}$ and $\mathbf{r}$ usually have physical unites, so do necessarily ϵ_u and ϵ_r. For a general purpose implementation of Newton–Raphson iteration this dependency on physical units is undesirable and it is more convenient to work with ratios that render the ϵ_u and ϵ_r dimensionless. Displacement Convergence Criteria can be rendered dimensionless by using ratio of scaled Euclidean norm of iterative displacement $\|\delta \mathbf{u}\|_{scaled}$ and scaled Euclidean norm of total displacement $\|\mathbf{u}\|_{scaled}$:

$$\frac{\|\delta \mathbf{u}\|_{scaled}}{\|\mathbf{u}\|_{scaled}} \leq \epsilon_u \tag{107.38}$$

The similar approach can be used for Residual Convergence Criteria:

$$\frac{\|\mathbf{r}\|_{scaled}}{\|\mathbf{r}^{predictor}\|_{scaled}} \leq \epsilon_r \tag{107.39}$$

Another important thing to be considered is Divergence. The Newton–Raphson scheme is not guaranteed to converge. Some sort of divergence detection scheme is therefor necessary in order to interrupt an erroneous iteration cycle. Divergence can be diagnosed if either of following inequalities occur:

$$\frac{\|\delta \mathbf{u}\|_{scaled}}{\|\mathbf{u}\|_{scaled}} \geq g_u \tag{107.40}$$

$$\frac{\|\mathbf{r}\|_{scaled}}{\|\mathbf{r}^{predictor}\|_{scaled}} \geq g_r \tag{107.41}$$

where g_u and g_r are dangerous growth factors that can be set to, for example $g_u = g_r = 100$.

In some cases, the Newton–Raphson iteration scheme will neither diverge nor converge, but rather exhibit oscillatory behavior. To avoid excessive bouncing around, a good practice is to put upper limit to the number of iterations performed in one iteration cycle. Typical limits to the iteration number are 20 to 50.

107.6.5 The Algorithm Progress

The progress of the scheme will be briefly described, in relation with the Figure (107.1). The procedure starts from a previously converged solution $(\mathbf{u}_0, \lambda_0 \mathbf{f}_{ext})$. An incremental, tangential predictor step $\Delta\mathbf{u}_1, \Delta\lambda_1$ is obtained[18] and the next point obtained is $(\mathbf{u}_1, \lambda_1 \mathbf{f}_{ext})$. The first iteration would then use quadratic equation 107.17 where constants a_1, a_2 and a_3 should be computed with $\Delta\mathbf{u}^{old} = \Delta\mathbf{u}_1$ and $\Delta\lambda^{old} = \Delta\lambda_1$, to calculate $\delta\lambda_1$ and[19] $\delta\mathbf{u}_1 = -\mathbf{K}_t^{-1}\mathbf{r}(\mathbf{u}_1, \lambda_1) + \delta\lambda_1 \mathbf{K}_t^{-1}\mathbf{f}_{ext}$. After these values are calculated, the updating procedure[20] would lead to:

$$\Delta\lambda_2 = \Delta\lambda_1 + \delta\lambda_1 \qquad \text{and} \qquad \Delta\mathbf{u}_2 = \Delta\mathbf{u}_1 + \delta\mathbf{u}_1 \tag{107.42}$$

When added to the displacements $\mathbf{u}_0$ and load level λ_0, at the end of the previous increment this process would lead to the next point $(\mathbf{u}_2, \lambda_2 \mathbf{f}_{ext})$.

The next iteration would then again use quadratic equation 107.17 where constants a_1, a_2 and a_3 should be computed with $\Delta\mathbf{u}^{old} = \Delta\mathbf{u}_2$ and $\Delta\lambda^{old} = \Delta\lambda_2$, to calculate $\delta\lambda_3$ and $\delta\mathbf{u}_3 = \delta\bar{\mathbf{u}} + \delta\lambda_2 \delta\mathbf{u}_t$. After these values are calculated, the updating procedure would lead to:

$$\Delta\lambda_3 = \Delta\lambda_2 + \delta\lambda_2 \qquad \text{and} \qquad \Delta\mathbf{u}_3 = \Delta\mathbf{u}_2 + \delta\mathbf{u}_2 \tag{107.43}$$

When added to the displacements $\mathbf{u}_0$ and load level λ_0, at the end of the previous increment this process would lead to the next point $(\mathbf{u}_3, \lambda_3 \mathbf{f}_{ext})$.

---

[18] As explained in Section (107.6.2).

[19] From equation (107.12).

[20] See (107.11) and (107.13)

Chapter 108

Solution of Dynamic Equations of Motion

(1989-2006-2016-2018-2019-)

(In collaboration with Dr. Nima Tafazzoli and Prof. José Abell)

108.1 Chapter Summary and Highlights

108.2 The Principle of Virtual Displacements in Dynamics

(see section 102.2 on page 100).

Great reading on this subject is a book by Argyris and Mlejnek (1991).

108.3 Direct Integration Methods for the Equations of Dynamic Equilibrium

This section follows Argyris and Mlejnek (1991) and Hughes (1987).

need to rewrite and improve! BJ

108.3.1 Newmark Integrator

The Newmark time integration method (Newmark, 1959) uses two parameters, β and γ, and is defined by the following two equations:

$$
\begin{aligned}
{}^{n+1}x &= {}^{n}x + \Delta t\ {}^{n}\dot{x} + \Delta^2 t\ [(\frac{1}{2}-\beta)\ {}^{n}\ddot{x} + \beta\ {}^{n+1}\ddot{x}] && (108.1)\\
{}^{n+1}\dot{x} &= {}^{n}\dot{x} + \Delta t\ [(1-\gamma)\ {}^{n}\ddot{x} + \gamma\ {}^{n+1}\ddot{x}] && (108.2)
\end{aligned}
$$

These equatons give relationship between knowns variables at time step n to the unknown variables at next time step $n+1$. Method is in general an implicit one, except when $\beta = 0$ and $\gamma = 1/2$.

There are several possible implementation methods for Newmark Integrator. One possible approach to integrating equations of motion using Newmark algorithm is to use displacement as the basic unknowns, and the following difference relations are used to relate ${}^{n+1}\dot{x}$ and ${}^{n+1}\ddot{x}$ to ${}^{n+1}x$ and the response quantities are

$$
\begin{aligned}
{}^{n+1}\dot{x} &= \frac{\gamma}{\beta\Delta t}\left({}^{n+1}x - {}^{n}x\right) + \left(1-\frac{\gamma}{\beta}\right){}^{n}\dot{x} + \left(1-\frac{\gamma}{2\beta}\right){}^{n}\ddot{x} && (108.3)\\
{}^{n+1}\ddot{x} &= \frac{1}{\beta\Delta t^2}\left({}^{n+1}x - {}^{n}x\right) - \frac{1}{\beta\Delta t}{}^{n}\dot{x} + \left(1-\frac{1}{2\beta}\right){}^{n}\ddot{x} && (108.4)
\end{aligned}
$$

The predictors are then:

$$
\begin{aligned}
{}^{n+1}\dot{x}^{\diamond} &= -\frac{\gamma}{\beta\Delta t}{}^{n}x + \left(1-\frac{\gamma}{\beta}\right){}^{n}\dot{x} + \Delta t\left(1-\frac{\gamma}{2\beta}\right){}^{n}\ddot{x} && (108.5)\\
{}^{n+1}\ddot{x}^{\diamond} &= -\frac{1}{\beta\Delta t^2}{}^{n}x - \frac{1}{\beta\Delta t}{}^{n}\dot{x} + \left(1-\frac{1}{2\beta}\right){}^{n}\ddot{x} && (108.6)
\end{aligned}
$$

and the correctors:

$$
\begin{aligned}
{}^{n+1}\dot{x} &= {}^{n+1}\dot{x}^{\diamond} + \frac{\gamma}{\beta\Delta t}{}^{n+1}x && (108.7)\\
{}^{n+1}\ddot{x} &= {}^{n+1}\ddot{x}^{\diamond} + \frac{1}{\beta\Delta t^2}{}^{n+1}x && (108.8)
\end{aligned}
$$

The Newton integration method becomes

$$\left[\frac{M}{\beta\Delta t^2}+\frac{\gamma C}{\Delta t}+K\right]\Delta x=-{}^{n+1}R \tag{108.9}$$

Equation 108.5 to 108.9 constitute an iterative solving procedure (Argyris and Mlejnek, 1991).

If the parameters β and γ satisfy

$$\gamma\geq\frac{1}{2},\quad \beta\geq\frac{1}{4}(\gamma+\frac{1}{2})^2 \tag{108.10}$$

the procedure is unconditionally stable and second-order accurate. Any γ value greater than 0.5 will introduce numerical damping. Well-known members of the Newmark time integration method family include: trapezoidal rule or average acceleration method for $\beta = 1/4$ and $\gamma = 1/2$, linear acceleration method for $\beta = 1/6$ and $\gamma = 1/2$, and (explicit) central difference method for $\beta = 0$ and $\gamma = 1/2$. If and only if $\gamma = 1/2$, the accuracy is second-order Hughes (1987). For values of $\beta = 1$ and $\gamma = 2/3$, the strongest numerical damping is obtained, as spectral ratio $\rho_\infty = 0$ (Hughes (1987), page 502)

108.3.2 HHT Integrator

Numerical damping introduced in the Newmark time integration method will degrade the order of accuracy. The Hilber-Hughes-Tailor (HHT) time integration α-method (Hilber et al., 1977), (Hughes and Liu, 1978a) and (Hughes and Liu, 1978b) using an alternative residual form by introducing an addition parameter α to improve the performance:

$${}^{n+1}R=M\,{}^{n+1}\ddot{x}+(1+\alpha)F({}^{n+1}\dot{x},{}^{n+1}x)-\alpha F({}^{n}\dot{x},{}^{n}x)-\ {}^{n+1}f \tag{108.11}$$

but retaining the Newmark finite-difference formulas 108.1 and 108.2 or 108.3 and 108.4. If $\alpha = 0$, the HHT time integration method becomes exactly the same Newmark time integration method. Decreasing α value increase numerical dissipation (Hughes, 1987).

The iteration method for HHT time integration is similar to that of Newmark time integration. Due to the change of Equation 108.11, Equation 108.9 becomes

$$\left[M+(1+\alpha)\gamma\Delta tC+(1+\alpha)\beta\Delta t^2K\right]\Delta\ddot{x}=-{}^{n+1}R \tag{108.12}$$

for acceleration iteration and

$$\left[\frac{1}{\beta\Delta t^2}M+\frac{(1+\alpha)}{\gamma\Delta t}C+(1+\alpha)K\right]\Delta x=-{}^{n+1}R \tag{108.13}$$

for displacement iteration.

If the parameters α, β and γ satisfy

$$-1/3\leq\alpha\leq 0,\quad \gamma=\frac{1}{2}(1-2\alpha),\quad \beta=\frac{1}{4}(1-\alpha)^2 \tag{108.14}$$

it is unconditionally stable and second-order accurate (Hughes, 1987).

108.4 Synthetic Viscous Damping for Solids and Structures

Presented here are commonly used, viscous damping methods for time domain analysis of solids and structures. These methods, Rayleigh and Caughey damping, are mimicking viscous damping of of the solids and structures by generating the damping matrix $\mathbf{C}$ using mass and stiffness matrices.

These synthetic viscous damping approaches should be distinguished from a natural viscous damping that is created during interaction of fluid and soil. For example, a natural viscous damping occurs when pore fluid and porous solid have differential displacements, as described in section 102.12.1.4 on page 143 (see for example equation ?? on page ??).

108.4.1 Synthetic Viscous Damping Approaches

There are different numerical methods available to simulate the seismic wave propagation through the soil-structure systems such as boundary elements, finite elements, finite differences, meshfree methods, and spectral elements. There are advantages of using methods such as finite elements or spectral elements for complex geometries or modeling the nonlinearities but also disadvantage such as numerical dispersion for low-order finite elements or reflection of the motions from the boundaries of the model (Semblat et al. (2010)).

Boundary element method can deal better with the issue of reflecting the motions from boundaries comparing to other numerical methods. Research has been done on coupling this method with other numerical methods for better applications. Domain reduction method is also available for large models implemented in finite element in order to reduce the problem of reflection (Bielak et al. (2003a)).

There are methods so called non-reflecting boundary conditions which directly can attenuate the reflections at the mesh boundaries. One of the commonly used method is absorbing boundary conditions. Absorbing boundary conditions have special conditions at the model boundaries in order to approximate the radiation condition for seismic waves (Givoli (1991)).

Another method applicable in finite element methods is so called infinite element method. These elements can absorb the waves using decaying laws at model boundaries at infinity (Nenning and Schanz (2010); Kallivokas et al. (1997)). In this method it is assumed that the element and nodes of the boundary are in infinity. In this case the seismic waves have enough distance to dissipate at the boundaries and not to reflect back to the model.

There has been a recently developed method to prevent reflection of the waves from boundaries called Perfectly Matched Layers (PML). This absorbing layer is based on attenuation laws with specific properties and finite thickness located at the model boundaries. There are several PML formulations proposed for finite element methods which allows the treatment of surface waves as well as body waves

(Festa and Nielsen (2003); Basu (2009)).

Bilbao et al. (2006) proposed two energy-based methods to model damping in structures with added viscoelastic dampers. these methods approximate the effects of the added viscoelastic dampers with a damping matrix in the form of Caughey damping matrix.

The method to be used here is so called Caughey Absorbing Layer Method. The 2^{nd} order of this method is also known as a Rayleigh damping. Caughey damping is a classical method in which the damping matrix is built based on the mass and stiffness matrices. Since the stiffness and mass matrices of have to be created for solving the system of equations, they are used for creating the damping matrix as well.

Considering the relationship between internal friction and frequency for damping, it is possible to build a model involving the same attenuation/frequency dependence for Caughey damping (Semblat (1997)). The relation of the inverse of the quality factor Q^{-1} and the damping ratio ξ can be written as:

$$Q^{-1} \approx 2\xi \tag{108.15}$$

Caughey damping formulation in general can be expressed as

$$C = [M] \sum_{j=0}^{m-1} a_j ([M]^{-1}[K])^j \tag{108.16}$$

where the order to be used depends on number of modes to be considered for damping in the problem.

The way it is implemented in ESSI Simulator gives the opportunity to the user to use different types of damping for different elements. There might not be a need to use damping for all the elements of the model. In this case, damping could be used for particular elements and leave the rest of them with no physical Caughey damping. It can also be used for damping out the residual waves coming out of the domain reduction boundary layer.

108.4.2 Caughey Damping 2^{nd} Order, aka Rayleigh Damping

The second order Caughey damping, is also known as a Rayleigh damping, with $j = 1$ in Equation (108.16). From dynamic parameters and formulation of the system following equations can be observed:

$$\xi_n = \frac{C_n}{2M_n\omega_n} \tag{108.17}$$

$$K_n = \omega_n^2 M_n \tag{108.18}$$

Considering the first two terms in Caughey damping formulation, if the damping matrix formulation is written separately for each term:

$$C = a_0 M \tag{108.19}$$

Based on Equation (108.17), a_0 can be written as:

$$a_0 = 2\xi\omega \tag{108.20}$$

Writing the damping matrix based on the second coefficient:

$$C = a_1 K = a_1 \omega^2 M \tag{108.21}$$

Then a_1 can be obtained as:

$$a_1 = \frac{2\xi}{\omega} \tag{108.22}$$

So the damping ratio of the n^{th} mode of the system is:

$$\xi_n = \frac{a_0}{2}\frac{1}{\omega_n} + \frac{a_1}{2}\omega_n \tag{108.23}$$

Presenting Equation (108.23) for first two modes in matrix form leads to:

$$\frac{1}{2}\begin{bmatrix} \frac{1}{\omega_i} & \omega_i \\ \frac{1}{\omega_j} & \omega_j \end{bmatrix}\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \xi_i \\ \xi_j \end{bmatrix} \tag{108.24}$$

The following procedure given by Hall (2006) is useful to conveniently determine Rayleigh damping coefficients a_0 and a_1. Select a desired amount of damping ξ and a frequency range from $\hat{\omega}$ to $R\hat{\omega}$, where $R > 1$. Compute Δ from:

$$\Delta = \xi\frac{1 + R - 2\sqrt{R}}{1 + R + 2\sqrt{R}} \tag{108.25}$$

where Δ determines bounds on the damping ratios that are imparted to those modes within the specified frequency range. Any such mode will have a damping ratio bounded by $\xi_{max} = \xi + \Delta$ and $\xi_{min} = \xi - \Delta$. If these bounds are satisfactorily narrow, the constants a_0 and a_1 are then calculated from:

$$a_0 = 2\xi\hat{\omega}\frac{2R}{1 + R + 2\sqrt{R}} \tag{108.26}$$

$$a_1 = 2\xi\frac{1}{\hat{\omega}}\frac{2}{1 + R + 2\sqrt{R}} \tag{108.27}$$

and can be used to compute an actual damping value ξ_n for mode n from Equation (108.23).

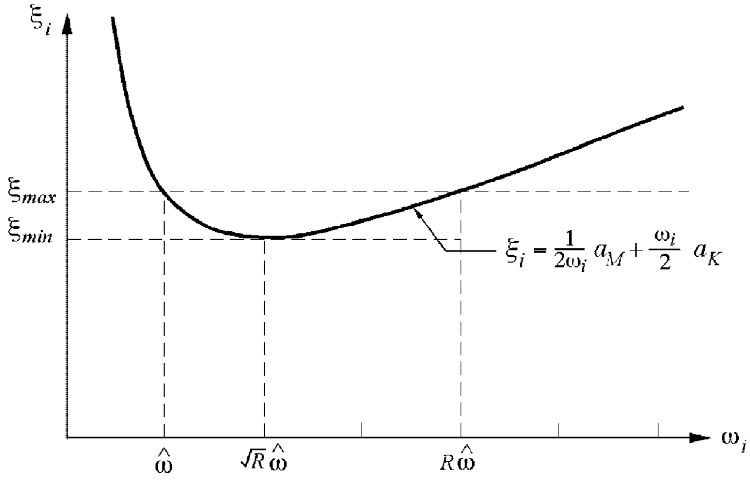


Figure 108.1: Actual damping ratio ξ_n of mode n as a function of frequency ω_n of mode n when using Rayleigh damping (Hall (2006)).

Figure (108.1) shows that $\xi_n = \xi_{max}$ if $\omega_n = \hat{\omega}$ or $\omega_n = R\hat{\omega}$, and that $\xi_n = \xi_{min}$ if $\omega_n = \sqrt{R}\hat{\omega}$. If ω_n is outside the range $\hat{\omega}$ to $R\hat{\omega}$, then $\xi_n > \xi_{max}$. Above $R\hat{\omega}$, ξ_n increases with ω_n, approaching a linear relation as the last term in Equation (108.23) dominates.

Note: It's worth pointing out that damping could be unrealistically high for motions outside the prescribed frequency range, if the frequency range is not well-chosen.

108.4.3 Caughey Damping 3^{rd} Order

Following the same logic as the 2^{nd} order, the last coefficient of the 3^{rd} order Caughey damping formulation can be written as following:

$$C = a_2 K M^{-1} K = a_2 \omega^4 M \tag{108.28}$$

So a_2 can be obtained as:

$$a_2 = \frac{2\xi}{\omega^3} \tag{108.29}$$

Considering the last coefficient in the formulation, the damping ratio of the system can now be shown as:

$$\xi_n = \frac{a_0}{2}\frac{1}{\omega_n} + \frac{a_1}{2}\omega_n + \frac{a_2}{2}\omega_n^3 \tag{108.30}$$

By solving the following set of equations, 3^{rd} order Caughey damping coefficients can be found:

$$\frac{1}{2}\begin{bmatrix} \frac{1}{\omega_i} & \omega_i & \omega_i^3 \\ \frac{1}{\omega_j} & \omega_j & \omega_j^3 \\ \frac{1}{\omega_k} & \omega_k & \omega_k^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \xi_i \\ \xi_j \\ \xi_k \end{bmatrix} \tag{108.31}$$

108.4.4 Caughey Damping 4^{th} Order

The 4^{th} coefficient of the Caughey damping formulation cab obtained as:

$$C = a_3 K M^{-1} K M^{-1} K = a_3 \omega^6 M \tag{108.32}$$

$$a_3 = \frac{2\xi}{\omega^5} \tag{108.33}$$

$$\xi_n = \frac{a_0}{2}\frac{1}{\omega_n} + \frac{a_1}{2}\omega_n + \frac{a_2}{2}\omega_n^3 + \frac{a_3}{2}\omega_n^5 \tag{108.34}$$

So the damping coefficients can be obtained by solving the following set of equations:

$$\frac{1}{2}\begin{bmatrix} \frac{1}{\omega_i} & \omega_i & \omega_i^3 & \omega_i^5 \\ \frac{1}{\omega_j} & \omega_j & \omega_j^3 & \omega_j^5 \\ \frac{1}{\omega_k} & \omega_k & \omega_k^3 & \omega_k^5 \\ \frac{1}{\omega_l} & \omega_l & \omega_l^3 & \omega_l^5 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \xi_i \\ \xi_j \\ \xi_k \\ \xi_l \end{bmatrix} \tag{108.35}$$

Chapter 109

Earthquake Soil Structure Interaction, Theoretical Aspects

(1989-2001-2006-2016-2018-2019-2020-2021-)

(In collaboration with Dr. Nima Tafazzoli, Prof. José Abell, Dr. Yuan Feng, Dr. Hexiang Wang)

109.1 Chapter Summary and Highlights

109.2 Seismic Energy Propagation and Dissipation

Jeremić (2010)

109.2.1 Seismic energy input into SSI system

Earthquakes release large amounts energy at the source[1] Part of released energy is radiated as mechanical waves ($\approx 1.6 \times 10^{-5}$) and part of that energy makes it to the surface where SSI system is located.

Mechanical seismic wave energy enters the SSI system through a closed surface Γ that encompasses (significant) soil volume as well as foundation system and the structure (see Figure 109.7). Kinetic energy flux through closed surface Γ includes both incoming and outgoing waves and can be calculated using Domain Reduction Method (Bielak et al., 2003a) as:

$$E_{flux} =$$
$$\left[0; -M^{\Omega+}_{be}\ddot{u}^0_e - K^{\Omega+}_{be}u^0_e; M^{\Omega+}_{eb}\ddot{u}^0_b + K^{\Omega+}_{eb}u^0_b\right]_i \times u_i$$

where $M^{\Omega+}_{be}$, $M^{\Omega+}_{eb}$, $K^{\Omega+}_{be}$, $K^{\Omega+}_{eb}$ are mass and stiffness matrices, respectively for a single layer of elements just outside of the boundary Γ, while $\ddot{u}^0_e$ and u^0_e are accelerations and displacements from a free field model for nodes belonging to that layer of elements. Alternatively, energy flux can be calculated using ((Aki and Richards, 2002), page 122):

$$E_{flux} = \rho A c \int_0^t \dot{u}_i^2 dt$$

Outgoing kinetic energy can be obtained from outgoing wave field w_i, (from DRM, Bielak et al. (2003a)), while the difference then represents the incoming kinetic energy that needs to be dissipated with SSI region.

109.2.2 Seismic Energy Dissipation in SSI System

Seismic energy that enters the SSI system will be dissipated in a number of ways. Part of the energy that enters SSI system is reflected and radiated back into domain outside Γ by

- wave reflection from impedance boundaries (free surface, soil/rock layers, foundations, etc.).
- SSI system oscillates and emits, radiates waves back into the domain

---

[1] for example, some of the recent large earthquake energy releases are listed: Northridge, 1994, $M_{Richter} = 6.7$, $E_r = 6.8 \times 10^{16} J$; Loma Prieta, 1989, $M_{Richter} = 6.9$, $E_r = 1.1 \times 10^{17} J$; Sumatra-Andaman, 2004, $M_{Richter} = 9.3$, $E_r = 4.8 \times 10^{20} J$; Valdivia, Chile, 1960, $M_{Richter} = 9.5$, $E_r = 7.5 \times 10^{20} J$;

The rest of seismic energy is dissipated through one of the following mechanisms within SSI system:

- Inelastic, elasto-plastic behavior of soil and rock
- Inelastic, elasto-plastic, damage behavior of the foundation system
- Inelastic, elasto-plastic, damage behavior of the structure
- Viscous coupling of porous solid with pore fluid (air, water)
- Viscous coupling of structure with surrounding, internal and external fluids (air, water)

It is also important to note that in numerical simulations, part of the energy can be dissipated or produced by purely numerical means. That is, numerical energy dissipation (damping) or production (negative damping) has to be carefully controlled (Argyris and Mlejnek, 1991), (Hughes, 1987).

109.2.2.1 Energy Dissipation by Plasticity

Elastic-plastic deformation of soil, foundation and structure is probably responsible for major part of the energy dissipation for large earthquakes. This, displacement proportional dissipation is a result of plastic dissipation and is present in all three components of the system (soil, foundation and the structure).

A note about plastic dissipation is important at this point. There is a misconception about plastic energy dissipation that is being widely used. Here are some details:

- Origins of the Misconception: The paper by Uang and Bertero (1990) has been considered the definitive work in using energy as a measure of structural demand by many researchers (Léger and Dussault, 1992; Cosenza et al., 1993; Kalkan and Kunnath, 2007, 2008; Symans et al., 2008; Chopra, 2000; Gajan and Saravanathiiban, 2011; Moustafa, 2011; Moustafa and Mahmoud, 2014; Mezgebo and Lui, 2017; Deniz et al., 2017). An energy analysis methodology based on absolute input energy (or energy demand) was presented and discussed. Numerical analysis results were compared with experiments on a multi-story building. In this paper, hysteretic energy is calculated indirectly by taking the difference of absorbed energy and elastic strain energy. The term *absorbed energy* of each time step was simply defined as force times incremental displacement. It was stated that *hysteretic energy* is irrecoverable, which indicates that this parameter was considered the same as *hysteretic dissipation* or *plastic dissipation*. An equation for energy balance is given:

$$E_i = E_k + E_\xi + E_a = E_k + E_\xi + E_s + E_h \tag{109.1}$$

where E_i is the (absolute) input energy, E_k is the (absolute) kinetic energy, E_ξ is the viscous damping energy, E_a is the absorbed energy, which is composed of elastic strain energy E_s and hysteretic energy E_h.

The problem of this theory is the absence of plastic free energy, which is necessary to correctly evaluate energy dissipation of elastic-plastic materials and to uphold the second law of thermodynamics. There was no direct plot of plastic dissipation (hysteretic energy) in this paper, since it was not defined directly. There were plots of other energy components and plastic dissipation can be easily calculated or deducted from these plots. After doing this, indications of negative incremental energy dissipation, which violates the basic principles of thermodynamics, were found in various sections of the paper.

This misconception could be clarified by renaming hysteretic energy to *plastic work*, which is the combination of plastic dissipation and plastic free energy. Both plastic work and plastic free energy can be incrementally negative, but plastic dissipation (defined as the difference of plastic work and plastic free energy) must be incrementally non-negative during any time period. Unfortunately, this misconception has been inherited (if not magnified) by almost all following studies on energy analysis of earthquake soils and structures.

Besides, another issue regarding energy dissipation is found in in this paper. *Viscous damping energy* (or viscous damping energy) was calculated directly using damping coefficient and velocity. The author stated that this term should always be non-negative. But it was ignored that the incremental viscous energy dissipation should also be non-negative. In fact, the equation used to compute viscous energy dissipation should be able to ensure that it remains non-negative incrementally. However, it appeared in one of the plots that (accumulated) viscous damping energy was clearly dropping during certain time periods, which was in contradiction with the equation derived in the same paper. Such result was also a violation of thermodynamics.

- Misconception in Other Studies: Although input energy was the key parameter used in Uang and Bertero (1990), the misconception of energy dissipation has been carried on in a vast number of studies on energy analysis of ESSI systems.

  Léger and Dussault (1992) used Equation 109.1 from Uang and Bertero (1990) to perform energy response analysis of multi-story buildings under earthquake loading. It was stated that the total input energy at the end of ground motion is approximately equal to the total dissipated energy. This statement is only valid if plastic free energy remains constant, which is generally not true in elastic-plastic materials.

  Kalkan and Kunnath (2007) calculated energy dissipation of a single-degree-of-freedom (SDOF) oscillator under earthquake loading. Inelastic material was used so plastic energy dissipation appeared in the results. Negative incremental hysteretic energy (plastic dissipation) was observed in the plots, which is a clear violation of thermodynamics. The change of plastic free energy was not considered in this study. Several papers by the same authors performed energy analysis on various structures using the same theory, and similar misconceptions can be noticed in these publications.

Symans et al. (2008) summarized current practice and recent developments in the application of passive energy dissipation systems for seismic protection of structures. There was no consideration of plastic free energy in the energy balance equation, which was very close to the ones present by Uang and Bertero (1990) and Léger and Dussault (1992). It was stated that the cumulative hysteretic energy is equal to the energy demand (absolute input energy) at the end of earthquake. Although no direct violation of thermodynamics was observed, such statement clearly indicated the same misconception of plastic work and plastic dissipation appeared in other publications.

Gajan and Saravanathiiban (2011) performed both numerical simulations and centrifuge experiments on a rocking foundation system. Energy dissipation in foundation soil and structural elements were calculated. It can be observed that hysteretic energy dissipation in both the soil and the structure was decreasing during certain time periods, which is a direct violation of thermodynamics. Again, this was a misconception of plastic work and plastic energy dissipation. The change in plastic free energy of the system was significant in this case, since large drops of plastic work were noticed in the plots.

Chopra (2000) used similar set of equations in section 7.9, and equation 7.9.6 is clearly wrong!

A number of recent studies Moustafa (2011); Moustafa and Mahmoud (2014); Mezgebo and Lui (2017); Deniz et al. (2017) performed energy analysis on ESSI systems without considering plastic free energy. Misleading results were obtained using the wrong energy balance equation. This means that the misconception of plastic work and plastic energy dissipation is still not realized by many researchers. Note that the influence of this mistake could be negligible or significant, depending on the case analyzed. Nevertheless, plastic free energy should not be ignored without plausible reasoning or experimental evidences.

- Early Studies on Plastic Free Energy (Cold Work): This issue has been pointed out and studied extensively by researchers from mechanical engineering and material science. In the early 20th century, Taylor and his colleagues Farren and Taylor (1925), Taylor and Quinney (1934) performed experiments on metals and proved that a large part, but not all, of the input mechanical energy is converted into heat. The remaining part of the non-recoverable plastic work is known as the stored energy of cold work. The ratio of plastic work converted into heating (Quinney–Taylor coefficient), usually denoted as β, has been used in almost all later papers on this topic. Based on large amount of experimental data, this ratio was assumed to be a constant between 0.6 to 1.0 in many studies (e.g. Clifton et al. (1984), Belytschko et al. (1991), Zhou et al. (1996), Dolinski et al. (2010), Ren and Li (2010), Osovski et al. (2013)). It has been realized that this assumption is not valid in all cases, but it's simply too complicated to involve the evolution of Quinney–Taylor coefficient in thermomechanical constitutive models.

Decades later, Mason et al. (1994) showed that the fraction of plastic work converted into heat is both strain and strain rate dependent. Infrared imaging was used in this study (and almost all future studies) to obtain temperature distribution in the material, because it is the only effective approach to directly measure energy dissipation (heat). In the recent 20 years, there has been many developments on this issue. Rittel (Rittel, 2000; Rittel and Rabin, 2000; Rittel et al., 2003) published several insightful papers on the energy dissipation (heat generation) of polymers during cyclic loading, presenting both experimental and theoretical works. Rosakis et al. (2000) presented a constitutive model based on thermoplasticity to model the evolution of in metal. This model is capable of calculating the evolution of energy dissipation and material properties, and is validated by sets of experiments. There are some follow-up papers by Rosakis and his colleagues (Hodowany et al., 2000; Ravichandran et al., 2002) on the same issue with some assumptions to simply the problem. One widely used assumption is the adiabatic condition, since air conducts heat much slower than metal. This assumption is reasonable in rapid monotonic or cyclic loading (impact, vibration, earthquake). One application of this theory in geotechnical engineering is presented in papers of (Veveakis et al., 2007, 2012), in which thermoporomechanics is used to model the heating and pore pressure increase in large landslides, like the 1963 Vajont slide in Italy.

In the field of civil engineering, the basic principles of thermodynamics are frequently used to derive new constitutive models Dafalias and Popov (1975); Ziegler and Wehrli (1987); Collins and Houlsby (1997); Houlsby and Puzrin (2000); Collins (2002); Collins and Kelly (2002); Collins (2003); Feigenbaum and Dafalias (2007). to enforce the second law of thermodynamics for developed constitutive models. Note that plastic free energy is the same concept as cold work. The former term is more popular in solid physics and geotechnical engineering, while the latter is used in mechanical engineering. energy dissipation due to plasticity and plastic work, which is often a source of a confusion. through a conceptual example that is analyzed on particle scale. development of plastic free energy is caused by particle rearrangement in granular assembly under external loading.

Ideally, majority of the incoming energy would be dissipated in soil, before reaching foundation and structures. The possibility to direct energy dissipation to soil can be used in design by recognizing energy dissipation capacity for different soils. For example, simple elastic-plastic models of stiff and soft clay as well as dense and loose send predict different energy dissipation capacities, as shown in Figure 109.1, for single loading-unloading-reloading cycle. While Figure 109.1 shows that stiff clay and dense sand have much higher dissipation capacity, it is important to note that soft/loose soils can undergo much larger deformation/strain, thus offering increased energy dissipation capacity through flexibility.

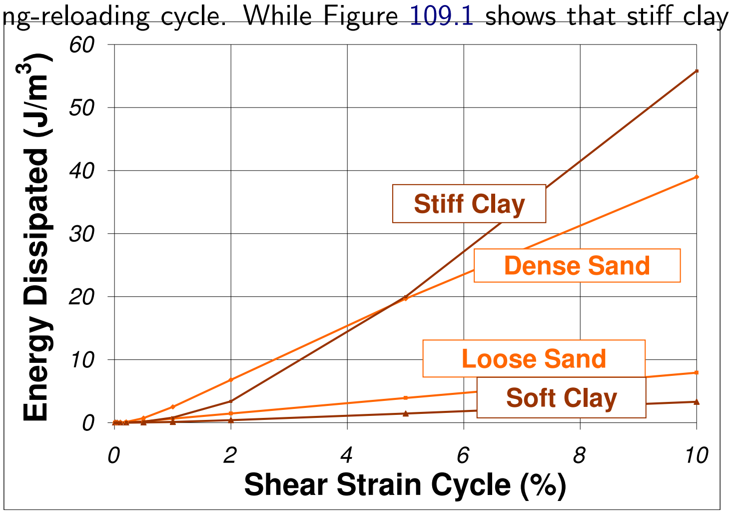


Figure 109.1: Energy dissipation capacity for one cycle at various strains for four generic soils.

109.2.2.2 Energy Dissipation by Viscous Coupling

Viscous coupling of pore fluid (air, water...) and soil particles and/or foundation or structural components is responsible for velocity proportional energy dissipation. In particular, viscous coupling of porous solid and fluid results in $E_{vc} = n^2 k^{-1} (\dot{U}_i - \dot{u}_i)^2$ energy loss per unit volume. It is noted that this type of dissipation is realistically modeled using $u-p-U$ formulation (Jeremić et al., 2008).

109.2.2.3 Numerical Energy Dissipation and Production

As noted above, numerical integration of nonlinear equations of motions affects calculated energy in various ways. Most common effect for nonlinear (elastic-plastic) systems is the positive (energy dissipation) and negative (energy production) damping. For example Newmark (N) (Newmark, 1959) and Hilber–Hughes–Taylor (HHT) (Hilber et al., 1977) are energy preserving for linear elastic system with proper choice of constants ($\alpha = 0.0; \beta = 0.25, \gamma = 0.5$). Both methods can also be used to dissipate higher frequency modes for linear elastic models by changing constants so that for N: $\gamma \geq 0.5, \quad \beta = 0.25(\gamma + 0.5)^2$, while for HHT: $-0.3\dot{3} \leq \alpha \leq 0, \quad \gamma = 0.5(1-2\alpha), \quad \beta = 0.25(1-\alpha)^2$. However, for nonlinear problems it is impossible to maintain energy of the system throughout computations (Argyris and Mlejnek, 1991).

109.2.2.4 Energy Dissipation by Nonlinearities in Soil/Rock

Elasto-plasticity of solid skeleton

Viscous (coupling) effects

109.2.2.5 Energy Dissipation by Nonlinearities in Soil/Rock – Foundation Interface Zone

Gap, Slip
  Dry
  Saturated

109.2.2.6 Energy Dissipation by Nonlinearities in Seismic Isolators

109.2.2.7 Energy Dissipation by Nonlinearities in Structures, Systems and Components

109.2.2.8 Numerical Energy Dissipation and Production

109.2.3 Seismic Motions: Empirical Models

109.2.4 1D/1C Wave Propagation Modeling

The theory of wave propagation is associated with vertical propagation of shear and/or comporessional waves through the linear viscoelastic system is described in this section. Developments shown here follow standard approach, as found by, for example, Kramer (1996a). Shown in Figure 109.2 is a 1C wave propagation setup. The model consists of N horizontal layers, that extend to infinity in the horizontal direction. At the bottom of layers is bedrock that represents a halfspace. Each layer is homogeneous and isotropic and is characterized by the thickness h, mass density ρ, shear wave velocity V_s[2], and compressional wave velocity V_p[3] and damping factor, β.

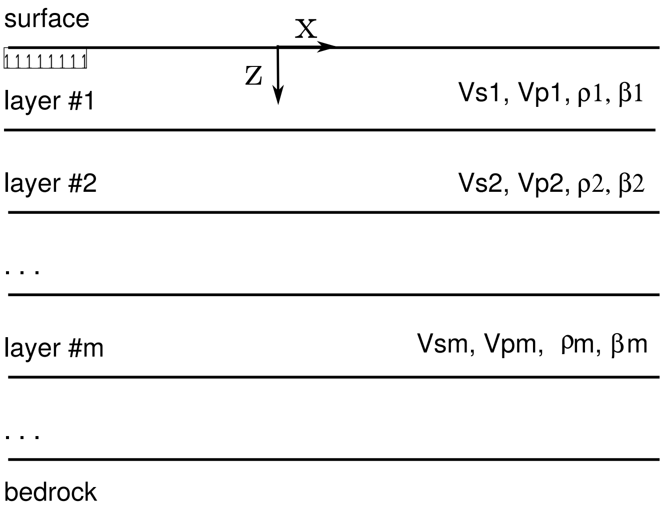


Figure 109.2: Problem Description: Wave Propagation

Vertical propagation of shear or compressional waves will cause only horizontal of vertical displacements respectively.

$$u = u(z, t) \tag{109.2}$$

Focusing on vertical propagation of shear waves, and with a presence of displacements in x direction only, a wave equation, that describes wave propagation in vertical, z directions can be written as

$$\rho \frac{\partial^2 u}{\partial t^2} = G \frac{\partial^2 u}{\partial z^2} + \eta \frac{\partial^3 u}{\partial z^2 \partial t} \tag{109.3}$$

---

[2] Shear Wave velocity is used to obtain shear shear modulus G, $V_s = \sqrt{G/\rho}$, $G = V_s^2 \rho$, $G = E/(2(1+\nu))$, $E = (9KG)/(3K+g)$, $E = (G(3M-4G))/(M-G))$.

[3] Compressional wave velocity is lused to obtain constrained modulus M $V_p = \sqrt{M/\rho}$, $M = V_p^2 \rho$, $M = K + 4G/3$, $M = E(1-\nu)/((1+\nu)(1-2\nu)))$.

Harmonic oscillation, displacements with frequency ω, can be written in the as:

$$u(z,t) = U(z) \cdot e^{i\omega t} \tag{109.4}$$

Substituting Eq (109.4) into Eq (109.3) one obtains

$$(G + i\omega\eta)\frac{\partial^2 u}{\partial z^2} = \rho\omega^2 U \tag{109.5}$$

which has the general solution:

$$U(z) = Ee^{ikz} + Fe^{-ikz} \tag{109.6}$$

in which

$$k^2 = \frac{\rho\omega^2}{G + i\omega\eta} = \frac{\rho\omega^2}{G^*} \tag{109.7}$$

where k is the complex wave number and G^* is the complex shear modulus. The critical damping ratio, β, is related to the viscosity η by

$$\omega\eta = 2G\beta \tag{109.8}$$

For convenience, one can use use soil damping ratio β to represent the complex shear modulus.

$$G^* = G + i\omega\eta = G(1 + 2i\beta) \tag{109.9}$$

By combine Eq. (109.4) and Eq. (109.6) one obtains the wave equation for a harmonic motion of frequency ω.

$$u(z,t) = Ee^{i(kz+\omega t)} + Fe^{-i(kz-\omega t)} \tag{109.10}$$

where the first term represents the incident wave traveling in the upward, in positive z direction and the second term represents the reflected wave traveling in the negative, downward z–direction.

This equation is valid for each of the soil and rock layers.

Introducing a local coordinate system Z for each layer, the displacements at the top and bottom of layer m are :

$$u_m(z = 0) = (E_m + F_m)e^{i\omega t} \tag{109.11}$$

$$u_m(z = h_m) = (E_m \cdot e^{ik_m h_m} + F_m e^{-ik_m h_m}) \cdot e^{i\omega t} \tag{109.12}$$

The shear stress on a horizontal plane is

$$\tau(z,t) = G \cdot \frac{\partial u}{\partial z} + \eta \frac{\partial u}{\partial z \partial t} = G^{*} \frac{\partial u}{\partial z} \tag{109.13}$$

In another form,

$$\tau(z,t) = ikG^{*}(Ee^{ikz} - Fe^{-ikz})e^{i\omega t} \tag{109.14}$$

and the shear stress at the top and bottom of layer m are respectively:

$$\tau_m(z=0) = ik_m G^{*}_m (E_m - F_m) e^{i\omega t} \tag{109.15}$$

$$\tau_m(z=h_m) = ik_m G^{*}_m (Ee^{ik_m h_m} - Fe^{-ik_m h_m}) e^{i\omega t} \tag{109.16}$$

Stresses and displacements must be continuous at all interfaces. Hence, by Eq (109.11), (109.12), (109.15) and (109.16), the coefficients are

$$E_{m+1} + F_{m+1} = E_m e^{ik_m h_m} + F_m e^{-ik_m h_m} \tag{109.17}$$

$$E_{m+1} - F_{m+1} = \frac{k_m G^{*}_m}{k_{m+1} G^{*}_{m+1}} (E_m e^{ik_m h_m} - F_m e^{-ik_m h_m}) \tag{109.18}$$

Subtraction and addition Eqs. 15 and 16 yield the following recursion formulas for the amplitudes, E_{m+1} and F_{m+1}, of the incident and reflected wave in layer m+1, expressed in terms of the amplitudes in layer m:

$$E_{m+1} = \frac{1}{2} E_m (1+\alpha_m) e^{ik_m h_m} + \frac{1}{2} F_m (1-\alpha_m) e^{-ik_m h_m} \tag{109.19}$$

$$F_{m+1} = \frac{1}{2} E_m (1-\alpha_m) e^{ik_m h_m} + \frac{1}{2} F_m (1+\alpha_m) e^{-ik_m h_m} \tag{109.20}$$

where α_m is the complex impedance ratio

$$\alpha_m = \frac{k_m G^{*}_m}{k_{m+1} G^{*}_{m+1}} = (\frac{\rho_m G^{*}_m}{\rho_{m+1} G^{*}_{m+1}})^{1/2} \tag{109.21}$$

At the free surface, the shear stresses must be zero. In addition, according to Eq. (109.15), it obtains $E_1 = F_1$. Namely, the amplitudes of the incident and reflected waves are always equal at the free surface. Beginning with the surface layer, repeated use of Eq (109.19) and Eq (109.20) to build the wave field:

$$E_m = e_m(\omega) E_1 \tag{109.22}$$

$$F_m = f_m(\omega)E_1 \tag{109.23}$$

The transfer function e_m and f_m are simply the amplitudes for the case $E_1 = F_1 = 1$, and can be determined by substituting this condition into the above recursion formulas. Other transfer functions are easily obtained from the e m and f m functions. The transfer function A n,m between the displacements at level n and m is defined by:

$$A_{n,m}(\omega) = \frac{u_m}{u_n} \tag{109.24}$$

and can be represented by

$$A_{n,m}(\omega) = \frac{e_m(\omega) + f_m(\omega)}{e_n(\omega) + f_n(\omega)} \tag{109.25}$$

Based on these equations, the transfer function $A(w)$ can be found between any two layers in the system. Hence, if the motion is known in any one layer in the system, the motion can be computed in any other layer.

In summary, after the E and F are computed for all layers in the system, the accelerations are expressed by the equaction:

$$\ddot{u}(z,t) = \frac{\partial^2 u}{\partial t^2} = -\omega^2 (Ee^{i(kz+\omega t) + Fe^{-i(kz-\omega t)}}) \tag{109.26}$$

109.2.5 Seismic Motions: 3D/3C Analytic Wave Propagation Modeling

Thomson (1950); Haskell (1953)

This is based in part on Wang et al. (2020a).

Considered is the inclined wave propagation in the layered ground as shown in Fig. 109.3. There are n layers with layer thickness d_m, density ρ_m, compressional velocity α_m and shear wave velocity β_m ($m = 1, 2, .., n$). Since the incidence of out-of-plane SH wave is simpler (no mode conversion), here we focus on the incidence of P and SV wave and account for the mode conversion between them. The wave potential formulations below are general and also applicable to incident SH wave (Haskell, 1953). Without loss of generality, incident waves is considered to be monochromatic with angular frequency w and horizontal phase velocity c. For incident waves with arbitrary time signal and multiple frequencies, free field motions can be Fourier synthesized from the monochromatic solutions.

According to Helmholtz decomposition theorem (Arfken and Weber, 1999), the displacement of wave propagation Eq. (109.27) in linear elastic media can be expressed with P wave scalar potential ϕ and S wave vector potential $\mathbf{\Psi}$ as shown in Eq. 109.28, where ϕ is the curl free part corresponding to volumetric deformation and Ψ is divergence free part corresponding to deviatoric deformation.

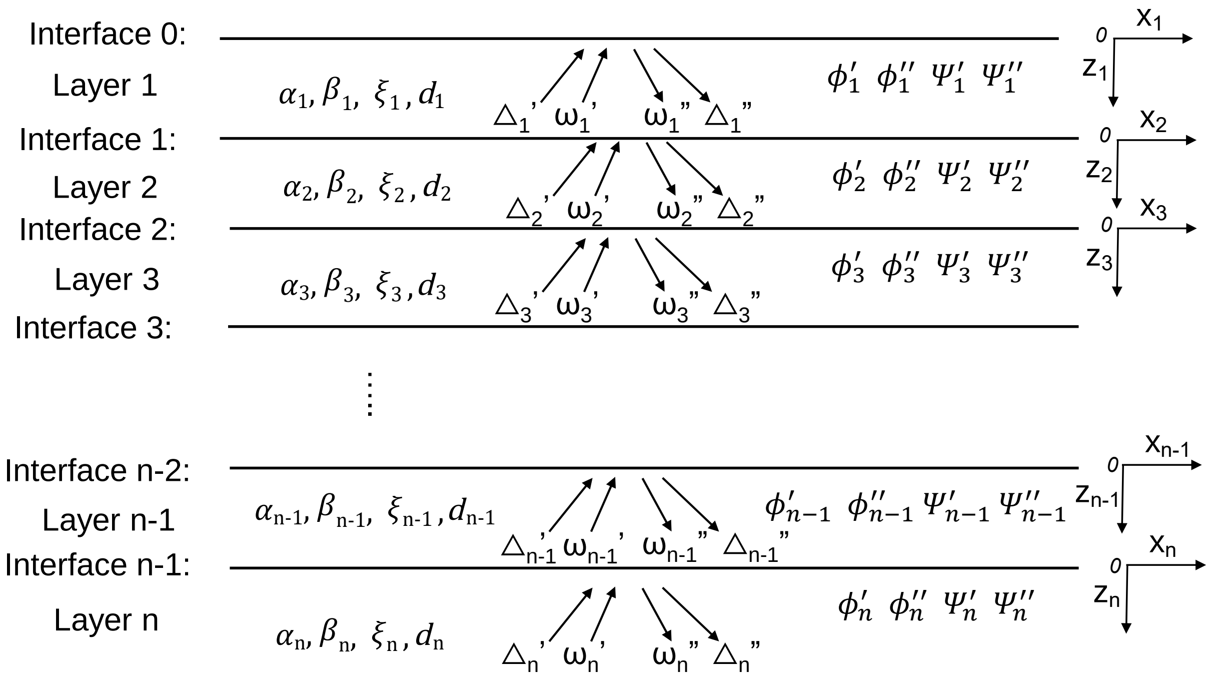


Figure 109.3: 2D layered ground and free field motion

$$\rho \ddot{\boldsymbol{u}} = (\lambda + 2\mu)\nabla\nabla \cdot \boldsymbol{u} - \mu\nabla \times \nabla \times \boldsymbol{u} \tag{109.27}$$

$$\boldsymbol{u} = \nabla\phi + \nabla \times \Psi \tag{109.28}$$

Therefore, the unknown variables for m^{th} layer are simplified into incident P wave potential magnitude $\phi_m^{'}$, reflected P wave potential magnitude $\phi_m^{''}$, incident SV wave potential magnitude $\Psi_m^{'}$ and reflected SV wave potential magnitude $\Psi_m^{''}$.

$$\begin{aligned} \phi_m &= [\phi_m^{'} e^{ik(x-\gamma_{\alpha m} z)} + \phi_m^{''} e^{ik(x+\gamma_{\alpha m} z)}] e^{-iwt} \\ \Psi_m &= [\Psi_m^{'} e^{ik(x-\gamma_{\beta m} z)} + \Psi_m^{''} e^{ik(x+\gamma_{\beta m} z)}] e^{-iwt} \end{aligned} \tag{109.29}$$

The P and SV wave potential can be expressed as Eq. (109.29), where k is the horizontal wave number, equals to w/c. And $cot^{-1}\gamma_{\alpha_m}$ and $cot^{-1}\gamma_{\beta_m}$ are incident and reflected angles for P and SV wave, respectively. The harmonic nature of the potential field is characterized by the time factor e^{-iwt}. It will be understood and omitted hereafter. γ_{α_m} and γ_{β_m} can be determined by Snell's law (equation ??). Note that when α_m or β_m is greater than c, the incidence from P or SV wave is beyond the critical angle. γ_{α_m} and γ_{β_m} become complex numbers. The plane wave magnitude exponentially decays along the depth. To be consistent with the original formulation by Haskell (1953), dilatational wave solutions Δ_m and rotational wave solutions ω_m are first introduced as Eq. (109.30).

$$
\begin{aligned}
\Delta &= \frac{\partial u_x}{\partial x} + \frac{\partial u_z}{\partial z} \\
\omega &= \frac{1}{2}(\frac{\partial u_x}{\partial z} - \frac{\partial u_z}{\partial x})
\end{aligned}
\tag{109.30}
$$

ϕ_m and Ψ_m are related to Δ_m and ω_m as follows:

$$
\begin{aligned}
\phi_m &= -(\frac{\alpha_m}{w})^2 \Delta_m \\
\Psi_m &= 2(\frac{\beta_m}{w})^2 \omega_m
\end{aligned}
\tag{109.31}
$$

The displacements (u_x, u_y) and interfacial stresses (σ_{zz}, τ_{zx}) can be expressed in wave potential magnitudes ϕ and Ψ.

From Eqs. (109.28)-(109.31), the displacement and stress field of m^{th} layer can be calculated from the dilatational wave and rotational wave solutions Δ_m and ω_m as follows:

$$
\begin{aligned}
u_x = \{&-ik(\frac{\alpha_m}{\omega})^2[(\Delta_m^{'} + \Delta_m^{''})cos(k\gamma_{\alpha_m}z) - i(\Delta_m^{'} - \Delta_m^{''})sin(k\gamma_{\alpha_m}z)] \\
&+2ik\gamma_{\beta_m}(\frac{\beta_m}{\omega})^2[(\omega_m^{'} - \omega_m^{''})cos(k\gamma_{\beta_m}z) + i(\omega_m^{''} + \omega_m^{'})sin(k\gamma_{\beta_m}z)]\}e^{ikx}
\end{aligned}
\tag{109.32}
$$

$$
\begin{aligned}
u_z = \{&ik\gamma_{\alpha_m}(\frac{\alpha_m}{\omega})^2[(\Delta_m^{'} - \Delta_m^{''})cos(k\gamma_{\alpha_m}z) - i(\Delta_m^{''} + \Delta_m^{'})sin(k\gamma_{\alpha_m}z)] \\
&+2ik(\frac{\beta_m}{\omega})^2[(\omega_m^{'} + \omega_m^{''})cos(k\gamma_{\beta_m}z) - i(\omega_m^{'} - \omega_m^{'})sin(k\gamma_{\beta_m}z)]\}e^{ikx}
\end{aligned}
\tag{109.33}
$$

$$
\begin{aligned}
\sigma_{zz} = \rho_m\{&\alpha_m^2(1 - 2\frac{\beta_m^2}{c^2})[(\Delta_m^{'} + \Delta_m^{''})cos(k\gamma_{\alpha_m}z) - i(\Delta_m^{'} - \Delta_m^{''})sin(k\gamma_{\alpha_m}z)] \\
&+4\frac{\beta_m^4}{c^2}\gamma_{\beta_m}[(\omega_m^{'} - \omega_m^{''})cos(k\gamma_{\beta_m}z) - i(\omega_m^{'} + \omega_m^{''})sin(k\gamma_{\beta_m}z)]\}e^{ikx}
\end{aligned}
\tag{109.34}
$$

$$
\begin{aligned}
\tau_{zx} = 2\rho_m\beta_m^2\{&-\gamma_{\alpha_m}(\frac{\alpha_m}{c})^2[(\Delta_m^{'} - \Delta_m^{''})cos(k\gamma_{\alpha_m}z) - i(\Delta_m^{''} + \Delta_m^{'})sin(k\gamma_{\alpha_m}z)] \\
&+[1 - 2(\frac{\beta_m}{c})^2][(\omega_m^{'} + \omega_m^{''})cos(k\gamma_{\beta_m}z) - i(\omega_m^{'} - \omega_m^{''})sin(k\gamma_{\beta_m}z)]\}e^{ikx}
\end{aligned}
\tag{109.35}
$$

Define the displacement and stress solution at m^{th} interface as $S^{(m)}$, which is equal to $[\dot{u}_x(z_m = d_m)/c, \dot{u}_z(z_m = d_m)/c, \sigma_{zz}(z_m = d_m), \tau_{zx}(z_m = d_m)]^T$. Eqs. (109.32) - (??) can be reduced to the following matrix notations (Haskell, 1953):

$$S^{(m-1)} = \boldsymbol{E_m}[\Delta_m^{''} + \Delta_m^{'}, \Delta_m^{''} - \Delta_m^{'}, \omega_m^{''} - \omega_m^{'}, \omega_m^{''} + \omega_m^{'}]^T \tag{109.36}$$

$$S^{(m)} = \boldsymbol{D_m}[\Delta_m^{''} + \Delta_m^{'}, \Delta_m^{''} - \Delta_m^{'}, \omega_m^{''} - \omega_m^{'}, \omega_m^{''} + \omega_m^{'}]^T \tag{109.37}$$

where transformation matrix $\boldsymbol{E_m}$ and $\boldsymbol{D_m}$ are given in Appendix (Eqs. ?? and ??). The recurrence relation between $S^{(m)}$ and $S^{(m-1)}$ then can be established as Eq. 109.38, where $\boldsymbol{G_m} = \boldsymbol{D_m}\boldsymbol{E_m^{-1}}$.

$$S^{(m)} = \boldsymbol{D_m}\boldsymbol{E_m^{-1}}S^{(m-1)} = \boldsymbol{G_m}S^{(m-1)} \tag{109.38}$$

Recursively applying Eq. 109.38 leads to Eq. 109.39. Using the relation between $S^{(m-1)}$ and Δ_m, ω_m, Eq. 109.40 bridges the gap between the upper boundary (i.e., response at ground surface $S^{(0)}$) and lower boundary (i.e., solutions of wave incident layer Δ_n and ω_n), upon which specific boundary conditions can be imposed.

$$S^{(n-1)} = \prod_{i=1}^{n-1} \boldsymbol{G_i} S^{(0)} \tag{109.39}$$

$$\begin{gathered} S^{(0)} = \boldsymbol{L}[\Delta_n^{''} + \Delta_n^{'}, \Delta_n^{''} - \Delta_n^{'}, \omega_n^{''} - \omega_n^{'}, \omega_n^{''} + \omega_n^{'}]^T \\ \boldsymbol{L} = (\prod_{i=1}^{n-1} \boldsymbol{G_i})^{-1}\boldsymbol{E_n} \end{gathered} \tag{109.40}$$

Following boundary conditions are incorporated: (1) At n^{th} layer, the incident in-plane P and SV wave potential magnitude $\phi_n^{'}$ and $\Psi_n^{'}$ are given as K_1 and K_2; (2) At the ground surface ($z = 0$), the traction is free (i.e., the third and fourth component of $S^{(0)}$ is 0). Therefore, the reflected dilatational wave magnitude and rotational wave magnitude can be solved by Eq. 109.41, where $\Delta_n^{'}$ is $-K_1\omega^2/\alpha_n^2$ and $\omega_n^{'}$ is $K_2 w^2/(2\beta_n^2)$.

$$\begin{bmatrix} \Delta_n^{''} \\ \omega_n^{''} \end{bmatrix} = \begin{bmatrix} L_{31} + L_{32} & L_{33} + L_{34} \\ L_{41} + L_{42} & L_{43} + L_{44} \end{bmatrix}^{-1} \begin{bmatrix} (L_{32} - L_{31})\Delta_n^{'} + (L_{33} - L_{34})\omega_n^{'} \\ (L_{42} - L_{41})\Delta_n^{'} + (L_{43} - L_{44})\omega_n^{'} \end{bmatrix} \tag{109.41}$$

Finally, recurrence relation Eq. 109.42 can be used to trace back dilatational wave magnitude Δ_m and rotational wave magnitudes ω_m for the rest $n-1$ layers. Based on these solved dilatational and rotational magnitudes of each layer, the whole displacement and stress field can be easily computed following Eqs. (109.32)-(??).

$$\begin{bmatrix} \Delta''_{m-1} + \Delta'_{m-1} \\ \Delta''_{m-1} - \Delta'_{m-1} \\ \omega''_{m-1} - \omega'_{m-1} \\ \omega''_{m-1} + \omega'_{m-1} \end{bmatrix} = \boldsymbol{D^{-1}_{m-1} E_m} \begin{bmatrix} \Delta''_{m} + \Delta'_{m} \\ \Delta''_{m} - \Delta'_{m} \\ \omega''_{m} - \omega'_{m} \\ \omega''_{m} + \omega'_{m} \end{bmatrix} \tag{109.42}$$

In addition, viscosity can also be included with slight modification. Considering Kelvin-Voigt viscoelastic material (Chiriță et al., 2008), viscosity can be handled with complex Lame modulus and wave velocities as shown in Eq. 109.43, where ξ is the damping ratio.

$$G^* = G(1+2\xi i) \quad \beta^*_m \simeq \beta_m(1+\xi i) \quad \alpha^*_m \simeq \alpha_m(1+\xi i) \tag{109.43}$$

109.2.6 Seismic Motions: Large Scale Geophysical Models

109.2.6.1 Regional Seismic Motion Modeling using Serpentine Wave Propagation, SW4

109.2.7 Site Response

Site response is ...

This is part of free field motions section and just uses free field motions for producing site response.

109.2.8 Seismic Motion Incoherence

Seismic motion incoherence (as it is called for frequency domain analysis, for time domain analysis it is called lack of correlation) is a phenomena that results in spatial variability of ground motions over small distances. Significant work has been done in researching seismic motion incoherence over the last few decades. The main sources of lack of spatial correlation, according to Zerva (2009) are due to:

- Attenuation effects,
- Wave passage effects,
- Scattering effects,
- Extended source effects

Figure 109.4 shows an illustration of main sources of lack of correlation.

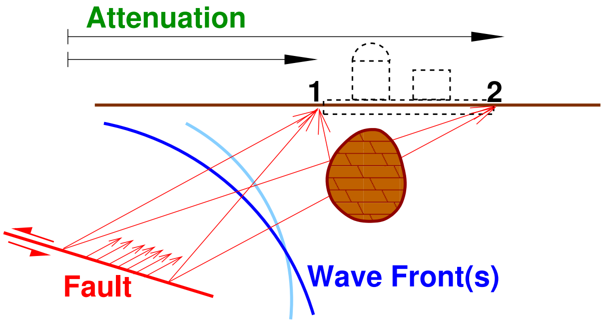


Figure 109.4: Four main sources contributing to the lack of correlation of seismic waves as measured at two observation points.

1. Attenuation effects are responsible for change in amplitude and phase of seismic motions due to the distance between observation points and losses (damping, energy dissipation) that seismic wave experiences along that distance. This is a significant source of lack of correlation for long structures (bridges), however for NPPSSS it is not of much significance.
2. Wave passage effects contribute to lack of correlation due to difference in recorded wave field at two location points as the (surface) wave travels, propagates from first to second point.

3. Scattering effects are responsible to lack of correlation by creating a scattered wave field. Scattering is due to (unknown or not known enough) subsurface geologic features that contribute to modification of wave field.

4. Extended source effects contribute to lack of correlation by creating a complex wave source field, as the fault ruptures, rupture propagates and generate seismic sources along the fault. Seismic energy is thus emitted from different points (along the rupturing fault) and will have different travel path and timing as it makes it observation points.

Early studies concluded that the correlation of motions increases as the separation distance between observation points decreases. In addition to that, correlation increased for decrease in frequency of observed motions. Moreover, there is a strong probabilistic nature of this phenomena, as significant uncertainty is present in relation to all four sources of lack of correlation, mentioned above. A number of excellent references are available on the subject of incoherent (or lacking correlation) seismic motions Abrahamson et al. (1991); Roblee et al. (1996); Abrahamson (1992a, 2005, 1992b); Zerva and Zervas (2002); Liao and Zerva (2006); Zerva (2009)

It is very important to note that all current models for modeling incoherent seismic motions make an ergodic assumption. That is all the models assume that a variability of seismic motions at a single site – source combination will be the same as variability in the ground motions from a data set that was collected over different site and source locations Walling (2009). Unfortunately, there does not exist a large enough data set for east North American seismic seismic events that can be used to develop incoherence models. Rather, there are models that are used to model possible incoherent behavior for east North American seismic wave propagations.

109.2.8.1 Lack of Correlation Modeling and Simulation

Incorporation of lack of correlation effects in seismic motions can be done using the following methods:

- along 1D, in one direction, usually one of the horizontal directions, where all the points that are the same distance (in 1D) from a control point, plane) share a single lack of correlation,

- along 2D, in two directions, usually in a vertical plane or in a horizontal plane, where all the points in a set of vertical planes or a set of horizontal planes, share same lack of correlation, and

- in full 3C, where every point of interest (in a 3D volume of soil/rock) has it own, specified, lack of correlation.

The method used here is using the so called seed motions, motions obtained through DRM in full 3C (inclined, body and surface waves), that are than enriched with appropriate uncorrelated (incoherent) components. It is important to note that only translational motions are used to model incoherence effects, while the rotational motions are not perturbed/made incoherent. This stems from the fact that currently vast majority of seismic recording stations only record translational motions and that only those translational motions are used to develop incoherent motions models. Code developed by Abrahamson (1992b) is used for this purpose.

109.2.9 Lack of Volume Change Data for Soil

Use of G/G_{max} and damping curves for describing and calibrating material behavior of soil is missing a very important (crucial) information about soil/rock volume change during shearing deformation. Volume change data is very important for soil behavior. It is important to emphasize that soil behavior is very much a function of volumetric response during shear. During shearing of soil there are two essential types of soil behavior:

- Dilative (dense) soils will increase volume due to shearing
- Compressive (loose) soils will decrease volume due to shearing

The soil volume response, that is not provided by G/G_{max} and damping curves data can significantly affect affect response due to volume constraints of soil. For example, for one dimensional site response (1C wave propagation, vertically propagating (SV) shear waves) the soil will try to change its volume (dilate if it is dense or compress if it is loose). However, such volume change can only happen vertically (since there is no constraint (foundation for example) on top, while horizontally the soil will be constraint by other soil. That means that any intended volume change in horizontal direction will be resisted by change in (increase for dense and decrease for loose soil) horizontal stress. For example for dilative (dense) soil, additional horizontal stress will contribute to the increase in mean pressure (confinement) of the soil, thus increasing the stiffness of that soil. It is the opposite for compressive soil where shearing will result in a reduction of confinement stress. Figure 109.5 shows three responses for no-volume change (left), compressive (middle) and dilative (right) soil with full volume constraint, resulting in changes in stiffness for compressive (reduction in stiffness), and dilative (increase in stiffness).

Changes in stiffness of soil during shearing deformation will influence wave propagation and amplification of different frequencies. Figure 109.6 shows response of no-volume change soil (as it is/should be assumed if only G/G_{max} and damping curves are available, with no volume change data) and a response of a dilative soil which stiffens up during shaking due to restricted intent to dilate. It is clear that dilative soil will show significant amplification of higher frequencies.

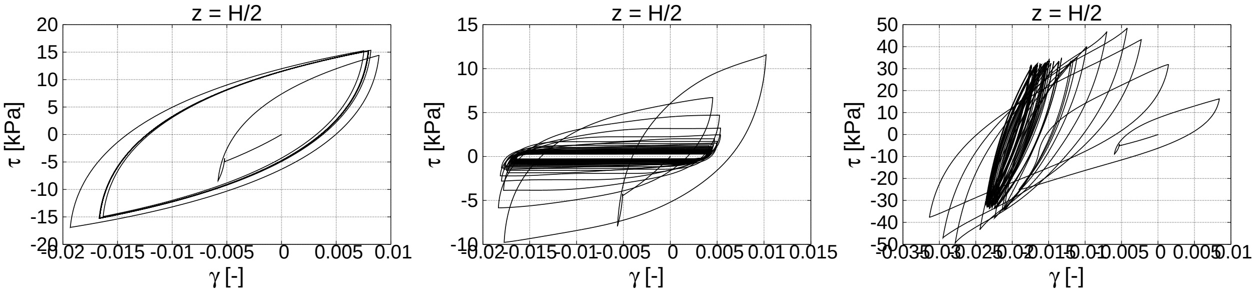


Figure 109.5: Constitutive Cyclic response of soils with constraint volumetric deformation: (left) no volume change (soil is at the critical state); (middle) compressive response with decrease in stiffness; (right) dilative response with increase in stiffness.

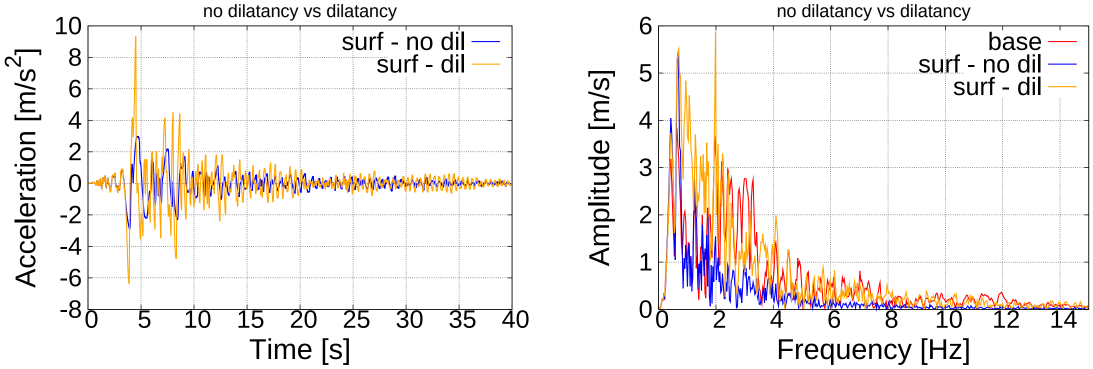


Figure 109.6: One dimensional seismic wave propagation through no-volume change and dilative soil. Please note the (significant) increase in frequency of motions for dilative soil. Left plot is a time history of motions, while the right plot shows amplitudes at different frequencies.

109.3 Earthquake Soil Structure Interaction

Current design practice for structures subject to earthquake loading regards dynamic SSII to be mainly beneficial to the behavior of structures (Jeremić and Preisig, 2005). Including the flexibility of the foundation reduces the overall stiffness of a system and therefore reduces peak loads caused by a given ground motion. Even if this is true in most cases there is the possibility of resonance occurring as a result of a shift of the natural frequencies of the SSI-system. This can lead to large inertial forces acting on a structure.

As a result of these large inertial forces caused by the structure oscillating in it's natural frequency the structure as well as the soil surrounding the foundation can undergo plastic deformations. This in turn further modifies the overall stiffness of the SSI-system and makes any prediction on the behavior very difficult.

Dynamic SSII also becomes important in the design of large infrastructure projects. As authorities and insurance companies try to introduce the concept of performance based design to the engineering community more sophisticated models are needed in order to obtain the engineering demand parameters (EDP's). A good numerical model of a soil-foundation-structure system can therefore not only prevent the collapse or damage of a structure but also help to save money by optimizing the design to withstand an earthquake with a certain return period.

A variety of methods of different levels of complexity are currently being used by engineers. In the following an overview over the most important ones is presented. A more thorough discussion on methods and specific aspects of dynamic SSII is available in Wolf (1985) and more recently in Wolf and Song (2002).

- No SSII

  The ground motion is applied directly to the base of the building. Alternatively, instead of applying the ground motion directly to the base of the structure, effective earthquake forces proportional to the base acceleration can be applied to the nodes.

  This procedure is reasonable only for flexible structures on very stiff soil or rock. In this case the displacement of the ground doesn't get modified by the presence of the structure. For stiffer structures on soil the ground motion has to be applied to the soil. The model has to incorporate propagation of the motion through the soil, its interaction with the structure and the radiation away from the structure.

- Direct methods

  Direct methods treat the SSI-system as a whole. The numerical model incorporates the spatial discretization of the structure and the soil. The analysis of the entire system is carried out in one step. This method provides most generality as it is capable of incorporating all nonlinear behavior of the structure, the soil and also the interface between those two (sliding, uplift).

- Substructure methods

  Substructure methods refer to the principle of superposition. The SSI-system is generally subdivided into a structure part and a soil part. Both substructures can be analyzed separately and the total displacement can be obtained by adding the contributions at the nodes on the interface.

  This method reduces the size of the problem considerably. As the time needed for an analysis doesn't increase linearly with an increasing number of equations the substructure method is much faster than the direct method. The biggest drawback of the method however is the fact that linearity is assumed. For nonlinear systems the substructure method cannot be used.

For the direct method different levels of sophistication are possible:

- Foundation stiffness approach

  The behavior of the soil is accounted for by simple mechanical elements such as springs, masses and dash pots. Different configurations of the subsoil can be taken into account by connecting several springs, masses and dash pots whose parameters have been determined by a curve fitting procedure Wolf (1994). This approach is very popular among structural engineers as it is relatively easy to be integrated in a commonly used finite element code.

  Other methods use frequency dependent springs and dash pots and therefore require an analysis in frequency domain. Relatively complex configurations of layered subsoil and embedded foundations can be modeled with good accuracy by replacing the (elastic) soil with a sequence of conical rods Wolf and Song (2002) and Wolf and Preisig (2003).

- p-y methods

  Attempts have been made to apply the static p-y approach for evaluating lateral loading on pile foundations to dynamic problems. Mostafa and El Naggar (2002) lists several references and provides a parametric study of single piles and pile groups in different soil types under simplified loading cases.

  Even if p-y curves are widely used for estimating lateral loading on piles they are rarely used in full dynamic soil-structure interaction analysis. Current work trying to implement these methods into finite element codes is likely to make them more popular with the engineering community.

- Full 3d

  Full nonlinear three-dimensional modeling of dynamic soil-foundation-structure interaction can be regarded as the 'brute force' approach. Displacements and forces can be obtained not only for the structure as in the above mentioned methods but also for the soil. In spite of the computational resources and modeling effort required for an analysis it is the only method that remains valid for all kinds of problems involving material nonlinearities, contact/interface problems, different loading cases and complex geometries.

109.4 Earthquake Soil Structure Interaction Modeling Details

109.4.1 Seismic Motions Input into Finite Element Model

A number of methods is used to input seismic motions into finite element model. Most of them are based on simple intuitive approaches, and as such are not based on rational mechanics. Most of those currently still widely used methods cannot properly model all three components of body waves as well as always present surface waves. There exist a method that is based on rational mechanics and can model both body and surface seismic waves input into finite element models with high accuracy. That method is called the Domain Reduction Method (DRM) (Bielak et al., 2003a; Yoshimura et al., 2003a)). The DRM aims to reduce the large computational domain, encompassing fault, rock, soil and the structure, to a much smaller domain, encompassing only local soil and the structure. The method was developed with earthquake ground motions in mind, with the main idea to replace the force couples at the fault with their counterpart acting on a continuous surface surrounding local feature of interest. The local feature can be any geologic or man made object that constitutes a difference from the simplified large domain, free field, for which displacements and accelerations are easier to obtain.

The DRM is applicable to a much wider range of problems. It is essentially a variant of global–local set of methods and as formulated can be used for any problems where the local feature can be bounded by a continuous surface (that can be closed or not). The local feature in general can represent a soil–foundation–structure system (bridge, building, dam, tunnel...), or it can be a crack in large domain, or some other type of inhomogeneity that is fairly small compared to the size of domain where it is found.

In what follow, the DRM is developed in a somewhat different way than it was done in original papers by Bielak et al. (2003a); Yoshimura et al. (2003a)). The main features of the DRM are then analyzed and appropriate practical modeling issues addressed.

109.4.1.1 The Domain Reduction Method (DRM) Development

A large physical domain is to be analyzed for dynamic behavior. The source of disturbance is a known time history of a force field $P_e(t)$. That source of loading is far away from a local feature which is dynamically excited by $P_e(t)$ $P_e(t)$ (see Figure 109.7).

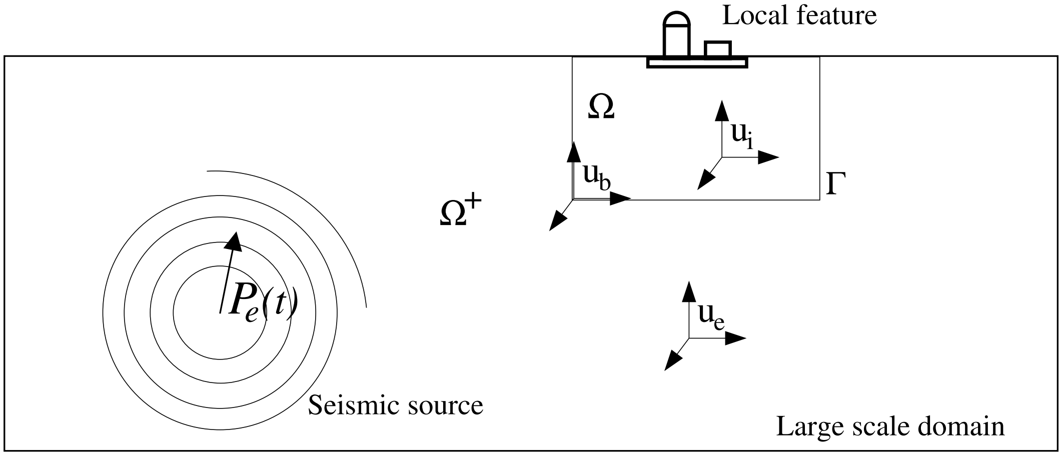


Figure 109.7: Large physical domain with the source of load $P_e(t)$ and the local feature (in this case a soil-structure system.

The system to be analyzed can be quite large, for example earthquake hypocenter can be many kilometers away from the local feature of interest. Similarly, the small local feature in a machine part can be many centimeters away from the source of dynamic loading which influences this local feature. In this sense the term large domain is relative to the size of the local feature and the distance to the dynamic forcing source.

It would be beneficial not to analyze the complete system, as we are only interested in the behavior of the local feature and its immediate surrounding, and can almost neglect the domain outside of some relatively close boundaries. In order to do this, we need to somehow transfer the loading from the source to the immediate vicinity of the local feature. For example we can try to reduce the size of the domain to a much smaller model bounded by surface Γ as shown in Figure 109.7. In doing so we must ensure that the dynamic forces $P_e(t)$ are appropriately propagated to the much smaller model boundaries Γ.

DRM Formulation In order to appropriately propagate dynamic forces $P_e(t)$ one actually has to solve the large scale problem which will include the effects of the local feature. Most of the time this is impossible as it involves all the complexities of large scale computations and relatively small local feature. Besides, the main goal of presented developments is to somehow reduce the large scale domain as to be able to analyze in details behavior of the local feature.

In order to propagate consistently the dynamic forces $P_e(t)$ we will make a simplification in that we will replace a local feature with a simpler domain that is much easer to be analyzed. That is, we replace the local feature (bridge, building. tunnel, crack) with a much simpler geometry and material. For example, Figure 109.8 shows a simplified model, without a foundation–building system.

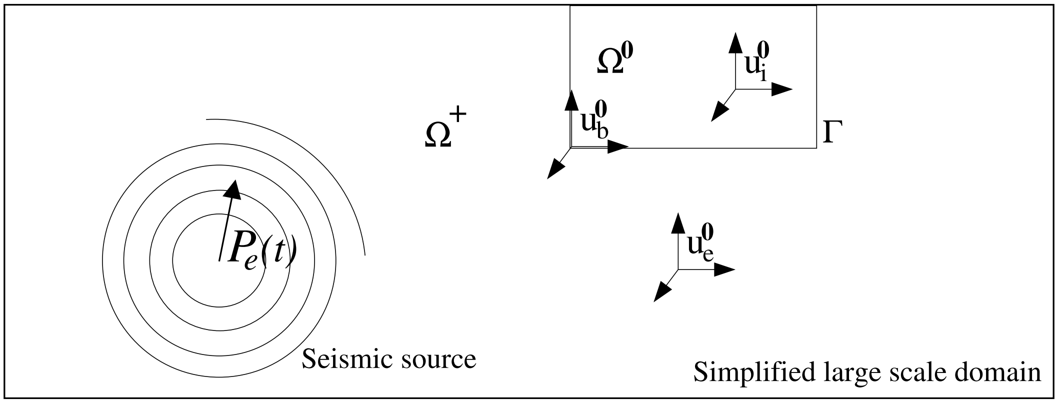


Figure 109.8: Simplified large physical domain with the source of load $P_e(t)$ and without the local feature (in this case a soil–foundation–building system. Instead of the local feature, the model is simplified so that it is possible to analyze it and simulate the dynamic response as to consistently propagate the dynamic forces $P_e(t)$

The idea is to simplify the model so that it is much easier to consistently propagate the dynamic forces to the boundary Γ. The notion that it is much easier to propagate those dynamic forces is of course relative. This is still a very complex problem, but at least the influence of local feature is temporarily taken out.

It is convenient to name different parts of domain. For example, the domain inside the boundary Γ is named Ω_0. The rest of the large scale domain, outside boundary Γ, is then named Ω^+. The outside domain Ω^+ is still the same as in the original model, while the change, simplification, is done on the domain inside boundary Γ. The displacement fields for exterior, boundary and interior of the boundary Γ are u_e, u_b and u_i, respectively, on the original domain.

The equations of motions for the complete system can be written as

$$\left[M \right] \left\{ \ddot{u} \right\} + \left[K \right] \left\{ u \right\} = \left\{ P_e \right\} \tag{109.44}$$

or, if written for each domain (interior, boundary and exterior of Γ) separately, the equations obtain the following form:

$$\begin{bmatrix} M_{ii}^{\Omega} & M_{ib}^{\Omega} & 0 \\ M_{bi}^{\Omega} & M_{bb}^{\Omega}+M_{bb}^{\Omega+} & M_{be}^{\Omega+} \\ 0 & M_{eb}^{\Omega+} & M_{ee}^{\Omega+} \end{bmatrix} \begin{Bmatrix} \ddot{u}_i \\ \ddot{u}_b \\ \ddot{u}_e \end{Bmatrix} + \begin{bmatrix} K_{ii}^{\Omega} & K_{ib}^{\Omega} & 0 \\ K_{bi}^{\Omega} & K_{bb}^{\Omega}+K_{bb}^{\Omega+} & K_{be}^{\Omega+} \\ 0 & K_{eb}^{\Omega+} & K_{ee}^{\Omega+} \end{bmatrix} \begin{Bmatrix} u_i \\ u_b \\ u_e \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ P_e \end{Bmatrix} \tag{109.45}$$

In these equations, the matrices $\mathbf{M}$ and $\mathbf{K}$ are mass and stiffness matrices respectively; the subscripts i, e, and b are referencing nodes in either the interior (i) or exterior (e) domain or on their common boundary (b), while the superscripts Ω and Ω^+ reference domains to which matrices belong.

The previous equation can be separated provided that we maintain the compatibility of displacements and equilibrium. The resulting two equations of motion are

$$\begin{bmatrix} M_{ii}^{\Omega} & M_{ib}^{\Omega} \\ M_{bi}^{\Omega} & M_{bb}^{\Omega} \end{bmatrix} \begin{Bmatrix} \ddot{u}_i \\ \ddot{u}_b \end{Bmatrix} + \begin{bmatrix} K_{ii}^{\Omega} & K_{ib}^{\Omega} \\ K_{bi}^{\Omega} & K_{bb}^{\Omega} \end{bmatrix} \begin{Bmatrix} u_i \\ u_b \end{Bmatrix} = \begin{Bmatrix} 0 \\ P_b \end{Bmatrix}, \quad \text{in}\Omega \tag{109.46}$$

and

$$\begin{bmatrix} M_{bb}^{\Omega+} & M_{be}^{\Omega+} \\ M_{eb}^{\Omega+} & M_{ee}^{\Omega+} \end{bmatrix} \begin{Bmatrix} \ddot{u}_b \\ \ddot{u}_e \end{Bmatrix} + \begin{bmatrix} K_{bb}^{\Omega+} & K_{be}^{\Omega+} \\ K_{eb}^{\Omega+} & K_{ee}^{\Omega+} \end{bmatrix} \begin{Bmatrix} u_b \\ u_e \end{Bmatrix} = \begin{Bmatrix} -P_b \\ P_e \end{Bmatrix}, \quad \text{in}\Omega^+ \tag{109.47}$$

Compatibility of displacements is maintained automatically since both equations contain boundary displacements u_b (on boundary Γ), while the equilibrium is maintained through action–reaction forces P_b.

In order to simplify the problem, the local feature is removed from the interior domain. Thus, the interior domain is significantly simplified. In other words, the exterior region and the material therein are identical to those of the original problem as the dynamic force source. On the other hand, the interior domain (denoted as Ω_0), is simplified, the localized features is removed (as seen in figure 109.8).

For this simplified model, the displacement field (interior, boundary and exterior, respectively) and action–reaction forces are denoted by u_i^0, u_b^0, u_e^0 and P_b^0. The entire simplified domain Ω_0 and Ω^+ is now easier to analyze.

The equations of motion in Ω^+ for the auxiliary problem can now be written as:

$$\begin{bmatrix} M_{bb}^{\Omega+} & M_{be}^{\Omega+} \\ M_{eb}^{\Omega+} & M_{ee}^{\Omega+} \end{bmatrix} \begin{Bmatrix} \ddot{u}_b^0 \\ \ddot{u}_e^0 \end{Bmatrix} + \begin{bmatrix} K_{bb}^{\Omega+} & K_{be}^{\Omega+} \\ K_{eb}^{\Omega+} & K_{ee}^{\Omega+} \end{bmatrix} \begin{Bmatrix} u_b^0 \\ u_e^0 \end{Bmatrix} = \begin{Bmatrix} -P_b^0 \\ P_e \end{Bmatrix} \tag{109.48}$$

Since there was no change to the exterior domain Ω^+ (material, geometry and the dynamic source are still the same) the mass and stiffness matrices and the nodal force P_e are the same as in Equations (109.46) and (109.47).

Previous equation 109.48 can be used to obtain the dynamic force P_e:

$$P_e = M_{eb}^{\Omega+}\ddot{u}_b^0 + M_{ee}^{\Omega+}\ddot{u}_e^0 + K_{eb}^{\Omega+}u_b^0 + K_{ee}^{\Omega+}u_e^0 \tag{109.49}$$

The total displacement, u_e, can be expressed as the sum of the free field u_e^0 (from the background, simplified model) and the residual field w_e (coming from the local feature) as following:

$$u_e = u_e^0 + w_e \tag{109.50}$$

It is important to note that this is just a change of variables and not an application of the principle of superposition. The residual displacement field, w_e is measured relative to the reference free field u_e^0.

By substituting Equation 109.50 in Equation 109.45 one obtains:

$$\begin{bmatrix} M_{ii}^{\Omega} & M_{ib}^{\Omega} & 0 \\ M_{bi}^{\Omega} & M_{bb}^{\Omega} + M_{bb}^{\Omega+} & M_{be}^{\Omega+} \\ 0 & M_{eb}^{\Omega+} & M_{ee}^{\Omega+} \end{bmatrix} \begin{Bmatrix} \ddot{u}_i \\ \ddot{u}_b \\ \ddot{u}_e^0 + \ddot{w}_e \end{Bmatrix} + \begin{bmatrix} K_{ii}^{\Omega} & K_{ib}^{\Omega} & 0 \\ K_{bi}^{\Omega} & K_{bb}^{\Omega} + K_{bb}^{\Omega+} & K_{be}^{\Omega+} \\ 0 & K_{eb}^{\Omega+} & K_{ee}^{\Omega+} \end{bmatrix} \begin{Bmatrix} u_i \\ u_b \\ u_e^0 + w_e \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ P_e \end{Bmatrix} \tag{109.51}$$

which, after moving the free field motions u_e^0 to the right hand side, becomes

$$\begin{bmatrix} M_{ii}^{\Omega} & M_{ib}^{\Omega} & 0 \\ M_{bi}^{\Omega} & M_{bb}^{\Omega} + M_{bb}^{\Omega+} & M_{be}^{\Omega+} \\ 0 & M_{eb}^{\Omega+} & M_{ee}^{\Omega+} \end{bmatrix} \begin{Bmatrix} \ddot{u}_i \\ \ddot{u}_b \\ \ddot{w}_e \end{Bmatrix} + \begin{bmatrix} K_{ii}^{\Omega} & K_{ib}^{\Omega} & 0 \\ K_{bi}^{\Omega} & K_{bb}^{\Omega} + K_{bb}^{\Omega+} & K_{be}^{\Omega+} \\ 0 & K_{eb}^{\Omega+} & K_{ee}^{\Omega+} \end{bmatrix} \begin{Bmatrix} u_i \\ u_b \\ w_e \end{Bmatrix} = \begin{Bmatrix} 0 \\ -M_{be}^{\Omega+}\ddot{u}_e^0 - K_{be}^{\Omega+}u_e^0 \\ -M_{ee}^{\Omega+}\ddot{u}_e^0 - K_{ee}^{\Omega+}u_e^0 + P_e \end{Bmatrix} \tag{109.52}$$

By substituting Equation 109.49 in previous Equation 109.52, the right hand side can now be written as

$$\begin{bmatrix} M_{ii}^{\Omega} & M_{ib}^{\Omega} & 0 \\ M_{bi}^{\Omega} & M_{bb}^{\Omega} + M_{bb}^{\Omega+} & M_{be}^{\Omega+} \\ 0 & M_{eb}^{\Omega+} & M_{ee}^{\Omega+} \end{bmatrix} \begin{Bmatrix} \ddot{u}_i \\ \ddot{u}_b \\ \ddot{w}_e \end{Bmatrix} + \begin{bmatrix} K_{ii}^{\Omega} & K_{ib}^{\Omega} & 0 \\ K_{bi}^{\Omega} & K_{bb}^{\Omega} + K_{bb}^{\Omega+} & K_{be}^{\Omega+} \\ 0 & K_{eb}^{\Omega+} & K_{ee}^{\Omega+} \end{bmatrix} \begin{Bmatrix} u_i \\ u_b \\ w_e \end{Bmatrix} = \begin{Bmatrix} 0 \\ -M_{be}^{\Omega+}\ddot{u}_e^0 - K_{be}^{\Omega+}u_e^0 \\ M_{eb}^{\Omega+}\ddot{u}_b^0 + K_{eb}^{\Omega+}u_b^0 \end{Bmatrix} \tag{109.53}$$

The right hand side of equation 109.53 is a dynamically consistent replacement force, the so called effective force, P^{eff} for the dynamic source forces P_e. In other words, the dynamic force P_e was consistently replaced by the effective force P^{eff}:

$$P^{eff} = \left\{ \begin{array}{c} P_i^{eff} \\ P_b^{eff} \\ P_e^{eff} \end{array} \right\} = \left\{ \begin{array}{c} 0 \\ -M_{be}^{\Omega+}\ddot{u}_e^0 - K_{be}^{\Omega+}u_e^0 \\ M_{eb}^{\Omega+}\ddot{u}_b^0 + K_{eb}^{\Omega+}u_b^0 \end{array} \right\} \tag{109.54}$$

DRM Discussion

Single Layer of Elements used for P^{eff}. The Equation (109.54) shows that the effective nodal forces P^{eff} involve only the sub-matrices M_{be}, K_{be}, M_{eb}, K_{eb}. These matrices vanish everywhere except the single layer of finite elements in domain Ω^+ adjacent to Γ. The significance of this is that the only wave-field (displacements and accelerations) needed to determine effective forces P^{eff} is that obtained from the simplified (auxiliary) problem at the nodes that lie on and between boundaries Γ and Γ_e, as shown in Figure 109.9.

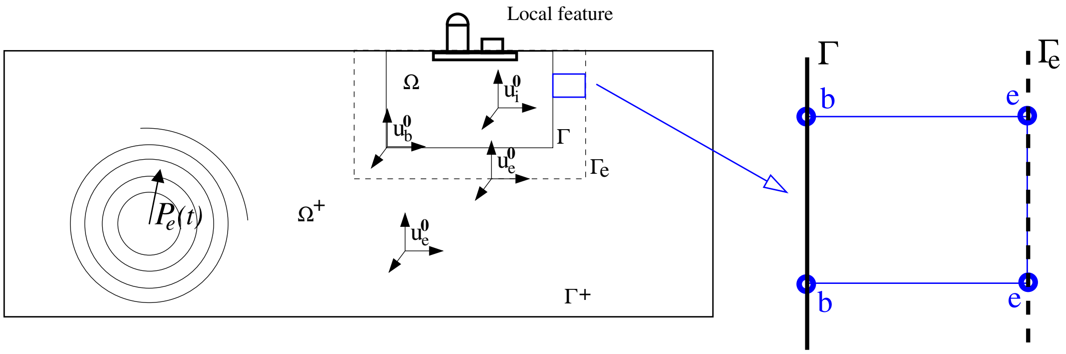


Figure 109.9: DRM: Single layer of elements between Γ and Γ_e is used to create P^{eff}, for a section of 8 node brick.

Figure 109.10 show boundary and external nodes for a section of a 27 node brick. Please note that for a 20, 20-17 and 27 node bricks, boundary nodes are nodes that belong to Γ surface, so for 27 node brick there are 9 nodes on that face, while the external nodes are all the nodes that are not boundary nodes, there will be 18 of those nodes.

Only residual waves outgoing. Another interesting observation is that the solution to problem described in Equation (109.53) comprises full unknowns (displacements and accelerations) inside and on the boundary Γ ($\mathbf{u}_i$ and $\mathbf{u}_b$ respectively). On the other hand, the solution for the domain outside single layer of finite elements (outside Γ_e) is obtained for the residual unknown (displacement and accelerations) field,

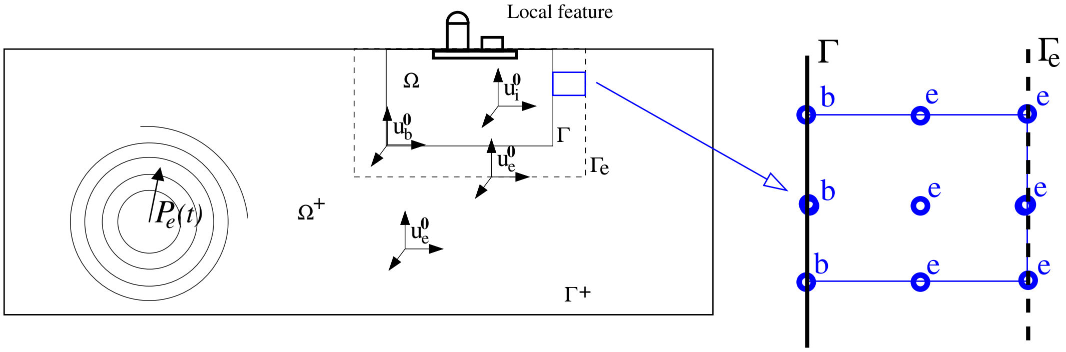


Figure 109.10: DRM: Single layer of elements between Γ and Γ_e is used to create P^{eff}, for 20, 20-17 and 27 node Brick.

w_e only. This residual unknown field is measured relative to the reference free field of unknowns (see comments on page 572). That effectively means that the solution to the equation Equation (109.53) outside the boundary Γ_e will only contain additional waves field resulting from the presence of a local feature. This in turn means that if the interest is in behavior of local feature and the surrounding media (all within boundary Γ) one can neglect the behavior of the full model (outside Γ_e in Ω^+) and provide appropriate supports (including fixity and damping) at some distance from the boundary Γ_e into region Ω^+. This is significant for a number of reasons:

- large models can be reduced in size to encompass just a few layers of elements outside boundary Γ_e (significant reduction for, say earthquake problems where the size of a local feature is orders of magnitudes smaller then the distance to the dynamic source force P_e (earthquake hypocenter).
- the residual unknown field can be monitored and analyzed for information about the dynamic characteristics of the local feature. Since the residual wave field is w_e is measured relative to the reference free field u_0^e, the solution for w_e has all the characteristics of the additional wave field stemming from the local feature.

Inside domain Ω can be inelastic. In all the derivations in section 109.4.1.1 no restriction was made on the type of material inside the plastic bowl (inside Γ_e). That is, the assumption that the material inside is linear elastic is not necessary as the DRM is not relying on principle of superposition. The Equation 109.50 was only describing the change of variables, and clearly there was no use of the principle of superposition, which is only valid for linear elastic solids and structures. It is therefore possible to assume that the derivations will still be valid with any type of material (linear or nonlinear, elastic or inelastic) inside Γ_e. With this in mind, the DRM becomes a very powerful method for analysis of soil–foundation–structure systems.

All types of realistic seismic waves are modeled. Since the effective forcing P^{eff} consistently replaces the effects of the seismic source, all appropriate (real) seismic waves are properly (analytically) modeled, including body (SV, SH, P) and surface (Rayleigh, Love, etc...) waves.

Properties of finite elements inside the DRM Layer. The DRM layers, a single layer of finite elements just outside Γ surfaces, where effective DRM forces P_{eff} are applied, needs to be carefully modeled. A number of conditions regarding the DRM layer need to be taken into account:

- The finite elements within the DRM layer need to be linear elastic.
- Material models for the finite elements within the DRM layer need to have same, or very similar material properties as the elastic part of material properties as the material inside the DRM layer. Although material inside the DRM layer can be elastic-plastic, it is beneficial if the linear elastic portion of material properties, for example for nonlinear elastic material at zero strain or for elastic-plastic material, elastic properties inside yield surface, for the DRM layer, is same, similar to the material used inside the DRM layer. All of the used elastic material properties need to be same as elastic material properties used for free field analysis in order to have consistent wave field.
- Dimensions of the DRM layer finite elements (thickness of the DRM layer) need to follow the same rule for element size (depending on chosen stiffness) so that there is no artificial (mesh dependent) filtering above certain frequencies. That means that 10 linear interpolation finite elements (8 node bricks) or 2 quadratic interpolation elements (27 node bricks) are needed per wave length (Bathe and Wilson, 1976; Hughes, 1987; Argyris and Mlejnek, 1991). For example if maximum modeling frequency is f_{max} = 20 Hz, and wave length is given as $\lambda_{min} = v/f_{max}$, where v is the wave velocity, maximum grid spacing (element size) for linear interpolation elements Δh^{LE} should not exceed

  $$\Delta h^{LE} \leq \frac{\lambda}{10} = \frac{v}{10 f_{max}} = \frac{v}{10 \times 20 \text{ Hz}} = \frac{v}{200 \text{ Hz}}$$

  while for quadratic interpolation elements such grid spacing (element size) Δh^{QE} is limited to

  $$\Delta h^{QE} \leq \frac{\lambda}{2} = \frac{v}{2 f_{max}} = \frac{v}{2 \times 20 \text{ Hz}} = \frac{v}{40 \text{ Hz}}$$

  The wave velocity v is the lowest wave velocity that is of interest in the simulation, usually the shear wave velocity.

Properties of the finite elements outside of the DRM Layer. Finite elements outside DRM layer need special considerations as well.

- Material outside the DRM layer needs to be linear elastic with addition of viscous damping, Caughey, Rayleigh, etc.
- At least one layer of finite elements outside DRM layer needs to be provided. If damping of outgoing waves, the so called radiation damping, is to be modeled, then more than one layer of finite elements outside DRM needs to be provided.
- If radiation damping is modeled, it is recommended to have more than just two layers, outside DRM layer. For example 4 or 5 additional layers work quite well.
- First layer outside DRM layer needs to be linear elastic, of similar/same properties as material inside DRM layer, and with NO viscous damping. The reason for this requirement, is explained by the fact that P^{eff} force, see equation 109.54 on page 573, is applied to finite elements within DRM layer. If finite elements that are outside/adjacent to the DRM layer have large viscous damping, then P^{eff} forces will be producing potentially significant reaction forces from large viscous damping that is placed on nodes of elements that are shared with DRM finite element nodes, and nodes of finite elements just outside DRM layer, and are connected to DRM finite elements. These reaction, viscous forces, will affects, change P^{eff} forces in a way that will not be consistent with seismic wave field that was used to develop P^{eff}.
- Viscous damping, Caughey, Rayleigh, should be placed on finite elements outside this first layer of elements, that is outside, adjacent to DRM layer, in order to damp out outgoing waves, the differential wave field, the "w_e" waves, see equation 109.50 on page 572. Additional viscous damping layers are added to damp out any additional waves, the so called radiation damping from structural oscilations. Damping in those additional layers is to be progressively larger, much larger than physical viscous damping. Values of equivalent damping of 20%, then 30%, then 50%, or higher have been used in order to damp outgoing waves, to model radiation damping.

A Note on Input Motions for DRM. Seismic motions (free field) that are used for input into a DRM model need to be consistent. In other words, a free field seismic wave that is used needs to fully satisfy equations of motion. For example, if free field motions are developed using a tool (SHAKE, or EDT or SW4, or fk, &c.) using time step $\Delta t = 0.01$s and then you decide that you want to run your analysis with a time step of $\Delta t = 0.001$s, simple interpolation (10 additional steps for each of the original steps) might create problems. Simple linear interpolation actually might (will) not satisfy wave propagation equations and if used will introduce additional, high frequency motions into the model. It is a very good idea to generate free field motions with the same time step as it will be used in ESSI simulation.

Similar problem might occur if spacial interpolation is done, that is if location of free field model nodes is not very close to the actual DRM nodes used in ESSI model. Spatial interpolation problems are actually a bit less acute, however one still has to pay attention and test the ESSI model for free conditions and only then add the structure(s) on top.

Input motions for the DRM are based on Free Field motions, that can be developed by a number of methods, as described in section 502.2.3 on page 2266.

DRM in Action, 1C vs 6C Free Field Motions

- One component of motions, 1C from 6C
- Excellent fit, however wrong physics

Figure 109.11: 6C real and 1C "horizontal motions fit" seismic motions. (Figure is a link to mp4 animation).

DRM in Action, 6C Free Field Motions, Variation in Input Frequency, Inclinded wave at $\theta = 60^o$.

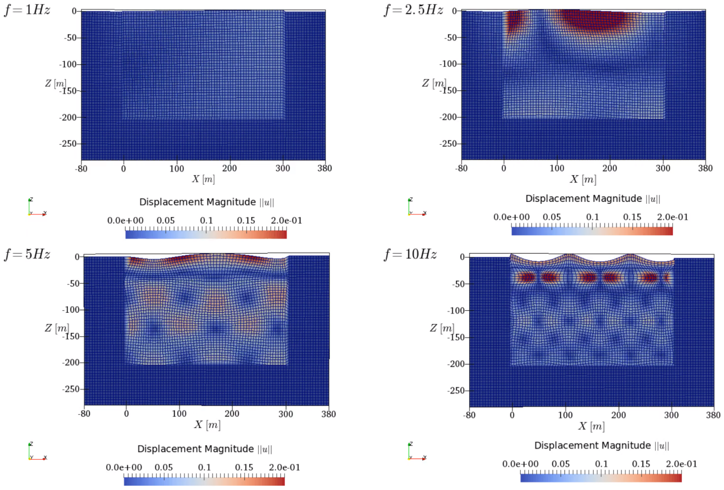


Figure 109.12: 6C real motions, variation in input frequency, inclined wave at $\theta = 60^o$.

Chapter 110

Parallel Computing in Computational Mechanics

(1998-2000-2005-2015-2016-2017-2018-2019-2021-)

(In collaboration with Dr. Guanzhou Jie and Dr. Yuan Feng, and Prof. Han Yang)

110.1 Chapter Summary and Highlights

110.2 Introduction

110.2.1 High Performance Computing on DMPs, SMPs, GPGPUs, FPGA

110.2.1.1 Distributed Memory Parallel (SMP) Computations

110.2.1.2 Shared Memory Parallel (SMP) Computations

110.2.1.3 General Purpose Graphical Processiong Units (GPGPUs)

110.2.1.4 Fast Programmable Gate Arrays (FPGAs)

110.2.2 Parallel Computing for Elastic-Plastic Solids and Structures

110.2.3 Problem Requirements

Stages, Increments, Iterations

For many classes of scientific simulations, an initial (static) decomposition of a finite element mesh needs to be computed such that the number of mesh elements assigned to each processor is roughly equal and the number of mesh elements that are adjacent to elements assigned to other processors (i.e., the size of the subdomain boundary) is minimized. Ensuring that the number of mesh elements is balanced will result in a load-balanced computation, while minimizing the size of the subdomain boundary will minimize the inter-processor communications overhead. Such a decomposition is usually obtained by a graph partitioning algorithm. Recently, a number of multilevel graph partitioning algorithms (e.g. Hendrickson and Leland (1995), Karypis and Kumar (1998b,a), Monien et al. (1999), Walshaw and Cross (1999)) have been developed that are able to compute excellent static decompositions for a wide range of scientific simulations.

110.2.3.1 Finite Element Computations in Geomechanics

The distinct feature of elasto–plastic finite element computations is the presence of two iteration levels. In a standard displacement based finite element implementation, constitutive driver at each Gauss point iterates in stress and internal variable space, computes the updated stress state, constitutive stiffness tensor and delivers them to the finite element functions. Finite element functions then use the updated stresses and stiffness tensors to integrate new (internal) nodal forces and element stiffness matrix. Then, on global level, nonlinear equations are iterated on until equilibrium between internal and external forces is satisfied within some tolerance. In more details:

Elastic computations. In the case of elastic computations constitutive driver has a simple task of computing increment in stresses ($\Delta\sigma_{ij}$) for a given deformation increment ($\Delta\epsilon_{kl}$), through a closed form equation ($\Delta\sigma_{ij} = E_{ijkl}\Delta\epsilon_{kl}$) (Jeremić and Sture, 1997). It is important to note that in this case the amount of work per Gauss point is known in advance. The amount of computational work is the same for every integration point. If we assume the same number of integration points per element, it follows that the amount of computational work is the same for each element and it is known in advance.

Elasto–plastic computations. On the other hand, for elasto–plastic problems, for a given incremental deformation the constitutive driver is iterating in stress and internal variable space until consistency condition is satisfied ($F = 0$). The number of iterations is not known in advance. Initially, all Gauss points are in elastic range, but as we incrementally apply loads, the plastic zones develop. For Gauss points still in elastic range, there are no iterations, the constitutive driver just computes incremental stresses from closed form solution. Computational load will increase significantly for integration of constitutive equations in plastic range. In particular, constitutive level integration algorithms (Jeremić et al., 1998, 1999; Jeremić and Yang, 2002) for soils are very computationally demanding. From the experience of the PI, more than 70% of wall clock time during an elasto–plastic finite element analysis is spent in constitutive level iterations. This is in sharp contrast with elastic computations where the dominant part is solving the system of equations which consumes about 80% of run time. The extent of additional, constitutive level iterations is not known before the actual computations are over. In other words, the extent of elastic-plastic domain is not known ahead of time.

The traditional preprocessing type of DD method (also known as topological DD) splits domain based on the initial geometry and assigns roughly the same number of elements to every computational node and minimizes the size of subdomain boundaries. This approach might result in serious computational load imbalance for elasto–plastic problems. For example one domain might be assigned all of the elasto–plastic elements and spend large amount of time in constitutive level iterations. The other domains will have elements in elastic state and thus spend far less computational time in computing stress increments. This results in program having to wait for the slowest domain (the one with large number of elasto–plastic finite elements) to complete constitutive level iterations and only proceed with global system iterations after that.

This illustrates a two–fold challenge with computational load balancing for elastic–plastic simulations in geomechanics.

110.2.3.2 Adaptive Computation

First, these computations are dynamic in nature. That is, the structure of elastic and elastic–plastic domains changes dynamically and unpredictably during the course of the computation. For this reason, a static decomposition computed as a pre-processing step is not sufficient to ensure the computational load-balance of the entire computation. Instead, periodic computational load-balancing is required during the course of the computation. The problem of computing a dynamic decomposition shares the same requirements as that of computing the initial decomposition (i.e., balance the mesh elements and minimize the inter-processor communications), while also requiring that the cost associated with redistributing the data in order to balance the computational load is minimized. This last requirement prevents us from simply computing a whole new static partitioning from scratch each time computational load-balancing is required.

Often, the objective of minimizing the data redistribution cost is at odds with the objective of minimizing the inter-processor communications. For applications in which the computational requirements of different regions of the domain change rapidly, or the amount of state associated with each element is relatively high, minimizing the data redistribution cost is preferred over minimizing the communications incurred during parallel processing.

For applications in which computational load-balancing occurs very infrequently, the key objective of a load-balancing algorithm is in obtaining the minimal inter-processor communications. For many application domains, it is straightforward to select a primary objective to minimize (i.e., minimize whichever cost dominates). However, one of the key issues concerning the elastic-plastic computation is that the number of iterations between computational load-balancing phases is both unpredictable and dynamic. For example, in the case of static problems, zones in the 3D solid may become plastic and then unload to elastic (during increments of loading) so that the extent of plastic zone is changing. The change can be both slow and rapid. Slow change usually occurs during initial loading phases, while the later deformation tends to localize in narrow zones rapidly and the rest of the solid unloads rapidly (becomes elastic again). The narrow, localized zone has heavy computational load on the constitutive level (in each integration point within elements). Similar phenomena is observed in seismic soil–structure interaction computations where stiff structure interacts with soft soil and elastic and elastic–plastic zones change significantly during loading cycles. In this type of computation, it is extremely difficult to select the type of computational load-balancing algorithm to employ. Furthermore, the preferred computational load-balancing algorithm is liable to change during the course of the computation, and so the selection must be made dynamically.

110.2.3.3 Multi-phase Computation

The second challenge associated with computational load-balancing elastic-plastic computations in geomechanics is that these are two-phase computations. That is, plastic computations follow up the elastic computations. There is a synchronization phase between the computations, as only after the elastic computation is finished is it possible to check if the plastic computation is required for a given integration (Gauss) point within an element. For regions of the mesh in which this check indicates that the plastic computation is necessary, lengthy plastic computations are then performed. The existence of the synchronization step between the two phases of the computation requires that each phase be individually load balanced. That is, it is not sufficient to simply sum up the relative times required for each phase and to compute a decomposition based on this sum. Doing so may lead to some processors having too much work during the elastic computation (and so, these may still be working after other processors are idle), and not enough work during the plastic computation, (and so these may be idle while other processors are still working), and vice versa. Instead, it is critical that every processor have an equal amount of work from both of the phases of the computation.

110.2.4 Parallel Computing Hardware

110.2.4.1 DMPs and SMPs

Scalability issues for SMPs Cache coherence

Compute Nodes, CPUs, Cores,

GPUs (band-with and latency with the main memeory)

Networks (band-with and latency) 10, 100, 1,000, 10,000, Infiniband,

Large parallel supercomputers

small, user owned parallel machines (clusters of clusters)

ESSI Computer

- 208 (784) CPU cores,
- 288GB (1056GB) of distributed RAM,
- 24TB (48TB) of distributed disk space, and
- dual neetwork, InfiniBand for MPI, and GigaBit for file system

Small ESSI Computer

- 32 CPU cores (AMD), 24 CPU cores (Intel)

- 64GB RAM,
- 4TB
- on-board network

110.2.5 Parallel Computing Software

110.2.5.1 Amdahl's Law

n is a number of parallel processes

B is the fraction of algorithm that is serial

Total time to finish (wall clock time) with n parallel processes $T(n)$

$$T(n) = T(1)(B + \frac{1}{n}(1 - B)) \tag{110.1}$$

Theoretical speedup

$$S(n) = \frac{T(1)}{T(n)} = \frac{T(1)}{T(1)(B + \frac{1}{n}(1 - B))} = \frac{1}{(B + \frac{1}{n}(1 - B))} \tag{110.2}$$

110.2.5.2 Static and Dynamic Graph Partitioning

google search, data mining etc.

110.2.5.3 Real parallel and embarrassingly parallel

Finite Element matrices computations (elastic and elastic-plastic)

System of equation solvers (non-iterative and iterative)

Examples for

elastic (elements (Seq) + solver(P))

elastic (elements (P) + solver (Seq))

elastic (elements (P) + solver (P))

elastic-plastic (Seq) + solver (P)

elastic-plastic (P) + solver (Seq)

elastic-plastic (P) + solver (P)

110.2.5.4 Parallel Computing for Elastic-Plastic FEM

- Current Parallel FEM are
  - Well developed for elastic FEM
  - Undeveloped for elastic–plastic FEM
  - Well developed for homogeneous distributed memory parallel (DMP) computers,
  - Undeveloped for multiple performance (multi–generation) DMPs (example MOOSE, ESSI...)
- Need: dynamic computational load balancing for
  - multiple element types,
  - multiple material models
  - multiple compute node performances
  - multiple network performance performances

110.2.5.5 Plastic Domain Decomposition

- Multi-objective optimization problem (minimize both the inter-processor communications, the data redistribution costs and create balanced partitions)
- computational load balancing adds overhead $T_{overhead} := T_{comm} + T_{regen}$
  - T_{comm} data communication load depending on network conditions.
  - T_{regen} model regeneration for new partitioning, application (model) dependent
- Computional load among CPUs $T_j := \sum_{i=1}^{nel} ElemCompLoad[i], \;\; j = 1, ..., nCPU$
- Goal: minimize maximum compute time (slowest CPU) $T_{max} := max(T_j) \;\; j = 1, ..., nCPU$
- Total compute time (not wall clock time) $T_{sum} := sum(T_j)$
- Best execution time (perfect load balancing) $T_{best} := T_{sum}/nCPU, \;\Rightarrow\; T_j \equiv T_{best}$ for each $j = 1, ..., nCPU$
- Best performance gain $T_{gain} := T_{max} - T_{best}$
- Computational load balancing is beneficial iff $T_{gain} \geq T_{overhead} = T_{comm} + T_{regen}$
- Scalability (saturation, superlinear ...)

110.2.5.6 Template Meta-programs

Fine grained parallelism

110.3 Plastic Domain Decomposition Algorithm

110.3.1 Introduction

Domain Decomposition approach is the most popular and effective method to implement parallel finite element method. The underlying idea is to physically divide the problem domain into subdomains and finite element calculations will be performed on each individual domain in parallel. Domain Decomposition can be overlapping or non-overlapping. The overlapping domain decomposition method divides the problem domain into several slightly overlapping subdomains. Non-overlapping domain decomposition is extensively used in continuum finite element modeling due to the relative ease to program and organize computations and is the one that will be examined in this chapter.

In general, a good non-overlapping decomposition algorithm should be able to

- handle irregular mesh of arbitrarily shaped domain.
- minimize the interface problem size by delivering minimum boundary connectivity, which will help reducing the communication overheads.

The well-known idea of domain decomposition method can be found in a 1870 paper by the father of domain decomposition, H.A. Schwarz (Rixena and Magoulès, 2007). Domain decomposition method is also the underlying paradigm of substructuring methods developed in the sixties, which aim at reducing the dimension of models in structural analysis by applying static condensation-type techniques to subdomains.

Other than static condensation, Farhat and Roux (1991a); Farhat (1991); Farhat and Geradin (1992) proposed FETI (Finite Element Tearing and Interconnecting) method for domain decomposition analysis. In FETI method, Lagrange multipliers are introduced to enforce compatibility at the interface nodes. Rigid body modes are eliminated in parallel from each local problem and a direct scheme is applied concurrently to all subdomains in order to recover each partial local solution. The contributions of these modes are then related to the Lagrange multipliers through an orthogonality condition. This FETI method has been shown that it can deliver high efficiency for parallel implicit transient simulations in structural mechanics (Crivelli and Farhat, 1993).

Domain decomposition itself has become a active topic as parallel processing techniques receive much more attention in mathematics and engineering world during recent years. Domain decomposition was revived as a natural paradigm for parallel solvers (Rixena and Magoulès, 2007). Many papers have discussed two algorithms that are currently receiving much research effort, namely the FETI-DP (or Dual Primal Finite Element Tearing and Interconnecting) method and the even more recent BDDC (or Balancing Domain Decomposition by Constraints).

FETI-DP is the third generation FETI method (Bavestrello et al., 2007) developed for the fast, scalable, and domain-decomposition-based iterative solution of symmetric systems of equations arising from the finite element (FE) discretization of static, dynamic, structural and acoustic problems (Farhat et al., 2001, 2000).

BDDC, on the other hand, derives its formulation from substructuring method by enforcing constraints associated with disjoint sets of nodes on substructure boundaries using constrained energy minimization concepts (Dohrmann, 2003; Mandel and Dohrmann, 2003).

An early endeavor on dynamic computational load balancing was presented by McKenna (1997). Limited number of examples show that run time, dynamic computational load balancing can indeed improve parallel program performance in some cases, particularly when nonlinearities are involved.

Although many works have been presented on domain decomposition methods, the most popular methods such as FETI-type and BDDC all stem from the root of subdomain interface constraints handling. The merging of iterative solving with domain decomposition-type preconditioning is promising as shown by many researchers (Pavarino, 2007; Li and Widlund, 2007). Schwartz-type preconditioners for parallel domain decomposition system solving have also shared part of the spotlight (Hwang and Cai, 2007; Sarkis and Szyld, 2007).

In solid finite element methods, it has been assumed that the equation solving is the most computational expensive part so it is totally reasonable that all focus has been set on equation solver during the past decades.

Work presented in this chapter, however, has originated from the observation that for highly nonlinear materials, the constitutive level computation can be at least equally costly as equation solving, if not more expensive. The novelty of this chapter is to break out of the existing substructuring or FETI frameworks to further address the fundamental load balance issue of parallel computing. Namely, in order to achieve better parallel performance, we want to keep all processors equally busy. Load imbalance issue resulted from nonlinear constitutive level computations is too important to be neglected. This chapter proposes the Plastic Domain Decomposition algorithm which focuses on adaptive load balancing operation for nonlinear finite elements.

From the implementation point of view, for mesh-based scientific computations, domain decomposition corresponds to the problem of mapping a mesh onto a set of processors, which is well defined as a graph partitioning problem (Schloegel et al., 1999).

Formally, the graph partitioning problem is as follows. Given a weighted, undirected graph $G = (V; E)$ for which each vertex and edge has an associated weight, the k-way graph partitioning problem is to split the vertices of V into k disjoint subsets (or subdomains) such that each subdomain has roughly an equal amount of vertex weight (referred to as the balance constraint), while minimizing the sum of the weights of the edges whose incident vertices belong to different subdomains (i. e., the edge-cut).

In computational solid mechanics, the element graph is naturally used in parallel finite element method due to the fact that elemental operation forms the basis of finite element method. On the other hand, for material nonlinearity simulations, the element calculations represent the most computationally expensive part. In order to facilitate consistent interfaces for computational load measuring and data migration, element graph has been utilized as fundamental graph structure in this chapter, although it has been shown that the node-graph can be used as well for structure dynamics problem and the element-cut partitioning can make certain algorithms simpler (Krysl and Bittnar, 2001).

The graph partitioning problem is known to be NP-complete [1]. Therefore, generally it is not possible to compute optimal partitioning for graphs of interesting size in a reasonable amount of time. Various heuristic approaches have been developed, which can be classified as either geometric, combinatorial, spectral, combinatorial optimization techniques, or multilevel methods (Dongarra et al., 2003).

In finite element simulations involving nonlinear material response, static graph partitioning mentioned above does not guarantee even load distribution among processors. Plastification introduces work load that is much heavier than pure elastic computation. So for this kind of multiphase simulation, adaptive computational load balancing scheme has to be considered to keep all processing units equally busy as much as possible. Traditional static graph partitioning algorithm is not adequate to do multiphase partition/repartitioning. A parallel multilevel graph partitioner has been introduced in this research to achieve dynamic load balancing for inelastic finite element simulations.

In this chapter, the algorithm of Plastic Domain Decomposition (PDD) is proposed. The adaptive multi-level graph partitioning kernel of the PDD algorithm is implemented through the ParMETIS interface. Studies are performed to extract optimal algorithmic parameters for our specific applications.

110.3.2 Inelastic Parallel Finite Element

The distinct feature of inelastic (elastic-plastic) finite element computations is the presence of two iteration levels. In a standard displacement based finite element implementation, constitutive driver

---

[1] The complexity class NP is the set of decision problems that can be solved by a non-deterministic Turing machine in polynomial time. the NP-complete problems are the most difficult problems in NP ("non-deterministic polynomial time") in the sense that they are the smallest subclass of NP that could conceivably remain outside of P, the class of deterministic polynomial-time problems, http://en.wikipedia.org/wiki/NP-complete

at each Gauss point iterates in stress and internal variable space, computes the updated stress state, constitutive stiffness tensor and delivers them to the finite element functions. Finite element functions then use the updated stresses and stiffness tensors to integrate new (internal) nodal forces and element stiffness matrix. Then, on global level, nonlinear equations are iterated on until equilibrium between internal and external forces is satisfied within some tolerance.

- Elastic Computations

  In the case of elastic computations constitutive driver has a simple task of computing increment in stresses ($\Delta\sigma_{ij}$) for a given deformation increment ($\Delta\epsilon_{kl}$), through a closed form equation ($\Delta\sigma_{ij} = E_{ijkl}\Delta\epsilon_{kl}$) It is important to note that in this case the amount of work per Gauss point is known in advance. The amount of computational work is the same for every integration point. If we assume the same number of integration points per element, it follows that the amount of computational work is the same for each element and it is known in advance.

- Elastic-Plastic Computations

  On the other hand, for elastic-plastic problems, for a given incremental deformation the constitutive driver is iterating in stress and internal variable space until consistency condition is satisfied ($F = 0$). The number of iterations is not known in advance. Initially, all Gauss points are in elastic range, but as we incrementally apply loads, the plastic zones develop. For Gauss points still in elastic range, there are no iterations, the constitutive driver just computes incremental stresses from closed form solution. Computational load will increase significantly for integration of constitutive equations in plastic range. In particular, constitutive level integration algorithms for soils, concrete, rocks, foams and other granular materials are very computationally demanding. More than 70% of wall clock time during an elastic-plastic finite element analysis is spent in constitutive level iterations. This is in sharp contrast with elastic computations where the dominant part is solving the system of equations which consumes about 80% of run time. The extent of additional, constitutive level iterations is not known before the actual computations are over. In other words, the extent of elastic-plastic domain is not known ahead of time.

  The traditional preprocessing type of Domain Decomposition method (also known as topological DD) splits domain based on the initial geometry and assigns roughly the same number of elements to every computational node and minimizes the size of subdomain boundaries. This approach might result in serious computational load imbalance for elastic-plastic problems. For example one domain might be assigned all of the elastic-plastic elements and spend large amount of time in constitutive level iterations. The other domains will have elements in elastic state and thus spend far less computational time in computing stress increments. This results in program having to wait for the slowest domain (the one with large number of elastic-plastic finite elements) to complete constitutive level iterations and only proceed with global system iterations after that.

This illustrates a two-fold challenge with computational load balancing for inelastic simulations in mechanics. These two challenges is described below in some more detail.

110.3.2.1 Adaptive Computation

First, these computations are dynamic in nature. That is, the structure of elastic and elastic-plastic domains changes dynamically and unpredictably during the course of the computation. For this reason, a static decomposition computed as a pre-processing step is not sufficient to ensure the computational load-balance of the entire computation. Instead, periodic computational load-balancing is required during the course of the computation. The problem of computing a dynamic decomposition shares the same requirements as that of computing the initial decomposition (i.e., balance the mesh elements and minimize the inter-processor communications), while also requiring that the cost associated with redistributing the data in order to balance the computational load is minimized. This last requirement prevents us from simply computing a whole new static partitioning from scratch each time computational load-balancing is required.

Often, the objective of minimizing the data redistribution cost is at odds with the objective of minimizing the inter-processor communications. For applications in which the computational requirements of different regions of the domain change rapidly, or the amount of state associated with each element is relatively high, minimizing the data redistribution cost is preferred over minimizing the communications incurred during parallel processing.

For applications in which computational load-balancing occurs very infrequently, the key objective of a load-balancing algorithm is in obtaining the minimal inter-processor communications. For many application domains, it is straightforward to select a primary objective to minimize (i.e., minimize whichever cost dominates). However, one of the key issues concerning the elastic-plastic computation is that the number of iterations between computational load-balancing phases is both unpredictable and dynamic. For example, in the case of static problems, zones in the 3D solid may become plastic and then unload to elastic (during increments of loading) so that the extent of plastic zone is changing. The change can be both slow and rapid. Slow change usually occurs during initial loading phases, while the later deformation tends to localize in narrow zones rapidly and the rest of the solid unloads rapidly (becomes elastic again). The narrow, localized zone has heavy computational load on the constitutive level (in each integration point within elements). Similar phenomena is observed in seismic soil-structure interaction computations where stiff structure interacts with soft soil and elastic and elasto-plastic zones change significantly during loading cycles. In this type of computation, it is extremely difficult to select the type of computational load-balancing algorithm to employ. Furthermore, the preferred computational load-balancing algorithm is liable to change during the course of the computation, and so the selection must be made dynamically.

110.3.2.2 Multiphase Computation

The second challenge associated with computational load-balancing elastic-plastic computations in geomechanics is that these are two-phase computations. That is, elastic-plastic computations follow up the elastic computations. There is a synchronization phase between the computations, as only after the elastic computation is finished is it possible to check if the elastic-plastic computation is required for a given integration (Gauss) point within an element. For regions of the mesh in which this check indicates that the elastic-plastic computation is necessary, lengthy elastic-plastic computations are then performed. The existence of the synchronization step between the two phases of the computation requires that each phase be individually load balanced. That is, it is not sufficient to simply sum up the relative times required for each phase and to compute a decomposition based on this sum. Doing so may lead to some processors having too much work during the elastic computation (and so, these may still be working after other processors are idle), and not enough work during the elastic-plastic computation, (and so these may be idle while other processors are still working), and vice versa. Instead, it is critical that every processor have an equal amount of work from both of the phases of the computation.

110.3.2.3 Multiconstraint Graph Partitioning

Elastic-plastic FE computation can be understood as a two-phase calculation, which is also dynamic in nature. Traditional graph partitioning formulations are not adequate to ensure its efficient execution on high performance parallel computers. In this chapter very recent progresses from the graph partitioning algorithm research will be investigated. We need new adaptive graph partitioning formulations, which can compute adaptive partitioning-repartitionings that can satisfy an arbitrary number of balance constraints.

- Static Graph Partitioning
  Given a weighted, undirected graph $G = (V, E)$, for which each vertex and edge has an associated weight, the k-way graph partitioning problem is to split the vertices of V into k disjoint subsets (or subdomains) such that each subdomain has roughly an equal amount of vertex weight (referred to as the balance constraint), while minimizing the sum of the weights of the edges whose incident vertices belong to different subdomains (i.e., the edge-cut).

1. Geometric Techniques

   Compute partitioning based solely on the coordinate information of the mesh nodes, without considering edge-cut. Popular methods include, Coordinate Nested Dissection (CND or Recursive Coordinate Bisection), Recursive Inertial Bisection (RIB), Space-Filling Curve techniques and Sphere-Cutting approach.

2. Combinatorial Techniques

   Attempt to group together highly connected vertices whether or not these are near each other in space. That is combinatorial partitioning schemes compute a partitioning based only on the adjacency information of the graph; they do not consider the coordinates of the vertices. They tend to have lower edge-cuts but generally slower. Popular methods include, Levelized Nested Dissection (LND) and Kernighan-Lin/Fiduccia-Mattheyses (KL/FM) partitioning refinement algorithm, which needs an initial partition input to do swapping refinement.

3. Multilevel Schemes

   The multilevel paradigm consists of three phases: graph coarsening, initial partitioning, and multilevel refinement. Firstly, we form coarse graph by collapsing together selected vertices of the input graph. After rounds of coarsening, we get coarsest graph, on which an initial bisection will be performed. Then the KL/FM algorithm can be used to refine the partition back to the finest graph.

   The multilevel paradigm works well for two reasons. First, a good coarsening scheme can hide a large number of edges on the coarsest graph, which makes the task of computing high-quality partitioning easier. Second reason, incremental refinement schemes such as KL/FM become much more powerful in the multilevel context.

   Popular algorithms include Multilevel Recursive Bisection and Multilevel k-Way Partitioning.

- Adaptive Graph Partitioning

  For large scale elasto-plastic FE simulations, it is necessary to dynamically load-balance the computations as the analysis progresses due to unpredictable plastification inside the domain. This dynamic load balancing can be achieved by using a graph partitioning algorithm.

  Adaptive graph partitioning shares most of the requirements and characteristics of static graph partitioning but also adds an additional objective. That is, the amount of data that needs to be redistributed among the processors in order to balance the load should be minimized. If the vertex weight represents the computational cost of the work carried by the vertex, another metric, size of the vertex needs to be considered as well, which reflects distribution cost of the vertex. Thus, the repartitioner should attempt to balance the partitioning with respect to vertex weight while minimizing vertex migration with respect to vertex size.

Different approaches are available. One can simply compute a new graph from scratch, so called Scratch-Remap Repartitioner, which expectedly introduces more data redistribution than necessary. Diffusion-Based Repartitioner attempt to minimize the difference between the original partitioning and the final repartitioning by making incremental changes in the partitioning to restore balance. This method has been an very active topic during recent years, Dongarra et al. (2003) gives up-to-date review.

- Multiconstraint Graph Partitioning

  We can see traditional graph partitioning typically balances only a single constraint (i.e., the vertex weight) and minimizes only a single objective (i.e., the edge-cut). If we replace the vertex weight, which is a single number, with a weight vector of size m, then the problem becomes that of finding a partitioning that minimizes the edge-cuts subject to the constraints that each of the m weights is balanced across subdomains.

  Multilevel graph partitioning algorithms for solving multiconstraint/multiobjective problems have been very successful Schloegel et al. (1999). The software libraries METIS and ParMETIS are widely used in computational mechanics research.

110.3.2.4 Adaptive PDD Algorithm

In this chapter, the Plastic Domain Decomposition (PDD) has been developed using multi-level, multi-objective graph partitioning algorithm. This algorithm automatically monitors load balancing condition and updates element graph structure accordingly as the simulation progresses. Element redistribution will be triggered to achieve load balance when nonlinearity of materials brings down the parallel performance.

110.3.3 Adaptive Multilevel Graph Partitioning Algorithm

Kaypis and Kumar (1998) present a k-way multilevel partitioning algorithm whose run time is linear in the number of edges $|E|$ (i.e., O($|E|$)); whereas the run time of multilevel recursive bisection schemes is O($|E|$log$k$) for k-way partitioning. Kaypis and Kumar (1998) show that the proposed multilevel partitioning scheme produces partitioning that are of comparable or better quality than those produced by multilevel recursive bisection, while requiring substantially less time. This paradigm consists of three phases: graph coarsening, initial partitioning, and multilevel refinement. In the graph coarsening phase, a series of graphs is constructed by collapsing together selected vertices of the input graph in order to form a related coarser graph. This newly constructed graph then acts as the input graph for another round of graph coarsening, and so on, until a sufficiently small graph is obtained. Computation of the initial bisection is performed on the coarsest (and hence smallest) of these graphs, and so is very fast. Finally, partition refinement is performed on each level graph, from the coarsest to the nest (i.e., original graph) using a KL/FM-type algorithm Dongarra et al. (2003). Figure 110.1 illustrates the multilevel paradigm. This algorithm is available in METIS Karypis and Kumar (1998d) which is used in this research to provide initial static partitioning.

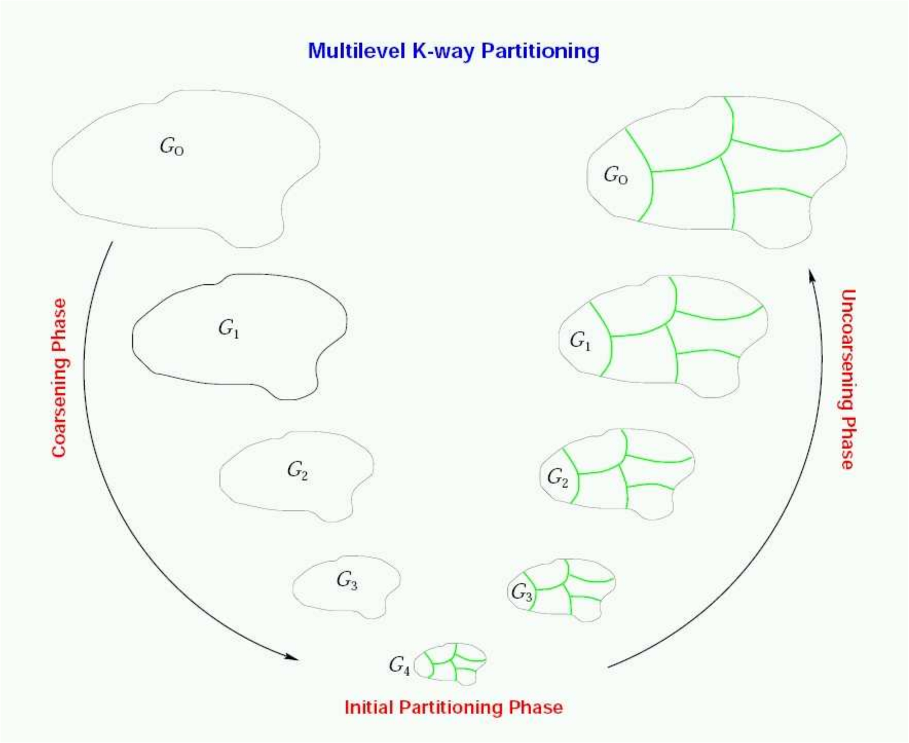


Figure 110.1: Multilevel Graph Partitioning Scheme Karypis et al. (2003)

Adaptive graph repartitioning algorithm can be used to achieve dynamic load balancing of multiphase elastic-plastic finite element simulations. Adaptive graph partitioning differs from static graph partitioning algorithm in the sense that one additional objective has to be targeted. That is, the amount of data the needs to be redistributed among the processors in order to balance the load should be minimized. In order to measure this redistribution cost, not only does the weight of a vertex, but also its size have to be considered. In our implementation for the purpose of this research, the vertex weight represents the computational load of each finite element, while the size reflects its redistribution cost. Thus, the application of adaptive graph partitioning algorithm aims at balancing the partitioning with respect to vertex weight while minimizing vertex migration with respect to vertex size.

A repartitioning of a graph can be obtained simply by partitioning a new graph from a scratch, which tends to bring much more unnecessary communications because the old distribution has not been taken into account. Diffusion-based Repartitioner is more popular in which one attempts to minimize the difference between the original partitioning and the final repartitioning by making incremental changes in the partitioning to restore balance. Dongarra et al. (2003) gives a comprehensive review on this subject. Adaptive repartitioning is available in ParMETIS Karypis et al. (2003) and Jostle Warshaw (1998). The former is chosen in this research considering the fact that ParMETIS provides seamless interface for METIS 4.0 which makes the comparison between static and adaptive partitioning schemes more consistent.

PARMETIS is an MPI-based parallel library that implements a variety of algorithms for partitioning and repartitioning unstructured graphs and for computing fill-reducing orderings of sparse matrices Karypis et al. (2003). PARMETIS is particularly suited for parallel numerical simulations involving large unstructured meshes. In this type of computation, PARMETIS dramatically reduces the time spent in communication by computing mesh decompositions such that the numbers of interface elements are minimized. The algorithms in PARMETIS are based on the multilevel partitioning and fill-reducing ordering algorithms that are implemented in the widely-used serial package METIS Karypis and Kumar (1998c). However, PARMETIS extends the functionality provided by METIS and includes routines that are especially suited for parallel computations and large-scale numerical simulations. In particular, PARMETIS provides the following functionality Karypis et al. (2003):

- Partition unstructured graphs and meshes.
- Repartition graphs that correspond to adaptively refined meshes.
- Partition graphs for multi-phase and multi-physics simulations.
- Improve the quality of existing partitioning.
- Compute fill-reducing orderings for sparse direct factorization.
- Construct the dual graphs of meshes.

Both METIS and PARMETIS are used in this research. METIS routines are called to construct static partitioning for commonly used one-step static domain decomposition, while adaptive load-balancing is achieved by calling PARMETIS routines regularly during the progress of nonlinear finite element simulations.

Adaptive load-balancing through domain repartitioning is a multi-objective optimization problem, in which repartitionings should minimize both the inter-processor communications incurred in the iterative mesh-based computation and the data redistribution costs required to balance the load. PARMETIS provides the routine ParMETIS_V3_AdaptiveRepart for repartitioning the previous unbalanced computational domain. This routine assumes that the existing decomposition is well distributed among the processors, but that (due to plastification of certain nonlinear elements) this distribution is poorly load balanced.

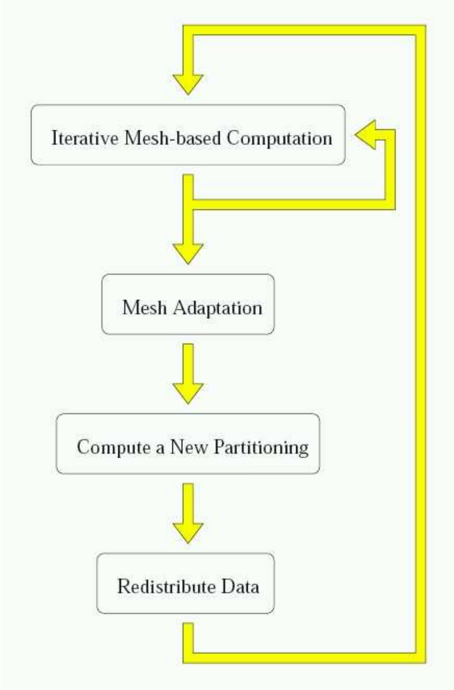


Figure 110.2: A diagram illustrating the execution of adaptive scientific simulations on high performance parallel computers Schloegel et al. (1999)

Figure 110.2 Schloegel et al. (2000) shows common steps involved in the execution of adaptive mesh-based simulations on parallel computers. Initially, the mesh is equally distributed on different processors. As all elements are elastic at the very beginning (carrying the same amount of elemental calculation work), computation load balance can be guaranteed with a even distribution. A number of iterations of the simulation are performed in parallel, after which plasticity occurs in certain nonlinear elements thus introducing some amount of load imbalance. A new partitioning based on the unbalanced domain is computed to re-balance the load, and then the mesh is redistributed among the processors, respectively. The simulation can then continue for another number of iterations until either more mesh adaptation is required or the simulation terminates.

If we consider each round of executing a number of iterations of the simulation, mesh adaptation, and load-balancing to be an epoch, then the run time of an epoch can be described by, Schloegel et al.

(2000)

$$(t_{comp} + f(|E_{cut}|))n + t_{repart} + g(|V_{move}|) \tag{110.3}$$

where n is the number of iterations executed, t_{comp} is the time to perform the computation for a single iteration of the simulation, $f(|E_{cut}|)$ is the time to perform the communications required for a single iteration of the simulation, and t_{repart} and $g(|V_{move}|)$ represent the times required to compute the new partitioning and to redistribute the data. Here, the inter-processor communication time is described as a function of the edge-cut of the partitioning and the data redistribution time is described as a function of the total amount of data that is required to be moved in order to realize the new partitioning. Adaptive repartitioning affects all of terms in Equation 110.3. How well the new partitioning is balanced influences t_{comp}. The inter-processor communications time is dependent on the edge-cut of the new partitioning. The data redistribution time is dependent on the total amount of data that is required to be moved in order to realize the new partitioning. It is critical for adaptive partitioning schemes to minimize both the edge-cut and the data redistribution when computing the new partitioning. Viewed in this way, adaptive graph partitioning is a multi-objective optimization problem.

There are various approaches how to handle this dual-objective problem. In general, two approaches have primarily been taken when designing adaptive partitioners. Schloegel et al. (2000) gives a comprehensive review on this topic. The first approach is to attempt to focus on minimizing the edge-cut and to minimize the data redistribution only as a secondary objective. This family of methods can be called scratch-remap repartitioner. These use some type of state-of-the-art graph partitioner to compute a new partitioning from scratch and then attempt to intelligently remap the subdomain labels to those of the original partitioning in order to minimize the data redistribution costs. Since a state-of-the-art graph partitioner is used to compute the partitioning, the resulting edge-cut tends to be extremely good. However, since there is no guarantee as to how similar the new partitioning will be to the original partitioning, data redistribution costs can be high, even after remapping. The second approach is to focus on minimizing the data redistribution cost and to minimize the edge-cut as a secondary objective, or so-called diffusion-based repartitioner. These schemes attempt to perturb the original partitioning just enough so as to balance it. This strategy usually leads to low data redistribution costs, especially when the partitioning is only slightly imbalanced. However, it can result in higher edge-cuts than scratch-remap methods because perturbing a partitioning in order to balance it also tends to adversely affect its quality.

These two types of repartitioner allow the user to compute partitioning that focus on minimizing either the edge-cut or the data redistribution costs, but give the user only a limited ability to control the tradeoffs among these objectives. This control of the tradeoffs is sufficient if the number of iterations that a simulation performs between load-balancing phases (i.e. the value of n in Equation 110.3) is either very high or very low. However, when n is neither very high nor very low, neither type of scheme precisely minimizes the combined costs of $f(E_{cut}|)n$ and $g(|V_{move}|)$. Another disadvantage exists for applications in which n is difficult to predict or those in which n can change dynamically throughout the course of the computation. As an example, one of the key issues concerning the elastic-plastic soil-structure interaction computations required for earthquake simulation is that the number of iterations between load-balancing phases is both unpredictable and dynamic. Here, zones in the 3D solid may become plastic and then unload (during increments of loading) so that the extent of the plastic zone is changing. The change can be both slow and rapid. Slow change usually occurs during initial loading phases, while the later deformation tends to localize in narrow zones rapidly and the rest of the solid unloads rapidly (becomes elastic again) Jeremić and Xenophontos (1999).

Schloegel et al. (2000) presents a parallel adaptive repartitioning scheme (called the Unified Repartitioning Algorithm) for the dynamic load-balancing of scientific simulations that attempts to solve the precise multi-objective optimization problem. By directly minimizing the combined costs of $f(E_{cut}|)n$ and $g(|V_{move}|)$, the proposed scheme is able to gracefully tradeoff one objective for the other as required by the specific application. The paper shows that when inter-processor communication costs are much greater in scale than data redistribution costs, the proposed scheme obtains results that are similar to those obtained by an optimized scratch-remap repartitioner and better than those obtained by an optimized diffusion-based repartitioner. When these two costs are of similar scale, the scheme obtains results that are similar to the diffusive repartitioner and better than the scratch-remap repartitioner. When the cost to perform data redistribution is much greater than the cost to perform inter-processor communication, the scheme obtains better results than the diffusive scheme and much better results than the scratch-remap scheme. They also show in the paper that the Unified Repartitioning Algorithm is fast and scalable to very large problems.

110.3.3.1 Unified Repartitioning Algorithm

A key parameter used in Unified Repartitioning Algorithm (URA) is the Relative Cost Factor (RCF). This parameter describes the relative times required for performing the inter-processor communications incurred during parallel processing and to perform the data redistribution associated with balancing the load. Using this parameter, it is possible to unify the two minimization objectives of the adaptive graph partitioning problem into the unified cost function

$$|E_{cut}| + \alpha|V_{move}| \tag{110.4}$$

where α is the Relative Cost Factor, $|E_{cut}|$ is the edge-cut of the partitioning, and $|V_{move}|$ is the total amount of data redistribution. The Unified Repartitioning Algorithm attempts to compute a repartitioning while directly minimizing this cost function.

The Unified Repartitioning Algorithm is based upon the multilevel paradigm that is illustrated in Figure 110.1, which can be described as three phases: graph coarsening, initial partitioning, and uncoarsening/refinement Schloegel et al. (2000). In the graph coarsening phase, coarsening is performed using a purely local variant of heavy-edge matching. That is, vertices may be matched together only if they are in the same subdomain on the original partitioning. This matching scheme has been shown to be very effective at helping to minimize both the edge-cut and data redistribution costs and is also inherently more scalable than global matching schemes.

110.3.3.2 Study of ITR in ParMETIS

The RCF in the URA implementation controls the tradeoff between two objectives, minimizing data redistribution cost or edge-cut. In our application, ParMETIS library has been linked to a MOSS (Modified OpenSees Services) analysis model to facilitate the partitioning/adaptive repartitioning scheme. The RCF is defined as a single parameter ITR in ParMETIS Karypis et al. (2003). This parameter describes the ratio between the time required for performing the inter-processor communications incurred during parallel processing compared to the time to perform the data redistribution associated with balancing the load. As such, it allows us to compute a single metric that describes the quality of the repartitioning, even though adaptive repartitioning is a multi-objective optimization problem. As recommended by Karypis et al. (2003), appropriate values to pass for the ITR Factor parameter can be determined depending on the times required to perform

1. all inter-processor communications that have occurred since the last repartitioning, and
2. the data redistribution associated with the last repartitioning/load balancing phase.

Simply divide the first time measurement by the second time measurement. The result is the correct ITR Factor. In case these times cannot be ascertained (e.g., for the first repartitioning/load balancing phase), Karypis et al. (2003) suggests that values between 100 and 1000 work well for a variety of situations. By default ITR is between 0.001 and 1000000. If ITR is set high, a repartitioning with a low edge-cut will be computed. If it is set low, a repartitioning that requires little data redistribution will be computed.

110.4 Performance Studies on PDD Algorithm

110.4.1 Introduction

In this chapter, parallel performance of the proposed PDD algorithm is thoroughly investigated. There are two major focuses for the timing analysis. Firstly we want to see how much performance gain we can have by introducing the PDD algorithm into inelastic finite element calculations. Secondly, we also want to show how scalable the proposed PDD algorithm is.

As our final objective is to apply PDD in large scale SFSI finite element simulations, finite element models of SFSI have been set up to study the parallel performance of the PDD based parallel program. Implicit constitutive integration scheme Jeremić and Sture (1997) has been used to expose the load imbalance by plasticity calculation. Only continuum element has been studied due to the fact they can be easily visualized to obtain partition and/or repartition figures.

Distributed memory Linux/Unix clusters are major platforms used in this chapter for speed up analysis.

110.4.2 Parallel Computers

Performance measurement has been carried out on two SMP-based clusters.

- IBM eServer p655

  The DataStar IBM eServer p655 cluster consists of 176 8-way P655+ nodes at San Diego Supercomputer Center. System configuration is shown in Fig 110.3. The network benchmark is shown in Table 110.1.

  Table 110.1: Latency and Bandwidth Comparison (as of August 2004)

	MPI Latencies (μsec)	Bandwidth (MBs)
Intra-node	3.9	3120.4
Inter-node	7.65	1379.1

- TeraGrid IA-64 Intel-Based Linux Cluster

  The TeraGrid project was launched by the the National Science Foundation with $53 million in funding to four sites: the National Center for Supercomputing Applications (NCSA) at the University of Illinois, Urbana-Champaign, the San Diego Supercomputer Center (SDSC) at the University of California, San Diego, Argonne National Laboratory in Argonne, IL, and Center for Advanced Computing Research (CACR) at the California Institute of Technology in Pasadena.

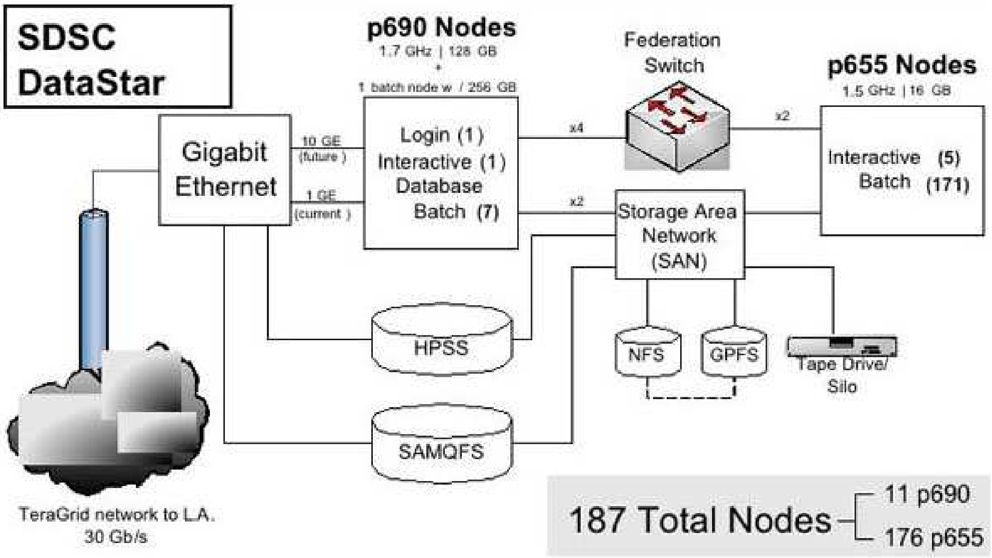


Figure 110.3: System Configuration of DataStar http://www.sdsc.edu/user_services/datastar/ (2020)

SDSC's TeraGrid cluster currently consists of 256 IBM cluster nodes, each with dual 1.5 GHz Intel Itanium 2 processors, for a peak performance of 3.1 teraflops. The nodes are equipped with four gigabytes (GBs) of physical memory per node. The cluster is running SuSE Linux and is using Myricom's Myrinet cluster interconnect network. Table 110.2 shows the technical configuration of the IA64 cluster, on which the second part of the performance study has been done.

110.4.3 Soil-Foundation Interaction Model

A soil-shallow-foundation interaction model as shown in Figure 110.4 has been set up to study the parallel performance. 3D brick element with 8 integration (Gaussian) points is used. The soil is modeled by Template3D elasto-plastic material model (Drucker-Prager model with Armstrong Frederick nonlinear kinematic hardening rule) and linear elasticity is assumed for the foundation. More advanced constitutive laws can be applied through Template3D model although the model used here suffices the purpose of this research to show repartitioning triggered by plastification. It is shown in this research that the speedup by adaptive load balancing is significant even for seemingly simple constitutive model. The material properties are shown in Table 110.3 and the vertical loading is applied at 5.0kN increments. The performance analysis has been carried out on DataStar supercomputer at San Diego Supercomputing Center (P655+ 8-way nodes).

Table 110.2: Technical Information of IA64 TeraGrid Cluster at SDSC

IA-64 Cluster (tg-login.sdsc.teragrid.org)	
COMPONENT	DESCRIPTION
Architecture	Linux Cluster
Access Nodes	⋆ quad-processor ⋆ ECC SDRAM memory: 8 GB ⋆ 2 nodes (8 processors)
Compute Nodes	⋆ dual-processor ⋆ ECC SDRAM memory: 4 GB ⋆ 262 nodes (524 processors)
Processor	⋆ Intel Itanium 2, 1.5 GHz ⋆ Integrated 6 MB L3 cache ⋆ Peak performance 3.1 Tflops
Network Interconnect	Myrinet 2000, Gigabit Ethernet, Fiber Channel
Disk	1.7 TB of NFS, 50 TB of GPFS (Parallel File System)
Operating System	Linux 2.4-SMP (SuSE SLES 8.0)
Compilers	⋆ Intel: Fortran77/90/95 C C++ ⋆ GNU: Fortran77 C C++
Batch System	Portable Batch System (PBS) with Catalina Scheduler

Table 110.3: Material Constants for Soil-Foundation Interaction Model

Soil	
Elastic modulus	$E = 17400\text{kPa}$
Poisson ratio	$\nu = 0.35$
Friction angle	$\phi = 37.1^{\circ}$
Cohesion	$c = 0$
Isotropic Hardening	Linear
Kinematic Hardening	A/F nonlinear ($h_a = 116.0$, $C_r = 80.0$)
Foundation	
Elastic modulus	$E = 21\text{GPa}$
Poisson ratio	$\nu = 0.2$

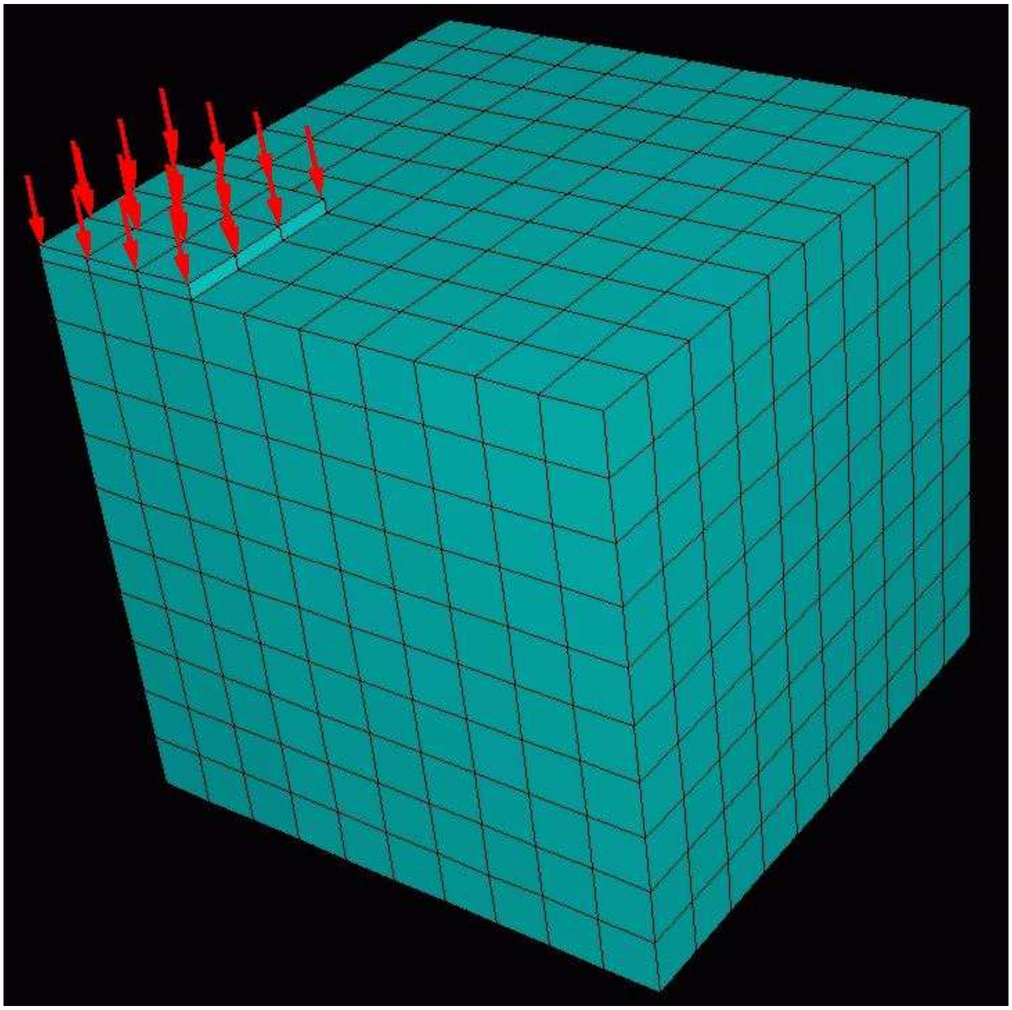


Figure 110.4: Example Finite Element Model of Soil-Foundation Interaction (Indication Only, Real Model Shown in Each Individual Section)

110.4.4 Numerical Study for ITR

As described in Section 110.3.3.2, the parameter ITR in ParMETIS describes the ratio between the time required for performing the inter-processor communications incurred during parallel processing compared to the time to perform the data redistribution associated with balancing the load. It acts like a switch on algorithmic approaches of ParMETIS repartitioning kernel. With ITR factor being very small, the ParMETIS tends to do that repartitioning which can minimize data redistribution cost. If the ITR factor is set to be very large, ParMETIS tends to minimize edge-cut of the final repartition.

In parallel design of PDD, if repartitioning is necessary to achieve load balance after each load increment, the whole AnalysisModel McKenna (1997) has to be wiped off thus a new analysis container can be defined to reload subsequent analysis steps. The data redistribution cost can be much higher than communication overhead only. In order to determine an adequate ITR value for our application, preliminary study needs to be performed to investigate the effectiveness of the URA. In this research, two extreme values of the ITR (0.001 and $1,000,000$) are prescribed and then parallel analysis is carried out on 2, 4 and 8 processors to see how the partition/repartition algorithm behaves. Two soil-structure interaction models as shown in Figure 110.6 have been used in this parametric study. Timing data and partition figures have been collected to investigate the performance of different approaches. The one that tends to bring better performance will be adopted in subsequent parallel analysis for prototype 3D soil structure interaction problems. Figure 110.7 to Figure 110.12 shows the initial partition and final repartition figures for two different types of algorithms. With ITR factor to be very small, the URA tends to present results that minimize data redistribution cost, in which diffusive repartitioning approach is used. On the other hand, if the ITR factor is set to be very large, then the URA algorithm tends to give repartitioning with lowest edge cut but with considerably higher data redistribution cost.

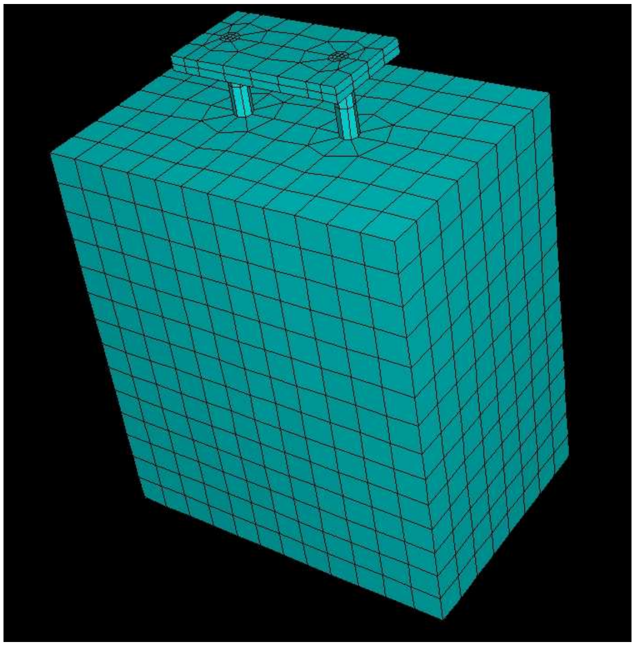


Figure 110.5: FE Models (1,968 Elements, 7,500 DOFs) for Studying Soil-Foundation Interaction Problems

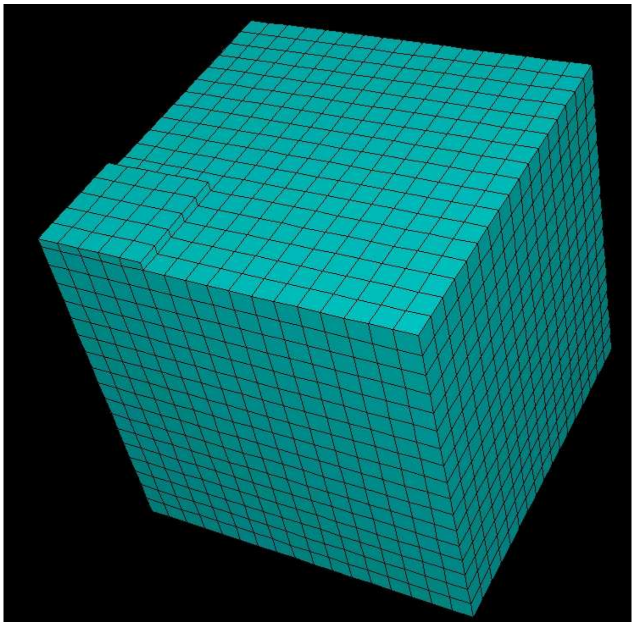


Figure 110.6: FE Models (4,938 Elements, 17,604 DOFs) for Studying Soil-Foundation Interaction Problems

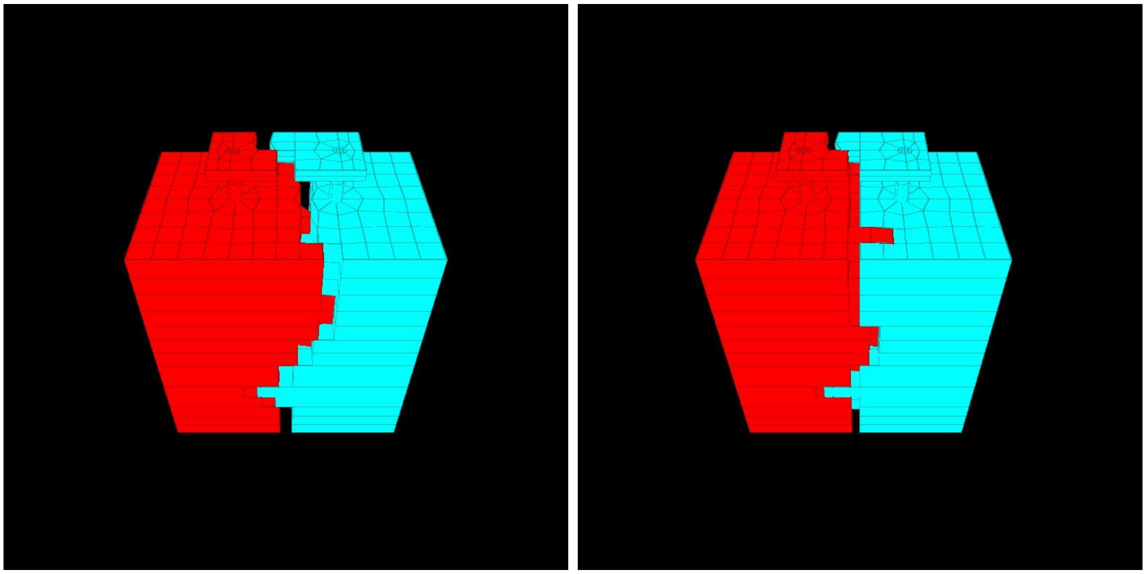


Figure 110.7: Partition and Repartition on 2 CPUs (ITR=1e-3, Imbal. Tol. 5%), FE Model (1,968 Elements, 7,500 DOFs)

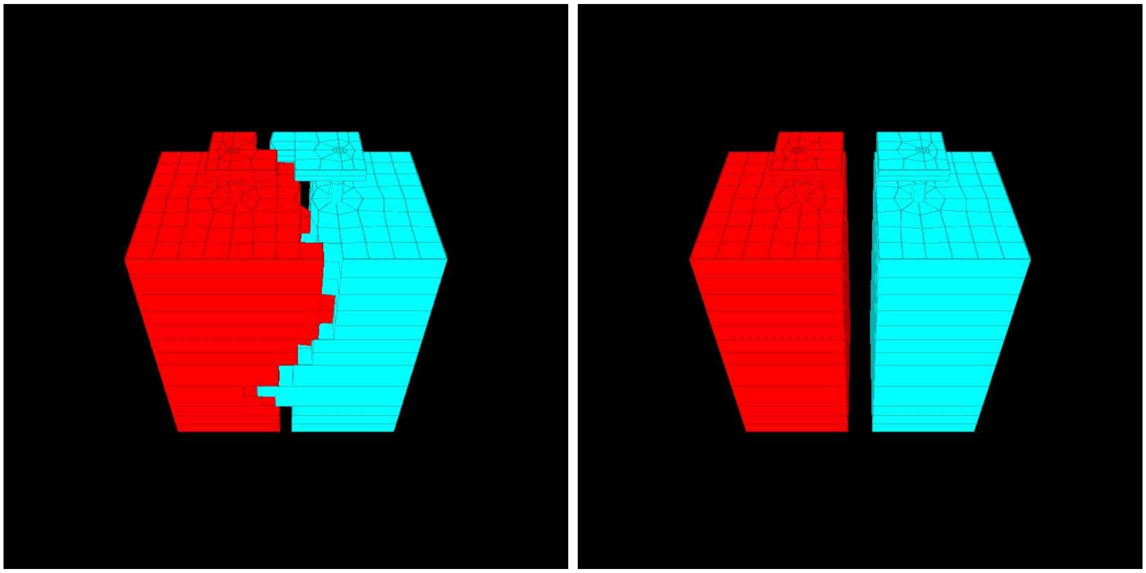


Figure 110.8: Partition and Repartition on 2 CPUs (ITR=1e6, Imbal. Tol. 5%), FE Model (1,968 Elements, 7,500 DOFs)

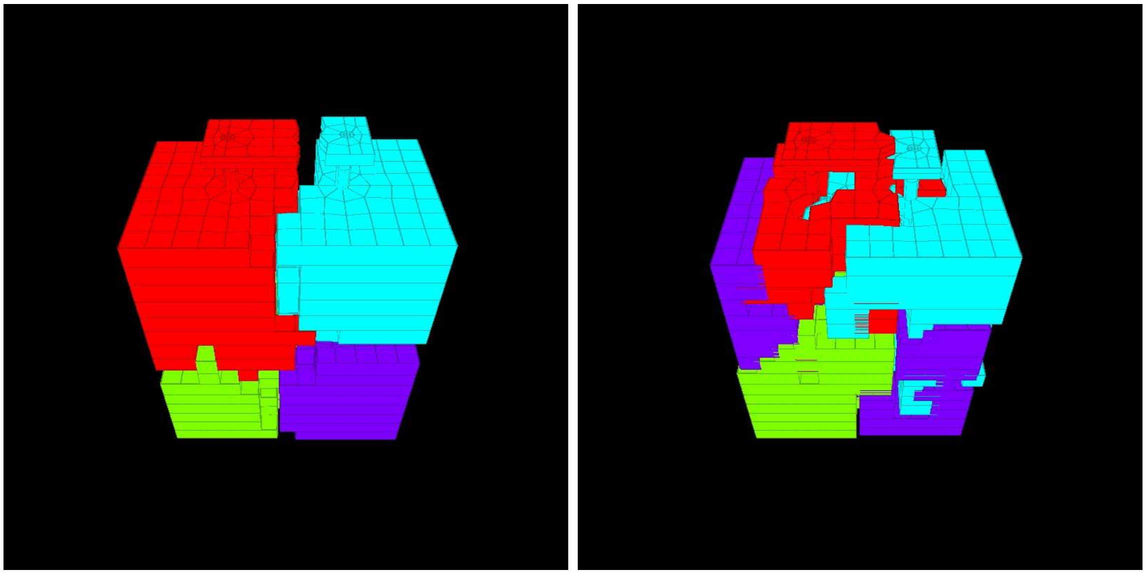


Figure 110.9: Partition and Repartition on 4 CPUs (ITR=1e-3, Imbal. Tol. 5%), FE Model (1,968 Elements, 7,500 DOFs)

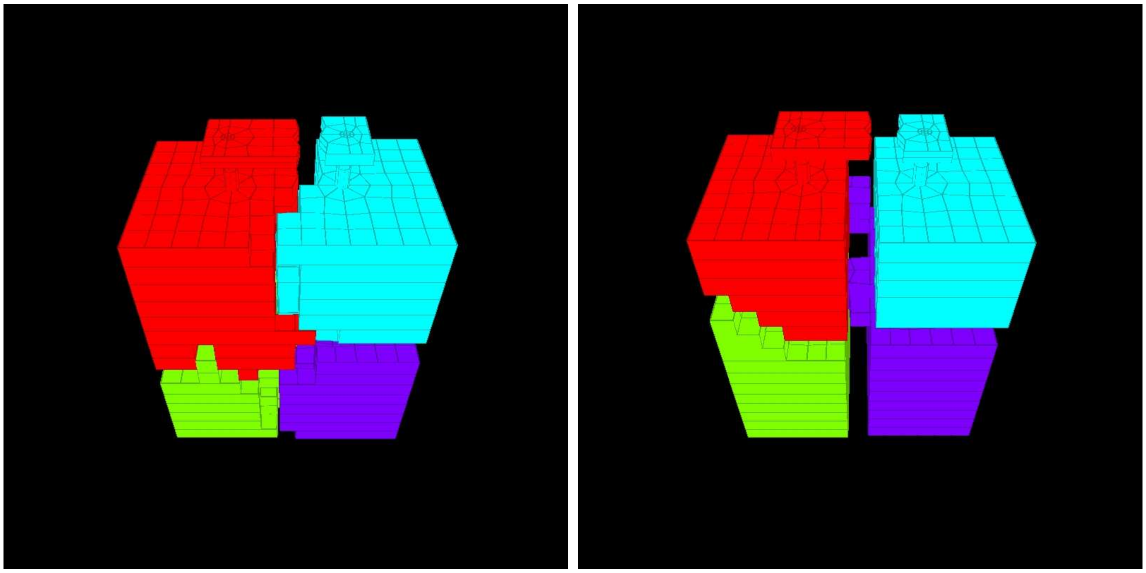


Figure 110.10: Partition and Repartition on 4 CPUs (ITR=1e6, Imbal. Tol. 5%), FE Model (1,968 Elements, 7,500 DOFs)

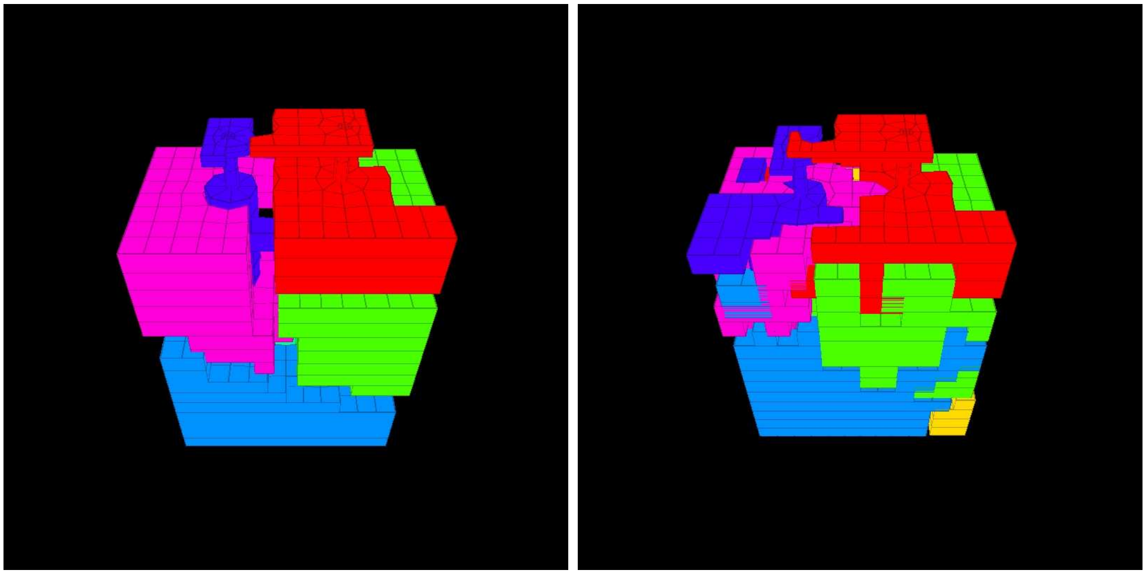


Figure 110.11: Partition and Repartition on 7 CPUs (ITR=1e-3, Imbal. Tol. 5%), FE Model (1,968 Elements, 7,500 DOFs)

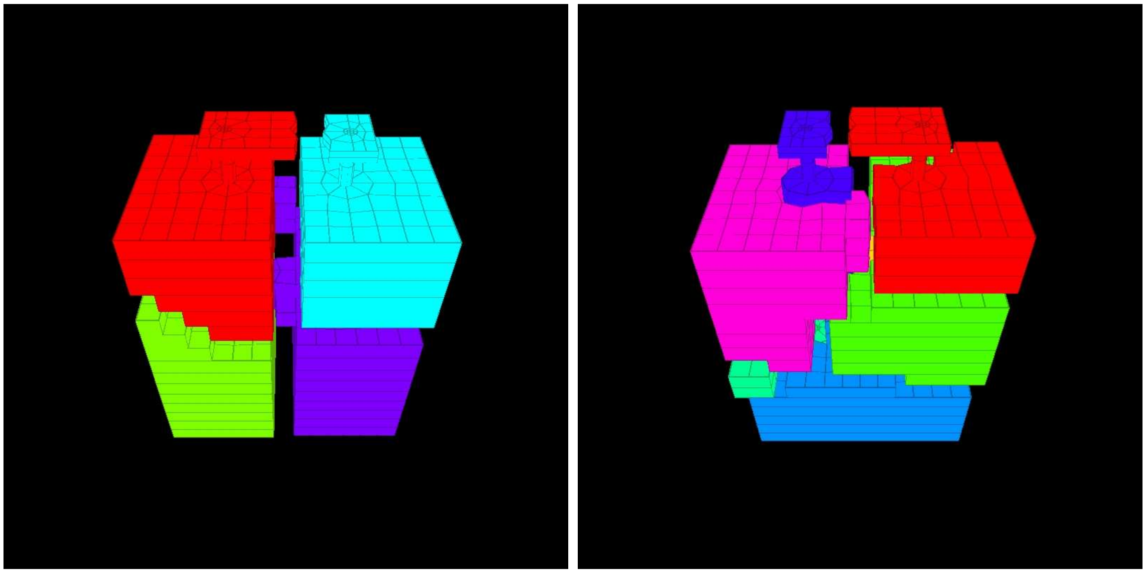


Figure 110.12: Partition and Repartition on 7 CPUs (ITR=1e6, Imbal. Tol. 5%), FE Model (1,968 Elements, 7,500 DOFs)

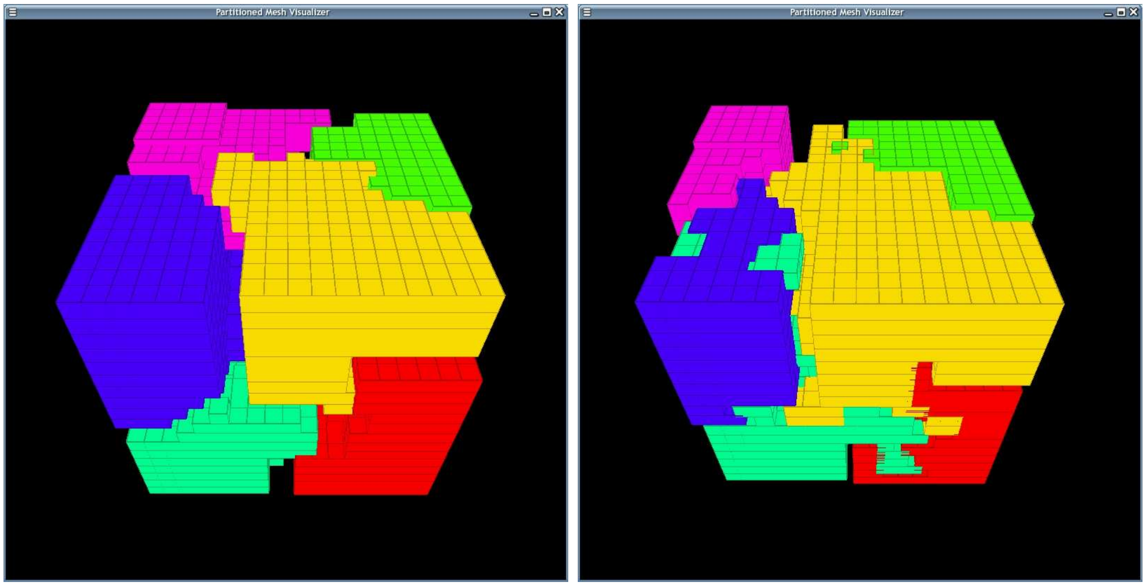


Figure 110.13: Partition and Repartition on 7 CPUs (ITR=1e-3, Imbal. Tol. 5%), FE Model (4,938 Elements, 17,604 DOFs)

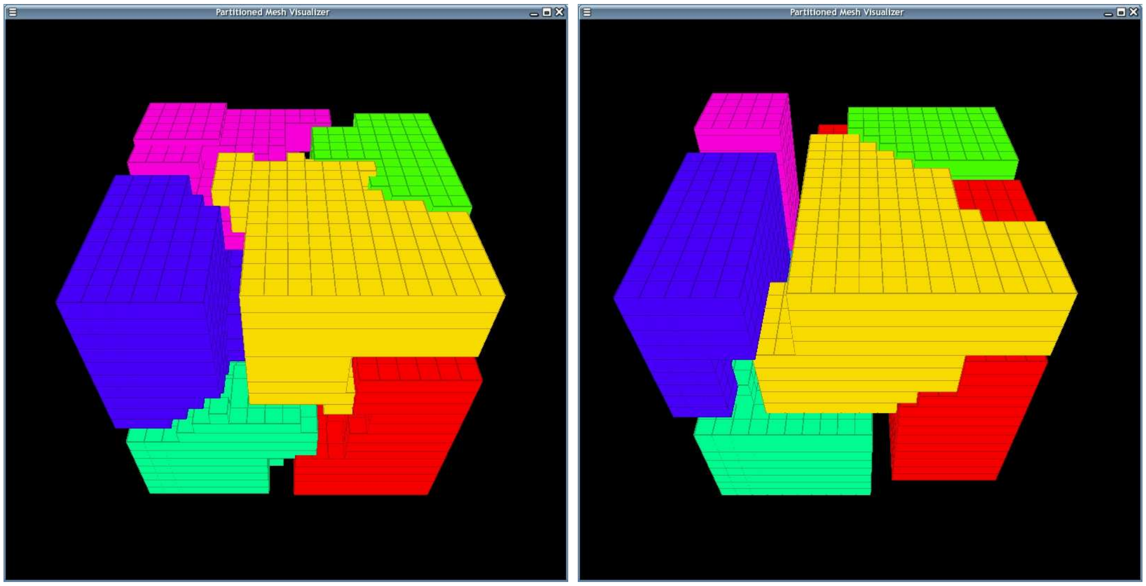


Figure 110.14: Partition and Repartition on 7 CPUs (ITR=1e6, Imbal. Tol. 5%), FE Model (4,938 Elements, 17,604 DOFs)

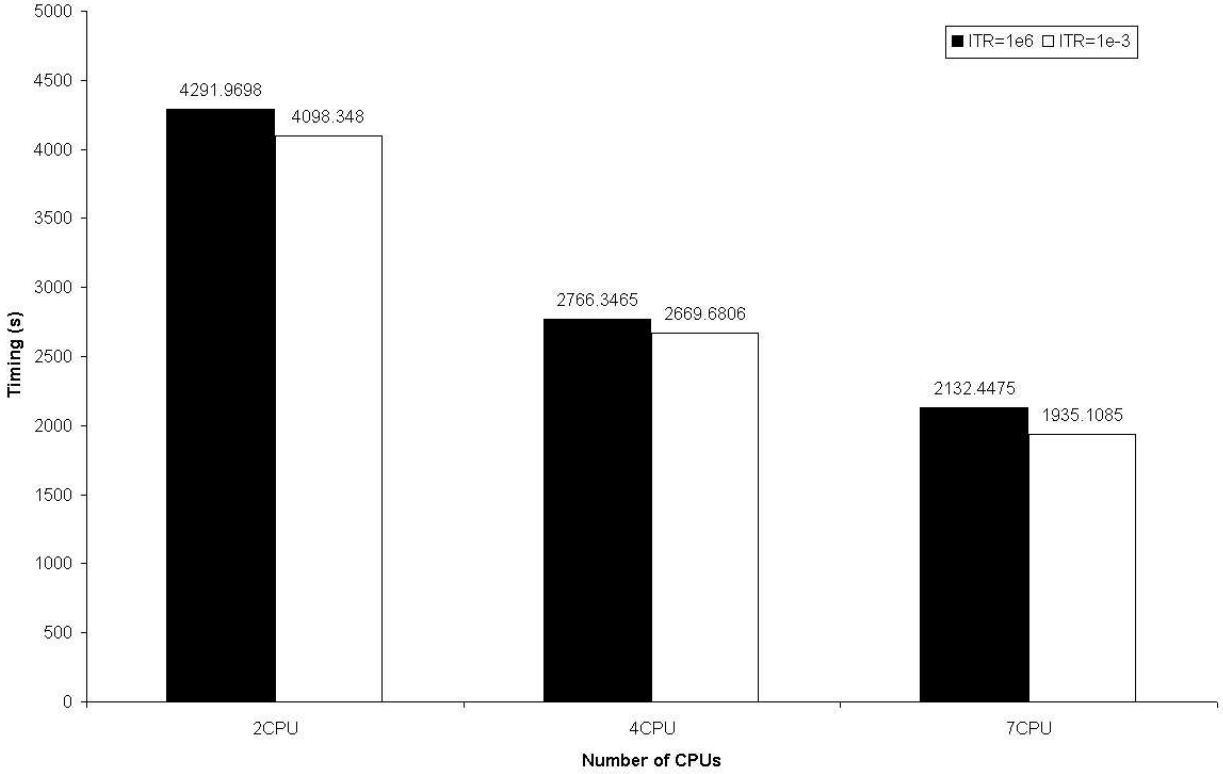


Figure 110.15: Timing Data of ITR Parametric Studies (1,968 Elements, 7,500 DOFs, Imbal. Tol. 5%)

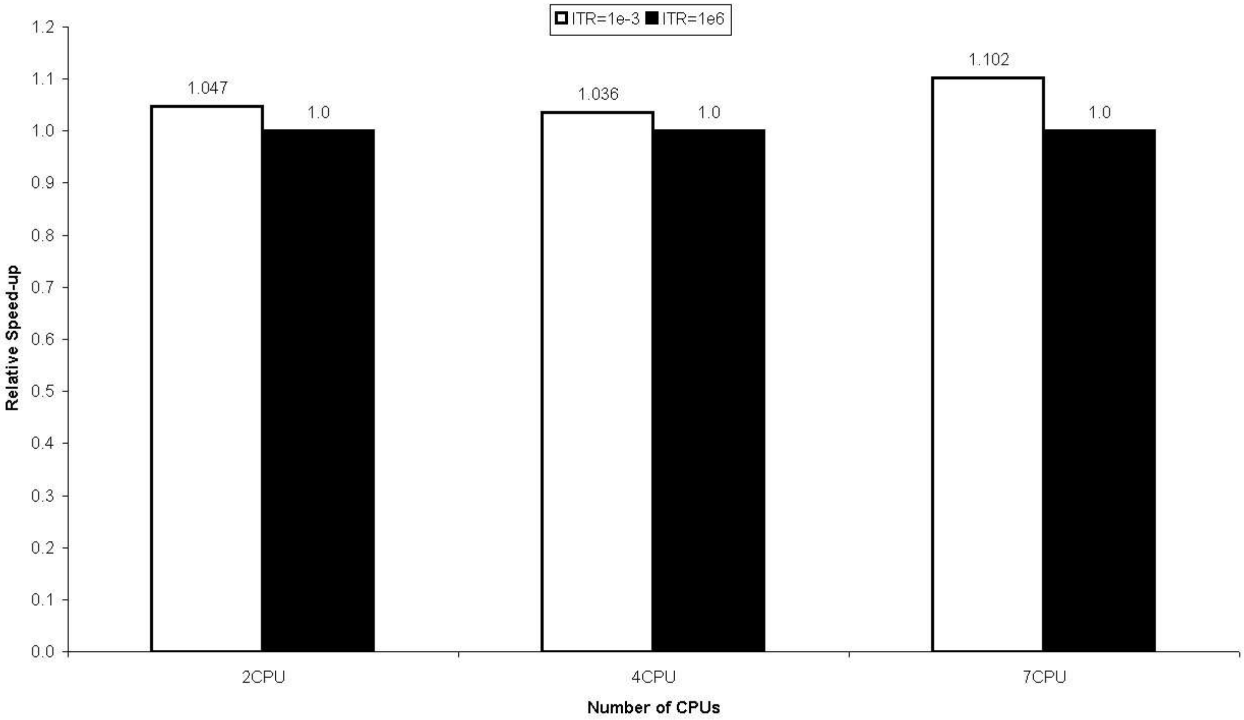


Figure 110.16: Relative Speedup of ITR=1e-3 over ITR=1e6 (1,968 Elements, 7,500 DOFs, Imbal. Tol. 5%)

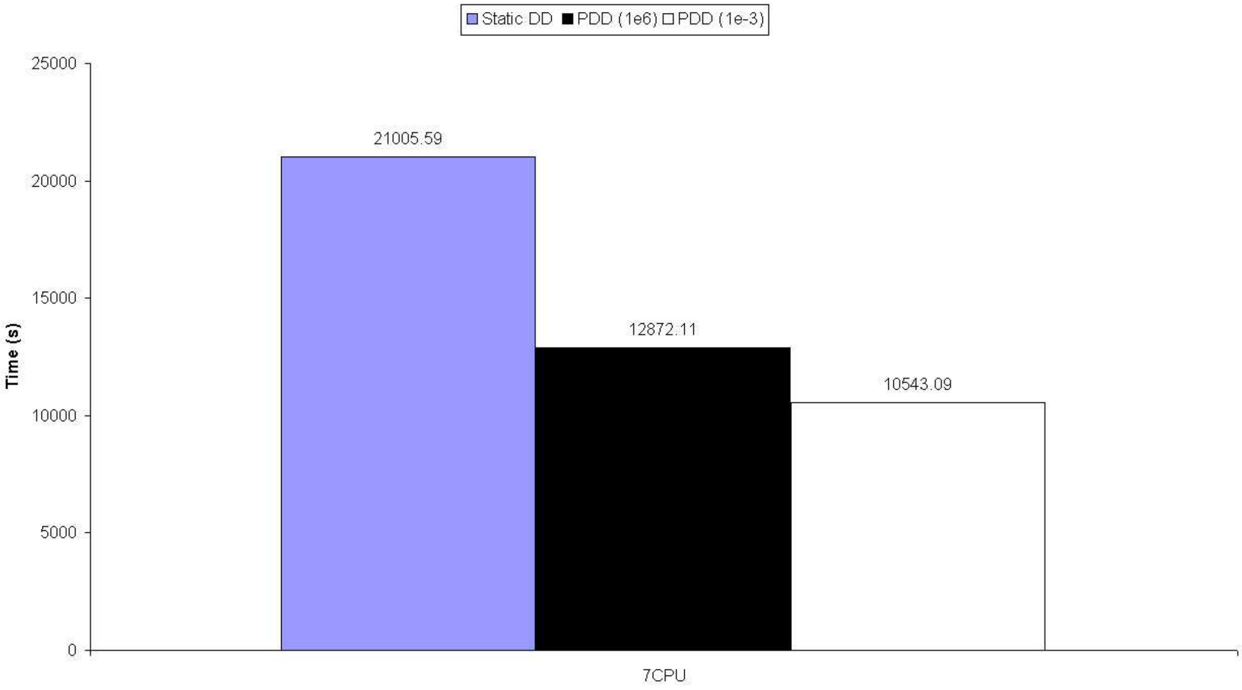


Figure 110.17: Timing Data of ITR Parametric Studies (4,938 Elements, 17,604 DOFs, Imbal. Tol. 5%)

Figures 110.15, 110.16 and 110.17 show the speedup data of parametric study on ITR factors. The purpose is to expose the the more efficient approach to do repartitioning for our specific parallel SFSI simulations, either scratch-remap approach ($ITR = 1e6$) or diffusive approach ($ITR = 1e-3$). Through the study of this chapter, some conclusions can be drawn.

1. Smaller value of ITR (1e-3) outperforms larger value (1e6). The performance gain is up to 22.1% for 7 processors. As the model gets larger, the speedup tends to get better.

2. With small ITR value, the URA algorithm tends to give results for diffusive partition/repartitioning scheme, which is good for performance for our application in overall due to the fact that the overhead associated with data redistribution in this research is very high. Diffusive approach minimizes possible data movement thus delivers better performance. The drawback is the diffusive approach typically gives very bad or even disconnected graphs with very high edge-cut as shown in Figures 110.7, 110.9 and 110.11. So careful attention must be paid to these graph structures when programing the finite element calculation. In this sense, the diffusive algorithm is not as robust as scratch/remapping. One very important observation was that repetitive repartitionings tend to yield totally ill-connected graph.

3. With large ITR value, the URA algorithm adopts the scratch/remapping scheme which inevitably introduce huge data redistribution cost. But this approach gives high quality graph and the integrity of original graph is well preserved as shown Figures 110.8, 110.10 and 110.12. This will be of great meaning for parallel finite element method based on substructure-type methods. Another important observation was, the scratch/remapping approach performed much more repartitionings than diffusive approach for same analysis. Repetitive repartitionings by scratch/remapping method tends to totally migrate all elements out of their initial partitioning and repartitioning never stops even though the computation is stabilized (in the sense of formation of plastic zones). This also explains in part why the diffusive approach can substantially outperform scratch/remapping.

4. Based on the timing analysis performed in this chapter, ITR=1e-3 is the best choice that brings substantially better performance over large ITR values. With the increase of number of processing units or the model size, the performance gain is more significant as shown in Figures 110.15, 110.16 and 110.17. Robustness of the diffusive approach has not caused much trouble in our application.

110.4.5 Parallel Performance Analysis

Timing routines have been implemented in PDD (MOSS and other used libraries, such as Template3DEP/NewTemplate3D to study the parallel performance. The preprocessing unit, like reading model data from file, has not been timed so the speed up here reflects only algorithmic gain by graph partitioning. In the current phase of this research, the equation solving problem has not been addressed yet. More meaningful perspective would be to consider performance gains by simply switching from plain graph partitioning to adaptive graph partitioning, which is also the basic aim of this research. As we can see from the results below, adaptive graph partitioning improves the overall performance of elasto-plastic finite element computations. The partitioning/repartitioning overhead has been minimized by using parallel partitioner.

As stated in previous sections of this chapter, there are a couple of key parameters that control performance of the adaptive load balancing algorithm. One is the ITR factor, and the other is the computational load imbalance tolerance.

1. ITR is the key parameter which determines the algorithmic approach of the adaptive load balancing scheme. Depending on different applications and network interconnections, this value can be set to very small (0.001) or very large (up to 1,000,000) and algorithm focus will be set to minimizing data redistribution or edge-cut respectively as explained in previous sections.
2. Computational Imbalance Load Tolerance is the other key factor affecting greatly the overall performance of the whole application codes. Basically speaking, with larger finite element model, the tolerance should be set higher due to the fact that data redistribution and subsequent analysis-restarting overhead can be substantially higher as the finite element model size increases.

The performance tunings on ITR factor tend to yield consistent results as stated previously that smaller ITR (0.001) brings better performance over large ITR values. Diffusive repartitioning algorithm outperforms scratch/remapping in our application.

While on the other hand, tuning on load imbalance studies has been more illusive. The first conclusion is that load imbalance tolerance larger than 5% was not able to work robustly as the size of finite element model increases in the application study of this chapter.

Detailed parametric studies have been performed on DataStar IBM Power4 and IA64 Intel clusters to indicate the effectiveness of the proposed adaptive PDD algorithm. Models with different sizes have been tested on various number of processors to show the scalability of computational performance. All results will be compared with static one-step Domain Decomposition approach to investigate the advantage of proposed PDD algorithm in nonlinear elastic-plastic finite element calculations.

In the following sections, timing data and partition/repartition figures will be presented and results will be discussed at the end of this chapter.

110.4.5.1 Soil-Foundation Model with 4,035 DOFs

The partition/repartition figures by PDD have been shown in Figure 110.18, 110.19, 110.20.

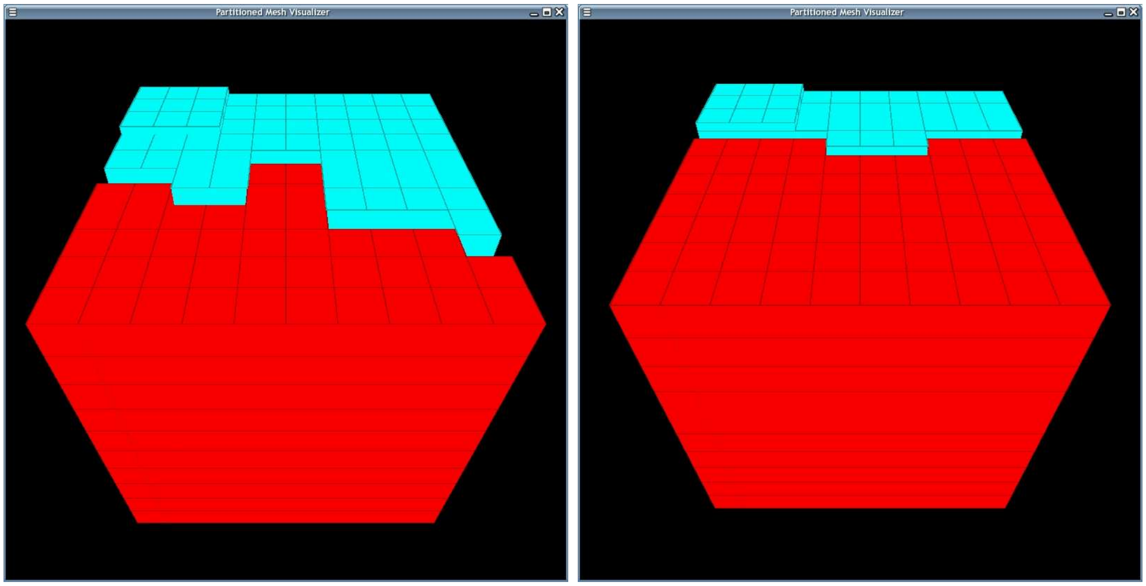


Figure 110.18: 4,035 DOFs Model, 2 CPUs, ITR=1e-3, Imbal Tol 5%, PDD Partition/Repartition

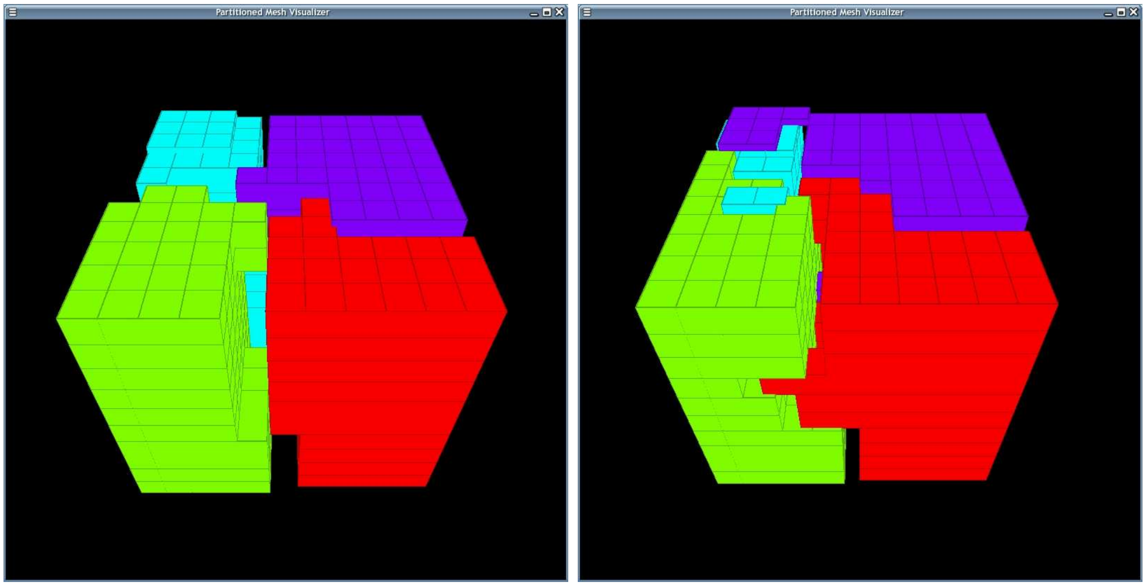


Figure 110.19: 4,035 DOFs Model, 4 CPUs, ITR=1e-3, Imbal Tol 5%, PDD Partition/Repartition

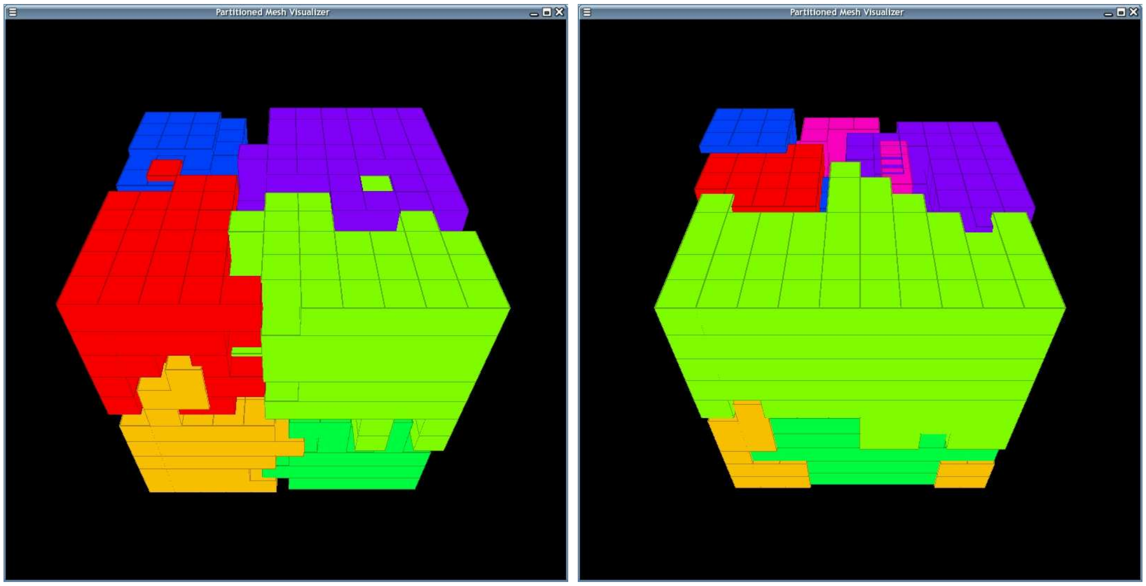


Figure 110.20: 4,035 DOFs Model, 8 CPUs, ITR=1e-3, Imbal Tol 5%, PDD Partition/Repartition

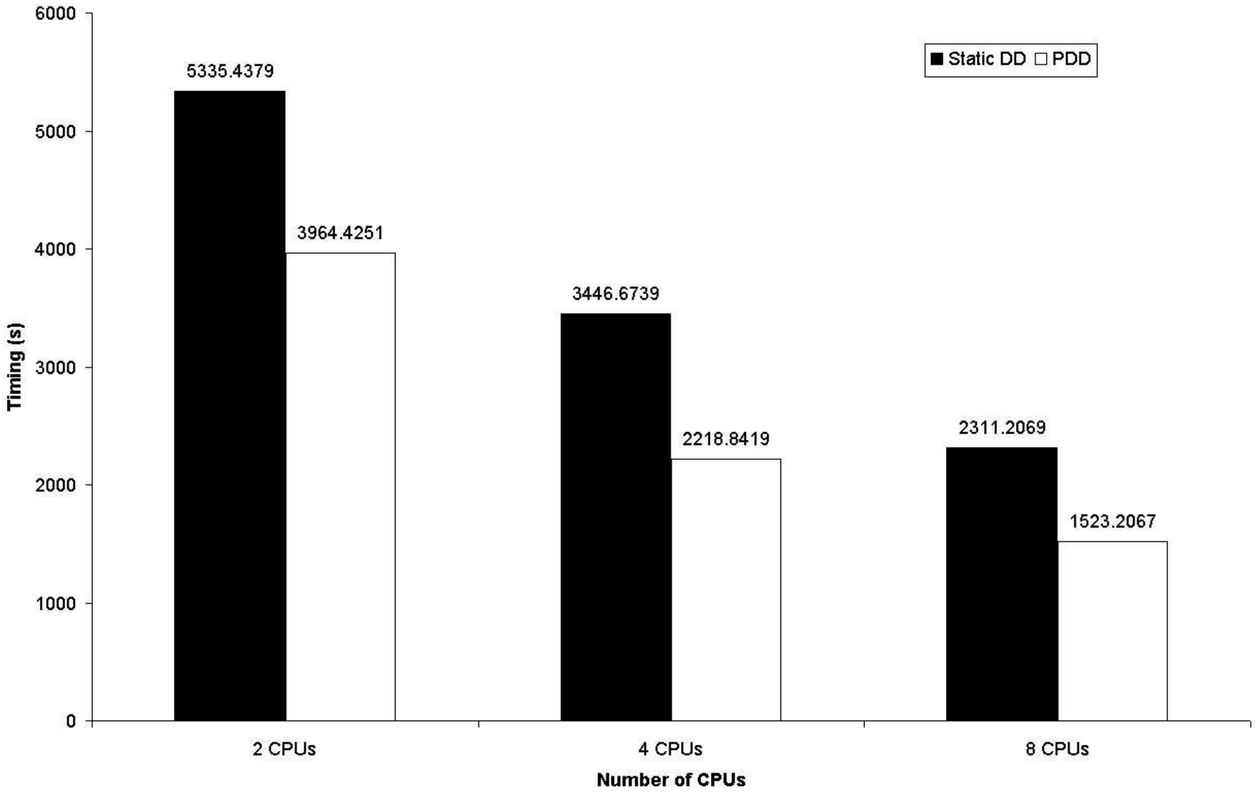


Figure 110.21: Timing Data of Parallel Runs on 4,035 DOFs Model, ITR=1e-3, Imbal Tol 5%

Table 110.4: Test Cases of Performance Studies

Model Sizes (DOF)	4,035, 17,604, 32,091, 68,451
# of CPUs	3, 5, 7, 16, 32, 64
ITR Factors	0.001, 1,000,000
Imbalance Tolerance	5%, 10%, 20%

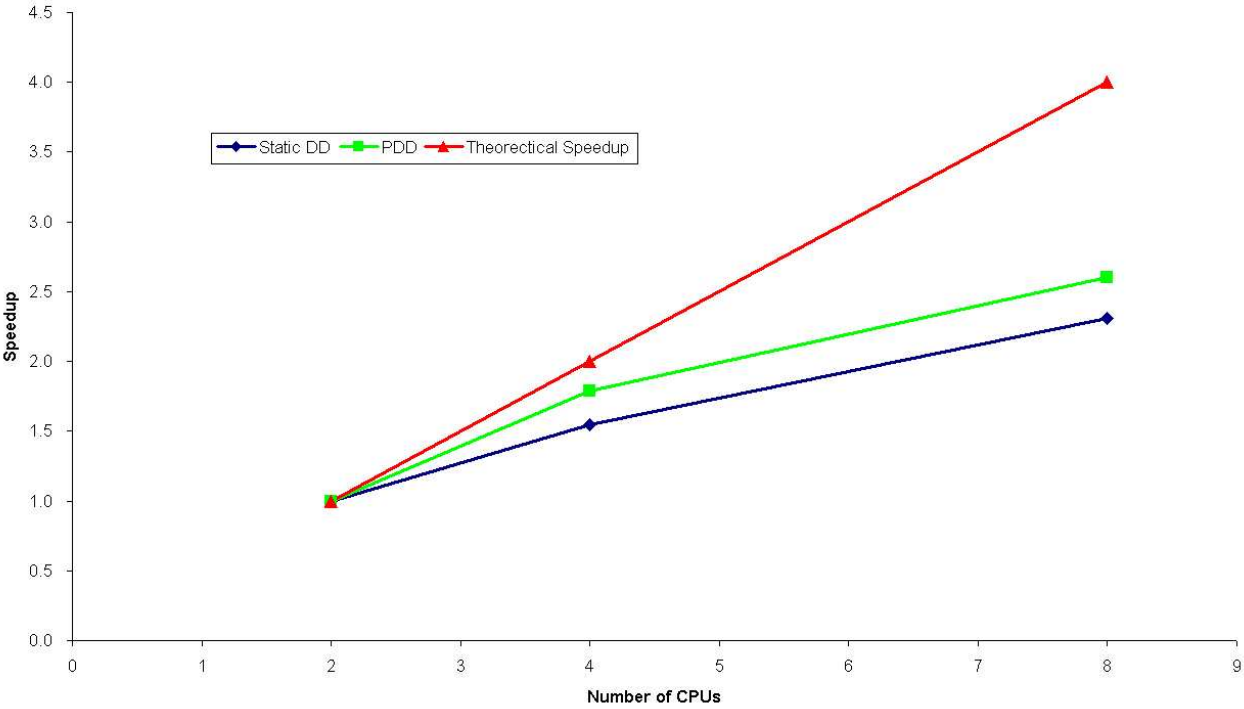


Figure 110.22: Absolute Speedup Data of Parallel Runs on 4,035 DOFs Model, ITR=1e-3, Imbal Tol 5%

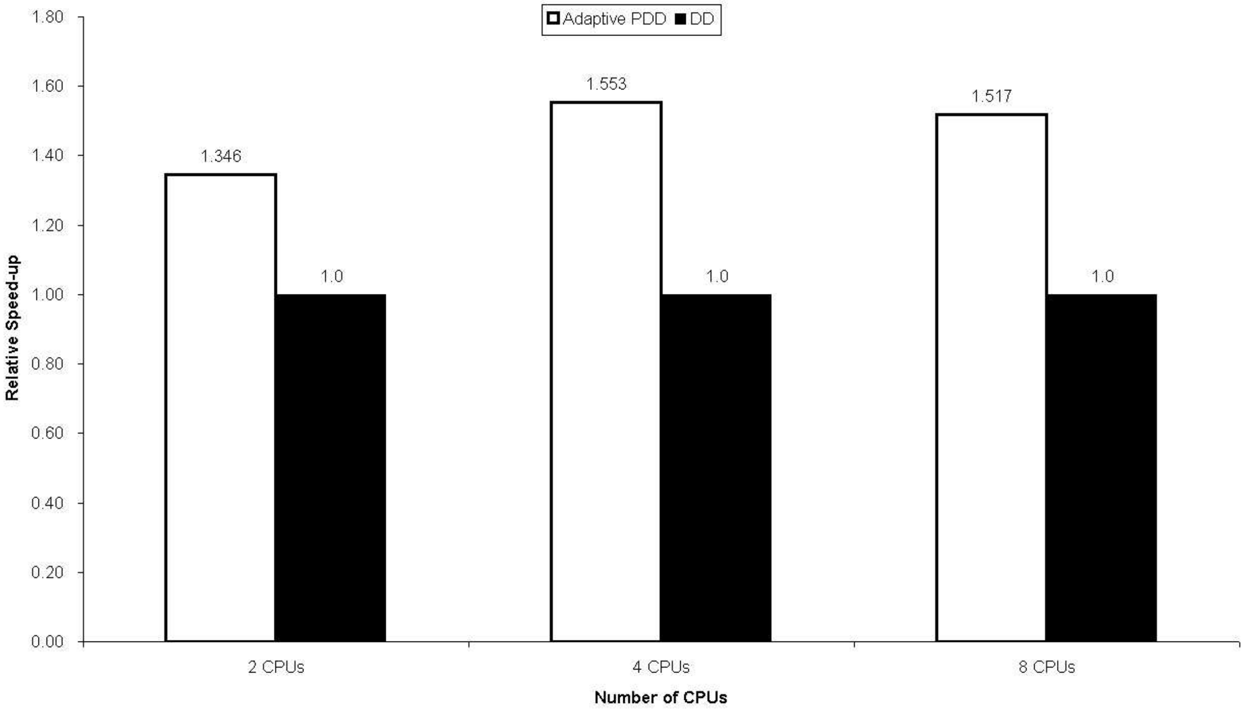


Figure 110.23: Relative Speedup of PDD over Static DD on 4,035 DOFs Model, ITR=1e-3, Imbal Tol 5%

110.4.5.2 Soil-Foundation Model with 4,938 Elements, 17,604 DOFs

This is the same model as described before but with more elements as shown in 110.24. Timing data has been collected to indicate performance gains by adaptive load balancing Partition and repartition figures are shown from Figure 110.28 to 110.30. The partition/repartition figures by PDD have been shown in Figure 110.28, 110.29, 110.30.

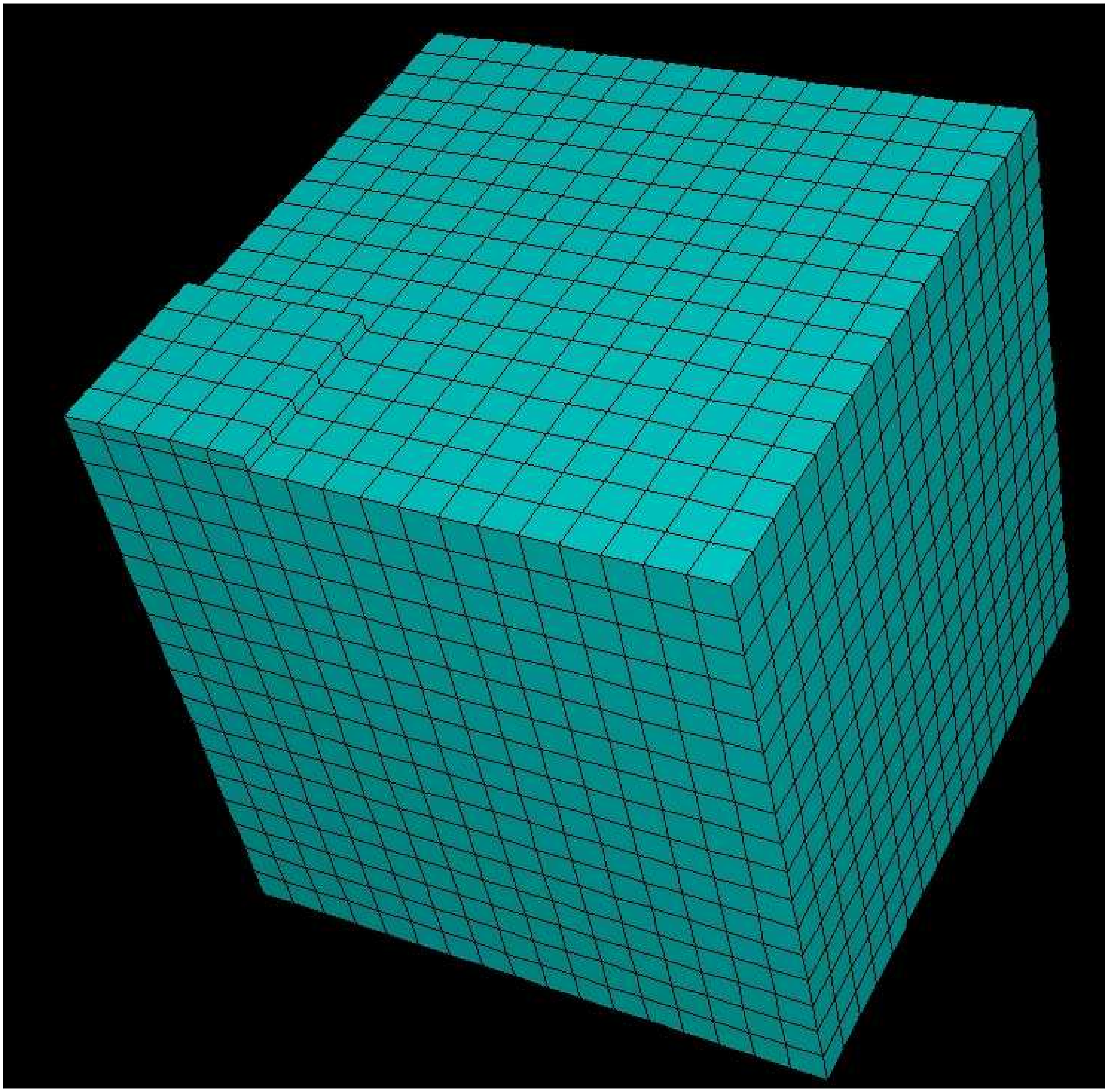


Figure 110.24: Finite Element Model of Soil-Foundation Interaction (4,938 Elements, 17,604 DOFs)

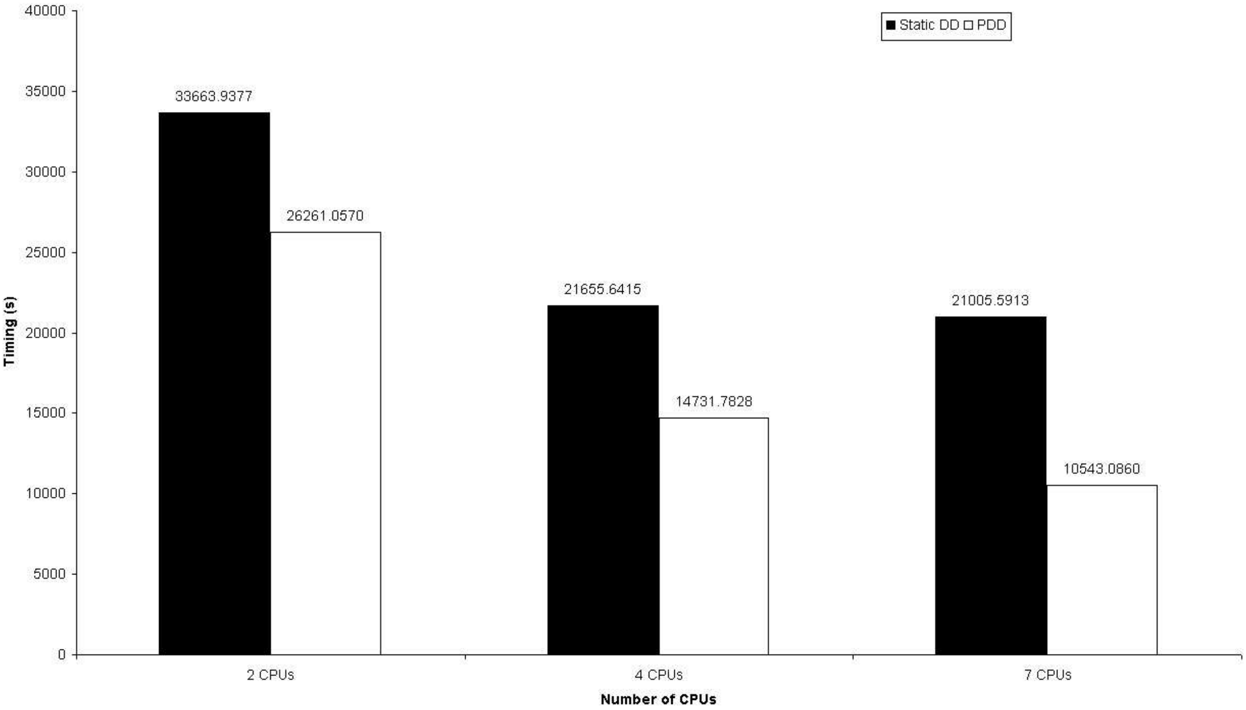


Figure 110.25: Timing Data of Parallel Runs on 4,938 Elements, 17,604 DOFs Model, ITR=1e-3, Imbal Tol 5%

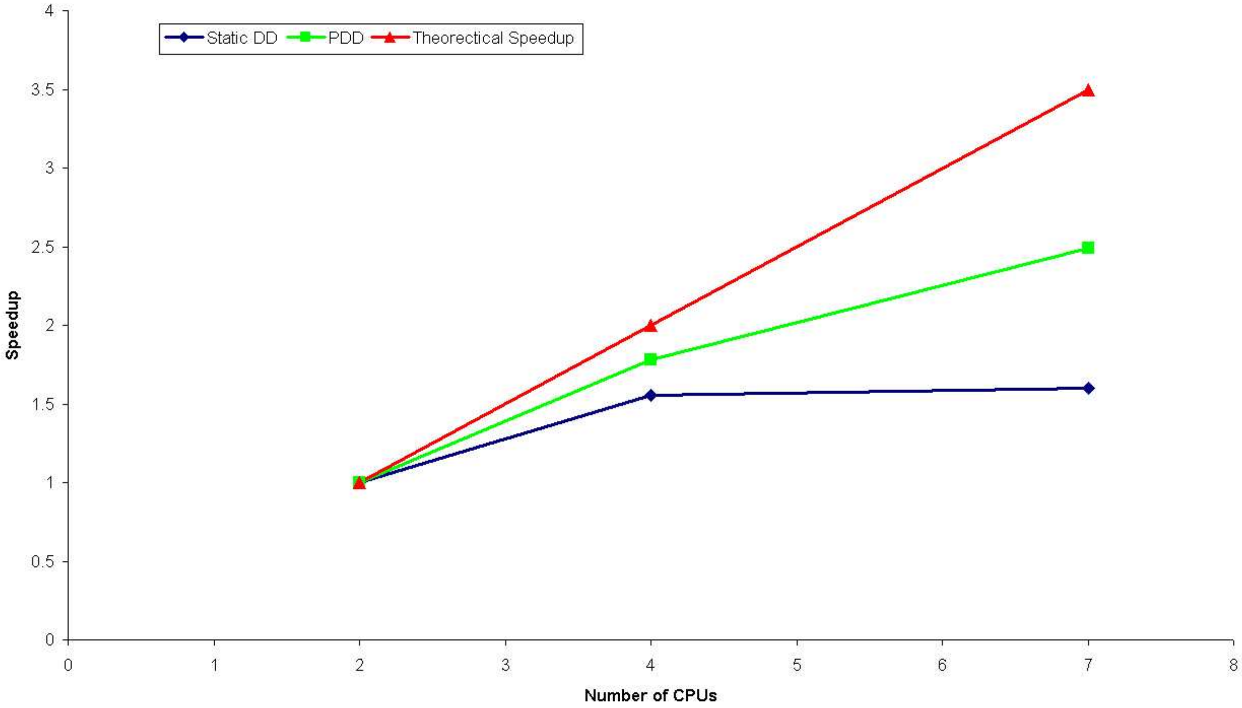


Figure 110.26: Absolute Speedup Data of Parallel Runs on 4,938 Elements, 17,604 DOFs Model, ITR=1e-3, Imbal Tol 5%

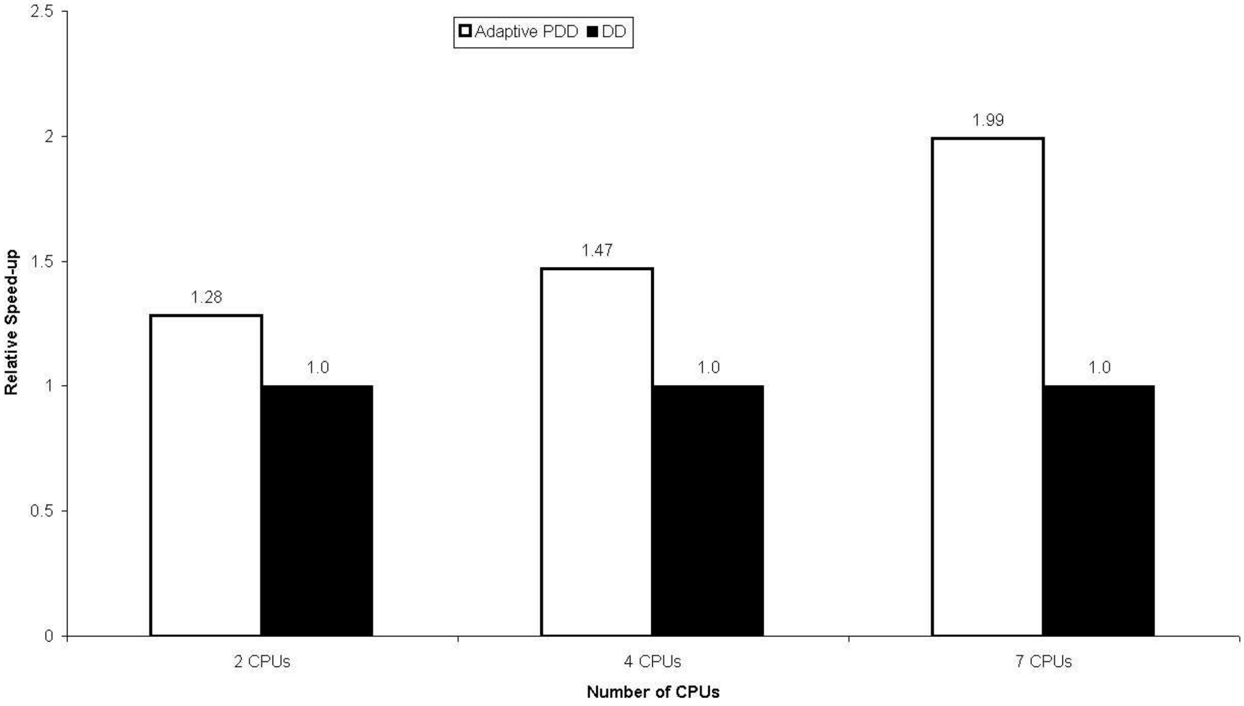


Figure 110.27: Relative Speedup of PDD over Static DD on 4,938 Elements, 17,604 DOFs Model, ITR=1e-3, Imbal Tol 5%

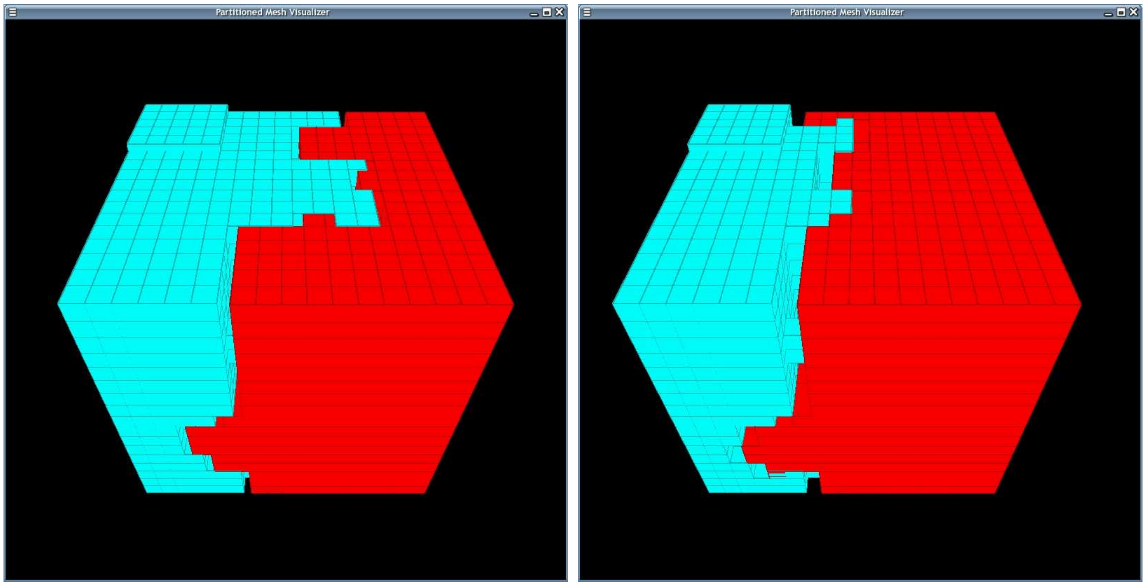


Figure 110.28: 4,938 Elements, 17,604 DOFs Model, 2 CPUs, PDD Partition/Repartition, ITR=1e-3, Imbal Tol 5%

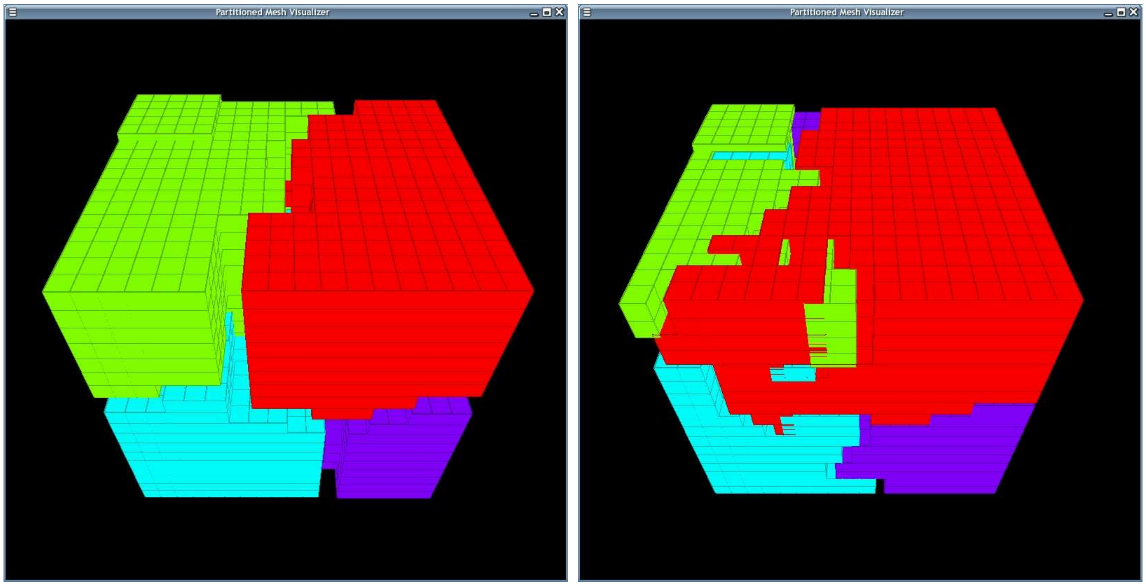


Figure 110.29: 4,938 Elements, 17,604 DOFs Model, 4 CPUs, PDD Partition/Repartition, ITR=1e-3, Imbal Tol 5%

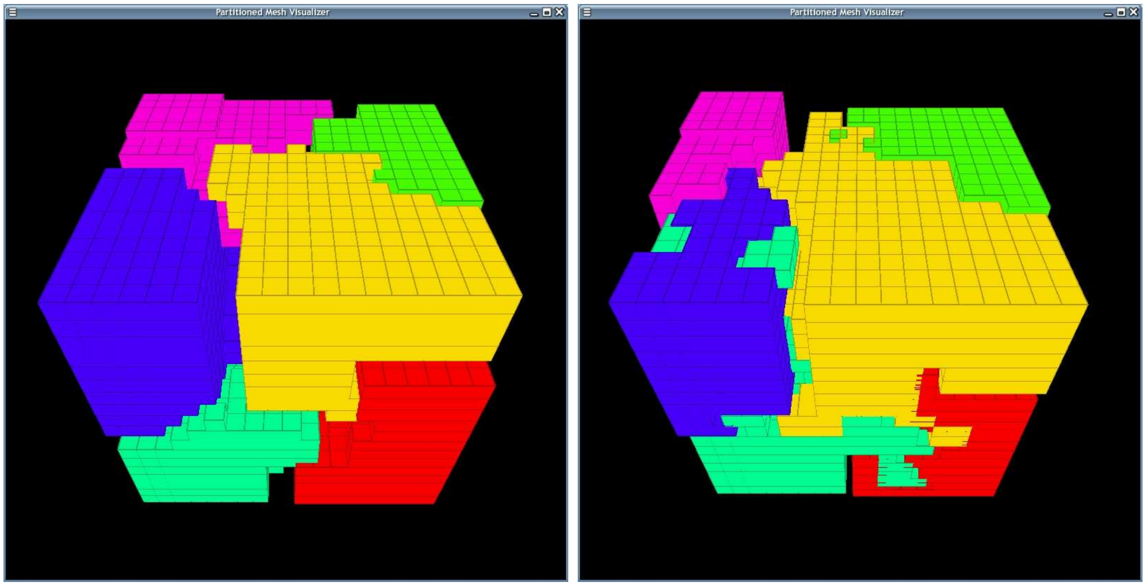


Figure 110.30: 4,938 Elements, 17,604 DOFs Model, 8 CPUs, PDD Partition/Repartition, ITR=1e-3, Imbal Tol 5%

110.4.5.3 Soil-Foundation Model with 9,297 Elements, 32,091 DOFs

The mesh is shown in Figure 110.31. Speed up results are shown from Figure 110.32 to Figure 110.34. Partition and repartition figures are shown from Figure 110.35 to Figure 110.39.

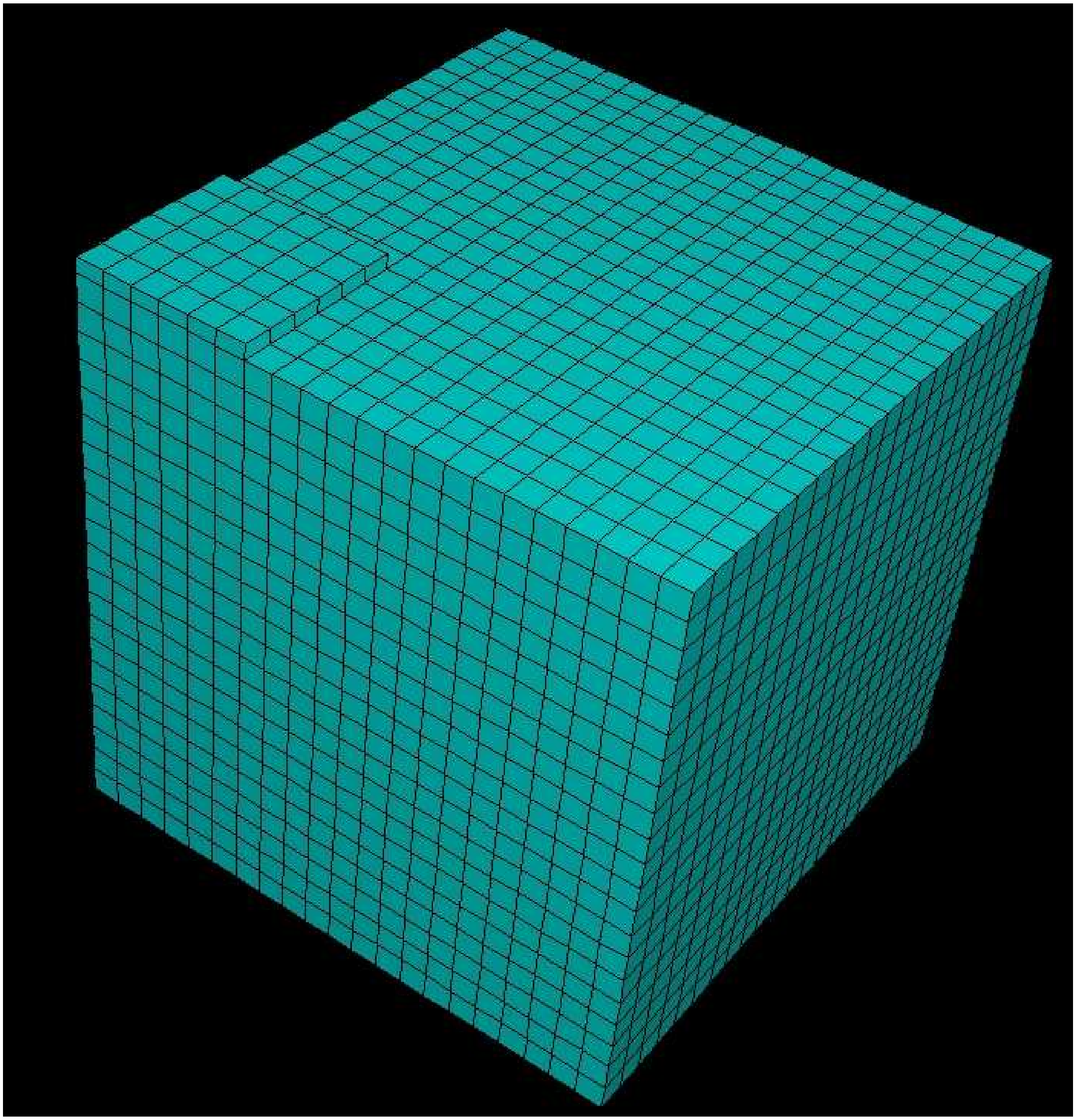


Figure 110.31: Finite Element Model of Soil-Foundation Interaction (9,297 Elements, 32,091 DOFs)

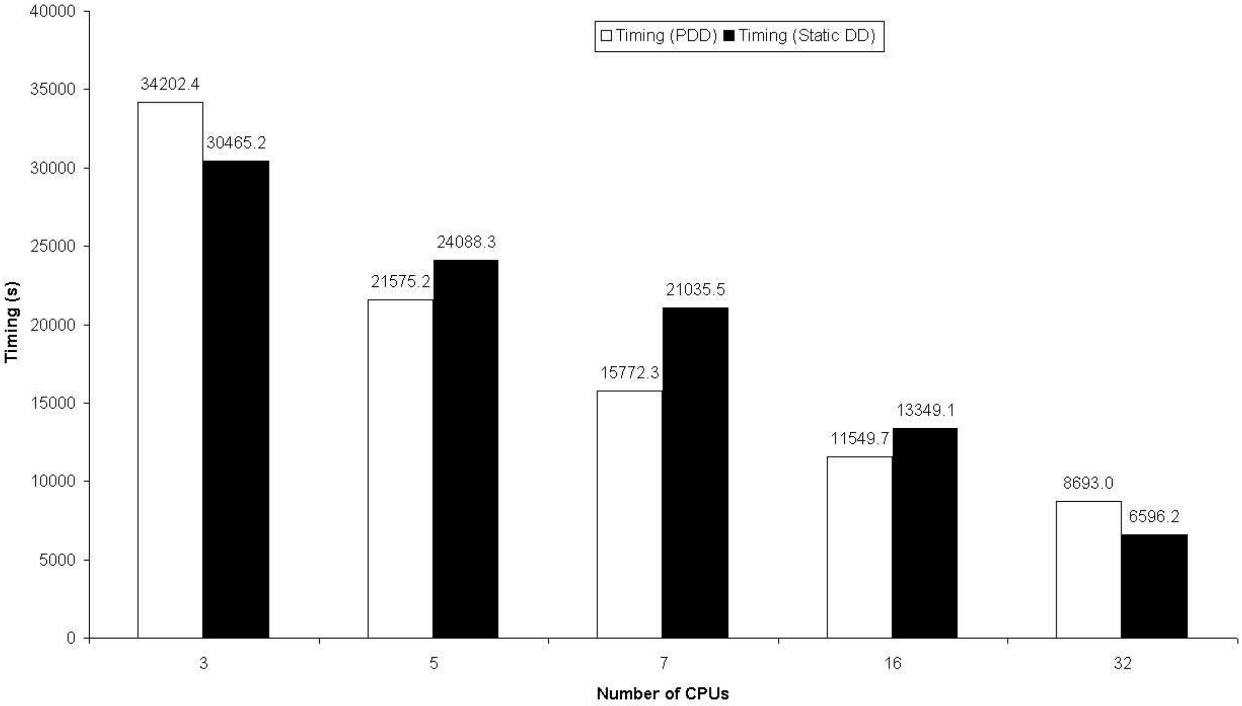


Figure 110.32: Timing Data of Parallel Runs on 9,297 Elements, 32,091 DOFs Model, ITR=1e-3, Imbal Tol 5%

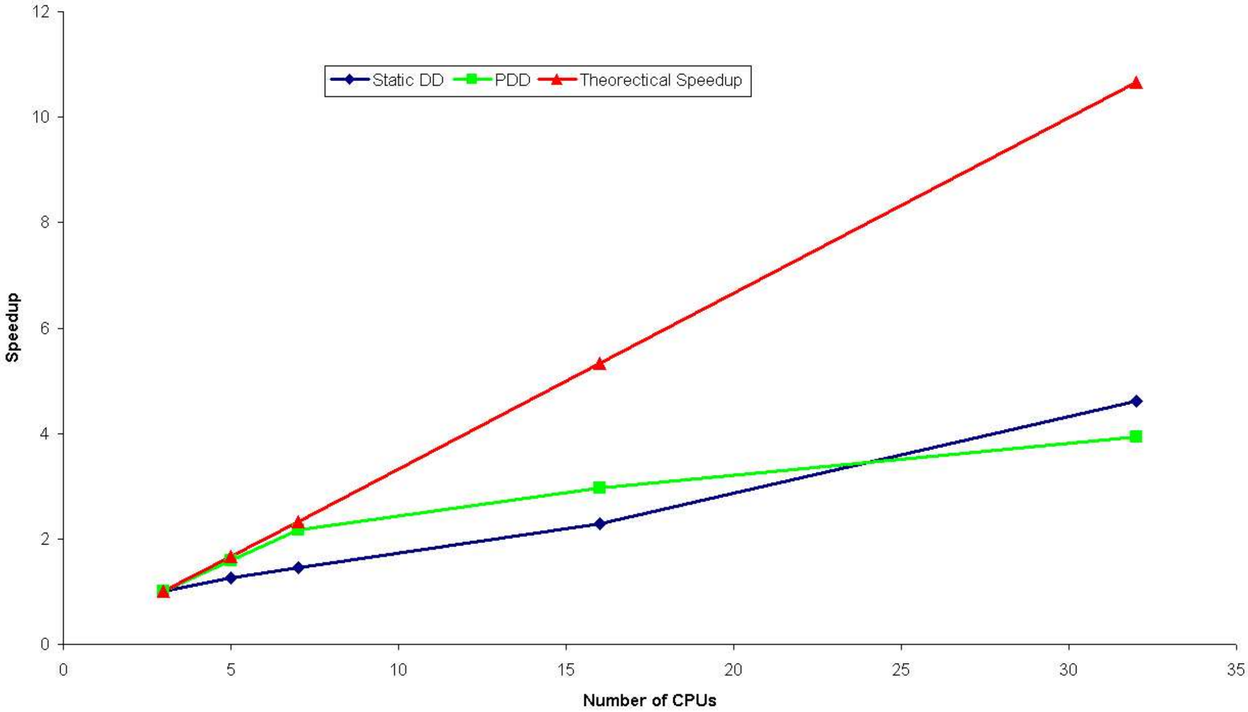


Figure 110.33: Absolute Speedup Data of Parallel Runs on 9,297 Elements, 32,091 DOFs Model, ITR=1e-3, Imbal Tol 5%

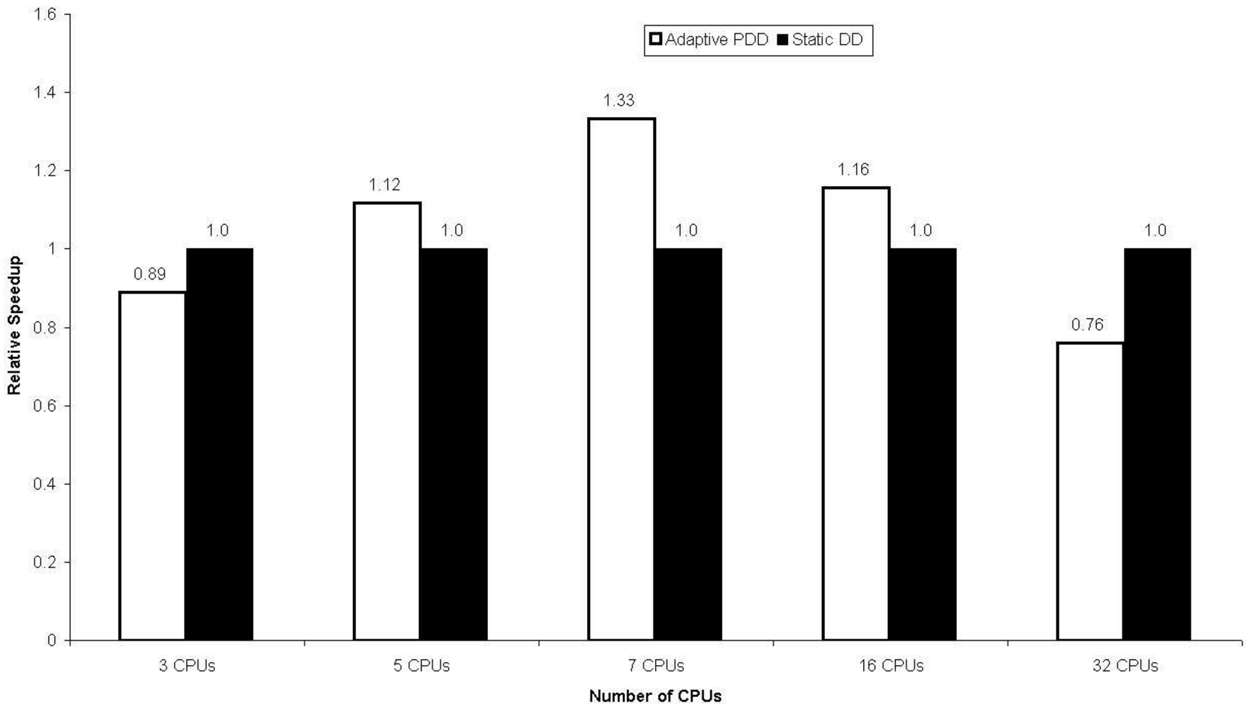


Figure 110.34: Relative Speedup of PDD over Static DD on 9,297 Elements, 32,091 DOFs Model, ITR=1e-3, Imbal Tol 5%

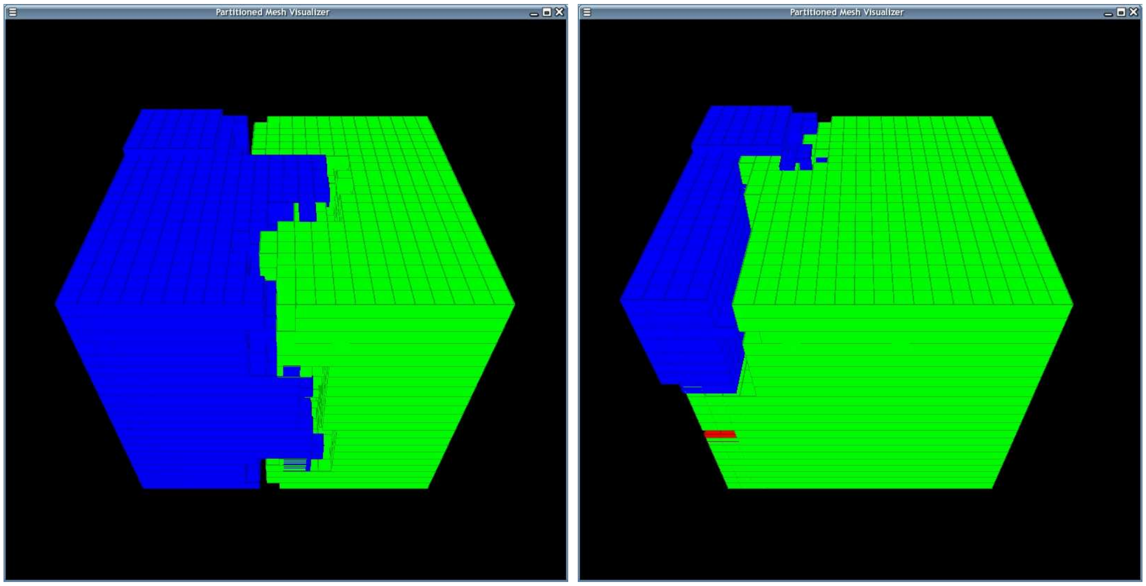


Figure 110.35: 9,297 Elements, 32,091 DOFs Model, 3 CPUs, PDD Partition/Repartition, ITR=1e-3, Imbal Tol 5%

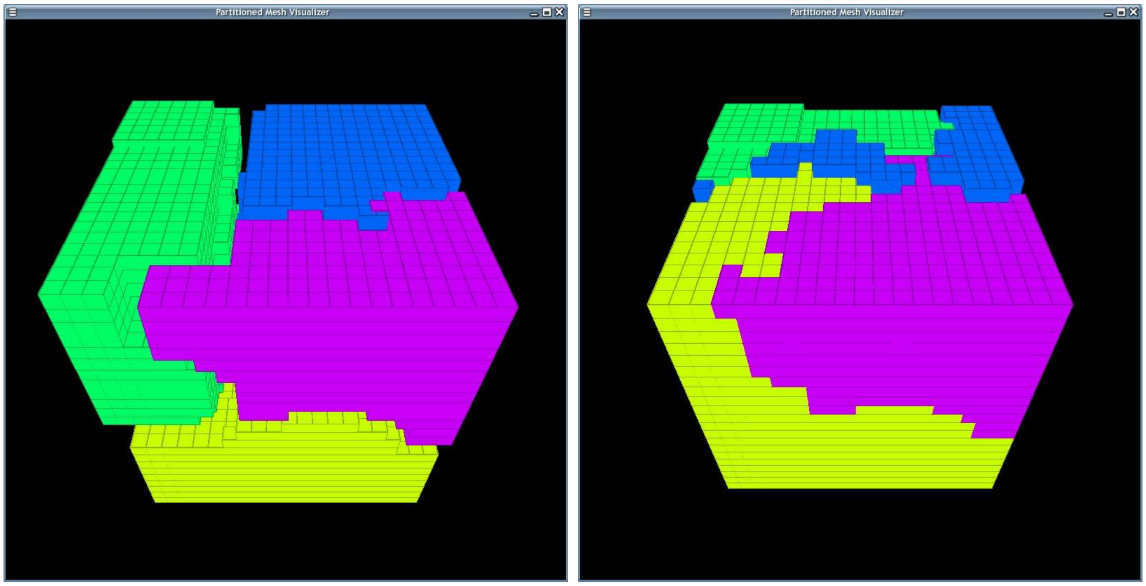


Figure 110.36: 9,297 Elements, 32,091 DOFs Model, 5 CPUs, PDD Partition/Repartition, ITR=1e-3, Imbal Tol 5%

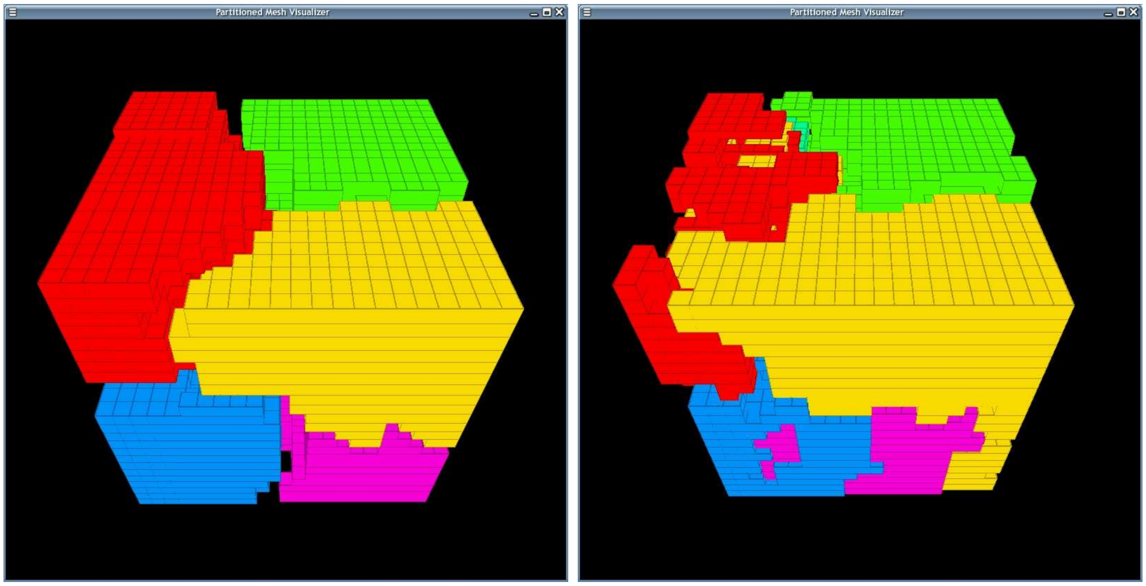


Figure 110.37: 9,297 Elements, 32,091 DOFs Model, 7 CPUs, PDD Partition/Repartition, ITR=1e-3, Imbal Tol 5%

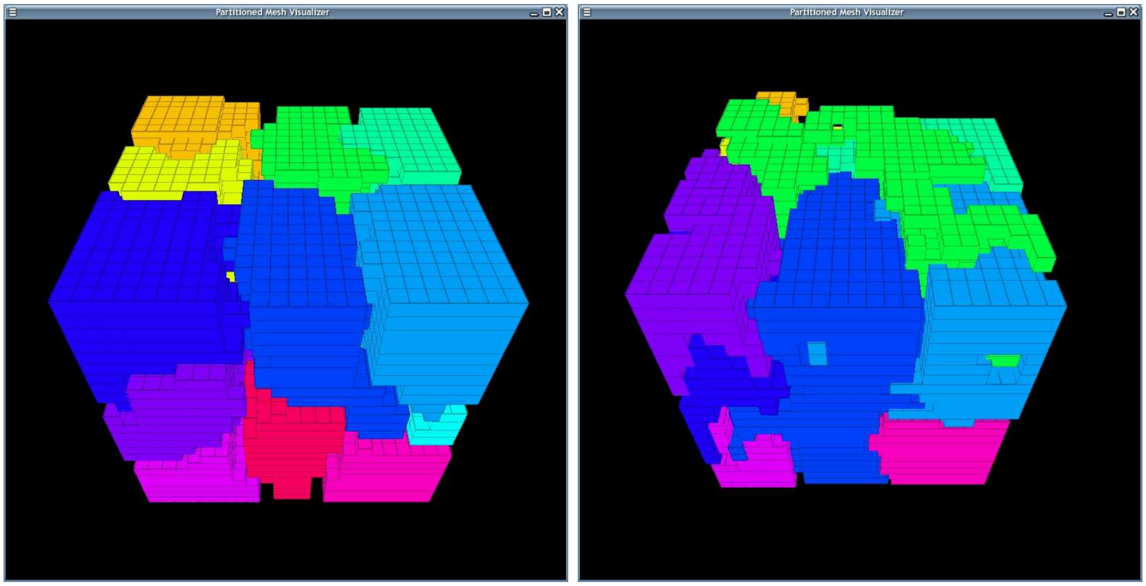


Figure 110.38: 9,297 Elements, 32,091 DOFs Model, 16 CPUs, PDD Partition/Repartition, ITR=1e-3, Imbal Tol 5%

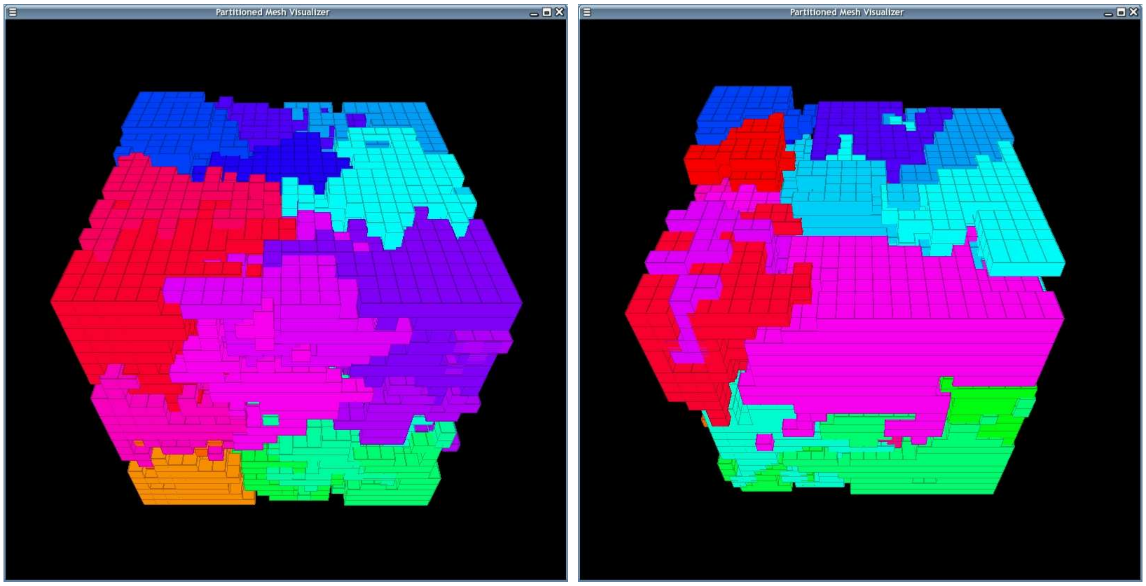


Figure 110.39: 9,297 Elements, 32,091 DOFs Model, 32 CPUs, PDD Partition/Repartition, ITR=1e-3, Imbal Tol 5%

110.4.6 Algorithm Fine-Tuning

From performance analysis results in previous sections, it has been shown that adaptive graph partitioning algorithm based on element graph can improve overall load balance for nonlinear elastic-plastic finite element calculations. Speed up has been observed on example problems. While on the other hand, we can also see as the model size increases, the efficiency of proposed PDD algorithm dropped sharply as shown in Figures 110.33 and 110.34.

So the naive implementation of PDD does not work as expected. With load balancing, one expects that the performance of PDD should not be worse than the DD case. It otherwise implies that the PDD does not bring performance gain that can completely offset its own extra load balancing operations-related overheads.

In this chapter, more detailed algorithm fine-tuning has been performed to address the problems we had in previous sections of the naive PDD implementation.

In order to improve the overall efficiency of proposed PDD algorithm. we have to consider two levels of costs when one wishes to balance the computational load among processing units. One is the data communication cost, and the other one is finite element model regeneration overhead associated with specific application problems.

Currently the adaptive graph partitioning algorithm does not consider the fact that the network communication patterns might differ much among processing nodes. The single ITR value indicates the algorithmic approach of the graph partitioning algorithm, but the real communication performance has not been addressed in the implementation.

On the other hand, certain applications impose extra problem-dependent overhead to repartitioning operations. For example, whenever data communications happen, the finite element model has to be wiped off and regenerated. This is not inherent with the graph partitioning algorithm but still needs to be addressed in order to get the best performance. As observed in this chapter, model regeneration overhead increases when the finite element model becomes bigger.

In order to improve the overall performance of our application, we hope to consider both data communication and model regeneration cost and create a new strategy through which we can adaptively monitor the extra overheads to assure that load balancing operation can offset both costs.

This chapter will first investigate the effect of load balance tolerance on performance and then a new globally adaptive strategy will be proposed to handle both communication and model regeneration overhead. Speedup analysis have been done to show performance gains.

110.4.7 Fine Tuning on Load Imbalance Tolerance

If one finds out that the application-associated overhead (say, model regeneration cost) overwhelms when repartitioning happens, the most natural way to improve performance is to increase the load imbalance tolerance of the adaptive repartition routine. In this way, one hopes to increase the critical load imbalance that can trigger the balancing routine and so that the repartition counts can be reduced. As a result, model regeneration cost can do less harm to the overall performance.

This should rather viewed as a work-around and has not been effective in our application.

The tuning approach aims at improving efficiency of previous runs that failed showing speedup over static domain decomposition method. Shallow foundation model with 9,297 Elements, 32,091 DOFs has been chosen to study the effect of imbalance tolerance on parallel performance. Model setup has been the same as in previous sections.

Speedup analysis results have been shown in Figures 110.40, 110.41 and 110.42.

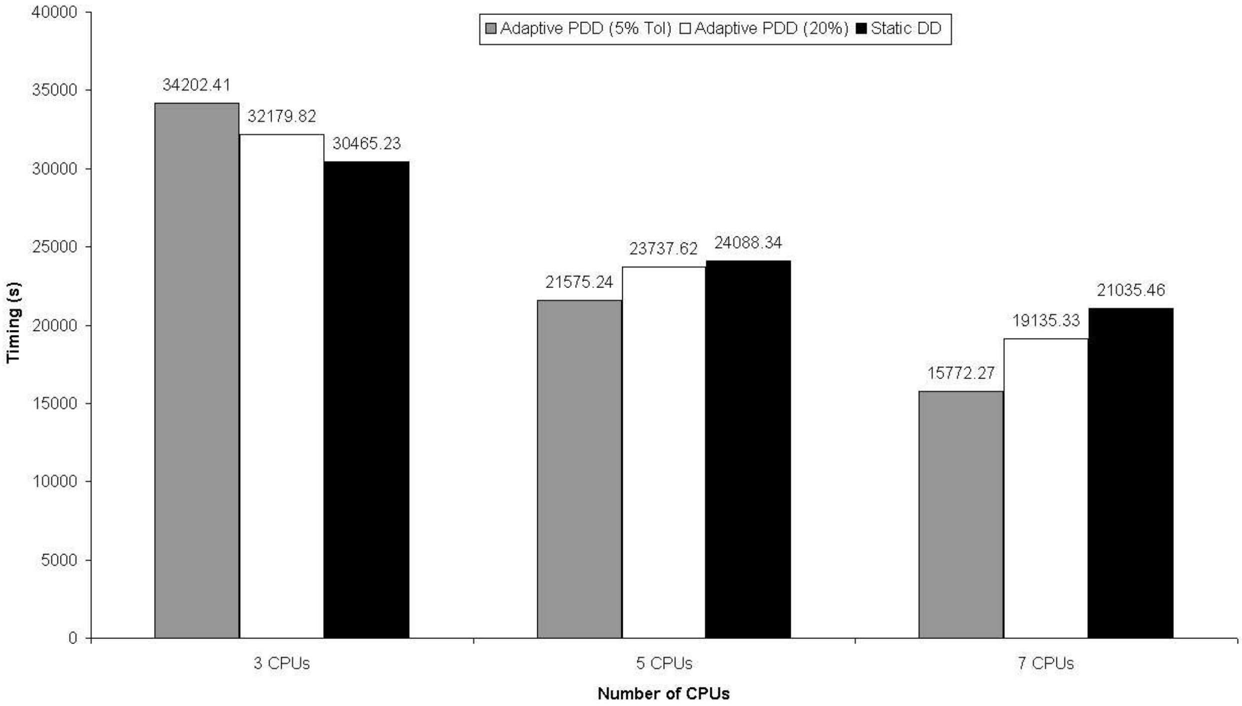


Figure 110.40: Timing Data of Parallel Runs on 9,297 Elements, 32,091 DOFs Model, ITR=1e-3, Imbal Tol 20%

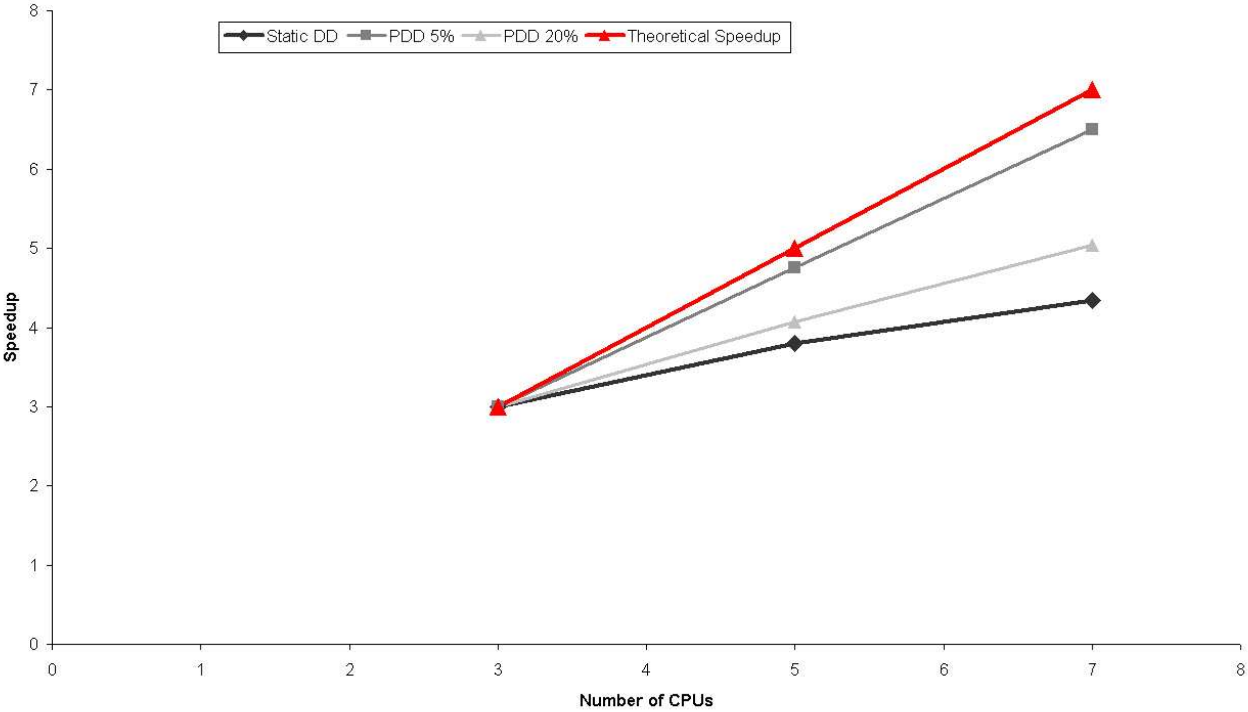


Figure 110.41: Absolute Speedup Data of Parallel Runs on 9,297 Elements, 32,091 DOFs Model, ITR=1e-3, Imbal Tol 20%

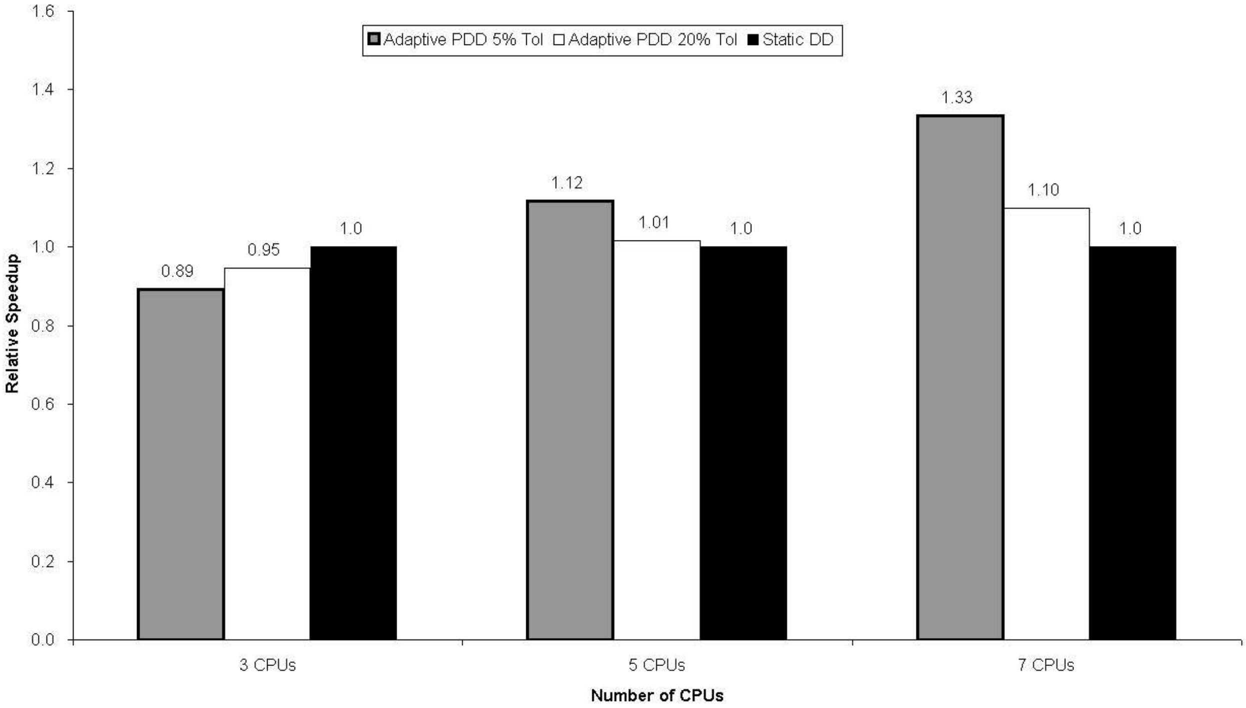


Figure 110.42: Relative Speedup of PDD over Static DD on 9,297 Elements, 32,091 DOFs Model, ITR=1e-3, Imbal Tol 20%

From the performance results, we can see that increasing load imbalance tolerance does not lead to efficiency for our application. As the number of processing units increases, the whole performance of application codes deteriorates. It is also important to note that the adaptive graph partitioning/repartitioning kernel in ParMETIS has not been capable of producing adequate partitions for finite element calculations when the load imbalance tolerance is larger than the recommended 5% Karypis et al. (2003). The application crushed with 20% imbalance tolerance for same models tested in previous sections.

The conclusion reached for the application in this chapter is that load imbalance tolerance larger than 5% has not been proved more efficient. This can also be explained in more details.

In the implementation of ParMETIS, load imbalance tolerance is one of the most important parameters in the sense that this value determines whether repartition will be switched on. The other equally significant implication of this value comes from the fact that it also establishes target load imbalance residual to be achieved after adaptive load balancing. That means for each repartition, the ParMETIS will only reduce the load imbalance to the provided tolerance.

In current implementation, the load imbalance tolerance is set to be the same for both switch-on and target values, which is not capable of bringing the best performance into our application due to the fact that aside from data redistribution cost, analysis model reconstruction is equally expensive. The dilemma is described by numerical example as shown in Table 110.5.

Table 110.5: Observation on Load Imbalance Tolerance %5

Model	20,476 Elements, 68,451 DOFs
CPUs	32
Imbalance Before	7.018%
Imbalance After	4.9%
Model Regeneration	57.2934 seconds
Total Step Time	140.961 seconds

We can easily see that tiny portion of data movement to balance out $7.018 - 4.9 = 2.228\%$ loads still invoked analysis model regeneration, which accounts for extra overhead that is about 40.6% of total step time.

Because the load balance tolerance is also the target value that the repartitioning operation hopes to achieve. The implication is that after repartitioning, the load distribution among processing units is barely under this acceptable tolerance. The performance study conducted so far showed that continuous plastification can easily creates load imbalance over this tolerance so another round of repartitioning would be launched again. It greatly brings down the performance of the whole application when the huge data redistribution overhead is taken just to overcome a tiny imbalance. This explains why changing the tolerance was not able to bring better performance in our application.

In order to improve performance while still minimizing load imbalance, we hope to maximize the efficiency of model regeneration routine in our application. This is a two-fold statement, firstly, we don't want to blindly increase the load imbalance because it basically claims we fail our adaptive PDD algorithm by not switching on repartitioning (5% is suggested by the author of ParMETIS Karypis et al. (2003) and has been proved to be the most stable value in this chapter), secondly, with each repartitioning, we hope to achieve "perfect balance" as much as possible and in this way, the huge model regeneration cost can be offset by performance gain. What was proposed as future extension of this chapter is the idea of dual load imbalance tolerances. Load balancing triggering tolerance and the target tolerance can be defined separately. We can set higher triggering tolerance to reduce the number of repartition counts, while on the other hand a strict target tolerance can be set close to 1.0 to get better load distribution out of the balancing routine. With proposed approach, our application in this chapter will be able to fully take advantage of the repartition routines without sacrificing too much on model regenerations.

110.4.8 Globally Adaptive PDD Algorithm

One significant drawback of current implementation is that neither network communication nor model regeneration cost has been considered in element-graph-based type domain decomposition algorithm. Element graph only records computational load carried by each element. Only one ITR factor characterizes algorithmic approach of the load balancing operation and this is apparently too crude for complicated network/hardware configurations. The ignorance of the repartitioning-associated overheads inherent with application codes can lead to serious performance drop of the proposed PDD algorithm as shown in Figure 110.43.

This drawback can harm the overall performance of the whole application code more seriously when the simulation is to be run on heterogeneous networks, which means we can have different network connections and nodes with varied computational power. The dilemma is, without exact monitoring of network communication and local model regeneration costs, we can easily sacrifice the performance gain by load balancing operations.

A second approach proposed in this chapter was the idea of modified Globally Adaptive PDD algorithm. The novelty comes from the fact that both data redistribution and analysis model regeneration costs will be monitored during execution. Load balancing will be triggered only when the performance gain necessarily offset the extra cost associated with the whole program. Domain graph structures will be kept intact till successful repartitioning happens. Meanwhile all elemental calculations will be timed to provide graph vertex weights. Data will be accumulated till algorithm restart happens, when all analysis

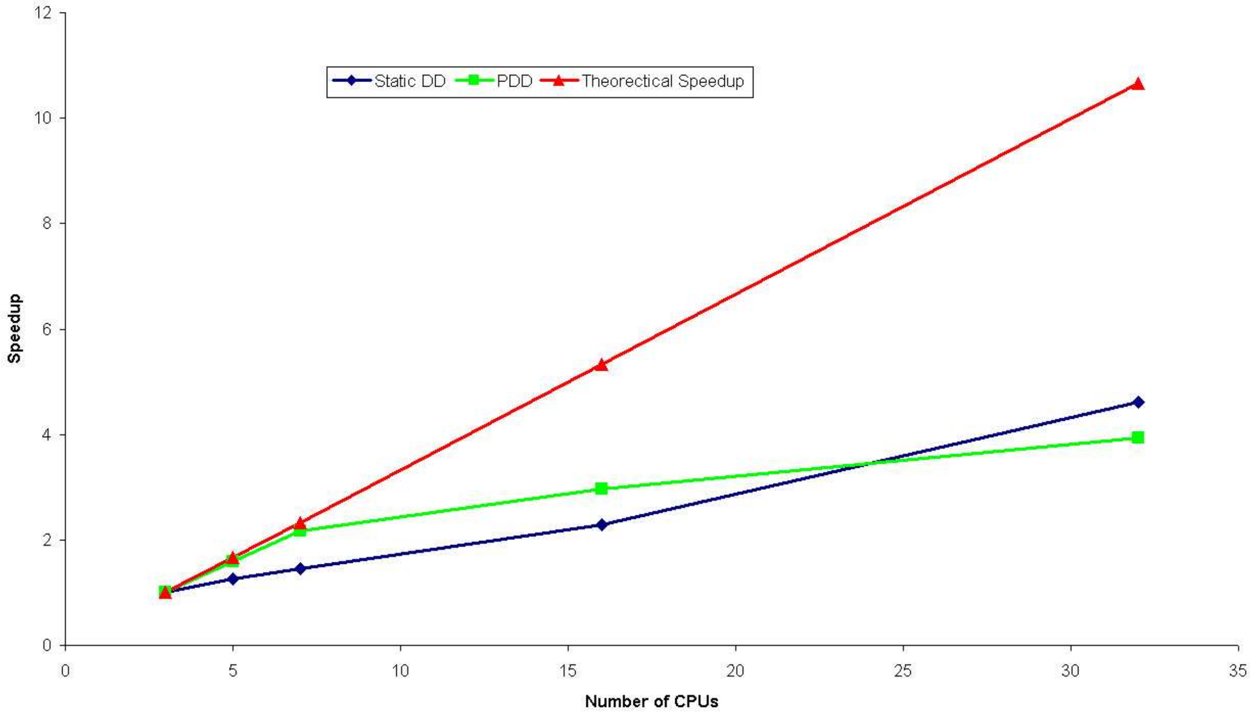


Figure 110.43: Absolute Speedup Data of Parallel Runs on 9,297 Elements, 32,091 DOFs Model, ITR=1e-3, Imbal Tol 5%

model and vertex weights will be nullified.

This improvement aims at handling network communication and any specific application-associated overheads automatically at the global level in order to remedy the drawback that the element graph repartitioning kernel currently supported by ParMETIS is not capable of directly reflecting this application level overheads. The new strategy is to automatically monitor network communication and local model regeneration timings which will be integrated to the entry of load balancing routines to act as additional triggers of the operation along with the load imbalance tolerance.

Performance study shows that PDD algorithm with the new additions significantly improve performance even when the number of processing units is large. This modification fixes the drawback shown in previous sections that the performance of PDD was beaten by static domain decomposition when the number of processors increases.

This strategy is called to be globally adaptive because both data communication and model regeneration costs are monitored at the application level, which tells best how the real application performs on all kinds of networks. Whatever the network/hardware configurations might be, real application runs always deliver the most accurate performance counters. This information can be applied on top of graph partitioning algorithm as a supplement to account for the drawback that the algorithm kernel is not capable of integrating global data communication costs.

110.4.8.1 Implementations

We can define the global overhead associated with load balancing operation as two parts, data communication cost T_{comm} and finite element model regeneration cost T_{regen},

$$T_{overhead} := T_{comm} + T_{regen} \tag{110.5}$$

Performance counters have been setup to study both.

- T_{comm}

  Data communication patterns characterizing the network configuration can be readily measured as the program runs the initial partitioning. As described in previous sections, initial domain decomposition needs to be done to send elements over to processing nodes. This step is necessary for parallel finite element processing and it provides perfect initial estimate how the communication pattern of the application performs on specific networks. Timing routines have been added to automatically measure the communication cost. This cost is inherently changing as the network condition might vary as simulation progresses, so whenever data redistribution happens, this metric will be automatically updated to reflect the network conditions.

- T_{regen}

  Model regeneration cost basically comes from the fact that if data redistribution happens, the analysis model needs to be regenerated to reflect changes of nodes and elements inside the domain. Detailed operations include renumbering DOFs and rehandling constraints. This part of cost is application-dependent. In current implementation of PDD, efforts have been made to set up timing stop at the entry and exit of model regeneration routines to get the accurate data for the extra overhead. It is also important to note that model regeneration happens when the initial data distribution finishes, again the initial domain decomposition phase provides perfect initial estimate of the model regeneration cost on any specific hardware configurations.

Naturally, for the load balancing operations to pay off, the $T_{overhead}$ has to be offset by the performance gain T_{gain}. This chapter also creates a strategy to estimate the performance gain T_{gain} even before the load balancing operation happens and this metric provides global control on top of the existing graph repartitioning algorithm.

As implemented in previous sections, the computational load on each element is represented by the associated vertex weight $vwgt[i]$. If the SUM operation is applied on every single processing node, the exact computational distribution among processors can be obtained as total wall clock time for each CPU as shown in Equation 110.6,

$$T_j := \sum_{i=1}^{n} vwgt[i],\ j = 1, 2, \ldots, np \tag{110.6}$$

in which n is the number of elements on each processing domain and np is the number of CPUs.

If we define,

$$T_{sum} := sum(T_j),\ T_{max} := max(T_j), \text{ and } T_{min} := min(T_j),\ j = 1, 2, \ldots, np \tag{110.7}$$

one always hope to minimize T_{max} because in parallel processing, T_{max} controls the total wall clock time. By load balancing operations, we mean to deliver evenly distributed computational loads among processors. So theoretically, the best execution time is,

$$T_{best} := T_{sum}/np, \text{ and } T_j \equiv T_{best},\ j = 1, 2, \ldots, np \tag{110.8}$$

if the perfect load balance is to be achieved.

Based on definitions above, the best performance gain T_{gain} one can obtain from load balancing operations can be calculated as,

$$T_{gain} := T_{max} - T_{best} \tag{110.9}$$

Finally, the load balancing operation will be beneficial IF AND ONLY IF

$$T_{gain} \geq T_{overhead} = T_{comm} + T_{regen} \tag{110.10}$$

110.4.8.2 Performance Results

The newly improved design has been compared to the old design to see the effectiveness of the globally adaptive switch of PDD algorithm.

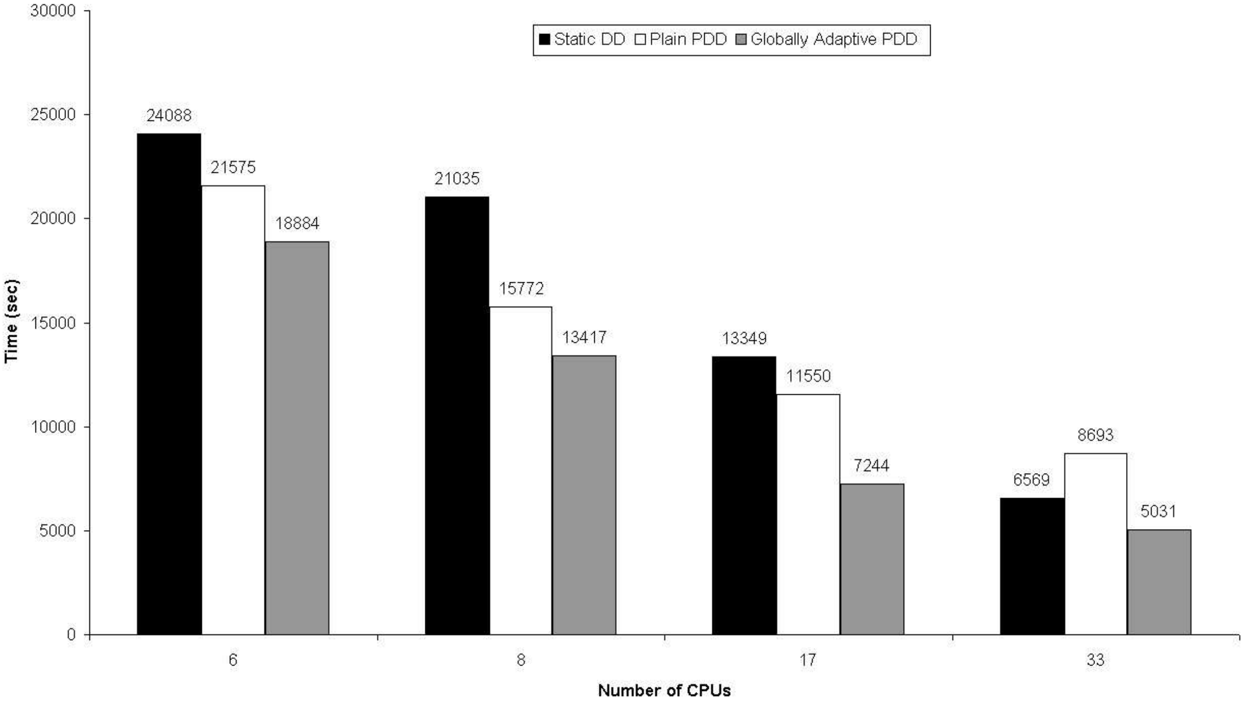


Figure 110.44: Performance of Globally Adaptive PDD on 9,297 Elements, 32,091 DOFs Model, ITR=1e-3, Imbal Tol 5%

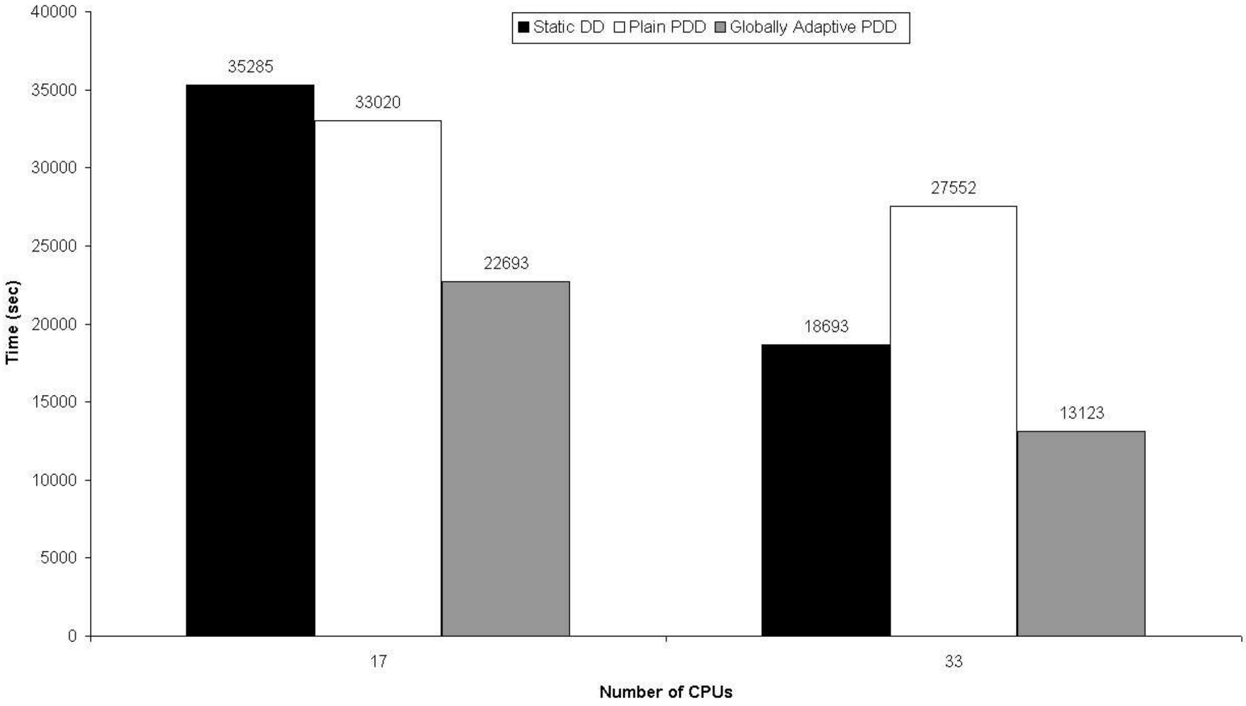


Figure 110.45: Performance of Globally Adaptive PDD on 20,476 Elements, 68,451 DOFs Model, ITR=1e-3, Imbal Tol 5%

From Figures 110.44 and 110.45, advantage of the improved globally adaptive PDD algorithm have clearly been shown. After considering the effect of both data communication and model regeneration costs, the adaptive PPD algorithm necessarily outperforms the static Domain Decomposition approach as expected. This new design also significantly improves the overall scalability of the proposed PDD algorithm as shown in Figure 110.46 and 110.47.

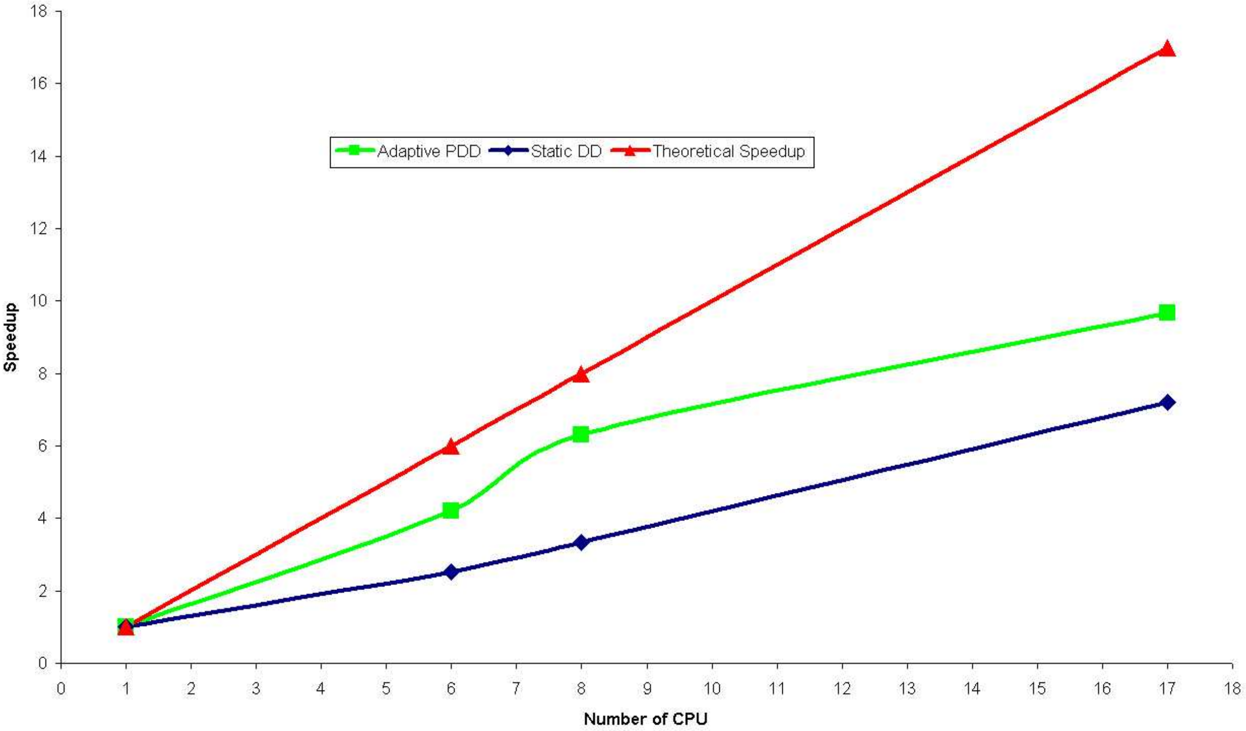


Figure 110.46: Scalability Study on 4,938 Elements, 17,604 DOFs Model, ITR=1e-3, Imbal Tol 5%

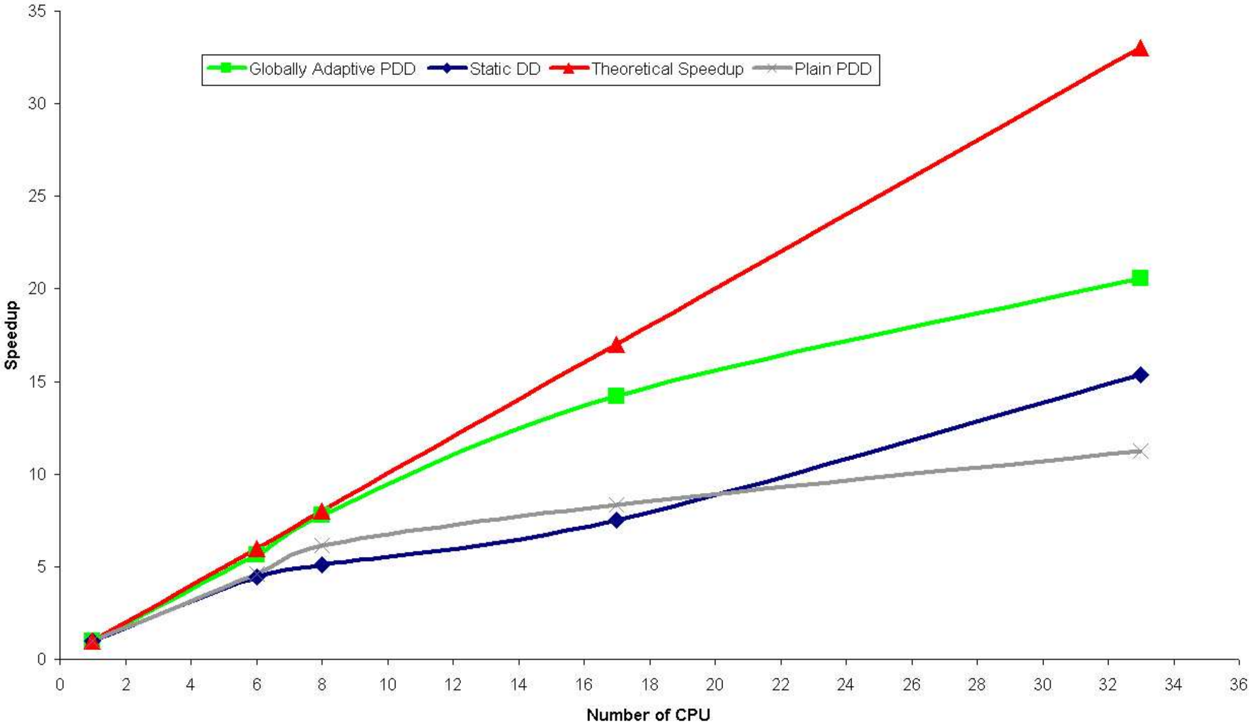


Figure 110.47: Scalability Study on 9,297 Elements, 32,091 DOFs Model, ITR=1e-3, Imbal Tol 5%

110.4.9 Scalability Study on Prototype Model

The ultimate purpose of this chapter is to develop an efficient parallel simulation tool for large scale earthquake analysis on prototype SFSI system. After in-depth development-refining process conducted in previous sections, real 3-bent production models have been set up to study the parallel performance of the proposed PDD algorithm using real world earthquake ground motions.

110.4.9.1 3 Bent SFSI Finite Element Models

As described in later sections, various sizes of a 3 bent bridge SFSI system has been developed to study dynamic behaviors of the whole system in different frequency domain. These models provide perfect test cases for parallel scalability study of our proposed PDD algorithm.

Detailed model description will be presented in later chapters of this chapter and only model size and mesh pictures are shown here to indicate the range of model sizes we have covered.

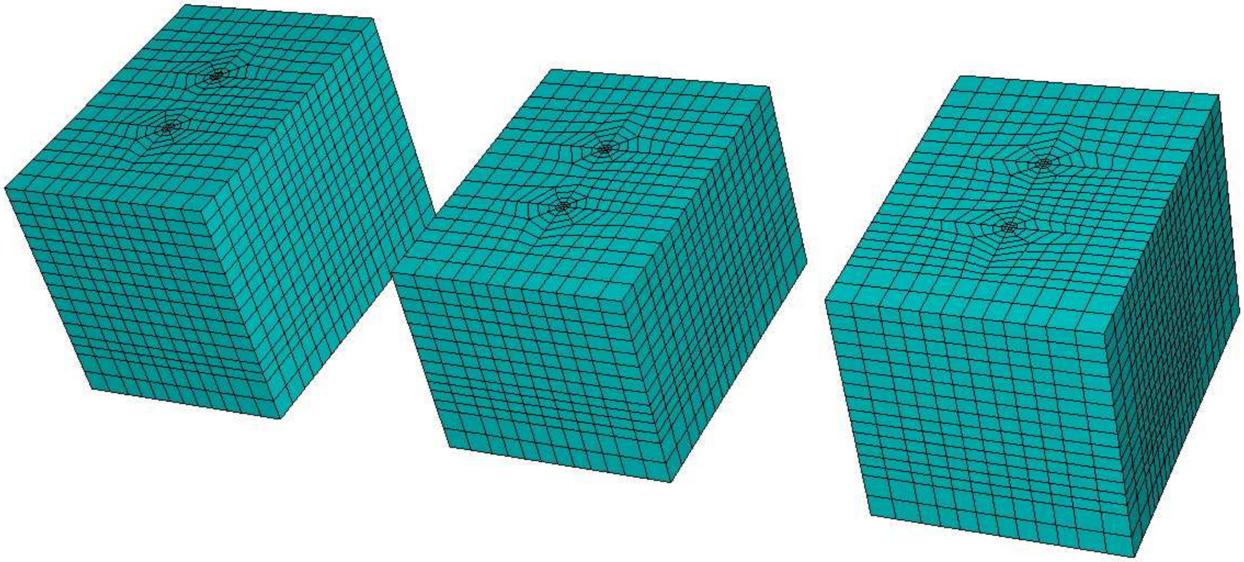


Figure 110.48: Finite Element Model - 3 Bent SFSI, 56,481 DOFs, 13,220 Elements, Frequency Cutoff > 3Hz, Element Size 0.9m, Minimum G/G_{max} 0.08, Maximum Shear Strain γ 1%

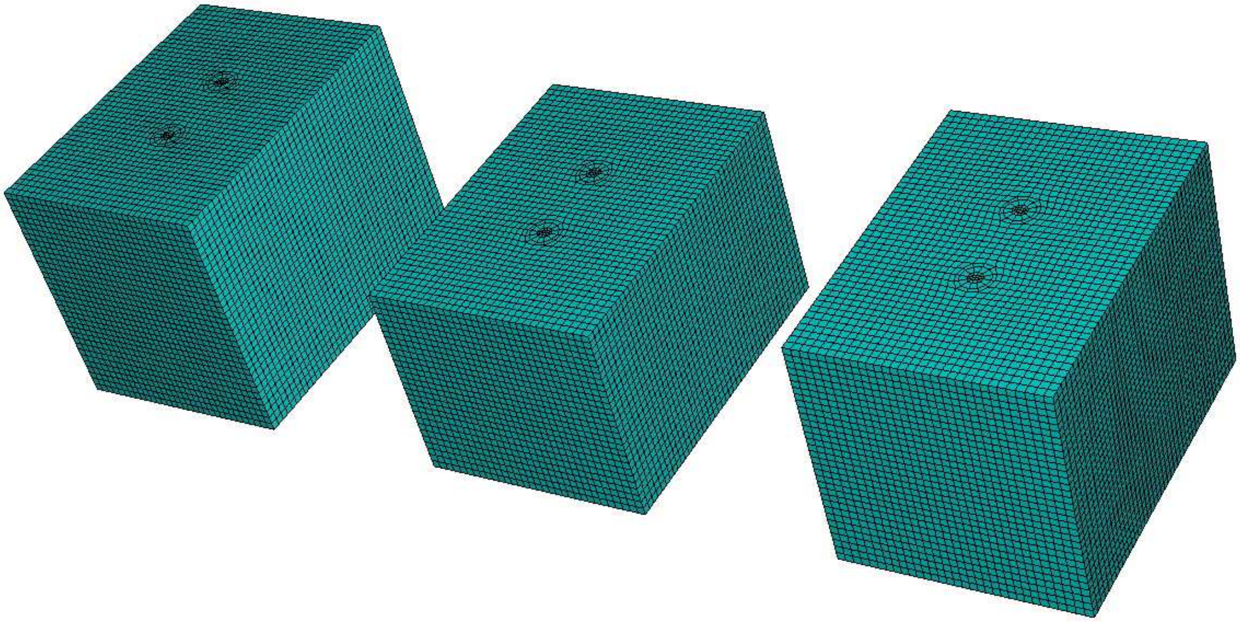


Figure 110.49: Finite Element Model - 3 Bent SFSI, 484,104 DOFs, 151,264 Elements, Frequency Cutoff 10Hz, Element Size 0.3m, Minimum G/G_{max} 0.08, Maximum Shear Strain γ 1%

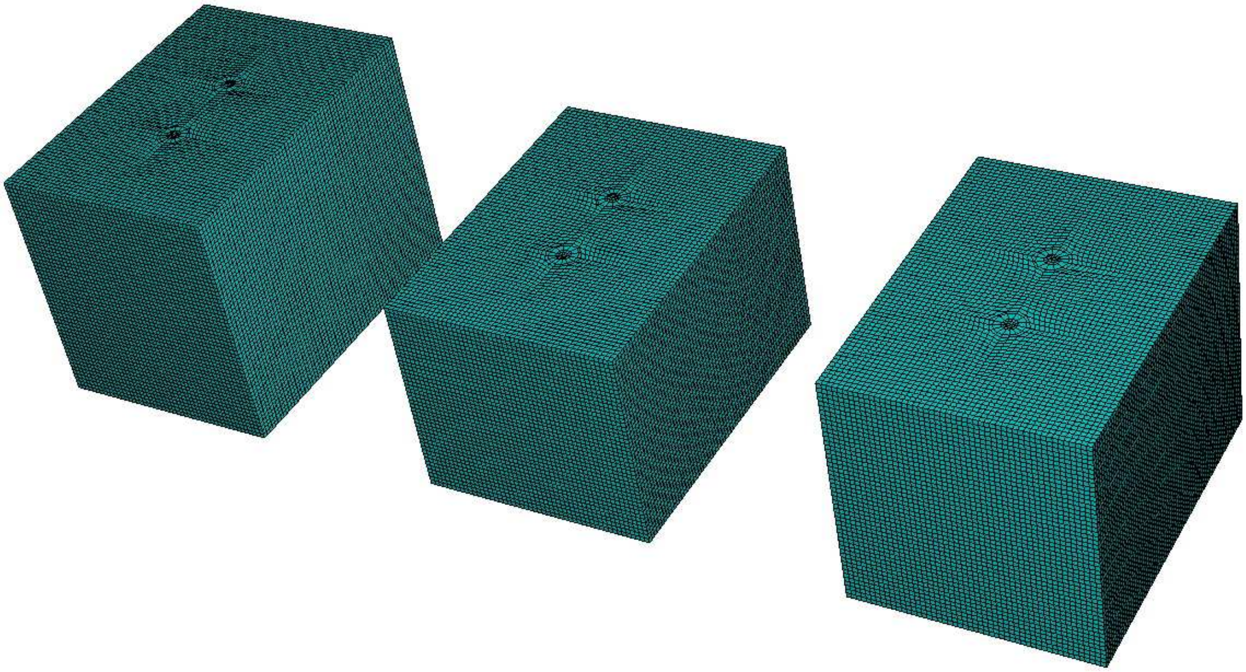


Figure 110.50: Finite Element Model - 3 Bent SFSI, 1,655,559 DOFs, 528,799 Elements, Frequency Cutoff 10Hz, Element Size 0.15m, Minimum G/G_{max} 0.02, Maximum Shear Strain γ 5%

110.4.9.2 Scalability Runs

The models with different detail levels have been subject to 1997 Northridge earthquake respectively for certain time steps and total wall clock time has been recorded to analyze the parallel scalability of our proposed PDD. The result is presented in Figure 110.51.

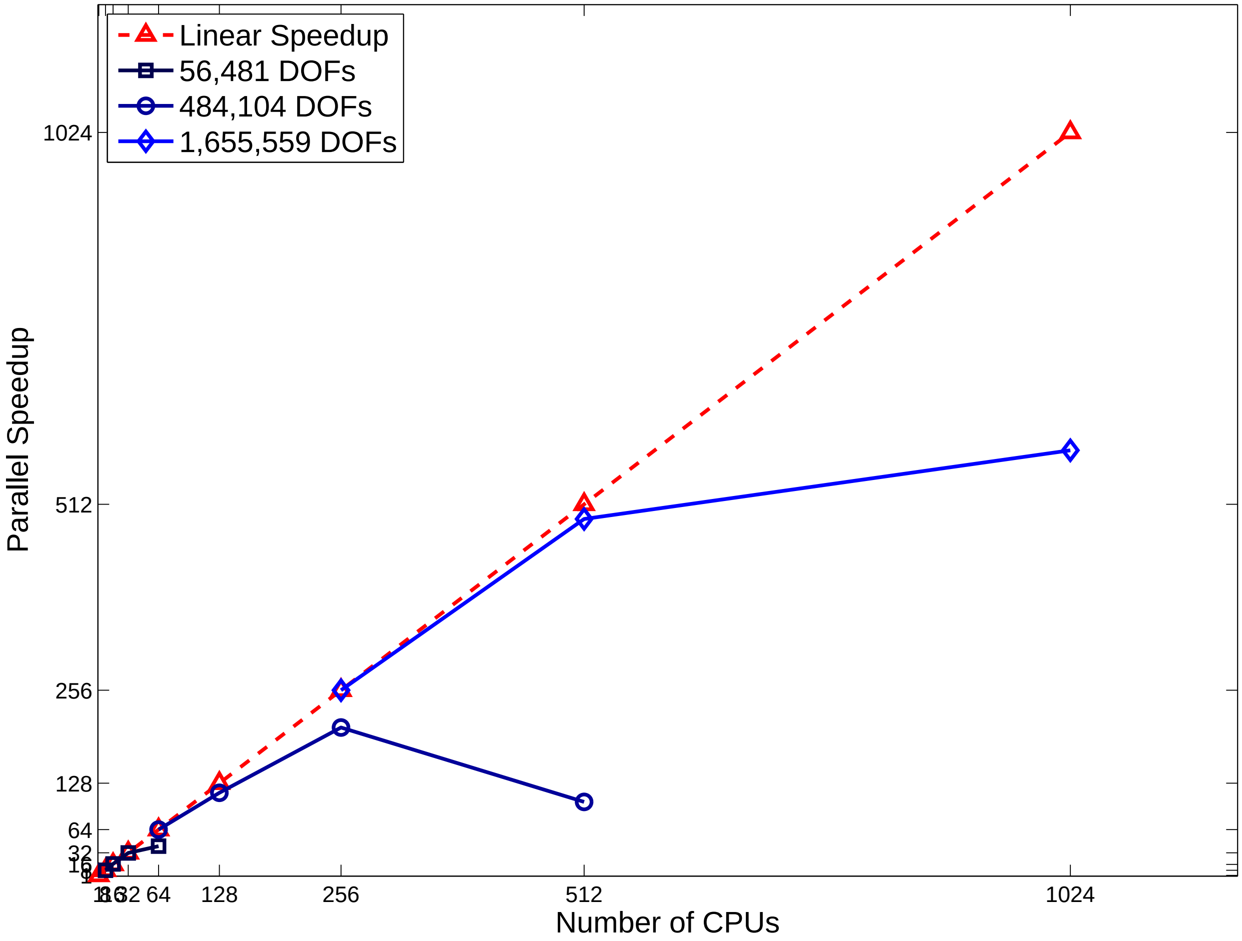


Figure 110.51: Scalability Study on 3 Bent SFSI Models, DRM Earthquake Loading, Transient Analysis, ITR=1e-3, Imbal Tol 5%, Performance Downgrade Due to Increasing Network Overhead

110.4.10 Conclusions

Through detailed performance studies as presented in previous sections, some conclusions can be drawn and future directions can be noted.

- Plastic Domain Decomposition (PDD) algorithm based on adaptive multilevel graph partitioning kernels has been shown to be effective for elastic-plastic parallel finite element calculations. PDD algorithm consistently outperforms classical Domain Decomposition method for models tested so far in this chapter as shown in Figures 110.52 and 110.54.

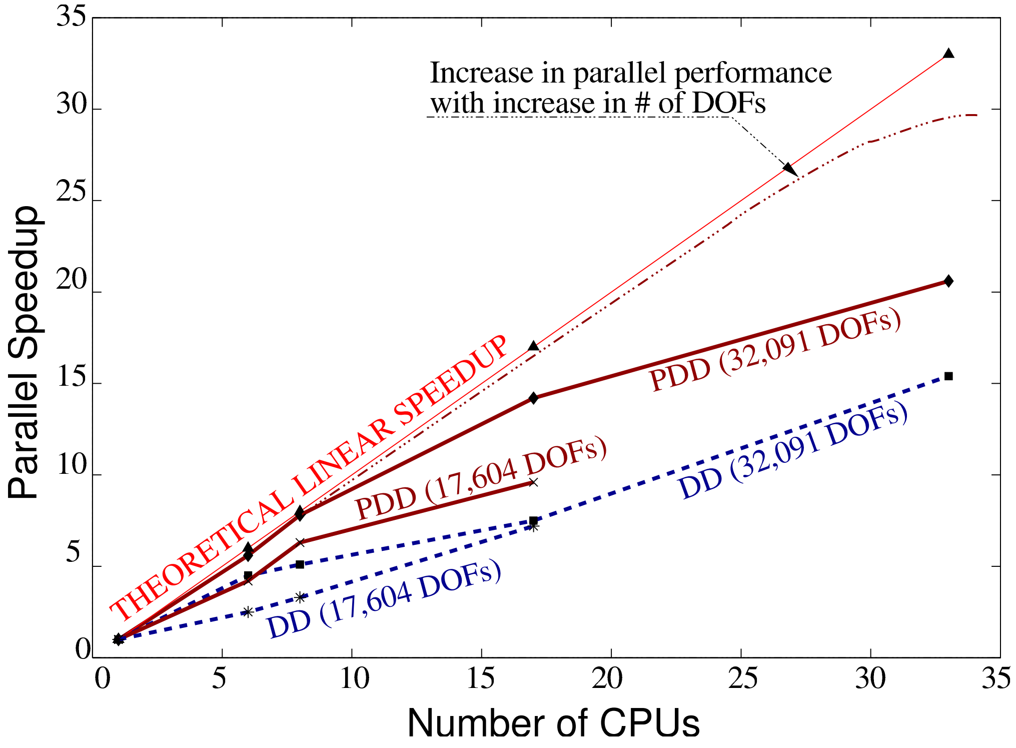


Figure 110.52: Relative Performance of PDD over DD, Shallow Foundation Model, Static Loading, ITR=1e-3, Imbal Tol 5%

- There are some parameters that can be calibrated in the current implementation. As indicated by results of thorough numerical tests, ITR=0.001 and load imbalance tolerance ubvec=1.05 (5%) should be adopted and studies on our application in this chapter have shown they are adequate and able to bring performance not worse than the commonly used domain decomposition method in parallel finite element analysis.

- For the parameters suggested in the chapter, we can see a general trend that the efficiency of PDD will drop as the number of processors increases. This can be explained. The implication of increasing processing units is that the subdomain problem size will decrease. It is naturally evident that the repartition load balancing won't be able to recover the overhead by balancing off small size local calculations. The improved design of globally adaptive PDD algorithm has been implemented in this chapter and both data communication and model regeneration costs associated with graph repartitioning have been integrated into the new globally adaptive strategy. With the new design, it has also been shown that the PDD algorithm consistently outperforms classic one step domain decomposition algorithm and better scalability can be obtained as shown in Figure 110.53. It has been shown that even for large number of processors, the current implementation can always guarantee that the performance of PDD is not worse than static DD method as shown in Figure 110.54. (the repartition routine has less than 5% overhead of the total wall clock time).

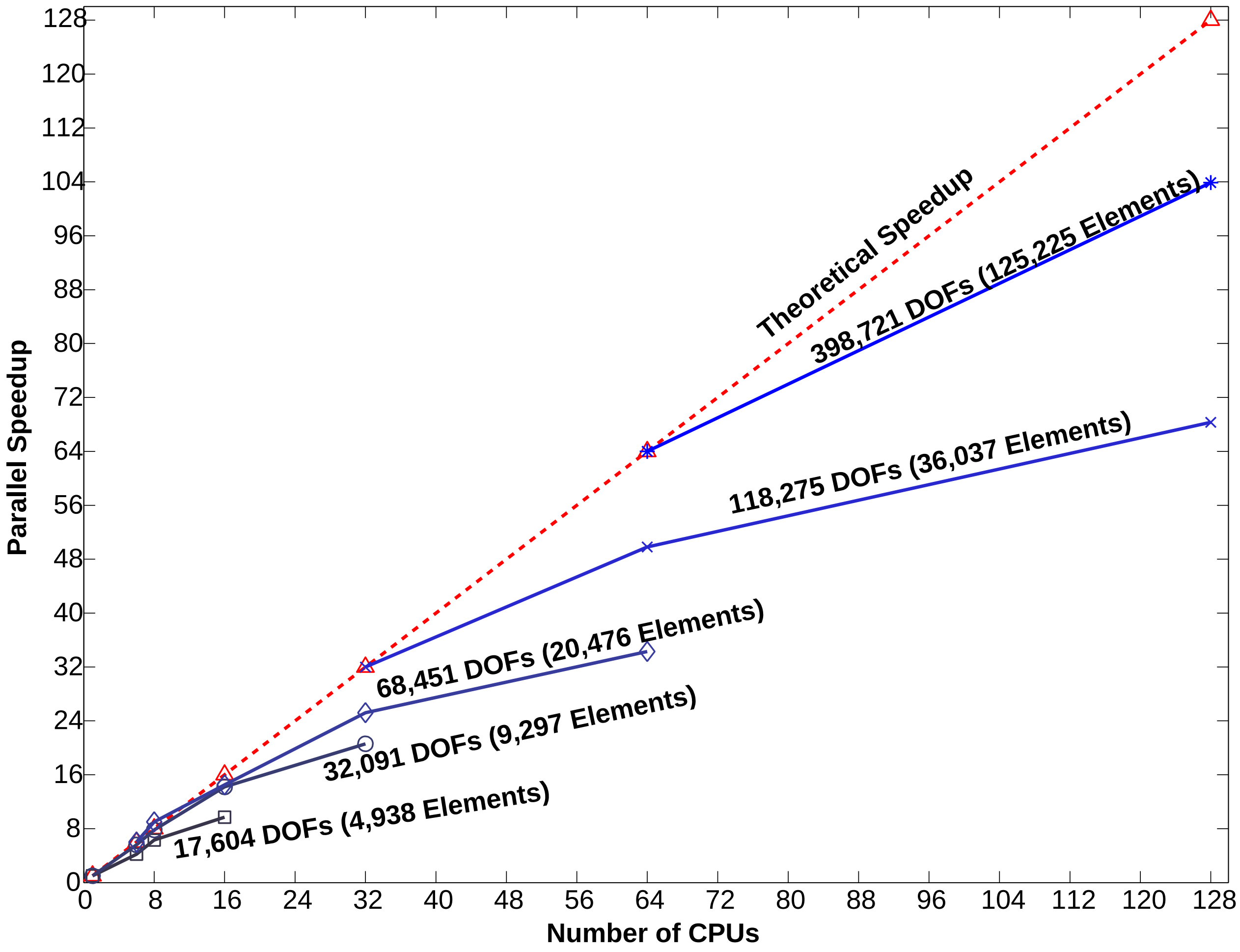


Figure 110.53: Scalability of PDD, Static Loading, Shallow Foundation Model, ITR=1e-3, Imbal Tol 5%

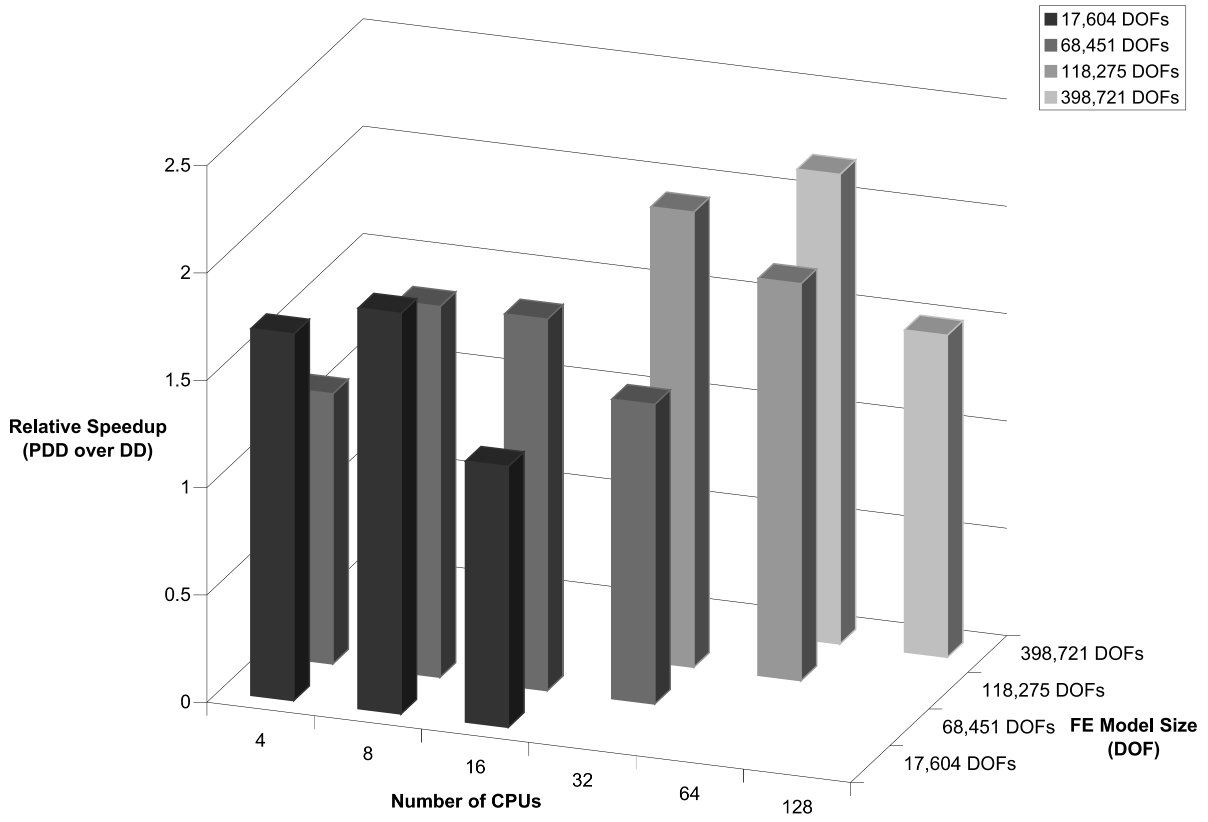


Figure 110.54: Relative Speedup of PDD over DD, Static Loading, Shallow Foundation Model, ITR=1e-3, Imbal Tol 5%

- If the problem size is fixed, there exists an optimum number of processors that can bring the best performance of the proposed load balancing algorithm. As the number of processing units increases after this number, the efficiency of proposed algorithm drops, which is understandable because the local load imbalance is so small overall that balancing gain won't offset the extra cost associated with repartitioning. But still the bottom line of proposed adaptive PDD algorithm is that it can run as fast as static one-step domain decomposition approach with less than 5% overhead of repartitioning routine calls. On the other hand, if the number of processing units is fixed, bigger finite element model will exhibit better performance. The conclusion is shown clearly in 3D in Figure 110.54.
- It is also worthwhile to point out that even without comparing with classical DD, PDD itself exhibits deteriorating performance as the number of processing units increases. Here the reproduction of Figure 110.53 is presented with some downside performance noted as shown in Figure 110.55.

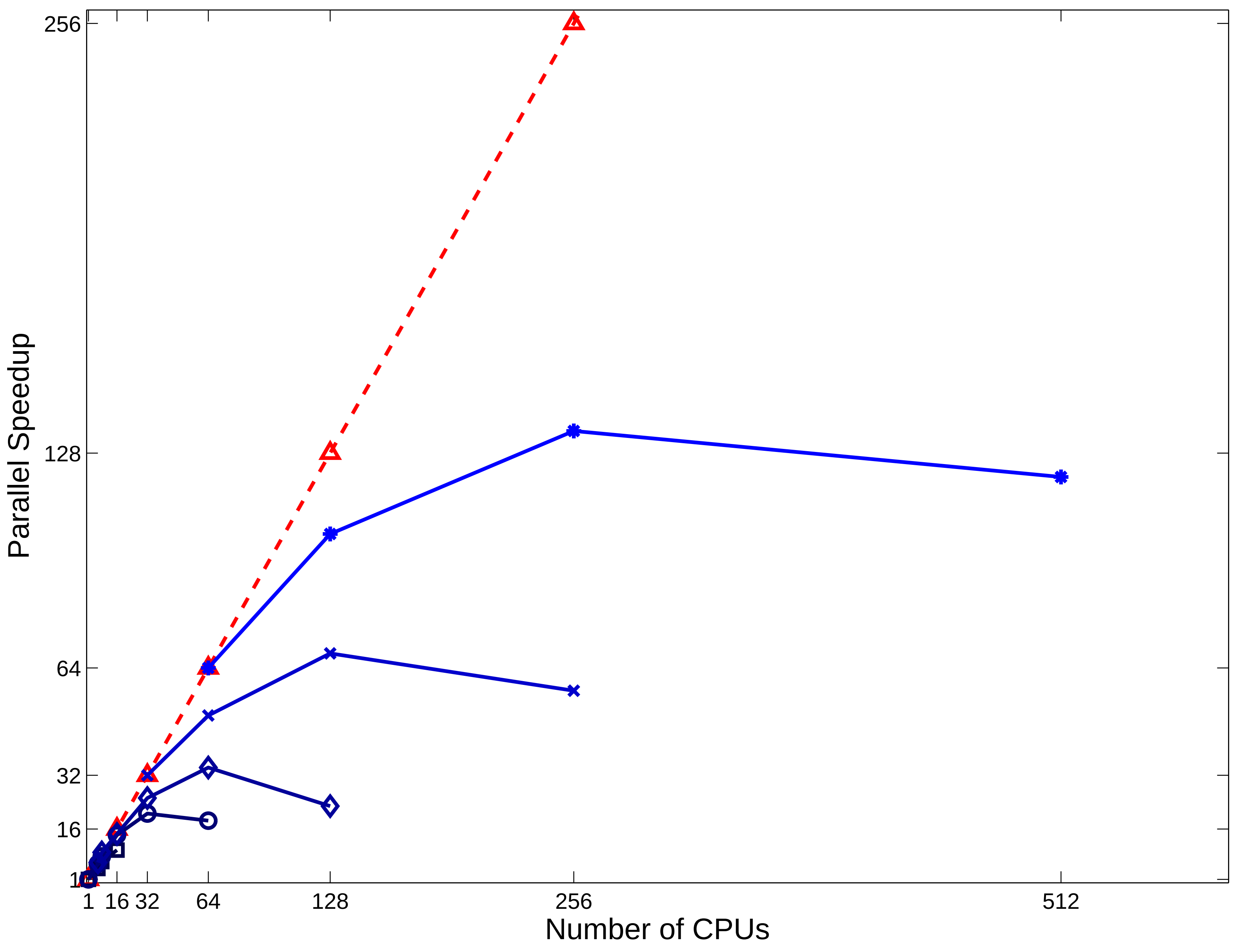


Figure 110.55: Full Range Scalability of PDD, Static Loading, Shallow Foundation Model, ITR=1e-3, Imbal Tol 5%, Performance Downgrade Due to Increasing Network Overhead

The implication is explained as follows:

- The performance drop partly is due to the communication overhead gets bigger and bigger so parallel processing will not be able to offset the communication loss.
- It is also noted that as the number of processing units increases, the elemental level calculation drops very scalably with the number of CPUs. This is inherently advantage of the proposed PDD algorithm. PDD through domain decomposition is very scalable for local level calculations because inherently local comp is element-based. when elements are distributed, loads are spread out evenly (during initial and redistribution). So as the number of CPU increases, the equation solving becomes more expensive.

  For the case of 56,481 DOFs prototype model with DRM earthquake loading, it has been observed that for sequential case (1 CPU), elemental computation takes 70% of time. As for parallel case (8 CPUs), we optimized parallel elemental computations through PDD, elemental computation only accounts for about 40%. As the number of CPU increases, parallel case (32 CPUs), the local level computation will only take less than 10% of total wall clock time.

  In other words, as the number of CPUs increases, PDD loses scalability because of the equation solving now dominates. As being discussed in Chapter 110.5, the parallel direct solver itself is not scalable up to large number of CPUs Demmel et al. (1999a). Parallel iterative solver is much more scalable but difficult to guarantee convergence. This is now also the most important topic in the whole scientific computing community.

For one set of fixed algorithm parameters, such as ITR and load imbalance tolerance, basic conclusion is there exists an optimal number of processors that can bring best performance and as finite element model size increases, this number increases as listed in Table 110.6.

Table 110.6: Best Performance Observed for ITR=0.001, Load Imbalance Tolerance %5

# of DOFs	Speedup	# of CPUs
4,035	1.553	4
17,604	1.992	7
32,091	1.334	7
68,451	1.068	16

The second point is related to the implementation of the multilevel graph partitioning algorithm. In current implementation of ParMETIS used in this chapter, vertex weight can only be specified as an int. That means in order to get timing data from local level calculation for each element, double data returned by MPI timing routine has to be converted to int. Significant digit loss can happen depending on what accuracy the system clock can carry. We can also adjust the vertex weight by amplifying the timing by scale factors in order to save effective digits. 10 millisecond has been used in this chapter to represent the effective timing digits when converting from double to int.

110.5 Application of Project-Based Iterative Methods in SFSI Problems

110.5.1 Introduction

Finite element method has been the most extensively used numerical method in computational mechanics. Equation solver is the numerical kernel of any finite element package. Gauss elimination type direct solver has dominated due to its robustness and predictability in performance.

As modern computer becomes more and more powerful, more advanced and detailed models need to be analyzed by numerical simulation. Direct solver is not the favorite choice for large scale finite element calculations because of high memory requirements and the inherent lack of parallelism of the method itself.

The motivation for presented work on iterative solvers stems from the need to expand the toolset of parallel iterative solvers for large scale simulation problems related to Earthquake-Soil-Structure interaction problems

In this section, the effectiveness of Krylov iterative methods has been tested in solving soil-structure interaction problems. Preconditioning techniques have been introduced. Robustness of iterative solvers has been investigated on equation systems from real soil-structure interaction problems. Several popular parallel algorithms and tools have been collected and implemented on PETSc platform to solve the SFSI problems. Performance study has been carried out using IA64 super computers at San Diego Supercomputing Center. A complete implementation has been developed within our computational system, within MOSS libraries, with extensive use of ParMETIS, and other material and numerical libraries.

110.5.2 Projection-Based Iterative Methods

Projection techniques are defined as methods to find approximate solutions $\hat{x}$ for $Ax = b$ ($A \in \mathcal{R}^{n \times n}$) in a subspace $\mathcal{W}$ of dimension m. Then in order to determine $\hat{x}$, we need m independent conditions. One way to obtain these is by requiring the residual $b - A\hat{x}$ is orthogonal to a subspace $\mathcal{V}$ of dimension m, i.e.,

$$\hat{x} \in \mathcal{W}, b - A\hat{x} \perp \mathcal{V} \tag{110.11}$$

The conditions shown in Equation 110.11 are known as Petrov-Galerkin conditions (Bai, 2007).

There are two key questions to answer if one wants to use projection techniques in solving large scale linear systems. Different answers lead to many variants of the projection method.

- Choice of Subspaces

  Krylov subspaces have been the favorite of most researchers and a large family of methods have been developed based on Krylov subspaces. Typically people choose either $\mathcal{V} = \mathcal{W}$ or $\mathcal{V} = A\mathcal{W}$ with $\mathcal{V}$ and $\mathcal{W}$ both Krylov subspaces.

- Enforcement of Petrov-Galerkin Conditions

  Arnoldi's procedure and Lanczos algorithm are two choices for building orthogonal or biorthogonal sequence to enforce the projection conditions.

The iterative methods discussed in this section are generally split into two categories, one based on Arnoldi's procedure and the other on Lanczos biorthoganalization. The most popular for the first family are Conjugate Gradient and General Minimum Residual methods, while Bi-Conjugate Gradient and Quasi-Minimum Residual methods represent the Lanczos family.

110.5.2.1 Conjugate Gradient Algorithm

The conjugate gradient (CG) algorithm is one of the best known iterative techniques for solving sparse symmetric positive definite (SPD) linear systems. This method is a realization of an orthogonal projection technique onto the Krylov subspace $\mathcal{K}_m(A, r_0)$, where r_0 is the initial residual. Because A is symmetry, some simplifications resulting from the three-term Lanczos recurrence will lead to more elegant algorithms (Demmel, 1997).

ALGORITHM CG (Saad, 2003)

```
1. Compute $r_0 := b - Ax_0$, $p_0 := r_0$
2. For $j = 0, 1, \cdots$, until convergence, Do
3.     $\alpha_j := (r_j, r_j)/(Ap_j, p_j)$
4.     $x_{j+1} := x_j + \alpha_j p_j$
5.     $r_{j+1} := r_j - \alpha_j Ap_j$
6.     $\beta_j := (r_{j+1}, r_{j+1})/(r_j, r_j)$
7.     $p_{j+1} := r_{j+1} + \beta_j p_j$
8. EndDo
```

- Applicability

  Matrix A is SPD.

- Subspaces

  Choose $\mathcal{W} = \mathcal{V} = \mathcal{K}_m(A, r_0)$, in which initial residual $r_0 = b - Ax_0$.

- Symmetric Lanczos Procedure

  This procedure can be viewed as a simplification of the Arnoldi's procedure when A is symmetric. Great three-term Lanczos recurrence is discovered when the symmetry of A is considered (Demmel, 1997).

- Optimality

  If A is SPD and one chooses $\mathcal{W} = \mathcal{V}$, enforcing Petrov-Galerkin conditions minimizes the A-norm of the error over all vectors $x \in \mathcal{W}$, i.e., $\hat{x}$ solves the problem,

  $$\min_{x \in \mathcal{W}} \|x - x^*\|_A, x^* = A^{-1}b \tag{110.12}$$

  From the lemma above, one can derive global minimization property of the Conjugate Gradient method. The vector x_k in the Conjugate Gradient method solves the minimization problem

  $$\min_x \phi(x) = \frac{1}{2}\|x - x^*\|_A^2, x - x_0 \in \mathcal{K}_k(A, r_0) \tag{110.13}$$

- Convergence

  In exact arithmetic, the Conjugate Gradient method will produce the exact solution to the linear system $Ax = b$ in at most n steps and it owns the superlinear convergence rate. The behavior of Conjugate Gradient algorithm in finite precision is much more complex. Due to rounding errors, orthogonality is lost quickly and finite termination does not hold anymore. What is more meaningful in application problems would be to use CG method for solving large, sparse, well-conditioned linear systems in far fewer than n iterations.

110.5.2.2 GMRES

The Generalized Minimum Residual method is able to deal with more general type of matrices.

ALGORITHM GMRES (Saad, 2003)

```
1. Compute $r_0 := b - Ax_0$, $\beta := \|r_0\|_2$, and $v_1 := r_0/\beta$
2. For $j = 1, 2, \cdots, m$, Do
3.         Compute $\omega_j := Av_j$
4.         For $i = 1, \cdots, j$, Do
5.             $h_{ij} := (\omega_j, v_i)$
6.             $\omega_j := \omega_j - h_{ij} v_i$
7.         EndDo
8.         $h_{j+1,j} = \|\omega_j\|_2$. If $h_{j+1,j} = 0$ set $m := j$ and go to 11
9.         $v_{j+1} = \omega_j / h_{j+1,j}$
10. EndDo
11. Define the $(m+1) \times m$ Hessenberg matrix $\bar{H}_m = \{h_{ij}\}_{1 \le i \le m+1, 1 \le j \le m}$
12. Compute $y_m$, the minimizer of $\|\beta_i e_1 - \bar{H}_m y\|_2$, and $x_m = x_0 + V_m y_m$
```

- Applicability

  Matrix A is nonsingular.

- Subspaces

  Choose $\mathcal{W} = \mathcal{K}_m(A, r_0)$ and $\mathcal{V} = A\mathcal{W} = A\mathcal{K}_m(A, r_0)$, in which initial residual $r_0 = b - Ax_0$.

- Arnoldi's Procedure

  Classic Arnoldi's procedure (modified Gram-Schmidt) is followed in GMRES (Bai, 2007).

- Optimality

  If one chooses $\mathcal{V} = A\mathcal{W}$, enforcing Petrov-Galerkin conditions solves the least square problem

$$\|b - A\tilde{x}\|_2 = \min_{x \in \mathcal{W}} \|b - Ax\|_2 \tag{110.14}$$

- Convergence

  It has been shown that in exact arithmetic, GMRES can not breakdown and will give exact solutions in at most n steps. In practice, the maximum steps GMRES can run depends on the memory due to the fact it needs to store all Arnoldi vectors. Restarting schemes have been proposed for a fixed m, which is denoted by GMRES(m). Typical value for m can be $m \in [5, 20]$. GMRES(m) can not breakdown in exact arithmetic before the exact solution has been reached. But it may never converge for $m < n$ (Bai, 2007).

110.5.2.3 BiCGStab and QMR

These two methods are based on nonsymmetric Lanczos procedure, which is quite different from Arnoldi's in the sense that it formulates biorthogonal instead of orthogonal sequence. They are counterparts of CG and GMRES method, which follows similar derivation procedure except the Lanczos biorthogonalization is used instead of Arnoldi's procedure (Bai, 2007).

110.5.3 Preconditioning Techniques

Lack of robustness is a widely recognized weakness of iterative solvers relative to direct solvers. Using preconditioning techniques can greatly improve the efficiency and robustness of iterative methods. Preconditioning is simply a means of transforming the original linear system into one with the same solution but easier to solve with an iterative solver. Generally speaking, the reliability of iterative techniques, when dealing with various applications, depends much more on the quality of the preconditioner than on the particular Krylov subspace accelerator used.

The first step in preconditioning is to find a preconditioning matrix M. The matrix M can be defined in many different ways but there are a few minimal requirements the M is supposed to satisfy (Benzi, 2002).

1. From practical point of view, the most important requirement of M is that it should be inexpensive to solve linear system $Mx = b$. This is because the preconditioned algorithm will all require a linear system solution with the matrix M at each step.
2. The matrix M should be somehow close to A and it should not be singular. We can see that actually most powerful preconditioners are constructed directly from A.
3. The preconditioned $M^{-1}A$ should be well-conditioned or has very few extreme eigenvalues thus M can accelerate convergence dramatically.

Once a preconditioner M is available, there are three ways to apply it.

1. Left Preconditioning

$$M^{-1}Ax = M^{-1}b \tag{110.15}$$

2. Right Preconditioning

$$AM^{-1}u = b, x \equiv M^{-1}u \tag{110.16}$$

3. Split Preconditioning

   It is a very common situation that M is available in factored form $M = M_L M_R$, in which, typically, M_L and M_R are triangular matrices. Then the preconditioning can be split,

$$M_L^{-1} A M_R^{-1} u = b, x \equiv M_R^{-1} u \tag{110.17}$$

   It is imperative to preserve symmetry when the original matrix A is symmetric, so the split preconditioner seems mandatory in this case.

Consider that a matrix A that is symmetric and positive definite and assume that a preconditioner M is available. The preconditioner M is a matrix that approximates A in some yet-undefined sense. We normally require that the M is also symmetric positive definite.

In order to preserve the nice SPD property, in the case when M is available in the form of an incomplete Cholesky factorization, $M = LL^T$, people can simply just use the split preconditioning, which yields the SPD matrix

$$L^{-1} A L^{-T} u = L^{-1} b, x \equiv L^{-T} u \tag{110.18}$$

However, it is not necessary to split the preconditioner in this manner in order to preserve symmetry. Observe that $M^{-1}A$ is self-adjoint for the M inner product

$$(x, y)_M \equiv (Mx, y) = (x, My) \tag{110.19}$$

since

$$(M^{-1}Ax, y)_M = (Ax, y) = (x, Ay) = (x, M(M^{-1}A)y) = (x, M^{-1}Ay)_M \tag{110.20}$$

Therefore, an alternative is to replace the usual Euclidean inner product in the CG algorithm with the M inner product (Saad, 2003).

If the CG algorithm is rewritten for this new inner product, denoting by $r_j = b - Ax_j$ the original residual and by $z_j = M^{-1} r_j$ the residual for the preconditioned system, the following sequence of operations is obtained, ignoring the initial step:

1. $\alpha_j := (z_j, z_j)_M / (M^{-1}Ap_j, p_j)_M$,

2. $x_{j+1} := x_j + \alpha_j p_j$,

3. $r_{j+1} := r_j - \alpha_j A p_j$ and $z_{j+1} := M^{-1} r_{j+1}$,

4. $\beta_j := (z_{j+1}, z_{j+1})_M / (z_j, z_j)_M$,

5. $p_{j+1} := z_{j+1} + \beta_j p_j$.

Since $(z_j, z_j)_M = (r_j, z_j)$ and $(M^{-1}Ap_j, p_j)_M = (Ap_j, p_j)$, the M inner products do not have to be computed explicitly. With this observation, the following algorithm is obtained.

ALGORITHM Preconditioned CG (Saad, 2003)

```
1. Compute $r_0 := b - Ax_0$, $z_0 := M^{-1}r_0$, $p_0 := z_0$
2. For $j = 0, 1, \cdots$, until convergence, Do
3.     $\alpha_j := (r_j, z_j)/(Ap_j, p_j)$
4.     $x_{j+1} := x_j + \alpha_j p_j$
5.     $r_{j+1} := r_j - \alpha_j Ap_j$
6.     $z_{j+1} := M^{-1}r_{j+1}$
7.     $\beta_j := (r_{j+1}, z_{j+1})/(r_j, z_j)$
8.     $p_{j+1} := z_{j+1} + \beta_j p_j$
9. EndDo
```

110.5.4 Preconditioners

Finding a good preconditioner to solve a given sparse linear system is often viewed as a combination of art and science. Theoretical results are rare and some methods work surprisingly well, often despite expectations. As it is mentioned before, the preconditioner M is always close to A in some undefined-yet sense. Some popular preconditioners will be introduced in this section.

110.5.4.1 Jacobi Preconditioner

This might be the simplest preconditioner people can think of. If A has widely varying diagonal entries, we may just use diagonal preconditioner $M = \text{diag}(a_{11}, \cdots, a_{nn})$. One can show that among all possible diagonal preconditoners, this choice reduces the condition number of $M^{-1}A$ to within a factor of n of its minimum value.

110.5.4.2 Incomplete Cholesky Preconditioner

Another simple way of defining a preconditioner that is close to A is to perform an incomplete Cholesky factorization of A. Incomplete factorization formulates an approximation of $A \approx \hat{L}\hat{L}^T$, but with less or no fill-ins relative to the complete factorization $A = LL^T$ (Demmel, 1997).

ALGORITHM Incomplete Cholesky Factorization (Saad, 2003)

```
1. For $j = 1, 2, \cdots, n$, Do
3.         $l_{jj} := \sqrt{a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2}$
4.         For $i = j + 1, \cdots, n$, Do
5.               $l_{ij} = (a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk}) / l_{jj}$
6.               Apply dropping rule to $l_{ij}$
7.         EndDo
8. EndDo
```

There are many ways to control the number of fill-ins in IC factorization. No fill-in version of incomplete Cholesky factorization IC(0) is rather easy and inexpensive to compute. On the other hand, it often leads to a very crude approximation of A, which may result in the Krylov subspace accelerator requiring too many iterations to converge. To remedy this, several alternative incomplete factorizations have been developed by researchers by allowing more fill-in in L, such as incomplete Cholesky factorization with dropping threshold IC(ϵ). In general, more accurate IC factorizations require fewer iterations to converge, but the preprocessing cost to compute the factors is higher.

110.5.4.3 Robust Incomplete Factorization

Incomplete factorization preconditioners are quite effective for many application problems but special care must be taken in order to avoid breakdowns due to the occurrence of non-positive pivots during the incomplete factorization process.

The existence of an incomplete factorization $A \approx \hat{L}\hat{L}^T$ has been established for certain classes of matrices. For the class of M-matrices, the existence of incomplete Cholesky factorization was proved for arbitrary choices of the sparsity pattern (Meijerink and van der Vorst, 1977). The existence result was extended shortly thereafter to a somewhat larger class (that of H-matrices with positive diagonal entries) (Manteuffel, 1980; Varga et al., 1980; Robert, 1982). Benzi and Tůma (2003) presents reviews on the topic of searching for robust incomplete factorization algorithms and an robust algorithm based on A-Orthogonalization has been proposed.

In order to construct triangular factorization of A, the well-known is not the only choice. Benzi and Tůma (2003) shows how the factorization $A = LDL^T$ (root-free factorization) can be obtained by means of an A-orthogonalization process applied to the unit basis vectors $e_1, e_2, \cdots, e_n$. This is simply the Gram-Schmidt process with respect to the inner product generated by the SPD matrix A. This idea is not new and as a matter of fact, it was originally proposed at as early as 1940's in Fox et al. (1948). It has been observed in Hestenes and Stiefel (1952) that A-orthogonalization of the unit basis vectors is closely related to Gaussian elimination but this algorithm costs twice as much as the Cholesky factorization in the dense case.

Factored Approximate Inverse Preconditioner In reference Benzi et al. (1996) A-orthogonalization has been exploited to construct factored sparse approximate inverse preconditioners noting the fact that A-orthogonalization also produces the inverse factorization $A^{-1} = ZD^{-1}Z^T$ (with Z unit upper triangular and D diagonal). Because the A-orthogonaliza-tion, even when performed incompletely, is not subject to pivot breakdowns, these preconditioners are reliable (Benzi et al., 2000). However, they are often less effective than incomplete Cholesky preconditioning at reducing the number of PCG iterations and their main interest stems from the fact that the preconditioning operation can be applied easily in parallel because triangular solve is not necessary in approximate inverse preconditioning.

Reference Benzi and Tůma (2003) investigates the use of A-orthogonalization as a way to compute an incomplete factorization of A rather than A^{-1} thus a reliable preconditioning algorithm can be developed. The basic A-orthogonalization procedure can be written as follows (Benzi et al., 2000).

ALGORITHM Incomplete Factored Approximate Inverse (Benzi et al., 1996)

```
1.  Let $z_i^{(0)} = e_i$, for $i = 1, 2, \cdots, n$
2.  For $i = 1, 2, \cdots, n$, Do
3.        For $j = i, i+1, \cdots, n$, Do
4.              $p_j^{(i-1)} := a_i^T z_j^{(i-1)}$
5.        EndDo
6.        For $j = i+1, \cdots, n$, Do
7.              $z_j^{(i)} := z_j^{(i-1)} - (\frac{p_j^{(i-1)}}{p_i^{(i-1)}})$
8.              Apply dropping to $z_j^{(i)}$
9.        EndDo
10. EndDo
11. Let $z_i := z_i^{(i-1)}$ and $p_i := p_i^{(i-1)}$, for $i = 1, 2, \cdots, n$.
12. Return $Z = [z_1, z_2, \cdots, z_n]$ and $D = diag(p_1, p_2, \cdots, p_n)$.
```

The basic algorithm described above can suffer a breakdown when a negative or zero value of a pivot p_i. When no dropping is applied, $p_i = z_i^T A z_i > 0$. The incomplete procedure is well defined, i.e., no breakdown can occur, if A is an H-matrix (in the absence of round-off). In the general case, breakdowns can occur. Breakdowns have a crippling effect on the quality of the preconditioner. A negative p_i would result in an approximate inverse which is not positive definite; a zero pivot would force termination of the procedure, since step (7) cannot be carried out.

The way proposed to avoid non-positive pivots is simply to recall that in the exact A-orthogonalization process, the p_i's are the diagonal entries of matrix D which satisfies the matrix equation

$$Z^T A Z = D \tag{110.21}$$

hence for $1 < i < n$

$$p_i = z_i^T A z_i > 0 \tag{110.22}$$

since A is SPD and $z_i \neq 0$. In the exact process, the following equality holds

$$p_i = z_i^T A z_i = a_i^T z_i \quad \text{and} \quad p_j = z_i^T A z_j = a_i^T z_j \tag{110.23}$$

Clearly it is more economical to compute the pivots using just inner product $a_i^T z_i$ rather than the middle expression involving matrix-vector multiply. However, because of dropping and the resulting loss of A-orthogonality in the approximate $\bar{z}$-vectors, such identities no longer hold in the inexact process and for some matrices one can have

$$a_i^T \bar{z}_i \ll \bar{z}_i^T A \bar{z}_i \tag{110.24}$$

The robust algorithm requires that the incomplete pivots $\bar{p}_i$'s be computed using the quadratic form $\bar{z}_i^T A \bar{z}_i$ throughout the AINV process, for $i = 1, 2, \cdots, n$.

ALGORITHM Stabilized Incomplete Approximate Inverse (Benzi et al., 2000)

```
Let $z_i^{(0)} = e_i$, for $i = 1, 2, \cdots, n$
For $i = 1, 2, \cdots, n$, Do
    $v_i := A z_i^{(i-1)}$
    For $j = i, i+1, \cdots, n$, Do
        $p_j^{(i-1)} := v_i^T z_j^{(i-1)}$
    EndDo
    For $j = i+1, \cdots, n$, Do
        $z_j^{(i)} := z_j^{(i-1)} - (\frac{p_j^{(i-1)}}{p_i^{(i-1)}})$
        Apply dropping to $z_j^{(i)}$
    EndDo
EndDo
Let $z_i := z_i^{(i-1)}$ and $p_i := p_i^{(i-1)}$, for $i = 1, 2, \cdots, n$.
Return $Z = [z_1, z_2, \cdots, z_n]$ and $D = diag(p_1, p_2, \cdots, p_n)$.
```

Obviously, the robust (referred to as SAINV) and plain algorithm are mathematically equivalent. However, the incomplete process obtained by dropping in the z-vectors in step (9) of the robust algorithm leads to a reliable approximate inverse. This algorithm, in exact arithmetic, is applicable to any SPD matrix without breakdowns. The computational cost of SAINV is higher than basic AINV and special care has to be taken to do sparse-sparse matrix-vector multiply.

Incomplete Factorization by SAINV Consider now the exact algorithm (with no dropping) and write $A = LDL^T$ with L unit lower triangular and D diagonal. Observe that L in the LDL^T factorization of A and the inverse factor satisfy

$$AZ = LD \quad \text{or} \quad L = AZD^{-1} \tag{110.25}$$

where D is the diagonal matrix containing the pivots. This easily follows from

$$Z^T AZ = D \quad \text{and} \quad Z^T = L^{-1} \tag{110.26}$$

If we recall that pivot $d_j = p_j = z_j^T Az_j = \langle Az_j, z_j\rangle$, then by equating corresponding entries of AZD^{-1} and $L = [l_{ij}]$ we find that (Benzi and Tůma, 2003; Bollhöfer and Saad, 2001)

$$l_{ij} = \frac{\langle Az_j, z_i\rangle}{\langle Az_j, z_j\rangle} \quad i \geq j \tag{110.27}$$

Hence, the L factor of A can be obtained as a by-product of the A-orthogonal-ization, at no extra cost. In the implementation of SAINV, the quantities l_{ij} in Equation 110.27 are the multipliers that are used in updating the columns of Z. Once the update is computed, they are no longer needed and are discarded. To obtain an incomplete factorization of A, we do just the opposite; we save the multipliers l_{ij} , and discard the column vectors z_j as soon as they have been computed and operated with. Hence, the incomplete L factor is computed by columns; these columns can be stored in place of the z_j vectors, with minimal modifications to the code. Here, we are assuming that the right-looking form of SAINV is being used. If the left-looking one is being used, then L would be computed by rows. Please refer to Benzi and Tůma (2003) for more implementation details.

110.5.5 Numerical Experiments

Matrices from soil-structure interaction finite element analysis have been extracted from simulation system to study the performance of different preconditioning techniques on PCG method. The prototype of soil structure model has been shown in Figures 110.56 and 110.57. In order to introduce both of the nonlinear theories for soil and structures, we use continuum elements to model the soil and beam elements for the structures. Matrices from static pushover analysis and dynamic ground motion analysis have been collected for this research.

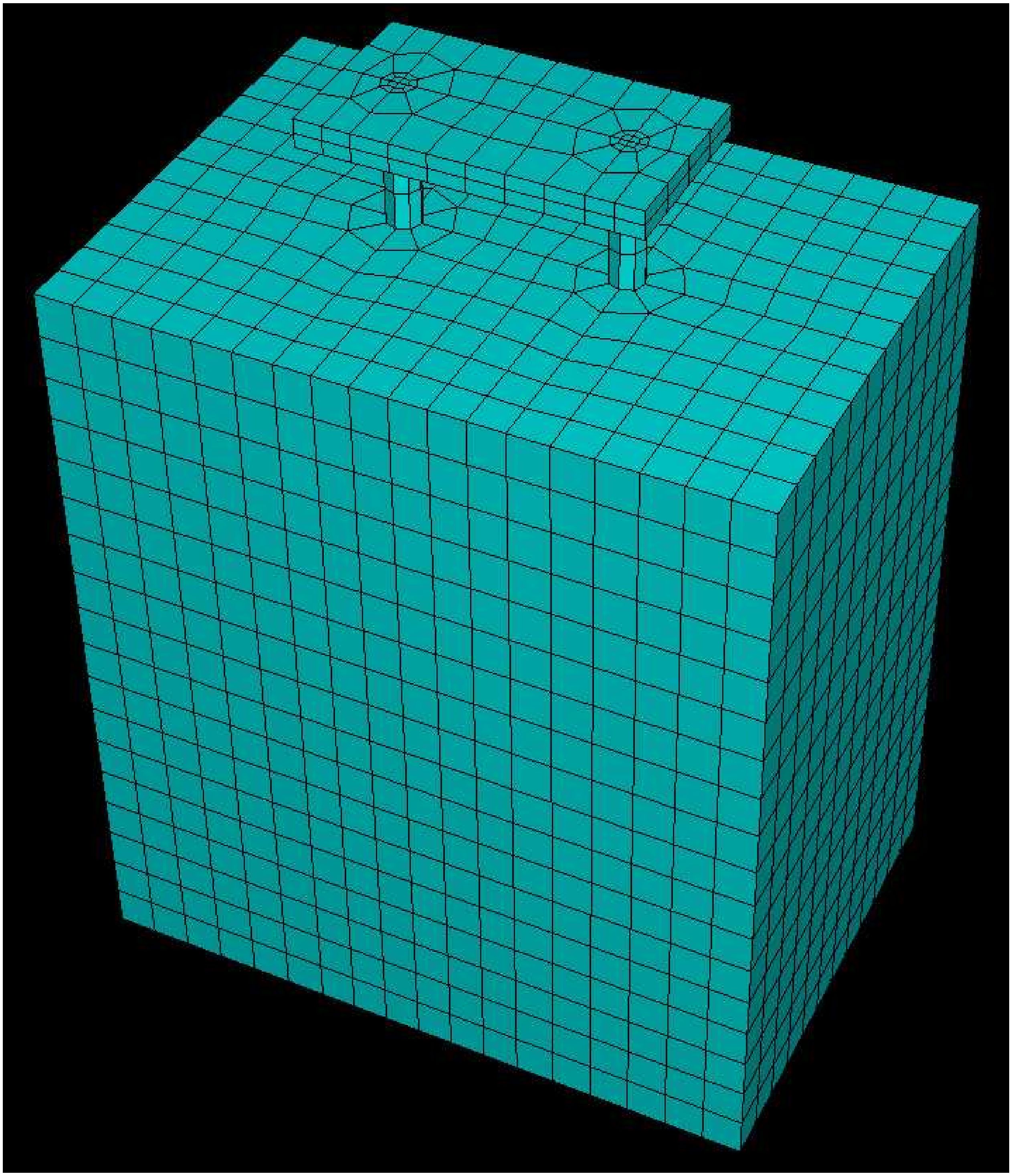


Figure 110.56: Finite Element Mesh of Soil-Structure Interaction Model

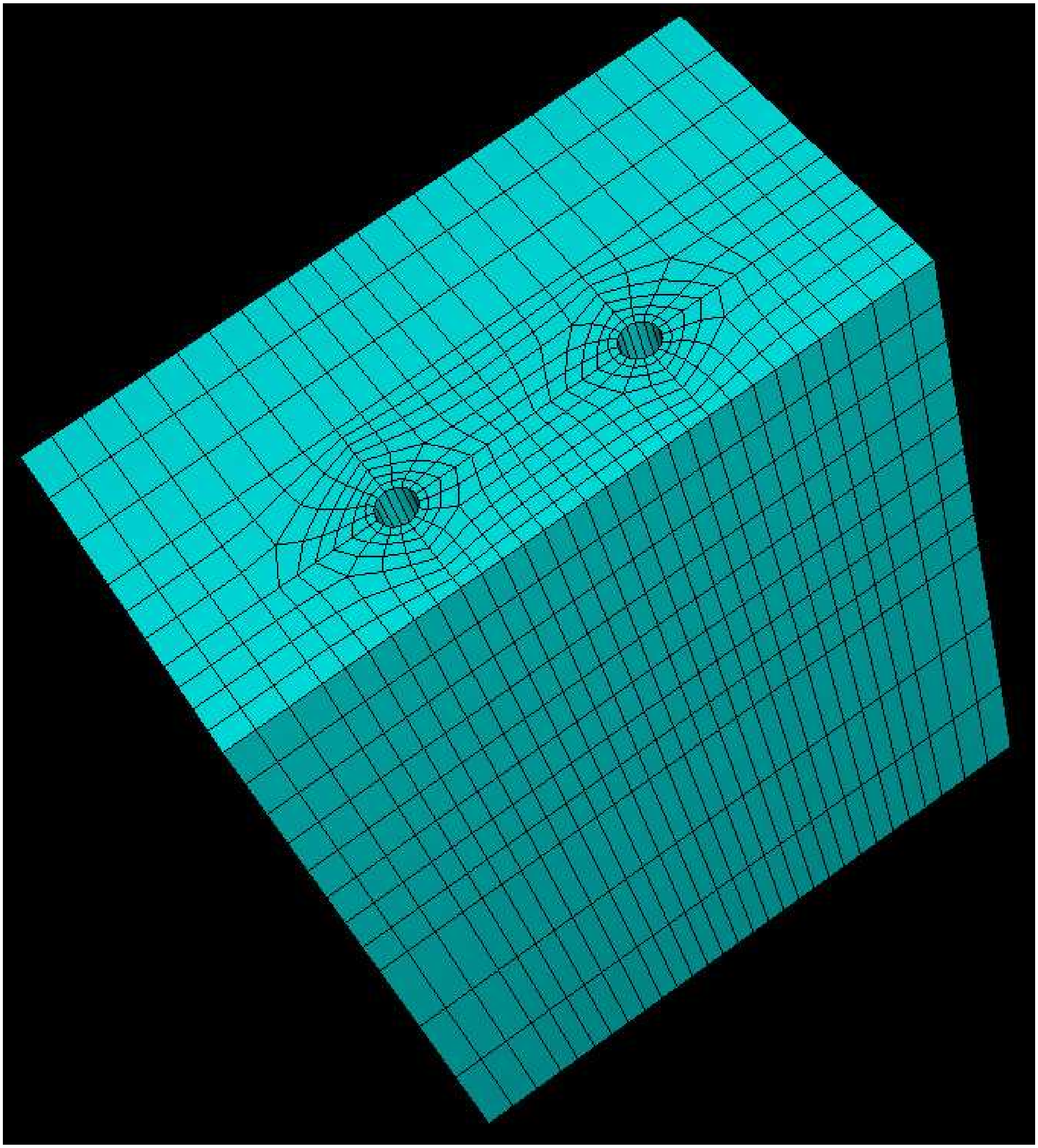


Figure 110.57: Finite Element Mesh of Soil-Structure Interaction Model

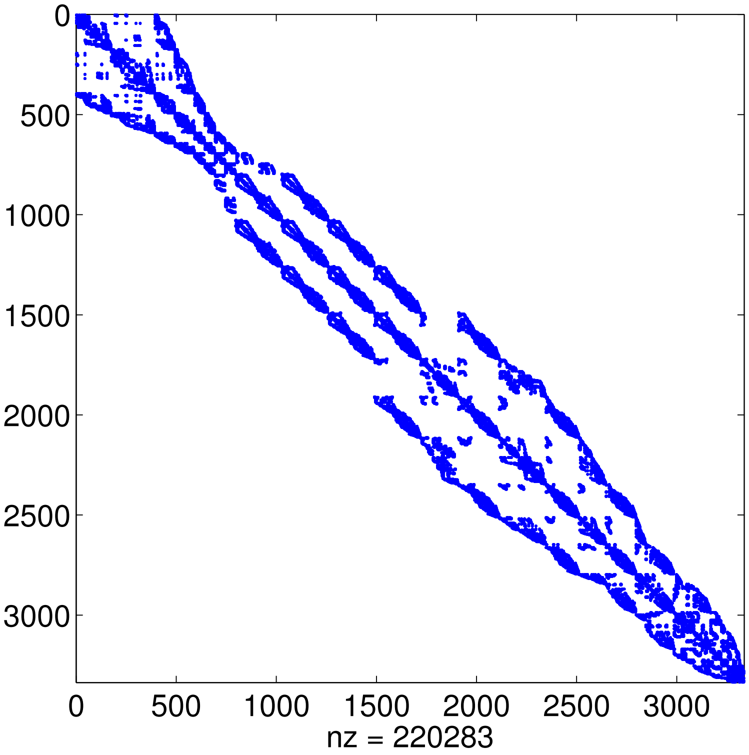


Figure 110.58: Matrices $N = 3336$ (Continuum FEM)

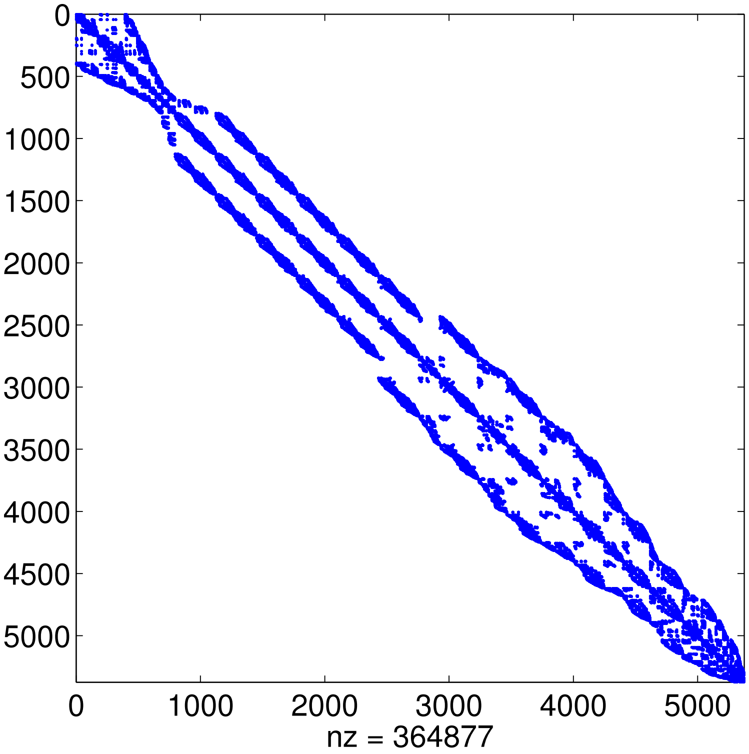


Figure 110.59: Matrices $N = 5373$ (Continuum FEM)

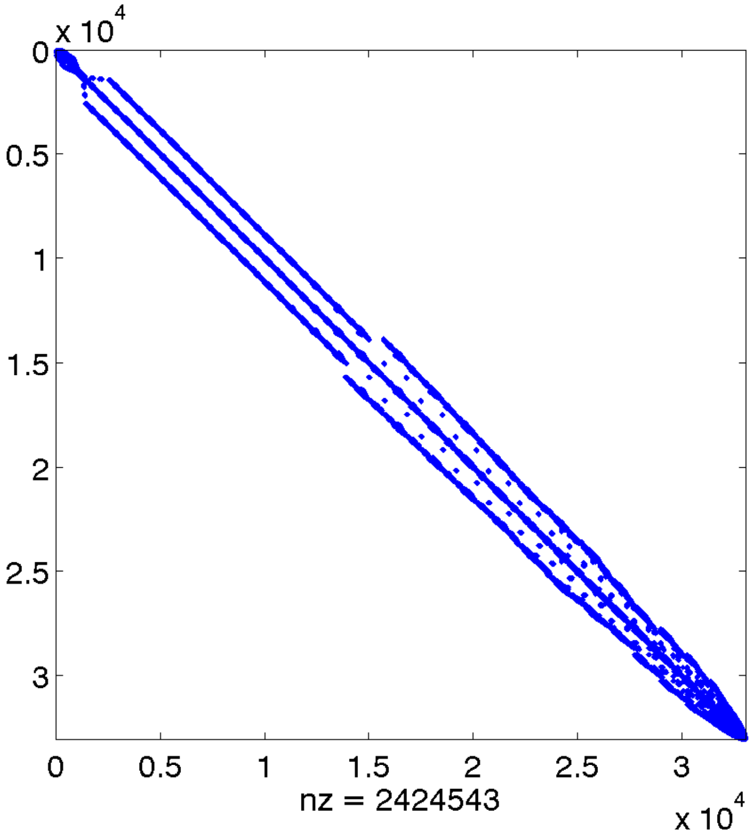


Figure 110.60: Matrices $N = 33081$ (Continuum FEM)

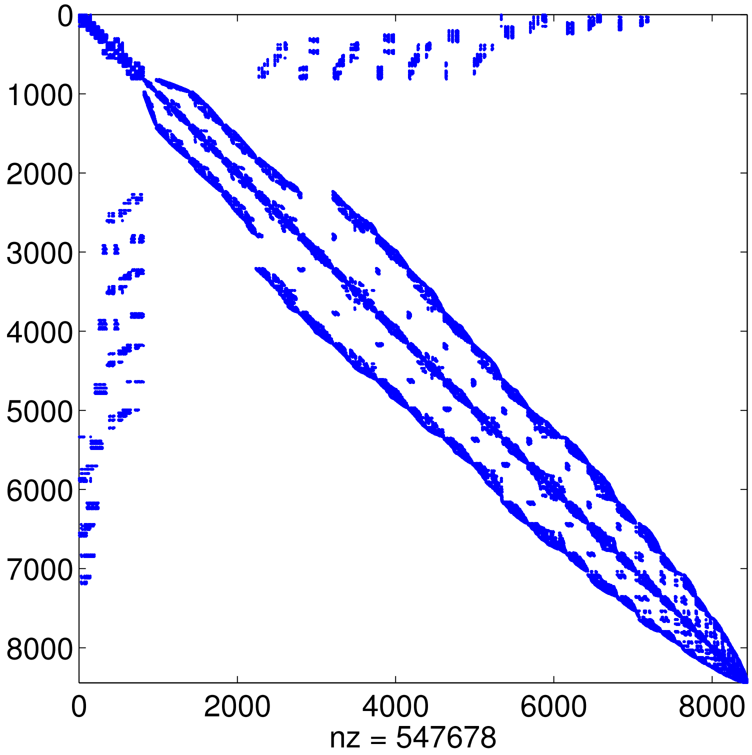


Figure 110.61: Matrices $N = 8842$ (Soil-Beam Static FEM)

Table 110.7: Matrices in FEM Models

Continuum Model (Static)			
Matrix	Property	Dimension	# Nonzeros
m1188	SPD	3336	220283
m1968	SPD	5373	364877
m11952	SPD	33081	2424543
Soil-Beam Model (Static and Dynamic)			
Matrix	Property	Dimension	# Nonzeros
SoilBeam	SPD	8442	547678
SoilBeamDyn	SPD	8442	547671

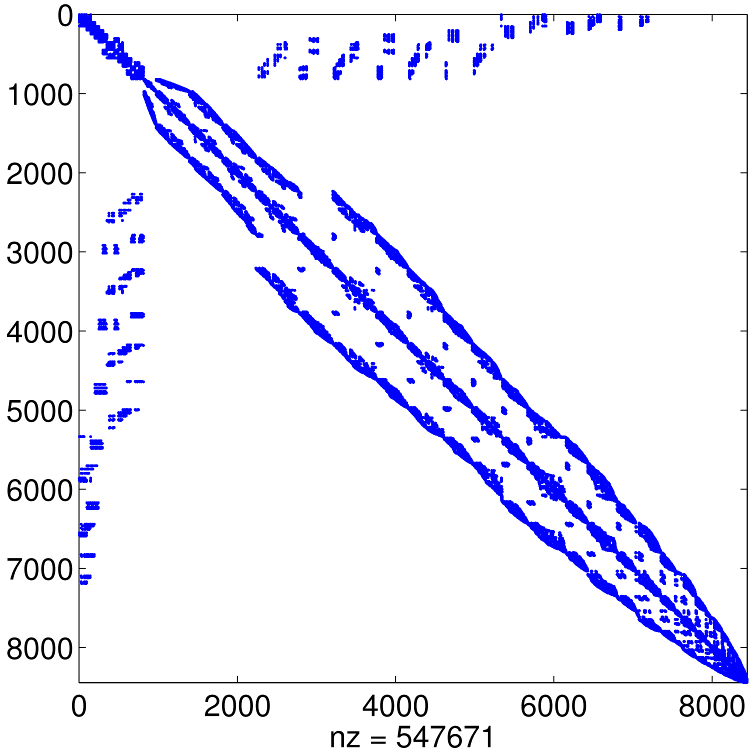


Figure 110.62: Matrices $N = 8842$ (Soil-Beam Dynamic FEM)

SPD matrices have been studied using Conjugate Gradient method with or without preconditioning. Performance has been summarized in Table 110.8.

Table 110.8: Performance of CG and PCG Method (Continuum FEM)

3336 DOFs FEM (Static)					
Preconditioner	# Iter	Pre Time(s)	Iter Time(s)	Total Time(s)	Density[2]
-	4376	-	54.82	54.82	-
Jacobi	1612	0.01	20.18	20.19	-
IC(0)	413	2.19	11.07	13.26	1.00
IC(1e-6)	5	5.90	0.47	6.37	5.88
RIF2(1e-2)	571	9.37	14.94	24.31	0.94
RIF3(1e-2)	541	6.80	14.13	20.93	0.94

5373 DOFs FEM (Static)					
Preconditioner	# Iter	Pre Time(s)	Iter Time(s)	Total Time(s)	Density
-	4941	-	103.78	103.78	-
Jacobi	1711	0.01	36.61	36.62	-
IC(0)	437	6.5	20.38	26.88	1.00
IC(1e-6)	6	19.81	1.3	21.11	8.10
RIF2(1e-2)	599	25.71	26.55	52.26	0.96
RIF3(1e-2)	566	21.31	25.23	46.54	0.96

33081 DOFs FEM (Static)					
Preconditioner	# Iter	Pre Time(s)	Iter Time(s)	Total Time(s)	Density
-	6754	-	952.53	952.53	-
Jacobi	2109	0.03	308.46	308.49	-
IC(0)	565	273.83	173.05	446.88	1.00
IC(1e-6)[3]					
RIF2(1e-2)	694	1172.7	211.88	1384.58	0.99
RIF3(1e-2)	664	1245.4	202.67	1448.07	0.99

[2] Density is defined as the number of non-zeros of the incomplete factor divided by the number of non-zeros in the lower triangular part of A.

[3] Could not continue because memory requirement larger than 1.4GB.

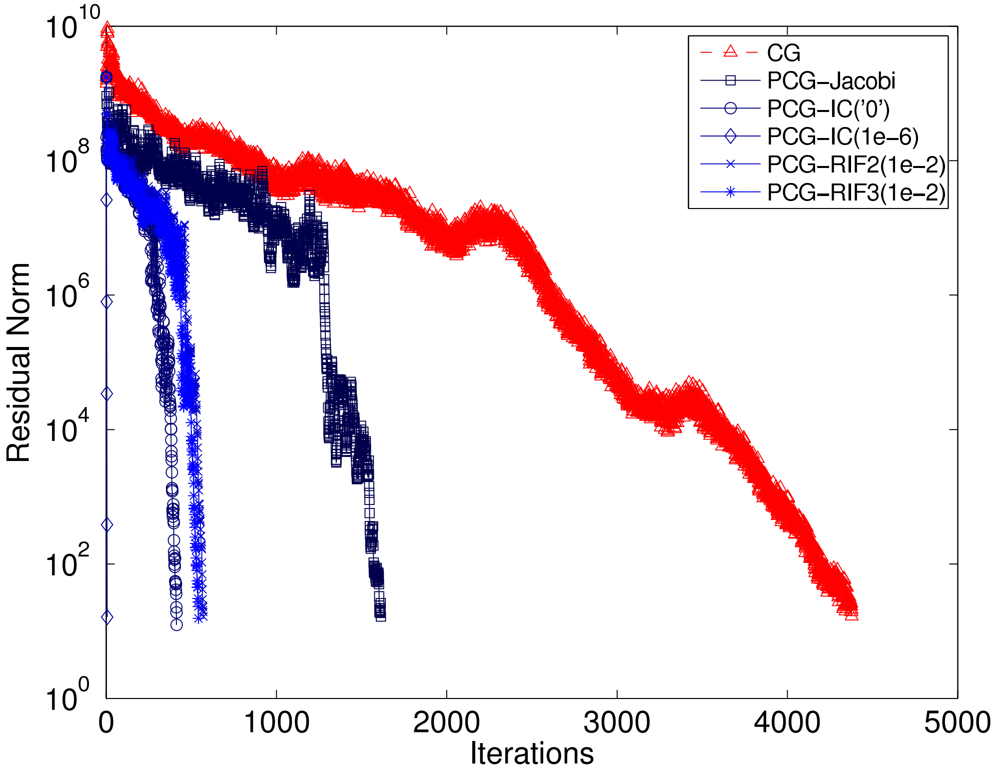


Figure 110.63: Convergence of CG and PCG Method (3336 DOFs Model)

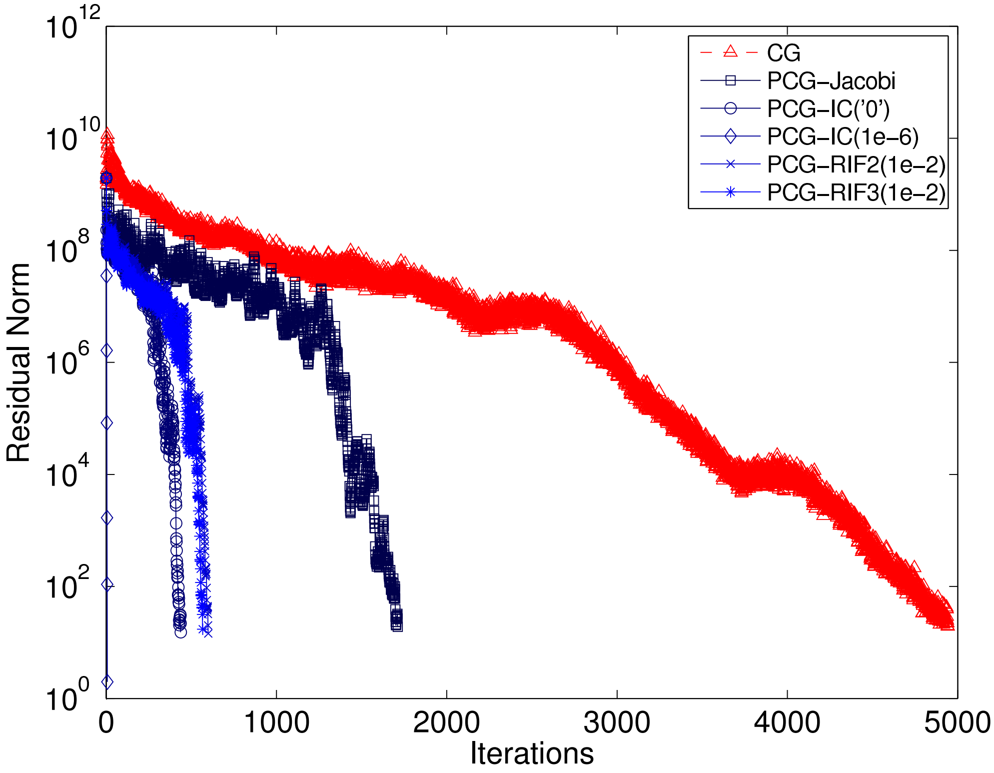


Figure 110.64: Convergence of CG and PCG Method (5373 DOFs Model)

Table 110.9: Performance of CG and PCG Method (Soil-Beam FEM)

8842 DOFs Soil-Beam FEM (Static)

Preconditioner	# Iter	Pre Time(s)	Iter Time(s)	Total Time(s)	Density[4]
-	3274	-	102.5	102.5	-
Jacobi	1687	0.01	54.56	54.57	-
IC(0)	26	15.77	1.95	17.72	1.00
IC(1e-6)	6	110.17	2.79	112.96	15.11
RIF2(1e-6)[5]	23	3364.8	3.44	3368.24	4.32
RIF3(1e-6)[5]	31	34541	9.26	34550.26	16.37

8842 DOFs Soil-Beam FEM (Dynamic)

Preconditioner	# Iter	Pre Time(s)	Iter Time(s)	Total Time(s)	Density
-	3276	-	136.7	136.7	-
Jacobi	MaxIt				
IC(0)	MaxIt				
IC(1e-6)	MaxIt				
RIF2(1e-2)	MaxIt				
RIF3(1e-2)	MaxIt				

110.5.6 Conclusion and Future Work

1. For the soil-structure interaction problems investigated in this section, Conjugate Gradient method works fine and the convergence is acceptable for most cases.

2. Incomplete Cholesky factorization preconditioner has been shown to be very powerful in static pushover problems.

3. Dynamic problems formulated by Newmark integration scheme have not been extensively tested. But according to the data available so far, neither IC nor RIF preconditioners performed well and further testing is necessary to reach a more persuasive conclusion. The difficulty in dynamic analysis results from the fact that consistent mass and damping matrices used in continuum finite element formulations significantly degrade the conditioning number of the final system. This situation deteriorates when penalty handler is used to apply multiple point constraints, which introduces huge off-diagonal numbers to stiffness, mass and damping matrices (Cook et al., 2002).

4. Robust incomplete factorization preconditioning based on A-orthogonalization has not been shown competitive with IC preconditioners in this research. It is also worth noting that all timings are

---

[4] Density is defined as the number of non-zeros of the incomplete factor divided by the number of non-zeros in the lower triangular part of A.

[5] Iteration with tolerance 1e-2 failed to converge.

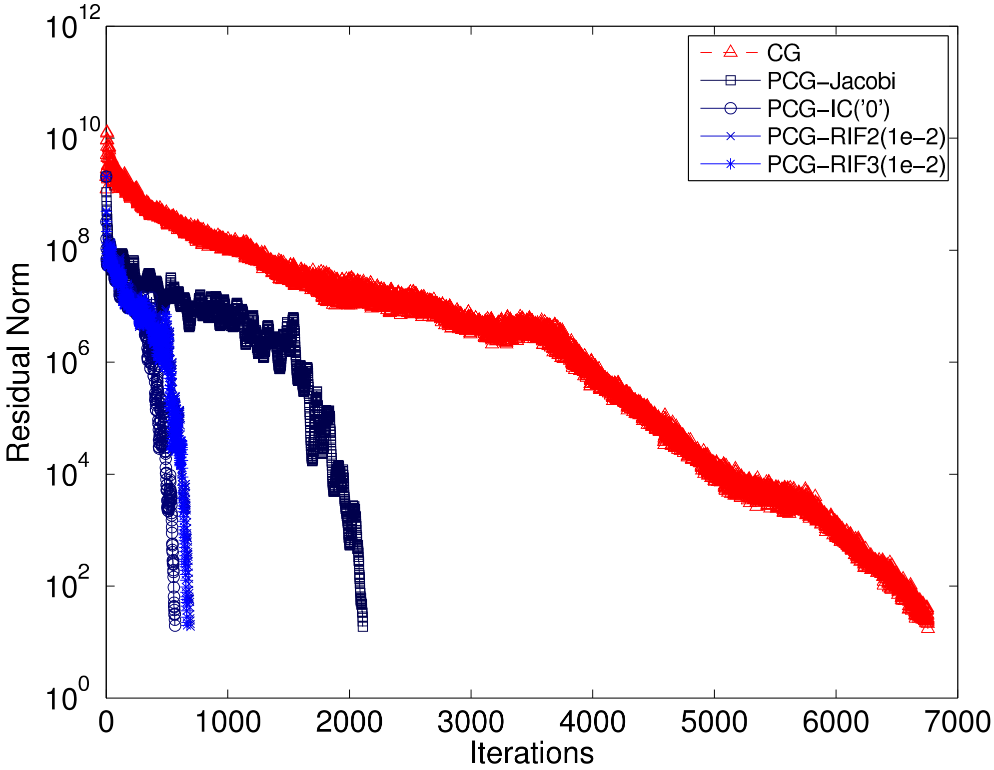


Figure 110.65: Convergence of CG and PCG Method (33081 DOFs Model)

taken in MATLAB. There are much more improvement can be achieved with a carefully coded FORTRAN program.

5. Static analysis has been extensively studied and it can be safely concluded that IC(0) and Jacobi preconditioners are good choices for the nonlinear soil-beam interaction simulations.

6. Dynamic analysis has also been studied but more work is needed to draw any detailed conclusion. Generally speaking, one should be alert if iterative solver is to be used for dynamic analysis. This partially comes from the fact that mass and damping matrices undoubtedly alter the structures of the coefficient matrix. This situation becomes more complicated if penalty handler is used to introduce off-diagonal numbers when handling multi-point constraints. So direct solver would be a more stable option for solving dynamic equations.

110.6 Performance Study on Parallel Direct/Iterative Solving in SFSI

The motivation of this section is to introduce a robust and efficient parallel equation solver into our parallel finite element analysis framework. Aside from sparsity, which has been well known as the

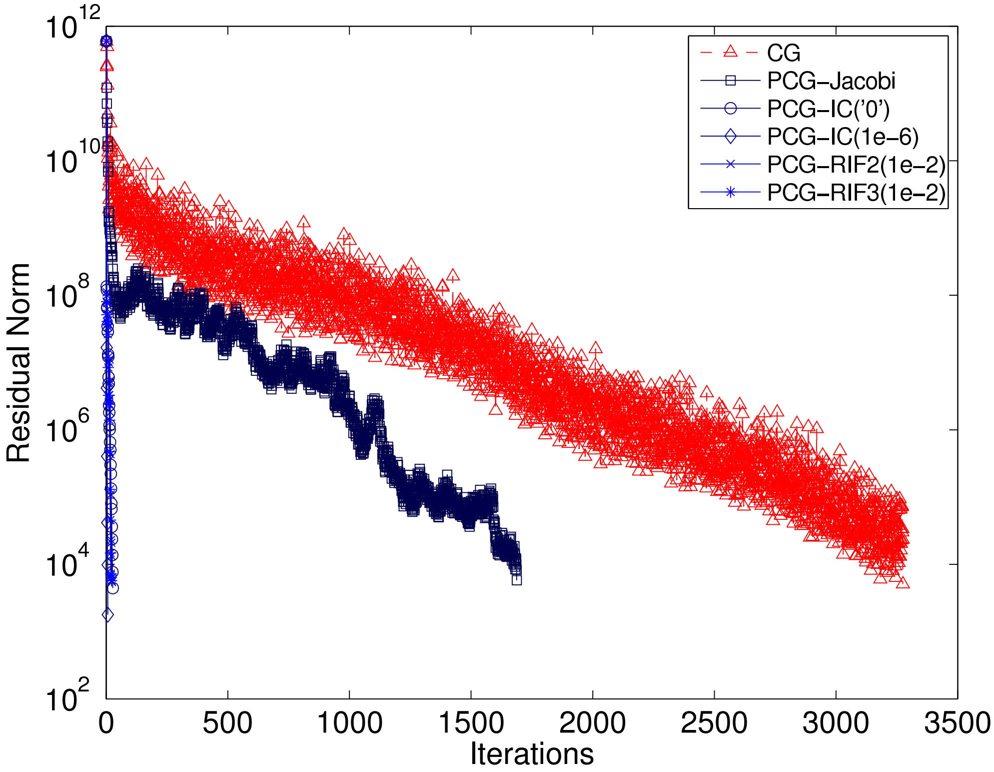


Figure 110.66: Convergence of CG and PCG Method (Soil-Beam Static Model)

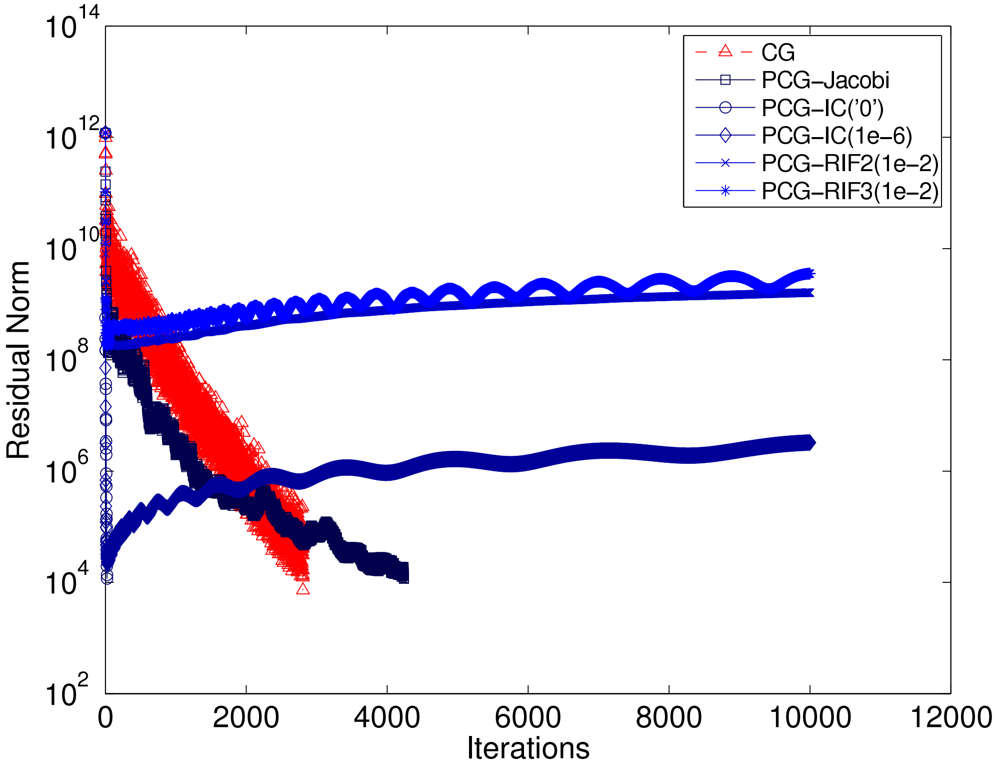


Figure 110.67: Convergence of CG and PCG Method (Soil-Beam Dynamic Model)

result of compact support that is inherent with finite element method, there exist some other special considerations that make the equation solving in finite element simulation a more involved problem.

In nonlinear finite element simulations, handling of constraints significantly affects the condition number of assembled equation systems. In SFSI simulations, multiple-point constraint is necessary to enforce the connection between soil and pile elements. In this research, penalty handler has been adopted to impose multiple point constraints on the assembled equation systems. Transformation and Lagrange multipliers are among those popular methods as well (Belytschko et al., 2000; Cook et al., 2002). The method of Lagrange multipliers adds extra constraints to the system and the resulted coefficient matrix will lose symmetric positive definiteness. Transformation is favorable especially in the sense that it reduces the order of the equation systems by condensing out slave/constrained DOFs. But the transformation is the most difficult to code and the situation of one single main/retained node with multiple follower/constrained nodes further complicates the problem.

Penalty method is chosen in this research due to the fact that it well preserves the symmetric positive definiteness of the system if the nice property is observed. Another consideration comes from the easiness with which the penalty methods can handle the single main/retained multiple follower/constrained situations. This is proven to be extremely valuable when data redistribution is required in adaptive parallel processing because the DOF_Graph object can be clearly tracked during partition and repartition phases.

The incapability of handling constraints accurately has been long known as the weakness of penalty method. The choice of the key penalty number seems arbitrary and largely depends on experience. The dilemma is with larger penalty number, the system can handle constraints more accurately while the coefficient matrix can become very ill-conditioned. This can lead to serious convergence problem for iterative solvers.

The majority of coefficient matrices resulted from finite element analysis are inherently symmetric positive definite, for which lots of numerical algorithms have been proposed and solving SPD, symmetric or closely symmetric systems has been relatively maturer than more common unsymmetric cases. Unfortunately, in geotechnical finite element simulations, unassociated constitutive models lead to unsymmetric stiffness matrices (Jeremić, 2004). More general parallel solvers must be coded to solve the problem.

In this section, both iterative and direct solvers are coded using the consistent PETSc interface (Balay et al., 2001, 2004, 1997). Popular direct solvers for general unsymmetric systems such as MUMPS, SPOOLES, SuperLU, PLAPACK have been introduced and performance study has been carried out to investigate the efficiency of different solvers on large scale SFSI simulations with penalty-handled unsymmetric equation systems. GMRES is always the first choice of iterative method when general unsymmetric systems are concerned. Preconditioning techniques have been thoroughly studied in this research to explore possible advantage of preconditioned iterative solver over direct solving. Jacobi, incomplete LU decomposition and approximate inverse preconditioners represent the most popular choices for Krylov methods and they are chosen in this performance survey.

All numerical algorithms have been implemented through interface of PETSc, which provides a consistent platform on which implementation issues can be avoided to expose individual algorithmic performance.

110.6.1 Parallel Sparse Direct Equation Solvers

The methods that we consider for the solution of sparse linear equations can be grouped into four main categories: general techniques, frontal methods, multifrontal approaches and supernodal algorithms (Dongarra et al., 1996).

110.6.1.1 General Techniques – SPOOLES

The so-called general approach can be viewed as parallel versions of sparse LU decomposition. Special cares must be taken to handle the sparse data structures. Sparsity ordering is crucial in parallel sparse equation solving in order to reduce fill-in and discover large-grain parallelism (Demmel et al., 1993).

Freely available package SPOOLES provides minimum degree (multiple external minimum degree (Liu, 1985)), generalized nested dissection and multisection ordering schemes for matrix sparsity ordering. Fundamental supernode tree built on top of vertex elimination tree is used to explore granularity in parallel (Ashcraft, 1999; Ashcraft et al., 1999).

110.6.1.2 Frontal and Multifrontal Methods – MUMPS

Frontal methods have their origins in the solution of finite element problems from structural analysis. The usual way to describe the frontal method is to view its application to finite element problems where the matrix A is expected as a sum of contributions from the elements of a finite element discretization (Dongarra et al., 1996). That is,

$$A = \sum_{l=1}^{m} A^{[l]}, \tag{110.28}$$

where $A^{[l]}$ is nonzero only in those rows and columns that correspond to variables in the lth element. If a_{ij} and $a_{ij}^{[l]}$ denote the (i,j)th entry of A and $A^{[l]}$, respectively, the basic assembly operation when forming A is of the form

$$a_{ij} \Leftarrow a_{ij} + a_{ij}^{[l]}. \tag{110.29}$$

It is evident that the basic operation in Gaussian elimination

$$a_{ij} \Leftarrow a_{ij} + a_{ip}[a_{pp}]^{-1}a_{pj}. \tag{110.30}$$

may be performed as soon as all the the terms in the triple product 110.30 are fully summed (that is, are involved in no more sums of the form 110.29). The assembly and Gaussian elimination processes can therefore be interleaved and the matrix A is never assembled explicitly. This allows all intermediate working to be performed in a dense matrix, termed frontal matrix, whose rows and columns correspond to variables that have not yet been eliminated but occur in at least one of the elements that have been assembled.

For general problems other than finite element, the rows of A (equations) are added into the frontal matrix one at a time. A variable is regarded as fully summed whenever the equation in which it last appears is assembled. The frontal matrix will, in this case, be rectangular.

The idea of multifrontal method is to couple a sparsity ordering with the efficiency of a frontal matrix kernel so allowing good exploitation of high performance computers. The basic approach is to develop separate fronts simultaneously which can be chosen using a sparsity preserving ordering such as minimum degree.

Elimination tree, again is the most important notion in the factorization process and also utilized to discover the potential of parallelism. An elimination tree defines the a precedence order within the factorization. The factorization commences at the leaves of of the tree and data is passed towards the root along the edges in the tree. To complete the work associated with a node, all the data must have been obtained from the children of the node, otherwise work at different nodes is independent.

Freely available package MUMPS (MUltifrontal Massively Parallel sparse direct Solver) has been used in this research to investigate the performance of multifrontal methods (http://graal.ens-lyon.fr/MUMPS/, 2006).

MUMPS is a package for solving systems of linear equations of the form $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ is a square sparse matrix that can be either unsymmetric, symmetric positive definite, or general symmetric. MUMPS uses a multifrontal technique which is a direct method based on either the $\mathbf{LU}$ or the $\mathbf{LDL^T}$ factorization of the matrix. MUMPS exploits both parallelism arising from sparsity in the matrix $\mathbf{A}$ and from dense factorizations kernels.

The main features of the MUMPS package include the solution of the transposed system, input of the matrix in assembled format (distributed or centralized) or elemental format, error analysis, iterative refinement, scaling of the original matrix, and return of a Schur complement matrix. MUMPS offers several built-in ordering algorithms, a tight interface to some external ordering packages such as METIS and PORD, and the possibility for the user to input a given ordering. Finally, MUMPS is available in various arithmetics (real or complex, single or double precision).

The software is written in Fortran 90 although a C interface is available. The parallel version of MUMPS requires MPI for message passing and makes use of the BLAS, BLACS, and ScaLAPACK libraries. The sequential version only relies on BLAS.

MUMPS distributes the work tasks among the processors, but an identified processor (the host) is required to perform most of the analysis phase, to distribute the incoming matrix to the other processors (slaves) in the case where the matrix is centralized, and to collect the solution. The system $\mathbf{Ax} = \mathbf{b}$ is solved in three main steps:

1. Analysis. The host performs an ordering based on the symmetrized pattern $\mathbf{A} + \mathbf{A^T}$, and carries out symbolic factorization. A mapping of the multifrontal computational graph is then computed, and symbolic information is transferred from the host to the other processors. Using this information, the processors estimate the memory necessary for factorization and solution.

2. Factorization. The original matrix is first distributed to processors that will participate in the numerical factorization. The numerical factorization on each frontal matrix is conducted by a main compute processor (determined by the analysis phase) and one or more slave processors (determined dynamically). Each processor allocates an array for contribution blocks and factors; the factors must be kept for the solution phase.

3. Solution. The right-hand side $\mathbf{b}$ is broadcast from the host to the other processors. These processors compute the solution $\mathbf{x}$ using the (distributed) factors computed during Step 2, and the solution is either assembled on the host or kept distributed on the processors.

Each of these phases can be called separately and several instances of MUMPS can be handled simultaneously. MUMPS allows the host processor to participate in computations during the factorization and solve phases, just like any other processor.

For both the symmetric and the unsymmetric algorithms used in the code, a fully asynchronous approach with dynamic scheduling of the computational tasks has been chosen. Asynchronous communication is used to enable overlapping between communication and computation. Dynamic scheduling was initially chosen to accommodate numerical pivoting in the factorization. The other important reason for this choice was that, with dynamic scheduling, the algorithm can adapt itself at execution time to remap work and data to more appropriate processors. In fact, the main features of static and dynamic approaches have been combined and the estimation obtained during the analysis to map some of the main computational tasks has been used; the other tasks are dynamically scheduled at execution time. The main data structures (the original matrix and the factors) are similarly partially mapped according to the analysis phase.

110.6.1.3 Supernodal Algorithm – SuperLU

The left-looking or column Cholesky algorithm can be implemented for sparse system and can be blocked by using a supernodal formulation. The idea of a supernode is to group together columns with the same nonzero structure, so they can be treated as a dense matrix for storage and computation. Supernodes were originally used for (symmetric) sparse Cholesky factorization (Demmel et al., 1999a). In the factorization $\mathbf{A} = \mathbf{L}\mathbf{L}^{\mathbf{T}}$ (or $\mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{L}^{\mathbf{T}}$), a supernode is a range $(r : s)$ of columns of $\mathbf{L}$ with the same nonzero structure below the diagonal; that is, $\mathbf{L}(r : s; r : s)$ is full lower triangular and every row of $\mathbf{L}(r : s; r : s)$ is either full or zero.

Then in left-looking Cholesky algorithm, all the updates from columns of a supernode are summed into a dense vector before the sparse update is performed. This reduces indirect addressing and allows the inner loops to be unrolled. In effect, a sequence of col-col updates is replaced by a supernode-column (sup-col) update. The sup-col update can be implemented using a call to a standard dense Level 2 BLAS matrix-vector multiplication kernel. This idea can be further extended to supernode-supernode (sup-sup) updates, which can be implemented using a Level 3 BLAS dense matrix-matrix kernel. This can reduce memory traffic by an order of magnitude, because a supernode in the cache can participate in multiple column updates (Demmel et al., 1999a). It has been reported in (Ng and Peyton, 1993) that a sparse Cholesky algorithm based on sup-sup updates typically runs 2.5 to 4.5 times as fast as a col-col algorithm. Indeed, supernodes have become a standard tool in sparse Cholesky factorization.

To sum up, supernodes as the source of updates help because of the following (Demmel et al., 1999a):

1. The inner loop (over rows) has no indirect addressing. (Sparse Level 1 BLAS is replaced by dense Level 1 BLAS.)
2. The outer loop (over columns in the supernode) can be unrolled to save memory references. (Level 1 BLAS is replaced by Level 2 BLAS.)
   Supernodes as the destination of updates help because of the following:
3. Elements of the source supernode can be reused in multiple columns of the destination supernode to reduce cache misses. (Level 2 BLAS is replaced by Level 3 BLAS.)

Supernodes in sparse Cholesky can be determined during symbolic factorization, before the numeric factorization begins. However, in sparse $\mathbf{LU}$, the nonzero structure cannot be predicted before numeric factorization, so supernodes must be defined dynamically. Furthermore, since the factors $\mathbf{L}$ and $\mathbf{U}$ are no longer transposes of each other, the definition of a supernode must be generalized.

Freely available package SuperLU proposed a couple of ways to generalize the symmetric definition of supernodes to unsymmetric factorization (Demmel et al., 1999a). It is now not possible to use Level 3 BLAS efficiently for unsymmetric systems. The implementation in SuperLU performs a dense matrix multiplication of a block of vectors and, although these can not be written as another dense matrix, it has been shown that this Level 2.5 BLAS has most of the performance characteristics of Level 3 BLAS since the repeated use of the same dense matrix allows good use of cache and memory hierarchy.

There are three versions of libraries collectively referred as SuperLU (Demmel et al., 2003),

- Sequential SuperLU is designed for sequential processors with one or more layers of memory hierarchy (caches).
- Multithreaded SuperLU (SuperLU_MT) is designed for shared memory multiprocessors (SMPs), and can effectively use up to 16 or 32 parallel processors on sufficiently large matrices in order to speed up the computation (Demmel et al., 1999b).
- Distributed SuperLU SuperLU_DIST is designed for distributed memory parallel processors, using MPI for interprocess communication. It can effectively use hundreds of parallel processors on sufficiently large matrices (Li and Demmel, 2003).

Parallelizing sparse direct solver for unsymmetric systems is more complicated than parallel sparse Cholesky case. The advantage of sparse Cholesky over the unsymmetric case is that pivots can be chosen in any order from the main diagonal while guaranteeing stability. This lets us perform pivot choice before numerical factorization begins, in order to minimize fill-in, maximize parallelism. precompute the nonzero structure of the Cholesky factor, and optimize the (2D) distributed data structures and communication pattern (Li and Demmel, 2003).

In contrast, for unsymmetric or indefinite systems, distributed memory codes can be much more complicated for at least two reasons. First and foremost, some kind of numerical pivoting is necessary for stability. Classical partial pivoting or the sparse variant of threshold pivoting typically cause the fill-ins and workload to be generated dynamically during factorization. Therefore, we must either design dynamic data structures and algorithms to accommodate these fill-ins, or else use static data structures which can grossly overestimate the true fill-in. The second complication is the need to handle two factored matrices $\mathbf{L}$ and $\mathbf{U}$, which are structurally different yet closely related to each other in the filled pattern. Unlike the Cholesky factor whose minimum graph representation is a tree (elimination tree), the minimum graph representations of the $\mathbf{L}$ and $\mathbf{U}$ factors are directed acyclic graphs (elimination DAGs).

In SuperLU_DIST, a static pivoting approach, called GESP (Gaussian Elimination with Static Pivoting) (Li and Demmel, 1998) is used. In order to parallelize the GESP algorithm, a 2D block-cyclic mapping of a sparse matrix to the processors is used. An efficient pipelined algorithm is also designed to perform parallel factorization. With GESP, the parallel algorithm and code are much simpler than dynamic pivoting.

The main algorithmic features of SuperLU_DIST solver are summarized as follows (Li and Demmel, 2003):

- supernodal fan-out (right-looking) based on elimination DAGs,
- static pivoting with possible half-precision perturbations on the diagonal,
- use of an iterative algorithm using the LU factors as a preconditioner, in order to guarantee stability,
- static 2D irregular block-cyclic mapping using supernodal structure, and
- loosely synchronous scheduling with pipelining.

In particular, static pivoting can be performed before numerical numerical factorization, allowing us to use all the techniques in good sparse Cholesky codes: choice of a (symmetric) permutation to minimize fill-in and maximize parallelism, precomputation of the fill pattern and optimization of 2D distributed data structures and communication patterns. Users are referred to Li and Demmel (2003) for algorithm details.

110.6.2 Performance Study on SFSI Systems

In this section, performance study on popular parallel direct and iterative solvers has been conducted. The purpose is to provide some guidelines on appropriate use of different solvers with the parallel finite simulation framework. Matrix systems from SFSI analysis are used as test cases. The performance investigation uses IA64 Intel-based cluster at SDSC.

110.6.2.1 Equation System

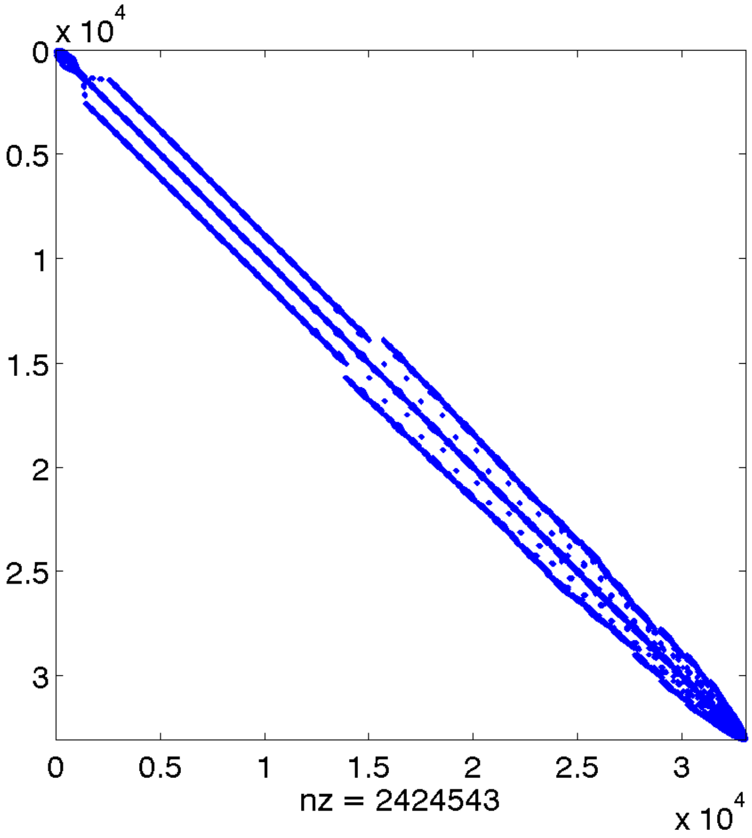


Figure 110.68: Matrices $N = 33081$ (Continuum FEM)

110.6.2.2 Performance Results

Table 110.10: Performance Study on SFSI Systems (N=33081)

Direct Solvers		
Solvers	Num of CPUs	Time (s)
MUMPS	4	6.0312
	8	7.0534
	16	5.3472
SuperLU_DIST	4	20.358
	8	13.803
	16	13.755
SPOOLES	4	10.696
	8	7.5338
	16	6.2448
Iterative Solvers (GMRES)		
Preconditioner	Num of CPUs	Time (s)
Jacobi	16	96.441
Parallel ILU(0)	4	277.49
	8	276.07
	16	135.78

110.6.3 Conclusion

This section presents the parallel solvers implemented in parallel finite element framework. Table 110.10, draws several conclusions about appropriate use of solvers:

- Direct solvers outperforms the iterative solver significantly for general cases. It is worthwhile to note that nonsymmetric solvers are used here due to their generality. For special cases such as SPD system, preconditioned CG will show much better performance.
- The Conjugate Gradient method applies only to Symmetric Positive Definite (SPD) system. This puts restriction on the material models we can use in our simulations. Generally speaking, elastic material will yield a SPD stiffness matrix. Plastic material with associative flow rule also satisfies

this category. Plastic material with non-associative flow rule has non-symmetric element stiffness matrix and so will be the global coefficient matrix of the equation system.

- Another category of matrix that deserves attention is the stiffness matrix from softening materials, which possesses at least one negative eigenvalue so the SPD property will be broken. For advanced geo-materials subject to complicated loadings, as the material develops nonlinearity, the condition of stiffness matrix might vary greatly from SPD (elastic phase), to singular (elastic-perfectly-plastic), and non-symmetric non-positive-definite (elastic-non-associative-plastic-softening) cases. This poses another challenge when one tries to use iterative solver for production runs. The unpredictability of stiffness matrix will disable the application of powerful solvers such as Conjugate Gradient for iterative case and Cholesky for direct case.
- The reason why iterative solver exhibits poor efficiency is partly due to the problem size. We can also see from the Table 110.10 that parallel direct solver is not scalable in general. Iterative solver, on the other hand, is more scalable and it is safe to project that when the size of matrix increases, iterative solver has the advantage from the memory requirement point of view.

Parallel equation solving itself is a complicated topic in numerical computing community. In this section, the main purpose is to introduce a robust and generally efficient parallel solver for finite element simulations. So in this sense, parallel direct solvers such as MUMPS and SPOOLES are recommended.

Chapter 111

Solid, Structure – Fluid Interaction

(2017-2019-2021-)

(In collaboration with Dr. Hexiang Wang)

111.1 Chapter Summary and Highlights

111.2 Introduction

The analysis of problems several civil engineering fields often requires a study of fluid-structure systems that are excited by dynamic loads. For example, the evaluation of the structural integrity of nuclear reactor components involves the analysis of structures of complex shape and their interaction with the fluid in which they are embedded Donea et al. (1982). In these cases, both the fluid and the structure might undergo non-linear response.

These needs for safety evaluations have motivated the development of computational methods capable to treat transient, non-linear fluid-structure interaction problems. Donea et al. (1982) presented an arbitrary Lagrangian-Eulerian (ALE) finite element method with automatic and continuous rezoning technique of the fluid mesh. With this method, dynamic response of nuclear reactor under solid fluid interaction has been simulated. Recently Park et al. (2014) examined the modal characteristics of Reactor vessel internals (RVIs) based on scale-similarity analysis with fluid-structure interaction (FSI). It was observed that the added-mass (A-M) model for submerged structures is considerably dependent on mode shapes and natural frequencies. Sigrist et al. (2006) conducted comparative dynamic analysis with FSI modeling for pressure vessel and internals in a nuclear reactor. They proved that the coupling effect is significant, whereas the effect of added-stiffness on global behavior is negligible. Je et al. (2017) stressed that improvement of numerical analysis methods has been required to solve complicated phenomena that occur in nuclear facilities. Particularly, fluid-structure interaction (FSI) behavior should be resolved for accurate design and evaluation of complex reactor vessel internals (RVIs) submerged in coolant. They investigated the FSI effect on dynamic characteristics of RVIs in a typical 1,000 MWe nuclear power plant. Modal analyses of an integrated assembly were conducted by employing the fluid-structure (F-S) model as well as the traditional added-mass model.

Though numerous efforts have been made on simulation of SFI problem in nuclear reactors, to the author's best knowledge, high fidelity modeling of this dynamic nonlinear phenomena with complex geometry is still unavailable. Full sets of Navier-stokes (N-S) equation is rarely solved for fully solid fluid coupling. Instead, many simplified analysis procedure is adopted: like added-mass (A-M) model (as shown in equation 111.1) Park et al. (2014); Sigrist et al. (2006) and simplified acoustic wave equation Je et al. (2017) (equation 111.2). These simplified methods introduce great modeling uncertainty to the simulation system.

$$
\begin{aligned}
&[M_s]\{\ddot{u}\} + [C_s]\{\dot{u}\} + [K_s]\{u\} = \{f_e\} + \{f_f\} \\
&\qquad f_f = \int_S \{N_p\}^T \{n\}\{P\} dS = -[M_a]\{\ddot{u}\}
\end{aligned}
\tag{111.1}
$$

$$
[M_f]\{\ddot{P}\} + [C_f]\{\dot{P}\} + [K_f]\{P\} + \rho [R_{int}]^T \{\ddot{u}\} = 0 \tag{111.2}
$$

On the other hand, ALE method, which was originally put forward as a powerful tool for fluid dynamics with deforming boundary Hirt et al. (1974), has been applied to fully solve the coupled N-S equation and solid mechanic equation Le Tallec and Mouro (2001); Murea and Sy (2017). The freedom in moving the fluid mesh offered by the ALE formulation is very attractive. However, it can be overshadowed by the burden of specifying grid velocities, well suited to a particular problem. As a consequence, the practical implementation of the ALE description requires that an automatic mesh displacement prescription algorithm to be supplied. Many methods have been put forward to overcome this difficulty. For example, pseudo-solid method was adopted by Van Loon et al. (2007) and Jasak and Tukovic (2006) came up with a simplified procedure by solving a Laplacian equation 111.3 of grid velocity with finite element discretization.

$$
\begin{aligned}
&\nabla \cdot (\gamma \nabla \boldsymbol{u}) = \boldsymbol{0} \\
&\boldsymbol{x}_{new} = \boldsymbol{x}_{old} + \boldsymbol{u}\Delta t
\end{aligned}
\tag{111.3}
$$

Seemingly, introducing these additional equations to specify the movement of fluid mesh can well resolve the inherent problem of ALE method. Based on specified grid velocity, solutions to ALE-formed N-S equations 111.4 can give precise response of fluid flow.

$$
\begin{aligned}
\frac{\partial}{\partial t}(\rho J) &= J \nabla \cdot (\boldsymbol{w} - \boldsymbol{v}) \\
\frac{\partial}{\partial t}(\rho \boldsymbol{v} J) &= J \nabla \cdot \boldsymbol{v}(\boldsymbol{w} - \boldsymbol{v}) + J(\rho \boldsymbol{b} - \nabla p)
\end{aligned}
\tag{111.4}
$$

However, mathematically the system equations (seen in section 111.3.1) to represent physical phenomena of SFI itself is sufficient and complete. Theoretically, no additional equations need to be added to the coupled system. The authors think that the ease gained here by introducing extra mesh movement equations to ALE method is sacrificed with the accuracy of the result of pressure field. Because in equation 111.4, the pressure is dependent on both absolute velocity $\boldsymbol{v}$ and relative velocity $(\boldsymbol{v} - \boldsymbol{w})$. Different configuration of mesh velocity can result in different relative velocity under the same absolute velocity, which in turn causes different pressure field.

This could be fine for pure fluid dynamics problem, where engineers care more about the fluid flow (i.e. velocity field). However, this may not be good enough for SFI, where hydrodynamic pressure at solid fluid interface is of great of importance for accurate Neumann boundary condition of solid domain. Therefore, precise pressure field is desired and indispensable in high-fidelity simulation of SFI.

The research presented here aims at realistic SFI modeling in nuclear reactors with solutions to fully coupled FSI system (i.e. N-S equations, solid mechanics equations and interface constraint equations). Geometric conformity is also achieved in this work. The great emphasis was put on accurate pressure field at solid fluid interface. A full sets of verification and validation tests are provided to guarantee the reliability of our modeling.

The limitation of current work is that relatively large displacement of solid fluid interface and accompanying Eulerian mesh distortion problem are not well resolved. Further development are needed for these topics.

111.3 Theoretical Formulation

111.3.1 Solid Fluid Interaction

The mathematical description of physical phenomenon of solid fluid interaction includes three parts Van Loon et al. (2007): The response of solid domain Ω_s is controlled by the theory of general continuum solid mechanics (equation 111.5). The governing equation in fluid domain is Navier-Stokes equation (N-S equations), which basically consists of mass conservation and momentum conservation equation (shown in equation 111.6). In the equations below, symbols $\boldsymbol{u}$, $\boldsymbol{\sigma}$, $\boldsymbol{f}$, p, G, ρ and η denote velocity, Cauchy stress tensor, body force, pressure, solid shear modulus, density and fluid viscosity. $\boldsymbol{F}$ is deformation gradient tensor defined as $\boldsymbol{F} = \nabla\chi(X,t)$.

$$\rho^s \frac{d\boldsymbol{u^s}}{dt} = \boldsymbol{\nabla} \cdot \boldsymbol{\sigma^s} + \rho^s \boldsymbol{f^s} \quad in\ \Omega_s \tag{111.5a}$$

$$det(\mathsf{F}) = 1 \quad in\ \Omega_s \tag{111.5b}$$

$$\boldsymbol{\sigma^s} = G(\boldsymbol{F} \cdot \boldsymbol{F^T} - \boldsymbol{I}) - p^s \boldsymbol{I} \quad in\ \Omega_s \tag{111.5c}$$

$$\rho^f \frac{\partial \boldsymbol{u^f}}{\partial t} + = \boldsymbol{\nabla} \cdot \boldsymbol{\sigma^f} + \rho^f \boldsymbol{f^f} \quad in\ \Omega_f \tag{111.6a}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{u^f} = 0 \quad in\ \Omega_f \tag{111.6b}$$

$$\boldsymbol{\sigma^f} = 2\eta \boldsymbol{D}(\boldsymbol{u^f}) - p^f \boldsymbol{I} \quad in\ \Omega_f \tag{111.6c}$$

At the solid fluid interface $D = \partial\Omega_s \cap \partial\Omega_f$, kinematic and dynamic constraints should be met, as shown in equation 111.7.

$$\boldsymbol{u}^s - \boldsymbol{u}^f = \boldsymbol{0} \quad in\ D \tag{111.7a}$$

$$\boldsymbol{\sigma}^s \cdot \boldsymbol{n} + \boldsymbol{\sigma}^f \cdot \boldsymbol{n} = \boldsymbol{0} \quad in\ D \tag{111.7b}$$

111.3.2 Finite Volume Discretization

For general purpose, the standard form pf the transport equation for a scalar property ϕ is considered here in 111.8. It is a second order equation as the diffusion term includes the second derivative of ϕ in space. Finite volume discretization will be applied to the integral form regarding to control volume V_p (equation 111.9) in both spatial and temporal sense Moukalled et al. (2016).

$$\underbrace{\frac{\partial\rho\phi}{\partial t}}_{\text{temporal derivative}} + \underbrace{\boldsymbol{\nabla}\cdot(\rho\boldsymbol{U}\phi)}_{\text{convection term}} - \underbrace{\boldsymbol{\nabla}\cdot(\rho\Gamma_\phi\boldsymbol{\nabla}\phi)}_{\text{diffusion term}} = \underbrace{S_\phi(\phi)}_{\text{source term}} \tag{111.8}$$

$$\underbrace{\int_{V_p}\frac{\partial\rho\phi}{\partial t}dV}_{\text{temporal derivative}} + \underbrace{\int_{V_p}\boldsymbol{\nabla}\cdot(\rho\boldsymbol{U}\phi)dV}_{\text{convection term}} - \underbrace{\int_{V_p}\boldsymbol{\nabla}\cdot(\rho\Gamma_\phi\boldsymbol{\nabla}\phi)dV}_{\text{diffusion term}} = \underbrace{\int_{V_p}S_\phi(\phi)dV}_{\text{source term}} \tag{111.9}$$

- Spatial discretization

  The spatial discretization of equation 111.9 includes three parts: discretization of convection term, diffusion term and source term, respectively. Using divergence theorem, the semi-discrete form of convection term at arbitrary control volume V_p can be given in equation 111.10, where F denotes the mass flux through the face defined in equation 111.11. In this semi-discrete form, we still need to calculate the face value ϕ_f and face mass flux F in order to evaluate the whole volume integral of convection term.

$$\int_{V_p} \boldsymbol{\nabla} \cdot (\rho \boldsymbol{U} \phi) dV = \sum_f \boldsymbol{S} \cdot (\rho \boldsymbol{U} \phi)_f \tag{111.10a}$$

$$= \sum_f \boldsymbol{S} \cdot (\rho \boldsymbol{U})_f \phi_f \tag{111.10b}$$

$$= \sum_f F \phi_f \tag{111.10c}$$

$$F = \boldsymbol{S} \cdot (\rho \boldsymbol{U})_f \tag{111.11}$$

The face value ϕ_f at face center f can be obtained from face interpolation scheme using ϕ value at control volume center ϕ_p and value at neighboring volume center ϕ_N (equation 111.12).

$$\phi_f = f_x \phi_p + (1 - f_x) \phi_N \tag{111.12}$$

Here f_x is the interpolation factor defined as the ratio of distances $\overline{fN}$ and $\overline{PN}$:

$$f_x = \frac{\overline{fN}}{\overline{PN}} \tag{111.13}$$

Similarly, for diffusion term the semi-discrete form is presented in equation 111.14. With the semi-discrete form, $(\boldsymbol{\nabla}\phi)_f$ still needs to be evaluated to achieve full discretization. For orthogonal mesh, following equation 111.15 can be used to simplify our analysis, where $|d|$ is the magnitude of vector $\overline{PN}$.

$$\int_{V_p} \boldsymbol{\nabla} \cdot (\rho \Gamma_\phi \boldsymbol{\nabla} \phi) dV = \sum_f \boldsymbol{S} \cdot (\rho \Gamma_\phi \boldsymbol{\nabla} \phi)_f \tag{111.14a}$$

$$= \sum_f (\rho \Gamma_\phi)_f \boldsymbol{S} \cdot (\boldsymbol{\nabla} \boldsymbol{\phi})_f \tag{111.14b}$$

$$\boldsymbol{S} \cdot (\nabla \phi)_f = |\boldsymbol{S}| \frac{\phi_N - \phi_P}{|\boldsymbol{d}|} \tag{111.15}$$

The source term $S_\phi(\phi)$ can be a general function of ϕ. Before the actual discretization, the source term is first linearized, where S_u and S_p also depend on ϕ. Then the volume integral of source term can be evaluated with equation 111.17.

$$S_\phi(\phi) = Su + Sp\phi \tag{111.16}$$

$$\int_{V_p} S_\phi(\phi)dV = S_u V_p + S_p V_p \phi_p \tag{111.17}$$

- Temporal discretization

Conducting time integration from t to $t + \Delta t$ with both sides of equation 111.9 yields equation 111.18.

$$\int_t^{t+\Delta t} [\frac{\partial}{\partial t}\int_{V_p} \rho\phi dV + \int_{V_p} \boldsymbol{\nabla}\cdot(\rho \boldsymbol{U}\phi)dV - \int_{V_p} \boldsymbol{\nabla}\cdot(\rho\Gamma_\phi \boldsymbol{\nabla}\phi)]dt = \int_t^{t+\Delta t} (\int_{V_p} S_\phi(\phi)dV)dt \tag{111.18}$$

Substituting the spatial semi-discretization shown above (equations 111.10, 111.14 and 111.17) into equation 111.18 and assuming that the control volumes do not change in time, equation 111.18 can be written as ;

$$\int_t^{t+\Delta t} [(\frac{\partial \rho\phi}{\partial t})_p V_p + \sum_f F\phi_f - \sum_f (\rho\Gamma_\phi)_f \boldsymbol{S}\cdot(\boldsymbol{\nabla}\phi)_f]dt = \int_t^{t+\Delta t} (S_u V_p + S_p V_p \phi_p)dt \tag{111.19}$$

Here further temporal discretization are needed to evaluate equation 111.19:

$$(\frac{\partial \rho\phi}{\partial t})_p = \frac{\rho_P^n \phi_P^n - \rho_P^0 \phi_P^0}{\Delta t} \tag{111.20a}$$

$$\int_t^{t+\Delta t} \phi(t)dt = \frac{1}{2}(\phi^0 + \phi^n)\Delta t \tag{111.20b}$$

where

$$\phi^n = \phi(t + \Delta t)$$

$$\phi^0 = \phi(t)$$

Assuming that the density and diffusivity do not change in time, the final semi-discrete form including both spatial and temporal discretization can be given in equation 111.21. Since in equation 111.21, ϕ_f^n, $(\nabla\phi)_f^n$ can be expressed with ϕ values at control cell ϕ_P and neighboring cells ϕ_N with equation 111.12 and 111.15. Therefore, equation 111.21 can be finalized into algebraic equation form (equation 111.22).

$$\begin{aligned} &\frac{\rho_P\phi_P^n - \rho_p\phi_P^0}{\Delta t}V_p + \frac{1}{2}\sum_f F\phi_f^n + \frac{1}{2}\sum_f F\phi_f^0 \\ -&\frac{1}{2}\sum_f(\rho\Gamma_\phi)_f\boldsymbol{S}\cdot(\nabla\phi)_f^n - \frac{1}{2}\sum_f(\rho\Gamma_\phi)_f\boldsymbol{S}\cdot(\nabla\phi)_f^0 \\ &= SuV_p + \frac{1}{2}S_pV_P\phi_P^n + \frac{1}{2}S_pV_P\phi_P^0 \end{aligned} \tag{111.21}$$

$$a_P\phi_P^n + \sum_N a_N\phi_N^n = R_P \tag{111.22}$$

For every control volume, one equation of this form is assembled. The value of ϕ_P^n depends on the values in the neighboring cells, thus creating a system of algebraic equations 111.23, where [A] is a sparse matrix, with coefficients a_P on the diagonal and a_N off the diagonal, $[\phi]$ is the vector of ϕ-s for all control volumes and $[R]$ is the source term vector.

$$[A][\phi] = [R] \tag{111.23}$$

111.3.3 Volume of Fluid Method

The physical phenomena of earthquake soil structure interaction in fluid domain is essentially free surface flow. There are three types of problems in the numerical treatment of free boundaries: (1) their discrete representation, (2) their evolution in time, and (3) the manner in which boundary conditions are imposed on them.

Volume of Fluid method is originally put forward by Hirt and Nichols (1981) to solve free surface flow problems (free boundary problems). Different from traditional one-phase method, VOF method considers more general two-phase flow problem. In special case where we take one phase as air and another phase as any liquid that needs to be simulated, then VOF becomes also applicable for free surface flow.

The idea of Volume of Fluid method is to introduce a new field variable - volume fraction value (α) for each control volume, which is defined in equation ?? where V_i is the total volume of cell i and V_i^w is the volume of water contained in the cell. From the definition, it can be seen that α values range between 0 and 1. If the cell is completely filled with fluid then $\alpha = 1$ and if it is filled with void phase then its value should be 0. In VOF method, momentum equation and continuity equation are solved for one composite fluid phase characterized by volume fraction α. The physical properties of this one fluid are calculated as weighted averages based on the volume fractions of two phases in one cell. E.g., the density of any point in the domain is calculated with equation 111.24.

$$\rho = \alpha\rho_w + (1-\alpha)\rho_a \tag{111.24}$$

The evolution of volume fraction α is controlled by an additional convection transport equation 111.26. By solving this equation, the distribution of volume fraction α can be obtained. Then the free fluid interface can be automatically identified as the zone where the volume fraction α ranges between 0 to 1. In this way, the fluid flow can be represented as the redistribution of α along with the time.

$$\alpha_i = \frac{V_i^w}{V_i} \tag{111.25}$$

$$\frac{\partial\alpha}{\partial t} + \boldsymbol{\nabla}\cdot(\alpha\boldsymbol{U}) = 0 \tag{111.26}$$

The free surface flow solver used here is interFoam implemented based on OpenFOAM. In interFoam, the necessary compression of the phase interface is achieved by introducing an extra artificial compression term into the transport convection equation of α, as shown in equation 111.27. U_r is the velocity field suitable to compress the interface. This artificial term is active only in the interface region due to the term $\alpha(1-\alpha)$.

$$\frac{\partial\alpha}{\partial t} + \boldsymbol{\nabla}\cdot(\alpha\boldsymbol{U}) + \boldsymbol{\nabla}\cdot(\alpha(1-\alpha)U_r) = 0 \tag{111.27}$$

111.3.4 Pressure-velocity coupling: PISO algorithm

For incompressible form of the fluid system (equation 111.28), two issues require special attention: non-linearity of the momentum equation and the pressure-velocity coupling Jasak (1996). The non-linear term in equation 111.29 is $\boldsymbol{\nabla}\cdot(\boldsymbol{U}\boldsymbol{U})$, *i.e.* velocity is "being transported by itself". The discretized form of this expression would be quadratic in velocity and the resulting system of algebraic equations would therefore be non-linear. There are two possible solutions for this problem - either use a solver for non-linear systems, or linearize the convection term.

$$\nabla \cdot \boldsymbol{U} = 0 \tag{111.28}$$

$$\frac{\partial \boldsymbol{U}}{\partial t} + \nabla \cdot (\boldsymbol{U}\boldsymbol{U}) - \nabla \cdot (\nu \nabla \boldsymbol{U}) = -\nabla p \tag{111.29}$$

According to section 111.3.2, convection term can be discretized with equation 111.30, where F, a_P and a_N are a function of $\boldsymbol{U}$. The important issue is that the fluxes F should satisfy the continuity equation 111.28. Linearization of the convection term implies than an existing velocity (flux) field that satisfies equation 111.28 will be used to calculate a_P and a_N.

$$\begin{aligned}\nabla \cdot (\boldsymbol{U}\boldsymbol{U}) &= \sum_f \boldsymbol{S} \cdot (\boldsymbol{U})_f (\boldsymbol{U})_f \\ &= \sum_f F(\boldsymbol{U})_f \\ &= a_P \boldsymbol{U}_P + \sum_N a_N \boldsymbol{U}_N\end{aligned} \tag{111.30}$$

PISO (Pressure-Implicit Splitting of Operators) procedure proposed by Venier et al. (2017) is used here for pressure-velocity coupling in transient calculations. In order to derive the pressure equation, a semi-discrete form (equation 111.31) of momentum equation is adopted, where pressure gradient term is not discretized at this stage. The $\boldsymbol{H}(\boldsymbol{U})$ term consists of two parts: the "transport part", including the matrix coefficients for all neighbors multiplied by corresponding velocities and the "source part" including the source part of the transient term.

$$\begin{aligned} a_P \boldsymbol{U}_P &= \boldsymbol{H}(\boldsymbol{U}) - \nabla p \\ \boldsymbol{H}(\boldsymbol{U}) &= -\sum_N a_N \boldsymbol{U}_N + \frac{\boldsymbol{U}^0}{\Delta t}\end{aligned} \tag{111.31}$$

In addition, the discrete form of continuity equation is:

$$\nabla \cdot \boldsymbol{U} = \sum_f \boldsymbol{S} \cdot \boldsymbol{U}_f = 0 \tag{111.32}$$

From equation 111.31, $\boldsymbol{U}$ can be explicitly expressed with equation 111.33. Then from the explicit solution $\boldsymbol{U}_P$ velocities at cell face $\boldsymbol{U}_f$ can be calculated through face interpolation (equation 111.34). Substituting equation 111.34 into equation 111.32 yields the equivalent form of continuity condition (equation 111.35).

$$\boldsymbol{U}_P = \frac{\boldsymbol{H}(\boldsymbol{U})}{a_P} - \frac{1}{a_P}\boldsymbol{\nabla} p \tag{111.33}$$

$$\boldsymbol{U}_f = (\frac{\boldsymbol{H}(\boldsymbol{U})}{a_P})_f - (\frac{1}{a_P})_f(\boldsymbol{\nabla} p)_f \tag{111.34}$$

$$\sum_f \boldsymbol{S} \cdot [(\frac{1}{a_P})_f(\boldsymbol{\nabla} p)_f] = \sum_f \boldsymbol{S} \cdot (\frac{\boldsymbol{H}(\boldsymbol{U})}{a_P})_f \tag{111.35}$$

The fully discrete form of momentum equation is given in equation 111.36. Equation 111.35 and equation 111.36 constitute the discrete form of incompressible Navier-Stokes system. The face flux F can be calculated using equation 111.37. When equation 111.35 is satisfied, the face fluxes are guaranteed to be conservative.

$$a_P \boldsymbol{U}_P = \boldsymbol{H}(\boldsymbol{U}) - \sum_f \boldsymbol{S}(p)_f \tag{111.36}$$

$$F = \boldsymbol{S} \cdot \boldsymbol{U}_f = \boldsymbol{S} \cdot [(\frac{\boldsymbol{H}(\boldsymbol{U})}{a_P})_f - (\frac{1}{a_P})_f(\boldsymbol{\nabla} p)_f] \tag{111.37}$$

With all the discrete form of system equations prepared, the PISO algorithm can be described as follows:

- The momentum equation is solved first. The exact pressure gradient source term is not known at this stage – the pressure field from the previous time-step is used instead. This stage is called the momentum predictor. The solution of the momentum equation, Eqn. 111.36, gives an approximation of the new velocity field.

- Using the predicted velocities, the $\boldsymbol{H}(\boldsymbol{U})$ operator can be assembled and the pressure equation can be formulated. The solution of the pressure equation 111.35 gives the first estimate of the new pressure field. This step is called the pressure correction.

- Eqn. 111.37 provides a set of conservative fluxes consistent with the new pressure field. The velocity field should also be corrected as a consequence of the new pressure distribution. Velocity correction is done in an explicit manner, using Eqn. 111.36. This is the explicit velocity correction stage.

- Eqn. 111.36 reveals that velocity correction consists of two parts: a correction due to the change in the pressure gradient and the transported influence of corrections of neighboring velocities. Explicit velocity correction means that the latter part is neglected. The whole velocity error is assumed to come from the error in pressure term. This is not true. It is therefore necessary to correct the $\boldsymbol{H}(\boldsymbol{U})$ term and repeat pressure correction and explicit velocity correction stage. In other words, the PISO loop consists of an implicit momentum predictor followed by a series of pressure solutions and explicit velocity corrections. The loop is repeated until a pre-determined tolerance is reached.

111.3.5 Explicit transient algorithm

The algorithm for solid fluid coupling is an explicit segregated approach. The solving of system equation 111.5 in solid domain and system equation 111.6 in fluid domain are performed by RealESSI and InterFoam respectively. A new container called SSFI is implemented in RealESSI to control all the boundary information of solid fluid interface. The algorithm is illustrated in figure 111.1.

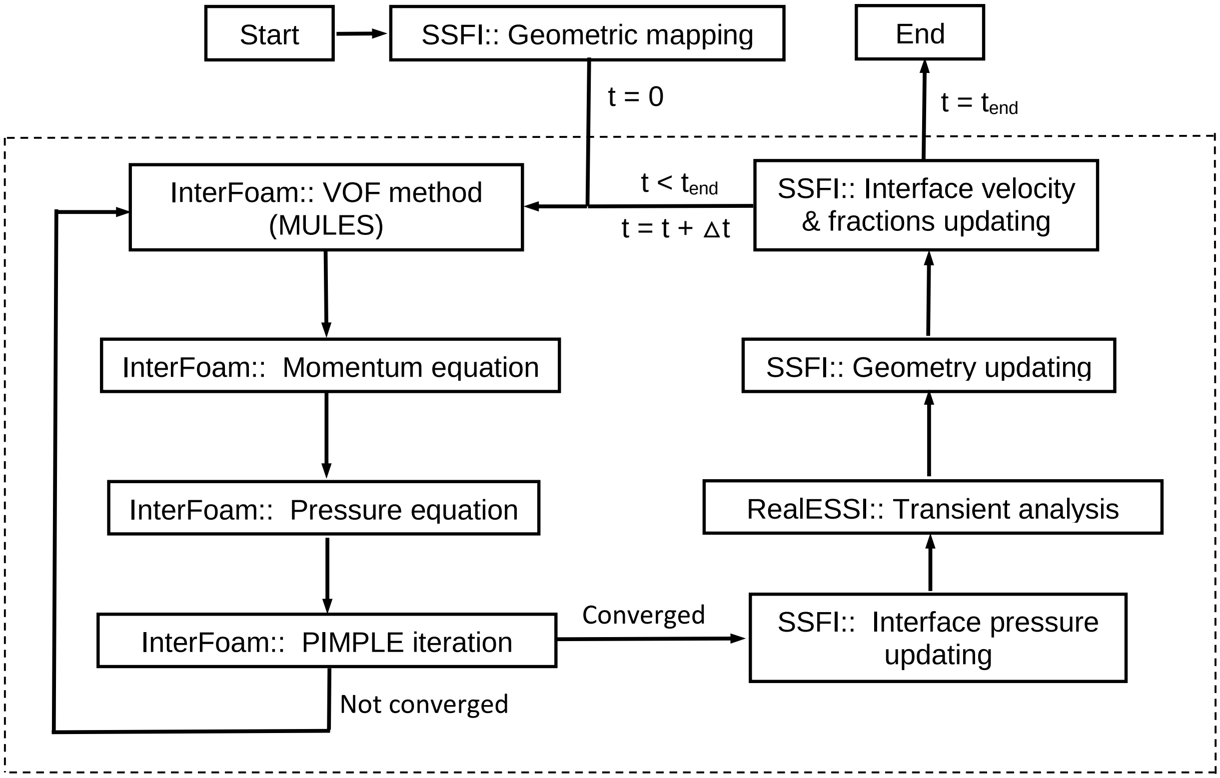


Figure 111.1: Flowchart of explicit transient algorithm

Initially all the geometric mapping information of solid fluid interaction is built by SSFI. There are 4 types of geometric mapping in SSFI: Foam node (interface node in fluid domain) maps to foam surface (interface surfaces in fluid domain), ESSI node (interface node in solid domain) maps to foam nodes, foam surface maps to ESSI nodes and foam node maps to ESSI nodes. The specific definitions of these geometric mappings and implementation details can be found in section 111.4.4. These geometric mapping information is indispensable part while conducting the interpolation for interface pressure, velocity and nodal displacement.

Then the equilibrium state of fluid domain is solved first by Interfoam based on the boundary conditions from the response of solid domain at last time step. Here the PIMPLE algorithm Chen et al. (2014) is implemented in InterFoam to couple pressure and velocity, which is a hybrid of the PISO and SIMPLE (Semi-Implicit Method for Pressure-Link Equations) algorithms. In the PIMPLE loop, the transport equation of volume fraction (equation 111.26) is firstly calculated based on existing velocities and surface fluxes. Following that, there is an implicit momentum predictor and several pressure-velocity correctors. After the fluid domain achieves equilibrium for the new time step, the pressures at fluid interface are transformed to equivalent nodal force and applied at the solid interface. This operation is called interface pressure updating.

With updated nodal force at solid interface and some other transient boundary conditions, the response of solid domain for the new step is obtained through transient analysis of RealESSI (Jeremić et al. (1988-2025)). Based on the latest location of solid interface, the geometry of fluid interface is updated correspondingly to make sure the geometric conformity of both domains.

No slip boundary condition is adopted here for the velocity at fluid interface. Therefore, the new boundary velocities are also calculated from the response of solid domain. In addition, it is crucially important to update volume fractions so that mass conservation is guaranteed. The detailed information about interpolation and updating of these physical fields (i.e. pressure, velocity and volume fractions) is presented in section 111.4.6 and 111.4.5.

For the above explicit transient algorithm, both solid domain and fluid domain can individually reach equilibrium states through iterations at both sides with updated interface boundary conditions. However, this two equilibrium states are not achieved at the same time. Current equilibrium state of solid domain matches with the equilibrium sate of fluid domain at the last time step. Therefore, the algorithm is only valid and accurate when time step is small enough. Different time step lengths in solid domain and fluid domain can also be handled through Shepherd method. SSFI can go through transient analysis in solid domain and fluid domain alternatively according to different time step length.

111.4 Implementation Details

111.4.1 Installation of OpenFoam

The official website of Openfoam: www.openfoam.com/down load/install-source.php gives detailed instruction about how to build Openfoam from source on different operating systems. Installation of customized version of OpenFOAM, that can interact, connect to Real-ESSI is described in some detail in chapter 209.6 on page 1335.

Note that after installation of Openfoam, you need to source the OpenFOAM environment by executing, e.g. for bash shells and OpenFoam version v1706:

```
source  /OpenFOAM/OpenFOAM-v1706/etc/bashrc
```

111.4.2 Integrated Preprocessor-gmFoam

gmFoam is designed as an integrated preprocessor developed for analysis of solid fluid interaction. It enables user to build an integrated geometric model (including both solid part and fluid part) in Gmsh and generate input files for both RealESSI and interFoam in a very easy way. Some simple examples can be seen in figure 111.2. Its main functions are listed below:

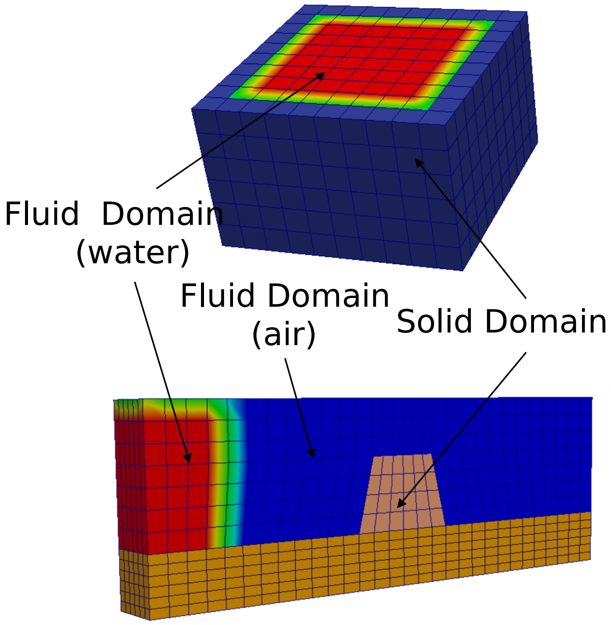


Figure 111.2: Numerical models built with gmFoam

- Mesh separation

Note that FVM mesh supported by Openfoam is very flexible. It can be any types of convex polyhedron as shown in figure 111.3). And the geometric description of FVM mesh required by Openfoam is totally different from that of FEM mesh. The description of FVM mesh is face-based. Faces (including both boundary surfaces and exterior surfaces) of control volumes are defined by a list a point IDs that consist of the face. Also the owner cell ID and neighbor cell ID of each face have to be specified. In contrast, the description of FEM mesh is element-based. After defining all the nodes in the model, the element is described by a list of node IDs. gmFoam supports both types of mesh description. User can build an integrated geometric model in Gmsh and define solid part and fluid part as different physical volume groups. After meshing it, gmFoam can separate the mesh information about solid domain and fluid domain and transfer them to FEM mesh and FVM mesh, respectively.

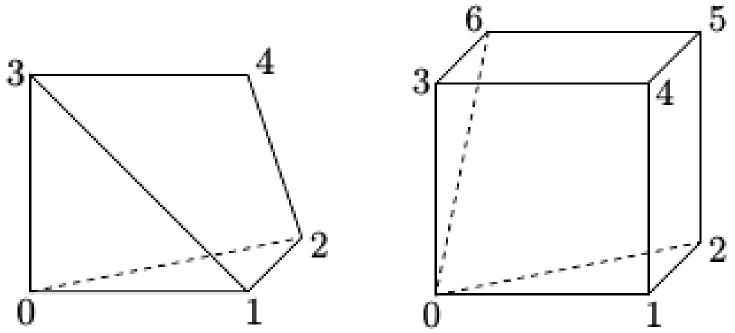


Figure 111.3: Mesh for FVM

- GmESSI incorporation

gmFoam perfectly incorporates current RealESSI preprocessor GmESSI so that it can quickly generate ESSI input files for the simulation of solid part. Also it has the capability to quickly generate input files for Openfoam. Currently the input file organization of Openfoam is very complicated. Several folders and files are needed to prepare in order to complete a very simple simulation. But with gmFoam all these basic information can be written in one single file with suffix as .gmfoam and gmFoam will automatically parse the content inside and produce all the input folders and files. In addition, with the help of physical group, gmFoam enables user to set different boundary conditions in a very convenient way. This is extremely helpful when we conduct solid fluid interaction analysis for big models with complicated boundary conditions.

- Interface geometry extraction

  gmFoam can extract the geometric information of solid-fluid interface and write down corresponding information as input files. The information is used to initialize SSFI object and build important geometric interface mapping.

- Support discontinuous mesh

  Discontinuous mesh is supported. This is very important, especially for large-scale simulation for solid fluid interaction. Usually refined mesh is needed for fluid part to get accurate enough result using VOF. Discontinuous mesh enables us to arbitrarily refine fluid mesh without changing solid mesh. Therefore, increase of computational efforts in solid part can be avoided. Figure 111.4 shows a model with discontinuous mesh (fluid mesh size : solid mesh size=1:3).

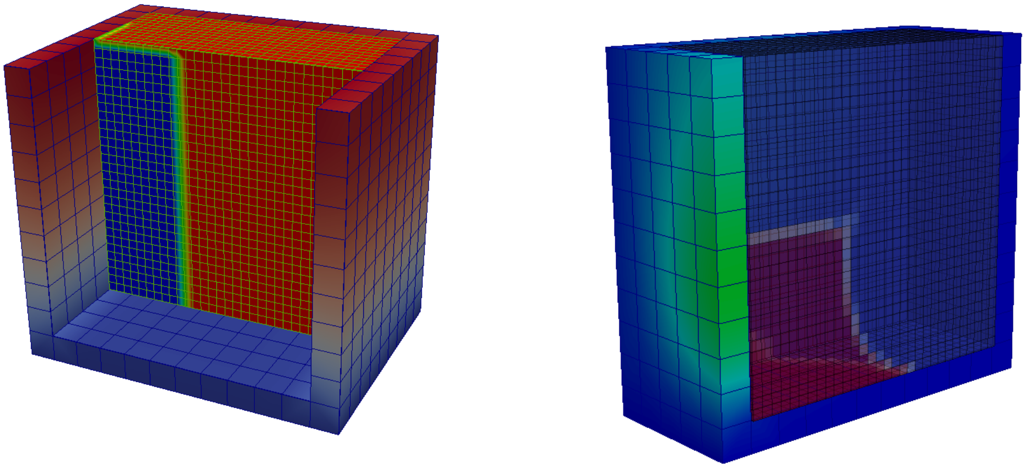


Figure 111.4: Numerical model with discontinuous mesh

111.4.3 Interface Domain-SSFI

An interface class SSFI was implemented in RealESSI to couple computations between solid domain and fluid domain. SSFI behaves like a container (called interface domain) to control geometric mapping, sequence of computation and boundary data interpolation and transmission. SSFI class contains and operates the objects of four base classes: ESSINode, ESSISurface, FoamNode, FoamSurface. Two core member functions: SSFI::FoamToESSIUpdate(double t) and SSFI::ESSIToFoamUpdate(Domain* theDomain, double t) are designed to perform all necessary updates on solid domain and fluid domain, respectively.

111.4.4 Geometric Mapping

Following geometrical mappings are built and maintained in SSFI.

- Foam node mapping to Foam surfaces

  For each foam node in solid fluid interface, this mapping returns all the IDs of its surrounding foam surfaces. For example, as shown in figure 111.5(a), foam node 5 is mapped to foam surface 1,2,3 and 4.

Foam node →	Foam Surfaces
5	(1,2,3,4)

- ESSI node mapping to Foam nodes

  For each ESSI node in solid fluid interface, this mapping returns all the IDs of its surrounding foam nodes within certain search radius (by default the radius is set as 0.1 meters). If the mesh size is very refined, reducing the searching radius helps to improve accuracy.

- Foam face mapping to ESSI nodes

  For each foam face in solid fluid interface, this mapping returns 4 vertex node IDs of an ESSI surface that contains the center of this foam face. Like in figure 111.5(b), foam face 1 (consists of foam node 1,2,3 and 4) is mapped to ESSI nodes 1,2,3 and 4.

Foam face →	ESSI nodes
1	(1,2,3,4)

- Foam node mapping to ESSI nodes

  For each foam node in solid fluid interface, this mapping returns 4 vertex node IDs of an ESSI surface that contains the Foam node. In figure 111.5(b), foam node 1 is mapped to ESSI nodes 1,3,8 and 7.

Foam node →	ESSI nodes
1	(1,3,8,7)

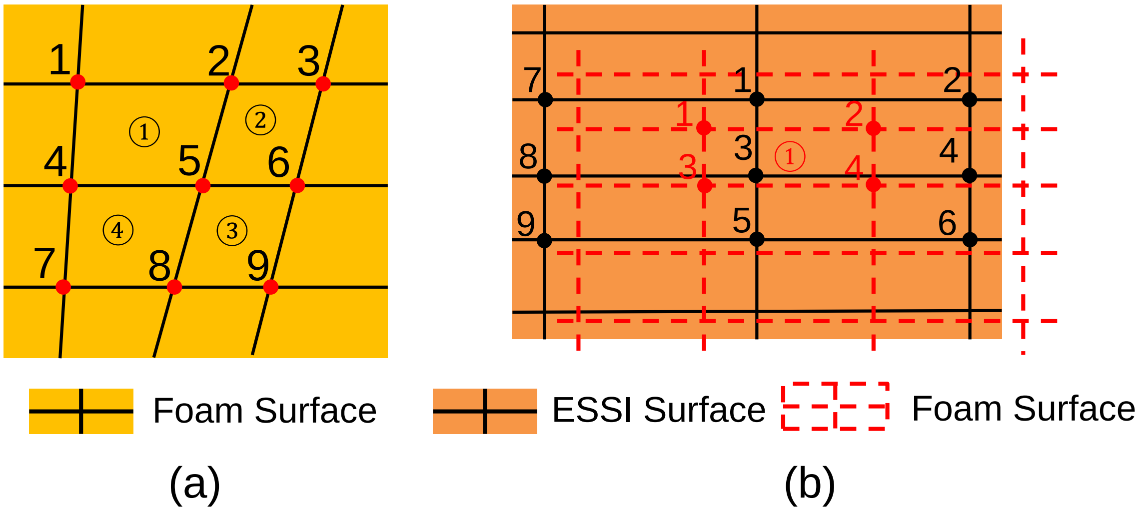


Figure 111.5: Geometric mapping in SSFI

111.4.5 SFI Interpolation

After building all of these geometric mapping, interpolation scheme is also needed to fully determine the values of interface variables (pressure, velocity and displacement) and update these values during the interaction process. There are three types of interpolation and updating involved here:

- Pressure interpolation

  The interpolation and updating of interface pressure happen during the process of FOAM to ESSI updating. After the fluid domain achieves its equilibrium, the new pressure values at interface foam faces need to be interpolated and transfered to corresponding ESSI nodes. The pressure interpolation scheme is following:

  Firstly, using the mapping from foam node to foam surfaces, pressure at each interface foam node is calculated by taking the average pressure values of its surrounding foam surfaces. Like in figure 111.5(a), the pressure at interface foam node 5 is the average value of pressure at interface foam surface 1,2,3 and 4. Then according to the mapping from ESSI node to foam nodes, the updated pressure at interface ESSI node is calculated as the average pressure value of corresponding foam nodes.

- Velocity interpolation

  Velocity interpolation takes place during the process of ESSI to FOAM updating. After RealESSI conducts transient analysis for solid domain, the velocities at interface ESSI nodes need to be fed back to corresponding interface foam surface. The velocity interpolation scheme is following:

The mapping from Foam face to ESSI nodes can give four vertex interface ESSI node ID. For example, in figure 111.5(b), foam face 1 is mapped to four ESSI nodes (1,2,3,4). With coordinates of foam face center and four ESSI nodes already known, inverse isoparametric mapping Hua (1990) is used here to determine the local coordinates (ζ, η) of foam face center. The formula of inverse isoparametric mapping is shown below:

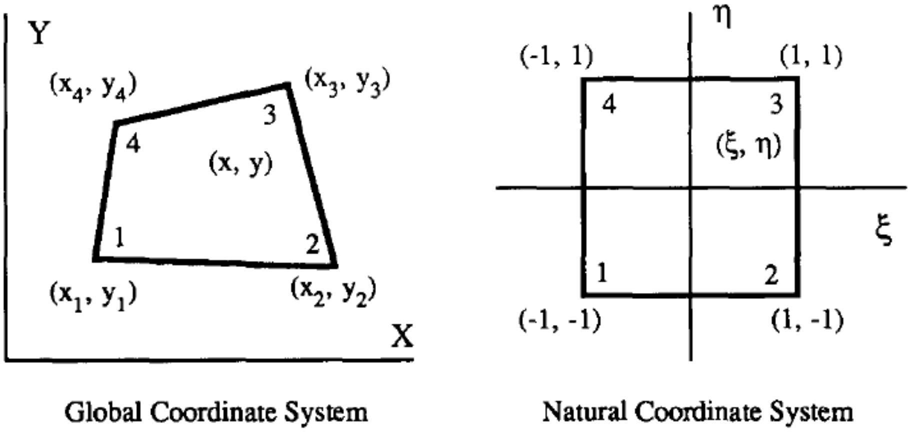


Figure 111.6: Illustration of isoparametric mapping Hua (1990)

$$\begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \\ c_1 & c_2 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix} \tag{111.38}$$

$$\begin{aligned} d_1 &= 4x - (x_1 + x_2 + x_3 + x_4) \\ d_2 &= 4y - (y_1 + y_2 + y_3 + y_4) \end{aligned} \tag{111.39}$$

A compact notation to represent the determinant of a 2 matrix is introduced as

$$r_s = \begin{vmatrix} r_1 & s_1 \\ r_2 & s_2 \end{vmatrix} = r_1 s_2 - r_2 s_1 \tag{111.40}$$

where $r,s = a,b,c,d$. Notice that $r_s = -s_r$.

With all the notations well defined, the solutions to local coordinate given by Hua (1990) are shown below:

- $a_1 a_2 a_b a_c \neq 0$

$$\begin{cases} a_b\zeta^2 + (c_b + d_a)\zeta + d_c = 0 \\ \eta = (a_d + b_a\zeta)/a_c \end{cases} \quad \text{where } \zeta \in [-1,1]$$

- $a_1 = 0$ *and* $a_2 c_1 \neq 0$

$$\begin{cases} a_b\zeta^2 + (c_b + d_a)\zeta + d_c = 0 \\ \eta = (a_d + b_a\zeta)/a_c \end{cases} \quad \text{where } \zeta \in [-1,1]$$

- $a_2 = 0$ *and* $a_1 b_2 \neq 0$

$$\begin{cases} a_b\zeta^2 + (c_b + d_a)\zeta + d_c = 0 \\ \eta = (a_d + b_a\zeta)/a_c \end{cases} \quad \text{where } \zeta \in [-1,1]$$

- $a_1 a_2 \neq 0$ *and* $a_b = 0$

$$\begin{cases} \zeta = (a_1 d_c)/(b_1 a_c + a_1 a_d) \\ \eta = a_d/a_c \end{cases}$$

- $a_1 a_2 \neq 0$ *and* $a_c = 0$

$$\begin{cases} \zeta = a_d/a_b \\ \eta = (a_1 d_b)/(c_1 a_b + a_1 a_d) \end{cases}$$

- *All other conditions*

$$\begin{cases} \zeta = d_c/(a_1 d_2 + b_c) \\ \eta = b_d/(a_2 d_1 + b_c) \end{cases}$$

After obtaining local coordinates (ζ,η) corresponding to foam surface center, isoparametric mapping can be conducted through the shape function of 4-node quadrilateral element. The interpolated velocity can be calculated with equation 111.41.

$$
\begin{aligned}
N_1 &= \frac{1}{4}(1-\zeta)(1-\eta) \\
N_2 &= \frac{1}{4}(1+\zeta)(1-\eta) \\
N_3 &= \frac{1}{4}(1+\zeta)(1+\eta) \\
N_4 &= \frac{1}{4}(1-\zeta)(1+\eta) \\
\boldsymbol{v} &= N_1\boldsymbol{v_1} + N_2\boldsymbol{v_2} + N_3\boldsymbol{v_3} + N_4\boldsymbol{v_4}
\end{aligned}
\tag{111.41}
$$

- Displacement interpolation

  Displacement interpolation also takes place during ESSI to FOAM updating. In order to meet the geometric conformity, the Eulerian mesh of fluid domain should dynamically move along with the real-time response of solid domain. Therefore, it is necessary to interpolate the displacement of interface ESSI nodes to interface foam nodes. The displacement interpolation scheme is similar to velocity interpolation scheme and shown below:

  The mapping from Foam node to ESSI nodes can give four vertex interface ESSI node ID. Inverse isoparametric mapping is first performed to compute local coordinate of Foam node. Then displacement of foam node is interpolated from displacement of four vertex ESSI nodes through isoparametric mapping.

111.4.6 Mass Conservation

There are two levels of mass conservation conditions needed to be satisfied during the SFI.

One is local level: Regarding each control volume, the amount of fluid flows in should equal to the amount of fluid that flows out for incompressible fluid. This local mass conservation condition is mathematically represented by continuity equation in equation 111.6 and can be approximately met through finite volume discretization. Another level of mass conservation is global level: For a closed fluid domain (no fluid transfer with other external fluid system), the total amount of fluid should keep constant during SFI. The global level of mass conservation is trivial for pure fluid flow system with static Eulerian boundary mesh. Since for this kind of flow system, global level of mass conservation is free and automatically holds based on local level of mass conservation.

However, for flow system with deforming boundary, especially when Lagrangian movement of Eulerian mesh is involved (like Arbitrary Lagrangian method Souli and Zolesio (2001)), the local level of mass conservation does not guarantee the mass conservation of global level. Demirdžić and Perić (1988) pointed out that for moving mesh, one more conservation equation so called space conservation needed to be solved simultaneously with the mass, momentum and energy conservation equations. Otherwise artificial mass addition or reduction is generated which may cause the solution to be greatly in error.

The space conservation law is expressed by equation 111.42, where J is the determinant of the metric tensor and $\boldsymbol{v_g}$ is the grid velocity of the mesh. Correspondingly, the finite volume discrete evaluation of volume integral of equation 111.42 can be given in equation 111.43, where $\delta V = V^n - V^0$ is the change of cell volume during Δt, $\boldsymbol{v_{g(f)}}$ is the mesh velocity of cell face and $\boldsymbol{S}$ is cell face vector.

$$\frac{\partial J}{\partial t} - J\boldsymbol{\nabla} \cdot \boldsymbol{v_g} = 0 \tag{111.42}$$

$$\frac{V^n - V^0}{\Delta t} = \sum_f \boldsymbol{v_{g(f)}} \cdot \boldsymbol{S} \tag{111.43}$$

In our implementation, consider the coupling with VOF method and our SFI simulation is generally for small deformation of fluid boundary, a simplified procedure is adopted here to guarantee the global level of mass conservation: Only the location of foam nodes at solid fluid interface is updated with Lagrangian motion of corresponding ESSI nodes while interior foam nodes remain static. After geometry updating of the mesh, the volume of foam cells at interface is re-evaluated and volume fraction values are also updated according to equation 111.44 to ensure the mass conservation. Then the transport equation 111.27 is solved based on new α values.

$$\alpha = \frac{\int \alpha^0 dV^0}{\int dV} = \frac{\alpha^0 V^0}{V} \tag{111.44}$$

Part 200

Software and Hardware Platform: Design, Development, Procurement and Use

Chapter 201

The Real ESSI Simulator System

(1986-1989-1993-1994-1996-1999-2003-2007-2019-2020-2021-)

201.1 Chapter Summary and Highlights

201.2 Introduction to the Real-ESSI Simulator System

The Real-ESSI Simulator (Realistic Modeling and Simulation of Earthquakes, and/or Soils, and/or Structures and their Interaction) is a software, hardware and documentation system for high performance, sequential or parallel, time domain, linear or nonlinear, elastic and inelastic, deterministic or probabilistic, finite element modeling and simulation of

- statics and dynamics of soil,
- statics and dynamics of rock,
- statics and dynamics of structures,
- statics of and dynamics of soil-structure systems,
- dynamics of earthquakes, and
- dynamic earthquake-soil-structure interaction.

The Real-ESSI Simulator systems is used for design and for assessment of static and dynamic behavior of infrastructure objects, including buildings, bridges, dams, nuclear installations, tunnels, etc.

Design: Multiple linear elastic load cases can be combined and design quantities, sectional forces exported for design.

Assessment: Practical, realistic, inelastic, nonlinear load staged analysis, with accurate modeling of elastic and inelastic, nonlinear components, and with all the simulation, algorithmic features available, as listed below, is performed to assess design, current safety margins and economy of objects.

The Real-ESSI Simulator is developed at the University of California, Davis, in collaboration and with partial financial support from the USDOE, USNRC, USNSF, USBR, USFEMA, CalTrans, CNSC-CCSN, UN-IAEA, Shimizu, Private Donors, etc. The Real-ESSI Simulator develops methods and models that inform and predict rather than (force) fit.

The Real-ESSI Simulator systems consists of the Real-ESSI Program, Real-ESSI Pre-Processing and Post-Processing tools, Real-ESSI Computer and Real-ESSI Notes.

201.2.1 Real-ESSI Program

The Real-ESSI program is a general purposes finite element program that features models and methods for analyzing static and dynamic behavior of Civil Engineering objects, such as buildings, bridges, dams, nuclear installations, tunnels, etc.

201.2.2 Real-ESSI Pre-Processing tools

The Real-ESSI Pre-Processing tools are a set of programs, scripts and modules that are used to develop Real-ESSI models. Mesh generation relies on Gmsh (Geuzaine and Remacle, 2009) and our own plugins Gm-ESSI, while there are also mesh translators from other input formats to Real-ESSI input format/language.

201.2.3 Real-ESSI Post-Processing tools

The Real-ESSI Post-Processing tools rely on Paraview (Ayachit, 2015) visualization platform, with our own plugins, as well as on a number of programs and scripts to visualize output using matlab, python, etc.

201.2.4 Real-ESSI Computer

The Real-ESSI Computer can be any single CPU, multiple CPU, and/or cluster of single/multiple CPUs computers using Linux operating system. The reason for using Linux is that state of the art development tools are available, and that Linux is used on virtually all large supercompters, so that the very same sources can be compiled and executables developed for small desktop computers, large server computers, local clusters of computers and large supercomputers. There is a possibility to build Real-ESSI and create executables on Macintosh and Windows platforms, as long as standard software tools and compiles, C++ and Fortran, are available. However, such work is not currently pursued.

201.2.5 Real-ESSI Notes

The Real-ESSI notes, documentation for the Real-ESSI Simulator System, is available through Lecture Notes by Jeremić et al. (1989-2025).

201.2.6 Real-ESSI Name

The Real-ESSI Simulator System name is based on an acronym: Realistic Modeling and Simulation of Earthquakes, and/or Soils, and/or Structures and their Interaction. Pronunciation of Real-ESSI is similar to ”real easy”, as in ”as easy as pie”. Translation of name Real ESSI to languages of developers and users is:

- Врло просто,
- Стварно просто,
- Стварно лако,
- Просто к’о пасуљ,
- Vrlo prosto,
- Stvarno prosto,
- 真简单 ,
- Muy fácil,
- অতি সহজ,
- **آسان واقعی**,
- Molto facile,
- 本当に簡単 ,
- Πραγματικά εύκολο,
- बहुत ही आसान,
- Très facile,
- Вистински лесно,
- Wirklich einfach,
- سهلٌ جداً
- Zelo enostavno
- Zares enostavno
- Muito Fácil
- Res lahko
- Dziecinna igraszka
- Çocuk oyuncaği

Chapter 202

Object Oriented Software Platform Design

(1992-1993-1994-1996-1999-2003-2005-2007-2008-2009-2010-2011-2015-2016-2019-)

(In collaboration with Dr. Guanzhou Jie)

202.1 Chapter Summary and Highlights

202.2 Object-Oriented Design Basics

Booch (1994); Gamma et al. (1995); Coplien (1992); Koenig (1989 - 1993); Stroustrup (1986); Stroustrup (1994); Ellis and Stroustrup (1990); Johnson (1994); Felippa (1992a); Dubois-Pelerin (1992); Dubois-Pèlerein and Zimmermann (1993); Raphael and Krishnamoorthy (1993); Menéntrey and Zimmermann (1993); Zimmermann et al. (1992a); Donescu and Laursen (1996); Zimmermann and Eyheramendy (1996); Eyheramendy and Zimmermann (1996); Jeremić and Sture (1998); Forde et al. (1990); Miller (1991); Scholz (1992); Fenves (1990); Eyheramendy (1997); Dubois-Pélerin and Zimmermann (1992); Zimmermann et al. (1992b); Dubois-Pélerin and Pegon (1998); Eyheramendy and Zimmermann (2001);

Veldhuizen (1995a); Veldhuizen (1995b); Veldhuizen (1996); Veldhuizen and Jernigan (1997); Archer (1996); Archer et al. (1999); McKenna (1997);

202.3 Object-Oriented Design of the Plastic Domain Decomposition (PDD)

202.3.1 Introduction

This section describes the object oriented design of the proposed PDD algorithm and its implementation into MOSS library framework. At the beginning of this section, the Object-Oriented approach to programming the Finite Element Method is reviewed based on the existing (as of 2005) implementation of OpenSees. Object-Oriented parallel design is then extended from the existing framework. Parallel algorithm adopts Main-Follower paradigm and the new design of data structures have strictly followed the Object-Oriented principle using C++ language. External utility libraries such as ParMETIS and PETSc have been incorporated to provide seamless parallel numerical manipulations including partitioning/repartitioning and equation solving.

In this chapter, the algorithm overview will be presented first. Then the implementation details in C++ will follow. The challenges of achieving load balancing in parallel Finite Element simulation have been divided into two parts, global level equation solving and constitutive level iterations. This research presents the PDD algorithm to demonstrate how to balance each stage systematically in applications.

202.3.2 Object-Oriented Parallel Finite Element Algorithm

Parts of OpenSees software framework (McKenna, 1997) have been used in this chapter. Object-Oriented design of OpenSees enables software reuse that greatly shortens the development life cycle of application codes.

OpenSees is comprised of a set of classes and objects that represent models perform computations for solving the governing equations, and provide access to processing results. There are four types of class objects in OpenSees McKenna (1997).

- Modeling Classes are used to create the Finite Element Model Classes for a given problem.
- Finite Element Model Classes are used to describe the finite element model and to store the results of the analysis performed on the model. Main class abstractions used in OpenSees are Node, Element, Constraint, Load and Domain. The relationship amongst these classes can be shown using the class diagram Figure 202.2 using the Rumbaugh notation as shown in Figure 202.1 Rumbaugh et al. (1991).
- Analysis Classes are used to perform the finite element analysis, i.e., to form and solve the global system of equations
- Numerical Classes are used to handle numerical operations in the solution procedure. Also included in this category are data structure classes such as Vector, Matrix and Tensor.

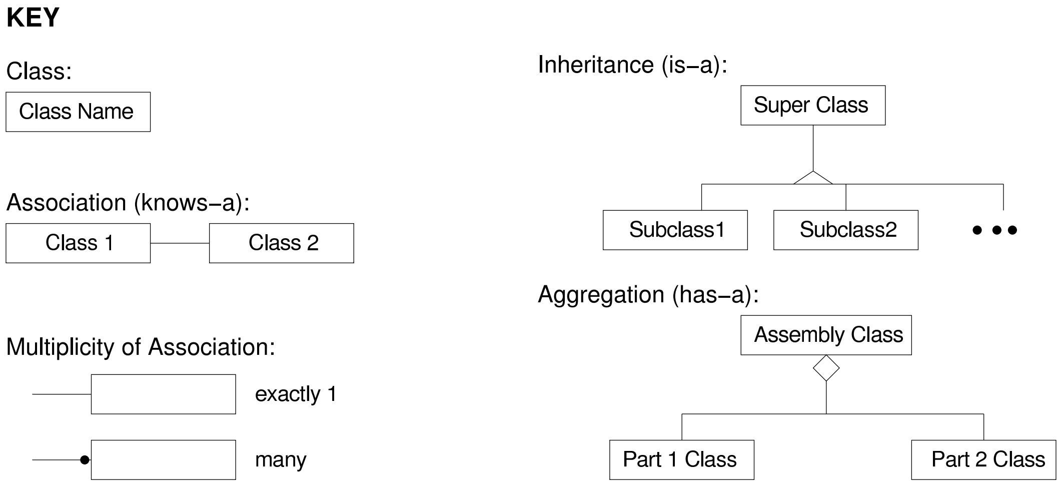


Figure 202.1: Rumbaugh Notation of-Object Oriented Design

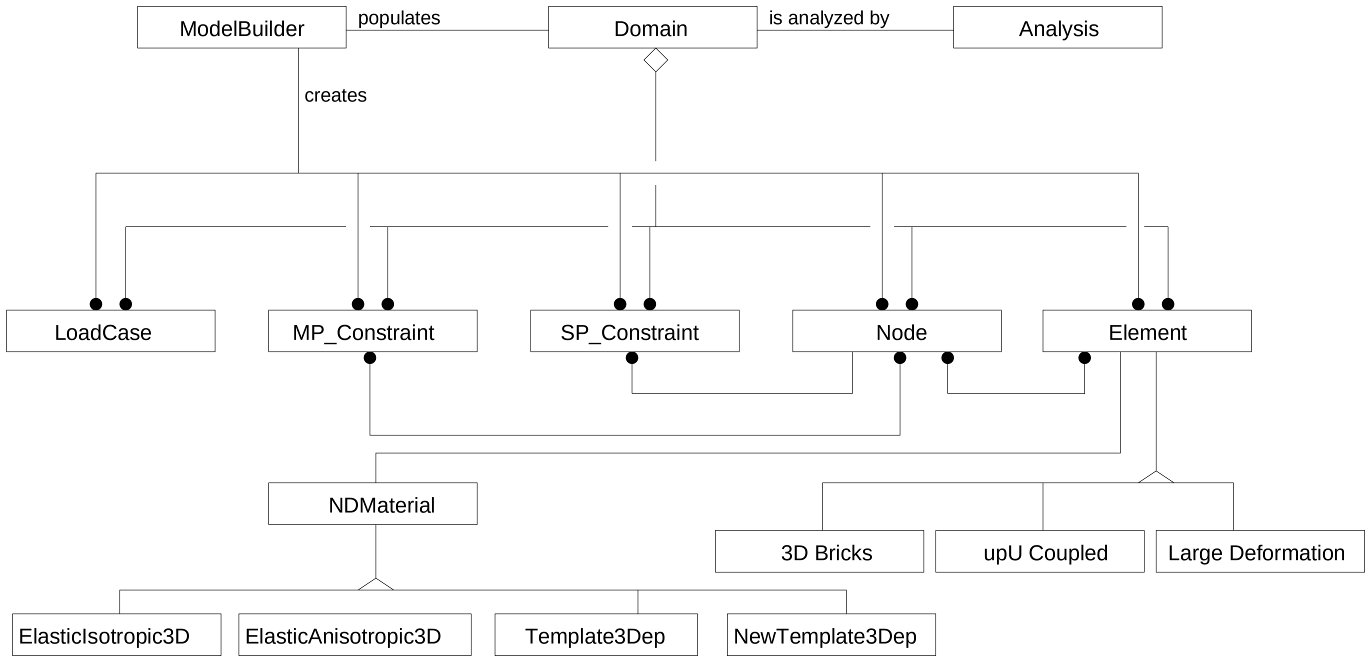


Figure 202.2: Class Diagram of Finite Element Model Classes

202.3.2.1 Modeling Classes

The modeling classes are responsible of creating the necessary components of the finite element model, such as nodes, elements, loads and constraints. There are a number of approaches proposed by various researchers. In some works, the user has the responsibility to create the finite element model in a single driver-type file Ross et al. (1992); Zeglinski et al. (1994); Cardona et al. (1994). In other works, an input file containing the model data is used to be read by the main program to create the model Forde et al. (1990); Dubois-Pelerin et al. (1992); Dubois-Pelerin and Zimmermann (1993); Menéntrey and Zimmermann (1993). Graphical interface for building models visually has also been proposed Ostermann et al. (1995); Mackie (1995).

In this research, the existing ModelBuilder interface class is reused to facilitate the finite element model construction. As shown in Figure 202.2, the ModelBuilder is associated with a single finite element Domain object. The interface (pure virtual function) buildFE_Model() must be redefined depending on the specific type of finite element model users want to build.

In parallel processing, PartitionedModelBuilder is used instead, in which the building process includes higher level control of building Subdomains from the PartitionedDomain. For each Subdomain, the PlaneFrameModelBuilder-type is invoked to build the finite element model on each Subdomain.

The Object-Oriented interface design of ModelBuilder through pure virtual function provides a consistent framework from which all kinds of engineering model can be readily extended.

202.3.2.2 Finite Element Model Class

In most of the works presented, main class abstractions used to describe a finite element model are: Node, Element, Constraint, Load and Domain Forde et al. (1990); Zimmermann et al. (1992a); Dubois-Pelerin et al. (1992); Dubois-Pelerin and Zimmermann (1993); Menéntrey and Zimmermann (1993); Pidaparti and Hudli (1993); Cardona et al. (1994); Chudoba and Bittnar (1995); Zahlten et al. (1995); Rucki and Miller (1996).

Node The most important feature of the Node class is the associativity with DOF class which contains the degree of freedoms of any specific instance of the Node class. The response quantities such as displacements of each DOF object will be stored in the Node class. Routines are available to set/get those solution quantities.

Element The functionality of an Element object is to provide the tangent stiffness, mass and the residual force corresponding to current loadings. Element class contains reference to its associated Node objects.

Element class is one of the most fundamental abstractions in Object-Oriented finite element software design. In this research, Element also acts as a container for material models, which is critical for simulations with nonlinear materials. Chudoba and Bittnar (1995) proposed a MaterialPoint object which is associated with GaussPoint object. In Zahlten et al. (1995), class abstractions such as cross section, material point, material law, yield surface, hardening rule and flow rule are introduced to model complicated materials within the Element class in an Object-Oriented flavor.

Jeremić and Yang (2002) present the complete formulation of Template3Dep material class, which is wrapped inside the Element class to enable a consistent interface for complex elastic-plastic material modeling.

Constraint There are two types of constraints in finite element simulations,

1. Single-Point constraints, which are applied to a specific DOF object;
2. Multi-Point constraints, which describe the relationship between more than one DOF objects.

In current implementation of OpenSees (version from 2005), the two classes SP_Constraint and MP_Contraint are designed but they do not handle the constraints. These two classes are responsible of setting up relations between Nodes and the constrained DOF_Groups. This will be covered shortly in Analysis class design.

Load There are also two types of loads that are commonly seen in finite element analysis:

1. node loads that act on specific Nodes;
2. element loads that act on specific Elements, which can be due to body forces, surface tractions, initial stresses and temperature gradients.

In the current (version version 2005) implementation of OpenSees, three extra classes are introduced to handle loading conditions, LoadPattern, NodalLoad and ElementLoad. The LoadPattern is a container class that provides methods in its interface to allow NodalLoad and ElementalLoad objects to be created, traversed and removed. As shown in Figure 202.2, each NodalLoad or ElementalLoad object is associated with a Node or Element object and is responsible of applying nodal or elemental loads to that object.

Domain The Domain class is the most important container class that is responsible of holding all components of the finite element model, i.e. all the Nodes, Elements, Constraints and Loads. Domain class acts as the interface between Analysis class and all the individual components of the finite element model. The interface of Domain enables component creation, information access and component removal.

202.3.2.3 Analysis

The Analysis class (McKenna, 1997) is responsible for forming and solving the governing equations for the finite element model. As for nonlinear problems, incremental solution techniques are required and iterative schemes such as Newton-Raphson needed to solver the nonlinear system of equations.

For incremental solution algorithm, the computational tasks are more involved for the finite element analysis.

- Assign equation numbers and map these to the nodal DOFs. This step can be of significant influence on the bandwidth of the coefficient matrix, which is inherently sparse due to the compact support of finite element formulation.
- Form the matrix equations using contributions from elements and nodes.
- Apply the constraints, which may involve transforming the element and nodal contributions or adding additional terms and unknowns to the matrix equations depending the method employed to handle constraints.
- Solve the matrix equations for the incremental nodal displacements.
- Determine the internal state and stresses in the elements.

The Object-Oriented design of the Analysis class is done by firstly breaking down the main tasks performed in a finite element analysis, abstracting them into separate classes, and then specifying the interface for these classes. The Analysis class is an aggregation of all the sub-functionality classes of following types:

1. SolutionAlgorithm class describes the complete computation procedure (steps) in the analysis.
2. AnalysisModel is a container class that stores and provides access to the following types of classes:
   (a) DOF_Group class represents the DOF at the Nodes or new DOF introduced into the analysis to enforce the constraints;
   (b) FE_Element class represents the real Elements in the Domain or they are introduced to add stiffness and/or load to the system of equations in order to enforce the constraints.
   It is worthwhile to mention that the FE_Elements and DOF_Groups have very important design implications although they might seem redundant at the first sight. The significance comes from the facts that
      i. they record the mapping between DOFs and equation numbers in the global system which greatly simplifies the interfaces of Node and Element class;

   ii. they also provide the interfaces for forming tangent and residual vectors which are used to form the global system of equations;
   iii. they are major utility classes of handling constraints.

3. Integrator defines how the FE_Elements and DOF_Groups contribute to the system of equations and how the response quantities should be updated given the solution to the global system of equations.
4. ConstraintHandler handles the constraint by creating adequate FE_Elements and DOF_Groups.
5. DOF_Numberer maps the equation number to the DOFs in the DOF_Groups.
6. SystemOfEqn encapsulates the global system of equations.

The aggregation of the Analysis object is shown in Figure 202.3.

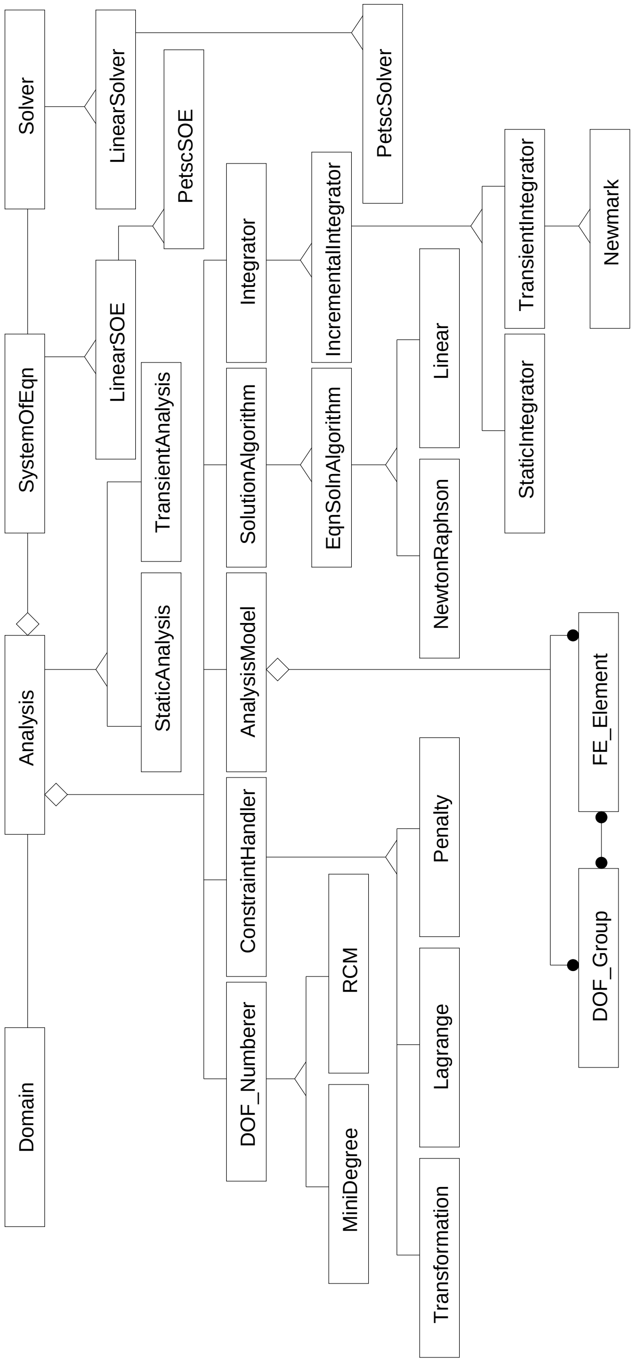


Figure 202.3: Class Diagram of Analysis Aggregation

Traditional program flow diagrams are used to describe how is the nonlinear finite element algorithm control flow implemented in OpenSees. These flow charts are organized as following:

Figure 202.4 shows the overall analysis algorithm flow for nonlinear finite elements.
Then this overall analysis flow is broken down into detailed subroutines, such as theIntegrator::newStep() and theAlgorithm::solveCurrentStep().

- Figure 202.5 explains in detail the function flow of theIntegrator::newStep(), which illustrates the (fairly standard) incremental finite element solution techniques implemented in OpenSees.
- Figure 202.6 shows the function flow of forming the tangent stiffness matrix, which is a loop assembling the global equation system involved in function theIntegrator::newStep().
- Figure 202.7 further describes the Newton-Raphson type iterative solution schemes involved in function theAlgorithm::solveCurrentStep().

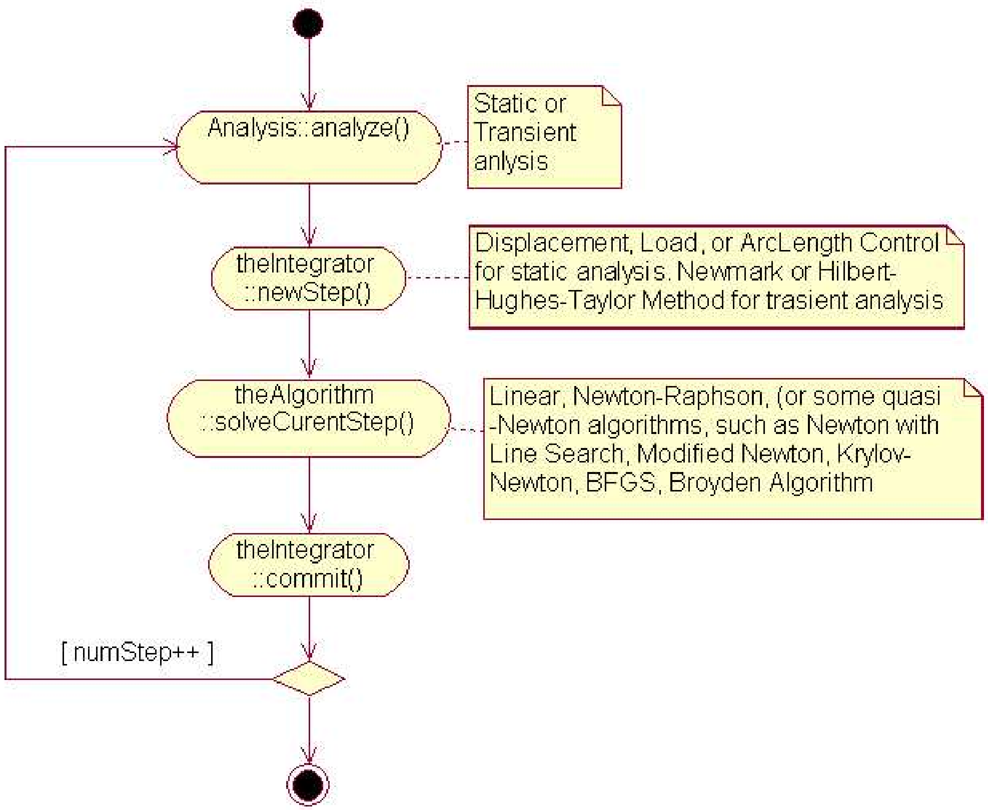


Figure 202.4: Overall Algorithm Flow Chart for Nonlinear Finite Element Analysis

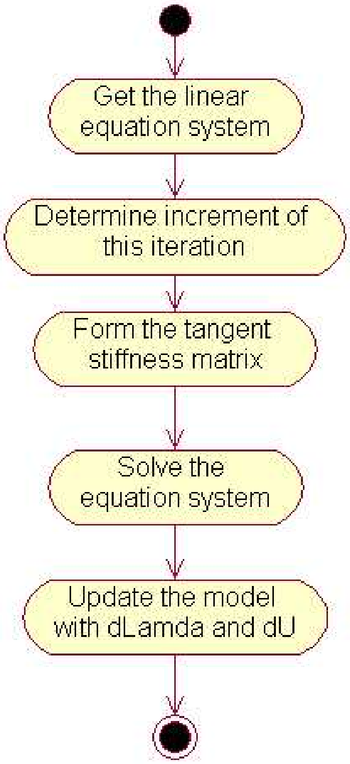


Figure 202.5: Detailed View: theIntegrator::newStep() - Incremental Solution Techniques for Nonlinear Finite Element Analysis

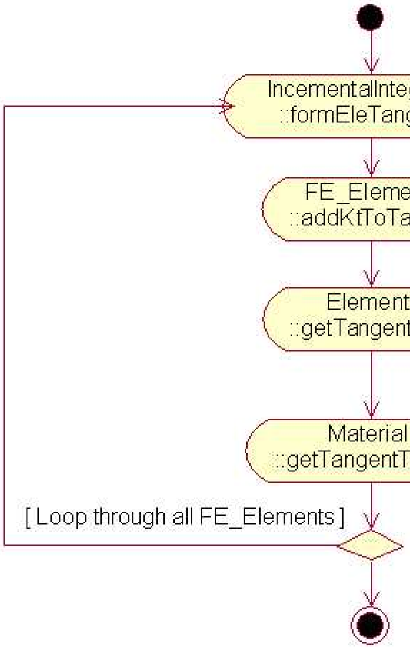


Figure 202.6: Detailed View: Assembly of Global Equation System in theIntegrator::newStep()

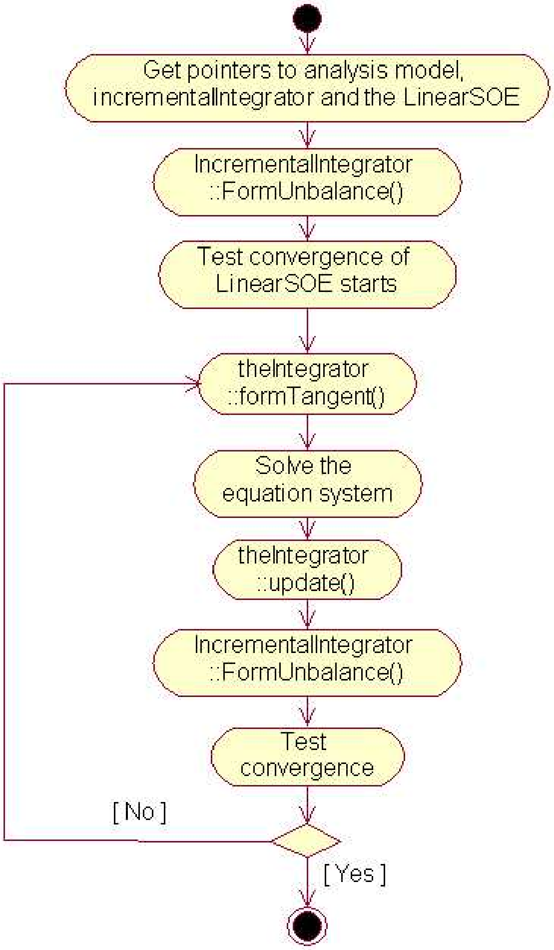


Figure 202.7: Detailed View: theAlgorithm::solveCurrentStep() - Newton-Raphson Iterative Schemes for Nonlinear Finite Element Analysis

Finite element simulations inherently are element-based operations, so little modification is needed to parallelize the algorithms described above, although special attention has to be paid to synchronize the computation among different processors. Figure 202.8 shows the activity flow for parallel nonlinear finite element simulations.

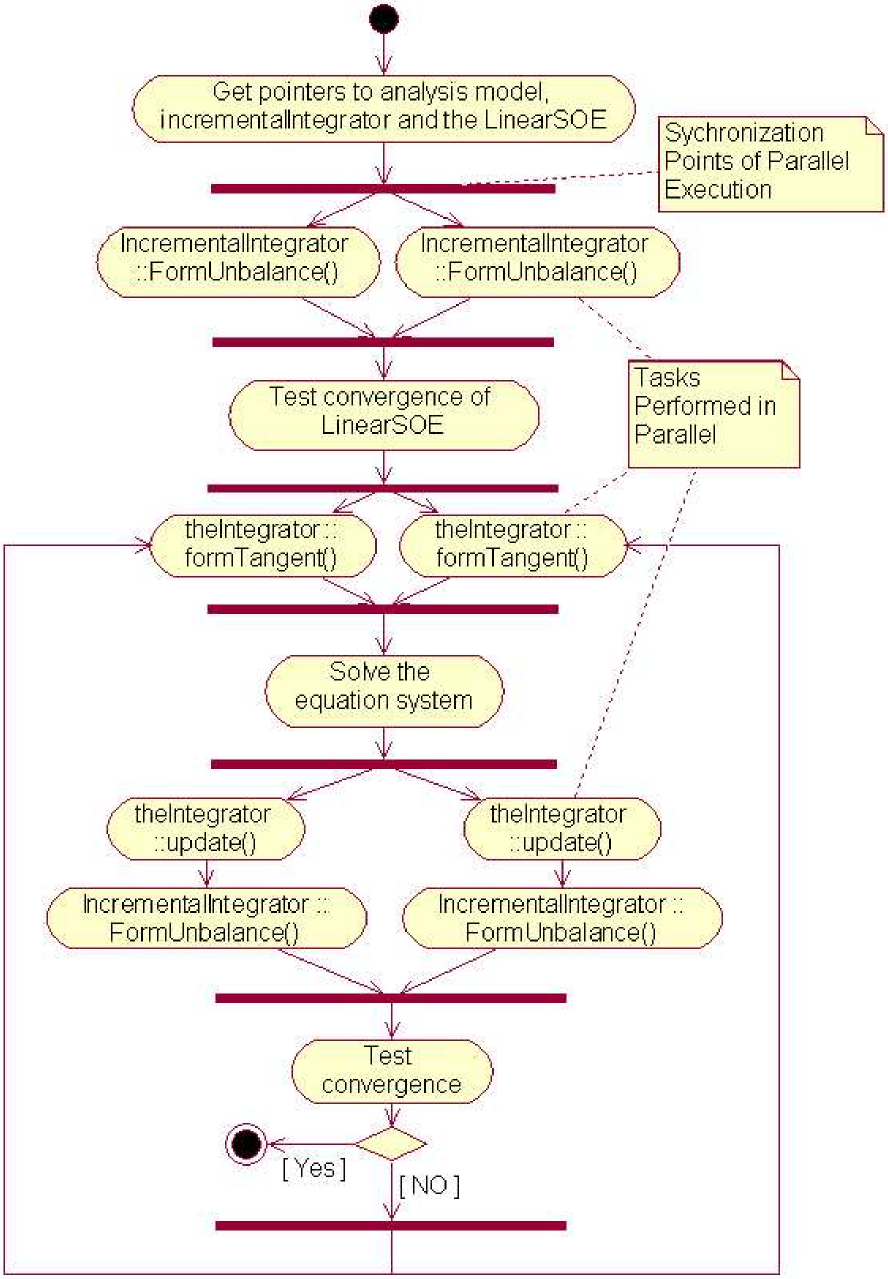


Figure 202.8: Parallel Activity Flow Diagram of Nonlinear Finite Element Analysis

202.3.2.4 Object-Oriented Domain Decomposition

There are three most notable designs of Domain Decomposition method in literature McKenna (1997).

1. Sause and Song (1994) presents an Object-Oriented design for linear static analysis using substructuring. The interface is restricted to substructuring or FETI Farhat and Roux (1991b) only, and repeated geometry limits the applicability of this design to large problems.
2. Archer (1996) proposes a SuperElement class that is a subclass of Element and has a Domain class aggregated. This design is conceptually inappropriate and it results excessive method calls as methods that are for the SuperElement must be called by the SuperElement on the associated Domain McKenna (1997).
3. Miller and Rucki (1993) introduces the Partition class which is associated with an Algorithm class. The Algorithm class is responsible for updating the state of a Partition so that it will be in equilibrium. Again, this design is good for substructuring type Domain Decomposition analysis. If we want to solve a problem before the interface solution can be determined, the design fails.

The current design of OpenSees McKenna (1997) proposes many new classes to facilitate flexible Object-Oriented Domain Decomposition. The main abstractions include PartitionedDomain, DomainDecompAnalysis, DomainDecompSolver Subdomain, DomainPartitioner and GraphPartitioner. The class diagram is shown in Figure 202.9.

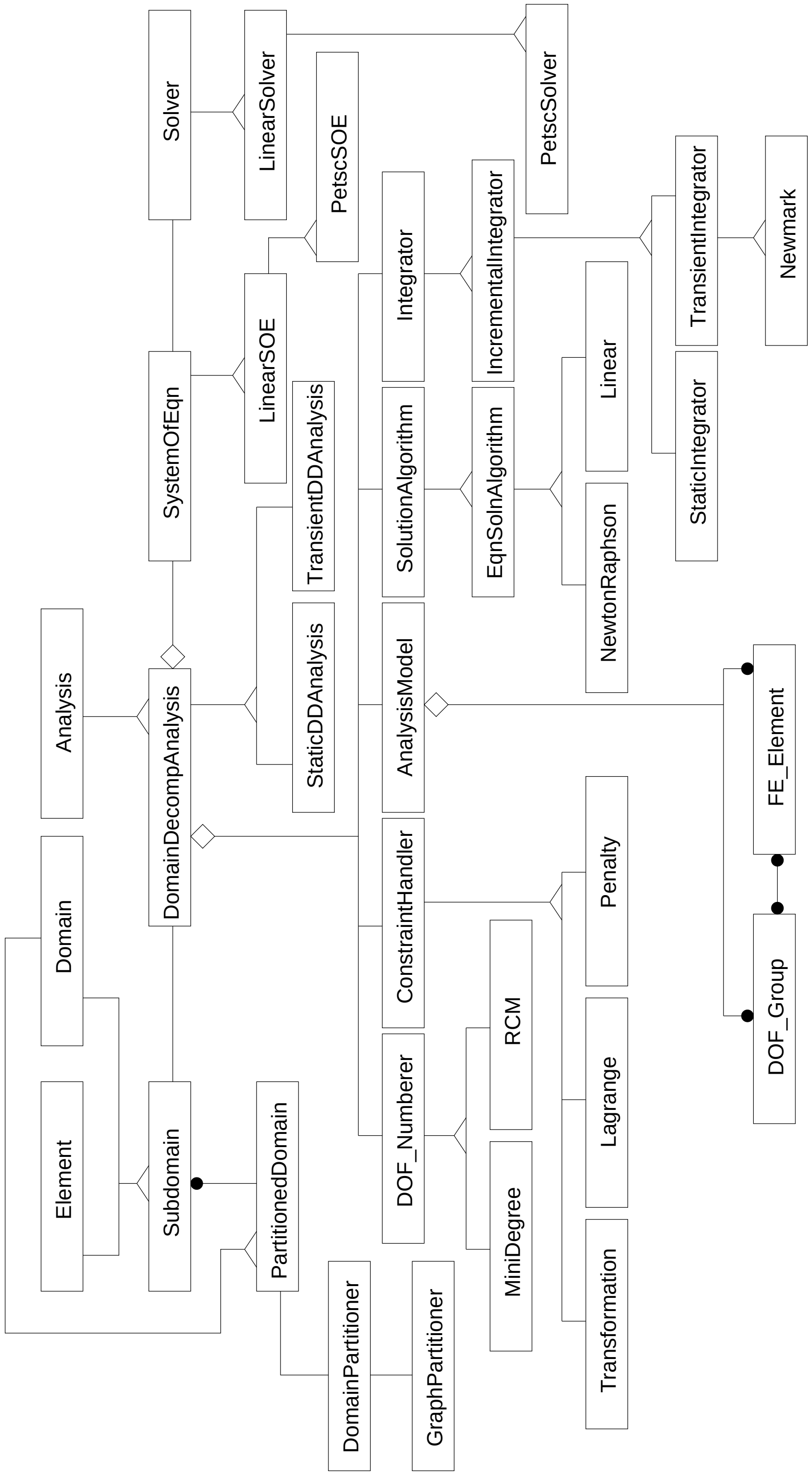


Figure 202.9: Class Diagram of Domain Decomposition Analysis

PartitionedDomain The PartitionedDomain class is a subclass of Domain whose objects can be partitioned into Subdomain objects. Aside from common functionality inherited from Domain, PartitionedDomain class provides methods for partitioning the Domain and retrieving information from Subdomains. PartitionedDomain the aggregation of Subdomains and is the major containing class in main compute process.

DomainPartitioner The DomainPartitioner class is responsible for performing the actual operation to split the PartitionedDomain. The DomainPartitioner will call its associated GraphPartitioner to partition the PartitionedDomain. It also provides the methods to migrate Elements, Nodes, Constraints, Loads amongst Subdomains.

DomainPartitioner is one of the most important utility class in OpenSees in the sense that all partitioning routine and data migration operations will be rooted from this class.

GraphPartitioner This class utilizes external graph partitioner to color the finite element connectivity graph, which will be constructed from the PartitionedDomain. The result will be fed back to DomainPartitioner to facilitate subsequent data distribution.

GraphPartitioner introduces graph partitioning into OpenSees and the main functionality of this class is to call API and provide necessary data structures from the specific application.

Subdomain The Subdomain class inherits from both Element and Domain. This has a dual-level design implication:

1. for the top PartitionedDomain, superclass Element is a proxy class of subclass Subdomain, in the sense that all the relevant operations on Elements invoked by PartitionedDomain will be redirected to the specific Subdomain;
2. for any specific Subdomain, it inherits all the interfaces of Domain to do all the computations required by PartitionedDomain.

202.3.2.5 Parallel Object-Oriented Finite Element Design

There has been much effort by researchers on parallel implementation of finite element computations, which can be categorized into either domain decomposition methods or parallel equation solving.

Domain decomposition is favored by many researchers due to its nice "divide and conquer" approach. The subdomains in the domain decomposition method are each assigned to a processing node, which will perform all the computations on that subdomain.

Of the domain decomposition methods, the substructuring method has been the most popular choice although other methods such as iterative substructuring Carter et al. (1989) and FETI (Finite Element Tearing and Interconnecting) Farhat and Roux (1991b); Farhat and Crivelli (1994) have also been used. In the substructuring method presented, static condensation is typically performed on the assembled system of equations.

Earlier works on parallel processing for inelastic mechanics focused on structural problems. We mention work by Noor et al. (1978); Utku et al. (1982); Storaasli and Bergan (1987) in which they used substructuring to achieve partitions. Fulton and Su (1992) developed techniques to account for different types of elements but used substructures of same element types (non–balanced computations). Hajjar and Abel (1988) developed techniques for dynamic analysis of framed structures with the objective of minimizing communications. Klaas et al. (1994) developed parallel computational techniques for elastic–plastic problems but tied the algorithm to the specific multiprocessor computers used (and specific network connectivity architecture). Farhat (1987) developed the so–called Greedy domain partitioning algorithm but stayed short of using redistribution of domains as a function of developed nonlinearities.

The major parallel programming model in OpenSees (McKenna, 1997) is the so-called Actor model, which is a mathematical model of concurrent computation that has its origins in Hewitt et al. (1973) . Actors Agha (1984) are autonomous and concurrently executing objects which execute asynchronously. Actors can create new actors and can send messages to other actors. The Actor model is an Object-Oriented version of message passing in which the Actors represent processes and the methods sent between Actors represent communications.

The Actor model adopts the philosophy that everything is an Actor. This is similar to the everything is an Object philosophy used by object-oriented programming languages, but differs in that object-oriented software is typically executed sequentially, while the Actor model is inherently concurrent, `http://en.wikipedia.org/wiki/Actor_model`.

An Actor is a computational entity with a behavior such that in response to each message received it can concurrently:

- send a finite number of messages to (other) Actors;
- create a finite number of new Actors;
- designate the behavior to be used for the next message received.

Note that there is no assumed sequence to above actions and that they could in fact be carried out in parallel.

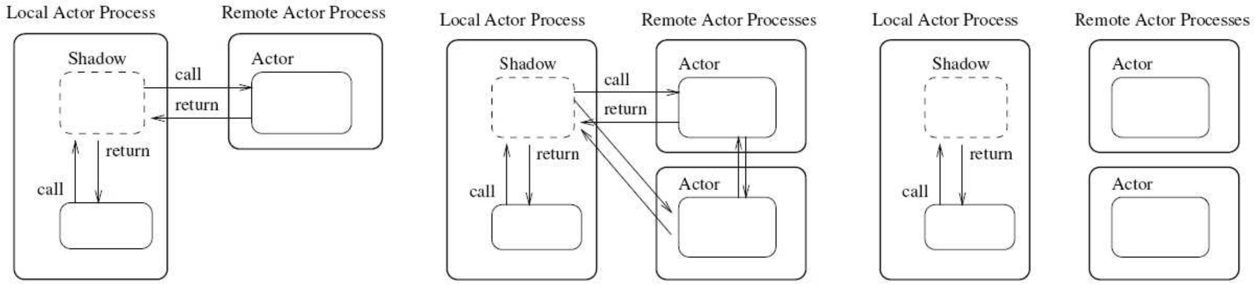
Communications with other Actors occur asynchronously (i.e. the sending Actor does not wait until the message has been received before proceeding with computation), which is the unblocking behavior.

Messages are sent to specific Actors, identified by address (sometimes referred to as the Actor's "mailing address"). As a result, an Actor can only communicate with Actors for which it has an address which it might obtain in the following ways:

- The address is in the message received;
- The address is one that the Actor already had, i.e. it was already an "acquaintance";
- The address is for a just created Actor.

The Actor model is characterized by inherent concurrency of computation within and among Actors, dynamic creation of Actors, inclusion of Actor addresses in messages, and interaction only through direct asynchronous message passing with no restriction on message arrival order.

In order to minimize the changes to the sequential Domain Decomposition design presented in previous sections, McKenna (1997) introduces the Shadow class. A Shadow object is an object in an Actor's local address space. Each Shadow is associated with one Actor or multiple Actors in the case of an aggregation. The Shadow object represents the remote object to the objects in the local Actor's space. The Shadow object is responsible for sending an appropriate message to the remote Actor or Actors if broadcasting. The remote Actor(s) will then, if required, return the result to the local Shadow object, which in turn replies to the local object. The communication process is shown in Figure 202.10.



(a) Actor - Communication Required

(b) Aggregate - Communication Required

(c) No Communication Required

Figure 202.10: Communication Pattern of Actor-Shadow Models McKenna (1997)

Some other new classes of parallel finite element programming are:

- Channel is the bridge through which the Actors and Shadows can communicate.
- Address represents the location of a Channel object in the machine space. Channel objects send/receive information to/from other Channel objects, whose locations are given by the Address objects.

- MovableObject is an object which can send its state from one actor process to another.
- ObjectBroker is an object in a local actor process for creating new objects.
- MachineBroker is an object in a local actor process that is responsible for creating remote actor processes at the request of Shadow objects in the same local process.

The relation between these classes is shown in Figure 202.11.

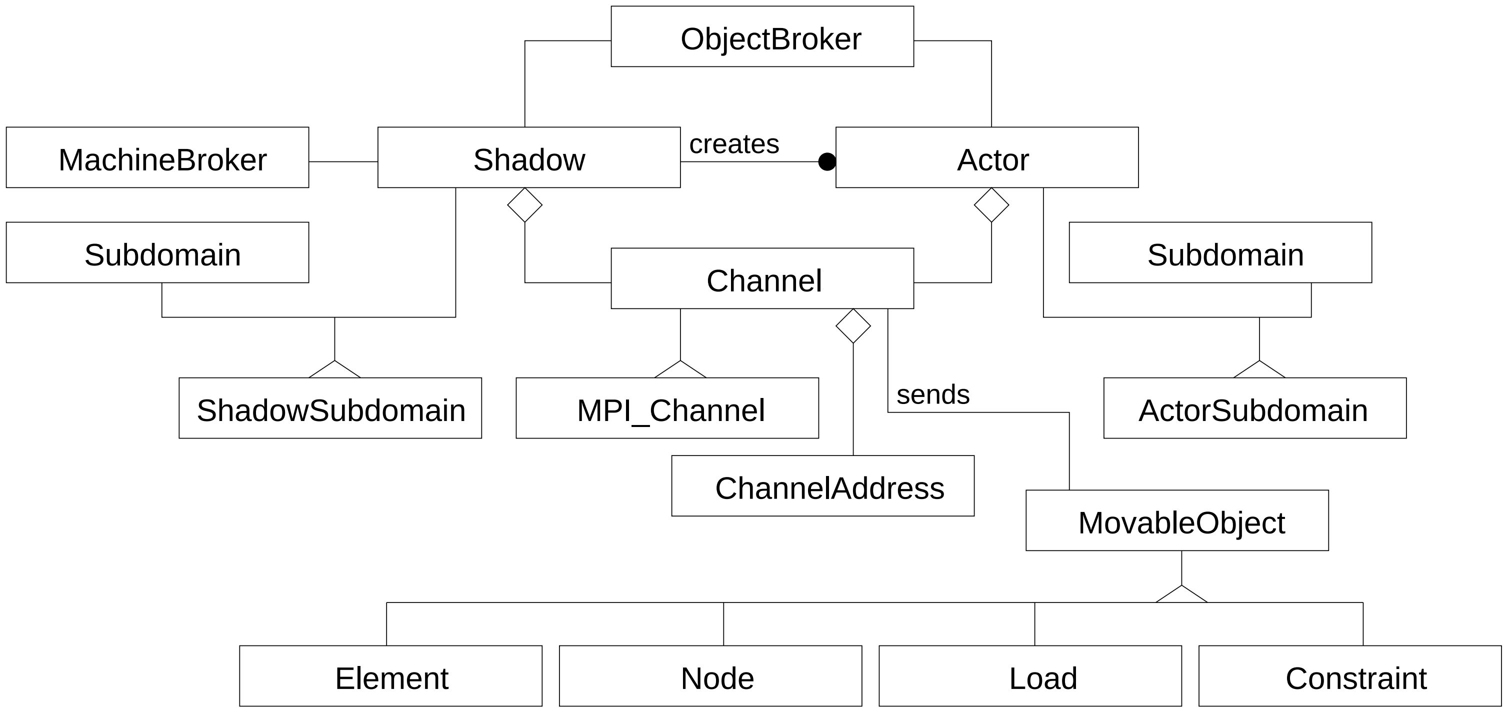


Figure 202.11: Class Diagram for Parallel Finite Element Analysis

202.3.3 Dual-Phase Adaptive Load Balancing

From the Figure 202.7, one can easily identify two computational phases that are fundamental to nonlinear elastic-plastic finite element simulations. One is well known as global level equation solving and the other is local level elemental calculations during which the elemental update happens for each element. In nonlinear elastic-plastic finite element simulations, the local computational phase can be much more expensive than the global equation solving phase due to the presence of complex material models and nonlinearity.

In this chapter, the implementation of proposed PDD algorithm has considered load balancing issues on both elemental level elastic-plastic computations and global level equation solving.

202.3.3.1 Elemental Level Load Balancing

The load balancing operation on constitutive level is built on the foundation of adaptive multilevel graph partitioning algorithm available through ParMETIS.

In this chapter, element-based graph is constructed from the Finite Element mesh on which the graph partitioning algorithm acts on to obtain partitions and/or repartitions. Each element will be assigned a vertex tag for identification.

When two elements at least share a single node, we assign an edge to both vertices because the element graph is deemed to be undirected, which means the edge is equally identified by two vertices without ordering required.

We creatively specify vertex weight to represent elemental level computational load for each vertex (element). In the implementation of this chapter, the vertex weight will be automatically updated as simulation progresses to reflect element computation cost. Performance timing has been added for constitutive update routines and the graph data structure will be refreshed every single iteration.

The last metric used is the vertex size of each vertex which basically contains the information that how much memory each vertex (element) requires in order to reproduce itself to other processes during data distribution. Adaptive load balancing is a multi-objective operation in the sense that both edge cut and data migration cost must be minimized simultaneously. The vertex size exactly describes the size of data that need to be shipped via communication. This metric must be correctly obtained for all available element types in order for the multi-objective load balancing algorithm to ensure the best performance.

202.3.3.2 Equation Solving Load Balancing

Parallel equation solving algorithm falls into two major different categories, direct solver and iterative solver.

Direct solver stems from Gaussian-type elimination and effective elimination tree is determined by the sparsity pattern of the stiffness matrix. Load balancing issue is addressed inherently when forming the elimination tree. Various packages such as SPOOLES and SuperLU provide scalable direct solutions to parallel equation systems. Chapter 110.5 discusses in further details about parallel direct solvers that are available as part of the release of this chapter.

Iterative solver has been the focus of this chapter in the sense that special care has been paid to achieve dynamic load balancing for each partition/repartition. The kernel of project-based iterative solvers is matrix-vector multiply. The issues of how to evenly distribute the stiffness matrix in parallel among different processors and how to reorder the sparse matrix to reduce data communications have been the focus of this chapter.

In order to achieve load balancing for parallel iterative solvers, parallel matrix/vector storage scheme and sparse matrix ordering are key factors. In the implementation of this chapter, even row-distribution of stiffness matrix among processing units is assumed. As shown in Figure 202.12, each processing unit has equal number of rows stored locally. The right hand side of the system is the force vector, which will be replicated for each processing unit. In this way, one can expect fastest matrix-vector multiply with the least amount of data needed to be communicated through network. As matrix-vector multiply is performed in parallel, load balancing issue is related to the number of nonzero numbers of the sparse stiffness matrix, which directly determines how many floating point multiplications are needed. In finite element computations, this nonzero pattern is determined by DOF numbering. Bandwidth reducing numbering scheme, or matrix ordering scheme, such as RCM Dongarra et al. (2003), can effectively lead to a sparse pattern that has similar number of nonzero elements on majority of rows as shown in Figure 202.12.

Finite element method inherently possesses compact support. Off-diagonal data of the stiffness matrix need to be synchronized among different processors. In order to reduce the extra overhead involved, in this chapter, several implementation solutions have been considered.

- Graph Partitioning Phase. As stated in previous chapters, minimizing edge-cut is one of the main objectives of the partitioning operation on the element graph. One extra benefit is that the bandwidth of the stiffness matrix will be greatly reduced. The number of nodes that need to be synchronized will be greatly reduced.

- DOF Numbering Phase. This phase is to renumber the DOFs of the finite element model after data redistribution in order to make sure contributions from local elements will sit on rows that are stored locally. This is done every time when the data migration is triggered. The idea is to start numbering the DOFs from local elements in Processor 1 to local elements in Processor N.

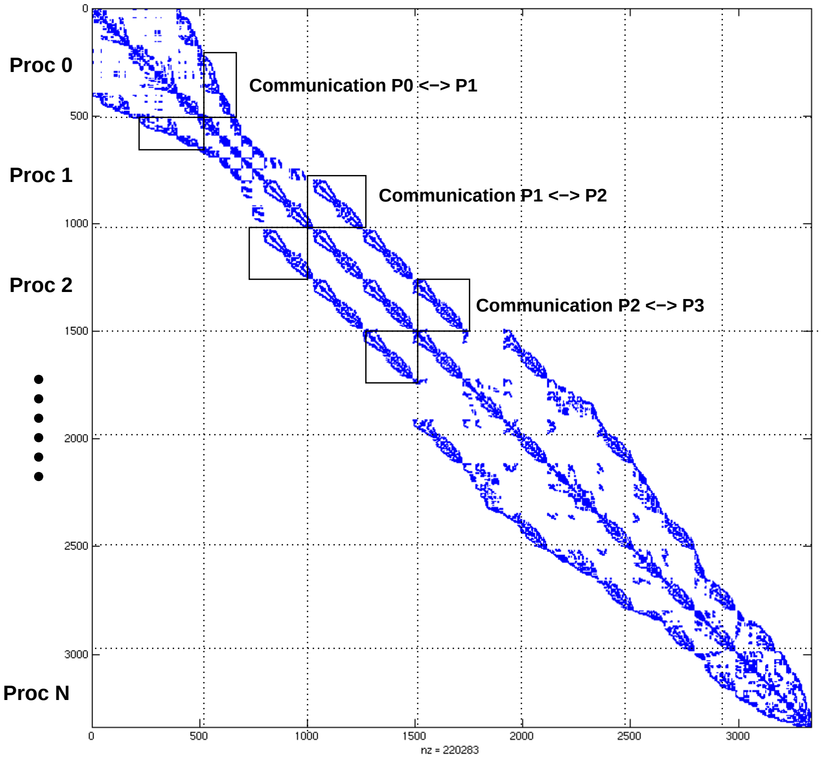


Figure 202.12: Parallel Data Organization of SFSI Equation System

In this way, when the global matrix is formed, local element stiffness matrix will always become clustered along the diagonal.

202.3.4 Object-Oriented Design of PDD

The parallel design of PDD basically follows Main-Follower algorithm structure as shown in Figure 202.13 and MPI has been adopted to facilitate inter-processor communications. The Actor/Shadow model described in previous sections is the used in PDD implementation and does nicely interact with parts of OpenSees framework, which uses Actor and Shadow classes to facilitate the inter-process communication between the main compute process and tied/follower compute processes.

- Main Compute Process

  Main compute process assumes the role to orchestrate the whole computation process. OpenSees uses tcl as an interpreter (or any other interpreted language that can be embedded into c or C++) to read input scripts from user. In parallel implementation of described here, main compute process

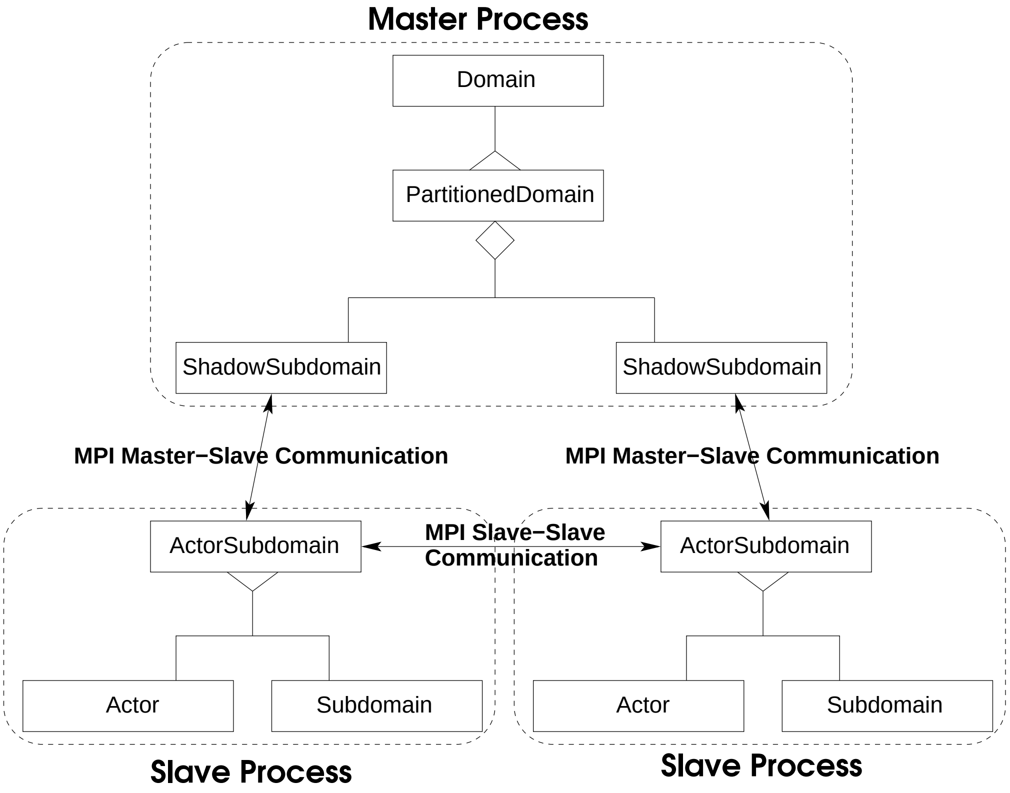


Figure 202.13: Main-Follower design used for PDD development.

is responsible for establishing the whole model for analysis and then distributing data among subprocessors. An important improvement in this chapter is that the main compute process does not actually create all finite element objects, whose memory space will only be allocated after they are sent to subdomains. This design helps avoid the high memory requirement on the main compute process side. Initial partitioning is done solely by main compute process process or in parallel by all working processes. Data movement is coordinated by the main compute process process, in which a complete element graph is kept intact.

As for repartitioning, the main compute process is still responsible for issuing commands to migrate data from this subprocessor to another even though the data is not in main compute process.

- Follower Compute Process

  The actor model has been used and modifications have been added to avoid unnecessary data communications. Basically speaking, actors in follower compute processes will be waiting for orders untill main compute process issues one and then do corresponding work on their own copy of data. The original design in OpenSees framework has disabled follower compute process to initiate communication, which means in order for a sub-processor to communicate with another sub-processor, it has to send all the data back to the main compute process first. This is highly inefficient and needed to be redeveloped, improved. In this research actor model has been implemented to enable direct communications between sub-processes and this improvement greatly reduced unnecessary communications.

All of the class designs for sequential version of OpenSees can be reused in parallel version following the Object-Oriented paradigm. There are some very important additions however in order to facilitate main–follower parallel processing. In this section, these classes will be revisited and updated/changes/improvements originally developed during this research will be explained thereafter.

- PartitionedDomain
  The PartitionedDomain class basically inherits all functionality from the Domain class in sequential version. This class acts as a container class in the main compute process. It differs from Domain class in the respect that all actions performed on the domain will be propagated to all subdomains when doing parallel processing.

- Subdomain
  The Subdomain is a child class of Domain. This class will be instanced by each follower compute process and it covers all functionality of the Domain class in sequential version. It can be called as an instance of Domain taking care of components only for the local follower compute process.

- ActorSubdomain & ShadowSubdomain
  The Actor/Shadow Subdomain classes are the most important classes for parallel version OpenSees. They are assuming the roles to initiate and facilitate all communications between main and follower compute processes. Both Actor/Shadow Subdomain will be instanced automatically when user creates follower compute process.
  ShadowSubdomain sits on main compute process. The function of this class is to represent a specific follower compute process in main compute process. Main compute process does not directly interact with follower compute process. Whatever action that needs to be performed by the follower compute process will be issued to ShadowSubdomain. This extra layer smooths the communication between main and follower nodes.
  On the other hand, ActorSubdomain sits on follower compute process and it hides main compute process from follower compute processes/nodes. All commands from main compute node will be received by ActorSubdomain and ActorSubdomain will match the command with some actions performed by Subdomain.
  Actor/Shadow Subdomain are extremely important classes in the parallel implementation of this

chapter. They carry all communication functionality required to finish the partition and adaptive repartition.

- Channel

  Channel is the class that really does the job of sending/receiving data between processors. Only MPI channel has been used in this chapter. Specific data structure, such as ID (integer array), vector (double array) or matrix needs to provide its own implementation for send/receive functionality.

- FEMObjectBroker

  This class is instanced only at follower compute processors, which is in charge of creating new model data for subdomains. This design isolates model creation from communication classes.

- Address

  Address class identifies parallel processes. With MPI channel used, the address corresponds to global process ID.

- DomainPartitioner

  DomainPartitioner assumes the responsibilities of invoking the GraphPartitioner and feeding necessary data to finish the partition/repartition. This class will also be in charge of data migration after partition/repartition is done.

- SendSelf & RecvSelf

  These two should be called functions rather than classes. SendSelf & RecvSelf are functions implemented to provide copy of model data to finish sending/receiving operations.

The old parallel design of OpenSees is not capable of performing elastic–plastic computations since it was designed and implemented for a single stage loading only. This single stage loading works fine for elastic analysis, but since elastic–plastic materials do have memory, staged loading is essential for any realistic computations with elastic–plastic material. This is particularly true for geotechnical and structural models, where simulations support for staged loading (self weight of soil medium for initial stress, construction process and subsequent static or dynamic loading) is essential if any modeling accuracy is to be achieved. One of new developments in this chapter was the addition of multi-stage elastic-plastic analysis. This improvement included modification of 3D solid and beam elements, Template3Dep/NewTemplate3Dep material models and DRM loading pattern for seismic analysis. Some of the old utility commands, such as "wipeAnalysis", were improved/redeveloped to enable parallel multi-stage analysis.

The most significant significant improvement developed during research over the old parallel design of OpenSees is the introduction of load balancing technique by adaptive graph partitioning algorithm through ParMETIS. Major improvements/updates have been introduced in PartitionedDomain, Actor/ShadowSubdomain, DomainPartitioner, FEM_ObjectBroker and Subdomain. Modifications done in this chapter also focus very much on performance issue. In order to reduce unnecessary data communication during partitioning/repartitioning, some functions have been rewritten. The functionality of Actor and ShadowSubdomain have been expanded so that any ActorSubdomain can initiate communication to another ActorSubdomain. The old design of OpenSees had to use main compute process as intermediate layer if subdomains want to exchange information.

For example, if Subdomain No. 1 needs to migrate an Element to Subdomain No. 2, the old design would issue a "remove Element" command from main compute process PartitionedDomain to Subdomain No. 1, then Subdomain No. 1 would remove the Element and send the Element back to main compute process, finally the Element would be migrated to Subdomain No. 2. We can clearly recognize the communication to main compute process is not necessary here. In order to develop adaptive load balancing while minimizing data redistribution cost, the improvement in this chapter is to allow ActorSubdomain at source Subdomain initiates communication with ActorSubdomain at target Subdomains and they can exchange information without recourse to main compute process. So the new communication pattern will be, again for the "migrate element" case, the main compute process will issue an "export element" command to Subdomain No. 1 and a "receive element from Subdomain No. 1" command to Subdomain No. 2, and then the element information will be directly sent from Subdomain No. 1 to No. 2.

Details of implementation are given in following sections.

202.3.4.1 MPI_Channel

- Functions sendnDarray and recvnDarray have been added to facilitate the data communication of Template3D material classes, which are based on nDarray tensor data structures.

  `int MPI_Channel::sendnDarray(int,int, const nDarray&, ChannelAddress*)`

  `int MPI_Channel::recvnDarray(int,int, const nDarray&, ChannelAddress*)`

202.3.4.2 MPI_ChannelAddress

- Function getOtherTag has been added to get MPI global ID for the specific MPI_Channel. This function is mainly used for data migration. It provides the MPI global communicator ID of the target process which the next communication will be directed to.

`int MPI_ChannelAddress::getOtherTag(void)`

202.3.4.3 FEM_ObjectBroker

- New functionality to instance 3D continuum brick elements has been added to getNewElement function.

  `Element* FEM_ObjectBroker::getNewElement(EightNodeBrickTag)`

- New functionality to instance Template3D/NewTemplate3D material models for continuum brick elements has been added to getNewNDMaterial function.

  `NDMaterial* FEM_ObjectBroker::getNewNDMaterial(int)`

- Template3D material is a stand-alone material library designed for general elastic-plastic materials. User can define separately YieldSurface, PotentialSurface, Scalar Evolution Law and Tensorial Evolution Law. Various material models have been implemented in OpenSees Jeremić and Yang (2002), such as Cam Clay, Drucker Prager and von Mises yield/potential surfaces, Armstrong Frederick nonlinear kinematic hardening law and bounding surface plasticity. All the material models have to be instanced by FEM_ObjectBroker during parallel processing.

  `YieldSurface* FEM_ObjectBroker::getYieldSurfacePtr(int)`

  `PotentialSurface* FEM_ObjectBroker::getPotentialSurfacePtr(int)`

  `EvolutionLaw_S* FEM_ObjectBroker::getEL_S(int)`

  `EvolutionLaw_T* FEM_ObjectBroker::getEL_T(int)`

- NewTemplate3D material is a newly designed material library which includes more advanced elastic-plastic constitutive models for geomaterials, such as Dafalias and Manzari 2004 model. The design of NewTemplate3D extends the principle of Template3D, in which key parameters describing plasticity model are abstracted as different class objects, such as YieldFunction, PlasticFlow, etc. In order to reduce unnecessary data allocation, new MaterialParameter class has been developed to carry all material parameters. New ElasticState has been used to store all intermediate and/or committed stress/strain data. All these material classes have to be instanced by FEM_ObjectBroker during parallel processing and new functions have been implemented in this chapter.

  `MaterialParameter* FEM_ObjectBroker::getNewMaterialParameterPtr(void)`

  `ElasticState* FEM_ObjectBroker::getNewElasticStatePtr(int)`

  `YieldFunction* FEM_ObjectBroker::getNewYieldFunctionPtr(int)`

  `PlasticFlow* FEM_ObjectBroker::getNewPlasticFlowPtr(int)`

  `ScalarEvolution* FEM_ObjectBroker::getNewScalarEvolutionPtr(int)`

  `TensorEvolution* FEM_ObjectBroker::getNewTensorEvolutionPtr(int)`

202.3.4.4 Domain

- Timing routines have been added to update function to measure computation time of constitutive level iterations for each element during every single loading increment. This metric will be assigned to the corresponding vertex of the element graph as the vertex weight. This metric represents element-level computational load against which subsequent load balancing techniques will be applied.

202.3.4.5 PartitionedDomain

- addElementalLoad function has been added to add ElementalLoad into LoadPattern, which was not supported in the old design.

  `bool PartitionedDomain::addElementalLoad(ElementalLoad*, int)`

- repartition function has been implemented to initiate adaptive repartitioning on the element graph of the Domain after every loading increment.

  `int PartitionedDomain::repartition(int)`

202.3.4.6 Node & DOF_Group

- sendSelf and recvSelf functions for Node class have been changed mainly to deal with the DOF_Group object associated with the Node. In the old design of parallel version of OpenSees, only one-step static domain partitioning would be invoked so that there is no need to pass the DOF_Group. But in this chapter, adaptive load balancing is developed to achieve better performance. The Node class should keep the information of its own DOF_Group, which guarantees the consistency of the DOF_Graph of the whole Domain. This point is extremely important when user tries to invoke Transformation constraint handler on the DOF_Graph. The addition of this feature in Node improved the robustness of the whole program.

- DOF_Group is a class carries information about the DOF_Graph of the analysis model, which will be used to finish assembling the stiff/mass/damping matrices. Each Node has its own DOF_Group to record the IDs of degree of freedoms in the global analysis model. Function unSetMyNode has been introduced to avoid segmentation fault. The reason is that after each round of repartitioning, if data movement is required, the AnalysisModel will be wiped off but Nodes are still in existence. Introduction of unSetMyNode function separates Node from its DOF_Group so the DOF_Group can be wiped and regenerated for the new model. `void DOF_Group::unSetMyNode(void)`

202.3.4.7 DomainPartitioner

DomainPartitioner is one of the most extensively changed classes in this chapter. This class acts as the entry point for PartitionedDomain to do domain decomposition and it basically has been rewritten to introduce new partition/repartition functionality and new data structures.

- Function repartition is implemented to do repartitioning after each loading increment. Partition and repartition are both implemented in parallel through ParMETIS library in this chapter. This function will collect ElementGraph from each Subdomain and pass them to GraphPartitioner. The global ElementGraph will be kept intact from which connectivity/adjacency information will be gathered to assemble child ElementGraphs and provide initial graph distribution data for repartition routines. After repartitioning by ParMETIS finishes, the function will verify the new partition against the original one to see if data redistribution is required to achieve load balancing. This repartition function also acts as a commander to control the data migration for adaptive load balancing. It issues commands to ShadowSubdomain to export/import Nodes, Elements, Constraintss, Loads, etc.

- The old design of OpenSees used multiplication of prime numbers as index number to record which partitions a specific node belongs to. This is a very good idea because with this approach, we only need one integer for each node to keep track of node partitions, which can be called as an index number for the node. The idea was to name each Subdomain with one specific prime number, if a node belongs to this Subdomain, we would multiply the index number of the node with the prime number of this Subdomain. In order to determine if a node belongs to on specific Subdomain, all we need is to divide the index number of the node with the prime number the Subdomain represents to see if we can get zero residual.

- The drawback of the old data structure based on prime numbers is that it only works when the number of processing units is small, say less than 16. In 3D continuum models, a single node might belong to up to 8 partitions simultaneously, which happens when a corner node sits on intersections of different Subdomains. As we know, prime numbers grow up very fast, multiplication with 8 prime numbers can easily overflow the index number of the node. A new data structure inspired by the Compressed Sparse Row (CSR) storage format popular in sparse matrix calculations has been introduced into in this chapter to solve the problem. One integer array has been used to store the partition data of all nodes, i.e. which partitions this node belongs to. Another integer array has been employed to record the count of partitions for each node. With these two arrays, we can load as many partitions as we want in our parallel processing.

202.3.4.8 Shadow/ActorSubdomain

As mentioned in previous sections of this chapter, Shadow/ActorSubdomain are the most important classes in parallel design of OpenSees McKenna (1997). ShadowSubdomains represent Subdomains in the main compute process PartitionedDomain. If PartitionedDomain requires one specific Subdomain to carry out some operations, it will send out orders to the ShadowSubdomain associated with the target Subdomain. Then the ShadowSubdomain sets up communication channel to communicate with the Subdomain through ActorSubdomain. ActorSubdomain, on the other hand, sits on each child process as an agent receiving and processing incoming operation requests. The major improvements in this chapter include new functionality for adaptive repartitioning and data migration, and several other minor changes to reduce unnecessary data communications, such as when the Subdomain is required to removeElement, the new design won't send the element information out, etc. New features will be introduced in this section.

- ShadowActorSubdomain_Partition
  New design used ParMETIS to do parallel graph partitioning instead of sequential partitioning by METIS in old design. This improvement helps to reduce partition/repartition overhead and enable the parallel adaptive repartitioning for PDD algorithms proposed in this chapter.

- ShadowActorSubdomain_BuildElementGraph
  In order to provide input graphs for adaptive load balancing, all Subdomains have to construct their own subElementGraph, which will be fed into ParMETIS routines for repartitioning.

- ShadowActorSubdomain_Repartition
  The repartitioning is implemented in parallel in this chapter so this entry point is set in the ActorSubdomain for each Subdomain.

- ShadowActorSubdomain_reDistributeData
  If data migration is needed to achieve load balancing, the main compute process will orchestrate the data redistribution process and the functionality here helps to facilitate the data communications between processes. This is one of the major additions to the existing design. Starting from this point, the ActorSubdomain is able to handle all required data movement on its own and ActorSubdomains representing other Subdomains will connect to the current working ActorSubdomain to receive/send data. Logically only one ActorSubdomain will be doing ShadowActorSubdomain_reDistributeData while others including the main compute process will be listening to separate MPI port for data migration requests.

- ShadowActorSubdomain_recvChangedNodeList

  This function is used to simplify the data migration routine. With this function, only Nodes/Elements and their associated Constraints, Loads etc. need to be moved between processors.

- ShadowActorSubdomain_ChangeMPIChannel

  This function prepares the current ActorSubdomain for messages from some specific processes. It changes the destination/source for subsequent outgoing/incoming communications, which helps redistributing data after load balancing.

- ShadowActorSubdomain_restoreChannel

  The default communication pattern in the old design of OpenSees was one to one, main to follower computer processes. This function helps restoring communication patterns of the whole model after data redistribution finishes.

- ShadowActorSubdomain_swapNodeFromInternalToExternal

  Nodes that only belongs to one single Subdomain is called internal nodes whose information will be stored only in that specific Subdomain. While for those nodes that belong to more than one Subdomains, their information should be accessible from all Subdomains with which the nodes are associated. Those nodes are called external nodes instead. It is possible that former internal nodes to one Subdomain become external after the adaptive repartitioning. What the old design would do is to remove the internal nodes from that Subdomain, gather the information back to the main compute process and then distribute it externally among those Subdomains as indicated by the newly obtained partitions. The improvement in this chapter avoids unnecessary data communication between current working Subdomain and the main compute process Domain. We can just swap the node in working Subdomain from internal status to external status and then export them to other specified Subdomains. This new design can improve performance if the data migration is extensive by avoiding unnecessary communications.

- ShadowActorSubdomain_swapNodeFromExternalToInternal

  This function is introduced due to the same reason as described previously although now the swapping direction is in reverse. It is noted that along with the swapping, removing operations must be invoked for those Subdomains that does not contain the node anymore.

- ShadowActorSubdomain_exportInternalNode

  This function handles the situation when a Node does not belong to the current Subdomain after adaptive repartitioning. The node will be removed from current Subdomain and exported to other Subdomains specified by the graph repartitioning. This again avoids the unnecessary data communication to/from main compute process by directly sending data to other Subdomains.

- ShadowActorSubdomain_resetRecorders
  The Recorders have to be reset after data migration to reflect component changes in each Subdomain.

202.3.4.9 Send/RecvSelf

As stated in previous sections, Send/RecvSelf must be provided by all domain components to finish data communication operations, such as Nodes, Elements, Loads, Constraints, Materials etc. In this chapter, new communication functions have been developed for EightNodeBrick element, ElasticIsotropic3D material, Template3Dep/NewTemplate3Dep material. The basic requirement to implement Send/RecvSelf is to replicate the source object instance in target process. For the old design, only one-step initial partitioning is performed and thus greatly simplifies the Send/RecvSelf routines because all the analysis-related information is null or void and only geometry-related data need to be transferred. But in this chapter, data migration is needed periodically to achieve load balance so the Send/RecvSelf has to be redesigned to carry analysis-related information besides the geometry model data. This is extremely important for Element and Material classes because they contain intermediate iteration/solution data of nonlinear finite element simulations. Figure 202.15 shows the class diagrams of brick Element and the associated Template3Dep material model. Send/RecvSelf operations have been implemented also for all classes associated with Template3Dep which are necessary to define a complete material model, such as Cam Clay, Drucker Prager and von Mises PotentialSurfaces, Cam Clay, Drucker Prager and von Mises YieldSurfaces, linear and nonlinear isotropic and kinematic hardening rules, etc.

202.3.5 Graph Partitioning

Graph partitioning approach has been extensively used in implementing domain decomposition type parallel finite element method. The element-based graph naturally becomes the favorite due to the fact that elemental operation forms the fundamental calculation unit in finite element analysis.

In this chapter, element graph has been constructed upon which graph partitioning algorithm acts to get domain decomposition for parallel finite element analysis. In the current implementation of this chapter, vertices of the element graph represent elements of the analysis model. Vertex weight is then specified as the computational load of each element. In elastic-plastic finite element simulations, the most expensive part has shown to be the elemental level calculations, which include constitutive-level stress update (strain-driven constitutive driver assumed) and formulation of elastic-plastic modulus (or so-called tangent stiffness tensor/matrix). In this research, the wall clock time used by elemental calculations has been dynamically collected and specified as the corresponding vertex weight for each element. The elemental calculation time clearly tells whether the element is elastic or plastified. With this timing metric, the graph can effectively reflect load distribution among elements thus load balancing repartition can be triggered on the graph to redistribute element between processors to achieve more balanced elastic-plastic calculation.

On the other hand, vertex size has to be defined for repartitioning problem as mentioned in previous sections. In this research, vertex size has been specified to be redistribution cost associated with each element. This information depends on the parallel implementation of the software and is discussed in the section immediately following.

202.3.5.1 Construction of Element Graph

Each element is considered as one vertex in the element graph. An edge is formed when two elements share a common node. In this chapter, the graph structure is assumed to be undirected, which means the same edge will be added to both vertex ends. The edge is weightless in our application considering the fact that the purpose of minimizing edge-cut is to reduce the data migration when assembling global stiffness matrix. In that sense, the edge of element graph should carry the same weight or, more directly no weight at all.

202.3.5.2 Interface to ParMETIS/METIS

Interfaces to both ParMETIS and METIS have been implemented in this chapter. ParMETIS is the parallel implementation of METIS and new adaptive repartitioning functionality is only available through ParMETIS.

All of the graph routines in ParMETIS/METIS take as input the adjacency structure of the graph, the weights of the vertices and edges (if any), and an array describing how the graph is distributed among the processors Karypis et al. (2003). The structure of the graph is represented by the compressed storage format (CSR), extended for the context of parallel distributed-memory computing. We will first describe the CSR format for serial graphs and then describe how it has been extended for storing graphs that are distributed among processors.

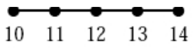
- Serial CSR Format The CSR format is a widely-used scheme for storing sparse graphs. Here, the adjacency structure of a graph is represented by two arrays, $xadj$ and $adjncy$. Weights on the vertices and edges (if any) are represented by using two additional arrays, $vwgt$ and $adjwgt$. For example, consider a graph with n vertices and m edges. In the CSR format, this graph can be described using arrays of the following sizes:

$$xadj[n+1], \quad vwgt[n], \quad adjncy[2m], \quad \text{and} \quad adjwgt[2m] \qquad (202.1)$$

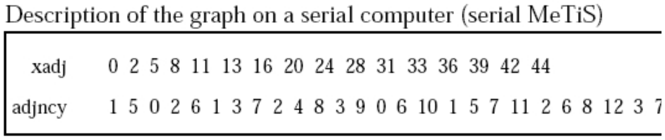
Note that the reason both $adjncy$ and $adjwgt$ are of size $2m$ is because every edge is listed twice (i.e., as (v, u) and (u, v)). Also note that in the case in which the graph is unweighted (i.e., all vertices and/or edges have the same weight), then either or both of the arrays $vwgt$ and $adjwgt$ can be set to $NULL$. ParMETIS_V3_AdaptiveRepart additionally requires a $vsize$ array. This array is similar to the $vwgt$ array, except that instead of describing the amount of work that is associated with each vertex, it describes the amount of memory that is associated with each vertex.

The adjacency structure of the graph is stored as follows. Assuming that vertex numbering starts from 0 (C style), the adjacency list of vertex i is stored in array $adjncy$ starting at index $xadj[i]$ and ending at (but not including) index $xadj[i+1]$ (in other words, $adjncy[xadj[i]]$ up through and including $adjncy[xadj[i+1]-1]$). Hence, the adjacency lists for each vertex are stored consecutively in the array $adjncy$. The array $xadj$ is used to point to where the list for each specific vertex begins and ends. Figure 202.14(a) illustrates the CSR format for the 15-vertex graph shown in Figure 202.14(b). If the graph has weights on the vertices, then $vwgt[i]$ is used to store the weight of vertex i . Similarly, if the graph has weights on the edges, then the weight of edge $adjncy[j]$ is stored in $adjwgt[j]$. This is the format that is used by (serial) METIS library routines.

- Distributed CSR Format ParMETIS uses an extension of the CSR format that allows the vertices of the graph and their adjacency lists to be distributed among the processors. In particular, PARMETIS assumes that each processor P_i stores n_i consecutive vertices of the graph and the corresponding m_i edges, so that $n = \sum_i n_i$, and $2m = \sum_i m_i$. Here, each processor stores its local part of the graph in the four arrays $xadj[n_i + 1]$, $vwgt[n_i]$, $adjncy[m_i]$, and $adjwgt[m_i]$, using the CSR storage scheme. Again, if the graph is unweighted, the arrays $vwgt$ and $adjwgt$ can be set to $NULL$. The straightforward way to distribute the graph for PARMETIS is to take n/p consecutive adjacency lists from $adjncy$ and store them on consecutive processors (where p is the number of processors). In addition, each processor needs its local $xadj$ array to point to where each of its local vertices' adjacency lists begin and end. Thus, if we take all the local $adjncy$ arrays and concatenate them, we will get exactly the same $adjncy$ array that is used in the serial CSR. However, concatenating the local $xadj$ arrays will not give us the serial $xadj$ array. This is because the entries in each local $xadj$ must point to their local $adjncy$ array, and so, $xadj[0]$ is zero for all processors. In addition to these four arrays, each processor also requires the array $vtxdist[p + 1]$ that indicates the range of vertices that are local to each processor. In particular, processor P_i stores the vertices from $vtxdist[i]$ up to (but not including) vertex $vtxdist[i + 1]$.



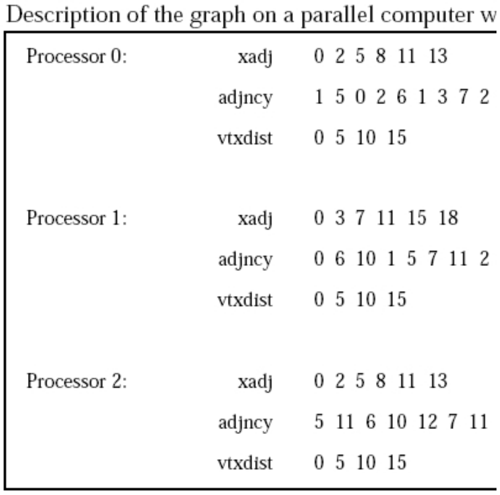
(a) A sample graph



Description of the graph on a serial computer (serial MeTiS)

xadj	0 2 5 8 11 13 16 20 24 28 31 33 36 39 42 44
adjncy	1 5 0 2 6 1 3 7 2 4 8 3 9 0 6 10 1 5 7 11 2 6 8 12 3

(b) Serial CSR format



Description of the graph on a parallel computer w

Processor 0:	xadj	0 2 5 8 11 13
	adjncy	1 5 0 2 6 1 3 7 2
	vtxdist	0 5 10 15
Processor 1:	xadj	0 3 7 11 15 18
	adjncy	0 6 10 1 5 7 11 2
	vtxdist	0 5 10 15
Processor 2:	xadj	0 2 5 8 11 13
	adjncy	5 11 6 10 12 7 11
	vtxdist	0 5 10 15

(c) Distributed CSR forn

Figure 202.14: An example of the parameters passed to PARMETIS in a three processor case Karypis et al. (2003).

Figure 202.14(c) illustrates the distributed CSR format by an example on a three-processor system. The 15-vertex graph in Figure 202.14(a) is distributed among the processors so that each processor gets 5 vertices and their corresponding adjacency lists. That is, Processor Zero gets vertices 0 through 4, Processor One gets vertices 5 through 9, and Processor Two gets vertices 10 through 14. This figure shows the *xadj*, *adjncy*, and *vtxdist* arrays for each processor. Note that the *vtxdist* array will always be identical for every processor. All five arrays that describe the distributed CSR format are defined in PARMETIS to be of type *idxtype*. By default *idxtype* is set to be equivalent to type *int* (i.e., integers). However, *idxtype* can be made to be equivalent to a *short int* for certain architectures that use 64-bit integers by default. (Note that doing so will cut the memory usage and communication time required approximately in half.) The conversion of *idxtype* from *int* to *short* can be done by modifying the file parmetis.h. (Instructions are included there.) The same *idxtype* is used for the arrays that store the computed partitioning and permutation vectors.

When multiple vertex weights are used for multi-constraint partitioning, the c vertex weights for each vertex are stored contiguously in the *vwgt* array. In this case, the *vwgt* array is of size nc, where n is the number of locally stored vertices and c is the number of vertex weights (and also the number of balance constraints).

New GraphPartitioner class ParMETIS has been developed in this chapter to provide seamless interface to adaptive partitioning/repartitioning routines.

202.3.6 Data Redistribution

Data redistribution after repartitioning has been a challenging problem which needs careful study to guarantee correctness of subsequent analysis. In this research, Object-Oriented philosophy has been followed to abstract container classes to facilitate analysis and model data redistribution after repartition. As for the initial partitioning, only model data, such as geometry parameters, has to be exported to sub-processors, while in adaptive repartitioning finite element simulation, analysis data has to be moved as well. It is extremely important to have well-designed container classes to carry data around. Basic units of finite element analysis, such as nodes and elements naturally become our first choices. Not to give up generality, the design in OpenSees adopts basic iterative approach for nonlinear finite element analysis Crisfield (1997b), important intermediate analysis data include trial data, commit data, incremental data, element residual, element tangent stiffness, etc. Vertex size of each element has been defined as the total number of bytes that have to be transferred between sub-processors.

1. Node

   Other than geometric data such as node coordinates and number of degree of freedoms, the Node class contains nodal displacement data which should be sent together with the node to preserve continuity of the analysis model.

2. Element

   Element class is the basic construction unit in finite element model. In the design of this research, Element class keeps internal links to Template3D material class Jeremić and Yang (2002). In order to facilitate elastic-plastic simulation, EPState class is constructed to hold all the intermediate response data. This object-oriented abstraction greatly systematize the communication pattern. The information on class design is shown in the class diagram Figure 202.15 by Rational Rose Boggs and Boggs (2002).

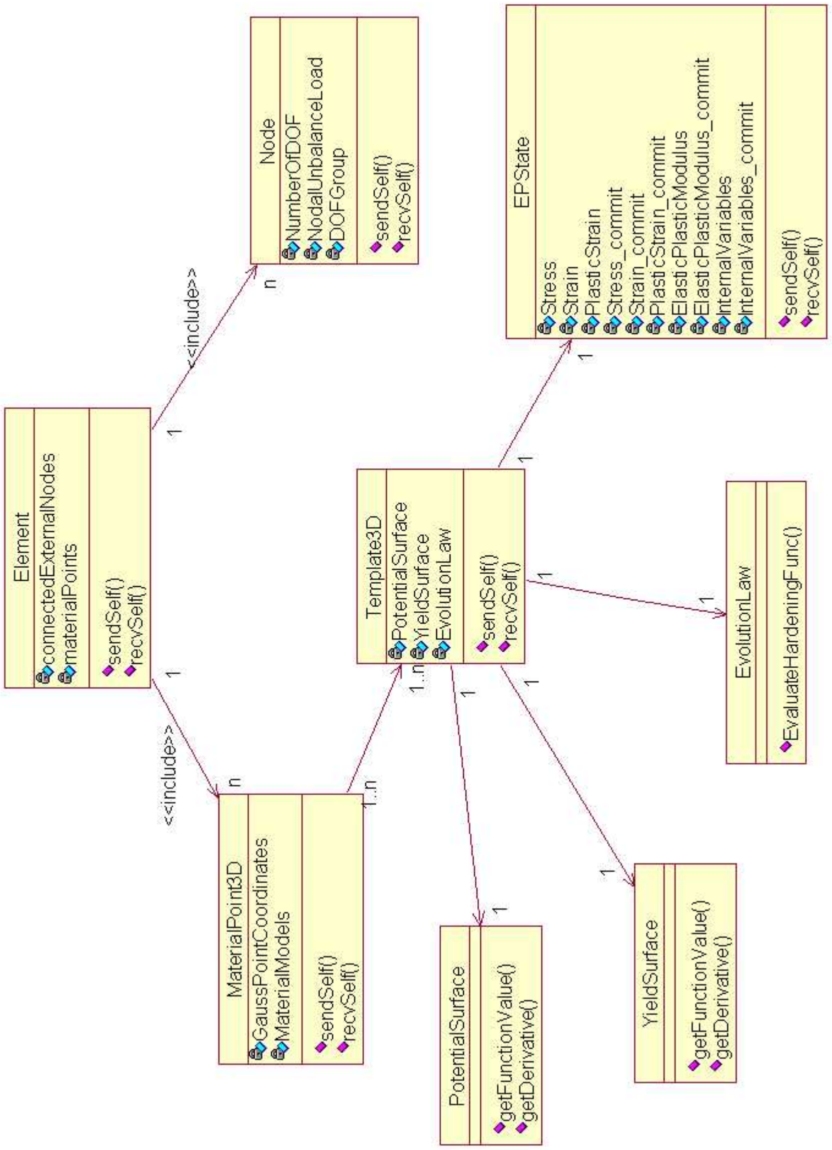


Figure 202.15: Class Diagram: Major Container Classes for Data Redistribution

All data communication operations have been implemented through the standard Send/RecvSelf interface, which forms a complete set of consistent point-to-point communication patterns and is convenient for future additions of new element/materials.

Chapter 203

Library Centric Software Platform Design

(1993-1994-1996-2005-2009-2010-2011-2019-)

203.1 Chapter Summary and Highlights

Veldhuizen (2005); Stroustrup (2005); Ramey (2005); Veldhuizen (2005);

203.1.1 Finite Elements

203.1.1.1 Single Phase Solid Elements

8 Node Brick

20 Node Brick

27 Node Brick

8-20 Node Brick

203.1.1.2 Fully Coupled, Two-Phase (Porous Solid – Pore Fluid) Solid Elements

u-p-U

u-p

203.1.1.3 Structural Elements

Truss

Beam

203.1.1.4 Special Elements

Contact Element

Seismic Isolator Element

203.1.2 Constitutive Integration and Material Models

203.1.2.1 Explicit Integration

203.1.2.2 Implicit Integration

203.1.2.3 Material Models

203.1.3 Modified OpenSees Services Library

PDD...

Chapter 204

Application Programming Interface

(2005-2009-2010-2011-2017-2019-)

(In collaboration with Dr. Nima Tafazzoli and Dr. Yuan Feng

204.1 Chapter Summary and Highlights

204.2 Introduction

Bloch (2005);

Dmitriev (2004); Stroustrup (2005); Niebler (2005); Mernik et al. (2005); Ward (2003);

204.3 Application Programming Interface for Domain Specific Language (DSL)

204.3.1 Modeling

Start new loading stage:

```
int start_new_stage(string CurrentStageName);
```

```
define_model_name(string theModelName)
```

```
obtain_pseudo_time()
```

```
wipe_model()
```

```
check_mesh(string outputfilename)
```

204.3.1.1 Modeling: Material Models

```
int add_constitutive_model_NDMaterial_linear_elastic_isotropic_3d(int MaterialNumber,
                double ElasticModulus,
                double nu,
                double rho)
```

MaterialNumber: Number of the predefined ND material to be used;

ElasticModulus: elastic modulus;

nu: Poisson's ratio;

rho: density;

```
add_constitutive_model_NDMaterial_linear_elastic_crossanisotropic(int MaterialNumber,
                double Ehp,
                double Evp,
                double nuhvp,
                double nuhhp,
                double Ghvp,
                double rhop)
```

`MaterialNumber`: Number of the ND material to be used ; `Ehp`: Elastic modulus on "horizontal" direction ; `Evp`: Elastic modulus on "vertical" direction ; `nuhvp`: Poisson ratio for "horizontal" - "vertical" direction ; `nuhhp`: Poisson ratio for "horizontal" - "horizontal" direction ; `Ghvp`: Shear moduls for "horizontal" - "vertical" direction ; `rhop`: density ;

```
add_constitutive_model_NDMaterial_vonmises_perfectly_plastic(int MaterialNumber,
                int Algorithm,
                double rho,
                double E,
                double v,
                double k,
```

```
                double initialconfiningstress,
                int number_of_subincrements,
                int maximum_number_of_iterations,
                double tolerance_1,
                double tolerance_2)
```

`MaterialNumber`: Number of the ND material to be used ; `Algorithm`: Explicit (=0) or Implicit (=1) ; `rho`: density ; `E`: Elastic modulus ; `v`: Poisson's ratio ; `k`: initial radius of von Mises cylinder ; `initialconfiningstress`: initial confining pressure (possitive for compression) ;

```
add_constitutive_model_NDMaterial_vonmises_isotropic_hardening(int MaterialNumber,
                int Algorithm,
                double rho,
                double E,
                double v,
                double k,
                double H,
                double initialconfiningstress,
                int number_of_subincrements,
                int maximum_number_of_iterations,
                double tolerance_1,
                double tolerance_2)
```

`MaterialNumber`: Number of the ND material to be used ; `Algorithm`: Explicit (=0) or Implicit (=1) ; `rho`: density ; `E`: Elastic modulus ; `v`: Poisson's ratio ; `k`: initial radius of von Mises cylinder ; `H`: rate of isotropic hardening ; `initialconfiningstress`: initial confining pressure (possitive for compression) ;

```
add_constitutive_model_NDMaterial_vonmises_kinematic_hardening(int MaterialNumber,
                int Algorithm,
                double rho,
                double E,
                double v,
```

```
                 double k,
                 double ha,
                 double Cr,
                 double initialconfiningstress,
                 int number_of_subincrements,
                 int maximum_number_of_iterations,
                 double tolerance_1,
                 double tolerance_2)
```

`MaterialNumber`: Number of the ND material to be used ; `Algorithm`: Explicit (=0) or Implicit (=1) ; `rho`: density ; `E`: Elastic modulus ; `v`: Poisson's ratio ; `k`: initial radius of von Mises cylinder ; `ha`: Armstrong-Frederick nonlinear kinematic hardening constant, initial slope ; `Cr`: Armstrong-Frederick nonlinear kinematic hardening constant, asymptote ; `initialconfiningstress`: initial confining pressure (possitive for compression) ;

```
add_constitutive_model_NDMaterial_vonmises_linear_kinematic_hardening(int MaterialNumber,
                 int Algorithm,
                 double rho,
                 double E,
                 double v,
                 double k,
                 double H,
                 double initialconfiningstress,
                 int number_of_subincrements,
                 int maximum_number_of_iterations,
                 double tolerance_1,
                 double tolerance_2)
```

`MaterialNumber`: Number of the ND material to be used ; `Algorithm`: Explicit (=0) or Implicit (=1) ; `rho`: density ; `E`: Elastic modulus ; `v`: Poisson's ratio ; `k`: initial radius of von Mises cylinder ; `H`: Kinematic hardening rate; `initialconfiningstress`: initial confining pressure (possitive for compression) ;

```
add_constitutive_model_NDMaterial_druckerprager_perfectly_plastic(int MaterialNumber,
                int Algorithm,
                double rho,
                double E,
                double v,
                double k,
                double initialconfiningstress,
                int number_of_subincrements,
                int maximum_number_of_iterations,
                double tolerance_1,
                double tolerance_2)
```

`MaterialNumber`: numer/Number of the nD material to be used ; `AlgorithmType`: Explicit (=0) or Implicit (=1) ; `rho`: density ; `E`: Elastic modulus ; `v`: Poisson's ratio ; `k`: initial equivalent friction angle ; `initialconfiningstress`: initial confining pressure (positive for compression) ;

```
add_constitutive_model_NDMaterial_druckerprager_isotropic_hardening(int MaterialNumber,
                int Algorithm,
                double rho,
                double E,
                double v,
                double k,
                double H,
                double initialconfiningstress,
                int number_of_subincrements,
                int maximum_number_of_iterations,
                double tolerance_1,
                double tolerance_2)
```

`MaterialNumber`: number/Number of the nD material to be used ; `AlgorithType`: Explicit (=0) or Implicit (=1) ; `rho`: density ; `E`: Elastic modulus ; `v`: Poisson's ratio ; `k`: initial equivalent friction angle ; `H`: rate of isotropic hardening ; `initialconfiningstress`: initial confining pressure (positive for compression) ;

```
add_constitutive_model_NDMaterial_druckerprager_kinematic_hardening(int MaterialNumber,
                int Algorithm,
                double rho,
                double E,
                double v,
                double k,
                double ha,
                double Cr,
                double initialconfiningstress,
                int number_of_subincrements,
                int maximum_number_of_iterations,
                double tolerance_1,
                double tolerance_2)
```

`MaterialNumber`: number/Number of the ND material to be used ; `Algorithm`: Explicit (=0) or Implicit (=1) ; `rho`: density ; `E`: Elastic modulus ; `v`: Poisson's ratio ; `k`: initial equivalent friction angle ; `ha`: Armstrong-Frederick nonlinear kinematic hardening constant, initial slope ; `Cr`: Armstrong-Frederick nonlinear kinematic hardening constant, asymptote ; `initialconfiningstress`: initial confining pressure (positive for compression) ;

```
add_constitutive_model_NDMaterial_camclay(int MaterialNumber,
                int Algorithm,
                double rho,
                double e0,
                double M,
                double lambda,
                double kappa,
                double v,
                double Kc,
                double P0,
                double initialconfiningstress,
```

```
                int number_of_subincrements,
                int maximum_number_of_iterations,
                double tolerance_1,
                double tolerance_2)
```

`MaterialNumber`: number/Number of the material to be used ; `AlgorithmType`: Explicit (=0) or Implicit (=1) ; `rho`: density ; `e0`: initial void ratio ; `M`: slope of the critical state line ; `lambda`: slope of the Normal Consolidation Line (NCL) ; `kappa`: slope of the Unloading-Reloading Line (URL) ; `v`: Poisson ratio ; `Kc`: Bulk modulus ; `initialconfiningstress`: initial confining stress/pressure (positive for compression) ;

```
add_constitutive_model_NDMaterial_sanisand_2004(int MaterialNumber,
                int Algorithm,
                double rho,
                double e0,
                double G0,
                double nu,
                double Pat,
                double p_cut,
                double Mc,
                double c,
                double lambda_c,
                double xi,
                double ec_ref,
                double m,
                double h0,
                double ch,
                double nb,
                double A0,
                double nd,
                double z_max,
                double cz,
                double initialconfiningstress,
```

```
                int number_of_subincrements,
                int maximum_number_of_iterations,
                double tolerance_1,
                double tolerance_2)
```

`MaterialNumber`: number/Number of the nD material to be used ; `AlgorithmType`: Explicit (=0) or Implicit (=1) ; `rho`: density ; `e0`: initial viod ratio ; `G0`: elastic shear modulus (same unit as stress) ; `nu`: Poisson's ratio ; `Pat`: atmospheric pressure ; `p_cut`: pressure cut-off ratio ; `Mc`: ; `c`: tension-compression strength ratio ; `lambda_c`: parameter for critical state line ; `xi`: parameter for critical state line ; `ec_ref`: reference void ratio for critical state line, ; $e_c = e_rlambda(p_c/P_{at})^xi$ `m`: opening of the yield surface ; `h0`: bounding parameter ; `ch`: bounding parameter ; `nb`: bounding parameter ; `A0`: dilatancy parameter ; `nd`: dilatancy parameter ; `z_max`: fabric parameter ; `cz`: fabric parameter ; `initialconfiningstress` initial confining pressure (positive for compression) ;

```
add_constitutive_model_NDMaterial_sanisand_2008(int MaterialNumber,
                int Algorithm,
                double rho,
                double e0,
                double G0,
                double K0,
                double Pat,
                double k_c,
                double alpha_cc,
                double c,
                double lambda,
                double xi,
                double ec_ref,
                double m,
                double h0,
                double ch,
                double nb,
                double A0,
                double nd,
```

```
                  double p_r,
                  double rho_c,
                  double theta_c,
                  double X,
                  double z_max,
                  double cz,
                  double p0,
                  double p_in,
                  int number_of_subincrements,
                  int maximum_number_of_iterations,
                  double tolerance_1,
                  double tolerance_2)
```

`MaterialNumber`: Number of the ND material to be used ; `Algorithm`: Explicit (=0) or Implicit (=1) ; `rho`: density ; `e0`: initial void ratio at zero strain ; `G0`: Reference elastic shear modulus (same unit as stress) ; `K0`: Reference elastic bulk modulus (same unit as stress) ; `Pat`: atmospherics pressure for critical state line ; `k_c`: cut-off factor; for $p < k_c P_{at}$, use $p = k_c P_{at}$ for calculation of G; (a default value of $k_c = 0.01$ should work fine) ; `alpha_cc`: critical state stress ratio ; `c`: tension-compression strength ratio ; `lambda`: parameter for critical state line ; `xi`: parameter for critical state line ; `ec_ref`: reference void for critical state line, ; $e_c = e_r lambda(p_c/P_{at})^x i$; `m`: opening of the yield surface ; `h0`: bounding parameter ; `ch`: bounding parameter ; `nb`: bounding parameter ; `A0`: dilatancy parameter ; `nd`: dilatancy parameter ; `p_r`: LCC parameter ; `rho_c`: LCC parameter ; `theta_c`: LCC parameter ; `X`: LCC parameter ; `z_max`: fabric parameter ; `cz`: fabric parameter ; `p0`: yield surface size ; `p_in` ;

```
add_constitutive_model_NDMaterial_pisano(int tag,
                                double E_in,
                                double v_in,
                                double M_in,
                                double kd_in,
                                double xi_in,
                                double h_in,
                                double m_in,
                                double rho_in,
```

```
                        double initialconfiningstress_in,
                        double beta_min_in)
```

```
add_constitutive_model_NDMaterial_accelerated_vonmises_perfectly_plastic(int MaterialNumber,
                  double rho,
                  double E,
                  double v,
                  double k,
                  double initialconfiningstress,
                  int maximum_number_of_iterations,
                  double tolerance_1,
                  double tolerance_2)
```

```
add_constitutive_model_NDMaterial_accelerated_vonmises_isotropic_hardening(int MaterialNumber,
                  double rho,
                  double E,
                  double v,
                  double k,
                  double H,
                  double initialconfiningstress,
                  int maximum_number_of_iterations,
                  double tolerance_1,
                  double tolerance_2)
```

```
add_constitutive_model_NDMaterial_accelerated_vonmises_kinematic_hardening(int MaterialNumber,
                  double rho,
                  double E,
                  double v,
                  double k,
                  double ha,
                  double Cr,
                  double initialconfiningstress,
```

```
                 int maximum_number_of_iterations,
                 double tolerance_1,
                 double tolerance_2)
```

```
add_constitutive_model_NDMaterial_accelerated_vonmises_linear_kinematic_hardening(int MaterialN
                 double rho,
                 double E,
                 double v,
                 double k,
                 double H,
                 double initialconfiningstress,
                 int maximum_number_of_iterations,
                 double tolerance_1,
                 double tolerance_2)
```

```
add_constitutive_model_NDMaterial_accelerated_druckerprager_perfectly_plastic(int MaterialNumbe
                 double rho,
                 double E,
                 double v,
                 double k,
                 double initialconfiningstress,
                 int maximum_number_of_iterations,
                 double tolerance_1,
                 double tolerance_2)
```

```
add_constitutive_model_NDMaterial_accelerated_druckerprager_isotropic_hardening(int MaterialNum
                 double rho,
                 double E,
                 double v,
```

```
                 double k,
                 double H,
                 double initialconfiningstress,
                 int maximum_number_of_iterations,
                 double tolerance_1,
                 double tolerance_2)
```

```
add_constitutive_model_NDMaterial_accelerated_druckerprager_kinematic_hardening(int MaterialNum
                 double rho,
                 double E,
                 double v,
                 double k,
                 double ha,
                 double Cr,
                 double initialconfiningstress,
                 int maximum_number_of_iterations,
                 double tolerance_1,
                 double tolerance_2)
```

```
add_constitutive_model_NDMaterial_accelerated_camclay(int MaterialNumber,
            double rho,
            double e0,
            double M,
            double lambda,
            double kappa,
            double v,
            double Kc,
            double p0,
            double initialconfiningstress,
            int maximum_number_of_iterations,
            double tolerance_1,
            double tolerance_2)
```

```
add_constitutive_model_uniaxial_elastic(int MaterialNumber,
                double elasticmodulus,
                double eta)
```

`MaterialNumber`: unique material Number ; `elasticmodulus`: elastic modulus of the material ; `eta`: damping tangent ;

```
add_constitutive_model_uniaxial_concrete02(int MaterialNumber,
  double fpc, double epsc0, double fpcu,
  double epscu, double rat, double ft,
  double Ets)
```

`MaterialNumber`: unique material Number ; `fpc`: compressive strength ; `epsc0`: strain at compressive strength ; `fpcu`: crushing strength ; `epsU`: strain at crushing strength ; `lambda`: ratio between unloading slope at epscu and initial slope ; `ft`: tensile strength ; `Ets`: tension softening stiffness (absolute value) (slope of the linear tension softening branch) ;

```
int add_constitutive_model_uniaxial_steel01(int MaterialNumber,
                double fy,
                double Ep,
                double Hd,
                int a1,
                int a2,
                int a3,
                int a4)
```

`MaterialNumber`: unique material Number ; `fy`: yield strength ; `Ep`: initial elastic tangent ; `Hd`: strain-hardening ratio (ratio between post-yield tangent and initial elastic tangent) ; `a1, a2, a3, a4`: isotropic hardening parameters ; `a1`: isotropic hardening parameter, increase of compression yield envelope as proportion of yield strength after a plastic strain of a2*(fy/Ep). ; `a2`: isotropic hardening parameter (see explanation under a1) ; `a3`: isotropic hardening parameter, increase of tension yield envelope as proportion of yield strength after a plastic strain of a4*(fy/Ep) ; `a4`: isotropic hardening parameter (see explanation under a3) ;

```
int add_constitutive_model_uniaxial_steel02(int MaterialNumber,
  double fy, double E0, double b,
  double R0, double cR1, double cR2,
  double a1, double a2, double a3, double a4)
```

`MaterialNumber`: unique material object Number ; `fy`: yield strength ; `E0`: initial elastic tangent ; `b`: strain-hardening ratio (ratio between post-yield tangent and initial elastic tangent) ; `R0, cR1, cR2`: control the transition from elastic to plastic branches. Recommended values: R0=between 10 and 20, cR1=0.925, cR2=0.15 ; `a1, a2, a3, a4`: isotropic hardening parameters ; `a1`: isotropic hardening parameter, increase of compression yield envelope as proportion of yield strength after a plastic strain of a2*(Fy/E). ; `a2`: isotropic hardening parameter (see explanation under a1) ; `a3`: isotropic hardening parameter, increase of tension yield envelope as proportion of yield strength after a plastic strain of a4*(Fy/E) ; `a4`: isotropic hardening parameter (see explanation under a3) ;

204.3.1.2 Modeling: Nodes

```
int add_node(int NodeNumber,
             int number_of_DOFs,
             double coordinate_x,
             double coordinate_y,
             double coordinate_z)
```

`NodeNumber`: integer Number identifying node ; `number_of_DOFs`: number of degrees of freedom for node ; `coordinate_x`: x coordinate of the node ; `coordinate_y`: y coordinate of the node ; `coordinate_z`: z coordinate of the node ;

```
int remove_node(int NodeNumber)
```

`NodeNumber`: integer Number identifying the node to be removed ;

```
int add_mass_to_node(int NodeNumber,
                     double massvalue1,
                     double massvalue2,
                     double massvalue3)

int add_mass_to_node(int NodeNumber,
                  double massvalue1,
                  double massvalue2,
                  double massvalue3,
                  double massvalue4,
                  double massvalue5,
                  double massvalue6)

int add_mass_to_node(int NodeNumber,
                  double massvalue1,
                  double massvalue2,
                  double massvalue3,
                  double massvalue4,
                  double massvalue5,
                  double massvalue6,
                  double massvalue7)
```

`NodeNumber`: integer Number of the node that mass would be applied to ; `massvalue(#)`: the amount of mass assigned to each degree of freedom ;

204.3.1.3 Modeling: Finite Elements

```
add_element_truss(int ElementNumber,
                int iNode,
                int jNode,
                int MaterialNumber,
                double sectionarea,
```

```
                double rho)
```

`ElementNumber`: unique element object Number ; `dimension`: number of dimensions of the beam ; `iNode` , `jNode`: end nodes ; `MaterialNumber`: Number of the uniaxial material to be used ; `sectionarea`: section area of the truss element ; `rho`: density ;

```
add_element_beam_elastic(int ElementNumber,
                double A,
                double E,
                double G,
                double Jx,
                double Iy,
                double Iz,
                int iNode,
                int jNode,
                double rho,
                double vecxzPlane_X, double vecxzPlane_Y, double vecxzPlane_Z,
                double jntOffsetI_X, double jntOffsetI_Y, double jntOffsetI_Z,
                double jntOffsetJ_X, double jntOffsetJ_Y, double jntOffsetJ_Z)
```

`ElementNumber`: unique element object Number ; `A`: section area ; `E`: Young's modulus ; `G`: Shear Modulus ; `Jx`: polar moment of inertia ; `Iy`: moment of inertia around y ; `Iz`: moment of inertia around z ; `iNode` , `jNode`: end nodes ; `TransformationNumber`: identifier for previously-defined coordinate-transformation (CrdTransf) object ; `rho`: density ; `sectionTag`: identifier for previously-defined section object ;

```
add_element_beam_elastic_lumped_mass(int ElementNumber,
                double A,
                double E,
                double G,
                double Jx,
                double Iy,
```

```
                  double Iz,
                  int iNode,
                  int jNode,
                  double rho,
                  double vecxzPlane_X, double vecxzPlane_Y, double vecxzPlane_Z,
                  double jntOffsetI_X, double jntOffsetI_Y, double jntOffsetI_Z,
                  double jntOffsetJ_X, double jntOffsetJ_Y, double jntOffsetJ_Z)
```

`ElementNumber`: unique element object Number ; `A`: section area ; `E`: Young's modulus ; `G`: Shear Modulus ; `Jx`: polar moment of inertia ; `Iy`: moment of inertia around y ; `Iz`: moment of inertia around z ; `iNode` , `jNode`: end nodes ; `TransformationNumber`: identifier for previously-defined coordinate-transformation (CrdTransf) object ; `rho`: density ; `sectionTag`: identifier for previously-defined section object ;

```
add_element_brick_8node(int ElementNumber,
                  int node_numb_1,
                  int node_numb_2,
                  int node_numb_3,
                  int node_numb_4,
                  int node_numb_5,
                  int node_numb_6,
                  int node_numb_7,
                  int node_numb_8,
                  int MaterialNumber)
```

`elementTag`: unique element object Number ; `node_numb_#`: eight node numbers specified in appropriate order ; `materialTag`: material Number associated with previsouly-defined NDMaterial object ;

```
add_element_brick_8node_elastic(int ElementNumber,
                  int node_numb_1,
                  int node_numb_2,
```

```
                 int node_numb_3,
                 int node_numb_4,
                 int node_numb_5,
                 int node_numb_6,
                 int node_numb_7,
                 int node_numb_8,
                 int MaterialNumber)
```

`elementTag`: unique element object Number ; `node_numb_#`: eight node numbers specified in appropriate order ; `materialTag`: material Number associated with previsouly-defined NDMaterial object ;

```
add_element_brick_20node(int ElementNumber,
                         int node_numb_1,
                         int node_numb_2,
                         int node_numb_3,
                         int node_numb_4,
                         int node_numb_5,
                         int node_numb_6,
                         int node_numb_7,
                         int node_numb_8,
                         int node_numb_9,
                         int node_numb_10,
                         int node_numb_11,
                         int node_numb_12,
                         int node_numb_13,
                         int node_numb_14,
                         int node_numb_15,
                         int node_numb_16,
                         int node_numb_17,
                         int node_numb_18,
                         int node_numb_19,
                         int node_numb_20,
```

```
                                    int MaterialNumber)
```

`ElementNumber`: unique element object Number ; `node_numb_#`: eight node numbers specified in appropriate order ; `MaterialNumber`: material Number associated with previsouly-difined NDMaterial object ;

```
add_element_brick_20node_elastic(int ElementNumber,
                                 int node_numb_1,
                                 int node_numb_2,
                                 int node_numb_3,
                                 int node_numb_4,
                                 int node_numb_5,
                                 int node_numb_6,
                                 int node_numb_7,
                                 int node_numb_8,
                                 int node_numb_9,
                                 int node_numb_10,
                                 int node_numb_11,
                                 int node_numb_12,
                                 int node_numb_13,
                                 int node_numb_14,
                                 int node_numb_15,
                                 int node_numb_16,
                                 int node_numb_17,
                                 int node_numb_18,
                                 int node_numb_19,
                                 int node_numb_20,
                                 int MaterialNumber)
```

`ElementNumber`: unique element object Number ; `node_numb_#`: eight node numbers specified in appropriate order ; `MaterialNumber`: material Number associated with previsouly-difined NDMaterial object ;

```
int add_element_brick_27node(int ElementNumber,
                             int node_numb_1,
                             int node_numb_2,
                             int node_numb_3,
                             int node_numb_4,
                             int node_numb_5,
                             int node_numb_6,
                             int node_numb_7,
                             int node_numb_8,
                             int node_numb_9,
                             int node_numb_10,
                             int node_numb_11,
                             int node_numb_12,
                             int node_numb_13,
                             int node_numb_14,
                             int node_numb_15,
                             int node_numb_16,
                             int node_numb_17,
                             int node_numb_18,
                             int node_numb_19,
                             int node_numb_20,
                             int node_numb_21,
                             int node_numb_22,
                             int node_numb_23,
                             int node_numb_24,
                             int node_numb_25,
                             int node_numb_26,
                             int node_numb_27,
                             int MaterialNumber)
```

`ElementNumber`: unique element object Number ; `node_numb_#`: eight node numbers specified in appropriate order ; `MaterialNumber`: material Number associated with previsouly-difined NDMaterial object ;

```
int add_element_brick_27node_elastic(int ElementNumber,
                                     int node_numb_1,
                                     int node_numb_2,
                                     int node_numb_3,
                                     int node_numb_4,
                                     int node_numb_5,
                                     int node_numb_6,
                                     int node_numb_7,
                                     int node_numb_8,
                                     int node_numb_9,
                                     int node_numb_10,
                                     int node_numb_11,
                                     int node_numb_12,
                                     int node_numb_13,
                                     int node_numb_14,
                                     int node_numb_15,
                                     int node_numb_16,
                                     int node_numb_17,
                                     int node_numb_18,
                                     int node_numb_19,
                                     int node_numb_20,
                                     int node_numb_21,
                                     int node_numb_22,
                                     int node_numb_23,
                                     int node_numb_24,
                                     int node_numb_25,
                                     int node_numb_26,
                                     int node_numb_27,
                                     int MaterialNumber)
```

`ElementNumber`: unique element object Number ; `node_numb_#`: eight node numbers specified in appropriate order ; `MaterialNumber`: material Number associated with previsouly-difined NDMaterial object ;

```
add_element_brick_8node_up(int ElementNumber,
                 int node_numb_1,
                 int node_numb_2,
                 int node_numb_3,
                 int node_numb_4,
                 int node_numb_5,
                 int node_numb_6,
                 int node_numb_7,
                 int node_numb_8,
                 int MaterialNumber,
                 double porosity,
                 double alpha,
                 double rho_s,
                 double rho_f,
                 double k_x,
                 double k_y,
                 double k_z,
                 double K_s,
                 double K_f)
```

```
add_element_brick_8node_upU(int ElementNumber,
                 int node_numb_1,
                 int node_numb_2,
                 int node_numb_3,
                 int node_numb_4,
                 int node_numb_5,
                 int node_numb_6,
                 int node_numb_7,
                 int node_numb_8,
                 int MaterialNumber,
                 double porosity,
                 double alpha,
                 double rho_s,
```

```
                 double rho_f,
                 double k_x,
                 double k_y,
                 double k_z,
                 double K_s,
                 double K_f)
```

```
add_element_brick_20node_upU(int ElementNumber,
                                  int node_numb_1,
                                  int node_numb_2,
                                  int node_numb_3,
                                  int node_numb_4,
                                  int node_numb_5,
                                  int node_numb_6,
                                  int node_numb_7,
                                  int node_numb_8,
                                  int node_numb_9,
                                  int node_numb_10,
                                  int node_numb_11,
                                  int node_numb_12,
                                  int node_numb_13,
                                  int node_numb_14,
                                  int node_numb_15,
                                  int node_numb_16,
                                  int node_numb_17,
                                  int node_numb_18,
                                  int node_numb_19,
                                  int node_numb_20,
                                  int MaterialNumber,
                                  double porosity,
                                  double alpha,
                                  double rho_s,
                                  double rho_f,
```

```
                                  double k_x,
                                  double k_y,
                                  double k_z,
                                  double K_s,
                                  double K_f)
```

```
add_element_brick_8node_variable_number_of_gauss_points(int ElementNumber,
                 int number_of_gauss_points,
                 int node_numb_1,
                 int node_numb_2,
                 int node_numb_3,
                 int node_numb_4,
                 int node_numb_5,
                 int node_numb_6,
                 int node_numb_7,
                 int node_numb_8,
                 int MaterialNumber)
```

```
add_element_brick_20node_variable_number_of_gauss_points(int ElementNumber,
                              int number_of_gauss_points,
                              int node_numb_1,
                              int node_numb_2,
                              int node_numb_3,
                              int node_numb_4,
                              int node_numb_5,
                              int node_numb_6,
                              int node_numb_7,
                              int node_numb_8,
                              int node_numb_9,
                              int node_numb_10,
                              int node_numb_11,
                              int node_numb_12,
```

```
                                          int node_numb_13,
                                          int node_numb_14,
                                          int node_numb_15,
                                          int node_numb_16,
                                          int node_numb_17,
                                          int node_numb_18,
                                          int node_numb_19,
                                          int node_numb_20,
                                          int MaterialNumber)
```

```
add_element_brick_27node_variable_number_of_gauss_points(int ElementNumber,
                                         int number_of_gauss_points,
                                         int node_numb_1,
                                         int node_numb_2,
                                         int node_numb_3,
                                         int node_numb_4,
                                         int node_numb_5,
                                         int node_numb_6,
                                         int node_numb_7,
                                         int node_numb_8,
                                         int node_numb_9,
                                         int node_numb_10,
                                         int node_numb_11,
                                         int node_numb_12,
                                         int node_numb_13,
                                         int node_numb_14,
                                         int node_numb_15,
                                         int node_numb_16,
                                         int node_numb_17,
                                         int node_numb_18,
                                         int node_numb_19,
                                         int node_numb_20,
                                         int node_numb_21,
```

```
                             int node_numb_22,
                             int node_numb_23,
                             int node_numb_24,
                             int node_numb_25,
                             int node_numb_26,
                             int node_numb_27,
                             int MaterialNumber)
```

```
add_element_brick_variable_node_8_to_27(int ElementNumber, int number_of_gauss_points,
                                            int node_numb_1,  int node_numb_2,  int node_numb_3
                                            int node_numb_4,  int node_numb_5,  int node_numb_6
                                            int node_numb_7,  int node_numb_8,  int node_numb_9
                                            int node_numb_10, int node_numb_11, int node_numb_1
                                            int node_numb_13, int node_numb_14, int node_numb_1
                                            int node_numb_16, int node_numb_17, int node_numb_1
                                            int node_numb_19, int node_numb_20, int node_numb_2
                                            int node_numb_22, int node_numb_23, int node_numb_2
                                            int node_numb_25, int node_numb_26, int node_numb_2
                                            int MaterialNumber)
```

```
add_element_contact_3dof_to_3dof(int ElementNumber,
                 int iNode,
                 int jNode,
                 double Knormal,
                 double Ktangent,
                 double frictionRatio,
                 double x_local_1,
                 double x_local_2,
                 double x_local_3)
```

```
add_element_contact_nonlinear_3dof_to_3dof(int ElementNumber,
                 int iNode,
```

```
                 int jNode,
                 double Knormal,
                 double Ktangent,
                 double frictionRatio,
   double maxmimum_gap,
   double maximum_normal_stiffness,
                 double x_local_1,
                 double x_local_2,
                 double x_local_3)
```

```
add_element_contact_nonlinear_3dof_to_7dof(int ElementNumber,
                 int iNode,
                 int jNode,
                 double Knormal,
                 double Ktangent,
                 double frictionRatio,
                 double maximum_gap,
                 double maximum_normal_stiffness,
                 double x_local_1,
                 double x_local_2,
                 double x_local_3)
```

```
add_element_shell_andes_3node(int ElementNumber,
                            int node_numb_1,
                            int node_numb_2,
                            int node_numb_3,
                            double thickness,
                            int MaterialNumber)
```

```
add_element_shell_andes_4node(int ElementNumber,
                            int node_numb_1,
                            int node_numb_2,
```

```
                            int node_numb_3,
                            int node_numb_4,
                            double thickness,
                            int MaterialNumber)
```

```
add_element_shell_MITC4(int ElementNumber,
                                int node_numb_1,
                                int node_numb_2,
                                int node_numb_3,
                                int node_numb_4,
                                double thickness,
                                int MaterialNumber)
```

```
add_element_shell_NewMITC4(int ElementNumber,
                                int node_numb_1,
                                int node_numb_2,
                                int node_numb_3,
                                int node_numb_4,
                                double thickness,
                                int MaterialNumber)
```

```
add_element_penalty(int ElementNumber,
int node1,
int node2,
double penalty_stiffness,
int dof)
```

```
add_element_penalty_for_applying_generalized_displacement(int ElementNumber,
                        int Exist_Node,
                        double penalty_stiffness,
                        int direction)
```

```
add_element_rank_one_deficient_elastic_pinned_fixed_beam(int ElementNumber,
                 double A,
                 double E,
                 double G,
                 double Jx,
                 double Iy,
                 double Iz,
                 int iNode,
                 int jNode,
                 double rho,
                 double vecxzPlane_X, double vecxzPlane_Y, double vecxzPlane_Z,
                 double jntOffsetI_X, double jntOffsetI_Y, double jntOffsetI_Z,
                 double jntOffsetJ_X, double jntOffsetJ_Y, double jntOffsetJ_Z)
```

```
add_element_beam_displacement_based(int ElementNumber,
                 int iNode,
                 int jNode,
                 int numberofintegrationpoints,
                 int SectionNumber,
                 double rho,
                 string integrationrule,
                 double vecInLocXZPlane_x, double vecInLocXZPlane_y, double vecInLocXZPlane_z,
                 double rigJntOffset1_x, double rigJntOffset1_y, double rigJntOffset1_z,
                 double rigJntOffset2_x, double rigJntOffset2_y, double rigJntOffset2_z)
```

```
int remove_element(int ElementNumber)
```

`ElementNumber`: number identifying the element to be removed ;

204.3.1.4 Modeling: Damping

```
int add_damping_rayleigh(int dampingNumber,
                         double a0,
                         double a1,
                         string which_stiffness_to_use)
```

`dampingNumber`: damping Number number to be used in element definition ; `a0, a1`: Rayleigh order damping coefficients ; `which_stiffness_to_use`: `Initial_Stiffness/Current_Stiffness/Last_Commited_Stif`

```
add_damping_caughey3rd(int dampingNumber, double a0, double a1, double a2, string which_stiffne
```

```
add_damping_caughey4th(int dampingNumber, double a0, double a1, double a2, double a3, string wh
```

```
int add_damping_to_element(int elementNumber,
                           int dampingNumber)
```

`dampingNumber`: damping number to be assigned to element ; `elementNumber`: element number which damping is going to be assigned to ;

```
int add_damping_to_node(int nodeNumber,
               int dampingNumber)
```

`dampingNumber`: damping number to be assigned to node (note that only the mass proportional coefficient will be used for node) ; `nodeNumber`: node number which damping is going to be assigned to ;

204.3.1.5 Modeling: Constraints, Supports, Tied Nodes Connections, etc.

```
int add_support_to_all_dofs_of_node(int NodeNumber)
```

`NodeNumber`: integer Number of the node to be fixed ;

```
  int add_support_to_node(int NodeNumber,
                          int dof_number)
```

`NodeNumber`: integer Number identifying the node to be constrained ; `dof_number`: dof to be fixed ;

```
int add_equaldof_to_two_nodes(int nodeRetain,
                    int nodeConstr,
                    int dofID1,
                    int dofID2,
                    ...,
                    int dofID7)
```

`nodeRetain`: integer Number identifying the retained, or master node (rNode) ; `nodeConstr`: integer Number identifying the constrained, or slave node (cNode) ; `dofID`: nodal degrees-of-freedom that are constrained at the nodeConstr to be the same as those at the nodeRetain Valid range is from 1 to 7. ;

```
int remove_support_from_node_by_fixity_number(int FixityNumber)
```

`FixityNumber`: integer Number identifying the fixity to be removed ;

```
remove_support_from_node(int NodeNumber, int dofNumber)
```

`NodeNumber`: integer Number identifying the node number ; `dofNumber`: integer Number identifying the dof number ;

```
  int remove_equaldof_from_node(int NodeNumber)
```

`FixityNumber`: integer Number identifying the fixity to be removed ;

204.3.1.6 Modeling: Static Loads

```
add_force_time_history_linear(int PatternNumber,
                                                int NodeNumber,
                                                int dof_to_be_shaken,
                                                double final_load_value)
```

```
add_force_time_history_path_series(int PatternNumber,
                                                int NodeNumber,
                                                int dof_to_be_shaken,
                                                double TimeIncrement,
                                                double LoadFactor,
                                                string Forceinputfilename)
```

```
add_force_time_history_path_time_series(int PatternNumber,
                                                int NodeNumber,
                                                int dof_to_be_shaken,
                                                double LoadFactor,
                                                string Forceinputfilename)
```

```
add_load_selfweight_to_element(int SelfWeightNumber,
                     int ElementNumber, int AccelerationFieldNumber)
```

```
add_acceleration_field(int GravityFieldNumber,
                 double accelerationfield_x,
                 double accelerationfield_y,
                 double accelerationfield_z)
```

```
add_load_constant_normal_pressure_to_8node_brick_surface(int SurfaceLoadNumber,
                           int ElementNumber,
                           int Node_1,
                           int Node_2,
                           int Node_3,
                           int Node_4,
                           double SurfaceLoadMagnitude)
```

```
add_load_different_normal_pressure_to_8node_brick_surface(int SurfaceLoadNumber,
                           int ElementNumber,
                           int Node_1,
                           int Node_2,
                           int Node_3,
                           int Node_4,
                           double SurfaceLoadMagnitude1,
                           double SurfaceLoadMagnitude2,
                           double SurfaceLoadMagnitude3,
                           double SurfaceLoadMagnitude4)
```

```
add_load_constant_normal_pressure_to_20node_brick_surface(int SurfaceLoadNumber,
                           int ElementNumber,
                           int Node_1,
                           int Node_2,
                           int Node_3,
                           int Node_4,
                           int Node_5,
                           int Node_6,
                           int Node_7,
                           int Node_8,
                           double SurfaceLoadMagnitude)
```

```
add_load_different_normal_pressure_to_20node_brick_surface(int SurfaceLoadNumber,
                           int ElementNumber,
                           int Node_1,
                           int Node_2,
                           int Node_3,
                           int Node_4,
                           int Node_5,
                           int Node_6,
                           int Node_7,
                           int Node_8,
                           double SurfaceLoadMagnitude1,
                           double SurfaceLoadMagnitude2,
                           double SurfaceLoadMagnitude3,
                           double SurfaceLoadMagnitude4,
                           double SurfaceLoadMagnitude5,
                           double SurfaceLoadMagnitude6,
                           double SurfaceLoadMagnitude7,
                           double SurfaceLoadMagnitude8)
```

```
add_load_constant_normal_pressure_to_27node_brick_surface(int SurfaceLoadNumber,
                           int ElementNumber,
                           int Node_1,
                           int Node_2,
                           int Node_3,
                           int Node_4,
                           int Node_5,
                           int Node_6,
                           int Node_7,
                           int Node_8,
                           int Node_9,
                           double SurfaceLoadMagnitude)
```

```
add_load_different_normal_pressure_to_27node_brick_surface(int SurfaceLoadNumber,
```

```
                                  int ElementNumber,
                                  int Node_1,
                                  int Node_2,
                                  int Node_3,
                                  int Node_4,
                                  int Node_5,
                                  int Node_6,
                                  int Node_7,
                                  int Node_8,
                                  int Node_9,
                                  double SurfaceLoadMagnitude1,
                                  double SurfaceLoadMagnitude2,
                                  double SurfaceLoadMagnitude3,
                                  double SurfaceLoadMagnitude4,
                                  double SurfaceLoadMagnitude5,
                                  double SurfaceLoadMagnitude6,
                                  double SurfaceLoadMagnitude7,
                                  double SurfaceLoadMagnitude8,
                                  double SurfaceLoadMagnitude9)
```

```
add_penalty_displacement_time_history_linear(int PatternNumber,
                                             int PenaltyElementNumber,
                                             int dof_to_be_shaken,
                                             double Final_Displacement_Value)
```

```
add_penalty_displacement_time_history_path_series(int PatternNumber,
                                             int PenaltyElementNumber,
                                             int dof_to_be_shaken,
                                             double TimeIncrement,
                                             double LoadFactor,
                                             string Displacementinputfilename)
```

```
add_single_point_constraint_to_node(int NodeNumber,
                                      int dof_number,
                                      double DOFvalue)
```

204.3.1.7 Modeling: Dynamic Loads

`add_load_pattern_domain_reduction_method`

```
add_load_pattern_domain_reduction_method(int PatternNumber,
                                              double dt,
                                              double factor,
                                              int numberofsteps,
                                              int numberofdrmnodes,
                                              int numberofdrmelements,
                                              double xpositive,
                                              double xminus,
                                              double ypositive,
                                              double yminus,
                                              double zpositive,
                                              double zminus,
                                              string ElementNumbersFilename,
                                              string NodeNumbersFilename,
                                              string DisplacementTimeHistoryFilename,
                                              string AccelerationTimeHistoryFilename)
```

Inputs: `PatternNumber`: number assigned to DRM load pettern ; `dt`: time interval of input files for time histories ; `factor`: factor to multiply to the input time history ; `numberofsteps`: Number of the time steps in acceleration/displacement time history ; `numberofdrmnodes`: Number of the nodes in DRM layer ; `numberofdrmelements`: Number of the elements in DRM layer ; `xpositive, xminus`: boundary layer range in x direction ; `ypositive, yminus`: boundary layer range in y direction ; `zpositive, zminus`: boundary layer range in z direction ; `ElementNumbersFilename`:

File including element numbers inside the platic bowl (1 element number per line in the input file) ; `NodeNumbersFilename`: File including node numbers inside the platic bowl (1 node number per line in the input file) ; `DisplacementTimeHistoryFilename`: File including displacement time history (in each line write the values of displacement in time for first degree of freedom of the first node defined in NodeNumbersFilename, next line should have the values for second dof of the first node and continue for all degrees of freedom. Then move to the second node defined in NodeNumbersFilename and ... ; `AccelerationTimeHistoryFilename`: File including acceleration time history (in each line write the values of displacement in time for first degree of freedom of the first node defined in NodeNumbersFilename, next line should have the values for second dof of the first node and continue for all degrees of freedom. Then move to the second node defined in NodeNumbersFilename and ... ;

```
  remove_load(int LoadPatternNumber)
```

```
add_load_pattern_domain_reduction_method_save_forces(int PatternNumber,
                                                     double dt,
                                                     double factor,
                                                     int numberofsteps,
                                                     int numberofdrmnodes,
                                                     int numberofdrmelements,
                                                     double xpositive,
                                                     double xminus,
                                                     double ypositive,
                                                     double yminus,
                                                     double zpositive,
                                                     double zminus,
                                                     string ElementNumbersFilename,
                                                     string NodeNumbersFilename,
                                                     string DisplacementTimeHistoryFilename,
                                                     string AccelerationTimeHistoryFilename,
                                                     string ForceTimeHistoryFilename)
```

```
add_load_pattern_domain_reduction_method_restore_forces(int PatternNumber,
```

```
                                                      double dt,
                                                      double factor,
                                                      int numberofsteps,
                                                      int numberofdrmnodes,
                                                      int numberofdrmelements,
                                                      double xpositive,
                                                      double xminus,
                                                      double ypositive,
                                                      double yminus,
                                                      double zpositive,
                                                      double zminus,
                                                      string ElementNumbersFilename,
                                                      string NodeNumbersFilename,
                                                      string ForceTimeHistoryFilename)
```

204.3.1.8 Modeling: Prescribed Displacements

```
add_imposed_motion(int GroundMotionNumber,
               int NodeNumber,
               int degree_of_freedom,
               double timestep,
               double displacement_scale,
               string displacementfilename,
               double velocity_scale,
               string velocityfilename,
               double acceleration_scale,
               string accelerationfilename)
```

204.3.1.9 Solid-Fluid Interface

Two new APIs and corresponding DSL commands have been added to RealESSI.

- define solid fluid interface “interface_name”
  This API is used to define solid fluid interface. Passing parameters into RealESSI to initialize our interface class SSFI.

- simulate No. steps using solid fluid interaction transient algorithm time_step = < time >
  This API aims to launch solid fluid transient interaction analysis. The time step defined here refers to the time step for the transient analysis in solid domain. The time step for fluid domain can be different and defined in the input files for OpenFOAM.

204.3.1.10 Outputs to mySQL database

```
restore_response_of_model_mysql_format(int Node_Number, int DOF_Number, int Step_Number,
                                       string databaseName, string host,
                                       string username, string password, unsigned int port,
                                       string socket)
```

```
restore_state_of_model_mysql_format(string databaseName, string host,
                                    string username, string password, unsigned int port,
                                    string socket)
```

```
save_response_of_model_mysql_format(string databaseName, string host,
                                    string username, string password, unsigned int port,
                                    string socket)
```

```
save_state_of_model_mysql_format(string databaseName, string host,
                                 string username, string password, unsigned int port,
                                 string socket)
```

204.3.2 Simulation

204.3.2.1 Simulation: Solvers

Definition of system of equation (linear) solvers to be used.

```
 int define_solver_profilespd_for_analysis()
```

```
int define_solver_umfpack_for_analysis()
```

```
int define_solver_petsc_for_analysis()
```

204.3.2.2 Simulation: Static Solution Advancement

Definition of static solution advancement algorithms (see more in section 107.6 on page 526).

```
int define_static_solution_advancement_integrator_displacement_control(int node_number,
         int doftomove,
         double dispincrement)
```

`dispincrement`: increment of displacement in each step of analysis ; `node_number`: node whose response controls the solution ; `doftomove`: degree-of-freedom whose response controls the solution. Valid range is from 1 through the number of nodal degrees-of-freedom. ;

```
int define_static_solution_advancement_integrator_load_control(double loadstep)
```

`loadstep`: load step size ;

204.3.2.3 Simulation: Dynamic Solution Advancement

Definition of dynamic, time integration/advancement algorithms (see more in section 108.3 on page 538).

```
int define_dynamic_solution_advancement_integrator_hilber_hughes_taylor(double HHT_Alpha)
```

`HHT_Alpha`: HHT α parameter ;

```
int define_dynamic_solution_advancement_integrator_newmark(double gamma, double beta)
```

`newmark_gamma, newmark_beta`: Newmark γ and β parameters ;

204.3.2.4 Simulation: Solution Algorithms

Definition of solution algorithms to be used:

```
int define_algorithm_with_no_convergence_check_for_analysis()
```

```
int define_algorithm_newton_for_analysis()
```

```
int define_algorithm_modifiednewton_for_analysis()
```

204.3.2.5 Simulation: Convergence Criteria

```
int define_convergence_test_energyincrement_for_analysis(double theTol,
                      int maxIter,
                      int PrintFlag)
```

`theTol`: convergence tolerance ; `maxIter`: maximum number of iterations that will be performed before "failure to converge" is returned ; `PrintFlag`: flag used to print information on convergence (optional) : 0: no print output ; 1: print information on each step ; 2: print information when convergence has been achieved ; 4: print norm, dU and dR vectors ; 5: at convergence failure, carry on, print error message, but do not stop analysis ;

```
int define_convergence_test_normdisplacementincrement_for_analysis(double theTol,
              int maxIter,
              int PrintFlag)
```

`theTol`: convergence tolerance ; `maxIter`: maximum number of iterations that will be performed before "failure to converge" is returned ; `PrintFlag`: flag used to print information on convergence (optional) ; 0: no print output ; 1: print information on each step ; 2: print information when convergence has been achieved ; 4: print norm, dU and dR vectors ; 5: at convergence failure, carry on, print error message, but do not stop analysis ;

```
int define_convergence_test_normunbalance_for_analysis(double theTol,
             int maxIter,
             int PrintFlag)
```

`theTol`: convergence tolerance ; `maxIter`: maximum number of iterations that will be performed before "failure to converge" is returned ; `PrintFlag`: flag used to print information on convergence (optional) ; 0: no print output ; 1: print information on each step ; 2: print information when convergence has been achieved ; 4: print norm, dU and dR vectors ; 5: at convergence failure, carry on, print error message, but do not stop analysis ;

204.3.2.6 Simulating Response

```
int simulate_using_static_multistep(int numSteps)
```

`numSteps`: number of static analysis steps which will advance the solution, ;

```
 int simulate_using_static_onestep()
```

```
int simulate_using_transient_multistep(double dT,
                        int numSteps)
```

dT: time-step increment ; `numSteps`: number of time steps ;

```
 int simulate_using_transient_onestep()
```

```
int simulate_using_transient_variable_multistep(double dT,
                                 int numSteps,
                                 double dtMin,
                                 double dtMax,
                                 int Jd)
```

`dT`: time-step increment ; `numSteps`: number of time steps ; `dtMin, dtMax`: minimum and maximum time steps ; `Jd`: ideal number of iterations performed at each step ;

```
int simulate_using_transient_variable_onestep()
```

```
int simulate_using_eigen_analysis(int number_of_eigen_values)
```

204.4 Application Programming Interface for Constitutive Simulations

204.5 Application Programming Interface for Finite Elements

204.6 Adding a New Command into Real-ESSI Simulator

This section describes how to add a new command, needed for the generation of random fields, into Real-ESSI Simulator by modifying an existing command.

204.6.1 Introduction

So far, random fields have been generated using the discrete form of auto-covariance. The size of the auto-covariance matrix is $n \times n$ with n equal to the number of Gauss points (GPs). In the case of a fine FE mesh, i.e., many GPs, this may result in a large computation time. Here, "shear beam" element is used. It has only one GP, in the middle of an element. Hence the number of GPs is here equal to the number of FE elements. It is easier for a user to input the number of FE elements than to input the number of GPs.

Generation of a random field in Real-ESSI is here extended with the possibility of the use of much less discrete points in the solution of the eigenproblem of auto-covariance (now, with the continuous form of auto-covariance using the PCE of auto-covariance function with Legendre polynomials, i.e., the eigenvalues and eigenfunctions are obtained instead of the eigenvalues and eigenvectors) than the GPs in the FEM.

204.6.2 Parser

Parser reads the DSL from input file `∗.fei` and interpretes it. `flex` and `bison` are used.

`HERMITE POLYNOMIAL CHAOS DIMENSION` is the chosen number of the first eigenvalues and eigenfunctions (or eigenvectors) of the auto-covariance function (or matrix) that will be calculated.

Copy existing commands that take

```
HERMITE POLYNOMIAL CHAOS DIMENSION
```

and modify them by adding `number_of_fe_elements`.

For example, copy:

```
Hermite polynomial chaos Karhunen Loeve expansion to random field # <.> with
Hermite polynomial chaos dimension <.> order <.> ↩
    correlation_kernel_inverse_order = <.>;
```

and modify it into:

```
Hermite polynomial chaos Karhunen Loeve expansion to random field # <.> with
Hermite polynomial chaos dimension <.> order <.> ↩
    correlation_kernel_inverse_order = <.> number_of_fe_elements = <.>;
```

In `.../GLOBAL_RELEASE/Real-ESSI/DSL/`, modify 2 files:

- `feiparser.yy` = grammar part handled by `bison`
- `feiparser.l` = lexical part handled by `flex`.

204.6.3 `feiparser.yy`

Change:

```
%token KARHUNEN LOEVE DIMENSION correlation_kernel_inverse_order AS GLOBAL dimension_file TRIPLE PRODUCT shape_parameter scale_parameter WEIBULL TRIANGULAR
```

into:

```
%token KARHUNEN LOEVE DIMENSION correlation_kernel_inverse_order AS GLOBAL dimension_file TRIPLE PRODUCT shape_parameter scale_parameter WEIBULL TRIANGULAR number_of_fe_elements
```

Then, copy the existing commands, like:

```
| HERMITE POLYNOMIAL CHAOS KARHUNEN LOEVE EXPANSION TO RANDOM FIELD TEXTNUMBER exp WITH HERMITE POLYNOMIAL CHAOS DIMENSION exp ORDER exp correlation_kernel_inverse_order '=' exp
    {
        args.clear(); signature.clear();
        args.push_back($11); signature.push_back(this_signature("RandomField_tag", &isAdimensional));
        args.push_back($17); signature.push_back(this_signature("Dimension_num", &isAdimensional));
        args.push_back($19); signature.push_back(this_signature("Order_num", &isAdimensional));
        args.push_back($22); signature.push_back(this_signature("correlation_kernel_inverse_order", &isAdimensional));
        $$ = new FeiDslCaller4<int, int, int, int>(&Hermite_polynomial_chaos_Karhunen_Loeve_expansion_inverse_order, args, signature,
        "Hermite_polynomial_chaos_Karhunen_Loeve_expansion_inverse_order");
        for(int i = 1; i <= 4; i++) nodes.pop();
        nodes.push($$);
    }
```

and modify it into:

```
| HERMITE POLYNOMIAL CHAOS KARHUNEN LOEVE EXPANSION TO RANDOM FIELD TEXTNUMBER exp WITH HERMITE POLYNOMIAL CHAOS DIMENSION exp ORDER exp correlation_kernel_inverse_order '=' number_of_fe_elements '=' exp
```

```
    {
        args.clear(); signature.clear();
        args.push_back($11); ↩
            signature.push_back(this_signature("RandomField_tag", ↩
            &isAdimensional));
        args.push_back($17); ↩
            signature.push_back(this_signature("Dimension_num", &isAdimensional));
        args.push_back($19); signature.push_back(this_signature("Order_num", ↩
            &isAdimensional));
        args.push_back($22); ↩
            signature.push_back(this_signature("correlation_kernel_inverse_order", ↩
            &isAdimensional));
        args.push_back($25); ↩
            signature.push_back(this_signature("number_of_fe_elements", ↩
            &isAdimensional));
        $$ = new FeiDslCaller5<int, int, int, int, ↩
            int>(&Hermite_polynomial_chaos_Karhunen_Loeve_expansion_inverse_order_FE_elements,
            args, signature,
        "Hermite_polynomial_chaos_Karhunen_Loeve_expansion_inverse_order_FE_elements");
        for(int i = 1; i <= 5; i++) nodes.pop();
        nodes.push($$);
    }
```

204.6.4 `feiparser.l`

Just add:

```
"number_of_fe_elements" {return token::number_of_fe_elements;}
```

204.6.5 `create_parallel.sh`

`create_parallel.sh` takes `feiparser.l` and `feiparser.yy`, and creates files needed for compilation of `essi.parallel`. For example, it creates `feiparser.lex.cpp`. In `feiparser.lex.cpp`: ∗`yytext` and `yyleng` will be defined twice and the compiler will complain during linking for `essi.parallel`. The second definitions of ∗`yytext` and `yyleng` need to be found in `feiparser.lex.cpp` and deleted manually. For this purpose, in `create_parallel.sh`, there are 2 lines:

```
sed -i '319d' feiparser.lex.cpp
```

```
sed -i '5642d' feiparser.lex.cpp
```

and lines, here 319 and 5643 (mind the preceding removal of 319), in `feiparser.lex.cpp` will be deleted.

204.6.6 `create_sequential.sh` and `create_parallel.sh`

In `.../GLOBAL_RELEASE/Real-ESSI/DSL/`, one can find `create_sequential.sh` and `create_parallel.sh`. Run both.

204.6.7 Application Programming Interface (API)

In `.../GLOBAL_RELEASE/Real-ESSI/API/MODELING/`, copy the following files:

`Hermite_polynomial_chaos_Karhunen_Loeve_expansion_inverse_order.h`

`Hermite_polynomial_chaos_Karhunen_Loeve_expansion_inverse_order_hdf5_input.h`

and change their names to:

`Hermite_polynomial_chaos_Karhunen_Loeve_expansion_inverse_order_FE_elements.h`

`Hermite_polynomial_chaos_Karhunen_Loeve_expansion_inverse_order_hdf5_input_FE_elements.h`

and modify them to account for the number of GPs (= the number of FE elements here).

In `.../GLOBAL_RELEASE/Real-ESSI/API/`, there is file `api.h`. In `api.h`, add the following lines:

```
#include "MODELING/Hermite_polynomial_chaos_Karhunen_Loeve_expansion_inverse_order_FE_elements.h"
#include "MODELING/Hermite_polynomial_chaos_Karhunen_Loeve_expansion_inverse_order_hdf5_input_FE_elements.h
```

In this case, existing files `RandomField.cpp` and `RandomField.h` will be modified and no new files `*.cpp` and `*.h` have to be created. If new files, for example, `RandomFieldGaussianCorrelation.cpp` and `RandomFieldGaussianCorrelation.h`, are created, then additionally in `ESSI_API.h`, one needs to add:

```
#include <../CompGeoMechUCD_StochasticFEM/PolynomialChaos/RandomFieldGaussianCorrelation.h>
```

204.7 Adding New Finite Element into Real-ESSI Simulator

This section illustrates how to add a new element in Real-ESSI simulator. A detailed description of each steps involved is given. The developer is expected to understand these steps and replicate it for their new element. Also, it is quite useful to look at some previous elements already implemented in Real-ESSI.

204.7.1 Introduction

This documents provides detailed description of steps for adding a new element into the Real-ESSI Simulator. New Element Template source (.ccp) and header (.h) files can be located inside source code in *CompGeoMechUCD_FiniteElements* directory, and are also shown below in subsection 204.7.3 and subsection 204.7.4.

The list of all the steps to be followed are listed below

1. SubSec 204.7.2:: Creating New Element Directory and Linking to Real-ESSI
2. SubSec 204.7.3:: Writing the New Element Header File
3. SubSec 204.7.4:: Writing the New Element Source File
4. SubSec 204.7.5:: Setting the *ELE_TAG_NewElement* class tag and its description
5. SubSec 204.7.6:: Integrating new element with parser.
6. SubSec 204.7.7:: Compiling Real-ESSI
7. SubSec 204.7.8:: Verification of Implementation

These steps are shown in each sub-section. The first step starts creating a directory for the new element. After step [3], the new element would be linked with Real-ESSI source code. So, its good to start compiling (step [6]) and fixing bugs rather than going to step [4] or further ahead.

204.7.2 Getting Started:: Creating New Element Directory

The new element can be added in *CompGeoMechUCD_FiniteElements* directory of Real-ESSI source. A directory for new element lets say NewElement must be created. The next step is to add CMakeLists.txt and place into that directory. The contents of the cmake file is

```
# Builds all the CompGeoMechUCD_FiniteElements/NewElement module
# message("scanning newelements module...")

BUILD_LIB("newelements" ESSI_LIBS)
```

Also, in that directory new element header (NewElement.h) and source files (NewElement.cpp) must be placed. The contents of the cpp files are shown in subsection 204.7.3 and subsection 204.7.4.

The new element then must be included to the header file of Real-ESSI Elements i.e. *CompGeoMechUCD_FiniteElements.h* . The element header file is loaded in *CompGeoMechUCD_FiniteElements* directory. So, just add the new element header as

```
// /////////////////////////////////////////////////
// New Elements [XYZ, Month, Year]
// /////////////////////////////////////////////////

#include "./NewElement/NewElement.h"
```

This would link the new element source code to Real-ESSI. Next is to write the source code and header files of the new element.

204.7.3 Element Header File

The header file is self documented (read fully and carefully).

```
// Rename the header guard
#ifndef NewElementTemplate_h
#define NewElementTemplate_h

#include <Element.h>
#include <Matrix.h>
#include <Vector.h>

class Node;
class Channel;

class NewElementTemplate: public Element
{

public:
  // Constructor
  NewElementTemplate(int tag, int node1, .....); //You must implement this
  // Empty constructor
  NewElementTemplate();
  //Destructor
  ~NewElementTemplate();

  /*********************************************************************/
  /********* Functions to obtain information about dof & connectivity *****/
  /*********************************************************************/
  // returns the number of external nodes of the element
  int getNumExternalNodes(void) const;
  // returns the ID of external nodes of the element
  const ID &getExternalNodes(void);
  // returns the node pointers array to the nodes of the elements
  Node **getNodePtrs(void);
  // rerturn the total number of degrees of freedom for the element
  int getNumDOF(void);
  // returns the DofList containing number of degrees of freedom for each node
  const ID &getDofList();
  // update all the necessary variables before simulation starts
  void setDomain(Domain *theDomain);
  /*********************************************************************/
  /**************** Functions to set the state of the element *************/
  /*********************************************************************/
  // Functions to update the state of the element on obtaining convergence
  int commitState(void);
  // Function to revert to the last commited (converged) state
  int revertToLastCommit(void);
  // Function to revert to the start of the state of the element at the ↩
     beginning of simulation
  int revertToStart(void);
  // Update the element variables for each iteration
```

```
int update(void);
// Remove the load from element
void zeroLoad(void);
// Add Element load
int addLoad(ElementalLoad *theLoad, double loadFactor);
// Send Current Intertial Load of the element
int addInertiaLoadToUnbalance(const Vector &accel);
/************************************************************************/
/********** Functions to obtain stiffness, mass and residual ***********/
/************************************************************************/
// return the current tangent stiffness of the element
const Matrix &getTangentStiff(void);
// return the initial tangent stiffness of the element
const Matrix &getInitialStiff(void);
// return the damping stiffness of the element
const Matrix &getDamp(void);
// return the mass of the element
const Matrix &getMass(void);
// return the resisting force of the element (static case)
const Vector &getResistingForce(void);
// return the resisting force of the element (dynamic case)
const Vector &getResistingForceIncInertia(void);
/************************************************************************/
/********** Functions to implement parallel processing ***********/
/************************************************************************/
// Send the variables to the other CPU in a unique order
int sendSelf(int commitTag, Channel &theChannel);
// For the send order to recieve the information
int receiveSelf(int commitTag, Channel &theChannel, FEM_ObjectBroker ↩
    &theBroker);
/************************************************************************/
/********** Function to Print information about elemnt ***********/
/************************************************************************/
// Print out element info
void Print(ostream &s, int flag = 0);
// Check element correctness
int CheckMesh(ofstream &);
// Give the element a name
std::string getElementName() const
{
    return "NewElementTemplate";
}
/************************************************************************/
/***************** Generate Output of the Element ********************/
/************************************************************************/
/************************************************************************/
/* No. of Element Outputs should be as per the Element_Class_tag_Desc
/* See the classtags.h for more description on encoding of Class_tag ↩
    Descriptions.
/* For Optimization all the information about elements are encoded
/* in the Element_Class_tag Description
```

```
  /* NOTE:- Element_Class_Description [see classTags.h] must be obeyed
  /**************************************************************************/
  // Declare if there is element output except at gauss points
  const vector<float> &getElementOutput() ;
  // Declare only if there is any gauss point and there is 18 outputs
  // per gauss points i.e stress, strain and plastic strain
  const vector<float> &getGaussOutput();
  // Send the Gauss Coordinates of the Elements
  Matrix &getGaussCoordinates(void);

protected:
  /**************************************************************************/
  //Implementation-specific member functions...
  // Should be protected, because they're not going to be called
  // from outside the class, but you might want to inherit them
  /**************************************************************************/

private:
  /**************************************************************************/
  // All data must be private. Provide setter and getter methods if
  // this class interacts with other classes.
  /**************************************************************************/
  // Declare if there is element output except at gauss points
  static vector<float> Element_Output_Vector() ;
  // Declare only if there is any gauss point and there are 18
  // outputs per gauss points i.e stress, strain and plastic strain
  static vector<float> Gauss_Output_Vector();
  // Contains information about Number of Dof for each node of the element
  static ID DofList;
};
#endif
```

204.7.4 Element Source File

The source file is self documented (read fully and carefully).

```
#include <NewElement.h>
// Must define the class tag for
// the new element in this file.
#include <classTags.h>

// NOTE!! Follow the Element_Class_Desc Encoding
// See classTags.h for more details about encoding
// Declare if there is element output except at gauss points
vector<float> ←↩
    NewElementTemplate::Element_Output_Vector(number_of_Element_outputs) ;
// Declare only if there is any gauss point and there is
// 18 outputs per gauss points i.e stress, strain and plastic strain
vector<float> NewElementTemplate::Gauss_Output_Vector(number_of_gauss_points*18);
// Contains information about Number of Dof for each node of the element
ID NewElementTemplate::DofList(number_of_element_nodes);

//*************************************************************
// Constructor. Receive all input parameters. Should not allocate resources!
// * Input: Defined by user. At least should receive an integer tag, so that base
// class can be initialized.
// * Output: void
NewElementTemplate::NewElementTemplate(int tag, int node1, .....):
    Element(tag, ELE_TAG_NewElement),
    // add more initializers
{
    //ATTENTION!
    // ELE_TAG_NewElement !! Define the class tag in classTags.h
    // with provided encoding formula
    // for setting the material id to the element
    this->setMaterialTag(material->getTag());
    // fill DofList container with number of dofs for each node
    you must implement
}

//*************************************************************
// Empty constructor. Create an empty element (with possibly a bad state)
// * Input: Defined by user. At least should receive an integer tag,
// so that base class can be initialized.
// * Output: void
NewElementTemplate::NewElementTemplate():
    Element(0, ELE_TAG_NewElement),
    // add more initializers setting internal variables to null values
{
    //ATTENTION!
    // ELE_TAG_NewElement !! Define the class tag in classTags.h
    // with provided encoding formula
    // fill DofList container with number of dofs for each node
    you must implement
```

```
}

//*****************************************************************
// Destructor. Deallocate resources used by element.
// * Input: void
// * Output: void
NewElementTemplate::~NewElementTemplate()
{
    you must implement
}

//*****************************************************************
// returns the number of nodes of the element.
// * Input: void
// * Output: number of nodes
int NewElementTemplate::getNumExternalNodes(void) const
{
    you must implement
    return number_of_nodes;
}

//*****************************************************************
// Return an ID (integer vector) with the external nodes
// * Input: void
// * Output: ID with tags of external nodes
const ID &NewElementTemplate::getExternalNodes(void)
{
    you must implement
    return external_nodes;
}

//*****************************************************************
// Return pointer array to the nodes
// * Input: void
// * Output: node pointer array.
Node **NewElementTemplate::getNodePtrs(void)
{
    you must implement
    return nodes;
}

//*****************************************************************
//Return the number of dofs in the element.
// * Input: void
// * Output: number of dofs (sum of dofs over all of element's nodes)
int NewElementTemplate::getNumDOF(void)
{
    you must implement
}

//*****************************************************************
```

```
//Return the number of dofs in the element.
// * Input: void
// * Output: DofList containing number of degrees of freedom for each node
const ID &getDofList(){

    you must implement
    return this->DofList;
}

//*****************************************************************
// Receives a domain pointer, and sets the local domain pointer through
// calling the base class setDomain.
// At this point the domain is defined and set, one can allocate resources
// (get nodal pointers, compute some internal variables like lengths, volumes, etc. ).
// Usually we'll set the node pointers here (will be needed for getNodePtrs function).
// Also, we'll check that the given nodes are defined (you get a valid pointer to them) and
// that they have the right number of DOFS (implementation specific)
// * Input: domain pointer (see Domain.h)
// * Output: void
void NewElementTemplate::setDomain(Domain *theDomain)
{
    // check Domain is not null - invoked when object removed from a domain
    if (theDomain == 0)
    {
        //set node pointers to null
    }
    else
    {
        //Use the domain to set the node pointers...
        //nodePointers[0] = theDomain->getNode(Nd1);
        //nodePointers[1] = theDomain->getNode(Nd2);
        // Check the pointers...
        // if (nodePointers == 0)
        // {
        // bad error, usually means node was never
        // return;
        // }
        // Check the number of DOFs
        // if(nodePointers[0]->getNumberDOF() != MY_NUMBER_OF_DOFS)
        // {
        // print a tantrum
        // return;
        // }
        //
        // More checks maybe
        //
        // Set the base class domain pointer

```

```
        this->DomainComponent::setDomain(theDomain);
    }

//Additionally one can allocate resources at this point.
   you must implement

}

//****************************************************************
// Accept current state of the element and save it. (If applicable)
// I this is a gauss-point based element, one calls commitState on
// the material pointers (Gauss points) owned by this element.
// return 0 if success.
// * Input: void
// * Output: error flag, 0 if success
int NewElementTemplate::commitState(void)
{
    you must implement
    return 0;
}

//****************************************************************
// Revert the state of the element to the last committed state.
// Must call for gausspoints if needed.
// * Input: void
// * Output: error flag, 0 if success
int NewElementTemplate::revertToLastCommit(void)
{
    you must implement
    return 0;
}

//****************************************************************
// Revert the state of the element to the initial state.
// Must call for gausspoints if needed.
// * Input: void
// * Output: error flag, 0 if success
int NewElementTemplate::revertToStart(void)
{
    you must implement
    return 0;
}

//****************************************************************
// Update the state of the element. I.E. compute new tangent stiffness,
// compute new resisting force, advance state variables.
// These changes should not be permanent until commit function is called.
// * Input: void
// * Output: error flag, 0 if success
int NewElementTemplate::update(void)
{
```

```
    you must implement
    return 0;
}

//*****************************************************************
// (optionl) Set the elemental load to zero.
// * Input: void
// * Output: void
void NewElementTemplate::zeroLoad(void)
{
    // optional to implement
    return 0;
}

//*****************************************************************
// (optionl) Add a new elemental load. This will modify the
// resisting force vector.
// * Input: ElementalLoad pointer and a loadFactor.
// * Output: error flag, 0 if success
// Notes:
// * ElementalLoads have a type interger (a tag defined elsewhere) and a Vector ↩
    (array
// of doubles) with data. Use these to generate the elemental load scaled by the
// load factor (which is also the time-step of the analysis).
int NewElementTemplate::addLoad(ElementalLoad *theLoad, double loadFactor)
{
    // optional to implement
    //
    // Some code to get the type and data. Example is for self_weight.
    // int type;
    // const Vector &data = theLoad->getData(type, loadFactor);
    //
    // if (type == LOAD_TAG_ElementSelfWeight) // Load tags are defined in ↩
        classTags.h
    // {
    // do something, like add a the forces to the resisting_force vector.
    // }
    return 0;
}

//*****************************************************************
// Add intertial terms to resisting force vector.
// * Input: A vector with nodal accelerations???
// * Output: error flag, 0 if success
// Notes: use node pointers to get accelerations from nodes,
// form an acceleration vector and multiply this with the mass matrix, then
// add this into the load unbalance (with negative sign, cause it is inertia)
int NewElementTemplate::addInertiaLoadToUnbalance(const Vector &accel)
{
    you must implement
    return 0;
```

```
}

//*****************************************************************
// Functions to obtain stiffness, mass, damping and residual information
// * Input: void
// * Output: reference to tangent stiffness matrix (of size nDOF x nDOF,
// where nDOF = NewElementTemplate::getNumDOF();
// Pro tip. If this matrix computes the tangent stiffness, then
// it can be stored as a static member variable so that all elements share
// the same memory space (each element overwrites the tangent). This saves
// memory.
const Matrix &NewElementTemplate::getTangentStiff(void)
{
    you must implement
    return K;
}

//*****************************************************************
// Functions to obtain initial stiffness
// * Input: void
// * Output: reference to initial tangent stiffness matrix (of size nDOF x nDOF,
// where nDOF = NewElementTemplate::getNumDOF();
const Matrix &NewElementTemplate::getInitialStiff(void)
{
    you must implement
    return *K;
    // --suggested variable name for stiffness :).
    // Will be a pointer, so that it can be after construction.
}

//*****************************************************************
// (optional) If element provides its own damping matrix, then this
// function returns it
// * Input: void
// * Output: reference to damping stiffness matrix (of size nDOF x nDOF,
// where nDOF = NewElementTemplate::getNumDOF();
const Matrix &NewElementTemplate::getDamp(void)
{
    // optional to you must implement
}

//*****************************************************************
// (optional) If element provides its own damping matrix, then this
// function returns it
// * Input: void
// * Output: reference to damping stiffness matrix (of size nDOF x nDOF,
// where nDOF = NewElementTemplate::getNumDOF();
const Matrix &NewElementTemplate::getMass(void)
{
    // optional to implement
}
```

```

//**************************************************************
// (optional) If element provides its own damping matrix,
// then this function returns it
// * Input: void
// * Output: reference to damping stiffness matrix (of size nDOF x nDOF,
// where nDOF = NewElementTemplate::getNumDOF();
const Vector &NewElementTemplate::getResistingForce(void)
{
    you must implement
}

//**************************************************************
// (optional) Add inertial terms to resisting force.
// * Input: void
// * Output: Vector of doubles with new resisting force.
// Note: Regularly, this function calls getResistingForce() and then
// adds inertial terms.
const Vector &NewElementTemplate::getResistingForceIncInertia(void)
{
    // (optional to implement)
}

//**************************************************************
// (optional, a must if you want to do parallel processing)
// Send all state data of the element through a channel (usually an MPI_Channel)
// * Input: a reference to the Channel to use.
// * Output: error flag, 0 if success
// Note: This function is usually very involved, and should do a lot of checking
// for pointers and for success of the send.
// Note2: setDomain(...) *might* not be called before using this function.
int NewElementTemplate::sendSelf(int commitTag, Channel &theChannel)
{
    // Useful constructs
    // int error_flag;
    // error_flag = theChannel.sendVector(0, 0, double_data); // the first two ↩
        parameters are deprecated
    //
    // Check that error_flag is not < 0
    //
    // theChannel.sendID(0, 0, integer_data); // the first two parameters are ↩
        deprecated
    //
    // Check that error_flag is not < 0
    you must implement
    return 0;
}

//**************************************************************
// (optional, a must if you want to do parallel processing)
// Receive all state data of the element through a channel
```

```
// (usually an MPI_Channel). This data comes from an element
// that is calling sendSelf in some other process.
// * Input: a reference to the Channel to use.
// * Output: error flag, 0 if success
// Note: This function is called after setDomain() so all resources should be made available.
int NewElementTemplate::receiveSelf(int commitTag, Channel &theChannel, FEM_ObjectBroker &theBroker)
{
    // Useful constructs
    // int error_flag;
    // theChannel.receiveVector(0, 0, double_data); // the first two parameters are deprecated
    // Check that error_flag is not < 0
    // theChannel.receiveID(0, 0, integer_data); // the first two parameters are deprecated
    // Check that error_flag is not < 0

    you must implement
    return 0;
}

//***************************************************************
// Print out element info
// * Input: an ostream to print stuff into, and a flag
// * Output: void
// Print stuff into the ostream by using the "<<" operator.
// The flag can be used to request different levels of printing, ie.
// a flag of 0 might be very basic information, while flag > 0 might
// give increasing ammount of info.
void NewElementTemplate::Print(ostream &s, int flag = 0)
{
    you must implement
}

//***************************************************************
// Check element correctness
// * Input: an ostream to print stuff into (print details of what is being checked here.)
// * Output: an error flag (<0 if element is not right in some way)
// Note: be verbose print element tag, etc. Print out only if an error is encountered.
int NewElementTemplate::CheckMesh(ofstream &)
{
    you must implement
}

//***************************************************************
// Output interface functions
// * Input: void
// * Output: Vector (array of doubles) with the element output.
```

```
const vector<float> &NewElementTemplate::getElementOutput()
{

    Fill the Element_Output_Vector

    return Element_Output_Vector;
}

//****************************************************************
// Output interface functions
// * Input: void
// * Output: Vector (array of doubles) with the element output.
const vector<float> &NewElementTemplate::getGaussOutput()
{
    Fill the Gauss_Output_Vector

    NOTE!!! - Exactly 18 outputs should be there per gauss point

    return Gauss_Output_Vector;
}

//****************************************************************
// Return a Matrix with the coordinates of Gauss points (or points
// where outputs are generated, could be the endpoints of a beam)
// * Input: void
// * Output: Matrix (array of doubles) with the gauss coordinates
// Note: Format is
// gauss_coordinates[0,:] = [x_0,y_0,z_0] -- Coordinates of first Gauss point
// gauss_coordinates[1,:] = [x_1,y_1,z_1] -- Coordinates of second Gauss point
// ...
// gauss_coordinates[Ngauss,:] = [x_Ngauss,y_Ngauss,z_Ngauss]
// -- Coordinates of Ngauss-th Gauss point
Matrix &NewElementTemplate::getGaussCoordinates(void)
{
    you must implement
}

//****************************************************************
// Add you own member functions at the end!
```

204.7.5 Element Class Tag Description

This subsection describes how to set up the *ELE_TAG_NewElement* for the new element. All the tags for element, material, load etc, must be included in *ClassTags.h* located in *ModifiedOpenSeesServices* directory. Each element has unique identifiers:

- Element Tag :: It is a unique tag given to each new element type. The element tags are in sequential order. So the new tag must be the next available tag in sequence.
- Element Tag Description :: It is an encoding containing information about the elements such as, type of element, number of nodes, number of gauss points , number of outputs etc. The element tag has 9 digits and follows a strict encoding a shown below

```
// All elements class tags would be in serial
// numbers from 1-N for optimization
/*********************************************************************
* Desc is [Dimension][N. Nodes][Dof per nodes][N. Gauss][No.of Outputs]
* <1-digit> <2-digit> <1-digit> <3-digit> <2-digit>
* - - - - - - - - - - - - - -- - - - - - - - -- - - - -
*
* [ElementCategory] = <num_of_digits = 1> Category of the element
* 1-> Structural Elements
* 2-> Contact Elements
* 3-> Brick Elements
* 4-> Special elements
*
* [N. Nodes] = <num_of_digits = 2> Number of nodes in elements
* xx-> number of nodes
*
* [Dof per nodes] = <num_of_digits = 1> Degree of freedom per node
* x-> DOFS per node
*
* [N. Gauss] = <num_of_digits = 3> Number of gauss points in elements
* xxx-> number of gauss points
*
* [No.of Outputs] = <num_of_digits = 2> no. of outputs other than at gauss points.
* xx-> no. of outputs other than at gauss points.
*
* Default Features
* - - - - - -- - -
* 1) Per gauss point there are in total 18 outputs
* of stress, plastic strain and total strain
* 2) No.of Outputs -> here means the extra outputs by an element
* except gauss points. For example:
* for eight node brick there is 000 No. of Outputs.
****************************************************************************/

//###----------------------------------------
```

```
//### NOTE!! :- Every Element have a responsibility to set
//### their tag_description Array. Based on the above encoding
//### NOTE!! :- Also increase the ELE_TAG_DESC_ARRAY_SIZE to the
//### number of element tags
//### ----------------------------------------
```

For example: Eight node brick element has element tag description as 308300800. Simple truss element has 102300002 as element tag description.

In order to set up the element tag, look for the next available element tag. Usually, the next available element tag would be equal to the *ELE_TAG_DESC_ARRAY_SIZE* which can be found inside *ClassTag.h* file below the initial element tags. The available number should be added as new element tag and the *ELE_TAG_DESC_ARRAY_SIZE* should be increased by 1.

This should be followed by appending the element class tag description in *ELE_TAG_DESC_ARRAY*. All the steps are shown below,

`ClassTags.h` Before:

```
........
#define ELE_TAG_DispBeamColumn3d 94 // 102600012
#define ELE_TAG_Cosserat_8node_brick 95 // 408600800

#define ELE_TAG_DESC_ARRAY_SIZE 96

.........................
..........................


#define ELE_TAG_DESC_ARRAY int ele_tag_desc_array[] = \
{ELE_TAG_DESC_ENCODING, \
100000000, \
308300800, \
308400800, \
308700800, \
308300100, \
308400100, \
........
........
202300009, \
102300002, \
102600024, \
103600006, \
104600000, \
302300100, \
102600012, \
102600012, \
408600800, \
}
```

`ClassTags.h` After:

```
........
#define ELE_TAG_DispBeamColumn3d 94 // 102600012
#define ELE_TAG_Cosserat_8node_brick 95 // 408600800
#define ELE_TAG_NewElement 96 // XXXXXXXXX

#define ELE_TAG_DESC_ARRAY_SIZE 97

........................
.........................


#define ELE_TAG_DESC_ARRAY int ele_tag_desc_array[] = \
{ELE_TAG_DESC_ENCODING, \
100000000, \
308300800, \
308400800, \
308700800, \
308300100, \
308400100, \
........
........
202300009, \
102300002, \
102600024, \
103600006, \
104600000, \
302300100, \
102600012, \
102600012, \
408600800, \
XXXXXXXXX, \
}
```

204.7.6 Integrating New Finite Element into Parser

Next is to integrate the new element with the parser using *feiparser.yy,feiparser.l* files located in *DSL* directory located in Real-ESSI source. This step requires some knowledge of yacc and lex. The NewElement DSL should added along with other defined element DSL's in *feiparser.yy* file. A typical parser for an element looks like as shown below

```
| TEXTNUMBER exp TYPE NewElement WITH NODES
 '(' exp ',' exp ',' exp ',' exp ')'
  USE MATERIAL TEXTNUMBER exp
  parameter1 '=' exp
  parameter2 '=' exp
{
 args.clear(); signature.clear();
 args.push_back($2); signature.push_back(this_signature("number", &isAdimensional));
 args.push_back($8); signature.push_back(this_signature("node1", &isAdimensional));
 args.push_back($10); signature.push_back(this_signature("node2", &isAdimensional));
 args.push_back($12); signature.push_back(this_signature("node3", &isAdimensional));
 args.push_back($14); signature.push_back(this_signature("node4", &isAdimensional));
 args.push_back($19); signature.push_back(this_signature("material", &isAdimensional));

 args.push_back($22); signature.push_back(this_signature("parameter1", &isThisUnit<-1,3,1>));
 //L^3*T/M
 args.push_back($25); signature.push_back(this_signature("parameter2", &isThisUnit<-1,3,1>));

 $$ = new FeiDslCaller8<int,int,int,int,int,int,
       double,double>(&add_element_new_element, args, signature, "add_element_new_element");

 for(int ii = 1;ii <=8; ii++) nodes.pop();
 nodes.push($$);
}
```

where, the code for DSL of the element corresponds as

```
TEXTNUMBER exp TYPE NewElement WITH NODES
 '(' exp ',' exp ',' exp ',' exp ')'
  USE MATERIAL TEXTNUMBER exp
  parameter1 '=' exp
  parameter2 '=' exp
```

which would look like the following in DSL language

```
add element # <.> type NewElement with nodes
 (<.>, <.>, <.>, <.>) use material # 1
 parameter1 = <.>
 parameter2 = <.>;
```

In the above DSL each word represents a token which must be included in *feiparser.yy* and defined in *feiparser.l*. There are some already defined tokens, which needs not to be defined. One can find if the token exits by searching it in *feiparser.l* or *feiparser.yy*.

In the above DSL, the tokens are TEXTNUMBER, TYPE, NewElement, WITH, NODES, USE, MATERIAL, TEXTNUMBER, parameter1 and parameter2. Among them, some of them like TEXTNUMBER, TYPE, NODES, .. etc are already defined and could be searched. But the tokens NewElement, parameter1 and parameter2 needs to be defined. First, these all undefined tokens needs to be included in *feiparser.yy* and then defined in *feiparser.l*.

204.7.6.1 feiparser.yy

The new tokens must be added in the beginning of the feiparser, where other tokens are defined.

The new tokens can be included as described below.

```
// Tokens for elements
%token EightNodeBrick TwentyNodeBrick TwentySevenNodeBrick
%token NewElement

// Element options tokens
%token porosity alpha rho_s rho_f k_x k_y k_z K_s K_f pressure cross_section ↩
   shear_modulus
%token friction_ratio maximum_gap
%token parameter1 parameter2
```

204.7.6.2 feiparser.l

The tokens needs to be defined as the following.

```
"HardContact" {return token::HardContact;}
"SoftContact" {return token::SoftContact;}
"NewElement" {return token::NewElement;}
```

204.7.6.3 Argument Stack, Signature and Units

Returning back to the code in parser:

```
| TEXTNUMBER exp TYPE NewElement WITH NODES
 '(' exp ',' exp ',' exp ',' exp ')'
  USE MATERIAL TEXTNUMBER exp
  parameter1 '=' exp
  parameter2 '=' exp
{
 args.clear(); signature.clear();
 args.push_back($2); signature.push_back(this_signature("number", ↩
     &isAdimensional));
 args.push_back($8); signature.push_back(this_signature("node1", ↩
     &isAdimensional));
 args.push_back($10); signature.push_back(this_signature("node2", ↩
     &isAdimensional));
 args.push_back($12); signature.push_back(this_signature("node3", ↩
     &isAdimensional));
 args.push_back($14); signature.push_back(this_signature("node4", ↩
     &isAdimensional));
 args.push_back($19); signature.push_back(this_signature("material", ↩
     &isAdimensional));

 args.push_back($22); signature.push_back(this_signature("parameter1", ↩
     &isThisUnit<-1,3,1>));
 //L^3*T/M
 args.push_back($25); signature.push_back(this_signature("parameter2", ↩
     &isThisUnit<-1,3,1>));

 $$ = new FeiDslCaller8<int,int,int,int,int,int,
        double,double>(&add_element_new_element, args, signature, ↩
            "add_element_new_element");

 for(int ii = 1;ii <=8; ii++) nodes.pop();
 nodes.push($$);
}
```

Each variable or parameter in Real-ESSI has units. So, for each of the variables UNIT must be specified. In the above code. *args* is a stack that should be filled with the tokens that are the parameters of element. The first step is to clear the *args* and the *signature* stacks. Pushing the element parameter tokens is done as described below

```
args.push_back($2); signature.push_back(this_signature("number", ↩
    &isAdimensional));
args.push_back($22); signature.push_back(this_signature("parameter1", ↩
    &isThisUnit<-1,3,1>));
 //L^3*T/M
```

Here, \$2 responds to the the second token i.e exp after *TEXTNUMBER*. Similarly, 5^{th} token in TYPE. In signature, the string can be anything, but should usually be the parameter name. The last is enforcing the units for each of the parameters by defining the required units. There are many units, such as *isAdimensional, isAdimensional, isMass, isLength, isTime, isFrequency, isArea, isVolume, isForce, isEnergy, isTorque, isPressure, isBodyForce, isDensity, isVelocity, isAcceleration, isAreaMomentOfInertia, isMassMomentOfInertia*. If the parameter has some other units then the other units can be defined as

$$\text{isThisUnit}< m, l, t >$$

which refers to the unit $M^m L^l T^t$ in standard units. Here, M, L, T are mass, length and time respectively.

204.7.6.4 FeiDslCaller

The next step is to send all the parameters to DSL Header File. The header file contains the code to create a new element inside simulation domain. The code that does this is

```
$$ = new FeiDslCaller8<int,int,int,int,int,int,
      double,double>
      (&add_element_new_element, args,
      signature, "add_element_new_element");

for(int ii = 1;ii <=8; ii++) nodes.pop();
nodes.push($$);
```

FeiDslCaller takes all the arguments and passes to the *add_element_new_element.h* header file. In the above code, the number 8 in function *FeiDslCaller* corresponds to the total number of arguments to the element and following that, the type of the arguments is defined. Here, the type of 8 arguments are *int, int, int,int, int, int, double, double*. The last step is the remove everything from the node stack by popping it equal to the number of times of argument and then finally pushing the *FeiDslCaller* to the nodes stack.

204.7.6.5 New DSL Header File

The header file of the new DSL must be created in */API/MODELING* directory. The header file must also be included in */API/api.h* header file. Here, the header file *add_element_new_element.h* has been created, which is called by *FeiDslCaller*.

```
int add_element_new_element(int ElementNumber, int Node1, int Node2, int Node3,
        int Node4, int MaterialId, double Parameter1, double Parameter2)
{
    Element* theElement = 0;
    theElement = new NewElement(ElementNumber, Node1, Node2, Node3, Node4,
    MaterialId, Parameter1, Parameter2);

    if (theElement == NULL){
```

```
        cerr << "Error: (add_element_new_element)
        memory allocation problem for theElement!" << endl;
        return -1;
    }

    if (theDomain.addElement(theElement) == false){
        cerr << "WARNING (add_element_new_element)
        could not add element to the domain\n";
        cerr << "Element Number: " << ElementNumber << endln;
        return -1;
    }

    return 0;
};
```

Header file in */API/api.h* file needs to be included:

```
// ##########################
// New Element [ABCD Month, Year]
// ##########################

#include "MODELING/add_element_new_element.h"
```

204.7.7 Compiling Real-ESSI

It is assumed that the person reading this document is developer and thus should already have the Real-ESSI dependencies. To compile Real-ESSI with the new element, it should be build from the beginning. Assuming that the build directory in *build* inside Real-ESSI source code, the steps to recompile are

```
cd build
rm -r *
cmake ..
make -j 20
```

204.7.8 Verification of Implementation

Once, the element is fully integrated with Real-ESSI Simulator, the developer should fully verify the implementation by carrying out verification runs. In addition, the developer should be able to run their examples in sequential and parallel and verify the implementation.

Chapter 205

Input, Domain Specific Language

(1991-2005-2010-2011-2012-2015-2016-2017-2018-2019-2020-2021-)

(In collaboration with Prof. José Abell, Dr. Yuan Feng, and Dr. Hexiang Wang)

205.1 Chapter Summary and Highlights

205.2 Introduction

This chapter presents the domain specific language developed for the Real-ESSI. The language was designed with a primary goal of developing FEA models and interfacing them with various Real-ESSI functionalities. In addition to that, syntax is used to self-document models, provide physical-unit safety, provide common flow control structures, provide modularity to scripting via user functions and "include" files, and provide an interactive environment within which models can be created, validated and verified.

The development of Real-ESSI Domain specific language (DSL) (the Finite Element Interpreter, FEI) is based on LEX (Lesk and Schmidt, 1975) and YACC (Johnson, 1975).

Self-documenting ensures that the resulting model script is readable and understandable with little or no reference to the users manual. This is accomplished by providing a command grammar structure and wording similar to what would be used in a natural language description of the problem.

FEA analysis is unitless, that is, all calculations are carried out without referencing a particular unit system. This leaves the task of unit correctness up to the user of FEA analysis. This represents a recurring source of error in FEA analysis. Physical unit safety is enforced in Real-ESSI by implementing all base variables as physical quantities, that is, all variables have a unit associated with it. The adimensional unit is the base unit for those variables which have no relevant unit (like node numbers). Command calls are sensitive to units. For example, the node creation command call expects the node coordinates to be input with the corresponding units (length in this case). Additionally, the programming/command language naturally supports operation with units like arithmetic operations (quantities with different unit types will not add or subtract but may be multiplied). This approach to FEA with unit awareness provides an additional layer of security to FEA calculations, and forces the user to carefully think about units. This can help catch some common mistakes.

The Real-ESSI language provides modularity through the `include` directive/command, and user functions. This allows complex analysis cases to be parameterized into modules and functions which can be reused in other models.

Finally, an emphasis is placed on model verification and validation. To this end, Real-ESSI provides an interactive programming environment with all the ESSI syntax available. By using this environment. the user can develop tests to detect errors in the model that are not programming errors. For example, the user can query nodes and elements to see if they are set to appropriate states. Also, several standard tools are provided to check element validity (Jacobian, etc.).

The ESSI language provides reduced model development time by providing the aforementioned features along with meaningful error reporting (of syntax and grammatical errors), a help system, command completion and highlighting for several open source and commercial text editors.

Some additional ideas are given by Dmitriev (2004), Stroustrup (2005), Niebler (2005), Mernik et al. (2005), Ward (2003), etc.

205.3 Domain Specific Language (DSL), English Language Binding

Overview of the language syntax.

- Each command line has to end with a semicolon ";"
- Comment on a line begins with either "//" or "!" and last until the end of current line.
- Units are required (see more below) for all quantities and variables.
- Include statements allow splitting source into several files
- All variables are double precision (i.e. floats) with a unit attached.
- All standard arithmetic operations are implemented, and are unit sensitive.
- Internally, all units are represented in the base SI units (m - s - kg).
- The syntax ignores extra white spaces, tabulations and newlines. Wherever they appear, they are there for code readability only. (This is why all commands need to end with a semicolon).
- The user should be familiar with the list of the reserved keywords from Section 205.7 on page 1163.

205.3.1 Running Real-ESSI

At the command line type "essi", to get to the ESSI prompt and start Real-ESSI in interactive mode.

Command line output

```

  The Finite Element Interpreter Endeavor

  The Real-ESSI Simulator
  Modeling and Simulation of Earthquakes, and Soils, and ↩
     Structures and their Interaction

  Sequential processing mode.

Version Name      : Real-ESSI Global Release, June2018. Release ↩
   date: Jun 13 2018 at 11:02:19. Tag: adc085ae70
```

```
Version Branch    : GLOBAL_RELEASE
Compile Date      : Jun 13 2018 at 14:36:56
Compile User      : jeremic
Compile Sysinfo   : sokocalo 4.13.0-43-generic x86_64 GNU/Linux
Runtime User      : jeremic
Runtime Sysinfo   : sokocalo 4.13.0-43-generic x86_64 GNU/Linux
Time Now          : Jun 13 2018 at 15:32:52
Days From Release : 0
PostProcessing Compatible Version: ParaView 5.1.2
PostProcessing Compatible Version: ESSI-pvESSI Date: Feb 15 2018 ↩
   at 11:00:28. Tag: 58fe430a19

Static startup tips:
 * Remember: Every command ends with a semicolon ';'.
 * Type 'quit;' or 'exit;' to finish.
 * Run 'essi -h' to see available command line options.

ESSI>
```

A number of useful information about Real-ESSI is printed on the screen. From here, commands can be input manually or a file may be included via the `include` command which is as follows.

```
include "foobar.fei";
```

to include the file `foobar.fei`.

A more efficient way to start Real-ESSI and analyze an example is to pass input file name to the command line. Real-ESSI command to execute an input file immediately is done by issuing the following command: `essi -f foobar.fei`. This will execute essi directly on input file `foobar.fei`. After executing the file, the essi interpreter will continue in interactive mode unless the command line flag `-n` or `--no-interactive` is set. A list of command line options is available by calling essi from the command line as `essi -h`.

Command line output

```


  The Finite Element Interpreter Endeavor

  The Real-ESSI Simulator
  Modeling and Simulation of Earthquakes, and Soils, and ↩
     Structures and their Interaction

  Sequential processing mode.

Version Name      : Real-ESSI Global Release, June2018. Release ↩
   date: Jun 13 2018 at 11:02:19. Tag: adc085ae70
Version Branch    : GLOBAL_RELEASE
Compile Date      : Jun 13 2018 at 14:36:56
```

```
Compile User      : jeremic
Compile Sysinfo   : sokocalo 4.13.0-43-generic x86_64 GNU/Linux
Runtime User      : jeremic
Runtime Sysinfo   : sokocalo 4.13.0-43-generic x86_64 GNU/Linux
Time Now          : Jun 13 2018 at 16:22:08
Days From Release : 0
PostProcessing Compatible Version: ParaView 5.1.2
PostProcessing Compatible Version: ESSI-pvESSI Date: Feb 15 2018 ↩
   at 11:00:28. Tag: 58fe430a19

Static startup tips:
 * Remember: Every command ends with a semicolon ';'.
 * Type 'quit;' or 'exit;' to finish.
 * Run 'essi -h' to see available command line options.

The Real-ESSI Simulator
Modeling and Simulation of Earthquakes, and Soils, and Structures ↩
   and their Interaction

Usage: essi [-cfhnsmbe FILENAME]
  -c --cpp-output                : Output cpp version of the model.
  -f --filename [FILENAME]       : run ESSI on a FILENAME.
  -h --help                      : Print this message.
  -n --no-interactive            : Disable interactive mode.
  -s --set-variable              : Set a variable from the ↩
     command line.
  -d --dry-run                   : Do not execute ESSI API calls. ↩
     Just parse.
  -m --model-name [NAME]         : Set the model name from the ↩
     command line.
  -p --profile-report [FILENAME] : Set the filename for the ↩
     profiler report (and activate lightweight profiling)


Example to set a variable name from command line:
    essi -s a=10,b=20,c=30
Runs ESSI with variables a, b, and c set to 10, 20 and 30 ↩
   respectively.
At this time, only ESSIunits::unitless variables can be set.
```

205.3.2 Finishing Real-ESSI Program Run

To properly finish Real-ESSI program run, and save and close all the output files, user has to use final, closure command:

```
bye;
```

Command `bye;` has to be included at the end of input file script, or at the end of each interactive/interpretative session. Command `bye;` ensures that Real-ESSI program gracefully exits simulation, and that all the output files are properly saved and closed. Proper finishing of simulation using Real-ESSI Simulator is very much necessary, while the choice of command `bye;` is done as an homage to Professor Knuth and his Literate Programming endeavor (Knuth, 1984), that is driving much of the Real-ESSI DSL development.

There are a number of alternative final commands, for example:

```
exit;
quit;
zdravo;
vozdra;
dvojka;
voljno;
zaijian;
tschuess;
geia-sou;
tchau;
sair;
khoda-hafez;
doei;
nasvidenje;
ajde-bok;
izhod;
konec;
czesc;
ciao;
hoscakal;
```

These additional, alternative final commands can all be written using original scripts:

zdravo ↔ здраво

vozdra ↔ воздра

dvojka ↔ двојка

voljno ↔ вољно

zaijian ↔ 再见

tschuess ↔ tschüss

geia-sou ↔ γεια σου

khoda-hafez ↔ خداحافظ

hoscakal ↔ hoşçakal

205.3.3 Real-ESSI Variables, Basic Units and Flow Control

Variables are defined using the assignment (=) operator. For example,

```
var_x = 7; //Results in the variable x be set to 7 (unitless)
var_y = 3.972e+2; //Scientific notation is available.
```

The language contains a list of reserved keywords. Throughout this documentation, reserved keywords are highlighted in blue or red.

All standard arithmetic operations are available between variables. These operations can be combined arbitrarily and grouped together with parentheses.

```
var_a = var_x + var_y; // Addition
var_b = var_x - var_y; // Subtraction
var_c = var_x * var_y; // Product
var_d = var_x / var_y; // Quotient
var_e = var_y % var_x; // Modulus (how many times x fits in y)
```

The '`print`' command can be used to display the current value of a variable.

```
print var_x;
print var_y;
print var_a;
print var_b;
print var_c;
print var_d;
print var_e;
```

Command line output

```
var_x = 7 []
var_y = 397.2 []
var_a = 404.2 []
var_b = -390.2 []
var_c = 2780.4 []
var_d = 0.0176234 []
var_e = 5.2 []
```

Here the “unit” (sign) [] means that the quantities are unitless.

The command '`whos`' is used to see all the currently defined variables and their values. After a fresh start of essi, needed to clear up all the previously defined variables, command `whos;`' produces a list of predefined variables:

Command line output

```
ESSI> whos;

Declared variables:
  *      Day =           86400 [s]
```

```
  *      GPa =               1 [GPa]
  *     Hour =            3600 [s]
  *       Hz =               1 [Hz]
  *      MPa =               1 [MPa]
  *   Minute =              60 [s]
  *        N =               1 [N]
  *       Pa =               1 [Pa]
  *     Week =          604800 [s]
  *       cm =               1 [cm]
  *     feet =          0.3048 [m]
  *       ft =          0.3048 [m]
  *        g =            9.81 [m*s^-2]
  *     inch =          0.0254 [m]
  *       kN =               1 [kN]
  *      kPa =               1 [kPa]
  *       kg =               1 [kg]
  *      kip =         4448.22 [N]
  *       km =               1 [km]
  *      ksi =     6.89476e+06 [Pa]
  *      lbf =         4.44822 [N]
  *      lbm =        0.453592 [kg]
  *        m =               1 [m]
  *     mile =         1609.35 [m]
  *       mm =               1 [mm]
  *       pi =         3.14159 []
  *      psi =         6894.76 [Pa]
  *        s =               1 [s]
  *     yard =          0.9144 [m]

    * = locked variable
ESSI>
```

Predefined variables shown above have a preceding asterisk to show they are locked variables which cannot be modified. The purpose of these locked variables are to provide names for units. Imperial units are also supported as shown above.

The units for variable are shown between the brackets. Note that unit variables have the same name as their unit, which is not the case for user defined variables. Variables preceded by a star (*) are locked variables which can't be modified.

For example, the variable 'm' defines 'meter'. So to define a new variable `L1` which has meter units we do:

```
L1 = 1*m; // Defines L1 to 1 m.
L2 = 40*mm; // Defines L2 to be 40 millimeters.
```

Even though L2 was created with millimeter units, it is stored in base units.

`print L2;` displays

Command line output

```
L2 = 0.04 [m]
```

As additional examples, let us define few forces:

```
F1 = 10*kN;
F2 = 300*N;
F3 = 4*kg*g;
```

Here g is the predefined acceleration due to gravity.

Arithmetic operations do check (and enforce) for unit consistency. For example, `foo = L1 + F1;` produces an error because units are not compatible. However, `bar = L1 + L2;` is acceptable. On the other hand, multiplication, division and modulus, always work because the result produces a quantity with new units (except when the adimensional quantity is involved).

```
A = L1*L2;
Stress_n = F1 / A;
```

Units for all variables are internally converted to SI units ($kg-m-s$) and stored in that unit system. Variables can be displayed using different units by using the `[]` operator. This does not change the variable, it just displays the value of variable with required unit. For example,

```
print Stress_n; //Print in base SI units.
print Stress_n in Pa; //Print in Pascal
print Stress_n in kPa; //Print in kilo Pascal
```

Command line output

```
Stress_n = 250000 [kg*m^-1*s^-2]

Stress_n = 250000 [Pa]

Stress_n = 250 [kPa]
```

The DSL provides functions to test the physical units of variables. For example,

`print isForce(F1);`

Will print an adimensional, Boolean `1` because `F1` has units of force. While,

`print isPressure(F);`

will print an adimensional, Boolean `0`. The language also provides comparison of quantities with same units (remember all values are compared in SI Units).

`print F1 > F2;`

will print an adimensional, Boolean `1` since `F1` is greater than `F2`.

The program flow can be controlled with `if` and `while` statements, i.e.:

```
if (isForce(F1))
{
 print F1; // This will be executed
};

if (isForce(L1))
{
 print L1; // This will not.
};
```

Note the necessary semicolon (;) at the closing brace. Unlike C/C++, the braces are always necessary. Closing colon is also always necessary.

The "else" statement is also available:

```
if (isForce(L1))
{
 print L1; // This will not execute
}
else
{
 print L2; // This will execute instead
};
```

While loops are also available:

```
i = 0;
while( i < 10)
{
 print i;
 i = i +1;
};
```

205.3.4 Modeling

This section details ESSI modeling commands. Angle brackets <> are used for quantity or variable placeholder, that is, they indicate where user input goes. Within the angle brackets, the expected unit type is given as well, i.e.. `<L>` means the command expects an input with a value and a length unit. The symbol `<.>` represents the adimensional quantity.

In addition to that, the vertical bar | ("OR" sign)) is used to separate two or more keyword options, i.e. `[a|b|c]` is used indicate keyword options `a` or `b` or `c`. The symbol `|...|` is used to denote where several long options exist and are explained elsewhere (an example of this is available below in a material model definitions).

All commands require unit consistency. Base units, SI or other can be used as indicated below:

- length, symbol L, units [m, inch, ft]
- mass, symbol M, units [kg, lbm],
- time, symbol T, units [s]

Derived units can also be used:

- angle, symbol rad (radian), unit $[dimensionless, L/L]$
- force, symbol N (Newton), units $[N, kN, MN, M * L/T^2]$,
- stress, symbol Pa (Pascal), units $[Pa, kPa, MPa, N/L^2, M/L/T^2]$
- strain, symbol (no symbol), units $[L/L]$
- mass density, symbol (no symbol), units $[M/L^3]$
- force density, symbol (no symbol), units $[M/L^2/T^2]$

All models have to be named: `model name "model_name_string";` This is important as output files are named based on model name.

Each loading stage has to be named as well. A new loading stage[1] is defined like this:

`new loading stage "loading stage name string";`

In addition to model name, loading stage name is used for output file name for given loading stage.

---

[1] See more in section 101.4.5 on page 97 in Jeremić et al. (1989-2025).

205.3.4.1 Modeling, Material Model: Adding a Material Model to the Finite Element Model

Adding constitutive material model to the finite element model/domain is done using command:

```
add material # <.> type |...|
                mass_density = <M/L^3>
                (more model dependent parameters) ;
```

- Material number # (or alternatively `No`) is a distinct integer number used to uniquely identify this material.
- Mass density should be defined for each material (even if only static analysis is performed, for example if self weight is to be used as a loading stage).
- Depending on material model, there will be additional material parameters that are defined for each material model/type below:

Starting with version `03-NOV-2015` all elastic-plastic material models require explicit specification of the constitutive integration algorithm. More information on this can be found in 205.3.5.15. Only the material `linear_elastic_isotropic_3d_LT` ignores this option.

Choices for `material_type` are listed below.

205.3.4.2 Modeling, Material Model: Linear Elastic Isotropic Material Model

The command is:

```
add material # <.> type linear_elastic_isotropic_3d
    mass_density = <M/L^3>
    elastic_modulus = <F/L^2>
    poisson_ratio = <.>;
```

where:

- `mass_density` is the mass density of material $[M/L^3]$
- `elastic_modulus` is an isotropic modulus of elasticity of a material (units: stress)
- `poisson_ratio` is a Poisson's ratio [dimensionless]

More on this material model can be found in Section 104.6.1 on Page 220 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.3 Modeling, Material Model: Cross Anisotropic Linear Elastic Material Model

The command is:

```
add material # <.> <material_number>
            type linear_elastic_crossanisotropic
            mass_density = <M/L^3>
            elastic_modulus_horizontal = <F/L^2>
            elastic_modulus_vertical = <F/L^2>
            poisson_ratio_h_v = <.>
            poisson_ratio_h_h = <.>
            shear_modulus_h_v = <F/L^2>;
```

where:

- `mass_density` is the mass density of material $[M/L^3]$
- `elastic_modulus_horizontal` is an anisotropic modulus of elasticity for horizontal plane of a material $[F/L^2]$
- `elastic_modulus_vertical` is an anisotropic modulus of elasticity for vertical direction of a material $[F/L^2]$
- `poisson_ratio_h_v` is a Poisson's ratio for horizontal-vertical directions [dimensionless]
- `poisson_ratio_h_h` is a Poisson's ratio for horizontal-horizontal directions [dimensionless]
- `shear_modulus_h_v` is a shear modulus for horizontal-vertical directions $[F/L^2]$

It is assumed that vertical axes is global Z axes.

More on this material model can be found in Section 104.6.1 on Page 220 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.4 Modeling, Material Model: von Mises Associated Material Model with Linear Isotropic and/or Kinematic Hardening

Implements von Mises family of constitutive models, with linear kinematic and/or isotropic hardening.

The command is:

```
add material # <.> type vonMises
  mass_density = <M/L^3>
  elastic_modulus = <F/L^2>
  poisson_ratio = <.>
  von_mises_radius = <F/L^2>
  kinematic_hardening_rate = <F/L^2>
  isotropic_hardening_rate = <F/L^2> ;
```

where:

- `mass_density` is the mass density of material $[M/L^3]$
- `elastic_modulus` is the elastic modulus of material $[F/L^2]$
- `poisson_ratio` is the Poisson's ratio material [ ]
- `von_mises_radius` is the radius of the deviatoric section of the von Mises yield surface $[F/L^2]$
- `kinematic_hardening_rate` is the rate of the kinematic hardening $[F/L^2]$
- `isotropic_hardening_rate` is the rate of the kinematic hardening $[F/L^2]$

More on this material model can be found in Section 104.6.6 on Page 226 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.5 Modeling, Material Model: von Mises Associated Material Model with Isotropic Hardening and/or Armstrong-Frederic Nonlinear Kinematic Hardening

This command is for von Mises family of constitutive models, with Armstrong-Frederick kinematic and/or isotropic hardening.

The command is:

```
add material # <.> type vonMisesArmstrongFrederick
    mass_density = <M/L^3>
    elastic_modulus = <F/L^2>
    poisson_ratio = <.>
    von_mises_radius = <.>
    armstrong_frederick_ha = <F/L^2>
    armstrong_frederick_cr = <.>
    isotropic_hardening_rate = <F/L^2> ;
```

where:

- `mass_density` is the mass density of material $[M/L^3]$
- `elastic_modulus` is the elastic modulus of material $[F/L^2]$
- `poisson_ratio` is the Poisson's ratio material [ ]
- `von_mises_radius` is the radius of the deviatoric section of the von Mises yield surface $[F/L^2]$
- `armstrong_frederick_ha` controls rate of the kinematic hardening $[F/L^2]$
- `armstrong_frederick_cr` controls the saturation limit for kinematic hardening [Dimensionless]
- `isotropic_hardening_rate` is the rate of the kinematic hardening $[F/L^2]$

More on this material model can be found in Section 104.6.6 on Page 226 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.6 Modeling, Material Model: Drucker-Prager Associated Material Model with Linear Isotropic and/or Kinematic Hardening

This command is for Drucker-Prager family of constitutive models, with linear kinematic and/or isotropic hardening. This material uses associate plastic flow rule.

The command is:

```
add material # <.> type DruckerPrager
  mass_density = <M/L^3>
  elastic_modulus = <F/L^2>
  poisson_ratio = <.>
  druckerprager_k = <.>
  kinematic_hardening_rate = <F/L^2>
  isotropic_hardening_rate = <F/L^2>
  initial_confining_stress = <F/L^2> ;
```

where:

- `mass_density` is the mass density of material $[M/L^3]$
- `elastic_modulus` is the elastic modulus of material $[F/L^2]$
- `poisson_ratio` is the Poisson's ratio material [ ]
- `druckerprager_k` slope of the Drucker-Prager yield surface in p-qm space (equivalent to M parameter) [dimensionless]
- `kinematic_hardening_rate` is the rate of the kinematic hardening $[F/L^2]$
- `isotropic_hardening_rate` is the rate of the isotropic hardening $[F/L^2]$

More on this material model can be found in Section 104.6.7 on Page 232 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.7 Modeling, Material Model: Drucker-Prager Associated Material Model with Isotropic Hardening and/or Armstrong-Frederick Nonlinear Kinematic Hardening

A Drucker-Prager constitutive model with associative plastic-flow rule, Armstrong-Frederick kinematic hardening, and linear isotropic hardening and linear elastic isotropic elasticity law.

The command is:

```
add material # <.> type DruckerPragerArmstrongFrederickLE
   mass_density = <M/L^3>
   elastic_modulus = <F/L^2>
   poisson_ratio = <.>
   druckerprager_k = <.>
   armstrong_frederick_ha = <F/L^2>
   armstrong_frederick_cr = <.>
   isotropic_hardening_rate = <F/L^2>
   initial_confining_stress = <F/L^2>;
```

where:

- `mass_density` is the mass density of material $[M/L^3]$
- `elastic_modulus` is the elastic modulus of material $[F/L^2]$
- `poisson_ratio` is the Poisson's ratio material [ ]
- `druckerprager_k` slope of the Drucker-Prager yield surface in p-qm space (equivalent to M parameter) [Dimensionless]
- `armstrong_frederick_ha` controls rate of the kinematic hardening $[F/L^2]$
- `armstrong_frederick_cr` controls the saturation limit for kinematic hardening [Dimensionless]
- `isotropic_hardening_rate` is the rate of the isotropic hardening $[F/L^2]$
- `initial_confining_stress` initial confining (mean) pressure $[F/L^2]$

More on this material model can be found in Section 104.6.7 on Page 232 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.8 Modeling, Material Model: Drucker-Prager Associated Material Model with Isotropic Hardening and/or Armstrong-Frederick Nonlinear Kinematic Hardening and Nonlinear Duncan-Chang Elasticity

A Drucker-Prager constitutive model with associative plastic-flow rule, Armstrong-Frederick kinematic hardening, and Duncan-Chang non-linear isotropic elasticity law.

The command is:

```
add material # <.> type DruckerPragerArmstrongFrederickNE
  mass_density = <M/L^3>
  DuncanChang_K = <.>
  DuncanChang_pa = <F/L^2>
  DuncanChang_n = <.>
  DuncanChang_sigma3_max = <F/L^2>
  DuncanChang_nu = <.>
  druckerprager_k = <.>
  armstrong_frederick_ha = <F/L^2>
  armstrong_frederick_cr = <.>
  isotropic_hardening_rate = <F/L^2>
  initial_confining_stress = <F/L^2>;
```

where:

- `mass_density` is the mass density of material $[M/L^3]$
- `DuncanChang_K` parameter controlling Young's modulus $[< . >]$
- `DuncanChang_pa` reference pressure $[F/L^2]$
- `DuncanChang_n` exponent $[< . >]$
- `DuncanChang_sigma3_max` maximum value for σ_3 ($\sigma_3 < 0$) elastic properties are constant for greater values of σ_3 $[F/L^2]$
- `DuncanChang_nu` Poisson's ratio $[F/L^2]$
- `druckerprager_k` slope of the Drucker-Prager yield surface in p-qm space (equivalent to M parameter) [Dimensionless]
- `armstrong_frederick_ha` controls rate of the kinematic hardening $[F/L^2]$
- `armstrong_frederick_cr` controls the saturation limit for kinematic hardening [Dimensionless]
- `isotropic_hardening_rate` is the rate of the isotropic hardening $[F/L^2]$
- `initial_confining_stress` initial confining (mean) pressure $[F/L^2]$

More on this material model can be found in Section 104.6.7 on Page 232 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.9 Modeling, Material Model: Drucker-Prager Nonassociated Material Model with Linear Isotropic and/or Kinematic Hardening

This command defines Drucker-Prager family of constitutive models, with linear kinematic and/or isotropic hardening. This material uses non-associate plastic flow rule.

The command is:

```
add material # <.> type DruckerPragerNonAssociateLinearHardening
    mass_density = <M/L^3>
    elastic_modulus = <F/L^2>
    poisson_ratio = <.>
    druckerprager_k = <.>
    kinematic_hardening_rate = <F/L^2>
    isotropic_hardening_rate = <F/L^2>
    initial_confining_stress = <F/L^2>
    plastic_flow_xi = <.>
    plastic_flow_kd = <.>;
```

where:

- `mass_density` is the mass density of material $[M/L^3]$
- `elastic_modulus` is the elastic modulus of material $[F/L^2]$
- `poisson_ratio` is the Poisson's ratio material [ ]
- `druckerprager_k` slope of the Drucker-Prager yield surface in p-qm space (equivalent to M parameter) [Dimensionless]
- `kinematic_hardening_rate` is the linear rate of the kinematic hardening $[F/L^2]$
- `isotropic_hardening_rate` is the linear rate of the isotropic hardening $[F/L^2]$
- `initial_confining_stress` initial confining (mean) pressure $[F/L^2]$
- `plastic_flow_xi` governs the amplitude of plastic volume changes. The higher ξ, the higher the dilatancy. If $\xi = 0$, the material model will only produce deviatoric plastic strains. [.]
- `plastic_flow_kd` governs the size of the dilatancy surface, a cone in the stress space on which no plastic volume changes occur. k_d governs the size of this cone: if k_d is equal to zero, the dilatancy surface shrinks to a line (the hydrostatic axis), so that only dilative soil deformation is possible. [.]

More on this material model can be found in Section 104.6.7 on Page 232 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.10 Modeling, Material Model: Drucker-Prager Nonassociated Material Model with Linear Isotropic and/or Armstrong-Frederick Nonlinear Kinematic Hardening

This command defines Drucker-Prager family of constitutive models, with nonlinear kinematic and/or linear isotropic hardening. This material uses non-associated plastic flow rule.

The command is:

```
add material # <.> type DruckerPragerNonAssociateArmstrongFrederick
    mass_density = <M/L^3>
    elastic_modulus = <F/L^2>
    poisson_ratio = <.>
    druckerprager_k = <.>
    armstrong_frederick_ha = <F/L^2>
    armstrong_frederick_cr = <.>
    isotropic_hardening_rate = <F/L^2>
    initial_confining_stress = <F/L^2>
    plastic_flow_xi = <.>
    plastic_flow_kd = <.>;
```

where:

- `mass_density` is the mass density of material $[M/L^3]$
- `elastic_modulus` is the elastic modulus of material $[F/L^2]$
- `poisson_ratio` is the Poisson's ratio material [ ]
- `druckerprager_k` slope of the Drucker-Prager yield surface in p-qm space (equivalent to M parameter) [Dimensionless]
- `armstrong_frederick_ha` a kinematic hardening parameter, which governs the initial stiffness after the yield $[F/L^2]$
- `armstrong_frederick_cr` a kinematic hardening parameter. $\frac{h_a}{c_r}$ governs the limit of the back-stress [Dimensionless]
- `isotropic_hardening_rate` is the rate of the kinematic hardening $[F/L^2]$
- `initial_confining_stress` initial confining (mean) pressure $[F/L^2]$
- `plastic_flow_xi` governs the amplitude of plastic volume changes - the higher ξ, the higher the dilatancy. If $\xi = 0$, the material model will only produce deviatoric plastic strains. [.]
- `plastic_flow_kd` governs the size of the dilatancy surface, a cone in the stress space on which no plastic volume changes occur. k_d governs the size of this cone: if k_d is equal to zero, the

> dilatancy surface shrinks to a line (the hydrostatic axis), so that only dilative soil deformation is possible. [.]

More on this material model can be found in Section 104.6.7 on Page 232 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

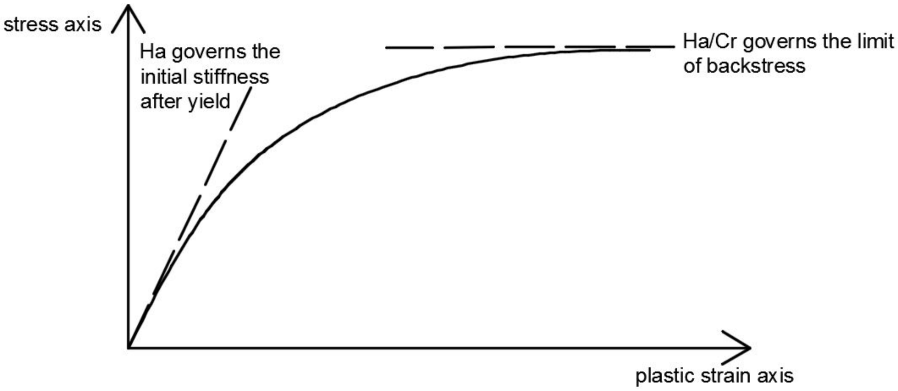


Figure 205.1: The physical meanings of h_a and c_r

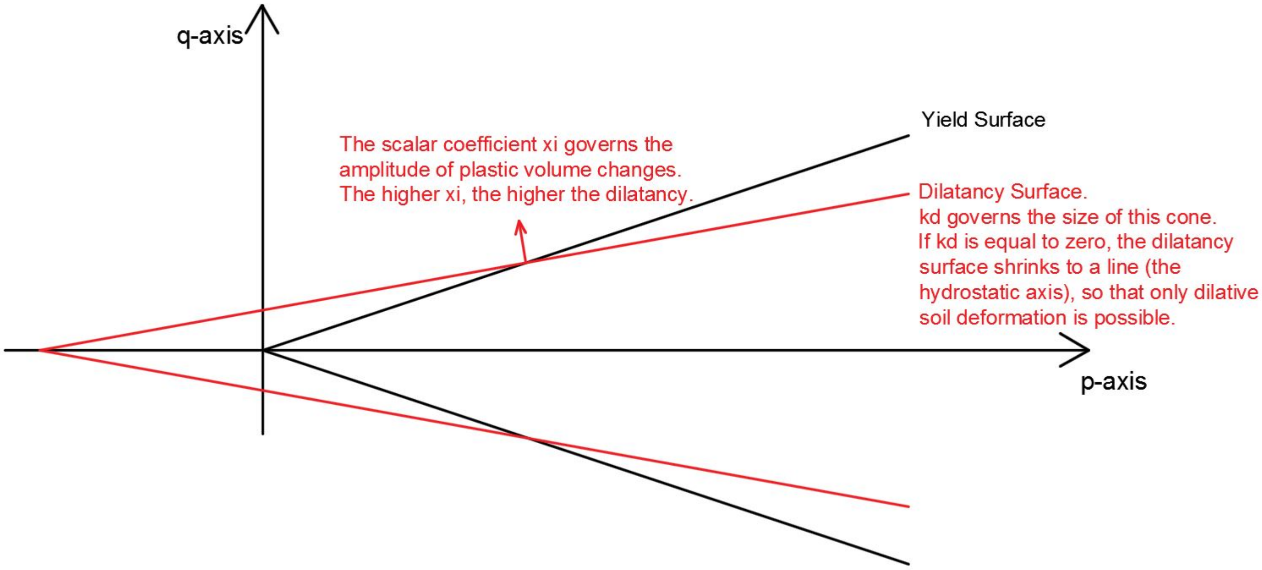


Figure 205.2: The physical meanings of ξ and k_d

The physical meanings of h_a, c_r, ξ, and k_d are shown in Figure (205.1) and Figure (205.2).

205.3.4.11 Modeling, Material Model: Hyperbolic Drucker-Prager Nonassociated Material Model with Linear Isotropic and/or Armstrong-Frederick Nonlinear Kinematic Hardening

This command defines a hyperbolic Drucker-Prager constitutive model, with nonlinear kinematic and/or linear isotropic hardening. This material uses non-associated plastic flow rule.

The command is:

```
add material # <.> type HyperbolicDruckerPragerNonAssociateArmstrongFrederick
   mass_density = <M/L^3>
   elastic_modulus = <F/L^2>
   poisson_ratio = <.>
   friction_angle = <.>
   cohesion = <F/L^2>
   rounded_distance = <F/L^2>
   armstrong_frederick_ha = <F/L^2>
   armstrong_frederick_cr = <.>
   isotropic_hardening_rate = <F/L^2>
   initial_confining_stress = <F/L^2>
   plastic_flow_xi = <.>
   plastic_flow_kd = <.>;
```

where:

- `mass_density` is the mass density of material $[M/L^3]$
- `elastic_modulus` is the elastic modulus of material $[F/L^2]$
- `poisson_ratio` is the Poisson's ratio for material [ ]
- `friction_angle` is the initial friction angle of the material. If isotropic hardening is present, friction angle will evolve. [rad]
- `cohesion` is a material constant that defines the cohesion of the material $[F/L^2]$
- `rounded_distance` is the parameter that controls the shape of the rounded apex of yield surface $[F/L^2]$
- `armstrong_frederick_ha` a kinematic hardening parameter, that governs the initial stiffness after the yield $[F/L^2]$
- `armstrong_frederick_cr` a kinematic hardening parameter. It is noted that ratio $\frac{h_a}{c_r}$ controls the asymptote the back-stress, that can be related to the ultimate shear strength [.]
- `isotropic_hardening_rate` is the rate of the isotropic hardening $[F/L^2]$

- `initial_confining_stress` initial confining (mean) pressure, a small value just get initial stress out of cone zone $[F/L^2]$
- `plastic_flow_xi` governs the amplitude of plastic volume changes - the higher ξ, the higher the dilatancy. If $\xi = 0$, the material model will only produce deviatoric plastic strains. [.]
- `plastic_flow_kd` governs the size of the dilatancy surface, a cone in the stress space on which no plastic volume changes occur. k_d governs the size of this cone: if k_d is equal to zero, the dilatancy surface shrinks to a line (the hydrostatic axis), so that only dilative soil deformation is possible. [.]

More on this material model can be found in Section 104.6.8.5 on Page 247 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.12 Modeling, Material Model: Rounded Mohr-Coulomb Associated Linear Isotropic Hardening Material Model

The command is:

```
add material # <.> type roundedMohrCoulomb
   mass_density = <M/L^3>
   elastic_modulus = <F/L^2>
   poisson_ratio = <.>
   RMC_m = <.>
   RMC_qa = <F/L^2>
   RMC_pc = <F/L^2>
   RMC_e = <.>
   RMC_eta0 = <.>
   RMC_Heta = <F/L^2>
   initial_confining_stress = <F/L^2>
```

where

- `mass_density` is the mass density of material $[M/L^3]$
- `elastic_modulus` is the elastic modulus of material $[F/L^2]$
- `poisson_ratio` is the Poisson's ratio material [ ]
- `RMC_m` $0 < m < 1$ parameter of the RMC yield function. Controls roundness of apex in p-q space. [ ]
- `RMC_qa` q_a parameter of the RMC yield function. Controls roundness of apex in p-q space. $[F/L^2]$
- `RMC_pc` p pressure offset $[F/L^2]$
- `RMC_e` e parameter controls roundness of the deviatoric cross-section of the yield surface. $0.5 < e <= 1$, $e = 0.5$ results in a triangular deviatoric section while $e = 1$ is round. [ ]
- `RMC_eta0` controls the opening of the yield surface [ ]
- `RMC_Heta` isotropic (linear) hardening of the yield surface $[F/L^2]$
- `initial_confining_stress` initial confining (mean) pressure $[F/L^2]$

More on this material model can be found in Section 104.6.9 on Page 250 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.13 Modeling, Material Model: Cam Clay Material Model

The command is:

```
add material # <.> type CamClay
  mass_density = <M/L^3>
  M = <.>
  lambda = <.>
  kappa = <.>
  e0 = <.>
  p0 = <F/L^2>
  poisson_ratio = <.>
  initial_confining_stress = <F/L^2>
```

where

- `mass_density` is the mass density of material $[M/L^3]$
- `e0` void ratio (e_0) at the reference pressure, [dimensionless]
- `M` Cam-Clay slope of the critical state line in stress space, [dimensionless]
- `lambda` Cam-Clay normal consolidation line slope, (unit: dimensionless)
- `kappa` Cam-Clay unload-reload line slope, (unit: dimensionless)
- `poisson_ratio` Constant Poisson-ratio
- `p0` Cam-Clay parameter (p_0). Tip of the yield surface in q-p space. $[F/L^2]$
- `initial_confining_stress` initial confining (mean) pressure $[F/L^2]$

More on this material model can be found in Section 104.6.10 on Page 251 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.14 Modeling, Material Model: von Mises Associated Multiple Yield Surface Material Model

The command is:

```
add material # <.> type vonMisesMultipleYieldSurface
  mass_density = <M/L^3>
  elastic_modulus = <F/L^2>
  poisson_ratio = <.>
  total_number_of_yield_surface = <.>
  radiuses_of_yield_surface = <string>
  radiuses_scale_unit = <F/L^2>
  hardening_parameters_of_yield_surfaces = <string>
  hardening_parameters_scale_unit = <F/L^2> ;
```

where

- `mass_density` is the mass density of material $[M/L^3]$
- `elastic_modulus` is the elastic modulus of the material $[F/L^2]$
- `poisson_ratio` is the constant Poisson-ratio [dimensionless]
- `total_number_of_yield_surface` is the total number of yield surfaces. [dimensionless]
- `radiuses_of_yield_surface` is the radius list of multiple yield surfaces. This parameter gives the radiuses of each yield surface from the smallest to the biggest. This parameter should be a string which contains the dimensionless radiuses. The radiuses should be separated by a blank space or a comma. [string]
- `radiuses_scale_unit` is the unit of the each yield surface. This parameter also provides a method to scale up or scale down the radiuses of each yield surfaces. $[F/L^2]$
- `hardening_parameters_of_yield_surfaces` is the hardening parameters corresponding to each yield surface. This parameter should be a string which contains the dimensionless hardening parameters. The hardening parameters should be separated by a blank space or a comma. [string]
- `hardening_parameters_scale_unit` The unit of the each hardening parameter. This parameter also provides a method to scale up or scale down the hardening parameter of each yield surfaces. $[F/L^2]$

205.3.4.15 Modeling, Material Model: von Mises Associated Multiple Yield Surface Material Model that Matches G/G_{max} Curves

The command is:

```
add material # <.> type vonMisesMultipleYieldSurfaceGoverGmax
  mass_density = <M/L^3>
  initial_shear_modulus = <F/L^2>
  poisson_ratio = <.>
  total_number_of_shear_modulus = <.>
  GoverGmax = <string>
  ShearStrainGamma = <string> ;
```

Command Example is

```
add material # 1 type vonMisesMultipleYieldSurfaceGoverGmax
   mass_density = 0.0*kg/m^3
   initial_shear_modulus = 3E8 * Pa
   poisson_ratio = 0.0
   total_number_of_shear_modulus = 9
   GoverGmax =
   "1,0.995,0.966,0.873,0.787,0.467,0.320,0.109,0.063"
   ShearStrainGamma =
   "0,1E-6,1E-5,5E-5,1E-4, 0.0005, 0.001, 0.005, 0.01";
```

where

- `mass_density` is the mass density of material $[M/L^3]$
- `initial_shear_modulus` is the initial maximum shear modulus, namely, the Gmax. $[F/L^2]$
- `poisson_ratio` is the constant Poisson-ratio. [dimensionless]
- `total_number_of_shear_modulus` is the total number of shear modulus, including the initial maximum shear modulus. The total number of yield surface is one less than the total number of shear modulus. Namely, (N+1) areas are divided by N surfaces. [dimensionless]
- `GoverGmax` is the G/Gmax from experiments, including the initial shear modulus. Namely, the first element should be 1.0. Each element is dimensionless. The input should be separated by a blank space or a comma. [string]
- `ShearStrainGamma` is the shear strain γ corresponding to the GoverGmax. Note that $\gamma = 2\varepsilon$ when the input is prepared. The first element should be 0.0 corresponding to the initial shear modulus. Each element is dimensionless. The input should be separated by a blank space or a comma. [string]

205.3.4.16 Modeling, Material Model: Drucker-Prager Nonassociated Multi-Yield Surface Material Model

The command is:

```
add material # <.> type DruckerPragerMultipleYieldSurface
  mass_density = <M/L^3>
  elastic_modulus = <F/L^2>
  poisson_ratio = <.>
  initial_confining_stress = <F/L^2>
  reference_pressure = <F/L^2>
  pressure_exponential_n = <.>
  cohesion = <F/L^2>
  dilation_angle_eta = <.>
  dilation_scale = <.>
  total_number_of_yield_surface = <.>
  sizes_of_yield_surfaces = <string>
  yield_surface_scale_unit = <F/L^2>
  hardening_parameters_of_yield_surfaces = <string>
  hardening_parameters_scale_unit = <F/L^2>;
```

where

- `mass_density` is the mass density of material $[M/L^3]$
- `elastic_modulus` is the elastic modulus of the material $[F/L^2]$
- `poisson_ratio` is the constant Poisson-ratio [dimensionless]
- `initial_confining_stress` is the initial confining (mean) pressure $[F/L^2]$
- `reference_pressure` is the reference pressure for the initial modulus. This parameter is usually 101kPa. $[F/L^2]$
- `pressure_exponential_n` is the exponential number of the pressure dependent modulus. [dimensionless]
- `cohesion` is the attraction force is the soil. $[F/L^2]$
- `dilation_angle_eta` controls the dilation and compaction of the material. When the stress ratio is smaller than this parameter, plastic compaction takes place. When the stress ratio is greater than this parameter, the plastic dilation takes place. [dimensionless]
- `dilation_scale` controls the rate of the dilation or compaction in the plastic flow. [dimensionless]
- `total_number_of_yield_surface` is the total number of yield surfaces. [dimensionless]

- `radiuses_of_yield_surface` is the radius list of multiple yield surfaces. This parameter gives the radiuses of each yield surface from the smallest to the biggest. This parameter should be a string which contains the dimensionless radiuses. The radiuses should be separated by a blank space or a comma. [string]
- `radiuses_scale_unit` is the unit of the each yield surface. This parameter also provides a method to scale up or scale down the radiuses of each yield surfaces. $[F/L^2]$
- `hardening_parameters_of_yield_surfaces` is the hardening parameters corresponding to each yield surface. This parameter should be a string which contains the dimensionless hardening parameters. The hardening parameters should be separated by a blank space or a comma. [string]
- `hardening_parameters_scale_unit` The unit of the each hardening parameter. This parameter also provides a method to scale up or scale down the hardening parameter of each yield surfaces. $[F/L^2]$

205.3.4.17 Modeling, Material Model: Drucker-Prager Nonassociated Material Model that Matches G/G_{max} Curves

The command is:

```
add material # <.> type DruckerPragerMultipleYieldSurfaceGoverGmax
  mass_density = <M/L^3>
  initial_shear_modulus = <F/L^2>
  poisson_ratio = <.>
  initial_confining_stress = <F/L^2>
  reference_pressure = <F/L^2>
  pressure_exponential_n = <.>
  cohesion = <F/L^2>
  dilation_angle_eta = <.>
  dilation_scale = <.>
  total_number_of_shear_modulus = <.>
  GoverGmax = <string>
  ShearStrainGamma = <string>
```

Command Example is

```
add material # 1 type DruckerPragerMultipleYieldSurfaceGoverGmax
  mass_density = 0.0*kg/m^3
  initial_shear_modulus = 3E8 * Pa
  poisson_ratio = 0.0
  initial_confining_stress = 1E5 * Pa
  reference_pressure = 1E5 * Pa
  pressure_exponential_n = 0.5
  cohesion = 0. * Pa
  dilation_angle_eta =1.0
  dilation_scale = 0.0
  total_number_of_shear_modulus = 9
  GoverGmax =
  "1,0.995,0.966,0.873,0.787,0.467,0.320,0.109,0.063"
  ShearStrainGamma =
  "0,1E-6,1E-5,5E-5,1E-4, 0.0005, 0.001, 0.005, 0.01";
```

where

- `mass_density` is the mass density of material $[M/L^3]$
- `elastic_modulus` is the elastic modulus of the material $[F/L^2]$
- `poisson_ratio` is the constant Poisson-ratio [dimensionless]
- `initial_confining_stress` is the initial confining (mean) pressure $[F/L^2]$
- `reference_pressure` is the reference pressure for the initial modulus. This parameter is usually 101kPa. $[F/L^2]$

- `pressure_exponential_n` is the exponential number of the pressure dependent modulus. [dimensionless]
- `cohesion` is the attraction force is the soil. $[F/L^2]$
- `dilation_angle_eta` controls the dilation and compaction of the material. When the stress ratio is smaller than this parameter, plastic compaction takes place. When the stress ratio is greater than this parameter, the plastic dilation takes place. [dimensionless]
- `dilation_scale` controls the rate of the dilation or compaction in the plastic flow. For this automatic G/Gmax match, the dilation scale has to be zero, which means only deviatoric plastic flow is allowed. If the users want to have volumetric dilation, they can match the G/Gmax manually with the other DruckerPragerMultipleYieldSurface command. [dimensionless]
- `total_number_of_shear_modulus` is the total number of shear modulus, including the initial maximum shear modulus. The total number of yield surface is one less than the total number of shear modulus. Namely, (N+1) areas are divided by N surfaces. [dimensionless]
- `GoverGmax` is the G/Gmax from experiments, including the initial shear modulus. Namely, the first element should be 1.0. Each element is dimensionless. The input should be separated by a blank space or a comma. [string]
- `ShearStrainGamma` is the shear strain γ corresponding to the GoverGmax. Note that $\gamma = 2\varepsilon$ when the input is prepared. The first element should be 0.0 corresponding to the initial shear modulus. Each element is dimensionless. The input should be separated by a blank space or a comma. [string]

205.3.4.18 Modeling, Material Model: Rounder Mohr-Coulomb Nonassociated Multi-Yield Surface Material Model

The command is:

```
add material # <.> type RoundedMohrCoulombMultipleYieldSurface
  mass_density = <M/L^3>
  elastic_modulus = <F/L^2>
  poisson_ratio = <.>
  initial_confining_stress = <F/L^2>
  reference_pressure = <F/L^2>
  pressure_exponential_n = <.>
  cohesion = <F/L^2>
  RMC_shape_k =<.>
  dilation_angle_eta = <.>
  dilation_scale = <.>
  total_number_of_yield_surface = <.>
  sizes_of_yield_surfaces = <string>
  yield_surface_scale_unit = <F/L^2>
  hardening_parameters_of_yield_surfaces = <string>
  hardening_parameters_scale_unit = <F/L^2>;
```

where

- `mass_density` is the mass density of material $[M/L^3]$
- `elastic_modulus` is the elastic modulus of the material $[F/L^2]$
- `poisson_ratio` is the constant Poisson-ratio [dimensionless]
- `initial_confining_stress` is the initial confining (mean) pressure $[F/L^2]$
- `reference_pressure` is the reference pressure for the initial modulus. This parameter is usually 101kPa. $[F/L^2]$
- `pressure_exponential_n` is the exponential number of the pressure dependent modulus. [dimensionless]
- `cohesion` is the attraction force is the soil. $[F/L^2]$
- `RMC_shape_k` controls the shape of the rounded Mohr-Coulomb yield surface. [dimensionless]
- `dilation_angle_eta` controls the dilation and compaction of the material. When the stress ratio is smaller than this parameter, plastic compaction takes place. When the stress ratio is greater than this parameter, the plastic dilation takes place. [dimensionless]
- `dilation_scale` controls the rate of the dilation or compaction in the plastic flow. [dimensionless]

- `total_number_of_yield_surface` is the total number of yield surfaces. [dimensionless]
- `radiuses_of_yield_surface` is the radius list of multiple yield surfaces. This parameter gives the radiuses of each yield surface from the smallest to the biggest. This parameter should be a string which contains the dimensionless radiuses. The radiuses should be separated by a blank space or a comma. [string]
- `radiuses_scale_unit` is the unit of the each yield surface. This parameter also provides a method to scale up or scale down the radiuses of each yield surfaces. $[F/L^2]$
- `hardening_parameters_of_yield_surfaces` is the hardening parameters corresponding to each yield surface. This parameter should be a string which contains the dimensionless hardening parameters. The hardening parameters should be separated by a blank space or a comma. [string]
- `hardening_parameters_scale_unit` The unit of the each hardening parameter. This parameter also provides a method to scale up or scale down the hardening parameter of each yield surfaces. $[F/L^2]$

205.3.4.19 Modeling, Material Model: Tsinghhua Liquefaction Material Model

The command is:

```
add material # <.> type TsinghuaLiquefactionModel
  mass_density = <M/L^3>
  poisson_ratio = <.>
  initial_confining_stress = <F/L^2>
  liquefaction_G0 = <.>
  liquefaction_EXPN = <.>
  liquefaction_c_h0 = <.>
  liquefaction_mfc = <.>
  liquefaction_mdc = <.>
  liquefaction_dre1 = <.>
  liquefaction_Dre2 = <.>
  liquefaction_Dir = <.>
  liquefaction_Alpha = <.>
  liquefaction_gamar = <.>
  liquefaction_pa = <.>
  liquefaction_pmin = <.>
```

Command Example is

```
add material # 1 type TsinghuaLiquefactionModel
  mass_density = 0.0*kg/m^3
  poisson_ratio = 0.1
  initial_confining_stress = 1E5 *Pa
  liquefaction_G0 = 800
  liquefaction_EXPN = 0.5
  liquefaction_c_h0 = 1.0
  liquefaction_mfc = 1.2
  liquefaction_mdc = 0.4
  liquefaction_dre1 = 0.5
  liquefaction_Dre2 = 1500
  liquefaction_Dir = 0.1
  liquefaction_Alpha = 0.01
  liquefaction_gamar = 0.01
  liquefaction_pa = 1E5
  liquefaction_pmin = 100 ;
```

where

- `mass_density` is the mass density of material $[M/L^3]$
- `poisson_ratio` is the constant Poisson ratio [dimensionless]
- `initial_confining_stress` is the initial confining (mean) pressure $[F/L^2]$
- `liquefaction_G0` is initial modulus scale at the reference pressure. For medium dense soil, G0 is 800. [dimensionless]

- `liquefaction_EXPN` is the exponential number of the pressure dependent modulus. [dimensionless]

- `liquefaction_c_h0` is the plastic modulus coefficient. This parameter should be determined by the G/Gmax curve. When the G/Gmax curve is hyperbolic, h is 1.2. The range of h is 0.7-1.2 [dimensionless].

- `liquefaction_mfc` is the slope of the failure surface in p-q plane. The range of $M_{f,c}$ is 1.4-1.8 [dimensionless].

- `liquefaction_mdc` is the slope of the phase transition surface in p-q plane. The range of $M_{d,c}$ is 0.3-1.0 [dimensionless].

- `liquefaction_dre1` is the accumulation coefficient of the reversible dilatancy. This parameter is usually 0.4 [dimensionless].

- `liquefaction_Dre2` is the release coefficient of the reversible dilatancy. This range of $d_{re,2}$ is 1000-1500 [dimensionless].

- `liquefaction_Dir` is the coefficient of irreversible dilatancy. The parameter d_{ir} controls the initial slope of the irreversible strain development with respect to the number of reversible loadings. Intuitively, when d_{ir} is bigger, the soil becomes liquefaction faster. The parameter d_{ir} can be around 0.2 [dimensionless].

- `liquefaction_Alpha` is the limit of the irreversible strain. Intuitively, α controls the maximum strain after the liquefaction. The parameter α can be around 0.03 [dimensionless].

- `liquefaction_gamar` is the maximum shear strain length in one liquefaction loading. Intuitively, this parameter controls the maximum strain size of one loop. This parameter can be around 0.05 [dimensionless].

- `liquefaction_pa` is the reference pressure. Usually, this parameter is 10000 [dimensionless].

- `liquefaction_pmin` is the minimum pressure in the calculation. If the pressure is smaller than p_{min} during the calculation, the pressure will be set to p_{min}. This parameter can be 1. Increasing this parameter can avoid the potential numerical errors on small numbers [dimensionless].

205.3.4.20 Modeling, Material Model: SANISand Material Model, version 2004

The command is:

```
add material # <.> type sanisand2004
  mass_density = <M/L^3>
  e0 = <.>
  sanisand2004_G0 = <.>
  poisson_ratio = <.>
  sanisand2004_Pat = <.>
  sanisand2004_p_cut = <.>
  sanisand2004_Mc = <.>
  sanisand2004_c = <.>
  sanisand2004_lambda_c = <.>
  sanisand2004_xi = <.>
  sanisand2004_ec_ref = <.>
  sanisand2004_m = <.>
  sanisand2004_h0 = <.>
  sanisand2004_ch = <.>
  sanisand2004_nb = <.>
  sanisand2004_A0 = <.>
  sanisand2004_nd = <.>
  sanisand2004_z_max = <.>
  sanisand2004_cz = <.>
  initial_confining_stress = <F/L^2>;
```

where

- `MaterialNumber`: Material tag
- `mass_density` is the mass density of material $[M/L^3]$
- `sanisand2004_e0` initial void ratio [ ]
- `sanisand2004_G0` normalized elastic shear modulus [ ]
- `poisson_ratio` Poisson's ratio [ ]
- `sanisand2004_Pat` atmospheric pressure $[F/L^2]$
- `sanisand2004_p_cut` pressure cut-off ratio $[F/L^2]$
- `sanisand2004_Mc` Critical stress ratio at triaxial compression [ ]
- `sanisand2004_c` tension-compression strength ratio $c = M_e/M_c$ [ ]
- `sanisand2004_lambda_c` parameter for critical state line [ ]
- `sanisand2004_xi` parameter for critical state line [ ]

- `sanisand2004_ec_ref` reference void for critical state line [ ]
- `sanisand2004_m` opening of the yield surface [ ]
- `sanisand2004_h0` bounding surface parameter [ ]
- `sanisand2004_ch` bounding surface parameter [ ]
- `sanisand2004_nb` bounding surface parameter [ ]
- `sanisand2004_A0` dilatancy parameter [ ]
- `sanisand2004_nd` dilatancy parameter [ ]
- `sanisand2004_z_max` maximum z fabric parameter [ ]
- `sanisand2004_cz` fabric hardening parameter [
- `initial_confining_stress` is the initial confining stress $p = -1/3\sigma_{ii}$ and it is positive in compressions (since there is that – (minus) sign in front of sum of normal stresses (σ_{ii} indicial notation summation convention applies) that are positive in tension [stress].

More on this material model can be found in section 104.6.11 on Page 255 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

Important note: This material model should be used together with explicit constitutive algorithms, e.g. `Forward_Euler` or `Forward_Euler_Subincrement`. For better result, it is suggested to apply strain increments, or sub-increments, smaller than 1e-4.

205.3.4.21 Modeling, Material Model: SANISand Material Model, version 2008

The command is:

```
add material # <.> type sanisand2008
    mass_density = <M/L^3>
    e0 = <.>
    sanisand2008_G0 = <.>
    sanisand2008_K0 = <.>
    sanisand2008_Pat = <.>
    sanisand2008_k_c = <.>
    sanisand2008_alpha_cc = <.>
    sanisand2008_c = <.>
    sanisand2008_lambda = <.>
    sanisand2008_ec_ref = <.>
    sanisand2008_m = <.>
    sanisand2008_h0 = <.>
    sanisand2008_ch = <.>
    sanisand2008_nb = <.>
    sanisand2008_A0 = <.>
    sanisand2008_nd = <.>
    sanisand2008_p_r = <.>
    sanisand2008_rho_c = <.>
    sanisand2008_theta_c = <.>
    sanisand2008_X = <.>
    sanisand2008_z_max = <.>
    sanisand2008_cz = <.>
    sanisand2008_p0 = <F/L^3>
    sanisand2008_p_in = <F/L^3>
    algorithm = explicit (or) implicit
    number_of_subincrements = <.>
    maximum_number_of_iterations = <.>
    tolerance_1 = <.>
    tolerance_2 = <.>;
```

where

- `MaterialNumber`: Number of the ND material to be used ;
- `Algorithm`: Explicit (=0) or Implicit (=1) ;
- `rho`: density ;
- `e0`: initial void ratio at zero strain ;
- `G0`: Reference elastic shear modulus [stress];
- `K0`: Reference elastic bulk modulus [stress];
- `sanisand2008_Pat`: atmospheric pressure for critical state line ;

- `sanisand2008_k_c`: cut-off factor; for $p < k_c P_{at}$, use $p = k_c P_{at}$ for calculation of G; (a default value of $k_c = 0.01$ should work fine) ;
- `sanisand2008_alpha_cc`: critical state stress ratio ;
- `sanisand2008_c`: tension-compression strength ratio ;
- `sanisand2008_lambda`: parameter for critical state line ;
- `sanisand2008_xi`: parameter for critical state line ;
- `sanisand2008_ec_ref`: reference void for critical state line, ; $e_c = e_r lambda (p_c/P_{at})^x i$;
- `sanisand2008_m`: opening of the yield surface ;
- `sanisand2008_h0`: bounding surface parameter ;
- `sanisand2008_ch`: bounding surface parameter ;
- `sanisand2008_nb`: bounding surface parameter ;
- `sanisand2008_A0`: dilatancy parameter ;
- `sanisand2008_nd`: dilatancy parameter ;
- `sanisand2008_p_r`: LCC parameter ;
- `sanisand2008_rho_c`: LCC parameter ;
- `sanisand2008_theta_c`: LCC parameter ;
- `sanisand2008_X`: LCC parameter ;
- `sanisand2008_z_max`: fabric parameter ;
- `sanisand2008_cz`: fabric parameter ;
- `sanisand2008_p0`: yield surface size ;
- `sanisand2008_p_in` ;
- `number_of_subincrements` number of subincrements in constitutive simulation
- `maximum_number_of_iterations` maximum number of iterations

- `tolerance_1` Explicit: tolerance for intersection point (distance between two consecutive points) Implicit: yield function tolerance
- `tolerance_2` Implicit: residual tolerance

More on this material model can be found in Section 104.6.12 on Page 262 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.22 Modeling, Material Model: Cosserat Linear Elastic Material Model

The command is:

```
add material # <.> type Cosserat_linear_elastic_isotropic_3d
    mass_density = <M/L^3>
    lambda = <F/L^2>
    mu = <F/L^2>
    chi = <F/L^2>
    pi1 = <F>
    pi2 = <F>
    pi3 = <F>
    ;
```

- `MaterialNumber` unique material Number.
- `mass_density` the density of the material.
- `lambda`, `mu`, `chi`, `pi1`, `pi2`, `pi3` are the 6 Cosserat elastic constants (Eringen, 2012).

The relations between elastic constants is as follows Eringen (2012). Note the Young's modulus and the Poisson's ratio are different from the classical elasticity:

- Young's modulus $E = (2\mu + \chi)(3\lambda + 2\mu + \chi)$.
- Shear modulus $G = \mu + 1/2\chi$.
- Poisson's ratio $\nu = \lambda/(2\lambda + 2\mu + \chi)$.
- Characteristic length for torsion $l_t = ((\pi_2 + \pi_3)/(2\mu + \chi))^{1/2}$.
- Characteristic length for bending $l_b = (\pi_3/2(2\mu + \chi))^{1/2}$.
- Coupling number $N = (\chi/2(\mu + \chi))$
- Polar ratio $\Phi = (\pi_2 + \pi_3)/(\pi_1 + \pi_2 + \pi_3)$

According to Eringen Eringen (2012), the 6 elastic constants should satisfy the following conditions

$$\begin{aligned} 3\lambda + 2\mu + \chi \geq 0 \ , \quad & 2\mu + \chi \geq 0 \ , \quad & \chi \geq 0 \ , \\ 3\pi_1 + \pi_2 + \pi_3 \geq 0 \ , \quad & \pi_3 + \pi_2 \geq 0 \ , \quad & \pi_3 - \pi_2 \geq 0 \ . \end{aligned} \tag{205.1}$$

205.3.4.23 Modeling, Material Model: von Mises Cosserat Material Model

The command is:

```
add material # <.> type Cosserat_von_Mises
    mass_density = <M/L^3>
    lambda = <F/L^2>
    mu = <F/L^2>
    chi = <F/L^2>
    pi1 = <F>
    pi2 = <F>
    pi3 = <F>
    plastic_internal_length = <L>
    von_mises_radius = <F/L^2>
    isotropic_hardening_rate = <F/L^2>
    ;
```

- `MaterialNumber` unique material Number.
- `mass_density` the density of the material.
- `lambda`, `mu`, `chi`, `pi1`, `pi2`, `pi3` are the 6 Cosserat elastic constantsEringen (2012).
- `plastic_internal_length` is the characteristic length in the plasticity.
- `von_mises_radius` is radius of the unified yield surface of force-stress and couple-stress.
- `isotropic_hardening_rate` is the rate of isotropic hardening.

205.3.4.24 Modeling, Material Model: Uniaxial Linear Elastic, Fiber Material Model

The command is:

```
add material # <.> type uniaxial_elastic
    elastic_modulus = <F/L^2>
    viscoelastic_modulus = <mass / length / time> ;
```

where

- `MaterialNumber` unique material Number.
- `elastic_modulus` elastic modulus of the material.
- `viscoelastic_modulus` damping tangent.

More on this material model can be found in Section ?? on Page ?? in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

As the name implies, `uniaxiale_elastic` material model works with uniaxial element only. For 3D elements, for example solid brick elements, please use 3D material models, for example, `linear_elastic_isotropic_3d`

205.3.4.25 Modeling, Material Model: Stochastic Uniaxial Linear Elastic Model

The command is:

```
add material # <.> type stochastic_uniaxial_elastic uncertain_elastic_modulus = ↩
    random variable # <.> elastic_modulus_scale_unit = <F/L^2>;
```

where

- `uncertain_elastic_modulus` specify uncertain elastic modulus of the material through a defined random variable.
- `elastic_modulus_scale_unit` specify the unit scale factor that would be multiplied with the polynomial chaos coefficients of the random variable.

As the name implies, `stochastic_uniaxial_elastic` material model works with stochastic uniaxial element only.

For example:

```
add material # 1 type stochastic_uniaxial_elastic uncertain_elastic_modulus = ↩
    random variable # 1 elastic_modulus_scale_unit = 1*Pa;
```

Add material #1 as `stochastic_uniaxial_elastic` material with uncertain elastic modulus characterized by the polynomial chaos coefficients of random variable 1 and scale factor $1 * Pa$.

205.3.4.26 Modeling, Material Model: Stochastic Uniaxial Nonlinear Armstrong Frederick Model

The command is:

```
add material # <.> type stochastic_uniaxial_Armstrong_Frederick
constitutive triple product # <.>
armstrong_frederick_ha = random variable # <.>
armstrong_frederick_ha_scale_unit = <F/L^2>
armstrong_frederick_cr = random variable # <.>
```

or

```
add material # <.> type stochastic_uniaxial_Armstrong_Frederick
constitutive triple product # <.>
armstrong_frederick_ha = random variable # <.>
armstrong_frederick_ha_scale_unit = <F/L^2>
armstrong_frederick_cr = random variable # <.>
polynomial_chaos_terms_ha = <.>
polynomial_chaos_terms_cr = <.>
polynomial_chaos_terms_incremental_strain = <.>;
```

Note that the difference between these two commands is that the first command would by default use the full polynomial chaos (PC) bases defined in the provided `constitutive triple product` for probabilistic constitutive modeling. The second command would support user-specified number of polynomial chaos terms for uncertain `armstrong_frederick_ha`, `armstrong_frederick_cr` and `incremental_strain`. This enables users to perform truncation of PC bases for probabilistic constitutive modeling.

The command input parameters are:

- `constitutive triple product #` specifies the ID of the triple product, that would be used in probabilistic constitutive updating. In stochastic finite element method (FEM), the first and second PC basis for this triple product should come from the joint PC representation of uncertain parameters `armstrong_frederick_ha` and `armstrong_frederick_cr`. The third PC basis for this triple product should come from the PC representation of uncertain FEM system response, e.g., uncertain structural displacement.
- `armstrong_frederick_ha = random variable #` specifies the uncertain Armstrong Frederick parameter ha through a defined random variable.
- `armstrong_frederick_ha_scale_unit` specifies the unit scale factor that would be multiplied with the polynomial chaos coefficients of the random variable of uncertain Armstrong Frederick parameter ha.

- `armstrong_frederick_cr = random variable #` specifies the uncertain Armstrong Frederick parameter cr through a defined random variable.
- `polynomial_chaos_terms_ha` specifies the number of polynomial chaos basis of uncertain ha involved in the probabilistic constitutive updating.
- `polynomial_chaos_terms_cr` specifies the number of polynomial chaos basis of uncertain cr involved in the probabilistic constitutive updating.
- `polynomial_chaos_terms_incremental_strain` specifies the number of polynomial chaos basis of uncertain incremental strain $d\epsilon$ involved in the probabilistic constitutive updating.

As the name implies, `stochastic_uniaxial_Armstrong_Frederick` material model works with stochastic uniaxial element only.

For example:

```
add material # 1 type stochastic_uniaxial_Armstrong_Frederick
constitutive triple product # 1
armstrong_frederick_ha = random variable # 1
armstrong_frederick_ha_scale_unit = 1*Pa
armstrong_frederick_cr = random variable # 2
polynomial_chaos_terms_ha = 10
polynomial_chaos_terms_cr = 10
polynomial_chaos_terms_incremental_strain = 30;
```

Add material # 1 as `stochastic_uniaxial_Armstrong_Frederick` material with triple product # 1 for probabilistic constitutive updating.

Uncertain parameter ha is characterized by random variable # 1 using scale unit $1*$Pa.

Uncertain parameter cr is characterized by random variable # 2. The number of polynomial chaos basis for uncertain parameters ha, cr and incremental strain in probabilistic constitutive updating are 10, 10 and 30, respectively.

205.3.4.27 Modeling, Material Model: Uniaxial Nonlinear Concrete, Fiber Material Model, version 02

The command is:

```
add material # <.> type uniaxial_concrete02
    compressive_strength = <F/L^2>
    strain_at_compressive_strength = <.>
    crushing_strength = <F/L^2>
    strain_at_crushing_strength = <.>
    lambda = <.>
    tensile_strength = <F/L^2>
    tension_softening_stiffness = <F/L^2>;
```

- `compressive_strength` compressive strength.
- `strain_at_compressive_strength` strain at compressive strength.
- `crushing_strength` crushing strength.
- `strain_at_crushing_strength` strain at crushing strength.
- `lambda` ratio between unloading slope at epscu and initial slope.
- `tensile_strength` tensile strength.
- `tension_softening_stiffness` tension softening stiffness (absolute value) (slope of the tension softening branch).

More on this material model can be found in Section ?? on Page ?? in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.28 Modeling, Material Model: Faria-Oliver-Cervera Concrete Material

The command is:

```
add material No (or #) <material_number>
    type FariaOliverCerveraConcrete
    elastic_modulus = <F/L^2>
    poisson_ratio = <.>
    tensile_yield_strength = <F/L^2>
    compressive_yield_strength = <F/L^2>
    plastic_deformation_rate = <.>
    damage_parameter_Ap = <.>
    damage_parameter_An = <.>
    damage_parameter_Bn = <.>
```

where

- `No (or #)<material_number>` is a unique material integer number (does not have to be sequential, any unique positive integer number can be used).
- `type FariaOliverCerveraConcrete` is the material type.
- `elastic_modulus` is the elastic modulus of material $[F/L^2]$
- `poisson_ratio` is the Poisson's ratio material.
- `tensile_yield_strength` is the tensile yield strength $[F/L^2]$
- `compressive_yield_strength` is the compressive yield strength $[F/L^2]$

205.3.4.29 Modeling, Material Model: Plane Stress Layered Material

The command is:

```
add material No (or #) <element_number>
    type PlaneStressLayeredMaterial
    number_of_layers = <.>
    thickness_array = <string>
    thickness_scale_unit = <L>
    with material # <string>
    ;
```

where

- `No (or #)<material_number>` is a unique material integer number (does not have to be sequential, any unique positive integer number can be used).
- `type PlaneStressLayeredMaterial` is the material type.
- `number_of_layers` is the number of layers in this layered material. For reinforced concrete wall element, this will be just 3 layers, inside/confined concrete, reinforcement, and outside/unconfined concrete.
- `thickness_array` is the thickness ratio of each individual material.
- `thickness_scale_unit` set the length unit and the scale factor for the thickness of the layered material.
- `material # <string>` is the string of predefined individual material tags.

205.3.4.30 Modeling, Material Model: Uniaxial Nonlinear Steel, Fiber Material Model, version 01

The command is:

```
add material # <.> type uniaxial_steel01
   yield_strength = <F/L^2>
   elastic_modulus = <F/L^2>
   strain_hardening_ratio = <.>
   a1 = <.>
   a2 = <.>
   a3 = <>
   a4 = <.> ;
```

- `yield_strength` yield strength.
- `elastic_modulus` initial elastic tangent.
- `strain_hardening_ratio` strain-hardening ratio (ratio between post-yield tangent and initial elastic tangent).
- `a1, a2, a3, a4` isotropic hardening parameters
  - a1: isotropic hardening parameter, increase of compression yield envelope as proportion of yield strength after a plastic strain of a2*(fy/Ep). ;
  - a2: isotropic hardening parameter (see explanation under a1) ;
  - a3: isotropic hardening parameter, increase of tension yield envelope as proportion of yield strength after a plastic strain of a4*(fy/Ep) ;
  - a4: isotropic hardening parameter (see explanation under a3) ;

More on this material model can be found in Section ?? on Page ?? in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.31 Modeling, Material Model: Uniaxial Nonlinear Steel, Fiber Material Model, version 02

The command is:

```
add material # <.> type uniaxial_steel02
    yield_strength = <F/L^2>
    elastic_modulus = <F/L^2>
    strain_hardening_ratio = <.>
    R0 = <.>
    cR1 = <.>
    cR2 = <.>
    a1 = <.>
    a2 = <.>
    a3 = <>
    a4 = <.> ;
```

- `yield_strength`: yield strength ;
- `elastic_modulus`: initial elastic tangent ;
- `strain_hardening_ratio`: strain-hardening ratio (ratio between post-yield tangent and initial elastic tangent) ;
- `R0, cR1, cR2`: control the transition from elastic to plastic branches. Recommended values: R0=between 10 and 20, cR1=0.925, cR2=0.15 ;
- `a1, a2, a3, a4`: isotropic hardening parameters ;
  - a1: isotropic hardening parameter, increase of compression yield envelope as proportion of yield strength after a plastic strain of a2*(Fy/E). ;
  - a2: isotropic hardening parameter (see explanation under a1) ;
  - a3: isotropic hardening parameter, increase of tension yield envelope as proportion of yield strength after a plastic strain of a4*(Fy/E) ;
  - a4: isotropic hardening parameter (see explanation under a3) ;

More on this material model can be found in Section ?? on Page ?? in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.32 Modeling, Material Model: Plane Stress Plastic Damage Concrete Material

This is a plane stress version of the plastic damage concrete model developed by Faria et al. (1998). This material was implemented as part of the endeavor to model reinforced concrete shells, plates and shear walls. It should only be used together with Inelastic Layered Shell Section and 4 Node Shell NLDKGQ/Xin-Zheng-Lu, see page 915.

The command is:

```
add material No (or #) <material_number> type PlasticDamageConcretePlaneStress
    elastic_modulus = <F/L^2>
    poisson_ratio = <.>
    tensile_yield_strength = <F/L^2>
    compressive_yield_strength = <F/L^2>
    plastic_deformation_rate = <.>
    damage_parameter_Ap = <.>
    damage_parameter_An = <.>
    damage_parameter_Bn = <.>
```

where

- `No (or #)<material_number>` is a unique material integer number (does not have to be sequential, any unique positive integer number can be used).
- `elastic_modulus` is the elastic modulus of material $[F/L^2]$
- `poisson_ratio` is the Poisson's ratio material.
- `tensile_yield_strength` is the tensile yield strength $[F/L^2]$
- `compressive_yield_strength` is the compressive yield strength $[F/L^2]$
- `plastic_deformation_rate` governs the post-yield hardening modulus in the effective (undamaged) space and the plastic strain rate
- `damage_parameter_Ap` governs the tensile fracture energy and affects the ductility of the tensile response
- `damage_parameter_An` governs the softening behavior of concrete in compression, it changes the ductility but does not alter the peak strength
- `damage_parameter_Bn` governs the softening behavior of concrete in compression, it changes both the ductility and the peak strength

205.3.4.33 Modeling, Material Model: Plane Stress Rebar Material

This is a plane stress version of the Uniaxial Nonlinear Steel material. This material was implemented as part of the endeavor to model reinforced concrete shells, plates and shear walls. This model should be used together with Inelastic Layered Shell Section and 4 Node Shell NLDKGQ/Xin-Zheng-Lu, see page 915.

The command is:

```
add material No (or #) <material_number> type PlaneStressRebarMaterial
    with uniaxial_material # <.>
    angle = <degree> ;
```

where

- `No (or #)<material_number>` is a unique material integer number (does not have to be sequential, any unique positive integer number can be used).
- `with uniaxial_material #` is the material tag of predefined uniaxial steel material
- `angle` is the angle of uniaxial steel rebars. The angle is 0 along the direction formed by the first two nodes of a 4 Node Shell element.

205.3.4.34 Modeling, Nodes: Adding Nodes

Nodes can be added to the finite element model.

The command is:

```
 add node # <.> at (<L>,<L>,<L>) with <.> dofs;
```

For example:

```
add node No 1 at (1.0*m, 2.5*m, 3.33*m) with 3 dofs;
```

adds a node number 1 at coordinates $x = 1.0m$, $y = 2.5m$ and $z = 3.33m$ with 3 dofs. The nodes can be of 3dofs $[u_x, u_y, u_z]$, 4dofs $[]u_x, u_y, u_z, p]$ (u-p elements) , 6dofs $[u_x, u_y, u_z, r_x, r_y, r_z]$ (beams and shells) and 7 dofs $[u_x, u_y, u_z, p, U_x, U_y, U_z]$ (upU element) types. Description of output for nodes of different dof types can be found in section 206.6

205.3.4.35 Modeling, Nodes: Adding Stochastic Nodes

Nodes can be added to the stochastic finite element model. Different from deterministic finite element analysis, nodes in stochastic FEM should specify the number of polynomial chaos (PC) terms for each physical nodal degree of freedom (dof).

The command is:

```
add node # <.> at (<L>,<L>,<L>) with <.> dofs polynomial_chaos_terms = <.>;
```

Where:

- `polynomial_chaos_terms` specifies the number of polynomial chaos terms for each physical nodal dof.

The stochastic nodes can also be added as:

```
add node # <.> at (<L>,<L>,<L>) with <.> dofs polynomial_chaos_terms as random ↩
   field # <.>;
```

Which specifies the number of polynomial chaos terms for each physical nodal dof using the number of Hermite PC basis of a defined random field.

For example:

```
add node # 1 at (1.0*m, 0.0*m, 0.0*m) with 3 dofs polynomial_chaos_terms = 10;
```

Add a node # 1 at coordinates $x = 1.0m$, $y = 0.0m$ and $z = 0.0m$ with 3 physical dofs. For each physical dof, the number of terms for polynomial chaos expansion is 10.

```
add node # 1 at (1.0*m, 0.0*m, 0.0*m) with 3 dofs polynomial_chaos_terms as ↩
   random field # 2;
```

Add a node # 1 at coordinates $x = 1.0m$, $y = 0.0m$ and $z = 0.0m$ with 3 physical dofs. For each physical dof, the number of terms for polynomial chaos expansion is equal to the number of PC basis of random field # 2.

205.3.4.36 Modeling, Nodes: Define Nodal Physical Group

Physical Group for nodes can be defined as well.

The command is:

```
define physical_node_group "string";
```

For example:

```
define physical_node_group "my_new_node_group";
```

this would create a new physical_node_group with name "my_new_node_group".

Description of output for physical groups can be found in section 206.5.5

205.3.4.37 Modeling, Nodes: Adding Nodes to Nodal Physical Group

Already created nodes can be added to the (any) physical_node_group.

The command is:

```
add nodes (<.>,<.>,...) to physical_node_group "string";
```

For example:

```
add nodes (1,2,3) to physical_node_group "my_new_node_group";
```

this would add node tag (1,2 and 3) to already created physical_node_group "my_new_node_group". Please note that the nodes (1,2 and 3) must be added to the model before they are added to the physical_node_group.

Description of output for physical groups can be found in section 206.5.5

205.3.4.38 Modeling, Nodes: Removing Nodal Physical Group

Already defined node physical group physical_node_group can be removed.

The command is

```
remove physical_node_group "string";
```

For example:

```
remove physical_node_group "my_new_node_group";
```

this would delete the physical_node_group "my_new_node_group".

205.3.4.39 Modeling, Nodes: Print Nodal Physical Group

Printing already defined nodal physical grouop physical_node_group is possible too.

The command is:

```
print physical_node_group "string";
```

For example:

```
print physical_node_group "my_new_node_group";
```

this would print the information about physical_node_group "my_new_node_group".

```
PHYSICAL_NODE_GROUP my_new_node_group
[1 2 3]
```

205.3.4.40 Modeling, Nodes: Removing Nodes

Nodes can be removed from the finite element model, for example during excavation, removal of finite elements.

The command is:

```
remove node No (or #) <.>;
```

For example:

```
remove node # 1;
```

205.3.4.41 Modeling, Nodes: Adding Nodal Mass, for 3DOFs and/or 6DOFs

Nodal mass can be added to nodes with 3 DOFs and/or 6DOFs. This is in addition to nodal mass that is obtained from finite elements.

The command for 3DOFs nodes (truss, solids, wall) is:

```
add mass to node # <.>
mx = <M>
my = <M>
mz = <M>;
```

Simularly, the command for 6DOFs nodes (beams and shells) is:

```
add mass to node # <.>
mx = <M>
my = <M>
mz = <M>
Imx = <M*L^2>
Imy = <M*L^2>
Imz = <M*L^2>;
```

205.3.4.42 Modeling, Finite Element: Adding Finite Elements

The basic structure for adding any finite element is:

```
add element No (or #)
   type <finite_element_type>
   with nodes (<.>, ..., <.>)|
   {element dependent parameters};
```

Choices for `finite_element_type` are listed below

205.3.4.43 Modeling, Finite Element: Define Finite Element Physical Group

Physical group for finite elements can be defined.

The command is:

```
define physical_element_group "string";
```

For example:

```
define physical_element_group "my_new_element_group";
```

this would create a new physical_element_group with name "my_new_element_group".

Description of output for physical groups can be found in Section 206.5.5.

205.3.4.44 Modeling, Finite Element: Adding Elements to Physical Element Group

Finite elements, that already exist in the finite element domain, can be added to the physical_element_group.

The command is:

```
add elements (<.>,<.>,...) to physical_node_group "string";
```

For example:

```
add elements (1,2,3) to physical_node_group "my_new_node_group";
```

this would add elements with tags/numbers (1,2 and 3) to already created physical_element_group "my_new_element_group". Please note that the elements (1,2 and 3) must be added to the model before they are added to the physical_element_group.

Description of output for physical groups can be found in Section 206.5.5.

205.3.4.45 Modeling, Finite Element: Remove Physical Finite Element Group

Finite elements can also be removed from the physical_element_group.

The command is:

```
remove physical_element_group "string";
```

For example:

```
remove physical_element_group "my_new_element_group";
```

this would delete the physical_element_group "my_new_element_group".

205.3.4.46 Modeling, Finite Element: Print Physical Finite Element Group

Details of the physical_element_group can be printed.

The commands is:

```
print physical_element_group "string";
```

For example:

```
print physical_element_group "my_new_element_group";
```

this would print the information about physical_element_group "my_new_element_group".

```
PHYSICAL_ELEMENT_GROUP my_new_element_group
[1 2 3]
```

205.3.4.47 Modeling, Finite Element: Remove Finite Element

Finite elements can be removed, for example if modeling requires excavation, removal of finite elements and nodes.

The command is:

```
remove element # <.>;
```

For example,

```
remove element # 1;
```

205.3.4.48 Modeling, Finite Element: Truss Element

The command is:

```
add element No (or #) <element_number> type truss
            with nodes (n1, n2)
            use material No (or #) <material_number>
            section_area <section_area> [unit];
            mass_density <mass_density> [unit];
```

where

- `No (or #)<element_number>` is a unique element integer number (does not have to be sequential, any unique positive integer number can be used)
- `type truss` is the element type
- `with nodes (n1, n2)` are the 2 nodes (node numbers) defining this element
- `use material No (or #)` is the material number which makes up the element. Material has to be a uniaxial material, and it can be either elastic or one of the elastic-plastic materials defined for uniaxial behavior.
- `section_area` is the cross section area $[L^2]$

Description of output by this element can be found in Section 206.8.1. more on this finite element can be found in Section 102.6 on Page 126 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.49 Modeling, Finite Element: Kelvin-Voigt Element

The command is:

```
add element # <.> type Kelvin_Voigt
    with nodes (<.>, <.>)
    axial_stiffness = <F/L>
    axial_viscous_damping = <F/L*T>;
```

where

- `No (or #)<element_number>` is a unique element integer number, that does not have to be sequential, any unique positive integer number can be used
- `type Kelvin_Voigt` is the element type
- `with nodes (n1, n2)` are the 2 nodes (node numbers) defining this element
- `axial_stiffness` represents the stiffness in the axial direction, $[F/L]$
- `axial_viscous_damping` represents the viscosity, or viscous damping coefficient, in the axial direction, $[F/L*T]$

Note: Nodes defining this element cannot be at the same location, that is, this is a two node element and direction of this element is calculated from two distinct locations/coordinates of nodes.

205.3.4.50 Modeling, Finite Element: Inerter Element

The command is:

```
add element # <.> type Inerter
    with nodes (<.>, <.>)
    inertance = <M>;
```

where

- `No (or #)<element_number>` is a unique element integer number, that does not have to be sequential, any unique positive integer number can be used
- `type Inerter` is the element type
- `with nodes (n1, n2)` are the 2 nodes (node numbers) defining this element
- `inertance` represents the inertance in the axial direction, $[M]$

Note: Nodes defining this element cannot be at the same location, that is, this is a two node element and direction of this element is calculated from two distinct locations/coordinates of nodes.

205.3.4.51 Modeling, Finite Element: Shear Beam Element

The command is:

```
add element # <.> type ShearBeam
  with nodes (<.>, <.>)
  cross_section = <l^2>
  use material # <.>;
```

where

- `No (or #)<element_number>` is a unique element integer number (does not have to be sequential, any unique positive integer number can be used)
- `with nodes (n1, n2)` are the 2 nodes (node numbers) defining this element. NOTE: element is supposed to be alligned along vertical, Z direction !!
- `use material No (or #)` is the material (LT-based material) number which makes up the element.
- `section_area` is the cross section area $[L^2]$

Description of output by this element can be found in Section 206.8.3. more on this finite element can be found in Section 102.9 on page 135 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.52 Modeling, Finite Element: Stochastic Shear Beam Element

Add stochastic shear beam element for stochastic finite element analysis.

```
add element # <.> type stochastic_shear_beam with nodes (<.>, <.>)
  use material # <.>
  triple product # <.>
  cross_section = <L^2>
  mass_density = <M/L^3>;
```

where

- `No (or #)<element_number>` is a unique element integer number (does not have to be sequential, any unique positive integer number can be used).
- `with nodes (n1, n2)` are the 2 nodes (node numbers) defining this element.
- `use material No (or #)` is the stochastic uniaxial material number that makes up the element.
- `triple product #` specifies the ID of the triple product, that would be used in the formation of elemental stochastic stiffness matrix. In stochastic finite element method (FEM), the first PC basis for this triple product should come from the PC representation of uncertain element stiffness. The second and third PC basis for this triple product should come from the PC representation of uncertain FEM system response, e.g., uncertain structural displacement.
- `section_area` is the cross section area $[L^2]$.
- `mass_density` is the density $[M/L^3]$.

For example:

```
add element # 1 type stochastic_shear_beam with nodes (1, 2) use material # 1 ↩
    triple product # 1 cross_section = 1*m^2 mass_density = 2000*kg/m^3;
```

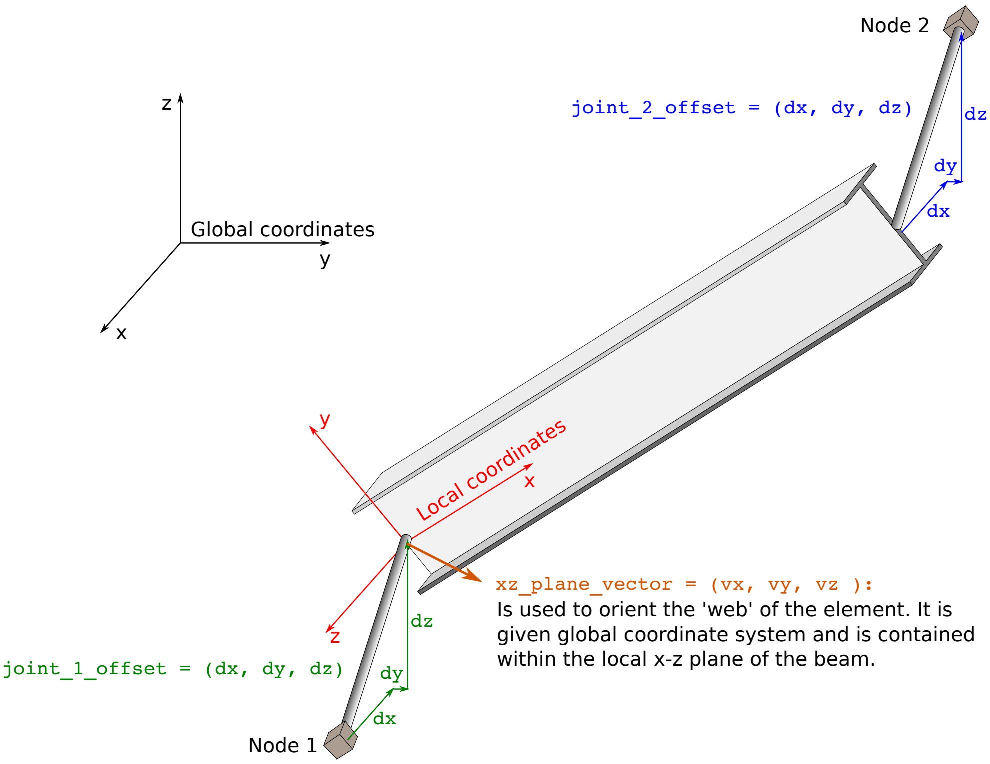
Add a stochastic shear beam element # 1 with stochastic nodes 1 and 2 using stochastic uniaxial material # 1.

The cross section of the element is 1 m^2 and mass density is 2000 kg/m^3.

205.3.4.53 Modeling, Finite Element: Elastic Beam–Column Element

The command is:

```
add element # <.> type beam_elastic with nodes (<.>, <.>)
    cross_section = <L^2>
    elastic_modulus = <F/L^2>
    shear_modulus = <F/L^2>
    torsion_Jx = <length^4>
    bending_Iy = <length^4>
    bending_Iz = <length^4>
    mass_density = <M/L^3>
    xz_plane_vector = (<.>, <.>, <.> )
    joint_1_offset = (<L>, <L>, <L> )
    joint_2_offset = (<L>, <L>, <L> );
```



Is used to orient the 'web' of the element. It is given global coordinate system and is contained within the local x-z plane of the beam.

Figure 205.3: Beam Element, sketch of main geometric components.

where

- `No (or #)<element_number>` is a unique element integer number (does not have to be sequential, any unique positive integer number can be used)
- `type beam_elastic` is the element type

- `with nodes (n1, n2)` are the 2 nodes (node numbers) defining this element
- `cross_section` is the cross section area, $[L^2]$
- `elastic_modulus` elastic modulus of the material which makes up the beam, $[F/L^2]$
- `shear_modulus` shear modulus of the material which makes up the beam, $[F/L^2]$
- `torsion_Jx` cross section polar (torsional) moment of inertia, $[L^4]$
- `bending_Iy` cross section moment of inertia about local y axis, $[L^4]$
- `bending_Iz` cross section moment of inertia about local z axis, $[L^4]$
- `mass_density` mass per unit volume of the material, $[M/L^3]$
- `xz_plane_vector` a vector which defines the orientation of the local (beam coordinate system) xz plane in global coordinates. NOTE: Please make sure that your `xz_plane_vector` is a bit away from the actuall local x axes, the axes that runs along the beam elelent, in order to prevent numerical problems that might appear when vector cross products are performed inside the program... It is suggested that your `xz_plane_vector` be closer to local z axes... See Figure 205.4 on Page 901 for more in depth explanation of `xz_plane_vector`.
- `joint_1_offset` vector defining the rigid offset between end of beam and connection node 1, $[L]$
- `joint_2_offset` vector defining the rigid offset between end of beam and connection node 2, $[L]$

Description of output by this element can be found in Section 206.8.4

more on this finite element can be found in Section 102.7 on Page 126 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

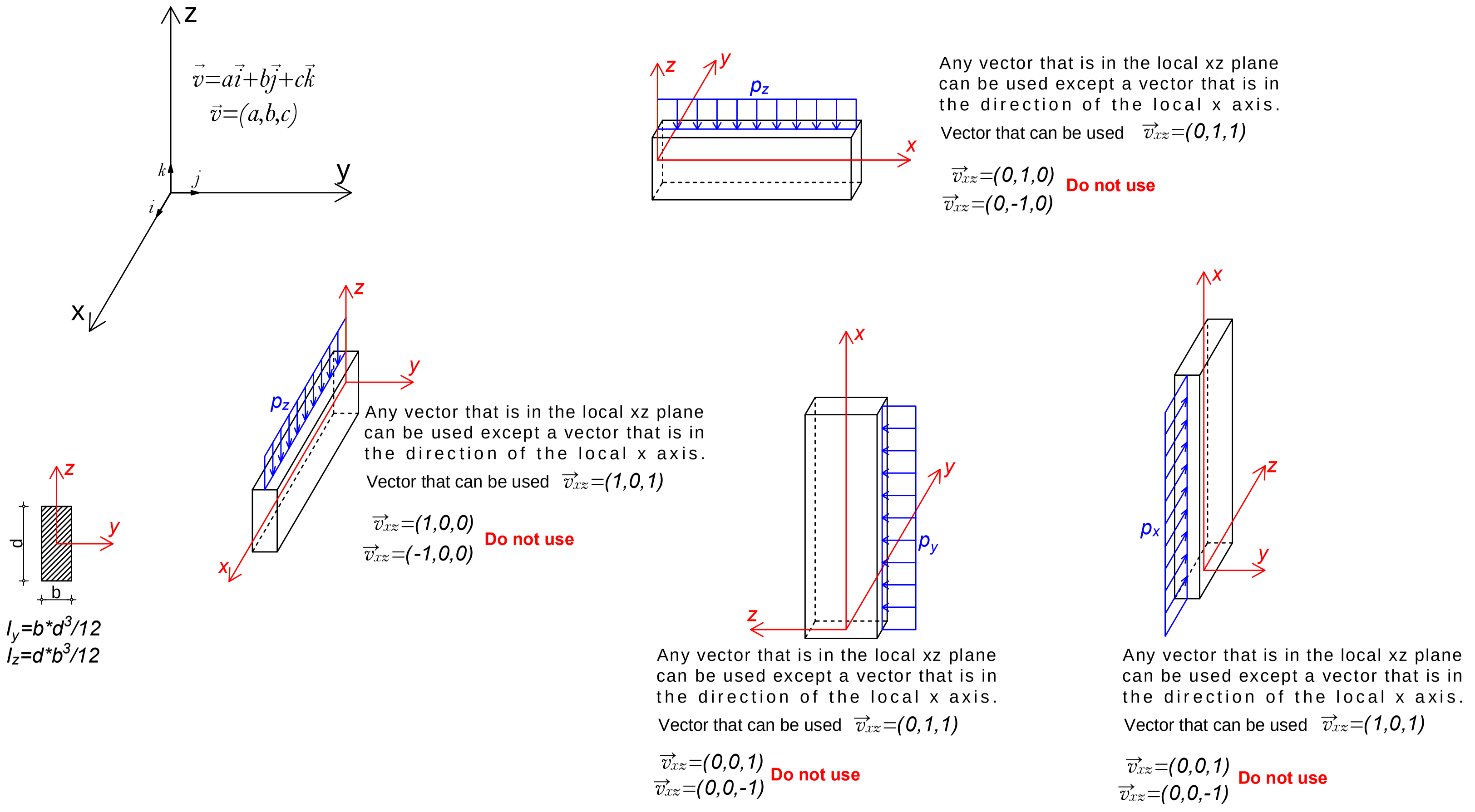


Figure 205.4: Beam Element, details of vector in X-Z plane.

205.3.4.54 Modeling, Finite Element: Large Displacement Elastic Beam–Column Element, with Corotational Transformation

The command is:

```
add element # <.> type beam_elastic_corotational with nodes (<.>, <.>)
   cross_section = <L^2>
   elastic_modulus = <F/L^2>
   shear_modulus = <F/L^2>
   torsion_Jx = <length^4>
   bending_Iy = <length^4>
   bending_Iz = <length^4>
   mass_density = <M/L^3>
   xz_plane_vector = (<.>, <.>, <.> )
   joint_1_offset = (<L>, <L>, <L> )
   joint_2_offset = (<L>, <L>, <L> );
```

where

- `No (or #)<element_number>` is a unique element integer number (does not have to be sequential, any unique positive integer number can be used)
- `type beam_elastic` is the element type
- `with nodes (n1, n2)` are the 2 nodes (node numbers) defining this element
- `cross_section` is the cross section area, $[L^2]$
- `elastic_modulus` elastic modulus of the material which makes up the beam, $[F/L^2]$
- `shear_modulus` shear modulus of the material which makes up the beam, $[F/L^2]$
- `torsion_Jx` cross section polar (torsional) moment of inertia, $[L^4]$
- `bending_Iy` cross section moment of inertia about local y axis, $[L^4]$
- `bending_Iz` cross section moment of inertia about local z axis, $[L^4]$
- `mass_density` mass per unit volume of the material, $[M/L^3]$
- `xz_plane_vector` a vector which defines the orientation of the local (beam coordinate system) xz plane in global coordinates.
- `joint_1_offset` vector defining the rigid offset between end of beam and connection node 1, $[L]$
- `joint_2_offset` vector defining the rigid offset between end of beam and connection node 2, $[L]$

205.3.4.55 Modeling, Finite Element: Timoshenko Elastic Beam–Column Element

The command is:

```
add element # <.> type beam_elastic_Timoshenko with nodes (<.>, <.>)
    cross_section = <L^2>
    elastic_modulus = <F/L^2>
    shear_modulus = <F/L^2>
    torsion_Jx = <length^4>
    bending_Iy = <length^4>
    bending_Iz = <length^4>
    mass_density = <M/L^3>
    shear_correction_coefficient = <.>
    xz_plane_vector = (<.>, <.>, <.> )
    joint_1_offset = (<L>, <L>, <L> )
    joint_2_offset = (<L>, <L>, <L> );
```

where

- `No (or #)<element_number>` is a unique element integer number (does not have to be sequential, any unique positive integer number can be used)
- `type beam_elastic` is the element type
- `with nodes (n1, n2)` are the 2 nodes (node numbers) defining this element
- `cross_section` is the cross section area, $[L^2]$
- `elastic_modulus` elastic modulus of the material which makes up the beam, $[F/L^2]$
- `shear_modulus` shear modulus of the material which makes up the beam, $[F/L^2]$
- `torsion_Jx` cross section polar (torsional) moment of inertia, $[L^4]$
- `bending_Iy` cross section moment of inertia about local y axis, $[L^4]$
- `bending_Iz` cross section moment of inertia about local z axis, $[L^4]$
- `mass_density` mass per unit volume of the material, $[M/L^3]$
- `shear_correction_coefficient` a parameter for shear correction. When this parameter becomes very large, the Timoshenko beam element becomes Euler-Bernoulli beam. If not specifically calibrated, can use 1.0 for this parameter.
- `xz_plane_vector` a vector which defines the orientation of the local (beam coordinate system) xz plane in global coordinates.

- `joint_1_offset` vector defining the rigid offset between end of beam and connection node 1, $[L]$
- `joint_2_offset` vector defining the rigid offset between end of beam and connection node 2, $[L]$

Description of output by this element can be found in Section 206.8.4. more on this finite element can be found in Section 102.7 on Page 126 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.56 Modeling, Finite Element: Timoshenko Elastic Beam–Column Element with Directional Shear Correction Coefficients

The command is:

```
add element # <.> type beam_elastic_Timoshenko_directional with nodes (<.>, <.>)
    cross_section = <L^2>
    elastic_modulus = <F/L^2>
    shear_modulus = <F/L^2>
    torsion_Jx = <length^4>
    bending_Iy = <length^4>
    bending_Iz = <length^4>
    mass_density = <M/L^3>
    shear_correction_coefficient_y = <.>
    shear_correction_coefficient_z = <.>
    xz_plane_vector = (<.>, <.>, <.> )
    joint_1_offset = (<L>, <L>, <L> )
    joint_2_offset = (<L>, <L>, <L> );
```

where

- `No (or #)<element_number>` is a unique element integer number (does not have to be sequential, any unique positive integer number can be used)
- `type beam_elastic` is the element type
- `with nodes (n1, n2)` are the 2 nodes (node numbers) defining this element
- `cross_section` is the cross section area, $[L^2]$
- `elastic_modulus` elastic modulus of the material which makes up the beam, $[F/L^2]$
- `shear_modulus` shear modulus of the material which makes up the beam, $[F/L^2]$
- `torsion_Jx` cross section polar (torsional) moment of inertia, $[L^4]$
- `bending_Iy` cross section moment of inertia about local y axis, $[L^4]$
- `bending_Iz` cross section moment of inertia about local z axis, $[L^4]$
- `mass_density` mass per unit volume of the material, $[M/L^3]$
- `shear_correction_coefficient_y` parameter for shear correction about local y axis. When this parameter becomes very large, the Timoshenko beam element becomes Euler-Bernoulli beam. If not specifically calibrated, can use 1.0 for this parameter.

- `shear_correction_coefficient_z` parameter for shear correction about local z axis. When this parameter becomes very large, the Timoshenko beam element becomes Euler-Bernoulli beam. If not specifically calibrated, can use 1.0 for this parameter.
- `xz_plane_vector` a vector which defines the orientation of the local (beam coordinate system) xz plane in global coordinates.
- `joint_1_offset` vector defining the rigid offset between end of beam and connection node 1, $[L]$
- `joint_2_offset` vector defining the rigid offset between end of beam and connection node 2, $[L]$

Description of output by this element can be found in Section 206.8.4. more on this finite element can be found in Section 102.7 on Page 126 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.57 Modeling, Finite Element: Adding 1D Fiber to a Beam Cross Section

Fibers can be added to the fiber beam cross section.

The command is:

```
add fiber # <.> using material # <.> to section # <.> fiber_cross_section = ↩
    <area> fiber_location = (<L>,<L>);
```

For example:

```
add fiber # 1 using material # 1 to section # 1 fiber_cross_section = 5*cm^2 ↩
    fiber_location = (10*cm,10*cm);
```

adds a fiber number 1 to section number 1 at coordinates $y = 10\text{cm}$, $z = 10\text{cm}$ with cross section area of 5cm^2 using material number 1.

The material for fiber must be a uniaxial material, for example uniaxial_concrete02, uniaxial_elastic, uniaxial_steel01, and uniaxial_steel02.

205.3.4.58 Modeling, Finite Element: Adding Fiber Section to the Finite Element Model

Fiber section can be added to the finite element model.

The command is:

```
add section # <.> type FiberSection
    TorsionConstant_GJ = <F*L^2);
```

where

- `TorsionConstant_GJ` provides a linear torsional stiffness to the element.

Fibers can be added to the section as described in section 205.3.4.57 on page 907.

The command is:

```
add fiber # <.> using material # <.> to section # <.> fiber_cross_section = <area>
  fiber_location = (<L>,<L>);
```

where

- `fiber_cross_section` is the area of the fiber element. (Total cross section are is the sum of all fiber areas) $[L^2]$
- `fiber_location` location of the fiber in the beam local Y-Z plane.

205.3.4.59 Modeling, Finite Element: 3D Displacement Based Fiber Beam-Column Element

```
add element # <.> type BeamColumnDispFiber3d with nodes (<.>, <.>)
   number_of_integration_points = <.>
   section_number = <.>
   mass_density = <M/L^3>
   xz_plane_vector = (<.>, <.>, <.> )
   joint_1_offset = (<L>, <L>, <L> )
   joint_2_offset = (<L>, <L>, <L> );
```

where

- `No (or #)<element_number>` is the unique element integer number (does not have to be sequential, any unique positive integer number can be used)
- `type BeamColumnDispFiber3d` is the element type
- `with nodes (n1, n2)` are the 2 nodes defining this element
- `number_of_integration_points` is number of integration points to be used along the beam element
- `section_number` is the number of predefined section
- `mass_density` mass per unit volume of the material, $[M/L^3]$
- `xz_plane_vector` unit vector which defines the orientation of the web of the beam in global coordinates.
- `joint_1_offset` vector defining the rigid offset between end of beam and connection node 1, $[L]$
- `joint_2_offset` vector defining the rigid offset between end of beam and connection node 2, $[L]$

Description of output by this element can be found in Section 206.8.6,

205.3.4.60 Modeling, Finite Element: 3D Displacement Based Fiber Beam-Column Element with Corotational Coordinate Transformation

```
add element # <.> type BeamColumnDispFiber3d_Corotational with nodes (<.>, <.>)
   number_of_integration_points = <.>
   section_number = <.>
   mass_density = <M/L^3>
   xz_plane_vector = (<.>, <.>, <.> )
   joint_1_offset = (<L>, <L>, <L> )
   joint_2_offset = (<L>, <L>, <L> );
```

where

- `No (or #)<element_number>` is the unique element integer number (does not have to be sequential, any unique positive integer number can be used)
- `type BeamColumnDispFiber3d_Corotational` is the element type
- `with nodes (n1, n2)` are the 2 nodes defining this element
- `number_of_integration_points` is number of integration points to be used along the beam element
- `section_number` is the number of predefined section
- `mass_density` mass per unit volume of the material, $[M/L^3]$
- `xz_plane_vector` unit vector which defines the orientation of the web of the beam in global coordinates.
- `joint_1_offset` vector defining the rigid offset between end of beam and connection node 1, $[L]$
- `joint_2_offset` vector defining the rigid offset between end of beam and connection node 2, $[L]$

Description of output by this element can be found in section 206.8.6.

The co-rotational formulation used in this element is based on Crisfield (1990).

205.3.4.61 Modeling, Finite Element: 3DOF+6DOF=9DOF Beam-Column Element

```
add element # <.> type beam_9dof_elastic
    with nodes (<.>, <.>)
    cross_section = <L^2>
    elastic_modulus = <F/L^2>
    shear_modulus = <F/L^2>
    torsion_Jx = <length^4>
    bending_Iy = <length^4>
    bending_Iz = <length^4>
    mass_density = <M/L^3>
    xz_plane_vector = (<.>, <.>, <.> )
    joint_1_offset = (<L>, <L>, <L> )
    joint_2_offset = (<L>, <L>, <L> );
```

where

- `No (or #)<element_number>` is the unique element integer number (does not have to be sequential, any unique positive integer number can be used)
- `type beam_9dof_elastic` is the element type
- `with nodes (n1, n2)` are the 2 nodes defining this element, where the first node (n1) is the one with 3 DOFs and the second (n2) is the one with 6 DOFs
- `cross_section` is the cross section area, $[L^2]$
- `elastic_modulus` elastic modulus of the material which makes up the beam, $[F/L^2]$
- `shear_modulus` shear modulus of the material which makes up the beam, $[F/L^2]$
- `torsion_Jx` cross section polar (torsional) moment of inertia, $[L^4]$
- `bending_Iy` cross section moment of inertia about local y axis, $[L^4]$
- `bending_Iz` cross section moment of inertia about local z axis, $[L^4]$
- `mass_density` mass per unit volume of the material, $[M/L^3]$
- `xz_plane_vector` unit vector which defines the orientation of the web of the beam in global coordinates.
- `joint_1_offset` vector defining the rigid offset between end of beam and connection node 1, $[L]$
- `joint_2_offset` vector defining the rigid offset between end of beam and connection node 2, $[L]$

This finite element has only 3DOFs (translations) at the first node, and full 6DOFs at the other, second node. Due to missing rotational stiffness on first, 3DOF node, this beam has zero torsional stiffness.

This element is useful for connection of solid (3DOFs per node) and structural (6DOFs per node) elements It his beam element is used on its own, DOF that corresponds to torsion of the second node (DOF number 7), should be fixed as this beam does not provide that stiffness.

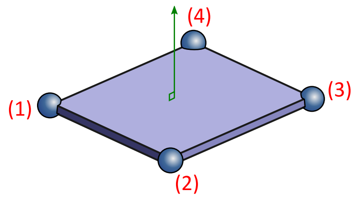
More on this finite element can be found in Section 102.8 on Page 129 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.62 Modeling, Finite Element: 4 Node ANDES Shell with Drilling DOFs

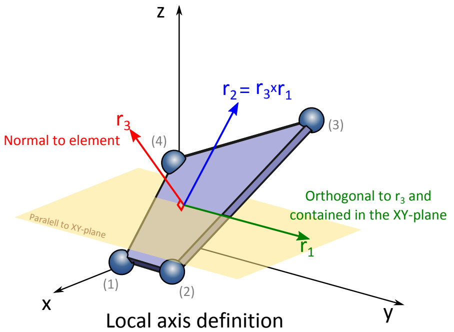
ANDES based 3D shell element including drilling degrees of freedom. Made up by patching together 4 ANDES shell triangle elements (and then averaging two and two squares made up two and two triangles).

The command is:

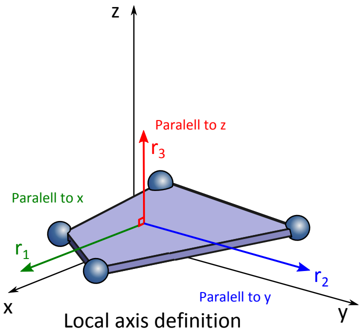
```
add element # <.> type 4NodeShell_ANDES
    with nodes (<.>, <.>, <.>)
    use material # <.>
    thickness = <L> ;
```



Node numbering
is counter-clockwise



Local axis definition
in general



Local axis definition
if r_3 is nearly vertical

- `No (or #)<element_number>` is the unique element integer number (does not have to be sequential, any unique positive integer number can be used)
- `material # <.>` number of a previously defined material. (see `add material ...`)
- `thickness` shell thickness, $[L]$

Description of output by this element can be found in Section 206.8.5.

More on this finite element can be found in Section 102.10 on page 135 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.63 Modeling, Finite Element: 3 Node ANDES Shell with Drilling DOFs

```
add element # <.> type 3NodeShell_ANDES
    with nodes (<.>, <.>, <.>)
    use material # <.>
    thickness = <L> ;
```

- `No (or #)<element_number>` is the unique element integer number (does not have to be sequential, any unique positive integer number can be used)

205.3.4.64 Modeling, Finite Element: 4 Node Shell NLDKGQ, or 4 Node Shell Xin-Zheng-Lu

This is a 3D quadrilateral shell element with membrane and drill DOFs based on the theory of generalized conforming element. This element accounts for the geometric nonlinearity of large deformation using a simplified version of updated Lagrangian formulation, where nodal coordinates are updated in each step, however strains and stresses are still calculated with reference to the original, undeformed system. It can be used together with elastic or inelastic sections. This element was originally developed by Professor Xin-Zheng Lu (Tsinghua University) and his students.

The command is:

```
add element # <.> type 4NodeShell_NLDKGQ
    with nodes (<.>, <.>, <.>, <.>)
    section_number = <.>;
```

It can also be called using the alternative command:

```
add element # <.> type 4NodeShell_XinZhengLu_Tsinghua
    with nodes (<.>, <.>, <.>, <.>)
    section_number = <.>;
```

- `No (or #)<element_number>` is the unique element integer number (does not have to be sequential, any unique positive integer number can be used)
- `with nodes (n1, n2, n3, n4)` are the 4 nodes defining this element
- `section_number` is the number of predefined shell cross section, described on page 916.

205.3.4.65 Modeling, Finite Element: Inelastic Layered Shell Section

This command is used to add a layered shell section. The section is made up of a number of layers with different thicknesses and different material properties (i.e., concrete layers or rebar layers). This type of section is used together with plane stress materials and shell elements.

The command is:

```
add section # <.> type LayeredShellFiber
   number_of_layers = <.>
   thickness_array = "<.>,<.>..."
   with material # "<.>,<.>..."
   thickness_scale_unit = <L>
   outofplane_shear_modulus = <F/L^2>;
```

where

- `number_of_layers` is the number of layers that the section has
- `thickness_array` is the relative thickness of each layer
- `with material #` is the material tag of each layer, only plane stress materials can be used here, see pages 878 and 879.
- `thickness_scale_unit` is the total thickness of the section
- `outofplane_shear_modulus` is the out-of-plane shear modulus of the section

205.3.4.66 Modeling, Finite Element: ElasticMembranePlaneStress Element (to be removed!)

NOTE: this element is being removed, and will not be available after Real ESSI version 19.07 (current). This is a 2D finite element, and we only maintain 3D finite elements. This element is replaced by a 3D 27 node elastic and/or elastic-plastic wall/plate/shell brick element.

The command is:

```
add element No (or #) <element_number>
    type ElasticMembranePlaneStress
    with nodes (n1, n2, n3, n4)
    use material No (or #) <material_number>
    thickness = <L> ;
```

where

- `No (or #)<element_number>` is a unique element integer number (does not have to be sequential, any unique positive integer number can be used).
- `type ElasticMembranePlaneStress` is the element type.
- `with nodes (n1, n2, n3, n4)` are the 4 nodes (node numbers) defining this element.
- `use material No (or #)` is the material number for linear elastic material that makes up the element.
- `thickness` is the thickness of the membrane.

205.3.4.67 Modeling, Finite Element: InelasticMembranePlaneStress Element (to be removed!)

NOTE: this element is being removed, and will not be available after Real ESSI version 19.07 (current). This is a 2D finite element, and we only maintain 3D finite elements. This element is replaced by a 3D 27 node elastic or elastic-plastic wall/plate/shell brick element or elastic-plastic shell element.

The command is:

```
add element No (or #) <element_number>
   type InelasticMembranePlaneStress
   with nodes (n1, n2, n3, n4)
   use material No (or #) <material_number>
   ;
```

where

- `No (or #)<element_number>` is a unique element integer number (does not have to be sequential, any unique positive integer number can be used).
- `type InelasticMembranePlaneStress` is the element type.
- `with nodes (n1, n2, n3, n4)` are the 4 nodes (node numbers) defining this element.
- `use material No (or #)` is the material number for inelastic material that makes up the element. Since this is a plane stress element, material needs to have plane stress constitutive integration algorithm available. In addition, this material should specify thickness of the element. Different layers and their thicknesses for different materials (for example concrete and steel) will be defined within material definition. `PlaneStressLayeredMaterial` is a material of this type.

205.3.4.68 Modeling, Finite Element: SuperElementLinearElasticImport

The command is:

```
add element No (or #) <element_number>
    type SuperElementLinearElasticImport
    with hdf5_file = <string>
    ;
```

where

- `No (or #)<element_number>` is a unique element integer number (does not have to be sequential, any unique positive integer number can be used).
- `type SuperElementLinearElasticImport` is the element type.
- `hdf5_file` specifies the HDF5 filename of the SuperElement with SuperElement data. The HDF5 file should contain the following datasets:
  - `Node` dataset within HDF5 file is organized in a column (a 1D dataset), and it specifies the node tags/numbers of nodes that make up the SuperElement.
  - `DofList` dataset within HDF5 file is a organized in a column, and it specifies the number of DOFs per each `Node`. For example if nodes are representing structural elements, they usually have 6 DOFs per node, while solids will have 3 DOFs per node. `DofList` dataset has to have the same number of entries as `Node` dataset, as each entry in `DofList` corresponds to one node from `Node` dataset.
  - `MassMatrix` is a matrix, that sets masses/numbers for a mass matrix of the SuperElement.
  - `StiffnessMatrix` is a matrix, that sets stiffness/numbers for a stiffness matrix of the SuperElement.
  - `ConnectNode` dataset within HDF5 file is organized in a column (a 1D dataset), and it specifies the node tags/numbers of nodes that are going to be connected to Real-ESSI mesh.
  - `ConnectNodeCoordinate` dataset within HDF5 file is organized in a matrix (a 2D dataset), and it specifies the nodal coordinates for nodes that are going to be connected to Real-ESSI mesh. Since each node has 3 coordinates, the length of `ConnectNodeCoordinate` is the same as the length of `ConnectNode` and each line has three entries, for X, Y and Z coordinates of given node.

In addition to the minimum dataset requirements above, users can get more output from Real-ESSI:

- Results for individual finite elements (internal forces, etc.), can be obtained if node, DofList, mass matrix and stiffness matrix for each finite element within the super element are provided.
- Graphical post-processing can be obtained if coordinates for all nodes and their connectivity into finite elements are provided (a mesh data).

205.3.4.69 Modeling, Finite Element: 8 Node Brick Element

The command is:

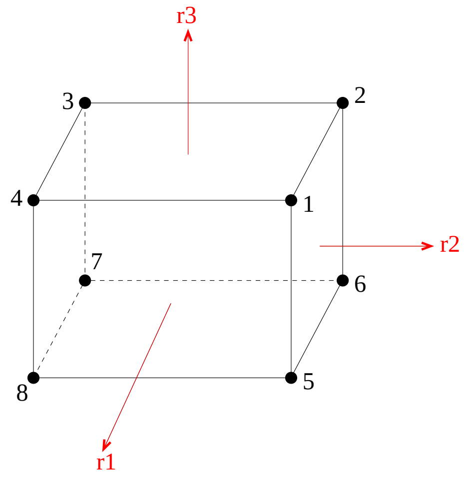
```
add element # <element_number> type 8NodeBrick
   using <.> Gauss points each direction
   with nodes (n1, n2, n3, n4, n5, n6, n7, n8)
   use material No (or #) <material_number>;
```

and/or;

```
add element # <element_number> type 8NodeBrick
   with nodes (n1, n2, n3, n4, n5, n6, n7, n8)
   use material No (or #) <material_number>;
```

where:

- `No (or #)<element_number>` is the unique element integer number (does not have to be sequential, any unique positive integer number can be used)
- `type 8NodeBrick` is the element type.
- `with nodes (n1, n2, n3, n4, n5, n6, n7, n8)` are the 8 nodes for this element, in the order as per figure below



- `using <.> Gauss points each direction` is the number of Gauss points to be used in each direction (r1, r2, and r3) for integration of finite element matrices (mass and stiffness). There can be from 1 to 6 Gauss points used (uniformly) in each direction (r1, r2, and r3). Command for the brick finite element (above) without number of Gauss points control is kept for back compatibility.

For 8 node bricks 2 Gauss points are used in each direction ($2 \times 3 \times 3$), while for 20 nodes, 8-20 node and 8-27 node bricks 3 Gauss points are used in each direction ($3 \times 3 \times 3$),

- `use material No (or #)` is the material number which makes up the element (nonlinear elastic or elastic-plastic material properties for each integration (Gauss) point will evolve independently as the element deforms). Use LT version with LT materials.

Description of output by this element can be found in section 206.8.2.

More on this finite element can be found in Section 102.4.1 on page 114 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.70 Modeling, Finite Element: 20 Node Brick Element

The command is:

```
add element No (or #) <element_number> type 20NodeBrick
   using <.> Gauss points each direction
   with nodes (n1, n2, n3, n4, n5, n6, n7, n8,
            n9, n10, n11, n12, n13, n14, n15, n16,
            n17, n18, n19, n20 )
   use material No (or #) <material_number>;
```

and/or

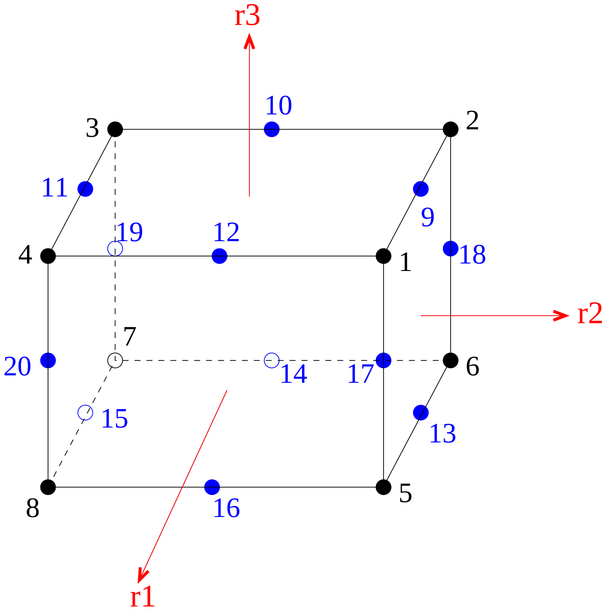
```
add element No (or #) <element_number> type 20NodeBrick
   with nodes (n1, n2, n3, n4, n5, n6, n7, n8,
            n9, n10, n11, n12, n13, n14, n15, n16,
            n17, n18, n19, n20 )
   use material No (or #) <material_number>;
```

where:

- `No (or #)<element_number>` is the unique element integer number (does not have to be sequential, any unique positive integer number can be used)
- `type 20NodeBrick` is the element type. `20NodeBrick_elastic` can be used if elastic material is used. It this case, the stiffness and mass matrices will not be updated at each step.
- `with nodes (n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15, n16, n17, n18, n19, n20)` are the 20 nodes for this element, written in the order defined as per figure below
- `using <.> Gauss points each direction` is the number of Gauss points to be used in each direction (r1, r2, and r3) for integration of finite element matrices (mass and stiffness). There can be from 1 to 6 Gauss points used (uniformly) in each direction (r1, r2, and r3). Command for the brick finite element (above) without number of Gauss points control is kept for back compatibility. For 8 node bricks 2 Gauss points are used in each direction ($2 \times 3 \times 3$), while for 20 nodes, 8-20 node and 8-27 node bricks 3 Gauss points are used in each direction ($3 \times 3 \times 3$),
- `use material No (or #)` is the material number which makes up the element (nonlinear elastic or elastic-plastic material properties for each integration (Gauss) point will evolve independently as the element deforms)

Description of output by this element can be found in Section 206.8.2.

More on this finite element can be found in Section 102.4.3 on page 116 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).



205.3.4.71 Modeling, Finite Element: 27 Node Brick Element

The command is:

```
add element # <element_number>
          type 27NodeBrick
          using <.> Gauss points each direction
          with nodes (n1, n2, n3, n4, n5, n6, n7, n8,
                    n9, n10, n11, n12, n13, n14, n15, n16,
                    n17, n18, n19, n20, n21, n22, n23,
                    n124, n25, n26, n27 )
          use material # <material_number>;
```

and/or

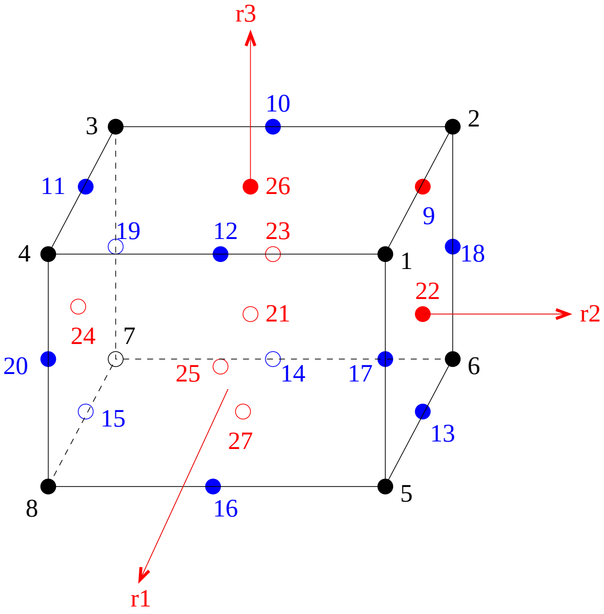
```
add element # <element_number>
          type 27NodeBrick
          with nodes (n1, n2, n3, n4, n5, n6, n7, n8,
                    n9, n10, n11, n12, n13, n14, n15, n16,
                    n17, n18, n19, n20, n21, n22, n23,
                    n124, n25, n26, n27 )
          use material # <material_number>;
```

where:

- `No (or #)<element_number>` is the unique element integer number (does not have to be sequential, any unique positive integer number can be used)
- `type 27NodeBrick` is the element type.
- `with nodes (n1, n2, n3, n4, n5, n6, n7, n8,`
  `n9, n10, n11, n12, n13, n14, n15, n16, n17, n18, n19, n20)`
  `n21, n22, n23, n24, n25, n26, n27)` are the 27 nodes for this element, written in the order defined as per this figure
- `using <.> Gauss points each direction` is the number of Gauss points to be used in each direction (r1, r2, and r3) for integration of finite element matrices (mass and stiffness). There can be from 1 to 6 Gauss points used (uniformly) in each direction (r1, r2, and r3). Command for the brick finite element (above) without number of Gauss points control is kept for back compatibility. For 8 node bricks 2 Gauss points are used in each direction ($2 \times 3 \times 3$), while for 20 nodes, 8-20 node and 8-27 node bricks 3 Gauss points are used in each direction ($3 \times 3 \times 3$),
- `use material No (or #)` is the material number which makes up the element (nonlinear elastic and/or elastic-plastic material properties for each integration (Gauss) point will evolve independently as the element deforms).

Description of output by this element can be found in Section 206.8.2.

More on this finite element can be found in Section 102.4.4 on page 118 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).



205.3.4.72 Modeling, Finite Element: Variable 8-27 Node Brick Element

The command is:

```
add element No (or #) <element_number> type variable_node_brick_8_to_27
          using <.> Gauss points each direction
          with nodes (n1, n2, n3, n4, n5, n6, n7, n8,
                    n9, n10, n11, n12, n13, n14, n15, n16,
                    n17, n18, n19, n20, n21, n22, n23, n24, n25, n26, n27)
          use material No (or #) <material_number>;
```

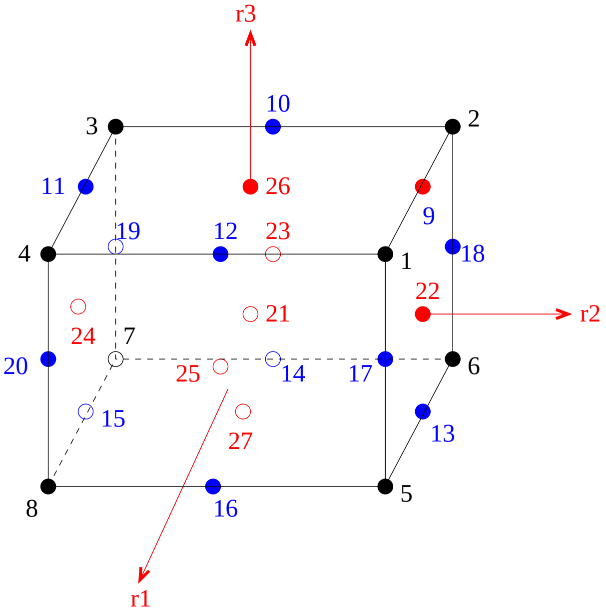
and/or

```
add element No (or #) <element_number> type variable_node_brick_8_to_27
          using <.> Gauss points each direction
          with nodes (n1, n2, n3, n4, n5, n6, n7, n8,
                    n9, n10, n11, n12, n13, n14, n15, n16,
                    n17, n18, n19, n20, n21, n22, n23, n24, n25, n26, n27)
          use material No (or #) <material_number>;
```

where:

- `No (or #)<element_number>` is the unique element integer number (does not have to be sequential, any unique positive integer number can be used)
- `type variable_node_brick_8_to_27` is the element type
- `with nodes (n1, n2, n3, n4, n5, n6, n7, n8,`
  `n9, n10, n11, n12, n13, n14, n15, n16, n17, n18, n19, n20,`
  `n21, n22, n23, n24, n25, n26, n27)`
  are the 8 to 27 nodes for this element, written in the order defined as per this figure. Nodes 1-8 are obligatory, while any other nodes can be used but do not have to, the element will automatically pick proper shape functions. This element is good for transitions in meshing.
- `using <.> Gauss points each direction` is the number of Gauss points to be used in each direction (r1, r2, and r3) for integration of finite element matrices (mass and stiffness). There can be from 1 to 6 Gauss points used (uniformly) in each direction (r1, r2, and r3). Command for the brick finite element (above) without number of Gauss points control is kept for back compatibility. For 8 node bricks 2 Gauss points are used in each direction ($2 \times 3 \times 3$), while for 20 nodes, 8-20 node and 8-27 node bricks 3 Gauss points are used in each direction ($3 \times 3 \times 3$),
- `material No (or #)` is the material number which makes up the element (nonlinear elastic and/or elastic-plastic material properties for each integration (Gauss) point will evolve independently as the element deforms)

Description of output by this element can be found in Section 206.8.2.



205.3.4.73 Modeling, Finite Element: 8 Node Brick u-p Element

The command is:

```
add element # <.> type 8NodeBrick_up
   using <.> Gauss points each direction
   with nodes (<.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>)
   use material # <.>
   porosity = <.>
   alpha = <.>
   rho_s = <M/L^3>
   rho_f = <M/L^3>
   k_x = <L^3*T/M>
   k_y = <L^3*T/M>
   k_z = <L^3*T/M>
   K_s = <F/L^2>
   K_f = <F/L^2>;
```

and/or

```
add element # <.> type 8NodeBrick_up
   with nodes (<.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>)
   use material # <.>
   porosity = <.>
   alpha = <.>
   rho_s = <M/L^3>
   rho_f = <M/L^3>
   k_x = <L^3*T/M>
   k_y = <L^3*T/M>
   k_z = <L^3*T/M>
   K_s = <F/L^2>
   K_f = <F/L^2>;
```

where:

- `No (or #)<element_number>` is the unique element integer number that does not have to be sequential, any unique positive integer number can be used.
- `type 8NodeBrick_up` is the element type/name.
- `with nodes (n1, n2, n3, n4, n5, n6, n7, n8)` are the 8 nodes for this element, is specified order.
- `using <.> Gauss points each direction` is the number of Gauss points to be used in each direction (r1, r2, and r3) for integration of finite element matrices (mass and stiffness). There can be from 1 to 6 Gauss points used (uniformly) in each direction (r1, r2, and r3). Command for the brick finite element (above) without number of Gauss points control is kept for back compatibility.

For 8 node bricks 2 Gauss points are used in each direction ($2 \times 3 \times 3$), while for 20 nodes, 8-20 node and 8-27 node bricks 3 Gauss points are used in each direction ($3 \times 3 \times 3$).

- `use material No (or #)` is the material number which makes up the element (nonlinear elastic or elastic-plastic material properties for each integration (Gauss) point will evolve independently as the element deforms). Use LT version with LT materials.
- `porosity` is the porosity ($n = V_{voids}/V_{total}$) of material in this element.
- `alpha` is the parameter controlling level of effective stress analysis. For soils, usually $\alpha = 1$ is used, while for other materials (saturated concrete, bone material, etc.) lower values are used ($0 \leq \alpha \leq 1$).
- `rho_s` is the density of particles of the solid phase. It is important to note that this is a density of the actual mineral that makes up solid particles!.
- `rho_f` is the density of pore fluid. It is usually density of water, however, for unsaturated and partially saturated materials, this density will be different, as described in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).
- `k_x` is the permeability in the x direction (global x) of the element. It is also important to note about the units used for permeability, as noted below. With isotropic permeability, usually the case, $k_x = k_y = k_z$.
- `k_y` is the permeability in the y direction (global y) of the element. It is also important to note about the units used for permeability, as noted below. With isotropic permeability, usually the case, $k_x = k_y = k_z$.
- `k_z` is the permeability in the z direction (global z) of the element. It is also important to note about the units used for permeability, as noted below. With isotropic permeability, usually the case, $k_x = k_y = k_z$.
- `K_s` is the bulk modulus of the soil phase particles. It is important to note that this is a bulk modulus of the actual mineral that makes up solid particles!
- `K_f` is the bulk modulus of the fluid phase that is found in porous material pores. It is usually bulk modulus of the fluid (physical value of the bulk modulus of fluid, for example water), however, for unsaturated and partially saturated materials, this density is a density of a mixture, as described in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

Note: the permeability k_x, k_y, k_z is used with dimensions of $[length]^3[time]/[mass]$, which is different from the usual soil mechanics convention, where the permeability has the dimension of velocity, i.e. $[length]/[time]$. Their values are related by $k = K/\rho_f g$, where g is the gravitational acceleration at which the permeability is measured.

More on theory for this finite element can be found in Section 102.12.3.3 on page 156 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

Description of output by this element can be found in Section 206.8.2 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.74 Modeling, Finite Element: 20 Node Brick u-p Element

The command is:

```
add element # <.> type 20NodeBrick_up
   using <.> Gauss points each direction
   with nodes (<.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>,<.>, <.>, <.>, <.>, <.>, ↩
      <.>, <.>, <.>,<.>, <.>, <.>, <.>)
   use material # <.>
   porosity = <.>
   alpha = <.>
   rho_s = <M/L^3>
   rho_f = <M/L^3>
   k_x = <L^3*T/M>
   k_y = <L^3*T/M>
   k_z = <L^3*T/M>
   K_s = <F/L^2>
   K_f = <F/L^2>;
```

and/or

```
add element # <.> type 20NodeBrick_up
   with nodes (<.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>,<.>, <.>, <.>, <.>, <.>, ↩
      <.>, <.>, <.>,<.>, <.>, <.>, <.>)
   use material # <.>
   porosity = <.>
   alpha = <.>
   rho_s = <M/L^3>
   rho_f = <M/L^3>
   k_x = <L^3*T/M>
   k_y = <L^3*T/M>
   k_z = <L^3*T/M>
   K_s = <F/L^2>
   K_f = <F/L^2>;
```

where:

- `No (or #)<element_number>` is the unique element integer number that does not have to be sequential, any unique positive integer number can be used.
- `type 8NodeBrick_up` is the element type/name.
- `with nodes (n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15,` ↩ `n16, n17, n18, n19, n20)` are the 20 nodes for this element, is specified order.
- `using <.> Gauss points each direction` is the number of Gauss points to be used in each direction (r1, r2, and r3) for integration of finite element matrices (mass and stiffness). There can be from 1 to 6 Gauss points used (uniformly) in each direction (r1, r2, and r3). Command for the

brick finite element (above) without number of Gauss points control is kept for back compatibility. For 8 node bricks 2 Gauss points are used in each direction ($2 \times 3 \times 3$), while for 20 nodes, 8-20 node and 8-27 node bricks 3 Gauss points are used in each direction ($3 \times 3 \times 3$).

- `use material No (or #)` is the material number which makes up the element (nonlinear elastic or elastic-plastic material properties for each integration (Gauss) point will evolve independently as the element deforms). Use LT version with LT materials.
- `porosity` is the porosity ($n = V_{voids}/V_{total}$) of material in this element.
- `alpha` is the parameter controlling level of effective stress analysis. For soils, usually $\alpha = 1$ is used, while for other materials (saturated concrete, bone material, etc.) lower values are used ($0 \leq \alpha \leq 1$).
- `rho_s` is the density of particles of the solid phase. It is important to note that this is a density of the actual mineral that makes up solid particles!.
- `rho_f` is the density of pore fluid. It is usually density of water, however, for unsaturated and partially saturated materials, this density will be different, as described in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).
- `k_x` is the permeability in the x direction (global x) of the element. It is also important to note about the units used for permeability, as noted below. With isotropic permeability, usually the case, $k_x = k_y = k_z$.
- `k_y` is the permeability in the y direction (global y) of the element. It is also important to note about the units used for permeability, as noted below. With isotropic permeability, usually the case, $k_x = k_y = k_z$.
- `k_z` is the permeability in the z direction (global z) of the element. It is also important to note about the units used for permeability, as noted below. With isotropic permeability, usually the case, $k_x = k_y = k_z$.
- `K_s` is the bulk modulus of the soil phase particles. It is important to note that this is a bulk modulus of the actual mineral that makes up solid particles!
- `K_f` is the bulk modulus of the fluid phase that is found in porous material pores. It is usually bulk modulus of the fluid (physical value of the bulk modulus of fluid, for example water), however, for unsaturated and partially saturated materials, this density is a density of a mixture, as described in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

Note: the permeability k_x, k_y, k_z is used with dimensions of $[length]^3[time]/[mass]$, which is different from the usual soil mechanics convention, where the permeability has the dimension of velocity, i.e. $[length]/[time]$. Their values are related by $k = K/\rho_f g$, where g is the gravitational acceleration at which the permeability is measured.

More on theory for this finite element can be found in section 102.12.3.3 on page 156 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.75 Modeling, Finite Element: 27 Node Brick u-p Element

The command is:

```
add element # <.> type 27NodeBrick_up
   using <.> Gauss points each direction
   with nodes (<.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>,<.>, <.>, <.>, <.>, <.>, ↩
      <.>, <.>, <.>,<.>, <.>, <.>, <.>, <.>, <.>, <.>,<.>, <.>, <.>, <.>)
   use material # <.>
   porosity = <.>
   alpha = <.>
   rho_s = <M/L^3>
   rho_f = <M/L^3>
   k_x = <L^3*T/M>
   k_y = <L^3*T/M>
   k_z = <L^3*T/M>
   K_s = <F/L^2>
   K_f = <F/L^2>;
```

and/or

```
add element # <.> type 27NodeBrick_up
   with nodes (<.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>,<.>, <.>, <.>, <.>, <.>, ↩
      <.>, <.>, <.>,<.>, <.>, <.>, <.>, <.>, <.>, <.>,<.>, <.>, <.>, <.>)
   use material # <.>
   porosity = <.>
   alpha = <.>
   rho_s = <M/L^3>
   rho_f = <M/L^3>
   k_x = <L^3*T/M>
   k_y = <L^3*T/M>
   k_z = <L^3*T/M>
   K_s = <F/L^2>
   K_f = <F/L^2>;
```

where:

- `No (or #)<element_number>` is the unique element integer number that does not have to be sequential, any unique positive integer number can be used.
- `type 8NodeBrick_up` is the element type/name.
- `with nodes (n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15, ↩ n16, n17, n18, n19, n20, n21, n22, n23, n24, n25, n26, n27)` are the 27 nodes for this element, is specified order.
- `using <.> Gauss points each direction` is the number of Gauss points to be used in each direction (r1, r2, and r3) for integration of finite element matrices (mass and stiffness). There can

be from 1 to 6 Gauss points used (uniformly) in each direction (r1, r2, and r3). Command for the brick finite element (above) without number of Gauss points control is kept for back compatibility. For 8 node bricks 2 Gauss points are used in each direction ($2 \times 3 \times 3$), while for 20 nodes, 8-20 node and 8-27 node bricks 3 Gauss points are used in each direction ($3 \times 3 \times 3$).

- `use material No (or #)` is the material number which makes up the element (nonlinear elastic or elastic-plastic material properties for each integration (Gauss) point will evolve independently as the element deforms). Use LT version with LT materials.
- `porosity` is the porosity ($n = V_{voids}/V_{total}$) of material in this element.
- `alpha` is the parameter controlling level of effective stress analysis. For soils, usually $\alpha = 1$ is used, while for other materials (saturated concrete, bone material, etc.) lower values are used ($0 \leq \alpha \leq 1$).
- `rho_s` is the density of particles of the solid phase. It is important to note that this is a density of the actual mineral that makes up solid particles!.
- `rho_f` is the density of pore fluid. It is usually density of water, however, for unsaturated and partially saturated materials, this density will be different, as described in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).
- `k_x` is the permeability in the x direction (global x) of the element. It is also important to note about the units used for permeability, as noted below. With isotropic permeability, usually the case, $k_x = k_y = k_z$.
- `k_y` is the permeability in the y direction (global y) of the element. It is also important to note about the units used for permeability, as noted below. With isotropic permeability, usually the case, $k_x = k_y = k_z$.
- `k_z` is the permeability in the z direction (global z) of the element. It is also important to note about the units used for permeability, as noted below. With isotropic permeability, usually the case, $k_x = k_y = k_z$.
- `K_s` is the bulk modulus of the soil phase particles. It is important to note that this is a bulk modulus of the actual mineral that makes up solid particles!
- `K_f` is the bulk modulus of the fluid phase that is found in porous material pores. It is usually bulk modulus of the fluid (physical value of the bulk modulus of fluid, for example water), however, for

unsaturated and partially saturated materials, this density is a density of a mixture, as described in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

Note that, the permeability $\mathbf{k}$ is used with dimensions of $[length]^3[time]/[mass]$, which is different from the usual soil mechanics convention, where the permeability has the dimension of velocity, i.e. $[length]/[time]$. Their values are related by $k = K/\rho_f g$, where g is the gravitational acceleration at which the permeability is measured.

More on theory for this finite element can be found in Section 102.12.3.3 on page 156 of the main document. Description of output by this element can be found in Section 206.8.2.

205.3.4.76 Modeling, Finite Element: 8 Node Brick u-p-U Element

The command is:

```
add element # <.> type 8NodeBrick_upU
   using <.> Gauss points each direction
   with nodes (<.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>)
   use material # <.>
   porosity = <.>
   alpha = <.>
   rho_s = <M/L^3>
   rho_f = <M/L^3>
   k_x = <L^3*T/M>
   k_y = <L^3*T/M>
   k_z = <L^3*T/M>
   K_s = <F/L^2>
   K_f = <F/L^2>;
```

and/or

```
add element # <.> type 8NodeBrick_upU
   with nodes (<.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>)
   use material # <.>
   porosity = <.>
   alpha = <.>
   rho_s = <M/L^3>
   rho_f = <M/L^3>
   k_x = <L^3*T/M>
   k_y = <L^3*T/M>
   k_z = <L^3*T/M>
   K_s = <F/L^2>
   K_f = <F/L^2>;
```

where:

- `No (or #)<element_number>` is the unique element integer number that does not have to be sequential, any unique positive integer number can be used.
- `type 8NodeBrick_up` is the element type/name.
- `with nodes (n1, n2, n3, n4, n5, n6, n7, n8)` are the 8 nodes for this element, is specified order.
- `using <.> Gauss points each direction` is the number of Gauss points to be used in each direction (r1, r2, and r3) for integration of finite element matrices (mass and stiffness). There can be from 1 to 6 Gauss points used (uniformly) in each direction (r1, r2, and r3). Command for the brick finite element (above) without number of Gauss points control is kept for back compatibility.

For 8 node bricks 2 Gauss points are used in each direction ($2 \times 3 \times 3$), while for 20 nodes, 8-20 node and 8-27 node bricks 3 Gauss points are used in each direction ($3 \times 3 \times 3$).

- `use material No (or #)` is the material number which makes up the element (nonlinear elastic or elastic-plastic material properties for each integration (Gauss) point will evolve independently as the element deforms). Use LT version with LT materials.
- `porosity` is the porosity ($n = V_{voids}/V_{total}$) of material in this element.
- `alpha` is the parameter controlling level of effective stress analysis. For soils, usually $\alpha = 1$ is used, while for other materials (saturated concrete, bone material, etc.) lower values are used ($0 \leq \alpha \leq 1$).
- `rho_s` is the density of particles of the solid phase. It is important to note that this is a density of the actual mineral that makes up solid particles!.
- `rho_f` is the density of pore fluid. It is usually density of water, however, for unsaturated and partially saturated materials, this density will be different, as described in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).
- `k_x` is the permeability in the x direction (global x) of the element. It is also important to note about the units used for permeability, as noted below. With isotropic permeability, usually the case, $k_x = k_y = k_z$.
- `k_y` is the permeability in the y direction (global y) of the element. It is also important to note about the units used for permeability, as noted below. With isotropic permeability, usually the case, $k_x = k_y = k_z$.
- `k_z` is the permeability in the z direction (global z) of the element. It is also important to note about the units used for permeability, as noted below. With isotropic permeability, usually the case, $k_x = k_y = k_z$.
- `K_s` is the bulk modulus of the soil phase particles. It is important to note that this is a bulk modulus of the actual mineral that makes up solid particles!
- `K_f` is the bulk modulus of the fluid phase that is found in porous material pores. It is usually bulk modulus of the fluid (physical value of the bulk modulus of fluid, for example water), however, for unsaturated and partially saturated materials, this density is a density of a mixture, as described in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

Note that, the permeability $\mathbf{k}$ is used with dimensions of $[length]^3[time]/[mass]$, which is different from the usual soil mechanics convention, where the permeability has the dimension of velocity, i.e. $[length]/[time]$. Their values are related by $k = K/\rho_f g$, where g is the gravitational acceleration at which the permeability is measured.

Please note that the $u-p-U$ element, and the $u-p-U$ formulation is a dynamic formulation and is meant to be used with dynamic analysis, and not static analysis, so that all the element matrices, as described in theory section, noted below, are developed and used.

More on theory for this finite element can be found in Section 102.12.1.7 on page 152 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL). Description of output by this element can be found in Section 206.8.2 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.77 Modeling, Finite Element: 20 Node Brick u-p-U Element

The command is:

```
add element # <.> type 20NodeBrick_upU
   using <.> Gauss points each direction
   with nodes (<.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>,<.>, <.>, <.>, <.>, <.>, ↩
      <.>, <.>, <.>,<.>, <.>, <.>, <.>)
   use material # <.>
   porosity = <.>
   alpha = <.>
   rho_s = <M/L^3>
   rho_f = <M/L^3>
   k_x = <L^3*T/M>
   k_y = <L^3*T/M>
   k_z = <L^3*T/M>
   K_s = <F/L^2>
   K_f = <F/L^2>;
```

and/or

```
add element # <.> type 20NodeBrick_upU
   with nodes (<.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>,<.>, <.>, <.>, <.>, <.>, ↩
      <.>, <.>, <.>,<.>, <.>, <.>, <.>)
   use material # <.>
   porosity = <.>
   alpha = <.>
   rho_s = <M/L^3>
   rho_f = <M/L^3>
   k_x = <L^3*T/M>
   k_y = <L^3*T/M>
   k_z = <L^3*T/M>
   K_s = <F/L^2>
   K_f = <F/L^2>;
```

where:

- `No (or #)<element_number>` is the unique element integer number that does not have to be sequential, any unique positive integer number can be used.
- `type 8NodeBrick_up` is the element type/name.
- `with nodes (n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15, ↩ n16, n17, n18, n19, n20)` are the 20 nodes for this element, is specified order.
- `using <.> Gauss points each direction` is the number of Gauss points to be used in each direction (r1, r2, and r3) for integration of finite element matrices (mass and stiffness). There can be from 1 to 6 Gauss points used (uniformly) in each direction (r1, r2, and r3). Command for the

brick finite element (above) without number of Gauss points control is kept for back compatibility. For 8 node bricks 2 Gauss points are used in each direction ($2 \times 3 \times 3$), while for 20 nodes, 8-20 node and 8-27 node bricks 3 Gauss points are used in each direction ($3 \times 3 \times 3$).

- `use material No (or #)` is the material number which makes up the element (nonlinear elastic or elastic-plastic material properties for each integration (Gauss) point will evolve independently as the element deforms). Use LT version with LT materials.
- `porosity` is the porosity ($n = V_{voids}/V_{total}$) of material in this element.
- `alpha` is the parameter controlling level of effective stress analysis. For soils, usually $\alpha = 1$ is used, while for other materials (saturated concrete, bone material, etc.) lower values are used ($0 \le \alpha \le 1$).
- `rho_s` is the density of particles of the solid phase. It is important to note that this is a density of the actual mineral that makes up solid particles!.
- `rho_f` is the density of pore fluid. It is usually density of water, however, for unsaturated and partially saturated materials, this density will be different, as described in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).
- `k_x` is the permeability in the x direction (global x) of the element. It is also important to note about the units used for permeability, as noted below. With isotropic permeability, usually the case, $k_x = k_y = k_z$.
- `k_y` is the permeability in the y direction (global y) of the element. It is also important to note about the units used for permeability, as noted below. With isotropic permeability, usually the case, $k_x = k_y = k_z$.
- `k_z` is the permeability in the z direction (global z) of the element. It is also important to note about the units used for permeability, as noted below. With isotropic permeability, usually the case, $k_x = k_y = k_z$.
- `K_s` is the bulk modulus of the soil phase particles. It is important to note that this is a bulk modulus of the actual mineral that makes up solid particles!
- `K_f` is the bulk modulus of the fluid phase that is found in porous material pores. It is usually bulk modulus of the fluid (physical value of the bulk modulus of fluid, for example water), however, for unsaturated and partially saturated materials, this density is a density of a mixture, as described in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

Note that, the permeability $\mathbf{k}$ is used with dimensions of $[length]^3[time]/[mass]$, which is different from the usual soil mechanics convention, where the permeability has the dimension of velocity, i.e. $[length]/[time]$. Their values are related by $k = K/\rho_f g$, where g is the gravitational acceleration at which the permeability is measured.

Please note that the $u-p-U$ element, and the $u-p-U$ formulation is a dynamic formulation and is meant to be used with dynamic analysis, and not static analysis, so that all the element matrices, as described in theory section, noted below, are developed and used.

More on theory for this finite element can be found in section 102.12.1.8 on page 152 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL). Description of output by this element can be found in Section 206.8.2.

205.3.4.78 Modeling, Finite Element: 27 Node Brick u-p-U Element

The command is:

```
add element # <.> type 27NodeBrick_upU
   using <.> Gauss points each direction
   with nodes (<.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>,<.>, <.>, <.>, <.>, <.>, ↩
      <.>, <.>, <.>,<.>, <.>, <.>, <.>, <.>, <.>, <.>,<.>, <.>, <.>, <.>)
   use material # <.>
   porosity = <.>
   alpha = <.>
   rho_s = <M/L^3>
   rho_f = <M/L^3>
   k_x = <L^3*T/M>
   k_y = <L^3*T/M>
   k_z = <L^3*T/M>
   K_s = <F/L^2>
   K_f = <F/L^2>;
```

and/or

```
add element # <.> type 27NodeBrick_upU
   with nodes (<.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>,<.>, <.>, <.>, <.>, <.>, ↩
      <.>, <.>, <.>,<.>, <.>, <.>, <.>, <.>, <.>, <.>,<.>, <.>, <.>, <.>)
   use material # <.>
   porosity = <.>
   alpha = <.>
   rho_s = <M/L^3>
   rho_f = <M/L^3>
   k_x = <L^3*T/M>
   k_y = <L^3*T/M>
   k_z = <L^3*T/M>
   K_s = <F/L^2>
   K_f = <F/L^2>;
```

where:

- `No (or #)<element_number>` is the unique element integer number that does not have to be sequential, any unique positive integer number can be used.
- `type 8NodeBrick_up` is the element type/name.
- `with nodes (n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15, ↩ n16, n17, n18, n19, n20, n21, n22, n23, n24, n25, n26, n27)` are the 27 nodes for this element, is specified order.
- `using <.> Gauss points each direction` is the number of Gauss points to be used in each direction (r1, r2, and r3) for integration of finite element matrices (mass and stiffness). There can

be from 1 to 6 Gauss points used (uniformly) in each direction (r1, r2, and r3). Command for the brick finite element (above) without number of Gauss points control is kept for back compatibility. For 8 node bricks 2 Gauss points are used in each direction ($2 \times 3 \times 3$), while for 20 nodes, 8-20 node and 8-27 node bricks 3 Gauss points are used in each direction ($3 \times 3 \times 3$).

- `use material No (or #)` is the material number which makes up the element (nonlinear elastic or elastic-plastic material properties for each integration (Gauss) point will evolve independently as the element deforms). Use LT version with LT materials.
- `porosity` is the porosity ($n = V_{voids}/V_{total}$) of material in this element.
- `alpha` is the parameter controlling level of effective stress analysis. For soils, usually $\alpha = 1$ is used, while for other materials (saturated concrete, bone material, etc.) lower values are used ($0 \leq \alpha \leq 1$).
- `rho_s` is the density of particles of the solid phase. It is important to note that this is a density of the actual mineral that makes up solid particles!.
- `rho_f` is the density of pore fluid. It is usually density of water, however, for unsaturated and partially saturated materials, this density will be different, as described in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).
- `k_x` is the permeability in the x direction (global x) of the element. It is also important to note about the units used for permeability, as noted below. With isotropic permeability, usually the case, $k_x = k_y = k_z$.
- `k_y` is the permeability in the y direction (global y) of the element. It is also important to note about the units used for permeability, as noted below. With isotropic permeability, usually the case, $k_x = k_y = k_z$.
- `k_z` is the permeability in the z direction (global z) of the element. It is also important to note about the units used for permeability, as noted below. With isotropic permeability, usually the case, $k_x = k_y = k_z$.
- `K_s` is the bulk modulus of the soil phase particles. It is important to note that this is a bulk modulus of the actual mineral that makes up solid particles!
- `K_f` is the bulk modulus of the fluid phase that is found in porous material pores. It is usually bulk modulus of the fluid (physical value of the bulk modulus of fluid, for example water), however, for

unsaturated and partially saturated materials, this density is a density of a mixture, as described in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

Note that, the permeability $\mathbf{k}$ is used with dimensions of $[length]^3[time]/[mass]$, which is different from the usual soil mechanics convention, where the permeability has the dimension of velocity, i.e. $[length]/[time]$. Their values are related by $k = K/\rho_f g$, where g is the gravitational acceleration at which the permeability is measured.

Please note that the $u-p-U$ element, and the $u-p-U$ formulation is a dynamic formulation and is meant to be used with dynamic analysis, and not static analysis, so that all the element matrices, as described in theory section, noted below, are developed and used.

More on theory for this finite element can be found in Section 102.12.1.9 on page 152 of the main document. Description of output by this element can be found in Section 206.8.2.

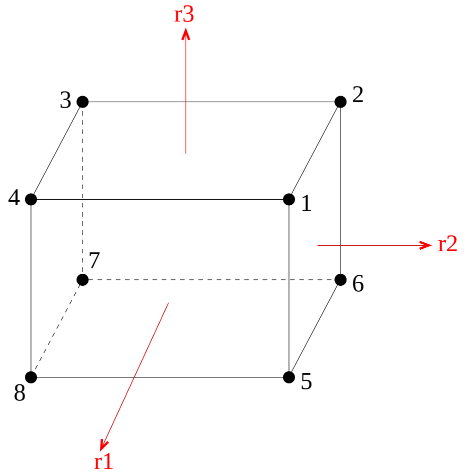
205.3.4.79 Modeling, Finite Element: 8 Node Cosserat Brick Element

The command is:

```
add element # <element_number> type Cosserat8NodeBrick
  with nodes (<.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>)
  use material # <.>;
```

where:

- `<element_number>` is the unique element integer number (does not have to be sequential, any unique positive integer number can be used)
- `type Cosserat8NodeBrick` is the element type.
- `with nodes (n1, n2, n3, n4, n5, n6, n7, n8)` are the 8 nodes for this element. Each node should have 6 DOFs for this element. The element should be in the order as per figure below



- `use material No (or #)` is the material number which makes up the element. The element can use materials `Cosserat_linear_elastic_isotropic_3d` and `Cosserat_von_Mises`.

205.3.4.80 Modeling, Finite Element: Bonded Contact/Interface/Joint Element

The command is:

```
add element # <.> type BondedContact
    with nodes (<.>, <.>)
    penalty_stiffness = <F/L>
```

where

- `penalty_stiffness` represents the penalty stiffness in the three orthogonal x, y and z directions, that connects two nodes of this element.

205.3.4.81 Modeling, Finite Element: Coupled Bonded Contact/Interface/Joint Element

The command is:

```
add element # <.> type CoupledBondedContact
    with nodes (<.>, <.>)
    penalty_stiffness = <F/L>
    axial_viscous_damping = <F/L>
    shear_viscous_damping = <F/L>
    contact_plane_vector = (<.>, <.>, <.> );
```

where

- `penalty_stiffness` represents the penalty stiffness in the three orthogonal x, y and z directions, that connects two nodes of this element. The penalty stiffness is used for both solid and fluid DOFs.
- `axial_viscous_damping` is the viscous damping in axial.
- `shear_viscous_damping` is the viscous damping in shear.
- `contact_plane_vector` defines the normal to the contact/interface/joint plane.

IMPORTANT NOTE No. 1: `contact_plane_vector` defines a direction from Node I to Node J, that is, from the first to the second node. If this normal vector is reversed, the contact/interface/joint element behaves as a hook and is likely to create convergence issues.

IMPORTANT NOTE No. 2: Two nodes that form the Contact/Interface/Joint Element, need to be placed at the same physical location, coordinates in order to prevent convergence issues when nodes are separated and element tries to close the gap in the very first step.

205.3.4.82 Modeling, Finite Element: Force Based Dry Hard Contact/Interface/Joint Element

The command is:

```
add element # <.> type ForceBasedHardContact
   with nodes (<.>, <.>)
   axial_stiffness = <F/L>
   shear_stiffness = <F/L>
   axial_viscous_damping = <F/L>
   shear_viscous_damping = <F/L>
   friction_ratio = <.>
   contact_plane_vector = (<.>, <.>, <.> );
```

The axial force F_a and axial stiffness E_a in defined as

$$F_a = E_a * \delta_a \tag{205.2}$$

where

δ_a refers to the axial relative displacement in axial contact/interface/joint direction,

E_a refers to the axial stiffness in axial contact direction, and

- `axial_stiffness (b)` represents the stiffness in the axial/axial direction (local x axis).
- `shear_stiffness` Is the stiffness in the tangential (shear, local y or z axis) directions.
- `axial_viscous_damping` Is the viscous damping in axial/axial.
- `shear_viscous_damping` Is the viscous damping in shear.
- `friction_ratio` Coulomb friction ratio.
- `contact_plane_vector` Vector defining the normal to the contact/interface/joint plane.

IMPORTANT NOTE No. 1: `contact_plane_vector` defines a direction from Node I to Node J, that is, from the first to the second node. If this normal vector is reversed, the contact/interface/joint element behaves as a hook and is likely to create convergence issues.

IMPORTANT NOTE No. 2: Two nodes that form the Contact/Interface/Joint Element, need to be placed at the same physical location, coordinates in order to prevent convergence issues when nodes are separated and element tries to close the gap in the very first step.

Description of output by this element can be found in Section 206.8.8.

205.3.4.83 Modeling, Finite Element: Force Based Dry Soft Contact/Interface/Joint Element

The command is:

```
add element # <.> type ForceBasedSoftContact
    with nodes (<.>, <.>)
    initial_axial_stiffness = <F/L>
    stiffening_rate = <1/m>
    max_axial_stiffness = <F/L>
    shear_stiffness = <F/L>
    axial_viscous_damping = <F/L>
    shear_viscous_damping = <F/L>
    friction_ratio = <.>
    contact_plane_vector = (<.>, <.>, <.> );
```

The axial force F_a and axial stiffness E_a in defined as

$$\begin{aligned} F_a &= b * exp(a * \delta_a) * \delta_a \\ E_a &= max(b * exp(a * \delta_a) * (1 + a * \delta_a), E_{max}) \end{aligned} \tag{205.3}$$

where

δ_a refers to the axial relative displacement in axial contact/interface/joint direction,

b refers to the axial stiffness in axial contact/interface/joint direction,

a refers to the stiffening rate in axial contact/interface/joint direction,

E_{max} refers to the maximum axial stiffness, and

- `initial_axial_stiffness (b)` represents the stiffness in the axial direction (local x axis).
- `stiffening_rate (a)` Represents exponential stiffening rate $exp(a * \delta_a)$ in axial direction.
- `max_axial_stiffness` (E_{max}) Defines the maximum stiffness in the axial direction (local x axis).
- `shear_stiffness` Is the stiffness in the tangential (shear, local y or z axis) directions.
- `axial_viscous_damping` Is the viscous damping in axial.
- `shear_viscous_damping` Is the viscous damping in shear.
- `friction_ratio` Coulomb friction ratio.
- `contact_plane_vector` Vector defining the normal to the contact/interface/joint plane.

IMPORTANT NOTE No. 1: `contact_plane_vector` defines a direction from Node I to Node J, that is, from the first to the second node. If this normal vector is reversed, the contact/interface/joint element behaves as a hook and is likely to create convergence issues.

IMPORTANT NOTE No. 2: Two nodes that form the Contact/Interface/Joint Element, need to be placed at the same physical location, coordinates in order to prevent convergence issues when nodes are separated and element tries to close the gap in the very first step.

Description of output by this element can be found in Section 206.8.8.

205.3.4.84 Modeling, Finite Element: Force Based Coupled Hard Contact/Interface/Joint Element

The command is:

```
add element # <.> type ForceBasedCoupledHardContact
   with nodes (<.>, <.>)
   axial_stiffness = <F/L>
   axial_penalty_stiffness = <F/L>
   shear_stiffness = <F/L>
   axial_viscous_damping = <F/L>
   shear_viscous_damping = <F/L>
   friction_ratio = <.>
   contact_plane_vector = (<.>, <.>, <.> );
```

The axial force F_a and axial stiffness E_a in defined as

$$
\begin{aligned}
F_a &= b * \delta_a \\
E_a &= b
\end{aligned}
\tag{205.4}
$$

where

δ_a refers to the axial relative displacement in axial contact/interface/joint direction,

b refers to the axial stiffness in axial contact/interface/joint direction, and

- `axial_stiffness (b)` represents the axial stiffness in the axial direction (local x axis).
- `axial_penalty_stiffness` (E_p) defines the penalty stiffness between U_i degree of freedom (DoF) (saturated, coupled u-p-U element) and u_i DoF (dry u element) to enforce movement of fluid in u-p-U element with solid in u element in contact/interface/joint axial direction. This is useful for pumping action, gap opens and draws the fluid from u-p-U element and then gap closes and pumps, pushes fluid into u-p-U element.
- `shear_stiffness` Is the stiffness in the tangential (shear, local y or z axis) directions.
- `axial_viscous_damping` Is the viscous damping in axial.
- `shear_viscous_damping` Is the viscous damping in shear.
- `friction_ratio` Coulomb friction ratio.
- `contact_plane_vector` Vector defining the normal to the contact/interface/joint plane.

IMPORTANT NOTE No. 1: `contact_plane_vector` defines a direction from Node I to Node J, that is, from the first to the second node. If this normal vector is reversed, the contact/interface/joint element behaves as a hook and is likely to create convergence issues.

IMPORTANT NOTE No. 2: Two nodes that form the Contact/Interface/Joint Element, need to be placed at the same physical location, coordinates in order to prevent convergence issues when nodes are separated and element tries to close the gap in the very first step.

Description of output by this element can be found in Section 206.8.8.

205.3.4.85 Modeling, Finite Element: Force Based Coupled Soft Contact/Interface/Joint Element

The command is:

```
add element # <.> type ForceBasedCoupledSoftContact
   with nodes (<.>, <.>)
   initial_axial_stiffness = <F/L>
   stiffening_rate = <1/m>
   max_axial_stiffness = <F/L>
   axial_penalty_stiffness = <F/L>
   shear_stiffness = <F/L>
   axial_viscous_damping = <F/L>
   shear_viscous_damping = <F/L>
   friction_ratio = <.>
   contact_plane_vector = (<.>, <.>, <.> );
```

The axial force F_a and axial stiffness E_a in defined as

$$\begin{aligned} F_a &= b * exp(a * \delta_a) * \delta_a \\ E_a &= max(b * exp(a * \delta_a) * (1 + a * \delta_a), E_{max}) \end{aligned} \tag{205.5}$$

where

δ_a refers to the axial relative displacement in axial contact/interface/joint direction,

b refers to the axial stiffness in axial contact/interface/joint direction,

a refers to the stiffening rate in axial contact/interface/joint direction,

E_{max} refers to the maximum axial stiffness, and

- `initial_axial_stiffness (b)` represents the stiffness in the axial direction (local x axis).
- `stiffening_rate (a)` Represents exponential stiffening rate $exp(a * \delta_a)$ in axial direction.
- `max_axial_stiffness` (E_{max}) Defines the maximum stiffness in the axial direction (local x axis).
- `axial_penalty_stiffness` (E_p) defines the penalty stiffness between U_i degree of freedom (DoF) (saturated, coupled u-p-U element) and u_i DoF (dry u element) to enforce movement of fluid in u-p-U element with solid in u element in contact/interface/joint axial direction. This is useful for pumping action, gap opens and draws the fluid from u-p-U element and then gap closes and pumps, pushes fluid into u-p-U element.
- `shear_stiffness` Is the stiffness in the tangential (shear, local y or z axis) directions.
- `axial_viscous_damping` Is the viscous damping in axial.
- `shear_viscous_damping` Is the viscous damping in shear.

- `friction_ratio` Coulomb friction ratio.
- `contact_plane_vector` Vector defining the normal to the contact/interface/joint plane.

IMPORTANT NOTE No. 1: `contact_plane_vector` defines a direction from Node I to Node J, that is, from the first to the second node. If this normal vector is reversed, the contact/interface/joint element behaves as a hook and is likely to create convergence issues.

IMPORTANT NOTE No. 2: Two nodes that form the Contact/Interface/Joint Element, need to be placed at the same physical location, coordinates in order to prevent convergence issues when nodes are separated and element tries to close the gap in the very first step.

Description of output by this element can be found in Section 206.8.8.

205.3.4.86 Modeling, Finite Element: Stress Based Dry Hard Contact/Interface/Joint Element with Elastic Perfectly Plastic Shear Behavior

The command is:

```
add element # <.> type StressBasedHardContact_ElPPlShear
   with nodes (<.>, <.>)
   axial_stiffness = <Pa>
   initial_shear_stiffness = <Pa>
   axial_viscous_damping = <Pa*s>
   shear_viscous_damping = <Pa*s>
   residual_friction_coefficient = <.>
   shear_zone_thickness = <m>
   contact_plane_vector = (<.>, <.>, <.> );
```

and/or;

```
add element # <.> type StressBasedHardContact_ElPPlShear
   with nodes (<.>, <.>)
   axial_stiffness = <Pa>
   initial_shear_stiffness = <Pa>
   axial_viscous_damping = <Pa*s>
   shear_viscous_damping = <Pa*s>
   residual_friction_coefficient = <.>
   shear_zone_thickness = <m>
   surface_vector_relative_tolerance = <.>;
```

The axial stress σ_a and axial stiffness E_a in defined as

$$\begin{aligned} \sigma_a &= b * \epsilon_a \\ E_a &= b \end{aligned} \tag{205.6}$$

where

$E_a = b$ refers to the axial stiffness in axial contact/interface/joint direction,

ϵ_a refers to the axial strain in axial contact/interface/joint direction $\epsilon_a = \delta_a/h$,

δ_a is the relative axial penetration in contact axial direction,

h is the shear zone thickness, and

- `axial_stiffness (b)` represents the stiffness in the axial direction (local x axis).
- `initial_shear_stiffness` (E_s) Is the stiffness in the tangential (shear, local y or z axis) directions at $101kPa$ axial stress described in Section 104.7.3.2
- `axial_viscous_damping` Is the viscous damping in axial.

- `shear_viscous_damping` Is the viscous damping in shear.
- `residual_friction_coefficient` (μ_r) Is the residual friction coefficient described in Section 104.7.3.2.
- `shear_zone_thickness` h Is the shear zone thickness.
- `contact_plane_vector` Vector defining the normal to the contact/interface/joint plane.
- `surface_vector_relative_tolerance` defines the relative tolerance to find all the contact/interface/joint normals and create multiple contact elements for a given contact node pairs for a conforming surface-to-surface mesh.

IMPORTANT NOTE No. 1: `contact_plane_vector` defines a direction from Node I to Node J, that is, from the first to the second node. If this normal vector is reversed, the contact/interface/joint element behaves as a hook and is likely to create convergence issues.

IMPORTANT NOTE No. 2: Two nodes that form the Contact/Interface/Joint Element, need to be placed at the same physical location, coordinates in order to prevent convergence issues when nodes are separated and element tries to close the gap in the very first step.

Description of output by this element can be found in Section 206.8.8.

205.3.4.87 Modeling, Finite Element: Stress Based Dry Hard Contact/Interface/Joint Element with Nonlinear Hardening Shear Behavior

The command is:

```
add element # <.> type StressBasedHardContact_NonLinHardShear
   with nodes (<.>, <.>)
   axial_stiffness = <Pa>
   initial_shear_stiffness = <Pa>
   axial_viscous_damping = <Pa*s>
   shear_viscous_damping = <Pa*s>
   residual_friction_coefficient = <.>
   shear_zone_thickness = <m>
   contact_plane_vector = (<.>, <.>, <.> );
```

and/or;

```
add element # <.> type StressBasedHardContact_NonLinHardShear
   with nodes (<.>, <.>)
   axial_stiffness = <Pa>
   initial_shear_stiffness = <Pa>
   axial_viscous_damping = <Pa*s>
   shear_viscous_damping = <Pa*s>
   residual_friction_coefficient = <.>
   shear_zone_thickness = <m>
   surface_vector_relative_tolerance = <.>;
```

The axial stress σ_a and axial stiffness E_a in defined as

$$\begin{aligned} \sigma_a &= b * \epsilon_a \\ E_a &= b \end{aligned} \tag{205.7}$$

where

$E_a = b$ refers to the axial stiffness in axial contact/interface/joint direction,

ϵ_a refers to the axial strain in axial contact direction $\epsilon_a = \delta_a/h$,

δ_a is the relative axial penetration in contact/interface/joint axial direction,

h is the shear zone thickness, and

- `axial_stiffness (b)` represents the stiffness in the axial direction (local x axis).
- `initial_shear_stiffness` (E_s) Is the stiffness in the tangential (shear, local y or z axis) directions at $101 kPa$ axial stress, described in Section 104.7.3.4
- `axial_viscous_damping` Is the viscous damping in axial.

- `shear_viscous_damping` Is the viscous damping in shear.
- `residual_friction_coefficient` (μ_r) Is the residual frictional parameter as described in Section 104.7.3.3
- `shear_zone_thickness` h Is the shear zone thickness.
- `contact_plane_vector` Vector defining the normal to the contact/interface/joint plane.
- `surface_vector_relative_tolerance` defines the relative tolerance to find all the contact/interface/joint normals and create multiple contact elements for a given contact node pairs for a conforming surface-to-surface mesh.

IMPORTANT NOTE No. 1: `contact_plane_vector` defines a direction from Node I to Node J, that is, from the first to the second node. If this normal vector is reversed, the contact/interface/joint element behaves as a hook and is likely to create convergence issues.

IMPORTANT NOTE No. 2: Two nodes that form the Contact/Interface/Joint Element, need to be placed at the same physical location, coordinates in order to prevent convergence issues when nodes are separated and element tries to close the gap in the very first step.

Description of output by this element can be found in Section 206.8.8.

205.3.4.88 Modeling, Finite Element: Stress Based Dry Hard Contact/Interface/Joint Element with Nonlinear Hardening and Softening Shear Behavior

The command is:

```
add element # <.> type StressBasedHardContact_NonLinHardSoftShear
    with nodes (<.>, <.>)
    axial_stiffness = <Pa>
    initial_shear_stiffness = <Pa>
    rate_of_softening = <>
    size_of_peak_plateau = <>
    axial_viscous_damping = <Pa*s>
    shear_viscous_damping = <Pa*s>
    peak_friction_coefficient_limit = <>
    peak_friction_coefficient_rate_of_decrease = <.>
    residual_friction_coefficient = <.>
    shear_zone_thickness = <m>
    contact_plane_vector = (<.>, <.>, <.> );
```

and/or;

```
add element # <.> type StressBasedHardContact_NonLinHardSoftShear
    with nodes (<.>, <.>)
    axial_stiffness = <Pa>
    initial_shear_stiffness = <Pa>
    rate_of_softening = <>
    size_of_peak_plateau = <>
    axial_viscous_damping = <Pa*s>
    shear_viscous_damping = <Pa*s>
    peak_friction_coefficient_limit = <>
    peak_friction_coefficient_rate_of_decrease = <.>
    residual_friction_coefficient = <.>
    shear_zone_thickness = <m>
    surface_vector_relative_tolerance = <.>;
```

The axial stress σ_a and axial stiffness E_a in defined as

$$\begin{aligned} \sigma_a &= b * \epsilon_a \\ E_a &= b \end{aligned} \tag{205.8}$$

where

$E_a = b$ refers to the axial stiffness in axial contact/interface/joint direction,

ϵ_a refers to the axial strain in axial contact direction $\epsilon_a = \delta_a/h$,

δ_a is the relative axial penetration in contact/interface/joint axial direction,

h is the shear zone thickness, and

- `axial_stiffness (b)` represents the stiffness in the axial direction (local x axis).

- `initial_shear_stiffness` (E_s) Is the stiffness in the tangential (shear, local y or z axis) directions at $101kPa$ axial stress, described in Section 104.7.3.4

- `rate_of_softening` (R_s) Is the parameter to control the rate of frictional softening described in Section 104.7.3.4. The frictional softening function is an inverse tangent function raised to power n with incremental form as

$$\Delta\mu = -\frac{n * R_s(\mu_p - \mu_r)}{(\pi/2)^n \theta^{1/n-1}} * cos^2\theta\Delta\gamma^p \tag{205.9}$$

$$\theta = \frac{\mu_p - \mu}{\mu_p - \mu_r}(\pi/2)^n \tag{205.10}$$

  where, R_s is the frictional softening rate parameter, $\Delta\gamma^p$ is the plastic shear strain and n represents the size of the peak plateau.

$$\Delta\gamma^p = \sqrt{\Delta\epsilon^p_{ij}\Delta\epsilon^p_{ij}} \tag{205.11}$$

- `size_of_peak_plateau` (n) Is the frictional softening parameter to control the size of plateau as described in Section 104.7.3.4. The frictional softening function is an inverse tangent function raised to power n with incremental form as shown in Equation 205.17.

- `axial_viscous_damping` Is the viscous damping in axial.

- `shear_viscous_damping` Is the viscous damping in shear.

- `peak_friction_coefficient_limit` (μ_{p0}) Is the limit to the peak frictional hardening parameter μ_p.

- `peak_friction_coefficient_rate_of_decrease` (k) Is the rate of decrease of peak frictional hardening parameter μ_p with axial stress, described in Section 104.7.3.4

$$\mu_p = max(\mu_{p0}, \mu_{p0} - k * log(\sigma_a/P_0)) \tag{205.12}$$

  where μ_{p0} is the peak frictional hardening limit, k is the peak frictional parameter rate of decrease and P_0 is the reference stress of $P_0 = 101kPa$.

- `residual_friction_coefficient` (μ_r) Is the residual frictional parameter as described in Section 104.7.3.4

- `shear_zone_thickness` h Is the shear zone thickness.

- `contact_plane_vector` Vector defining the normal to the contact/interface/joint plane.

- `surface_vector_relative_tolerance` defines the relative tolerance to find all the contact/interface/joint normals and create multiple contact elements for a given contact node pairs for a conforming surface-to-surface mesh.

IMPORTANT NOTE No. 1: `contact_plane_vector` defines a direction from Node I to Node J, that is, from the first to the second node. If this normal vector is reversed, the contact/interface/joint element behaves as a hook and is likely to create convergence issues.

IMPORTANT NOTE No. 2: Two nodes that form the Contact/Interface/Joint Element, need to be placed at the same physical location, coordinates in order to prevent convergence issues when nodes are separated and element tries to close the gap in the very first step.

Description of output by this element can be found in Section 206.8.8.

205.3.4.89 Modeling, Finite Element: Stress Based Dry Soft Contact/Interface/Joint Element with Elastic Perfectly Plastic Shear Behavior

The command is:

```
add element # <.> type StressBasedSoftContact_ElPPlShear
   with nodes (<.>, <.>)
   initial_axial_stiffness = <Pa>
   stiffening_rate = <>
   max_axial_stiffness = <Pa>
   initial_shear_stiffness = <Pa>
   axial_viscous_damping = <Pa*s>
   shear_viscous_damping = <Pa*s>
   residual_friction_coefficient = <.>
   shear_zone_thickness = <m>
   contact_plane_vector = (<.>, <.>, <.> );
```

and/or;

```
add element # <.> type StressBasedSoftContact_ElPPlShear
   with nodes (<.>, <.>)
   initial_axial_stiffness = <Pa>
   stiffening_rate = <>
   max_axial_stiffness = <Pa>
   initial_shear_stiffness = <Pa>
   axial_viscous_damping = <Pa*s>
   shear_viscous_damping = <Pa*s>
   residual_friction_coefficient = <.>
   shear_zone_thickness = <m>
   surface_vector_relative_tolerance = <.>;
```

The axial stress σ_a and axial stiffness E_a in defined as

$$\begin{aligned} \sigma_a &= b * exp(a * \epsilon_a) * \epsilon_a \\ E_a &= max(b * exp(a * \epsilon_a) * (1 + a * \epsilon_a), E_{max}) \end{aligned} \tag{205.13}$$

where

b refers to the initial axial stiffness in axial contact/interface/joint direction,

a refers to the stiffening rate in axial contact direction,

E_{max} refers to the maximum axial stiffness,

E_a refers to the axial stiffness,

ϵ_a refers to the axial strain in axial contact/interface/joint direction $\epsilon_a = \delta_a/h$,

δ_a is the relative axial penetration in contact axial direction,

h is the shear zone thickness, and

- `initial_axial_stiffness (b)` represents the stiffness in the axial direction (local x axis).
- `stiffening_rate (a)` Represents exponential stiffening rate $exp(a * \epsilon_a)$ in axial direction.
- `max_axial_stiffness`(E_{max}) Defines the maximum stiffness in the axial direction (local x axis) for the contact/interface/joint element.
- `initial_shear_stiffness` (E_s) Is the stiffness in the tangential (shear, local y or z axis) directions at $101 kPa$ axial stress described in Section 104.7.3.2
- `axial_viscous_damping` Is the viscous damping in axial.
- `shear_viscous_damping` Is the viscous damping in shear.
- `residual_friction_coefficient` (μ_r) Is the residual friction coefficient described in Section 104.7.3.2
- `shear_zone_thickness` h Is the shear zone thickness
- `contact_plane_vector` Vector defining the normal to the contact/interface/joint plane.
- `surface_vector_relative_tolerance` defines the relative tolerance to find all the contact/interface/joint normals and create multiple contact elements for a given contact node pairs for a conforming surface-to-surface mesh.

IMPORTANT NOTE No. 1: `contact_plane_vector` defines a direction from Node I to Node J, that is, from the first to the second node. If this normal vector is reversed, the contact/interface/joint element behaves as a hook and is likely to create convergence issues.

IMPORTANT NOTE No. 2: Two nodes that form the Contact/Interface/Joint Element, need to be placed at the same physical location, coordinates in order to prevent convergence issues when nodes are separated and element tries to close the gap in the very first step.

Description of output by this element can be found in Section 206.8.8.

205.3.4.90 Modeling, Finite Element: Stress Based Dry Soft Contact/Interface/Joint Element with Nonlinear Hardening Shear Behavior

The command is:

```
add element # <.> type StressBasedSoftContact_NonLinHardShear
   with nodes (<.>, <.>)
   initial_axial_stiffness = <Pa>
   stiffening_rate = <>
   max_axial_stiffness = <Pa>
   initial_shear_stiffness = <Pa>
   axial_viscous_damping = <Pa*s>
   shear_viscous_damping = <Pa*s>
   residual_friction_coefficient = <.>
   shear_zone_thickness = <m>
   contact_plane_vector = (<.>, <.>, <.> );
```

and/or;

```
add element # <.> type StressBasedSoftContact_NonLinHardShear
   with nodes (<.>, <.>)
   initial_axial_stiffness = <Pa>
   stiffening_rate = <>
   max_axial_stiffness = <Pa>
   initial_shear_stiffness = <Pa>
   axial_viscous_damping = <Pa*s>
   shear_viscous_damping = <Pa*s>
   residual_friction_coefficient = <.>
   shear_zone_thickness = <m>
   surface_vector_relative_tolerance = <.>;
```

The axial stress σ_a and axial stiffness E_a in defined as

$$\begin{aligned}\sigma_a &= b * exp(a * \epsilon_a) * \epsilon_a \\ E_a &= max(b * exp(a * \epsilon_a) * (1 + a * \epsilon_a), E_{max})\end{aligned} \tag{205.14}$$

where

b refers to the initial axial stiffness in axial contact/interface/joint direction,

a refers to the stiffening rate in axial contact direction,

E_{max} refers to the maximum axial stiffness,

E_a refers to the axial stiffness,

ϵ_a refers to the axial strain in axial contact/interface/joint direction $\epsilon_a = \delta_a/h$,

δ_a is the relative axial penetration in contact axial direction,

h is the shear zone thickness, and

- `initial_axial_stiffness (b)` represents the stiffness in the axial direction (local x axis) for 1m penetration.
- `stiffening_rate (a)` Represents exponential stiffening rate $exp(sr * \epsilon_a)$ in axial direction.
- `max_axial_stiffness`(E_{max}) Defines the maximum stiffness in the axial direction (local x axis) for the contact/interface/joint element.
- `initial_shear_stiffness` (E_s) Is the stiffness in the tangential (shear, local y or z axis) directions at $101kPa$ axial stress described in Section 104.7.3.3
- `axial_viscous_damping` Is the viscous damping in axial.
- `shear_viscous_damping` Is the viscous damping in shear.
- `residual_friction_coefficient` (μ_r) Is the residual frictional parameter as described in Section 104.7.3.3
- `shear_zone_thickness` h Is the shear zone thickness
- `contact_plane_vector` Vector defining the normal to the contact/interface/joint plane.
- `surface_vector_relative_tolerance` defines the relative tolerance to find all the contact/interface/joint normals and create multiple contact elements for a given contact node pairs for a conforming surface-to-surface mesh.

IMPORTANT NOTE No. 1: `contact_plane_vector` defines a direction from Node I to Node J, that is, from the first to the second node. If this normal vector is reversed, the contact/interface/joint element behaves as a hook and is likely to create convergence issues.

IMPORTANT NOTE No. 2: Two nodes that form the Contact/Interface/Joint Element, need to be placed at the same physical location, coordinates in order to prevent convergence issues when nodes are separated and element tries to close the gap in the very first step.

Description of output by this element can be found in Section 206.8.8.

205.3.4.91 Modeling, Finite Element: Stress Based Dry Soft Contact/Interface/Joint Element with Nonlinear Hardening and Softening Shear Behavior

The command is:

```
add element # <.> type StressBasedSoftContact_NonLinHardSoftShear
   with nodes (<.>, <.>)
   initial_axial_stiffness = <Pa>
   stiffening_rate = <>
   max_axial_stiffness = <Pa>
   initial_shear_stiffness = <Pa>
   rate_of_softening = <>
   size_of_peak_plateau = <>
   axial_viscous_damping = <Pa*s>
   shear_viscous_damping = <Pa*s>
   peak_friction_coefficient_limit = <>
   peak_friction_coefficient_rate_of_decrease = <.>
   residual_friction_coefficient = <.>
   shear_zone_thickness = <m>
   contact_plane_vector = (<.>, <.>, <.> );
```

and/or;

```
add element # <.> type StressBasedSoftContact_NonLinHardSoftShear
   with nodes (<.>, <.>)
   initial_axial_stiffness = <Pa>
   stiffening_rate = <>
   max_axial_stiffness = <Pa>
   initial_shear_stiffness = <Pa>
   rate_of_softening = <>
   size_of_peak_plateau = <>
   axial_viscous_damping = <Pa*s>
   shear_viscous_damping = <Pa*s>
   peak_friction_coefficient_limit = <>
   peak_friction_coefficient_rate_of_decrease = <.>
   residual_friction_coefficient = <.>
   shear_zone_thickness = <m>
   surface_vector_relative_tolerance = <.>;
```

The axial stress σ_a and axial stiffness E_a in defined as

$$\begin{aligned}
\sigma_a &= b * exp(a * \epsilon_a) * \epsilon_a \\
E_a &= max(b * exp(a * \epsilon_a) * (1 + a * \epsilon_a), E_{max})
\end{aligned} \tag{205.15}$$

where

b refers to the initial axial stiffness in axial contact/interface/joint direction,

a refers to the stiffening rate in axial contact direction,

E_{max} refers to the maximum axial stiffness,

E_a refers to the axial stiffness,

ϵ_a refers to the axial strain in axial contact/interface/joint direction $\epsilon_a = \delta_a/h$,

δ_a is the relative axial penetration in contact axial direction,

h is the shear zone thickness, and

- `initial_axial_stiffness (b)` represents the stiffness in the axial direction (local x axis) for 1m penetration.
- `stiffening_rate (a)` Represents exponential stiffening rate $exp(sr * \epsilon_n)$ in axial direction.
- `max_axial_stiffness`(E_{max}) Defines the maximum stiffness in the axial direction (local x axis) for the contact/interface/joint element.
- `initial_shear_stiffness` (E_s) Is the stiffness in the tangential (shear, local y or z axis) directions at $101kPa$ axial stress, described in Section 104.7.3.4
- `rate_of_softening` (R_s) Is the parameter to control the rate of frictional softening described in Section 104.7.3.4. The frictional softening function is an inverse tangent function raised to power n with incremental form as

$$\Delta\mu = -\frac{n * R_s(\mu_p - \mu_r)}{(\pi/2)^n \theta^{1/n-1}} * cos^2\theta \Delta\gamma^p \tag{205.16}$$

$$\theta = \frac{\mu_p - \mu}{\mu_p - \mu_r}(\pi/2)^n \tag{205.17}$$

where, R_s is the frictional softening rate parameter, $\Delta\gamma^p$ is the plastic shear strain and n represents the size of the peak plateau.

$$\Delta\gamma^p = \sqrt{\Delta\epsilon^p_{ij}\Delta\epsilon^p_{ij}} \tag{205.18}$$

- `size_of_peak_plateau` (n) Is the frictional softening parameter to control the size of plateau as described in Section 104.7.3.4. The frictional softening function is an inverse tangent function raised to power n with incremental form as shown in Equation 205.17.
- `axial_viscous_damping` Is the viscous damping in axial.
- `shear_viscous_damping` Is the viscous damping in shear.
- `peak_friction_coefficient_limit` (μ_{p0}) Is the limit to the peak frictional hardening parameter μ_p.

- `peak_friction_coefficient_rate_of_decrease` (k) Is the rate of decrease of peak frictional hardening parameter μ_p with axial stress, described in Section 104.7.3.4

  $$\mu_p = max(\mu_{p0}, \mu_{p0} - k * log(\sigma_a/P_0)) \tag{205.19}$$

  where μ_{p0} is the peak frictional hardening limit, k is the peak frictional parameter rate of decrease and P_0 is the reference stress of $P_0 = 101kPa$.

- `residual_friction_coefficient` (μ_r) Is the residual frictional parameter as described in Section 104.7.3.4

- `shear_zone_thickness` h Is the shear zone thickness

- `contact_plane_vector` Vector defining the normal to the contact/interface/joint plane.

- `surface_vector_relative_tolerance` defines the relative tolerance to find all the contact/interface/joint normals and create multiple contact elements for a given contact node pairs for a conforming surface-to-surface mesh.

IMPORTANT NOTE No. 1: `contact_plane_vector` defines a direction from Node I to Node J, that is, from the first to the second node. If this normal vector is reversed, the contact/interface/joint element behaves as a hook and is likely to create convergence issues.

IMPORTANT NOTE No. 2: Two nodes that form the Contact/Interface/Joint Element, need to be placed at the same physical location, coordinates in order to prevent convergence issues when nodes are separated and element tries to close the gap in the very first step.

Description of output by this element can be found in Section 206.8.8.

205.3.4.92 Modeling, Finite Element: Stress Based Coupled Hard Contact/Interface/Joint Element with Elastic Perfectly Plastic Shear Behavior

The command is:

```
add element # <.> type StressBasedCoupledHardContact_ElPPlShear
   with nodes (<.>, <.>)
   axial_stiffness = <Pa>
   axial_penalty_stiffness = <Pa>
   initial_shear_stiffness = <Pa>
   axial_viscous_damping = <Pa*s>
   shear_viscous_damping = <Pa*s>
   residual_friction_coefficient = <.>
   shear_zone_thickness = <m>
   contact_plane_vector = (<.>, <.>, <.> );
```

and/or;

```
add element # <.> type StressBasedCoupledHardContact_ElPPlShear
   with nodes (<.>, <.>)
   axial_stiffness = <Pa>
   axial_penalty_stiffness = <Pa>
   initial_shear_stiffness = <Pa>
   axial_viscous_damping = <Pa*s>
   shear_viscous_damping = <Pa*s>
   residual_friction_coefficient = <.>
   shear_zone_thickness = <m>
   surface_vector_relative_tolerance = <.>;
```

The axial stress σ_a and axial stiffness E_a in defined as

$$\begin{aligned} \sigma_a &= b * \epsilon_a \\ E_a &= b \end{aligned} \tag{205.20}$$

where

$E_a = b$ refers to the axial stiffness in axial contact/interface/joint direction,

ϵ_a refers to the axial strain in axial contact direction $\epsilon_a = \delta_a / h$,

δ_a is the relative axial penetration in contact/interface/joint axial direction,

h is the shear zone thickness, and

- `axial_stiffness` (b) represents the stiffness in the axial direction (local x axis).
- `axial_penalty_stiffness` (E_p) defines the penalty stiffness between U_i degree of freedom (DoF) (saturated, coupled u-p-U element) and u_i DoF (dry u element) to enforce movement of fluid in u-p-U element with solid in u element in contact/interface/joint axial direction. This is

useful for pumping action, gap opens and draws the fluid from u-p-U element and then gap closes and pumps, pushes fluid into u-p-U element.

- `initial_shear_stiffness` (E_s) Is the stiffness in the tangential (shear, local y or z axis) directions at $101kPa$ axial stress described in Section 104.7.3.2
- `axial_viscous_damping` Is the viscous damping in axial.
- `shear_viscous_damping` Is the viscous damping in shear.
- `residual_friction_coefficient` (μ_r) Is the residual friction coefficient described in Section 104.7.3.2
- `shear_zone_thickness` h Is the shear zone thickness
- `contact_plane_vector` Vector defining the normal to the contact/interface/joint plane.
- `surface_vector_relative_tolerance` defines the relative tolerance to find all the contact/interface/joint normals and create multiple contact elements for a given contact node pairs for a conforming surface-to-surface mesh.

IMPORTANT NOTE No. 1: `contact_plane_vector` defines a direction from Node I to Node J, that is, from the first to the second node. If this normal vector is reversed, the contact/interface/joint element behaves as a hook and is likely to create convergence issues.

IMPORTANT NOTE No. 2: Two nodes that form the Contact/Interface/Joint Element, need to be placed at the same physical location, coordinates in order to prevent convergence issues when nodes are separated and element tries to close the gap in the very first step.

Description of output by this element can be found in Section 206.8.8.

205.3.4.93 Modeling, Finite Element: Stress Based Coupled Hard Contact/Interface/Joint Element with Nonlinear Hardening Shear Behavior

The command is:

```
add element # <.> type StressBasedCoupledHardContact_NonLinHardShear
   with nodes (<.>, <.>)
   axial_stiffness = <Pa>
   axial_penalty_stiffness = <Pa>
   initial_shear_stiffness = <Pa>
   axial_viscous_damping = <Pa*s>
   shear_viscous_damping = <Pa*s>
   residual_friction_coefficient = <.>
   shear_zone_thickness = <m>
   contact_plane_vector = (<.>, <.>, <.> );
```

and/or;

```
add element # <.> type StressBasedCoupledHardContact_NonLinHardShear
   with nodes (<.>, <.>)
   axial_stiffness = <Pa>
   axial_penalty_stiffness = <Pa>
   initial_shear_stiffness = <Pa>
   axial_viscous_damping = <Pa*s>
   shear_viscous_damping = <Pa*s>
   residual_friction_coefficient = <.>
   shear_zone_thickness = <m>
   surface_vector_relative_tolerance = <.>;
```

The axial stress σ_a and axial stiffness E_a in defined as

$$\begin{aligned} \sigma_a &= b * \epsilon_a \\ E_a &= b \end{aligned} \tag{205.21}$$

where

$E_a = b$ refers to the axial stiffness in axial contact/interface/joint direction,

ϵ_a refers to the axial strain in axial contact direction $\epsilon_a = \delta_a/h$,

δ_a is the relative axial penetration in contact/interface/joint axial direction,

h is the shear zone thickness, and

- `axial_stiffness (b)` represents the stiffness in the axial direction (local x axis) for 1m penetration.

- `axial_penalty_stiffness` (E_p) defines the penalty stiffness between U_i degree of freedom (DoF) (saturated, coupled u-p-U element) and u_i DoF (dry u element) to enforce movement of fluid in u-p-U element with solid in u element in contact/interface/joint axial direction. This is useful for pumping action, gap opens and draws the fluid from u-p-U element and then gap closes and pumps, pushes fluid into u-p-U element.

- `initial_shear_stiffness` (E_s) Is the stiffness in the tangential (shear, local y or z axis) directions at $101kPa$ axial stress described in Section 104.7.3.3
- `axial_viscous_damping` Is the viscous damping in axial.
- `shear_viscous_damping` Is the viscous damping in shear.
- `residual_friction_coefficient` (μ_r) Is the residual frictional parameter as described in Section 104.7.3.3
- `shear_zone_thickness` h Is the shear zone thickness
- `contact_plane_vector` Vector defining the normal to the contact/interface/joint plane.
- `surface_vector_relative_tolerance` defines the relative tolerance to find all the contact/interface/joint normals and create multiple contact elements for a given contact node pairs for a conforming surface-to-surface mesh.

IMPORTANT NOTE No. 1: `contact_plane_vector` defines a direction from Node I to Node J, that is, from the first to the second node. If this normal vector is reversed, the contact/interface/joint element behaves as a hook and is likely to create convergence issues.

IMPORTANT NOTE No. 2: Two nodes that form the Contact/Interface/Joint Element, need to be placed at the same physical location, coordinates in order to prevent convergence issues when nodes are separated and element tries to close the gap in the very first step.

Description of output by this element can be found in Section 206.8.8.

205.3.4.94 Modeling, Finite Element: Stress Based Coupled Hard Contact/Interface/Joint Element with Nonlinear Hardening and Softening Shear Behavior

The command is:

```
add element # <.> type StressBasedCoupledHardContact_NonLinHardSoftShear
    with nodes (<.>, <.>)
    axial_stiffness = <Pa>
    axial_penalty_stiffness =<Pa>
    initial_shear_stiffness = <Pa>
    rate_of_softening = <>
    size_of_peak_plateau = <>
    axial_viscous_damping = <Pa*s>
    shear_viscous_damping = <Pa*s>
    peak_friction_coefficient_limit = <>
    peak_friction_coefficient_rate_of_decrease = <.>
    residual_friction_coefficient = <.>
    shear_zone_thickness = <m>
    contact_plane_vector = (<.>, <.>, <.> );
```

and/or;

```
add element # <.> type StressBasedCoupledHardContact_NonLinHardSoftShear
    with nodes (<.>, <.>)
    axial_stiffness = <Pa>
    axial_penalty_stiffness =<Pa>
    initial_shear_stiffness = <Pa>
    rate_of_softening = <>
    size_of_peak_plateau = <>
    axial_viscous_damping = <Pa*s>
    shear_viscous_damping = <Pa*s>
    peak_friction_coefficient_limit = <>
    peak_friction_coefficient_rate_of_decrease = <.>
    residual_friction_coefficient = <.>
    shear_zone_thickness = <m>
    surface_vector_relative_tolerance = <.>;
```

The axial stress σ_a and axial stiffness E_a in defined as

$$\begin{aligned} \sigma_a &= b * \epsilon_a \\ E_a &= b \end{aligned} \tag{205.22}$$

where

$E_a = b$ refers to the axial stiffness in axial contact/interface/joint direction,

ϵ_a refers to the axial strain in axial contact direction $\epsilon_a = \delta_a/h$,

δ_a is the relative axial penetration in contact/interface/joint axial direction,

h is the shear zone thickness, and

- `axial_stiffness` (b) represents the stiffness in the axial direction (local x axis) for 1m penetration.

- `axial_penalty_stiffness` (E_p) defines the penalty stiffness between U_i degree of freedom (DoF) (saturated, coupled u-p-U element) and u_i DoF (dry u element) to enforce movement of fluid in u-p-U element with solid in u element in contact/interface/joint axial direction. This is useful for pumping action, gap opens and draws the fluid from u-p-U element and then gap closes and pumps, pushes fluid into u-p-U element.

- `initial_shear_stiffness` (E_s) Is the stiffness in the tangential (shear, local y or z axis) directions at $101kPa$ axial stress, described in Section 104.7.3.4

- `rate_of_softening` (R_s) Is the parameter to control the rate of frictional softening described in Section 104.7.3.4. The frictional softening function is an inverse tangent function raised to power n with incremental form as

$$\Delta\mu = -\frac{n * R_s(\mu_p - \mu_r)}{(\pi/2)^n \theta^{1/n-1}} * cos^2\theta\Delta\gamma^p \tag{205.23}$$

$$\theta = \frac{\mu_p - \mu}{\mu_p - \mu_r}(\pi/2)^n \tag{205.24}$$

  where, R_s is the frictional softening rate parameter, $\Delta\gamma^p$ is the plastic shear strain and n represents the size of the peak plateau.

$$\Delta\gamma^p = \sqrt{\Delta\epsilon^p_{ij}\Delta\epsilon^p_{ij}} \tag{205.25}$$

- `size_of_peak_plateau` (n) Is the frictional softening parameter to control the size of plateau as described in Section 104.7.3.4. The frictional softening function is an inverse tangent function raised to power n with incremental form as shown in Equation 205.24.

- `axial_viscous_damping` Is the viscous damping in axial.

- `shear_viscous_damping` Is the viscous damping in shear.

- `peak_friction_coefficient_limit` (μ_{p0}) Is the limit to the peak frictional hardening parameter μ_p.

- `peak_friction_coefficient_rate_of_decrease` (k) Is the rate of decrease of peak frictional hardening parameter μ_p with axial stress, described in Section 104.7.3.4

$$\mu_p = max(\mu_{p0}, \mu_{p0} - k * log(\sigma_a/P_0)) \tag{205.26}$$

  where μ_{p0} is the peak frictional hardening limit, k is the peak frictional parameter rate of decrease and P_0 is the reference stress of $P_0 = 101kPa$.

- `residual_friction_coefficient` (μ_r) Is the residual frictional parameter as described in Section 104.7.3.4
- `shear_zone_thickness` h Is the shear zone thickness.
- `contact_plane_vector` Vector defining the normal to the contact/interface/joint plane.
- `surface_vector_relative_tolerance` defines the relative tolerance to find all the contact/interface/joint normals and create multiple contact elements for a given contact node pairs for a conforming surface-to-surface mesh.

IMPORTANT NOTE No. 1: `contact_plane_vector` defines a direction from Node I to Node J, that is, from the first to the second node. If this normal vector is reversed, the contact/interface/joint element behaves as a hook and is likely to create convergence issues.

IMPORTANT NOTE No. 2: Two nodes that form the Contact/Interface/Joint Element, need to be placed at the same physical location, coordinates in order to prevent convergence issues when nodes are separated and element tries to close the gap in the very first step.

Description of output by this element can be found in Section 206.8.8.

205.3.4.95 Modeling, Finite Element: Stress Based Coupled Soft Contact/Interface/Joint Element with Elastic Perfectly Plastic Shear Behavior

The command is:

```
add element # <.> type StressBasedCoupledSoftContact_ElPPlShear
   with nodes (<.>, <.>)
   initial_axial_stiffness = <Pa>
   stiffening_rate = <>
   max_axial_stiffness = <Pa>
   axial_penalty_stiffness = <Pa>
   initial_shear_stiffness = <Pa>
   axial_viscous_damping = <Pa*s>
   shear_viscous_damping = <Pa*s>
   residual_friction_coefficient = <.>
   shear_zone_thickness = <m>
   contact_plane_vector = (<.>, <.>, <.> );
```

and/or;

```
add element # <.> type StressBasedCoupledSoftContact_ElPPlShear
   with nodes (<.>, <.>)
   initial_axial_stiffness = <Pa>
   stiffening_rate = <>
   max_axial_stiffness = <Pa>
   axial_penalty_stiffness = <Pa>
   initial_shear_stiffness = <Pa>
   axial_viscous_damping = <Pa*s>
   shear_viscous_damping = <Pa*s>
   residual_friction_coefficient = <.>
   shear_zone_thickness = <m>
   surface_vector_relative_tolerance = <.>;
```

The axial stress σ_a and axial stiffness E_a in defined as

$$\begin{aligned} \sigma_a &= b * exp(a * \epsilon_a) * \epsilon_a \\ E_a &= max(b * exp(a * \epsilon_a) * (1 + a * \epsilon_a), E_{max}) \end{aligned} \tag{205.27}$$

where

b refers to the initial axial stiffness in axial contact/interface/joint direction,

a refers to the stiffening rate in axial contact direction,

E_{max} refers to the maximum axial stiffness,

E_a refers to the axial stiffness,

ϵ_a refers to the axial strain in axial contact/interface/joint direction $\epsilon_a = \delta_a/h$,

δ_a is the relative axial penetration in contact axial direction,

h is the shear zone thickness, and

- `initial_axial_stiffness (b)` represents the stiffness in the axial direction (local x axis) for 1m penetration.
- `stiffening_rate (a)` Represents exponential stiffening rate $exp(sr * \epsilon_n)$ in axial direction.
- `max_axial_stiffness`(E_{max}) Defines the maximum stiffness in the axial direction (local x axis) for the contact/interface/joint element.
- `axial_penalty_stiffness` (E_p) defines the penalty stiffness between U_i degree of freedom (DoF) (saturated, coupled u-p-U element) and u_i DoF (dry u element) to enforce movement of fluid in u-p-U element with solid in u element in contact/interface/joint axial direction. This is useful for pumping action, gap opens and draws the fluid from u-p-U element and then gap closes and pumps, pushes fluid into u-p-U element.
- `initial_shear_stiffness` (E_s) Is the stiffness in the tangential (shear, local y or z axis) directions at $101 kPa$ axial stress described in Section 104.7.3.2
- `axial_viscous_damping` Is the viscous damping in axial.
- `shear_viscous_damping` Is the viscous damping in shear.
- `residual_friction_coefficient` (μ_r) Is the residual friction coefficient described in Section 104.7.3.2
- `shear_zone_thickness` h Is the shear zone thickness
- `contact_plane_vector` Vector defining the normal to the contact/interface/joint plane.
- `surface_vector_relative_tolerance` defines the relative tolerance to find all the contact/interface/joint normals and create multiple contact elements for a given contact node pairs for a conforming surface-to-surface mesh.

IMPORTANT NOTE No 1: `contact_plane_vector` defines a direction from Node I to Node J, that is, from the first to the second node. If this normal vector is reversed, the contact/interface/joint element behaves as a hook and is likely to create convergence issues.

IMPORTANT NOTE No. 2: Two nodes that form the Contact/Interface/Joint Element, need to be placed at the same physical location, coordinates in order to prevent convergence issues when nodes are separated and element tries to close the gap in the very first step.

Description of output by this element can be found in Section ??.

205.3.4.96 Modeling, Finite Element: Stress Based Coupled Soft Contact/Interface/Joint Element with Nonlinear Hardening Shear Behavior

The command is:

```
add element # <.> type StressBasedCoupledSoftContact_NonLinHardShear
   with nodes (<.>, <.>)
   initial_axial_stiffness = <Pa>
   stiffening_rate = <>
   max_axial_stiffness = <Pa>
   axial_penalty_stiffness = <Pa>
   initial_shear_stiffness = <Pa>
   axial_viscous_damping = <Pa*s>
   shear_viscous_damping = <Pa*s>
   residual_friction_coefficient = <.>
   shear_zone_thickness = <m>
   contact_plane_vector = (<.>, <.>, <.> );
```

and/or;

```
add element # <.> type StressBasedCoupledSoftContact_NonLinHardShear
   with nodes (<.>, <.>)
   initial_axial_stiffness = <Pa>
   stiffening_rate = <>
   max_axial_stiffness = <Pa>
   axial_penalty_stiffness = <Pa>
   initial_shear_stiffness = <Pa>
   axial_viscous_damping = <Pa*s>
   shear_viscous_damping = <Pa*s>
   residual_friction_coefficient = <.>
   shear_zone_thickness = <m>
   surface_vector_relative_tolerance = <.>;
```

The axial stress σ_a and axial stiffness E_a in defined as

$$\begin{aligned}\sigma_a &= b * exp(a * \epsilon_a) * \epsilon_a \\ E_a &= max(b * exp(a * \epsilon_a) * (1 + a * \epsilon_a), E_{max})\end{aligned} \tag{205.28}$$

where

b refers to the initial axial stiffness in axial contact/interface/joint direction,

a refers to the stiffening rate in axial contact direction,

E_{max} refers to the maximum axial stiffness,

E_a refers to the axial stiffness,

ϵ_a refers to the axial strain in axial contact/interface/joint direction $\epsilon_a = \delta_a/h$,

δ_a is the relative axial penetration in contact axial direction,

h is the shear zone thickness, and

- `initial_axial_stiffness (b)` represents the stiffness in the axial direction (local x axis) for 1m penetration.
- `stiffening_rate (a)` Represents exponential stiffening rate $exp(sr * \epsilon_n)$ in axial direction.
- `max_axial_stiffness`(E_{max}) Defines the maximum stiffness in the axial direction (local x axis) for the contact/interface/joint element.
- `axial_penalty_stiffness` (E_p) defines the penalty stiffness between U_i degree of freedom (DoF) (saturated, coupled u-p-U element) and u_i DoF (dry u element) to enforce movement of fluid in u-p-U element with solid in u element in contact/interface/joint axial direction. This is useful for pumping action, gap opens and draws the fluid from u-p-U element and then gap closes and pumps, pushes fluid into u-p-U element.
- `initial_shear_stiffness` (E_s) Is the stiffness in the tangential (shear, local y or z axis) directions at $101kPa$ axial stress described in Section 104.7.3.3
- `axial_viscous_damping` Is the viscous damping in axial.
- `shear_viscous_damping` Is the viscous damping in shear.
- `residual_friction_coefficient` (μ_r) Is the residual frictional parameter as described in Section 104.7.3.3
- `shear_zone_thickness` h Is the shear zone thickness
- `contact_plane_vector` Vector defining the normal to the contact/interface/joint plane.
- `surface_vector_relative_tolerance` defines the relative tolerance to find all the contact/interface/joint normals and create multiple contact elements for a given contact node pairs for a conforming surface-to-surface mesh.

IMPORTANT NOTE No. 1: `contact_plane_vector` defines a direction from Node I to Node J, that is, from the first to the second node. If this normal vector is reversed, the contact/interface/joint element behaves as a hook and is likely to create convergence issues.

IMPORTANT NOTE No. 2: Two nodes that form the Contact/Interface/Joint Element, need to be placed at the same physical location, coordinates in order to prevent convergence issues when nodes are separated and element tries to close the gap in the very first step.

Description of output by this element can be found in Section 206.8.8.

205.3.4.97 Modeling, Finite Element: Stress Based Coupled Soft Contact/Interface/Joint Element with Nonlinear Hardening and Softening Shear Behavior

The command is:

```
add element # <.> type StressBasedCoupledSoftContact_NonLinHardSoftShear
    with nodes (<.>, <.>)
    initial_axial_stiffness = <Pa>
    stiffening_rate = <>
    max_axial_stiffness = <Pa>
    axial_penalty_stiffness =<Pa>
    initial_shear_stiffness = <Pa>
    rate_of_softening = <>
    size_of_peak_plateau = <>
    axial_viscous_damping = <Pa*s>
    shear_viscous_damping = <Pa*s>
    peak_friction_coefficient_limit = <>
    peak_friction_coefficient_rate_of_decrease = <.>
    residual_friction_coefficient = <.>
    shear_zone_thickness = <m>
    contact_plane_vector = (<.>, <.>, <.> );
```

and/or;

```
add element # <.> type StressBasedCoupledSoftContact_NonLinHardSoftShear
    with nodes (<.>, <.>)
    initial_axial_stiffness = <Pa>
    stiffening_rate = <>
    max_axial_stiffness = <Pa>
    axial_penalty_stiffness =<Pa>
    initial_shear_stiffness = <Pa>
    rate_of_softening = <>
    size_of_peak_plateau = <>
    axial_viscous_damping = <Pa*s>
    shear_viscous_damping = <Pa*s>
    peak_friction_coefficient_limit = <>
    peak_friction_coefficient_rate_of_decrease = <.>
    residual_friction_coefficient = <.>
    shear_zone_thickness = <m>
    surface_vector_relative_tolerance = <.>;
```

The axial stress σ_a and axial stiffness E_a in defined as

$$\begin{aligned}\sigma_a &= b * exp(a * \epsilon_a) * \epsilon_a \\ E_a &= max(b * exp(a * \epsilon_a) * (1 + a * \epsilon_a), E_{max})\end{aligned} \tag{205.29}$$

where

b refers to the initial axial stiffness in axial contact/interface/joint direction,

a refers to the stiffening rate in axial contact direction,

E_{max} refers to the maximum axial stiffness,

E_a refers to the axial stiffness,

ϵ_a refers to the axial strain in axial contact/interface/joint direction $\epsilon_a = \delta_a/h$,

δ_a is the relative axial penetration in contact axial direction,

h is the shear zone thickness, and

- `initial_axial_stiffness` (b) represents the stiffness in the axial direction (local x axis) for 1m penetration.
- `stiffening_rate` (a) Represents exponential stiffening rate $exp(sr * \epsilon_n)$ in axial direction.
- `max_axial_stiffness`(E_{max}) Defines the maximum stiffness in the axial direction (local x axis) for the contact/interface/joint element.
- `axial_penalty_stiffness` (E_p) defines the penalty stiffness between U_i degree of freedom (DoF) (saturated, coupled u-p-U element) and u_i DoF (dry u element) to enforce movement of fluid in u-p-U element with solid in u element in contact/interface/joint axial direction. This is useful for pumping action, gap opens and draws the fluid from u-p-U element and then gap closes and pumps, pushes fluid into u-p-U element.
- `initial_shear_stiffness` (E_s) Is the stiffness in the tangential (shear, local y or z axis) directions at $101kPa$ axial stress, described in Section 104.7.3.4
- `rate_of_softening` (R_s) Is the parameter to control the rate of frictional softening described in Section 104.7.3.4. The frictional softening function is an inverse tangent function raised to power n with incremental form as

  $$\Delta\mu = -\frac{n * R_s(\mu_p - \mu_r)}{(\pi/2)^n \theta^{1/n-1}} * cos^2\theta\Delta\gamma^p \tag{205.30}$$

  $$\theta = \frac{\mu_p - \mu}{\mu_p - \mu_r}(\pi/2)^n \tag{205.31}$$

  where, R_s is the frictional softening rate parameter, $\Delta\gamma^p$ is the plastic shear strain and n represents the size of the peak plateau.

  $$\Delta\gamma^p = \sqrt{\Delta\epsilon^p_{ij}\Delta\epsilon^p_{ij}} \tag{205.32}$$

- `size_of_peak_plateau` (n) Is the frictional softening parameter to control the size of plateau as described in Section 104.7.3.4. The frictional softening function is an inverse tangent function raised to power n with incremental form as shown in Equation 205.31.

- `axial_viscous_damping` Is the viscous damping in axial.
- `shear_viscous_damping` Is the viscous damping in shear.
- `peak_friction_coefficient_limit` (μ_{p0}) Is the limit to the peak frictional hardening parameter μ_p.
- `peak_friction_coefficient_rate_of_decrease` (k) Is the rate of decrease of peak frictional hardening parameter μ_p with axial stress, described in Section 104.7.3.4

  $$\mu_p = max(\mu_{p0}, \mu_{p0} - k * log(\sigma_a/P_0)) \tag{205.33}$$

  where μ_{p0} is the peak frictional hardening limit, k is the peak frictional parameter rate of decrease and P_0 is the reference stress of $P_0 = 101kPa$.
- `residual_friction_coefficient` (μ_r) Is the residual frictional parameter as described in Section 104.7.3.4
- `shear_zone_thickness` h Is the shear zone thickness.
- `contact_plane_vector` Vector defining the normal to the contact/interface/joint plane.
- `surface_vector_relative_tolerance` defines the relative tolerance to find all the contact/interface/joint normals and create multiple contact elements for a given contact node pairs for a conforming surface-to-surface mesh.

IMPORTANT NOTE No. 1: `contact_plane_vector` defines a direction from Node I to Node J, that is, from the first to the second node. If this normal vector is reversed, the contact/interface/joint element behaves as a hook and is likely to create convergence issues.

IMPORTANT NOTE No. 2: Two nodes that form the Contact/Interface/Joint Element, need to be placed at the same physical location, coordinates in order to prevent convergence issues when nodes are separated and element tries to close the gap in the very first step.

Description of output by this element can be found in Section 206.8.8.

205.3.4.98 Modeling, Finite Element: Neoprene Isolator Finite Element

(command syntax is in development),

. . .

more on this finite element can be found in Section 102.11 on Page 136 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.99 Modeling, Finite Element: Lead Core Rubber Isolator/Dissipator Element

(command syntax is in development),

. . .

more on this finite element can be found in Section 102.11 on Page 136 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.100 Modeling, Finite Element: Frictional Pendulum Isolator/Dissipator Finite Element version01

(command syntax is in development),

. . .

more on this finite element can be found in Section 102.11 on Page 136 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.101 Modeling, Finite Element: Frictional Pendulum Isolator/Dissipator Finite Element version03

(command syntax is in development),

. . .

more on this finite element can be found in Section 102.11 on Page 136 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

205.3.4.102 Modeling, Damping: Adding Rayleigh Damping

First, define the Rayleigh damping.

```
add damping # <.> type Rayleigh with a0 = <1/T> a1 = <T> stiffness_to_use = ↩
    <Initial_Stiffness|Current_Stiffness|Last_Committed_Stiffness>;
```

then apply it to element or node.

```
add damping # <.> to element # <.>;
```

```
add damping # <.> to node # <.>;
```

NOTE:

- If the simulation model is a distributed mass system (e.g. using solid brick elements with nonzero density), users should add damping to elements. In other words, if no additional mass were added to nodes, the command add damping to nodes won't have any effect in ESSI.
- If the simulation model is a lumped mass model (e.g. using the massless beam/truss with lumped mass at nodes), users should add damping to nodes.

205.3.4.103 Modeling, Damping: Adding 3rd Order Caughey Damping

First, define the 3rd-order Caughey damping

```
add damping # <.> type Caughey3rd
with a0 = <T> a1 = <1/T> a2 = <T^3> stiffness_to_use = ↩
    <Initial_Stiffness|Current_Stiffness|Last_Committed_Stiffness>;
```

then apply it to element or node.

```
add damping # <.> to element # <.>;
```

```
add damping # <.> to node # <.>;
```

NOTE:

- If the simulation model is a distributed mass system (e.g. using solid brick elements with nonzero density), users should add damping to elements. In other words, if no additional mass were added to nodes, the command add damping to nodes won't have any effect in ESSI.
- If the simulation model is a lumped mass model (e.g. using the massless beam/truss with lumped mass at nodes), users should add damping to nodes.

205.3.4.104 Modeling, Damping: Adding 4th Caughey Damping

First, define the 4th-order Caughey damping

```
add damping # <.> type Caughey4th
with a0 = <1/T> a1 = <T> a2 = <T^3> a3 = <T^5> stiffness_to_use = ↩
    <Initial_Stiffness|Current_Stiffness|Last_Committed_Stiffness>;
```

then apply it to element or node.

```
add damping # <.> to element # <.>;
```

```
add damping # <.> to node # <.>;
```

NOTE:

- If the simulation model is a distributed mass system (e.g. using solid brick elements with nonzero density), users should add damping to elements. In other words, if no additional mass were added to nodes, the command add damping to nodes won't have any effect in ESSI.
- If the simulation model is a lumped mass system (e.g. using the massless beam/truss with lumped mass at nodes), users should add damping to nodes.

205.3.4.105 Modeling, Constraints and Supports: Adding Constraints or Supports

```
fix node # <.> dofs [ux uy uz p Ux Uy Uz rx ry rz all];
```

where at least one of the DOF fixity codes (`ux uy uz p Ux Uy Uz rx ry rz all`) has to be invoked. These codes are

- `ux`, translation in x direction, for structures and solids (solid phase only in $u-p-U$ and $u-p$ elements)
- `uy`, translation in y direction, for structures and solids (solid phase only in $u-p-U$ and $u-p$ elements)
- `uz`, translation in z direction, for structures and solids (solid phase only in $u-p-U$ and $u-p$ elements)
- `p`, pore fluid pressure (for fluid phase in $u-p-U$ and $u-p$ elements))
- `Ux`, translation of pore fluid phase in x direction (for $u-p-U$ elements)
- `Uy`, translation of pore fluid phase in y direction (for $u-p-U$ elements)
- `Uz`, translation of pore fluid phase in z direction (for $u-p-U$ elements)
- `rx`, rotation around x axes (for structural elements)
- `ry`, rotation around y axes (for structural elements)
- `rz`, rotation around z axes (for structural elements)
- `all`, all applicable DOFs for a given node

Example fix translation x and y for node #3 `fix node # 3 dofs ux uy;`

Example fix all appropriate DOFs for node #7. `fix node # 7 dofs all;`

205.3.4.106 Modeling, Constraints and Supports: Adding Stochastic Constraints or Supports

Define fixities, boundary conditions for a stochastic node. The command would fix all the polynomial chaos expanded dofs associated with the specified physical dof.

```
fix node # <.> stochastic dofs [ux uy uz all];
```

where at least one of the DOF fixity codes (`ux uy uz all`) has to be invoked. These codes are

- `ux`, translation in x direction, including all the associated polynomial chaos expanded dofs.
- `uy`, translation in y direction, including all the associated polynomial chaos expanded dofs.
- `uz`, translation in z direction, including all the associated polynomial chaos expanded dofs.
- `all`, all applicable DOFs for a given node, including all the associated polynomial chaos expanded dofs.

For example,

```
fix node # 3 stochastic dofs ux uy;
```

Fix translation dofs ux and uy, including all the associated polynomial chaos expanded dofs, for stochastic node # 3.

205.3.4.107 Modeling, Constraints and Supports: Free Constraint or Support

Free the specified DOFs on a designated node.

```
free node # <.> dofs [ux uy ux p Ux Uy Uz rx ry rz];
```

205.3.4.108 Modeling, Constraints and Supports: Add Tied/Connected Main-Fillower Nodes for the Same DOFs

Add the equal dof for tied/connected nodes for the same degree of freedom.

```
add constraint equal_dof with
master node # <.> and
slave node # <.>
dof to constrain <.>;
```

205.3.4.109 Modeling, Constraints and Supports: Adding Tied/Connected, Main-Follower Nodes for Different DOFs

Add the equal dof for tied/connected nodes for different degree of freedom.

```
add constraint equal_dof with node # <.> dof <.> master and node # <.> dof <.> ↩
   slave;
```

205.3.4.110 Modeling, Constraints and Supports: Remove Tied/Connected Main-Follower equal DOFs

Remove the tied/connected nodes equal_dofs.

```
remove constraint equal_dof node # <.>
```

205.3.4.111 Modeling, Constraints and Supports: Adding Single Point Constraint to Nodes

Define the single point constraint to nodes on a particular degree of freedom for a specified value.

```
add single point constraint to node # <.>
dof to constrain <dof_type>
constraint value of <.>
```

205.3.4.112 Modeling, Acceleration Field: Adding Acceleration/Inertia Field

```
add acceleration field # <.>
    ax = <acceleration in x direction>*[L/T^2]
    ay = <acceleration in y direction>*[L/T^2]
    az = <acceleration in z direction>*[L/T^2];
```

Example adding acceleration induced loading field for (some) elements

```
add acceleration field # 1
ax = 0*m/s^2
ay = 0*m/s^2
az = -9.81*m/s^2;
```

NOTE: see note on page 1004 for command

```
add load # <.> to element # <.> type self_weight use acceleration field # <.>;
```

205.3.4.113 Modeling, Loads: Nodal Loads

The general signature to add loads is

```
add load # <.> to node # <.>
    type <load type> <direction> = <force_amplitude>
    {more parameters};
```

The load # is a unique number assigned to each load. The node # is the number of a node which has already been defined. The load type refers to the functional form in time or pseudo-time (for static analysis) and can be any of the list

- `linear` Constant rate time dependence.
- `path` Use an arbitrary function defined in an external file.

Each force type except linear have additional parameters which will be explained later.

The force direction refers to the degree of freedom the force will be added to. These force directions are the conjugate in energy of the DOFs defined earlier. These are,

- `Fx`, force in x direction [1]
- `Fy`, force in y direction [1]
- `Fz`, force in z direction [1]
- `F_fluid_x`, force to the pore fluid phase in x direction [2]
- `F_fluid_y`, force to the pore fluid phase in y direction [2]
- `F_fluid_z`, force to the pore fluid phase in z direction [2]
- `Mx`, moment about x axes [3]
- `My`, moment about y axes [3]
- `Mz`, moment about z axes [3]

Example command for adding three linear forces ($fx = -10 * kN, fy = -10 * kN, fz = -10 * kN$) to node # 1:

---

[1] Applies to solid phase only when connected to coupled elements

[1] Applies to fluid phase when connected to coupled elements. HOWEVER, please note that these are NOT pore fluid pressures, see section 102.12.1.5 on page 148, in Lecture Notes (Jeremić et al., 1989-2025) (Lecture Notes URL).

[1] For elements with rotational DOFs, i.e. beams, shells

```
add load # 1 to node #1 type linear Fx = -10*kN;
add load # 2 to node #1 type linear Fy = -10*kN;
add load # 3 to node #1 type linear Fz = -10*kN;
```

The force `type` refers to the functional dependence in time (or pseudo-time) that the force will have. The possible functional forms have been listed before. Listed are additional parameters which define these forces.

1. `linear`

   Receives no extra parameters. In this case the magnitude of the force is interpreted as the magnitude of the force after one second of time (or pseudo-time) has passed.

2. `path`

   How to add path loads is in the next page.

205.3.4.114 Modeling, Loads: Nodal Path Loads

To add forces which follow a path other than linear, we have the `path_series`, for equally spaced time series data, and `path_time_series` for variable spaced time series data.

The commands are:

```
add load # <.> to node # <.> type path_series
  FORCETYPE = <force or moment scale factor>
  time_step = <T>
  series_file = "STRING";
```

```
add load # <.> to node # <.> type path_time_series
  FORCETYPE = <force or moment scale factor>
  series_file = "STRING";
```

As before, `FORCETYPE` can be `Fx`, `Fy`, `Fz`, `Mx`, `My`, `Mz`, `F_fluidx`, `F_fluidy`, `F_fluidz`.

The format of the series_file is one column of text for the equally spaced case (`path_series`) and double column, one for time and second one for data values, for (`path_time_series`).

205.3.4.115 Modeling, Loads: Nodal Loads From Reactions

Loads can be added from reactions.

The command is:

```
add load # <.> to node # <.> type from_reactions;
add load # <.> to all nodes type from_reactions;
```

The load # is a unique number assigned to each load. The node # is the number of a node which has already been defined. This DSL applies an external load equal to the reaction calculated at that node. It is useful, for stage loading where a constrained dof gets relaxed. The first command add load for a specified node whereas, the second command applies load for all nodes.

For example:

```
add load # 3 to node #5 type from_reactions;
```

Adds an external load to node 5 from its reaction force calculated in previous stage.

```
add load # 3 to all nodes type from_reactions;
```

Adds an external load to all nodes from their reaction force calculated in previous stage.

205.3.4.116 Modeling, Loads: Selfweight Element Load

```
add load # <.>
to element # <.>
type self_weight
use acceleration field # <.>;
```

NOTE: since the gravity acceleration field is $g = 9.81\mathrm{m/s}^2$, meaning that there is an increment of $9.81\mathrm{m/s}$ of velocity each second (please note that this defines a rate of increase in velocity), gravity is then applied in 1 second! This is sometimes (most of the time) too harsh numerically! It helps if one defines an acceleration field of say $0.0981\mathrm{m/s}^2$ and then apply it in 100 seconds. This is to be done in command `add acceleration field ...` on page 999.

205.3.4.117 Modeling, Loads: Selfweight Nodal Load

```
add load # <.> to node # <.> type self_weight use acceleration field # <.>;
```

NOTE: For this command to take effect, there should be concentrated mass defined at the node.

205.3.4.118 Modeling, Loads: 8 Node Brick Surface Load with the Constant Pressure

Surface of 8 node brick element with same pressure magnitudes at all nodes:

```
add load # <.> to element # <.> type surface at nodes (<.> , <.> , <.> , <.>) ↩
    with magnitude <Pa>;
```

Note: This command works for the dry `8NodeBrick` element and the coupled `8NodeBrick_upU` element. For the coupled upU element, this command applies all surface load on the solid phase, simulating a drained surface loading condition.

A new command for undrained surface loading on upU element will be added soon...

205.3.4.119 Modeling, Loads: 8 Node Brick Surface Load with Variable Pressure

Surface of 8 node brick element with variable pressure magnitudes at all nodes:

```
add load # <.> to element # <.> type surface at nodes (<.> , <.> , <.> , <.>) ↩
   with magnitudes ( <Pa> , <Pa> , <Pa> , <Pa>);
```

205.3.4.120 Modeling, Loads: 20Node Brick Surface Load with the Constant Pressure

Surface of 20 node brick element with same pressure magnitudes at all nodes:

```
add load # <.> to element # <.> type surface at nodes (<.> , <.> , <.> , <.>, ↩
    <.>, <.>, <.>, <.>) with magnitude <Pa>;
```

205.3.4.121 Modeling, Loads: 20 Node Brick, Surface Load with Variable Pressure

Surface of 20 node brick element with variable pressure magnitudes at all nodes:

```
add load # <.> to element # <.> type surface at nodes (<.> , <.> , <.> , <.>, ↩
    <.>, <.>, <.>, <.>) with magnitudes ( <Pa> , <Pa> , <Pa> , <Pa>, <Pa>, ↩
    <Pa>, <Pa>, <Pa>);
```

205.3.4.122 Modeling, Loads: 27 Node Brick Surface Load with the Constant Pressure

Surface of 27 node brick element with same pressure magnitudes at all nodes:

```
add load # <.> to element # <.> type surface at nodes (<.> , <.> , <.> , <.>, ↩
    <.>, <.>, <.>, <.>, <.>) with magnitude <Pa>;
```

205.3.4.123 Modeling, Loads: 27 Node Brick Surface Load with Variable Pressure

Surface of 27 node brick element with variable pressure magnitudes at all nodes:

```
add load # <.> to element # <.> type surface at nodes (<.> , <.> , <.> , <.>, ↩
    <.>, <.>, <.>, <.>, <.>) with magnitudes ( <Pa> , <Pa> , <Pa> , <Pa>, <Pa>, ↩
    <Pa>, <Pa>, <Pa>, <Pa>);
```

205.3.4.124 Modeling, Loads: Removing Loads

Loads can be removed using:

```
remove load # <.>
```

205.3.4.125 Modeling, Loads: Domain Reduction Method, DRM

```
add load # <.> type domain reduction method hdf5_file = <string>;
add load # <.> type domain reduction method hdf5_file = <string> scale_factor = ↩
    <.>;
```

- `hdf5_file` HDF5 file with information for the DRM specification. See section....
- `scale_factor` Factor to linearly scale the motion.

Creating DRM input in HDF5 format. As shown in Fig.(205.5), eight components are required for the DRM input.

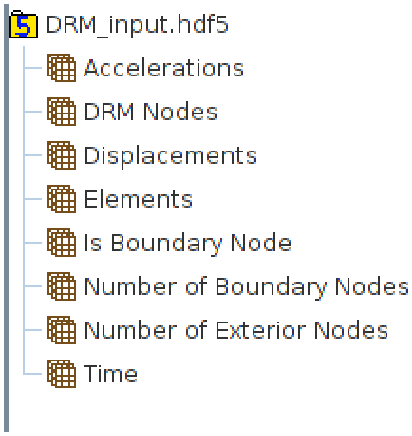


Figure 205.5: Components of DRM input in HDF5 format.

The name of the sub-folders must be exactly the same as shown here.

1. Elements: element numbers for DRM elements, a single layer of elements used to add the earthquake motion.
2. DRM Nodes: Node numbers for DRM elements.
3. Is Boundary Node: used to describe whether each of nodes in "DRM Nodes" is a boundary node or an exterior node.
    - If this value is "1", the corresponding node in "DRM Nodes" is a boundary node.
    - If this value is "0", the corresponding node in "DRM Nodes" is an exterior node in the DRM element.
4. Number of Boundary Nodes: the number of boundary nodes.

5. Number of Exterior Nodes: the number of exterior nodes.
6. Displacements: the displacement components of input earthquake motion on corresponding DRM Nodes. Displacements are a 2D array,
    - column number represents a time-step,
    - rows represents the displacement in one direction.

      If every node has 3 degrees of freedom (DOFs, say ux, uy, and uz), the first three rows represent the input displacements on first DRM node in directions of ux, uy, and uz. Then, next three rows represent the input displacements for the next node. So the total row number should be three times of the number of DRM Nodes.
7. Accelerations: same data structure as displacements, above.
8. Time: real time for each time-step in the input earthquake motion.

Python example script to generate the DRM HDF5-based input is given below. Please note that script only generates simplest possible rigid body motion, and that for any realistic motions, those will have to be created using 1C or 3C seismic motions codes, for example, SW4, SynAcc, fk, Hisada, EDT, MS ESSI, or even SHAKE for 1C motions.

```
# Created by Jose Antonio Abell Mena
# This file reads old-format DRM input files and translates them into new
# HDF5-based format.
#

# This file produces a rigid body input to the DRM layer. That is, all DRM
# nodes have same X-direction displacement and acceleration. In this case a
# sine wave is used. This is not realistic, its just for demonstration
# purposes. DRM won't work in this case but can be used to verify input if a
# pseudo-static analysis is done (zero density on all elements and apply loads
# with transient analysis.)
# For real input motions, produced using some other means, say SW4, fk,
# Hisada or MS ESSI program, this simple program can be used as an example,
# where DRM input file format used here can be (re) used, while those real
# motions are read form output of above mentioned programs.

import scipy as sp
import h5py
import time

# Write elements and nodes data
elements = sp.loadtxt("DRMelements.txt",dtype=sp.int32)
exterior_nodes = sp.loadtxt("DRMexterior.txt",dtype=sp.int32)
boundary_nodes = sp.loadtxt("DRMbound.txt",dtype=sp.int32)
```

```

Ne = sp.array(exterior_nodes.size)
Nb = sp.array(boundary_nodes.size)

Nt = Ne+Nb

all_nodes = sp.hstack((boundary_nodes, exterior_nodes))
is_boundary_node = sp.zeros(Nt, dtype=sp.int32)
is_boundary_node[0:Nb] = 1

h5file = h5py.File("small.h5.drminput","w")

h5file.create_dataset("Elements", data=elements)
h5file.create_dataset("DRM Nodes", data=all_nodes)

# This array has 1 if the node at the corresponding position in "DRM nodes"
# array is a boundary node and zero if not
h5file.create_dataset("Is Boundary Node", data=is_boundary_node)

h5file.create_dataset("Number of Exterior Nodes", data=Ne)
h5file.create_dataset("Number of Boundary Nodes", data=Nb)

# Write timestamp (time format used is that of c "asctime" Www Mmm dd
# hh:mm:ss yyyy example: Tue Jan 13 10:17:09 2009)
localtime = time.asctime( time.localtime(time.time()) )
h5file.create_dataset("Created",data=str(localtime))

# Generate motions
t = sp.linspace(0,10,1001)
w = 2*sp.pi/0.5
d = sp.sin(w*t)
a = -w**2*sp.sin(w*t)

# Output accelerations, displacements and time-vector

# Format is:
#
# Array/matrix for Accelerations and Displacements has the following shape
# [3*(N_boundary_nodes + N_exterior_nodes) , Ntimesteps]
#
# where component
# A[3*n], A[3*n+1], A[3*n+2]
# correspond to accelerations/displacements in X, Y, and Z directions
# at node n.
# The location corresponding to node n is that of the n-th component of array
# "DRM Nodes"

# Time vector

h5file.create_dataset("Time", data=t)

```

```
acc = h5file.create_dataset("Accelerations", (3*Nt,len(t)), dtype=sp.double)
dis = h5file.create_dataset("Displacements", (3*Nt,len(t)), dtype=sp.double)

for node_index in range(Nt):
 acc[3*node_index,:] = a
 acc[3*node_index+1,:] = 0*a #Zero acceleration in y and z
 acc[3*node_index+2,:] = 0*a
 dis[3*node_index,:] = d
 dis[3*node_index+1,:] = 0*d #Zero displacement in y and z
 dis[3*node_index+2,:] = 0*d

h5file.close()
```

205.3.4.126 Modeling, Wave Field for Creating DRM Loads: Add Wave Field

```
add wave field # <.> with
    acceleration_filename = <string>
    unit_of_acceleration = <L/T^2>
    displacement_filename = <string>
    unit_of_displacement = <L>
    add_compensation_time = <T>
    motion_depth = <L>
    monitoring_location = <within_soil_layer|equivalent_rock_outcropping>
    soil_profile_filename = <string>
    unit_of_Vs = <L/T>
    unit_of_rho = <M/L^3>
    unit_of_damping = <absolute|percent>
    unit_of_thickness = <L>
    ;
```

Example adding a wave field

```
add wave field # 1 with
    acceleration_filename = "acc.txt"
    unit_of_acceleration = 1 * m/s^2
    displacement_filename = "dis.txt"
    unit_of_displacement = 1 * m
    add_compensation_time = 0.5 * s
    motion_depth = 0 * m
    monitoring_location = within_soil_layer
    soil_profile_filename = "soil_profile.txt"
    unit_of_Vs = 1 * m/s
    unit_of_rho = 1 * kg/m^3
    unit_of_damping = absolute
    unit_of_thickness = 1*m
    ;
```

where:

- `No (or #)<.>` is the unique wave field ID. The wave field ID does not have to be sequential, any unique positive integer number can be used. Each wave field is a 1C wave field. The wave field does not have a direction. Later, if users want to add load with the wave field, users should specify the direction with each wave field.
- `acceleration_filename` is the filename of a plain text file, which contains the acceleration of the input motion. The file should have two columns, where the first column is the accumulated time and the second column is the corresponding acceleration. For the DRM loading from a wave field, if the simulation time is longer than the earthquake motion, the remaining simulation will continue with zero motions. The wave field does NOT conduct any base correction on the input

motion, so the simulation results may have permanent deformation after the earthquake. If users want to have base corrections, users should pre-process the earthquake motion by themselves.

- `displacement_filename` is the filename of a plain text file, which contains the displacement of the input motion. The file should have two columns, where the first column is the accumulated time and the second column is the corresponding displacement. For the DRM loading from a wave field, if the simulation time is longer than the earthquake motion, the remaining simulation will continue with zero motions. The wave field does NOT conduct any base correction on the input motion, so the simulation results may have permanent deformation after the earthquake. If users want to have base corrections, users should pre-process the earthquake motion by themselves.

- `add_compensation_time` is a feature to add zero-motion in the beginning and at the end of the earthquake motion. Since the wave propagation theory is solving the wave equation in frequency domain (steady state), without additional zeros, the beginning and the end of wave may be mixed up. If the user does not want to add the additional zeros, user can specify 0*s.

- `motion_depth` is the depth of the input motion. Usually, the `motion_depth` is at the surface (namely, 0*m). Later, users can specify the request depth for deconvolution. However, users can also specify a specific depth of the input motion. In this case, users can request both convolution and deconvolution.

  - If the request depth is deeper than this input acceleration depth, the wave propagation will generate the deconvolution results.
  - If the request depth is shallower than this input acceleration depth, the wave propagation will generate the convolution results. It is recommended to add a small damping for wave convolution.

  The acceleration depth is the relative depth to the soil surface, so both negative and positive depth are acceptable and result in the same results.

- `monitoring_location` is the location of the earthquake monitoring station. When the monitoring location is within soil layer, the wave propagation is conducted inside the soil layer directly. When the monitoring location is equivalent rock outcropping, wave deconvolution is conducted back to the bedrock first and then propagate into the soil layers.

- `soil_profile_filename` contains the soil properties for each layer. The soil profile file should have four columns, which are the shear wave velocity, density, damping ratio and thickness of each layer respectively. The soil layers should be from the soil surface to the bedrock. The last layer is bedrock, which has three columns only. User should NOT give the thickness of the last layer.

One Example of soil profile file is given below.

```
// Vs rho damp thickness
200 2000 0.03 35
250 2000 0.04 35
2000 2400 0.05
```

205.3.4.127 Modeling, Wave Field for Creating DRM Loads: Deconvolution

This command performs deconvolution or convolution of given motions at surface or certain depth and writes accelerations, velocities and displacements in 3 directions, in files, that can then be used to create DRM loads.

```
generate wave propagation results of wave field
  # <.> at depth <L> to file <output_filename_prefix>
```

One example of deconvolution is

```
generate wave propagation results of wave field
  # 1 at depth -60*m to file "Northridge_record" ;
```

where

- `wave field # <.>` specifies the wave field number which will be used for wave propagation.
- `depth <L>` is the request depth of the output motion.
  - If the request depth is deeper than the input motion that defined in the wave field, this command will generate the deconvolution results.
  - If the request depth is shallower than the input motion that defined in the wave field, this command will generate the convolution results. It is recommended to add a small damping for wave convolution.

  The depth specifies the *relative* location between the soil surface and the request depth, which means both positive and negative depth are acceptable and will result in the same results.
- `output_filename_prefix` specified the prefix of the output filenames. This command will generate 3 output files, whose suffix are `at_str(depth)_acc.txt` , `at_str(depth)_vel.txt` , `at_str(depth)_dis.txt` .

205.3.4.128 Modeling, Wave Field for Creating DRM Input: Motions

This command performs deconvolution or convolution of given motions at surface or certain depth and directly generates DRM motions in 1, 2, or 3 directions.

```
generate DRM motion file from wave field
  # <.> in direction <ux|uy|uz>
  soil_surface at z = <L>
  hdf5_file = <string> ;
```

```
generate DRM motion file from wave field
  # <.> in direction <ux|uy|uz>
  # <.> in direction <ux|uy|uz>
  soil_surface at z = <L>
  hdf5_file = <string> ;
```

```
generate DRM motion file from wave field
  # <.> in direction <ux|uy|uz>
  # <.> in direction <ux|uy|uz>
  # <.> in direction <ux|uy|uz>
  soil_surface at z = <L>
  hdf5_file = <string> ;
```

One example of deconvolution to DRM is

```
generate DRM motion file from wave field
  # 1 in direction ux
  soil_surface at z = 0*m
  hdf5_file = "input.hdf5" ;
```

where:

- `in direction <ux,uy,uz>` specifies the direction of the wave field. Each wave field is a 1C wave field. At most 3 wave fields can be associated with the load.
- `soil_surface` specifies the relation between the FEM coordinate systems and the soil profile depths inside the wave field. The soil surface should always be above the DRM nodes. Namely, soil surface is generally the surface between the soil and the structure, NOT the bedrock surface.
- `hdf5_file` specifies the HDF5 file which contain the information about the DRM elements and DRM nodes.

205.3.4.129 Modeling, Wave Field for Creating DRM Input: Forces

This command performs deconvolution or convolution of given motions at surface or certain depth and directly generates DRM motions and forces in 1, 2, or 3 directions.

```
generate DRM force file from wave field
  # <.> in direction <ux|uy|uz>
  soil_surface at z = <L>
  hdf5_file = <string> ;
```

```
generate DRM force file from wave field
  # <.> in direction <ux|uy|uz>
  # <.> in direction <ux|uy|uz>
  soil_surface at z = <L>
  hdf5_file = <string> ;
```

```
generate DRM force file from wave field
  # <.> in direction <ux|uy|uz>
  # <.> in direction <ux|uy|uz>
  # <.> in direction <ux|uy|uz>
  soil_surface at z = <L>
  hdf5_file = <string> ;
```

One example of deconvolution to DRM is

```
generate DRM force file from wave field
  # 1 in direction ux
  soil_surface at z = 0*m
  hdf5_file = "input.hdf5" ;
```

where:

- `in direction <ux,uy,uz>` specifies the direction of the wave field. Each wave field is a 1C wave field. At most 3 wave fields can be associated with the load.
- `soil_surface` specifies the relation between the FEM coordinate systems and the soil profile depths inside the wave field. The soil surface should always be above the DRM nodes. Namely, soil surface is generally the surface between the soil and the structure, NOT the bedrock surface.
- `hdf5_file` specifies the HDF5 file which contain the information about the DRM elements and DRM nodes.

205.3.4.130 Modeling, Wave Field for Creating DRM Loads: Add Inclined Plane Wave Field from Incident SV Wave Potential Magnitude

```
add wave field # <.> type inclined_plane_wave with
   anticlockwise_angle_of_SV_wave_plane_from <x|y|z> = <degrees>
   SV_incident_magnitude = <L^2>
   SV_incident_angle = <degrees>
   SV_incident_frequency = <1/T>
   motion_time_step = <T>
   number_of_time_steps = <.>
   soil_profile_filename = <string>
   soil_surface at <x|y|z> = <L>
   unit_of_vs_and_vp = <L/T>
   unit_of_rho = <M/L^3>
   unit_of_damping = <absolute|percent>
   unit_of_thickness = <L>
   ;
```

Example of adding an inclined plane wave field

```
add wave field # 1 type inclined_plane_wave with
  anticlockwise_angle_of_SV_wave_plane_from x= 30
  SV_incident_magnitude = 2*m^2
  SV_incident_angle = 60
  SV_incident_frequency = 5/s
  motion_time_step = 0.01*s
  number_of_time_steps = 600
  soil_profile_filename = "soil.txt"
  soil_surface at z = 0*m
  unit_of_vs_and_vp = 1*m/s
  unit_of_rho = 1*kg/m^3
  unit_of_damping = absolute
  unit_of_thickness = 1*m;
```

where:

- `No (or #)<.>` is the unique wave field ID/number. The wave field ID does not have to be sequential, any unique positive integer number can be used. Each wave field is an inclined plane SV wave field.
- `anticlockwise_angle_of_SV_wave_plane_from <x|y|z>` specifies the orientation of the inclined wave field propagation plane. User should give the anticlockwise angle in degrees between the wave propagation plane and the specified reference axis. The reference axis could be x or y or z. As shown in figure 205.7, the `anticlockwise_angle_of_SV_wave_plane_from` x axis is α.
- `SV_incident_magnitude` specifies the incident SV wave potential magnitude. The displacement magnitude of incident SV wave is related to the potential magnitude as follows: $|u| = \phi\omega/V_s$, where

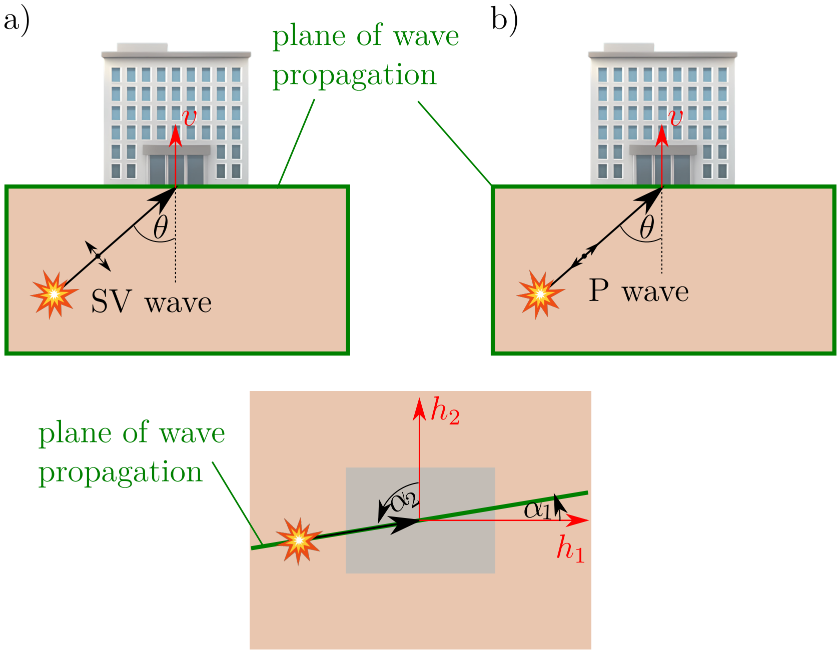


Figure 205.6: Orientation of an inclined plane wave with respect to the vertical, v, and horizontal, h_1, h_2, directions: a) a SV wave and b) a P-wave. The user defines v, h_1, h_2 directions. Either α_1 or α_2 can be input into the DSL command.

$|u|$ is the displacement magnitude, ϕ is potential magnitude, ω is incident angular frequency and V_s is the shear wave velocity of the incident soil/rock layer.

- `SV_incident_angle` specifies the inclination angle of incident SV wave, measured from vertical axis of wave plane to the wave propagation axis. In figure 205.7, the incident angle of SV wave is θ.
- `motion_time_step` is the time step/interval used for discretizing the harmonic motion into time domain.
- `number_of_time_steps` is the number of total time steps for the discretized harmonic motion.
- `soil_profile_filename` is a file name for a file that contains the soil properties for each layer. The soil profile file should have fives columns: (i) shear wave velocity, (ii) compressional wave velocity, (iii) density, (iv) damping ratio and (v) thickness of each layer respectively. The soil layers count from the soil surface to the bedrock. The last layer is bedrock, which has four columns only. User should NOT give the thickness of the last layer, as it extends into halfspace.

  One Example of soil profile file is given below.

```
// Vs Vp rho damp thickness
  200 333 2000 0.02 100
  250 408 2000 0.02 200
 2000 3400 2400 0.02
```

- `soil_surface at <x|y|z>` defines the location of soil surface in the global coordinate system of Real-ESSI.

205.3.4.131 Modeling, Wave Field for Creating DRM Loads: Add Inclined Plane Wave Field from Incident SV Wave Time Series Signal

```
add wave field # <.> type inclined_plane_wave with
   anticlockwise_angle_of_SV_wave_plane_from <x|y|z> = <degrees>
   SV_incident_acceleration_filename = <string>
   unit_of_acceleration = <L/T^2>
   SV_incident_displacement_filename = <string>
   unit_of_displacement = <L>
   SV_incident_angle = <degrees>
   add_compensation_time = <T>
   source_location = (<T>, <T>, <T>)
   soil_profile_filename = <string>
   soil_surface at <x|y|z> = <L>
   unit_of_vs_and_vp = <L/T>
   unit_of_rho = <M/L^3>
   unit_of_damping = <absolute|percent>
   unit_of_thickness = <L>
   ;
```

Example of adding an inclined plane wave field

```
add wave field # 1 type inclined_plane_wave with
  anticlockwise_angle_of_SV_wave_plane_from x = 0
  SV_incident_acceleration_filename = "Kobe_acc.txt"
  unit_of_acceleration = 1*m/s^2
  SV_incident_displacement_filename = "Kobe_disp.txt"
  unit_of_displacement = 1*m
  SV_incident_angle = 15
  add_compensation_time = 0.5*s
  source_location = (-150*m, 0*m, -100*m)
  soil_profile_filename = "soil_profile.txt"
  soil_surface at z = 0*m
  unit_of_vs_and_vp = 1*m/s
  unit_of_rho = 1*kg/m^3
  unit_of_damping = absolute
  unit_of_thickness = 1*m;
```

where:

- `No (or #)<.>` is the unique wave field ID/number. The wave field ID does not have to be sequential, any unique positive integer number can be used. Each wave field is an inclined plane SV wave field.
- `anticlockwise_angle_of_SV_wave_plane_from <x|y|z>` specifies the orientation of the inclined wave field propagation plane. User should give the anticlockwise angle in degrees between the wave propagation plane and the specified reference axis. The reference axis could be x or y or

z. As shown in figure 205.7, the `anticlockwise_angle_of_SV_wave_plane_from` x axis is α.

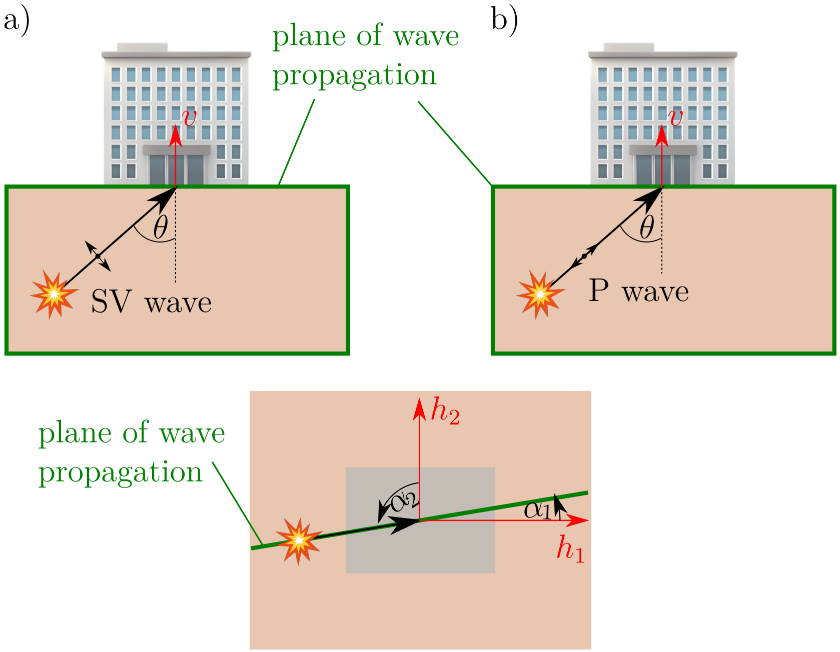


Figure 205.7: Orientation of an inclined plane wave with respect to the vertical, v, and horizontal, h_1, h_2, directions: a) a SV wave and b) a P-wave. The user defines v, h_1, h_2 directions. Either α_1 or α_2 can be input into the DSL command.

- `SV_incident_acceleration_filename` is the filename of a plain text file, which contains the acceleration of the input motion. The file should have two columns, where the first column is the accumulated time and the second column is the corresponding acceleration. For the DRM loading from a wave field, if the simulation time is longer than the earthquake motion, the remaining simulation will continue with zero motions. The wave field implementation in Real-ESSI does NOT conduct any baseline correction on the input motion. The users should perform baseline correction for earthquake motions by themselves.

- `SV_incident_displacement_filename` is the filename of a plain text file, which contains the displacement of the input motion. The file should have two columns, where the first column is the accumulated time and the second column is the corresponding displacement. For the DRM loading from a wave field, if the simulation time is longer than the earthquake motion, the remaining simulation will continue with zero motions. The wave field implementation in Real-ESSI does NOT conduct any baseline correction on the input motion. The users should perform baseline correction for earthquake motions by themselves.

- `SV_incident_angle` specifies the inclination angle of incident SV wave, measured from vertical axis of wave plane to the wave propagation axis. In figure 205.7, the incident angle of SV wave is θ.

- `add_compensation_time` is a feature to add zero-motion in the beginning and at the end of the earthquake motion. If the user does not want to add the additional zeros, user can specify compensation time as 0*s.

- `source_location` specify the location of seismic source, where the seismic motion is input as in `SV_incident_acceleration_filename` and `SV_incident_displacement_filename`, it is used for determining phase of the wave, and the source location can be inside or outside of the model.

- `soil_profile_filename` is a file name for a file that contains the soil properties for each layer. The soil profile file should have fives columns: (i) shear wave velocity, (ii) compressional wave velocity, (iii) density, (iv) damping ratio and (v) thickness of each layer respectively. The soil layers count from the soil surface to the bedrock. The last layer is bedrock, which has four columns only. User should NOT give the thickness of the last layer, as it extends into half-space.

  One Example of soil profile file is given below.

```
 // Vs Vp rho damp thickness
   200 333 2000 0.02 100
   250 408 2000 0.02 200
  2000 3400 2400 0.02
```

- `soil_surface at <x|y|z>` defines the location of soil surface in the global coordinate system of Real-ESSI.

205.3.4.132 Modeling, Wave Field for Creating DRM Loads: Add Inclined Plane Wave Field from Incident P Wave Potential Magnitude

```
add wave field # <.> type inclined_plane_wave with
    anticlockwise_angle_of_P_wave_plane_from <x|y|z> = <degrees>
    P_incident_magnitude = <L^2>
    P_incident_angle = <degrees>
    P_incident_frequency = <1/T>
    motion_time_step = <T>
    number_of_time_steps = <.>
    soil_profile_filename = <string>
    soil_surface at <x|y|z> = <L>
    unit_of_vs_and_vp = <L/T>
    unit_of_rho = <M/L^3>
    unit_of_damping = <absolute|percent>
    unit_of_thickness = <L>
    ;
```

Example adding an inclined plane wave field

```
add wave field # 1 type inclined_plane_wave with
  anticlockwise_angle_of_P_wave_plane_from x= 30
  P_incident_magnitude = 2*m^2
  P_incident_angle = 60
  P_incident_frequency = 5/s
  motion_time_step = 0.01*s
  number_of_time_steps = 600
  soil_profile_filename = "soil.txt"
  soil_surface at z = 0*m
  unit_of_vs_and_vp = 1*m/s
  unit_of_rho = 1*kg/m^3
  unit_of_damping = absolute
  unit_of_thickness = 1*m;
```

where:

- `No (or #)<.>` is the unique wave field ID/number. The wave field ID does not have to be sequential, any unique positive integer number can be used. Each wave field is an inclined plane P wave field.

- `anticlockwise_angle_of_P_wave_plane_from <x|y|z>` specifies the orientation of the inclined wave propagation plane. User should give the anticlockwise angle in degrees between the wave propagation plane and the specified reference axis. The reference axis could be x or y or z. As shown in figure 205.7, the `anticlockwise_angle_of_P_wave_plane_from` x axis is α.

- `P_incident_magnitude` specifies the incident P wave potential magnitude. The displacement magnitude of incident P wave is related to the potential magnitude as following: $|u| = \phi\omega/V_p$, where $|u|$ is the displacement magnitude, ϕ is potential magnitude, ω is incident angular frequency and V_p is the compressional wave velocity of the incident layer.

- `P_incident_angle` specifies the inclination of incident P wave. The angle is measured from vertical axis of wave plane to the wave propagation axis. In figure 205.7, the incident angle of P wave is θ.
- `motion_time_step` is the time interval when discretized the harmonic motion into time domain.
- `number_of_time_steps` is the number of total time steps for the discretized harmonic motion.
- `soil_profile_filename` is a file name for a file that contains the soil properties for each layer. The soil profile file should have fives columns: (i) shear wave velocity, (ii) compressional wave velocity, (iii) density, (iv) damping ratio and (v) thickness of each layer respectively. The soil layers count from the soil surface to the bedrock. The last layer is bedrock, which has four columns only. User should NOT give the thickness of the last layer, as it extends into halfspace.

  One Example of soil profile file is given below.

```
// Vs Vp rho damp thickness
  200 333 2000 0.02 100
  250 408 2000 0.02 200
 2000 3400 2400 0.02
```

- `soil_surface at <x|y|z>` defines the location of soil surface in the global coordinate system of Real-ESSI.

205.3.4.133 Modeling, Wave Field for Creating DRM Loads: Add Inclined Plane Wave Field from Incident P Wave Time Series Signal

```
add wave field # <.> type inclined_plane_wave with
   anticlockwise_angle_of_P_wave_plane_from <x|y|z> = <degrees>
   P_incident_acceleration_filename = <string>
   unit_of_acceleration = <L/T^2>
   P_incident_displacement_filename = <string>
   unit_of_displacement = <L>
   P_incident_angle = <degrees>
   add_compensation_time = <T>
   source_location = (<T>, <T>, <T>)
   soil_profile_filename = <string>
   soil_surface at <x|y|z> = <L>
   unit_of_vs_and_vp = <L/T>
   unit_of_rho = <M/L^3>
   unit_of_damping = <absolute|percent>
   unit_of_thickness = <L>
   ;
```

Example of adding an inclined plane wave field

```
add wave field # 1 type inclined_plane_wave with
  anticlockwise_angle_of_P_wave_plane_from x = 0
  P_incident_acceleration_filename = "Kobe_acc.txt"
  unit_of_acceleration = 1*m/s^2
  P_incident_displacement_filename = "Kobe_disp.txt"
  unit_of_displacement = 1*m
  P_incident_angle = 15
  add_compensation_time = 0.5*s
  source_location = (-150*m, 0*m, -100*m)
  soil_profile_filename = "soil_profile.txt"
  soil_surface at z = 0*m
  unit_of_vs_and_vp = 1*m/s
  unit_of_rho = 1*kg/m^3
  unit_of_damping = absolute
  unit_of_thickness = 1*m;
```

where:

- `No (or #)<.>` is the unique wave field ID/number. The wave field ID does not have to be sequential, any unique positive integer number can be used. Each wave field is an inclined plane P wave field.
- `anticlockwise_angle_of_P_wave_plane_from <x|y|z>` specifies the orientation of the inclined wave field propagation plane. User should give the anticlockwise angle in degrees between the wave propagation plane and the specified reference axis. The reference axis could be x or y or

z. As shown in figure 205.7, the `anticlockwise_angle_of_P_wave_plane_from` x axis is α.

- `P_incident_acceleration_filename` is the filename of a plain text file, which contains the acceleration of the input motion. The file should have two columns, where the first column is the accumulated time and the second column is the corresponding acceleration. For the DRM loading from a wave field, if the simulation time is longer than the earthquake motion, the remaining simulation will continue with zero motions. The wave field implementation in Real-ESSI does NOT conduct any baseline correction on the input motion. The users should perform baseline correction for earthquake motions by themselves.

- `P_incident_displacement_filename` is the filename of a plain text file, which contains the displacement of the input motion. The file should have two columns, where the first column is the accumulated time and the second column is the corresponding displacement. For the DRM loading from a wave field, if the simulation time is longer than the earthquake motion, the remaining simulation will continue with zero motions. The wave field implementation in Real-ESSI does NOT conduct any baseline correction on the input motion. The users should perform baseline correction for earthquake motions by themselves.

- `P_incident_angle` specifies the inclination angle of incident P wave, measured from vertical axis of wave plane to the wave propagation axis. In figure 205.7, the incident angle of P wave is θ.

- `add_compensation_time` is a feature to add zero-motion in the beginning and at the end of the earthquake motion. If the user does not want to add the additional zeros, user can specify compensation time as 0*s.

- `source_location` specify the location of seismic source, where the seismic motion is input as in `P_incident_acceleration_filename` and `P_incident_displacement_filename`, it is used for determining phase of the wave, and the source location can be inside or outside of the model.

- `soil_profile_filename` is a file name for a file that contains the soil properties for each layer. The soil profile file should have fives columns: (i) shear wave velocity, (ii) compressional wave velocity, (iii) density, (iv) damping ratio and (v) thickness of each layer respectively. The soil layers count from the soil surface to the bedrock. The last layer is bedrock, which has four columns only. User should NOT give the thickness of the last layer, as it extends into half-space.

  One Example of soil profile file is given below.

```
// Vs Vp rho damp thickness
  200 333 2000 0.02 100
  250 408 2000 0.02 200
```

```
  2000 3400 2400 0.02
```

- `soil_surface at <x|y|z>` defines the location of soil surface in the global coordinate system of Real-ESSI.

205.3.4.134 Modeling, Wave Field for Creating DRM Loads: DRM Inclined Motion

This command generates inclined DRM motion with pre-defined inclined wave field.

```
generate DRM motion file from wave field # <.> hdf5_file = <string>;
```

One example of generating inclined DRM motion is:

```
generate DRM motion file from wave field # 1 hdf5_file = "DRMinput.hdf5";
```

where

- `wave field # <.>` specifies the inclined plane wave field number which will be used for wave propagation.
- `hdf5_file` specifies the HDF5 file which contains the geometric information about the DRM elements and DRM nodes.

205.3.4.135 Modeling, Imposed Motions: through Loads, Motion Time History, Constant Time Step

Impose motions (displacements, velocities and accelerations) through loads. This one is used if time increment is constant during the analysis. Input files have one column only, corresponding file for displacements, velocities, and accelerations.

```
add load # <.> type imposed motion to node # <.> dof DOFTYPE
time_step = <T>
displacement_scale_unit = <L>
displacement_file = "filename"
velocity_scale_unit = <L/T>
velocity_file = "filename"
acceleration_scale_unit = <L/L^2>
acceleration_file = "filename";
```

The above command generates load to the corresponding node to get the applied imposed motion.

205.3.4.136 Modeling, Imposed Motions: through Loads, Stochastic Motion Time History, Constant Time Step

Impose stochastic motions (uncertain displacements, velocities and accelerations) through stochastic loads. This one is used if time increment is constant during the analysis.

```
add load # <.> type imposed random motions to node # <.> dof DOFTYPE
time_step = <T>
displacement_scale_unit = <L>
displacement_file = "filename"
velocity_scale_unit = <L/T>
velocity_file = "filename"
acceleration_scale_unit = <L/L^2>
acceleration_file = "filename"
penalty_stiffness = <N/L>
using double product # <.>;
```

where

- `DOFTYPE` specify the dof to impose the uncertain motion. It can be either `ux`, `uy` or `uz`.
- `time_step` specify the time step of the imposed uncertain motion.
- `displacement_scale_unit` specify the scale unit of the imposed uncertain displacement polynomial chaos (PC) coefficients.
- `displacement_file` specify the filename of a text file containing PC coefficients of uncertain displacement random process. The number of rows of the file content should be equal to total number of polynomial chaos basis of the displacement random process. The number of columns of the file content should be equal to total number of time steps of the displacement random process. The value at the i^{th} row and j^{th} column of the file gives the PC coefficient of the i^{th} PC basis of the displacement random process at the j^{th} time step.
- `velocity_scale_unit` specify the scale unit of the imposed uncertain velocity polynomial chaos (PC) coefficients.
- `velocity_file` specify the filename of a text file containing PC coefficients of uncertain velocity random process. The number of rows of the file content should be equal to total number of polynomial chaos basis of the velocity random process. The number of columns of the file content should be equal to total number of time steps of the velocity random process. The value at the i^{th} row and j^{th} column of the file gives the PC coefficient of the i^{th} PC basis of the velocity random process at the j^{th} time step.

- `acceleration_scale_unit` specify the scale unit of the imposed uncertain acceleration polynomial chaos (PC) coefficients.

- `acceleration_file` specify the filename of a text file containing PC coefficients of uncertain acceleration random process. The number of rows of the file content should be equal to total number of polynomial chaos basis of the acceleration random process. The number of columns of the file content should be equal to total number of time steps of the acceleration random process. The value at the i^{th} row and j^{th} column of the file gives the PC coefficient of the i^{th} PC basis of the acceleration random process at the j^{th} time step.

- `penalty_stiffness` specify the penalty stiffness for the input of uncertain motion using penalty method. The penalty stiffness is expected to be several magnitudes, e.g., $10^3 \sim 10^6$, larger than the elemental stiffness.

- `double product #` specify the ID of the double product that would be used in the formation of stochastic force. In stochastic finite element method (FEM), the first PC basis for this double product should come from the PC representation of uncertain FEM system response, e.g., uncertain structural displacement. The second PC basis for this double product should come from the uncertain imposed motion representation.

205.3.4.137 Modeling, Imposed Motions: through Loads, Stochastic Random Process Motions, Constant Time Step

Impose a defined random process motions.

This one is used if time increment is constant during the analysis.

```
add load # <.> type imposed random motions to node # <.> dof DOFTYPE
time_step = <T>
uncertain_displacement = random field # <.>
displacement_scale_unit = <L>
penalty_stiffness = <N/L>
using double product # <.>;
```

where

- `DOFTYPE` specify the dof to impose the uncertain motion. It can be either `ux`, `uy` or `uz`.
- `time_step` specify the time step of the imposed uncertain motion.
- `uncertain_displacement` specify the imposed uncertain motion through a defined random field-/process.
- `displacement_scale_unit` specify the scale unit of the imposed uncertain displacement.
- `penalty_stiffness` specify the penalty stiffness for the input of uncertain motion using penalty method. The penalty stiffness is expected to be several magnitudes, e.g., $10^3 \sim 10^6$, larger than the elemental stiffness.
- `double product #` specify the ID of the double product that would be used in the formation of stochastic force. In stochastic finite element method (FEM), the first PC basis for this double product should come from the PC representation of uncertain FEM system response, e.g., uncertain structural displacement. The second PC basis for this double product should come from the uncertain imposed motion representation.

205.3.4.138 Modeling, Imposed Motions: through Loads, Motion Time History, Variable Time Step

Impose motions (displacements, velocities and accelerations) through loads. This one is used if time increment is variable during the analysis. Input files have two columns, first column is time and the second column in corresponding file for displacements, velocities, and accelerations. Time steps have to be the same in each file.

```
add load # <.> type imposed motion to node # <.> dof DOFTYPE
displacement_scale_unit = <displacement>
displacement_file = "filename"
velocity_scale_unit = <velocity>
velocity_file = "filename"
acceleration_scale_unit = <acceleration>
acceleration_file = "filename";
```

The above command generates load to the corresponding node to get the applied imposed motion.

205.3.4.139 Modeling, Imposed Motions: Adding Load for Uniform Acceleration Time History

Defines a non-inertial reference frame from which all displacements are measured. This reference frame (fixed to the base of the model) accelerates according to a given acceleration record. All output quantities are derived from this relative coordinate system (not-inertial). To get total displacements, the twice-integrated acceleration record must be added to the results.

The command is:

```
add load # <.> type uniform acceleration to all nodes dof <.>
  time_step = <T>
  scale_factor = <L/T^2>
  initial_velocity = <L/T>
  acceleration_file = <string>;
```

Where

- `time_step` Is the time step of the record in time units.
- `scale_factor` Is a dimensionless factor with which the record is scaled before it's applied.
- `initial_velocity` Initial velocity for all translational DOFs of the system.
- `acceleration_file` String containing the path (relative or absolute) to the record text file.

File format is a single value of the record in acceleration units (m/s/s) per line for each time step. If a time-step different from the record is used for analysis, then the record is interpolated linearly.

205.3.4.140 Modeling, Imposed Motions: Remove Imposed Motions

Motions can be removed using:

```
remove imposed motion # <.>
```

205.3.4.141 Modeling, Random Variable: Adding Gaussian Random Variables

Gaussian random variable can be added for probabilistic analysis.

The command is:

```
add random variable # <.> with Gaussian distribution mean = <.> ↩
    standard_deviation = <.>;
```

where:

- `mean` is the mean of the Gaussian random variable
- `standard_deviation` is the standard deviation of the Gaussian random variable

For example:

```
add random variable # 1 with Gaussian distribution mean = 3.0 ↩
    standard_deviation = 1.0;
```

Adds a Gaussian random variable 1 with mean equal to 3.0 and standard deviation equal to 1.0.

205.3.4.142 Modeling, Random Variable: Adding Gaussian Random Variables with Location

Gaussian random variable with spatial location can be added for probabilistic analysis. The location information of the defined random variable is important to calculate the correlation structure of random field, that can consist of many random variables.

The command is:

```
add random variable # <.> with Gaussian distribution mean = <.> ↩
   standard_deviation = <.> at (<L>, <L>, <L>);
```

where:

- `mean` is the mean of the Gaussian random variable
- `standard_deviation` is the standard deviation of the Gaussian random variable

For example:

```
add random variable # 1 with Gaussian distribution mean = 3.0 ↩
   standard_deviation = 1.0 at (3*m, 0*m, 0*m);
```

Adds a Gaussian random variable 1 with mean equal to 3.0 and standard deviation equal to 1.0 at location $x = 3$m, $y = 0$m and $z = 0$m.

205.3.4.143 Modeling, Random Variable: Adding Lognormal Random Variables

Lognormal random variable can be added for probabilistic analysis.

The command is:

```
add random variable # <.> with Lognormal distribution mean = <.> ↩
    standard_deviation = <.>;
```

where:

- `mean` is the mean of the lognormal random variable
- `standard_deviation` is the standard deviation of the lognormal random variable

For example:

```
add random variable # 1 with Lognormal distribution mean = 3.0 ↩
    standard_deviation = 1.0;
```

Adds a Lognormal random variable 1 with mean equal to 3.0 and standard deviation equal to 1.0.

205.3.4.144 Modeling, Random Variable: Adding Lognormal Random Variables with Location

Lognormal random variable with spatial location can be added for probabilistic analysis. The location information of the defined random variable is important to calculate the correlation structure of random field, that can consist of many random variables.

The command is:

```
add random variable # <.> with Lognormal distribution mean = <.> ↩
    standard_deviation = <.> at (<L>, <L>, <L>);
```

where:

- `mean` is the mean of the lognormal random variable
- `standard_deviation` is the standard deviation of the lognormal random variable

For example:

```
add random variable # 1 with Lognormal distribution mean = 3.0 ↩
    standard_deviation = 1.0 at (3*m, 0*m, 0*m);
```

Adds a Lognormal random variable 1 with mean equal to 3.0 and standard deviation equal to 1.0 at location $x = 3$m, $y = 0$m and $z = 0$m.

205.3.4.145 Modeling, Random Variable: Adding Lognormal Random Variables using Logarithmic Input

Lognormal random variable can be added for probabilistic analysis.

The command is:

```
add random variable # <.> with Lognormal distribution lognormal_mean = <.> ↩
    lognormal_standard_deviation = <.>;
```

where:

- `lognormal_mean`: μ is the mean of the natural logarithm of the lognormal random variable X
- `lognormal_standard_deviation`: σ is the standard deviation of the natural logarithm of the lognormal random variable X

In other words, for lognormal distributed random variable X with parameters μ and σ, we have:

$$ln(X) \sim N(\mu, \sigma) \tag{205.34}$$

It is noted that the mean m and variance v of lognormal random variable X is related to parameters μ and σ as follows:

$$m = e^{\mu+\sigma^2/2} \tag{205.35}$$

$$v = e^{2\mu+\sigma^2}(e^{\sigma^2} - 1) \tag{205.36}$$

For example:

```
add random variable # 1 with Lognormal distribution lognormal_mean = 3.0 ↩
    lognormal_standard_deviation = 1.0;
```

adds a Lognormal random variable 1. The natural logarithm of such random variable follows Gaussian distribution with mean equal to 3.0 and standard deviation equal to 1.0.

205.3.4.146 Modeling, Random Variable: Adding Lognormal Random Variables using Logarithmic Input with Location

Lognormal random variable with spatial location can be added for probabilistic analysis. The location information of the defined random variable is important to calculate the correlation structure of random field, which can consist of many random variables.

The command is:

```
add random variable # <.> with Lognormal distribution lognormal_mean = <.> ↩
    lognormal_standard_deviation = <.> at (<L>, <L>, <L>);
```

where:

- `lognormal_mean`: μ is the mean of the natural logarithm of the lognormal random variable X
- `lognormal_standard_deviation`: σ is the standard deviation of the natural logarithm of the lognormal random variable X

In other words, for lognormal distributed random variable X with parameters μ and σ, we have:

$$ln(X) \sim N(\mu, \sigma) \tag{205.37}$$

It is noted that the mean m and variance v of lognormal random variable X is related to parameters μ and σ as follows:

$$m = e^{\mu+\sigma^2/2} \tag{205.38}$$

$$v = e^{2\mu+\sigma^2}(e^{\sigma^2} - 1) \tag{205.39}$$

For example:

```
add random variable # 1 with Lognormal distribution lognormal_mean = 3.0 ↩
    lognormal_standard_deviation = 1.0 at (3*m, 0*m, 0*m);
```

Adds a Lognormal random variable 1 at location $x = 3$m, $y = 0$m and $z = 0$m. The natural logarithm of such random variable follows Gaussian distribution with mean equal to 3.0 and standard deviation equal to 1.0.

205.3.4.147 Modeling, Random Variable: Adding Gamma Random Variables using Shape and Scale Parameters

Random variable following Gamma distribution can be added for probabilistic analysis.

The command is:

```
add random variable # <.> with Gamma distribution shape_parameter = <.> ↩
    scale_parameter = <.>;
```

where:

- `shape_parameter` is the shape parameter of the Gamma random variable
- `scale_parameter` is the scale parameter of the Gamma random variable

For example:

```
add random variable # 1 with Gamma distribution shape_parameter = 5.0 ↩
    scale_parameter = 2.0;
```

Adds a Gamma random variable 1 with the shape parameter equal to 5.0 and scale parameter equal to 2.0.

205.3.4.148 Modeling, Random Variable: Adding Gamma Random Variables using Shape and Scale Parameters with Location

Random variable following Gamma distribution can be added for probabilistic analysis.

The location information of the defined random variable is important to calculate the correlation structure of random field, that can consist of many random variables.

The command is:

```
add random variable # <.> with Gamma distribution shape_parameter = <.> ↩
    scale_parameter = <.> at (<L>, <L>, <L>);
```

where:

- `shape_parameter` is the shape parameter of the Gamma random variable
- `scale_parameter` is the scale parameter of the Gamma random variable

For example:

```
add random variable # 1 with Gamma distribution shape_parameter = 5.0 ↩
    scale_parameter = 2.0 at (3*m, 0*m, 0*m);
```

Adds a Gamma random variable 1 with the shape parameter equal to 5.0 and scale parameter equal to 2.0 at location $x = 3$m, $y = 0$m and $z = 0$m.

205.3.4.149 Modeling, Random Variable: Adding Gamma Random Variables using Mean and Standard Deviation Parameters

Random variable for given mean and standard deviation parameter following Gamma distribution can be added for probabilistic analysis.

The command is:

```
add random variable # <.> with Gamma distribution mean = <.> standard_deviation ↩
    = <.>;
```

where:

- `mean` is the mean of the Gamma random variable
- `standard_deviation` is the standard deviation of the Gamma random variable

For example:

```
add random variable # 1 with Gamma distribution mean = 5.0 standard_deviation = ↩
    2.0;
```

Adds a Gamma random variable 1 with mean equal to 5.0 and standard deviation equal to 2.0.

205.3.4.150 Modeling, Random Variable: Adding Gamma Random Variables using Mean and Standard Deviation Parameters with Location

Random variable for given mean and standard deviation parameter following Gamma distribution can be added for probabilistic analysis.

The location information of the defined random variable is important to calculate the correlation structure of random field, that can consist of many random variables.

The command is:

```
add random variable # <.> with Gamma distribution mean = <.> standard_deviation ↩
    = <.> at (<L>, <L>, <L>);
```

where:

- `mean` is the mean of the Gamma random variable
- `standard_deviation` is the standard deviation of the Gamma random variable

For example:

```
add random variable # 1 with Gamma distribution mean = 5.0 standard_deviation = ↩
    2.0 at (3*m, 0*m, 0*m);
```

Adds a Gamma random variable 1 with mean equal to 5.0 and standard deviation equal to 2.0 at location $x = 3$m, $y = 0$m and $z = 0$m.

205.3.4.151 Modeling, Random Variable: Adding Weibull Random Variables using Shape and Scale Parameters

Random variable following Weibull distribution can be added for probabilistic analysis.

The command is:

```
add random variable # <.> with Weibull distribution shape_parameter = <.> ↩
    scale_parameter = <.>;
```

where:

- `shape_parameter` is the shape parameter of the Weibull random variable
- `scale_parameter` is the scale parameter of the Weibull random variable

For example:

```
add random variable # 1 with Weibull distribution shape_parameter = 5.0 ↩
    scale_parameter = 2.0;
```

Adds a Weibull random variable 1 with the shape parameter equal to 5.0 and scale parameter equal to 2.0.

205.3.4.152 Modeling, Random Variable: Adding Weibull Random Variables using Shape and Scale Parameters with Location

Random variable following Weibull distribution can be added for probabilistic analysis.

The location information of the defined random variable is important to calculate the correlation structure of random field, which can consist of many random variables.

The command is:

```
add random variable # <.> with Weibull distribution shape_parameter = <.> ↩
    scale_parameter = <.> at (<L>, <L>, <L>);
```

where:

- `shape_parameter` is the shape parameter of the Weibull random variable
- `scale_parameter` is the scale parameter of the Weibull random variable

For example:

```
add random variable # 1 with Weibull distribution shape_parameter = 5.0 ↩
    scale_parameter = 2.0 at (3*m, 0*m, 0*m);
```

adds a Weibull random variable 1 with the shape parameter equal to 5.0 and scale parameter equal to 2.0 at location $x = 3$m, $y = 0$m and $z = 0$m.

205.3.4.153 Modeling, Random Variable: Remove Random Variables

Remove random variables.

The command is:

```
remove random variable # <.> ;
```

For example:

```
remove random variable # 2;
```

Remove random variable 2 from the analysis.

205.3.4.154 Modeling, Random Variable: Hermite Polynomial Chaos Expansion

Hermite polynomial chaos expansion Xiu (2010) can be performed for random variable with any type of distribution.

The command is:

```
Hermite polynomial chaos expansion to random variable # <.> with order <.>;
```

where:

- `order` specifies the order of Hermite polynomial chaos expansion

For example:

```
Hermite polynomial chaos expansion to random variable # 1 with order 6;
```

Performs Hermite polynomial chaos expansion to random variable 1 using Hermite polynomial chaos up to order 6.

205.3.4.155 Modeling, Random Variable: Output Hermite Polynomial Chaos Expansion Result

A HDF5 (.hdf5) file contains computed polynomial chaos coefficients of Hermite polynomial chaos expansion for random variable can be generated.

The command is:

```
generate Hermite polynomial chaos expansion file from random variable # <.> ↩
   hdf5_file = "file_name";
```

where:

- `file_name` is a string that specifies the name of the output hdf5 file.

The generated hdf5 file contains two datasets:

- Dataset PC is a 2D array that describes the multi-dimensional Hermite polynomial chaos (PC) basis. $\boldsymbol{PC_{ij}}$ denotes the order of polynomial chaos dimension ξ_j that contributes to the i^{th} multi-dimensional Hermite PC basis.
- Dataset PC Coefficients is a column vector. The i^{th} component of PC Coefficients is the polynomial chaos coefficient corresponding to the i^{th} PC base as described by PC.

For example:

```
generate Hermite polynomial chaos expansion file from random variable # 2 ↩
   hdf5_file = "PC_RV1.hdf5";
```

Generate HDF5 file named "PC_RV2.hdf5" that contains the computed polynomial chaos coefficients of Hermite polynomial chaos expansion of random variable 2.

205.3.4.156 Modeling, Random Variable: Hermite Polynomial Chaos Expansion & Output Results

Hermite polynomial chaos expansion Xiu (2010) can be performed for random variable with any type of distribution. A HDF5 (.hdf5) file contains computed polynomial chaos coefficients of Hermite polynomial chaos expansion for random variable can be generated.

The command is:

```
generate Hermite polynomial chaos expansion file from random variable # <.> ↩
    with order <.> hdf5_file = "file_name";
```

where:

- `order` specifies the order of Hermite polynomial chaos expansion
- `file_name` is a string that specifies the name of the output hdf5 file.

The generated hdf5 file contains two datasets:

- Dataset PC is a 2D array that describes the multi-dimensional Hermite polynomial chaos (PC) basis. $\boldsymbol{PC_{ij}}$ denotes the order of polynomial chaos dimension ξ_j that contributes to the i^{th} multi-dimensional Hermite PC basis.
- Dataset PC Coefficients is a column vector. The i^{th} component of PC Coefficients is the polynomial chaos coefficient corresponding to the i^{th} PC base as described by PC.

For example:

```
generate Hermite polynomial chaos expansion file from random variable # 1 with ↩
    order 6 hdf5_file = "PC_RV1.hdf5";
```

Perform Hermite polynomial chaos expansion to random variable 1 using Hermite polynomial chaos up to order 6 and generate HDF5 file named "PC_RV1.hdf5" that contains the computed polynomial chaos coefficients.

205.3.4.157 Modeling, Random Field: Adding Random Field with Dimension and Order

Random field with specific Hermite polynomial chaos dimension and order can be added for probabilistic analysis.

The command is:

```
add random field # <.> with Hermite polynomial chaos dimension <.> order <.>;
```

where:

- `dimension` defines the dimension of Hermite polynomial chaos expansion of the random field
- `order` defines the order of Hermite polynomial chaos expansion of the random field

For example:

```
add random field # 1 with Hermite polynomial chaos dimension 4 order 3;
```

adds random field 1 with Hermite polynomial chaos expansion of dimension 4 and order 3.

205.3.4.158 Modeling, Random Field: Define Global Dimension Index of Random Field

Define the global dimension index of local Hermite polynomial chaos (PC) dimension for the uncertainty characterization of the random field.

If the global dimension index is not specified, by default Real-ESSI takes the ID of local Hermite polynomial chaos (PC) dimension as the global dimension index.

Please note that correctly specifying global dimension index for local Hermite PC dimensions of the random field is very important, especially when there are multiple random fields exist in the system and need to compute the triple products of Hermite PC basis of these random fields.

The command is:

```
define random field # <> Hermite polynomial chaos dimension # <> as global ↩
   dimension # <>;
```

where:

- `Hermite polynomial chaos dimension` defines the local dimension ID for Hermite polynomial chaos (PC) basis of the random field. It should be an integer no more than the total number of dimensions adopted in the Hermite polynomial chaos (PC) Karhunen Loève expansion of the random field.
- `global dimension` defines the corresponding global dimension index for the local Hermite PC dimension

For example:

```
define random field # 1 Hermite polynomial chaos dimension # 1 as global ↩
   dimension # 10;
```

defines the global dimension index of local Hermite PC dimension 1 of random field 1 is 10.

205.3.4.159 Modeling, Random Field: Define Global Dimension Index of Random Field from File Input

Define the global dimension index of local Hermite polynomial chaos (PC) dimension for the uncertainty characterization of the random field using inputs from a text file.

If the global dimension index is not specified, by default Real-ESSI takes the ID of local Hermite polynomial chaos (PC) dimension as the global dimension index.

Please note that correctly specifying global dimension index for local Hermite PC dimensions of the random field is very important, especially when there are multiple random fields exist in the system and need to compute the triple products of Hermite PC basis of these random fields.

The command is:

```
define random field # <.> Hermite polynomial chaos dimension from ↩
    dimension_file = "file_name";
```

where:

- `dimension_file` specifies the name of a text file that contains two columns: The first column is local dimension ID of Hermite PC basis; The second column is corresponding global dimension ID for the local dimension of Hermite PC basis. Comments lines starts with "//"

For example:

```
define random field # 1 Hermite polynomial chaos dimension from dimension_file ↩
    = "dimension_info_RF1.txt";
```

defines the global dimension index of local Hermite PC dimensions for random field 1 with input from a text file "dimension_info_RF1.txt".

An example file of "dimension_info_RF1.txt" is provided below:

```
//====================================================================
// This file specify the global dimension index of local dimensions
// of Hermite PC basis
// File should have two columns separated by spaces:
// The first column is local KL dimension ID
// The second column is the global KL dimension ID
//====================================================================
1 10
2 11
3 12
4 13
```

205.3.4.160 Modeling, Random Field: Set Number of Polynomial Chaos Terms of Random Field

Specify the number of polynomial chaos terms of a random field involved in the stochastic finite element analysis. By default, the full polynomial chaos basis of a random field would be used for uncertainty propagation. The specified number of polynomial chaos terms in this command is used for truncation of polynomial chaos basis.

The command is:

```
set random field # <.> polynomial_chaos_terms = <.>;
```

where:

- `polynomial_chaos_terms` defines the number of truncated Hermite polynomial chaos basis of the random field

For example:

```
set random field # 1 polynomial_chaos_terms = 100;
```

Set the number of truncated Hermite polynomial chaos basis of random field 1 to be 100.

205.3.4.161 Modeling, Random Field: Adding Random Field with Zero Correlation

Random field with uncorrelated random variables can be added for probabilistic analysis.

The command is:

```
add random field # <.> with zero correlation;
```

205.3.4.162 Modeling, Random Field: Adding Random Field with Exponential Correlation

Random field with exponential correlation can be added for probabilistic analysis.

The command is:

```
add random field # <.> with exponential correlation correlation_length = <L> ;
```

where:

- `correlation_length` l_c defines the correlation length of random field such that the correlation $\rho(RV_i, RV_j)$ of any two random variables RV_i and RV_j is given as:

$$\rho(RV_i, RV_j) = exp(-d/l_c) \tag{205.40}$$

Variable d is the Euclidean distance between RV_i and RV_j, that is calculated from the spatial locations of random variables within the random field.

For example:

```
add random field # 1 with exponential correlation correlation_length = 10*m;
```

adds an exponentially correlated random field number 1 with correlation length 10m.

205.3.4.163 Modeling, Random Field: Adding Random Field with Triangular Correlation

Random field with triangular correlation can be added for probabilistic analysis.

The command is:

```
add random field # <.> with triangular correlation correlation_length = <L> ;
```

where:

- `correlation_length` l_c defines the correlation length of random field such that the correlation $\rho(RV_i, RV_j)$ of any two random variables RV_i and RV_j is given as:

$$\rho(RV_i, RV_j) = \max\{1 - \mathrm{d}/\mathrm{l_c},\ 0\} \tag{205.41}$$

Variable d is the Euclidean distance between RV_i and RV_j, that is calculated from the spatial locations of random variables within the random field.

For example:

```
add random field # 1 with triangular correlation correlation_length = 10*m;
```

Adds an triangular correlated random field number 1 with correlation length 10m.

205.3.4.164 Modeling, Random Field: Adding Random Field with Exponentially Damped Cosine Correlation

Random field with exponentially damped cosine correlation can be added for probabilistic analysis.

The command is:

```
add random field # <.> with exponentially damped cosine correlation ↩
   correlation_length = <L> ;
```

where:

- `correlation_length` l_c defines the correlation length of random field such that the correlation $\rho(RV_i, RV_j)$ of any two random variables RV_i and RV_j is given as:

$$\rho(RV_i, RV_j) = exp(-d/l_c) * cos(d/l_c) \tag{205.42}$$

Variable d is the Euclidean distance between RV_i and RV_j, that is calculated from the spatial locations of random variables within the random field.

For example:

```
add random field # 1 with exponentially damped cosine correlation ↩
   correlation_length = 10*m;
```

adds an exponentially damped cosine correlated random field number 1 with correlation length 10m.

205.3.4.165 Modeling, Random Field: Adding Random Field with Gaussian Correlation

Random field with Gaussian correlation can be added for probabilistic analysis.

The command is:

```
add random field # <.> with Gaussian correlation correlation_length = <L> ;
```

where:

- `correlation_length` l_c defines the correlation length of random field such that the correlation $\rho(RV_i, RV_j)$ of any two random variables RV_i and RV_j is given as:

$$\rho(RV_i, RV_j) = exp(-d^2/l_c^2) \tag{205.43}$$

Variable d is the Euclidean distance between RV_i and RV_j, that is calculated from the spatial locations of random variables within the random field.

For example:

```
add random field # 1 with Gaussian correlation correlation_length = 10*m;
```

adds a random field number 1 with Gaussian correlation and correlation length 10m.

205.3.4.166 Modeling, Random Field: Remove Random Fields

Remove random Fields.

The command is:

```
remove random field # <.> ;
```

For example:

```
remove random field # 2;
```

Remove random field 2 from the analysis.

205.3.4.167 Modeling, Random Field: Adding Random Variable to Random Field

Add random variable to random field.

The command is:

```
add random variable # <.> to random field # <.>;
```

For example:

```
add random variable # 2 to random field # 1;
```

Adds random variable 2 to random field 1.

205.3.4.168 Modeling, Random Field: Remove Random Variable From Random Field

Remove random variable from random field.

The command is:

```
remove random variable # <.> from random field # <.>;
```

For example:

```
remove random variable # 2 from random field # 1;
```

Remove random variable 2 from random field 1.

205.3.4.169 Modeling, Random Field: Hermite Polynomial Chaos Karhunen Loève Expansion

Perform Hermite polynomial chaos Karhunen Loève expansion for random field of any arbitrary marginal distribution and correlation structure according to Sakamoto and Ghanem (2002).

The command is:

```
Hermite polynomial chaos Karhunen Loeve expansion to random field # <.> with ↩
   Hermite polynomial chaos dimension <.> order <.>;
```

Where:

- `dimension`: specifies the number of dimensions of Hermite polynomial chaos to capture the correlation structure of the random field
- `order`: specifies the order of Hermite polynomial chaos to capture the marginal distribution of the random field

For example:

```
Hermite polynomial chaos Karhunen Loeve expansion to random field # 1 with ↩
   Hermite polynomial chaos dimension 4 order 3;
```

Perform the Hermite polynomial chaos Karhunen Loève expansion for random field 1 using Hermite polynomial chaos of dimension 4 and order 3.

205.3.4.170 Modeling, Random Field: Hermite Polynomial Chaos Karhunen Loève Expansion with Inverse Order

Perform Hermite polynomial chaos Karhunen Loève expansion for random field of any arbitrary marginal distribution and correlation structure according to Sakamoto and Ghanem (2002).

The user can explicitly state the order used in the inversion of underlying Gaussian correlation kernel. The command is:

```
Hermite polynomial chaos Karhunen Loeve expansion to random field # <.> with ↩
    Hermite polynomial chaos dimension <.> order <.> ↩
    correlation_kernel_inverse_order = <.>;
```

Where:

- `dimension`: specifies the number of dimensions of Hermite polynomial chaos to capture the correlation structure of the random field
- `order`: specifies the order of Hermite polynomial chaos to capture the marginal distribution of the random field
- `correlation_kernel_inverse_order`: specifies the order used in the inversion of underlying Gaussian correlation kernel. For the exact Gaussian kernel inversion, set up `correlation_kernel_inverse_order` equal to order of Hermite polynomial chaos. `correlation_kernel_inverse_order` should not exceed `order` of Hermite polynomial chaos. If `correlation_kernel_inverse_order` is not stated, by default linear Gaussian kernel inversion, i.e., `correlation_kernel_inverse_order` equal to 1, is performed as the approximation of higher order inversion. See Sakamoto and Ghanem (2002) for more details.

For example:

```
Hermite polynomial chaos Karhunen Loeve expansion to random field # 1 with ↩
    Hermite polynomial chaos dimension 4 order 2 ↩
    correlation_kernel_inverse_order = 2;
```

Perform the Hermite polynomial chaos Karhunen Loève expansion for random field 1 using Hermite polynomial chaos of dimension 4 and order 2. The 2^{nd} order Gaussian correlation kernel inversion is adopted.

205.3.4.171 Modeling, Random Field: Hermite Polynomial Chaos Karhunen Loève Expansion with Number of FE Elements Larger than Dimension of Hermite Polynomials

Perform Hermite polynomial chaos Karhunen Loève expansion for random field described by either Gaussian or lognormal distribution and Gaussian auto-correlation coefficient function. The PCE coefficients are computed using Eq. (11) from Sakamoto and Ghanem (2002).

This command should be used when the number of Gauss points (GPs), i.e., integration points, is much larger than the number of discrete locations needed to solve the eigenproblem of auto-covariance, that is, than the dimension of Hermite polynomials. This allows to save some computation time. In such case, the eigenproblem of auto-covariance function will be solved instead of the eigenproblem of auto-covariance matrix.

Here, "shear beam" element is used. It has only one GP, in the middle of an element. Hence the number of GPs is here equal to the number of FE elements. It is easier for a user to input the number of FE elements than to input the number of GPs.

The command is:

```
Hermite polynomial chaos Karhunen Loeve expansion to random field # <.> with ↩
   Hermite polynomial chaos dimension <.> order <.> ↩
   correlation_kernel_inverse_order = 1 number_of_FE_elements = <.>;
```

Where:

- `dimension`: specifies the number of dimensions of Hermite polynomial chaos to capture the correlation structure of the random field, in other words, it is the number of discrete points used in the solution of the eigenproblem of auto-covariance
- `order`: specifies the order of Hermite polynomial chaos to capture the marginal distribution of the random field
- `correlation_kernel_inverse_order = 1`: does not specify anything and must be equal to 1
- `number_of_FE_elements`: here, "shear beam" element is used and it has only one GP, in the middle of an element, hence the number of FE elements is here equal to the number of GPs, the user must provide (via command `add random variable...`) the type of distribution (here, either Gaussian or lognormal), mean and standard deviation in the middle of each FE element (mean and standard deviation at discrete points used in the solution of the eigenproblem of auto-covariance function are interpolated using the values at GPs), they should be sorted for increasing coordinates in 1D

  `number_of_FE_elements = dimension` is equivalent to:

```
Hermite polynomial chaos Karhunen Loeve expansion to random field # <.> ↩
   with Hermite polynomial chaos dimension <.> order <.> ;
```

and:

```
Hermite polynomial chaos Karhunen Loeve expansion to random field # <.> ↩
   with Hermite polynomial chaos dimension <.> order <.> ↩
   correlation_kernel_inverse_order = 1;
```

For example:

```
Hermite polynomial chaos Karhunen Loeve expansion to random field # 1 with ↩
   Hermite polynomial chaos dimension 4 order 2 ↩
   correlation_kernel_inverse_order = 1 number_of_FE_elements = 20;
```

Perform the Hermite polynomial chaos Karhunen Loève expansion for random field 1 using Hermite polynomial chaos of dimension 4 and order 2 with number of FE elements 20 (i.e., with 20 GPs, in the middle of elements).

Command:

```
Hermite polynomial chaos Karhunen Loeve expansion to random field # <.> with ↩
   Hermite polynomial chaos dimension <.> order <.> number_of_FE_elements = <.>;
```

with optional argument `number_of_FE_elements` cannot be defined because

```
Hermite polynomial chaos Karhunen Loeve expansion to random field # <.> with ↩
   Hermite polynomial chaos dimension <.> order <.> ↩
   correlation_kernel_inverse_order = <.>;
```

with optional argument `correlation_kernel_inverse_order` exists already.

205.3.4.172 Modeling, Random Field: Hermite Polynomial Chaos Karhunen Loève Expansion Using HDF5 Input

Add a random field with marginal distribution and correlation information defined in a given HDF5 (.hdf5) file. Perform Hermite polynomial chaos Karhunen Loève expansion for the random field.

The command is:

```
Hermite polynomial chaos Karhunen Loeve expansion to random field # <.> with ↩
   Hermite polynomial chaos dimension <.> order <.> hdf5_file = "file_name";
```

Where:

- `dimension`: specifies the number of dimensions of Hermite polynomial chaos to capture the correlation structure of the random field
- `order`: specifies the order of Hermite polynomial chaos to capture the marginal distribution of the random field
- `hdf5_file`: specifies the filename of the input HDF5 file that defines the marginal distribution and correlation information of the random field

The input HDF5 file should contain the following datasets:

- Dataset Random Field contains a single integer, which is the ID of the random field.
- Dataset Marginal Mean is a column vector specifying marginal mean of the random field corresponding to each random variable.
- Dataset Marginal Variance is a column vector specifying marginal variance of the random field for each random variable.
- Dataset Marginal Distributions is a column vector integers specifying the marginal distribution IDs of the random field for each random variable. Specifically, the ID is 1 for Gaussian distribution, 2 for Lognormal distribution, 3 for Gamma distribution and 4 for Weibull distribution.
- Dataset Correlation is a 2D array specifying correlation of the random field among random variables.
- Dataset PC Order contains a single integer, which specifies the order of Hermite polynomial chaos (PC) Karhunen Loève expansion.
- Dataset PC Dimension contains a single integer, which specifies the dimension of Hermite polynomial chaos (PC) Karhunen Loève expansion.

- Dataset Index to Global Dimension contains a column vector of integers, which specifies the index of corresponding global PC dimension for each local PC dimension used in the uncertainty expansion of the random field.

For example:

```
Hermite polynomial chaos Karhunen Loeve expansion to random field # 1 with ↩
   Hermite polynomial chaos dimension 4 order 4 hdf5_file = "PC_RF1.hdf5";
```

Perform the Hermite polynomial chaos Karhunen Loève expansion for random field 1 using Hermite polynomial chaos of dimension 4 and order 4 with input marginal distribution and correlation information defined in HDF5 file "PC_RF1.hdf5".

205.3.4.173 Modeling, Random Field: Hermite Polynomial Chaos Karhunen Loève Expansion with Inverse Order Using HDF5 Input

Add a random field with marginal distribution and correlation information defined in a given HDF5 (.hdf5) file. Perform Hermite polynomial chaos Karhunen Loève expansion for the random field.

The user can explicitly state the order used in the inversion of underlying Gaussian correlation kernel.

The command is:

```
Hermite polynomial chaos Karhunen Loeve expansion to random field # <.> with ↩
    Hermite polynomial chaos dimension <.> order <.> ↩
    correlation_kernel_inverse_order = <.> hdf5_file = "file_name";
```

Where:

- `dimension`: specifies the number of dimensions of Hermite polynomial chaos to capture the correlation structure of the random field
- `order`: specifies the order of Hermite polynomial chaos to capture the marginal distribution of the random field
- `correlation_kernel_inverse_order`: specifies the order used in the inversion of underlying Gaussian correlation kernel. For the exact Gaussian kernel inversion, set up `correlation_kernel_inverse_order` equal to `order` of Hermite polynomial chaos. `correlation_kernel_inverse_order` should not exceed `order` of Hermite polynomial chaos. If `correlation_kernel_inverse_order` is not stated, by default linear Gaussian kernel inversion, i.e., `correlation_kernel_inverse_order` equal to 1, is performed as the approximation of higher order inversion. See Sakamoto and Ghanem (2002) for more details.
- `hdf5_file`: specifies the filename of the input HDF5 file that defines the marginal distribution and correlation information of the random field

The input HDF5 file should contain the following datasets:

- Dataset Random Field contains a single integer, which is the ID of the random field.
- Dataset Marginal Mean is a column vector specifying marginal mean of the random field corresponding to each random variable.
- Dataset Marginal Variance is a column vector specifying marginal variance of the random field for each random variable.

- Dataset Marginal Distributions is a column vector integers specifying the marginal distribution IDs of the random field for each random variable. Specifically, the ID is 1 for Gaussian distribution, 2 for Lognormal distribution, 3 for Gamma distribution and 4 for Weibull distribution.
- Dataset Correlation is a 2D array specifying correlation of the random field among random variables.
- Dataset PC Order contains a single integer, which specifies the order of Hermite polynomial chaos (PC) Karhunen Loève expansion.
- Dataset PC Dimension contains a single integer, which specifies the dimension of Hermite polynomial chaos (PC) Karhunen Loève expansion.
- Dataset Index to Global Dimension contains a column vector of integers, which specifies the index of corresponding global PC dimension for each local PC dimension used in the uncertainty expansion of the random field.

For example:

```
Hermite polynomial chaos Karhunen Loeve expansion to random field # 1 with ↩
   Hermite polynomial chaos dimension 4 order 4 ↩
   correlation_kernel_inverse_order = 3 hdf5_file = "PC_RF1.hdf5";
```

Perform the Hermite polynomial chaos Karhunen Loève expansion for random field 1 using Hermite polynomial chaos of dimension 4 and order 4. The 3^{rd} order Gaussian correlation kernel inversion is adopted. The input marginal distribution and correlation information are defined in HDF5 file ”PC_RF1.hdf5”.

205.3.4.174 Modeling, Random Field: Output Hermite Polynomial Chaos Karhunen Loève Expansion Result

A HDF5 (.hdf5) file contains all the information for Hermite polynomial chaos Karhunen Loève expansion for random field can be generated.

The command is:

```
generate Hermite polynomial chaos Karhunen Loeve expansion file from random ↩
   field # <.> hdf5_file = "file_name";
```

where:

- `file_name` is a string that specifies the name of the output hdf5 file.

The generated hdf5 file contains the following datasets:

- Dataset Random Field contains a single integer, which is the ID of the random field.
- Dataset Random Variables contains a column vector of integers, which are the IDs of the random variables that constitute the random field.
- Dataset Marginal Mean is a column vector specifying marginal mean of the random field corresponding to each random variable.
- Dataset Marginal Variance is a column vector specifying marginal variance of the random field for each random variable.
- Dataset Marginal Distributions is a column vector integers specifying the marginal distribution IDs of the random field for each random variable. Specifically, the ID is 1 for Gaussian distribution, 2 for Lognormal distribution, 3 for Gamma distribution and 4 for Weibull distribution.
- Dataset Correlation is a 2D array specifying correlation of the random field among random variables.
- Dataset PC Order contains a single integer, which specifies the order of Hermite polynomial chaos (PC) Karhunen Loève expansion.
- Dataset PC Dimension contains a single integer, which specifies the dimension of Hermite polynomial chaos (PC) Karhunen Loève expansion.
- Dataset Index to Global Dimension contains a column vector of integers, which specifies the index of corresponding global PC dimension for each local PC dimension used in the uncertainty expansion of the random field.

- Dataset PC is a 2D array that describes the multi-dimensional Hermite polynomial chaos (PC) basis. $\boldsymbol{PC_{ij}}$ denotes the order of polynomial chaos dimension ξ_j that contributes to the i^{th} multi-dimensional Hermite PC basis.
- Dataset PC Coefficients is a 2D array. The j^{th} component of the i^{th} row is the polynomial chaos coefficient corresponding to the j^{th} PC base as described by PC for the i^{th} random variable specified in Random Variables.
- Dataset PC Variance is a column vector specifying the variances of Hermite PC basis.

For example:

```
generate Hermite polynomial chaos Karhunen Loeve expansion file from random ↩
    field # 1 hdf5_file = "PC_RF1.hdf5";
```

Generate HDF5 file named “PC_RF1.hdf5” that contains all the information for Hermite polynomial chaos Karhunen Loève expansion of random field 1.

205.3.4.175 Modeling, Random Field: Adding Random Field from Hermite Polynomial Chaos Karhunen Loève Expansion HDF5 File

Add a random field with marginal distribution, correlation information and multi-dimensional Hermite polynomial chaos (PC) coefficients specified in a given HDF5 (.hdf5) file.

The command is:

```
add random field # <.> with Hermite polynomial chaos Karhunen Loeve expansion ↩
   hdf5_file = "file_name";
```

where:

- `file_name` is a string that specifies the name of the input hdf5 file.

The input hdf5 file should contain the following datasets:

- Dataset Random Field contains a single integer, that represents the ID of the random field.
- Dataset Random Variables contains a column vector of integers, that are the IDs of the random variables that constitute the random field.
- Dataset Marginal Mean is a column vector specifying marginal mean of the random field corresponding to each random variable.
- Dataset Marginal Variance is a column vector specifying marginal variance of the random field for each random variable.
- Dataset Marginal Distributions is a column vector of integers specifying the marginal distribution IDs of the random field for each random variable. Specifically, the ID is 1 for Gaussian distribution, 2 for Lognormal distribution, 3 for Gamma distribution and 4 for Weibull distribution.
- Dataset Correlation is a 2D array specifying correlation of the random field among random variables.
- Dataset PC Order contains a single integer, that specifies the order of Hermite polynomial chaos (PC) Karhunen Loève expansion.
- Dataset PC Dimension contains a single integer, that specifies the dimension of Hermite polynomial chaos (PC) Karhunen Loève expansion.
- Dataset Index to Global Dimension contains a column vector of integers, that specify the index of corresponding global PC dimension for each local PC dimension used in the uncertainty expansion of the random field.

- Dataset PC is a 2D array that describes the multi-dimensional Hermite polynomial chaos (PC) basis. $\boldsymbol{PC_{ij}}$ denotes the order of polynomial chaos dimension ξ_j that contributes to the i^{th} multi-dimensional Hermite PC basis.
- Dataset PC Coefficients is a 2D array. The j^{th} component of the i^{th} row is the polynomial chaos coefficient corresponding to the j^{th} PC base as described by PC for the i^{th} random variable specified in Random Variables.
- Dataset PC Variance is a column vector specifying the variances of Hermite PC basis.

For example:

```
add random field # 1 with Hermite polynomial chaos Karhunen Loeve expansion ↩
   hdf5_file = "PC_RF1.hdf5";
```

Add random field 1 with the marginal distribution, correlation structure and Hermite polynomial chaos Karhunen Loève expansion information defined in HDF5 file named “PC_RF1.hdf5”.

205.3.4.176 Modeling, Random Field: Adding Random Field from Marginal Distribution and Correlation

Add a random field with specified marginal distribution and correlation information.

The command is:

```
add random field # <.> with <distribution_type> distribution
marginal_mean_file = "file_name"
marginal_standard_deviation_file = "file_name"
correlation_file = "file_name";
```

where:

- `distribution_type` is a string specifying the marginal distribution type of the random field, can be `Gaussian`, `Lognormal`, or `Gamma`.
- `marginal_mean_file` is a string specifying the name of a plain text file that contains a single column of marginal mean of the random field.
- `marginal_standard_deviation_file` is a string specifying the name of a plain text file that contains a single column of marginal standard deviation of the random field.
- `correlation_file` is a string specifying the name of a plain text file that contains a 2D array of the correlation structure of the random field.

205.3.4.177 Modeling, Random Field: Add Triple Product of Hermite Polynomial Chaos Basis

Compute and add triple product of Hermite polynomial chaos basis from three different random fields.

The command is:

```
add triple product # <.> with Hermite polynomial chaos from random field (<.>, ↩
   <.>, <.>);
```

For example:

```
add triple product # 1 with Hermite polynomial chaos from random field (1, 2, 3);
```

Compute and add triple product #1 with Hermite polynomial chaos basis from random field 1, 2 and 3;

205.3.4.178 Modeling, Random Field: Add Double Product of Hermite Polynomial Chaos Basis

Compute and add double product of Hermite polynomial chaos basis from two different random fields.

The command is:

```
add double product # <.> with Hermite polynomial chaos from random field (<.>, ↩
    <.>);
```

For example:

```
add double product # 1 with Hermite polynomial chaos from random field (2, 3);
```

Compute and add double product #1 with Hermite polynomial chaos basis from random field 2 and 3;

205.3.4.179 Modeling, Random Field: Generate Triple Product of Hermite Polynomial Chaos Basis

The computation of triple product of Hermite polynomial chaos basis for different random fields is a key part for stochastic finite element analysis.

A HDF5 (.hdf5) file contains triple product of Hermite polynomial chaos basis for random fields can be generated.

The command is:

```
generate triple product of Hermite polynomial chaos from random field (<.>, ↩
    <.>, <.>) hdf5_file = "file_name";
```

where:

- `file_name` is a string that specifies the name of the output HDF5 file.

The generated HDF5 file contains the following datasets:

- Dataset PC1 is a 2D array that describes the multi-dimensional Hermite polynomial chaos (PC) basis of the first random field. $\boldsymbol{PC1_{ij}}$ denotes the order of polynomial chaos dimension ξ_j that contributes to the i^{th} basis of PC1.
- Dataset PC1 Index to Global Dimension contains a column vector of integers, which specifies the global dimension index of each local PC dimension in basis PC1.
- Dataset PC1 Variance is a column vector specifying the variances of basis PC1.
- Dataset PC2 is a 2D array that describes the multi-dimensional Hermite polynomial chaos (PC) basis of the first random field. $\boldsymbol{PC2_{ij}}$ denotes the order of polynomial chaos dimension ξ_j that contributes to the i^{th} basis of PC2.
- Dataset PC2 Index to Global Dimension contains a column vector of integers, which specifies the global dimension index of each local PC dimension in basis PC2.
- Dataset PC2 Variance is a column vector specifying the variances of basis PC2.
- Dataset PC3 is a 2D array that describes the multi-dimensional Hermite polynomial chaos (PC) basis of the first random field. $\boldsymbol{PC3_{ij}}$ denotes the order of polynomial chaos dimension ξ_j that contributes to the i^{th} basis of PC3.
- Dataset PC3 Index to Global Dimension contains a column vector of integers, which specifies the global dimension index of each local PC dimension in basis PC3.

- Dataset PC3 Variance is a column vector specifying the variances of basis PC3.
- Dataset Triple Product is a column vector containing the non-zero triple products of polynomial chaos basis from PC1, PC2 and PC3.
- Dataset Triple Product PC1 Index is a column vector containing the indexes of polynomial chaos basis from PC1 that contributes to the non-zero triple products in dataset Triple Product.
- Dataset Triple Product PC2 Index is a column vector containing the indexes of polynomial chaos basis from PC2 that contributes to the non-zero triple products in dataset Triple Product.
- Dataset Triple Product PC3 Index is a column vector containing the indexes of polynomial chaos basis from PC3 that contributes to the non-zero triple products in dataset Triple Product.

For example:

```
generate triple product of Hermite polynomial chaos from random field (1, 2, 3) ↩
    hdf5_file = "Triple_product_4(3)_4(3)_4(3).hdf5";
```

Compute the triple product of Hermite polynomial chaos basis of random field 1, 2 and 3 and write all the results into a HDF5 file named "Triple_product_4(3)_4(3)_4(3).hdf5";

205.3.4.180 Modeling, Random Field: Generate Double Product of Hermite Polynomial Chaos Basis

The computation of double product of Hermite polynoial chaos basis for different random fields is a key part for stochastic finite element analysis.

A HDF5 (.hdf5) file contains double product of Hermite polynomial chaos basis for random fields can be generated.

The command is:

```
generate double product of Hermite polynomial chaos from random field (<.>, ↩
    <.>) hdf5_file = "file_name";
```

where:

- `file_name` is a string that specifies the name of the output HDF5 file.

The generated HDF5 file contains the following datasets:

- Dataset PC1 is a 2D array that describes the multi-dimensional Hermite polynomial chaos (PC) basis of the first random field. $\boldsymbol{PC1_{ij}}$ denotes the order of polynomial chaos dimension ξ_j that contributes to the i^{th} basis of PC1.
- Dataset PC1 Index to Global Dimension contains a column vector of integers, which specifies the global dimension index of each local PC dimension in basis PC1.
- Dataset PC1 Variance is a column vector specifying the variances of basis PC1.
- Dataset PC2 is a 2D array that describes the multi-dimensional Hermite polynomial chaos (PC) basis of the first random field. $\boldsymbol{PC2_{ij}}$ denotes the order of polynomial chaos dimension ξ_j that contributes to the i^{th} basis of PC2.
- Dataset PC2 Index to Global Dimension contains a column vector of integers, which specifies the global dimension index of each local PC dimension in basis PC2.
- Dataset PC2 Variance is a column vector specifying the variances of basis PC2.
- Dataset Double Product is a column vector containing the non-zero double products of polynomial chaos basis from PC1 and PC2.
- Dataset Double Product PC1 Index is a column vector containing the indexes of polynomial chaos basis from PC1 that contributes to the non-zero double products in dataset Double Product.

- Dataset Double Product PC2 Index is a column vector containing the indexes of polynomial chaos basis from PC2 that contributes to the non-zero double products in dataset Double Product.

For example:

```
generate double product of Hermite polynomial chaos from random field (1, 2) ↩
   hdf5_file = "doubleproduct_153(2)_150(1).hdf5";
```

Compute the triple product of Hermite polynomial chaos basis of random field 1 and 2 and write all the results into a HDF5 file named "doubleproduct_153(2)_150(1).hdf5";

205.3.4.181 Modeling, Random Field: Add Triple Product of Hermite Polynomial Chaos Basis Using HDF5 Input

Add triple product of Hermite polynomial chaos basis using HDF5 input.

The command is:

```
add triple product # <.> from hdf5_file = "file_name";
```

where:

- `file_name` is a string that specifies the name of the input HDF5 file.

The input HDF5 file should contain the following datasets:

- Dataset PC1 is a 2D array that describes the multi-dimensional Hermite polynomial chaos (PC) basis of the first random field. $\boldsymbol{PC1_{ij}}$ denotes the order of polynomial chaos dimension ξ_j that contributes to the i^{th} basis of PC1.
- Dataset PC1 Index to Global Dimension contains a column vector of integers, which specifies the global dimension index of each local PC dimension in basis PC1.
- Dataset PC1 Variance is a column vector specifying the variances of basis PC1.
- Dataset PC2 is a 2D array that describes the multi-dimensional Hermite polynomial chaos (PC) basis of the first random field. $\boldsymbol{PC2_{ij}}$ denotes the order of polynomial chaos dimension ξ_j that contributes to the i^{th} basis of PC2.
- Dataset PC2 Index to Global Dimension contains a column vector of integers, which specifies the global dimension index of each local PC dimension in basis PC2.
- Dataset PC2 Variance is a column vector specifying the variances of basis PC2.
- Dataset PC3 is a 2D array that describes the multi-dimensional Hermite polynomial chaos (PC) basis of the first random field. $\boldsymbol{PC3_{ij}}$ denotes the order of polynomial chaos dimension ξ_j that contributes to the i^{th} basis of PC3.
- Dataset PC3 Index to Global Dimension contains a column vector of integers, which specifies the global dimension index of each local PC dimension in basis PC3.
- Dataset PC3 Variance is a column vector specifying the variances of basis PC3.
- Dataset Triple Product is a column vector containing the non-zero triple products of polynomial chaos basis from PC1, PC2 and PC3.

- Dataset Triple Product PC1 Index is a column vector containing the indexes of polynomial chaos basis from PC1 that contributes to the non-zero triple products in dataset Triple Product.
- Dataset Triple Product PC2 Index is a column vector containing the indexes of polynomial chaos basis from PC2 that contributes to the non-zero triple products in dataset Triple Product.
- Dataset Triple Product PC3 Index is a column vector containing the indexes of polynomial chaos basis from PC3 that contributes to the non-zero triple products in dataset Triple Product.

For example:

```
add triple product # 1 from hdf5_file = "tripleproduct_3(2)_153(2)_153(2).hdf5";
```

Add triple product #1 using HDF5 input file named "tripleproduct_3(2)_153(2)_153(2).hdf5".

205.3.4.182 Modeling, Random Field: Add Double Product of Hermite Polynomial Chaos Basis Using HDF5 Input

Add double product of Hermite polynomial chaos basis using HDF5 input.

The command is:

```
add double product # <.> from hdf5_file = "file_name";
```

where:

- `file_name` is a string that specifies the name of the input HDF5 file.

The input HDF5 file should contain the following datasets:

- Dataset PC1 is a 2D array that describes the multi-dimensional Hermite polynomial chaos (PC) basis of the first random field. $\boldsymbol{PC1_{ij}}$ denotes the order of polynomial chaos dimension ξ_j that contributes to the i^{th} basis of PC1.
- Dataset PC1 Index to Global Dimension contains a column vector of integers, which specifies the global dimension index of each local PC dimension in basis PC1.
- Dataset PC1 Variance is a column vector specifying the variances of basis PC1.
- Dataset PC2 is a 2D array that describes the multi-dimensional Hermite polynomial chaos (PC) basis of the first random field. $\boldsymbol{PC2_{ij}}$ denotes the order of polynomial chaos dimension ξ_j that contributes to the i^{th} basis of PC2.
- Dataset PC2 Index to Global Dimension contains a column vector of integers, which specifies the global dimension index of each local PC dimension in basis PC2.
- Dataset PC2 Variance is a column vector specifying the variances of basis PC2.
- Dataset Double Product is a column vector containing the non-zero double products of polynomial chaos basis from PC1 and PC2.
- Dataset Double Product PC1 Index is a column vector containing the indexes of polynomial chaos basis from PC1 that contributes to the non-zero double products in dataset Double Product.
- Dataset Double Product PC2 Index is a column vector containing the indexes of polynomial chaos basis from PC2 that contributes to the non-zero double products in dataset Double Product.

For example:

```
add double product # 1 from hdf5_file = "doubleproduct_153(2)_150(1).hdf5";
```

Add double product #1 using HDF5 input file named "doubleproduct_153(2)_150(1).hdf5".

205.3.4.183 Modeling, Solid-Fluid Interaction: Adding Solid-Fluid Interface

For solid-fluid interaction analysis, solid fluid interface should be defined and added to the analysis domain.

The command is:

```
add solid fluid interface <string>
```

where:

- `<string>` specifies the name of the boundary of fluid domain that is solid fluid interface. It is noted that the boundary name should be consistent with the definition in the OpenFOAM input file at *constant/polyMesh/boundary*. More information about the organization and format of OpenFOAM input files can be found at OpenFOAM User Guide (OpenCFD Ltd, 2019).

For example:

```
add solid fluid interface "bottom_fluid_surface";
```

Adds fluid boundary named “bottom_fluid_surface” as one of the interface boundaries between solid domain and fluid domain.

205.3.4.184 Modeling, Solid-Fluid Interaction: Defining Solid-Fluid Interface, ESSI Element Nodes

For solid-fluid interaction analysis, solid fluid interface contains information about the Real-ESSI solid interface and OpenFOAM fluid interface. The Real-ESSI solid interface defines Real-ESSI interface nodes and faces.

Real-ESSI interface nodes can be defined as the following:

```
define solid fluid interface ESSI nodes <string>;
```

where:

- `<string>` specifies the plain text file name that contains the information about Real-ESSI interface nodes.

The format of such text file containing Real-ESSI interface nodes information is:

- Comment line starts with "//"
- Each line defining a Real-ESSI interface node has four entries separated by space(s):
  - 1^{st} entry: Real-ESSI node ID;
  - 2^{nd} entry: x coordinate of Real-ESSI interface node;
  - 3^{rd} entry: y coordinate of Real-ESSI interface node;
  - 4^{th} entry: z coordinate of Real-ESSI interface node;

For example:

```
define solid fluid interface ESSI nodes "ESSI_nodes_info.fei";
```

Defines Real-ESSI nodes at solid fluid interface with file named "ESSI_nodes_info.fei". An example file of "ESSI_nodes_info.fei" is provided below:

```
//=====================================================
// Files contains information about ESSI nodes at solid fluid interface, have 4 ↩
   columns
// 1st column: ESSI node ID
// 2nd column: coordinate x
// 3rd column: coordinate y
// 4th column: coordinate z
//======================================================
1 0.00 0.00 0.00
12 30.00 0.00 0.00
33 0.00 0.00 10.00
...
```

205.3.4.185 Modeling, Solid-Fluid Interaction: Defining Solid-Fluid Interface, ESSI Element Faces

For solid-fluid interaction analysis, solid fluid interface contains information about Real-ESSI solid interface and OpenFOAM fluid interface. The Real-ESSI solid interface defines Real-ESSI interface nodes and faces.

Real-ESSI interface faces are quads that can be defined as:

```
define solid fluid interface ESSI faces <string>;
```

where:

- `<string>` specifies the plain text file name that contains the information about Real-ESSI interface elements faces.

The format of such text file containing Real-ESSI interface element faces information is:

- Comment line starts with "//"
- Each line defining a Real-ESSI interface element face, i.e., a face of a brick element, in effect a quad consisting of four Real-ESSI interface nodes. Each line has six entries separated by spaces:
  - 1^{st} entry: Real-ESSI interface face ID;
  - 2^{nd} entry: Real-ESSI element ID that contains the Real-ESSI interface face;
  - 3^{rd} entry: Real-ESSI node ID for Real-ESSI interface face, i.e., a brick face, a quad, vertex 1;
  - 4^{th} entry: Real-ESSI node ID for Real-ESSI interface face, i.e., a brick face, a quad, vertex 2;
  - 5^{th} entry: Real-ESSI node ID for Real-ESSI interface face, i.e., a brick face, a quad, vertex 3;
  - 6^{th} entry: Real-ESSI node ID for Real-ESSI interface face, i.e., a brick face, a quad, vertex 4;

  It is noted that the ordering of the four vertex Real-ESSI nodes is very important. The ordering should be taken such that the face normal vector points outwards to the fluid domain following the convention of right hand rule.

For example:

```
define solid fluid interface ESSI faces "ESSI_faces_info.fei";
```

Defines Real-ESSI faces at solid fluid interface with file named "ESSI_faces_info.fei". An example file of "ESSI_faces_info.fei" is provided below:

```
//===================================================
// Files contains information about ESSI faces (quads) at solid fluid ↩
   interface, have 6 columns
// 1st column: ESSI face ID
// 2nd column: ESSI Element ID that ESSI face belongs to
// 3rd column: ESSI node ID for quad vertex 1
// 4th column: ESSI node ID for quad vertex 2
// 5th column: ESSI node ID for quad vertex 3
// 6th column: ESSI node ID for quad vertex 4
//===================================================
1 276 1 25 272 5
2 277 25 26 273 272
3 278 26 27 274 273
4 279 27 28 275 274
...
```

205.3.4.186 Modeling, Solid-Fluid Interaction: Defining Solid-Fluid Interface FOAM Nodes

For solid-fluid interaction analysis, solid fluid interface contains information about the Real-ESSI solid interface and OpenFOAM fluid interface. The OpenFOAM fluid interface needs to define OpenFOAM interface nodes and faces.

OpenFOAM interface nodes can be defined as the following:

```
define solid fluid interface FOAM nodes <string>;
```

where:

- `<string>` specifies the plain text file name that contains the information about OpenFOAM interface nodes.

The format of such text file containing OpenFOAM interface nodes information is:

- Comment line starts with "//"
- Each line defining a OpenFOAM interface node has four entries separated by space(s):
  - 1^{st} entry: OpenFOAM node ID;
  - 2^{nd} entry: x coordinate of OpenFOAM interface node;
  - 3^{rd} entry: y coordinate of OpenFOAM interface node;
  - 4^{th} entry: z coordinate of OpenFOAM interface node;

For example:

```
define solid fluid interface FOAM nodes "foam_nodes_info.fei";
```

Defines OpenFOAM nodes at solid fluid interface with file named "foam_nodes_info.fei". An example file of "foam_nodes_info.fei" can be:

```
//=====================================================
// Files contains information about Foam nodes at solid fluid interface, have 4 ↩
   columns
// 1st column: Foam node ID
// 2nd column: coordinate x
// 3rd column: coordinate y
// 4th column: coordinate z
//=====================================================
1 0.00 0.00 0.00
12 30.00 0.00 0.00
33 0.00 0.00 10.00
...
```

205.3.4.187 Modeling, Solid-Fluid Interaction: Defining Solid-Fluid Interface FOAM Faces

For solid-fluid interaction analysis, solid fluid interface contains information about the Real-ESSI solid interface and OpenFOAM fluid interface. The OpenFOAM fluid interface needs to define OpenFOAM interface nodes and faces.

OpenFOAM interface faces are quads that can be defined as the following:

```
define solid fluid interface FOAM faces <string>;
```

where:

- `<string>` specifies the plain text file name that contains the information about OpenFOAM interface faces.

The format of such text file containing OpenFOAM interface faces information is:

- Comment line starts with "//"
- Each line defining a OpenFOAM interface face, i.e., a quad consisting of four OpenFOAM interface nodes. Each line has five entries separated by space(s):
  - 1^{st} entry: OpenFOAM face ID;
  - 2^{nd} entry: OpenFOAM node ID for OpenFOAM interface face, i.e., a quad, vertex 1;
  - 3^{rd} entry: OpenFOAM node ID for OpenFOAM interface face, i.e., a quad, vertex 2;
  - 4^{th} entry: OpenFOAM node ID for OpenFOAM interface face, i.e., a quad, vertex 3;
  - 5^{th} entry: OpenFOAM node ID for OpenFOAM interface face, i.e., a quad, vertex 4;

  It is noted that the ordering of the four vertex OpenFOAM nodes is very important. The ordering should be consistent with the OpenFOAM faces definition in the OpenFOAM input file at *constant/polyMesh/faces*. Detailed information about the organization and format of OpenFOAM input files can be found at OpenFOAM User Guide (OpenCFD Ltd, 2019).

For example:

```
define solid fluid interface FOAM faces "foam_faces_info.fei";
```

Defines OpenFOAM faces at solid fluid interface with file named "foam_faces_info.fei". An example file of "foam_faces_info.fei" can be:

```
//=============================================
// Files contains information about Foam faces (quads) at solid fluid ↩
   interface, have 5 columns
// 1st column: Foam face ID
// 2rd column: Foam node ID for quad vertex 1
// 3th column: Foam node ID for quad vertex 2
// 4th column: Foam node ID for quad vertex 3
// 5th column: Foam node ID for quad vertex 4
//=============================================
50 8 0 4 404
51 9 8 404 405
52 10 9 405 406
53 11 10 406 407
54 12 11 407 408
55 13 12 408 409
...
```

205.3.5 Simulation

205.3.5.1 Simulation, Solvers: Sequential Solvers

```
define solver sequential <profilespd|umfpack >;
```

Available sequential solvers are:

- `ProfileSPD` is used for symmetric matrices
- `UMFPack` is used for non-symmetric matrices and indefinite matrices

NOTE: USE THE SAME SOLVER FOR EACH ANALYSIS, FOR ALL LOAD STAGES!

Use the same solver, parallel or sequential, with same solver options, for all loading stages in an analysis. System matrices, mass, damping and stiffness and packaged in a different way for different solvers, and different solver options, so changing solver type between different stages will affect access to those matrices, and will affect results...

205.3.5.2 Simulation, Solvers: Parallel Solvers

```
define solver parallel petsc <petsc_options> ;
```

NOTE: USE THE SAME SOLVER FOR EACH ANALYSIS, FOR ALL LOAD STAGES!

Use the same solver, parallel or sequential, with same solver options, for all loading stages in an analysis. System matrices, mass, damping and stiffness and packaged in a different way for different solvers, and different solver options, so changing solver type between different stages will affect access to those matrices, and will affect results...

Direct Solvers

Command Example for a direct solver:

```
define solver parallel petsc "-ksp_type preonly -pc_type lu" ;
define solver parallel petsc "-pc_type lu -pc_factor_mat_solver_package mumps" ;
define solver parallel petsc "-pc_type lu -pc_factor_mat_solver_package ↩
    superlu" ;
```

As shown in the Command Example, "-ksp_type" represents the solver type, "-pc_type" represents the preconditioner types. By defining "preonly", petsc will use the direct solver, and its type is defined in the preconditioner types. In addition, "lu" represents LU factorization in the direct solver. The solver package "mumps" is designed for finite-element methods, and can interleave Gauss elimination process with the assembly process of global stiffness from the local element stiffness matrices. It is also noted that "mumps" can solve symmetric indefinite matrices.

The solver package "superlu" pivots the large-scale sparse matrices to numerous small-scale dense matrices for acceleration.

Iterative Solvers

Command Example for an iterative solver:

```
define solver parallel petsc "-ksp_type gmres -pc_type jacobi";
define solver parallel petsc "-ksp_type cg -pc_type ilu";
```

PETSc contains many iterative solvers and preconditioner for large-scale problems, and they are all available in with Real-ESSI.

Tables 205.1 and 205.2 on next pages present a full set of options for iterative solvers and preconditioners.

Table 205.1: Available Parallel Iterative Solvers

Solver Name	Method
"richardson"	Richardson
"chebyshev"	Chebyshev
"cg"	Conjugate Gradient
"bicg"	BiConjugate Gradient
"gmres"	Generalized Minimal Residual
"fgmres"	Flexible Generalized Minimal Residual
"dgmres"	Deflated Generalized Minimal Residual
"gcr"	Generalized Conjugate Residual
"bcgs"	BiCGSTAB
"cgs"	Conjugate Gradient Squared
"tfqmr"	Transpose-Free Quasi-Minimal Residual (1)
"tcqmr"	Transpose-Free Quasi-Minimal Residual (2)
"cr"	Conjugate Residual
"lsqr"	Least Squares Method

Table 205.2: Available Parallel Iterative Preconditioners

Preconditioner Name	Method
jacobi	Jacobi
bjacobi	Block Jacobi
sor	SOR (and SSOR)
eisenstat	SOR with Eisenstat trick
icc	Incomplete Cholesky
ilu	Incomplete LU
asm	Additive Schwarz
gasm	Generalized Additive Schwarz
gamg	Algebraic Multigrid
bddc	Balancing Domain Decomposition by Constraints
ksp	Linear solver
composite	Combination of preconditioners

For more available solver options please consult the PETSc documentation (`http://www.mcs.anl.gov/petsc/petsc-current/docs/manual.pdf`).

205.3.5.3 Simulation: Static Solution Advancement

```
simulate <.> steps using static algorithm;
```

205.3.5.4 Simulation: Dynamic Solution Advancement with the Constant Time Step

```
simulate <.> steps using transient algorithm time_step = <T>;
```

205.3.5.5 Simulation: Dynamic Solution Advancement with Variable Time Step

```
simulate <.> steps using variable transient algorithm
  time_step = <T>
  minimum_time_step = <.>
  maximum_time_step = <.>
  number_of_iterations = <.>
```

205.3.5.6 Simulation: Generalized Eigenvalue Analysis

At any given point in an analysis a generalized eigenvalue analysis of the system can be performed, based on the current mass and tangent stiffness matrices. The command to do this is:

```
simulate using eigen algorithm number_of_modes = <.>;
```

The first `number_of_modes` eigenvalues are displayed on screen after the analysis is performed. If more eigenvalues are requested than degrees-of-freedom the system has, the excess reported values are set to NaN (not a number).

Description of output for nodes of different dof types can be found in section 206.5.6.

205.3.5.7 Simulation: Displacement Control

```
define static integrator displacement_control using node # <.> dof DOFTYPE ↩
   increment <L>;
```

205.3.5.8 Simulation: Load, Control, Factor Increment

```
define load factor increment <.>;
```

205.3.5.9 Simulation: Dynamic Integrator, Newmark Method

```
define dynamic integrator Newmark with gamma = <.> beta = <.>;
```

See also the list of the reserved keywords from Section 205.7 on page 1163.

205.3.5.10 Simulation: Dynamic Integrator, Hilber Hughes Taylor, HHT, α Method

```
define dynamic integrator Hilber_Hughes_Taylor with alpha = <.>;
```

See also the list of the reserved keywords from Section 205.7 on page 1163.

205.3.5.11 Simulation: Absolute Convergence Criteria

```
define convergence test
  < Absolute_Norm_Unbalanced_Force | Absolute_Norm_Displacement_Increment >
  tolerance = <.>
  maximum_iterations = <.>;
```

This command sets the convergence criteria for global iterative solvers. If the system-of-equation to be solved is

$$K_T \Delta U = \Delta R$$

where K_T is the current tangent stiffness operator/matrix, ΔU is the displacement increment, and ΔR is the residual. The convergence criteria is based on:

- The l^2 norm of the displacement increment: $\|\Delta U\|_2 < TOL$.
- The l^2 norm of the unbalanced force: $\|\Delta R\|_2 < TOL$.

The convergence test should be defined before the algorithms.

205.3.5.12 Simulation: Average Convergence Criteria

```
define convergence test ↩
    <Average_Norm_Unbalanced_Force|Average_Norm_Displacement_Increment>
tolerance = <.>
maximum_iterations = <.> ;
```

This command sets the convergence criteria for global iterative solvers. If the system-of-equation to be solved is

$$K_T \Delta U = \Delta R$$

Where K_T is the current tangent stiffness operator (dynamic tangent for dynamic analysis), ΔU is the displacement increment, and ΔR is the residual. The convergence criteria can be based off

- The average l^2 norm of the displacement increment: $\|\Delta U\|_2/\sqrt{N} < TOL$.
- The average l^2 norm of the unbalanced force: $\|\Delta R\|_2/\sqrt{N} < TOL$.

where N is the number of DOFs in the system-of-equations.

The convergence test should be defined before the algorithms.

205.3.5.13 Simulation: Relative Convergence Criteria

```
define convergence test ↩
    <Relative_Norm_Unbalanced_Force|Relative_Norm_Displacement_Increment>
tolerance = <.>
minimum_absolute_tolerance = <.>
maximum_iterations = <.> ;
```

This command sets the convergence criteria for global iterative solvers. If the system-of-equation to be solved is

$$K_T \Delta U = \Delta R$$

Where K_T is the current tangent stiffness operator (dynamic tangent for dynamic analysis), ΔU is the displacement increment, and ΔR is the residual. The convergence criteria can be based on

- The relative l^2 norm of the displacement increment: $\|\Delta U\|_2 / \|U_0\|_2 < TOL$ or $\|\Delta U\|_2 < MIN\_ABS\_TOL$.
- The relative l^2 norm of the unbalanced force: $\|\Delta R\|_2 / \|R_0\|_2 < TOL$ or $\|\Delta R\|_2 < MIN\_ABS\_TOL$.

Where,
R_0 is the external force in the beginning
U_0 is the solution after the first iteration.
Since U_0 is zero before the first iteration, the relative norm of the displacement increment in the first iteration would be equal to 1.

The convergence test should be defined before the algorithms.

205.3.5.14 Simulation: Solution Algorithms

```
define algorithm < With_no_convergence_check | linear_elastic | Newton | ↩
    Modified_Newton | Newton_With_LineSearch >;
```

```
define algorithm < Newton_With_Subincrement >
  using minimum_time_step = <.> ;
```

If the current specified load factor $\Delta\lambda$ (for static) or time step Δt (for dynamic) fails to achieve the convergence in the specified maximum number of iterations, the algorithm `Newton_With_Subincrement` will subdivide the current step into two sub steps of load increment $\Delta\lambda^{new} = \Delta\lambda/2$ (for static) or time step $\Delta t^{new} = \Delta t/2$ (for dynamic).

- `minimum_time_step` specifies the allowed minimum load factor $\Delta\lambda$ (for static) or time step Δt (for dynamic), that the algorithm should sub-divide to achieve convergence. If the subdivided step size becomes less than the `minimum_time_step`, the algorithm returns failure to convergence.

Note: If any Newton algorithm is used, the convergence test should be defined before the algorithms.

205.3.5.15 Simulation: Constitutive Integration Algorithm

Starting with version 03-NOV-2015, NDMaterial class of materials require explicit specification of the constitutive integration algorithm. This is done with the command:

```
define NDMaterial constitutive integration algorithm Forward_Euler;
```

```
define NDMaterial constitutive integration algorithm Forward_Euler_Subincrement
  number_of_subincrements = <.> ;
```

```
define NDMaterial constitutive integration algorithm Backward_Euler
  yield_function_relative_tolerance = <.>
  stress_relative_tolerance = <.>
  maximum_iterations = <.>;
```

```
define NDMaterial constitutive integration algorithm Backward_Euler_Subincrement
  yield_function_relative_tolerance = <.>
  stress_relative_tolerance = <.>
  maximum_iterations = <.>
  allowed_subincrement_strain = <.> ;
```

The command specifies the method, tolerances and maximum number of iterations used to do material point integrations. The parameters are:

- `number_of_subincrements` Specify the number of subincrements in forward Euler subincrement algorithm.
- `yield_function_relative_tolerance` Specify the relative tolerance of the yield surface value in the family of backward Euler algorithm.
- `stress_relative_tolerance` Specify the relative stress tolerance in the family of backward Euler algorithm. The stress increment is within this tolerance for each step unless the integration fails. Frobenius norm is used to calculate the stress norm.
- `maximum_iterations` Specify the maximum number of iterations in backward Euler algorithm.
- `allowed_subincrement_strain` defines the maximum value of allowed strain increment in backward Euler subincrement method. If one of strain component increments is greater than the user-defined allowed strain increment, strain increment will be divided into subincrements based on the allowed subincrement. For example, if the `strain_increment` is 0.05, and the `allowed_subincrement_strain` is 0.01. The number of subincrements will be $0.05/0.01 = 5$. A small allowed subincrement leads to more accurate results, however, it takes more time. For the simple nonlinear materials, like

von Mises linear hardening, the allowed subincrement can be as big as 5 percent. For the complicated nonlinear materials, like hyperbolic Drucker-Prager Armstrong-Frederick hardening material, the allowed subincrement should be much smaller in the range of 1E-4.

205.3.5.16 Simulation: Status Check

All simulate commands set the variable `SIMULATE_EXIT_FLAG` automatically upon exit. This flag can be used to check whether the simulation concluded normally (`SIMULATE_EXIT_FLAG` = 0), failed (`SIMULATE_EXIT_FLAG < 0`), or finished with warnings (`SIMULATE_EXIT_FLAG > 0`).

For example, the following simulations will fail.

```
atmospheric_pressure = 101325*Pa;

pstart = 3000*kPa;

//SAniSand 2004 calibration for Toyoura Sand.
add material # 1 type sanisand2004
    mass_density = 2100.0 * kg / m^3
    e0 = 0.735
    sanisand2004_G0 = 125. poisson_ratio = 0.05
    sanisand2004_Pat = atmospheric_pressure
    sanisand2004_p_cut = 0.1*atmospheric_pressure
    sanisand2004_Mc = 1.25 sanisand2004_c = 0.712
    sanisand2004_lambda_c = 0.019 sanisand2004_xi = 0.7
    sanisand2004_ec_ref = 0.934 sanisand2004_m = 0.01
    sanisand2004_h0 = 7.05 sanisand2004_ch = 0.968
    sanisand2004_nb = 1.1 sanisand2004_A0 = 0.704
    sanisand2004_nd = 3.5 sanisand2004_z_max = 4.
    sanisand2004_cz = 600.
    initial_confining_stress = 1*Pa ;

simulate constitutive testing DIRECT_STRAIN
    use material # 1
    scale_factor = 1.
    series_file = "increments.txt"
    sigma0 = ( -pstart*kPa , -pstart*kPa , -pstart*kPa , 0*Pa , 0*Pa , 0*Pa )
    verbose_output = 1;


if(SIMULATE_EXIT_FLAG == 0)
{
    print "All Good!";
}
else
{
    print "Something went wrong. Error code = ";
    print SIMULATE_EXIT_FLAG;
}

bye;
```

The above simulation fails because the integration method for the constitutive model is not set (see 205.3.5.15). Therefore, the second branch of the 'if' statement will execute.

205.3.5.17 Simulation: Save State

Save ESSI system state to a file, to prepare for a restart.

```
 save model;
```

This command will save the state of a model, an ESSI system, in file. Filename for the save file will be created from a model name, loading stage name, and current loading time.

For example, for a model that contains the following commands in the input file:

```
model name "ESSI_model";
. . .
new loading stage "Loading_stage_2";
. . .
simulate 100 steps using transient algorithm time_step = 0.005*s;

save model;
```

will be saved in file with the following name:

```
ESSI_model_Loading_stage_2_at_time_0.5second_RESTART.essi
```

since there were 100 steps with $\Delta t = 0.005$s, and that advances the solution to $t = 0.5$s.

205.3.5.18 Simulation: Restart Simulation

Restart simulation after stage or a step, from a saved ESSI system model file.

```
restart model using file "filename";
```

Here, ESSI system model is saved in a file `filename`, see command for saving model on page 1120.

For example, to restart simulation from a saved file described above, restart will be initiated in a new input file by using the following command:

```
restart model using file ←↩
   "ESSI_model_Loading_stage_2_at_time_0.5second_RESTART.essi";
```

All the results from previous loading stages will be saved in the restart file. From here on, analyst can start new loading stages, etc.

205.3.5.19 Simulation: Return Value for `simulate` Command

Simulate command, `simulate`, returns status of simulation progress. For each successful step, `simulate` returns value 0 while for a failed step it returns -1. This is useful as analyst can control solution process, and change algorithm if predefined algorithm fails to converge.

For example the example if listing 205.8, simulation part of a larger examples, will perform a change of stepping algorithm from the load control to displacement control upon failure of load control to converge.

```
step=0;
Nsteps = 100;
define load factor increment 0.01; // Start with load-control

simulation_status=simulate 1 steps using static algorithm;

while (step<(Nsteps-1))
  {
    if(simulation_status>=0) // Converged, continue using load-control
      {
        simulation_status=simulate 1 steps using static algorithm;
      }
    else // Not converged, so change to displacement-control
     {
       define static integrator displacement_control using node # 1 dof ux ↩
           increment 1E-3*m;
       simulate 1 steps using static algorithm;
     }
    step=step+1;
  }
```

Figure 205.8: Interactive simulation control using feedback (return value) from the `simulate` command.

It should be noted that the idea for interactive control of simulation process comes from FEAP (Zienkiewicz and Taylor, 1991b) and later from OpenSEES (Mazzoni et al., 2002) where it was implemented with early extension of OpenSees command language with using Tcl in early 2000s.

An example of the above feedback mechanism is provided below

```
model name "vm";
add material # 1 type vonMises
    mass_density = 0.0*kg/m^3
    elastic_modulus = 2E7*N/m^2
    poisson_ratio = 0.0
    von_mises_radius = 1E5*Pa
    kinematic_hardening_rate = 0*Pa
    isotropic_hardening_rate = 0*Pa;
// define the node:
add node # 1 at (0*m,0*m,1*m) with 3 dofs;
add node # 2 at (1*m,0*m,1*m) with 3 dofs;
add node # 3 at (1*m,1*m,1*m) with 3 dofs;
add node # 4 at (0*m,1*m,1*m) with 3 dofs;
add node # 5 at (0*m,0*m,0*m) with 3 dofs;
add node # 6 at (1*m,0*m,0*m) with 3 dofs;
add node # 7 at (1*m,1*m,0*m) with 3 dofs;
add node # 8 at (0*m,1*m,0*m) with 3 dofs;
// Define the element.
add element # 1 type 8NodeBrick using 2 Gauss points each direction with nodes ↩
    (1, 2, 3, 4, 5, 6, 7, 8) use material # 1;

new loading stage "shearing";
//fix the bottom totally
fix node # 5 dofs all;
fix node # 6 dofs all;
fix node # 7 dofs all;
fix node # 8 dofs all;
// Fix the other 2 directions on the top.
fix node # 1 dofs uy uz ;
fix node # 2 dofs uy uz ;
fix node # 3 dofs uy uz ;
fix node # 4 dofs uy uz ;
add load # 101 to node # 1 type linear Fx = 40 * kN;
add load # 102 to node # 2 type linear Fx = 40 * kN;
add load # 103 to node # 3 type linear Fx = 40 * kN;
add load # 104 to node # 4 type linear Fx = 40 * kN;
define solver UMFPack;
//define algorithm With_no_convergence_check ↩
    ;Norm_Displacement_Increment;Norm_Unbalance
define convergence test Absolute_Norm_Displacement_Increment
    tolerance = 1E-3
    maximum_iterations = 5
    ;
define algorithm Newton;
define NDMaterial constitutive integration algorithm Backward_Euler
    yield_function_relative_tolerance = 1E-7
    stress_relative_tolerance = 1E-7
    maximum_iterations = 100;

// ****************************************************************************
step=0;
```

```
Nsteps = 10;
define load factor increment 1/Nsteps; // Start with load-control
// ***************************************************************************
// Simulate with status check:
// ***************************************************************************
mystatus=simulate 1 steps using static algorithm;
while (step<(Nsteps-1)){
    step=step+1;
    if(mystatus>=0){ // Converged, so continue using load-control
        mystatus=simulate 1 steps using static algorithm;
    }
    else{ // Not converged, so change to displacement-control
        define static integrator displacement_control using node # 1 dof ux ↩
            increment 1E-3*m ;
        simulate 1 steps using static algorithm;
    }
}

bye;
```

Resulting terminal output, showing a switch between two solution control mechanisms is provided below:

```



  The Finite Element Interpreter

  MS ESSI
  Earthquake Soil Structure Interaction Simulator

  Sequential processing mode.

Version Name : Academic Version. Release date: Apr 14 2017 at 17:19:55. Tag: ↩
    c24e557a56
Version Branch : yuan
Compile Date : Apr 15 2017 at 20:28:11
Compile User : yuan
Compile Sysinfo: cml01 4.4.0-72-generic x86_64 GNU/Linux
Runtime User : Runtime Sysinfo: Time Now : Apr 15 2017 at 21:15:40

Static startup tips:
 * Remember: Every command ends with a semicolon ';'.
 * Type 'quit;' or 'exit;' to finish.
 * Run 'essi -h' to see available command line options.

Including: "main.fei"


Model name is being set to "vm"

```

```


Starting new stage: shearing

changing previous_stage_name from to shearing
Setting set_constitutive_integration_method = 2

Starting sequential static multi-step analysis
===============================================================================================
Creating analysis ↩
   model........................................................................Pass!
Checking constraint ↩
   handler....................................................................Pass!
Checking ↩
   numberer..............................................................................Pass!
Checking analysis ↩
   algorithm....................................................................Pass!
Checking system of equation ↩
   handler............................................................Pass!
Checking static integration ↩
   handler............................................................Pass!


 Writing Initial Conditions and (0) - Outputting mesh.

Static Analysis: [ 1/1 ]
  [iteration 1 /5 ] Convergence Test: Absolute Norm Displacement ↩
     Increment::(tol: 0.001)
                   Absolute Norm deltaF: 1.6396e-12
                   Absolute Norm deltaU: 0.0032
                   Average Norm deltaF: 8.1981e-13
                   Average Norm deltaU: 0.0016
                   Relative Norm deltaF: 2.0495e-16
                   Relative Norm deltaU: 0.0032
  [iteration 2 /5 ] Convergence Test: Absolute Norm Displacement ↩
     Increment::(tol: 0.001)
                   Absolute Norm deltaF: 4.5475e-13
                   Absolute Norm deltaU: 5.2683e-19
                   Average Norm deltaF: 2.2737e-13
                   Average Norm deltaU: 2.6341e-19
                   Relative Norm deltaF: 5.6843e-17
                   Relative Norm deltaU: 1.6463e-16

> Analysis End ↩
   -------------------------------------------------------------------------------

Starting sequential static multi-step analysis
===============================================================================================
Creating analysis ↩
   model........................................................................Pass!
Checking constraint ↩
```

```
    handler....................................................................Pass!
Checking ↩
    numberer..............................................................................Pass!
Checking analysis ↩
    algorithm....................................................................Pass!
Checking system of equation ↩
    handler............................................................Pass!
Checking static integration ↩
    handler............................................................Pass!

Static Analysis: [1 /1 ]
  [iteration 1 /5 ] Convergence Test: Absolute Norm Displacement ↩
      Increment::(tol: 0.001)
                   Absolute Norm deltaF: 2.7285e-12
                   Absolute Norm deltaU: 0.0032
                   Average Norm deltaF: 1.3642e-12
                   Average Norm deltaU: 0.0016
                   Relative Norm deltaF: 3.4106e-16
                   Relative Norm deltaU: 0.0032
  [iteration 2 /5 ] Convergence Test: Absolute Norm Displacement ↩
      Increment::(tol: 0.001)
                   Absolute Norm deltaF: 3.5225e-12
                   Absolute Norm deltaU: 7.8429e-19
                   Average Norm deltaF: 1.7612e-12
                   Average Norm deltaU: 3.9214e-19
                   Relative Norm deltaF: 4.4031e-16
                   Relative Norm deltaU: 2.4509e-16

> Analysis End ↩
    -------------------------------------------------------------------------------

Starting sequential static multi-step analysis
===============================================================================================
Creating analysis ↩
    model........................................................................Pass!
Checking constraint ↩
    handler....................................................................Pass!
Checking ↩
    numberer..............................................................................Pass!
Checking analysis ↩
    algorithm....................................................................Pass!
Checking system of equation ↩
    handler............................................................Pass!
Checking static integration ↩
    handler............................................................Pass!

Static Analysis: [1 /1 ]
  [iteration 1 /5 ] Convergence Test: Absolute Norm Displacement ↩
      Increment::(tol: 0.001)
                   Absolute Norm deltaF: 1.819e-12
                   Absolute Norm deltaU: 0.0032
```

```
                   Average Norm deltaF: 9.0949e-13
                   Average Norm deltaU: 0.0016
                   Relative Norm deltaF: 2.2737e-16
                   Relative Norm deltaU: 0.0032
  [iteration 2 /5 ] Convergence Test: Absolute Norm Displacement Increment::(tol: 0.001)
                   Absolute Norm deltaF: 2.5724e-12
                   Absolute Norm deltaU: 5.0807e-19
                   Average Norm deltaF: 1.2862e-12
                   Average Norm deltaU: 2.5403e-19
                   Relative Norm deltaF: 3.2155e-16
                   Relative Norm deltaU: 1.5877e-16

> Analysis End -------------------------------------------------------------------------------

Starting sequential static multi-step analysis
===============================================================================================
Creating analysis model........................................................................Pass!
Checking constraint handler....................................................................Pass!
Checking numberer..............................................................................Pass!
Checking analysis algorithm....................................................................Pass!
Checking system of equation handler............................................................Pass!
Checking static integration handler............................................................Pass!

Static Analysis: [1 /1 ]
  [iteration 1 /5 ] Convergence Test: Absolute Norm Displacement Increment::(tol: 0.001)
                   Absolute Norm deltaF: 3132.5
                   Absolute Norm deltaU: 0.0032
                   Average Norm deltaF: 1566.2
                   Average Norm deltaU: 0.0016
                   Relative Norm deltaF: 0.39156
                   Relative Norm deltaU: 0.0032
  [iteration 2 /5 ] Convergence Test: Absolute Norm Displacement Increment::(tol: 0.001)
                   Absolute Norm deltaF: 3132.5
                   Absolute Norm deltaU: 0.001253
                   Average Norm deltaF: 1566.2
                   Average Norm deltaU: 0.0006265
                   Relative Norm deltaF: 0.39156
                   Relative Norm deltaU: 0.39156
  [iteration 3 /5 ] Convergence Test: Absolute Norm Displacement Increment::(tol: 0.001)
                   Absolute Norm deltaF: 3132.5
```

```
                   Absolute Norm deltaU: 0.001253
                   Average Norm deltaF: 1566.2
                   Average Norm deltaU: 0.0006265
                   Relative Norm deltaF: 0.39156
                   Relative Norm deltaU: 0.39156
  [iteration 4 /5 ] Convergence Test: Absolute Norm Displacement ↩
     Increment::(tol: 0.001)
                   Absolute Norm deltaF: 3132.5
                   Absolute Norm deltaU: 0.001253
                   Average Norm deltaF: 1566.2
                   Average Norm deltaU: 0.0006265
                   Relative Norm deltaF: 0.39156
                   Relative Norm deltaU: 0.39156
  [iteration 5 /5 ] Convergence Test: Absolute Norm Displacement ↩
     Increment::(tol: 0.001)
                   Absolute Norm deltaF: 3132.5
                   Absolute Norm deltaU: 0.001253
                   Average Norm deltaF: 1566.2
                   Average Norm deltaU: 0.0006265
                   Relative Norm deltaF: 0.39156
                   Relative Norm deltaU: 0.39156
  [iteration 5/5 ] Convergence Test: Absolute Norm Displacement ↩
     Increment::(tol: 0.001) !!!FAILED TO CONVERGE!!! [EXITING..]
                   Absolute Norm deltaF: 3132.5
                   Absolute Norm deltaU: 0.001253
                   Average Norm deltaF: 1566.2
                   Average Norm deltaU: 0.0006265
                   Relative Norm deltaF: 0.39156
                   Relative Norm deltaU: 0.39156
NewtonRaphson::solveCurrentStep() -the ConvergenceTest object failed in test()

Static Analysis: [1 /1 ] The Algorithm failed at load factor 0.4
> Analysis End ↩
   -------------------------------------------------------------------------------

Starting sequential static multistep analysis
===============================================================================================
Creating analysis ↩
   model........................................................................Pass!
Checking constraint ↩
   handler....................................................................Pass!
Checking ↩
   numberer..............................................................................Pass!
Checking analysis ↩
   algorithm....................................................................Pass!
Checking system of equation ↩
   handler............................................................Pass!
Checking static integration ↩
   handler............................................................Pass!

Static Analysis: [1 /1 ]
```

```
  [iteration 1 /5 ] Convergence Test: Absolute Norm Displacement ↩
      Increment::(tol: 0.001)
                  Absolute Norm deltaF: 9310.6
                  Absolute Norm deltaU: 2.6478e-19
                  Average Norm deltaF: 4655.3
                  Average Norm deltaU: 1.3239e-19
                  Relative Norm deltaF: 0.42857
                  Relative Norm deltaU: 2.6478e-19

> Analysis End ↩
    -------------------------------------------------------------------------------

Starting sequential static multistep analysis
===============================================================================================
Creating analysis ↩
    model........................................................................Pass!
Checking constraint ↩
    handler....................................................................Pass!
Checking ↩
    numberer..............................................................................Pass!
Checking analysis ↩
    algorithm....................................................................Pass!
Checking system of equation ↩
    handler............................................................Pass!
Checking static integration ↩
    handler............................................................Pass!

Static Analysis: [1 /1 ]
  [iteration 1 /5 ] Convergence Test: Absolute Norm Displacement ↩
      Increment::(tol: 0.001)
                  Absolute Norm deltaF: 1437.5
                  Absolute Norm deltaU: 1.0012e-18
                  Average Norm deltaF: 718.76
                  Average Norm deltaU: 5.006e-19
                  Relative Norm deltaF: 0.10425
                  Relative Norm deltaU: 1.0012e-18

> Analysis End ↩
    -------------------------------------------------------------------------------

Starting sequential static multistep analysis
===============================================================================================
Creating analysis ↩
    model........................................................................Pass!
Checking constraint ↩
    handler....................................................................Pass!
Checking ↩
    numberer..............................................................................Pass!
Checking analysis ↩
    algorithm....................................................................Pass!
Checking system of equation ↩
```

```
    handler...........................................................Pass!
Checking static integration ←↩
    handler...........................................................Pass!

Static Analysis: [1 /1 ]
  [iteration 1 /5 ] Convergence Test: Absolute Norm Displacement ←↩
      Increment::(tol: 0.001)
                   Absolute Norm deltaF: 112.08
                   Absolute Norm deltaU: 5.1789e-19
                   Average Norm deltaF: 56.038
                   Average Norm deltaU: 2.5895e-19
                   Relative Norm deltaF: 0.017652
                   Relative Norm deltaU: 5.1789e-19

> Analysis End ←↩
    -------------------------------------------------------------------------------

Starting sequential static multistep analysis
=============================================================================================
Creating analysis ←↩
    model.....................................................................Pass!
Checking constraint ←↩
    handler...................................................................Pass!
Checking ←↩
    numberer..............................................................................Pass!
Checking analysis ←↩
    algorithm.................................................................Pass!
Checking system of equation ←↩
    handler...........................................................Pass!
Checking static integration ←↩
    handler...........................................................Pass!

Static Analysis: [1 /1 ]
  [iteration 1 /5 ] Convergence Test: Absolute Norm Displacement ←↩
      Increment::(tol: 0.001)
                   Absolute Norm deltaF: 7.142
                   Absolute Norm deltaU: 8.6792e-19
                   Average Norm deltaF: 3.571
                   Average Norm deltaU: 4.3396e-19
                   Relative Norm deltaF: 0.001399
                   Relative Norm deltaU: 8.6792e-19

> Analysis End ←↩
    -------------------------------------------------------------------------------

Starting sequential static multistep analysis
=============================================================================================
Creating analysis ←↩
    model.....................................................................Pass!
Checking constraint ←↩
    handler...................................................................Pass!
```

```
Checking numberer..............................................................................Pass!
Checking analysis algorithm...................................................................Pass!
Checking system of equation handler...................................................Pass!
Checking static integration handler...................................................Pass!

Static Analysis: [1 /1 ]
  [iteration 1 /5 ] Convergence Test: Absolute Norm Displacement Increment::(tol: 0.001)
                  Absolute Norm deltaF: 0.44693
                  Absolute Norm deltaU: 1.0155e-18
                  Average Norm deltaF: 0.22347
                  Average Norm deltaU: 5.0774e-19
                  Relative Norm deltaF: 8.9267e-05
                  Relative Norm deltaU: 1.0155e-18

> Analysis End -------------------------------------------------------------------------------

Starting sequential static multistep analysis
=============================================================================================
Creating analysis model.......................................................................Pass!
Checking constraint handler.................................................................Pass!
Checking numberer..............................................................................Pass!
Checking analysis algorithm...................................................................Pass!
Checking system of equation handler...................................................Pass!
Checking static integration handler...................................................Pass!

Static Analysis: [1 /1 ]
  [iteration 1 /5 ] Convergence Test: Absolute Norm Displacement Increment::(tol: 0.001)
                  Absolute Norm deltaF: 0.027935
                  Absolute Norm deltaU: 2.1658e-19
                  Average Norm deltaF: 0.013968
                  Average Norm deltaU: 1.0829e-19
                  Relative Norm deltaF: 5.5866e-06
                  Relative Norm deltaU: 2.1658e-19

> Analysis End -------------------------------------------------------------------------------
How polite! Bye, have a nice day!
```

205.3.5.20 Simulation: New Elastic Loading Case

For design applications, linear elastic analysis cases are performed and later combined, using factors of safety (see section 205.3.5.21 on page 1133) to obtain sectional forces for design.

The command for elastic analysis is:

```
new elastic loading case <string> ;
```

One example is

```
new elastic loading case "case1" ;
```

In a new elastic loading case, all previous loads, load patterns are removed.

To guarantee a fresh start, all commit-displacement at nodes are reset to 0, and all commit-stress/strain at Gauss points are reset to 0.

The following components are kept unchanged in a new elastic loading case:

- material properties.
- mesh connectivity
- boundary conditions.
- acceleration fields.
- damping.

If users want to modify the mesh, a new model is suggested instead of a new elastic loading case.

205.3.5.21 Simulation: Combine Elastic Load Cases

For design applications, elastic load cases, that have been analyzed beforehand, can be superimposed, combined using factors of safety, to obtain internal forces that used for design.

The command for this is

```
combine elastic load cases
  hdf5_filenames_list = <string>
  load_factors_list = <string>
  output_filename = <string>
  ;
```

One example is

```
combine elastic load cases
  hdf5_filenames_list = "test_case1.h5.feioutput test_case2.h5.feioutput"
  load_factors_list = "1.2 1.5"
  output_filename = "combine.h5.feioutput"
  ;
```

- `hdf5_filenames_list` specifies the list of HDF5 output filenames. The list should be separated by either space or comma.
- `load_factors_list` specifies the list of scale factors for each loading case. The list should be separated by either space or comma.
- `output_filename` specifies one output filename of the combined loading cases.

The number of specified files in `hdf5_filenames_list` should be equal to the number of scale factors (factors of safety) in `load_factors_list`.

205.3.5.22 Simulation, Dynamic Solution Advancement for Solid-Fluid Interaction

Dynamic analysis of solid-fluid interaction can be performed using:

```
simulate <.> steps using solid fluid interaction transient algorithm time_step ↩
    = <T>;
```

where:

- `<.>` is an integer specifying total number of time steps in transient solid fluid interaction analysis.
- `time_step = <T>` defines the time step for solid fluid interaction transient analysis.

For example:

```
simulate 300 steps using solid fluid interaction transient algorithm time_step ↩
    = 0.01*s;
```

Performs transient solid fluid interaction analysis for 300 steps with time step 0.01 s.

205.3.5.23 Simulation, Dynamic Solution Advancement for Stochastic Finite Element Method

Dynamic analysis of stochastic finite element modeling can be performed using:

```
simulate <.> steps using stochastic transient algorithm time_step = <T>;
```

where:

- `<.>` is an integer specifying total number of time steps in transient stochastic finite element analysis.
- `time_step = <T>` defines the time step for stochastic finite element transient analysis.

Please note that the stochastic transient algorithm is different from the general transient algorithm in section 205.3.5.4 in the formulation of unbalanced load for each time step. The stochastic transient algorithm uses directly the incremental external loads to compute the incremental displacements, while the general transient algorithm accounts for the correction from resisting forces in the formulation of unbalanced forces.

For deterministic and probabilistic, linear elastic problems, both transient algorithms would produce the same response. Stochastic transient algorithm is more efficient because there is no need to compute resisting forces.

For deterministic, nonlinear inelastic problems, the general transient algorithm is more accurate due to the corrections from resisting forces. The accuracy of stochastic transient algorithm can be improved using smaller loading increments. For probabilistic, nonlinear inelastic problems, the accuracy of the general transient algorithm can only be guaranteed if the number of polynomial chaos terms used in probabilistic constitutive modeling is equal or close enough to the number of polynomial chaos terms in global level. Otherwise, it is recommended to use the stochastic transient algorithm for dynamic analysis of stochastic finite element modeling.

For example:

```
simulate 300 steps using stochastic transient algorithm time_step = 0.01*s;
```

Performs transient stochastic finite element analysis for 300 steps with time step 0.01s.

205.3.5.24 Simulation, Sobol Sensitivity Analysis

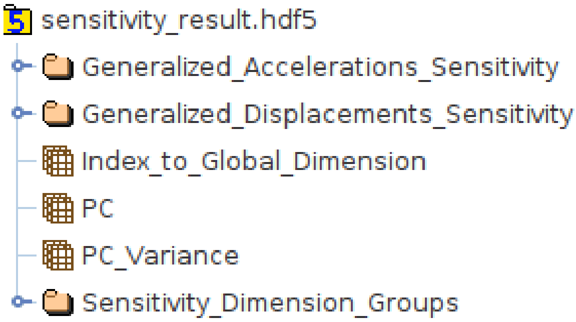
Sobol sensitivity analysis can be performed using:

```
Sobol sensitivity analysis of node # <.> dof DOFTYPE peak response from random field # <.>
pc_coefficient_hdf5 = "pc_coefficient_hdf5_file_name"
output_hdf5 = "output_hdf5_file_name";
```

where:

- `node #` specify the node tag.
- `DOFTYPE` specify the dof to perform sensitivity analysis. It can be either `ux`, `uy` or `uz`.
- `random field #` specify the random field polynomial chaos bases of the stochastic nodal response.
- `pc_coefficient_hdf5` specify the name of a hdf5 file that contains simulation results of polynomial chaos coefficients.
- `output_hdf5` specify the name of the output hdf5 file for sensitivity analysis.

The output hdf5 format for sensitivity analysis is given as below:



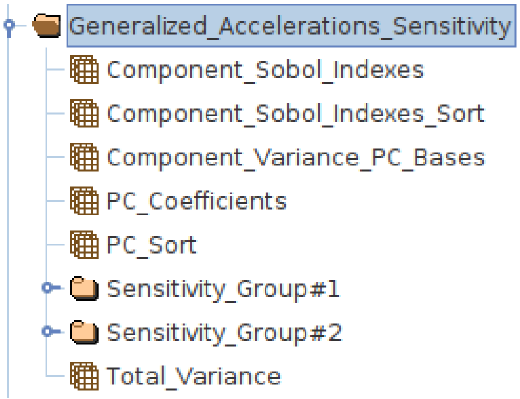
sensitivity_result.hdf5
Generalized_Accelerations_Sensitivity
Generalized_Displacements_Sensitivity
Index_to_Global_Dimension
PC
PC_Variance
Sensitivity_Dimension_Groups

Figure 205.9: Overall data structure of output hdf5 file for sensitivity analysis.

Figure 205.9 shows the overall data structure organization of the output hdf5 file of sensitivity analysis.

- Generalized_Accelerations_Sensitivity data group:

  Contains Sobol sensitivity analysis results for stochastic nodal acceleration response. It contains the following datasets and data groups as shown in Figure 205.10.



Generalized_Accelerations_Sensitivity
- Component_Sobol_Indexes
- Component_Sobol_Indexes_Sort
- Component_Variance_PC_Bases
- PC_Coefficients
- PC_Sort
- Sensitivity_Group#1
- Sensitivity_Group#2
- Total_Variance

Figure 205.10: Datasets and data groups in Generalized_Accelerations_Sensitivity data group.

- Component_Sobol_Indexes dataset:

  Is a column vector containing the computed Sobol Indexes for each component of polynomial chaos (PC) bases.

- Component_Sobol_Indexes_Sort dataset:

  Is a column vector containing the computed Sobol Indexes for each component of polynomial chaos (PC) bases, in descending order.

- Component_Variance_PC_Bases dataset:

  Is a column vector containing the computed variance for each component of polynomial chaos (PC) bases.

- PC_Coefficients dataset:

  Is a column vector containing the polynomial chaos coefficients corresponding to PC bases specified in PC dataset.

- PC_Sort dataset:

  Is a 2D array that describes the sorted PC bases corresponding to Component_Sobol_Indexes_Sort dataset. PC_Sort$_{ij}$ denotes the order of polynomial chaos dimension ξ_j that contributes to the i^{th} sorted, PC basis.

- Total_Variance dataset:

  Is a scalar, variance of nodal stochastic response.

In addition to the above datasets, there will be sub data group(s) containing sensitivity analysis results corresponding to each of the defined sensitivity dimension groups. For example, we have two sub data groups Sensitivity_Group#1 and Sensitivity_Group#2 for this specific hdf5 output for sensitivity analysis. Within these sub data groups, taking Sensitivity_Group#1 as an example, the output data is organized as shown in Figure 205.11.

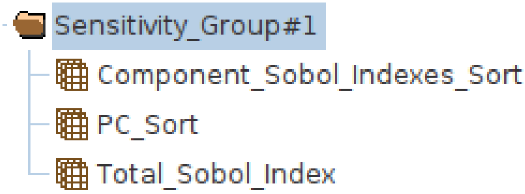


Figure 205.11: Datasets in Sensitivity_Group#1.

It includes the following datasets:

  - Component_Sobol_Indexes_Sort dataset:

    Is a column vector containing the computed Sobol Indexes for part of polynomial chaos (PC) bases specified through the sensitivity dimension group, in descending order.

  - PC_Sort dataset:

    Is a 2D array that describes the sorted, part of PC bases specified through the sensitivity dimension group, corresponding to Component_Sobol_Indexes_Sort dataset. PC_Sort$_{\boldsymbol{ij}}$ denotes the order of polynomial chaos dimension ξ_j that contributes to the i^{th} sorted, PC basis.

  - Total_Sobol_Index dataset:

    Is a scalar, total Sobol sensitivity index for PC bases specified in the sensitivity dimension group.

- Generalized_Displacements_Sensitivity data group:

  Contains Sobol sensitivity analysis results for stochastic nodal displacement response. The configuration of data structure for Generalized_Displacements_Sensitivity data group is the same as Generalized_Accelerations_Sensitivity data group.

- Index_to_Global_Dimension dataset:

  Contains a column vector of integers, which specifies the global dimension IDs for the polynomial chaos bases used to represent the nodal stochastic response.

- PC dataset:

Is a 2D array that describes the multi-dimensional Hermite polynomial chaos (PC) bases for representing the nodal stochastic response. $\boldsymbol{PC_{ij}}$ denotes the order of polynomial chaos dimension ξ_j that contributes to the i^{th} multidimensional Hermite PC basis.

- PC_Variance dataset:

  Is a column vector specifying the variances of Hermite PC basis.

- Sensitivity_Dimension_Groups data group:

  Contains the information about the sensitivity dimension groups defined for sensitivity analysis. Each sensitivity dimension group would be defined by a dataset within the data group as shown in Figure 205.12. Each dataset contains a column vector of integers, specifying the dimension IDs in specific sensitivity dimension group.

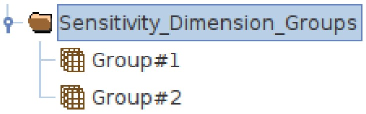


Figure 205.12: Datasets in Sensitivity_Dimension_Groups data group.

205.3.5.25 Simulation: 3D 3C Wave Field Inversion

```
generate domain reduction method motion file from 3D_3C_wave_field_inversion
    target_nodes_filename = <string>
    target_motions_filename = <string>
    time_step = <T>
    hdf5_file = <string>;
```

Example:

```
generate domain reduction method motion file from 3D_3C_wave_field_inversion
    target_nodes_filename = "Target_Nodes.txt"
    target_motions_filename = "Target_Motions.txt"
    time_step = 0.01*s
    hdf5_file = "DRM_Input_from_Inverse_Motion.hdf5";
```

where:

- `target_nodes_filename` is the file name for a file that contains the list of nodes, represented by their tags, where target motions are designated.

  One example of target nodes file is given below.

  ```
  5
  15
  25
  35
  46
  48
  50
  52
  ```

- `target_motions_filename` is the file name for a file that contains the list of target motions at the corresponding target nodes. Note that the length, or number of rows, and the ordering of the target motions file must be the same as those of the target nodes file.

  One example of target motions file is given below.

  ```
  Displacement_1000steps_with_CT.txt
  Displacement_1000steps_with_CT.txt
  Displacement_1000steps_with_CT.txt
  Displacement_1000steps_with_CT.txt
  Zero_Displacement_1000steps_with_CT.txt
  Zero_Displacement_1000steps_with_CT.txt
  Zero_Displacement_1000steps_with_CT.txt
  Zero_Displacement_1000steps_with_CT.txt
  ```

  Time series of target motion is defined in each of the file listed in `target_motions_filename`.

- `time_step` is the time step/interval corresponding to the target motions.
- `hdf5_file` specifies the HDF5 file which contains the geometric information about the DRM elements and DRM nodes. This is also the file in which the DRM forces obtained from 3D wave field inversion are stored. Then this file can be used in future simulations as the DRM input. See previous DRM-related DSLs for the format of this file.

205.3.6 Output Options

Real-ESSI Simulator outputs total displacements at all the nodes, as well total stress, total strain and total plastic strain at all the Gauss points of the element in each time step of each stage of loading. Real-ESSI also outputs any/all other element output in addition to the integration/Gauss point output. Generally, 3-D elements have only integration/Gauss point outputs and structure elements have only element output. The output options are reset to the default options in the beginning of each loading stage. More information about output organization is given in section 206.2.

205.3.6.1 Output Options: Enable/Disable Output

This option is used to enable or disable the outputting of results from all nodes and elements to HDF5 (.feioutput) output file.

Note:- By default output is always enabled for each loading stage.

Command to disable output is

```
disable all output;
```

Command to enable output is

```
enable all output;
```

205.3.6.2 Output Options: Enable/Disable Element Output

This option is used to enable or disable the outputting of element results from all elements to HDF5 (.feioutput) output file, per stage of loading.

Note:- By default all results from elements are output for each loading stage, so this option can be used to enable or disable output per loading stage.

Command to disable element output is

```
disable element output;
```

Command to enable element output is

```
enable element output;
```

205.3.6.3 Output Options: Enable/Disable Displacement Output

This option is used to enable or disable the displacement output at nodes to HDF5 (.feioutput) file.

Note:- By default displacement output is enabled.

Command to disable displacement output is

```
disable displacement output;
```

Command to enable displacement output is

```
enable displacement output;
```

205.3.6.4 Output Options: Enable/Disable Acceleration Output

This option is used to enable or disable the acceleration output at nodes to HDF5 (.feioutput) file.

Note:- By default acceleration output is disabled.

Command to disable acceleration output is

```
disable acceleration output;
```

Command to enable acceleration output is

```
enable acceleration output;
```

205.3.6.5 Output Options: Enable/Disable Asynchronous Output

This option is used to enable or disable the asynchronous method of writing output to HDF5 (.feioutput) file.

Note:- By default asynchronous output is disabled. Asynchronous output is an advanced output feature. Asynchronous output is suitable for I/O-bound simulation.

Command to disable asynchronous output is

```
disable asynchronous output;
```

Command to enable asynchronous output is

```
enable asynchronous output;
```

205.3.6.6 Output Options: Output Every n Steps

This option is used to output results at intervals of n time steps.

Note:- By default results are output for every time step.

Command to enable output only at n^{th} time step interval

```
output every <.> steps;
```

For example: To output only at interval of two time steps for a simulation of 100 steps. One can write

```
output every 2 steps;
```

This will only output for steps 2,4,6,... until 100th step.

205.3.6.7 Output Options: Output Support Reactions

This option is used to output reactions at constrained supports.

Note:- By default output reactions at constrained supports are disabled.

Command to enable reactions for support is

```
output support reactions;
```

205.4 Checking the Model

Real-ESSI provides model check capability:

```
check model ;
```

This command will cycle over all the domain components, including Nodes, Elements, Loads, Constraints, etc. and execute the `checkModel()` function for each. Each domain component writes/reports to the terminal and to the *essi.log* file, if an error is found. For example, bricks will report when the computed Jacobian is negative and other similar errors. Nodes that are not connected will be reported as well. If the diagnostic log is empty, it means that the mesh has passed all tests. Additionally, an output HDF5 file is produced that can be used to display the mesh and do further visual inspections of the model. This file will have initial conditions as outputs for element sand nodes.

Command `check model;` represents a dry run through the model that is used to check the model before a full analysis. Model check is highly recommended before initial stages of a full analysis are executed.

205.5 Constitutive Testing

Material models can be tested using constitutive drivers which exercise single material models. RealE-SSI implements two such drivers.

1. Bardet Driver. Bardet-type constraints can be used to simulate conditions such as drained or undrained triaxial testing with strain or stress control or direct shear testing with shear control.
2. Direct Strain Driver. This driver applies a given strain history (specified by the user) to a material model.

Both these drivers produce identical output: the files `Stress.feioutput` and `Strain.feioutput` which contain stress and strain tensor components at each step. Additionally, the drivers may print out material internal information to the file `Material_Output.feioutput`. For the stress and strain files, each line of these files contain the stresses and strains organized in the following manner:

`Stress.feioutput` → σ_{11} σ_{22} σ_{33} σ_{12} σ_{13} σ_{23}.

`Strain.feioutput` → ϵ_{11} ϵ_{22} ϵ_{33} ϵ_{12} ϵ_{13} ϵ_{23}.

The Bardet driver has the following format.

```
simulate constitutive testing BARDETMETHOD use material # <.>
  scale_factor = <.>
  series_file = <string>
  sigma0 = ( <F/L^2> , <F/L^2> , <F/L^2> , <F/L^2> , <F/L^2> , <F/L^2> )
  verbose_output = <.>
```

Where,

- `BARDETMETHOD` can have any one of the following values:
  - `CONSTANT_P_TRIAXIAL_LOADING_STRAIN_CONTROL`: Triaxial loading with p kept constant. In this case the input file is interpreted as strain increments in the ϵ_{11} component.
  - `DRAINED_TRIAXIAL_LOADING_STRESS_CONTROL`: Drained Triaxial loading. In this case the input file is interpreted as stress increments in the σ_{11} component.
  - `DRAINED_TRIAXIAL_LOADING_STRAIN_CONTROL`: Drained Triaxial loading. In this case the input file is interpreted as strain increments in the ϵ_{11} component.
  - `UNDRAINED_TRIAXIAL_LOADING_STRAIN_CONTROL`: Undrained Triaxial loading. In this case the input file is interpreted as strain increments in the ϵ_{11} component.
  - `UNDRAINED_TRIAXIAL_LOADING_STRESS_CONTROL`: Undrained Triaxial loading. In this case the input file is interpreted as stress increments in the σ_{11} component.

  - `UNDRAINED_SIMPLE_SHEAR_LOADING_STRAIN_CONTROL`: Undrained simple-shear loading. In this case the input file is interpreted as angular strain increments in the $\gamma_{12} = 2\epsilon_{12}$ component.

- `scale_factor`: Can be used to scale the series file arbitrarily.

- `series_file`: String specifying the path to the file containing the increments (might be interpreted as strain or stress depending on the method chosen). Each line of the file contains one increment.

- `sigma0`: Components of the initial stress for the material, given in the order: $(\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{13}, \sigma_{23})$.

- `verbose_output(=N)` Whether the driver should print extra information about the material model every N steps. If Takes value 0 (no output) or N (do output every N increments). Each material implements its own output, so the format of the `Material_Output.feioutput` file is variable and material dependent.

The direct strain driver has the following format.

```
simulate constitutive testing DIRECT_STRAIN use material # <.>
  scale_factor = <.>
  series_file = <string>
  sigma0 = ( <F/L^2> , <F/L^2> , <F/L^2> , <F/L^2> , <F/L^2> , <F/L^2> )
  verbose_output = <.>
```

Where al the arguments are the same as the Bardet driver. In this case each line of the file contains all six components of the strain increment to be applied. For example:
`series_file = "increments.txt"` where each line in `increments.txt` contains $d\epsilon_{11}$ $d\epsilon_{22}$ $d\epsilon_{33}$ $d\epsilon_{12}$ $d\epsilon_{13}$ $d\epsilon_{23}$.

205.6 List of Available Commands (tentative, not up to date)

```
add acceleration field # <.> ax = <accel> ay = <accel> az = <aaccel> ;
add constraint equal_dof with master node # <.> and slave node # <.> dof to ↩
    constrain <.>;
add constraint equal_dof with node # <.> dof <.> master and node # <.> dof <.> ↩
    slave;
add damping # <.> to element # <.>;
add damping # <.> to node # <.>;
add damping # <.> type Caughey3rd with a0 = <1/time> a1 = <time> a2 = <time^3> ↩
    stiffness_to_use = ↩
    <Initial_Stiffness|Current_Stiffness|Last_Committed_Stiffness>;
add damping # <.> type Caughey4th with a0 = <1/time> a1 = <time> a2 = <time^3> ↩
    a3 = <time^5> stiffness_to_use = ↩
    <Initial_Stiffness|Current_Stiffness|Last_Committed_Stiffness>;
add damping # <.> type Rayleigh with a0 = <1/time> a1 = <time> stiffness_to_use ↩
    = <Initial_Stiffness|Current_Stiffness|Last_Committed_Stiffness>;
add domain reduction method loading # <.> hdf5_file = <string> scale_factor = <.>;
add domain reduction method loading # <.> hdf5_file = <string>;
add element # <.> type 20NodeBrick using <.> Gauss points each direction with ↩
    nodes (<.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, ↩
    <.>, <.>, <.>, <.>, <.>, <.>, <.>) use material # <.>;
add element # <.> type 20NodeBrick with nodes (<.>, <.>, <.>, <.>, <.>, <.>, ↩
    <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>) use ↩
    material # <.>;
add element # <.> type 20NodeBrick_up using <.> Gauss points each direction ↩
    with nodes (<.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, ↩
    <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>) use material # <.> and porosity = ↩
    <.> alpha = <.> rho_s = <M/L^3> rho_f = <M/L^3> k_x = <L^3T/M> k_y = ↩
    <L^3T/M> k_z = <L^3T/M> K_s = <stress> K_f = <stress>;
add element # <.> type 20NodeBrick_up with nodes (<.>, <.>, <.>, <.>, <.>, <.>, ↩
    <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>) use ↩
    material # <.> and porosity = <.> alpha = <.> rho_s = <M/L^3> rho_f = ↩
    <M/L^3> k_x = <L^3T/M> k_y = <L^3T/M> k_z = <L^3T/M> K_s = <stress> K_f = ↩
    <stress>;
add element # <.> type 20NodeBrick_upU using <.> Gauss points each direction ↩
    with nodes (<.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, ↩
    <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>) use material # <.> and porosity = ↩
    <.> alpha = <.> rho_s = <M/L^3> rho_f = <M/L^3> k_x = <L^3T/M> k_y = ↩
    <L^3T/M> k_z = <L^3T/M> K_s = <stress> K_f = <stress>;
add element # <.> type 20NodeBrick_upU with nodes (<.>, <.>, <.>, <.>, <.>, ↩
    <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>) ↩
    use material # <.> and porosity = <.> alpha = <.> rho_s = <M/L^3> rho_f = ↩
```

```
    <M/L^3> k_x = <L^3T/M> k_y = <L^3T/M> k_z = <L^3T/M> K_s = <stress> K_f = ↩
    <stress>;
add element # <.> type 27NodeBrick using <.> Gauss points each direction with ↩
    nodes (<.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, ↩
    <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>) use ↩
    material # <.>;
add element # <.> type 27NodeBrick with nodes (<.>, <.>, <.>, <.>, <.>, <.>, ↩
    <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, ↩
    <.>, <.>, <.>, <.>, <.>, <.>) use material # <.>;
add element # <.> type 27NodeBrick_up using <.> Gauss points each direction ↩
    with nodes (<.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, ↩
    <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>) ↩
    use material # <.> and porosity = <.> alpha = <.> rho_s = <M/L^3> rho_f = ↩
    <M/L^3> k_x = <L^3T/M> k_y = <L^3T/M> k_z = <L^3T/M> K_s = <stress> K_f = ↩
    <stress>;
add element # <.> type 27NodeBrick_up with nodes (<.>, <.>, <.>, <.>, <.>, <.>, ↩
    <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>) use ↩
    material # <.> and porosity = <.> alpha = <.> rho_s = <M/L^3> rho_f = ↩
    <M/L^3> k_x = <L^3T/M> k_y = <L^3T/M> k_z = <L^3T/M> K_s = <stress> K_f = ↩
    <stress>;
add element # <.> type 27NodeBrick_upU using <.> Gauss points each direction ↩
    with nodes (<.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, ↩
    <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>) ↩
    use material # <.> and porosity = <.> alpha = <.> rho_s = <M/L^3> rho_f = ↩
    <M/L^3> k_x = <L^3T/M> k_y = <L^3T/M> k_z = <L^3T/M> K_s = <stress> K_f = ↩
    <stress>;
add element # <.> type 27NodeBrick_upU with nodes (<.>, <.>, <.>, <.>, <.>, ↩
    <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>) ↩
    use material # <.> and porosity = <.> alpha = <.> rho_s = <M/L^3> rho_f = ↩
    <M/L^3> k_x = <L^3T/M> k_y = <L^3T/M> k_z = <L^3T/M> K_s = <stress> K_f = ↩
    <stress>;
add element # <.> type 3NodeShell_ANDES with nodes (<.>, <.>, <.>) use material ↩
    # <.> thickness = <l> ;
add element # <.> type 4NodeShell_ANDES with nodes (<.>, <.>, <.>, <.>) use ↩
    material # <.> thickness = <l> ;
add element # <.> type 4NodeShell_MITC4 with nodes (<.>, <.>, <.>, <.>) use ↩
    material # <.> thickness = <L>;
add element # <.> type 4NodeShell_NewMITC4 with nodes (<.>, <.>, <.>, <.>) use ↩
    material # <.> thickness = <L>;
add element # <.> type 8_27_NodeBrick using <.> Gauss points each direction ↩
    with nodes (<.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, ↩
    <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>) ↩
    use material # <.>;
```

```
add element # <.> type 8_27_NodeBrick with nodes (<.>, <.>, <.>, <.>, <.>, <.>, ↩
    <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, ↩
    <.>, <.>, <.>, <.>, <.>, <.>) use material # <.>;
add element # <.> type 8_27_NodeBrick_up using <.> Gauss points each direction ↩
    with nodes (<.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, ↩
    <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>) ↩
    use material # <.> and porosity = <.> alpha = <.> rho_s = <M/L^3> rho_f = ↩
    <M/L^3> k_x = <L^3T/M> k_y = <L^3T/M> k_z = <L^3T/M> K_s = <stress> K_f = ↩
    <stress>;
add element # <.> type 8_27_NodeBrick_up with nodes (<.>, <.>, <.>, <.>, <.>, ↩
    <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>) ↩
    use material # <.> and porosity = <.> alpha = <.> rho_s = <M/L^3> rho_f = ↩
    <M/L^3> k_x = <L^3T/M> k_y = <L^3T/M> k_z = <L^3T/M> K_s = <stress> K_f = ↩
    <stress>;
add element # <.> type 8_27_NodeBrick_upU using <.> Gauss points each direction ↩
    with nodes (<.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, ↩
    <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>) ↩
    use material # <.> and porosity = <.> alpha = <.> rho_s = <M/L^3> rho_f = ↩
    <M/L^3> k_x = <L^3T/M> k_y = <L^3T/M> k_z = <L^3T/M> K_s = <stress> K_f = ↩
    <stress>;
add element # <.> type 8_27_NodeBrick_upU with nodes (<.>, <.>, <.>, <.>, <.>, ↩
    <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>) ↩
    use material # <.> and porosity = <.> alpha = <.> rho_s = <M/L^3> rho_f = ↩
    <M/L^3> k_x = <L^3T/M> k_y = <L^3T/M> k_z = <L^3T/M> K_s = <stress> K_f = ↩
    <stress>;
add element # <.> type 8NodeBrick using <.> Gauss points each direction with ↩
    nodes (<.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>) use material # <.>;
add element # <.> type 8NodeBrick with nodes (<.>, <.>, <.>, <.>, <.>, <.>, ↩
    <.>, <.>) use material # <.>;
add element # <.> type 8NodeBrick_up using <.> Gauss points each direction with ↩
    nodes (<.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>) use material # <.> porosity ↩
    = <.> alpha = <.> rho_s = <M/L^3> rho_f = <M/L^3> k_x = <L^3T/M> k_y = ↩
    <L^3T/M> k_z = <L^3T/M> K_s = <stress> K_f = <stress>;
add element # <.> type 8NodeBrick_up with nodes (<.>, <.>, <.>, <.>, <.>, <.>, ↩
    <.>, <.>) use material # <.> porosity = <.> alpha = <.> rho_s = <M/L^3> ↩
    rho_f = <M/L^3> k_x = <L^3T/M> k_y = <L^3T/M> k_z = <L^3T/M> K_s = <stress> ↩
    K_f = <stress>;
add element # <.> type 8NodeBrick_upU using <.> Gauss points each direction ↩
    with nodes (<.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>) use material # <.> ↩
    porosity = <.> alpha = <.> rho_s = <M/L^3> rho_f = <M/L^3> k_x = <L^3T/M> ↩
    k_y = <L^3T/M> k_z = <L^3T/M> K_s = <stress> K_f = <stress>;
add element # <.> type 8NodeBrick_upU with nodes (<.>, <.>, <.>, <.>, <.>, <.>, ↩
    <.>, <.>) use material # <.> porosity = <.> alpha = <.> rho_s = <M/L^3> ↩
    rho_f = <M/L^3> k_x = <L^3T/M> k_y = <L^3T/M> k_z = <L^3T/M> K_s = <stress> ↩
```

```
    K_f = <stress>;
add element # <.> type beam_9dof_elastic with nodes (<.>, <.>) cross_section = ↩
    <area> elastic_modulus = <F/L^2> shear_modulus = <F/L^2> torsion_Jx = ↩
    <length^4> bending_Iy = <length^4> bending_Iz = <length^4> mass_density = ↩
    <M/L^3> xz_plane_vector = (<.>, <.>, <.> ) joint_1_offset = (<L>, <L>, <L> ↩
    ) joint_2_offset = (<L>, <L>, <L> );
add element # <.> type beam_displacement_based with nodes (<.>, <.>) with # <.> ↩
    integration_points use section # <.> mass_density = <M/L^3> IntegrationRule ↩
    = "" xz_plane_vector = (<.>, <.>, <.> ) joint_1_offset = (<L>, <L>, <L> ) ↩
    joint_2_offset = (<L>, <L>, <L> );
add element # <.> type beam_elastic with nodes (<.>, <.>) cross_section = ↩
    <area> elastic_modulus = <F/L^2> shear_modulus = <F/L^2> torsion_Jx = ↩
    <length^4> bending_Iy = <length^4> bending_Iz = <length^4> mass_density = ↩
    <M/L^3> xz_plane_vector = (<.>, <.>, <.> ) joint_1_offset = (<L>, <L>, <L> ↩
    ) joint_2_offset = (<L>, <L>, <L> );
add element # <.> type beam_elastic_lumped_mass with nodes (<.>, <.>) ↩
    cross_section = <area> elastic_modulus = <F/L^2> shear_modulus = <F/L^2> ↩
    torsion_Jx = <length^4> bending_Iy = <length^4> bending_Iz = <length^4> ↩
    mass_density = <M/L^3> xz_plane_vector = (<.>, <.>, <.> ) joint_1_offset = ↩
    (<L>, <L>, <L> ) joint_2_offset = (<L>, <L>, <L> );
add element # <.> type BeamColumnDispFiber3d with nodes (<.>, <.>) ↩
    number_of_integration_points = <.> section_number = <.> mass_density = ↩
    <M/L^3> xz_plane_vector = (<.>, <.>, <.> ) joint_1_offset = (<L>, <L>, <L> ↩
    ) joint_2_offset = (<L>, <L>, <L> );
add element # <.> type HardContact with nodes (<.>, <.>) axial_stiffness = ↩
    <F/L> shear_stiffness = <F/L> normal_damping = <F/L> tangential_damping = ↩
    <F/L> friction_ratio = <.> contact_plane_vector = (<.>, <.>, <.> );
add element # <.> type HardWetContact with nodes (<.>, <.>) axial_stiffness = ↩
    <F/L> shear_stiffness = <F/L> normal_damping = <F/L> tangential_damping = ↩
    <F/L> friction_ratio = <.> contact_plane_vector = (<.>, <.>, <.> );
add element # <.> type ShearBeam with nodes (<.>, <.>) cross_section = <l^2> ↩
    use material # <.>;
add element # <.> type SoftContact with nodes (<.>, <.>) ↩
    initial_axial_stiffness = <F/L> stiffening_rate = <m^-1> shear_stiffness = ↩
    <F/L> normal_damping = <F/L> tangential_damping = <F/L> friction_ratio = ↩
    <.> contact_plane_vector = (<.>, <.>, <.> );
add element # <.> type SoftWetContact with nodes (<.>, <.>) ↩
    initial_axial_stiffness = <F/L> stiffening_rate = <m^-1> shear_stiffness = ↩
    <F/L> normal_damping = <F/L> tangential_damping = <F/L> friction_ratio = ↩
    <.> contact_plane_vector = (<.>, <.>, <.> );
add element # <.> type truss with nodes (<.>, <.>) use material # <.> ↩
    cross_section = <length^2> mass_density = <M/L^3> ;
add element # <.> type variable_node_brick_8_to_27 using <.> Gauss points each ↩
    direction with nodes (<.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, ↩
```

```
    <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, <.>, ↩
    <.>, <.>) use material # <.>;
add elements (<.>) to physical_element_group "string";
add fiber # <.> using material # <.> to section # <.> fiber_cross_section = ↩
    <area> fiber_location = (<L>,<L>);
add imposed motion # <.> to node # <.> dof DOFTYPE displacement_scale_unit = ↩
    <displacement> displacement_file = "disp_filename" velocity_scale_unit = ↩
    <velocity> velocity_file = "vel_filename" acceleration_scale_unit = ↩
    <acceleration> acceleration_file = "acc_filename";
add imposed motion # <.> to node # <.> dof DOFTYPE time_step = <t> ↩
    displacement_scale_unit = <length> displacement_file = "disp_filename" ↩
    velocity_scale_unit = <velocity> velocity_file = "vel_filename" ↩
    acceleration_scale_unit = <acceleration> acceleration_file = "acc_filename";
add load # <.> to all elements type self_weight use acceleration field # <.>;
add load # <.> to element # <.> type self_weight use acceleration field # <.>;
add load # <.> to element # <.> type surface at nodes (<.> , <.> , <.> , <.>) ↩
    with magnitude <.>;
add load # <.> to element # <.> type surface at nodes (<.> , <.> , <.> , <.>) ↩
    with magnitudes (<.> , <.> , <.> , <.>);
add load # <.> to element # <.> type surface at nodes (<.> , <.> , <.> , <.>, ↩
    <.>, <.>, <.>, <.>) with magnitude <.>;
add load # <.> to element # <.> type surface at nodes (<.> , <.> , <.> , <.>, ↩
    <.>, <.>, <.>, <.>) with magnitudes (<.> , <.> , <.> , <.>, <.>, <.>, <.>, ↩
    <.>);
add load # <.> to element # <.> type surface at nodes (<.> , <.> , <.> , <.>, ↩
    <.>, <.>, <.>, <.>, <.>) with magnitude <.>;
add load # <.> to element # <.> type surface at nodes (<.> , <.> , <.> , <.>, ↩
    <.>, <.>, <.>, <.>, <.>) with magnitudes (<.> , <.> , <.> , <.>, <.>, <.>, ↩
    <.>, <.>, <.>);
add load # <.> to node # <.> type from_reactions;
add load # <.> to node # <.> type linear FORCETYPE = <force or moment>; ↩
    //FORCETYPE = Fx Fy Fz Mx My Mz F_fluid_x F_fluid_y F_fluid_z
add load # <.> to node # <.> type path_series FORCETYPE = <force or moment> ↩
    time_step = <time> series_file = "filename";
add load # <.> to node # <.> type path_time_series FORCETYPE = <force or ↩
    moment> series_file = "filename";
add load # <.> to node # <.> type self_weight use acceleration field # <.>;
add mass to node # <.> mx = <mass> my = <mass> mz = <mass> Imx = ↩
    <mass*length^2> Imy = <mass*length^2> Imz = <mass*length^2>;
add mass to node # <.> mx = <mass> my = <mass> mz = <mass>;
add material # <.> type CamClay mass_density = <M/L^3> M = <.> lambda = <.> ↩
    kappa = <.> e0 = <.> p0 = <F/L^2> Poisson_ratio = <.> ↩
    initial_confining_stress = <F/L^2>
```

```
add material # <.> type DruckerPrager mass_density = <M/L^3> elastic_modulus = ↩
    <F/L^2> poisson_ratio = <.> druckerprager_k = <> kinematic_hardening_rate = ↩
    <F/L^2> isotropic_hardening_rate = <F/L^2> initial_confining_stress = exp;
add material # <.> type DruckerPragerArmstrongFrederickLE mass_density = ↩
    <M/L^3> elastic_modulus = <F/L^2> poisson_ratio = <.> druckerprager_k = <> ↩
    armstrong_frederick_ha = <F/L^2> armstrong_frederick_cr = <F/L^2> ↩
    isotropic_hardening_rate = <F/L^2> initial_confining_stress = <F/L^2>;
add material # <.> type DruckerPragerArmstrongFrederickNE mass_density = ↩
    <M/L^3> DuncanChang_K = <.> DuncanChang_pa = <F/L^2> DuncanChang_n = <.> ↩
    DuncanChang_sigma3_max = <F/L^2> DuncanChang_nu = <.> druckerprager_k = <> ↩
    armstrong_frederick_ha = <F/L^2> armstrong_frederick_cr = <F/L^2> ↩
    isotropic_hardening_rate = <F/L^2> initial_confining_stress = <F/L^2>;
add material # <.> type DruckerPragerNonAssociateArmstrongFrederick ↩
    mass_density = <M/L^3> elastic_modulus = <F/L^2> poisson_ratio = <.> ↩
    druckerprager_k = <> armstrong_frederick_ha = <F/L^2> ↩
    armstrong_frederick_cr = <F/L^2> isotropic_hardening_rate = <F/L^2> ↩
    initial_confining_stress = <F/L^2> plastic_flow_xi = <> plastic_flow_kd = ↩
    <> ;
add material # <.> type DruckerPragerNonAssociateLinearHardening mass_density = ↩
    <M/L^3> elastic_modulus = <F/L^2> poisson_ratio = <.> druckerprager_k = <> ↩
    kinematic_hardening_rate = <F/L^2> isotropic_hardening_rate = <F/L^2> ↩
    initial_confining_stress = <F/L^2> plastic_flow_xi = <> plastic_flow_kd = ↩
    <> ;
add material # <.> type DruckerPragervonMises mass_density = <M/L^3> ↩
    elastic_modulus = <F/L^2> poisson_ratio = <.> druckerprager_k = <> ↩
    kinematic_hardening_rate = <F/L^2> isotropic_hardening_rate = <F/L^2> ↩
    initial_confining_stress = exp;
add material # <.> type linear_elastic_crossanisotropic mass_density = ↩
    <mass_density> elastic_modulus_horizontal = <F/L^2> ↩
    elastic_modulus_vertical = <F/L^2> poisson_ratio_h_v = <.> ↩
    poisson_ratio_h_h = <.> shear_modulus_h_v = <F/L^2>;
add material # <.> type linear_elastic_isotropic_3d mass_density = <M/L^3> ↩
    elastic_modulus = <F/L^2> poisson_ratio = <.>;
add material # <.> type linear_elastic_isotropic_3d_LT mass_density = <M/L^3> ↩
    elastic_modulus = <F/L^2> poisson_ratio = <.>;
add material # <.> type roundedMohrCoulomb mass_density = <M/L^3> ↩
    elastic_modulus = <F/L^2> poisson_ratio = <.> RMC_m = <.> RMC_qa = <F/L^2> ↩
    RMC_pc = <F/L^2> RMC_e = <.> RMC_eta0 = <.> RMC_Heta = <F/L^2> ↩
    initial_confining_stress = <F/L^2>
add material # <.> type sanisand2004 mass_density = <M/L^3> e0 = <.> ↩
    sanisand2004_G0 = <.> poisson_ratio = <.> sanisand2004_Pat = <stress> ↩
    sanisand2004_p_cut = <.> sanisand2004_Mc = <.> sanisand2004_c = <.> ↩
    sanisand2004_lambda_c = <.> sanisand2004_xi = <.> sanisand2004_ec_ref = <.> ↩
    sanisand2004_m = <.> sanisand2004_h0 = <.> sanisand2004_ch = <.> ↩
```

```
    sanisand2004_nb = <.> sanisand2004_A0 = <.> sanisand2004_nd = <.> ↩
    sanisand2004_z_max = <.> sanisand2004_cz = <.> initial_confining_stress = ↩
    <stress> ;
add material # <.> type sanisand2004_legacy mass_density = <M/L^3> e0 = <.> ↩
    sanisand2004_G0 = <.> poisson_ratio = <.> sanisand2004_Pat = <stress> ↩
    sanisand2004_p_cut = <.> sanisand2004_Mc = <.> sanisand2004_c = <.> ↩
    sanisand2004_lambda_c = <.> sanisand2004_xi = <.> sanisand2004_ec_ref = <.> ↩
    sanisand2004_m = <.> sanisand2004_h0 = <.> sanisand2004_ch = <.> ↩
    sanisand2004_nb = <.> sanisand2004_A0 = <.> sanisand2004_nd = <.> ↩
    sanisand2004_z_max = <.> sanisand2004_cz = <.> initial_confining_stress = ↩
    <stress> algorithm = <explicit|implicit> number_of_subincrements = <.> ↩
    maximum_number_of_iterations = <.> tolerance_1 = <.> tolerance_2 = <.>;
add material # <.> type sanisand2008 mass_density = <M/L^3> e0 = <.> ↩
    sanisand2008_G0 = <.> sanisand2008_K0 = <.> sanisand2008_Pat = <stress> ↩
    sanisand2008_k_c = <.> sanisand2008_alpha_cc = <.> sanisand2008_c = <.> ↩
    sanisand2008_xi = <.> sanisand2008_lambda = <.> sanisand2008_ec_ref = <.> ↩
    sanisand2008_m = <.> sanisand2008_h0 = <.> sanisand2008_ch = <.> ↩
    sanisand2008_nb = <.> sanisand2008_A0 = <.> sanisand2008_nd = <.> ↩
    sanisand2008_p_r = <.> sanisand2008_rho_c = <.> sanisand2008_theta_c = <.> ↩
    sanisand2008_X = <.> sanisand2008_z_max = <.> sanisand2008_cz = <.> ↩
    sanisand2008_p0 = <stress> sanisand2008_p_in = <.> algorithm = ↩
    <explicit|implicit> number_of_subincrements = <.> ↩
    maximum_number_of_iterations = <.> tolerance_1 = <.> tolerance_2 = <.>;
add material # <.> type uniaxial_concrete02 compressive_strength = <F/L^2> ↩
    strain_at_compressive_strength = <.> crushing_strength = <F/L^2> ↩
    strain_at_crushing_strength = <.> lambda = <.> tensile_strength = <F/L^2> ↩
    tension_softening_stiffness = <F/L^2>;
add material # <.> type uniaxial_elastic elastic_modulus = <F/L^2> ↩
    viscoelastic_modulus = <mass / length / time> ;
add material # <.> type uniaxial_steel01 yield_strength = <F/L^2> ↩
    elastic_modulus = <F/L^2> strain_hardening_ratio = <.> a1 = <.> a2 = <.> a3 ↩
    = <> a4 = <.> ;
add material # <.> type uniaxial_steel02 yield_strength = <F/L^2> ↩
    elastic_modulus = <F/L^2> strain_hardening_ratio = <.> R0 = <.> cR1 = <.> ↩
    cR2 = <.> a1 = <.> a2 = <.> a3 = <> a4 = <.> ;
add material # <.> type vonMises mass_density = <M/L^3> elastic_modulus = ↩
    <F/L^2> poisson_ratio = <.> von_mises_radius = <F/L^2> ↩
    kinematic_hardening_rate = <F/L^2> isotropic_hardening_rate = <F/L^2> ;
add material # <.> type vonMisesArmstrongFrederick mass_density = <M/L^3> ↩
    elastic_modulus = <F/L^2> poisson_ratio = <.> von_mises_radius = <> ↩
    armstrong_frederick_ha = <F/L^2> armstrong_frederick_cr = <F/L^2> ↩
    isotropic_hardening_rate = <F/L^2> ;
add node # <.> at (<length>,<length>,<length>) with <.> dofs;
add nodes (<.>) to physical_node_group "string";
```

```
add section # <.> type elastic3d elastic_modulus = <F/L^2> cross_section = ↩
    <L^2> bending_Iz = <L^4> bending_Iy=<L^4> torsion_Jx=<L^4> ;
add section # <.> type Elastic_Membrane_Plate elastic_modulus = <F/L^2> ↩
    poisson_ratio = <.> thickness = <length> mass_density = <M/L^3>;
add section # <.> type FiberSection TorsionConstant_GJ = <F*L^2>
add section # <.> type Membrane_Plate_Fiber thickness = <length> use material # ↩
    <.>;
add single point constraint to node # <.> dof to constrain <dof_type> ↩
    constraint value of <corresponding unit>;
add uniform acceleration # <.> to all nodes dof <.> time_step = <T> ↩
    scale_factor = <.> initial_velocity = <L/S> acceleration_file = <string>;
check mesh filename;
compute reaction forces;
define algorithm With_no_convergence_check / Newton / Modified_Newton;
define convergence test Norm_Displacement_Increment / Energy_Increment / ↩
    Norm_Unbalance / Relative_Norm_Displacement_Increment / ↩
    Relative_Energy_Increment / Relative_Norm_Unbalance tolerance = <.> ↩
    maximum_iterations = <.> verbose_level = <0>|<1>|<2>;
define dynamic integrator Hilber_Hughes_Taylor with alpha = <.>;
define dynamic integrator Newmark with gamma = <.> beta = <.>;
define load factor increment <.>;
define NDMaterial constitutive integration algorithm Forward_Euler;
define NDMaterial constitutive integration algorithm Forward_Euler_Subincrement ↩
    number_of_subincrements =<.>;
define NDMaterial constitutive integration algorithm ↩
    Forward_Euler|Forward_Euler_Subincrement|Backward_Euler|Backward_Euler_Subincrement| ↩
    yield_function_relative_tolerance = <.> stress_relative_tolerance = <.> ↩
    maximum_iterations = <.>;
define physical_element_group "string";
define physical_node_group "string";
define solver ProfileSPD / UMFPack;
define static integrator displacement_control using node # <.> dof DOFTYPE ↩
    increment <length>;
disable asynchronous output;
disable element output;
disable output;
enable asynchronous output;
enable element output;
enable output;
fix node # <.> dofs <.>;
fix node # <.> dofs all;
free node # <.> dofs <.>;
help;
if (.) { } else {};
```

```
if (.) { };
model name "name_string";
new loading stage "name_string";
output every <.> steps;
output non_converged_iterations;
output support reactions;
print <.>;
print element # <.>;
print node # <.>;
print physical_element_group "string";
print physical_node_group "string";
remove constraint equal_dof node # <.>;
remove displacement from node # <.>;
remove element # <.>;
remove imposed motion # <.>;
remove load # <.>;
remove node # <.>;
remove physical_node_group "string";
remove strain from element # <.>;
remove physical_element_group "string";
runTest;
set output compression level to <.>;
simulate <.> steps using static algorithm;
simulate <.> steps using transient algorithm time_step = <time>;
simulate <.> steps using variable transient algorithm time_step = <time> ↩
   minimum_time_step = <time> maximum_time_step = <time> number_of_iterations ↩
   = <.>;
simulate constitutive testing BARDETMETHOD use material # <.> scale_factor = ↩
   <.> series_file = <string> sigma0 = ( <F/L^2> , <F/L^2> , <F/L^2> , <F/L^2> ↩
   , <F/L^2> , <F/L^2> ) verbose_output = <.>
simulate constitutive testing constant mean pressure triaxial strain control ↩
   use material # <.> strain_increment_size = <.> maximum_strain = <.> ↩
   number_of_times_reaching_maximum_strain = <.>;
simulate constitutive testing DIRECT_STRAIN use material # <.> scale_factor = ↩
   <.> series_file = <string> sigma0 = ( <F/L^2> , <F/L^2> , <F/L^2> , <F/L^2> ↩
   , <F/L^2> , <F/L^2> ) verbose_output = <.>
simulate constitutive testing drained triaxial strain control use material # ↩
   <.> strain_increment_size = <.> maximum_strain = <.> ↩
   number_of_times_reaching_maximum_strain = <.>;
simulate constitutive testing undrained simple shear use material # <.> ↩
   strain_increment_size = <.> maximum_strain = <.> ↩
   number_of_times_reaching_maximum_strain = <.>;
simulate constitutive testing undrained triaxial stress control use material # ↩
   <.> strain_increment_size = <.> maximum_strain = <.> ↩
```

```
   number_of_times_reaching_maximum_strain = <.>;
simulate constitutive testing undrained triaxial use material # <.> ↩
   strain_increment_size = <.> maximum_strain = <.> ↩
   number_of_times_reaching_maximum_strain = <.>;
simulate using eigen algorithm number_of_modes = <.>;
ux uy uz Ux Uy Uz rx ry rz;
while (.) { };
whos;
```

205.7 List of reserved keywords

The following keywords are reserved and cannot be used as variables in a script or interactive session. Doing so would result in a syntax error.

First Order (commands)

```
a0
a1
a2
a3
a4
acceleration
acceleration_depth
acceleration_file
acceleration_filename
acceleration_scale_unit
add
algorithm
algorithm
all
all
allowed_subincrement_strain
alpha
alpha1
alpha2
and
angle
armstrong_frederick_cr
armstrong_frederick_ha
asynchronous
at
ax
axial_penalty_stiffness
axial_stiffness
axial_viscous_damping
ay
az
bending_Iy
bending_Iz
beta
Beta
beta_min
case
cases
characteristic_strength
check
chi
cohesion
combine
compression
```

```
compressive_strength
compressive_yield_strength
compute
confinement
confinement_strain
constitutive
constrain
constraint
contact_plane_vector
control
convergence
cR1
cR2
cross_section
crushing_strength
Current_Stiffness
cyclic
damage_parameter_An
damage_parameter_Ap
damage_parameter_Bn
damping
define
depth
dilatancy_angle
dilation_angle_eta
dilation_scale
direction
disable
displacement
displacement_file
displacement_scale_unit
dof
dofs
dofs
domain
druckerprager_k
DuncanChang_K
DuncanChang_n
DuncanChang_nu
DuncanChang_pa
DuncanChang_sigma3_max
DYNAMIC_DOMAIN_PARTITION
e0
each
elastic
elastic_modulus
elastic_modulus_horizontal
elastic_modulus_vertical
element
elements
else
```

```
enable
every
factor
fiber
fiber_cross_section
fiber_location
field
file
fix
fluid
free
friction_angle
friction_ratio
from
gamma
Gamma
Gauss
generate
GoverGmax
h_in
hardening_parameters_of_yield_surfaces
hardening_parameters_scale_unit
hdf5_file
hdf5_filenames_list
help
if
imposed
Imx
Imy
Imz
in
inclined
increment
initial_axial_stiffness
initial_confining_stress
initial_elastic_modulus
initial_shear_modulus
initial_shear_stiffness
initial_velocity
integration
integration_points
IntegrationRule
integrator
interface
isotropic_hardening_rate
joint_1_offset
joint_2_offset
K_f
K_s
k_x
k_y
```

```
k_z
kappa
kd_in
kinematic_hardening_rate
lambda
level
line_search_beta
line_search_eta
line_search_max_iter
liquefaction_Alpha
liquefaction_c_h0
liquefaction_Dir
liquefaction_dre1
liquefaction_Dre2
liquefaction_EXPN
liquefaction_gamar
liquefaction_mdc
liquefaction_mfc
liquefaction_pa
liquefaction_pmin
load
load_factors_list
loading
local_y_vector
local_z_vector
M
M_in
magnitude
magnitudes
mass
mass_density
master
material
max_axial_stiffness
maximum_iterations
maximum_number_of_iterations
maximum_strain
maximum_stress
maximum_time_step
method
minimal
minimum_time_step
model
model
moment_x_stiffness
moment_y_stiffness
monotonic
motion
mu
mx
my
```

```
mz
name
NDMaterial
new
newton_with_subincrement
node
nodes
number_of_cycles
number_of_files
number_of_increment
number_of_integration_points
number_of_iterations
number_of_layers
number_of_modes
number_of_subincrements
number_of_times_reaching_maximum_strain
of
output
output
output_filename
p0
parallel
peak_friction_coefficient_limit
peak_friction_coefficient_rate_of_decrease
penalty_stiffness
pi1
pi2
pi3
plastic_deformation_rate
plastic_flow_kd
plastic_flow_xi
plot
point
points
poisson_ratio
poisson_ratio_h_h
poisson_ratio_h_v
porosity
print
propagation
pure
R0
radiuses_of_yield_surface
radiuses_scale_unit
rate_of_softening
reduction
reference_pressure
remove
rempve
residual_friction_coefficient
restart
```

```
restart_files
results
rho_a
rho_f
rho_s
rho_w
RMC_e
RMC_eta0
RMC_Heta
RMC_m
RMC_pc
RMC_qa
RMC_shape_k
rounded_distance
runTest
sanisand2004_A0
sanisand2004_c
sanisand2004_ch
sanisand2004_cz
sanisand2004_ec_ref
sanisand2004_G0
sanisand2004_h0
sanisand2004_lambda_c
sanisand2004_m
sanisand2004_Mc
sanisand2004_nb
sanisand2004_nd
sanisand2004_p_cut
sanisand2004_Pat
sanisand2004_xi
sanisand2004_z_max
sanisand2008_A0
sanisand2008_alpha_cc
sanisand2008_c
sanisand2008_ch
sanisand2008_cz
sanisand2008_ec_ref
sanisand2008_G0
sanisand2008_h0
sanisand2008_K0
sanisand2008_k_c
sanisand2008_lambda
sanisand2008_m
sanisand2008_nb
sanisand2008_nd
sanisand2008_p0
sanisand2008_p_in
sanisand2008_p_r
sanisand2008_Pat
sanisand2008_rho_c
sanisand2008_theta_c
```

```
sanisand2008_X
sanisand2008_xi
sanisand2008_z_max
save
scale_factor
SCOTCHGRAPHPARTITIONER
section
section_number
sequential
series_file
set
shear
shear_length_ratio
shear_modulus
shear_modulus_h_v
shear_stiffness
shear_viscous_damping
shear_zone_thickness
ShearStrainGamma
sigma0
simulate
single
size_of_peak_plateau
sizes_of_yield_surfaces
slave
slave
soil
soil_profile_filename
soil_surface
solid
solver
stage
steps
steps
stiffening_rate
stiffness_to_use
strain
strain_at_compressive_strength
strain_at_crushing_strength
strain_hardening_ratio
strain_increment_size
stress
stress_increment_size
stress_relative_tolerance
sub-stepping
surface
surface_vector_relative_tolerance
tensile_strength
tensile_yield_strength
tension_softening_stiffness
test
```

```
test
testing
thickness
time_step
to
tolerance_1
tolerance_2
torsion_Jx
torsional_stiffness
TorsionConstant_GJ
total_number_of_shear_modulus
total_number_of_yield_surface
triaxial
type
uniaxial
uniaxial_material
uniform
unit_of_acceleration
unit_of_damping
unit_of_rho
unit_of_vs
use
using
value
velocity_file
velocity_scale_unit
verbose_output
viscoelastic_modulus
von_mises_radius
wave
wave1c
wave3c
while
whos
with
xi_in
xz_plane_vector
yield_function_relative_tolerance
yield_strength
yield_surface_scale_unit
x
y
z
```

Second Order (inside commands)

```
20NodeBrick
20NodeBrick_up
20NodeBrick_upU
27NodeBrick
27NodeBrick_up
```

```
27NodeBrick_upU
3NodeShell_ANDES
4NodeShell_ANDES
4NodeShell_MITC4
4NodeShell_NewMITC4
8_27_NodeBrick
8_27_NodeBrick_up
8_27_NodeBrick_upU
8NodeBrick
8NodeBrick_fluid_incompressible_up
8NodeBrick_up
8NodeBrick_upU
Absolute_Norm_Displacement_Increment
Absolute_Norm_Unbalanced_Force
arclength_control
Average_Norm_Displacement_Increment
Average_Norm_Unbalanced_Force
Backward_Euler
BARDETMETHOD
beam_9dof_elastic
beam_displacement_based
beam_elastic
beam_elastic_lumped_mass
BeamColumnDispFiber3d
BearingElastomericPlasticity3d
BFGS
BondedContact
CamClay
Caughey3rd
Caughey4th
constant mean pressure triaxial strain control
Cosserat8NodeBrick
Cosserat_linear_elastic_isotropic_3d
Cosserat_von_Mises
DIRECT_STRAIN
displacement_control
DOFTYPE
domain reduction method
drained triaxial strain control
DruckerPrager
DruckerPragerArmstrongFrederickLE
DruckerPragerArmstrongFrederickNE
DruckerPragerMultipleYieldSurface
DruckerPragerMultipleYieldSurfaceGoverGmax
DruckerPragerNonAssociateArmstrongFrederick
DruckerPragerNonAssociateLinearHardening
DruckerPragervonMises
dynamic
eigen
elastic3d
Elastic_Membrane_Plate
```

```
ElasticFourNodeQuad
Energy_Increment
equal_dof
F_fluid_x
F_fluid_y
F_fluid_z
FiberSection
ForceBasedCoupledHardContact
ForceBasedCoupledSoftContact
ForceBasedElasticContact
ForceBasedHardContact
ForceBasedSoftContact
FORCETYPE
Forward_Euler
Forward_Euler_Subincrement
from_reactions
Fx
Fy
Fz
Hilber_Hughes_Taylor
HyperbolicDruckerPragerArmstrongFrederick
HyperbolicDruckerPragerLinearHardening
HyperbolicDruckerPragerNonAssociateArmstrongFrederick
HyperbolicDruckerPragerNonAssociateLinearHardening
linear
linear_elastic_crossanisotropic
linear_elastic_isotropic_3d
linear_elastic_isotropic_3d_LT
Membrane_Plate_Fiber
Modified_Newton
Mx
My
Mz
Newmark
Newton
non_converged_iterations
NonlinearFourNodeQuad
Norm_Displacement_Increment
Norm_Unbalance
Parallel
path_series
path_time_series
petsc
petsc_options_string
physical_element_group
physical_node_group
Pisano
PlaneStressLayeredMaterial
PlaneStressRebarMaterial
PlasticDamageConcretePlaneStress
pressure
```

```
ProfileSPD
Rayleigh
reaction forces
reactions
Relative_Energy_Increment
Relative_Norm_Displacement_Increment
Relative_Norm_Unbalance
Relative_Norm_Unbalanced_Force
roundedMohrCoulomb
RoundedMohrCoulombMultipleYieldSurface
sanisand2004
sanisand2004_legacy
sanisand2008
self_weight
ShearBeam
solid fluid interaction transient
static
StressBasedCoupledHardContact_ElPPlShear
StressBasedCoupledHardContact_NonLinHardShear
StressBasedCoupledHardContact_NonLinHardSoftShear
StressBasedCoupledSoftContact
StressBasedCoupledSoftContact_ElPPlShear
StressBasedCoupledSoftContact_NonLinHardShear
StressBasedCoupledSoftContact_NonLinHardSoftShear
StressBasedHardContact_ElPPlShear
StressBasedHardContact_NonLinHardShear
StressBasedHardContact_NonLinHardSoftShear
StressBasedSoftContact_ElPPlShear
StressBasedSoftContact_NonLinHardShear
StressBasedSoftContact_NonLinHardSoftShear
SuperElementLinearElasticImport
support
surface
transient
truss
TsinghuaLiquefactionModelCirclePiPlane
TsinghuaLiquefactionModelNonCirclePiPlane
UMFPack
undrained simple shear
undrained triaxial
undrained triaxial stress control
uniaxial_concrete02
uniaxial_elastic
uniaxial_steel01
uniaxial_steel02
variable transient
variable_node_brick_8_to_27
vonMises
vonMisesArmstrongFrederick
vonMisesMultipleYieldSurface
vonMisesMultipleYieldSurfaceGoverGmax
```

```
With_no_convergence_check

beta
gamma
delta

ux
uy
uz
rx
ry
rz
Ux
Uy
Uz
p
M
m
kg
s
cm
mm
km
Hz
Minute
Hour
Day
Week
ms
ns
N
kN
Pa
kPa
MPa
GPa
pound
lbm
lbf
inch
in
feet
ft
yard
mile
psi
ksi
kip
g
pi

```

```
NUMBER_OF_NODES
NUMBER_OF_ELEMENTS
CURRENT_TIME
NUMBER_OF_SP_CONSTRAINTS
NUMBER_OF_MP_CONSTRAINTS
NUMBER_OF_LOADS
IS_PARALLEL
SIMULATE_EXIT_FLAG
then
while
do
let
vector

cos
sin
tan
cosh
sinh
tanh
acos
asin
atan
atan2
sqrt
exp
log10
ceil
fabs
floor
log
```

205.8 Integrated Development Environment (IDE) for DSL

205.9 Mesh Generation using GiD

1. Download the latest version of GiD from `http://www.gidhome.com/`, and also get a temporary license (or purchase it...).
2. Download essi.gid.tar.gz, unpack it (`tar -xvzf essi.gid.tar.gz`) in `problemtypes` directory that is located in GiD's root directory.
3. When you run GiD, you will see `essi` in "Data > Problem types", and can start using it...
4. A simple movie with instructions for mesh generation is available: (Link to a movie, 11MB).

205.10 Model Development and Mesh Generation using gmesh

205.11 Model Input File Editing using Sublime

`http://www.sublimetext.com/`

Chapter 206

Output Formats

(2012-2014-2017-2019-2021-)

(In collaboration with Prof. José Abell, Prof. Sumeet Kumar Sinha and Dr. Yuan Feng, and Prof. Han Yang)

206.1 Chapter Summary and Highlights

206.2 Introduction

All output from ESSI simulator is stored inside a database format, specifically designed for handling scientific array-oriented data, called HDF5 (Group, 2020). HDF stands for 'Hierarchical Data Format' and is a self-describing data format suitable for portable sharing of scientific data. The format was created and is maintained by the HDF group (`http://www.hdfgroup.org/`)

Data is stored within the file using a hierarchy similar to a unix filesystem, with groups to store related data and the actual data stored in so-called 'datasets' within each group.

HDF5 was chosen because it meets our design goals of provides:

- A simplified output format. Output is a single HDF5 file per analysis stage.
- An efficient binary (possibly compressed) file format that optimizes random access to data.
- A data format that is amenable to store output from parallel computations.
- Has a reasonable API exposed in several languages so that users can easily and customizably access simulation data.

One very convenient tool for the basic exploration of HDF5 files is the viewer 'hdfview' (`http://www.hdfgroup.org/products/java/hdfview/index.html`).

206.3 Output Filename and Format

On running any simulation on Real-ESSI simulator output files are produced for each analysis stage. The number of outputs and the filename is slightly different for sequential and parallel runs. Each output file, contains the information about the model mesh, nodal displacements, elements output, boundary conditions, material tags.. etc. The output files are designed as completely independent files containing all the data for the loading stage. In parallel each of the follower compute process outputs contains all the data corresponding to only the follower compute process. This is done to make the visualization and output process efficient.

206.3.1 Sequential

For sequential runs a single output file is produced per analysis stage. The files are named according to `model` and `stage names`, not by the filename that runs the analysis. The extension is set to be '`.h5.feioutput`', to distinguish from future possible alternative output formats.

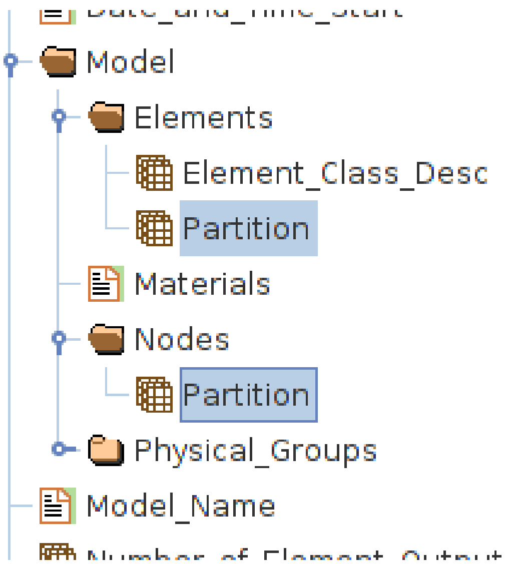


Figure 206.1: Partition info in output file produced by main compute node in parallel run

For example, if the model name is ‘`site_response`’ and the stage name is ‘`earthquake_shaking`’ the corresponding output filename will be ‘`site_response_earthquake_shaking.h5.feioutput`’.

206.3.2 Parallel

In parallel, for each stage, output files produced are equal to the number of CPU’s used. For example, a simulation run of 8 CPU’s will produce 8 output files per stage for each corresponding CPU’s (cores).The filename remains the same as sequential output each CPU (process id) used, but the extension is set to be as ‘`.h5.pid.feioutput`’, where *pid* refers to the process id of the CPU. However, the main compute process having *pid* equal to 0 follows the extension ‘`.h5.feioutput`’.

For example, In parallel, if the model name is ‘`site_response`’ and the stage name is ‘`earthquake_shaking`’ and the analysis is run on n CPU’s, the corresponding output filename for the main compute process ($pid = 0$) would be ‘`site_response_earthquake_shaking.h5.feioutput`’. All the follower compute process having $pid > 0$ would have output filename ‘`site_response_earthquake_shaking.h5.pid.feioutput`’.

The main compute process usually does not contain any nodes and element once the partition is achieved and nodes and elements are transfered to their respective CPU’s or cores. Thus, the output produced by main compute process does not contain any mesh or output results. However, it contains the partition data as shown in Figure 206.1 describing the process id on which any node or element is assigned. This information is quite useful, during post processing when the result of a particular node or element needs to be extracted. The main compute output can be read to find out the follower compute process id on which the data is located and then the output of that process id can be read to get the data of interest.

206.4 Output Units

Real-ESSI `.feioutput` file stores all the results in standard units. The table below shows the units of all the data stored in HDF5 file.

Quantity	Unit
Force F_x, F_y, F_z	N
Moments M_x, M_y, M_z	$N-m$
Pressure p	Pa
Displacement u_x, u_y, u_z	m
Rotation r_x, r_y, r_z	$radian$
Stress σ	Pa
Strain ϵ	unit less
Acceleration a_x, a_y, a_z	m/s^2
Time t	s

206.5 Data organization

In HDF5 jargon a multidimensional array is called a dataset. Datasets are indexed arrays (up to 32 dimensions) that can contain different types of data. Supported data types are: integers (various sizes), floating point numbers (`float`, `float`, `long double`, etc.), strings of text (fixed and variable size `char *`), and arbitrary structures of data (similar to C language `struct`). A file can contain as many independent datasets as needed. Datasets can be organized into 'groups', which are like folders in a file system. HDF5 provides additionally convenience data-types such as 'references', which provide views (slices) into diferent datasets or portions of them.

In the particular case of the ESSI HDF5 output, the files are designed with the contents and structure explained hereafter and depicted in Figure 206.3.

206.5.1 The `Root` group

The root of the HDF5 file contains information about each stage of loading. In parallel simulations, the information corresponds to the process Id (follower compute node) involved in that stage. The objects under this group are shown in Figure 206.4 and described in List 206.5.1

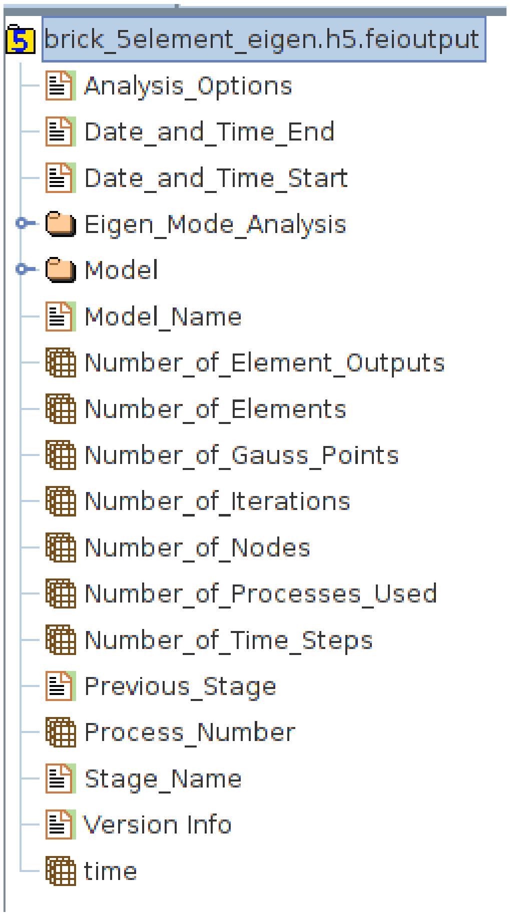


Figure 206.2: Output from a typical analysis.

- `time` : (`float`) A floating point array named which contains the available time steps for this analysis.
- `Number_of_Time_Steps`: (`int`) A single scalar integer array with the number of times steps.
- `Model_Name`: (`string`) A single string with the model name.
- `Stage_Name`: (`string`) A single string with the stage name.
- `Previous_Stage`: (`string`) A single string containing previous stage name.
- `Process_Number`: (`int`) An integer representing the *process id* by which output was generated. For sequential runs, *process id* would be zero. In parallel runs, *process id* corresponds to the *mpi rank* or *follower compute id* of processor involved in computation.
- `Number_of_Processes_Used`: (`int`) An integer representing total number of processors/CPU/nodes used in the simulation. For sequential runs, it is equal to one whereas for parallel runs, it is

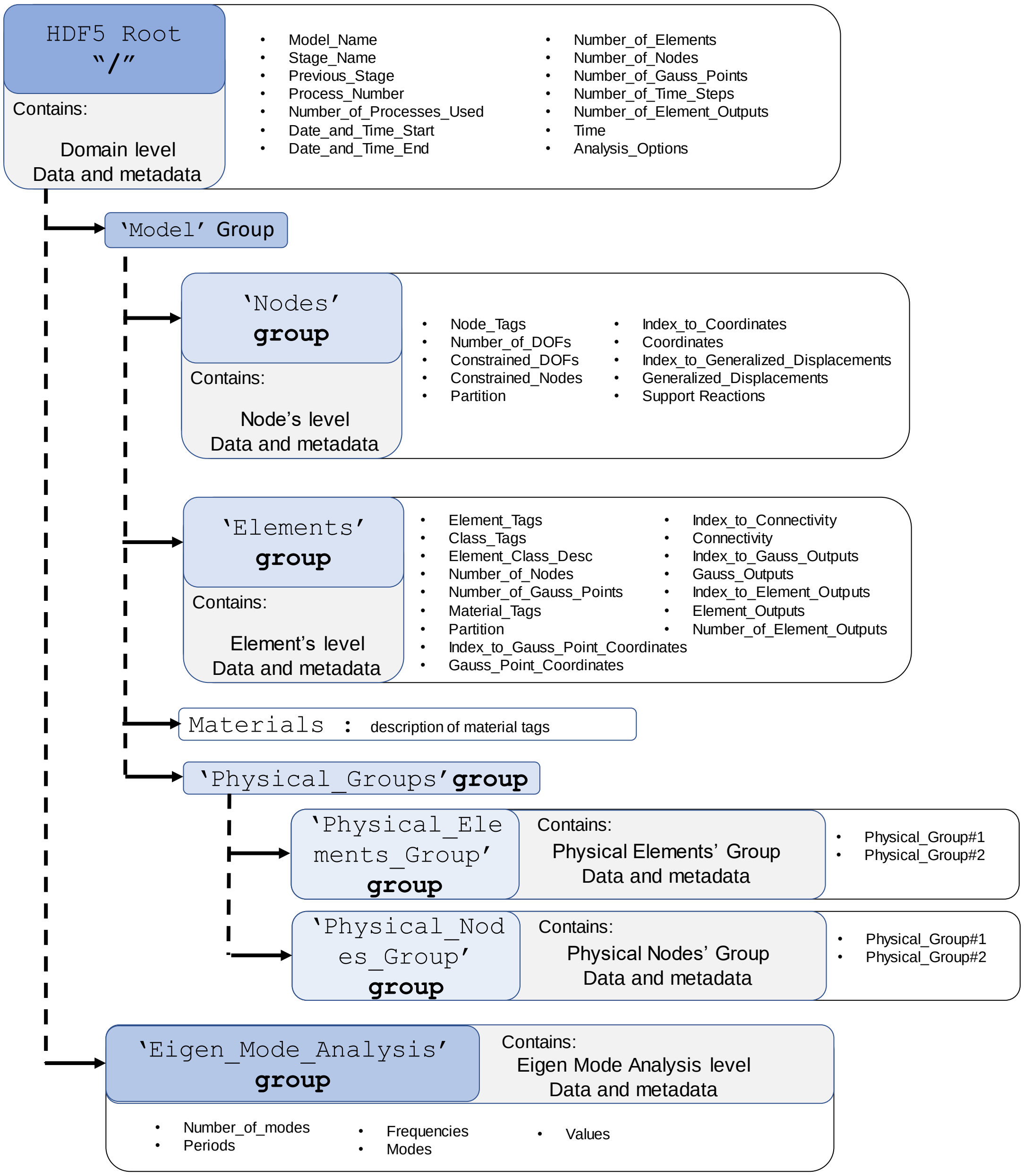


Figure 206.3: Design of the ESSI `.h5.feioutput` data format.

equal to the number of CPU's/Cores used.

- `Number_of_Nodes`: (`int`) A single scalar integer array with the number of nodes defined in that domain.
- `Number_of_Elements`: (`int`) A single scalar integer array with the number of elements defined in that domain.

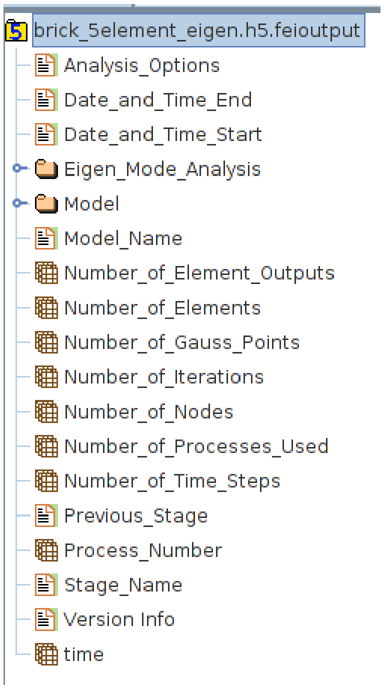


Figure 206.4: Data accessible in the `Root` directory of HDF5 file.

- `Number_of_Gauss_Points`: (`int`) A single scalar integer array with the number of gauss points in that domain.
- `Number_of_Element_Outputs`: (`int`) A single scalar integer array with stores the total length of `Element_Output` array in that domain.
- `Analysis_Options`: (`string`) An array of strings with the analysis options selected for the current analysis.
- `Date_and_Time_Start`: (`string`) A single string with the Date and Time of the start of the analysis. (In Coordinated Universal Time, UTC)
- `Date_and_Time_End`: (`string`) A single string with the Date and Time of the end of the analysis. (In Coordinated Universal Time, UTC)

  item `Version_Info`:(`string`) A Long string containing the version information of Real-ESSI simulator.
- `Model`: A group that contains the `Nodes` and `Elements` groups. It contains essential information about the mesh and analysis results for nodes and elements. See Section 206.5.2

- `Eigen_Mode_Analysis`: A group that contains the information about the the eigen mode analysis results of the doamin. See Section 206.5.6

206.5.2 The `Model` group

The `Model` group contains information about the mesh and analysis outputs. It contains the following groups as shown in Figure 206.5 and is also described below

- `Nodes`: A group that contains the information about the defined nodes and their output for this analysis. See Section 206.5.3
- `Elements`: A group that contains the information about the defined Elements and their output for this analysis. See Section 206.5.4
- `Physical_Groups`: A group that contains the information physical group of elements and nodes defined in that domain. See Section 206.5.5
- `Material`: (`string`) A string array which contains information about the material tag defined in the analysis for that loading stage. Section 206.5.7

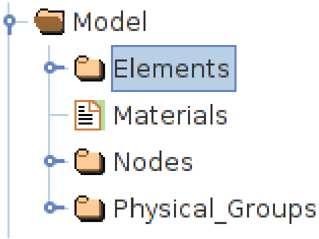


Figure 206.5: `Model` group directory of HDF5 file.

Subgroups `Nodes` and `Elements` store several integer and double precision arrays, that contain all necessary information for post processing.

206.5.3 The `Nodes` group

The `Nodes` group contains information about the nodal coordinates of the model, their tags, the number of DOFs defined at each node, and the corresponding solution variables (DOF results or generalized displacements) for each time step.

The format used to store the data is designed to give the fastest possible access time to the data of interest. Stored within the `Nodes` groups (and also in `Elements`) are two types of arrays: *data arrays* and *index arrays*.

- Data Arrays :: Data arrays might be floating-point arrays or integer arrays and have names not starting with '`Index_to_`'.
- Index Arrays :: All index arrays are integer arrays and have names starting with the word '`Index_to_`' followed by the name of the array which this array indexes.

The concept of *index array* is an important one regarding speed of access to data. These arrays map the integer tag number of the nodes (or elements) to the data. This allows fast access to components which minimizes searching within arrays to find the data of interest.

The `Nodes` group contains the following index arrays.

- `Index_to_Coordinates`: (`int`) Indexes the coordinates of nodes. See section 206.5.3.6
- `Index_to_Generalized_Displacements`: (`int`) Indexes the outputs of nodes (generalized displacements). See section 206.5.3.7

The following are the data arrays available in the `Nodes` group (shown in Figure ?? along with their respective indexing array:

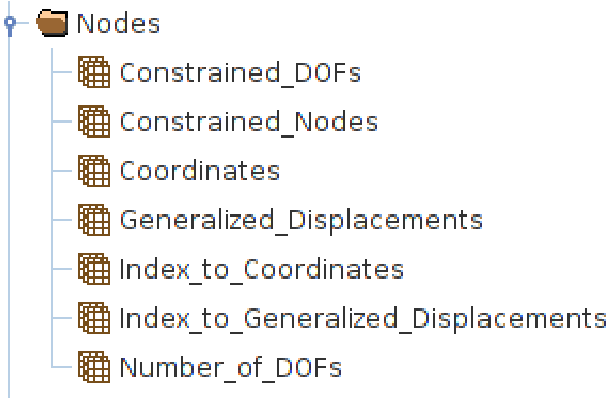

Nodes
Constrained_DOFs
Constrained_Nodes
Coordinates
Generalized_Displacements
Index_to_Coordinates
Index_to_Generalized_Displacements
Number_of_DOFs

Figure 206.6: `Nodes` group directory of HDF5 file.

- `Coordinates`: (`float`) 1-D array containing nodal coordinates fixed in time. [Indexed by `Index_to_Coordinates` array]. See section 206.5.3.6
- `Generalized_Displacements`: (`float`) 2-D array containing the DOF values for the solution at each time step. [Indexed by `Index_to_Generalized_Displacements` array]. See section 206.5.3.7
- `Number_of_DOFs`: (`int`) 1-D array mapping the integer tag of each node to the number of DOFs at that node. See section 206.5.3.1

- `Constrained_Nodes`: (`int`) 1-D array mapping the integer tag of each node to the number of DOFs at that node. See section 206.5.3.3
- `Constrained_DOFs`: (`int`) 1-D array mapping the integer tag of each node to the number of DOFs at that node. See section 206.5.3.4
- `Partition`: (`int`) 1-D array mapping the integer tag of each node to the number of DOFs at that node. See section 206.5.3.2
- `Support_Reactions`: (`float`) 1-D array mapping the integer tag of each node to the number of DOFs at that node. See section 206.5.3.5

For example, lets imagine that the user has defined 4 nodes and applied the following constrained as shown below :

Listing 206.1: Node_Example

```
// defining nodes
add node # 2 at (0*m, 0*m, 0*m) with 3 dofs;
add node # 4 at (1*m, 1*m, 1*m) with 6 dofs;
add node # 5 at (2*m, 2*m, 2*m) with 3 dofs;
add node # 6 at (1*m, 0*m, 5*m) with 3 dofs;

// applying constraints
fix node # 2 dofs ux uy;
fix node # 4 rx ry rz;
fix node # 6 Ux p;
```

The index and the data arrays for the given example would look like the following as shown in the subsections ahead.

206.5.3.1 `Number_of_DOFs`

`Number_of_DOFs` array defines the number of degrees of freedom for each node defined in the model. It is an `integer` array of length equal to the maximum node tag + 1 (including tag 0). If a node tag does not exists, the corresponding dofs is output as -1. Figure 206.7 shows how to read `Number_of_DOFs` array. In the given example Listing 206.1 , node tag 2 has 3 degrees of freedom. Similarly, node tag 4 has 6 degrees of freedom and so on.

206.5.3.2 `Partition`

`Partition` array contains the domain or process id on which nodes tags were defined in case of a parallel simulation. For sequnetial runs, this dataset is not available. If a node tag does not exists, the corresponding partition process id is output as -1. Figure 206.7 shows how to read `Partition` array. In the given example Listing 206.1, node tag 2 is assigned to process id 1. Similarly, node tag 4 is assigned to process id 2 and so on.

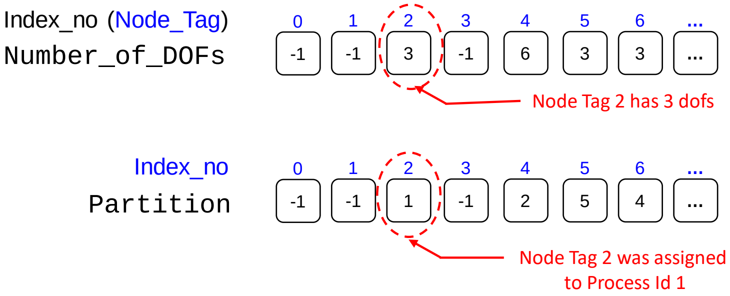


Figure 206.7: Arrays describing node information in `Nodes` group directory of HDF5 file.

206.5.3.3 Constrained_Nodes

`Constrained_Nodes` array contains a list of node tags for each dof on which fixities were applied. Figure 206.8 shows how to read `Constrained_Nodes`. In the given example Listing 206.1, dof ux and uy of node tag 2 is fixed. Similarly, for node tag 4 dofs rx ry and rz are fixed. Thats why in the `Constrained_Nodes` array contains node tags 2,2,4,4,4 and so on multiple times for each dof of the corresponding node tag fixed.

206.5.3.4 Constrained_DOFs

`Constrained_DOFs` array contains a list of dofs of the corresponding node tag on which fixities were applied. Figure 206.8 shows how to read `Constrained_DOFs`. In the given example Listing 206.1, dof ux (0) and uy (1) of node tag 2 is fixed. Similarly, for node tag 4 dofs rx (3) ry (4) and rz (5) are fixed. Figure 206.8 also show the DOF if numbering for different dof types i.e. for 3dof, 4dof, 6dof and 7dof nodes.

206.5.3.5 Support_Reactions

`Support_Reactions` contains a (`float`) array of reaction forces for the constrained degree of freedoms (DOFs). Figure 206.8 shows how to read `Constrained_DOFs`. In the given example Listing 206.1, dof ux (0) and uy (1) of node tag 2 has support reactions 1N and -5N respectively. Similarly, for node tag 4 dofs rx (3) ry (4) and rz (5) have reactions 45, 3 and -5 $N-m$ respectively. The reaction forces for displacement dofs (u_x, u_y, u_z) are forces in units of N, for rotational dofs (r_x, r_y, r_z) are moments in units

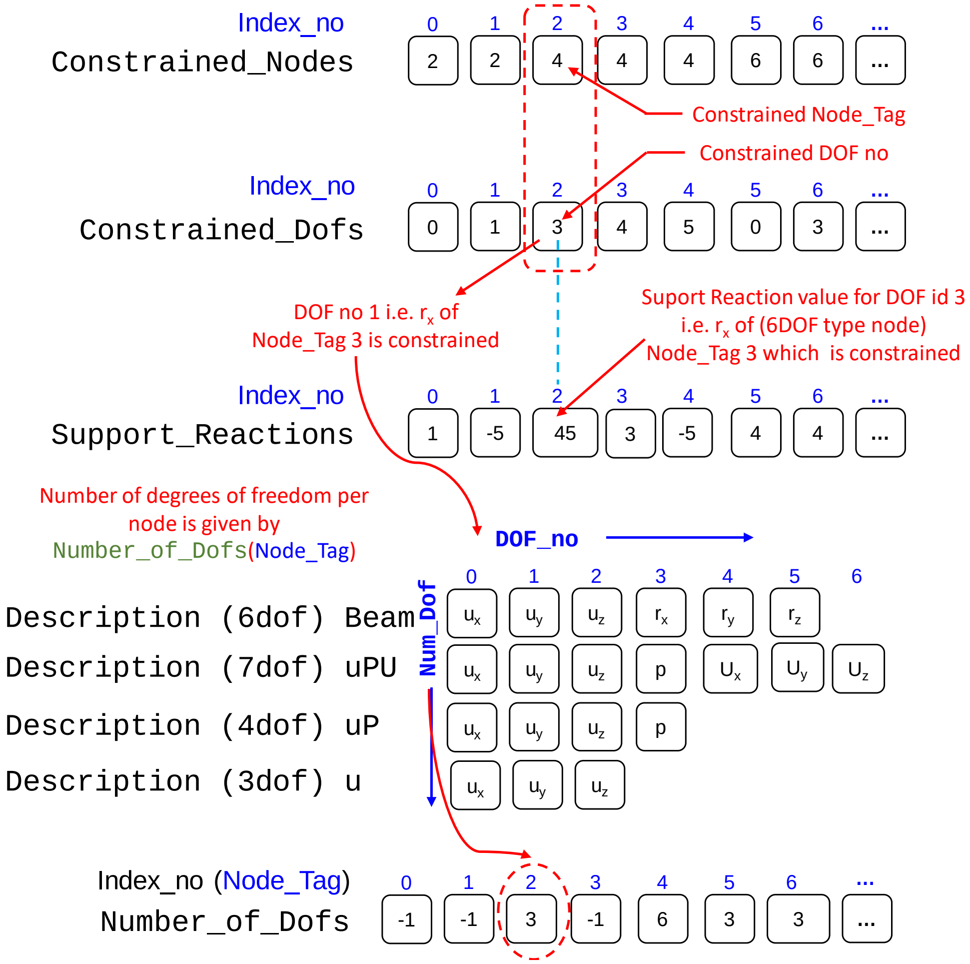


Figure 206.8: Arrays describing constrained nodes and reaction information in `Nodes` group directoory of HDF5 file.

of $N-m$ and for pressure dof (p) is pascal (Pa). Section 206.6 describes the output definitions for nodes with different dof-types.

206.5.3.6 `Coordinates`

The `Coordinates` array is a vertical stack of the nodal coordinate values. it is indexed by the `Index_to_Coordinates`, which relates the integer tag of each nodes (defined at the moment of creation of every node) with the position on this array of the 3 nodal coordinates. If the node with a given tag is not defined (if a tag number or several are skipped) this array will contain a negative number (-1) for that tag value. Figure 206.9 shows how to read `Coordinates` and `index_to_Coordinates` of nodes.

The size of `Index_to_Coordinates` is always the maximum tag defined plus one (zero can be a tag too). The size of the `Coordinates` array is three times the number of nodes defined. In the given

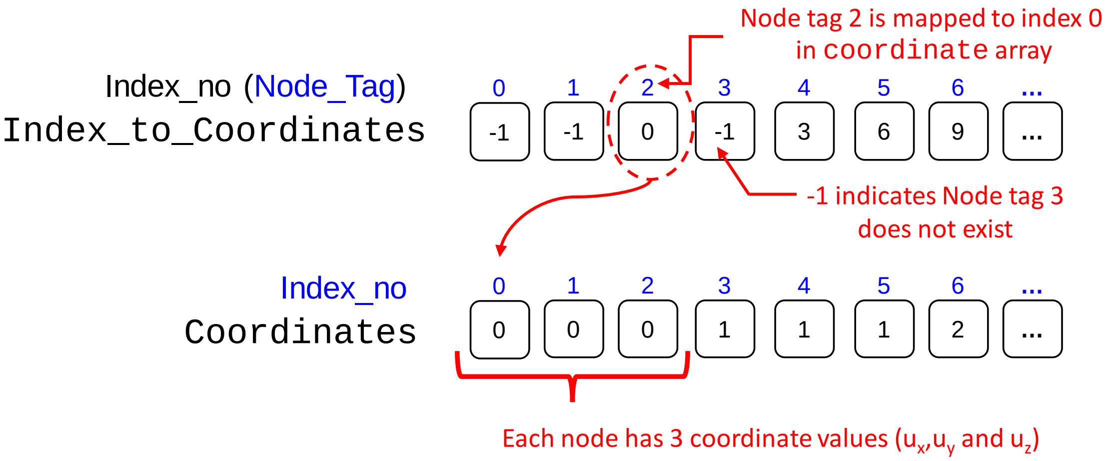


Figure 206.9: `Coordinates` and `Index_to_Coordinates` arrays in `Nodes` group directory of HDF5 file.

example Listing 206.1, the coordinates of node tag 2 is (0*m,0*m,0*m). The coordinate of node tag 4 is (1*m,1*m,1*m) and so on. The coordinates have unit of meter (m).

206.5.3.7 `Generalized_Displacements`

The `Generalized_Displacements` array is a 2-D array containing the computed solution at the nodal degrees of freedom for all nodes and all times steps. It is indexed by the `Index_to_Generalized_Displacements` array (first index) and time (second index). Figure 206.10 shows how to read `Generalized_Displacements` and `index_to_Generalized_Displacements` of nodes.

Output for displacement dofs (u_x, u_y, u_z) are in units of m, for rotational dofs (r_x, r_y, r_z) are in units of $radian$ and for pressure dof (p) is in pascal (Pa).

With every time step, another column is added to the `Generalized_Displacements` array. This means that the time index (starting at 0) is directly related to the `time` array at the root of the HDF5 file.

The rows of the `Generalized_Displacements` array contain the results for each DOF of the current node. For a 3-DOF node these will be displacement in X, Y, and Z (u_x, u_y, and u_z) respectively. For higher number-of-dof nodes the first three components are the same but the remaining ones are going to depend on the connecting elements. For example, 6-DOF nodes might carry information on nodal rotations (beams and shells) or might be fluid displacements in case the elements connecting are u–U coupled fluid-porous-solid elements as shown in Figure 206.8. Section 206.6 describes the output definitions for nodes with different dof-types.

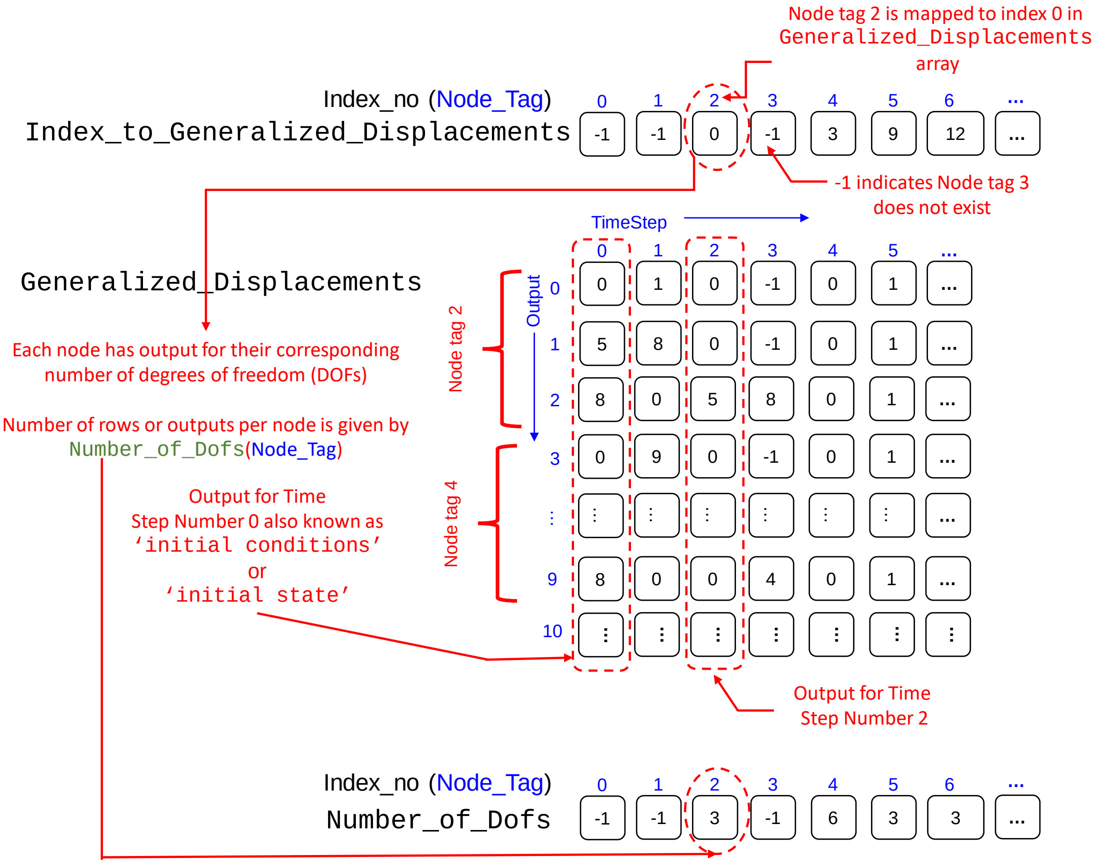


Figure 206.10: The `Index_to_Generalized_Displacements` and `Generalized_Displacements` in `Nodes` group directory of HDF5 file.

206.5.4 The `Elements` group

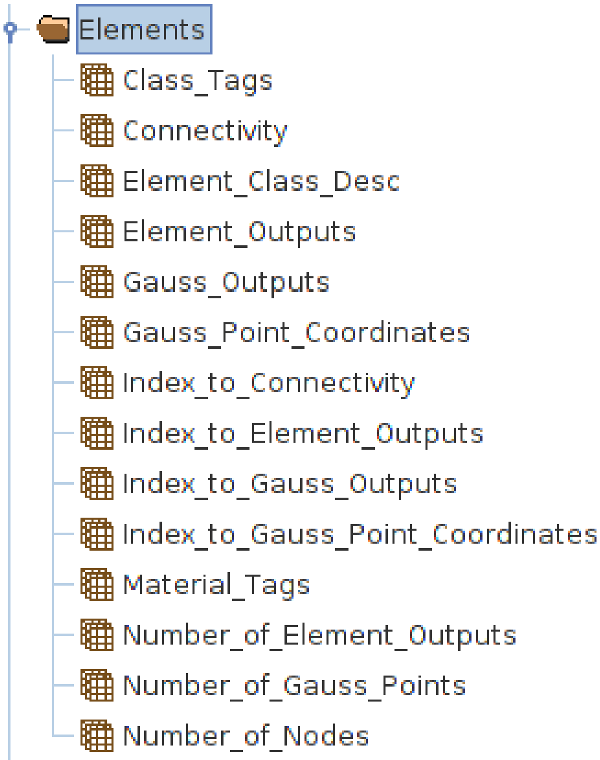
The `Elements` group contains information on the finite element mesh such as: connectivity array, element types, location of Gauss-point integration coordinates (global), materials defined at each element, and all available output from the elements. At this point it is important to note that the kind and amount of output contained in the element output arrays are dependent on element implementation as it is the elements who are in charge of controlling their output. For information on the specific output of each element, consult the documentation for the respective element.

The idea and organization of the datasets contained herein is analogous to the of the `Nodes` group. This group contains the following *index arrays*.

- `Index_to_Connectivity`: (`int`) Maps element tag to location within the `Connectivity` array.
- `Index_to_Gauss_Point_Coordinates`: Maps element tag to location of Gauss-point coordinates in the `Gauss_Point_Coordinates` array.

- `Index_to_Gauss_Outputs`: (`int`) Maps element tag to location of gauss output of element in the `Gauss_Outputs` array.
- `Index_to_Element_Outputs`: (`int`) Maps element tag to location of output in the `Elements_Outputs` array.

The following are the data arrays available in the `Elements` group (shown in Figure ?? along with their respective indexing array:



Elements
Class_Tags
Connectivity
Element_Class_Desc
Element_Outputs
Gauss_Outputs
Gauss_Point_Coordinates
Index_to_Connectivity
Index_to_Element_Outputs
Index_to_Gauss_Outputs
Index_to_Gauss_Point_Coordinates
Material_Tags
Number_of_Element_Outputs
Number_of_Gauss_Points
Number_of_Nodes

Figure 206.11: `Elements` group directory of HDF5 file.

- `Class_Tags`: (`int`) It is an array that contains the integer ids for each elements tag defined in the model and present in thet domain. See section 206.5.4.4
- `Number_of_Nodes`: (`int`) Maps element tag number to number of nodes in the element (-1 if element tag is not defined). See section 206.5.4.1.
- `Number_of_Gauss_Points`: (`int`) Maps tag number to the number of Gauss-integration points in that element (-1 if not element tag is not defined). It stores 0 in case of no gauss points (mostly for structural elements). See section 206.5.4.3

- `Material_Tags`: (`int`) Maps tag number to the tag number of the material contained in that element (-1 if element tag is not defined or material for that element is not defined). See section 206.5.4.6
- `Partition`: (`int`) Maps tag number to the tag number of the processor id on which it is assigned (-1 if element tag is not defined). See section 206.5.4.5
- `Connectivity`:(`int`) Contains the nodes tags which are connected by this element [indexed by the `Index_to_Connectivity` array] (-1 if element tag is not defined). See section 206.5.4.7
- `Gauss_Point_Coordinates`: (`float`) Contains the coordinates of all the gauss pints of that element tag [indexed by the `Index_to_Gauss_Point_Coordinates` array] (-1 if element tag is not defined) . See section 206.5.4.8
- `Gauss_Outputs`: (`float`) Contains the stress, strain and plastic strain outputs for each gauss point present in the corresponding element tag[indexed by the `Index_to_Gauss_Outputs` array]. See section 206.5.4.9
- `Element_Outputs`: (`float`) Contains output other than gauss points by the [indexed by the `Index_to_Element_Outputs` array]. See section 206.5.4.2

For example, lets imagine that the user has defined 4 elements and some materials as shown below :

Listing 206.2: Element_Example

```

// defining materials

add material #1 type uniaxial_elastic elastic_modulus = 1*Pa ↩
    viscoelastic_modulus = 0*Pa*s;
add material #2 type linear_elastic_isotropic_3d mass_density = 2000*kg/m^3 ↩
    elastic_modulus = 200*MPa poisson_ratio = 0.3;

// defining elements
add element #2 type truss with nodes (1,2) use material # 1 cross_section = ↩
    1*m^2 mass_density = 0*kg/m^3;
add element #4 type 8NodeBrick with nodes (1,8,6, 4, 3, 9, 2, 5) use material #2;
add element #5 type HardContact with nodes (3,2) normal_stiffness =1e10*N/m ↩
    tangential_stiffness = 1e4*Pa*m normal_damping = 0*kN/m*s ↩
    tangential_damping = 0*N/m*s friction_ratio = 1 contact_plane_vector = ↩
    (0,0,1);
add element #6 type 8NodeBrick with nodes (11,18,61,14, 3,19,22,15) use ↩
    material #2;
```

The index and the data arrays for the given example would look like the following as shown in the subsections ahead.

206.5.4.1 `Number_of_Nodes`

`Number_of_Nodes` array defines the number of nodes for each element defined in the model. It is an `integer` array of length equal to the maximum element tag + 1 (including tag 0). If a element tag does not exists, the corresponding number of nodes is output as -1. Figure 206.12 shows how to read `Number_of_Nodes` array. In the given example Listing 206.2 , element tag 2 has 2 nodes. Similarly, element tag 4 has 8 nodes and so on.

206.5.4.2 `Number_of_Element_Outputs`

`Number_of_Element_Outputs` array defines the number of outputs for each element defined in the model. It is an `integer` array of length equal to the maximum element tag + 1 (including tag 0). If a element tag does not exists, the corresponding number of outputs is stored as-1. Figure 206.12 shows how to read `Number_of_Element_Outputs` array. In the given example Listing 206.2 , element tag 2 has 2 outputs. Similarly, element tag 5 has 9 outputs but element tag 4 has 0 outputs and so on.

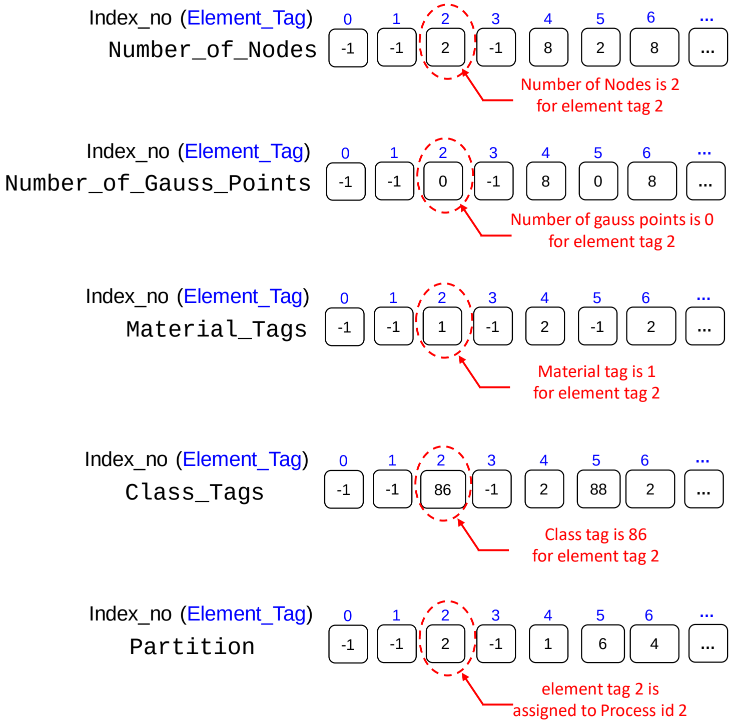


Figure 206.12: Arrays describing element information in `Elements` group directory of HDF5 file.

206.5.4.3 Number_of_Gauss_Points

`Number_of_Gauss_Points` array defines the number of gauss points for each element defined in the model. It is an `integer` array of length equal to the maximum element tag + 1 (including tag 0). If a element tag does not exists, the corresponding number of outputs is stored as-1. Figure 206.12 shows how to read `Number_of_Gauss_Points` array. In the given example Listing 206.2 , element tag 2 and 5 has 0 gauss points. Similarly, element tag 4 has 8 gauss points and so on.

206.5.4.4 Class_Tags

`Class_Tags` array defines an (unique) element type for each of the element defined in the model. It is an `integer` array of length equal to the maximum element tag + 1 (including tag 0). If a element tag does not exists, the corresponding number of outputs is stored as-1. Figure 206.12 shows how to read `Class_Tags` array. In the given example Listing 206.2 , element tag 2 has class tag of 88 (i.e. truss element) . Similarly, element tag 5 has class tag 2 (i.e. 8 node brick element) and so on. Table 206.1 and Table 206.2 shows class tags for different element types.

206.5.4.5 Partition

`Partition` array contains the domain or process id on which element tags were defined in case of a parallel simulation. For sequential runs, this dataset is not available. If a element tag does not exists, the corresponding partition process id is output as -1. Figure 206.12 shows how to read `Partition` array. In the given example Listing 206.2, element tag 2 is assigned to process id 2. Similarly, element tag 4 is assigned to process id 1 and so on.

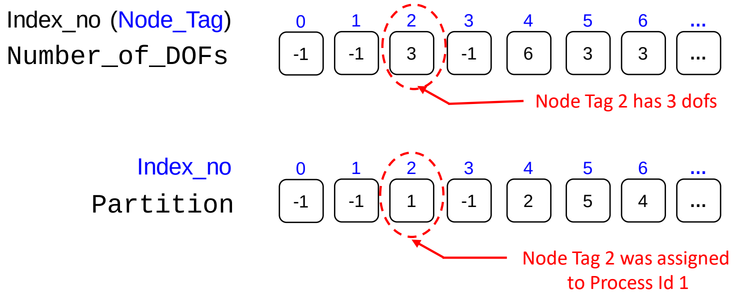


Figure 206.13: Arrays describing node information in `Nodes` group directory of HDF5 file.

Element Type	Class Tag	Element Type	Class Tag
EightNodeBrick	2	EightNodeBrick_up	3
EightNodeBrick_upU	4	TwentyNodeBrick	5
TwentyNodeBrick_up	6	TwentyNodeBrick_upU	7
TwentySevenNodeBrick	8	TwentySevenNodeBrick_up	9
TwentySevenNodeBrick_upU	10	VariableNodeBrick	11
VariableNodeBrick_up	12	VariableNodeBrick_upU	13
EightNodeBrickOrderOne	14	EightNodeBrickOrderOne_up	15
EightNodeBrickOrderOne_upU	16	TwentyNodeBrickOrderOne	17
TwentyNodeBrickOrderOne_up	18	TwentyNodeBrickOrderOne_upU	19
TwentySevenNodeBrickOrderOne	20	TwentySevenNodeBrickOrderOne_up	21
TwentySevenNodeBrickOrderOne_upU	22	VariableNodeBrickOrderOne	23
VariableNodeBrickOrderOne_up	24	VariableNodeBrickOrderOne_upU	25
EightNodeBrickOrderTwo	26	EightNodeBrickOrderTwo_up	27
EightNodeBrickOrderTwo_upU	28	TwentyNodeBrickOrderTwo	29
TwentyNodeBrickOrderTwo_up	30	TwentyNodeBrickOrderTwo_upU	31
TwentySevenNodeBrickOrderTwo	32	TwentySevenNodeBrickOrderTwo_up	33
TwentySevenNodeBrickOrderTwo_upU	34	VariableNodeBrickOrderTwo	35
VariableNodeBrickOrderTwo_up	36	VariableNodeBrickOrderTwo_upU	37
EightNodeBrickOrderThree	38	EightNodeBrickOrderThree_up	39
EightNodeBrickOrderThree_upU	40	TwentyNodeBrickOrderThree	41
TwentyNodeBrickOrderThree_up	42	TwentyNodeBrickOrderThree_upU	43
TwentySevenNodeBrickOrderThree	44	TwentySevenNodeBrickOrderThree_up	45
TwentySevenNodeBrickOrderThree_upU	46	VariableNodeBrickOrderThree	47
VariableNodeBrickOrderThree_up	48	VariableNodeBrickOrderThree_upU	49

Table 206.1: Class Tags for Real-ESSI Elements (Part -1)

206.5.4.6 `Material_Tags`

`Material_Tags` array defines the material tag number for each element defined in the model. It is an `integer` array of length equal to the maximum element tag + 1 (including tag 0). If a element tag does not exists, the corresponding number of outputs is stored as-1. Figure 206.12 shows how to read `Material_Tags` array. In the given example Listing 206.2 , element tag 2 has material tag of 1. Similarly, element tag 4 has material tag 2. Whereas, element tag 5 have material tag of -1.

Element Type	Class Tag	Element Type	Class Tag
EightNodeBrickOrderFour	50	EightNodeBrickOrderFour_up	51
EightNodeBrickOrderFour_upU	52	TwentyNodeBrickOrderFour	53
TwentyNodeBrickOrderFour_up	54	TwentyNodeBrickOrderFour_upU	55
TwentySevenNodeBrickOrderFour	56	TwentySevenNodeBrickOrderFour_up	57
TwentySevenNodeBrickOrderFour_upU	58	VariableNodeBrickOrderFour	59
VariableNodeBrickOrderFour_up	60	VariableNodeBrickOrderFour_upU	61
EightNodeBrickOrderFive	62	EightNodeBrickOrderFive_up	63
EightNodeBrickOrderFive_upU	64	TwentyNodeBrickOrderFive	65
TwentyNodeBrickOrderFive_up	66	TwentyNodeBrickOrderFive_upU	67
TwentySevenNodeBrickOrderFive	68	TwentySevenNodeBrickOrderFive_up	69
TwentySevenNodeBrickOrderFive_upU	70	VariableNodeBrickOrderFive	71
VariableNodeBrickOrderFive_up	72	VariableNodeBrickOrderFive_upU	73
EightNodeBrickOrderSix	74	EightNodeBrickOrderSix_up	75
EightNodeBrickOrderSix_upU	76	TwentyNodeBrickOrderSix	77
TwentyNodeBrickOrderSix_up	78	TwentyNodeBrickOrderSix_upU	79
TwentySevenNodeBrickOrderSix	80	TwentySevenNodeBrickOrderSix_up	81
TwentySevenNodeBrickOrderSix_upU	82	VariableNodeBrickOrderSix	83
VariableNodeBrickOrderSix_up	84	VariableNodeBrickOrderSix_upU	85
HardContact	86	SoftContact	87
Truss	88	ElasticBeam	89
ThreeNodeAndesShell	90	FourNodeAndesShell	91
ShearBeam	92	rank_one_deficient_elastic_pinned_fixed_beam	93
DispBeamColumn3d	94	Cosserat_8node_brick	95

Table 206.2: Class Tags for Real-ESSI Elements (Part -2)

206.5.4.7 `Connectivity`

`Connectivity` array stores the list of node tags in the same order as they were defined along with element declaration in the model. It is an `integer`. It is indexed by the `Index_to_Connectivity` array and the number of rows is defined by `Number_of_Nodes`. Figure 206.14 shows how to read `Connectivity` array for a particular element. In the given example Listing 206.2 , element tag 2 includes node tag 1 and 2. Similarly, element tag 4 includes 8 node tas 1,8,6, 4, 3, 9, 2 and 5 respectively.

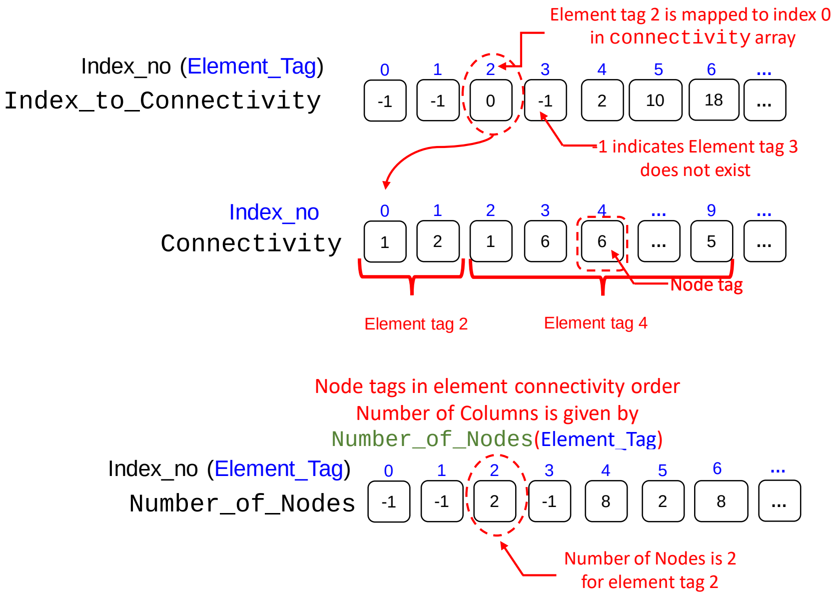


Figure 206.14: Arrays describing connectivity information in `Elements` group directory of HDF5 file.

206.5.4.8 `Gauss_Point_Coordinates`

`Gauss_Point_Coordinates` array stores the coordinates of the gauss points declared inside for each element defined in the model. It is a `float` array indexed by the `Index_to_Gauss_Point_Coordinates` array and the number of rows is defined by 3 (x, y, z) times `Number_of_Gauss_Points`. Figure 206.15 shows how to read `Gauss_Point_Coordinates` array for a particular element. In the given example Listing 206.2, element tag 2 and 5 has no gauss points. Since, element tag 4 has 8 gauss points, the total length of gauss point coordinates output for that element is $8 \times 3 = 24$. The index from which the coordinates information start is 0. Coordinate values for first 3 index $(0, 1, 2)$ corresponds to gauss point 1 and next 3 index $(3, 4, 5)$ corresponds to gauss point 2 and so on.

206.5.4.9 `Gauss_Output`

`Gauss_Outputs` array stores the coordinates of the gauss points declared inside for each element defined in the model. It is a 2D `float` array indexed by the `Index_to_Gauss_Outputs` array and the number of rows is defined by 18 (6 total strain, 6 plastic strain and 6 stress) times `Number_of_Gauss_Points`. The column index is represented by the time step of the simulation. Time index 0 represents initial state conditions, i.e. the state before the start of new stage and end of previous stage.

Figure 206.16 shows how to read `Gauss_Outputs` array for a particular element. In the given example Listing 206.2, element tag 2 and 5 has no gauss points. Since, element tag 4 has 8 gauss points, the

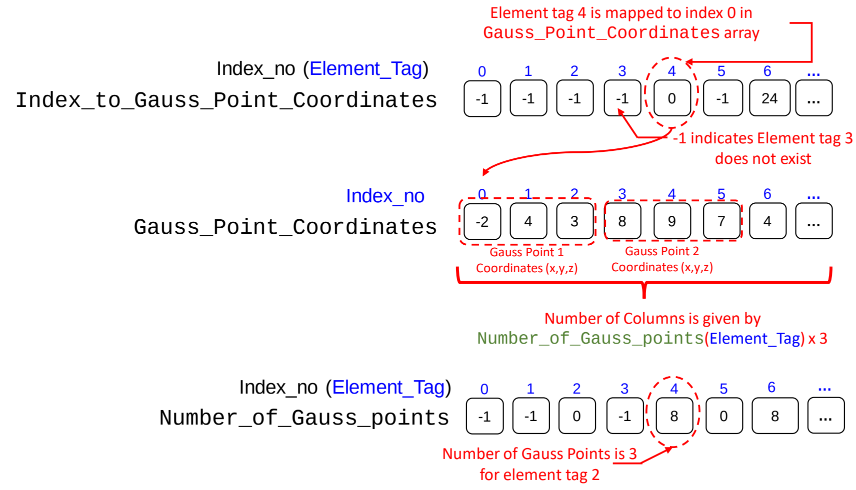


Figure 206.15: Arrays describing gauss coordinates information in `Elements` group directory of HDF5 file.

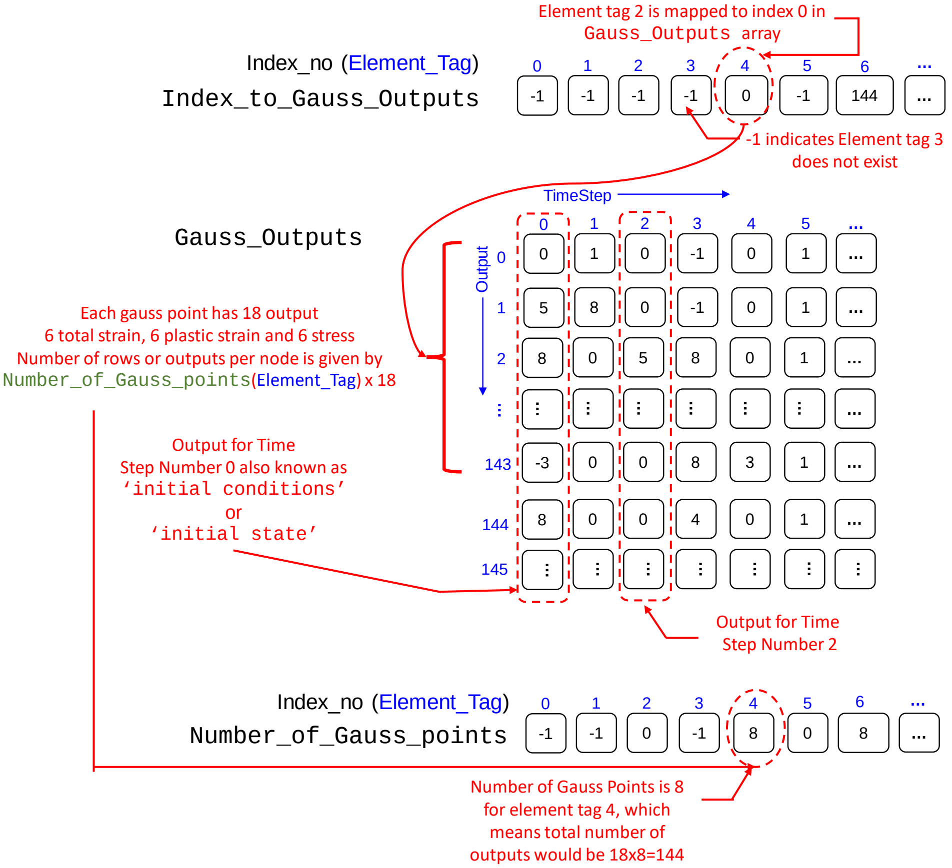


Figure 206.16: Arrays describing gauss output information in `Elements` group directory of HDF5 file.

total length of gauss output for that element is $8 \times 18 = 144$. The index from which the coordinates information start is 0. gauss values for first 18 index $(0, 1..17)$ corresponds to gauss point 1 and next 18 index $(18..35)$ corresponds to gauss point 2 and so on. For gauss point 1, first 6 index $(0, 1..5)$ corresponds to total strain, next 6 index $(6..11)$ corresponds to plastic strain and next 6 index $(12..17)$ corresponds to stress. Section 206.7 describes how gauss output is stored and what are the units and meaning of those outputs.

206.5.4.10 `Element_Outputs`

`Element_Outputs` array stores output for the elements except those stored at gauss points (i.e. stress, total strain and plastic strain). It is a 2D `float` array indexed by the `Index_to_Element_Outputs` array and the number of rows is defined `Number_of_Element_Outputs`. he column index is represented by the time step of the simulation. Time index 0 represents initial state conditions, i.e. the state before the start of new stage and end of previous stage.

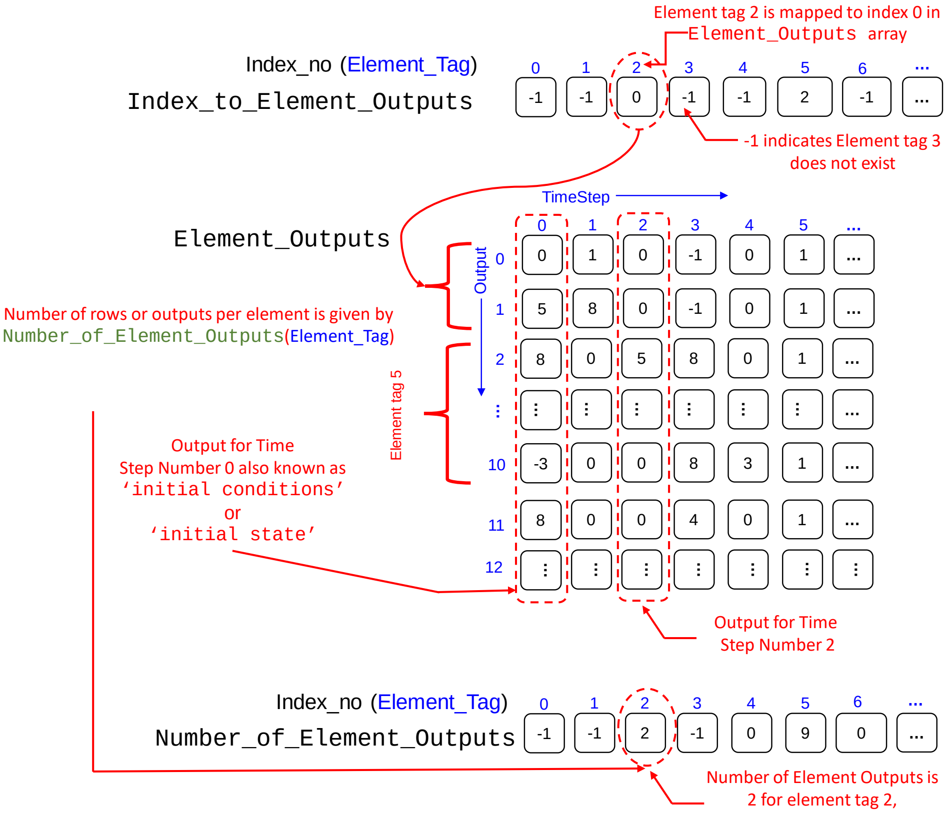


Figure 206.17: Arrays describing output information in `Elements` group directory of HDF5 file.

Figure 206.17 shows how to read `Element_Outputs` array for a particular element and particular time step. In the given example Listing 206.2, element tag 2 has two outputs. Similarly, element tag 5 has 9 outputs. On the other hand element tag 4 has no output. For element 2, the output starts at row index 0 and ends at 1. Similarly, for element tag 4, the output is stored in row index $2..10$.

Since the `Element_Outputs` array format depends on the elements present in the model, one must refer to each element specifically (Section 206.8) to identify what each output component means.

206.5.5 The `Physical_Groups` group

`Physical_Groups` group contains the physical groups of nodes or elements defined in the analysis. It contains two subgroups: `Physical_Node_Groups` and `Physical_Element_Groups` as shown in Figure 206.18 and is described in the following sections.

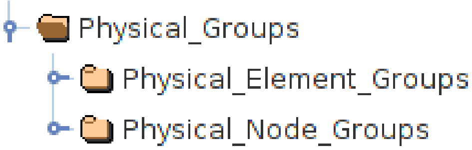


Figure 206.18: `Physical_Groups` group directory of HDF5 file.

For example, lets imagine that the user has defined one phyical group of nodes and elements respectively as shown below :

Listing 206.3: PhyGrp_Example

```
// defining physical groups
define physical_node_group ``Physical_Node_Group#1'';
define physical_element_group ``Physical_Element_Group#1'';

// adding items to allready defined physical groups
add nodes (1,4,5,7,2,30,42) to physical_node_group ``Physical_Node_Group#1''
add elements (1,4,5,7,2,30,42) to physical_node_group ``Physical_Element_Group#1''
```

The data arrays for the given example would look like the following as shown in the subsections ahead.

206.5.5.1 The `Physical_Element_Groups`

This group contains information about physical groups of elements defined in the analysis. For each of the physical group defined, a new `integer` data array is created inside `Physical_Element_Groups` which stored the element tags belonging to that group. Figure 206.19 shows how to read `Physical_Element_Groups` dataset array for a particular defined physical group.

In the given example Listing 206.3, a physical group array with name ``Physical_Element_Group#1'' is created which contains the list of element tags that were part of that physical group. Figure 206.19 shows that ``Physical_Element_Group#1'' contains element tags 1,4,5,7, and so on;

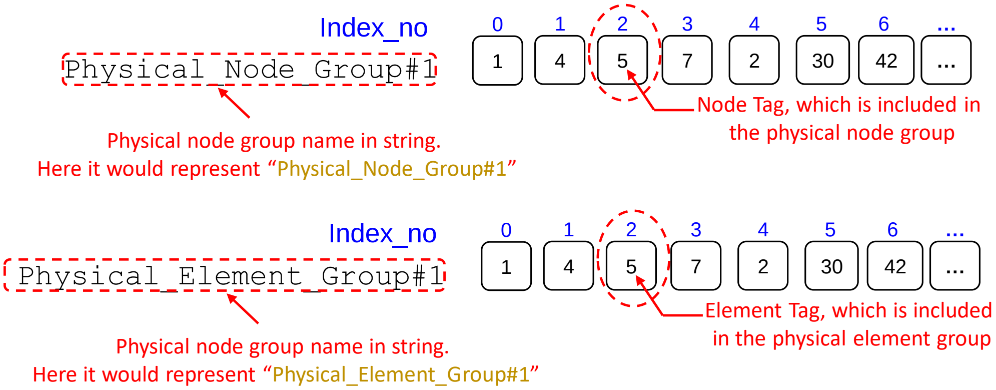


Figure 206.19: Arrays describing physical groups information in `Physical_Groups` group directory of HDF5 file.

206.5.5.2 The `Physical_Node_Groups`

This group contains information about physical groups of nodes defined in the analysis. For each of the physical group defined, a new `integer` data array is created inside `Physical_Node_Groups` which stored the node tags belonging to that group. Figure 206.19 shows how to read `Physical_Node_Groups` dataset array for a particular defined physical group.

In the given example Listing 206.3, a physical group array with name ``Physical_Node_Group#1'' is created which contains the list of node tags that were part of that physical group. Figure 206.19 shows that ``Physical_Node_Group#1'' contains node tags 1,4,5,7, and so on;

206.5.6 The `Eigen_Mode_Analysis` group

`Eigen_Mode_Analysis` group gets created in HDF5 .feioutput file after an eigen mode analysis. The data arrays available inside this group are described below and is also shown in Figure 206.20.

206.5.6.1 `Number_of_Eigen_Modes`

`Number_of_Modes` is an `integer` that stores information about the number of modes solved for the eigen problem. Figure 206.21 shows how to read it.

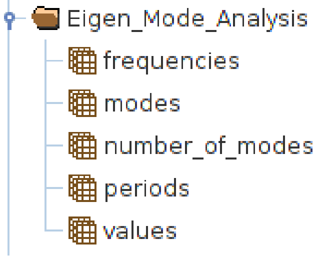
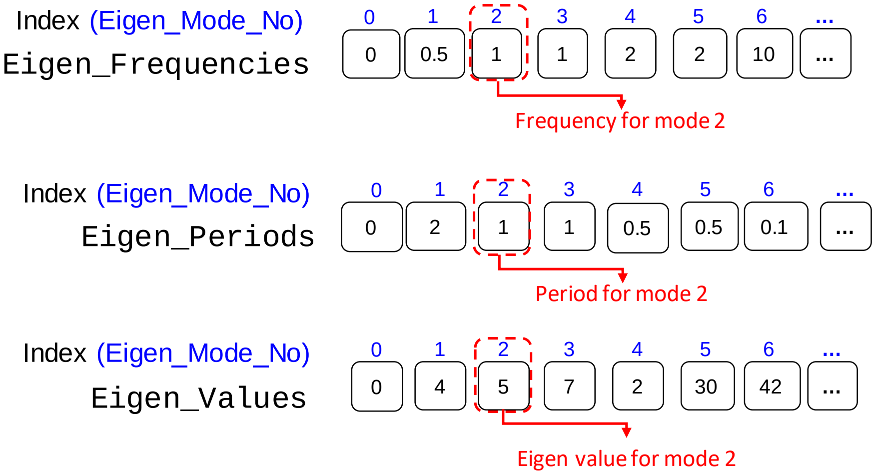


Figure 206.20: `Eigen_Mode_Analysis` group directory of HDF5 file.

206.5.6.2 `Eigen_Frequencies`

`Frequencies` is an `float` array that stores the natural frequencies of the model corresponding to different modes. The length of the array is number of modes + 1. Index '0' does not correspond to any eigen mode and thus stores -1. Figure 206.21 shows how to read it.



Index (Eigen_Mode_No)	0	1	2	3	4	5	6	...
Eigen_Frequencies	0	0.5	1	1	2	2	10	...

Frequency for mode 2

Index (Eigen_Mode_No)	0	1	2	3	4	5	6	...
Eigen_Periods	0	2	1	1	0.5	0.5	0.1	...

Period for mode 2

Index (Eigen_Mode_No)	0	1	2	3	4	5	6	...
Eigen_Values	0	4	5	7	2	30	42	...

Eigen value for mode 2

Figure 206.21: `Eigen_Mode_Analysis` group directory of HDF5 file.

206.5.6.3 `Eigen_Periods`

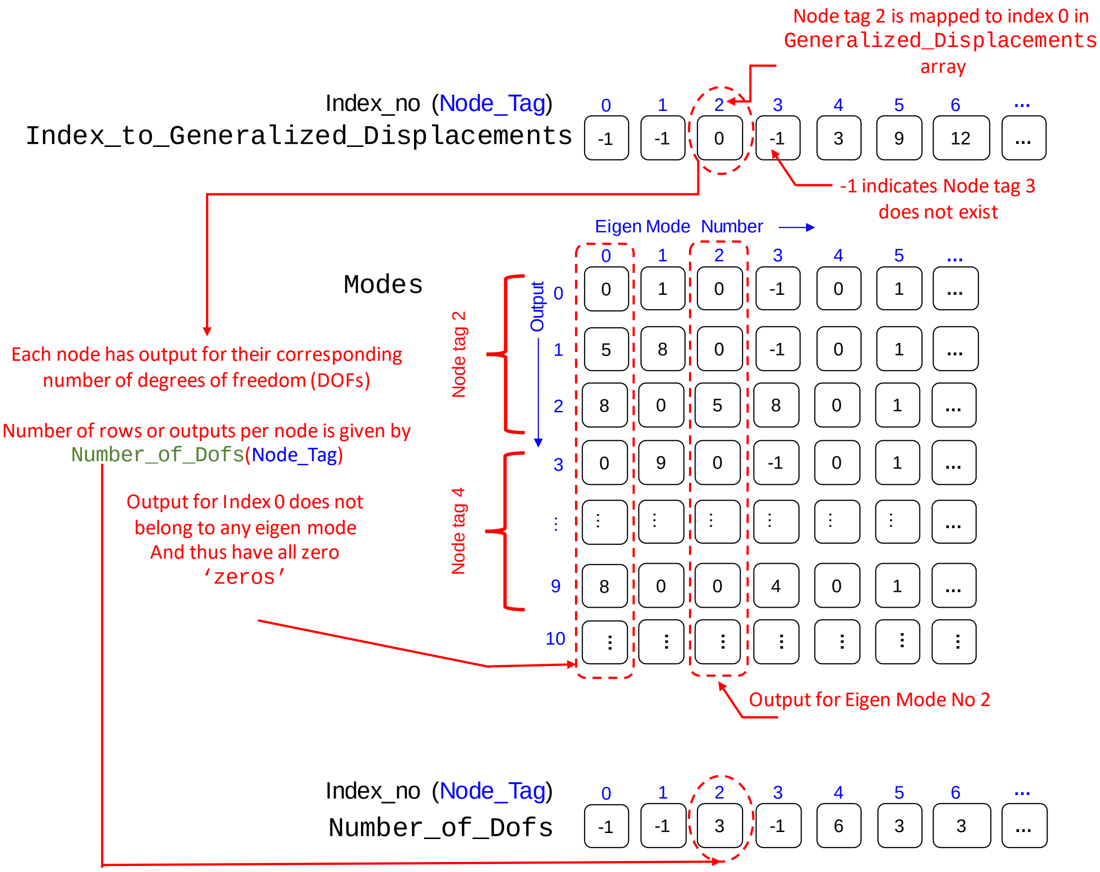
`Frequencies` is an `float` array that stores the natural periods of the model corresponding to different modes. The length of the array is number of modes + 1. Index '0' does not correspond to any eigen mode and thus stores -1. Figure 206.21 shows how to read it.

206.5.6.4 `Eigen_Values`

`Frequencies` is an `float` array that stores the natural eigen values of the model corresponding to different modes. The length of the array is number of modes + 1. Index '0' does not correspond to any eigen mode and thus stores -1. Figure 206.21 shows how to read it.

206.5.6.5 `Modes`

`Modes` is a 2-D `float` array that stores the generalized displacements of the nodes defined in the model corresponding to different modes. The column index no. n this 2-D array defines the mode no i.e. column index 1 corresponds to mode number 1 and so on. Figure ?? shows how to read it.



Index_no (Node_Tag)	0	1	2	3	4	5	6	...
Index_to_Generalized_Displacements	-1	-1	0	-1	3	9	12	...

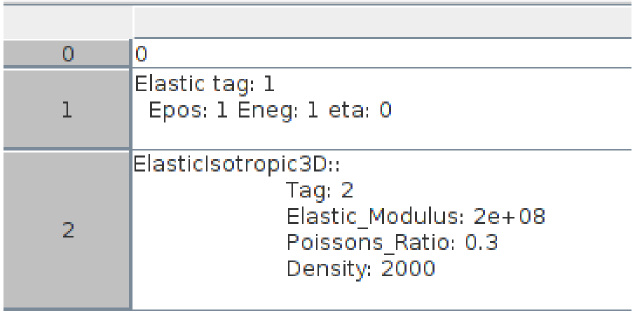
Output	0	1	2	3	4	5	...
0	0	1	0	-1	0	1	...
1	5	8	0	-1	0	1	...
2	8	0	5	8	0	1	...
3	0	9	0	-1	0	1	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	...
9	8	0	0	4	0	1	...
10	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Index_no (Node_Tag)	0	1	2	3	4	5	6	...
Number_of_Dofs	-1	-1	3	-1	6	3	3	...

Figure 206.22: `Eigen_Mode_Analysis` group directory of HDF5 file.

206.5.7 The `Material` data array

Material is a 1-D `string` array that stores information about all materials defined in the model. The length of the array is the maximum material tag + 1 (including tag 0). The index of this array corresponds to the `material tag`. A value of '-1' means that the material tag was not defined. Figure 206.23 shows the `Material` data array for the example Listing 206.2. Here, index no 1 and index no 2 stores the information corresponding to material tag 1 and 2 respectively.



0	0
1	Elastic tag: 1 Epos: 1 Eneg: 1 eta: 0
2	ElasticIsotropic3D:: Tag: 2 Elastic_Modulus: 2e+08 Poissons_Ratio: 0.3 Density: 2000

Figure 206.23: `Materials` data array in `Model` directory of HDF5 file.

206.6 Node-specific output format

In Real-ESSI simulator, nodes can be defined with different number of degree of freedoms (DOFs). Nodes with with different dofs can be thought of different types of nodes. As a result, their corresponding output also changes. Here is described, all the different node types available and the output format and their descriptions for each of them. The following subsections would describe the definition of outputs that are expected in `Generalized_Displacements` and `Support_Reactions` data arrays in `Nodes` group of HDF5 output file for node tags of different dof types.

206.6.1 3DOF

Nodes defined with 3DOF type has u_x, u_y, u_z degrees of freedom. They correspond to the displacement degrees of freedom in x, y and z direction respectively. The dof id, generalized_displacement and support reactions for 3dof type node are summarized in the Table 206.6.1.

DOF Id	Description	`Generalized_Displacements`	`Support_Reactions`
0	disp. in x-dir	$u_x[m]$	$F_x[N]$
1	disp. in y-dir	$u_y[m]$	$F_y[N]$
2	disp. in z-dir	$u_z[m]$	$F_z[N]$

Table 206.3: DOF id and output for 3DOF type node

206.6.2 4DOF

Nodes defined with 3DOF type has u_x, u_y, u_z, p degrees of freedom. They correspond to the displacement degrees of freedom in x, y and z direction and pressure respectively. The dof id, generalized_displacement and support reactions for 4dof type node are summarized in the Table 206.6.2. up elements have nodes with 4 dofs.

DOF Id	Description	`Generalized_Displacements`	`Support_Reactions`
0	disp. in x-dir	$u_x[m]$	$F_x[N]$
1	disp. in y-dir	$u_y[m]$	$F_y[N]$
2	disp. in z-dir	$u_z[m]$	$F_z[N]$
3	pressure	$p[Pa]$	$p[Pa]$

Table 206.4: DOF id and output for 4DOF type node

206.6.3 6DOF

Nodes defined with 6DOF type has $u_x, u_y, u_z, r_x, r_y, r_z$ degrees of freedom. They correspond to the displacement and rotational degrees of freedom in x, y and z direction respectively. The dof id, generalized_displacement and support reactions for 6 dof type node are summarized in the Table 206.6.3. Beam and Shell elements have nodes with 6 dofs.

DOF Id	Description	`Generalized_Displacements`	`Support_Reactions`
0	disp. in x-dir	$u_x[m]$	$F_x[N]$
1	disp. in y-dir	$u_y[m]$	$F_y[N]$
2	disp. in z-dir	$u_z[m]$	$F_z[N]$
3	rotation about x-axis	$r_x[radian]$	$M_x[N-m]$
4	rotation about y-axis	$r_y[radian]$	$M_y[N-m]$
5	rotation about z-axis	$r_z[radian]$	$M_z[N-m]$

Table 206.5: DOF id and output for 6DOF type node

206.6.4 7DOF

Nodes defined with 6DOF type has $u_x, u_y, u_z, p, U_x, U_y, U_z$ degrees of freedom. They correspond to the solid-displacement , pore-fluid pressure and and fluid-displacement in x, y and z direction respectively. The dof id, generalized_displacement and support reactions for 7dof type node are summarized in the Table 206.6.3. upU elements have nodes with 7 dofs.

DOF Id	Description	`Generalized_Displacements`	`Support_Reactions`
0	solid disp. in x-dir	$u_x[m]$	$F^s{}_x[N]$
1	solid disp. in y-dir	$u_y[m]$	$F^s{}_y[N]$
2	solid disp. in z-dir	$u_z[m]$	$F^s{}_z[N]$
3	pore-pressure	$p[Pa]$	$p[Pa]$
4	fluid disp. in x-dir	$U_x[m]$	$F^f{}_x[N]$
5	fluid disp. in y-dir	$U_y[m]$	$F^f{}_y[N]$
6	fluid disp. in z-dir	$U_z[m]$	$F^f{}_z[N]$

Table 206.6: DOF id and output for 7DOF type node

206.7 Element-gauss output format

`Gauss_Outputs` array stores the coordinates of the gauss points declared inside for each element defined in the model. It is a 2D `float` array indexed by the `Index_to_Gauss_Outputs` array and the number of rows is defined by 18 (6 total strain, 6 plastic strain and 6 stress) times `Number_of_Gauss_Points`. The column index is represented by the time step of the simulation. Time index 0 represents initial state conditions, i.e. the state before the start of new stage and end of previous stage.

Figure 206.24 shows how to read `Gauss_Outputs` array for a particular element. In the given example Listing 206.2, element tag 2 and 5 has no gauss points. Since, element tag 4 has 8 gauss points, the total length of gauss output for that element is $8 \times 18 = 144$. The index from which the coordinates information start is 0. gauss values for first 18 index $(0, 1..17)$ corresponds to gauss point 1 and next 18 index $(18..35)$ corresponds to gauss point 2 and so on. For gauss point 1, first 6 index $(0, 1..5)$ corresponds to total strain, next 6 index $(6..11)$ corresponds to plastic strain and next 6 index $(12..17)$ corresponds to stress. Table 206.7 shows how data is stored for each gauss points. The strains are unit less and stress have unit of Pascal Pa.

In the Table 206.7 `start` corresponds to the starting position for the elements output as determined with reference to the `Index_to_Element_Outputs` for the element of interest. To this number we add the corresponding `offset` as determined by each table below and interpret the row according to the meaning established below.

For each gauss outputs there are 6 components of strain tensor, 6 components of plastic-strain tensor and 6 components of stress tensor. This makes a total of $3 \times 6 \times NumGauss$ rows of output per element. The specific meaning of the rows is as follows. For Gauss-point 1 (with starting index given in the `Index_to_Gauss_Outputs` array), the outputs are stored as the following

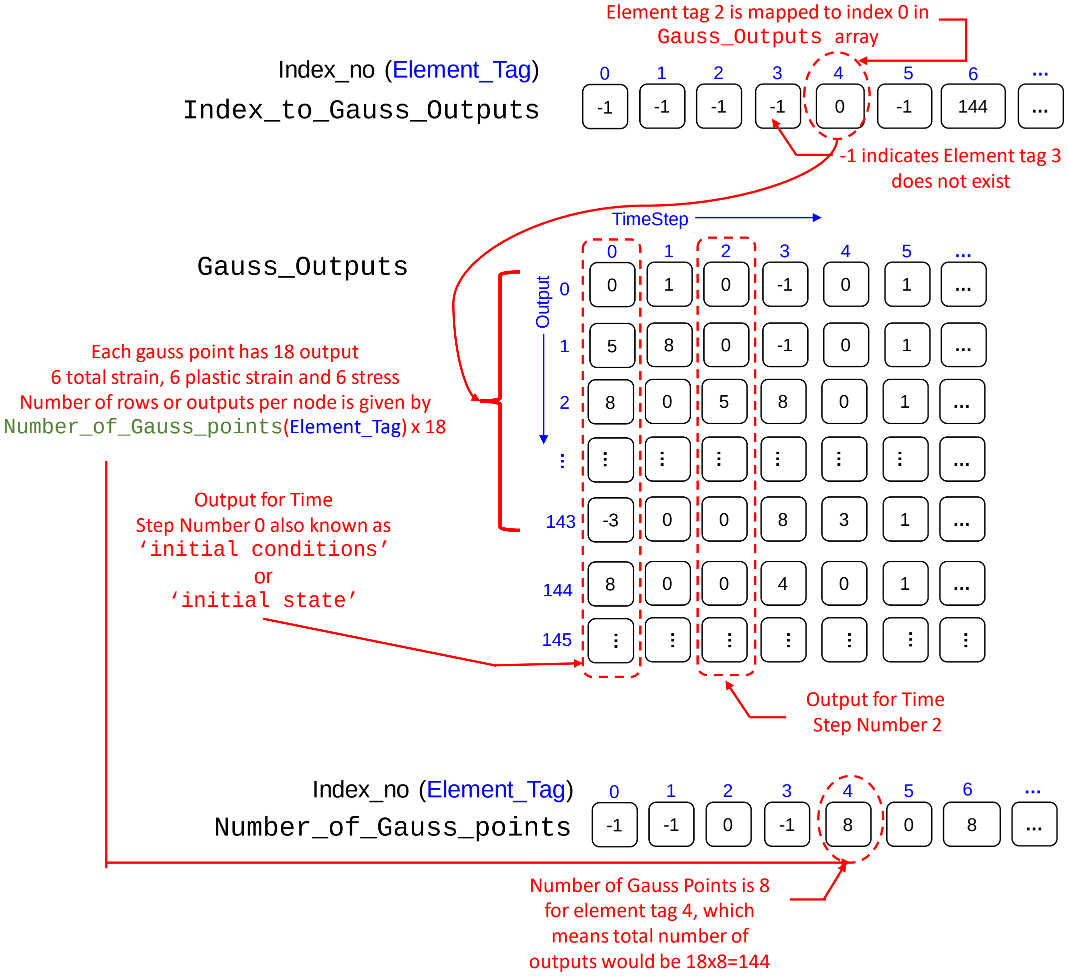


Figure 206.24: Arrays describing gauss output information in `Elements` group directory of HDF5 file.

Position (`start+offset`)	Content meaning	Position (`start+offset`)	Content meaning	Position (`start+offset`)	Content meaning
`start+0`	$\epsilon_{xx}[]$	`start+6`	$\epsilon^{\text{plastic}}_{xx}[]$	`start+12`	$\sigma_{xx}[Pa]$
`start+1`	$\epsilon_{yy}[]$	`start+7`	$\epsilon^{\text{plastic}}_{yy}[]$	`start+13`	$\sigma_{yy}[Pa]$
`start+2`	$\epsilon_{zz}[]$	`start+8`	$\epsilon^{\text{plastic}}_{zz}[]$	`start+14`	$\sigma_{zz}[Pa]$
`start+3`	$\epsilon_{xy}[]$	`start+9`	$\epsilon^{\text{plastic}}_{xy}[]$	`start+15`	$\sigma_{xy}[Pa]$
`start+4`	$\epsilon_{xz}[]$	`start+10`	$\epsilon^{\text{plastic}}_{xz}[]$	`start+16`	$\sigma_{xz}[Pa]$
`start+5`	$\epsilon_{yz}[]$	`start+11`	$\epsilon^{\text{plastic}}_{yz}[]$	`start+17`	$\sigma_{yz}[Pa]$

Table 206.7: Output Format for each gauss point defined inside element

Gauss-point 2 will then start at position 18 through 36 with the same meaning for each row. And so-on for the other Gauss-points.

206.8 Element-specific output format

`Element_Outputs` array stores output for the elements except those stored at gauss points (i.e. stress, total strain and plastic strain). It is a 2D `float` array indexed by the `Index_to_Element_Outputs` array and the number of rows is defined `Number_of_Element_Outputs`. he column index is represented by the time step of the simulation. Time index 0 represents initial state conditions, i.e. the state before the start of new stage and end of previous stage.

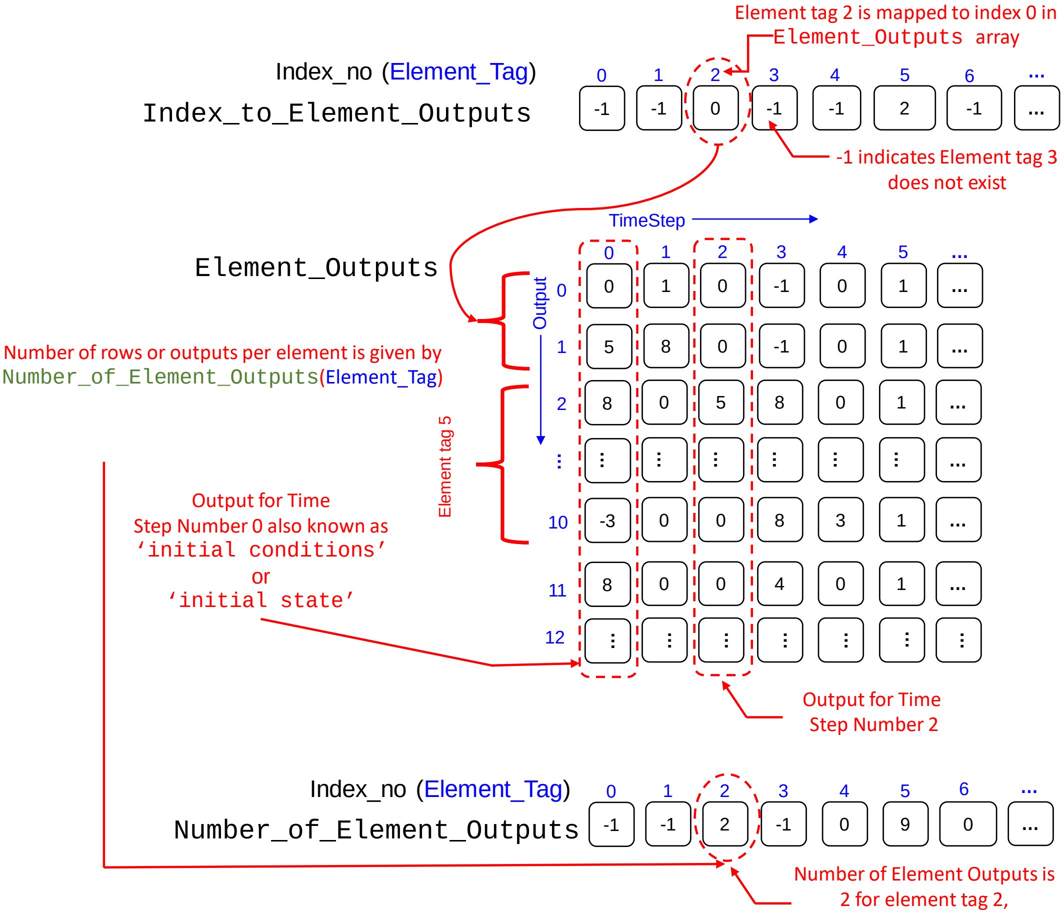


Figure 206.25: Arrays describing output information in `Elements` group directory of HDF5 file.

Figure 206.25 shows how to read `Element_Outputs` array for a particular element and particular time step. In the given example Listing 206.2, element tag 2 has two outputs. Similarly, element tag 5 has 9 outputs. On the other hand element tag 4 has no output. For element 2, the output starts at row index 0 and ends at 1. Similarly, for element tag 4, the output is stored in row index $2..10$.

Since the `Element_Outputs` array format depends on the elements present in the model, one must refer to each element specifically. Each element writes information into the `Element_Outputs` array in a different way. The user can determine the rows of the `Element_Outputs` array that belong to a given element by looking into the `Index_to_Element_Outputs` array which relates element tag to the starting position of that element's output within the `Element_Outputs` array. Additionally, the `Number_of_Element_Outputs` tells the user how many rows after the starting position correspond to the given element output.

The actual meaning of each row is element dependent and is detailed in the following pages. Please note that elements not in this list have no output defined at the moment.

In what follows `start` corresponds to the starting position for the elements output as determined with reference to the `Index_to_Element_Outputs` for the element of interest. To this number we add the corresponding `offset` as determined by each table below and interpret the row according to the meaning established below.

206.8.1 Truss

This element outputs 2 rows in total. First row is the uniaxial change in length while second component is the axial force. Truss does not have any gauss points and thus do not have any gauss outputs. The description of each of these rows are shown in Table 206.8:

Position (`start+offset`)	Content meaning
`start+0`	$\Delta L[m]$
`start+1`	$Force F[N]$

Table 206.8: Element Output description for Truss

206.8.2 Brick Elements

All the bricks (u, up, upU and variable) have 0 element outputs. But, they do output total strain, plastic strain and stress for all their corresponding number of gauss points involved in elasto-plastic integration. The format in which gauss outputs are stored for each gauss point is described in Section 206.8

206.8.3 ShearBeam

Like brick elements, Shear Beam element also does not have any element output. But it does output for the *one* gauss point it has. Thus, in total it has 18 outputs for the only one gauss point. The format in which gauss outputs are stored for each gauss point is described in Section 206.8

206.8.4 ElasticBeam

This element outputs 6 components of local nodal displacements at each of its two nodes, and 6 components of end forces at each of its two nodes. Total number of outputs is thus $2 \times 6 \times 2 = 24$ rows per element. The description of each of these rows are shown in Table 206.9:

Position (`start+offset`)	Content meaning	Position (`start+offset`)	Content meaning
`start+0`	$u_x^{\text{local},1}[m]$	`start+6`	$u_x^{\text{local},2}[m]$
`start+1`	$u_y^{\text{local},1}[m]$	`start+7`	$u_y^{\text{local},2}[m]$
`start+2`	$u_z^{\text{local},1}[m]$	`start+8`	$u_z^{\text{local},2}[m]$
`start+3`	$\theta_x^{\text{local},1}[radian]$	`start+9`	$\theta_x^{\text{local},2}[radian]$
`start+4`	$\theta_y^{\text{local},1}[radian]$	`start+10`	$\theta_y^{\text{local},2}[radian]$
`start+5`	$\theta_z^{\text{local},1}[radian]$	`start+11`	$\theta_z^{\text{local},2}[radian]$
Position (`start+offset`)	Content meaning	Position (`start+offset`)	Content meaning
`start+12`	$F_x^{\text{local},1}[N]$	`start+18`	$F_x^{\text{local},2}[N]$
`start+13`	$F_y^{\text{local},1}[N]$	`start+19`	$F_y^{\text{local},2}[N]$
`start+14`	$F_z^{\text{local},1}[N]$	`start+20`	$F_z^{\text{local},2}[N]$
`start+15`	$M_x^{\text{local},1}[N-m]$	`start+21`	$M_x^{\text{local},2}[N-m]$
`start+16`	$M_y^{\text{local},1}[N-m]$	`start+22`	$M_y^{\text{local},2}[N-m]$
`start+17`	$M_z^{\text{local},1}[N-m]$	`start+23`	$M_z^{\text{local},2}[N-m]$

Table 206.9: Element Output description for Elastic Beam

206.8.5 4NodeShell_ANDES

This element currently does not have any gauss or element outputs.

206.8.6 BeamColumnDispFiber3d

This element outputs 6 components of end forces at each of its two nodes. Each row is described in Table 206.10:

Position (`start+offset`)	Content meaning	Position (`start+offset`)	Content meaning
`start+0`	$F_x^{\text{local},1}[N]$	`start+6`	$F_x^{\text{local},2}[N]$
`start+1`	$F_y^{\text{local},1}[N]$	`start+7`	$F_y^{\text{local},2}[N]$
`start+2`	$F_z^{\text{local},1}[N]$	`start+8`	$F_z^{\text{local},2}[N]$
`start+3`	$M_x^{\text{local},1}[N-m]$	`start+9`	$M_x^{\text{local},2}[N-m]$
`start+4`	$M_y^{\text{local},1}[N-m]$	`start+10`	$M_y^{\text{local},2}[N-m]$
`start+5`	$M_z^{\text{local},1}[N-m]$	`start+11`	$M_z^{\text{local},2}[N-m]$

Table 206.10: Element Output description for displacement based fiber beams

206.8.7 Force Based Contact/Interface Elements

This element outputs 9 components: 3 components of gap displacement and 3 components of contact/interface forces and 3 components of incremental slip in the local axis definition. The first two components of gap are transverse components (g_{t1}, g_{t2}), while the third is the normal gap component g_n. Similarly, the first two components of force (F_{t1}, F_{t2}) are transverse (shear on contact/interface plane) while the third is the normal contact/interface force F_n. The last three components are incremental slip $\Delta g^{inc_s lip}{}_1, \Delta g^{inc_s lip}{}_2$ in the local transverse direction and total uplift Δn in local normal vector direction. If $\Delta n > 0$, there is uplift i.e. loss of contact/interface else it is in contact. Each row is described in Table 206.12:

Position (`start+offset`)	Content meaning	Position (`start+offset`)	Content meaning	Position (`start+offset`)	Content meaning
`start+0`	$g_{t1}[m]$	`start+3`	$F_{t1}[N]$	`start+6`	$\Delta g^{inc_s lip}{}_1[m]$
`start+1`	$g_{t2}[m]$	`start+4`	$F_{t2}[N]$	`start+7`	$\Delta g^{inc_s lip}{}_2[m]$
`start+2`	$g_n[m]$	`start+5`	$F_n[N]$	`start+8`	$Uplift \Delta g_n[m]$

Table 206.11: Element Output description for force based contact/interface elements

206.8.8 Stress Based Contact/Interface Elements

This element outputs 9 components: 3 components of gap displacement and 3 components of contact/interface forces and 3 components of incremental slip in the local axis definition. The first two components of gap are transverse components (g_{t1}, g_{t2}), while the third is the normal gap component g_n. Similarly, the first two components of stress $(\sigma_{t1}, \sigma_{t2})$ are transverse (shear on contact/interface plane) while the third is the normal contact/interface stress σ_n. The last three components are incremental slip $\Delta g^{inc_s lip}{}_1, \Delta g^{inc_s lip}{}_2$ in the local transverse direction and total uplift Δn in local normal vector direction. If $\Delta n > 0$, there is uplift i.e. loss of contact/interface else it is in contact. Each row is described in Table 206.12:

Position (`start+offset`)	Content meaning	Position (`start+offset`)	Content meaning	Position (`start+offset`)	Content meaning
`start+0`	$g_{t1}[m]$	`start+3`	$\sigma_{t1}[N/m^2]$	`start+6`	$\Delta g^{inc_s lip}{}_1[m]$
`start+1`	$g_{t2}[m]$	`start+4`	$\sigma_{t2}[N/m^2]$	`start+7`	$\Delta g^{inc_s lip}{}_2[m]$
`start+2`	$g_n[m]$	`start+5`	$\sigma_n[N/m^2]$	`start+8`	$Uplift \Delta g_n[m]$

Table 206.12: Element Output description for stress based contact/interface elements

206.9 Energy Output Format

`Energy` folder in the output file stores simulation results for energy-related quantities. Energy density quantities are calculated at element integration points. Averaged energy density quantities are calculated for each element. Energy quantities are calculated for each element. Nodal input energy values are calculated at each node. All energy terms are calculated and stored at each time step.

Under the `Energy` folder, there are currently seven datasets. Four of them contain energy calculation results. The other three are index datasets, which can be used to find energy output for specific elements or nodes.

In general, energy output datasets are 2D float arrays. The column index is represented by the time step of the simulation. The row index represents various energy quantities that are calculated and stored. The three index datasets are used to locate the row index for specific energy quantities.

206.9.1 Input Energy

All types of nodal load, including DRM, are applied at nodes. Elemental loads are automatically converted to nodal loads, then also applied at nodes. Therefore, input energy or input work from all types of external load can be calculated at each node.

Under the `Energy` folder, the `Nodal_Input_Energy` dataset stores input energy at each node. Note that this is input energy accumulated from the start of simulation to the current time step. To find the input energy time history at a specific node, find the index in the `Index_to_Nodal_Input_Energy` dataset then find the corresponding entry in the `Nodal_Input_Energy` dataset.

For example, here are the steps to get the input energy time history at the node with node tag 73, Go to `Index_to_Nodal_Input_Energy`, find the index for node 73 to be 105. Then the accumulated input energy at your chosen node is stored at row 105 in `Nodal_Input_Energy`.

206.9.2 Energy Density Quantity at Gauss Point

`Energy_Density_GP` dataset stores energy density quantities at each Gauss point. To find the energy density time history at a specific Gauss point, find the index in the `Index_to_Energy_Density_GP` dataset then find the corresponding entry in the `Energy_Density_GP` dataset.

For each Gauss point, 12 data slots are occupied in the following order:

- Incremental kinetic energy density
- Accumulated kinetic energy density
- Incremental strain energy density

- Accumulated strain energy density
- Incremental plastic free energy density
- Accumulated plastic free energy density
- Incremental plastic dissipation density
- Accumulated plastic dissipation density
- The last 4 slots are currently empty, reserved for potential future use

Note that "incremental" means the change of energy density during the time step, "accumulated" means current cumulative energy density at the time step.

Here is an example showing how to obtain the energy density evolution at a specific Gauss point. Say you are interested in looking at the accumulated plastic dissipation density at the third Gauss point of an `8NodeBrick` element with element tag 73. Go to `Index_to_Energy_Density_GP`, find the index for element 73 to be 3904. Since each Gauss point occupies 12 slots in `Energy_Density_GP`, the energy outputs for the third Gauss point starts at index 3904+(3-1)*12=3928. Finally, since accumulated plastic dissipation density is the 8th entry among the 12 slots, the row index for the data of your interest is 3928+(8-1) = 3935. This means that the accumulated plastic dissipation density at your chosen location is stored at row 3935 of `Energy_Density_GP`.

206.9.3 Average Energy Density Quantity for Element

`Energy_Density_Element_Average` dataset stores averaged energy density quantities of each element. To find the energy density time history of a specific element, find the index in the `Index_to_Energy_Element` dataset then find the corresponding entry in the `Energy_Density_Element_Average` dataset.

For each element, 12 data slots are occupied in the following order:

- Incremental kinetic energy density
- Accumulated kinetic energy density
- Incremental strain energy density
- Accumulated strain energy density
- Incremental plastic free energy density
- Accumulated plastic free energy density

- Incremental plastic dissipation density
- Accumulated plastic dissipation density
- The last 4 slots are currently empty, reserved for potential future use

Note that "incremental" means the change of energy density during the time step, "accumulated" means current cumulative energy density at the time step.

Here is an example showing how to obtain the energy density evolution of a specific element. Say you are interested in looking at the accumulated plastic dissipation density of an `8NodeBrick` element with element tag 73. Go to `Index_to_Energy_Element`, find the index for element 73 to be 612. Since accumulated plastic dissipation density is the 8th entry among the 12 slots, the row index for the data of your interest is 612+(8-1) = 619. This means that the accumulated plastic dissipation density of your chosen element is stored at row 619 of `Energy_Density_Element_Average`.

206.9.4 Energy Quantity for Element

`Energy_Element` dataset stores averaged energy density quantities of each element. To find the energy time history of a specific element, find the index in the `Index_to_Energy_Element` dataset then find the corresponding entry in the `Energy_Element` dataset.

For each element, 12 data slots are occupied in the following order:

- Incremental kinetic energy
- Accumulated kinetic energy
- Incremental strain energy
- Accumulated strain energy
- Incremental plastic free energy
- Accumulated plastic free energy
- Incremental plastic dissipation
- Accumulated plastic dissipation
- Incremental viscous energy dissipation
- Accumulated viscous energy dissipation
- The last 2 slots are currently empty, reserved for potential future use

Note that "incremental" means the change of energy density during the time step, "accumulated" means current cumulative energy density at the time step.

Here is an example showing how to obtain the energy evolution of a specific element. Say you are interested in looking at the accumulated plastic dissipation of an `8NodeBrick` element with element tag 73. Go to `Index_to_Energy_Element`, find the index for element 73 to be 612. Since accumulated plastic dissipation is the 8th entry among the 12 slots, the row index for the data of your interest is 612+(8-1) = 619. This means that the accumulated plastic dissipation of your chosen element is stored at row 619 of `Energy_Element`.

Chapter 207

Real-ESSI Pre Processing and Model Development Methods

(2010-2015-2017-2020-2021-)

(In collaboration with Prof. Sumeet Kumar Sinha, Dr. Hexiang Wang and Dr. Yuan Feng)

207.1 Introduction

207.2 Model Development Using gmsh

207.2.1 Introduction to gmESSI

The gmESSI, pronounced as *[gm-ESSI]*, is a translator that converts mesh file from gmsh (a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities) to Real-ESSI DSL format. The primary aim of this program is to provide an efficient pre-processing tool to develop Finite Element (FE) models in gmsh and make them interface with various Real-ESSI functionalities. The gmESSI translator package contains the translator, sublime plugin and the manual.

The gmESSI package is available at `http://sokocalo.engr.ucdavis.edu/~jeremic/lecture_notes_online_material/_Chapter_SoftwareHardware_Pre_Processing/Real-ESSI_gmESSI.tgz`.

The text editor sublime plugin *[gmESSI-Tools]* can be downloaded here: `http://sokocalo.engr.ucdavis.edu/~jeremic/lecture_notes_online_material/_Chapter_SoftwareHardware_Pre_Processing/fei-syntax-n-snippets.tar.gz`.

207.2.1.1 Getting Started

The translator utilizes the physical and entity group concept of *Gmsh* (`http://geuz.org/gmsh/doc/texinfo/gmsh.html`) (Geuzaine and Remacle, 2009), which gets imprinted in the mesh ".msh" file. The translator then manipulates these groups to convert the whole mesh to ESSI commands. Thus, making physical groups is the essential, key for conversion. The Translator basically provides some strict syntax for naming these Physical Groups which provides gmESSI information about the elements or (nodes) on which the translation operates. The translator is made so general that any other FEM program can use it with little tweaks to have their own conversion tool. A quick look at some important features of the program are:

- It has a lot of predefined commands which do the conversion at the blink of an eye. These commands make it easier to define elements, boundary conditions, contacts/interfaces, fixities, loads ....
- It provides a python module "gmessi". The users can import this modulue and can extend the functional capability of gmESSI.
- The *[gmESSI-Tools]* sublime plugin makes it easy by providing syntax coloring and auto-text-completion for gmESSI commands. *[gmsh-Tools]* sublime plugin can also be installed for gmsh syntax coloring and auto-completion.

- The translator uses a *mapping.fei* file to check for its command syntax and conversion. A user can easily add a command in *mapping.fei* and it would get reflected automatically in the translation.
- It automatically optimizes the Real-ESSI tags (node, element, load) for space and time efficiency while running simulation.

Installation Process: The Translator have its dependencies on Octave (3.2 or higher), Boost(1.58 or higher), (Python 2.7 or higher). One should make sure to have them before compiling it. On Linux Ubuntu distros the dependencies can be installed as

```
sudo apt-get install liboctave-dev
# Boost version should be higher than 1.48
sudo apt-get install libboost-all-dev
sudo apt-get install python-dev
```

Installation of the gmESSI translator is easy, just follow steps below.

```
## go to folder where you want to store and build gmESSI application

## download the package from main Real-ESSI repository
## this line below should be all one line
## HOWEVER it had to be broken in two lines to be readable
## so please make a single command out of two lines below
##
#
# using curly brackets to help in checking scripts, that rely on these
# brackets being available around URL
#
wget {http://sokocalo.engr.ucdavis.edu/~jeremic/lecture_notes_online_material/\
_Chapter_SoftwareHardware_Pre_Processing/Real-ESSI_gmESSI.tgz}
##
# make directory, move files, expand archive
mkdir Real-ESSI-gmESSI
mv _all_files_gmESSI_.tgz Real-ESSI-gmESSI
cd Real-ESSI-gmESSI
tar -xvzf _all_files_gmESSI_.tgz

## build the package
make # builds the application in curr_dir/build

## install the package
# -- by default the pacakage is installed in /usr/local
make install # installs the package in /usr/local
# -- to change the install directory
make install INSTALL_DIR=install_dir_specified_by_user

############## For installation of gmESSI plugin in sublime ##############
# open sublime-text
```

```
# make sure you have installed package control
# if not then install it first from
# {https://packagecontrol.io/installation}
# go to Preferences->PackageControl->InstallPackage
# search for [gmESSI-Tools] and install it
# also install [gmsh-Tools] following the same steps
# restart sublime
############################################################################
```

207.2.1.2 Running gmESSI

gmESSI can be invoked from the bash terminal by typing *gmessy*. It can take one or multiple *xyz.gmessi* files as an argument and convert them to Real-ESSI files in their respective simulation directory defined by the user. By default, the 'gmessi' python module is automatically imported and available as 'gmESSI' in '.gmeesi' input file.

gmessy is a top level python script that parses the .gmessi file and categories commands in the following order as

- gmESSI Command: gmESSI Commands are one line commands. They start and end with '[' and ']' respectively. Section 207.2.3 describes the syntax.
- gmESSI Comments: The lines that start with '//' are considered as gmESSI comments. It gets translated and copied to the main file (See Section 207.2.4.6).
- Singular Commands: The lines that start with '!' are directly copied to the main file (See Section 207.2.4.6 and Section 207.2.5.1). Real-ESSI domain specific language (DSL) are written following the exclamation mark '!' sign.
- Python Comments: The lines that start with '#' are considered as python comments.
- Python Commands: Whatever lines left are considered as python commands. This option is only for the advanced user and is not documented to make the manual simple. Only some useful commands required are explained in the manual.

The categorized commands then generates an equivalent python (.py) script, which gets finally run in python interpretor. The generated equivalent python script can be seen by adding '-l' or '–logfile= LOG_FILE' option during execution. It is important to note that nodes, coordinates, element no etc generated from the translator have a precision associated with them. By default the precision is up-to '6' significant digits. The user can change the precision anywhere in the .gmessi file as

```
gmESSI.setPrecision(10);
```

This will set the precision to ‘10’ significant digits. Lowering precision can be helpful in generating same coordinates for contact/interface node pairs. See [Example_4.gmessi] for its usage.

The full description of *gmessy* can be invoked from the terminal as

```
$gmessy --help

usage: gmessy [-h] [-l] [-nm] [-em] [-ne] [--logfile= LOG_FILE]
[--nodemap= NM_FILE] [--elemap= ELM_FILE]
gmessi_filename

positional arguments:
gmessi_filename filename containing semantics of conversion

optional arguments:
-h, --help show this help message and exit
-l generate the log file at the current location
-nm generate the node map file at the current location
-em generate the element map file at the current location
-ne don't carry out the conversion
--logfile= LOG_FILE generate the log file at specified location
--nodemap= NM_FILE generate the node-map (gmsh-to-Real_ESSI) file at specified ↩
    location
--elemap= ELM_FILE generate the element-map (gmsh-to-Real_ESSI) file at ↩
    specified location
```

Since gmESSI optimizes the ‘node’ and ‘element’ tag for Real-ESSI, it provides an interface to retrieve the node map and element map containing mapping from gmsh_tag to Real_ESSI_Tag.

Running gmESSI requires, the *.gmessi* input file and the gmsh mesh (.msh) mesh file containing physical groups. Let’s go and run an example to see how gmESSI works. [Example_1] can be obtained here.

Alternatively in the gmESSI directory, navigate to the Examples directory and then to Example_1 directory.

```
$cd ./Examples/Example_1
$ls
Example_1.geo # geometry file [gmsh]
Example_1.msh # mesh file [gmsh]
Example_1.gmessi # gmessi input file [gmESSI]
```

Contents of Example_1.gmessi input file: As described above, the .gmessi input file contains gmESSI commands, singular commands and python commands. Also, it can contain comments followed by // or #.

At the beginning of the input file, the simulation directory, main, node, element, load filenames must be specified. Also, before adding any gmESSI command, mesh must be loaded using 'gmESSI.loadGmshFile' command.

```
$ cat Example_1.gmessi

### loading the msh file
gmESSI.loadGmshFile("Example_1.msh")

### Physical Groups defined in the msh file.
#2 2 "Base_Surface"
#2 3 "Top_Surface"
#3 1 "Soil"

### Defining the Simulation Directory
gmESSI.setSimulationDir("./Example_1_ESSI_Simulation")
gmESSI.setMainFile(gmESSI.SimulationDir+ "main.fei")
gmESSI.setNodeFile(gmESSI.SimulationDir+ "node.fei")
gmESSI.setElementFile(gmESSI.SimulationDir+ "element.fei")
gmESSI.setLoadFile(gmESSI.SimulationDir+ "load.fei")


// My new model
! model name "Soil_Block";

[Add_All_Node{ unit:= m, nof_dofs:= 3}]

// Adding Material layer wise and also assigning it to elements
[Vary_Linear_Elastic_Isotropic_3D{Physical_Group#Soil, ElementCommand:= ↩
    [Add_8NodeBrick{}], Density:= 1600+10*(10-z)\ 0 \kg/m^3, ElasticModulus:= ↩
    20e9+10e8*(10-z)\-8\Pa, PoissonRatio:= 0.3}]

! include "node.fei";
! include "element.fei";
! new loading stage "Stage1_Surface_Loading";

# Applying Fixities
[Fix_Dofs{Physical_Group#Base_Surface, all}]

#### For applying Surface load on the Top Surface of the Soil Block
#[Add_8NodeBrick_SurfaceLoad{Physical_Group#1,Physical_Group#3,10*Pa}]

#### For applying Nodal loads to all the nodes of the top surface
[Add_Node_Load_Linear{Physical_Group#Top_Surface, ForceType:= Fx, Mag:= 10*kN}]

#### For applying Self-Weight Load to the soil elements
! add acceleration field # 1 ax = 0*g ay = 0*g az = -1*g ;
! add load #18 to all elements type self_weight use acceleration field # 1;

# Updating the tag inside gmESSI as user entered by himself load tag
gmESSI.setESSITag("load",19)
```

```

! include "load.fei";
! NumStep = 10;
!
! define algorithm With_no_convergence_check;
! define solver UMFPack;
! define load factor increment 1/NumStep;
! simulate NumStep steps using static algorithm;
! bye;
```

Running Example_1 in Terminal:

```
$ gmessy Example_1.gmessi
Message:: newDirectory created as ./Example_1_ESSI_Simulation


Add_All_Node{ unit:= m, nof_dofs:= 3}
   Found!!
   Successfully Converted

Vary_Linear_Elastic_Isotropic_3D{Physical_Group#Soil, ElementCommand:= ↩
    [Add_8NodeBrick{}], Density:= 1600+10*(10-z)\ 0 \kg/m^3, ElasticModulus:= ↩
    20e9+10e8*(10-z)\-8\Pa, PoissonRatio:= 0.3}
   Found!!
   Successfully Converted

Fix_Dofs{Physical_Group#Base_Surface, all}
   Found!!
   Successfully Converted

Add_Node_Load_Linear{Physical_Group#Top_Surface, ForceType:= Fx, Mag:= 10*kN}
   Found!!
   Successfully Converted

*********************** Updated New Tag Numbering *********************
damping = 1
displacement = 1
element = 28
field = 1
load = 17
material = 4
motion = 19
node = 65
nodes = 65
Gmsh_Elements = 46
Gmsh_Nodes = 65
```

It must be noted that the terminal only displays information about gmESSI commands. The singular commands are directly copied to the main file. The translator creates a user defined directory *Example_1_ESSI_Simulation* and places

1. node.fei
2. element.fei
3. load.fei
4. main.fei
5. Example_1.msh

The terminal displays the WARNING, ERROR messages and log of command conversions as shown above. At the end, it displays the *Available ESSITag's* numbering, which can be refereed and used for further conversion.

ESSITags are explained later in this manual in Section 207.2.4.8.

The Real-ESSI input files produced can be tweaked a little if required. Once all is set, the model can be run through Real-ESSI Simulator

```
cd Example_1_ESSI_Simulation

### To run ESSI in sequential
# -- assuming sequential executable name is 'essi'
essi -f main.fei

### To run ESSI in parallel
# -- assuming parallel executable name is 'pessi'
mpirun -np 4 pessi -f main.fei
```

207.2.2 Gmsh Physical Groups and Geometrical Entities

207.2.2.1 Geometrical Entities

Geometrical entities are the most elementary group in Gmsh. Each point, line, surface and volume is a geometrical entity and possess a unique identification number. Elementary geometrical entities can then be manipulated in various ways, for example using the *Translate, Rotate, Scale or Symmetry* commands. They can be deleted with the *Delete* command, provided that no higher-dimension entity references them. Example_2.geo shows description of a geometry (.geo) file in gmsh for creating a *cantilever beam*. The files can be downloaded here. Alternatively, it can be located in the gmESSI directory by navigating to the Examples/Example_2 directory.

```
$ cat Example_2.geo
// Creating a point
Point(1) = {0,0,0};

// Dividing the beam length in 5 parts
Extrude (4,0,0) {Point{1}; Layers{5};}

// Dividing the beam width in 2 parts
Extrude (0,1,0) {Line{1}; Layers{2};Recombine;}

// Dividing the beam depth in 2 parts
Extrude (0,0,1) {Surface{5}; Layers{2};Recombine;}
```

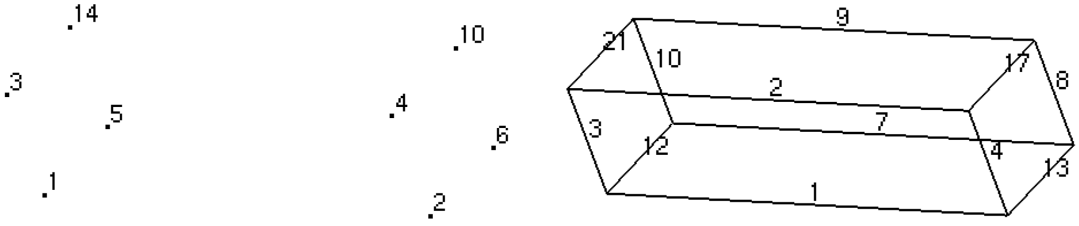
Figure 207.1 shows, the different unique identification number attached to each of the nodes, lines, surface and volume of the geometry of *cantilever beam*. Physical groups can now be created of type {nodes, lines, surface or volume} containing one or more geometrical entities of their respective type.

207.2.2.2 Physical Groups

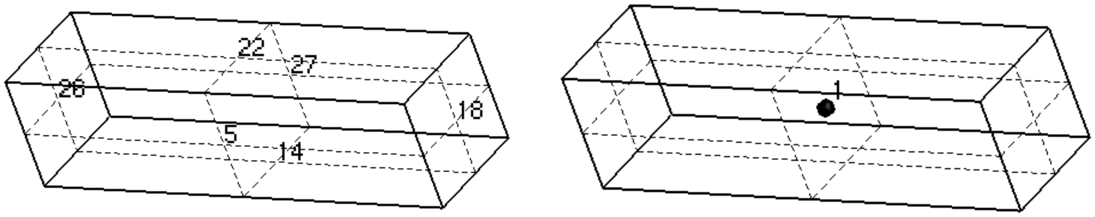
Physical groups are groups of same type {nodes, lines, surface, volume} of elementary geometrical entities. These Physical Groups cannot be modified by geometry commands. Their only purpose is to assemble elementary entities into larger groups, possibly modifying their orientation, so that they can be referred to by the mesh module as single entities. As is the case with elementary entities, each physical point, physical line, physical surface or physical volume are also assigned a unique identification number.

NOTE:- A geometrical entity has only one elementary entity number but can be a part of many physical groups by sharing their unique identification number.

Below is the continuation of Example_2.geo in Gmsh for creating physical Groups of *cantilever beam*.



(a) All Points (b) All Lines



(c) All Surface (d) All Volume

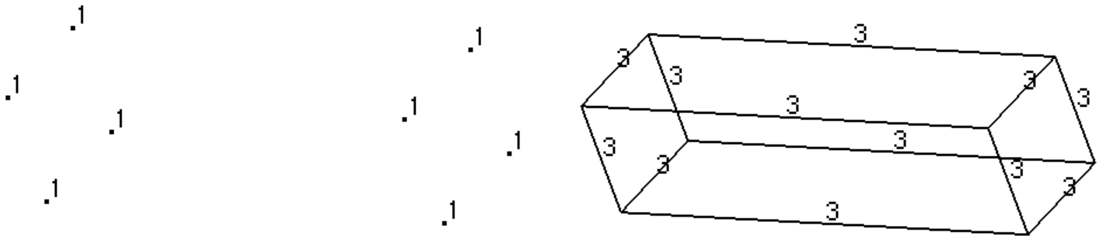
Figure 207.1: Showing Geometrical Entities. Every point, line, surface and volume has an unique identification number assigned to it.

Just for the sake of example, 4 physical groups are created which consist of all points, lines, surface and volume respectively of the *cantilever beam model*. Also physical groups of the surface where fixities and load is applied is created.

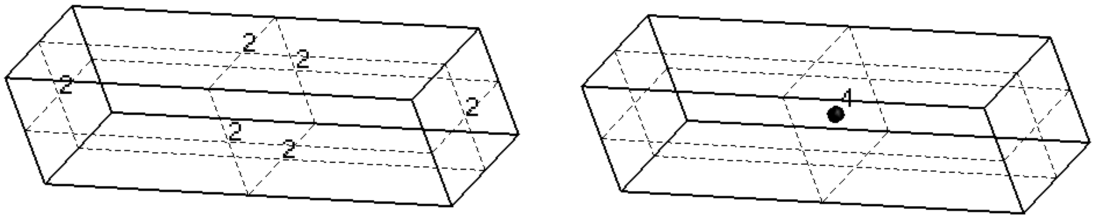
```
$ cat Example_2.geo
.....
Physical Point ("All_Points") ={1,2,3,4,5,6,10,14};
Physical Surface("All_Surfaces") = {5,14,22,27,18,26};
Physical Line("All_Lines") ={1,2,3,4,12,13,21,17,7,8,9,10};
Physical Volume("All_Volumes") ={1};
Physical Surface("ApplySurfaceLoad") ={27};
Physical Surface("SurfaceToBeFixed") ={26};
```

In generated mesh (.msh) file, all the geometrical entities have a tag list which contains the ids of the physical groups to which it belongs or is associated. In the above example shown in Figure 207.2, every point, line, surface, volume belongs to only one physical group and thus are showing only one associative number against themselves. Figure 207.3 shows geometrical entities which are part of many physical groups. For example:- the volume shown in Figure 207.3 shows physical group of volumes having id 1 and 7.

The whole idea of creating a Physical Group of points, lines, surfaces and volumes and giving it a unique string name is to allow quick identification and manipulation during gmESSI commands. In Gmsh the



(a) Physical group of Points (b) Physical group of Lines



(c) Physical group of Surface (d) Physical group of Volume

Figure 207.2: Showing all 4 Physical Groups with entities numbered by their physical group id's.

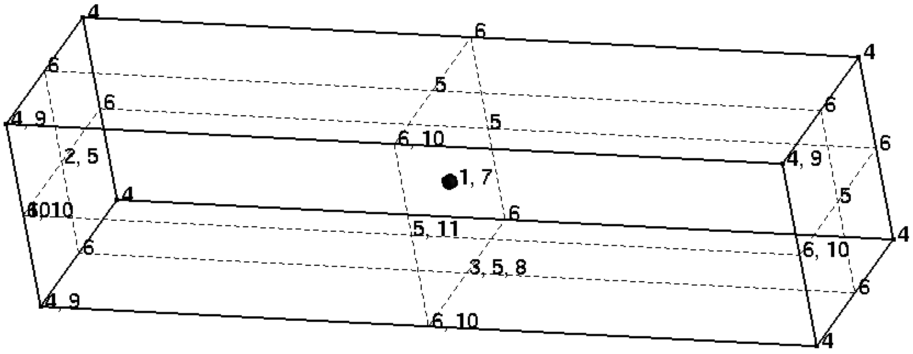


Figure 207.3: Showing geometrical entities associated with more than one physical group.

name of these Physical Group along with their corresponding elements and nodes gets transferred to the mesh .msh file as shown below. Figure 207.4 shows how Gmsh interprets these Physical groups in .msh file.

```
$cat Example_1.msh
.....
$PhysicalNames
6
0 1 "All_Points"
1 3 "All_Lines"
2 2 "All_Surfaces"
```

```
2 5 "ApplySurfaceLoad"
2 6 "SurfaceToBeFixed"
3 4 "All_Volumes"
$EndPhysicalNames
......
```

NOTE:- While creating a physical group in Gmsh, only the information (nodes and elements) of that physical group gets written in the .msh file and rest are not written. So one must be careful to create physical groups of all entities which is needed during post-processing or conversion. More information about Gmsh syntax, physical groups, commands, .msh file, save options, is available at the main online documentation web site: http://geuz.org/gmsh/doc/texinfo/gmsh.html

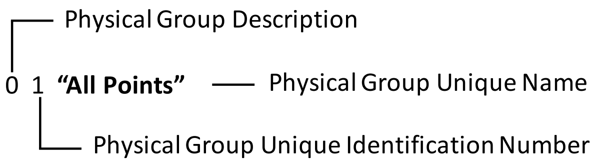


Figure 207.4: Description showing how gmsh interprets the Physical Groups.

- Physical Group Description :: Gmsh uses it to identify the type of physical group. $0, 1, 2$ *and* 3 represents the physical group of geometric points, lines, surface and volume respectively.
- Physical Group Unique Identification Number :: It is an unique identification number automatically assigned to each physical group by gmsh.
- Physical Group Unique Name :: It is also the same as *Physical Group Unique Identification Number* but the difference is that it is not automatic but defined by the user and that too in the form of string.

The gmESSI Translator utilizes the property of naming the physical group as "string" to get gmESSI commands from the user along with specific physical group on which it is operated. Below in shown [Example_2.gmessi] input file for a *Cantilever analysis*. It shows how to write gmESSI commands with physical group information on which it is operated. gmESSI utilizes the mesh (.msh) file to get the respective physical group and translated it to ESSI input (.fei) files.

```
$ cat Example_2.gmessi

gmESSI.loadGmshFile("Example_2.msh")
```

```

####### Physical Groups Available in Example_2.msh file
#0 1 "All_Points"
#1 3 "All_Lines"
#2 2 "All_Surfaces"
#2 5 "ApplySurfaceLoad"
#2 6 "SurfaceToBeFixed"
#3 4 "All_Volumes"

############ Important!! to set the file names ###################
gmESSI.setSimulationDir("./Example_2_ESSI_Simulation")
gmESSI.setMainFile(gmESSI.SimulationDir+ "main.fei")
gmESSI.setNodeFile(gmESSI.SimulationDir+ "node.fei")
gmESSI.setElementFile(gmESSI.SimulationDir+ "element.fei")
gmESSI.setLoadFile(gmESSI.SimulationDir+ "load.fei")

// My new model
! model name "Cantilever_Analysis";

[Add_All_Node{Unit:= m, NumDofs:= 3}]

// Adding Material
! add material 1 type linear_elastic_isotropic_3d mass_density = 2000*kg/m^3 ↩
    elastic_modulus = 200*MPa poisson_ratio = 0.2;

[Add_8NodeBrick{Physical_Group#All_Volumes, material_no:= 1}]
[Fix_Dofs{Physical_Group#SurfaceToBeFixed, all}]

! include "node.fei";
! include "element.fei";

! new loading stage "Stage1_Uniform_Surface_Load";

# Adding Surface Load
#[Add_8NodeBrick_SurfaceLoad{Physical_Group#All_Volumes, ↩
    Physical_Group#ApplySurfaceLoad, -10*Pa}]
[Add_Node_Load_Linear{Physical_Group#ApplySurfaceLoad, ForceType:= Fz, Mag:= ↩
    -10*kN}]

! include "load.fei";
! define algorithm With_no_convergence_check;
! define solver UMFPack;
! define load factor increment 1;
! simulate 10 steps using static algorithm;
! bye;
```

NOTE:- The first command in [.gmessi] file should be to load the mesh (.msh) file. The syntax to load the gmsh generated mesh file is

```
gmESSI.LoadGmshFile("meshfile.msh")
```

The gmESSI translator reads the command *[Add_All_Node{ Unit:= m, NumDofs:= 3}]* and adds all the nodes from mesh file to ESSI input files. Similarly it translates all the other commands as well.

207.2.3 gmESSI Command Description

gmESSI Translator as said above utilizes the naming of the physical groups to get commands from the user and then carry out the conversion by acting on the defined physical group.

207.2.3.1 gmESSI Syntax

gmESSI follows strict syntax. gmESSI parses the physical group name string in mesh (.msh) file. Let us have a quick look at the syntax of physical group name.

Physical Group Names : Physical group names are created inside gmsh geometry file. gmESSI follows special syntax as described below.

1. Physical group names used in gmsh should be unique for gmESSI to identify them during post processing.
2. Physical group names should not contain any space
3. Physical group tags can be any alphanumeric sequence but should not contain any of these [ ]$ literals in their names. Example "Physical_Group_1"

gmESSI Command Syntax : gmESSI translator commands are always enclosed between opening/closing square brackets [and] respectively. A typical gmESSI command syntax is shown in Figure 207.5

Figure 207.5: gmESSI command description.

- Command Name : Just as regular function gmESSI Commands have a name and take arguments. The names are usually self explanatory of its function like *Add_8NodeBrick*{...}, *Free_Dofs*{...} .. *etc*
- Physical Group Argument : Usually the gmESSI commands have first argument as physical group . For Example:- *Add_8NodeBrick*{*PhysicalGroup#5,...*}, *Add_8NodeBrick*{*PhysicalGroup#All_Volumes,...*}, *Free_Dofs*{*PhysicalGroup#4,...*},.. *etc*.

  Physical Group Id can be the gmsh unique string or number representing that physical group (as shown in .msh file).

- Arguments : Arguments as always are separated by comma ','.
  - Argument Tag The arguments of gmESSI commands can also have tags associated with them so that it becomes easy for the user to interpret the argument and make changes in future. The tag and the argument is separated by :=. Tag itself has no meaning but it serves as an important information center for user. An example is shown below to show how tags are applied.

    – gmESSI command having arguments without tags

    1. [Add_Node_Load_Linear{Physical_Group#ApplySurfaceLoad, Fz, -10*kN}]

    – gmESSI command having arguments with tags

    1. [Add_Node_Load_Linear{Physical_Group#ApplySurfaceLoad, ForceType:= Fz, Mag:= -10*kN}]
    2. [Add_Node_Load_Linear{Physical_Group#ApplySurfaceLoad, ForceType:= Fz, -10*kN}]
    3. [Add_Node_Load_Linear{Physical_Group#ApplySurfaceLoad, Force_Direction:= Fz, Strength:= -10*kN}]

It can be seen from above examples that the tags are optional and also the user can put their own tag names. The sublime plugin [*gmESSI-Tools*] comes with elaborative tags for the parameters and a lot more with syntax coloring and text-completion for gmESSI commands. It is encouraged to use the plugin and take its advantage.

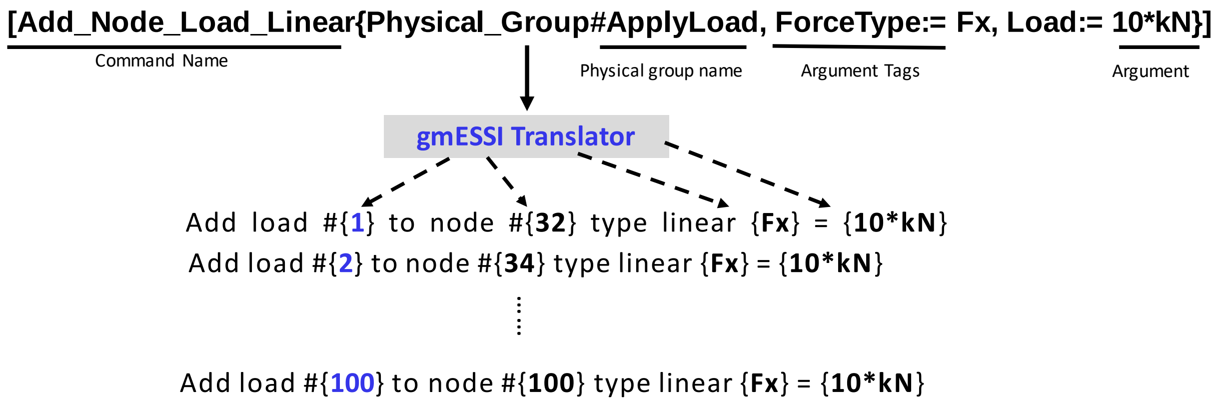


Figure 207.6: gmESSI conversion description.

Figure 207.6 shows the illustration how gmESSI works. Load gets added to all the nodes of the physical group 'ApplyLoad'. gmESSI translator automatically assigns the unique load tag sequentially. It retrieves the node tag from the physical group. Rest of the information like 'ForceType' and 'Magnitude' is obtained from the arguments.

Most of the time these arguments are dummy which means that they just get copied to their equivalent ESSI command at their respective places. These arguments thus have a "string" data-type. For example: the command Add_Node_Load_Linear{Physical_Group#ApplySurfaceLoad, Fz, -10*kN} is equivalent to the Real-ESSI command add load #{} to node #{} type linear {} = {}. Fz and -10*kN goes to their respective position directly through the translator as shown in the Figure 207.6 load number 1 and node number 32 are computed by the translator and then inserted in the ESSI command.

NOTE:- The gmESSI Translator does not provide syntax checking for those dummy arguments. It means that, whatever is written gets copied at the respective position in the equivalent ESSI command, so the one must be careful with what they are writing in these arguments. For Example the command Add_Node_Load_Linear{Physical_Group#Id, ForceDirection, Magnitude} based on the arguments can get converted as

1. [Add_Node_Load_Linear{Physical_Group#ApplySurfaceLoad, ForceType:= Fz, Mag:= -10*kN}]
   —> add load #1 to node #32 type linear Fz = -10*kN
   —> add load #2 to node #33 type linear Fz = -10*kN
   ....
   —> add load #100 to node #100 type linear Fz = -10*kN
2. [Add_Node_Load_Linear{Physical_Group#ApplySurfaceLoad, ForceType:= Fz, Mag:= -10}]
   —> add load #1 to node #32 type linear Fz = -10
   —> add load #2 to node #33 type linear Fz = -10
   ....
   —> add load #100 to node #100 type linear Fz = -10
3. [Add_Node_Load_Linear{Physical_Group#ApplySurfaceLoad, ForceType:= Ft, Mag:= -10*kN}]
   —> add load #1 to node #32 type linear Ft = -10*kN
   —> add load #2 to node #33 type linear Ft = -10*kN
   ....
   —> add load #100 to node #100 type linear Ft = -10*kN

All the above conversions are correct. But only conversion (1.) is correct as an input for Real-ESSI Simulator because force direction is one of *Fx,Fy,Fz* and magnitude *10*kN* has *proper units*. So one must be very careful while writing the arguments.

Note: Some of the arguments are not string but represents numerical quantities, which are manipulated by the translator during conversion. Thus, the one must supply only numbers without any alphabets else it would lead an unexpected termination of program. These arguments corresponds to *Special Commands* such as *Connect Command* and *Variational Commands*. The manual talks about them later in Section 207.2.5.7.

207.2.3.2 gmESSI Command's Physical Group

As iterated earlier, gmESSI commands operates on physical groups. The gmESSI command usually have their first argument as physical on which it operates. The gmESSI syntax allows the users to operates it's command on specific physical groups. The user specifies the group by including an argument Physical_Group#Tag in front of the gmESSI commands describing the command. The *tag* can be either *Physical_Group_Id, Physical_Group_Name*. Let's look at some of them

- [Add_Node_Load_Linear{Physical_Group#5,Fz,-10*kN}] operates on physical group 5
- [Add_8NodeBrick{Physical_Group#All_Volumes, 1}] operates on physical group which has string_tag as All_Volumes

For example in reference to *[Example_2.gmessi]* Physical_Group#All_Volumes or Physical_Group#4 refers the same physical group.

A physical group is a group of point, line, surface or volume defined by the user which contains all the geometrical entities that falls under that domain/group. Figure 207.2 shows physical groups.

207.2.4 gmESSI Output

gmESSI Translator translates the gmESSI commands operated on mesh (.msh) file to different ESSI input (node, element, load and main) (.fei) files and put them in user-defined directory. It also updates the mesh (.msh) file and puts it in the same directory. The *log of translation, errors and warnings* are displayed on the terminal. Below is the demonstration of log messages one by one using *[Example_2.gmessi]* with mesh-file name *Example_2.msh*. The folders and Reall ESSI input (.fei) files that are created by the translator for Example_2.gmessi input file are.

207.2.4.1 Directory Example_2_ESSI_Simulation

gmESSI Translator creates simulation directory as specified by the user. The user is expected to create the necessary node (Section 207.2.4.4), element (Section 207.2.4.3) , load (Section 207.2.4.5) and main (Section 207.2.4.6) file to that directory. The user is expected to provide the directory and filenames before executing any gmESSI command. In case the directory already exists a warning messages is shown on the terminal and a new directory following the original name with '_n' (n is number) is created. A new Real-ESSI simulation directory is assigned by the following command

```
gmESSI.setSimulationDir("./Example_2_ESSI_Simulation", overwrite\_mode)
```

where, 'overwrite_mode=0' means that in case of already existing folder, a new directory following the original name with '_n' (n is number) is created. 'overwrite_mode=1' would not check for any conflicts and use the same directory as specified by user. For example:- running [Example_.gmessi] file would produce the following message.

```
$ gmessy Example_2.gmessi
Files converted to Examples/Example_2_ESSI_Simulation
```

Again, running the same example would produce the following message as shown below. In [Example_2.gmessi] overwrite is turned off and thats why it creates new-non conflicting directory by appending $_1$ to end.

```
$ gmessy Example_2.gmessi
Message:: newDirectory created as ./Example_2_ESSI_Simulation_1
```

The execution of *gmessy XYZ.gmessi* produces warnings/errors in the following situations.

- ERROR:: Please Enter the gmessi File :: It occurs if the user does not give a filename. The possible situation for getting this error is

```
$ gmessy
```

- ERROR:: The program failed to open the file XYZ.msh It occurs if the given file or one of the files in the argument does not exist or fails to open because of some reason.
- WARNING::Directory Already Present.The contents of the Folder may get changed :: It occurs when users translates the mesh file file XYZ.msh in overwrite mode and the corresponding folder XYZ_ESSI_Simulation already exists at the execution location.
- Files converted to Examples/Example_2_ESSI_Simulation :: The message refers to the location of the folder where the translations have been saved.

207.2.4.2 Translation Log Terminal

gmESSI Translator displays the log of translation of gmESSI commands to corresponding *Real-ESSI commands* on the terminal. Proper *Errors Messages* and *Warnings* are echoed to the user. The execution of the commands are sequential which means the commands written first are executed first and similarly their success and failure is also echoed first. Let us look at this aspect with Example_2.gmessi.

```
$cat Example_2.gmessi
......
! add material 1 type linear_elastic_isotropic_3d mass_density = 2000*kg/m^3 ↩
    elastic_modulus = 200*MPa poisson_ratio = 0.2;

[Add_8NodeBrick{Physical_Group#All_Volumes, material_no:= 1}]

[Fix_Dofs{Physical_Group#SurfaceToBeFixed, all}]

! include "node.fei";
! include "element.fei";
......
```

Here, the sequence of execution of commands is '! add material # 1 type linear_elastic_isotropic_3d mass_density $= 2000 * kg/m^3$ elastic_modulus $= 200 * MPa$ poisson_ratio $= 0.2$; ', [Add_8NodeBrick{ Physical_Group#All_Volumes, material_no:= 1}], [Fix_Dofs{ Physical_Group#SurfaceToBeFixed, all}] and '! include "node.fei";'. Notice that the same order gets reflected in the translation log on the terminal as shown below. Also, it must be noted that the commands followed by '!' or '//' or '#' or python commands do not have any log messages corresponding to them.

It must be noted that the lines following '!' are directly copied to the main (Section 207.2.4.6). Usually Real-ESSI domain specific language that does not operate/require any physical group should be written following exclamation '!' sign.

```
$ gmessy ./Example_2.gmessi
......
```

```

Add_8NodeBrick{Physical_Group#All_Volumes, material_no:= 1}
Found!!
Successfully Converted

Fix_Dofs{Physical_Group#SurfaceToBeFixed, all}
Found!!
Successfully Converted
```

Apart from displaying the log details on the terminal, similar log is added for each translation of gmESSI commands in their respective files in which they are translated. In these files, each successful translation is enclosed between corresponding *RespectiveGmESSICommand Begins* and *RespectiveGmESSICommand Ends*. The same is shown below through the contents of node.fei. Notice that all the translations are enclosed between Begins and Ends Tag.

```
$ cat Examples/Example_2_ESSI_Simulation/node.fei

//****************************************************************
// Add_All_Node{Unit:= m, NumDofs:= 3}Starts
//****************************************************************

add node # 1 at (0.000000*m,0.000000*m,0.000000*m) with 3 dofs;
add node # 2 at (4.000000*m,0.000000*m,0.000000*m) with 3 dofs;
add node # 3 at (0.000000*m,1.000000*m,0.000000*m) with 3 dofs;
...............................................................

//****************************************************************
// Add_All_Node{Unit:= m, NumDofs:= 3}Ends
//****************************************************************
```

```
$ cat Examples/Example_2_ESSI_Simulation/element.fei

//***********************************************************************
// Add_8NodeBrick{Physical_Group#All_Volumes, material_no:= 1}Starts
//***********************************************************************

add element #1 type 8NodeBrick with nodes (51,46,29,37,33,17,1,9) use material #1;
add element #2 type 8NodeBrick with nodes (47,28,5,19,51,46,29,37) use material ↩
   #1;
add element #3 type 8NodeBrick with nodes (42,32,46,51,13,3,17,33) use material ↩
   #1;
..............................................................

//*********************************************************************
// Add_8NodeBrick{Physical_Group#All_Volumes, material_no:= 1}Ends
//*********************************************************************
```

NOTE:- The ordering/sequence of commands in ESSI analysis file is important and so the user must make sure that the translations are made in the same order or if not the user should change it manually by (cut/copy/paste) in *node.fei, load.fei and main.fei* files before execution.

Having given a short description of the other translation log/error messages. Let us look more closely one by one and understand the messages,errors and warnings prompted on the terminal.

- Found!! : This message in front of the gmESSI command as shown above on translation log in the terminal means that, the corresponding command was found in the gmESSI Command Library.
- Successfully Converted : As the message itself describes, it occurs if the command has been successfully translated.
- Not Found!! : It occurs if the gmESSI Translator could not find the arbitrary command XYZ in the gmESSI Command library. Example:- *Loading{Fx,10*kN} NotFound!!*
- WARNING:: Execution of the command escaped. The Gmessi command XYZ could not be found : The gmESSI Translator does not terminate the translation if a command is not found, instead gives this warning message following the *Not Found!! Error*.
- Error:: The command XYZ has a syntax error in Physical_Group# tag : It occurs if there is a syntax error in Physical_Group# argument. The correct represent ion for Physical group Tags is Physical_Group#n, where n is the group id as 1,2,3.. etc. Examples of improper representation are *Phy#2, Physical#Node, ..*
- Warning:: The command XYZ failed to convert as there is no such Physical Group :: It occurs if one of the arguments in the command is Physical_Group# and the specified physical group by the user does not exists in the .msh file.
- Warning:: The command XYZ could not find any nodes/elements on which it operates : It occurs if for a specified command, the required element types for translation could not be found in the specified Physical group. For Examples:- [Add_8NodeBrick{Physical_Group#1,1}] would give this warning as the Physical_Group#1 being a Physical line group does not contain any 8-Noded Brick elements on which this command operates.
- ERROR:: Gmsh File has invalid symbols in Node Section. Unable to convert string to integer in Gmsh File : It occurs if there is perhaps a string inside the Nodes section of .msh file.
- ERROR:: The command XYZ has a syntax errors :: It occurs if the specified command by the user contain any syntax errors caught while parsing the command.

- ERROR:: Gmsh File has invalid symbols in Element Section. Unable to convert string to integer in Gmsh File : It occurs if there is perhaps a string inside the Element section of .msh file.

207.2.4.3 Element File (element.fei)

Element file *element.fei* is one of four parts of Real-ESSI input file that contains the translation of commands related to only initialization of elements of the FEM mesh. Generally, all the conversions from Elemental Command (Section 207.2.5.5) are written to element file.

A new analysis element file is assigned by the following python command

```
gmESSI.setElementFile(gmESSI.SimulationDir+ "element.fei")
```

where, 'gmESSI.SimulationDir' returns the Real-ESSI Simulation directory specified by the user (see section 207.2.4.1).

207.2.4.4 Node File (node.fei)

Node file *node.fei* is one of four parts of Real-ESSI input file that contains the translation of commands related to only initialization of nodes of the FEM mesh. All the conversions from Add Node Command (Section 207.2.5.2) are written to node file.

A new analysis node file is assigned by the following python command

```
gmESSI.setNodeFile(gmESSI.SimulationDir+ "node.fei")
```

where, 'gmESSI.SimulationDir' returns the Real-ESSI Simulation directory specified by the user (see section 207.2.4.1).

207.2.4.5 Load File (load.fei)

Load file *load.fei* contains the translation of commands related to the load and boundary conditions on the structure, for example declaration of fixities, boundary conditions, tied/connected nodes, nodal loads, surface loads etc....

A new load file is assigned by the following python command

```
gmESSI.setLoadFile(gmESSI.SimulationDir+ "load.fei")
```

where, 'gmESSI.SimulationDir' returns the Real-ESSI Simulation directory specified by the user (see section 207.2.4.1).

207.2.4.6 Analysis File (main.fei)

Analysis file *main.fei* is the main file which is run on Real-ESSI Simulator. The main file must include load, node and element file through *include 'filename.fei'* command.

A new analysis main file is assigned by the following python command

```
gmESSI.setMainFile(gmESSI.SimulationDir+ "main.fei")
```

where, 'gmESSI.SimulationDir' returns the Real-ESSI Simulation directory specified by the user (see section 207.2.4.1). A typical analysis file after conversion looks like the following.

```
$ cat Examples/Example_2_ESSI_Simulation/Example_2_analysis.fei

// My new model
 model name "Cantilever_Analysis";

// Adding Material
add material 1 type linear_elastic_isotropic_3d mass_density = 2000*kg/m^3 ↩
    elastic_modulus = 200*MPa poisson_ratio = 0.2;

include "node.fei";
include "element.fei";

new loading stage "Stage1_Uniform_Surface_Load";

include "load.fei";
 define algorithm With_no_convergence_check;
 define solver UMFPack;
 define load factor increment 1;
 simulate 10 steps using static algorithm;
bye;
```

The user can now add solver, time steps and even rearrange the file structure accordingly to Real-ESSI syntax.

NOTE: Real-ESSI Interpreter is sequential and follows certain ordering in commands like materials should be declared before assigning to elements, main-follower nodes can be assigned only when both nodes are declared .. etc.. One should be careful with the order in which conversions are made and if necessary should change it manually by (cut/copy/paste) later in the files geometry.fei, load.fei and analysis.fei or use the python module discussed later before running in ESSI.
Please refer to the Real-ESSI manual for more details on the ordering of the commands.

207.2.4.7 Mesh File (XYZ.msh)

Mesh file *XYZ.msh* is the input required by the translator. The translator updates the mesh file with users addition. For example:- if Connect-Command (Section 207.2.5.8) is used, the file contains additional physical group, nodes and 2-noded elements. The Connect Command is discussed in the more detail later in Section 207.2.5.8.

207.2.4.8 Updated ESSI Tags Terminal

Updated ESSI Tags refers to the new tag numbering reference associated with ESSI Tags. ESSI has tag numberings associated for damping, displacement, element, field, load, material, and node/nodes. For example in Real-ESSI Command *add node # 1 at (x,y,z) with 3 dofs*, node is a tag and requires a new number like 1 to be associated with that node. The translator displays the new numberings available for each ESSI Tag so that the user is made aware of new numberings for manually specifying an ESSI command after the translation.

gmESSI also provides a python command to set the ESSI Tag. The command is

```
gmESSI.setESSITag(ESSI_Tag_Name,Tag)
```

where,

- ESSI_Tag_Name : It refers to a string representing to the Real-ESSI tag such as 'node', 'element', 'field'...etc
- Tag : It refers to an integer representing the next available tag.

NOTE : If user is writing its own Real-ESSI domain specific language (DSL), it is expected that the user will update the corresponding Real-ESSI tag used in that DSL. Otherwise, gmESSI would not be able to know the updated available tags. See Example_1.gmessi for its usage.

```
$ gmessy Example_2.gmessi
.....................
******* Updated New Tag Numbering ********
Damping = 1
displacement = 1
element = 21
field = 1
load = 19
material = 2
motion = 1
node = 55
nodes = 55
Gmsh_Elements = 127
Gmsh_Nodes = 55
```

207.2.5 gmESSI Commands

Having the knowledge about the syntax, output files, errors and warnings, its time to move on to different types of commands that gmESSI offers. it provide commands operated on physical group to allow conversion for to equivalent Real-ESSI commands. There are also some special command that gmESSI supports. For simplicity, the commands are categorized on the basis of their operation on nodes/elements. As stated earlier, the commands are translated to one of the four files *node.fei, element.fei, load.fei and main.fei*. Let us look at them closely one by one along with all its supported commands.

207.2.5.1 Singular Commands

Singular Commands does not require any physical group to operate. All the text following exclamation mark '!' are copied directly to the *main.fei* (Section 207.2.4.6). For Example:- ' ! include 'load.fei'; ' is translated as 'include "load.fei" ' in main.fei analysis file. See [Example_1.gmessi] for its usage.

Note:- Real-ESSI DSL/commands must be followed by the exclamation mark '!'.

207.2.5.2 Add Node Commands

Add Node Commands have only two commands. *[Add_All_Node{unit,nof_dofs}]* adds all the nodes generated in mesh (.msh) file to 'node.fei' file. Whereas, *[Add_Node{Unit,NumDofs}]* add all the nodes of only specified physical group by the user. These commands operates on all the nodes of the physical group and generate an equivalent Real-ESSI DSL for each of them.

NOTE:- Every Add Node commands get translated into the *node.fei* (Section 207.2.4.4).

- gmESSI : [Add_Node{PhysicalGroup , Unit , NumDofs}]
  translates to series of
  Real-ESSI DSL : add node $\# < . >$ at ($< L >$,$< L >$,$< L >$) with $< . >$ dofs;
  operated over all the nodes defined in the gmsh '.msh' file.

- gmESSI : [Add_All_Node{Unit , NumDofs}]
  translates to series of
  Real-ESSI DSL : add node $\# < . >$ at ($< L >$,$< L >$,$< L >$) with $< . >$ dofs;
  operated over all the nodes of the defined physical group

207.2.5.3 Nodal Commands : Operates On All Nodes of the defined Physical Group

Nodal commands operates on all the nodes of the physical group defined by the user. For example:- [Fix_Dofs{Physical_Group#Lateral_Surface,ux}] would fix ux degree of freedom of all the nodes of physical group 'Lateral_Surface'. It will generate equivalent Real-ESSI DSL 'fix node # < . > dof < . >' and apply to all the nodes of that physical group. Figure 207.6 shows how gmESSI operated on physical groups.

As earlier stated, that the arguments of gmESSI commands are dummy and gets copied directly to the ESSI equivalent command, so one must be very much aware while writing the arguments to the commands. The arguments should be filled with values of the corresponding ESSI command along with required units if any. For more details about the values to the arguments, please refer to ESSI Manual.

NOTE:- Every Nodal command gets translated to the *load.fei* file (Section 207.2.4.5).

The different commands under this category and their corresponding Real-ESSI commands are listed below

1. gmESSI : [Add_Nodes_To_Physical_Group{PhysicalGroup , Physical_Node_Group_String}]
   translates to series of
   Real-ESSI DSL : add nodes (< . >) to [physical_node_group] "*string*";
   operated over all the nodes of the defined physical group

2. gmESSI : [Add_Self_Weight_To_Node{PhysicalGroup , field#1}]
   translates to series of
   Real-ESSI DSL : add load # < . > to node # < . > type [self_weight] use acceleration field # < . >;
   operated over all the nodes of the defined physical group

3. gmESSI : [Add_Node_Load_Linear{PhysicalGroup , Force_Type , Magnitude}]
   translates to series of
   Real-ESSI DSL : add load # < . > to node # < . > type [linear] [FORCETYPE] = < *forceormoment* >; //[FORCETYPE] = [Fx] [Fy] [Fz] [Mx] [My] [Mz] [F_fluid_x] [F_fluid_y] [F_fluid_z]
   operated over all the nodes of the defined physical group

4. gmESSI : [Add_Node_Load_Path_Time_Series{PhysicalGroup , Force_Type , Magnitude , Series_File}]
   translates to series of
   Real-ESSI DSL : add load # < . > to node # < . > type [path_time_series] [FORCETYPE] =

$< forceormoment >$ series_file = "$string$";
operated over all the nodes of the defined physical group

5. gmESSI : [Add_Node_Load_Path_Series{PhysicalGroup , Force_Type , Magnitude , Time_Step , Series_File}]
   translates to series of
   Real-ESSI DSL : add load # < . > to node # < . > type [path_series] [FORCETYPE] = $< forceormoment >$ time_step = $< T >$ series_file = "$string$";
   operated over all the nodes of the defined physical group

6. gmESSI : [Add_Node_Load_From_Reaction{PhysicalGroup}]
   translates to series of
   Real-ESSI DSL : add load # < . > to node # < . > type [from_reactions];
   operated over all the nodes of the defined physical group

7. gmESSI : [Add_Node_Load_Imposed_Motion_Time_Series{PhysicalGroup , Dof_Type , Time_Step , Disp_Scale , Disp_File , Vel_Scale , Vel_File , Acc_Scale , Acc_File}]
   translates to series of
   Real-ESSI DSL : add imposed motion # < . > to node # < . > dof $< DOFTYPE >$ time_step = $< T >$ displacement_scale_unit = $< L >$ displacement_file = "$string$" velocity_scale_unit = $< L/T >$ velocity_file = "$string$" acceleration_scale_unit = $< L/T^2 >$ acceleration_file = "$string$";
   operated over all the nodes of the defined physical group

8. gmESSI : [Add_Node_Load_Imposed_Motion_Time_Series{PhysicalGroup , Dof_Type , Time_Step , Disp_Scale , Disp_File , Vel_Scale , Vel_File , Acc_Scale , Acc_File}]
   translates to series of
   Real-ESSI DSL : add load # < . > type imposed motion to node # < . > dof $< DOFTYPE >$ time_step = $< T >$ displacement_scale_unit = $< L >$ displacement_file = "$string$" velocity_scale_unit = $< L/T >$ velocity_file = "$string$" acceleration_scale_unit = $< L/T^2 >$ acceleration_file = "$string$";
   operated over all the nodes of the defined physical group

9. gmESSI : [Add_Node_Load_Imposed_Motion_Series{PhysicalGroup , Dof_Type , Disp_Scale , Disp_File , Vel_Scale , Vel_File , Acc_Scale , Acc_File}]
   translates to series of
   Real-ESSI DSL : add imposed motion # < . > to node # < . > dof $< DOFTYPE >$ displacement_scale_unit = $< L >$ displacement_file = "$string$" velocity_scale_unit = $< L/T >$ velocity_file

= "*string*" acceleration_scale_unit = $< L/T^2 >$ acceleration_file = "*string*";
operated over all the nodes of the defined physical group

10. gmESSI : [Add_Node_Load_Imposed_Motion_Time_Series{PhysicalGroup , Dof_Type , Time_Step , Disp_Scale , Disp_File , Vel_Scale , Vel_File , Acc_Scl , Acc_File}]
translates to series of
Real-ESSI DSL : add load # < . > type imposed motion to node # < . > dof $< DOFTYPE >$ displacement_scale_unit = $< L >$ displacement_file = "*string*" velocity_scale_unit = $< L/T >$ velocity_file = "*string*" acceleration_scale_unit = $< L/T^2 >$ acceleration_file = "*string*";
operated over all the nodes of the defined physical group

11. gmESSI : [Add_Damping_To_Node{PhysicalGroup , damping#1}]
translates to series of
Real-ESSI DSL : add damping # < . > to node # < . >;
operated over all the nodes of the defined physical group

12. gmESSI : [Add_Mass_To_Node{PhysicalGroup , MassX , MassY , MassZ}]
translates to series of
Real-ESSI DSL : add mass to node # < . > mx = $< M >$ my = $< M >$ mz = $< M >$;
operated over all the nodes of the defined physical group

13. gmESSI : [Add_Beam_Mass_To_Node{PhysicalGroup , MassX , MassY , MassZ , ImassX , ImassY , ImassZ}]
translates to series of
Real-ESSI DSL : add mass to node # < . > mx = $< M >$ my = $< M >$ mz = $< M >$ Imx = $< ML^2 >$ Imy = $< ML^2 >$ Imz = $< ML^2 >$;
operated over all the nodes of the defined physical group

14. gmESSI : [Fix_Dofs{PhysicalGroup , Dof_Types}]
translates to series of
Real-ESSI DSL : fix node # < . > dofs $< DofTypes >$;
operated over all the nodes of the defined physical group

15. gmESSI : [Free_Dofs{PhysicalGroup , Dof_Types}]
translates to series of
Real-ESSI DSL : free node # < . > dofs < . >;
operated over all the nodes of the defined physical group

16. gmESSI : [Remove_Node{PhysicalGroup}]
    translates to series of
    Real-ESSI DSL : remove node # $<$. $>$;
    operated over all the nodes of the defined physical group

17. gmESSI : [Remove_Equal_Dof_Constrain{PhysicalGroup}]
    translates to series of
    Real-ESSI DSL : remove constraint [equal_dof] node # $<$. $>$;
    operated over all the nodes of the defined physical group

18. gmESSI : [Remove_Displacement_From_Node{PhysicalGroup}]
    translates to series of
    Real-ESSI DSL : remove displacement from node # $<$. $>$;
    operated over all the nodes of the defined physical group

207.2.5.4 General Elemental Commands : Operates On All Elements of the defined Physical Group

General Elemental Commands operates on all the elements of a physical group. The translations are written in *load.fei* file. For example:- [Add_SelfWeight_To_Element{Physical_Group#Soil,Field:= 1}] would add self-weight to all the elements of the physical group 'Soil' along the field#1 direction using series of equivalent Real-ESSI DSL 'add load # < . > to element # < . > type [self_weight] use acceleration field # < . >;'

The different commands under this category and their corresponding ESSI commands are listed below

1. gmESSI : [Add_Elements_To_Physical_Group{PhysicalGroup , Physical_Element_Group_String}]
   translates to series of
   Real-ESSI DSL : add elements (< . >) to [physical_element_group] "*string*";
   operated over all the nodes of the defined physical group
2. gmESSI : [Add_Self_Weight_To_Element{PhysicalGroup , field#1}]
   translates to series of
   Real-ESSI DSL : add load # < . > to element # < . > type [self_weight] use acceleration field # < . >;
   operated over all the nodes of the defined physical group
3. gmESSI : [Add_Damping_To_Element{PhysicalGroup , damping#1}]
   translates to series of
   Real-ESSI DSL : add damping # < . > to element # < . >;
   operated over all the nodes of the defined physical group
4. gmESSI : [Remove_Element{PhysicalGroup}]
   translates to series of
   Real-ESSI DSL : remove element # < . >;
   operated over all the nodes of the defined physical group
5. gmESSI : [Remove_Strain_From_Element{PhysicalGroup}]
   translates to series of
   Real-ESSI DSL : remove strain from element # < . >;
   operated over all the nodes of the defined physical group

207.2.5.5 Elemental Commands : Operates On All Elements of the defined Physical Group

Elemental Commands operates only to specific elements of a physical group. The translations are written in *element.fei* file. For example:- [Add_8NodeBrick{Physical_Group#Soil,1}] would initialize all the hexahedron elements of physical group 'Soil' to equivalent Real-ESSI commands for defining 8-noded bricks elements 'add element # $< . >$ type [8NodeBrick] with nodes ($< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$) use material # $< . >$;'. Figure 207.6 shows how gmESSI operated on physical groups. The different commands under this category and their corresponding ESSI commands are listed below

1. gmESSI : [Add_20NodeBrick{PhysicalGroup , Num_Gauss_Points , material#1}]
   translates to series of
   Real-ESSI DSL : add element # $< . >$ type [20NodeBrick] with nodes ($< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$) use material # $< . >$;
   operated over all the elements of the defined physical group

2. gmESSI : [Add_20NodeBrick_Variable_GaussPoints{PhysicalGroup , Num_Gauss_Points , material#1}]
   translates to series of
   Real-ESSI DSL : add element # $< . >$ type [20NodeBrick] using $< . >$ Gauss points each direction with nodes ($< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$) use material # $< . >$;
   operated over all the elements of the defined physical group

3. gmESSI : [Add_20NodeBrick_upU{PhysicalGroup , material#1 , Porosity , Alpha , Solid_Density , Fluid_Density , Perm_X , Perm_Y , Perm_Z , Solid_Bulk_Modulus , Fluid_Bulk_Modulus}]
   translates to series of
   Real-ESSI DSL : add element # $< . >$ type [20NodeBrick_upU] with nodes ($< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$) use material # $< . >$ and porosity = $< . >$ alpha = $< . >$ rho_s = $< M/L^3 >$ rho_f = $< M/L^3 >$ k_x = $< L^3T/M >$ k_y = $< L^3T/M >$ k_z = $< L^3T/M >$ K_s = $< stress >$ K_f = $< stress >$;
   operated over all the elements of the defined physical group

4. gmESSI : [Add_20NodeBrick_upU_Variable_GaussPoints{PhysicalGroup , Num_Gauss_Points , material#1 , Porosity , Alpha , Solid_Density , Fluid_Density , Perm_X , Perm_Y , Perm_Z , Solid_Bulk_Modulus

, Fluid_Bulk_Modulus}]

translates to series of

Real-ESSI DSL : add element # $< . >$ type [20NodeBrick_upU] using $< . >$ Gauss points each direction with nodes ($< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$) use material # $< . >$ and porosity = $< . >$ alpha = $< . >$ rho_s = $< M/L^3 >$ rho_f = $< M/L^3 >$ k_x = $< L^3T/M >$ k_y = $< L^3T/M >$ k_z = $< L^3T/M >$ K_s = $< stress >$ K_f = $< stress >$;

operated over all the elements of the defined physical group

5. gmESSI : [Add_20NodeBrick_up{PhysicalGroup , material#1 , Porosity , Alpha , Solid_Density , Fluid_Density , Perm_X , Perm_Y , Perm_Z , Solid_Bulk_Modulus , Fluid_Bulk_Modulus}]

   translates to series of

   Real-ESSI DSL : add element # $< . >$ type [20NodeBrick_up] with nodes ($< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$) use material # $< . >$ and porosity = $< . >$ alpha = $< . >$ rho_s = $< M/L^3 >$ rho_f = $< M/L^3 >$ k_x = $< L^3T/M >$ k_y = $< L^3T/M >$ k_z = $< L^3T/M >$ K_s = $< stress >$ K_f = $< stress >$;

   operated over all the elements of the defined physical group

6. gmESSI : [Add_20NodeBrick_up_Variable_GaussPoints{PhysicalGroup , Num_Gauss_Points , material#1 , Porosity , Alpha , Solid_Density , Fluid_Density , Perm_X , Perm_Y , Perm_Z , Solid_Bulk_Modulus , Fluid_Bulk_Modulus}]

   translates to series of

   Real-ESSI DSL : add element # $< . >$ type [20NodeBrick_up] using $< . >$ Gauss points each direction with nodes ($< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$) use material # $< . >$ and porosity = $< . >$ alpha = $< . >$ rho_s = $< M/L^3 >$ rho_f = $< M/L^3 >$ k_x = $< L^3T/M >$ k_y = $< L^3T/M >$ k_z = $< L^3T/M >$ K_s = $< stress >$ K_f = $< stress >$;

   operated over all the elements of the defined physical group

7. gmESSI : [Add_27NodeBrick{PhysicalGroup , material#1}]

   translates to series of

   Real-ESSI DSL : add element # $< . >$ type [27NodeBrick] with nodes ($< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$) use material # $< . >$;

operated over all the elements of the defined physical group

8. gmESSI : [Add_27NodeBrick_upU{PhysicalGroup , material#1 , Porosity , Alpha , Solid_Density , Fluid_Density , Perm_X , Perm_Y , Perm_Z , Solid_Bulk_Modulus , Fluid_Bulk_Modulus}]
   translates to series of
   Real-ESSI DSL : add element $\# < . >$ type [27NodeBrick_upU] with nodes ($< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$) use material $\# < . >$ and porosity $= < . >$ alpha $= < . >$ rho_s $= < M/L^3 >$ rho_f $= < M/L^3 >$ k_x $= < L^3T/M >$ k_y $= < L^3T/M >$ k_z $= < L^3T/M >$ K_s $= < stress >$ K_f $= < stress >$;
   operated over all the elements of the defined physical group

9. gmESSI : [Add_27NodeBrick_upU_Variable_GaussPoints{PhysicalGroup , NumGaussPoints , material#1 , Porosity , Alpha , Solid_Density , Fluid_Density , Perm_X , Perm_Y , Perm_Z , Solid_Bulk_Modulus , Fluid_Bulk_Modulus}]
   translates to series of
   Real-ESSI DSL : add element $\# < . >$ type [27NodeBrick_upU] using $< . >$ Gauss points each direction with nodes ($< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$) use material $\# < . >$ and porosity $= < . >$ alpha $= < . >$ rho_s $= < M/L^3 >$ rho_f $= < M/L^3 >$ k_x $= < L^3T/M >$ k_y $= < L^3T/M >$ k_z $= < L^3T/M >$ K_s $= < stress >$ K_f $= < stress >$;
   operated over all the elements of the defined physical group

10. gmESSI : [Add_27NodeBrick_up{PhysicalGroup , material#1 , Porosity , Alpha , Solid_Density , Fluid_Density , Perm_X , Perm_Y , Perm_Z , Solid_Bulk_Modulus , Fluid_Bulk_Modulus}]
    translates to series of
    Real-ESSI DSL : add element $\# < . >$ type [27NodeBrick_up] with nodes ($< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$) use material $\# < . >$ and porosity $= < . >$ alpha $= < . >$ rho_s $= < M/L^3 >$ rho_f $= < M/L^3 >$ k_x $= < L^3T/M >$ k_y $= < L^3T/M >$ k_z $= < L^3T/M >$ K_s $= < stress >$ K_f $= < stress >$;
    operated over all the elements of the defined physical group

11. gmESSI : [Add_27NodeBrick_up_Variable_GaussPoints{PhysicalGroup , Num_Gauss_Points , material#1 , Porosity , Alpha , Solid_Density , Fluid_Density , Perm_X , Perm_Y , Perm_Z , Solid_Bulk_Modulus

, Fluid_Bulk_Modulus}]

translates to series of

Real-ESSI DSL : add element # $< . >$ type [27NodeBrick_up] using $< . >$ Gauss points each direction with nodes ($< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$) use material # $< . >$ and porosity $= < . >$ alpha $= < . >$ rho_s $= < M/L^3 >$ rho_f $= < M/L^3 >$ k_x $= < L^3T/M >$ k_y $= < L^3T/M >$ k_z $= < L^3T/M >$ K_s $= < stress >$ K_f $= < stress >$;

operated over all the elements of the defined physical group

12. gmESSI : [Add_Equal_Dof{PhysicalGroup , Dof_Type}]

    translates to series of

    Real-ESSI DSL : add constraint [equal_dof] with master node # $< . >$ and slave node # $< . >$ dof to constrain $< . >$;

    operated over all the elements of the defined physical group

13. gmESSI : [Add_Equal_Dof{PhysicalGroup , Master_Dof ,]

    translates to series of

    Real-ESSI DSL : add constraint [equal_dof] with node # $< . >$ dof $< . >$ master and node # $< . >$ dof $< . >$ slave;

    operated over all the elements of the defined physical group

14. gmESSI : [Add_ShearBeam{PhysicalGroup , CrossSection , material#1}]

    translates to series of

    Real-ESSI DSL : add element # $< . >$ type [ShearBeam] with nodes ($< . >$, $< . >$) cross_section $= < l^2 >$ use material # $< . >$;

    operated over all the elements of the defined physical group

15. gmESSI : [Add_DispBeamColumn3D{PhysicalGroup , Num_Integr_Points , Section_Number , Density , XZ_Plane_Vect_x , XZ_Plane_Vect_y , XZ_Plane_Vect_z , Joint1_Offset_x , Joint1_Offset_y , J1_z , Joint2_Offset_x , J2_y , J2_Offset_z}]

    translates to series of

    Real-ESSI DSL : add element # $< . >$ type [BeamColumnDispFiber3d] with nodes ($< . >$, $< . >$) number_of_integration_points $= < . >$ section_number $= < . >$ mass_density $= < M/L^3 >$ xz_plane_vector $= (< . >, < . >, < . >)$ joint_1_offset $= (< L >, < L >, < L >)$ joint_2_offset $= (< L >, < L >, < L >)$;

operated over all the elements of the defined physical group

16. gmESSI : [Add_Beam_Elastic{PhysicalGroup , Cross_Section , Elastic_Modulus , Shear_Modulus , Jx , Iy , Iz , Density , XZ_PlaneVect_x , XZ_PlaneVect_y , XZ_Plane_Vect_z , Joint1_Offset_x , Joint1_y , Joint1_Offset_z , Joint2_Offset_x , Joint2_Offset_y , J2_Offset_z}]
    translates to series of
    Real-ESSI DSL : add element # $< . >$ type [beam_elastic] with nodes ($< . >$, $< . >$) cross_section $= < area >$ elastic_modulus $= < F/L^2 >$ shear_modulus $= < F/L^2 >$ torsion_Jx $= < length^4 >$ bending_Iy $= < length^4 >$ bending_Iz $= < length^4 >$ mass_density $= < M/L^3 >$ xz_plane_vector $=$ ($< . >$, $< . >$, $< . >$) joint_1_offset $=$ ($< L >$, $< L >$, $< L >$) joint_2_offset $=$ ($< L >$, $< L >$, $< L >$);
    operated over all the elements of the defined physical group

17. gmESSI : [Add_Beam_Elastic_LumpedMass{PhysicalGroup , Cross_Section , Elastic_Modulus , Shear_Modulus , Jx , Iy , Iz , Density , XZ_Plane_Vect_x , XZ_Plane_Vect_y , XZ_Plane_Vect_z , Joint1_Offset_x , Joint1_Offset_y , Joint1_Offset_z , Joint2_Offset_x , Joint2_Offset_y , Joint2_Offset_z}]
    translates to series of
    Real-ESSI DSL : add element # $< . >$ type [beam_elastic_lumped_mass] with nodes ($< . >$, $< . >$) cross_section $= < area >$ elastic_modulus $= < F/L^2 >$ shear_modulus $= < F/L^2 >$ torsion_Jx $= < length^4 >$ bending_Iy $= < length^4 >$ bending_Iz $= < length^4 >$ mass_density $= < M/L^3 >$ xz_plane_vector $=$ ($< . >$, $< . >$, $< . >$) joint_1_offset $=$ ($< L >$, $< L >$, $< L >$) joint_2_offset $=$ ($< L >$, $< L >$, $< L >$);
    operated over all the elements of the defined physical group

18. gmESSI : [Add_Beam_DisplacementBased{PhysicalGroup , Num_Integration_Points , Section_Number , Density}]
    translates to series of
    Real-ESSI DSL : add element # $< . >$ type [beam_displacement_based] with nodes ($< . >$, $< . >$) with # $< . >$ integration_points use section # $< . >$ mass_density $= < M/L^3 >$ IntegrationRule = "" xz_plane_vector $=$ ($< . >$, $< . >$, $< . >$) joint_1_offset $=$ ($< L >$, $< L >$, $< L >$) joint_2_offset $=$ ($< L >$, $< L >$, $< L >$);
    operated over all the elements of the defined physical group

19. gmESSI : [Add_HardContact{PhysicalGroup , Normal_Stiffness , Tangential_Stiffness , Normal_Damping , Tangential_Damping , Friction_Ratio , Norm_Vect_x , Norm_Vect_y , Norm_Vect_z}]
    translates to series of

Real-ESSI DSL : add element # $< . >$ type [HardContact] with nodes ($< . >$, $< . >$) normal_stiffness = $< F/L >$ tangential_stiffness = $< F/L >$ normal_damping = $< F/L >$ tangential_damping = $< F/L >$ friction_ratio = $< . >$ contact_plane_vector = ($< . >$, $< . >$, $< . >$);
operated over all the elements of the defined physical group

20. gmESSI : [Add_CoupledHardContact{PhysicalGroup , Normal_Stiffness , Tangential_Stiffness , Normal_Damping , Tangential_Damping , Friction_Ratio , Norm_Vect_x , Norm_Vect_y , Norm_Vect_z}]
translates to series of
Real-ESSI DSL : add element # $< . >$ type [CoupledHardContact] with nodes ($< . >$, $< . >$) normal_stiffness = $< F/L >$ normal_penalty_stiffness = $< F/L >$ tangential_stiffness = $< F/L >$ normal_damping = $< F/L >$ tangential_damping = $< F/L >$ friction_ratio = $< . >$ contact_plane_vector = ($< . >$, $< . >$, $< . >$);
operated over all the elements of the defined physical group

21. gmESSI : [Add_SoftContact{PhysicalGroup , Initial_Normal_Stiffness , Stiffning_Rate , Maximum_Normal_Stiffness , Tangential_Stiffness , Normal_Damping , Tangential_Damping , Friction_Ratio , Norm_Vect_x , Norm_Vect_y , Norm_Vect_z}]
translates to series of
Real-ESSI DSL : add element # $< . >$ type [SoftContact] with nodes ($< . >$, $< . >$) initial_normal_stiffness = $< F/L >$ stiffening_rate = $< 1/L >$ max_normal_stiffness = $< F/L >$ tangential_stiffness = $< F/L >$ normal_damping = $< F/L >$ tangential_damping = $< F/L >$ friction_ratio = $< . >$ contact_plane_vector = ($< . >$, $< . >$, $< . >$);
operated over all the elements of the defined physical group

22. gmESSI : [Add_CoupledSoftContact{PhysicalGroup , Initial_Normal_Stiffness , Stiffning_rate , Maximum_Normal_Stiffness , Tangential_Stiffness , Normal_Damping , Tangential_Damping , Friction_Ratio , Norm_Vect_x , Norm_Vect_y , Norm_Vect_z}]
translates to series of
Real-ESSI DSL : add element # $< . >$ type [CoupledSoftContact] with nodes ($< . >$, $< . >$) initial_normal_stiffness = $< F/L >$ stiffening_rate = $< 1/L >$ max_normal_stiffness = $< F/L >$ tangential_stiffness = $< F/L >$ normal_damping = $< F/L >$ tangential_damping = $< F/L >$ friction_ratio = $< . >$ contact_plane_vector = ($< . >$, $< . >$, $< . >$);
operated over all the elements of the defined physical group

23. gmESSI : [Add_Truss{PhysicalGroup , material#1 , Cross_Sectin , Density}]
translates to series of

Real-ESSI DSL : add element # $< . >$ type [truss] with nodes ($< . >$, $< . >$) use material # $< . >$ cross_section = $< length^2 >$ mass_density = $< M/L^3 >$;
operated over all the elements of the defined physical group

24. gmESSI : [Add_8NodeBrick{PhysicalGroup , material#1}]
translates to series of
Real-ESSI DSL : add element # $< . >$ type [8NodeBrick] with nodes ($< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$) use material # $< . >$;
operated over all the elements of the defined physical group

25. gmESSI : [Add_Cosserat8NodeBrick{PhysicalGroup , material#1}]
translates to series of
Real-ESSI DSL : add element # $< . >$ type [Cosserat8NodeBrick] with nodes ($< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$) use material # $< . >$;
operated over all the elements of the defined physical group

26. gmESSI : [Add_8NodeBrick_Variable_GaussPoints{PhysicalGroup , NumGaussPoints , material#1}]
translates to series of
Real-ESSI DSL : add element # $< . >$ type [8NodeBrick] using $< . >$ Gauss points each direction with nodes ($< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$) use material # $< . >$;
operated over all the elements of the defined physical group

27. gmESSI : [Add_8NodeBrick_upU{PhysicalGroup , material#1 , Porosity , Alpha , Solid_Density , Fluid_Density , Perm_X , Perm_Y , Perm_Z , Solid_Bulk_Modulus , Fluid_Bulk_Modulus}]
translates to series of
Real-ESSI DSL : add element # $< . >$ type [8NodeBrick_upU] with nodes ($< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$) use material # $< . >$ porosity = $< . >$ alpha = $< . >$ rho_s = $< M/L^3 >$ rho_f = $< M/L^3 >$ k_x = $< L^3T/M >$ k_y = $< L^3T/M >$ k_z = $< L^3T/M >$ K_s = $< stress >$ K_f = $< stress >$;
operated over all the elements of the defined physical group

28. gmESSI : [Add_8NodeBrick_upU_Variable_GaussPoints{PhysicalGroup , Num_Gauss_Points , material#1 , Porosity , Alpha , Solid_Density , Fluid_Density , Perm_X , Perm_Y , Perm_Z , Solid_Bulk_Modulus , Fluid_Bulk_Modulus}]
translates to series of
Real-ESSI DSL : add element # $< . >$ type [8NodeBrick_upU] using $< . >$ Gauss points each direction with nodes ($< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$) use material #

$< . >$ porosity $= < . >$ alpha $= < . >$ rho_s $= < M/L^3 >$ rho_f $= < M/L^3 >$ k_x $= < L^3T/M >$ k_y $= < L^3T/M >$ k_z $= < L^3T/M >$ K_s $= < stress >$ K_f $= < stress >$;
operated over all the elements of the defined physical group

29. gmESSI : [Add_8NodeBrick_up{PhysicalGroup , material#1 , Porosity , Alpha , Solid_Density , Fluid_Density , Perm_X , Perm_Y , Perm_Z , Solid_Bulk_Modulus , Fluid_Bulk_Modulus}]
translates to series of
Real-ESSI DSL : add element # $< . >$ type [8NodeBrick_up] with nodes ($< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$) use material # $< . >$ porosity $= < . >$ alpha $= < . >$ rho_s $= < M/L^3 >$ rho_f $= < M/L^3 >$ k_x $= < L^3T/M >$ k_y $= < L^3T/M >$ k_z $= < L^3T/M >$ K_s $= < stress >$ K_f $= < stress >$;
operated over all the elements of the defined physical group

30. gmESSI : [Add_8NodeBrick_up_Variable_GaussPoints{PhysicalGroup , Num_Gauss_Points , material#1 , Porosity , Alpha , Solid_Density , Fluid_Density , Perm_X , Perm_Y , Perm_Z , Solid_Bulk_Modulus , Fluid_Bulk_Modulus}]
translates to series of
Real-ESSI DSL : add element # $< . >$ type [8NodeBrick_up] using $< . >$ Gauss points each direction with nodes ($< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$) use material # $< . >$ porosity $= < . >$ alpha $= < . >$ rho_s $= < M/L^3 >$ rho_f $= < M/L^3 >$ k_x $= < L^3T/M >$ k_y $= < L^3T/M >$ k_z $= < L^3T/M >$ K_s $= < stress >$ K_f $= < stress >$;
operated over all the elements of the defined physical group

207.2.5.6 Elemental Compound Commands : Operates On All Surface Elements of the defined Physical Group [Surface Loads]

Elemental Compound Commands operates on two physical groups, one for surface and another for the element on which surface is present. It is used mainly for adding surface loads, which require surface number as well as element no in Real-ESSI DSL. For example:- [Add_8NodeBrick_SurfaceLoad{Physical_Group#Volume, Physical_Group#Surface, 10*Pa}] would initialize surface load of $10Pa$ on surfaces defined by physical_group 'Surface' on elements defined by physical_group 'Volume'.

The different commands under this category and their corresponding ESSI commands are listed below

NOTE:- Every Elemental commands get translated into the *load.fei* (Section 207.2.4.5).

1. gmESSI : [Add_20NodeBrick_SurfaceLoad{PhysicalGroup#Volume , PhysicalGroup#Surface , Pressure}]
   translates to series of
   Real-ESSI DSL : add load # $< . >$ to element # $< . >$ type [surface] at nodes ($< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$) with magnitude $< Pa >$;
   operated over all the elements of the defined physical group

2. gmESSI : [Add_20NodeBrick_SurfaceLoad{PhysicalGroup#Volume , PhysicalGroup#Surface , Press1 , Press2 , Press3 , Press4 , Press5 , Press6 , Press7 , Press8}]
   translates to series of
   Real-ESSI DSL : add load # $< . >$ to element # $< . >$ type [surface] at nodes ($< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$) with magnitudes ($< Pa >$, $< Pa >$, $< Pa >$, $< Pa >$, $< Pa >$, $< Pa >$, $< Pa >$, $< Pa >$);
   operated over all the elements of the defined physical group

3. gmESSI : [Add_27NodeBrick_SurfaceLoad{PhysicalGroup#Volume , PhysicalGroup#Surface , Pressure}]
   translates to series of
   Real-ESSI DSL : add load # $< . >$ to element # $< . >$ type [surface] at nodes ($< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$) with magnitude $< Pa >$;
   operated over all the elements of the defined physical group

4. gmESSI : [Add_27NodeBrick_SurfaceLoad{PhysicalGroup#Volume , PhysicalGroup#Surface , Press1 , Press2 , Press3 , Press4 , Press5 , Press6 , Press7 , Press8 , Press9}]
   translates to series of

Real-ESSI DSL : add load $\# < . >$ to element $\# < . >$ type [surface] at nodes ($< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$, $< . >$) with magnitudes ($< Pa >$, $< Pa >$, $< Pa >$, $< Pa >$, $< Pa >$, $< Pa >$, $< Pa >$, $< Pa >$, $< Pa >$);
operated over all the elements of the defined physical group

5. gmESSI : [Add_8NodeBrick_SurfaceLoad{PhysicalGroup#Volume , PhysicalGroup#Surface , Pressure}]
   translates to series of
   Real-ESSI DSL : add load $\# < . >$ to element $\# < . >$ type [surface] at nodes ($< . >$, $< . >$, $< . >$, $< . >$) with magnitude $< Pa >$;
   operated over all the elements of the defined physical group

6. gmESSI : [Add_8NodeBrick_SurfaceLoad{PhysicalGroup#Volume , PhysicalGroup#Surface , Press1 , Press2 , Press3 , Press4}]
   translates to series of
   Real-ESSI DSL : add load $\# < . >$ to element $\# < . >$ type [surface] at nodes ($< . >$, $< . >$, $< . >$, $< . >$) with magnitudes ($< Pa >$, $< Pa >$, $< Pa >$, $< Pa >$);
   operated over all the elements of the defined physical group

207.2.5.7 Special Commands

The translator supports some special commands to perform some special functions that are regularly required in simulations. It supports the Connect Command (Section 207.2.5.8) allows to join or create nodes between two physical groups.

207.2.5.8 Connect Command

Connect Commands creates/find layers of 2-noded elements between any two parallel geometrical physical entities like two lines, two surface or two volumes and creates a physical group of those elements and updates this information in the *XYZ.msh* file. Since gmsh does not include the feature of defining or creating 2-noded elements after the mesh creation, this command can be very useful in that case. For example;- defining contacts/interfaces, embedded piles, boundary conditions, connections etc. The command syntax for connect command is
gmESSI :: [Connect{Physical_Group#tag_From , Physical_Group#tag_To, Physical_Group#tag_Between, dir_vect, mag, no_times, $algo\_(find|create)$, tolerance,New_Physical_Group_Name}]

- Physical_Group#tag_From :: It defines the starting nodes
- Physical_Group#tag_To :: It defines the set of end nodes
- Physical_Group#tag_Between:: It defines the set of nodes where the intermediary nodes can be found, while searching. While creating nodes, it does not play any role.
- dir_vect :: It defines the direction in which the user wants to create or find the nodes. The direction vector argument is given as {x_comp \y_comp \z_comp}. Example:- {0 \0 \-1} , {1 \1 \0} .. etc.
- mag :: It defines the length of each 2-noded line elements
- no_times :: It defines number of layers of 2-noded elements, the user want to create/find
- algo_(find/create) :: It defines the algo which is either 0 or 1 meaning whether to find or create the intermediary node
- tolerance :: It defines the tolerance is required to finding the nodes. It should be less than the minimum of the distance of neighboring nodes.
- New_Physical_Group_Name :: This argument enables the user to give a name to the 2-noded new-physical group formed

Figure 207.7 graphically describes arguments of connect command.
This command updates and creates additional nodes and 2-noded elements and also assigns a physical group name "$New_Physical_Group_Name$". gmESSI automatically adds the next id available to the new physical group. The user can then manipulate this newly created physical group with any other gmESSI commands.

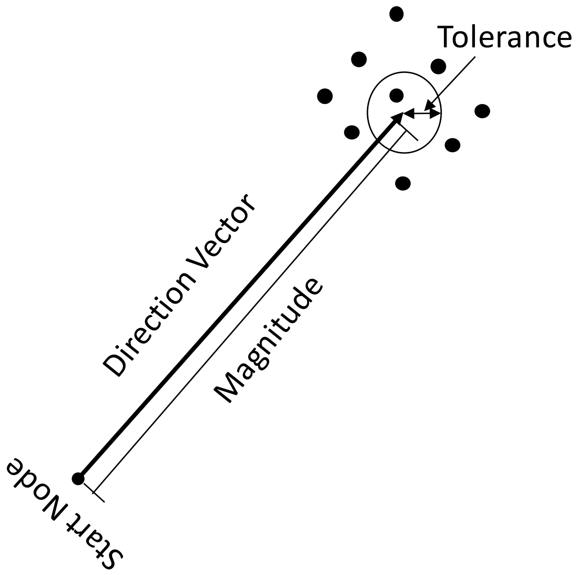


Figure 207.7: Pictorial representation of working of connect command.

The working of this command would be more clear through examples. [Example_3] can be downloaded here. [Example_4] can be downloaded here. These examples describes two situation one where new nodes are to be created and the other where already present nodes needs to be found respectively. In both the cases 2-noded line elements are always created. The examples can also be alternatively located in Examples folder of gmESSI directory

[Example_3] is a simple example where a tower of certain height above ground surface and also its base embedded in soil is modeled. It starts with a mesh file that creates a node for tower at a certain height and then using algorithm -'create' new nodes are created at certain intervals to generate the beam elements. On the other hand, the embedded beam is created by "-find" algorithm. Let us look at the [Example_3.geo] file.

```
$ cat Example_3.geo

// Size of the soil block in meter
Size = 10;

// Height of the Tower in meter
Height = 6;

// Mesh Size of the soil block
Mesh_Size = 1;

// Adding Points and extruding
Point(1)={-Size/2,-Size/2,-Size/2};
```

```
Extrude{Size,0,0}{Point{1};Layers{Size/Mesh_Size};Recombine;}
Extrude{0,Size,0}{Line{1};Layers{Size/Mesh_Size};Recombine;}
Extrude{0,0,Size}{Surface{5};Layers{Size/Mesh_Size};Recombine;}

// Make the tower located at height 6 m from the ground surface
Tower = newp;
Point (Tower) = {0,0,Size/2+Height};

//// Create Physical Groups
Physical Volume ("Soil") = {1};
Physical Surface ("Soil_Base_Surface") ={5};
Physical Surface ("Soil_Top_Surface") ={27};
Physical Point ("Tower") = {Tower};
```

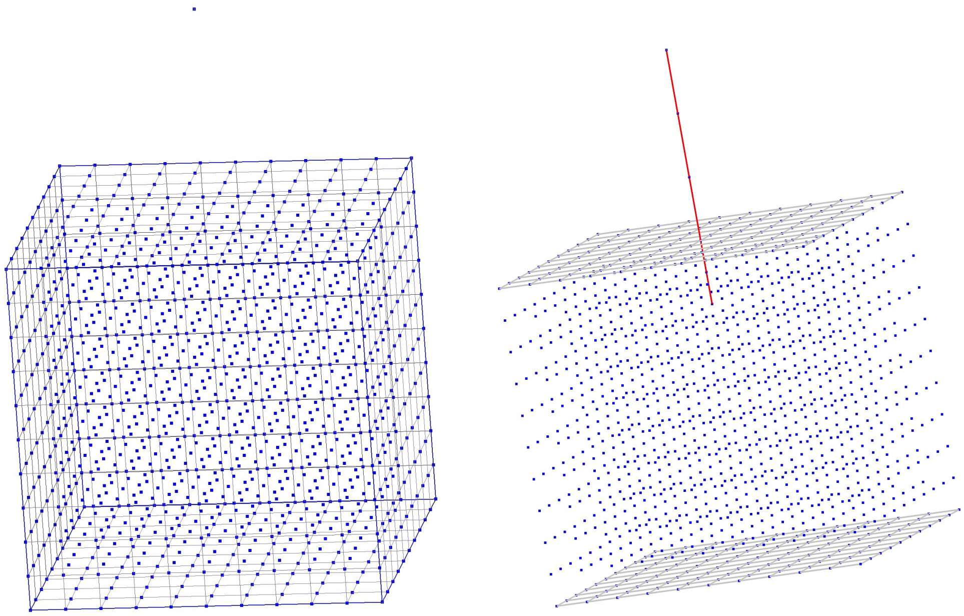
Running [Example_3.gmessi] with the .msh output of geometry file would produces additional nodes and elements as shown in Figure 207.8. An excerpt showing use of connect command with create algo in [Example_3.gmessi] is shown below. The effect of the command is shown in Figure 207.8.

```
$ cat Example_3.gmessi
..........
[Connect{Physical_Group#Tower, Physical_Group#Soil_Top_Surface, ↩
   Physical_Group#Soil_Top_Surface, dv1:= 0 \ 0 \ -1, mag:= 2, Tolerance:= 0, ↩
   algo:= create, noT:= 3, PhysicalGroupName:= Tower_Beam_Above_Ground}]
..............
```

The terminal displays the information about number of elements and nodes created and also displays the information about the new physical group information i.e id and name. The new physical group creation can be seen in the [Example_3.gmsh] in *Example_3_ESSI_Simulation* folder. The terminal message and mesh file is shown below. It also displays error message if more than one node is found in the tolerance provided.

```
$ gmessy Example_3.gmessi
New Physical Group "Tower_Beam_Above_Ground" having id 5 consisting of 4 Nodes ↩
   and 3 2-noded elements created
```

```
$ cat Example_3_ESSI_Simulation\Example_3.msh
............
$PhysicalNames
7
0 4 "Tower"
2 2 "Soil_Base_Surface"
2 3 "Soil_Top_Surface"
3 1 "Soil"
1 5 "Tower_Beam_Above_Ground"
3 6 "TowerBaseNode"
1 7 "Tower_Embedded_Beam"
$EndPhysicalNames
```



(a) Initial mesh file Example_3.msh generated by gmsh. The dir vector is in Z axis 0,0,-1

(b) Final mesh after gmESSI

Figure 207.8: Example 3 Contact Problem. (b) shows the nodes and elements generated by gmESSI Translator.

```
...........
```

[Example_4] describes a foundation on soil problem with contact/interface between them. The contact element is created with the help of comment command using algo "-find". Let us look at the [Example_4.geo] file.

```
$ cat Example_4.geo

// Size of the soil block
Size = 1;

// Thickness of Foundation
Thick = 0.1;
Foundation_Layers = 2;

//// Mesh Size of the block
Mesh_Size = 0.2;

// Adding Points an1 extruding
Point(1)={-Size/2,-Size/2,-Size/2};
Extrude{Size,0,0}{Point{1};Layers{Size/Mesh_Size};Recombine;}
```

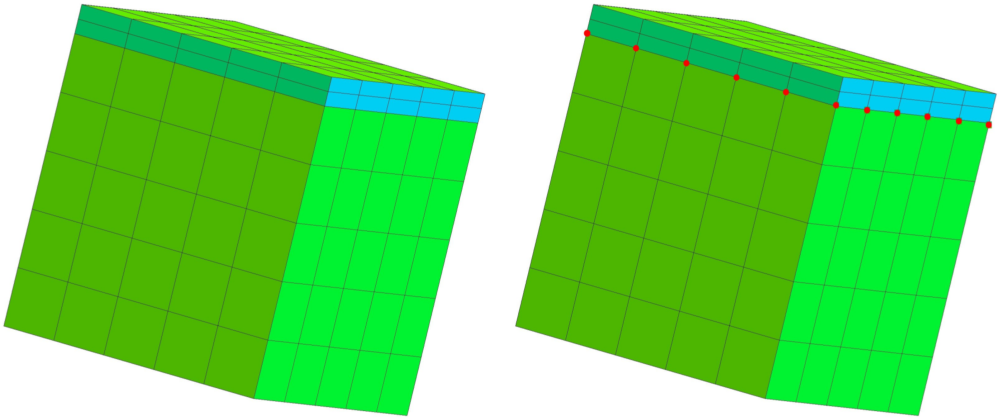
```
Extrude{0,Size,0}{Line{1};Layers{Size/Mesh_Size};Recombine;}
Extrude{0,0,Size}{Surface{5};Layers{Size/Mesh_Size};Recombine;}

// Make sure in Tools -> Geometry -> General
// Geometry tolerance is set smaller than Epsilon
// such as Geometry tolerance = 1e-14

Epsilon = 1e-8;
Translate {0, 0, Epsilon} {Duplicata{Surface{27};}}
Transfinite Line {29,30,31,32} = Size/Mesh_Size +1;
Transfinite Surface {28};
Recombine Surface {28};

//// Extruding the surface to foundation thickness
Extrude{0,0,Thick}{Surface{28};Layers{Foundation_Layers};Recombine;}

//// Create Physical Groups
Physical Volume ("Soil") = {1};
Physical Surface ("Soil_Base_Surface") ={5};
Physical Surface ("Soil_Top_Surface") ={27};
Physical Surface ("Foundation_Base_Surface")={28};
Physical Surface ("Foundation_Top_Surface") ={54};
Physical Volume ("Foundation") = {2};


Physical Surface("Fix_X") = {26, 53, 45, 18};
Physical Surface("Fix_Y") = {22, 49, 14, 41};
Physical Volume("3_Dofs") = {1,2};
```

The above geometry file is then meshed with gmsh to get the .msh file. In this file, the connect command is applied between physical group *Foundation_Base_Surface* and *Soil_Top_Surface* to create contact/interface elements. The corresponding connect command would be as

```
$ gmessy Example_4.gmessi
..........
[Connect{Physical_Group#Soil_Top_Surface, ↩
   Physical_Group#Foundation_Base_Surface, ↩
   Physical_Group#Foundation_Base_Surface, dv1:= 0\0\1, mag:= 0, Tolerance:= ↩
   0.001, algo:= find, noT:= 1, PhysicalGroupName:= Contact_Elements}]
..............
```

Similarly the updated [Example_4.msh] contains the new physical group and terminal shows the new physical group of 2-noded elements created. Figure 207.9 shows the new nodes found and creation of 2-noded elements.

```
$ gmessy Example_3.gmessi
New Physical Group "Contact_Elements" having id 10 consisting of 72 Nodes and ↩
   36 2-noded elements created
```



(a) Initial mesh file Example4.msh generated by gmsh. The dir vector is in Z axis {0,0,1}

(b) New physical group Contact_Element

Figure 207.9: Example 4 finding nodes problem.(b) shows the nodes and elements generated by gmESSI Translator.

```
$ cat Example_4_ESSI_Simulation\Example_4.msh
10
2 2 "Soil_Base_Surface"
2 3 "Soil_Top_Surface"
2 4 "Foundation_Base_Surface"
2 5 "Foundation_Top_Surface"
2 7 "Fix_X"
2 8 "Fix_Y"
3 1 "Soil"
3 6 "Foundation"
3 9 "3_Dofs"
1 10 "Contact_Elements"
............
```

NOTE : Since the algo is to only find the nodes, so no new nodes are created, but only elements are created. The same message can be seen on the terminal.

207.2.5.9 Write Command

Write command takes filename as an argument and writes the content of a physical group in two separate files one containing all the nodes info and other containing all the elements info and places in the same XYZ_ESSI_Simulation folder. The command syntax is gmESSI:: [Write_Data{PhyEntyTag,filename}]

- Creates files *XYZ_filename_Nodes.txt* and *XYZ_filename_Elements.txt*
- XYZ_filename_Nodes.txt :: Contains data for all nodes in a physical group. Each node data is represented in one line as

  Node_no x_coord y_cord z_cord

  with meanings as usual.
- XYZ_filename_Elements.txt :: Contains data for all elements in a physical group.Each element data is represented in one line as

  Element_no Element_type node1 node2 node3 ..

  with meanings as usual. Element_type refers to the same as in Gmsh Manual.

[Example_4.gmessi] shows the usage of write command.

207.2.5.10 Write DRM HDF5 Command

Domain reduction method (DRM) is a very useful method to input 3D seismic excitations into earthquake soil structure interacting system. With a defined physical group as DRM layer, a HDF5 file containing geometric information of the DRM layer, can be generated with the following commands for 1D, 2D and 3D mesh, respectively:

- gmESSI::[Generate_DRM_HDF5_1D{Physical_Group#<PhyEnty Name or Tag>, Surface_Normal:= <X | Y | Z>, Node_Coordinate_Tol:=<tolerance>, FileName:=<HDF5 file name>}]
- gmESSI::[Generate_DRM_HDF5_2D{Physical_Group#<PhyEnty Name or Tag>, Surface_Plane:=<XY|XZ|YZ>, Surface_Normal:= <X | Y | Z>, Node_Coordinate_Tol:=<tolerance>, FileName:=<HDF5 file name>}]
- gmESSI::[Generate_DRM_HDF5_3D{Physical_Group#<PhyEnty Name or Tag>, Surface_Normal:= <X | Y | Z>, Node_Coordinate_Tol:=<tolerance>, FileName:=<HDF5 file name>}]

Where:

- Physical_Group# defines the physical group name or tag for the DRM layer.
- Surface_Normal:= defines the surface normal direction of the DRM layer. It can be X or Y or Z.
- Surface_Plane:= defines the surface plane of the 2D DRM layer. It can be XY or YZ or XZ.

- Node_Coordinate_Tol:= defines the tolerance to distinguish two different DRM nodes. The tolerance should be much smaller than the FEM mesh size!
- FileName:= defines the file name of the HDF5 file to be generated.

207.2.6 Steps For Using gmESSI tool

Using gmESSI it is very easy to convert a .msh file to ESSI (.fei) file. This section guides the user through a simple [Example_1.geo], to show the steps necessary for generating Real-ESSI files directly from .msh file through gmESSI. Lets define a problem as shown in Figure 207.10. The [Example_1.geo] can be located in the gmESSI ''Examples' directory. Alternatively, it can be downloaded here.

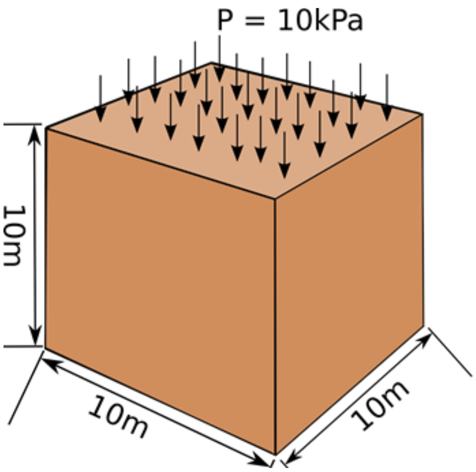


Figure 207.10: Example_1 description of a block of soil with surface load.

It is a block of dimension $10m \times 10m \times 10m$ of soil mass whose all 4 lateral faces are fixed in ux, uy dofs. The bottom face is fixed in ux, uy, uz dofs. A uniform pressure surface load of $10Pa$ is applied. The density and elastic modulus of the soil increases from $2000 * kg/m^3$ and Young's modulus is taken as $200MPa$ as shown in Figure 207.10.

207.2.6.1 Building geometry (.geo) file in Gmsh

The first step is to make the geometry file in *Gmsh*. While creating the geometry the user should also define all the physical groups on which they intend to either apply boundary condition, define elements, loads etc. In [Example_1.geo], 3 physical groups are needed : one for applying surface load, one for fixities, and one for defining the soil volume and assigning material. The content of [Example_1.geo] file is shown below

```
$cat Example_1.geo

// Size of the block
Size = 10;

//// Mesh Size of the block
Mesh_Size = 2;
```

```

// Adding Points and extruding
Point(1)={-Size/2,-Size/2,-Size/2};
Extrude{Size,0,0}{Point{1};Layers{Size/Mesh_Size};Recombine;}
Extrude{0,Size,0}{Line{1};Layers{Size/Mesh_Size};Recombine;}
Extrude{0,0,Size}{Surface{5};Layers{Size/Mesh_Size};Recombine;}

//// Create Physical Groups
Physical Volume ("Soil") = {1};
Physical Surface ("Base_Surface") = {5};
Physical Surface ("Lateral_Surface") = {18,22,14,26};
Physical Surface ("Top_Surface") ={27};
```

207.2.6.2 Generate mesh (.msh) file in Gmsh

Once .geo file is ready with all the physical groups, next step is to mesh the model. The mesh operation will generate the mesh file (.msh) that contains all the mesh information.

The model can be meshed from the terminal directly by running:

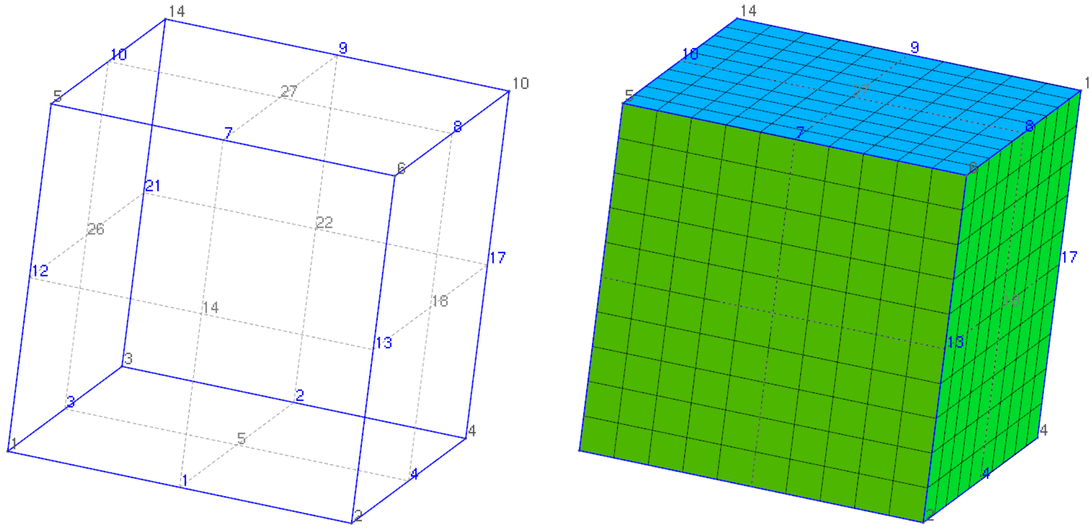
```
gmsh Example_1.geo -3
```

Here -3 means we are meshing a 3D object, which will automatically mesh all the 3D volumes, 2D surfaces and 1D lines object defined in the geometry model. If there are only 2D surfaces and/or 1D lines object defined in the geometry (.geo) file, use -2 instead. If there are only 1D lines object defined in the geometry (.geo) file, use -1 instead.

A quick look at the generated [Example_1.msh] file containing physical groups is shown below:

```
$cat Example_5.geo
..............
$PhysicalNames
4
2 2 "Base_Surface"
2 3 "Lateral_Surface"
2 4 "Top_Surface"
3 1 "Soil"
$EndPhysicalNames
................
```

Figure 207.11 shows the geometry and mesh visualization in Gmsh. It is noted that Gmsh performs meshing for linear interpolation elements by default. In other words, the above cubic block geometry object is meshed into eight-node bricks, that have linear isoparametric interpolation, `8NodeBrick`. For higher order interpolation meshing options, that is for meshing twenty-seven node brick elements mesh, `27NodeBrick` for example, additional -order int should be used. int here is the integer specifying the



(a) Geometry File

(b) Mesh File

Figure 207.11: Gmsh geometry and mesh file for Example_1

order of meshing. For example, the following terminal command with -order 2, i.e., 2^{nd} order meshing, generates twenty-seven node brick meshes (`27NodeBrick`):

```
gmsh Example_1.geo -3 -order 2
```

207.2.6.3 Writing all gmESSI Commands for the model

Using gmESSI for mesh conversion is very easy. To achieve this, a [Example_1.gmessi] file is created containing all the required gmESSI commands to be executed sequentially. Let us look at each of them

Since physical group names and ids are required for referring the gmESSI commands, its always best to copy all the physical group data from the .msh file (in this case *[Example_1.msh]* file) in the header of .gmessi file, so that its easier for th user to refer to the physical groups while writing commands in .gmessi file. The contents of the .gmessi file are shown below.

```
$cat Example_1.gmessi

### Physical Groups defined in the msh file.
#2 2 "Base_Surface"
#2 3 "Lateral_Surface"
#2 4 "Top_Surface"
#3 1 "Soil"

### loading the gmsh file
```

```
gmESSI.loadGmshFile("Example_1.msh")

### Defining the Simulation Directory and node, element, load and main file
### Its important to define the directory and these files at the beginning of ↩
    any gmESSI command conversion
### 1 refers as overwrite mode ( will overwrite the directory if present) --- 0 ↩
    would not overwrite
gmESSI.setSimulationDir("./Example_1_ESSI_Simulation",1)
gmESSI.setMainFile(gmESSI.SimulationDir+ "main.fei")
gmESSI.setNodeFile(gmESSI.SimulationDir+ "node.fei")
gmESSI.setElementFile(gmESSI.SimulationDir+ "element.fei")
gmESSI.setLoadFile(gmESSI.SimulationDir+ "load.fei")

#### // is used to provide commands and gets translated in the main.fei file
#### Also, the commands followed by exclamation '!' get directly copied to the ↩
    main.fei file
#### Usually, the user would write Real-ESSI DSL against the exclamation mark.

// My new model
! model name "Soil_Block";

[Add_All_Node{ unit:= m, nof_dofs:= 3}]

// Adding Material also assigning it to elements
! add material #1 type linear_elastic_isotropic_3d_LT mass_density = ↩
    2000*kg/m^3 elastic_modulus = 200*MPa poisson_ratio = 0.3;
[Add_8NodeBrick{Physical_Group#Soil, MaterialNo:= 1}]

! include "node.fei";
! include "element.fei";
! new loading stage "Stage1_Self_Weight";

# Applying Fixities
[Fix_Dofs{Physical_Group#Base_Surface, all}]
[Fix_Dofs{Physical_Group#Lateral_Surface, ux uy}]

#### For applying Self-Weight Load to the soil elements
! add acceleration field # 1 ax = 0*g ay = 0*g az = -1*g ;
! add load #1 to all elements type self_weight use acceleration field # 1;

 #Updating the tag inside gmESSI as user entered by himself load tag
gmESSI.setESSITag("load",2)


! include "load.fei";
! NumStep = 10;
!
! define algorithm With_no_convergence_check;
! define solver UMFPack;
! define load factor increment 1/NumStep;
! simulate NumStep steps using static algorithm;
```

```


#### updating the new load file before new loading stage
gmESSI.setLoadFile(gmESSI.SimulationDir+ "Surface_Load.fei")
! new loading stage "Stage2_Surface_Loading";

### For applying Surface load on the Top Surface of the Soil Block
[Add_8NodeBrick_SurfaceLoad{Physical_Group#Soil,Physical_Group#Top_Surface,10*Pa}]

##### For applying Nodal loads to all the nodes of the top surface
#[Add_Node_Load_Linear{Physical_Group#Top_Surface, ForceType:= Fx, Mag:= 10*kN}]


! include "Surface_Load.fei";
! NumStep = 10;
!
! define algorithm With_no_convergence_check;
! define solver UMFPack;
! define load factor increment 1/NumStep;
! simulate NumStep steps using static algorithm;

! bye;
```

NOTE: The gmESSI commands are executed and written to the file sequentially, so the user should be careful with the order of translation.

207.2.6.4 Executing gmESSI on Example_1.gmessi input file

Once .gmessi input file is ready, the next task is to run it using the ‘gmessy’ command in terminal. Running would carryout the translation to all and produce the log of translation, displayed on the terminal

```
$ gmessy Example_1.gmessi

Message:: newDirectory created as ./Example_1_ESSI_Simulation

Add_All_Node{ unit:= m, nof_dofs:= 3}
   Found!!
   Successfully Converted

Add_8NodeBrick{Physical_Group#Soil, MaterialNo:= 1}
   Found!!
   Successfully Converted

Fix_Dofs{Physical_Group#Base_Surface, all}
   Found!!
   Successfully Converted
```

```

Fix_Dofs{Physical_Group#Lateral_Surface, ux uy}
   Found!!
   Successfully Converted

Add_8NodeBrick_SurfaceLoad{Physical_Group#Soil,Physical_Group#Top_Surface,10*Pa}
   Found!!
   Successfully Converted


************************ Updated New Tag Numbering **********************
damping = 1
displacement = 1
element = 126
field = 1
load = 27
material = 1
motion = 126
node = 217
nodes = 217
Gmsh_Elements = 276
Gmsh_Nodes = 217
```

It would create a folder [Example_1_ESSI_Simulation] and places load.fei, node.fei, element.fei and main.fei files. The user at this point do not need to write anything in the Example_5_analysis.fei file as every command was sequentially written down in .gmessi file and is converted. The content of the main.fei is shown below.

```
$cat Example_5_analysis.fei
// My new model
 model name "Soil_Block";

// Adding Material also assigning it to elements
 add material #1 type linear_elastic_isotropic_3d_LT mass_density = 2000*kg/m^3 elastic_modulus = 200*MPa poisson_ratio = 0.3;

 include "node.fei";
 include "element.fei";
 new loading stage "Stage1_Self_Weight";


 add acceleration field # 1 ax = 0*g ay = 0*g az = -1*g ;
 add load #1 to all elements type self_weight use acceleration field # 1;
 include "load.fei";
 NumStep = 10;

  define algorithm With_no_convergence_check;
  define solver UMFPack;
  define load factor increment 1/NumStep;
  simulate NumStep steps using static algorithm;
```

```
new loading stage "Stage2_Surface_Loading";

include "Surface_Load.fei";
NumStep = 10;

 define algorithm With_no_convergence_check;
 define solver UMFPack;
 define load factor increment 1/NumStep;
 simulate NumStep steps using static algorithm;
bye;
```

207.2.6.5 Running Real-ESSI and visualization in paraview

With all files ready in their place, the next step is to run the main.fei file directly in ESSI.

```
$essi -f main.fei
```

Running ESSI creates .feioutput file which can be visualized in paraview using PVESSIReader plugin. Figure 207.12 shows the visualization of hdf5 output produced in paraview.

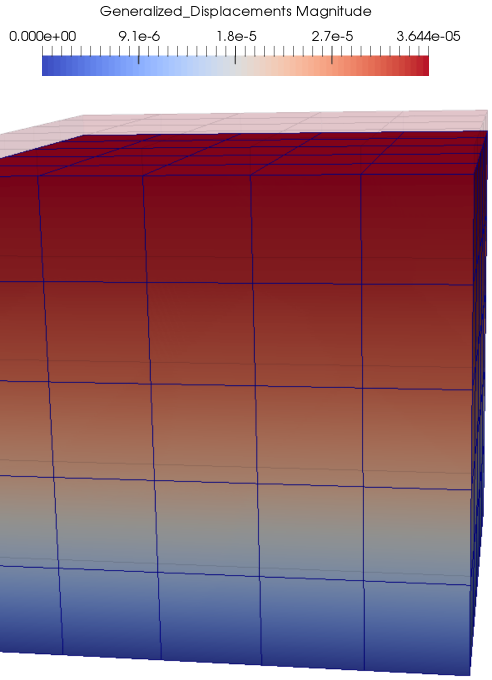


Figure 207.12: Visualizing output in Paraview.

207.2.7 Illustrative Examples

The *Examples* directory of gmESSI folder contains five examples as Example_1,Example_2.... and Example_5. They are summarized as

1. [Example_1] : Modeling of Surface load on block of Soil. The geometry (.geo), mesh (.msh) and .gmessi input files can be downloaded HERE.
2. [Example_2] : Modeling of Cantilever Beam. The geometry (.geo), mesh (.msh) and .gmessi input files can be downloaded HERE.
3. [Example_3] : Modeling of Tower (beam) located above the ground and embedded in soil using contact/interface elements. The geometry (.geo), mesh (.msh) and .gmessi input files can be downloaded HERE.
4. [Example_4] : Modeling of a concrete foundation on Soil connected by contact elements. The geometry (.geo), mesh (.msh) and .gmessi input files can be downloaded HERE.
5. [Example_5] : Modeling of a embedded shells and beam in Solids. The geometry (.geo), mesh (.msh) and .gmessi input files can be downloaded HERE.

[Example_1] was discussed in the previous section. Examples 1 to 4 are discussed and refereed in the manual at several instances. The user is encouraged to over these examples and learn to create geometry '.geo' and .gmessi input files. Here, two examples Example_2 about cantilever beam analysis and Example_5 about beams and shell is discussed.

207.2.7.1 Modeling of Cantilever Beam With Surface Load [Example_2]

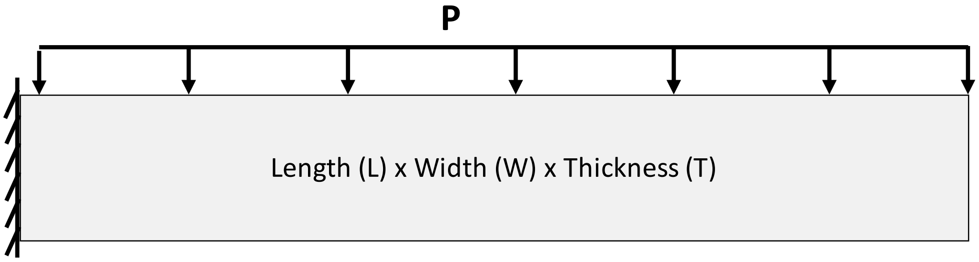


Figure 207.13: Illustration of the cantilever problem.

The problem consist of a cantilever beam with its left end fixed. A uniform surface load of P is applied. The geometry (.geo) and the gmessi input file for this problem can be downloaded here. Figure 207.14 shows visualization of output after running the model in Real-ESSI.

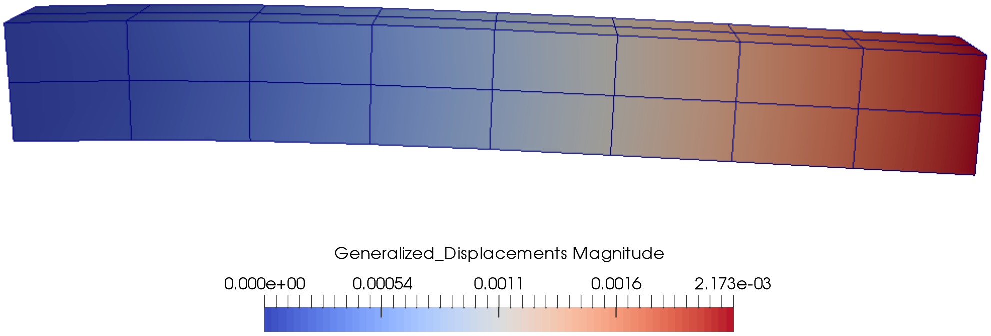


Figure 207.14: Illustration of the cantilever problem.

207.2.7.2 Modeling of a embedded shells and beam in Solids [Example_5]

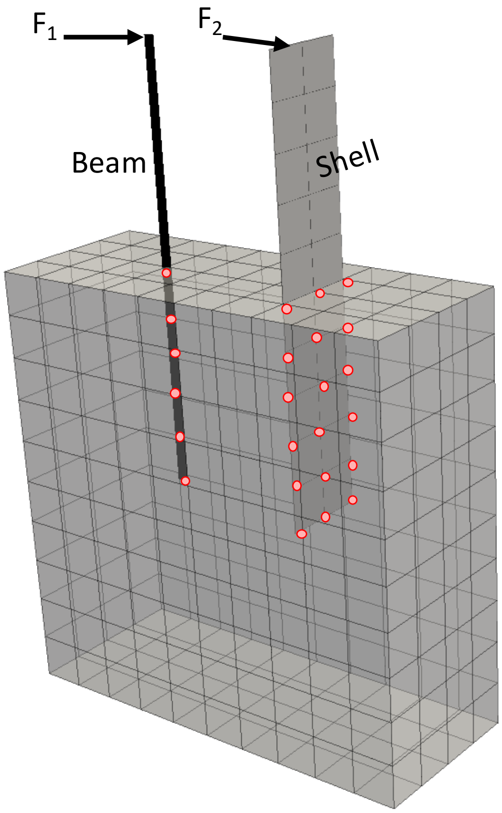


Figure 207.15: Illustration of the cantilever problem.

The problem consist of solid of 3 dofs in which beams and shells of 6dofs are embedded. The embedded beams and shell elements are connected by contact/interface elements. A nodal load to the top of the beam and shell is applied. The geometry (.geo) and the gmessi input file for this problem can be downloaded here. Figure 207.16 shows visualization of output after running the model in Real-ESSI.

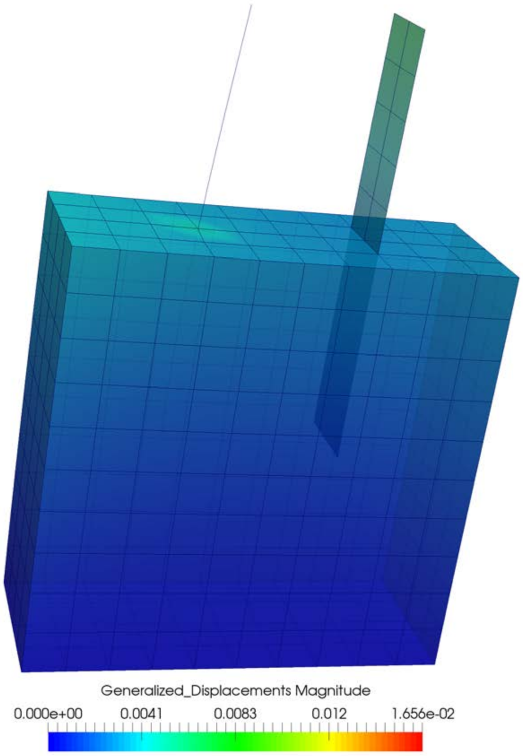


Figure 207.16: Visualizing displacement field in Paraview.

207.2.8 Realistic Models Developed Using gmESSI

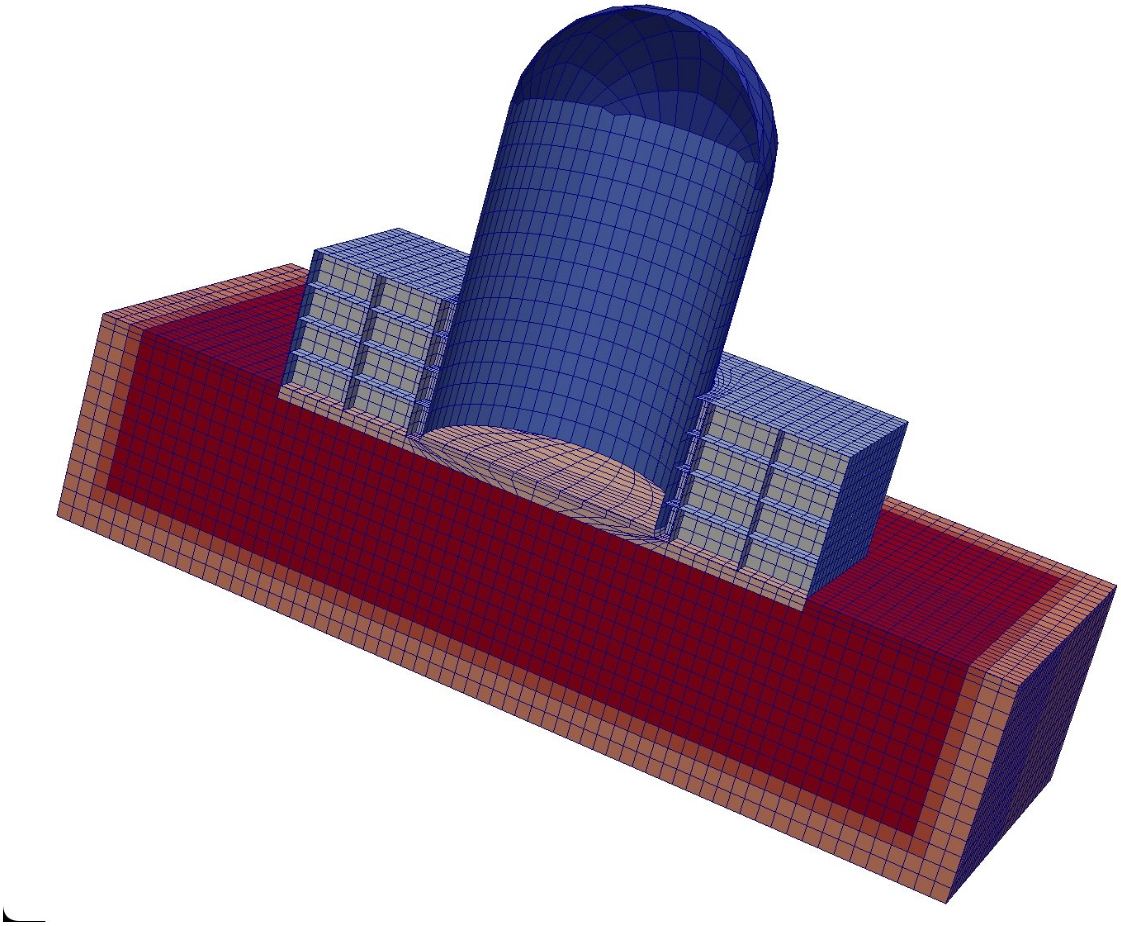


Figure 207.17: Nuclear Power Plant model 1, half model shown, with vertical plane cut.

207.3 Introduction to SASSI-ESSI Translator

This section will cover a simple mesh translator that translates mesh from SASSI format into ESSI format.

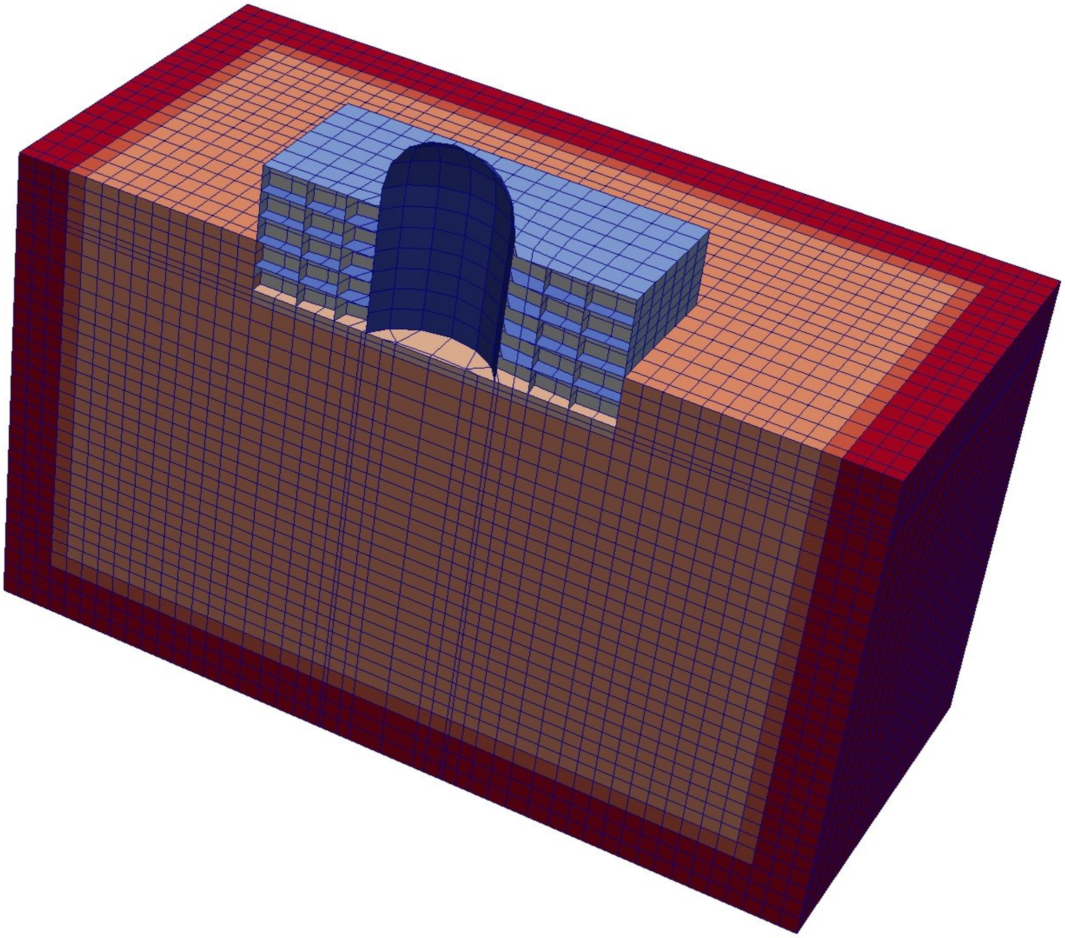


Figure 207.18: Nuclear Power Plant model 2, half model shown, with vertical plane cut.

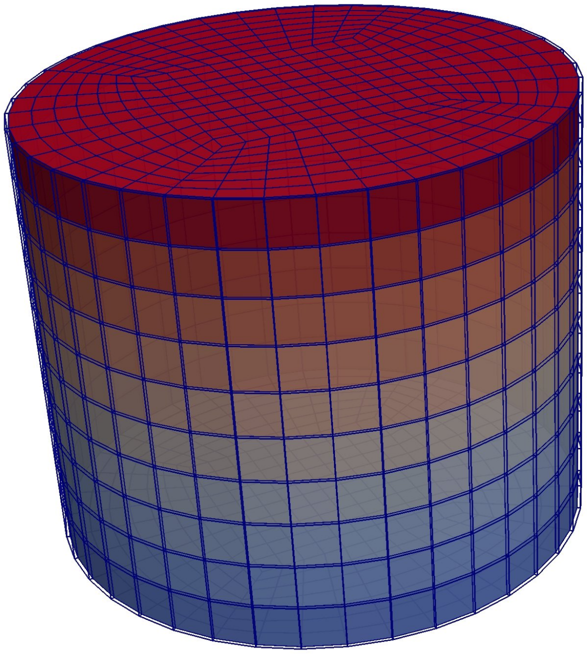


Figure 207.19: Shear Box.

Chapter 208

Real-ESSI Post Processing Methods

(2010-2014-2016-2017-2018-2019-2020-2021-)

(In collaboration with Prof. Sumeet Kumar Sinha, Dr. Yuan Feng, Prof. Han Yang, and Dr. Hexiang Wang)

208.1 Introduction

208.2 Model Results Post-Processing

This chapter describes methodology for post processing simulation results from the Real-ESSI Simulator. Two main approaches are used:

- Plotting time histories of scalar results (described in section 208.3 on page 1290), using Python and/or Matlab for:
  - components of displacements, velocities, accelerations, pore fluid pressures, for finite element nodes,
  - components of stress and/or strain at integration (Gauss) points within each finite element,
  - components of section forces for structural finite elements
  - energy input and dissipation for parts or whole of the volume/model, in incremental and/or cumulative form
- Visualization of a part or a complete model for displacements, velocities, accelerations, stress and strain components, sectional forces, energy dissipation through visualization system ParaView, as described in section 208.4 on page 1291.

208.3 Time Histories Plotting

Time histories of various scalar results (as listed above) can be extracted from output files, saved in HDF5 (Group, 2020), in a format described in chapter 206, on page 1180 in Jeremić et al. (1989-2025).

An excellent set of Python postprocessing tools was developed by Dr, Konstantinos Kanellopoulos, from ETH Zürich! You can find these tools at his github:
`https://github.com/ConstantinosKanellopoulos/Real-ESSI_postprocessing_tools`.

208.4 Post Processing and Visualization using ParaView

ParaView package http://www.paraview.org/ (Ayachit, 2015) is a very powerful multi-platform data analysis and visualization program available as an Open Source. Paraview can be run on supercomputers to analyze datasets of peta-scale size as well as on laptops for smaller data, and has become an integral tool in many national laboratories, universities and industry, and has won several awards related to high performance computation.

PVESSIReader is a plugin for paraview that integrates Real-ESSI Simulator output to Paraview for visualization. PVESSIReader reads Real-ESSI output file, in HDF5 format, files with extension .feioutput. The plugin works for sequential, parallel as well as remote visualization mode. It has a number of visualization features to visualize stresses, eigen modes, relative displacement, physical groups, energy dissipation, etc.

The installation of both Paraview and PVESSIReader is described in some detail in section 209.8.2 on page 1345 of the main document (Jeremić et al., 1989-2025).

208.4.1 Visualization in ParaView : Features

PVESSIReader has been consistently developed and added with lot of visualization options which are built on ParaView Visualization Toolkit (VTK) framework. This section shows all the visualization options that PVESSIReader offers other than what is available in ParaView. The features are illustrated in subsections below with help of examples in *Examples* folder of PVESSIReader source directory.

Before, looking at the features, it's important to know how PVESSIReader works. PVESSIReader takes Real-ESSI HDF5 output (.feioutput) file format as input and creates a *PVESSIReader* folder inside the HDF5 file. PVESSIReader does this to ensure the visualization to be optimized. The contents inside this folder are not important for any regular user. The contents of "PVESSIReader" folder is shown in Figure 208.1, although it is not important to regular users. User must know that, the plug-ins first creates this folder and then uses the content of this folder for visualization in ParaView. So creation/reading of this folder is the essence to visualization in ParaView.

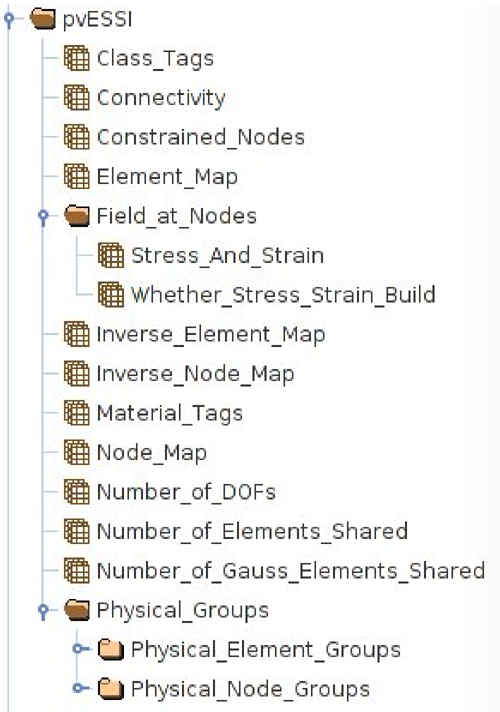
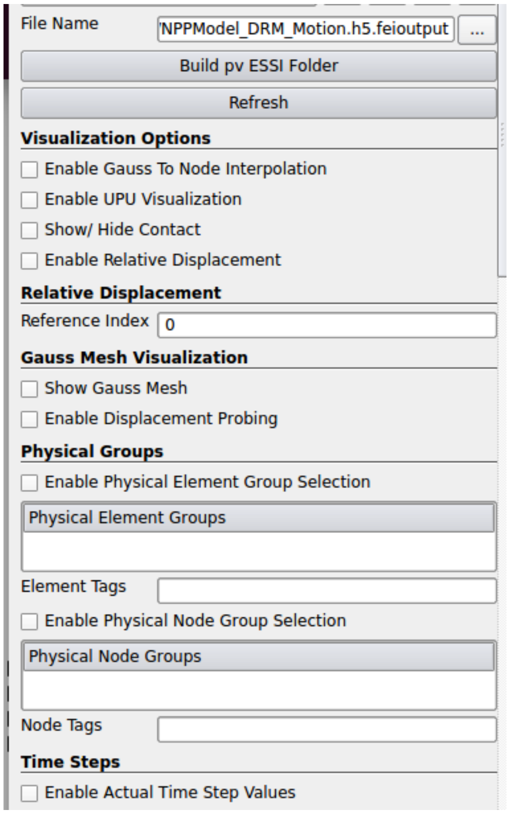


Figure 208.1: Contents of PVESSIReader folder.

208.4.1.1 PVESSIReader Visualization Options

By default PVESSIReader builds a node mesh. Figure 208.2 shows the various visualization options that is available for PVESSIReader. The Gauss to node interpolation is turned off and other options shown in Figure 208.2 is turned off. As stated in previous Section 208.4.1, the plugin creates the *PVESSIReader* *folder* inside the output HDF5 file only once and uses it for rest of visualization (even after you close and reopen it). The ‘Build PVESSIReader folder’ button shown in Figure 208.2 on clicking rebuilds the ‘PVESSIReader folder’. If in case the loading of ‘.feioutput’ file fails in ParaView, the user should clicks the ‘Build PVESSIReader Folder’ before hitting ‘Apply’ button.



File Name: NPPModel_DRM_Motion.h5.feioutput

Visualization Options

☐ Enable Gauss To Node Interpolation
☐ Enable UPU Visualization
☐ Show/ Hide Contact
☐ Enable Relative Displacement

Relative Displacement

Reference Index: 0

Gauss Mesh Visualization

☐ Show Gauss Mesh
☐ Enable Displacement Probing

Physical Groups

☐ Enable Physical Element Group Selection

Element Tags:

☐ Enable Physical Node Group Selection

Node Tags:

Time Steps

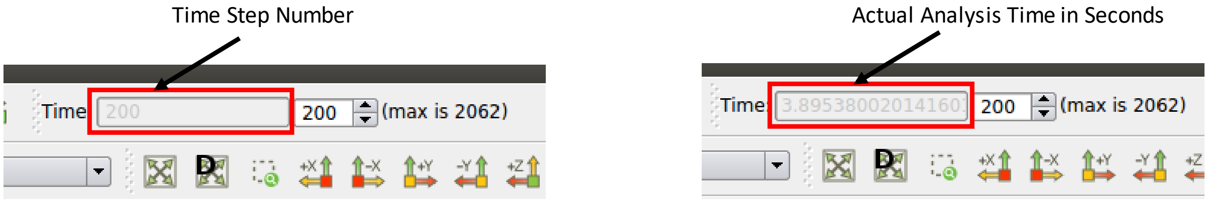
☐ Enable Actual Time Step Values

Figure 208.2: PVESSIReader build options.

Description of various PVESSIReader visualization options in the order shown in Figure 208.2 is listed below.

- Build PVESSIReader Folder - Rebuild the content of PVESSIReader folder inside output file
- Refresh - It reloads the current visualization view.
- Enable Gauss to Node Interpolation - Enables the interpolation from Gauss points to node using shape functions. It works only for 8 node and 27 node brick with 8 and 27 Gauss points respectively. See Section 208.4.1.9.
- Enable uPU Visualization - Enables the visualization of fluid displacements (U) and pore-pressure (P) at nodes. See Section 208.4.1.10.

- Enable Relative Displacement - When this is enabled, the displacement of any time step can be visualized with respect to any other reference time step. See Section 208.4.1.6.
- Reference Time Step No -This option is used to set the reference time step number about which the relative displacements would be visualized. See Section 208.4.1.6.
- Show Gauss Mesh - This option can be enabled to visualize the Gauss points (mesh) of the entire model. See Section 208.4.1.8.
- Enable Displacement Probing - This option only works if Gauss mesh option is enabled. With this option, displacements are calculated at Gauss points using ParaView interpolation functions for each elements containing those Gauss points. See Section 208.4.1.8.
- Physical Groups - This option is enabled to visualize pre-defined physical groups in Real-ESSI input or manually defined selected nodes or elements. See Section 208.4.1.12.
- Enable Actual Time Step Values - By default instead of actual simulation time (in seconds), time step number of analysis is provided to ParaView VCR. Figure 208.3 shows the result of enable/disable of this option.



(a) Actual Time Step Values Disabled

(b) Actual Time Step Values Enabled

Figure 208.3: Illustration of difference between enable and disable of `actual time step values`.

In the Figure 208.3, the enabled option gives the exact simulation time of $3.895s$. Whereas, disabling the option shows time step number of 200.

The visualization gets automatically updated on enable/disable of options. When one hit's apply, the corresponding changed result gets updated i.e. the mesh would get real time updated with enable/disable of these options.

208.4.1.2 Sequential Visualization

Sequential visualization means visualizing the Real-ESSI output on a single core of laptop/desktop. This is used for single output files that Real-ESSI produces for sequential runs. Figure 208.4 shows the visualization of output file produced by sequential Real-ESSI simulation.

```
cd pvESSI/Examples
ParaView ShearBox_Sequential.h5.feioutput
```

The parallel output files of ESSI can also be visualized sequentially. Each individual (core) file can be sequentially visualized showing only a part of the model results. Also all the parallel files at once can be opened as well in ParaView as shown in Figure 208.5. All PVESSIReader examples can be obtained at `http://sokocalo.engr.ucdavis.edu/~jeremic/lecture_notes_online_material/Real-ESSI_pvESSI/Examples`.

1. To open one core output in sequential

```
cd pvESSI/Examples
ParaView ShearBox_Parallel.h5.1.feioutput
```

2. To open all cores output in parallel

```
cd pvESSI/Examples
ParaView ShearBox_Parallel.h5.feioutput
```

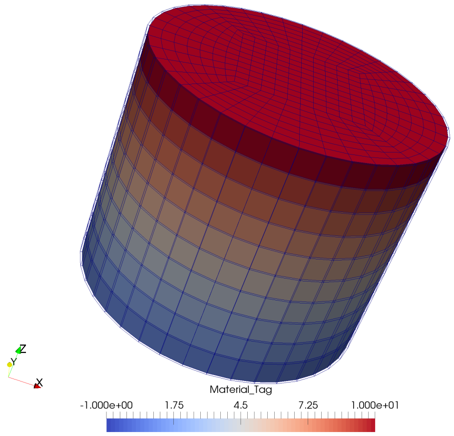
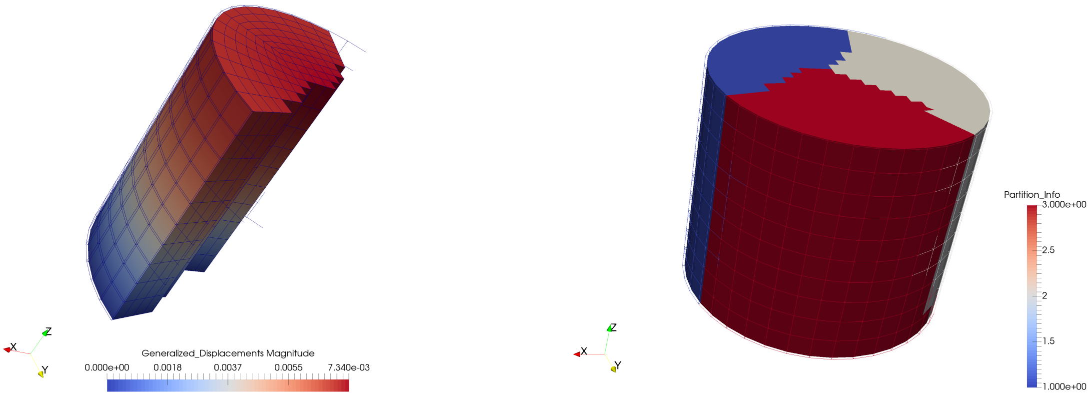


Figure 208.4: Sequential Visualization of output produced by sequential Real-ESSI simulation.



(a) Visualizing only one slave output file

(b) Visualizing all slave output files at once

Figure 208.5: Sequential Visualization of output produced by parallel Real-ESSI simulation.

208.4.1.3 Remote Visualization

Remote visualization is an important feature that ParaView offers. This is an important feature in need, when simulations are run on super computers with thousands of cores. The steps for remote visualization are shown below with an example on local desktop.

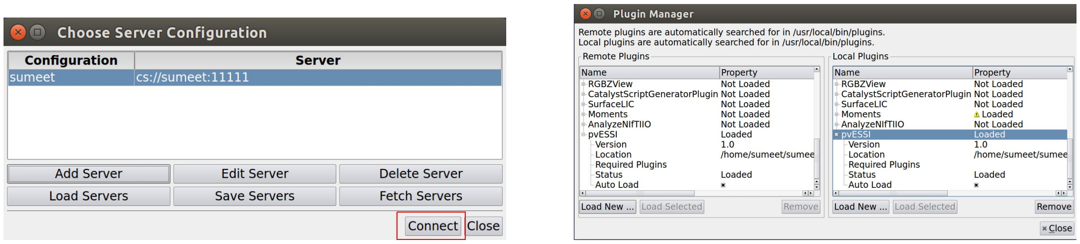
1. Run pvserver on server

```
$pvserver
Waiting for client...
%Connection URL: cs://sumeet:11111
%Accepting connection(s): sumeet:11111
Connection URL: cs://jeremic:11111
Accepting connection(s): jeremic:11111
```

2. Open ParaView on client side and click on connect button located on top left window.

Figure 208.6: Connect Server.

3. Select and connect to the server and then load the plugins on both client and server side as shown in Figure 208.7



Configuration	Server
sumeet	cs://sumeet:11111

(a) Connect to Server

Remote plugins are automatically searched for in /usr/local/bin/plugins.
Local plugins are automatically searched for in /usr/local/bin/plugins.

Remote Plugins

Name	Property
RGBZView	Not Loaded
CatalystScriptGeneratorPlugin	Not Loaded
SurfaceLIC	Not Loaded
Moments	Not Loaded
AnalyzeNIfTIIO	Not Loaded
pvESSI	Loaded
Version	1.0
Location	/home/sumeet/sumee
Required Plugins	
Status	Loaded
Auto Load	☒

Local Plugins

Name	Property
RGBZView	Not Loaded
CatalystScriptGeneratorPlugin	Not Loaded
SurfaceLIC	Not Loaded
Moments	⚠ Loaded
AnalyzeNIfTIIO	Not Loaded
pvESSI	Loaded
Version	1.0
Location	/home/sumeet/sumee
Required Plugins	
Status	Loaded
Auto Load	☒

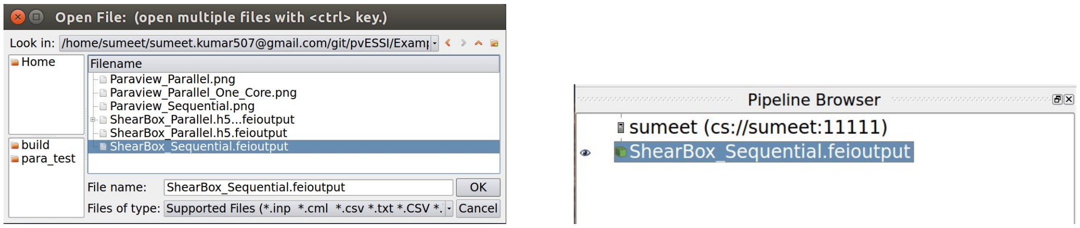
(b) All Core Outputs

Figure 208.7: Connect to the server and load plugins.

4. Navigate to *pvESSI/Examples/ShearBox_Parallel.h5.1.feioutput* and hit apply

208.4.1.4 Parallel Visualization

Parallel visualization is similar to remote visualization. The only difference is to start the pvserver in parallel on multiple cores. It is recommended to have the same number of cores that was used in Real-ESSI parallel simulation. Below shows steps on how to do parallel visualization in ParaView.



Look in: /home/sumeet/sumeet.kumar507@gmail.com/git/pvESSI/Examp

Paraview_Parallel.png
Paraview_Parallel_One_Core.png
Paraview_Sequential.png
ShearBox_Parallel.h5...feioutput
ShearBox_Parallel.h5.feioutput
ShearBox_Sequential.feioutput

File name: ShearBox_Sequential.feioutput

Files of type: Supported Files (*.inp *.cml *.csv *.txt *.CSV *.

sumeet (cs://sumeet:11111)
ShearBox_Sequential.feioutput

(a) Locating file on server side

(b) Client Side Pipeline

Figure 208.8: ParaView Remote Visualization.

```
$ mpirun -np $(nop) pvserver
# $(nop) is replaced by number of cores on which parallel visualization is to be run.
```

The next following steps are same to that of Remote visualization as shown in Section 208.4.1.3. Thus, parallel visualization can be performed remotely as well as locally.

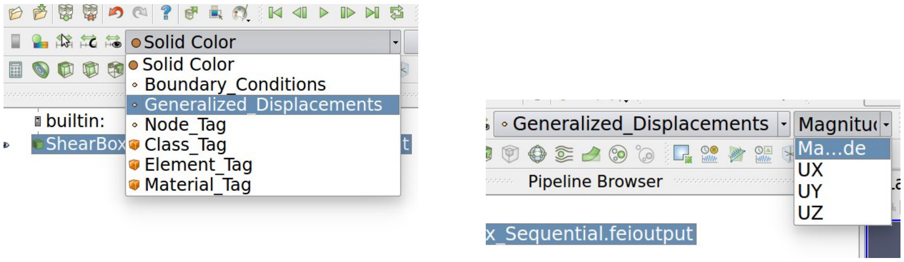
208.4.1.5 General Field Visualization

Below is the list of general visualization variables available for any model in ParaView using PVESSIReader plugin. The following subsections describes each option through an example. The examples can be found in pvESSI/Examples directory. The example file ‘`ShearBox_Sequential.h5.feioutput`’ can be downloaded here.

```
cd pvESSI/Examples
ParaView ShearBox_Sequential.h5.feioutput
```

Displacement Field : The displacement field represents the total displacement from the start of the Real-ESSI simulation. There are two modes of displacement field visualization available in ParaView. NOTE: Please remember to change step number from 0 to any other step number, as all output, including displacements, is 0.0 at step 0.

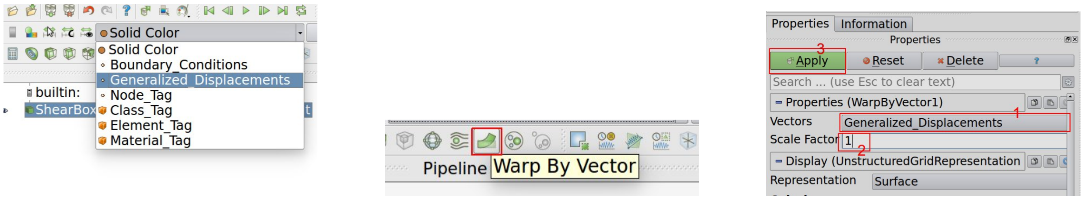
1. Scalar field visualization : The options available are each individual displacement vector components u_x, u_y and u_z in x,y,z directions respectively. It also shows the displacement magnitude $|u| = \sqrt{u_x^2 + u_y^2 + u_z^2}$. The units of displacements field in $[m]$.



(a) Select Displacement Field (b) Select scalar field

Figure 208.9: Displacement Scalar Field Visualization.

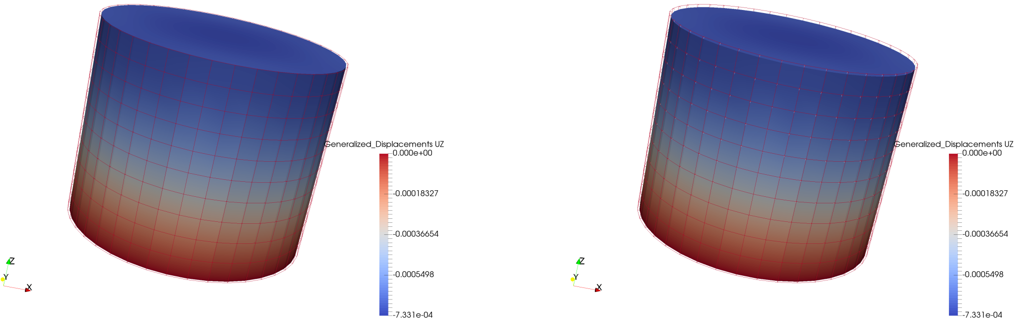
2. Vector Field visualization : This is achieved using ‘`Wrap by Vector`’ plugin available in ParaView. Figure 208.10 and Figure 208.11 shows steps to visualize deformed mesh.



Vectors Generalized_Displacements
Scale Factor 1
Representation Surface

(a) Select Displacement Field (b) Select plugin (c) Plugin properties

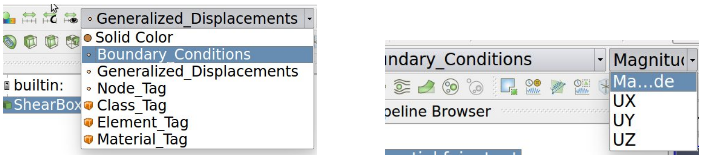
Figure 208.10: Deformation Visualization.



(a) Displacement Field Visualization in Uz (b) Deformation Visualization

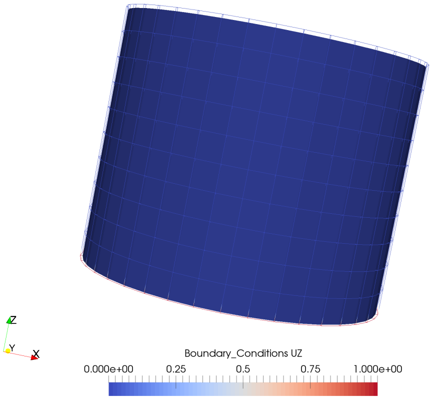
Figure 208.11: Displacement field visualization in ParaView.

Boundary Conditions: Again this a vector field which contains information about boundary conditions i.e fixities applied in u_x, u_y and u_z directions. A value of 1 means the node is fixed while 0 means it is free. Figure 208.12 shows steps to visualize boundary conditions.



(a) Select Boundary Conditions

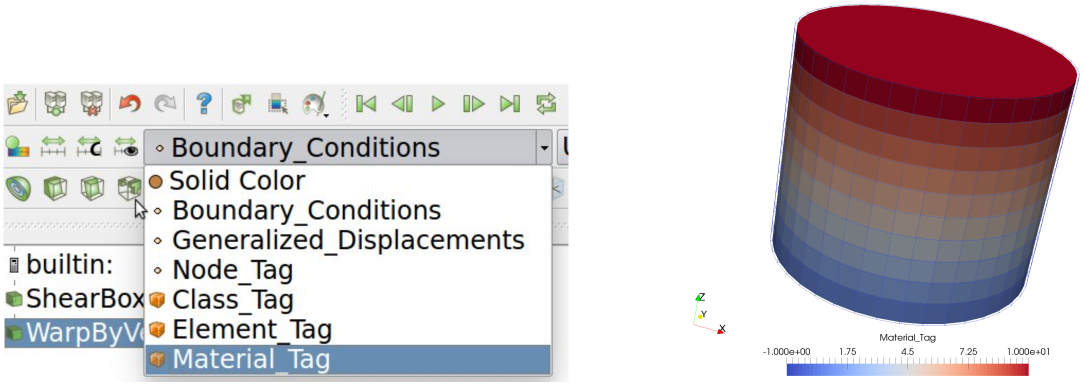
(b) Select Fixities Type



(c) Boundary Conditions in Uz

Figure 208.12: Boundary Conditions Visualization.

Material Tag Visualization : This is a scalar field visualization that shows the *material tag no* associated with the elements in Real-ESSI simulation. Figure 208.13 shows steps to visualize element's material tag.

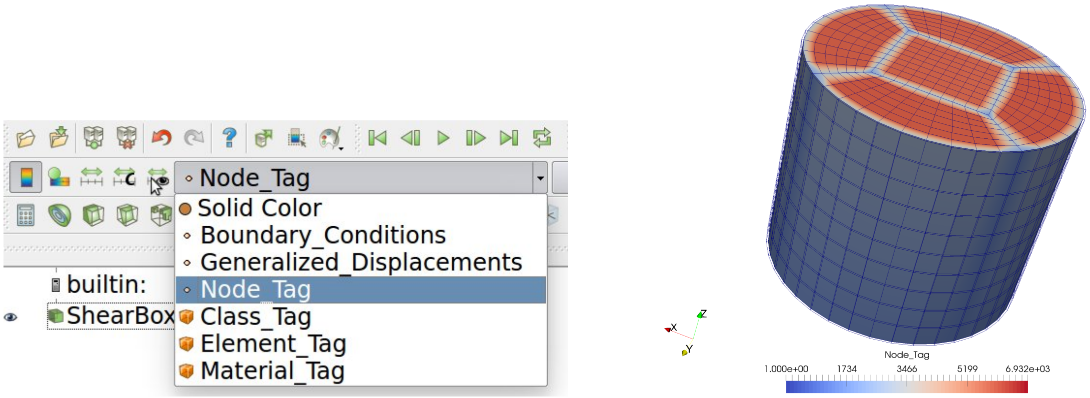


(a) Select Material Tag

(b) Material Tag Field

Figure 208.13: Material Tag Visualization.

Node Tag Visualization : This is a scalar field visualization that shows the `node no` associated with the nodes in Real-ESSI simulation. Figure 208.14 shows steps to visualize node's tag.

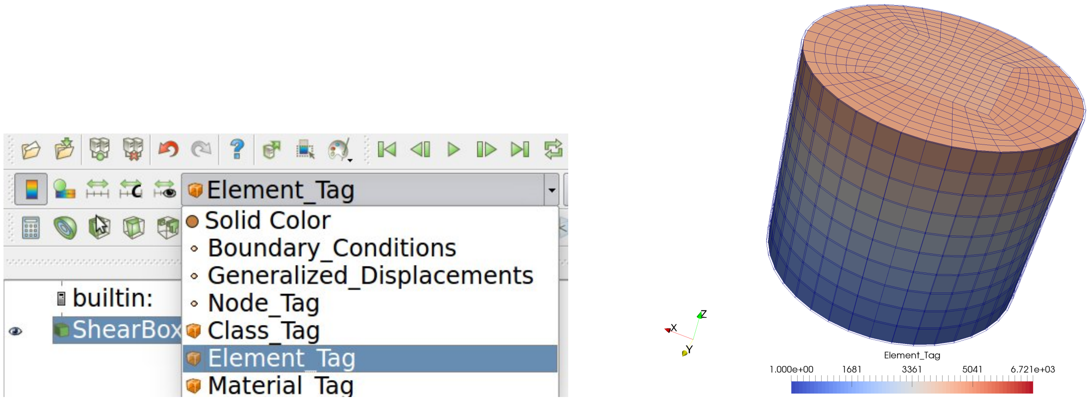


(a) Select Node Tag

(b) Node Tag Field

Figure 208.14: Node Tag Visualization.

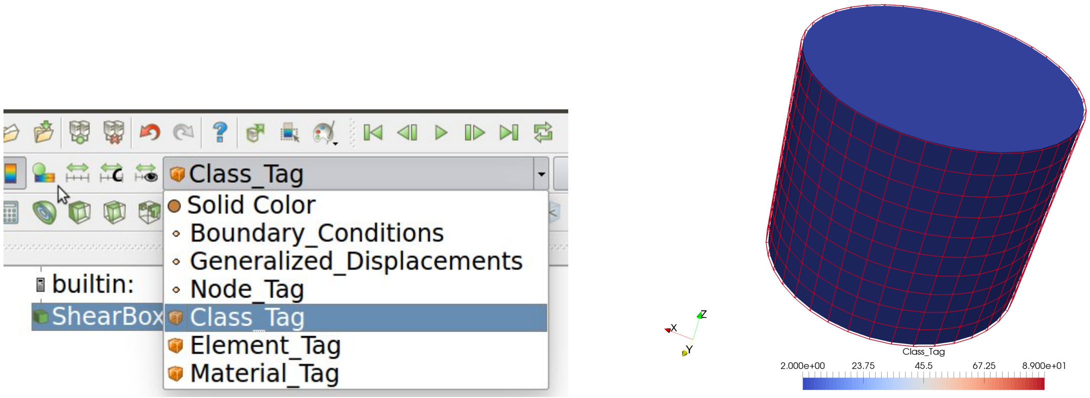
Element Tag Visualization : This is a scalar field visualization that shows the `element no` associated with the elements in Real-ESSI simulation. Figure 208.15 shows steps to visualize element's tag.



(a) Select Element Tag (b) Element Tag Field

Figure 208.15: Element Tag Visualization.

Element Class Tag Visualization : This is a scalar field visualization that shows the Element's Class Tag number associated with the each *element type* in Real-ESSI simulation. Figure 208.15 shows steps to visualize element's tag. Section 206.5.4.4 shows the class tag for various element types available in Real-ESSI.

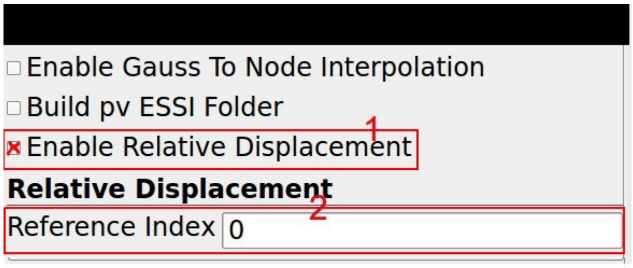


(a) Select Class Tag (b) Class Tag Field

Figure 208.16: Class Tag Visualization.

208.4.1.6 Relative Displacement Visualization

When the '`Enable Relative Displacement`' is checked, the relative displacement visualization option becomes active. BY default, the relative displacement time step number is set to '0' as shown in Figure 208.17. Time step number '0' corresponds to initial conditions of the loading stage output file.



☐ Enable Gauss To Node Interpolation

☐ Build pv ESSI Folder

☒ Enable Relative Displacement

Relative Displacement

Reference Index 0

Figure 208.17: Enable Relative Displacement.

Reference Time Step Number - It defines the relative time step index number for relative displacement visualization. By default it is set to 0 i.e. to the initial conditions. Its very useful, when one wants to visualize deformation coming from the stage itself. For example:- Separating self-weight from Static Pushover Analysis in the Shear Box simulation. The steps to do the same is shown below. The example file `ShearBox_PushOver.h5.feioutput` can be downloaded here.

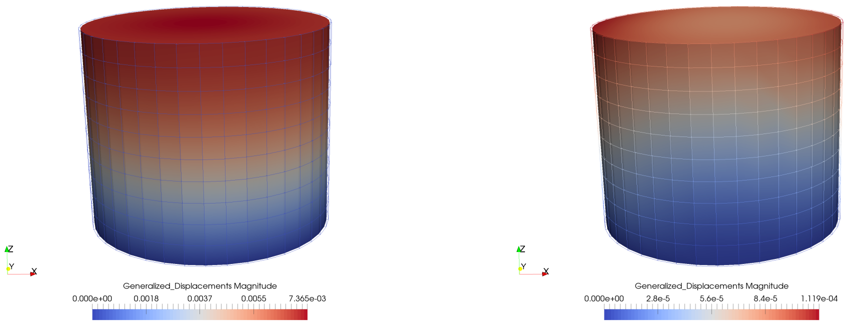
1. Open an example in ParaView

```
cd pvESSI/Examples
ParaView ShearBox_PushOver.h5.feioutput
```

2. Check on `Enable Relative Displacement` under visualization options

3. Apply `warp by vector` plugin and follow the steps as shown in Figure 208.10

Figure 208.18 shows the visualization with and without relative displacement.



(a) Without Relative Displacement

(b) With Relative Displacement

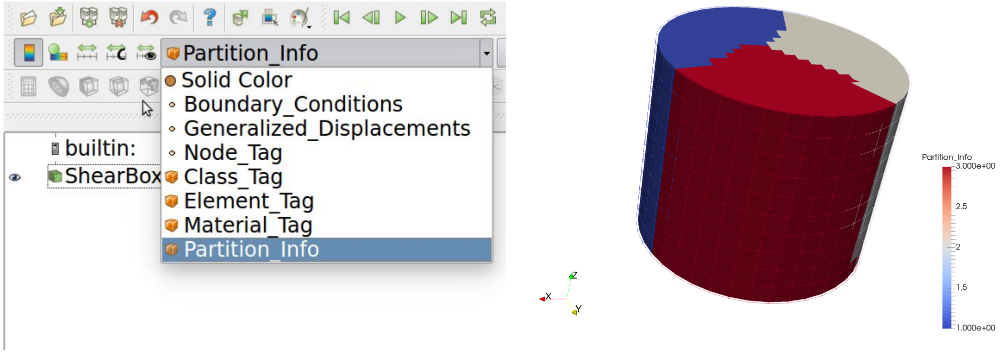
Figure 208.18: Pushover analysis of Shear Box after self-weight load application.

208.4.1.7 Visualizing Element's Partition

If the ESSI simulation was run in parallel mode, it becomes important to visualize the elements distribution between different cores. In ParaView, one can see the element distribution by selecting "`Partition Info`". Following is shown in Figure 208.19 an example to visualize mesh partitioning. All the example files can be obtained here.

```
cd pvESSI/Examples
ParaView ShearBox_Parallel.h5.feioutput
```

and then select Partition_Info as shown below in Figure 208.19



(a) Select Partition Info

(b) Mesh Partitioning

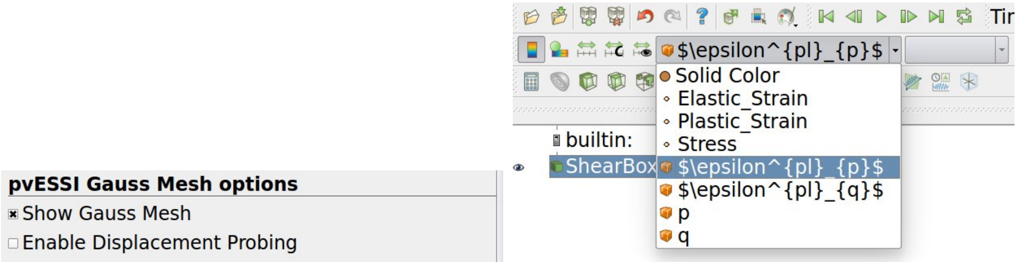
Figure 208.19: Visualizing mesh partitioning.

208.4.1.8 Gauss Mesh Visualization Options

Often, it is required to visualize stress and strain fields. Since stress or strains are evaluated at Gauss points in 3-D elements, Gauss mesh is needed to visualize them. PVESSIReader offers option to visualize Gauss mesh and it's fields.

- Show Gauss Mesh - Shows only Gauss mesh with Gauss attributes.
- Enable Displacement Probing - When this option is enabled, displacements are probed to the Gauss location. Its useful in the situation, when one wants to visualize the change in stress with deformation. With this as active, one can apply '`warp by vector`' filter.

It must be noted that the `Enable Displacement Probing` options only works when `Show Gauss Mesh` mode is enabled. Figure 208.20 shows the steps to visualize Gauss mesh.



pvESSI Gauss Mesh options

☒ Show Gauss Mesh

☐ Enable Displacement Probing

(a) Gauss Mesh Options (b) Gauss Mesh Fields

Figure 208.20: Visualizing Gauss mesh and it's fields.

The various fields that can be visualized in Gauss mesh mode as shown in Figure 208.20 are shown below.

- Total Strain ϵ : It defines the total strain from the start of the simulation. It has six independent component $\epsilon_{xx}, \epsilon_{xy}, \epsilon_{xz}, \epsilon_{yy}, \epsilon_{yz}$ and ϵ_{zz}. The magnitude of the total stress in ParaView is defined as $\sqrt{\epsilon_{ij} : \epsilon_{ij}}$.
- Total Plastic Strain ϵ^{pl} : It defines the total plastic strain from the start of the simulation. It has six independent component $\epsilon^{pl}_{xx}, \epsilon^{pl}_{xy}, \epsilon^{pl}_{xz}, \epsilon^{pl}_{yy}, \epsilon^{pl}_{yz}$ and ϵ^{pl}_{zz}. The magnitude of the total plastic strain in ParaView is defined as $\sqrt{\epsilon^{pl}_{ij} : \epsilon^{pl}_{ij}}$.
- Total Effective Stress $\sigma\prime$: It defines the total effective stress from the start of the simulation. It has six independent component $\sigma'_{xx}, \sigma'_{xy}, \sigma'_{xz}, \sigma'_{yy}, \sigma'_{yz}$ and σ'_{zz}. The magnitude of the total effective stress in ParaView is defined as $\sqrt{\sigma'_{ij} : \sigma'_{ij}}$. The unit of visualization is in $[Pa]$.

- Total Mean Effective Stress p : It defines the total mean of the effective stress σ' from the start of the simulation. It is defined as $p = -\sigma'_{ii}/3$ as described in Equation ??. The unit of Visualization is in $[Pa]$.

- Total Deviatoric Effective Stress q : It defines the deviatoric invariant of the total effective stress σ' from the start of the simulation. It is defined as $q = \sqrt{3J_2}$ as described in Equation ??. Where, J_2 is the second invariant of the deviatoric stress tensor $s_{ij} = \sigma'_{ij} - \sigma'_{kk}/3\delta_{ij}$. The unit of visualization is in $[Pa]$.

- Total Mean Plastic Strain ϵ_p^{pl} : It defines the mean total plastic strain ϵ^{pl} invariant from the start of the simulation. It is defined as $\epsilon_p^{pl} = -\epsilon_{ii}^{pl}/3$. This visualization parameter is unit-less.

- Total Deviatoric Plastic Strain ϵ_p^{pl} : It defines the deviatoric invariant of the total plastic strain ϵ^{pl} from the start of the simulation. It is defined as $\epsilon_p^{pl} = \sqrt{3J'_2}$. Where, J'_2 is the second invariant of the deviatoric plastic strain tensor $e_{ij}^{pl} = \epsilon_{ij}^{pl} - \epsilon_{kk}^{pl}/3\delta_{ij}$. This visualization parameter is unit-less.

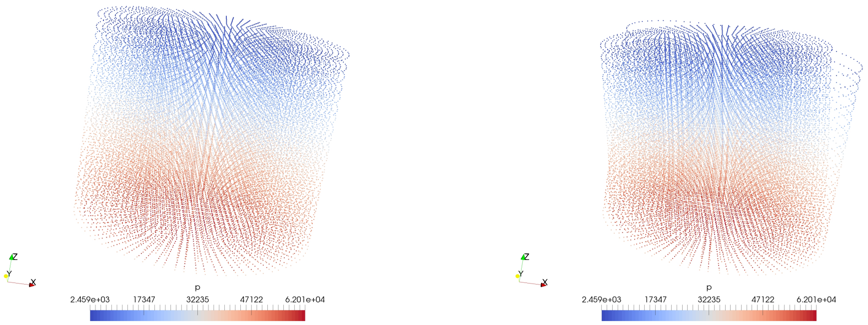
1. Open an example in ParaView. All the example files can be obtained at `http://sokocalo.engr.ucdavis.edu/~jeremic/lecture_notes_online_material/Real-ESSI_pvESSI/Examples`.

```
cd pvESSI/Examples
ParaView ShearBox_PushOver.h5.feioutput
```

2. Check on *Enable Relative Displacement* under PVESSIReader build options

3. Enable Gauss mesh as shown in Figure 208.20(a). Select `Mean Effective Stress` p `[Pa]`. The resulting visualization is shown in Figure 208.21(a).

4. Enable `displacement probing` as shown in Figure 208.20(a). Apply a `warp by vector` filter and select the vector `displacement` as shown in Figure 208.10. Now select again the `Mean Effective Stress` p `[Pa]` field option to visualize. The resulting visualization is shown in Figure 208.21(b).



(a) Gauss Mesh (b) Gauss Mesh with displacement probing

Figure 208.21: Visualization of mean effective stress p invariant in Gauss mesh.

208.4.1.9 Gauss To Node Interpolation Mode Visualization

This visualization mode can be enabled by checking the '`Gauss To Node Interpolation`' option as shown in Figure 208.22(a). In this mode, the total effective stress σ'_{ij}, total strain ϵ_{ij}, total plastic strain ϵ^{pl}_{ij}, total mean effective stress p, total deviatoric effective stress q, total mean plastic strain ϵ^{pl}_{p} and total deviatoric plastic strain ϵ^{pl}_{q} are interpolated from the Gauss points to the nodes of individual element. Individual shape functions of the element (with full Gauss integration) are used to obtain the stress or strain field at nodes. To smooth out the jumps in stress or strain field at the node by adjacent elements, unweighted averaging is performed. For the elements (usually structural) with no Gauss points, the stress or strain contribution at nodes are considered as zero. While taking the averaging, their contributions are not taken, as Real-ESSI does not output stress/strain for them.

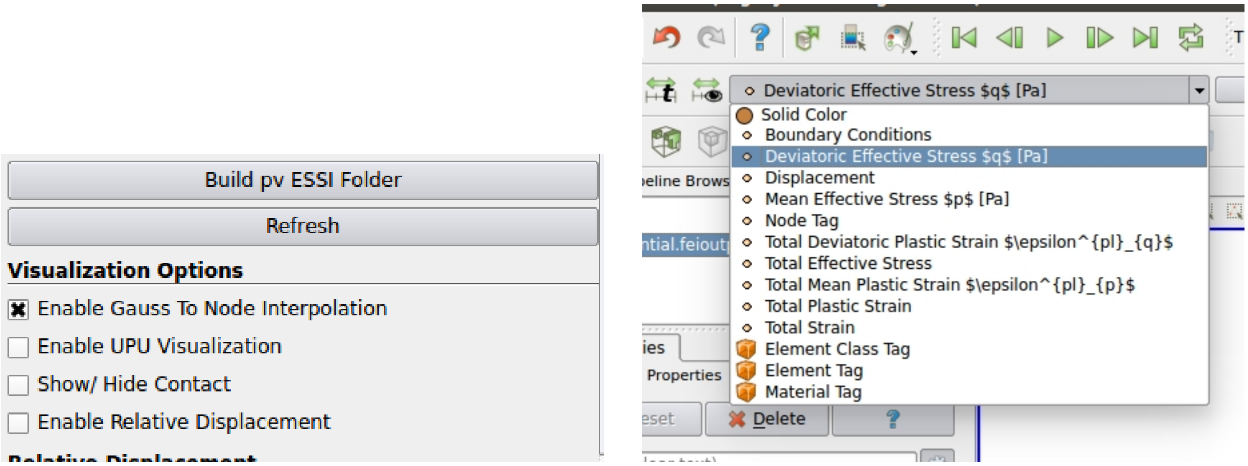
In this mode, visualization of all the parameters listed and described in Section 208.4.1.8 is available. Figure 208.22 show the steps to enable and use `Gauss to Node Interpolation` option.

1. Open an example in ParaView. All the example files can be obtained at `http://sokocalo.engr.ucdavis.edu/~jeremic/lecture_notes_online_material/Real-ESSI_pvESSI/Examples`.

```
cd pvESSI/Examples
ParaView ShearBox_Sequential.feioutput
```

2. Follow the steps as shown in Figure 208.22

Note : The option `Gauss to node interpolation` is provides only an approximate estimate for stress and strains at nodes. The values obtained at nodes is not accurate and thus `Gauss Mesh Visualization` option described in Section 208.4.1.8 must be performed to get the accurate stress and strains at Gauss points. Also, it must be noted that this option works only for 8 node brick with 8 Gauss points and 27 node brick with 27 node points. For elements which have less number of nodes that Gauss points, the total number of equations (unknowns) is not equal to constraints (knowns). In this case, only the shape function defined at the nodes are used to get the stress or strain back to the node.



Visualization Options

☒ Enable Gauss To Node Interpolation

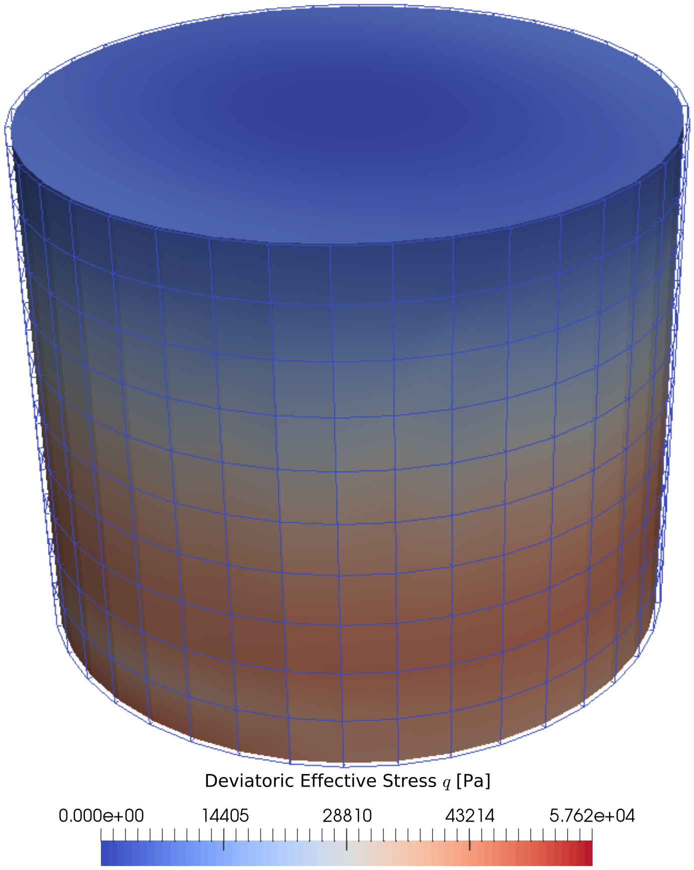
☐ Enable UPU Visualization

☐ Show/ Hide Contact

☐ Enable Relative Displacement

(a) Enable Gauss To Node Visualization Option

(b) Select the visualization variable



(c) Visualization of Deviatoric Stress interpolated to nodes

Figure 208.22: Steps to visualize stress and strain interpolated from Gauss points to nodes.

208.4.1.10 upU Visualization

This mode is to visualize the upU elements used in Real-ESSI simulation. Enabling this mode, produces additional outputs of '`Pore Pressure` $p[Pa]$' and '`Fluid Displacement` $U_x[m]$`,` $U_y[m]$ `and` $U_z[m]$' at nodes. These additional outputs are described below.

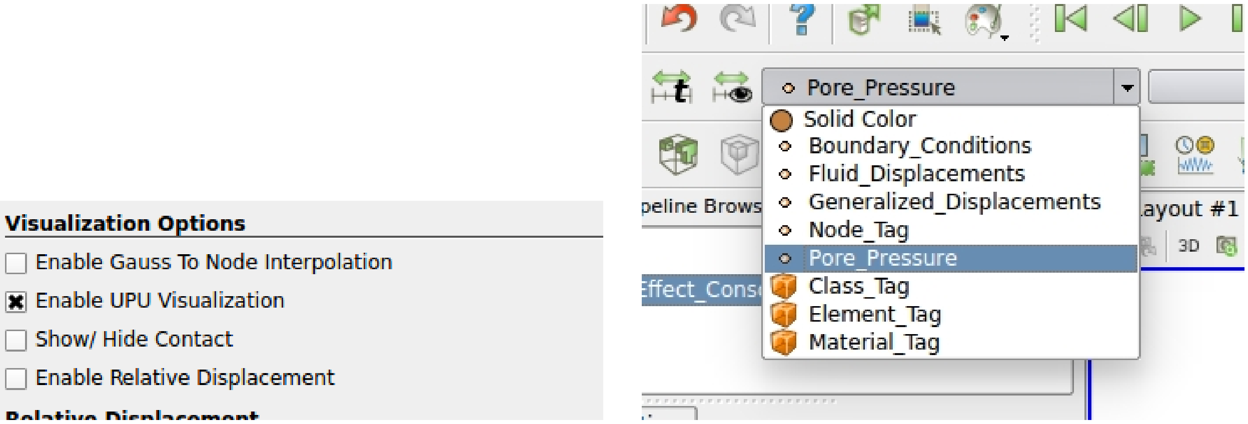
- Pore Pressure $p[Pa]$: It defines the pore-fluid pressure in the upU element at the nodes. The magnitude of the pore pressure is $[Pa]$.
- Fluid Displacement $U[m]$: It defines the displacement by the fluid particles of upU at nodes. The units is in meters $[m]$. The solid displacement is termed as u and refers to the '`Displacement` u' variable in visualization as described in Section 208.4.1.5.

Since general dry elements does not have any fluid, enabling this option would produce '`zero`' pore fluid pressure and fluid displacements at nodes. Below is shown an example that shows how to use the upU visualization feature. Figure 208.23 shows the steps.

1. Open an example in ParaView. All the example files can be obtained at `http://sokocalo.engr.ucdavis.edu/~jeremic/lecture_notes_online_material/Real-ESSI_pvESSI/Examples`.

```
cd pvESSI/Examples
ParaView upU_Visualization_Example.feioutput
```

2. Follow the steps as shown in Figure 208.23



Visualization Options

☐ Enable Gauss To Node Interpolation

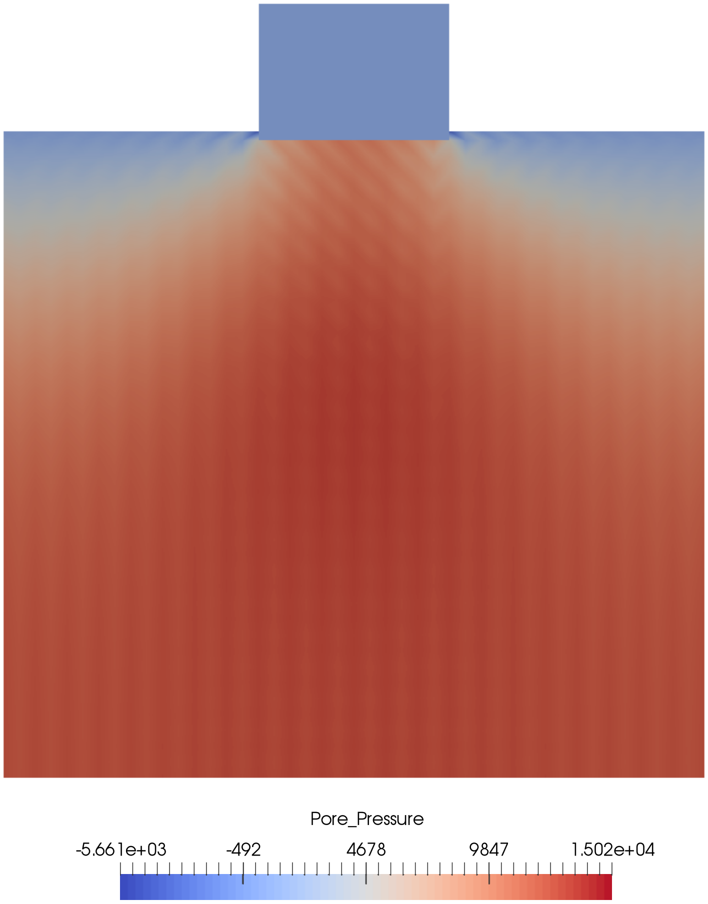
☒ Enable UPU Visualization

☐ Show/ Hide Contact

☐ Enable Relative Displacement

(a) Enable upU Mode Visualization Option

(b) Select the visualization variable either pore pressure or fluid displacement



(c) Visualization of Pore Pressure p at nodes

Figure 208.23: Steps to visualize pore pressure p or fluid displacements U in upU visualization mode.

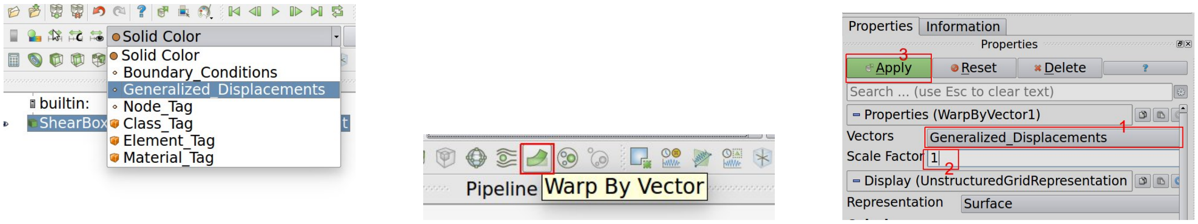
208.4.1.11 Eigen Mode Visualization

Visualization of eigen modes is that same as visualizing "displacements" and applying "warp by vector" filter on *Eigen Value Analysis* output of Real-ESSI simulation.

1. Open an *eigen value analysis* output. All the example files can be obtained at `http://sokocalo.engr.ucdavis.edu/~jeremic/lecture_notes_online_material/Real-ESSI_pvESSI/Examples`.

```
cd pvESSI/Examples
ParaView ShearBoxWall_Eigen_Analysis.h5.feioutput
```

2. Select displacement field and then apply warp by vector plugin and selected its properties



Vectors Generalized_Displacements

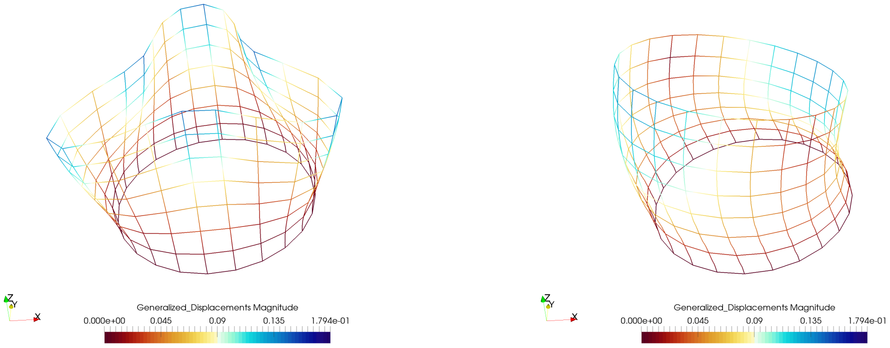
Scale Factor 1

Representation Surface

(a) Select Displacement Field (b) Select plugin (c) Plugin properties

Figure 208.24: Eigen Modes visualization.

3. Now n^{th} time steps here, corresponds to the n^{th} eigen mode.



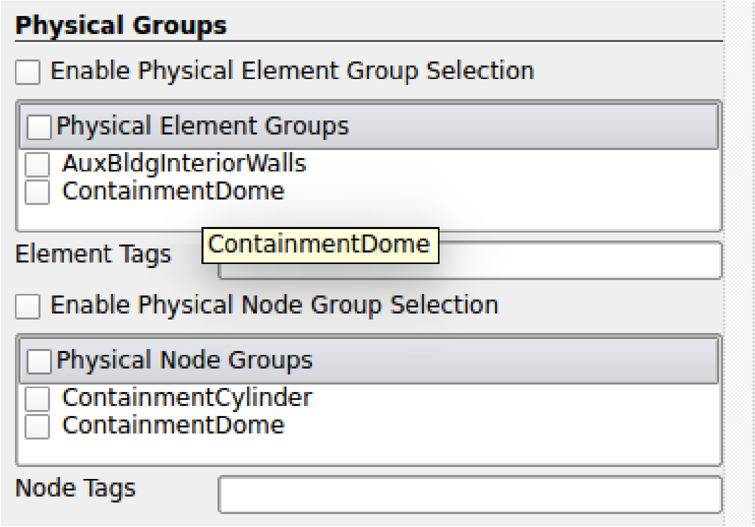
(a) Eigen Mode 5 (b) Eigen Mode 9

Figure 208.25: Few eigen modes.

208.4.1.12 Visualizing Physical Node and Element groups

In Real-ESSI it is possible to define different physical groups, for nodes and for elements. If one has defined physical groups in Real-ESSI, you can visualize the same in ParaView. There are two sections here that shows all the physical groups (nodes and elements) defined in the model as shown in Figure 208.26. Section 205.3.4.36 and Section 205.3.4.43 shows how to define and add physical group of nodes and elements respectively in Real-ESSI. All the example files can be obtained at `http://sokocalo.engr.ucdavis.edu/~jeremic/lecture_notes_online_material/Real-ESSI_pvESSI/Examples`.

```
cd pvESSI/Examples
ParaView Model_With_Physical_Groups.h5.feioutput
```



Physical Groups

☐ Enable Physical Element Group Selection

☐ Physical Element Groups
- ☐ AuxBldgInteriorWalls
- ☐ ContainmentDome

Element Tags: ContainmentDome

☐ Enable Physical Node Group Selection

☐ Physical Node Groups
- ☐ ContainmentCylinder
- ☐ ContainmentDome

Node Tags:

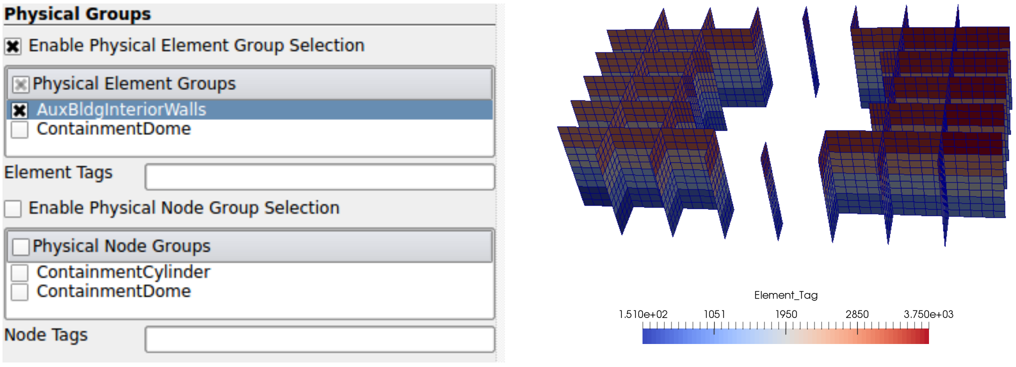
Figure 208.26: Physical Group Visualization Options.

This feature is very useful, when one is interested only in some specific regions of the model than the whole model. Also, This feature becomes very useful, for complicated "interested region/parts of the mesh", which cannot be selected by usual clip/box/..etc filters

1. Enable Physical Element Group Selection - Enables the selection of Physical Element Group. By default, it is disabled and one would see the whole mesh. By enabling it, one would only see the selected ‘`Physical Element Groups`’.

2. Physical Element Groups - It shows all the physical element groups defined in Input file of Real-ESSI. The user can select (one or more) of physical groups and hit apply to visualize them. It would show any effect only if the above options Enable Physical Element Group Selection is checked. Figure 208.27 shows steps to visualize the physical element groups defined in input files using

Real-ESSI DSL.



Physical Groups

☒ Enable Physical Element Group Selection

☒ Physical Element Groups

☒ AuxBldgInteriorWalls

☐ ContainmentDome

Element Tags

☐ Enable Physical Node Group Selection

☐ Physical Node Groups

☐ ContainmentCylinder

☐ ContainmentDome

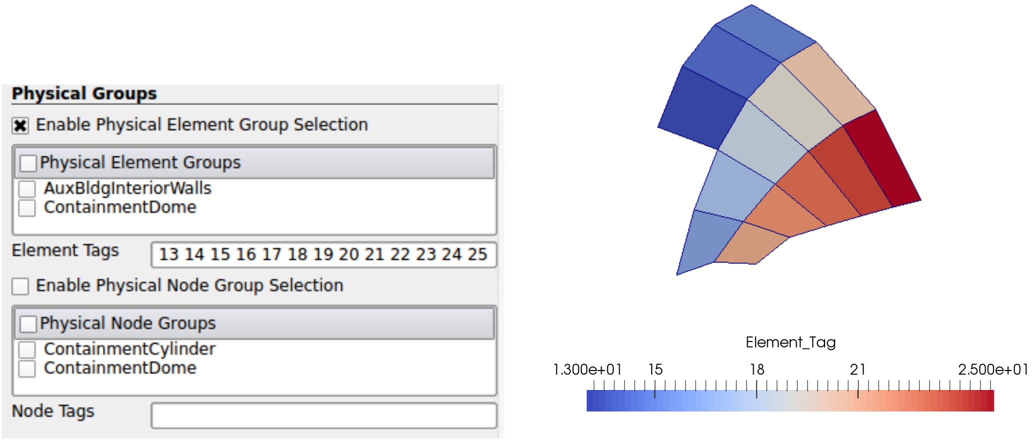
Node Tags

(a) Select `‘Physical Element Group Selection’` Option

(b) ElementGroup : Auxiliary Building Interior Walls

Figure 208.27: Visualization of physical groups predefined in input file using Real-ESSI DSL.

3. Element Tags - This options provides user and interface to manually write element tags to be visualized. The user should enter the element tags against this option as a integer list separated by space. For example:- ‘2 10 12 13 16’, where each of the number corresponds to the element tag defined in the model. Again, this option would only work if Enable Physical Element Group Selection option is checked. Figure 208.28 shows steps to visualize elements defined manually.



Physical Groups

☒ Enable Physical Element Group Selection

☐ Physical Element Groups

☐ AuxBldgInteriorWalls

☐ ContainmentDome

Element Tags: 13 14 15 16 17 18 19 20 21 22 23 24 25

☐ Enable Physical Node Group Selection

☐ Physical Node Groups

☐ ContainmentCylinder

☐ ContainmentDome

Node Tags

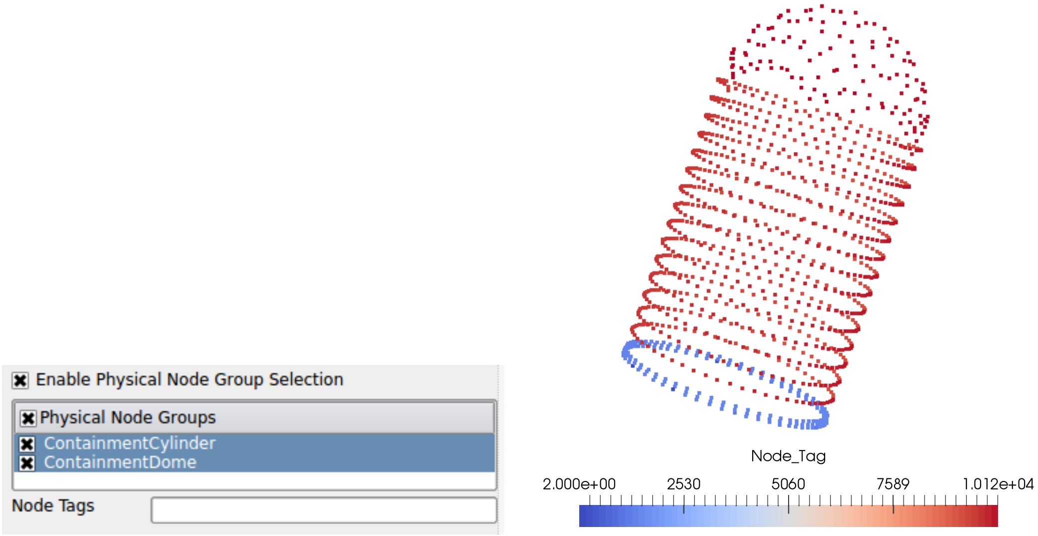
(a) Select `‘Physical Element Group Selection’` Option

(b) ElementGroup : Manually Defined Element Tags

Figure 208.28: Visualization of physical element group manually defined using PVESSIReader option.

4. Enable Physical Node Group Selection - Enables the selection of Physical Node Group. By default, it is disabled and you would see the whole mesh. By enabling it you would only see the selected Physical Node Groups.

5. Physical Node Groups - It shows all the physical node groups defined in Input file of Real-ESSI. The user can select (one or more) of physical groups and hit apply to visualize them.It would show any effect only if the above options Enable Physical Node Group Selection is checked. Figure 208.29 shows steps to visualize the physical element groups defined in input files using Real-ESSI DSL.



☒ Enable Physical Node Group Selection

☒ Physical Node Groups
- ☒ ContainmentCylinder
- ☒ ContainmentDome

Node Tags

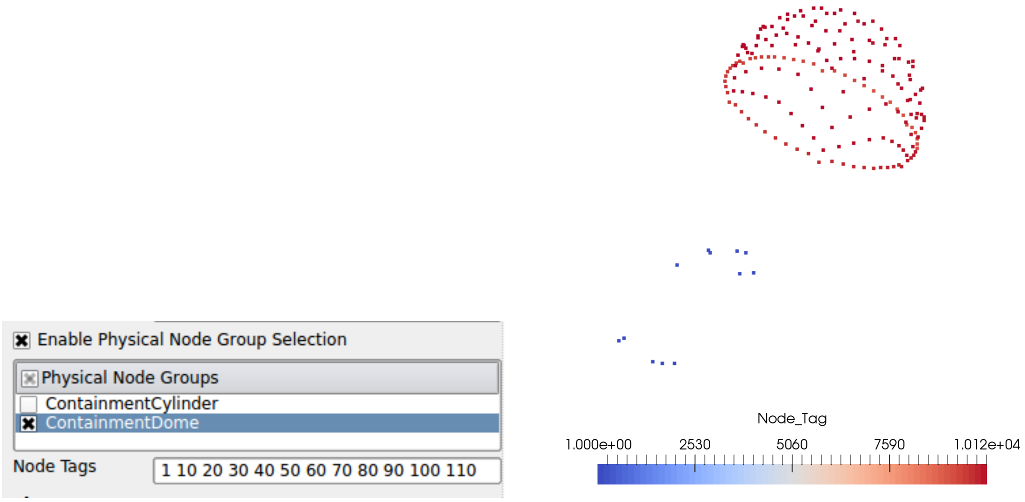
(a) Select `'Physical Node Group Selection'` Option

(b) NodeGroup : Auxiliary Building Interior Walls

Figure 208.29: Visualization of physical groups predefined in input file using Real-ESSI DSL.

6. Node Tags - This options provides user and interface to manually write node tags to be visualized. The user should enter the node tags against this option as a integer list separated by space. For example:- '`2 10 12 13 16`', where each of the number corresponds to the node tag defined in the model. Again, this option would only work if Enable Physical Node Group Selection option is checked. Figure 208.30 shows steps to visualize nodes defined manually.

NOTE: The user can also select both at once, i.e physical element group and physical node group, from the above menu. Figure 208.31 shows mixed selection.



☒ Enable Physical Node Group Selection

☒ Physical Node Groups

☐ ContainmentCylinder

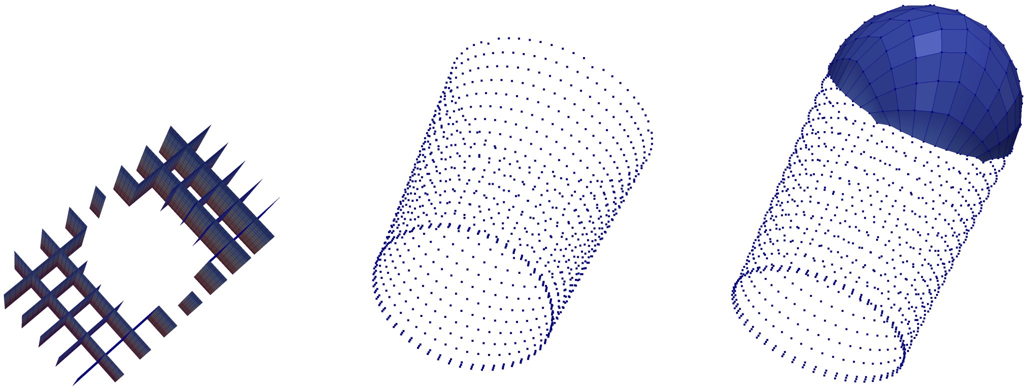
☒ ContainmentDome

Node Tags 1 10 20 30 40 50 60 70 80 90 100 110

(a) Select ‘`Physical Node Group Selection`’ Option

(b) NodeGroup : Manually Defined Node Tags

Figure 208.30: Visualization of physical node group manually defined using PVESSIReader option.



(a) ElementGroup : Auxiliary Building Interior Walls

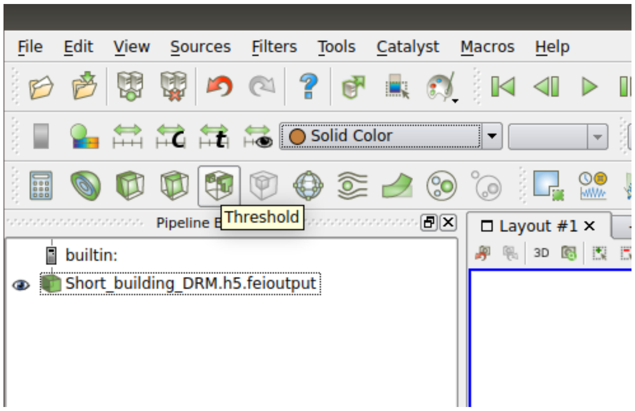
(b) Node Group : Containment Cylinder

(c) Mixed Group : Containment Cylinder with Dome

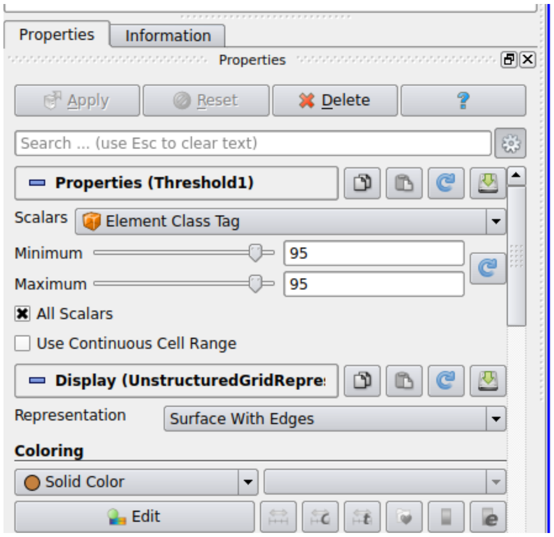
Figure 208.31: Visualization of physical groups.

208.4.1.13 Using Threshold to Visualize Certain Elements

ParaView allows user to choose specific element types and only visualize selected elements. This function is achieved using Threshold. As shown in Figure 208.32, first click on the Threshold button in toolbar. Then, choose Element Class Tag in the drop-down list of Scalars, which can be found in Properties. A certain range of Element Class Tag can be chosen by setting the minimum and maximum values. If the minimum and maximum values are the same, only one element type will be selected and visualized.



(a) Click on Threshold



Scalars: Element Class Tag

Minimum: 95

Maximum: 95

☒ All Scalars

☐ Use Continuous Cell Range

Representation: Surface With Edges

Coloring: Solid Color

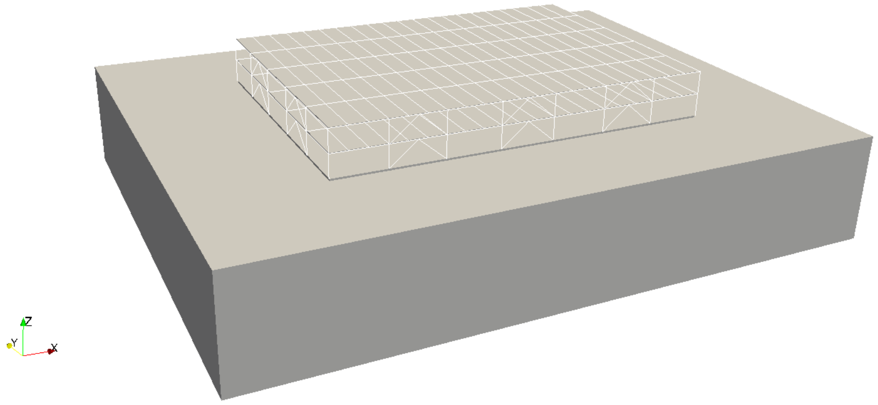
(b) Choose Element Class Tag

Figure 208.32: Using Threshold to Visualize Certain Elements.

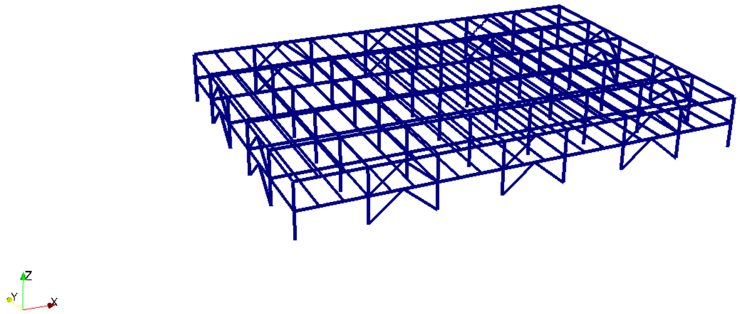
A list of available element class tags in Real-ESSI is provided in Table 208.1.

Table 208.1: Available element class tags in Real-ESSI.

Finite Element Name	Element Class Tag
Truss Element	88
Shear Beam Element	93
Elastic Beam–Column Element	89
Timoshenko Elastic Beam–Column Element	129
Elastic Beam-Column Element with Lumped Mass	90
3D Displacement Based Fiber Beam-Column Element	95
4 Node ANDES Shell with Drilling DOFs	92
3 Node ANDES Shell with Drilling DOFs	91
Super Element Linear Elastic Import	9904
8 Node Brick Element (Order One, Two, Three, Four, Five, Six)	2 (14, 26, 38, 50, 62, 74)
8 Node Brick u-p Element (Order One, Two, Three, Four, Five, Six)	3 (15, 27, 39, 51, 63, 75)
8 Node Brick u-p-U Element (Order One, Two, Three, Four, Five, Six)	4 (16, 28, 40, 52, 64, 76)
20 Node Brick Element (Order One, Two, Three, Four, Five, Six)	5 (17, 29, 41, 53, 65, 77)
20 Node Brick u-p Element (Order One, Two, Three, Four, Five, Six)	6 (18, 30, 42, 54, 66, 78)
20 Node Brick u-p-U Element (Order One, Two, Three, Four, Five, Six)	7 (19, 31, 43, 55, 67, 79)
27 Node Brick Element (Order One, Two, Three, Four, Five, Six)	8 (20, 32, 44, 56, 68, 80)
27 Node Brick u-p Element (Order One, Two, Three, Four, Five, Six)	9 (21, 33, 45, 57, 69, 81)
27 Node Brick u-p-U Element (Order One, Two, Three, Four, Five, Six)	10 (22, 34, 46, 58, 70, 82)
Variable Node Brick Element (Order One, Two, Three, Four, Five, Six)	11 (23, 35, 47, 59, 71, 83)
Variable Node Brick u-p Element (Order One, Two, Three, Four, Five, Six)	12 (24, 36, 48, 60, 72, 84)
Variable Node Brick u-p-U Element (Order One, Two, Three, Four, Five, Six)	13 (25, 37, 49, 61, 73, 85)
8 Node Cosserat Brick Element	96
Bonded Contact/Interface/Joint Element	102
Force Based Dry Hard Contact/Interface/Joint Element	86
Force Based Dry Soft Contact/Interface/Joint Element	87
Force Based Coupled Hard Contact/Interface/Joint Element	97
Force Based Coupled Soft Contact/Interface/Joint Element	98
Stress Based Dry Hard Contact/Interface/Joint Element with Elastic Perfectly Plastic Shear Behavior	99
Stress Based Dry Hard Contact/Interface/Joint Element with Nonlinear Hardening Shear Behavior	100
Stress Based Dry Hard Contact/Interface/Joint Element with Nonlinear Hardening and Softening Shear Behavior	101
Stress Based Dry Soft Contact/Interface/Joint Element with Elastic Perfectly Plastic Shear Behavior	107
Stress Based Dry Soft Contact/Interface/Joint Element with Nonlinear Hardening Shear Behavior	108
Stress Based Dry Soft Contact/Interface/Joint Element with Nonlinear Hardening and Softening Shear Behavior	109
Stress Based Coupled Hard Contact/Interface/Joint Element with Elastic Perfectly Plastic Shear Behavior	104
Stress Based Coupled Hard Contact/Interface/Joint Element with Nonlinear Hardening Shear Behavior	105
Stress Based Coupled Hard Contact/Interface/Joint Element with Nonlinear Hardening and Softening Shear Behavior	106
Stress Based Coupled Soft Contact/Interface/Joint Element with Elastic Perfectly Plastic Shear Behavior	110
Stress Based Coupled Soft Contact/Interface/Joint Element with Nonlinear Hardening Shear Behavior	111
Stress Based Coupled Soft Contact/Interface/Joint Element with Nonlinear Hardening and Softening Shear Behavior	112



(a) Full Model



(b) Only Fiber Beam-Column Elements

Figure 208.33: An example of using threshold to visualize fiber beam-column elements in a soil-structure model.

Chapter 209

Software Platform Build Process

(1993-1994-1996-1999-2003-2005-2007-2008-2009-2010-2011-2015-2017-2018-2019-2020-2021-)

(In collaboration with Prof. José Abell, Dr. Yuan Feng, Dr. Hexiang Wang, and Prof. Han Yang)

209.1 Chapter Summary and Highlights

This Chapter gives a brief description of build and installation procedures for the Real-ESSI Simulator. It is noted that current installation procedures rely on Real-ESSI Simulator program Debian package distribution, with pre and post processing modules that are installed separately. Both sequential and parallel version of the Real-ESSI Simulator program Debian packages are distributed, for Linux/Ubuntu operating system.

209.2 Introduction to the Real-ESSI Simulator Program

The Real-ESSI Simulator systems consists of the Real-ESSI Program, Real-ESSI Computer and Real-ESSI Notes. Alternative name for the Real-ESSI Simulator system is Real-ESSI Simulator system. The name Real-ESSI, is explained in section 201.2.6 on page 710.

209.3 Real-ESSI Simulator System Install

In adition to the Real-ESSI Program, Real-ESSI Simulator system consists of a pre-processing modules and post-processing modules. Installation of pre-processing modules is described in Chapter 207, on page 1221 of the main document, lecture notes (Jeremić et al., 1989-2025). Installation of post-processing modules is described in Chapter 208, on page 1287 of the main document, lecture notes (Jeremić et al., 1989-2025).

Both pre and post processing manuals are also available through the main Real-ESSI Simulator web site: `http://real-essi.info/`.

209.4 Build Procedures for the Real-ESSI Program and Modules

Note: This section describes build procedure for the Global Release 25.04 version of Real-ESSI. The very same procedures will apply to future version...

These build procedures are meant for users that have access to Real-ESSI Program source code. Required operating system is Ubuntu 24.04 LTS, unless otherwise specified. Building Real-ESSI on older versions of Ubuntu is no longer supported. Users that do not have source code can install Real-ESSI form Debian package, that is also available.

209.4.1 System Libraries Update/Upgrade

```
sudo apt update
sudo apt upgrade
sudo apt dist-upgrade
sudo apt autoremove
```

209.4.2 Install Build Dependencies

```
sudo apt install -y bison
sudo apt install -y build-essential
sudo apt install -y cmake
sudo apt install -y flex
sudo apt install -y git
sudo apt install -y mpich
sudo apt install -y ssh
sudo apt install -y valgrind
sudo apt install -y wget
sudo apt install -y zlib1g-dev
sudo apt install -y libboost-all-dev
sudo apt install -y libgmp3-dev
sudo apt install -y libhdf5-serial-dev
sudo apt install -y liblapack-dev
sudo apt install -y libmpfr-dev
sudo apt install -y libopenblas-dev
sudo apt install -y libopenmpi-dev
sudo apt install -y libpthread-workqueue-dev
sudo apt install -y libssl-dev
sudo apt install -y libtbb-dev
```

You may need to manually link the HDF5 libs to their proper names so that the compiler can find them. The HDF5 maybe in different versions. NOTE: what is needed is a latest version of libhdf5 for serial execution, the one "cpp" and the one without "cpp", so search for it by doing:

```
cd /usr/lib/x86_64-linux-gnu
dir `find . -name "*hdf5*serial*cpp*"`
```

and

```
dir `find . -name "*hdf5*serial*"` | grep -v cpp
```

Then the linking has to be for both versions, for example:

```
cd /usr/lib/x86_64-linux-gnu
sudo ln -s libhdf5_serial.so.103.3.0 libhdf5.so
sudo ln -s libhdf5_serial_cpp.so.103.3.0 libhdf5_cpp.so
```

If the libs `libhdf5.so` and `libhdf5_cpp.so` are already there, just move on.

209.4.3 Download Real-ESSI Source

It is important to note that Real-ESSI sources are not available for public download. This is so that we can control and guarantee Real-ESSI quality. Only developer collaborators are contributing sources, and those sources are quality checked and quality assured. In addition, there are a number of unique solutions, unique formulations, unique implementation details within Real-ESSI sources that are not available in any other research of commercial program. If you happen to obtain Real-ESSI sources from Prof. Jeremić, you can proceed with the installation procedure below.

Please make sure that you are in the main directory where your Real-ESSI global release is placed ((`GLOBAL_RELEASE`). Make a directory where all the sources will reside and go there:

```
cd
mkdir Real-ESSI
cd Real-ESSI
```

Obtain Real-ESSI sources from the github:

```
git clone git@github.com:BorisJeremic/Real-ESSI.git
```

Go to the Real-ESSI source directory:

```
cd Real-ESSI
```

Remember to 'git checkout' to the proper branch.

209.4.4 Download and Compile Real-ESSI Dependencies

Make directories for the dependencies:

```
mkdir -p ../RealESSI_Dependencies
mkdir -p ../RealESSI_Dependencies/include
mkdir -p ../RealESSI_Dependencies/lib
mkdir -p ../RealESSI_Dependencies/bin
mkdir -p ../RealESSI_Dependencies/SRC
```

Go to the directory, download and extract the sources of the dependencies:

```
cd ../RealESSI_Dependencies
wget http://sokocalo.engr.ucdavis.edu/~jeremic/RealESSI/Dependencies_SRC.tar.gz
tar -xzvf ./Dependencies_SRC.tar.gz -C ./SRC --strip-components 1
```

Go to the Real-ESSI directory and compile the dependencies:

```
cd ../Real-ESSI
./build_libraries suitesparse
./build_libraries arpack
```

```
./build_libraries lapack
./build_libraries parmetis
./build_libraries petsc_itself
```

209.4.5 Configure, Build, and Install the Real-ESSI Program

Configure and build the sequential version of Real-ESSI:

```
mkdir build
cd build
cmake ..
make -j 8
cd ..
```

Configure and build the parallel version of Real-ESSI:

```
mkdir pbuild
cd pbuild
cmake -DCMAKE_CXX_COMPILER=/usr/bin/mpic++ -DPROGRAMMING_MODE=PARALLEL ..
make -j 16
cd ..
```

Copy the Real-ESSI executables to system directory:

```
sudo cp build/essi /usr/local/bin/essi-sequential
sudo cp pbuild/essi /usr/local/bin/essi-parallel
```

209.4.6 Install Sublime Text and Real-ESSI Packages

Sublime Text (`https://www.sublimetext.com/`) is the recommended editor for Real-ESSI input files and pre-processing files. Install Sublime Text following the official installation steps, or using the following command:

```
wget -qO - https://download.sublimetext.com/sublimehq-pub.gpg | gpg --dearmor | ↩
   sudo tee /etc/apt/trusted.gpg.d/sublimehq-archive.gpg
echo "deb https://download.sublimetext.com/ apt/stable/" | sudo tee ↩
   /etc/apt/sources.list.d/sublime-text.list
sudo apt-get update
sudo apt-get install sublime-text
```

Open Sublime Text. Open the ‘Tools’ menu and select ‘Install Package Control...’. Open the ‘Preferences’ menu, select ‘Package Control’, then select ‘Package Control: Install Package’.

In the opened search bar, type the package name and click on the package to install it. Three packages should be installed:

FEI Syntax-n-Snippets, Real-ESSI syntax and auto completion plugin for [].fei files (input files for Real-ESSI program).

gmsh-Tools, syntax and autotext completion for Gmsh model development tools for Real-ESSI.

gmESSI-Tools, syntax and autotext completion for gmESSI model development tools for Real-ESSI.

209.4.7 Install HDFView

`HDFView` can be used to open Real-ESSI output files, which are in HDF5 format. Download the latest version of `HDFView` from `https://support.hdfgroup.org/ftp/HDF5/releases/HDF-JAVA/`. Click on the latest version, which is `hdfview-3.2.0` as of June 2022. Go to `bin/`, click `HDFView-3.2.0-ubuntu2004_64.tar.gz` and save the file in your `./Downloads/` directory. Then extract and install HDFView:

```
cd
tar -xvf ./Downloads/HDFView-3.2.0-ubuntu2004_64.tar.gz -C ./Downloads
sudo apt install -y ./Downloads/hdfview_3.2.0-1_amd64.deb
sudo ln -s /opt/hdfview/bin/HDFView /usr/local/bin/hdfview
```

Now you can use `HDFView` from a terminal. To be able to use `HDFView` when you click on a Real-ESSI output file, do the following additional steps. First open the file using the following command:

```
sudo gedit /usr/share/applications/hdfview-HDFView.desktop
```

Find the line:

```
Exec=/opt/hdfview/bin/HDFView
```

Replace it with:

```
Exec=/opt/hdfview/bin/HDFView %F
```

Save the file and close it.

Go to a Real-ESSI output file, which should have the suffix ‘h5.feioutput’. Right click on the file and select ‘Open with Other Application’. Click ‘View All Applications’ and choose `HDFView` from the list. Note that you only need to do this once. Next time when you click on a Real-ESSI output file, it will be opened automatically using `HDFView`.

209.4.8 Compile ParaView and PVESSIReader for Post-Processing

Install the build dependencies for ParaView:

```
sudo apt install -y libgl1-mesa-dev
sudo apt install -y libxt-dev
```

```
sudo apt install -y libqt5x11extras5-dev
sudo apt install -y libqt5help5
sudo apt install -y qttools5-dev
sudo apt install -y qtxmlpatterns5-dev-tools
sudo apt install -y libqt5svg5-dev
sudo apt install -y libtbb-dev
sudo apt install -y python3-dev
sudo apt install -y python3-numpy
sudo apt install -y ninja-build
```

Go to the directory where you want to install ParaView. Suggested location is the parent directory of the Real-ESSI source. If you are continuing from the previous subsection, do the following:

```
cd ..
```

Download the ParaView source from GitHub:

```
git clone --recursive https://gitlab.kitware.com/paraview/paraview.git
```

Make the build directory:

```
cd paraview
mkdir paraview_build
```

Modify the cmake file to include PVESSIReader plugin in the building process. Open the file `CMakeLists.txt` in the ParaView source directory. Find "set(paraview_default_plugins" and add "PVESSIReader" to the end of the list of plugins. Download the PVESSIReader source from GitHub:

```
cd Plugins
git clone git@github.com:BorisJeremic/Real-ESSI-pvESSI.git
mv Real-ESSI-pvESSI PVESSIReader
```

Go to the build directory and compile ParaView:

```
cmake -GNinja -DPARAVIEW_USE_PYTHON=ON -DPARAVIEW_USE_MPI=ON ↩
   -DVTK_SMP_IMPLEMENTATION_TYPE=TBB -DCMAKE_BUILD_TYPE=Release ..
ninja
```

Copy the ParaView executable to system directory:

```
sudo cp bin/paraview /usr/local/bin/paraview
```

Start `ParaView` and click 'Tools' → 'Manage Plugins...'. Click 'Load New...' and find the plugin named 'PVESSIReader.so' under directory `paraview/lib/paraview-5.10/plugins/PVESSIReader/`. Also check the box 'Auto Load' then close `ParaView`.

The procedures described in this subsection are based on the official build instruction of ParaView (https://gitlab.kitware.com/paraview/paraview/blob/master/Documentation/dev/build.md).

209.5 Build Real-ESSI Debian Package

Note: This section describes Debian package build procedure for the Global Release 22.07 and later versions of Real-ESSI. As noted before, the very same procedures will apply to future Ubuntu versions ...

Starting from the Global Release 22.07 version, the Real-ESSI Simulator system is distributed as a Debian package for Linux users. This section documents the build procedure of a Real-ESSI Debian package. Note that the steps described here are for building a "basic" or "quick" stand-alone Debian package containing the already-compiled Real-ESSI program, pre-processor gmsh/gmESSI and results viewer Paraview/pvESSI modules. This is different from building a Debian package containing the program sources. For more information see:

- https://wiki.debian.org/Packaging
- https://www.internalpointers.com/post/build-binary-deb-package-practical-guide
- https://ubuntuforums.org/showthread.php?t=910717

209.5.1 Build the Real-ESSI Program and Modules

Before starting to build the Debian package, you should have finalized building the Real-ESSI program and modules from source. To build Real-ESSI from source, please follow the build procedure described in section 209.4.

209.5.2 Build the Debian Package

209.5.2.1 Package Name

Standard Debian notation is all lowercase in the following format:

```
<project>_<major version>.<minor version>-<package revision>
```

The current Real-ESSI Debian package has the name:

```
real-essi_25.04_amd64
```

Note that the version names will change and be consistent with the version of Real-ESSI program, as decribed at http://real-essi.info/, so that users will have to change the above name to reflect the actual Real-ESSI version and the Debian package.

209.5.2.2 Create Directory

Create a directory to make your package in. The name should be the same as the package name.

```
mkdir real-essi_25.04_amd64
```

209.5.2.3 Create Internal Structure

Good idea is to put packaging directory in the root of Real-ESSI system. So go to where all of this happening, for example:

```
cd /home/jeremic/oofep/Rad_na_Sokocalu/GLOBAL_RELEASE/Real-ESSI
```

Make space for files of your program where they would be installed on a linux system.

```
mkdir real-essi_25.04_amd64/usr
mkdir real-essi_25.04_amd64/usr/local
mkdir real-essi_25.04_amd64/usr/local/bin
mkdir real-essi_25.04_amd64/usr/lib
mkdir real-essi_25.04_amd64/usr/lib/x86_64-linux-gnu
mkdir real-essi_25.04_amd64/opt
mkdir real-essi_25.04_amd64/opt/gmESSI
mkdir real-essi_25.04_amd64/opt/paraview
```

209.5.2.4 Copy Files

Copy the files to the packaging directory. Note that you should use your own directory paths....

For example:

```
cp bin/essi.sequential real-essi_25.04_amd64/usr/local/bin/essi.sequential
cp bin/essi.parallel real-essi_25.04_amd64/usr/local/bin/essi.parallel
```

209.5.2.5 Create the `control` File

Create a special metadata file that is used by the package manager to install program. The control file lives inside the `DEBIAN` directory. Mind the uppercase: a similar directory named `debian` (lowecase) is used to store source code for the so-called source packages. This tutorial is about binary packages,so we don't need source code. Create the empty `control` file:

```
mkdir real-essi_25.04_amd64/DEBIAN
touch real-essi_25.04_amd64/DEBIAN/control
```

Open the file previously created with text editor of your choice. The `control` file is just a list of data fields, as seen in listing below:

```
Package: real-essi
Version: 25.04
Architecture: amd64
Authors: Han Yang <hhhyang@ucdavis.edu>, Boris Jeremic <jeremic@ucdavis.edu>
Maintainer: Han Yang <hhhyang@ucdavis.edu>, Boris Jeremic <jeremic@ucdavis.edu>
Depends: libboost-all-dev, libhdf5-dev, libtbb-dev, libssl-dev, libopenmpi-dev, ↩
    mpich, libgl1-mesa-dev, libxt-dev, libqt5x11extras5-dev, libqt5help5, ↩
    qttools5-dev, qtxmlpatterns5-dev-tools, libqt5svg5-dev, libtbb-dev, ↩
    python3-dev, python3-scipy, python3-numpy, python3-matplotlib, python3-pip, ↩
    python3-pygments, liboctave-dev, python2.7-dev
Section: misc
Priority: optional
Provides: real-essi
Description: The Real-ESSI Simulator.
 The Real-ESSI Simulator (Realistic Modeling and Simulation
 of Earthquakes, and/or Soils, and/or Structures and their
 Interaction) is a software, hardware and documentation
 system for high performance, sequential or parallel, time
 domain, linear or nonlinear, elastic and inelastic,
 deterministic or probabilistic, finite element modeling and
 simulation of
  - statics and dynamics of soil,
  - statics and dynamics of rock,
  - statics and dynamics of structures,
  - statics of and dynamics of soil-structure systems,
  - dynamics of earthquakes, and
  - dynamic earthquake-soil-structure interaction.
Homepage: http://real-essi.info
```

209.5.2.6 Create the Post-Installation and Post-Remove Files

```
touch real-essi_25.04_amd64/DEBIAN/postinst
touch real-essi_25.04_amd64/DEBIAN/postrm
```

Both of these files have to have the following permissions::

```
chmod u+rwx real-essi_25.04_amd64/DEBIAN/postinst
chmod u+rwx real-essi_25.04_amd64/DEBIAN/postrm

chmod og=rx real-essi_25.04_amd64/DEBIAN/postinst
chmod og=rx real-essi_25.04_amd64/DEBIAN/postrm
```

The `postinst` file/script is executed after a successful installation of the Debian package. This script looks like this:

```
#!/bin/sh
# postinst script for real-essi

set -e

case "$1" in
    configure)
# not used
# ln -s /opt/gmESSI/build/bin/gmessy /usr/local/bin/
# ln -s /opt/paraview/bin/paraview /usr/local/bin/

update-alternatives --install /usr/bin/python python /usr/bin/python2.7 1
pip install h5py
    ;;

    abort-upgrade|abort-remove|abort-deconfigure)
    ;;

    *)
        echo "postinst called with unknown argument \`$1'" >&2
        exit 1
    ;;
esac

exit 0
```

The postrm file/script is executed after a successful removal of the Debian package. This script looks like this:

```
#!/bin/sh
# postrm script for real-essi

set -e

case "$1" in
    purge|remove|upgrade|failed-upgrade|abort-install|abort-upgrade|disappear)
# not used
# rm /usr/local/bin/gmessy
# rm /usr/local/bin/paraview
    ;;

    *)
        echo "postrm called with unknown argument \`$1'" >&2
        exit 1
    ;;
esac

exit 0
```

209.5.2.7 Build the Package

Build the package:

```
dpkg-deb --build --root-owner-group real-essi_25.04_amd64
```

The `--root-owner-group` flag makes all deb package content owned by the root user, which is the standard way to go. Without such flag, all files and folders would be owned by your user, which might not exist in the system the deb package would be installed to.

The command above will generate a nice `.deb` file alongside the working directory or print an error if something is wrong or missing inside the package. If the operation is successful you have created debian package ready for distribution.

209.6 Real-ESSI and OpenFOAM, Connecting

Note: This section describes build procedure for old versions of Real-ESSI and/or its modules.

OpenFOAM is a free, open source computational fluid dynamics (CFD) software developed primarily by OpenCFD Ltd since 2004 (https://www.openfoam.com/). Real-ESSI supports numerical interface with OpenFOAM and can perform solid/structure fluid interaction analysis through Real-ESSI – OpenFOAM connection.

209.6.1 Installation of Customized OpenFOAM

We have made in-house modifications and developments to the InterFOAM application (Deshpande et al., 2012) of OpenFOAM-v1612+ for solid fluid interaction. This section presents the installation of the Customized OpenFOAM:

Install the dependencies:

```
sudo apt-get update
sudo apt-get install build-essential
sudo apt-get install flex
sudo apt-get install bison
sudo apt-get install cmake
sudo apt-get install zlib1g-dev
sudo apt-get install libboost-system-dev
sudo apt-get install libboost-thread-dev
sudo apt-get install libopenmpi-dev
sudo apt-get install openmpi-bin
sudo apt-get install gnuplot
sudo apt-get install libreadline-dev
sudo apt-get install libncurses-dev
sudo apt-get install libxt-dev
```

```
sudo apt-get install qt4-dev-tools
sudo apt-get install libqt4-dev
sudo apt-get install libqt4-opengl-dev
sudo apt-get install freeglut3-dev
sudo apt-get install libqtwebkit-dev
sudo apt-get install libscotch-dev
sudo apt-get install libcgal-dev
```

Also, make sure gcc and cmake meet the following minimum version requirements:

- gcc: version 4.8.5 or above
- cmake: version 3.3 or above

Check the version of gcc and cmake by running the following commands on terminal. If you are installing on Ubuntu 16.04 and above, the system version of gcc and cmake should already meet the requirements.

```
gcc --version
cmake --version
```

Downloaded the source code of Customized OpenFOAM:

```
wget http://sokocalo.engr.ucdavis.edu/~jeremic/lecture_notes_online_material/
_Chapter_SoftwareHardware_Build_Process/OpenFOAM/sources/OpenFOAM.tar.gz
```

Choose a directory and extract the downloaded compressed file to the target directory.

```
tar -xzvf OpenFOAM.tar.gz -C /target/directory
```

For example, hereafter we choose $HOME as target directory. Replace $HOME with your chosen directory accordingly.

```
tar -xzvf OpenFOAM.tar.gz -C $HOME
```

Go to the extracted folder and source OpenFOAM environment configurations:

```
cd $HOME/OpenFOAM
source $HOME/OpenFOAM/OpenFOAM-v1612+/etc/bashrc
```

Setup CGAL and Boost version for compilation:

```
cgal_version=CGAL-4.9.1
boost_version=boost-system
```

Check the system readiness

```
foamSystemCheck
```

Change to the main OpenFOAM directory:

```
foam
```

Note: if running *foam* cannot change to the main OpenFOAM directory, in this case the directory is *$HOME/OpenFOAM*, source the environment configuration again by running the following terminal command.

```
source $HOME/OpenFOAM/OpenFOAM-v1612+/etc/bashrc
```

Compile OpenFOAM:

```
./Allwmake
```

Since OpenFOAM is shipped with ParaView for post-processing OpenFOAM field results using developed plug-in paraFoam (https://cfd.direct/openfoam/user-guide/v6-paraview/). We also need to compile customized ParaView with paraFoam plug-in:

```
cd $WM_THIRD_PARTY_DIR
./makeParaView
```

209.6.2 Check the Customized OpenFOAM Installation

Open a new terminal and source the OpenFOAM environment:

```
source $HOME/OpenFOAM/OpenFOAM-v1612+/etc/bashrc
```

Validate the build by running:

```
foamInstallationTest
```

Create a user run directory:

```
mkdir -p $FOAM_RUN
```

go to the user run directory:

```
run
```

Copy a simulation case from OpenFOAM tutorial to the user run directory:

```
cp -r $FOAM_TUTORIALS/incompressible/simpleFoam/pitzDaily ./
```

go to the copies case directory:

```
cd pitzDaily
```

Generate the mesh:

```
blockMesh
```

Perform the analysis with the application simpleFoam:

```
simpleFoam
```

Visualize the simulation results:

```
paraFoam
```

209.6.3 Compile Real-ESSI with Link to OpenFOAM

Go to Real-ESSI source directory under directory RealESSI_ROOT and clean any previous old compilation of Real-ESSI:

```
cd RealESSI_ROOT/Real-ESSI
rm -rf bin
rm -rf lib
rm -rf build_sequential
mkdir bin
mkdir lib
mkdir build_sequential
cd build_sequential
```

Build and install the executable, using 16 CPUs in this case. Of course, if you have more CPUs available, you can use most of them. Please make sure to specify your OpenFOAM installation directory with CMake argument -DOPENFOAM_DIR. For example, in this case, we specify the installation directory as *$HOME/OpenFOAM*.

```
time cmake -DUSE_OPENFOAM=TRUE -DOPENFOAM_DIR=$HOME/OpenFOAM ..
time make -j 16
make install
```

Rename essi to essi.sequential just so to distinguish it from the parallel executable:

```
cd ../bin
cp essi essi.sequential
```

Finally, install essi.sequential in system binary directory so that others can use it:

```
sudo rm /usr/bin/essi /usr/bin/essi.sequential
sudo cp essi.sequential /usr/bin/essi.sequential
sudo chmod a+x /usr/bin/essi.sequential
```

209.7 Code Verification After the Build Process

After build process, test cases to verify that installation is successful should be run. There are four groups of verification cases. The first two groups are designed for users. The last two groups are designed for developers.

1. The first group of test cases compares the sequential essi results to the analytic solutions.
2. The second group of test cases compares the parallel essi results to the analytic solutions.
3. The third group of test cases tests the version stability between two essi executables.
4. The fourth group of test cases tests the memory management of Real-ESSI with valgrind.

209.7.1 Run all verification test cases

In order to run all test cases to verify the installation, users can run

```
cd $RealESSI_PATH/
bash run_all_verification.sh
```

Please make sure that sequential essi is available as 'essi' in the PATH, and parallel essi is available as 'essi_parallel' in the PATH before running all the verification test cases.

In addition, if users want to clean the test results, users can run

```
cd $RealESSI_PATH/
bash clean_all_verification.sh
```

Finally, users can also run a single group of test cases as follows.

209.7.2 Test Sequential Real-ESSI

In order to test whether the installation of sequential essi is successful, open the sequential example folder and run the bash script.

```
cd $RealESSI_PATH/CompGeoMechUCD_Miscellaneous/examples/analytic_solution
bash make_comparison.sh
```

This bash script will run all the examples automatically and compare the results to the analytic solutions. The comparison results are not only printed in the Terminal but also saved as a .log file in the same folder. Before you run the examples, make sure *essi* is in your PATH.

209.7.3 Test Parallel Real-ESSI

In order to test whether the installation of parallel essi is successful, open the parallel example folder and run the bash script.

```
cd $RealESSI_PATH/CompGeoMechUCD_Miscellaneous/examples/parallel
bash make_comparison.sh
```

This bash script will run all the examples automatically and compare the results to the analytic solutions. The comparison results are not only printed in the Terminal but also saved as a .log file in the same folder. Before you run the examples, make sure *essi_parallel* is in your PATH.

209.7.4 Version Stability Test

Since new features are continuously updated and improved in Real-ESSI, the version stability test helps the developers to guarantee their modification will not affect the correct operation of other code.

In order to test version stability,

```
cd $RealESSI_PATH/CompGeoMechUCD_Miscellaneous/examples/version_stability
bash generate_original.sh
```

This bash script will run all the examples automatically and save the results for reference later. This bash script above should run with the previous stable essi.

Then, to test the new essi and compare the results

```
cd $RealESSI_PATH/CompGeoMechUCD_Miscellaneous/examples/version_stability
bash make_comparison.sh
```

This bash script will run all the examples again and compare the results to the previous saved results. This bash script should run with the new essi. The comparison results are not only printed in the Terminal but also saved as a .log file in the same folder.

209.7.5 Memory Management Test

Memory management is important in C/C++ programming. This group of test cases helps the developers to track the memory leak in Real-ESSI. For the details about the code stability verification, please refer to the Section 303.2 on Page 1459.

Before you run the test cases, make sure Valgrind is installed. You can install Valgrind by this command.

```
sudo apt-get install valgrind
```

You can also download the source of Valgrind and compile it from scratch.

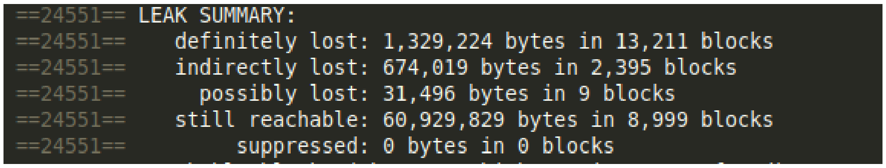
It is important to test memory leak in parallel simulations.

```
mpirun -np 3 valgrind --log-file='log_%p.valgrind' --leak-check=yes ↩
   essi-parallel-debug -f main.fei
```

A few important things to mention here:

- To test memory leak in parallel simulation, you obviously need a parallel version of Real-ESSI.
- Real-ESSI needs to be compiled in debug mode. This is important for Valgrind to capture and location the source of memory leaks.
- Running Real-ESSI in debug mode and in Valgrind means the simulation will be very slow. So it's not practical to run memory leak test using a large model. You should have a model with only a few elements/nodes (but more than 1 element so that it runs in parallel) that includes the specific functions you want to test.
- Valgrind log files will be saved in the location where you run the model. There will be multiple log files named as `log_processID.valgrind`. Each process will have its own Valgrind log file. There might be a few empty Valgrind log files generated, you can just ignore those. The number of Valgrind log files that actually contain memory leak information should be the same as the number of cores you use in your simulation.
- Valgrind is a powerful tool with many options. The command shown above is rather basic but serves as a good starting point. Memory leaks can be very tricky to track and fix. You should learn and experiment with Valgrind options for different issues you want to fix.

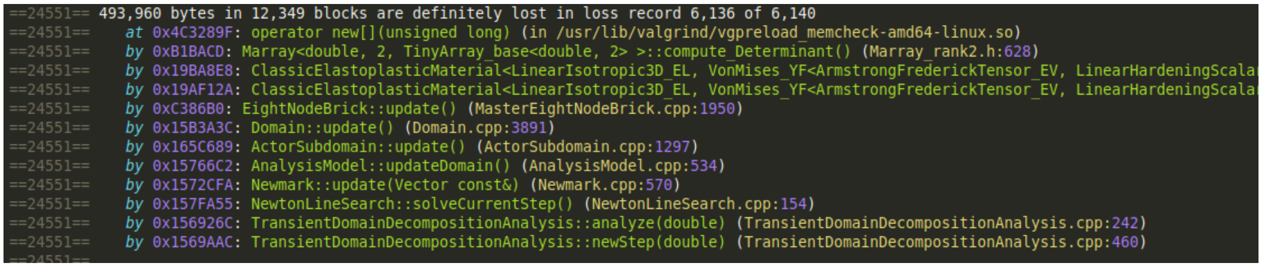
Valgrind log file can be very long and hard to read. At the bottom, there is a leak summary that looks like this: You should primarily focus on the 'definitely lost' result. 'Indirectly lost' and 'possibly lost' can also be problematic but should go away once you fix the source of 'definitely lost'. 'Still reachable' is usually not considered as actual memory leak but is something that can be optimized. Refer to the Valgrind User Manual for more information.



```
==24551== LEAK SUMMARY:
==24551==    definitely lost: 1,329,224 bytes in 13,211 blocks
==24551==    indirectly lost: 674,019 bytes in 2,395 blocks
==24551==      possibly lost: 31,496 bytes in 9 blocks
==24551==    still reachable: 60,929,829 bytes in 8,999 blocks
==24551==         suppressed: 0 bytes in 0 blocks
```

Figure 209.1: Valgrind log file: Memory leak summary.

Valgrind log file contains detailed information on each memory leak. A typical leak detail looks like this. Following the trail, you should be able to locate the source of a specific leak and then fix it properly.



```
==24551== 493,960 bytes in 12,349 blocks are definitely lost in loss record 6,136 of 6,140
==24551==    at 0x4C3289F: operator new[](unsigned long) (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==24551==    by 0xB1BACD: Marray<double, 2, TinyArray_base<double, 2> >::compute_Determinant() (Marray_rank2.h:628)
==24551==    by 0x19BA8E8: ClassicElastoplasticMaterial<LinearIsotropic3D_EL, VonMises_YF<ArmstrongFrederickTensor_EV, LinearHardeningScala
==24551==    by 0x19AF12A: ClassicElastoplasticMaterial<LinearIsotropic3D_EL, VonMises_YF<ArmstrongFrederickTensor_EV, LinearHardeningScala
==24551==    by 0xC386B0: EightNodeBrick::update() (MasterEightNodeBrick.cpp:1950)
==24551==    by 0x15B3A3C: Domain::update() (Domain.cpp:3891)
==24551==    by 0x165C689: ActorSubdomain::update() (ActorSubdomain.cpp:1297)
==24551==    by 0x15766C2: AnalysisModel::updateDomain() (AnalysisModel.cpp:534)
==24551==    by 0x1572CFA: Newmark::update(Vector const&) (Newmark.cpp:570)
==24551==    by 0x157FA55: NewtonLineSearch::solveCurrentStep() (NewtonLineSearch.cpp:154)
==24551==    by 0x156926C: TransientDomainDecompositionAnalysis::analyze(double) (TransientDomainDecompositionAnalysis.cpp:242)
==24551==    by 0x1569AAC: TransientDomainDecompositionAnalysis::newStep(double) (TransientDomainDecompositionAnalysis.cpp:460)
==24551==
```

Figure 209.2: Valgrind log file: Memory leak detail.

A serious memory leak issue caused by external solvers used by PETSc was found. As shown in Figure 209.3, when the `mumps` option was used in parallel solver, a significant amount of memory leak was detected by Valgrind. More importantly, such memory leak was observed to increase with the number of time steps. This means large-scale, long-duration simulation could be interrupted due to not enough memory in the operating system. Note that this issue was also reported in other occasions where the `mumps` package is used within PETSc, as recent as June 2020.

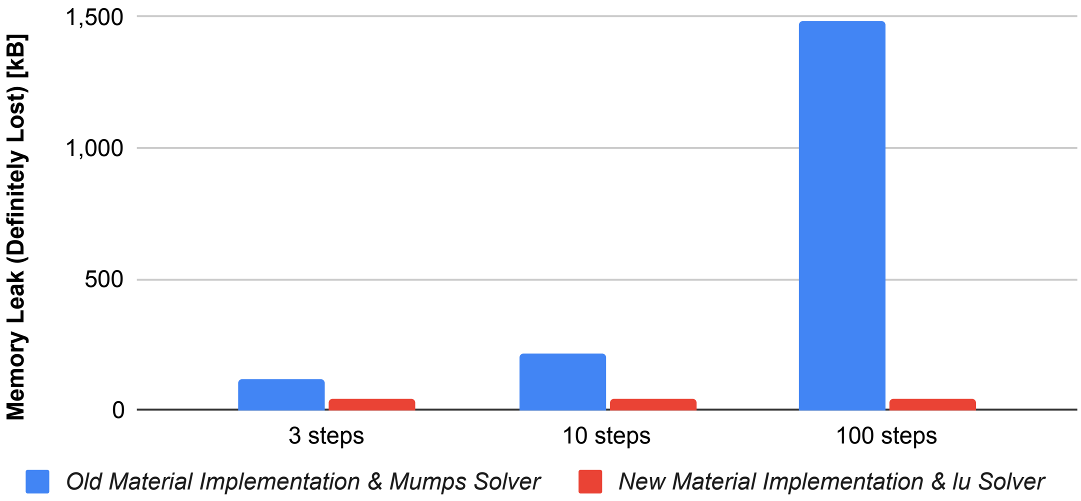


Figure 209.3: Comparison of memory leak between different PETSc solver options.

After extensive tests, it has been found out that other options/packages in PETSc don't have the memory leak issue mentioned above. Therefore it is recommended to use options other than `mumps` for large-scale, long-duration simulations. For example, the following command calls the default direct solver of PETSc:

```
define solver parallel petsc "-ksp_type preonly -pc_type lu" ;
```

209.8 Compiling Real-ESSI Utilities

Note: This section describes build procedure for old versions of Real-ESSI and/or its modules.

Real-ESSI comes with a lot of utilities to help the users speed up the simulation process. It provides mesh building, auto-input generation and visualization features which makes it quite nice.

Real-ESSI source code contains build_utilities script which can be used to build all the avialble utilities. We will go through the following subsection to introduce each utility and how to compile them.

The first step is to download all the sources of utilities that needs to be build. To do this, one has to run

```
cd Real-ESSI
./build_utilities download
```

This would download all the utilities sources in tar.gz format and would place them in "/SRC" of RealESSI_Utilities directory. The script is very powerfull and accepts targets that can be used to build a particular utility or all utilities at once. The available options to the scripts can be found by running the target help as shown below. A snippet is shown below

```
./build_utilities help

#usage: make [target]
#
#Utilities:
# gmessi Builds gmessi
# paraview Builds paraview
# pvessi Builds pvessi
# gmsh Builds gmsh
# visit Builds visit
# visitessi Builds visitessi
#
#Sequential:
# clean_utilities Cleans all utilities
#
#Default:
# all Builds all the necessary utilities for REAL-ESSI
# all_utils Builds all the necessary utilities for REAL-ESSI
# clean_all Cleans everything
# clean Cleans everything
#
#Check:
# check_utilities Checks if all utilities libraries are build
#
```

```
#Miscellaneous:
# list_utilities Lists all the available utilities version from SRC folder
# list_build_utilities Lists all the utilities library allready build in lib folder
# help Show this help.
# download Downloads the Utilities Sources
#
#Update:
# update_gmessi update gmessi utility
# update_pvessi update pvessi utility
# update_visitessi update visitessi utility
#
#Clean:
# clean_gmessi Clean gmessi utility
# clean_paraview Clean paraview utility
# clean_pvessi Clean pvessi utility
# clean_gmsh Cleans gmsh utility
# clean_visit Cleans visit
# clean_visitessi Cleans visitessi utility
```

The user can compile individual utilities by running just running

```
./build_utilities <utilitu_name>
```

Note: All the binaries of the utilities after build gets linked/copied to the RealESSI_Utilities/bin directory inside RealESSI_ROOT.

209.8.1 Installation of gmsh and gmESSI

gmsh is a 3-D finite element mesh generator for academic problems with parametric input and advanced visualization capabilities. It can be downloaded and installed from http://geuz.org/gmsh/. Additionally, the user can also install gmsh from terminal:

```
sudo apt-get install gmsh
```

gmESSI is effective pre-processor for generating Real-ESSI input files directly for the mesh file provided by gmsh. More information about gmESSI and how it works is given in Chapter 207 of the main document, lecture notes (Jeremić et al., 1989-2025). The gmESSI package is available from the main repository site: http://sokocalo.engr.ucdavis.edu/~jeremic/lecture_notes_online_material/_Chapter_SoftwareHardware_Pre_Processing/Real-ESSI_gmESSI.tgz. Before installing gmESSI, please install required libraries, as explained in Section ?? on Page ??. To install gmESSI, go to the Real-ESSI directory and then run

```
./build_utilities gmessi
```

To update the utility at the user just needs to run

```
./build_utilities update_gmessi
```

Refer to section 209.9 on page 1351 for instructions on what and how to install autocompletion and syntax coloring for gmESSI and Real-ESSI syntax on sublime text editor.

209.8.2 Installation of `ParaView` and `PVESSIReader`

ParaView package `http://www.paraview.org/` is a powerfull multi-platform data analysis and visualization application avialable in Open Source. It can be run on supercomputers to analyze datasets of petascale size as well as on laptops for small datasets. ParaView can be used to visualize results of Real-ESSI simulations. A plug-in was developed for ParaView so that all the simulations results from Real-ESSI finite elements, material models and analysis types can be directly visualized, animated, etc.

209.8.2.1 Building ParaView and PVESSIReader Plugin from Source on Linux System

Note that ParaView, as well as its building procedure, has recently (during 2020) gone through some major changes. The building procedures shown in this section are mostly based on the information available at: `https://gitlab.kitware.com/paraview/paraview/blob/master/Documentation/dev/build.md`.

1. Install Dependencies

```
sudo apt-get install libgl1-mesa-dev libxt-dev qt5-default ↩
   libqt5x11extras5-dev libqt5help5 qttools5-dev qtxmlpatterns5-dev-tools ↩
   libqt5svg5-dev python3-dev python3-numpy ninja-build
```

2. Obtain the source of ParaView

```
git clone --recursive https://gitlab.kitware.com/paraview/paraview.git
cd paraview
git checkout v5.8.1
git submodule update --init --recursive
mkdir paraview_build
```

3. Obtain the source of PVESSIReader plugin

   - The source of PVESSIReader plugin can be downloaded from `http://sokocalo.engr.ucdavis.edu/~jeremic/lecture_notes_online_material/_Chapter_SoftwareHardware_Post_Processing/_Real_ESSI_PVESSIReader_.zip`

  - Extract the files and move the `PVESSIReader` folder to `./Plugins/`

4. Modify the cmake file to include PVESSIReader plugin in the building process. Open the file `paraview/CMakeLists.txt` using your choice of text editor. Find "`set(paraview_default_plugins`" and add "`PVESSIReader`" to the end of the list of plugins.

5. Build

```
cd paraview_build
cmake -GNinja -DPARAVIEW_USE_PYTHON=ON -DPARAVIEW_USE_MPI=ON ↩
    -DVTK_SMP_IMPLEMENTATION_TYPE=TBB -DCMAKE_BUILD_TYPE=Release ..
ninja
```

6. Load PVESSIReader plugin into ParaView

  - Run the ParaView executable and click on Tools → Manage Plugins → Load New ...
  - Find `PVESSIReader.so` at `paraview/paraview_build/lib/paraview-5.9/plugins/PVESSIReader` and click OK to load it.
  - Now you should see PVESSIReader loaded in the list of plugins. Double click on it to expand advanced options and check Auto Load.
  - Close the ParaView application and reopen it. Now the PVESSIReader plugin should be automatically loaded and ready to use.

209.8.2.2 Building ParaView and PVESSIReader Plugin from Source on Windows System

Note that ParaView, as well as its building procedure, has recently (during 2020) gone through some major changes. The building procedures shown in this section are mostly based on the information available at: `https://gitlab.kitware.com/paraview/paraview/blob/master/Documentation/dev/build.md`. It is noted that user should be prepared to spend some time (perhaps hours) on installing procedure...

1. Install Dependencies

  - Download and install git bash for windows. Use the latest release version.
  - Download and install cmake. Use the lastest release version.
  - Download and install Visual Studio 2017 Community Edition. Please make sure that you tick the packages related to "Desktop Development with C++" and "Universal Windows Platform development".

- Download ninja-build and drop `ninja.exe` in `C:\Windows\`. Use the latest release version.
- Download and install both `msmpisetup.exe` and `msmpisdk.msi` from Microsoft MPI. Use the latest release version from Microsoft.
- Download and install Python for Windows. Latest release version should work fine. To avoid potential compatibility issues, install the same Python version that is used for the latest release of ParaView. (Currently, 01May2023, use Python 3.8.10 for Windows, as this version is the same as version used by Paraview 5.9.1.)
- Download and install Qt 5.12.3 for Windows, make sure to check the MSVC 2015 64-bit component during installation, make sure to add `C:\Qt\Qt5.12.3\5.12.3\msvc2017_64\bin` to your `PATH` environnement variable. Note that Qt for Windows is x86 but it works for x64 machine as well.

2. Obtain the source of ParaView, (Currently, 01May2023, version is Paraview 5.9.1.)

   - Open your preferred Windows command prompt. Windows PowerShell is a nice tool for people usually work with Linux system. Git Bash application also works nice.
   - To build ParaView developement version 5.9.1 (usually refered as "master"), run the following commands:

   ```
   cd C:
   mkdir pv
   cd pv
   git clone --recursive https://gitlab.kitware.com/paraview/paraview.git
   mv paraview pv
   mkdir pvb
   cd pv
   git checkout v5.9.1
   git submodule update --init --recursive
   ```

   - To build a specific ParaView version, please refer to `https://gitlab.kitware.com/paraview/paraview/blob/master/Documentation/dev/build.md`.

3. Obtain the source of PVESSIReader plugin

   - The source of PVESSIReader plugin can be downloaded from `http://sokocalo.engr.ucdavis.edu/~jeremic/lecture_notes_online_material/_Chapter_SoftwareHardware_Post_Processing/_Real_ESSI_PVESSIReader_.zip`
   - Extract the files and move the `PVESSIReader` folder to `C:\pv\Plugins\`

4. Modify the cmake file to include PVESSIReader plugin in the building process. Open the file `C:\pv\CMakeLists.txt` using your choice of text editor. Find "`set(paraview_default_plugins`" and add "`PVESSIReader`" to the end of the list of plugins.

5. Build

   - Open VS2017 x64 Native Tools Command Prompt and run the following commands

   ```
   cd C:\pv\pvb
   cmake -GNinja -DPARAVIEW_USE_PYTHON=ON -DPARAVIEW_USE_MPI=ON ↩
       -DCMAKE_BUILD_TYPE=Release ..\pv
   ninja
   ```

   - This step could be take a few hours. If no configuration or compilation error is encountered, you should have the ParaView executable at `C:\pv\pvb\bin\`.
   - Download and install Python 3.9.11 for Windows, as needed to run ParaView executable.

6. Load PVESSIReader plugin into ParaView

   - Run the ParaView executable and click on Tools → Manage Plugins → Load New ...
   - Find `PVESSIReader.dll` at `C:\pv\pvb\bin\paraview-5.9\plugins\PVESSIReader\` and click OK to load it.
   - Now you should see PVESSIReader loaded in the list of plugins. Double click on it to expand advanced options and check Auto Load.
   - Close the ParaView application and reopen it. Now the PVESSIReader plugin should be automatically loaded and ready to use.

209.8.2.3 Building ParaView and PVESSIReader Plugin from Source on AWS

Currently, the AWS image has Ubuntu 18.04. This may change in the future. Because AWS is a remote server, properly running ParaView needs more steps in compilation. Note that most information here are based on this discussion.

1. Install Dependencies

```
sudo apt-get install libgl1-mesa-dev libxt-dev qt5-default ↩
    libqt5x11extras5-dev libqt5help5 qttools5-dev qtxmlpatterns5-dev-tools ↩
    libqt5svg5-dev python3-dev python3-numpy ninja-build gettext python-mako
```

2. Download, Build and Install LLVM

```
wget http://releases.llvm.org/7.0.1/llvm-7.0.1.src.tar.xz
mkdir llvm
cd llvm
tar -xvf /path/to/llvm-7.0.1.src.tar.xz
mkdir llvm_build
mkdir llvm_install
cd llvm_build
cmake \
  -DCMAKE_BUILD_TYPE=Release \
  -DBUILD_SHARED_LIBS=ON \
  -DCMAKE_INSTALL_PREFIX=/home/ubuntu/RealESSI_ROOT/RealESSI_Utilities/llvm/llvm_install ↩
      \
  -DLLVM_ENABLE_RTTI=ON \
  -DLLVM_INSTALL_UTILS=ON \
  -DLLVM_TARGETS_TO_BUILD:STRING=X86 \
  ../llvm-7.0.1.src
make -j8 install
```

3. Download, Build and Install Mesa

```
wget ↩
    https://gitlab.freedesktop.org/mesa/mesa/-/archive/mesa-18.3.3/mesa-mesa-18.3.3.tar.bz2
mkdir mesa
cd mesa
tar -xvf /path/to/mesa-mesa-18.3.3.tar.bz2
cd mesa-mesa-18.3.3
autoreconf --force --verbose --install
cd ../
mkdir mesa_build
mkdir mesa_install
cd mesa_build
../mesa-mesa-18.3.3/configure ↩
    --prefix=/home/ubuntu/RealESSI_ROOT/RealESSI_Utilities/mesa/mesa_install ↩
    \
--enable-opengl --disable-osmesa --disable-gallium-osmesa \
--enable-glx --with-platforms=x11 --disable-gles1 --disable-gles2 \
--disable-va --disable-gbm --disable-xvmc --disable-vdpau \
--disable-shared-glapi --disable-dri --with-dri-drivers= \
--enable-llvm ↩
    --with-llvm-prefix=/home/ubuntu/RealESSI_ROOT/RealESSI_Utilities/llvm/llvm_install ↩
    \
--with-gallium-drivers=swrast,swr --with-swr-archs=avx,avx2 --disable-egl
make -j8 install
```

4. Obtain the source of ParaView

```
git clone --recursive https://gitlab.kitware.com/paraview/paraview.git
cd paraview
git checkout v5.8.1
```

```
git submodule update --init --recursive
```

5. Obtain the source of PVESSIReader plugin

   - The source of PVESSIReader plugin can be downloaded from http://sokocalo.engr.ucdavis.edu/~jeremic/lecture_notes_online_material/_Chapter_SoftwareHardware_Post_Processing/_Real_ESSI_PVESSIReader_.zip
   - Extract the files and move the `PVESSIReader` folder to `./Plugins/`

6. Modify the cmake file to include PVESSIReader plugin in the building process. Open the file `paraview/CMakeLists.txt` using your choice of text editor. Find "`set(paraview_default_plugins`" and add "`PVESSIReader`" to the end of the list of plugins.

7. Build ParaView with Mesa

```
mkdir paraview_build
cd paraview_build
cmake -GNinja ↩
    -DOPENGL_gl_LIBRARY=/home/ubuntu/RealESSI_ROOT/RealESSI_Utilities/mesa/mesa_install/lib
    \
-DOPENGL_INCLUDE_DIR=/home/ubuntu/RealESSI_ROOT/RealESSI_Utilities/mesa/mesa_install/inclu
    \
-DOPENGL_EGL_INCLUDE_DIR= -DOPENGL_GLES2_INCLUDE_DIR= \
-DOPENGL_GLES3_INCLUDE_DIR= -DOPENGL_GLX_INCLUDE_DIR= \
-DOPENGL_egl_LIBRARY= -DOPENGL_gles2_LIBRARY= \
-DOPENGL_gles3_LIBRARY= -DOPENGL_glu_LIBRARY= \
-DOPENGL_glx_LIBRARY= -DOPENGL_opengl_LIBRARY= ../paraview \
-DPARAVIEW_USE_PYTHON=ON -DPARAVIEW_USE_MPI=ON \
-DVTK_SMP_IMPLEMENTATION_TYPE=TBB -DCMAKE_BUILD_TYPE=Release ../paraview
LD_LIBRARY_PATH=/home/ubuntu/RealESSI_ROOT/RealESSI_Utilities/llvm/llvm_install/lib ↩
    ninja
```

   Note: There are many warnings showed up when I was doing this step, but it doesn't seem to terminate the build process.

8. Run ParaView with Mesa

```
LD_LIBRARY_PATH=/home/ubuntu/RealESSI_ROOT/RealESSI_Utilities/llvm/llvm_install/lib:/home/
    paraview
```

9. Load PVESSIReader plugin into ParaView

   - Run the ParaView executable and click on Tools → Manage Plugins → Load New ...

- Find `PVESSIReader.so` at `paraview/paraview_build/lib/paraview-5.9/plugins/PVESSIReader` and click OK to load it.
- Now you should see PVESSIReader loaded in the list of plugins. Double click on it to expand advanced options and check Auto Load.
- Close the ParaView application and reopen it. Now the PVESSIReader plugin should be automatically loaded and ready to use.

209.9 Sublime Text Editor

Note: This section describes build procedure for old versions of Real-ESSI and/or its modules.

Install `sublime` text editor from `http://www.sublimetext.com/`. Then install package control to sublime in order to install plugins. (go to preferences, package control, install package.) Then install two packages:

> FEI Syntax-n-Snippets, Real-ESSI syntax and auto completion plugin for [].fei files (input files for Real-ESSI program).
>
> gmsh-Tools, syntax and autotext completion for gmsh model development tools for Real-ESSI.
>
> gmESSI-Tools, syntax and autotext completion for gmESSI model development tools for Real-ESSI.

209.10 Model Conversion/Translation using `FeConv`

FeConv allows conversion/translation of input files (models) between Real-ESSI and SASSI, Sofistik, Ansys, OpenSees and Strudyn. FeConv was developed and is maintained by Mr. Viktor Vlaski.

209.11 Build Procedures on Amazon Web Service

This section shows the steps to install a new Real-ESSI image on Amazon Web Service (AWS). This document is only intended for Real-ESSI developers, not for general users. For using Real-ESSI on AWS, please refer to Chapter 211, on page 1373 in Jeremić et al. (1989-2025).

Noted that when creating a new image, the instance type should be consistent with future usage. For example, if the user intend to launch a Real-ESSI instance using the instance type "General Purpose", such as the T2 series, the image should also be created with the same instance type. If the image is created with a different instance type, Real-ESSI will not be able to run, and the following error message will be observed:

```
Illegal instruction (core dumped)
```

209.11.1 Sign In to AWS

Here is the link to the AWS sign in page. Click "Sign In to the Console" button on the upper right corner of the page. No need to register a new account. You should already have the account ID, IAM user name, and password for AWS sign in. If not, please contact an administrator to add you to the developers' group.

After sign in, go to the "EC2" tab under "Service". Here you can view all your instances and AMIs. This is where you can start new simulations or install new images.

Note that you probably also need to choose the correct region. On the upper right corner of the page, you can see your current region and switch to another one if necessary.

209.11.2 Copy an Existing Image

Since we already have a few images for Real-ESSI, the most efficient way to create a new image is to simply copy an existing one. To do this, go to the "AMIs" tab under "IMAGES" on the left part of the page. Now you should be able to view all existing images.

Select the image that you want to copy. Click the "Actions" button and choose "Copy AMI". On the pop-up window, enter the informations of this new image that you want to create. Then just click the "Copy AMI" button.

Now you should have a new image that has been installed with all the Real-ESSI components. To make any change inside this image, you need to launch it as a new instance and access it using X2GO. Procedures to install and use X2GO can be found in Chapter 211. For cloud server, on AWS or similar, the build procedures are the same as those for local installation, which can be found in previous sections of this chapter.

209.11.3 Create a New Image

If you need to create a new Real-ESSI image from scratch, this section shows the steps to do so. First sign in to AWS and go to "EC2". Choose the correct region. Click the "Instance" tab under "INSTANCES" on the left part of the page. Choose "Launch Instance" to start a new instance that later will be saved as your new image.

Then, follow these steps:

1. Choose AMI: Ubuntu Server 16.04 LTS (HVM), SSD Volume Type.

2. Choose Instance Type: Family = Compute optimized, Type = c5.4xlarge, vCPUs = 16, Memory (GiB) = 32.

3. Keep other options as default, and click "Review and Launch".

4. Review the information of the new instance, and click "Launch".

Next, you are asked to choose a key pair for your instance. It's recommended to create a new key pair for the first time, then use it in the future. First, choose "Create a new key pair", and enter a name. Click the "Download Key Pair" button. Save the key in a secure directory in your local computer for future use, for example in .ssh directory.

Now, you can select "Choose an existing key pair", and select your key pair that should be visible. Check the box for acknowledging the use of a private key. Finally, your new instance is launched. Note that this new instance is a brand new Ubuntu server, which means that you need to install everything.

At this point, the new Ubuntu server on AWS does not have X2GO for remote access or a GUI desktop to operate. We will now install these necessary softwares. First, run the following command to access the remote Ubuntu server on AWS using ssh. Note that you need to change the name of your ssh key to the one you just created. The public IP address can be found on the AWS webpage where you launched your new instance. Go the description of your instance to find the "IPv4 Public IP".

```
chmod 400 your_ssh_key.pem
ssh -i your_ssh_key.pem ubuntu@your_AWS_public_IP_address
```

Run the following command to install X2GO server on Ubuntu Linux.

```
sudo apt-get install software-properties-common
sudo add-apt-repository ppa:x2go/stable
sudo apt-get update
sudo apt-get install x2goserver x2goserver-xsession
```

Xfce is a lightweight desktop and ideal for usage on a remote server. Run the following command to install xfce on Ubuntu.

```
sudo apt-get install xfce4 xfce4-goodies
```

Now you can access your new instance (the remote Ubuntu server) using X2GO. Steps to do this can be found in Chapter 211. After you established remote control of the Ubuntu server on AWS, the build procedures are the same as those for local installation, which can be found in previous sections of this chapter.

The last step is to create a new image from this instance so that you can launch it in the future. Go the "Instances", and choose the correct instance. Click "Actions", and select "Create Image" under

"Image". You can change the size of the instance volume, but it's not necessary at this moment. Give your image a name and a description, and click "Create Image". Now you have sucessfully created a new image for Real-ESSI. If you go to "AMIs", you should be able to see this new image you just created.

209.11.4 Build AWS ESSI Image from Scratch

This section is a developer guide, which presents the procedures to build AWS ESSI Image from scratch. ESSI AWS users do not need to know the technical details in this section.

1. Launch EC2 instance from an AWS blank image: Ubuntu 16.04 Server.

   EC2 Dashboard → Instances → Instance → Launch Instance.

   Choose

   ```
   Ubuntu Server 16.04 (HVM), SSD Volume Type.
   ```

   Since there is no Desktop version available, so we have to launch the server version and install desktop by ourself.

   You will need to download a .pem key to launch the instance.

2. Login to the Remote Instance using Terminal.

   Copy the external IP address of the remote instance from the Browser.

   Use the downloaded .pem key to login to the remote instance.

   ```
   chmod 400 your_key.pem
   ssh -i your_key.pem ubuntu@your_remote_instance_IP
   ```

3. Install Desktop and git on AWS Remote Instance

   ```
   sudo apt update
   sudo apt install -y ubuntu-desktop git
   ```

4. Install remote-desktop-server (x2goserver) on AWS Remote Instance

   ```
   sudo add-apt-repository ppa:x2go/stable
   sudo apt update
   sudo apt install -y x2goserver x2goserver-xsession xfce4
   ```

5. Set up the automatic launch of remote desktop server

   ```
   sudo systemctl enable x2goserver.service
   sudo systemctl start x2goserver.service
   ```

6. Install ESSI

```
# Install prerequisite
sudo apt install -y cmake
sudo apt install -y build-essential
sudo apt install -y zlib1g-dev
sudo apt install -y libtbb-dev
sudo apt install -y bison flex
sudo apt install -y libboost-dev
sudo apt install -y python
sudo apt install -y gfortran
sudo apt install -y libopenblas-dev
sudo apt install -y liblapack-dev
sudo apt install -y python-scipy
sudo apt install -y libhdf5-dev libhdf5-cpp-11
sudo apt install -y python-h5py
sudo apt install -y python-matplotlib
sudo apt install -y libssl-dev

# Download ESSI
#
# using curly brackets to help in checking scripts, that rely on these
# brackets being available around URL
#
git clone {https://github.com/BorisJeremic/Real-ESSI.git} # Need ↩
   permission from Boris Jeremic for Real-ESSI on github
cd Real-ESSI

# Build ESSI Dependencies
./build_libraries download
./build_libraries sequential
./build_libraries hdf5_sequential
./build_libraries suitesparse
./build_libraries arpack
./build_libraries parmetis
./build_libraries petsc

# Build Sequential ESSI
mkdir build
cd build
cmake ..
make -j $(nproc)
cd ..


# Build Parallel ESSI
mkdir build_parallel
cd build_parallel
cmake -DCMAKE_CXX_COMPILER=/usr/bin/mpic++ -DPETSC_HAS_MUMPS=TRUE ↩
   -DPROGRAMMING_MODE=PARALLEL ..
make -j $(nproc)
cd ..
```

7. Install gmsh

```
sudo apt install -y gmsh
```

8. Install gmESSI

```
# Install the prerequisite
sudo apt install -y libboost-all-dev
sudo apt install -y build-essential
sudo apt install -y python-dev
sudo apt install -y liboctave-dev

# Install gmESSI
## download the package from the main Real-ESSI repository
#
# using curly brackets to help in checking scripts, that rely on these
# brackets being available around URL
#
wget ↩
   {http://sokocalo.engr.ucdavis.edu/~jeremic/Real_ESSI_Simulator/gmESSI/_all_files_gmESSI
mkdir Real-ESSI-gmESSI
mv _all_files_gmESSI_.tgz Real-ESSI-gmESSI
cd Real-ESSI-gmESSI

make -j $(nproc)

# Add binary PATH to ~/.bashrc
cd ./build/bin/
part1="export PATH=\""
part2=$PWD
part3=":\$PATH\""
newline=$part1$part2$part3
echo $newline >> ~/.bashrc
```

9. Install ParaView with PVESSIReader plugin

```
# Install the prerequisite
sudo apt install -y libavformat-dev
sudo apt install -y libswscale-dev
sudo apt install -y ffmpeg
sudo apt install -y libphonon-dev libphonon4 qt4-dev-tools
sudo apt install -y libqt4-core libqt4-gui qt4-qmake libxt-dev
sudo apt install -y g++ gcc cmake-curses-gui libqt4-opengl-dev
sudo apt install -y mesa-common-dev python-dev
sudo apt install -y libvtk6.2
sudo apt install -y mpich libopenmpi-dev
sudo apt install -y libxmu-dev libxi-dev

# Download the ParaView
#
```

```
# using curly brackets to help in checking scripts, that rely on these
# brackets being available around URL
#
git clone {https://github.com/Kitware/ParaView.git}
cd ParaView
git checkout v5.1.2
git submodule update --init

# Download the Plugin
cd Plugins
#
# using curly brackets to help in checking scripts, that rely on these
# brackets being available around URL
#
wget ↩
   {http://sokocalo.engr.ucdavis.edu/~jeremic/Real_ESSI_Simulator/pvESSI/_pvESSI_all_files
tar -xvzf _pvESSI_all_files_.tgz
cd ..

# Compile ParaView along with PVESSIReader
mkdir build && cd build
cmake -DPARAVIEW_USE_MPI=true -DPARAVIEW_ENABLE_PYTHON=true ↩
   -DPARAVIEW_ENABLE_FFMPEG=true ..
make -j $(nproc) # require Internet during ParaView compilation.

# Add binary PATH to ~/.bashrc
cd bin
part1="export PATH=\""
part2=$PWD
part3=":\$PATH\""
newline=$part1$part2$part3
echo $newline >> ~/.bashrc
```

10. Install Sublime Text 3 and ESSI plugin. Following this link.

```
#
# using curly brackets to help in checking scripts, that rely on these
# brackets being available around URL
#
wget -qO - {https://download.sublimetext.com/sublimehq-pub.gpg} | sudo ↩
   apt-key add -
sudo apt-get install apt-transport-https
echo "deb {https://download.sublimetext.com/ apt/stable/}" | sudo tee ↩
   /etc/apt/sources.list.d/sublime-text.list
sudo apt-get update
sudo apt-get install sublime-text
```

11. Install Sublime Text Plugin:

```
# Inside Sublime Text Window
```

```
# Ctrl+Shift+P, then Type
install package control

# Ctrl+Shift+P, then Type
install package # press ENTER, then type
fei syntax-n-snippets

# Ctrl+Shift+P, then Type
install package # press ENTER, then type
gmESSI-Tools

# Ctrl+Shift+P, then Type
install package # press ENTER, then type
gmsh-Tools
```

12. Create Image inside Browser.

    Select the launched Image with the above software installed.

    Choose Actions → Image → Create Image.

    Type your Image Name and descriptions.

    You will then see your image in EC2 Dashboard → Images → AMIs

209.11.5 Update an Existing Image

For updating an existing image, for example for a new version or release follow instruction below. First sign in to AWS and go to "EC2". Choose the correct region. Click the "Instance" tab under "INSTANCES" on the left part of the page. Choose "Launch Instance" to start a new instance that later will be saved as your new image.

Then, follow these steps:

1. Choose an existing AMI, for example GlobalRelease...
2. Choose Instance Type, for example: Family = Compute optimized, Type = c5.4xlarge, vCPUs = 16, Memory (GiB) = 32.
3. Keep other options as default, and click "Review and Launch".
4. Review the information of the new instance, and click "Launch".

Next, you are asked to choose a key pair for your instance. It's recommended to create a new key pair for the first time, then use it in the future. That is the keypair that is saved, for example in .ssh.

Now, you can select "Choose an existing key pair", and select your key pair that should be visible. Check the box for acknowledging the use of a private key. Finally, your new instance is launched. Note that this new instance is an already existing Ubuntu server/image. This image is the one we will update.

Now you can access your new instance (the remote Ubuntu server) using X2GO. Steps to do this can be found in Chapter 211. After you established remote control of the Ubuntu server on AWS, the build procedures are the same as those for local installation, which can be found in previous sections of this chapter.

The last step is to create a new image from this instance so that you can launch it in the future. Go the "Instances", and choose the correct instance. Click "Actions", and select "Create Image" under "Image". You can change the size of the instance volume, but it's not necessary at this moment. Give your image a (new) name and a description, and click "Create Image". Now you have sucessfully created a new image for Real-ESSI. If you go to "AMIs", you should be able to see this new image you just created.

Now you can go to `Software` directory and follow install procedures from section ?? on page ??.

After compiling and linking both sequential and parallel Real-ESSI, and install them on /usr/bin (follow procedures for build), and delete source code (!), one can make this instance into a new image. Create new image inside AWS EC2 Management Console Browser window. Select the launched Image with the above software installed. Choose Actions → Image → Create Image. Type your Image Name and descriptions. Click Create Image. This might take some time. You will then see your image in EC2 Dashboard → Images → AMIs (on the left side bar).

Make sure that you terminate all the running instances so that you do not get charged. Find: Action, Instance State, Terminate.

209.11.6 Upload an Existing Real-ESSI Simulator Image to AWS MarketPlace

- Copy to private image for region North Virginia
- Go to the AWS market place https://aws.amazon.com/marketplace,
- Choose sell in AWS marketplace,
- Choose AMIs selection the new private in Region North Virginia to publish.
- Proceed until finalizing the AWS Marketplace Image.

Chapter 210

Software Platform Procurement, Distribution

(2019-2020-2021-)

(In collaboration with Prof. Han Yang and Dr. Hexiang Wang)

210.1 Chapter Summary and Highlights

210.2 Introduction

The Real-ESSI Simulator program (Jeremić et al., 1988-2025) can be installed on user's computers in a number of different ways:

- The most efficient executables are created when Real-ESSI sources are compiled on user computer. Compilation is performed using batch scripts that execute all the necessary operations. This process takes approximately 40 minutes, for both sequential and parallel versions of the Real-ESSI. It is assumed that all the necessary libraries are installed prior to this. More details about this mode of installation are given in section 209, on page 1324 of the main document (Jeremić et al., 1989-2025). For this mode of installation, sources for the Real-ESSI need to be made available. Sources for the Real-ESSI program are usually not distributed, except to collaborators and in some other special circumstances.

- The Real-ESSI program can also be downloaded and installed as a Debian package, starting from version 22.07, built for Ubuntu 22.04 LTS. The Debian package contains the sequential and parallel Real-ESSI executables, The gmESSI tool for pre-processing using Gmsh, and pvESSI tool for post-processing using ParaView and other useful external programs, like Gmsh and ParaView, will NOT be automatically installed when installing the Real-ESSI Debian package. This change was made since those other packages should be installed using their own installation procedures, that have gone though some recent changes. Therefor installation of those packages is best done directly using downloaded version from their own web site, and then connecting them to the Real-ESSI Simulator systems using Gmsh and pvESSI tools. Installation of Gmsh and pvESSI tools is described in:

  - Installation of pre-processing modules is described in Chapter 207, on page 1221 in Jeremić et al. (1989-2025).
  - Installation of post-processing modules is described in Chapter 208, on page 1287 in Jeremić et al. (1989-2025).

  It is noted that old installations of Real-ESSI main program and gmsh and ParaView should be removed before the Real-ESSI Simulator systems is installed from Debian package.

- The Real-ESSI program can also be installed through direct download of program executables, as noted in section 210.4. These executable were build without use of any special optimization options, so they are not very efficient, and do not use special, high performance features of most modern CPUs. On the other hand these generic executables will run, execute on most computers.

- Docker support is discontinued since Windows users can now use Windows Subsystem for Linux (WSL), and install Real-ESSI Simulator Debian package. The Real-ESSI program can also be installed through a docker container, as described in section 210.5. Similar to the previous case, these executable were developed without special optimization options, so they are not very efficient. However, Real-ESSI program will these generic executables will run, within docker container, on all computers.

210.3 Real-ESSI Program Debian Package Download and Install

210.3.1 System Libraries Update/Upgrade

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get dist-upgrade
sudo apt-get autoremove
```

`sudo apt-get update`

`sudo apt-get upgrade`

`sudo apt-get dist-upgrade`

`sudo apt-get autoremove`

210.3.2 Real-ESSI Debian Package Download

The Real-ESSI program Debian package can be downloaded from Real-ESSI Simulator website: `http://real-essi.info/`. Alternatively, contact Prof. Jeremić to arrange for customized Real-ESSI Debian package.

210.3.3 Real-ESSI Debian Package Install

Start the Real-ESSI Simulator Debian package install by removing the old installations of Real-ESSI program, pre-processor gmsh/gmESSI and post-processor ParaView/pvESSI. Then, go to the directory where you have downloaded the Real-ESSI Debian package. Install the Debian package, for example use the following command:

```
sudo apt install ./real-essi_22.07-1_amd64.deb
```

Note that some warning messages might appear but they don't affect the installation.

After a successful installation, the sequential and parallel Real-ESSI executables. are installed and ready to use.

210.3.4 Load pvESSI Plugin in ParaView

Install ParaView system using installation procedure described on their web site. Then install pvESSI plugin. Start `ParaView` and click 'Tools' → 'Manage Plugins...'. Click 'Load New...' and find the plugin named 'PVESSIReader.so' under directory `/opt/paraview/lib/paraview-5.10/plugins/PVESSIReader/`. Also check the box 'Auto Load' then close `ParaView`. Next time when ParaView is started, Real-ESSI output files can be visualized and post-processed.

210.3.5 Install Other Useful Programs

210.3.5.1 HDFView

`HDFView` can be used to open Real-ESSI output files, which are in HDF5 format. Download the latest version of `HDFView` from `https://support.hdfgroup.org/ftp/HDF5/releases/HDF-JAVA/`. Click on the latest version, which is `hdfview-3.2.0` as of June 2022. Go to `bin/`, click `HDFView-3.2.0-ubuntu2004_64.tar.gz` and save the file in your `./Downloads/` directory. Then extract and install HDFView:

```
cd
tar -xvf ./Downloads/HDFView-3.2.0-ubuntu2004_64.tar.gz -C ./Downloads
sudo apt install -y ./Downloads/hdfview_3.2.0-1_amd64.deb
sudo ln -s /opt/hdfview/bin/HDFView /usr/local/bin/hdfview
```

Now you can use `HDFView` from a terminal. To be able to use `HDFView` when you click on a Real-ESSI output file, do the following additional steps. First open the file using the following command:

```
sudo gedit /usr/share/applications/hdfview-HDFView.desktop
```

Find the line:

```
Exec=/opt/hdfview/bin/HDFView
```

Replace it with:

```
Exec=/opt/hdfview/bin/HDFView %F
```

Save the file and close it.

Go to a Real-ESSI output file, which should have the suffix 'h5.feioutput'. Right click on the file and select 'Open with Other Application'. Click 'View All Applications' and choose `HDFView` from the list.

Note that you only need to do this once. Next time when you click on a Real-ESSI output file, it will be opened automatically using `HDFView`.

210.3.5.2 Sublime Text

Sublime Text (https://www.sublimetext.com/) is the recommended editor for Real-ESSI input files and pre-processing files. Install Sublime Text using the following command:

```
wget -qO - https://download.sublimetext.com/sublimehq-pub.gpg | gpg --dearmor | ↩
   sudo tee /etc/apt/trusted.gpg.d/sublimehq-archive.gpg
echo "deb https://download.sublimetext.com/ apt/stable/" | sudo tee ↩
   /etc/apt/sources.list.d/sublime-text.list
sudo apt-get update
sudo apt-get install sublime-text
```

Open Sublime Text. Open the 'Tools' menu and select 'Install Package Control...'. Open the 'Preferences' menu, select 'Package Control', then select 'Package Control: Install Package'.

In the opened search bar, type the package name and click on the package to install it. Three packages should be installed: `FEI Syntax-n-Snippets`, `gmsh-Tools`, and `gmESSI-Tools`.

210.4 Real-ESSI Program Executables Download and Install

Executables for the Real-ESSI Simulator program (Jeremić et al., 1988-2025) are available online. Prebuilt executables are available for Linux, Ubuntu 18.04, and can be downloaded and installed by analyst.

In order for prebuild executables to be able to run on a user/analyst computer, system libraries have to be brought up to date and additional libraries installed. System libraries update/upgrade:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get dist-upgrade
sudo apt-get autoremove
```

For sequential and/or parallel version of Real-ESSI, additional libraries are needed, as described below.

210.4.1 Sequential Version of Real-ESSI Program.

Libraries required to be installed for using sequential version of the Real ESSI program:

```
sudo apt-get install libboost-all-dev
sudo apt-get install libhdf5-dev
sudo apt-get install libtbb-dev
sudo apt-get install libssl1.0.0
```

210.4.2 Parallel Version of Real-ESSI Program.

Libraries required to be installed for executing parallel version of the Real ESSI program:

```
sudo apt-get install libboost-all-dev
sudo apt-get install libhdf5-dev
sudo apt-get install libtbb-dev
sudo apt-get install mpich
sudo apt-get install libopenmpi-dev
sudo apt-get install libssl1.0.0
```

210.4.3 Real-ESSI Executable Downloads.

The Real-ESSI program executables can be downloaded from Real-ESSI Simulator website: `http://real-essi.info/`. Alternatively, contact Prof. Jeremić to arrange for customized Real-ESSI executables.

210.5 DISCONTINUED, use WSL! Real-ESSI Simulator Install as Container through Docker

Docker support is discontinued since Windows users can now use Windows Subsystem for Linux (WSL), and install Real-ESSI Simulator Debian package.

Recent developments in virtualization of operating systems (OS) has created an opportunity to deploy programs and software systems as container images. Container images are used by the host OS (Linux, Windows, MacOS) to create a container. A container is a running instance of a container image, and is represented by a Linux/Windows/MacOS process that can be used to run programs that are installed within container. Programs that are installed within a container have all the necessary libraries available within container and are fully self sufficient, irrespective of what container host OS is used, be it Linux or Windows or MacOS.

More information used virtualization, containers, docker, etc. can be found at:

- `https://en.wikipedia.org/wiki/OS-level_virtualization`
- `https://en.wikipedia.org/wiki/Docker_(software)`
- `https://developers.redhat.com/blog/2018/02/22/container-terminology-practical-introduction`

Starting from Real-ESSI version 20.07, Real-ESSI Simulator is now available as a Docker Container Image, and can be installed and used on Linux, Windows and MacOS.

210.5.1 DISCONTINUED, use WSL! Real-ESSI Docker Image Development

This section is intended for Real-ESSI developers, users can skip this section. The development of Real-ESSI Docker image follows typical steps to 'dockerize' any application. Here are some very helpful sources:

- Official documentation: `https://docs.docker.com/`
- A Docker Tutorial for Beginners:
  `https://docker-curriculum.com/#our-first-image`
- How to dockerize any application:
  `https://hackernoon.com/how-to-dockerize-any-application-b60ad00e76da`
- Slimming Down Your Docker Images:
  `https://towardsdatascience.com/slimming-down-your-docker-images-275f0ca9337e`

It should be mentioned that there are many different ways and styles that can be employed to create Docker image. Here, multistage build is used to save build/debug time and, more importantly, reduce size of the final image.

Provided below are steps used to create the Real-ESSI Docker image.

- Obtain the source code of Real-ESSI.
- The following 'Dockerfile' is created to build the Real-ESSI Docker image.

```
FROM ubuntu:18.04 AS basesystem

MAINTAINER Han Yang <hhhyang@ucdavis.edu>

WORKDIR /usr/src

COPY . .

RUN useradd -m ubuntu && \
       apt-get update && apt-get install -y \
       bison \
       build-essential \
       cmake \
       flex \
       libboost-all-dev \
       libhdf5-serial-dev \
       liblapack-dev \
       libopenblas-dev \
       libopenmpi-dev \
       libpthread-workqueue-dev \
```

```
        libssl-dev \
        libtbb-dev \
        mpich \
        ssh \
        valgrind \
        wget \
        zlib1g-dev



FROM basesystem AS dependencies

RUN cd Real-ESSI && \
        mkdir -p ../RealESSI_Dependencies && \
        mkdir -p ../RealESSI_Dependencies/include && \
        mkdir -p ../RealESSI_Dependencies/lib && \
        mkdir -p ../RealESSI_Dependencies/bin && \
        mkdir -p ../RealESSI_Dependencies/SRC && \
        cd ../RealESSI_Dependencies && \
        wget ↩
            http://sokocalo.engr.ucdavis.edu/~jeremic/RealESSI/Dependencies_SRC.tar.gz ↩
            && \
        tar -xzvf ./Dependencies_SRC.tar.gz -C ./SRC --strip-components 1 ↩
            && \
        cd ../Real-ESSI && \
        ./build_libraries suitesparse && \
        ./build_libraries arpack && \
        ./build_libraries hdf5_sequential && \
        ./build_libraries tbb && \
        ./build_libraries lapack && \
        ./build_libraries parmetis && \
        ./build_libraries petsc_itself



FROM dependencies AS builder

RUN cd Real-ESSI && \
        mkdir build && \
        cd build && \
        cmake .. && \
        make -j 16 && \
        cp essi essi_sequential && \
        cd .. && \
        mkdir pbuild && \
        cd pbuild && \
        cmake -DCMAKE_CXX_COMPILER=/usr/bin/mpic++ ↩
            -DPROGRAMMING_MODE=PARALLEL .. && \
        make -j 16 && \
        cp essi essi_parallel
```

```
FROM ubuntu:18.04

MAINTAINER Han Yang <hhhyang@ucdavis.edu>

RUN useradd -m ubuntu && \
       apt-get update && apt-get install -y \
       libboost-all-dev \
       libhdf5-dev \
       libopenmpi-dev \
       libtbb-dev \
       mpich \
       ssh

COPY --from=builder /usr/src/Real-ESSI/build/essi_sequential ↩
   /usr/src/Real-ESSI/pbuild/essi_parallel /usr/bin/

USER ubuntu

WORKDIR /workspace

VOLUME ["/workspace"]
```

- Put the 'Dockerfile' in the same directory with the source code of Real-ESSI.
- Build the Real-ESSI Docker image. This step usually takes a long time, especially for the first time.

```
docker build -t realessilocal:test .
```

- Correctly tag your image. This is not only necessary for later push but also just a good practice to organize your Docker images.

```
docker tag realessilocal:test realessi/real-essi-repo:<tag>
```

  Replace <tag> with the tag you want to use. It's usually a version name.

- Push your build to Docker Hub. Make sure you have the proper permission to do so.

```
docker push realessi/real-essi-repo:<tag>
```

210.5.2 DISCONTINUED, use WSL! Running Real-ESSI Container through Docker

Provided below are steps needed to install and run Real-ESSI within a Docker Container. The following steps work for both Linux and Windows systems. In a Linux system, run the following commands in a terminal. In a Windows system, run these commands in PowerShell. It should also work for Mac OS but hasn't been tested yet.

- Install Docker on the local computer, desktop, laptop. Documentation on how to install Docker on user OS can be found here:
  - Linux: https://docs.docker.com/engine/install/#server
  - Windows: https://docs.docker.com/docker-for-windows/install/
  - MacOS: https://docs.docker.com/docker-for-mac/install/
- Manage Docker as a non-root user on Linux hosts

  If you are using a Linux host, by default you need to run Docker using `sudo`. If you don't want to preface the docker command with `sudo`, create a group called `docker` and add users to it.

  To create the `docker` group and add your user:

  1. Create the `docker` group.

     ```
     sudo groupadd docker
     ```

     Sometimes the `docker` group might already exist after the installation of Docker. This is okay, just move on to the next step.
  2. Add your user to the `docker` group.

     ```
     sudo usermod -aG docker $USER
     ```

     Replace `$USER` with your user name.
  3. Log out and log back in so that your group membership is re-evaluated. On Linux, you can also run the following command to activate the changes to groups:

     ```
     newgrp docker
     ```

  4. Verify that you can run `docker` commands without `sudo`.

     ```
     docker run hello-world
     ```

     This command downloads a test image and runs it in a container. When the container runs, it prints an informational message and exits.

  More information on managing Docker as a non-root user can be found here: https://docs.docker.com/engine/install/linux-postinstall/

- Pull the Real-ESSI image

```
docker pull realessi/real-essi-repo:tag
```

  Replace `tag` with the latest version of Real-ESSI. For example, if the latest version is 23.01, then the pull command is `docker pull realessi/real-essi-repo:23.01`.

  Current Real-ESSI Simulator version is kept up to date at the Real-ESSI web site HERE. In adition, you can find tags of Real-ESSI at `https://hub.docker.com/repository/docker/realessi/real-essi-repo/tags`.

- Run the Real-ESSI image:

```
docker run -it --rm -v your_working_directory:/workspace ↩
   realessi/real-essi-repo:tag
```

  Again, replace `tag` with the version of Real-ESSI you pulled. Once you start running the Real-ESSI Docker image, you are working inside the container. The container is Ubuntu 18.04 with Real-ESSI installed. Note that you should replace `your_working_directory` with the absolute path of your working directory.

- Run Real-ESSI:

```
essi_sequential -f main.fei
```

Note that the current directory on your local machine is shared with the container, so it can work with any files there. The files need to have the correct permissions to be run by a non-administrator user. You can move files after the container started and they will be recognized by the container.

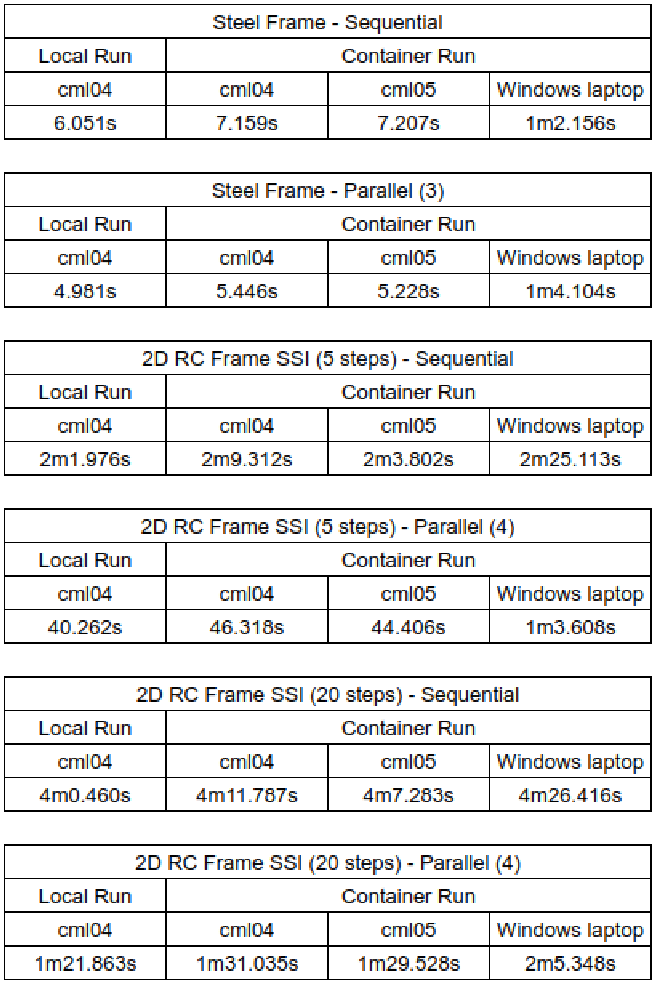
After the simulation is finished, simply exit the container. You will see the output files and log file in your current directory. They will not be erased when you exit the container.

210.5.3 DISCONTINUED, use WSL! Performance of Real-ESSI Container

To test the performance of Real-ESSI container, a series of sequential and parallel simulations are conducted. The results and comparison are summarized in Figure 210.1.

210.6 Real-ESSI Simulator System Install

In adition to the Real-ESSI Program, Real-ESSI Simulator system consists of a pre-processing modules and post-processing modules. Installation of pre-processing modules is described in Chapter 207, on



Steel Frame - Sequential			
Local Run	Container Run		
cml04	cml04	cml05	Windows laptop
6.051s	7.159s	7.207s	1m2.156s

Steel Frame - Parallel (3)			
Local Run	Container Run		
cml04	cml04	cml05	Windows laptop
4.981s	5.446s	5.228s	1m4.104s

2D RC Frame SSI (5 steps) - Sequential			
Local Run	Container Run		
cml04	cml04	cml05	Windows laptop
2m1.976s	2m9.312s	2m3.802s	2m25.113s

2D RC Frame SSI (5 steps) - Parallel (4)			
Local Run	Container Run		
cml04	cml04	cml05	Windows laptop
40.262s	46.318s	44.406s	1m3.608s

2D RC Frame SSI (20 steps) - Sequential			
Local Run	Container Run		
cml04	cml04	cml05	Windows laptop
4m0.460s	4m11.787s	4m7.283s	4m26.416s

2D RC Frame SSI (20 steps) - Parallel (4)			
Local Run	Container Run		
cml04	cml04	cml05	Windows laptop
1m21.863s	1m31.035s	1m29.528s	2m5.348s

Figure 210.1: Comparison of Real-ESSI performance on local Linux machine and Linux/Windows containers.

page 1221 in Jeremić et al. (1989-2025). Installation of post-processing modules is described in Chapter 208, on page 1287 in Jeremić et al. (1989-2025).

Both pre and post processing manuals are also available through the main Real-ESSI Simulator web site: `http://real-essi.info/`.

210.6.1 Student Manual for Real-ESSI Simulator System Install

Students at ETH, Mr. Max Sieber and Mr. Antonio Felipe Salazar created a manual for installation of the Real-ESSI Simulator system on virtual machine computers. The manual is available HERE.

Chapter 211

Cloud Computing

(2017-2018-2019-2021-2023)

(In collaboration with Dr. Yuan Feng, Prof. Han Yang, and Dr. Hexiang Wang)

211.1 Chapter Summary and Highlights

Described in this chapter are details of accessing and using Real-ESSI Simulator using remote computers, the so called "cloud" computational resources. Current focus is on using Amazon Web Services (AWS) computers.

211.2 Real-ESSI Cloud Computing Overview

Cloud computing refers to the accessing and computing over the Internet rather than on local computers. Cloud computing is a model for enabling on-demand access to a shared pool of configurable computing resources, which can be setup and released rapidly.[1]

Using Real-ESSI Cloud Service, users can get computing instances on demand without requiring a lot of maintenance and financial resources a common, local parallel computer, cluster would require. In addition, users do not need to go through the installation of the dependent libraries, source-code compilation and the installation of other related software, for example preprocessing and post-processing environment. The complete Real-ESSI Simulator system is pre-configured and built within the image such that Real-ESSI Simulator system is portable over the cloud. A stable, release version of Real-ESSI is built and can be used anywhere and anytime.

There are two ways to obtain a Real-ESSI image on Amazon Web Services (AWS):

- Obtain a Real-ESSI private image from Prof. Boris Jeremić, see Section 211.3.1 on page 1376.
- Use a public image of Real-ESSI on AWS marketplace, as described in Section 211.3.2 on page 1388.

After a Real-ESSI image is launched, a Real-ESSI EC2 instance is generated on AWS. The instance can be accessed through a X2GO client. The procedures are written in Section 211.4 on page 1388.

When the simulation on the Real-ESSi instance is finished and all the output result files are fetched, remember to terminate the running instance so that AWS would not keep charging you. Section 211.5 on page 1391 describes how to terminate a running Real-ESSI instance. See Section 211.8 on page 1394 for more information about the cost of AWS cloud computing services.

211.2.1 Real-ESSI Cloud Service Content

One image is built for a single-machine setup, which contains

---

[1] This is an excerpt from Jeremić et al. (1989-2025)

- Ubuntu 16.04 LTS Desktop and X2GO Server
- Real-ESSI sequential program
- Real-ESSI parallel program
- Real-ESSI 3C seismic motion developments (SW4)
- Real-ESSI pre-processing (gmESSI)
- Real-ESSI post-processing (PVESSIReader)
- Real-ESSI Editor, Sublime plug-ins
- Real-ESSI Documentation
- Real-ESSI Examples

211.3 Launch Real-ESSI Instance on AWS

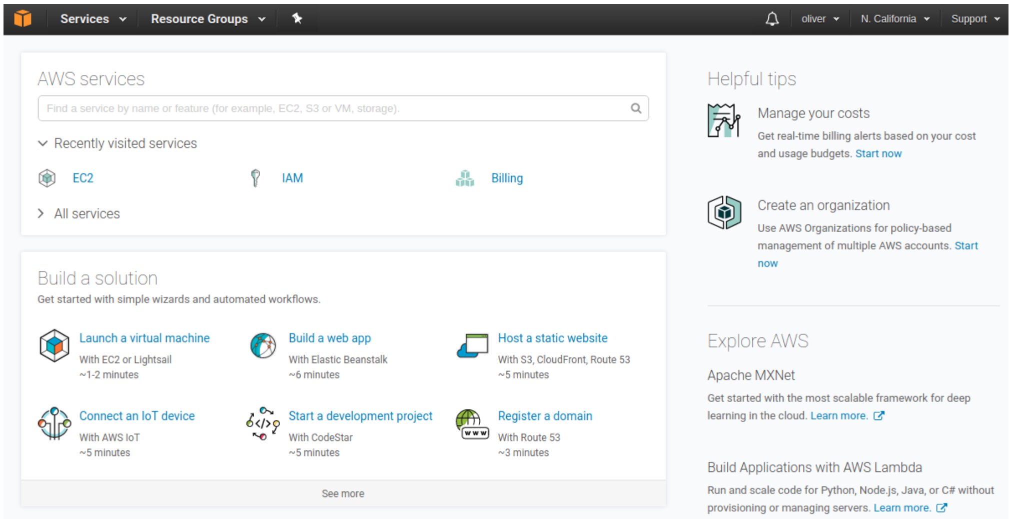
A Real-ESSI instance can be launched either from the private image with authorization of Prof. Boris Jeremić or from the public image on AWS market place.

211.3.1 Launch Real-ESSI Instance from AWS Private Images

Follow the steps below to launch instances from Real-ESSI Private Image.

1. Create an AWS account.

   AWS is the most widely used cloud service provider. If you do not have one, creating an AWS account is easy. You can create an AWS account through their website https://aws.amazon.com/. After you login, you can see the services on AWS `Console Home` as follows.



AWS services

Recently visited services

Build a solution

Get started with simple wizards and automated workflows.

Launch a virtual machine
With EC2 or Lightsail
~1-2 minutes

Build a web app
With Elastic Beanstalk
~6 minutes

Host a static website
With S3, CloudFront, Route 53
~5 minutes

Connect an IoT device
With AWS IoT
~5 minutes

Start a development project
With CodeStar
~5 minutes

Register a domain
With Route 53
~3 minutes

Helpful tips

Manage your costs
Get real-time billing alerts based on your cost and usage budgets. Start now

Create an organization
Use AWS Organizations for policy-based management of multiple AWS accounts. Start now

Explore AWS

Apache MXNet
Get started with the most scalable framework for deep learning in the cloud. Learn more.

Build Applications with AWS Lambda
Run and scale code for Python, Node.js, Java, or C# without provisioning or managing servers. Learn more.

Figure 211.1: AWS Console Home.

2. Request the Real-ESSI image.

   Real-ESSI image is currently a private Amazon Machine Images (AMI). After you get the 12-digit AWS account ID, email the AWS account ID to Prof. Boris Jeremić to obtain the Real-ESSI image. From AWS Console Home, go to `Services` → `EC2`

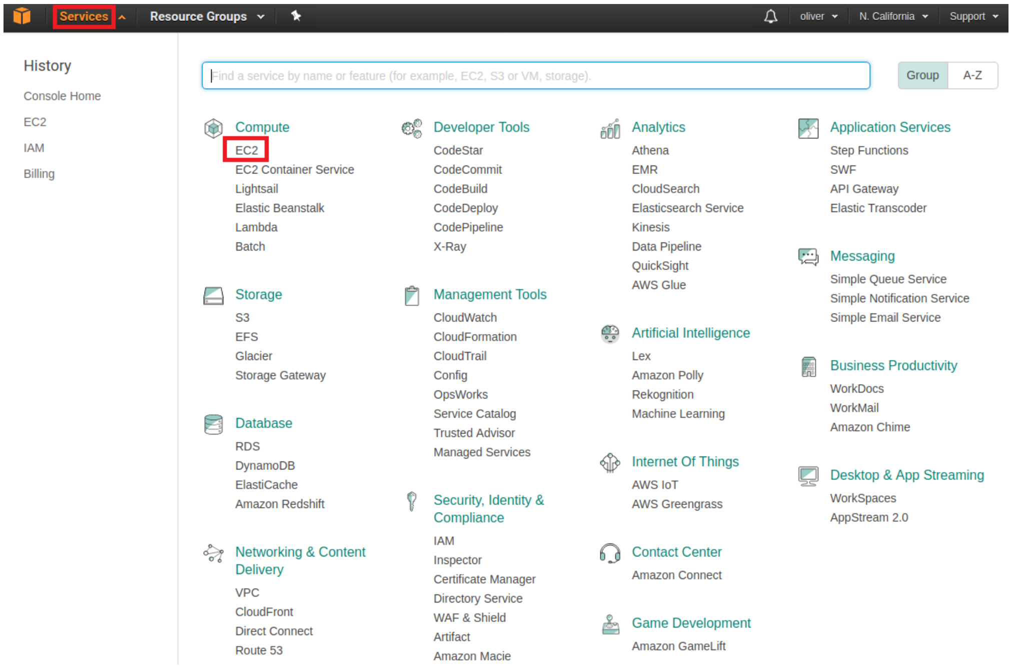
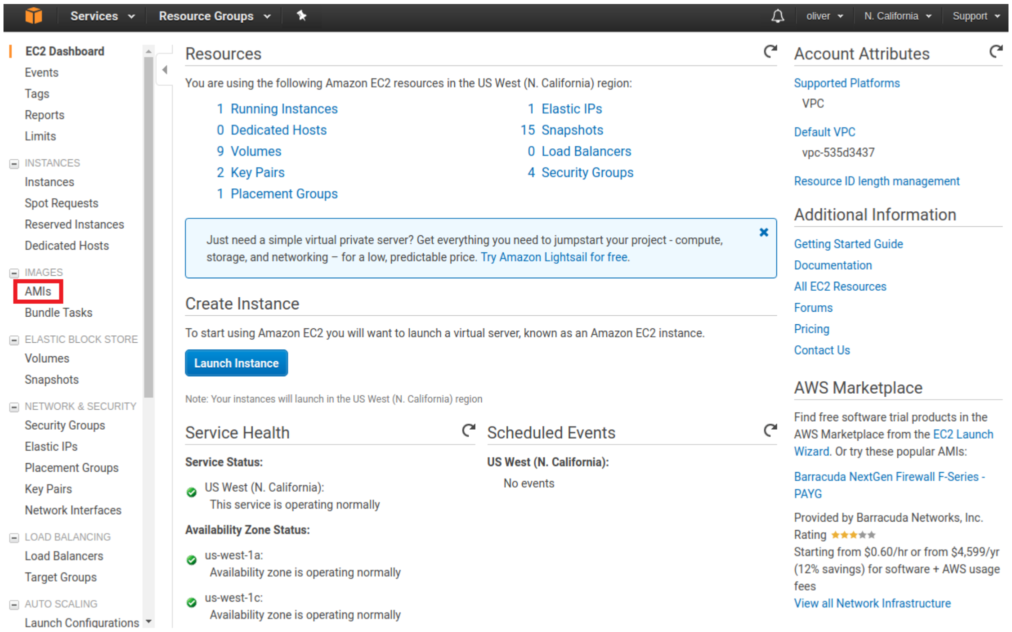


Figure 211.2: AWS Services.

From EC2 Dashboard, go to AMIs to check the Real-ESSI image.



Resources

You are using the following Amazon EC2 resources in the US West (N. California) region:

1 Running Instances
0 Dedicated Hosts
9 Volumes
2 Key Pairs
1 Placement Groups
1 Elastic IPs
15 Snapshots
0 Load Balancers
4 Security Groups

Just need a simple virtual private server? Get everything you need to jumpstart your project - compute, storage, and networking – for a low, predictable price. Try Amazon Lightsail for free.

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the US West (N. California) region

Service Health

Service Status:

US West (N. California):
This service is operating normally

Availability Zone Status:

us-west-1a:
Availability zone is operating normally

us-west-1c:
Availability zone is operating normally

Scheduled Events

US West (N. California):
No events

Account Attributes

Supported Platforms
VPC

Default VPC
vpc-535d3437

Resource ID length management

Additional Information

Getting Started Guide
Documentation
All EC2 Resources
Forums
Pricing
Contact Us

AWS Marketplace

Find free software trial products in the AWS Marketplace from the EC2 Launch Wizard. Or try these popular AMIs:

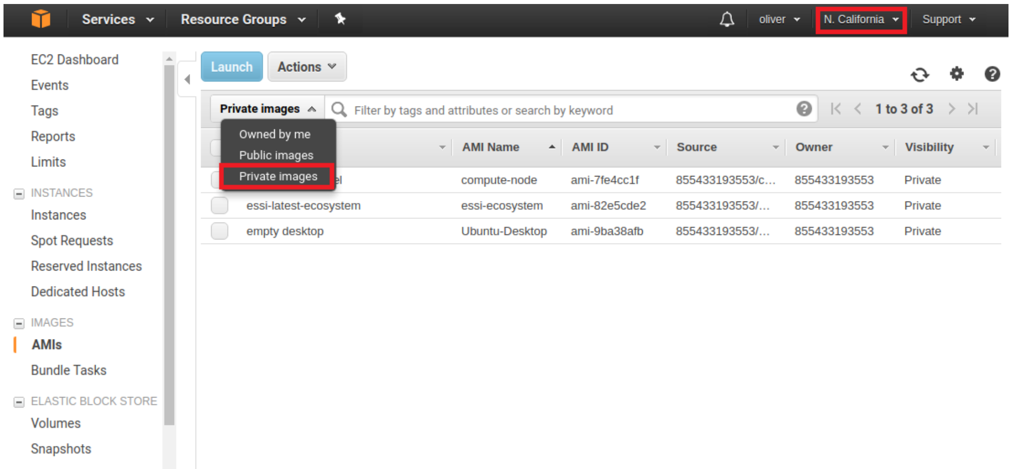
Barracuda NextGen Firewall F-Series - PAYG

Provided by Barracuda Networks, Inc.
Rating ★★★★★
Starting from $0.60/hr or from $4,599/yr (12% savings) for software + AWS usage fees
View all Network Infrastructure

Figure 211.3: AWS EC2 Dashboard AMIs.

If users cannot find the Real-ESSI image, please make sure you are in the same AWS region with Prof. Boris Jeremić, the `region` is shown in the top-right corner on EC2 dashboard. The current Real-ESSI AMIs region are in both `North California` and `Oregon`.

3. Launch the Real-ESSI image.

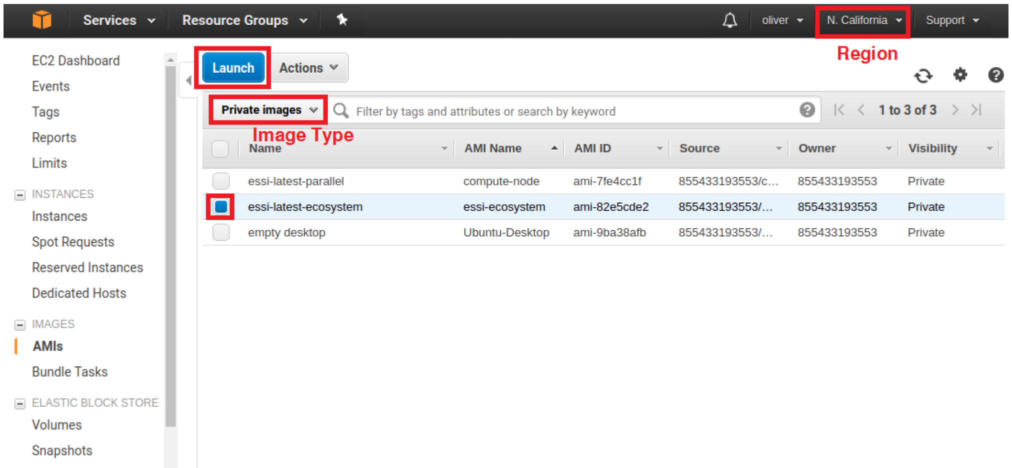


		AMI Name	AMI ID	Source	Owner	Visibility
☐	el	compute-node	ami-7fe4cc1f	855433193553/c...	855433193553	Private
☐	essi-latest-ecosystem	essi-ecosystem	ami-82e5cde2	855433193553/...	855433193553	Private
☐	empty desktop	Ubuntu-Desktop	ami-9ba38afb	855433193553/...	855433193553	Private

Figure 211.4: AWS EC2 Private AMIs.

Follow the steps below to launch instances from the Real-ESSI image.

(a) Choose AMI.

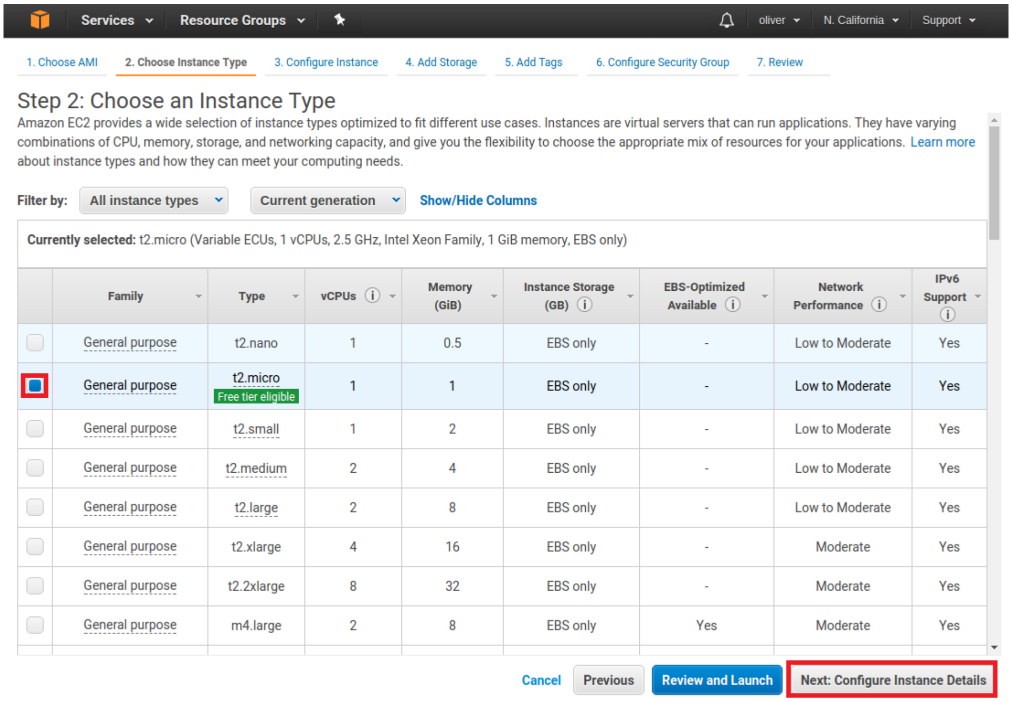


☐	Name	AMI Name	AMI ID	Source	Owner	Visibility
☐	essi-latest-parallel	compute-node	ami-7fe4cc1f	855433193553/c...	855433193553	Private
☑	essi-latest-ecosystem	essi-ecosystem	ami-82e5cde2	855433193553/...	855433193553	Private
☐	empty desktop	Ubuntu-Desktop	ami-9ba38afb	855433193553/...	855433193553	Private

Figure 211.5: EC2 Launch Steps: Choose AMI.

(b) Choose Instance Type

From AMIs, users can launch any number and type of instances and choose the desired EC2 configurations. In order to have the best experiences, the compute-optimized instances (C4, C5 as the latest one, as of early 2019) are recommended.



Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

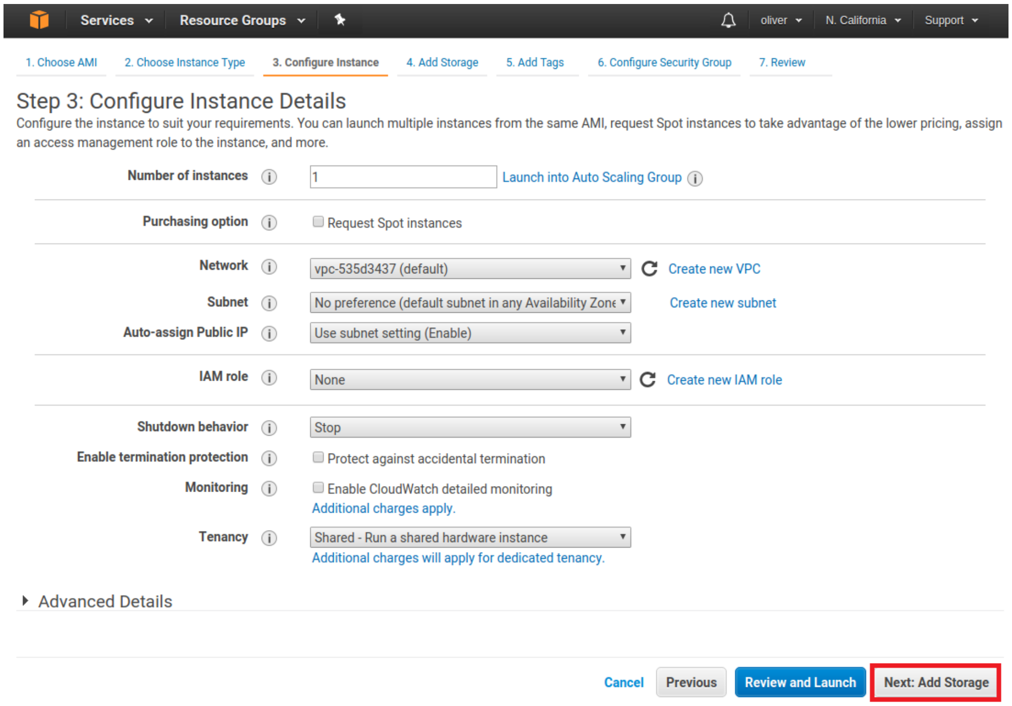
Filter by: All instance types Current generation Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
☐	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
■	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
☐	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
☐	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
☐	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
☐	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
☐	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
☐	General purpose	m4.large	2	8	EBS only	Yes	Moderate	Yes

Figure 211.6: EC2 Launch Steps: Choose Instance Type.

(c) Configure Instance



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 — Launch into Auto Scaling Group

Purchasing option: ☐ Request Spot instances

Network: vpc-535d3437 (default) — Create new VPC

Subnet: No preference (default subnet in any Availability Zone — Create new subnet

Auto-assign Public IP: Use subnet setting (Enable)

IAM role: None — Create new IAM role

Shutdown behavior: Stop

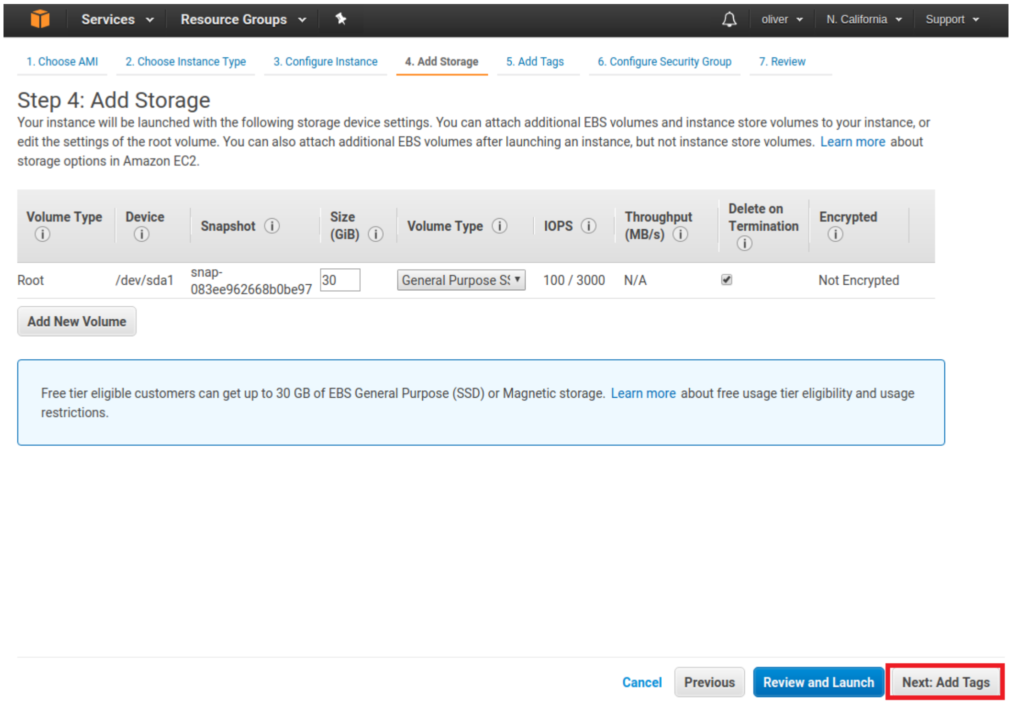
Enable termination protection: ☐ Protect against accidental termination

Monitoring: ☐ Enable CloudWatch detailed monitoring
Additional charges apply.

Tenancy: Shared - Run a shared hardware instance
Additional charges will apply for dedicated tenancy.

Figure 211.7: EC2 Launch Steps: Configure Instance.

(d) Add Storage



Step 4: Add Storage

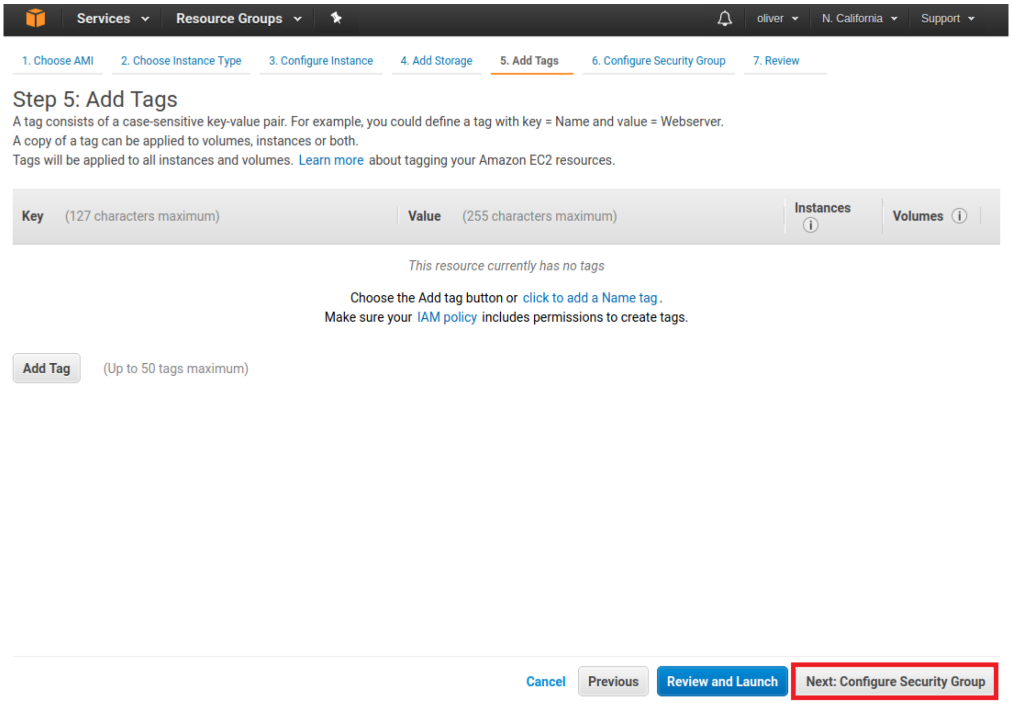
Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	snap-083ee962668b0be97	30	General Purpose S!	100 / 3000	N/A	☑	Not Encrypted

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

Figure 211.8: EC2 Launch Steps: Add Storage.

(e) Add Tags



Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)	Instances	Volumes

This resource currently has no tags

Choose the Add tag button or click to add a Name tag.
Make sure your IAM policy includes permissions to create tags.

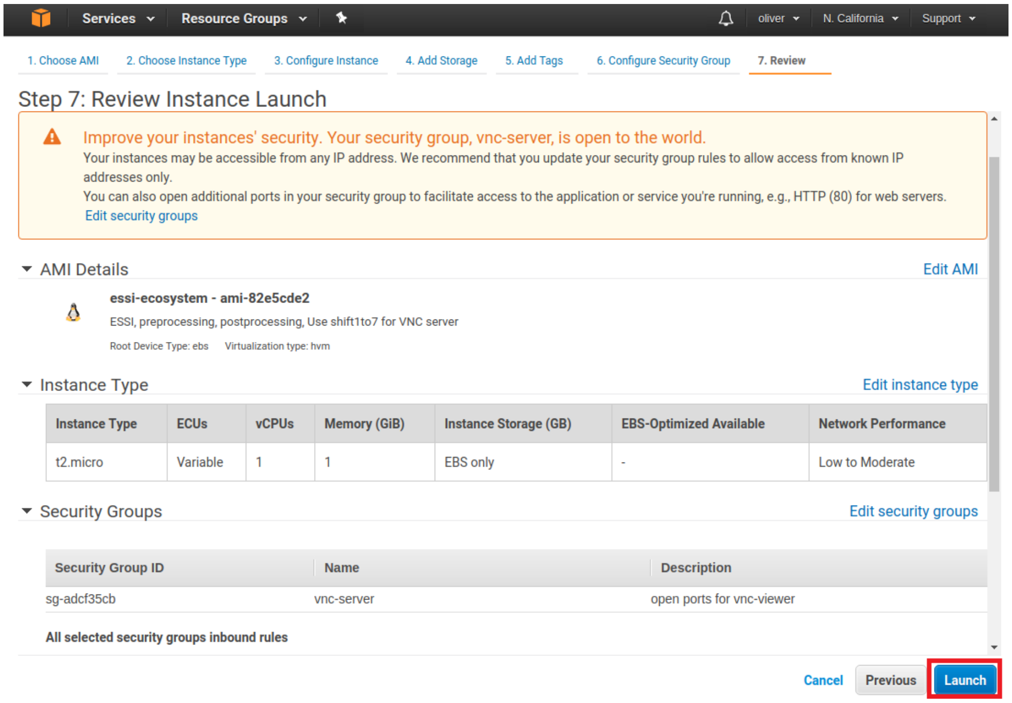
Figure 211.9: EC2 Launch Steps: Add Tags.

(f) Configure Security Group.

Please keep the default Security Group setting.

(g) Review

You may be asked to create a key-pair for later access of the instance you created. The key-pair can be reused later if you created other instances. Besides, the key-pair is portable across other machines. Last but not least, the key-pair cannot be recreated after you launch the instance, so please make sure you save the key-pair in a safe place.



Step 7: Review Instance Launch

Improve your instances' security. Your security group, vnc-server, is open to the world.

Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only.

You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers.

Edit security groups

AMI Details — Edit AMI

essi-ecosystem - ami-82e5cde2

ESSI, preprocessing, postprocessing, Use shift1to7 for VNC server

Root Device Type: ebs Virtualization type: hvm

Instance Type — Edit instance type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups — Edit security groups

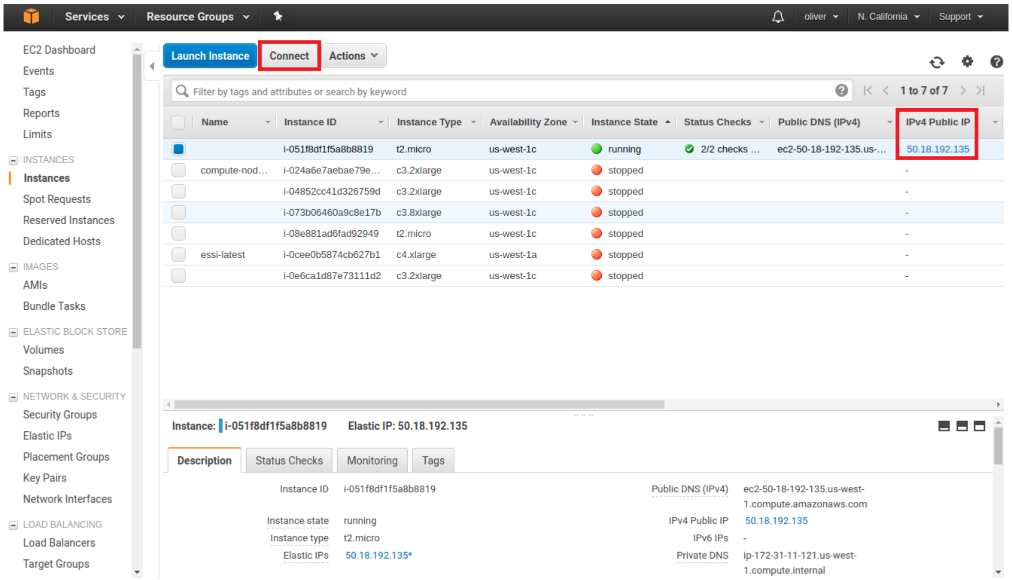
Security Group ID	Name	Description
sg-adcf35cb	vnc-server	open ports for vnc-viewer

All selected security groups inbound rules

Figure 211.10: EC2 Launch Steps: Review.

4. Check the launched instances

   After the launch, you can view the running instance through `EC2 Dashboard` → `Instances`



	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Public DNS (IPv4)	IPv4 Public IP
☑		i-051f8df1f5a8b8819	t2.micro	us-west-1c	● running	✓ 2/2 checks ...	ec2-50-18-192-135.us-...	50.18.192.135
☐	compute-nod...	i-024a6e7aebae79e...	c3.2xlarge	us-west-1c	● stopped			-
☐		i-04852cc41d326759d	c3.2xlarge	us-west-1c	● stopped			-
☐		i-073b06460a9c8e17b	c3.8xlarge	us-west-1c	● stopped			-
☐		i-08e881ad6fad92949	t2.micro	us-west-1c	● stopped			-
☐	essi-latest	i-0cee0b5874cb627b1	c4.xlarge	us-west-1a	● stopped			-
☐		i-0e6ca1d87e73111d2	c3.2xlarge	us-west-1c	● stopped			-

Instance: i-051f8df1f5a8b8819 Elastic IP: 50.18.192.135

Instance ID i-051f8df1f5a8b8819
Instance state running
Instance type t2.micro
Elastic IPs 50.18.192.135*
Public DNS (IPv4) ec2-50-18-192-135.us-west-1.compute.amazonaws.com
IPv4 Public IP 50.18.192.135
IPv6 IPs -
Private DNS ip-172-31-11-121.us-west-1.compute.internal

Figure 211.11: EC2 Running Instances.

You can login to your instances either by `ssh` or by using X2GO client 211.4. Please note that every time when you restart the instances, the public IP address will change.

5. Fix Public IP Address (Optional)

   The public IP address of Real-ESSI instances change for each reboot. If users want to have a fixed public IP address for every login, users can allocate one elastic IP address and associate the IP address to a Real-ESSI instance such that users can have a fixed public IP address for each login.

6. Attach more Storage (Optional)

   The Real-ESSI Image holds 30GB Hard disk and already uses 15GB. In the case of a real large simulations, this size hard drive might not be enough for the full output. Users can attach more storage through elastic block store.

211.3.2 Launch Real-ESSI Instance from AWS Market Place

This section gives a quick start guide for using Real-ESSI on AWS market place.

Real-ESSI Simulator system (pre processing, main Real-ESSI program, post processing) is available on Amazon Web Services MarketPlace. Point your web browser to the Amazon Web Services Market Place, and search for "Real ESSI", "Real-ESSI" or "MS ESSI".

In summary, a quick guide to launching an instance from AWS Market Place is:

- Go to the ESSI Cloud Product Page.
- Click Continue to go to Launch ESSI from the Cloud.
- Click Manual Launch (use 1-Click Launch, if comfortable with settings).
- Click Launch from the EC2 Console for your preferred region.
- Select your preferred instance from the table, e.g. t2.micro.
- Click Review and Launch.

211.4 Connect to Real-ESSI Instance on AWS

211.4.1 Install X2GO Client

Before connecting to the Real-ESSI cloud, users should install the client-side of X2GO. X2Go is a remote desktop software that can visualize the launched Real-ESSI instance. Installation of X2GO for different operating systems is fairly straightforward, and users can find installation instructions on their own or follow installation instructions below.

211.4.1.1 Installing X2GO client on Ubuntu Linux

User can directly install X2GO client by using debian install utility, to install x2goclient.

211.4.1.2 Installing X2GO client on Apple Mac

Users can download the package through this link: `http://code.x2go.org/releases/X2GoClient_latest_macosx_10_9.dmg`.

211.4.1.3 Installing X2GO client on Windows

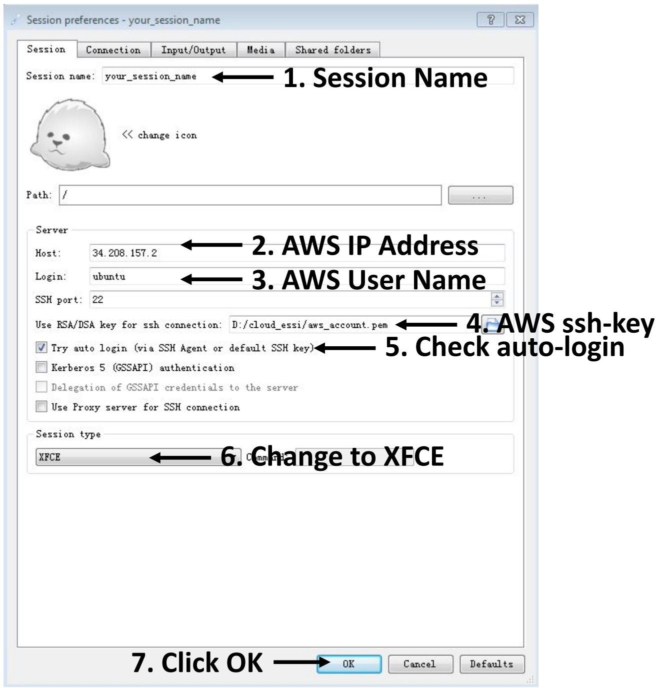
Users can download the package through this link: `http://code.x2go.org/releases/X2GoClient_latest_mswin32-setup.exe`.

211.4.1.4 Installing X2GO client on other operating systems

If you are using a different operating system, please refer to X2GO website for the installation. The X2GO website for client installation is `https://wiki.x2go.org/doku.php/download:start`

211.4.2 Configure the Client-Side of X2GO

For all operating systems, users will see the same session when they open the x2goclient new-session, as shown in Fig. 211.12.



Session name: your_session_name

1. Session Name

Path: /

Server

Host: 34.208.157.2

2. AWS IP Address

Login: ubuntu

3. AWS User Name

SSH port: 22

Use RSA/DSA key for ssh connection: D:/cloud_essi/aws_account.pem

4. AWS ssh-key

☑ Try auto login (via SSH Agent or default SSH key)

5. Check auto-login

☐ Kerberos 5 (GSSAPI) authentication

☐ Delegation of GSSAPI credentials to the server

☐ Use Proxy server for SSH connection

Session type

XFCE

6. Change to XFCE

7. Click OK

Figure 211.12: Configuration of X2GO client.

1. Users can name their own session.
2. AWS IP address is to be copied from EC2 management console, from the description TAB of launched instance, at the bottom of the page. This is IPv4 Public IP... it goes into Host: ...
3. AWS User Name is "ubuntu".
4. AWS ssh-key is the one saved from before, in .ssh directory
5. Please check the auto-login.
6. Please change the session type to XFCE.
7. Click OK to finish the configuration.

In addition to the Desktop login, users can also use `ssh` to login the Real-ESSI Terminal.

```
chmod 400 your_ssh_key.pem
ssh -i your_ssh_key.pem ubuntu@your_AWS_public_IP_address
```

211.4.3 Connect to the Launched Instance

Click the configured session to connect to the ESSI instance. You should see a virtual desktop pop up on your local machine, as shown in Fig. 211.13. Now you have successfully connected to the Real-ESSI Simulator instance on AWS. You can now use Real-ESSI Simulator within the virtual desktop.

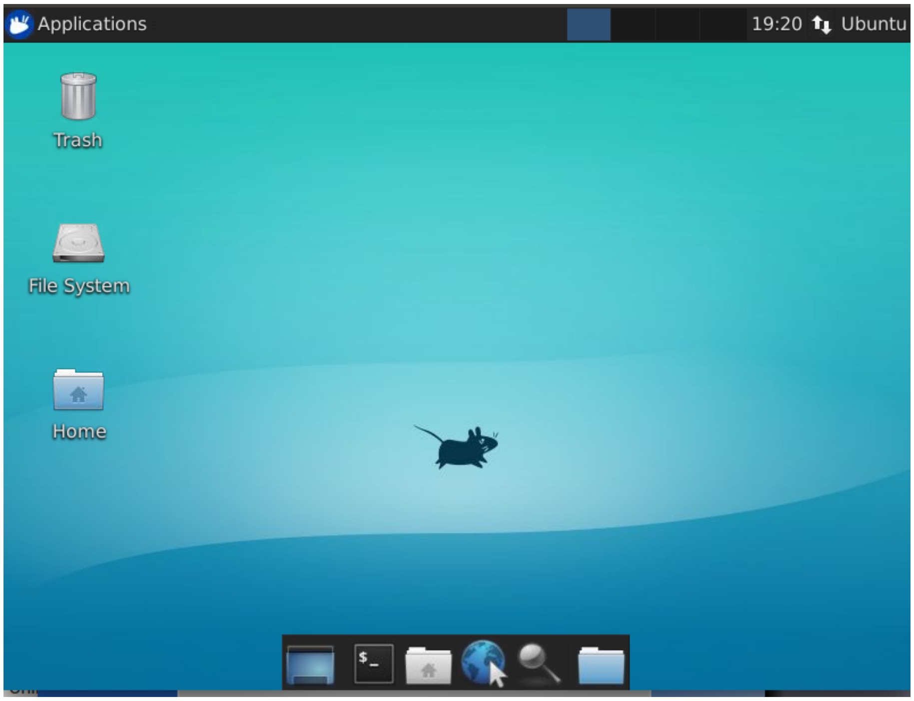


Figure 211.13: Connected to the already launched Real-ESSI instance.

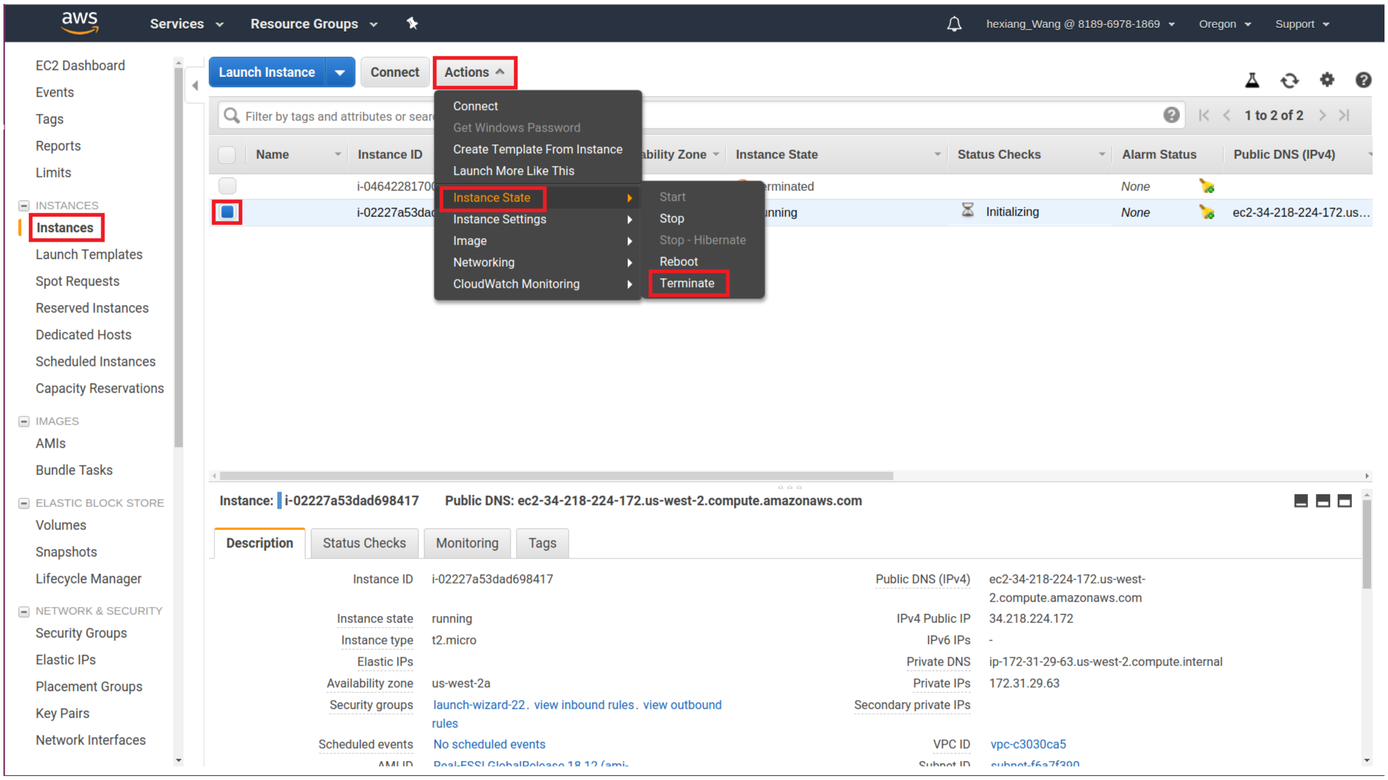
211.5 Terminate Real-ESSI Instance on AWS

Once the Real-ESSI simulation on AWS is finished, the user can transfer output files to the local local computer, or leave them on AWS, preferably on cheap S3 storage Section 211.8 on page 1394 provides detailed description of storage and transfer options and costs. NOTE: Users need to terminate the running Real -ESSI instance on AWS to avoid additional charges. The terminate operation is done on AWS console that is the same place where you launch the Real-ESSI instance. As shown Fig. 211.14, following steps are required:

1. Click 'Instances' from the sidebar to see all your running instances on AWS.
2. Choose the instance you want to terminate.
3. Click 'Actions'.

4. Click 'Instance State'

5. Click 'Terminate'



Instance ID	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
i-04642281700	rminated		None	
i-02227a53da	nning	Initializing	None	ec2-34-218-224-172.us...

Instance: i-02227a53dad698417 Public DNS: ec2-34-218-224-172.us-west-2.compute.amazonaws.com

Instance ID	i-02227a53dad698417	Public DNS (IPv4)	ec2-34-218-224-172.us-west-2.compute.amazonaws.com
Instance state	running	IPv4 Public IP	34.218.224.172
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-29-63.us-west-2.compute.internal
Availability zone	us-west-2a	Private IPs	172.31.29.63
Security groups	launch-wizard-22. view inbound rules. view outbound rules	Secondary private IPs	
Scheduled events	No scheduled events	VPC ID	vpc-c3030ca5

Figure 211.14: Terminate a Real-ESSI Simulator instance.

211.6 Adding Permission for Private Real-ESSI Image to User AWS Accounts

login to AWS

sign in to console

go to image in a region, say N, California

then go to EC2

go to AMIs on left side

select image to be shared

go to Actions

go to Modify Image Permissons and put user account number then click Add Permission and then Save...

211.7 Real-ESSI Instructional Videos Cloud Computing

This section presents few short instructional videos about how to use Real ESSI on Amazon Web Services (AWS) computers.

211.7.1 Installing X2GO for Windows

Youtube instructional video.

211.7.2 Installing X2GO for Macintosh

Youtube instructional video.

211.7.3 Installing X2GO for Linux

Youtube instructional video.

211.7.4 Launch AWS Marketplace

Youtube instructional video.

211.7.5 Access Running Instance on AWS

Youtube instructional video.

211.7.6 Start Real-ESSI Program on AWS

Youtube instructional video.

211.7.7 Run Real-ESSI Example Model on AWS

Youtube instructional video.

211.7.8 Visualize Real-ESSI Example Model on AWS

Youtube instructional video.

211.7.9 Post-Process, Visualize Real-ESSI Results on AWS

Youtube instructional video.

211.8 Cost of AWS EC2

The cost breakdown for using Real-ESSI on AWS (EC2) is:

- AWS computer cost

  There are 3 ways to pay for AWS computer cost (EC2 instances)

  - On-Demand instance, offers a real, instant pay-per-use model. On-Demand instance is sold at a fixed price, and AWS computer availability is guaranteed (within the limits of the service-level agreement). Running Real-ESSI On-Demand Instance: User prepares simulation runs, and then can simulate problems at hand immediately.
  - Spot instance, uses spare AWS computers that users can bid for. Prices for those spot instances fluctuate based on the supply and demand of available AWS computers. When a user makes a bid for a Spot instance, a spot instance is launched when the bid exceeds the current Spot market price, and continues until terminated by the user. The user is charged the Spot market price, not the bid price while the instance runs. Spot instances can offer substantial savings over On-Demand instances, as shown in the AWS Spot Bid Advisor. Running Real-ESSI using Spot instance: User can prepare simulation runs, and then bid on computer hardware and run simulations at later time, when cost is acceptable.
  - Reserved instance, uses spare AWS computers during scheduled, later time as determined by AWS and reserved by the user. Running Real-ESSI using Reserved Instance: User prepares simulation runs, and then reserves AWS computer to simulate problem at hand at predetermined/reserved time.

- AWS data storage cost

  Input data/files and output data/files are stored using:

  - Amazon Elastic Block Store (EBS), attached to a AWS computer (EC2 instance) during simulation run. Storage cost is charged by the size of storage in GB per month, pro-rated to the hour, until the storage is released. The cost of EBS is typically $\$0.10$ per GB per month. When running Real-ESSI program on AWS computer, the storage is used during simulation, while the data (input and output) is transferred out of the AWS computer, to other type of storage that is less expensive (the so called S3 storage, see below), or to user's desktop computer, before AWS computer/instance is terminated and storage released.

  - Amazon Simple Storage Service (S3), offers better value for longer term data storage. S3 pricing varies by region and frequency of access. Cost of S3 storage is typically between $\$0.0125$ are $\$0.03$ per GB per month.
  - Amazon Glacier, provides storage at an even lower cost of $\$0.007$ per GB per month for data archiving.

- AWS data transfer cost

  Data transfer charges are listed as part of the On-Demand EC2 pricing. Transfer is typically charged at $\$0.09$ per GB beyond the first 1GB of data and up to the first 1TB of a given month. After the first TB, price drops down.

- Real-ESSI program cost

  Use of Real-ESSI for educational purposes is free. For commercial use of Real-ESSI, please contact Prof. Jeremić or one of the commercial companies that offer access to Real-ESSI on AWS.

211.8.1 Cost of Running Real-ESSI on AWS

211.8.1.1 Small Size Real-ESSI Example

Imposed Motion Real-ESSI modeling and simulation on AWS summary:

- DOFs in the Model: 5,000
- Number of Time Step: 210
- Running Time: 30 Second
- Disk Space: 25 MB.
- Recommended Machine: Free Instance Amazon EC2 t2.micro

The Real-ESSI input files for this example are available HERE. The compressed package of input files is HERE.

The Modeling parameters are listed below

- Elastic Material Properties
  - Mass density, ρ, 2000 kg/m^3
  - Shear wave velocity, V_s, 500 m/s
  - Young's modulus, E, 1.1 GPa
  - Poisson's ratio, ν, 0.1

The thickness of the shell structure is 2 meters. The simulation model is shown below.

The simulation results:

The time series of simulation results is shown in Fig. 410.22.

The response spectrum of motion is shown in Fig. 410.23.

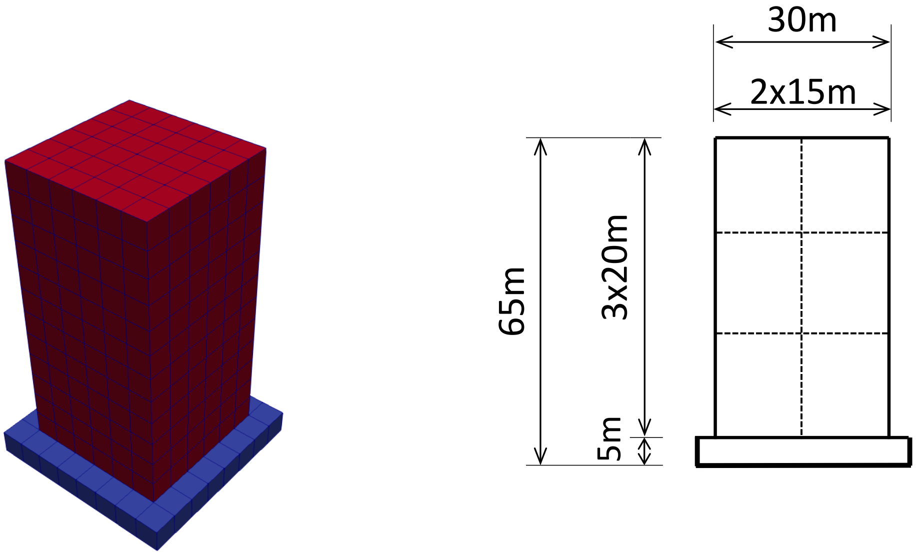


Figure 211.15: Simulation Model.

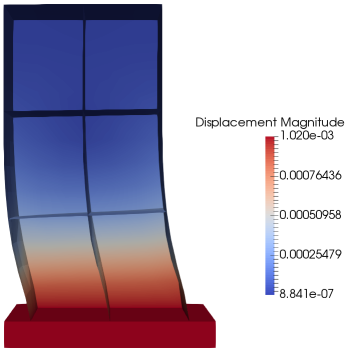


Figure 211.16: Simulation Results.

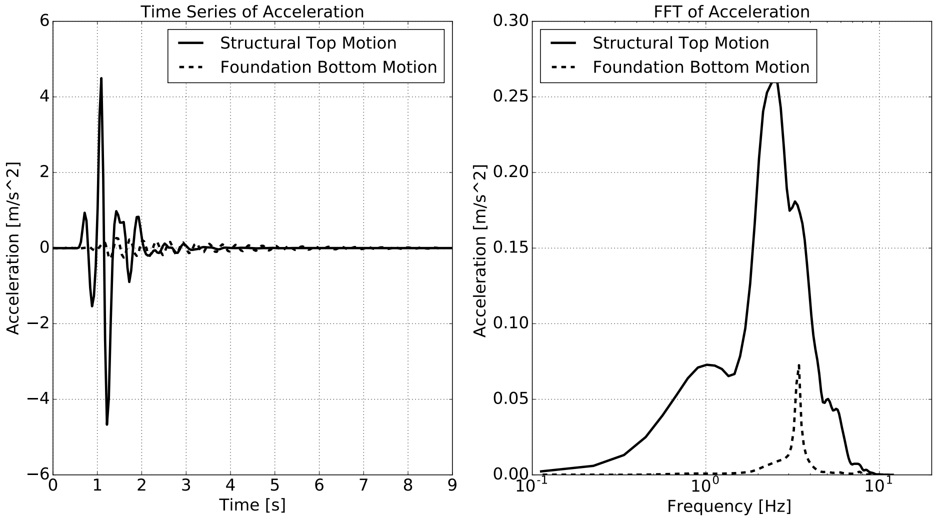


Figure 211.17: Simulation Results: Acceleration Time Series with 1C imposed motion.

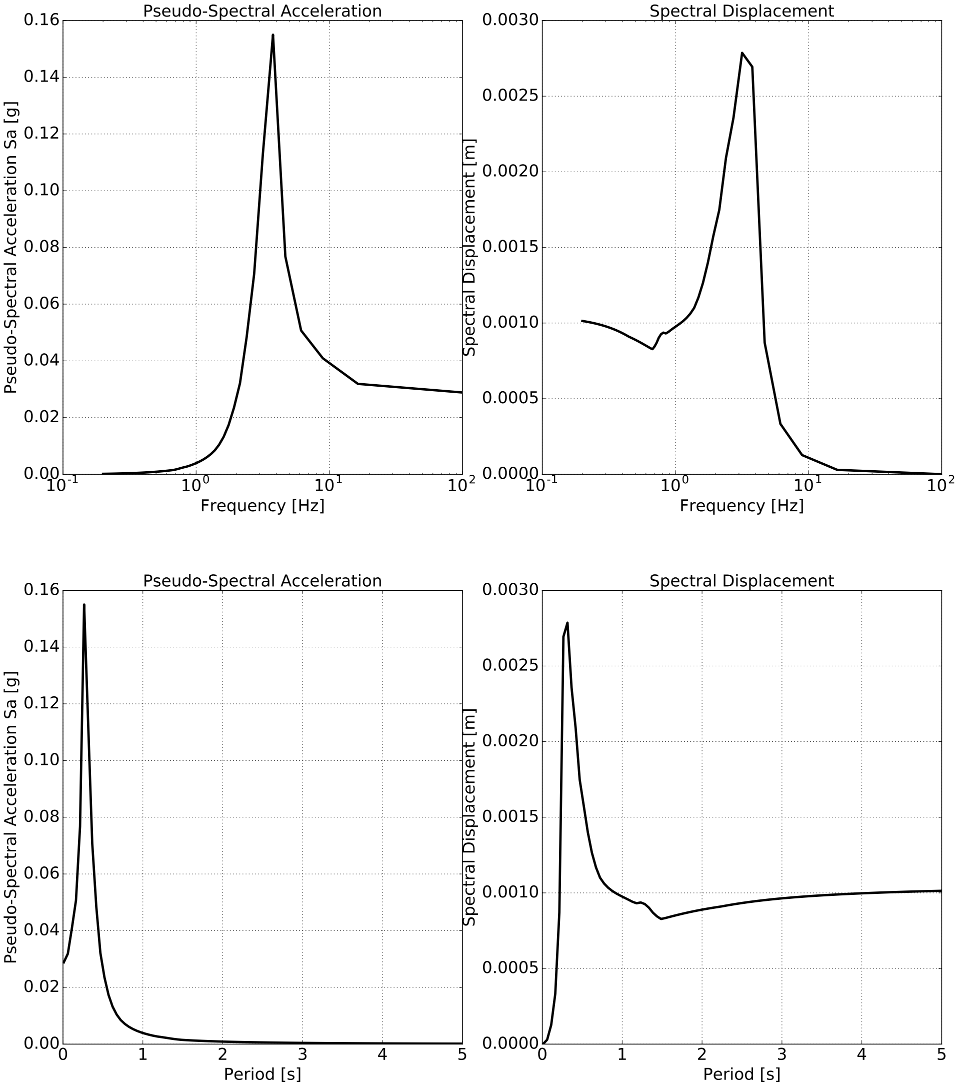


Figure 211.18: Simulation Results: Response Spectrum of Structure Top with 1C imposed motion.

Eigen Analysis Real-ESSI modeling and simulation on AWS summary:

- DOFs in the Model: 5,000
- Number of Eigenmodes: 10
- Running Time: 3 Second
- Disk Space: 25 MB.
- Recommended Machine: Free Instance Amazon EC2 t2.micro

The Real-ESSI input files for this example are available HERE. The compressed package of input files is HERE.

The thickness of the shell structure is 2 meters. The simulation model is shown below.

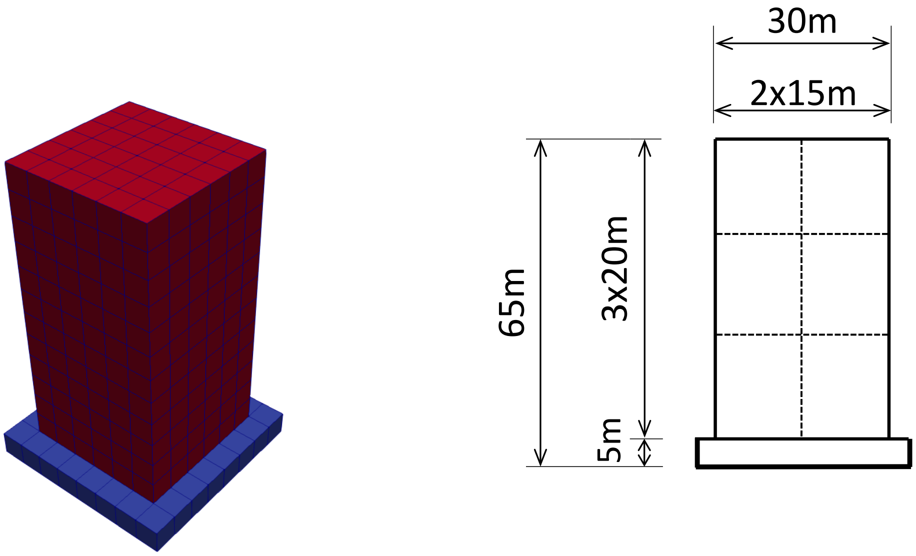


Figure 211.19: Simulation Model.

The eigen results:

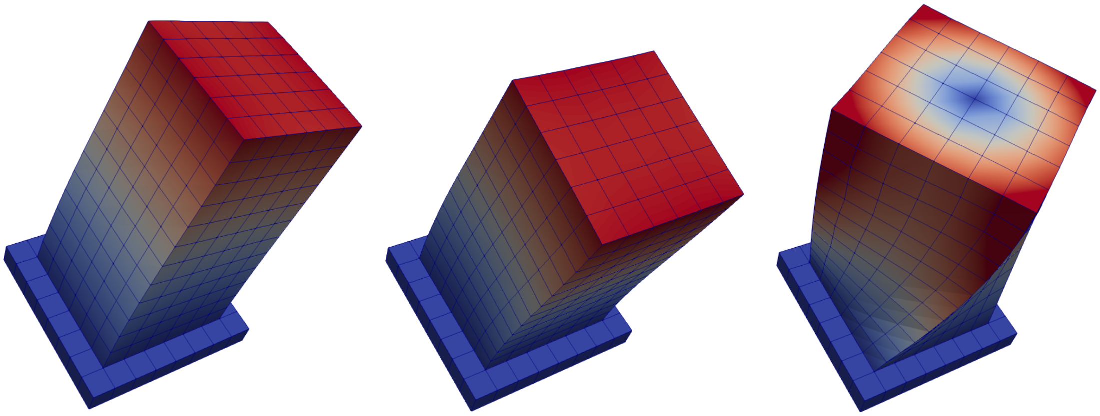


Figure 211.20: Eigen Results (Eigen Mode 1 to 3 from left to right).

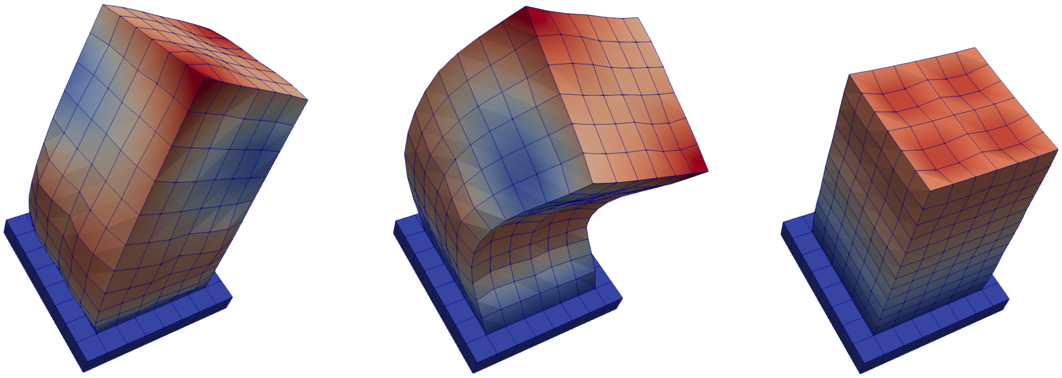


Figure 211.21: Eigen Results (Eigen Mode 4 to 6 from left to right).

211.8.1.2 Medium Size Real-ESSI Example

Elastic Material The compressed package of input files is available HERE.

Real-ESSI modeling and simulation on AWS summary:

- DOFs in the Model: 132,000
- Number of Time Steps: 210
- Running Time: 10 minutes
- Disk Space: 3GB
- Recommended Machine: Amazon EC2 c4.2xlarge instance 8 cores.
- Estimated Bill in AWS Region Oregon/Ohio/Northern Virginia:
  - For simulation time: $\$0.398 * 10/60 = \$0.07$
  - For General Purpose (SSD) Storage: $\$0.1 * 3 = \$0.3$ (monthly)
  - For S3 Storage: $\$0.023 * 3 = \$0.069$ (monthly)

The Modeling parameters are listed below

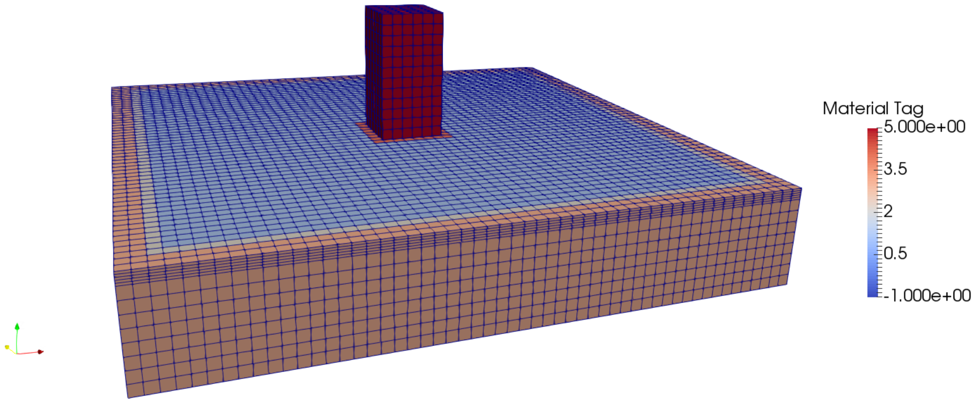
- Elastic Material Properties
  - Mass density, ρ, 2000 kg/m^3
  - Shear wave velocity, V_s, 500 m/s
  - Young's modulus, E, 1.1 GPa
  - Poisson's ratio, ν, 0.1

The illustration results of the simulation is shown in Fig. 410.12. It is noted that outside the DRM layer, there are no outgoing waves.

von-Mises Armstrong-Frederick Material The compressed package of input files is available HERE.

Real-ESSI modeling and simulation on AWS summary:

- DOFs in the Model: 132,000
- Number of Time Steps: 210
- Running Time: 46 minutes



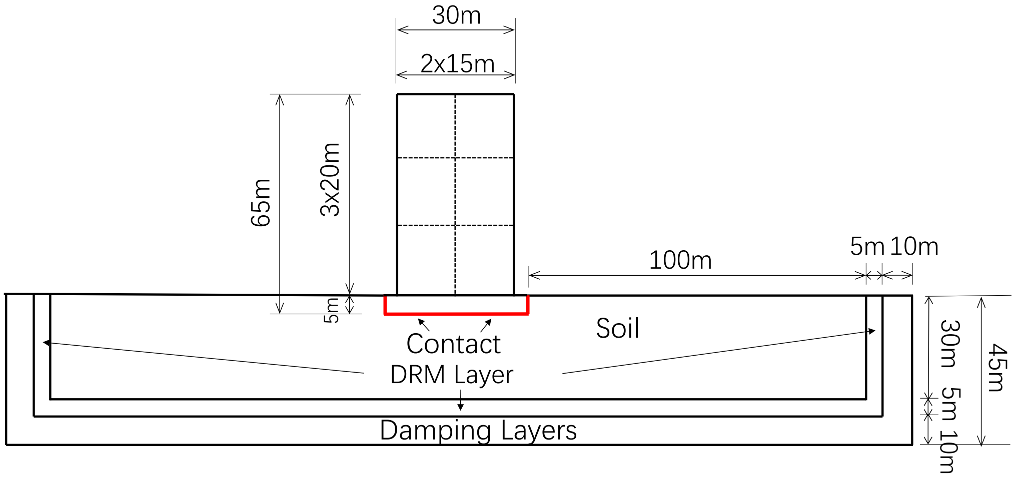


Figure 211.22: Simulation Model.

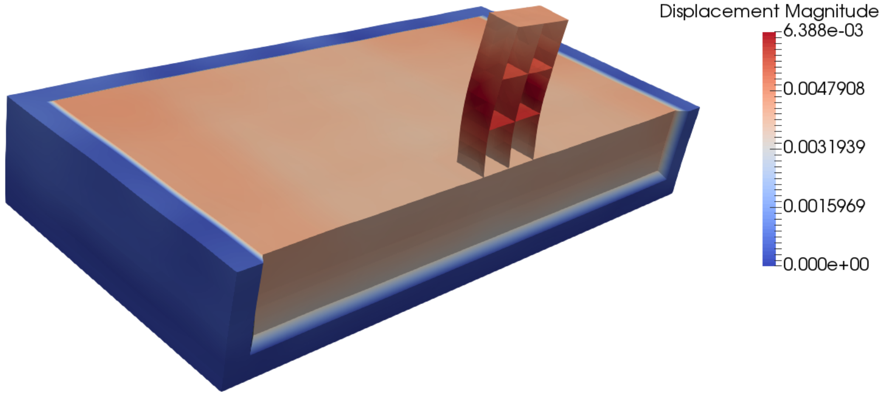


Figure 211.23: Simulation Model.

- Disk Space: 3GB
- Recommended Machine: Amazon EC2 c4.2xlarge instance 8 cores.
- Estimated Bill in AWS Region Oregon/Ohio/Northern Virginia:
  - For simulation time: $\$0.398 * 46/60 = \$0.31$
  - For General Purpose (SSD) Storage: $\$0.1 * 3 = \$0.3$ (monthly)
  - For S3 Storage: $\$0.023 * 3 = \$0.069$ (monthly)

The Modeling parameters are listed below

- von-Mises nonlinear hardening material model
  - Mass density, ρ, 2000 kg/m^3
  - Shear wave velocity, V_s, 500 m/s
  - Young's modulus, E, 1.1 GPa
  - Poisson's ratio, ν, 0.1
  - von Mises radius, k, 60 kPa
  - Nonlinear kinematic hardening, H_a, 30 MPa
  - Nonlinear kinematic hardening, C_r, 60
  - Shear strength ($\approx \sqrt{2/3}\; H_a/C_r$), S_u, 408 kPa
  - Isotropic hardening rate, K_{iso}, 0 Pa

SIMULATION TIME: With 8 cores on AWS EC2 c4.2xlarge instance, the running time for this example is 46 minutes.

211.8.1.3 Large Example

Elastic Simulation The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

Real-ESSI modeling and simulation on AWS summary:

- DOFs in the Model: 210,000
- Number of Time Steps: 2065
- Running Time: 17 hours
- Disk Space: 45GB
- Recommended Machine: Amazon EC2 c4.8xlarge instance 36 cores.
- Estimated Bill in AWS Region Oregon/Ohio/Northern Virginia:
  - For simulation time: $\$1.591 * 17 = \$27.05$
  - For General Purpose (SSD) Storage: $\$0.1 * 45 = \$4.5$ (monthly)
  - For S3 Storage: $\$0.023 * 45 = \$1.035$ (monthly)
  - For Network Bandwidth if transfer: $\$0.09 * 45 = \$4.05$

SIMULATION TIME: With 32 cores on AWS EC2 c4.8xlarge instance, the running time for this example is 17 hours.

The Modeling parameters are listed below

- Soil
  - Unit weight, γ, 21.4 kPa
  - Shear velocity, Vs, 500 m/s
  - Young's modulus, E, 1.3 GPa
  - Poisson's ratio, ν, 0.25
  - Shear strength, S_u, 650 kPa
  - von Mises radius, k, 60 kPa
  - kinematic hardening, H_a, 30 MPa
  - kinematic hardening, C_r, 25
- Structure

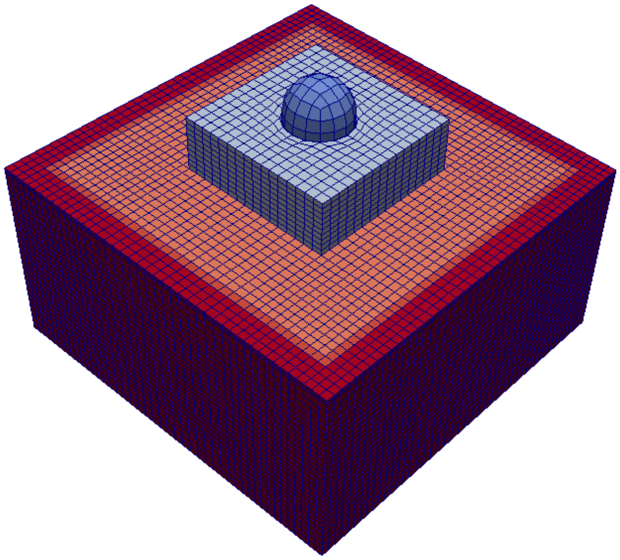


Figure 211.24: Simulation Model.

- Unit weight, γ, 24 kPa
- Young's modulus, E, 20 GPa
- Poisson's ratio, ν, 0.21

The input motion is a 3C wave from SW4.

Inelastic Simulation The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

Real-ESSI modeling and simulation on AWS summary:

- DOFs in the Model: 210,000
- Number of Time Steps: 2065
- Running Time: 30 hours
- Disk Space: 45GB
- Recommended Machine: Amazon EC2 c4.8xlarge instance 36 cores.
- Estimated Bill in AWS Region Oregon/Ohio/Northern Virginia:
  - For simulation time: $\$1.591 * 30 = \$47.73$
  - For General Purpose (SSD) Storage: $\$0.1 * 45 = \$4.5$ (monthly)
  - For S3 Storage: $\$0.023 * 45 = \$1.035$ (monthly)
  - For Network Bandwidth if transfer: $\$0.09 * 45 = \$4.05$

SIMULATION TIME: With 32 cores on AWS EC2 c4.8xlarge instance, the running time for this example is 30 hours.

The Modeling parameters are listed below

- Soil
  - Unit weight, γ, 21.4 kPa
  - Shear velocity, Vs, 500 m/s
  - Young's modulus, E, 1.3 GPa
  - Poisson's ratio, ν, 0.25
  - Shear strength, S_u, 650 kPa
  - von Mises radius, k, 60 kPa
  - kinematic hardening, H_a, 30 MPa
  - kinematic hardening, C_r, 25
- Structure

  - Unit weight, γ, 24 kPa
  - Young's modulus, E, 20 GPa
  - Poisson's ratio, ν, 0.21
- Contact
  - Initial axial stiffness, k_n^{init}, 1e9 N/m
  - Stiffening rate, S_r, 1000 /m
  - Maximum axial stiffness, k_n^{max}, 1e12 N/m
  - Shear stiffness, k_t, 1e7 N/m
  - Axial viscous damping, C_n, 100 $N \cdot s/m$
  - Shear viscous damping, C_t, 100 $N \cdot s/m$
  - Friction ratio, μ, 0.25

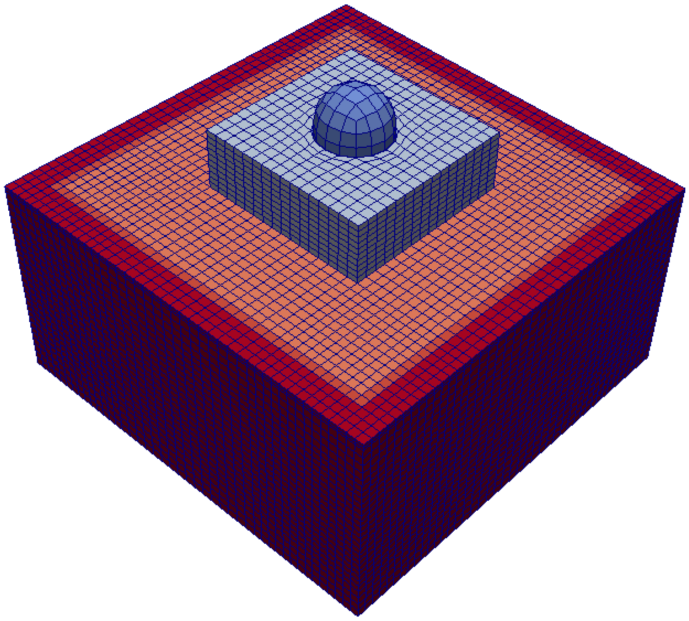


Figure 211.25: Simulation Model.

211.8.2 Real-ESSI AWS Manual, April 2023

Real-ESSI AWS manual developed for the Real-ESSI Short Course, in March, April 2023, is provided below.

1. Summary and Highlights

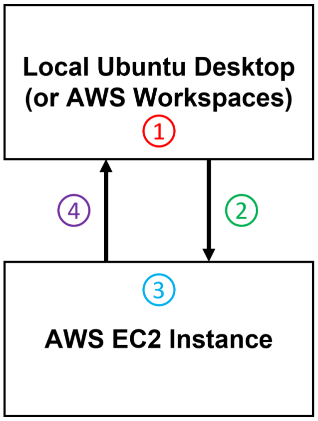
This chapter describes details of accessing and using Real-ESSI Simulator using remote computers, the so-called "cloud" computational resources. The current focus is on using Amazon Web Services (AWS) computers.

Note: If you have a local Ubuntu desktop, you may download and install the Debian package for Real-ESSI. The procedures are documented in Real-ESSI Simulator System Procurement Procedures.

2. Real-ESSI Cloud Computing Overview

Cloud computing refers to the accessing and computing over the Internet rather than on local computers. Cloud computing is a model for enabling on-demand access to a shared pool of configurable computing resources, which can be setup and released rapidly.[1]

Using Real-ESSI Cloud Service, users can get computing instances on demand without requiring a lot of maintenance and financial resources a common, local parallel computer cluster would require. In addition, users do not need to go through the installation of the dependent libraries, source-code compilation and the installation of other related software, for example preprocessing and post-processing environments. The complete Real-ESSI Simulator system is pre-configured and built within the image such that the Real-ESSI Simulator system is portable over the cloud. A stable, release version of Real-ESSI is built and can be used anywhere and anytime.



① Prepare input files on local desktop

② Transfer input files to AWS instance

③ Run the simulation on AWS instance

④ Transfer output files to local desktop

The suggested workflow for Real-ESSI cloud computing using AWS is shown above. The recommended workflow is a client-server style workflow, the most efficient and economical way to perform cloud computing. First, a local Ubuntu desktop (*AWS Workspaces is a good*

---

[1] This is an excerpt from Jeremic et al. (1989-2023)

substitute if you don't have a local Ubuntu desktop) is used to prepare input files, receive output files, and post-processing simulation results. Then, an AWS EC2 instance is used to conduct high-performance parallel computation from the Real-ESSI simulation. We will introduce detailed procedures in the following chapters.

3. Create AWS Account

3.1 AWS account types

There are two types of accounts on AWS, the Root user and the IAM user, each with unique credentials.

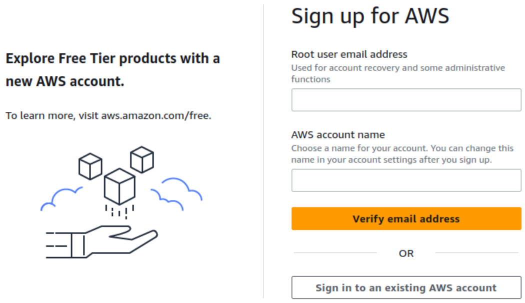
- The root user in Amazon Web Services (AWS) is the initial administrative user created when creating an AWS account. The root user has full access to all AWS services, financials, and resources in the account and can perform any action on them. **It is highly recommended to avoid using the root user account for regular day-to-day operations in AWS due to security reasons.**
- On the other hand, an IAM (Identity and Access Management) user is a user account that is created within your AWS account, separate from the root account. IAM users have a set of permissions that are defined by an AWS administrator (or yourself) to limit what actions they can perform in AWS. This allows you to grant specific permissions to users or groups of users without giving them full access to the AWS account.

IAM users can be created with unique usernames and credentials, and their permissions can be managed separately from the root user. This provides better security and allows you to grant different levels of access to different users or groups, based on their roles and responsibilities.

If you're a first-time user of AWS, your first step is to sign up for a Root user AWS account. When you sign up, AWS creates an AWS account with the details you provide and assigns the account to you. We also suggest activating multi-factor authentication (MFA) for the root user and assigning administrative access to a user. You can find complete documentation on AWS Account Management here.

3.2 Create a root user account

- To create your AWS account, open the AWS home page in your browser and choose **Create an AWS account**.
- Supply your email, then the code sent to your email address.



Explore Free Tier products with a new AWS account.

To learn more, visit aws.amazon.com/free.

Sign up for AWS

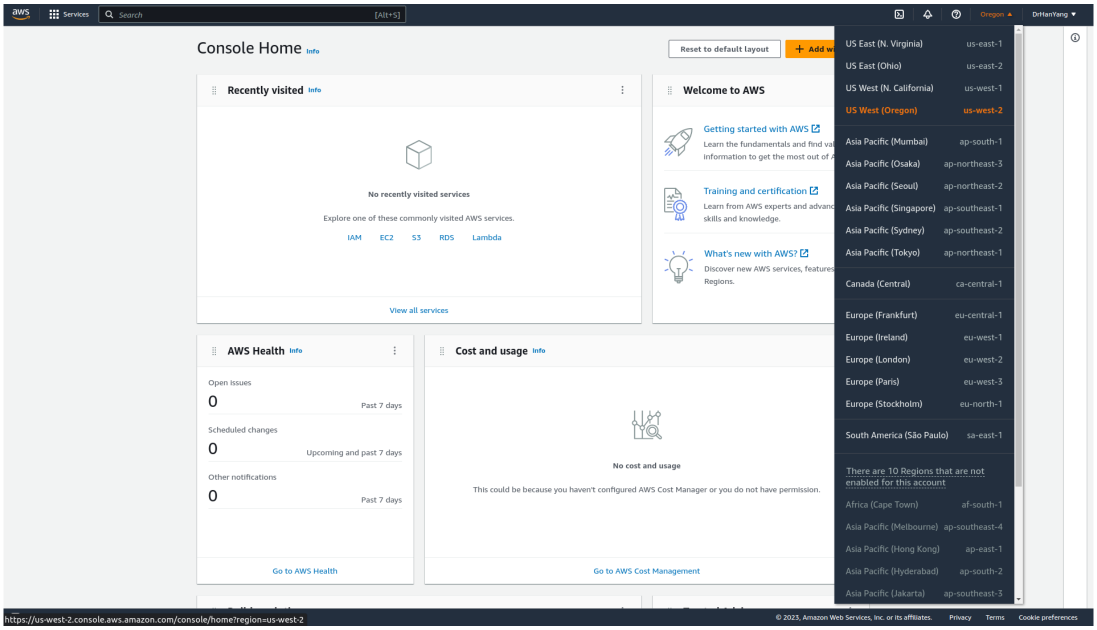
Root user email address
Used for account recovery and some administrative functions

AWS account name
Choose a name for your account. You can change this name in your account settings after you sign up.

- Create a root user password.
- Provide contact information.
- Provide billing information.
- Confirm your identity.
- Select a support plan. The **Basic support - Free** option is enough for using Real-ESSI.
- Select **Complete sign up**

3.3 Initial setups for your new AWS account

- Sign **into** your root account.
- Switch your region to **US West (Oregon)**. Your region is located near the top right corner of your webpage.



No recently visited services

Explore one of these commonly visited AWS services.

Learn the fundamentals and find val information to get the most out of A

Learn from AWS experts and advanc skills and knowledge.

Discover new AWS services, features Regions.

Open issues 0 Past 7 days

Scheduled changes 0 Upcoming and past 7 days

Other notifications 0 Past 7 days

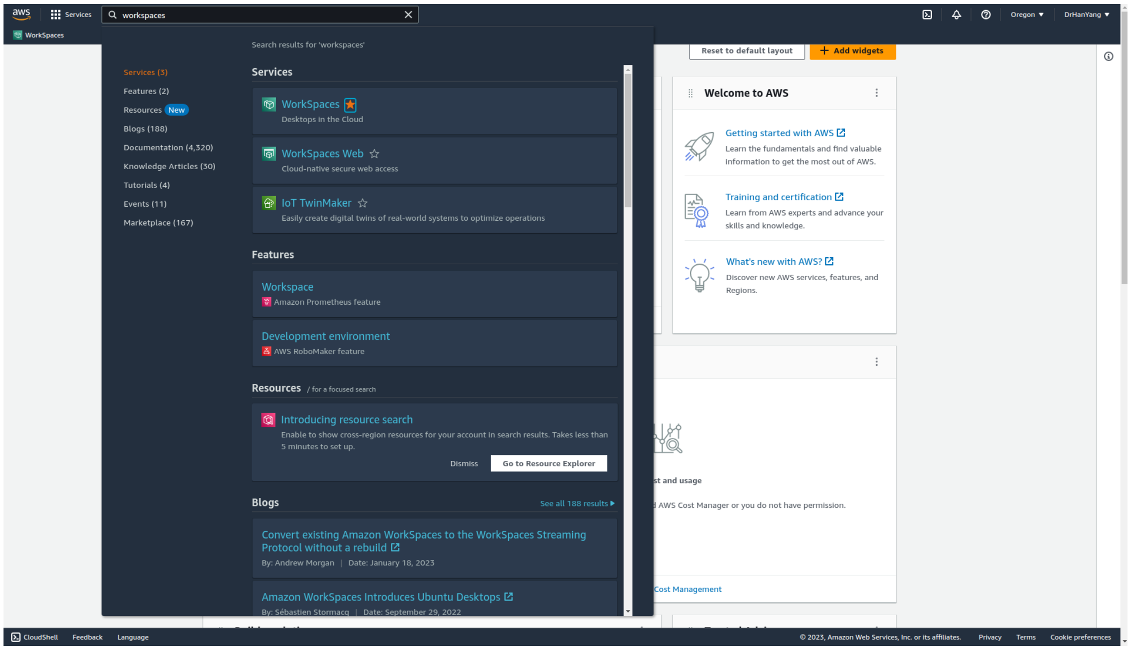
No cost and usage

This could be because you haven't configured AWS Cost Manager or you do not have permission.

https://us-west-2.console.aws.amazon.com/console/home?region=us-west-2

© 2023, Amazon Web Services, Inc. or its affiliates.

- Search and select services for quick access later. In this manual, we will use several different AWS services. We can find them and add them to favorites. Go to the search bar near the top left of your webpage and search for the services we need. Click on the star symbol next to the service name to add it to your favorites. The service should appear near the top of your webpage for future quick access.



Easily create digital twins of real-world systems to optimize operations

Enable to show cross-region resources for your account in search results. Takes less than 5 minutes to set up.

Convert existing Amazon WorkSpaces to the WorkSpaces Streaming Protocol without a rebuild
By: Andrew Morgan | Date: January 18, 2023

Amazon WorkSpaces Introduces Ubuntu Desktops
By: Sébastien Stormacq | Date: September 29, 2022

Learn the fundamentals and find valuable information to get the most out of AWS.

Learn from AWS experts and advance your skills and knowledge.

Discover new AWS services, features, and Regions.

The services we need are **Workspaces** and **EC2**.

4. Amazon Workspaces

If you already have and want to use your own local Ubuntu desktop, skip this chapter.

Summary: This chapter presents setting up, launching, and connecting to your Amazon Workspace. Note that Amazon Workspace is used as a substitute for local Ubuntu desktops.

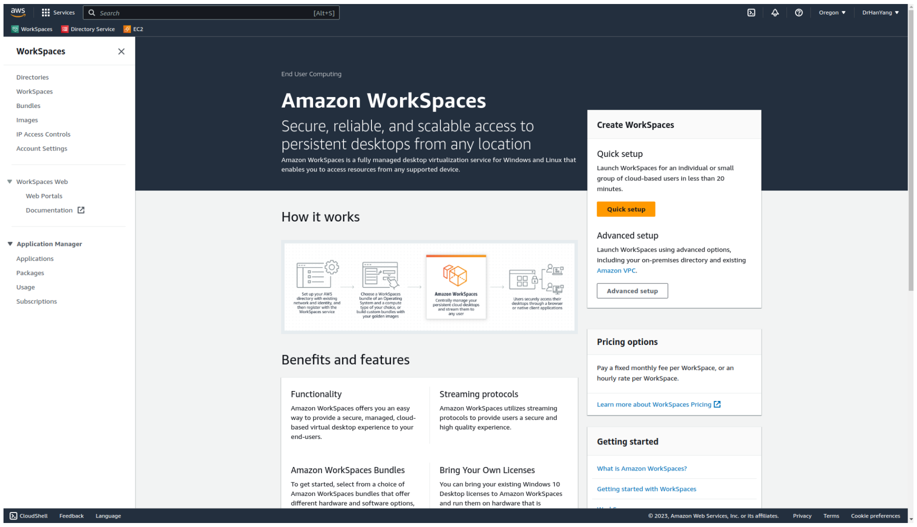
Amazon WorkSpaces enables you to provision virtual, cloud-based Microsoft Windows, Amazon Linux, or Ubuntu Linux desktops, known as *WorkSpaces*. WorkSpaces eliminates the need to purchase and set up your own Ubuntu desktop. Instead, users can access their virtual desktops from multiple devices or web browsers. Complete documentation regarding Amazon Workspaces can be found here.

4.1 WorkSpaces quick setup

This tutorial uses the **Quick Setup** option to launch your WorkSpace. This option is available only if you have never launched a WorkSpace. Alternatively, for the full documentation see here.

Step 1: Launch the WorkSpace

- Open the WorkSpaces console at https://console.aws.amazon.com/workspaces/.
- Choose **Quick setup**. If you don't see this button, either you have already launched a WorkSpace in this Region, or you aren't using one of the Regions that support Quick Setup. In this case, see Launch a virtual desktop using WorkSpaces.



End User Computing

Amazon WorkSpaces

Secure, reliable, and scalable access to persistent desktops from any location

Amazon WorkSpaces is a fully managed desktop virtualization service for Windows and Linux that enables you to access resources from any supported device.

How it works

Benefits and features

Functionality

Amazon WorkSpaces offers you an easy way to provide a secure, managed, cloud-based virtual desktop experience to your end-users.

Streaming protocols

Amazon WorkSpaces utilizes streaming protocols to provide users a secure and high quality experience.

Amazon WorkSpaces Bundles

To get started, select from a choice of Amazon WorkSpaces bundles that offer different hardware and software options,

Bring Your Own Licenses

You can bring your existing Windows 10 Desktop licenses to Amazon WorkSpaces and run them on hardware that is

Create WorkSpaces

Quick setup

Launch WorkSpaces for an individual or small group of cloud-based users in less than 20 minutes.

Advanced setup

Launch WorkSpaces using advanced options, including your on-premises directory and existing Amazon VPC.

Pricing options

Pay a fixed monthly fee per WorkSpace, or an hourly rate per WorkSpace.

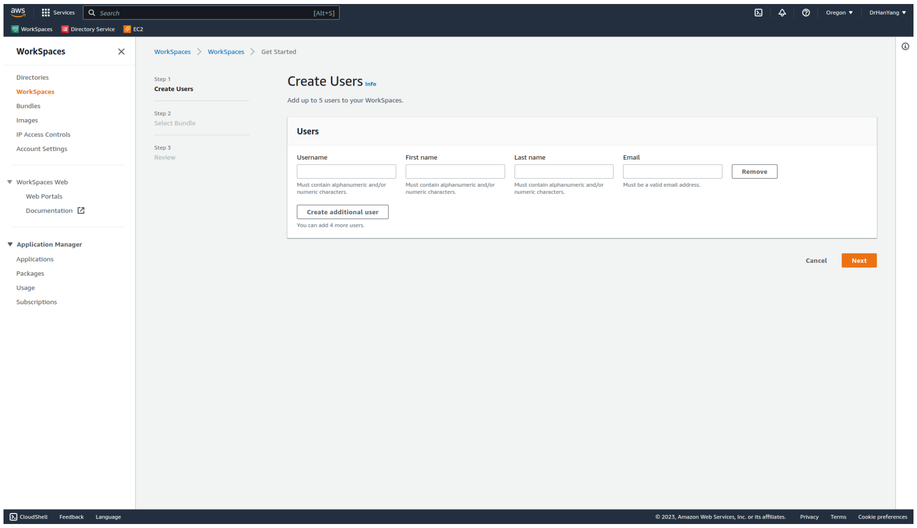
Learn more about WorkSpaces Pricing

Getting started

What is Amazon WorkSpaces?

Getting started with WorkSpaces

- For **Create Users**, enter the **Username**, **First Name**, **Last Name**, and **Email**. Then choose **Next**. Note that you can enter multiple users here, but this doesn't mean they can use the same Workspace. Instead, multiple Workspaces will be created, one for each user.



Step 1
Create Users

Step 2
Select Bundle

Step 3
Review

Create Users Info

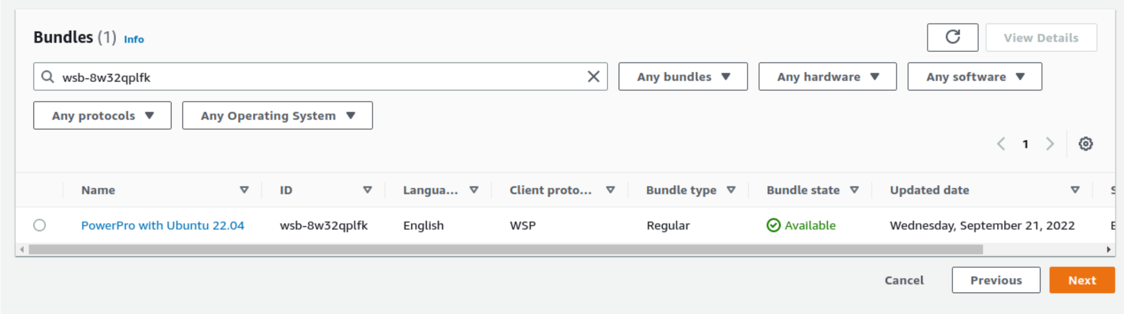
Add up to 5 users to your WorkSpaces.

Users

Username	First name	Last name	Email	
				Remove
Must contain alphanumeric and/or numeric characters.	Must contain alphanumeric and/or numeric characters.	Must contain alphanumeric and/or numeric characters.	Must be a valid email address.	

You can add 4 more users.

- For **Bundles**, select a bundle (hardware and software) for the user with the appropriate protocol (PCoIP or WSP). For Real-ESSI cloud computing, choose PowerPro with Ubuntu 22.04 with the WSP protocol. You can find this bundle by its ID wsb-8w32qplfk.



	Name	ID	Langua...	Client proto...	Bundle type	Bundle state	Updated date
○	PowerPro with Ubuntu 22.04	wsb-8w32qplfk	English	WSP	Regular	✓ Available	Wednesday, September 21, 2022

- Review your **information**. Then choose **Create WorkSpace**.

- It takes approximately 20 minutes, up to 40 minutes, for your WorkSpace to be created. When the launch is complete, the status is AVAILABLE and an invitation is sent to the email address that you specified for each user. If the users don't receive their invitation emails, see [Send an invitation email](#).

Step 2: Connect to the WorkSpace

After you receive the invitation email, you can connect to the WorkSpace using the client of your choice. After you sign in, the client displays the WorkSpace desktop.

- If you haven't set up credentials for the user already, open the link in the invitation email and follow the directions. Remember the password that you specify as you will need it to connect to your WorkSpace.

- When prompted, download one of the client applications or launch **Web Access**. If you aren't prompted and you haven't installed a client application already, open https://clients.amazonworkspaces.com/ and download one of the client applications or launch **Web Access**.

- Start the client, enter the registration code from the invitation email, and choose **Register**.

- When prompted to sign in, enter the sign-in credentials, and then choose **Sign In**.

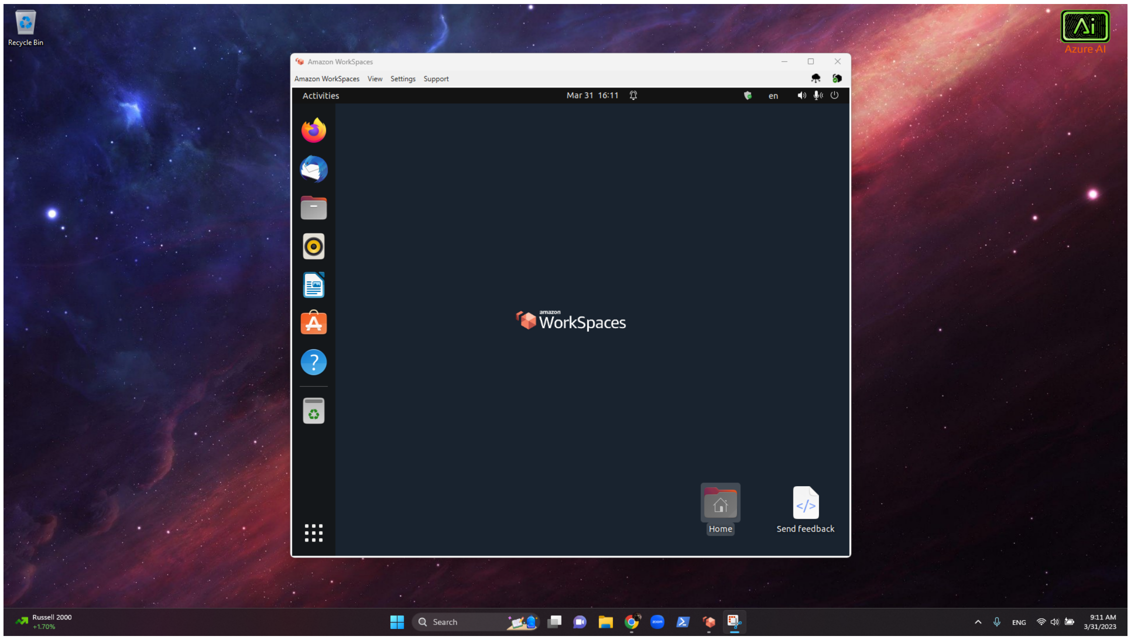
- (Optional) When prompted to save your credentials, choose Yes.

Step 3: Clean up (Optional)

If you are finished with the WorkSpace that you created for this tutorial, you can delete it. For more information, see Delete a WorkSpace.

4.2 Install Real-ESSI Debian package on your Workspace

Once you successfully connect to your Workspace, it will be the same as if you are working with a local Ubuntu desktop.



The next step is to install Real-ESSI on your Workspace. Full documentation can be found in Section 1.3 of the Real-ESSI Simulator Procurement Manual. Quick setup steps are summarized below.

Step 1: System libraries update/upgrade

Open a terminal and use the following commands.

```
sudo apt update
sudo apt upgrade
sudo apt autoremove
```

You will be asked to provide a password. The password is the same one you used to connect to your Workspace.

Step 2: Real-ESSI Debian package download

The Real-ESSI program Debian package can be downloaded here. Alternatively, contact Prof. Jeremic to arrange for a customized Real-ESSI Debian package.

Step 3: Real-ESSI Debian package install

Start the Real-ESSI Simulator Debian package install by removing the old installations of Real-ESSI. Then, go to the directory where you have downloaded the Real-ESSI Debian package. Install the Debian package, for example use the following command.

```
sudo apt install ./real-essi_23.01-1_amd64.deb
```

Note that some warning messages might appear but they don't affect the installation. After a successful installation, the sequential and parallel Real-ESSI executables, gmsh/gmESSI preprocessor, paraview/pvESSI post-processor/visualizer, Gmsh, and ParaView are all installed and ready to use.

Step 4: Load pvESSI plugin in ParaView

Start ParaView. Click **Tools**, then **Manage Plugins**. Click **Load New** and find the plugin **PVESSIReader.so** under directory /opt/paraview/lib/paraview-5.10/plugins/PVESSIReader/. Also, check the box **Auto Load** then close ParaView. Next time when ParaView is started, Real-ESSI output files can be visualized and post-processed.

Step 5: Install other useful programs

- **HDFView** can be used to open Real-ESSI output files, which are in HDF5 format.
- **Sublime Text** is the recommended editor for Real-ESSI input files and pre-processing files.

Documentation on how to install these programs can be found in Section 1.3.5 of the Real-ESSI Simulator Procurement Manual.

4.3 Build your model and prepare the input files

Once you have installed Real-ESSI on your Workspace, you are ready to start building your Real-ESSI model. You should finish preparing all the input files on your Workspace before moving on to the next Chapter. Full documentation on Real-ESSI pre-processing and input file formats is available.

Note that if the model is sufficiently small, you can simply run the simulation on your Workspace, without using the AWS Real-ESSI instance which is designed to be used in cases with large models.

5. Launch Real-ESSI Instance on AWS

A Real-ESSI instance can be launched either from the private image with authorization of Prof. Boris Jeremic or from the public image on AWS Marketplace (coming soon). Full documentation regarding launching an instance on AWS can be found here.

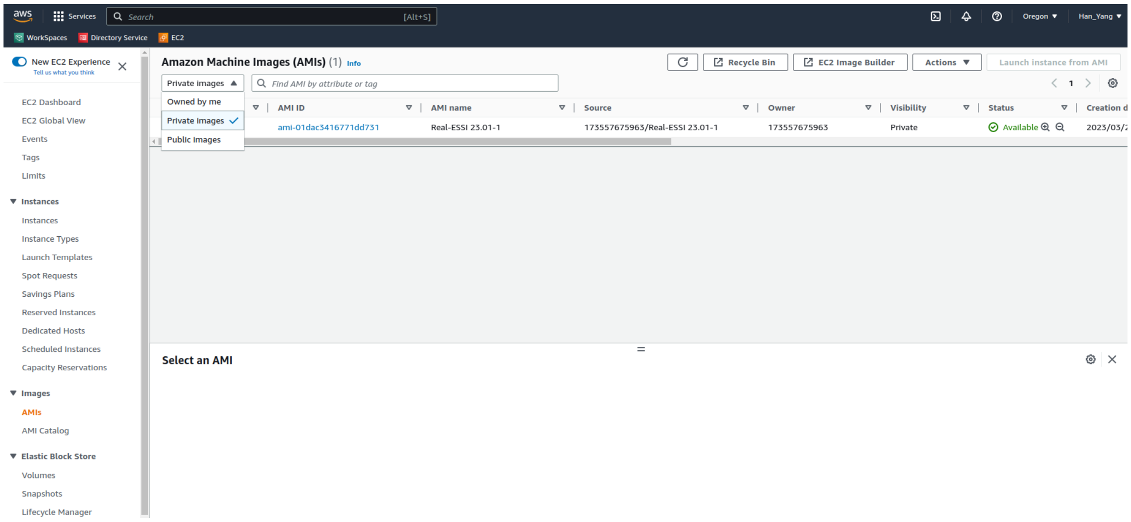
5.1 Launch Real-ESSI instance from AWS private images

Follow the steps below to launch instances from Real-ESSI Private Image.

Step 1: Request the Real-ESSI image

Real-ESSI image is currently a private Amazon Machine Images (AMI). After you get the 12-digit AWS account ID, email the AWS account ID to Prof. Boris Jeremic to obtain the Real-ESSI image.

To check if you have access to the Real-ESSI image, open the Amazon EC2 console at https://console.aws.amazon.com/ec2/. Go to **AMIs** and choose **Private images** to see the Real-ESSI image. Currently, the Real-ESSI AMIs are available in the **Oregon** region. The region is shown in the top-right corner on the EC2 dashboard.

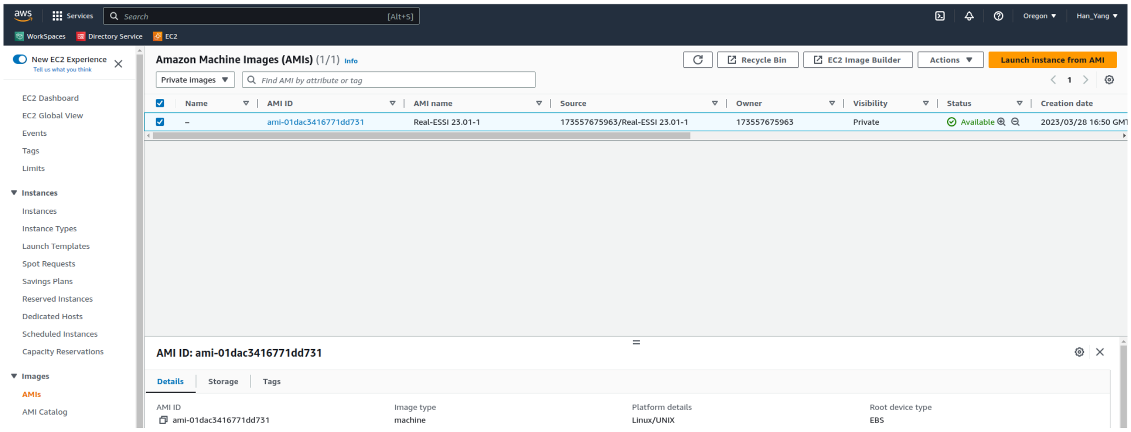


AMI ID	AMI name	Source	Owner	Visibility	Status	Creation d
ami-01dac3416771dd731	Real-ESSI 23.01-1	173557675963/Real-ESSI 23.01-1	173557675963	Private	Available	2023/03/

Step 2: Launch the Real-ESSI instance

You can launch the Real-ESSI instance using the AWS Management Console as described in the following procedure. This tutorial is intended to help you quickly launch your first instance, so it doesn't cover all possible options. For information about advanced options, see [Launch an instance using the new launch instance wizard](). For information about other ways to launch your instance, see [Launch your instance]().

- Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
- From the EC2 console dashboard, Go to **AMIs** and choose **Private images** to see the Real-ESSI image. Choose the image and choose **Launch instance from AMI**.



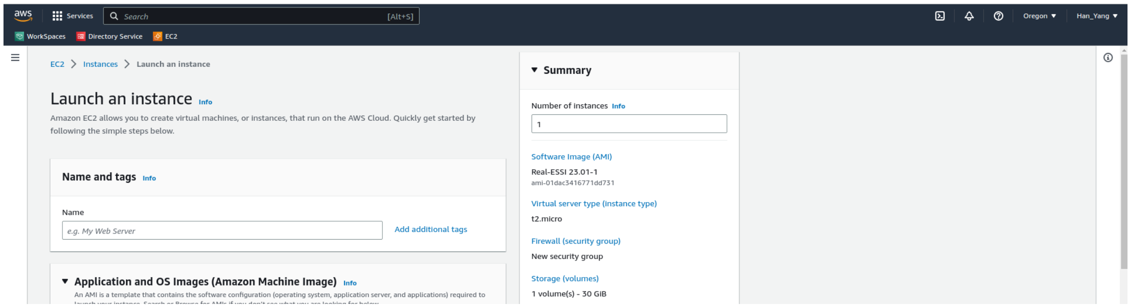
Amazon Machine Images (AMIs) (1/1) Info

☑	Name	AMI ID	AMI name	Source	Owner	Visibility	Status	Creation date
☑	–	ami-01dac3416771dd731	Real-ESSI 23.01-1	173557675963/Real-ESSI 23.01-1	173557675963	Private	Available	2023/03/28 16:50 GMT

AMI ID: ami-01dac3416771dd731

AMI ID: ami-01dac3416771dd731
Image type: machine
Platform details: Linux/UNIX
Root device type: EBS

- Under **Name and tags**, for **Name**, enter a descriptive name for your instance.



Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name: e.g. My Web Server

▼ Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to

▼ Summary

Number of instances: 1

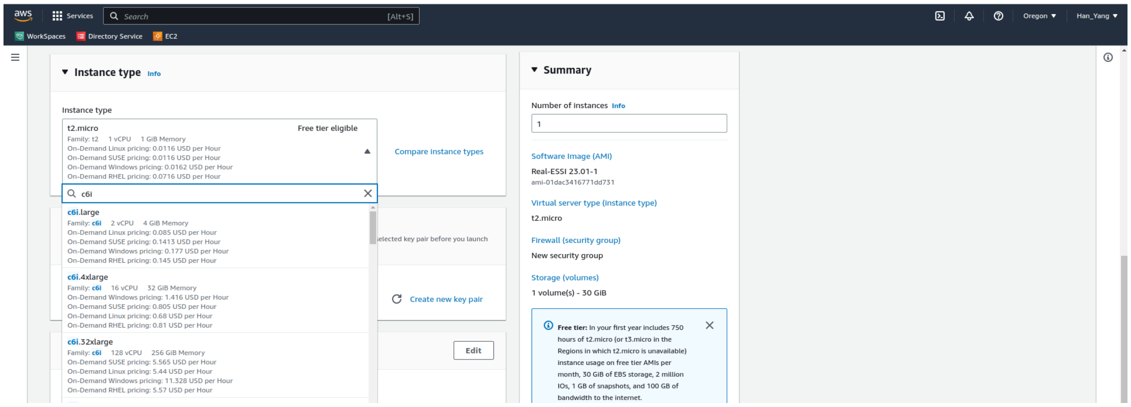
Software Image (AMI): Real-ESSI 23.01-1, ami-01dac3416771dd731

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 30 GiB

- Under **Instance type**, you can select the hardware configuration for your instance. For Real-ESSI instances, the compute-optimized **c6i** series is recommended. Click the drop-down list under **instance type** and type **c6i** in the search bar. Depending on the size of your model, you can choose an instance type with appropriate computing power.



▼ Instance type

Instance type

t2.micro Free tier eligible
Family: t2 1 vCPU 1 GiB Memory
On-Demand Linux pricing: 0.0116 USD per Hour
On-Demand SUSE pricing: 0.0116 USD per Hour
On-Demand Windows pricing: 0.0162 USD per Hour
On-Demand RHEL pricing: 0.0716 USD per Hour

c6i

c6i.large
Family: c6i 2 vCPU 4 GiB Memory
On-Demand Linux pricing: 0.085 USD per Hour
On-Demand SUSE pricing: 0.1413 USD per Hour
On-Demand Windows pricing: 0.177 USD per Hour
On-Demand RHEL pricing: 0.145 USD per Hour

c6i.4xlarge
Family: c6i 16 vCPU 32 GiB Memory
On-Demand Windows pricing: 1.416 USD per Hour
On-Demand SUSE pricing: 0.805 USD per Hour
On-Demand Linux pricing: 0.68 USD per Hour
On-Demand RHEL pricing: 0.81 USD per Hour

c6i.32xlarge
Family: c6i 128 vCPU 256 GiB Memory
On-Demand SUSE pricing: 5.565 USD per Hour
On-Demand Linux pricing: 5.44 USD per Hour
On-Demand Windows pricing: 11.328 USD per Hour
On-Demand RHEL pricing: 5.57 USD per Hour

elected key pair before you launch

▼ Summary

Number of instances: 1

Software Image (AMI): Real-ESSI 23.01-1, ami-01dac3416771dd731

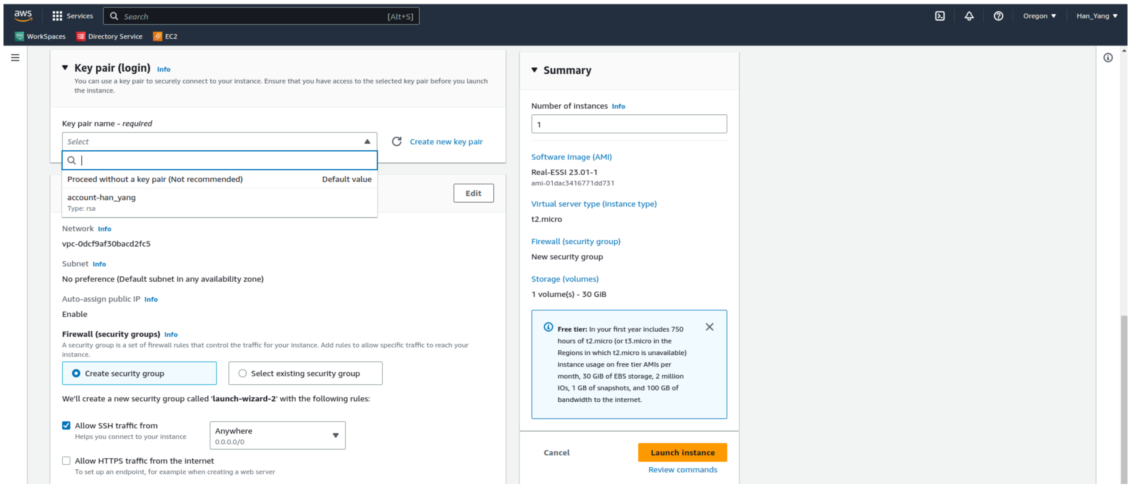
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 30 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

- Under **Key pair (login)**, select the key pair for your instance.



Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

Select

Proceed without a key pair (Not recommended) Default value

account-han_yang
Type: rsa

Network Info
vpc-0dcf9af30bacd2fc5

Subnet Info
No preference (Default subnet in any availability zone)

Auto-assign public IP Info
Enable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

We'll create a new security group called '**launch-wizard-2**' with the following rules:

☑ Allow SSH traffic from — Anywhere 0.0.0.0/0
Helps you connect to your instance

☐ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

Summary

Number of instances Info: 1

Software Image (AMI)
Real-ESSI 23.01-1
ami-01dac3416771dd731

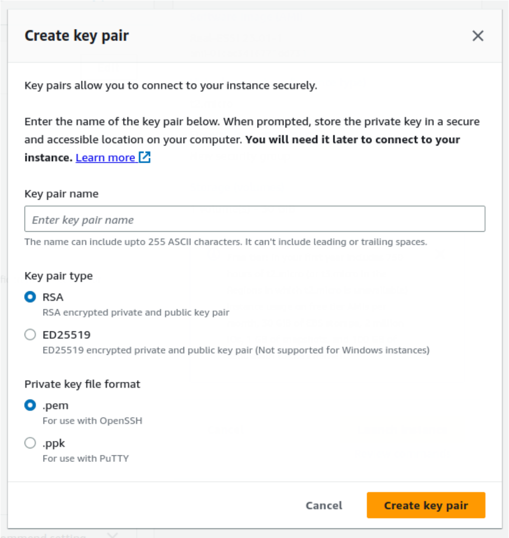
Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 30 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

If you have not created a key pair before, choose **Create new key pair**.



Create key pair

Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** Learn more

Key pair name

Enter key pair name

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

● RSA
RSA encrypted private and public key pair

○ ED25519
ED25519 encrypted private and public key pair (Not supported for Windows instances)

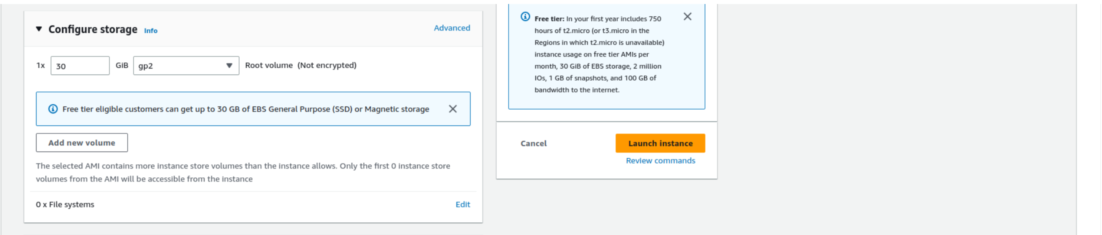
Private key file format

● .pem
For use with OpenSSH

○ .ppk
For use with PuTTY

Enter a descriptive name for your key pair. Leave everything else the same. Then click **Create key pair**. You will be prompted to save the key pair. Note that the key pair cannot be recreated after you launch the instance, so please make sure you save it in a safe place. The key pair can be reused later when you launch other instances.

- Under **Configure storage**, change the Root volume depending on the size of your model and simulation options.



1x 30 GiB gp2 Root volume (Not encrypted)

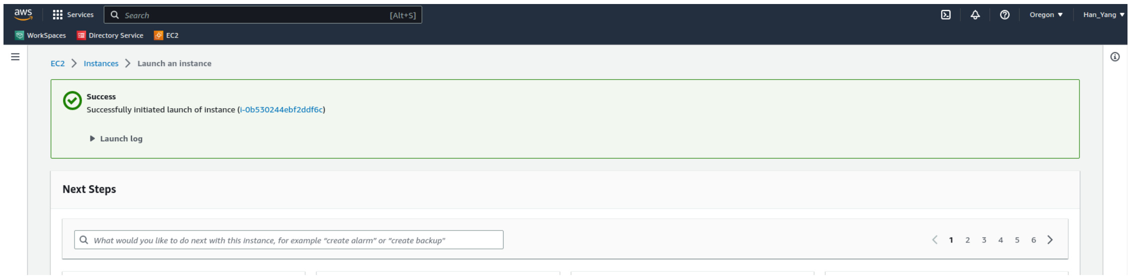
Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

0 x File systems

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

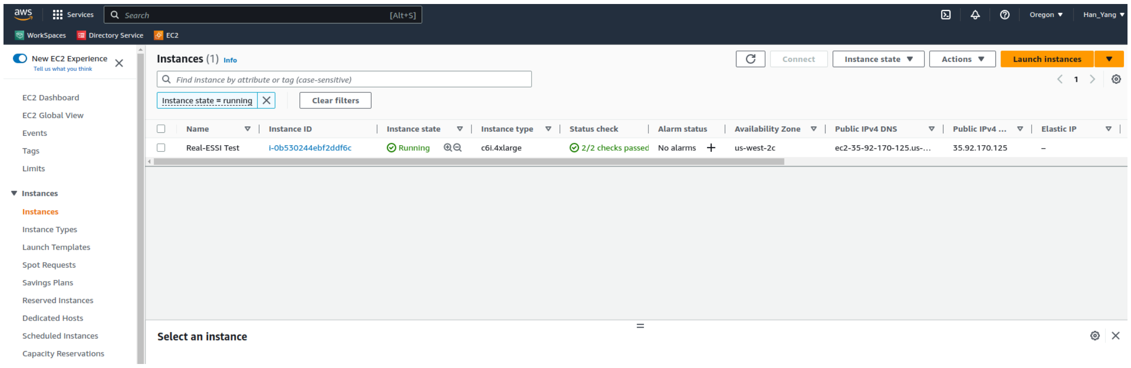
- Click **Launch instance**. You should see the message below if the launch is successful.



Success
Successfully initiated launch of instance (i-0b530244ebf2ddf6c)

What would you like to do next with this instance, for example "create alarm" or "create backup"

- You can view your running instances by clicking **Instances** on the left side list of your screen.



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
Real-ESSI Test	i-0b530244ebf2ddf6c	Running	c6i.4xlarge	2/2 checks passed	No alarms	us-west-2c	ec2-35-92-170-125.us-...	35.92.170.125	–

5.2 Launch Real-ESSI instance from AWS Marketplace

Coming soon…

6. Connect to Real-ESSI Instance

This chapter provides information about how to connect to a Real-ESSI instance after you have launched it, and how to transfer files between your local computer and your instance. For more information, please refer to the AWS documentation here. To troubleshoot connecting to your instance, see Troubleshoot connecting to your instance.

After you launch your instance, you can connect to it and use it the way that you'd use a computer sitting in front of you. The following instructions explain how to connect to your instance using an SSH client. For more connection options, see Connect to your Linux instance.

6.1 Prerequisites

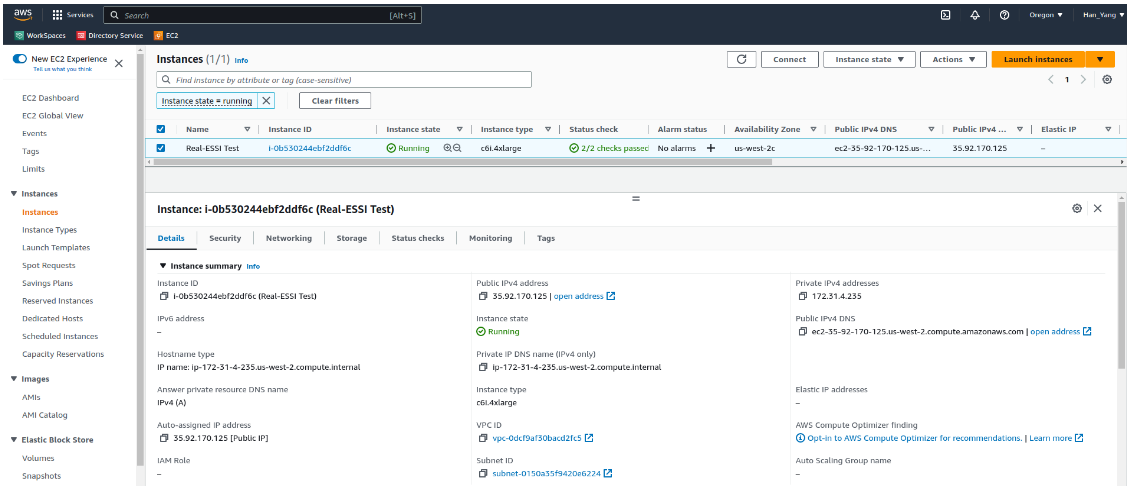
Before you connect to your Linux instance, complete the following prerequisites.

Check your instance status

After you launch an instance, it can take a few minutes for the instance to be ready so that you can connect to it. Check that your instance has passed its status checks. You can view this information in the **Status check** column on the **Instances** page.

Get the public IP address to connect to your instance

Click on your instance to show more information about it. You can find the public IPv4 address from either the summary or details. For example, the public IPv4 address is 35.92.170.125 for the instance shown below.



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
☑	Real-ESSI Test	i-0b530244ebf2ddf6c	Running	c6i.4xlarge	2/2 checks passed	No alarms	us-west-2c	ec2-35-92-170-125.us-...	35.92.170.125	–

Instance: i-0b530244ebf2ddf6c (Real-ESSI Test)

Instance summary

Instance ID i-0b530244ebf2ddf6c (Real-ESSI Test)	Public IPv4 address 35.92.170.125 | open address	Private IPv4 addresses 172.31.4.235
IPv6 address –	Instance state Running	Public IPv4 DNS ec2-35-92-170-125.us-west-2.compute.amazonaws.com | open address
Hostname type IP name: ip-172-31-4-235.us-west-2.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-4-235.us-west-2.compute.internal	
Answer private resource DNS name IPv4 (A)	Instance type c6i.4xlarge	Elastic IP addresses –
Auto-assigned IP address 35.92.170.125 [Public IP]	VPC ID vpc-0dcf9af30bacd2fc5	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role –	Subnet ID subnet-0150a35f9420e6224	Auto Scaling Group name –

Locate the private key and set the permissions

You must know the location of your private key file to connect to your instance. For SSH connections, you must set the permissions so that only you can read the file.

Get the fully-qualified path to the location on your computer of the `.pem` file for the key pair that you specified when you launched the instance.

Use the following command to set the permissions of your private key file so that only you can read it. Replace `key-pair-name` with the actual name of your key pair.

```
chmod 400 key-pair-name.pem
```

If you do not set these permissions, then you cannot connect to your instance using this key pair. For more information, see Error: Unprotected private key file.

6.2 Connect to your Real-ESSI instance using an SSH client

Use the following procedure to connect to your Linux instance using an SSH client. If you receive an error while attempting to connect to your instance, see Troubleshoot connecting to your instance.

- In a terminal window, use the **ssh** command to connect to the instance. You specify the path and file name of the private key (.pem) and the IPv4 address for your instance. To connect to your instance, use the following command.

```
ssh -i /path/key-name.pem ubuntu@IPv4-address
```

Replace /path/ with the full absolute path to your key pair. Replace key-name with the actual name of your key pair. Replace IPv4-address with the public IPv4 address of your instance.

You will see a response like the following:

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com
   (198-51-100-1)' can't be established.
   ECDSA key fingerprint is l4UB/neBad9tvkgJf1QZWxheQmR59WgrgzEimCG6kZY.
   Are you sure you want to continue connecting (yes/no)?
```

- (Optional) Verify that the fingerprint in the security alert matches the fingerprint that you previously obtained in (Optional) Get the instance fingerprint. If these fingerprints don't match, someone might be attempting a man-in-the-middle attack. If they match, continue to the next step.

- Enter `yes`.

You will see a response like the following:

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com'
   (ECDSA) to the list of known hosts.
```

- (Optional) Create a directory to organize your Real-ESSI simulation files. Replace test_folder with your folder name.

```
mkdir test_folder
```

6.3 Transfer Input Files to Real-ESSI Instance

This section describes how to transfer files with the secure copy protocol (SCP). The procedure is similar to the procedure for connecting to an instance with SSH.

- Open a terminal on your local Ubuntu desktop (or Amazon Workspace).

- Determine the file location on your local Ubuntu desktop (or Amazon Workspace) and the destination path on the instance. In the following example, the name of the private key file is key-name.pem, the file to transfer is main.fei, and the IPv4 address of the instance is IPv4-address. Enter the following command in your terminal.

```
scp -i /path1/key-name.pem /path2/main.fei ubuntu@IPv4-address:/path3/
```

  There are three paths in this command.
  ➢ Replace /path1/ with the full absolute path to your key pair on your local desktop.
  ➢ Replace /path2/ with the full absolute path to the file to transfer on your local desktop.
  ➢ Replace /path3/ with the full absolute path to the destination on the Real-ESSI instance.

  For example, your **scp** command may be:

```
scp -i /home/han/Documents/han-key.pem /home/han/My_Model/main.fei
ubuntu@52.26.2.245:/home/ubuntu/Test/
```

- If you haven't already connected to the instance using SSH, you will see a response like the following:

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com
(10.254.142.33)' can't be established.
RSA key fingerprint is
1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.
Are you sure you want to continue connecting (yes/no)?
```

  (Optional) You can optionally verify that the fingerprint in the security alert matches the instance fingerprint. For more information, see (Optional) Get the instance fingerprint.

  Enter `yes`.

- If the transfer is successful, the response is similar to the following:

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (RSA)
to the list of known hosts.
main.fei                                100%   164KB     1.3MB/s   00:00
```

- It's also possible to transfer entire folders using the **scp** command. In the following example, the name of the private key file is key-name.pem, the folder to transfer is folder, and the IPv4 address of the instance is IPv4-address. Enter the following command in your terminal.

```
scp -i /path1/key-name.pem -r /path2/folder ubuntu@IPv4-address:/path3/
```

  There are three paths in this command.

- ➢ Replace /path1/ with the full absolute path to your key pair on your local desktop.
- ➢ Replace /path2/ with the full absolute path to the folder to transfer on your local desktop.
- ➢ Replace /path3/ with the full absolute path to the destination on the Real-ESSI instance.

For example, your **scp** command may be:

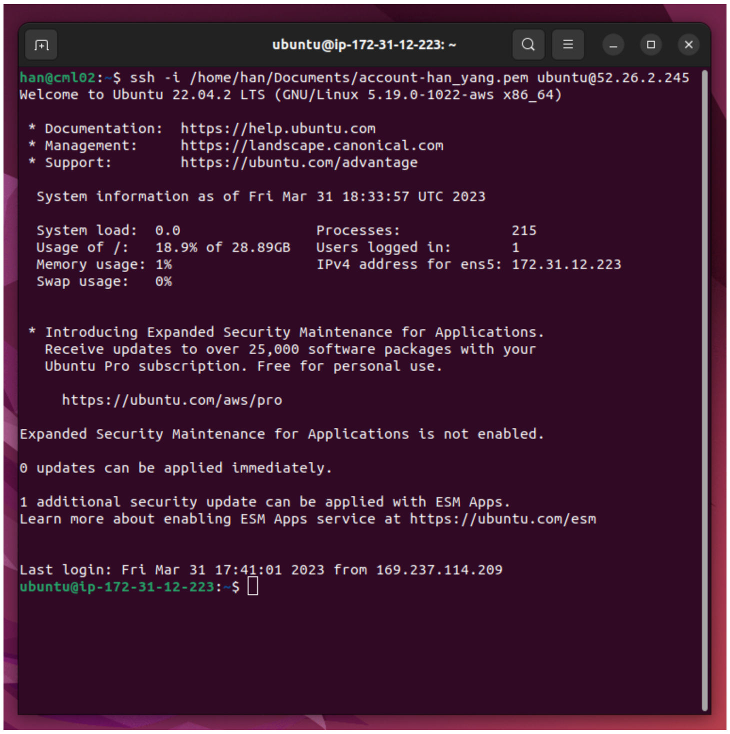
```
scp -i /home/han/Documents/han-key.pem -r /home/han/My_Model_Folder
ubuntu@52.26.2.245:/home/ubuntu/Test/
```

8. Run Simulations on Real-ESSI Instance

This chapter provides information about running your simulations on a launched Real-ESSI instance. Note that you should only attempt to do this after you have done the following:

- Create an AWS account
- Have access to a local Ubuntu desktop (or Amazon Workspace)
- Launch a Real-ESSI instance on AWS
- Connect to the Real-ESSI instance you have launched

Once you have successfully transferred your input files, open a new terminal and connect to the Real-ESSI instance. Your terminal should look like the following:



```
han@cml02:~$ ssh -i /home/han/Documents/account-han_yang.pem ubuntu@52.26.2.245
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1022-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri Mar 31 18:33:57 UTC 2023

  System load:  0.0               Processes:             215
  Usage of /:   18.9% of 28.89GB   Users logged in:       1
  Memory usage: 1%                IPv4 address for ens5: 172.31.12.223
  Swap usage:   0%


 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

     https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


Last login: Fri Mar 31 17:41:01 2023 from 169.237.114.209
ubuntu@ip-172-31-12-223:~$ 
```

- Go to the directory where you transferred your input files. Use the following command:

```
cd /path/
```

  Replace /path/ with the full absolute path to your input files on the Real-ESSI instance.

- Change the permission of your file so that it can be read and executed. Use the following command:

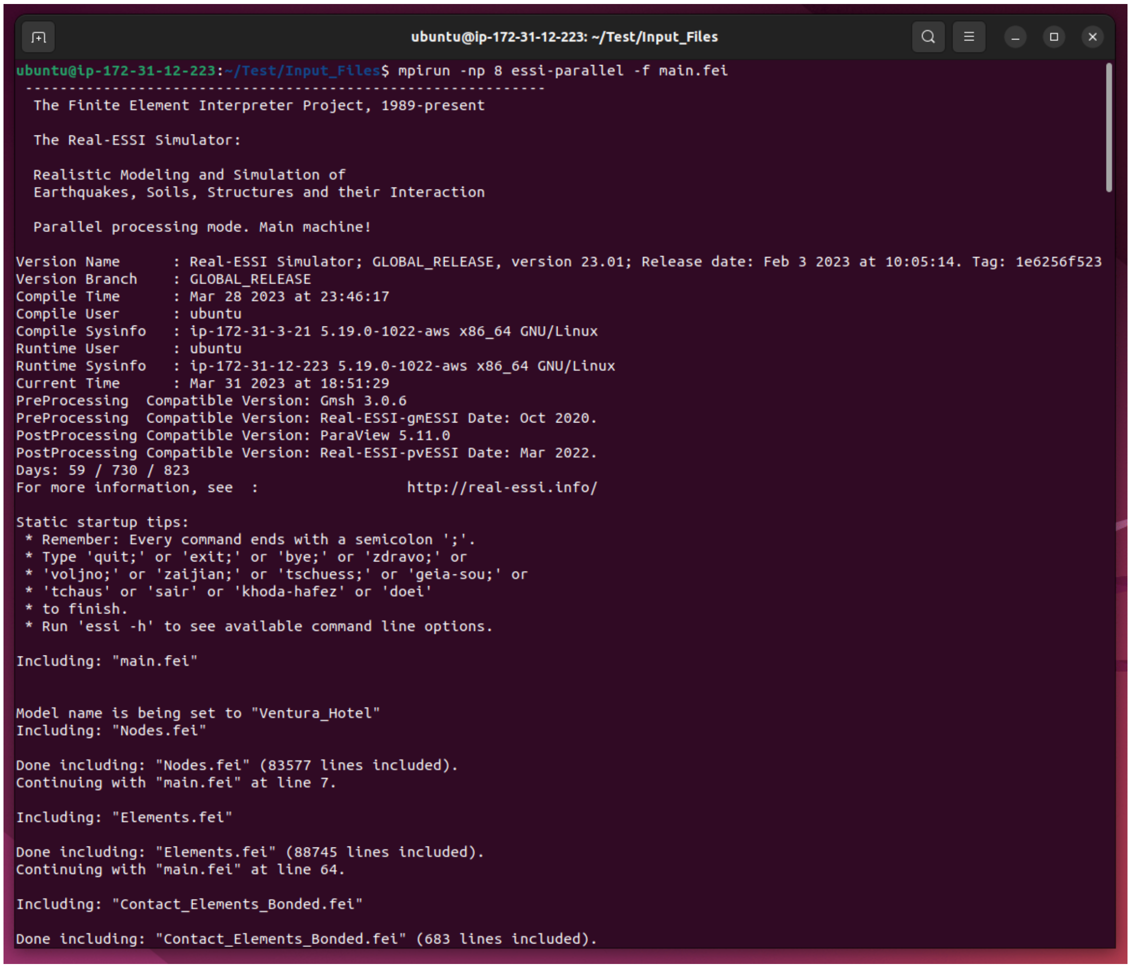
```
chmod a+rx main.fei
```

- Enter the following command to start your simulation:

```
mpirun -np num_pro essi-parallel -f main.fei
```

  Replace num_pro with the number of processes you want to use to run the simulation. Note that this number must be smaller than the number of available processes on your Real-ESSI instance. Consider using a different instance type with more available processes if your current set up is not enough for your model.

- Once the simulation has successfully started, you should see something like the following:



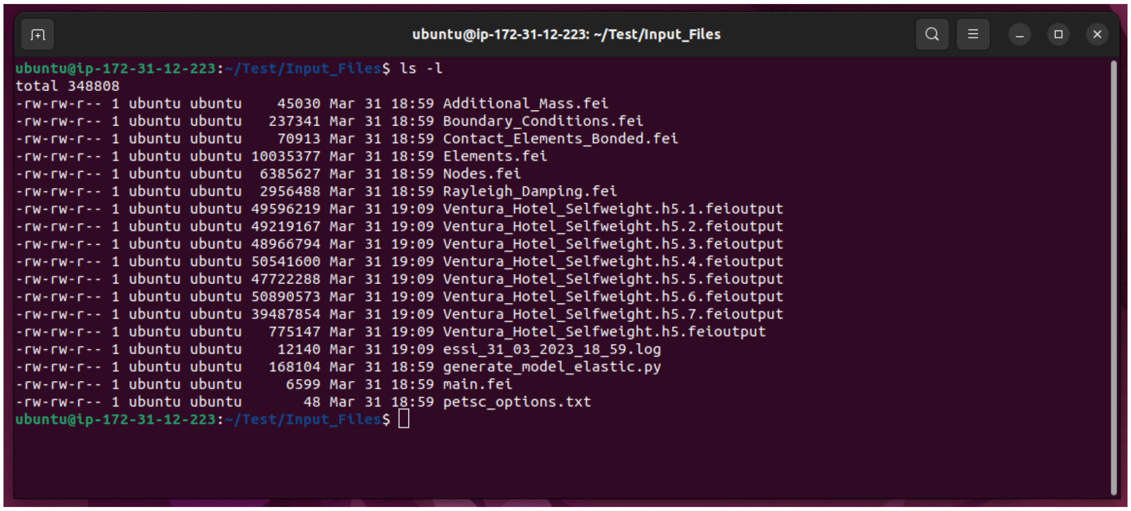
```
ubuntu@ip-172-31-12-223:~/Test/Input_Files$ mpirun -np 8 essi-parallel -f main.fei
 ------------------------------------------------------------
  The Finite Element Interpreter Project, 1989-present

  The Real-ESSI Simulator:

  Realistic Modeling and Simulation of
  Earthquakes, Soils, Structures and their Interaction

  Parallel processing mode. Main machine!

Version Name      : Real-ESSI Simulator; GLOBAL_RELEASE, version 23.01; Release date: Feb 3 2023 at 10:05:14. Tag: 1e6256f523
Version Branch    : GLOBAL_RELEASE
Compile Time      : Mar 28 2023 at 23:46:17
Compile User      : ubuntu
Compile Sysinfo   : ip-172-31-3-21 5.19.0-1022-aws x86_64 GNU/Linux
Runtime User      : ubuntu
Runtime Sysinfo   : ip-172-31-12-223 5.19.0-1022-aws x86_64 GNU/Linux
Current Time      : Mar 31 2023 at 18:51:29
PreProcessing  Compatible Version: Gmsh 3.0.6
PreProcessing  Compatible Version: Real-ESSI-gmESSI Date: Oct 2020.
PostProcessing Compatible Version: ParaView 5.11.0
PostProcessing Compatible Version: Real-ESSI-pvESSI Date: Mar 2022.
Days: 59 / 730 / 823
For more information, see  :                http://real-essi.info/

Static startup tips:
 * Remember: Every command ends with a semicolon ';'.
 * Type 'quit;' or 'exit;' or 'bye;' or 'zdravo;' or
 * 'voljno;' or 'zaijian;' or 'tschuess;' or 'geia-sou;' or
 * 'tchaus' or 'sair' or 'khoda-hafez' or 'doei'
 * to finish.
 * Run 'essi -h' to see available command line options.

Including: "main.fei"


Model name is being set to "Ventura_Hotel"
Including: "Nodes.fei"

Done including: "Nodes.fei" (83577 lines included).
Continuing with "main.fei" at line 7.

Including: "Elements.fei"

Done including: "Elements.fei" (88745 lines included).
Continuing with "main.fei" at line 64.

Including: "Contact_Elements_Bonded.fei"

Done including: "Contact_Elements_Bonded.fei" (683 lines included).
```

Now you just need to wait for the simulation to finish.

- Once the simulation is finished, you can use the following command to list all the files in your current directory:

```
ls -l
```

You should see something like the following:



```
ubuntu@ip-172-31-12-223:~/Test/Input_Files$ ls -l
total 348808
-rw-rw-r-- 1 ubuntu ubuntu    45030 Mar 31 18:59 Additional_Mass.fei
-rw-rw-r-- 1 ubuntu ubuntu   237341 Mar 31 18:59 Boundary_Conditions.fei
-rw-rw-r-- 1 ubuntu ubuntu    70913 Mar 31 18:59 Contact_Elements_Bonded.fei
-rw-rw-r-- 1 ubuntu ubuntu 10035377 Mar 31 18:59 Elements.fei
-rw-rw-r-- 1 ubuntu ubuntu  6385627 Mar 31 18:59 Nodes.fei
-rw-rw-r-- 1 ubuntu ubuntu  2956488 Mar 31 18:59 Rayleigh_Damping.fei
-rw-rw-r-- 1 ubuntu ubuntu 49596219 Mar 31 19:09 Ventura_Hotel_Selfweight.h5.1.feioutput
-rw-rw-r-- 1 ubuntu ubuntu 49219167 Mar 31 19:09 Ventura_Hotel_Selfweight.h5.2.feioutput
-rw-rw-r-- 1 ubuntu ubuntu 48966794 Mar 31 19:09 Ventura_Hotel_Selfweight.h5.3.feioutput
-rw-rw-r-- 1 ubuntu ubuntu 50541600 Mar 31 19:09 Ventura_Hotel_Selfweight.h5.4.feioutput
-rw-rw-r-- 1 ubuntu ubuntu 47722288 Mar 31 19:09 Ventura_Hotel_Selfweight.h5.5.feioutput
-rw-rw-r-- 1 ubuntu ubuntu 50890573 Mar 31 19:09 Ventura_Hotel_Selfweight.h5.6.feioutput
-rw-rw-r-- 1 ubuntu ubuntu 39487854 Mar 31 19:09 Ventura_Hotel_Selfweight.h5.7.feioutput
-rw-rw-r-- 1 ubuntu ubuntu   775147 Mar 31 19:09 Ventura_Hotel_Selfweight.h5.feioutput
-rw-rw-r-- 1 ubuntu ubuntu    12140 Mar 31 19:09 essi_31_03_2023_18_59.log
-rw-rw-r-- 1 ubuntu ubuntu   168104 Mar 31 18:59 generate_model_elastic.py
-rw-rw-r-- 1 ubuntu ubuntu     6599 Mar 31 18:59 main.fei
-rw-rw-r-- 1 ubuntu ubuntu       48 Mar 31 18:59 petsc_options.txt
ubuntu@ip-172-31-12-223:~/Test/Input_Files$ 
```

Notice that all the output files and a log file are now present in your working directory. Note that all Real-ESSI output files have the suffix .feioutput. For more information, refer to the Real-ESSI Simulator Output Format Manual.

9. Transfer Output Files to Local Desktop (or Amazon Workspace)

This chapter provides information about how to transfer the output files of your Real-ESSI simulation back to your local desktop (or Amazon Workspace). Note that you should only attempt to do this after your simulation has finished.

The **scp** command is used to transfer the output files from the launched Real-ESSI instance to your local Ubuntu desktop (or Amazon Workspace).

- Open a terminal on your local Ubuntu desktop (or Amazon Workspace).
- In the following example, the name of the private key file is key-name.pem and the IPv4 address of the instance is IPv4-address. Enter the following command in your terminal.

```
scp -i /path1/key-name.pem ubuntu@IPv4-address:/path2/*.feioutput /path3/
```

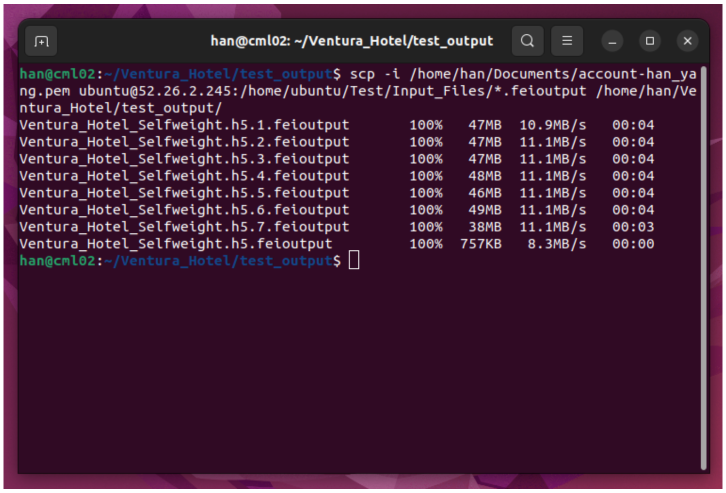
There are three paths in this command.
➢ Replace /path1/ with the full absolute path to your key pair on your local desktop.
➢ Replace /path2/ with the full absolute path to the location on the Real-ESSI instance.
➢ Replace /path3/ with the full absolute path to the location on your local desktop.

Note that the format *.feioutput means that the scp command will be executed for all the files with the suffix .feioutput. This is useful since you will have multiple output files that need to be transferred.

For example, your **scp** command may be:

```
scp -i /home/han/Documents/han-key.pem
ubuntu@52.26.2.245:/home/ubuntu/Test/*.feioutput /home/han/output/
```

- If the transfer is successful, you should see something like the following:



```
han@cml02:~/Ventura_Hotel/test_output$ scp -i /home/han/Documents/account-han_ya
ng.pem ubuntu@52.26.2.245:/home/ubuntu/Test/Input_Files/*.feioutput /home/han/Ve
ntura_Hotel/test_output/
Ventura_Hotel_Selfweight.h5.1.feioutput       100%   47MB  10.9MB/s   00:04
Ventura_Hotel_Selfweight.h5.2.feioutput       100%   47MB  11.1MB/s   00:04
Ventura_Hotel_Selfweight.h5.3.feioutput       100%   47MB  11.1MB/s   00:04
Ventura_Hotel_Selfweight.h5.4.feioutput       100%   48MB  11.1MB/s   00:04
Ventura_Hotel_Selfweight.h5.5.feioutput       100%   46MB  11.1MB/s   00:04
Ventura_Hotel_Selfweight.h5.6.feioutput       100%   49MB  11.1MB/s   00:04
Ventura_Hotel_Selfweight.h5.7.feioutput       100%   38MB  11.1MB/s   00:03
Ventura_Hotel_Selfweight.h5.feioutput         100%  757KB   8.3MB/s   00:00
han@cml02:~/Ventura_Hotel/test_output$ 
```

- Now that you have all the output files on your local Ubuntu desktop (or Amazon Workspace), you can proceed to work with them as you wish, e.g. post-processing. For more information on what you can do with your output files, refer to the Real-ESSI Simulator Post-Processing Manual.

211.8.3 AWS for Education

Amazon Web Services provides grants for educators and students from member institution[2] through AWS Educate program. AWS Educate offers cloud content, training, collaboration tools and AWS technology at no cost. Some of the AWS Educate program benefits:

- For Educators
  - $200 in AWS credits per educator - at member institutions.
  - $75 in AWS credits per educator - at non-member institutions.
  - Free AWS Technical Essentials eLearning course.
  - Free access to AWS content for classes.
- For Students
  - $100 in AWS credits per student - at member institutions.
  - $40 in AWS credits per student - at non-member institutions.
  - Access to AWS Technical Essentials Training Course (a $600 value).

If you have an email address from an educational institutions, you can use Real-ESSI on AWS for free through AWS Educate.

---

[2] List of member institution is available at this LINK.

211.8.4 AWS for Government

211.8.4.1 AWS GovCloud

211.8.4.2 AWS Secret Region

Chapter 212

Hardware Platform Design and Development

(1996-1999-2009-2011-)

212.1 Chapter Summary and Highlights

212.2 Introduction

Parallel computer used for simulations is based on a Beowulf concept (Sterling et al., 1995; Reschke et al., 1996; Sterling et al., 1998, 1999; Warren et al., 1998; Ridge et al., 1997). Hardware for a specific application to parallel computing for elastic-plastic finite elements has gone through a number of iterations (Jeremić et al., 1998, 1999) and is still evolving as the hardware components change in time and as our algorithms change/improve.

The choice of hardware platform is for a cost effective, off the shelf PC components, with about 2GB of memory per CPU/core and plenty of disk space (about 4TB for large dynamic runs). Linux is chosen as an operating system as it offers the best performance, is available in open source, which ensures that significant number of developers can contribute their expertise and can be customized to suite the needs of a parallel hardware and software. Microsoft Windows or Apple IOS operating systems are not best suited for parallel computing as their main development goal is user friendliness and not efficiency.

212.3 The NRC ESSI Computer

212.3.1 Version: December 2010

Operating System: Linux Fedora Core 14.

Kernel: `2.6.35.10-74.fc14.x86_64`

Compute Nodes (two):

- CPU: 2 × Intel Xeon E5620 Westmere 2.4 GHz Quad Core (8 threads) 32nm CPU with 256 KB Cache/core and 12MB Shared L3, DDR3-1066, 5.86GT/sec QPI, 80W
- RAM: 6 × 4GB DDR3 1333 MHz ECC/Registered Memory (24GB Total Memory 1066MHz)
- Disk: 8 × 500 GB Seagate Constellation ES 3.5" SATA/300 ST3500514NS 32MB Cache, 3Gb/s, NCQ, 7200RPM, 1.2 million hours MTBF Maximum Sustained Transfer Rate: 140 MB/sec (Linux Software RAID10)

Network: single GigaBit

212.3.2 Version: April 2012

In addition to the previous version.

Operating System:

Kernel:

Controller Node 1 (one):

- CPU: 2 x Opteron 6234 (2.4GHz, 12-Core, G34, 16MB L3 Cache) 115W TDP, 32nm
- RAM: 32GB (8 x 4GB) Operating at 1333MHz Max (DDR3-1333 ECC Registered DIMMs)
- NICs: Integrated Intel 82576 Dual-Port Gigabit Ethernet Controller
- Disk: 8 × 2TB Toshiba MK2002TSKB (3Gb/s, 7.2K RPM, 64MB Cache) 3.5" SATA

Compute Nodes, 8 (eight):

- CPU: 2 × Opteron 6234 (2.4GHz, 12-Core, G34, 16MB L3 Cache) 115W TDP, 32nm
- RAM: 32GB (8 x 4GB) Operating at 1333MHz Max (DDR3-1333 ECC Registered DIMMs)
- NICs:
  - Intel 82576 Dual-Port Gigabit Ethernet Controller
  - InfiniBand: ConnectX-2 QDR IB 40Gb/s Controller with QSFP Connector
- Disk: 1TB Toshiba MK1002TSKB (3Gb/s, 7.2K RPM, 64MB Cache) 3.5" SATA

Network (dual):

- HP ProCurve Switch 1810-48G 48 Port 10/100/1000 ports Web Managed Switch
- IB Switch: Mellanox MIS5030Q-1SFCA 36-port QDR switch; Cables: 9 x 3mtr QSFP-QSFP - Rating: QDR

Part 300

Verification and Validation

Chapter 301

Verification and Validation Introduction

(2003-2007-2009-2017-)

301.1 Chapter Summary and Highlights

301.2 Important Literature

Suggested reading:

Roache (1998);

Oberkampf et al. (2002); Oberkampf (2003); Oberkampf et al. (2007); Oberkampf and Trucano (2008); Oberkampf and Roy (2010); Oberkampf and Pilch (2017);

Roy and Oberkampf (2011);

Babuška and Oden (2004); Babuska et al. (2004); Oden et al. (2005); Oden et al. (2010a); Oden et al. (2010b); Szabó and Actis (2011) Szabó and Actis (2012)

ASME-VV-10 (2019) ASME-VV-20 (2009) ASME-VV-40 (2018)

ISO-90003 (2018)

NASA (2008) NASA (2016)

301.3 Verification and Validation

301.3.1 Definitions

Some definitions, as seen in ASME-VV-10 (2019).

Code is a computer implementation of the algorithm developed to facilitate formulation and approximations, approximate solutions a physical problem.

Model is a representation of a system, in our case a soil and/or structure with all the loads. physical conditions. Representation includes conceptual mathematical and computational models can also include physical

Verification and validation terms revision have been used interchangeably. However, it's important to follow precise definitions:

verification has to do with mathematics

validation has to do with physics

Objectives The objective of the verification and validation are

- demonstrate credibility of simulation results
- assess reliability of computer software and numerical methods used in simulation, and to
- assess the accuracy of simulations, with respect to available experimental observations.

- How do we use experimental simulations to develop and improve models
- How much can (should) we trust model implementations (verification)
- How much can (should) we trust numerical simulations (validation)

301.3.2 Trusting Simulation Tools

- Verification: The process of determining that a model implementation accurately represents the developer's conceptual description and specification. Mathematics issue. *Verification provides evidence that the model is solved correctly.*
- Validation: The process of determining the degree to which a model is accurate representation of the real world from the perspective of the intended uses of the model. Physics issue. *Validation provides evidence that the correct model is solved.*

301.3.3 Importance of V & V

- V & V procedures are the primary means of assessing accuracy in modeling and computational simulations
- V & V procedures are the tools with which we build confidence and credibility in modeling and computational simulations

301.3.4 Maturity of Computational Simulations

NRC committee (1986) identified stages of maturity in CFD

- Stage 1: Developing enabling technologies (scientific papers published)
- Stage 2: Demonstration of and Confidence in technologies and tools (capabilities and limitations of technology understood)
- Stage 3: Compilation of technologies and tools (capabilities and limitations of technology understood)
- Stage 4: Spreading of the effective use (changes the engineering process, value exceeds expectations)
- Stage 5: Mature capabilities (fully dependable, cost effective design applications)

301.3.5 Role of Verification and Validation

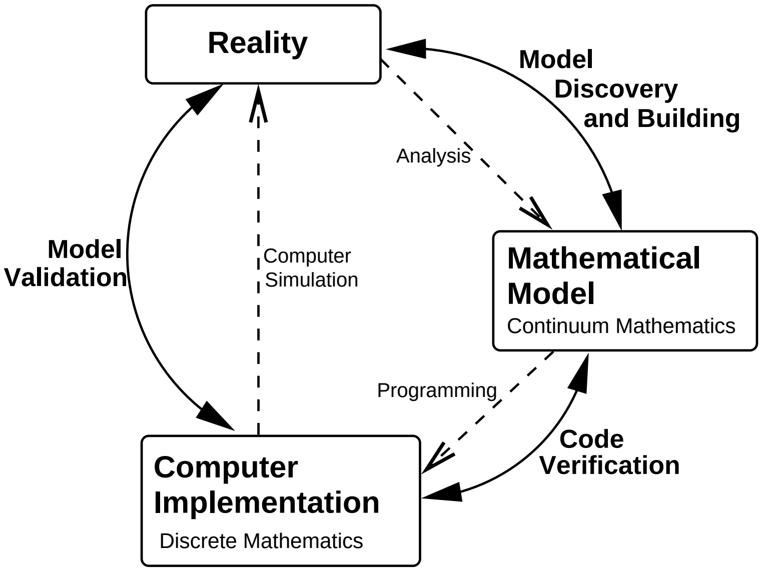


Figure 301.1: Role of Verification and Validation per Oberkampf et al. (2002).

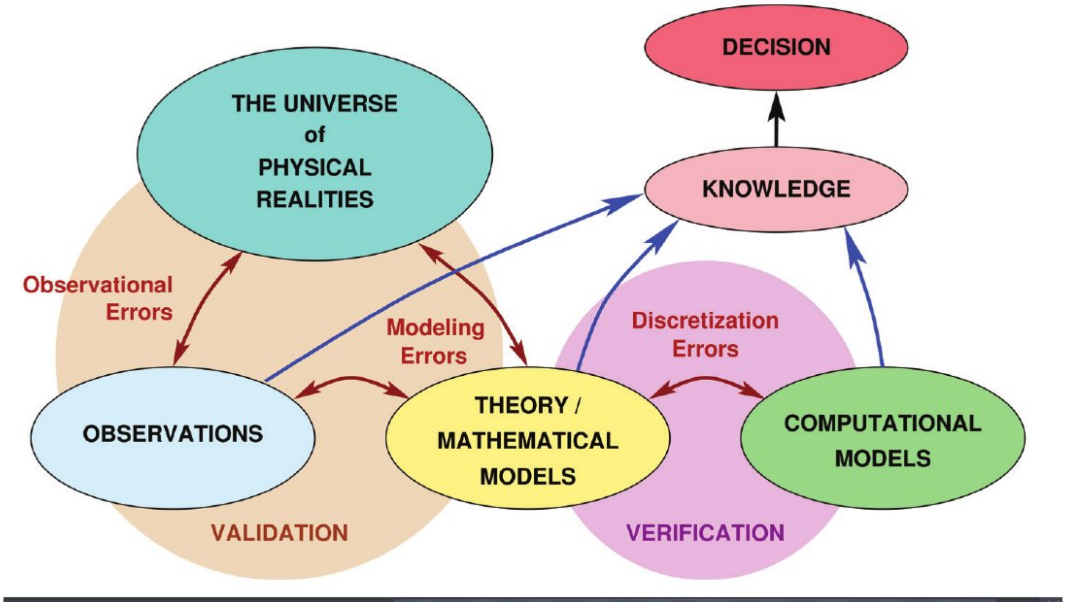


Figure 301.2: Role of Verification and Validation per Oden et al. (2010a).

301.3.5.1 Alternative V & V Definitions

IEEE V & V definitions (1984):

- Verification: The process of determining whether the products of a given phase of the software development cycle fulfill the requirements established during the previous phase
- Validation: The process of evaluating software at the end of the software development process to ensure compliance with software requirements.
- Other organization have similar definitions:
    - Software quality assurance community
    - American Nuclear Society (safety analysis of commercial nuclear reactors)
    - International Organization for Standardization (ISO)

301.3.5.2 Certification and Accreditation

- Certification: A written guarantee that a system or component complies with its specified requirements and is acceptable for operational use (IEEE (1990)).
    - Written guarantee can be issued by anyone (code developer, code user, independent code evaluator)
    - Code certification is more formal than verification and validation documentation
- Accreditation: The official certification that a model or simulation is acceptable for use for a specific purpose (DOD/DMSO (1994))
    - Only officially designated entities can accredit
    - Normally appointed by the customers/users of the code or legal authority
    - Appropriate for major liability or public safety applications

301.3.5.3 Independence of Computational Confidence Assessment

1. V&V conducted by the computational tool developer; *No Independence*
2. V&V conducted by a user from same organization
3. V&V conducted by a computational tool evaluator contracted by developer's organization
4. V&V conducted by a computational tool evaluator contracted by the customer

5. V&V conducted by a computational tool evaluator contracted by the a legal authority; *High Independence*

301.3.6 Simulation-Informed Decision Making

Based on Oberkampf and Pilch (2017)...

301.3.6.1 Purpose of Modeling and Simulation

Decision process:

- Low risk decisions: Simulation-based
- High risk decisions: computer simulations together with testing, Simulation-informed decisions process when simulation is integrated

Computer simulation is an Information Product:

- Supports a decision-making process
- Provides an improved understanding of the uncertainties and risk sustained by the use of the simulation results, a Simulation Customer.

Computer simulations are used to explore the

- design space
- use-misuse space

of a systems where physical testing is impractical and unaffordable...
What if studies over operating environment.
Operating environment:

- Normal
- Accident
- Misuse

301.3.7 Decision Making by Industry and by Regulatory Authorities

Based on NAFEMS short course on Credibility and Decision Making, June 2021, online...

Oberkampf...

Industry decision process to achieve decision result:

- return of investment, profit margin
- organizational goals
- competition
- personal goals
- organizational/personal value system
- experience with available options
- risk tolerance vs potential reward
- familiarity with information sources
- reliability of information sources

Regulatory Authority decision making:

- Risk to public
- Risk to environmental safety
- loss of political support for your regulatory function
- very risk averse value system

Information sources for decision support

- Previous experience with
  - Similar system of process, for example operating and reliability history
  - Modeling and simulation information, for example good and bad experiences
- Experimental data from testing prototype systems and subsystems
  - testing over portion of the operating envelope (application domain)

  - limited testing in adverse/abnormal environments, hazard loads...
  - limited testing of failure modes

- Credibility of modeling and simulation information depends on THREE elements:
  1. Suitable Training and Experienced personnel (example Sandia NL Fracture challenge (Boyce et al., 2014)), is a necessary element but not sufficient
  2. Quality Control of Modeling and Simulation Process, to achieve SIMUALTION CREDIBILITY
     - Physics modeling fidelity (geometric fidelity, spatial scales, temporal scales, initial conditions, boundary conditions, matgerial characteristics)
     - Verification activities (software quality assurance, static testing, dynamic testing, traditional analytic solutions, manufactured solutions, order of accuracy assessment)
     - Validation activities (validation experiments, hierarchical experiments, validation simulations, validation metrics, spatial discretization error, temporal discretization)
     - Uncertainty quantification (parametric uncertainty, model-form (modeling/epistemic) uncertainty, normal environments, abnormal environments, hostile environments, sensitivity analysis, extrapolation uncertainty)
  3. Assessment of Maturity of Modeling and Simulation (M&S) Results
     - Improved clarity of credibility in M&S results
       - Explicit statement of assumptions
       - Explicit statement of approximations
       - Explicit statement of limitations of simulation
       - Explicit statement of restrictions of simulation
       - Use of blind prediction of experimental tests
       - Use of uncertainty quantification of results
     - Use of M&S maturity assessment techniques
       - Predictive Capability Maturity Model (PCMM)
       - NASA standard for M&S (NASA, 2008, 2016)
       - ASME VVUQ techniques (ASME-VV-10, 2019; ASME-VV-40, 2018)

Responsibility for VVUQ

- Software Developer

  - Software quality assurance, bug fixes
  - Testing on different hardware and OS platforms
  - Code verification (determination of observed order of convergence)
  - Solution verification tools to estimate numerical solution errors
  - Model validation
  - Documentation of all components, models, algorithms, API, V&V, examples
  - Reproducibility of the simulation

- Simulation Producer, Analyst
  - Solution verification
  - Model parameter calibration using experimental data
  - Model validation, together with experimentalistis
  - Probabilistic simulation and uncertainty quantification
  - Software quality assurance
  - Code verification, in particular coverafe of features used
  - Documentation of simulation
  - Reproducibility of the simulation

- Simulation, Analysis Manager
  - Specifying how will simulation be used
  - Specifying what is the purpose of simulation to be used
  - Developing quality requirements for simulation
  - Developing review requirements for simulation
  - Assure, allocate adequate resources for the simulation
  - Assure, allocate adequate time for the simulation
  - Define documentation requirements for simulations
  - Define documentation requirements for experiments
  - Document in detail assumption, approximations and limitations of the simulation
  - Ensure training for analysis, modeling and simulation, personnel
  - Ensure expertise level of analysis, modeling and simulation, personnel

- Simulation Customer
  - Specifying what is the purpose of simulation to be used
  - Developing quality requirements for simulation
  - Developing review requirements for simulation
  - Allocate adequate resources for analyzing simulation results
  - Define documentation requirements for simulations
  - Ensure understanding of assumption, approximations and limitations of the simulation
  - Ensure expertise level of analysis users

Simulation Credibility Versus Risk, a Trade

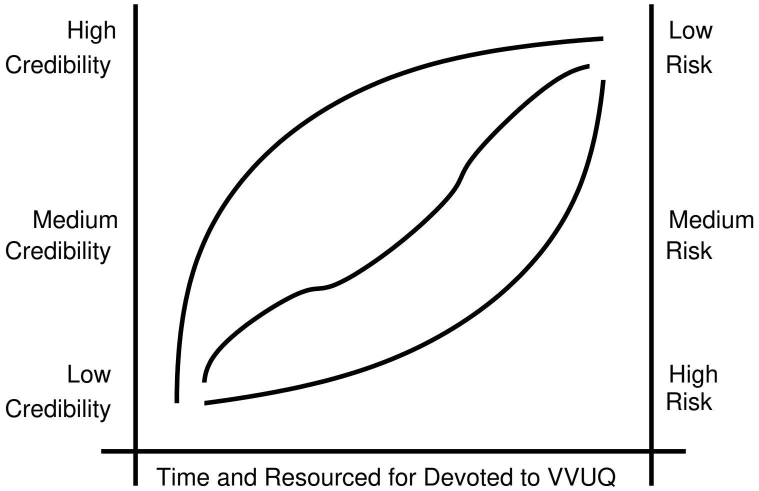


Figure 301.3: Simulation credibility versus risk, a trade that must be observed (Oberkampf...)

What one can find Under the Simulation Hood can be Surprising.

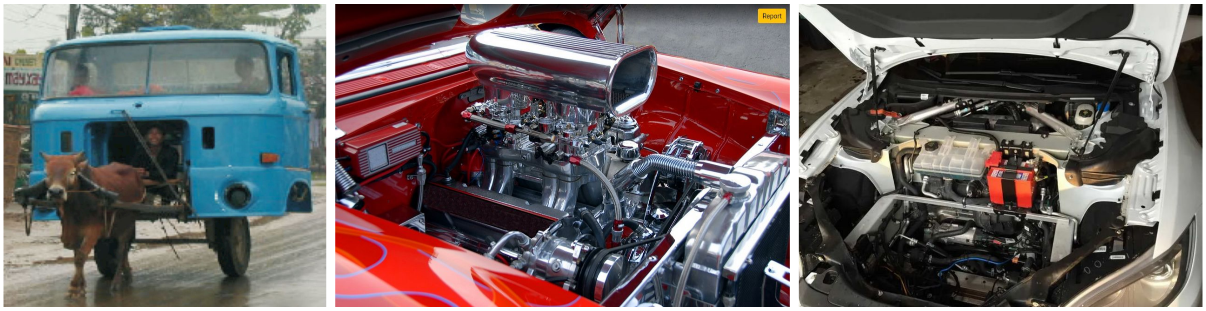


Figure 301.4: Under the Simulation Hood...

Simulation is not a magic bullet!

Simulation is Fragile!

301.3.8 Simulation Governance

Based on Szabó and Actis (2011, 2012)

Simulation Governance (SG) is needed to address prediction challenges in engineering practice. SG is a process to rank analysis models and to improve them over time with new experimental data. Control of numerical and modeling error is essential! Performance, relative and absolute, of analysis models is objectively evaluated.

301.3.8.1 Modeling, Experimental, Analytic and Numerical

Based on Szabó and Actis (2011, 2012)

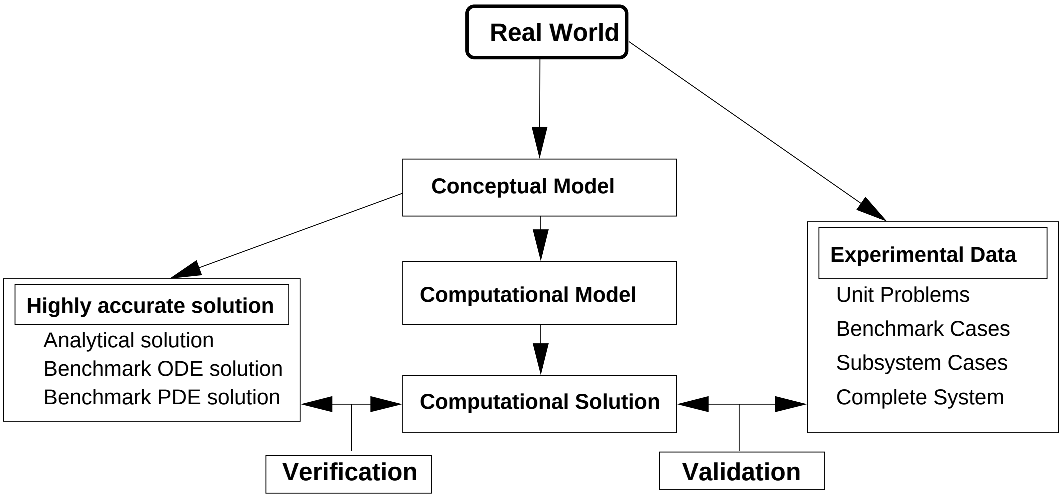
- Model is a transformation of data D that describes physical reality, into the data of interest F, results
- Data D includes all geometrical information, material information, calibration data and loading information...
- Data D features many uncertainties
- These uncertainties are transferred into corresponding uncertainties in F

- Transformation, D → F consists of operations that include modeling and simulation of mathematical problems, mechanics problems, statistical problems,
- Epistemic, parametric and aleatory, modeling uncertainties are mixed
- Simulation governance is set to minimize, control epistemic, modeling uncertainties,

Choice of mathematical model.

- Choice of mathematical model involves simplifying assumptions that restrict the scope of applicability of the model
- Are these simplifying assumptions justified for a particular application?
- This question can be addressed by performing virtual experiments
- Virtual experiments, sensitivity studies to determine if simplifying, restrictive modeling assumptions affect the date of interest to a significant degree
- Using virtual experiments Analyst, engineer can make informed decisions regarding the choice of modeling level of sophistication
- Software for virtual experiments should be able to analyze mathematical problem independent of choice of discretization
- For fixed discretization, one should be able to choose alternative mathematical model and investigate modeling change on the data of interest, results
- Similarly, for fixed mathematical problems, one should be able to develop a sequence of models that data of interest, results converge to their exact values...

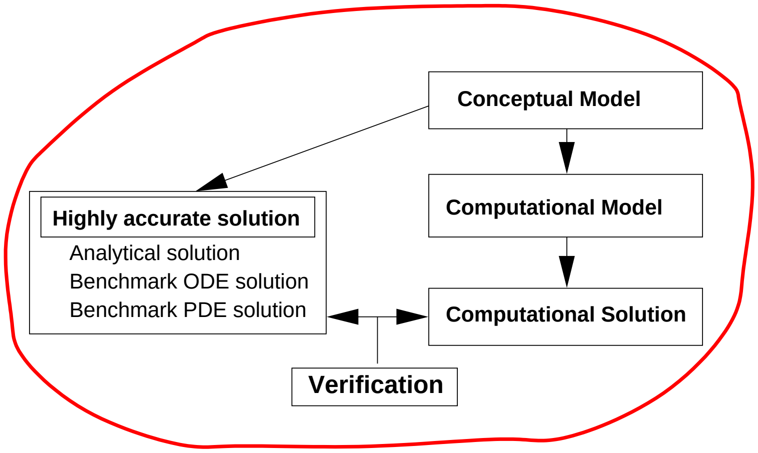
301.3.9 Detailed Look at Verification and Validation



301.3.9.1 On Verification

Verification: The process of determining that a model implementation accurately represents the developer's conceptual description and specification.

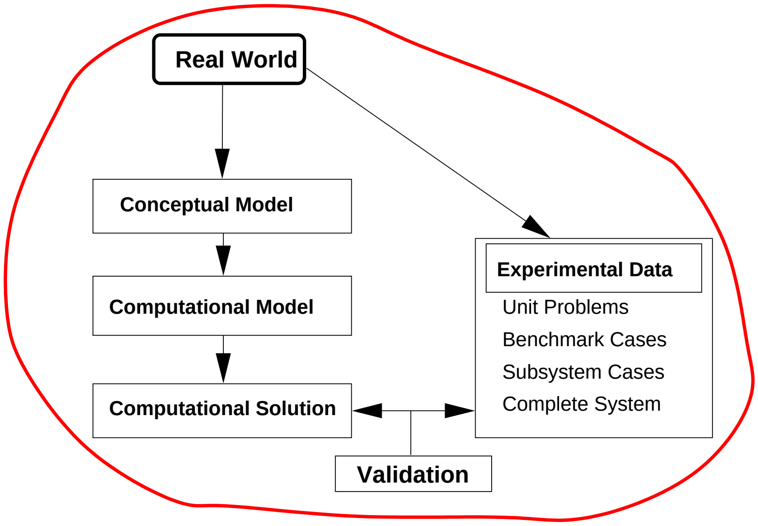
- Identify and remove errors in computer coding
  - Numerical algorithm verification
  - Software quality assurance practice
- Quantification of the numerical errors in computed solution



301.3.9.2 On Validation

Validation: The process of determining the degree to which a model is accurate representation of the real world from the perspective of the intended uses of the model.

- Tactical goal: Identification and minimization of uncertainties and errors in the computational model
- Strategic goal: Increase confidence in the quantitative predictive capability of the computational model



Goals of Validation Quantification of uncertainties and errors in the computational model and the experimental measurements

- Goals on validation
  - Tactical goal: Identification and minimization of uncertainties and errors in the computational model
  - Strategic goal: Increase confidence in the quantitative predictive capability of the computational model
- Strategy is to reduce as much as possible the following:
  - Computational model uncertainties and errors
  - Random (precision) errors and bias (systematic) errors in the experiments
  - Incomplete physical characterization of the experiment

Validation Procedure Uncertainty

- Aleatory uncertainty → inherent variation associated with the physical system of the environment (variation in external excitation, material properties...). Also know known as irreducible uncertainty, variability and stochastic uncertainty.
- Epistemic uncertainty → potential deficiency in any phase of the modeling process that is due to lack of knowledge (poor understanding of mechanics...). Also known as reducible uncertainty, model form uncertainty and subjective uncertainty

Types of Physical Experiments

- Traditional Experiments
  - Improve the fundamental understanding of physics involved
  - Improve the mathematical models for physical phenomena
  - Assess component performance
- Validation Experiments
  - Model validation experiments
  - Designed and executed to quantitatively estimate mathematical model's ability to simulate well defined physical behavior
  - The simulation tool (SimTool) (conceptual model, computational model, computational solution) is the customer

Validation Experiments

- A validation experiment should be jointly designed and executed by experimentalist and computationalist
  - Need for close working relationship from inception to documentation
  - Elimination of typical competition between each
  - Complete honesty concerning strengths and weaknesses of both experimental and computational simulations
- A validation Experiment should be designed to capture the relevant physics
  - Measure all important modeling data in the experiment

  - Characteristics and imperfections of the experimental facility should be included in the model

- A validation experiment should use any possible synergism between experiment and computational approaches
  - Offset strength and weaknesses of computations and experiments
  - Use high confidence simulations for simple physics to calibrate of improve the characterization of the experimental facility
  - Conduct experiments with a hierarchy of physics complexity to determine where the computational simulation breaks (remember, SimTool is the customer!)

- Maintain independence between computational and experimental results
  - Blind comparison, the computational simulations should be predictions
  - Neither side is allowed to use fudge factors, parameters

- Validate experiments on unit level problems, hierarchy of experimental measurements should be made which present an increasing range of computational difficulty
  - Use of qualitative data (e.g. visualization) and quantitative data
  - Computational data should be processed to match the experimental measurement techniques

- Experimental uncertainty analysis should be developed and employed
  - Distinguish and quantify random and correlated bias errors
  - Use symmetry arguments and statistical methods to identify correlated bias errors
  - Make uncertainty estimates on input quantities needed by the SimTool

301.4 Prediction

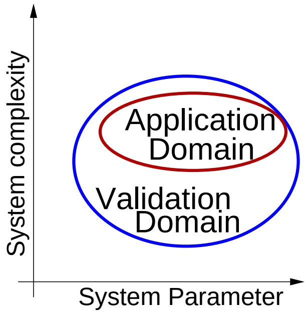
- Prediction: use of computational model to foretell the state of a physical system under consideration under conditions for which the computational model has not been validated
- Validation does not directly make a claim about the accuracy of a prediction
  - Computational models are easily misused (unintentionally or intentionally)
  - How closely related are the conditions of the prediction and specific cases in validation database
  - How well is physics of the problem understood

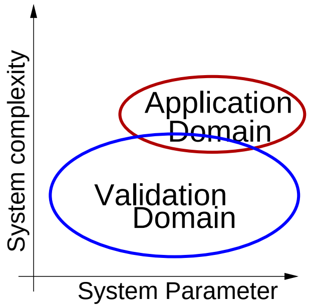
301.4.1 Relation Between Validation and Prediction

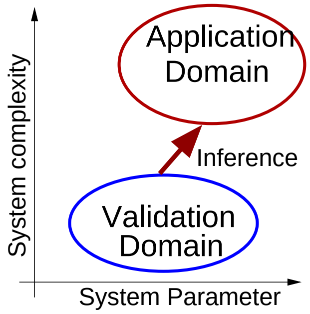
Quantification of confidence in a prediction:

- How do I quantify validation and its inference value in a predictions?
- How do I quantify verification and its inference value in a prediction?
- How far are individual experiments in my validation database from my physical system of interest?

301.5 Application Domain







- Rarely applicable to engineering systems (certainly not for infrastructure objects like bridges, buildings, port facilities, dams...)
- Even if the engineering system is small, environmental influences (generalized loads, conditions, wear and tare) are hard to predict
- Human factors (take Mars rover example with a memory overflow, operator forgot to flush the memory...)
- Inference ⇒ Based on physics or statistics
- Validation domain is actually an aggregation of tests and thus might not be convex (bifurcation of behavior)
- Experimental facilities provide validation domain that is (for the most part) exclusively non–overlapping with the application domain.

301.5.1 Importance of Models and Numerical Simulations

- Verified and Validated models can be used for assessing behavior of
  - components or
  - complete systems,
- with the understanding that the environmental influences cannot all be taken into the account prior to operation
- but with a good model, their influence on system behavior can be assessed as need be (before or after the event)

301.5.2 Prediction under Uncertainty

- Ever present uncertainty needs to be estimated for predictions
- Identify all relevant sources of uncertainty
- Create mathematical representation of individual sources
- Propagate representation of sources through modeling and simulation process (Probabilistic Elastic Plastic Theory)

301.6 Intended use of Model

(ASME-VV-10, 2019), from Executive Summary, section 1

Verification assess the numerical accuracy of a computational model regardless of the physics being modeled.

- Code verification, addressing errors in the software and numerical algorithms
- Calculation verification, estimating numerical error due to under-resolved discrete representation of the mathematical model

Validation assess the degree to which the computational model is an accurate representation of the physics being modeled.

301.6.1 System Being Modeled

Relevant Physics

Modeling and experimental activities are guided by response quantities of interest.

301.6.2 Simulation Governance

Szabó and Actis (2011)

Szabó and Actis (2012)

Chapter 302

Source Code Verification

(1989-1990-1994-1995-2002-2005-2007-2009-2010-2017-2019-2021-)

(In collaboration with Dr. Yuan Feng, and Prof. Han Yang)

302.1 Chapter Summary and Highlights

Hatton (1997); Roache (1998); Oberkampf et al. (2002); Oberkampf (2003); Oden et al. (2005); Babuška and Oden (2004); Oden et al. (2010a); Oden et al. (2010b);

302.1.1 Numerical Algorithm Verification

302.1.2 Software Quality Assurance

Chapter 303

Code Stability Verification

(2002-2016-2017-2019-2021-)

(In collaboration with Dr. Yuan Feng and Prof. Han Yang)

303.1 Chapter Summary and Highlights

303.2 Introduction to Code Stability

This activity addresses source code stability. Source code verification is addressed elsewhere.

303.3 Motivation

In the software development process, Real-ESSI program is a framework and new features are continuously added. From time to time, some revisions for one feature may affect the normal operation of other existing components, like a specific element or material. To guarantee the stability and correctness of Real-ESSI program for each update. A group of test cases are collected in an automatic test suite. Then, after each revision, the developers are required to successfully pass all the test cases before the revisions are finally accepted to the trunk branch.

The features of the automatic Real-ESSI test is listed below:

- Automatic comparison of the maximum displacement output between the original and the new ESSI results.
- Automatic comparison of all the displacement and stress/strain output between the original and the new ESSI results.
- Automatic comparison of the terminal output/log between the original and the new ESSI results.
- Relative difference between the original and the new ESSI results.
- Colorful diagnostic information in the terminal.
- Support for the report in HTML format.
- Version information of both the original and the new ESSI.
- Number of passed cases and the statistics.

303.4 The framework of the automatic test

In practice, all the test cases are collected in one main folder and each test case has an independent subdirectory. In addition, *Bash* and *Python* are employed go over each leaf directory of the test cases folder to execute essi and compare the results. A verification report is generated automatically after all the test cases are executed.

Regarding the test cases for the version stability, it is not necessary to choose the great model with lots of elements. The goal of version stability is to guarantee that the revision for one feature should not affect the normal operation of other commands. So the selection rule of test cases is to cover as much Real-ESSI DSL (domain specific language) commands as possible.

303.5 Installation and Tutorial

303.5.1 Installation

Makefile, *Bash* and *Python* are used to run all the test cases. So the automatic test is portable over various Linux platform as long as *Makefile*, *Bash* and *Python* are available. Besides, *git* is also required to download the test suite. In addition, if the user wants an additional report in HTML format, another package called *aha* is required. Install *aha* package on Ubuntu by using this command.

The automatic test suite is distributed within Real-ESSI source code.

```
${Real-ESSI}/CompGeoMechUCD_Miscellaneous/examples/
```

Notes: the automatic test will call the executable *essi* in the system/user PATH. So please make sure you have compiled or installed Real-ESSI first and then run this automatic test suite.

303.5.2 Tutorial

303.5.2.1 Run all verification test cases

In order to run all test cases to verify the installation, users can run

```
cd $RealESSI_PATH/
bash run_all_verification_sequential.sh $EXECUTABLE_PATH
bash run_all_verification_parallel.sh $EXECUTABLE_PATH
```

In addition to the conventional "-DDEBUG_MODE=DEBUG or OPTIMIZED", Users are required to do the test for executables compiled using the compiler options "-DDEBUG_MODE=O1 or O2" to fully verify the code. The verification will list all the results and errors. The errors may be big when the mesh is too coarse, or when Poisson's ratio is too high.

Furthermore, if developers want to verify against a previous version, developers can run

```
cd $RealESSI_PATH/
```

```
bash run_code_stability.sh $GitTAG
```

to test the verification results against one previous git version. The script will automatically checkout to the previous git-tag in detached mode and compile the old essi. After running test cases, developers can checkout to the original testing branch by running

```
cd $RealESSI_PATH/
git checkout $TestingBranch
```

In addition, if users want to clean the test results, users can run

```
cd $RealESSI_PATH/
bash clean_all_verification.sh
```

303.5.2.2 Run a single type of verification test cases

The usage of automatic test is written after the build process of source-code in Section 209.7.

For a single type of verification, for instance, in the folder

```
cd $RealESSI_PATH/CompGeoMechUCD_Miscellaneous/examples/analytic_solution
```

there are two clean options available in the main folder.

- The first one is *make clean*. This will only clean the new essi output results, including HDF5 files, terminal logs, and comparison logs.
- The second one is *make cleanall*. This will clean both the new and old essi output results for version stability test.

303.6 The underlying implementation of the automatic test

In most cases, the developers are not required to read and modify the underlying implementation of the automatic test. However, a basic introduction to the underlying implementation will help the future developers to customize the automatic test suite when necessary.

303.6.1 Generate the original results

When the command below is called,

```
bash generate_original.sh
```

Three things will be done. First, *make cleanall* will be called. This means that all the new and original output will be removed. Second, *essi* will be called to run each test cases and to generate the original HDF5 and original terminal output logs. Third, a bash command is employed to rename the HDF5 file and terminal log by adding *original* at the end of the filename.

303.6.2 Run essi and make comparison

When the command below is called,

```
bash make_comparison.sh
```

Four things will be done.

- First, *essi* will be called again to re-run each test cases. Please note that this *essi* should be the newly compiled *essi* in the development and debug stages.
- Second, a python-based comparison function will be called to compare the maximum displacement in the HDF5 output for each test case.
- Third, another python-based comparison function will be called to compare all the displacement and stress/strain results in the HDF5 output for each test case.
- Fourth, the terminal output log will be compared.

303.6.2.1 The terminal output/log comparison

The motivation to compare the terminal output is to avoid the unnecessary debug-purpose messages in the terminal output. During the debug stages, developers usually print out the variable values in the terminal. However, in the production stages, the debug-purpose messages should be disabled.

During the terminal output/log comparison, some lines are always different for each *essi* execution. These lines should be removed from comparison to avoid the false mismatch.

- The first type of different lines are the version information. The *essi* version information includes the *essi* compile time and execution time. They are always different for each execution. Therefore, these lines are extracted and the version information is printed out at the end of the comparison report.
- The second type of different lines are the *ETA* information. *ETA* stands for estimated time of arrival, which is a prediction for the *essi* execution time. However, the *ETA* information is inaccurate and they are different for each execution. Therefore, the *ETA* information is not compared either.

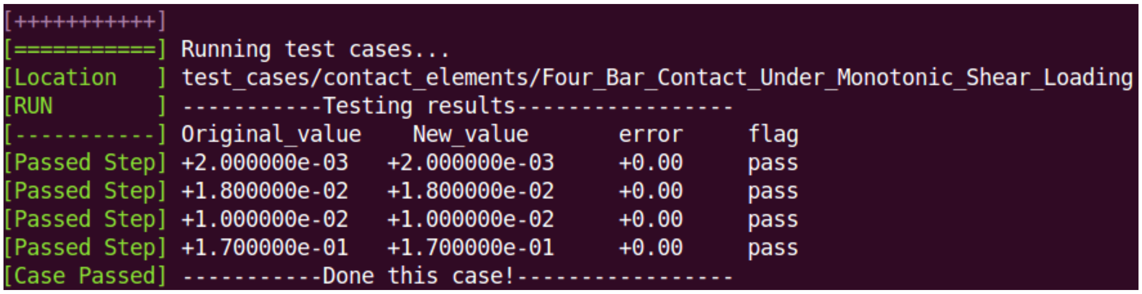
303.6.2.2 Reduce the comparison items and comparison time

During the debug stages, the developers might only want to compare the HDF5 output and the developers want to keep the verbose terminal messages so developers do not want to compare the terminal output log. Besides, during the debug stages, the developers may not want the time-consuming HDF5 output comparison. Therefore, to reduce the comparison items and comparision time, the developers can comment out the last two lines in *make_comparison.sh*.

303.7 Report Sample

For the sake of convenience, automatic test provides colorful diagnostic information. The green color is for the passed (matched) test case, while the red color is for the failed (mismatched) test case. The illustrative results are shown below. In addition, the automatic test also reports the relative path (location) of the test case. So if one of the test cases failed, developers can locate the subdirectory easily and check the mismatched model.

303.7.1 Passed test case

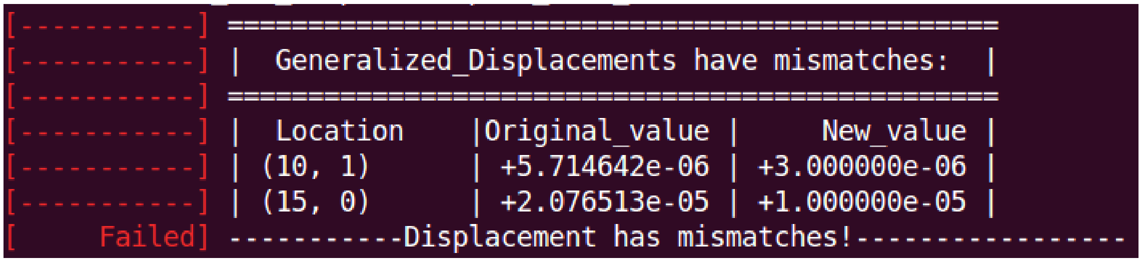


```
[+++++++++++]
[===========] Running test cases...
[Location   ] test_cases/contact_elements/Four_Bar_Contact_Under_Monotonic_Shear_Loading
[RUN        ] ----------Testing results----------------
[-----------] Original_value    New_value       error     flag
[Passed Step] +2.000000e-03   +2.000000e-03     +0.00     pass
[Passed Step] +1.800000e-02   +1.800000e-02     +0.00     pass
[Passed Step] +1.000000e-02   +1.000000e-02     +0.00     pass
[Passed Step] +1.700000e-01   +1.700000e-01     +0.00     pass
[Case Passed] ----------Done this case!----------------
```

Figure 303.1: The report sample for a passed test case

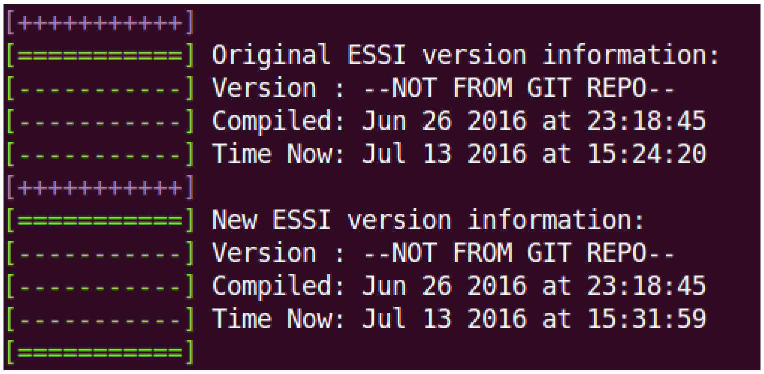
303.7.2 Failed test case

In Figure 303.2, the *location* means the value location in the displacement results matrix of a HDF5 file. In the displacement results matrix, the column number is the step number and the row number is the dof (degree of freedom) number.



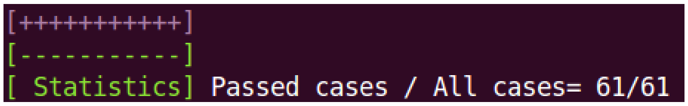
```
[-----------] ===============================================
[-----------] |  Generalized_Displacements have mismatches:  |
[-----------] ===============================================
[-----------] |  Location     |Original_value |      New_value |
[-----------] | (10, 1)       | +5.714642e-06 | +3.000000e-06 |
[-----------] | (15, 0)       | +2.076513e-05 | +1.000000e-05 |
[     Failed] -----------Displacement has mismatches!----------------
```

Figure 303.2: The report sample for a failed test case



```
[+++++++++++]
[===========] Original ESSI version information:
[-----------] Version : --NOT FROM GIT REPO--
[-----------] Compiled: Jun 26 2016 at 23:18:45
[-----------] Time Now: Jul 13 2016 at 15:24:20
[+++++++++++]
[===========] New ESSI version information:
[-----------] Version : --NOT FROM GIT REPO--
[-----------] Compiled: Jun 26 2016 at 23:18:45
[-----------] Time Now: Jul 13 2016 at 15:31:59
[===========]
```

Figure 303.3: The report sample for the version information



```
[+++++++++++]
[-----------]
[ Statistics] Passed cases / All cases= 61/61
```

Figure 303.4: The report sample for statistics information

303.7.3 Version information

303.7.4 Statistics

303.8 Future contribution

The automatic test is a test framework. It is easy to contribute your new test cases to the framework. The newly added test case must meet the following two requirements.

- The test case should be added as an independent leaf subdirectory within the *test_cases* folder.
- The test case should have a *main.fei* as the main model file.

Chapter 304

Validation Experiments

(2021-)

304.1 Chapter Summary and Highlights

304.2 Design of Experiments

Design of Experiments (DoE)

Chapter 305

Verification and Validation for Constitutive Problems

(1989-1991-1992-1994-1999-2003-2007-2009-2010-2017-2018-2019-2021-)

(In collaboration with Dr. Yuan Feng and Prof. Han Yang)

305.1 Chapter Summary and Highlights

305.2 Verification of Constitutive Integration

In this section, the accuracy analysis of the implicit algorithm is assessed. Examples of simple models (von Mises and Drucker-Prager) for accuracy analysis are demonstrated to verify general implicit algorithm. Convergence performance analysis is conducted. More details on accuracy analysis and consistent tangent stiffness are explained. Numerical simulation examples are demonstrated using the implemented framework. Special concerns are on the comparison of experimental data and numerical results of Dafalias-Manzari model.

305.2.1 Error Assessment

There are various error measures for the integration algorithms. Simo and Hughes (1998), Manzari and Prachathananukit (2001) used the relative stress norm by Equation 305.1,

$$\delta^r = \frac{\sqrt{(\sigma_{ij} - \sigma^*_{ij})(\sigma_{ij} - \sigma^*_{ij})}}{\sqrt{\sigma^*_{pq}\sigma^*_{pq}}} \tag{305.1}$$

where σ^*_{ij} is the 'exact' stress solution, and σ_{ij} the calculated stress solution. Alternatively, Jeremić and Sture (1997) used the normalized energy norm by Equation 305.2,

$$\delta^n = \frac{\left\| \sigma_{ij} - \sigma^*_{ij} \right\|}{\left\| p^{unit} \right\|} \tag{305.2}$$

where $\left\|\sigma_{ij}\right\|^2 = \sigma_{ij} D_{ijkl} \sigma_{kl}$, and D_{ijkl} is the elastic compliance fourth-order tensor, p^{unit} is the 'unit' energy norm for normalization.

The relative stress norm by Equation 305.1 is more reasonable since two points having the same $\left\|\sigma_{ij} - \sigma^*_{ij}\right\|$ but different $\sigma^*_{pq}\sigma^*_{pq}$ should have different error measures. However, this norm becomes singular and possible meaningless when $\sigma^*_{pq}\sigma^*_{pq}$ close to zero. The normalized energy norm by Equation 305.2 have no such singularity problem but it may give the same error index for two points having the same $\left\|\sigma_{ij} - \sigma^*_{ij}\right\|$ but different $\sigma^*_{pq}\sigma^*_{pq}$. In this work, we use these two error measure methods, but for simplicity, Equation 305.2 is modified into

$$\delta^r = \frac{\sqrt{(\sigma_{ij} - \sigma^*_{ij})(\sigma_{ij} - \sigma^*_{ij})}}{\sqrt{\sigma^0_{pq}\sigma^0_{pq}}} \tag{305.3}$$

where $\sigma^0_{pq}\sigma^0_{pq}$ is evaluated at some non-zero initial isotropic stress state. That is, the normalized error is evaluated by Equation 305.3, and the relative error is evaluated by Equation 305.1.

In our examples, the initial stress state point is set $p^0 = 100$ kPa, $q^0 = 0$ kPa, $\theta^0 = 0$, which is the σ^0_{pq} in Equation 305.3. The one-step predicted stress state point for the implicit algorithm is within the range of $0.1 \leq p \leq 100$ kPa, $0 \leq q \leq 100$ kPa, $0 \leq \theta \leq \pi/3$. The 'exact' solution is actually unknown for most elastoplastic problems. Here the 'exact' solution is simply replaced by 50 substep solution of the explicit algorithm in the same one-step prediction incremental. All these error evaluations are within the material constitutive level.

The first test examples are von Mises models with the uniaxial yield strength $k = 50$ kPa, with linear elasticity parameters are Young's modulus $E = 1\times10^5$ kPa, and Poisson's ratio $\nu = 0.25$.

Figures 305.1 and 305.2 show the iso-error maps for the von Mises model with linear isotropic hardening. The linear hardening modulus $H = 2\times10^4$ kPa. The blue lines represents the yield surface boundary. It can be seen that the error magnitudes are as small as 10^{-10} to 10^{-9}, which implies that the solutions by implicit algorithm for this linear isotropic hardening von Mises model are numerically accurate.

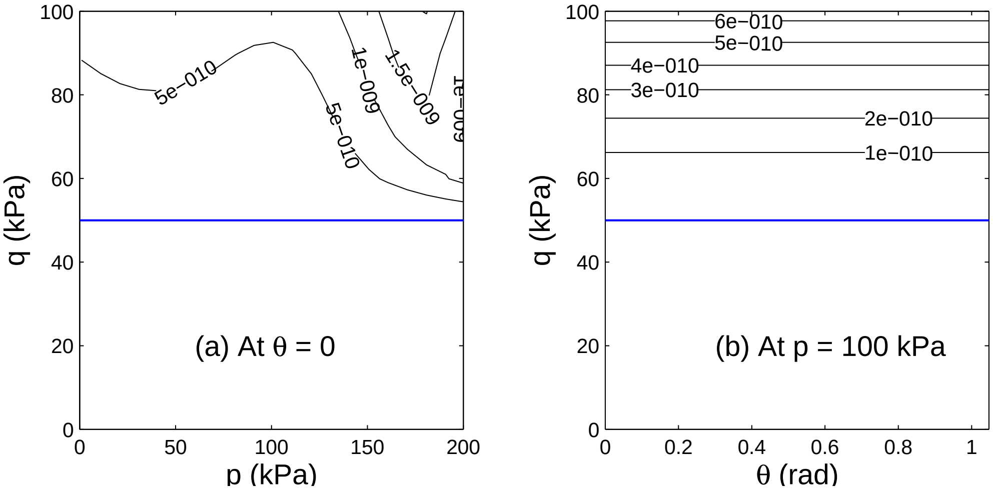


Figure 305.1: Normalized iso-error maps of von Mises model with linear isotropic hardening.

Figures 305.3 and 305.4 show the iso-error maps for the von Mises model with Armstrong-Frederick translational kinematic hardening. The hardening parameters are $h_a = 5\times10^4$ kPa and $C_r = 2.5\times10^3$. It can be seen that errors are very small which proves the good performance of the implicit algorithm. The iso-error map gives a good trend, i.e., the further away from the yield surface, the errors become more pronounced; the normalized errors are pressure-independent, which fits well the feature of von Mises model; the iso-error lines in the $q-\theta$ figure are parallel to the yield surface and are independent of the Lode's angle θ, which again fits well with von Mises model which is only q-related.

The second test examples are Drucker-Prager model with yield surface constant $q/p = 0.8$. Linear

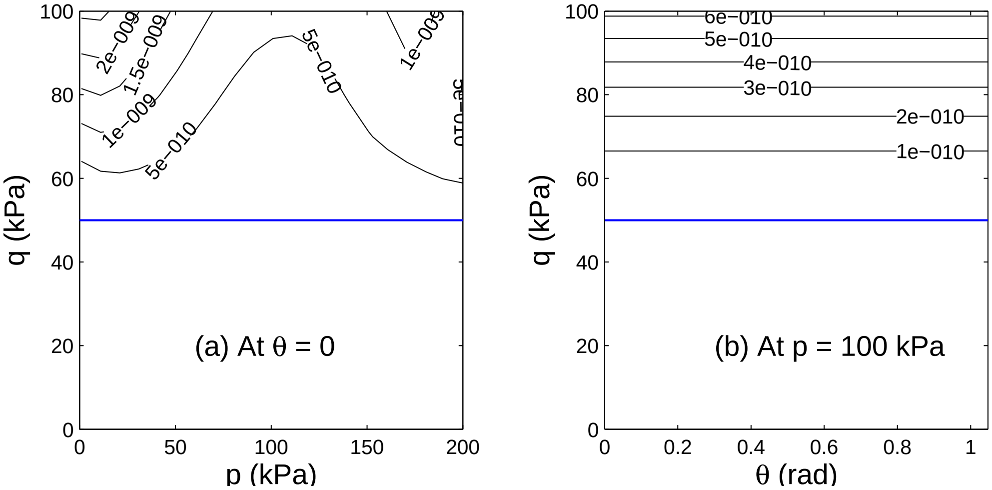


Figure 305.2: Relative iso-error maps of von Mises model with linear isotropic hardening.

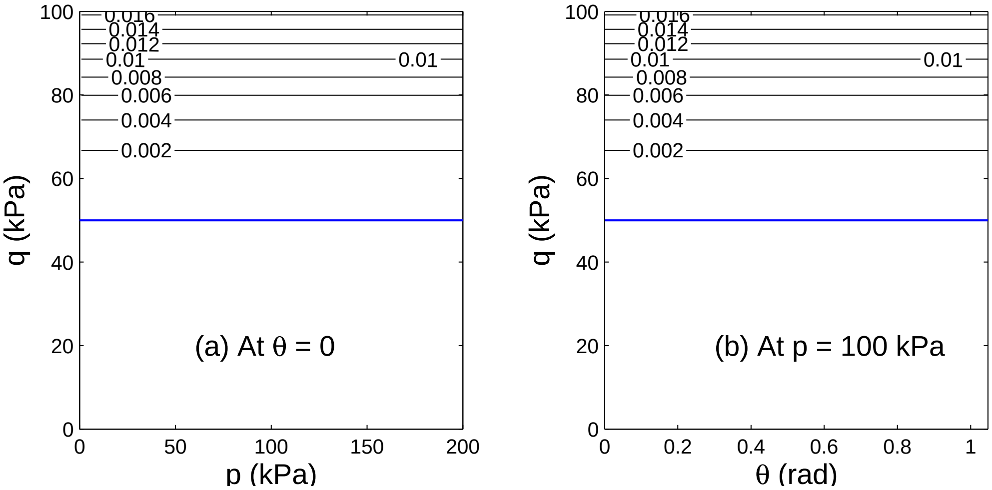


Figure 305.3: Normalized iso-error maps of von Mises model with Armstrong-Frederick kinematic hardening.

elasticity parameters are Young's modulus $E = 1\times10^5$ kPa, and Poisson's ratio $\nu = 0.25$.

The iso-error maps for perfectly plastic Drucker-Prager model are shown in Figures 305.5 and 305.6. The blue lines represents the yield surface boundary. It can be seen that the error magnitudes are as small as 10^{-11} to 10^{-9}. Again, these errors are so small that we can consider that the implicit algorithm give accurate solutions numerically.

Another Drucker-Prager model is with Armstrong-Frederick rotational kinematic hardening, and the parameters are $h_a = 20$, $C_r = 2$. The iso-error maps are shown in Figures 305.7 and 305.8. Unlike

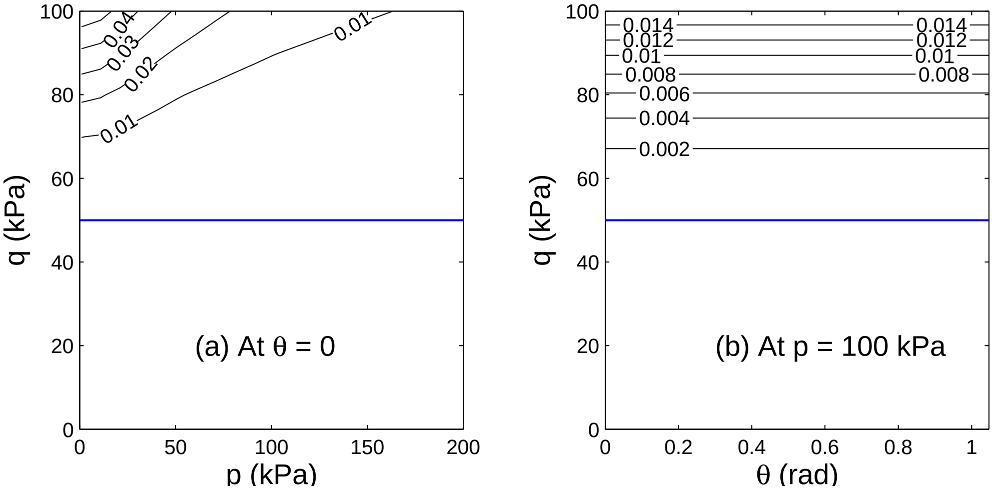


Figure 305.4: Relative iso-error maps of von Mises model with Armstrong-Frederick kinematic hardening.

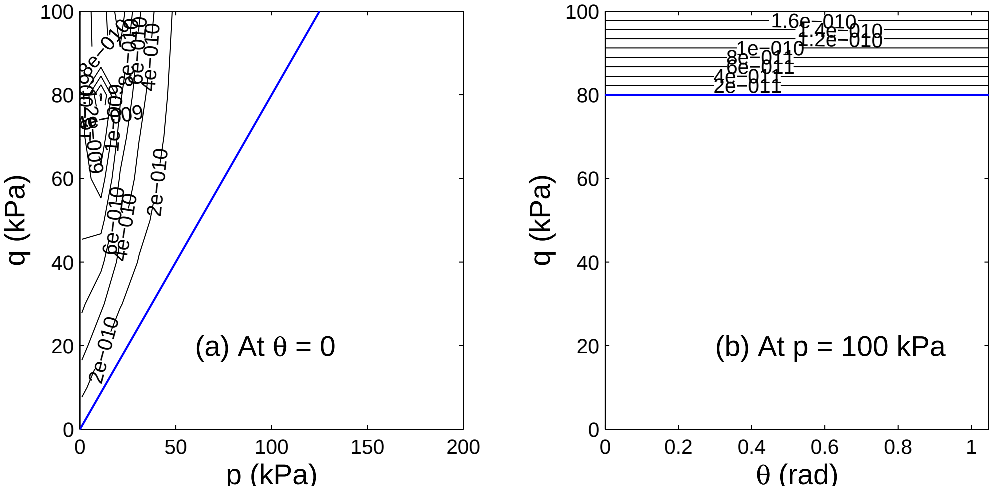


Figure 305.5: Normalized iso-error maps of Drucker-Prager perfectly plastic model.

von Mises model, the normalized errors are pressure-dependent, which fits well the feature of Drucker-Prager model; the iso-error lines in the $q - \theta$ figure are parallel to the yield surface and are independent of the Lode's angle θ, which still fits well with Drucker-Prager model which does not consider the third stress invariant, Lode's angle θ. From the relative iso-error maps in Figure 305.8, very dense iso-error lines are investigated in the region of small pressure, which is evidently due to the cone apex singularity of Drucker-Prager yield surface.

From the error analysis by the above von Mises and Drucker-Prager models, One finds that the implemented implicit algorithm can offer accurate solutions for simple models with simple hardening laws,

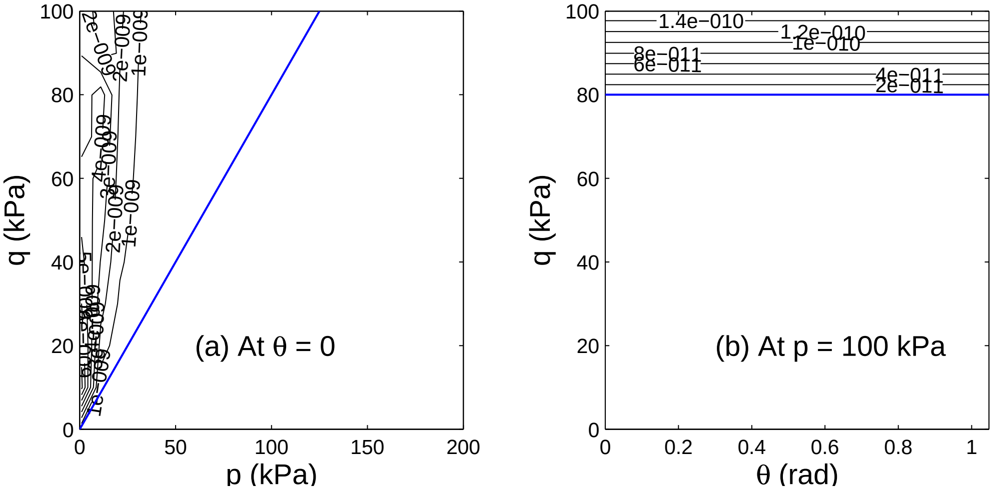


Figure 305.6: Relative iso-error maps of Drucker-Prager perfectly plastic model.

e.g. von Mises model with linear hardening and Drucker-Prager model with perfectly plastic hardening (no hardening). Complicated hardening laws increases the error even for simple plastic models, although the errors are still small. These observations match the well known conclusion that the error of the implicit algorithm is pretty dependent on the smoothness of the solution. The implemented implicit algorithm proves very robust for von Mises and Drucker-Prager model with simple or complicated hardening laws.

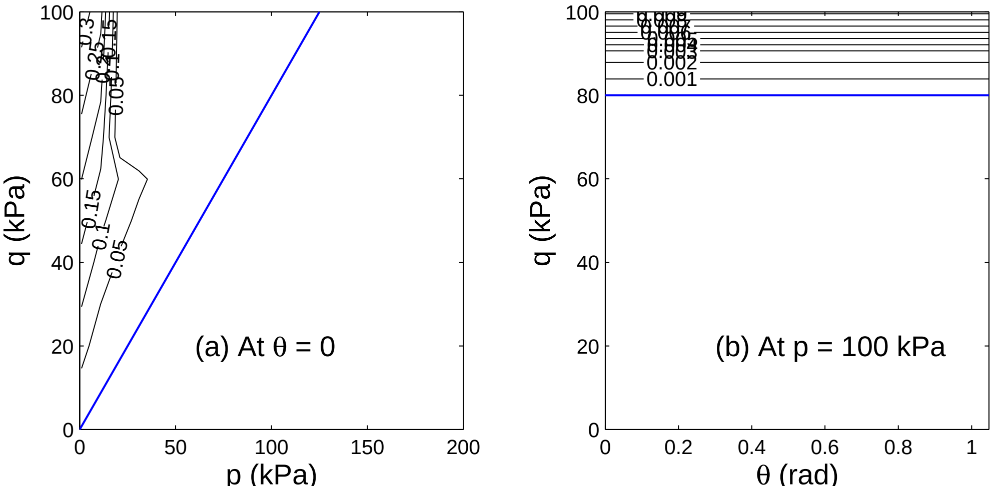


Figure 305.7: Normalized iso-error maps of Drucker-Prager model with Armstrong-Frederick kinematic hardening.

Figures 305.9 and 305.10 present the iso-error maps of Dafalias-Manzari model. The initial void ratio is 0.8, and the other parameters are from Dafalias and Manzari (2004a). The blue lines represents

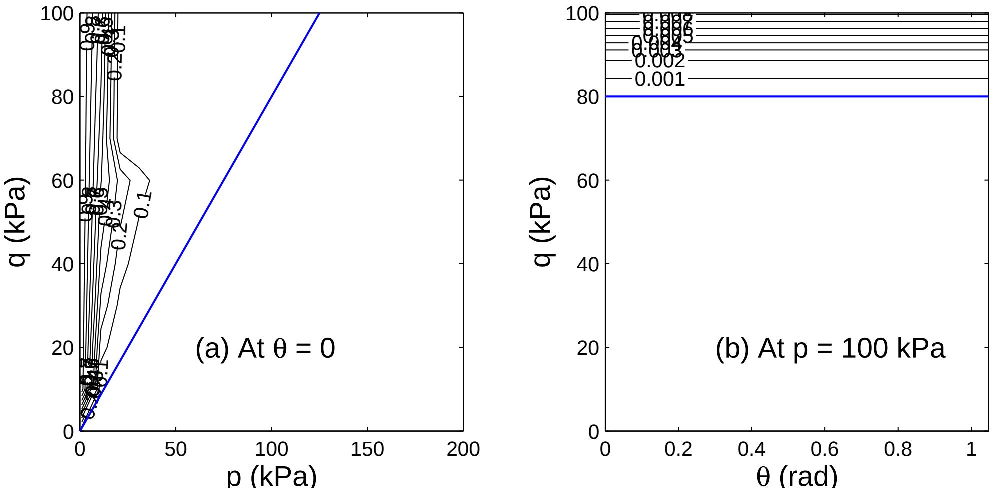


Figure 305.8: Relative iso-error maps of Drucker-Prager model with Armstrong-Frederick kinematic hardening.

the yield surface boundary (slope ratio $m = 0.01$). Unlike von Mises and Drucker-Prager models, the iso-error lines in the $q - \theta$ figure of Dafalias-Manzari model are not parallel to the yield surface and are dependent of the Lode's angle θ, which was one of the highlighting improvements upon the previous version (Manzari and Dafalias, 1997). From Figure 305.10, when the predicted stress q close to 100 kPa, or or about 100 times the yield strain increment, the relative errors can reach up to 100%, which implies that even for implicit algorithm, Dafalias-Manzari model still requires small strain increments. However, when $q < 30$ kPa, or about 30 times of the yield strain increment, the relative errors are less than 5%, excepts at the region close to the yield surface apex.

It should be pointed out that errors for the complex Dafalias-Manzari model are much bigger than those of simple models (e.g. von Mises and Drucker-Prager), due to its high non-linearity. However, if the predicted stress (or in other words, the strain increment) is small enough, the algorithm errors are within a small tolerant range.

Figures 305.9 and 305.10 are based on an approach of averaged elastic moduli. Instead, Figures 305.11 and 305.12 present iso-error maps based on constant elastic moduli approach.

305.2.2 Constitutive Level Convergence

In the implemented implicit algorithm, the iteration continues until the absolute value of yield function and the residue norm of considering variables are less than some small tolerances, or if by equations,

$$|f| \leq Tol1; \quad r_{norm} = \|\mathbf{r}\| \leq Tol2 \tag{305.4}$$

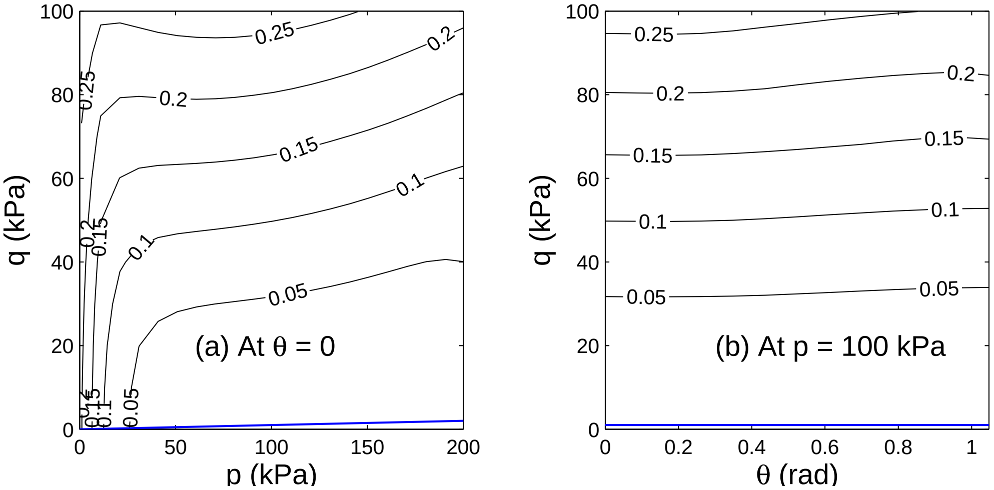


Figure 305.9: Normalized iso-error maps of Dafalias-Manzari model with average elastic moduli.

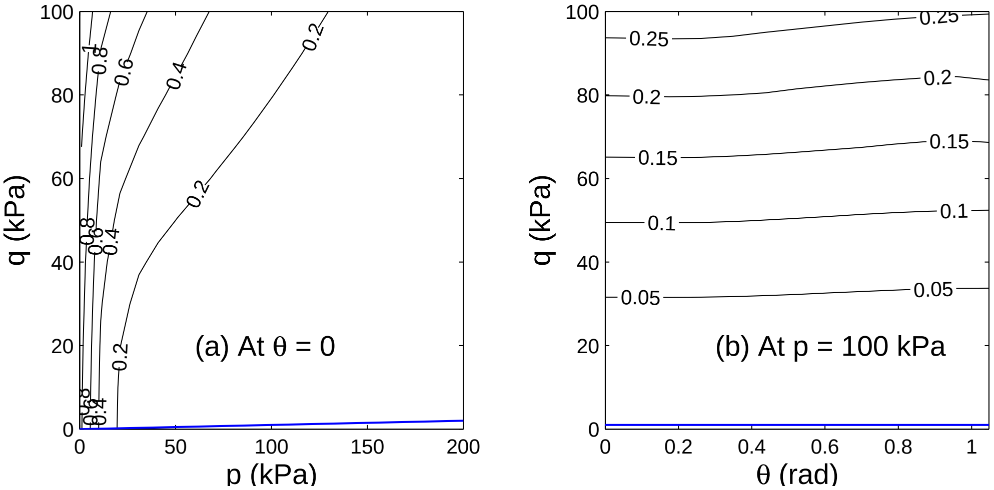


Figure 305.10: Relative iso-error maps of Dafalias-Manzari model with average elastic moduli.

Three examples including simple von Mises model with linear isotropic hardening, relative complicated Drucker-Prager model with Armstrong-Frederick kinematic hardening, and even more complicated Dafalias-Manzari model considering fabric dilation effect are presented here to show the constitutive level convergence performances for the implemented implicit algorithm. In all these examples, both $|f|$ and r_{norm} v.s. iteration numbers are plotted. Iteration number 0 represents the 'virtual' iteration number before return mapping implicit iteration cycle. $|f|$ at iteration number 0 thus means $|f|$ at the first predicted stress for each load increment; there is no value of r_{norm} at iteration number 0. A tolerance of $Tol1 = Tol2 = 1\times10^{-7}$ is for both $|f|$ and r_{norm}. The iteration stops when $|f| \leq Tol1$

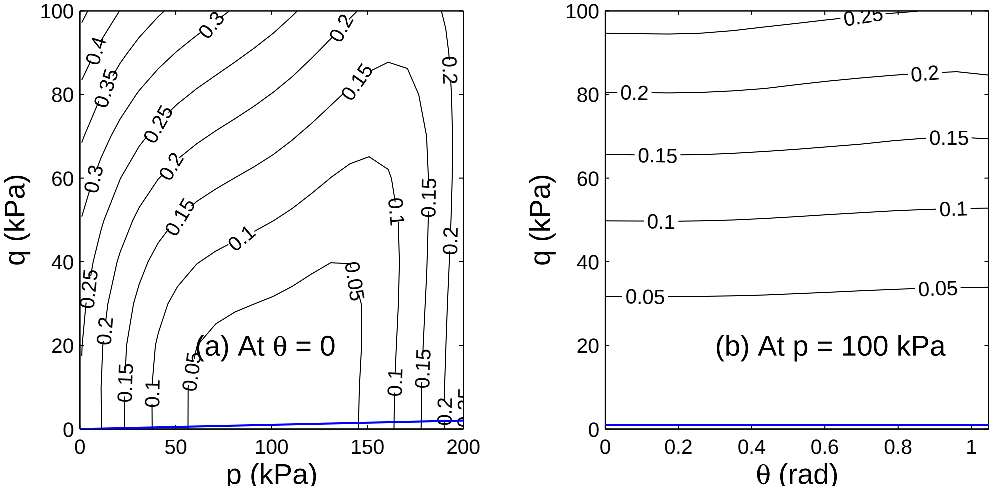


Figure 305.11: Normalized iso-error maps of Dafalias-Manzari model with constant elastic moduli.

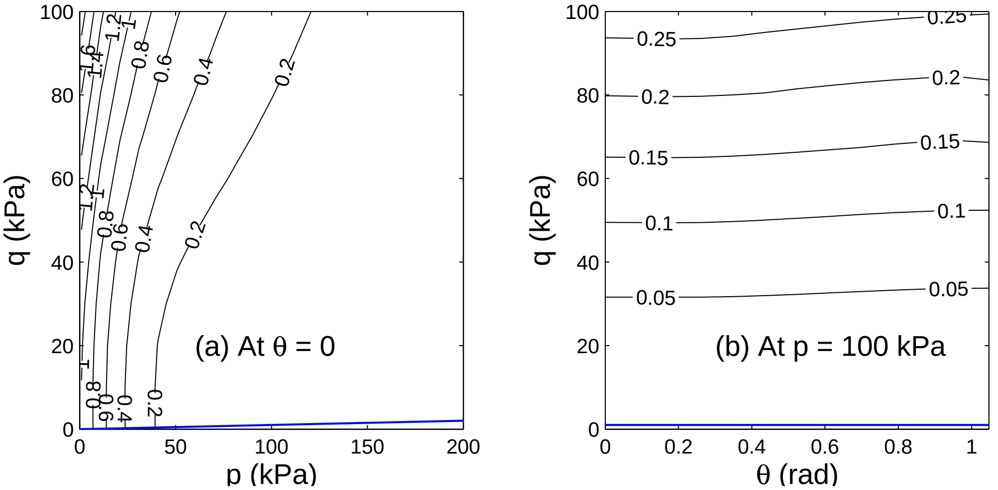


Figure 305.12: Relative iso-error maps of Dafalias-Manzari model with constant elastic moduli.

and $r_{norm} =\leq Tol2$ are satisfied, even if there is only one iteration number. The initial stress is an isotropic stress state of $p_0 = 100$ kPa. The undrained-like load increment is adopted by strain control as $\epsilon_{11} = -2\epsilon_{22} = -2\epsilon_{33} = n \times \Delta\epsilon$, where n is the load increment number and $\Delta\epsilon$ is the strain increment interval, ϵ_{ij} are strain components.

Figure 305.13 shows the typical constitutive level convergence performance for von Mises model with linear isotropic hardening. The input parameters are Young's Modulus $E = 1 \times 10^5$ kPa, Poisson's ratio $\nu = 0.25$, the material strength $k = 50$ kPa, and the linear isotropic hardening modulus $H = 2 \times 10^4$ kPa. The strain increment interval $\Delta\epsilon$ is set 2×10^{-4}. It can be seen that for this simple example, only two iteration steps are needed and $|f|$ and r_{norm} are far smaller than the tolerances and in fact close to the machine floating error value, or in other words, the stresses are exactly at the yield surface and the residue norm is zero.

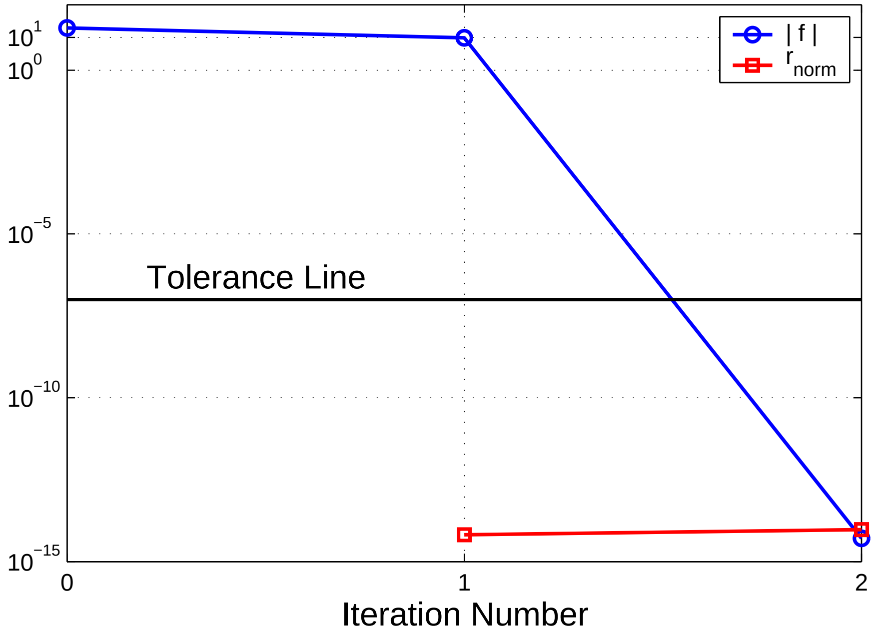


Figure 305.13: Typical convergence for von Mises model with linear isotropic hardening (tolerance value 1×10^{-7}).

Figure 305.14 shows the typical constitutive level convergence performance for Drucker-Prager model with Armstrong-Frederick kinematic hardening. The input parameters are Young's Modulus $E = 1 \times 10^{4}$ kPa, Poisson's ratio $\nu = 0.25$, the material q/p ratio is 0.8, and the Armstrong-Frederick parameters are $h_a = 20$, $C_r = 2$. The strain increment interval $\Delta\epsilon$ is set -2×10^{-4}. For this example, both $|f|$ and r_{norm} are stably decreasing with the increasing iteration number; However, $|f|$ and r_{norm} show different rates; $|f|$ needs 5 iteration steps while r_{norm} needs 7 iteration steps; The convergence rate of r_{norm} lags behind that of $|f|$.

Figure 305.15 shows the typical constitutive level convergence performance for the complicated Dafalias-Manzari model considering fabric dilation effect. The input parameters are as in Table 305.1, and the initial void ration is set as 0.8. Different from the above examples, The strain increment interval $\Delta\epsilon$ is set a much smaller value of -1×10^{-5}. In this example, again, both $|f|$ and r_{norm} are stably decreasing with the increasing iteration number; However, $|f|$ and r_{norm} show different rates; $|f|$ needs less iteration steps than r_{norm}; The convergence rate of r_{norm} lags behind that of $|f|$. It should be mentioned here for this complicated Dafalias-Manzari model considering fabric dilation effect,

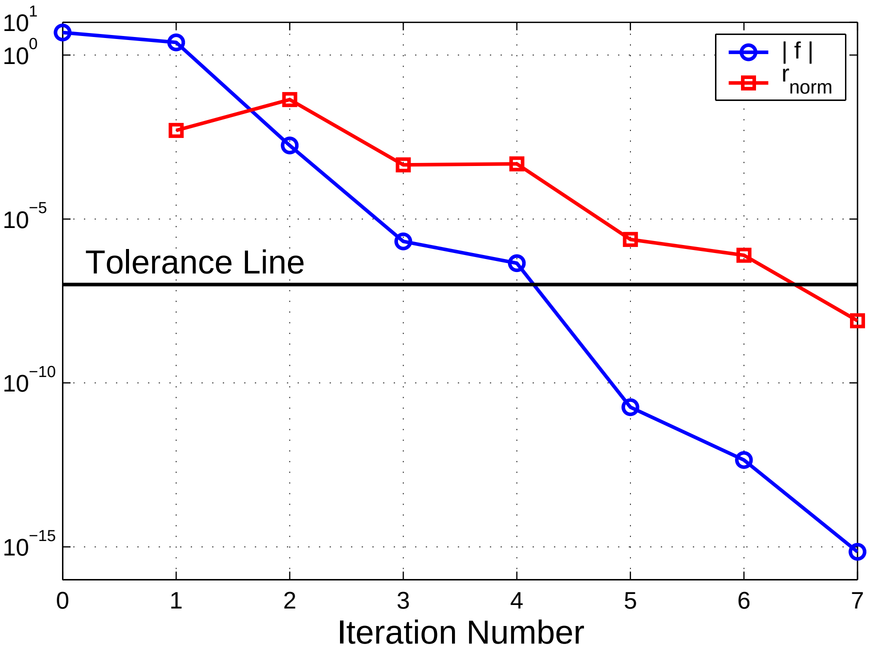


Figure 305.14: Typical convergence for Drucker-Prager model with Armstrong-Frederick kinematic hardening (tolerance value 1×10^{-7}).

the typical constitutive level convergence performance is similar to that of Drucker-Prager model with Armstrong-Frederick kinematic hardening, but with much smaller strain increment interval.

From the above examples, it is clear that the simpler the model is , the better constitutive level convergence performances are observed. This is consistent to the error assessment in section 305.2.1. Generally, the implemented implicit algorithm shows stable constitutive level convergence performances provided an appropriate small strain increment interval for the material model.

305.3 Validation of Constitutive Model Predictions

305.3.1 Dafalias Manzari Material Model

Validation is performed by comparing experimental (physical) results and numerical (constitutive) simulations for the Toyoura sand. It should be noted that we have not done validation against 2D or 3D tests (say centrifuge tests) as characterization of sand used in centrifuge experiments is usually less than complete for use with advanced constitutive models. Moreover, as our approach seeks to make predictions of prototype behavior, scaling down models (and using them for comparison with numerical predictions) brings forward issues of physics of scaling which we would rather stay out of. The material parameters

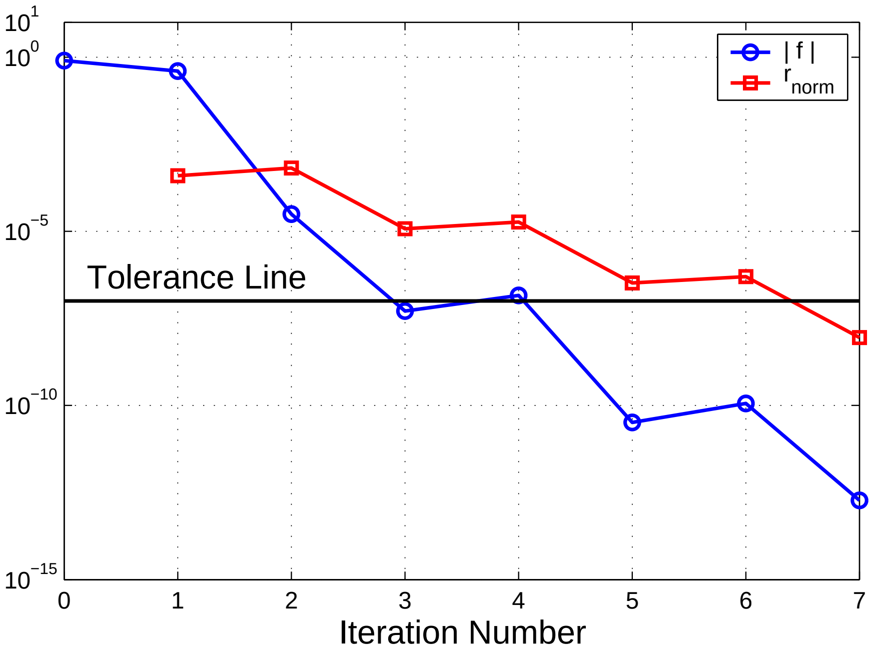


Figure 305.15: Typical convergence for Dafalias-Manzari model (tolerance value 1×10^{-7}).

used are from Dafalias and Manzari (2004a) and are listed in Table (305.1). Several simulated results are compared with the experimental data published by Verdugo and Ishihara (1996).

Figure (305.16) presents both loading and unloading triaxial drained test simulation results for a relatively low triaxial isotropic pressure of 100 kPa but with different void ratios of $e_0 = 0.831, 0.917, 0.996$ at the end of isotropic compression. During the loading stage, one can observes the hardening and then softening together with the contraction and then dilation for the denser sand, while only hardening together with contraction for the looser sand. The significance of the state parameter to the soil modeling is clear from the very different responses with different void ratios at the same triaxial isotropic pressure.

Figure (305.16) also shows both loading and unloading triaxial drained test simulation results for a relatively high triaxial isotropic pressure of 500 kPa but with different void ratio of e_0=0.810, 0.886, 0.960 at the end of isotropic compression. Similar phenomenon are observed as with tests (physical and numerical) for relatively low triaxial isotropic pressure. However, due to the higher confinement pressure, the stress-strain responses are higher at the same strain, which proves the significant pressure dependent feature for the sands.

Figure (305.17) presents both loading and unloading triaxial undrained test simulation results for a dense sand with the void ratio of e_0=0.735 at the end of isotropic compression but with different

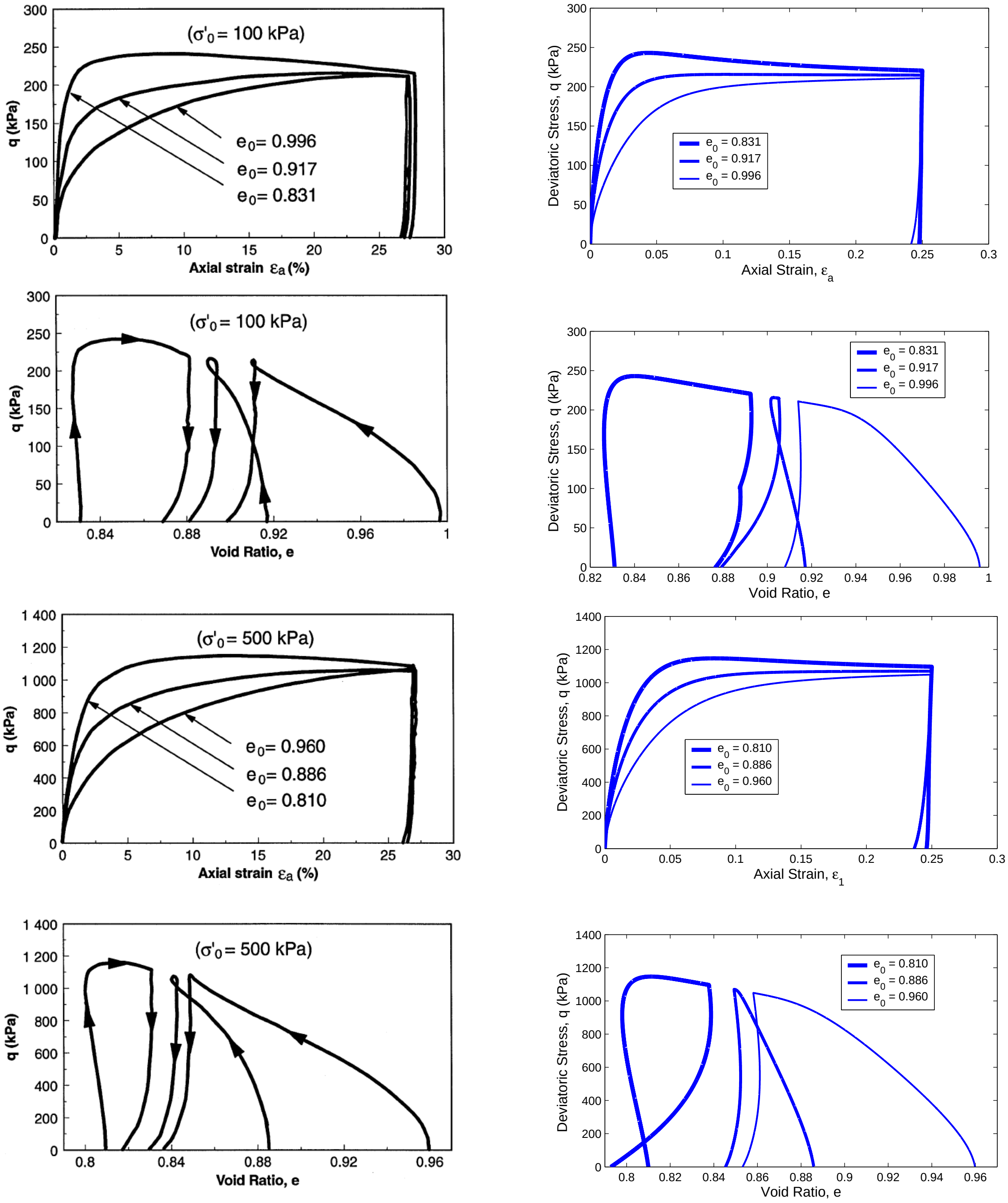


Figure 305.16: Left: Experimental data; Right: Simulated results.

Table 305.1: Material parameters of Dafalias-Manzari model.

material parameter		value	material parameter		value
Elasticity	G_0	125 kPa	Plastic modulus	h_0	7.05
	v	0.05		c_h	0.968
Critical sate	M	1.25		n_b	1.1
	c	0.712	Dilatancy	A_0	0.704
	λ_c	0.019		n_d	3.5
	ξ	0.7	Fabric-dilatancy	z_{max}	4.0
	e_r	0.934		c_z	600.0
Yield surface	m	0.01			

isotropic compression pressures of $p_0 = 100, 1000, 2000, 3000$ kPa. During the loading stage, one observes that each of responses are close to the critical state line for the very various range of isotropic compression pressures. For the higher isotropic compression pressure, the contraction response with softening is clearly observed, while for the smaller isotropic compression pressure, it is a dilation response without softening.

Close matching of physical testing data with constitutive predictions represents a satisfactory validation of our material model. This validation with previous verification gives us confidence that predictions (presented in next section) represent well the real, prototype behavior.

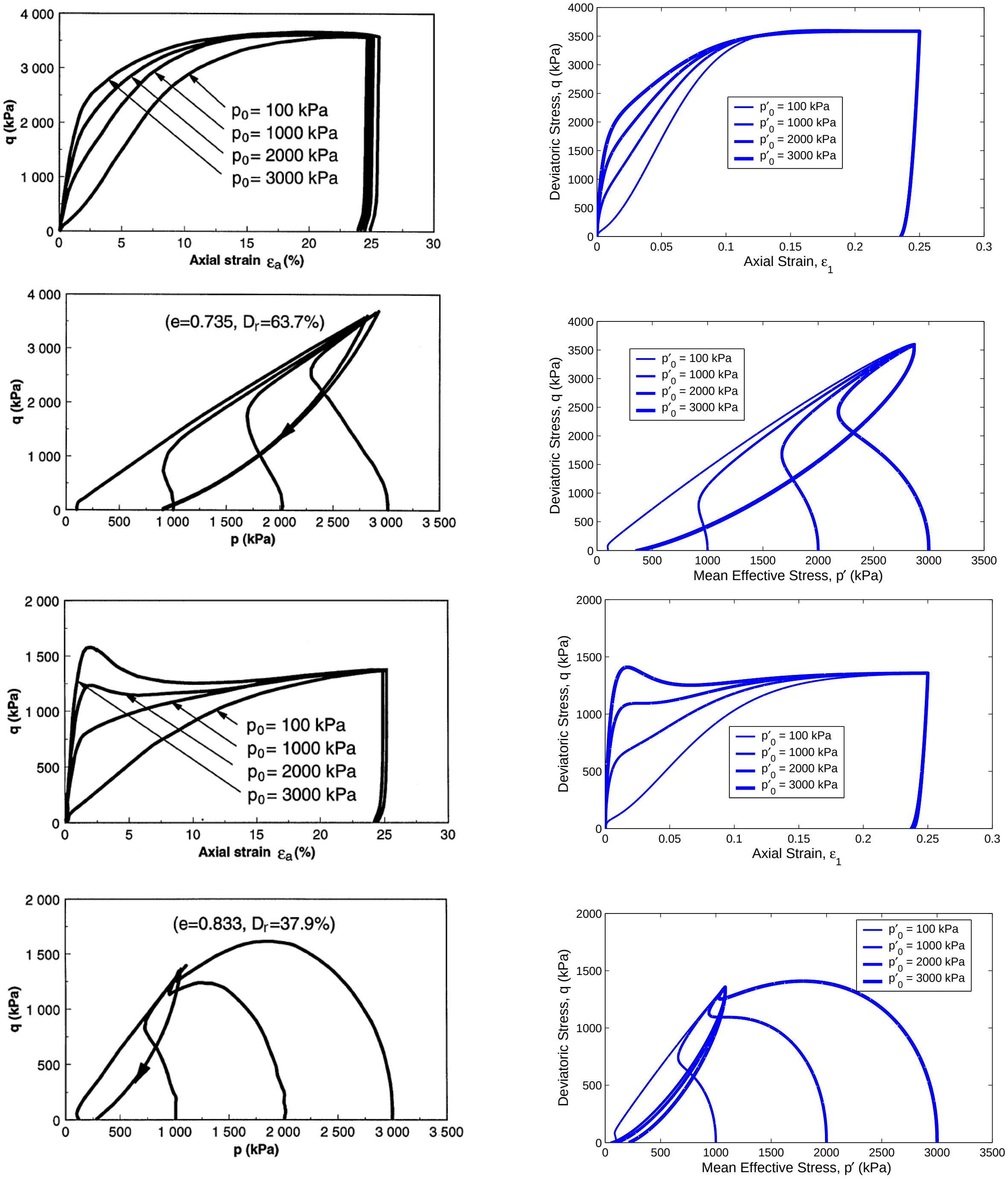


Figure 305.17: Left: Experimental data; Right: Simulated results.

Chapter 306

Verification and Validation for Static and Dynamic Finite Element Level Solution Advancement Algorithms

(1994-2003-2009-2012-2015-2017-2019-)

(In collaboration with Prof. José Antonio Abell Mena and Dr. Yuan Feng)

306.1 Chapter Summary and Highlights

306.2 Verification for Static Solution Advancement

306.3 Verification for Dynamic Solution Advancement

Spectral Radii, Argyris and Mlejnek (1991)...

- Test consists of a 2-DOF system periods $T_1 = 4.0\,\mathrm{s}$ and $T_2 = 1.0\,\mathrm{s}$.
- Different integration steps.
- Different values of method parameters (α, β, γ).
- Results are compared with theoretical predictions for algorithmic damping and period shifts.

306.3.1 Verification for Dynamic Solution Advancement, Newmark Method

(Newmark, 1959)

Iff

$$\gamma \geq \frac{1}{2}, \quad \beta \geq \frac{1}{4}(\gamma + \frac{1}{2})^2 \tag{306.1}$$

the procedure is unconditionally stable and second-order accurate.

Different values for γ and β can be used to create various integration methods:

- For $\gamma = 0.5$ (and corresponding $\beta = 0.25$) there is no numerical damping.
- Any γ value greater than 0.5 will introduce numerical damping.
- Trapezoidal rule or average acceleration method for $\beta = 1/4$ and $\gamma = 1/2$,
- Linear acceleration method for $\beta = 1/6$ and $\gamma = 1/2$,
- Explicit, central difference method for $\beta = 0$ and $\gamma = 1/2$.
- Strongest numerical damping values is obtained for values $\beta = 1$ and $\gamma = 2/3$, as spectral ratio $\rho_\infty = 0$ (Hughes (1987), page 502)

For more details see chapter 108 on page 537.

Following Argyris and Mlejnek (1991); Hughes (1987), to calculate the analytic ξ and analytic $\bar{\omega}$ a matrix A is constructed. The explicit definition of amplification matrix A for the Hilber Hughes Taylor (HHT) family of algorithms (where Newmark is obtained by setting $\gamma = 0$) is

$$A = \frac{1}{D}\begin{bmatrix} 1+\alpha\beta\Omega^2 & 1 & \frac{1}{2}-\beta \\ -\gamma\Omega^2 & 1-(1+\alpha)(\gamma-\beta)\Omega^2 & 1-\gamma-(1+\alpha)(\frac{1}{2}\gamma-\beta)\Omega^2 \\ -\Omega^2 & -(1+\alpha)\Omega^2 & -(1+\alpha)(\frac{1}{2}-\beta)\Omega^2 \end{bmatrix} \tag{306.2}$$

where

$$\begin{aligned} D =& 1+(1+\alpha)\beta\Omega^2 \\ \Omega =& \omega\triangle t \\ \omega =& (K/M)^{\frac{1}{2}} \end{aligned} \tag{306.3}$$

The eigenvalue of amplification matrix A will be two complex conjugate roots $\lambda_{1,2}$ and a so-called spurious root λ_3 which satisfy $|\lambda_3| < |\lambda_{1,2}| \leq 1$. The roots $\lambda_{1,2}$ will be

$$\lambda_{1,2} = A \pm Bi \tag{306.4}$$

Then, the analytic damping ratio ξ and analytic period $\bar{\omega}$ becomes

$$\begin{aligned} \bar{\xi} =& -ln(A^2+B^2) \\ \bar{\omega} =& \bar{\Omega}/\triangle t \\ \bar{\Omega} =& arctan(B/A) \end{aligned} \tag{306.5}$$

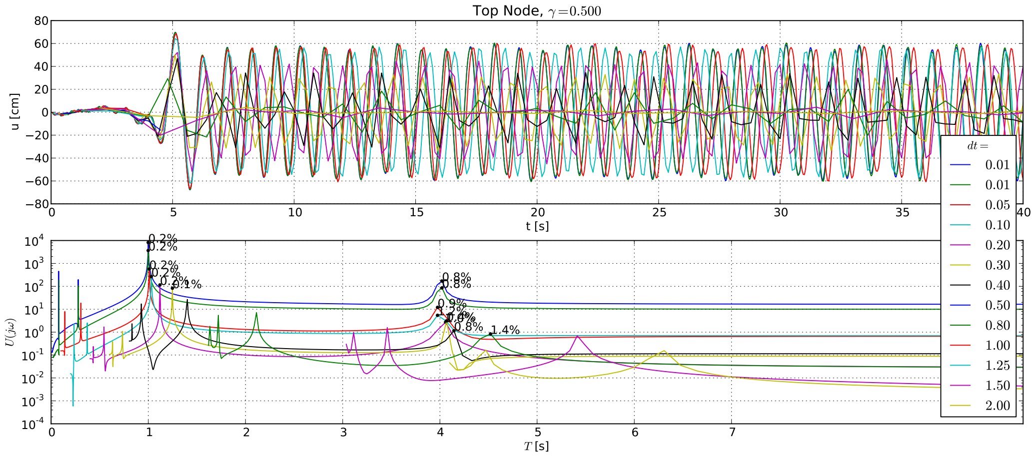


Figure 306.1: Verification: Dynamic solution advancement, Newmark method (p2a-newmark-05top)...

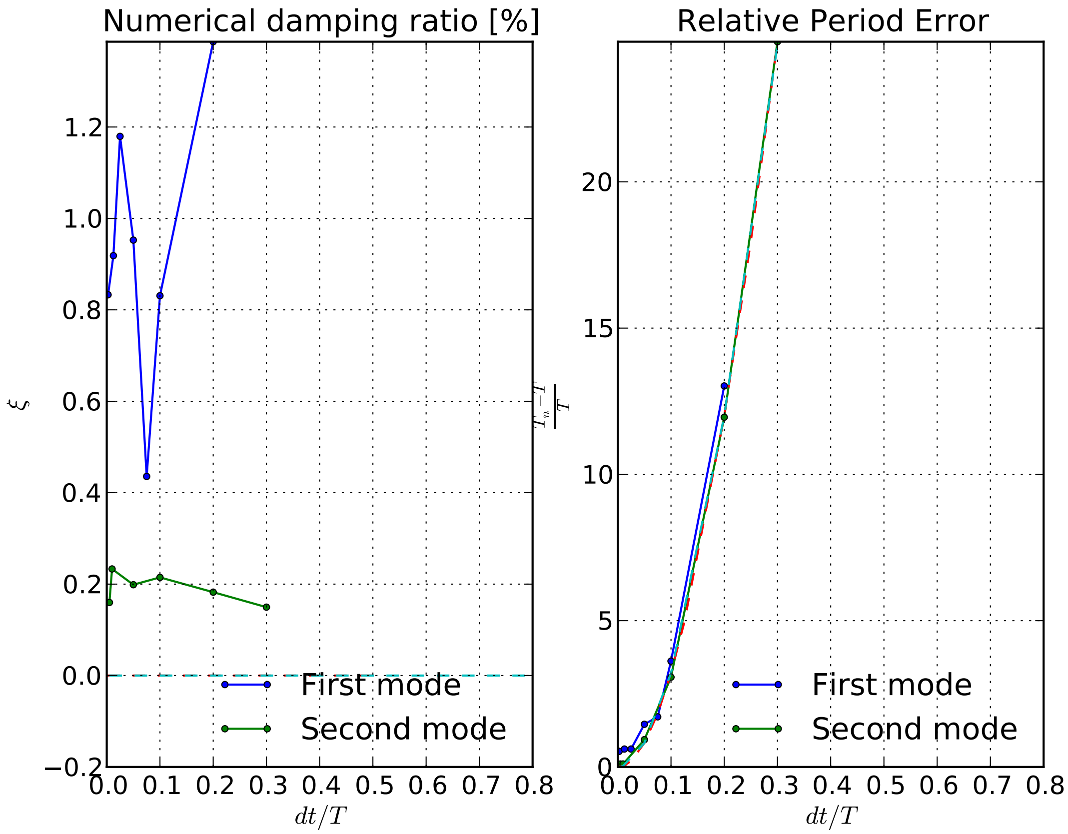


Figure 306.2: Verification: Dynamic solution advancement, Newmark method (p2a-newmark-05errors)...

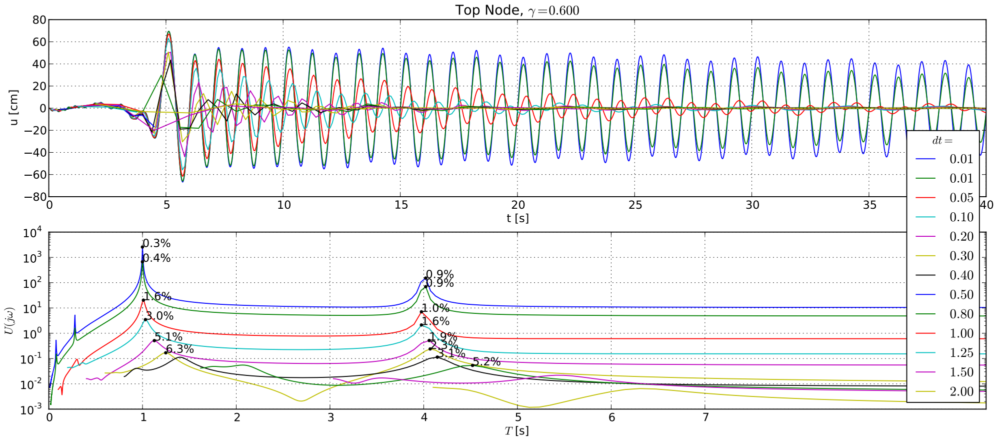


Figure 306.3: Verification: Dynamic solution advancement, Newmark method (p2a-newmark-06top)...

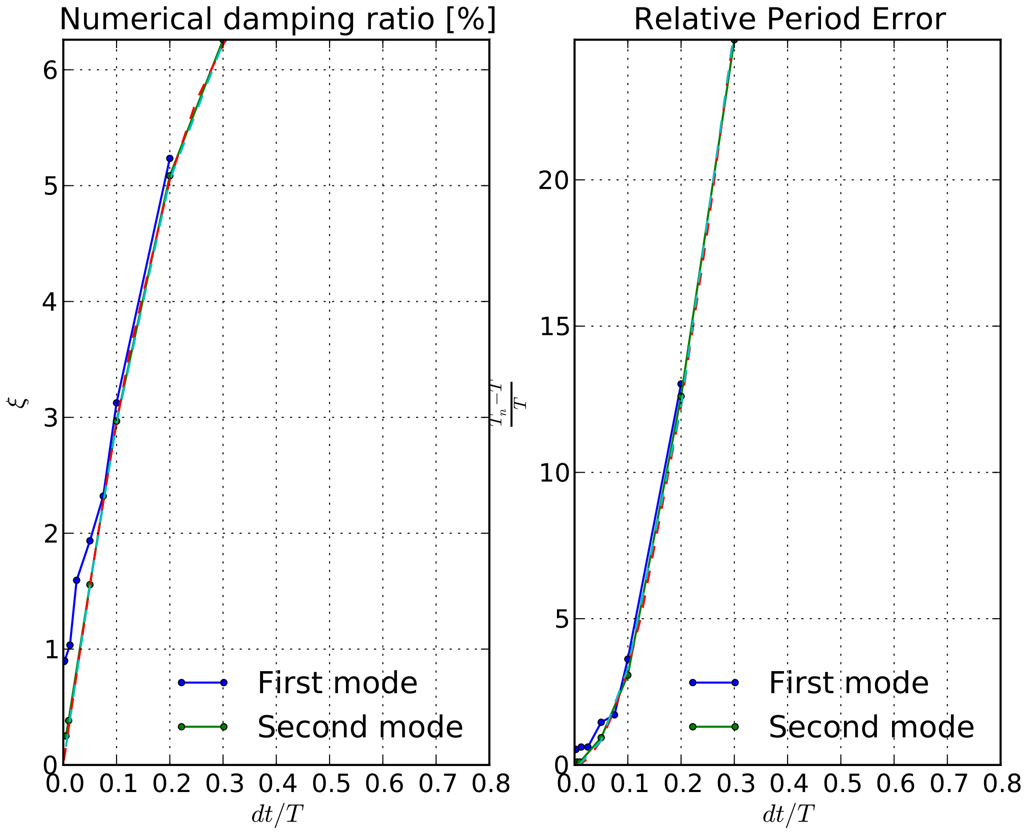


Figure 306.4: Verification: Dynamic solution advancement, Newmark method (p2a-newmark-05errors)...

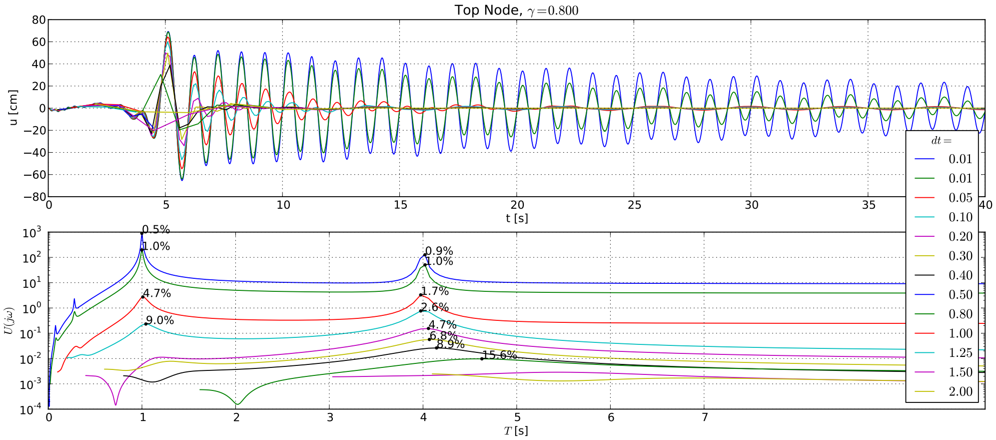


Figure 306.5: Verification: Dynamic solution advancement, Newmark method (p2a-newmark-08top)...

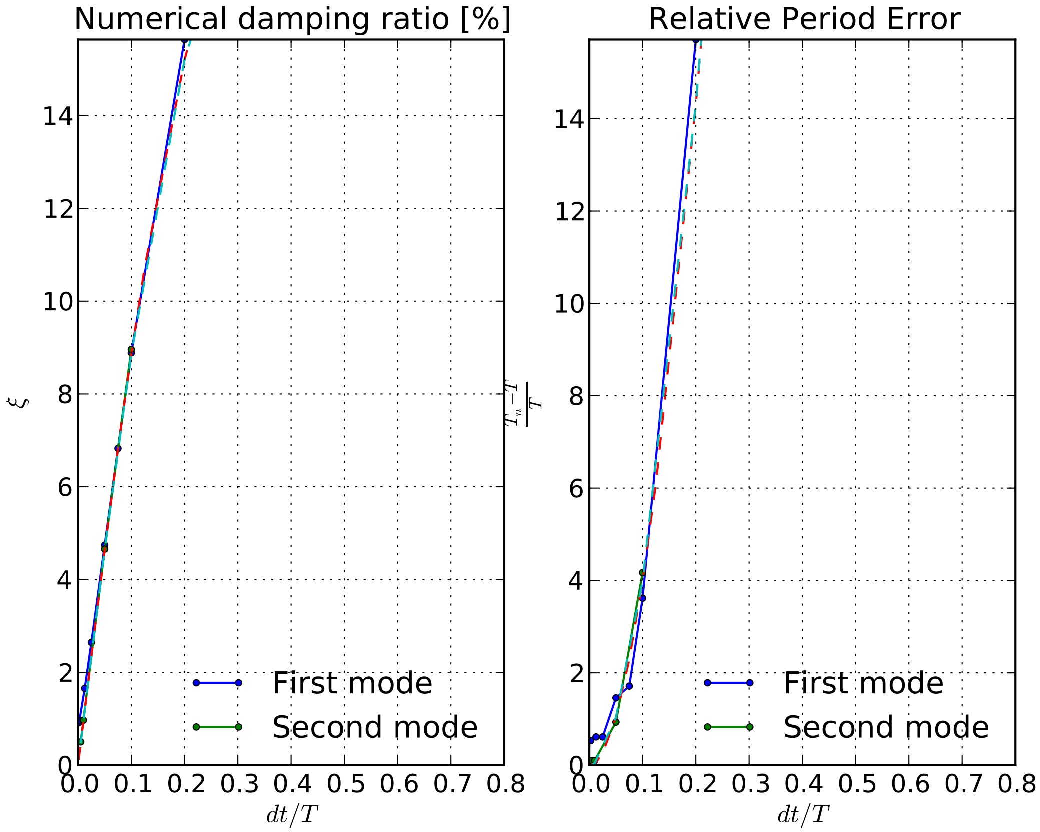


Figure 306.6: Verification: Dynamic solution advancement, Newmark method (p2a-newmark-08errors)...

```
dt gamma damping[\%] Th. damp.[\%] T shift [\%] Th. T. shift[\%]
==========================================================================
0.0100 1.5000 3.398552 3.139529 0.00000000 -0.13145647
0.0050 0.5000 0.295322 -0.000000 0.00000000 -0.00822413
0.0500 0.5000 0.368976 0.000000 -1.01010101 -0.81712426
0.1250 0.5000 0.027518 0.000000 -4.95382032 -4.94456582
0.0050 0.5100 0.298179 0.015705 0.00000000 -0.00822660
0.0500 0.5100 0.395657 0.154550 -1.01010101 -0.81736312
0.1250 0.5100 0.358214 0.357055 -4.95382032 -4.94584070
0.0050 0.6000 0.346769 0.157054 0.00000000 -0.00847079
0.0500 0.6000 1.600134 1.545504 -1.01010101 -0.84101009
0.1250 0.6000 3.571434 3.570958 -5.04201681 -5.07208349
0.0050 0.7000 0.418286 0.314108 0.00000000 -0.00921077
0.0500 0.7000 3.113067 3.091044 -1.01010101 -0.91266978
0.1250 0.7000 7.130075 7.144404 -5.13036165 -5.45499548
0.0050 0.9000 0.625858 0.628215 0.00000000 -0.01217067
0.0500 0.9000 6.177709 6.182372 -1.01010101 -1.19934156
```

306.3.2 Verification Example Description.

A one degree of freedom (DOF) example was made to verify the Newmark and HHT algorithm for Real-ESSI simulator. The example was plot below in Fig.(306.7). The beam stiffness and the mass were designed to make the natural period to be 1 second. In the first loading stage, the beam was given a horizontal force to generate an initial displacement. By the way, at the top node, all DOFs were fixed except the DOF along initial displacement. Then, in the second loading stage, the beam start free vibration.

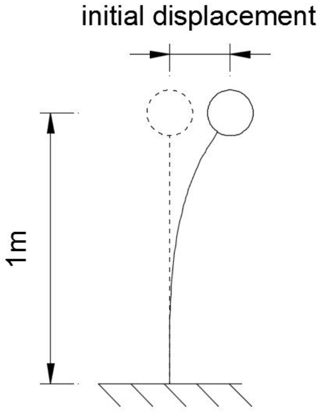


Figure 306.7: Verification example description.

The results were listed and plotted below.

Table 306.1: Verification results for the Newmark solution advancement algorithm.

dt	γ	Measured ξ	Analytic ξ	Measured T shift	Analytic T shift
0.005	0.5	0.0030	0.0000	0.0000	0.0001
0.01	0.5	0.0030	0.0000	0.0000	0.0003
0.05	0.5	0.0037	0.0000	0.0101	0.0082
0.1	0.5	0.0034	0.0000	0.0309	0.0321
0.005	0.6	0.0035	0.0016	0.0000	0.0001
0.01	0.6	0.0042	0.0031	0.0000	0.0003
0.05	0.6	0.0160	0.0155	0.0101	0.0084
0.1	0.6	0.0296	0.0295	0.0309	0.0329
0.005	0.7	0.0042	0.0031	0.0000	0.0001
0.01	0.7	0.0063	0.0063	0.0000	0.0004
0.05	0.7	0.0311	0.0309	0.0101	0.0091
0.1	0.7	0.0590	0.0590	0.0309	0.0356
0.005	0.8	0.0051	0.0047	0.0000	0.0001
0.01	0.8	0.0093	0.0094	0.0000	0.0004
0.05	0.8	0.0465	0.0464	0.0101	0.0103
0.1	0.8	0.0882	0.0886	0.0309	0.0399
0.005	0.9	0.0063	0.0063	0.0000	0.0001
0.01	0.9	0.0130	0.0126	0.0000	0.0005
0.05	0.9	0.0618	0.0618	0.0101	0.0120
0.1	0.9	0.1180	0.1181	0.0417	0.0460

Verification Results for Newmark Solution Advancement Algorithm. The Real-ESSI model fei/DSL files for the results above can be downloaded HERE.

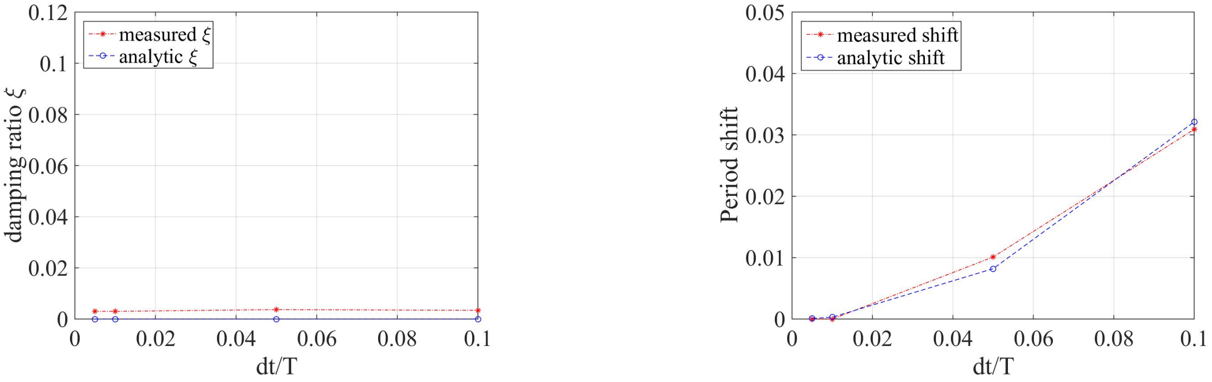


Figure 306.8: Comparison for Newmark algorithm with $\gamma = 0.5$. Damping ratio comparison, Period shift comparison.

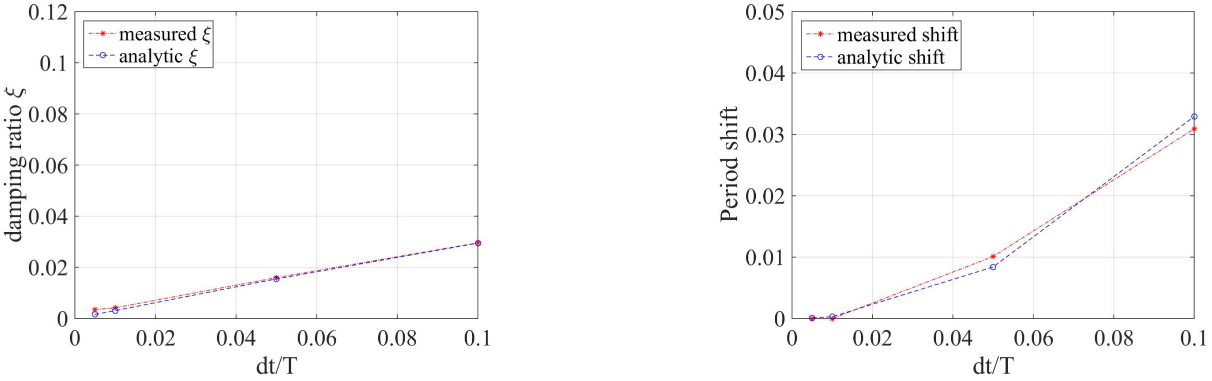


Figure 306.9: Comparison for Newmark algorithm with $\gamma = 0.6$. Damping ratio comparison, Period shift comparison.

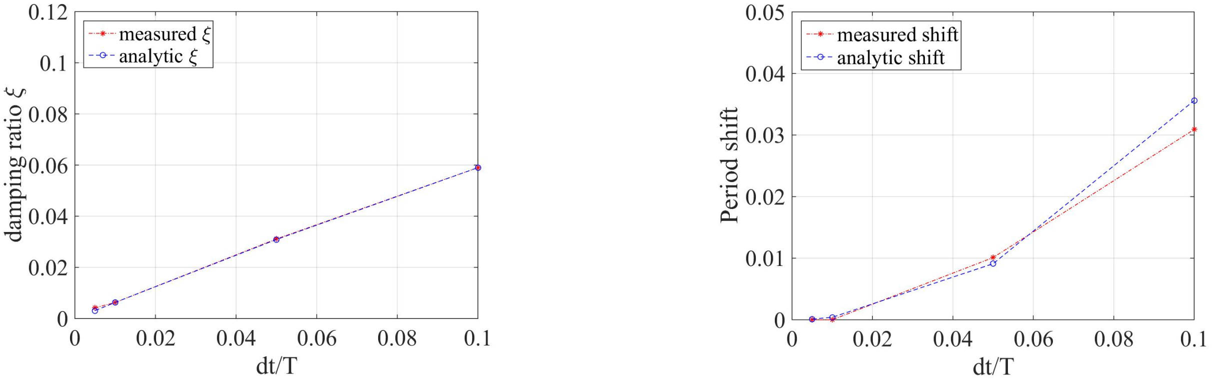


Figure 306.10: Comparison for Newmark algorithm with $\gamma = 0.7$. Damping ratio comparison, Period shift comparison.

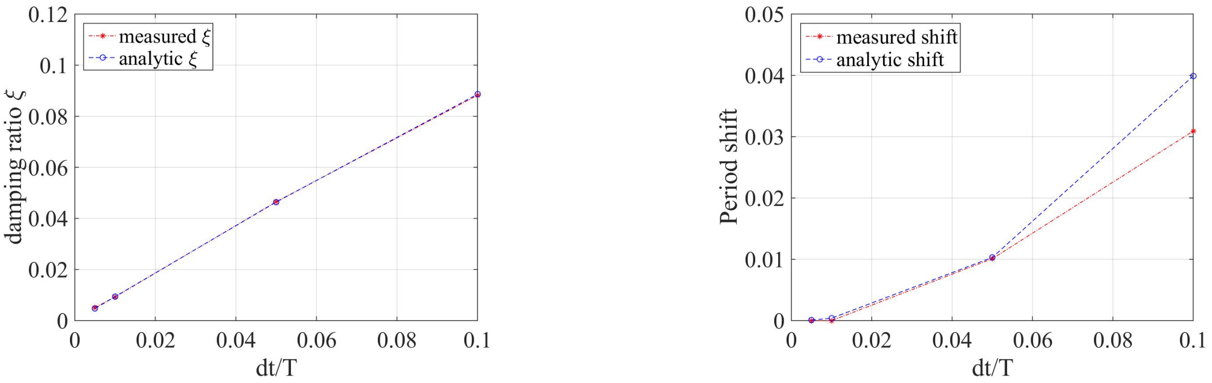


Figure 306.11: Comparison for Newmark algorithm with $\gamma = 0.8$. Damping ratio comparison, Period shift comparison.

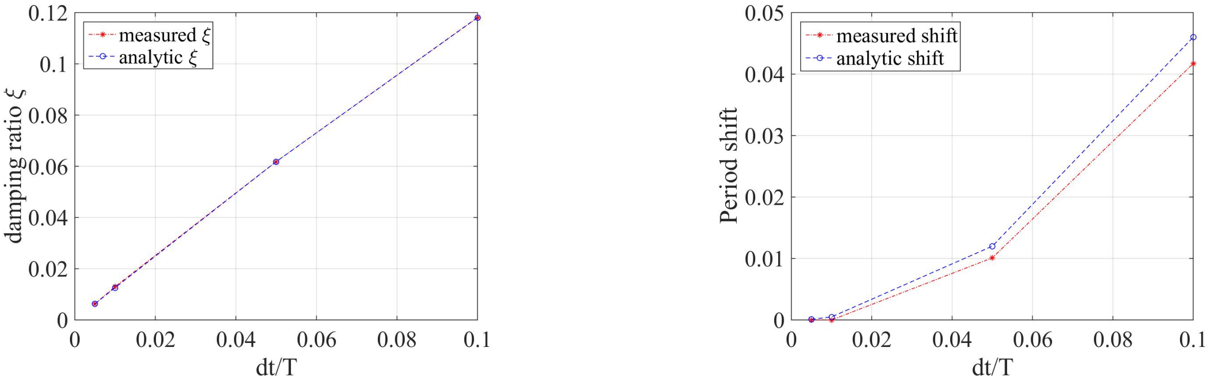


Figure 306.12: Comparison for Newmark algorithm with $\gamma = 0.9$. Damping ratio comparison, Period shift comparison.

306.3.3 Verification for Dynamic Solution Advancement, Hilber-Hughes-Taylor Method

(Hilber et al., 1977), (Hughes and Liu, 1978a) and (Hughes and Liu, 1978b)

If the parameters α, β and γ satisfy

$$-1/3 \leq \alpha \leq 0, \quad \gamma = \frac{1}{2}(1-2\alpha), \quad \beta = \frac{1}{4}(1-\alpha)^2 \tag{306.6}$$

it is unconditionally stable and second-order accurate (Argyris and Mlejnek, 1991; Hughes, 1987).

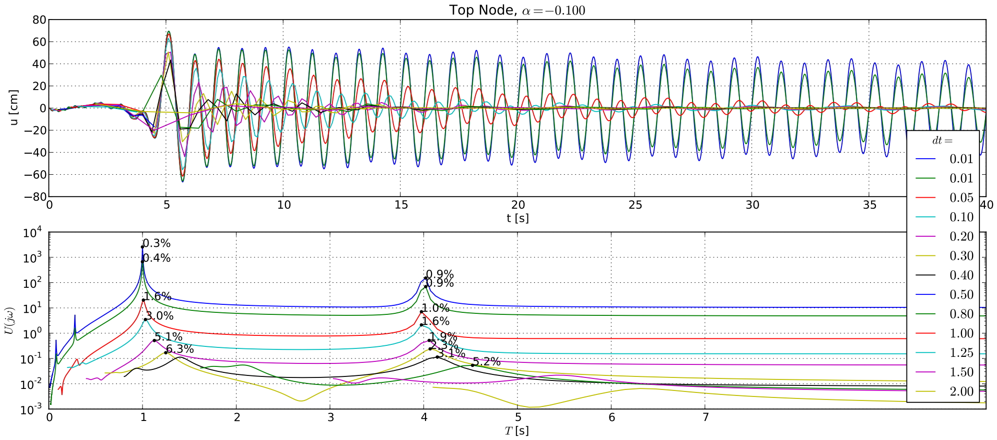


Figure 306.13: Verification: Dynamic solution advancement, Hilber-Hughes-Taylor method (p2a-hht-01top)...

Summary

```
dt alpha damping[%] Th. damp.[%] T shift [%] Th. T. shift[%]
=============================================================================================
      0.0050 -0.0000 0.295322 -0.000000 0.00000000 -0.00822413
      0.0100 -0.0000 0.302864 0.000000 0.00000000 -0.03289003
      0.0500 -0.0000 0.368976 0.000000 -1.01010101 -0.81712426
      0.1000 -0.0000 0.339004 0.000000 -3.09278351 -3.20749106
      0.2500 -0.0000 0.458456 -0.000000 -17.64705882 -17.96772753
      0.0050 -0.0100 0.295394 0.000004 0.00000000 -0.00846709
      0.0100 -0.0100 0.303170 0.000030 0.00000000 -0.03386091
      0.0500 -0.0100 0.357316 0.003645 -1.01010101 -0.84065236
      0.1000 -0.0100 0.376804 0.025924 -3.09278351 -3.29331088
      0.2500 -0.0100 1.026595 0.213468 -17.64705882 -18.29362011
      0.0050 -0.1000 0.295948 0.000031 0.00000000 -0.01032073
      0.0100 -0.1000 0.305522 0.000251 0.00000000 -0.04126821
      0.0500 -0.1000 0.305401 0.029968 -1.01010101 -1.02022480
      0.1000 -0.1000 0.432090 0.210613 -4.16666667 -3.95057551
      0.2500 -0.1000 1.807861 1.706433 -20.48192771 -20.93638414
      0.0050 -0.2000 0.296354 0.000050 0.00000000 -0.01167737
      0.0100 -0.2000 0.307270 0.000396 0.00000000 -0.04668953
```

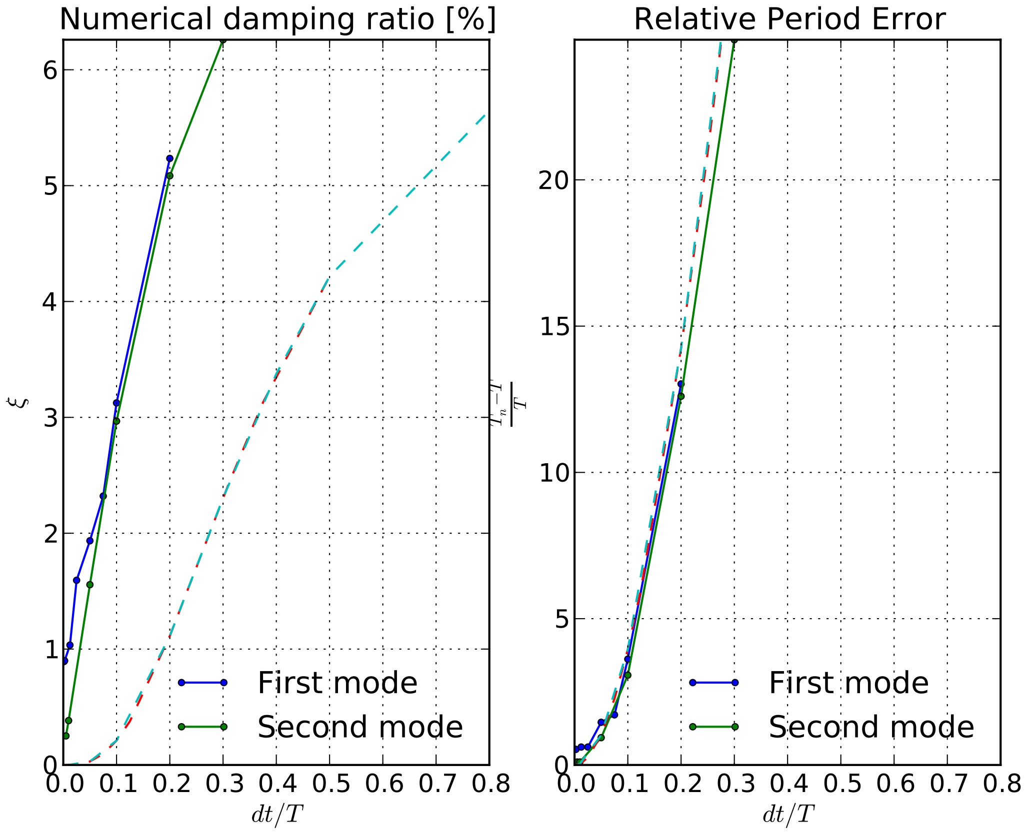


Figure 306.14: Verification: Dynamic solution advancement, Hilber-Hughes-Taylor method (p2a-hht-01errors)...

```
      0.0500 -0.2000 0.347865 0.047176 -1.01010101 -1.15178012
      0.1000 -0.2000 0.519421 0.328633 -4.16666667 -4.43495111
```

306.3.3.1 Verification Results for Hilber Hughes Taylor (HHT) Solution Advancement Algorithm.

The Real-ESSI model fei/DSL files for the results above can be downloaded HERE.

Table 306.2: Verification results for Hilber Hughes Taylor (HHT) solution advancement algorithm.

dt	α	Measured ξ	Analytic ξ	Measured T shift	Analytic T shift
0.005	-0.0	0.002953	0.000000	0.000000	0.000082
0.01	-0.0	0.003029	0.000000	0.000000	0.000329
0.05	-0.0	0.003690	0.000000	0.010101	0.008171
0.1	-0.0	0.003390	0.000000	0.030928	0.032075
0.005	-0.01	0.002954	0.000000	0.000000	0.000085
0.01	-0.01	0.003032	0.000000	0.000000	0.000339
0.05	-0.01	0.003573	0.000036	0.010101	0.008407
0.1	-0.01	0.003768	0.000259	0.030928	0.032933
0.005	-0.05	0.002957	0.000000	0.000000	0.000094
0.01	-0.05	0.003043	0.000001	0.000000	0.000374
0.05	-0.05	0.003227	0.000167	0.010101	0.009276
0.1	-0.05	0.009382	0.001184	0.030928	0.036111
0.005	-0.1	0.002959	0.000000	0.000000	0.000103
0.01	-0.1	0.003055	0.000003	0.000000	0.000413
0.05	-0.1	0.003054	0.000300	0.010101	0.010202
0.1	-0.1	0.004321	0.002106	0.041667	0.039506
0.005	-0.2	0.002964	0.000000	0.000000	0.000117
0.01	-0.2	0.003073	0.000004	0.000000	0.000467
0.05	-0.2	0.003479	0.000472	0.010101	0.011518
0.1	-0.2	0.005194	0.003286	0.041667	0.044350

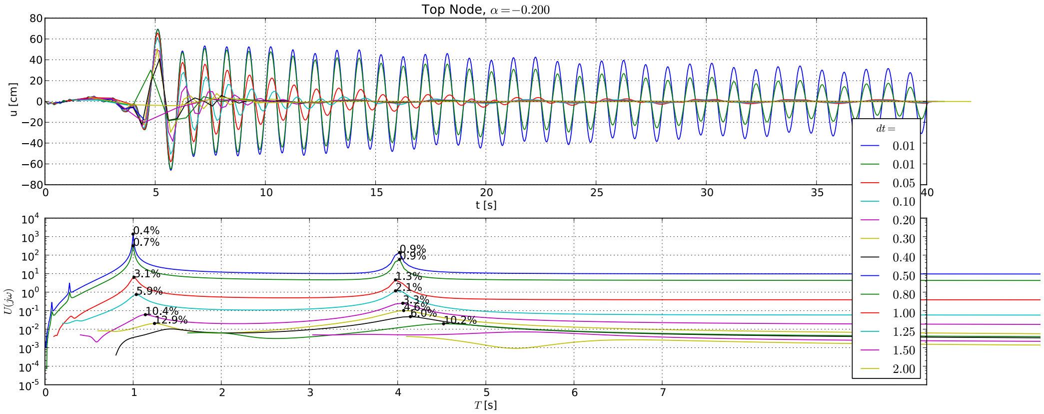


Figure 306.15: Verification: Dynamic solution advancement, Hilber-Hughes-Taylor method (p2a-hht-02top)...

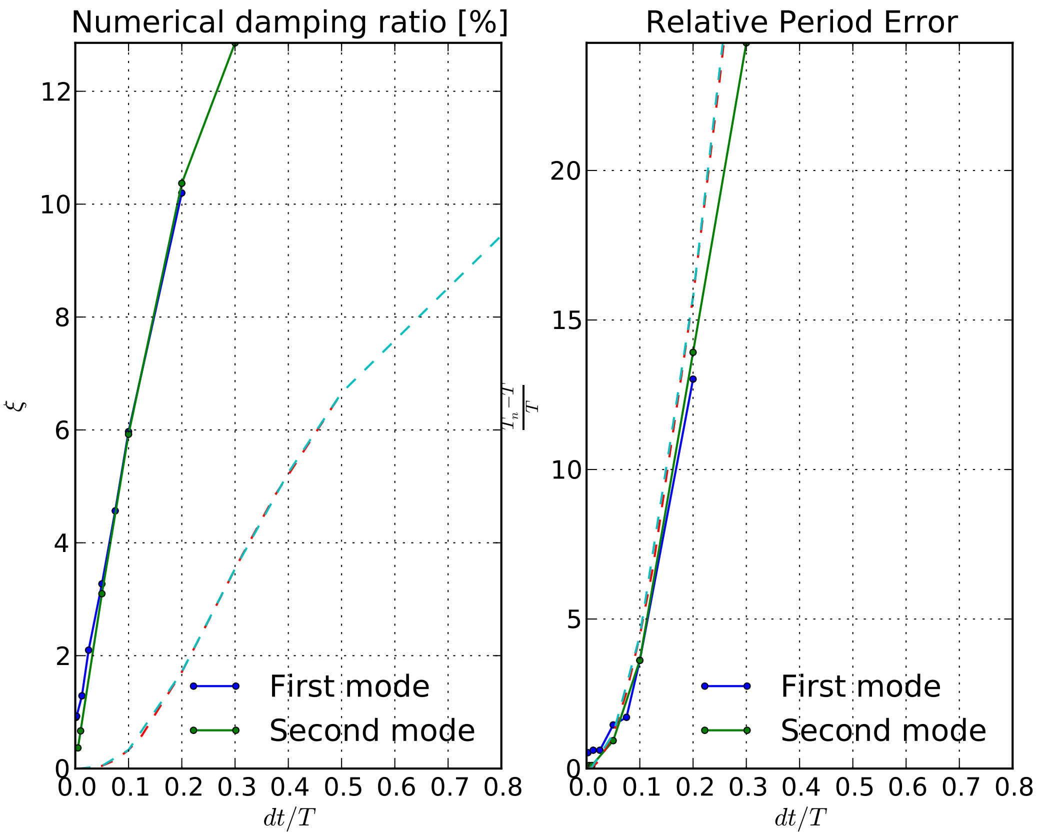


Figure 306.16: Verification: Dynamic solution advancement, Hilber-Hughes-Taylor method (p2a-hht-01errors)...

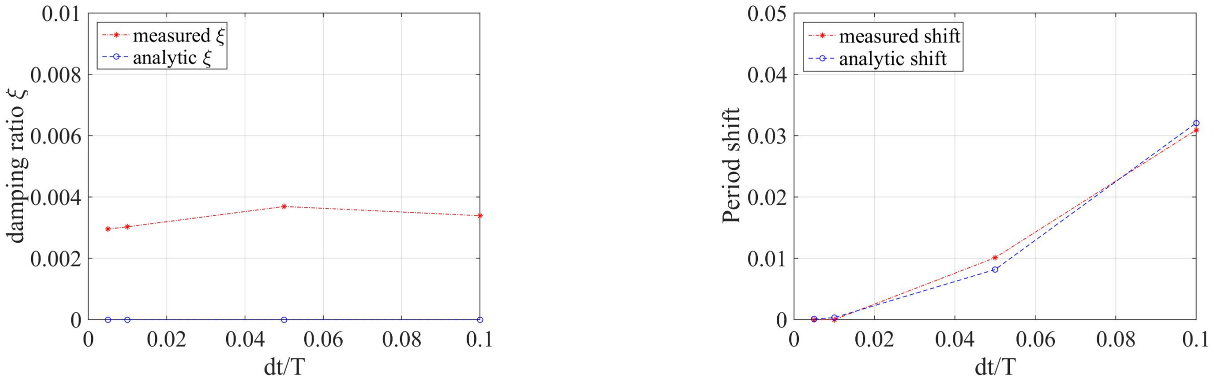


Figure 306.17: Comparison for HHT algorithm with $\alpha = -0.0$. Damping ratio comparison, Period shift comparison.

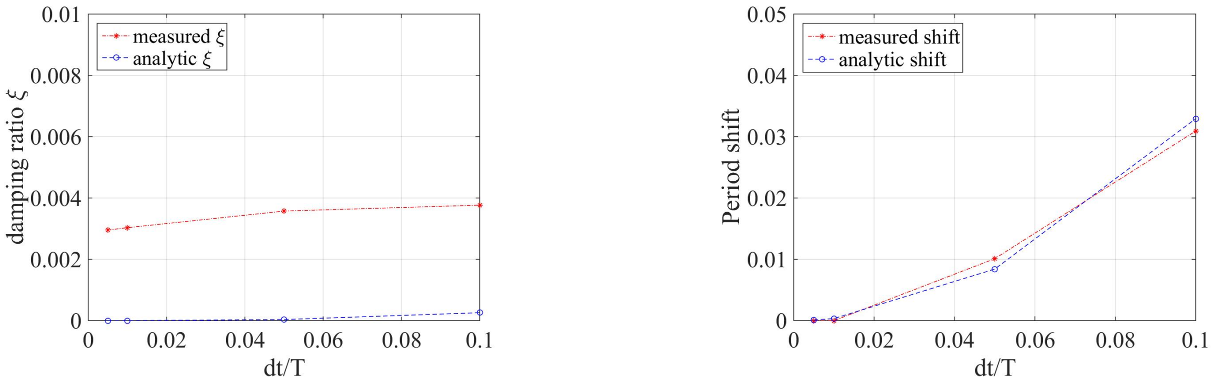


Figure 306.18: Comparison for HHT algorithm with $\alpha = -0.01$. Damping ratio comparison, Period shift comparison.

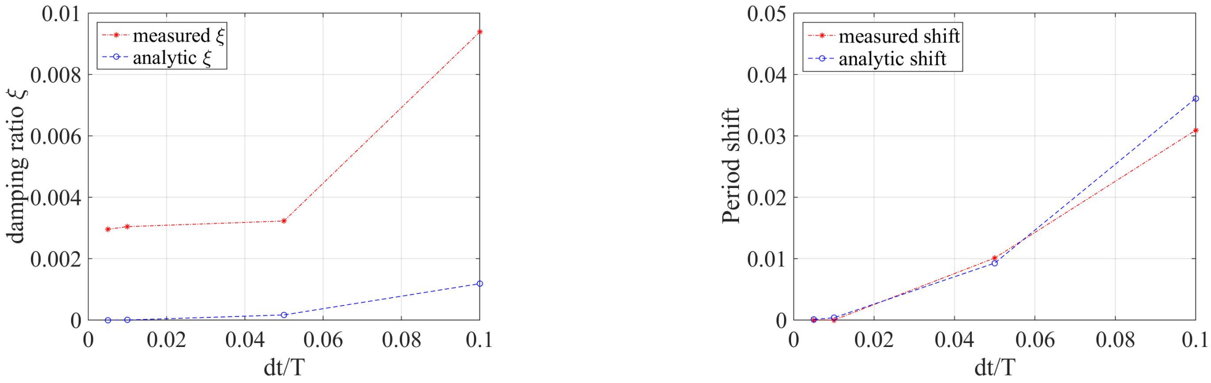


Figure 306.19: Comparison for HHT algorithm with $\alpha = -0.05$. Damping ratio comparison, Period shift comparison.

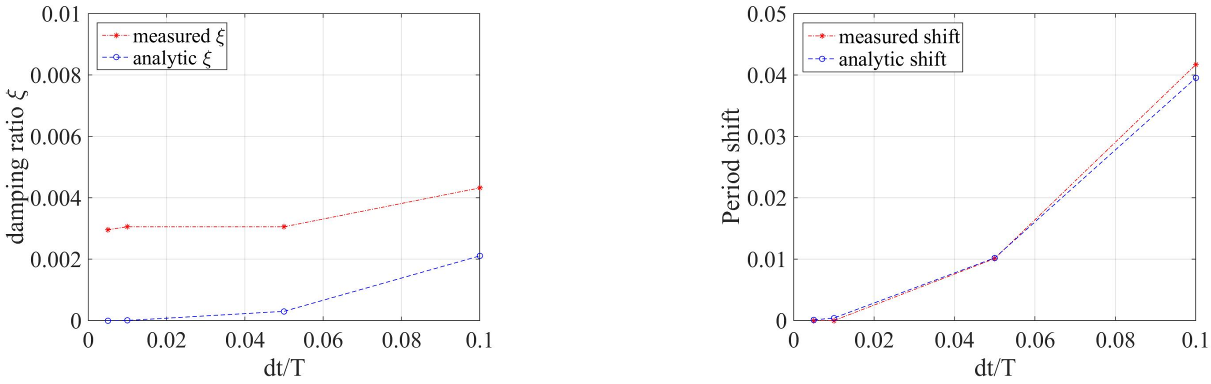


Figure 306.20: Comparison for HHT algorithm with $\alpha = -0.10$. Damping ratio comparison, Period shift comparison.

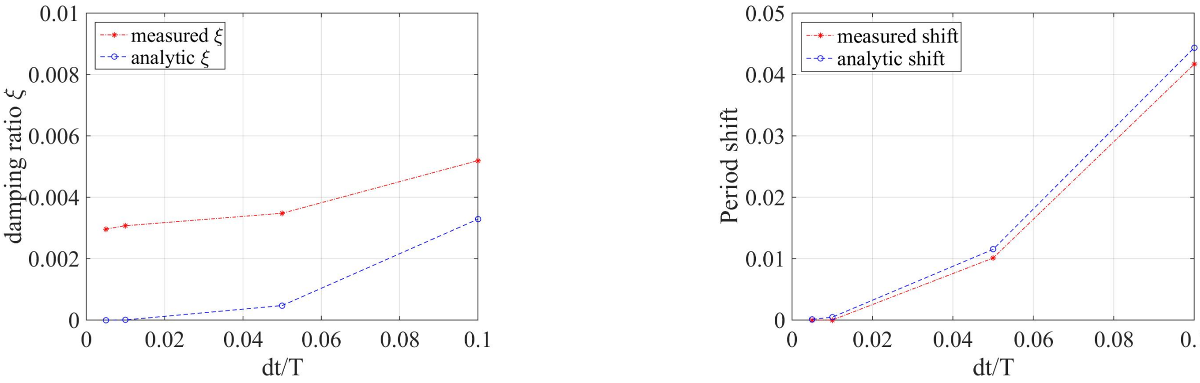


Figure 306.21: Comparison for HHT algorithm with $\alpha = -0.20$. Damping ratio comparison, Period shift comparison.

Chapter 307

Verification and Validation for Static and Dynamic Behavior of Single Phase, Solid Elements

(1989-1994-2011-2015-2017-2019-)

(In collaboration with Dr. Yuan Feng, and Prof. Han Yang)

307.1 Chapter Summary and Highlights

307.2 Verification of Static, Single Phase Solid Modeling and Simulation

307.2.1 Beam theory

This section provides basic beam theory that is used for verification solutions.

Problem description: Length=6m, Width=1m, Height=1m, F=100N, E=1E8Pa, $\nu = 0.0$. The force direction was shown in Figure (307.1).

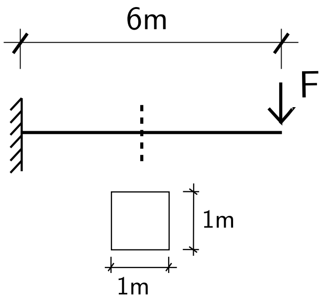


Figure 307.1: Problem description for cantilever beams.

The basic idea to calculate the shear deformation of a beam is

$$\delta = \frac{FL}{GA_v} \tag{307.1}$$

where A_v is the not the gross cross sectional area of the beam. A_v should be the shear area. Thus,

$$\kappa = \frac{A}{A_v} \tag{307.2}$$

where κ is the form factor, shear correction factor or shear deformation coefficient, A is the gross sectional area and A_v is the shear area of the section.

REWRITE, use bibtex! The history of κ value is long.

1. Timoshenko (1940)[1] define the form factor for rectangular section is 1.5.
2. Cowper (1970)[2] gave the formula for the form factor:

$$\kappa = \frac{12 + 11\nu}{10(1 + \nu)} \tag{307.3}$$

---

[1] Strength of materials, Timoshenko, Krieger Pub Co, 1940

[2] Cowper, G. R. "The shear coefficient in Timoshenko's beam theory." Journal of applied mechanics 33.2 (1966): 335-340.

3. Renton (1991) [3] provided a closed form solution for shear area of rectangular sections. For a rectangular section of depth $2a$ and breadth $2b$.

$$\kappa = \frac{6}{5} + (\frac{\nu}{1+\nu})^2 \sum_{m=0}^{\infty}\sum_{n=1}^{\infty} \frac{144(b/a)^4}{\pi^6(2m+1)^2 n^2[(2m+1)^2(b/2a)^2+n^2]} \tag{307.4}$$

For square cross section, $b = a$, therefore,

$$\kappa = \frac{6}{5} + (\frac{\nu}{1+\nu})^2 \sum_{m=0}^{\infty}\sum_{n=1}^{\infty} \frac{144}{\pi^6(2m+1)^2 n^2[(2m+1)^2(1/2)^2+n^2]} \tag{307.5}$$

To simplify the equation above, according to the Renton (1991), the intermediate values are given by

$$\kappa = \frac{6}{5} + C_1(\frac{\nu}{1+\nu})^2(\frac{b}{a})^4 \tag{307.6}$$

where C_1 is the parameter determined by the ratio of a and b. When $b = a$, the equation becomes

$$\kappa = \frac{6}{5} + 0.1392(\frac{\nu}{1+\nu})^2 \tag{307.7}$$

307.2.2 Verification of 8 node brick cantilever beam (static)

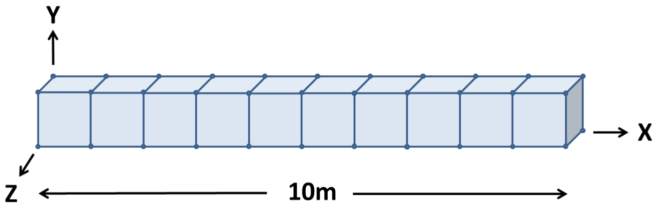
Problem description: Lenght=10m, Force=4N, E=100000Pa, $I = \frac{1}{12}$

Theoretical displacement:

$$d = \frac{PL^3}{3EI} = \frac{4 \times 1000}{3 \times 100000 \times \frac{1}{12}} = 0.16m \tag{307.8}$$

Numerical simulation results:

1m element size (10 elements):

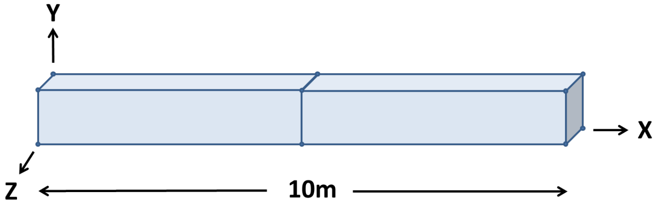


$$error = \frac{0.16 - 0.1072}{0.16} = 33\% \tag{307.9}$$

5m element size (2 elements):

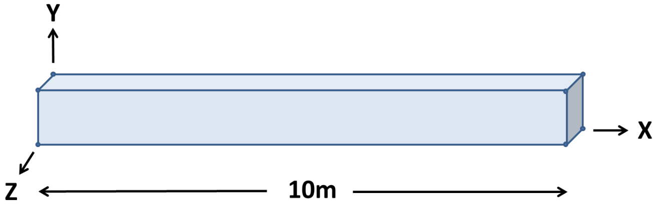
---

[3] Renton, J. D. "Generalized beam theory applied to shear stiffness." International Journal of Solids and Structures 27.15 (1991): 1955-1967.



$$error = \frac{0.16 - 0.011911}{0.16} = 92.5\% \qquad (307.10)$$

10m element size (1 element):



$$error = \frac{0.16 - 0.00315}{0.16} = 98\% \qquad (307.11)$$

1m element size with 10% nodal offset (10 elements):

$$error = \frac{0.16 - 0.1057}{0.16} = 34\% \qquad (307.12)$$

1m element size with 20% nodal offset (10 elements):

$$error = \frac{0.16 - 0.1016}{0.16} = 36\% \qquad (307.13)$$

307.2.3 Verification of 27 node brick cantilever beam (static)

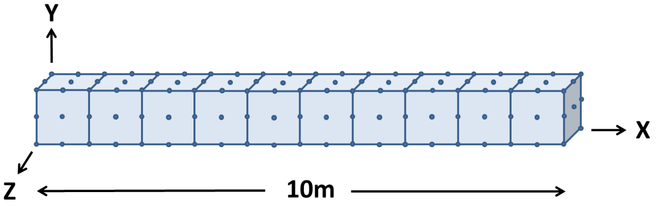
Problem description: Lenght=10m, Force=9N, E=100000Pa, $I = \frac{1}{12}$

Theoretical displacement:

$$d = \frac{PL^3}{3EI} = \frac{9 \times 1000}{3 \times 100000 \times \frac{1}{12}} = 0.36m \qquad (307.14)$$

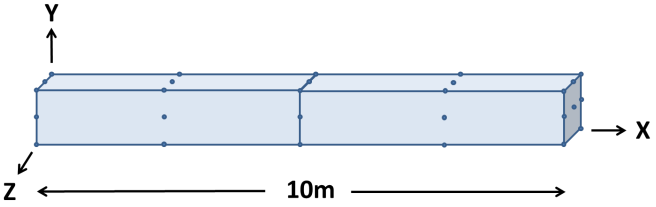
Numerical simulation results:

1m element size (10 elements):



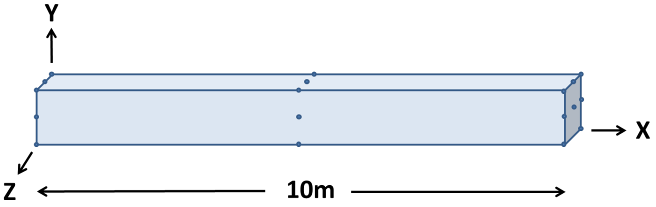
$$error = \frac{0.361721 - 0.36}{0.36} = 0.47\% \qquad (307.15)$$

5m element size (2 elements):



$$error = \frac{0.36 - 0.345719}{0.36} = 3.96\% \qquad (307.16)$$

10m element size (1 element):



$$error = \frac{0.36 - 0.279989}{0.36} = 22\% \qquad (307.17)$$

1m element size with 10% nodal offset (10 elements):

$$error = \frac{0.361225 - 0.36}{0.36} = 0.35\% \qquad (307.18)$$

1m element size with 20% nodal offset (10 elements):

$$error = \frac{0.36 - 0.359741}{0.36} = 0.07\% \qquad (307.19)$$

1m element size with 30% nodal offset (10 elements):

$$error = \frac{0.36 - 0.357004}{0.36} = 0.83\% \tag{307.20}$$

1m element size with 40% nodal offset (10 elements):

$$error = \frac{0.36 - 0.352604}{0.36} = 2\% \tag{307.21}$$

307.2.4 Verification of 8NodeBrick cantilever beams

Problem description: Length=6m, Width=1m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.0$. Use the shear deformation coefficient $\kappa = 1.2$. The force direction was shown in Figure (307.2).

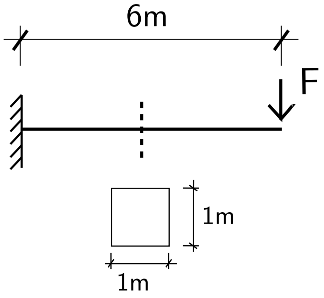


Figure 307.2: Problem description for cantilever beams.

Theoretical displacement (bending and shear deformation):

$$\begin{aligned} d &= \frac{FL^3}{3EI} + \frac{FL}{GA_v} \\ &= \frac{FL^3}{3E\frac{bh^3}{12}} + \frac{FL}{\frac{E}{2(1+\nu)}\frac{bh}{\kappa}} \\ &= \frac{100N \times 6^3 m^3}{3 \times 10^8 N/m^2 \times \frac{1}{12} m^4} + \frac{100N \times 6m}{\frac{10}{2} \times 10^7 N/m^2 \times 1m^2 \times \frac{5}{6}} \\ &= 8.64 \times 10^{-4} m + 0.144 \times 10^{-4} m \\ &= 8.784 \times 10^{-4}\ m \end{aligned} \tag{307.22}$$

Numerical model:

The 8NodeBrick elements are shown in Figure (307.6).

An example Real-ESSI script is shown below.

All the Real-ESSI results are listed in Table (307.1). The theoretical solution is 8.784E-04 m.

Table 307.1: Results for 8NodeBrick cantilever beams of different element numbers.

Element number	1	2	6
8NodeBrick	4.61E-05 m	1.59E-04 m	5.84E-04 m
Error	94.75%	81.87%	33.52%

The errors are plotted in Figure (307.7).

The Real-ESSI model fei/DSL files for the table above are HERE.

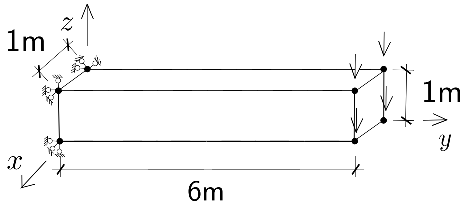


Figure 307.3: One 8NodeBrick element.

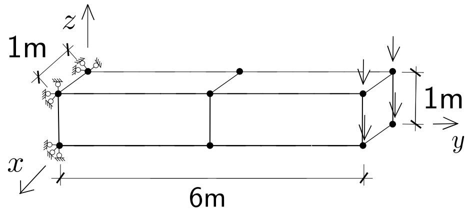


Figure 307.4: Two 8NodeBrick elements.

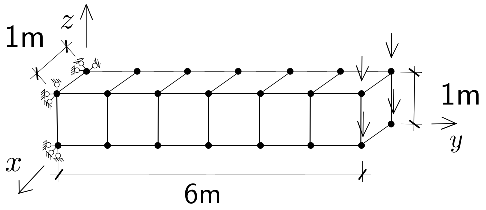


Figure 307.5: Six 8NodeBrick elements.

Figure 307.6: 8NodeBrick elements for cantilever beams.

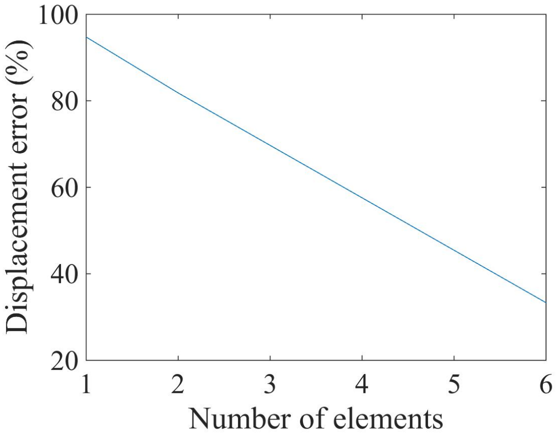


Figure 307.7: 8NodeBrick cantilever beam for different element number Displacement error versus Number of elements

307.2.5 Verification of 8NodeBrick cantilever beam for different Poisson's ratio

Problem description: Length=6m, Width=1m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.0 - 0.49$. The force direction was shown in Figure (307.8).

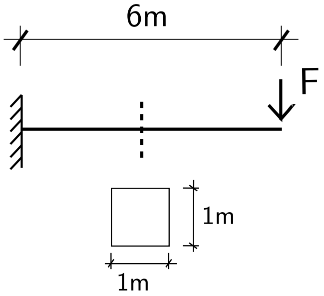


Figure 307.8: Problem description for cantilever beams of different Poisson's ratios.

The theoretical solution for $\nu = 0.0$ was calculated below, while the solution for other Poisson's ratio are calculated by the similar process.

Theoretical displacement (bending and shear deformation):

$$\begin{aligned} d &= \frac{FL^3}{3EI} + \frac{FL}{GA_v} \\ &= \frac{FL^3}{3E\frac{bh^3}{12}} + \frac{FL}{\frac{E}{2(1+\nu)}\frac{bh}{\kappa}} \\ &= \frac{100N \times 6^3 m^3}{3 \times 10^8 N/m^2 \times \frac{1}{12} m^4} + \frac{100N \times 6m}{\frac{10}{2} \times 10^7 N/m^2 \times 1m^2 \times \frac{5}{6}} \\ &= 8.64 \times 10^{-4} m + 0.144 \times 10^{-4} m \\ &= 8.784 \times 10^{-4}\ m \end{aligned} \tag{307.23}$$

The rotation angle at the end:

$$\theta = \frac{FL^2}{2EI} = \frac{100N \times 6^2 m^2}{2 \times 10^8 N/m^2 \times \frac{1}{12} m^4} = 2.16 \times 10^{-4}\ rad = 0.0124^\circ \tag{307.24}$$

The 8NodeBrick elements for cantilever beams of different Poisson's ratios are shown in Figure (307.9):

All the displacement results are listed in Table (307.2) - (307.4).

Using the same geometry, the element was meshed usingm much smaller element (0.5m).

Finally, in the same geometry, the element side length was cut into 0.25m.

The errors are plotted in Figure (307.10).

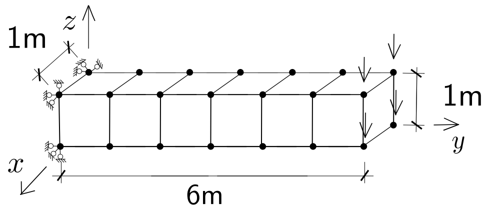


Figure 307.9: 8NodeBrick elements for cantilever beams of different Poisson's ratios.

Table 307.2: Displacement results for 8NodeBrick cantilever beams with element side length 1 m.

Poisson's ratio	8NodeBrick displacement	Theory displacement (bending)	Theory displacement (shear)	Theory displacement(all)	Error
0.00	5.840E-04 m	8.640E-04 m	1.440E-05 m	8.784E-04 m	33.52%
0.05	5.924E-04 m	8.640E-04 m	1.512E-05 m	8.791E-04 m	32.62%
0.10	5.969E-04 m	8.640E-04 m	1.586E-05 m	8.799E-04 m	32.16%
0.15	5.971E-04 m	8.640E-04 m	1.659E-05 m	8.806E-04 m	32.20%
0.20	5.922E-04 m	8.640E-04 m	1.734E-05 m	8.813E-04 m	32.81%
0.25	5.814E-04 m	8.640E-04 m	1.808E-05 m	8.821E-04 m	34.09%
0.30	5.634E-04 m	8.640E-04 m	1.884E-05 m	8.828E-04 m	36.19%
0.35	5.364E-04 m	8.640E-04 m	1.959E-05 m	8.836E-04 m	39.29%
0.40	4.970E-04 m	8.640E-04 m	2.035E-05 m	8.844E-04 m	43.80%
0.45	4.353E-04 m	8.640E-04 m	2.111E-05 m	8.851E-04 m	50.82%
0.49	3.142E-04 m	8.640E-04 m	2.173E-05 m	8.857E-04 m	64.52%

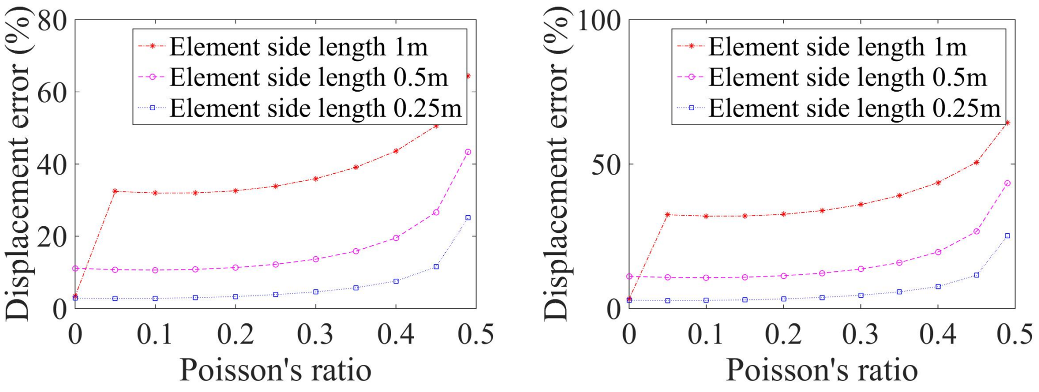


Figure 307.10: 8NodeBrick cantilever beam for different Poisson's ratio˙ Displacement error versus Poisson's ratio. Left: Error scale 0% - 80%, Right: Error scale 0% - 100%.

Table 307.3: Displacement results for 8NodeBrick cantilever beams with element side length 0.5 m.

Poisson's ratio	8NodeBrick displacement	Theory displacement (bending)	Theory displacement (shear)	Theory displacement(all)	Error
0.00	7.787E-04 m	8.640E-04 m	1.440E-05 m	8.784E-04 m	11.35%
0.05	7.824E-04 m	8.640E-04 m	1.512E-05 m	8.791E-04 m	11.00%
0.10	7.839E-04 m	8.640E-04 m	1.586E-05 m	8.799E-04 m	10.91%
0.15	7.829E-04 m	8.640E-04 m	1.659E-05 m	8.806E-04 m	11.09%
0.20	7.790E-04 m	8.640E-04 m	1.734E-05 m	8.813E-04 m	11.61%
0.25	7.717E-04 m	8.640E-04 m	1.808E-05 m	8.821E-04 m	12.51%
0.30	7.597E-04 m	8.640E-04 m	1.884E-05 m	8.828E-04 m	13.95%
0.35	7.406E-04 m	8.640E-04 m	1.959E-05 m	8.836E-04 m	16.18%
0.40	7.089E-04 m	8.640E-04 m	2.035E-05 m	8.844E-04 m	19.84%
0.45	6.466E-04 m	8.640E-04 m	2.111E-05 m	8.851E-04 m	26.95%
0.49	4.990E-04 m	8.640E-04 m	2.173E-05 m	8.857E-04 m	43.66%

Table 307.4: Displacement results for 8NodeBrick cantilever beams with element side length 0.25 m.

Poisson's ratio	8NodeBrick displacement	Theory displacement (bending)	Theory displacement (shear)	Theory displacement(all)	Error
0.00	8.511E-04 m	8.640E-04 m	1.440E-05 m	8.784E-04 m	3.11%
0.05	8.525E-04 m	8.640E-04 m	1.512E-05 m	8.791E-04 m	3.03%
0.10	8.527E-04 m	8.640E-04 m	1.586E-05 m	8.799E-04 m	3.09%
0.15	8.518E-04 m	8.640E-04 m	1.659E-05 m	8.806E-04 m	3.27%
0.20	8.494E-04 m	8.640E-04 m	1.734E-05 m	8.813E-04 m	3.62%
0.25	8.455E-04 m	8.640E-04 m	1.808E-05 m	8.821E-04 m	4.15%
0.30	8.393E-04 m	8.640E-04 m	1.884E-05 m	8.828E-04 m	4.93%
0.35	8.299E-04 m	8.640E-04 m	1.959E-05 m	8.836E-04 m	6.08%
0.40	8.141E-04 m	8.640E-04 m	2.035E-05 m	8.844E-04 m	7.94%
0.45	7.801E-04 m	8.640E-04 m	2.111E-05 m	8.851E-04 m	11.86%
0.49	6.603E-04 m	8.640E-04 m	2.173E-05 m	8.857E-04 m	25.45%

The angle results are listed in Table (307.5).

Then, in the same geometry, element side length was cut into 0.5m. The angle results are listed in Table (307.6).

Table 307.5: Rotation angle results for 8NodeBrick cantilever beams with element side length 1 m.

Poisson's ratio	8NodeBrick angle(unit:°)	Theory angle (unit:°)	Error
0.00	8.25E-03	1.24E-02	33.46%
0.05	8.36E-03	1.24E-02	32.55%
0.10	8.42E-03	1.24E-02	32.08%
0.15	8.42E-03	1.24E-02	32.10%
0.20	8.35E-03	1.24E-02	32.67%
0.25	8.20E-03	1.24E-02	33.90%
0.30	7.95E-03	1.24E-02	35.89%
0.35	7.59E-03	1.24E-02	38.83%
0.40	7.07E-03	1.24E-02	43.00%
0.45	6.30E-03	1.24E-02	49.21%
0.49	4.93E-03	1.24E-02	60.20%

Table 307.6: Rotation angle results for 8NodeBrick cantilever beams with with element side length 0.5 m.

Poisson's ratio	8NodeBrick angle (unit:°)	Theory angle (unit:°)	Error
0.00	1.10E-02	1.24E-02	11.28%
0.05	1.10E-02	1.24E-02	10.91%
0.10	1.11E-02	1.24E-02	10.78%
0.15	1.10E-02	1.24E-02	10.90%
0.20	1.10E-02	1.24E-02	11.32%
0.25	1.09E-02	1.24E-02	12.09%
0.30	1.07E-02	1.24E-02	13.33%
0.35	1.05E-02	1.24E-02	15.29%
0.40	1.01E-02	1.24E-02	18.53%
0.45	9.32E-03	1.24E-02	24.87%
0.49	7.52E-03	1.24E-02	39.35%

Finally, in the same geometry, element side length was cut into 0.25m. The angle results are listed in Table (307.7).

Table 307.7: Rotation angle results for 8NodeBrick cantilever beams with with element side length 0.25 m.

Poisson's ratio	8NodeBrick angle (unit:°	Theory angle (unit:°	Error
0.00	1.20E-02	1.24E-02	3.06%
0.05	1.20E-02	1.24E-02	2.97%
0.10	1.20E-02	1.24E-02	2.99%
0.15	1.20E-02	1.24E-02	3.12%
0.20	1.20E-02	1.24E-02	3.38%
0.25	1.19E-02	1.24E-02	3.79%
0.30	1.19E-02	1.24E-02	4.40%
0.35	1.17E-02	1.24E-02	5.33%
0.40	1.15E-02	1.24E-02	6.87%
0.45	1.11E-02	1.24E-02	10.22%
0.49	9.64E-03	1.24E-02	22.23%

The errors are plotted in Figure (307.11).

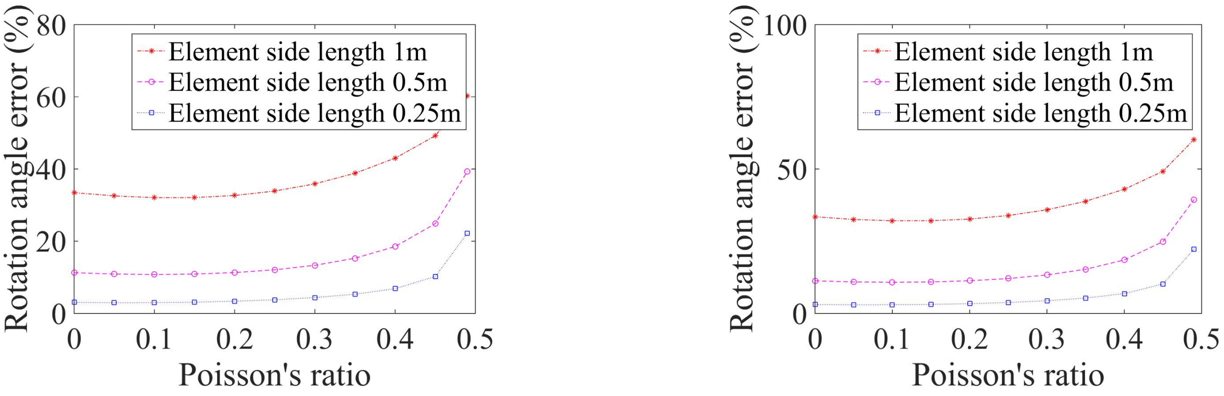


Figure 307.11: 8NodeBrick cantilever beam for different Poisson's ratio Rotation angle error versus Poisson's ratio, Left: Error scale 30% - 70%, Right: Error scale 0% - 100%.

The Real-ESSI model fei/DSL files for the table above are HERE.

307.2.6 Test of irregular shaped 8NodeBrick cantilever beams

Cantilever model was used as an example. Three different shapes are tested.

In the first test, the upper two nodes of each element are moved one half element size along the $y-axis$, while the lower two nodes are kept at the same location. The element shape was shown in Figure (307.12).

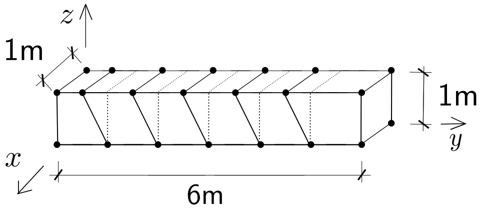


Figure 307.12: 8NodeBrick cantilever beams for irregular Shape 1.

In the second test, the upper two nodes of each element are moved 90% element size along the $y-axis$, while the lower two nodes are moved 90% element size in the other direction along the $y-axis$. The element shape was shown in Figure (307.13).

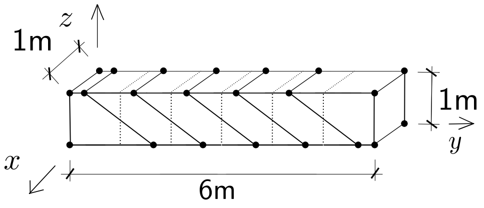


Figure 307.13: 8NodeBrick cantilever beams for irregular Shape 2.

In the third test, the upper two nodes of each element are moved one half element size with different directions along the $y-axis$, while the lower two nodes are kept at the same location. The element shape was shown in Figure (307.14).

The boundary conditions are shown in Figures (307.17), (307.20) and (307.23).

The Real-ESSI results are listed in Table (307.8).

The errors are listed in Table (307.9) and (307.10).

The Real-ESSI model fei/DSL files for the table above are HERE.

Then, the irregular beam was divided into small elements.

Problem description: Length=12m, Width=2m, Height=2m, q=400N/m, E=1E8Pa, $\nu = 0.0$. Use the shear deformation coefficient $\kappa = 1.2$. The force direction was shown in Figure (307.24).

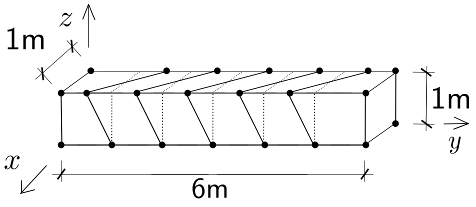


Figure 307.14: 8NodeBrick cantilever beams for irregular Shape 3.

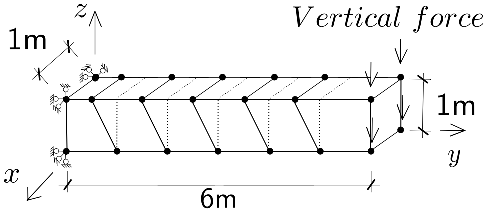


Figure 307.15: Veritical force.

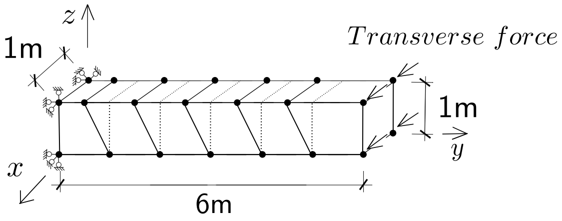


Figure 307.16: Horizontal force.

Figure 307.17: 8NodeBrick cantilever beam boundary conditions for irregular Shape 1.

Table 307.8: Results for 8NodeBrick cantilever beams of irregular shapes.

Element Type	Force direction	Normal shape	Shape 1	Shape 2	Shape 3
8NodeBrick	Vertical (z)	5.840E-04 m	5.751E-04 m	2.959E-04 m	3.883E-04 m
8NodeBrick	Transverse (y)	5.840E-04 m	4.529E-04 m	1.390E-04 m	4.744E-04 m
Theoretical	-	8.784E-04 m	8.784E-04 m	8.784E-04 m	8.784E-04 m

Table 307.9: Errors for irregular shaped 8NodeBrick compared to theoretical solution.

Element Type	Force direction	Normal shape	Shape 1	Shape 2	Shape 3
8NodeBrick	Vertical (z)	33.52%	34.53%	66.31%	55.79%
8NodeBrick	Transverse (y)	33.52%	48.44%	84.18%	45.99%

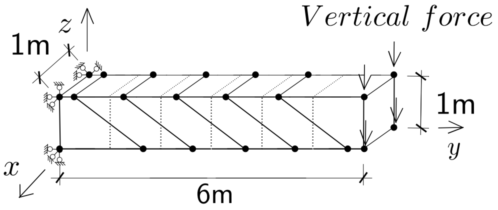


Figure 307.18: Veritical force.

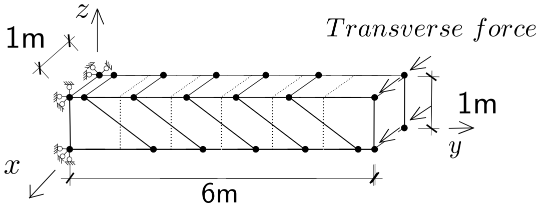


Figure 307.19: Horizontal force.

Figure 307.20: 8NodeBrick cantilever beam boundary conditions for irregular Shape 2.

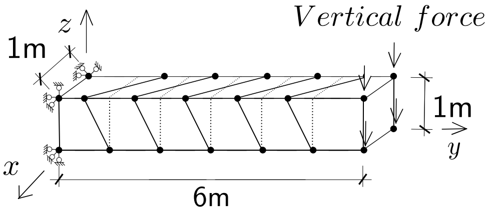


Figure 307.21: Veritical force.

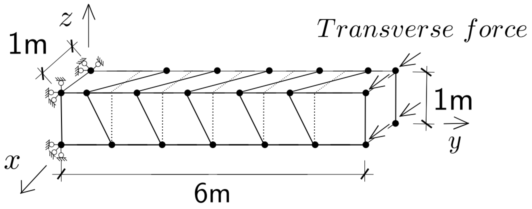


Figure 307.22: Horizontal force.

Figure 307.23: 8NodeBrick cantilever beam boundary conditions for irregular Shape 3.

Theoretical displacement (bending and shear deformation):

$$d = \frac{qL^4}{8EI} + \frac{q\frac{L^2}{2}}{GA_v} \tag{307.25}$$

$$= \frac{qL^4}{8E\frac{bh^3}{12}} + \frac{q\frac{L^2}{2}}{\frac{E}{2(1+\nu)}\frac{bh}{\kappa}} \tag{307.26}$$

$$= \frac{400N/m\times 12^4 m^4}{8\times 10^8 N/m^2\times\frac{2^4}{12}m^4} + \frac{400N/m\times\frac{12^2}{2}m^2}{\frac{10^8}{2}N/m^2\times 2m\times 2m\times\frac{5}{6}} \tag{307.27}$$

$$= 7.776\times 10^{-3}m + 1.728\times 10^{-4}m \tag{307.28}$$

Table 307.10: Errors for irregular shaped 8NodeBrick compared to normal shape.

Element Type	Force direction	Normal shape	Shape 1	Shape 2	Shape 3
8NodeBrick	Vertical (z)	0.00%	1.52%	49.33%	33.51%
8NodeBrick	Transverse (y)	0.00%	22.45%	76.20%	18.77%

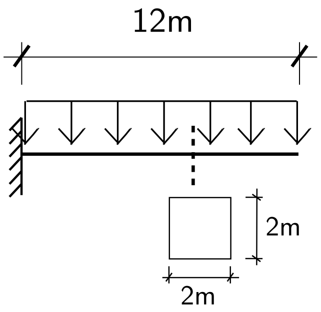


Figure 307.24: Problem description for cantilever beams under uniform load .

The Real-ESSI displacement results are listed in Table (307.11).

Table 307.11: Results for 8NodeBrick cantilever beams of irregular shapes with more elements.

Element Type	Shape	Force direction	Number of division		
			1	2	4
8NodeBrick	shape1	Vertical (z)	5.37E-03 m	7.08E-03 m	7.71E-03 m
8NodeBrick	shape1	Transverse (y)	4.60E-03 m	6.66E-03 m	7.58E-03 m
8NodeBrick	shape2	Vertical (z)	2.74E-03 m	4.75E-03 m	6.43E-03 m
8NodeBrick	shape2	Transverse (y)	1.46E-03 m	2.72E-03 m	4.63E-03 m
8NodeBrick	shape3	Vertical (z)	9.21E-04 m	6.60E-03 m	7.56E-03 m
8NodeBrick	shape3	Transverse (y)	1.09E-03 m	6.09E-03 m	7.37E-03 m
Theoretical solution			7.95E-03 m	7.95E-03 m	7.95E-03 m

The error are listed in Table (307.12).

The errors are shown in Figures (307.25), (307.26) and (307.27).

The Real-ESSI model fei/DSL files for the table above are HERE.

Table 307.12: Errors for 8NodeBrick cantilever beams of irregular shapes with more elements.

Element Type	Shape	Force direction	Number of division		
			1	2	4
8NodeBrick	shape1	Vertical (z)	32.42%	10.95%	3.01%
8NodeBrick	shape1	Transverse (y)	42.16%	16.17%	4.69%
8NodeBrick	shape2	Vertical (z)	65.59%	40.22%	19.05%
8NodeBrick	shape2	Transverse (y)	81.57%	65.76%	41.81%
8NodeBrick	shape3	Vertical (z)	88.42%	16.97%	4.89%
8NodeBrick	shape3	Transverse (y)	86.24%	23.36%	7.28%

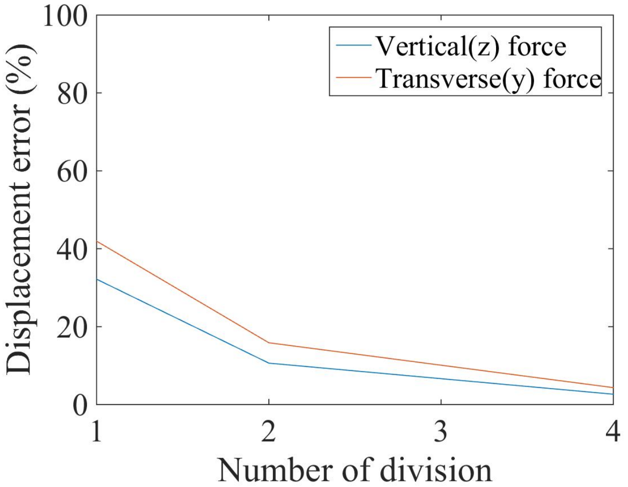


Figure 307.25: 8NodeBrick cantilever beam for irregular Shape 1: Displacement error versus Number of division

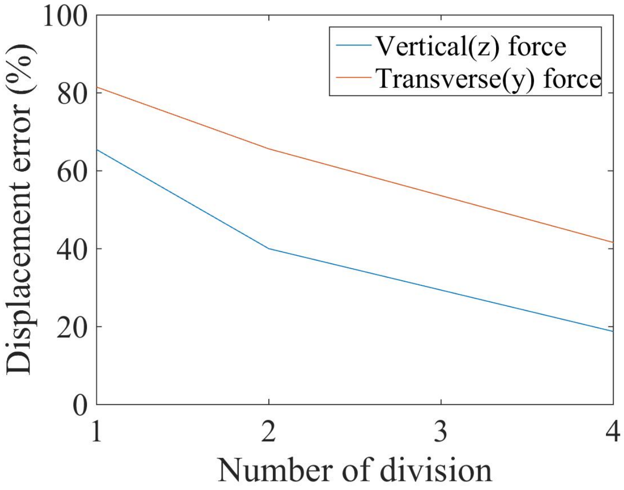


Figure 307.26: 8NodeBrick cantilever beam for irregular Shape 2 ˙ Displacement error versus Number of division

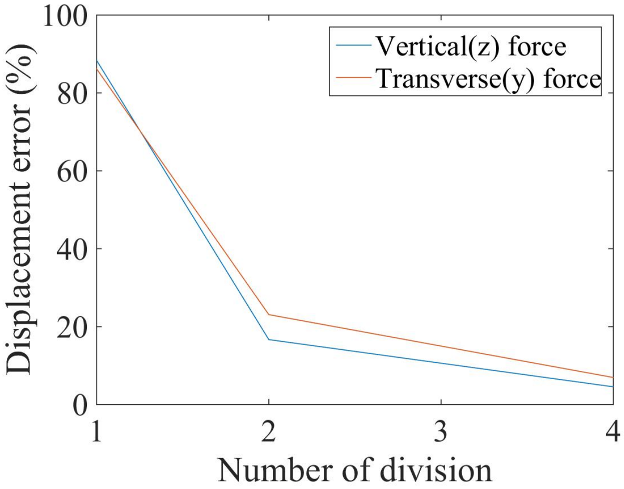


Figure 307.27: 8NodeBrick cantilever beam for irregular Shape 3˙ Displacement error versus Number of division

In this section, the beam was cut into smaller elements with element side length 0.5m and 0.25m respectively. And the element side length of the original models is 1.0m. The numerical models are shown in Figures (307.28), (307.29) and (307.30).

Number of division 1:

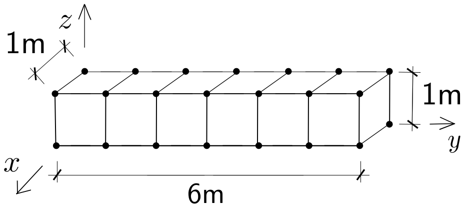


Figure 307.28: 8NodeBrick clamped beams with element side length 1.0m.

Number of division 2:

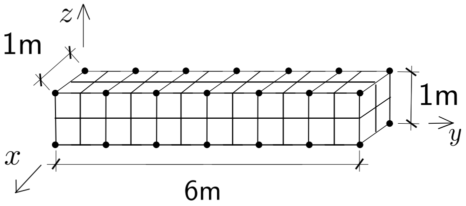


Figure 307.29: 8NodeBrick clamped beams with element side length 0.5m.

Number of division 4:

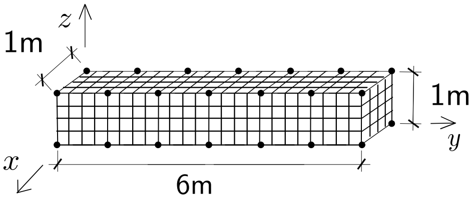


Figure 307.30: 8NodeBrick clamped beams with element side length 0.25m.

The Real-ESSI results are listed in Table (307.13). The theoretical solution is 1.60E-5 m.

The errors are plotted in Figure (307.33).

The Real-ESSI model fei/DSL files for the table above are HERE.

Table 307.13: Results for 8NodeBrick clamped beams with more elements.

Element Type	Element side length		
	1 m	0.5 m	0.25 m
8NodeBrick	1.10E-05 m	1.47E-05 m	1.64E-05 m
Error	33.33%	11.09%	0.73%

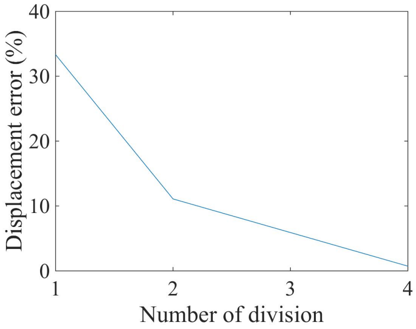


Figure 307.31: Error scale 0% - 40%.

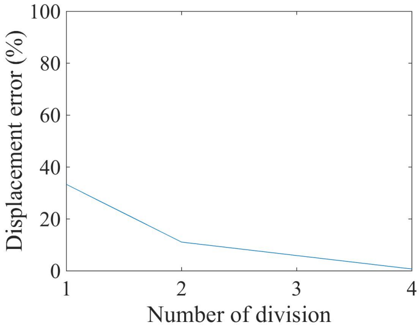


Figure 307.32: Error scale 0% - 100%.

Figure 307.33: 8NodeBrick clamped beam for different element number˙ Displacement error versus Number of division

307.2.7 Verification of 8NodeBrick stress in cantilever beams

Problem description: Length=6m, Width=1m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.0$. Use the shear deformation coefficient $\kappa = 1.2$. The force direction was shown in Figure (307.34).

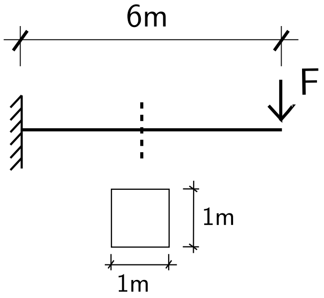


Figure 307.34: Problem description for cantilever beams of stress verification.

The theoretical solution for the stress was calculated below.

The 8NodeBrick elements are shown in Figure (307.35).

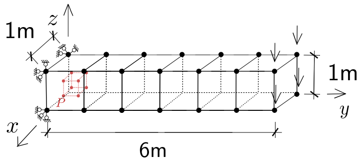


Figure 307.35: 8NodeBrick for cantilever beams of stress verification.

The bending moment at the Gassian Point is

$$M = F(L - P_y) = 100N \times (6 - 0.2113)m = 578.87N \cdot m \tag{307.30}$$

The bending modulus is

$$I = \frac{bh^3}{12} = \frac{1}{12}m^4 \tag{307.31}$$

Therefore, the theoretical stress is

$$\sigma = \frac{M \cdot z}{I} = \frac{578.87N \cdot m \times (0.5 - 0.2113)m}{\frac{1}{12}m^4} = 2005Pa \tag{307.32}$$

To get a better result, the same geometry beam was also cut into small elements. When more elements are used, the theoretical stress was calculated again with the new coordinates. The calculation process is similar to the process above.

The numerical models are shown in Figures(307.36), (307.37) and (307.38).

Number of division 1:

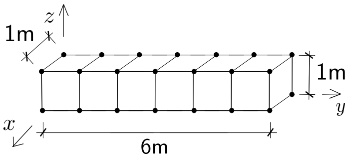


Figure 307.36: 8NodeBrick stress with element side length 1.0m.

Number of division 2:

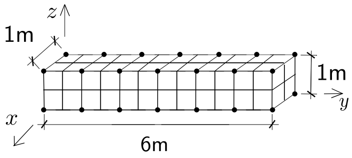


Figure 307.37: 8NodeBrick stress with element side length 0.5m.

Number of division 4:

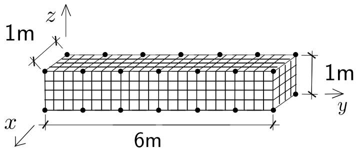


Figure 307.38: 8NodeBrick stress with element side length 0.25m.

All the stress results are listed in Table (307.14).

Table 307.14: Results for 8NodeBrick stress with more elements.

Element Type	Element side length		
	1 m	0.5 m	0.25 m
8NodeBrick	1270.17 Pa	2418.60 Pa	3085.48 Pa
Theoretical	2005.26 Pa	2789.23 Pa	3191.27 Pa
Error	36.66%	13.29%	3.31%

The errors are plotted in Figure (307.41).

The Real-ESSI model fei/DSL files for the table above are HERE.

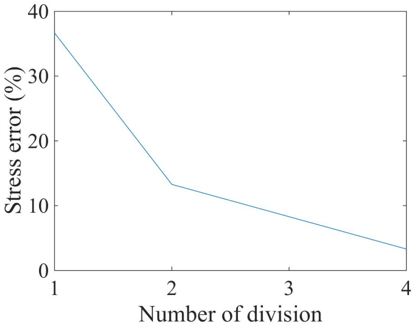


Figure 307.39: Error scale 0% - 40%.

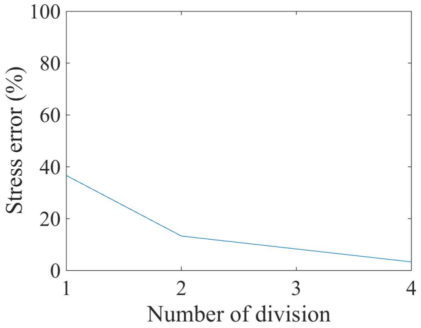


Figure 307.40: Error scale 0% - 100%.

Figure 307.41: 8NodeBrick cantilever beams for stress verification˙ Stress error versus Number of division

307.2.8 Verification of 8NodeBrick square plate with four edges clamped

Problem description: Length=20m, Width=20m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.3$.

The four edges are clamped.

The load is the uniform normal pressure on the whole plate.

The plate flexural rigidity is

$$D = \frac{Eh^3}{12(1-\nu^2)} = \frac{10^8 N/m^2 \times 1^3 m^3}{12 \times (1-0.3^2)} = 9.1575 \times 10^6 \; N \cdot m \tag{307.33}$$

The theoretical solution is

$$d = \alpha_c \frac{qa^4}{D} = 0.00406 \times \frac{100N/m^2 \times 20^4 m^4}{9.1575 \times 10^6 \; N \cdot m} = 2.2015 \times 10^{-3} m \tag{307.34}$$

where α_c is a coefficient, which depends on the ratio of plate length to width. In this problem, the coefficient[4] α_c is 0.00406.

The 8NodeBrick are shown in Figures (307.42) - (307.47).

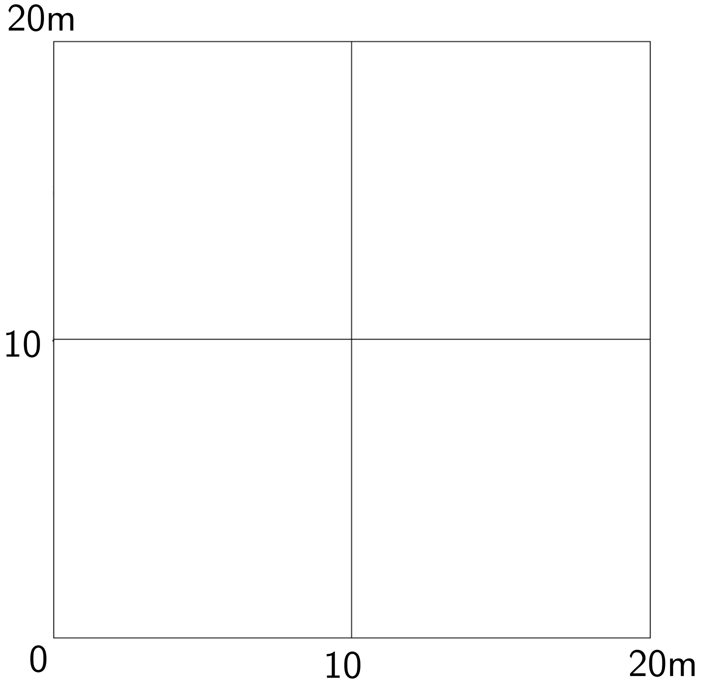


Figure 307.42: 8NodeBrick edge clamped square plate with element side length 10m.

The results are listed in Table (307.15).

The errors are listed in Table (307.2.8).

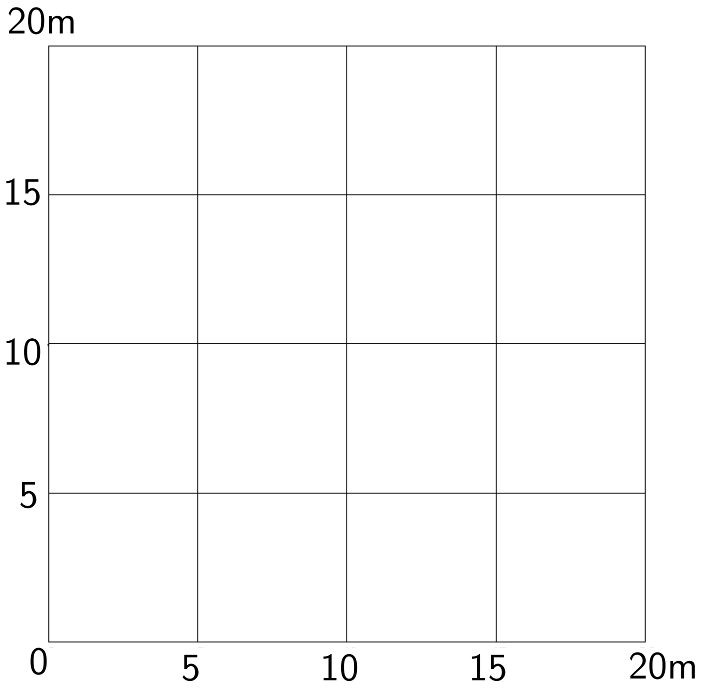


Figure 307.43: 8NodeBrick edge clamped square plate with element side length 5m.

Table 307.15: Results for 8NodeBrick square plate with four edges clamped.

Element type	8NodeBrick	8NodeBrick	8NodeBrick	Theoretical displacement
Number of layers	1layer	2layers	4layers	
Element side length	Height:1.00m	Height:0.50m	Height:0.25m	
10m	9.75E-05 m	9.75E-05 m	9.75E-05 m	2.20E-03 m
5m	3.28E-04 m	3.32E-04 m	3.32E-04 m	2.20E-03 m
2m	1.04E-03 m	1.10E-03 m	1.12E-03 m	2.20E-03 m
1m	1.56E-03 m	1.74E-03 m	1.79E-03 m	2.20E-03 m
0.5m	1.80E-03 m	2.30E-03 m	2.12E-03 m	2.20E-03 m
0.25m	1.87E-03 m	2.14E-03 m	2.23E-03 m	2.20E-03 m

The errors are plotted in Figure (307.48).

The Real-ESSI model fei/DSL files for the table above are HERE.

---

[4] Stephen Timoshenko, Theory of plates and shells (2nd edition). MrGRAW-Hill Inc, page120, 1959.

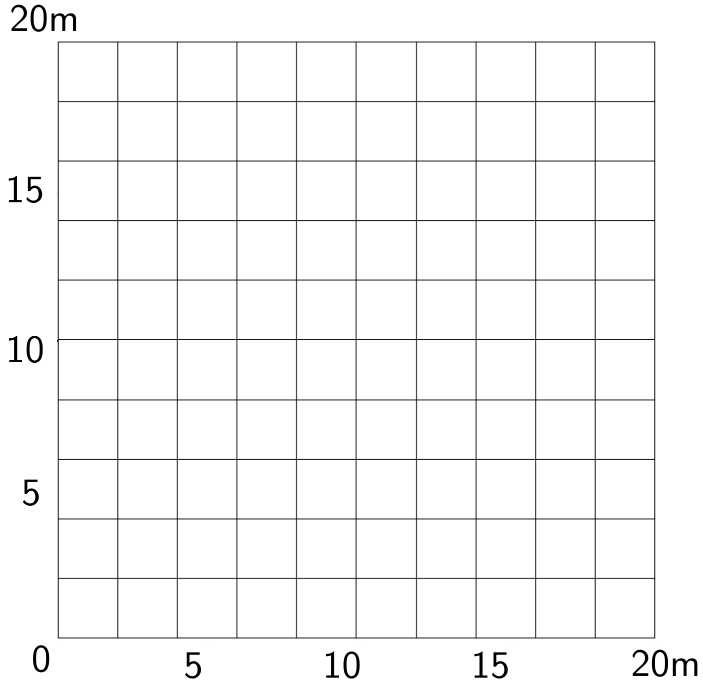


Figure 307.44: 8NodeBrick edge clamped square plate with element side length 2m.

Element type	8NodeBrick	8NodeBrick	8NodeBrick
Number of layers	1layer	2layers	4layers
Element side length	Height:1.00m	Height:0.50m	Height:0.25m
10m	95.57%	95.57%	95.57%
5m	85.09%	84.94%	84.91%
2m	52.98%	50.09%	49.25%
1m	28.93%	21.17%	18.72%
0.5m	18.26%	4.58%	3.56%
0.25m	15.05%	2.70%	1.37%

Table 307.16: Errors for 8NodeBrick square plate with four edges clamped.

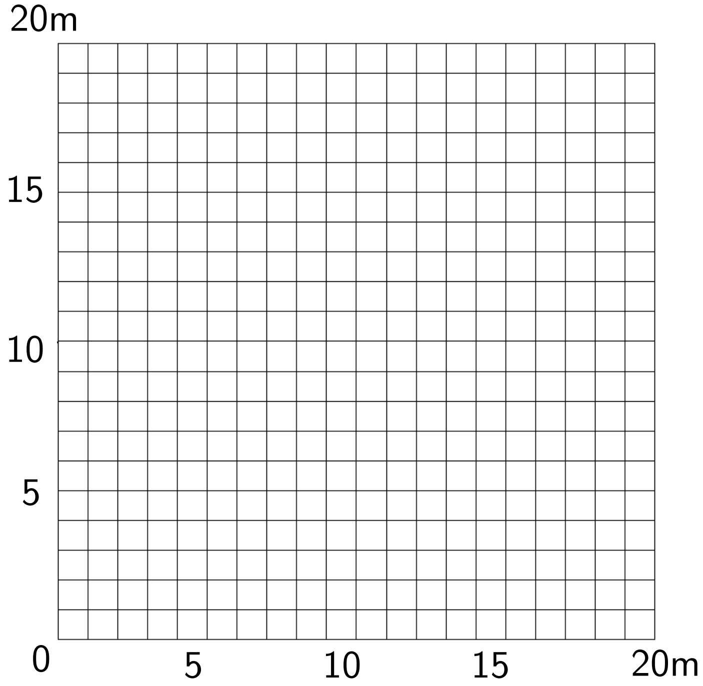


Figure 307.45: 8NodeBrick edge clamped square plate with element side length 1m.

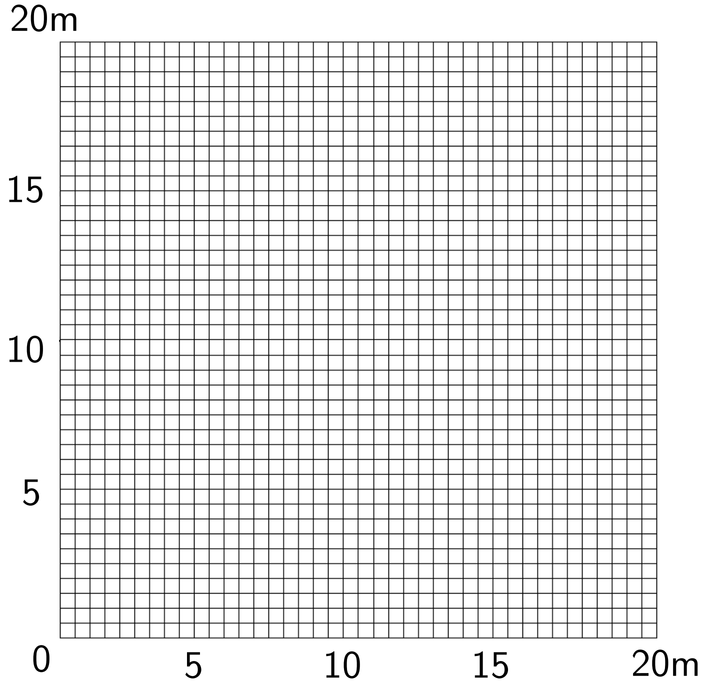


Figure 307.46: 8NodeBrick edge clamped square plate with element side length 0.5m.

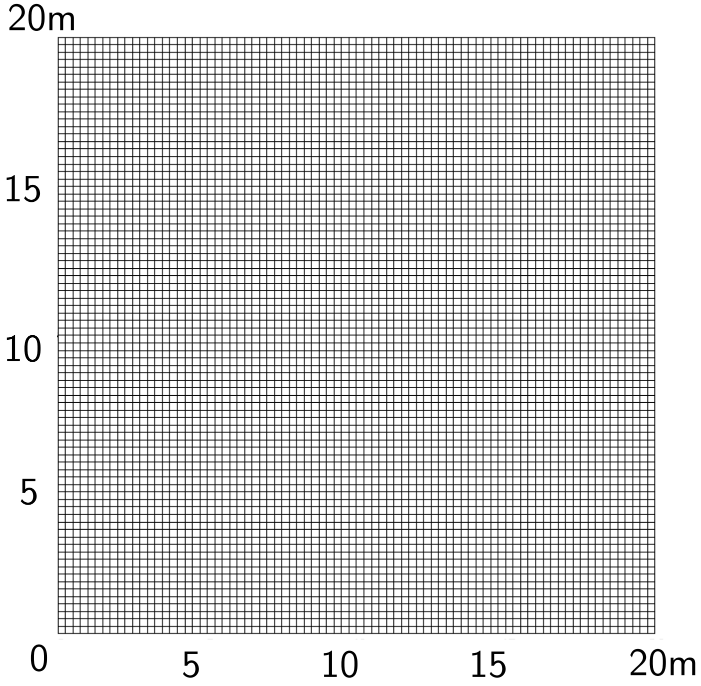


Figure 307.47: 8NodeBrick edge clamped square plate with element side length 0.25m.

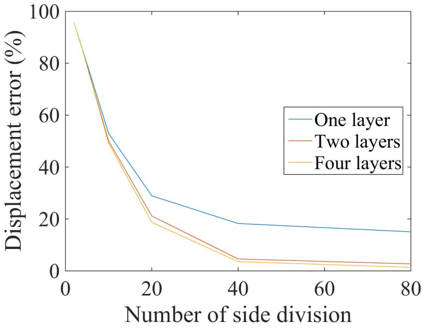


Figure 307.48: 8NodeBrick square plate with edge clamped˙ Displacement error versus Number of side division

307.2.9 Verification of 8NodeBrick square plate with four edges simply supported

Problem description: Length=20m, Width=20m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.3$.

The four edges are simply supported.

The load is the uniform normal pressure on the whole plate.

The plate flexural rigidity is

$$D = \frac{Eh^3}{12(1-\nu^2)} = \frac{10^8 N/m^2 \times 1^3 m^3}{12 \times (1-0.3^2)} = 9.1575 \times 10^6 \ N \cdot m \tag{307.35}$$

The theoretical solution is

$$d = \alpha_s \frac{qa^4}{D} = 0.00126 \times \frac{100N/m^2 \times 20^4 m^4}{9.1575 \times 10^6 \ N \cdot m} = 7.0936 \times 10^{-3} m \tag{307.36}$$

where α_s is a coefficient, which depends on the ratio of plate length to width. In this problem, the coefficient[5] α_s is 0.00126.

The 8NodeBrick are shown in Figures (307.49) - (307.54).

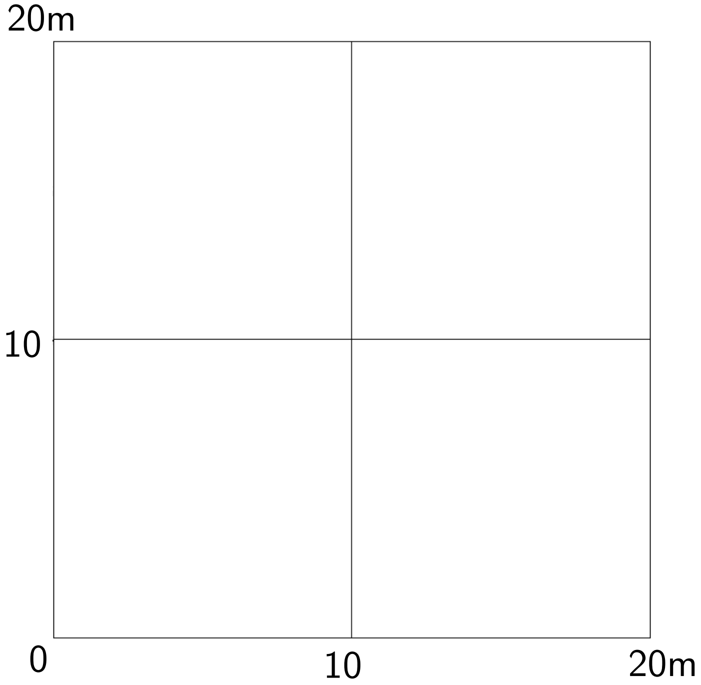


Figure 307.49: 8NodeBrick edge simply supported square plate with element side length 10m.

The results are listed in Table (307.17).

The errors are listed in Table (307.18).

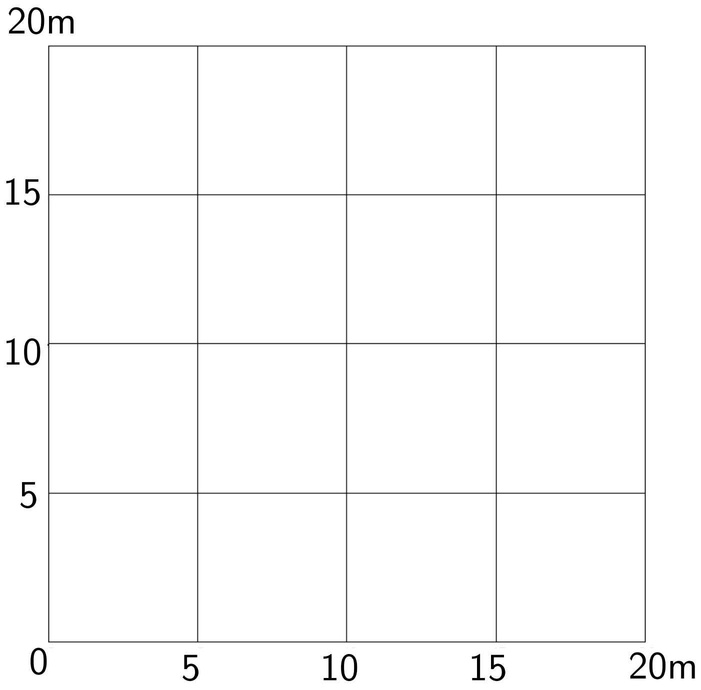


Figure 307.50: 8NodeBrick edge simply supported square plate with element side length 5m.

Table 307.17: Results for 8NodeBrick square plate with four edges simply supported.

Element type	8NodeBrick	8NodeBrick	Theoretical displacement
Number of layers	2layers	4layers	
Element side length	Height:0.50m	Height:0.25m	
10m	3.75E-004 m	3.76E-004 m	7.09E-03 m
5m	1.34E-003 m	1.35E-003 m	7.09E-03 m
2m	4.16E-003 m	4.27E-003 m	7.09E-03 m
1m	5.98E-003 m	6.22E-003 m	7.09E-03 m
0.5m	6.75E-003 m	7.04E-003 m	7.09E-03 m
0.25m	8.07E-003 m	7.30E-003 m	7.09E-03 m

The errors are plotted in Figure (307.55).

The Real-ESSI model fei/DSL files for the table above are HERE.

---

[5] Stephen Timoshenko, Theory of plates and shells (2nd edition). MrGRAW-Hill Inc, page202, 1959.

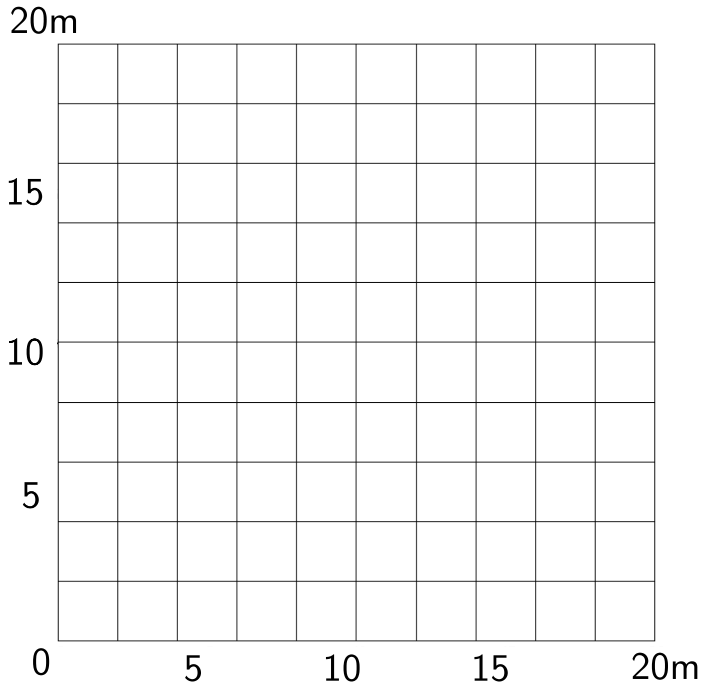


Figure 307.51: 8NodeBrick edge simply supported square plate with element side length 2m.

Table 307.18: Errors for 8NodeBrick square plate with four edges simply supported.

Element type	8NodeBrick	8NodeBrick
Number of layers	2layers	4layers
Element side length	Height:0.50m	Height:0.25m
10m	94.72%	94.71%
5m	81.05%	80.91%
2m	41.31%	39.79%
1m	15.64%	12.38%
0.5m	4.88%	0.70%
0.25m	13.74%	2.86%

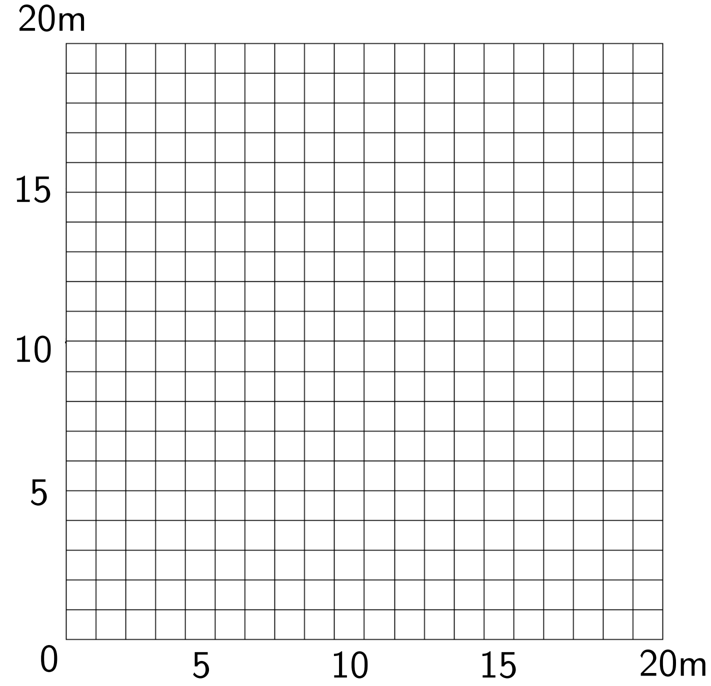


Figure 307.52: 8NodeBrick edge simply supported square plate with element side length 1m.

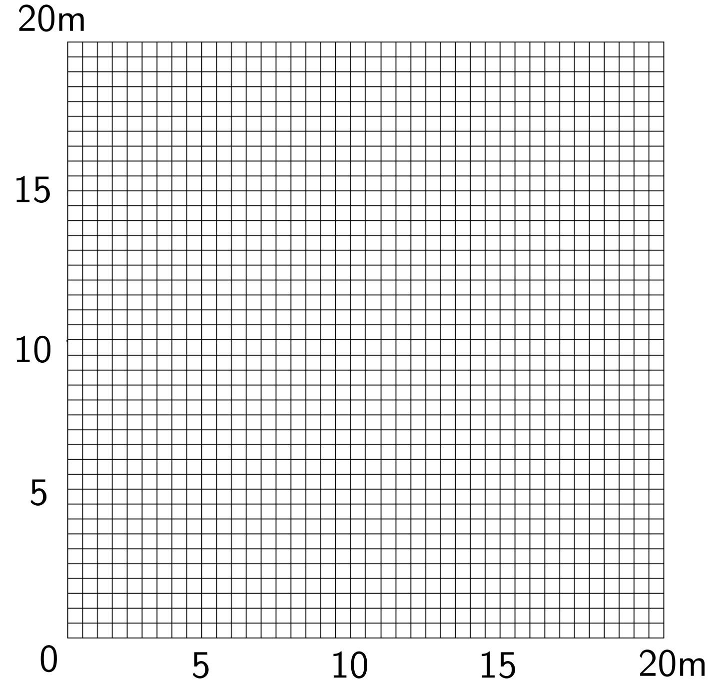


Figure 307.53: 8NodeBrick edge simply supported square plate with element side length 0.5m.

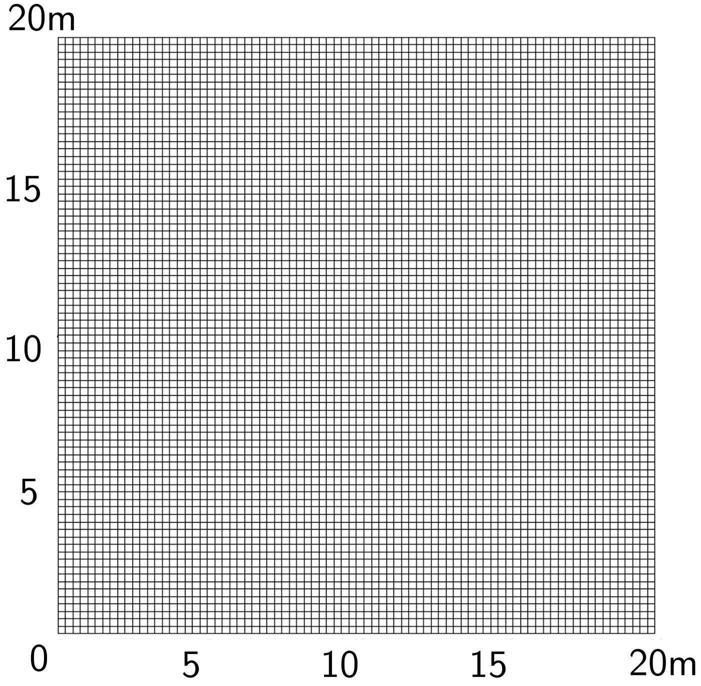


Figure 307.54: 8NodeBrick edge simply supported square plate with element side length 0.25m.

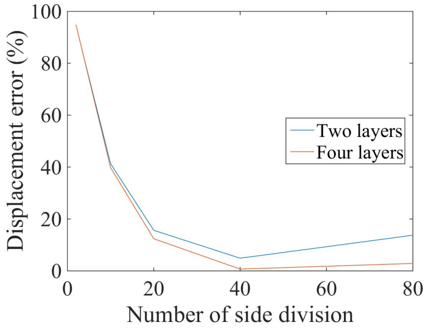


Figure 307.55: 8NodeBrick square plate with four edges simply supported˙ Displacement error versus Number of side division

307.2.10 Verification of 8NodeBrick circular plate with all edges clamped

Problem description: Diameter=20m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.3$.

The four edges are clamped.

The load is the uniform normal pressure on the whole plate.

The plate flexural rigidity is

$$D = \frac{Eh^3}{12(1-\nu^2)} = \frac{10^8 N/m^2 \times 1^3 m^3}{12 \times (1-0.3^2)} = 9.1575 \times 10^6 \; N \cdot m \tag{307.37}$$

The theoretical solution[6] is

$$d = \frac{qa^4}{64D} = \frac{100 N/m^2 \times 10^4 m^4}{64 \times 9.1575 \times 10^6 \; N \cdot m} = 1.7106 \times 10^{-3} m \tag{307.38}$$

The 8NodeBrick are shown in Figures (307.56) - (307.61).

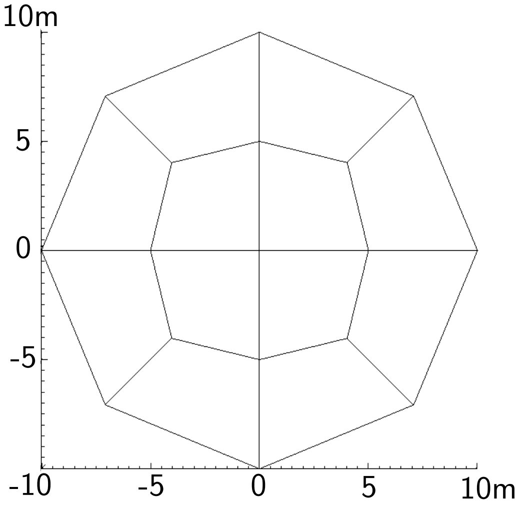


Figure 307.56: 8NodeBrick edge clamped circular plate with element side length 10m.

The results are listed in Table (307.19).

The errors are listed in Table (307.20).

The errors are shown in Figure (307.62).

The Real-ESSI model fei/DSL files for the table above are HERE.

---

[6] Stephen Timoshenko, Theory of plates and shells (2nd edition). MrGRAW-Hill Inc, page55, 1959.

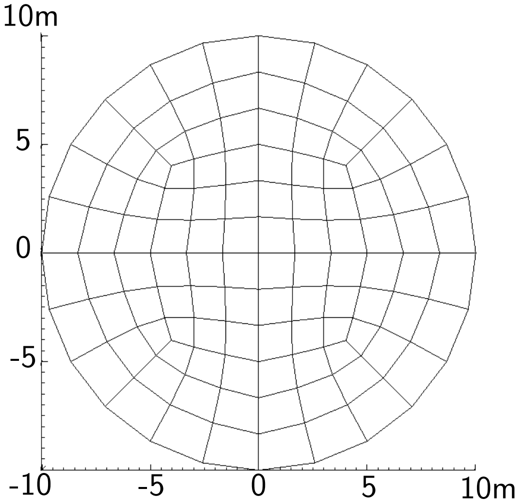


Figure 307.57: 8NodeBrick edge clamped circular plate with element side length 5m.

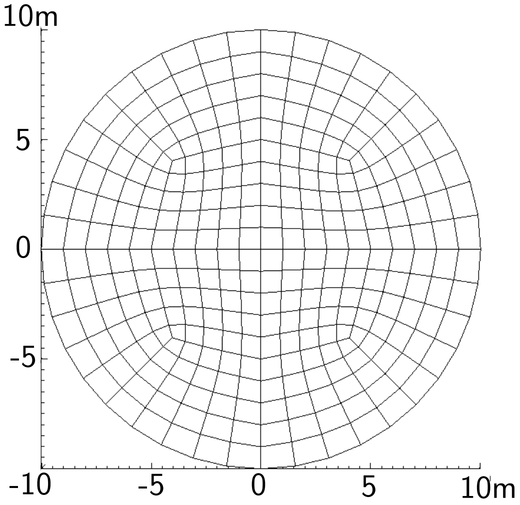


Figure 307.58: 8NodeBrick edge clamped circular plate with element side length 2m.

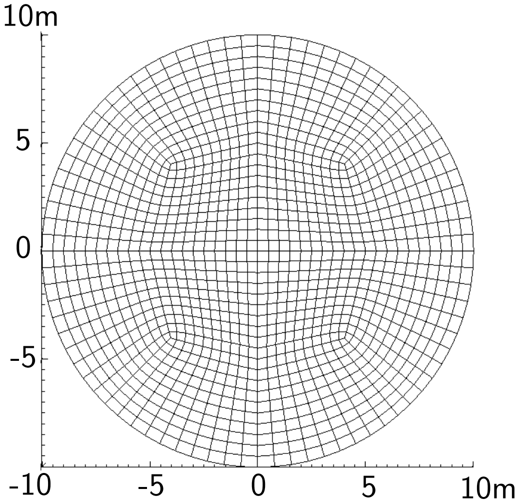


Figure 307.59: 8NodeBrick edge clamped circular plate with element side length 1m.

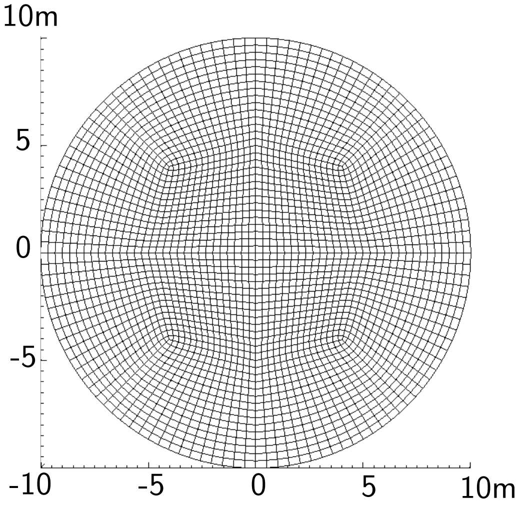


Figure 307.60: 8NodeBrick edge clamped circular plate with element side length 0.5m.

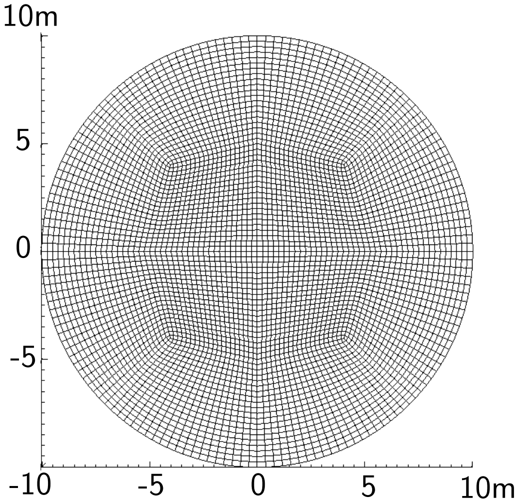


Figure 307.61: 8NodeBrick edge clamped circular plate with element side length 0.25m.

Table 307.19: Results for 8NodeBrick circular plate with four edges clamped.

Element type	8NodeBrick	8NodeBrick	8NodeBrick	Theoretical displacement
Number of layers	1layer	2layers	4layers	
Number of diameter divisions	Height:1.00m	Height:0.50m	Height:0.25m	
4	1.97E-04 m	1.99E-04 m	2.00E-04 m	1.71E-03 m
12	7.95E-04 m	8.47E-04 m	8.62E-04 m	1.71E-03 m
20	1.13E-03 m	1.25E-03 m	1.28E-03 m	1.71E-03 m
40	1.36E-03 m	1.54E-03 m	1.60E-03 m	1.71E-03 m
60	1.41E-03 m	1.62E-03 m	1.68E-03 m	1.71E-03 m
80	1.43E-03 m	1.64E-03 m	1.71E-03 m	1.71E-03 m

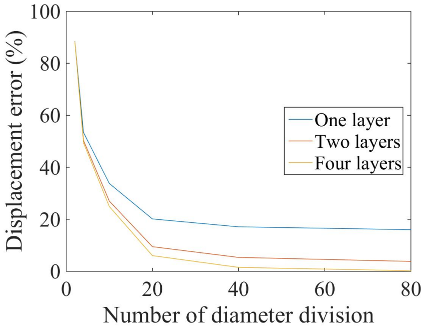


Figure 307.62: 8NodeBrick circular plate with edge clamped˙ Displacement error versus Number of side division.

Table 307.20: Errors for 8NodeBrick circular plate with four edges clamped.

Element type	8NodeBrick	8NodeBrick	8NodeBrick
Number of layers	1layer	2layers	4layers
Number of diameter divisions	Height:1.00m	Height:0.50m	Height:0.25m
4	88.43%	88.32%	88.30%
12	53.43%	50.35%	49.47%
20	33.79%	27.00%	24.93%
40	20.14%	9.47%	6.03%
60	17.11%	5.34%	1.51%
80	16.01%	3.80%	0.19%

307.2.11 Verification of 8NodeBrick circular plate with all edges simply supported

Problem description: Diameter=20m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.3$.

The four edges are simply supported.

The load is the uniform normal pressure on the whole plate.

The plate flexural rigidity is

$$D = \frac{Eh^3}{12(1-\nu^2)} = \frac{10^8 N/m^2 \times 1^3 m^3}{12 \times (1-0.3^2)} = 9.1575 \times 10^6 \ N \cdot m \tag{307.39}$$

The theoretical solution[7] is

$$d = \frac{(5+\nu)qa^4}{64(1+\nu)D} = \frac{(5+0.3) \times 100 N/m^2 \times 10^4 m^4}{64 \times (1+0.3) \times 9.1575 \times 10^6 \ N \cdot m} = 6.956 \times 10^{-3} m \tag{307.40}$$

The 8NodeBrick are shown in Figures (307.63) - (307.68).

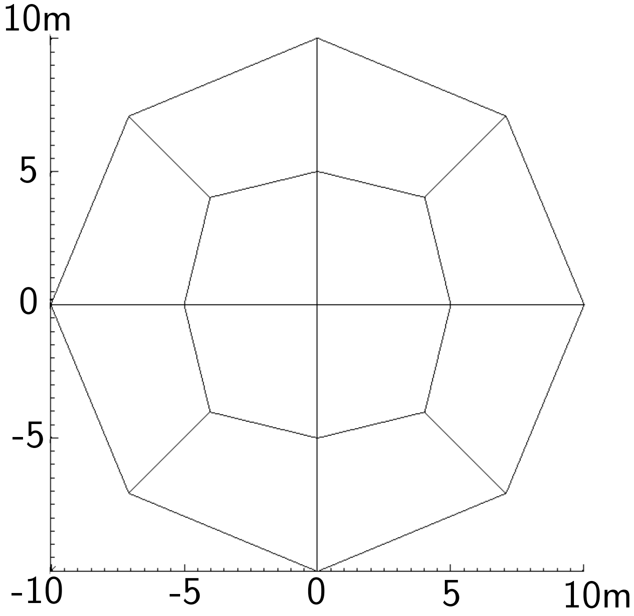


Figure 307.63: 8NodeBrick edge simply supported circular plate with element side length 10m.

The results are listed in Table (307.21).

The errors are listed in Table (307.22).

The errors are plotted in Figure (307.69).

The Real-ESSI model fei/DSL files for the table above are HERE.

---

[7] Stephen Timoshenko, Theory of plates and shells (2nd edition). MrGRAW-Hill Inc, page55, 1959.

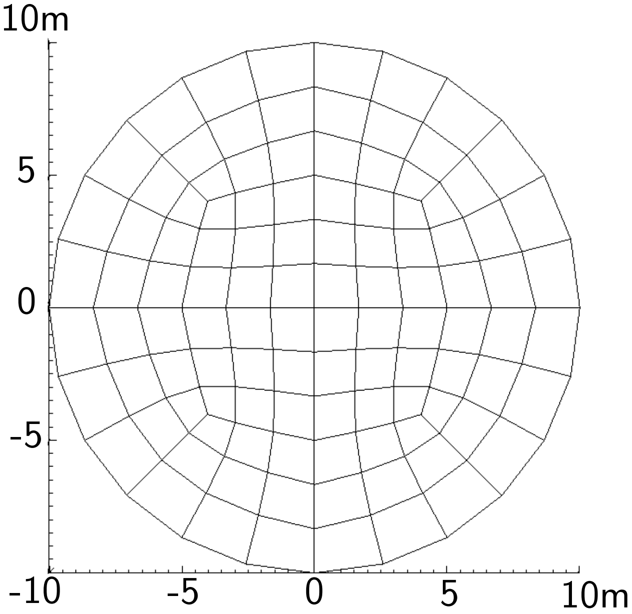


Figure 307.64: 8NodeBrick edge simply supported circular plate with element side length 5m.

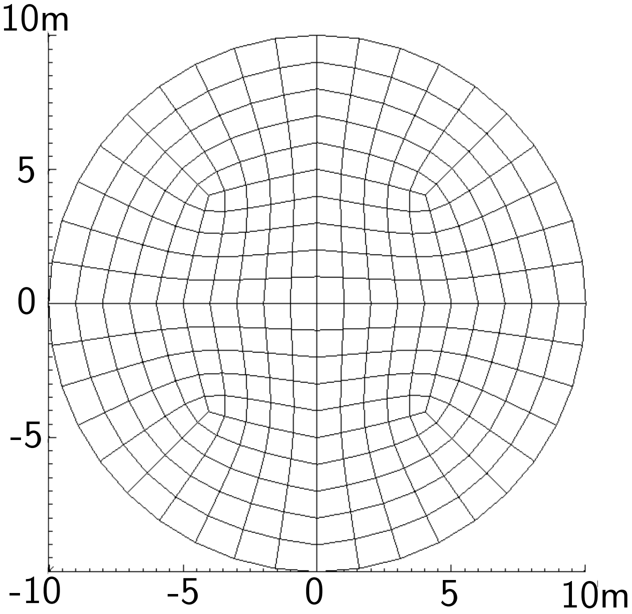


Figure 307.65: 8NodeBrick edge simply supported circular plate with element side length 2m.

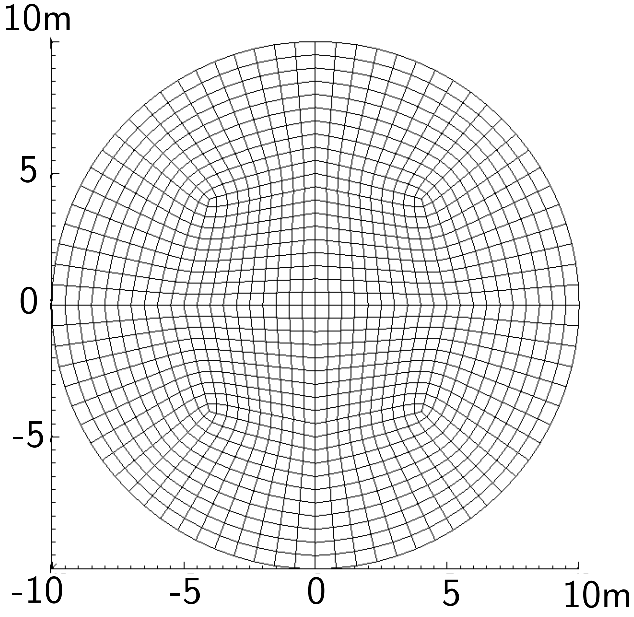


Figure 307.66: 8NodeBrick edge simply supported circular plate with element side length 1m.

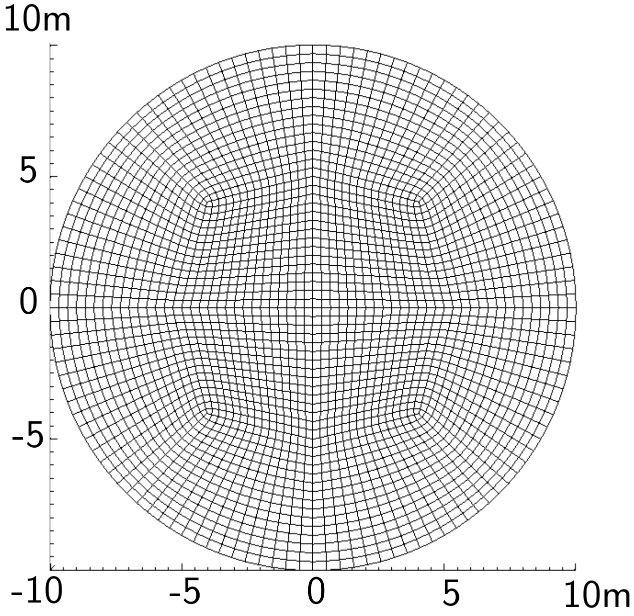


Figure 307.67: 8NodeBrick edge simply supported circular plate with element side length 0.5m.

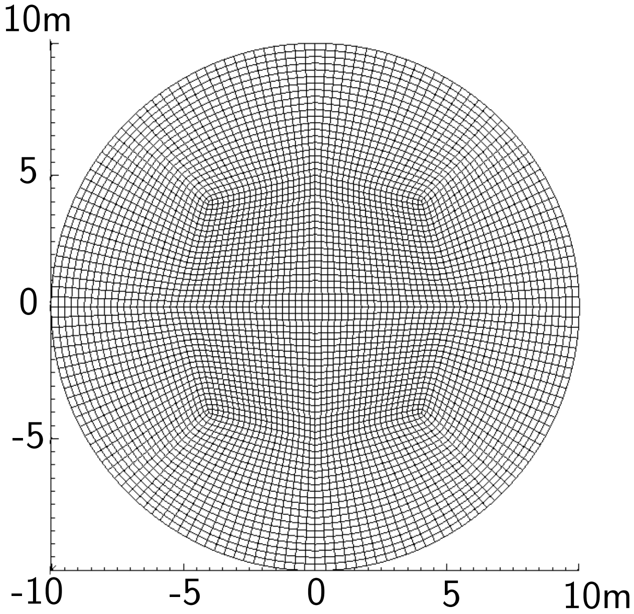


Figure 307.68: 8NodeBrick edge simply supported circular plate with element side length 0.25m.

Table 307.21: Results for 8NodeBrick cicular plate with four edges simply supported.

Element type	8NodeBrick	8NodeBrick	Theoretical displacement
Number of layers	2layers	4layers	
Number of diameter divisions	Height:0.50m	Height:0.25m	
4	6.35E-04 m	6.39E-04 m	6.96E-03 m
12	3.46E-03 m	3.57E-03 m	6.96E-03 m
20	4.96E-03 m	5.18E-03 m	6.96E-03 m
40	6.05E-03 m	6.37E-03 m	6.96E-03 m
60	6.30E-03 m	6.65E-03 m	6.96E-03 m
80	6.39E-03 m	6.76E-03 m	6.96E-03 m

Table 307.22: Errors for 8NodeBrick cicular plate with four edges simply supported.

Element type	8NodeBrick	8NodeBrick
Number of layers	2layers	4layers
Number of diameter divisions	Height:0.50m	Height:0.25m
4	90.87%	90.82%
12	50.19%	48.65%
20	28.64%	25.47%
40	13.09%	8.40%
60	9.45%	4.36%
80	8.10%	2.85%

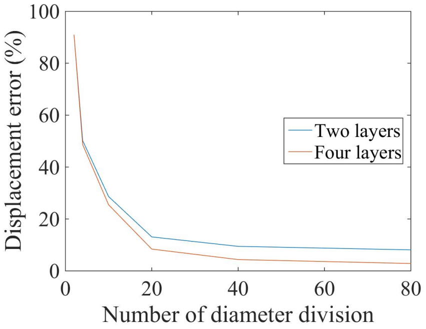


Figure 307.69: 8NodeBrick circular plate with edge simply supported˙ Displacement error versus Number of side division

307.2.12 Verification of 8NodeBrick Finite Element for Boussinesq Problem

307.2.12.1 Introduction

The Boussinesq problem is finding the displacement distribution in the isotropic linearly elastic half-space, subject to a concentrated load applied on the surface and perpendicular to it. The Boussinesq problem diagram is shown in Fig.(307.166).

Boussinesq problem is widely used in geotechnical engineering, especially when designing a foundation which transfers the superstructure load to the soil. To estimate the foundation settlements, it is important to have a reliable numerical solution for the Boussinesq problem.

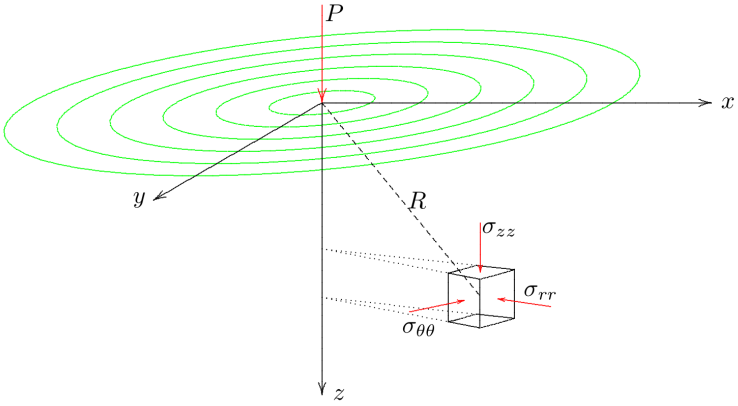


Figure 307.70: Boussinesq problem description. (Figure Reference: Verruijt, Arnold, and Stefan Van Baars. Soil mechanics. Delft, 2007.)

In 1885, the French scientist Joseph Boussinesq solved the analytic solutions of displacements in the homogeneous isotropic linear elastic half space. In general, the vertical displacement of the surface is

$$z = 0: \quad u_z = \frac{P(1-\nu^2)}{\pi E R} \tag{307.41}$$

where P is the vertical load, ν is the Poisson's ratio, E is the elastic modulus, and R is the distance from the measured point to the loading point.

In this section, the Real-ESSI numerical solution is verified by the analytic solution for the Boussinesq problem.

307.2.12.2 Description of the Verification Model

Since the problem is cylindrical symmetry, a quarter of the entire cube was employed to represent the whole cube. The reduced model was shown in Fig.(307.167).

The side length is 20 meters and the load P is 1N. The elastic modulus $E = 1 \times 10^3 Pa$ and the Poisson's ratio $\nu = 0.0$.

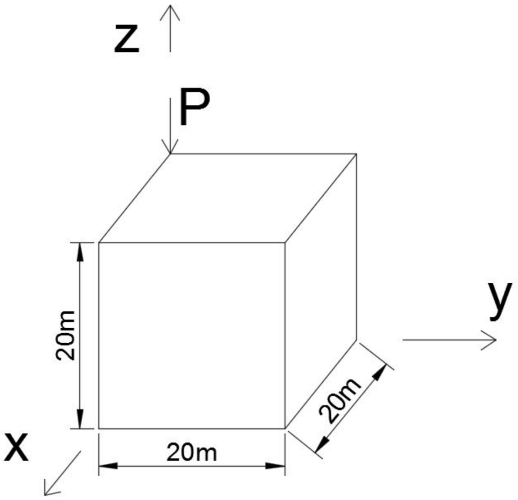


Figure 307.71: Reduced model (One quarter model) for the point load on the half space

The boundary conditions are shown in Fig.(307.72) and (307.73).

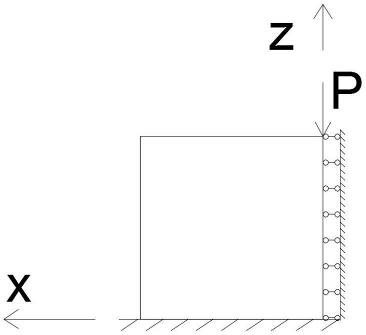


Figure 307.72: X-Z view for the reduced model

307.2.12.3 Results

Analytic solution for this model

According to the previous introduction, the analytic solution on the surface for this problem is

$$z = 0: \quad u_z = \frac{P(1-\nu^2)}{\pi E R} = \frac{1}{10^3 \pi} \frac{1}{R} \tag{307.42}$$

On the face $x = 0$, the distance R on the surface is actually the value of y, therefore, the analytic

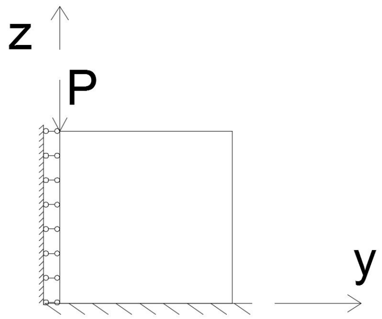


Figure 307.73: Y-Z view for the reduced model

solution is

$$u_z = \frac{1}{10^3 \pi R} = \frac{1}{10^3 \pi} \frac{1}{y} \tag{307.43}$$

As long as the y values are substituted, the displacement u_z is obtained immediately.

Real-ESSI solution with 8NodeBrickLT

In Real-ESSI, 8NodeBrickLT elements were used to simulate this model. Each element is $2m \times 2m \times 2m$. Since the model is $20m \times 20m \times 20m$, the element number is $10 \times 10 \times 10 = 1000$. The vertical displacement at the surface was recorded.

Since the model is symmetric, when the results were plotted, the other half results were obtained by symmetry.

Comparison between the analytic and 8NodeBrickLT solution

The Real-ESSI and analytic results were plotted in Fig.(307.74). Note that the analytic solution for location $y = 0$ is infinity, which was not plotted in the figure below.

307.2.12.4 Error Analysis

1. Mismatch at the loading point.

   First of all, when $x = 0$ at the loading point, the analytic solution is infinite. From the perspective of practical engineering, this analytic solution is flawed because the displacement cannot be infinite. The infinite solution is due to the elastic assumption. In consideration of the plasticity, the analytic solution will not be infinite.

   In Real-ESSI, the displacement at the loading point is not infinite because the infinite value is averaged by the integration during the finite element calculation. Also, at the loading point,

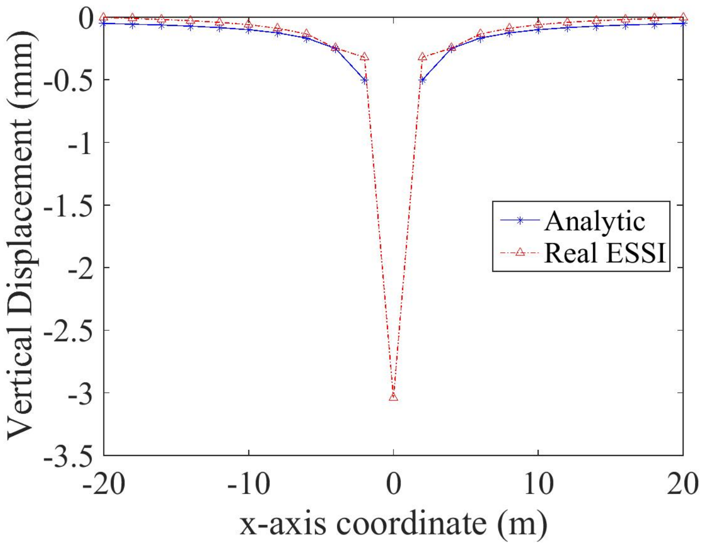


Figure 307.74: Comparison between analytic solution and Real-ESSI 8NodeBrickLT solution

27NodeBrickLT has a much larger displacement than that of 8NodeBrickLT. This is because 27NodeBrickLT has a relatively denser mesh than 8NodeBrickLT. So the maximum value at the loading point is higher than that of 8NodeBrickLT.

2. Mismatch at other locations.

   Except at the loading point, the analytic solution is not exactly equal to the numerical solution at other locations. This is because the verification example employs a simplified bounded cube to represent unbounded half space. The original analytic solution is for the half space, which is not true for the verification model. Not only the horizontal space but also the bottom space are removed from the model. This means the analytic solution is not perfect for this bounded cube. However, since the cube is very great, the analytic solution is similar to the Real-ESSI numerical solution. In addition, the brick elements are also verified by other models, like beam, plate and shells.

307.2.13 Verification of 8NodeBrick Finite Element for Collapsed Brick Shapes

307.2.13.1 Test procedure

With reference to the "patch test" put forward by Taylor et al. (1986), the test procedures are:

- A standard solution is given by testing two different loading modes on a single normal 8 node brick element: (1) Pure confinement loading, where same pressure are applied on three different directions. (2) Simple shearing, where shearing force is applied on four nodes of top layer, while four nodes on bottom layer are totally fixed. Linear elastic material is adopted here with Young's modulus $E = 125\text{MPa}$ and Poisson's ratio $\mu = 0.25$. The setup of standard test is shown in Figure 307.75.

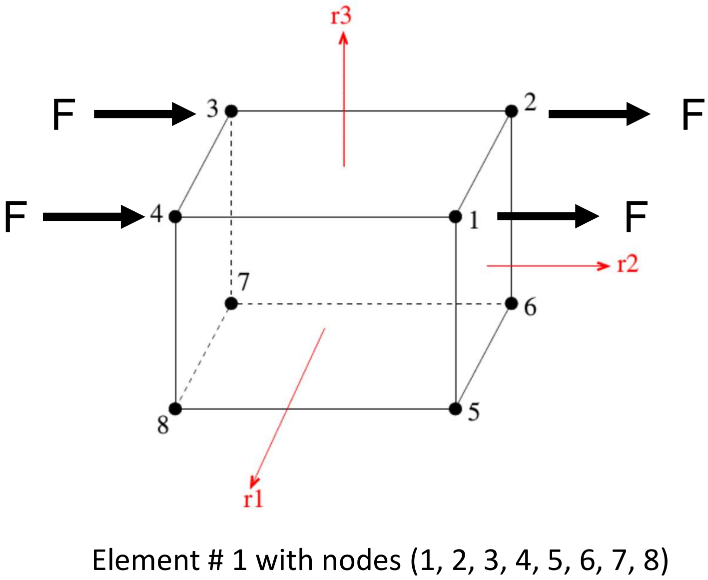


Figure 307.75: Setup of standard 8-node element.

- Build the same geometric model with collapsed 8 node brick elements and conduct numerical simulation under the same loading and boundary conditions as first step.

  Specifically, the geometric configure for 7-node collapsed element is shown in Figure 307.76, where the cubic consists of two 7-node collapsed elements. A dummy node 11 is generated at the same location as node 2.

  The geometric configuration for 6-node collapsed element is shown in Figure 307.77. Again the cubic is composed of two 6-node collapsed elements. Two dummy Real-ESSI nodes 9 and 10 are

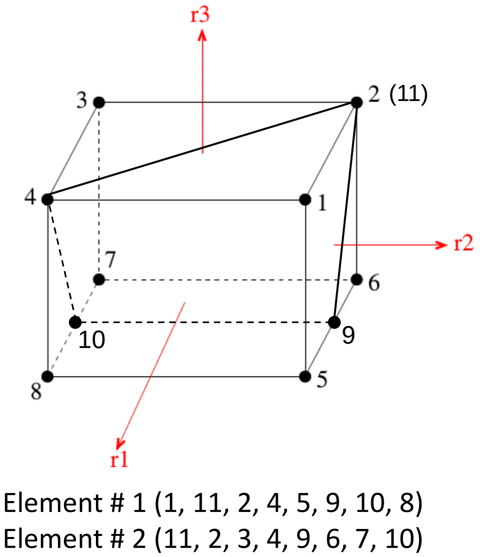


Figure 307.76: Geometric configuration of numerical test for 7-node collapsed element.

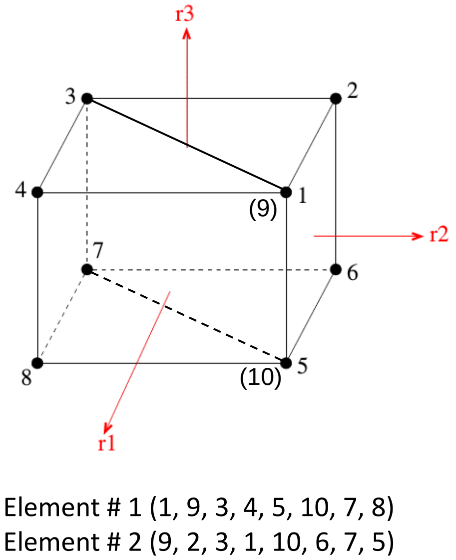
generated at the same location with original node 1 and 2.

And Figure 307.78 gives the geometric configuration of 5-node collapsed element, where the cubic is divided into 3 5-node elements.

- Compare the response of models in step 1 and step 2. If the difference is small enough, the strategy described in section ?? is feasible and valid.

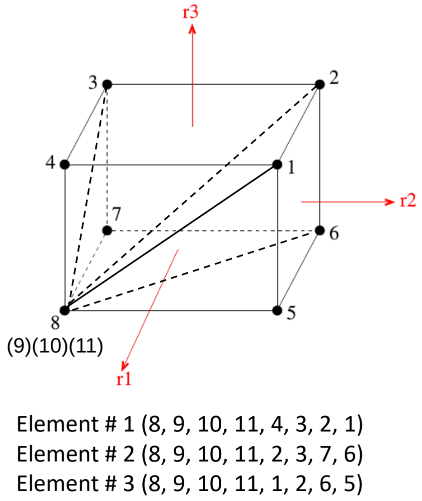
307.2.13.2 Test result

The comparison of displacement response for confinement loading is shown in Figure 307.79. Figure 307.80 demonstrates the test results of simple shearing loading. It can be seen that the simulation results of these types of collapsed element are close to result of standard 8 node brick element. The line of 6 node collapsed element is almost overlap with the line of standard test. Collapsed 7-node element and 5-node element experience certain decrease of accuracy. The main error reflects on the decrease of stiffness. Both bulk modulus and shear modulus of 7-node element and 5-node element are around 7% lower than the standard 8-node element.



Element # 1 (1, 9, 3, 4, 5, 10, 7, 8)
Element # 2 (9, 2, 3, 1, 10, 6, 7, 5)

Figure 307.77: Geometric configuration of numerical test for 6-node collapsed element.



Element # 1 (8, 9, 10, 11, 4, 3, 2, 1)
Element # 2 (8, 9, 10, 11, 2, 3, 7, 6)
Element # 3 (8, 9, 10, 11, 1, 2, 6, 5)

Figure 307.78: Geometric configuration of numerical test for 5-node collapsed element.

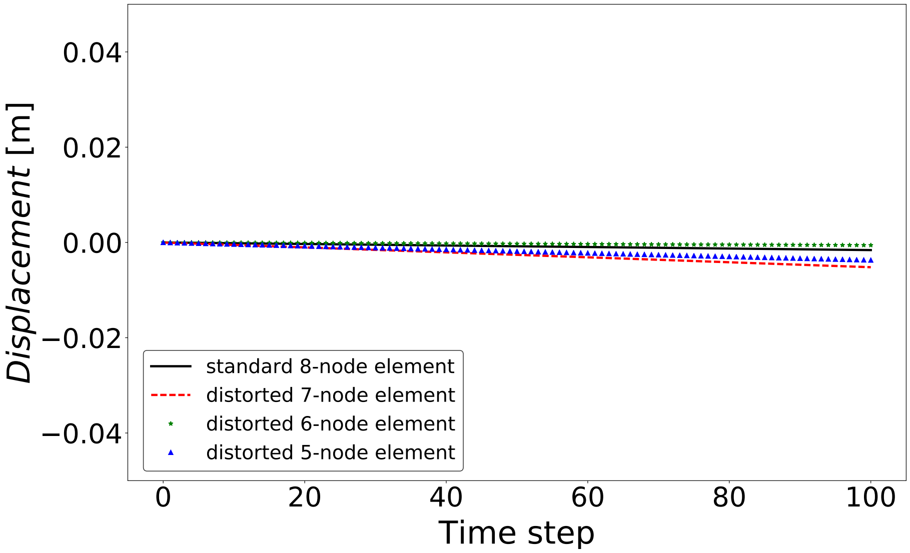


Figure 307.79: Comparison of displacement response under confinement loading.

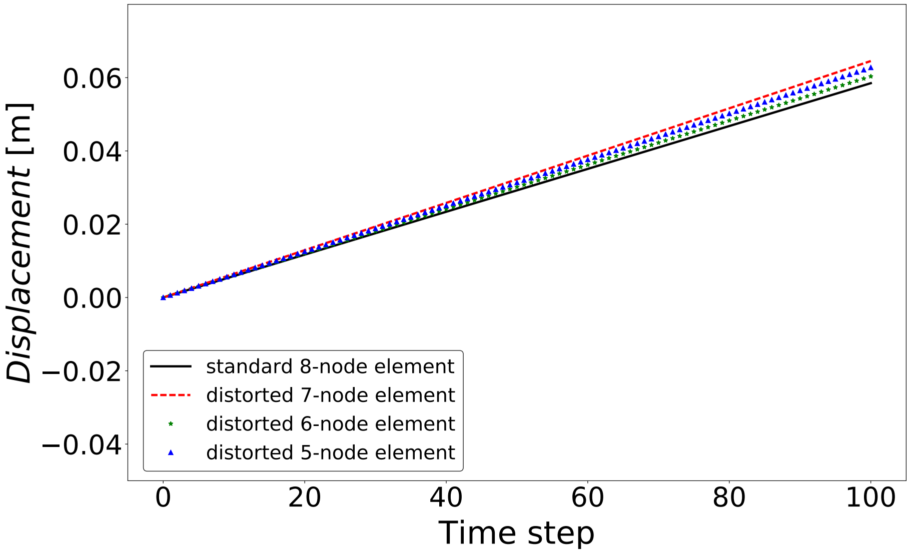


Figure 307.80: Comparison of displacement response under simple shearing.

307.2.14 Verification of 27 node brick cantilever beam (static)

Problem description: Length=6m, Width=1m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.0$. Use the shear deformation coefficient $\kappa = 1.2$. The force direction was shown in Figure (307.81).

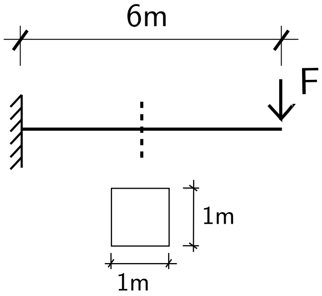


Figure 307.81: Problem description for cantilever beams.

Theoretical displacement (bending and shear deformation):

$$d = \frac{FL^3}{3EI} + \frac{FL}{GA_v} = \frac{FL^3}{3E\frac{bh^3}{12}} + \frac{FL}{\frac{E}{2(1+\nu)}\frac{bh}{\kappa}} = \frac{100N \times 6^3 m^3}{3 \times 10^8 N/m^2 \times \frac{1}{12} m^4} + \frac{100N \times 6m}{\frac{10}{2} \times 10^7 N/m^2 \times 1m^2 \times \frac{5}{6}} = 8.64 \times 10^{-4} m + 0.144 \times 10^{-4} m = \tag{307.44}$$

Numerical model:

The 27NodeBrick elements are shown in Figure (307.85).

All the Real-ESSI results are listed in Table (307.23).

Table 307.23: Results for 27NodeBrick cantilever beams of different element numbers.

Element number	1	2	6
27NodeBrick	7.07E-04 m	8.50E-04 m	8.75E-04 m
Error	19.52%	3.19%	0.34%

The errors are plotted in Figure (307.88).

The Real-ESSI model fei/DSL files for the table above are HERE.

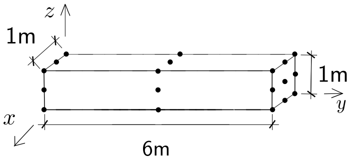


Figure 307.82: One 27NodeBrick element.

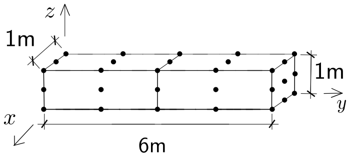


Figure 307.83: Two 27NodeBrick elements.

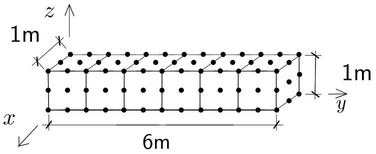


Figure 307.84: Six 27NodeBrick elements.

Figure 307.85: 27NodeBrick elements for cantilever beams.

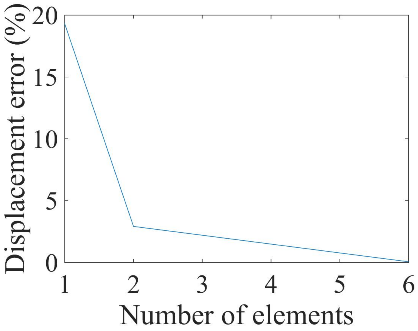


Figure 307.86: Error scale 0% - 20%.

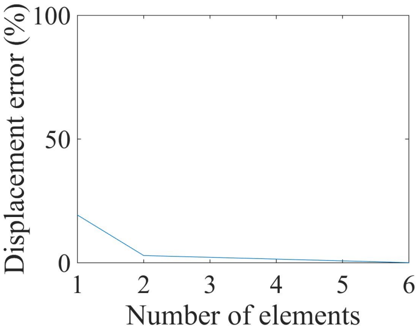


Figure 307.87: Error scale 0% - 100%.

Figure 307.88: 27NodeBrick cantilever beam for different element number˙ Displacement error versus Number of elements

307.2.15 Verification of 27NodeBrick cantilever beam for different Poisson's ratio

Problem description: Length=6m, Width=1m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.0 - 0.49$. The force direction was shown in Figure (307.89).

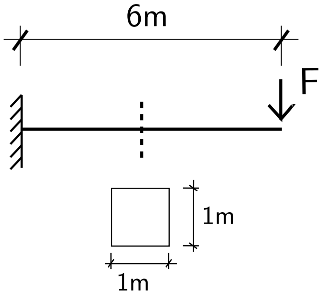


Figure 307.89: Problem description for cantilever beams of different Poisson's ratios.

The theoretical solution for $\nu = 0.0$ was calculated below, while the solution for other Poisson's ratio is calculated by the similar process.

Theoretical displacement (bending and shear deformation):

$$\begin{aligned} d &= \frac{FL^3}{3EI} + \frac{FL}{GA_v} \\ &= \frac{FL^3}{3E\frac{bh^3}{12}} + \frac{FL}{\frac{E}{2(1+\nu)}\frac{bh}{\kappa}} \\ &= \frac{100N \times 6^3m^3}{3 \times 10^8N/m^2 \times \frac{1}{12}m^4} + \frac{100N \times 6m}{\frac{10}{2} \times 10^7N/m^2 \times 1m^2 \times \frac{5}{6}} \\ &= 8.64 \times 10^{-4}m + 0.144 \times 10^{-4}m \\ &= 8.784 \times 10^{-4}\ m \end{aligned} \tag{307.45}$$

The rotation angle at the end:

$$\theta = \frac{FL^2}{2EI} = \frac{100N \times 6^2m^2}{2 \times 10^8N/m^2 \times \frac{1}{12}m^4} = 2.16 \times 10^{-4}\ rad = 0.0124^{\circ} \tag{307.46}$$

The 27NodeBrick elements for cantilever beams of different Poisson's ratios are shown in Figure (307.90).

All the displacement results are listed in Table (307.24).

The errors are plotted in Figure (307.93).

The angle results are listed in Table (307.27).

The errors are plotted in Figure (307.96).

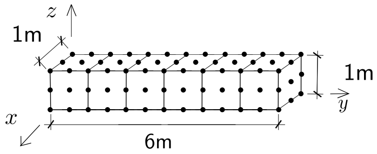


Figure 307.90: 27NodeBrick elements for cantilever beams of different Poisson's ratios.

Table 307.24: Displacement results for 27NodeBrick cantilever beams with element side length 1 m.

Poisson's ratio	27NodeBrick displacement	Theory displacement (bending)	Theory displacement (shear)	Theory displacement(all)	Error
0.00	8.755E-04 m	8.640E-04 m	1.440E-05 m	8.784E-04 m	0.34%
0.05	8.757E-04 m	8.640E-04 m	1.512E-05 m	8.791E-04 m	0.39%
0.10	8.751E-04 m	8.640E-04 m	1.586E-05 m	8.799E-04 m	0.54%
0.15	8.735E-04 m	8.640E-04 m	1.659E-05 m	8.806E-04 m	0.80%
0.20	8.708E-04 m	8.640E-04 m	1.734E-05 m	8.813E-04 m	1.19%
0.25	8.667E-04 m	8.640E-04 m	1.808E-05 m	8.821E-04 m	1.74%
0.30	8.608E-04 m	8.640E-04 m	1.884E-05 m	8.828E-04 m	2.50%
0.35	8.520E-04 m	8.640E-04 m	1.959E-05 m	8.836E-04 m	3.57%
0.40	8.385E-04 m	8.640E-04 m	2.035E-05 m	8.844E-04 m	5.18%
0.45	8.147E-04 m	8.640E-04 m	2.111E-05 m	8.851E-04 m	7.96%
0.49	7.711E-04 m	8.640E-04 m	2.173E-05 m	8.857E-04 m	12.94%

The Real-ESSI model fei/DSL files for the table above are HERE.

Then, different values of elastic modulus were also tried. The errors are plotted below.

According to Fig.(307.97)), the different values of elastic modulus will not influence the error.

However, the different Poisson's ratio will influence the error. The error will increase with the Poisson's ratio increase.

Table 307.25: Displacement results for 27NodeBrick cantilever beams with element side length 0.5 m.

Poisson's ratio	27NodeBrick displacement	Theory displacement (bending)	Theory displacement (shear)	Theory displacement(all)	Error
0.00	8.804E-04 m	8.640E-04 m	1.440E-05 m	8.784E-04 m	0.23%
0.05	8.808E-04 m	8.640E-04 m	1.512E-05 m	8.791E-04 m	0.19%
0.10	8.805E-04 m	8.640E-04 m	1.586E-05 m	8.799E-04 m	0.08%
0.15	8.796E-04 m	8.640E-04 m	1.659E-05 m	8.806E-04 m	0.12%
0.20	8.778E-04 m	8.640E-04 m	1.734E-05 m	8.813E-04 m	0.40%
0.25	8.752E-04 m	8.640E-04 m	1.808E-05 m	8.821E-04 m	0.78%
0.30	8.715E-04 m	8.640E-04 m	1.884E-05 m	8.828E-04 m	1.28%
0.35	8.663E-04 m	8.640E-04 m	1.959E-05 m	8.836E-04 m	1.95%
0.40	8.588E-04 m	8.640E-04 m	2.035E-05 m	8.844E-04 m	2.89%
0.45	8.465E-04 m	8.640E-04 m	2.111E-05 m	8.851E-04 m	4.36%
0.49	8.248E-04 m	8.640E-04 m	2.173E-05 m	8.857E-04 m	6.88%

Table 307.26: Displacement results for 27NodeBrick cantilever beams with element side length 0.25 m.

Poisson's ratio	27NodeBrick displacement	Theory displacement (bending)	Theory displacement (shear)	Theory displacement(all)	Error
0.00	8.797E-04 m	8.640E-04 m	1.440E-05 m	8.784E-04 m	0.15%
0.05	8.801E-04 m	8.640E-04 m	1.512E-05 m	8.791E-04 m	0.11%
0.10	8.799E-04 m	8.640E-04 m	1.586E-05 m	8.799E-04 m	0.01%
0.15	8.792E-04 m	8.640E-04 m	1.659E-05 m	8.806E-04 m	0.16%
0.20	8.778E-04 m	8.640E-04 m	1.734E-05 m	8.813E-04 m	0.40%
0.25	8.758E-04 m	8.640E-04 m	1.808E-05 m	8.821E-04 m	0.71%
0.30	8.730E-04 m	8.640E-04 m	1.884E-05 m	8.828E-04 m	1.12%
0.35	8.692E-04 m	8.640E-04 m	1.959E-05 m	8.836E-04 m	1.63%
0.40	8.641E-04 m	8.640E-04 m	2.035E-05 m	8.844E-04 m	2.29%
0.45	8.567E-04 m	8.640E-04 m	2.111E-05 m	8.851E-04 m	3.21%
0.49	8.452E-04 m	8.640E-04 m	2.173E-05 m	8.857E-04 m	4.58%

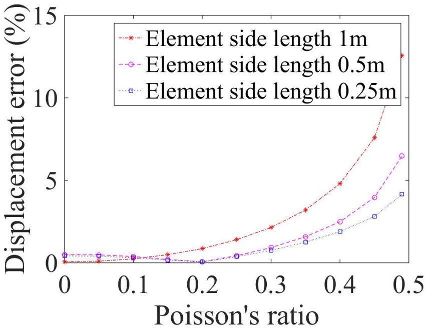


Figure 307.91: Error scale 0% - 15%.

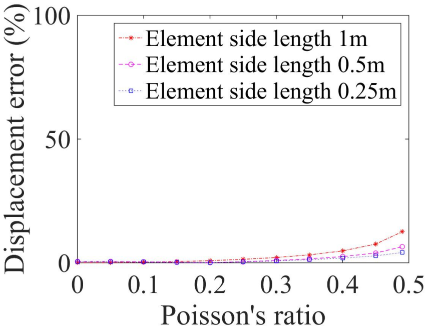


Figure 307.92: Error scale 0% - 100%.

Figure 307.93: 27NodeBrick cantilever beam for different Poisson's ratio: Displacement error versus Poisson's ratio

Table 307.27: Rotation angle results for 27NodeBrick cantilever beams with element side length 1 m.

Poisson's ratio	27NodeBrick angle (unit:°)	Theory angle (unit:°)	Error
0.00	1.238E-02	1.24E-02	0.19%
0.05	1.237E-02	1.24E-02	0.24%
0.10	1.236E-02	1.24E-02	0.34%
0.15	1.233E-02	1.24E-02	0.53%
0.20	1.230E-02	1.24E-02	0.80%
0.25	1.225E-02	1.24E-02	1.18%
0.30	1.219E-02	1.24E-02	1.70%
0.35	1.210E-02	1.24E-02	2.45%
0.40	1.196E-02	1.24E-02	3.55%
0.45	1.172E-02	1.24E-02	5.47%
0.49	1.130E-02	1.24E-02	8.89%

Table 307.28: Rotation angle results for 27NodeBrick cantilever beams with element side length 0.5 m.

Poisson's ratio	27NodeBrick angle (unit:°)	Theory angle (unit:°)	Error
0.00	1.242E-02	1.24E-02	0.12%
0.05	1.241E-02	1.24E-02	0.11%
0.10	1.241E-02	1.24E-02	0.06%
0.15	1.239E-02	1.24E-02	0.05%
0.20	1.237E-02	1.24E-02	0.21%
0.25	1.235E-02	1.24E-02	0.44%
0.30	1.231E-02	1.24E-02	0.74%
0.35	1.226E-02	1.24E-02	1.16%
0.40	1.218E-02	1.24E-02	1.76%
0.45	1.206E-02	1.24E-02	2.76%
0.49	1.183E-02	1.24E-02	4.63%

Table 307.29: Rotation angle results for 27NodeBrick cantilever beams with element side length 0.25 m.

Poisson's ratio	27NodeBrick angle(unit:^{circ})	Theory angle (unit:$^{\circ}$)	Error
0.00	1.242E-02	1.24E-02	0.17%
0.05	1.242E-02	1.24E-02	0.15%
0.10	1.241E-02	1.24E-02	0.09%
0.15	1.240E-02	1.24E-02	0.02%
0.20	1.238E-02	1.24E-02	0.17%
0.25	1.235E-02	1.24E-02	0.38%
0.30	1.232E-02	1.24E-02	0.64%
0.35	1.228E-02	1.24E-02	0.98%
0.40	1.222E-02	1.24E-02	1.42%
0.45	1.214E-02	1.24E-02	2.06%
0.49	1.202E-02	1.24E-02	3.08%

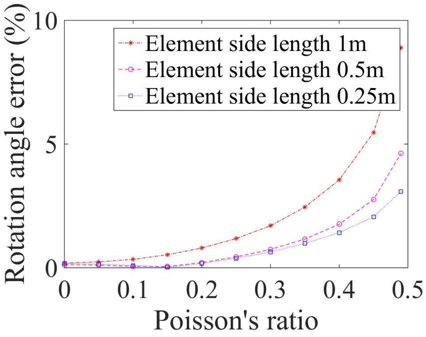


Figure 307.94: Error scale 0% - 10%.

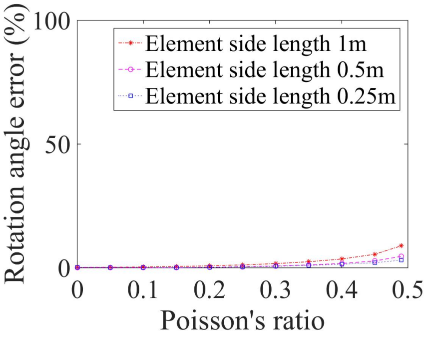


Figure 307.95: Error scale 0% - 100%.

Figure 307.96: 27NodeBrick cantilever beam for different Poisson's ratio˙ Rotation angle error versus Poisson's ratio

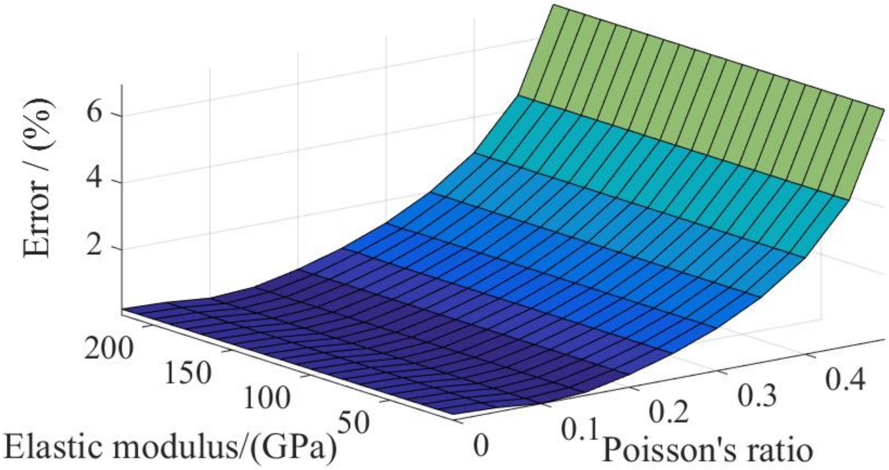


Figure 307.97: The influence of Poisson's ratio and elastic modulus on the errors

.

307.2.16 Test of irregular shaped 27NodeBrick cantilever beams

Cantilever model was used as an example. Three different shapes are tested.

In the first test, the upper two nodes of each element were moved one half element size along the $y - axis$, while the lower two nodes were kept at the same location. The element shape was shown in Figure (307.98).

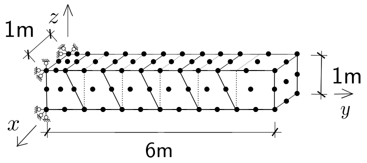


Figure 307.98: 27NodeBrick cantilever beams for irregular Shape 1.

In the second test, the upper two nodes of each element were moved 90% element size along the $y-axis$, while the lower two nodes were moved 90% element size in the other direction along the $y-axis$. The element shape was shown in Figure (307.99).

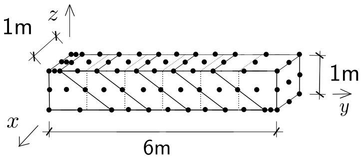


Figure 307.99: 27NodeBrick cantilever beams for irregular Shape 2.

In the third test, the upper two nodes of each element are moved one half element size with different directions along the $y - axis$, while the lower two nodes were kept at the same location. The element shape was shown in Figure (307.100).

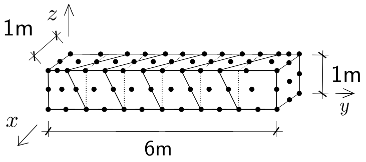


Figure 307.100: 27NodeBrick cantilever beams for irregular Shape 3.

The boundary conditions are shown in Figure (307.103), (307.106) and (307.109) .

The Real-ESSI results are listed in Table (307.30).

The errors are listed in Table (307.31) and (307.32).

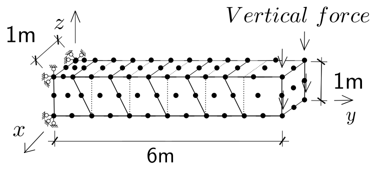


Figure 307.101: Veritical force.

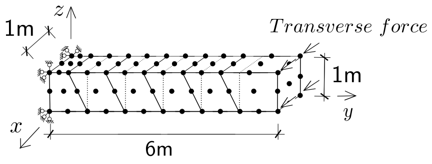


Figure 307.102: Horizontal force.

Figure 307.103: 27NodeBrick cantilever beam boundary conditions for irregular Shape 1.

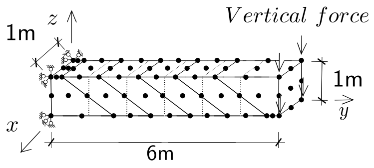


Figure 307.104: Veritical force.

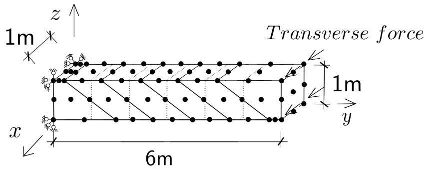


Figure 307.105: Horizontal force.

Figure 307.106: 27NodeBrick cantilever beam boundary conditions for irregular Shape 2.

Table 307.30: Results for 27NodeBrick cantilever beams of irregular shapes.

Displacements for irregular shaped element					
Element Type	Force direction	Normal shape	Shape 1	Shape 2	Shape 3
27NodeBrick	Vertical (z)	8.755E-04 m	8.819E-04 m	8.709E-04 m	8.837E-04 m
27NodeBrick	Transverse (y)	8.755E-04 m	8.831E-04 m	8.462E-04 m	8.824E-04 m
Theoretical	-	8.784E-04 m	8.784E-04 m	8.784E-04 m	8.784E-04 m

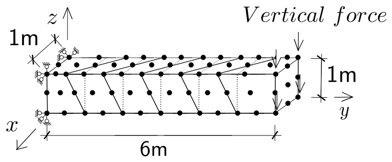


Figure 307.107: Veritical force.

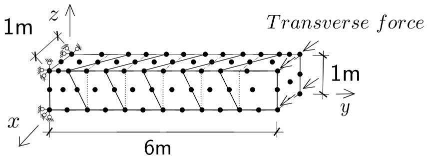


Figure 307.108: Horizontal force.

Figure 307.109: 27NodeBrick cantilever beam boundary conditions for irregular Shape 3.

Table 307.31: Errors for irregular shaped 27NodeBrick compared to theoretical solution.

Errors for irregular shaped element, compared to theoretical solutions					
Element Type	Force direction	Normal shape	Shape 1	Shape 2	Shape 3
27NodeBrick	Vertical (z)	0.34%	0.40%	0.85%	0.60%
27NodeBrick	Transverse (y)	0.34%	0.54%	3.67%	0.46%

Table 307.32: Errors for irregular shaped 27NodeBrick compared to normal shape.

Errors for irregular shaped element, compared to normal shape					
Element Type	Force direction	Normal shape	Shape 1	Shape 2	Shape 3
27NodeBrick	Vertical (z)	0.00%	0.74%	0.52%	0.94%
27NodeBrick	Transverse (y)	0.00%	0.87%	3.34%	0.79%

The Real-ESSI model fei/DSL files for the table above are HERE.

Then, the beam was divided into small elements.

Problem description: Length=12m, Width=2m, Height=2m, Force=400N/m, E=1E8Pa, $\nu = 0.0$. Use the shear deformation coefficient $\kappa = 1.2$. The force direction was shown in Figure (307.110).

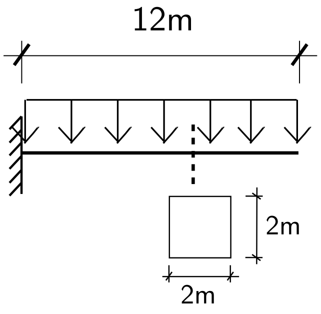


Figure 307.110: Problem description for cantilever beams under uniform pressure .

Theoretical displacement (bending and shear deformation):

$$
\begin{aligned}
d &= \frac{qL^4}{8EI} + \frac{q\frac{L^2}{2}}{GA_v} \\
&= \frac{qL^4}{8E\frac{bh^3}{12}} + \frac{q\frac{L^2}{2}}{\frac{E}{2(1+\nu)}\frac{bh}{\kappa}} \\
&= \frac{400N/m \times 12^4 m^4}{8 \times 10^8 N/m^2 \times \frac{2^4}{12} m^4} + \frac{400N/m \times \frac{12^2}{2} m^2}{\frac{10^8}{2} N/m^2 \times 2m \times 2m \times \frac{5}{6}} \\
&= 7.776 \times 10^{-3} m + 1.728 \times 10^{-4} m \\
&= 7.9488 \times 10^{-3} m
\end{aligned}
\qquad (307.47)
$$

The Real-ESSI displacement results are listed in Table (307.33).

Table 307.33: Results for 27NodeBrick cantilever beams of irregular shapes with more elements.

Element Type	Shape	Force direction	Number of division		
			1	2	4
27NodeBrick	shape1	Vertical (z)	7.913E-03 m	7.946E-03 m	7.948E-03 m
27NodeBrick	shape1	Transverse (y)	7.903E-03 m	7.946E-03 m	7.948E-03 m
27NodeBrick	shape2	Vertical (z)	7.741E-03 m	7.930E-03 m	7.947E-03 m
27NodeBrick	shape2	Transverse (y)	7.371E-03 m	7.894E-03 m	7.944E-03 m
27NodeBrick	shape3	Vertical (z)	1.982E-03 m	7.946E-03 m	7.948E-03 m
27NodeBrick	shape3	Transverse (y)	1.979E-03 m	7.947E-03 m	7.948E-03 m
Theoretical solution			7.9488E-03 m	7.9488E-03 m	7.9488E-03 m

The error are listed in Table (307.34).

The errors are shown in Figures (307.113), (307.116) and (307.119).

The Real-ESSI model fei/DSL files for the table above are HERE.

Table 307.34: Errors for 27NodeBrick cantilever beams of irregular shapes with more elements.

Element Type	Shape	Force direction	Number of division		
			1	2	4
27NodeBrick	shape1	Vertical (z)	0.45%	0.04%	0.01%
27NodeBrick	shape1	Transverse (y)	0.32%	0.03%	0.01%
27NodeBrick	shape2	Vertical (z)	2.61%	0.23%	0.03%
27NodeBrick	shape2	Transverse (y)	7.27%	0.69%	0.06%
27NodeBrick	shape3	Vertical (z)	75.06%	0.04%	0.01%
27NodeBrick	shape3	Transverse (y)	75.11%	0.03%	0.01%

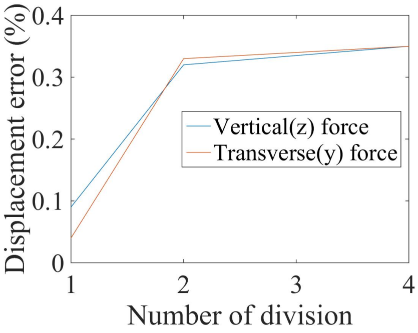


Figure 307.111: Error scale 0% - 0.4%.

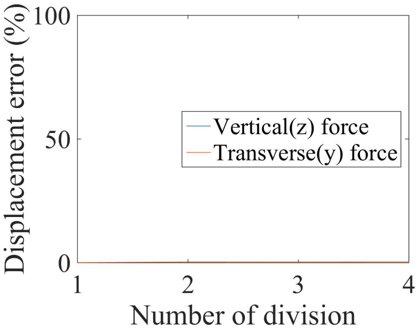


Figure 307.112: Error scale 0% - 100%.

Figure 307.113: 27NodeBrick cantilever beam for irregular Shape 1˙ Displacement error versus Number of division

$\nu = 0.0$. Use the shear deformation coefficient $\kappa = 1.2$. The force direction was shown in Figure (??).

In this section, the beam was cut into smaller elements with element side length 0.5m and 0.25m respectively. And the element side length of the original models is 1.0m. The numerical models are shown in Figure (307.120), (307.121) and (307.122).

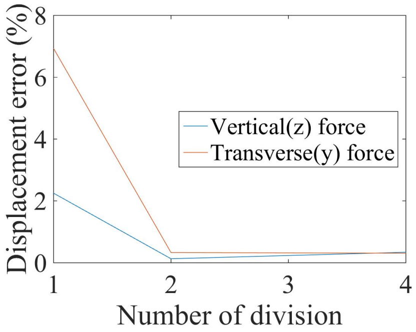


Figure 307.114: Error scale 0% - 8%.

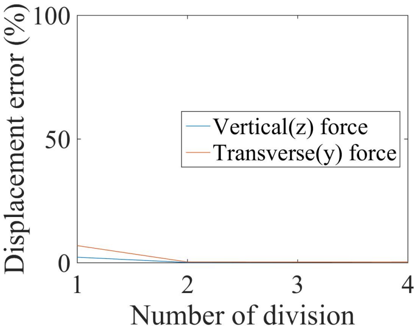


Figure 307.115: Error scale 0% - 100%.

Figure 307.116: 27NodeBrick cantilever beam for irregular Shape 2˙ Displacement error versus Number of division

Number of division 1:

Number of division 2:

Number of division 4:

The Real-ESSI results are listed in Table (307.35). The theoretical solution is 1.60E-5 m.

Table 307.35: Results for 27NodeBrick clamped beams with more elements.

Element Type	Element side length		
	1 m	0.5 m	0.25 m
27NodeBrick	1.64E-05 m	1.70E-05 m	1.71E-05 m
Error	0.83%	3.25%	3.70%

The errors are plotted in Figure (307.125).

The Real-ESSI model fei/DSL files for the table above are HERE.

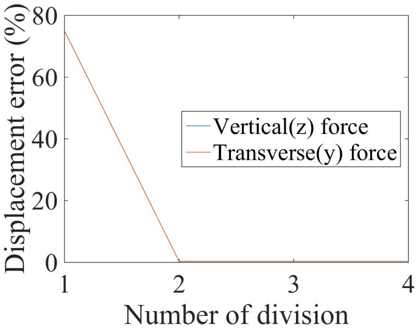


Figure 307.117: Error scale 0% - 80%.

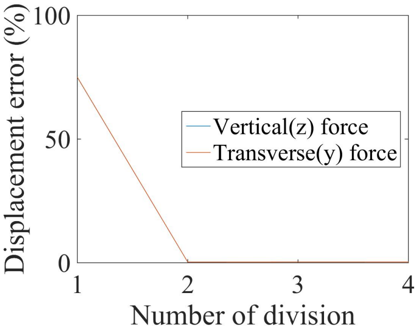


Figure 307.118: Error scale 0% - 100%.

Figure 307.119: 27NodeBrick cantilever beam for irregular Shape 3˙ Displacement error versus Number of division

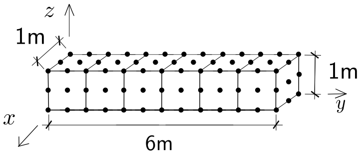


Figure 307.120: 27NodeBrick clamped beams with element side length 1.0m.

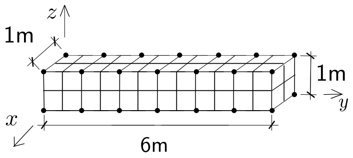


Figure 307.121: 27NodeBrick clamped beams with element side length 0.5m.

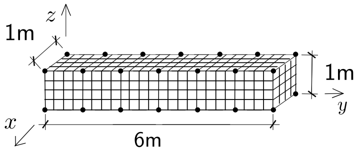


Figure 307.122: 27NodeBrick clamped beams with element side length 0.25m.

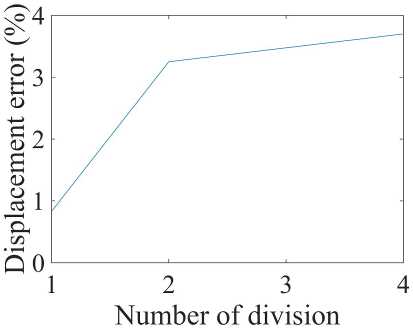


Figure 307.123: Error scale 0% - 4%.

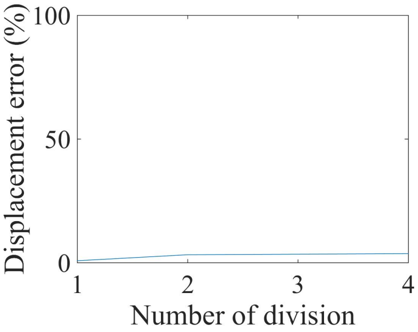


Figure 307.124: Error scale 0% - 100%.

Figure 307.125: 27NodeBrick clamped beam for different element number˙ Displacement error versus Number of division

307.2.17 Verification of 27NodeBrick stress in cantilever beams

Problem description: Length=6m, Width=1m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.0$. Use the shear deformation coefficient $\kappa = 1.2$. The force direction was shown in Figure (307.126).

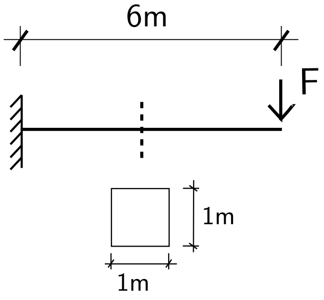


Figure 307.126: Problem description for cantilever beams of stress verification.

The theoretical solution for the stress was calculated below.

The 27NodeBrick elements are shown in Figure (307.127).

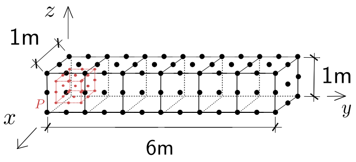


Figure 307.127: 27NodeBrick for cantilever beams of stress verification.

The bending moment at the Gassian Point is

$$M = F(L - P_y) = 100N \times (6 - 0.1127)m = 588.73N \cdot m \tag{307.48}$$

The bending modulus is

$$I = \frac{bh^3}{12} = \frac{1}{12}m^4 \tag{307.49}$$

Therefore, the theoretical stress is

$$\sigma = \frac{M \cdot z}{I} = \frac{588.73N \cdot m \times (0.5 - 0.1127)m}{\frac{1}{12}m^4} = 2736Pa \tag{307.50}$$

To get a better result, the same geometry beam was also cut into small elements. When more elements are used, the theoretical stress was calculated again with the new coordinates. The calculation process is similar to the process above.

The numerical models are shown in Figure (307.128), (307.129) and (307.130).

Number of division 1:

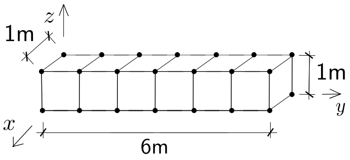


Figure 307.128: 27NodeBrick stress with element side length 1.0m.

Number of division 2:

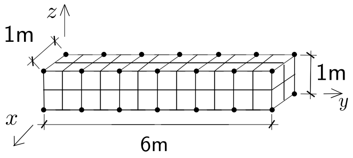


Figure 307.129: 27NodeBrick stress with element side length 0.5m.

Number of division 4:

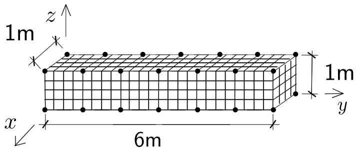


Figure 307.130: 27NodeBrick stress with element side length 0.25m.

All the stress results are listed in Table (307.36).

Table 307.36: Results for 27NodeBrick stress with more elements.

Element Type	Element side length		
	1 m	0.5 m	0.25 m
27NodeBrick	2719.81 Pa	3198.19 Pa	3464.76 Pa
Theoretical	2736.17 Pa	3164.27 Pa	3381.18 Pa
Error	0.60%	1.07%	2.47%

The Real-ESSI model fei/DSL files for the table above are HERE.

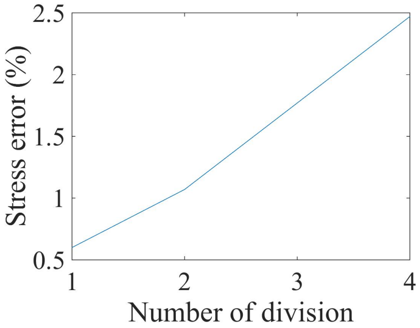


Figure 307.131: Error scale 0% - 2.5%.

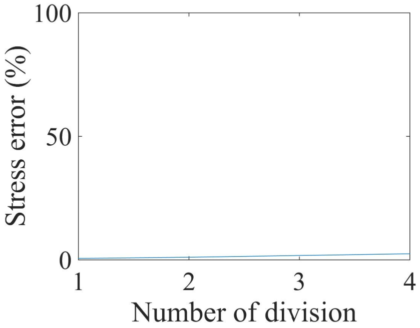


Figure 307.132: Error scale 0% - 100%.

Figure 307.133: 27NodeBrick cantilever beams for stress verification˙ Stress error versus Number of division

307.2.18 Verification of 27NodeBrick square plate with four edges clamped

Problem description: Length=20m, Width=20m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.3$.

The four edges are clamped.

The load is the uniform normal pressure on the whole plate.

The plate flexural rigidity is

$$D = \frac{Eh^3}{12(1-\nu^2)} = \frac{10^8 N/m^2 \times 1^3 m^3}{12 \times (1-0.3^2)} = 9.1575 \times 10^6 \; N \cdot m \tag{307.51}$$

The theoretical solution is

$$d = \alpha_c \frac{qa^4}{D} = 0.00406 \times \frac{100 N/m^2 \times 20^4 m^4}{9.1575 \times 10^6 \; N \cdot m} = 2.2015 \times 10^{-3} m \tag{307.52}$$

where α_c is a coefficient, which depends on the ratio of plate length to width. In this problem, the coefficient[8] α_c is 0.00406.

The 27NodeBrick are shown in Figure (307.134) - (307.139).

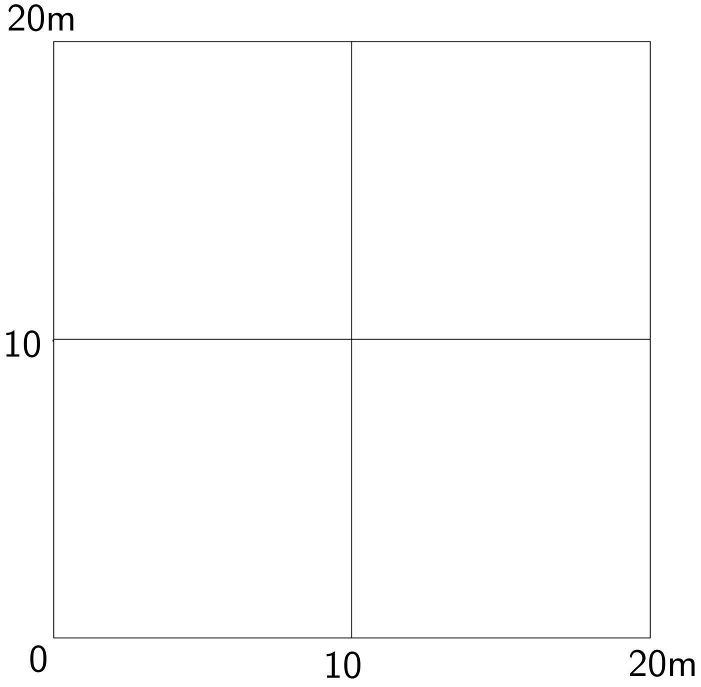


Figure 307.134: 27NodeBrick edge clamped square plate with element side length 10m.

The results were listed in Table (307.37).

---

[8] Stephen Timoshenko, Theory of plates and shells (2nd edition). MrGRAW-Hill Inc, page120, 1959.

[9] This model run out of memory on machine cml01 (memory: 23.5GB). This model has 233,289 nodes with 3 dofs, which may require 40GB memory.

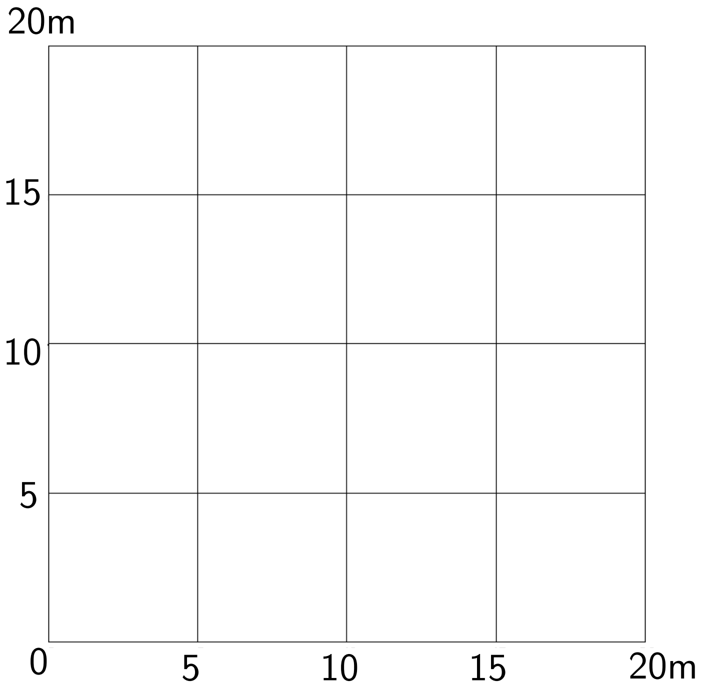


Figure 307.135: 27NodeBrick edge clamped square plate with element side length 5m.

Table 307.37: Results for 27NodeBrick square plate with four edges clamped.

Element type	27NodeBrick	27NodeBrick	27NodeBrick	Theoretical displacement
Number of layers	1layer	2layers	4layers	
Element side length	Height:1.00m	Height:0.50m	Height:0.25m	
10m	4.82E-004 m	4.82E-004 m	4.82E-004 m	2.20E-03 m
5m	1.97E-003 m	1.98E-003 m	1.98E-003 m	2.20E-03 m
2m	2.25E-003 m	2.26E-003 m	2.26E-003 m	2.20E-03 m
1m	2.28E-003 m	2.29E-003 m	2.29E-003 m	2.20E-03 m
0.5m	2.29E-003 m	2.30E-003 m	2.30E-003 m	2.20E-03 m
0.25m	2.29E-003 m	2.30E-003 m	-[9]	2.20E-03 m

The errors were listed in Table (307.38).

The errors were plotted in Figure (307.140).

The Real-ESSI model fei/DSL files for the table above are HERE.

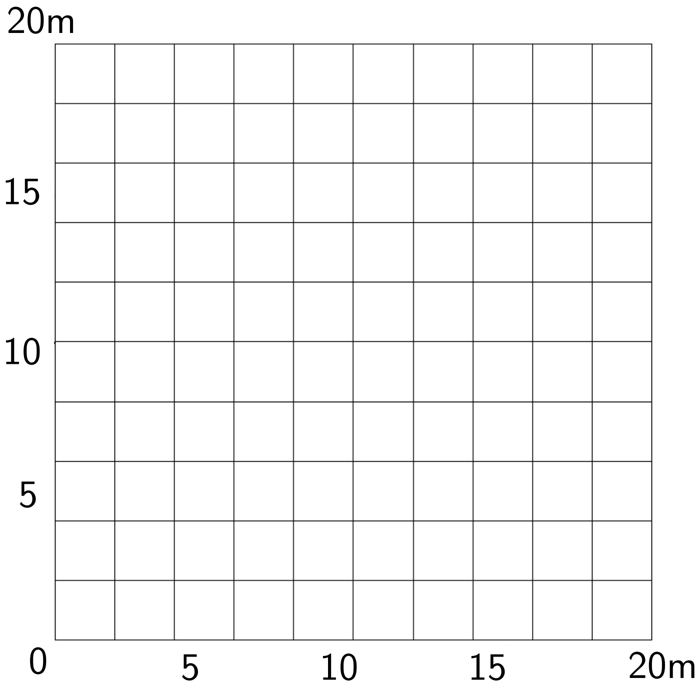


Figure 307.136: 27NodeBrick edge clamped square plate with element side length 2m.

Table 307.38: Errors for 27NodeBrick square plate with four edges clamped.

Element type	27NodeBrick	27NodeBrick	27NodeBrick
Number of layers	1layer	2layers	4layers
Element side length	Height:1.00m	Height:0.50m	Height:0.25m
10m	78.11%	78.10%	78.10%
5m	10.67%	10.19%	10.16%
2m	2.23%	2.79%	2.83%
1m	3.56%	4.16%	4.22%
0.5m	3.96%	4.58%	4.65%
0.25m	4.08%	4.70%	-

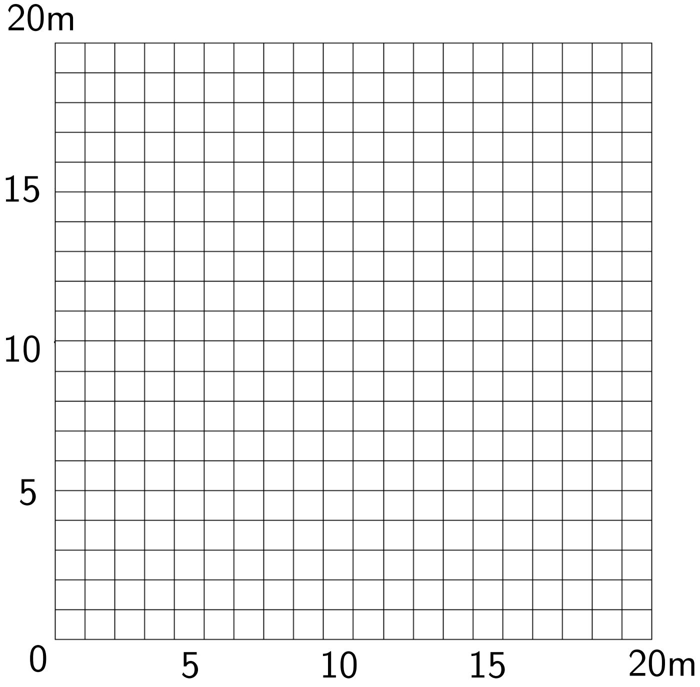


Figure 307.137: 27NodeBrick edge clamped square plate with element side length 1m.

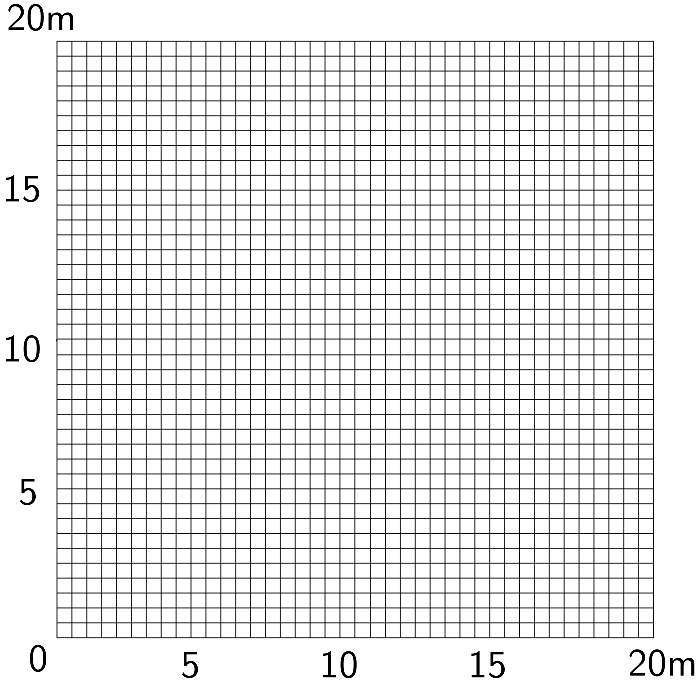


Figure 307.138: 27NodeBrick edge clamped square plate with element side length 0.5m.

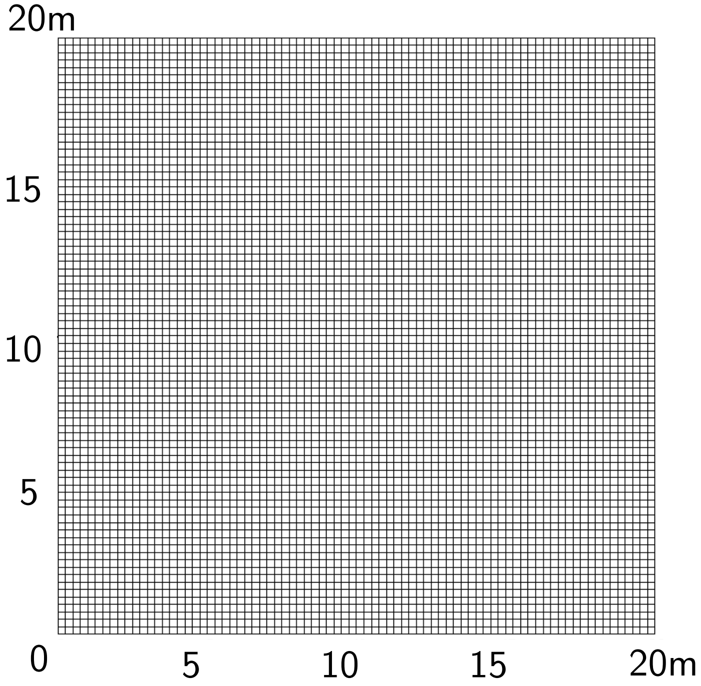


Figure 307.139: 27NodeBrick edge clamped square plate with element side length 0.25m.

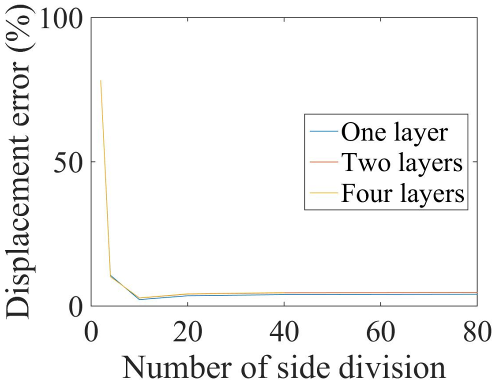


Figure 307.140: 27NodeBrick square plate with edge clamped˙ Displacement error versus Number of side division

307.2.19 Verification of 27NodeBrick square plate with four edges simply supported

Problem description: Length=20m, Width=20m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.3$.

The four edges are simply supported.

The load is the uniform normal pressure on the whole plate.

The plate flexural rigidity is

$$D = \frac{Eh^3}{12(1-\nu^2)} = \frac{10^8 N/m^2 \times 1^3 m^3}{12 \times (1-0.3^2)} = 9.1575 \times 10^6 \ N \cdot m \tag{307.53}$$

The theoretical solution is

$$d = \alpha_s \frac{qa^4}{D} = 0.00126 \times \frac{100 N/m^2 \times 20^4 m^4}{9.1575 \times 10^6 \ N \cdot m} = 7.0936 \times 10^{-3} m \tag{307.54}$$

where α_s is a coefficient, which depends on the ratio of plate length to width. In this problem, the coefficient[10] α_s is 0.00126.

The 27NodeBrick were shown in Figure (307.141) - (307.146).

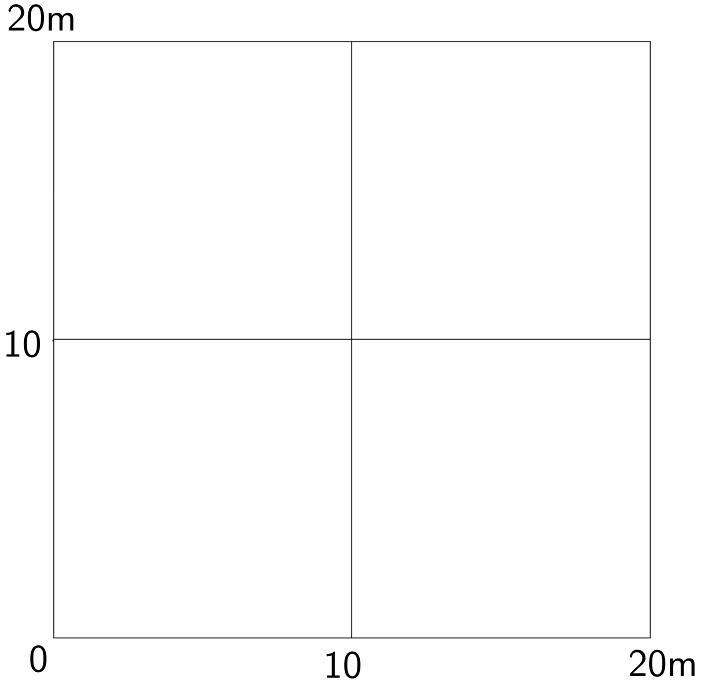


Figure 307.141: 27NodeBrick edge simply supported square plate with element side length 10m.

The results were listed in Table (307.39).

---

[10] Stephen Timoshenko, Theory of plates and shells (2nd edition). MrGRAW-Hill Inc, page202, 1959.

[11] This model run out of memory on machine cml01 (memory: 23.5GB). This model has 233,289 nodes with 3 dofs, which may require 40GB memory.

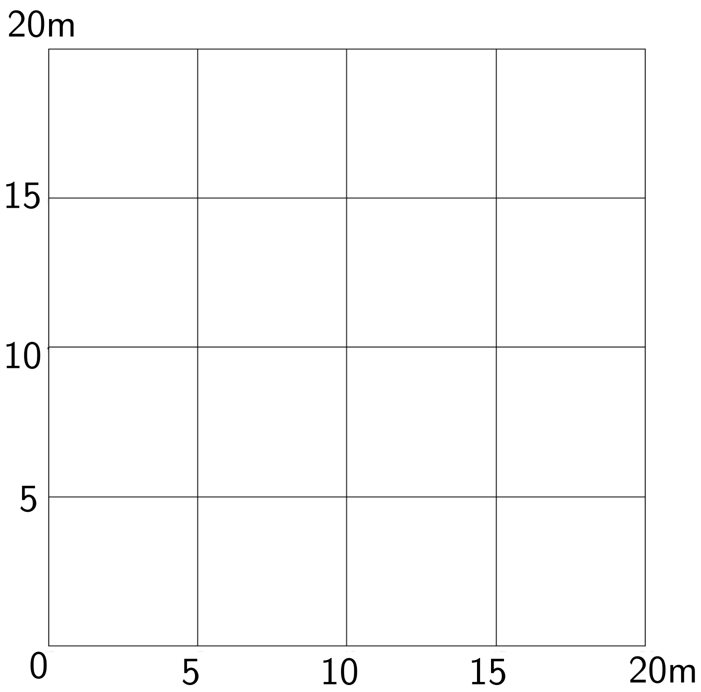


Figure 307.142: 27NodeBrick edge simply supported square plate with element side length 5m.

Table 307.39: Results for 27NodeBrick square plate with four edges simply supported.

Element type	27NodeBrick	27NodeBrick	Theoretical displacement
Number of layers	2layers	4layers	
Element side length	Height:0.50m	Height:0.25m	
10m	6.54E-003 m	6.54E-003 m	7.09E-03 m
5m	7.24E-003 m	7.24E-003 m	7.09E-03 m
2m	7.44E-003 m	7.44E-003 m	7.09E-03 m
1m	7.49E-003 m	7.49E-003 m	7.09E-03 m
0.5m	7.50E-003 m	7.50E-003 m	7.09E-03 m
0.25m	7.51E-003 m	-[11]	7.09E-03 m

The errors were listed in Table (307.40).

The errors were plotted in Figure (307.149).

The Real-ESSI model fei/DSL files for the table above are HERE.

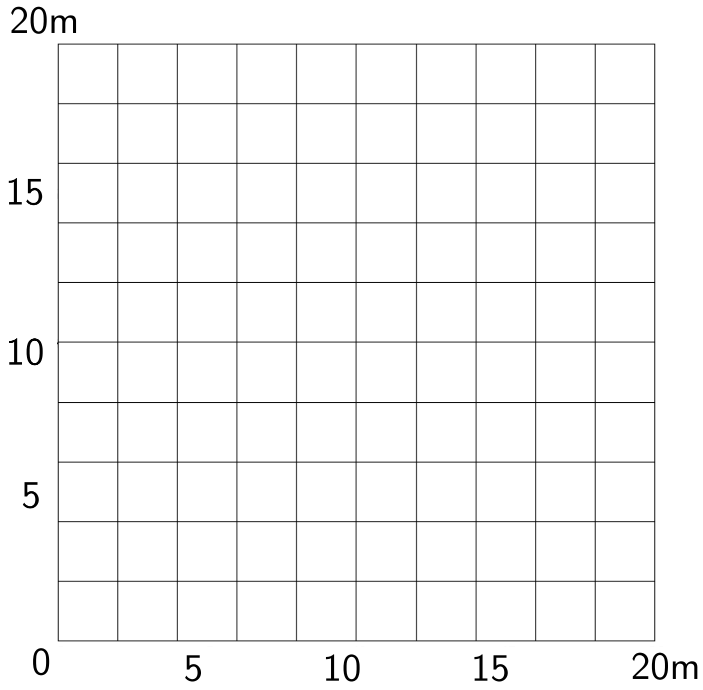


Figure 307.143: 27NodeBrick edge simply supported square plate with element side length 2m.

Table 307.40: Errors for 27NodeBrick square plate with four edges simply supported.

Element type	27NodeBrick	27NodeBrick
Number of layers	2layers	4layers
Element side length	Height:0.50m	Height:0.25m
10m	7.87%	7.85%
5m	2.07%	2.10%
2m	4.85%	4.89%
1m	5.54%	5.58%
0.5m	5.74%	5.79%
0.25m	5.80%	-

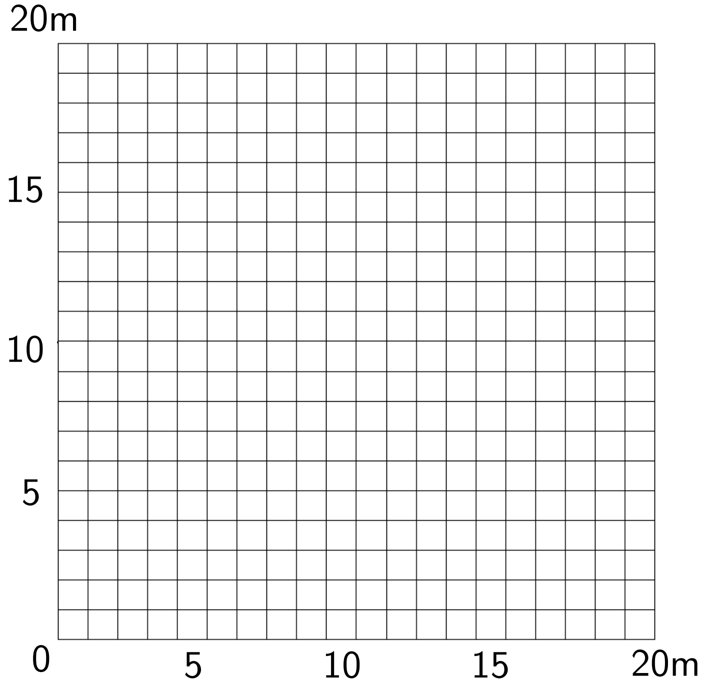


Figure 307.144: 27NodeBrick edge simply supported square plate with element side length 1m.

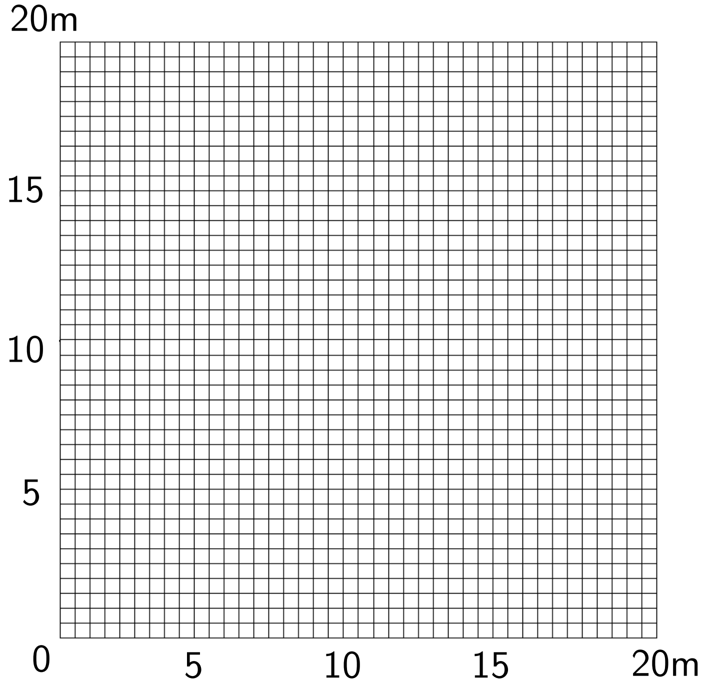


Figure 307.145: 27NodeBrick edge simply supported square plate with element side length 0.5m.

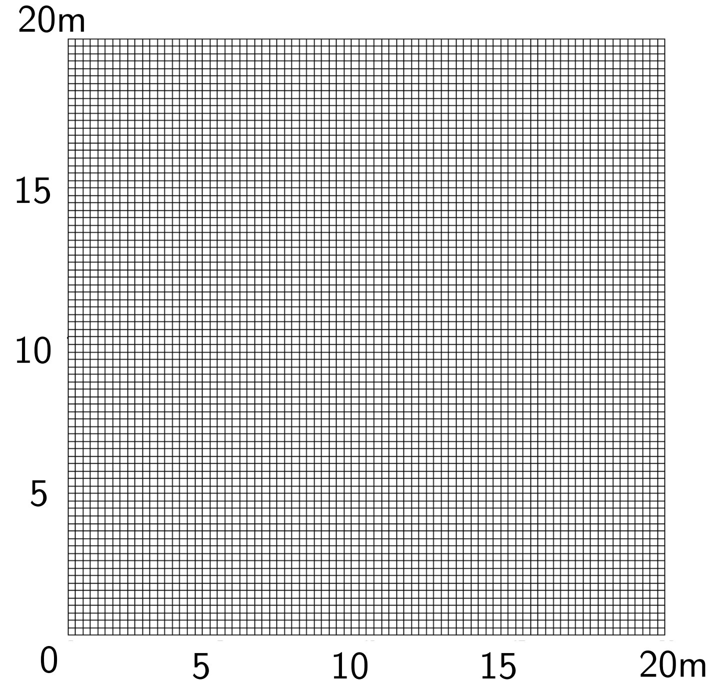


Figure 307.146: 27NodeBrick edge simply supported square plate with element side length 0.25m.

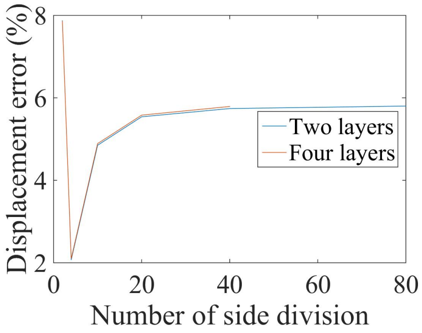


Figure 307.147: Error scale 0% - 8%.

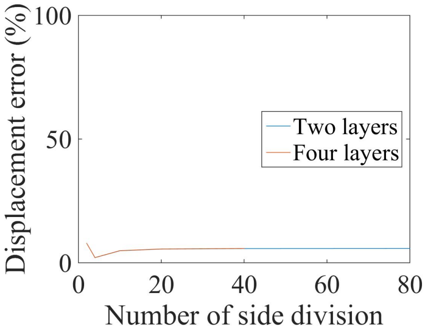


Figure 307.148: Error scale 0% - 100%.

Figure 307.149: 27NodeBrick square plate with edge simply supported˙ Displacement error versus Number of side division

307.2.20 Verification of 27NodeBrick circular plate with all edges clamped

Problem description: Diameter=20m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.3$.

The four edges are clamped.

The load is the uniform normal pressure on the whole plate.

The plate flexural rigidity is

$$D = \frac{Eh^3}{12(1-\nu^2)} = \frac{10^8 N/m^2 \times 1^3 m^3}{12 \times (1-0.3^2)} = 9.1575 \times 10^6 \ N \cdot m \tag{307.55}$$

The theoretical solution[12] is

$$d = \frac{qa^4}{64D} = \frac{100N/m^2 \times 10^4 m^4}{64 \times 9.1575 \times 10^6 \ N \cdot m} = 1.7106 \times 10^{-3} m \tag{307.56}$$

The 27NodeBrick were shown in Figure (307.150) - (307.155).

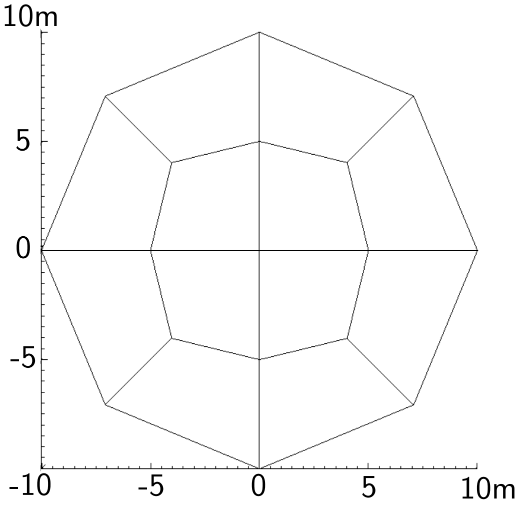


Figure 307.150: 27NodeBrick edge clamped circular plate with element side length 10m.

The results were listed in Table (307.41).

The errors were listed in Table (307.42).

The errors were shown in Figure (307.156).

The Real-ESSI model fei/DSL files for the table above are HERE.

---

[12] Stephen Timoshenko, Theory of plates and shells (2nd edition). MrGRAW-Hill Inc, page55, 1959.

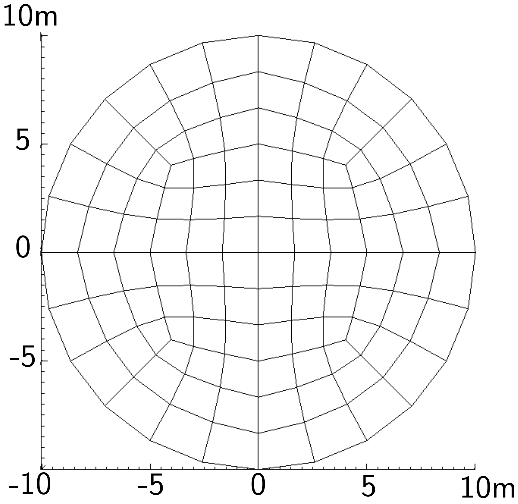


Figure 307.151: 27NodeBrick edge clamped circular plate with element side length 5m.

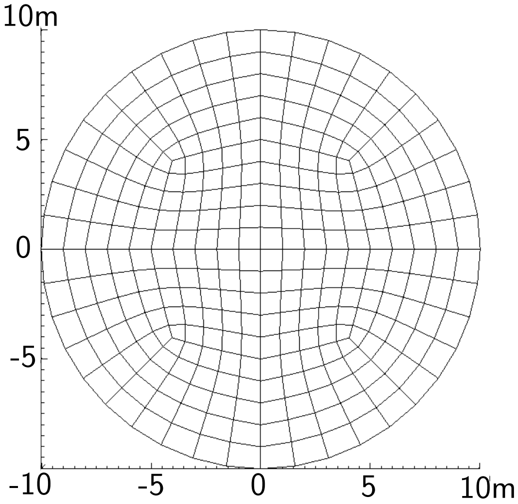


Figure 307.152: 27NodeBrick edge clamped circular plate with element side length 2m.

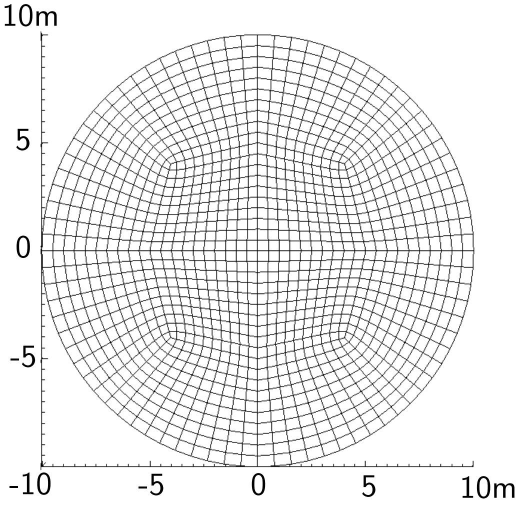


Figure 307.153: 27NodeBrick edge clamped circular plate with element side length 1m.

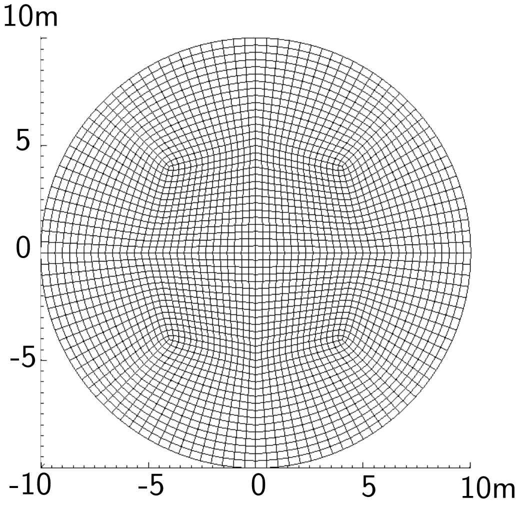


Figure 307.154: 27NodeBrick edge clamped circular plate with element side length 0.5m.

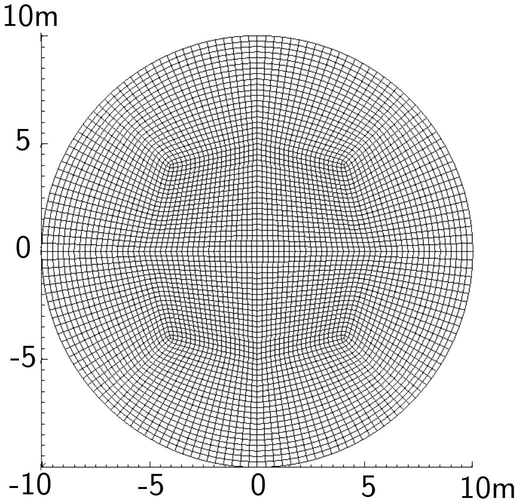


Figure 307.155: 27NodeBrick edge clamped circular plate with element side length 0.25m.

Table 307.41: Results for 27NodeBrick circular plate with four edges clamped.

Element type	27NodeBrick	27NodeBrick	27NodeBrick	Theoretical displacement
Number of layers	1layer	2layers	4layers	
Number of diameter divisions	Height:1.00m	Height:0.50m	Height:0.25m	
4	2.777E-03 m	2.788E-03 m	2.789E-03 m	1.706E-03 m
12	2.772E-03 m	2.786E-03 m	2.787E-03 m	1.706E-03 m
20	2.545E-03 m	2.556E-03 m	2.558E-03 m	1.706E-03 m
40	1.758E-03 m	1.768E-03 m	1.769E-03 m	1.706E-03 m
60	1.762E-03 m	1.772E-03 m	1.773E-03 m	1.706E-03 m
80	1.763E-03 m	1.773E-03 m	1.774E-03 m	1.706E-03 m

Table 307.42: Errors for 27NodeBrick circular plate with four edges clamped.

Element type	27NodeBrick	27NodeBrick	27NodeBrick
Number of layers	1layer	2layers	4layers
Number of diameter divisions	Height:1.00m	Height:0.50m	Height:0.25m
4	62.75%	63.42%	63.47%
12	62.46%	63.27%	63.34%
20	49.14%	49.82%	49.91%
40	3.03%	3.62%	3.68%
60	3.25%	3.83%	3.91%
80	3.32%	3.91%	3.99%

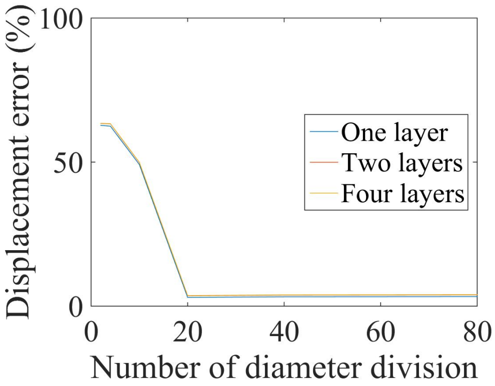


Figure 307.156: 27NodeBrick circular plate with edge clamped˙ Displacement error versus Number of side division

307.2.21 Verification of 27NodeBrick circular plate with all edges simply supported

Problem description: Diameter=20m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.3$.

The four edges are simply supported.

The load is the uniform normal pressure on the whole plate.

The plate flexural rigidity is

$$D = \frac{Eh^3}{12(1-\nu^2)} = \frac{10^8 N/m^2 \times 1^3 m^3}{12 \times (1-0.3^2)} = 9.1575 \times 10^6 \ N \cdot m \tag{307.57}$$

The theoretical solution[13] is

$$d = \frac{(5+\nu)qa^4}{64(1+\nu)D} = \frac{(5+0.3) \times 100N/m^2 \times 10^4 m^4}{64 \times (1+0.3) \times 9.1575 \times 10^6 \ N \cdot m} = 6.956 \times 10^{-3} m \tag{307.58}$$

The 27NodeBrick were shown in Figure (307.157) - (307.162).

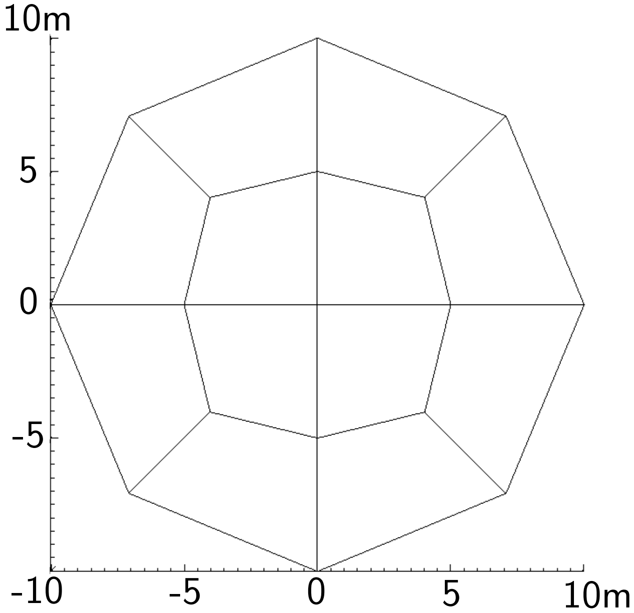


Figure 307.157: 27NodeBrick edge simply supported circular plate with element side length 10m.

The results were listed in Table (307.43).

The errors were listed in Table (307.44).

The errors were plotted in Figure (307.165).

The Real-ESSI model fei/DSL files for the table above are HERE.

---

[13] Stephen Timoshenko, Theory of plates and shells (2nd edition). MrGRAW-Hill Inc, page55, 1959.

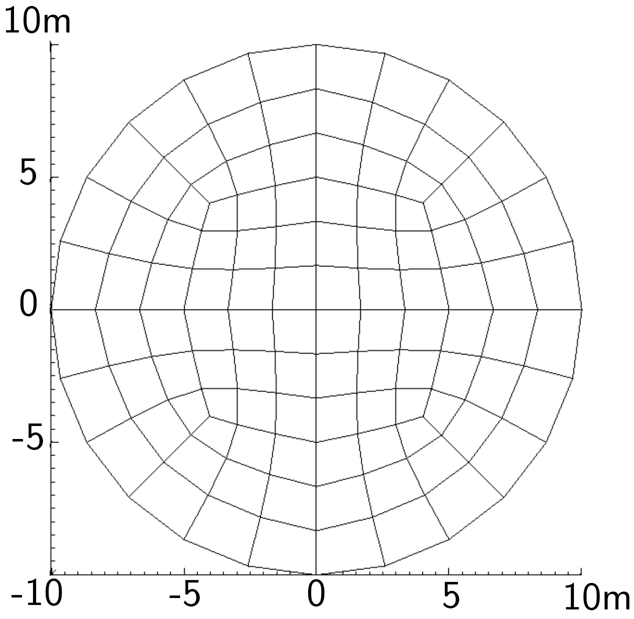


Figure 307.158: 27NodeBrick edge simply supported circular plate with element side length 5m.

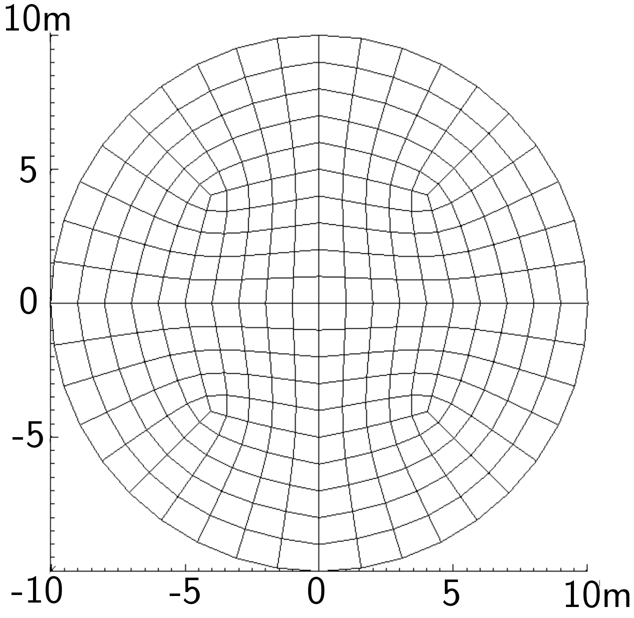


Figure 307.159: 27NodeBrick edge simply supported circular plate with element side length 2m.

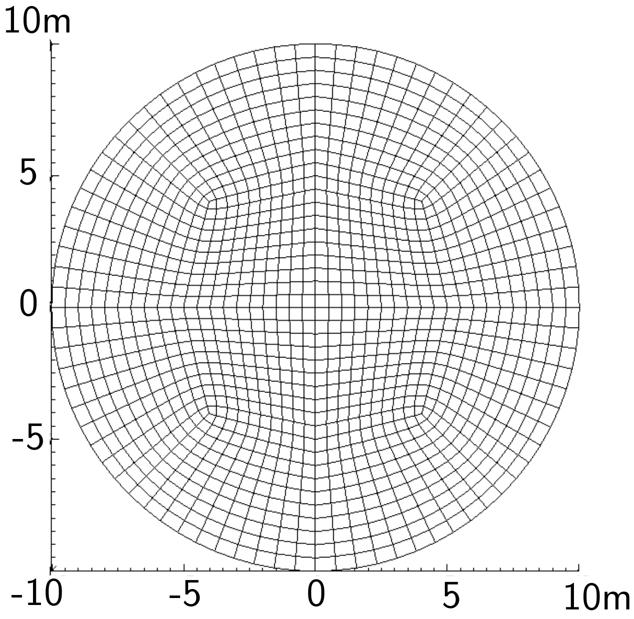


Figure 307.160: 27NodeBrick edge simply supported circular plate with element side length 1m.

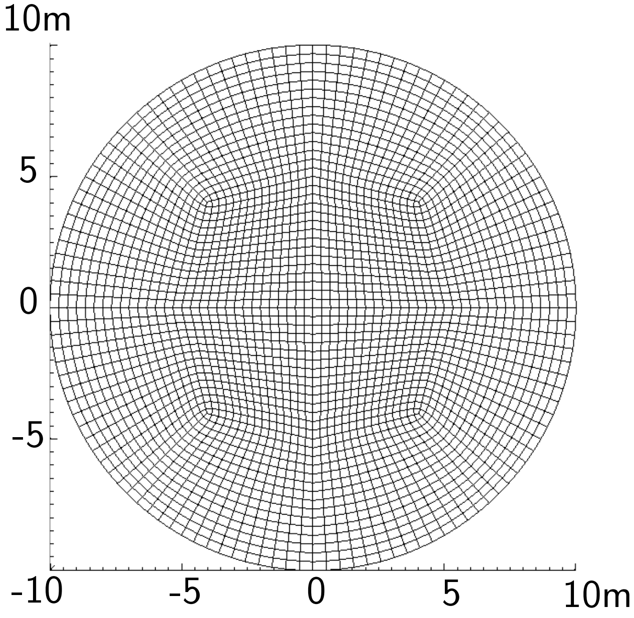


Figure 307.161: 27NodeBrick edge simply supported circular plate with element side length 0.5m.

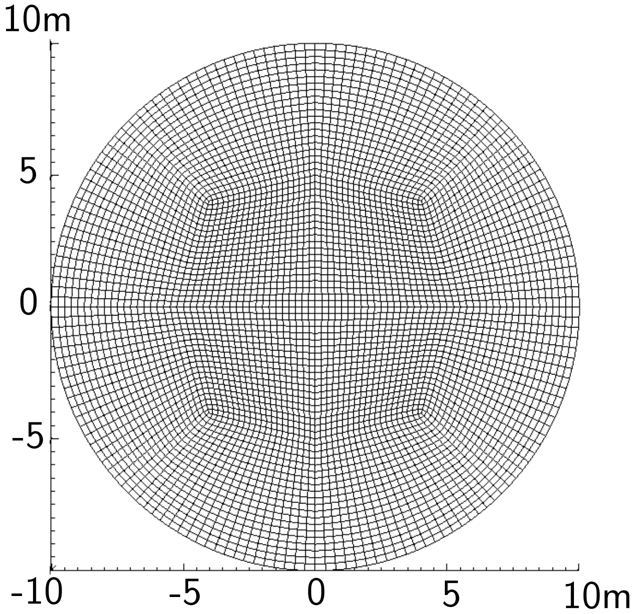


Figure 307.162: 27NodeBrick edge simply supported circular plate with element side length 0.25m.

Table 307.43: Results for 27NodeBrick cicular plate with four edges simply supported.

Element type	27NodeBrick	27NodeBrick	Theoretical displacement
Number of layers	2layers	4layers	
Number of diameter divisions	Height:0.50m	Height:0.25m	
4	7.259E-03 m	7.261E-03 m	6.956E-03 m
12	7.083E-03 m	7.084E-03 m	6.956E-03 m
20	7.064E-03 m	7.065E-03 m	6.956E-03 m
40	7.018E-03 m	7.019E-03 m	6.956E-03 m
60	7.029E-03 m	7.030E-03 m	6.956E-03 m
80	7.032E-03 m	7.034E-03 m	6.956E-03 m

Table 307.44: Errors for 27NodeBrick cicular plate with four edges simply supported.

Element type	27NodeBrick	27NodeBrick
Number of layers	2layers	4layers
Number of diameter divisions	Height:0.50m	Height:0.25m
4	4.36%	4.38%
12	1.82%	1.83%
20	1.56%	1.57%
40	0.88%	0.90%
60	1.04%	1.06%
80	1.09%	1.11%

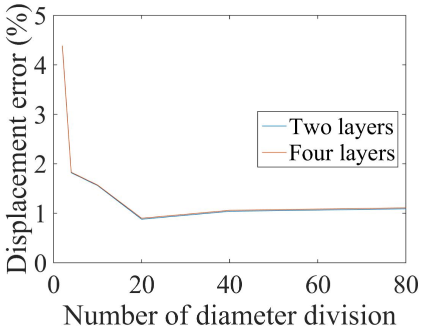


Figure 307.163: Error scale 0% - 5%.

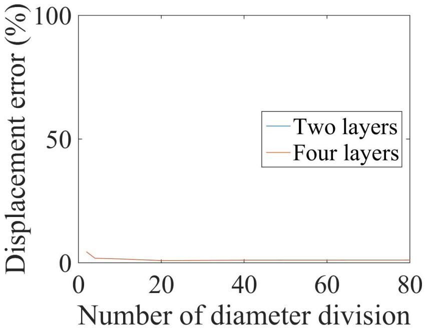


Figure 307.164: Error scale 0% - 100%.

Figure 307.165: 27NodeBrick circular plate with edge simply supported Displacement error versus Number of side division.

307.2.22 Verification of 27NodeBrick Finite Element for Boussinesq Problem

307.2.22.1 Introduction

The Boussinesq problem is finding the displacement distribution in the isotropic linearly elastic half-space, subject to a concentrated load applied on the surface and perpendicular to it. The Boussinesq problem diagram is shown in Fig.(307.166).

Boussinesq problem is widely used in geotechnical engineering, especially when designing a foundation which transfers the superstructure load to the soil. To estimate the foundation settlements, it is important to have a reliable numerical solution for the Boussinesq problem.

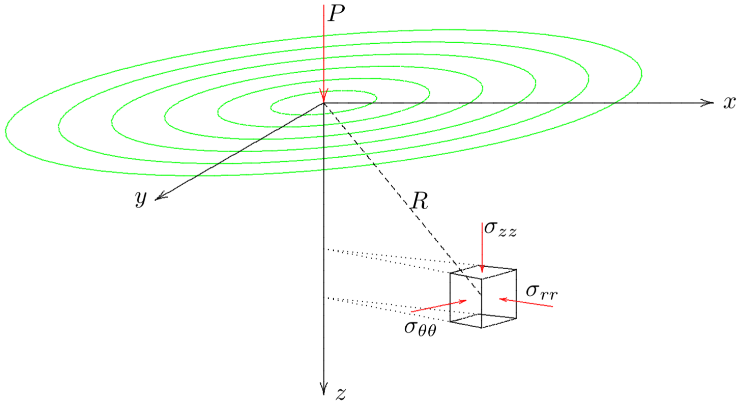


Figure 307.166: Boussinesq problem description (Figure Reference: Verruijt, Arnold, and Stefan Van Baars. Soil mechanics. Delft, 2007.)

In 1885, the French scientist Joseph Boussinesq solved the analytic solutions of displacements in the homogeneous isotropic linear elastic half space. In general, the vertical displacement of the surface is

$$z = 0: \quad u_z = \frac{P(1-\nu^2)}{\pi E R} \tag{307.59}$$

where P is the vertical load, ν is the Poisson's ratio, E is the elastic modulus, and R is the distance from the measured point to the loading point.

In this section, the Real-ESSI numerical solution is verified by the analytic solution for the Boussinesq problem.

307.2.22.2 Description of the Verification Model

Since the problem is cylindrical symmetry, a quarter of the entire cube was employed to represent the whole cube. The reduced model was shown in Fig.(307.167).

The side length is 20 meters and the load P is 1N. The elastic modulus $E = 1 \times 10^3 Pa$ and the Poisson's ratio $\nu = 0.0$.

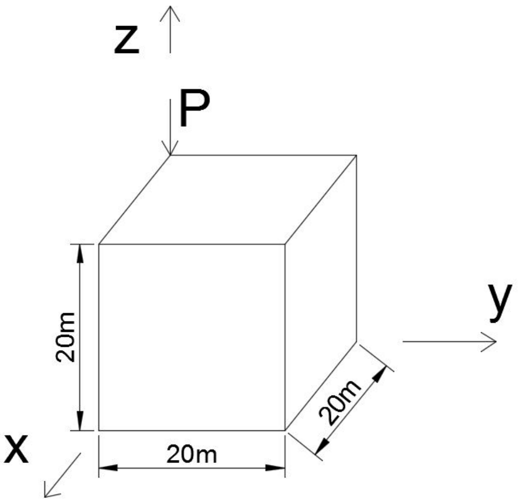


Figure 307.167: Reduced model (One quarter model) for the point load on the half space

The boundary conditions are shown in Fig.(307.168) and (307.169).

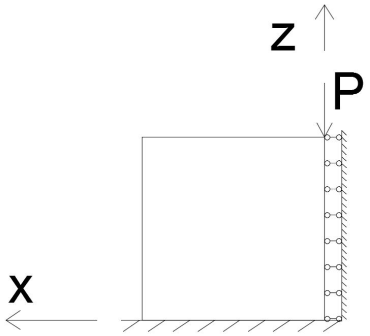


Figure 307.168: X-Z view for the reduced model

307.2.22.3 Results

Analytic solution for this model

According to the previous introduction, the analytic solution on the surface for this problem is

$$z = 0 : \quad u_z = \frac{P(1-\nu^2)}{\pi E R} = \frac{1}{10^3 \pi} \frac{1}{R} \tag{307.60}$$

On the face $x = 0$, the distance R on the surface is actually the value of y, therefore, the analytic

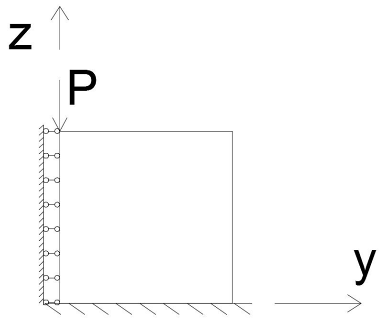


Figure 307.169: Y-Z view for the reduced model

solution is

$$u_z = \frac{1}{10^3 \pi R} = \frac{1}{10^3 \pi} \frac{1}{y} \tag{307.61}$$

As long as the y values are substituted, the displacement u_z is obtained immediately.

Real-ESSI solution with 27NodeBrickLT

In Real-ESSI, 27NodeBrickLT elements were used to simulate this model. Each element is $2m \times 2m \times 2m$. Since the model is $20m \times 20m \times 20m$, the element number is $10 \times 10 \times 10 = 1000$. The vertical displacement at the surface was recorded.

Since the model is symmetric, when the results were plotted, the other half results were achieved by symmetry.

Comparison between the analytic and 27NodeBrickLT solution

The Real-ESSI and analytic results were plotted in Fig.(307.170). Note that the analytic solution for location $y = 0$ is infinity, which was not plotted in the figure below.

307.2.22.4 Error Analysis

1. Mismatch at the loading point.

   First of all, when $x = 0$ at the loading point, the analytic solution is infinite. From the perspective of practical engineering, this analytic solution is flawed because the displacement cannot be infinite. The infinite solution is due to the elastic assumption. In consideration of the plasticity, the analytic solution will not be infinite.

   In Real-ESSI, the displacement at the loading point is not infinite because the infinite value is averaged by the integration during the finite element calculation. Also, at the loading point,

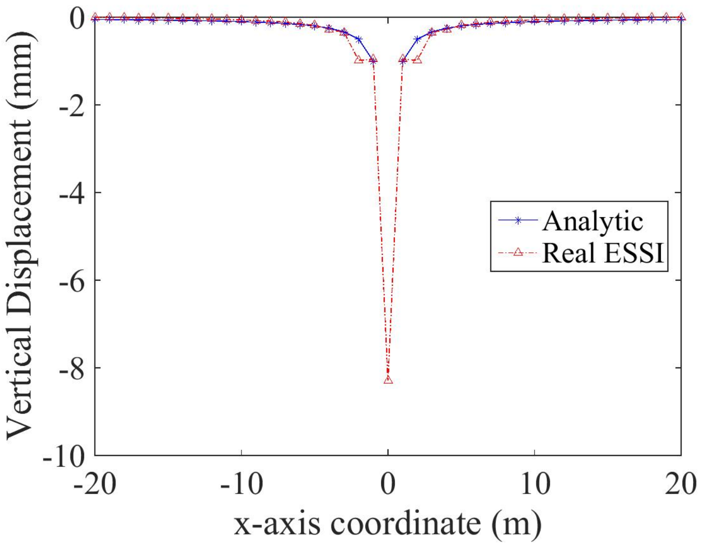


Figure 307.170: Comparison between analytic solution and Real-ESSI 27NodeBrickLT solution

27NodeBrickLT has a much larger displacement than that of 8NodeBrickLT. This is because 27NodeBrickLT has a relatively denser mesh than 8NodeBrickLT. So the maximum value at the loading point is higher than that of 8NodeBrickLT.

2. Mismatch at other locations.

   Except at the loading point, the analytic solution is not exactly equal to the numerical solution at other locations. This is because the verification example employs a simplified bounded cube to represent unbounded half space. The original analytic solution is for the half space, which is not true for the verification model. Not only the horizontal space but also the bottom space are removed from the model. This means the analytic solution is not perfect for this bounded cube. However, since the cube is very great, the analytic solution is similar to the Real-ESSI numerical solution. In addition, the brick elements are also verified by other models, like beam, plate and shells.

307.3 Verification of Dynamic, Single Phase Solid Modeling and Simulation

. . .

Chapter 308

Verification and Validation for Static and Dynamic Behavior of Structural Elements

(1986-2011-2015-2017-2019-2021-)

(In collaboration with Prof. José Abell, Dr. Yuan Feng, and Prof. Han Yang)

308.1 Chapter Summary and Highlights

308.2 Verification of Static, Beam-Column Finite Element Modeling and Simulation

308.3 Bernoulli Beam Elements with 12DOFs and 9DOFs

Figures 308.1 and 308.2

308.3.1 FEM Model

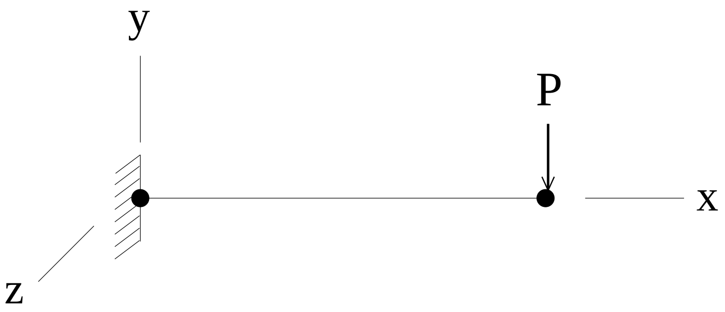


Figure 308.1: Finite element model for static analysis

Figure 308.2: Finite element model for dynamic analysis

308.3.2 Static Analysis

Figures 308.3

$$E = 5Pa$$
$$I = 16m^4$$
$$L = 2m$$
$$P = 1N$$
$$\Delta = \frac{PL^3}{3EI} = 0.033m$$

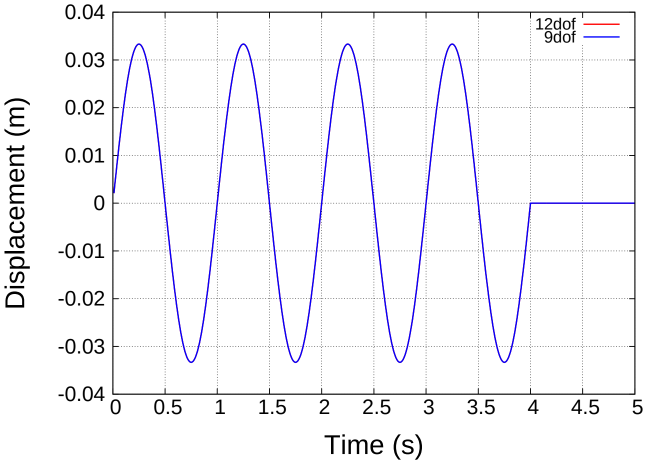


Figure 308.3: Comparison of static displacements (pseudo time) of the top nodes, Force time history applied to the top node

308.3.3 Dynamic Analysis

Figures 308.4 and 308.5

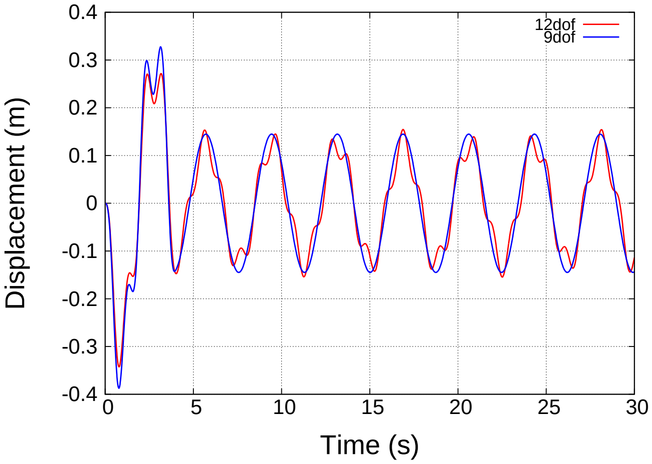


Figure 308.4: Comparison of displacement time histories of the top nodes, Displacement time history applied to the node with 6DOF, mass comes from beam density

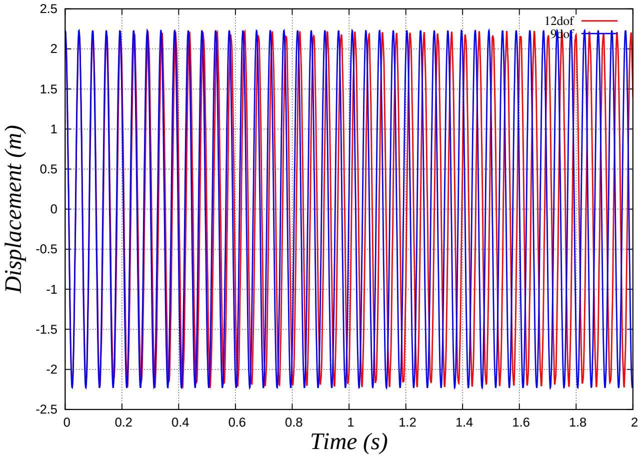


Figure 308.5: Comparison of free vibration displacement time histories of the top nodes, mass comes from beam density

308.3.4 Bernoulli Beam, Comparison of Eigen Frequencies

Table 308.1

Table 308.1: Comparison of eigen frequencies between models of using 9DOF beam and 12DOF beam

Mode	9DOF	12DOF
1	0.264559	0.264236
2	0.268474	0.268064
3	0.308202	0.308202

308.4 Timoshenko Beam

Models used:

- Model with 27-node-brick model is used as the benchmark. The model is $10\mathrm{m} \times 10\mathrm{m} \times 60\mathrm{m}$, each element is $2\mathrm{m} \times 2\mathrm{m} \times 2\mathrm{m}$, so there are $5 \times 5 \times 30 = 750$ elements in total.
- Model with 5 Timoshenko beams are used to test the performance of the Timoshenko element. Material properties and cross-section properties are kept the same as those of the brick model. Various values of the shear correction factor are tested.
- Model with 5 Bernoulli beam elements is also tested.
- For both beam models, consistent mass is used.

Input files for all the models shown are available HERE.

Figures of models are shown below:

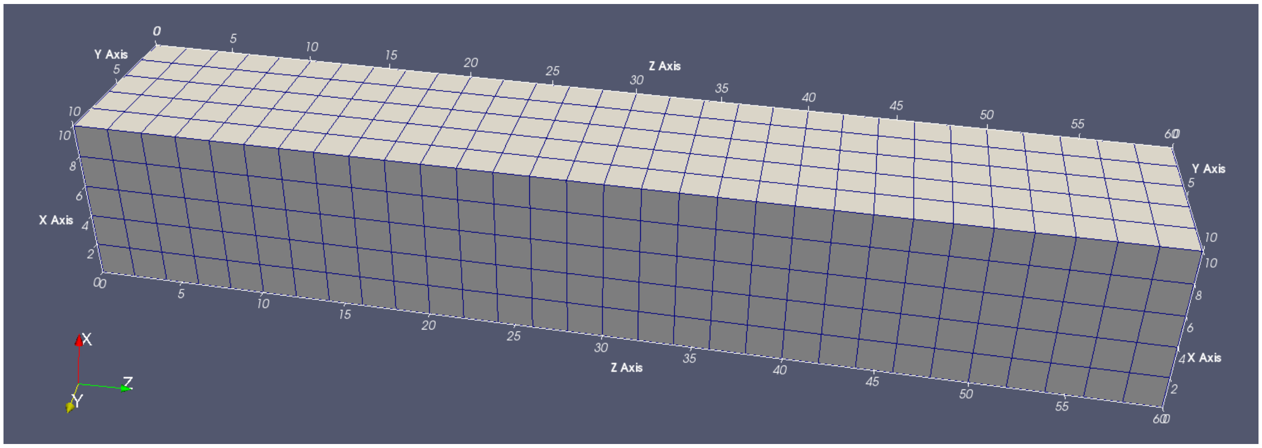


Figure 308.6: Cantilever model made of 27-node-brick elements.

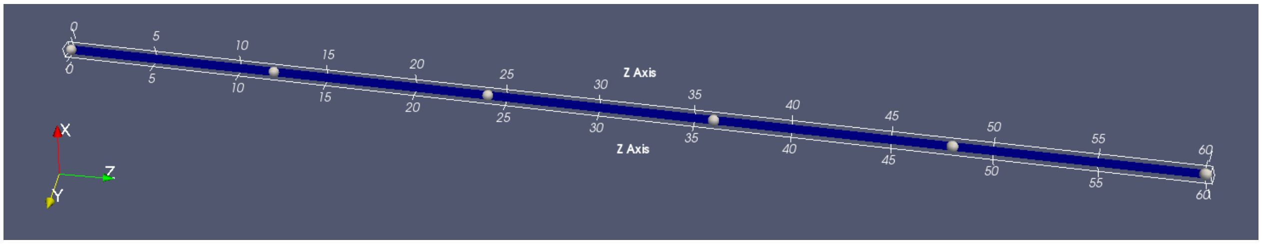


Figure 308.7: Cantilever model made of multiple Timoshenko or Bernoulli elements.

Eigenanalysis results are shown in the following table and figures.

Table 308.2: Comparison of eigenfrequencies for 27-node-brick, Timoshenko beam with different shear correction factors, and Bernoulli beam models.

Eigenmode		Eigenfrequency (Hz)				
		27-Node-Brick	Timoshenko (shear factor=1)	Timoshenko (shear factor=2)	Timoshenko (shear factor=100)	Bernoulli
Bending (1st)	1	1.63364	1.63375	1.64358	1.65336	1.65356
	2	1.63364	1.63375	1.64358	1.65336	1.65356
Bending (2nd)	3	9.23543	9.36137	9.6899	10.0443	10.0519
	4	9.23543	9.36137	9.6899	10.0443	10.0519
Torsion (1st)	5	9.30771	10.1473	10.1473	10.1473	10.1473
Axial (1st)	6	15.4575	15.5002	15.5002	15.5002	15.5002
Bending (3rd)	7	22.8703	23.7943	25.246	26.9177	26.9542
	8	22.8703	23.7943	25.246	26.9177	26.9542
Torsion (2nd)	9	27.9199	31.4476	31.4476	31.4476	31.4476

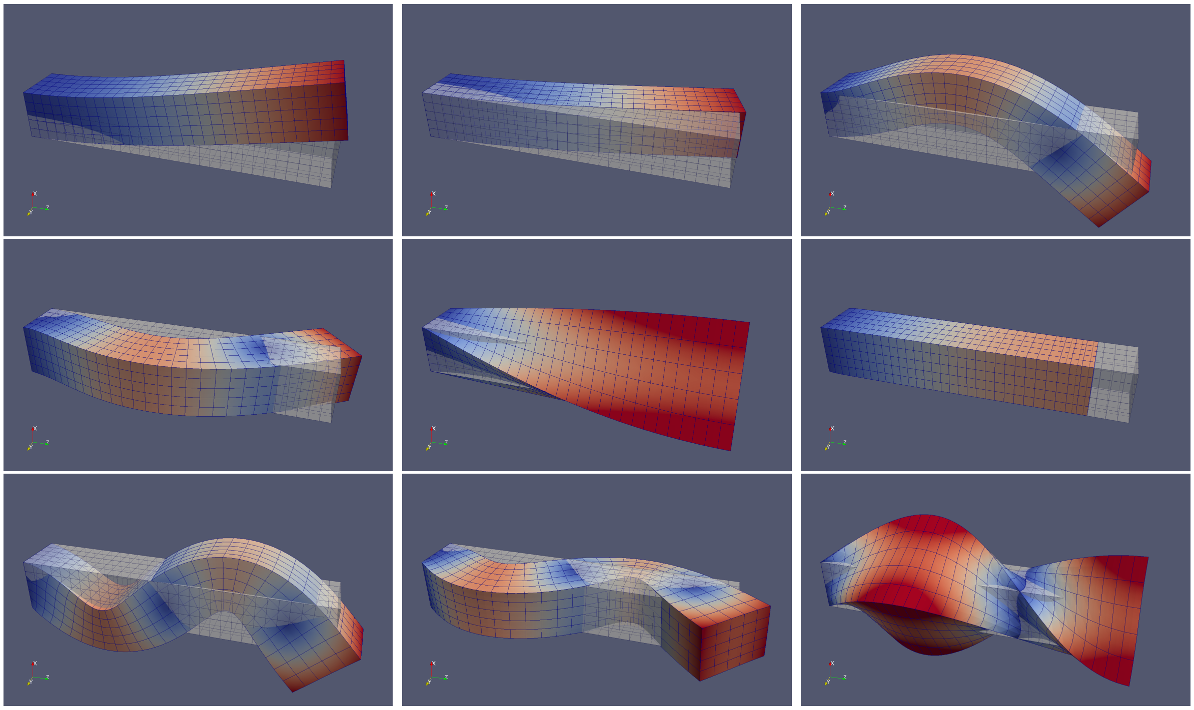


Figure 308.8: First 9 eigenmodes of the cantilever model made of 27-node-brick elements.

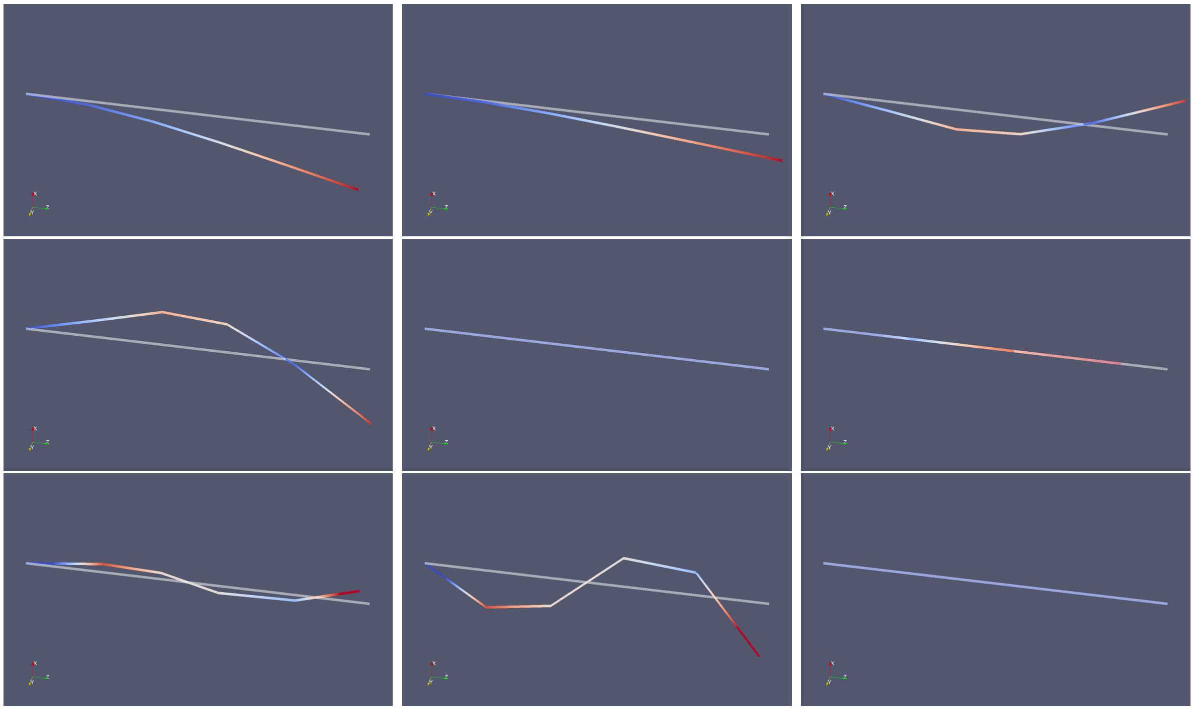


Figure 308.9: First 9 eigenmodes of the cantilever model made of Timoshenko beam elements.

308.5 Verification of Shell (Felippa-ANDES) Finite Element Modeling and Simulation

The verification and validation of the behavior of the ANDES (Assumed Natural Deviatoric Strain) shell finite element implemented in essi is described in this section. The verification is split up into several cases intended to test different aspects of the implementation. The tests are based on well-known closed form solutions to elasticity problems which can be modelled using shells. Further, the verification is divided into static and dynamic tests.

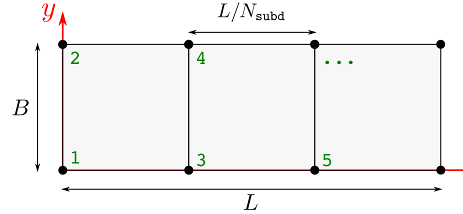
308.5.1 Static Tests

The purpose of the static tests is to verify that the stiffness matrices generated by the 4 Node ANDES Shell are useful to approximate well known cases of solutions to beam theory.

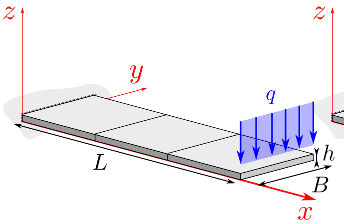
Tests are done to verify the bending component and the membrane component independently, because the behaviour of both is assumed de-coupled in this linear implementation.

308.5.1.1 Bending Component Verification

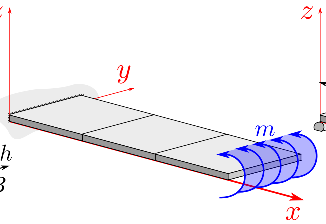
These tests compare the bending performance of a simple mesh of ANDES shells for the case of simple (Bernoulli) beam examples.



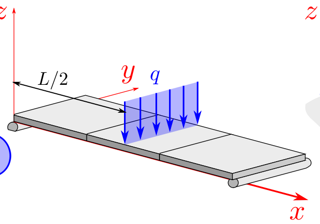
Mesh



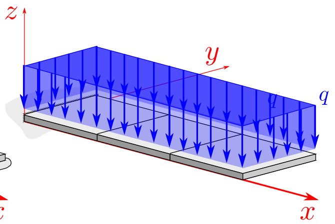
Test 1



Test 2



Test 3



Test 4

Test 1: Concentrated transversal tip load

The test file for this case can be found in

(ESSI SOURCE FOLDER)/Verification_Examples/...
.../Static_and_Dynamic_Behavior_of_Structural_Elements/...
.../Shell_ANDES/static/Test_shell_andes_1_bending_transverse.fei

For a prismatic beam made of a homogeneous elastic isotropic material of modulus of elasticity E, cross section I and length L we have that, under Bernoulli-Euler theory, the displacement at the tip due to a concentrated load P is

$$\delta_{\text{tip}} = \frac{PL^3}{3EI}$$

and for this test case's geometry

$$\delta_{\text{tip}} = \frac{qBL^3}{3EBh^3/12} = \frac{4qL^3}{Eh^3}$$

With the purpose of measuring the errors as percent agreement with the theoretical results, the load q is computed such that the tip displacement is 100 units. Thus, in terms of the geometry of the problem.

$$q = 100 \times \frac{Eh^3}{4L^3}$$

This force is translated into nodal equivalent forces by applying half the total load qB to each tip node in the vertical direction and fixing the x direction rotational degrees of freedom for the tip nodes.

The parameters chosen for this case are

```
h = 1*m; // Shell thickness
Eshell = 1*N/m^2; // Elastic modulus
nu = 0.0; // Poisson's ratio
L = 1*m; // Beam length
B = 0.2*m; // Beam width
rho = 0*kg/m^3; // Mass density
```

For different number of subdivisions, here are the results of the tip displacement.

N_{subd}	u_z	N_{subd}	u_z	N_{subd}	u_z
2	96.2118	42	100.009	83	100.002
7	100.096	48	100.007	89	100.002
13	100.068	54	100.006	95	100.002
19	100.039	60	100.005	101	100.002
25	100.024	66	100.004	107	100.001
31	100.016	72	100.003	112	100.001
37	100.012	77	100.003		

The element displays sensitivity to the aspect ratio of its sides. Indeed, chosing a different set of parameters

```
h = 1*m; // Shell thickness
Eshell = 1*N/m^2; // Elastic modulus
nu = 0.0; // Poisson's ratio
L = 1*m; // Beam length
B = 0.2*m; // Beam width
rho = 0*kg/m^3; // Mass density
```

yields slightly different results

N_{subd}	u_z	N_{subd}	u_z	N_{subd}	u_z
2	93.7897	42	99.9958	83	100.001
7	99.5267	48	99.998	89	100.001
13	99.8587	54	99.9993	95	100.001
19	99.9409	60	100.	101	100.001
25	99.9713	66	100.001	107	100.001
31	99.985	72	100.001	112	100.001
37	99.992	77	100.001		

Test 2: Concentrated tip moment

The test file for this case can be found in

`(ESSI SOURCE FOLDER)/Verification_Examples/...`

`.../Static_and_Dynamic_Behavior_of_Structural_Elements/...`

`.../Shell_ANDES/static/Test_shell_andes_2_bending_transverse.fei`

Analogous to the previous test, for a cantilever beam with a tip moment the tip rotation is

$$\theta_{\text{tip}} = \frac{ML}{EI}$$

and for this test case's geometry

$$\theta_{\text{tip}} = \frac{mBL}{EBh^3/12} = \frac{12mL}{Eh^3}$$

In this case, the parameters are selected such that this tip rotation is of 100 units. Execution of the test case shows that with $N_{\text{subd}} = 2$ the theoretical value is met with 100% accuracy. This is because the moment field is constant inside the beam and the elements are capable of reproducing this field with accuracy.

Test 3: Concentrated mid-span transversal load

This case is similar to test case 1. The accuracy for 2 elements is 96.2% agreement with the theoretical solution. For $N_{\texttt{subd}} = 4$ (4 elements total) the accuracy climbs to 99.7%. The test file for this case can be found in

`(ESSI SOURCE FOLDER)/Verification_Examples/...`

`.../Static_and_Dynamic_Behavior_of_Structural_Elements/...`

`.../Shell_ANDES/static/Test_shell_andes_3_bending_transverse.fei`

Test 4: Transversal distributed load

The distributed load is generated by accelerating the beam transversally with an acceleration which produces a 100 (unit) displacement. This provides an indirect test to the mass matrix which will be further tested in dynamic tests. The uniformly distributed load q which produces the δ_{tip} unit displacement and the corresponding tip rotation are

$$q = 8\frac{EI}{L^4}\delta_{\text{tip}} \qquad \text{and} \qquad \theta_{y\text{tip}} = \frac{4}{3L}\delta_{\text{tip}}$$

from this value, the required acceleration is computed as

$$a = \frac{q}{\rho BH}$$

where ρ is the unit-weight of the material used to compute the mass matrix.

For two subdivisions (6 nodes) the results of the two nodes located at the tip of the beam are

```
Node : 5
 ux = 0.000000, rx = 30.550500
 uy = 0.000000, ry = -133.333000
 uz = 102.541000, rz = 0.000000
Node : 6
 ux = 0.000000, rx = -30.550500
 uy = 0.000000, ry = -133.333000
 uz = 102.541000, rz = 0.000000
```

First, a 2.5% accuracy is reached in the tip displacement. Second, it is noteworthy to mention that the current formulation of the mass matrix will produce rotations around the x axis for the nodes. One reason for this is that the chosen mass matrix is not consistent with the stiffness matrix instead, it is borrowed from a similar element [1] [2] [3]. The reason behind this decision is the fact that the very accurate ANDES stiffness formulation lacks a displacement interpolation scheme which is necessary to produce

---

[1] The First ANDES Elements: 9-DOF Plate Bending Trianges Carmello Militello & Carlos A. Felippa December 1989 Report No. CU-CSSC-89-22

[2] Chapter 32 of Felippa's Lecture Notes Finite element templates for bending

[3] C. A. Felippa and P. G. Bergan, A triangular plate bending element based on an energy-orthogonal free formulation, Comp. Meth. Appl. Mech. Engrg., 61, 129{160, 1987.

a consistent mass matrix. Furthermore, restraining these x roation degrees of freedom does not lead to an improved solution.

For 4 subdivisions, the results at the tip are:

```
Node : 9
 ux = 0.000000, rx = 31.671000
 uy = 0.000000, ry = -133.333000
 uz = 100.891000, rz = 0.000000
Node : 10
 ux = 0.000000, rx = -31.671000
 uy = 0.000000, ry = -133.333000
 uz = 100.891000, rz = 0.000000
```

and for 20.

```
Node : 41
 ux = 0.000000, rx = 32.095800
 uy = 0.000000, ry = -133.333000
 uz = 100.039000, rz = 0.000000
Node : 42
 ux = 0.000000, rx = -32.095800
 uy = 0.000000, ry = -133.333000
 uz = 100.039000, rz = 0.000000
```

and back to 2 subdivisions but this time with an aspect ratio $L/B = 2$, by changing B so that the expected tip displacement remains the same, we get

```
Node : 5
 ux = 0.000000, rx = 3.890200
 uy = 0.000000, ry = -133.333000
 uz = 99.810100, rz = 0.000000
Node : 6
 ux = 0.000000, rx = -3.890200
 uy = 0.000000, ry = -133.333000
 uz = 99.810100, rz = 0.000000
```

which shows that the effect of the inconsistent mass matrix is ameliorated for elements with a better aspect ratio. The elements in this case are square in shape since $L/B = 2$ and $N_{\texttt{subd}} = 2$.

In conclusion increasing accuracy for bending problems not only involves making the elements smaller but also improving their aspect ratio. This is especially critical in dynamic problems (or self weight problems). The test file for this case can be found in

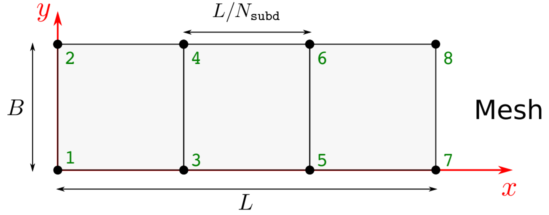
`(ESSI SOURCE FOLDER)/Verification_Examples/...`

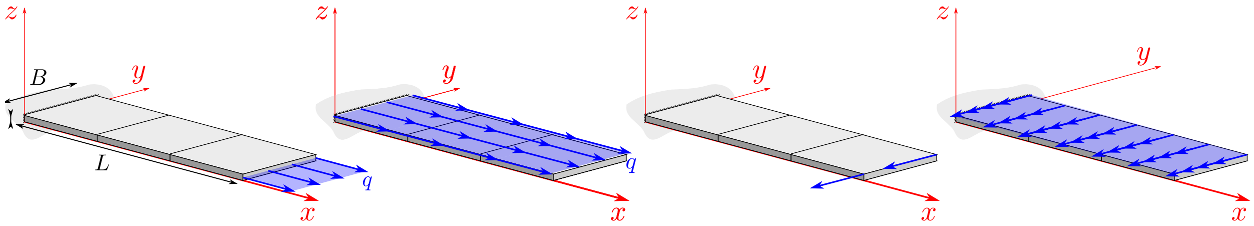
`.../Static_and_Dynamic_Behavior_of_Structural_Elements/...`

`.../Shell_ANDES/static/Test_shell_andes_4_bending_transverse.fei`

308.5.1.2 Membrane Component Verification

These tests compare the performance of the membrane component of the ANDES Shell by modelling simple beams and comparing the approximation with Bernoulli-beam theoretical results.





Test 5 Test 6 Test 7 Test 8

Test 5: Longitudinal tip load

The parameters used to test this case are,

```
h = 0.2*m; // Shell thickness
Eshell = 1*N/m^2; // Elastic modulus
nu = 0.0; // Poisson's ratio
L = 30*m; // Beam length
B = 3*m; // Beam width
rho = 0.0*kg/m^3; // Mass density
delta_tip = 100*m; // Target tip displacement

Nsubd = 2; // Number of side subdivisions
```

the tip load was computed to give $\delta_{\text{tip}} = 100$ from

$$F_{\text{tip}} = \frac{AE}{L}\delta_{\text{tip}}$$

Additionally, the tip rotation degrees of freedom about z axis (`rz`) were fixed to enforce the uniform loading condition. The results for 2 subdivisions observed at the two tip nodes is

```
Node : 5
 ux = 100.000000, rx = 0.000000
 uy = -0.000000, ry = 0.000000
 uz = 0.000000, rz = 0.000000
Node : 6
 ux = 100.000000, rx = 0.000000
 uy = -0.000000, ry = 0.000000
 uz = 0.000000, rz = 0.000000
```

The test file for this case can be found in `(ESSI SOURCE FOLDER)/Verification_Examples/...`
`.../Static_and_Dynamic_Behavior_of_Structural_Elements/...`
`.../Shell_ANDES/static/Test_shell_andes_5_membrane_axial.fei`

Test 6: Longitudinal self-weight load

As in test 4, the distributed load for this test was generated by using an acceleration field with a magnitude such that it generates 100 units of displacement in DOF `ux`. The required body force q (per unit length) is derived from linear elasticity to be

$$q = \frac{2EA}{L^2}\delta_{\text{tip}}$$

from which the acceleration is found to be given by $a = \dfrac{q}{\rho BH}$. The tip nodes were fixed to move only in the x direction. The model parameters used for this test are,

```
h = 1*m; // Shell thickness
Eshell = 1*N/m^2; // Elastic modulus
nu = 0.0; // Poisson's ratio
L = 1*m; // Beam length
B = 1*m; // Beam width
rho = 100*kg/m^3; // Mass density
delta_tip = 100*m; // Target tip displacement

Nsubd = 2; // Number of side subdivisions
```

and the tip displacements

```
Node : 5
 ux = 100.000000, rx = 0.000000
 uy = 0.000000, ry = 0.000000
 uz = 0.000000, rz = 0.000000
Node : 6
 ux = 100.000000, rx = 0.000000
 uy = 0.000000, ry = 0.000000
 uz = 0.000000, rz = 0.000000
```

Which means that in the quadratic displacement field coming from a uniform external load can be captured exactly by this element.

The test file for this case can be found in `(ESSI SOURCE FOLDER)/Verification_Examples/...`
`.../Static_and_Dynamic_Behavior_of_Structural_Elements/...`
`.../Shell_ANDES/static/Test_shell_andes_6_membrane_axial.fei`

Test 7: Transversal tip load This test is identical to test 1, except the beam mesh is placed sideways so that the membrane component is used instead of the bending one. Since the membrane part can capture deformation due to shear, the ratio L/B is set to 10 so that this does not affect the results.

```
h = 0.2*m; // Shell thickness
Eshell = 1*N/m^2; // Elastic modulus
nu = 0.0; // Poisson's ratio
L = 10*m; // Beam length
B = 1.0*m; // Beam height
rho = 0*kg/m^3; // Mass density
d_tip = 100*m; // Target tip displacement

Nsubd = 10; // Number of side subdivisions
```

Results at the tip show less than 1% error with theoretical results when shear component is made small.

```
Node : 21
 ux = -7.513920, rx = 0.000000
 uy = -100.650000, ry = 0.000000
 uz = 0.000000, rz = -15.072400
Node : 22
 ux = 7.513920, rx = 0.000000
 uy = -100.650000, ry = 0.000000
 uz = 0.000000, rz = -15.072400
```

It would be interestign to test this component vs. a beam theory which incorporates deformation due to shear such as Timoshenko beams.

The test file for this case can be found in `(ESSI SOURCE FOLDER)/Verification_Examples/...`
`.../Static_and_Dynamic_Behavior_of_Structural_Elements/...`
`.../Shell_ANDES/static/Test_shell_andes_7_membrane_transverse.fei`

Test 8: Transversal self-weight load This test is analogous to test 4, except the membrane component is being tested instead of the bending. Again, the L/B ratio is kept at 10 to avoid shear deformation creeping into the results perceptively.

```
h = 1*m; // Shell thickness
Eshell = 1*N/m^2; // Elastic modulus
nu = 0.0; // Poisson's ratio
L = 10*m; // Beam length
B = 0.5*m; // Beam width
rho = 100*kg/m^3; // Mass density
delta_tip = 100*m; // Target tip displacement

Nsubd = 10; // Number of side subdivisions
```

Displacement results at the tip nodes show less than 1% error when compared to Bernoulli beam theory.

```
Node : 21
 ux = 3.325110, rx = 0.000000
 uy = 99.780900, ry = 0.000000
 uz = 0.000000, rz = 13.301200
```

```
Node : 22
 ux = -3.325110, rx = 0.000000
 uy = 99.780900, ry = 0.000000
 uz = 0.000000, rz = 13.301200
```

The test file for this case can be found in `(ESSI SOURCE FOLDER)/Verification_Examples/...`

`.../Static_and_Dynamic_Behavior_of_Structural_Elements/...`

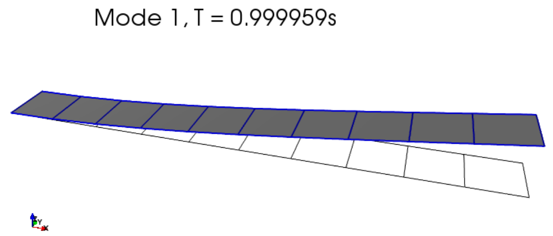
`.../Shell_ANDES/static/Test_shell_andes_8_membrane_transverse.fei`

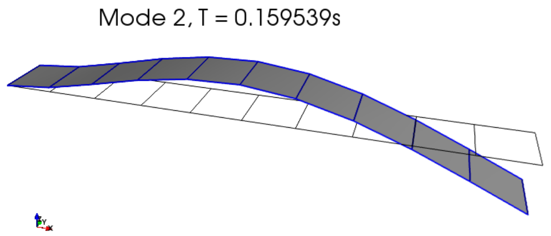
308.5.2 Dynamic Tests

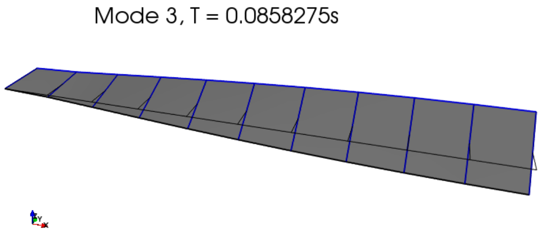
The purpose of the dynamic tests is to verify that the mass matrix adopted for these elements (which is not consistent with the stiffness) is adequate for Dynamic analysis. Also tested herein are the geometric transformations.

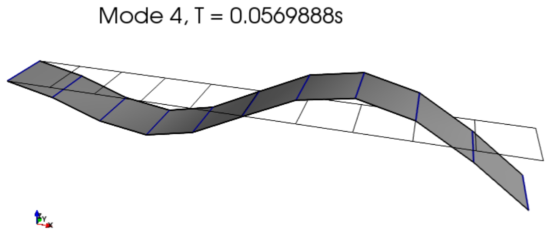
Again, the tests are divided into bending and membrane components which are tested independently. Also, an eigenvalue analysis is performed to verify accordance with theoretical results for continuous Bernoulli beams. Finally, geometric transformations are tested by performing an eigenvalue analysis for the same (unrestrained) beam in different orientations (pitch, yaw and roll) which should have invariant eigenvalues.

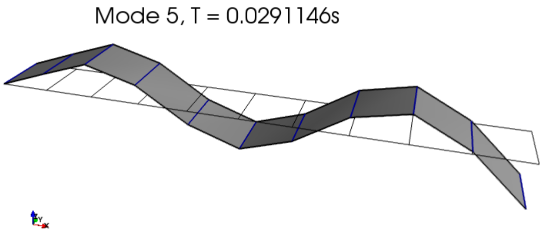
308.5.2.1 Bending Component

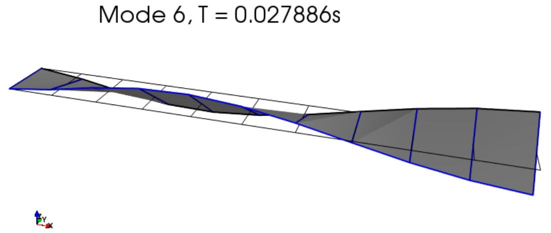


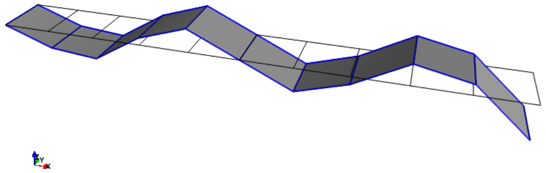


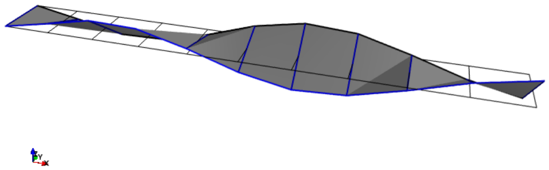


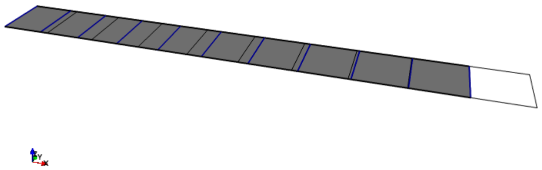


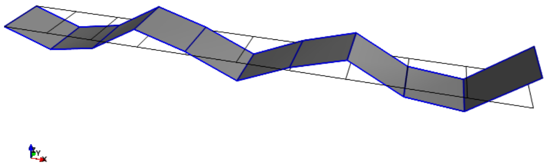




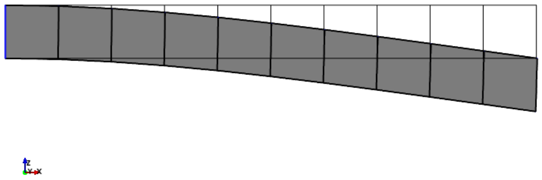

Mode 7, T = 0.0176583s

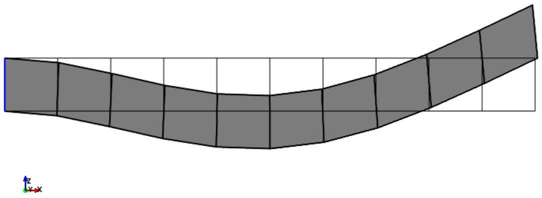

Mode 8, T = 0.015962s

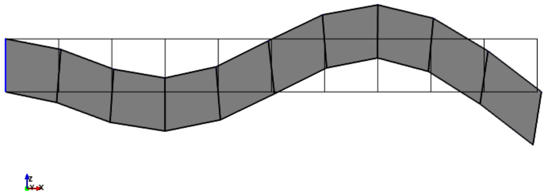

Mode 9, T = 0.0129099s

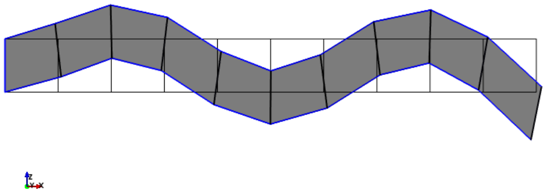

Mode 10, T = 0.0118701s

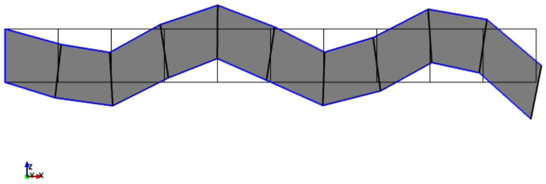
308.5.2.2 Membrane Component


Mode 1, T = 0.998022s


Mode 2, T = 0.15862s

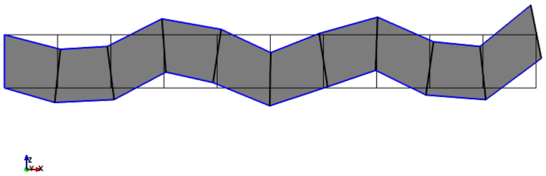

Mode 3, T = 0.0564893s


Mode 4, T = 0.0287508s

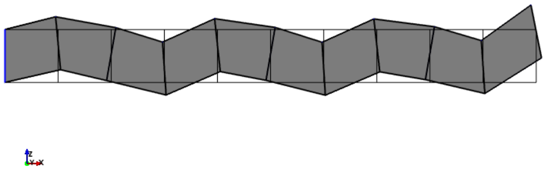

Mode 5, T = 0.0173475s


Mode 6, T = 0.0129099s

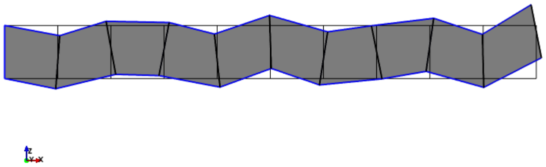
Mode 7, T = 0.0115949s



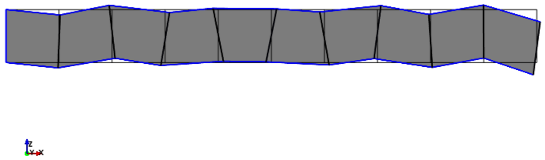
Mode 8, T = 0.00833005s



Mode 9, T = 0.00637652s



Mode 10, T = 0.00526733s



308.5.2.3 Geometric Transformations

308.6 Verification of 4NodeANDES elements

308.6.1 Verification of 4NodeANDES cantilever beams

Problem description: Length=6m, Width=1m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.0$. Use the shear deformation coefficient $\kappa = 1.2$. The force direction was shown in Figure (308.10).

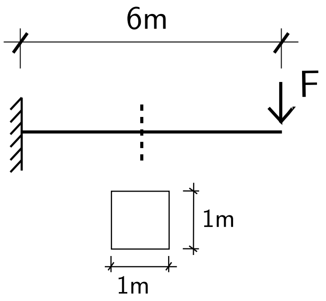


Figure 308.10: Problem description for cantilever beams.

Theoretical displacement (bending and shear deformation):

$$
\begin{aligned}
d &= \frac{FL^3}{3EI} + \frac{FL}{GA_v} \\
&= \frac{FL^3}{3E\frac{bh^3}{12}} + \frac{FL}{\frac{E}{2(1+\nu)}\frac{bh}{\kappa}} \\
&= \frac{100N \times 6^3 m^3}{3 \times 10^8 N/m^2 \times \frac{1}{12} m^4} + \frac{100N \times 6m}{\frac{10}{2} \times 10^7 N/m^2 \times 1m^2 \times \frac{5}{6}} \\
&= 8.64 \times 10^{-4} m + 0.144 \times 10^{-4} m \\
&= 8.784 \times 10^{-4}\ m
\end{aligned}
\tag{308.1}
$$

4NodeANDES element model:

- Force direction: perpendicular to plane (bending)

When the force direction is perpendicular to the plane, only the bending deformation is calculated in 4NodeANDES elements.

The 4NodeANDES elements were shown in Figure (308.14).

- Force direction: inplane force

When the force direction is inplane, both the bending and shear deformation are calculated in 4NodeANDES elements.

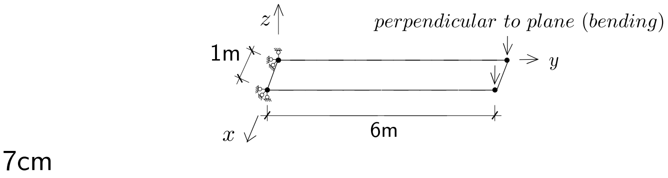


Figure 308.11: One 4NodeANDES element.

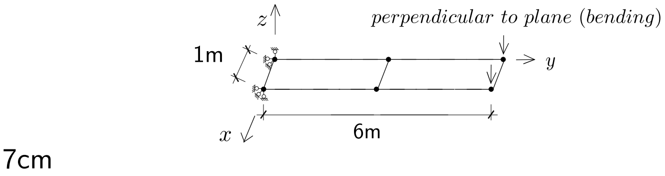


Figure 308.12: Two 4NodeANDES elements.

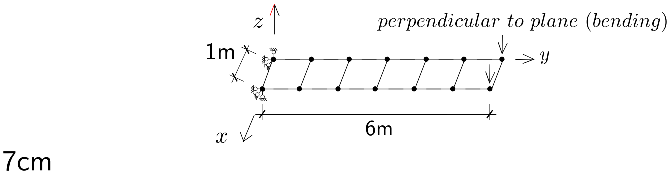


Figure 308.13: Six 4NodeANDES elements.

Figure 308.14: 4NodeANDES elements for cantilever beams under force perpendicular to plane.

The 4NodeANDES elements under inplane force were shown in Figure (308.18).

The Real-ESSI results for the force perpendicular to plane (bending) were listed in Table (308.3). The theoretical solution is 8.784E-04 m. The Real-ESSI results for the inplane force were listed in Table (308.4).

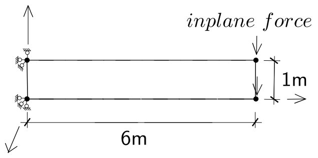
Table 308.3: Results for 4NodeANDES cantilever beams under the force perpendicular to plane (bending).

Element number	1	2	6
4NodeANDES	6.56E-04 m	8.27E-04 m	8.86E-04 m
Error	25.34%	5.87%	0.83%

The theoretical solution is 8.784E-04 m.

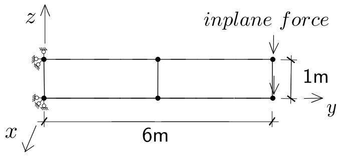
The errors were plotted in Figure (308.21).

The Real-ESSI model fei/DSL files for the table above are HERE.



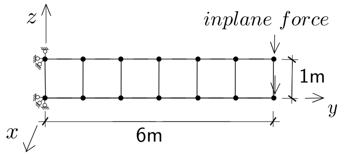
7cm

Figure 308.15: One 4NodeANDES element.



7cm

Figure 308.16: Two 4NodeANDES elements.



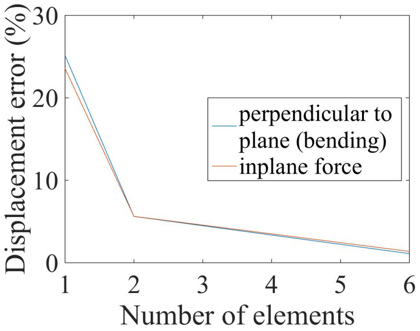
7cm

Figure 308.17: Six 4NodeANDES elements.

Figure 308.18: 4NodeANDES elements for cantilever beams under inplane force.

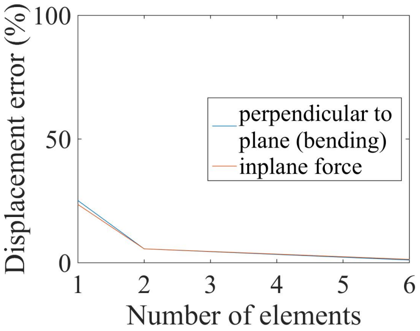
Table 308.4: Results for 4NodeANDES cantilever beams under the inplane force.

Element number	1	2	6
4NodeANDES	6.70E-04 m	8.27E-04 m	8.64E-04 m
Error	23.77%	5.89%	1.65%



7cm

Figure 308.19: Error scale 0% - 30%.



7cm

Figure 308.20: Error scale 0% - 100%.

Figure 308.21: 4NodeANDES cantilever beam for different element number˙ Displacement error versus Number of elements

308.6.2 Verification of 4NodeANDES cantilever beam for different Poisson's ratio

Problem description: Length=6m, Width=1m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.0 - 0.49$. The force direction was shown in Figure (308.22).

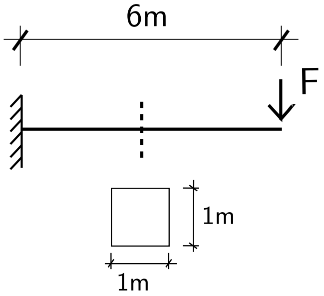


Figure 308.22: Problem description for cantilever beams of different Poisson's ratios.

The theoretical solution for $\nu = 0.0$ was calculated below, while the solution for other Poisson's ratio were calculated by the similar process.

Theoretical displacement (bending and shear deformation):

$$\begin{aligned} d &= \frac{FL^3}{3EI} + \frac{FL}{GA_v} \\ &= \frac{FL^3}{3E\frac{bh^3}{12}} + \frac{FL}{\frac{E}{2(1+\nu)}\frac{bh}{\kappa}} \\ &= \frac{100N \times 6^3 m^3}{3 \times 10^8 N/m^2 \times \frac{1}{12} m^4} + \frac{100N \times 6m}{\frac{10}{2} \times 10^7 N/m^2 \times 1m^2 \times \frac{5}{6}} \\ &= 8.64 \times 10^{-4} m + 0.144 \times 10^{-4} m \\ &= 8.784 \times 10^{-4}\ m \end{aligned} \tag{308.2}$$

The rotation angle at the end:

$$\theta = \frac{FL^2}{2EI} = \frac{100N \times 6^2 m^2}{2 \times 10^8 N/m^2 \times \frac{1}{12} m^4} = 2.16 \times 10^{-4}\ rad = 0.0124^\circ \tag{308.3}$$

The 4NodeANDES elements for cantilever beams of different Poisson's ratios were shown in Figure (308.23) and (308.24):

The Real-ESSI results for the force perpendicular to plane (bending) were listed in Table (308.5) - (308.7).

The errors were plotted in Figure (308.27).

The Real-ESSI results for the inplane force were listed in Table (308.8) - (308.10).

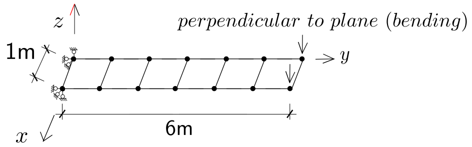


Figure 308.23: 4NodeANDES elements for different Poisson's ratios under the force perpendicular to plane (bending).

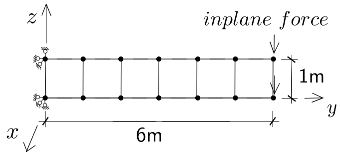


Figure 308.24: 4NodeANDES elements for different Poisson's ratios under the inplane force.

Table 308.5: Displacement error results for 4NodeANDES with element side length 1 m under the force perpendicular to plane (bending).

Poisson's ratio	4NodeANDES displacement	Theory displacement (bending)	Theory displacement (shear)	Theory displacement(all)	Error
0.00	8.639E-04 m	8.640E-04 m	1.440E-05 m	8.784E-04 m	1.38%
0.05	8.635E-04 m	8.640E-04 m	1.512E-05 m	8.791E-04 m	1.49%
0.10	8.622E-04 m	8.640E-04 m	1.586E-05 m	8.799E-04 m	1.71%
0.15	8.599E-04 m	8.640E-04 m	1.659E-05 m	8.806E-04 m	2.04%
0.20	8.566E-04 m	8.640E-04 m	1.734E-05 m	8.813E-04 m	2.48%
0.25	8.522E-04 m	8.640E-04 m	1.808E-05 m	8.821E-04 m	3.05%
0.30	8.466E-04 m	8.640E-04 m	1.884E-05 m	8.828E-04 m	3.75%
0.35	8.398E-04 m	8.640E-04 m	1.959E-05 m	8.836E-04 m	4.59%
0.40	8.315E-04 m	8.640E-04 m	2.035E-05 m	8.844E-04 m	5.60%
0.45	8.216E-04 m	8.640E-04 m	2.111E-05 m	8.851E-04 m	6.78%
0.49	8.124E-04 m	8.640E-04 m	2.173E-05 m	8.857E-04 m	7.88%

The errors were plotted in Figure (308.27).

plane(bending)˙

The angle results for the force perpendicular to plane (bending) were listed in Table (308.11).

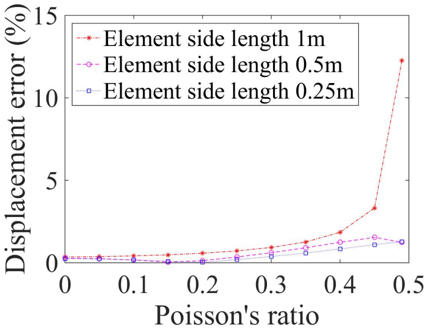
The errors were plotted in Figure (308.33).

Table 308.6: Displacement error results for 4NodeANDES with element side length 0.5 m under the force perpendicular to plane (bending).

Poisson's ratio	4NodeANDES displacement	Theory displacement (bending)	Theory displacement (shear)	Theory displacement(all)	Error
0.00	8.724E-04 m	8.640E-04 m	1.440E-05 m	8.784E-04 m	0.68%
0.05	8.724E-04 m	8.640E-04 m	1.512E-05 m	8.791E-04 m	0.76%
0.10	8.717E-04 m	8.640E-04 m	1.586E-05 m	8.799E-04 m	0.93%
0.15	8.703E-04 m	8.640E-04 m	1.659E-05 m	8.806E-04 m	1.17%
0.20	8.682E-04 m	8.640E-04 m	1.734E-05 m	8.813E-04 m	1.49%
0.25	8.652E-04 m	8.640E-04 m	1.808E-05 m	8.821E-04 m	1.91%
0.30	8.615E-04 m	8.640E-04 m	1.884E-05 m	8.828E-04 m	2.42%
0.35	8.569E-04 m	8.640E-04 m	1.959E-05 m	8.836E-04 m	3.02%
0.40	8.514E-04 m	8.640E-04 m	2.035E-05 m	8.844E-04 m	3.73%
0.45	8.449E-04 m	8.640E-04 m	2.111E-05 m	8.851E-04 m	4.54%
0.49	8.388E-04 m	8.640E-04 m	2.173E-05 m	8.857E-04 m	5.30%

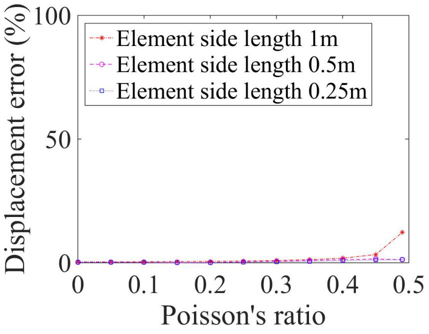
Table 308.7: Displacement error results for 4NodeANDES with element side length 0.25 m under the force perpendicular to plane (bending).

Poisson's ratio	4NodeANDES displacement	Theory displacement (bending)	Theory displacement (shear)	Theory displacement(all)	Error
0.00	8.640E-04 m	8.640E-04 m	1.440E-05 m	8.784E-04 m	1.64%
0.05	8.637E-04 m	8.640E-04 m	1.512E-05 m	8.791E-04 m	1.75%
0.10	8.627E-04 m	8.640E-04 m	1.586E-05 m	8.799E-04 m	1.95%
0.15	8.611E-04 m	8.640E-04 m	1.659E-05 m	8.806E-04 m	2.21%
0.20	8.588E-04 m	8.640E-04 m	1.734E-05 m	8.813E-04 m	2.56%
0.25	8.559E-04 m	8.640E-04 m	1.808E-05 m	8.821E-04 m	2.97%
0.30	8.523E-04 m	8.640E-04 m	1.884E-05 m	8.828E-04 m	3.46%
0.35	8.480E-04 m	8.640E-04 m	1.959E-05 m	8.836E-04 m	4.03%
0.40	8.429E-04 m	8.640E-04 m	2.035E-05 m	8.844E-04 m	4.69%
0.45	8.370E-04 m	8.640E-04 m	2.111E-05 m	8.851E-04 m	5.44%
0.49	8.316E-04 m	8.640E-04 m	2.173E-05 m	8.857E-04 m	6.11%



7cm

Figure 308.25: Error scale 0% - 15%.



7cm

Figure 308.26: Error scale 0% - 100%.

Figure 308.27: 4NodeANDES cantilever beam for force perpendicular to the plane(bending)˙ Displacement error versus Poisson's ratio

The Real-ESSI results for the inplane force were listed in Table (308.14 - (308.16).

The errors were plotted in Figure (308.33).

The Real-ESSI model fei/DSL files for the table above are HERE.

Table 308.8: Displacement error results for 4NodeANDES with element side length 1 m under the inplane force.

Poisson's ratio	4NodeANDES displacement	Theory displacement (bending)	Theory displacement (shear)	Theory displacement(all)	Error
0.00	8.790E-04 m	8.640E-04 m	1.440E-05 m	8.784E-04 m	0.07%
0.05	8.799E-04 m	8.640E-04 m	1.512E-05 m	8.791E-04 m	0.09%
0.10	8.809E-04 m	8.640E-04 m	1.586E-05 m	8.799E-04 m	0.12%
0.15	8.821E-04 m	8.640E-04 m	1.659E-05 m	8.806E-04 m	0.17%
0.20	8.835E-04 m	8.640E-04 m	1.734E-05 m	8.813E-04 m	0.25%
0.25	8.853E-04 m	8.640E-04 m	1.808E-05 m	8.821E-04 m	0.37%
0.30	8.878E-04 m	8.640E-04 m	1.884E-05 m	8.828E-04 m	0.56%
0.35	8.913E-04 m	8.640E-04 m	1.959E-05 m	8.836E-04 m	0.87%
0.40	8.971E-04 m	8.640E-04 m	2.035E-05 m	8.844E-04 m	1.44%
0.45	9.107E-04 m	8.640E-04 m	2.111E-05 m	8.851E-04 m	2.89%
0.49	9.901E-04 m	8.640E-04 m	2.173E-05 m	8.857E-04 m	11.79%

Table 308.9: Displacement error results for 4NodeANDES with element side length 0.5 m under the inplane force.

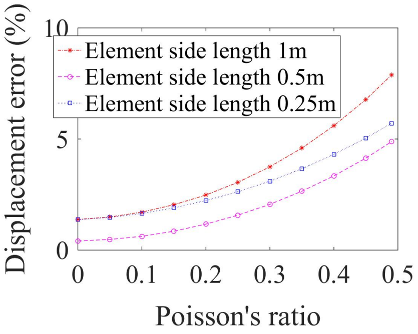
Poisson's ratio	4NodeANDES displacement	Theory displacement (bending)	Theory displacement (shear)	Theory displacement(all)	Error
0.00	8.784E-04 m	8.640E-04 m	1.440E-05 m	8.784E-04 m	0.00%
0.05	8.788E-04 m	8.640E-04 m	1.512E-05 m	8.791E-04 m	0.04%
0.10	8.787E-04 m	8.640E-04 m	1.586E-05 m	8.799E-04 m	0.13%
0.15	8.782E-04 m	8.640E-04 m	1.659E-05 m	8.806E-04 m	0.27%
0.20	8.772E-04 m	8.640E-04 m	1.734E-05 m	8.813E-04 m	0.47%
0.25	8.759E-04 m	8.640E-04 m	1.808E-05 m	8.821E-04 m	0.70%
0.30	8.742E-04 m	8.640E-04 m	1.884E-05 m	8.828E-04 m	0.98%
0.35	8.722E-04 m	8.640E-04 m	1.959E-05 m	8.836E-04 m	1.29%
0.40	8.699E-04 m	8.640E-04 m	2.035E-05 m	8.844E-04 m	1.63%
0.45	8.679E-04 m	8.640E-04 m	2.111E-05 m	8.851E-04 m	1.94%
0.49	8.709E-04 m	8.640E-04 m	2.173E-05 m	8.857E-04 m	1.67%

Table 308.10: Displacement error results for 4NodeANDES with element side length 0.25 m under the inplane force.

Poisson's ratio	4NodeANDES displacement	Theory displacement (bending)	Theory displacement (shear)	Theory displacement(all)	Error
0.00	8.782E-04 m	8.640E-04 m	1.440E-05 m	8.784E-04 m	0.02%
0.05	8.786E-04 m	8.640E-04 m	1.512E-05 m	8.791E-04 m	0.06%
0.10	8.788E-04 m	8.640E-04 m	1.586E-05 m	8.799E-04 m	0.12%
0.15	8.786E-04 m	8.640E-04 m	1.659E-05 m	8.806E-04 m	0.23%
0.20	8.781E-04 m	8.640E-04 m	1.734E-05 m	8.813E-04 m	0.37%
0.25	8.774E-04 m	8.640E-04 m	1.808E-05 m	8.821E-04 m	0.53%
0.30	8.763E-04 m	8.640E-04 m	1.884E-05 m	8.828E-04 m	0.74%
0.35	8.750E-04 m	8.640E-04 m	1.959E-05 m	8.836E-04 m	0.97%
0.40	8.734E-04 m	8.640E-04 m	2.035E-05 m	8.844E-04 m	1.24%
0.45	8.717E-04 m	8.640E-04 m	2.111E-05 m	8.851E-04 m	1.52%
0.49	8.706E-04 m	8.640E-04 m	2.173E-05 m	8.857E-04 m	1.71%

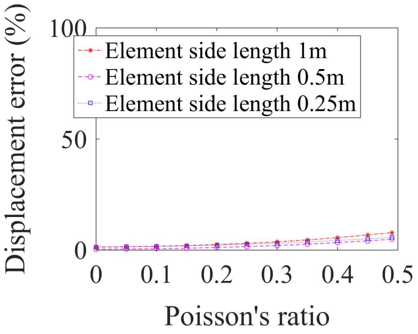
Table 308.11: Rotation angle results for element side length 1 m under the force perpendicular to plane (bending).

Poisson's ratio	4NodeANDES angle (unit:°)	Theory angle (unit:°)	Error
0.00	1.238E-02	1.240E-02	0.19%
0.05	1.237E-02	1.240E-02	0.23%
0.10	1.236E-02	1.240E-02	0.34%
0.15	1.234E-02	1.240E-02	0.52%
0.20	1.230E-02	1.240E-02	0.78%
0.25	1.226E-02	1.240E-02	1.12%
0.30	1.221E-02	1.240E-02	1.54%
0.35	1.214E-02	1.240E-02	2.07%
0.40	1.206E-02	1.240E-02	2.70%
0.45	1.197E-02	1.240E-02	3.46%
0.49	1.188E-02	1.240E-02	4.16%



7cm

Figure 308.28: Error scale 0% - 10%.



7cm

Figure 308.29: Error scale 0% - 100%.

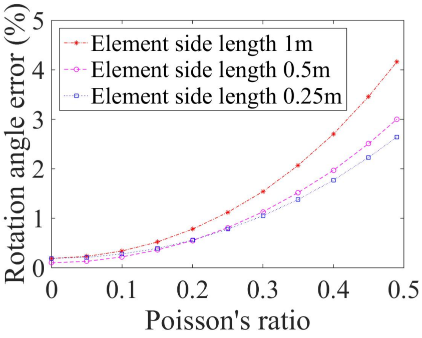
Figure 308.30: 4NodeANDES cantilever beam for inplane force˙ Displacement error versus Poisson's ratio

Table 308.12: Rotation angle results for element side length 0.5 m the force perpendicular to plane (bending).

Poisson's ratio	4NodeANDES angle (unit:°)	Theory angle (unit:°)	Error
0.00	1.239E-02	1.240E-02	0.10%
0.05	1.238E-02	1.240E-02	0.13%
0.10	1.237E-02	1.240E-02	0.22%
0.15	1.236E-02	1.240E-02	0.36%
0.20	1.233E-02	1.240E-02	0.55%
0.25	1.230E-02	1.240E-02	0.81%
0.30	1.226E-02	1.240E-02	1.13%
0.35	1.221E-02	1.240E-02	1.52%
0.40	1.216E-02	1.240E-02	1.97%
0.45	1.209E-02	1.240E-02	2.51%
0.49	1.203E-02	1.240E-02	3.00%

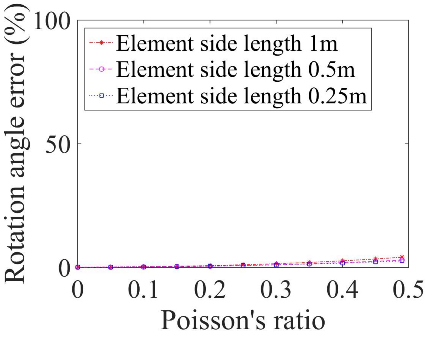
Table 308.13: Rotation angle results for element side length 0.25 m under the force perpendicular to plane (bending).

Poisson's ratio	4NodeANDES angle (unit:°)	Theory angle (unit:°)	Error
0.00	1.238E-02	1.240E-02	0.19%
0.05	1.237E-02	1.240E-02	0.21%
0.10	1.237E-02	1.240E-02	0.28%
0.15	1.235E-02	1.240E-02	0.39%
0.20	1.233E-02	1.240E-02	0.56%
0.25	1.230E-02	1.240E-02	0.78%
0.30	1.227E-02	1.240E-02	1.05%
0.35	1.223E-02	1.240E-02	1.38%
0.40	1.218E-02	1.240E-02	1.77%
0.45	1.212E-02	1.240E-02	2.23%
0.49	1.207E-02	1.240E-02	2.64%



7cm

Figure 308.31: Error scale 0% - 5%.



7cm

Figure 308.32: Error scale 0% - 100%.

Figure 308.33: 4NodeANDES cantilever beam for force perpendicular to the plane(bending). Rotation angle error versus Poisson's ratio

Table 308.14: Rotation angle results for element side length 1 m under the inplane force.

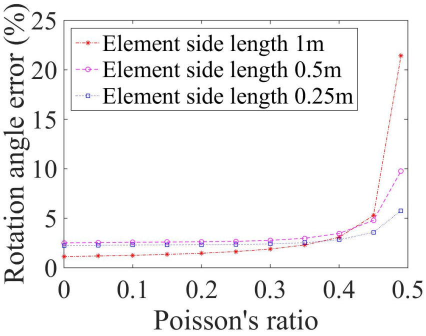
Poisson's ratio	4NodeANDES angle (unit:$^\circ$)	Theory angle (unit:$^\circ$)	Error
0.00	1.254E-02	1.240E-02	1.14%
0.05	1.255E-02	1.240E-02	1.19%
0.10	1.256E-02	1.240E-02	1.26%
0.15	1.257E-02	1.240E-02	1.35%
0.20	1.258E-02	1.240E-02	1.47%
0.25	1.260E-02	1.240E-02	1.64%
0.30	1.263E-02	1.240E-02	1.89%
0.35	1.269E-02	1.240E-02	2.30%
0.40	1.278E-02	1.240E-02	3.08%
0.45	1.305E-02	1.240E-02	5.28%
0.49	1.506E-02	1.240E-02	21.43%

Table 308.15: Rotation angle results for element side length 0.5 m under the inplane force.

Poisson's ratio	4NodeANDES angle (unit:$^\circ$)	Theory angle (unit:$^\circ$)	Error
0.00	1.271E-02	1.240E-02	2.51%
0.05	1.272E-02	1.240E-02	2.56%
0.10	1.272E-02	1.240E-02	2.58%
0.15	1.272E-02	1.240E-02	2.60%
0.20	1.273E-02	1.240E-02	2.63%
0.25	1.273E-02	1.240E-02	2.67%
0.30	1.274E-02	1.240E-02	2.77%
0.35	1.277E-02	1.240E-02	2.98%
0.40	1.283E-02	1.240E-02	3.47%
0.45	1.299E-02	1.240E-02	4.79%
0.49	1.361E-02	1.240E-02	9.78%

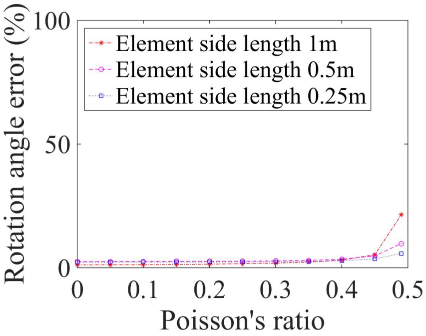
Table 308.16: Rotation angle results for element side length 0.25 m under the inplane force.

Poisson's ratio	4NodeANDES angle (unit:°)	Theory angle (unit:°)	Error
0.00	1.268E-02	1.240E-02	2.24%
0.05	1.268E-02	1.240E-02	2.27%
0.10	1.268E-02	1.240E-02	2.30%
0.15	1.269E-02	1.240E-02	2.31%
0.20	1.269E-02	1.240E-02	2.33%
0.25	1.269E-02	1.240E-02	2.35%
0.30	1.270E-02	1.240E-02	2.41%
0.35	1.271E-02	1.240E-02	2.53%
0.40	1.275E-02	1.240E-02	2.83%
0.45	1.284E-02	1.240E-02	3.58%
0.49	1.312E-02	1.240E-02	5.77%



7cm

Figure 308.34: Error scale 0% - 25%.



7cm

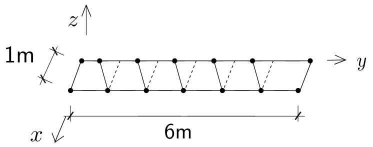
Figure 308.35: Error scale 0% - 100%.

Figure 308.36: 4NodeANDES cantilever beam for inplane force˙ Rotation angle error versus Poisson's ratio

308.6.3 Test of irregular shaped 4NodeANDES cantilever beams

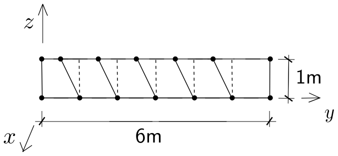
Cantilever model was used as an example. Three different shapes were tested.

In the first test, the upper two nodes of each element were moved one half element size along the $y-axis$, while the lower two nodes were kept at the same location. The element shape was shown in Figure (308.39).



7cm

Figure 308.37: Horizontal plane.



7cm

Figure 308.38: Veritical plane.

Figure 308.39: 4NodeANDES cantilever beam for irregular Shape 1.

In the second test, the upper nodes of each element were moved 50% element size along the $y-axis$, while the lower nodes were moved 50% element size in the other direction along the $y-axis$. The element shape was shown in Figure (308.42).

In the third test, the upper two nodes of each element were moved 90% element size with different directions along the $y-axis$, while the lower nodes were moved 90% element size in the other direction along the $y-axis$. The element shape was shown in Figure (308.45).

The boundary conditions were shown in Figures (308.48), (308.51) and (308.54).

The Real-ESSI results were listed in Table (308.17).

The errors were listed in Tables (308.18) and (308.19).

The Real-ESSI model fei/DSL files for the table above are HERE.

Then, the beam was divided into small elements.

Problem description: Length=6m, Width=1m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.0$. Use the shear deformation coefficient $\kappa = 1.2$. The force direction was shown in Figure (308.55).

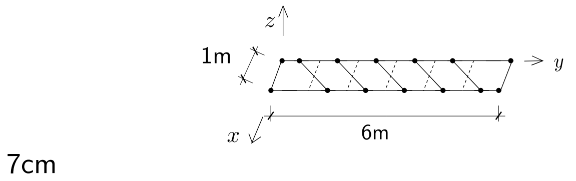


Figure 308.40: Horizontal plane.

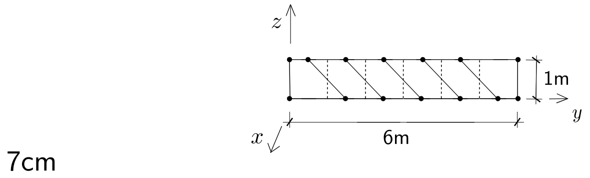


Figure 308.41: Veritical plane.

Figure 308.42: 4NodeANDES cantilever beam for irregular Shape 2.

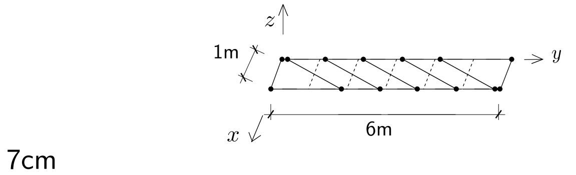


Figure 308.43: Horizontal plane.

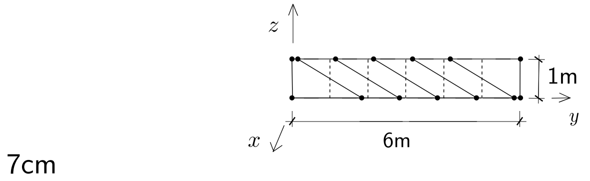
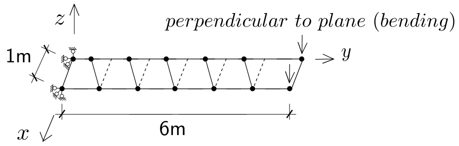


Figure 308.44: Veritical plane.

Figure 308.45: 4NodeANDES cantilever beam for irregular Shape 3.

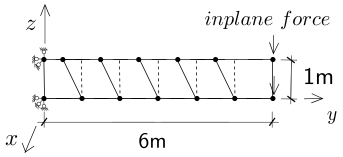
Theoretical displacement (bending and shear deformation):

$$\begin{aligned} d &= \frac{qL^4}{8EI} + \frac{q\frac{L^2}{2}}{GA_v} \\ &= \frac{qL^4}{8E\frac{bh^3}{12}} + \frac{q\frac{L^2}{2}}{\frac{E}{2(1+\nu)}\frac{bh}{\kappa}} \\ &= \frac{400N/m \times 12^4 m^4}{8 \times 10^8 N/m^2 \times \frac{2^4}{12} m^4} + \frac{400N/m \times \frac{12^2}{2} m^2}{\frac{10^8}{2} N/m^2 \times 2m \times 2m \times \frac{5}{6}} \\ &= 7.776 \times 10^{-3} m + 1.728 \times 10^{-4} m \\ &= 7.9488 \times 10^{-3} m \end{aligned} \quad (308.4)$$



7cm

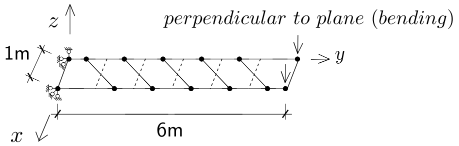
Figure 308.46: Horizontal plane.



7cm

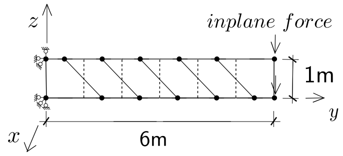
Figure 308.47: Veritical plane.

Figure 308.48: 4NodeANDES cantilever beam boundary conditions for irregular Shape 1.



7cm

Figure 308.49: Horizontal plane.



7cm

Figure 308.50: Veritical plane.

Figure 308.51: 4NodeANDES cantilever beam boundary conditions for irregular Shape 2.

The Real-ESSI displacement results were listed in Table (308.20).

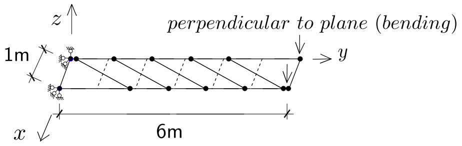
The error were listed in Table (308.21).

The errors were shown in Figures (308.58), (308.61) and (308.64).

The Real-ESSI model fei/DSL files for the table above are HERE.

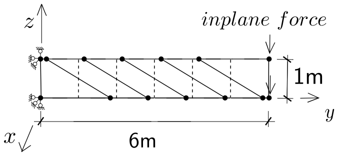
In this section, the beam was cut into smaller elements with element side length 0.5m and 0.25m respectively. And the element side length of the original models is 1.0m. The numerical models were shown in Figure (308.67), (308.70) and (308.73).

Number of division 1:



7cm

Figure 308.52: Horizontal plane.



7cm

Figure 308.53: Veritical plane.

Figure 308.54: 4NodeANDES cantilever beam boundary conditions for irregular Shape 3.

Table 308.17: Results for 4NodeANDES cantilever beams of irregular shapes.

Displacements for irregular shaped element					
Element Type	Force direction	Normal shape	Shape 1	Shape 2	Shape 3
4NodeANDES	perpendicular to plane (bending)	8.639E-04 m	8.602E-04 m	8.534E-04 m	7.851E-04 m
4NodeANDES	inplane force	8.857E-04 m	7.036E-04 m	4.263E-04 m	1.909E-04 m
Theoretical	-	8.784E-04 m	8.784E-04 m	8.784E-04 m	8.784E-04 m

Table 308.18: Errors for irregular shaped 4NodeANDES compared to theoretical solution.

Errors for irregular shaped element, compared to theoretical solutions					
Element Type	Force direction	Normal shape	Shape 1	Shape 2	Shape 3
4NodeANDES	perpendicular to plane (bending)	1.65%	2.07%	2.85%	10.63%
4NodeANDES	inplane force	0.83%	19.90%	51.47%	78.27%

Number of division 2:

Number of division 4:

The Real-ESSI results for the force perpendicular to plane (bending) were listed in Table (308.22). The theoretical solution is 1.60E-5 m.

The Real-ESSI results for the inplane force were listed in Table (308.23). The theoretical solution is 1.60E-5 m.

Table 308.19: Errors for irregular shaped 4NodeANDES compared to normal shape.

Errors for irregular shaped element, compared to normal shape					
Element Type	Force direction	Normal shape	Shape 1	Shape 2	Shape 3
4NodeANDES	perpendicular to plane (bending)	0.00%	0.42%	1.22%	9.12%
4NodeANDES	inplane force	0.00%	20.56%	51.87%	78.45%

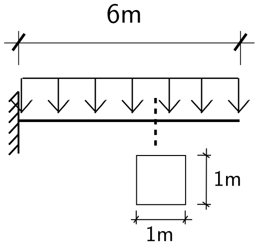


Figure 308.55: Problem description for cantilever beams under uniform pressure .

Table 308.20: Results for 4NodeANDES cantilever beams of irregular shapes with more elements.

Element Type	Shape	Force direction	Number of division		
			1	2	4
4NodeANDES	shape1	perpendicular to plane (bending)	7.750E-03 m	7.768E-03 m	7.774E-03 m
4NodeANDES	shape1	inplane force	6.822E-03 m	7.569E-03 m	7.832E-03 m
4NodeANDES	shape2	perpendicular to plane (bending)	7.656E-03 m	7.734E-03 m	7.765E-03 m
4NodeANDES	shape2	inplane force	3.875E-03 m	5.855E-03 m	7.074E-03 m
4NodeANDES	shape3	perpendicular to plane (bending)	6.637E-03 m	7.139E-03 m	7.521E-03 m
4NodeANDES	shape3	inplane force	1.555E-03 m	2.424E-03 m	3.896E-03 m
Theoretical solution			7.9488E-03 m	7.9488E-03 m	7.9488E-03 m

The errors were plotted in Figure (308.76).

The Real-ESSI model fei/DSL files for the table above are HERE.

Table 308.21: Errors for 4NodeANDES cantilever beams of irregular shapes with more elements.

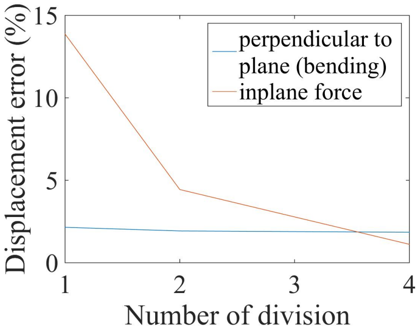
Element Type	Shape	Force direction	Number of division		
			1	2	4
4NodeANDES	shape1	perpendicular to plane (bending)	2.51%	2.28%	2.20%
4NodeANDES	shape1	inplane force	14.18%	4.78%	1.48%
4NodeANDES	shape2	perpendicular to plane (bending)	3.68%	2.71%	2.31%
4NodeANDES	shape2	inplane force	51.25%	26.34%	11.00%
4NodeANDES	shape3	perpendicular to plane (bending)	16.51%	10.19%	5.38%
4NodeANDES	shape3	inplane force	80.44%	69.51%	50.98%

Table 308.22: Results for 4NodeANDES clamped beams under the force perpendicular to plane (bending).

Element Type	Element side length		
	1 m	0.5 m	0.25 m
4NodeANDES	1.347E-05 m	1.35E-05 m	1.35E-05 m
Error	18.36%	18.24%	18.18%

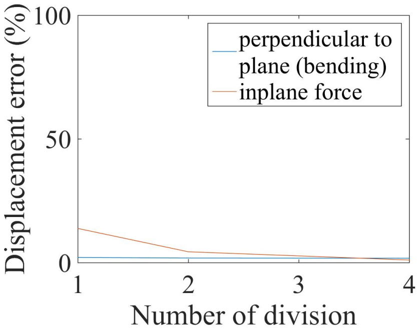
Table 308.23: Results for 4NodeANDES clamped beams under the inplane force.

Element Type	Element side length		
	1 m	0.5 m	0.25 m
4NodeANDES	1.62E-05 m	1.65E-05 m	1.69E-05 m
Error	1.70%	0.12%	2.12%



7cm

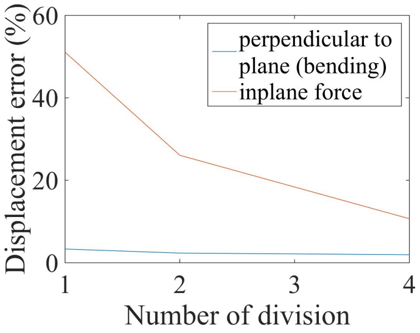
Figure 308.56: Error scale 0% - 15%.



7cm

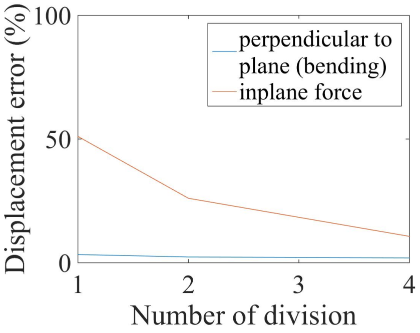
Figure 308.57: Error scale 0% - 100%.

Figure 308.58: 4NodeANDES cantilever beam for irregular Shape 1˙ Displacement error versus Number of division



7cm

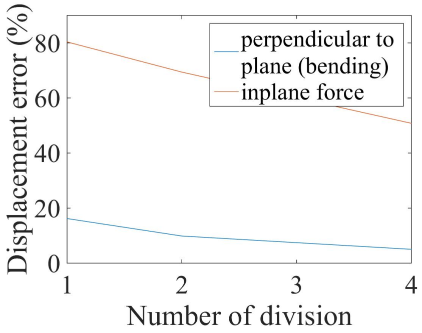
Figure 308.59: Error scale 0% - 60%.



7cm

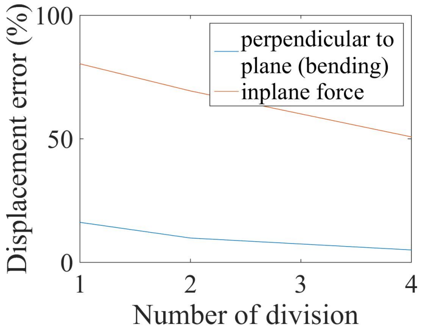
Figure 308.60: Error scale 0% - 100%.

Figure 308.61: 4NodeANDES cantilever beam for irregular Shape 2˙ Displacement error versus Number of division



7cm

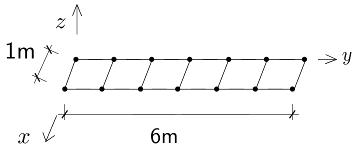
Figure 308.62: Error scale 0% - 80%.



7cm

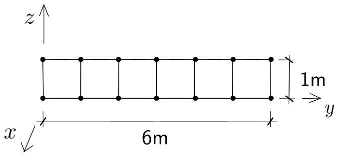
Figure 308.63: Error scale 0% - 100%.

Figure 308.64: 4NodeANDES cantilever beam for irregular Shape 3˙ Displacement error versus Number of division



7cm

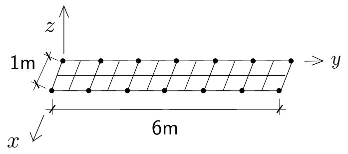
Figure 308.65: Horizontal plane.



7cm

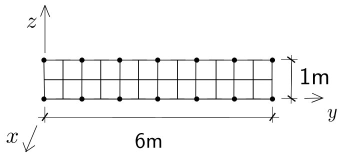
Figure 308.66: Veritical plane.

Figure 308.67: 4NodeANDES clamped beam with element side length 1.0m.



7cm

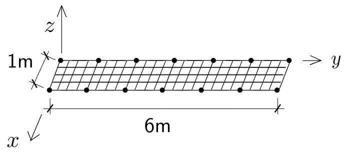
Figure 308.68: Horizontal plane.



7cm

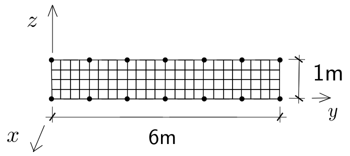
Figure 308.69: Veritical plane.

Figure 308.70: 4NodeANDES clamped beam with element side length 0.5m.



7cm

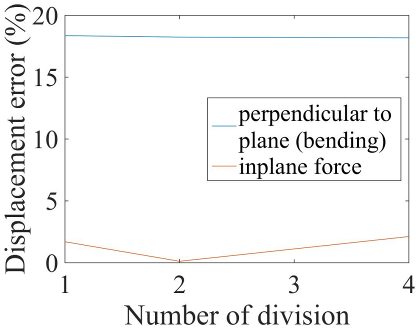
Figure 308.71: Horizontal plane.



7cm

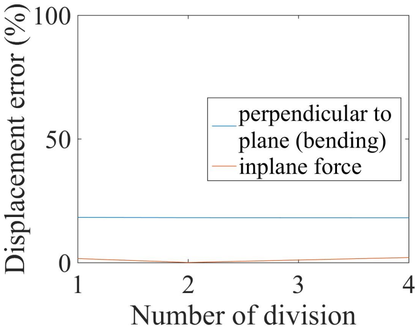
Figure 308.72: Veritical plane.

Figure 308.73: 4NodeANDES clamped beam with element side length 0.25m.



7cm

Figure 308.74: Error scale 0% - 20%.



7cm

Figure 308.75: Error scale 0% - 100%.

Figure 308.76: 4NodeANDES clamped beam for different element number˙ Displacement error versus Number of division

308.6.4 Verification of 4NodeANDES square plate with four edges clamped

Problem description: Length=20m, Width=20m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.3$.

The four edges are clamped.

The load is the uniform normal pressure on the whole plate.

The plate flexural rigidity is

$$D = \frac{Eh^3}{12(1-\nu^2)} = \frac{10^8 N/m^2 \times 1^3 m^3}{12 \times (1-0.3^2)} = 9.1575 \times 10^6 \ N \cdot m \tag{308.5}$$

The theoretical solution is

$$d = \alpha_c \frac{qa^4}{D} = 0.00406 \times \frac{100 N/m^2 \times 20^4 m^4}{9.1575 \times 10^6 \ N \cdot m} = 2.2015 \times 10^{-3} m \tag{308.6}$$

where α_c is a coefficient, which depends on the ratio of plate length to width. In this problem, the coefficient[4] α_c is 0.00406.

The 4NodeANDES were shown in Figures (308.77) - (308.82).

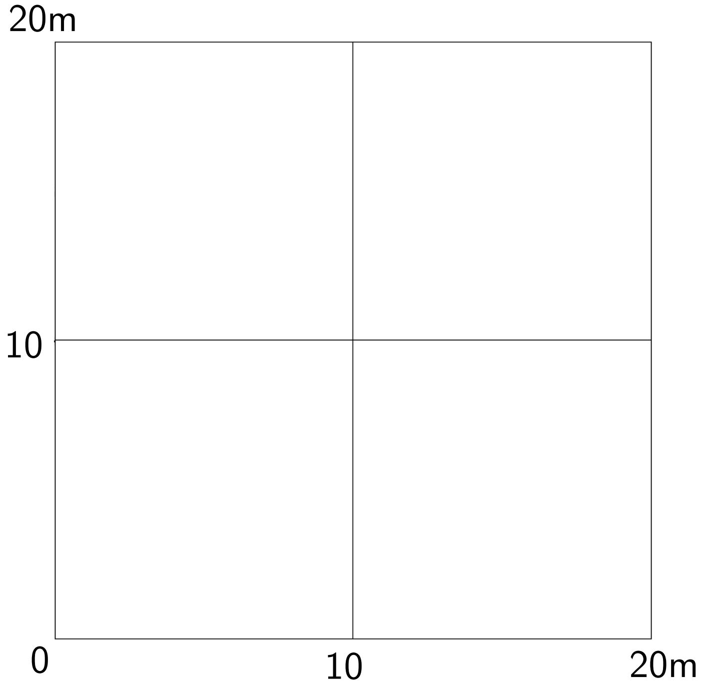


Figure 308.77: 4NodeANDES edge clamped square plate with element side length 10m.

The results were listed in Table (308.24).

The errors were listed in Table (308.25).

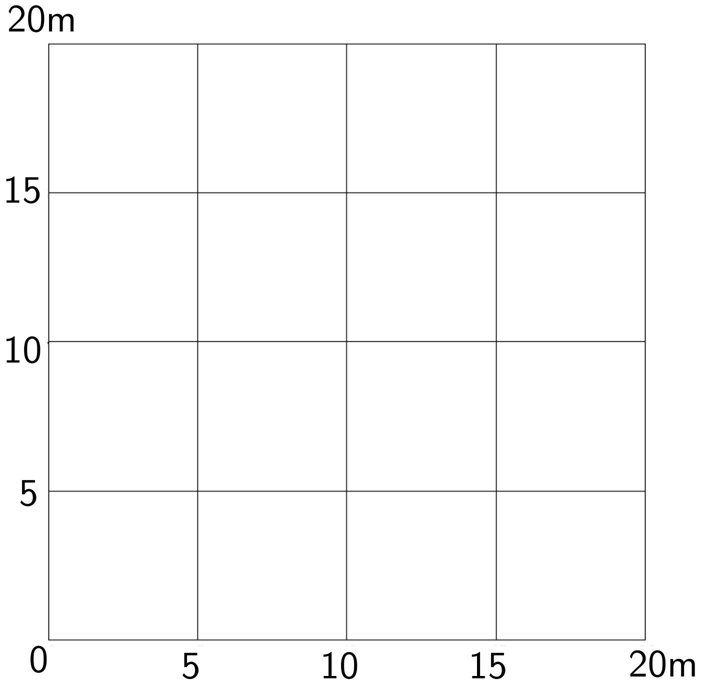


Figure 308.78: 4NodeANDES edge clamped square plate with element side length 5m.

Table 308.24: Results for 4NodeANDES square plate with four edges clamped.

Element type	4NodeANDES	Theoretical
Element side length	Height:1.00m	displacement
10m	2.33E-003 m	2.20E-03 m
5m	2.75E-003 m	2.20E-03 m
2m	2.58E-003 m	2.20E-03 m
1m	2.54E-003 m	2.20E-03 m
0.5m	2.53E-003 m	2.20E-03 m
0.25m	2.53E-003 m	2.20E-03 m

The errors were plotted in Figure (308.85).

The Real-ESSI model fei/DSL files for the table above are HERE.

---

[4] Stephen Timoshenko, Theory of plates and shells (2nd edition). MrGRAW-Hill Inc, page120, 1959.

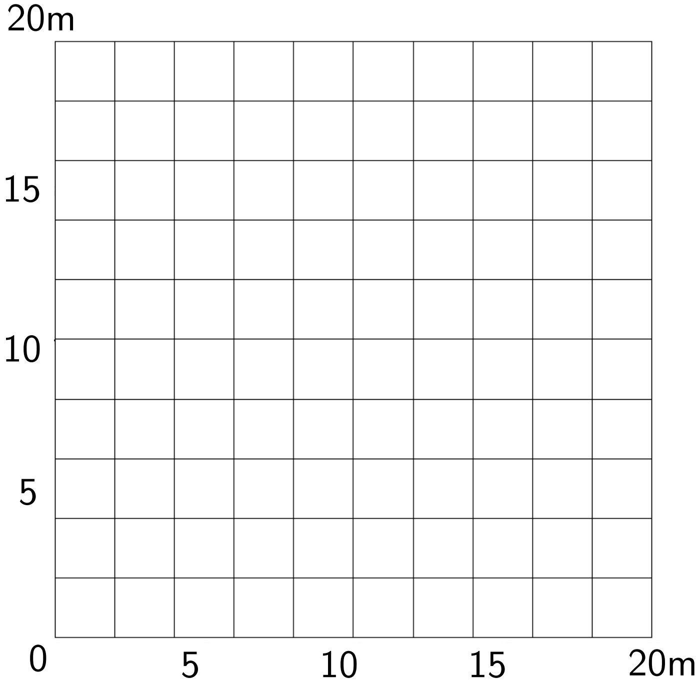


Figure 308.79: 4NodeANDES edge clamped square plate with element side length 2m.

Table 308.25: Errors for 4NodeANDES square plate with four edges clamped.

Element type	4NodeANDES
Element side length	Height:1.00m
10m	5.65%
5m	24.98%
2m	16.97%
1m	15.28%
0.5m	14.84%
0.25m	14.73%

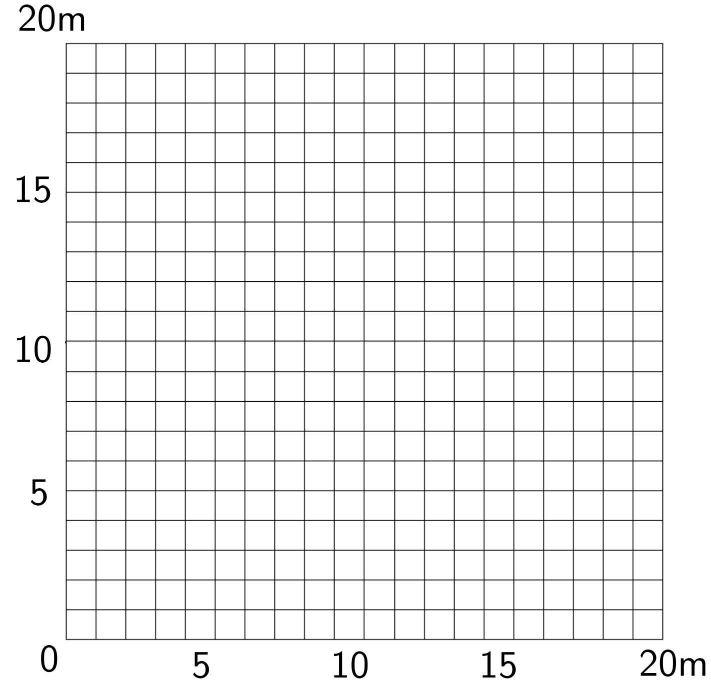


Figure 308.80: 4NodeANDES edge clamped square plate with element side length 1m.

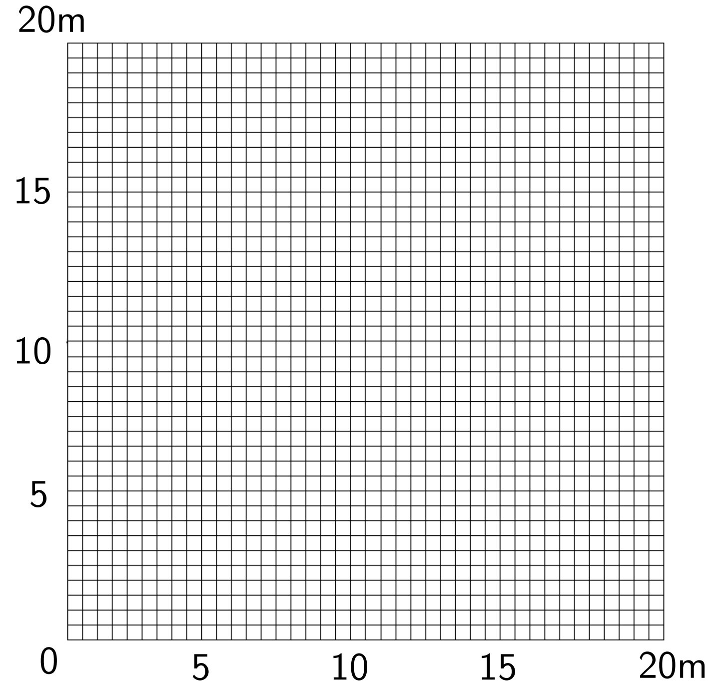


Figure 308.81: 4NodeANDES edge clamped square plate with element side length 0.5m.

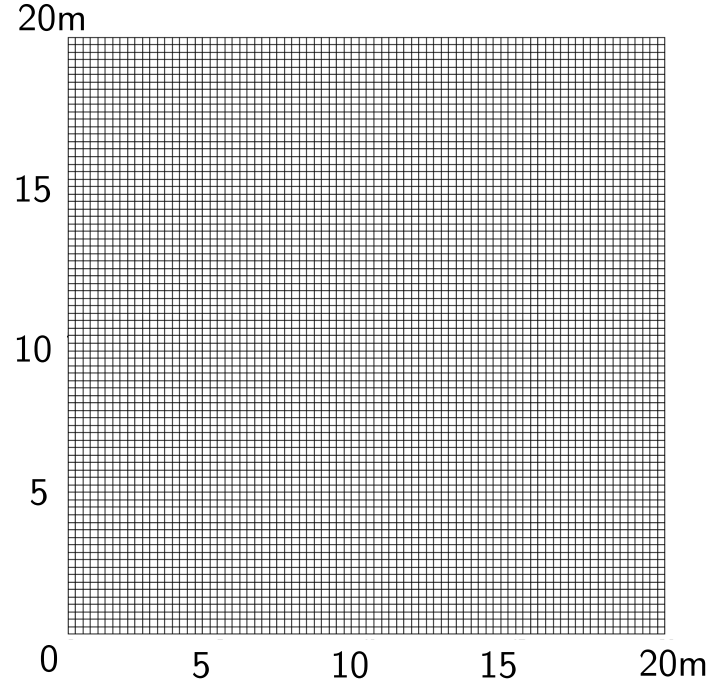
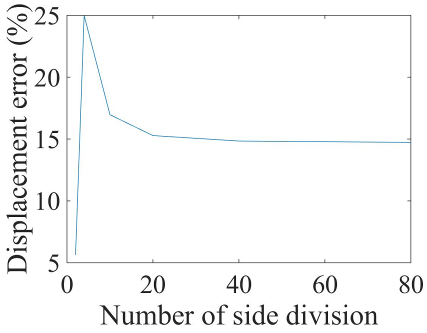
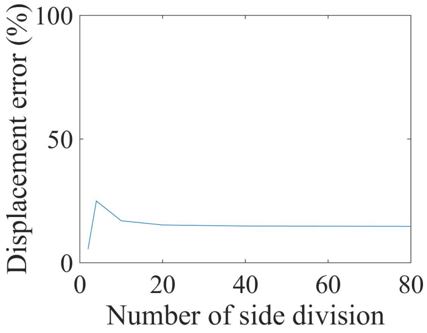


Figure 308.82: 4NodeANDES edge clamped square plate with element side length 0.25m.



7cm

Figure 308.83: Error scale 0% - 25%.



7cm

Figure 308.84: Error scale 0% - 100%.

Figure 308.85: 4NodeANDES square plate with edge clamped˙ Displacement error versus Number of side division

308.6.5 Verification of 4NodeANDES square plate with four edges simply supported

Problem description: Length=20m, Width=20m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.3$.

The four edges are simply supported.

The load is the uniform normal pressure on the whole plate.

The plate flexural rigidity is

$$D = \frac{Eh^3}{12(1-\nu^2)} = \frac{10^8 N/m^2 \times 1^3 m^3}{12 \times (1-0.3^2)} = 9.1575 \times 10^6 \; N \cdot m \tag{308.7}$$

The theoretical solution is

$$d = \alpha_s \frac{qa^4}{D} = 0.00126 \times \frac{100 N/m^2 \times 20^4 m^4}{9.1575 \times 10^6 \; N \cdot m} = 7.0936 \times 10^{-3} m \tag{308.8}$$

where α_s is a coefficient, which depends on the ratio of plate length to width. In this problem, the coefficient[5] α_s is 0.00126.

The 4NodeANDES were shown in Figure (308.86) - (308.91).

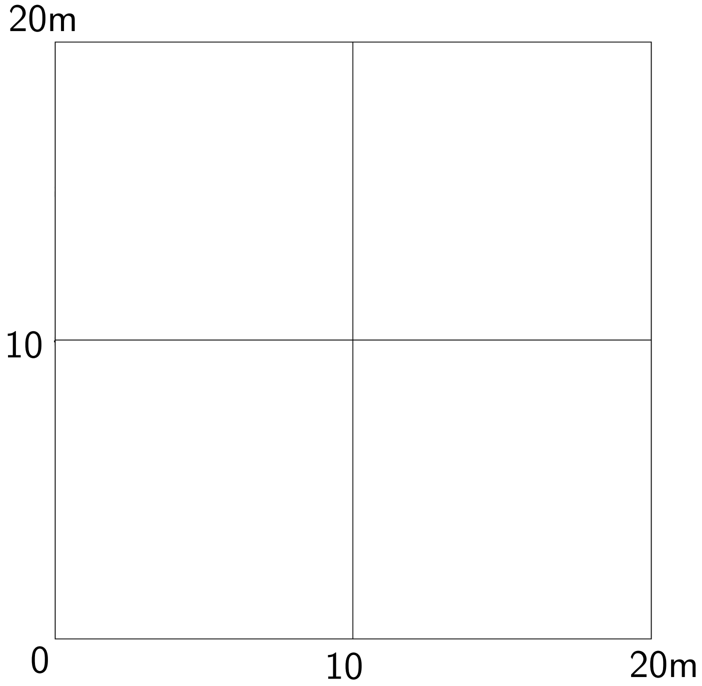


Figure 308.86: 4NodeANDES edge simply supported square plate with element side length 10m.

The results were listed in Table (308.26).

The errors were listed in Table (308.27).

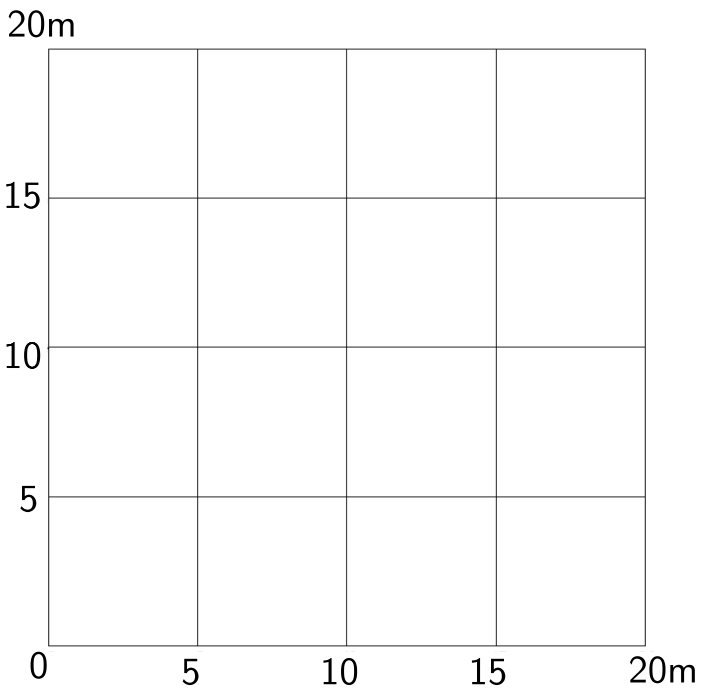


Figure 308.87: 4NodeANDES edge simply supported square plate with element side length 5m.

Table 308.26: Results for 4NodeANDES square plate with four edges simply supported.

Element type	4NodeANDES	Theoretical
Element side length	Height:1.00m	displacement
10m	1.14E-002 m	7.09E-03 m
5m	1.03E-002 m	7.09E-03 m
2m	9.78E-003 m	7.09E-03 m
1m	9.70E-003 m	7.09E-03 m
0.5m	9.68E-003 m	7.09E-03 m
0.25m	9.67E-003 m	7.09E-03 m

The errors were plotted in Figure (308.94).

The Real-ESSI model fei/DSL files for the table above are HERE.

---

[5] Stephen Timoshenko, Theory of plates and shells (2nd edition). MrGRAW-Hill Inc, page202, 1959.

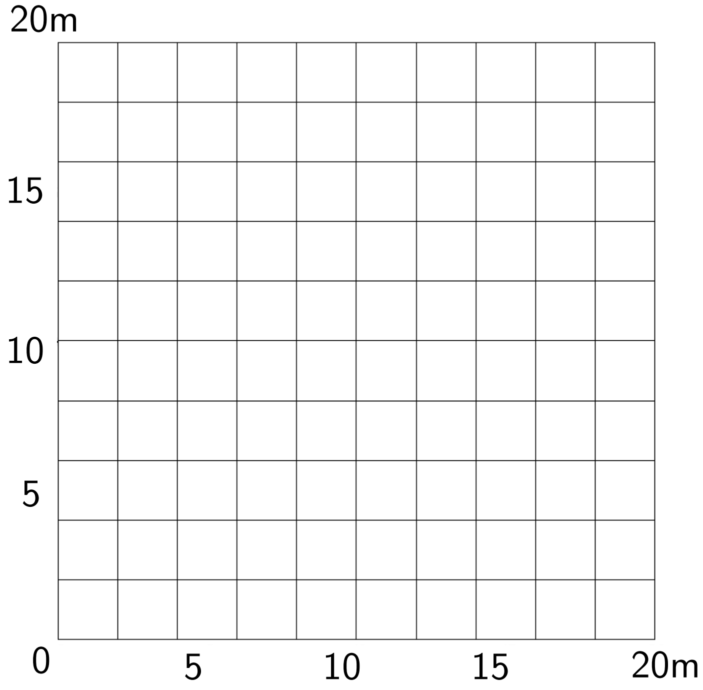


Figure 308.88: 4NodeANDES edge simply supported square plate with element side length 2m.

Table 308.27: Errors for 4NodeANDES square plate with four edges simply supported.

Element type	4NodeANDES
Element side length	Height:1.00m
10m	60.34%
5m	45.14%
2m	37.83%
1m	36.69%
0.5m	36.40%
0.25m	36.32%

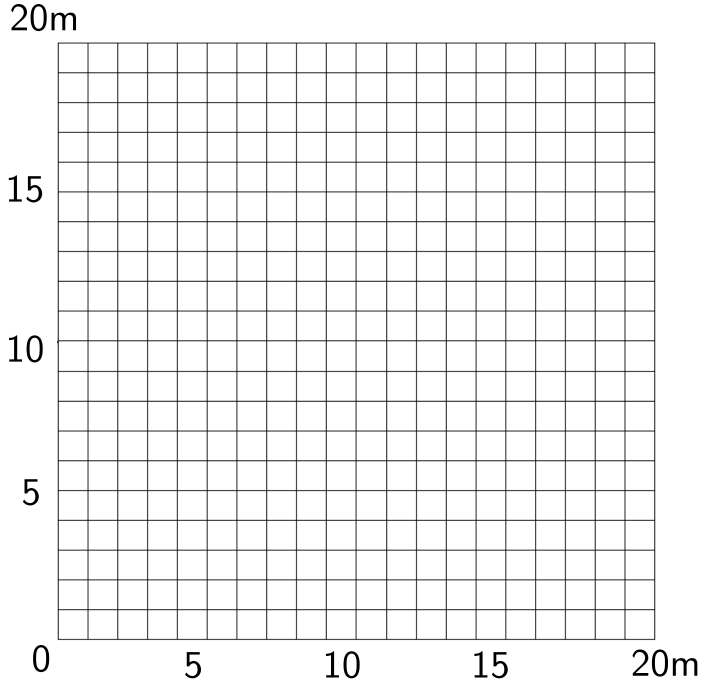


Figure 308.89: 4NodeANDES edge simply supported square plate with element side length 1m.

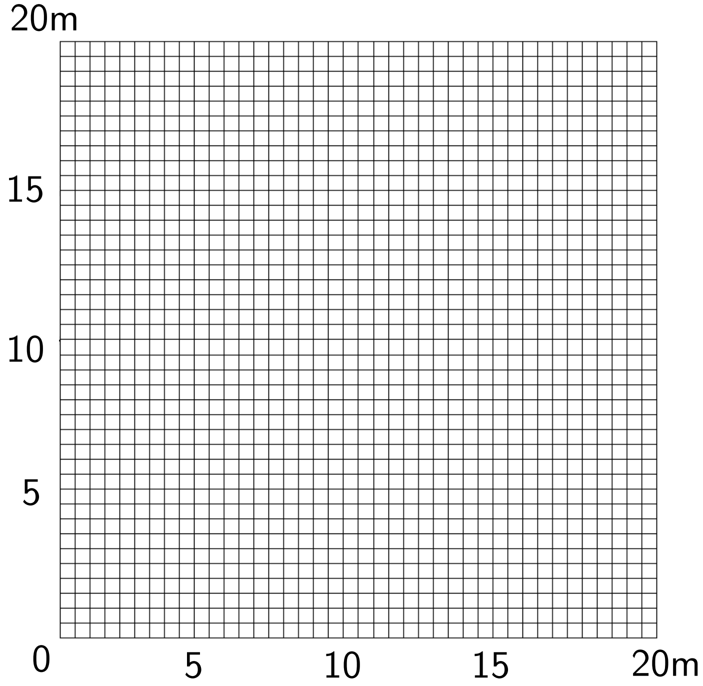


Figure 308.90: 4NodeANDES edge simply supported square plate with element side length 0.5m.

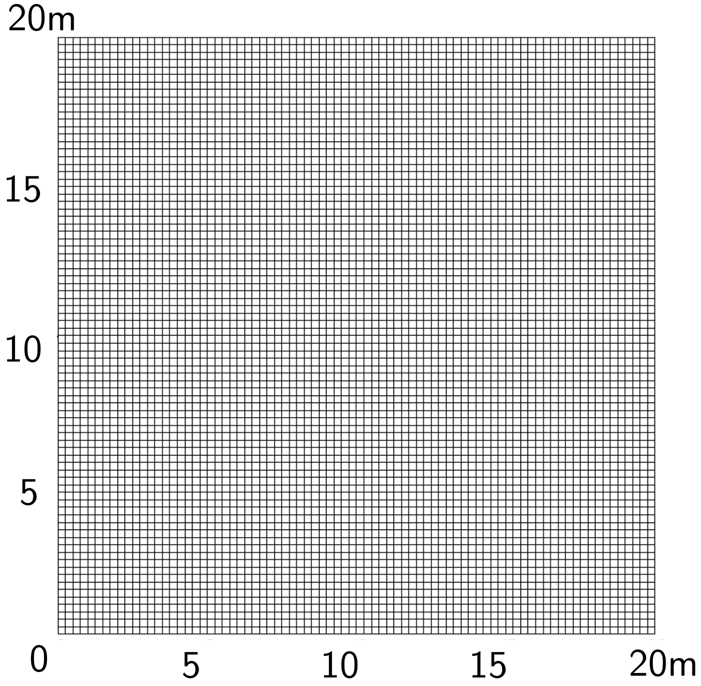
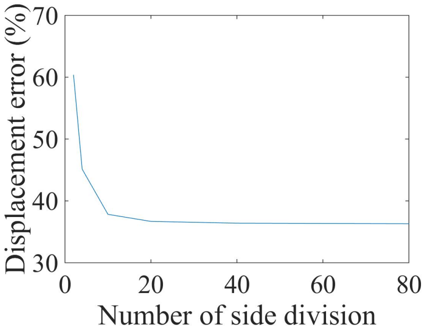
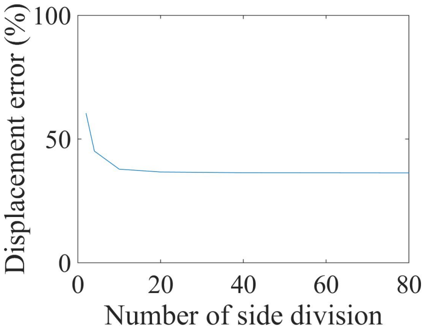


Figure 308.91: 4NodeANDES edge simply supported square plate with element side length 0.25m.



7cm

Figure 308.92: Error scale 0% - 70%.



7cm

Figure 308.93: Error scale 0% - 100%.

Figure 308.94: 4NodeANDES square plate with edge simply supported˙ Displacement error versus Number of side division

308.6.6 Verification of 4NodeANDES circular plate with all edges clamped

Problem description: Diameter=20m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.3$.

The four edges are clamped.

The load is the uniform normal pressure on the whole plate.

The plate flexural rigidity is

$$D = \frac{Eh^3}{12(1-\nu^2)} = \frac{10^8 N/m^2 \times 1^3 m^3}{12 \times (1-0.3^2)} = 9.1575 \times 10^6 \; N \cdot m \tag{308.9}$$

The theoretical solution[6] is

$$d = \frac{qa^4}{64D} = \frac{100 N/m^2 \times 10^4 m^4}{64 \times 9.1575 \times 10^6 \; N \cdot m} = 1.7106 \times 10^{-3} m \tag{308.10}$$

The 4NodeANDES were shown in Figures (308.95) - (308.100).

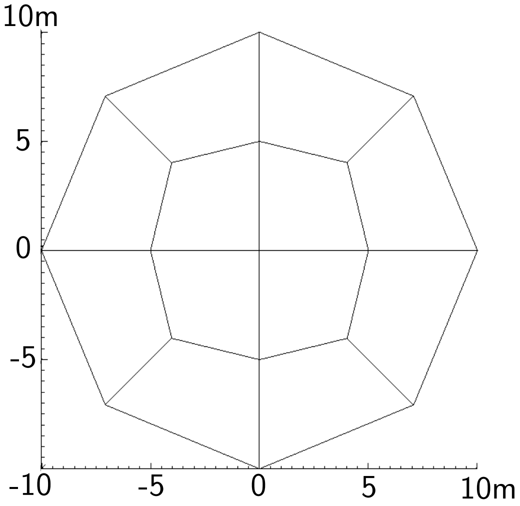


Figure 308.95: 4NodeANDES edge clamped circular plate with element side length 10m.

The results were listed in Table (308.28).

The errors were listed in Table (308.29).

The errors were shown in Figure (308.103).

The Real-ESSI model fei/DSL files for the table above are HERE.

---

[6] Stephen Timoshenko, Theory of plates and shells (2nd edition). MrGRAW-Hill Inc, page55, 1959.

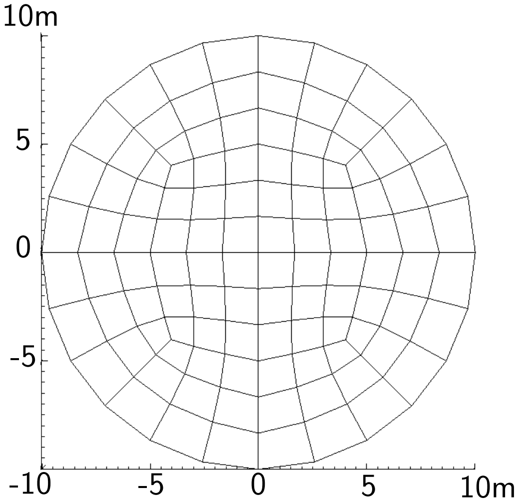


Figure 308.96: 4NodeANDES edge clamped circular plate with element side length 5m.

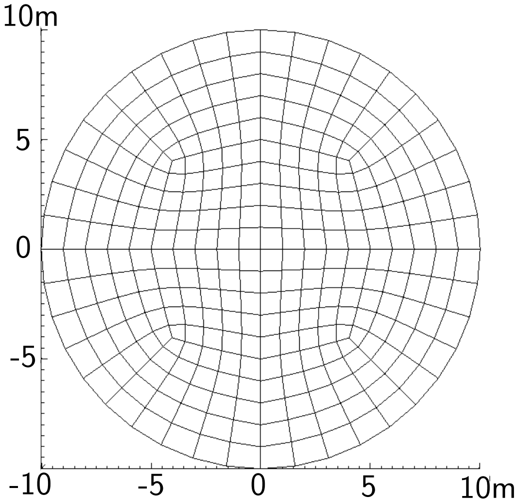


Figure 308.97: 4NodeANDES edge clamped circular plate with element side length 2m.

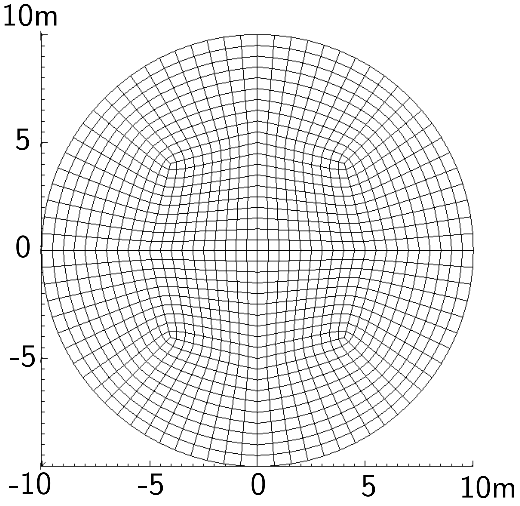


Figure 308.98: 4NodeANDES edge clamped circular plate with element side length 1m.

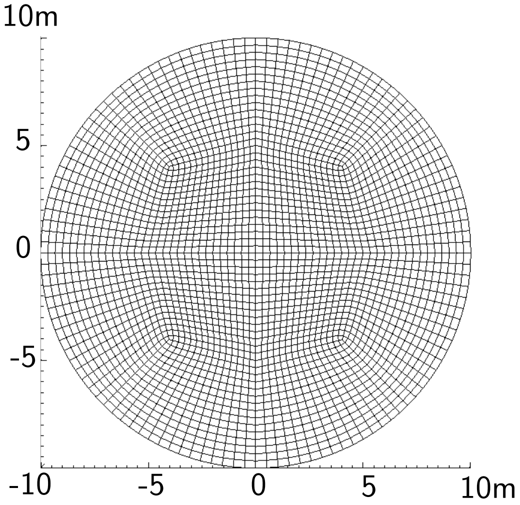


Figure 308.99: 4NodeANDES edge clamped circular plate with element side length 0.5m.

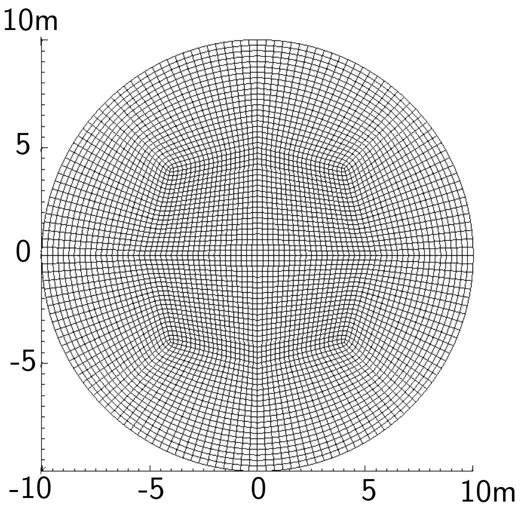


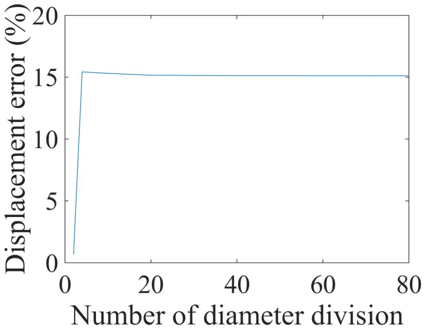
Figure 308.100: 4NodeANDES edge clamped circular plate with element side length 0.25m.

Table 308.28: Results for 4NodeANDES circular plate with four edges clamped.

Element type	4NodeANDES	Theoretical displacement
Element side length	Height:1.00m	
10m	1.69E-003 m	1.706E-03 m
5m	1.97E-003 m	1.706E-03 m
2m	1.97E-003 m	1.706E-03 m
1m	1.96E-003 m	1.706E-03 m
0.5m	1.96E-003 m	1.706E-03 m
0.25m	1.96E-003 m	1.706E-03 m

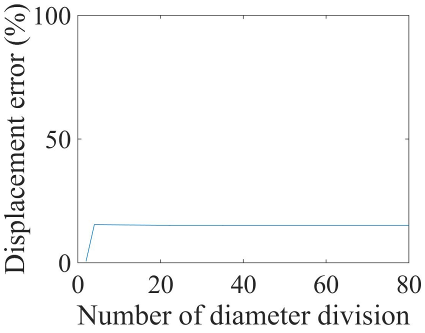
Table 308.29: Errors for 4NodeANDES circular plate with four edges clamped.

Element type	4NodeANDES
Element side length	Height:1.00m
10m	0.71%
5m	15.43%
2m	15.31%
1m	15.16%
0.5m	15.13%
0.25m	15.12%



7cm

Figure 308.101: Error scale 0% - 20%.



7cm

Figure 308.102: Error scale 0% - 100%.

Figure 308.103: 4NodeANDES circular plate with edge clamped˙ Displacement error versus Number of side division

308.6.7 Verification of 4NodeANDES circular plate with all edges simply supported

Problem description: Diameter=20m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.3$.

The four edges are simply supported.

The load is the uniform normal pressure on the whole plate.

The plate flexural rigidity is

$$D = \frac{Eh^3}{12(1-\nu^2)} = \frac{10^8 N/m^2 \times 1^3 m^3}{12 \times (1-0.3^2)} = 9.1575 \times 10^6 \; N \cdot m \tag{308.11}$$

The theoretical solution[7] is

$$d = \frac{(5+\nu)qa^4}{64(1+\nu)D} = \frac{(5+0.3) \times 100N/m^2 \times 10^4 m^4}{64 \times (1+0.3) \times 9.1575 \times 10^6 \; N \cdot m} = 6.956 \times 10^{-3} m \tag{308.12}$$

The 4NodeANDES were shown in Figure (308.104) - (308.109).

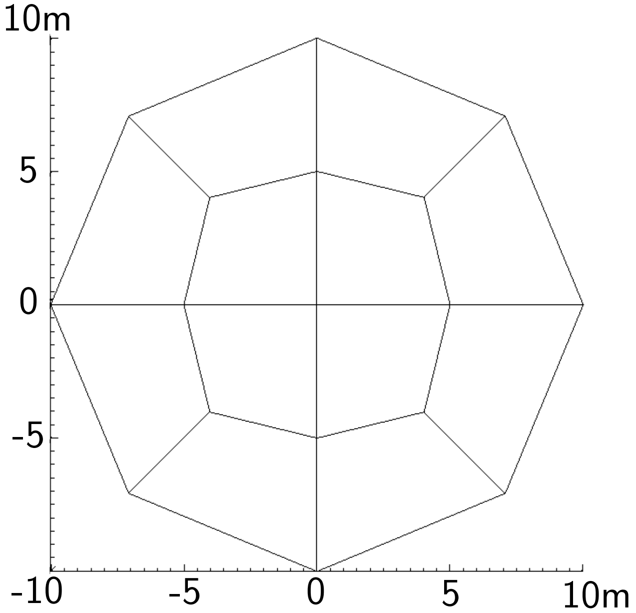


Figure 308.104: 4NodeANDES edge simply supported circular plate with element side length 10m.

The results were listed in Table (308.30).

The errors were listed in Table (308.31).

The errors were plotted in Figure (308.112).

The Real-ESSI model fei/DSL files for the table above are HERE.

---

[7] Stephen Timoshenko, Theory of plates and shells (2nd edition). MrGRAW-Hill Inc, page55, 1959.

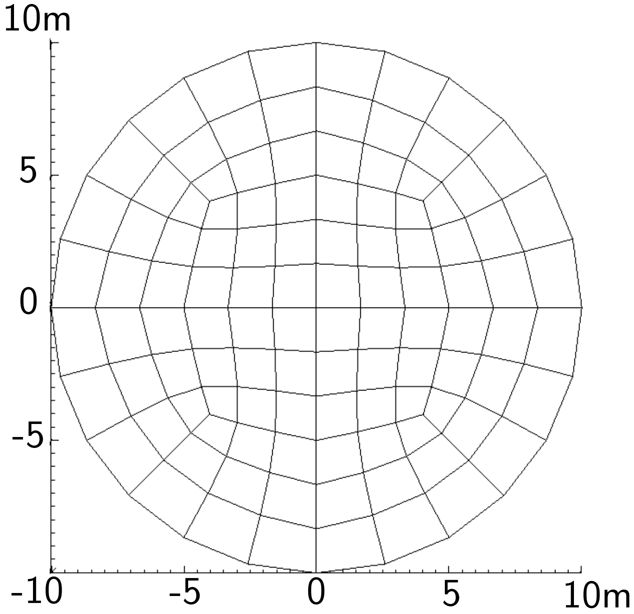


Figure 308.105: 4NodeANDES edge simply supported circular plate with element side length 5m.

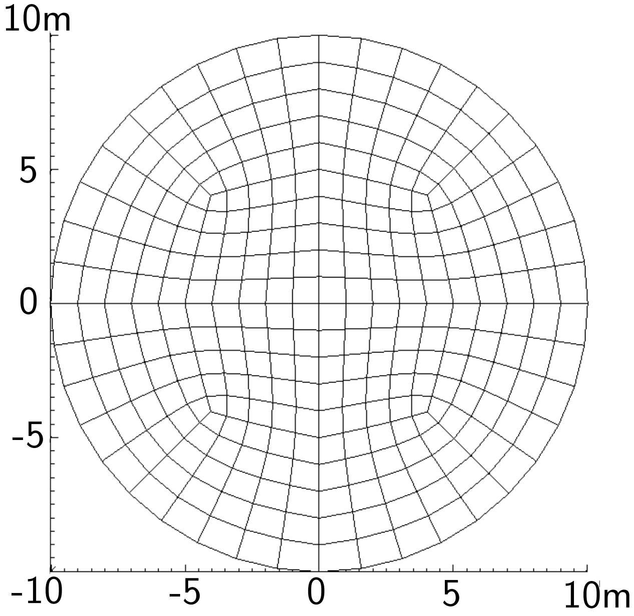


Figure 308.106: 4NodeANDES edge simply supported circular plate with element side length 2m.

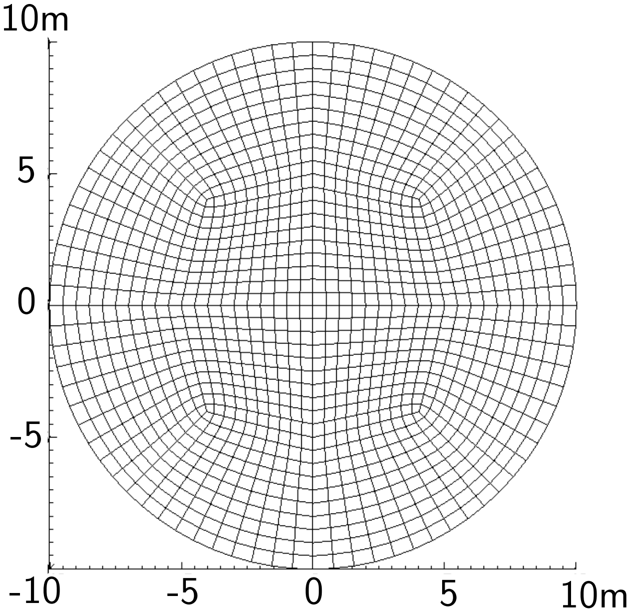


Figure 308.107: 4NodeANDES edge simply supported circular plate with element side length 1m.

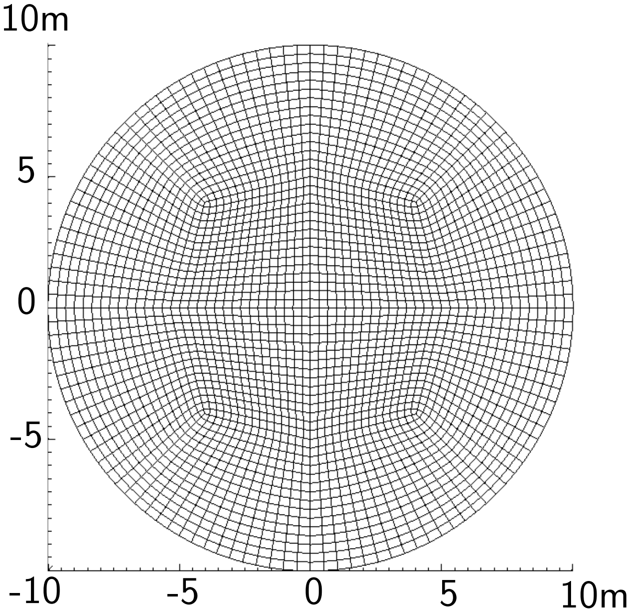


Figure 308.108: 4NodeANDES edge simply supported circular plate with element side length 0.5m.

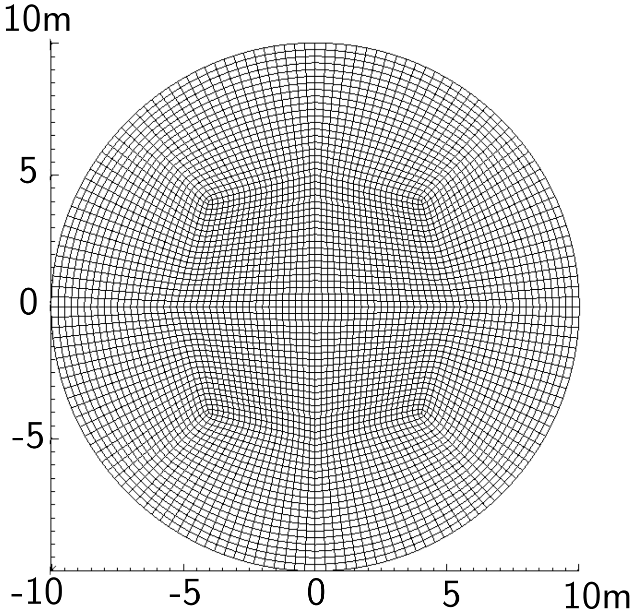


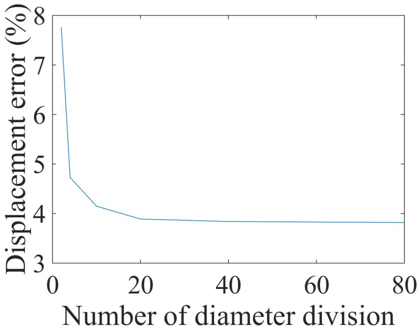
Figure 308.109: 4NodeANDES edge simply supported circular plate with element side length 0.25m.

Table 308.30: Results for 4NodeANDES cicular plate with four edges simply supported.

Element type	4NodeANDES	Theoretical
Element side length	Height:1.00m	displacement
10m	7.50E-003 m	6.956E-03 m
5m	7.29E-003 m	6.956E-03 m
2m	7.25E-003 m	6.956E-03 m
1m	7.23E-003 m	6.956E-03 m
0.5m	7.22E-003 m	6.956E-03 m
0.25m	7.22E-003 m	6.956E-03 m

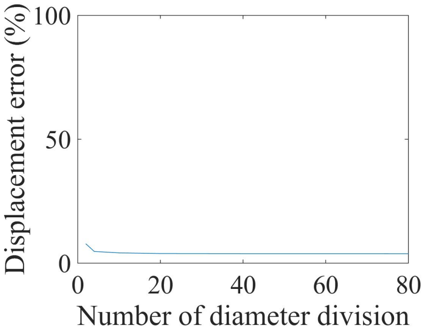
Table 308.31: Errors for 4NodeANDES cicular plate with four edges simply supported.

Element type	4NodeANDES
Element side length	Height:1.00m
10m	7.75%
5m	4.73%
2m	4.15%
1m	3.89%
0.5m	3.84%
0.25m	3.82%



7cm

Figure 308.110: Error scale 0% - 8%.



7cm

Figure 308.111: Error scale 0% - 100%.

Figure 308.112: 4NodeANDES circular plate with edge simply supported˙ Displacement error versus Number of side division

Chapter 309

Verification and Validation for Static and Dynamic Behavior of Special Elements (Contacts/Interfaces/Joints, Gap/Frictional, Isolators)

(2010-2011-2016-2017-2019-2021-)

(In collaboration with Prof. Sumeet Kumar Sinha and Dr. Yuan Feng, and Prof. Han Yang)

309.1 Chapter Summary and Highlights

309.2 Verification of Static Penalty Contact/Interface/Joint Element Modeling and Simulation

References for interface elements: Hird and Russell (1990), AG (2020).

This section presents the verification of Penalty Stiffness based Frictional Contact/Interface/Joint Element using analytical simple solutions to verify the numerical solutions obtained by the application of the developed model. The examples show the response of element for different cases. Solution sensitivity on penalty stiffness is also discussed in details for those examples.

Theoretically, the penalty stiffness should be infinite, but for numerical stability of the solution, it can go upto 10^{16}. This is because for a double precision computer, machine epsilon $\epsilon \approx 10^{-16}$ and thus the corresponding displacement for the penalty springs can go low only till 10^{-16}. For all the cases considred below in this cases, the convergence criteria was as $||\delta U|| <= 10^{-12}$.

309.2.1 Static Normal Contact/Interface/Joint Verification

A Two-bar truss example is considered here to verify the normal contact/interface/joint for different normal loading conditions and different penalty stiffness K_n. This is an example of normal loading on a 1-D contact/interface/joint between two bars separated by an initial gap of $\delta_{in} = 0.1m$. An illustrative diagram of the problem statement is shown below.

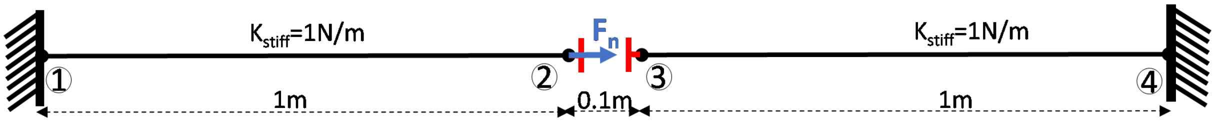


Figure 309.1: Illustration of two bar normal Contact/Interface/Joint problem under monotonic loading with inital gap.

A snapshot of the code for the contact/interface/joint element is shown below. The Real-ESSI model fei/DSL files for this example can be downloaded HERE.

```
add element #3 type FrictionalPenaltyContact with nodes (2,3)
 normal_stiffness =1e10*N/m
 tangential_stiffness = 1e10*Pa*m
 normal_damping = 0*kN/m*s
 tangential_damping = 0*kN/m*s
 friction_ratio = 0.3
 contact_plane_vector = (1,0,0);
```

309.2.1.1 Case 1: Monotonic Loading with initial gap $\delta_{in} = 0.1m$

In this case a force of 0.3N is applied to Node 2. From Figure 309.2, the solution converges to the analytical result for $K_n = 100N/m$ i.e 100 times the stiffness of bar element. Please note that, the for penalty stiffness $> 10e15$, the convergence fails (when the bars contact/interface/joint mode chnges), as the global stiffness matrix becomes *ill conditions*. Thus, the penalty stiffness cannot be too large.

309.2.1.2 Case 2: Monotonic Loading with no initial gap $\delta_{in} = 0m$

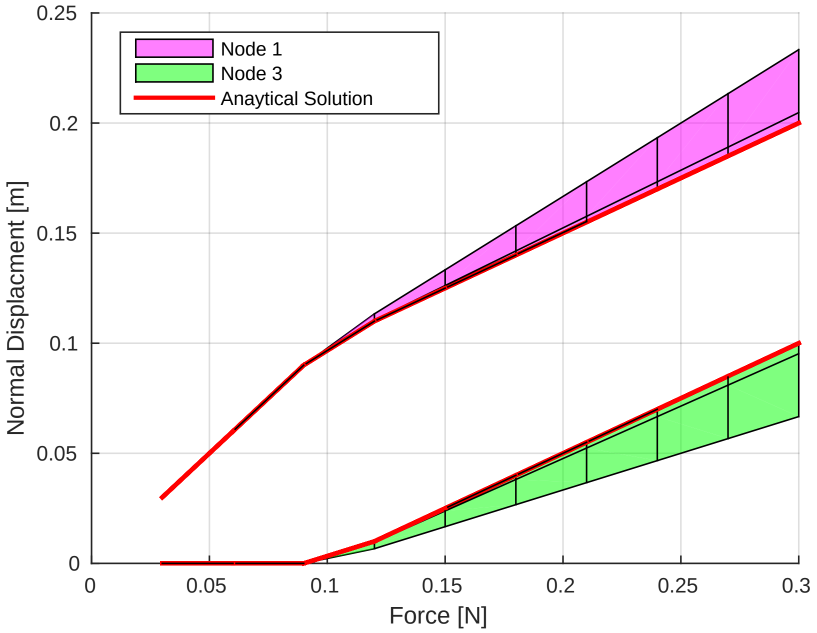
In this case a force of 0.3N is applied to Node 2. From Figure 309.3, the solution again converges to the analytical result for $K_n = 100N/m$ i.e 100 times the stiffness of bar element.

309.2.1.3 Case 3: Cyclic Loading with initial gap $\delta_{in} = 0.1m$

For cyclic loading cases considered below, the loading force F_n applied is shown in Figure 309.4. From Figure 309.5, the solution again converges to the analytical result for $K_n = 100N/m$ i.e 100 times the stiffness of bar element.

309.2.1.4 Case 4: Cyclic Loading with no initial gap $\delta_{in} = 0m$

The same cyclic load shown in Figure 309.4 is again applied fpor this case. From Figure ??, the solution again converges to the analytical result for $K_n = 100N/m$ i.e 100 times the stiffness of bar element.



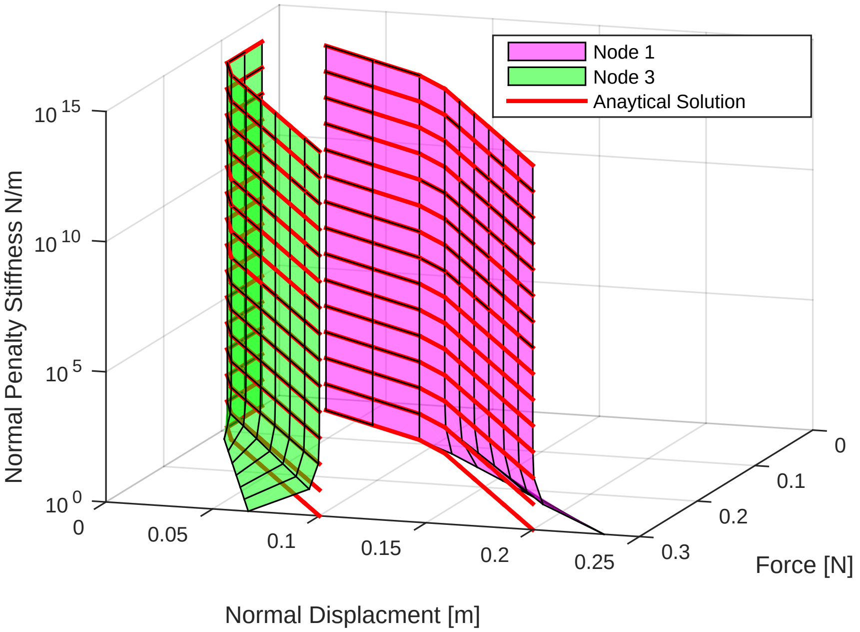


Figure 309.2: Displacements of Node 2 and Node 3 with change in normal penalty stiffness for $\delta_{in} = 0.1m$

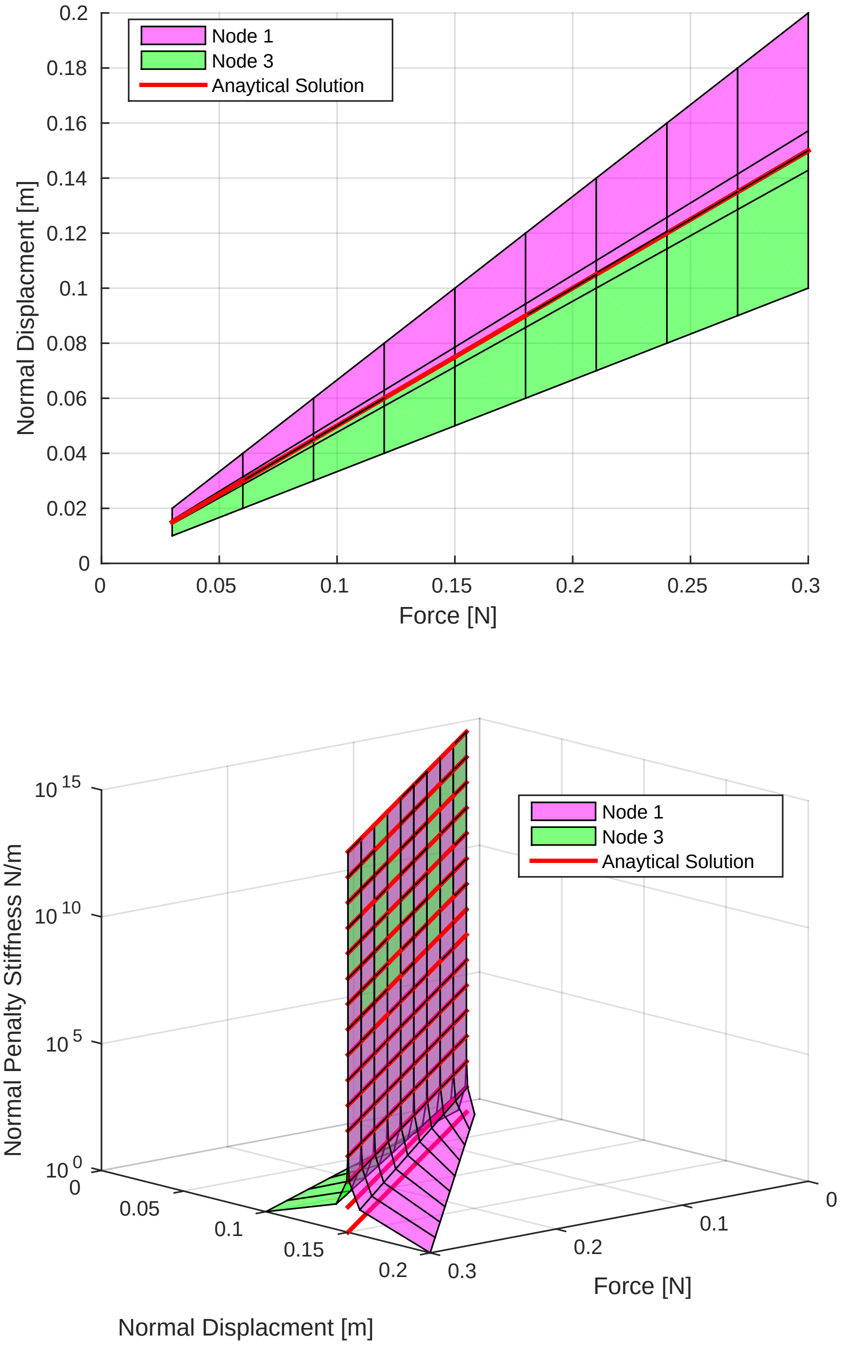


Figure 309.3: Displacements of Node 2 and Node 3 with change in normal penalty stiffness for $\delta_{in} = 0m$

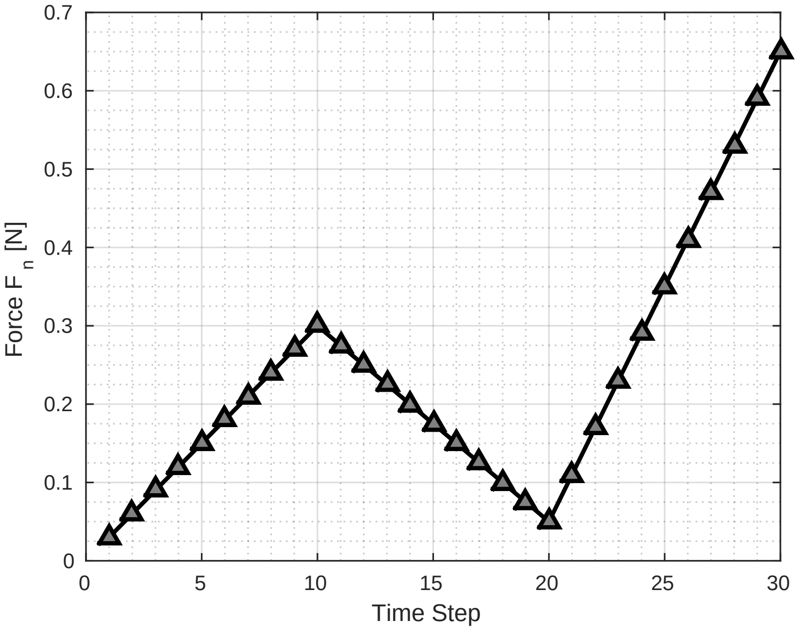
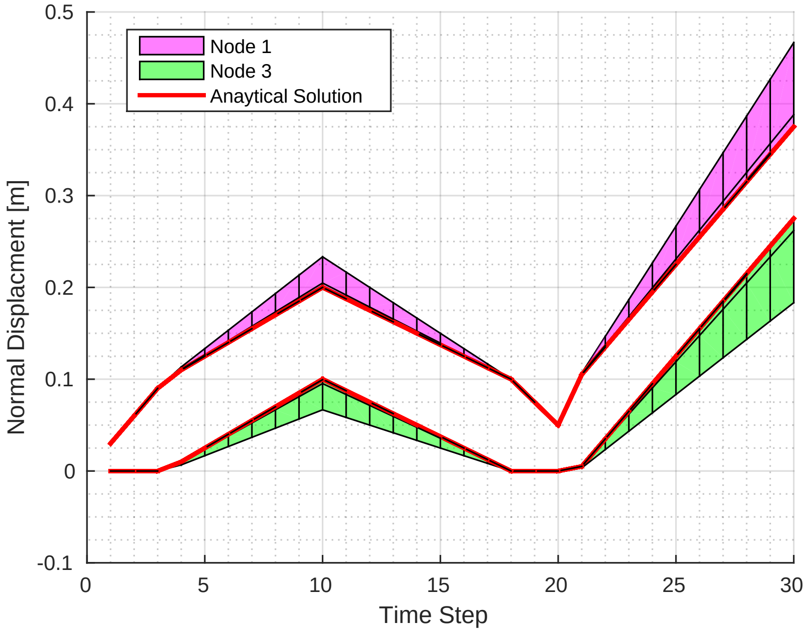


Figure 309.4: Cyclic normal load applied on two bar contact/interface/joint problem.



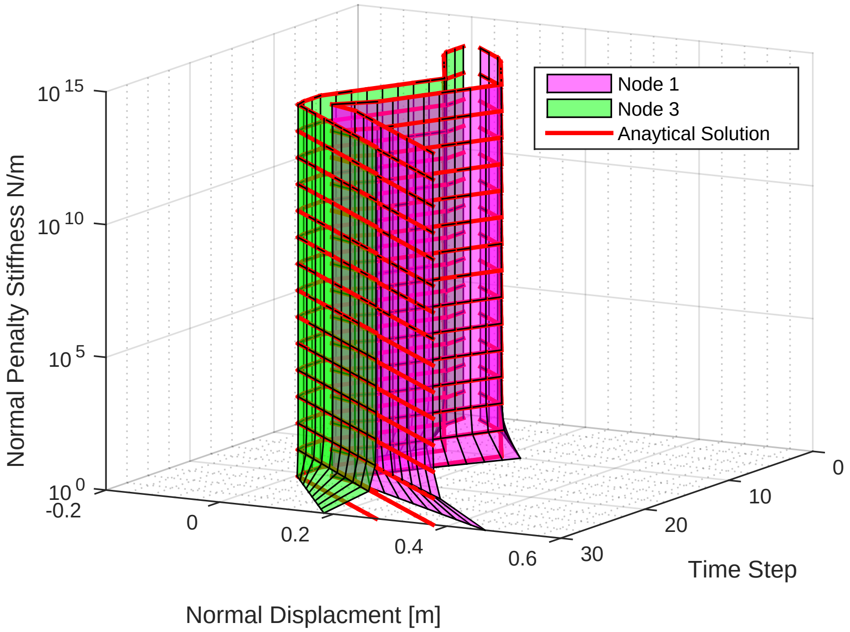


Figure 309.5: Displacements of Node 2 and Node 3 with change in normal penalty stiffness for $\delta_{in} = 0m$

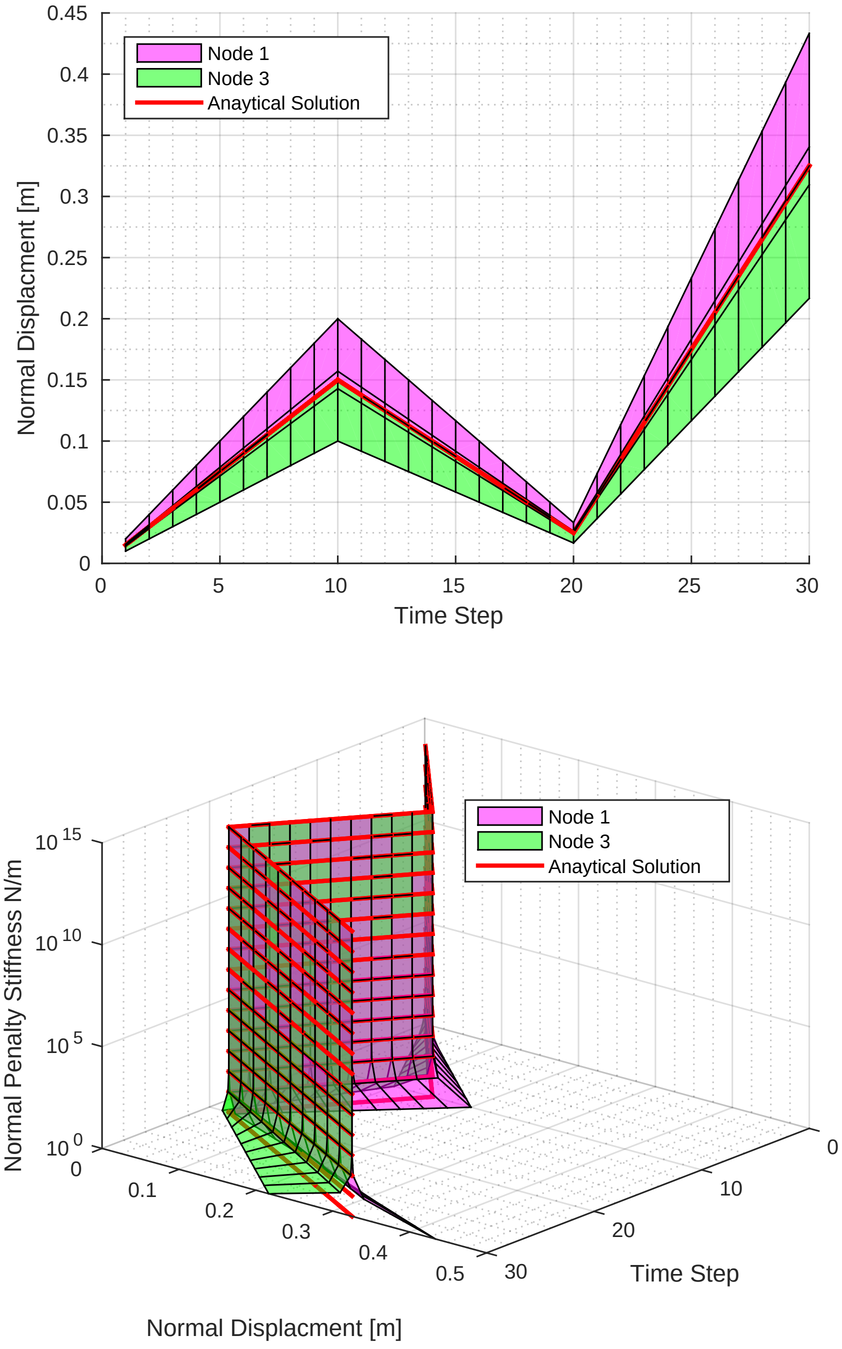


Figure 309.6: Displacements of Node 2 and Node 3 with change in normal penalty stiffness for $\delta_{in} = 0m$

309.2.2 Static Frictional Tangential Contact/Interface/Joint Verification

A simple 3-D truss example with Normal confinement in z-direction of $F_N = 0.5N$, friction coefficient $\mu = 0.2$ and shear loading of magnitude $F_s = 0.5N$ is considered to verify the tangential behaviour of contact/interface/joint element. Different cases as dicussed below are considered.

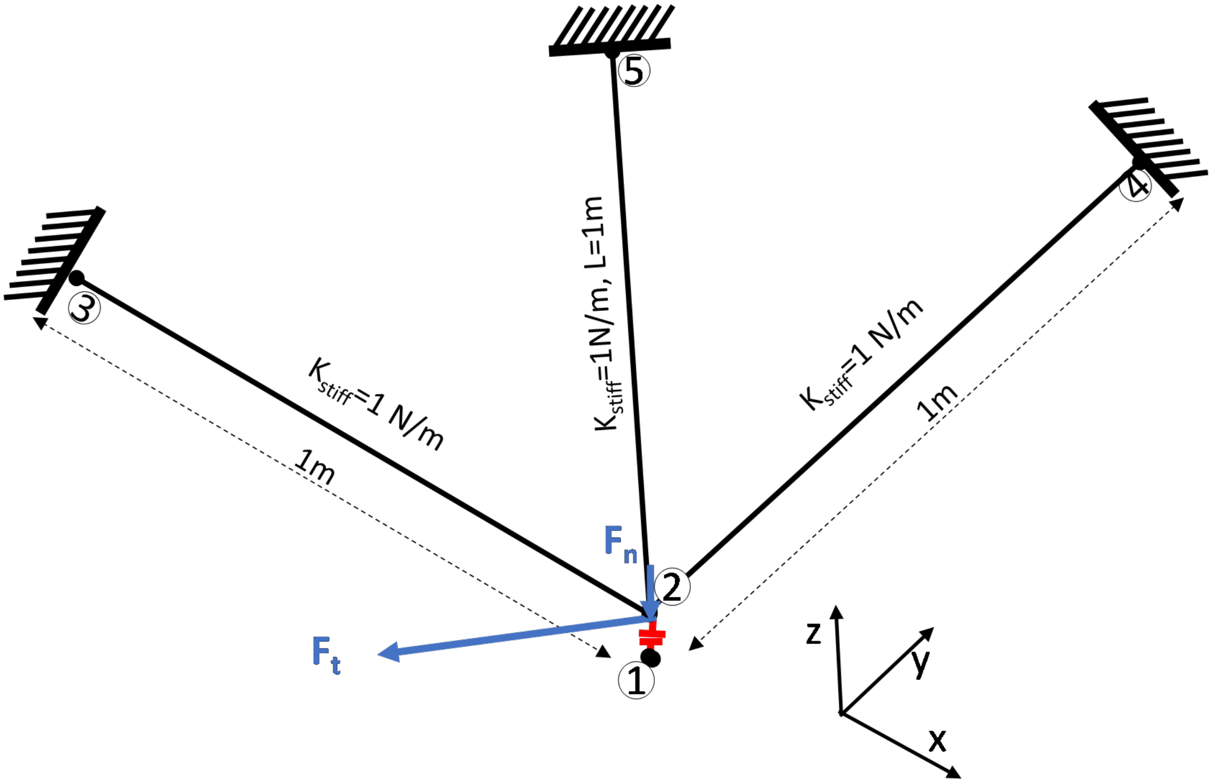


Figure 309.7: Illustration of 3-D three bar contact/interface/joint problem.

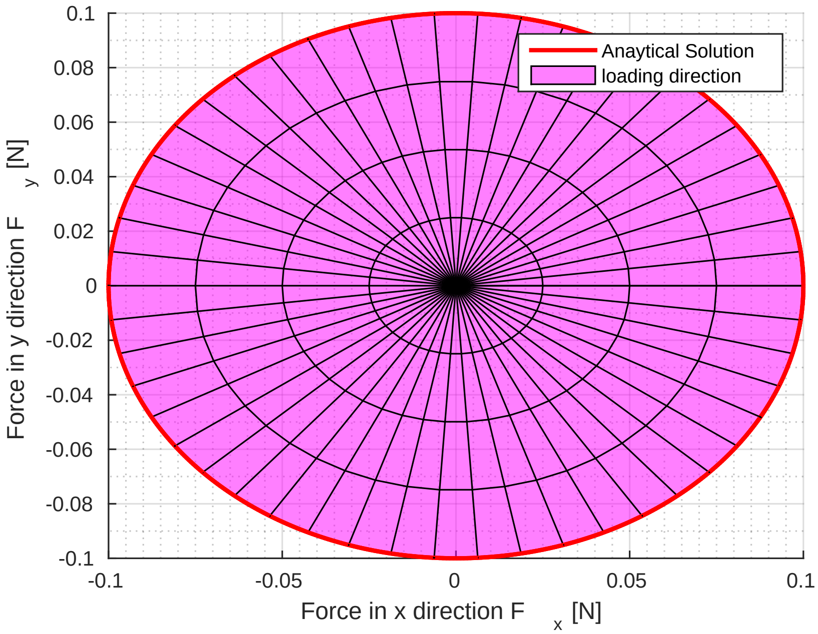
A snapshot of the properties of contact/interface/joint element os shown below. The Real-ESSI model fei/DSL files for this example can be downloaded HERE.

```
add element #4 type FrictionalPenaltyContact with nodes (1,2)
 normal_stiffness =1e10*N/m
 tangential_stiffness = 1e10*Pa*m
 normal_damping = 0*kN/m*s
 tangential_damping = 0*kN/m*s
 friction_ratio = 0.2
 contact_plane_vector = (0,0,1);
```

309.2.2.1 Case 1: Verfication of the yield surface for different loading angles with fixed normal confinement.

A Shear force of magnitude of $F_s = 0.5N$ was applied in 20 steps in different loading directions. The response of the contact/interface/joint element and the displacement of node 2 is shown in Figure 309.8 and Figure 309.9 respectively . It can be bserved that the contact/interface/joint element slips at magnitude of force $||F||- > (F_n = 0.5) * (\mu = 0.2) - > 0.1N$ for all loading angles.

In Figure 309.9, it can be observed that for the first 4 steps, there is no (zero) displacement for node 2 because of the stick case. When the load exceeds $0.1N$, slip occurs and node 2 starts to undergo deformation.



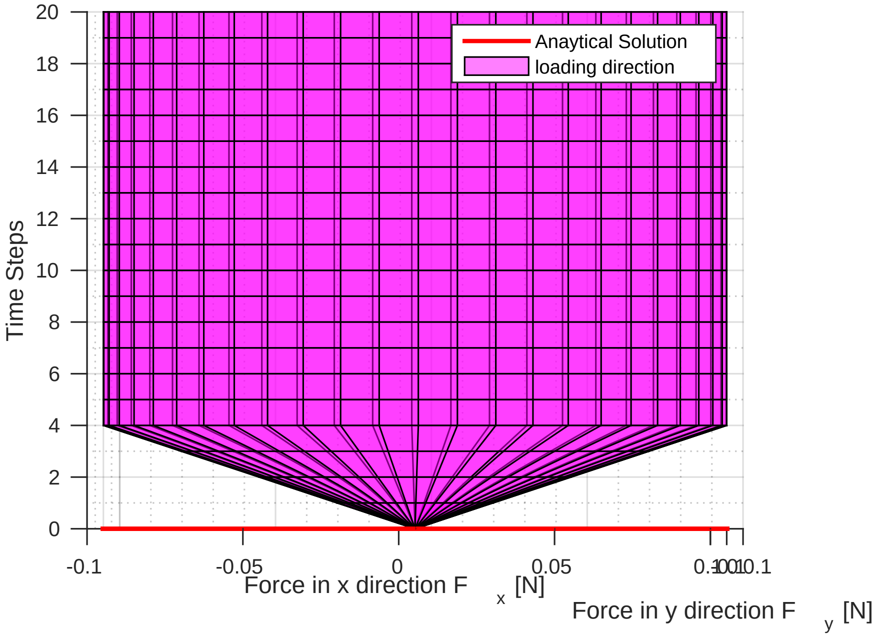
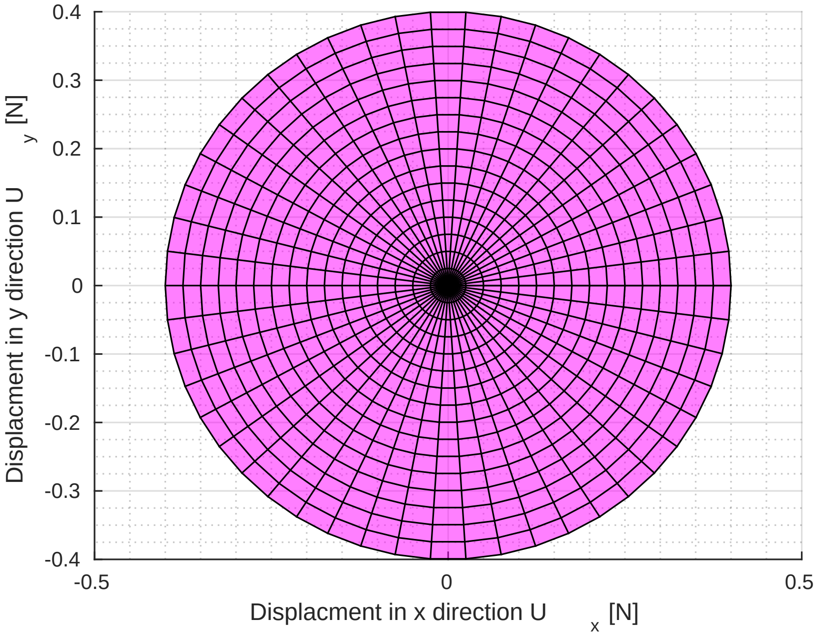


Figure 309.8: Response of contact/interface/joint element for different loading angles for confinement of $0.5N$ and coefficient of friction as 0.2

.



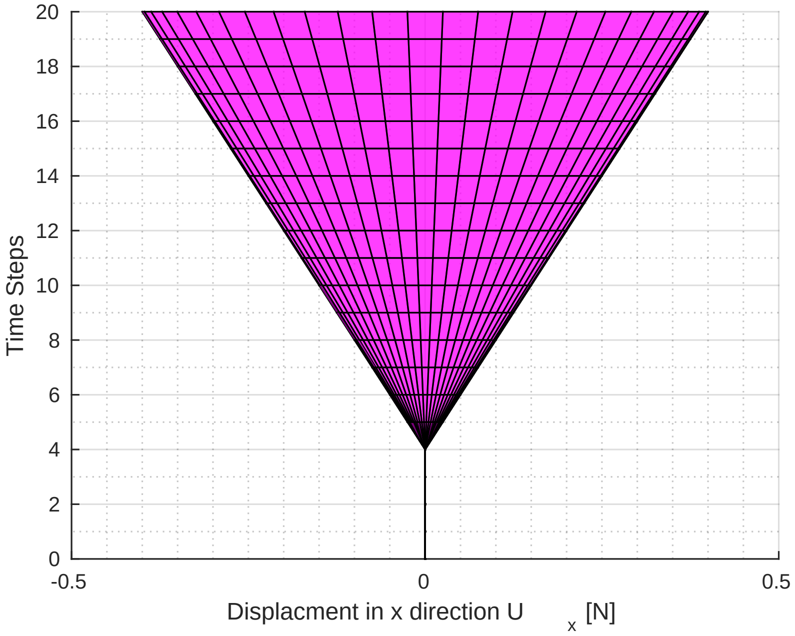


Figure 309.9: Displacement of Node No. 2 in x and y direction for different loading angles for confinement of $0.5N$ and coefficient of friction as 0.2

.

309.3 Verification of Static and Dynamic Contact/interface/Joint Element Modeling and Simulation

Solution verification of the contact/interface/joint element formulation and its implementation is presented in what follows. Analytical simple solutions for the frictional contact/interface/joint element are used to verify numerical solutions obtained by application of the developed model. The examples provided show the response of the contact element in several situations. Initially, the element is tested by connecting two nodes that each have *3dof*, subsequently, the element is even implemented to simulate a contact/interface/joint between two nodes that each have *7dof* and two nodes with different dofs: *3dof* for the first one and *7dof* for the second one.

The parameters used for the contact/interface/joint element are listed in Table 309.1.

Parameter	Value
C_N [kN/m]	10420
$\overline{v}_{max}$ [m]	0.001
K_T [kN/m]	1e7
μ [-]	0.6

Table 309.1: Contact/interface/joint element parameters.

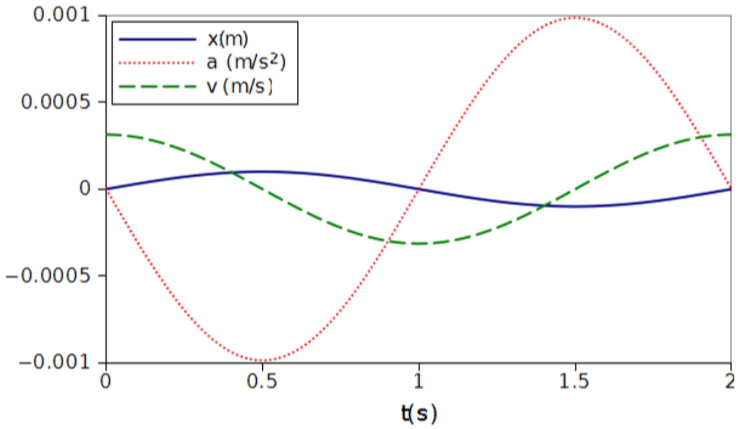


Figure 309.10: Input signal: time history of displacement, acceleration and velocity.

309.3.1 Truss Examples

The first example (Figure 309.11) represents five nodes: `1,2,3,4,5`. Nodes `2,3,4,5` are connected by three truss elements and a contact/interface/joint element links node `1` and `2`. All degrees of freedom of nodes `1,3,4,5` are fixed, whereas a sine wave displacement time-history (Figure 309.10) is applied to node `2` along `x` direction and the normal force acting within the contact/interface/joint element is recorded. The results, represented in Figure 309.12, show the normal response of this new contact/interface/joint element. As the timestep decreases, the force-displacement curves tends to be similar to the one represented by (??).

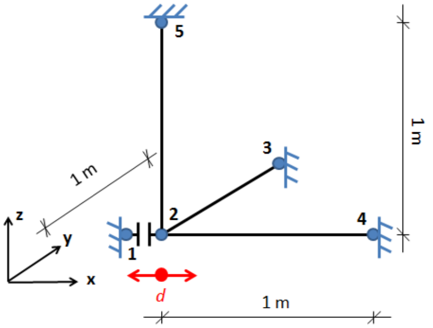


Figure 309.11: System composed of one contact/interface/joint element and three truss elements. A sine wave displacement time-history is applied to node (2).

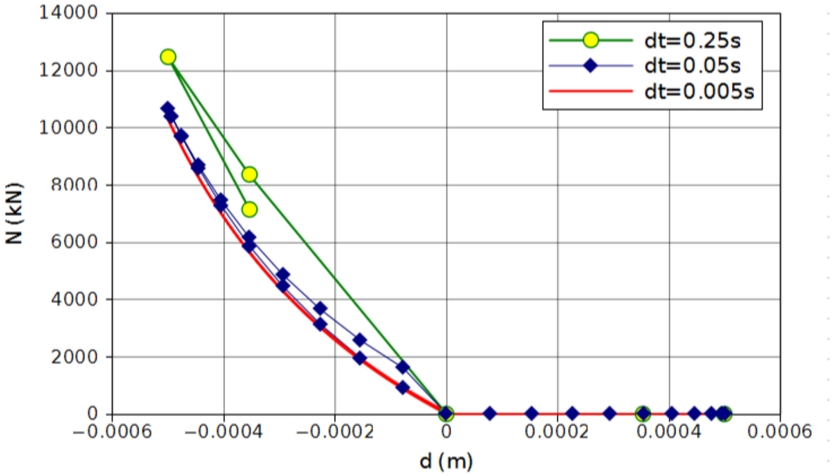


Figure 309.12: Normal force vs normal-relative displacement in contact/interface/joint element.

The second example (Figure 309.13) shows the tangential response of this new contact/interface/joint element. The geometry of the problem is the same as Figure 309.11, but an axial force (p = 1140 kN), constant in time, and a sine wave time-history displacement are applied to node 2. The results, represented in Figure 309.14, show that the response is not dependent on the timestep used for the analysis. Due to the elastic-perfectly-plastic behavior, associated with Mohr-Coulomb yield criteria, the maximum shear force (t_{max}) that the contact/interface/joint element can sustain is 684 kN , equal to $t_{max} = \mu \cdot p = 0.6 \cdot 1140 kN$.

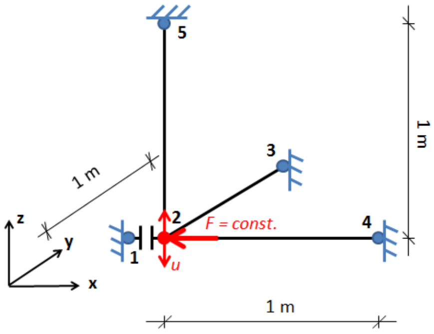


Figure 309.13: Same system used in Figure 309.11. A normal force constant in time (F = 1140 kN) and a sine wave displacement time history are applied to node (2).

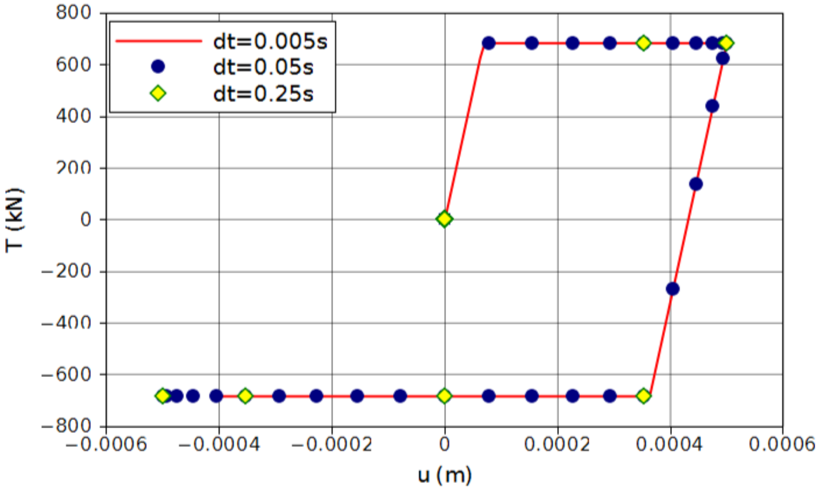


Figure 309.14: p = 1140 kN. Transversal force in the contact/interface/joint element vs transversal-relative displacement

309.3.2 Single Brick Element Examples

The contact/interface/joint element is now used to connect each node (`5,6,7,8`) of the bottom face of an eight node hexahedral element (`hex8`) to the corresponding node attached to the ground (`1,2,3,4`).

A linear elastic constitutive model is used for the eight node brick and the parameters are listed in Table 309.2.

Parameter	Value
E [kPa]	1.5e10
ν [-]	0.0

Table 309.2: Brick Element Parameters

In the first example (Figure 309.15) the time-history displacement, shown in Figure 309.10, is applied in the vertical direction to each node of the top surface of the brick element (`9,10,11,12`) and the time-history normal force induced in each contact element is represented in Figure 309.16. It is worth noting that the normal force is positive if the displacement of the top surface is downward, whereas it is zero if the detachement occurs caused by the upward movement.

Through the second example, shown in Figure 309.17, the transversal response of the contact/interface/joint elements is highlighted. The vertical normal force ($F_v = 50kN$), constant in time, and a horizontal time-history displacment are applied to each node of the top surface and the transversal response is shown in Figure 309.18. As stated in section 309.3.1, the maximum shear force that the contact element can sustain is is 30 kN , equal to $t_{max} = \mu \cdot p = 0.6 \cdot 50kN$.

The third example is focused on the transversal response of the contact/interface/joint element under variable normal forces. In fact, a sine wave time-history horizontal displacement and a vertical force are applied to each node of the top surface of the brick. The normal force is variable in time according to the factor ($Fact(t)$) shown in Figure 309.20 and the vertical force is computed as $F_v(t) = Fact(t) \cdot F_{v,max}$, and $F_{v,max}$ equal to 50 kN. The response of the contact element, shown in Figure 309.21, is independent of the timestep used for the analysis enphasizing the correct numerical implementation.

309.3.3 Double Brick Element Examples

Few other examples are produced taking in consideration two brick elements. The constitutive model used for these two brick elements is linear elastic with the same parameters listed in Table 309.2. Vertical and horizontal time-history displacement are applied to the top surface, shown in Figure 309.22 and Figure 309.23, and variable vertical forces are considered in the example represented in Figure 309.24. The results are the same shown in Figure 309.16, Figure 309.18 and Figure 309.19.

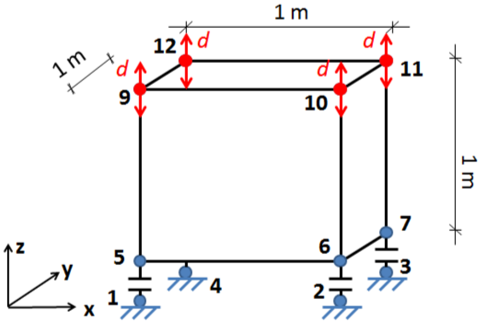


Figure 309.15: Eight-node brick element over four contact/interface/joint elements. A sine wave time-history vertical displacement applied to each node of the top surface.

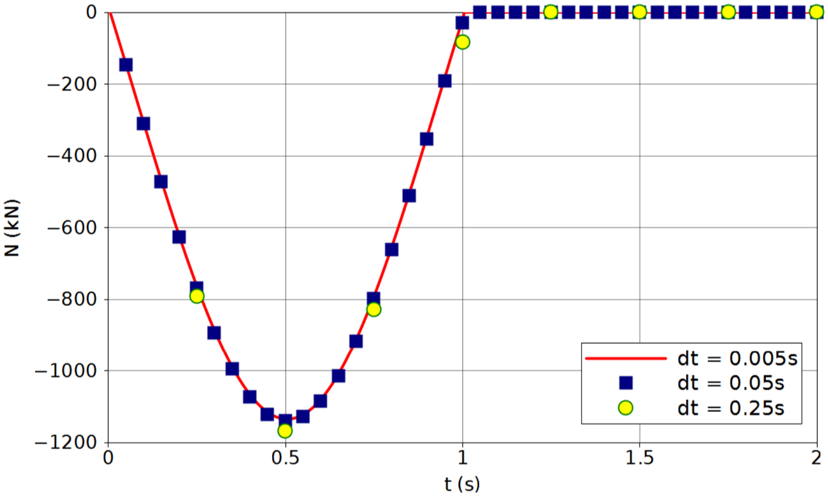


Figure 309.16: Normal force vs time in each contact element.

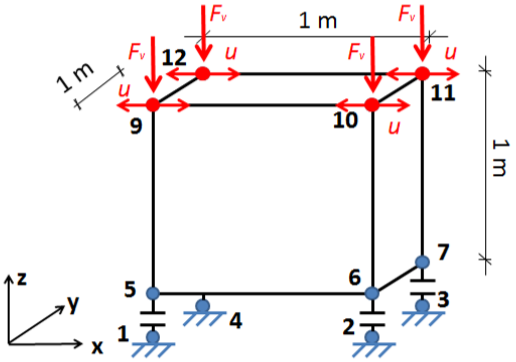


Figure 309.17: Eight-node brick element over four contact/interface/joint elements. F_v= 50 kN and a sine wave time-history horizontal displacement applied to each node of the top surface.

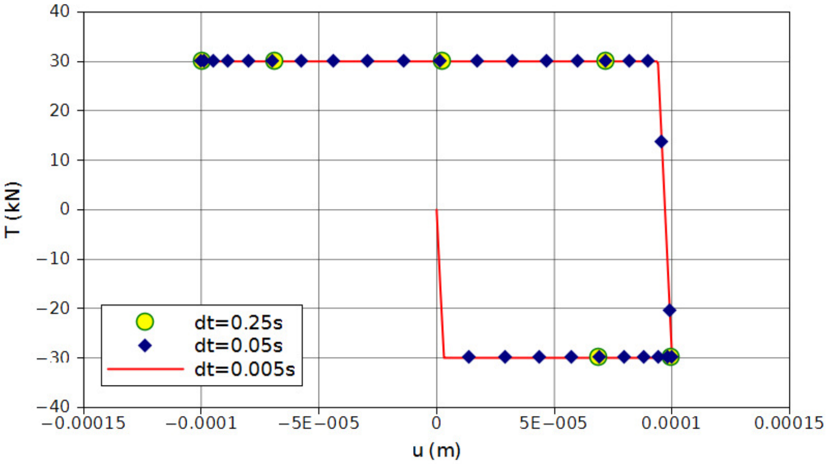


Figure 309.18: Transversal force vs transversal relative-displacement in the contact element with normal force F equal to 50 kN.

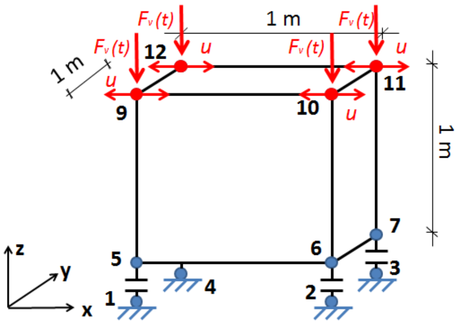


Figure 309.19: Eight-node brick element over four contact/interface/joint elements. Variable vertical force $(F_v(t))$ and a sine wave time-history horizontal displacement applied to each node of the top surface.

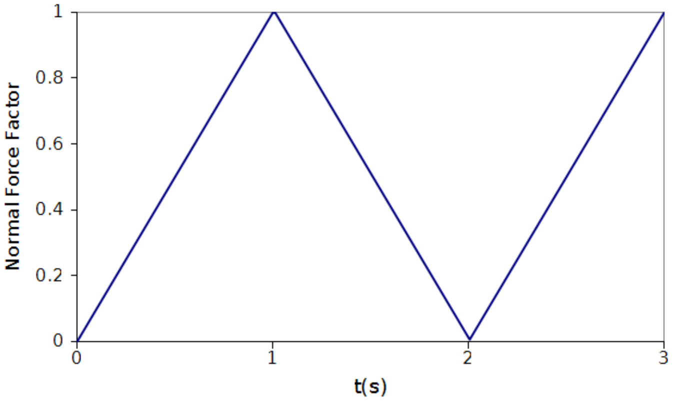


Figure 309.20: Time-history of the normal force factor $Fact(t)$.

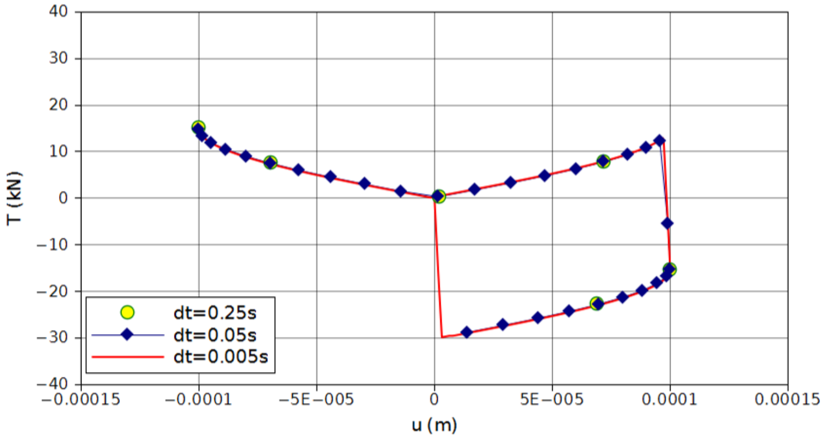


Figure 309.21: Variable normal force. Transversal force vs transversal relative-displacement in each contact element.

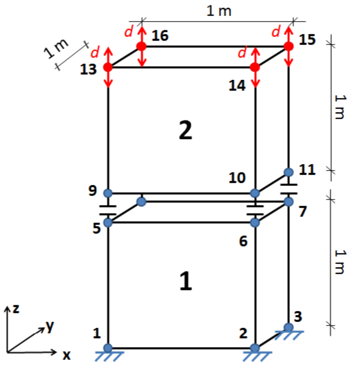


Figure 309.22: Two eight-node brick elements connected by four contact/interface/joint elements. Vertical time-history displacement applied to the nodes of the top surface.

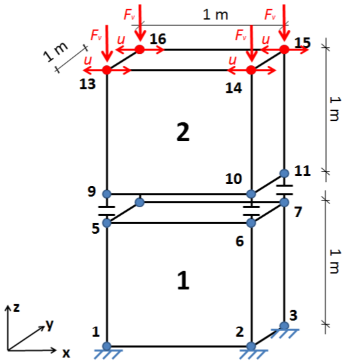


Figure 309.23: Two eight-node brick elements connected by four contact/interface/joint elements. Vertical force, equal to 50 kN and constant in time, and horizontal time-history displacement applied to the nodes of the top surface.

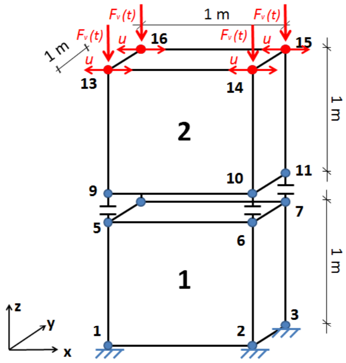


Figure 309.24: Two eight-node brick elements connected by four contact/interface/joint elements. Vertical force, variable in time, and horizontal time-history displacement applied to the nodes of the top surface.

309.4 Verification of Static and Dynamic Coupled (Saturated) Contact/Interface/Joint Element Modeling and Simulation

309.4.1 Dry u-p-U Contact/Interface/Joint

A single brick *u*-*p*-*U* finite element is used to model an oedometric compression shown in Figure 309.25. Horizontal displacements and pore pressure are fixed in each node in order to guarantee the one-dimensional and dry conditions. Since the ground is modeled as an undeformable and impermeable layer (`1,2,3,4`), vertical soil and fluid displacements are fixed. The time-history displacement, shown in Figure 309.10, is applied in vertical direction to each node of the top surface (`9,10,11,12`).

The time-history normal force induced in each contact/interface/joint element is represented in Figure 309.16 and compared with the one obtained with the dry brick element with *u* formulation, shown in Figure 309.15. It is worth noting that the normal force patterns are perfectly overlapped: this is due to the fact that excess pore pressure is fixed to zero and the oedometric stiffness are the same in the two cases.

Parameter	Symbol	Value
Young's Modulus	E $[kPa]$	$1.5 \cdot 10^{10}$
Poisson ratio	ν [-]	0.0
Solid particle bulk modulus	K_s $[kPa]$	$3.6 \cdot 10^{7}$
Fluid bulk modulus	K_f $[kPa]$	$2.17 \cdot 10^{6}$
Solid density	ρ_s $[Mg/m^3]$	2.7
Fluid density	ρ_f $[Mg/m^3]$	1.0
Porosity	n [-]	$1.0 \cdot 10^{-8}$
Darcy permeability	K $[m/s]$	$1.0 \cdot 10^{3}$

Table 309.3: Soil parameters.

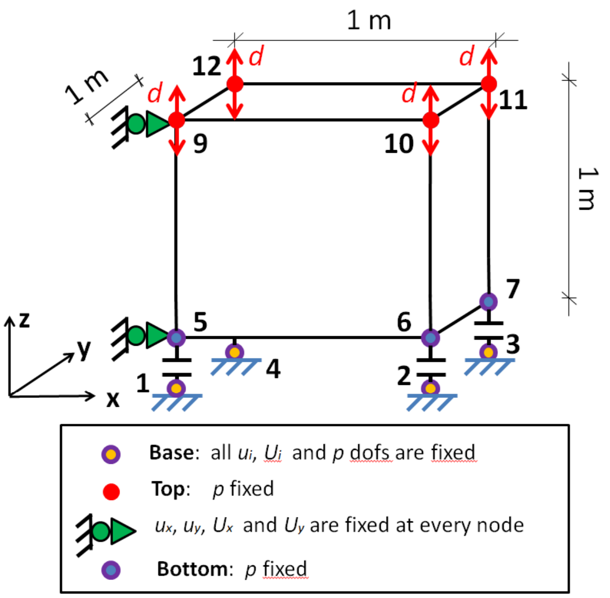


Figure 309.25: Single eight-node brick element. The nodes of the bottom surface are connected to the ground floor through contact/interface/joint elements. Vertical time-history displacement applied to the nodes of the top surface.

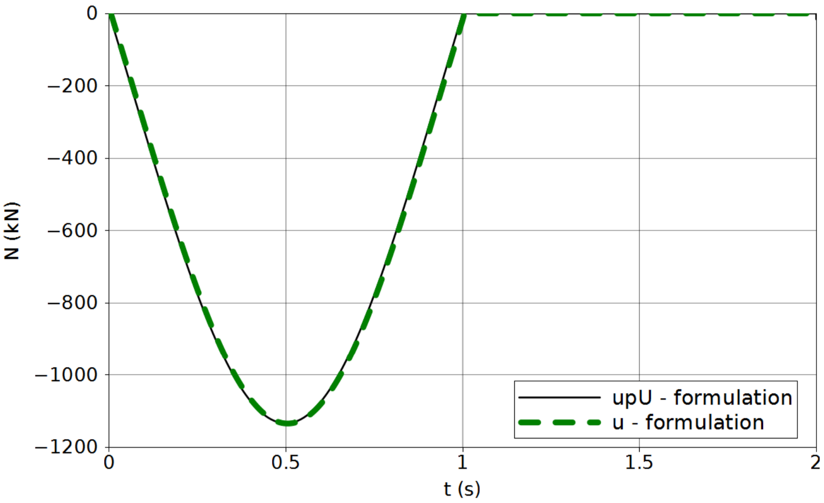


Figure 309.26: Normal force vs time in each contact/interface/joint element. *u-formulation* represents the results shown in Figure 309.16 whereas *upU-formulation* represents the results obtained through the model shown in Figure 309.25.

309.4.2 u-p-U Contact/Interface/Joint

A column of four brick *u-p-U* finite elements is used to model the horizontal layer. The height of the soil column is 1 m and the height of each element has dimensions $1m \cdot 1m \cdot 1m$, illustrated in Figure 309.27.

The following boundary conditions are applied to the model. As the bottom of the column is modeled as an undeformable and impermeable layer, both the solid and fluid displacements are fixed. The pore pressure is kept constant as zero at the top surface of the soil column because of the perfectly drained condition. In order to simulate the 1D compression problem, all the lateral movement of the solid and fluid phase are constrained so that the vertical displacement is the only non-zero displacement. A vertical time-history displacement, shown in Figure 309.10, is applied to the solid *dof* of the top surface nodes. In order to simulate the one dimensional compression, all the degrees of freedom at the same level are connected in a masterslave fashion.

The nodes of the bottom surface (`5,6,7,8`) are connected to the ground (`1,2,3,4`) through contact/interface/joint elements. Under the hypothesis of laminar flow, no cavitation and one-dimensional type of problem, the water has to fill up all the voids generated into the media while the displacement filed is acting on the top surface. This means that if the soil and the ground are separated because of the detachment of the contact/interface/joint element, the water has to fill the gap. Even the pore pressure at both sides of the gap has to assume the same value. Such boundary conditions can be introduced by adding masterslave between each node of the bottom surface (`5,6,7,8`) and the corresponding one belonging to the ground (`1,2,3,4`).

In this paragraph a parametric study is performed in order to examine the performance of this *u-p-U* contact/interface/joint element. The analyses consider several configuration of permeability and soil stiffness. In fact, three values of elastic modulus (E) are taken into consideration referring to a Stiff, Medium and Soft soil (called respectively `StS`, `MS` and `SoS`) and three values of the darcy permeability are similarly defined. All soil and contact/interface/joint parameters are listed respectively in Table 309.4 and Table 309.1.

This kind of excitation (a sine wave) applied at the top of the model is clearly composed of waves of all kinds of frequency, expetially at the initial strong change in acceleration and velocity for $t = 0s$. Due to this fact, a fairly dense mesh of 100 u-p-U brick finite elements of dimensions $1m \cdot 1m \cdot 1cm$ was chosen. Therefore, the time step Δt needs to be limited to $\Delta t = \Delta h/v$, where v is the highest wave velocity. In our case, the temporal integration involves 4000 steps of , which allows a maximum wave velocity of . The propagation velocity can be calculated by the following equation given by de Boer et al. (1993) and is equal to

$$v = \sqrt{\frac{n^2(1-\nu)E}{(1+\nu)(1-2\nu)[n^2(1-n)\rho_s + (1-n)^2 n\rho_f]}} \tag{309.1}$$

The artificial oscillation are diminished by introducing some numerical damping into the analysis through $\alpha = 0.6$ and $\beta = 0.3025$ for the Newmark integrator.

Figure 309.27: Four eight-node brick elements. The nodes of the bottom surface are connected to the ground floor through contact/interface/joint elements. Vertical time-history displacement applied to the nodes of the top surface.

The specific permeability, k and the time needed for completion of the 1D consolidation process, t, can be estimated using the darcy permeability K through (309.2) and (309.3) respectively, where ρ_w is the mass density of the fluid (water), g is the acceleration of gravity equal to $9.81 m/s^2$, H is the thickness of the soil layer and E_{oed} is the one dimensional soil stiffness:

$$k = \frac{K}{\rho_w \cdot g} \tag{309.2}$$

$$t = \frac{H^2}{c_v} = \frac{H^2 \cdot \rho_w \cdot g}{K \cdot E_{oed}} \tag{309.3}$$

The time t is computed and listed in Table 309.4.2 for each soil type condition. As can be seen, for the stiff soil (`StS`), the consolidation time is much lower than the loading acting on the top surface, therefore no excess pore pressure is developped into the soil and drained response is occurred.

STIFF SOIL - (`StS`)				
Parameter	Symbol	Value		
Young's Modulus	E $[kPa]$	$1.0 \cdot 10^{10}$	$1.0 \cdot 10^{10}$	$1.0 \cdot 10^{10}$
Poisson ratio	ν [-]	0.0	0.0	0.0
Solid particle bulk modulus	K_s $[kPa]$	$3.6 \cdot 10^7$	$3.6 \cdot 10^7$	$3.6 \cdot 10^7$
Fluid bulk modulus	K_f $[kPa]$	$2.17 \cdot 10^6$	$2.17 \cdot 10^6$	$2.17 \cdot 10^6$
Solid density	ρ_s $[Mg/m^3]$	2.7	2.7	2.7
Fluid density	ρ_f $[Mg/m^3]$	1.0	1.0	1.0
Porosity	n [-]	0.46	0.46	0.46
Darcy permeability	K $[m/s]$	$1.0 \cdot 10^{-3}$	$1.0 \cdot 10^{-5}$	$1.0 \cdot 10^{-7}$
MEDIUM SOIL - (`MS`)				
Young's Modulus	E $[kPa]$	$1.0 \cdot 10^{7}$	$1.0 \cdot 10^{7}$	$1.0 \cdot 10^{7}$
Poisson ratio	ν [-]	0.0	0.0	0.0
Solid particle bulk modulus	K_s $[kPa]$	$3.6 \cdot 10^7$	$3.6 \cdot 10^7$	$3.6 \cdot 10^7$
Fluid bulk modulus	K_f $[kPa]$	$2.17 \cdot 10^6$	$2.17 \cdot 10^6$	$2.17 \cdot 10^6$
Solid density	ρ_s $[Mg/m^3]$	2.7	2.7	2.7
Fluid density	ρ_f $[Mg/m^3]$	1.0	1.0	1.0
Porosity	n [-]	0.46	0.46	0.46
Darcy permeability	K $[m/s]$	$1.0 \cdot 10^{-3}$	$1.0 \cdot 10^{-5}$	$1.0 \cdot 10^{-7}$
SOFT SOIL - (`SoS`)				
Young's Modulus	E $[kPa]$	$1.0 \cdot 10^{4}$	$1.0 \cdot 10^{4}$	$1.0 \cdot 10^{4}$
Poisson ratio	ν [-]	0.0	0.0	0.0
Solid particle bulk modulus	K_s $[kPa]$	$3.6 \cdot 10^7$	$3.6 \cdot 10^7$	$3.6 \cdot 10^7$
Fluid bulk modulus	K_f $[kPa]$	$2.17 \cdot 10^6$	$2.17 \cdot 10^6$	$2.17 \cdot 10^6$
Solid density	ρ_s $[Mg/m^3]$	2.7	2.7	2.7
Fluid density	ρ_f $[Mg/m^3]$	1.0	1.0	1.0
Porosity	n [-]	0.46	0.46	0.46
Darcy permeability	K $[m/s]$	$1.0 \cdot 10^{-3}$	$1.0 \cdot 10^{-5}$	$1.0 \cdot 10^{-7}$

Table 309.4: Stiff, Medium and Soft soil parameters (`StS`, `MS` and `SoS`).

Soil type	Symbol	Value		
	K $[m/s]$	$1.0 \cdot 10^{-3}$	$1.0 \cdot 10^{-5}$	$1.0 \cdot 10^{-7}$
Stiff soil (`StS`)	t $[s]$	$1.0 \cdot 10^{-6}$	$1.0 \cdot 10^{-4}$	$1.0 \cdot 10^{-2}$
Medium-Stiff soil (`MS`)	t $[s]$	$1.0 \cdot 10^{-3}$	$1.0 \cdot 10^{-1}$	$1.0 \cdot 10^{1}$
Soft soil (`SoS`)	t $[s]$	$1.0 \cdot 10^{0}$	$1.0 \cdot 10^{2}$	$1.0 \cdot 10^{4}$

Table 309.5: The time needed for completion of the 1D consolidation process, t, estimated thrugh the darcy permeability K for each soil type condition.

309.5 Verification of Static, Isolator Element Modeling and Simulation

Chapter 310

Verification and Validation for Coupled, Porous Solid – Pore Fluid Problems

(2000-2003-2007-2009-2010-2016-2017-2020-2021-)

(In collaboration with Prof. Zhao Cheng, Dr. Panagiota Tasiopoulou, Ms. Fatemah Behbehani, Prof. Han Yang and Mr. Yusheng Yang)

310.1 Chapter Summary and Highlights

310.2 Introduction

presented here are verification examples for u-p-U formulation. Examples include:

1. Drilling of a borehole
2. The case of a spherical cavity
3. Consolidation of a soil layer
4. Line injection of a fluid in a reservoir
5. Shock wave propagation
6. Vertical Consolidation of a soil layer by Coussy (2004)
7. One dimensional shock wave propagation with a step displacement boundary condition by Gajo and Mongiovi (1995)
8. One dimensional shock wave propagation with step loading at the surface by de Boer et al. (1993)
9. One dimensional shock wave propagation with a step velocity boundary condition by Hiremath et al. (1988)

310.3 Drilling of a well

310.3.1 The Problem

Let us consider an infinite half space domain composed of an isotropic, homogeneous and saturated thermoporoelastic material. At its reference state, it is assumed that the temperature, fluid pressure and stress fields are uniform and equal respectively, to T_0, p_0 and $\boldsymbol{\sigma}^0 = \sigma^0\mathbf{1}$(with $\sigma^0 < 0$). At time $=$ 0, an infinite cylinder of radius r_0 is instantaneously drilled parallel to the vertical axis Oz. It is filled with a fluid of the same nature as that saturating the porous medium but at a different pressure and temperature at the values of p_1 and T_1 respectively. The interface $r = r_0$ between the well and the porous medium is assumed to be in thermodynamic equilibrium.

In cylindrical coordinates $(r,\ \theta,\ z)$, the boundary conditions can be summarized as follows (see Fig.310.1):

$$
\begin{aligned}
t \leq 0 \rightarrow \qquad & \boldsymbol{\sigma}^0 = \sigma^0\mathbf{1} \qquad p(r) = p_0 \qquad T(r) = T_0 && (310.1)\\
t > 0 \rightarrow \qquad & \sigma_{rr}(r_0) = -p_1 \qquad \sigma_{r\theta}(r_0) = \sigma_{rz}(r_0) = 0\\
\sigma_{rr}(r \rightarrow \infty) \rightarrow \sigma^0 \qquad & \sigma_{r\theta}(r \rightarrow \infty) = \sigma_{rz}(r \rightarrow \infty) \rightarrow 0\\
& p(r_0, t) = p_1 \qquad p(r \rightarrow \infty) \rightarrow p_0\\
& T(r_0) = T_1 \qquad T(r \rightarrow \infty) \rightarrow T_0 && (310.2)
\end{aligned}
$$

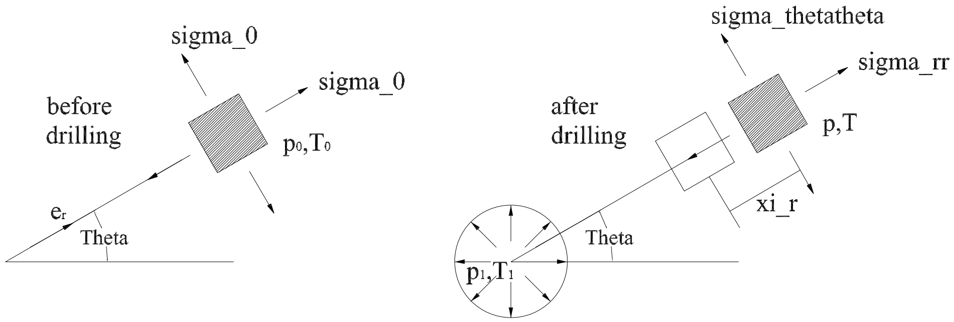


Figure 310.1: Boundary Conditions for Drilling of a Borehole

310.3.2 Analytical Solution

Since the well is assumed to be infinite long in its vertical axis Oz, the analysis is performed under plane strain hypothesis($\epsilon_{zz} = 0$). Therefore,

$$\boldsymbol{\xi} = \xi^r(r)e_r \qquad p = p(r) \qquad T = T(r) \tag{310.3}$$

in which ξ^r is the radial displacement. In cylindrical coordinates, Eqn. 310.48 yields

$$\epsilon_{rr} = \frac{\partial \xi^r}{\partial r} \qquad \epsilon_{\theta\theta} = \frac{\xi^r}{r} \qquad other \qquad \epsilon_{ij} = 0 \tag{310.4}$$

Based on the constitutive equations from Coussy (1995), it follows that

$$\sigma_{rr} = \sigma^0 + \lambda_0(\frac{\partial \xi^r}{\partial r} + \frac{\xi^r}{r}) + 2\mu\frac{\partial \xi^r}{\partial r} - b(p - p_0) - 3\alpha K_0(T - T_0) \tag{310.5}$$

$$\sigma_{\theta\theta} = \sigma^0 + \lambda_0(\frac{\partial \xi^r}{\partial r} + \frac{\xi^r}{r}) + 2\mu\frac{\xi^r}{\partial r} - b(p - p_0) - 3\alpha K_0(T - T_0) \tag{310.6}$$

$$\sigma_{zz} = \sigma^0 + \lambda_0(\frac{\partial \xi^r}{\partial r} + \frac{\xi^r}{r}) - b(p - p_0) - 3\alpha K_0(T - T_0) \tag{310.7}$$

$$other \qquad \sigma_{ij} = 0 \tag{310.8}$$

Finally combined with the Eqns. 310.5-310.8, it yields the near field or long-term solution (Coussy, 1995)

$$\xi^r = \frac{\sigma^0 + p_1}{2\mu}\frac{r_0{}^2}{r} + \frac{r_0[b(p_1 - p_0) + 3\alpha_0 K_0(T_1 - T_0)]}{2(\lambda_0 + 2\mu)}(\frac{r}{r_0} - \frac{r_0}{r}) \tag{310.9}$$

$$\sigma_{rr} = -p_1\frac{r_0{}^2}{r^2} - \{\sigma_0 - \frac{\mu[b(p_1 - p_0) + 3\alpha K_0(T_1 - T_0)]}{\lambda_0 + 2\mu}\}(1 - \frac{r_0{}^2}{r^2}) \tag{310.10}$$

$$\sigma_{\theta\theta} = (2\sigma_0 + p_1)\frac{r_0{}^2}{r^2} - \frac{\mu[b(p_1 - p_0) + 3\alpha K_0(T_1 - T_0)]}{\lambda_0 + 2\mu}(1 + \frac{r_0{}^2}{r^2}) \tag{310.11}$$

$$\sigma_{zz} = \sigma^0 - \frac{2\mu}{\lambda_0 + 2\mu}[b(p_1 - p_0) + 3\alpha K_0(T_1 - T_0)] \tag{310.12}$$

And the diffusion process can be achieved if that the time are large enough with respect to the characteristics diffusion time relative to point r. When the boundary conditions for $r = r_0$ in fluid pressure and temperature which are $p = p_1$ and $T = T_1$ apply for the whole model, the following equations correspond to the undrained solution of the instantaneous drilling of a borehole in an infinite elastic medium.

$$\xi^r = \frac{\sigma^0 + p_1}{2\mu}\frac{r_0{}^2}{r} \qquad \sigma_{rr} = \sigma^0 - (\sigma_0 + p_1)\frac{r_0{}^2}{r^2}$$
$$\sigma_{\theta\theta} = \sigma^0 + (\sigma^0 + p_1)\frac{r_0{}^2}{r^2} \qquad \sigma_{zz} = \sigma_0 \tag{310.13}$$

310.3.3 Discussion of the Results

As the problem is Axisymmetric, we construct the model as a quarter of a donut. The inside diameter of the donut is 10 cm and the outside diameter is 1 m. To accommodate both the plain strain hypothesis and the geometry of the element for finite element, the thickness of the model is chosen to be 5 cm. The final mesh is generated as Fig.310.2. And the boundary conditions is as follows: As a consequence of plain strain problem, all the movements for solid and fluid in vertical direction Oz are suppressed; the solid and fluid displacement for the nodes along the X axis and Y axis are fixed in Y and X direction respectively for the reason of axisymmetry; the nodes along the outside perimeter are fixed in the solid and fluid displacement with the assumption of infinite medium. the pressure is translated into nodal forces

Parameter	Symbol	Value	Units
Poisson Ratio	ν	0.2	-
Young's Modulus	E	1.2E+6	kN/m^2
Solid Bulk Modulus	K_s	3.6E+7	kN/m^2
Fluid Bulk Modulus	K_f	1.0E+17	kN/m^2
Solid Density	ρ_s	2.7	ton/m^3
Fluid Density	ρ_f	1.0	ton/m^3
Porosity	n	0.4	-

Table 310.1: Material Properties used to study borehole problem

and applied on the nodes along the inside perimeter. For simplicity, the hydrostatic stress σ_0 is equal to zero and with the assumption of thermodynamic equilibrium through the process, the temperature factor can be neglected. Also the initial fluid pressure p_0 is set to be 0 kPa. The analytical solution is studied below using the following set of parameters shown in Table 310.1.

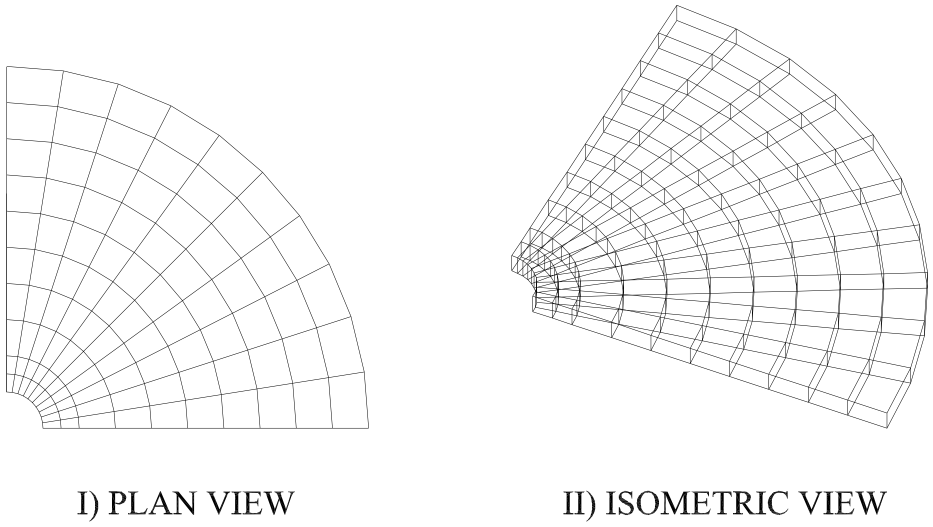


Figure 310.2: The mesh generation for the study of borehole problem

In the analysis, ten loading cases for final fluid pressure from 10 kPa to 100 kPa are studied. And by manipulating the permeability, it is possible to investigate both the drained behavior and undrained behavior. For the drained behavior, we choose the permeability as $k = 3.6 \times 10^{-4} m/s$, which is a typical value for sand, the comparison between the close solution and experimental result is shown in Fig.310.3. From the results, we can see that along the inside perimeter, the close solution and experimental result provide very good agreement to each other. But as the increase of the radius, we can see the analytical solution is getting more and more distant from the experimental results. In another word, the analytical solution can be interpreted as that with the increase from the loading surface, the radial displacement is larger. This is unreasonable in the point of view in soil mechanics. While the experimental result show the effect that with the increase of the radial distance, the radial displacement is decreasing.

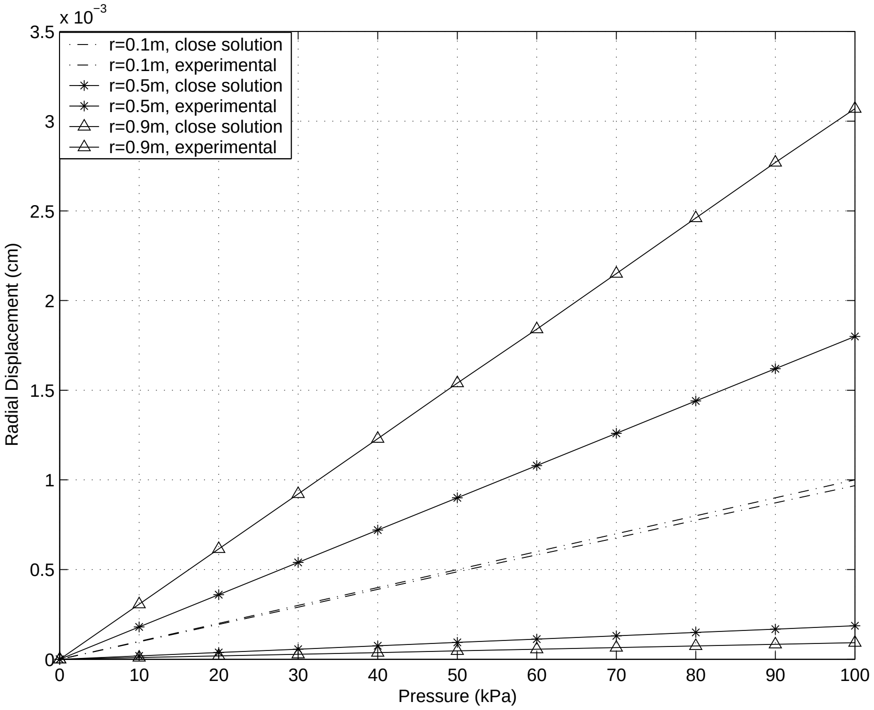


Figure 310.3: The comparison of radial solid displacement between analytical solution and experimental result for drained behavior

For the undrained behavior, the permeability of $k = 3.6 \times 10^{-8} m/s$ is selected as a representative value for typical clayey soil. The comparison between the close solution and experimental result is provided as well. From the Fig.310.4 we can see that, the analytical solution is linearly away from the experimental result by a ratio of approximately 1.6. It should also be noticed that the close solution of the drained and undrained behavior for the nodes along the inside perimeter are exactly the same, which is contradictory to the definition of drained and undrained behavior. For the drained behavior, as the water easily dissipate from the soil body, the problem can be treated with the knowledge of continuum mechanics using the parameters of the solid skeleton. While for the undrained behavior, with the involvement of the pore water, the elastic parameters for the mixture should be different, so the response will not be the same as well. As a result of this, the experimental results give a more reasonable conclusion.

As for the drained behavior, the fluid totally flows out of the soil body and all excessive pore pressure

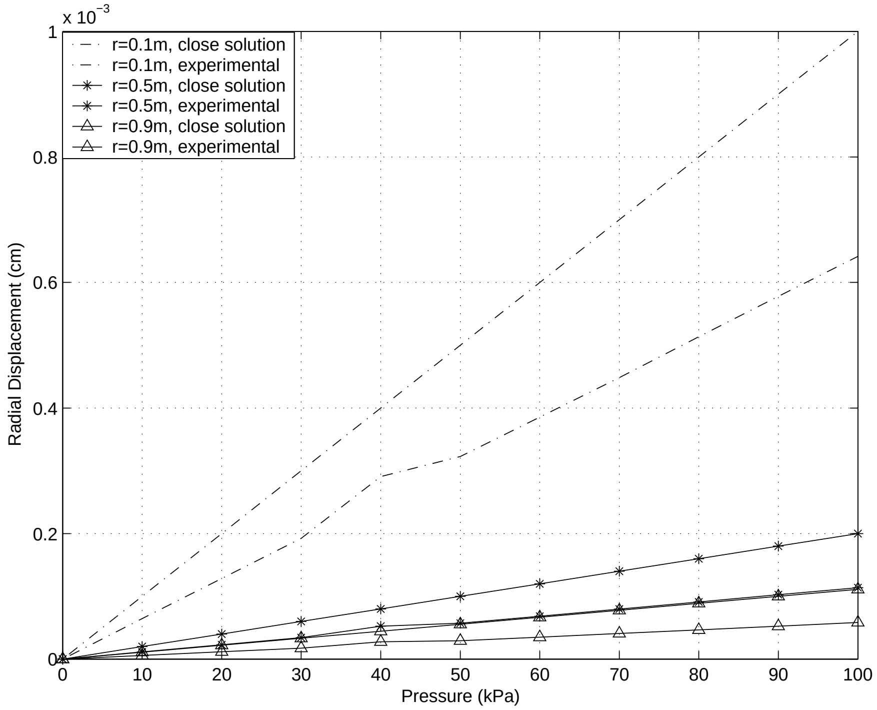


Figure 310.4: The comparison of radial solid displacement between analytical solution and experimental result for undrained behavior

dissipates, there is small coupling between the solid and fluid phase. We can use the continuum mechanics to treat this problem. Here introduces a problem of an infinite cylindrical tube, with the inner radius R_1 and outer radius R_0, subjected to an internal pressure P_1 and an external pressure P_2. The displacement field as follows (S.Timoshenko and D.H.Young, 1940):

$$\xi^r = \frac{{R_1}^2 P_1}{2({R_0}^2 - {R_1}^2)}\left(\frac{r}{\lambda + \mu} + \frac{{R_0}^2}{\mu r}\right) \tag{310.14}$$

With $P_0 = 0$ and take the limit of $R_0 \to \infty$, we can obtain the following equation:

$$\xi^r = \frac{P_1}{2\mu}\frac{{R_1}^2}{r} \tag{310.15}$$

which is identical to Eq.310.10. Also to minimize the effect of infinite boundary, we introduce the result from another model which is exactly the same as the previous one besides the expansion of the outer radius to 30m. At the final fluid pressure of 50 kPa, the results are shown in Fig.310.5. From the plot we can make a conclusion that the undrained analytical solution from Coussy (1995) is actually the drained solution and the undrained solution still needs to be investigated.

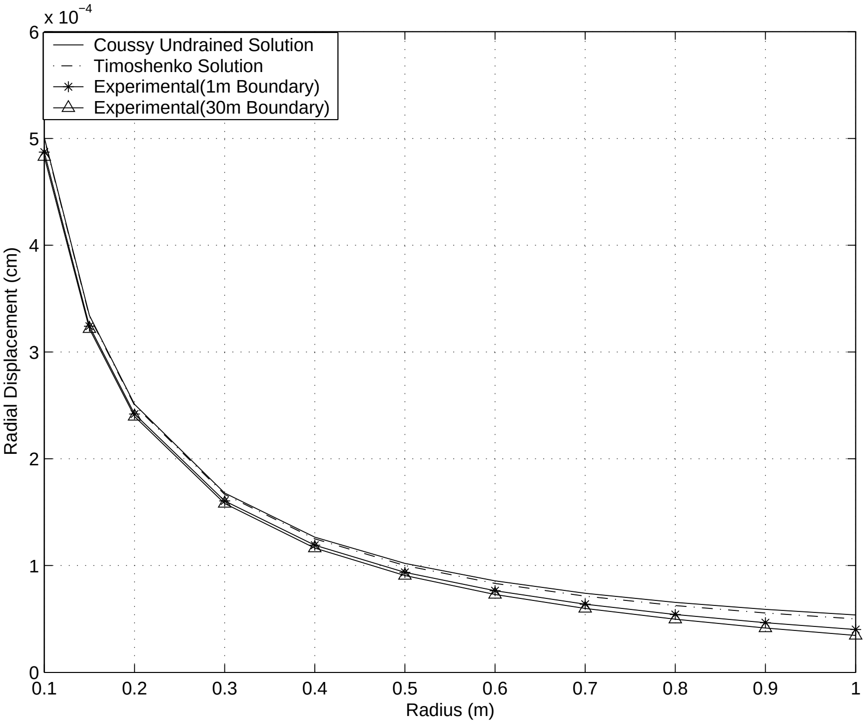


Figure 310.5: The comparison of radial solid displacement between two analytical solutions and expanded boundary

310.4 The Case of a Spherical Cavity

310.4.1 The Problem

Considering a medium composed of an isotropic, homogeneous, saturated thermoporoelastic material. In its initial state, it is assumed that the temperature, fluid pressure and stress fields are uniform and equal respectively, to T_0, p_0 and $\boldsymbol{\sigma}^0 = \sigma^0\mathbf{1}$(with $\sigma^0 < 0$). At time t= 0, a spherical cavity of radius r_0 is immediately drilled and filled with the same saturating fluid in the medium. For $t > 0$, the temperature and the pressure of the fluid are kept constant with the value of T_1 and p_1 respectively. The interface $r = r_0$ between the well and the porous medium is assumed to be in the thermodynamic equilibrium.

In spherical coordinates (r, θ, φ), the boundary conditions can be summarized as follows:

$$t \leq 0 \rightarrow \qquad \boldsymbol{\sigma}^0 = \sigma^0 \mathbf{1} \qquad p(r) = p_0 \qquad T(r) = T_0 \tag{310.16}$$

$$\begin{aligned} t > 0 \rightarrow \qquad & \sigma_{rr}(r_0) = -p_1 \qquad \sigma_{r\theta}(r_0) = \sigma_{r\varphi}(r_0) = 0 \\ \sigma_{rr}(r \rightarrow \infty) \rightarrow \sigma^0 \qquad & \sigma_{r\theta}(r \rightarrow \infty) = \sigma_{r\varphi}(r \rightarrow \infty) \rightarrow 0 \\ & p(r_0, t) = p_1 \qquad p(r \rightarrow \infty) \rightarrow p_0 \\ & T(r_0) = T_1 \qquad T(r \rightarrow \infty) \rightarrow T_0 \end{aligned} \tag{310.17}$$

Strictly speaking, the expressions for $r \rightarrow \infty$ are not boundary conditions. They are complementary conditions to be satisfied by the solution. It is used to model that at the point far from the disturbed area, the state of the medium are held as its initial state.

310.4.2 Analytical Solution

This is a problem of spherical symmetry. The radial displacement is the only non-zero displacement and all the fields are r and t dependent. Therefore,

$$\boldsymbol{\xi} = \xi^r(r) e_r \qquad p = p(r) \qquad T = T(r) \tag{310.18}$$

in which ξ^r is the radial displacement. In spherical coordinates, Eqn.310.18 yields

$$\epsilon_{rr} = \frac{\partial \xi^r}{\partial r} \qquad \epsilon_{\theta\theta} = \frac{\xi^r}{r} \qquad other \qquad \epsilon_{ij} = 0 \tag{310.19}$$

Based on the constitutive equations from Coussy (1995), it follows that

$$\sigma_{rr} = \sigma^0 + \lambda_0 (\frac{\partial \xi^r}{\partial r} + \frac{\xi^r}{r}) + 2\mu \frac{\partial \xi^r}{\partial r} - b(p - p_0) - 3\alpha K_0 (T - T_0) \tag{310.20}$$

$$\sigma_{\theta\theta} == \sigma_{\varphi\varphi} = \sigma^0 + \lambda_0 (\frac{\partial \xi^r}{\partial r} + \frac{\xi^r}{r}) + 2\mu \frac{\xi^r}{\partial r} - b(p - p_0) - 3\alpha K_0 (T - T_0) \tag{310.21}$$

$$other \qquad \sigma_{ij} = 0 \tag{310.22}$$

Finally combined with the Eqns. 310.19-310.22, it yields the near field or long-term solution (Coussy, 1995)

$$\xi^r = \frac{\sigma^0 + p_1}{4\mu} \frac{{r_0}^3}{r^2} + \frac{r_0 [b(p_1 - p_0) + 3\alpha K_0 (T_1 - T_0)]}{2(\lambda_0 + 2\mu)} (1 - \frac{{r_0}^2}{r^2}) \tag{310.23}$$

$$\sigma_{rr} = -p_1 \frac{{r_0}^3}{r^3} + \sigma_0 (1 - \frac{{r_0}^3}{r^3}) - \frac{2\mu [b(p_1 - p_0) + 3\alpha K_0 (T_1 - T_0)]}{\lambda_0 + 2\mu} (\frac{r_0}{r} - \frac{{r_0}^3}{r^3}) \tag{310.24}$$

$$\sigma_{\theta\theta} = \sigma_{\varphi\varphi} = p_1 \frac{{r_0}^3}{2r^3} - \sigma_0(1 + \frac{{r_0}^3}{2r^3}) - \frac{\mu[b(p_1 - p_0) + 3\alpha K_0(T_1 - T_0)]}{\lambda_0 + 2\mu}[\frac{r_0}{r}(1 + \frac{{r_0}^2}{r^2})] \tag{310.25}$$

And the diffusion process can be achieved if that the time are large enough with respect to the characteristics diffusion time relative to point r. When the boundary conditions for $r = r_0$ in fluid pressure and temperature which are $p = p_1$ and $T = T_1$ apply for the whole model, the following equations correspond to the undrained solution of the instantaneous drilling of a borehole in an infinite elastic medium.

$$\xi^r = \frac{\sigma_0 + p_1}{4\mu} \frac{{r_0}^3}{r^2} \qquad \sigma_{rr} = -p_1 \frac{{r_0}^3}{r^3} + \sigma_0(1 + \frac{{r_0}^3}{r^3})$$

$$\sigma_{\theta\theta} = \sigma_{\varphi\varphi} = p_1 \frac{{r_0}^3}{2r^3} + \sigma_0(1 + \frac{{r_0}^3}{2r^3}) \tag{310.26}$$

310.4.3 Discussion of the Results

The model is constructed as a quarter of a half ball. The cavity radius is $10cm$. As the outside boundary is fixed, to minimize the possibility of the sudden increase of the fluid bulk modulus, the outside radius of the sphere is set to be 2 m. The final mesh is generated as Fig.310.6. And the following boundary conditions apply: The nodes on XZ and YZ plane are fixed for solid and fluid displacement in Y and X direction respectively; the vertical solid and fluid displacement for the nodes on the XY plane are suppressed; for the nodes along the outside surface, to satisfy the complementary conditions, all the solid and fluid displacements are set to be zero as well. The pressure is translated in to nodal forces and applied in the radial direction. For simplicity, the hydrostatic stress σ_0 is equal to zero and with the assumption of thermodynamic equilibrium through the process, the temperature factor can be neglected. Also the initial fluid pressure p_0 is set to be 0 kPa. The analytical solution is studied below using the following set of parameters shown in Table 310.2.

As the same procedure in the previous drilling of borehole problem, we compared both the drained and undrained behavior. The drained and undrained behavior are tested by the permeability of $k = 3.6 \times 10^{-4} m/s$ and $k = 3.6 \times 10^{-8} m/s$ respectively. In drained behavior, we can see along the cavity surface, the experimental result of the radial displacement match the analytical solution very well. While with the increase of the radius, the decrease of the radial displacement for close solution is much smaller that of the experimental results. For the undrained behavior, we can see the radial displacement of the experimental results are always smaller than the close solution. Again it should be noted that the close solutions for the drained and undrained behavior along the cavity surface are exactly the same. This can be explained in the same way as the previous drilling of the borehole problem. When the experimental results from drained behavior are compared with the analytical undrained solution, it is observed they provide good agreement to each other as well.

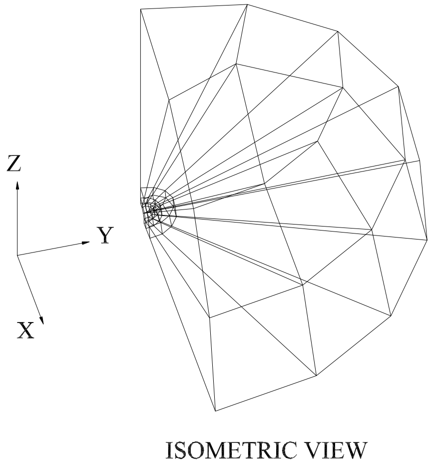


Figure 310.6: The mesh generation for the study of spherical cavity

Parameter	Symbol	Value	Units
Poisson Ratio	ν	0.2	-
Young's Modulus	E	1.2E+6	kN/m^2
Solid Bulk Modulus	K_s	3.6E+7	kN/m^2
Fluid Bulk Modulus	K_f	1.0E+17	kN/m^2
Solid Density	ρ_s	2.7	ton/m^3
Fluid Density	ρ_f	1.0	ton/m^3
Porosity	n	0.4	-

Table 310.2: Material Properties used to study spherical cavity problem

310.5 Line Injection of a fluid in a Reservoir

310.5.1 The Problem

Liquid water is usually injected into a reservoir from a primary well in order to recover the oil from a secondary well in petroleum engineering. This induces a problem of injecting a fluid into a cylindrical well of negligible dimensions.

Consider a reservoir of infinite extent composed of an isotropic, homogeneous and saturated poroelastic material. Through a cylindrical well of negligible dimensions, the injection of the same fluid is performed in all directions orthogonal to the well axis forming the Oz axis of coordinates. As a result of the axisymmetry and cylindrically infinite, all quantities spatially depends on r only. The injection starts at time $t = \Gamma$ and stops at time $t = \Gamma$. The flow rate of fluid mass injection is constant and equal to

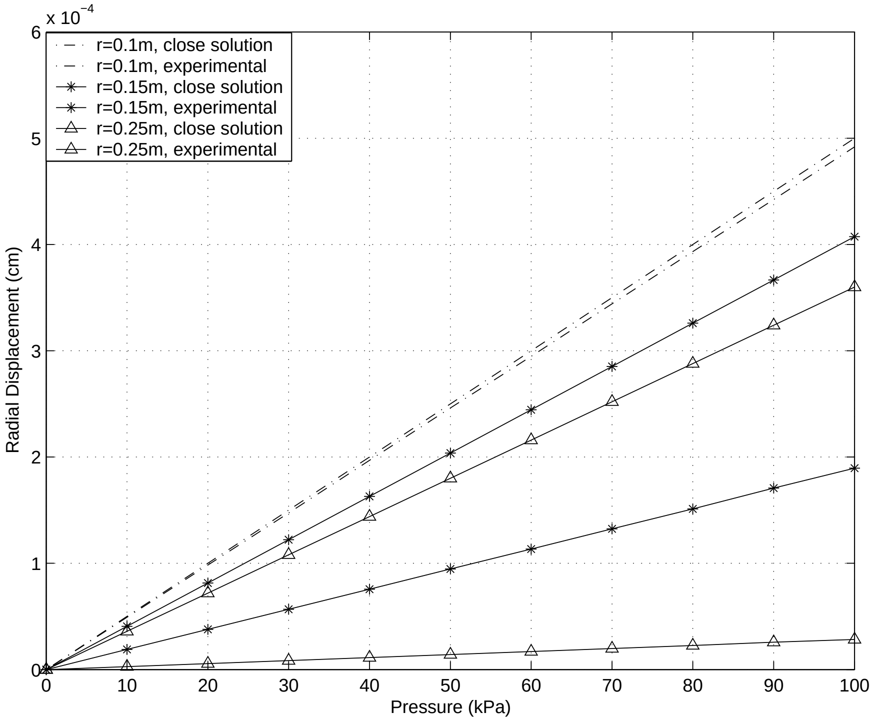


Figure 310.7: The comparison of radial solid displacement between analytical solution and experimental result for drained behavior

q.As a finite amount of Ω of fluid mass is injected instantaneously(i.e. $\Gamma q \to \Omega as \Gamma \to 0$).

310.5.2 Analytical Solution

This is a problem of cylindrically symmetry. Consequently the cylindrical coordinates(r,θ,z)is adopted. The vector of relative flow of fluid mass $\boldsymbol{w}$ reads

$$\boldsymbol{w} = w(r,t)e_r \tag{310.27}$$

where e_r is the unit vector along the radius. Using the fluid mass balance relationship, it yields

$$\int_0^r \frac{\partial m}{\partial t}(r,t) 2\pi r dr = q - 2\pi r w(r,t) \qquad \forall r,t \tag{310.28}$$

In addition, we require the fluid flow to reduce to zero infinitely far from the well

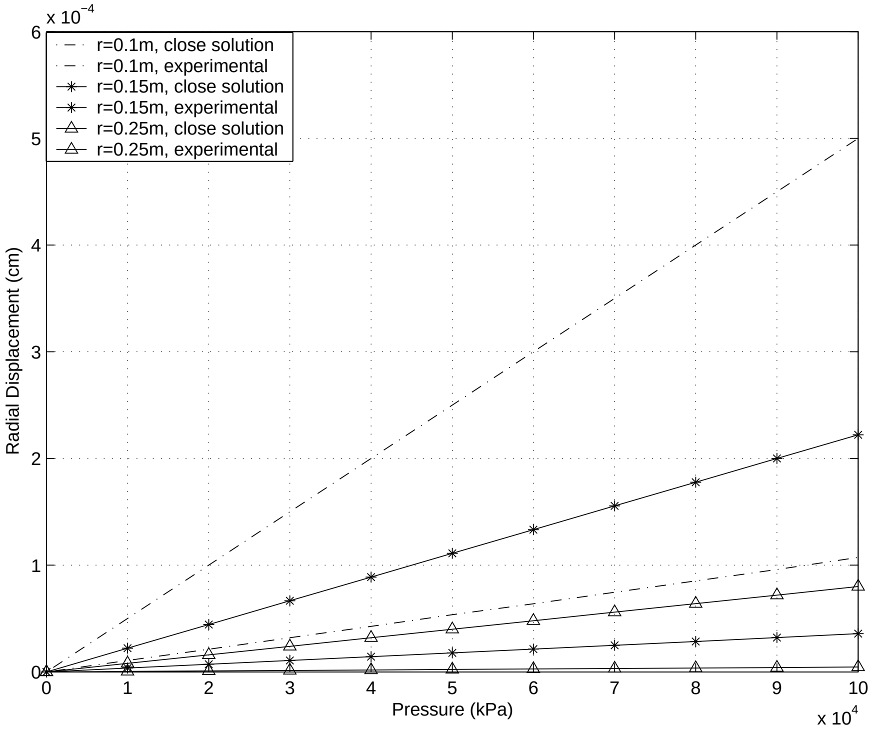


Figure 310.8: The comparison of radial solid displacement between analytical solution and experimental result for undrained behavior

$$rw \to 0 \qquad r \to \infty \qquad t < \infty$$

$$\int_0^\infty \frac{\partial m}{\partial t}(r,t) r dr = \frac{q}{2r} \qquad \forall 0 < t < \infty \tag{310.29}$$

Based on above Eqs.310.28-310.29, the radial displacement is derived in the form

$$p = \frac{\Omega}{4\pi \rho_0^{fl} k t} exp(-\frac{r^2}{4c_m t})$$

$$\xi_r = \frac{b\boldsymbol{M}\Omega}{2\pi \rho_0^{fl}(\lambda + 2\mu) r}[1 - exp(-\frac{r^2}{4c_m t})] \tag{310.30}$$

Using the constitutive equation, the stress field can be derived as follows:

$$\sigma_{rr} = -2\mu \frac{\xi_r}{r} \qquad \sigma_{\theta\theta} = 2\mu \frac{\xi^r}{r} - \frac{2\mu b}{\lambda_0 + 2\mu} p$$

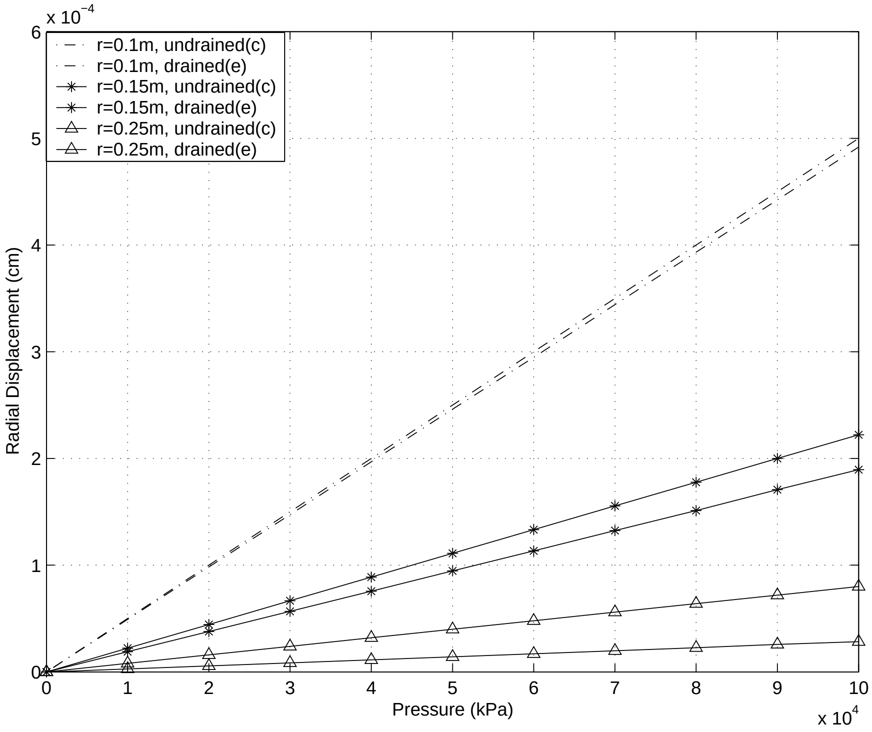


Figure 310.9: The comparison of radial solid displacement between analytical solution for undrained behavior and experimental result for drained behavior

$$\sigma_{zz} = -\frac{2\mu b}{\lambda_0 + 2mu} p \tag{310.31}$$

310.5.3 Discussion of the Results

As a result of axisymmetry, the model can be constructed as a quarter of a pie. The radius of the pie is 1 m and the thickness of the pie is 5 cm. A cylindrical well is drilled at the center of the pie, and its radius is 1 cm, which can be neglected in dimension when compared with the whole pie. The final mesh is shown as Fig.310.10. And the boundary conditions is as follows: As a consequence of plain strain problem, all the movements for solid and fluid in vertical direction Oz are suppressed; the solid and fluid displacement for the nodes along the X axis and Y axis are fixed in Y and X direction respectively for the reason of axisymmetry; the nodes along the outside perimeter are fixed in the solid and fluid displacement with the assumption of infinite medium. To the difference with the previous problems, the traction boundary conditions are applied on the fluid displacement. It should be noted that the Ω mention in the above equations is the volume of the fluid injected per unit of vertical well length and has a unit of m^3/m. In order to generate the volume of 1 cm^3/m, the corresponding fluid displacement of the nodes along the well has been calculated and applied as a step function at the time of 0 sec. For simplicity, the initial fluid pressure p_0 is set to be 0 kPa. The analytical solution is studied below using the following set of parameters shown in Table 310.3.

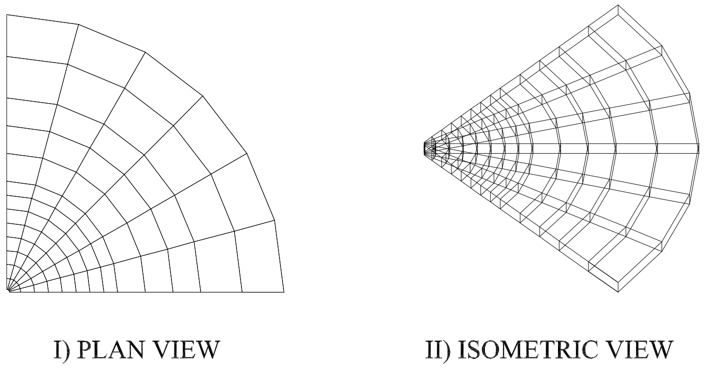


Figure 310.10: The mesh generation for the study of line injection problem

In the analysis, the pore pressure and the radial displacement are studied. The results are recorded from three points at the radius of 10 cm, 50 cm and 85 cm. The close solution and experimental results are shown in Fig.?? and Fig.??. As the time step is set to be 1 sec, the first data point starts at the time of 1 sec. From the pore pressure plot we can see that the build-up of the pore pressure reach the peak value of 34 kPa at the radius of 85 cm. With the decrease of the radius, the pore pressure decreases as well. This can be explained by the fact that the closer the point to the injection location, the earlier and the larger load is applied, so the pore pressure dissipates faster. And as time passes by, we can see the pore pressure progressively dissipates and finally almost reaches the same value within the model. The same phenomena can be been observed from the radial solid displacement. The maximum solid displacement occurs at the radius of 85 cm, which means more coupling between the solid and fluid phase, as consequence, the pore pressure should have the largest value. This corresponds to the previous result. With the increase of the time, the radial solid displacement get closer to zero, which means the fluid moves out the solid skeleton.

Parameter	Symbol	Value	Units
Poisson Ratio	ν	0.2	-
Young's Modulus	E	1.2E+6	kN/m^2
Solid Bulk Modulus	K_s	3.6E+7	kN/m^2
Fluid Bulk Modulus	K_f	1.0E+17	kN/m^2
Undrained Bulk Modulus	K_u	6.0E+7	kN/m^2
Bulk Modulus	K	6.7E+5	kN/m^2
Solid Density	ρ_s	2.7	ton/m^3
Fluid Density	ρ_f	1.0	ton/m^3
Fluid Diffusivity coefficient	c_f	0.4973	m^2/s
Porosity	n	0.4	-
Permeability	k	3.6E-6	m/s

Table 310.3: Material Properties used to study the line injection problem

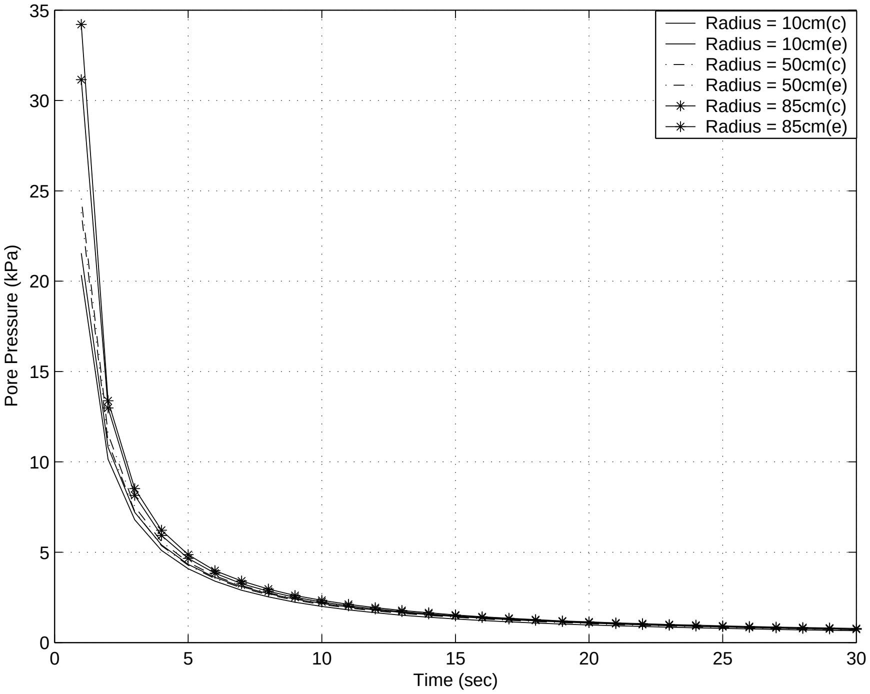


Figure 310.11: The comparison between analytical solution and experimental result for pore pressure

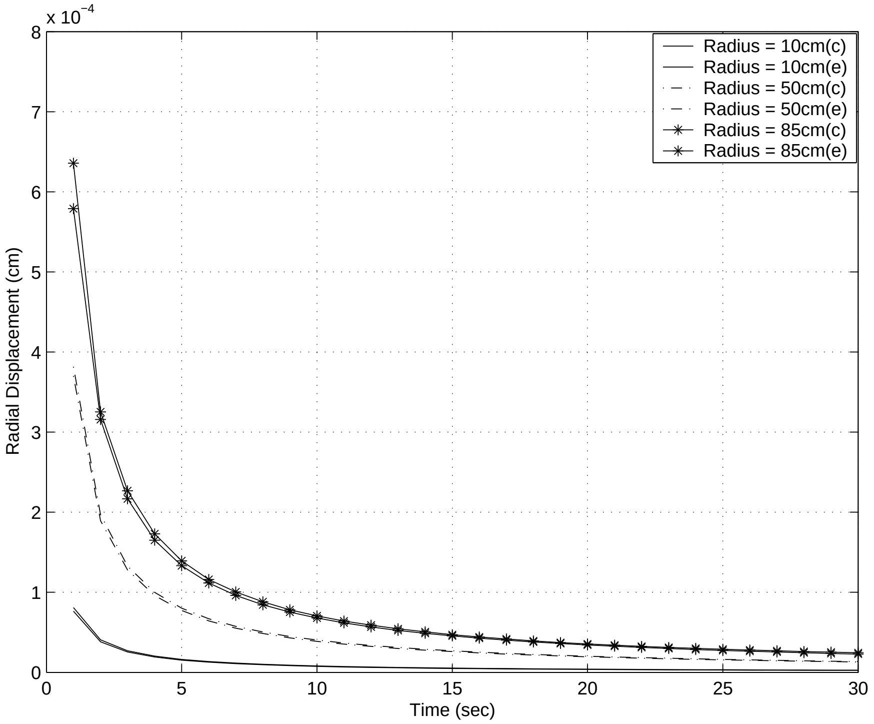


Figure 310.12: The comparison between analytical solution and experimental result for radial displacement

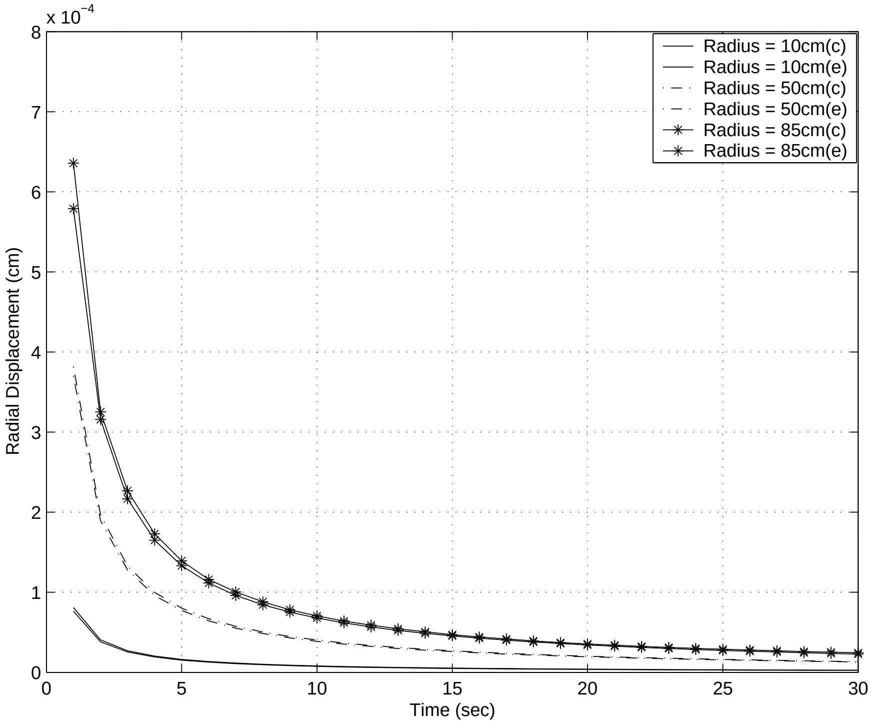


Figure 310.13: The comparison between analytical solution and experimental result for radial displacement

Table 310.4: Simulation parameters used for the shock wave propagation verification problem.

Parameter	Symbol	Value
Poisson ratio	ν	0.3
Young's modulus	E	1.2×10^6 kN/m^2
Solid particle bulk modulus	K_s	3.6×10^7 kN/m^2
Fluid bulk modulus	K_f	2.17×10^6 kN/m^2
Solid density	ρ_s	2700 kg/m^3
Fluid density	ρ_f	1000 kg/m^3
Porosity	n	0.4
Newmark parameter	γ	0.6

310.6 Shock Wave Propagation in Saturated Porous Medium

In order to verify the dynamic behavior of the system, an analytic solution developed by Gajo (1995) and Gajo and Mongiovi (1995) for 1C shock wave propagation in elastic porous medium was used. A model was developed consisting of 1000 eight node brick elements, with boundary conditions that mimic 1D behavior. In particular, no displacement of solid $(u_x = 0, u_y = 0)$ and fluid $(U_x = 0, U_y = 0)$ in x and y directions is allowed along the height of the model. Bottom nodes have full fixity for solid $(u_i = 0)$ and fluid $(U_i = 0)$ displacements while all the nodes above base are free to move in z direction for both solid and fluid. Pore fluid pressures are free to develop along the model. Loads to the model consist of a unit step function (Heaviside) applied as (compressive) displacements to both solid and fluid phases of the model, with an amplitude of 0.001 cm. The u–p–U model dynamic system of equations was integrated using Newmark algorithm (see section 108.3). Table 310.4 gives relevant parameters for this verification.

Two set of permeability of material were used in our verification. The first model had permeability set $k = 10^{-6}$ cm/s which creates very high coupling between porous solid and pore fluid. The second model had permeability set to $k = 10^{-2}$ cm/s which, on the other hand creates a low coupling between porous solid and pore fluid. Comparison of simulations and the analytical solution are presented in Figure 310.14.

Before proceeding to the analysis, the following assumptions are made: For high-frequency components, the permeability remains constant; thus, the dependency of the permeability on the frequency is neglected. Unless specified, all the models in this report are elastic isotropic.

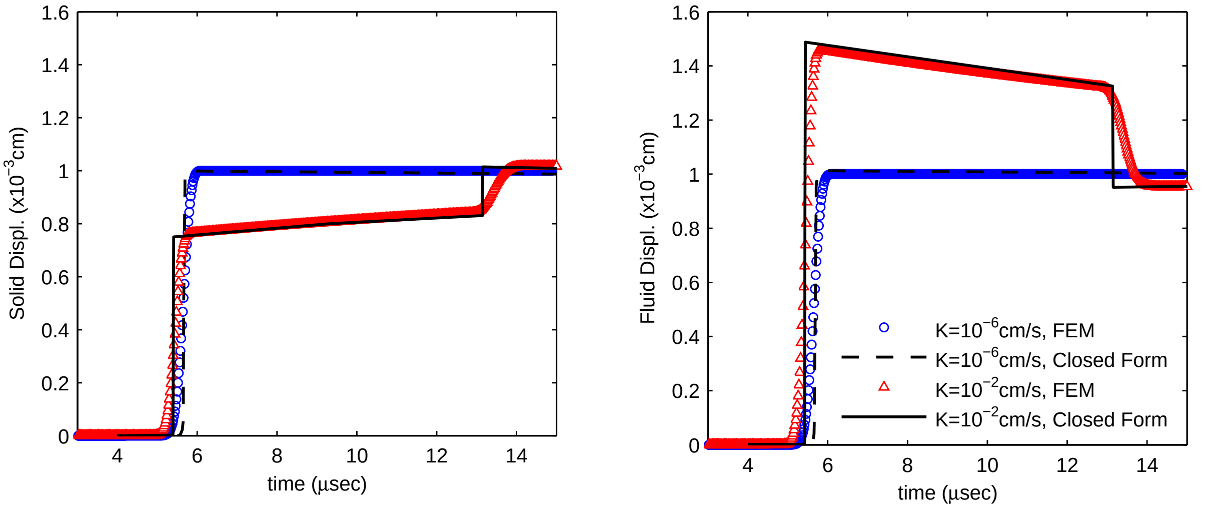


Figure 310.14: Compressional wave in both solid and fluid, comparison with closed form solution.

310.7 Vertical Consolidation of a soil layer by Coussy (2004)

310.7.1 Brief review of Analytical Solution for Consolidation by Coussy (2004)

The consolidation process can be defined as follows: When an elastic soil layer is subjected to an external change in mean normal stress, immediately the water will alone sustain this increment of mean normal stress and cause the build-up the excessive pore water pressure. In the progress of the flow of the water to the surface, the load is gradually transferred to the soil skeleton and the excessive pore water pressure will dissipate. At the same time, the settlement of the soil layer occurs. As settlement is usually a major concern in geotechnical engineering, this is a key problem in soil mechanics.

Consider a soil layer composed of an isotropic, homogeneous and saturated thermoporoelastic material. The layer has a thickness of h in the Oy direction and of infinite extent in the two other directions Ox and Oy. The layer is underlain by a rigid and impervious base at $y = 0$. And the top surface at $y = h$ is so perfectly drained that the pore pressure is held constant as zero.

At the initial state of the soil layer, the thermal effects are neglected so that the boundary conditions follow that:

$$t \leq 0 \rightarrow \qquad y = h \qquad p = 0 \tag{310.32}$$

$$y = 0 \qquad \frac{\partial p}{\partial z} = 0 \tag{310.33}$$

At time $t = 0$, tan instantaneous vertical load $-\varpi e_y$ is suddenly applied on the top surface $y = h$, the induced boundary conditions require that

$$t > 0 \rightarrow \qquad y = h \qquad \boldsymbol{\sigma} e_y = -\varpi e_y \tag{310.34}$$

The undeformability of the substratum reads

$$y = 0 \qquad \xi = 0 \tag{310.35}$$

The impermeability implies

$$y = 0 \qquad -w \cdot e_y = -w_y = 0 \tag{310.36}$$

The problem is then to determine the new fields of fluid pressure, stress and displacement induced by the external loading.

Since this is a one-dimensional problem, the only non-zero displacement is the vertical displacement ξ^y. But in particular the fluid pressure depends only on y and t.

$$\boldsymbol{\xi} = \xi^y(y,t)e_y \qquad p = p(y,t) \tag{310.37}$$

Based on the constitutive equations from Coussy (2004), it follows that

$$\sigma_{yy} = (\lambda_0 + 2\mu)\frac{\partial \xi^y}{\partial y} + bp \tag{310.38}$$

$$\sigma_{xx} = \sigma_{zz} = \frac{\lambda_0}{(\lambda + 2\mu)}\sigma_{yy} - \frac{2\mu b}{\lambda_0 + 2\mu}p \tag{310.39}$$

And because the fluid pressure p must be an ordinary function of time t, although the derivative of p is infinite at time $t = 0$ according to the consolidation equation (Coussy, 2004), the discontinuity of the fluid pressure p at time $t = 0$ must satisfy

$$p(y, t = 0^{+}) = \eta\varpi \qquad \varpi = \frac{\nu - \nu_0}{(1-\nu)(1-2\nu_0)b} \tag{310.40}$$

where ν and ν_0 are the drained and undrained Poisson ratio, respectively. For time $t > 0$, the vertical stress $\sigma_{yy} = -\varpi$ is constant in time and space, therefore the diffusion equation reads

$$t > 0 \qquad c_m \frac{\partial^2}{\partial y^2} p = \frac{\partial}{\partial t} p \tag{310.41}$$

Collecting the above results, finally the fluid pressure reads

$$p(y,t) = \eta\varpi \sum_{n=0}^{\infty} \frac{4(-1)^n}{\pi(2n+1)} cos[\frac{(2n+1)\pi}{2}\frac{y}{h}] exp[-\frac{(2n+1)^2\pi^2}{4}\frac{t}{\tau}] \tag{310.42}$$

Each term of the series decreases exponentially with respect to the ratio $\frac{t}{\tau}$, in which τ is a characteristics consolidation time

$$\tau = \frac{h^2}{c_m} \qquad c_m = \boldsymbol{kM}\frac{\lambda_0 + 2\mu}{\lambda + 2\mu} \tag{310.43}$$

where λ and λ_0 are the drained and undrained Lame coefficient, respectively. Given by the Eqn.310.39, the only non-zero displacement ξ^y satisfies

$$\frac{\partial \xi^y}{\partial y} = \frac{1}{\lambda_0 + 2\mu}(\sigma_{yy} + bp) \tag{310.44}$$

By substituting the value of $-\varpi$ of the vertical stress and expression of (310.44), the series converges and it can integrated term by term yielding

$$\xi^y(y,t) = \frac{\varpi}{\lambda_0 + 2\mu}(\frac{y}{h} + \frac{8\eta b}{\pi^2}) \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)^2} sin[\frac{(2n+1)\pi}{2}\frac{y}{h}] exp(-\frac{(2n+1)^2\pi^2}{4}\frac{t}{\tau}] \tag{310.45}$$

$$\tag{310.46}$$

Using Eqn.310.46 and substitute $y = h$, the settlement can be expressed as

$$s(t) = s_{\infty} + (s_{0^+} - s_{\infty}) \sum_{n=0}^{\infty} \frac{8}{\pi^2(2n+1)^2} exp\{-\frac{(2n+1)^2\pi^2}{4}\frac{t}{\tau}\} \tag{310.47}$$

$$s_{0^+} = \frac{h\varpi}{\lambda + 2\mu} \qquad s_{\infty} = \frac{h\varpi}{\lambda_0 + 2\mu} \tag{310.48}$$

310.7.2 Numerical Analysis

A soil column of ten brick $u-p-U$ finite elements is used to model the horizontal layer. The height of the soil column is 10 m and the height of each element has dimensions $1m \times 1m \times 1m$, illustrated in Fig. 310.15. The material properties, shown in Table 310.5, are chosen as representative values for the natural soil deposit. Developed $u-p-U$ finite element model, can simulate realistic compressibility of the pore fluid. However, it is important to note that the analytical solution for the vertical consolidation is based on the assumption that both the soil particles and the pore fluid (water) are completely incompressible. A uniform vertical pressure of 400 kPa is applied on the top surface of the soil column. The numerical analysis was performed in two stages:

(I) Self-Weight application (see Figures 310.15 to 310.19).

(II) Consolidation with drainage at top due to uniform vertical pressure of 400 kPa at the surface (see Figures 310.20 to 310.26).

Table 310.5: Material Properties used to study consolidation of a soil layer.

Parameter	Symbol	Value
gravity acceleration	g	9.81 m/s^2
soil matrix Young's Modulus	E	10×10^3 kN/m^2
soil matrix Poisson's ratio	v	0.25
solid particle density	ρ_s	2.65×10^3 kg/m^3
water density	ρ_f	1.0×10^3 kg/m^3
solid particle bulk modulus	K_s	37.0×10^6 kN/m^2
fluid bulk modulus	K_f	2.2×10^6 kN/m^2
porosity	n	0.46
Biot coefficient	α	1.0

The following boundary conditions are applied to the model (Fig 310.27): As the bottom of the soil column is modeled as an undeformable and impermeable layer, both the solid and fluid displacements are fixed. The pore pressure is kept constant as zero at the top surface of the soil column because of the perfectly drained condition. In order to simulate the 1D consolidation problem, all the lateral movement of the solid and fluid phase are suppressed so that the vertical displacement is the only non-zero displacement for the intermediate nodes. To capture both the long term (t >0.1 sec) and short term (t <0.1 sec) response of the soil column, two different time steps are adopted: 0.1 sec and 0.005 sec, respectively. In order to observe the dissipation of the excessive pore water pressure in a reasonable

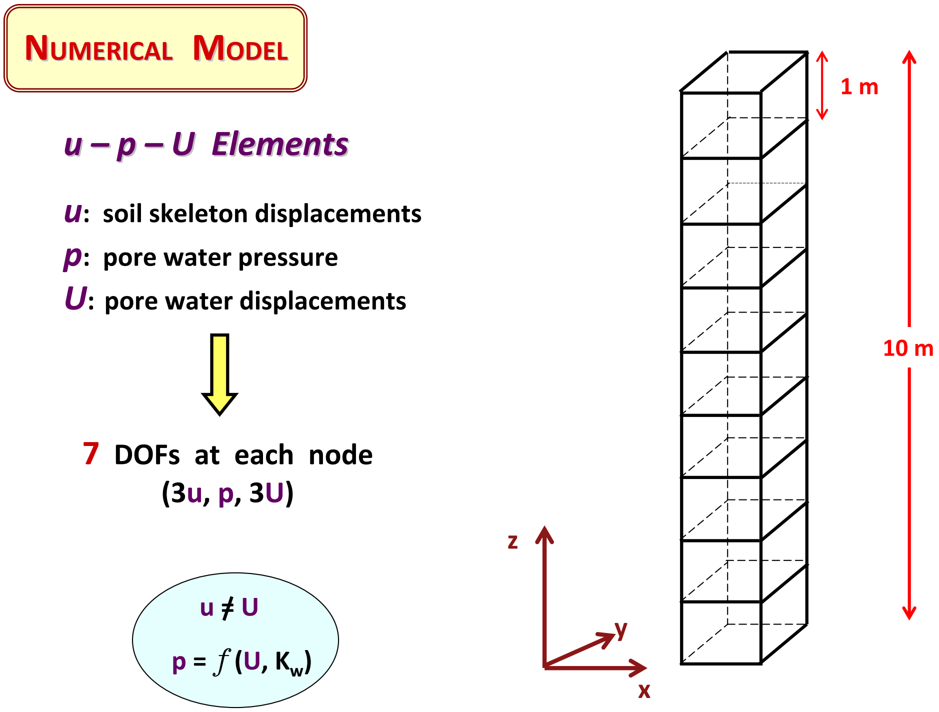


Figure 310.15: Numerical Simulation of self weigh application and consolidation of a soil layer using u-p-U brick elements.

and convenient period, we select $k = 1.0 \times 10^{-4} m/s$ as the value for the permeability. To cure the artificial oscillation, some numerical damping is introduced into the analysis by using $\gamma = 0.6$ and $\beta = 0.3025$ for the Newmark integrator. Fig. 310.20 illustrates the physical geometry of the problem whereas Fig. 310.21 shows the numerical modeling.

Based on the above parameters, the other relative parameters can be calculated as follows:

The bulk modulus of the mixture:

$$K = \frac{E}{3(1-2\nu)} = 6.67 \times 10^3 kPa \tag{310.49}$$

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} = 6.67 \times 10^3 kPa \quad \mu = \frac{E}{2(1+\nu)} = 4 \times 10^3 kPa \tag{310.50}$$

The Biot coefficient:

$$b = 1 - \frac{K}{K_s} = 0.9998 \tag{310.51}$$

The undrained bulk modulus of the mixture:

$$N = \frac{K_s}{b-n} = 6.85 \times 10^7 kPa \quad M = \frac{K_f N}{K_f + Nn} = 4.47 \times 10^6 kPa \tag{310.52}$$

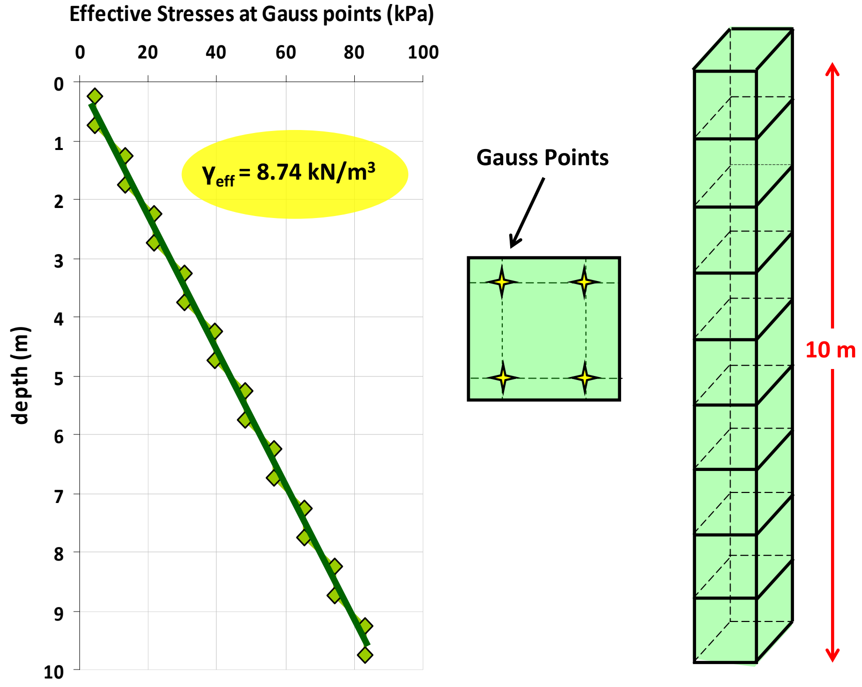


Figure 310.16: The distribution of the vertical effective stresses with depth due to self-weight. The stresses are calculated at the Gauss points within each brick element. The stresses obtained from the numerical analysis are equal for all the Gauss points within the element, due to the compatibility of deformations at the element interfaces.

$$K_u = K + b^2 M = 4.4 \times 10^6 kPa \tag{310.53}$$

The diffusion coefficient and characteristic time of consolidation:

$$c_f = \frac{kM}{\gamma_w} \frac{K + \frac{4\mu}{3}}{K_u + \frac{4\mu}{3}} = 1.2 m^2/s \qquad t = \frac{h^2}{c_f} = 83.33 s \tag{310.54}$$

310.7.3 Discussion of Numerical Results - Conclusions

The stage of self weight application shows in Fig. 310.19 that the expected estimated settlement is quite close to the one obtained from the analysis. The difference in the two values is due to the stress distribution coming out of the analysis which is slightly different than the one considered in theory (see Fig. 310.16) and possibly due to the compressibility of soil particles in the numerical analysis.

In Fig.310.24, the normalized fluid pressure is plotted against the location for various normalized times. For normalized time $T_v = 0.1$, only the nodes close to the top free flow surface display the dissipation of the pore pressure. The experimental result provide good agreement with the analytical

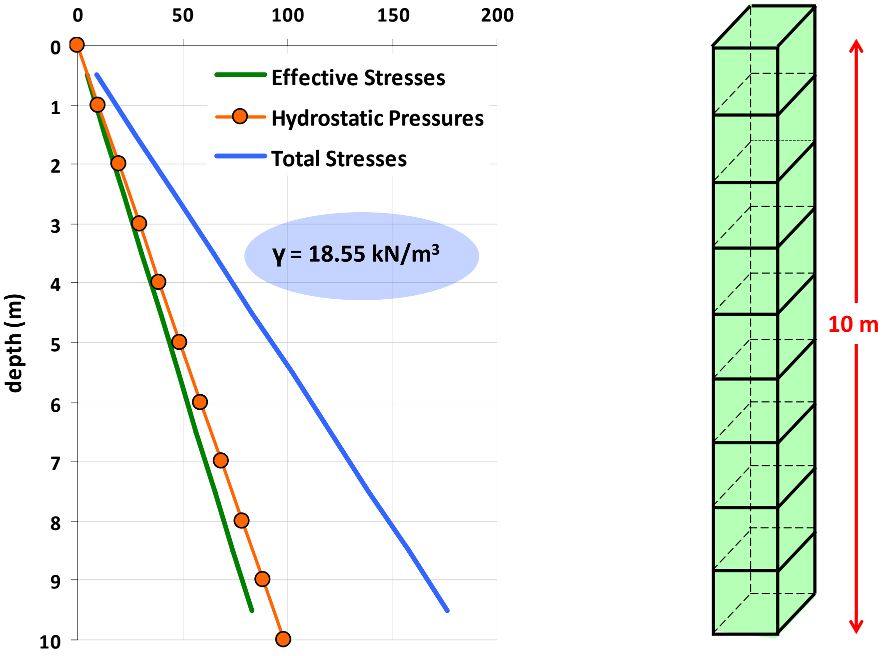


Figure 310.17: The distribution of the hydrostatic pore pressures, the effective and total stresses with depth, obtained from the numerical analysis after self-weight application.

solution. With the increase of the normalized time, we can clearly see the tendency of the dissipation of the water. At normalized time $T_v = 1.0$ (natural time t $=$ 83 sec), the maximum normalized pore pressure is only about 0.11. It can be concluded that the numerical analysis can effectively demonstrate the process of the dissipation of the pore pressure.

In Fig. 310.26, the change of the porosity is predicted due to the consolidation of the soil layer. Both the change of volume of the soil $(0.1782m^3)$ and the fluid $(0.1794m^3)$ have been calculated. Theoretically, these two values should be the same, according to the fact that the settlement of the soil layer is due the fluid which is squeezed out, assuming that the soil grains are incompressible. However, this difference in the values is due to the compressibility of the soil grains, which was not considered infinite in the numerical analysis.

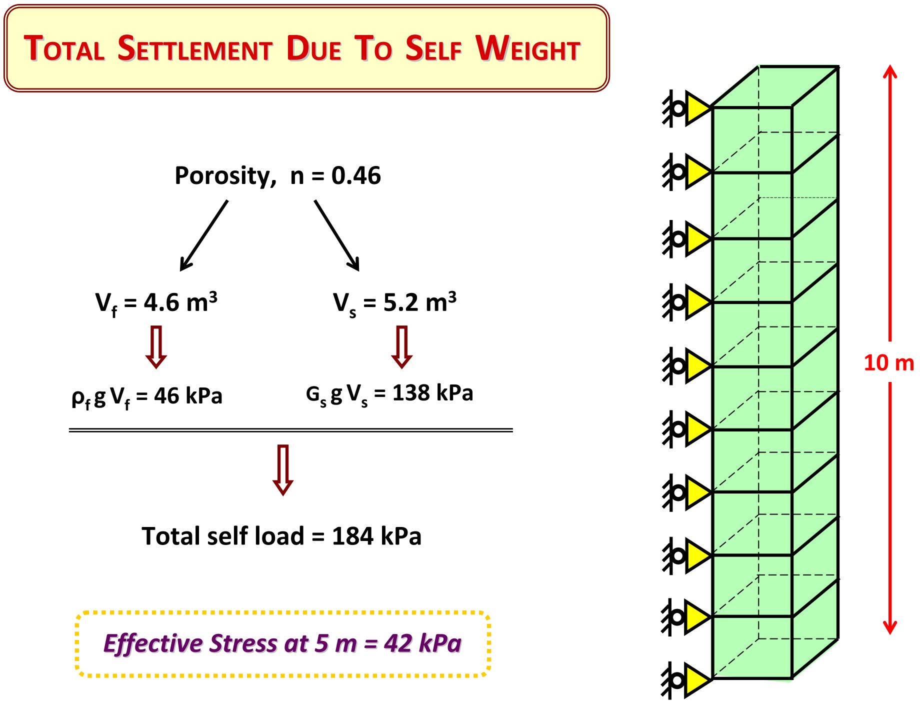


Figure 310.18: Estimation of the total self load, as it is obtained from the numerical analysis, given the porosity and the densities of the fluid (ρ_f) and the grains (ρ_s or G_s).

310.8 One dimensional wave propagation in elastic porous media subjected to step displacement boundary condition

310.8.1 Brief review of Analytical Solution by Gajo and Mongiovi (1995)

An one-dimensional exact analytical solution of the Biot's equations is provided by Gajo and Mongiovi (1995) for the completely general solution of the transient problem in saturated, linear, elastic, porous media. The analytical solution was obtained was obtained by Fourier series. This solution is considered to be completely general because it is not based on any assumptions with respect to the inertial, viscous or mechanical coupling. Furthermore, it can be applied to any type of boundary-initial value problem.

The advantage of this analytical solution consists of showing the mechanics of dispersive wave propagation in saturated elastic solids. This is achieved by allowing the detailed analysis of wave fronts of the first and second kind of longitudinal waves and by analyzing accurately the effects of each term of coupling on the transient behavior of saturated porous media. In particular, since each term of the Fourier series represents a frequency component of the excitation signal, the analytical solution can

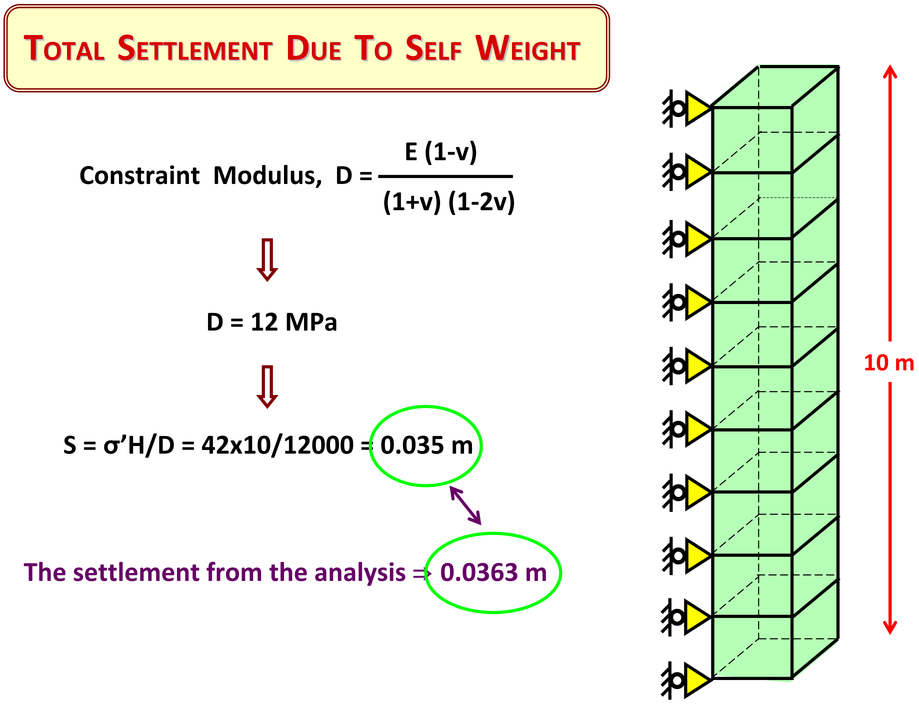
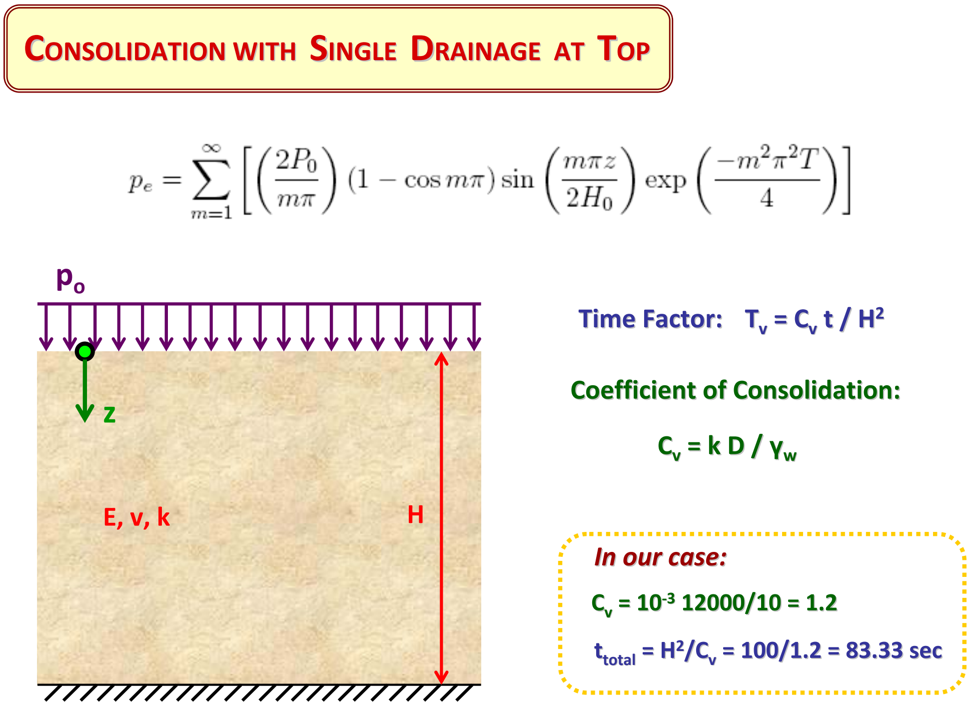


Figure 310.19: Prediction of the total settlement due to self-weight and comparison with the numerical result.

describe the behavior of each frequency component. Thus, it can illustrate the mechanics of dispersive wave propagation in which higher frequencies propagate with two waves and lower frequencies with only one wave, as a function of permeability and travel length.

Considering the above mentioned arguments, the analytical solution can provide a useful comparative term towards the verification and the validation of the existing numerical solutions based on the finite element method. Such a study was conducted by Gajo et al. (1994), by comparing analytical results with numerical ones obtained by a $u-p-U$ numerical formulation.

In the paper by Gajo and Mongiovi (1995), the transient response of porous media is shown for typical material properties of a natural granular deposit and for different degrees of viscous coupling. Specifically, analytical results are given from the solution of the following one-dimensional boundary value problem: at the top and bottom surfaces of a soil layer of finite thickness L, the excitation consisting of a step displacement boundary condition (Heaviside function) is applied to both solid and fluid phases. This problem can demonstrate better the mechanics of dispersive wave propagation, since the step excitation contains waves of all kind of frequencies. The analytical solution is relative only to the first arrival of the waves of the first and second kind.



CONSOLIDATION WITH SINGLE DRAINAGE AT TOP

$$p_e = \sum_{m=1}^{\infty} \left[\left(\frac{2P_0}{m\pi} \right) (1 - \cos m\pi) \sin \left(\frac{m\pi z}{2H_0} \right) \exp \left(\frac{-m^2\pi^2 T}{4} \right) \right]$$

Time Factor: $T_v = C_v \, t / H^2$

Coefficient of Consolidation:

$C_v = k \, D / \gamma_w$

In our case:

$C_v = 10^{-3} \; 12000/10 = 1.2$

$t_{total} = H^2/C_v = 100/1.2 = 83.33$ sec

Figure 310.20: The physical geometry of the problem.

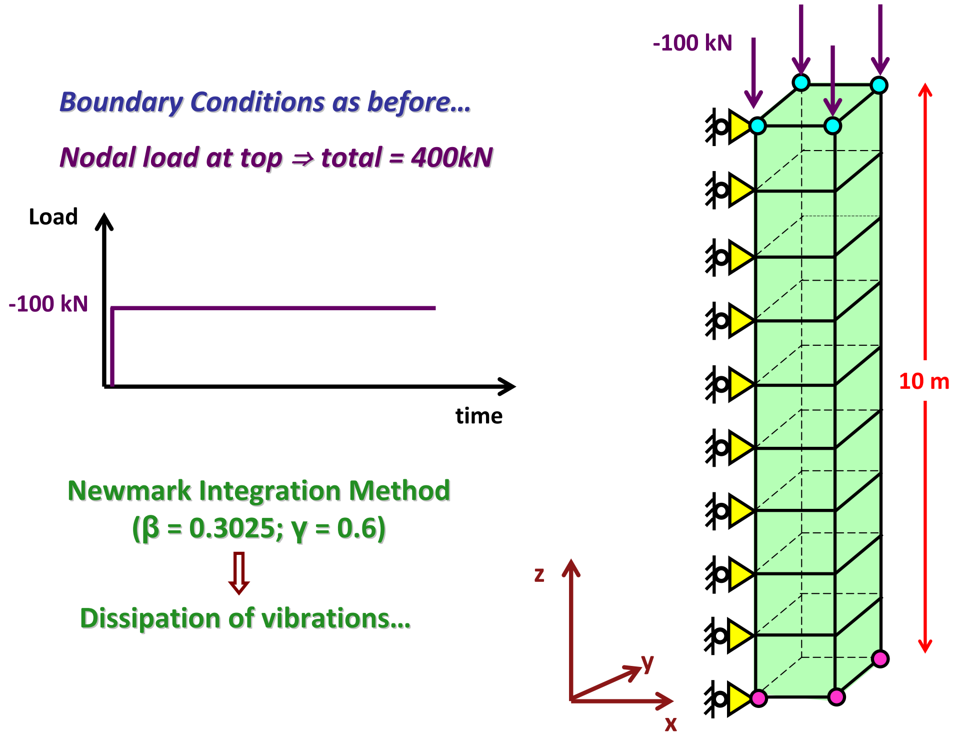


Figure 310.21: Numerical simulation of 1D consolidation of a soil layer.

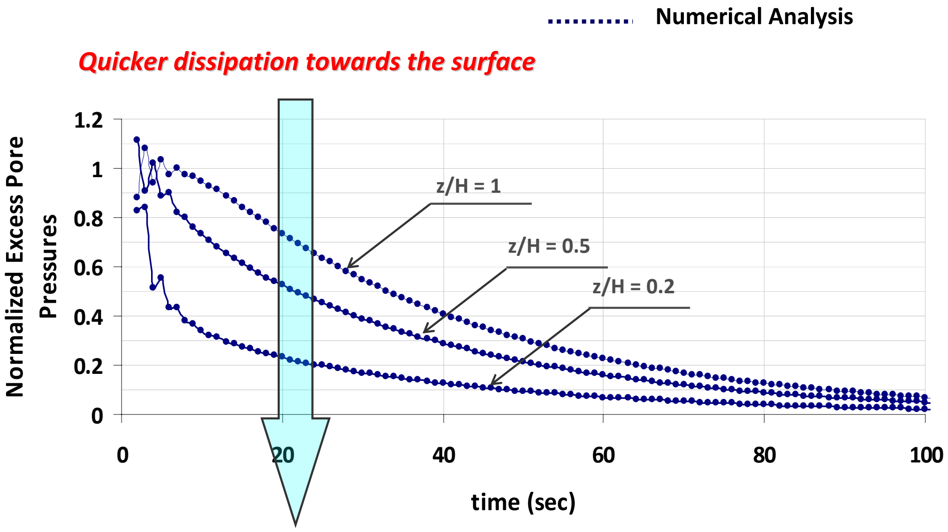


Figure 310.22: Time history of the normalized excess pore pressure for three different normalized depths indicating faster dissipation close to the surface. The dissipation has practically been completed at $t = 83.33 sec$, as predicted.

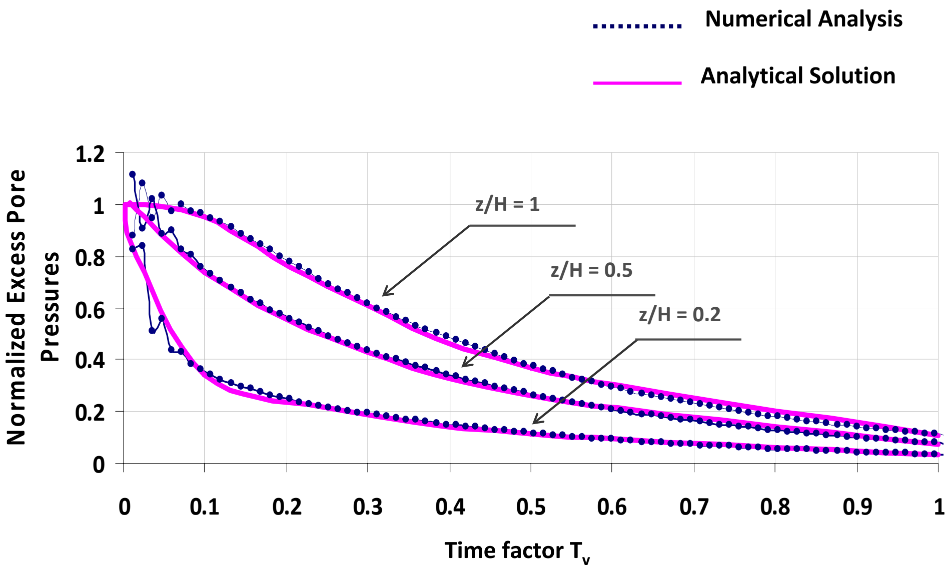


Figure 310.23: Normalized excess pore pressures versus normalized time for three different normalized depths. Comparison of numerical results with the analytical ones.

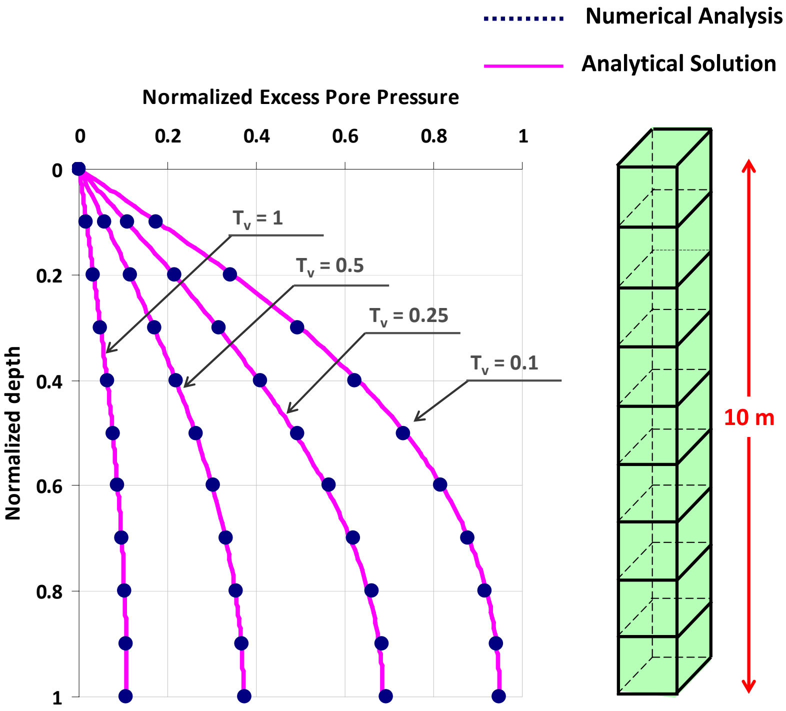


Figure 310.24: Distribution of normalized excess pore pressures with normalized depth for four different time factors. Comparison of numerical results with the analytical ones.

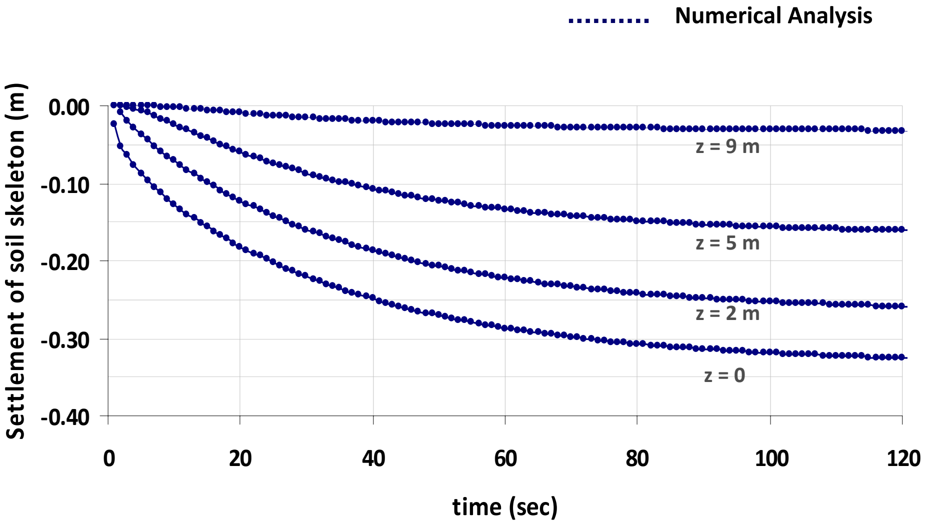
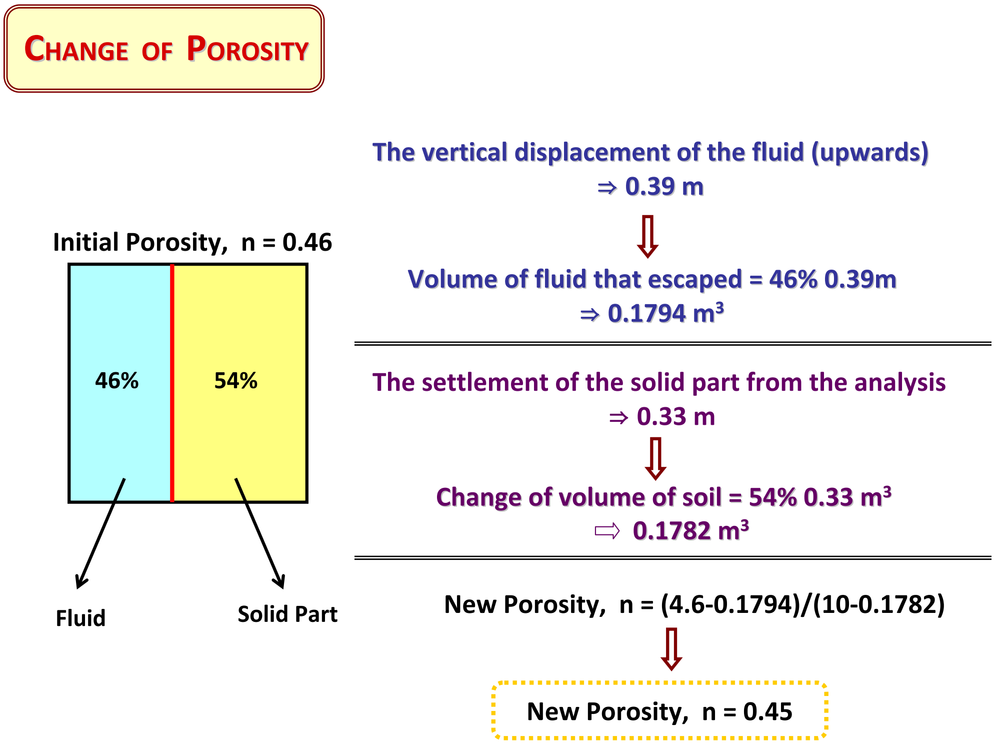


Figure 310.25: Time history of the settlements for four different depths.



CHANGE OF POROSITY

The vertical displacement of the fluid (upwards)
⇒ 0.39 m

⇩

Volume of fluid that escaped = 46% 0.39m
⇒ 0.1794 m^3

The settlement of the solid part from the analysis
⇒ 0.33 m

⇩

Change of volume of soil = 54% 0.33 m^3
⇨ 0.1782 m^3

New Porosity, n = (4.6-0.1794)/(10-0.1782)

⇩

New Porosity, n = 0.45

Figure 310.26: Prediction of the change in porosity of the soil layer due to consolidation.

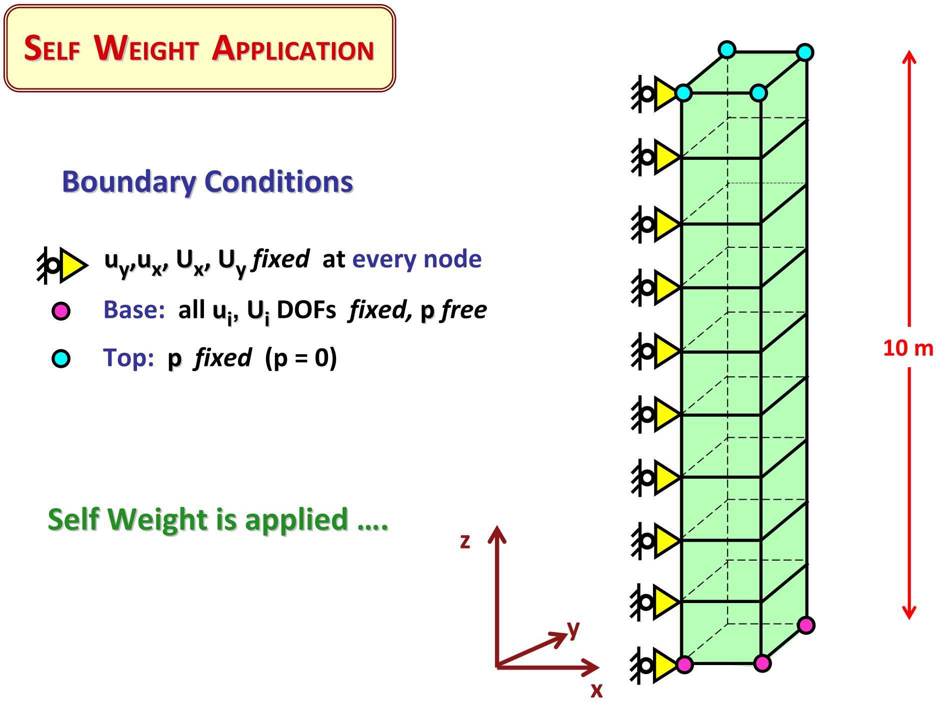


Figure 310.27: Boundary Conditions applied to the numerical model throughout the self-weight application and the process of consolidation due to extra load applied to the surface of the soil column.

310.8.2 Numerical Analysis

Numerical examples for three different values of viscous coupling ($k = 10^{-8} cm^3 s/g$, $k = 10^{-6} cm^3 s/g$, $k = 10^{-5} cm^3 s/g$) were solved in order to verify the previously mentioned $u - p - U$ formulation in a wide range of drag. The numerical model used for the simulation of the 1C shock wave propagation consists of 400 u-p-U brick finite elements of dimensions $0.01cm \times 0.01cm \times 0.01cm$ creating a soil column 4cm thick. Figure 310.28 illustrates the transition from the physical configuration of the problem to its numerical simulation. Table 310.6 shows the soil properties of the numerical model.

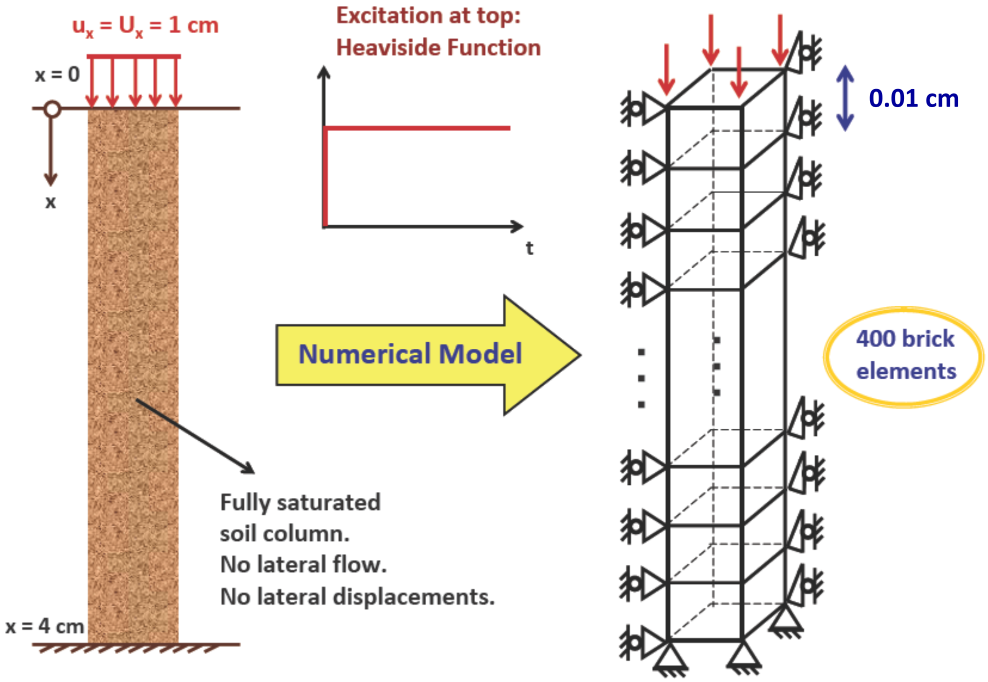


Figure 310.28: The numerical model used for the verification of the finite element implementation through comparison with the analytical results provided by Gajo and Mongiovi (1995).

At the top surface of the soil column, a step displacement of $1.0 \times 10^{-3} cm$ is applied both to the solid and the fluid phase. Only the vertical displacement is free. There is no lateral flow or displacement. The degree of freedom related to the pore pressures is constrained at the top surface to be equal to the atmospheric pressure and is free at the rest of the nodes. The base of the model is rigid and impervious.

This kind of excitation (Heaviside function) applied at the top of the model, results clearly in waves of all kinds of frequency, first due to its nature and secondarily due to the way of its application. This fact together with the great stiffness of the solid skeleton (see Table 310.6 require a very dense spatial

Table 310.6: Soil Properties for 1C shock wave propagation for the problem by Gajo and Mongiovi (1995).

Parameter	Symbol	Value
gravity acceleration	g	9.81 m/s^2
soil matrix Young's Modulus	E	1.2×10^6 kN/m^2
soil matrix Poisson's ratio	v	0.3
solid particle density	ρ_s	2.7×10^3 kg/m^3
water density	ρ_f	1.0×10^3 kg/m^3
solid particle bulk modulus	K_s	36.0×10^6 kN/m^2
fluid bulk modulus	K_f	2.177×10^6 kN/m^2
porosity	n	0.4
Biot coefficient	α	1.0

discretization. Here, 400 u-p-U brick finite elements of dimensions $0.01cm \times 0.01cm \times 0.01cm$, following similar discretization with Gajo et al. (1994). The time step, δt required needs to be limited to

$$\delta t < \frac{\delta h}{v} \tag{310.55}$$

$$\tag{310.56}$$

where v is the highest wave velocity. In our case, the temporal integration involves 800 steps of $2.0 \times 10^{-8} sec$, which allows a maximum wave velocity of $5.0 \times 10^5 m/s$.

Two different time integration methods where used: i) The Newmark integrator and ii) The Hilber-Hughes-Taylor (HHT) Integrator. Sets of parameters, assuring unconditionally numerical stability, were chosen for both integrators. For the case of Newmark integrator (see Figures 310.29 - 310.44), the following sets of parameters were used:

a) $\gamma = 0.5$ and $\beta = 0.25$,

b) $\gamma = 0.6$ and $\beta = 0.3025$,

c) $\gamma = 0.7$ and $\beta = 0.4$.

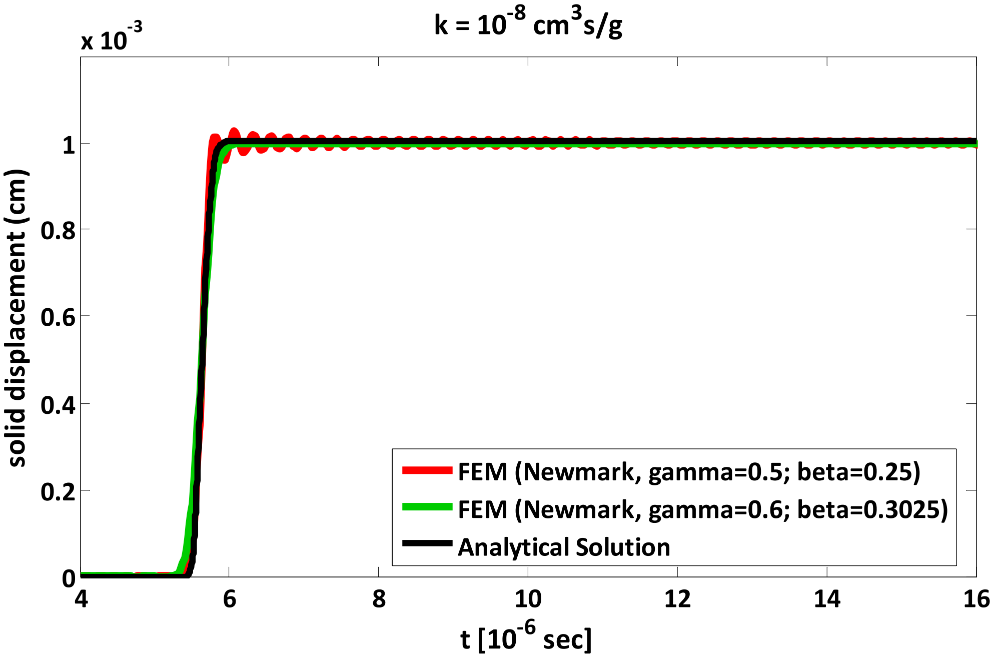


Figure 310.29: Time history of solid displacements of longitudinal waves at 1 cm below the surface. Comparison of numerical results (FEM) with the analytical solution by Gajo and Mongiovi (1995) for the case of viscous coupling ($k = 10^{-8} cm^3 s/g$). Two different sets of Newmark parameters were used for the numerical analysis.

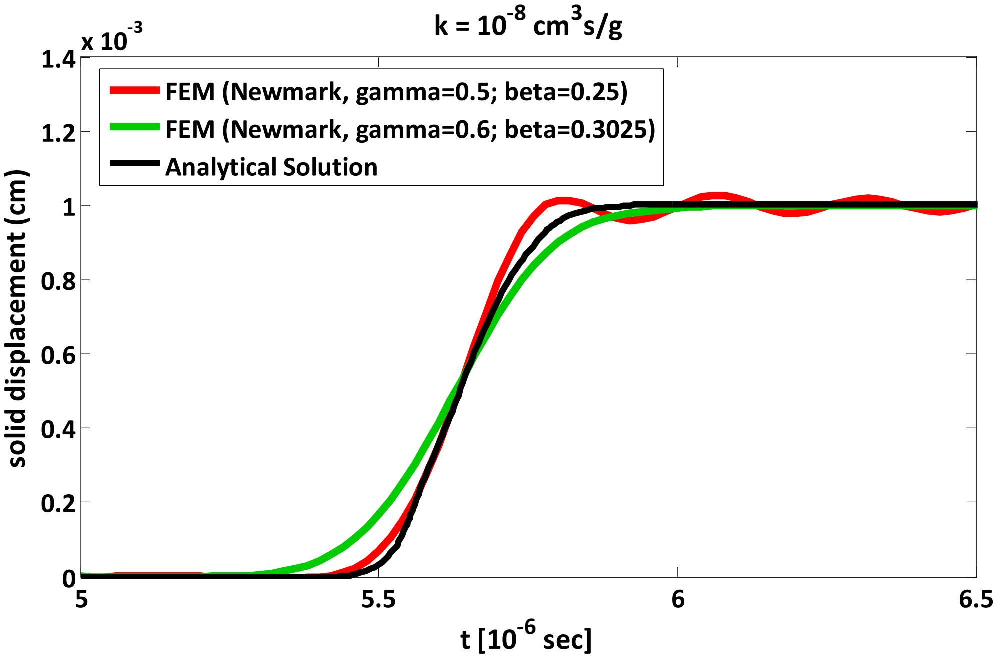


Figure 310.30: A magnified view of Figure 310.29 illustrating the details of wave front of the longitudinal wave of first kind.

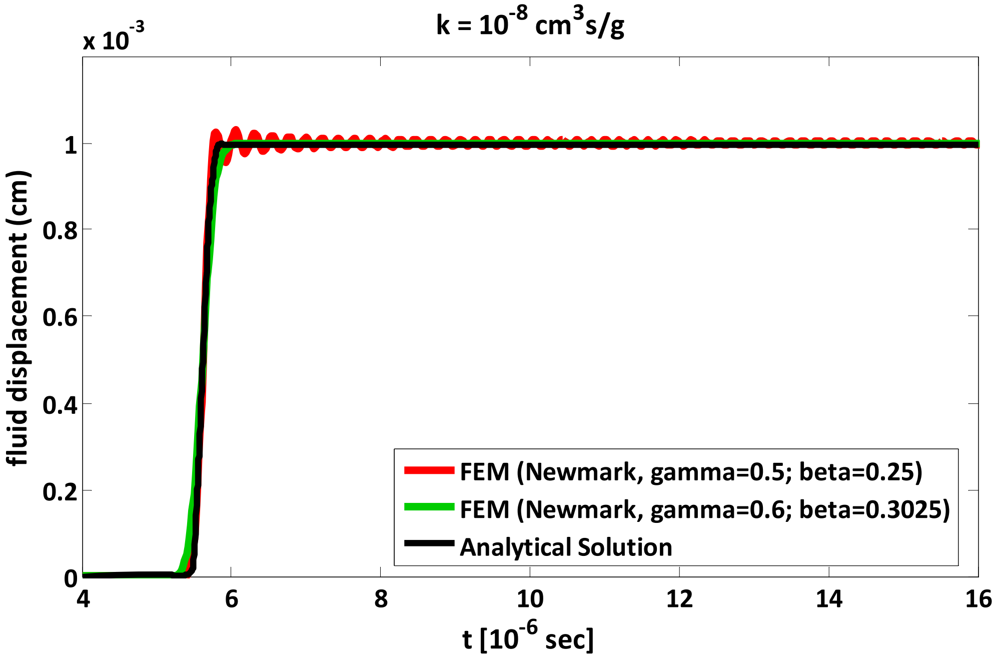


Figure 310.31: Time history of fluid displacements of longitudinal waves at 1 cm below the surface. Comparison of numerical results (FEM) with the analytical solution by Gajo and Mongiovi (1995) for the case of viscous coupling ($k = 10^{-8} cm^3 s/g$). Two different sets of Newmark parameters were used for the numerical analysis.

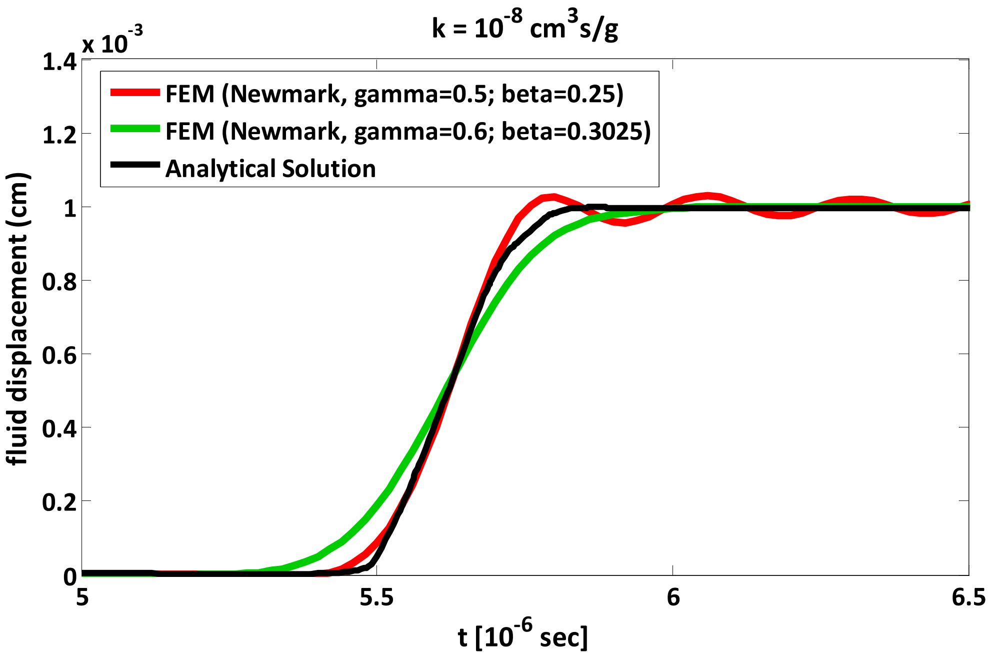


Figure 310.32: A magnified view of Figure 310.31 illustrating the details of wave front of the longitudinal wave of first kind.

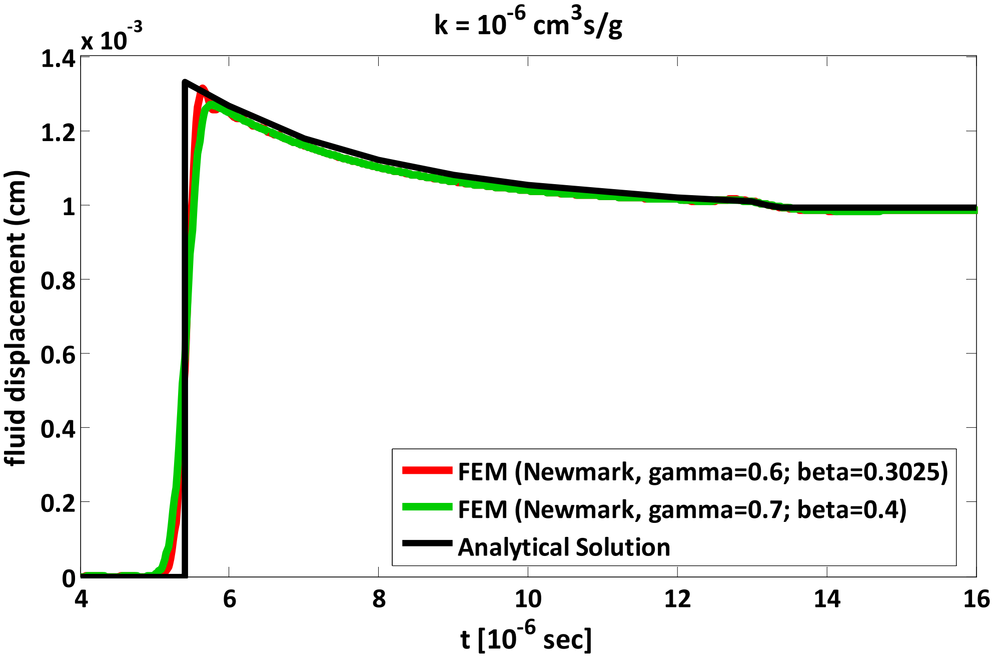


Figure 310.33: Time history of solid displacements of longitudinal waves at 1 cm below the surface. Comparison of numerical results (FEM) with the analytical solution by Gajo and Mongiovi (1995) for the case of viscous coupling ($k = 10^{-6} cm^3 s/g$). Two different sets of Newmark parameters were used for the numerical analysis.

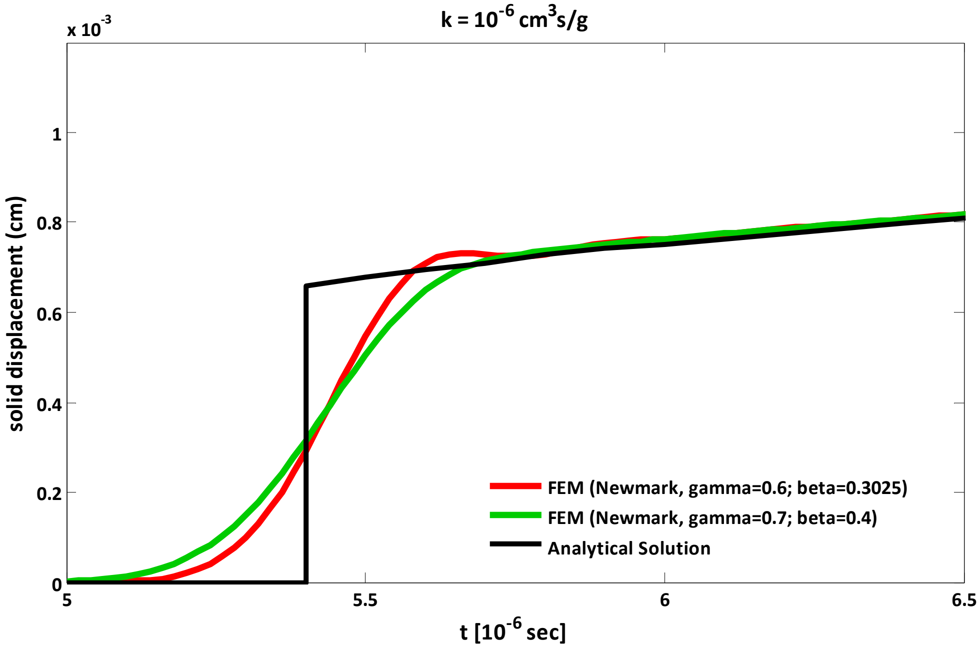


Figure 310.34: A magnified view of Figure 310.33 illustrating the details of wave front of the longitudinal wave of first kind.

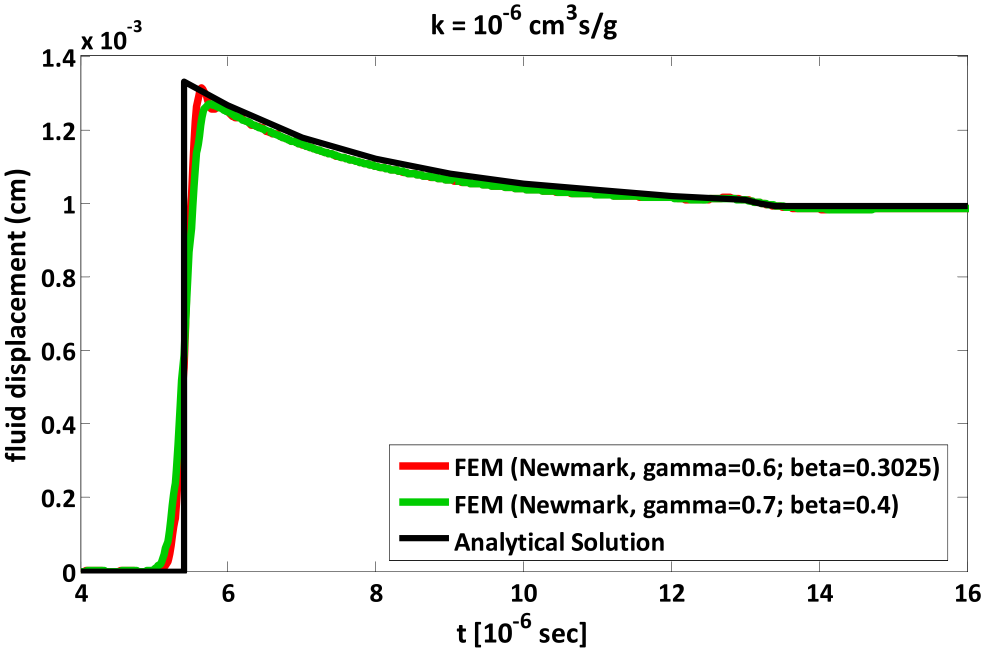


Figure 310.35: Time history of fluid displacements of longitudinal waves at 1 cm below the surface. Comparison of numerical results (FEM) with the analytical solution by Gajo and Mongiovi (1995) for the case of viscous coupling ($k = 10^{-6} cm^3 s/g$). Two different sets of Newmark parameters were used for the numerical analysis.

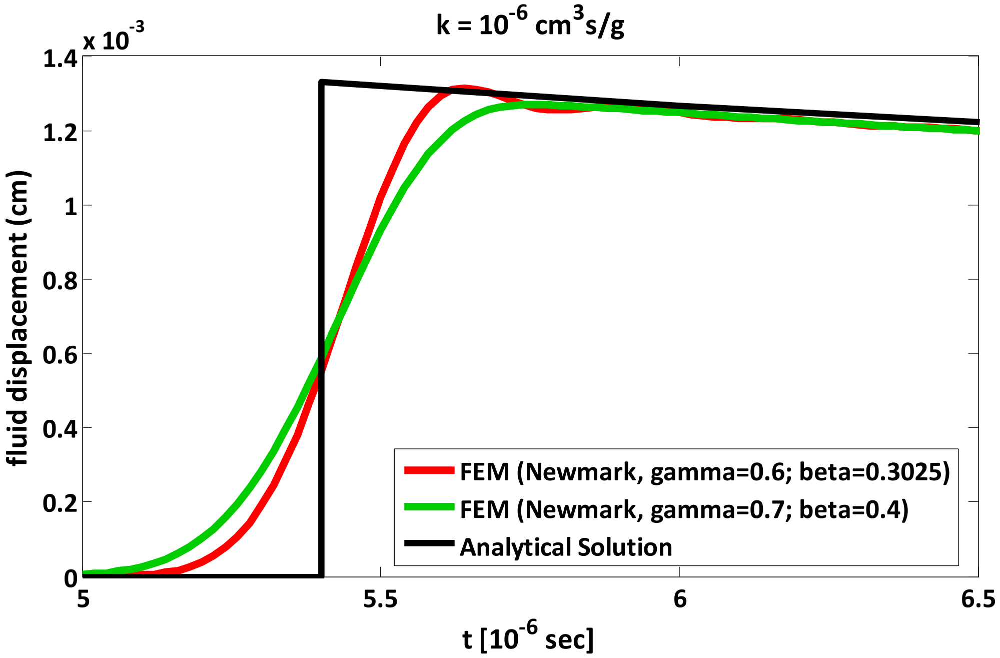


Figure 310.36: A magnified view of Figure 310.35 illustrating the details of wave front of the longitudinal wave of first kind.

For the HHT integrator (see Figures 310.45 - 310.56), the following sets of parameters were used:

a) $\alpha = -0.1$, $\gamma = 0.6$ and $\beta = 0.3025$,

b) $\alpha = -0.3$, $\gamma = 0.8$ and $\beta = 0.4225$.

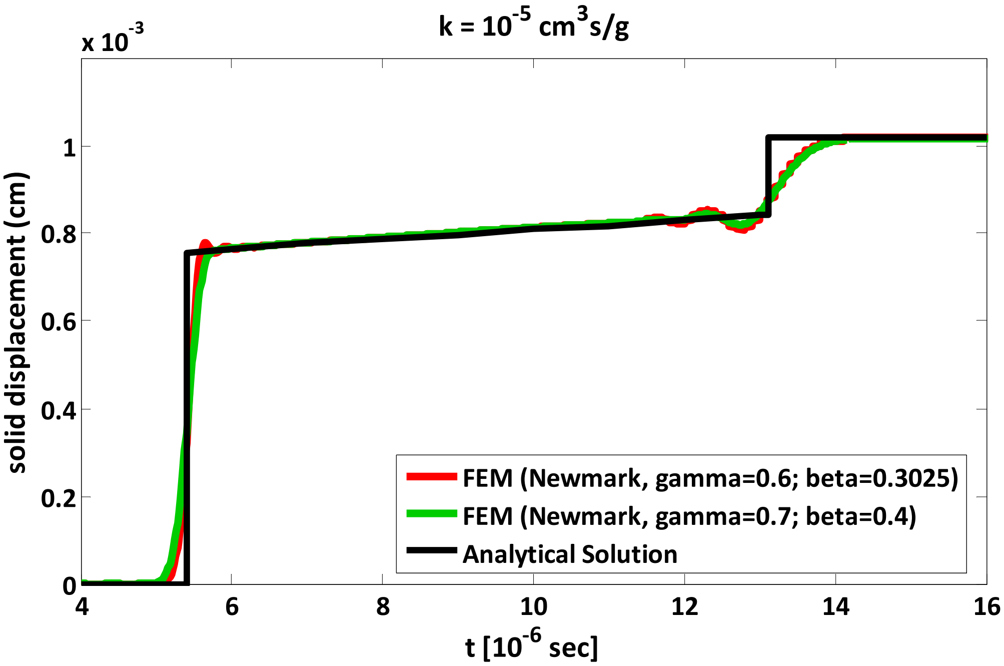


Figure 310.37: Time history of solid displacements of longitudinal waves at 1 cm below the surface. Comparison of numerical results (FEM) with the analytical solution by Gajo and Mongiovi (1995) for the case of viscous coupling ($k = 10^{-5} cm^3 s/g$). Two different sets of Newmark parameters were used for the numerical analysis.

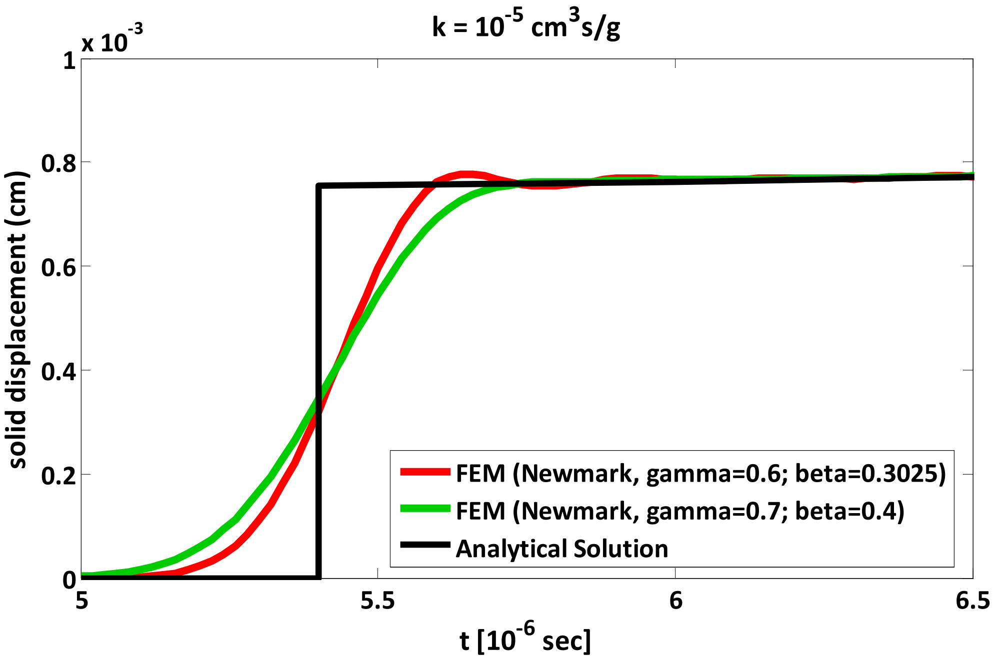


Figure 310.38: A magnified view of Figure 310.37 illustrating the details of wave front of the longitudinal wave of first kind.

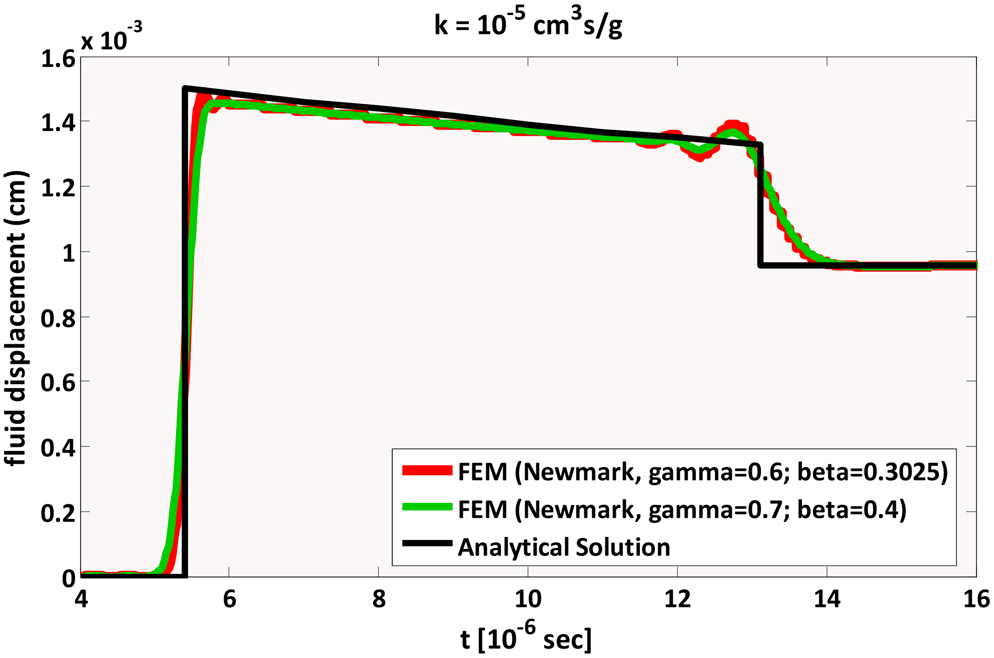


Figure 310.39: Time history of fluid displacements of longitudinal waves at 1 cm below the surface. Comparison of numerical results (FEM) with the analytical solution by Gajo and Mongiovi (1995) for the case of viscous coupling ($k = 10^{-5} cm^3 s/g$). Two different sets of Newmark parameters were used for the numerical analysis.

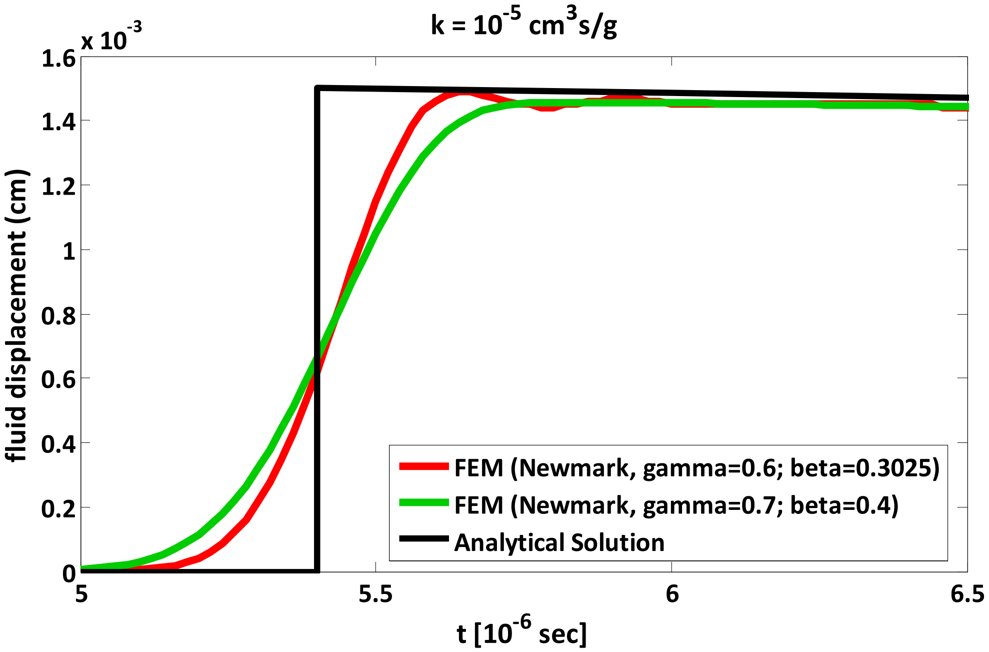


Figure 310.40: A magnified view of Figure 310.39 illustrating the details of wave front of the longitudinal wave of first kind.

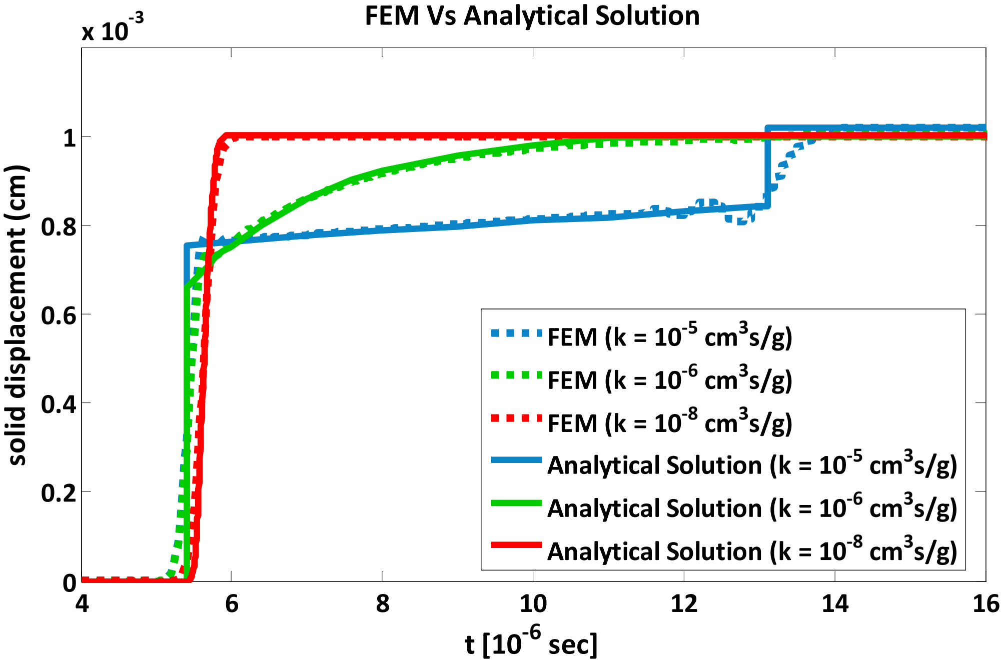


Figure 310.41: Time history of solid displacements of longitudinal waves at 1 cm below the surface. Comparison of numerical results (FEM) with the analytical solution by Gajo and Mongiovi (1995) for three different values of viscous coupling. The Newmark set of parameters used for the numerical solution was: $\gamma = 0.6$ and $\beta = 0.3025$.

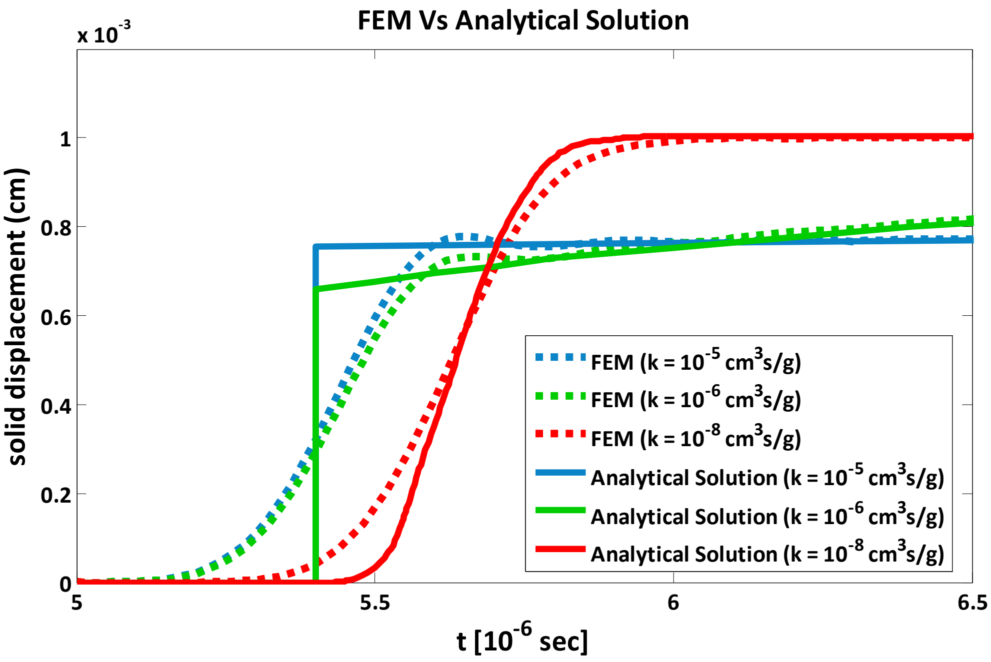


Figure 310.42: A magnified view of Figure 310.41 illustrating the details of wave front of the longitudinal wave of first kind.

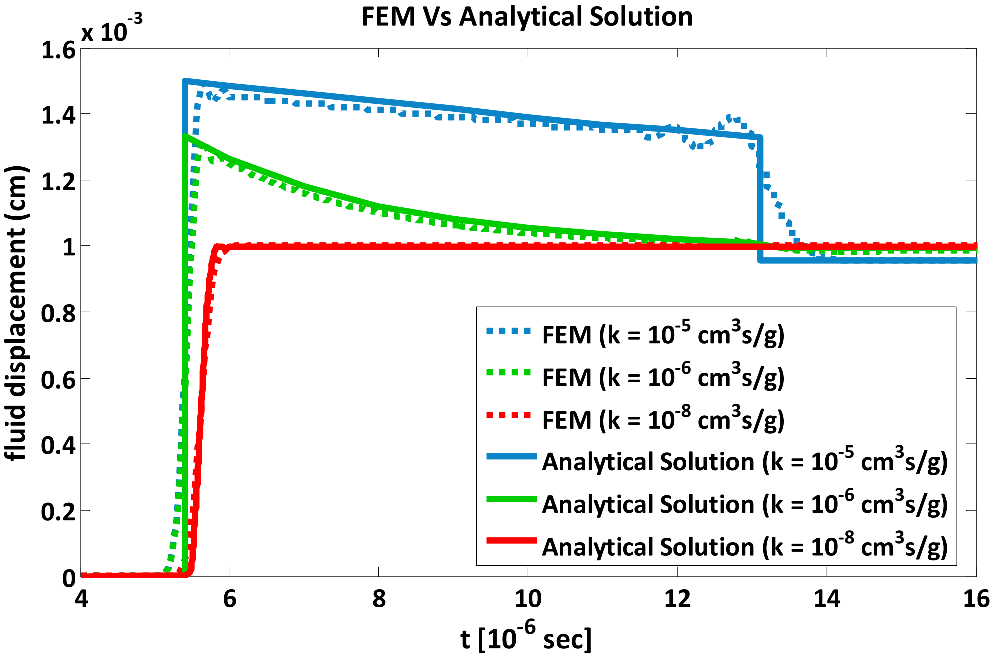


Figure 310.43: Time history of fluid displacements of longitudinal waves at 1 cm below the surface. Comparison of numerical results (FEM) with the analytical solution by Gajo and Mongiovi (1995) for three different values of viscous coupling. The Newmark set of parameters used for the numerical solution was: $\gamma = 0.6$ and $\beta = 0.3025$.

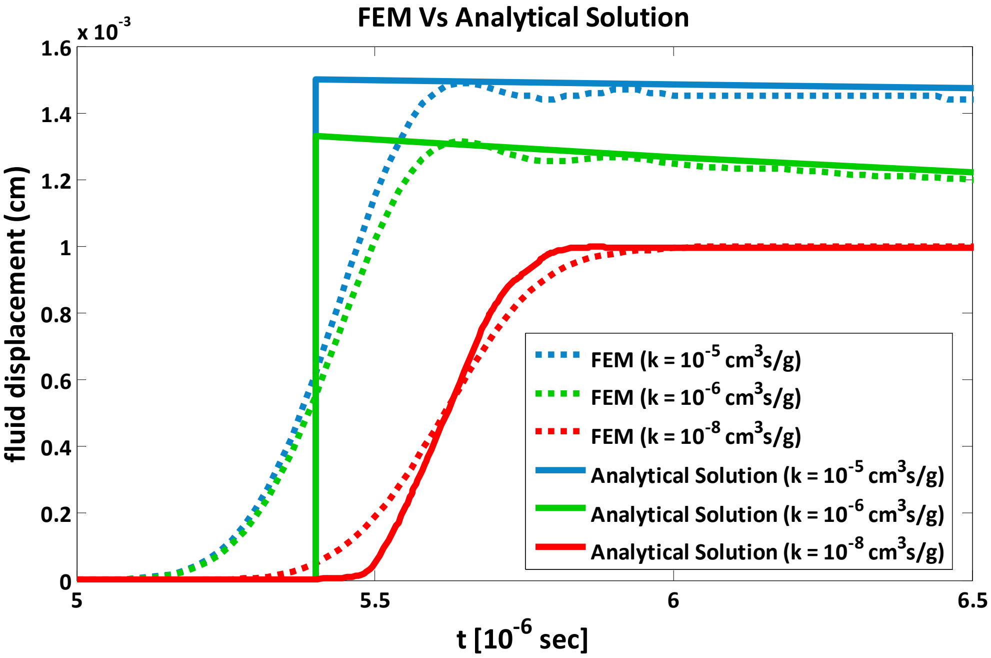


Figure 310.44: A magnified view of Figure 310.43 illustrating the details of wave front of the longitudinal wave of first kind.

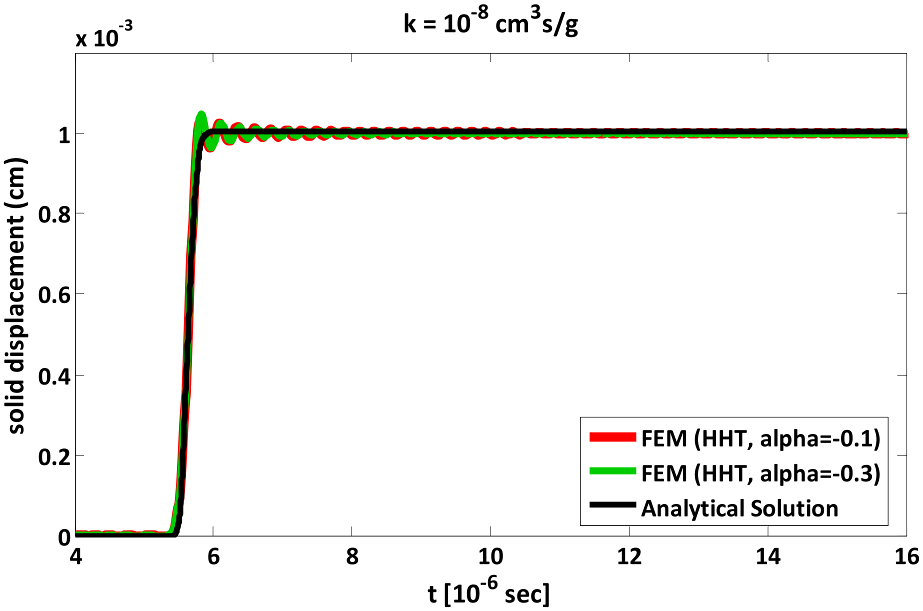


Figure 310.45: Time history of solid displacements of longitudinal waves at 1 cm below the surface. Comparison of numerical results (FEM) with the analytical solution by Gajo and Mongiovi (1995) for the case of viscous coupling($k = 10^{-8} cm^3 s/g$). Two different sets of unconditional stable HHT parameters were used for the numerical analysis.

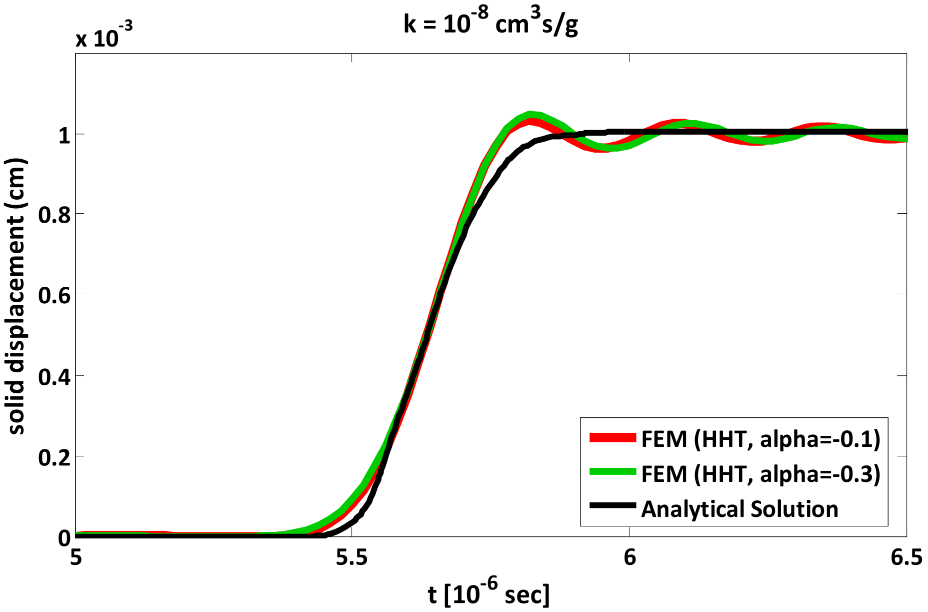


Figure 310.46: A magnified view of Figure 310.45 illustrating the details of wave front of the longitudinal wave of first kind.

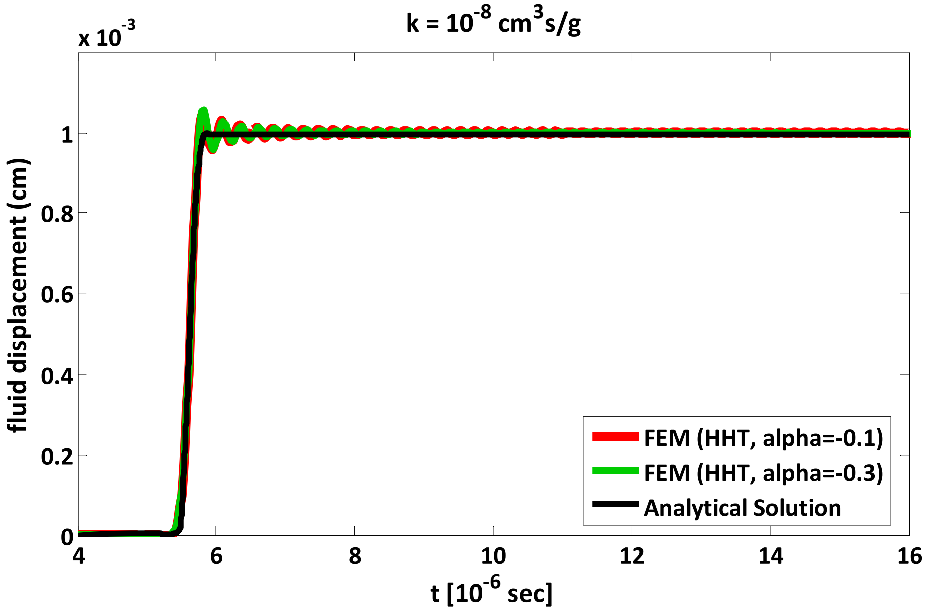


Figure 310.47: Time history of fluid displacements of longitudinal waves at 1 cm below the surface. Comparison of numerical results (FEM) with the analytical solution by Gajo and Mongiovi (1995) for the case of viscous coupling($k = 10^{-8} cm^3 s/g$). Two different sets of unconditional stable HHT parameters were used for the numerical analysis.

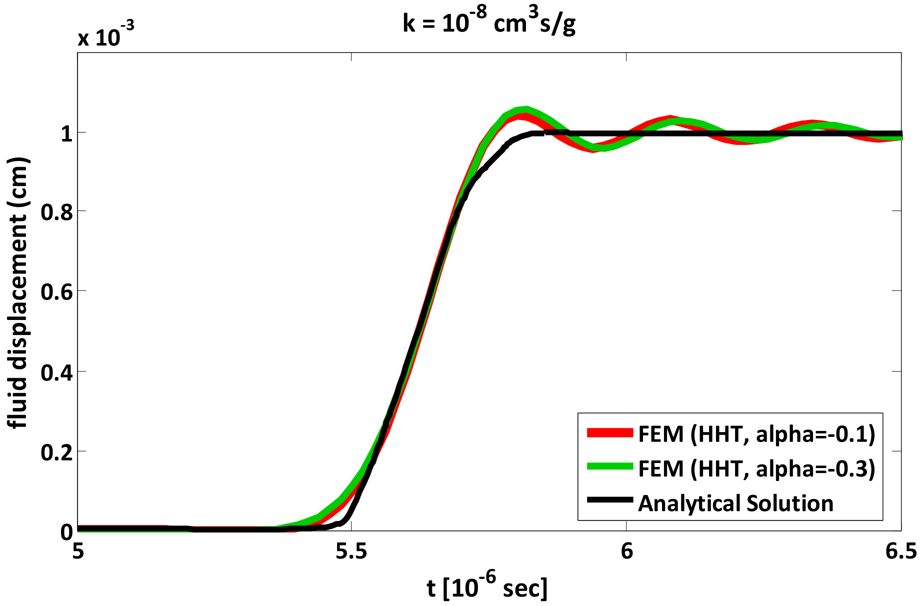


Figure 310.48: A magnified view of Figure 310.47 illustrating the details of wave front of the longitudinal wave of first kind.

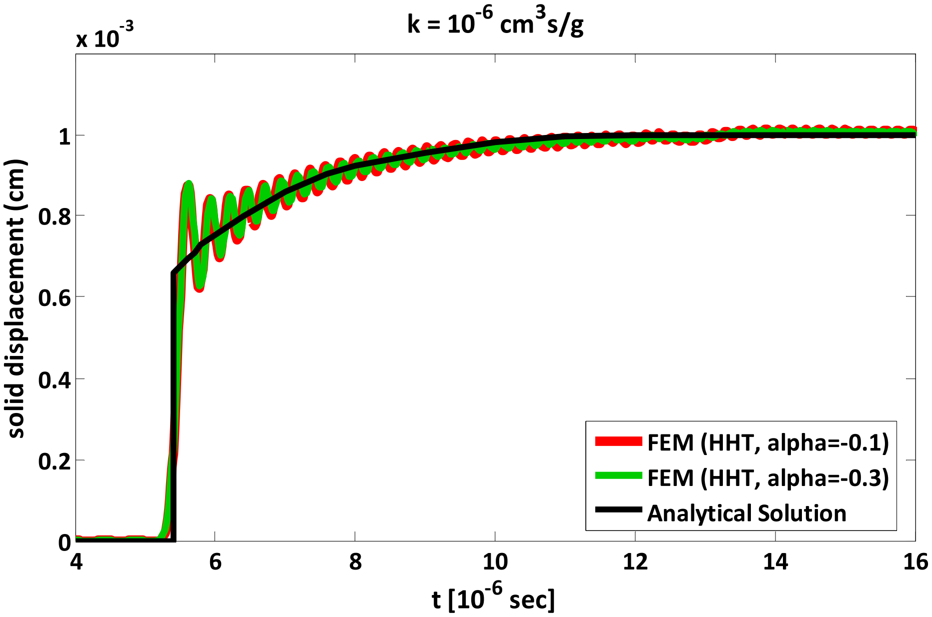


Figure 310.49: Time history of solid displacements of longitudinal waves at 1 cm below the surface. Comparison of numerical results (FEM) with the analytical solution by Gajo and Mongiovi (1995) for the case of viscous coupling($k = 10^{-6} cm^3 s/g$). Two different sets of unconditional stable HHT parameters were used for the numerical analysis.

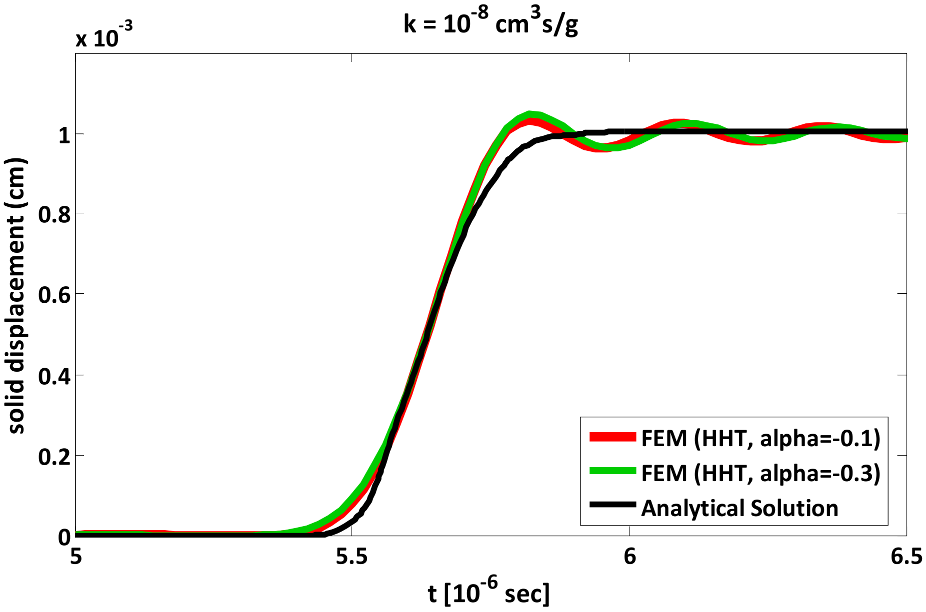


Figure 310.50: A magnified view of Figure 310.49 illustrating the details of wave front of the longitudinal wave of first kind.

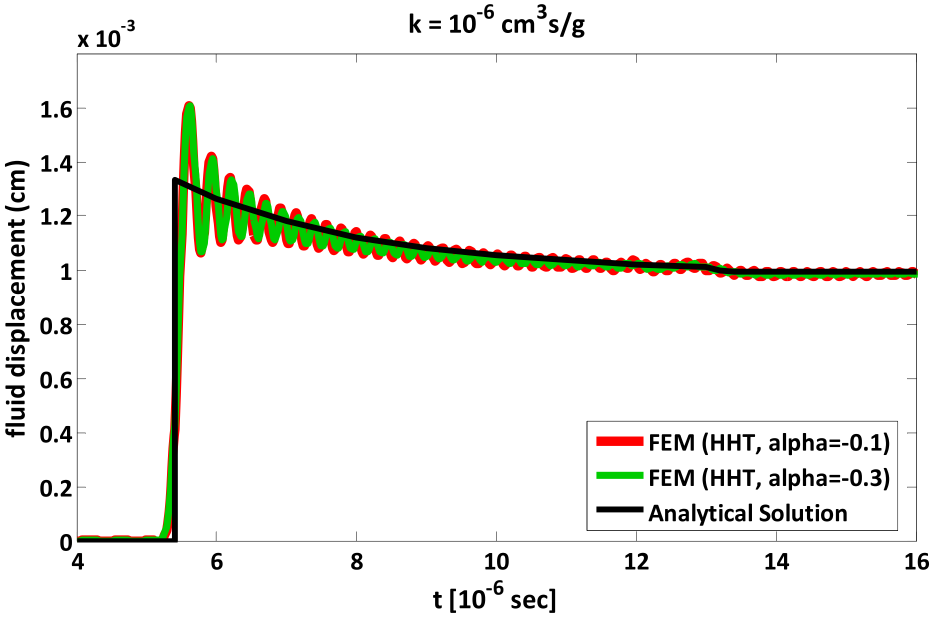


Figure 310.51: Time history of fluid displacements of longitudinal waves at 1 cm below the surface. Comparison of numerical results (FEM) with the analytical solution by Gajo and Mongiovi (1995) for the case of viscous coupling($k = 10^{-6} cm^3 s/g$). Two different sets of unconditional stable HHT parameters were used for the numerical analysis.

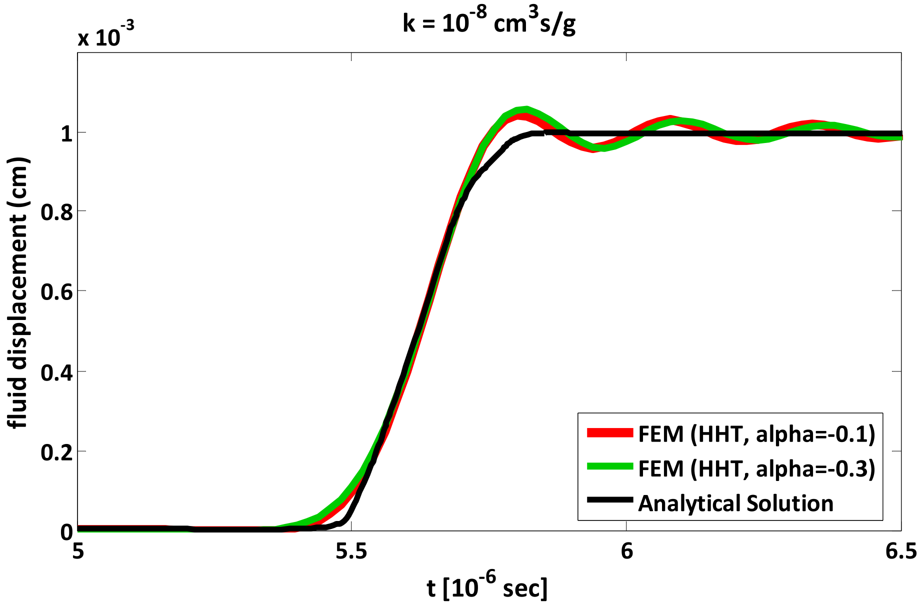


Figure 310.52: A magnified view of Figure 310.51 illustrating the details of wave front of the longitudinal wave of first kind.

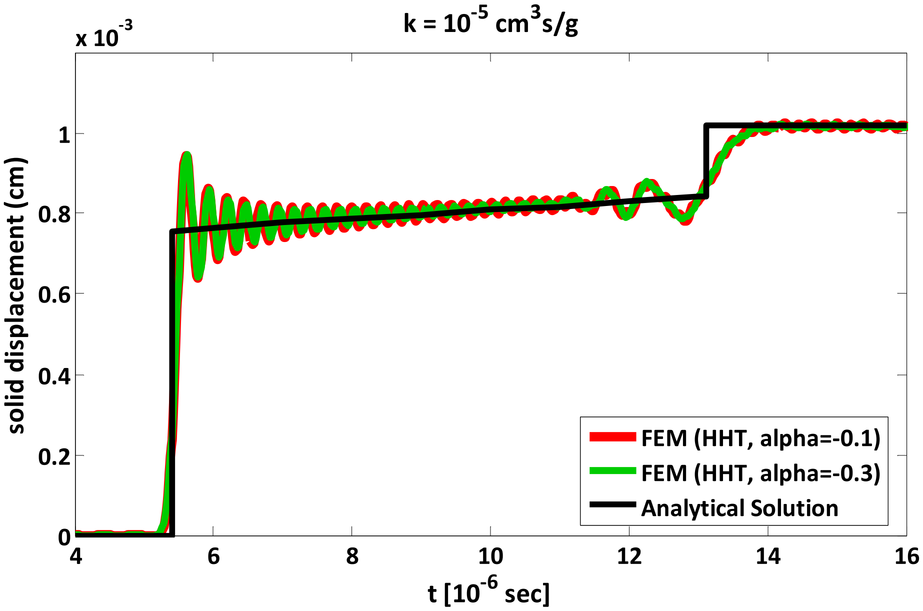


Figure 310.53: Time history of solid displacements of longitudinal waves at 1 cm below the surface. Comparison of numerical results (FEM) with the analytical solution by Gajo and Mongiovi (1995) for the case of viscous coupling($k = 10^{-5} cm^3 s/g$). Two different sets of unconditional stable HHT parameters were used for the numerical analysis.

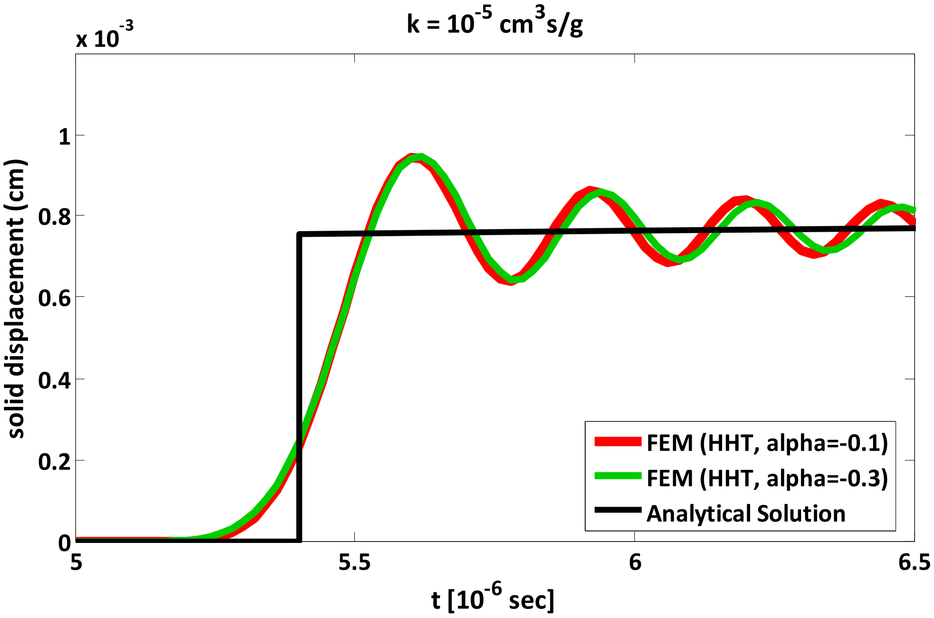


Figure 310.54: A magnified view of Figure 310.53 illustrating the details of wave front of the longitudinal wave of first kind.

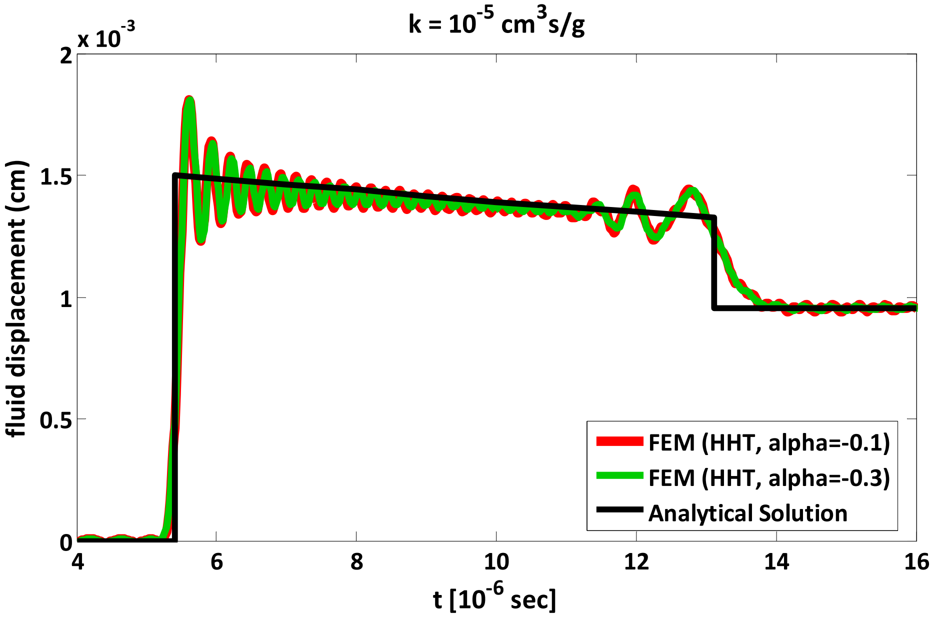


Figure 310.55: Time history of fluid displacements of longitudinal waves at 1 cm below the surface. Comparison of numerical results (FEM) with the analytical solution by Gajo and Mongiovi (1995) for the case of viscous coupling($k = 10^{-5} cm^3 s/g$). Two different sets of unconditional stable HHT parameters were used for the numerical analysis.

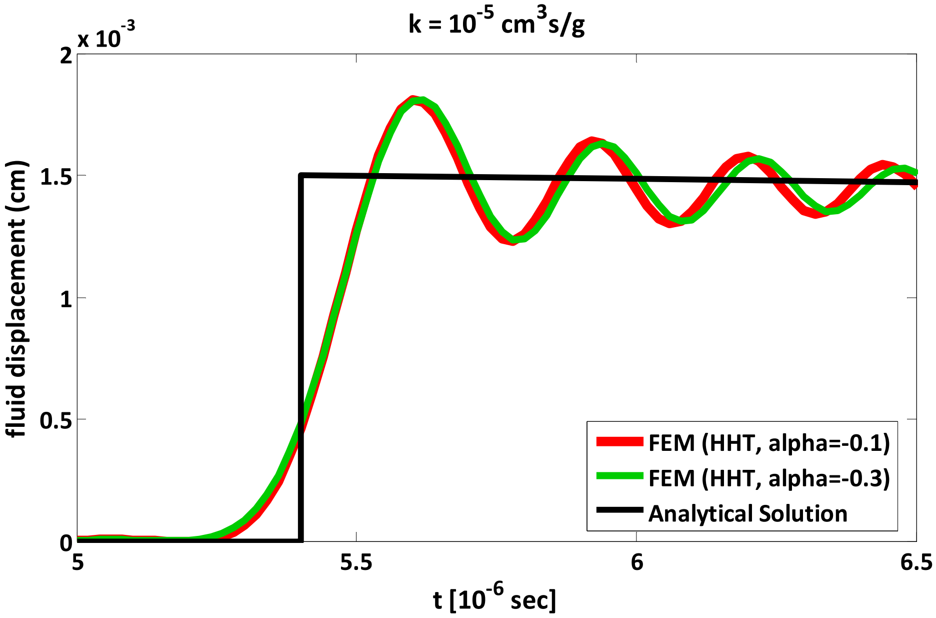


Figure 310.56: A magnified view of Figure 310.55 illustrating the details of wave front of the longitudinal wave of first kind.

310.8.3 Discussion of Numerical Results - Conclusions

Biot has shown than when dissipation is present, each frequency component propagated with its own velocity. Thus, especially in the case of numerical solutions using a finite element procedure, the response is very sensitive to the numerical damping introduced to the system. Generally, a drawback of all types types of numerical solutions is the distortion and the smearing of the wave fronts, which are linked to the highest frequency that is allowed by the computational grid and the numerical damping due to the time integration method. The numerical results presented here, show a larger rise time than the analytical solution, that it could potentially be improved by using a finer spatial and temporal discretization.

In particular, the dissipation of high frequency oscillations is achieved more efficiently by the Newmark integrator that the HHT one. Due to the fact that the filtering of high frequencies is less in case of HHT integrator, the smearing of the wave front of the first kind of longitudinal waves is not so extensive as in the Newmark case. Obviously, for both cases, as the numerical damping increases by changing the sets of parameters, the rise time of the water fronts increases too. It is also worth mentioning that the rise time of the wave front of the second kind low-frequency longitudinal wave is even longer than that of the first kind.

Figures 310.41 and 310.44 illustrate the comparative results for all the three different values of viscous coupling using the Newmark integration method. In general, it is worth noting that the numerical results are in good agreement with the main characteristics of the mechanics of dispersive wave propagation in fully saturated, porous media, as indicated by the analytical results. For example, numerical results well demonstrate that during the propagation of the first wave, the solid and fluid displacement are in phase with each other, whereas during the propagation of second wave, the displacements of the two phases are in opposition. Overall, the finite element solutions reproduce correctly the forms of wave propagation for a wide range of permeability.

310.9 One dimensional wave propagation in elastic porous media subjected to step loading at the surface

310.9.1 Brief review of Analytical Solution by de Boer et al. (1993)

An analytical solution for the one-dimensional transient wave propagation fluid-saturated elastic porous media is provided by de Boer et al. 1993. The fluid-saturated porous material is modeled as a two phase system composed of an incompressible solid phase and an incompressible fluid phase. An exact analytical solution is obtained via Laplace transform technique considering initial and boundary conditions, which exhibits only one independent compressive wave in both the solid and fluid phases, as a result of the incompressibility constraint.

The problem configuration, which the analytical solution is addressed to, consists of an one-dimensional infinitely long column, separated from the half-space of a fluid-saturated porous elastic skeleton material. The motion of both the solid and the fluid materials is constrained to occur in the vertical direction. Loading as a function of time, $\sigma(z = 0, t) = f(t)$, is applied to the half space surface boundary by a permeable punch with ideal permeability. Homogeneous pore distribution and free pore fluid surface are assumed. The wave motion in the porous medium is expressed by the solid and fluid displacements or the solid extra stresses, respectively, but it cannot be expressed by the pore pressure which is just the Lagrangian multiplier corresponding to the incompressibility constraint of the medium.

In particular, in the paper by de Boer et al. (1993), the solid and the fluid displacements, the solid skeleton extra stresses and the pore pressure are given with respect to time and with respect to different depths in the soil column within the framework of three loading forms: i) sinusoidal, ii) step loading and iii) impulsive loading. These results can be taken for a quantitative comparison to various numerical solutions.

310.9.2 Numerical Analysis

Numerical example for the step loading case was solved in order to verify the previously mentioned $u-p-U$ formulation. The numerical model used for the simulation of the 1C shock wave propagation consists of 1000 u-p-U brick finite elements of dimensions $1cm \times 1cm \times 1cm$ creating a soil column $10m$ thick. Obviously, the numerical simulation of a semi infinite soil column is not possible; thus, a soil column of thickness of 10 cm was considered adequate for the current problem configuration. Figure 310.57 illustrates the transition from the physical configuration of the problem to its numerical simulation. Table 310.7 shows the soil properties of the numerical model, which are the same with those used for the analytical results presented in the paper by de Boer et al. (1993).

The only difference is noted on the elastic modulus, which was selected to be $20MN/m^2$ for the numerical solution (FEM) instead of $30MN/m^2$, as it is mentioned in the previously mentioned paper. This is due to the fact that the numerical results indicated that the results given in the paper correspond to a soil column with elastic modulus equal to $20MN/m^2$ instead of $30MN/m^2$. Moreover, it should be mentioned that the solid and fluid compressibility were given realistic values (see Table 310.7), which practically means that the two constituents are incompressible.

At the top surface of the soil column, a step loading of $\sigma(z = 0, t) = 3kN/m^2$ is applied to the solid part, as a nodal load equally distributed to the four top nodes. The nodal load is expressed as:

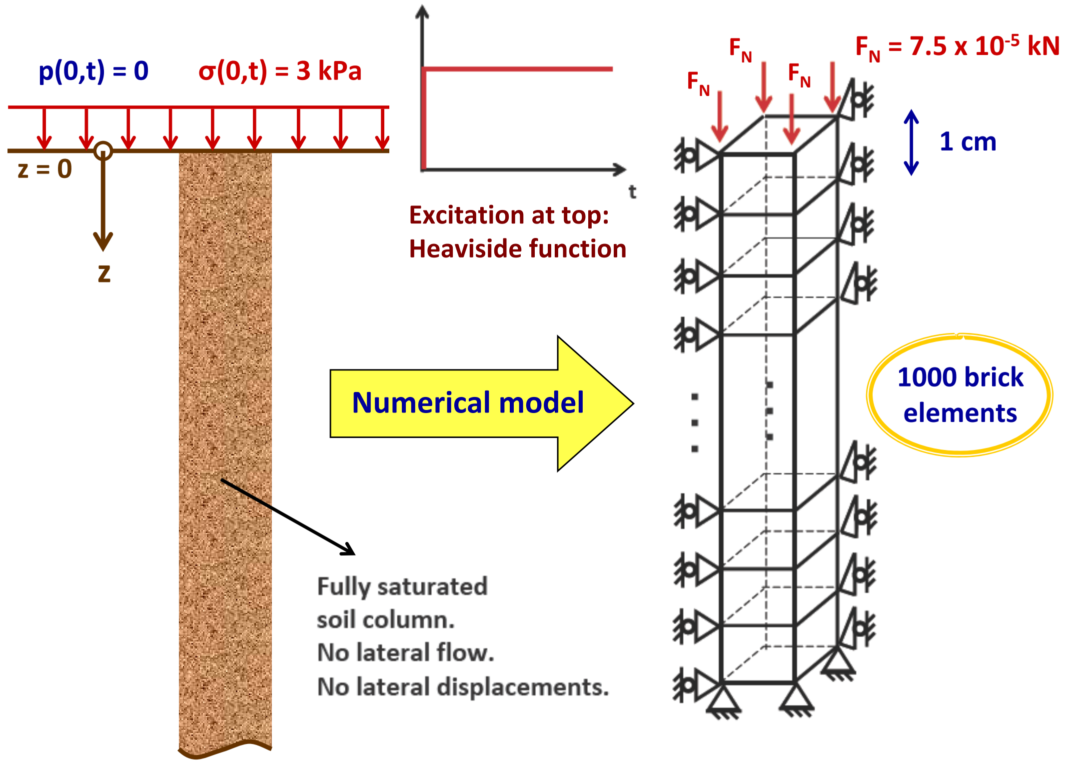


Figure 310.57: The numerical model used for the verification of the finite element implementation through comparison with the analytical solution provided by de Boer et al. (1993).

$$F_N(z=0,t) = \frac{\sigma(z=0,t)\times A}{4} = \frac{3kN/m^2 \times 0.01m \times 0.01m}{4} = 7.5 \times 10^{-5} kN \tag{310.57}$$

Only the vertical displacement is free. There is no lateral flow or displacement. The degree of freedom related to the pore pressure is constrained at the top surface to be equal to the atmospheric pressure, while it is free at the rest of the nodes. The base of the model is rigid and impervious.

This kind of excitation (Heaviside function) applied at the top of the model, results clearly in waves of all kinds of frequency, first due to its nature and secondarily due to the way of its application. Due to this fact, a fairly dense mesh of 1000 u-p-U brick finite elements of dimensions $1cm \times 1cm \times 1cm$ was chosen. The time step, δt required needs to be limited to

$$\delta t < \frac{\delta h}{v} \tag{310.58}$$

$$\tag{310.59}$$

Table 310.7: Soil Properties for 1C shock wave propagation for the problem by de Boer et al. (1993)

Parameter	Symbol	Value
gravity acceleration	g	$9.81\ m/s^2$
soil matrix Young's Modulus	E	$20 \times 10^3\ kN/m^2$
soil matrix Poisson's ratio	ν	0.2
soil matrix Lame's constant	λ	$5.55 \times 10^3\ kN/m^2$
soil matrix shear modulus	μ	$8.33 \times 10^3\ kN/m^2$
solid particle density	ρ_s	$2.0 \times 10^3\ kg/m^3$
water density	ρ_f	$1.0 \times 10^3\ kg/m^3$
solid particle bulk modulus	K_s	$36.0 \times 10^6\ kN/m^2$
fluid bulk modulus	K_f	$2.177 \times 10^6\ kN/m^2$
porosity	n	0.33
Darcy's permeability	k_D	$0.01\ m/s$

where v is the highest wave velocity. In our case, the temporal integration involves 4000 steps of $1.0 \times 10^{-4} sec$, which allows a maximum wave velocity of $100 m/s$. The propagation velocity can be calculated by the following equation given by de Boer et al. (1993) and is equal to $90.7 m/s$.

$$v = \sqrt{\frac{n^2(\lambda\ +\ 2\mu)}{n^2(1-n)\rho_s\ +\ (1-n)^2(n\rho_f)}} = 90.7 m/s \tag{310.60}$$

The Newmark time integration method was used, which dissipates more efficiently the high frequencies introduced in the system due to numerics than HHT integrator, as shown in section 4.3. The following set of parameters was chosen, assuring unconditionally numerical stability: $\gamma = 0.7$ and $\beta = 0.4$.

310.9.3 Discussion of Numerical Results - Conclusions

Figures 310.58 to 310.67 illustrate the comparative results between analytical and numerical solution. In general, it is worth noting that the numerical results are in good agreement, with respect to time and with respect to depth, with those obtained by the analytical solution. The responses of the medium due to step loading are indicative of the consolidation process in case of a free pore water surface. The solid moves downwards, indicating that settlement occurs and the fluid is squeezed out from the pore volume creating an upward flow. During the consolidation process, the extra solid skeleton stresses increase with time at a certain depth. However, they decrease with the distance from the loading surface at a certain time. In opposition to the extra solid skeleton stresses, the pore pressure decreases with time tending to zero, while it increases with depth.

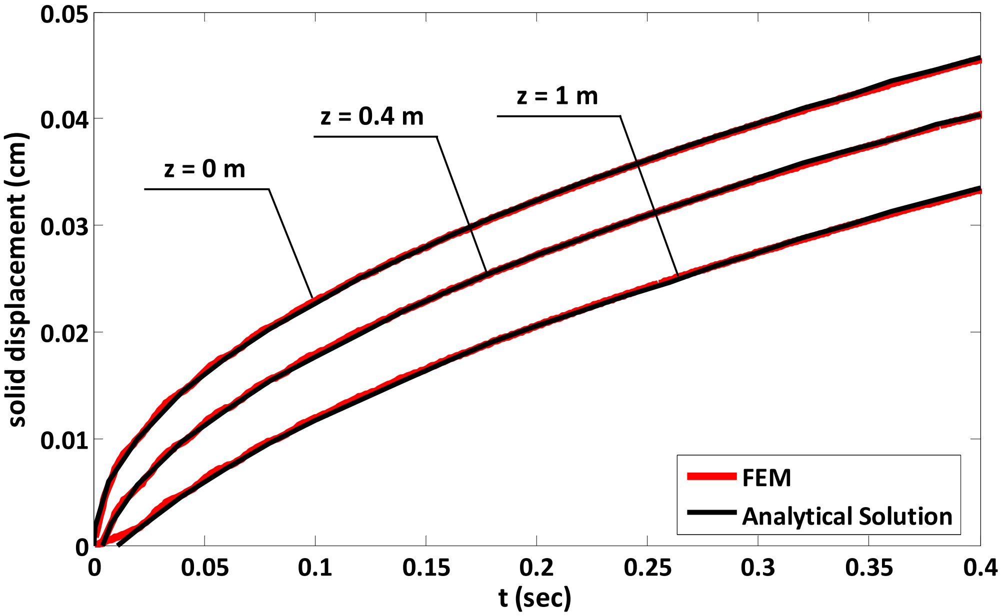


Figure 310.58: Time history of solid displacements at different depths due to step loading. Comparison of numerical results (FEM) with the analytical solution by de Boer et al. (1993).

Overall, it is worth mentioning that the results obtained from the finite element procedure practically coincide with the ones given by the analytical solution. The only difference is located to the pore pressure (see Figures 310.64 to 310.67), where numerical response is oscillatory in contrast to the analytical solution. This may be due to the high frequencies introduced to the system by the temporal and spatial discretization and/or the compressibility of the solid and fluid phases. It should be mentioned again, that in the analytical solution, the two constituents are assumed incompressible whereas in the numerical model, the solid and fluid bulk moduli have realistic values (see Table 310.7. That is why in Figure 310.68, numerical examples with different values of fluid compressibility were solved. It is obvious that the oscillations decrease as the the fluid becomes more and more compressible because the stiffness of the system decreases and the high frequencies are dissipated faster. Moreover, Figure 310.68 indicated that better quantitative agreement between the numerical and analytical solution is achieved when the bulk modulus of the fluid is 2.2×10^6 kPa - realisic value - instead of 2.2×10^9 kPa, which would be expected since the pore fluid is assumed to be incompressible in the framework of the analytical solution. Comparing the pore pressures obtained from these two cases, it can be observed that the pore pressure

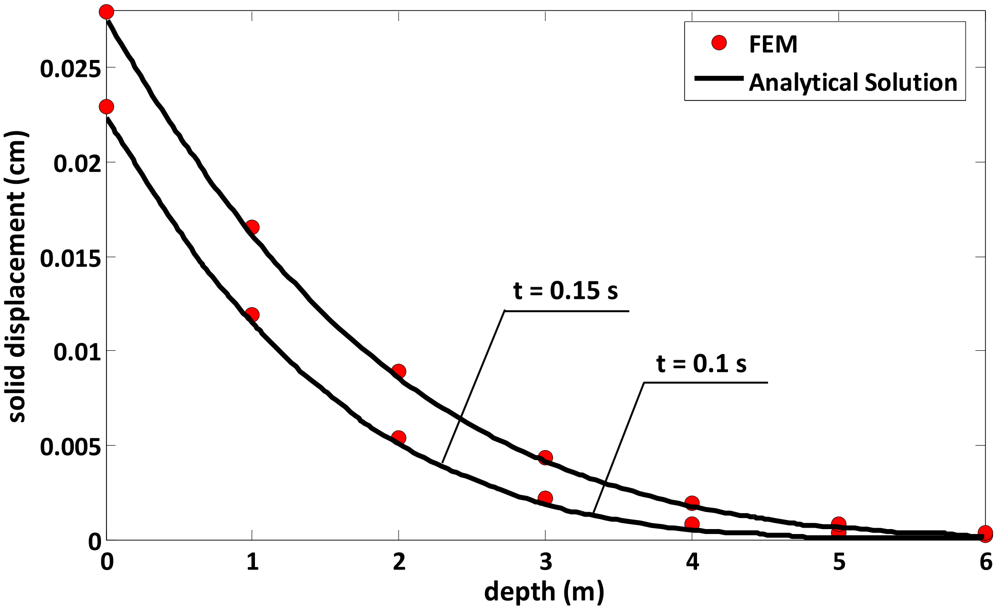


Figure 310.59: Response of solid displacements versus depth at different time moments due to step loading. Comparison of numerical results (FEM) with the analytical solution by de Boer et al. (1993).

generation in the more compressible fluid is slightly higher than that of the almost incompressible fluid because of the existence of the oscillatory waves, as mentioned by (Zienkiewicz and Shiomi, 1984).

310.10 One dimensional wave propagation in elastic porous media subjected to step velocity boundary condition

310.10.1 Brief review of Analytical Solution by Hiremath et al. (1988)

Hiremath et al. (1988) present a solution of Biot's dynamic equation of motion for one-dimensional wave propagation in a fluid-saturated linear elastic isotropic soil using Laplace transformation followed by numerical inversion. This study is considered to be an extension of the exact transient solution presented by Garg et al. (1974) for two limiting cases of infinitely small and infinitely large viscous coupling. In both cases, a soil column of finite dimension subjected to velocity boundary conditions was analyzed, allowing for reflection of waves at the boundaries.

In particular, Hiremath et al. (1988) examines two cases allowing for weak and strong viscous coupling, or else, as it is referred in the related paper, low and high drag, respectively. Moreover, two different

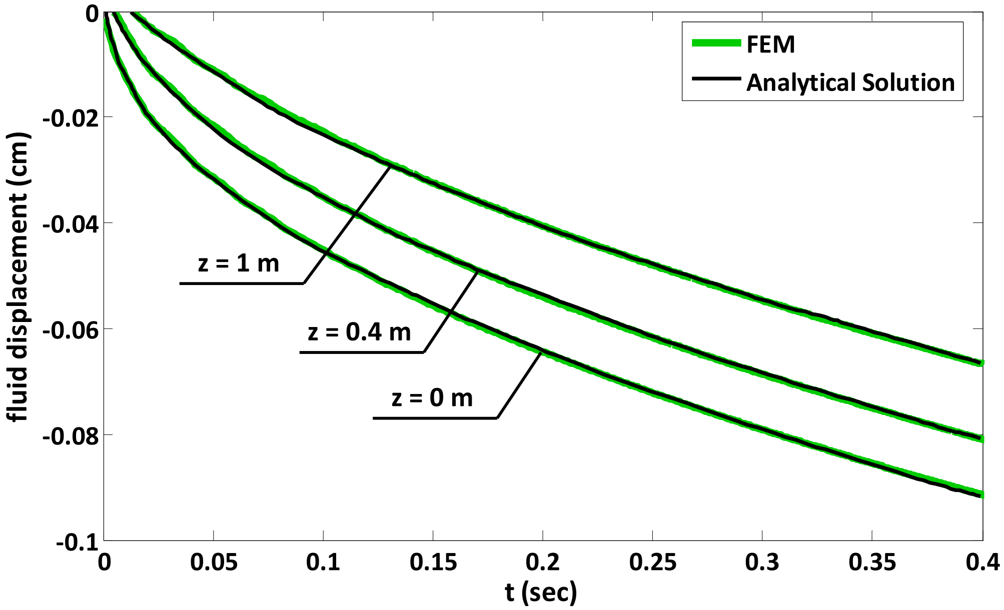


Figure 310.60: Time history of fluid displacements at different depths due to step loading. Comparison of numerical results (FEM) with the analytical solution by de Boer et al. (1993).

types of excitations were applied at the boundary surface in terms of solid and fluid velocity. In the first case, a unit step boundary condition boundary condition was applied at the top surface for both solid and fluid phases. In the second case, the fluid velocity specified at the boundary is different from the specified solid velocity increasing gradually to unity over the time scale. The results obtained from the numerical inversion allowed for six reflections of the fast compressional wave of first kind and two reflections of the secondary slow longitudinal wave.

One of the most important observations which both Garg et al. (1974) and Hiremath et al. (1988) concluded to, is that in case of strong viscous coupling (high drag), the material behaves as a single continuum with internal dissipation and the two wave fronts tend to become a single one.

Hiremath et al. (1988) presented a comparison of finite element solution of Biot's equations of motion with one based on numerical inversion of the Laplace transform solution. It is explained in the paper that a proper choice of element type and time domain integration is essential for capturing the results coming from the semi-analytical solution. Moreover, the spatial discretization needs to be combined with an appropriate temporal one, so that the wave does not traverse more than one element length during a single time step. In detail, Hiremath et al. (1988) suggests 100 linear elements of 0.005 m length and 986 time steps of size 10^{-6} sec.

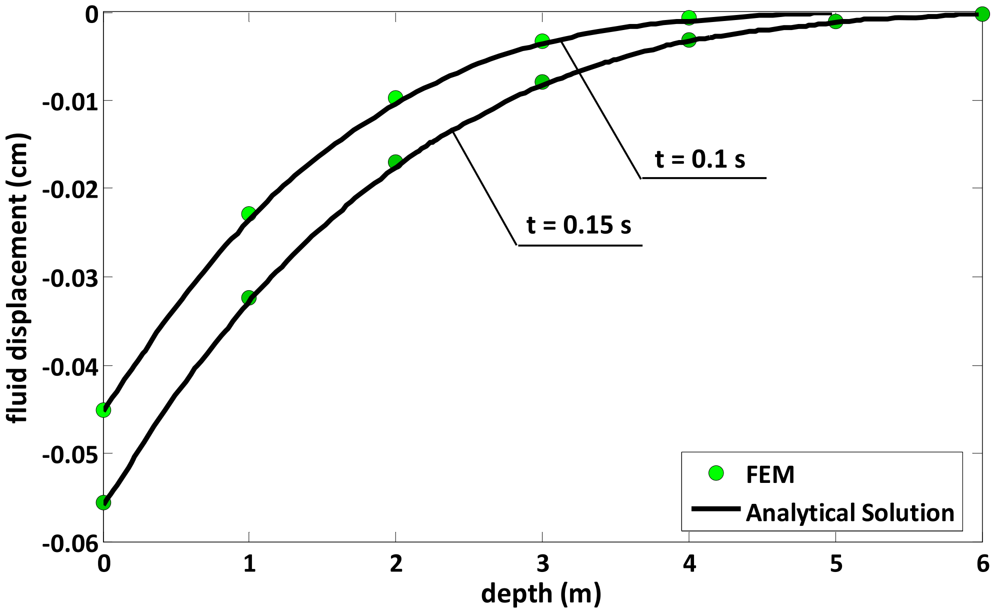


Figure 310.61: Response of fluid displacements versus depth at different time moments due to step loading. Comparison of numerical results (FEM) with the analytical solution by de Boer et al. (1993).

310.10.2 Numerical Analysis

Numerical examples for two extreme values of viscous coupling: a) high drag ($k = 0.148 \times 10^{-8} cm^3 s/g$) and b)low drag ($k = 0.148 \times 10^{-2} cm^3 s/g$), were solved in order to verify the previously mentioned $u-p-U$ formulation by comparing the results with the semi-analytical solution provided by Hiremath et al. (1988). The numerical model used for the simulation of the 1C shock wave propagation consists of 100 u-p-U brick finite elements of dimensions $0.005m \times 0.005m \times 0.005m$ creating a soil column 50cm thick. Figure 310.69 illustrates the transition from the physical configuration of the problem to its numerical simulation. Table 310.8 shows the soil properties of the numerical model.

At the top surface of the soil column, a step velocity of $1.0 \times 10^{-2} m/sec$ is applied both to the solid and the fluid phase. Only the vertical translational degrees of freedom are free. The horizontal translational degrees of freedom are constrained so that there is no lateral flow or displacement. The base of the model is rigid and impervious.

This kind of excitation (Heaviside function) applied at the top of the model, results clearly in waves of all kinds of frequency, first due to its nature and secondarily due to the way of its application. This fact together with the great stiffness of the solid skeleton (see Table 310.8 require a very dense spatial

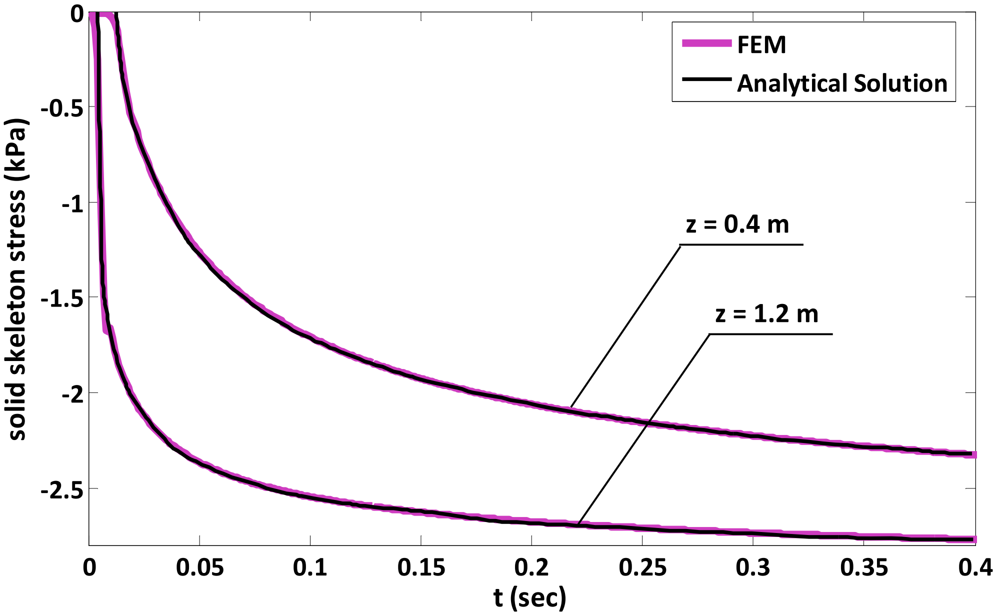


Figure 310.62: Time history of solid skeleton stresses at different depths due to step loading. Comparison of numerical results (FEM) with the analytical solution by de Boer et al. (1993).

Table 310.8: Soil Properties for 1C shock wave propagation for example by Hiremath et al. (1988).

Parameter	Symbol	Value
gravity acceleration	g	$9.81\ m/s^2$
soil matrix Young's Modulus	E	$23.21 \times 10^6\ kN/m^2$
soil matrix Poisson's ratio	ν	0.171
solid particle density	ρ_s	$2.66 \times 10^3\ kg/m^3$
water density	ρ_f	$1.0 \times 10^3\ kg/m^3$
solid particle bulk modulus	K_s	$36.0 \times 10^6\ kN/m^2$
fluid bulk modulus	K_f	$2.2 \times 10^6\ kN/m^2$
porosity	n	0.18
Biot coefficient	α	0.6772

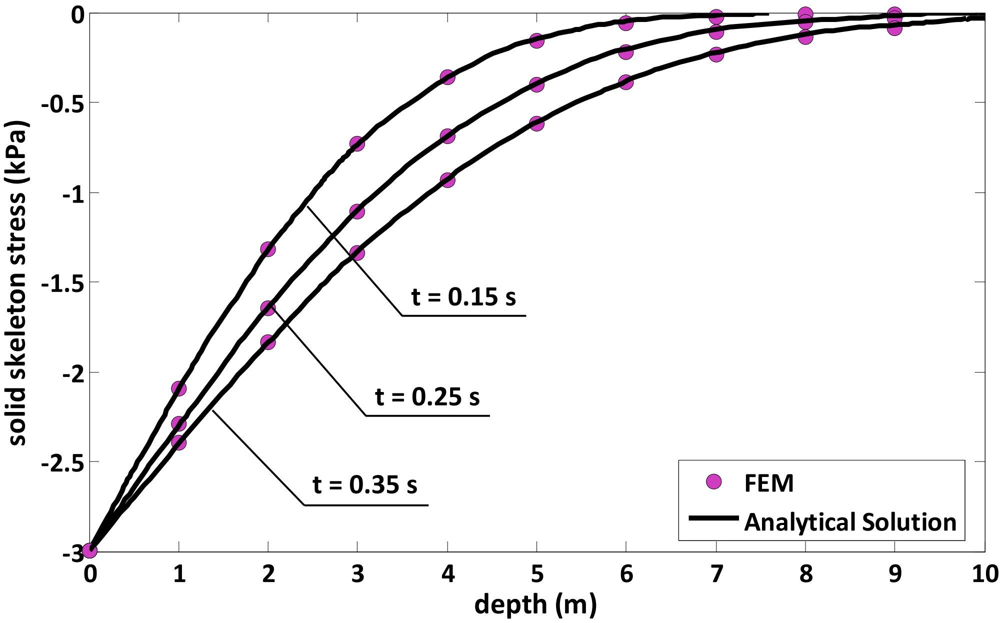


Figure 310.63: Response of solid skeleton stresses versus depth at different time moments due to step loading. Comparison of numerical results (FEM) with the analytical solution by de Boer et al. (1993).

discretization. Here, 100 u-p-U brick finite elements of dimensions $0.005m \times 0.005m \times 0.005m$ were chosen, following similar discretization with Hiremath et al. (1988). The time step, δt required needs to be limited to

$$\delta t < \frac{\delta h}{v} \tag{310.61}$$

(310.62)

where v is the highest wave velocity. In our case, the temporal integration involves 1972 steps of $5.0 \times 10^{-7} sec$, in comparison with 986 time steps of size $10^{-6} sec$, used by Hiremath et al. (1988). The time integration method used was the Newmark integrator with parameters: $\gamma = 0.6$ and $\beta = 0.3025$.

310.10.3 Discussion of Numerical Results - Conclusions

Biot has shown than when dissipation is present, each frequency component propagates with its own velocity. Thus, especially in the case of numerical solutions using a finite element procedure, the response is very sensitive to the numerical damping introduced to the system. Generally, a drawback of all types

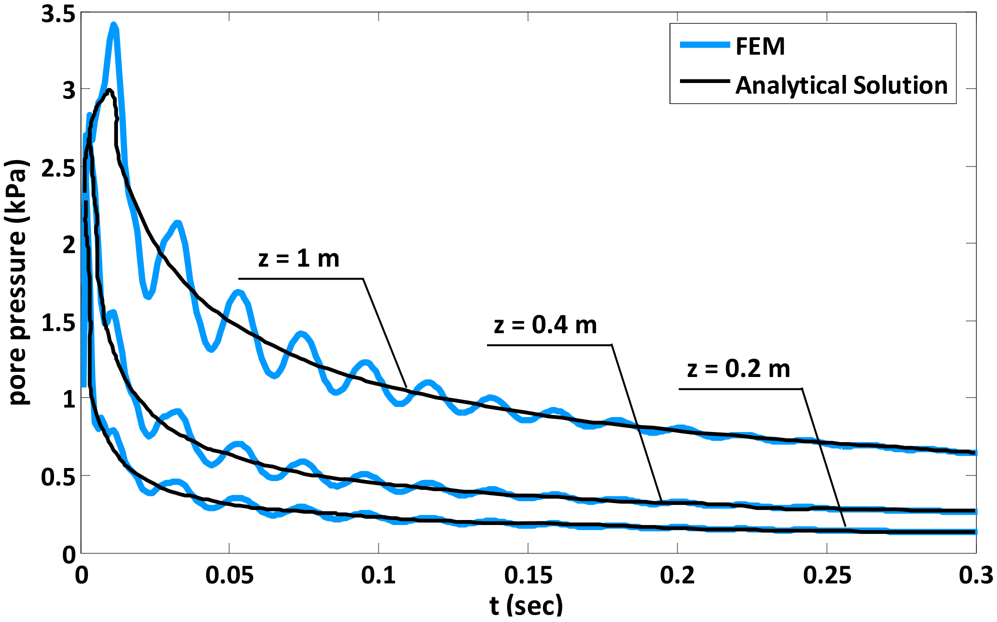


Figure 310.64: Time history of pore pressure at different depths due to step loading. Comparison of numerical results (FEM) with the analytical solution by de Boer et al. (1993).

types of numerical solutions is the distortion and the smearing of the wave fronts, which are linked to the highest frequency that is allowed by the computational grid and the numerical damping due to the time integration method. The numerical results presented here, show some oscillations at the rough changes in velocity due to reflection of wave fronts that could be possibly diminished by using a finer spatial and temporal discretization.

Figures 310.70 to 310.77 illustrate the comparative results for both extreme cases of viscous coupling. In general, it is worth noting that the numerical results are in good agreement with the main characteristics of the mechanics of dispersive wave propagation in fully saturated, porous media, as indicated by the semi-analytical results. For example, numerical results well demonstrate that for the case of strong viscous coupling (high drag), the solid and fluid are in phase with each other, implying that the two-phase material behaves as a single continuum. Overall, the finite element solutions reproduce correctly the trends of wave propagation in both limiting cases of viscous coupling.

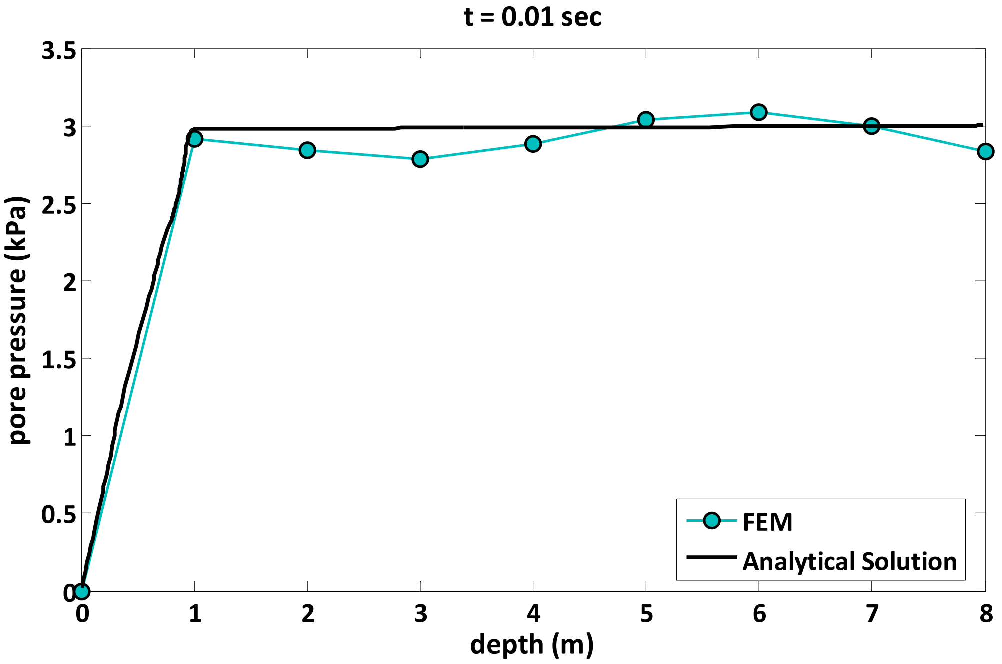


Figure 310.65: Response of pore pressure versus depth at $t = 0.01 sec$ due to step loading. Comparison of numerical results (FEM) with the analytical solution by de Boer et al. (1993).

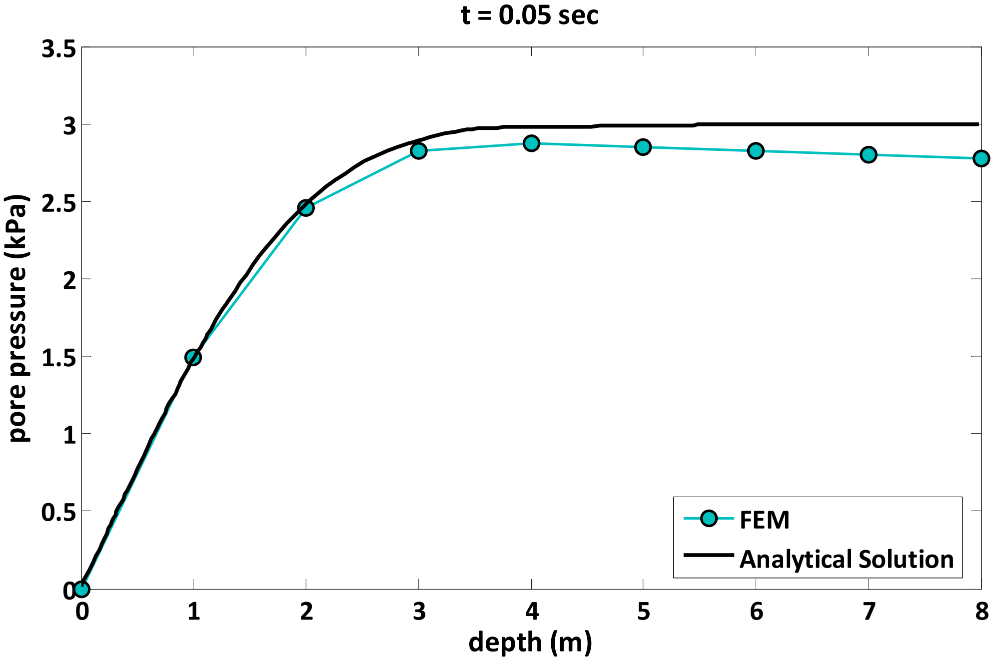


Figure 310.66: Response of pore pressure versus depth at $t = 0.05 sec$ due to step loading. Comparison of numerical results (FEM) with the analytical solution by de Boer et al. (1993).

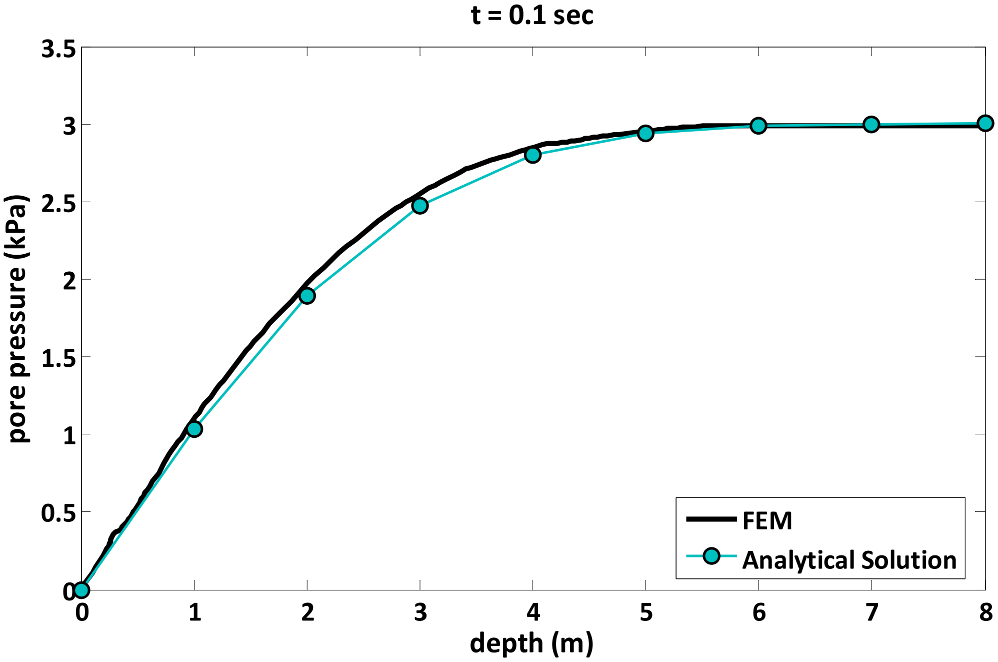


Figure 310.67: Response of pore pressure versus depth at $t = 0.1 sec$ due to step loading. Comparison of numerical results (FEM) with the analytical solution by de Boer et al. (1993).

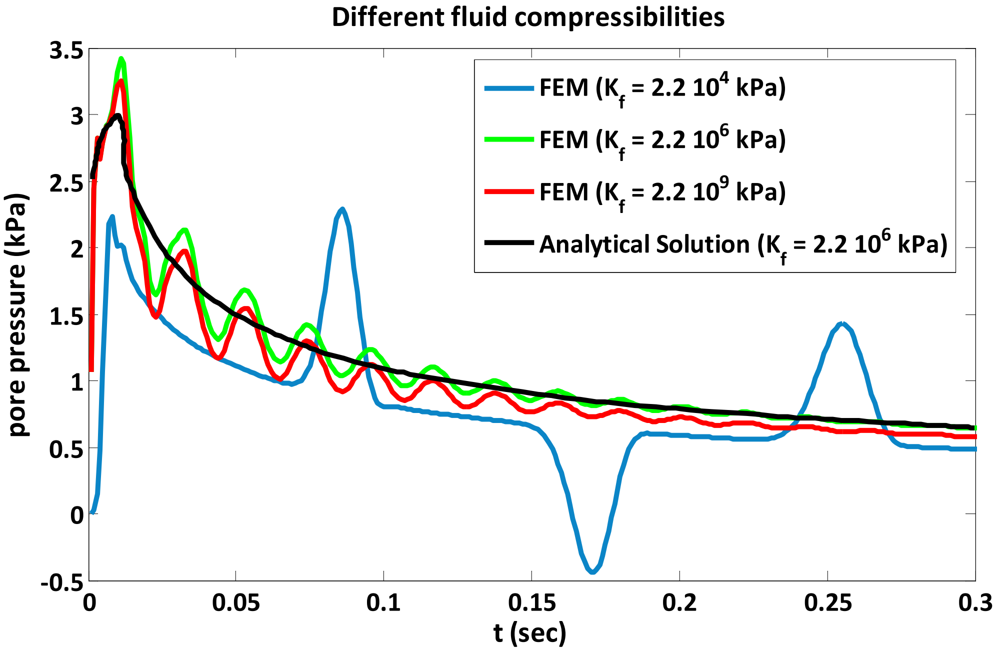


Figure 310.68: Time history of pore pressure at 1 m below the ground surface due to step loading for different fluid compressibility. Comparison of numerical results (FEM) with the analytical solution by de Boer et al. (1993).

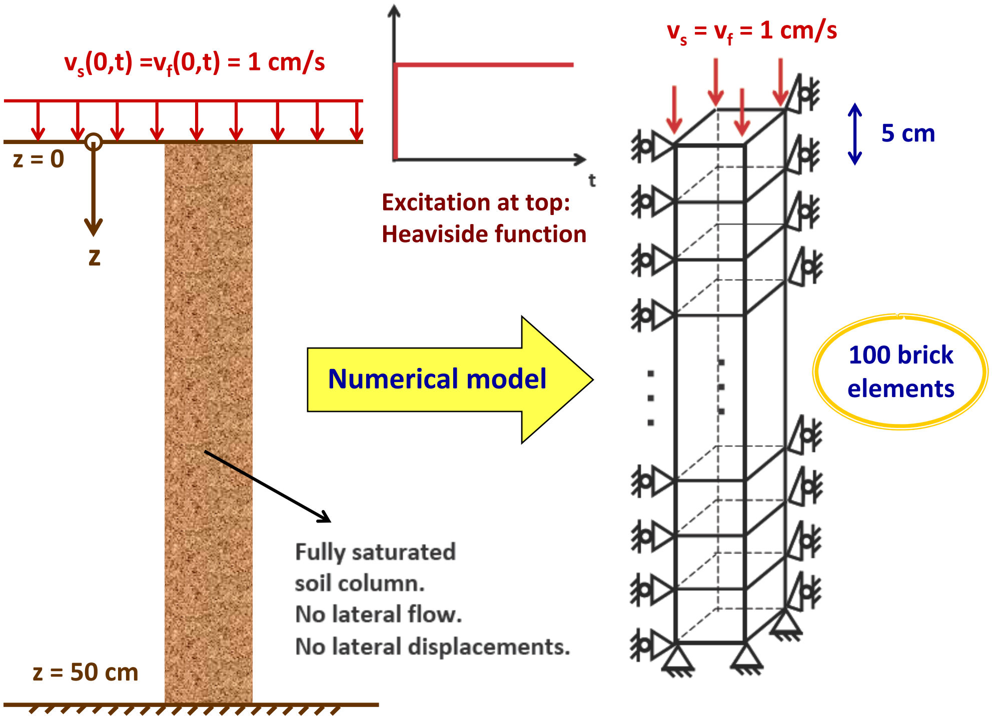


Figure 310.69: The numerical model used for the verification of the finite element implementation through comparison with the semi-analytical results provided by Hiremath et al. (1988).

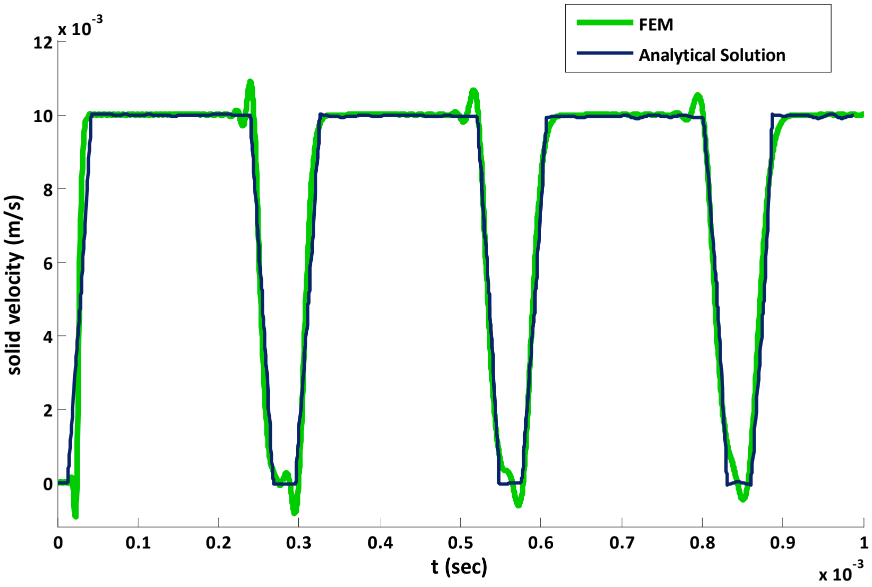


Figure 310.70: Time history of solid velocity at 10 cm below the surface. Comparison of numerical results (FEM) with the semi-analytical solution by Hiremath et al. (1988) for the case of high drag ($k = 0.148 \times 10^{-8} cm^3 s/g$).

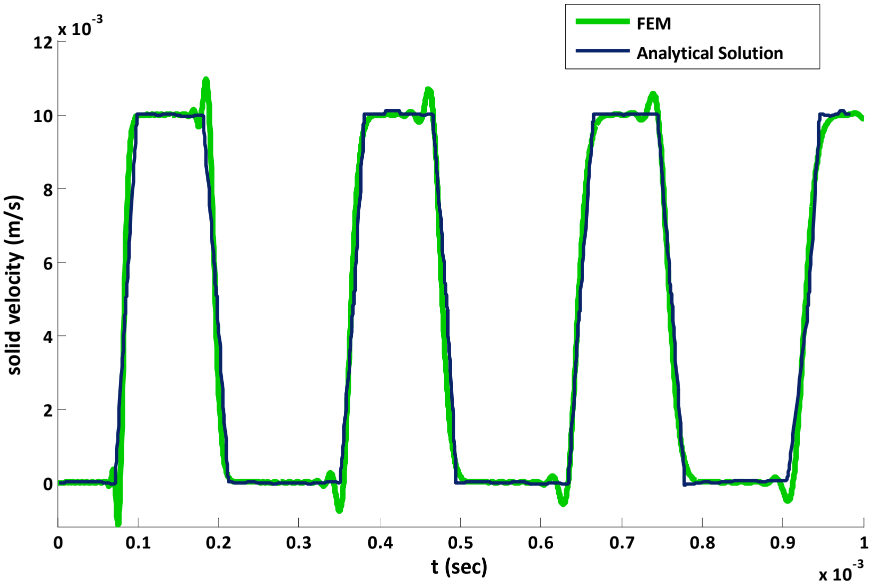


Figure 310.71: Time history of solid velocity at 30 cm below the surface. Comparison of numerical results (FEM) with the semi-analytical solution by Hiremath et al. (1988) for the case of high drag ($k = 0.148 \times 10^{-8} cm^3 s/g$).

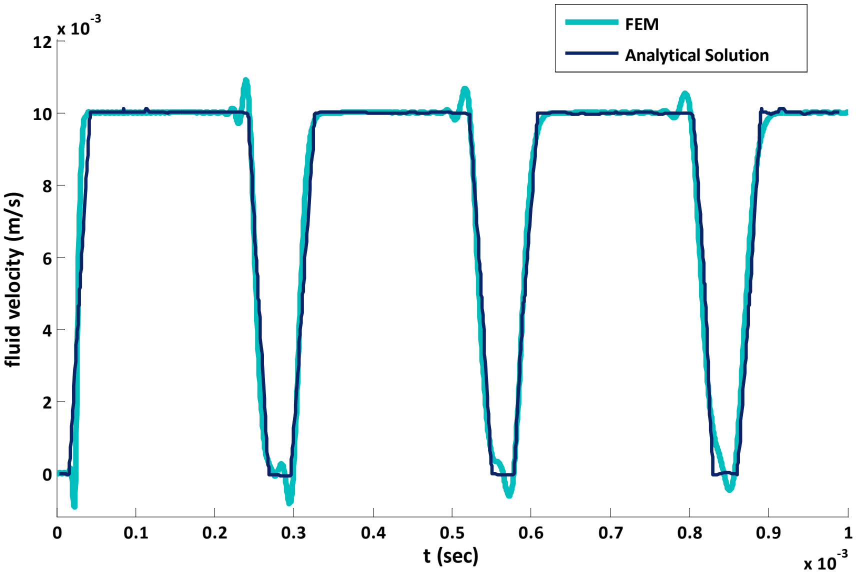


Figure 310.72: Time history of fluid velocity at 10 cm below the surface. Comparison of numerical results (FEM) with the semi-analytical solution by Hiremath et al. (1988) for the case of high drag ($k = 0.148 \times 10^{-8} cm^3 s/g$).

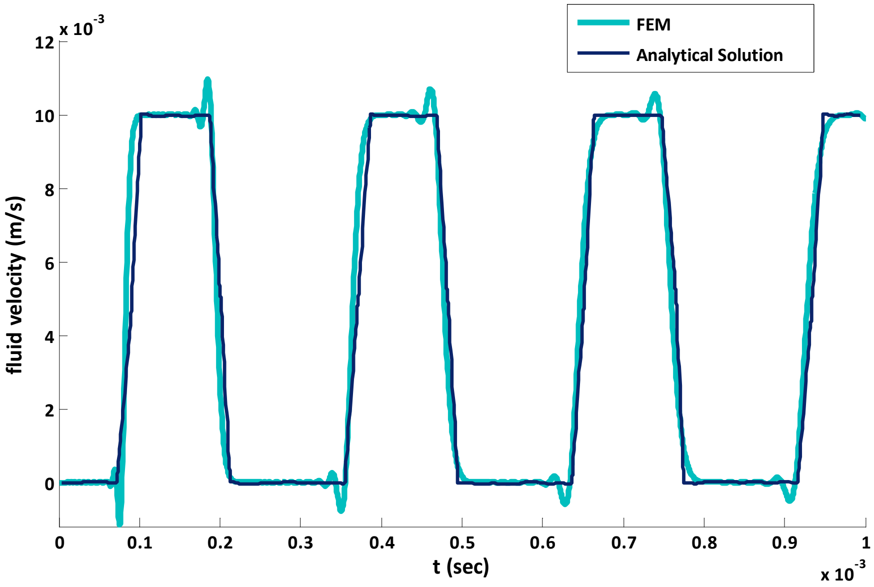


Figure 310.73: Time history of fluid velocity at 30 cm below the surface. Comparison of numerical results (FEM) with the semi-analytical solution by Hiremath et al. (1988) for the case of high drag ($k = 0.148 \times 10^{-8} cm^3 s/g$).

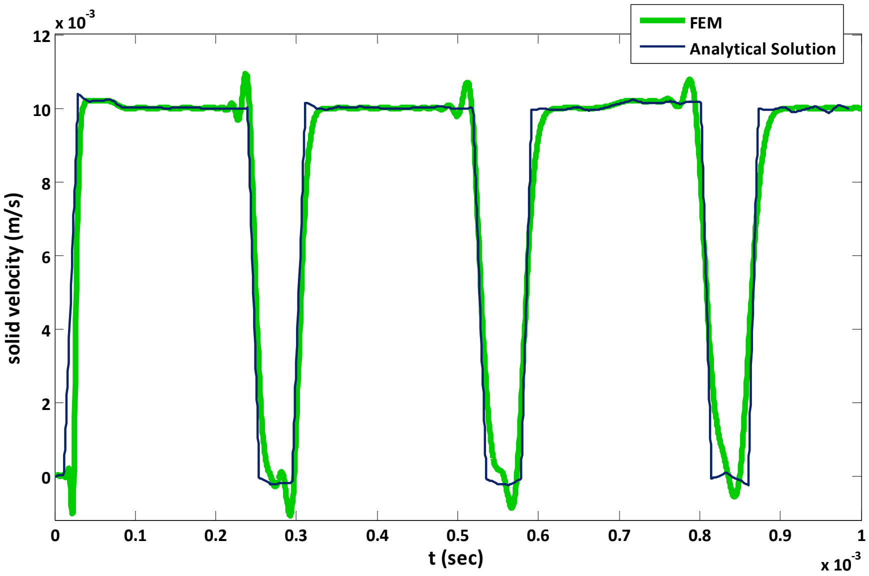


Figure 310.74: Time history of solid velocity at 10 cm below the surface. Comparison of numerical results (FEM) with the semi-analytical solution by Hiremath et al. (1988) for the case of high drag ($k = 0.148 \times 10^{-2} cm^3 s/g$).

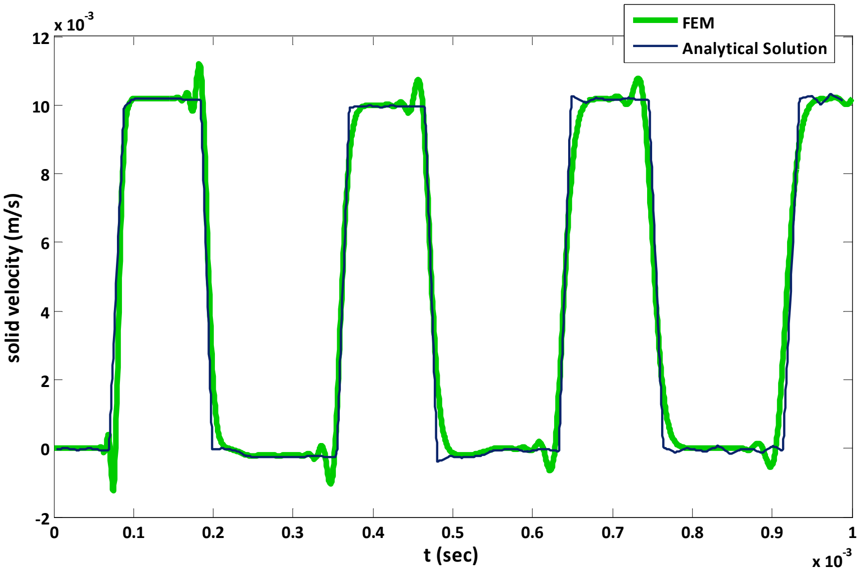


Figure 310.75: Time history of solid velocity at 30 cm below the surface. Comparison of numerical results (FEM) with the semi-analytical solution by Hiremath et al. (1988) for the case of high drag ($k = 0.148 \times 10^{-2} cm^3 s/g$).

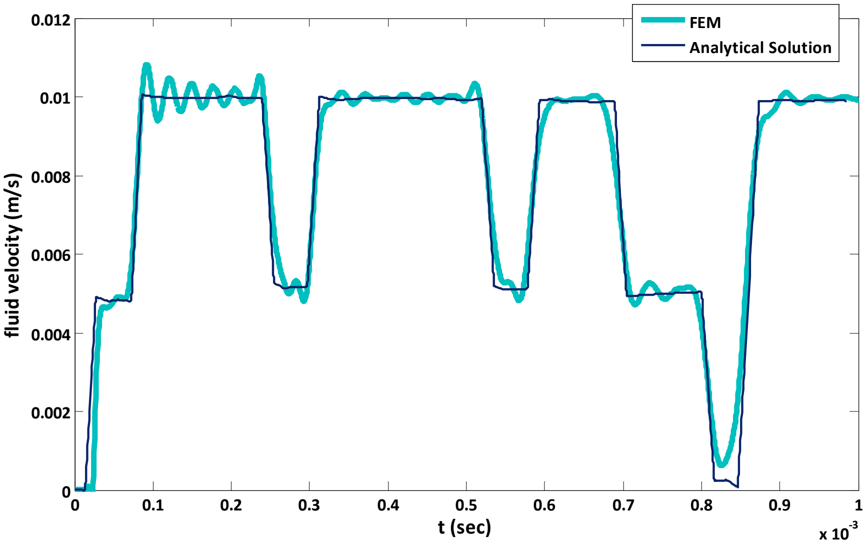


Figure 310.76: Time history of fluid velocity at 10 cm below the surface. Comparison of numerical results (FEM) with the semi-analytical solution by Hiremath et al. (1988) for the case of high drag ($k = 0.148 \times 10^{-2} cm^3 s/g$).

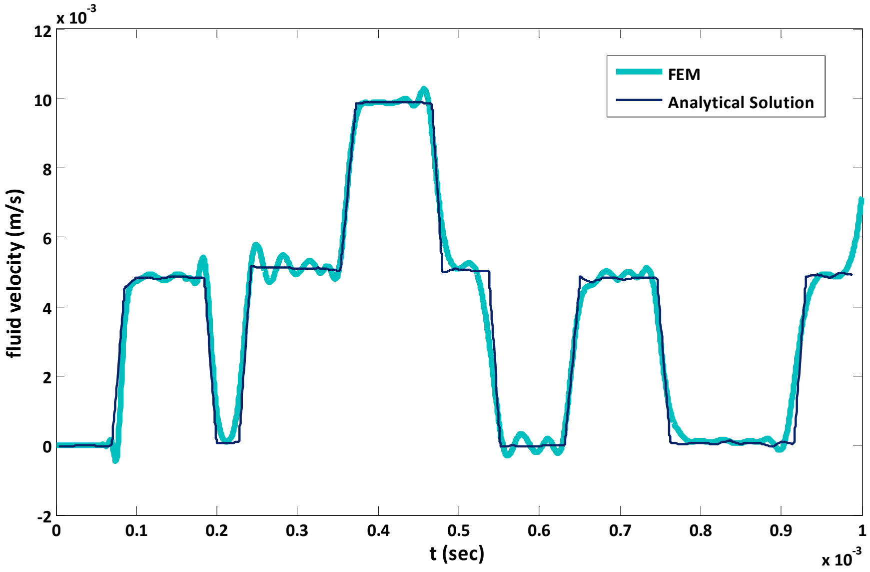


Figure 310.77: Time history of fluid velocity at 30 cm below the surface. Comparison of numerical results (FEM) with the semi-analytical solution by Hiremath et al. (1988) for the case of high drag ($k = 0.148 \times 10^{-2} cm^3 s/g$).

Chapter 311

Verification and Validation for Seismic Wave Propagation Problems

(1989-2000-2004-2005-2008-2009-2010-2011-2017-2018-2019-2021-)

(In collaboration with Dr. Nima Tafazzoli, Dr. Matthias Preisig, Dr. Federico Pisanò, Mr. Kohei Watanabe, Mr. Chao Luo, and Dr. Hexiang Wang)

311.1 Chapter Summary and Highlights

311.2 Wavelet Seismic Signals

A wavelet is a wave with specific definitions and parameters. The amplitude of a wavelet usually starts at zero, increases by time and ended up at zero again. Typically a wavelet can be plotted as a brief oscillation such as a the small oscillation recorded by seismogram. There are different types of wavelets each with their own properties used for specific purpose in signal processing. For specific purposes different wavelets might be summed up to come up with new type of wave. A recently developed wavelet analysis has become a powerful tool to analyze the soil-structure systems for transient loads providing information both in time and frequency domains. In wavelet representation the basis functions are localized and contained in finite time domains (Sarica and Rahman (2003)).

311.2.1 Ricker Wavelet

One type of wavelet motions is the Ricker wave (Ryan (1994), Mavroeidis and Papageorgiou (2003)). The formulation of Ricker wavelet is shown in Equations (311.1):

$$R(t) = A \quad (1 - 2\pi^2 f^2 \; t^2) \quad \exp(-\pi^2 f^2 t^2) \tag{311.1}$$

where $R(t)$ is the amplitude of the function in time, A if the maximum amplitude, and f is the peak frequency on the wavelet's frequency spectrum. Figure (311.1) shows the actual time history and fast Fourier transform of Ricker wavelet, where A is taken as 1 and f is taken as $5Hz$. As it is shown, the frequency range of the motion is narrower compared to the real earthquake motion.

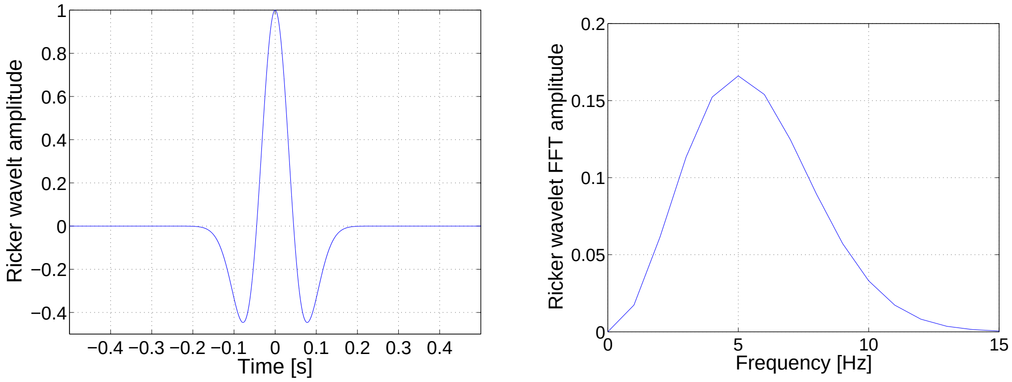


Figure 311.1: Frequency content and a time domain representation of Ricker wavelet

311.2.2 Ormsby Wavelet

Another example of interesting wavelet is called Ormsby wavelet (Ryan (1994)) which features a controllable flat frequency content with formulation shown in Equation (311.2).

$$\begin{aligned} f(t) &= A((\frac{\pi f_4{}^2}{f_4 - f_3} sinc(\pi f_4(t-t_s))^2 - \frac{\pi f_3{}^2}{f_4 - f_3} sinc(\pi f_3(t-t_s))^2) \\ &- (\frac{\pi f_2{}^2}{f_2 - f_1} sinc(\pi f_2(t-t_s))^2 - \frac{\pi f_1{}^2}{f_2 - f_1} sinc(\pi f_1(t-t_s))^2)) \end{aligned} \tag{311.2}$$

where f_1 and f_2 define the lower range frequency band, f_3 and f_4 define the higher range frequency band, A is the amplitude of the function, and t_s is the time that maximum amplitude is happening, and $sinc(x) = sin(x)/x$.

Figure (311.2) shows an example of Ormsby wavelet in time domain and frequency domain. In this case, wave has a flat frequency range of 5Hz to 20Hz. Shown in Figure (311.3) is half of the Ormsby wavelet in frequency domain which the frequency range starts from 0 and remains constant up to $20Hz$. This type of motion could be useful when low frequency range of motions are required for dynamic analysis of the systems.

Such broad band signals could be used to assess different aspects of soil-structure systems and with different incoming wave inclinations. While wavelet time domain motions are not the same as actual earthquakes, the idea is to use them for dynamic analysis of soil-structure systems for possible problems coming out of dynamic behavior, at different frequencies and for different energy input levels.

When used with the DRM, motions developed from different directions, different incident angles and different energies, will create a full envelope of these motions, which then can be used to evaluate performance based response of the soil-structure systems.

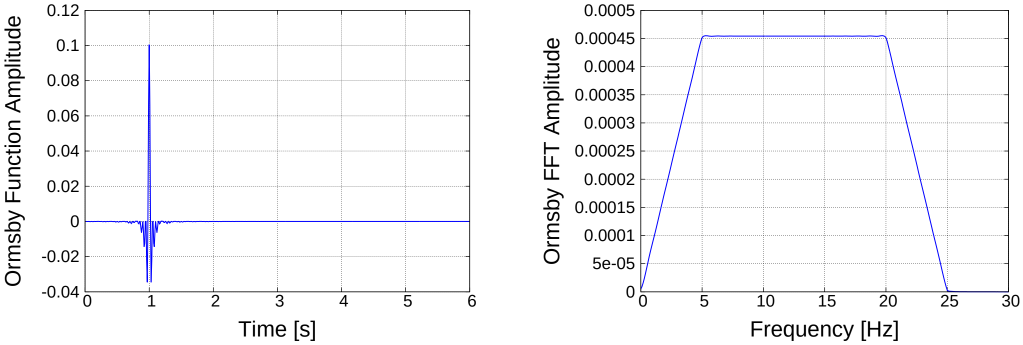


Figure 311.2: Frequency content and a time domain representation of an Ormsby wavelet, with constant frequency content between 5Hz and 20Hz

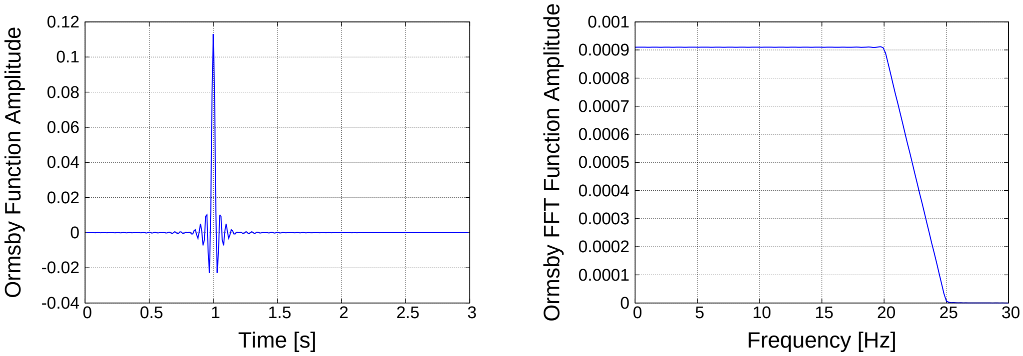


Figure 311.3: Frequency content and a time domain representation of half of Ormsby wavelet formulation, with minimum frequency of zero and maximum of 20Hz

311.3 Finite Element Mesh Size Effects on Seismic Wave Propagation Modeling and Simulation

311.3.1 Analysis Cases

Summary of the cases is shown in Table below. the input motion used is Ormsby wavelet which the corner cutoff frequency is shown in the table.

Case Number	Model Height (m)	Shear Wave Velocity (m/s)	Element Size (m)	Frequncy Cutoff (Hz)	Maximum Propagation Frequency (Hz)
1	1000	1000	10	3	10
2	1000	1000	20	3	5
3	1000	1000	10	8	10
4	1000	1000	20	8	5
5	1000	1000	50	3	2
6	1000	1000	50	8	2
7	100	100	1	3	10
8	100	100	2	3	5
9	100	100	1	8	10
10	100	100	2	8	5
11	100	100	10	8	1
12	1000	100	10	8	1
13	1000	100	20	8	0.5
14	1000	100	50	8	0.2

311.3.2 Comparison of Case 1 and 2

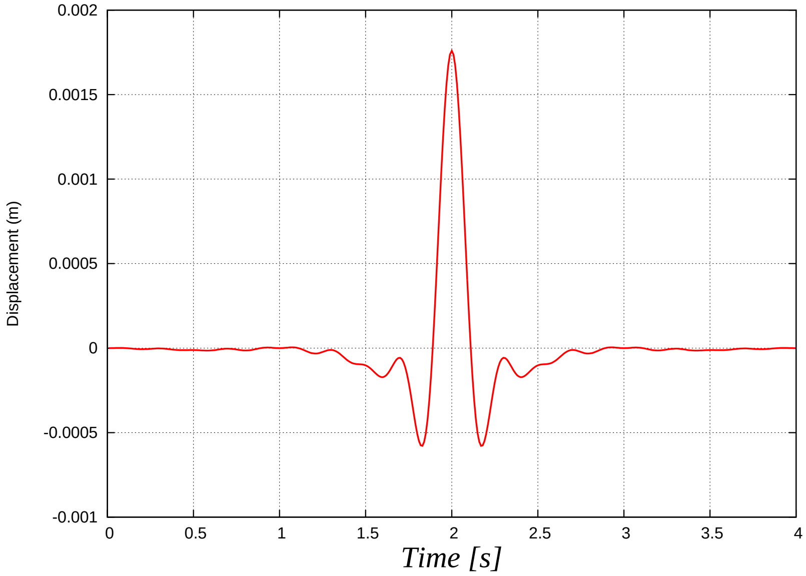


Figure 311.4: Displacement time history of input motion (Ormsby Wavelet)

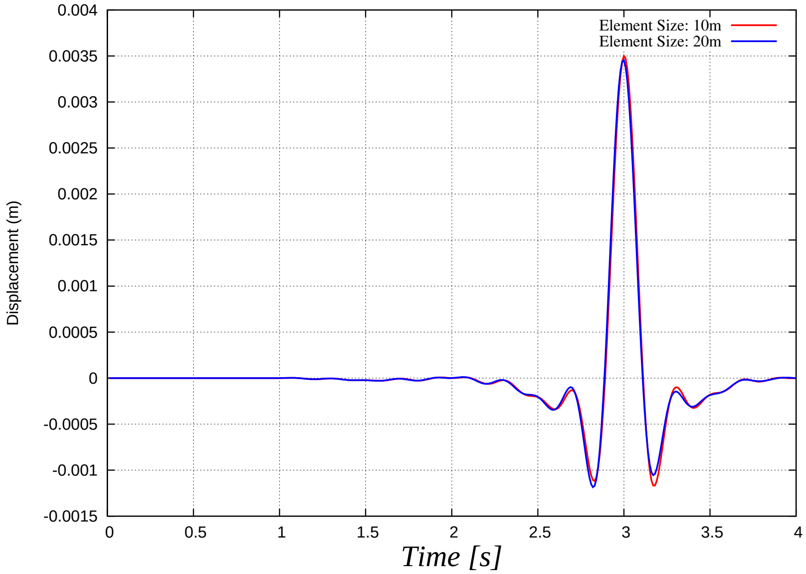


Figure 311.5: Comparison of displacement time histories of case 1 and 2 at top of the model

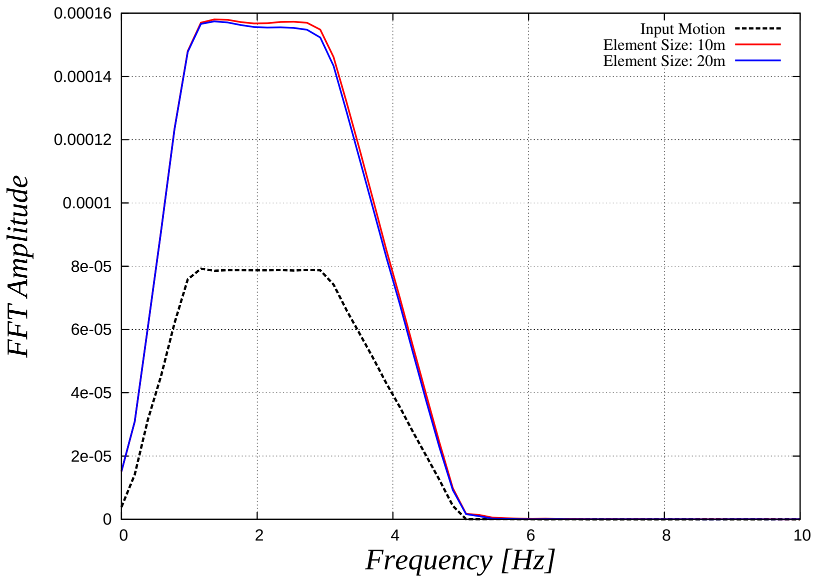


Figure 311.6: Comparison of FFT of case 1 and 2 at top of the model and input motion at the bottom of model

311.3.3 Comparison of Case 3 and 4

311.3.4 Comparison of Cases 3, 4, and 6

311.3.5 Comparison of Case 7 and 8

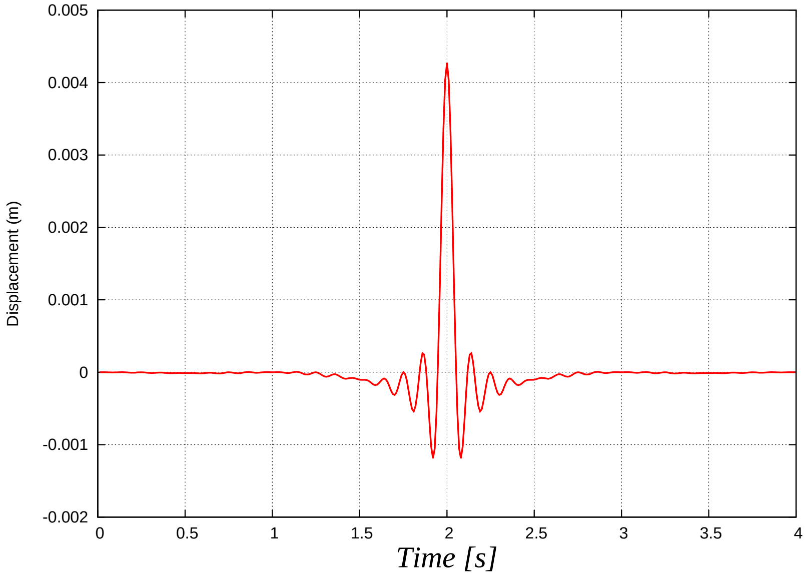


Figure 311.7: Displacement time history of input motion (Ormsby Wavelet)

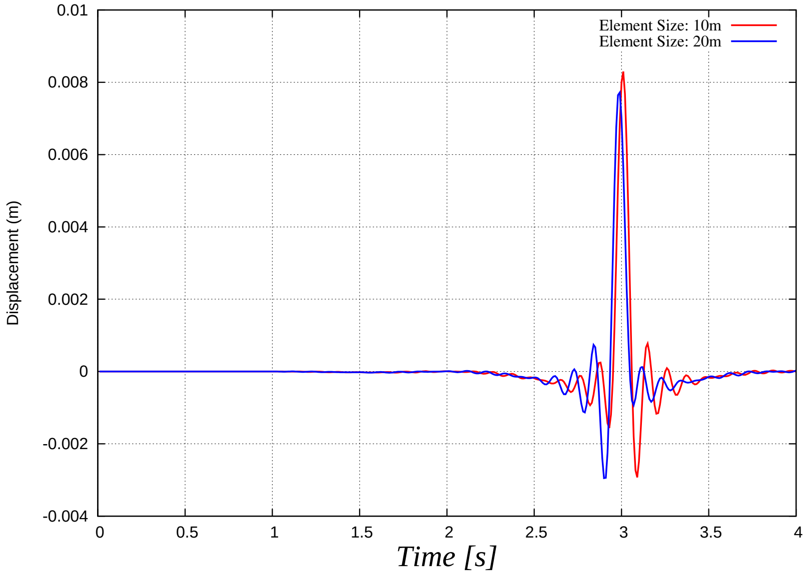


Figure 311.8: Comparison of displacement time histories of case 3 and 4 at top of the model

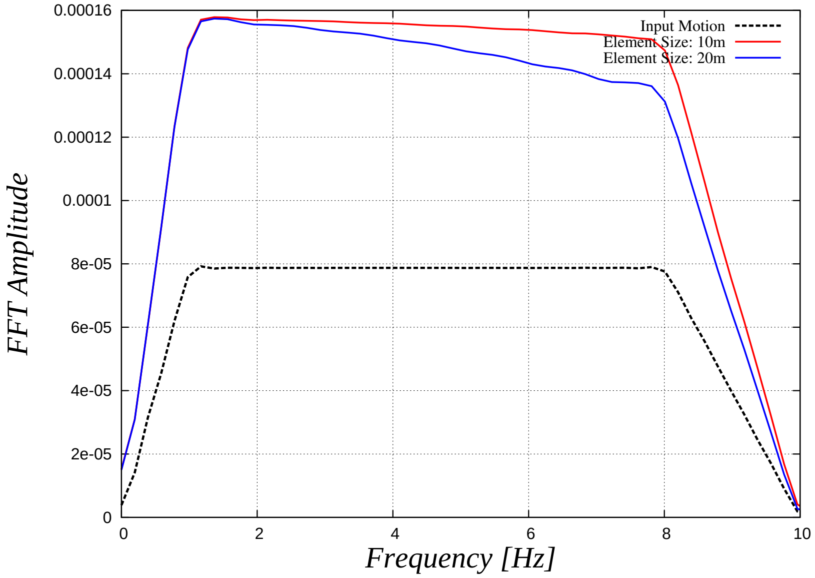


Figure 311.9: Comparison of FFT of case 3 and 4 at top of the model and input motion at the bottom of model

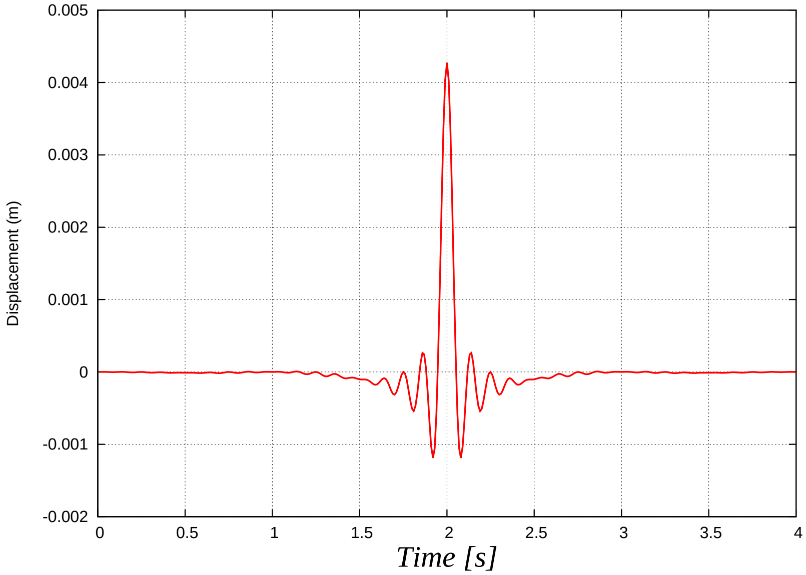


Figure 311.10: Displacement time history of input motion (Ormsby Wavelet)

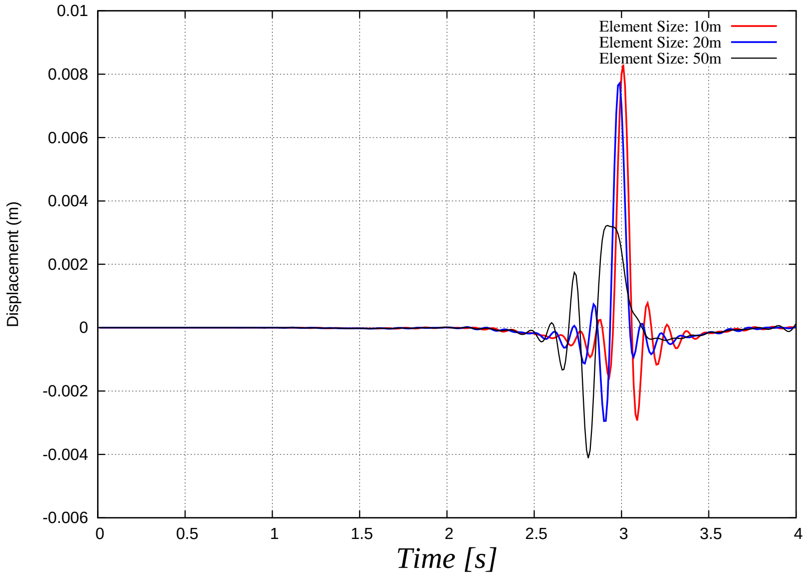


Figure 311.11: Comparison of displacement time histories of case 3, 4, and 6 at top of the model

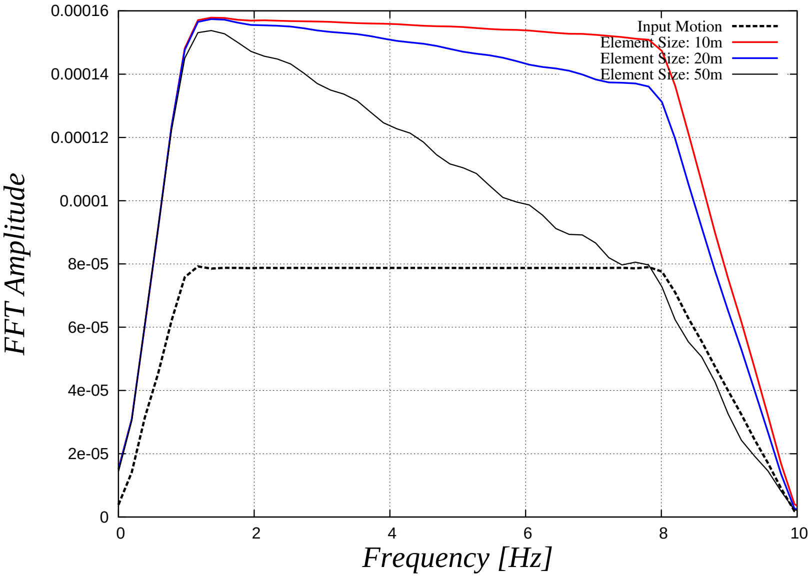


Figure 311.12: Comparison of FFT of case 3, 4, and 6 at top of the model and input motion at the bottom of model

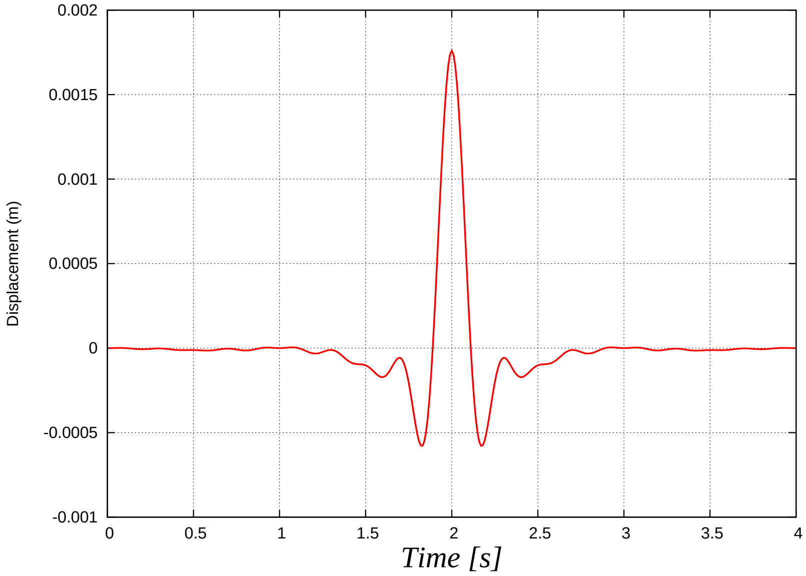


Figure 311.13: Displacement time history of input motion (Ormsby Wavelet)

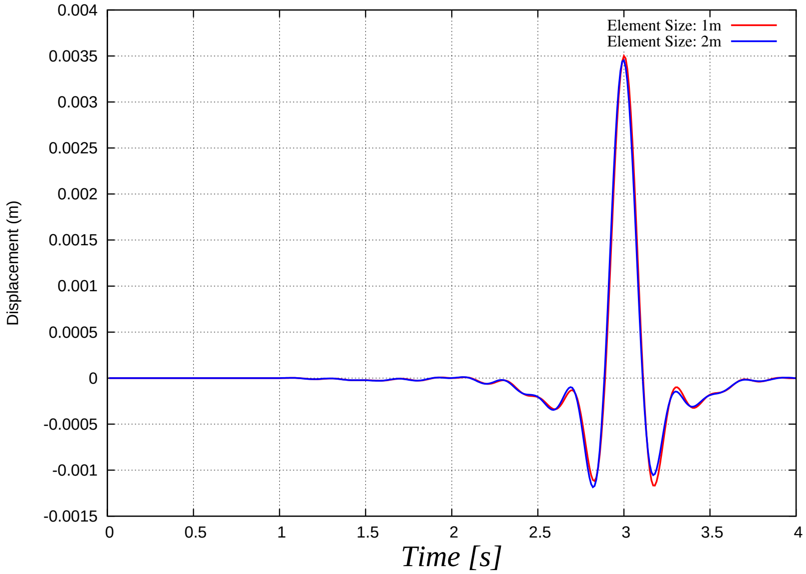


Figure 311.14: Comparison of displacement time histories of case 7 and 8 at top of the model

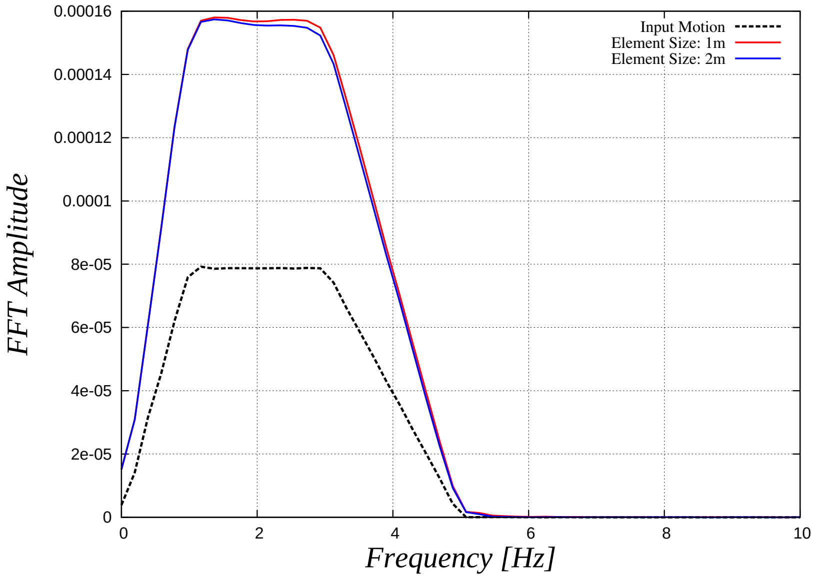


Figure 311.15: Comparison of FFT of case 7 and 8 at top of the model and input motion at the bottom of model

311.3.6 Comparison of Case 9, 10, and 11

311.3.7 Comparison of Case 12, 13, and 14

311.4 Damping of the Outgoing Waves

311.4.1 Comparison of Rayleigh Damping and Caughey 4th Order Damping

As mentioned before, Caughey damping in general will damp out the motions at specified modes (frequencies) to be specified which could also be the natural frequencies of the system. Depending on the type of damping to be used, the response of those modes would be affected. In order to observe the damping effect on certain modes, a soil profile is made with thickness of $50m$ and shear wave velocity of $100m/s$. For input motion, an Ormsby wavelet with frequency range of 0 to 7 Hz is considered at the base of model. The wave is propagated through the soil layer using the elastic transfer functions and comparison is made between the case which Rayleigh damping is used versus the case which the frequency independent damping is used in the model. The same procedure is done by using Caughey damping.

Figures (311.22) and (311.23) show the base motion (Ormsby wavelet), motion at the surface

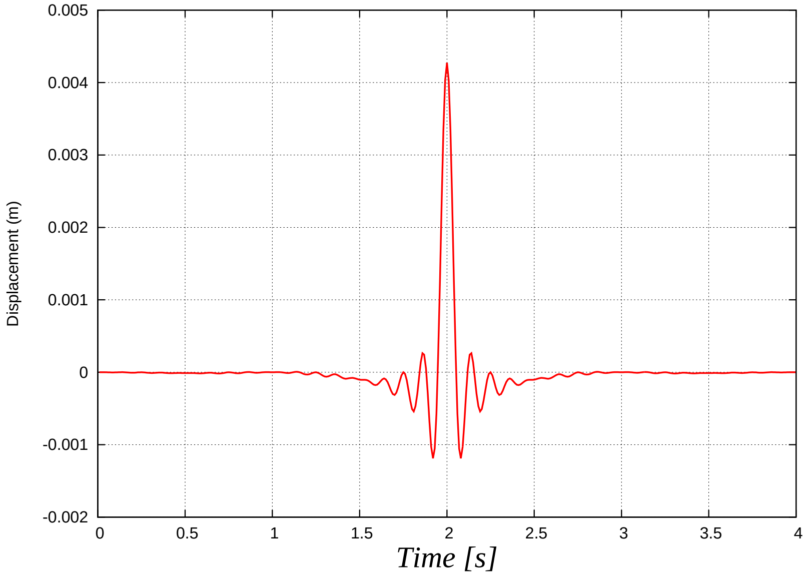


Figure 311.16: Displacement time history of input motion (Ormsby Wavelet)

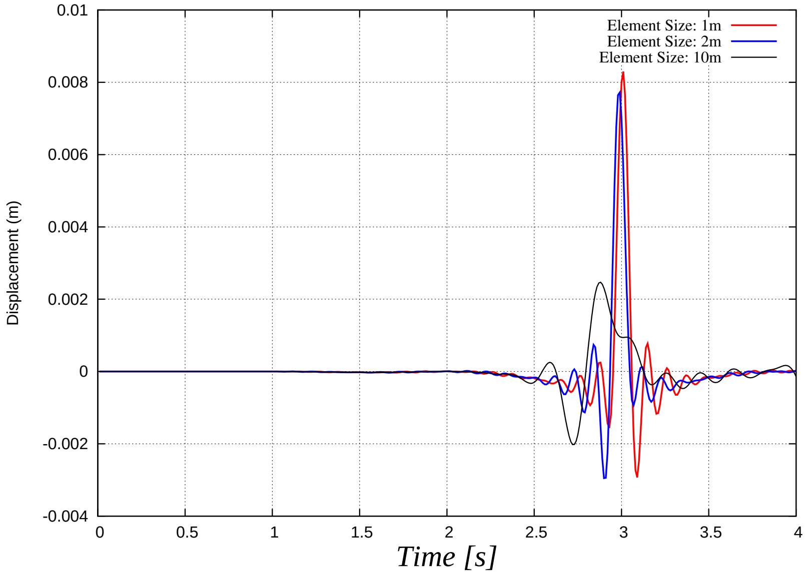


Figure 311.17: Comparison of displacement time histories of case 9, 10, and 11 at top of the model

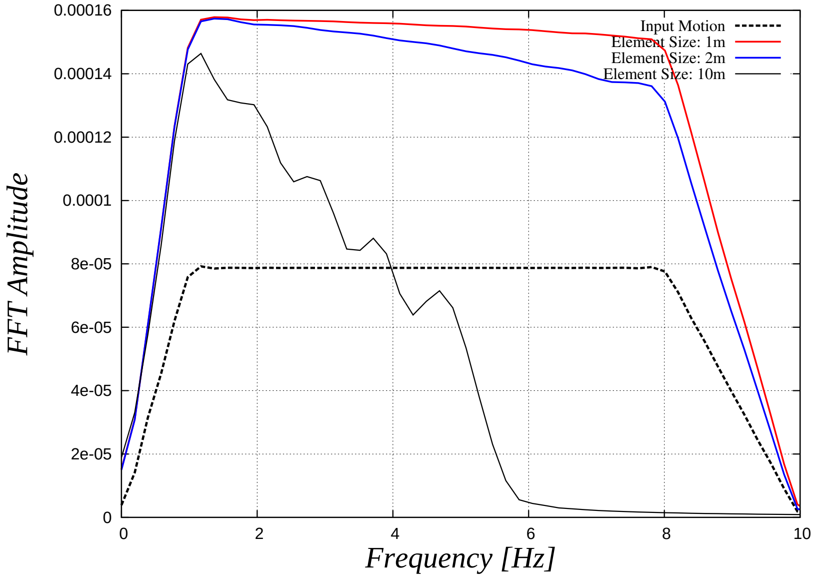


Figure 311.18: Comparison of FFT of case 9, 10, and 11 at top of the model and input motion at the bottom of model

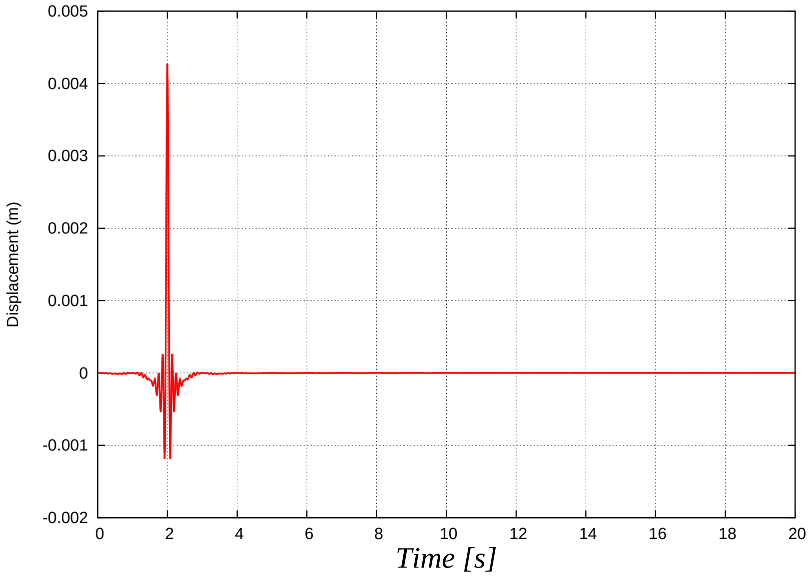


Figure 311.19: Displacement time history of input motion (Ormsby Wavelet)

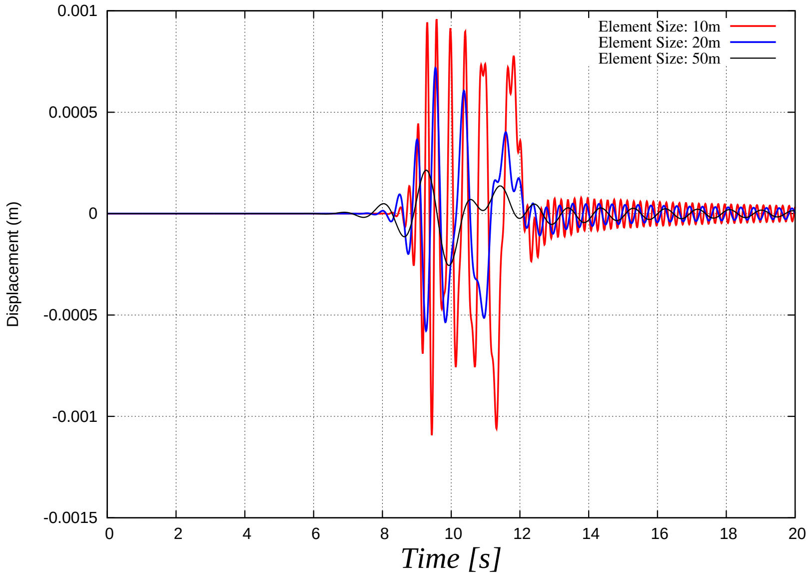


Figure 311.20: Comparison of displacement time histories of case 12, 13, and 14 at top of the model

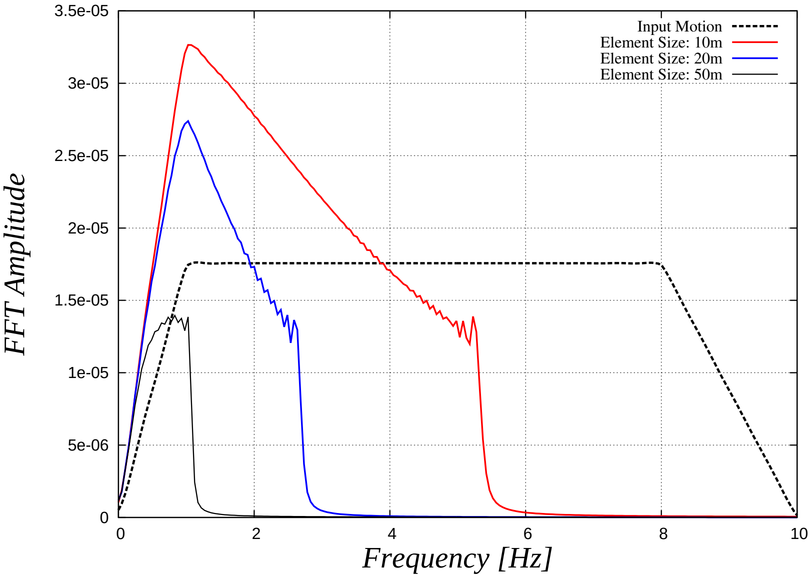


Figure 311.21: Comparison of FFT of case 12, 13, and 14 at top of the model and input motion at the bottom of model

considering the frequency independent damping, motion at the surface using Rayleigh wave (frequency dependent), motion at the surface using Caughey damping of 4th order (frequency dependent) as well as how Rayleigh and Caughey damping ratio change with frequency. It can be observed how the response is affected at different modes using Rayleigh damping versus using Caughey damping.

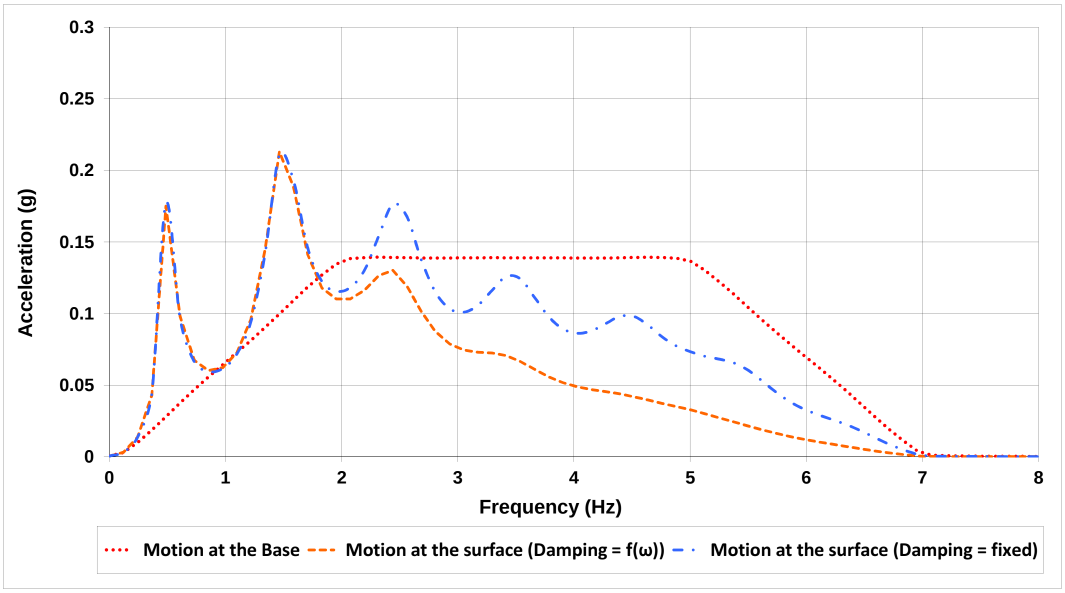


Figure 311.22: Comparison of obtained motion at the surface using frequency independent damping and frequency dependent Rayleigh damping.

311.4.2 Parametric Study on Effect of Rayleigh Damping on Reflected Waves

As mentioned in previous chapter, one of the issues of the modeling in dynamic analysis is reflecting of the motions from the boundaries since there are limitations regarding the size of the problems we can model. In order to reduce the computational cost of the problems, the size of the mesh has to be reduced. By reducing the size of the model the chance of reflecting the motions from the boundaries gets higher since there is less volume for the waves to get dissipated.

There are different ways to reduce reflection of the waves from the numerical boundaries such as PML, viscous dampers, infinite elements, or considering Rayleigh damping for specific elements. Presented here show the results of wave propagation models considering Rayleigh damping. There are different damping patterns used here such as constant damping ratio for all the elements in the damping zone or linear pattern of increasing the damping ratio.

In order to find the Rayleigh damping coefficients, two frequencies have to be considered. In these examples both cases of using the natural frequencies of the soil column and also using the dominant

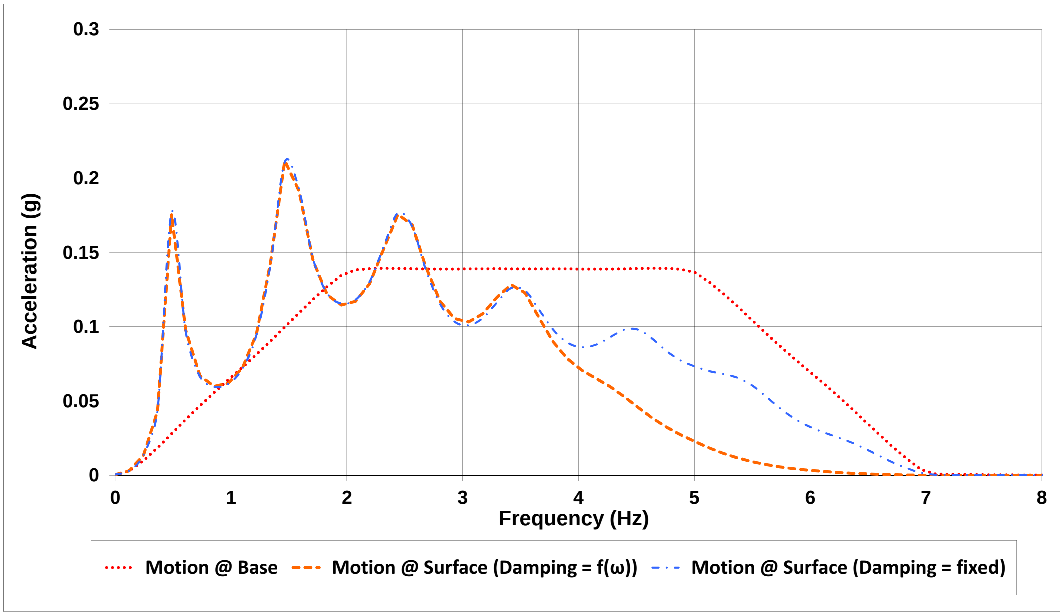


Figure 311.23: Comparison of obtained motion at the surface using frequency independent and frequency dependent Caughey 4th order damping.

periods of the motions are used and comparison is made. Different shear wave velocities and input motion frequencies are used which is mentioned for each case. The input motion considered for simulation is Ricker wavelet considering different dominant frequencies. V_s is the soil profile shear wave velocity and f_r is the frequency of the Ricker wavelet, and xi is the Rayleigh damping ratio at considered frequencies.

The height of the finite element model is $60m$ and boundary conditions are introduced in order to model 1C wave propagation. The motion is imposed at one side of the model and Rayleigh damping is applied to couple of the elements on the other side of the model in order to damp out the waves. Results are recorded at the boundary of damped and undamped zones.

Figure (311.24) shows the comparison of time histories for the soil column with shear wave velocity of $100m/s$ and input motion frequency of $8Hz$. Frequencies used to calculate Rayleigh damping coefficients in this case are natural frequencies of the soil column. The same damping ratio is used for all the damping zone elements. It can be observed that the one with constant damping ratio of 0.5 has done better job in terms of damping out the reflected motions.

Same analysis is done by using frequencies of $6Hz$ and $12Hz$ for the Rayleigh damping. As shown in Figure (311.25), in this case the reflected waves are damped out more comparing to previous case where natural frequencies of the soil were used for Rayleigh damping. This fact shows that the frequencies to be used for calculating the Rayleigh damping coefficients, do not have to be the natural frequencies of the soil which sometimes used in practice and depends on the frequency range of the input motion as well. Since the Ricker wavelet used here has a peak frequency of $8Hz$, the higher values of frequencies should be used for Rayleigh damping coefficients. Figure (311.26) shows the results of the same case except that linear increasing pattern is used for damping ratio of the 5 elements in damping zone. It seems that using the linear pattern starting from 0.3 to 1.1 results in less reflected motions.

The reason could be because of the nature of Rayleigh damping which is frequency dependent. So different damping ratios are observed at different frequencies. In deed by changing the damping ratio at each element, five different patterns of Rayleigh damping are being used which has more capability of absorbing motions with different frequencies and amplitudes.

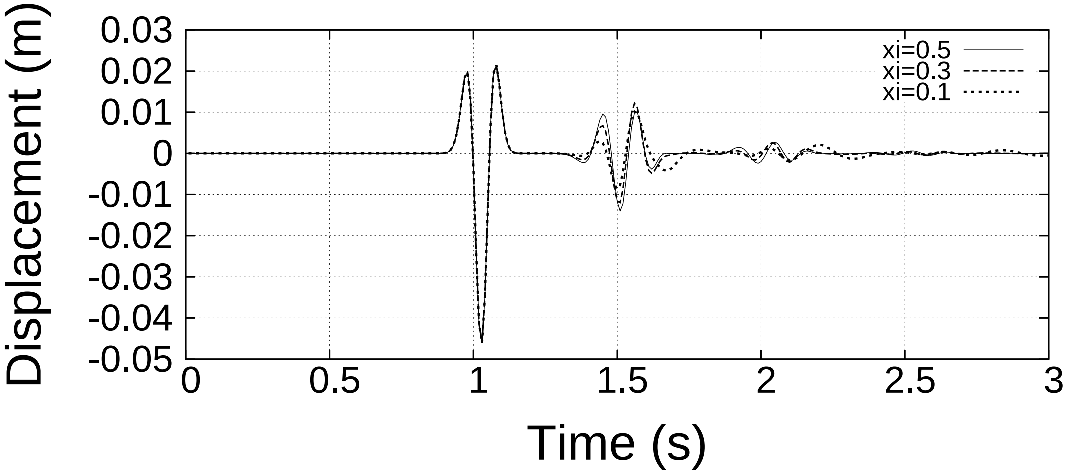


Figure 311.24: Displacement time history considering Rayleigh damping using natural frequencies of the soil, V_s=100m/s , f_r=8Hz

Figures (311.27) and (311.28) show the displacement time histories for same patterns of constant damping ratio for all elements in the damping zone and linearly increasing damping ratios respectively but for shear wave velocity of $V_s = 300m/s$ and input motion frequency of $5Hz$. Same conclusion can be made here as previous case regarding the pattern of damping ratios and frequencies to be used for Rayleigh damping.

Figures (311.29) and (311.30) are comparisons of recorded displacement time histories between patterns of same damping ratio for damping zone elements, damping ratio changes along the length of damping zone, and case of with out damping. Figure (311.29) is the case which shear wave velocity of the soil column is $100m/s$ and frequency of input Ricker motion is $1Hz$ while soil profile used in Figure (311.30) has shear wave velocity of $300m/s$ with input motion frequency of $8Hz$.

It can be observed that in case of having no physical damping, waves are getting trapped in the model and are reflecting back from the boundaries. Displacement time histories obtained from mentioned

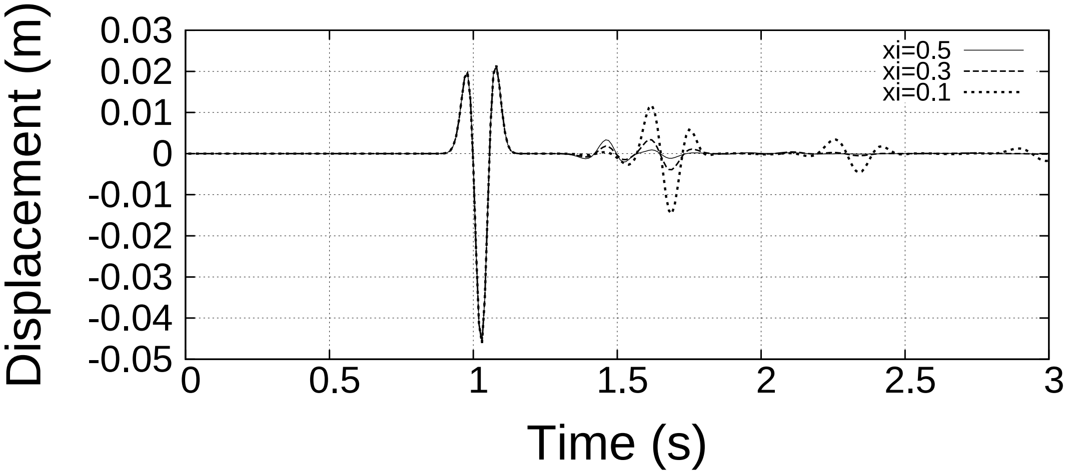


Figure 311.25: Displacement time history considering Rayleigh damping using f1,f2 = 6,12 Hz , V_s=100m/s , f_r=8Hz

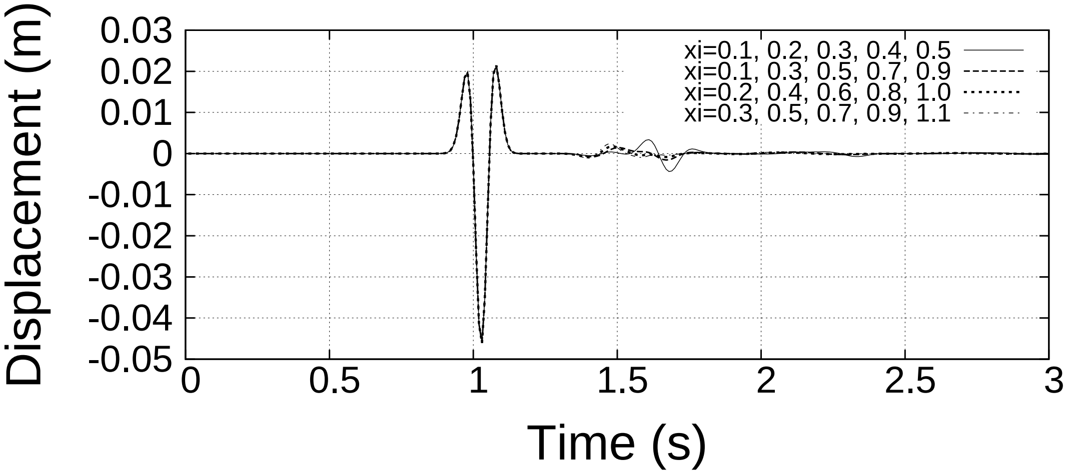


Figure 311.26: Displacement time history considering Rayleigh damping using f1,f2 = 6,12 Hz , V_s=100m/s , f_r=8Hz

patterns of damping ratios have minor differences which does not mean always will be this close but still the pattern of using linearly increasing of damping ratio seems to do a better job for damping the reflecting waves.

In order to have a better understanding of these patterns of damping, wave propagation through the depth of model is recorded for case of shear wave velocity of $100 m/s$ and input motion frequency of $8 Hz$. Displacement time histories in Figures (311.31) to (311.33) show wave propagation through the model for cases of using uniform damping ratios, linearly increasing damping ratios, and with out damping

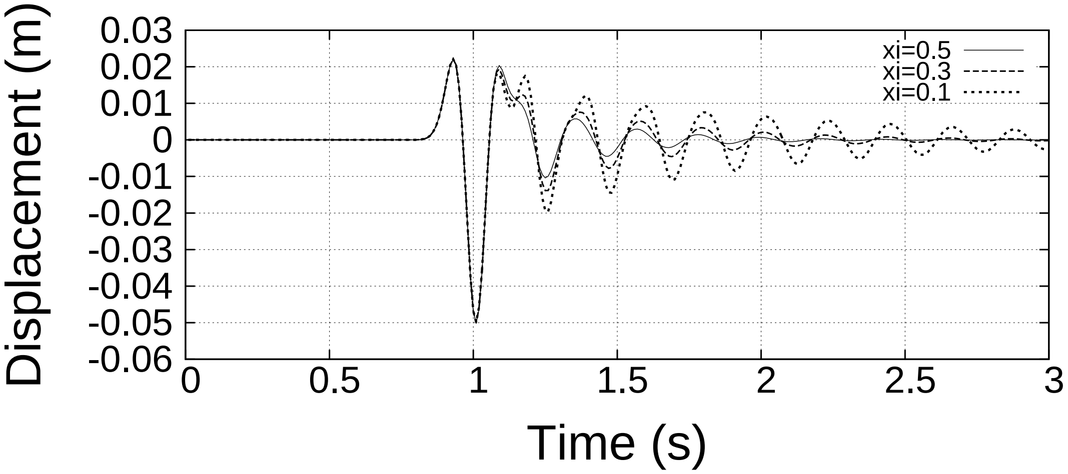


Figure 311.27: Displacement time history considering Rayleigh damping using f1,f2 = 5,8 Hz , V_s=300m/s , f_r=5Hz

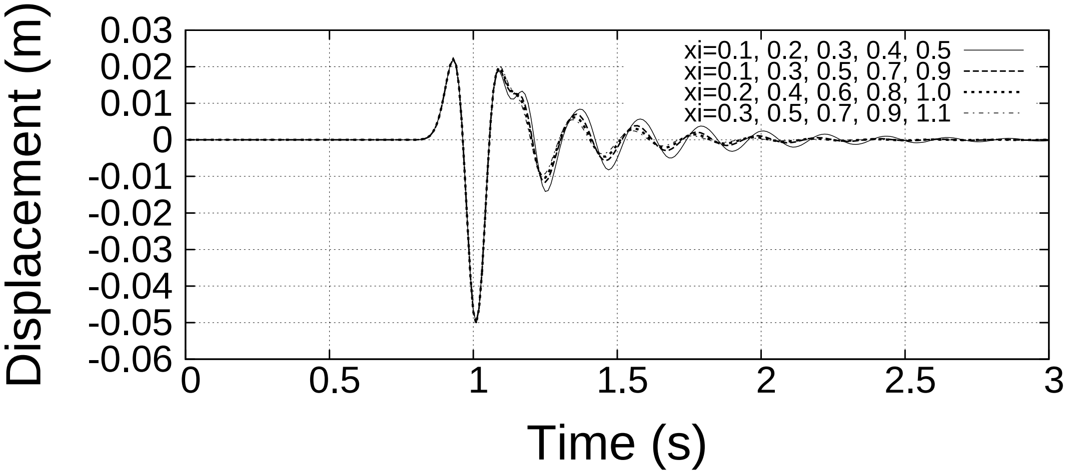


Figure 311.28: Displacement time history considering Rayleigh damping using f1,f2 = 5,8 Hz , V_s=300m/s , f_r=5Hz

respectively. By looking at the wave propagation through the whole soil profile it can be concluded that for this soil profile using the linearly increasing of damping ratios does a better job for damping the reflected motions at different depths.

Figure (311.34) shows the comparison of cumulative total energy time histories for the soil profile with shear wave velocity of $100m/s$ and input motion frequency of $8Hz$ for different Rayleigh damping patterns of uniform, increasing linearly, and case of no damping. What is expected to be observed is that total energy keeps increasing until the input motion gets to zero in time which energy should remain

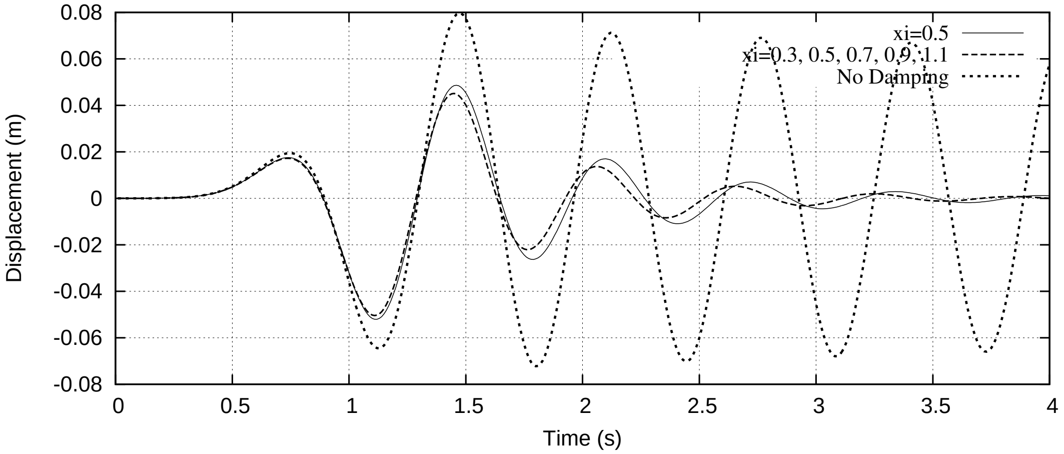


Figure 311.29: Displacement time history considering Rayleigh damping using f1,f2 = 0.5,2 Hz , V_s=100m/s , f_r=1Hz

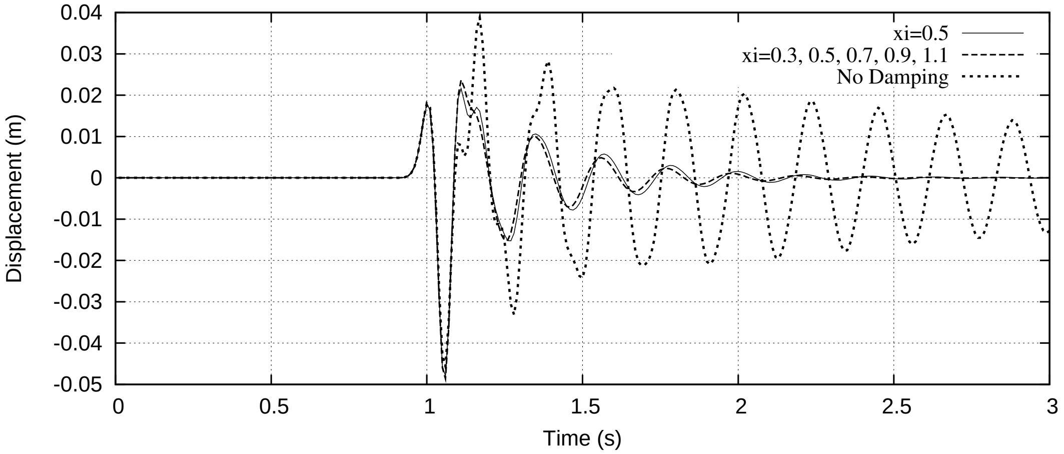


Figure 311.30: Displacement time history considering Rayleigh damping using f1,f2 = 6,12 Hz , V_s=300m/s , f_r=8Hz

constant unless there are waves reflecting back from boundaries. As it is shown, total energy slightly increase by time due to the reflected motions. This difference is much higher for case of no physical damping used since higher portion of the motions will get trapped in the model.

In order to be able to see the effect of size of damping zone on reflected motions, analysis is done on the soil profile with shear wave velocity of $100m/s$ and frequency of $8Hz$ for input motion. Comparison of displacement time histories for different size of the damping zones is shown in Figure (311.35). As expected, by reducing the size of damping zone, more waves are reflecting back from model boundaries.

The effect of number of elements to be used in damping zone is also studied here. Comparison is

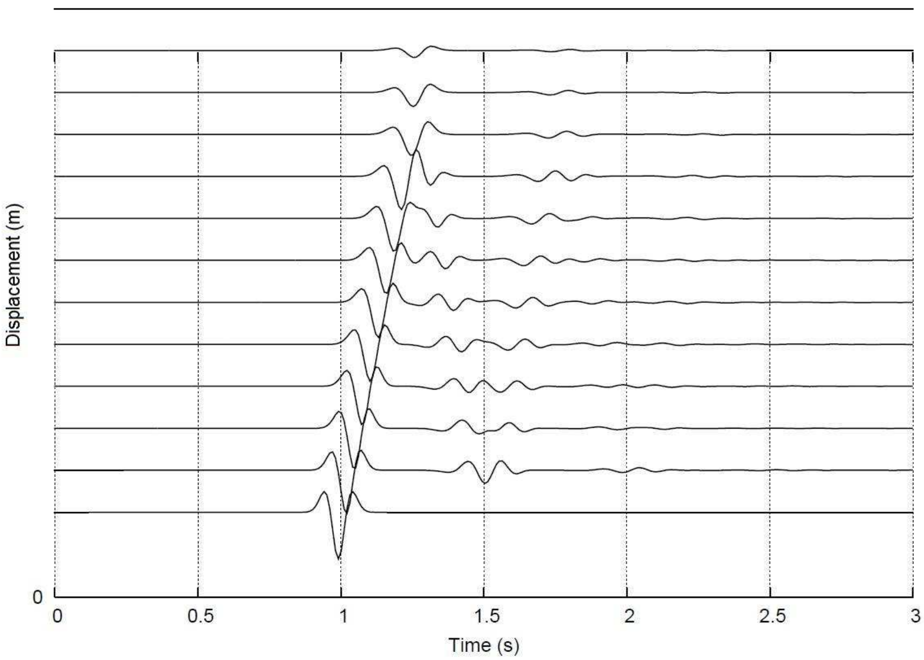


Figure 311.31: Displacement time history in depth considering homogeneous damping (15m), xi=0.5, V_s=100m/s, f_r=8Hz

made for cases which the size of the damping zone is the same but the size of the elements (and therefore number of the elements) in that zone is changed. The size of the damping zone assumed to be $15m$ while the number of the elements used in that zone is considered to be 3, 5, and 15. The comparison for this change of number of the elements is shown in Figure (311.36). Rayleigh damping ratio with pattern of increasing linearly from 0.3 to 1.1 is used. As it is observed, by reducing number of elements in the damping zone, the amount of reflected waves are getting higher.

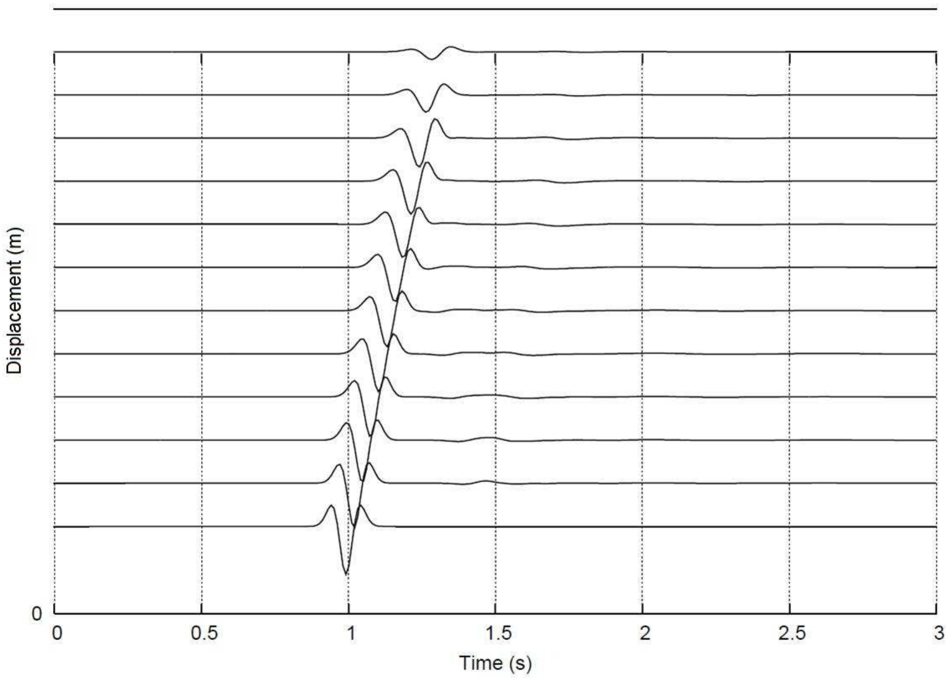


Figure 311.32: Displacement time history in depth considering linear increasing of xi in Rayleigh damping (every 3m), xi=0.3, 0.5, 0.7, 0.9, 1.1, V_s=100m/s, f_r=8Hz

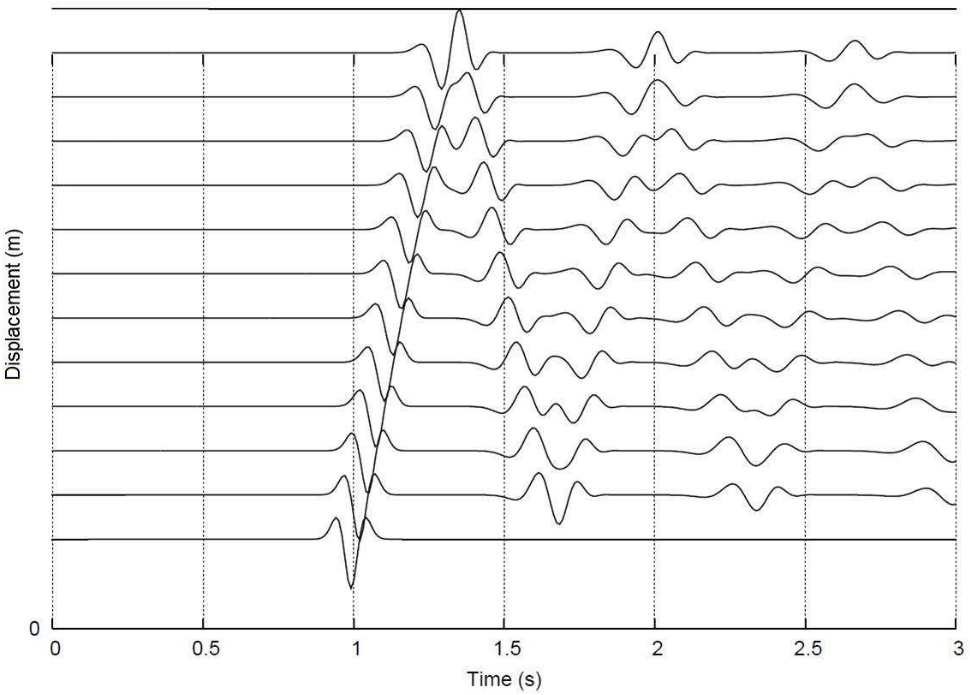


Figure 311.33: Displacement time history in depth with out considering damping, V_s=100m/s, f_r=8Hz

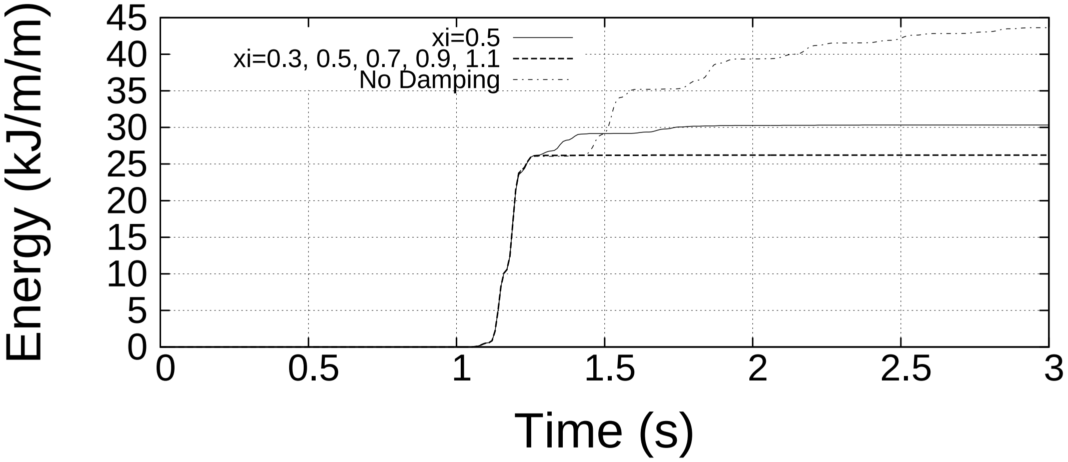


Figure 311.34: Comparison of energy time history by considering different Rayleigh damping patterns (in the non-damping zone at the middle of model), V_s=100m/s, f_r=8Hz

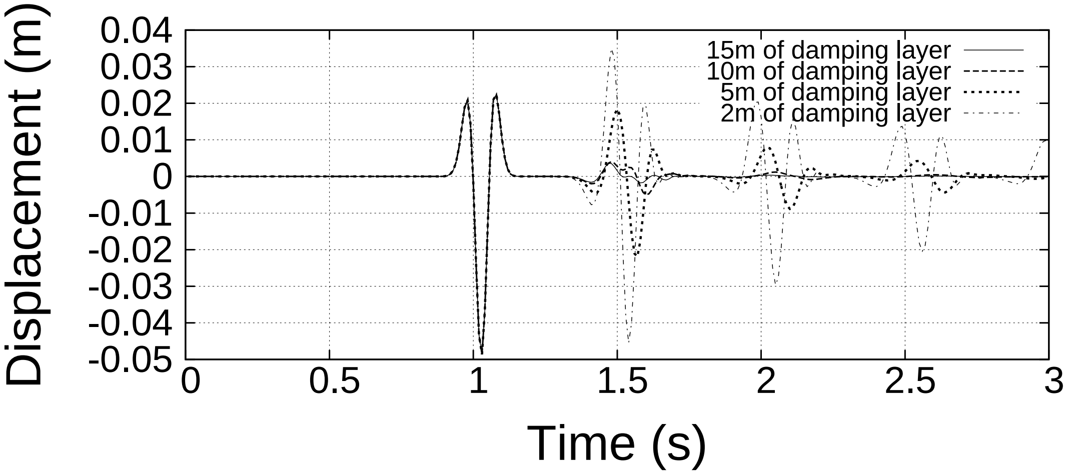


Figure 311.35: Comparison of displacement time histories for different size of damping zones

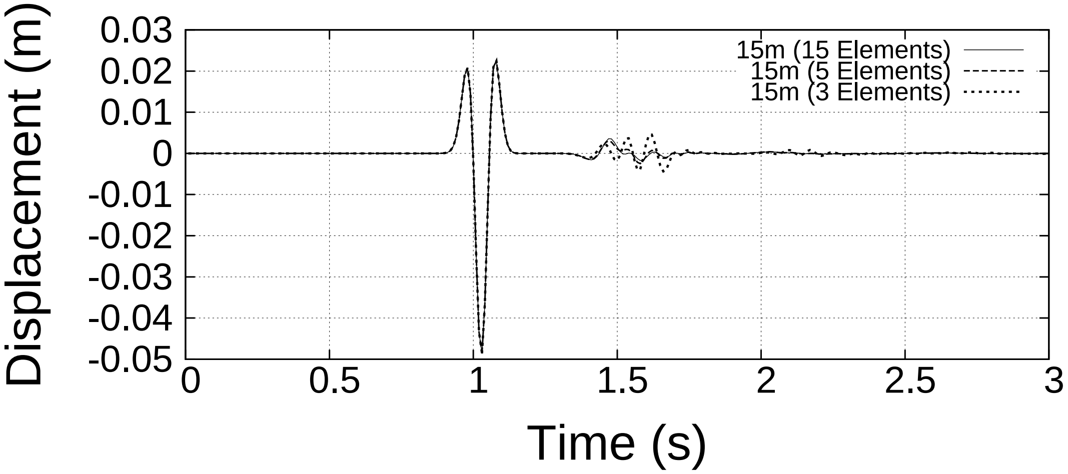


Figure 311.36: Displacement time history at a point in the non-damping zone close to the boundary of imposing motion

311.5 Mesh Size Effects for Linear (8 Node Brick) and Quadratic (27 Node Brick) Finite Elements on Wave Propagation

Generally, the results of numerical analysis using finite element method technique for the dynamic problem are affected by size of mesh (grid spacing). According to Argyris and Mlejnek (1991), about 10 nodes per wavelengths are required to simulate accurately for the given frequency and fewer than 10 nodes may induce an artificial damping due to the numerical reason.

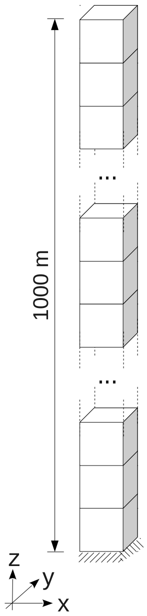


Figure 311.37: One dimensional column test model to inspect the mesh size effect

In this section, mesh size effect is inspected to decide an appropriate size of the mesh to build finite element model for verification. One dimensional column model is built as shown in figure 311.37. Total height of the model is 1000 m. Two models are built with element height of 20 m and 50 m, and each model have two different shear wave velocities (100 m/s and 1000 m/s). Density is set as 2000 kg/m^3, and Poisson's ratio is set as 0.3, for all test models. Various cases are set and tested as shown in table 311.1. Both 8 node and 27 node brick elements are used for all models. Thus, total 24 parametric study cases are inspected. Linear elastic elements are used for all analyses. All analyses are performed in time domain with Newmark dynamic integrator without any numerical damping (γ = 0.5, and β = 0.25, no numerical damping, unconditionally stable).

Ormsby wavelet (Ryan, 1994) is used as an input motion and imposed at the bottom of the model.

Table 311.1: Analysis cases to determine a mesh size

Case number	Vs (m/s)	Cutoff freq. (Hz)	Element height (m)	Max. propagation freq. (Hz)
1	1000	3	10	10
2	1000	8	10	10
3	1000	15	10	10
4	100	3	10	1
5	100	8	10	1
6	100	15	10	1
7	1000	3	20	5
8	1000	8	20	5
9	1000	15	20	5
10	100	3	20	0.5
11	100	8	20	0.5
12	100	15	20	0.5

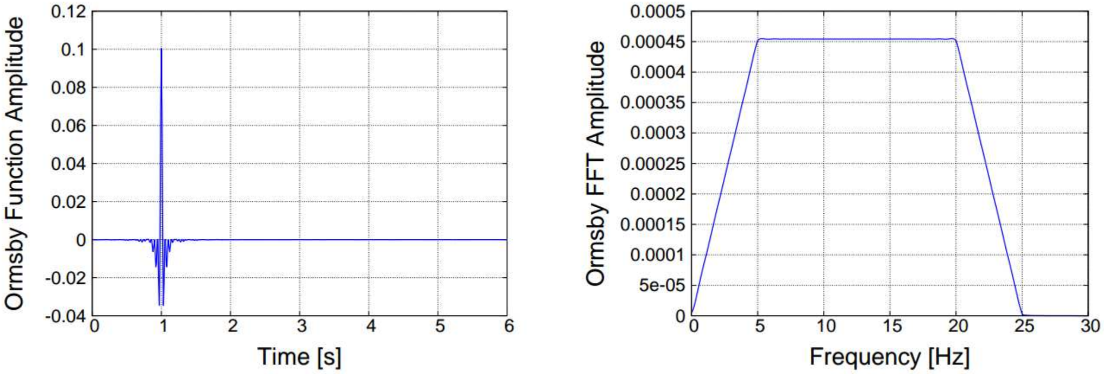


Figure 311.38: Ormsby wavelet in time and frequency domain with flat frequency content from 5 Hz to 20 Hz

Ormsby wavele features a controllable flat frequency content with formulation shown in equation 311.3.

$$
\begin{aligned}
f(t) = \quad & A\left(\left(\frac{\pi {f_4}^2}{f_4 - f_3} sinc(\pi f_4 (t - t_s))^2 - \frac{\pi {f_3}^2}{f_4 - f_3} sinc(\pi f_3 (t - t_s))^2\right)\right. \\
- \quad & \left(\frac{\pi {f_2}^2}{f_2 - f_1} sinc(\pi f_2 (t - t_s))^2 - \frac{\pi {f_1}^2}{f_2 - f_1} sinc(\pi f_1 (t - t_s))^2\right)
\end{aligned}
\tag{311.3}
$$

where f_1 and f_2 define the lower range frequency band, f_3 and f_4 define the higher range frequency band, A is the amplitude of the function, and t_s is the time that maximum amplitude is happening, and

$sinc(x) = sin(x)/x$. Figure 311.38 shows an example of Ormsby wavelet with flat frequency content from 5 Hz to 20 Hz.

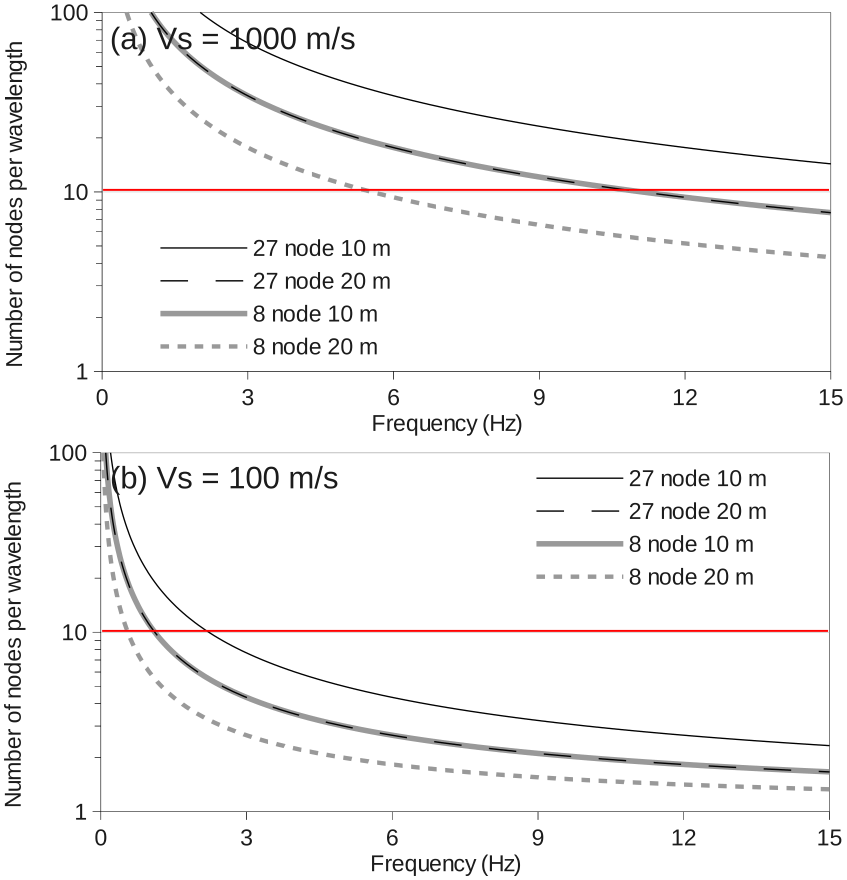


Figure 311.39: Number of nodes per wavelength along frequencies, and element sizes (a) Vs = 1000 m/s (b) Vs = 100 m/s

For this example, cutoff frequencies of Ormsby wavelets are set as 3, 8, and 15 Hz (table 311.1). Figure 311.39 shows number of nodes per wavelength along frequencies and figure 311.40 – 311.46 show comparison of analysis results. As shown in figure 311.40, case 1 and 7 (analysis using Ormsby wavelet with 3 Hz cutoff frequency) predict exactly identical results to the analytic solution in both time and frequency domain. Since, number of nodes per wavelength for both cases are over 10 (see figure 311.39(a) and table 311.1, all cases under 3 Hz shows more than 10 nodes per wavelength), those exact results are expected.

Increasing cutoff frequency from 3 Hz to 8 Hz induces numerical errors as shown in figure 311.41. In frequency domain, both 10 m and 20 m element height model with 27 node brick element predict exactly same results with the analytic one. However, in time domain, asymmetric shape of time history

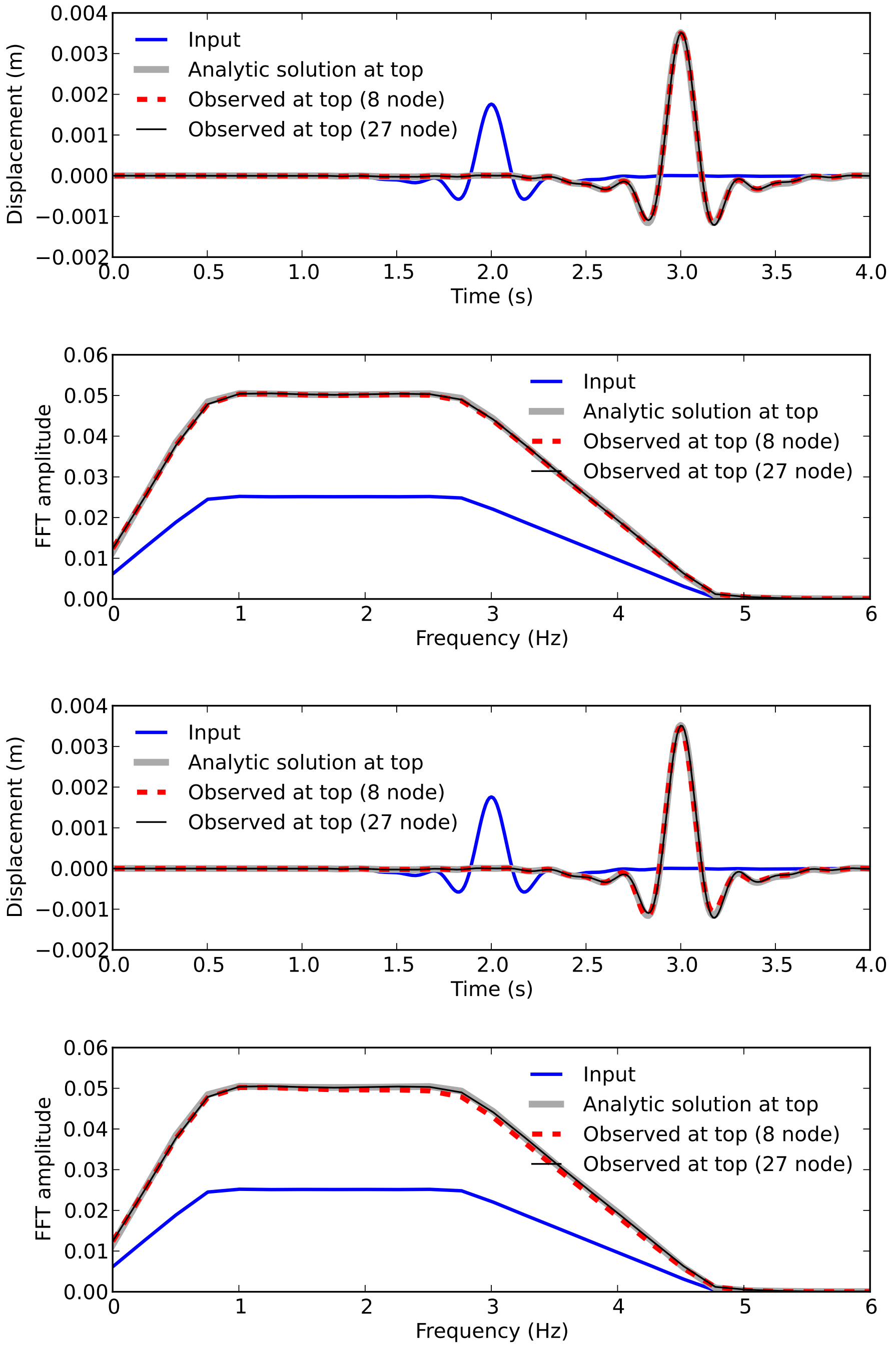


Figure 311.40: Comparison between (a) case 1 (top, Vs = 1000 m/s, 3 Hz, element size = 10m) and (b) case 7 (bottom, Vs = 1000 m/s, 3 Hz, element size = 20m)

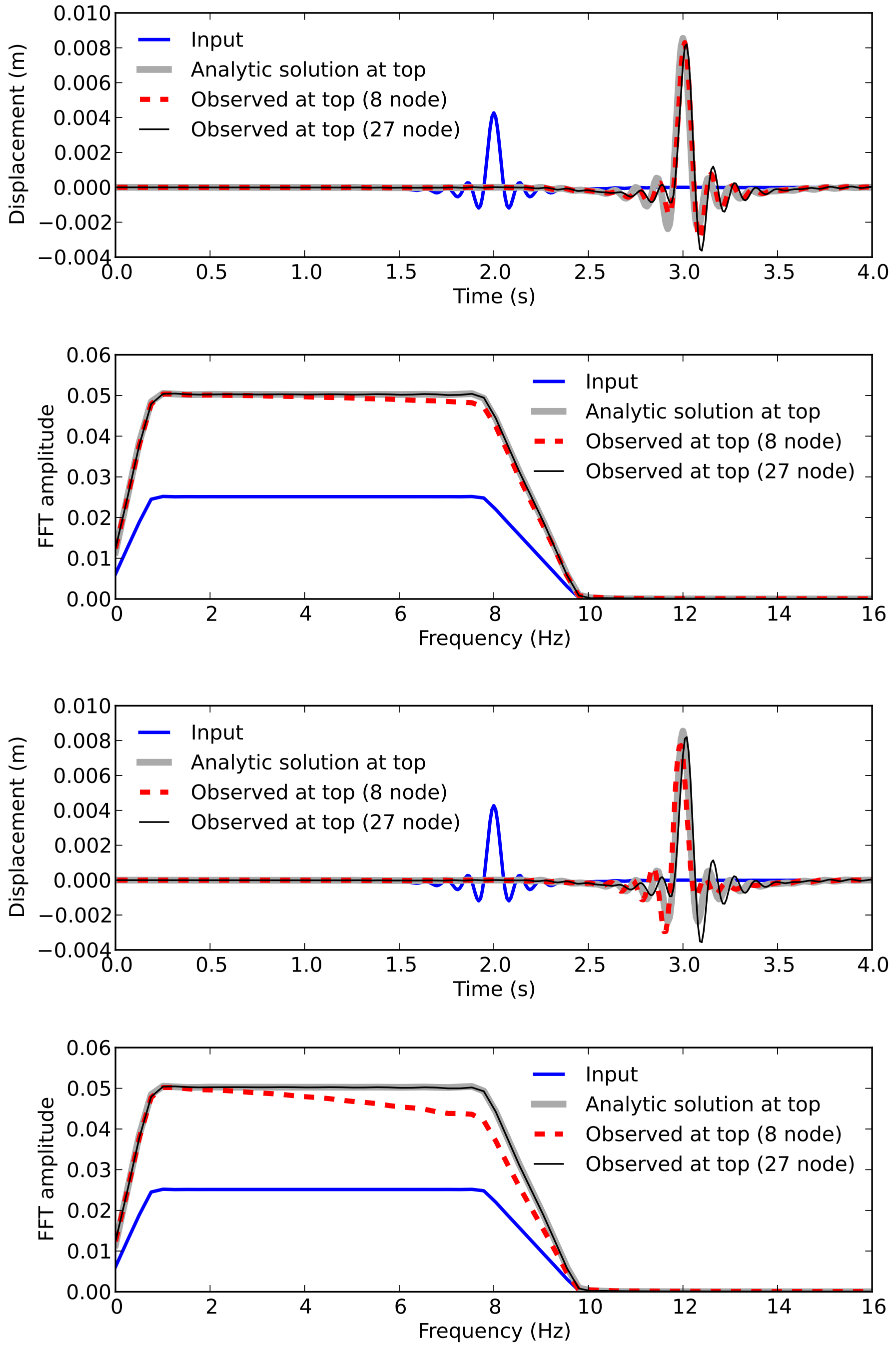


Figure 311.41: Comparison between (a) case 2 (top, Vs = 1000 m/s, 8 Hz, element size = 10m) and (b) case 8 (bottom, Vs = 1000 m/s, 8 Hz, element size = 20m)

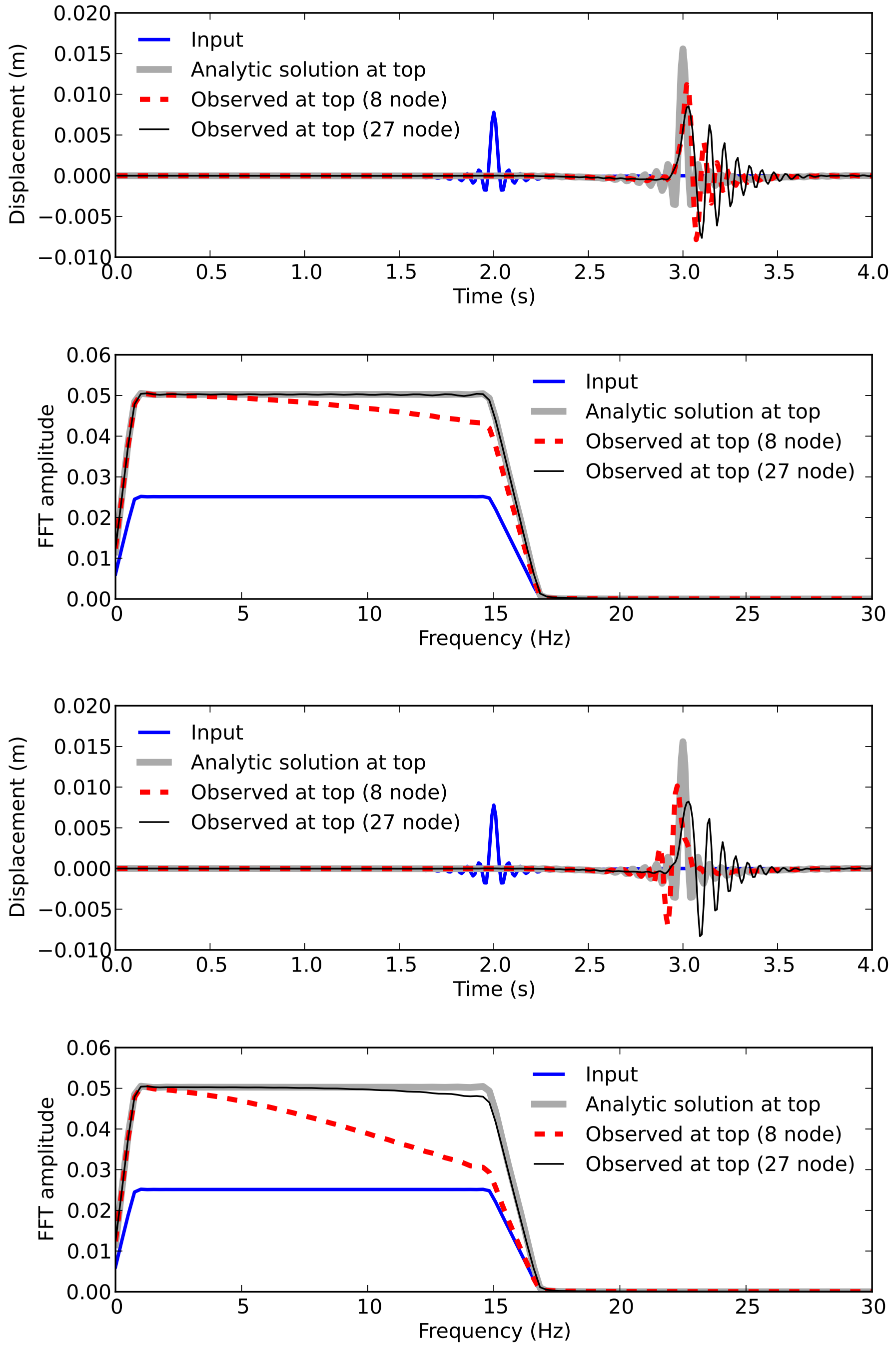


Figure 311.42: Comparison between (a) case 3 (top, Vs = 1000 m/s, 15 Hz, element size = 10m) and (b) case 9 (bottom, Vs = 1000 m/s, 15 Hz, element size = 20m)

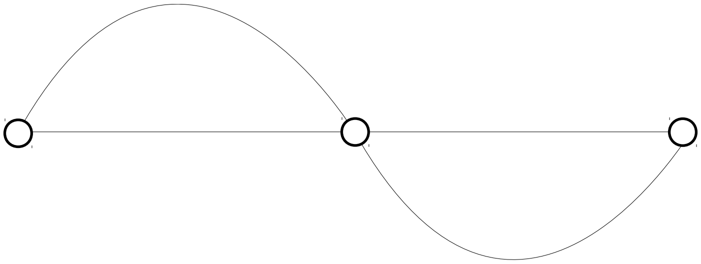


Figure 311.43: Schematic cartoon to show that nodes can behave like frequency contents filter

displacements are observed. Observations from top of 8 node brick element models show more numerical error in both time and frequency domain due to the decreasing number of nodes per wavelength (figure 311.39). Figure 311.42 shows analysis results with 15 Hz cutoff frequency. Results from 27 node brick elements are almost same in frequency domain but asymmetric shapes are also observed in time domain. Decreasing amplitudes in frequency domain along increasing frequencies are observed from 8 node brick element cases.

Figure 311.44 – 311.46 show results predicted from Vs = 100 m/s cases. Similar as Vs = 1000 m/s cases, decreasing amplitude along increasing frequencies are observed in all cases. One interesting observation is bumps in frequency domain which can be seen at natural frequencies (natural modes) of the elements (n th mode of elements, $f = (2n-1)Vs/4H$, 2.5 Hz, 5.0 Hz, and so on). This observation may mean that if certain condition is satisfied between modes and size of the element, it will behave like frequency contents filter. Figure 311.43 shows possible explanation of this observation. As in the case of figure 311.43, nodes (circle in the figure) cannot capture harmonic oscillation of the frequency since amplitude of the oscillation is always zero. As a result, the frequency contents at the frequency cannot be predicted by the analysis.

The results shown here are used as a reference to determine mesh size and frequency range of input motions for the verification.

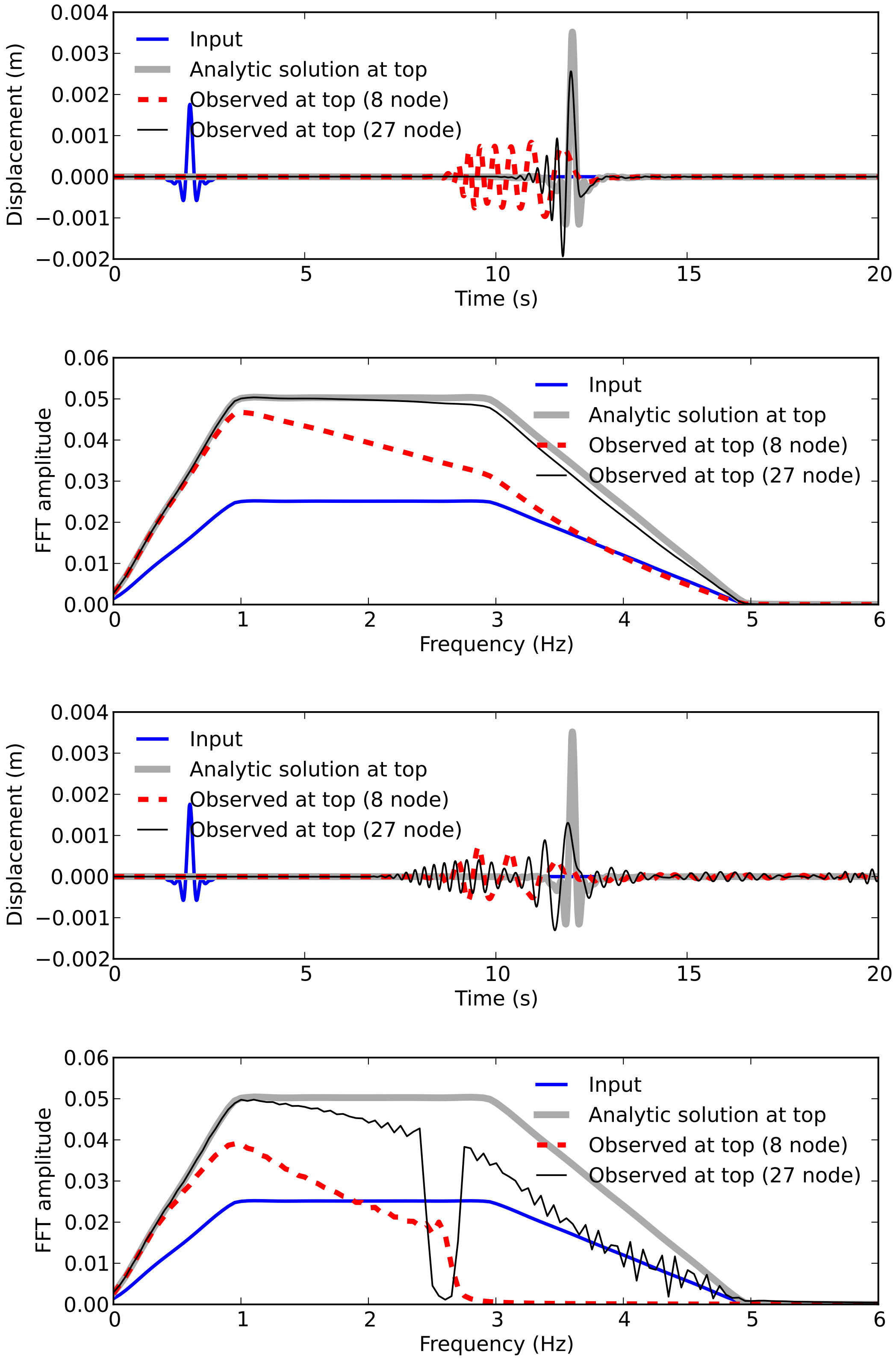


Figure 311.44: Comparison between (a) case 4 (top, Vs = 100 m/s, 3 Hz, element size = 10m) and (b) case 10 (bottom, Vs = 100 m/s, 3 Hz, element size = 20m)

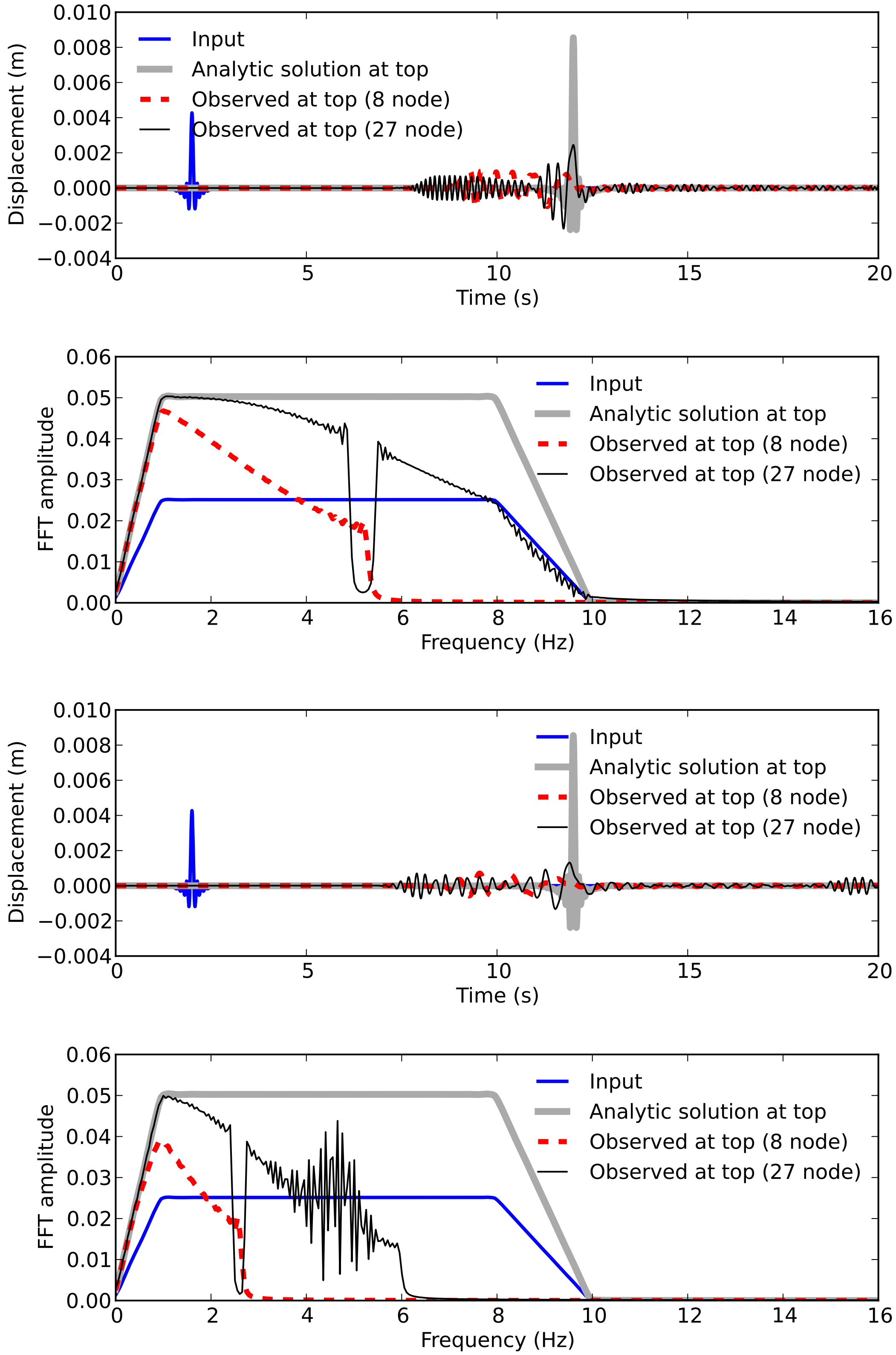


Figure 311.45: Comparison between (a) case 5 (top, Vs = 100 m/s, 8 Hz, element size = 10m) and (b) case 11 (bottom, Vs = 100 m/s, 8 Hz, element size = 20m)

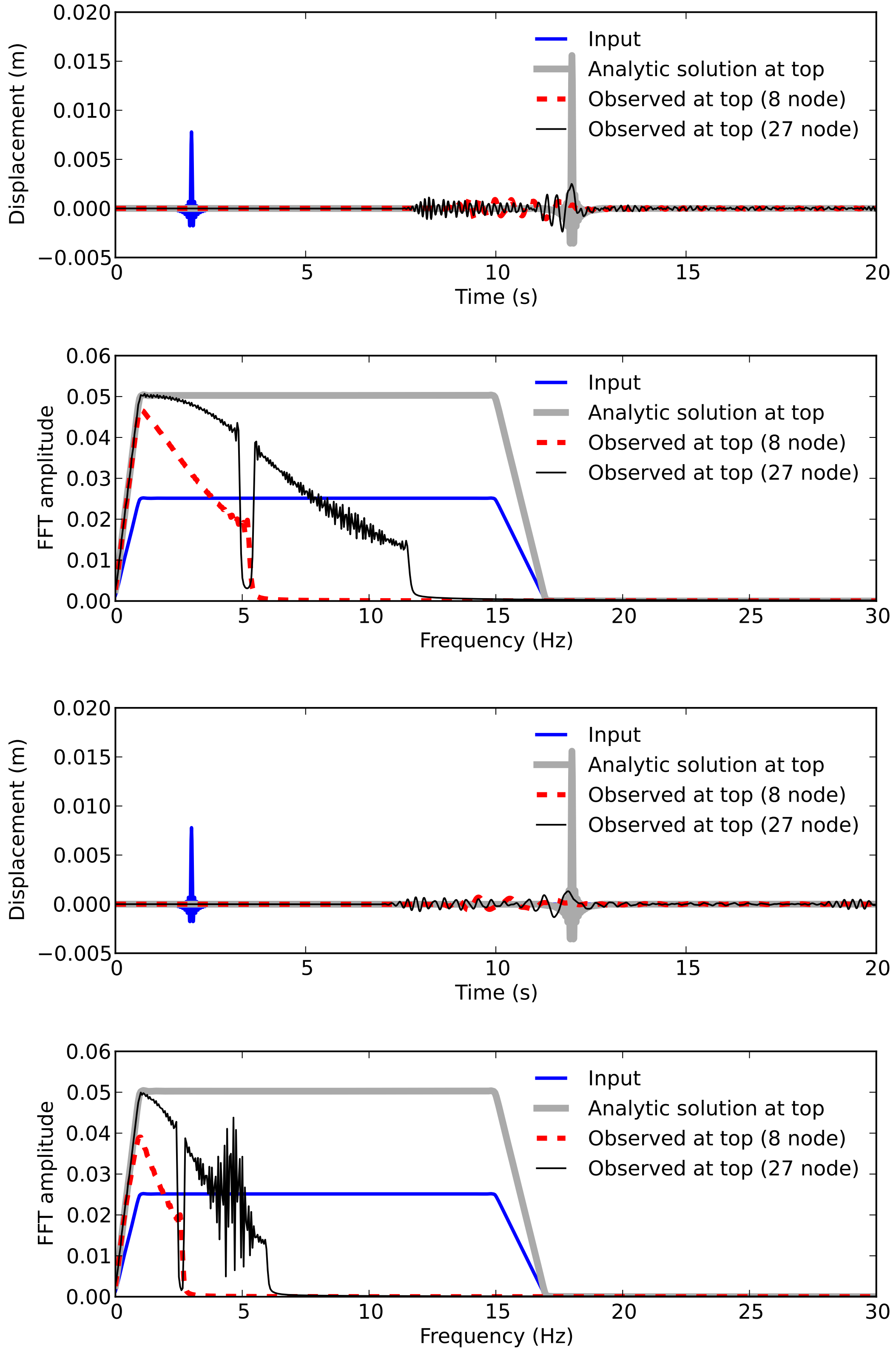


Figure 311.46: Comparison between (a) case 6 (top, Vs = 100 m/s, 15 Hz, element size = 10m) and (b) case 12 (bottom, Vs = 100 m/s, 15 Hz, element size = 20m)

311.6 Verification of the Seismic Input (Domain Reduction Method) for 3C, Inclined Seismic Wave Fields

311.6.1 Inclined, 3C Seismic Waves in a Free Field

In this section verification of the 3C wave propagation problem using Domain Reduction Method will be studied. In order to do so, a finite element model with dimensions of $10000m \times 50m \times 5000m$ is considered. Two cases are studied here with the source of motion (fault) to be located at $(x = 3000m, y = 0, z = 3000m)$ and $(x = 3000m, y = 0, z = 3000m)$. Figures (311.47) and (311.48) show these two models respectively.

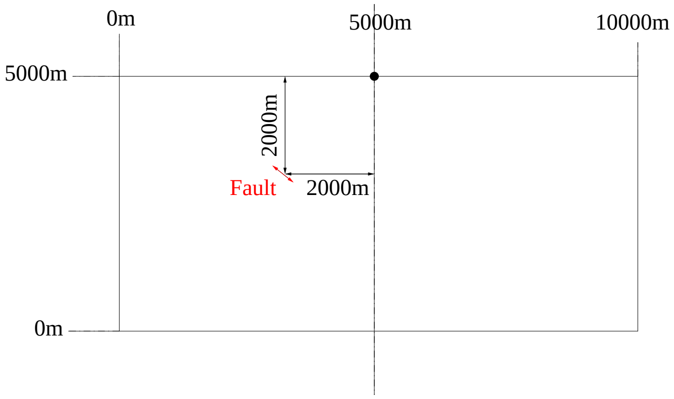


Figure 311.47: Domain to be analyzed for the 1^{st} stage of DRM with fault located at an angle of 45° with respect to the top middle point of the model

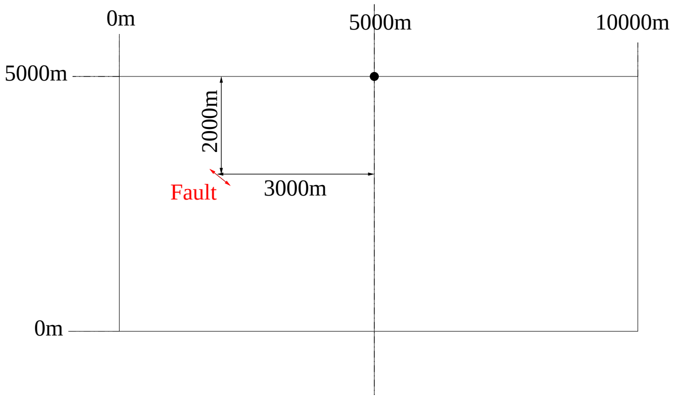


Figure 311.48: Domain to be analyzed for the 1^{st} stage of DRM with fault located at an angle of 34° with respect to the top middle point of the model

The size of the elements is chosen to be $50m$ in all directions for both cases in order to reduce the computational time. The soil parameters are: shear wave velocity of $700m/s$, density of $1800kg/m^3$, and Poisson's ration of 0.1. Analyses for the fault slip model are done by applying the motion at the nodes of one element. This is done in order to represent the the wave propagation starting from the fault using Multiple Support Excitation. This is representing the first stage of analysis of DRM in which a big model including the fault is considered for free field case in order to obtain the required motions for DRM layer. For simulating the second stage of DRM, a smaller model with dimensions of $240m \times 5m \times 70m$ is considered as shown in Figure (311.49). The size of the plastic bowl is $200m \times 5m \times 50m$. Size of the elements for this model is chosen to be $5m$.

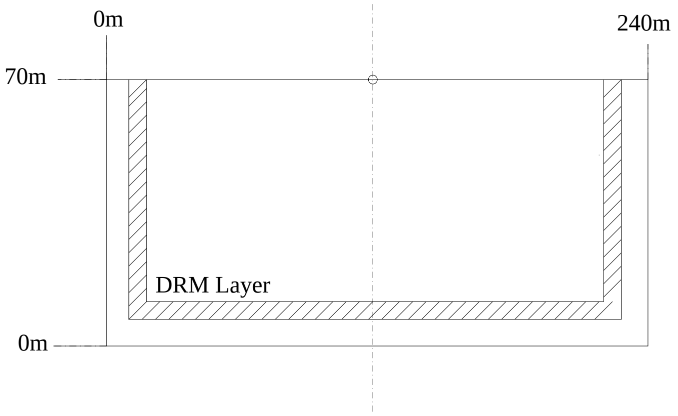


Figure 311.49: Domain to be analyzed for the 2^{nd} analysis stage of DRM with smaller size comparing to the original model

Displacement and acceleration time histories of corresponding nodes of DRM layer are obtained by interpolating between the the results obtained from the first model. These displacement and accelerations are used to calculate the effective forces as an input for DRM analysis. Input motions to be used here are Ricker wave, Morgan Hill, and Kocaeli earthquakes. The maximum allowable frequency to be propagated through this model can be calculated based on Equation (311.4):

$$\Delta h \leq \lambda/10 = V_s/(10 f_{max}) \tag{311.4}$$

Based on the shear wave velocity of $700m/s$ and element size of $50m$, maximum allowable frequency to be propagated through this model would be $1.4Hz$ for the original model and based on element size of $5m$ would be $14Hz$ for the DRM model.

311.6.1.1 Ricker Wavelets

Figure (311.50) show the displacement time history and FFT of Ricker wave of 2^{nd} order with dominant frequency of $1 Hz$ and maximum amplitude occurring at 1 second.

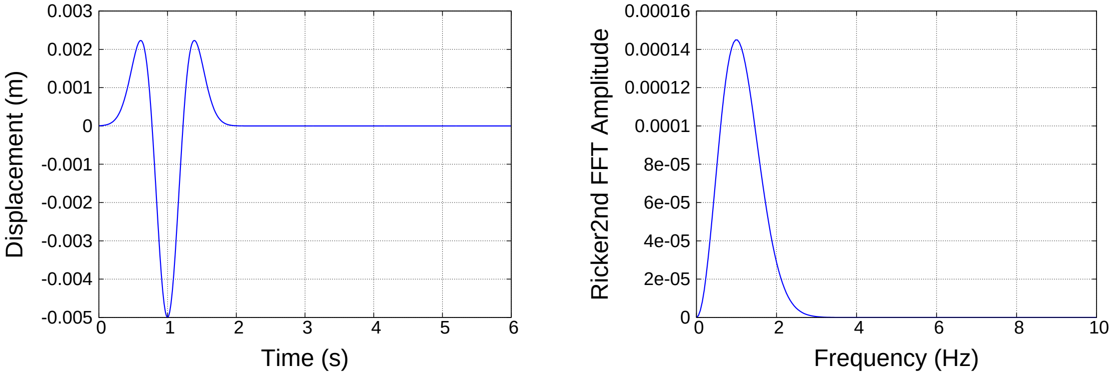


Figure 311.50: Displacement time history and FFT of Ricker wave with dominant frequency of $1 Hz$

The first case to be studied here is the one with the fault source located at $(x = 3000m, y = 0, z = 3000m)$ which has the angle of 45° with respect to the top middle point of the model. Results to be discussed here are comparison of displacement and acceleration time histories at the top middle point of the model $(x = 5000m, y = 0, z = 5000m)$ between the fault slip and DRM models. Comparison of displacement time histories in X and Z directions are shown in Figure (311.51). As it can be observed, the results of DRM model matches perfectly with the ones obtained from the fault slip model.

Figure (311.53) is the displacement and acceleration time history of a point located outside of DRM layer in X direction $(x = 10m, y = 0, z = 40m)$. As mentioned before in definition of DRM, no motion should come out of the DRM layer in case of free field. As shown in these figures, displacement and acceleration time histories at this point are zero which verifies this fact.

The same motion is applied to the model with fault source located at $(x = 2000m, y = 0, z = 3000m)$ which has the angle of 34° with respect to the top middle point of the model. Displacement time histories of the top middle point show the perfect match between results obtained from fault slip model with the ones obtained from DRM mode.

As shown in Figure (311.55), the second motion to be used for analysis is Ricker wave with frequency of $0.5 Hz$ and maximum amplitude occurring at 3 seconds. Figure (311.56) shows the displacement time histories of X and Z directions for the same point as before $(x = 5000m, y = 0, z = 5000m)$. As it is shown, results of the fault slip and DRM model are the same which verifies the solution from DRM formulation for this motion as well.

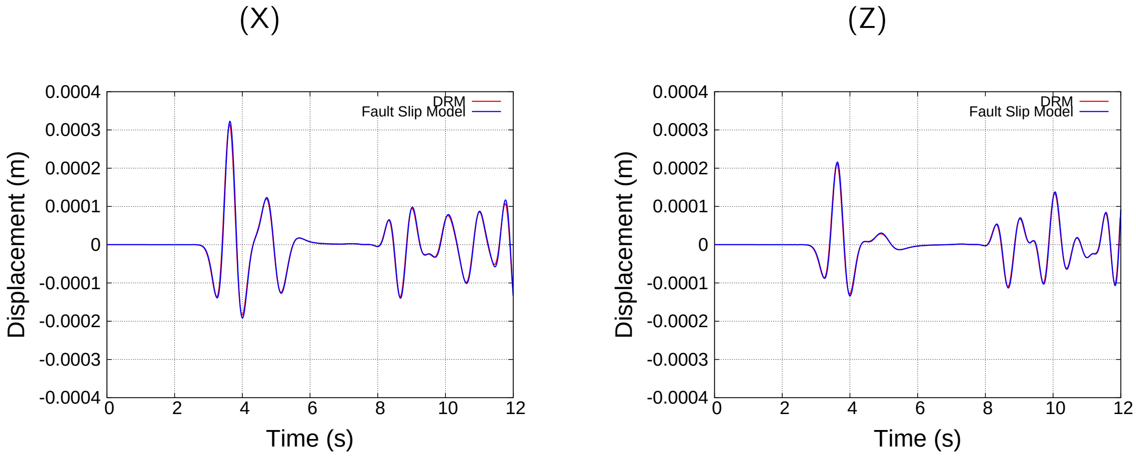


Figure 311.51: Comparison of displacements for top middle point using Ricker wave ($f = 1Hz$) as an input motion

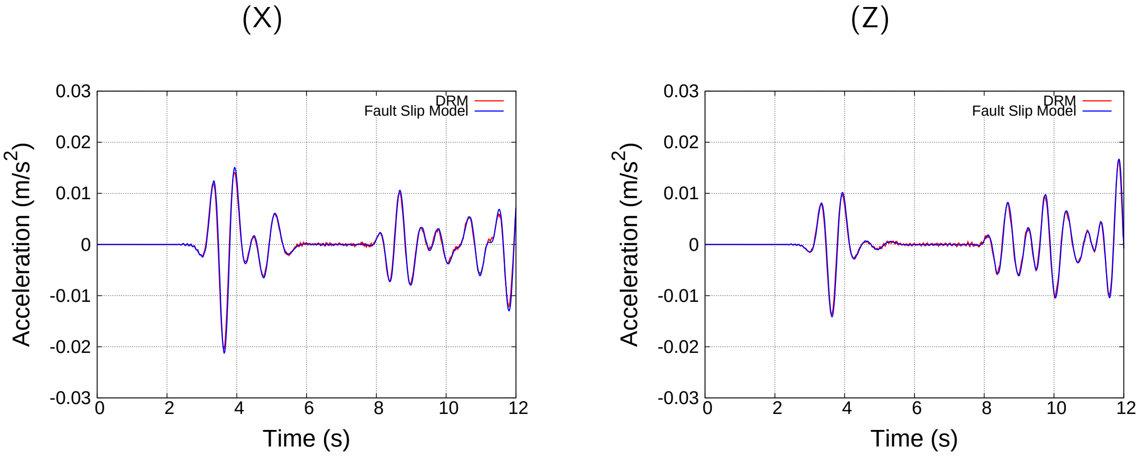


Figure 311.52: Comparison of accelerations for top middle point using Ricker wave ($f = 1Hz$) as an input motion

The third motion to be used is Ricker wave with frequency of $2Hz$ and maximum amplitude happening at 1 second as shown in Figure (311.57). Comparison of displacement time histories between the fault slip and DRM model has been done and shown in Figure (311.58) along X and Z directions respectively.

In this case, results do not match for the top middle point of the model. The main reason is due to the frequency of the motion. The maximum allowable frequency to be propagated in the fault slip model is $1.4Hz$ while it is $14Hz$ in DRM model. Dominant frequency of the Ricker wave as input motion is $2Hz$. Frequencies above the $1.4Hz$ can not be propagated in the fault slip model while they will propagate in

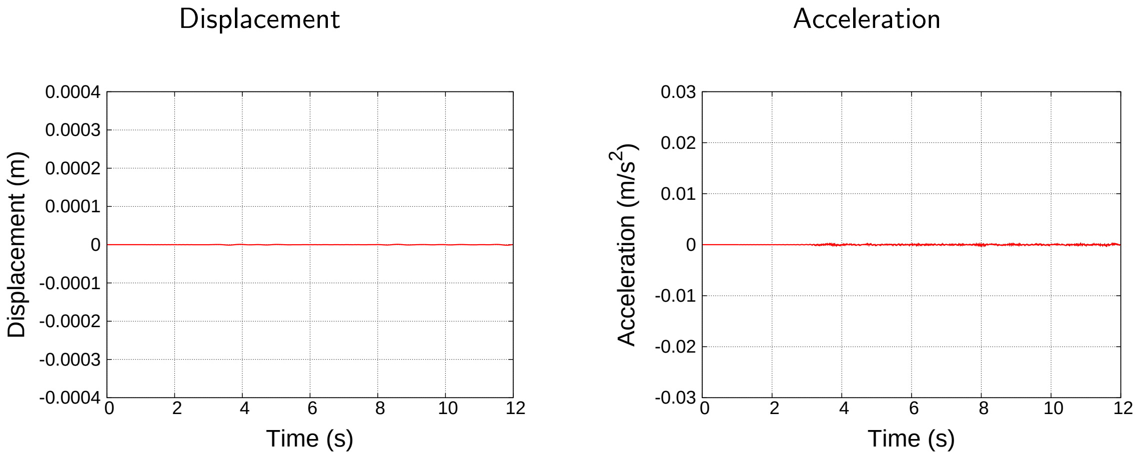


Figure 311.53: Displacement and acceleration time history for a point outside of DRM layer in (x) direction

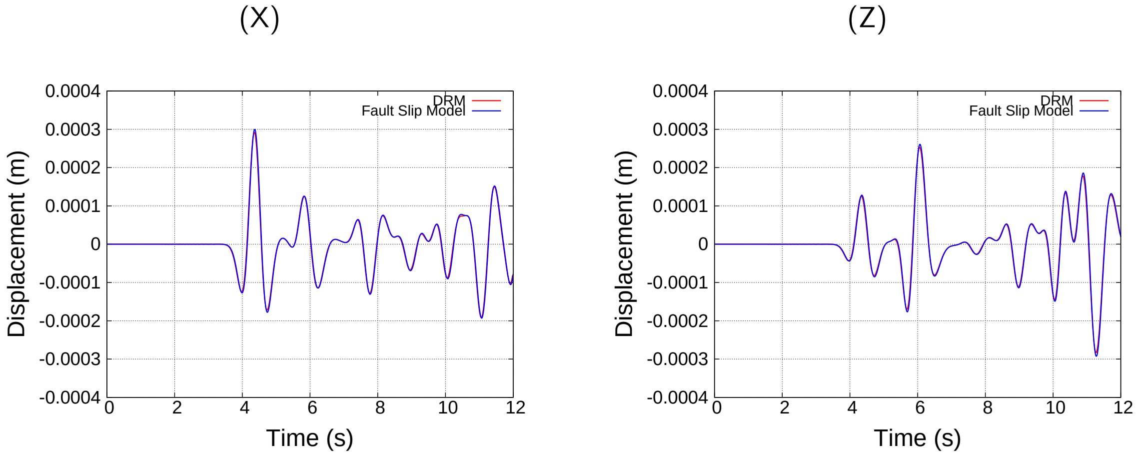


Figure 311.54: Comparison of displacements for top middle point using Ricker wave ($f = 1Hz$) as an input motion

the DRM model. this can change the characteristics of the motion propagating through the model and is the main reason of differences between the obtained results.

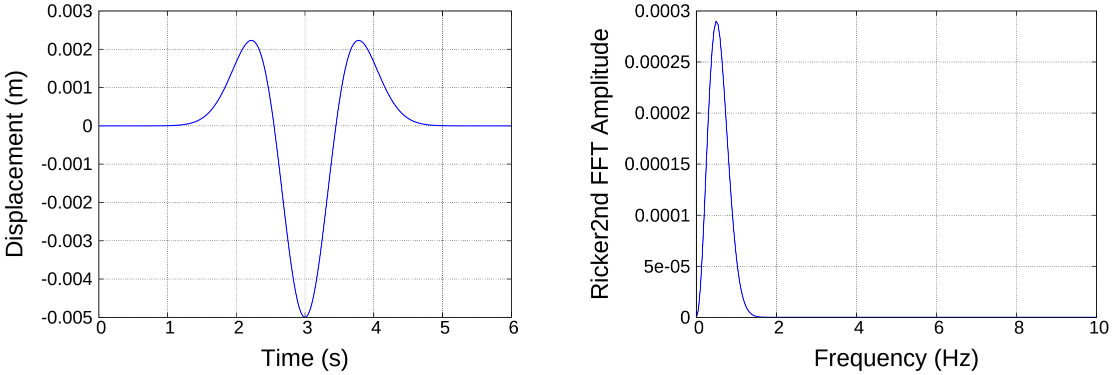


Figure 311.55: Displacement time history and FFT of Ricker wave with dominant frequency of $0.5Hz$

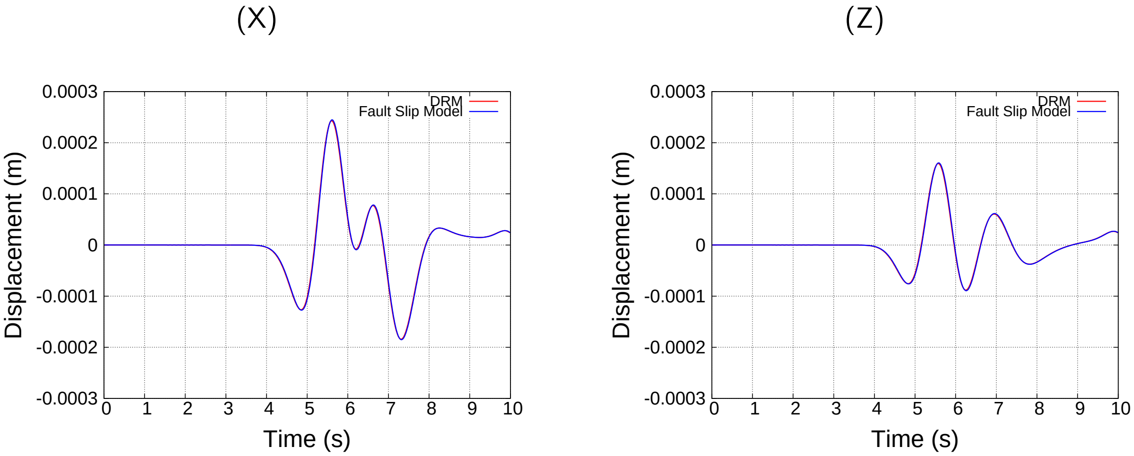


Figure 311.56: Comparison of displacements for top middle point using Ricker wave ($f = 0.5Hz$) as an input motion

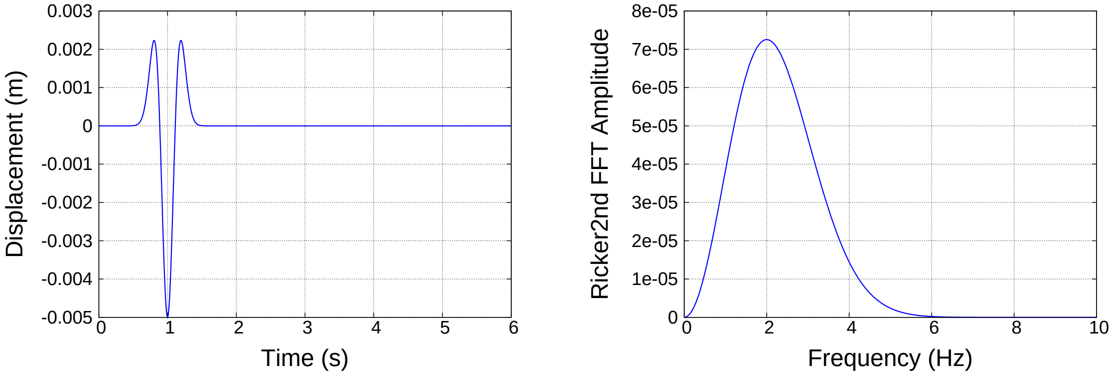


Figure 311.57: Displacement time history and FFT of Ricker wave with dominant frequency of $2Hz$

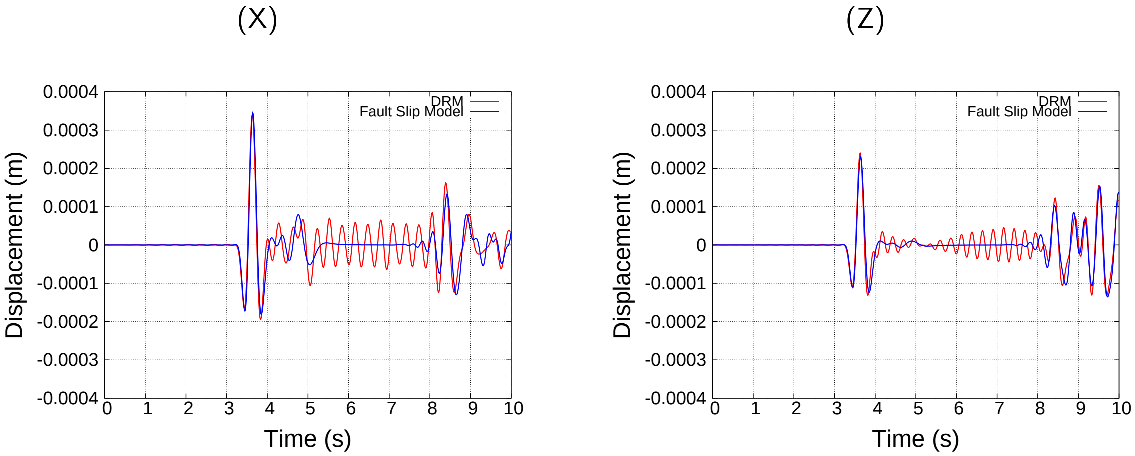


Figure 311.58: Comparison of displacements for top middle point using Ricker wave ($f = 2Hz$) as an input motion

311.6.2 Vertical (1C) Seismic Waves in a Free Field

311.6.2.1 Morgan Hill and Kocaeli Earthquakes

In order to investigate more, Morgan Hill and Kocaeli earthquakes are used as an input motions for the same models as before. These earthquakes were recorded during the ground shaking and obtained from PEER motion database. Figure (311.59) shows the acceleration time history and FFT of Morgan Hill earthquake with major frequency range of up to $4Hz$. Acceleration time history and FFT of Kocaeli earthquake are shown in Figure (311.60). Major part of the frequency range for Kocaeli earthquake is up to frequency of $4Hz$.

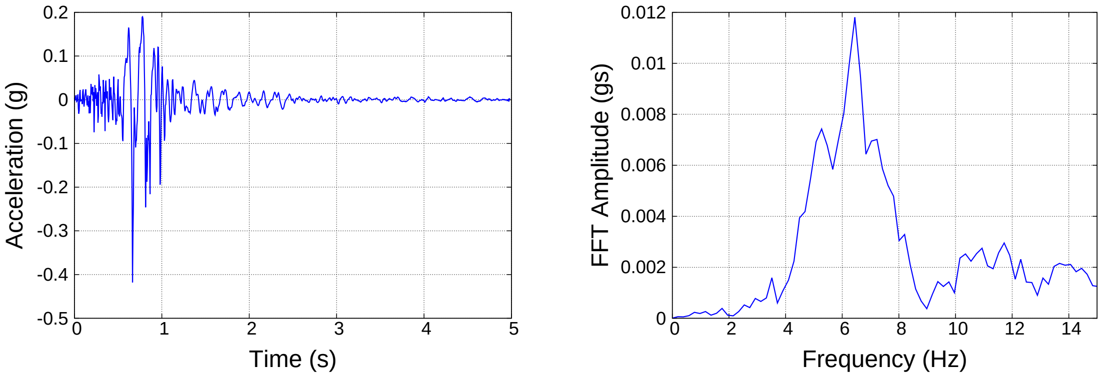


Figure 311.59: Acceleration time history and FFT of Morgan Hill earthquake

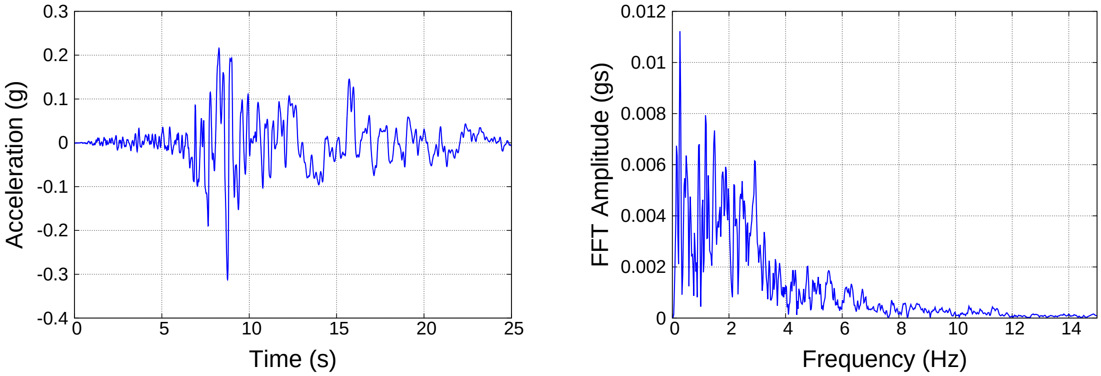


Figure 311.60: Acceleration time history and FFT of Kocaeli earthquake

Figure (311.61) shows the displacement time histories of the top middle point of the model for

Morgan Hill earthquake while the ones from Kocaeli earthquake are shown in Figure (311.62). As it is observed, results of fault slip model and DRM model do not match since the majority of the energy in the earthquake is in the range of up to $4Hz$ which is higher than the maximum allowable frequency to be propagated in the original model $(1.4Hz)$.

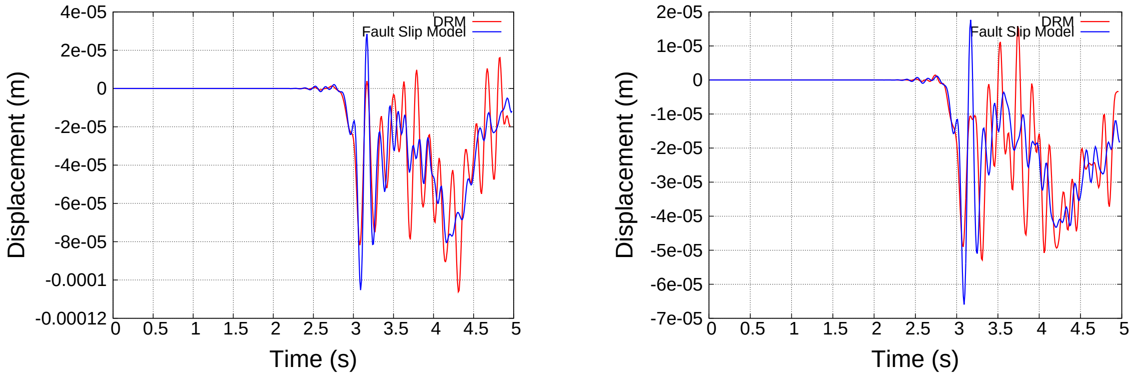

(X) (Z)

Figure 311.61: Comparison of displacements for top middle point using Morgan Hill earthquake as an input motion

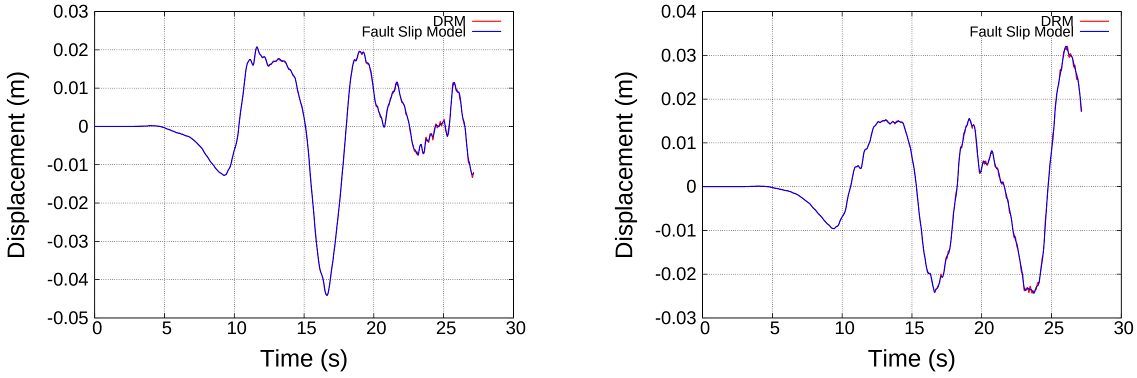

(X) (Z)

Figure 311.62: Comparison of displacements for top middle point using Kocaeli earthquake as an input motion

In order to investigate more regarding the frequency content issue, Kocaeli acceleration time history is considered and frequencies above $1.4Hz$ are filtered out of the record. Acceleration time history and

FFT of the filtered record are shown in Figure (311.63). The majority of the energy is in the frequency range of below $1.4Hz$ while still there are frequencies up to $2Hz$ in the motion as can be observed in FFT of the filtered motion.

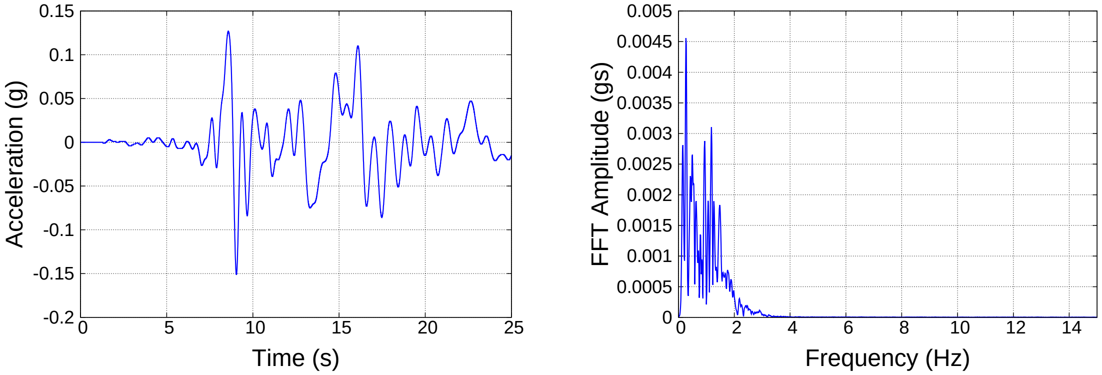
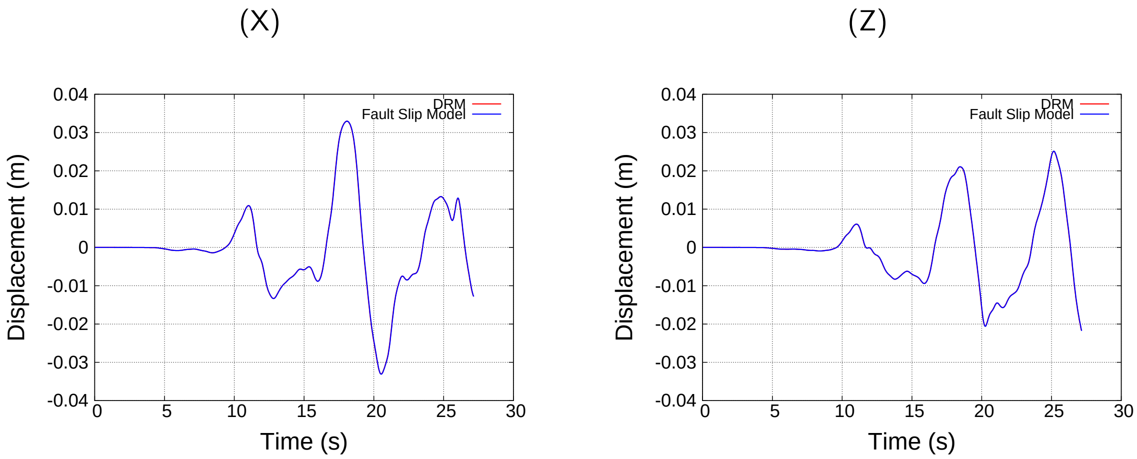


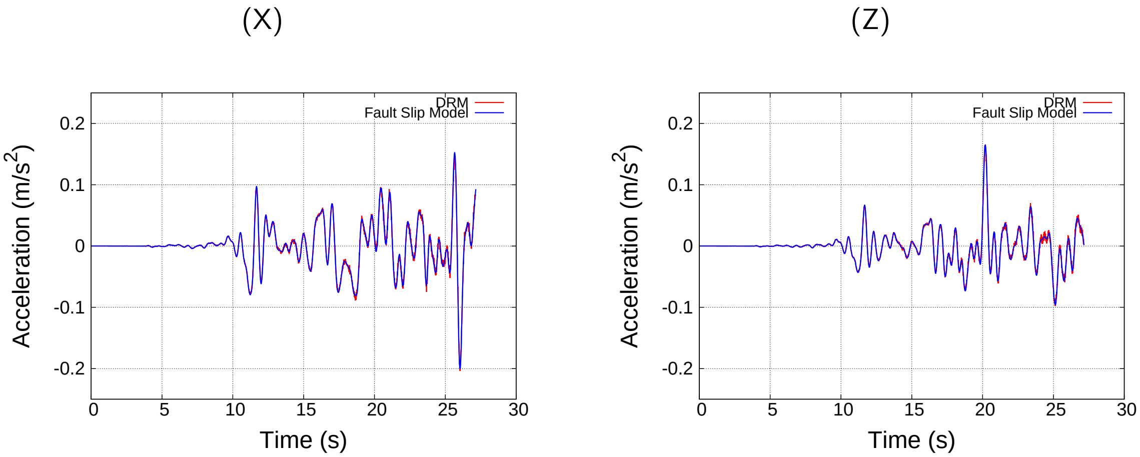
Figure 311.63: Acceleration time history and FFT of filtered Kocaeli earthquake

Figure (311.64) shows the displacement time histories for the same point as the one studied for the actual record. As it is observed, the obtained time histories match perfectly between the case of fault slip and DRM models. Figure (311.65) shows the acceleration time histories. Comparing the time histories shows an acceptable match between the results. There are tiny differences in acceleration time histories (specially at the peaks) which can be due to the fact that there are still frequencies above $1.4Hz$ in the input motion but with much less impact in terms of amplitude.


(X)

(Z)

Figure 311.64: Comparison of displacements for top middle point using filtered Kocaeli earthquake as an input motion


(X)

(Z)

Figure 311.65: Comparison of accelerations for top middle point using filtered Kocaeli earthquake as an input motion

311.6.3 Earthquake-Soil-Structure Interaction Verification for Simulated Northridge Seismic Motions

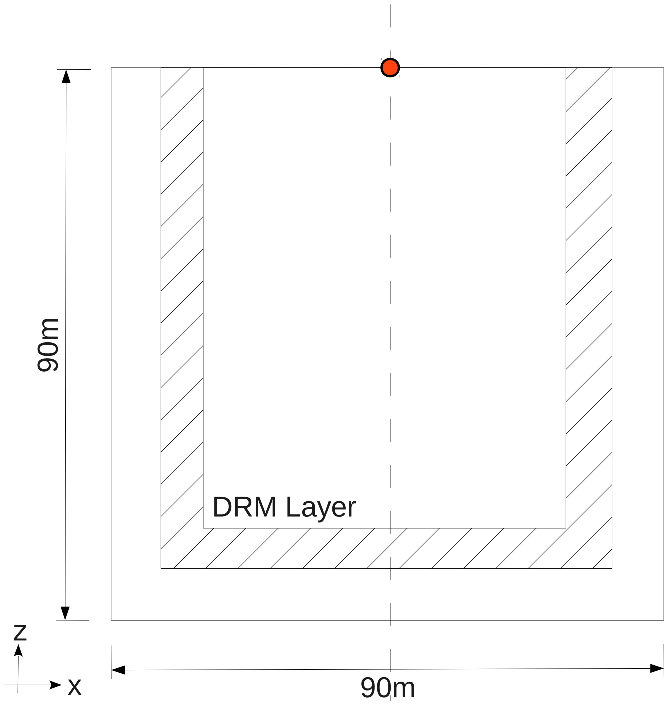


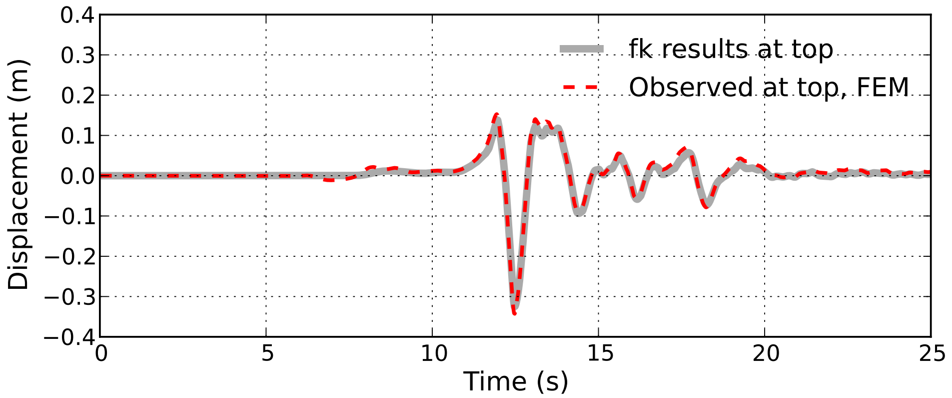
Figure 311.66: Finite element model to be used on analyses with input motions computed by integration equation (x-z plane view)

More realistic example is shown here. Seismic wave fields of Northridge earthquake simulated by program fk are applied as an input motion for this example. Figure 311.66 shows x-z plane view of three dimensional model. Similar as analytic case, using fk program, acceleration and displacement fields are generated at all nodes in DRM layer.

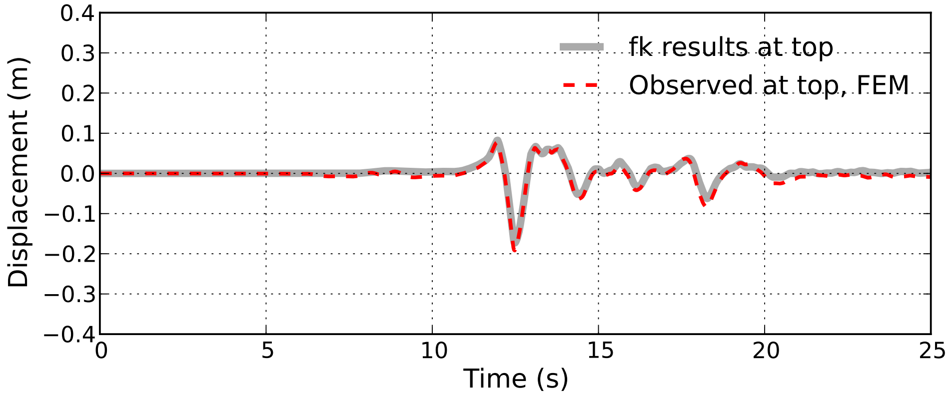
Figure 311.67 shows analysis results observed at the top-midpoint of the finite element model. As shown in figure 311.67, both results show perfect match.

311.6.4 Curious Case of 1C versus 3C modeling

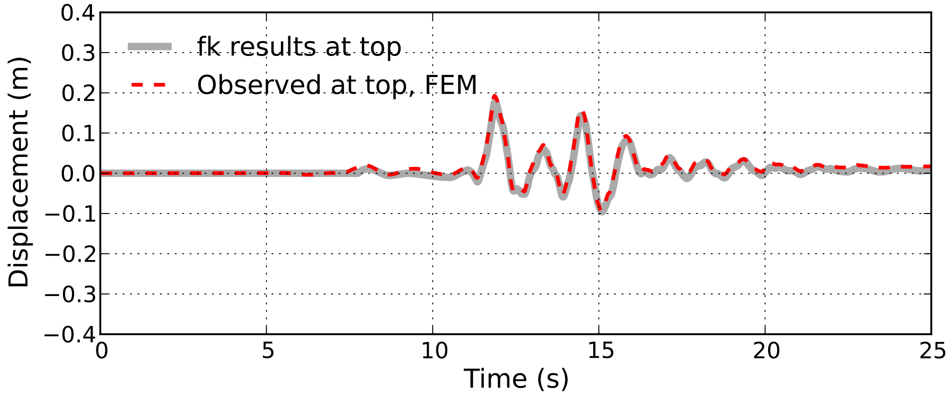
To inspect more, artificial downhole array is prepared as shown in fingure 311.68. Total 2 observation points are set on 0 m, and 50 m depth from the ground surface. one dimensional site response analyses are performed along artificial downhole array usind DEEPSOIL v5.0 (Hashash and Park, 2002). 1D soil column model is built to run DEEPSOIL with identical soil properties to finite element model. Linear time domain site response analyses are performed. Displacements recorded at 200 m depth are used as an input motion. Site response analyses results on the observation points are compared with fk, and



.pdf



.pdf



.pdf

Figure 311.67: Comparison between results computed from program fk and finite element analysis, observed at the top middle point of the finite element model

finite element analyses results.

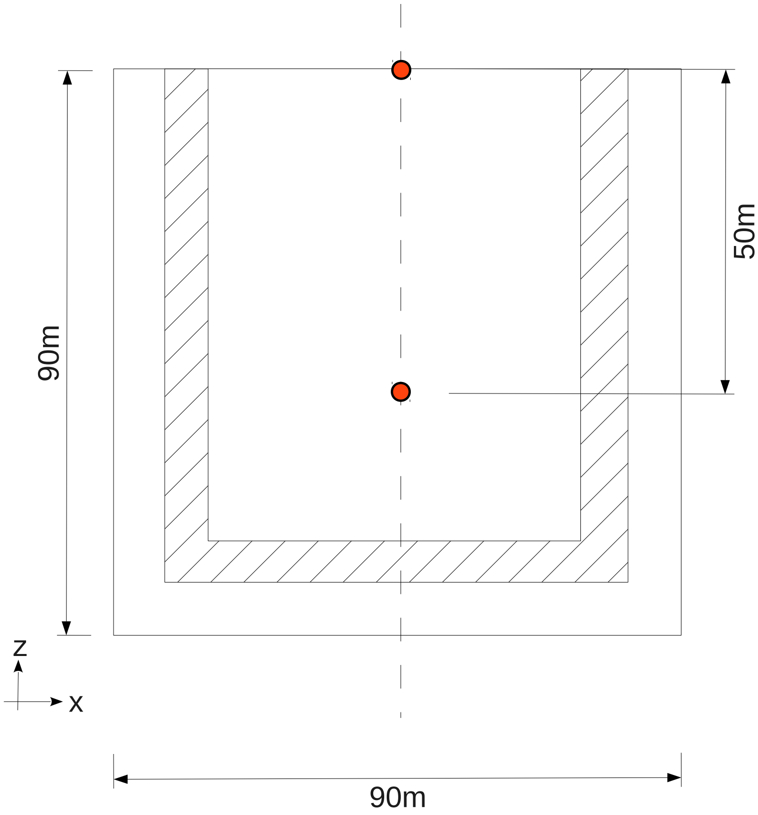


Figure 311.68: Comparison between analytic solution and FEM analysis result observed at top, middle point of the model (SV (imposed on x direction) Ricker wave input with 0°, x component)

Figure 311.69 – 311.74 are analyses results. Figure 311.69, 311.70, and 311.71 show comparison of results observed at the ground surface, EW, NS, and UD components, respectively. For the case of EW and NS components, 1C site response analyses results predict similar response as fk and FEM results compared to UD case. For all cases, 1C analyses results shows larger amplitude especially on UD case, 1Hz frequency contents show unrealistic response amplification. The same trend can be observed at 50 m depth cases (figure 311.72 – 311.74).

Possible explanation are as follows. fk results includes all components of waves (body and surface) and interaction between them. However, 1C wave propagation analyses cannot incorporate such effect. Also, 1C analyses is very sensitive to material properties (stiffness, damping ratio, and so on) and frequency contents of input waves.

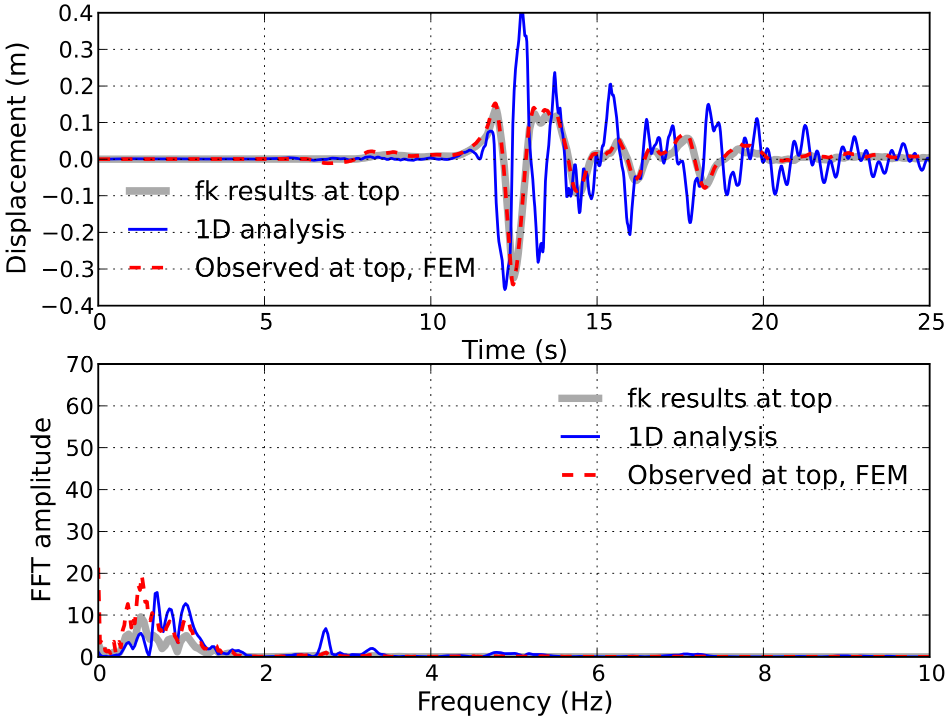


Figure 311.69: Comparison between results computed from program fk, finite element analysis, and 1C analysis, observed at the top middle point, EW component

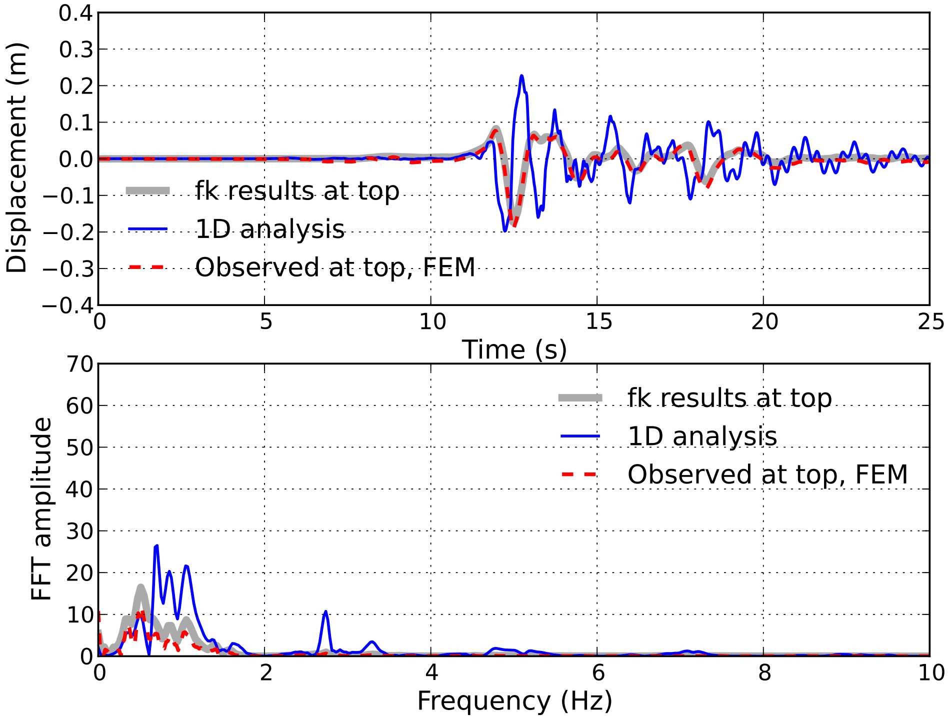


Figure 311.70: Comparison between results computed from program fk, finite element analysis, and 1C analysis, observed at the top middle point, NS component

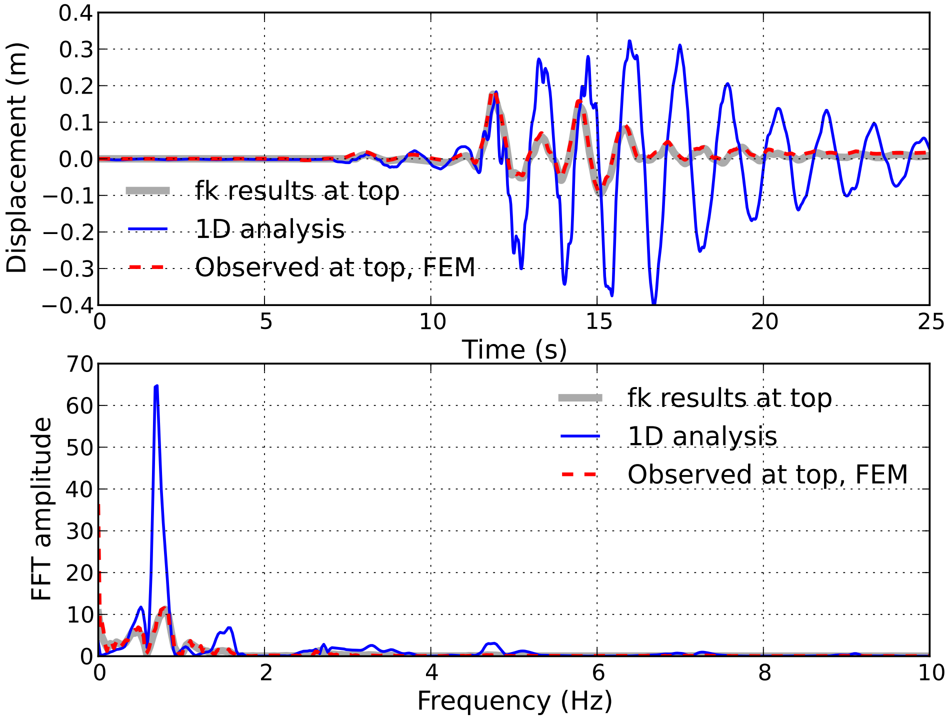


Figure 311.71: Comparison between results computed from program fk, finite element analysis, and 1C analysis, observed at the top middle point, UD component

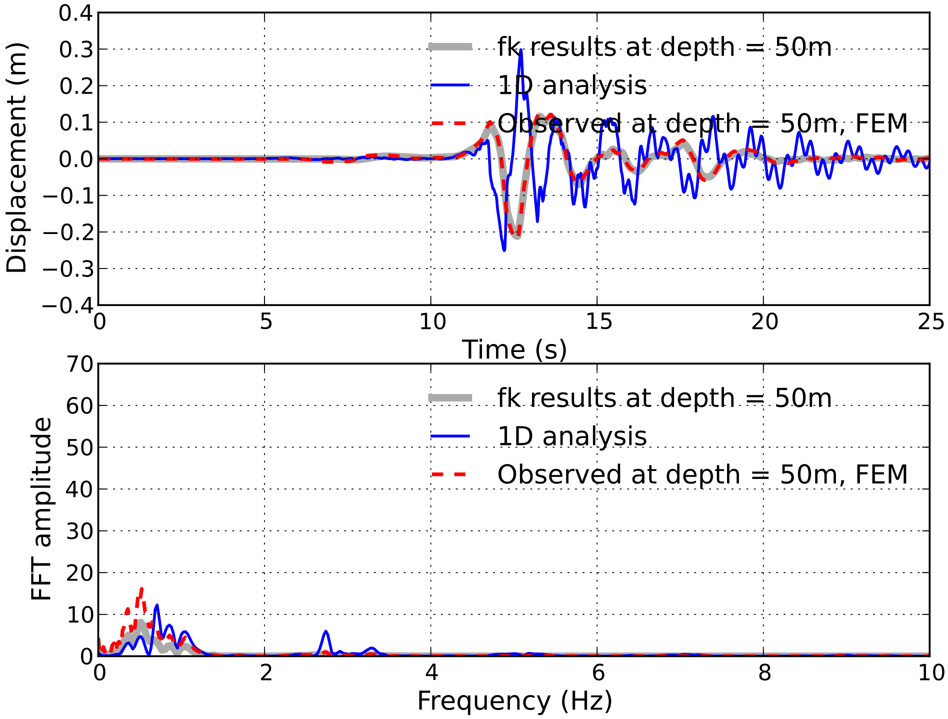


Figure 311.72: Comparison between results computed from program fk, finite element analysis, and 1C analysis, observed at the depth = 50m, EW component

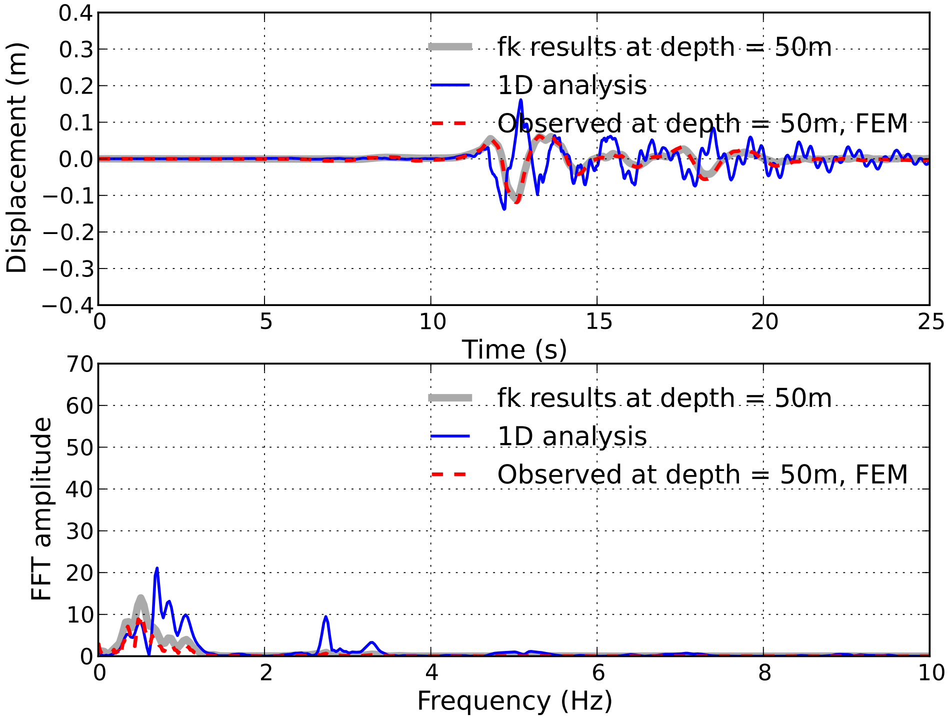


Figure 311.73: Comparison between results computed from program fk, finite element analysis, and 1C analysis, observed at the depth = 50m, NS component

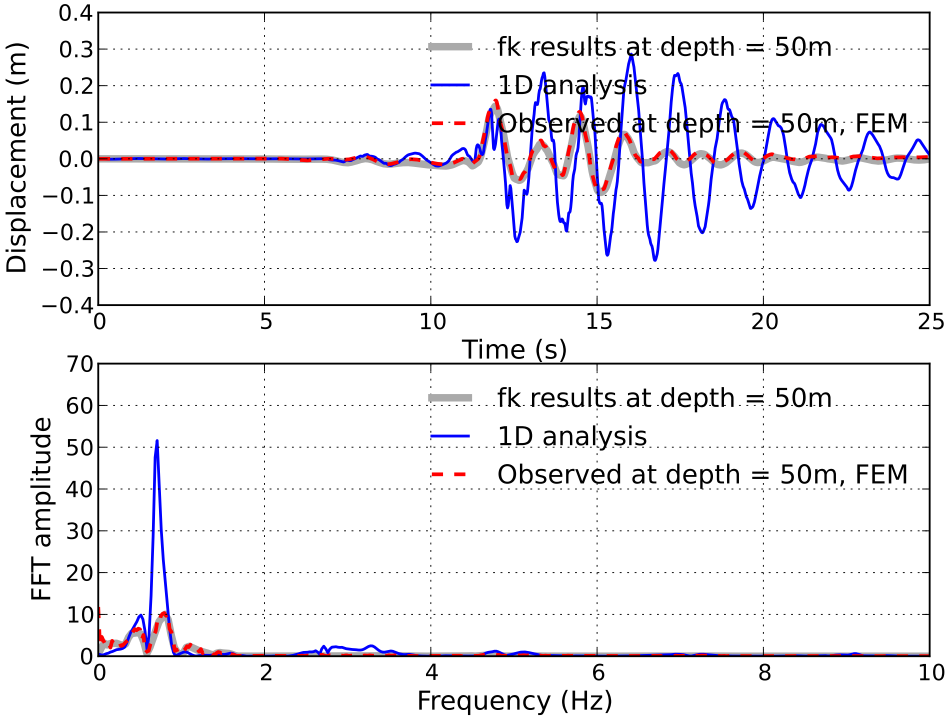


Figure 311.74: Comparison between results computed from program fk, finite element analysis, and 1C analysis, observed at the depth = 50m, UD component

311.6.5 Earthquake-Soil-Structure Interaction for Surface and Embedded Structures

Luco (1974) Gazetas and Roesset (1979) Wong and Luco (1978) Iguchi and Luco (1981) Kausel and Roesset (1975) Wong and Luco (1985) Day (1977) Zhang and Chopra (1991) Papageorgiou and Pei (1998) Luco et al. (1990) Pais and Kausel (1989) Apsel and Luco (1987) Gazetas and Roesset (1979) Kausel and Roesset (1975)

311.6.5.1 Uniform half-space

311.6.5.2 Layered half-space

311.6.5.3 Layered Layered over rigid lower boundary

311.7 Case History: Simple Structure on Nonlinear Soil

311.7.1 Simplified Models for Verification

Due to the complexity of full scale finite element models it is helpful to perform preliminary tests on simplified models in order to verify the adequacy of the time and mesh discretization with respect to the input motion. It also provides good insight in the performance of the nonlinear material model. To achieve this a series of tests on a one-dimensional soil column have been proposed:

- Static pushover test on nonlinear soil column

  Through the static pushover test the behavior of the nonlinear material model can be verified.

- Dynamic test of elastic soil column

  By applying an earthquake motion to the elastic soil column it can be tested whether the selected grid spacing is capable of representing the motion correctly without filtering out any relevant frequencies. This test also allows to choose appropriate damping parameters. It should be noted that this is additional (small) damping that is used for stability of the numerical scheme and should not be relied upon to provide major energy dissipation. Major energy dissipation should be coming from inelastic deformations of the SFS system.

- Dynamic test of nonlinear soil column

  Finally the stability and the accuracy of the numerical method can be examined by applying the earthquake motion to the nonlinear column of soil. A second analysis with a time step reduced by 50% should not give a significantly different result.

  Furthermore it will be examined how propagation through an elastic-plastic material will change the frequency content of the motion.

311.7.1.1 Model Description

The one-dimensional soil column used for verification has the same depth and element sizes as the 2d and 3d models that will be addressed later. Its total depth is 10.5 meters and it consists of a single stack of 8-node brick elements of 1.5 meters side length. In order to achieve one-dimensional wave propagation in vertical direction the movement of four nodes at each level of depth is constrained to be equal. The input motion is applied to the four nodes at the base of the model. As input motions four time histories from the Northridge Earthquake are selected (Figure 311.75).

The material properties of the soil are given in Table 311.7.1.1.

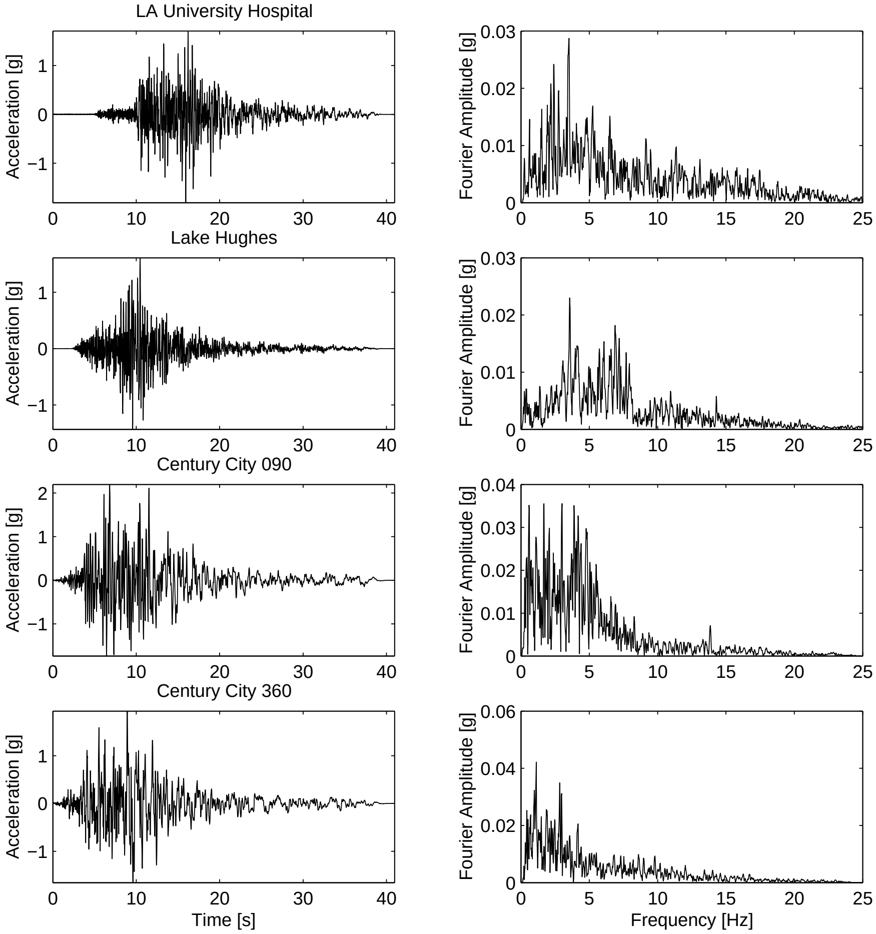


Figure 311.75: Acceleration time histories and Fourier amplitude spectra's of the selected ground motions

Friction angle ϕ'	37°
Undrained shear strength c_u	10 kPa
Mass density ρ	1800 kg/m^3
Shear wave velocity v_s	200 m/s

The discretization parameters, the time step Δt and the maximum grid spacing Δh, are determined following the guidelines outlined in Section 502.3.3. This yields a maximum grid spacing of

$$\Delta h \leq \frac{v_s}{10 f_{max}} = \frac{200}{10\,10} = 2\,\mathrm{m} \tag{311.5}$$

For the following analysis $\Delta h = 1.5$ m is selected. The maximum time step is

$$\Delta t \leq \frac{\Delta h}{v_s} = \frac{1.5}{200} = 0.0075\,\mathrm{s} \tag{311.6}$$

Taking into account a further reduction of the time step by about 60% due to the use of nonlinear material models $\Delta t = 0.002$ s is chosen.

311.7.1.2 Static Pushover Test on Elastic-Plastic Soil Column

For the static pushover test, an elastic perfectly plastic Drucker-Prager material model as specified in Table 311.7.1.1 is used.

After applying self weight a horizontal load of 100 kN is applied to a surface node in increments of 0.1 kN. The system of equations is solved using a full Newton-Raphson algorithm. The predicted shear strength of the first element that is expected to fail, the one at the surface, is:

$$\begin{aligned}\tau_f &= c_u + z\,\rho\,g\,\tan\phi' \\ &= 10 + 0.75\,x\,1.8\,x\,9.81\,\tan 37^\circ \\ &= 19.98\,\mathrm{kPa}\end{aligned} \tag{311.7}$$

where z is the depth of the center of the first element.

Self weight produces the following stresses in the element at the surface:

$$\begin{aligned}\sigma_x = \sigma_y &= 8.83\,\mathrm{kPa} \\ \sigma_z &= 13.24\,\mathrm{kPa}\end{aligned}$$

The maximum shear stress is

$$\tau_{max} = \sqrt{\left(\frac{\sigma_z - \sigma_x}{2}\right)^2 + \tau_{xz}^2} \tag{311.8}$$

The theoretical failure load can be obtained as follows:

$$\begin{aligned}P_f &= \tau_{xz}A \\ &= \sqrt{\tau_f^2 - \left(\frac{\sigma_z - \sigma_x}{2}\right)^2}A \\ &= 44.7\,kN\end{aligned} \tag{311.9}$$

The static failure load is underestimated by about 6%. This accuracy is acceptable for the given model because the boundary conditions cannot assure constant stresses at a given depth (no shear stress is applied to the lateral surfaces).

311.7.1.3 Dynamic Test on Elastic Soil Column

In order to test the spatial discretization of the model an earthquake motion is propagated through an elastic soil column. The grid spacing of the finite element mesh can be considered sufficiently fine if frequencies up to f_{max} = 10 Hz are represented accurately in the numerical analysis. A good way to verify this is to calculate transfer functions between the base and the surface of the soil column.

Because transfer functions don't depend on the input motion they can easily be compared with closed form solutions.

The transfer function of a soil deposit describes the amplification between the frequencies of the motion at the base and at the soil surface:

$$TF(\omega) = \frac{u(z = 0, \omega)}{u(z = H, \omega)} \tag{311.10}$$

where z is the depth measured from the surface and H is the thickness of the soil deposit above the bedrock. $\omega = 2\pi f$ is the circular frequency.

For elastic soil with viscous damping the wave equation can be written as (Kramer, 1996a)

$$\rho \frac{\partial^2 u}{\partial t^2} = G \frac{\partial^2 u}{\partial z^2} + \eta \frac{\partial^3 u}{\partial z^2 \partial t} \tag{311.11}$$

η is the damping coefficient, defined as

$$\eta = \frac{2G}{\omega} \xi \tag{311.12}$$

where ξ is the frequency independent hysteretic material damping.

After solving the wave equation the transfer function can be written as

$$TF(\omega) = \frac{1}{\cos \omega H / v_s^*} \tag{311.13}$$

where v_s^* is the complex shear wave velocity

$$v_s^* = \sqrt{\frac{G^*}{\rho}} = \sqrt{\frac{G(1 + i2\xi)}{\rho}} \tag{311.14}$$

In a finite element model with mass- and stiffness proportional Rayleigh damping the damping coefficient η is constant. Therefore the hysteretic material damping ratio ξ needs to be frequency dependent in order to satisfy Equation 311.12. Solving Equation 311.12 for ξ and substituting it into Equation 311.14 and then into Equation 311.13 yields a new transfer function:

$$TF(\omega) = \frac{1}{\cos\left(\omega H \sqrt{\frac{\rho}{G + i\omega\eta}}\right)} \tag{311.15}$$

Figure 311.76 shows a comparison between the closed form solution and the numerical transfer functions obtained from the finite element analysis. Rayleigh damping is used to obtain the damping matrix $\mathbf{C}$:

$$\mathbf{C} = \alpha \mathbf{M} + \beta \mathbf{K} \tag{311.16}$$

The analysis are performed using stiffness proportional Rayleigh damping of $\beta = 0.001$ and $\beta = 0.01$. No mass proportional damping is applied ($\alpha = 0$). The damping coefficients of the closed form solution are chosen to be $\eta = \beta G$.

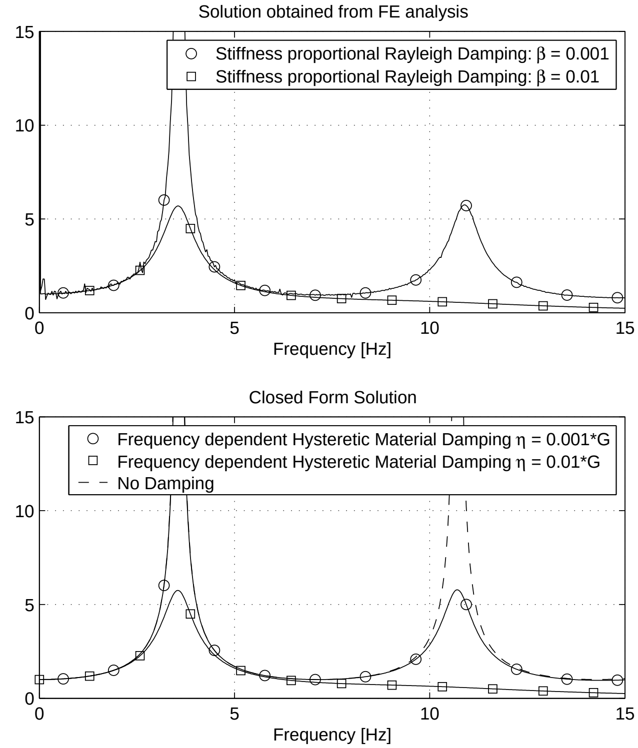


Figure 311.76: Transfer Function between Surface and Base of Soil Layer

It can be seen that the numerical transfer functions are very close to the closed form solutions for $\eta = \beta G$. The peak corresponding to the second natural frequency of the soil layer is slightly shifted to the right in the result of the FE analysis. For the FE analysis the Rayleigh damping cannot be reduced any further as the solution would become unstable. This result proves that a FE analysis involving Rayleigh damping with $\alpha = 0$ and $\beta = \eta / G$ is equivalent to the closed form solution of the wave equation with frequency-dependent hysteretic material damping.

Based on the above observations a stiffness proportional Rayleigh damping of $\beta = 0.01$ is selected for the finite element analysis. This choice damps frequencies above 10 Hz appropriately.

311.7.1.4 Dynamic Test on Elastic-Plastic Soil Column

As the next step an elastic-plastic material model of Drucker-Prager type with kinematic strain hardening has been selected. Previous analysis involving material with isotropic hardening have proved to be unsuitable because energy can only be dissipated as the yield surface expands. For dynamic problems this can lead to an unreasonably large extension of the yield surface, especially if resonance frequencies are present. Therefore only kinematic hardening has been selected in this analysis.

The analysis were performed with four different ground motions using time steps of $\Delta t = 0.002 s$ and

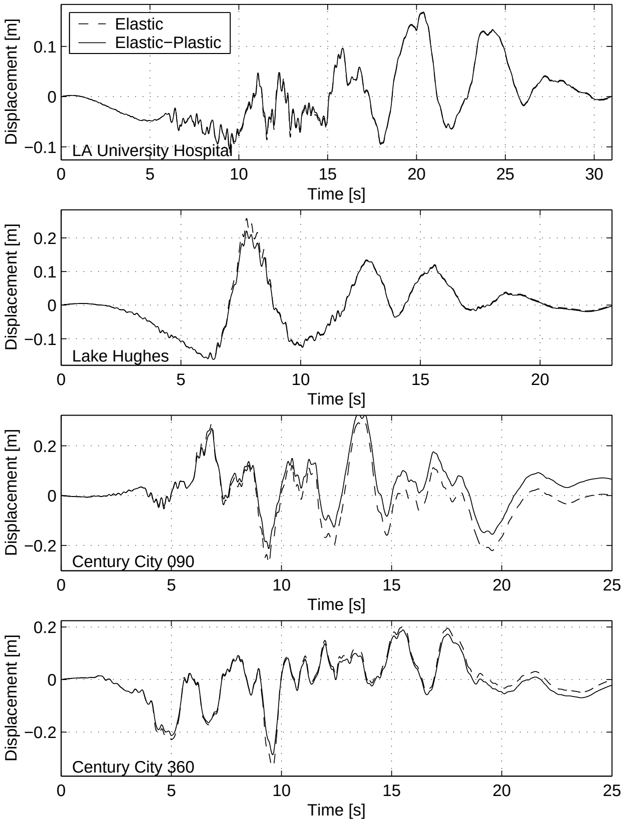


Figure 311.77: Displacement Time-Histories at surface of 1d Soil Column, elastic and elastic-plastic material

$\Delta t = 0.001s$. A linear integrator without iterations within a time step was used. All ground motions were scaled to a maximum acceleration of $1g$. For comparison the analysis were also performed on elastic material. Figure 311.77 shows the displacement time histories at the surface for all four ground motions. While the overall shapes of the displacements are the same as for the elastic case there is some residual plastic displacement resulting in the time histories of the Century City motions.

The Fourier amplitude spectra's of the acceleration recorded at the surface (Figure 311.78) have the same general shape for the case of elastic and elastic-plastic material. The amplification at the first resonance frequency ($f = 4.75Hz$) is bigger in the elastic analysis. Higher frequencies resulting from plastic slip are damped out effectively in the nonlinear analysis.

Figure 311.79 shows the acceleration time history at the upper node of the lowest element, that is the first free node above the base. The record shows large peaks of the order of about 6 g. These peaks are caused by plastic slip and counter balancing of the resulting plastic deformation. The periods of the peaks are of the order of a few time steps, they add a very high frequency component to the

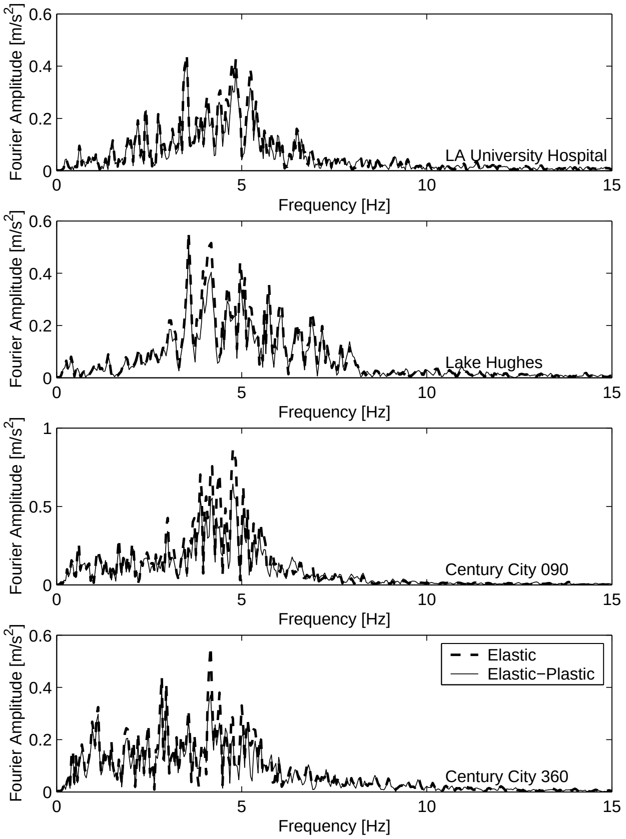


Figure 311.78: Fourier Amplitudes at surface of 1d Soil Column

acceleration. Because these frequencies are due to a purely numerical phenomenon, they should not be allowed to propagate through the model. This can be achieved easily by specifying an appropriate numerical procedure (Newmark with appropriate combination of γ and β) or with Rayleigh Damping.

As for the elastic model transfer functions were also computed for the nonlinear model. In Figure 311.80 the transfer functions between the acceleration at the soil surface and the base are compared. The functions for the nonlinear model are not smooth anymore but the general shape is the same as for the linear elastic model, i.e. the first natural frequency of the layer is clearly visible. The peaks that are present in the range of 25 Hz are purely numerical as they appear due to the division by a very small value.

A second set of analysis performed with half the time step of the previous analysis gives an idea of the accuracy of the numerical method. In Figure 311.81 the difference between the displacement (or acceleration) of the analysis with $\Delta t = 0.002s$ and $\Delta t = 0.001s$, divided by the corresponding maximum

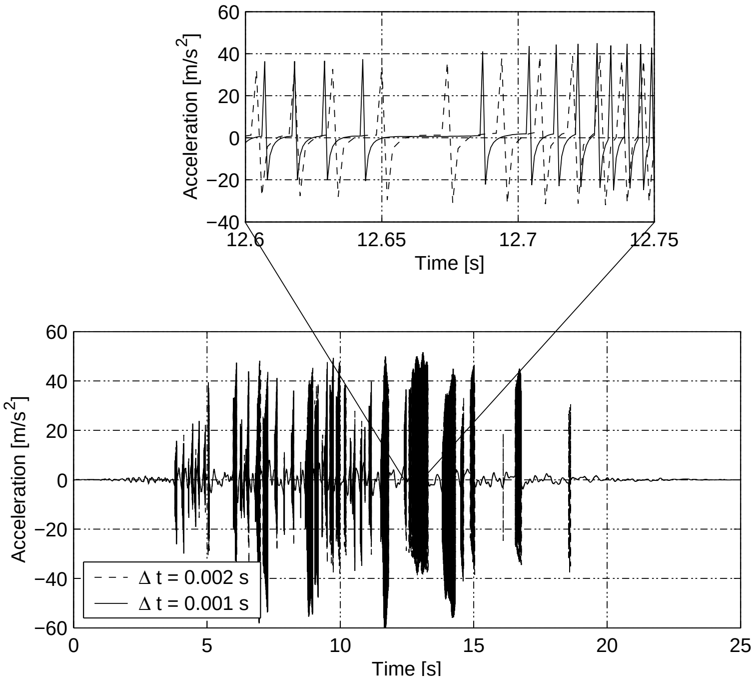


Figure 311.79: Acceleration time history at lowest free node

value is given for the entire time history:

$$\Delta d = \frac{d_{0.002}(t) - d_{0.001}(t)}{\max d_{0.001}} \quad and \quad \Delta a = \frac{a_{0.002}(t) - a_{0.001}(t)}{\max a_{0.001}} \tag{311.17}$$

In Figure 311.82 an integral measure for the difference in displacements and accelerations between the two analysis is given for all depths. The integral measures are defined as

$$diff_d = \frac{1}{\max |d|} \frac{1}{T} \sum_0^T |d_{0.002}(t) - d_{0.001}(t)| \; dt \tag{311.18}$$

$$diff_a = \frac{1}{\max |a|} \frac{1}{T} \sum_0^T |a_{0.002}(t) - a_{0.001}(t)| \; dt \tag{311.19}$$

$$\tag{311.20}$$

The integral differences in accelerations are quite large in the elements that are close to the base, that is where the motion is applied. Toward the surface the difference becomes smaller than 1%. This is a result of the fact that most of the plastic deformation occurs near the base which represents an undesired boundary effect. Again this result underlines the importance of an appropriate choice of the size of the computational domain.

With a point wise difference not exceeding 5% for accelerations and 2% for displacements the time step $\Delta t = 0.002s$ is sufficiently small to ensure stable and accurate results.

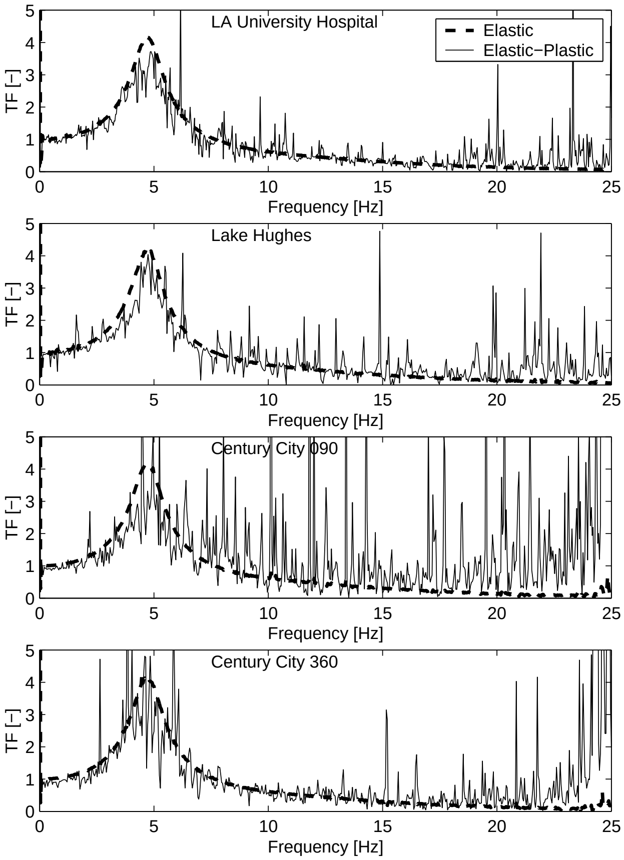


Figure 311.80: Transfer functions between acceleration at the soil surface and the base

311.7.1.5 2d Model

A 2d-model is proposed as a simplification of the full 3d-model. Representing a cross section of the full model it is expected to provide insight into its dynamic behavior while requiring considerably less computational resources. The 2d-model consists of one slice of eight-node brick elements as shown in Figure 311.83. The nodes of the two lateral faces are constrained to move together in x- and z-direction, the out-of-plane displacement in y-direction is fixed. The model approximates a plane strain situation.

The earthquake motion is applied to the model by the DRM method.

311.7.1.6 Input Motions

As input for the 2d model the motion from the Northridge earthquake recorded at LA University Hotel (Figure 311.75) is used. The acceleration time history is scaled to a peak ground acceleration of 1 g. Motion is applied in x-direction only, that is, this is a 1–D wave propagation.

Acceleration time histories at all the nodes of the boundary layer are obtained by vertically propagating

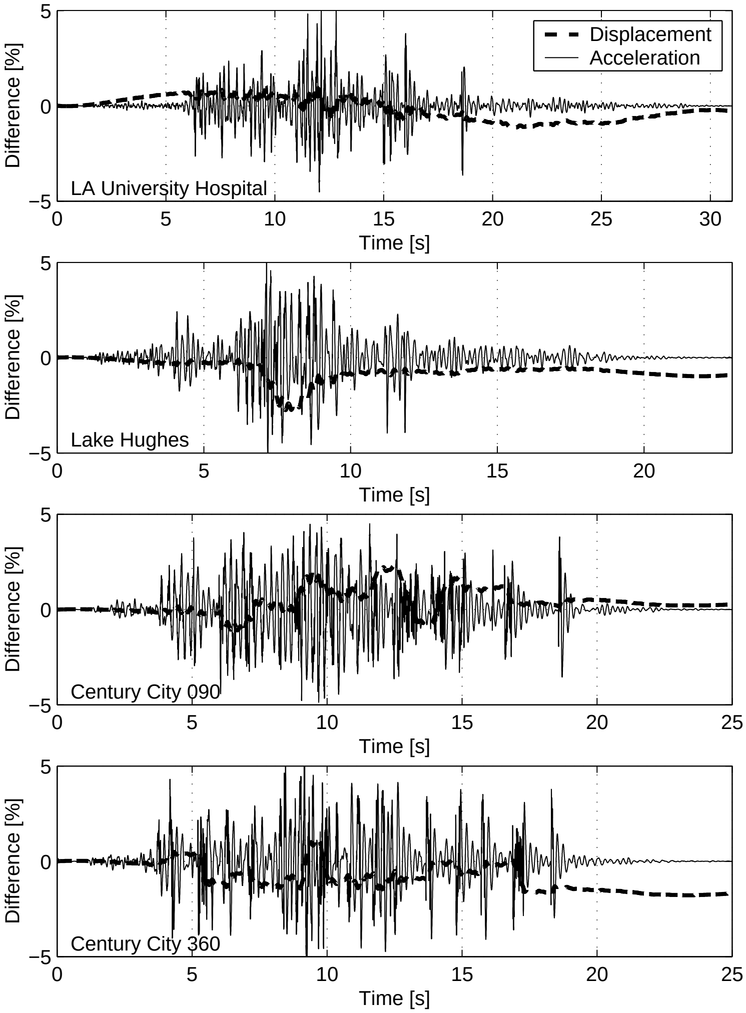
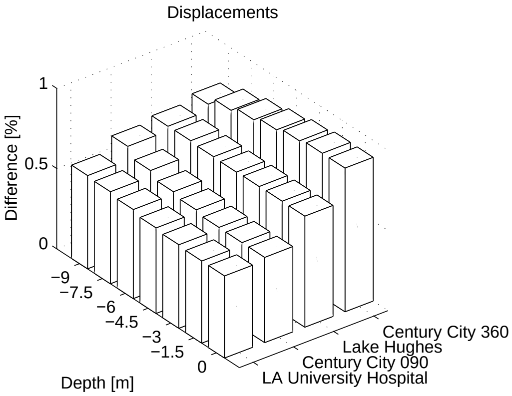


Figure 311.81: Difference between results of analysis with different time steps, in percent of the maximum value

a plane wave using the program SHAKE91 (Idriss and Sun, 1992). Because the free-field model has to match the properties of the free field as represented by the finite element model for the reduced domain, only linear elastic material without strain dependent reduction of shear modulus and a constant amount of hysteretic material damping is used in the SHAKE91-analysis. The earthquake motion obtained in this way corresponds to a shear wave propagating upward through a homogeneous linearly elastic half space.

The acceleration time histories from the SHAKE91-analysis are then integrated twice to obtain displacements. Before integration the acceleration and velocity time histories are transformed into Fourier space, multiplied with a high pass filter and transformed back into time domain. Then a simple parabolic baseline correction is performed in order to obtain zero initial, final and mean values.



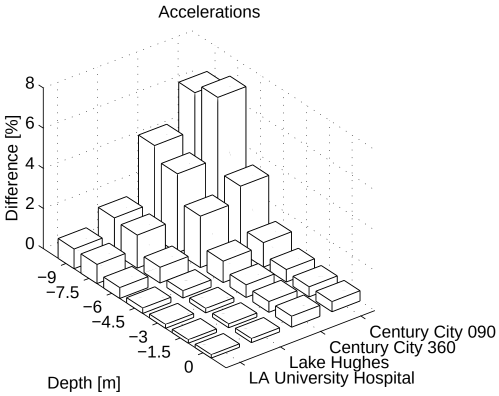


Figure 311.82: Averaged differences between results of analysis with different time steps

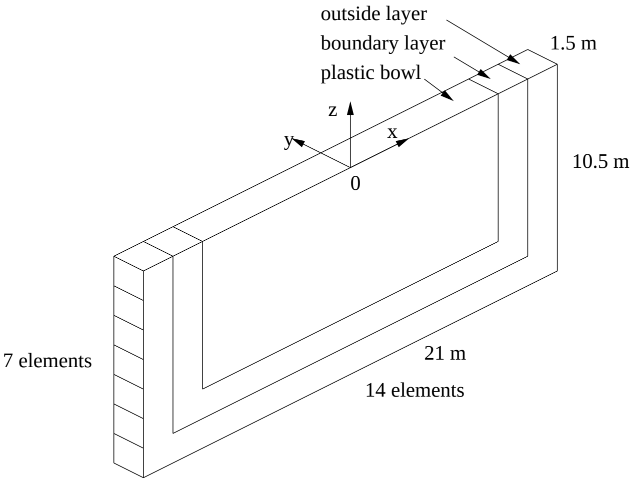


Figure 311.83: Two-dimensional quasi-plane-strain model

311.7.1.7 Boundary Conditions

Different boundary conditions are tested on the free-field model. First all outside boundary nodes are fully fixed as shown in Figure 311.84 a). Then they are released and attached to dash pots that are both perpendicular and tangential to the boundary (schematically shown in Figure 311.84 b)). The dash pots perpendicular to the boundary are specified to absorb p-waves, those tangential to the boundary to absorb s-waves. Because in this configuration no displacement constraint is imposed to the model on the faces at $x = \pm 10.5$ meter the horizontal at-rest soil pressure has to be applied to the corresponding nodes manually. This is done by recording the reaction forces in the model with fixed boundaries and applying them with opposite sign to the model with absorbing boundaries. The horizontal displacements after applying self weight should be very small.

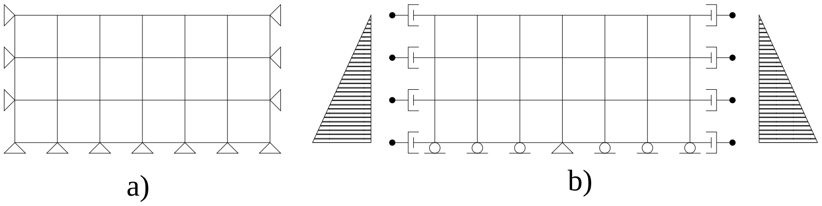


Figure 311.84: The boundary conditions of the 2d model

This configuration of boundary conditions has no fixed point in x-direction. Because the dash pots only provide resistance to high velocity motions the model is very sensitive to low frequency components of the motion. The slightest imbalance in acceleration causes the entire model to move as a rigid body in x-direction. To avoid this to happen the node at the center of the base (x = 0.0 m, z = -10.5 m) is fully fixed in the following analysis.

Figure 311.85 shows results from a free-field analysis on a homogeneous elastic model. Displacements on an exterior boundary node as well as transfer functions between a point at the surface and a point on the exterior boundary of the plastic bowl are presented for the two configurations of boundary conditions shown in Figure 311.84. It can be seen that the displacements outside the plastic bowl in the model using absorbing boundary conditions are much larger compared to the model with fixed boundaries. This result is as expected considering the immediate proximity of the boundary. It also gives an idea about the constraints the fixed boundary imposes on the motions. The transfer function in Figure 311.85 b) is defined as the ratio between the Fourier amplitude spectra of a point at the surface and a point on the exterior boundary:

$$TF(\omega) = \frac{A_1(\omega)}{A_2(\omega)} \tag{311.21}$$

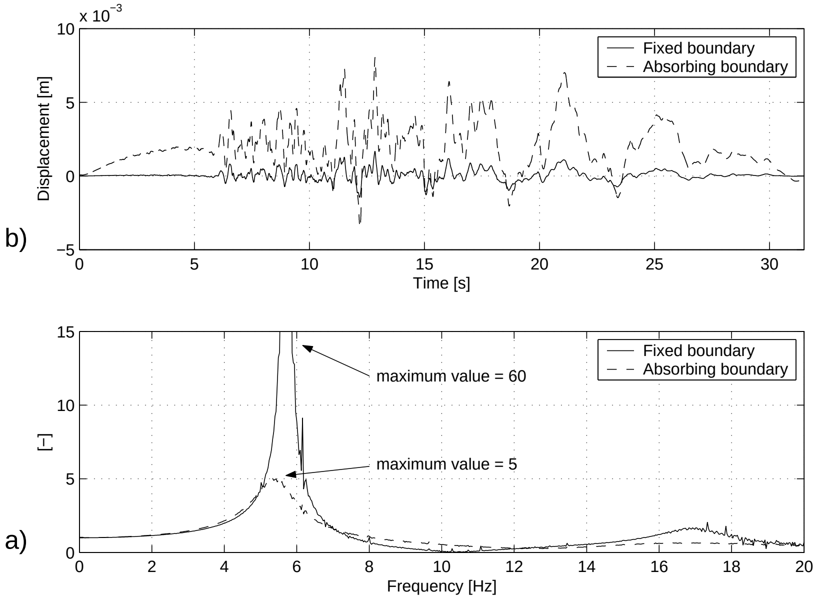


Figure 311.85: Elastic homogeneous free-field model: a) Displacements of an exterior boundary node (x,z) = (9.0,0.0) m, b) Acceleration transfer function between surface and depth $\frac{A(\omega)_1}{A(\omega)_2}$

where $A_1(\omega)$ is the Fourier amplitude spectrum of the acceleration time history at the point (x,z) = (0,0)m and $A_2(\omega)$ the corresponding spectrum at the point (x,z) = (9.0,-7.5)m. The figure shows that the large peak representing the first natural frequency of the system, corresponding to a standing shear wave in a soil layer of 10.5 meter depth, gets reduced considerably by the absorbing boundary. An energy build-up in the model due to reflection of waves on the model boundaries can be reduced effectively with the configuration of boundary conditions shown in Figure 311.84 b). By releasing the fixed node at (x,z) = (0,-10.5)m the resonance peak could be reduced by another 10% approximately, however at the cost of remaining permanent displacements at the end of the analysis.

Alternatively to imposing a rigid constraint to a single node at the base the model can be prevented to move horizontally as a rigid body through uniaxial springs. This gives the possibility to adjust the frequency of the eigenmode that corresponds to a vertically propagating plane shear wave. By appropriately choosing the spring constants the model can therefore be adjusted in such a way that it represents the natural frequency of a soil deposit on bedrock.

311.7.1.8 Structure

Four very simple structures are chosen to illustrate the effects of dynamic SFSI. A beam-column element of length L and moment of inertia I_y is fixed to a footing. A lumped mass of $M = 100,000$ kg is added to

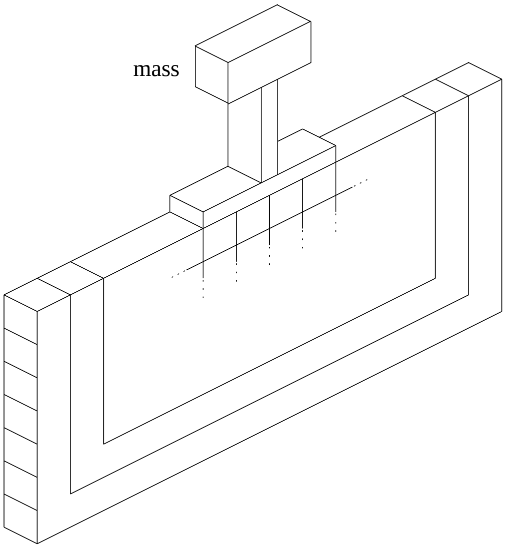


Figure 311.86: The 2d SFSI-model

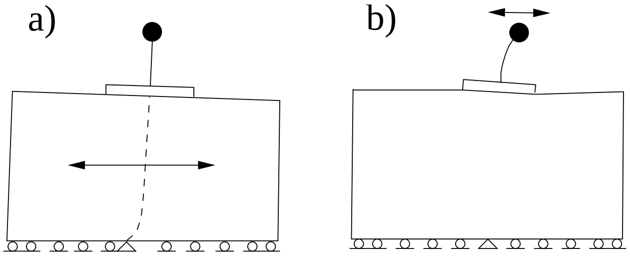


Figure 311.87: a) First eigenmode, b) second eigenmode of SFSI-system

the translational degrees of freedom of the top of the structure. The footing is 0.5 m deep, spans over four soil elements and is rigidly connected to the adjacent soil nodes. Its Young's modulus is chosen large enough so that the footing can be considered rigid. The mass density of the footing is $\rho = 2400\,\text{kg/m}^3$, the column is considered massless. The moment connection between the nodes of the footing, having 3 (translational) degrees of freedom, and the 6 degrees of freedom of the nodes of the column is assured by a very stiff beam element that is connected to a node at the bottom and a node at the top of the footing. The column is then simply connected to the upper node of this auxiliary beam element.

The parameters of the four columns are chosen such that the second natural frequency, that is the natural frequency attributed to bending of the column (Figure 311.87 b)), is evenly distributed over the frequency range of the input motion (Figure 311.88). Structure 4 is designed such that it's second natural frequency matches the largest spike in the input motion. Table 311.2 lists the properties of the structures used in the analysis. For the nonlinear columns a strain hardening material is chosen

Structure	Length [m]	Stiffness $E\,I_y$ $[MN\,m^2]$	Mass [kg]	Yield Moment [kNm]
1	5.5	1680	100,000	800
2	3.5	5670	100,000	1,800
3	2.5	13440	100,000	320
4	5.0	5670	100,000	1,800

Table 311.2: Properties of the analyzed structures

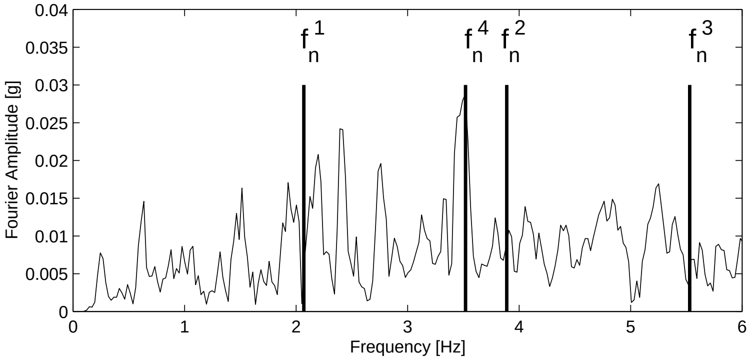


Figure 311.88: Fourier amplitude spectrum of input motion with second natural frequencies of the 4 SFSI-systems

that consists of an initial elastic branch with tangent modulus E and a post-yield branch with tangent modulus $0.2\,E$. The Young's modulus for all four structures is $E = 210$ GPa. The yield stress f_y for structures 1, 2 and 4 is 20 MPa and for structure 3 it is 2 MPa.

311.7.1.9 Structure with Fixed Base

To begin with a parametric study of a series of structures with varying stiffnesses is analyzed. The stiffness is varied by changing the width of the column section. The different structures are expected to respond specifically to the frequency range of the input motion that is in the neighborhood of the natural frequency of the column. The input motion that is applied at the base of the structure has been recorded in a previous free-field analysis of the 2d-model.

The results of this parametric study are shown in Figures 311.89 and 311.90 for linear and nonlinear structures, respectively. The Fourier amplitudes spectra's of the acceleration at the top of the structure are plotted for 15 structures with variable natural frequency f_n. A line of equal frequency is also provided. The input motion is plotted in the background of the figure. It can be seen that the maxima of the

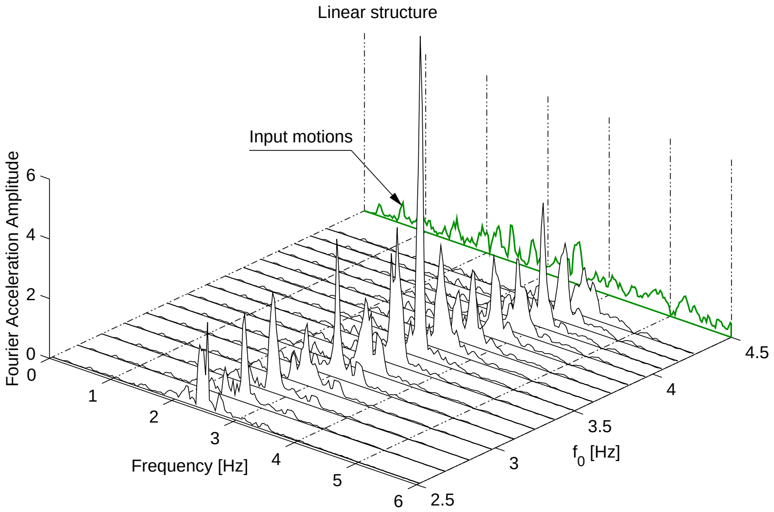


Figure 311.89: Parametric study of 15 *linear* structures with varying natural frequency.

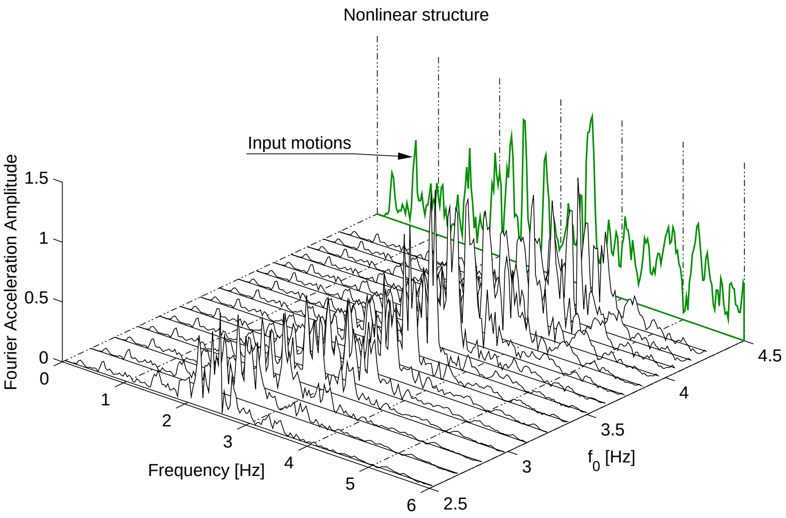


Figure 311.90: Parametric study of 15 *nonlinear* structures with varying natural frequency

frequency spectra's are almost perfectly aligned along the line of equal frequency. This is even more obvious in the case of a linear structure. In that case the responses of the structures are very narrow banded. As the structure remains elastic the top of the structure oscillates mainly in its initial natural frequency. Lower and higher frequencies are eliminated to a great extent.

In the case of the inelastic structure there is clearly more damping and reduction of the responses at some (most) frequencies. The nonlinearity in structure is producing a longer effective period for the structure, and that effective period changes during shaking. This in turn widens the frequency range of structural response. That is, the response is lower, but the frequency characteristic is (much) wider.

A series of fixed-base analysis is also performed on the four structures mentioned in Section 311.7.1.8. The first natural frequencies of the four structures with its base fixed, corresponding to the second mode of vibration of the SFSI-model, are given in Table 311.3. It can be seen that the influence of the soil on the natural frequency of the SFSI-system increases as the overall stiffness of the structure increases.

Structure	1^{st} natural frequency of fixed-base system [Hz]	2^{nd} natural frequency of SFSI-system [Hz]
1	2.71	2.07
2	9.82	3.89
3	25.1	5.53
4	5.75	3.52

Table 311.3: Eigenfrequencies of the analyzed models

311.7.1.10 Results

The results of the SFSI- as well as the fixed-base analysis are presented in the following. The displacements at the top of the nonlinear structures are recorded and plotted in Figure 311.91. It can be seen that the results from the SFSI- and the fixed-base-model differ considerably in terms of maximum as well as permanent displacement. In contrast to this the displacements at the base of the column are almost identical for the two models (results not plotted). Figure 311.92 displays the displacements at the top of structures 1 and 2 for all the combinations of linear and nonlinear soil and structures that have been analyzed. Due to the low yield moment the permanent displacement for structure 1 is relatively large in the analysis involving nonlinear columns. The results involving nonlinear columns on linear and on nonlinear soil are very similar in their overall shape, however permanent deformations are very different. It seems that the forces that trigger plastic deformations in the column strongly depend on the behavior of the soil beneath the foundation.

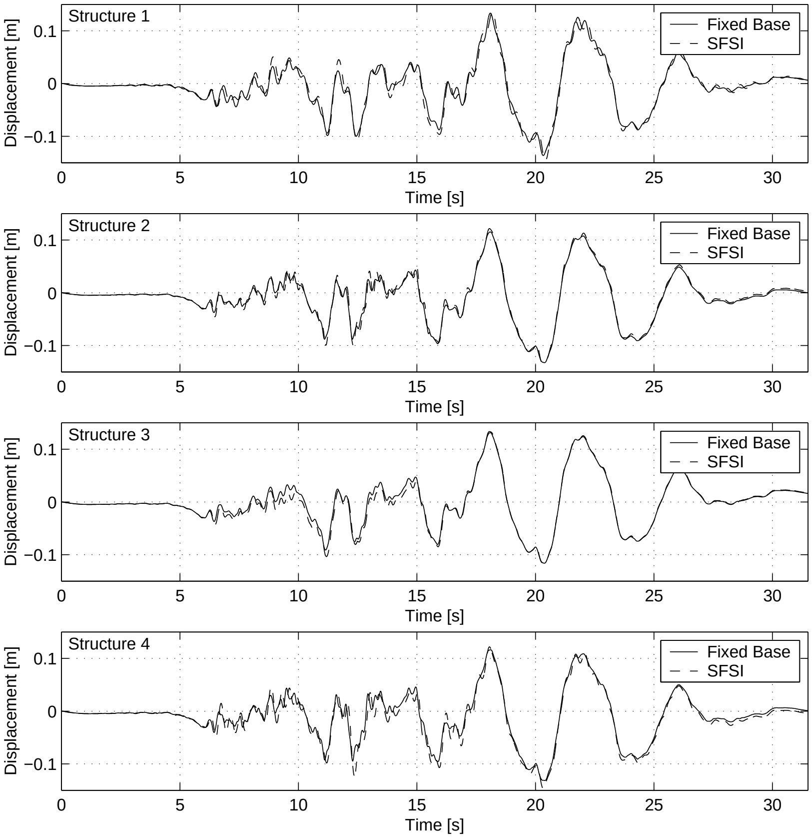


Figure 311.91: Displacements in x-direction at the top of the *nonlinear* structures

In order to investigate the forces causing plastic deformations in the structures we look at the base moments between foundation and column. In Figure 311.93 the moments at the base of the linear structures are plotted.

For structures 1 and 4 the moments for the fixed-base model are higher than for the SFSI-model. This means that in this case neglecting the effects of SFSI leads to a conservative design. Structures 2 and 3 however have to resist higher moments when SFSI is taken into account. Because the SFSI-system is more flexible than the fixed-base structure its modes of vibration are excited by a different range of frequencies contained in the input motion. For a particular motion this can lead to resonance of the SFSI-system. This result is in contradiction with current engineering practice suggesting that neglecting SFSI in general leads to a more conservative design.

Figure 311.94 shows the moments at the base of structures 1 to 4, this time for the analysis involving nonlinear column elements. The evolution of the second natural frequency of the SFSI-system is also provided as a qualitative indication for when plastic deformations occur. The base moments for structures 1 and 3 in the fixed-base- and the SFSI-analysis are very similar. Due to the low yield moment of the

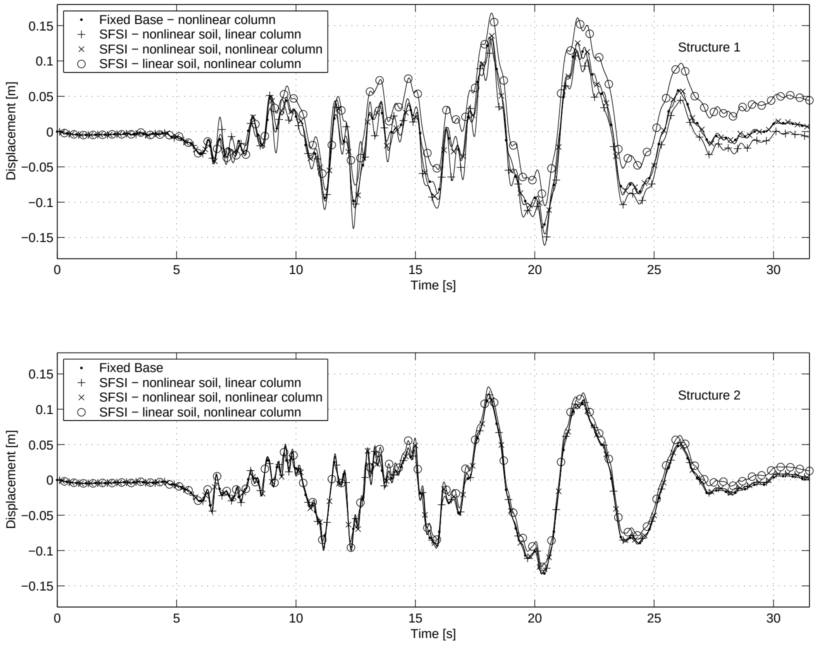


Figure 311.92: Displacements in x-direction at the top of structures 1 and 2

structure no resonance with the input motion occurs as a lot of energy is dissipated through plastic deformation.

Figure 311.95 shows an interesting aspect of nonlinear SFSI. In the analysis involving elastic-plastic soil the Fourier amplitudes of the moment at the base of the structure are reduced in the neighborhood of the natural frequency of the system. This is most likely due to dissipation of energy caused by elastic-plastic deformations in the soil that, in their turn, are a result of large loads provoked by resonance between the SFSI-system and the input motion.

As a measure of the plastic strain occurring beneath the footing the equivalent plastic strains averaged over all the Gauss points are calculated. The results are given at $t = 12$ s and at $t = 14$ s, that is shortly before and after the largest plastic deformation occurs (Figures 311.96 and 311.97).

Plastic strains are larger in the analysis involving an elastic structure. This reflects the fact that elastic structures don't dissipate any energy by themselves. For structure 2 no significant difference can be observed because of its high yield moment. Structure 4 is characterized by the same yield moment, its slightly smaller natural frequency however causes resonance with the input motion which leads to larger plastic strains beneath the footing. The largest plastic strains develop in the layer of elements adjacent to the boundary layer. This can be due to an input motion that isn't fully compatible with the elastic properties of the DRM-model. It should be possible to reduce these undesired plastic strains by either increasing the size of the soil model or by selecting a method to obtain the free-field motions that

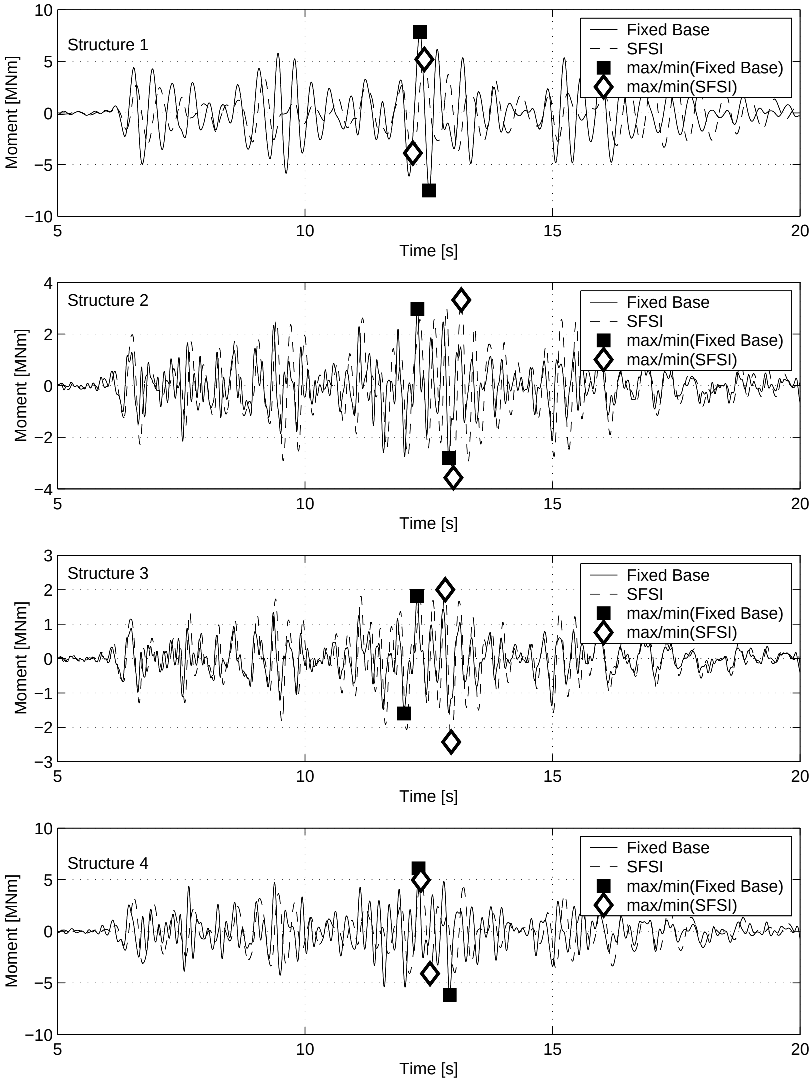


Figure 311.93: Moments at the base of the *linear* column

represents the soil properties of the DRM-model more closely.

311.7.2 Full nonlinear 3d Model

The 2d SFSI-model presented in the previous section is extended to a 3d model in the following. The goal is to show that the considerations for accuracy and stability of the numerical method obtained from the 1d-model remain valid for the 3d-model. Even if the simplicity of the analyzed problem doesn't necessarily justify the additional computational effort it is important to show that it is possible to obtain reliable results for a problem that involves the following elements:

- 3d model with about 700 elements, 960 nodes and 2700 equations

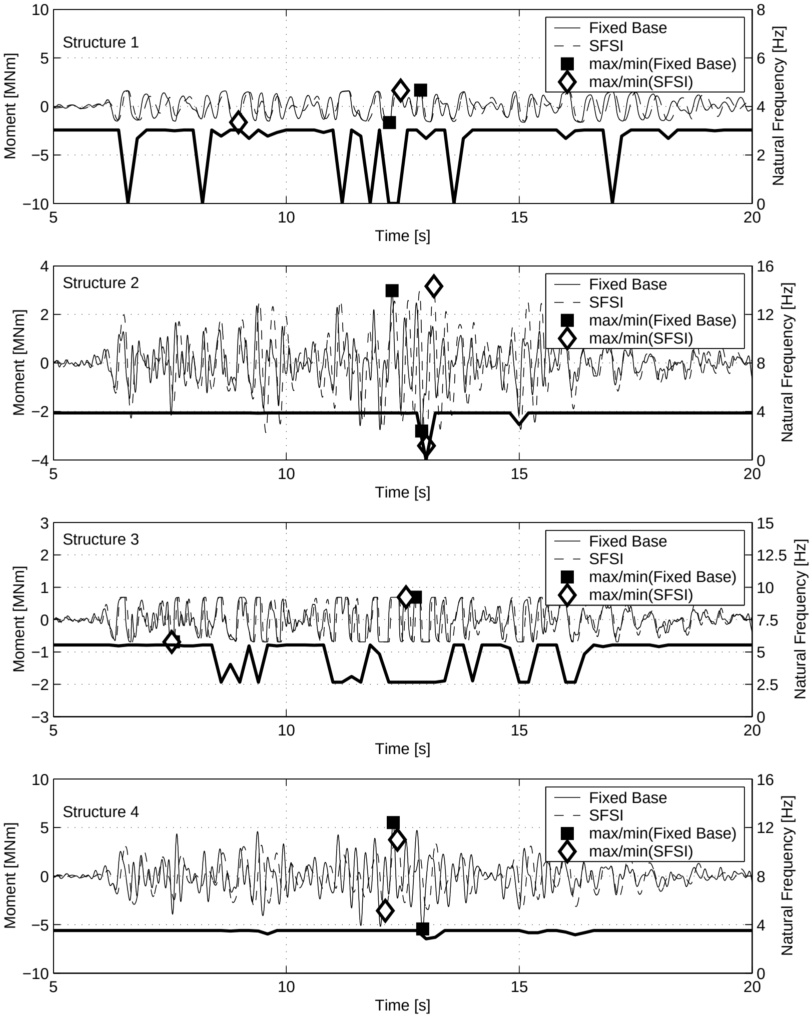


Figure 311.94: Moments at the base of the *nonlinear* column

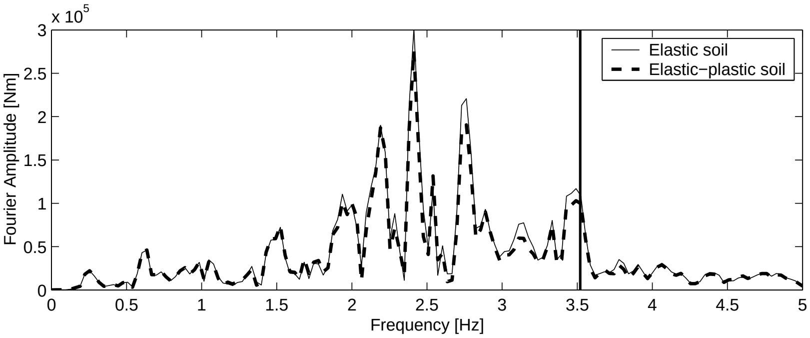


Figure 311.95: Fourier amplitude spectra of moments at the base of *nonlinear* column, SFSI-analysis

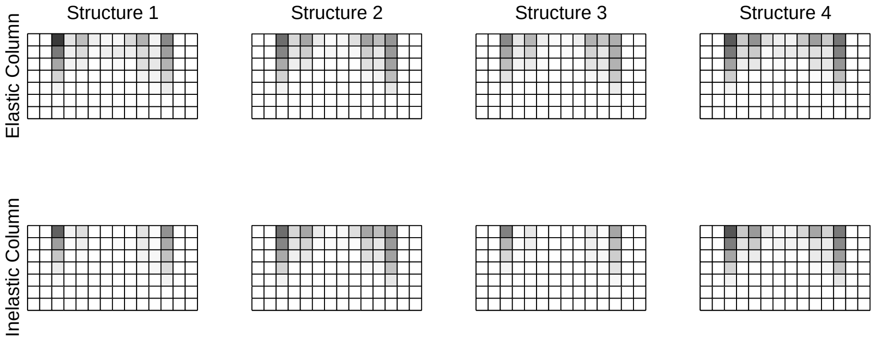


Figure 311.96: Average equivalent plastic strain at time t = 12 s

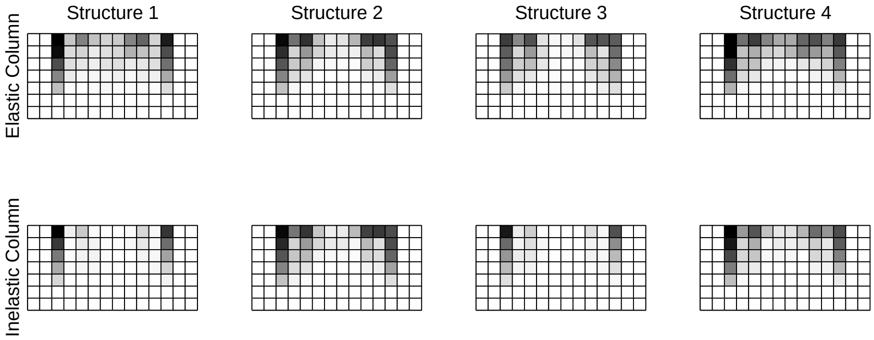


Figure 311.97: Average equivalent plastic strain at time t = 14 s

- Elastic-plastic soil (Drucker-Prager with kinematic hardening)
- Nonlinear structure (bilinear material model)
- Ground motion applied through the Domain Reduction Method (DRM)
- Absorbing boundary of Lysmer type

311.7.2.1 Description of Model

The 3d model is based on the 2d model shown in Figure 311.86. In y-direction 6 more slices of 7 x 14 elements are added (Figure 311.98). The x-z plane at y = 0 represents a plane of symmetry. Lysmer boundaries are attached to all outside boundaries with the exception of the plane of symmetry and the soil surface. The main difference to the 2d model is that 3d wave propagation is possible which leads to higher radiation damping.

The structure was chosen to have the same geometric and material properties as Structure 4 in the previous section.

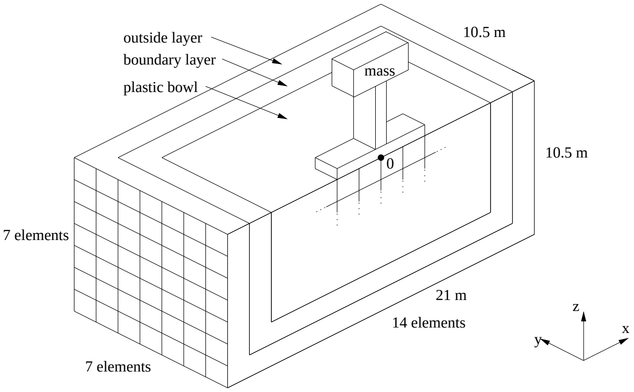


Figure 311.98: The full 3d-model

311.7.2.2 Results

Some results of the 3d-analysis together with the corresponding data of the 2d analysis are presented in Figure 311.99. Due to limited memory only the first 20 seconds of the time history were processed. A more efficient implementation of the application of effective forces for the DRM-method inside the finite element code should solve this problem. The analysis took 66 hours to finish.

The displacements obtained at the top of the structure as well as the moments at its base are very close to the results of the 2d-analysis. This shows that the analysis provides reliable results for a full 3d nonlinear SFSI problem. The amplitude of the base moment is at several instances larger for the 3d-model than for the 2d-model. This can be explained with the fact that more energy is present in the 3d-model whereas the energy the structure can absorb is the same as in the 2d-model. Also it is obvious that the natural frequencies of the 3d-model are not exactly the same as for the 2d-model and therefore changes the dynamic behavior in a way that is almost impossible to predict beforehand.

Because of the simple geometry of the problem the 2d-model is absolutely sufficient for analyzing the forces acting on the structure. If one is interested in the stress history in the soil surrounding the footing then the 3d-model can provide valuable additional information.

. . .

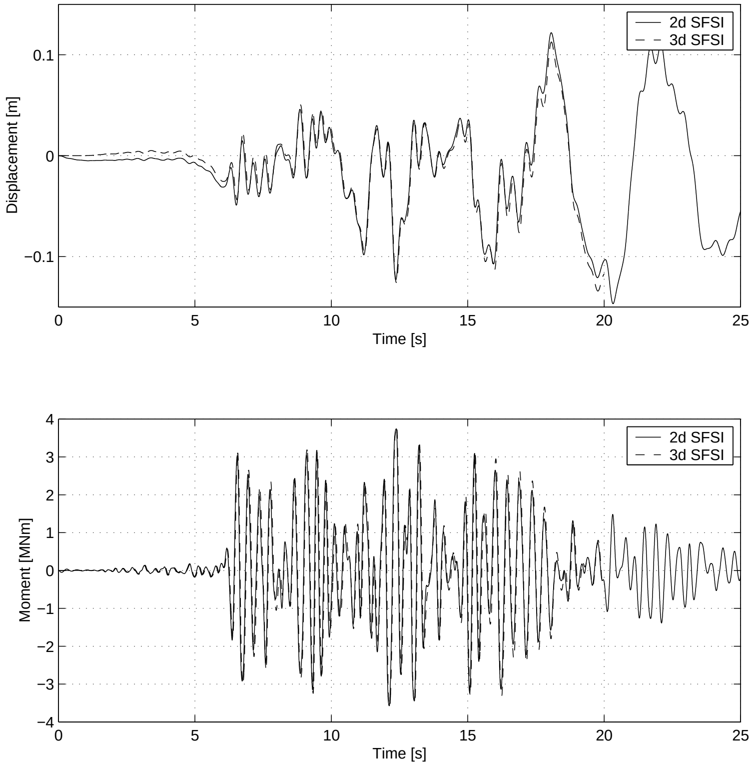


Figure 311.99: Top: Displacements at the top of Structure 4, Bottom: Moments at the base of Structure 4

311.8 Lotung Large Scale Seismic Test (LLSST) Earthquake 07

311.8.1 Introduction

Figure 311.100 shows the G/Gmax and Damping data used for Pisano model.

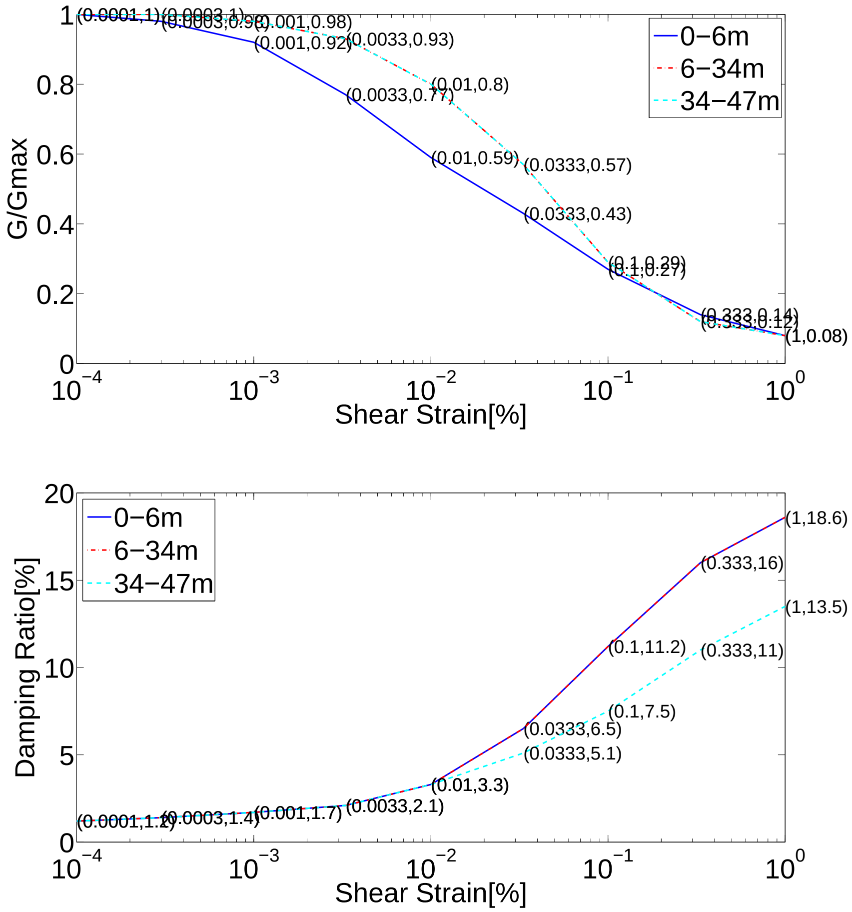


Figure 311.100: G/Gmax and Damping Curves

311.8.2 Input motion and input method

We use the data from Lotung Large Scale Seismic Test (LLSST) which is operated by TaiPower and IES during the period from 1985 to 1990. We choose event 07 to verify our modeling. Seismic motion data is available for download HERE.

311.8.3 Results

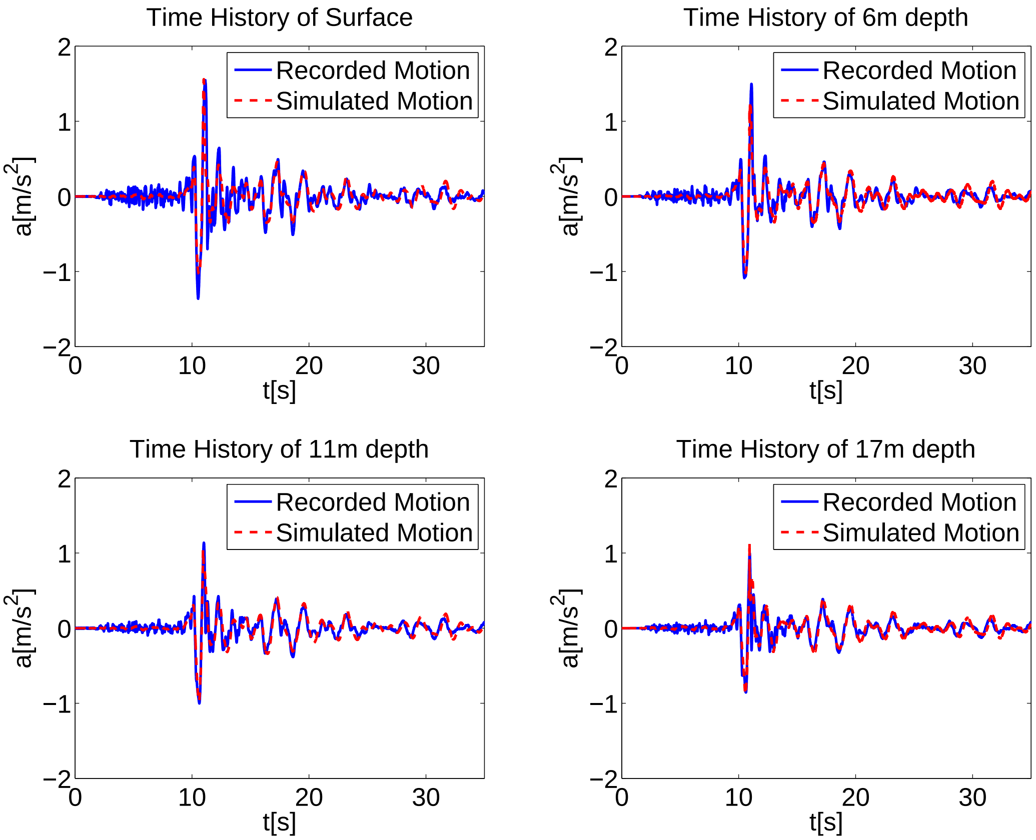


Figure 311.101: Time history comparison at different depths

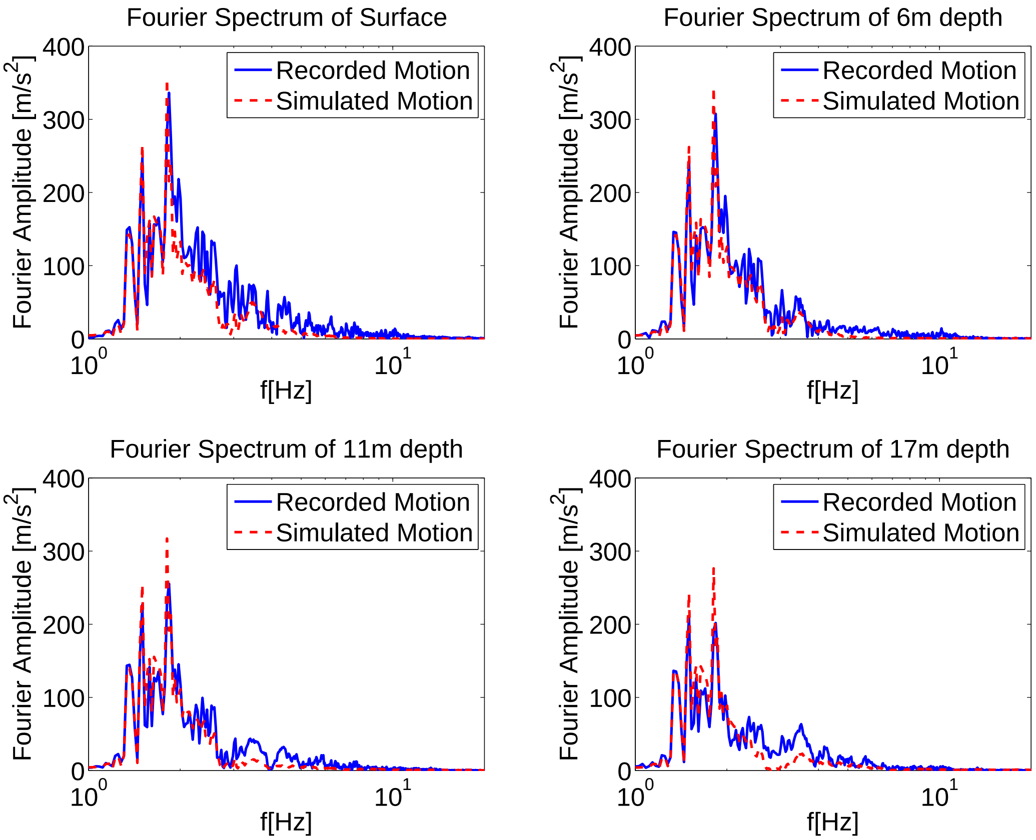


Figure 311.102: Fourier spectrum comparison at different depths

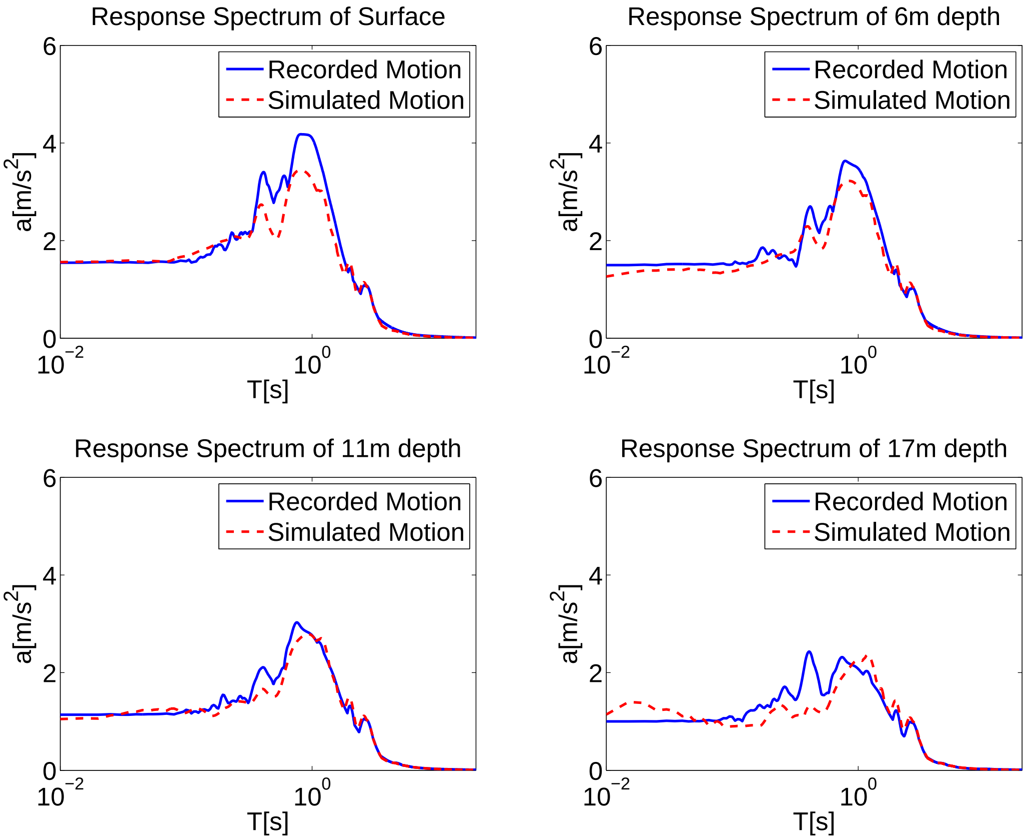


Figure 311.103: Response spectrum comparison at different depths (damping ratio 5%)

Chapter 312

Verification and Validation for Static and Dynamic Behavior of Soil-Structure-Interaction

(2012-2017-2018-2019-2021-)

(In collaboration with Dr. Nima Tafazzoli, Dr. Yuan Feng, Prof. Han Yang, and Dr. Hexiang Wang)

312.1 Chapter Summary and Highlights

312.2 Solid-Beam Model-Comparison of Real-ESSI eigen frequencies with ANSYS and Sofistik

Figure 312.1 and Table 312.1

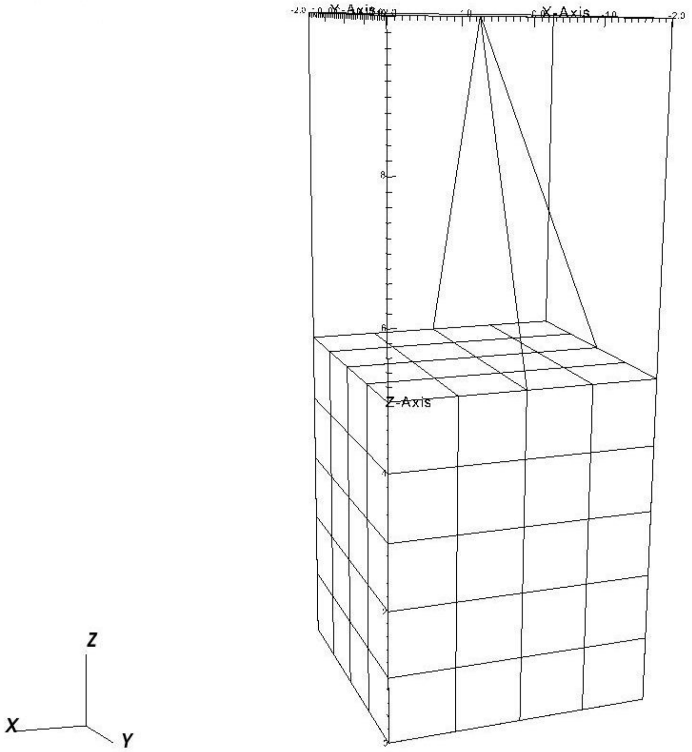


Figure 312.1: Finite element model

Table 312.1: Comparison of eigen frequencies obtained from Real-ESSI, ANSYS, and Sofistik

Mode	Real-ESSI (Hz)	ANSYS (Hz)	Sofistik (Hz)
1	5.4887	5.3868	5.439
2	7.1729	7.0711	7.311
3	12.6907	12.4670	12.751
4	13.197	13.1137	15.688

312.3 Solid-Beam Model-Comparison of model responses using elastic beams with 12dofs and 9dofs

312.3.1 FEM Model

Figure 312.2

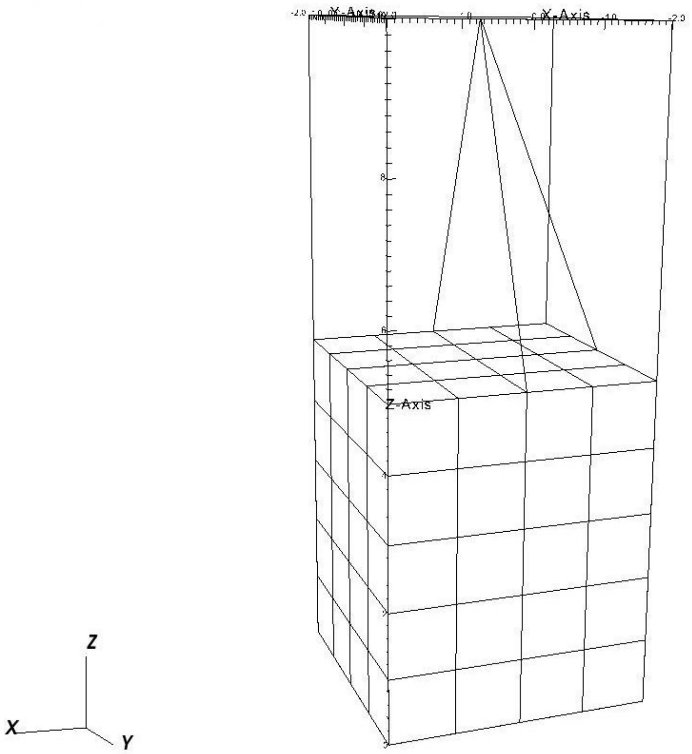


Figure 312.2: Finite element model

312.3.2 Static Analysis

Figure 312.3

312.3.3 Dynamic Analysis-Applying Force

Figures 312.4 and 312.5

312.3.4 Dynamic Analysis-Applying Displacement

Figures 312.6 and 312.7 and 312.8

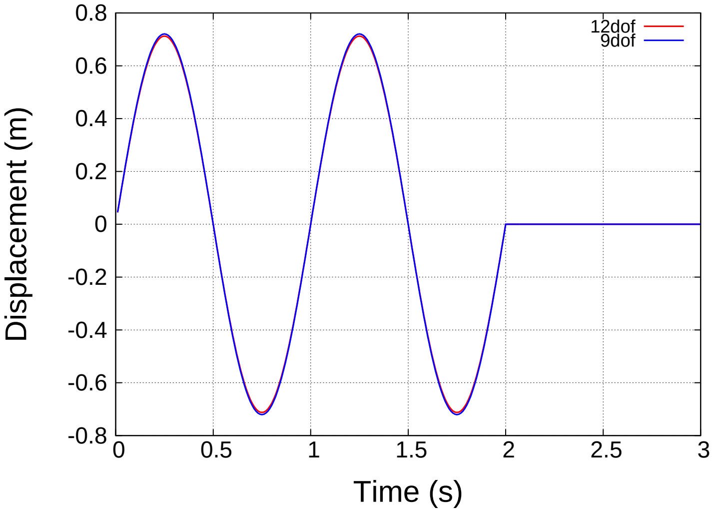


Figure 312.3: Comparison of static displacements (pseudo time) of the top nodes, Force time history applied to the top node

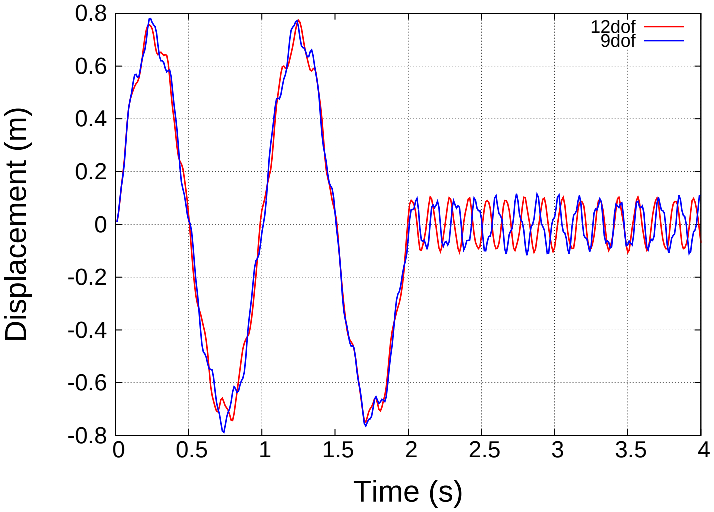


Figure 312.4: Comparison of displacement time histories of top node, Force time history applied to the top node, mass comes from the lumped mass added to the top node as well as the beams density

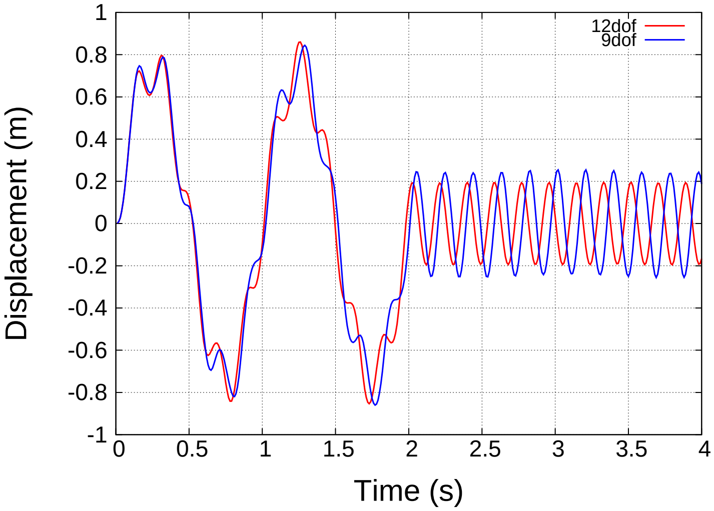


Figure 312.5: Comparison of displacement time histories of top node, Force time history applied to the top node, mass comes only from the beams density

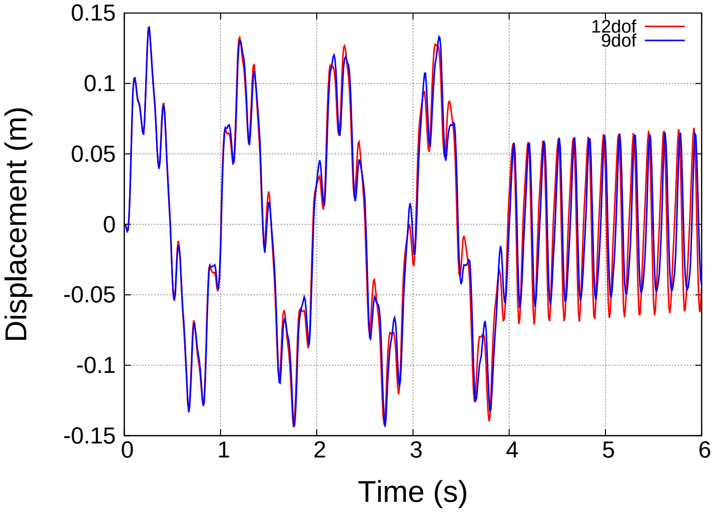


Figure 312.6: Comparison of displacement time histories of top node, Displacement time history applied at the bottom, mass comes only from the lumped mass added to the top

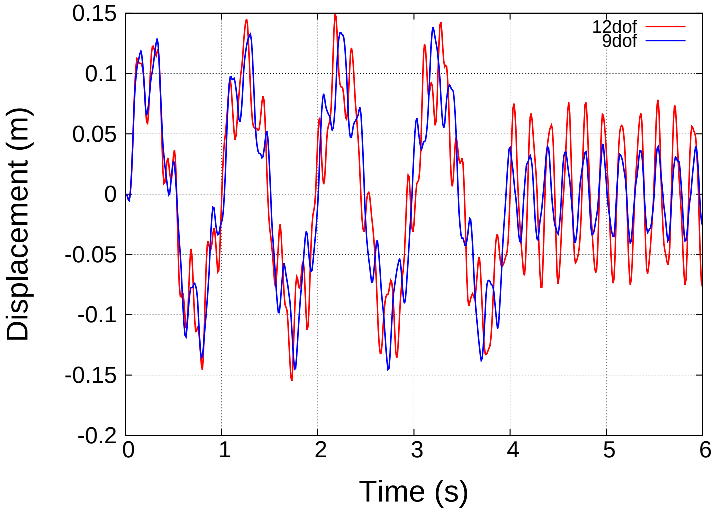


Figure 312.7: Comparison of displacement time histories of top node, Displacement time history applied at the bottom, mass comes from the lumped mass added to the top node as well as the beams density

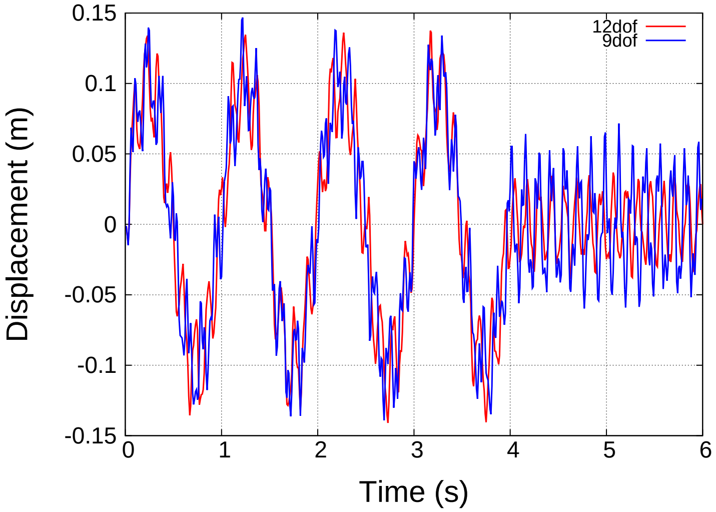


Figure 312.8: Comparison of displacement time histories of top node, Displacement time history applied at the bottom, mass comes only from the beams density

312.3.5 Comparison of eigen frequencies between models of using 9dof beam and 12dof beam

Table 312.2

Table 312.2: Comparison of eigen frequencies

Mode	9dof	12 dof
1	5.253	5.412
2	6.867	7.112
3	9.399	12.570
4	11.60	13.183
5	11.76	13.479
6	12.95	14.605
7	15.97	15.998
8	18.85	19.101
9	25.27	21.378

312.3.6 Eigen modes of model using 12dof beam

Figures 312.9 and 312.10 and 312.11 and 312.12 and 312.13

312.3.7 Eigen Modes of model using 9dof beam

Figures 312.14 and 312.15 and 312.16 and 312.17 and 312.18

312.4 Validation Using UNR Soil Box Test Setup

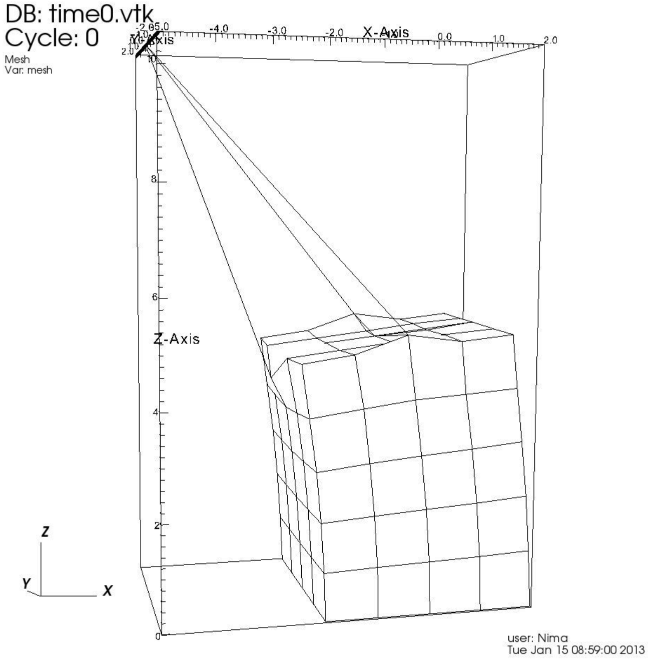


Figure 312.9: Mode 1

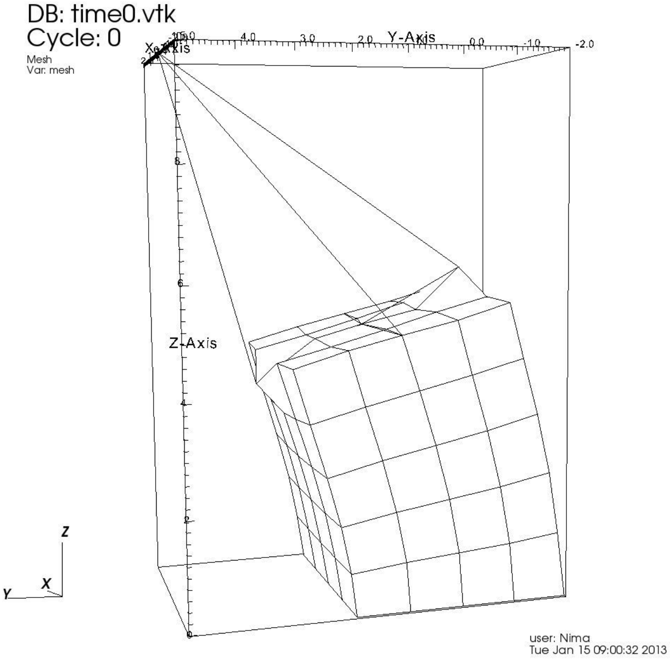


Figure 312.10: Mode 2

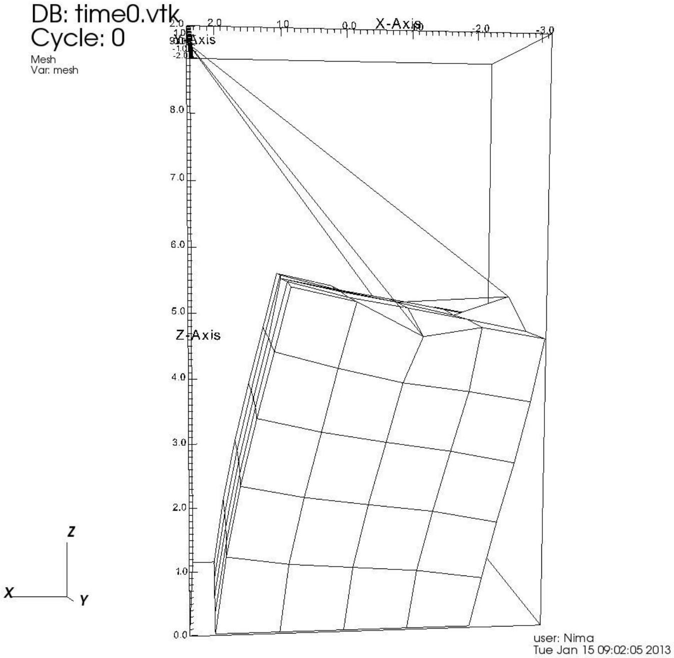


Figure 312.11: Mode 3

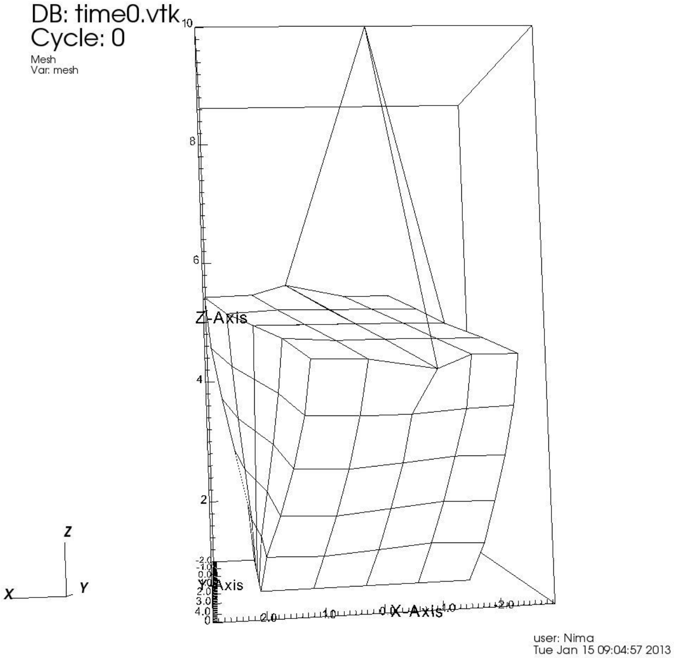


Figure 312.12: Mode 4

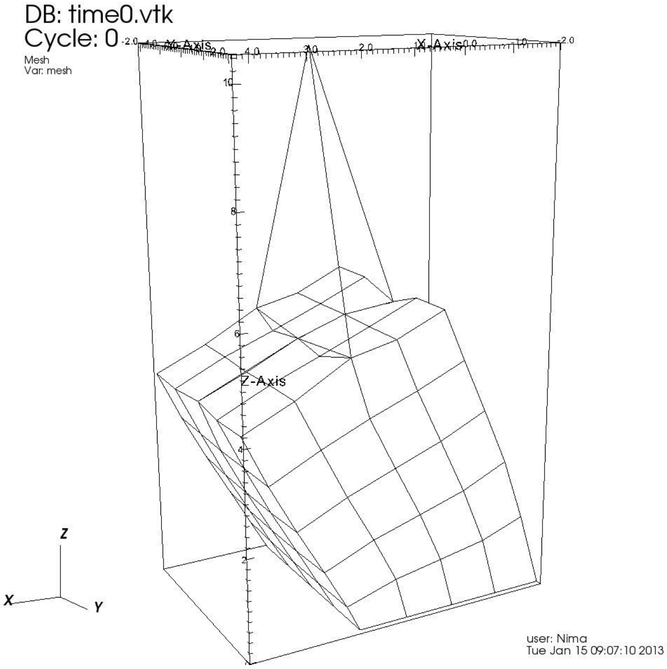


Figure 312.13: Mode 5

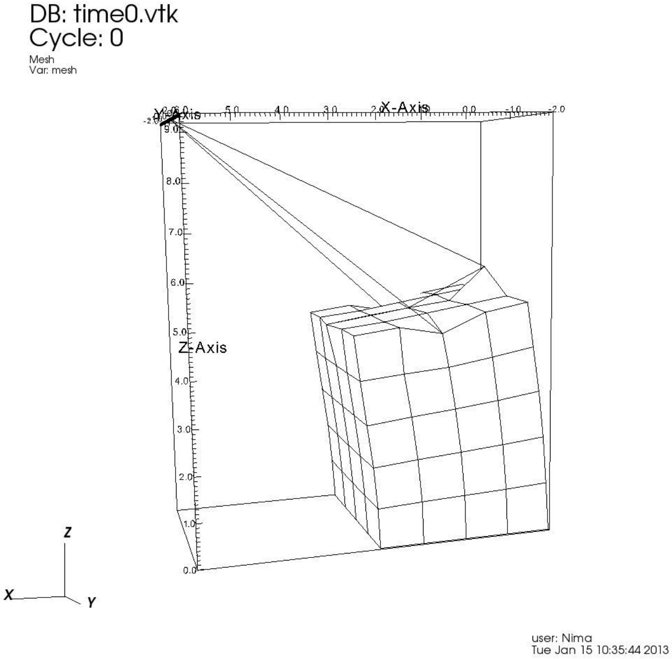


Figure 312.14: Mode 1

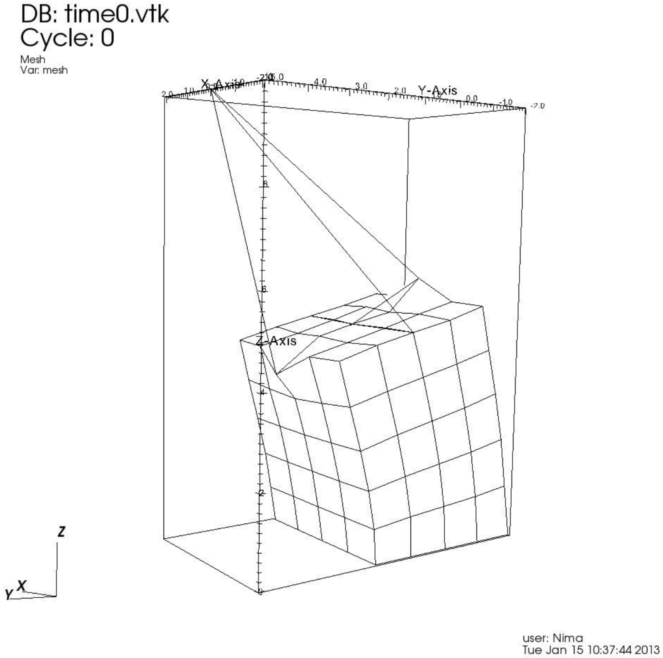


Figure 312.15: Mode 2

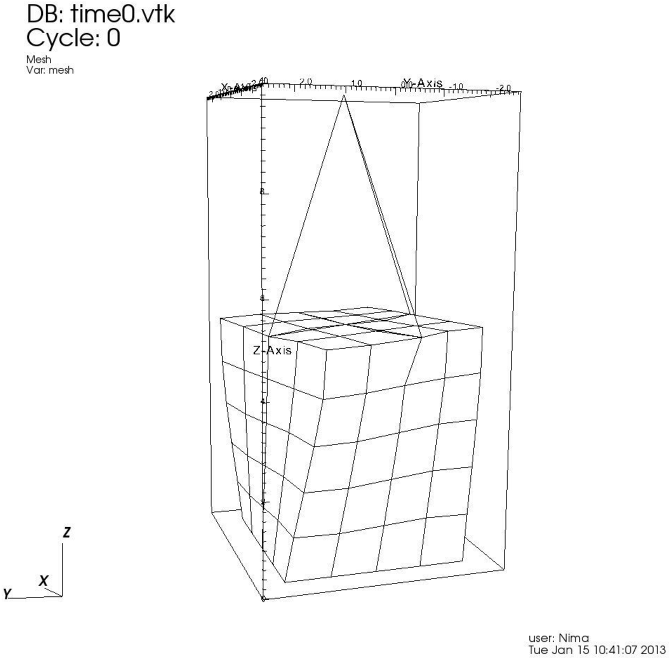


Figure 312.16: Mode 3

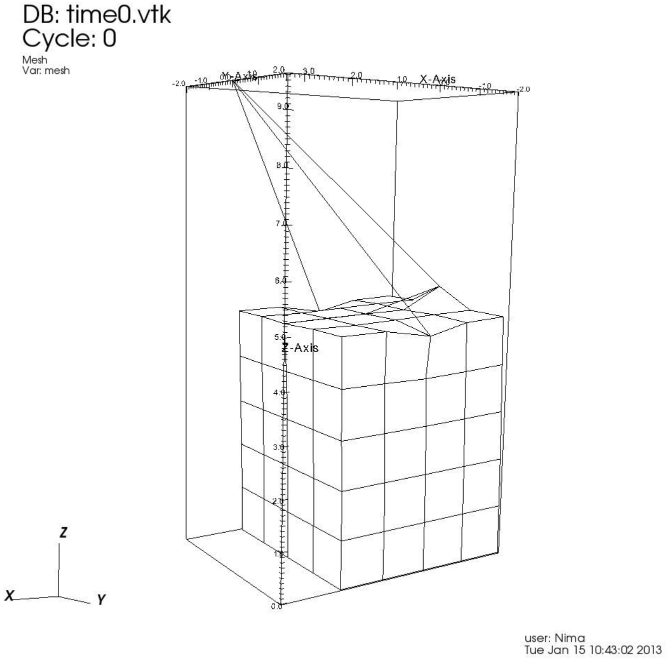


Figure 312.17: Mode 4

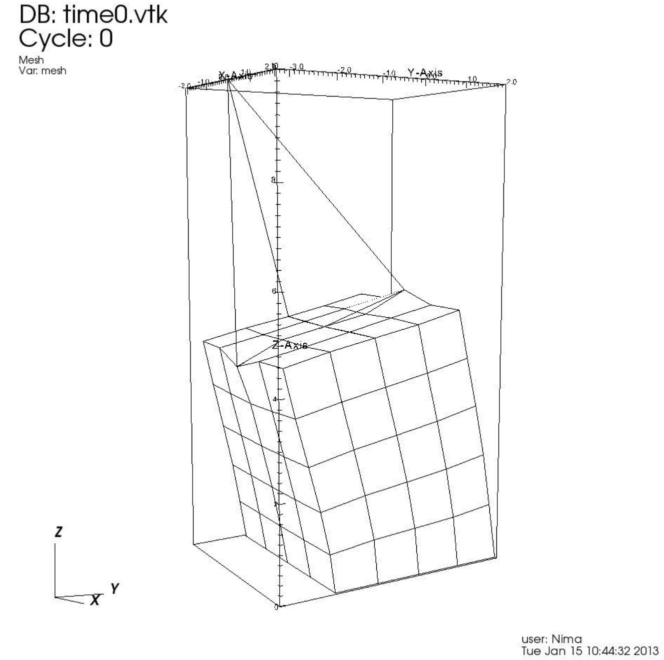


Figure 312.18: Mode 5

Chapter 313

Verification and Validation for Dynamic Solid-Fluid Interaction

(2017-2018-2019-2020-2021-)

(In collaboration with Dr. Hexiang Wang)

313.1 Chapter Summary and Highlights

313.2 V&V Examples

313.2.1 Box sloshing

A numerical example of earthquake-driven box sloshing is provided here. The initial configuration is shown in figure 313.1. The length of the box is 30 meters and the height is 10 meters. The thickness is 3 meters filled with 8-meter-deep water. The box is shaken by a uniform 1D horizontal excitation plotted in figure 313.2. The box is modeled with elastic material with $E = 12GPa, \mu = 0.2$.

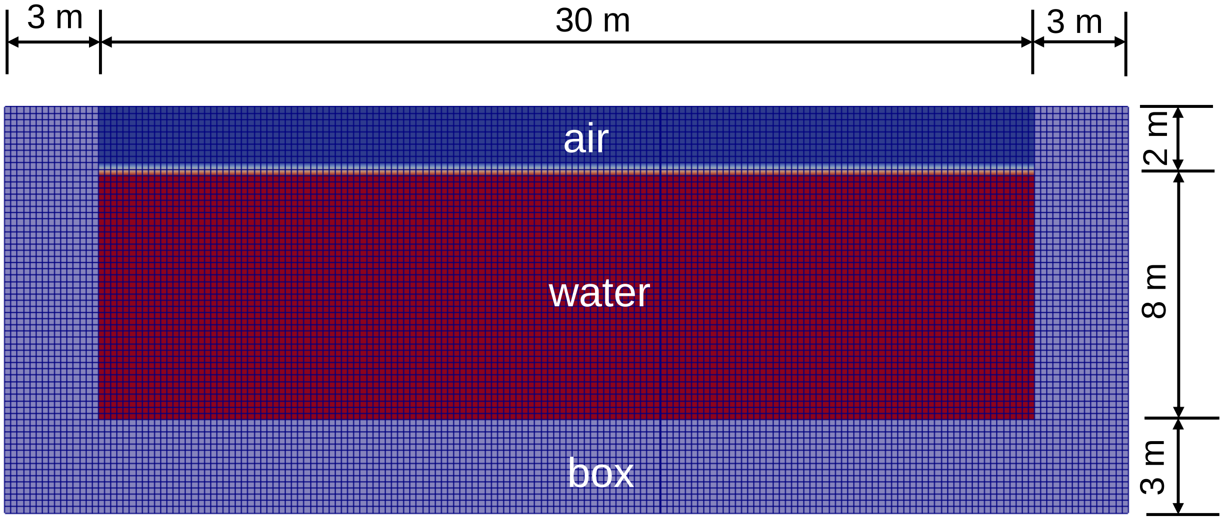


Figure 313.1: Model configuration of 2D sloshing box

The simulation result can be seen in figure 313.3. Clearly, elevations of free water surface can be observed under the excitation.

Another sloshing box example is driven by falling water flow. Figure 313.4(a) shows the result where solid domain and fluid domain has totally compatible mesh size (mesh size ratio 1:1). As mentioned before, VOF method has high requirement for the mesh size. In order to get accurate enough result with limited computation resources, refined mesh in the fluid domain is usually required for soil-structure-fluid interaction analysis. Analysis of model with discontinuous mesh is supported here through generalized interpolation scheme. Figure 313.4(b) shows the result of refined finite volume mesh in fluid domain (mesh size ratio 1:3).

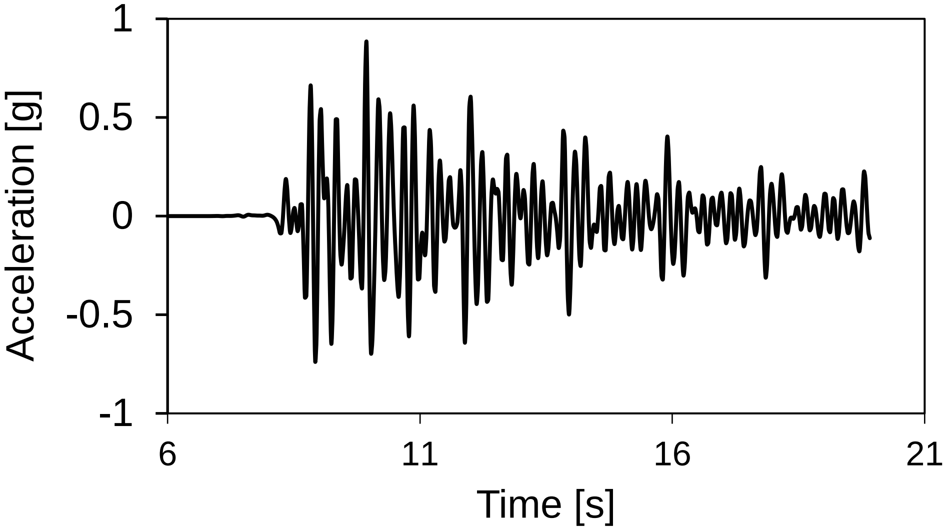


Figure 313.2: Time history of 1D excitation

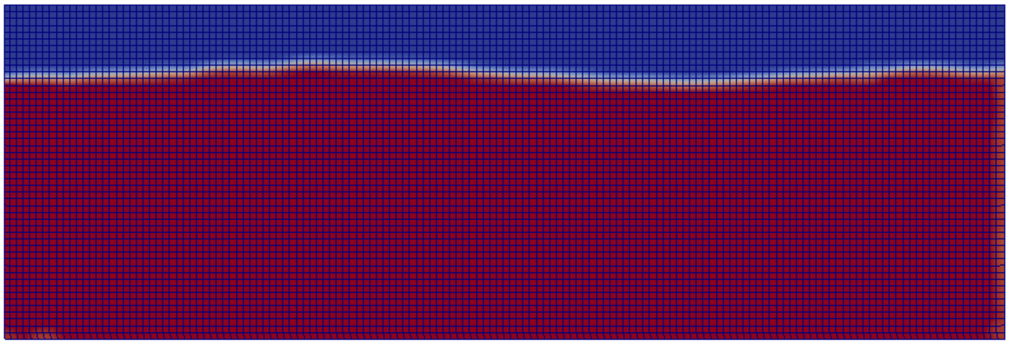


Figure 313.3: Simulation result of 2D sloshing box

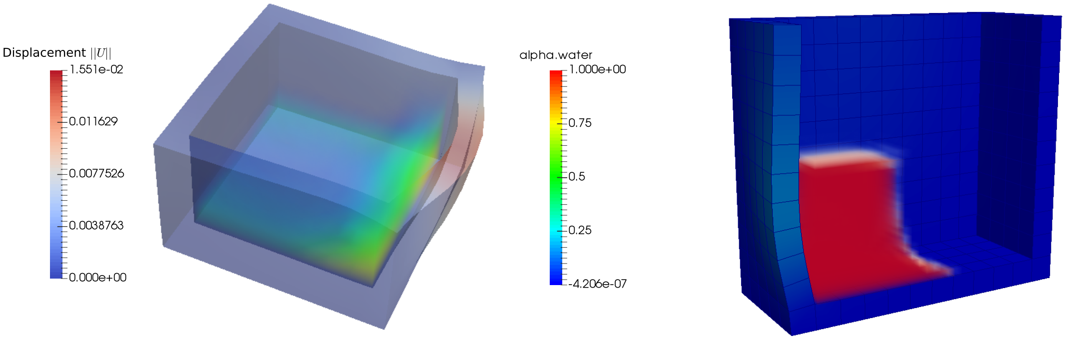


Figure 313.4: Box sloshing under falling water

313.2.2 Dam Break

313.3 Verification & Validation

Verification and validation is crucially important to guarantee the accuracy of simulation result. Verification and validation (V & V) procedure of SFI problem involve three aspects: V & V for response of solid domain (i.e. V & V for RealESSI as a solver for solid mechanics), V & V for response of fluid domain (i.e. V & V for OpenFOAM as a solver for free surface flow) and V & V for SFI (i.e. V & V for interaction between solid domain and fluid domain).

Since RealESSI developed by Jeremić et al. (1988-2025) has rigorous V & V procedure, the focus here is on V & V of OpenFOAM and SFI.

313.3.1 Free Surface Flow validation

The functionality of OpenFOAM as a solver for free surface flow is validated in this section. A numerical validation test is conducted based on the experiments reported by Martin and Moyce (1952).

A rectangular column of water, in hydrostatic equilibrium, is confined between two vertical walls, as shown in figure 313.5. The water column is 1 unit wide and 2 unit high. At the beginning of the calculation, the right wall (dam) is removed and water is allowed to flow out along a dry horizontal floor.

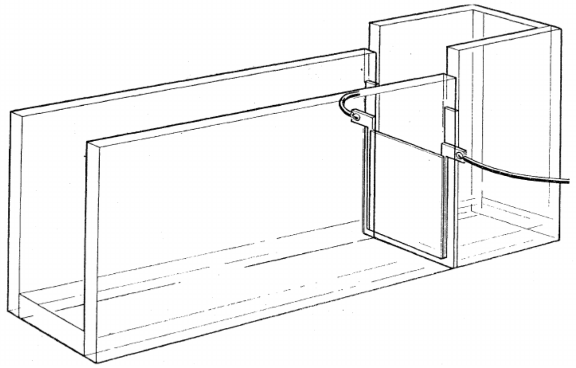


Figure 313.5: Experiment set up by]Martin and Moyce (1952)

The real-time position of the leading edge of the water is recorded during the experiment. This is a good test problem because it has simple boundary conditions and a simple initial configuration. 2D numerical models with two different types of mesh size ($\Delta x = 0.1m$ and $\Delta x = 0.05m$) are built (figure 313.6).

The comparison result between numerical solution, experiment result and benchmark solution by Hirt and Nichols (1981) is presented in figure 313.7.

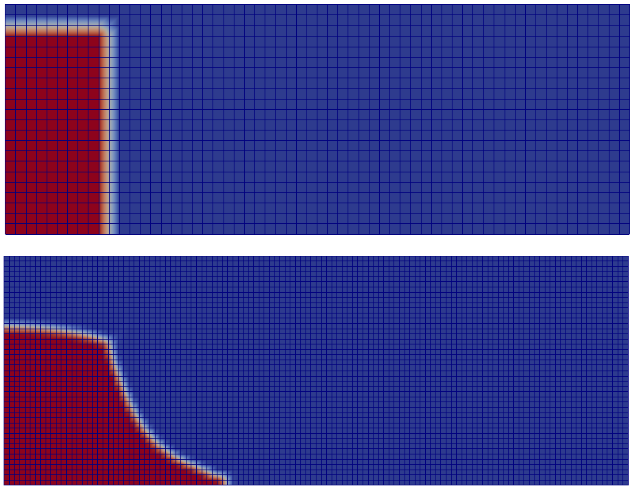


Figure 313.6: Numerical model for validation of free surface flow

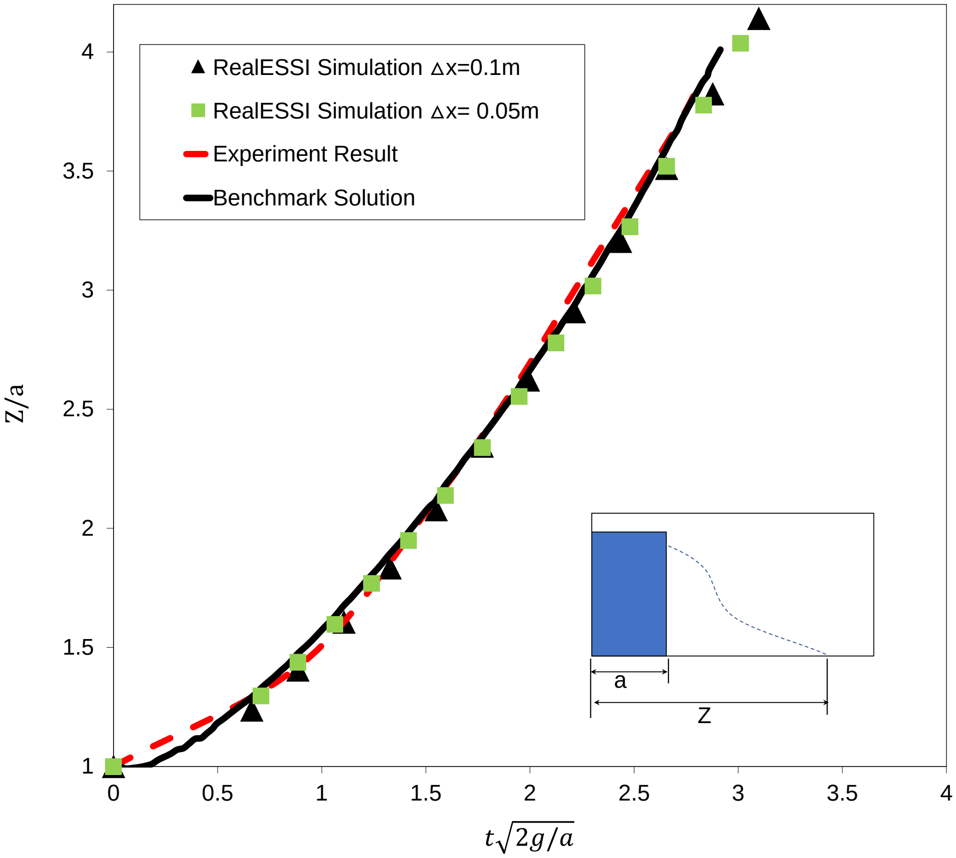


Figure 313.7: Validation result of free surface flow

It can be seen that numerical simulation matches well with both experiment result and benchmark solution. OpenFOAM, as a finite volume solver for free surface flow based on VOF method, is reliable.

313.3.2 Mass conservation verification

As mentioned in section 111.4.6, for a closed fluid system, mass conservation should be strictly satisfied. The total volume of fluid in the system can be calculated with equation 313.1.

$$V_{total} = \sum_{i=1}^{n} \alpha_i V_i \tag{313.1}$$

The time history record of total volume in the numerical example (section 313.2.1) is given in figure 313.8. It can be seen that the total volume remains almost constant during the simulation of SFI. In this example, after 400 time steps, the relative mass change is only 0.25%, which demonstrates that our coupling program has excellent performance regardng mass conservation.

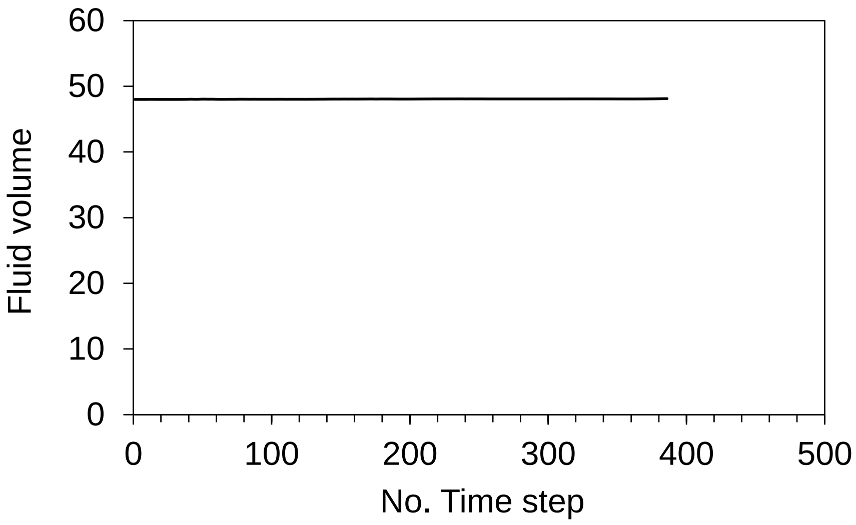


Figure 313.8: Time history record of total fluid volume

313.3.3 SFI Verification & Validation

A box sloshing numerical experiment (rectangular tank under sway oscillations $X(t) = Asin(wt)$) is taken as the verification and validation test for solid fluid interaction. The box is 1.0 m long (L) and 0.8 m height (H). The depth (D) of submerged water is 0.5 m. The natural frequency of the tank can be calculated according to Lamb (1932):

$$\omega_i = \sqrt{g\frac{\pi i}{L}tanh(\frac{\pi i}{L}D)} \quad i = 1, 2, 3, ... \tag{313.2}$$

where ω_i is the natural frequency, and g is gravitational acceleration. The lowest linear mode ω_i is of primary importance for the tank sloshing phenomenon.

Based on the original analytical solution of Linton and McIver (2001), Jin et al. (2014) gives the equation for non-dimensional free surface elevation η_{max}/A as shown in equation 313.3, where $b = L/2, \mu_m = (m + 0.5)\pi/b, K_m = \mu_m tanh \lambda_m D, K = \omega^2/g$.

$$\frac{\eta_{max}}{A} = \frac{\omega}{gA}|A\omega b + \sum_{m=0}^{\infty}|\frac{2K}{\mu_m^2 b(K_m - K)}| \tag{313.3}$$

Jin et al. (2014) also conducted a 2D sloshing experiment and report detailed response of free surface elevation under different excitation magnitude A and frequency ω. Here A is fixed as 2.5 mm and different frequency values ($\omega/\omega_1 = 0.5 \sim 2$) are adopted to implement verification and validation numerical test. The initial setup of numerical experiment can be seen in figure 313.9(a). Figure 313.9(b) and figure 313.9(c) are the sloshing response under excitation of first-mode ($\omega/\omega_1 = 1$) and third-mode ($\omega/\omega_3 = 1.793$) resonance frequency.

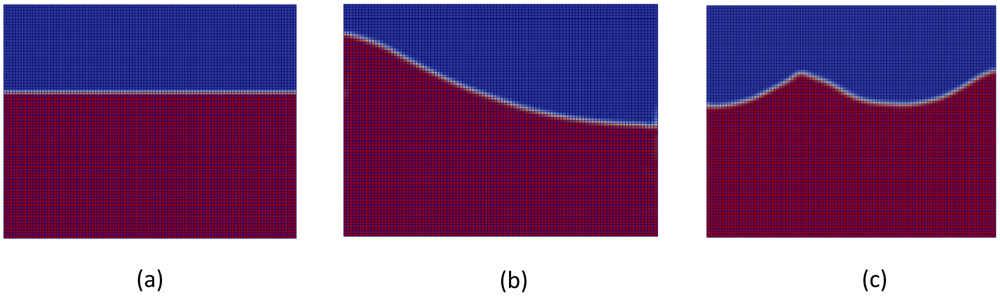

(a) (b) (c)

Figure 313.9: Numerical experiment of solid fluid interaction: (a) Initial setup (b) Sloshing response under first-mode resonance excitation (c) Sloshing response under third-mode resonance excitation

Compared with experiment record (figure 313.10) by Jin et al. (2014), it can be seen that the mode shape from our numerical simulation is same as the experimental observations.

The numerical results of η_{max}/A are also plotted into the same figure as theoretical predication and experiment record, as shown in figure 313.11. Very good agreement can be observed, especially between numerical simulation and experiment result. This manifests that suitable mathematical equations about SFI has been numerically solved in a correct way. The SFI coupling implementation in RealESSI is reliable.

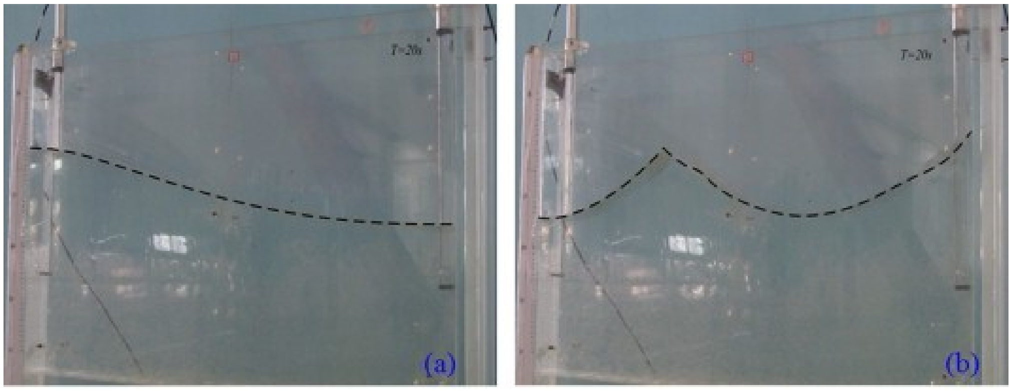


Figure 313.10: Resonant wave shape: (a) First-order mode (b) Third-order mode reproduced from Jin et al. (2014)

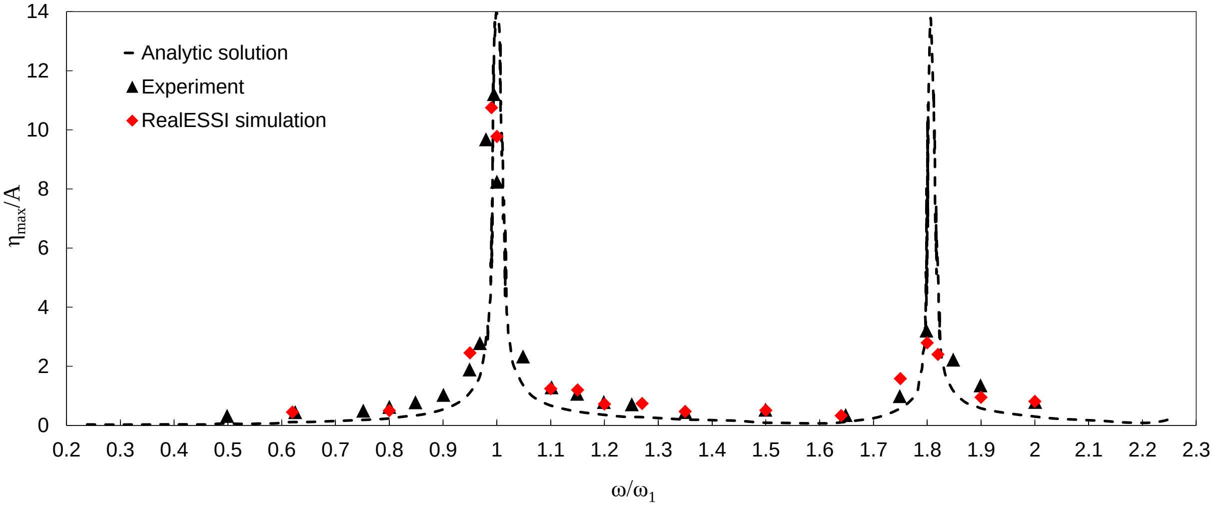


Figure 313.11: Comparison among theoretical prediction, experiment observation and numerical simulation

Chapter 314

Quality Management System

(2002-2005-2018-2019-2020-2021-)

314.1 Chapter Summary and Highlights

314.2 Reasoning Behind this Activity

Quality assurance for a numerical modeling system is of highest importance. A good, sound Quality Management System is therefor very important. This section provides details of quality management system used for the development and quality assurance (QA) for the Real-ESSI Simulator system (Jeremić et al., 1988-2025).

314.3 Real-ESSI Simulator System Quality Management System, based on ISO/IEC/IEEE 90003 Standard

This section is based on ISO/IEC/IEEE 90003 Developers et al. (2018). ISO/IEC/IEEE 9003 is a standard developed by the International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), and Institute of Electrical and Electronics Engineers (IEEE), as a guidance for the application of ISO 9001:2015 standard to the acquisition, supply, development, operation and maintenance of computer software and related support services.

314.3.1 Real-ESSI Simulator Developer Organization

314.3.1.1 Internal Issues

Internal issues that are relevant to the Real-ESSI Simulator quality management system:

- legal
- Technological
- Competitive
- Market
- Cultural
- Social
- Economic environment, local, national, international

314.3.1.2 Internal Issues

External issues that are relevant to the Real-ESSI Simulator quality management system:

- Values
- Culture
- Knowledge
- Performance

314.3.1.3 External and Internal Issues for the Real-ESSI Simulator

External and Internal issues for the Real-ESSI Simulator can include:

- Use of "cloud" services, that is beneficial for ease of access and business continuity, however needs research to ensure lowering all beneficial effects. The Real-ESSI Simulator is fully deployed on Amazon Web Services (AWS), and is tightly managed by development group as well as ESSI Consultants. AWS was chosen after in depth examining other cloud services, namely Google cloud, that actually provided no dedicated tightly parallel computers as of 2018, Microsoft Azure, that was fast but very expensive, and local parallel cloud, Real-ESSI Parallel Computer, that works well for developers, but requires much management for outside users.
- Use of personal computers, laptops, can create a problem in managing safety and security of Real-ESSI Simulator sources. Developers are aware of this issue and have agree do tight safety and security and have signed a licensing agreement that commits them to managing such safety and security.
- Risk of external attack on developers computers and network and on Real-ESSI Simulator deployed computers is controlled by regular, up to date Linux and AWS security.
- Delivery of the Real-ESSI Simulator is mostly handled through AWS, while there are a number of example of remote deployment, whereby remote users commit fully utilizing and following the Real-ESSI Simulator quality management system.
- Legal and operational issues for Real-ESSI Simulator use in contgext of safety, security and mission assurance.

314.3.1.4 Needs and expectations of Interested Parties

Interested parties that are relevant to Quality Management System and their requirements

- Customers, professional practice companies, require working system, that is efficient, easy to use and that can provide more optimal, more economic and more safe designs than what is currently available.
- Partners, collaborators, require in depth knowledge of the system, possibility to influence changes and additions to the Real-ESSI Simulator
- Staff, require stable and nourishing research and development environment

- DOE, not sure anymore?
- NRC, not sure anymore?
- UN-IAEA, ...
- Professional Practice: development of the system that improves design and assessment process, while maintaining practicality, achieving highly efficient, minimally disrupting training for use Real-ESSI
- Research Community: Contributions to modeling and simulations, and use of Real-ESSI to investigate new behavior of ESSI systems.
- Developers form the Real-ESSI Simulator group:
- Competitors:
  - SimCenter ?
  - French project?
  - Linear Elastic community, SASSI,
  - nonlinear FEM programs, LS-Dyna...

314.3.2 Scope of the Real-ESSI Simulator Quality Management System

Plan-do-Check-Act (PDCA) cycle

Risk-based thinking

Quality Management Principles

314.4 Real-ESSI Simulator System Quality Management System, based on ASME NQA-1 Standard

ASME Nuclear Quality Assurance (NQA-1) is a standard developed by the American Society of Mechanical Engineers (ASME) that provides quality assurance guidance and certification for organizations supplying items and services which provide a safety function for nuclear installations.

314.4.1 ASME NQA-1 for the Real-ESSI Simulator System

Chapter 315

Comparison with Other Programs

(2016-2017-2020-2021-)

(In collaboration with Dr. Nima Tafazzoli, and Dr. Yuan Feng)

315.1 Chapter Summary and Highlights

315.1.1 Reasoning Behind this Activity

I personally do not like or approve code to code comparison. If results between two codes are the same, this does not prove that any tested code is right. If results between two codes are different, which code is right?

However, a number of professionals and industry in general feel very comfortable with some usually used codes/programs. In this sense, we provide code comparison with select widely used codes, to satisfy professionals. If results turn out to be the same (impossible to have exactly same results, but say very close) that is good for all codes involved, as we have a full verification suit and guaranty our accuracy to within limits of verification tests we used. If results are different, we still claim accuracy, as we have full verification suit and guaranty our accuracy to within limits of verification tests we used.

Part 400

Education, Training and Modeling, Simulation Examples

Chapter 401

Ten Section Course on Nonlinear Finite Element Methods for Realistic Modeling and Simulation of Earthquakes, and Soils, and Structures, and their Interaction, Real-ESSI

(1998-2021-)

401.1 Delivery

Instructor: Boris Jeremić, email: `Jeremic@ucdavis.edu`

Class meeting: two hour lecture/meetings, twice a week, flipped classroom method

Office hours: two hours, twice a week

Course delivery: live and recorded lectures, and live discussions

Course WWW: `http://sokocalo.engr.ucdavis.edu/~jeremic/Classes/ECI280A/`

401.2 Objectives

This course will provide students with state of the art finite element methods, numerical analysis tools and models for solving elastic–plastic problems in geotechnical and structural engineering. Focus is on analysis of Soils, Structures and their Interaction (SSI). Presented will be computational formulation, numerical techniques and models for static, nonlinear, elastic-plastic finite element methods that are used in professional practice and research. Both sequential and parallel computational approaches will be presented and used.

During this course students will:

- Learn about linear and nonlinear finite element modeling and simulation
- Select and calibrate nonlinear, elastic-plastic models for soil, rock, concrete, steel and interfaces
- Perform linear elastic and nonlinear, elastic-plastic analysis of solids and structures made of soil, rock, concrete, steel and their interfaces
- Perform elastic and nonlinear, inelastic analysis of soils-structure systems
- Become proficient in performing nonlinear analysis for soils, structures, interfaces, using different levels of sophistication, from simplified models to high fidelity elastic-plastic models of soil structure systems on sequential and parallel computers.

Who Should Attend?

Students and practicing engineers who want to learn about and expand their knowledge of modeling and simulation for nonlinear/inelastic material behavior, for soils, rock, and structures,

401.3 Additional Information

Lecture Notes: `http://sokocalo.engr.ucdavis.edu/~jeremic/LectureNotes/`.

Recorded Lectures: `http://sokocalo.engr.ucdavis.edu/~jeremic/Online-Education/`.

Computers: Most of the problems in this course will require numerical simulations. A finite element modeling system called Real-ESSI Simulator (`http://real-essi.us`) is available for computers running Windows (through WSL), MacOS, Linux. Please refer to `http://real-essi.us` to find out how to use Real-ESSI on local computers or on AWS computers. Other programs can be used as well, as long as they provide modeling and simulation capabilities that are required for assignments, example problems and term project. Both sequential and parallel computers will be used.

Problems: Assigned weekly, students are expected to attempt to develop solutions. You are encouraged to discuss the approach to problem solutions with other students in the course as well as with the instructor.

Examples: Model development, finite element models, finite element mesh, boundary conditions, material models, loads, model verification process, linear and nonlinear elastic FEM with solids and structural elements,

There are a large number of examples available at the Real-ESSI Simulator web site:
`http://real-essi.info/`, or `http://real-essi.us/`, in these documents:

- Real-ESSI Simulator Examples Collection
- Real-ESSI Simulator Short Course Examples Collection

Term Project: Term project will involve work related to developing or using numerical models for numerically simulating elastic–plastic problem of your choice, related to your interests. Term projects will be presented at the end of quarter.

Grading: TBD

Examination: TBD

Literature:

- The Finite Element Method, *Olgierd Cecil Zienkiewicz and Robert L. Taylor*, McGraw-Hill Book Company, Volumes 1 and 2, ISBN 0-07-084175-6
- Non - Linear Finite Element Analysis of Solids and Structures Volume 1: Essentials, *Crisfield, M. A.*, John Wiley and Sons, Inc. New York, 1991 , ISBN 0 471 92956 5 v.1
- Finite Element Procedures in Engineering Analysis, *Klaus-Juergen Bathe*, Prentice Hall, ISBN 0-13-301458-4

- Constitutive Laws for Engineering Materials With Emphasis on Geologic Materials *Chandakant S. Desai and Hema J. Siriwardane*, Prentice–Hall, Inc. Englewood Cliffs, NJ 07632, ISBN 0-13-167940-6
- Plasticity for Structural Engineers *W. F. Chen and D. J. Han* , Springer Verlag, 1988 ISBN 0-387-96711-7
- Boris Jeremić, Zhaohui Yang, Zhao Cheng, Guanzhou Jie, Nima Tafazzoli, Matthias Preisig, Panagiota Tasiopoulou, Federico Pisano, Jose Abell, Kohei Watanabe, Yuan Feng, Sumeet Kumar Sinha, Fatemah Behbehani, Han Yang, and Hexiang Wang. Nonlinear Finite Elements: Modeling and Simulation of Earthquakes, Soils, Structures and their Interaction. University of California, Davis, CA, USA, 1989-2022. ISBN: 978-0-692-19875-9

401.4 Teaching Plan, Topics

401.4.1 Section I, Introduction

Introduction: Course objectives, methodology, computer modeling and simulation

Modeling and Simulation System Setup: Introduction to the Real-ESSI Simulator system. Computational Mechanics field of study, kinematics of deformation, strain, stress, linear and nonlinear elasticity, equilibrium relations, finite element method review, nonlinear analysis cycles; Introduction to sequential and parallel computers.

Lectures, recordings and slides:

- Introduction to Modeling and Simulation: PDF slides, MP4 recording
- Introduction to Modeling Simplifications, Epistemic Uncertainty: PDF slides, MP4 recording
- Introduction to Parametric, Aleatory Uncertainty: PDF slides, MP4 recording

Reading: Lecture Notes: 101, 201, 205; Papers/Reports: CM988, CM2714, CM2715, CM3170

Examples: Model development, finite element models, finite element mesh, boundary conditions, material models, loads, model verification process, linear and nonlinear elastic FEM with solids and structural elements. see examples collection at `http://real-essi.info/` or `http://real-essi.us/`,

Problems:

1. Model a $0.1m \times 0.1m \times 1.0m$ linear elastic cantilever beam, with transversal end forcing, using:
    - Single and ten Bernoulli beam elements
    - Single and ten Timoshenko beam elements
    - 1 $(1 \times 1 \times 1)$; 10 $(1 \times 1 \times 10)$; 24 node solid brick elements
2. Model a $0.1m \times 0.1m \times 1.0m$ simple shear linear elastic test using
    - 1 $(1 \times 1 \times 1)$ and 32 $(4 \times 4 \times 4)$ 8 node solid brick elements
    - 1 $(1 \times 1 \times 1)$ and 32 $(4 \times 4 \times 4)$ 24 node solid brick elements
3. Model a $0.1m \times 0.1m \times 1.0m$ pure shear (no rotations) linear elastic test using
    - 1 $(1 \times 1 \times 1)$ and 32 $(4 \times 4 \times 4)$ 8 node solid brick elements
    - 1 $(1 \times 1 \times 1)$ and 32 $(4 \times 4 \times 4)$ 24 node solid brick elements

401.4.2 Section II, Inelastic Finite Elements

Theory: Expanding the matrix deformation method, linear elastic truss element, Beams (Bernoulli, Timoshenko), solids, plates, walls (plane stress), shells, stiffness, forces, displacements, interpolating functions for displacements. Local and global equilibrium. Internal and external forces,

Lectures, recordings and slides:

- Introduction to the Finite Element Method (FEM): PDF slides, MP4 recording
- Derivation of FEM equations of motions for single phase, dry material: PDF slides, MP4 recording
- Derivation of FEM equations of motions for coupled, two phase, fully and partially saturated material, u-p-U formulation: PDF slides, MP4 recording

Reading: Lecture Notes: 101, 102; Papers/Reports: CM81, CM125, CM1835, CM2714, CM3155, CM3155

Examples: Truss, beam, solid bricks. external forces. internal forces (sectional forces, stresses). Generalized nodal displacements and internal deformation (curvature, axial, shear, volumetric, general strains),

Problems:

1. Develop a set of simple, single element examples using truss, beam and brick finite elements with simple static loads, and extract sectional forces, stress, strain and strain energy from results.

401.4.3 Section III, Micromechanics of Elasto-Plasticity

Theory: Micro-mechanical origins of elasto-plasticity, particles in contact, friction, Hertz, Mindlin-Deresiewicz contact/interface

Lectures, recordings and slides:

- Micromechanical origins of elasto-plasticity:
- Lectures by Prof. Stein Sture (University of Colorado, Boulder) on micromechanical origins of elasto-plasticity are available HERE,

Reading: Lecture Notes: 103; Papers/Reports: CM1000, CM1830, CM1831,

Examples: Particle contact problems.

Problems:

1. Develop a simple, 2D, plane-strain model of two particles in contact and apply normal and then shear forces,
2. For the above model, vary normal and shear forces. Comment on results.

401.4.4 Section IV, Incremental Elastic-Plastic Theory

Theory: Incremental, continuum elasto-plasticity, Material Models, perfectly plastic, hardening and softening. Explicit, forward Euler and Implicit, backward Euler, constitutive integrations,
Lectures, recordings and slides:

- Introduction to the incremental theory of elasto-plasticity: PDF slides, MP4 recording, YouTube video
- Explicit solution to the constitutive elastic-plastic problem: PDF slides, MP4 recording, YouTube video
- Implicit solution to the constitutive elastic-plastic problem: PDF slides, MP4 recording, YouTube video

Reading: Lecture Notes: 104; Papers/Reports: CM3199
Examples: Constitutive integrations, explicit and implicit computations: single element response using select elastic-plastic material models: von Mises, Drucker-Prager, Cam Clay. Perfectly plastic, isotropic hardening, kinematic hardening models and cyclic response. Inelastic, fiber (1D) and 3D structural models for concrete and steel.
Problems:

1. Develop a constitutive only linear elastic example. with monotonic loading and vary elastic modulus and Poisson's ratio. Comment on results
2. Develop a constitutive only elastic-perfectly plastic von-Mises example, for monotonic loading, and vary elastic properties, yield strength. Comment on results.
3. For the above developed example, develop results using explicit and implicit constitutive integrations. Vary step size, integration algorithm, tolerances. Comment on results.
4. For the above developed examples, use two cycles of cyclic loading. Comment on Results.

401.4.5 Section V, Inelastic, Elasto-Plastic Solids Modeling

Theory: Continuation: Incremental elasto-plasticity. Material modeling for practical applications. Advanced topics in constitutive elasto–plasticity, stability and accuracy, errors in constitutive integrations, problematic incremental steps, energy dissipation, sub-incrementation, line search, model calibrations.

Lectures, recordings and slides:

- Choice of elastic-plastic material models for soils and interfaces/contacts/joints: PDF slides, MP4 recording YouTube video
- Calibration of elastic-plastic material models for sand: PDF slides, MP4 recording YouTube video
- Calibration of elastic-plastic material models for clay: PDF slides, MP4 recording YouTube video

Reading: Lecture Notes: 104, 402, 403, 512; Papers/Reports:

Examples: Errors in constitutive modeling, sub-increments. Material model calibration, Constitutive modeling of soil, rock, concrete, steel: von Mises, Drucker-Prager, Cam Clay, SaniSand, rounded Mohr-Coulomb, Pisano. Modeling G/G_{max} and damping response. Nonlinear, elastic-plastic structural models for concrete and steel (1D, 3D).

Problems:

1. Develop a single element, 8 and 24 node brick, linear elastic example. with monotonic loading and vary elastic modulus and Poisson's ratio. Comment on results
2. Develop a single element elastic-perfectly plastic von-Mises example, for monotonic loading, and vary elastic properties, yield strength. Vary number of integration, Gauss points. Comment on results.
3. For the above developed example, develop results using explicit and implicit constitutive integrations. Vary step size, integration algorithm, tolerances. Comment on results.
4. For the above developed examples, use two cycles of cyclic loading. Comment on Results. Use both axial loading and shear loading.

401.4.6 Section VI, Inelastic, Elastic-Plastic Interfaces, Joints, Contacts Modeling

Theory: Interface/Joint/Contact modeling: Hard contact, Soft contact. Axial contact stiffness, shear contact stiffness. Interface gap opening and closing. Saturated contacts, effective stress and buoyant forces on foundations.

Lectures, recordings and slides:

- Choice of elastic-plastic material models for soils and interfaces/contacts/joints: PDF slides, MP4 recording YouTube video
- Calibration of elastic-plastic material models for interfaces/contacts/joints: PDF slides, MP4 recording YouTube video

Reading: Lecture Notes: 104, 403, 512; Papers/Reports:

Examples: Interface: concrete to soil and rock, steel to soil and rock. Gap opening, closing. Shear interface, slip, no slip.

Problems:

1. Develop a two solid element example that are connected using force based interface elements.
2. Develop a two solid element example that are connected using stress based interface elements.
3. For the above developed example, use normal loading and vary load step size. Comment on results.
4. For the above developed example, use normal and then shear loading and vary load step size. Comment on results.
5. For the above developed example, vary interface properties, use interface/contact/joint properties for soil on concrete, soil on steel, concrete on concrete, &c. Comment on results.

401.4.7 Section VII, Inelastic, Elastic-Plastic Structural Modeling

Theory: Inelastic structural models, beams, plates, walls and shells.

Lectures, recordings and slides:

- Choice of elastic-plastic material models for structural elements, beams and walls/plates/shells: PDF slides, MP4 recording, YouTube video
- Calibration of elastic-plastic material models for concrete, in reinforced beams and walls/plates/shells: PDF slides, MP4 recording YouTube video
- Calibration of elastic-plastic material models for steel, in reinforced beams and walls/beams/shells: PDF slides, MP4 recording YouTube video

Reading: Lecture Notes: 102, 403, 512; Papers/Reports:

Examples: Nonlinear analysis of structures. Steel Frames. Reinforces concrete frames, walls, plates, shells.

Problems:

1. Develop a nonlinear truss model, and load it using monotonic and cyclic loading up to yielding and past yielding. Comment on results.
2. Develop a nonlinear beam model, and load it in bending using monotonic and cyclic loading up to yielding and past yielding. Comment on results.
3. Develop a two solid element example that are connected using stress based interface elements. Comment on results.

401.4.8 Section VIII, Nonlinear Analysis Progress

Theory: Analysis Progress. Stages, increments, iterations, elastic–plastic stiffness matrix, pure incremental methods, force residuals, Newton iterative algorithm for finite element level iterations, constraints to the global (force residual) system of equations, equilibrium iterations, convergence, load control, displacement control, arc-length, hyper-spherical constraint, convergence criteria, automatic step size control, line search, stability and accuracy. Sequential and parallel computations for inelastic, nonlinear solids and structures.

Lectures, recordings and slides:

-

Reading: Lecture Notes: 102, 107, 403; Papers/Reports:

Examples: Nonlinear analysis of structures and solids, elastic plastic solids, structures and contacts. Staged analysis steps, incremental only analysis with no equilibrium enforcement, incremental-iterative analysis, with equilibrium enforcement, convergence criteria (force, displacement), convergence tolerances, step size control.

Problems:

1. Develop a nonlinear analysis, using all of previous examples, that will feature explicit, no equilibrium check simulation. Comment on results.
2. Develop a nonlinear analysis, using all of previous examples, that will feature implicit, equilibrium check simulation. Comment on results.

401.4.9 Section IX, Verification and Validation

Theory: Verification, Validation and Prediction, basic theory, solution verification, manufactured solutions, validation experiments, prediction under uncertainty,

Lectures, recordings and slides:

-

Reading: Lecture Notes: 301, 302, 303..., 313, 314...; Papers/Reports:

Examples: Solution verification examples for elements, material models, constitutive integration algorithms, solution advancement algorithms.

Problems:

1. Choose a model of your interest, and develop a list of verification examples for all components of your model.
2. For the above model, develop a list of validation examples.

401.4.10 Section X, Practical Considerations for Nonlinear Analysis

Theory: Elastic–plastic FEM modeling (practical recommendations for development and analysis of nonlinear (elastic-plastic) finite element models, phased development of general FEM (and ESSI in particular) models. Core Functionality for inelastic/nonlinear modeling, Energy dissipation. Notes on sequential and parallel computing.

Lectures, recordings and slides:

-

Reading: Lecture Notes: 510, 512; Papers/Reports:

Examples: Illustrations of algorithms and models described above, benefits and detriments of different algorithms and models.

Problems:

1. Develop a realistic nonlinear analysis model of your choice, perhaps the one developed in previous section and experiment with all/most above models and algorithms. Comment on results.

Chapter 402

Ten Section Course on Dynamic Finite Element Methods for Realistic Modeling and Simulation of Earthquakes, and Soils, and Structures, and their Interaction, Real-ESSI

(1998-2021-)

402.1 Delivery

Instructor: Boris Jeremić, email: `Jeremic@ucdavis.edu`

Class meeting: two hour lecture/meetings, twice a week, flipped classroom approach

Office hours: two hours, twice a week

Course delivery: live and recorded lectures, and live discussions

Course WWW: `http://sokocalo.engr.ucdavis.edu/~jeremic/Classes/ECI280B/`

402.2 Objectives

This course will provide students with state of the art finite element methods, numerical analysis tools and models for solving dynamic problems in geotechnical and structural engineering, Focus is on analysis of Earthquakes, Soils, Structures and their Interaction (ESSI). Presented will be computational formulation, numerical techniques and models for dynamic, nonlinear, elastic-plastic finite element methods that are used in professional practice and research. In addition, sequential and parallel computing approaches will be explained, and used.

During this course, students will:

- Learn about dynamic finite element modeling and simulation
- Develop dynamic modeling and simulations for linear and nonlinear soils and structures
- Perform dynamic, linear and nonlinear analysis of solids and structures made of soil, rock, concrete, steel and their interfaces, joints and contacts
- Develop and use of one compoinent (1C), 3×1C and 3C seismic motions from given earthquake records and from analytic wave solutions
- Perform dynamic, nonlinear/inelastic earthquake soil structure interaction (ESSI) analysis
- Become proficient in performing nonlinear ESSI analysis using different levels of sophistication, from simplified models to high fidelity elastic-plastic ESSI models on sequential and parallel computers

Who Should Attend?

Students and practicing engineers who want to learn about and expand their knowledge of modeling and simulation for dynamic, nonlinear/inelastic material behavior, for soils, rock, and structures,

402.3 Additional Information

Lecture Notes: `http://sokocalo.engr.ucdavis.edu/~jeremic/LectureNotes/`.

Recorded Lectures: `http://sokocalo.engr.ucdavis.edu/~jeremic/Online-Education/`.

Computers: Most of the problems in this course will require numerical simulations. A finite element modeling system called Real-ESSI Simulator (`http://real-essi.us`) is available for computers running Windows (through WSL), MacOS, Linux. Please refer to `http://real-essi.us` to find out how to use Real-ESSI on local computers or on AWS computers. Other programs can be used as well, as long as they provide modeling and simulation capabilities that are required for assignments, example problems and term project.

Problems: Assigned for each section, students are expected to attempt to develop solutions. You are encouraged to discuss the approach to problem solutions with other students in the course as well as with the instructor.

Term Project: Term project will involve work related to developing or using numerical models for numerically simulating elastic–plastic problem of your choice, related to your interests. Term projects will be presented at the end of quarter.

Grading: TBD

Examination: TBD

Literature:

- The Finite Element Method, *Olgierd Cecil Zienkiewicz and Robert L. Taylor*, McGraw-Hill Book Company, Volumes 1 and 2, ISBN 0-07-084175-6
- Non - Linear Finite Element Analysis of Solids and Structures Volume 1: Essentials, *Crisfield, M. A.*, John Wiley and Sons, Inc. New York, 1991 , ISBN 0 471 92956 5 v.1
- Finite Element Procedures in Engineering Analysis, *Klaus-Juergen Bathe*, Prentice Hall, ISBN 0-13-301458-4
- Constitutive Laws for Engineering Materials With Emphasis on Geologic Materials *Chandakant S. Desai and Hema J. Siriwardane*, Prentice–Hall, Inc. Englewood Cliffs, NJ 07632, ISBN 0-13-167940-6
- Plasticity for Structural Engineers *W. F. Chen and D. J. Han* , Springer Verlag, 1988 ISBN 0-387-96711-7
- Boris Jeremić, Zhaohui Yang, Zhao Cheng, Guanzhou Jie, Nima Tafazzoli, Matthias Preisig, Panagiota Tasiopoulou, Federico Pisano, Jose Abell, Kohei Watanabe, Yuan Feng, Sumeet Kumar

Sinha, Fatemah Behbehani, Han Yang, and Hexiang Wang. Nonlinear Finite Elements: Modeling and Simulation of Earthquakes, Soils, Structures and their Interaction. University of California, Davis, CA, USA, 1989-2022. ISBN: 978-0-692-19875-9

402.4 Teaching Plan, Topics

402.4.1 Section I, Introduction

Introduction: Course objectives, methodology, computer modeling and simulation; Dynamics of nonlinear structures and soils during earthquakes, examples Preliminary theory, terminology, issues to be addressed: Deformation, kinematics of moving systems, elasticity, dynamic equilibrium relations, d'Alembert's principle, forces in dynamic equilibrium, mass, damping, stiffness, external force, single degree of freedom systems,

Modeling and Simulation System Setup: Introduction to the Real-ESSI Simulator system. Computational Mechanics field of study, kinematics of deformation, strain, stress, linear and nonlinear elasticity, equilibrium relations, finite element method review, nonlinear analysis cycles;

Lectures: Recorded lectures covering topics for this section can be found in Section 404.1.1 on Page 1937 in Jeremić et al. (1989-2025) (Lecture Notes URL).

Reading: Lecture notes: 101, 102; Papers/Reports:

Examples: Model Development, simple models vs sophisticated models, pre-processing, post-processing, results visualization.

Problems:

1.
2.
3.

402.4.2 Section II, Dynamic FEM

Dynamic FEM Theory: Dynamic finite element method (FEM) equations, virtual work method in dynamics, nonlinear dynamic equations of motion, consistent and lumped mass, velocity and displacement proportional damping/energy dissipation, Rayleigh and Caughey viscous damping, linear and nonlinear material behavior.

Lectures: Recorded lectures covering topics for this week can be found in Section 404.1.3 on Page 1939 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

Reading: Lecture notes: 102; Papers/Reports:

Examples: Structural and solid elements and models, dynamic excitations, resonance, linear and nonlinear (elastic and inelastic/elastic-plastic) material models, viscous damping, consistent and lumped mass matrix.

Problems:

1.

2.

3.

402.4.3 Section III, Nonlinear FEM

Nonlinear FEM: Elasto-plasticity, material models for dynamics of soils and structures, material parameter calibration, uncertainty in material parameters, explicit and implicit constitutive integrations. Sequential and parallel computations

Lectures: Recorded lectures covering topics for this week can be found in Section 404.1.3 on Page 1939 and Section 404.1.4 on Page 1940 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

Reading: Lecture notes: 103, 104; Papers/Reports:

Examples: Elastic plastic solids, beams and shells, material energy dissipation, material damping

Problems:

1.

2.

3.

402.4.4 Section IV, Time Domain Nonlinear Dynamic FEM

Time Domain Nonlinear Dynamic FEM: Direct, time marching solution for dynamics of nonlinear, inelastic systems, general Newmark family of methods, stability and accuracy, nonlinear resonance, numerical damping, explicit and implicit algorithms, unconditionally and conditionally stable Newmark and Hilber–Hughes–Taylor α–method, stability and accuracy, examples)

Lectures: Recorded lectures covering topics for this week can be found in Section 404.1.3 on Page 1939 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

Reading: Lecture notes: 108; Papers/Reports:

Examples: Nonlinear solid and structural models direct time integration, step size, damping (material, viscous, numerical), stable and unstable computations.

Problems:

1.

2.

3.

402.4.5 Section V, Earthquake Soil Structure Interaction (ESSI)

Earthquake Soil Structure Interaction (ESSI): Background, problem definition, seismic motions, seismic body and surface wave field, seismic energy propagation, free field motions, beneficial and detrimental effects, balancing input and dissipated energy.

Lectures: Recorded lectures covering topics for this week can be found in Section 404.1.8 on Page 1946 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

Reading: Lecture notes: 502; Papers/Reports:

Examples: Analytic development of ground motions, 3D vs 1D motions, seismic energy calculations.

Problems:

1.

2.

3.

402.4.6 Section VI, Seismic Motions

Seismic Motions: Free field vs ESSI motions, incoherent motions, Domain Reduction Method, boundary conditions, radiation damping, 3D inclined wave fields vs 1D vertical motions, nonlinear wave propagation simulations, time step size, element size, earthquake modeling.

Free field motions development, 1D motions, 3D/6D motions, regional scale models, Geophysical models,

Lectures: Recorded lectures covering topics for this week can be found in Section 404.1.7 on Page 1945 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

Reading: Lecture notes: 502, 511, 705, 706; Papers/Reports:

Examples: Real ESSI and analytic wave field models for free field and local (DRM) motions, element and time step size and propagation of (required) frequencies. Vertical and inclined waves development, and input into SSI models

Problems:

1.

2.

3.

402.4.7 Section VII, Coupling with Internal and External Fluids

Dynamics of Coupling with Pore Fluids and External Fluids: Fully coupled, porous solid – pore fluid systems formulation, discretization, basic system of DOFs, coupling damping forces, specialization to slow (consolidation) and fast phenomena (ESSI, liquefaction), boundary conditions, initial conditions, stability and accuracy of various algorithms. Coupling with external fluids, pools, reservoirs.

Lectures: Recorded lectures covering topics for this week can be found in Section 404.1.3 on Page 1939 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

Reading: Lecture notes: 102, 505; Papers/Reports:

Examples: 1D and 3D coupled examples, consolidation, liquefaction and de-liquefaction waves, piles in liquefied soil... Coupling with external fluids, using OpenFOAM...

Problems:

1.

2.

3.

402.4.8 Section VIII, Dynamic Stochastic Elastic-Plastic FEM (SEPFEM)

Stochastic Elastic-Plastic Dynamic FEM:

Lectures: Recorded lectures covering topics for this week can be found in Section 404.1.6 on Page 1943 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

Reading: Lecture notes: Papers/Reports:

Examples:

Problems:

1.

2.

3.

402.4.9 Section IX, Verification and Validation

Verification and Validation: Definition, procedures, code verification, solution verification, validation experiments, model verification

Lectures: Recorded lectures covering topics for this week can be found in Section 404.1.9 on Page 1947 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

Reading: Lecture notes: 301-314; Papers/Reports:

Examples: modeling verification examples, verification for algorithms, elements. Availability of validation data.

Problems:

1.

2.

3.

402.4.10 Section X, ESSI Modeling and Simulation Synthesis

ESSI Modeling and Simulation Synthesis: Example building structure (boundary conditions, initial conditions, nonlinear interface/contact (gap/slip), nonlinear soil/rock, 1D vs 3D seismic motions development, buoyant forces at foundation level, etc.). Use of sequential and parallel computers for analysis.

Lectures: Recorded lectures covering topics for this week can be found in Section 404.1.11 on Page 1949 in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

Reading: Lecture notes: 503, 504, 509 510, 512; Papers/Reports:

Examples: Real ESSI illustrative examples

Problems:

1.

2.

3.

Chapter 403

Nonlinear ESSI for Professional Practice, A Short Course

(2017-2022-)

403.1 Short Course Delivery

Instructor: Boris Jeremić, email: `Jeremic00@gmail.com`; `Jeremic@ucdavis.edu`

Short Course Meeting: Two hour lectures, twice a week, online, using zoom; live and recorded lectures and discussions.

403.2 Objectives

The nonlinear analysis of Earthquakes, Soils, Structures, and their Interaction (ESSI) is crucial in ensuring the safety and efficiency of various structures, such as bridges, dams, buildings, tunnels, nuclear facilities, and other parts of the built environment. This analysis involves a deep understanding of statics and dynamics in soil mechanics, structural mechanics, and numerical modeling in order to accurately analyze ESSI systems' nonlinear behavior under static and dynamic loads.

This online course provides training on nonlinear, inelastic analysis for soil-structure systems, specifically focusing on earthquake soil-structure interaction (ESSI). The course is designed for practicing engineers, consultants, managers, and regulators and covers practical aspects of nonlinear analysis for ESSI systems. The main advantages and disadvantages of using nonlinear analysis for ESSI systems will also be discussed.

Who Should Attend: This short online course is designed for practicing engineers, consultants, managers, and regulators who want to reinforce and expand their knowledge of nonlinear, inelastic analysis for soil-structure systems. If you are looking to improve your understanding of the nonlinear behavior of ESSI systems and become proficient in performing nonlinear ESSI analysis, this course is will help you to achieve your goals.

What will you learn? Upon completion of this course, the participants will be able to:

1. Plan, develop, and verify different levels of sophistication ESSI models,
2. Select and calibrate elastic and elastic-plastic material models for soil, interfaces/contacts, and structural components such as beams and plates/shells,
3. Develop one component (1C), 3×1C, and 3C seismic motions from chosen earthquake records and/or stress test motions,
4. Choose numerical simulation parameters, convergence tolerances, and efficient solvers for high-performance sequential and parallel computing,
5. Identify and explain the limitations of nonlinear, inelastic finite element analysis,

6. Perform nonlinear ESSI analysis with proficiency, ranging from simplified models to high- fidelity elastic-plastic models of soil-structure systems.

Course Delivery: Online course and online office hours will use zoom platform.

Lecture Notes: `http://sokocalo.engr.ucdavis.edu/~jeremic/LectureNotes/`.

Computers: Course will rely on use of analysis system called the Real-ESSI Simulator (`http://real-essi.us`). The Real-ESSI Simulator program is available for computers running Windows (through WSL), MacOS and Linux, Please refer to `http://real-essi.us` to find out how to use Real-ESSI on local computers through Docker. The system is also available in the cloud, on Amazon Web Services (AWS) computers.

Short Course Program:

- Part I: Introduction to Nonlinear Finite Element Analysis
  - Brief overview of static and dynamic finite elements method (FEM)
  - Brief overview of linear and material nonlinear FEM
  - Nonlinear FEM analysis: stages, increments and iterations
  - Quality assurance: verification and validation (V&V) procedures
  - Introduction to the Real ESSI Simulator system, documentation, examples
- Part II: Nonlinear Material Models. Participants will learn about and use nonlinear, inelastic, elastic-plastic models:
  - Nonlinear material models for soil and rock
  - Nonlinear material models for concrete
  - Nonlinear material models for steel
  - Fiber section Nonlinear material models for concrete and steel, for beams and walls,
  - 3D soil-foundation interface/contact nonlinear material models,
- Part III: Nonlinear Finite Element Analysis. Participants will learn about and perform nonlinear finite element analysis:
  - Develop stages-increments-iterations cycles
  - Develop and perform purely incremental nonlinear analysis
  - Develop and perform iterative incremental nonlinear analysis
  - Liquefaction analysis, fully coupled, saturated, porous solid – pore fluid,
  - Energy dissipation during elasto-plastic deformation.
- Part IV: Hands on elasto-plastic example exercise. Participants will develop and use elastic-plastic models for nonlinear material analysis:
  - Calibration of nonlinear/inelastic elastic-plastic models for
    soil,
    rock,
    concrete,
    steel,
    interface/contact,
    base isolators and dissipators
  - Use of best practices for nonlinear, inelastic model development and calibration.

- Part V: Introduction to Seismic Ground Motions. Participants will learn about and develop seismic ground motions from:
    - Ground motions from given surface motions, in 1C, 3×1C, and 3C
    - Analytic, stress test ground motions, in 1C, 3×1C, and 3C
    - Ground motions from small and regional scale models

- Part VI: Ground Motions for ESSI Modeling. Participants will learn about, apply and propagate ground motions through the free field and soil-structure systems:
    - Input ground motion into FEM model
    - Application of free field and ESSI ground motions
    - Ground motions using Domain Reduction Method (DRM)
    - Control ground motion frequency content

- Part VII: Hands on ESSI Ground Motion Exercise. Participants will analyze soil-structure systems using developed ground motions:
    - Develop hierarchy of input ground motions: 1C, 3×1C and 3C.
    - Free field analysis with 1C, 3×1C and 3C motions.
    - ESSI analysis with 1C, 3×1C and 3C motions.
    - Use of best practice for ground motion ESSI modeling

- Part VIII: Hands on nonlinear ESSI examples exercise. Participants will analyze nonlinear ESSI FEM models:
    - Hierarchy of sound engineering judgement ESSI model development steps
    - Nonlinear Finite Element, elastic-plastic examples for: soils, structures, interfaces,
    - Energy dissipation in dynamic ESSI analysis: viscous, plastic, algorithmic...
    - Nonlinear finite element models for ESSI using solids, structural elements, interfaces,
    - Use of best practices for nonlinear/inelastic ESSI modeling

Chapter 404

Online Education and Training

(2019-2020-2021-)

(In collaboration with Prof. Han Yang and Dr. Hexiang Wang)

404.1 Real-ESSI Simulator Online Education and Training

This chapter was created to present online material for the theory for modeling and simulation of earthquakes, soils, structures and their interaction, as well practical examples using the Real-ESSI Simulator, `http://real-essi.us/`.

It is worth noting that some early recorded material for use of the Real-ESSI Simulator on Amazon Web Services was created in 2019, however, majority of presented, recorded material was created during Corona-Virus (COVID-19) pandemic and quarantine from March through June of 2020, in Zürich Switzerland, where Boris Jeremić was locked-up, and in Davis, California, where Han Yang and Hexiang Wang were locked-up... Internet worked very good across the Atlantic ocean, zoom.us worked really well as well. Development of online educational material continued with all three contributors now in Davis, California during Summer and Fall 2020, Winter and Spring 2021, still during partial/full lock-down, shelter in place, still using zoom.us, and still keeping physical distance, wearing face masks, etc.

In addition to organizing slides and vide lectures through this document, a youtube video channel for the Real-ESSI is available here: Real-ESSI youtube channel.

It is hoped that this material will be helpful to students and engineers that work in the area of modeling and simulation of earthquakes, soils, structures and their interaction.

404.1.1 Modeling and Simulations for ESSI

The following recorded lectures modeling and simulation approaches for Earthquakes, Soils, Structures and their Interaction are available:

404.1.1.1 Introduction to Modeling and Simulation

PDF slides, and a direct MP4 recording and/or alternatively YouTube video

404.1.1.2 Introduction to Modeling Simplifications, Epistemic Uncertainty

PDF slides, and a direct MP4 recording and/or alternatively YouTube video

404.1.1.3 Introduction to Parametric, Aleatory Uncertainty

PDF slides, and a direct MP4 recording and/or alternatively YouTube video

404.1.2 Real-ESSI Simulator Modeling and Simulation System

The following recorded lectures about the Real-ESSI Simulator modeling and simulation system are available:

1. The Real-ESSI Simulator, Introduction:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
2. The Real-ESSI Simulator, Modeling Features:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
3. The Real-ESSI Simulator, Domain Specific Language:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
4. The Real-ESSI Simulator, Model Development:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
5. The Real-ESSI Simulator, Results Post Processing:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
6. The Real-ESSI Simulator, Quick Analysis Startup Guide, how to run simple example models:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video

404.1.3 Finite Element Method

The following recorded lectures on the finite element method are available:

404.1.3.1 Backgound

1. Introduction to the Finite Element Method (FEM):
   PDF slides, MP4 recording
2. Derivation of FEM equations of motions for single phase, dry material:
   PDF slides, MP4 recording
3. Derivation of FEM equations of motions for coupled, two phase, fully and partially saturated material, u-p-U formulation:
   PDF slides, MP4 recording

404.1.3.2 Nonlinear, Inelastic FEM

4. Nonlinear, Inelastic FEM, residual equations:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
5. Solution of nonlinear, inelastic FEM equations:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video

404.1.3.3 Dynamic FEM

6. Dynamic FEM equations:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
7. Time marching algorithms for dynamic FEM equations:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video

404.1.4 Deterministic Elasto-Plasticity

404.1.4.1 Introduction to Modeling and Simulation

PDF slides, and a direct MP4 recording and/or alternatively YouTube video

The following recorded lectures on deterministic elasto-plasticity are available:

404.1.4.2 Theory Backgound

1. Micromechanical origins of elasto-plasticity:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
2. Introduction to the incremental theory of elasto-plasticity:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
3. Explicit solution to the constitutive elastic-plastic problem:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
4. Implicit solution to the constitutive elastic-plastic problem:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video

404.1.4.3 Elastic-Plastic Material Model Choices

5. Choice of elastic-plastic material models for soils and interfaces/contacts/joints:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
6. Choice of elastic-plastic material models for structural elements, beams and walls/plates/shells:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video

404.1.4.4 Calibrating Elastic-Plastic Material Models

7. Calibration of elastic-plastic material models for sand:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
8. Calibration of elastic-plastic material models for clay:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
9. Calibration of elastic-plastic material models for interfaces/contacts/joints:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video

10. Calibration of elastic-plastic material models for concrete, in reinforced beams and walls/plates/shells: PDF slides, and a direct MP4 recording and/or alternatively YouTube video

11. Calibration of elastic-plastic material models for steel, in reinforced beams and walls/beams/shells: PDF slides, and a direct MP4 recording and/or alternatively YouTube video

404.1.5 Seismic Energy Dissipation

The following recorded lectures on energy dissipation are available:

404.1.5.1 Theory Background

1. Energy dissipation introduction:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
2. Energy dissipation in solids:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
3. Energy dissipation in fiber beams:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
4. Energy dissipation in interfaces/joints/contacts:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
5. Energy dissipation due to viscous effects:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
6. Energy dissipation due to time integration, algorithmic, numerical effects:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video

404.1.5.2 Illustrative Examples

7. Energy dissipation examples:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video

404.1.6 Probabilistic Elasto-Plasticity and Stochastic Elastic-Plastic Finite Element Method

404.1.6.1 Theory Background

1. Introduction to the Polynomial Chaos (PC) expansion:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
2. Introduction to the Karhunen-Loève (KL) expansion:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
3. Introduction to the Stochastic Elastic-Plastic Finite Element Method (SEPFEM)
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
4. Introduction to Sensitivity Analysis
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video

404.1.6.2 Choice and Calibration of Probabilistic Material Models and Probabilistic Seismic Loads

5. Choice, analysis and calibration of probabilistic elastic material parameters:
   PDF slides, and a direct MP4 recording and/or alternatively; YouTube video
6. Choice, analysis and calibration of probabilistic elastic-plastic, nonlinear, inelastic material parameters:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
7. Choice, analysis and calibration of probabilistic seismic motions:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video

404.1.6.3 Simple Probabilistic Examples

8. SEPFEM, Two Examples:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
9. SEPFEM, Seismic Risk Analysis Example:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video

404.1.6.4 Probabilistic Wave Propagation Examples

10. Analysis of one component (1C) seismic wave propagation with uncertain motions and uncertain elastic material parameters:

PDF slides, and a direct MP4 recording and/or alternatively YouTube video

11. Analysis of one component (1C) seismic wave propagation with uncertain motions and uncertain elastic-plastic, nonlinear, inelastic material parameters:
    PDF slides, and a direct MP4 recording and/or alternatively YouTube video
12. Sensitivity analysis for uncertain motions and uncertain elastic material parameters:
    PDF slides, and a direct MP4 recording and/or alternatively YouTube video
13. Sensitivity analysis for uncertain motions and uncertain elastic-plastic, nonlinear, inelastic material parameters:
    PDF slides, and a direct MP4 recording and/or alternatively YouTube video

404.1.7 Seismic Motions

The following recorded lectures on seismic motions are available:

1. On earthquakes:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
2. On six component (6C) seismic motions:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
3. On the Domain Reduction Method (DRM):
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
4. Development of DRM motions from surface records, 1C, 2×1C, and 3×1C:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
5. Development of DRM motions from inclined, 3C seismic waves:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video

404.1.8 Earthquake Soil Structure Interaction

The following recorded lectures on Earthquake Soil Structure Interaction (ESSI) are available:

1. On ESSI:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video

404.1.9 Verification and Validation

Basic theory of Verification and Validation (V&V), as well as V&V examples for the Real ESSI Simulator are shown in recorded lectures below:

1. Verification and Validation introduction:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
2. Real ESSI Simulator Verification and Validation system:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
3. Real ESSI Simulator Verification examples:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
4. Real ESSI Simulator Validation examples:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video

404.1.10 High Performance Computing

High Performance Computing (HPC) is helping with analysis of sophisticated models efficiently on sequential and parallel computers.

404.1.10.1 HPC Introduction

1. HPC, an Introduction:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
2. Fine grainded HPC:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
3. Coarse grainded HPC, a Distributed Memory Parallel (DMP) Introduction:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video

404.1.10.2 HPC and Real-ESSI

1. Real-ESSI Simulator HPC Approach:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video

404.1.10.3 Real-ESSI Parallel Computing Examples

1. Real-ESSI Simulator Parallel Examples:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video

404.1.11 Real-ESSI Simulator Examples

Select Real-ESSI examples are shown in recorded lectures below:

1. How to run already installed Real-ESSI program on a simple example:
   MP4 recording and/or alternatively YouTube video
2. Running Real-ESSI program for a frame model:
   MP4 recording and/or alternatively YouTube video
3. Running Real-ESSI program for a elastic-plastic solids model:
   MP4 recording and/or alternatively YouTube video
4. Running Real-ESSI program for a solids, beams and shells model:
   MP4 recording and/or alternatively YouTube video
5. Post-processing Real-ESSI results using Paraview for frame model:
   MP4 recording and/or alternatively YouTube video
6. Post-processing Real-ESSI results using Paraview for a solids, beams and shells model:
   MP4 recording and/or alternatively YouTube video
7. Developing a DRM SSI model, solids and beams:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video
8. Running a DRM SSI model, solids and beams:
   PDF slides, and a direct MP4 recording and/or alternatively YouTube video

Chapter 405

Constitutive, Material Behaviour Examples

(2016-2017-2019-2023-)

(In collaboration with Dr. Yuan Feng and Dr. Han Yang)

405.1 Chapter Summary and Highlights

In this Chapter constitutive behavior of elastic-plastic material is illustrated through a number of examples.

All the examples described here, and many more, organized in sub-directories, for constitutive behavior, static and dynamic behavior can be directly downloaded from a repository at: `http://sokocalo.engr.ucdavis.edu/~jeremic/lecture_notes_online_material/Real-ESSI_Examples/education_examples`. These examples can then be tried, analyzed using Real-ESSI Simulator that is available on Amazon Web Services (AWS) computers around the word. Login to AWS market place and search for Real-ESSI...

405.2 Elastic Solid Constitutive Examples

405.2.1 Linear Elastic Constitutive Examples

405.2.1.1 Pure Shear, Monotonic Loading

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Material properties in Real-ESSI input:

```
model name "test";
add material # 1 type linear_elastic_isotropic_3d
   mass_density = 2E3 * kg/m^3
   elastic_modulus = 2E7 * Pa
   poisson_ratio= 0.0 ;
simulate constitutive testing strain control pure shear monotonic loading use ↩
    material # 1
   confinement_strain = 0.001
   strain_increment_size = 0.0001
   number_of_increment = 100;
bye;
```

Material Response:

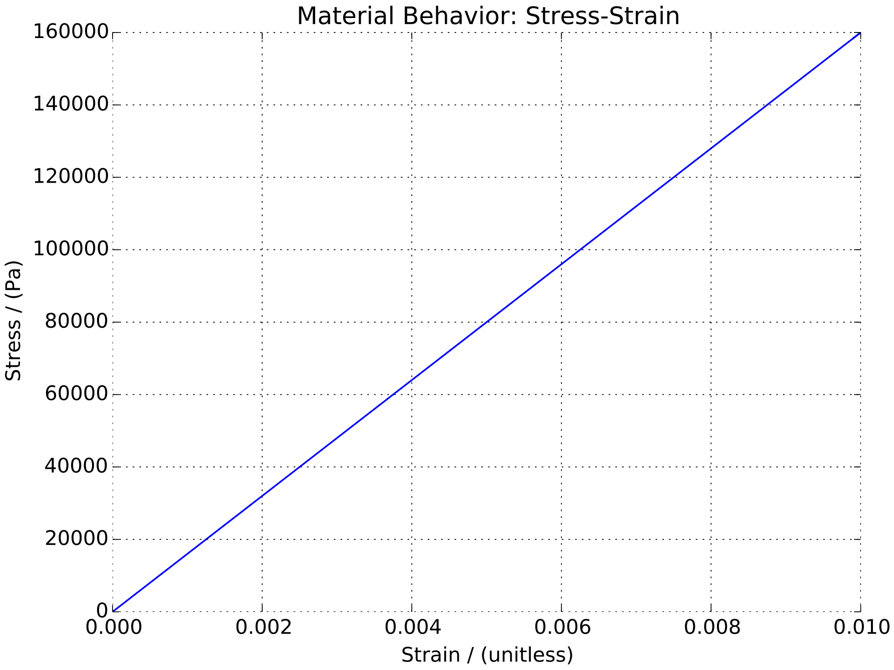


Figure 405.1: Linear Elastic Pure Shear Monotonic Loading

405.2.1.2 Pure Shear, Cyclic Loading

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Material properties in Real-ESSI input:

```
model name "test";
add material # 1 type linear_elastic_isotropic_3d
   mass_density = 2E3 * kg/m^3
   elastic_modulus = 2E7 * Pa
   poisson_ratio= 0.25 ;
simulate constitutive testing strain control pure shear cyclic loading use ↩
    material # 1
   confinement_strain = 0.001
   strain_increment_size = 0.0001
   maximum_strain = 0.01
   number_of_cycles = 1;
bye;
```

Material Response:

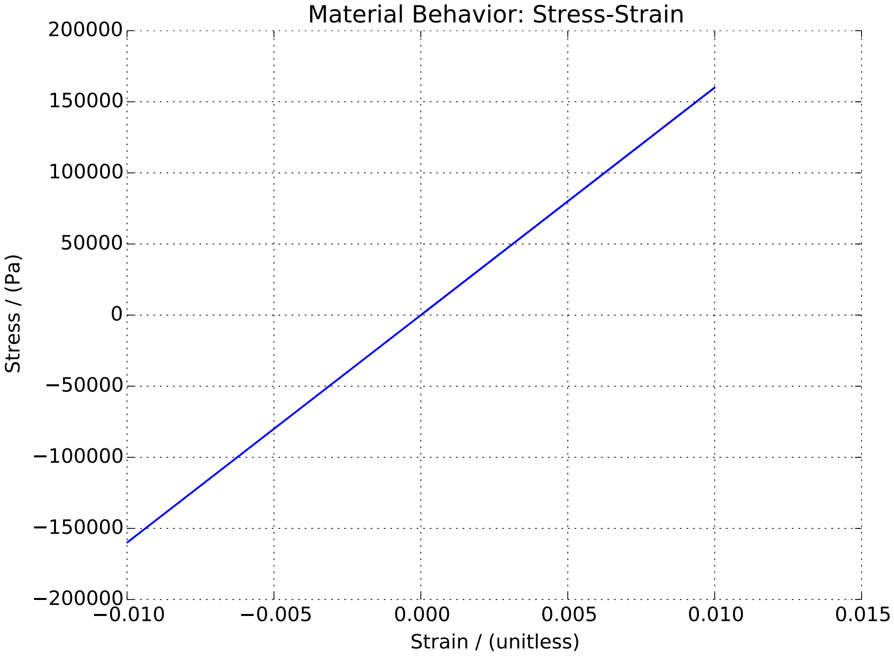


Figure 405.2: Linear Elastic Pure Shear Cyclic Loading.

405.2.1.3 Uniaxial Strain, Monotonic Loading

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Material properties in Real-ESSI input:

```
model name "test";
add material # 1 type linear_elastic_isotropic_3d
   mass_density = 2E3 * kg/m^3
   elastic_modulus = 2E7 * Pa
   poisson_ratio= 0.0 ;
simulate constitutive testing strain control uniaxial monotonic loading use ↩
    material # 1
   confinement_strain = 0.001
   strain_increment_size = 0.0001
   number_of_increment = 100;
bye;
```

Material Response:

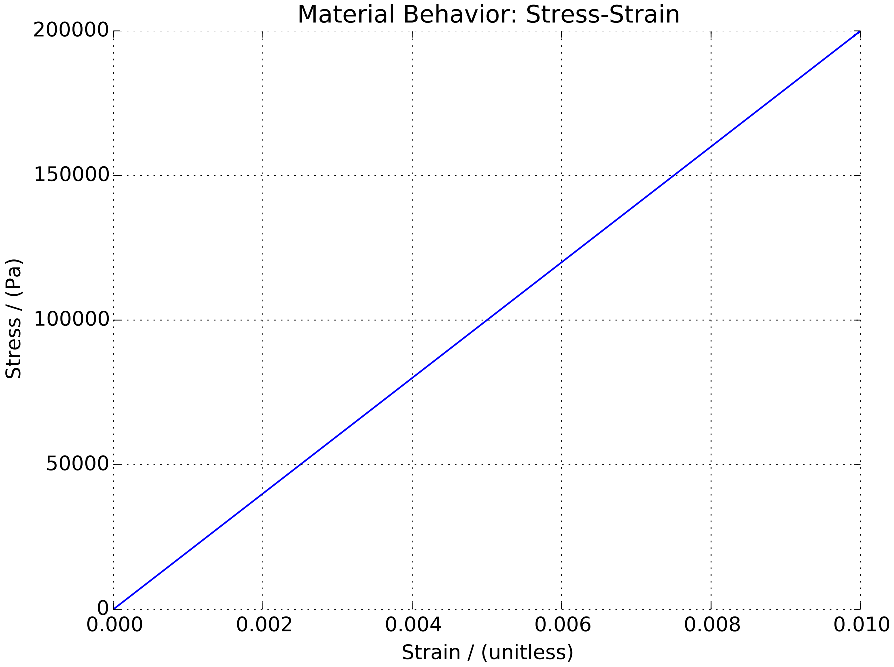


Figure 405.3: Linear Elastic Uniaxial Monotonic Loading

405.2.1.4 Uniaxial Strain, Cyclic Loading

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Material properties in Real-ESSI input:

```
model name "test";
add material # 1 type linear_elastic_isotropic_3d
   mass_density = 2E3 * kg/m^3
   elastic_modulus = 2E7 * Pa
   poisson_ratio= 0.25 ;
simulate constitutive testing strain control pure shear cyclic loading use ↩
    material # 1
   confinement_strain = 0.001
   strain_increment_size = 0.0001
   maximum_strain = 0.01
   number_of_cycles = 1;
bye;
```

Material Response:

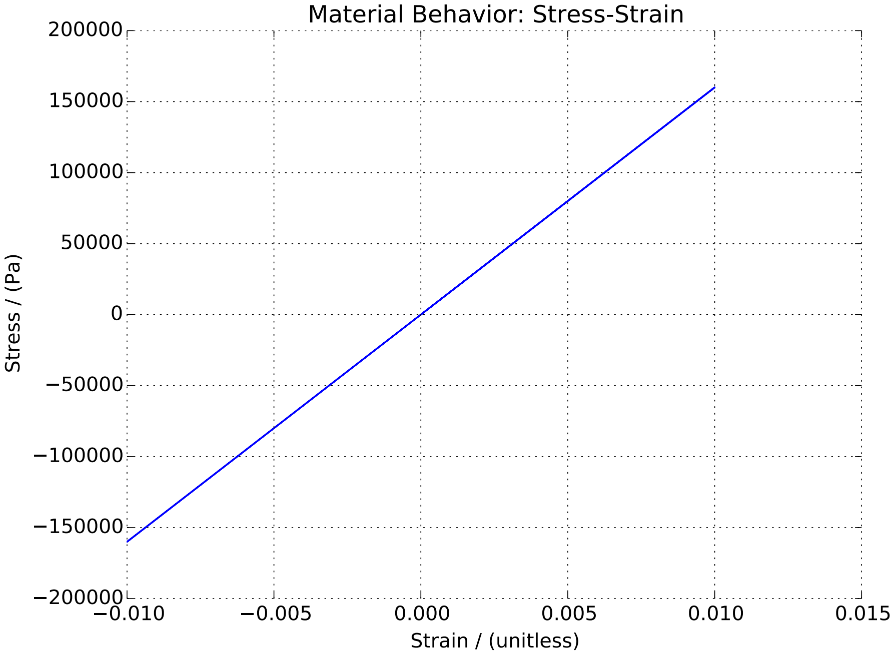


Figure 405.4: Linear Elastic Uniaxial Cyclic Loading

405.2.2 Nonlinear Elastic Constitutive Examples

405.2.2.1 Triaxial Uniform Pressure, Monotonic Loading

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

The Duncan-Chang nonlinear elastic materials:

$$E = K p_a \left(\frac{\sigma_3}{p_a}\right)^n \tag{405.1}$$

where K and n are material constants. And pressure p_a is atmospheric pressure. And stress σ_3 is the minor principal stress.

Material properties in Real-ESSI input:

```
model name "test";
add material # 1 type Duncan_Chang_nonlinear_elastic_isotropic_3d_LT
    mass_density = 2E3 * kg/m^3
    initial_elastic_modulus = 3E5 * Pa
    poisson_ratio= 0.15
    DuncanChang_K = 1E3
    DuncanChang_pa = 1E5 * Pa
    DuncanChang_n = 0.5 ;
simulate constitutive testing strain control triaxial confinement loading use ↩
    material # 1
    strain_increment_size = 0.00001
    maximum_strain = 0.01
    number_of_increment = 2000;
bye;
```

Material Response:

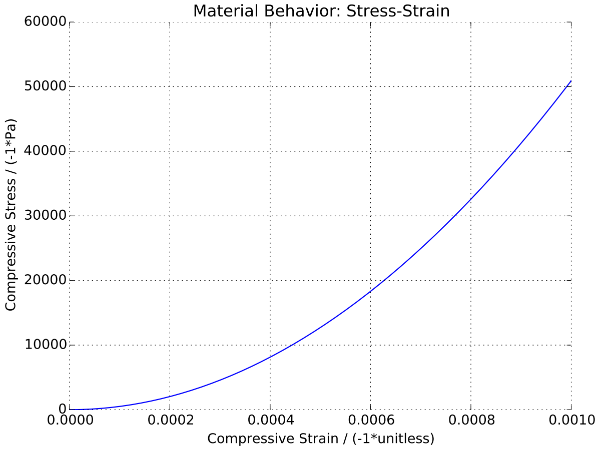


Figure 405.5: Results of Duncan-Chang Nonlinear Elastic Monotonic Loading

405.3 Elastic Plastic Solid Constitutive Examples

405.3.1 Elastic Perfectly Plastic Constitutive Examples

405.3.1.1 Pure Shear

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Material properties in Real-ESSI input:

```
model name "test";
add material # 1 type VonMises
   mass_density = 2E3*kg/m^3
   elastic_modulus = 2E7 * Pa
   poisson_ratio=0.25
   von_mises_radius = 1E5*Pa
   kinematic_hardening_rate = 0.0 *Pa
   isotropic_hardening_rate = 0.0*Pa ;
define NDMaterial constitutive integration algorithm Backward_Euler
   yield_function_relative_tolerance = 1E-2
   stress_relative_tolerance = 1E-3
   maximum_iterations = 30;
simulate constitutive testing strain control pure shear cyclic loading use ↩
    material # 1
   confinement_strain = 0.001
   strain_increment_size = 0.0001
   maximum_strain = 0.01
   number_of_cycles = 1;
bye;
```

Material Response:

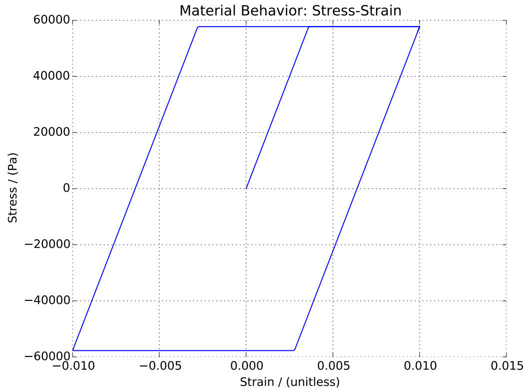


Figure 405.6: Perfectly Plastic Pure Shear Cyclic Loading.

405.3.1.2 Uniaxial Strain

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Material properties in Real-ESSI input:

```
model name "test";
add material # 1 type VonMises
   mass_density = 2E3*kg/m^3
   elastic_modulus = 2E7 * Pa
   poisson_ratio=0.25
   von_mises_radius = 1E5*Pa
   kinematic_hardening_rate = 0.0 *Pa
   isotropic_hardening_rate = 0.0*Pa ;
define NDMaterial constitutive integration algorithm Backward_Euler
   yield_function_relative_tolerance = 1E-2
   stress_relative_tolerance = 1E-3
   maximum_iterations = 30;
simulate constitutive testing strain control uniaxial cyclic loading use material # 1
   confinement_strain = 0.001
   strain_increment_size = 0.0001
   maximum_strain = 0.01
   number_of_cycles = 1;
bye;
```

Material Response:

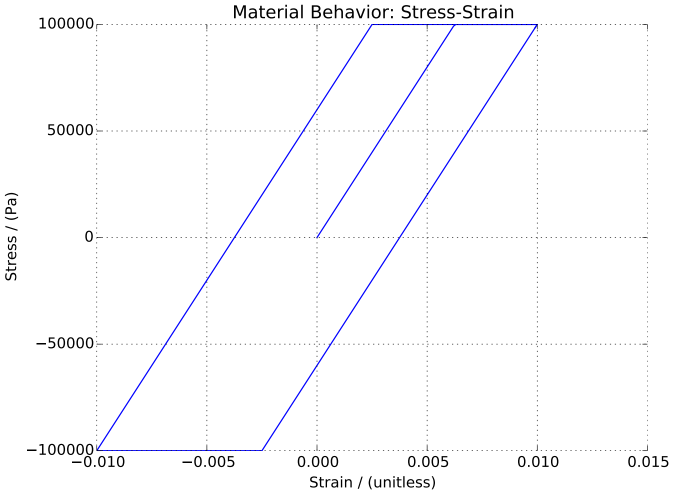


Figure 405.7: Perfectly Plastic Uniaxial Cyclic Loading

405.3.2 Elastic Plastic, Isotropic Hardening, Constitutive Examples

405.3.2.1 Pure Shear, Monotonic Loading

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Material properties in Real-ESSI input:

```
model name "test";
add material # 1 type VonMises
   mass_density = 2E3*kg/m^3
   elastic_modulus = 2E7 * Pa
   poisson_ratio=0.25
   von_mises_radius = 1E5*Pa
   kinematic_hardening_rate = 0.0*Pa
   isotropic_hardening_rate = 2E6 *Pa ;
define NDMaterial constitutive integration algorithm Backward_Euler
   yield_function_relative_tolerance = 1E-2
   stress_relative_tolerance = 1E-3
   maximum_iterations = 30;
simulate constitutive testing strain control pure shear monotonic loading use ↩
    material # 1
   confinement_strain = 0.001
   strain_increment_size = 0.0001
   number_of_increment = 99;
bye;
```

Material Response:

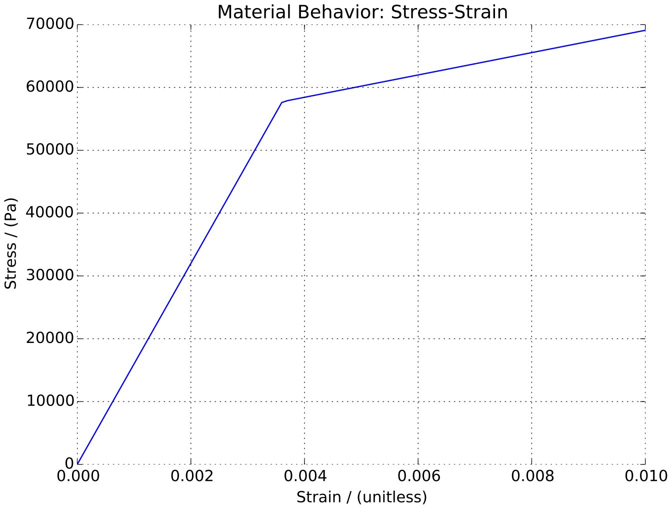


Figure 405.8: Isotropic Hardening Pure Shear Monotonic Loading

405.3.2.2 Pure Shear, Cyclic Loading

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Material properties in Real-ESSI input:

```
model name "test";
add material # 1 type VonMises
   mass_density = 2E3*kg/m^3
   elastic_modulus = 2E7 * Pa
   poisson_ratio=0.25
   von_mises_radius = 1E5*Pa
   kinematic_hardening_rate = 0.0*Pa
   isotropic_hardening_rate = 2E6 *Pa ;
define NDMaterial constitutive integration algorithm Backward_Euler
   yield_function_relative_tolerance = 1E-2
   stress_relative_tolerance = 1E-3
   maximum_iterations = 30;
simulate constitutive testing strain control pure shear cyclic loading use ↩
    material # 1
   confinement_strain = 0.001
   strain_increment_size = 0.0001
   maximum_strain = 0.01
   number_of_cycles = 1;
bye;
```

Material Response:

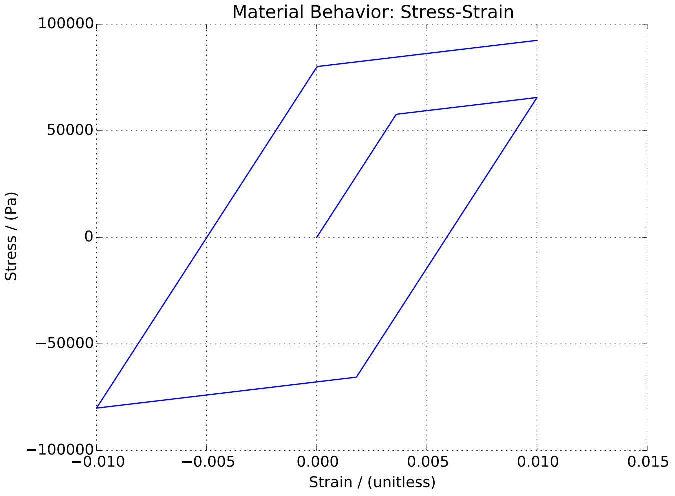


Figure 405.9: Isotropic Hardening Pure Shear Cyclic Loading.

405.3.2.3 Uniaxial Strain, Monotonic Loading

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Material properties in Real-ESSI input:

```
model name "test";
add material # 1 type VonMises
   mass_density = 2E3*kg/m^3
   elastic_modulus = 2E7 * Pa
   poisson_ratio=0.25
   von_mises_radius = 5E4*Pa
   kinematic_hardening_rate = 0.0*Pa
   isotropic_hardening_rate = 2E6 *Pa ;
define NDMaterial constitutive integration algorithm Backward_Euler
   yield_function_relative_tolerance = 1E-2
   stress_relative_tolerance = 1E-3
   maximum_iterations = 30;
simulate constitutive testing strain control uniaxial monotonic loading use ↩
    material # 1
   confinement_strain = 0.001
   strain_increment_size = 0.0001
   number_of_increment = 99;
bye;
```

Material Response:

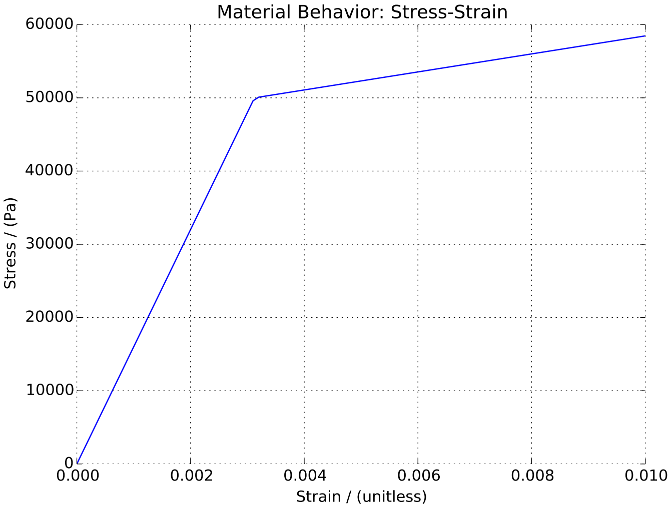


Figure 405.10: Isotropic Hardening Uniaxial Monotonic Loading

405.3.2.4 Uniaxial Strain, Cyclic Loading

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Material properties in Real-ESSI input:

```
model name "test";
add material # 1 type VonMises
   mass_density = 2E3*kg/m^3
   elastic_modulus = 2E7 * Pa
   poisson_ratio=0.25
   von_mises_radius = 5E4*Pa
   kinematic_hardening_rate = 0.0*Pa
   isotropic_hardening_rate = 2E6 *Pa ;
define NDMaterial constitutive integration algorithm Backward_Euler
   yield_function_relative_tolerance = 1E-2
   stress_relative_tolerance = 1E-3
   maximum_iterations = 30;
simulate constitutive testing strain control uniaxial cyclic loading use ↩
    material # 1
   confinement_strain = 0.001
   strain_increment_size = 0.0001
   maximum_strain = 0.01
   number_of_cycles = 1;
bye;
```

Material Response:

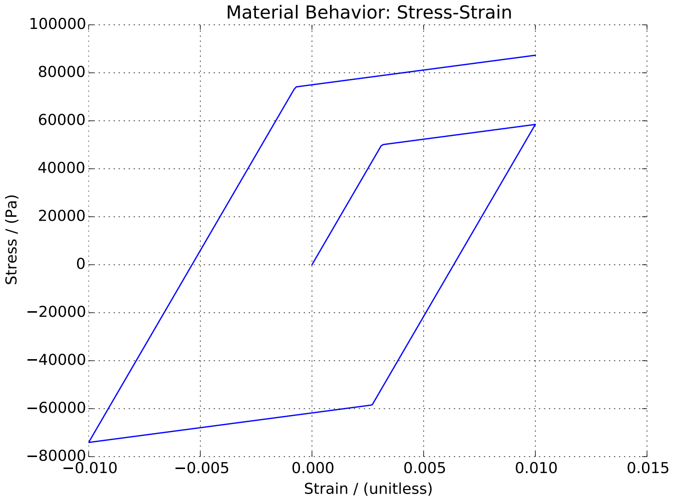


Figure 405.11: Isotropic Hardening Uniaxial Cyclic Loading

405.3.3 Elastic Plastic, Kinematic Hardening, Constitutive Examples

405.3.3.1 Pure Shear, Monotonic Loading

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Material properties in Real-ESSI input:

```
model name "test";
add material # 1 type VonMises
   mass_density = 2E3*kg/m^3
   elastic_modulus = 2E7 * Pa
   poisson_ratio=0.25
   von_mises_radius = 1E5*Pa
   kinematic_hardening_rate = 2E6*Pa
   isotropic_hardening_rate = 0.0*Pa ;
define NDMaterial constitutive integration algorithm Backward_Euler
   yield_function_relative_tolerance = 1E-2
   stress_relative_tolerance = 1E-3
   maximum_iterations = 30;
simulate constitutive testing strain control pure shear monotonic loading use ↩
    material # 1
   confinement_strain = 0.001
   strain_increment_size = 0.0001
   number_of_increment = 99;
bye;
```

Material Response:

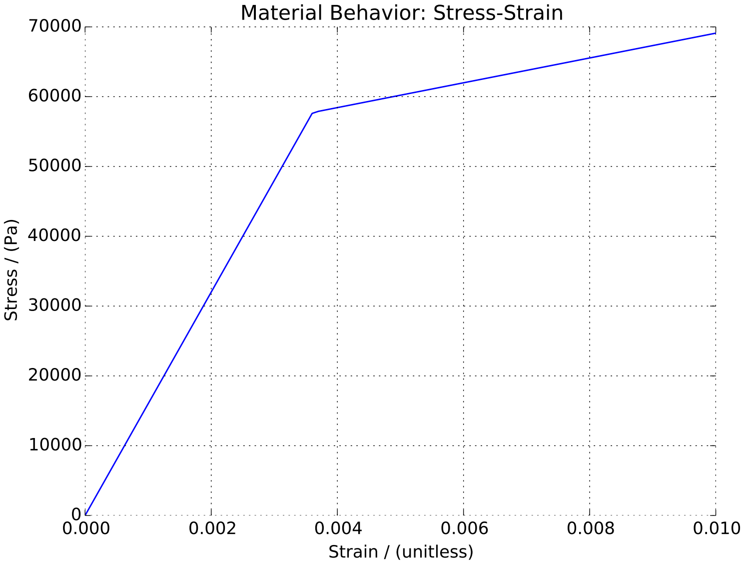


Figure 405.12: Kinematic Hardening Monotonic Cyclic Loading

405.3.3.2 Pure Shear, Cyclic Loading

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Material properties in Real-ESSI input:

```
model name "test";
add material # 1 type VonMises
   mass_density = 2E3*kg/m^3
   elastic_modulus = 2E7 * Pa
   poisson_ratio=0.25
   von_mises_radius = 1E5*Pa
   kinematic_hardening_rate = 2E6*Pa
   isotropic_hardening_rate = 0.0*Pa ;
define NDMaterial constitutive integration algorithm Backward_Euler
   yield_function_relative_tolerance = 1E-2
   stress_relative_tolerance = 1E-3
   maximum_iterations = 30;
simulate constitutive testing strain control pure shear cyclic loading use ↩
    material # 1
   confinement_strain = 0.001
   strain_increment_size = 0.0001
   maximum_strain = 0.01
   number_of_cycles = 1;
bye;
```

Material Response:

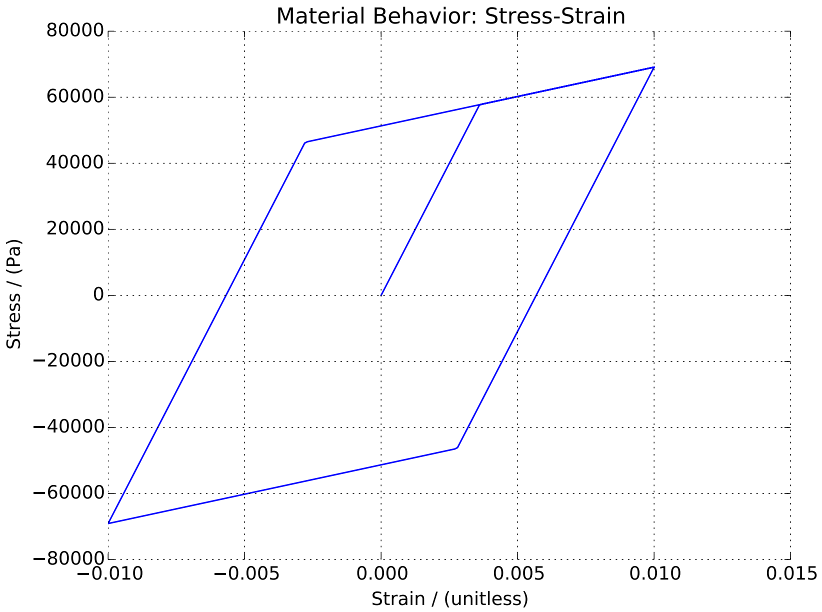


Figure 405.13: Kinematic Hardening Pure Shear Cyclic Loading.

405.3.3.3 Uniaxial Strain, Monotonic Loading

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Material properties in Real-ESSI input:

```
model name "test";
add material # 1 type VonMises
   mass_density = 2E3*kg/m^3
   elastic_modulus = 2E7 * Pa
   poisson_ratio=0.25
   von_mises_radius = 5E4*Pa
   kinematic_hardening_rate = 2E6*Pa
   isotropic_hardening_rate = 0.0*Pa ;
define NDMaterial constitutive integration algorithm Backward_Euler
   yield_function_relative_tolerance = 1E-2
   stress_relative_tolerance = 1E-3
   maximum_iterations = 30;
simulate constitutive testing strain control uniaxial monotonic loading use ↩
    material # 1
   confinement_strain = 0.001
   strain_increment_size = 0.0001
   number_of_increment = 99;
bye;
```

Material Response:

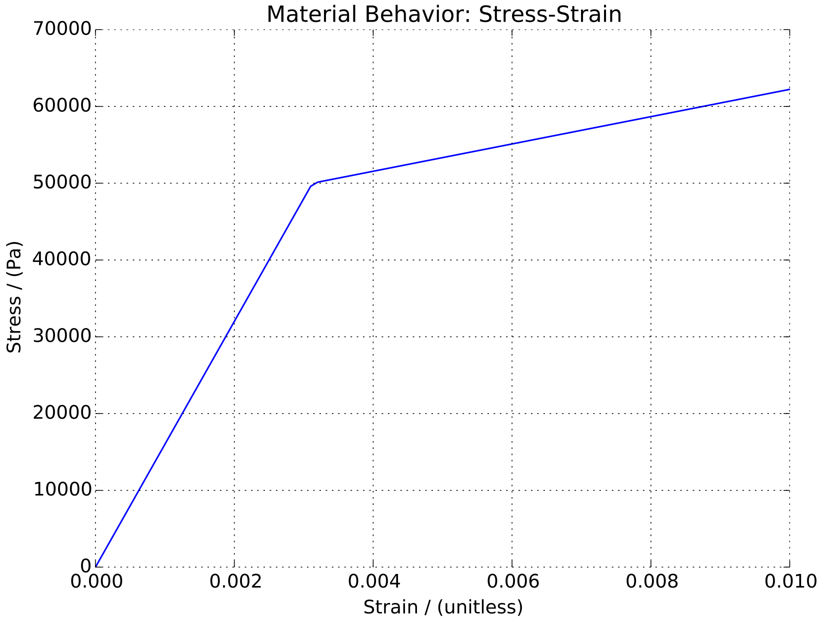


Figure 405.14: Kinematic Hardening Uniaxial Monotonic Loading

405.3.3.4 Uniaxial Strain, Cyclic Loading

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Material properties in Real-ESSI input:

```
model name "test";
add material # 1 type VonMises
   mass_density = 2E3*kg/m^3
   elastic_modulus = 2E7 * Pa
   poisson_ratio=0.25
   von_mises_radius = 5E4*Pa
   kinematic_hardening_rate = 2E6*Pa
   isotropic_hardening_rate = 0.0*Pa ;
define NDMaterial constitutive integration algorithm Backward_Euler
   yield_function_relative_tolerance = 1E-2
   stress_relative_tolerance = 1E-3
   maximum_iterations = 30;
simulate constitutive testing strain control uniaxial cyclic loading use ↩
    material # 1
   confinement_strain = 0.001
   strain_increment_size = 0.0001
   maximum_strain = 0.01
   number_of_cycles = 1;
bye;
```

Material Response:

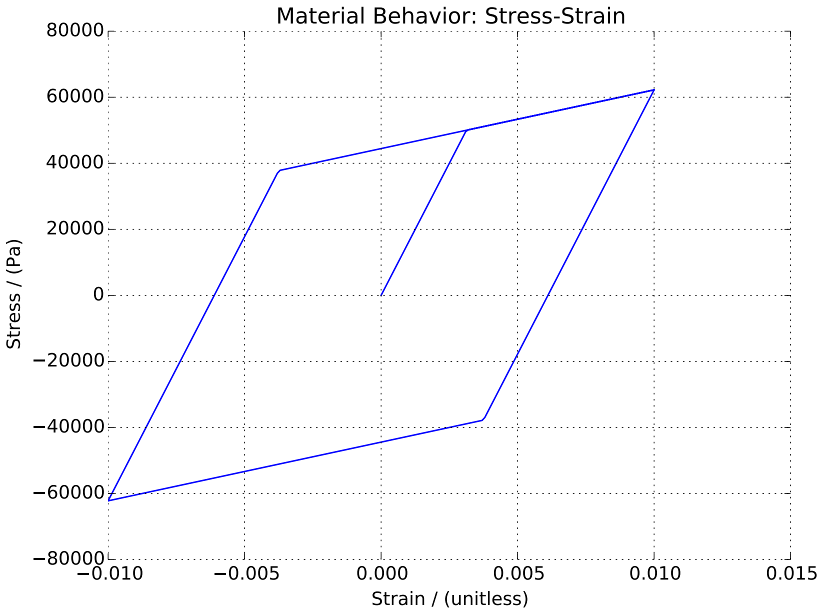


Figure 405.15: Kinematic Hardening Uniaxial Cyclic Loading

405.3.4 Elastic Plastic, Armstrong-Frederick, von-Mises, Constitutive Examples

405.3.4.1 Pure Shear, Cyclic Loading

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Material properties in Real-ESSI input:

```
model name "vmaf";
add material # 1 type vonMisesArmstrongFrederick
    mass_density = 0.0*kg/m^3
    elastic_modulus = 2E7*N/m^2
    poisson_ratio = 0.0
    von_mises_radius = 100 * Pa
    armstrong_frederick_ha = 2E7*N/m^2
    armstrong_frederick_cr = 1000
    isotropic_hardening_rate = 0*Pa ;
define NDMaterial constitutive integration algorithm Backward_Euler
   yield_function_relative_tolerance = 1E-6
   stress_relative_tolerance = 1E-6
   maximum_iterations = 30;
simulate constitutive testing strain control pure shear cyclic loading use ↩
    material # 1
   confinement_strain = 0.001
   strain_increment_size = 0.0001
   maximum_strain = 0.01
   number_of_cycles = 1;
bye;
```

Material Response:

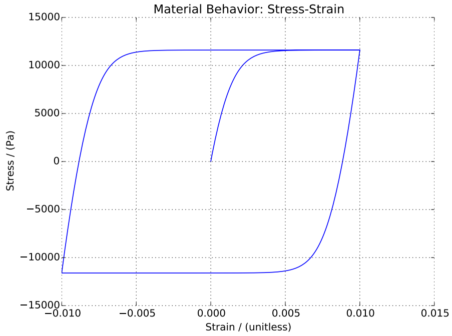


Figure 405.16: Material von-Mises Armstrong-Frederick under Pure Shear Cyclic Loading.

405.3.5 Elastic Plastic, Armstrong-Frederick, Drucker-Prager, Constitutive Examples

405.3.5.1 Pure Shear, Cyclic Loading

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Material properties in Real-ESSI input:

```
model name "test";
phi = 5;
phirad = pi*phi/180;
eta = 6*sin(phirad)/(3-sin(phirad));
add material # 1 type DruckerPragerNonAssociateArmstrongFrederick
    mass_density = 0.0*kg/m^3
    elastic_modulus = 2E7*N/m^2
    poisson_ratio = 0.0
    druckerprager_k = eta
    armstrong_frederick_ha = 2E7*N/m^2
    armstrong_frederick_cr = 100
    isotropic_hardening_rate = 0*Pa
    initial_confining_stress = 1*Pa
    plastic_flow_xi = 0.0
    plastic_flow_kd = 0.0 ;
define NDMaterial constitutive integration algorithm Backward_Euler
   yield_function_relative_tolerance = 1E-6
   stress_relative_tolerance = 1E-6
   maximum_iterations = 30;
simulate constitutive testing strain control pure shear cyclic loading use ↩
    material # 1
   confinement_strain = 0.001
   strain_increment_size = 0.0001
   maximum_strain = 0.01
   number_of_cycles = 1;
bye;
```

Material Response:

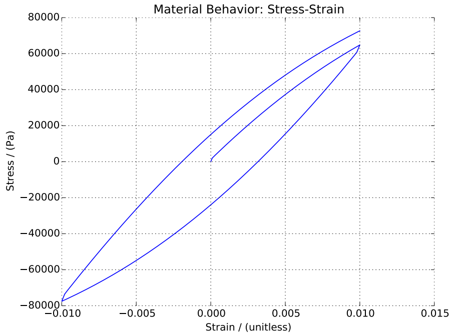


Figure 405.17: Drucker-Prager Armstrong-Frederick under Pure Shear Cyclic Loading.

405.3.6 Elastic Plastic, SaniSAND, Constitutive Examples

405.3.6.1 Bardet Constraint Examples

The compressed package of Real-ESSI input files and postprocessing scripts and results for this example is available HERE. Material Response is shown in Figure 405.18

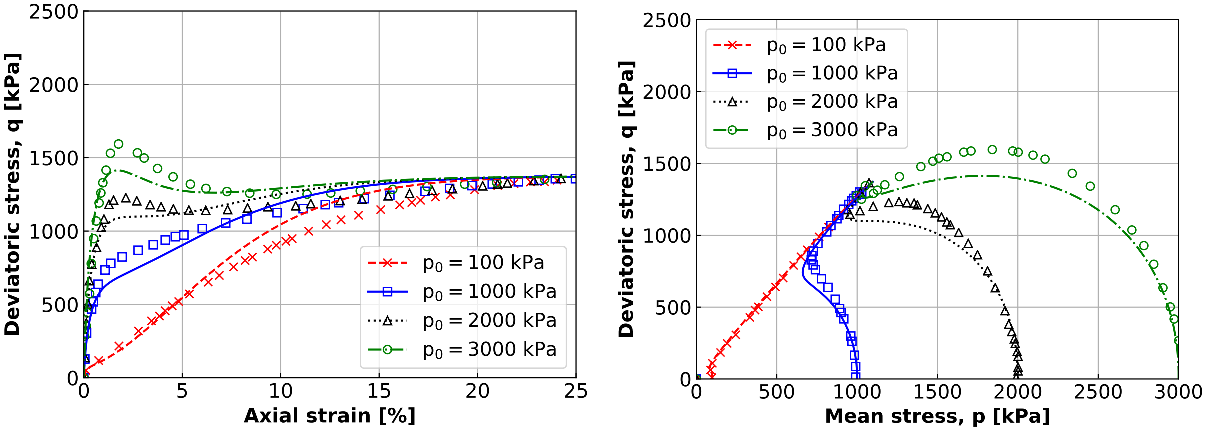


Figure 405.18: SaniSAND response.

405.4 Stiffness Reduction and Damping Curves Modeling

405.4.1 Multi-yield-surface von-Mises

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

405.4.1.1 Model description

This model illustrates the G/Gmax input to multi-yield-surface von-Mises material. This example is based on one Gauss-point with multi-yield-surface von-Mises material. The G/Gmax is converted to material modeling parameters (yield-surface size and hardening parameter) inside the DSL.

405.4.1.2 Real-ESSI input file

```
model name "test";
add material # 1 type vonMisesMultipleYieldSurfaceGoverGmax
   mass_density = 0.0*kg/m^3
   initial_shear_modulus = 3E8 * Pa
   poisson_ratio = 0.0
   total_number_of_shear_modulus = 9
   GoverGmax =
   "1,0.995,0.966,0.873,0.787,0.467,0.320,0.109,0.063"
   ShearStrainGamma =
   "0,1E-6,1E-5,5E-5,1E-4, 0.0005, 0.001, 0.005, 0.01"
   ;
define NDMaterial constitutive integration algorithm Backward_Euler
    yield_function_relative_tolerance = 1E-6
    stress_relative_tolerance = 1E-6
    maximum_iterations = 30
    ;
incr_size = 0.000001 ;
max_strain= 0.005 ;
num_of_increm = max_strain/incr_size -1 ;
simulate constitutive testing strain control pure shear use material # 1
   confinement_strain = 0.0
   strain_increment_size = incr_size
   maximum_strain = max_strain
   number_of_increment = num_of_increm;
bye;
```

Material Response at Gauss Point:

Computed G/Gmax curve exactly matches the one used for input at control points.

The difference in G/Gmax between control points can be reduced by using more than just 9 control points as in this example.

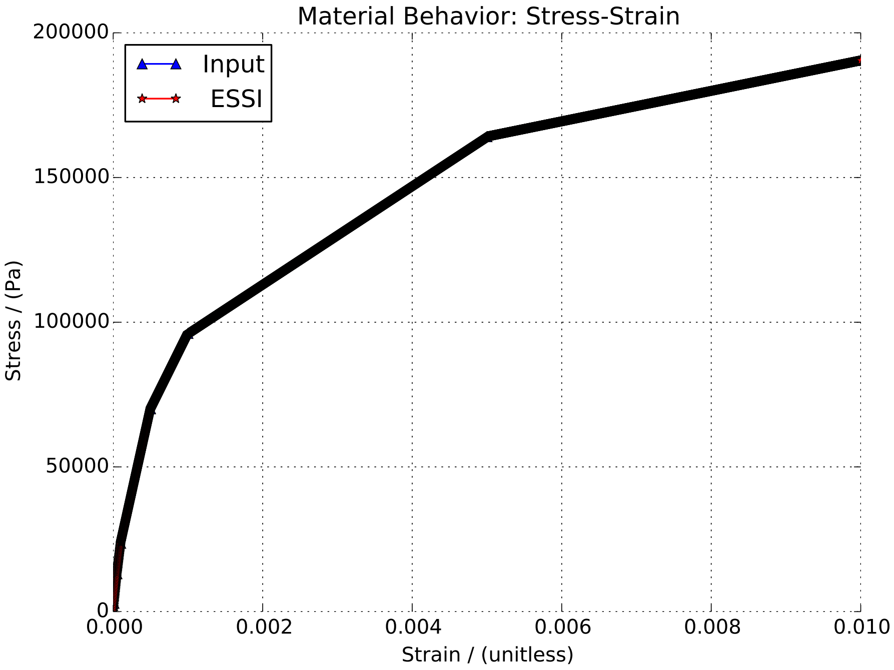


Figure 405.19: Stress-Strain Relationship

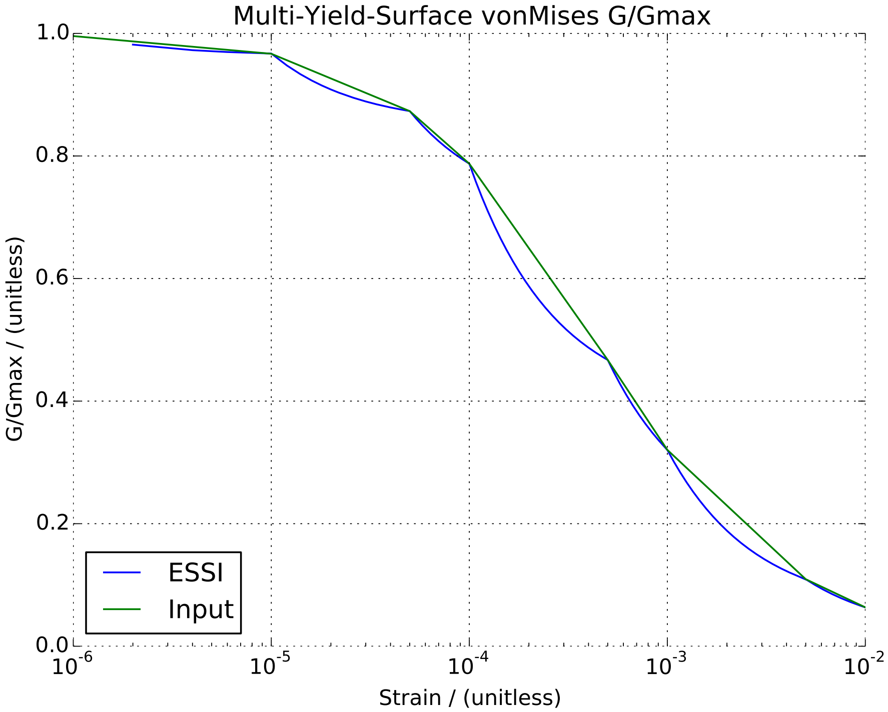


Figure 405.20: The G/Gmax results.

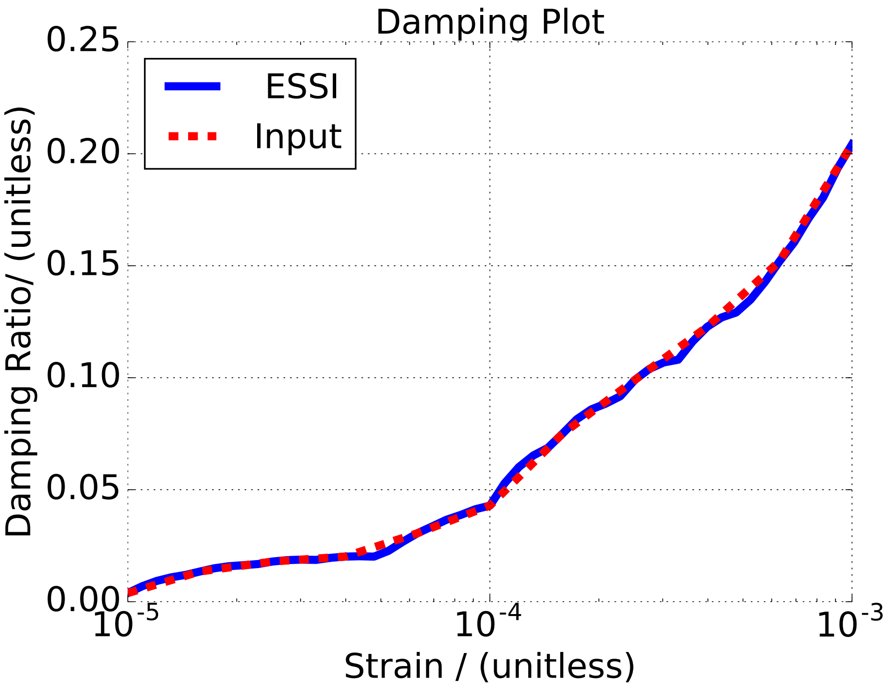


Figure 405.21: Damping Ratio Plot

405.4.2 Multi-yield-surface Drucker-Prager

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

405.4.2.1 Problem description

This model illustrates the G/Gmax input to multi-yield-surface Drucker-Prager material. Purely deviatoric plastic flow is used in this material, which means that the parameter dilation_scale is set to zero. If user wants to model change of volume (dilation or compression) for this material, then G/Gmax curve need to be iterated upon manually by changing yield surface size directly, which is done using different DruckerPragerMultipleYieldSurface command. This example is based on one Gauss-point which use multi-yield-surface Drucker-Prager material. The G/Gmax is converted to the yield-surface size and hardening parameter inside the DSL.

405.4.2.2 Real-ESSI input file:

```
model name "test";

add material # 1 type DruckerPragerMultipleYieldSurfaceGoverGmax
   mass_density = 0.0*kg/m^3
   initial_shear_modulus = 3E8 * Pa
   poisson_ratio = 0.0
   initial_confining_stress = 1E5 * Pa
   reference_pressure = 1E5 * Pa
   pressure_exponential_n = 0.5
   cohesion = 0. * Pa
   dilation_angle_eta =1.0
   dilation_scale = 0.0
   total_number_of_shear_modulus = 9
   GoverGmax =
   "1,0.995,0.966,0.873,0.787,0.467,0.320,0.109,0.063"
   ShearStrainGamma =
   "0,1E-6,1E-5,5E-5,1E-4, 0.0005, 0.001, 0.005, 0.01"
   ;
define NDMaterial constitutive integration algorithm Backward_Euler
   yield_function_relative_tolerance = 1E-6
   stress_relative_tolerance = 1E-6
   maximum_iterations = 30;
simulate constitutive testing strain control pure shear use material # 1
   confinement_strain = 0.0
   strain_increment_size = 0.000001
   maximum_strain = 0.005
   number_of_increment = 0.005 / 0.000001 -1 ;
bye;
```

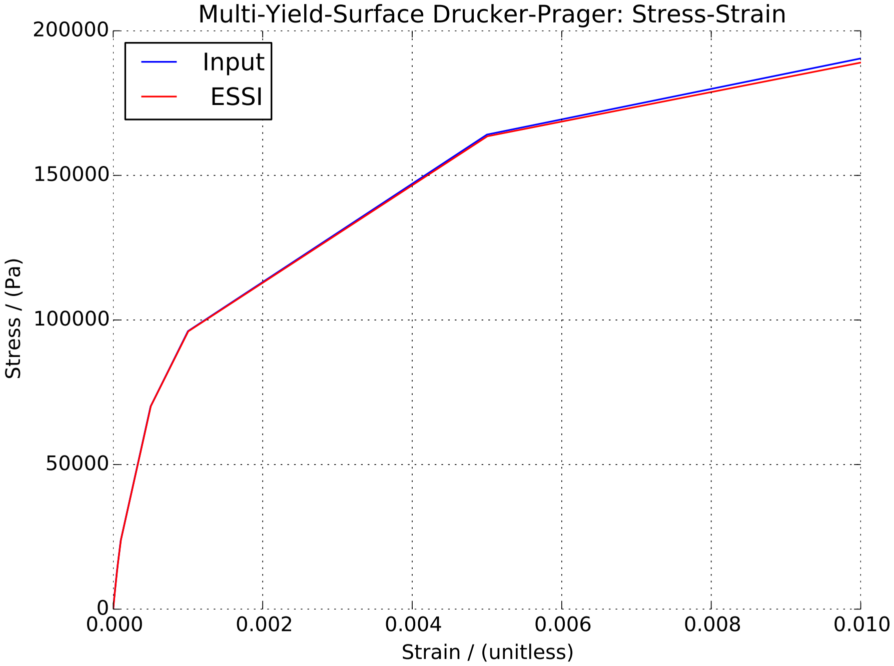


Figure 405.22: Nested-Yield-Surface Drucker-Prager Stress-Strain Relationship

Inside the DSL, the yield surface radius is calculated as $\sqrt{3}\sigma_y$, where σ_y is the yield stress of the corresponding yield surface. Then, the radius is divided by the confinement to obtain the slope (opening angle).

The hardening parameter is calculated as

$$\frac{1}{H_i'} = \frac{1}{H_i} - \frac{1}{2G} \tag{405.2}$$

where H_i' is the current hardening parameter corresponding to yield surface i. H_i is the current tangent shear modulus to surface i, namely, $H_i = 2(\tau_{i+1} - \tau_i)/(\gamma_{i+1} - \gamma_i)$. And G is the initial shear modulus.

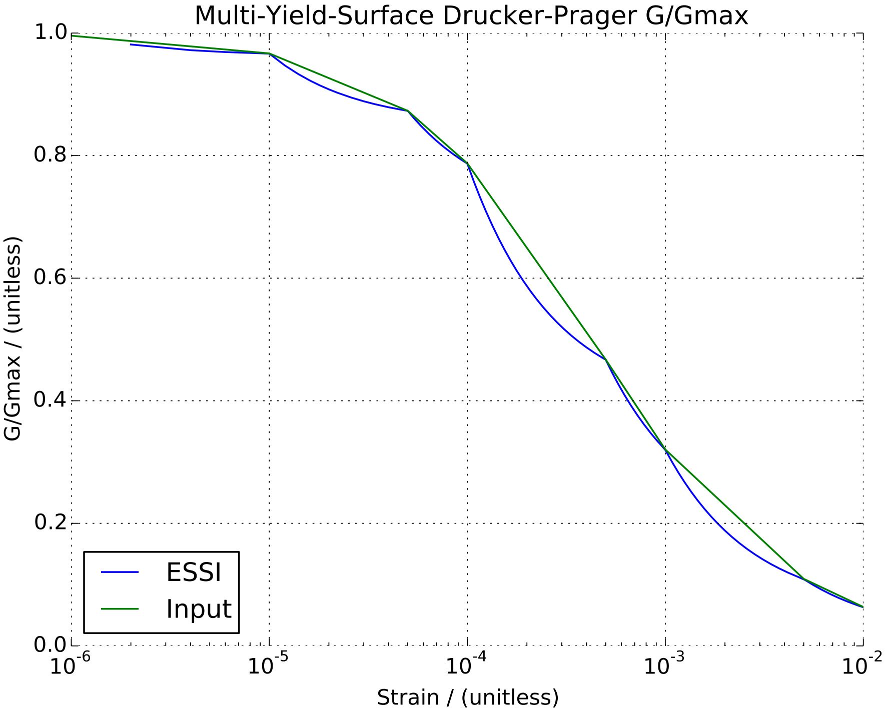


Figure 405.23: Nested-Yield-Surface Drucker-Prager G/Gmax results

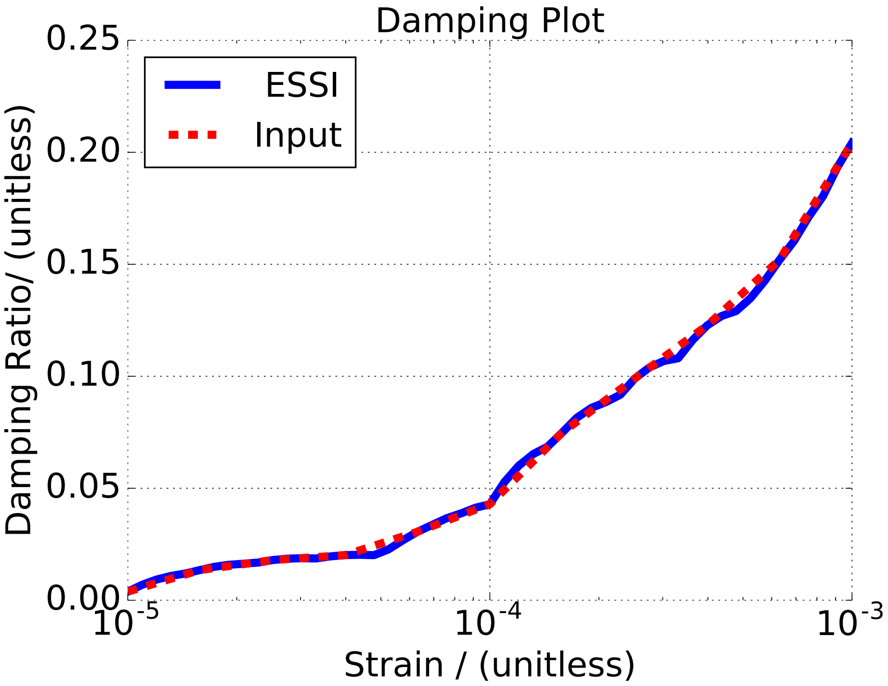


Figure 405.24: Damping Ratio Plot

405.4.3 Simulate Stiffness Reduction using von-Mises Armstrong-Frederick

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

405.4.3.1 Model description

This model illustrates the simulation of stiffness reduction using von-Mises Armstrong-Frederick. This example is based on one Gauss-point.

405.4.3.2 Real-ESSI input file:

```
model name "test";

add material # 1 type vonMisesArmstrongFrederick
    mass_density = 2500.0*kg/m^3
    elastic_modulus = 3E7*N/m^2
    poisson_ratio = 0.2
    von_mises_radius = 300 * Pa
    armstrong_frederick_ha = 5*3E7*N/m^2
    armstrong_frederick_cr = 25000
    isotropic_hardening_rate = 0*Pa
    ;
define NDMaterial constitutive integration algorithm Backward_Euler
    yield_function_relative_tolerance = 1E-6
    stress_relative_tolerance = 1E-6
    maximum_iterations = 30
    ;
incr_size = 0.000001 ;
max_strain= 0.005 ;
num_of_increm = max_strain/incr_size -1 ;
simulate constitutive testing strain control pure shear use material # 1
 confinement_strain = 0.0
 strain_increment_size = incr_size
 maximum_strain = max_strain
 number_of_increment = num_of_increm;
bye;
```

The von-Mises Armstrong-Frederick material behavior matches the stiffness reduction curve.

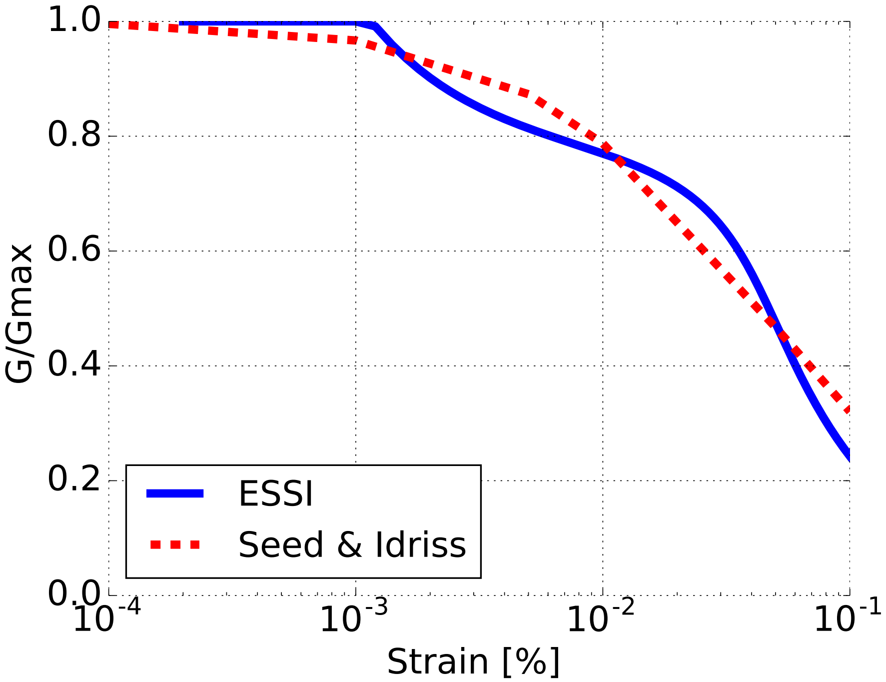


Figure 405.25: The stiffness reduction results.

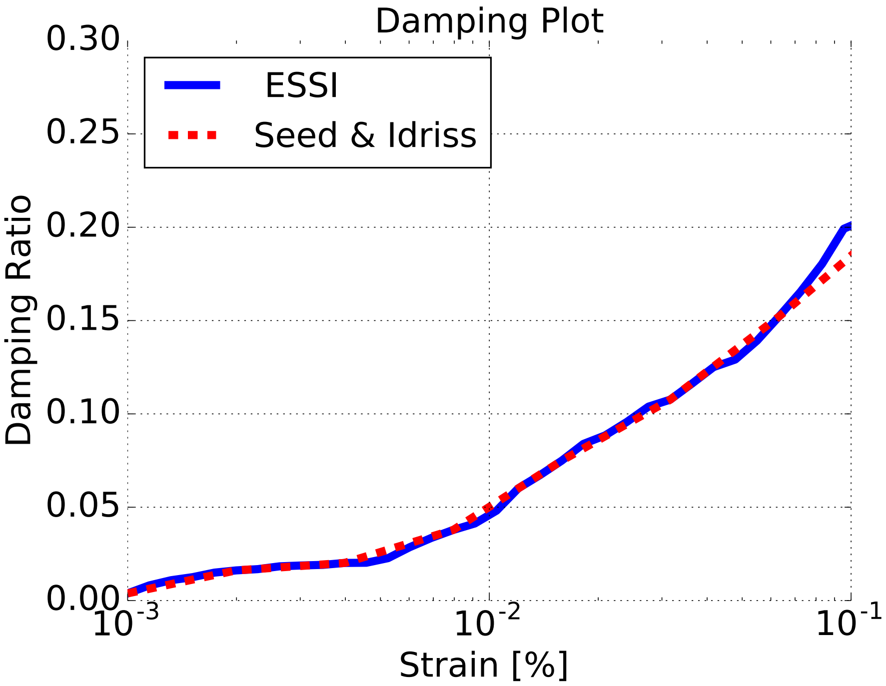


Figure 405.26: Damping Ratio Plot

405.5 Cosserat, Micropolar Material Modeling

405.5.1 Cosserat, Micropolar Elastic Material Model (example in development)

405.5.2 Cosserat, Micropolar Elastic-Plastic von Mises Material Model (example in development)

405.5.3 Cosserat, Micropolar Elastic-Plastic Druekcr Prager Material Model (example in development)

Chapter 406

Static Examples

(2016-2017-2019-2021-)

(In collaboration with Prof. José Abell, Dr. Yuan Feng, Mr. Sumeet Kumar Sinha, and Dr. Han Yang)

406.1 Chapter Summary and Highlights

In this Chapter static modeling and simulation of solids and structures is illustrated through a number of examples.

All the examples described here, and many more, organized in sub-directories, for constitutive behavior, static and dynamic behavior can be directly downloaded from a repository at: `http://sokocalo.engr.ucdavis.edu/~jeremic/lecture_notes_online_material/Real-ESSI_Examples/education_examples`. These examples can then be tried, analyzed using Real-ESSI Simulator that is available on Amazon Web Services (AWS) computers around the word. Login to AWS market place and search for Real-ESSI...

406.2 Static Elastic Solid Examples

406.2.1 Statics, Bricks, with Nodal Forces

406.2.1.1 Statics, 8 Node Brick, with Nodal Forces

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Problem description: a cantilever with a nodal force at the tip. Length=6m, Width=1m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.0$. The force direction was shown in Figure (406.1).

The mesh is generated with elastic 8 node brick.

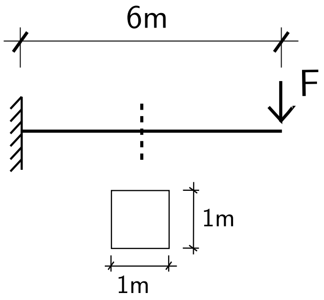


Figure 406.1: Problem description for cantilever beams.

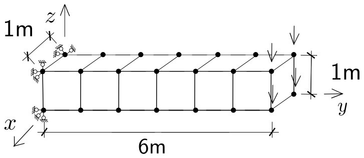


Figure 406.2: Six 8NodeBrick elements.

406.2.1.2 Statics, 27 Node Brick, with Nodal Forces

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Problem description: a cantilever with a nodal force at the tip. Length=6m, Width=1m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.0$. The force direction was shown in Figure (406.3).

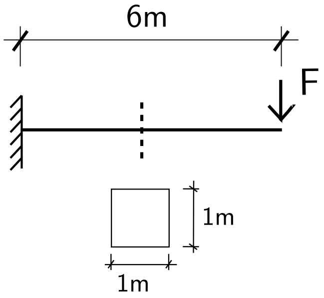


Figure 406.3: Problem description for cantilever beams.

The mesh is generated with elastic 27 node brick.

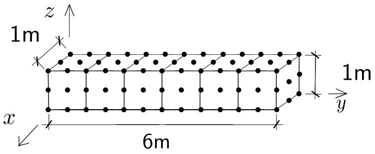


Figure 406.4: Six 27NodeBrick elements.

406.2.1.3 Statics, 8-27 Node Brick, with Nodal Forces

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Problem description: a cantilever with a nodal force at the tip. Length=2m, Width=2m, Height=2m, $\nu = 0.0$. The force direction was shown in Figure (406.5).

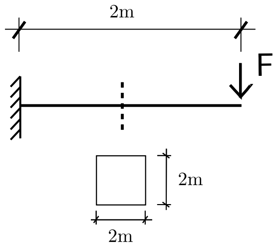


Figure 406.5: Problem description for cantilever beams.

The mesh is generated with an elastic 8-27 node brick. As shown in the Figure 406.16, some of the nodes are missing on purpose. The variable node brick element is usually used as the transition mesh between 8 node brick and 27 node brick.

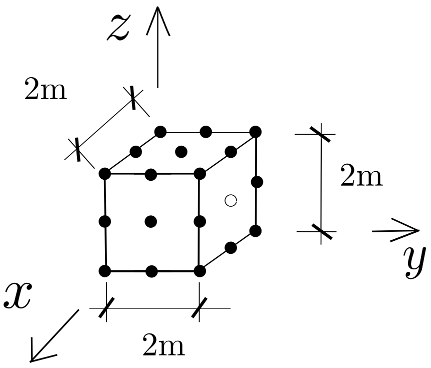


Figure 406.6: One 8-27 Node elements.

406.2.2 Statics, Bricks, with Surface Loads

406.2.2.1 Statics, 8 Node Brick, with Surface Forces

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Problem description: a cantilever with the load on one surface. Length=2m, Width=2m, Height=2m, $\nu = 0.0$. The force distribution was shown in Figure (406.7).

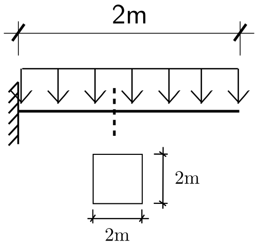


Figure 406.7: Problem description for cantilever beams.

The mesh is generated with an elastic 8 node brick.

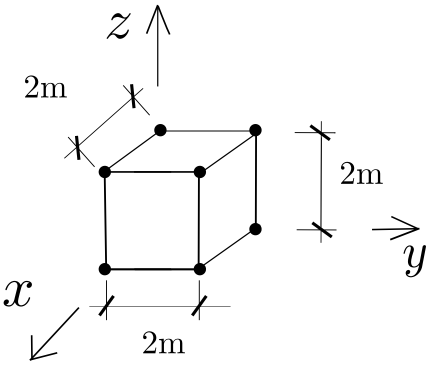


Figure 406.8: One element with surface load.

406.2.2.2 Statics, 27 Node Brick, with Surface Forces

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Problem description: a cantilever with the load on one surface. Length=2m, Width=2m, Height=2m. The force distribution was shown in Figure (406.9).

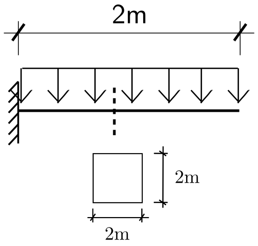


Figure 406.9: Problem description for cantilever beams.

The mesh is generated with an elastic 27 node brick.

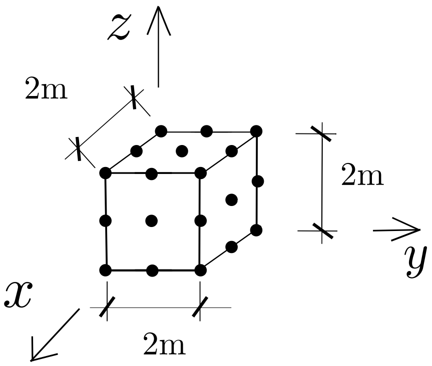


Figure 406.10: One element with surface load.

406.2.3 Statics, Bricks, with Body Forces

406.2.3.1 Statics, 8 Node Brick, with Body Forces

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Problem description: a cantilever with self weight on the whole element. Length=6m, Width=1m, Height=1m, $\nu = 0.3$. The force direction was shown in Figure (406.11).

The mesh is generated with an elastic 8 node brick.

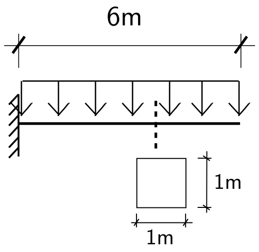


Figure 406.11: Problem description for cantilever beams.

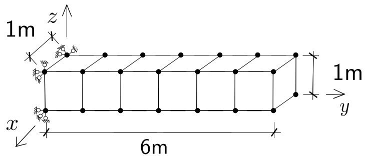


Figure 406.12: Six 8NodeBrick elements.

406.2.3.2 Statics, 27 Node Brick, with Body Forces

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Problem description: a cantilever with self weight on the whole element. Length=6m, Width=1m, Height=1m, $\nu = 0.3$. The force direction was shown in Figure (406.13).

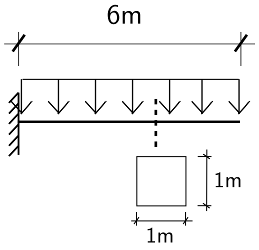


Figure 406.13: Problem description for cantilever beams.

The mesh is generated with an elastic 27 node brick.

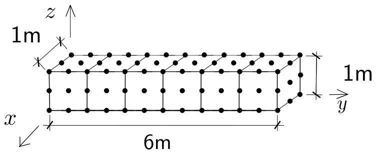


Figure 406.14: Six 27NodeBrick elements.

406.2.3.3 Statics, 8-27 Node Brick, with Body Forces

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Problem description: a cantilever with self weight on the whole element. Length=2m, Width=2m, Height=2m, $\nu = 0.3$. The force direction was shown in Figure (406.15).

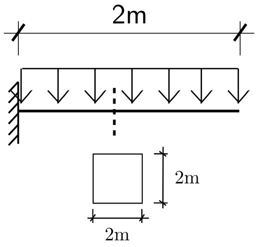


Figure 406.15: Problem description for cantilever beams.

The mesh is generated with an elastic 8-27 node brick. As shown in the Figure 406.16, some of the nodes are missing on purpose. The variable node brick element is usually used as the transition mesh between 8 node brick and 27 node brick.

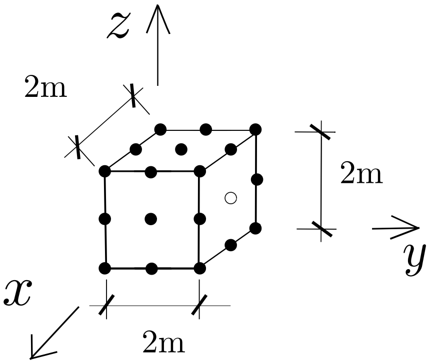


Figure 406.16: One variable Node Brick elements.

406.3 Static Elastic Structural Examples

406.3.1 Statics, Truss, with Nodal Forces

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Problem description: a cantilever with the nodal load on the tip. Length=1m, Cross Section=$1m^2$. The cross section shape is not necessarily a square. The force direction was shown in Figure (406.17). Truss only takes axial force.

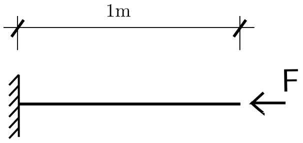


Figure 406.17: Problem description for a cantilever.

406.3.2 Statics, Elastic Beam, with Nodal Forces

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Problem description: a cantilever with nodal load on the tip. Length=1m, Width=1m, Height=1m, E=1Pa. The force direction was shown in Figure (406.18).

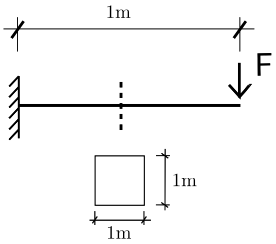


Figure 406.18: Problem description for cantilever beams.

406.3.3 Statics, Elastic Beam, with Body Forces

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Problem description: a cantilever with self weight. Length=1m, Width=1m, and Height=1m. The force direction was shown in Figure (406.19).

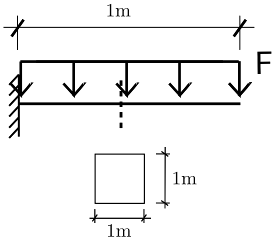


Figure 406.19: Problem description for cantilever beams.

406.3.4 Statics, ShearBeam Element

406.3.4.1 Problem description

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

In the element type "ShearBeam", only one Gauss point exists. ShearBeam element was used here to test the von Mises Armstrong-Frederickó material model. Vertical force F_z was used to apply confinement to the element. Then, cyclic force F_x is used to load. Usually, pressure-dependent materials, like Drucker-Prager, require the confinement. The pressure-independent materials, like von Mises, do not require the confinement.

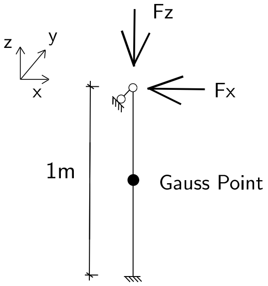


Figure 406.20: ShearBeam element.

406.3.4.2 Results

Resulting stress-strain relationship is shown in Fig.(707.51).

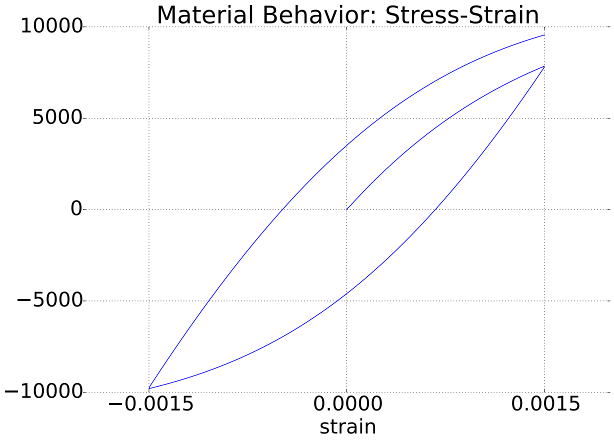


Figure 406.21: Shear stress-strain response.

406.3.5 Statics, Elastic Shell, with Nodal Forces

406.3.5.1 ANDES Shell, out of Plane Force

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Problem description: Length=6m, Width=1m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.0$.

The force direction was shown in Figure (406.22).

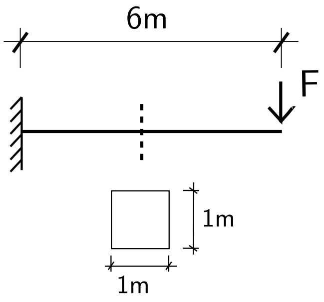


Figure 406.22: Problem description for cantilever beams.

406.3.5.2 ANDES Shell, Perpendicular to Plane, bending

The mesh and the out-of-plane force is shown in Fig. 406.23.

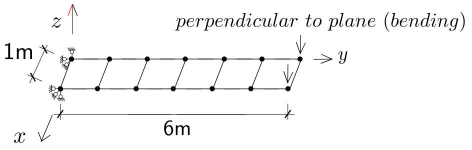


Figure 406.23: Six 4NodeANDES elements.

406.3.5.3 ANDES Shell, In-plane Force

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Problem description: a cantilever with a nodal force at the tip. Length=6m, Width=1m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.0$. The force direction was shown in Figure (406.24).

The mesh and the inplane force is shown in Fig. 406.25.

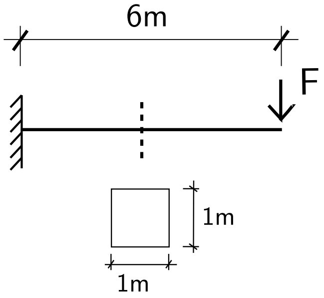


Figure 406.24: Problem description for cantilever beams.

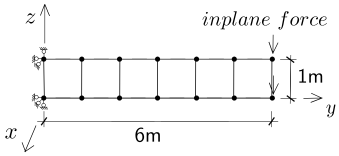


Figure 406.25: Six 4NodeANDES elements.

406.3.6 Statics, Elastic Shell, with Body Forces

406.3.6.1 ANDES shell under the out-of-Plane Body Force

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Problem description: Length=6m, Width=1m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.0$. The force direction was shown in Figure (406.26).

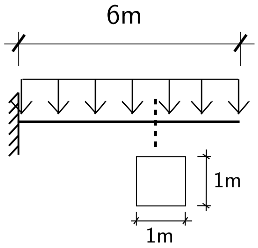


Figure 406.26: Problem description for cantilever beams.

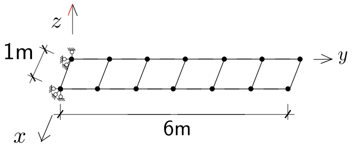


Figure 406.27: Six 4NodeANDES elements.

406.3.6.2 ANDES Shell, In-plane Body Force

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Problem description: a cantilever with a nodal force at the tip. Length=6m, Width=1m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.0$. The force direction was shown in Figure (406.28).

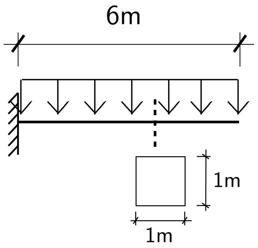


Figure 406.28: Problem description for cantilever beams.

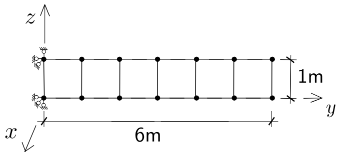


Figure 406.29: Six 4NodeANDES elements.

406.4 Statics, Interface/Contact Elements

406.4.1 Statics, Two Bar Normal Interface/Contact Problem Under Monotonic Loading.

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

This is an example of normal monotonic loading on a 1-D contact/interface between two bars separated by an initial gap of 0.1 unit. An illustrative diagram of the problem statement is shown below.

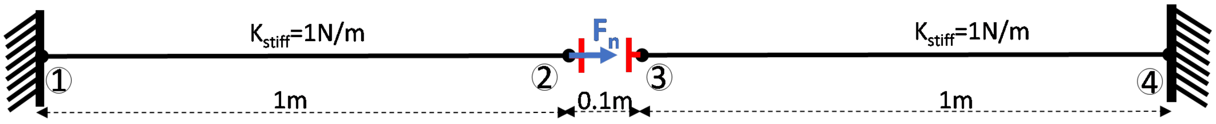


Figure 406.30: Illustration of Two Bar Normal Interface/Contact Problem under monotonic loading with initial gap.

The displacement output of *Node 2 and Node 3* are shown below.

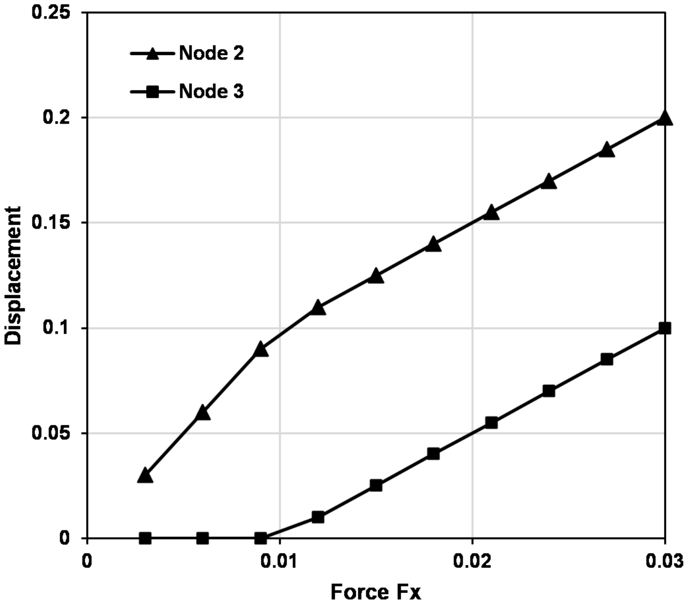


Figure 406.31: Displacement of Nodes 2 and 3.

406.4.2 Statics, Four Bar Interface/Contact Problem With Normal and Shear Force Under Monotonic Loading

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

This is an example to show the normal and tangential behavior (stick and slip case) of contacts using four bars in 2-D plane. The bars in x-directions are in contact/interface (initial gap=0).

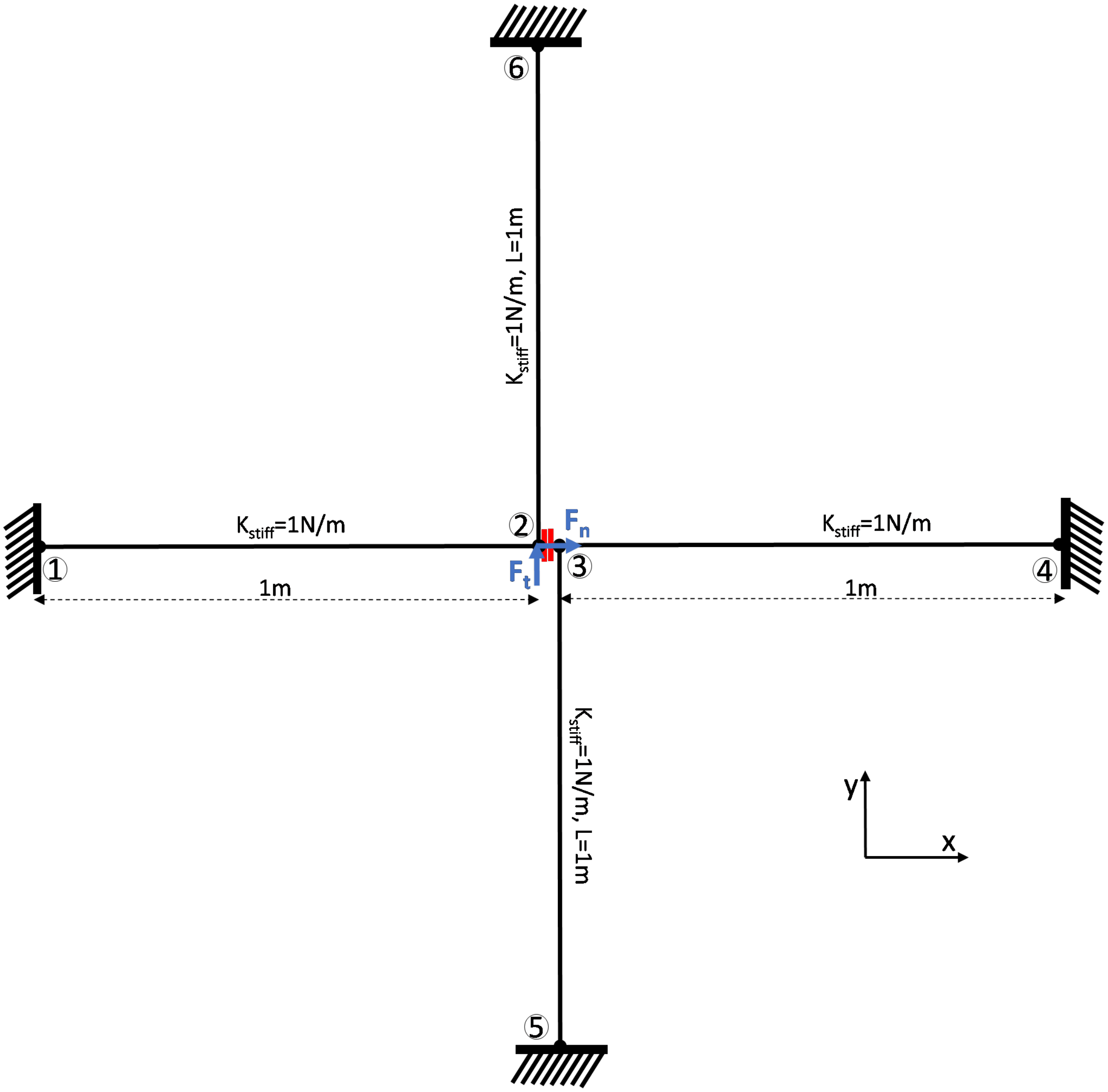


Figure 406.32: Illustration of Four Bar Normal Interface/Contact Problem With Normal and Shear Force Under Monotonic Loading with no initial gap.

The displacement output of *Node 2 and Node 3* are shown below.

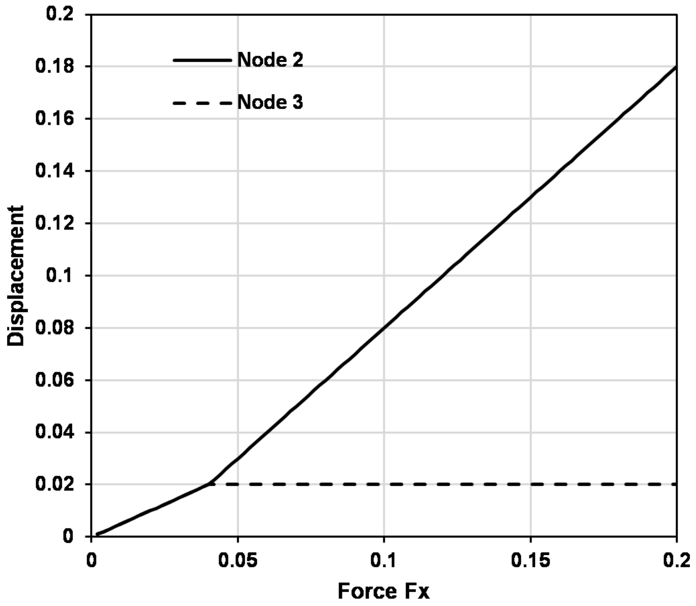


Figure 406.33: Displacement of Nodes 2 and 3 along y direction.

406.4.3 Statics, 3-D Truss example with normal confinement and Shear Loading

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

A simple 3-D truss example with Normal confinement in z-direction of $F_N = 0.5N$, friction coefficient $\mu = 0.2$ and shear loading of magnitude $F_s = 0.5N$. Figure 707.57 below, shows the description of the problem.

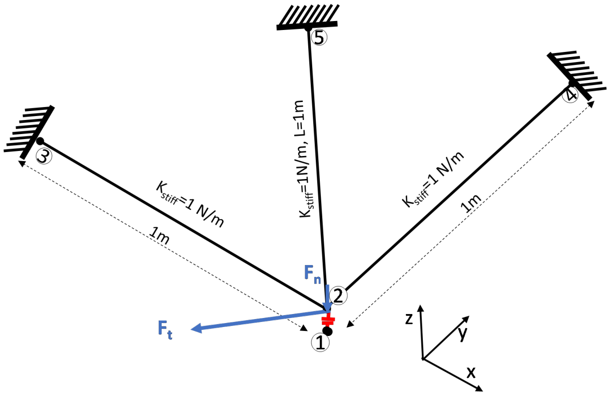


Figure 406.34: Illustration of 3-D Truss Problem with confinement loading in z-direction of 0.5N and then shear loading of 0.5N in x-y plane.

The generalized displacement response of the tangential loading stage is shown below.

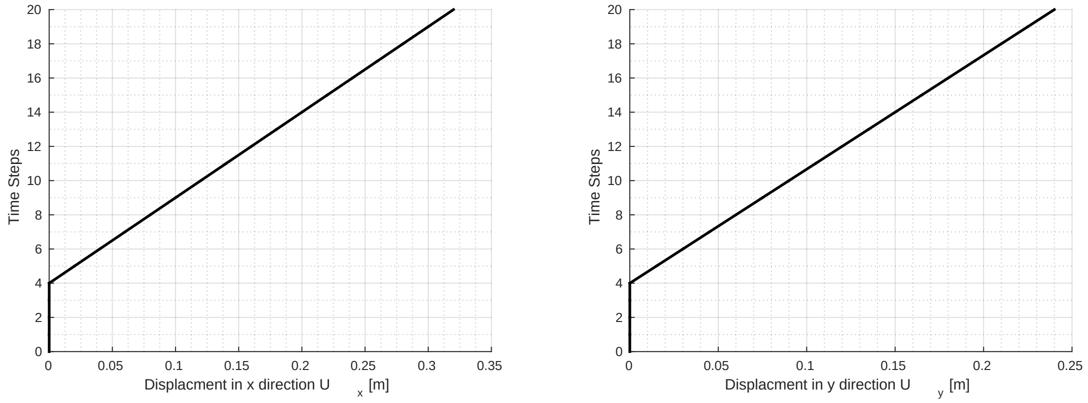


Figure 406.35: Displacements of Node 2 with applied shear tangential load step.

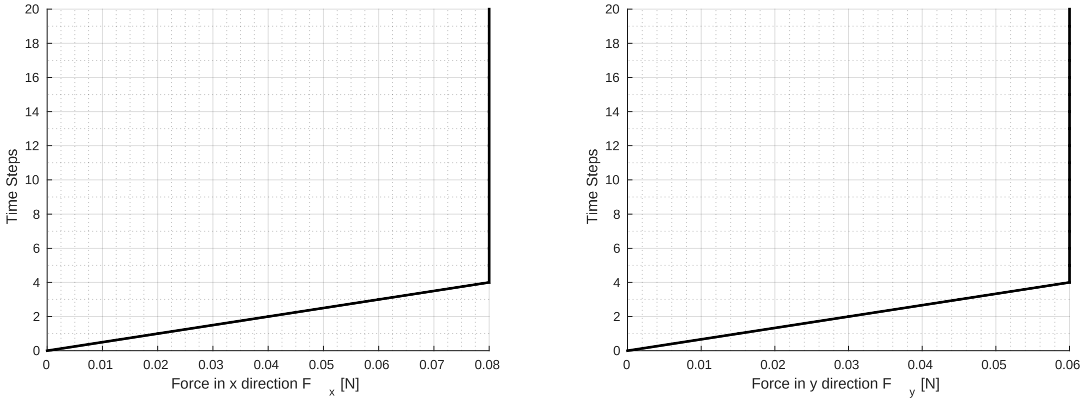


Figure 406.36: Resisting force by the contact/interface element with applied shear tangential load step.

406.4.4 Statics, Six Solid Blocks Example With Interface/Contact

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

This is a 3-D solid block example with initial normal and then tangential load on different surfaces as shown below.

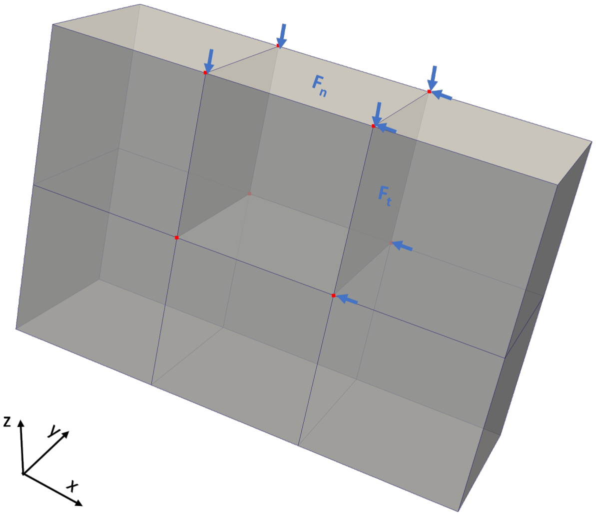


Figure 406.37: Illustration of Six Solid Blocks Example with Interface/Contact with first normal and then tangential loading stages.

The generalized displacement field of the two loading stages `normal loading` and *tangential loading* is shown below..

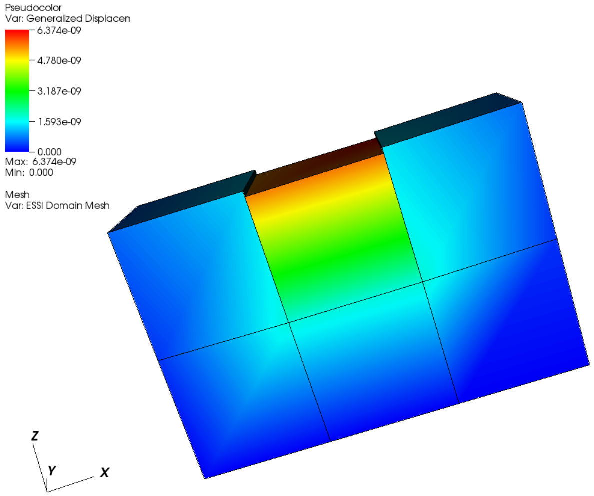


Figure 406.38: Generalized displacement magnitude visualization of normal loading.

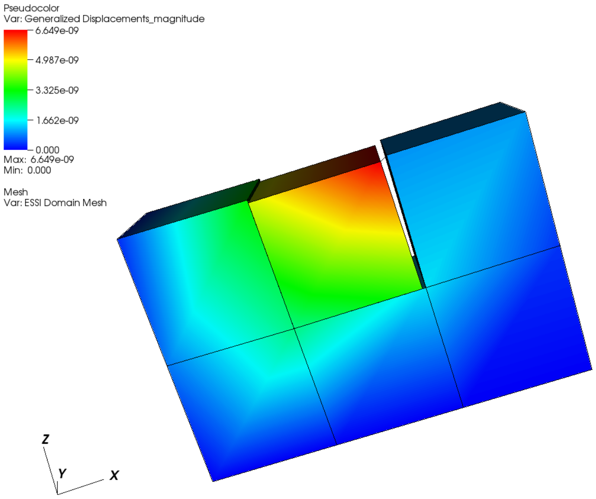


Figure 406.39: Generalized displacement magnitude visualization of tangential loading.

406.5 Static Inelastic Solid Examples

406.5.1 Statics, Bricks, Elastic-Plastic, von Mises, with Nodal Forces

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Model Description:

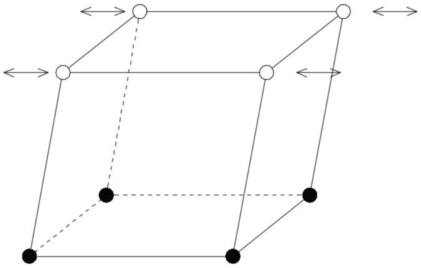


Figure 406.40: Perfectly Plastic Pure Shear Cyclic Loading.

Material Response at Gauss Point:

406.5.2 Statics, Bricks, Elastic-Plastic, Drucker Prager, with Nodal Forces

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

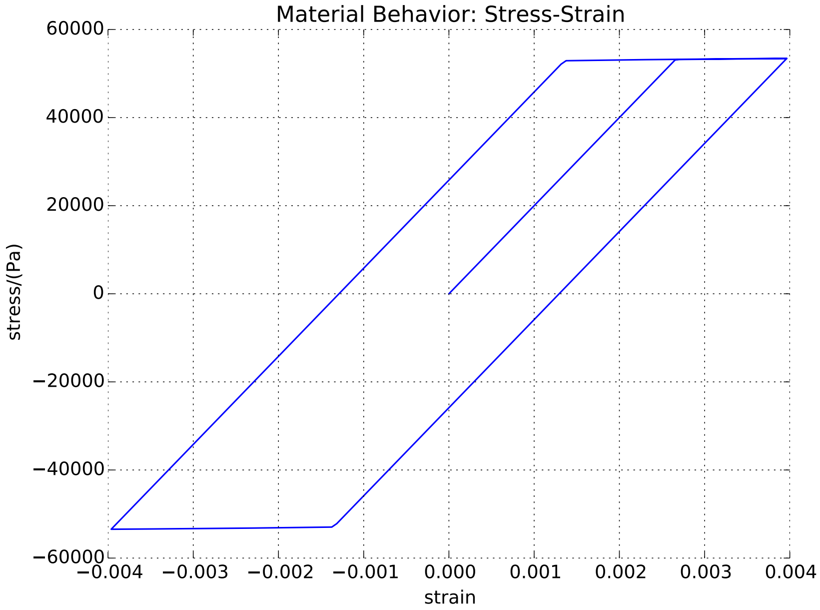


Figure 406.41: Results of Perfectly Plastic Pure Shear Cyclic Loading.

Model Description:

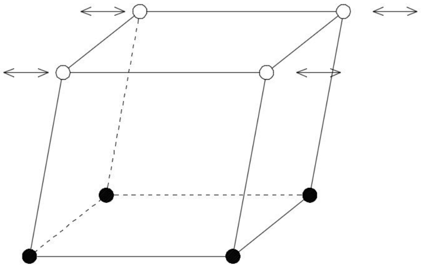


Figure 406.42: Diagram of Drucker-Prager Armstrong-Frederick Pure Shear Cyclic Loading.

Material Response at Gauss Point:

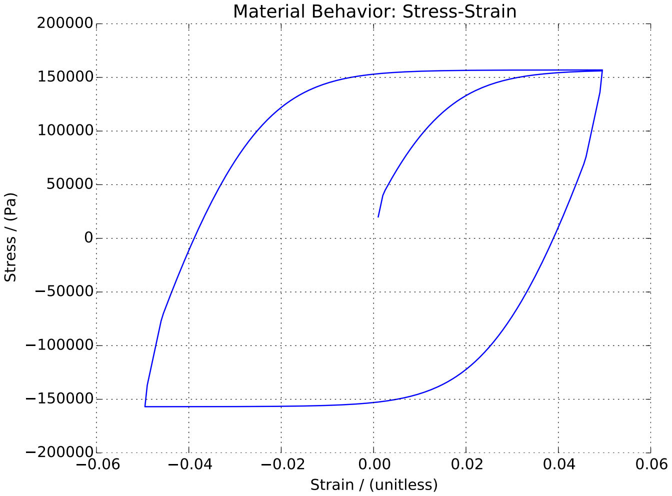


Figure 406.43: Result of Drucker-Prager Armstrong-Frederick Pure Shear Cyclic Loading.

406.6 Static Inelastic Shell Examples (example in development)

406.7 Statics, Elastic Single Solid Finite Finite Element Examples

406.7.1 Statics, Linear Elastic, Solid Examples

406.7.1.1 Statics, Pure Shear, Monotonic Loading

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Model Description:

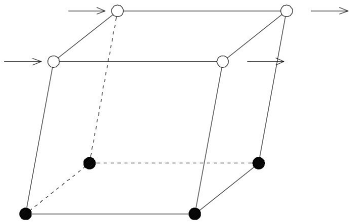


Figure 406.44: Diagram Linear Elastic Solid Pure Shear Monotonic Loading.

Material Response at Gauss Point:

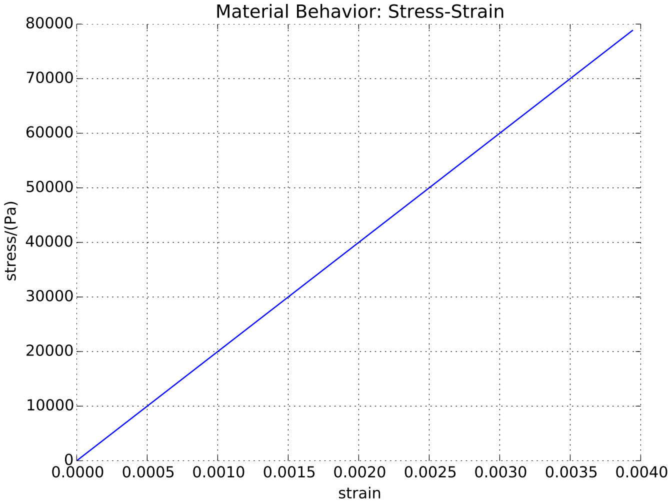


Figure 406.45: Results of Linear Elastic Solid Pure Shear Monotonic Loading.

406.7.1.2 Pure Shear, Cyclic Loading

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Model Description:

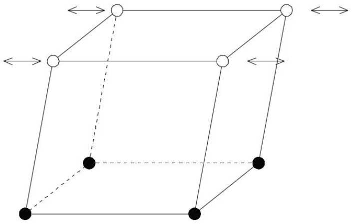


Figure 406.46: Diagram Linear Elastic Solid Pure Shear Cyclic Loading.

Material Response at Gauss Point:

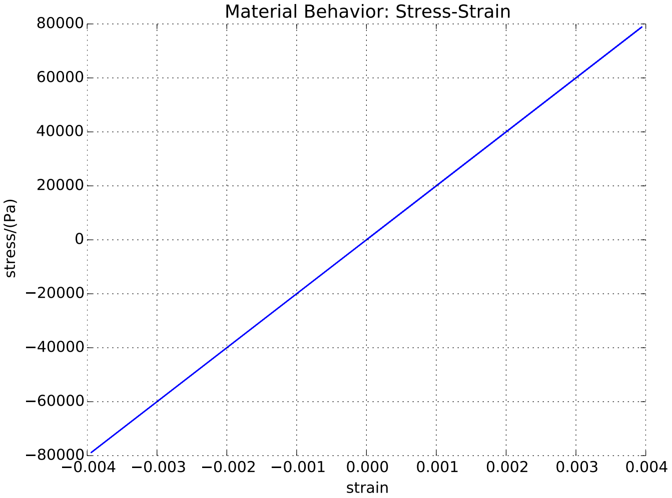


Figure 406.47: Results of Linear Elastic Solid Pure Shear Cyclic Loading.

406.7.1.3 Uniaxial Strain, Monotonic Loading

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Model Description:

Material Response at Gauss Point:

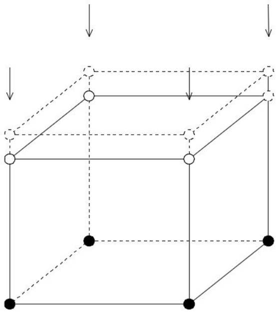


Figure 406.48: Diagram Linear Elastic Uniaxial Strain Solid Monotonic Loading.

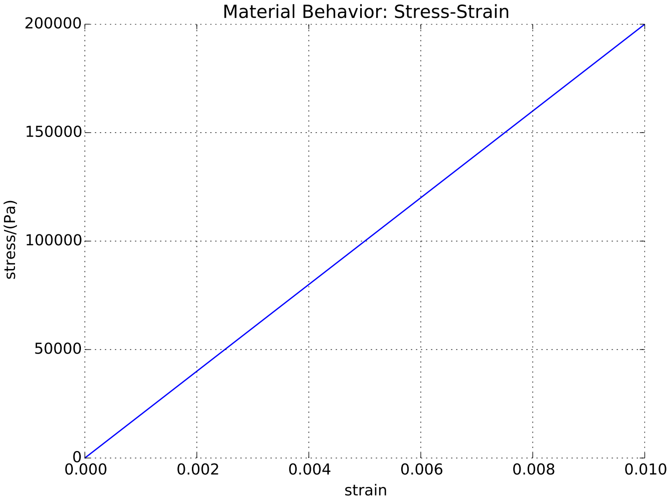


Figure 406.49: Results of Linear Elastic Pure Shear Monotonic Loading.

406.7.1.4 Uniaxial Strain, Cyclic Loading

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Model Description:

Material Response at Gauss Point:

406.7.2 Statics, Nonlinear Elastic, Duncan-Chang, Pure Shear, Solid Examples

406.7.2.1 Pure Shear, Monotonic Loading

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

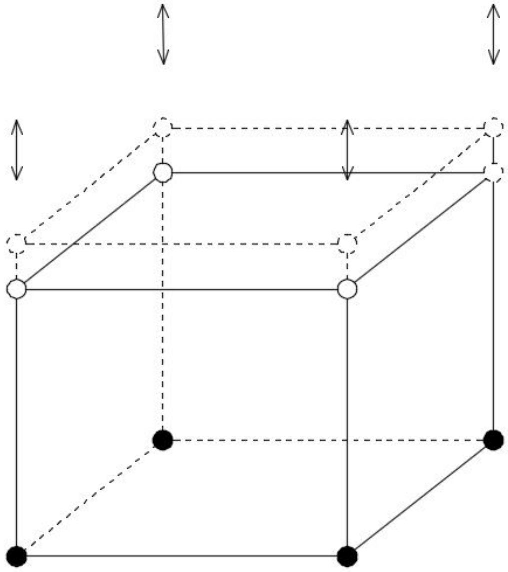


Figure 406.50: Linear Elastic Uniaxial Strain Cyclic Loading.

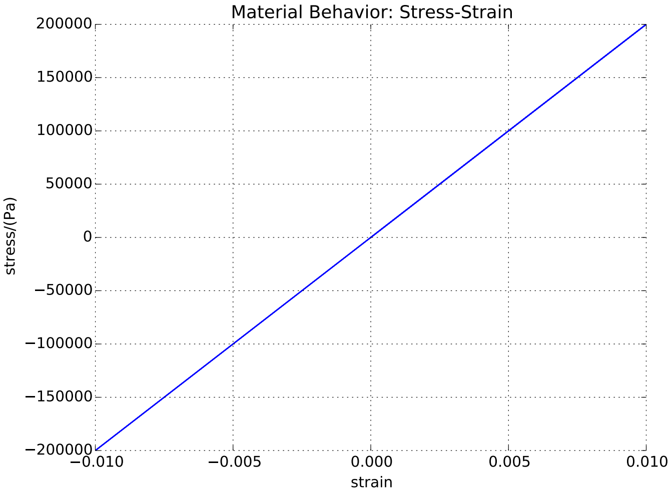


Figure 406.51: Results of Linear Elastic Pure Shear Cyclic Loading.

Model Description:

Material Response at Gauss Point:

406.7.2.2 Pure Shear, Cyclic Loading

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

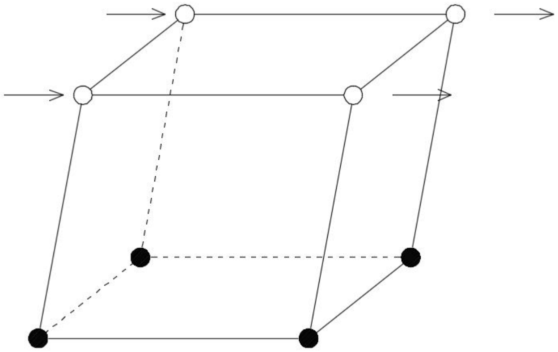


Figure 406.52: Nonlinear Elastic Uniaxial Strain Monotonic Loading.

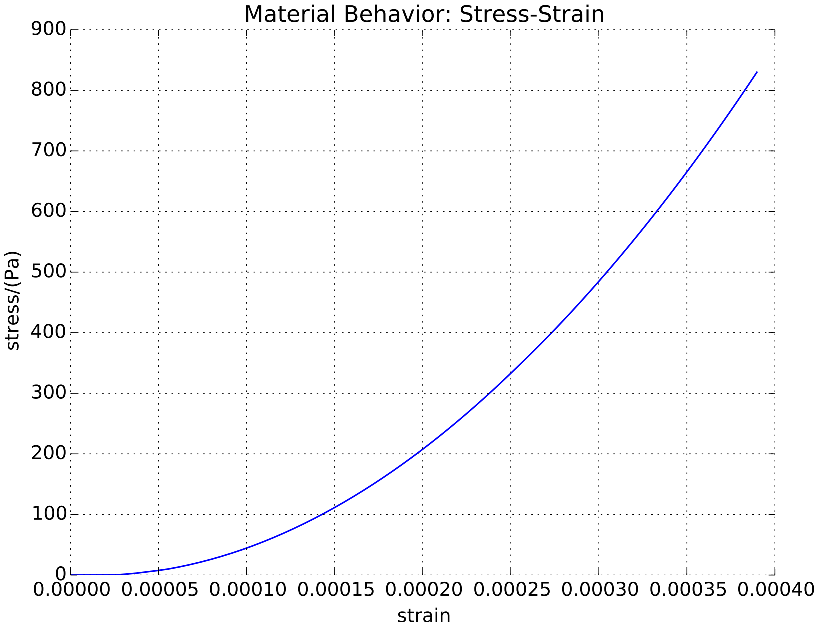


Figure 406.53: Results of Nonlinear Elastic Pure Shear Monotonic Loading.

Model Description:

Material Response at Gauss Point:

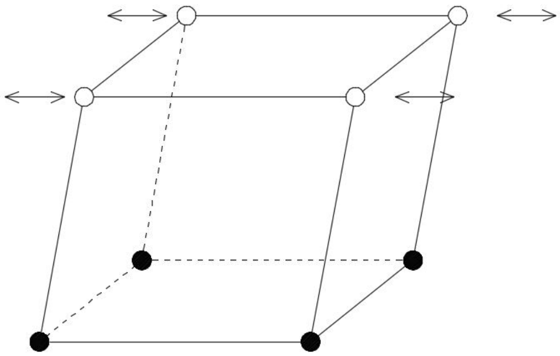


Figure 406.54: Nonlinear Elastic Uniaxial Strain Cyclic Loading.

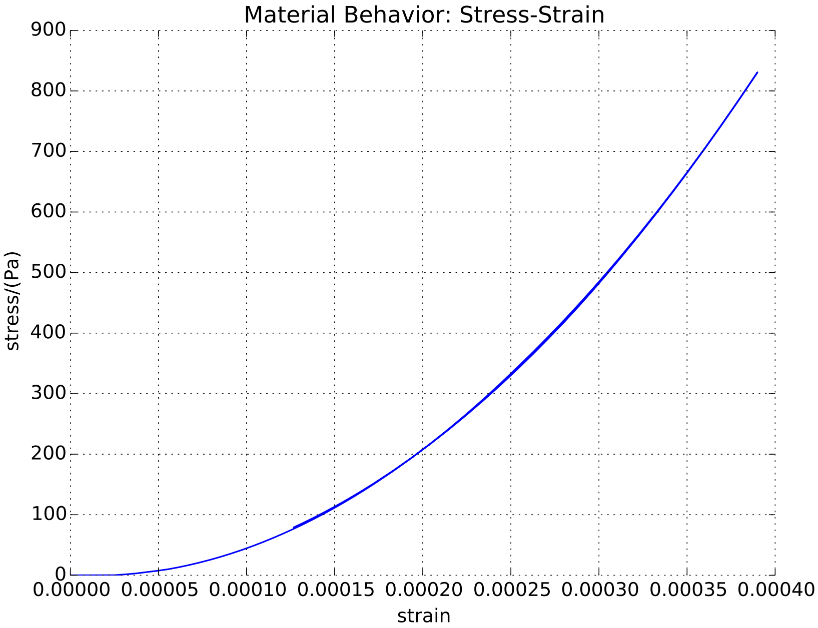


Figure 406.55: Results of Nonlinear Elastic Pure Shear Cyclic Loading.

406.8 Statics, Elastic-Plastic Single Solid Finite Element Examples

406.8.1 Statics, Elastic Perfectly Plastic, Cyclic Loading, Pure Shear Solid Examples

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Model Description:

Material Response at Gauss Point:

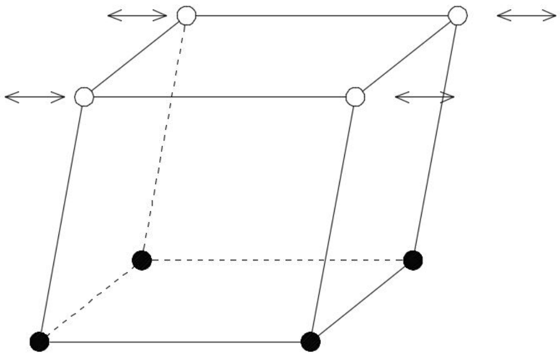


Figure 406.56: Perfectly Plastic Pure Shear Cyclic Loading.

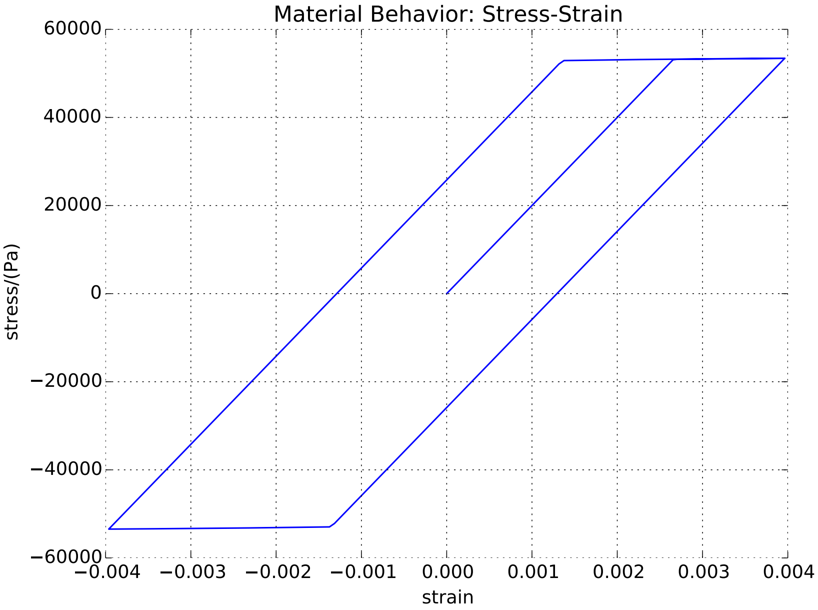


Figure 406.57: Results of Perfectly Plastic Pure Shear Cyclic Loading.

406.8.1.1 Statics, von-Mises Yield Function, Isotropic Hardening

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Model Description:

Material Response at Gauss Point:

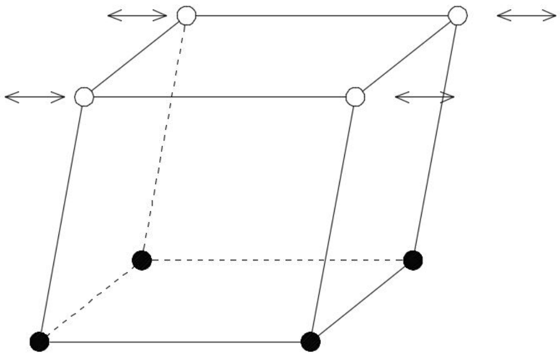


Figure 406.58: Pure Shear Cyclic Loading.

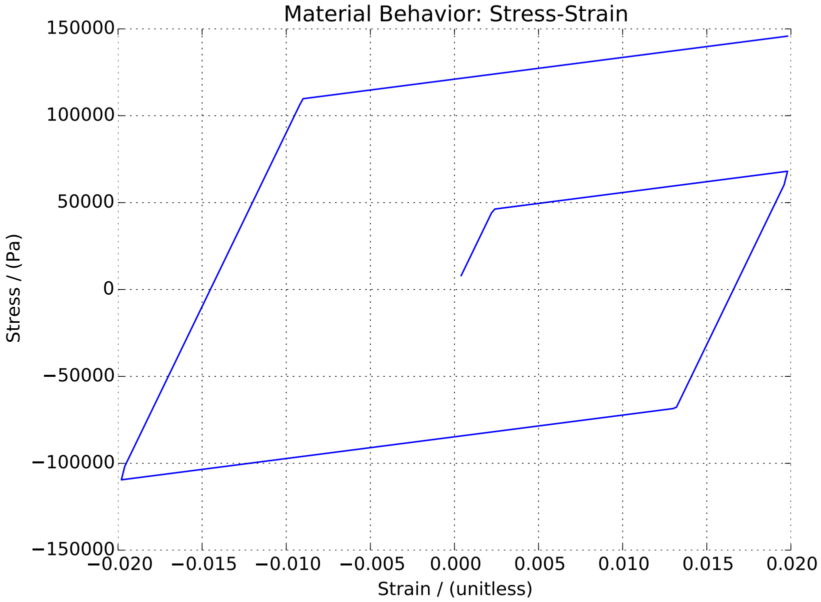


Figure 406.59: Material von-Mises Isotropic Hardening under Pure Shear Cyclic Loading.

406.8.1.2 Statics, von Mises Yield Function, Kinematic Hardening

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Model Description:

Material Response at Gauss Point:

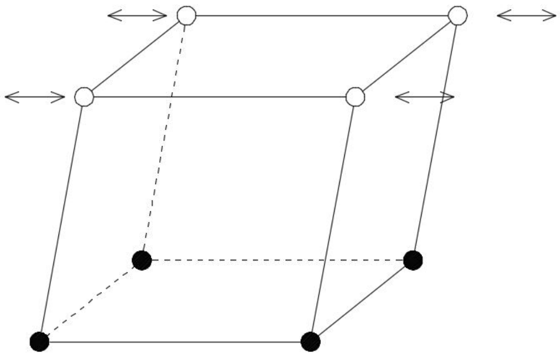


Figure 406.60: Pure Shear Cyclic Loading.

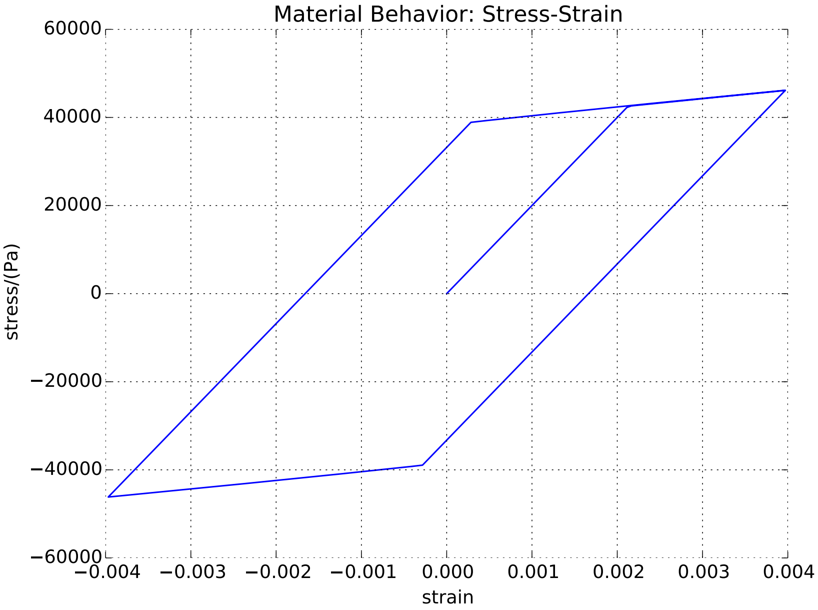


Figure 406.61: Results of von-Mises Kinematic Hardening Pure Shear Cyclic Loading.

406.8.1.3 Statics, Drucker Prager Yield Function, von-Mises Plastic Potential Function, Perfectly Plastic Hardening Rule

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Model Description:

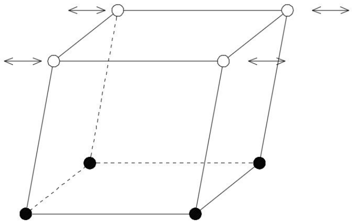


Figure 406.62: Pure Shear Cyclic Loading.

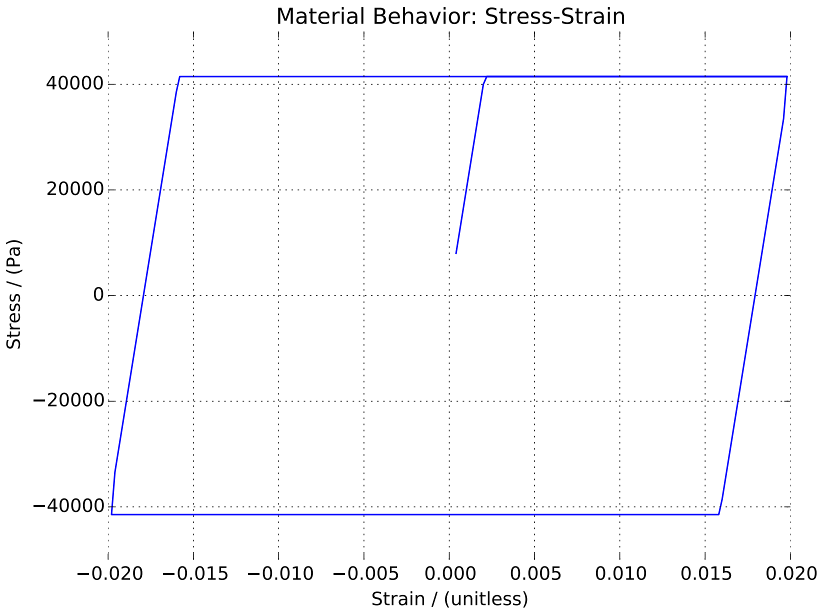


Figure 406.63: Results of Drucker Prager Yield Surface with Purely Deviatoric Plastic Flow under Pure Shear Cyclic Loading.

406.8.1.4 Statics, Drucker Prager Yield Function, Drucker Prager Plastic Potential Function, Perfectly Plastic Hardening Rule

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Model Description:

Material Response at Gauss Point:

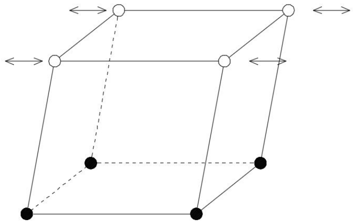


Figure 406.64: Pure Shear Cyclic Loading.

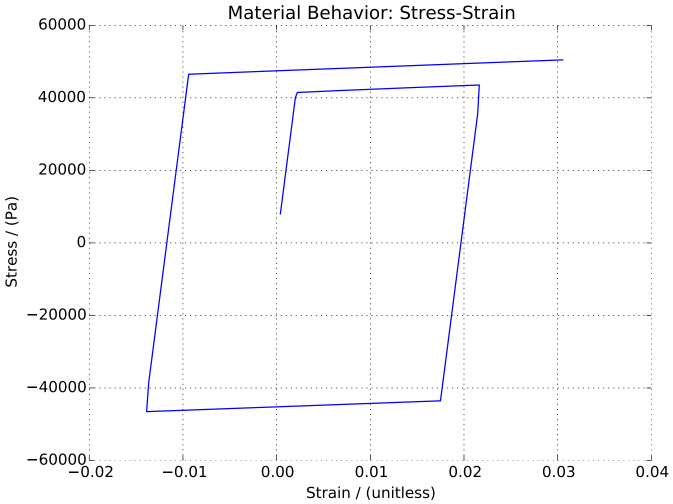


Figure 406.65: Results of Associative Drucker Prager Pure Shear Cyclic Loading.

406.8.2 Statics, Drucker Prager with Armstrong Frederick Nonlinear Kinematic Hardening Material Model

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Model Description:

Material Response at Gauss Point:

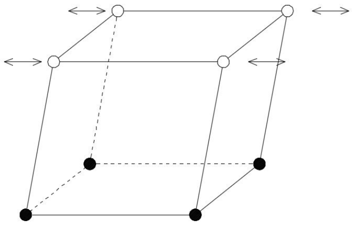


Figure 406.66: Pure Shear Cyclic Loading.

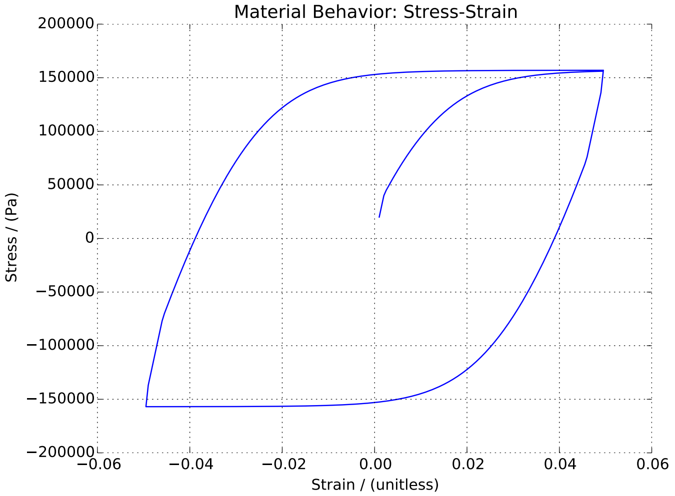


Figure 406.67: Result of Drucker-Prager Armstrong-Frederick Pure Shear Cyclic Loading.

406.9 Statics, Elastic, Fiber Cross Section Beam Finite Element Examples

406.9.1 Statics, Linear Elastic, Normal Loading and Pure Bending Fiber Cross Section Beam Finite Element Examples

406.9.1.1 Linear Elastic Normal Loading, Fiber Cross Section Beam Finite Element Examples

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

The linear elastic beam is represented by the elastic section. This example is under the load of normal loading.

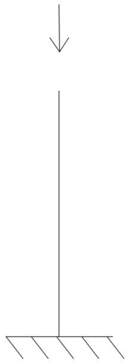


Figure 406.68: Normal Loading on the Fiber Beam with Elastic Section.

The elastic section represents the cross section properties of the beam.

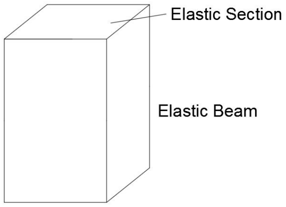


Figure 406.69: Diagram of the Fiber Beam with Elastic Section.

406.9.1.2 Linear Elastic Pure Bending, Fiber Cross Section Beam Finite Element Examples

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

The linear elastic beam is represented by the elastic section. This example is under the bending load.

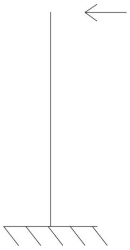


Figure 406.70: Bending on the Fiber Beam with Elastic Section.

The elastic section represents the cross section properties of the beam.

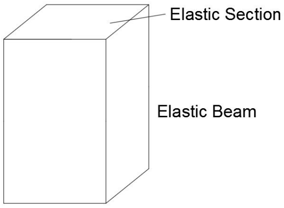


Figure 406.71: Diagram of the Fiber Beam with Elastic Section.

406.10 Statics, Elastic-Plastic, Fiber Cross Section Beam Finite Element Examples

406.10.1 Statics, Elastic-Plastic, Normal Loading and Pure Bending Fiber Cross Section Beam Finite Element

406.10.1.1 Elastic-Plastic Normal Loading, (Fiber Cross Section) Beam Finite Element Examples

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

The Elastic-Plastic beam is represented by the fiber section. This example is under the load of normal loading.

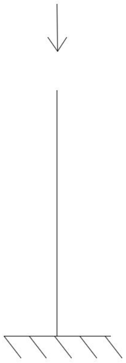


Figure 406.72: Normal Loading on the Fiber Beam with Elastic-Plastic Section.

The fiber represents the rebar. The section of all fibers represents the cross section properties of the inelastic beam.

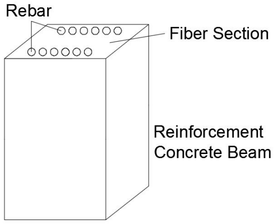


Figure 406.73: Diagram of the Fiber Beam with Elastic-Plastic Section.

406.10.1.2 Elastic-Plastic Pure Bending, (Fiber Cross Section) Beam Finite Element Examples

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

The Elastic-Plastic beam is represented by the fiber section. This example is under the load of normal loading.

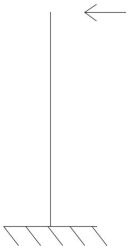


Figure 406.74: Bending on the Fiber Beam with Elastic-Plastic Section.

The fiber represents the rebar. The section of all fibers represents the cross section properties of the inelastic beam.

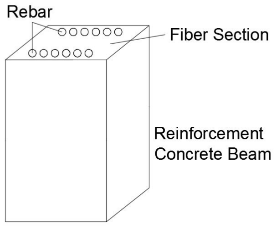


Figure 406.75: Diagram of the Fiber Beam with Elastic-Plastic Section.

406.11 Statics, Elastic, Inelastic Wall Finite Element Examples

406.11.1 Statics, Linear Elastic, Wall Finite Element Examples

406.11.1.1 Statics, Linear Elastic, Wall Finite Element Examples

406.11.1.2 Linear Elastic, Bi-Axial, Wall Finite Element Examples

406.11.1.3 Linear Elastic, Shear, (Fiber Cross Section) Wall Finite Element Examples

406.12 Statics, Elastic-Plastic Wall Finite Element Examples

406.12.1 Statics, Elastic-Plastic, in Plane, Wall Finite Element Examples

406.12.1.1 Elastic-Plastic, Uni-Axial, Wall Finite Element Examples

406.12.1.2 Elastic-Plastic, Bi-Axial, Wall Finite Element Examples

406.12.1.3 Elastic-Plastic, Shear, Wall Finite Element Examples

406.13 Statics, Solution Advancement Control

406.13.1 Increments: Load Control

When load-control is used as the solution advancement method, perfectly plastic model will fail immediately after the yield point. Load-control works with isotropic hardening and kinematic hardening.

406.13.1.1 Solids Example, Elastic Plastic Isotropic Hardening

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Model Description:

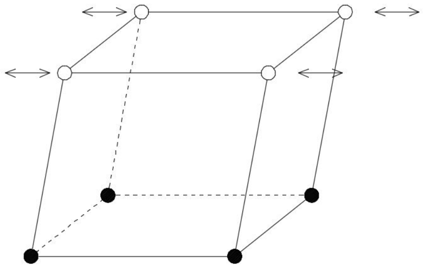


Figure 406.76: Pure Shear Cyclic Loading.

Material Response at Gauss Point:

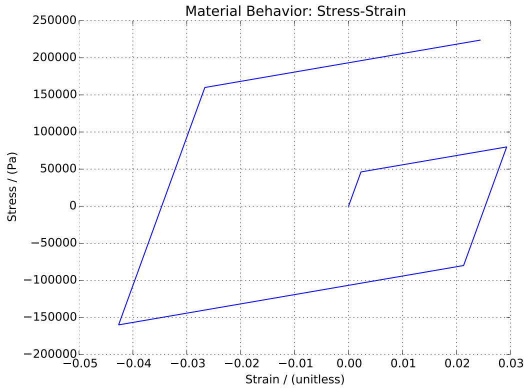


Figure 406.77: Material von-Mises Isotropic Hardening under Pure Shear Cyclic Loading.

406.13.1.2 Solids Example, Elastic Plastic Kinematic Hardening

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Model Description:

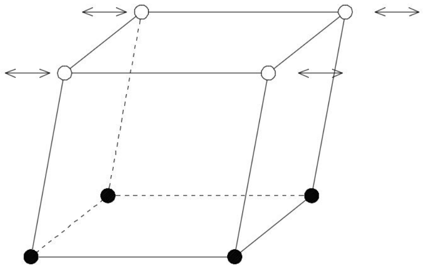


Figure 406.78: Pure Shear Cyclic Loading.

Material Response at Gauss Point:

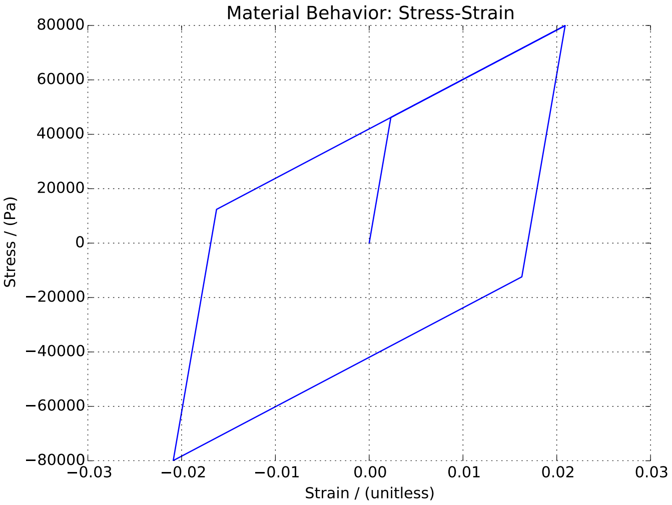


Figure 406.79: Material von-Mises Kinematic Hardening under Pure Shear Cyclic Loading.

406.13.1.3 Inelastic Beam Example, Steel and Reinforced Concrete

The Real-ESSI input files for this example are available HERE.

The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

The Elastic-Plastic beam is represented by the fiber section. This example is under the load of normal loading.

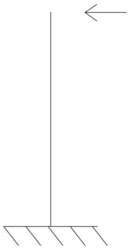


Figure 406.80: Normal Loading on the Beam with Fiber Section.

The fiber represents the rebar. The section of all fibers represents the cross section properties of the inelastic beam.

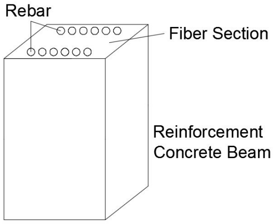


Figure 406.81: Diagram of the Beam with Fiber Section.

406.13.2 Statics, Increments: Displacement Control

406.13.2.1 Statics, Single Displacement Control

406.13.2.2 Solids Example, Elastic-Perfectly Plastic

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Model Description:

Material Response at Gauss Point:

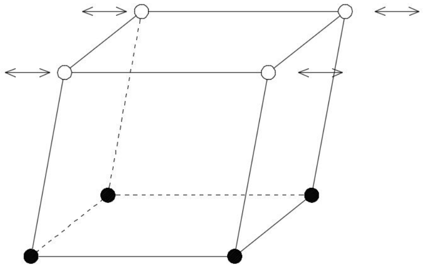


Figure 406.82: Pure Shear Cyclic Loading.

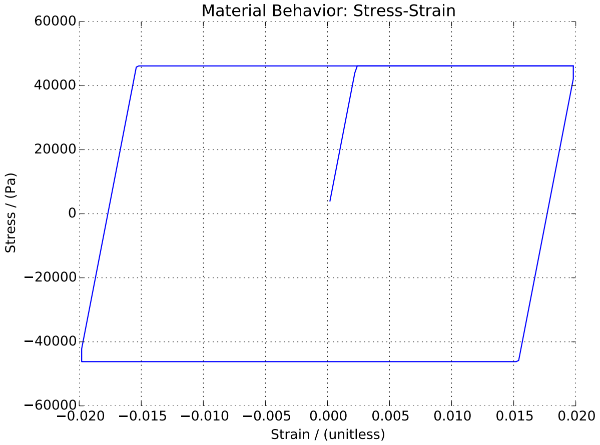


Figure 406.83: Displacement-Control of Perfectly Plastic Material under Pure Shear Cyclic Loading.

406.13.2.3 Solids Example, Elastic Plastic Isotropic Hardening

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Model Description:

Material Response at Gauss Point:

406.13.2.4 Solids Example, Elastic Plastic Kinematic Hardening

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Model Description:

Material Response at Gauss Point:

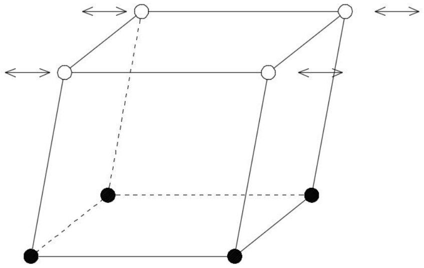


Figure 406.84: Pure Shear Cyclic Loading.

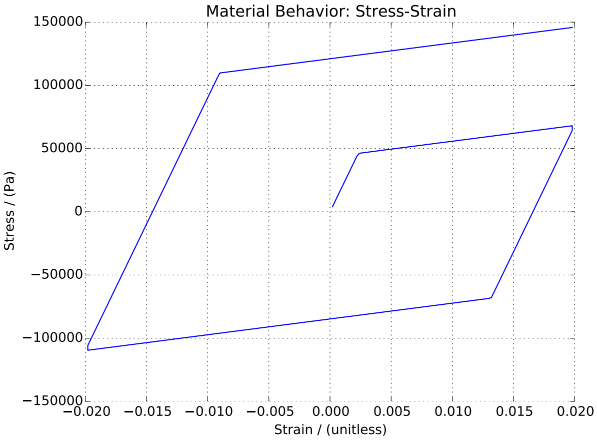


Figure 406.85: Displacement-Control of Isotropic Hardening Material under Pure Shear Cyclic Loading.

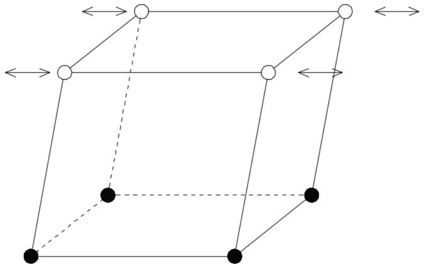


Figure 406.86: Pure Shear Cyclic Loading.

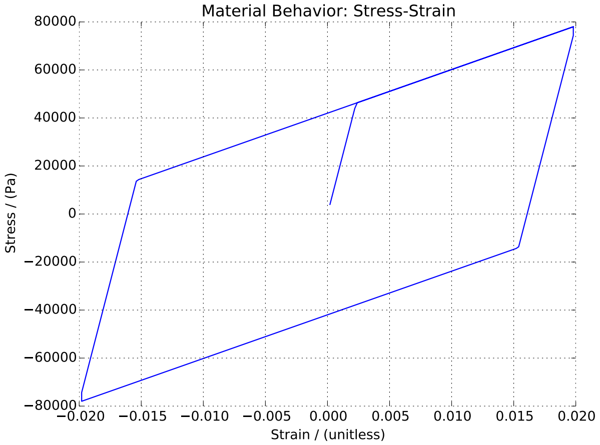


Figure 406.87: Displacement-Control of Kinematic Hardening Material under Pure Shear Cyclic Loading.

406.13.2.5 Inelastic Beam Example, Steel and Reinforced Concrete

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

The Elastic-Plastic beam is represented by the fiber section. This example is under the load of normal loading.

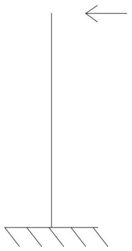


Figure 406.88: Bending on the Fiber Beam with Elastic-Plastic Section

The fiber represents the rebar. The section of all fibers represents the cross section properties of the inelastic beam.

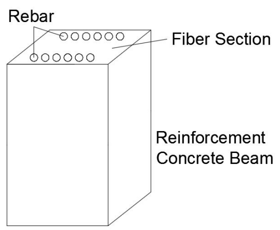


Figure 406.89: Diagram of the Fiber Beam with Elastic-Plastic Section.

406.13.3 Statics, Solution Algorithms

406.13.3.1 Statics, Solution Algorithm: No Convergence Check

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

When no convergence check is used, the stress-strain curves drift away a little. The stress-strain curve did not close, as shown in Figure 406.57.

Model Description:

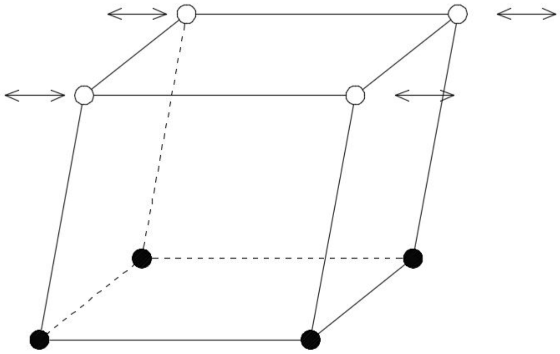


Figure 406.90: Pure Shear Cyclic Loading.

Material Response at Gauss Point:

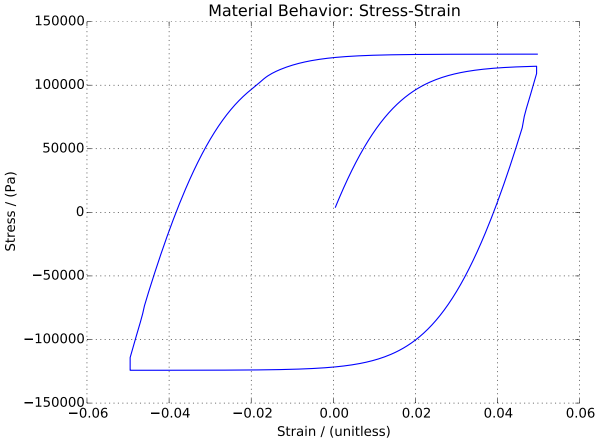


Figure 406.91: Results of No-Convergence-Check Pure Shear Cyclic Loading.

406.13.3.2 Statics, Solution Algorithm: Newton Algorithm

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Model Description:

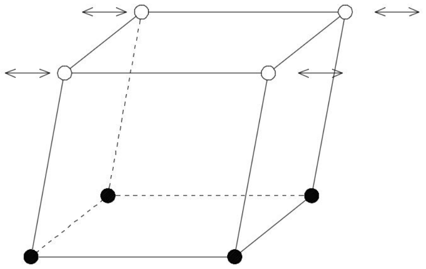


Figure 406.92: Pure Shear Cyclic Loading.

Material Response at Gauss Point:

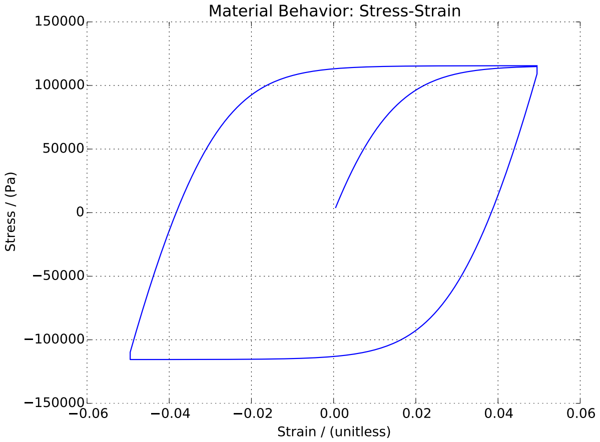


Figure 406.93: Results of Convergence Check with Newton-Raphson Iteration under Pure Shear Cyclic Loading.

406.13.3.3 Statics, Solution Algorithm: Newton Algorithm with Line Search

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Model Description:

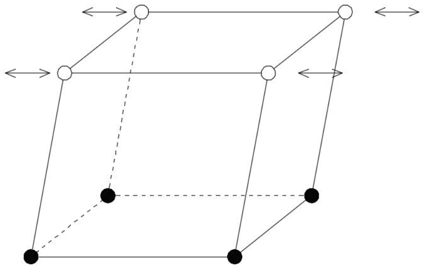


Figure 406.94: Pure Shear Cyclic Loading.

Material Response at Gauss Point:

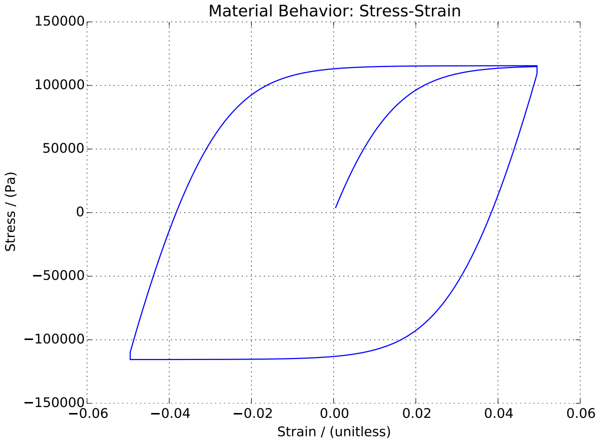


Figure 406.95: Results of Convergence Check with Newton-Raphson Iterations and Line Search under Pure Shear Cyclic Loading.

406.13.4 Statics, Solution Advancement Control, Iterations: Convergence Criteria

406.13.4.1 Statics, Solution Advancement Control, Convergence Criteria: Unbalanced Force

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Model Description:

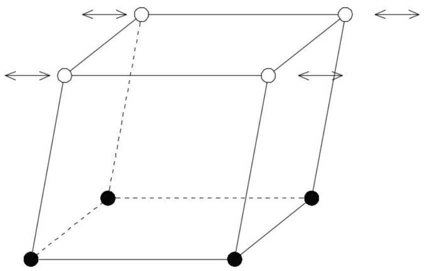


Figure 406.96: Pure Shear Cyclic Loading.

Material Response at Gauss Point:

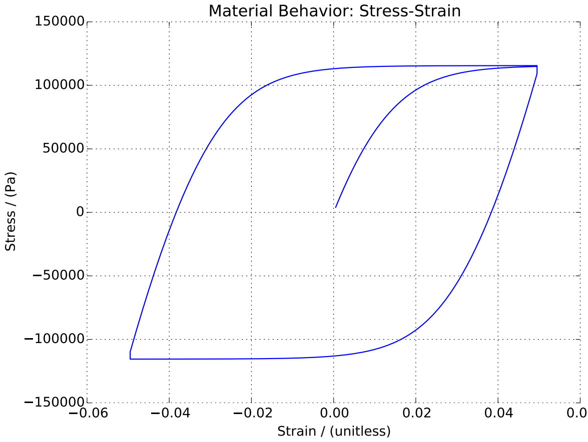


Figure 406.97: Results of Convergence Check with Unbalanced Force Criteria under Pure Shear Cyclic Loading.

406.13.4.2 Statics, Solution Advancement Control, Convergence Criteria: Displacement Increment

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Model Description:

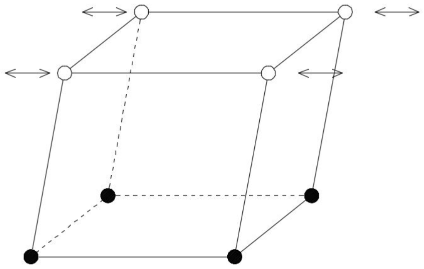


Figure 406.98: Pure Shear Cyclic Loading.

Material Response at Gauss Point:

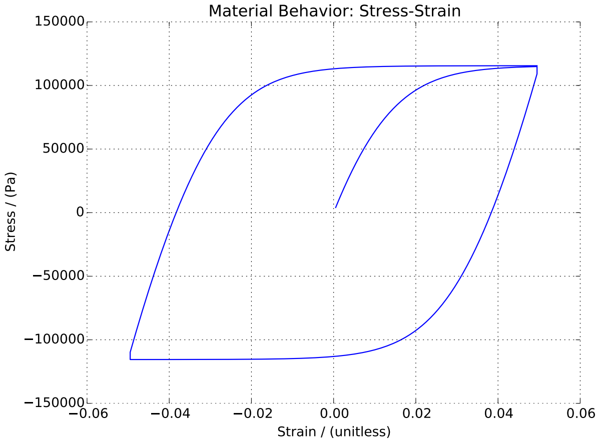


Figure 406.99: Results of Convergence Check with Displacement Increment under Pure Shear Cyclic Loading.

406.13.5 Statics, Solution Advancement Control, Different Convergence Tolerances (Examples in preparation)

406.14 Statics, Small Practical Examples

406.14.1 Statics, Elastic Beam Element for a Simple Frame Structure

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

406.14.1.1 Problem Description

- Dimensions: width=$6m$, height=$6m$, force=$100N$
- Element dimensions: length=6m, cross section width=1m, cross section height=1m, mass density $\rho = 0.0\text{kN/m}^3$, Young's modulus $E = 1E8$ Pa, Poisson's ratio $\nu = 0.0$.

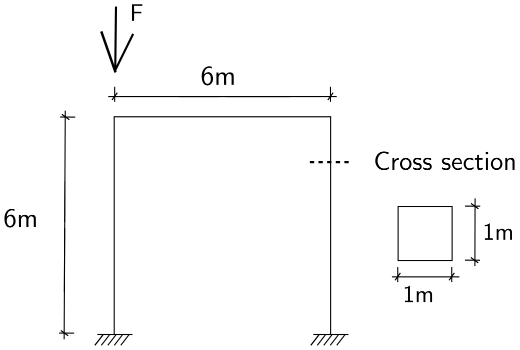


Figure 406.100: Elastic frame with beam_elastic elements.

406.14.2 Statics, 4NodeANDES Square Plate, Four Edges Clamped

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

406.14.2.1 Problem description

Length=20m, Width=20m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.3$.

The four edges are clamped.

The load is a self weight.

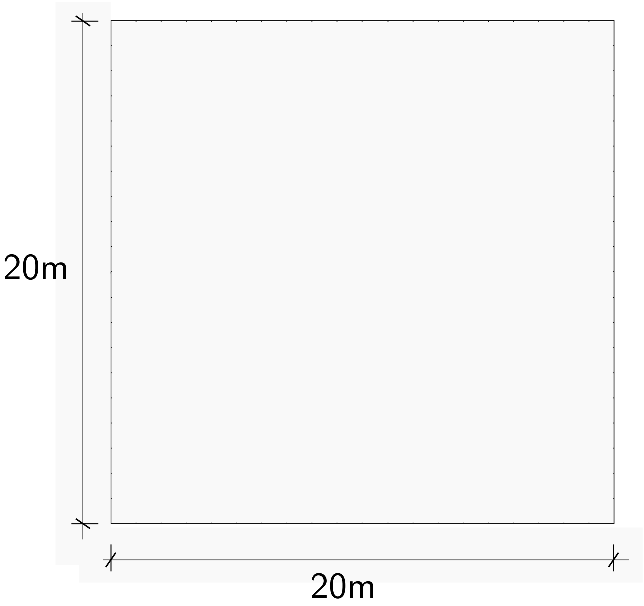


Figure 406.101: Square plate with four edges clamped.

406.14.2.2 Numerical model

The element side length is 1 meter.

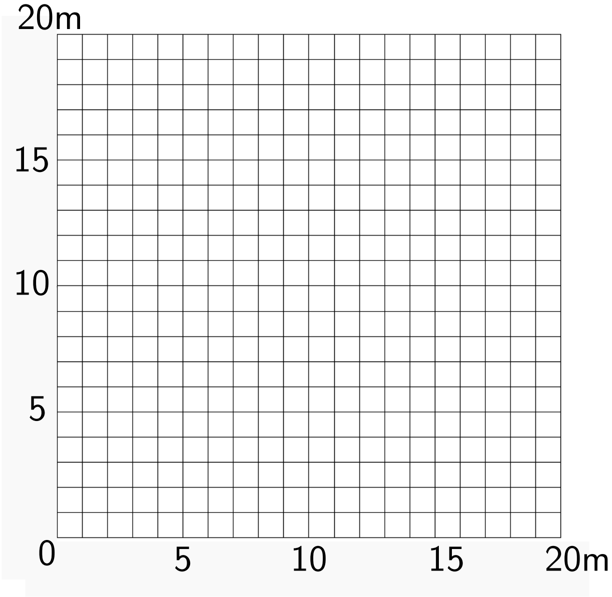


Figure 406.102: 4NodeANDES edge clamped square plate with element side length 1m.

406.14.3 Statics, Six Solid Blocks Example With Interface/Contact

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

This is a 3-D solid block example with initial normal and then tangential load on different surfaces as shown below.

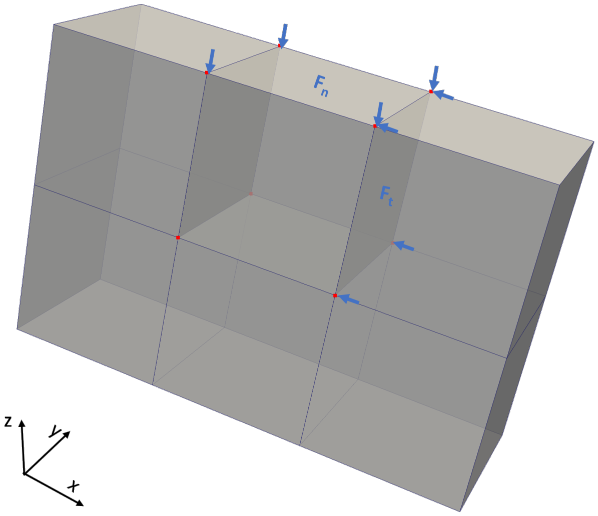


Figure 406.103: Illustration of Six Solid Blocks Example with Interface/Contact with first normal and then tangential loading stages.

The generalized displacement field of the two loading stages normal loading and tangential loading is shown below..

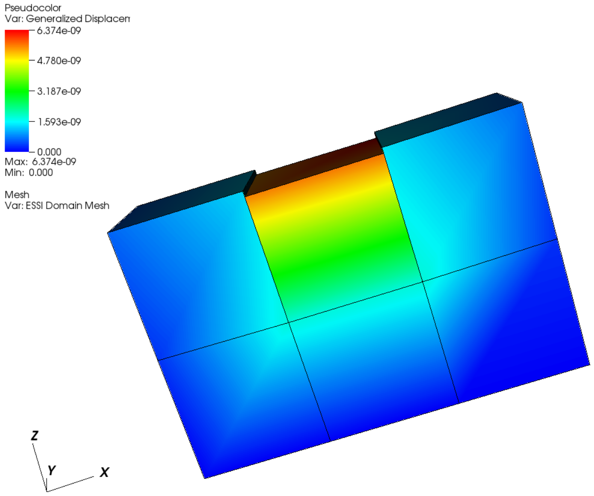


Figure 406.104: Generalized displacement magnitude visualization of normal loading.

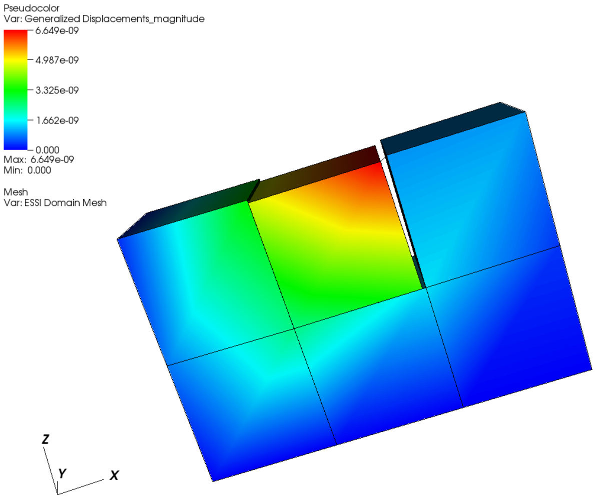


Figure 406.105: Generalized displacement magnitude visualization of tangential loading.

Chapter 407

Dynamic Examples

(2016-2017-2018-2019-2021-)

(In collaboration with Prof. José Abell, Dr. Yuan Feng, Mr. Sumeet Kumar Sinha, Dr. Hexiang Wang, and Dr. Han Yang)

407.1 Chapter Summary and Highlights

In this Chapter dynamic/transient modeling and simulation of solids and structures is illustrated through a number of examples.

All the examples described here, and many more, organized in sub-directories, for constitutive behavior, static and dynamic behavior can be directly downloaded from a repository at: `http://sokocalo.engr.ucdavis.edu/~jeremic/lecture_notes_online_material/Real-ESSI_Examples/education_examples`. These examples can then be tried, analyzed using Real-ESSI Simulator that is available on Amazon Web Services (AWS) computers around the word. Login to AWS market place and search for Real-ESSI...

407.2 Dynamic Solution Advancement (in Time)

407.2.1 Dynamics: Newmark Method

407.2.1.1 Newmark Model Description

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Firstly, the model is given an initial displacement at one side. Second, the model starts free vibration.

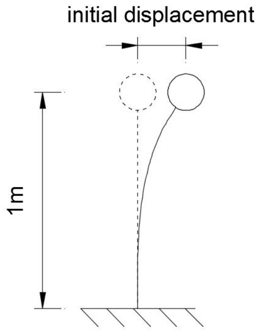


Figure 407.1: Problem Description for Newmark Method

407.2.1.2 Newmark Results

With damping, the displacement peak is smaller and smaller. The displacement at the top is

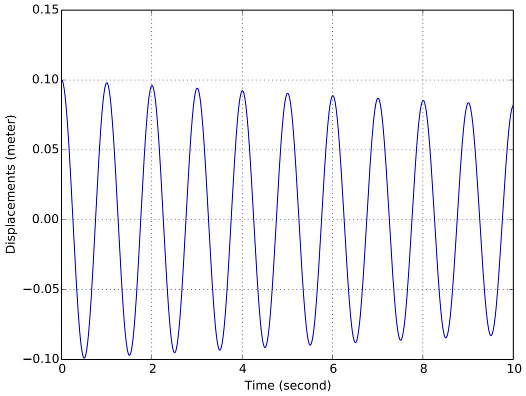


Figure 407.2: Results for Newmark Method

407.2.2 Dynamics: Hilber-Hughes-Taylor (α) Method

407.2.2.1 HHT Model Description

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Firstly, the model is given an initial displacement at one side. Second, the model starts free vibration.

407.2.2.2 HHT Results

With NO damping, the displacement peak keeps the same. The displacement at the top is

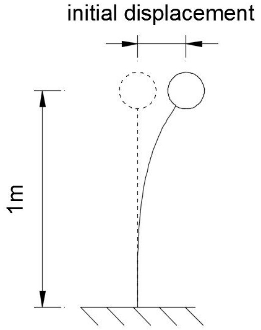


Figure 407.3: Problem Description for HHT Method

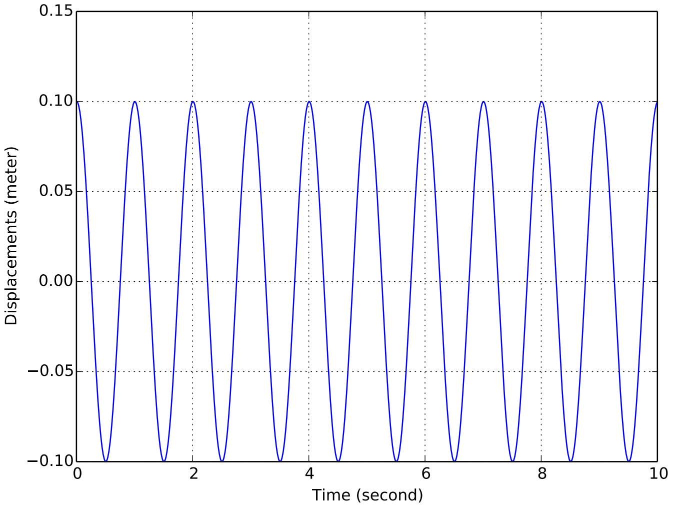


Figure 407.4: Results for HHT Method

407.3 Dynamics: Solution Advancement: Time Step Size

407.3.1 Dynamics: Solution Advancement: Equal Time Step

407.3.1.1 Model Description

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

The model is given an earthquake input motion at the bottom with equal time step. After the wave propagation, the motion at the top is recorded.

407.3.1.2 Results

The input motion is on the left, while the output motion is on the right.

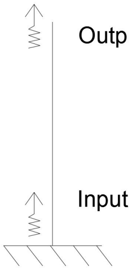


Figure 407.5: Problem Description for Solution Advancement

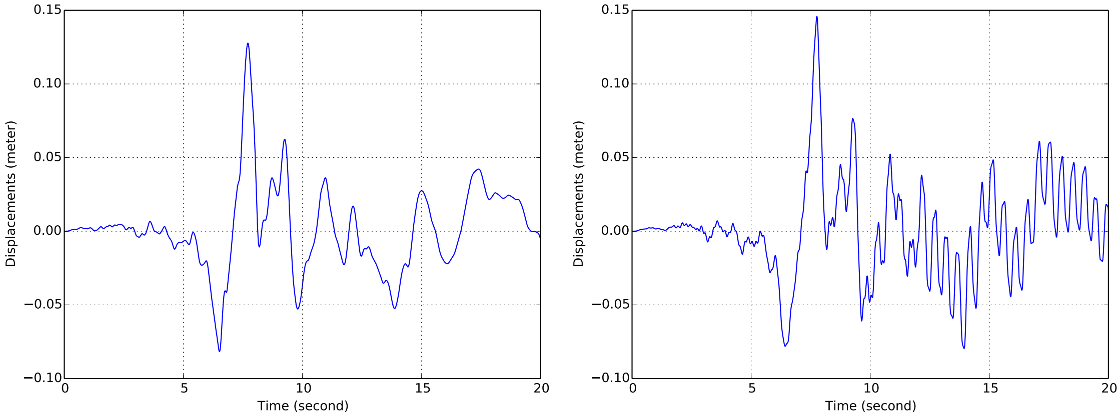


Figure 407.6: Input Motion (Left) And Output Motion (Right)

407.3.2 Dynamics Solution Advancement: Variable Time Step

407.3.2.1 Model Description

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

The model is given an earthquake input motion at the bottom with variable time step. After the wave propagation, the motion at the top is recorded.

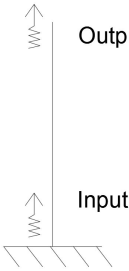


Figure 407.7: Problem Description for Newmark Method

407.3.2.2 Results

The input motion is on the left, while the output motion is on the right. The input motion is in variable time step. As shown in Fig 407.8, from time 10-11 second, the input motion is a straight line (a big time step) without the small time steps.

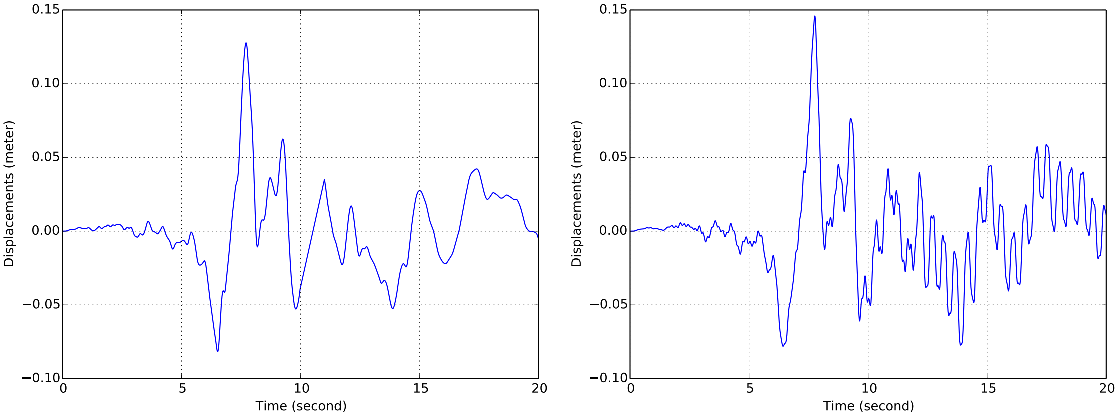


Figure 407.8: Input Motion (Left) And Output Motion (Right)

407.4 Dynamics: Energy Dissipation, Damping

407.4.1 Dynamics: Energy Dissipation: Viscous Damping

407.4.1.1 Dynamics: Energy Dissipation, Viscous Damping: Rayleigh Damping

Model Description The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Firstly, the model is given an initial displacement at the top from 0 to 1 second. Second, after the time 1 second, the model starts free vibration.

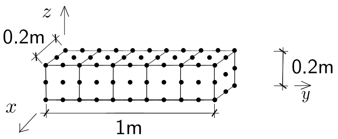


Figure 407.9: Problem Description for Newmark Method

Results This model employs Rayleigh damping. The displacement at the top is

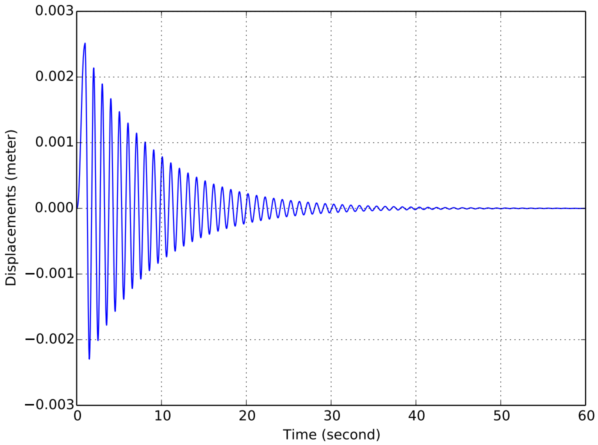


Figure 407.10: Results for Newmark Method

407.4.1.2 Dynamics: Energy Dissipation, Viscous Damping: Caughey Damping

Model Description The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Firstly, the model is given an initial displacement at the top from 0 to 1 second. Second, after the time 1 second, the model starts free vibration.

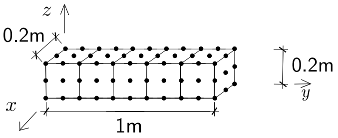


Figure 407.11: Problem Description for Newmark Method

Results This model employs Caughey damping. The displacement at the top is

407.4.2 Dynamics: Energy Dissipation: Material (Elastic-Plastic, Hysteretic) Damping

407.4.2.1 Dynamics: Energy Dissipation, Material Damping: Elastic Perfectly Plastic Models

Model Description The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

The model is a one-element solid brick example with perfectly plastic materials.

Results The Hysteretic loop at the Gauss point is

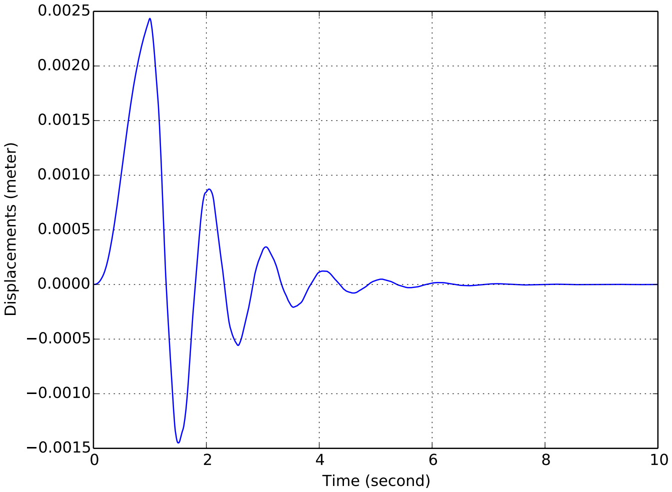


Figure 407.12: Results for Newmark Method

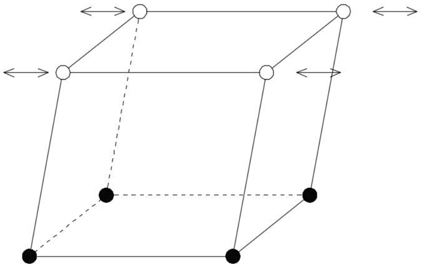


Figure 407.13: Problem Description for Newmark Method

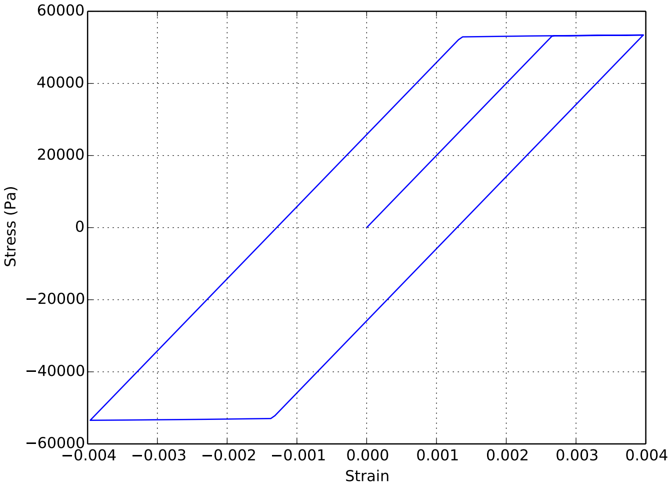


Figure 407.14: Results for Newmark Method

407.4.2.2 Dynamics: Energy Dissipation, Material/Hysteretic Damping: Elastic Plastic Isotropic Hardening Models

Model Description The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

The model is a one-element solid brick example with isotropic hardening materials.

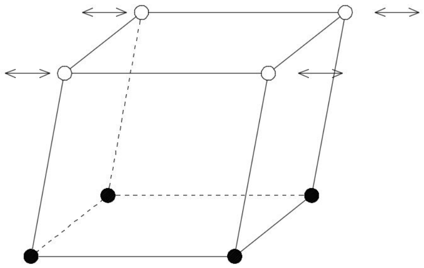


Figure 407.15: Problem Description for Newmark Method

Results The Hysteretic loop at the Gauss point is

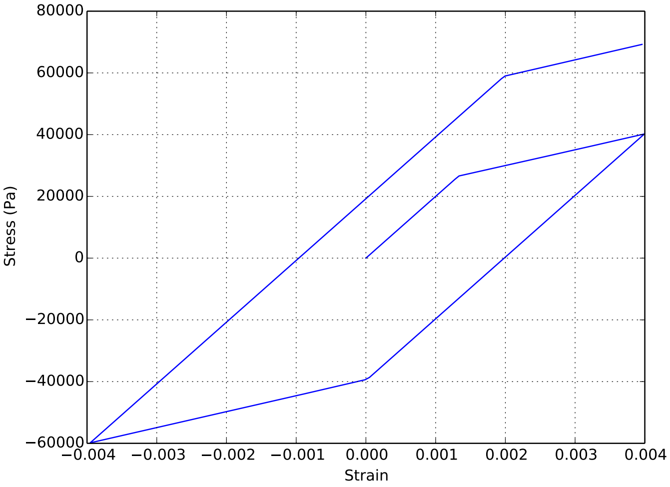


Figure 407.16: Results for Newmark Method

407.4.2.3 Dynamics: Energy Dissipation, Material/Hysteretic Damping: Elastic Plastic Kinematic Hardening Models

Model Description The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

The model is a one-element solid brick example with kinematic hardening materials.

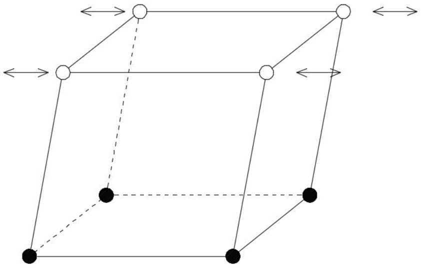


Figure 407.17: Problem Description for Newmark Method

Results The Hysteretic loop at the Gauss point is

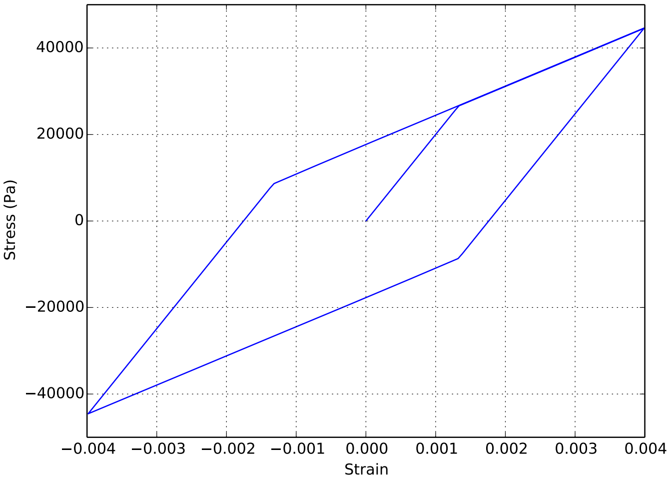


Figure 407.18: Results for Newmark Method

407.4.2.4 Dynamics: Energy Dissipation, Material/Hysteretic Damping: Elastic Plastic Armstrong-Frederick Models

Model Description The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

The model is a one-element solid brick example with materials with nonlinear hardening Armstrong-Frederick.

Results The Hysteretic loop at the Gauss point is

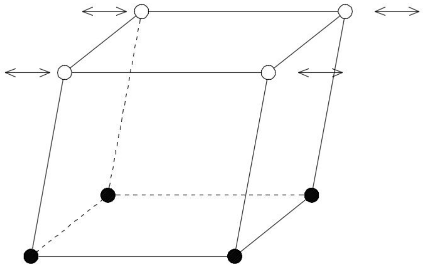


Figure 407.19: Problem Description for Newmark Method

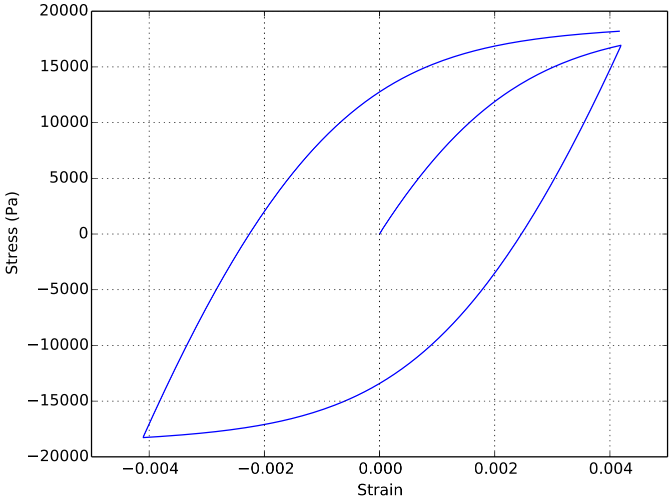


Figure 407.20: Results for Material Armstrong-Frederick

407.4.3 Dynamics: Energy Dissipation: Numerical Damping

407.4.3.1 Energy Dissipation, Numerical Damping: Newmark Method

Model Description The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Firstly, the model is given an initial displacement in the first loading stage. In the second loading stage, the model starts free vibration.

Results This model employs Newmark numerical damping. The displacement at the top in the second loading stage is

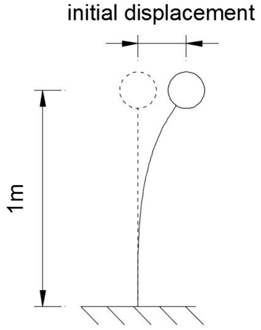


Figure 407.21: Problem Description for Newmark Method

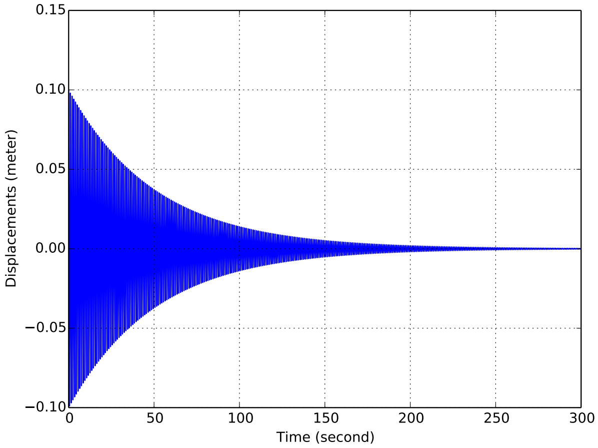


Figure 407.22: Results for Newmark Method

407.4.3.2 Dynamics: Energy Dissipation, Numerical Damping: Hilber-Hughes-Taylor (α) Method

Model Description The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Firstly, the model is given an initial displacement in the first loading stage. In the second loading stage, the model starts free vibration.

Results This model employs HHT numerical damping. The displacement at the top in the second loading stage is

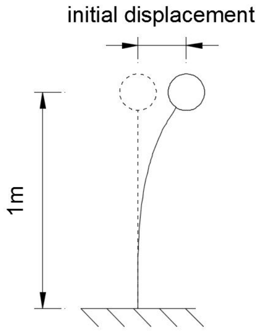


Figure 407.23: Problem Description

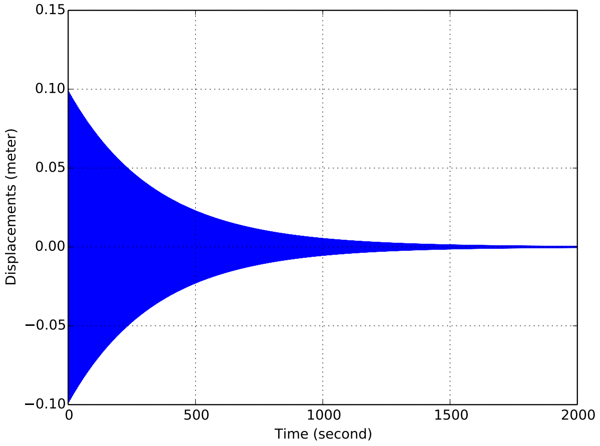


Figure 407.24: Results for HHT Method

407.5 Dynamics: Elastic Solid Dynamic Examples

407.5.1 Model Description

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Firstly, the model is given an initial displacement at the top from 0 to 1 second. Second, after the time 1 second, the model starts free vibration.

407.5.2 Results

This model employs Caughey damping. The displacement at the top is

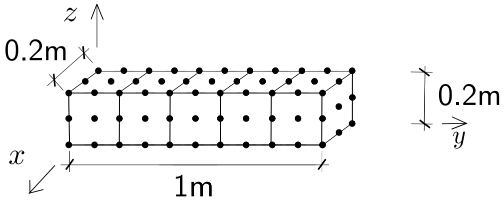


Figure 407.25: Problem Description for Newmark Method

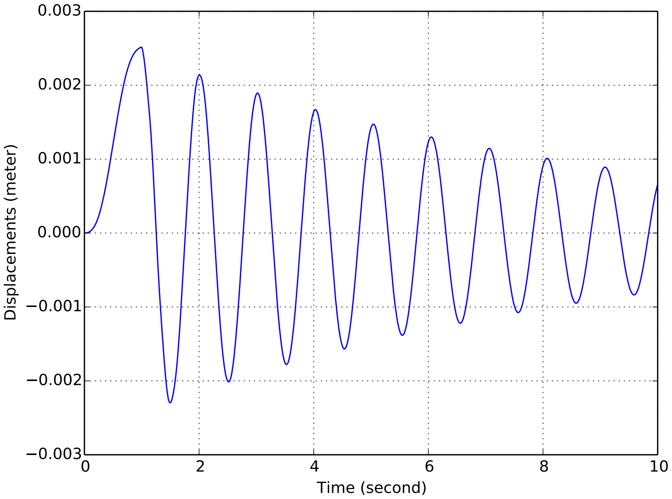


Figure 407.26: Results for Newmark Method

407.6 Dynamics: Elastic Structural Dynamic Examples

407.6.1 Model Description

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Firstly, the model is given an initial displacement in the first loading stage. In the second loading stage, the model starts free vibration.

407.6.2 Results

With NO damping, the displacement peak keeps the same. The displacement at the top is

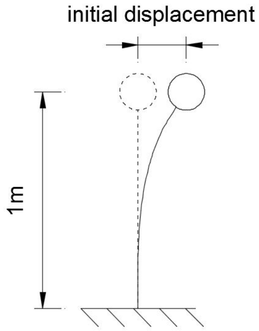


Figure 407.27: Problem Description for Newmark Method

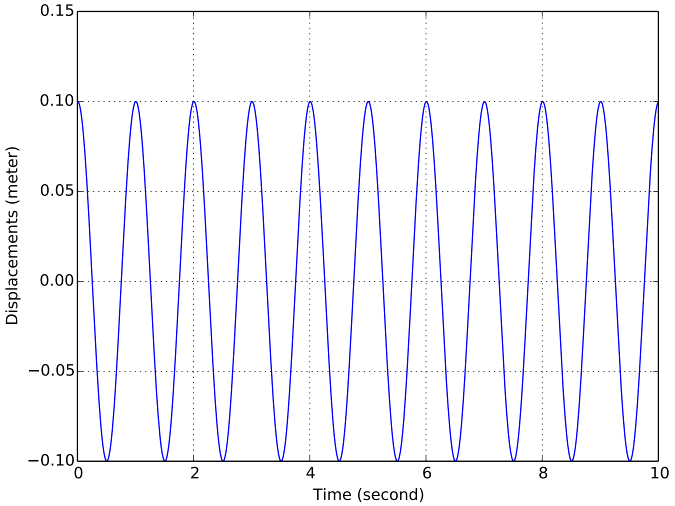


Figure 407.28: Results for Newmark Method

407.7 Dynamics: Interface/Contact Elements

407.7.1 Dynamics: Hard Interface/Contact: One Bar Normal Interface/Contact Dynamics

407.7.1.1 Model Description

This is an example of a ball, bouncing on a solid flat surface. There is only normal contact/interface between the ball and the floor. An upward force is first applied to the concentrated mass lifting it up by $0.1m$ and then the force is removed, resulting in free vibration of the ball. An illustrative diagram of the problem is shown below.

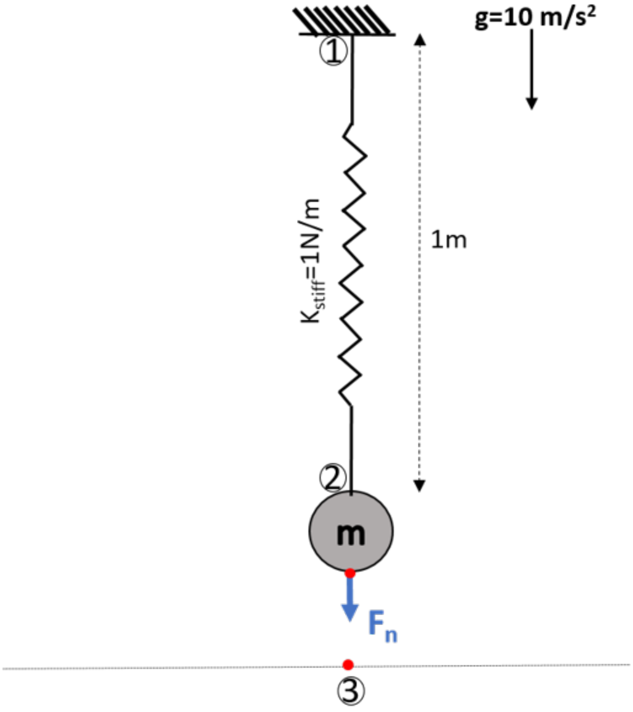


Figure 407.29: Illustration of one bar normal contact/interface dynamics.

The same example can be modeled with different contact/interface and simulation parameters as shown below. For all the different cases shown below, no numerical damping is applied. Only the contact parameters are changed to expose their functionality. The response of node 2 is plotted for all the cases.

407.7.1.2 Dynamics: No Viscous Damping

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Results Here, no viscous damping between the contact/interface pair nodes is applied. The displacement output of *Node 2* is shown below.

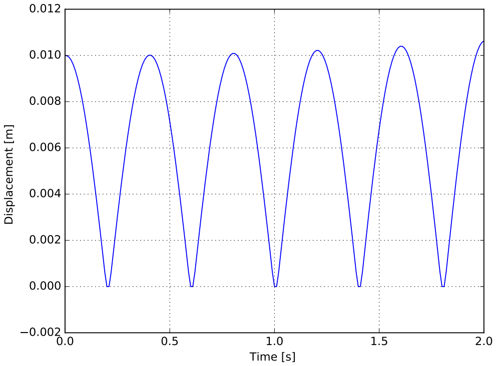


Figure 407.30: Displacement of Node 2

407.7.1.3 Dynamics: Normal Viscous Damping Between Interface/Contact Node Pairs

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Results Viscous damping between the contact/interface pair nodes is applied in normal contact/interface direction. The displacement output of *Node 2* is shown below.

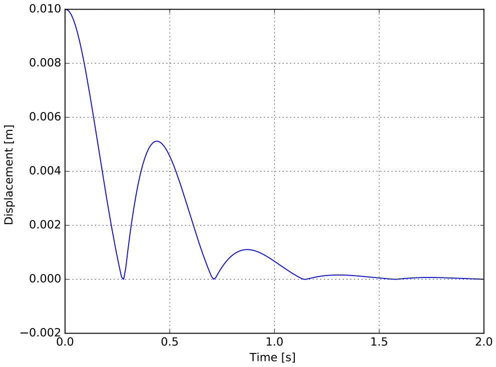


Figure 407.31: Displacement of Node 2

407.7.1.4 Dynamics: Explicit Simulation

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Results With no viscous damping, the analysis is run explicitly without any convergence check. The displacement output of *Node 2* is shown below.

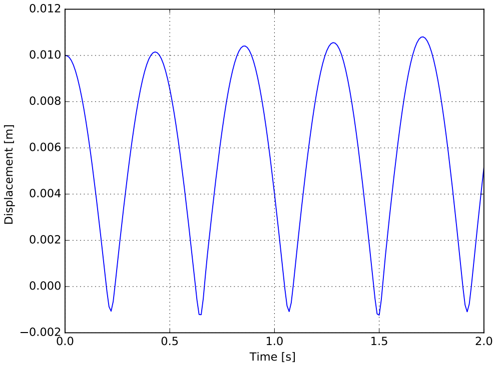


Figure 407.32: Displacement of Node 2

407.7.2 Dynamics: Hard Interface/Contact: Frictional Single Degree of Freedom Problem

Model Description This is an example of a block on a rough surface under gravity. It has been attached to a spring at one end. At the other end a tangential load is applied greater than the coulomb friction and is then removed. The block oscillates back and forth with continuously loosing energy because of frictional force and then stops, with some permanent deformation. This kind of damping is called frictional damping which is linear as compared to exponential in case of viscous damping. An illustrative diagram of the problem is shown below.

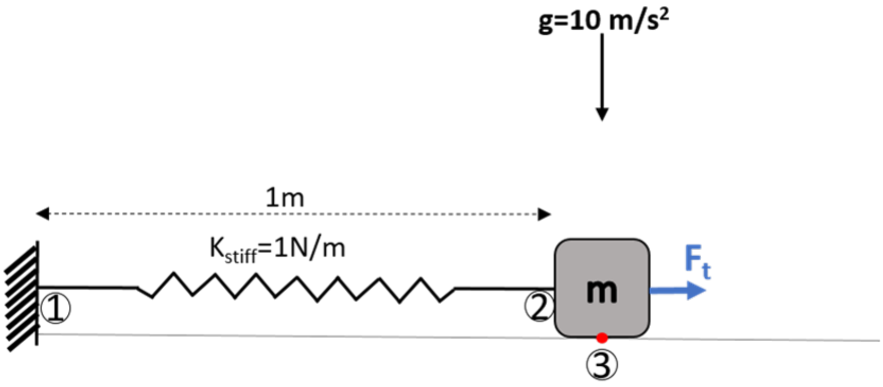


Figure 407.33: Illustration of frictional single degree of freedom problem

The same example can be modeled with different contact/interface and simulation parameters as shown below. For all the different cases shown below, no numerical damping is applied. Only the contact/interface parameters are changed to expose their functionality. The response of node 2 is plotted for all the cases.

407.7.2.1 Dynamics: No Viscous Damping

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Results In this examples, no viscous damping between the contact/interface pair nodes is applied. The displacement output of *Node 2* is shown below.

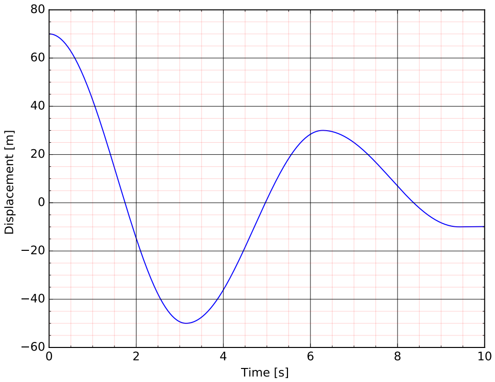


Figure 407.34: Displacement of Node 2

407.7.2.2 Dynamics: Tangential Viscous Damping Between Interface/Contact Node Pairs

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Results Viscous damping between the contact/interface pair nodes is applied in tangential contact/interface direction. The displacement output of *Node 2* is shown below.

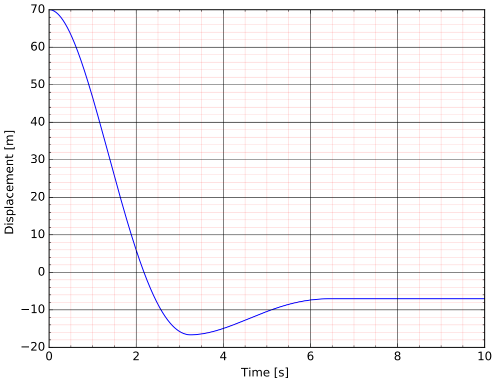


Figure 407.35: Displacement of Node 2

407.7.2.3 Dynamics: Explicit Simulation

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Results With no viscous damping, the analysis is run explicitly without any convergence check. The displacement output of *Node 2* is shown below.

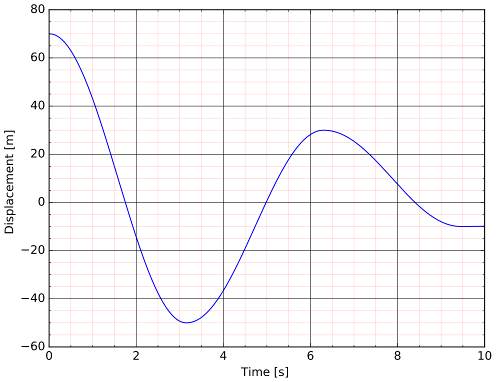


Figure 407.36: Displacement of Node 2

407.7.3 Dynamics: Soft Interface/Contact: One Bar Normal Interface/Contact Dynamics

Model Description This is an example of a ball, bouncing on a solid flat surface. There is only normal contact/interface between the ball and the floor. An upward force is first applied to the concentrated mass lifting it up by $0.1m$ and then the force is removed, resulting in free vibration of the ball. An illustrative diagram of the problem is shown below.

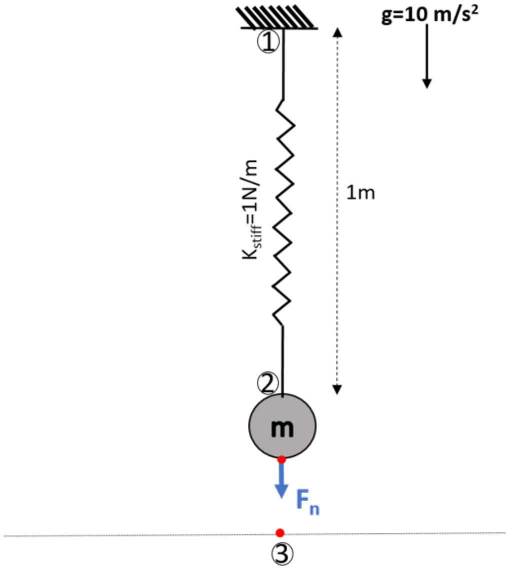


Figure 407.37: Illustration of one bar normal contact/interface dynamics

The same example can be modeled with different contact/interface and simulation parameters as shown below. For all the different cases shown below, no numerical damping is applied. Only the contact/interface parameters are changed to expose their functionality. The response of node 2 is plotted for all the cases.

407.7.3.1 Dynamics: No Viscous Damping

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Results In this example, no viscous damping between the contact/interface pair nodes is applied. The displacement output of *Node 2* is shown below.

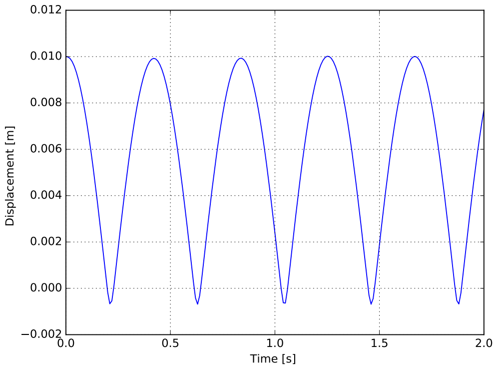


Figure 407.38: Displacement of Node 2

407.7.3.2 Dynamics: With Normal Viscous Damping Between Interface/Contact Node Pairs

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Results Viscous damping between the contact/interface pair nodes is applied in normal contact/interface direction. The displacement output of *Node 2* is shown below.

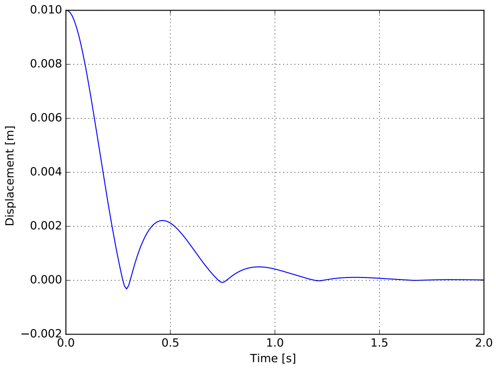


Figure 407.39: Displacement of Node 2

407.7.3.3 Dynamics: Explicit Simulation

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Results With no viscous damping, the analysis is run explicitly without any convergence check. The displacement output of *Node 2* is shown below.

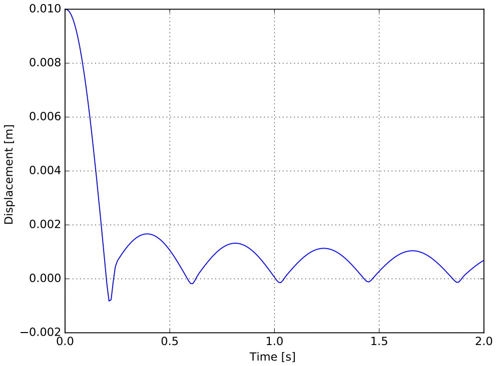


Figure 407.40: Displacement of Node 2

407.7.4 Dynamics: Soft Interface/Contact: Frictional Single Degree of Freedom Problem

Model Description This is an example of a block on a rough surface under gravity. It has been attached to a spring at one end. At the other end a tangential load is applied greater than the coulomb friction and is then removed. The block oscillates back and forth with continuously loosing energy because of frictional force and then stops, with some permanent deformation. This kind of damping is called frictional damping which is linear as compared to exponential in case of viscous damping. An illustrative diagram of the problem is shown below.

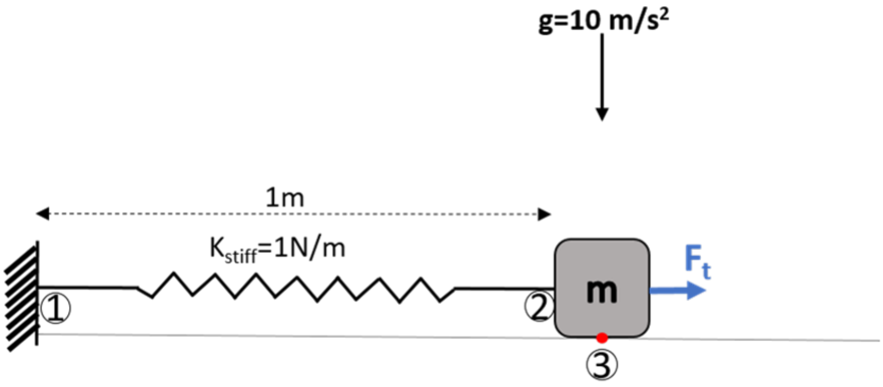


Figure 407.41: Illustration of frictional single degree of freedom problem

The same example can be modeled with different contact/interface and simulation parameters as shown below. For all the different cases shown below, no numerical damping is applied. Only the contact/interface parameters are changed to expose their functionality. The response of node 2 is plotted for all the cases.

407.7.4.1 Dynamics: No Viscous Damping

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Results In this example, no viscous damping between the contact/interface pair nodes is applied. The displacement output of *Node 2* is shown below.

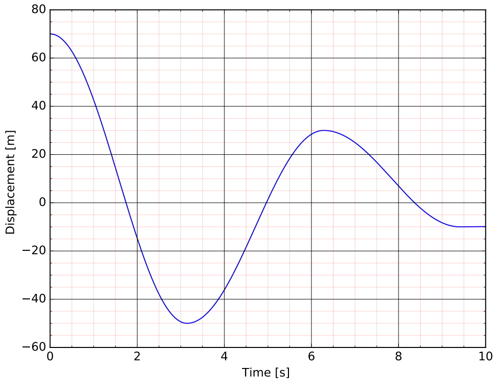


Figure 407.42: Displacement of Node 2

407.7.4.2 Dynamics: Tangential Viscous Damping Between Interface/Contact Node Pairs

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Results Viscous damping between the contact/interface pair nodes is applied in tangential contact/interface direction. The displacement output of *Node 2* is shown below.

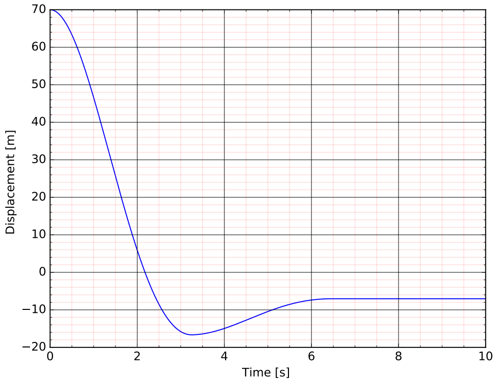


Figure 407.43: Displacement of Node 2

407.7.4.3 Dynamics: Explicit Simulation

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Results With no viscous damping, the analysis is run explicitly without any convergence check. The displacement output of *Node 2* is shown below.

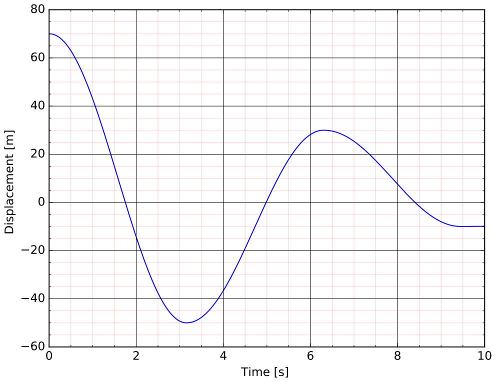


Figure 407.44: Displacement of Node 2

407.7.5 Dynamics: Split Beam

407.7.5.1 Model Description

In this example, a normal beam is split into two halves along its depth. A uniform surface load of 50 Pa is applied to the top half of the beam, pulling it away from its lower part. Then, the load is removed, to allow free vibration between the split beams. An illustrative diagram of the problem is shown below.

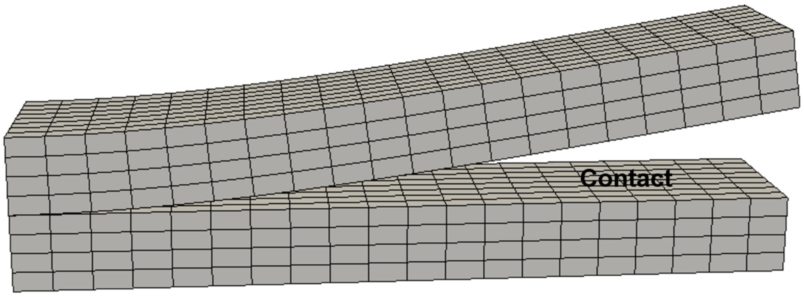


Figure 407.45: Illustration of Split Beam Analysis

The same example was modelled with soft and hard contact/interface. Numerical as well as viscous damping between contact/interface pair nodes was applied. The displacement response of the extreme right mid node of top half beam is plotted.

407.7.5.2 Dynamics: Split Beam With Hard Interface/Contact

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

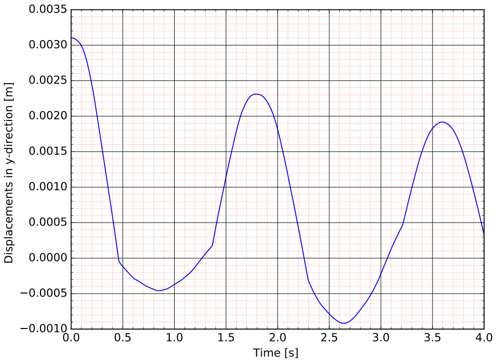


Figure 407.46: Displacement response of extreme mid node of top half beam

407.7.5.3 Dynamics: Split Beam With Soft Interface/Contact

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

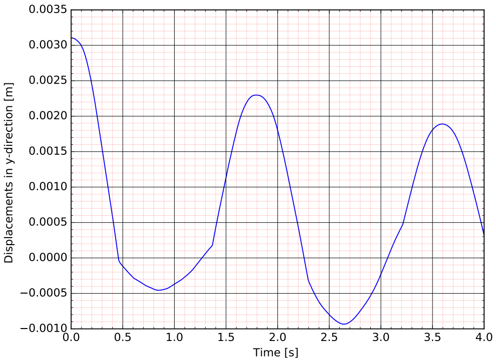


Figure 407.47: Displacement response of extreme mid node of top half beam

407.7.6 Dynamics: Block on Soil ESSI

Model Description The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

A solid block is placed in the soil. There is contact/interface between the interface of solid and the soil. First, self-weight and then a uniform acceleration in x-direction is applied to the whole model. This analysis would provide relative displacement, velocity and acceleration response for the given shaking. An illustrative diagram of the problem is shown below.

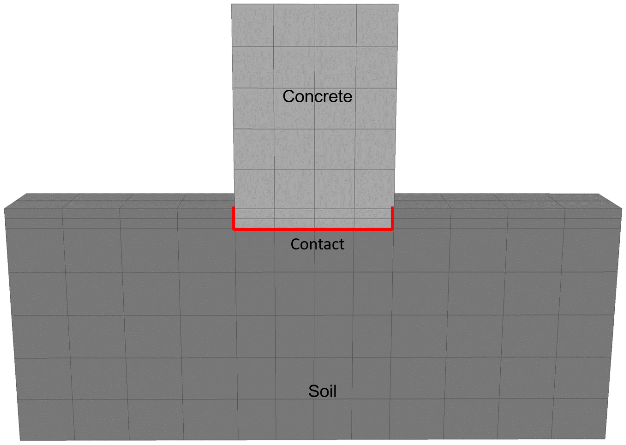


Figure 407.48: Illustration of frictional single degree of freedom problem

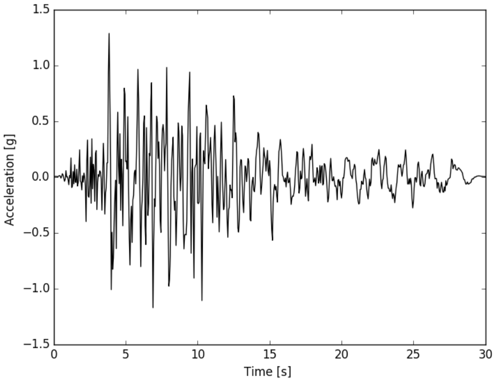


Figure 407.49: Applied Motion

Results Displacement response of the top of the solid block is shown below. Numerical Damping, Raleigh damming and viscous damping between contact/interface node pairs are applied.

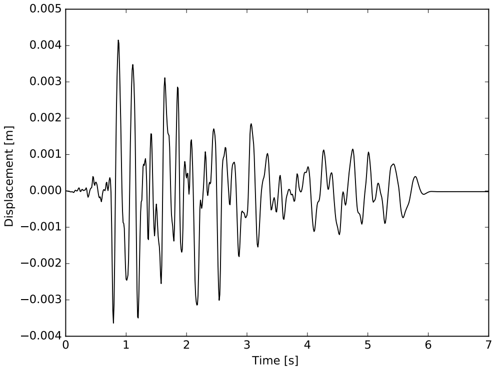
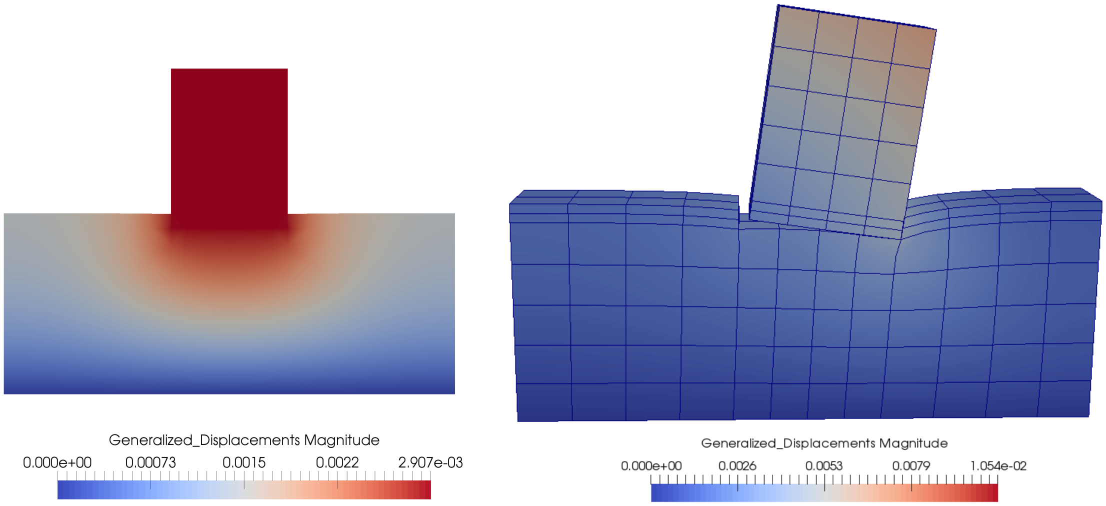


Figure 407.50: Displacement response at the top of the block



(a) Self Wight Stage

(b) Uniform Acceleration Stage at 0.85 sec

Figure 407.51: Simulation results visualization

407.8 Dynamics: Inelastic Solid Examples

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Firstly, the model is given an initial displacement at the top from 0 to 1 second. Second, after the time 1 second, the model starts free vibration.

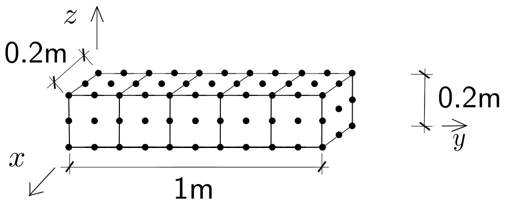


Figure 407.52: Problem Description for Newmark Method

Results This model has material damping. The displacement at the top is

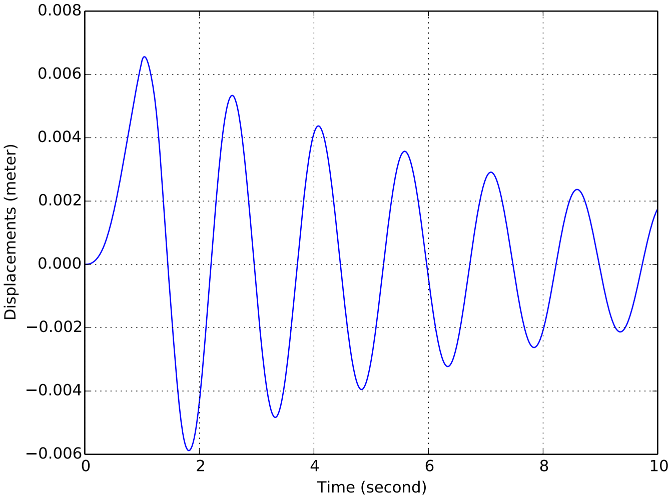


Figure 407.53: Results for Newmark Method

407.9 Dynamics: Inelastic Structural Examples

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

The column beam is represented by the fiber section. This example is under the dynamic load of ground motion.

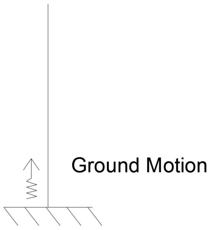


Figure 407.54: Ground motion on the Fiber Beam with Column Section

The fiber represents the rebar. The section of all fibers represents the cross section properties of the inelastic beam.

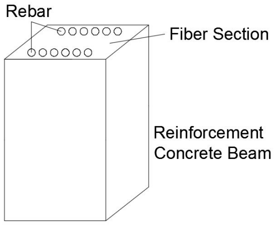


Figure 407.55: Diagram of the Fiber Beam with Column Section

407.10 Dynamics: Domain Reduction Method (DRM)

407.10.1 Dynamics: DRM One Dimensional (1D) Model

The Real-ESSI input files with 8NodeBrick for this example are available

HERE.

The same model for this example with 27NodeBrick is available

HERE.

A simple 1D DRM model is shown in Fig.(707.44). The "DRM element", "Exterior node" and "Boundary node" are required to be designated in the DRM HDF5 input. The format and script for the HDF5 input is available in DSL/input manual.

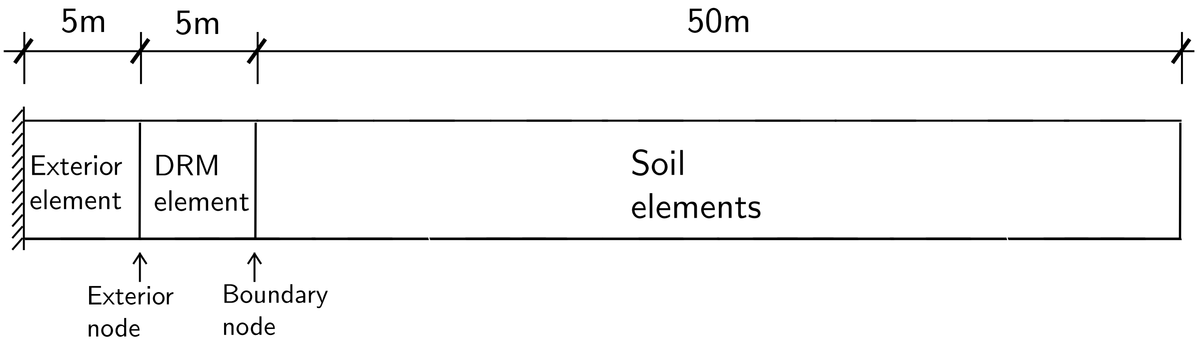


Figure 407.56: 1D DRM model.

Numerical model

Long 1D DRM model 1000:1 The Real-ESSI input files for this example are available

HERE.

The results can also be seen from this

ANIMATION.

To show the wave propagation explicitly, a long 1D model (1000:1) similar to the 1D DRM model above was made in this section.

The model description is same to Fig.(707.44) except this model use far more soil elements.

The general view is shown in Fig.(707.46) below.

There is still now outgoing waves at the exterior layers, which is shown in Fig(707.47).

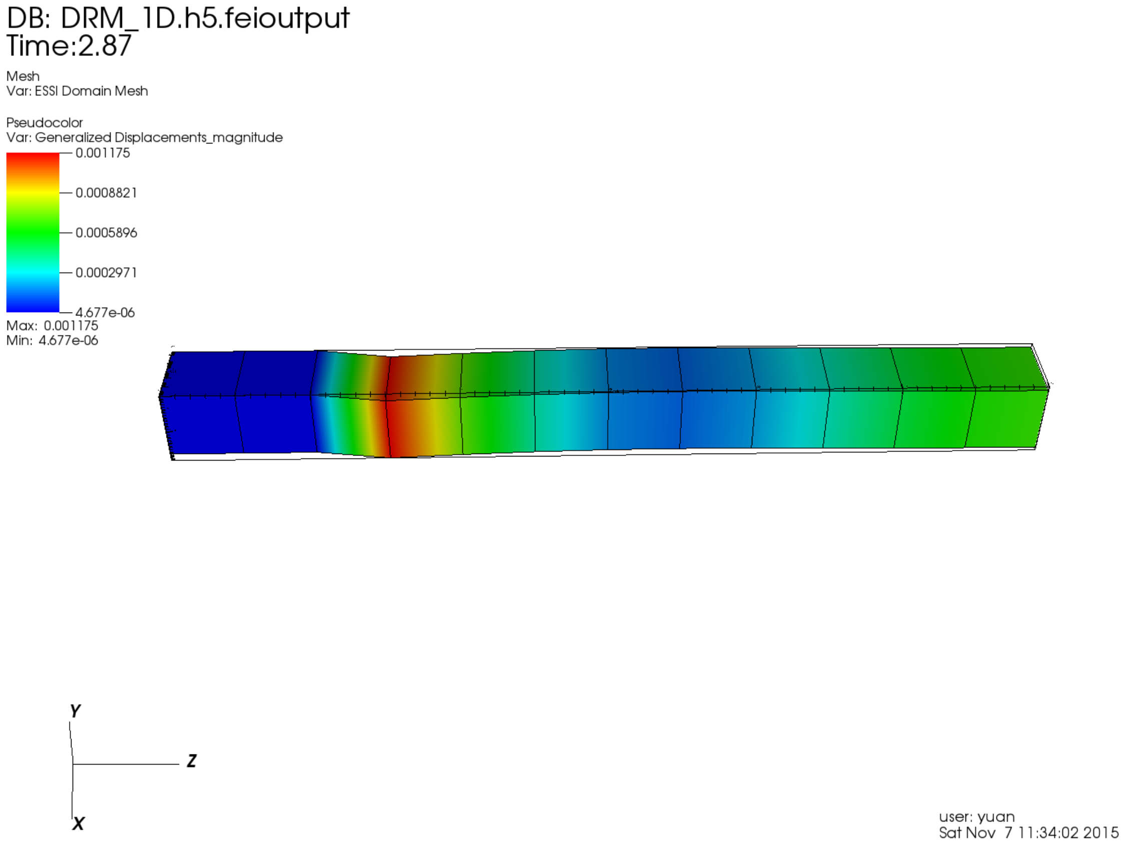


Figure 407.57: 1D DRM model.

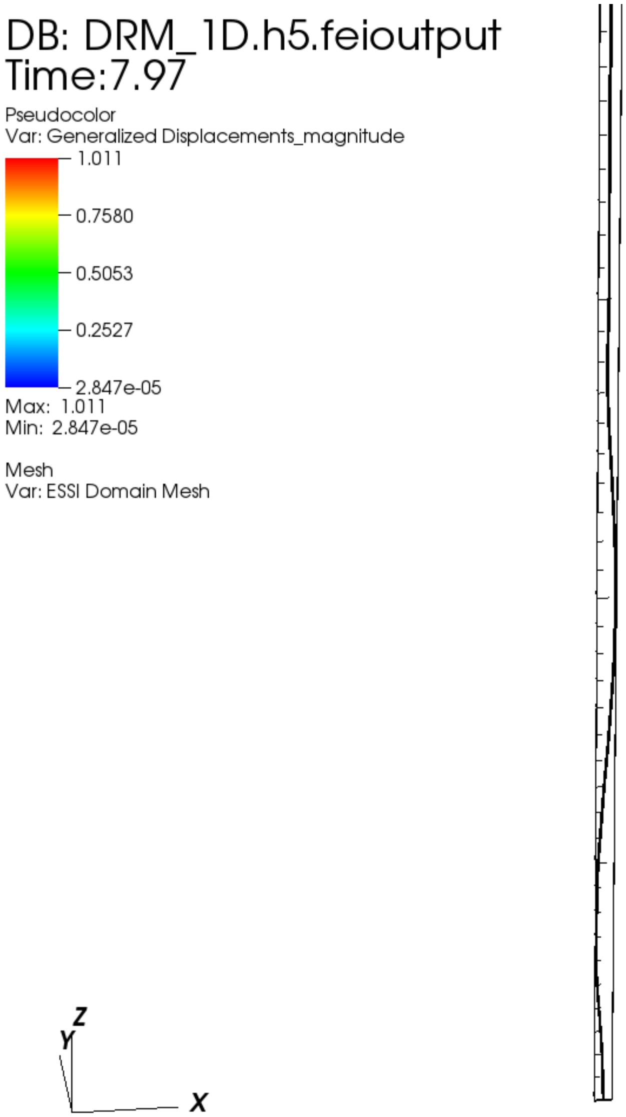


Figure 407.58: Long 1D DRM model

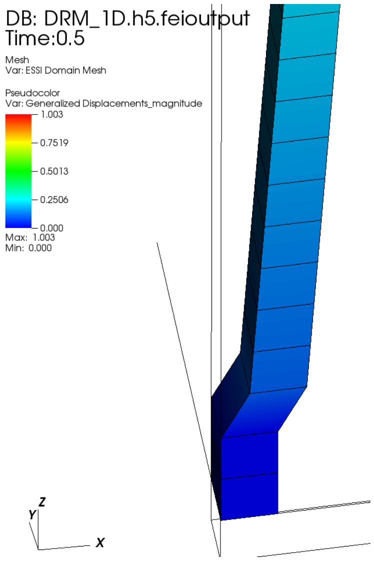


Figure 407.59: Long 1D DRM model: exterior layer

407.10.2 Dynamics: Three Dimensional (3D) DRM Model

The Real-ESSI input files with 8NodeBrick for this example are available

HERE.

The same model for this example with 27NodeBrick is available

HERE.

As shown in Fig.(707.48), the DRM layer is used to add the earthquake motion.

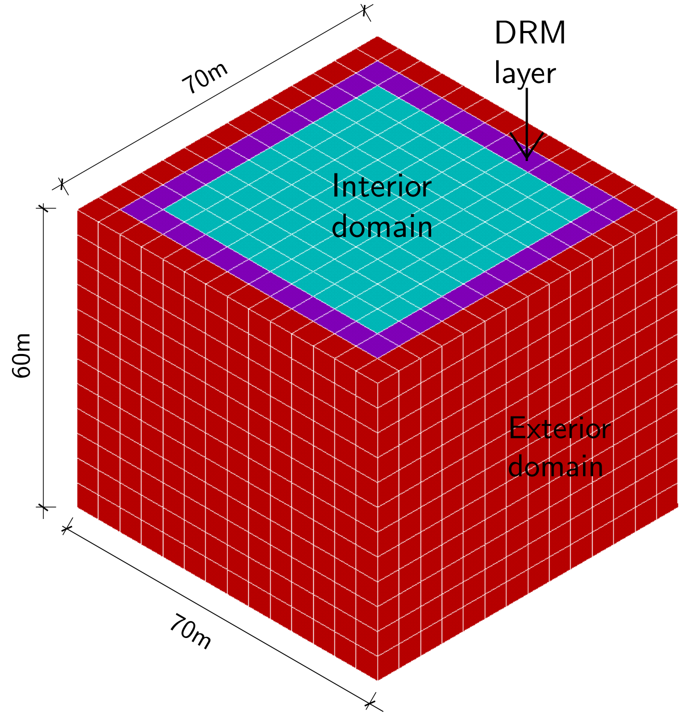


Figure 407.60: The diagram for 3D Domain Reduction Method example.

Numerical result

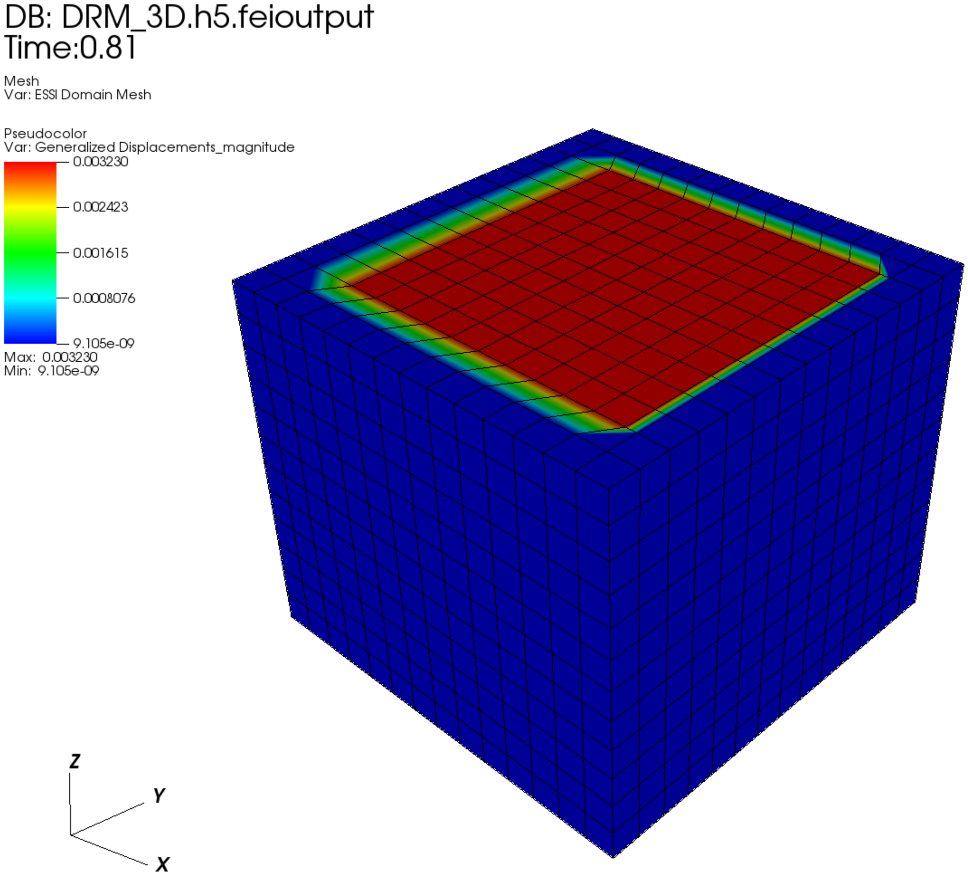


Figure 407.61: Diagram for the 3D DRM model.

407.10.3 Dynamics: DRM Model with Structure

Problem description The Real-ESSI input files for this example are available HERE.

The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

As shown in Fig.(407.62), the structure is placed in the middle. Five different materials are assigned to structure, contact/interface zones, soil, DRM layer, and damping layers, respectively.

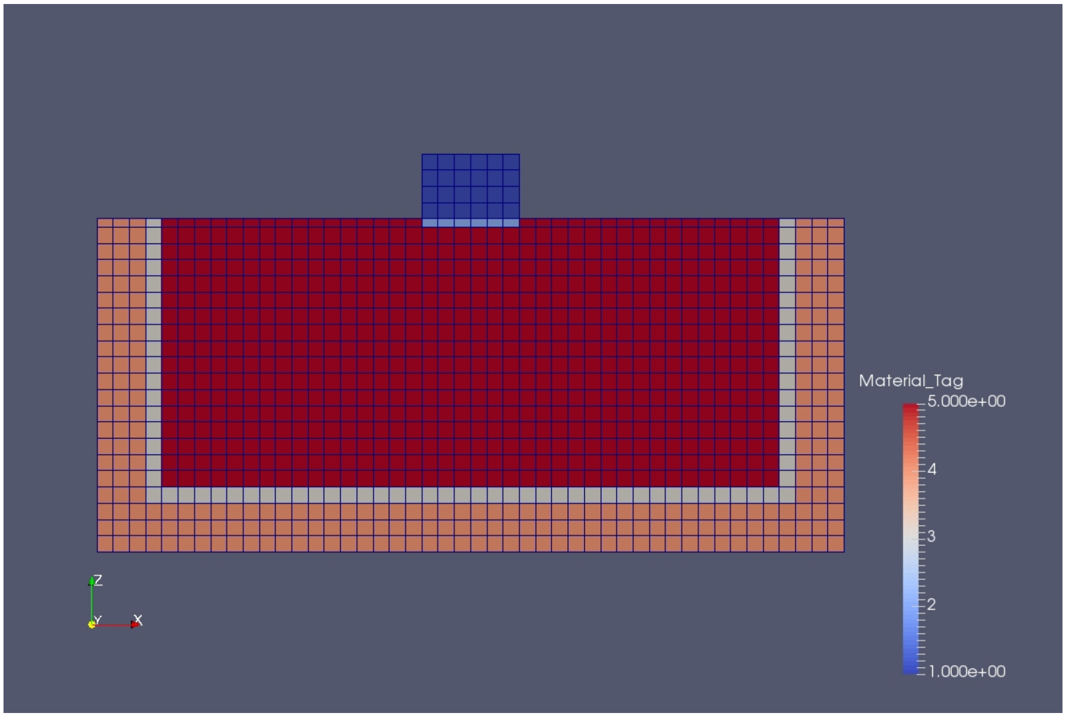


Figure 407.62: A Domain Reduction Method example with a Simple Structure.

407.11 Dynamics: Eigen Analysis

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files and postprocessing results for this example is available HERE.

Model is a brick beam with distributed mass.

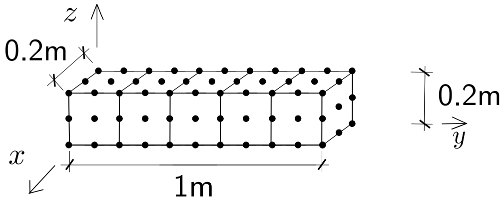


Figure 407.63: Problem Description for Newmark Method

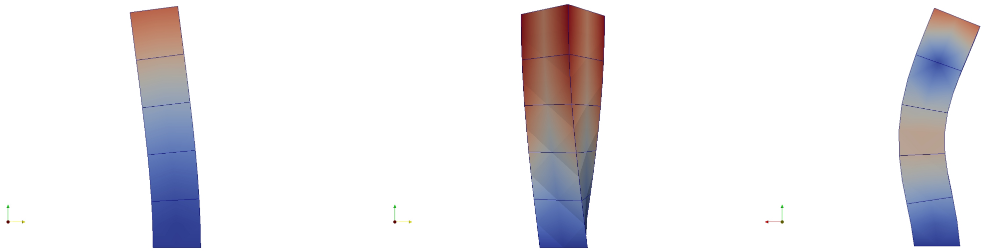


Figure 407.64: Solid Brick Cantilever Eigen Mode 1, 3, 4(From left to Right)

Results

407.12 Dynamics: Fully Coupled u-p-U and u-p Elements

The Real-ESSI input files for coupled example are available HERE.

407.13 Dynamics: Partially Saturated / Unsaturated u-p-U Element (example in development)

407.14 Dynamics: Coupled Interface/Contact Element (example in development)

407.15 Dynamics: Buoyant Forces (example in development)

407.16 Chapter Summary and Highlights

In this Chapter stochastic/probabilistic modeling and simulation is illustrated through a number of examples. These examples can then be analyzed using Real-ESSI Simulator that is available for Linux, Windows (through ESL) or MacOS, and on Amazon Web Services (AWS) computers. Please refer to the Real-ESSI web site `real-essi.us`, for more information on how to install Real-ESSI on your computer (Linux, Windows, MacOS...).

Chapter 408

Stochastic Examples

(2018-2019-2020-2021-)

(In collaboration with Dr. Hexiang Wang)

408.1 Probabilistic Constitutive Modeling

408.1.1 Probabilistic Constitutive Modeling: Linear Elastic

The model description:

The Real-ESSI input files for this example are available in a zip archive HERE.

A stochastic uniaxial elastic material with lognormal distributed random elastic modulus, mean 155 MPa and coefficient of variation 30%.

Results:

The probabilistic stress strain response of the stochastic uniaxial elastic material is shown in Figure 408.1.

408.1.2 Probabilistic Constitutive Modeling: Elasto-Plastic

The model description:

The Real-ESSI input files for this example are available in a zip archive HERE.

A stochastic uniaxial elastoplastic material with vanishing elastic region and nonlinear Armstrong-Frederick kinematic hardening rule is modeled. The model parameters are: Armstrong-Frederick pa-

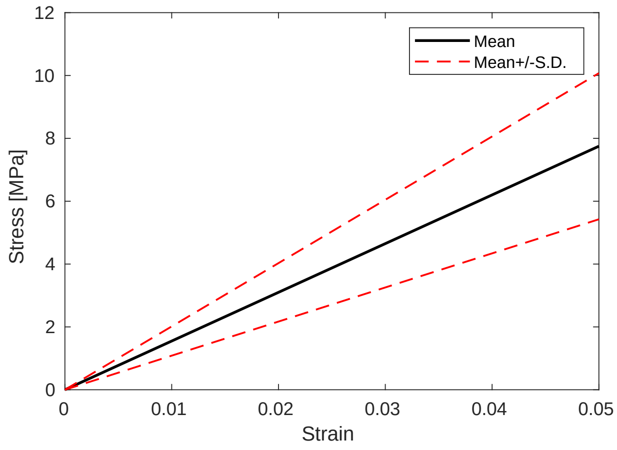


Figure 408.1: Constitutive behavior of stochastic uniaxial elastic material.

rameter H_a follows lognormal distribution with marginal mean 12 MPa and coefficient of variation 20%. Armstrong-Frederick parameter C_r follows lognormal distribution with marginal mean 200 and coefficient of variation (CV) 20%.

Results:

The probabilistic stress strain response of the stochastic uniaxial elastic material is shown in Figure 408.2.

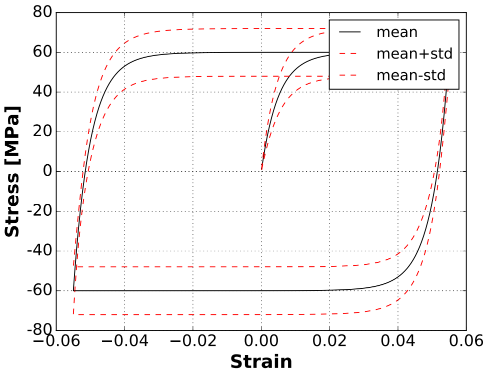


Figure 408.2: Constitutive behavior of stochastic uniaxial elastoplastic material.

408.2 Probabilistic Characterization of Seismic Motions

The model description:

The Real-ESSI input files for this example are available in a zip archive HERE.

For stochastic analysis with uncertain seismic excitations, it is important to characterize input uncertain motions as a non-stationary random process. The random process can be quantified through marginal mean, marginal standard deviation and correlation structure, and can be represented as Hermite polynomial chaos (PC). This example presents such a random process of seismic motions with marginal mean, marginal standard deviation and correlation structure defined through plain text files. It is noted that this random process is used as input bedrock excitations in the subsequent stochastic wave propagation analysis.

Results:

It is important to check that the statistics synthesized from PC representation matches well with the input. Figures 408.3 and 408.2 compare the marginal statistics and correlation structure synthesized from PC representation with the target input.

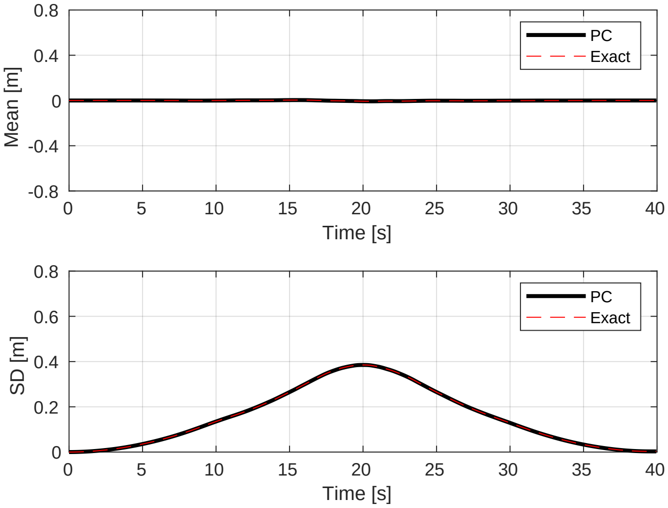


Figure 408.3: Verification of marginal statistics of random process motions.

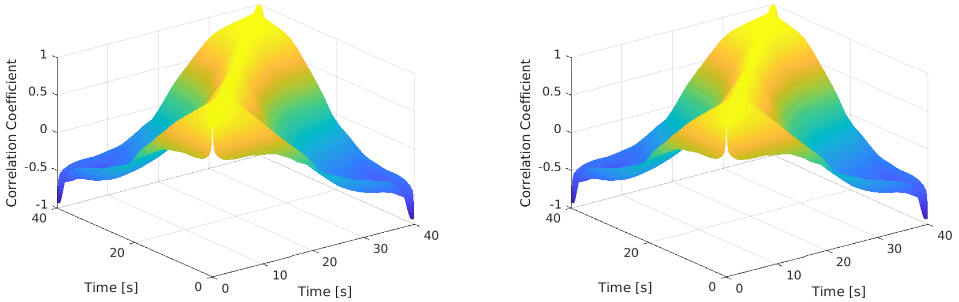


Figure 408.4: Input correlation structure (Left) and PC-synthesized correlation structure (Right).

408.3 1D Stochastic Seismic Wave Propagation

408.3.1 1D Stochastic Seismic Wave Propagation: Linear Elastic

The model description:

The Real-ESSI input files for this example are available in a zip archive HERE.

Presented is 1D stochastic seismic wave propagation through uncertain linear elastic, layered ground. The uncertain motions characterized in section 408.2 is adopted as bedrock input. The ground is 10m thick with three layers and discretized with 10 stochastic shear beam elements as shown in Figure 408.5.

- Layer #1: Thickness 3m, uncertain elastic modulus follows lognormal distribution with marginal mean 120 MPa and 20% coefficient of variation.
- Layer #2: Thickness 3m, uncertain elastic modulus follows lognormal distribution with marginal mean 150 MPa and 25% coefficient of variation.
- Layer #3: Thickness 4m, uncertain elastic modulus follows lognormal distribution with marginal mean 180 MPa and 25% coefficient of variation.

The correlation structure of the uncertain elastic modulus random field follows exponential correlation with correlation length as 10m.

Results:

Time evolving marginal mean and marginal standard deviation of surface probabilistic displacement and acceleration response are shown in Figure 408.6 and 408.7.

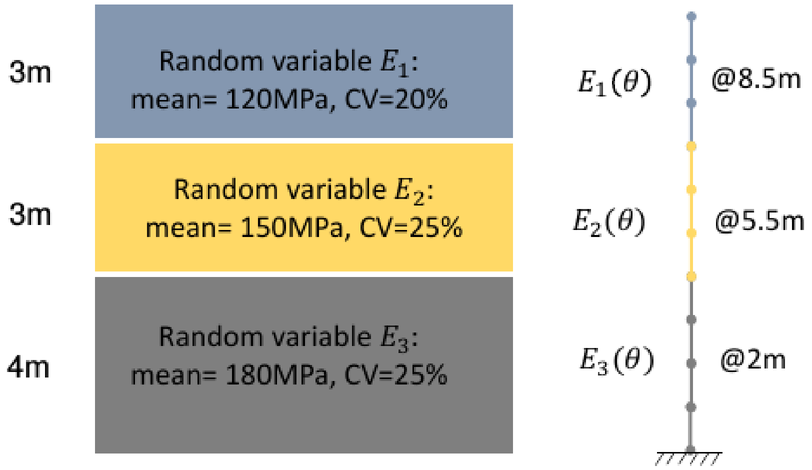


Figure 408.5: 1D layered ground and stochastic shear beam FEM model.

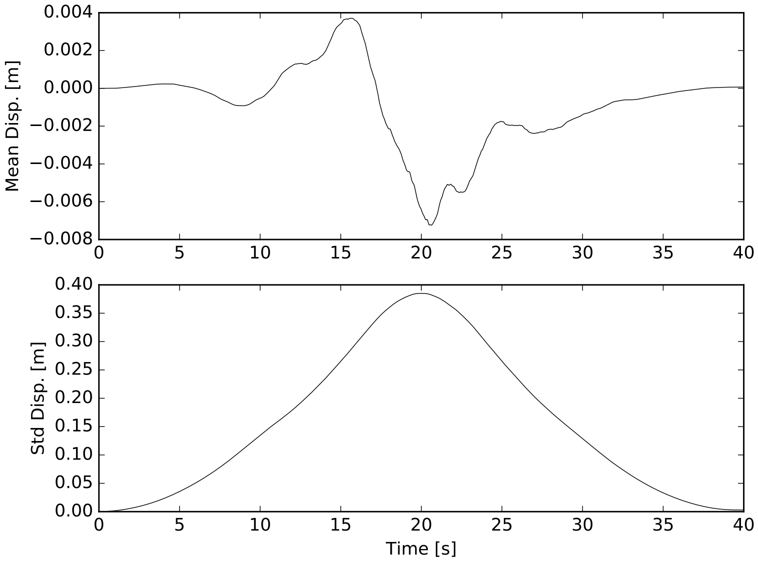


Figure 408.6: Probabilistic displacement response of ground surface.

408.3.2 1D Stochastic Seismic Wave Propagation: Elasto-Plastic

The model description:

The Real-ESSI input files for this example are available in a zip archive HERE.

The model geometry and input seismic excitations are identical to the example in section 408.3.1. The only difference is the constitutive model of soil. In this example, probabilistic elastoplastic soil model with vanishing elastic region and Armstrong-Frederick kinematic hardening is adopted.

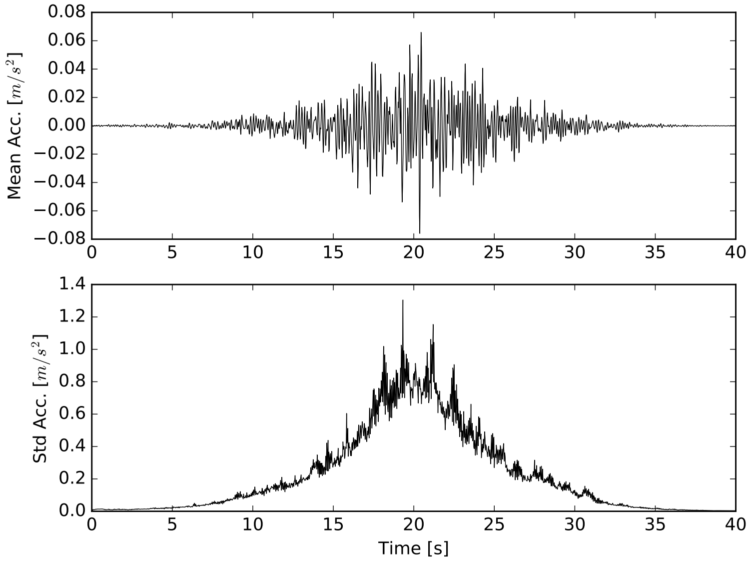


Figure 408.7: Probabilistic acceleration response of ground surface.

Results:

Time evolving marginal mean and marginal standard deviation of surface probabilistic displacement and acceleration response are shown in Figure 408.8 and 408.9.

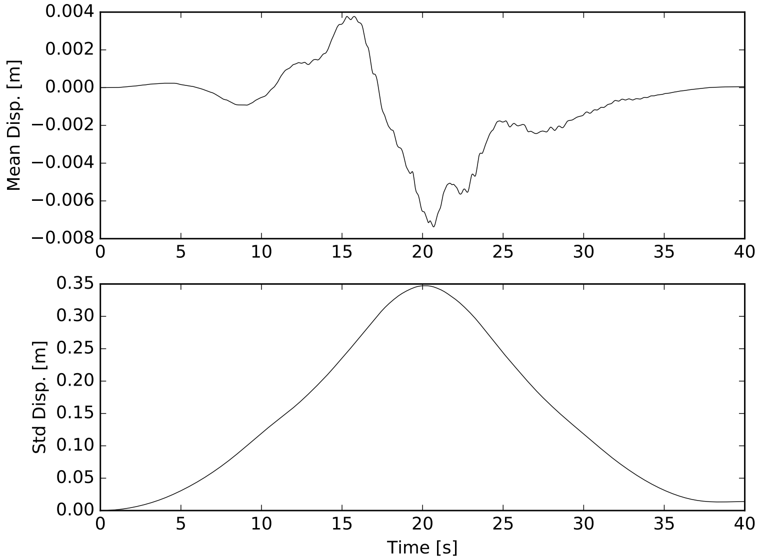


Figure 408.8: Probabilistic displacement response of ground surface.

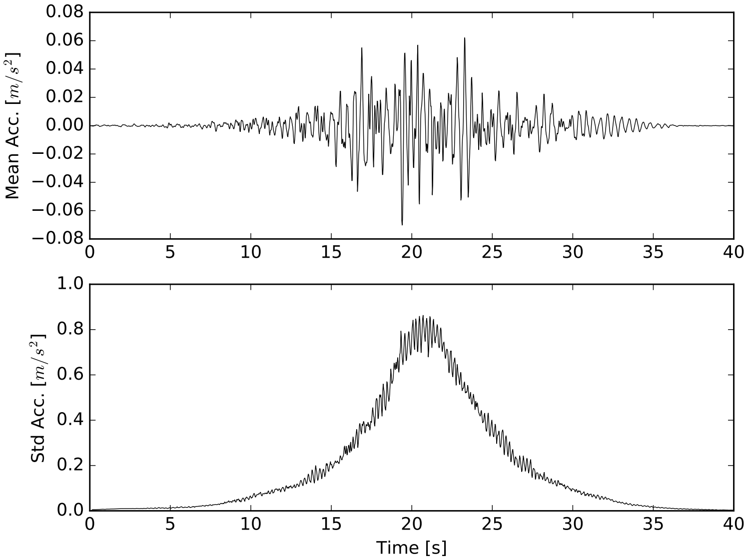


Figure 408.9: Probabilistic acceleration response of ground surface.

408.4 1D Stochastic Seismic Wave Propagation: Sobol Sensitivity Analysis

The Real-ESSI input files for this example are available in a zip archive HERE.

Sobol sensitivity analysis is performed for the stochastic wave propagation example in section 408.3.1. From the sensitivity analysis results for probabilistic response at ground surface, it is shown that for this specific case most of the variance comes from the uncertain input motions.

Chapter 409

Large Scale, Realistic Examples

(2016-2018-)

(In collaboration with Dr. Yuan Feng, Mr. Sumeet Kumar Sinha, Dr. Han Yang, and Dr. Hexiang Wang)

Full scale, realistic examples of statics and dynamics of bridges, dams, buildings and nuclear power plants are presented in part 500 on page 2189 in Jeremić et al. (1989-2025).

Chapter 410

Short Course Examples

(2017-2023-)

(In collaboration with Dr. Yuan Feng and Dr. Han Yang)

410.1 Nonlinear Analysis Steps

410.1.1 Free Field 1C

Elastic Material. The Real-ESSI input files for elastic example are available HERE.

The modeling parameters are listed below:

- Elastic Material Properties
  - Mass Density, ρ, 2000 kg/m^3
  - Shear wave velocity, V_s, 500 m/s
  - Young's modulus, E, 1.1 GPa
  - Poisson's ratio, ν, 0.1

Elastoplastic Material, von Mises with Armstrong-Frederick Kinematic Hardening The Real-ESSI input files for elastoplastic material example are available HERE.

The modeling parameters are listed below

- von-Mises nonlinear hardening material model
  - Mass density, ρ, 2000 kg/m^3
  - Shear wave velocity, V_s, 500 m/s
  - Young's modulus, E, 1.1 GPa
  - Poisson's ratio, ν, 0.1
  - von Mises radius, k, 60 kPa
  - Nonlinear kinematic hardening, H_a, 30 MPa
  - Nonlinear kinematic hardening, C_r, 60
  - Shear strength ($\approx \sqrt{2/3}\, H_a/C_r$), S_u, 408 kPa
  - Isotropic hardening rate, K_{iso}, 0 Pa

Results of the simulation are shown in Fig. 410.1.

The time series of simulation results is shown in Fig. 410.3.

The response spectrum of motion is shown in Fig. 410.4.

Figure 410.1: Simulation model.

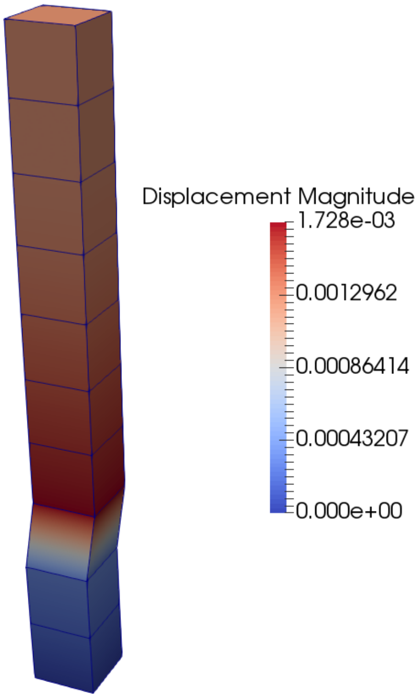


Figure 410.2: Simulation model.

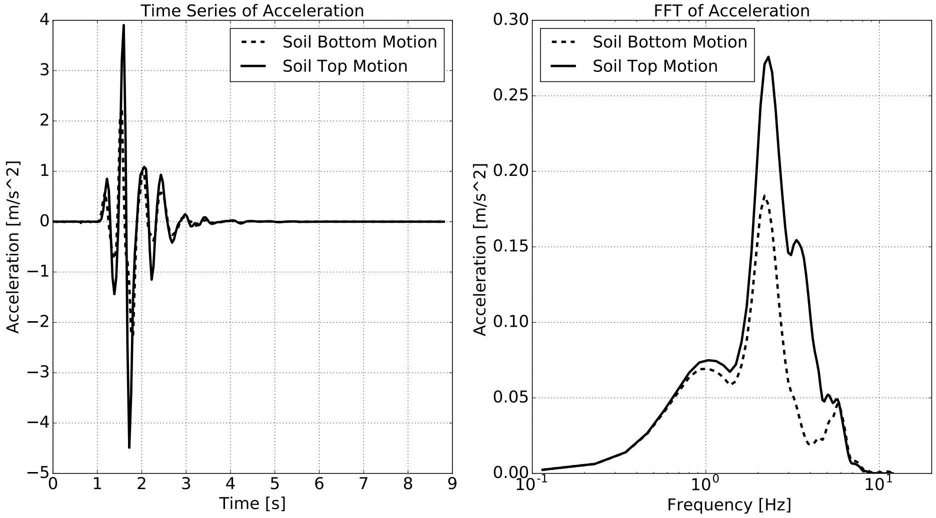


Figure 410.3: Simulation results: acceleration time series.

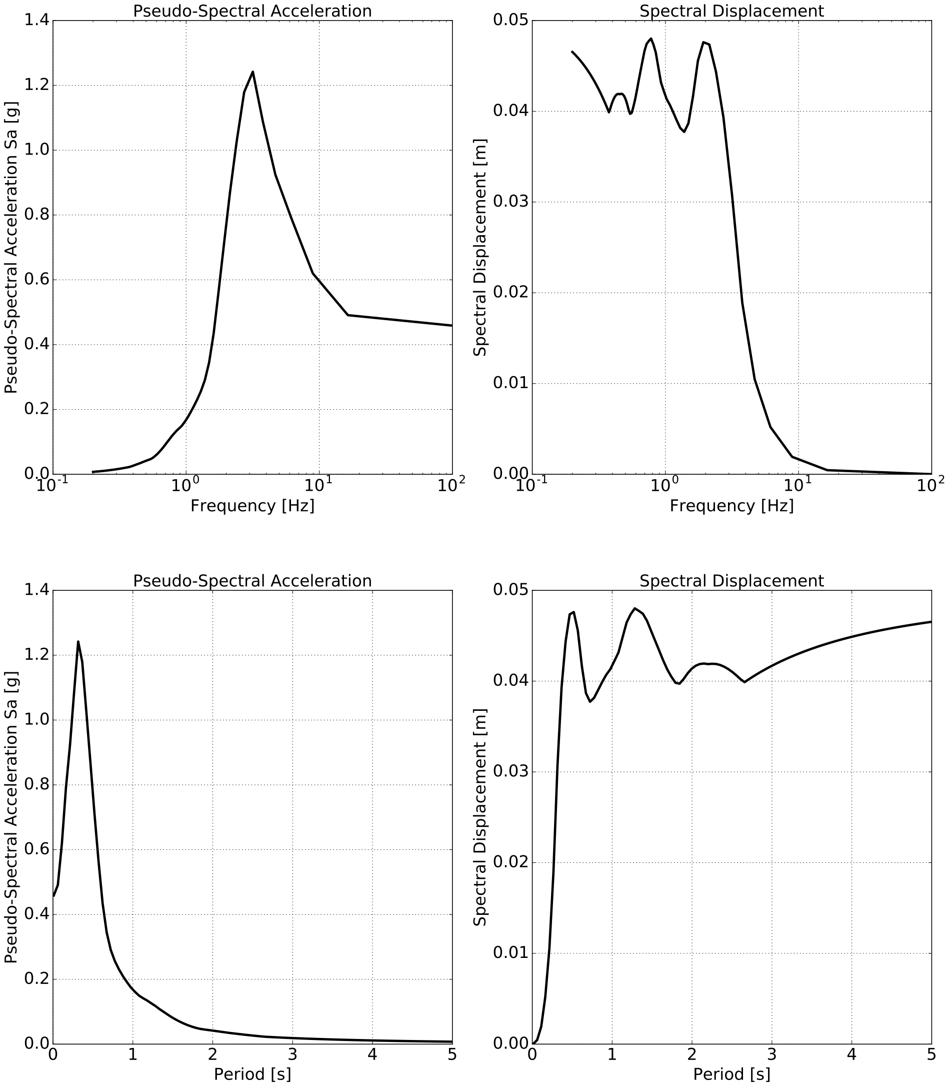


Figure 410.4: Simulation results: response spectrum at soil top.

410.1.2 Free Field 3C

Elastic Material. The compressed package of input files for this example is HERE.

The Modeling parameters are listed below:

- Elastic Material Properties
  - Mass density, ρ, 2000 kg/m^3
  - Shear wave velocity, V_s, 500 m/s
  - Young's modulus, E, 1.1 GPa
  - Poisson's ratio, ν, 0.1

SIMULATION TIME: With 8 cores on AWS EC2 c4.2xlarge instance, the running time for this example is 5 minutes.

von-Mises Armstrong-Frederick Material. The compressed package of input files is HERE.

The Modeling parameters are listed below:

- von-Mises nonlinear hardening material model
  - Mass density, ρ, 2000 kg/m^3
  - Shear wave velocity, V_s, 500 m/s
  - Young's modulus, E, 1.1 GPa
  - Poisson's ratio, ν, 0.1
  - von Mises radius, k, 60 kPa
  - Nonlinear kinematic hardening, H_a, 30 MPa
  - Nonlinear kinematic hardening, C_r, 60
  - Shear strength ($\approx \sqrt{2/3}\ H_a/C_r$), S_u, 408 kPa
  - Isotropic hardening rate, K_{iso}, 0 Pa

SIMULATION TIME: With 8 cores on AWS EC2 c4.2xlarge instance, the running time for this example is 17 minutes.

von-Mises G/Gmax Material. The compressed package of input files is HERE.

The Modeling parameters are listed below:

- von-Mises G/Gmax material model

  - Mass density, ρ, 2000 kg/m^3
  - Shear wave velocity, V_s, 500 m/s
  - Young's modulus, E, 1.1 GPa
  - Poisson's ratio, ν, 0.1
  - Total number of shear modulus 9
  - G over Gmax, 1,0.995,0.966,0.873,0.787,0.467,0.320,0.109,0.063
  - Shear strain gamma, 0,1E-6,1E-5,5E-5,1E-4, 0.0005, 0.001, 0.005, 0.01

SIMULATION TIME: With 8 cores on AWS EC2 c4.2xlarge instance, the running time for this example is 565 minutes.

Drucker-Prager G/Gmax Material. The compressed package of input files is HERE.

The Modeling parameters are listed below:

- Drucker-Prager G/Gmax material model
  - Mass density, ρ, 2000 kg/m^3
  - Shear wave velocity, V_s, 500 m/s
  - Young's modulus, E, 1.1 GPa
  - Poisson's ratio, ν, 0.1
  - Initial confining stress, p_0, 100 kPa
  - Reference pressure, p_{refer}, 100 kPa
  - Pressure exponential, n, 0.5
  - Cohesion, n, 1 kPa
  - Total number of Shear Modulus 9
  - G over Gmax, 1,0.995,0.966,0.873,0.787,0.467,0.320,0.109,0.063
  - Shear strain gamma, 0,1E-6,1E-5,5E-5,1E-4, 0.0005, 0.001, 0.005, 0.01

SIMULATION TIME: With 8 cores on AWS EC2 c4.2xlarge instance, the running time for this example is 565 minutes.

Results are shown in Fig. 410.56.

SIMULATION TIME: With 8 cores on AWS EC2 c4.2xlarge instance, the running time for this example is 871 minutes.

The time series of simulation results is shown in Fig. 410.7.

The response spectrum of motion is shown in Fig. 410.8.

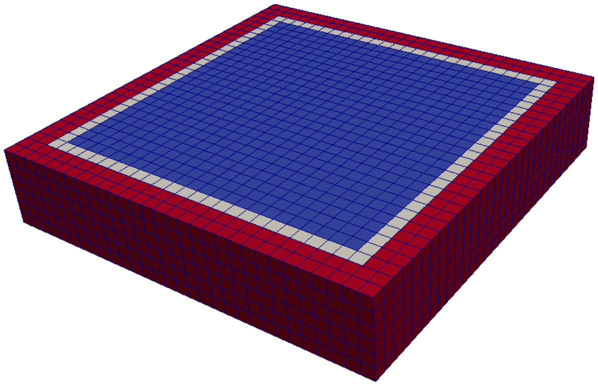


Figure 410.5: Simulation model.

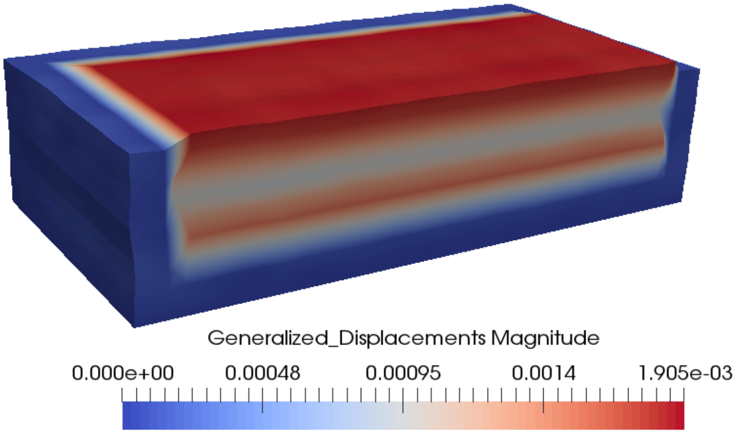


Figure 410.6: Simulation model.

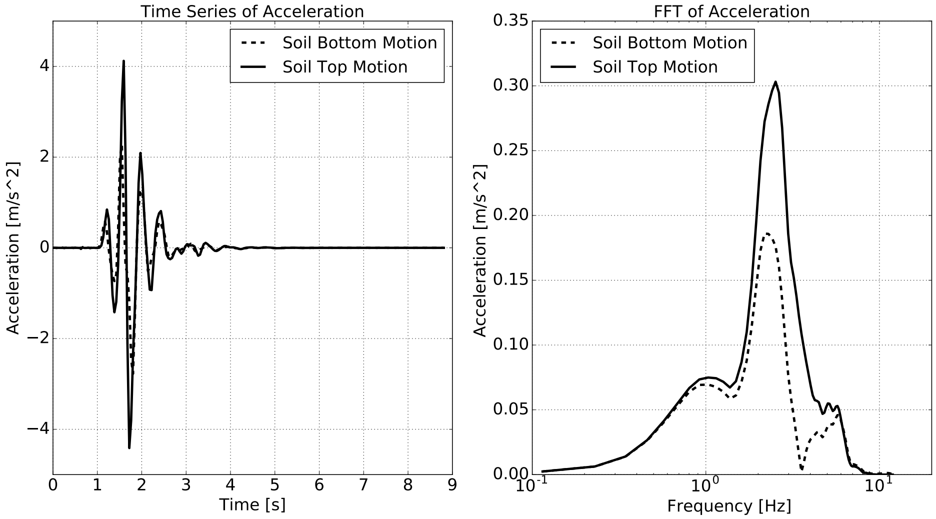


Figure 410.7: Simulation results: acceleration time series.

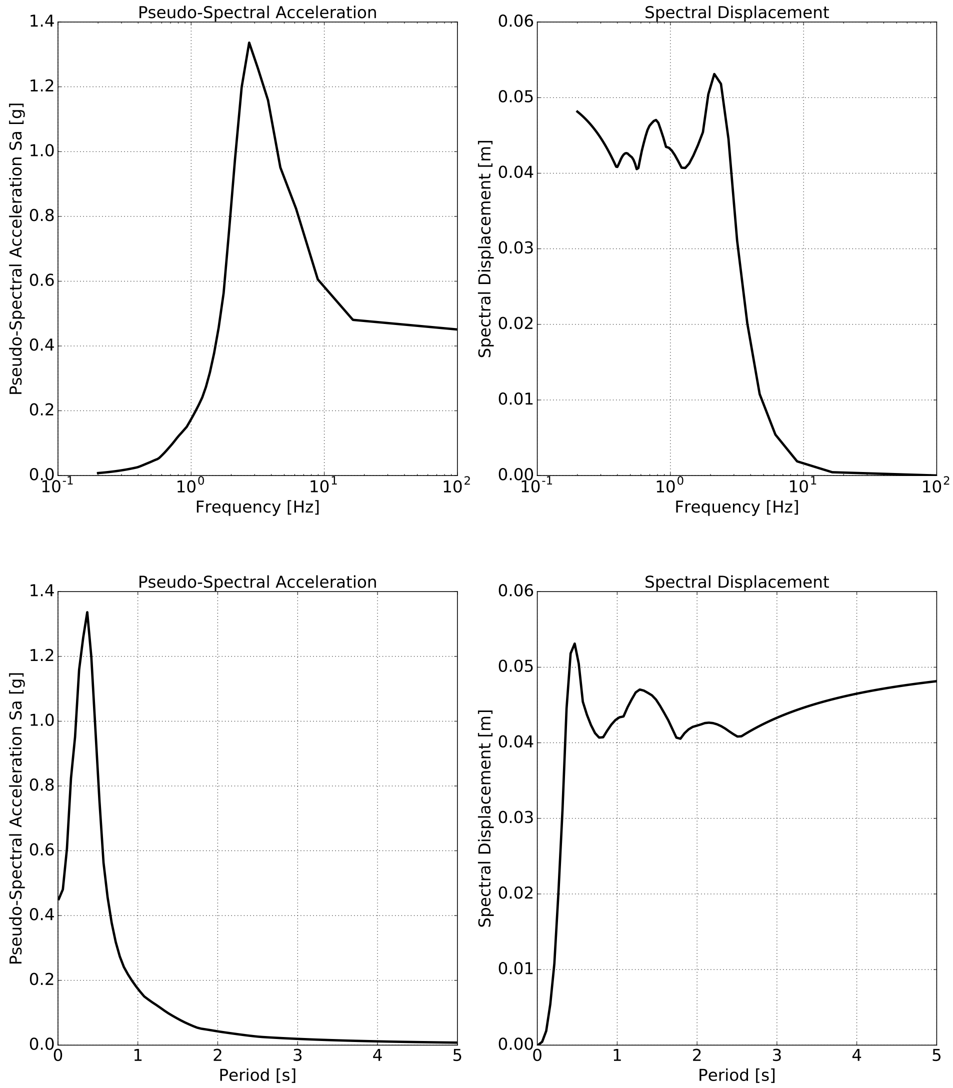


Figure 410.8: Simulation results: response spectrum at soil top.

410.1.3 Soil-Foundation Interaction 3D

Elastic Material. The compressed package of input files is HERE.

The Modeling parameters are listed below:

- Elastic Material Properties
  - Mass density, ρ, 2000 kg/m^3
  - Shear wave velocity, V_s, 500 m/s
  - Young's modulus, E, 1.1 GPa
  - Poisson's ratio, ν, 0.1

SIMULATION TIME: With 8 cores on AWS EC2 c4.2xlarge instance, the running time for this example is 13 minutes.

von-Mises Armstrong-Frederick Material. The compressed package of input files is HERE.

The Modeling parameters are listed below:

- von-Mises nonlinear hardening material model
  - Mass density, ρ, 2000 kg/m^3
  - Shear wave velocity, V_s, 500 m/s
  - Young's modulus, E, 1.1 GPa
  - Poisson's ratio, ν, 0.1
  - von Mises radius, k, 60 kPa
  - Nonlinear kinematic hardening, H_a, 30 MPa
  - Nonlinear kinematic hardening, C_r, 60
  - Shear strength ($\approx \sqrt{2/3}\ H_a/C_r$), S_u, 408 kPa
  - Isotropic hardening rate, K_{iso}, 0 Pa

SIMULATION TIME: With 8 cores on AWS EC2 c4.2xlarge instance, the running time for this example is 36 minutes.

von-Mises G/Gmax Material. The compressed package of input files is HERE.

The Modeling parameters are listed below:

- von-Mises G/Gmax material model

  - Mass density, ρ, 2000 kg/m^3
  - Shear wave velocity, V_s, 500 m/s
  - Young's modulus, E, 1.1 GPa
  - Poisson's ratio, ν, 0.1
  - Total number of shear modulus 9
  - G over Gmax, 1,0.995,0.966,0.873,0.787,0.467,0.320,0.109,0.063
  - Shear strain gamma, 0,1E-6,1E-5,5E-5,1E-4, 0.0005, 0.001, 0.005, 0.01

SIMULATION TIME: With 8 cores on AWS EC2 c4.2xlarge instance, the running time for this example is 726 minutes.

Drucker-Prager G/Gmax Material. The compressed package of input files is HERE.

The Modeling parameters are listed below:

- Drucker-Prager G/Gmax material model
  - Mass density, ρ, 2000 kg/m^3
  - Shear wave velocity, V_s, 500 m/s
  - Young's modulus, E, 1.1 GPa
  - Poisson's ratio, ν, 0.1
  - Initial confining stress, p_0, 100 kPa
  - Reference pressure, p_{refer}, 100 kPa
  - Pressure exponential, n, 0.5
  - Cohesion, n, 1 kPa
  - Total number of Shear Modulus 9
  - G over Gmax, 1,0.995,0.966,0.873,0.787,0.467,0.320,0.109,0.063
  - Shear strain gamma, 0,1E-6,1E-5,5E-5,1E-4, 0.0005, 0.001, 0.005, 0.01

SIMULATION TIME: With 8 cores on AWS EC2 c4.2xlarge instance, the running time for this example is 1252 minutes.

Contact/Interface/Joint Elements. The compressed package of input files is HERE.

The Modeling parameters are listed below:

- Elastic Material Properties
  - Mass density, ρ, 2000 kg/m^3
  - Shear wave velocity, V_s, 500 m/s
  - Young's modulus, E, 1.1 GPa
  - Poisson's ratio, ν, 0.1

SIMULATION TIME: With 8 cores on AWS EC2 c4.2xlarge instance, the running time for this example is 24 minutes.

Both Elastoplastic Material and Contact/Interface/Joint Elements. The compressed package of input files is HERE.

SIMULATION TIME: With 8 cores on AWS EC2 c4.2xlarge instance, the running time for this example is 41 minutes.

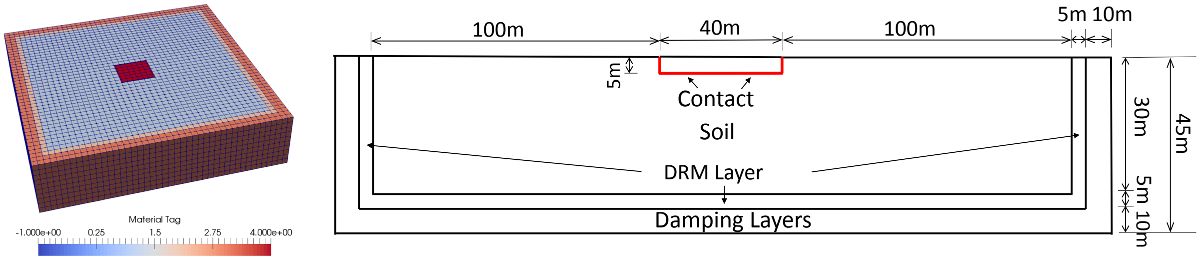


Figure 410.9: Simulation model.

Results of the simulation are shown in Fig. 410.12.

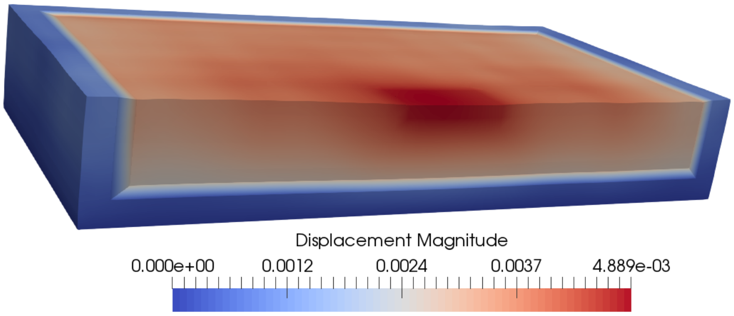


Figure 410.10: Soil foundation interaction results.

410.1.4 Soil-Structure Interaction 3D

Elastic Material. The compressed package of input files is HERE.

The Modeling parameters are listed below:

- Elastic Material Properties
  - Mass density, ρ, 2000 kg/m^3
  - Shear wave velocity, V_s, 500 m/s
  - Young's modulus, E, 1.1 GPa
  - Poisson's ratio, ν, 0.1

SIMULATION TIME: With 8 cores on AWS EC2 c4.2xlarge instance, the running time for this example is 10 minutes.

von-Mises Armstrong-Frederick Material. The compressed package of input files is HERE.

The Modeling parameters are listed below:

- von-Mises nonlinear hardening material model
  - Mass density, ρ, 2000 kg/m^3
  - Shear wave velocity, V_s, 500 m/s
  - Young's modulus, E, 1.1 GPa
  - Poisson's ratio, ν, 0.1
  - von Mises radius, k, 60 kPa
  - Nonlinear kinematic hardening, H_a, 30 MPa
  - Nonlinear kinematic hardening, C_r, 60
  - Shear strength ($\approx \sqrt{2/3}\ H_a/C_r$), S_u, 408 kPa
  - Isotropic hardening rate, K_{iso}, 0 Pa

SIMULATION TIME: With 8 cores on AWS EC2 c4.2xlarge instance, the running time for this example is 46 minutes.

von-Mises G/Gmax Material. The compressed package of input files is HERE.

The Modeling parameters are listed below:

- von-Mises G/Gmax material model

  - Mass density, ρ, 2000 kg/m^3
  - Shear wave velocity, V_s, 500 m/s
  - Young's modulus, E, 1.1 GPa
  - Poisson's ratio, ν, 0.1
  - Total number of shear modulus 9
  - G over Gmax, 1,0.995,0.966,0.873,0.787,0.467,0.320,0.109,0.063
  - Shear strain gamma, 0,1E-6,1E-5,5E-5,1E-4, 0.0005, 0.001, 0.005, 0.01

SIMULATION TIME: With 8 cores on AWS EC2 c4.2xlarge instance, the running time for this example is 755 minutes.

Drucker-Prager G/Gmax Material. The compressed package of input files is HERE.

SIMULATION TIME: With 8 cores on AWS EC2 c4.2xlarge instance, the running time for this example is 1178 minutes.

The Modeling parameters are listed below:

- Drucker-Prager G/Gmax material model
  - Mass density, ρ, 2000 kg/m^3
  - Shear wave velocity, V_s, 500 m/s
  - Young's modulus, E, 1.1 GPa
  - Poisson's ratio, ν, 0.1
  - Initial confining stress, p_0, 100 kPa
  - Reference pressure, p_{refer}, 100 kPa
  - Pressure exponential, n, 0.5
  - Cohesion, n, 1 kPa
  - Total number of Shear Modulus 9
  - G over Gmax, 1,0.995,0.966,0.873,0.787,0.467,0.320,0.109,0.063
  - Shear strain gamma, 0,1E-6,1E-5,5E-5,1E-4, 0.0005, 0.001, 0.005, 0.01

SIMULATION TIME: With 8 cores, the running time for this example is

Contact/Interface/Joint Elements. The compressed package of input files is HERE.

SIMULATION TIME: With 8 cores on AWS EC2 c4.2xlarge instance, the running time for this example is 15 minutes.

Both Elastoplastic Material and Contact/Interface/Joint Elements. The compressed package of input files is HERE.

The thickness of the shell structure is 2 meters.

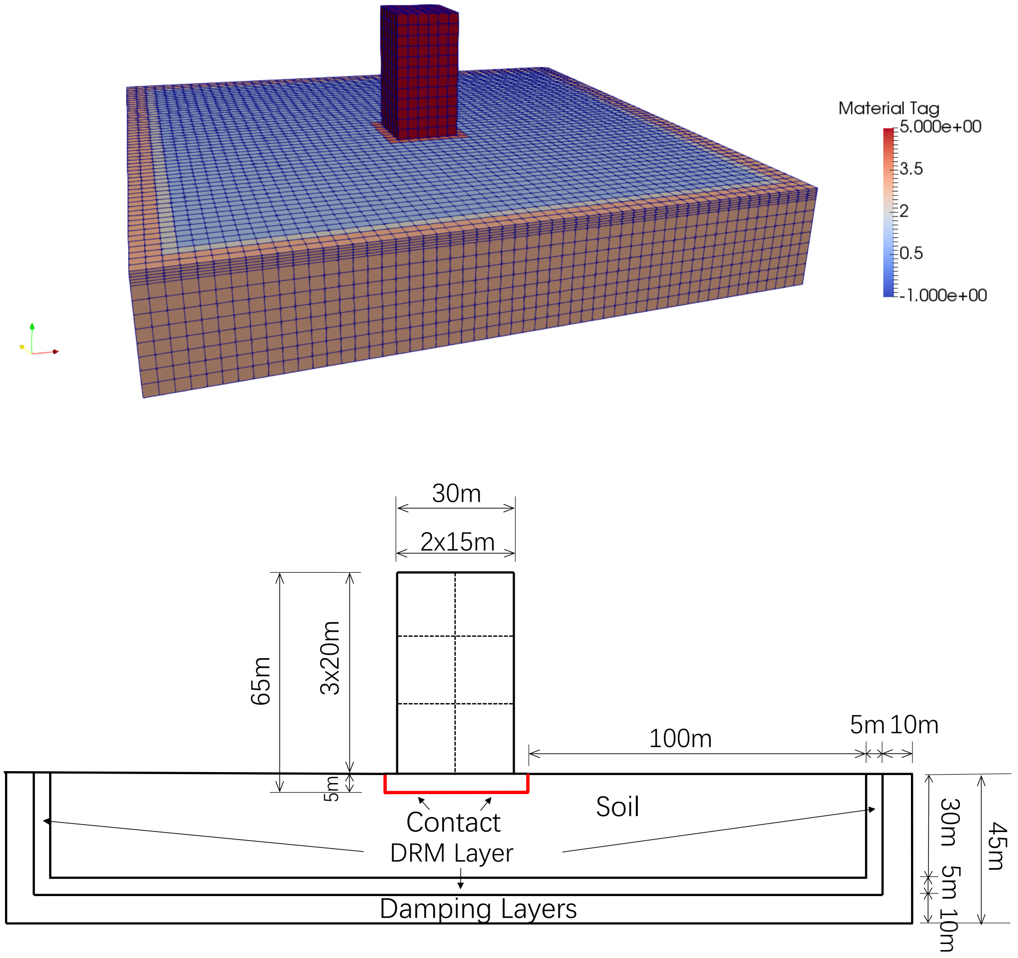


Figure 410.11: Simulation Model.

Results of the simulation are shown in Fig. 410.12.

SIMULATION TIME: With 8 cores on AWS EC2 c4.2xlarge instance, the running time for this example is 47 minutes.

Simulation with 1C motion. The time series of simulation results is shown in Fig. 410.13.

The response spectrum of motion is shown in Fig. 410.14.

Simulation with 3 × 1C motion. The time series of simulation results is shown in Fig. 410.15.

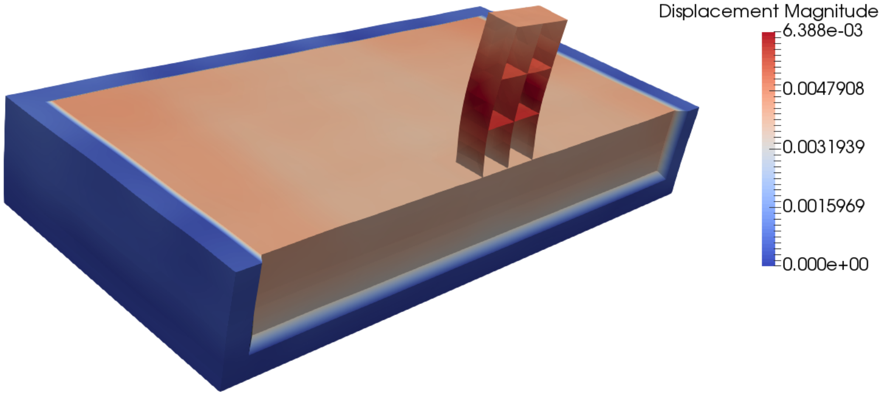


Figure 410.12: Simulation Model.

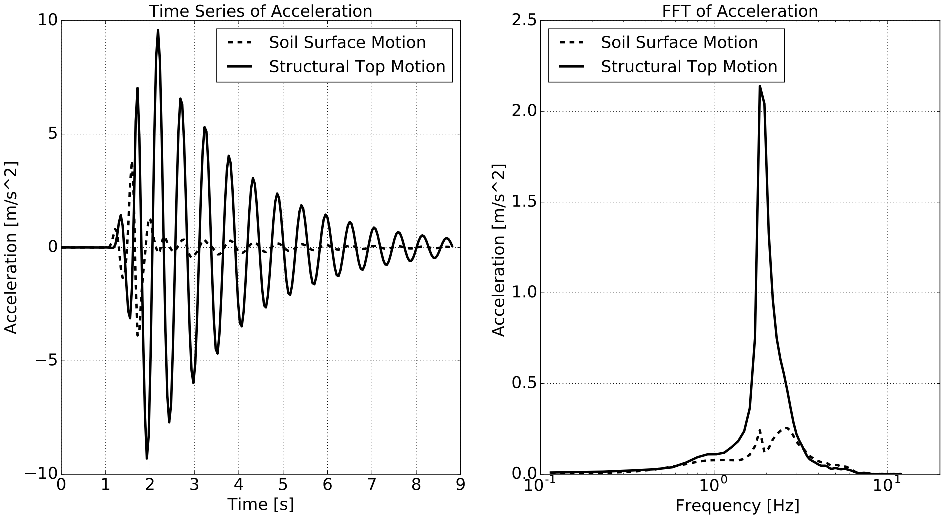


Figure 410.13: Simulation Results: Acceleration Time Series with 1C motion.

The response spectrum of motion is shown in Fig. 410.16.

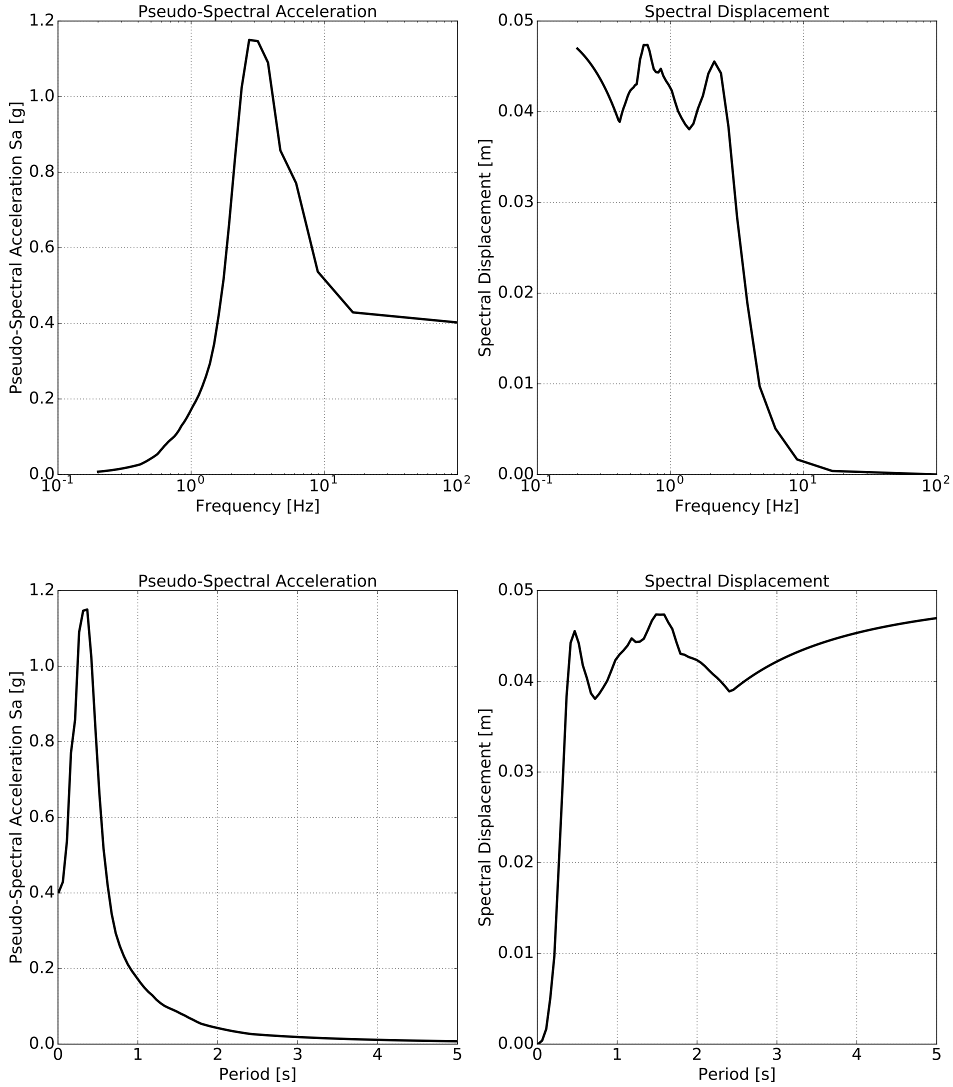


Figure 410.14: Simulation Results: Response Spectrum of Structure Top with 1C motion.

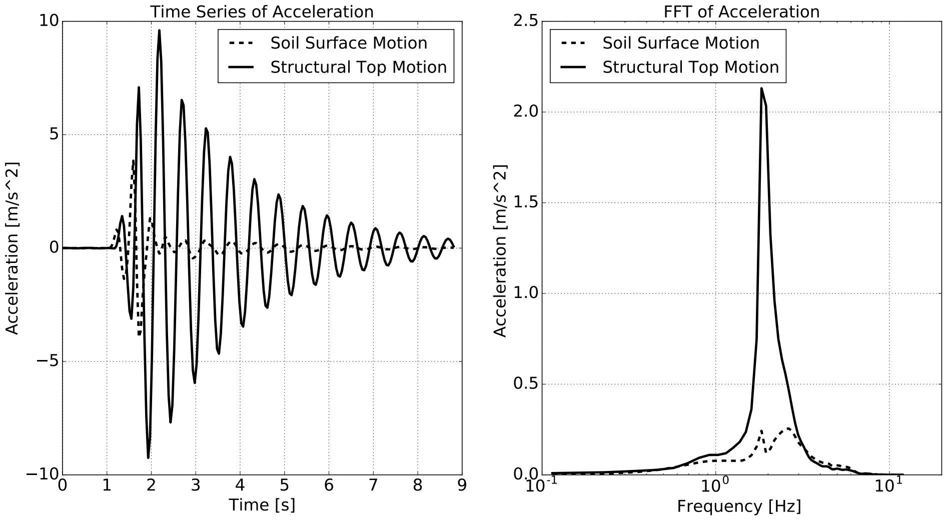


Figure 410.15: Simulation Results: Acceleration Time Series with 3C motion.

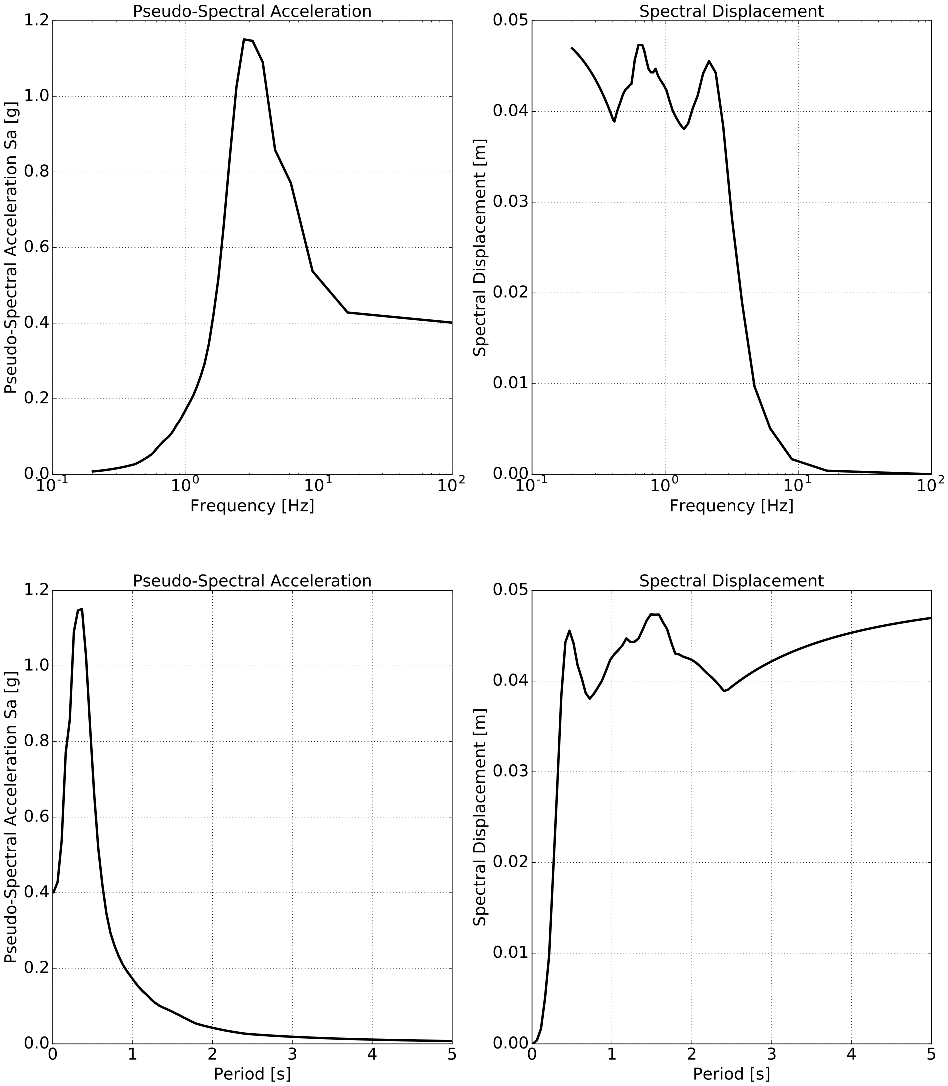


Figure 410.16: Simulation Results: Response Spectrum of Structure Top with 3C motion.

410.1.5 Analysis of a Structure without Soil

410.1.5.1 Eigen Analysis

Eigen analysis of a fixed base structural model should provide a good check of the structural model, natural (eigen) frequencies, and natural (eigen) modes.

The compressed package of input files is HERE.

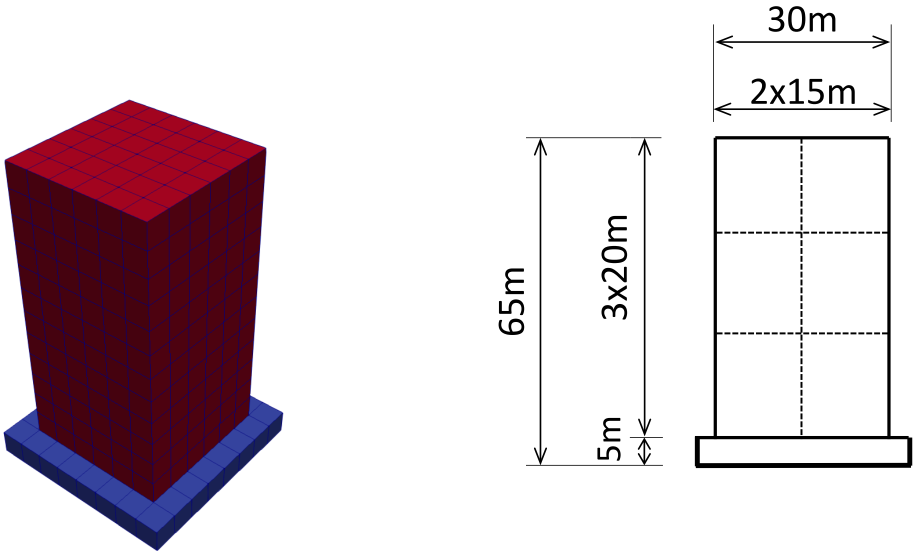


Figure 410.17: Structure on a fixed based simulation model.

For this particular example, eigen modes and frequencies are given in Figures 410.18 and 410.19

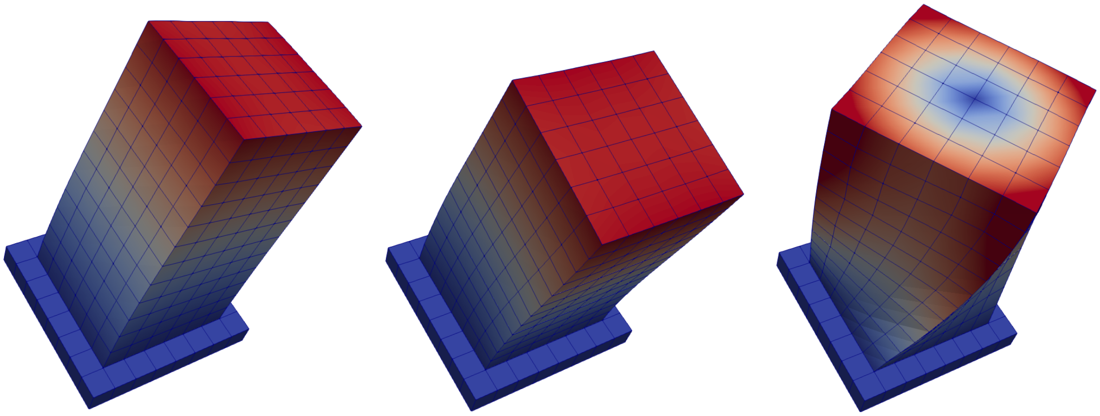


Figure 410.18: Eigen frequencies: f_1 = 3.47Hz f_2 = 3.47Hz f_3 = 6.88Hz (eigen mode 1 to 3 from left to right).

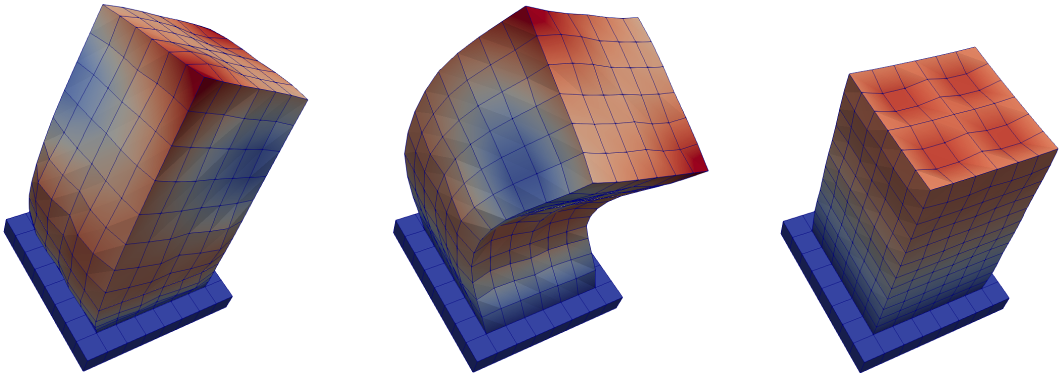


Figure 410.19: Eigen frequencies: $f_4 = 11.50\text{Hz}$ $f_5 = 11.50\text{Hz}$ $f_6 = 12.13\text{Hz}$ (eigen modes 4 to 6 from left to right).

Input files for eigen analysis of the fixed base structure are available at this LINK, and can be directly simulated using Real-ESSI Simulator, `http://real-essi.us/`, that is available on Amazon Web Services, `https://aws.amazon.com/`.

410.1.5.2 Imposed Motion

The Real-ESSI input files for this example are available HERE. The compressed package of input files is HERE.

In addition to eigen analysis, fixed base structural model is used to test response of a fixed base structure. This is important as it provides an opportunity to compare results between different finite element programs, some of which can only model dynamics of fixed base structures.l

The simulation model is shown below.

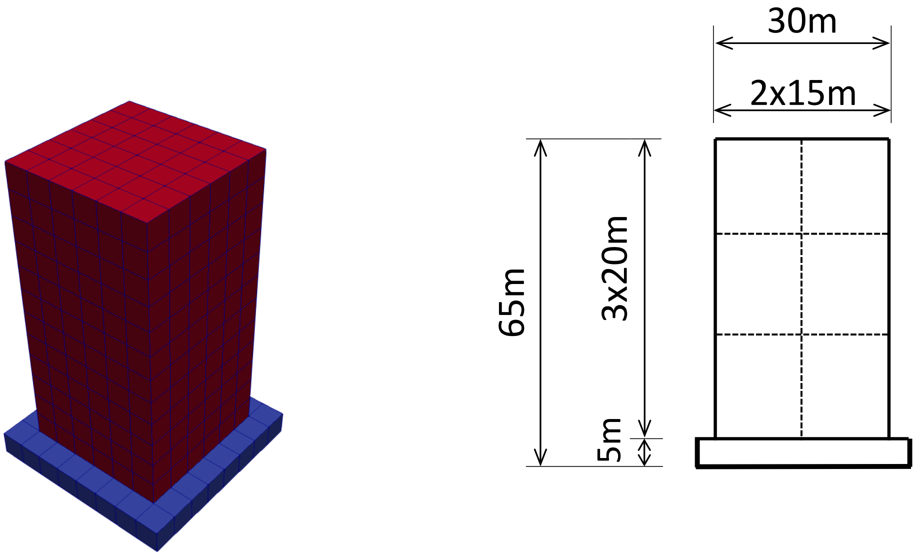


Figure 410.20: Simulation Model.

The simulation results:

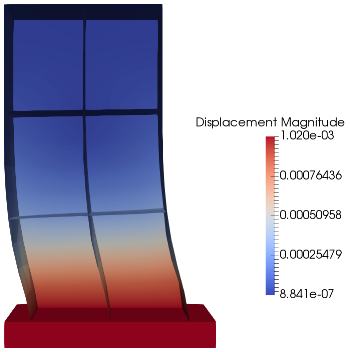


Figure 410.21: Simulation Results.

The time series of simulation results is shown in Fig. 410.22.

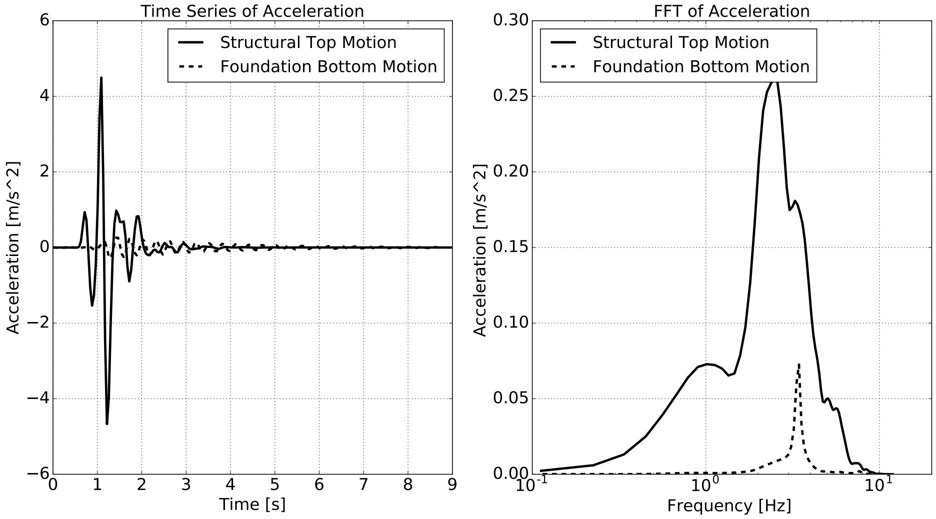


Figure 410.22: Simulation Results: Acceleration Time Series with 1C imposed motion.

The response spectrum of motion is shown in Fig. 410.23.

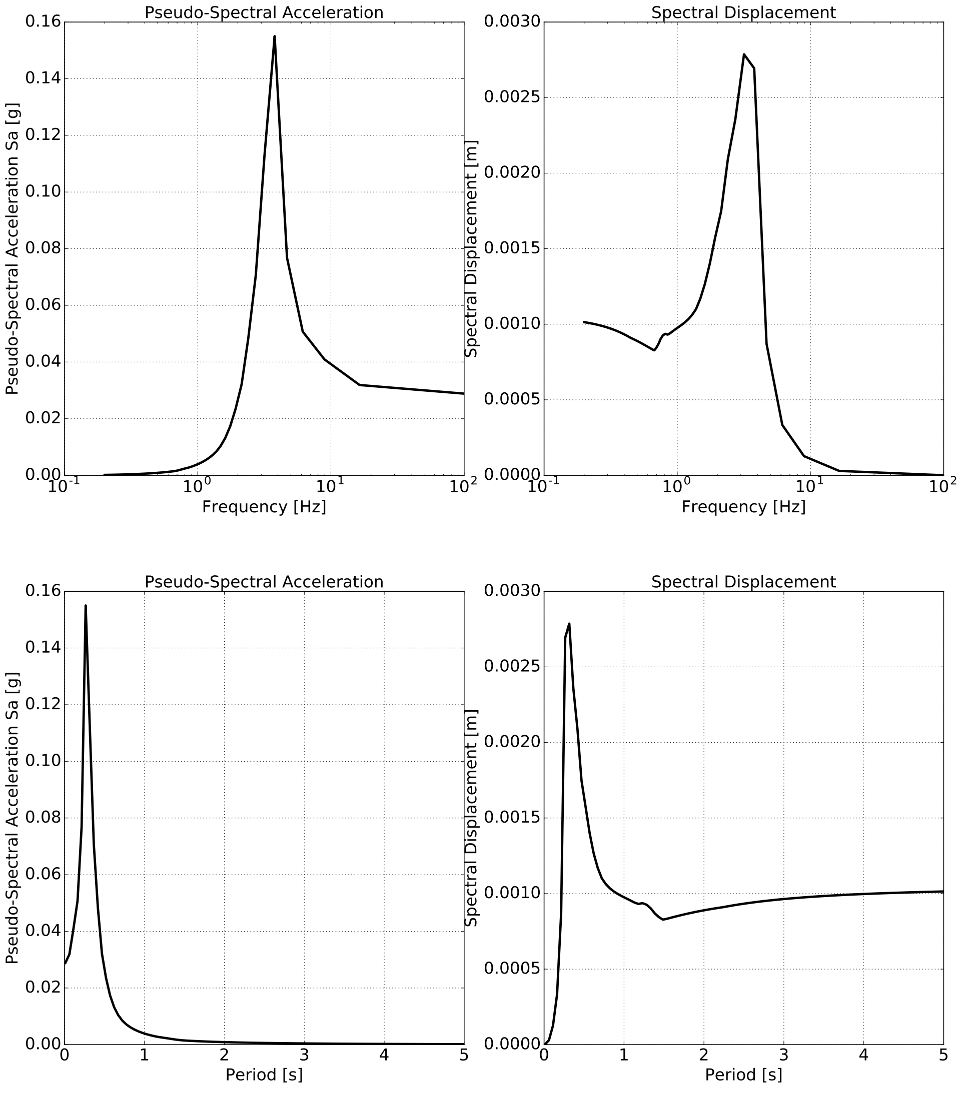


Figure 410.23: Simulation Results: Response Spectrum of Structure Top with 1C imposed motion.

410.2 Day 1: Overview

410.2.1 Nuclear Power Plant with 3C motions from SW4

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

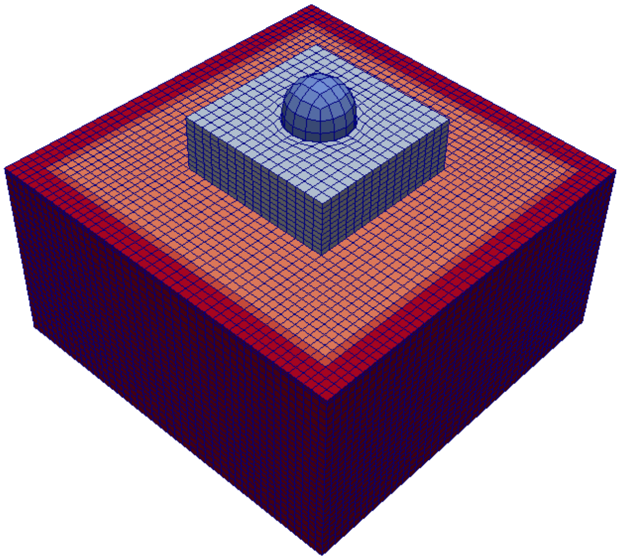


Figure 410.24: Simulation Model.

The Modeling parameters are listed below:

- Soil
  - Unit weight, γ, 21.4 kPa
  - Shear velocity, Vs, 500 m/s
  - Young's modulus, E, 1.3 GPa
  - Poisson's ratio, ν, 0.25
  - Shear strength, S_u, 650 kPa
  - von Mises radius, k, 60 kPa
  - kinematic hardening, H_a, 30 MPa
  - kinematic hardening, C_r, 25
- Structure
  - Unit weight, γ, 24 kPa

  - Young's modulus, E, 20 GPa
  - Poisson's ratio, ν, 0.21

The input motion at the bottom is a 3C wave from SW4.

SIMULATION TIME: With 32 cores on AWS EC2 c4.8xlarge instance, the running time for this example is 17 hours.

410.2.2 Nuclear Power Plant with 1C motions from Deconvolution

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

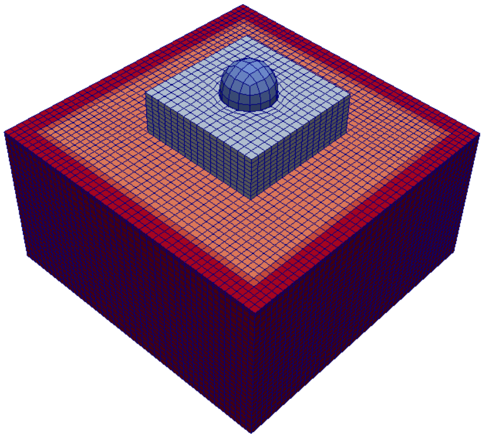


Figure 410.25: Simulation Model.

The input motion at the bottom is the deconvolution of the Northridge earthquake records.

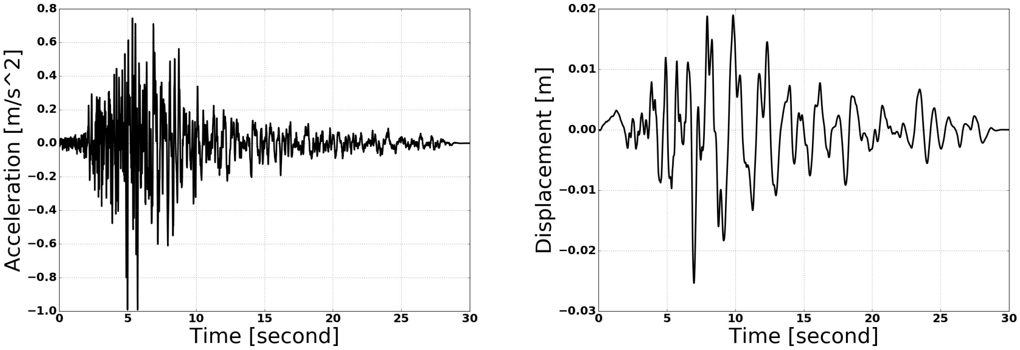


Figure 410.26: Motion Deconvolution.

The Modeling parameters are listed below:

- Soil
  - Unit weight, γ, 21.4 kPa
  - Shear velocity, Vs, 500 m/s
  - Young's modulus, E, 1.3 GPa
  - Poisson's ratio, ν, 0.25

  - Shear strength, S_u, 650 kPa
  - von Mises radius, k, 60 kPa
  - kinematic hardening, H_a, 30 MPa
  - kinematic hardening, C_r, 25

- Structure
  - Unit weight, γ, 24 kPa
  - Young's modulus, E, 20 GPa
  - Poisson's ratio, ν, 0.21

410.2.3 Nuclear Power Plant with 3×1C motions from Deconvolution

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

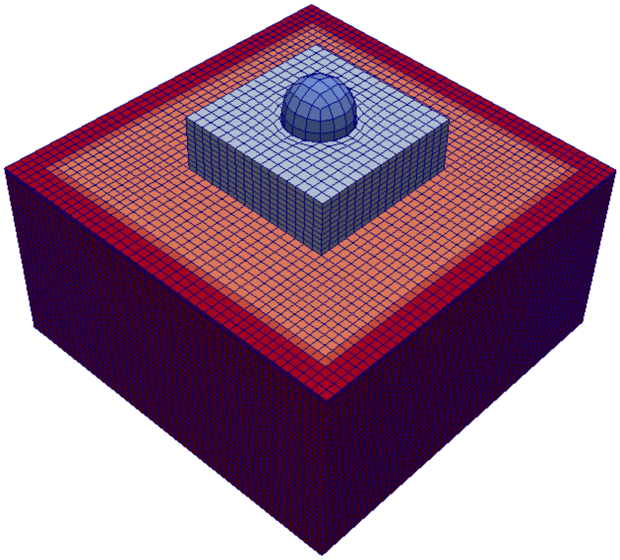


Figure 410.27: Simulation Model.

The input motion at the bottom is the deconvolution of the Northridge earthquake records.

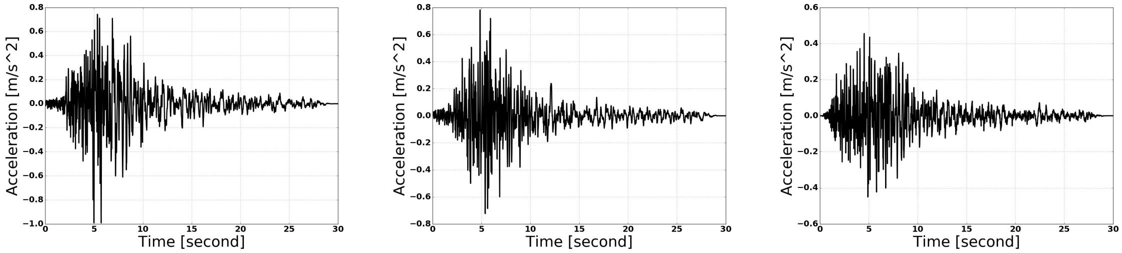


Figure 410.28: Acceleration Deconvolution, from left to right in x, y, z directions respectively. .

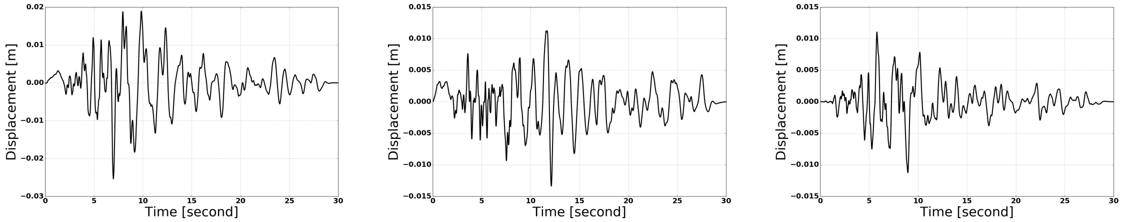


Figure 410.29: Displacement Deconvolution, from left to right in x, y, z directions respectively. .

The Modeling parameters are listed below:

- Soil
  - Unit weight, γ, 21.4 kPa
  - Shear velocity, Vs, 500 m/s
  - Young's modulus, E, 1.3 GPa
  - Poisson's ratio, ν, 0.25
  - Shear strength, S_u, 650 kPa
  - von Mises radius, k, 60 kPa
  - kinematic hardening, H_a, 30 MPa
  - kinematic hardening, C_r, 25
- Structure
  - Unit weight, γ, 24 kPa
  - Young's modulus, E, 20 GPa
  - Poisson's ratio, ν, 0.21

410.2.4 Single Element Models: Illustration of the Elastic-Plastic Behavior

The compressed package of Real-ESSI input files for this example with von-Mises material model are available HERE.

The compressed package of Real-ESSI input files for this example with Drucker-Prager material model are available HERE.

The Modeling parameters are listed below:

- von-Mises linear hardening material model
  - Mass Density, ρ, 0.0 kg/m^3
  - Young's modulus, E, 20 MPa
  - Poisson's ratio, ν, 0.0
  - von Mises radius, k, 100 kPa
  - kinematic hardening rate, K_{kine}, 2 MPa
  - isotropic hardening rate, K_{iso}, 0 Pa
- Drucker-Prager nonlinear hardening material model
  - Mass Density, ρ, 0.0 kg/m^3
  - Young's modulus, E, 20 MPa
  - Poisson's ratio, ν, 0.0
  - Drucker-Prager, k, 0.179527
  - nonlinear kinematic hardening, H_a, 20 MPa
  - nonlinear kinematic hardening, C_r, 100
  - isotropic hardening rate, K_{iso}, 0 Pa
  - initial confining stress, p_0, 1 Pa

Inelastic/nonlinear material behavior is shown in Fig. 410.31.

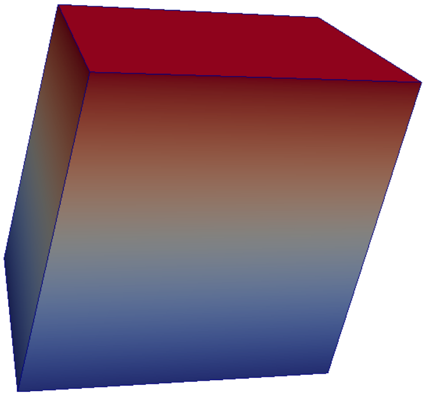


Figure 410.30: Simulation Model of Single Element.

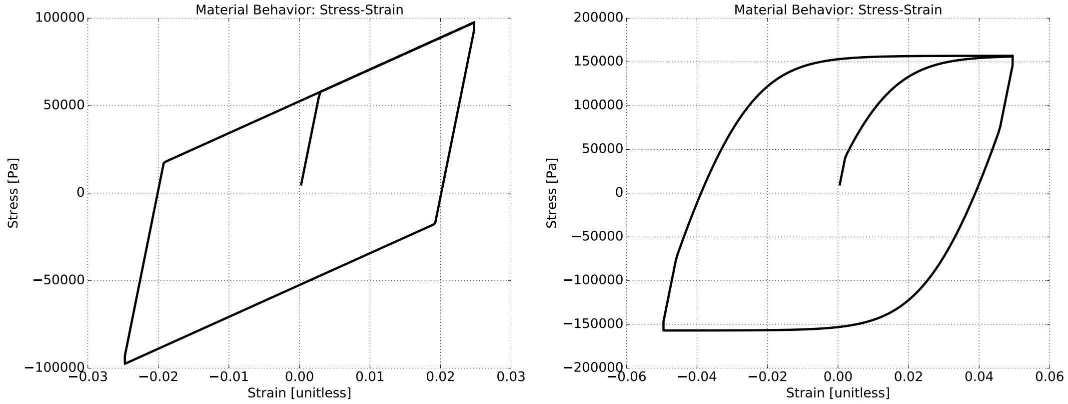


Figure 410.31: Inelastic/Nonlinear material behavior.

410.2.5 Pushover for Nonlinear Frame

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

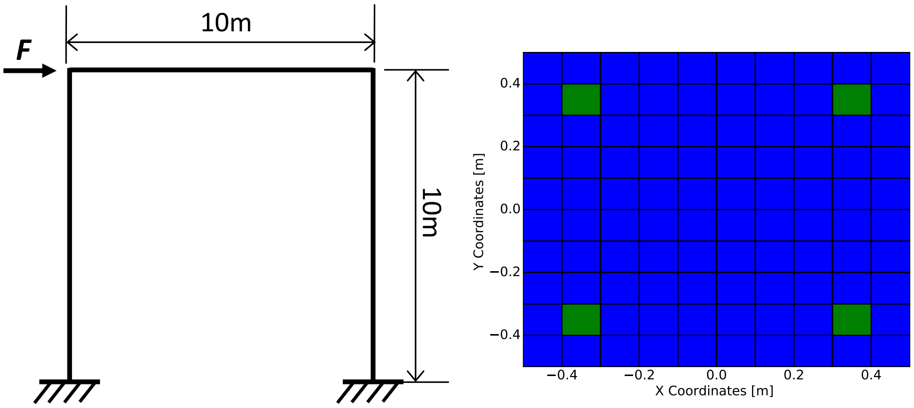


Figure 410.32: Model for pushover simulation and the cross section of fiber beam (concrete and reinforcement).

Result are shown in Fig. 410.33.

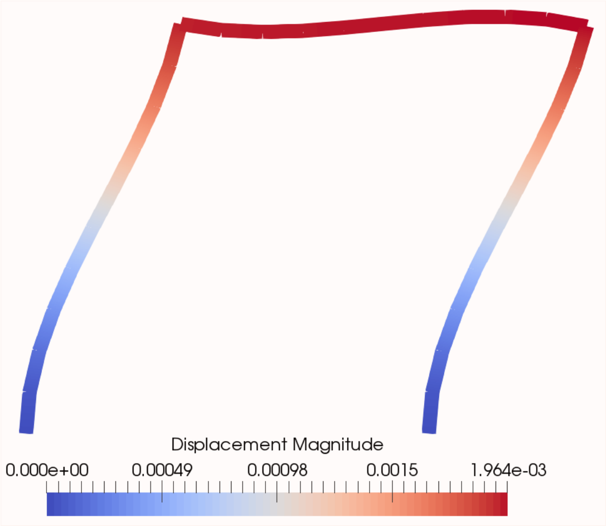


Figure 410.33: Results for fiber beam pushover.

The Modeling parameters are listed below:

- Uniaxial concrete
  - Compressive strength, 24 MPa
  - Strain at compressive strength, 0.001752

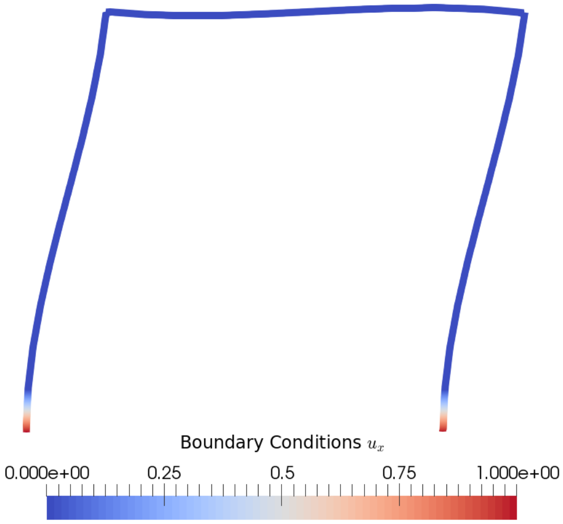


Figure 410.34: Boundary condition u_x for fiber beam pushover.

  - Crushing strength, 0.0 Pa
  - Strain at compressive strength, 0.003168
  - lambda, 0.5
  - Tensile strength, 0 Pa
  - Tension softening stiffness, 0 Pa

- Uniaxial steel
  - Yield strength, 413.8 MPa
  - Young's modulus, 200 GPa
  - Strain hardening ratio, 0.01
  - R0, 18.0
  - cR1, 0.925
  - cR2, 0.15
  - a1, 0.0
  - a2, 55.0
  - a3, 0.0
  - a4, 55.0

410.2.6 Pre-Processing examples with Gmsh

410.2.6.1 Cantilever Example

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

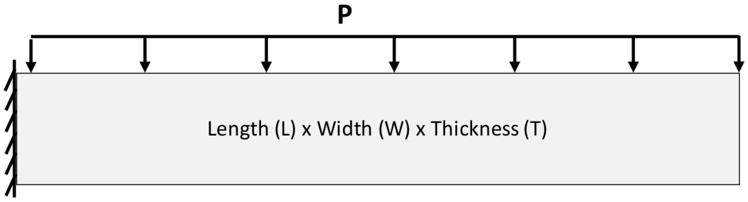


Figure 410.35: Simulation Model Cantilever.

Results are shown in Fig. 410.36.

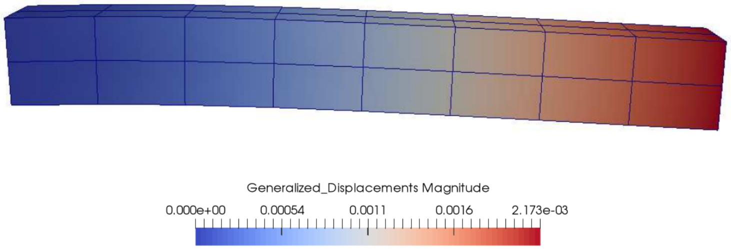


Figure 410.36: Simulation model. cantilever, results.

410.2.6.2 Brick-shell-beam Example

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

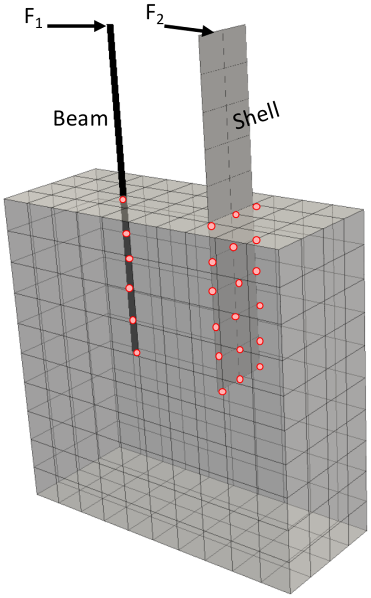


Figure 410.37: Simulation Model Brick-Shell-Beam.

Results are shown in Fig. 410.38.

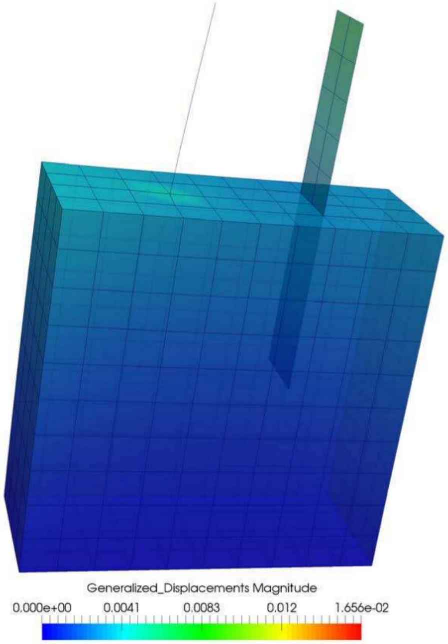


Figure 410.38: Brick-Shell-Beam, Results.

410.2.6.3 DRM 2D Example

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

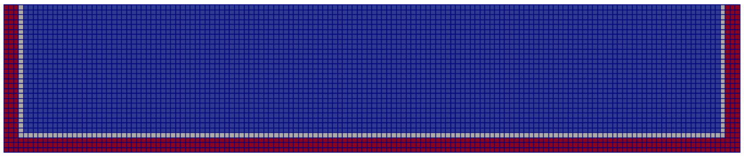


Figure 410.39: Simulation Model DRM 2D.

Results of free field DRM 2D Model under 1C motion are shown in Fig. 410.40.

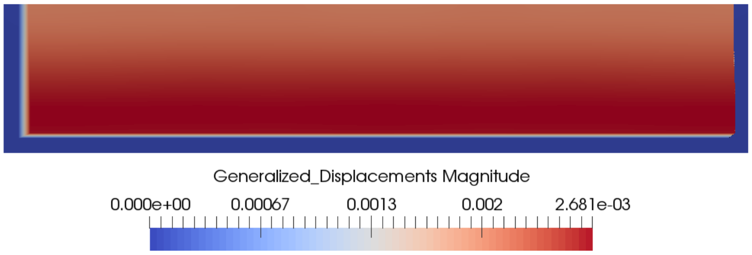


Figure 410.40: Simulation Model DRM 2D.

410.2.6.4 DRM 3D Example

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

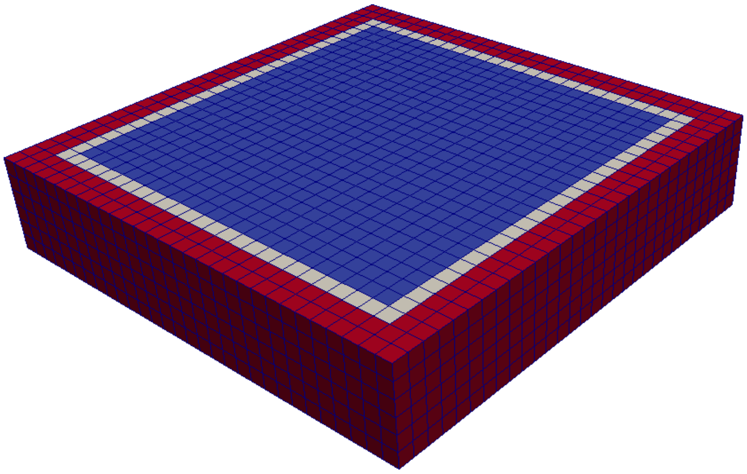


Figure 410.41: Simulation Model DRM 3D.

Results of free field DRM 3D Model under 1C motion are shown in Fig. 410.42.

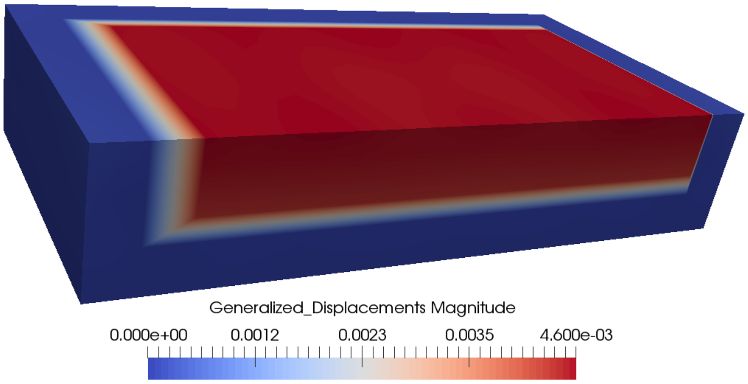


Figure 410.42: Simulation Model DRM 2D.

410.2.7 Post-processing examples with ParaView

410.2.7.1 Slice Visualization

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

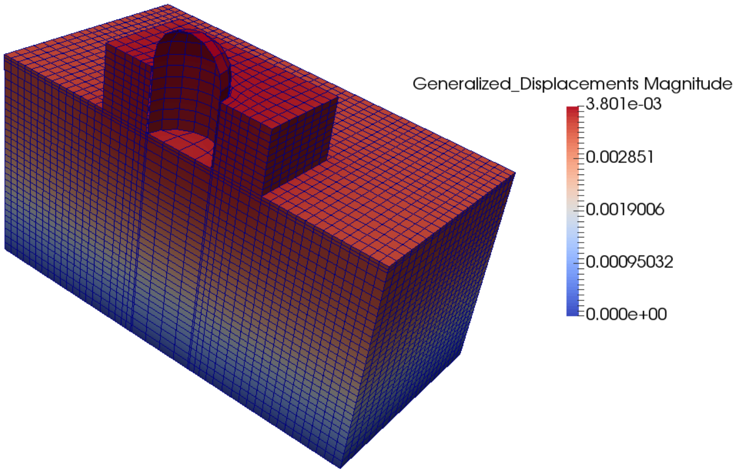


Figure 410.43: Slice Visualization with ParaView.

410.2.7.2 Stress Visualization

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

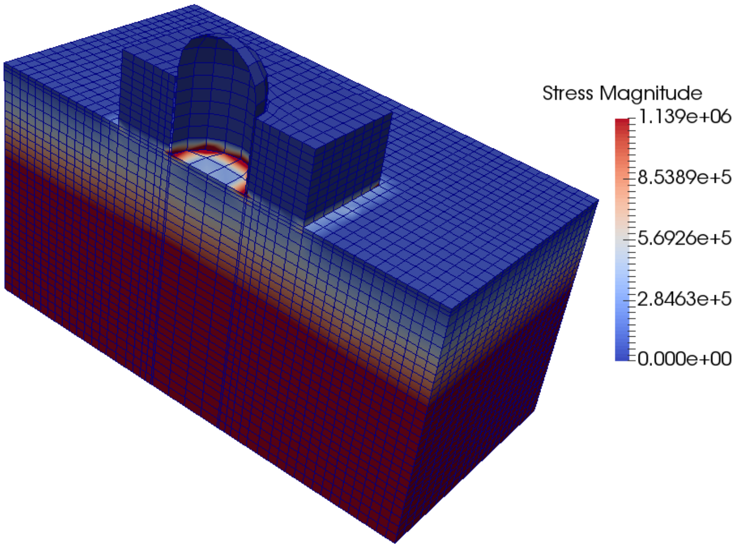


Figure 410.44: Stress Visualization with ParaView.

410.2.7.3 Pore Pressure Visualization with upU Element

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

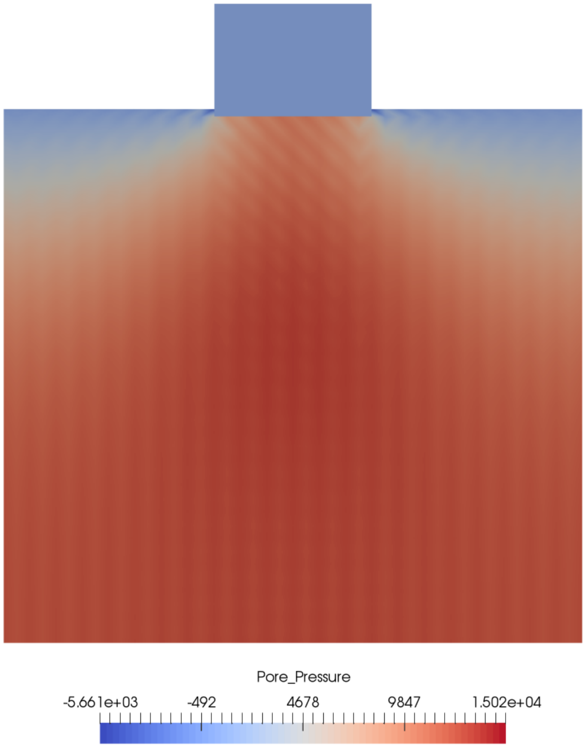


Figure 410.45: Pore Pressure Visualization with Paraview.

410.2.7.4 Eigen Visualization

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

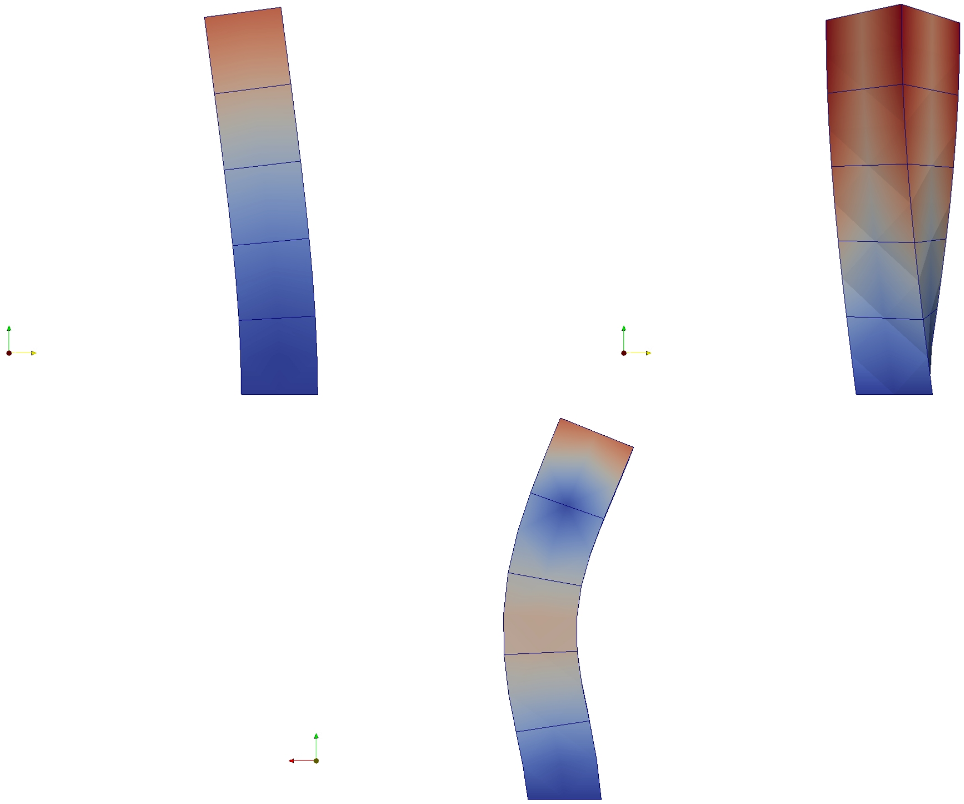


Figure 410.46: Eigen Mode Visualization with Paraview.

410.2.8 Check Model and Visualization of Boundary Conditions

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

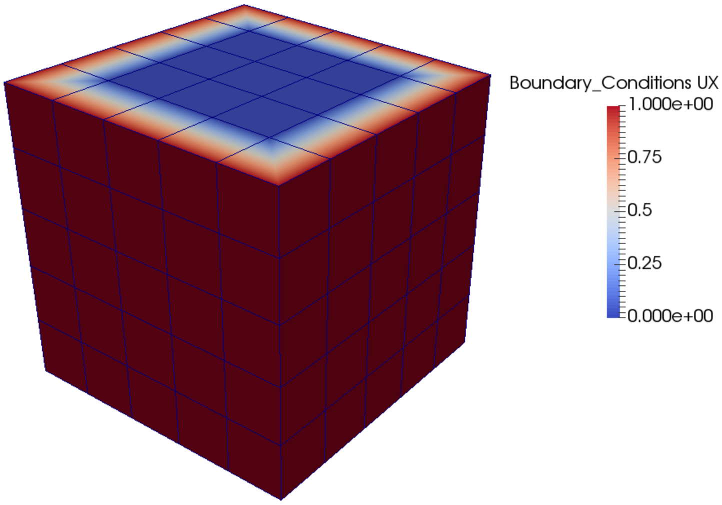


Figure 410.47: Partition Information Visualization with Paraview.

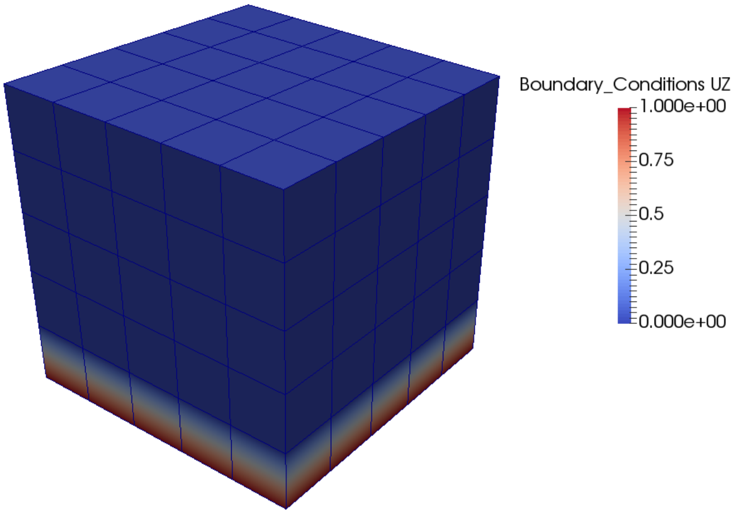


Figure 410.48: Partition Information Visualization with Paraview.

410.2.9 Restart Simulation

410.2.9.1 Restart in the next stage

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

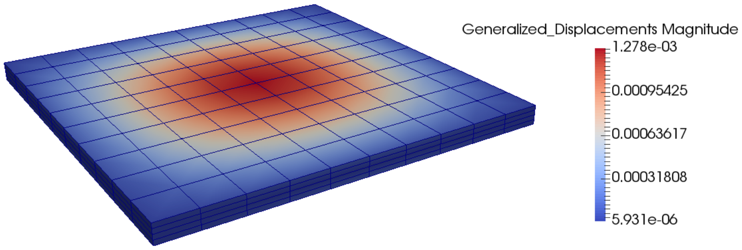


Figure 410.49: Restart Simulation.

This group of examples illustrates the restart functionality between loading stages. There are three test cases in this example. The two loading stages in the first test case is split into two test cases to show the restart feature.

- The first test case run through two loading stages.
- The second test case only run the first loading stage and saves model state at the end.
- The third test case restart the simulation from the saved model state of the second test case. Then, with the restart model state, the test case run the second loading stage only.

Results of the third test case are exactly the same to the first test case.

410.2.9.2 Restart inside the stage

For the case of lack of convergence, restart with the previous loading stage.

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

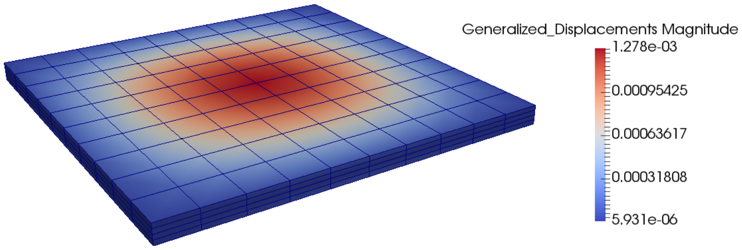


Figure 410.50: Restart Simulation.

This group of examples illustrate the restart functionality inside one loading stage when the simulation cannot converge in the nonlinear analysis. The nonlinear material model, von-Mises Armstrong-Frederick, is used in all test cases.

There are three test cases in this example.

- The first test case run through the whole simulation with a relatively big tolerance of the unbalanced force.
- The second test case failed in the middle of the simulation with a relatively small tolerance of the unbalanced force. When the second test failed, the model reverted to the last commit model state and saved model state.
- The third test case load the saved model state, increased the tolerance of the unbalanced force, and added the remaining load to the model to continue the simulation.

Results of the third test case are exactly the same to the first test case.

Note that in the third test case only the remaining load should be added to the model. Whenever the new loading stage is used, the previous loading are all finished, which means that the static loading becomes constant and the dynamic loading vanishes.

410.3 Day 2: Seismic Motions

410.3.1 Deconvolution and Propagation of 1C Motions, 1D Model

Various deconvolution and propagation 1D models for one component (1C) wave propagation are provided through links below.

Note: Please make sure that the input acceleration record is baseline corrected and the displacement record has no permanent deformation. Otherwise, the unrealistic high frequency components can be brought into the simulation results.

- Deconvolution of Ormsby wavelet, input files are available HERE.
- Deconvolution of Northridge earthquake, input files are available HERE.
- Deconvolution of and DRM propagation of Ormsby wavelet, input files are available HERE.
- Deconvolution of and DRM propagation of Northridge earthquake, input files are available HERE.

410.3.2 Convolution and Propagation of 1C Motions, 1D Model

Various convolution and propagation 1D models for one component (1C) wave propagation are provided through links below:

Note: Please make sure that the input acceleration record is baseline corrected and the displacement record has no permanent deformation. Otherwise, the unrealistic high frequency components can be brought into the simulation results.

- Convolution of Ormsby wavelet, input files are available HERE.
- Convolution of Northridge earthquake, input files are available HERE.
- Convolution of and DRM propagation of Ormsby wavelet, input files are available HERE.
- Convolution of and DRM propagation of Northridge earthquake, input files are available HERE.

410.3.3 Convolution, Deconvolution and Propagation of 1C Motions, 2D Model

Various convolution, deconvolution and propagation 2D models for one component (1C) wave propagation are provided through links below.

Note #1: Please make sure that the input acceleration record is baseline corrected and the displacement record has no permanent deformation. Otherwise, the unrealistic high frequency components can be brought into the simulation results.

Note #2: Please make sure that you develop seismic motions by doing deconvolution and then convolution before analyzing the actual model. File `run.sh` in examples directory has a proper sequence of commands, that is one should first run Real-ESSI on `Deconvolution_DRM_motion.fei` and then, when motions are developed, analyze model.

Examples are available through links below:

- Convolution/Deconvolution of and DRM propagation of Ormsby wavelet, input files are available HERE.
- Convolution/Deconvolution of and DRM propagation of Kobe earthquake records, input files are available HERE.

410.3.3.1 ESSI 3D building model, deconvolution 1C model, shell model with DRM

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

The Modeling parameters are listed below:

- Elastic Soil Material Properties
  - Mass density, ρ, 2000 kg/m^3
  - Shear Wave Velocity, V_s, 500 m/s
  - Young's modulus, E, 1.1 GPa
  - Poisson's ratio, ν, 0.1
- Elastic Structure Material Properties
  - Mass density, ρ, 2500 kg/m^3
  - Young's modulus, E, 20 GPa
  - Poisson's ratio, ν, 0.1

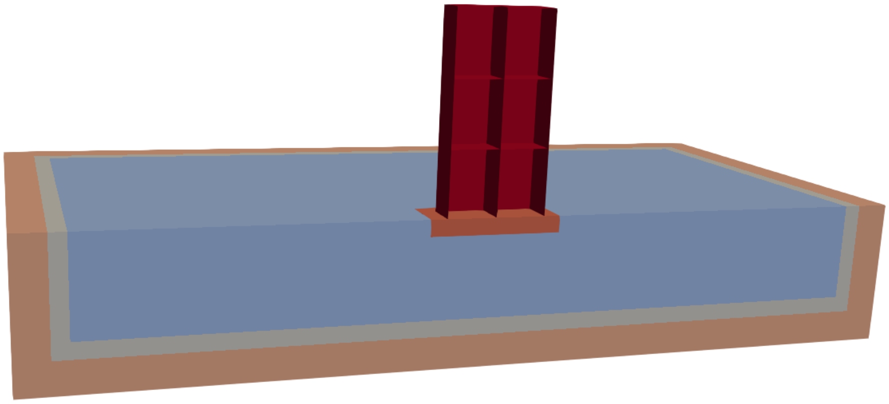


Figure 410.51: Simulation Model.

Results of DRM 3D shell Structure Model under 1C motion are shown in Fig. 410.52.

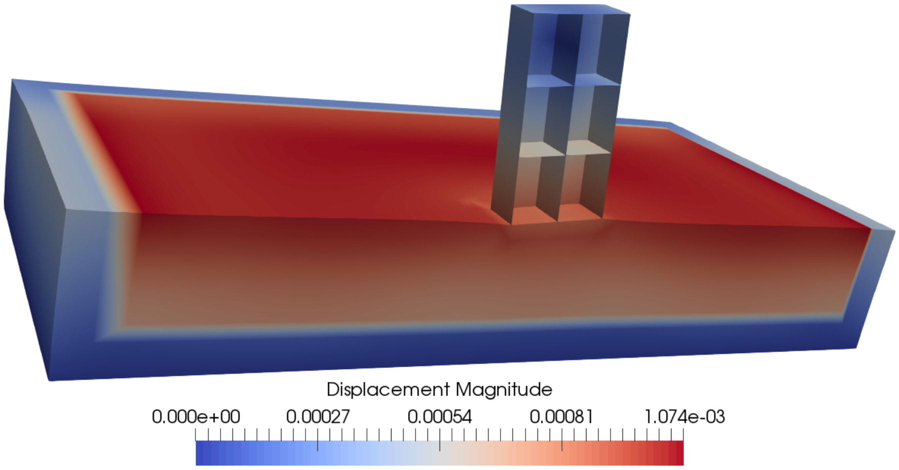


Figure 410.52: Simulation Model.

410.3.4 Deconvolution 3×1C Motions

410.3.4.1 Free field 1C model, deconvolution 3×1C motion, model with DRM

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

The Modeling parameters are listed below:

- Elastic Material Properties
  - Mass density, ρ, 2000 kg/m^3
  - Shear Wave Velocity, V_s, 500 m/s
  - Young's modulus, E, 1.1 GPa
  - Poisson's ratio, ν, 0.1

Figure 410.53: Simulation Model.

Results of the simulation are shown in Fig. 410.1.

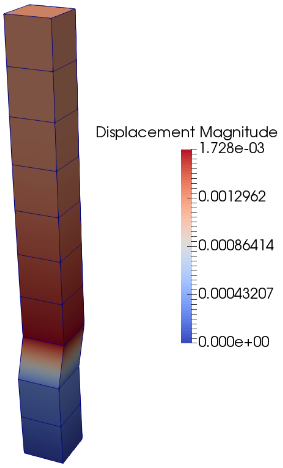


Figure 410.54: Simulation Model.

410.3.4.2 Free field 3D model, deconvolution 3×1C motion, model with DRM

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

The Modeling parameters are listed below:

- Elastic Soil Material Properties
  - Mass density, ρ, 2000 kg/m^3
  - Shear Wave Velocity, V_s, 500 m/s
  - Young's modulus, E, 1.1 GPa
  - Poisson's ratio, ν, 0.1

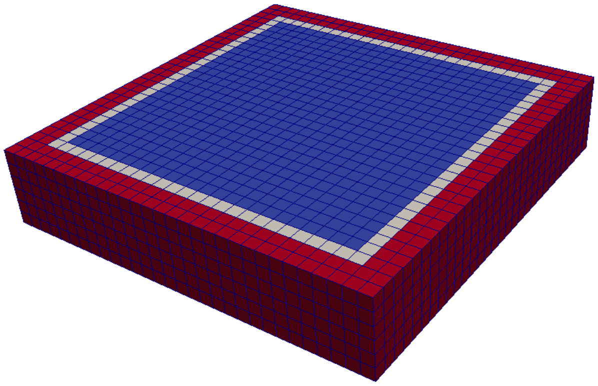


Figure 410.55: Simulation Model.

Results of the simulation are shown in Fig. 410.56.

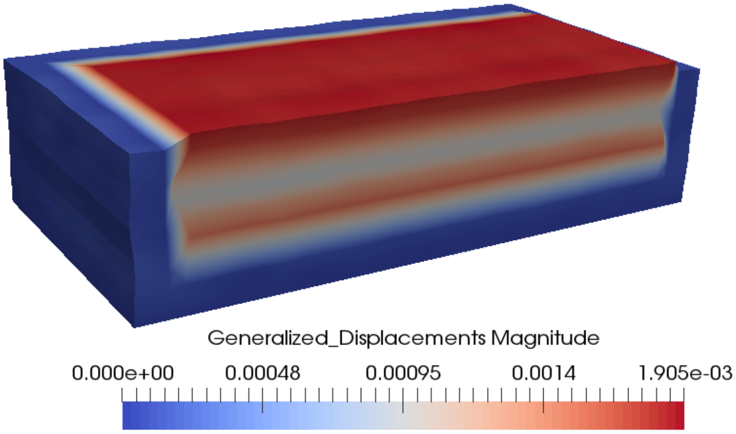


Figure 410.56: Simulation Model.

410.3.4.3 ESSI 3D building model, deconvolution 3×1C motion, shell model with DRM

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

The Modeling parameters are listed below:

- Elastic Soil Material Properties
  - Mass density, ρ, 2000 kg/m^3
  - Shear Wave Velocity, V_s, 500 m/s
  - Young's modulus, E, 1.1 GPa
  - Poisson's ratio, ν, 0.1
- Elastic Structure Material Properties
  - Mass density, ρ, 2500 kg/m^3
  - Young's modulus, E, 20 GPa
  - Poisson's ratio, ν, 0.1

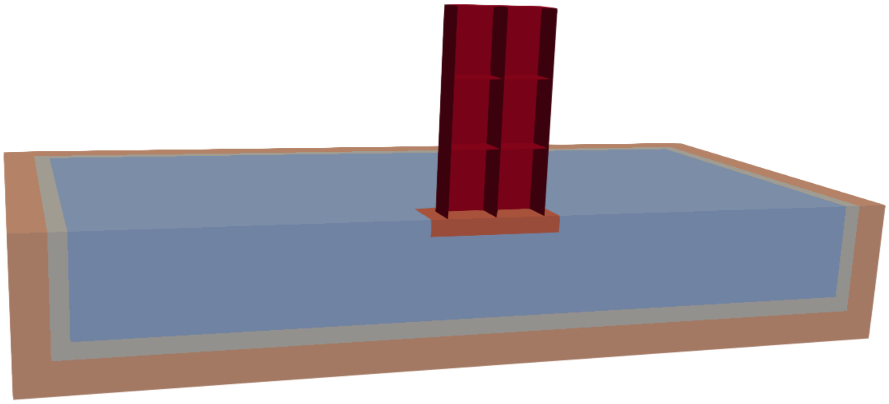


Figure 410.57: Simulation Model.

Results of DRM 3D shell Structure Model under 1C motion are shown in Fig. 410.58.

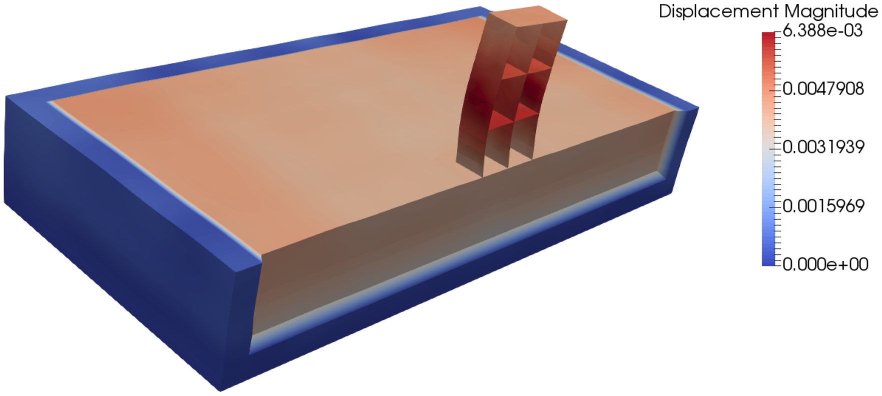


Figure 410.58: Simulation Model.

410.3.5 Mesh Dependence of Wave Propagation Frequencies

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

Show the mesh dependence of high frequency wave with Ormsby wavelet.

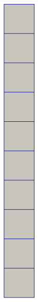


Figure 410.59: Simulation Model.

Results of mesh dependence are shown in Fig. 410.60.

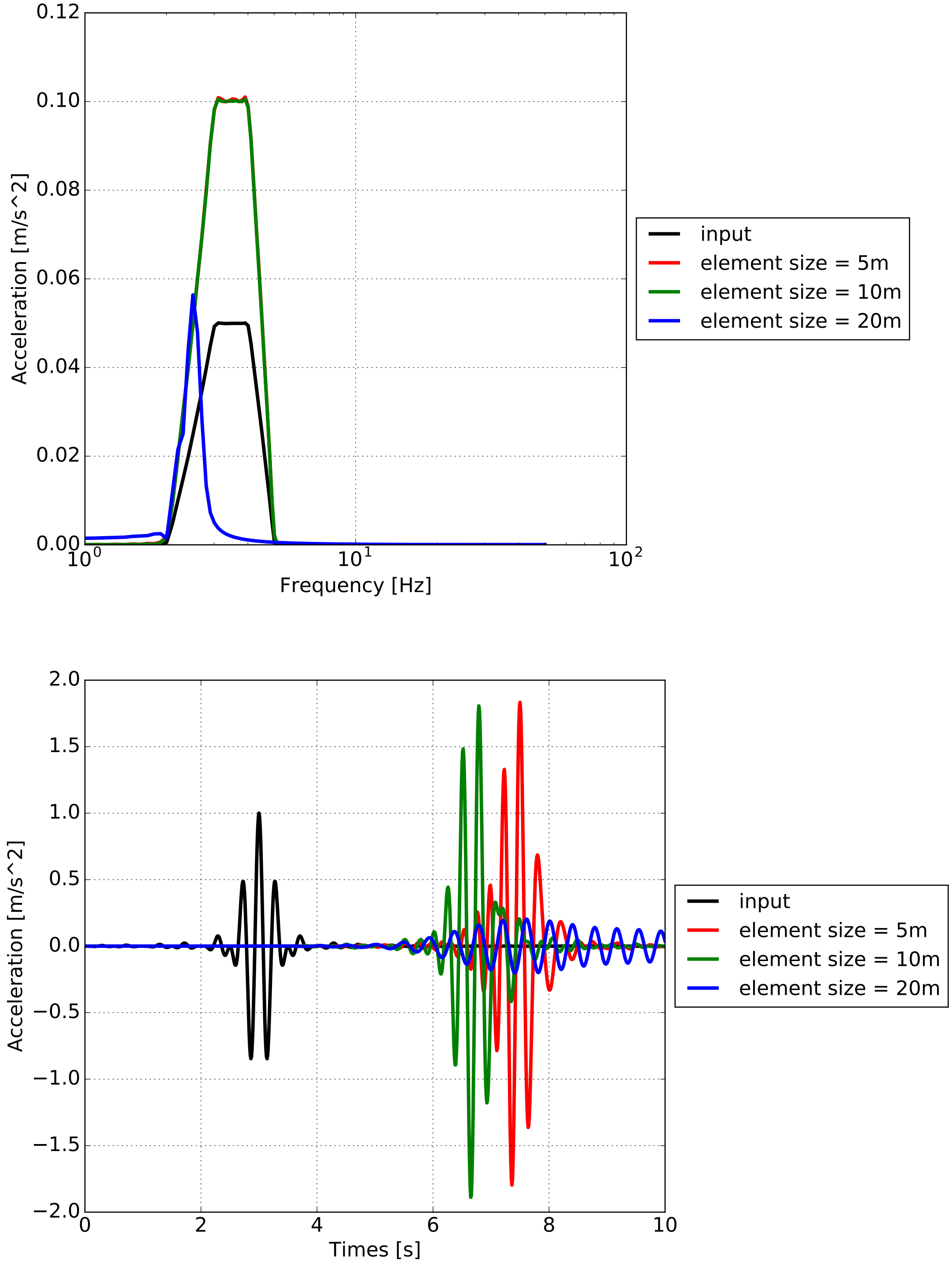


Figure 410.60: Convolution Results and Mesh Dependence.

410.3.6 Application of 3C Motions from SW4

410.3.6.1 3C Seismic Motion from SW4

A 3C seismic motion field has been developed by using SW4. The characteristic parameters of the seismic motion are given below:

- Geological model: length $3km$, width $3km$, height $1.7km$, grid size $50m$, width of super grid damping layer $30m$.
- Material model: Elastic material, First $1km$: $V_p = 4630.76m/s$, $V_s = 2437.56m/s$, $\rho = 2600kg/m^3$. $1km \sim 1.7km$: $V_p = 6000m/s$, $V_s = 3464m/s$, $\rho = 2700kg/m^3$
- Source type: point moment source, moment seismic moment $M_{xy} = 5e^{15} N \cdot m$, moment magnitude 4.5.
- Time function: Gaussian function, with dominant frequency $2.5Hz$ and maximum frequency $6.5Hz$.

The time series displacement and acceleration response at the center of the model is shown below in figure 410.61. And figure 410.62 gives corresponding FFT response.

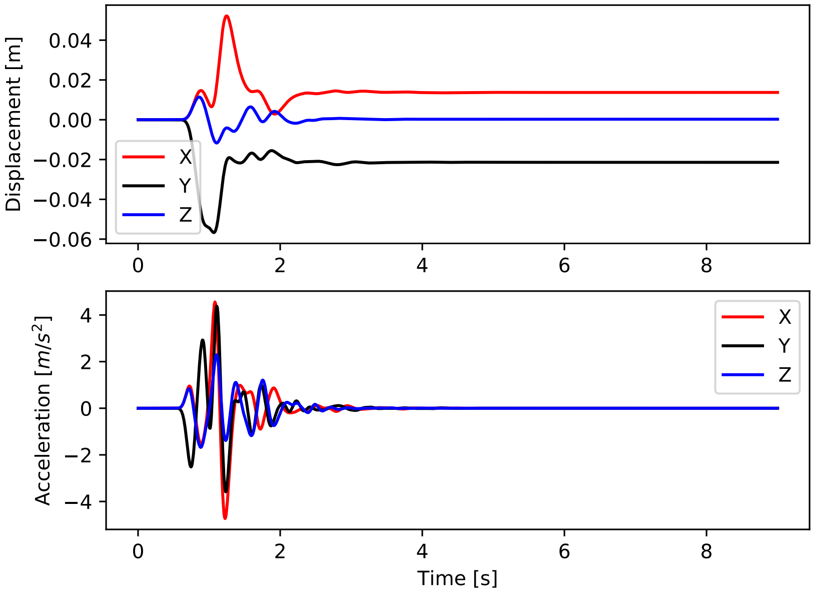


Figure 410.61: Time series response of 3C motion.

During the simulation of SW4, the time series motions at many ESSI nodes (basically are some pre-defined record stations) of an ESSI box ($300m \times 300m \times 100m$) are recorded and written into SAC

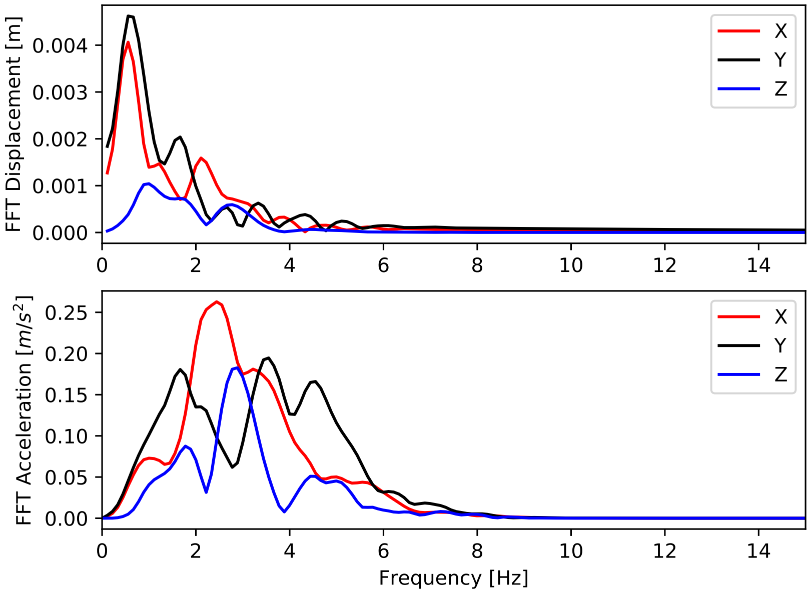


Figure 410.62: FFT response of 3C motion.

files. Then an transition program SW42ESSI has been developed to interpolate these motions to DRM nodes of localized ESSI model by specifying some geometric translational and rotational transformation, as shown in figure ??.

To launch SW42ESSI, following parameters are needed:

- DRM input: specify the name of DRM input files. This DRM file just contains the geometric information of DRM layer in ESSI model (e.g. DRM node IDs, nodal coordinates, etc).
- SW4 motion directory: specify the output directory of SW4, that contains SAC files.
- origin coordinates of ESSI box (x, y, z): the SW4 coordinates of the origin of ESSI box, i.e. the coordinates of ESSI nodes, whose station ID is (0, 0, 0).
- dimensions of ESSI Box (length, width, height): specify the dimension (length, width and height) of ESSI box.
- spacing of ESSI nodes: specify the grid spacing of ESSI nodes (i.e. motion recording stations)
- interval of time steps for sampling: specify the sampling frequency, if 1 is used here, ESSI simulation time step is the same as the simulation time step of SW4.

- reference point in ESSI model for translational transformation (x, y, z): specify the coordinate of reference point for translational transformation in ESSI model.
- reference point in SW4 model for translational transformation (x, y, z): specify the coordinate of reference point for translational transformation in SW4 model.
- conduct rotational transformation (yes/no): input yes and provide more rotational transformation parameters to enable rotational transformation. If input no, no more parameters are required.
- reference point in SW4 model for rotational transformation (x, y, z): specify the coordinate of reference point for rotational transformation in SW4 model.
- degrees of rotation along three axes (x, y, z): specify the degrees of rotation along three axes. The sign of rotation degrees follows right hand rule.

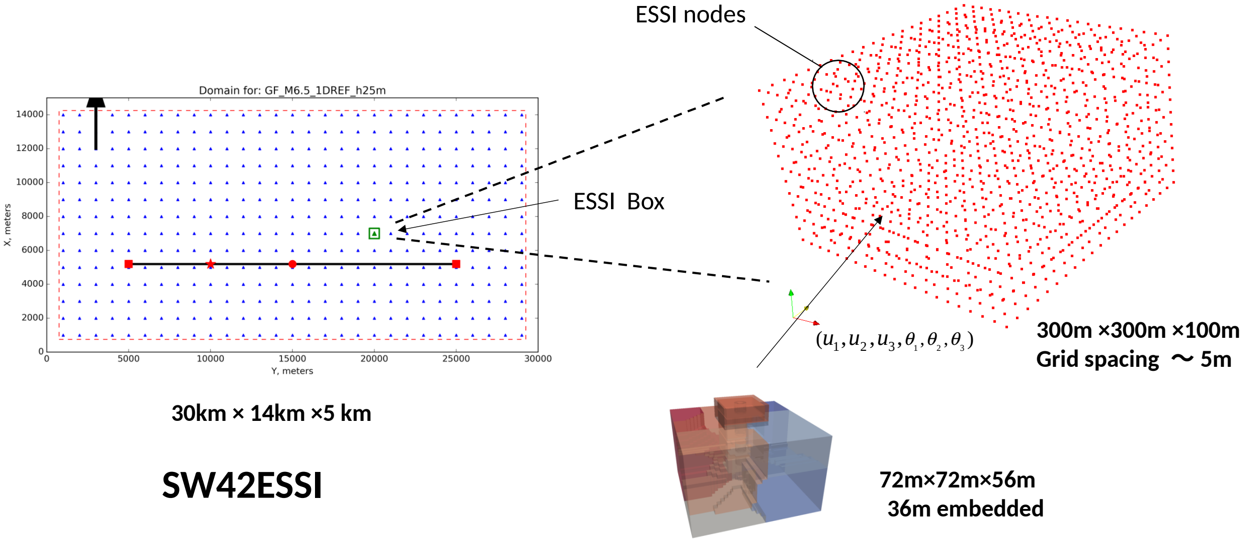


Figure 410.63: Illustration of transition from SW4 to Real-ESSI.

410.3.6.2 Free field 3D model, 3C motion, model with DRM

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

Results of free field DRM 3D Model under 3C motion are shown in figure 410.65.

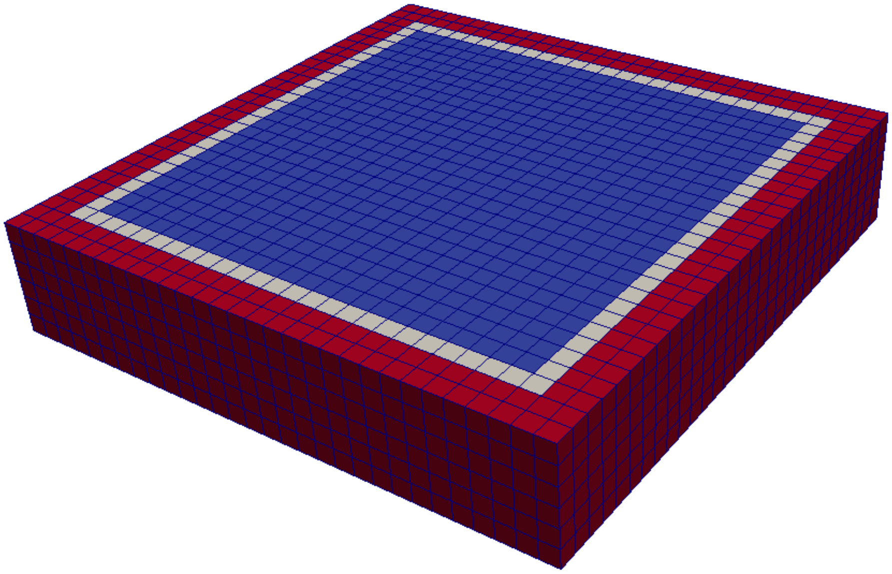


Figure 410.64: Simulation Model.

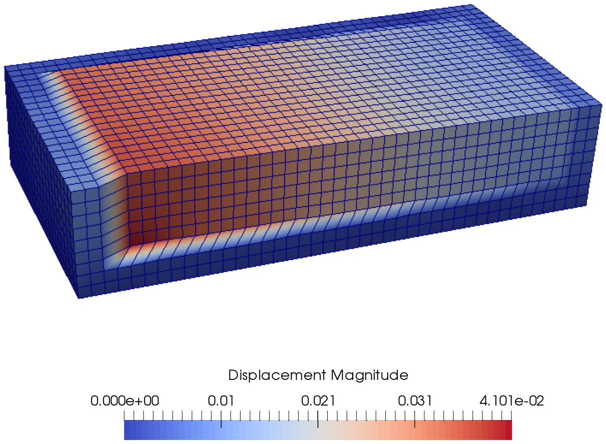


Figure 410.65: Simulation of 3D free field model under 3C seismic motion.

410.3.6.3 ESSI 3D building model, 3C motion, shell model with DRM

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

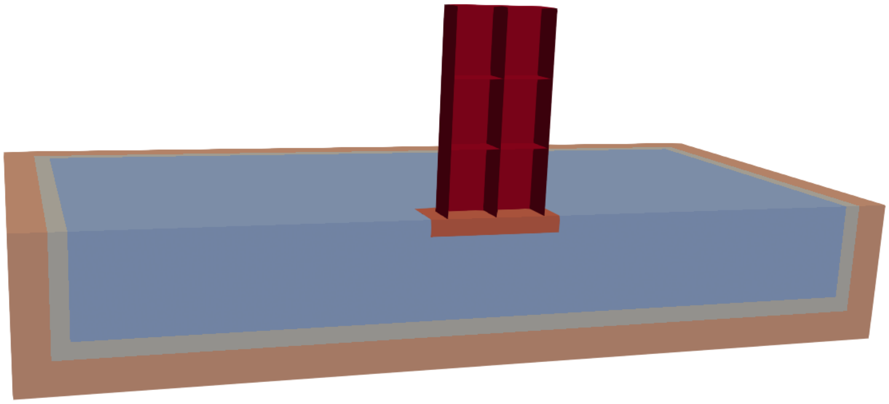


Figure 410.66: Simulation Model.

410.4 Day 3: Inelastic, Nonlinear Analysis

410.4.1 Single Element Models: Illustration of the Elastic-Plastic Behavior

410.4.1.1 von-Mises Perfectly Plastic Material Model.

The Real-ESSI input files for von-Mises perfectly plastic example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

410.4.1.2 von-Mises Armstrong-Frederick Material Model.

The Real-ESSI input files for von-Mises Armstrong-Frederick example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

The Modeling parameters are listed below:

- Left: von-Mises linear hardening material model
    - Mass Density, ρ, 0.0 kg/m^3
    - Young's modulus, E, 20 MPa
    - Poisson's ratio, ν, 0.0
    - von Mises radius, k, 100 kPa
    - kinematic hardening rate, K_{kine}, 2 MPa
    - isotropic hardening rate, K_{iso}, 0 Pa
- Right: Drucker-Prager nonlinear hardening material model
    - Mass Density, ρ, 0.0 kg/m^3
    - Young's modulus, E, 20 MPa
    - Poisson's ratio, ν, 0.0
    - Drucker-Prager, k, 0.179527
    - nonlinear kinematic hardening, H_a, 20 MPa
    - nonlinear kinematic hardening, C_r, 100
    - isotropic hardening rate, K_{iso}, 0 Pa
    - initial confining stress, p_0, 1 Pa

Results are shown in Fig. 410.68.

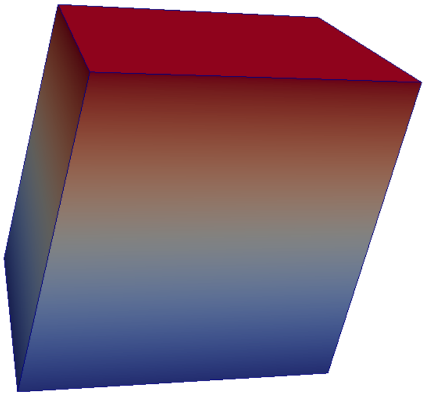


Figure 410.67: Simulation Model of Single Element.

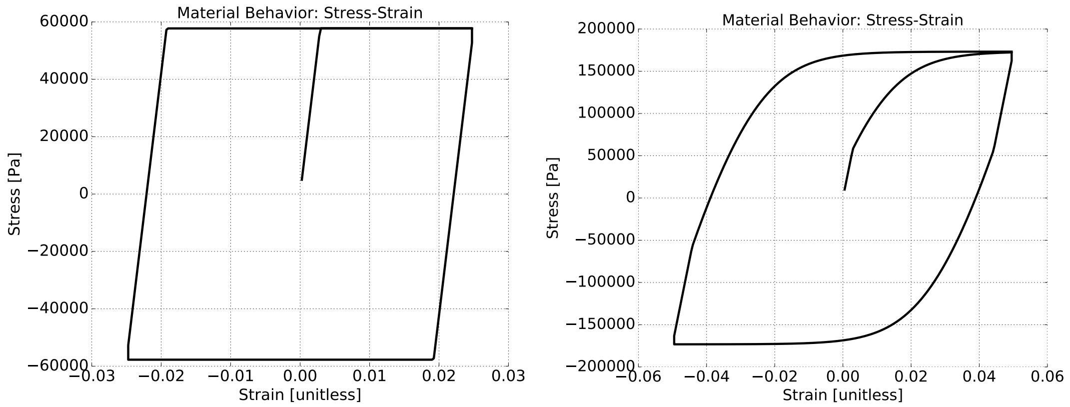


Figure 410.68: Simulation Results of Single Element.

410.4.1.3 von-Mises G/Gmax Material Model

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

The Modeling parameters are listed below:

- von-Mises G/Gmax material model
  - Mass density, ρ, 2000 kg/m^3
  - Young's modulus, E, 200 MPa
  - Poisson's ratio, ν, 0.1
  - Total number of shear modulus 9
  - G over Gmax, 1,0.995,0.966,0.873,0.787,0.467,0.320,0.109,0.063
  - Shear strain gamma, 0,1E-6,1E-5,5E-5,1E-4, 0.0005, 0.001, 0.005, 0.01

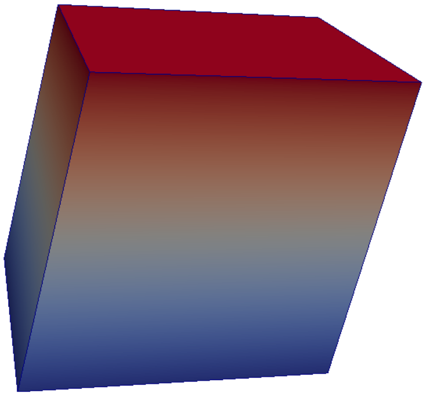
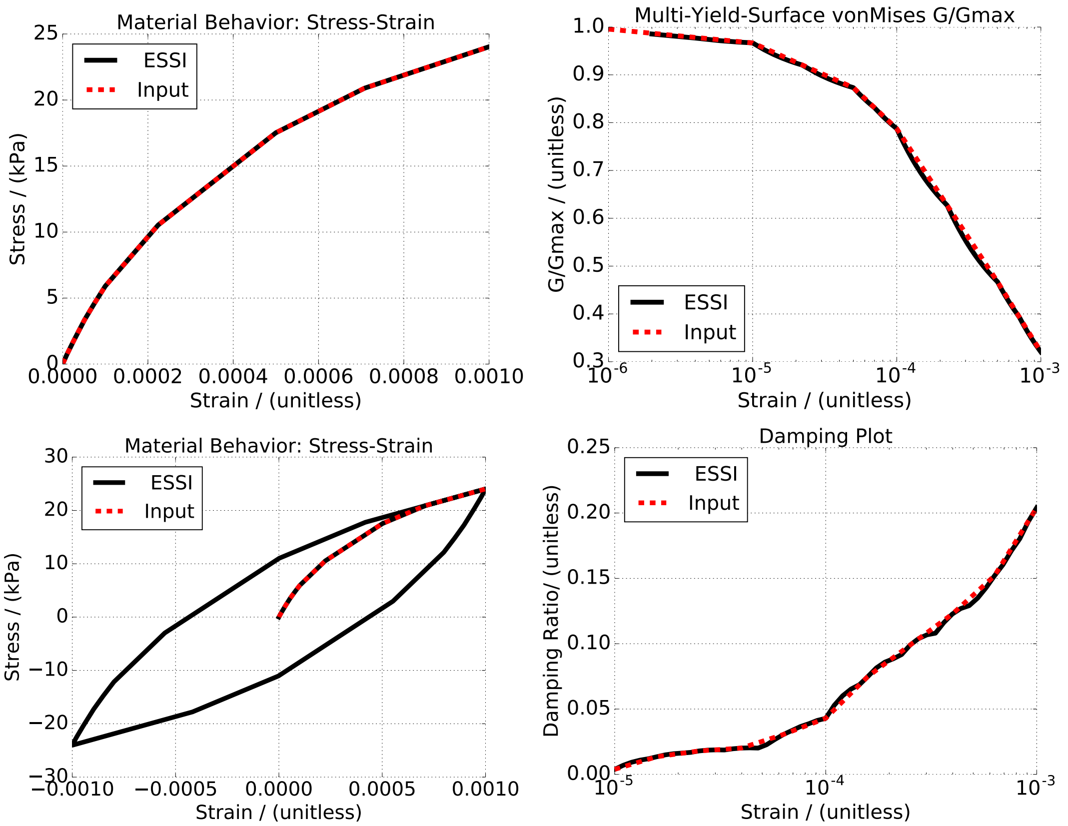


Figure 410.69: Simulation Model of Single Element.


Material Behavior: Stress-Strain
ESSI
Input
Stress / (kPa)
0
5
10
15
20
25
0.0000
0.0002
0.0004
0.0006
0.0008
0.0010
Strain / (unitless)
Multi-Yield-Surface vonMises G/Gmax
G/Gmax / (unitless)
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
10^{-6}
10^{-5}
10^{-4}
10^{-3}
Strain / (unitless)
ESSI
Input
Material Behavior: Stress-Strain
ESSI
Input
Stress / (kPa)
−30
−20
−10
0
10
20
30
−0.0010
−0.0005
0.0000
0.0005
0.0010
Strain / (unitless)
Damping Plot
ESSI
Input
Damping Ratio/ (unitless)
0.00
0.05
0.10
0.15
0.20
0.25
10^{-5}
10^{-4}
10^{-3}
Strain / (unitless)

410.4.1.4 Drucker-Prager Perfectly Plastic Material Model

The Real-ESSI input files for this Drucker-Prager perfectly plastic example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

410.4.1.5 Drucker-Prager Armstrong-Frederick Non-Associated Material Model

The Real-ESSI input files for this Drucker-Prager Armstrong-Frederick example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

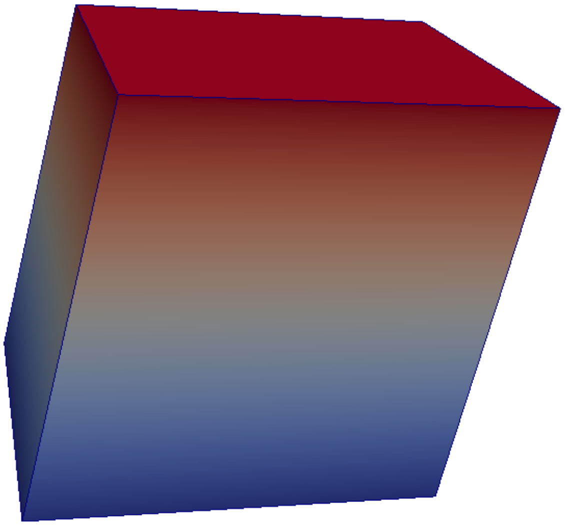


Figure 410.70: Simulation model, single element.

Results are shown in Fig. 410.71.

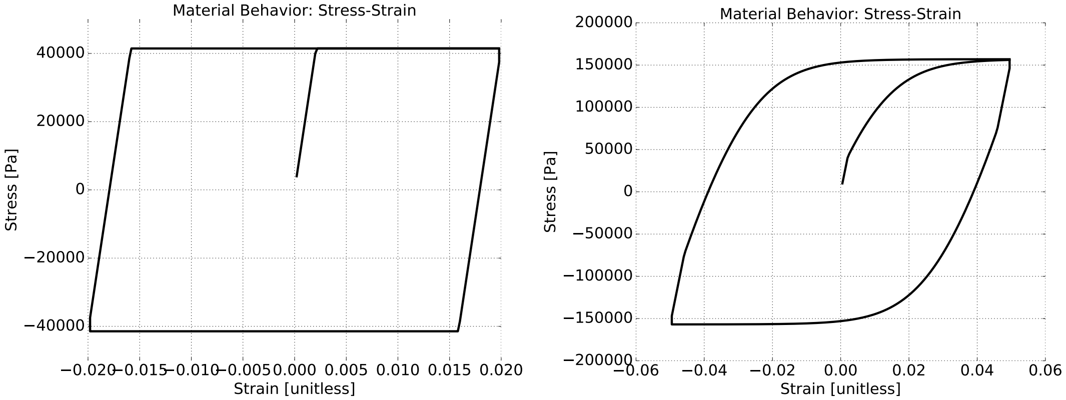


Figure 410.71: Simulation results for single element.

410.4.1.6 Drucker-Prager G/Gmax Non-Associated Material Model

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

The Modeling parameters are listed below:

- Drucker-Prager G/Gmax material model
  - Mass density, ρ, 2000 kg/m^3
  - Young's modulus, E, 200 MPa
  - Poisson's ratio, ν, 0.1
  - Initial confining stress, p_0, 100 kPa
  - Reference pressure, p_{refer}, 100 kPa
  - Pressure exponential, n, 0.5
  - Cohesion, n, 1 kPa
  - Total number of Shear Modulus 9
  - G over Gmax, 1,0.995,0.966,0.873,0.787,0.467,0.320,0.109,0.063
  - Shear strain gamma, 0,1E-6,1E-5,5E-5,1E-4, 0.0005, 0.001, 0.005, 0.01

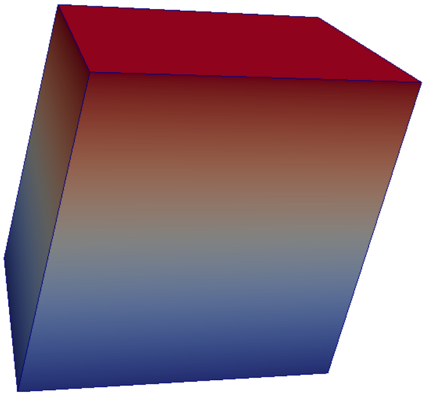


Figure 410.72: Simulation Model of Single Element.

Results are shown in Fig. 410.73.

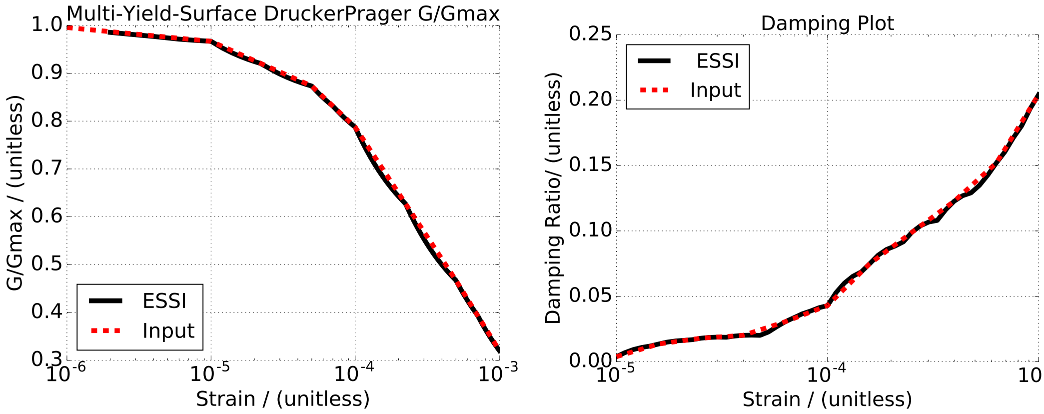


Figure 410.73: Simulation Results of Single Element.

410.4.2 Wave Propagation Through Elasto-plastic Soil

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

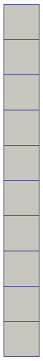


Figure 410.74: Wave Propagation through elastoplastic Soils.

The displacement series at the surface are plotted in time and frequency domain.

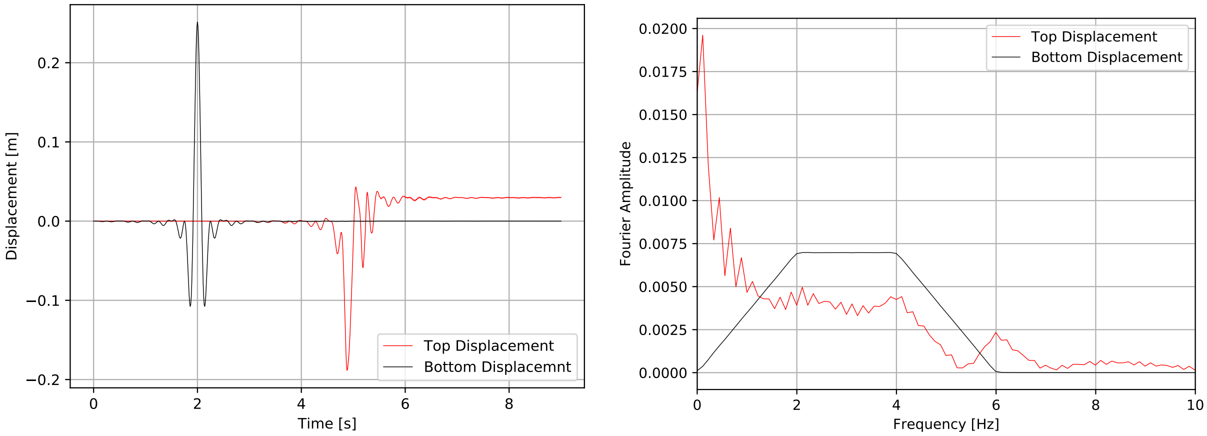


Figure 410.75: Simulation Results of Wave Propagation.

410.4.3 Contact/Interface/Joint Examples

410.4.3.1 Axial Behavior: Stress-Based Hard Contact/Interface/Joint Example

The Real-ESSI input files for hard contact/interface example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

410.4.3.2 Axial Behavior: Stress-Based Soft Contact/Interface/Joint Example

The Real-ESSI input files for soft contact/interface example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

The axial behavior of hard contact/interface and soft contact/interface is illustrated in Fig. 410.76.

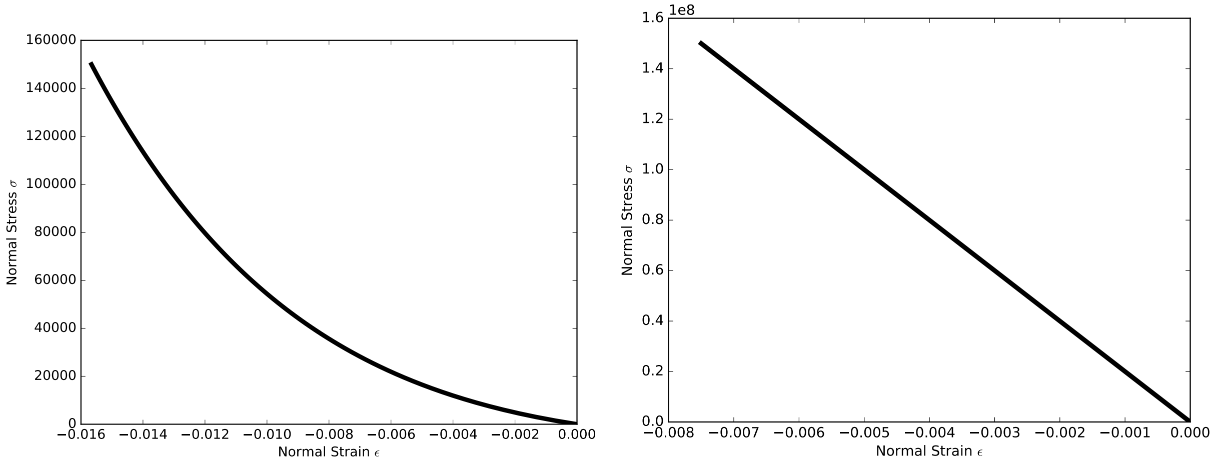


Figure 410.76: Simulation results for axial behavior of (left) soft contact/interface and (right) hard contact.

410.4.3.3 Shear behavior: Stress-Based Elastic Perfectly Plastic Contact/Interface/Joint

The Real-ESSI input files for the the elastic-perfectly plastic example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

410.4.3.4 Shear behavior: Stress-Based Elastic-Hardening Contact/Interface/Joint

The Real-ESSI input files for the elastic-hardening contact/interface example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

410.4.3.5 Shear behavior: Stress-Based Elastic-Hardening-Softening Contact/Interface/Joint

The Real-ESSI input files for the elastic-hardening-softening example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

The shear behavior of elastic-perfectly plastic, elastic-hardening plastic, elastic and hardening and softening plastic is illustrated in Fig. 410.77.

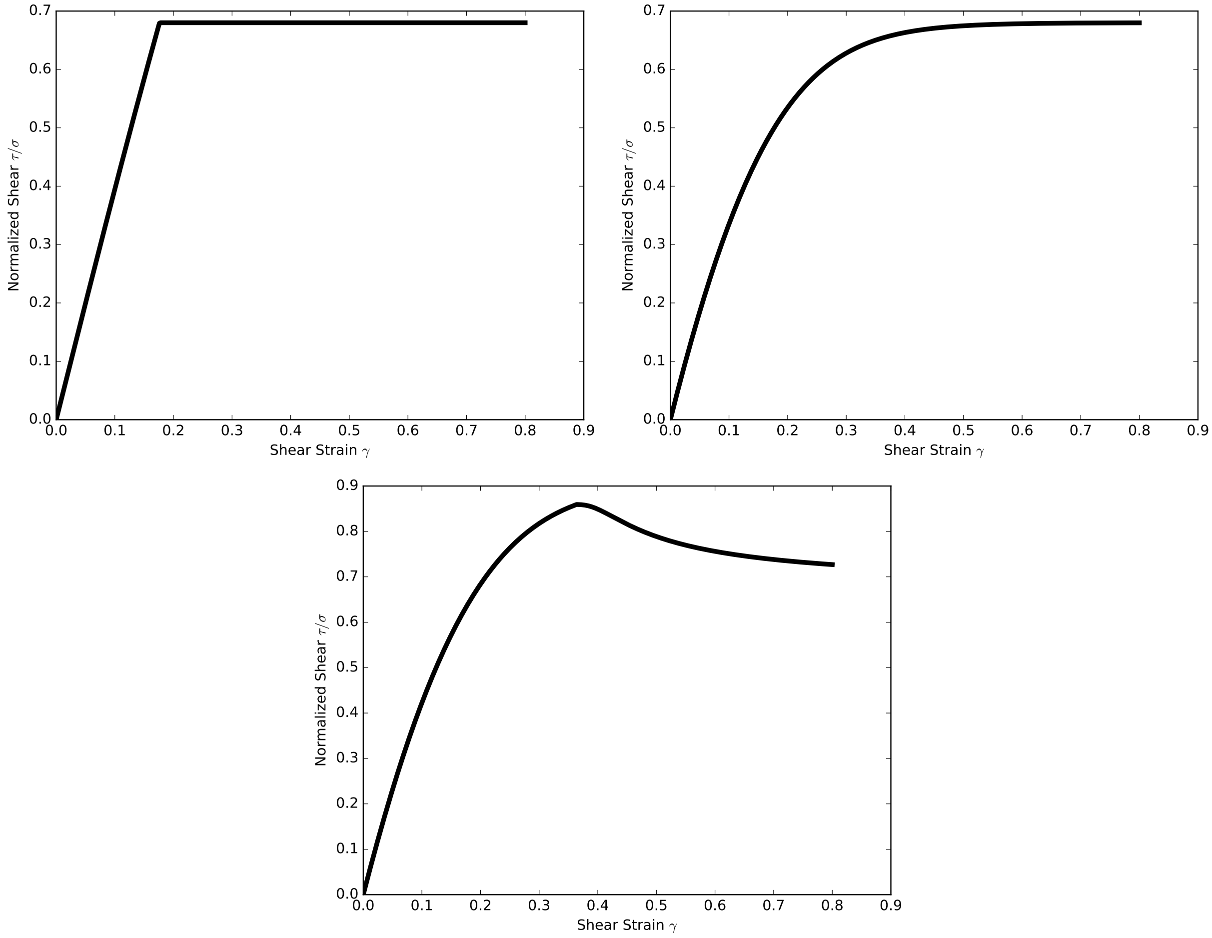


Figure 410.77: Simulation results for shear behavior for stress based contac.t elements: elastic-perfectly plastic, elastic-hardening plastic, elastic, hardening and softening plastic.

410.4.3.6 Force Based Contact/Interface/Joint Example: Base Isolator

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

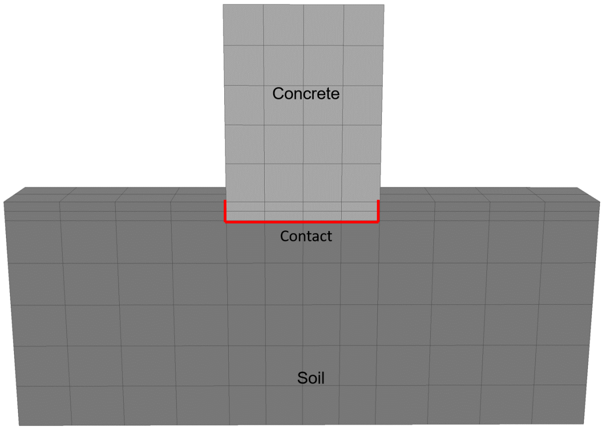


Figure 410.78: Simulation Model.

Results are show in Fig.410.79.

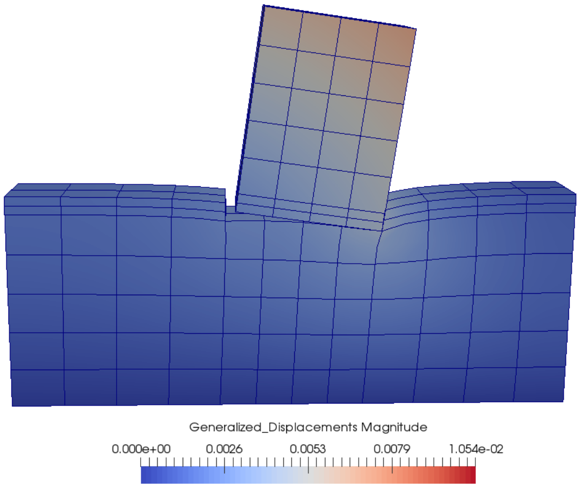


Figure 410.79: Simulation Results for Contact/Interface/Joint Examples.

410.4.4 Inelastic Frame Pushover

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

The Modeling parameters are listed below:

- Uniaxial concrete
    - Compressive strength, 24 MPa
    - Strain at compressive strength, 0.001752
    - Crushing strength, 0.0 Pa
    - Strain at compressive strength, 0.003168
    - lambda, 0.5
    - Tensile strength, 0 Pa
    - Tension softening stiffness, 0 Pa
- Uniaxial steel
    - Yield strength, 413.8 MPa
    - Young's modulus, 200 GPa
    - Strain hardening ratio, 0.01
    - R0, 18.0
    - cR1, 0.925
    - cR2, 0.15
    - a1, 0.0
    - a2, 55.0
    - a3, 0.0
    - a4, 55.0

Result is shown in Fig. 410.81.

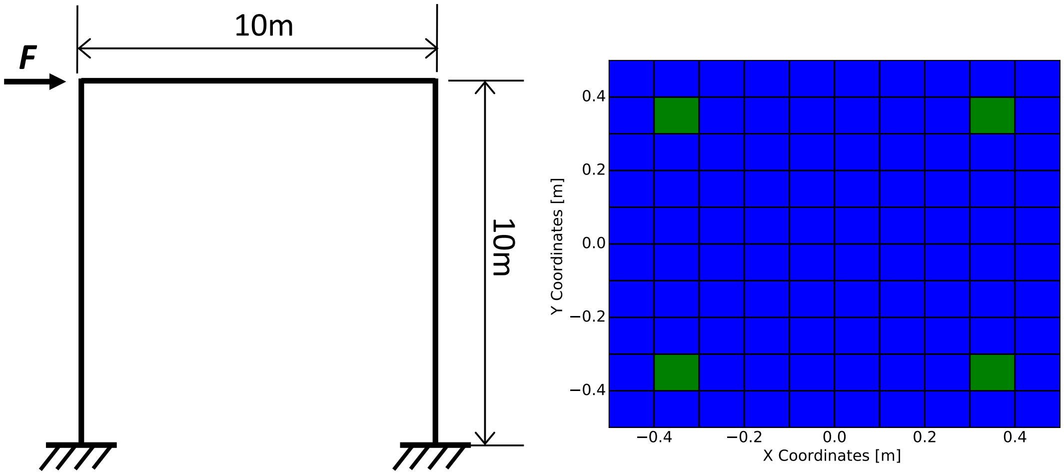


Figure 410.80: Model for pushover simulation and the cross section of fiber beam (concrete and reinforcement).

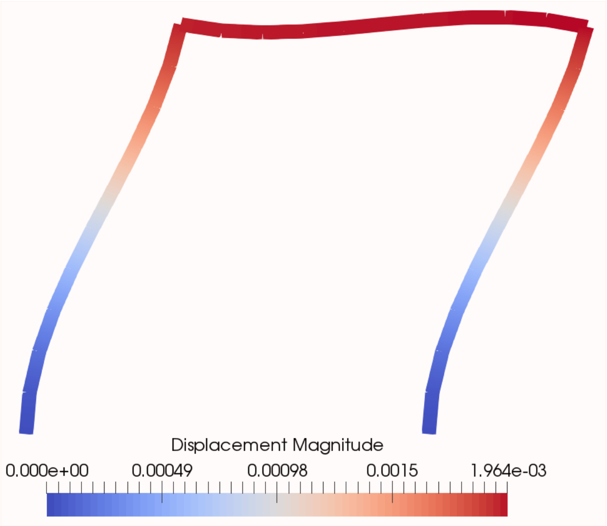


Figure 410.81: Results for fiber pushover.

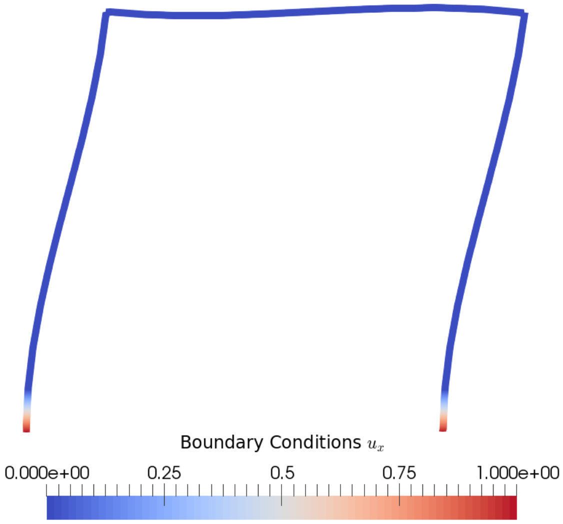


Figure 410.82: Boundary condition u_x for a fiber beam pushover.

410.4.5 Inelastic Wall Pushover

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

The Modeling parameters are listed below:

- Concrete Wall
  - Young's modulus, 36.9 GPa
  - Poisson's ratio, 0.2
  - Tensile yield strength, 5 MPa
  - Compressive yield strength, 56 MPa
  - Plastic deformation rate, 0.4
  - Damage parameter Ap, 0.1
  - Damage parameter An, 1.5
  - Damage parameter Bn, 0.75
- Uniaxial steel
  - Yield strength, 457.5 MPa
  - Young's modulus, 200 GPa
  - Strain hardening ratio, 0.011042
  - a1, 0.0
  - a2, 55.0
  - a3, 0.0
  - a4, 55.0

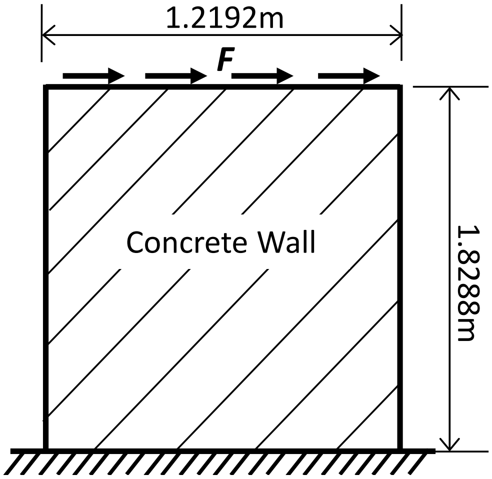


Figure 410.83: Model for wall element pushover.

410.4.6 Viscous Nonlinear behavior

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

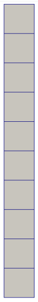


Figure 410.84: Simulation Model.

Result are shown in Fig. 410.85 and Fig. 410.86.

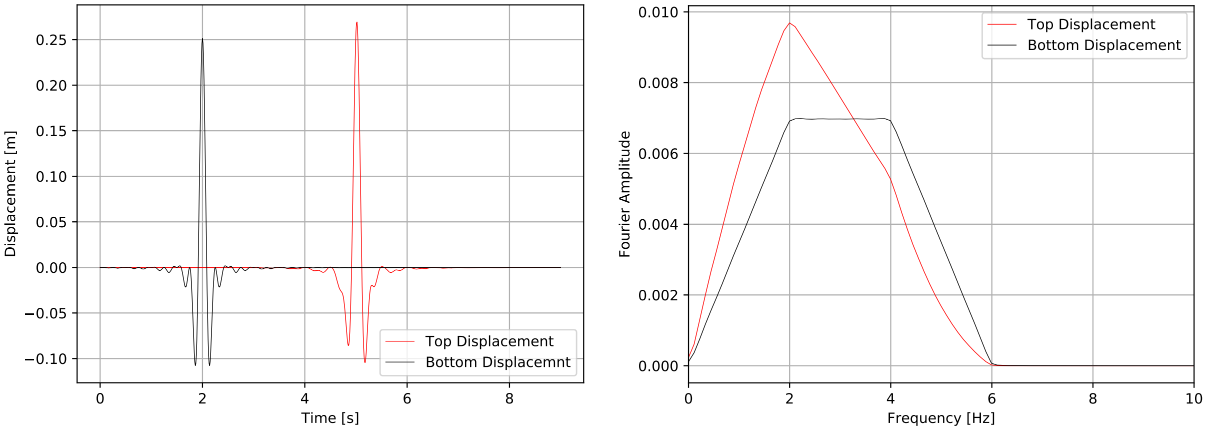


Figure 410.85: Results for low viscous damping.

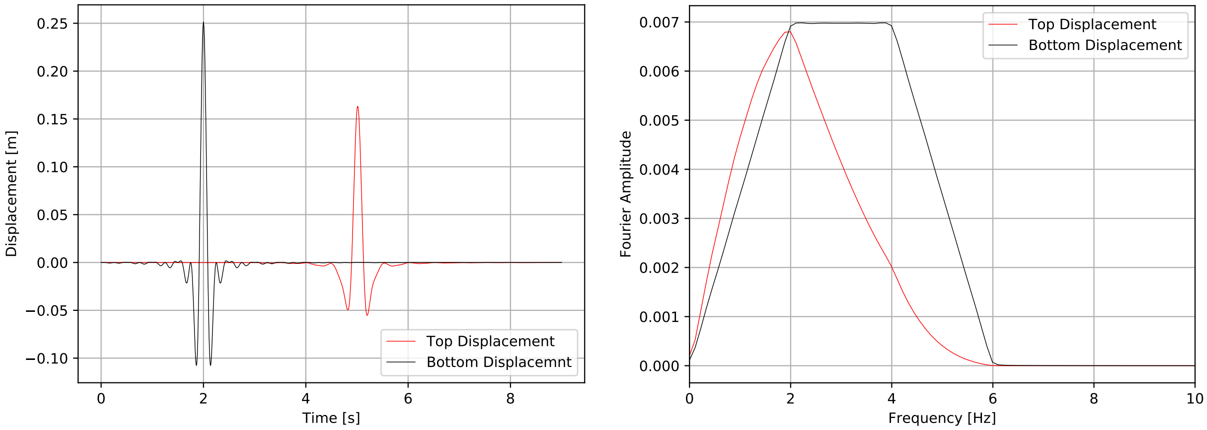


Figure 410.86: Results for high viscous.

410.4.7 Numerical Damping Example

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

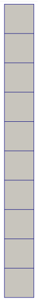


Figure 410.87: Simulation Model.

Result are shown in Fig. 410.85 and Fig. 410.89 .

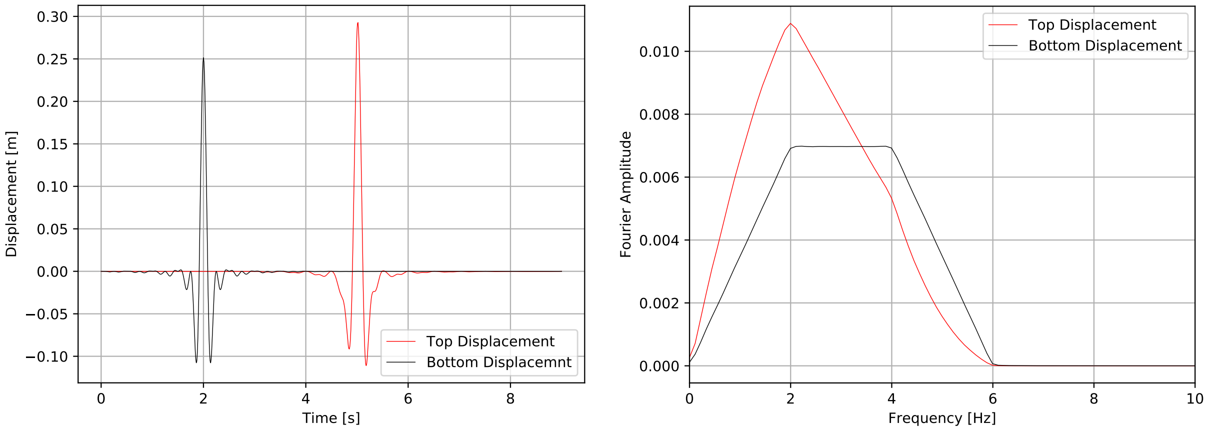


Figure 410.88: Results of low numerical damping.

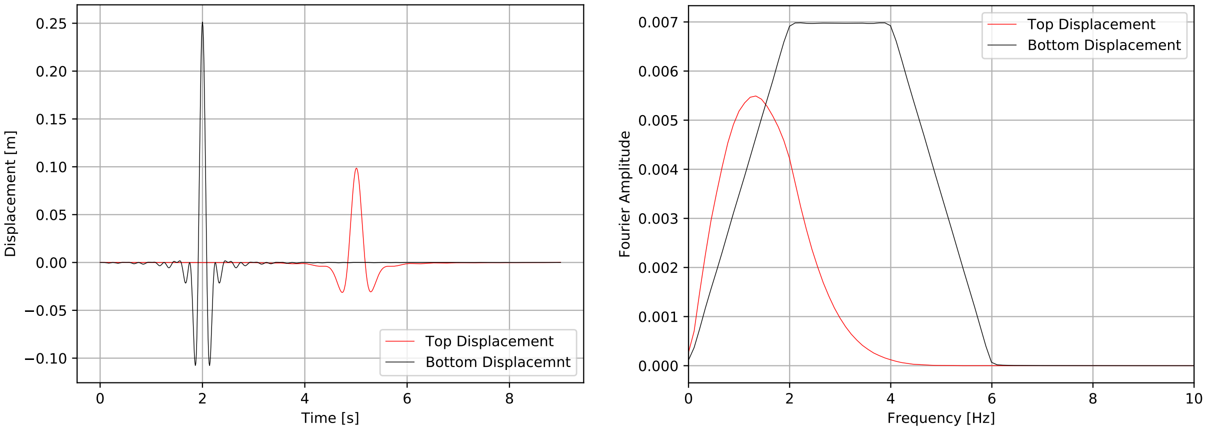


Figure 410.89: Results of high numerical damping.

410.4.8 Nuclear Power Plant Example with Nonlinearities

The Real-ESSI input files for this example are available HERE. The compressed package of Real-ESSI input files for this example is available HERE.

The Modeling parameters are listed below:

- Soil
  - Unit weight, γ, 21.4 kPa
  - Shear velocity, Vs, 500 m/s
  - Young's modulus, E, 1.3 GPa
  - Poisson's ratio, ν, 0.25
  - Shear strength, S_u, 650 kPa
  - von Mises radius, k, 60 kPa
  - kinematic hardening, H_a, 30 MPa
  - kinematic hardening, C_r, 25
- Structure
  - Unit weight, γ, 24 kPa
  - Young's modulus, E, 20 GPa
  - Poisson's ratio, ν, 0.21
- Contact/Interface/Joint
  - Initial axial stiffness, k_n^{init}, 1e9 N/m
  - Stiffening rate, S_r, 1000 /m
  - Maximum axial stiffness, k_n^{max}, 1e12 N/m
  - Shear stiffness, k_t, 1e7 N/m
  - Axial viscous damping, C_n, 100 $N \cdot s/m$
  - Shear viscous damping, C_t, 100 $N \cdot s/m$
  - Friction ratio, μ, 0.25

SIMULATION TIME: With 32 cores on AWS EC2 c4.8xlarge instance, the running time for this example is 30 hours.

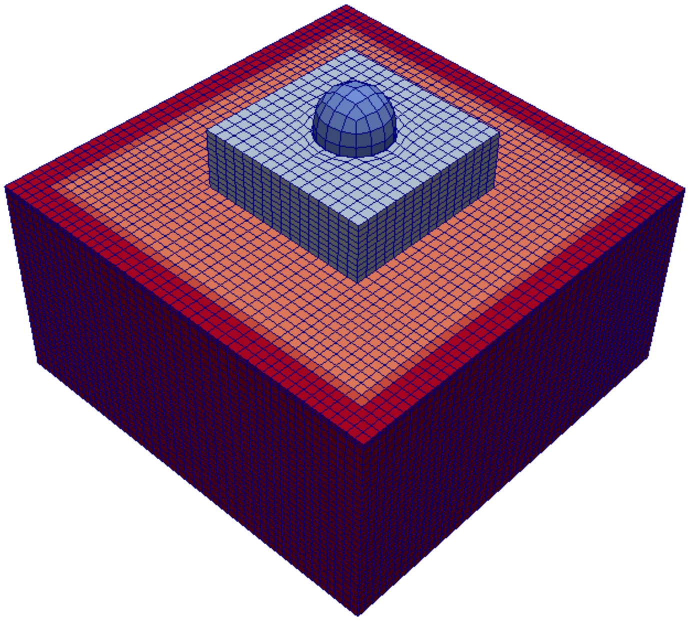


Figure 410.90: Simulation Model.

410.4.9 Buildings, ATC-144/FEMA-P-2091 Examples

The Real-ESSI building examples, models from FEMA-P-2091 report are available in Sections 509.2 (page 2676), 509.4 (page 2684), and 509.5 (page 2688), in Lecture Notes by Jeremić et al. (1989-2025) (Lecture Notes URL).

Part 500

Application to Practical Engineering Problems

Chapter 501

Static Soil-Pile and Soil-Pile Group Interaction in Single Phase Soils

(1999-2002-)

(In collaboration with Prof. Zhaohui Yang)

501.1 Chapter Summary and Highlights

501.2 Numerical Analysis of Pile Behavior under Lateral Loads in Layered Elastic–Plastic Soils

Material presented here has been previously published in our paper Yang and Jeremić (2005).

501.2.1 Introduction

The $p-y$ approach (Reese et al. Reese et al. (2000a)) has been widely used to design piles subjected to lateral loading. Based on the Winkler foundation theory, the method models the lateral soil-structure interaction with empirically derived nonlinear springs. The advancement of computer technology has made it possible to study this problem using more rigorous Finite Element Method (FEM).

Mentioned are a few representative finite element applications. Maqtadir and Desai Muqtadir and Desai (1986) studied the behavior of a pile-group using a three dimensional program with nonlinear elastic soil model. An axisymmetric model with elastic-perfectly plastic soil was used by Pressley and Poulos Pressley and Poulos (1986) to study group effects. Brown and Shie Brown and Shie (1990a) Brown and Shie (1990b) Brown and Shie (1991) and Trochanis Trochanis et al. (1991) conducted a series of 3D FEM studies on the behavior of single pile and pile group with elastic-plastic soil model. In particular, interface element was used to account for pile-soil separation and slippage. Moreover, Brown and Shie derived $p-y$ curves from FEM data, which provide some comparison of the FEM results with the empirical design procedures in use. A number of model tests of free- or fixed-headed pile groups under lateral loading has been simulated by Kimura et al. Kimura et al. (1995) and Wakai et al. Wakai et al. (1999) using 3D elasto-plastic FEM. A good correlation between the experiments and the analysis has been observed in these studies. All these results demonstrated that FEM can capture the essential aspects of the nonlinear problem. It is noted that there is not much literature reporting on FEM studies of pile behavior under lateral loading in layered soil system. In addition to that, there is a very small number of studies on the effects of layering system on the commonly used $p-y$ curve approach.

This paper describes four 3D finite element models of a laterally loaded pile embedded in uniform and layered soil profiles with the dimensions and soil parameters similar to those used in the centrifuge study by McVay et al. McVay et al. (1998) and Zhang et al. Zhang et al. (1999). The bending moments derived by integrating vertical stresses from FEM are numerically differentiated once and twice to compute the shear force and pressure diagrams, respectively. Particularly, $p-y$ curves are generated and cross compared to illustrate the effects of soft clay (sand) layer on the $p-y$ curves of the overlaid sand (soft clay) layer. The results from FEM are also compared with those from centrifuge test and LPILE.

In addition, a limited parametric study of pressure redistribution is conducted by changing the undrained shear strength of the soft clay layer and the friction angle of the sand layer to further investigate the layering effects. An early version of OpenSees OpenSees Development Team (Open Source Project) (2000-2006) finite element program was used in presented computations. Developed models are now available within our new framework FEI. Soil modeling was performed using Template Elastic–Plastic approach (Jeremić and Yang Jeremić and Yang (2002)).

501.2.2 Constitutive Models

Two simple models were used in this numerical study. Specifically, clay was modeled by a simple von Mises material model which is completely defined with the undrained shear strength. Sand was simulated by a Drucker–Prager material model with non-associated flow rule. The reason for using such simple models is that the experimental results used to compare our simulations against did specify only those two material properties for sands and clays. Figure 501.1 presents yield surfaces for both models. In both material models, the Young's moduli vary with confining pressure, as shown in Eqn. (501.1).

$$E = E_o \left(\frac{p}{p_a}\right)^a \tag{501.1}$$

where E_o is Young's Modulus at atmospheric pressure, p is the effective mean normal stresses, p_a is the atmospheric pressure, and a is constant for a given void ratio. In this work, 0.5 was used.

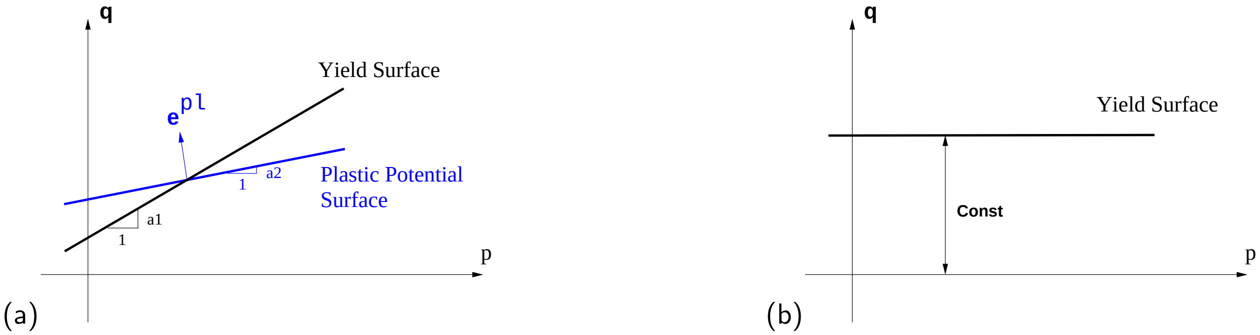


Figure 501.1: Elastic plastic models used in this study: (a) Drucker–Prager model specified with friction angle and dilation angle, and (b) von Mises model specified with undrained shear strength C_u.

The following parameters were used for medium dense sand: friction angle ϕ of 37.1^o, Shear modulus at a depth of 13.7 m of 8960 kPa (E_o = 17400 kPa), Poisson's ratio of 0.35 and unit weight of 14.50 kN/m^3. These parameters were given by Zhang et al. Zhang et al. (1999). A dilation angle of 0^o is used in this work (Brown and Shie Brown and Shie (1990a)). The undrained shear strength, Young's modulus, Poisson's ratio and unit weight of clay were chosen to be 21.7 kPa, 11000 kPa, 0.45, 13.7

kN/m^3, respectively. It should be noted that the above material models are available within the Template Elastic–Plastic Material Modeling paradigm (Jeremić and Yang Jeremić and Yang (2002)). It should also be noted that the use of simple Drucker–Prager model can over-predicted the friction angle to triaxial extension stress path. However this influence is limited to the zone behind the pile, within the interface zone and thus this drawback of the Drucker–Prager model was neglected.

501.2.3 Simulation Results

Presented in this subsection are representative results related to the behavior of piles in uniform and layered soil systems. Presented results are compared with those from the centrifuge study (McVay et al. McVay et al. (1998)), and with results obtained using LPILE program (Reese et al. Reese et al. (2000a,b).

501.2.3.1 Pile Models

A number of static pushover tests for single pile models were simulated using uniform soil and layered soil setups. Figure 501.2 shows the model setups. There are four main setups. Two of these are dealing with uniform sand and clay soils, while two others are featuring layered soils. In particular, the case # 1

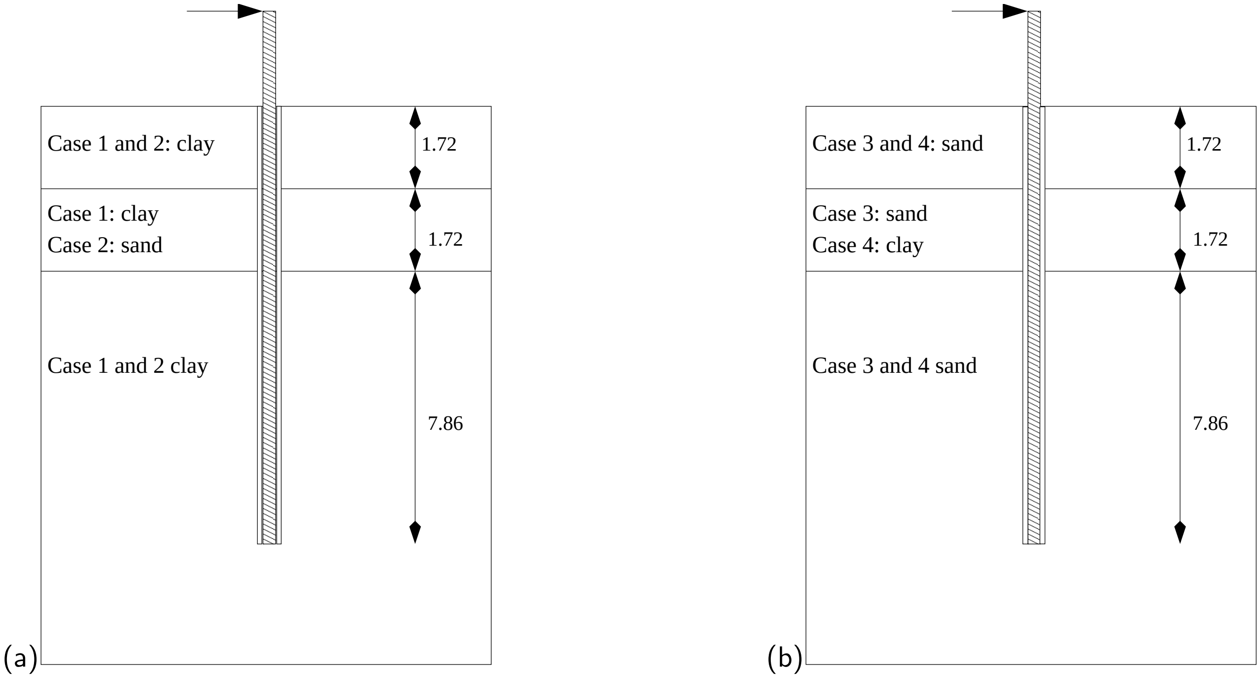


Figure 501.2: (a) Single pile models, dimensions and layers of case #1 and #2. (b) Single pile models, dimensions and layers of case #3 and #4.

is a uniform soft clay soil, case # 2 includes top and bottom layers of soft clay with an in–between layer of medium dense sand. On the other hand, case # 3 features uniform medium dense sand soil, while case # 4 features top and bottom layers of medium dense sand with an in–between layer of soft clay. Detailed layering setup is given in Figure 501.2.

Figure 501.3 shows the finite element mesh for all four cases. Based on symmetry, only half of the model is meshed. Twenty node brick elements are used for both soil, pile and interface. It should be noted that these quadratic elements exhibit high accuracy even for high aspect ratios and can model accurately bending of solid piles with two layers of elements. During mesh design stage, a study was performed to decide on appropriate (balanced) mesh size. That study showed that a much larger mesh, with many more elements (with lower aspect ratios) would account for a fairly small change in results, so it was decided that the current mesh is sufficient for our analysis.

The square pile, with a width of 0.429 m, consist of four elements (per cross subsection) with the elastic property of aluminum. The fine mesh in the upper part of the model is to provide data points for the computation of shear forces and $p-y$ curves of sufficient reliability as well as for the investigation of layering effects. The sides and bottom of the model are fixed with the exception of the symmetric boundary, which is only supported in Y direction. The interface layer between aluminum pile and surrounding soil is represented by one thin layer of elements. The purpose of this layer is to mimic the installation effects on piles (drilled or driven). It also serves a purpose of a simplified interface which allows for tension cut-off (gaping) and controlled, coupled horizontal and vertical stiffness. All interface elements were simulated by Drucker–Prager model with a friction angle of 25^o, and a dilation angle of 0^o.

501.2.3.2 Plastic Zones

The static pushover test were conducted using load control at pile head. The final plastic zones are depicted in Figures 501.4, 501.5. Plastic zones are actually presented by plastified Gauss points. In particular, Figure 501.4(a) shows developed plastic zones for the uniform clay soil (case # 1). It is interesting to note that the plastic zone propagates fairly deep while it does not extend far from the pile in clay. Moreover, compression side (right side) features much larger plastic zone while the plastic zone for the extension side (left side) is confined to the interface layer and a few Gauss points outside the interface layer. The case with clay and sand layer in–between is shown in Figure 501.4(b). The main difference is that the plastic zone is even smaller than for uniform clay layer. It is worth mentioning that this case, which includes sand layer, is stiffer than the uniform clay case, thus displacements are smaller in clay and the plastic zone does not propagate as much as in uniform clay soil.

Figure 501.5(a)(b) shows plastic zones at the end of loading process for sand and sand and clay soils.

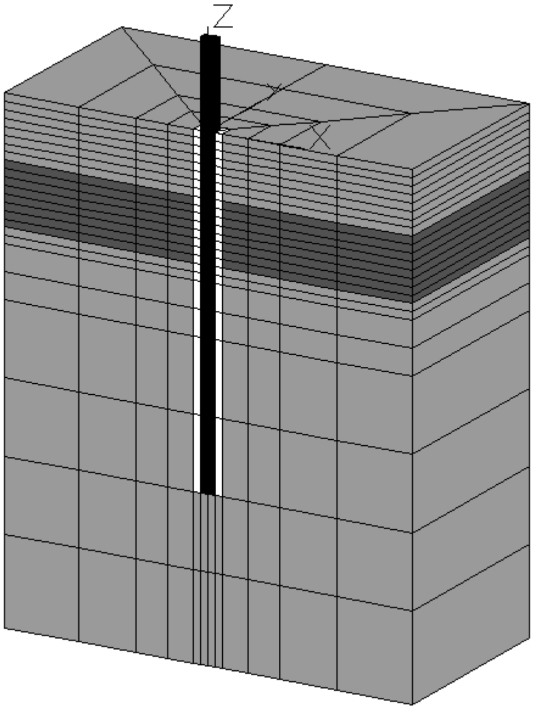
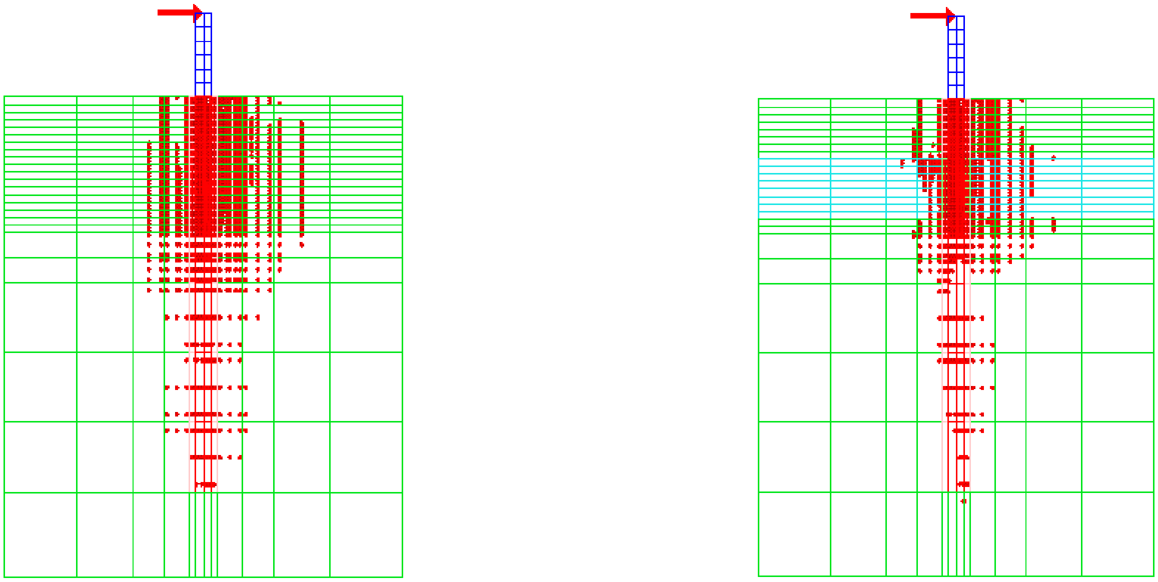


Figure 501.3: Mesh of single pile model, side view, top eight layers of finite elements are either clay or sand (depending on the cases), middle eight layers of finite elements are sand or clay (again depending on the cases) and the bottom is all uniform clay or sand, interface zone around the aluminum pile is also present.

In particular, Figure 501.5(a) shows the plastic zone for uniform sand. It is interesting to note that the plastic zone propagates toward the surface with the collapse mechanics similar to the active and passive failure. In this case of course the system is 3D and so the failure propagation angles do not match the active and passive failure angles, however the difference between active and passive zones propagation angles is almost exactly $\pi/2$. Figure 501.5(b) shows plastic zone for the case # 4 which includes a layer of clay between -1.72m and -3.44m (Z coordinate, origin is in the pile center at the ground surface) . It is noted that the plastic zone is deeper, but not as nicely defined as in the previous case.

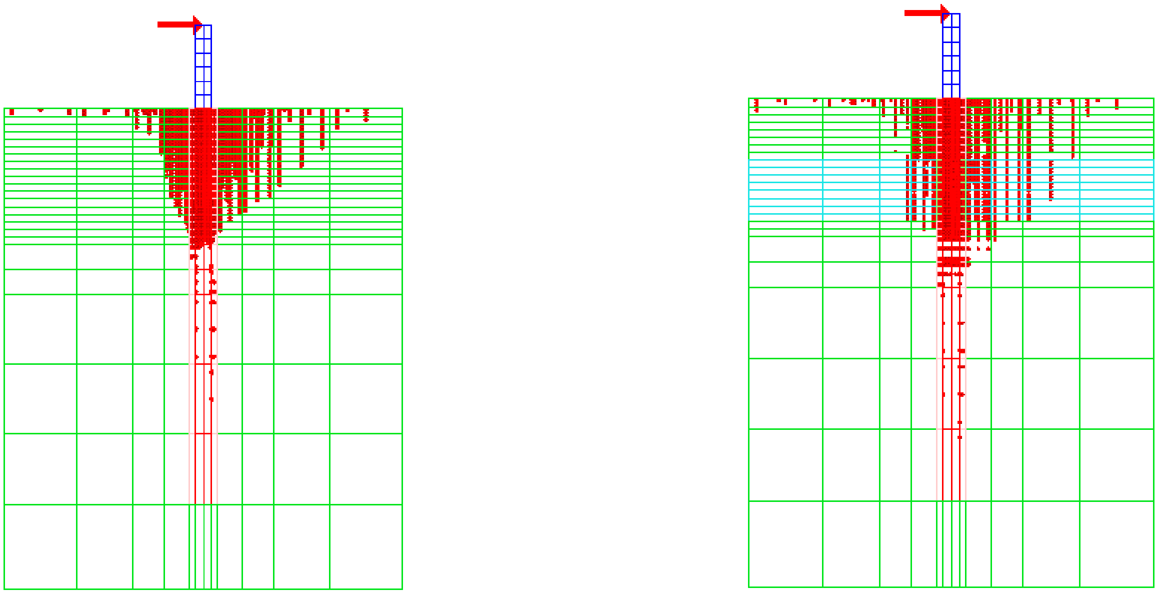
501.2.3.3 $p-y$ Curves

Results from static pushover tests on piles were used to generate $p-y$ curves. The bending moments derived by integrating vertical stresses are numerically differentiated once and twice to compute the shear force and pressure diagrams, respectively. Direct integration of shear stresses was also performed to check results and it was found that shear forces were within 5% accuracy. The combination of calculated



a) b)

Figure 501.4: The plastic zones for (a) case # 1, and (b) case # 2 at lateral loading of 400kN.



a) b)

Figure 501.5: The plastic zones of case 3 and 4 at lateral loading of 400kN.

pressures (p) and displacements obtained from the finite element solution, allowed for generation of $p-y$ curves at various depths along the pile.

In what follows, presented are generated $p-y$ curves for both uniform soils (sand and clay) as well as for layered systems. It is noted that the graphical presentation of results for bending moments, shear forces and lateral pressures (load) on a pile beam are shown with 10 lines, each one representing results for one increment $(1/10)$ of the total load.

Uniform Clay Soil. Figure 501.6 shows bending moments, shear forces and pressures along the depth of a pile in clay soil. It should be noted that the maximum bending moment, as well as the switching of sign for shear force, moves quite a bit from the depth of approximately -1.7m all the way to the depth of -3.4m. Pressure distribution shows that the top layers are already at the ultimate values of pressures and thus the pressure diagram propagates downward. There is a slight fluctuation of pressures at the depths of $4-5$m, which is attributed to the small numerical problems while doing double differentiations.

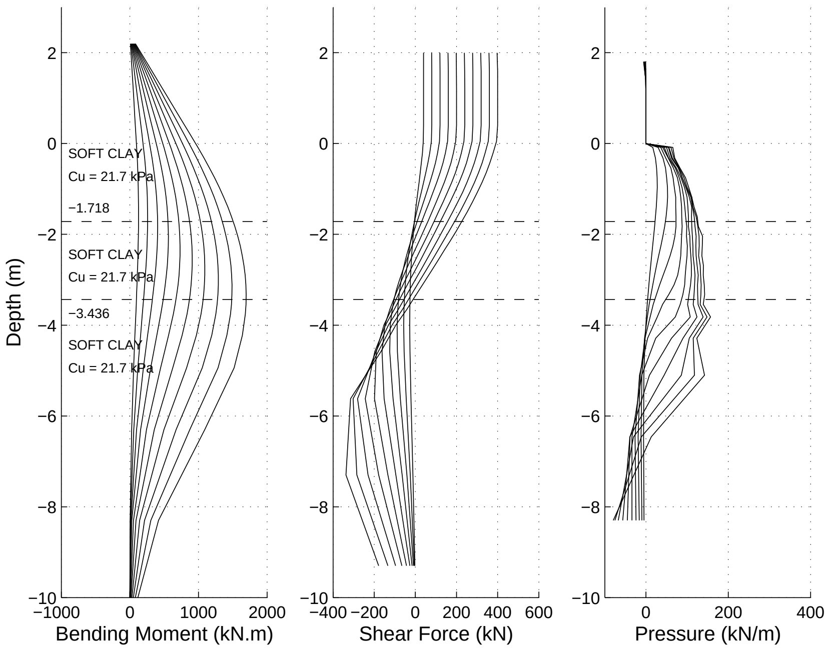


Figure 501.6: Bending moment, shear force and pressure distributions for the uniform clay profile.

Figure 501.7 shows generated $p-y$ curves for uniform clay layer. It is obvious that most of the clay (at least until the depth of -2.6m) has reached its peak resistance.

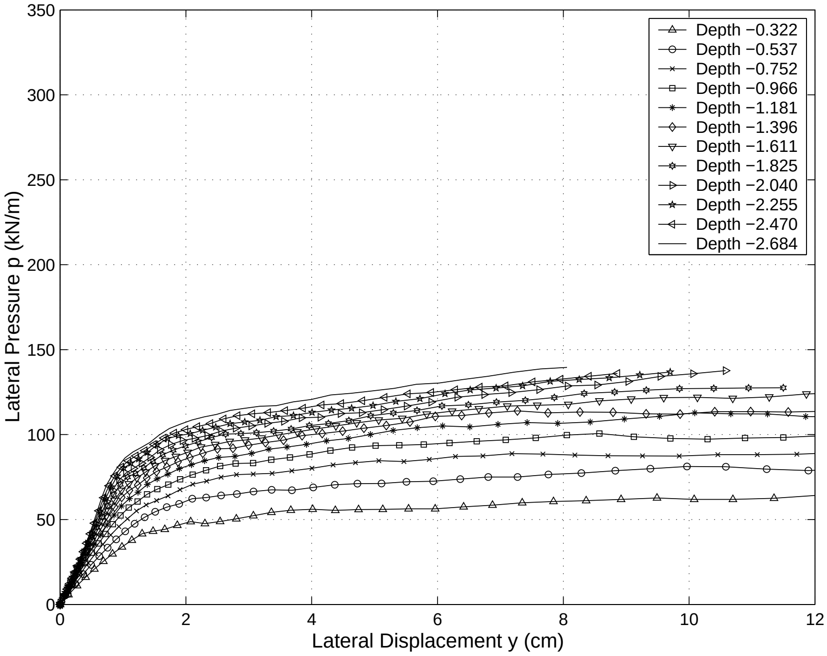


Figure 501.7: Calculated $p-y$ curves for the uniform clay profile.

Uniform Sand Soil. Figure 501.8 shows bending moments, shear forces and pressures for a pile in a uniform sand soil. In this case it is interesting to note that the maximum bending moment, as well as the change of sign for the shear force is moving only between the depths -1.8m and -2.0m. Moreover, the pressure diagram shows steady increase (with top layers reaching ultimate pressures) until the depth of -1.7m and then steadily decreases, and changes sign at greater depths (below -4.0m).

Figure 501.9 shows generated $p-y$ curves for the uniform sand case. It is interesting to note that only the top layer at the depth of about -0.3m will reach the ultimate pressure. All the other sand material is far away from corresponding ultimate pressures. It is also worth noting that the displacements in the case of uniform sand are much smaller (almost twice as small) than what has been observed in uniform clay case.

Clay Soil with a Layer of Sand. Figure 501.10 shows bending moments, shear forces and pressures for a layered soil case. In this case a layer of sand extends from $-1.72m$ to $-3.44m$. The rest of soil is soft clay. It is interesting to note a large jump in pressures for the sand layer (as expected) and that the pressures in the top clay layer (from the surface to $-1.7m$) reaches ultimate values. Small non–uniform distribution of the pressures at the interface of sand and clay at $-3.44m$ is attributed to the coarseness of the finite element mesh. In comparing Figure 501.10 with the results for uniform clay case (Figure 501.6) it is obvious that the sand layer arrests the propagation of deformation and forces in depth and

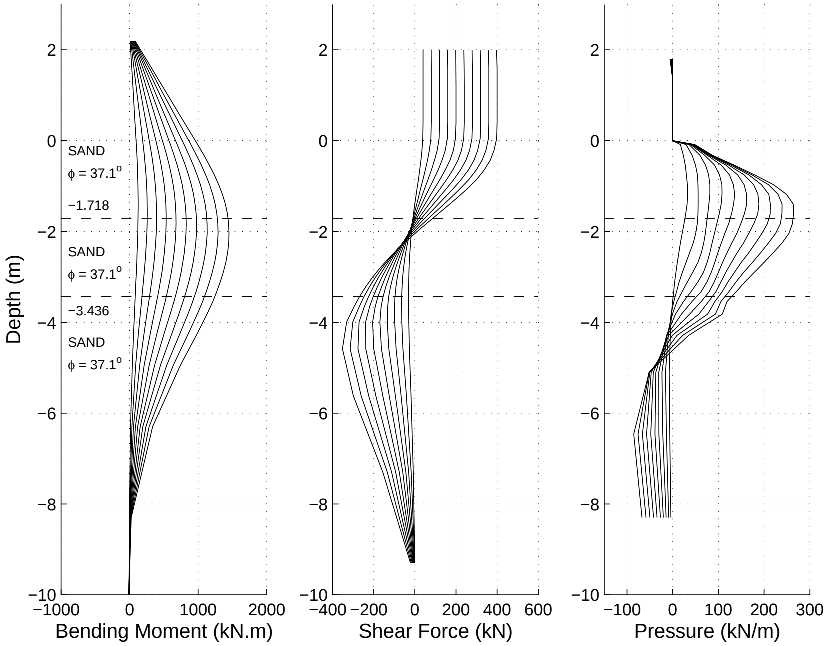


Figure 501.8: Bending moment, shear force and pressure distributions for the uniform sand profile.

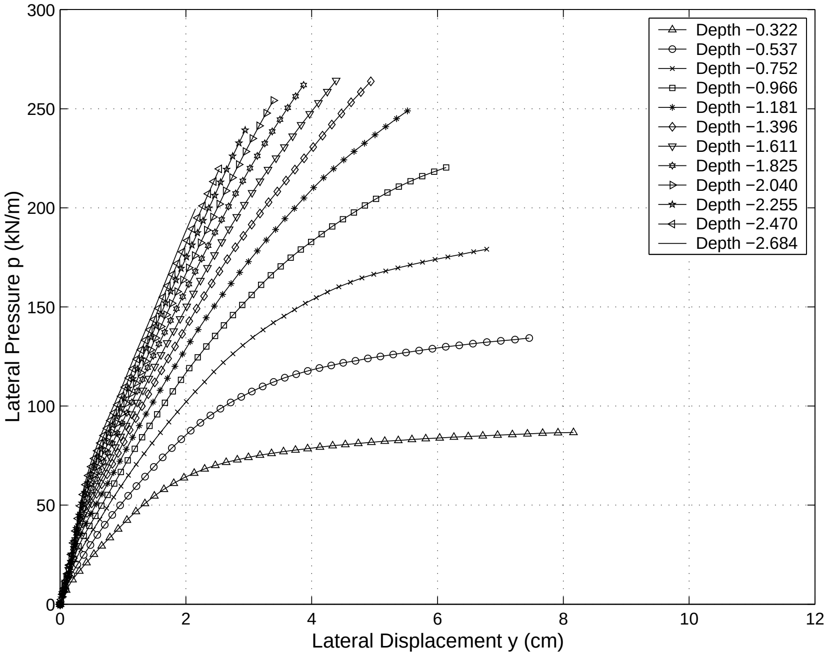


Figure 501.9: Calculated $p-y$ curves for the uniform sand profile.

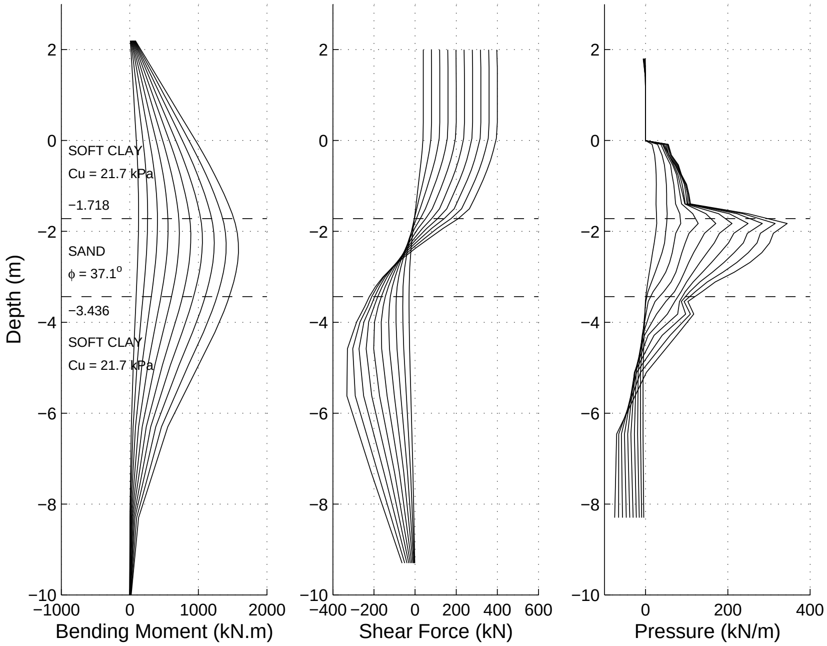


Figure 501.10: Bending moment, shear force and pressure distributions for the clay soil with a sand layer.

fixes the maximum moment to approx. $-2.1m$.

Figure 501.11 shows generated $p-y$ curves for the layered case (single layer of sand in clay). The $p-y$ curves were generated only for the top layer of clay and middle layer of sand, to the depth of $-2.7m$. It is interesting to note that the $p-y$ curve for clay at the depth of $-1.61m$ (close to the sand layer) exhibits strong hardening, unlike similar curve for the uniform clay soil, in Figure 501.7. The increase in pressure (transversal loading on the pile) between uniform clay (Fig. 501.7) and clay underlain by a medium dense sand layer (Fig. 501.11) at the displacement of $0.06m$ is more than two times.

Sand Soil with a Layer of Clay. Figure 501.12 shows bending moments, shear forces and transversal pressures for a case where a layer of soft clay is present within sand soil. Unlike the case of uniform sand soil (Figure 501.8) the presence of soft clay layer will change the depth of maximum moment by almost $1m$ (from $-2.0m$ to $-3.0m$). In addition to that, the distribution of pressures on a pile is changed significantly, as seen in the right plot of Figure 501.12. The reduction of pressures will extend into the sand layer and present significant influence of soft clay on pressures in sand.

Figure 501.13 shows generated $p-y$ curves for the case of sand with a soft clay layer. It is noted that the $p-y$ curves for sand that is some distance away from the interface with clay are much the same as for the uniform sand case (refer to Fig. 501.9 and Fig. 501.19(a)). However, the $p-y$ curves in sand

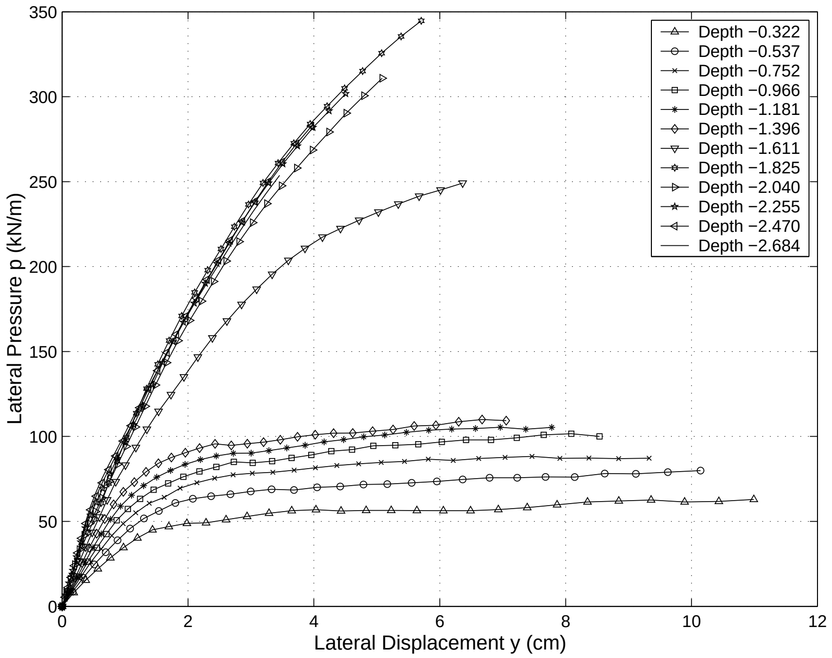


Figure 501.11: Calculated $p-y$ curves for the clay soil underlain by a medium dense sand layer.

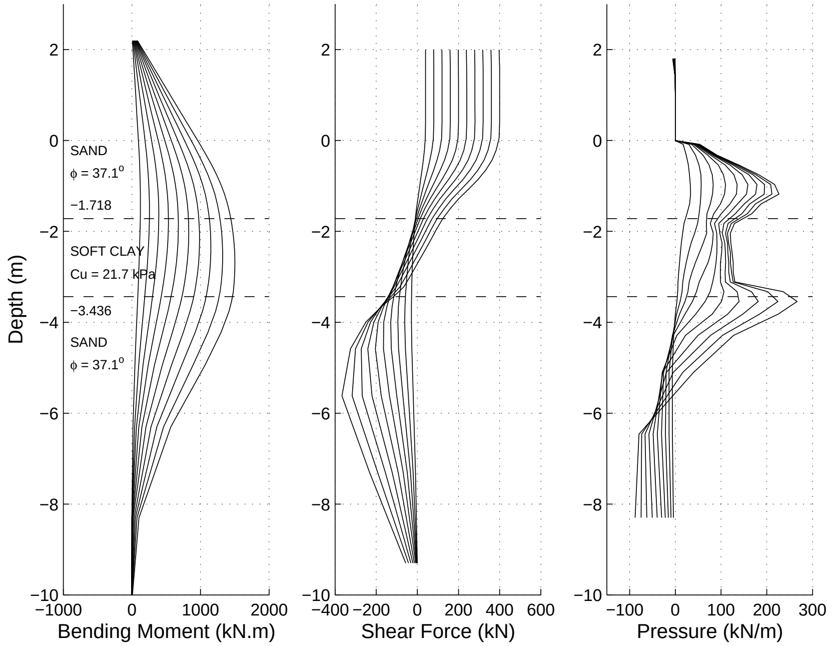


Figure 501.12: Bending moment, shear force and pressure distributions for the sand soil with a soft clay layer.

close to the interface are changed in some cases significantly. For example, the $p - y$ curve at depth of $-1.61m$ is showing pressure of approx. $p = 265kN/m$ at the displacement of $0.042m$ for the uniform sand case, while the same $p - y$ curve, still in sand, has a drop in pressure at the same displacement to $p = 140kN/m$. Similar trend is observed for other $p - y$ curves close to the interface of sand with clay.

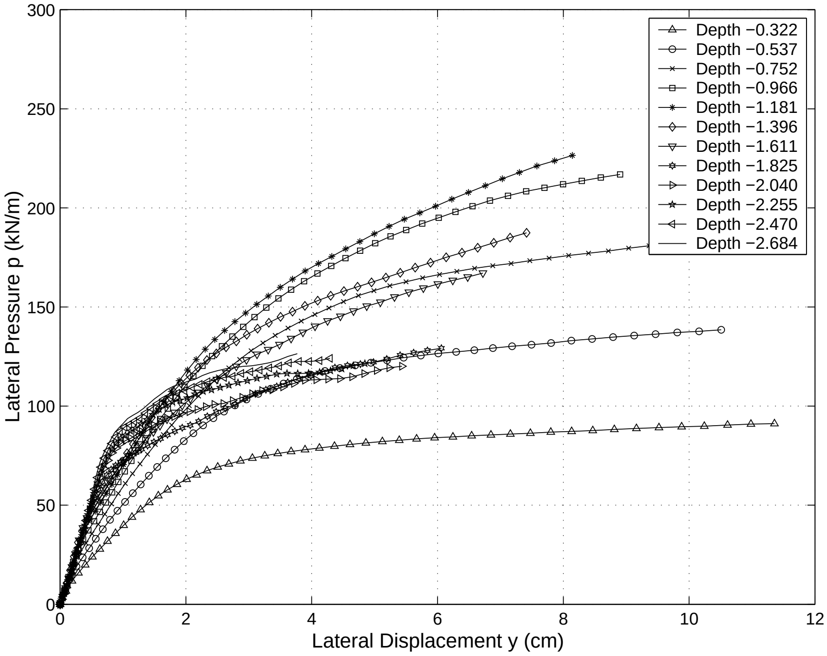


Figure 501.13: Calculated $p - y$ curves for the sand soil underlain by a soft clay layer.

501.2.3.4 Comparisons of Pile Behavior in Uniform and Layered Soils

Comparison of pile behavior in uniform and layered soils can also be performed by looking at the displacement and bending moment distributions. For example, Figure 501.14 compares the distributions of displacements for the uniform sand case with the sand and clay layer case. First observation is that the uniform sand layer allows smaller displacements of the pile head ($0.12m$) while the inclusion of clay layer raises those displacements to $0.22m$. Second observation is that the point of rotation for the pile (point which does not move as the loading is applied) is pushed deeper, from $5m$ to approximately $6m$. Moreover, the propagation of displacements along the depth of a pile is much greater for a layered case, the surface displacement is extended from $0.09m$ to almost $0.13m$.

Figure 501.15 shows similar results for uniform clay and clay with a layer of sand case. In this case, the inclusion of a sand layer will increase the stiffness of the pile (as expected) and will also reduce propagation of displacements with depth.

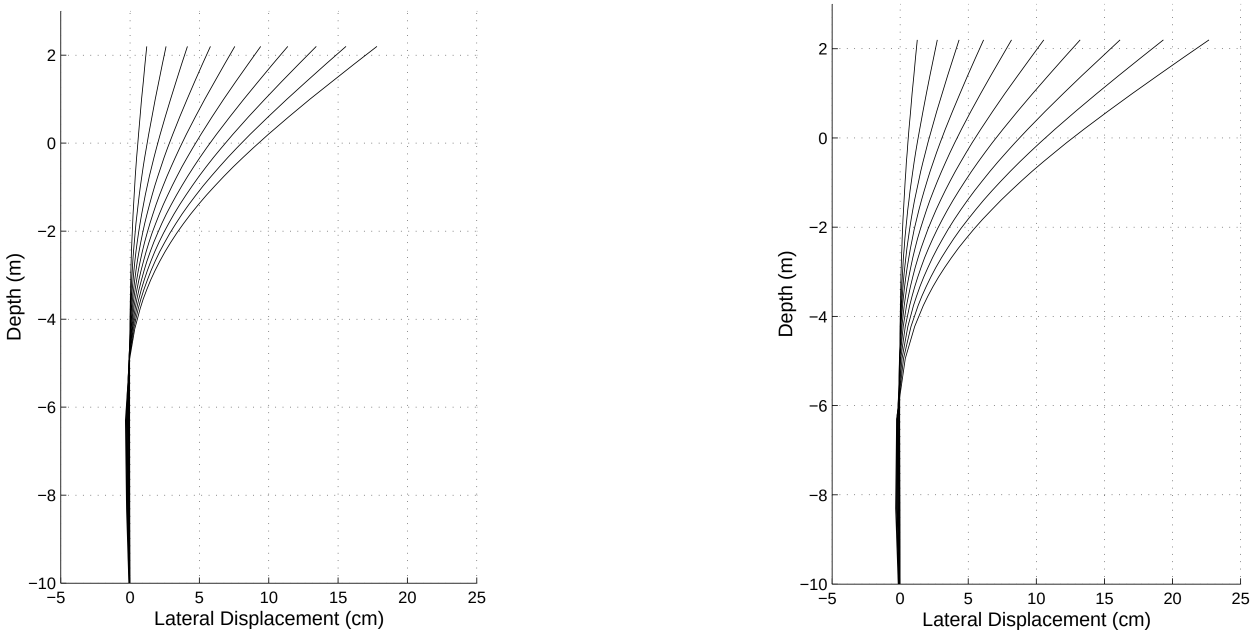


Figure 501.14: Pile displacement distributions along the depth in a uniform sand profile (left) and sand with clay layer profile (right).

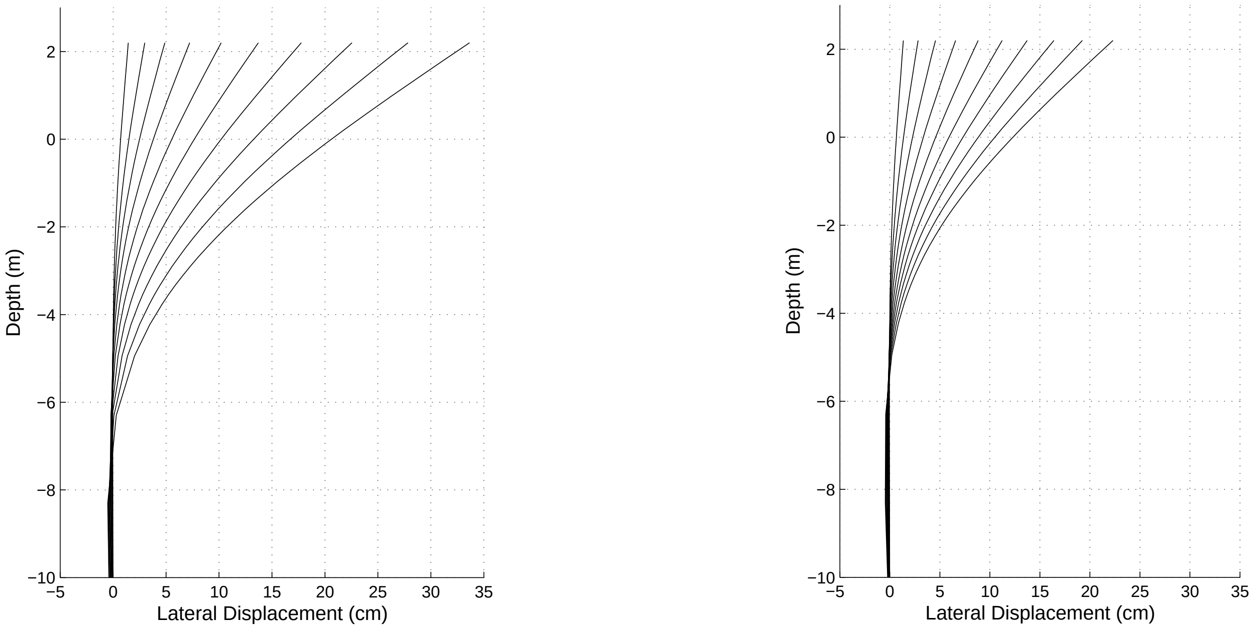


Figure 501.15: Pile displacement distributions along the depth in a uniform clay profile (left) and clay with sand layer profile (right).

Figure 501.16 shows comparison of pile head displacements for all four cases. It is noted that the two layered cases exhibit similar behavior in terms of displacements, both at the pile head and in terms of displacement profiles (compare right plot in Fig. 501.14 and left plot in Fig. 501.15).

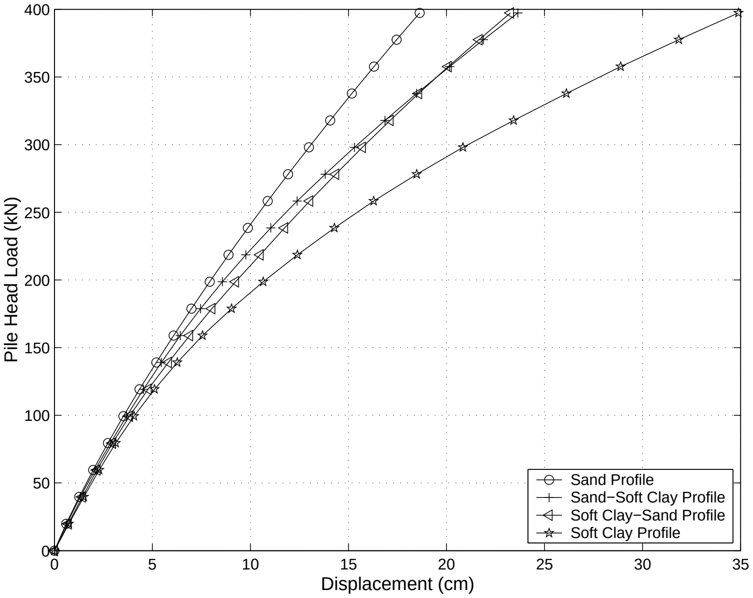


Figure 501.16: Pile head displacement comparison for all the four cases.

Figure 501.17 shows comparison of the maximum bending moment calculated for the pile for all four cases. It is interesting to note that the difference between the two uniform soil cases (uniform sand and uniform clay) is not that pronounced. Of course one has to remember that the material for pile was assumed to be linear elastic, no yielding was allowed for the aluminum pile.

The $p-y$ curves for uniform clay and clay with a layer of sand were plotted together in Figure 501.18 (a) for comparison. It can be seen that all the $p-y$ curves in clay except the one right next to the layer interface are almost identical. In order to measure the magnitude of the effects of sand layer on the pressure of soft clay layer, the ratio of pressures in clay layer for clay soils with a sand layer and uniform clay soils lateral displacement of 12%D, i.e. 5.15 cm, were computed and plotted against the distance in terms of times of pile width D in Figure 501.18. It is noted that the disturbance to the pressure field is much more confined to the immediate vicinity (within 0.75D) of the layer interface. In addition, the results from two more analysis of the same model with different sands (friction angles $\phi^{'} = 25^{o}, 30^{o}$ respectively, other parameters remain the same.) were included in Figure 501.18. It is shown that the lateral pressure ratio is affected considerably when sand friction angle increases from 25^{o} to 37^{o} (from 1.5 times to 2.2 times more pressure).

The $p-y$ curves for uniform sand and sand with a layer of soft clay were also plotted together for

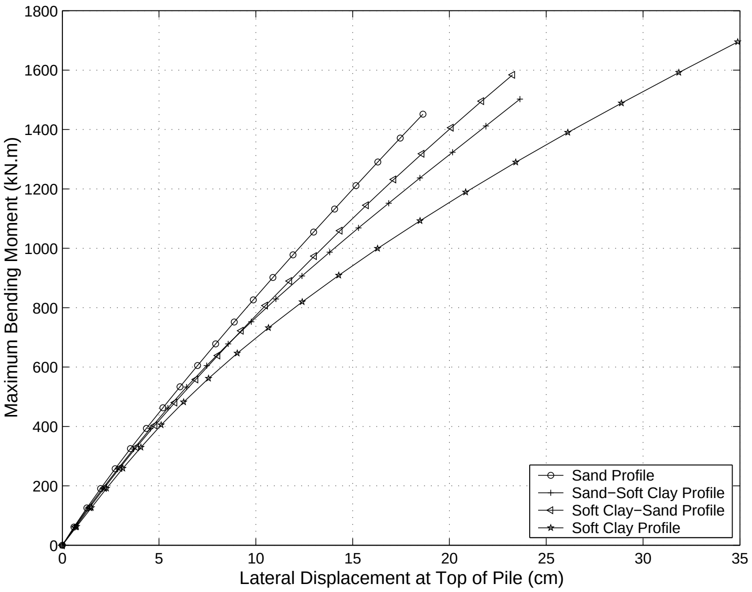


Figure 501.17: Maximum bending moment comparison for all the four cases.

comparison purposes. It was found that the effect of soft clay on the pressures in sand propagates far away from the layer interface. Therefore, three cases of an additional model with a thicker sand layer (2.4m in thickness) underlain by a soft clay layer were analyzed by varying the undrained shear strength ($C_u = 13.0\ kPa,\ 21.7\ kPa,\ 30.3\ kPa$) of the soft clay layer. Similarly, the pressure ratios at 6.5% D, i.e. 2.8 cm, were plotted in Figure 501.19. It is noted that the effects extends to as far as 4.75D from the layer interface and the reduction of pressures adjacent to the interface is about 0.6 in all three cases.

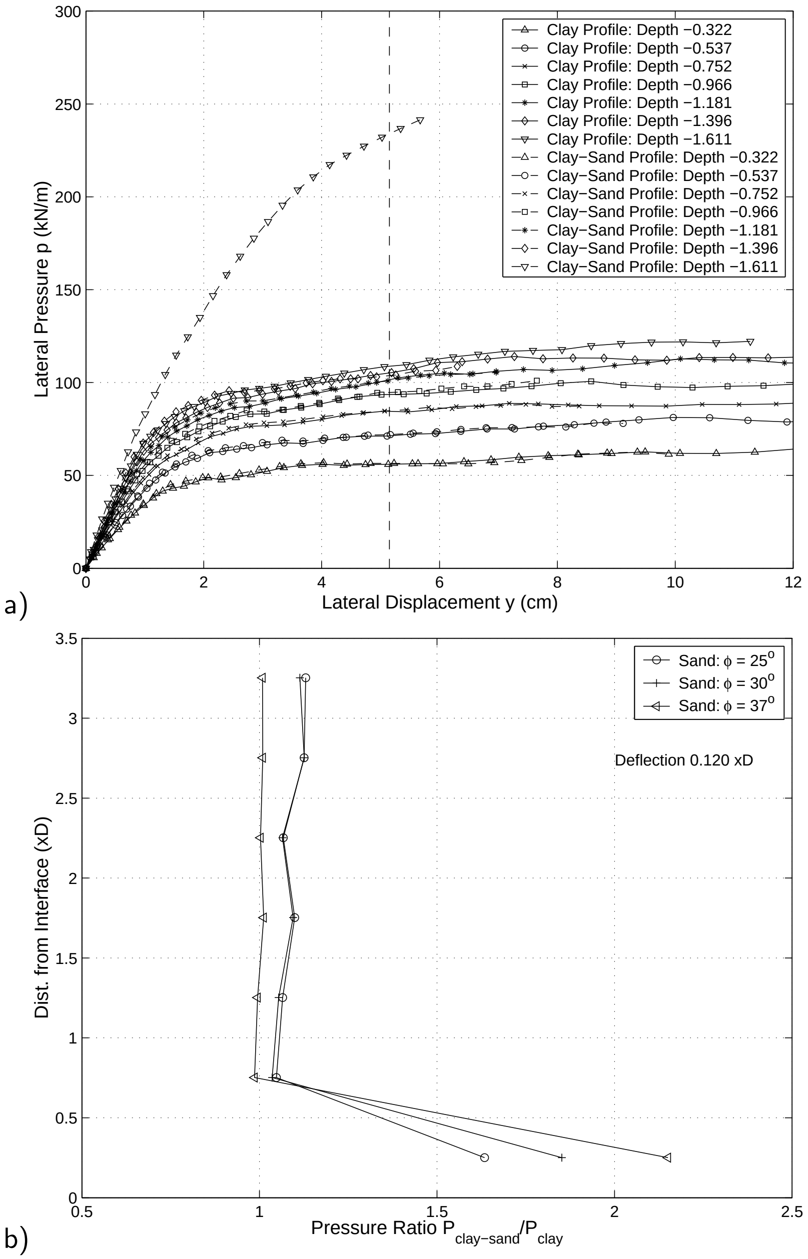


Figure 501.18: (a) Comparison of p–y curves for uniform clay versus clay with a layer of sand ($\phi' = 37^o$). (b) Pressure ratio distributions in clay layer for sands with different friction angle ($\phi' = 25^o,\ 30^o,\ 37^o$).

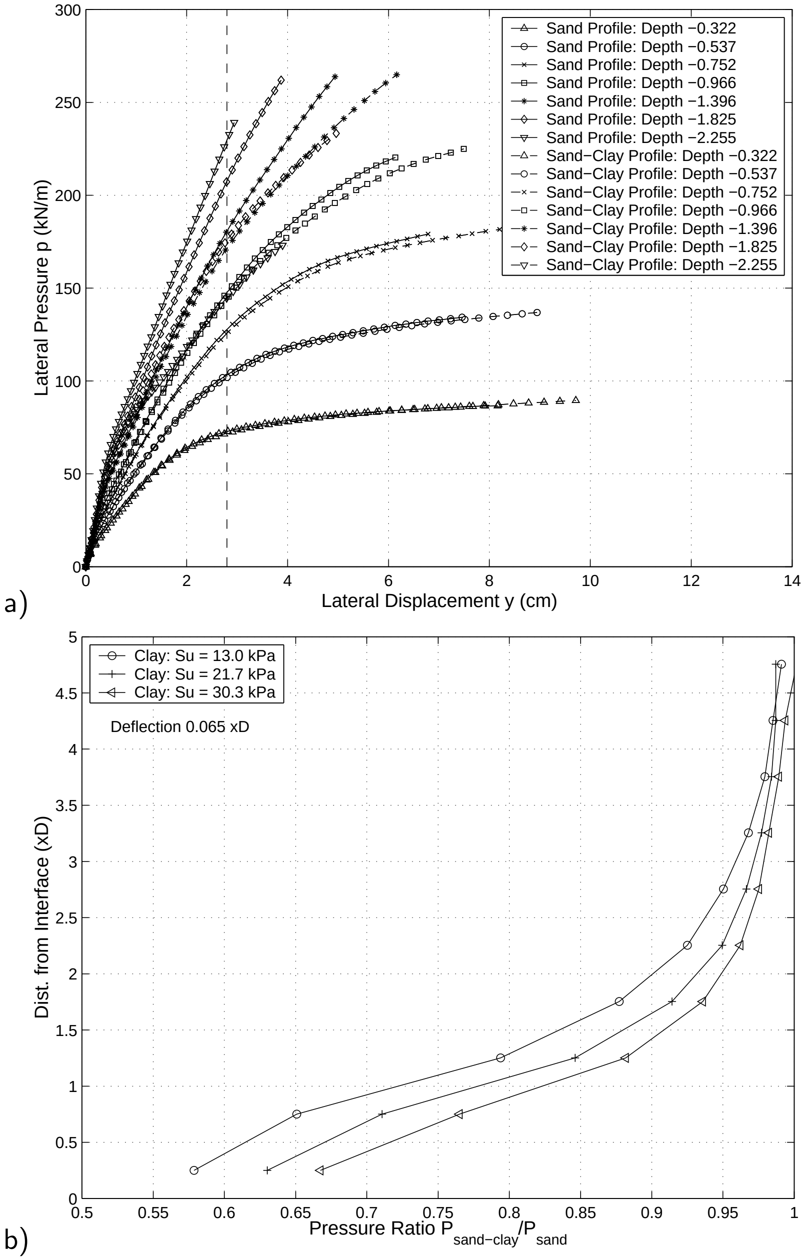


Figure 501.19: (a) Comparison of $p-y$ curves for uniform sand versus sand with a layer of soft clay ($C_u = 21.7\ kPa$). (b) Pressure ratio distributions in sand layer for clays with different undrained shear strength ($C_u = 13.0\ kPa,\ 21.7\ kPa,\ 30.3\ kPa$).

501.2.3.5 Comparison to Centrifuge Tests and LPile Results

The pile head displacements for uniform sand profile from 3D FEM, LPILE (Reese et al. Reese et al. (2000a,b)), and centrifuge test (McVay et al. McVay et al. (1998)) were plotted against pile head load in Figure 501.20. It can be seen that they agree with each other fairly well. It should be noted that the material properties for our 3D finite element simulations were not in any particular way calibrated to improve the results. They were simply used as presented in the centrifuge study by McVay et al. McVay et al. (1998) and numerical simulation by Zhang et al. Zhang et al. (1999). Whereas, the results from LPILE were back-fitted since the coefficient of subgrade reaction η_h was back-calculated as 2714 kN/m^3 (Zhang et al. Zhang et al. (1999)).

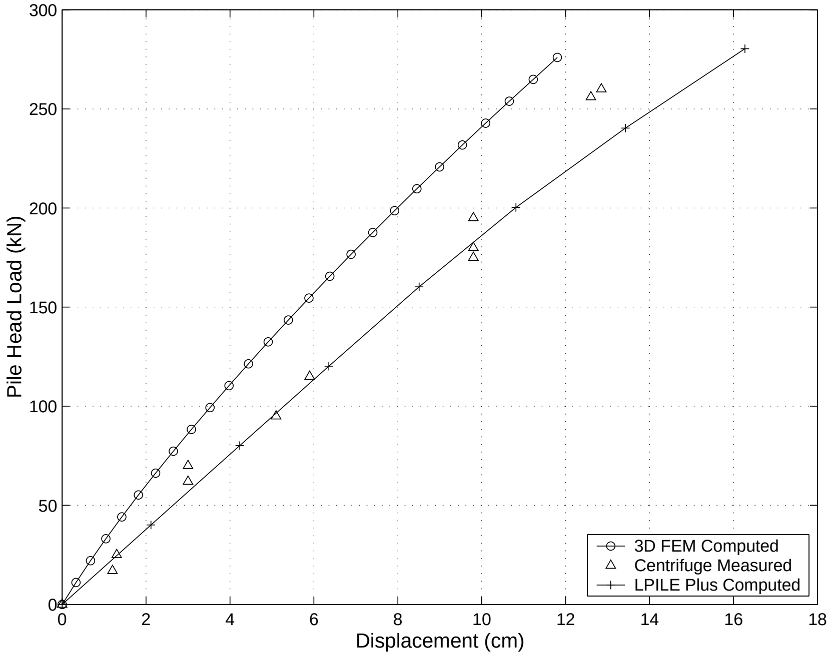


Figure 501.20: Simulated versus experimental pile head displacements.

The bending moments, shear forces and lateral pressures of uniform sand and clay profiles from 3D FEM and LPILE were plotted against pile depth at several pile head loads in Figure 501.21 and 501.22. In general, there is a good agreement between the results from FEM and LPILE in uniform sand profile. In uniform soft clay profile, it is noted that the pressures at shallow depth from LPILE are smaller than those computed by FEM, which agrees with one of the findings by the work of Steven and Audibert Stevens and Audibert (1979). For example, the pressures at lateral load of 120kN and 200kN from LPILE are only about half of those from FEM. Because the pressures at shallow depths are so small in LPILE that the pile head has to deform much more than in FEM and the passive pressure zone in LPILE extends to fairly large depth.

Since LPILE currently uses the equivalent depth method developed by Geogiadis Georgiadis (1983) for layered soil profiles, the LPILE output pressure distribution along pile depth, especially across the layer interface does not take into account of the layering effect, thus it is not that meaningful to compare pressure distributions of layered profiles from LPILE versus FEM.

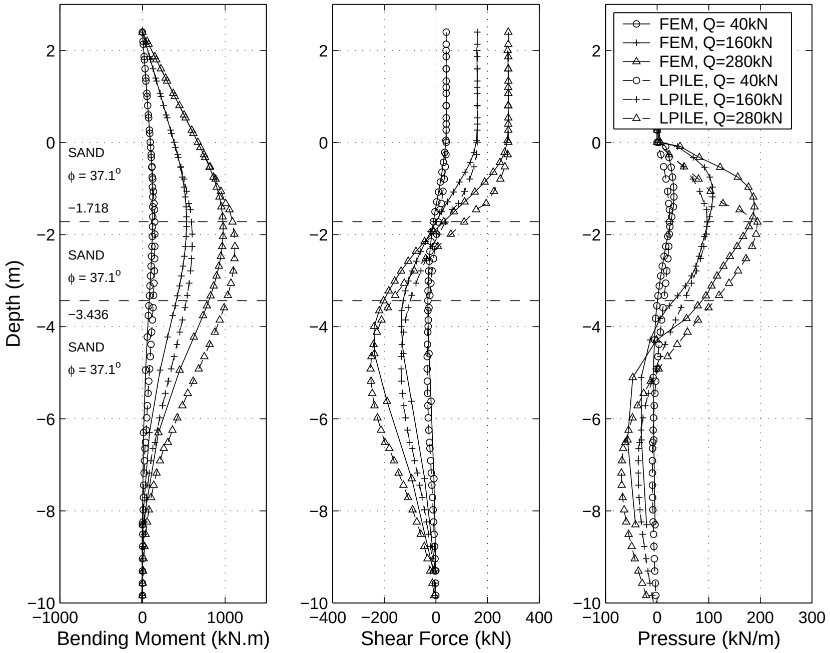


Figure 501.21: Comparison of bending moment, shear force and pressure computed by FEM and LPILE in uniform sand profile(case #3).

It is also interesting to compare the $p-y$ curves derived from FEM with those used in LPILE. Figures 501.24 and 501.23 show FEM derived and LPILE used $p-y$ curves for uniform clay and sand profiles, respectively. It should be noted that the coefficient of subgrade reaction η_h was again back-calculated as 8969 kN/m^3 in order to get a reasonable $p-y$ curves. From Figures 501.24 (a) and (b), it is clear that $p-y$ curves in sand profile from LPILE have lower resistance at depth close to ground surface. The $p-y$ curves for clay profile shown in Figures 501.24 (a) and (b) are seen to have much lower resistance at shallow depths.

501.2.4 Summary

This paper presents results from a finite element study on the behavior of a single pile in elastic–plastic soils. The analysis included single pile behavior in sand, clay and layered soils. Based on the results

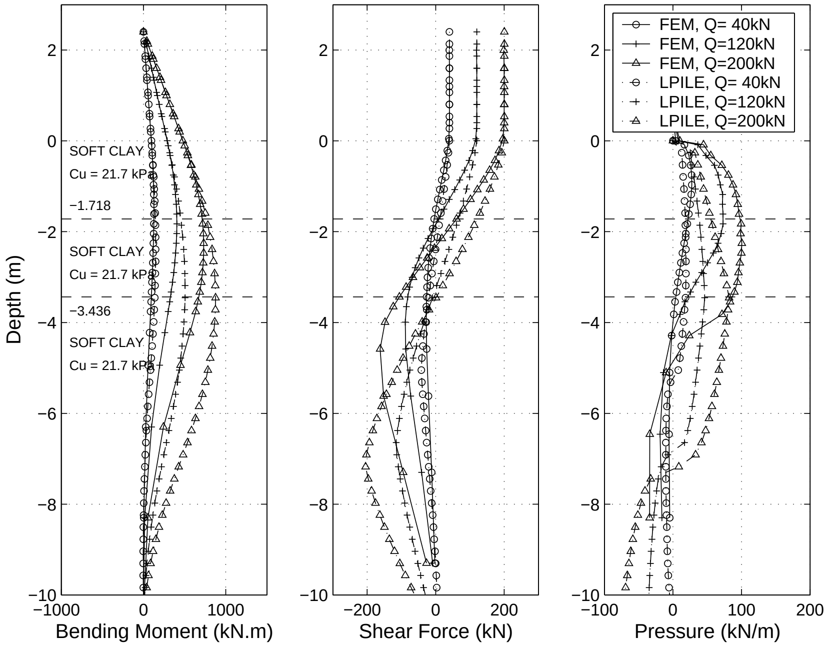


Figure 501.22: Comparison of bending moment, shear force and pressure computed by FEM and LPILE in uniform soft clay profile(case #1).

presented, it is concluded that three dimensional finite element analysis using very simple elastic-plastic soil models can predict the pile head deflection with very good accuracy.

The main findings of this numerical study can be summarized as follows:

- When a sand layer is present within a clay deposit, the increase in lateral pressure in clay near the interface is confined to a narrow zone, up to two times of pile width, therefore the layering effect in this case is not prominent.
- When a clay layer is present within a sand deposit, the reduction in pressures spread well into the sand layer (up to four times of pile width). The layering effects are of more importance in this case since the disturbance zone is large and the pressure reduction is significant. Reduction factors are given in terms of charts of pressure reduction versus the distance from the interface.

In addition, comparison with centrifuge data shows generally a good agreement between the bending moments, shear forces and lateral resistance. Moreover, a comparison with results from program LPILE, used in extensively in practice, show some discrepancies ultimate pressures in shallow soil layers.

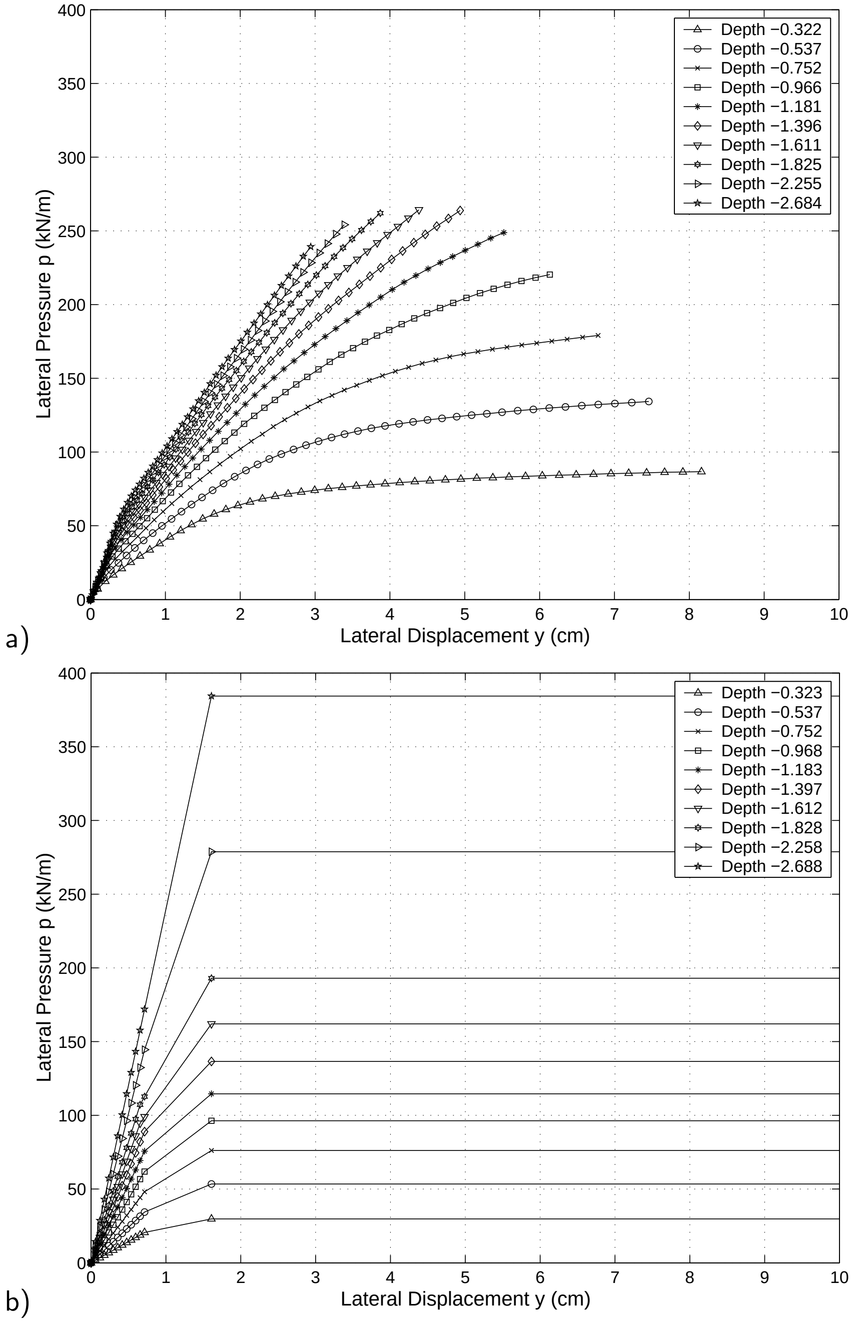


Figure 501.23: $p-y$ curves from FEM (a) and LPILE (b) in uniform sand profile ($\eta_h = 8969 \ kN/m^3, \ \phi = 37.1^o$).

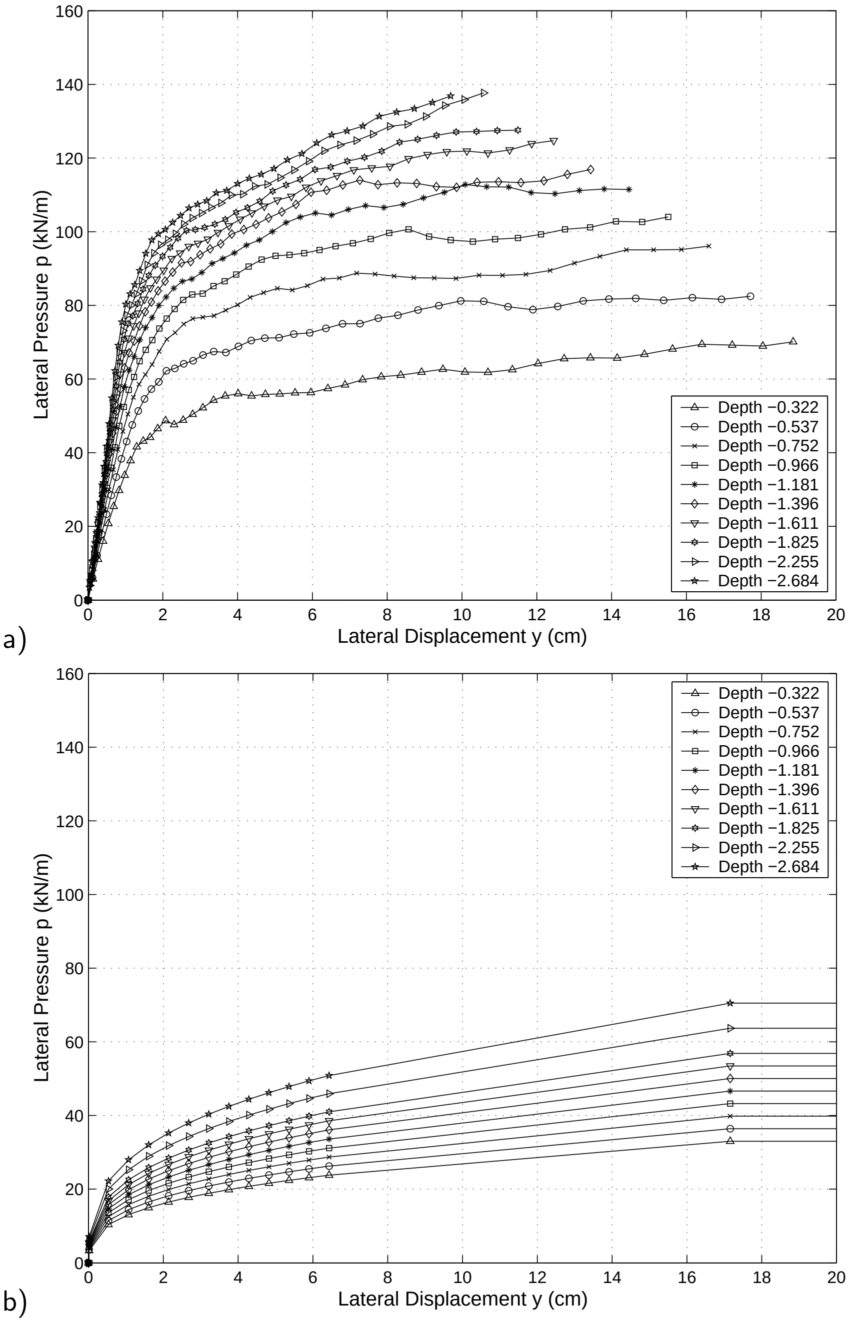


Figure 501.24: $p-y$ curves from FEM (a) and LPILE (b) in uniform clay profile ($\epsilon_{50} = 0.02$, $C_u = 21.6\ kPa$).

501.3 Study of soil layering effects on lateral loading behavior of piles

Material presented here has been previously published in our paper Yang and Jeremić (2005).

501.3.1 Introduction

The theory of beams on a Winkler-type subgrade (Hartog (1952)), also known as the p–y approach, has been widely used to design piles subjected to lateral loading. Based on that theory, the method models the lateral soil–foundation interaction with empirically derived nonlinear springs (p–y curves). The advancement of computer technology has made it possible to study this problem using more rigorous elastic–plastic Finite Element Method (FEM).

Here mentioned are a few representative examples of finite element studies of pile foundations. Muqtadir and Desai (1986) studied the behavior of a pile–group using a three dimensional (3D) program with nonlinear elastic soil model. An axisymmetric model with elastic-perfectly plastic soil was used by Pressley and Poulos (1986) to study group effects. Brown and Shie (1990a), Brown and Shie (1990b), Brown and Shie (1991), and Trochanis et al. (1991) conducted a series of 3D FEM studies on the behavior of a single pile and a pile group with elastic-plastic soil model. These researchers used interface elements to account for pile–soil separation and slippage. Moreover, Brown and Shie derived p–y curves from FEM data, which provide some comparison of the FEM results with the empirical design procedures in use. Kimura et al. (1995) conducted 3D FEM analysis of the ultimate behavior of laterally loaded pile groups in layered soil profiles with the soil modeled by Drucker–Prager model and pile modeled by nonlinear beam elements. A number of model tests of free– or fixed–headed pile groups under lateral loading in homogeneous soil profiles have been simulated by Wakai et al. (1999) using 3D elasto-plastic FEM. Pan et al. (2002) studied the performance of single piles embedded in soft clay under lateral soil movements. A good correlation between the experiments and the analysis has been observed in these studies. All these results demonstrated that FEM can capture the essential aspects of the nonlinear problem.

Information about the lateral behavior of piles in layered soil profiles is very limited. Some analytical studies have been conducted by Davisson and Gill (1963) and Lee and Karunaratne (1987) to define the influence of pile length, the thickness of upper layer and the ratio of stiffness ratio of adjacent layers on the pile response based on the assumption that the soil is elastic. Reese et al. (1981) conducted small scale laboratory tests on a 25 mm diameter pile and a field test with 152 mm diameter pile in layered soils and found that there was a relatively good agreement between deflections measured in the tests and deflections computed using homogeneous p–y curves at small loads. Georgiadis (1983) proposed an approach which is currently used in the `LPILE` program (Reese et al. (2000a,b)). This method assumes the p–y curves of the first layer are the same as those for homogeneous soils. The effects of upper layers on the p–y curves of the lower layers are accounted for by the equivalent depth of the overlying layers based on strength parameters.

To the Authors' knowledge, there is no literature reporting on FEM study of layering effects on the behavior of laterally loaded piles in layered profiles. However, it is of great interest to investigate the layering effects since in practice, most of soil deposits are layered systems. In a predominantly clay site with a minor sand layer, the sand layer will still be counted on to provide most of the soil resistance. In this case, the layering effects (probably reduction of resistance in the sand layer) must be considered. Current practice is to *"make an educated guess to reduce the sand p–y curves to account for the soil layering effects"* (Lam and Law (1996)). Obviously, an educated guess might not result in optimal design. It is very important to find out how these layers in the layered system affect each other in order to carry out a more accurate analysis of pile foundation and therefore provide a more effective way for the design of pile foundations in layered soil systems.

This paper describes four 3D finite element models of a laterally loaded pile embedded in uniform and layered soil profiles, with the dimensions and soil parameters similar to those used in the centrifuge studies by McVay et al. (1998) and Zhang et al. (1999). Visualization tool Joey3D (Yang (2002)) was used to compute the bending moment, shear force and lateral resistance diagrams along the pile. Model calibration, comparison of finite-element analysis results with those from centrifuge tests and the `LPILE` program, and comparison of finite-element generated p–y curves with traditional p–y curves are summarized in a separate paper (Yang and Jeremić (2003). In this paper, p–y curves from each model were cross compared to illustrate both the effects of an intermediate soft clay (or sand) layer on the p–y curves of the sand (or soft clay) layers and the effects of sand (or soft clay) layers on the intermediate soft clay (or sand) layer. In addition, a limited parametric study was conducted to further investigate the layering effects in terms of lateral resistance ratios. The OpenSees OpenSees Development Team (Open Source Project) (2000-2006) finite element framework was employed for all the computations. Soil modeling was performed using the Template Elasto–Plastic Framework (Jeremić and Yang (2002)) and solid elements while the piles were modeled using linear elastic solid elements, all developed by the Authors.

501.3.2 Finite Element Pile Models

Single pile finite element models with the dimensions similar to the prototype model described in the above centrifuge tests were developed and a number of static pushover tests were simulated with 3D FEM using uniform soil and layered soil cases. The models for all cases were illustrated in Figure 501.25 (a). There are four main analysis models. Two of them are dealing with uniform sand and clay deposits,

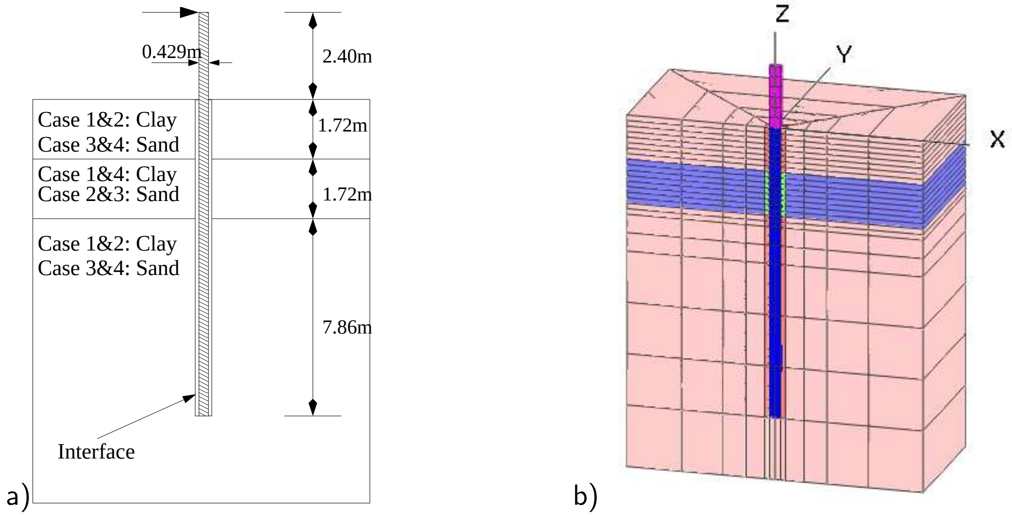


Figure 501.25: (a) Single pile models, dimensions and layers for models #1, #2, #3 and #4, including pile-soil interfaces and (b) 3D mesh of the single pile model.

while the other two are featuring layered soil deposits. In particular, model # 1 has a uniform soft clay deposit, model # 2 includes top and bottom layers of soft clay with an interlayer of medium dense sand. Model # 3 features uniform medium dense sand deposit, while model # 4 features top and bottom layers of medium dense sand with an interlayer of soft clay.

Figure 501.25 (b) shows the finite element mesh for all four models. Based on symmetry, only half of the model is meshed. Twenty–node brick elements are used to mesh the soil, pile and pile–soil interface. The square pile, with a width of 0.429 m and length of 13.7 m[1], is divided into four elastic elements (per cross subsection) with the properties of aluminum. The mesh is refined at the upper part of the model in order to provide data points for the computation of shear force and lateral resistance of sufficient reliability as well as for investigation of the layering effects. Additional finite element analysis of a cantilever beam using the same mesh as the pile was carried out and comparison of the beam displacement from FEM and beam theory solution indicated that the mesh was fine enough to capture the pile behavior. As to the boundaries, the sides and bottom of the model are fixed with the exception of the symmetric boundary, which is only supported in Y direction. Since the sides are 13 times of the pile width away from the pile center, it is believed that the fixed boundaries have very limited effects on the results. In addition to that the model size is closely following that of the physical, centrifuge model, which resided in a container of similar size. The pile–soil interface is represented by one thin layer of elements. The purpose of this layer is to mimic the installation effects on the pile (drilled or driven). It

---

[1] All dimensions are from the centrifuge study, prototype scale.

also serves a purpose of a simplified interface which allows for tension cut-off (gapping) and controlled, coupling of horizontal and vertical resistance according to Coulomb frictional laws.

501.3.3 Constitutive Models

Two simple models were used in this numerical study. Specifically, clay was modeled by von Mises material model which is completely defined with the undrained shear strength. Sand was simulated by Drucker–Prager material model with nonassociated flow rule, defined with the friction and dilation angles. The reason for using such simple models is that the experimental results used in comparison with simulations did specify only very limited number of material properties for sands. Furthermore, a small number of model parameters needed by simple models are convenient for parametric study. In both material models, the Young's moduli vary with confining pressure, as shown in Eqn. (501.2) (cf. Janbu (1963), Duncan and Chang (1970)):

$$E = E_o \left(\frac{p}{p_a}\right)^a \tag{501.2}$$

where E_o is Young's Modulus at atmospheric pressure, p is the effective mean normal stresses, p_a is the atmospheric pressure, and a is constant for a given void ratio. In this work, 0.5 was used.

The following parameters were used for medium dense sand: friction angle $\phi = 37.1^o$, Shear modulus G at a depth of 13.7 m = 8960 kPa (E_o = 17400 kPa), Poisson's ratio $\nu = 0.35$ and unit weight $\gamma = 14.50$ kN/m^3. These parameters were given by Zhang et al. (1999). A dilation angle of $\psi = 0^o$ is used in this work (Brown and Shie (1990a)). The undrained shear strength, Young's modulus, Poisson's ratio and unit weight of clay were chosen to be $C_u = 21.7$ kPa, $E_0 = 11000$ kPa, $\nu = 0.45$, $\gamma = 11.8$ kN/m^3, respectively. The interface elements were simulated by Drucker–Prager model with a friction angle $\phi = 25^o$, and a dilation angle $\psi = 0^o$. All material properties were summarized in Table 501.1.

Table 501.1: Material properties of sand, clay, pile and soil–pile interface used in FEM analysis.

Soil	E_o (kPa)	ν	γ (kN/m^3)	ϕ $(^o)$	ψ $(^o)$	C_u (kPa)
Medium dense sand	17400	0.35	14.5	37.1	0	–
Loose sand	16000	0.35	14.1	34.5	0	–
Clay	11000	0.47	11.8	–	–	21.7
Pile	69000000	0.33	26.8	–	–	–
Soil–pile interface	Variable	Variable	Variable	25	0	–

501.3.4 Comparison of p–y Behavior in Uniform and Layered Soil Deposits

This subsection presents representative results related to the behavior of piles in uniform and layered soil deposits. Specifically the p–y response curves derived for 3D FEM results for homogeneous and layered soil deposits are compared with each other to investigate the layering effects.

501.3.4.1 Uniform Clay Deposit and Clay Deposit with an Interlayer of Sand.

The p–y curves of uniform clay deposit and clay deposit with a layer of sand were compared in Figure 501.26. It is clearly seen that the p–y curve ($Z = -3.75D$) close to the interface ($Z = -4D$) is significantly different from that in uniform soil profile.

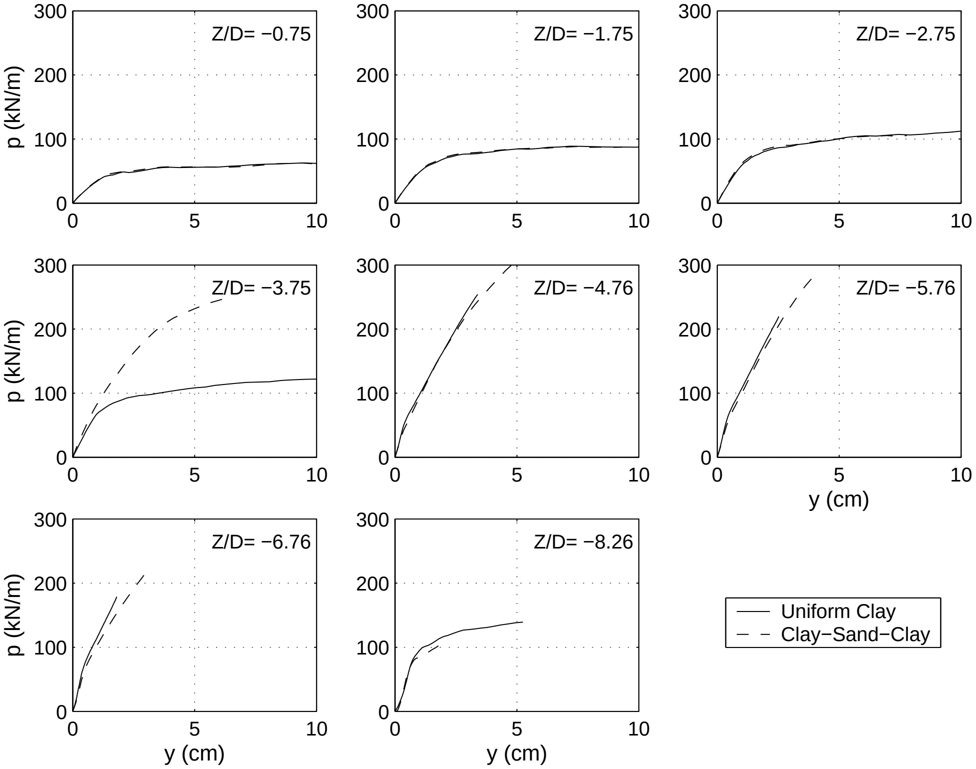


Figure 501.26: Comparison of p–y curves of uniform clay deposit versus clay deposit with an interlayer of sand (Sand: $\phi = 37^o$; Clay: $C_u = 21.7$ kPa).

In order to measure the magnitude of the effects of the intermediate sand layer on the lateral resistance of the soft clay layers and vice versa, the ratios of soil lateral resistances in the layered (p) and uniform models ($p_{homog.\ model}$) at several lateral displacements (i.e. 0.5%, 1.0%, 2.0%, 2.5%, 8.0% and 10.0% of pile width D) were computed and plotted against vertical coordinate (Z) normalized by pile width D in Figures 501.27 and 501.28[2]. In addition, the results from two more analyses of the same model with different sands (friction angle ϕ were varied from 25^o to 30^o, while originally, the friction angle was set to 37^o) were also included in these figures.

From Figure 501.27, it is observed that the lateral resistance ratios are independent of friction angle ϕ of sand at small lateral displacements ranging from 0.5% to 1.0% of pile width D. When the lateral displacement is greater than 1.0%, the variation in ϕ starts to affect the lateral resistance ratio, as shown in Figure 501.28.

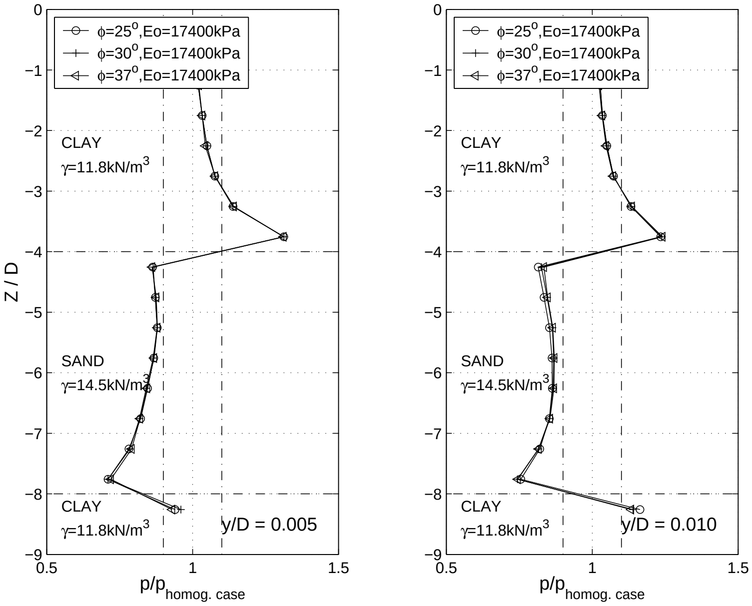


Figure 501.27: Lateral resistance ratio distributions (Clay:$C_u = 21.7\ kPa, E_o = 11000\ kPa$) for sands with various ϕ at lateral displacements of 0.5% and 1.0% pile width.

Overall, the effect of the sand layer reduces to less than 10% at about one pile width above the upper sand interface and the lateral resistance ratio at a quarter pile width above the upper sand interface is 1.3 at lateral deflection of 0.5% pile width. It may be noted that the two dashed vertical lines in the lateral resistance plots correspond to lateral resistance ratios of 0.9 and 1.1, indicating $\pm 10\%$ change in lateral resistance. The 10% change will be used to judge the extent of influence throughout the rest of the paper. The resistance ratio below the lower sand interface was not processed since the mesh is becoming coarse and the results are affected by mesh effects and numerical differentiation, and the pile displacements are very small.

Besides the effect the sand interlayer has on the clay layer, it is interesting to observe in Figure 501.27

---

[2] The lateral resistance ratio is only shown for the upper clay layer since the resistance corresponding to large y is not available at larger depth due to the fact that the pile is loaded at the pile head and the deflection decreases quickly as depth increases. Also due to the limit of space, plots for 2.0%, 2.5% pile width are not shown in this paper.

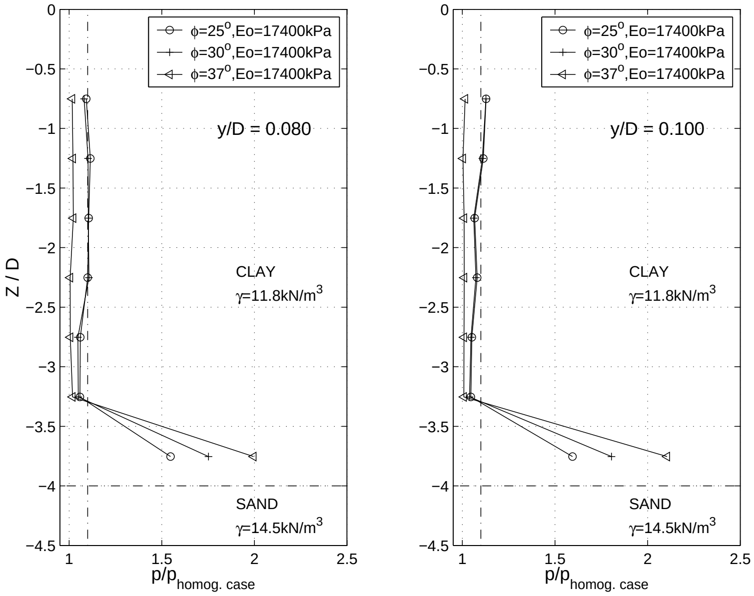


Figure 501.28: Lateral resistance ratio distributions in clay layer ($C_u = 21.7\ kPa, E_o = 11000\ kPa$) for sands with various ϕ at lateral displacements of 8% and 10% pile width.

that the soft clay layers also have significant effect on the lateral resistance of the intermediate sand layer. The lateral resistance ratios are less than 0.9 throughout the interlayer of sand. Surprisingly, the effects are not symmetric at lateral deflection of 0.5%. The resistance ratio is 0.85 at 0.25D below the upper sand interface, while that is 0.72 at 0.25D above the lower sand interface. This non–symmetry is probably due to the non–symmetric deformation[3] mode in the pile. As the pile is loaded laterally at the pile head, the right–hand–side sand close to the pile below certain depth tends to move downward to the right, which can be observed in Figure 501.32 (b). Therefore, the sand close to the upper interface moves against sand, while that close to the lower interface moves against soft clay. This type of movement results in the larger reduction in resistance at the lower sand interface than at the upper sand interface. The decrease in lateral resistance is mainly due to the lower stiffness in the adjacent soft clay layers. In addition, the smaller unit weight of the soft clay results in smaller mean effective normal stresses in the sand layer than the homogeneous model, which will reduce the stiffness of the sand and therefore also contribute to the reduction in lateral resistance at the intermediate sand layer.

---

[3] Non–symmetric with respect to the horizontal plane in between the interfaces (midway through the sand layer).

501.3.4.2 Uniform Sand Deposit and Sand Deposit with an Interlayer of Soft Clay.

By comparing the p–y curves of uniform sand deposit and sand deposit with an interlayer of soft clay, it was found that the effect of soft clay on the lateral resistance of sand propagates further away from the interface than Clay-Sand-Clay case, as described above in subsection 501.3.4.1. In addition to that, it was found that the heave in front of the pile will affect the lateral resistance of sand at shallow depth. Therefore, for sand deposit with an interlayer of soft clay, the thickness of upper sand layer was increased from 1.72 m to 2.36 m (the thickness of the soft clay layer was kept the same) to investigate the range of layering effects. Three models were analyzed by only varying the undrained shear strength C_u (i.e. 13.0, 21.7 and 30.3 kPa) of the soft clay layer.

Similar to the previous analysis, the p–y curves from the uniform deposit and the re-configured layered deposit were compared in Figure 501.29 and the lateral resistance ratios at several lateral displacements (i.e. 0.5%, 1.0%, 2.0%, 2.5%, 5.0% and 6.5% of pile width D) for all three models were computed and shown in Figures 501.30 and 501.31. It may be observed from Figure 501.29 that obvious difference may be observed in several $p--y$ curves further away from the interface.

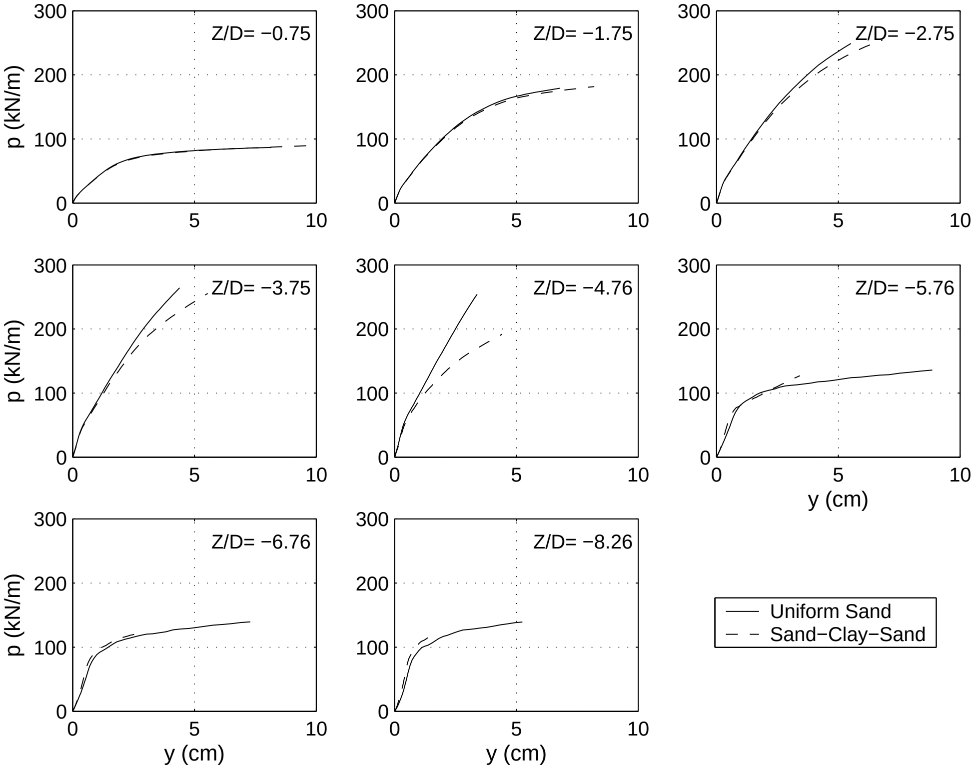


Figure 501.29: Comparison of p–y curves for uniform sand deposit versus sand deposit with an interlayer of soft clay (Sand: $\phi = 37^o$; Clay: $C_u = 21.7\ kPa$).

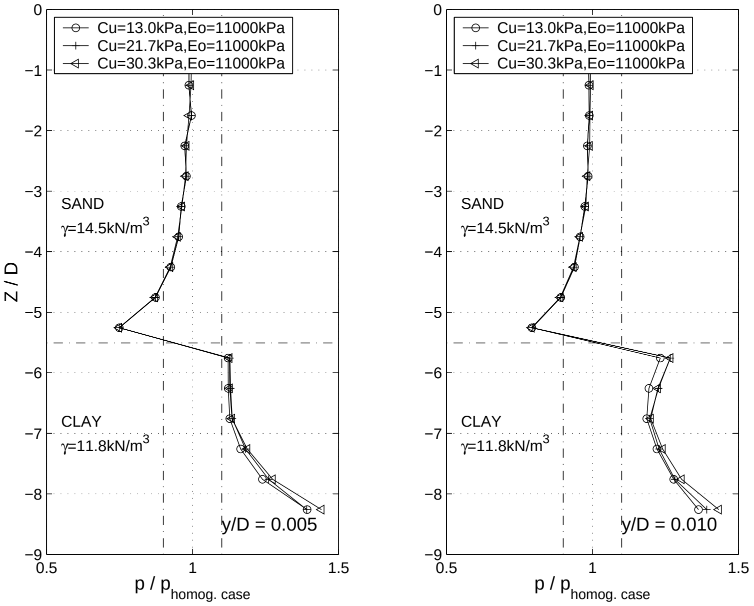


Figure 501.30: Lateral resistance ratio distributions (Sand:$\phi = 37^o, E_o = 17400\ kPa$) for clays with various C_u at lateral displacements of 0.5% and 1.0% pile width.

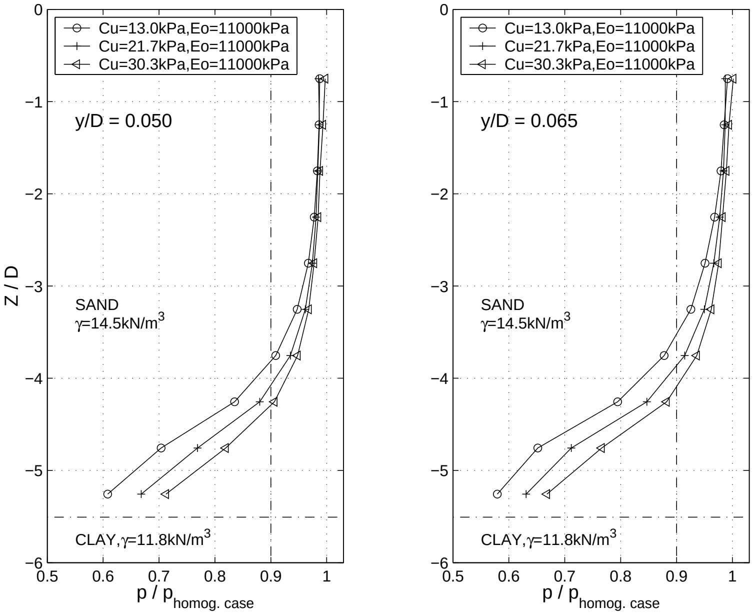


Figure 501.31: Lateral resistance ratio distributions (Sand:$\phi = 37^o, E_o = 17400\ kPa$) for clays with various C_u at lateral displacements of 5.0% and 6.5% pile width.

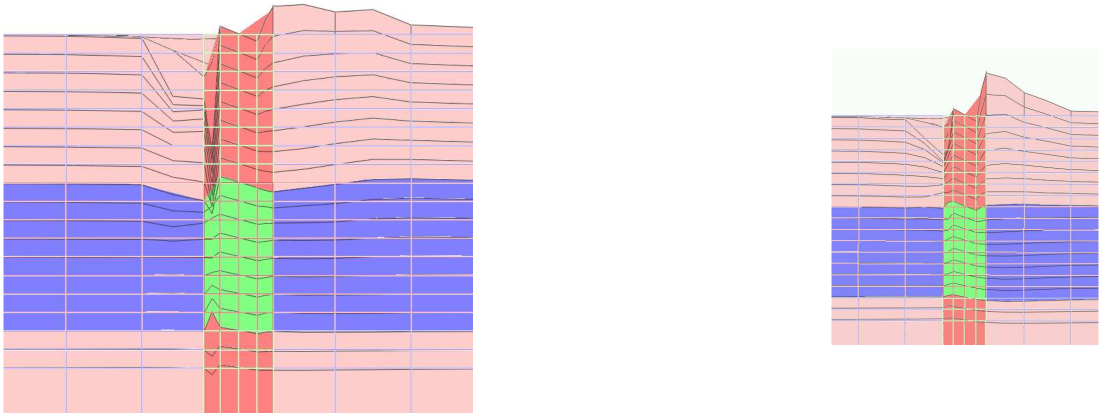
From Figure 501.30, it is noted that the effects of the intermediate soft clay layer are also independent of its undrained shear strength at small lateral displacements ranging from 0.5% D to 1.0% D. When the lateral displacement is greater than 1.0% D, the change in C_u starts to affect the lateral resistance ratio, as shown in Figure 501.31. Similar to the Clay–Sand–Clay model, the effect of the intermediate soft clay layer reduces to less than 10% at one pile width above the clay interface. The lateral resistance ratio at 0.25D above the clay interface is about 0.75. For large lateral displacements ranging from 5.0% D to 6.5% D, the 10% change in lateral resistance extends to 1.5 D - 2 D, as can be observed in Fig. 501.31. It may be noted that, at a lateral displacement of 6.5% D, the lateral resistance ratio at 0.25D above the clay interface changes from 0.58 to 0.67 when C_u increases from 13.0 kPa to 30.3 kPa.

Figures 501.32 (a) and (b) show the details of displaced models around the interfaces for the Sand-Clay-Sand and Clay-Sand-Clay profiles, respectively. The deformed model was overlapped with undeformed model for comparison. Ground heave can be easily observed in front of the pile from both figures. It is noted from Figure 501.32 (a) that the sand crosses the upper clay interface and moves into the intermediate soft clay layer. The movement slightly strengthens the soft clay soil and partially causes the slight increase of lateral resistance at the top of soft clay layer. Most importantly, the movement will soften the sand close to the upper layer interface, due to the reduction of confinement to the sand. For the Clay-Sand-Clay profile, the stronger sand layer penetrates into the softer clay layers at both interfaces. This penetration softens the sand close to both interface, due to the same reason as above.

501.3.5 Parametric Study for the Lateral Resistance Ratios in Terms of Stiffness and Strength Parameters.

To further investigate the effects of soil stiffness on the lateral resistance ratios at small displacement and/or large displacement, further analyses were carried out for the Clay–Sand–Clay and Sand–Clay–Sand models by changing both stiffness parameter (i.e. E_o) and strength parameter (C_u for clay, or ϕ for sand) using the same finite element models as above. The model configurations and intermediate layer soil parameters were summarized in Tables 501.2 and 501.3.

Lateral resistance ratios were plotted in Figures 501.33 and 501.34 for the Clay–Sand–Clay model, and in Figures 501.35 and 501.36 for the Sand–Clay–Sand model. By comparing Figures 501.28 and 501.34 for pile displacements of 8% D and 10% D, and Figures 501.31 and 501.36 for pile displacements of 5% D and 6.5% D, it is clear that the lateral resistance ratios are almost the same for the upper layer soil even if the stiffness parameter E_o of intermediate layer soil was varied by more than 30%. However, the lateral resistance ratios at small displacement (0.5% D and 1.0% D) were obviously influenced by the variation of E_o, as can be observed by comparing Figures 501.27 and 501.33 for the Clay–Sand–Clay model, and Figures 501.30 and 501.35 for the Sand–Clay–Sand model. For medium pile displacements



(a) (b)

Figure 501.32: Details of displaced model indicating ground heave and movement of soils across the layer interfaces at lateral load of 400 kN: (a) sand deposit with an interlayer of soft clay and (b) clay deposit with an interlayer of medium sand. The pile elements are removed so that the interface layer in the middle can be seen clearly.

Table 501.2: Summary of model configurations and intermediate sand layer parameters for Clay–Sand–Clay model in the parametric study.

Case	Soil Profile	Depth of Interfaces		Intermediate Sand Layer	
		Upper	Lower	E_o (kPa)	ϕ
1	Clay–Sand–Clay	-1.72 m	-3.43 m	11500	25^o
2	Clay–Sand–Clay	-1.72 m	-3.43 m	13500	30^o
3	Clay–Sand–Clay	-1.72 m	-3.43 m	17400	37^o

Table 501.3: Summary of model configurations and intermediate clay layer parameters for Sand–Clay–Sand model in the parametric study.

Case	Soil Profile	Depth of Interfaces		Intermediate Clay Layer	
		Upper	Lower	E_o (kPa)	C_u (kPa)
1	Sand–Clay–Sand	-2.36 m	-4.08 m	8000	13.0
2	Sand–Clay–Sand	-2.36 m	-4.08 m	11000	21.7
3	Sand–Clay–Sand	-2.36 m	-4.08 m	12500	30.3

(e.g. 2% D and 2.5% D), both stiffness and strength parameters have effects on the lateral resistance ratios.

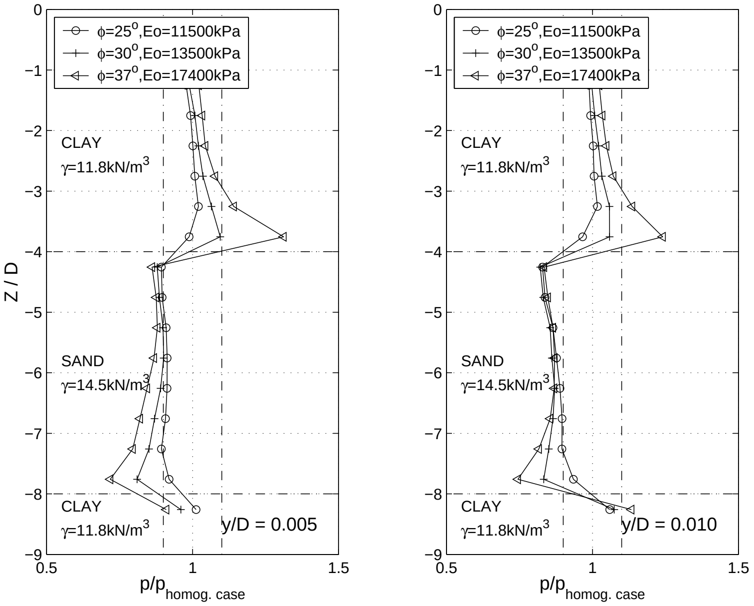


Figure 501.33: Lateral resistance ratio distributions (Clay:$C_u = 21.7\ kPa, E_o = 11000\ kPa$) for sands with various ϕ and E_o at lateral displacements of 0.5% and 1.0% pile width.

It will be useful to relate the effects of (a) the relative stiffness which controls the lateral resistance ratio at small lateral displacements and (b) the relative strength which determines the lateral resistance ratio at large lateral displacements with the lateral resistance ratio. To exclude the effects of unit weight, only the results above the upper interface in the Sand–Clay–Sand model are processed. The ratio of Young's moduli of clay and sand soils was used to define the relative stiffness $R_{stiffness}$ of the two layers.

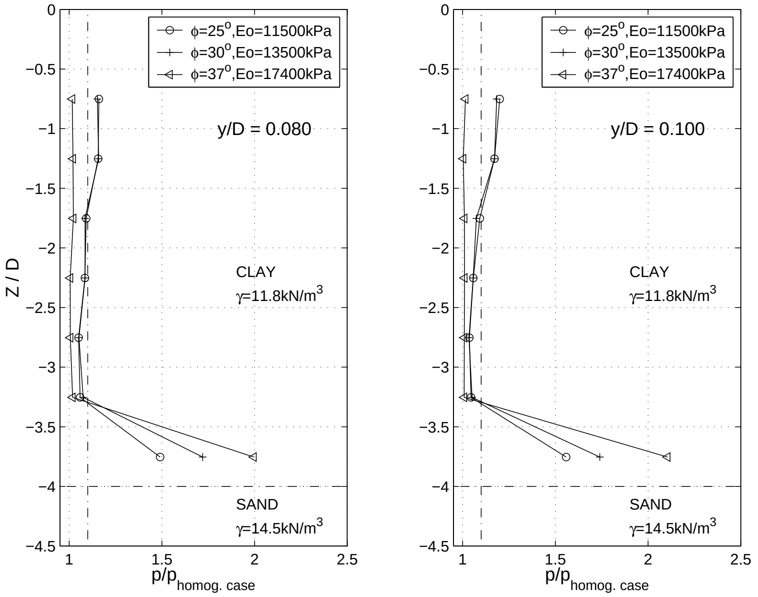


Figure 501.34: Lateral resistance ratio distributions in clay layer ($C_u = 21.7\ kPa, E_o = 11000\ kPa$) for sands with various ϕ and E_o at lateral displacements of 8% and 10% pile width.

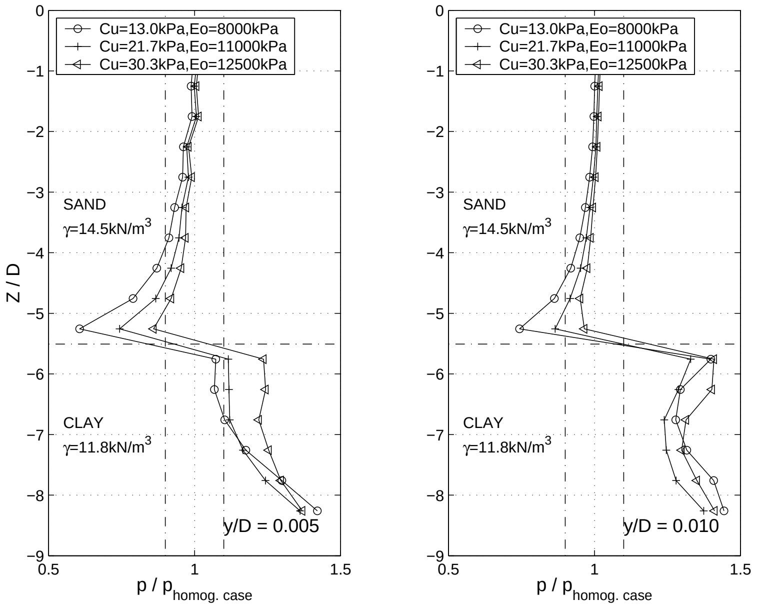


Figure 501.35: Lateral resistance ratio distributions (Sand:$\phi = 37^o, E_o = 17400\ kPa$) for intermediate layer of clays with various C_u and E_o at lateral displacements of 0.5% and 1.0% pile width.

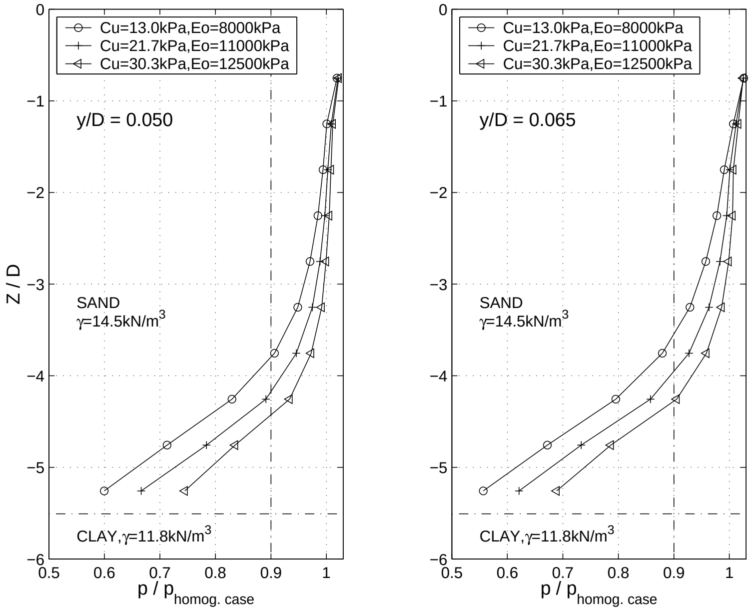


Figure 501.36: Lateral resistance ratio distributions (Sand:$\phi = 37^o, E_o = 17400\ kPa$) for clays with various C_u and E_o at lateral displacements of 5.0% and 6.5% pile width.

On the other hand, the ratio of largest lateral resistances of uniform clay and sand[4] at the upper interface (-2.36 m) was used to define the relative strength $R_{strength-FEM}$, as described in Equations (501.3) and (501.4).

$$R_{stiffness} = \frac{E_{o-clay}}{E_{o-sand}} \tag{501.3}$$

$$R_{strength-FEM} = \frac{p_{clay-FEM}}{p_{sand-FEM}} \tag{501.4}$$

The lateral resistance ratios at lateral displacement of 6.5% D were plotted against C_u in Figure 501.37. For comparison, the relative stiffness $R_{stiffness}$ and relative strength $R_{strength-FEM}$ were also included in the same plot.

As can be observed from this plot, the lateral resistance ratio decreases from 0.69 to 0.56 almost proportionally as C_u drops from 30 kPa to 13 kPa at 0.25 D above the upper interface, and the ratio is greater than the relative strength $R_{strength-FEM}$. Since the ultimate resistance of uniform sand will be larger than the computed largest value (which is still increasing , as can be observed from Figure 501.29 at Z=-3.75D) and that of uniform clay almost will almost remain the same (refer to Figure

---

[4] It would be better to use the ultimate lateral resistances for both clay and sand to define the relative strength but these values are not available from the current numerical results since pile displacement y is not large enough.

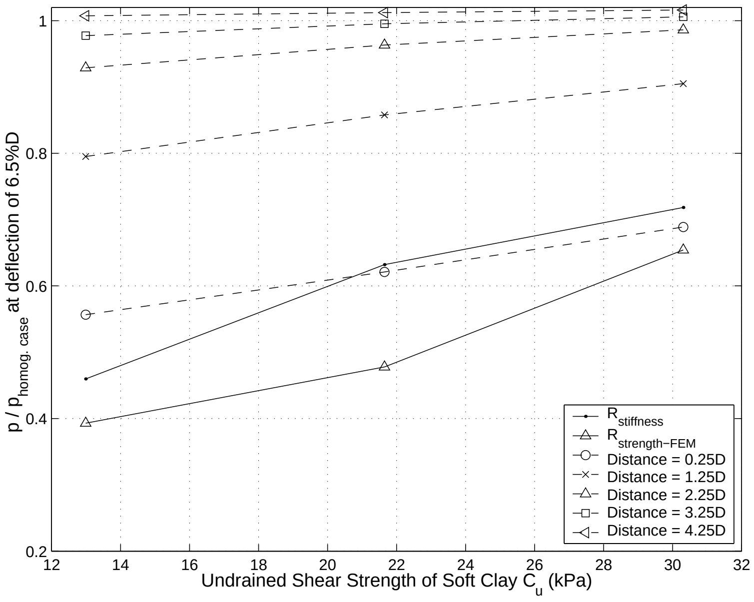


Figure 501.37: Lateral resistance ratios in the upper sand layer ($\phi = 37^o$) at various distances from the interface for pile displacement of 6.5% pile width.

501.26 at Z=-3.75D), this relative strength value will drop and the above statement still holds. There is certain correlation between the lateral resistance ratio close to the upper interface and $R_{strength-FEM}$ at 6.5%D pile displacement. As the distance to the upper interface increases, this correlation diminishes. The relative stiffness curve intercepts with the lateral resistance ratio curves at 0.25D above the upper interface. This implies that the presence of the clay, which is softer than the sand, somehow caused the layered system to be softer than either of the homogeneous models. This seems illogical, and in fact previous discussions and comparisons showed that $R_{strength-FEM}$ is more important than $R_{stiffness}$ at these large relative displacements.

It is also interesting to examine the relationship between the lateral resistance ratio and the relative variables (i.e. strength and stiffness) when lateral displacement increases, as presented in Figures 501.38 and 501.39. Figure 501.38 shows that the lateral resistance ratios at 0.25D above the interface decreases and come closer to the relative strength $R_{strength-FEM}$ curve as the lateral displacement increases from 4.0%D to 6.5%D. The relative stiffness $R_{stiffness}$ was also plotted in Figure 501.38 and it intercepts with the lateral resistance ratio curve, which has similar implications as the above discussion for Figure 501.37 and is illogical. On the other hand, as the lateral displacement decreases from 1.5%D to 0.5% D, the lateral resistance ratios keep decreasing and come closer to the relative stiffness ratio $R_{stiffness}$, as shown in Figure 501.39. There is almost a linear relationship between the lateral resistance ratio and the relative stiffness at small displacements.

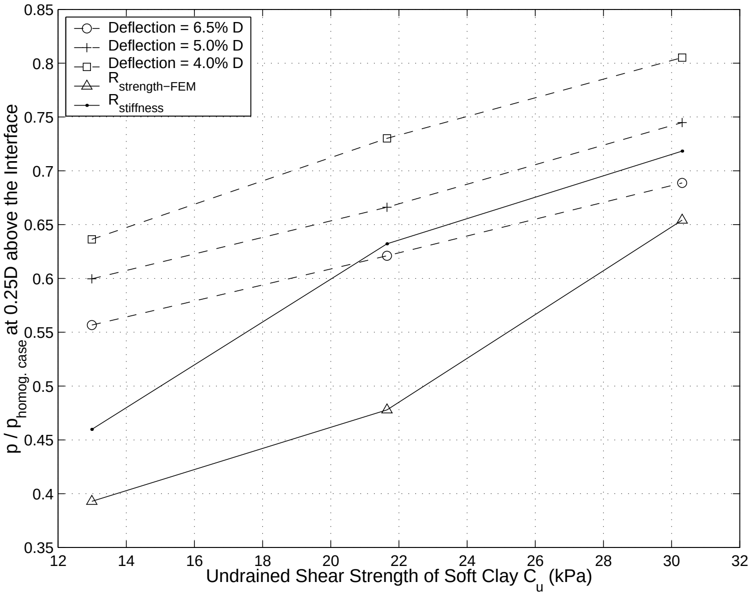


Figure 501.38: Lateral resistance ratio at a quarter pile width above the upper clay interface for various deflections.

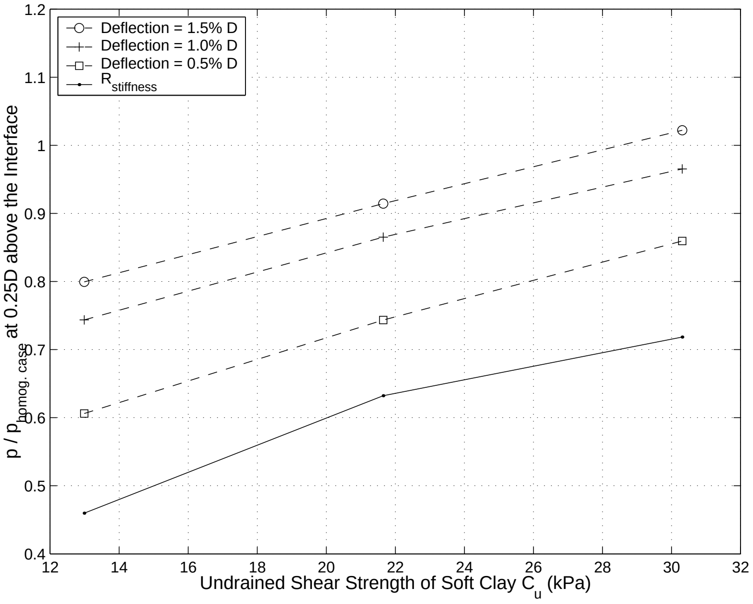


Figure 501.39: Lateral resistance ratio at one quarter pile width above the upper clay interface for clays with various C_u and E_o at small pile displacements ranging from 0.5% to 1.5% pile width.

From the above analysis, it is safe to say that the lateral resistance ratio is dominated by the relative stiffness $R_{stiffness}$ at small displacement (i.e. $\leq$ 0.5%D), while that is controlled by the relative strength $R_{strength-FEM}$ at large displacement (i.e. $\geq$ 4.0%D). For small displacement, the smaller the displacement is, the closer the lateral resistance ratio is to the relative stiffness; for large displacement, the larger the displacement, the closer the lateral resistance ratio is to the relative strength.

Figures 501.40, 501.41 and 501.42 summarize observed lateral resistance ratios in layered profiles. Figure 501.40 shows the lateral resistance ratios in the intermediate sand layer corresponding to various relative stiffness $R_{stiffness}$ at pile displacement of 0.5%D for the Clay–Sand–Clay model. Figures 501.41 and 501.42 show the lateral resistance ratios corresponding to various relative stiffness $R_{stiffness}$ and relative strength $R_{strength-FEM}$ at pile displacements of 0.5% D and 6.5% D for the Sand–Clay–Sand model. The effects of the intermediate clay layer on the upper sand layer reduce to less than 10% at a distance of 0.5 to 1.5 D above the interface at small pile displacement (e.g. 0.5% D), while that effects reduce to less than 10% at a distance of 1.25 to 2.0 D above the interface at large pile displacement (e.g. 6.5%D).

One may notice that the lateral resistance ratios corresponding to the relative stiffness $R_{stiffness} =$ 0.63 in Figures 501.40 and 501.41 are not the same. The ratios close to the lower sand interface in the Clay–Sand–Clay model is slightly larger than that in the Sand–Clay–Sand model. This difference is due to the fact that the lateral resistance ratios in the intermediate sand layer also include the effects of smaller unit weight of upper layer clay.

501.3.6 Summary

This subsection summarizes results from finite element analysis on the behavior of a single pile in elastic–plastic layered soils. Based on the results presented, the following conclusions can be drawn.

1. The layering effects are two–way. Not only the lower layers are affected by the upper layers, but the upper layers are also affected by the lower layers. Furthermore, the layering effects are not symmetric. In the case of pile laterally loaded at the pile head, the effect of an interface extends further into the layer above the interface than it does into the layer below the interface at small displacements.

2. In the Clay–Sand–Clay model, the lateral resistance of soft clay increases by as much as 30% and the effect extends to one pile width above the upper sand interface for $R_{stiffness} = 0.63$ at small pile displacement (0.5%D). Nonetheless, the increase of lateral resistance in the upper clay layer at large pile displacement (8–10%D) extends only one finite element above the upper sand interface.

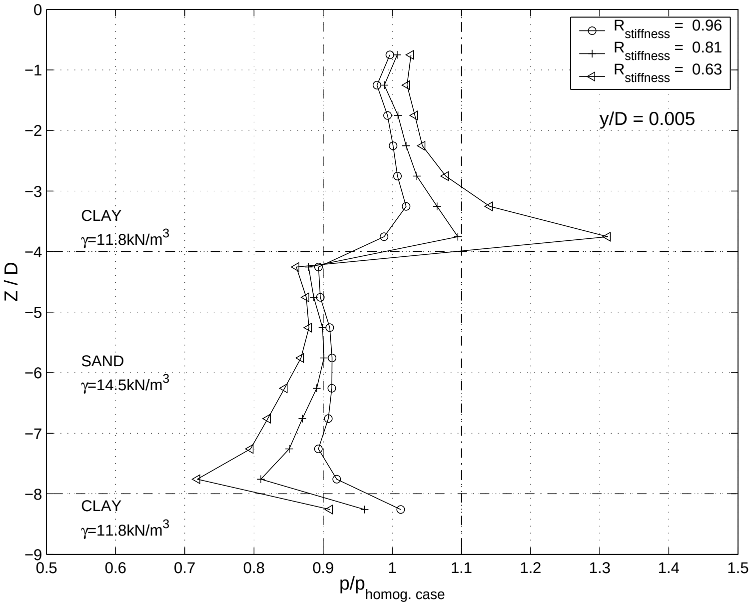


Figure 501.40: Summary of observed lateral resistance ratios from FEM analysis for the Clay–Sand–Clay profile at small deflection (y/D=0.5%).

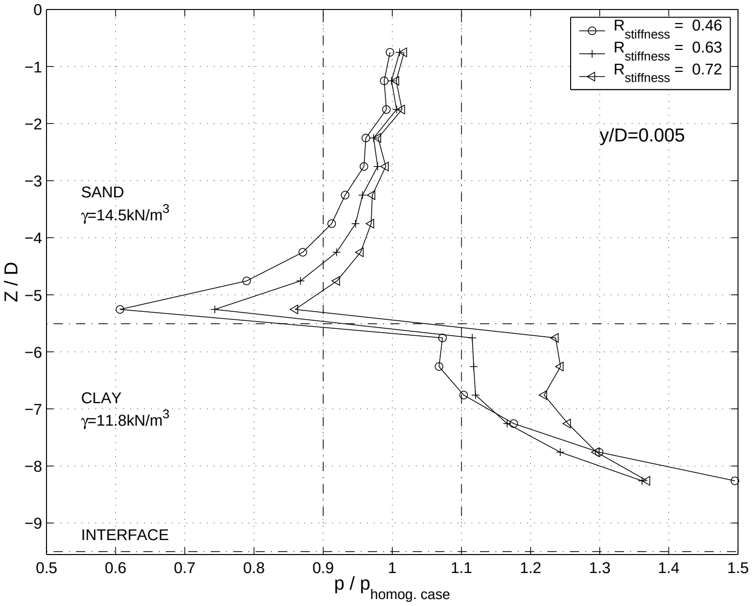


Figure 501.41: Summary of observed lateral resistance ratios from FEM analysis for the Sand–Clay–Sand profile at small deflection (y/D=0.5%).

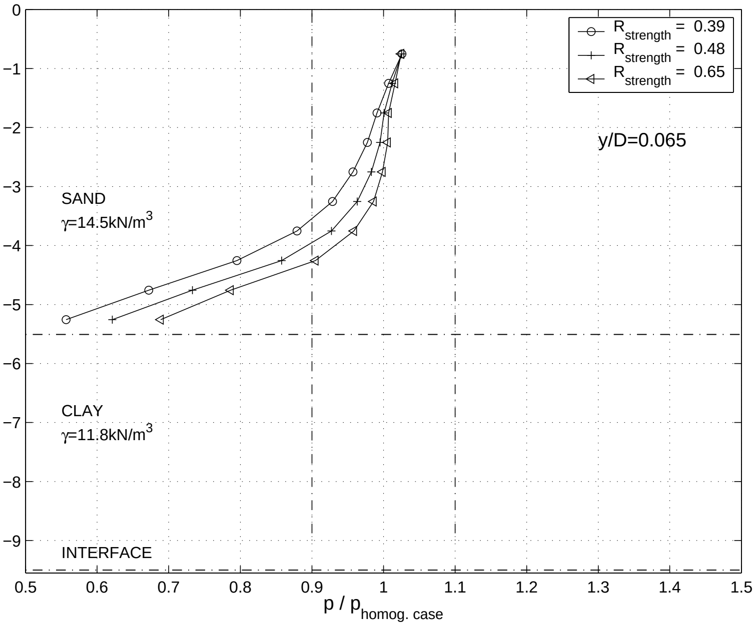


Figure 501.42: Summary of observed lateral resistance ratios from FEM analysis for the Sand–Clay–Sand profile at large deflection (y/D=6.5%).

On the other hand, the clay layers also have significant effects on the lateral resistance of sand throughout the intermediate layer.

3. In the Sand–Clay–Sand model, the intermediate clay layer has considerable effects on the lateral resistance of the upper sand layer, and the sand layers also have significant effects on the lateral resistance of the intermediate clay layer, causing 10 to 40% increase in its lateral resistance.
4. The lateral resistance ratio is dominated by the relative stiffness at small displacements (i.e. $\leq$ 1.0%D), while that is controlled by the relative strength at large displacements (i.e. $\geq$ 5.0%D).

It must be pointed out that the above observed lateral resistance ratios may only be applied to similar stratigraphies, pile deformation modes, and other conditions considered in this work. Further analyses are needed to investigate the effects of other stratigraphies, pile deformation modes, pile diameters, and other factors, in order to draw more general guidelines. Future studies with a refined mesh around the interface will provide better resolution of the resistance ratio around the interface. Future studies of the effects of the interface layer on the layering effects will also be very interesting.

501.4 Numerical Study of Group Effects for Pile Groups in Sands

Material presented here has been previously published in our paper Yang and Jeremić (2005).

501.4.1 Introduction

Single pile foundations have been successfully modeled by the $p-y$ approach, as implemented in LPILE (Reese et al. Reese et al. (2000a)). However, the behavior of a pile within a group may differ greatly from that of a single pile and vary depending on the position due to the interaction between the neighboring piles. To study this interaction effects, only a couple of field tests have been carried out because of the large costs incurred. Brown et al. Brown et al. (1988) conducted cyclic loading tests on instrumented 3×3 steel pile group. The p–multiplier concept was presented based on the measured soil resistance data and specific p–multipliers were suggested for the three rows. Ruesta and Townsend Ruesta and Townsend (1997) reported an in-situ test on piles at Roosevelt Bridge. Rollins et al. Rollins et al. (1997) tested another full–scale pile group founded in clay and suggested a set of p–multipliers for corresponding pile groups. Ng et al. Ng et al. (2001) presented results on full–scale lateral load tests of one single pile and three pile groups with large–diameter bored piles. Besides in-situ testing, many centrifuge tests were conducted to predict the behaviors of pile groups under static and dynamic loading. Recently, McVay et al. McVay et al. (1995) McVay et al. (1998) conducted a series of lateral load tests on large pile groups (3×3 to 7×3) founded in sands to study the interaction effects within a group.

Based on these field and centrifuge tests, it was found that:

1. All the test results have clearly shown that the lateral resistance of a pile within the group is strongly influenced by its row position and the p–multiplier method was suggested by Brown et al. Brown et al. (1988) to account for this behavior. Specifically, each row within the group is assigned a different p–multiplier f_m and the p–y curve for a single pile is multiplied by f_m to produce p–y curves for all the piles in the same row.

2. The p–multipliers are independent of soil density and only depend on the pile geometry. And at sufficient deformation and under static loading, they are constant for practical purposes (McVay et al. McVay et al. (1998)).

3. For 3–diameter spacing, the suggested p–multiplier f_m was 0.8, 0.4, 0.3 (front row to back row) for the 3×3 group (Brown et al. Brown et al. (1988)), 0.8, 0.4, 0.3, 0.3 for the 4×3 group, and 0.8, 0.4, 0.3, 0.2, 0.3 for the 5×3 group, and 0.8, 0.4, 0.3, 0.2, ..., 0.3 for all larger group size (McVay et al. McVay et al. (1998)).

4. For concentric loading (located at the geometrical center of the pile group), the difference between the side and middle piles within a row is small and may be neglected, while the moments in the side piles within a given row are slightly larger than that in the middle piles but may be represented by the average (McVay et al. McVay et al. (1998)).

Together with the physical modeling, a few numerical simulations have also been performed. We mention a few representative finite element studies of pile groups. Maqtadir and Desai Muqtadir and Desai (1986) studied the behavior of a pile-group using a three dimensional program with nonlinear elastic soil model. An axisymmetric model with elastic-perfectly plastic soil was used by Pressley and Poulos Pressley and Poulos (1986) to study group effects. Brown and Shie Brown and Shie (1990a) Brown and Shie (1990b) Brown and Shie (1991) and Trochanis Trochanis et al. (1991) conducted a series of 3D Finite Element Method (FEM) studies on the behavior of single pile and pile group with elastic-plastic soil model. In particular, interface element was used to account for pile-soil separation and slippage. Moreover, several model and field tests of free- or fixed-head pile groups have been analyzed by Kimura et al. Kimura et al. (1995) and Wakai et al. Wakai et al. (1999) using 3D elasto-plastic FEM.

This paper describes 3D elastic-plastic finite element modeling of two pile groups founded in sands with emphasis on the interaction effects within pile group. Specifically, bending moment and load distribution in individual piles were examined and compared with centrifuge test data. Special attention was given to out-of-loading-plane bending moment and p–y behavior of individual piles in a group. The OpenSees OpenSees Development Team (Open Source Project) (2000-2006) finite element framework was employed to complete all the computations. Soil modeling was performed using the Template Elastic–Plastic approach (Jeremić and Yang Jeremić and Yang (2002)).

This paper is organized as follows. Section 501.4.2 summarizes the centrifuge tests and describes finite element models including the soil elastic–plastic model used for 3×3 and 4×3 pile group simulations. Section 501.4.5 presents a number of results and discussion describing simulated behavior of analyzed pile groups. In particular, presented are developed plastic zones (Section 501.4.6), pile bending moments (Section 501.4.7), pile load distributions (Section 501.4.8), comparison of $p-y$ curves for individual piles (Section 501.4.9), and comparison with centrifuge tests (Section 501.4.10). Section 501.4.11 gives concluding remarks.

501.4.2 Pile Models

501.4.3 Summary of Centrifuge Tests

In the centrifuge tests reported by McVay et al. McVay et al. (1998), 3×3 to 7×3 pile groups embedded in homogeneous sands under lateral load were tested. The rectangular–shaped sample container was fabricated from aluminum alloy with an inside dimensions of 0.254 m wide, 0.457 m long, and 0.305 m high. The piles were spaced by three times the pile width and the pile caps (made of aluminum) were rigidly connected with the piles. The model square piles and pile cap were fabricated from solid square aluminum (alloy 6061) bars. Each individual pile is 9.5 mm wide and 304.8 mm long. To simulate the installing effects of field driven piles, the piles were driven in flight into sands by hydraulic equipment and tested at 45 g.

The sands (artificially mixed by a number of different gradations) studied were at two different relative densities: a loose sand with relative density Dr=36%, unit weight γ = 14.05 kN/m^3 and a medium dense sand with Dr = 55%, γ = 14.50 kN/m^3. The shear modulus G, Poisson's ratio ν and friction angle ϕ are 8230 kN/m^2, 0.35, 34.5^o for the loose sand and 8960 kN/m^2, 0.35, 37.1^o for the medium dense sand. It is noted that the friction angles were determined from drained triaxial compression tests, and the shear moduli were back–computed from instrumented vertical load tests (Zhang et al. Zhang et al. (1998)) and were valued at a depth of 13.7 m.

501.4.4 Finite Element Pile Models

Among these tested pile groups, 3×3 and 4×3 groups were chosen to be modeled in prototype scale using 3D elasto–plastic finite element method to investigate pile group interaction effects. The typical layout of 4×3 pile group is shown in Fig. 501.43. The whole centrifuge model in prototype scale is 22.8 m wide, 20.6 m long and 13.2 m deep. Only half of each centrifuge model is meshed considering the symmetry . Figure 501.44 shows the finite element mesh for the 4×3 pile group. Additional finite element analysis of a cantilever beam using the same mesh as an individual pile in the group was carried out and comparison of the displacement at the top of the beam from FEM and beam theory solution indicated that the mesh was fine enough to capture the pile behavior. Soil, pile and soil–pile interface are all modeled with twenty node brick elements. Each pile consists of four elements (per cross section) made of elastic material with properties corresponding to aluminum. There are 1268 and 1414 brick elements in the two models respectively. The sides and bottom of each model are fixed in all three coordinate directions with the exception of the symmetric boundary, which is only supported in the direction perpendicular to the symmetry plane. This type of boundary conditions are fairly close to the actual friction boundary conditions in the centrifuge tests.

All the parameters except the Young's moduli for sands were the same as from the centrifuge studies. The Young's modulus is assumed to depend on the mean effective normal stress p' (Manzari and Dafalias Manzari and Dafalias (1997)) as:

$$E = E_o \left(\frac{p'}{p_a} \right)^n \tag{501.5}$$

where E_o is Young's modulus at the atmospheric pressure, $p'=\sigma_{ii}/3$ is the mean effective normal stress, p_a is the atmospheric pressure, and n is constant for a given void ratio. Usually 0.5 is used for n. For the medium dense sand, the Young's modulus at the atmospheric pressure computed by Eqn. (501.5) (Lateral pressure coefficient $K_o = 0.55$ was used) from the back–computed shear modulus was 200000 kPa and then adjusted to be 17400 kPa for the medium dense sand, so as to well simulate the load–displacement curve obtained from centrifuge tests. For the loose sand, the computed Young's modulus at the atmospheric pressure from the back–computed shear modulus was 18700 kPa and similarly adjusted to be 16000 kPa.

Sand was simulated by Drucker–Prager material model with nonassociated flow rules. Since the centrifuge studies we used to compare our simulations against did specify only the friction angle of test sands which were obtained from drained triaxial compression tests, the yield surface was chosen to agree with Mohr–Coulomb hexagon at triaxial compression. Future study using Mohr–Coulomb material model will be useful to determine the effects of varying friction angle in Drucker–Prager model on the results. Since there is no test data on the dilation angle ψ of the tested sands, a dilation angle of 0^o was used in this work, as similar dilation angle was also used in Brown and Shie Brown and Shie (1990a). The soil–pile interface was represented by one thin layer of elements. The material of the interface element was also simulated by Drucker–Prager model with a friction angle of 25^o, a dilation angle of 0^o, and the same Young's modulus and Poisson's ratio as corresponding sands. In the future, a realistic dilation angle needs to be used for sand to further investigate the effects of dilation angle on the pile group interaction behavior, especially the out–of–loading–plane bending moment.

501.4.5 Simulation Results

In this section we present results related to the behavior of 3×3 and 4×4 pile groups in loose and medium dense sands. A number of static pushover tests were simulated using FEM. Specifically, modified Newton–Raphson method were used to solve the system of equations in the finite element level and implicit algorithm (Jeremić and Yang Jeremić and Yang (2002)) was used in constitutive level integration. Results are also compared with those from the centrifuge studies by McVay et al. McVay et al. (1998) and Zhang et al. Zhang et al. (1999).

Figure 501.43: Layout of 4 × 3 pile group: a) top view, b) side view.

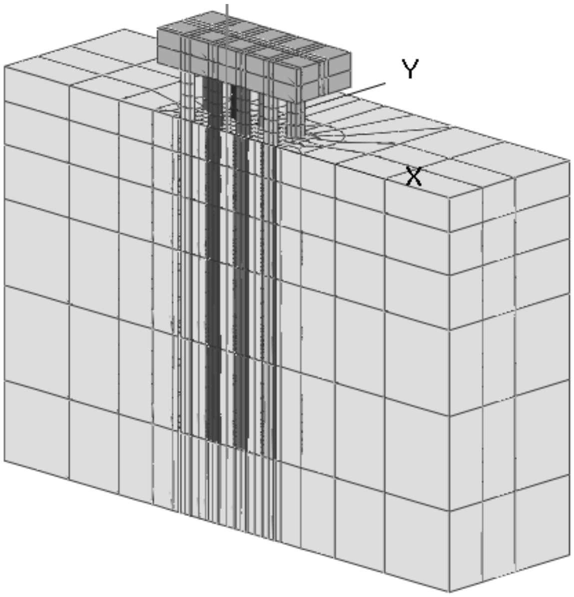


Figure 501.44: Finite element mesh for half of 4 × 3 pile group.

501.4.6 Plastic Zone

The static pushover tests were conducted using load control at the pile head with the loading applied in the X direction. The final plastic zones (represented by plastified Gauss points) for two pile groups are depicted in Figures 501.45 and 501.46. In particular, Figure 501.45 shows different views of the 3D plastic zone developed in the 3 × 3 pile group at the lateral load of 2,200 kN. Figure 501.46 shows the different views of the 3D plastic zone developed in the 4 × 3 pile group at the lateral load of 2,970 kN. Both figures clearly show a wedge shaped plastic zone at the shallow depth. Also apparent is the propagation of the plastic zone (shear yielding) along the pile–soil interface, resulting from the rocking behavior of the group.

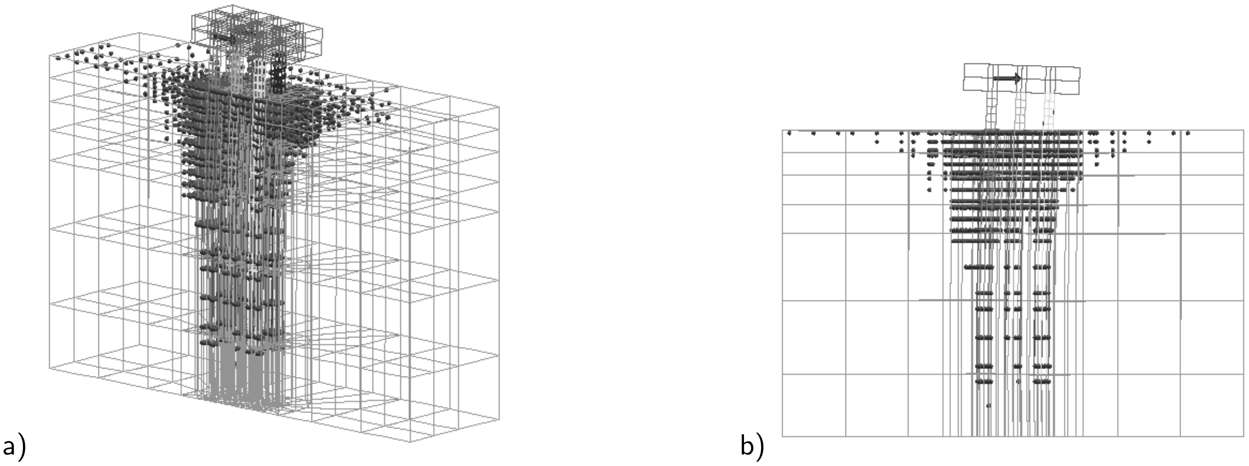


Figure 501.45: Plastic Gauss–Points for the 3×3 pile group: (a) 3D view and (b) side view.

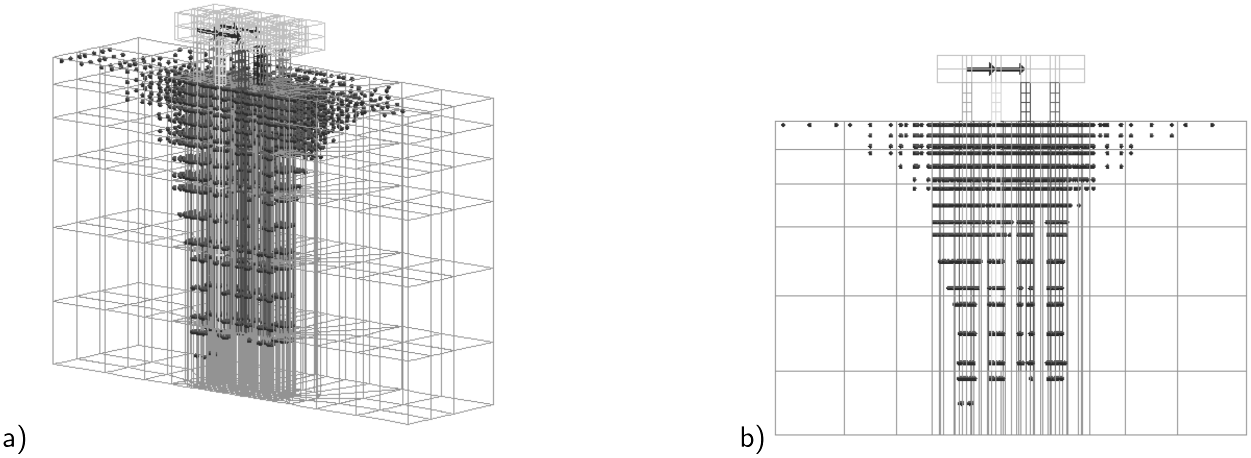


Figure 501.46: Plastic Gauss–Points for the 4×3 pile group: (a) 3D view and (b) side view.

501.4.7 Bending Moment

The maximum bending moment in the piles often controls the design of pile groups and therefore has to be analyzed accurately. In order to generate the bending moments from the stress field in piles vertical stresses at Gauss points from each pile element are integrated numerically. The moments with reference to the Y-axis (M_y) of each individual pile in the 3×3 and 4×3 pile groups are plotted in Figure 501.47 and 501.48, respectively. Figure 501.47 shows the moment diagrams of the 3×3 group at a lateral load of 2200 kN. The calculated maximum moment for each pile occurs at the pile cap, which is consistent with the fixed pile cap condition. The lead row piles, for both side and middle piles, in both 3×3 and 4×3 pile groups, carry the maximum bending moment. It is also interesting to note that in the 4×3 group, the moment diagrams for the piles in the third and fourth rows appear to be almost identical, which implies that they behave almost the same within the group.

The variations of maximum bending moments in each pile of 3×3 and 4×3 groups are illustrated in Figures 501.49 and 501.50, respectively. The maximum moments develop in the lead–row side piles, while the smallest maximum moments occur in the trail–row middle piles in both groups. It is obvious that the maximum moments developed in the middle and side piles within the lead row are quite different for both pile groups, implying the load shared by each pile in the same row is different. For example, in the 3×3 group, the maximum moment in the middle pile at the end load is 600 kN.m, while that of the side pile is 670 kN.m, the difference is about 11%. For the 4×3 pile group, the maximum moment on the lead–row side pile was about 10% greater than that for the lead–row middle pile.

It is interesting to look at the moments with reference to the X-axis (out-of-loading-plane moment, M_x). Figure 501.51 (a, b) shows M_x diagram for each pile as well as deformed piles for the 3×3 pile group. Similar plots for the 4×3 pile group are shown in Figure 501.53 (a, b). The maximum value of M_x in the 3×3 pile group reaches 50 kN.m, which is about 8% of the maximum value of M_y. For the 4×3 pile group, the maximum value of M_x is about the same amount, which is about 6% of the maximum value of M_y.

It is noted that the signs of the M_x moments in the lead and trail rows are different, indicating the bending directions are opposite to each other. This is further verified by looking at the deformed shape of the pile group shown in Figure 501.51(b) with only the displacement in Y direction shown. This kind of bending is caused by the the complex displacement field of the soil surrounding the pile group, as illustrated by the horizontal displacement vector and contour plots of displacement in X direction in Figures 501.52 and 501.54 for the 3×3 and 4×3 pile groups, respectively. The soil in front of the lead row tends to "squeeze into" the group, while the soil outside of the trail row tends to "come back" toward the pile group when the pile group is moving forward, which consequently results in the fact that

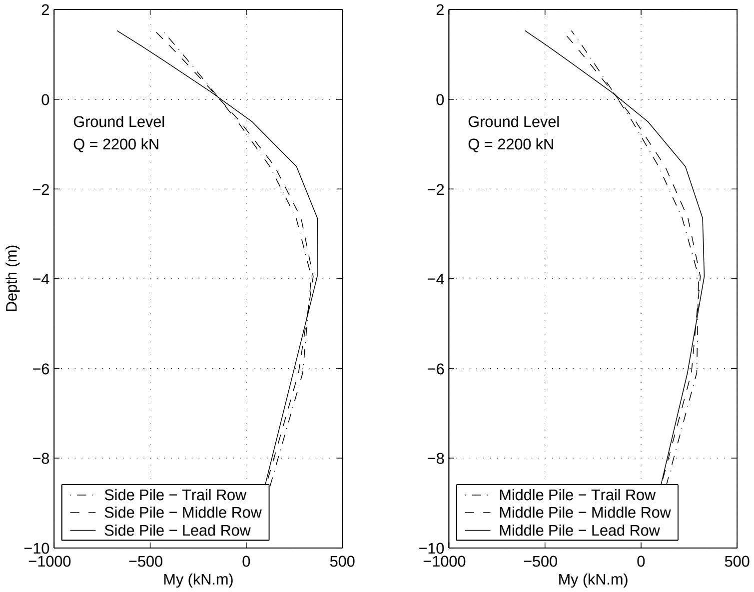


Figure 501.47: Comparison of bending moment diagram at lateral load of 2200 kN for piles in 3×3 group.

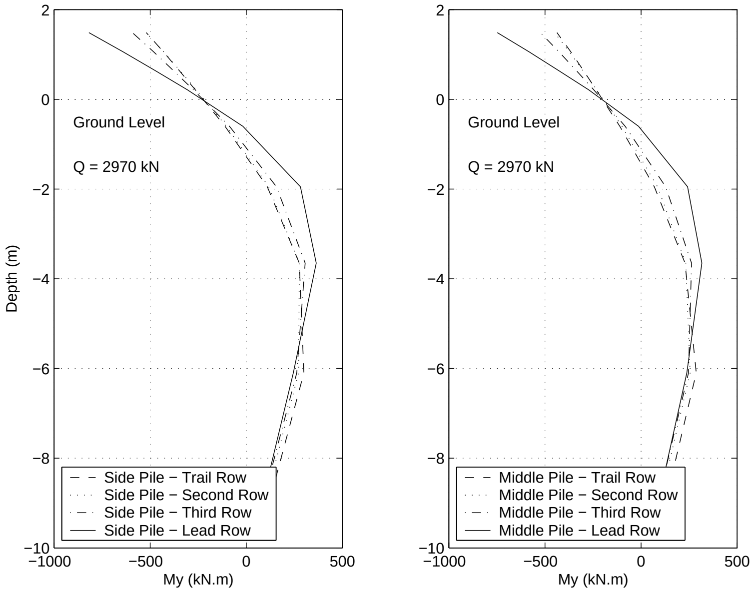


Figure 501.48: Comparison of bending moment diagram at lateral load of 2970 kN for piles in 4×3 group.

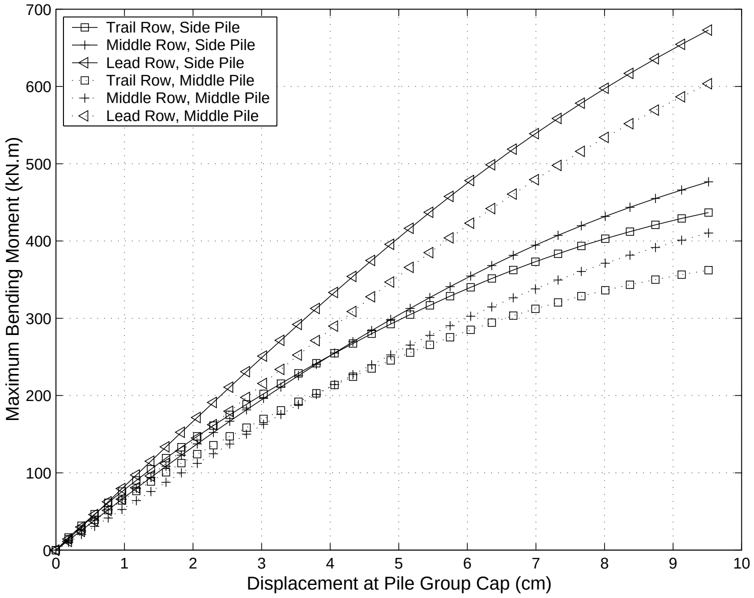


Figure 501.49: Maximum bending moments in individual piles in 3×3 group.

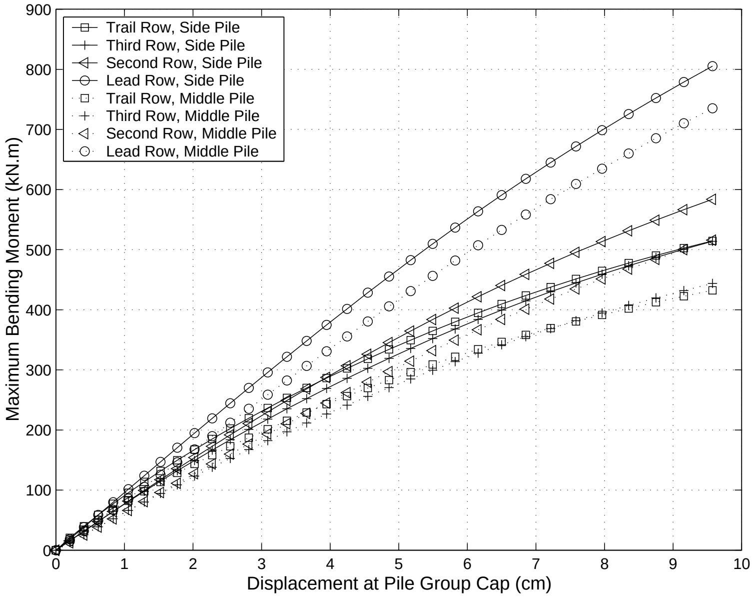


Figure 501.50: Maximum bending moments in individual piles in 4×3 group.

the lead row bends outward and the trail row bends inward.

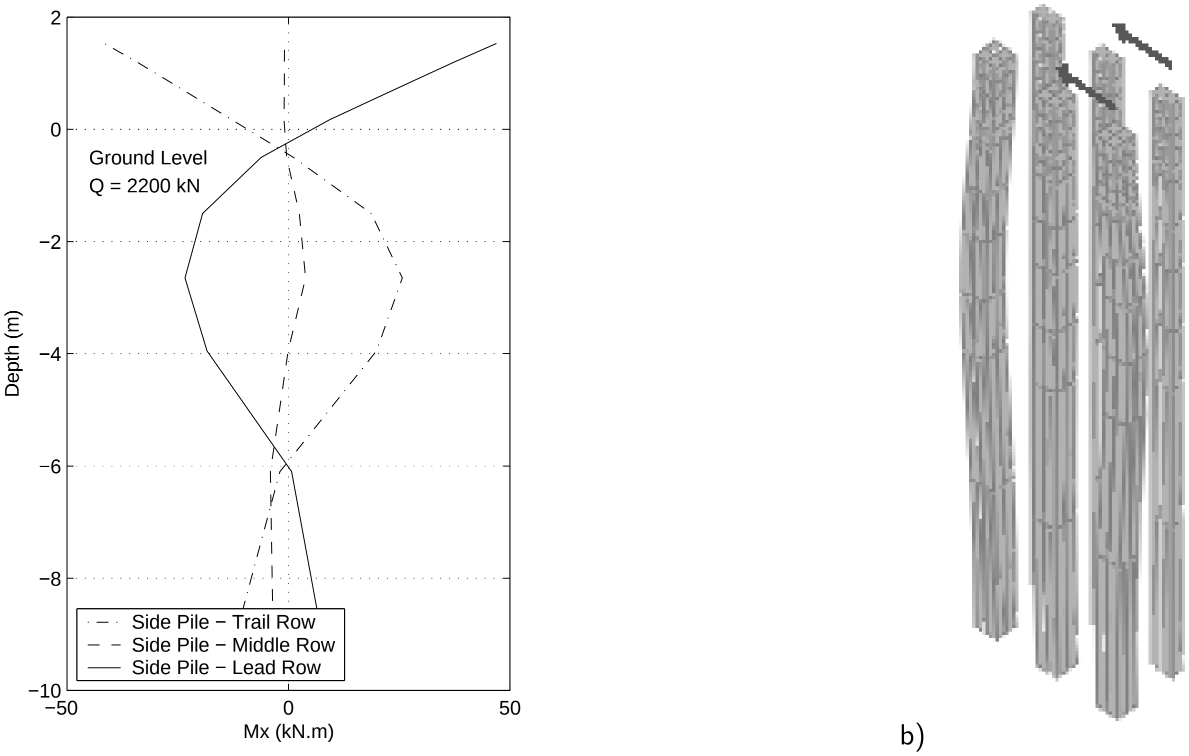


Figure 501.51: (a) Out-of-loading-plane bending moment diagram and (a) Out-of-loading-plane deformation for the 3 × 3 pile group.

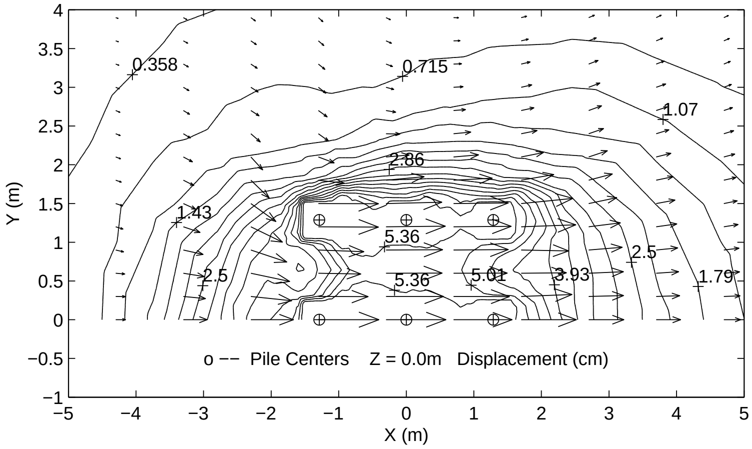


Figure 501.52: Horizontal displacement vector and contour of displacement in X direction at ground surface for the 3 × 3 pile group.

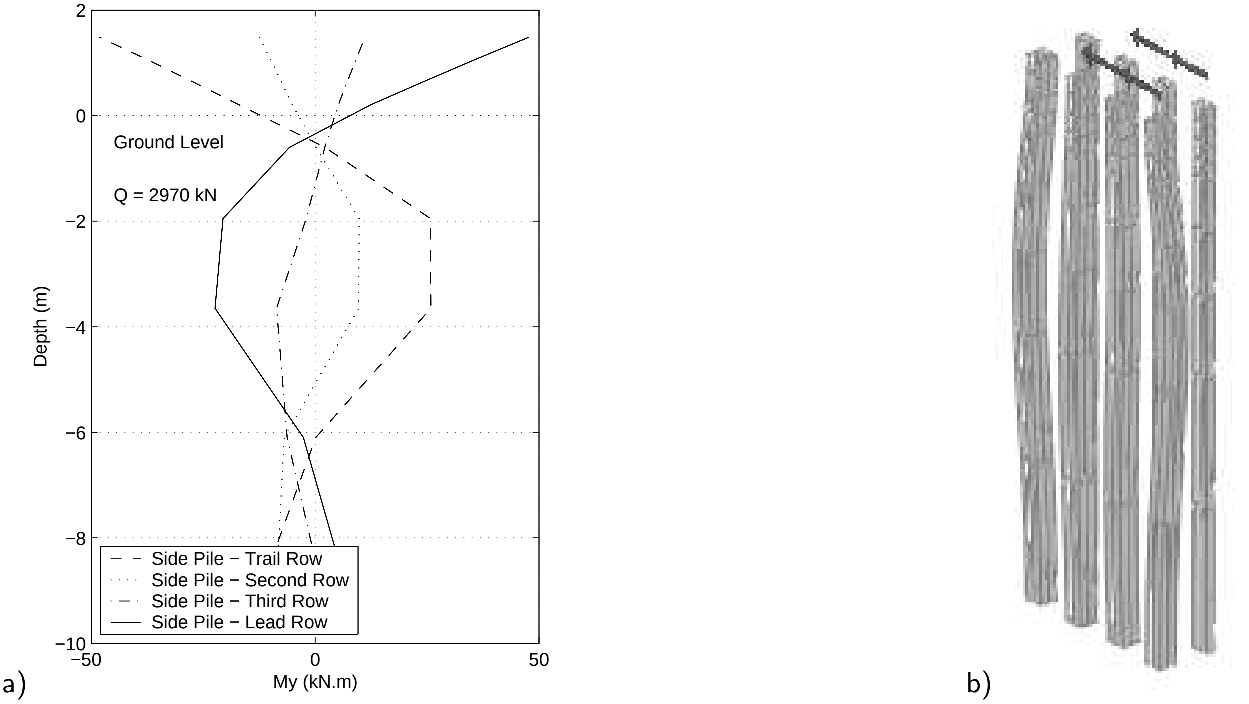


Figure 501.53: (a) Out-of-loading-plane bending moment diagram and (b) Out-of-loading-plane deformation for the 4×3 pile group.

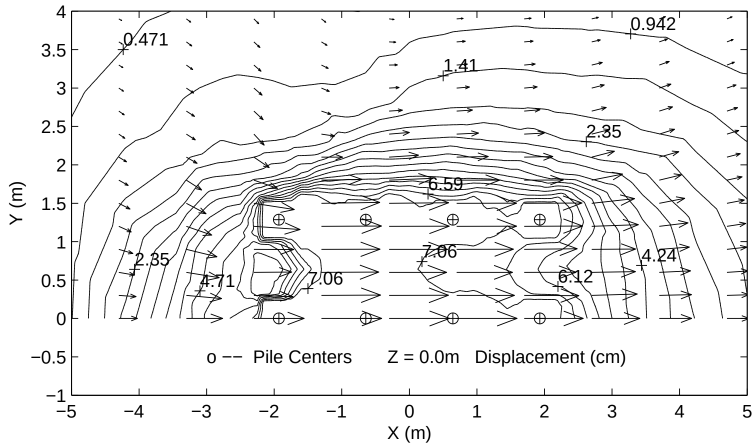


Figure 501.54: Horizontal displacement vector and contour of displacement (cm) in X direction at ground surface for 4×3 pile group.

501.4.8 Load Distribution

In order to compute the load taken by each pile, the values of bending moment at element centers along with the boundary condition at the bottom of pile (zero moment) were fitted with a 5th order polynomial by least square technique. According to the theory of beam on a Winkler-type subgrade (Hartog Hartog (1952)), the moment curve was differentiated once to compute the shear force. Then, the shear forces at three sampling points between the ground surface and the pile cap were averaged to compute the load carried by each individual pile. The accuracy of the load measuring scheme has been verified by comparing the total load actually applied on the pile cap and the sum of all loads carried by each individual pile.

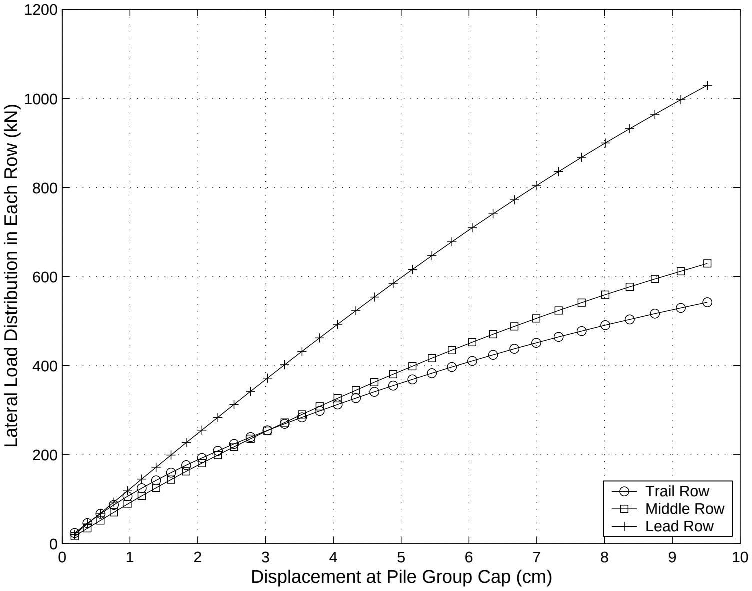


Figure 501.55: Variations of load taken by each row in the 3×3 group.

Figures 501.55 and 501.56 show the load and percentage of total load carried by each row of the 3×3 group. It is easily seen that not only the load but the percentage of total load taken by each row, especially the lead and trail rows, change steadily during loading process. It is observed in Figure 501.55 that the lead row and trail row share almost the same amount of load at small lateral displacement. However, as deflection increases, the lead row picks up the load much faster that the trail row, although the load taken by the three rows all increase.

More interestingly, the variation in the percentage of load carried by each row exhibits completely different trends. In the initial loading stage, the percentage shared by the lead row increases and that by the trail row drops quickly, while the percentage shared by the middle row almost remains constant.

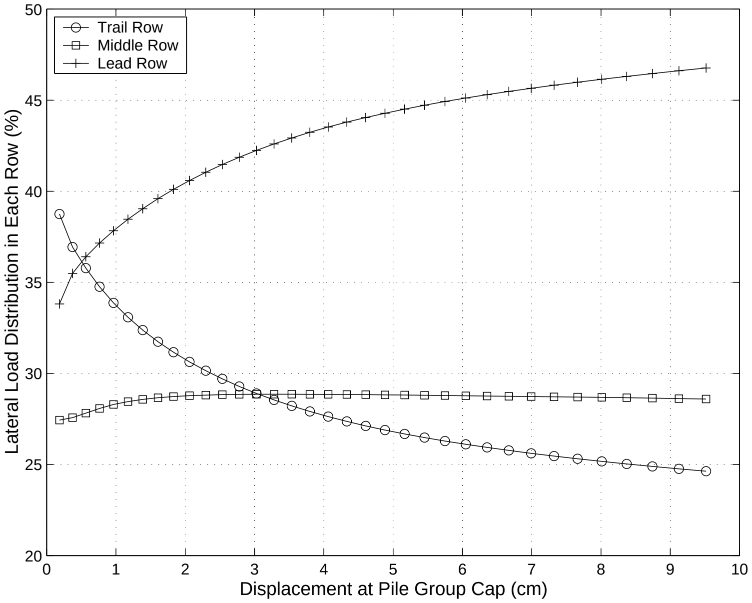


Figure 501.56: Variations of percentage of total load taken by each row in the 3×3 group.

In the final loading stage, however, the percentages carried by the lead and trail rows tend to stabilize. The lead row takes the most load, more than 46%, while the trail row takes the least, only around 25%. In addition, Figure 501.56 shows that the trail row takes a greater percentage of load than the lead row at small lateral deformation, which were attributed to some load measuring error and the fact that the denominator is relatively small.

Similar plots for the 4×3 group are shown in Figures 501.57 and 501.58. While the lead row still carries much more load that the trail row, the third row and the trail row share almost the same amount of load at large lateral displacement, which is in agreement with the fact that the same p–multiplier was recommended for the third and fourth row by McVay et al. McVay et al. (1998). It should also be observed from Figures 501.57 and 501.58 that the lead row carries more than twice the load of the trail row.

It is worthwhile noting that the distribution of load in the same row can be quite different. Figures 501.59 and 501.60 show the variations of load and the percentage of total load taken by each pile of the 3×3 group, respectively. Similar plots for the 4×3 group are shown in Figures 501.61 and 501.62. It is obvious that the piles at the sides take more load than the piles in the middle at the same row. For the 3×3 group, the side pile in lead row takes 350 kN or 16% of total load while the middle pile takes 325kN or 14.6% of total load at the end of loading.

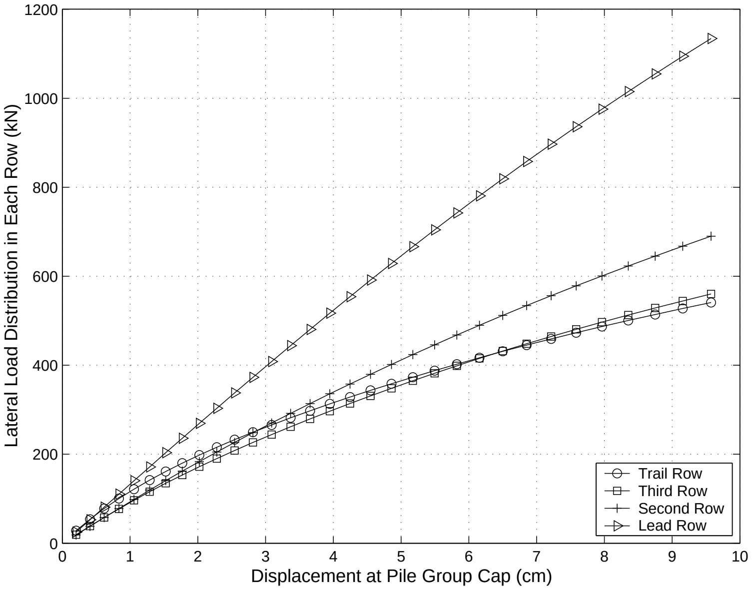


Figure 501.57: Variations of load taken by each row in the 4 × 3 group.

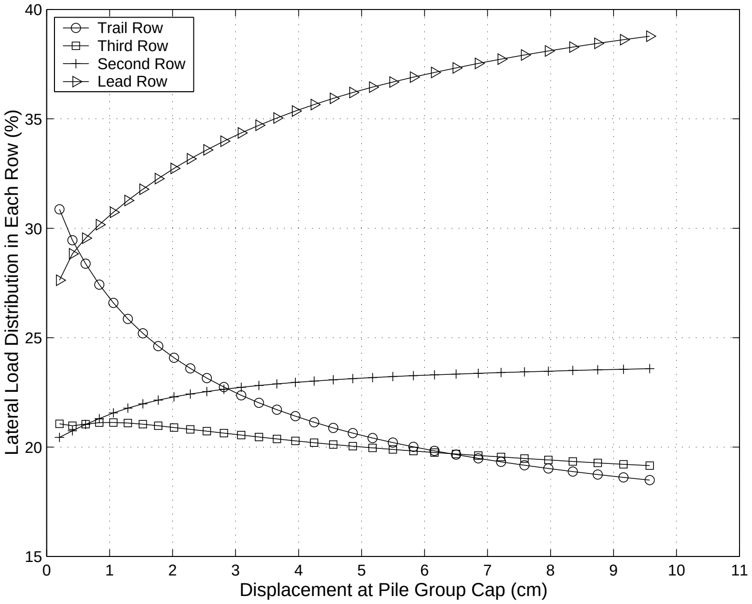


Figure 501.58: Variations of percentage of total load taken by each row in the 4 × 3 group.

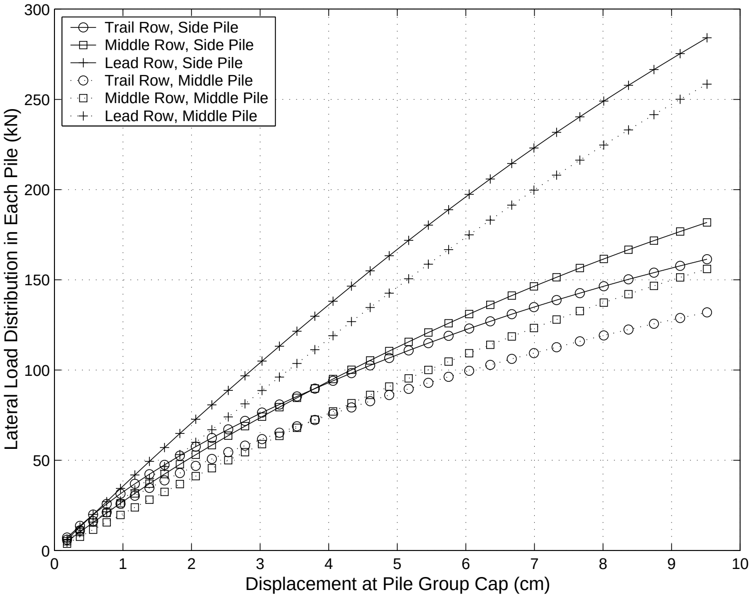


Figure 501.59: Variations of load taken by each pile in the 3×3 group.

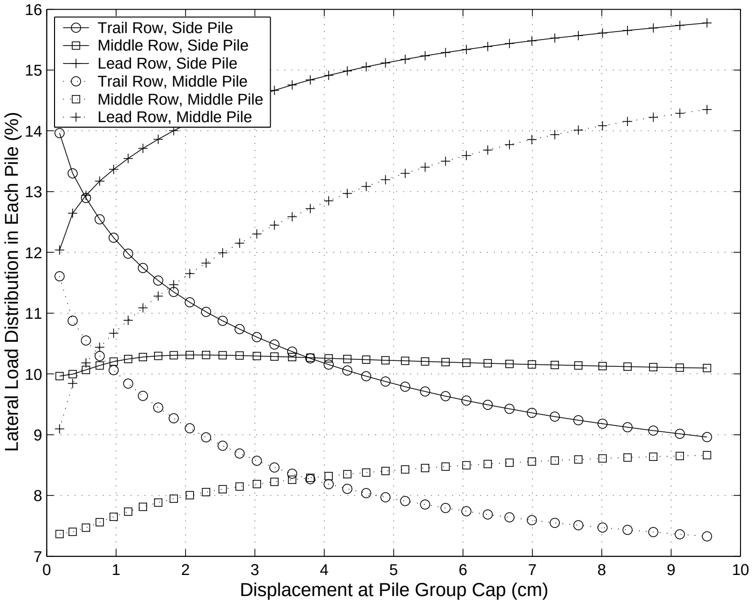


Figure 501.60: Variations of percentage of total load taken by each pile in the 3×3 group.

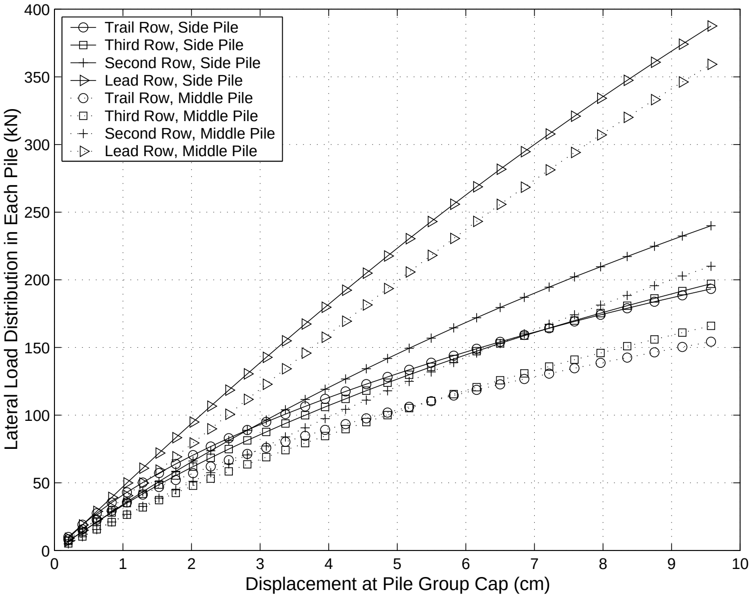


Figure 501.61: Variations of load taken by each pile in the 4 × 3 group.

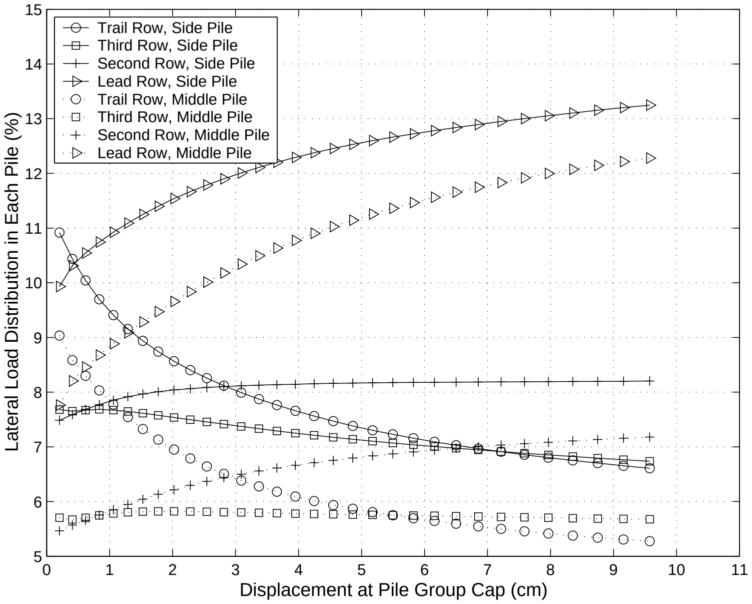


Figure 501.62: Variations of percentage of total load taken by each pile in the 4 × 3 group.

501.4.9 $p-y$ Curve

To further investigate the behavior of each pile, it is worthwhile to examine the p–y behavior of each pile in the group. The fitted moment curve and the resulting displacement from FEM were used to derive the $p-y$ curves according to beam theory. Figures 501.63, 501.64, 501.65 and 501.66 show the derived $p-y$ curves at given depths for all individual piles in the two groups. As is evident from $p-y$ plots for the two groups, the piles in the lead row exhibit much larger resistances than the piles in the middle and trail rows at large lateral displacement, due to the well–known "shadowing effect". Comparison of the $p-y$ curves at different depths in the same pile shows that the lateral resistance p increases as depth increases. For example, at deflection of 4 cm in the 3D (3 pile diameters) spaced 3×3 group, the lateral resistance on the lead-row side pile is 120 kN/m at a depth of -1.54 m, while it is only 90 kN/m at a depth of -0.58 m. This is caused by the increases in vertical stress and Young's modulus as depth increases.

More interestingly, it is seen that the p–y curves of the lead–row and trail–row piles at the depth of -1.54 m are identical at small lateral displacement ($y <$ 0.5 cm). Then the p–y curves of the piles within the trail row soften drastically as lateral displacement increases in the 3×3 group. Similarly for the 4×3 group, both the p–y curves of the lead–row and trail–row piles, and the p–y curves of the third–row and second–row piles at the depth of -2.04 m are almost identical at small lateral displacement ($y <$ 0.7 cm). As deflection increases, the p–y curves of the trail–row and third–row piles soften quickly.

Obviously, each pile in the group exhibits quite different behavior than each other. It is believed that the different behavior of each pile is directly related to the yielding of soil in front of these piles. This observation can be verified by the fact that the softening behavior of the trail-row piles starts at larger lateral displacement as the depth increases, since the plastic zone first develops at the ground surface and then extends downward as deflection increases. These observations also imply that the p–multiplier approach might not be appropriate, especially at small deflection, since it obtains the $p-y$ curves for piles within the group by simply scaling the single–pile $p-y$ curve. In addition, as a comparison with 3D (3 pile diameters) spaced pile group, also plotted in Figures 501.63 and 501.64 is the p–y curve of a lead–row side pile from a 6D spaced pile group. It is apparent that the lateral resistance increases with the increase of spacing.

501.4.10 Comparison with the Centrifuge Tests

The pile head displacements for the two pile groups from 3D FEM and centrifuge tests (McVay et al. McVay et al. (1998)) were plotted against pile head load in Figures 501.67 and 501.68. It can be seen that they agree with each other fairly well at the small lateral displacement and the FEM model is

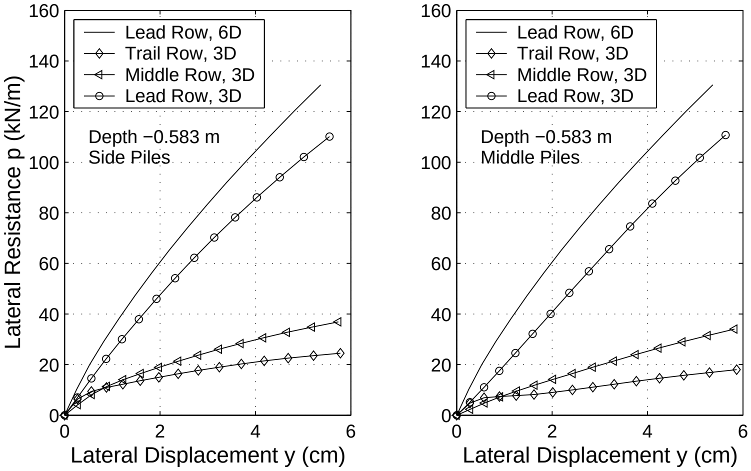


Figure 501.63: p-y curves for each individual pile in the 3×3 group at the depth -0.58m.

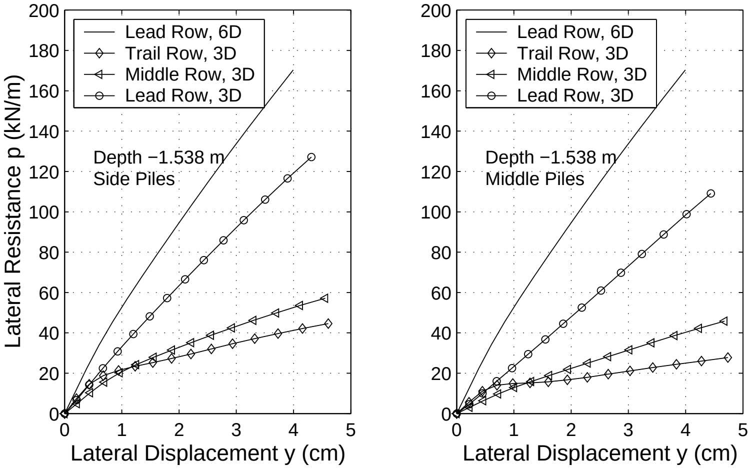


Figure 501.64: p-y curves for individual piles in the 3×3 group at the depth of -1.54m.

relatively stiffer at large lateral displacement.

The accuracy of finite element modeling can also be examined in terms of maximum bending moments. The maximum moment developed in 4×3 group was compared with that from the centrifuge study (Zhang et al. Zhang et al. (1999)) in Figure 501.69. The results from the centrifuge study are slightly larger than that from FEM, which is partially due to the relatively simple elastic-plastic soil model used. Moreover, all the above results can be further improved if the mesh is refined, since the refined pile group mesh will exhibit softer load–displacement response, and therefore develop larger bending moment in individual piles.

The percentage of total lateral load taken by each row from 3D FEM and centrifuge tests (McVay

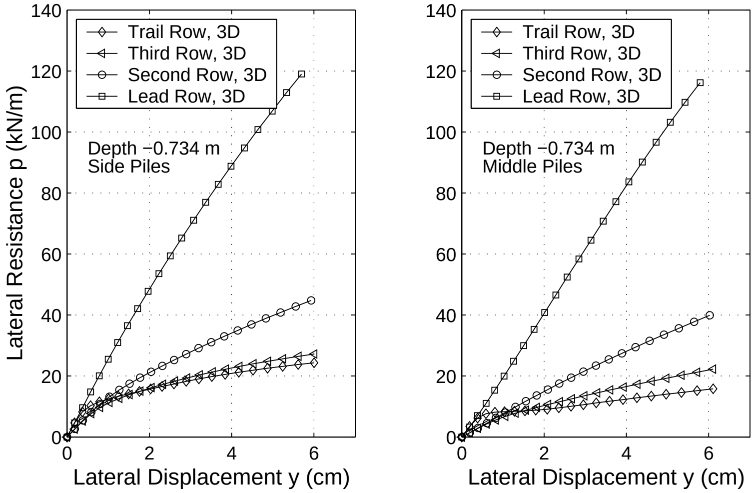


Figure 501.65: p-y curves for individual piles in the 4 × 3 group.

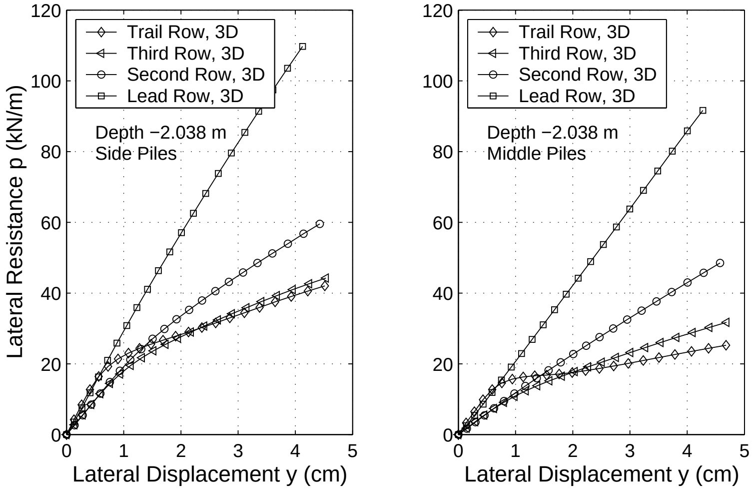


Figure 501.66: p-y curves for individual piles in the 4 × 3 group.

et al. McVay et al. (1998)) at a lateral load of 1650 kN and 2300 kN for the 3 × 3 and 4 × 3 pile groups, respectively, were compared in Figures 501.70 and 501.71. Results for both loose and medium dense sand cases are included. Figures 501.70 and 501.71 show that the density of sand does not have much effects on the load distribution. It is evident that the load distributions to all rows as obtained from FEM and centrifuge tests in all the cases for both pile groups agree very well (the differences are within 3%).

Finally, the variation of distribution of total load to each row as obtained from 3D FEM and centrifuge tests for the 4 × 3 pile group was compared in Figure 501.72. At small lateral displacement, the FEM

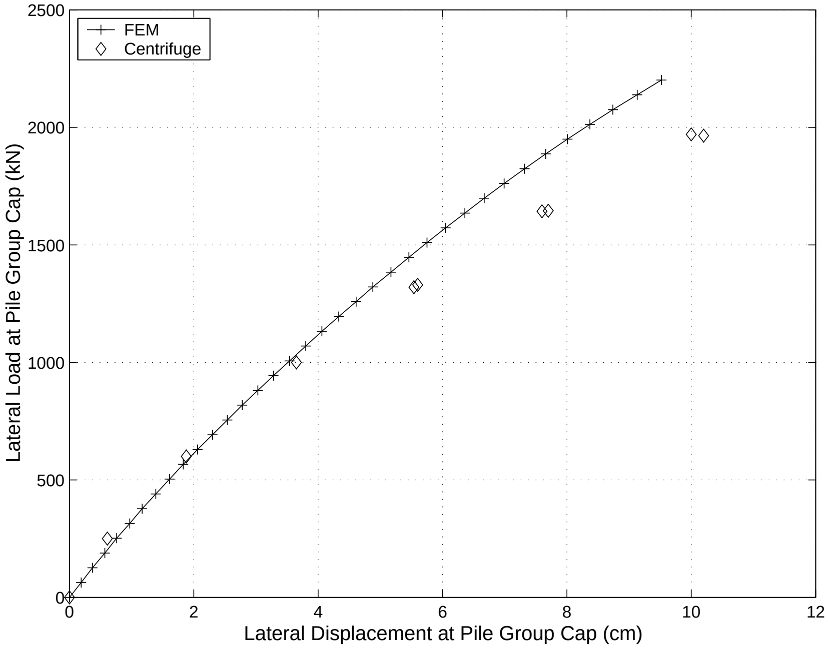


Figure 501.67: Comparison of load displacement response for the 3×3 pile group.

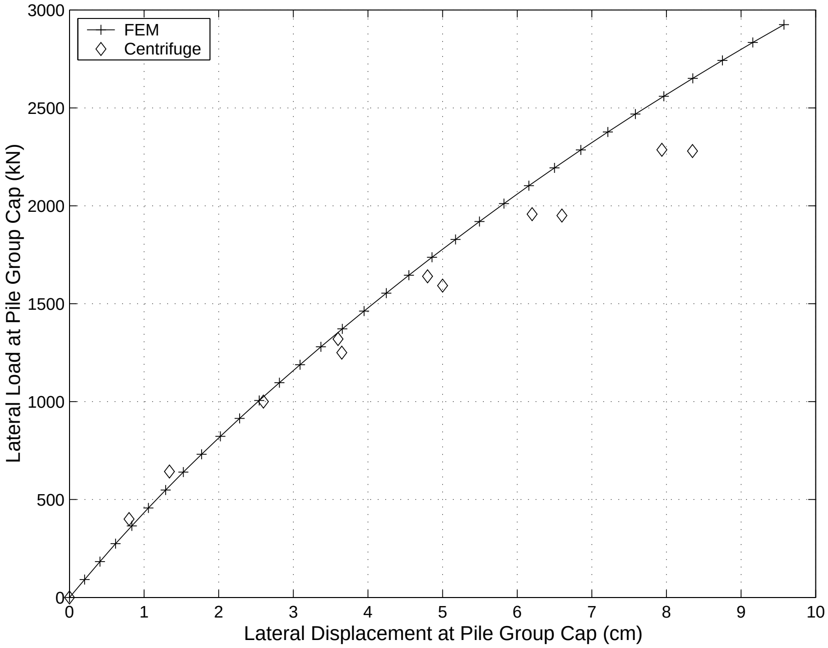


Figure 501.68: Comparison of load displacement response for the 4×3 pile group.

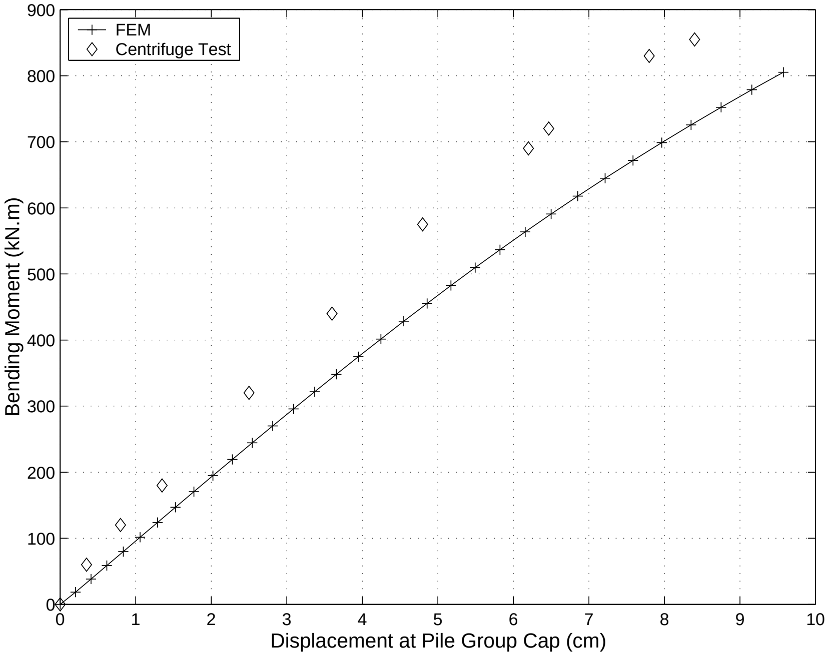


Figure 501.69: Comparison of maximum bending moment response for the 4 × 3 group.

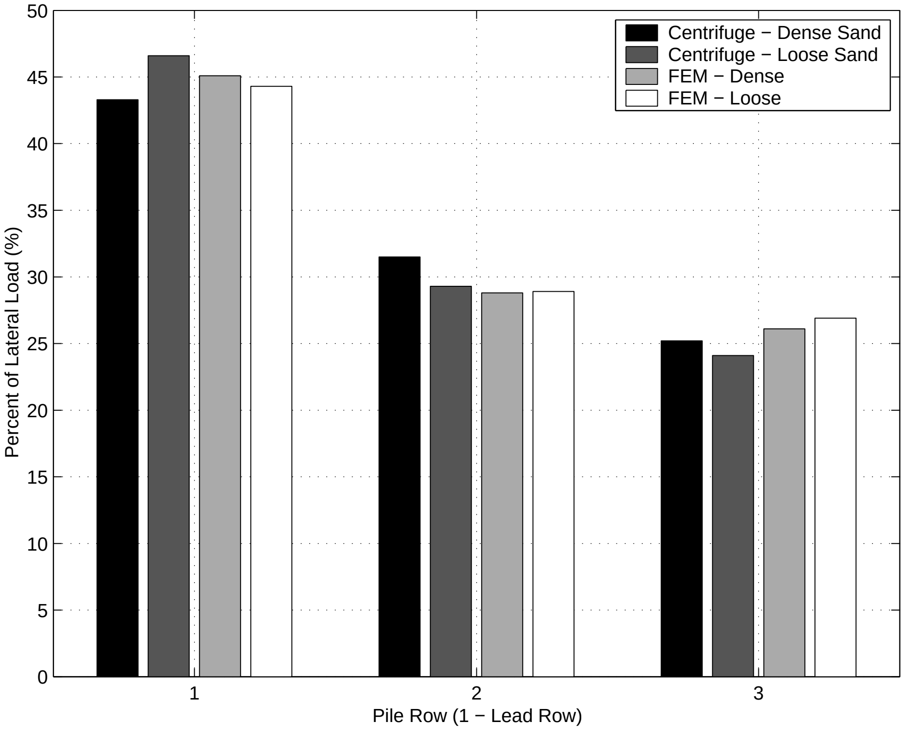


Figure 501.70: Comparison of percentage of total lateral load taken by each row in the 3 × 3 group

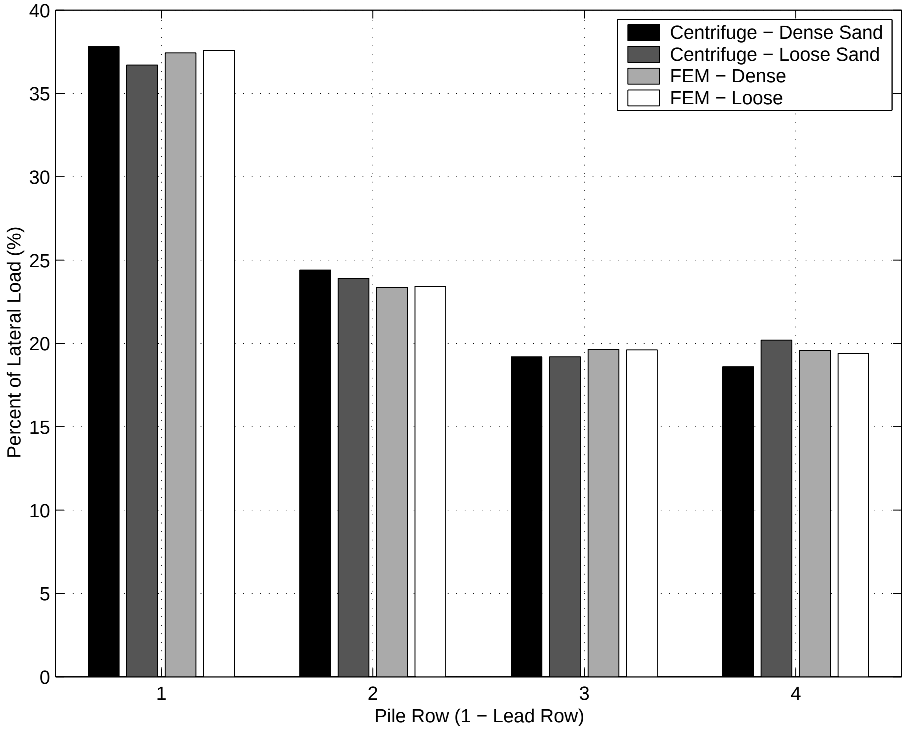


Figure 501.71: Comparison of percentage of total lateral load taken by each row in the 4×3 group

computed percentage of total load acting on the lead row was smaller than measured in the centrifuge tests and the load acting on the trail row was larger than measured in the centrifuge tests. When lateral displacement is beyond 3–4 cm, however, the FEM computed load distribution tends to stabilize and agrees well with that from centrifuge tests.

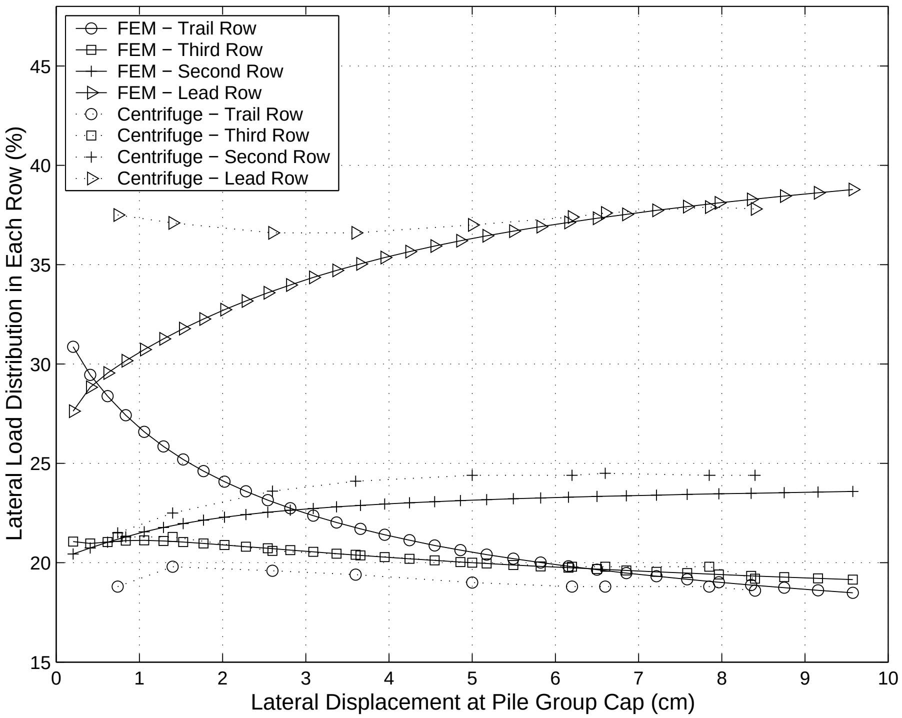


Figure 501.72: Variation of percentage of total load taken by each row in the 4 × 3 pile group.

501.4.11 Conclusions

This paper presents results from the finite element study on the interaction effects of pile groups founded in sands. Specifically the 3×3 and 4×3 pile groups were analyzed in terms of plastic zone, bending moment and load distribution among individual piles. Comparison of results from FEM and centrifuge study shows that elasto–plastic finite element analysis can predict the behavior of pile group with very good accuracy. Particularly, load distribution results from finite element analyses agree very well with that from centrifuge study.

It was shown that not only the load taken by each row in the group is different, but the load shared by individual piles and maximum bending moment developed in individual piles within the same row vary quite a bit, as observed in the centrifuge tests. Although the difference between the loads taken by lead–row middle and side piles is less than 2% of the total load, the difference between the maximum bending moments developed in the lead–row middle and side piles reaches 10–11% in the two pile groups. Interestingly, it was found that bending moment also occurs in the plane perpendicular to the loading direction.

The numerically generated p–y curves were used to study the behavior of each pile in a group. It was found that individual piles in the group exhibit quite different p–y behavior at small deflection, which means that the interaction in elastic range is different than that for loading in plastic range. Therefore, different interaction factors would be more appropriate depending on the loading range.

Since FEM can capture the critical aspects of group effects, it could now be used to systematically study various pile group configurations at much smaller cost than actual load tests, and derive interaction factors for elastic and plastic loading levels that could be used in standard design practice.

501.4.12 Single Pile in Dry Soil Modeling

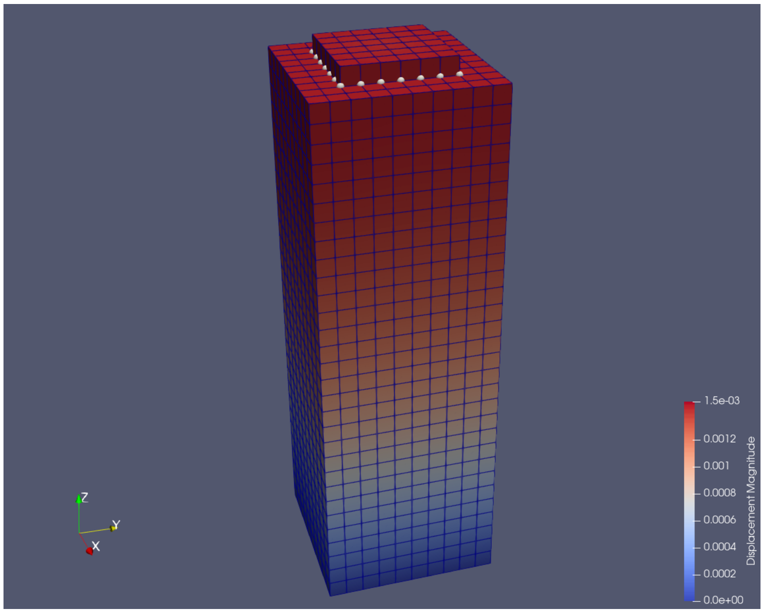


Figure 501.73: Han Yang Pile Model # 1.

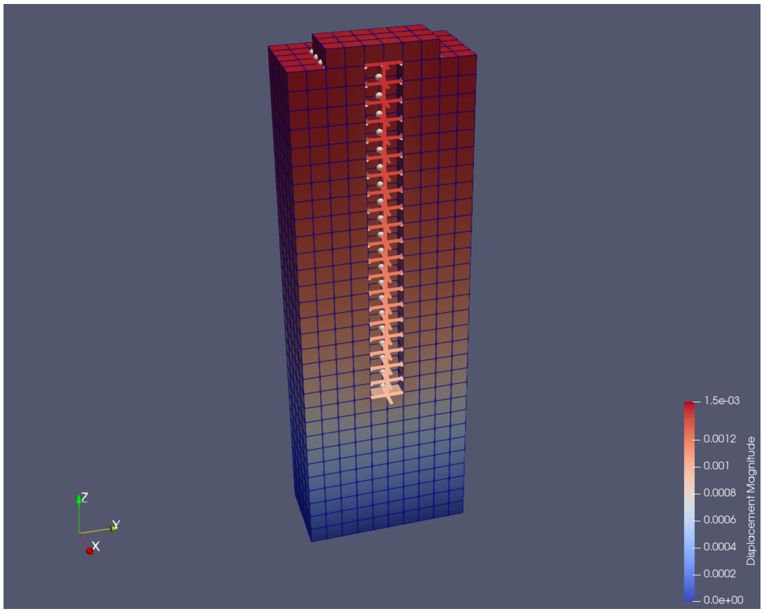


Figure 501.74: Han Yang Pile Model # 2.

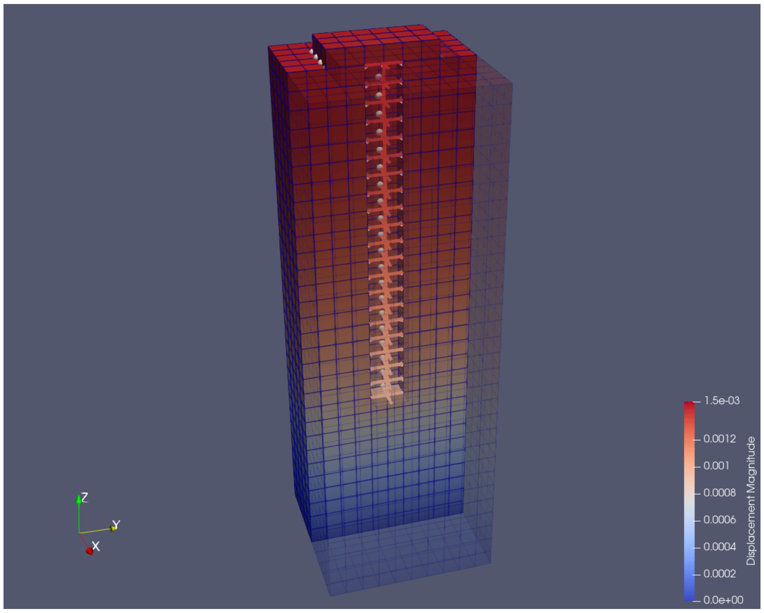


Figure 501.75: Han Yang Pile Model # 3.

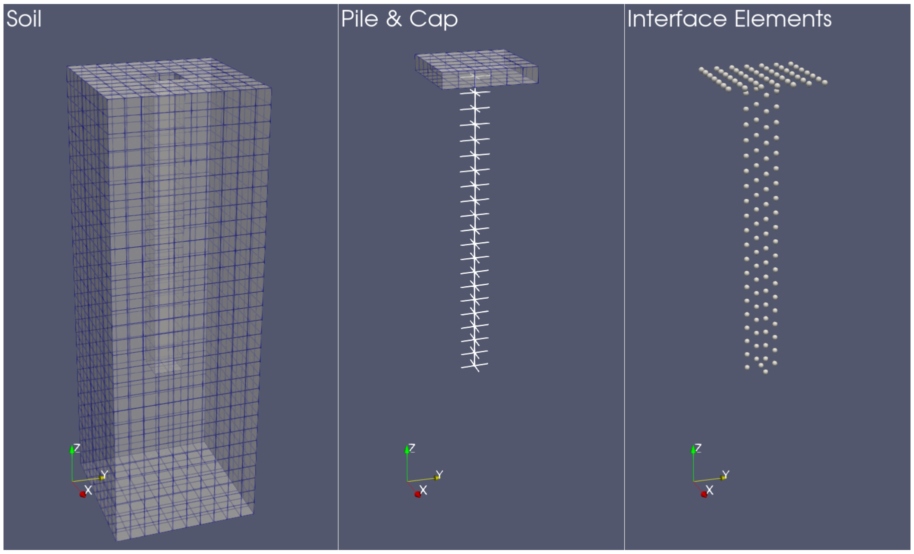


Figure 501.76: Han Yang Pile Model # 4.

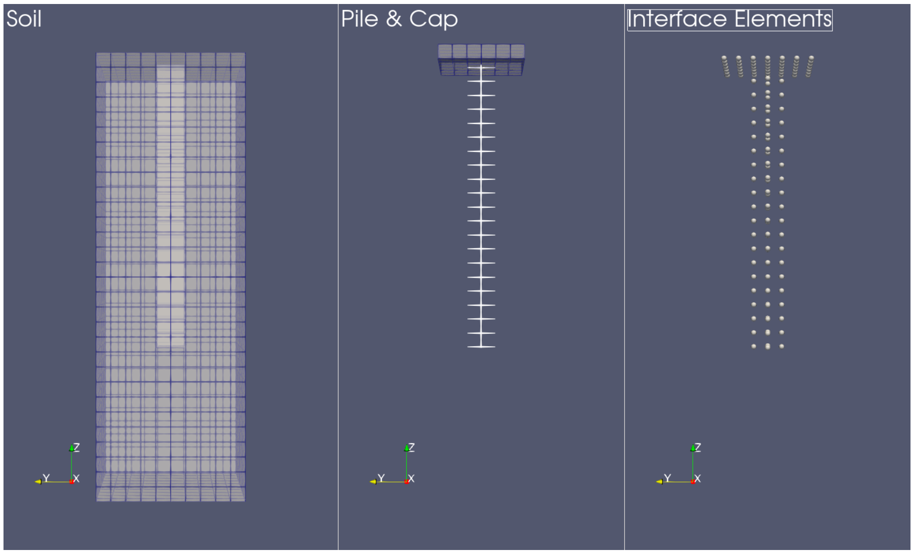


Figure 501.77: Han Yang Pile Model # 5.

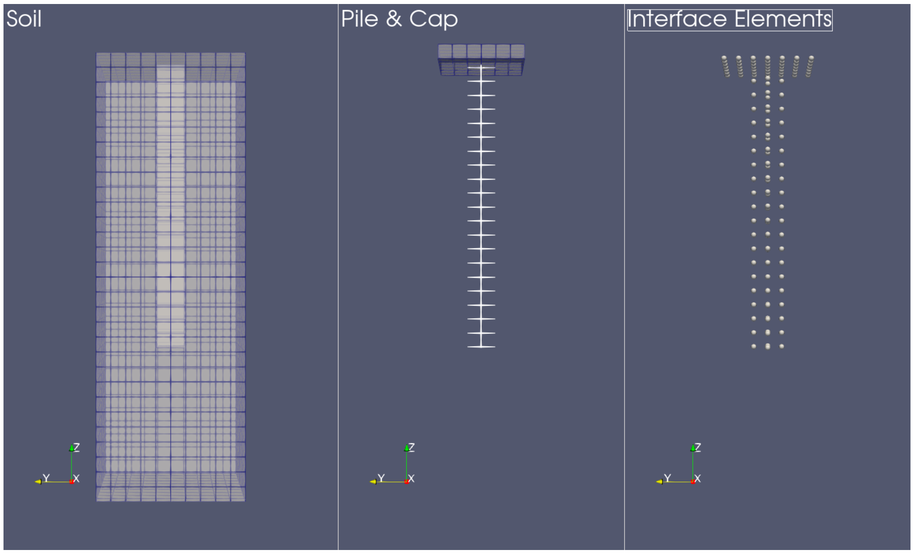


Figure 501.78: Han Yang Pile Model # 5.

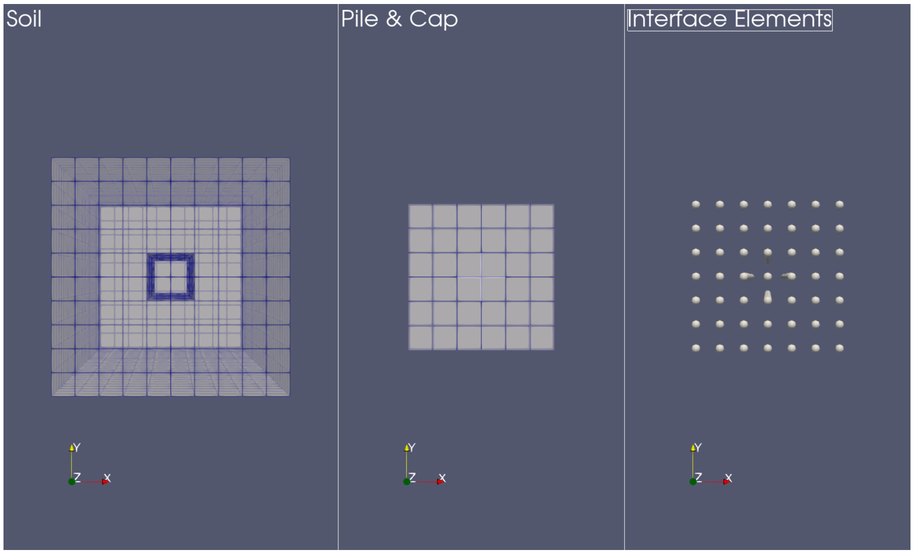


Figure 501.79: Han Yang Pile Model # 6.

Chapter 502

Earthquake Soil Structure Interaction, General Aspects

(1989-2002-2009-2010-2011-2017-2018-2019-2020-2021-)

(In collaboration with Dr. Nima Tafazzoli, Prof. José Abell, Dr. Yuan Feng, Prof. Han Yang, and Dr. Hexiang Wang)

502.1 Chapter Summary and Highlights

502.2 Free Field Ground Motions

Seismic waves propagate from the source (rupturing fault) through the bedrock, surface rock and soil layers to the site of interest where a Nuclear Power Plant is located (or planned). Seismic (compressional and shear) waves do travel through various rock and soil domains, which can be represented by layers, which are sometimes horizontal, but mostly inclined. In addition to that, layers usually have variable thicknesses, creating a complex underground picture of soil/rock domains of different stiffness, mass, energy dissipating characteristics (damping). Most commonly, the stiffness of layers (horizontal and/or inclined) increases with depth. This change in stiffness results in seismic wave refraction, as shown in Figure 502.1.

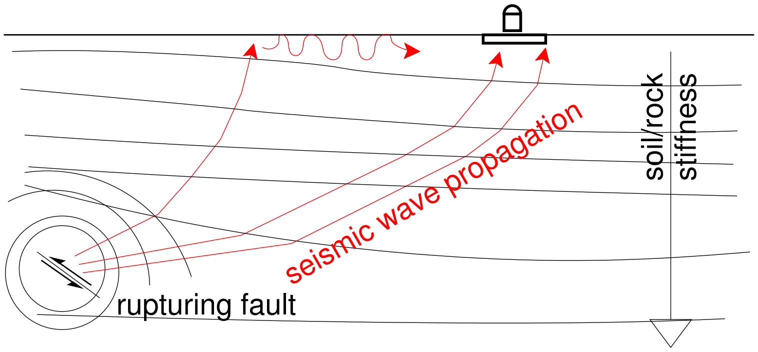


Figure 502.1: Propagation of seismic waves in nearly horizontal local geology, with stiffness of soil/rock layers increasing with depth, and refraction of waves toward the vertical direction.

When soil/rock layers are mostly horizontal, inter-layer refraction causes seismic waves (both P and S) to tend toward vertical propagation. This case is shown in Figure 502.1. However, (near) horizontal layering of geologic layers is not very frequent, unless young deposits of soil in river valleys are of concern. More often the soil/rock layers are inclined, thus creating conditions for variable directivity of seismic motions due to refraction. Figure 502.2 shows one such case where inclined soil/rock layers contribute to mainly horizontal propagation of seismic motions close to the surface, in the vicinity of an NPP.

A general conclusion can be made that seismic waves arriving at the particular site (surface) will be fully three dimensional (3C), uncorrelated, and incoming at an oblique angle. As noted by Zerva (2009), seismic motions will feature lack of correlations of motions between two monitored points at the surface. Lack of correlation is mainly due to (see more details in section 109.2.8):

- Attenuation effects,

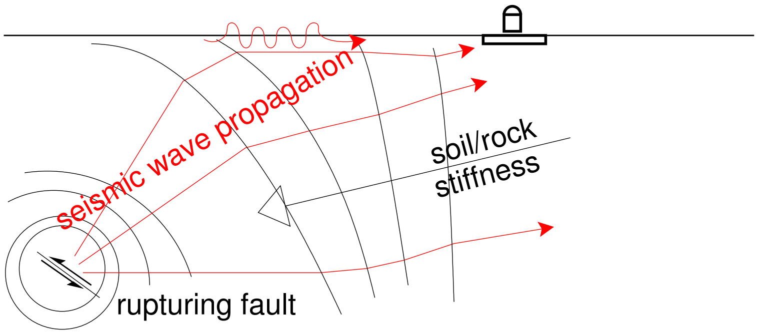


Figure 502.2: Propagation of seismic waves in inclined (close to vertical) local geology, with stiffness of soil/rock layers increasing through geologic layers, and refraction of waves away from the vertical direction.

- Wave passage effects,
- Scattering effects,
- Extended source effects

Due to all the above mentioned 3C effects, realistic modeling of seismic wave propagation can only properly be done by fully and realistically incorporating:

- Body waves (P and S),
- Surface waves (Love and Rayleigh),
- seismic waves coming at an oblique angle (which will actually be done when the above two types of waves are properly modeled),
- Full three dimensional (3C) wave field,
- Lack of correlation (incoherence) effects,
- Energy dissipation effects (damping).

502.2.1 Seismic Motions: Available Data

There exist a large number of recorded earthquake motions. Most records feature data in three perpendicular directions, East-West (E-W), North-South (N-S) and Up-Down (U-D). Number of recorded strong motions, is (much) smaller. A number of strong motion databases (publicly available) exist, mainly in the east and south of Asia, west cost of north and south America, and Europe. There are regions of world that are not well covered with recording stations. These same regions are also seismically fairly inactive. However, in some of those regions, return periods of (large) earthquakes are long, and recording of even small events would greatly help gain knowledge about tectonic activity and geology.

Ergodic Assumption. Development of models for predicting seismic motions based on empirical evidence (recorded motions) rely on Ergodic assumption. Ergodic assumption allows statistical data (earthquake recordings) obtained at one (or few) worldwide location(s), over a long period of time, to be used at other specific locations at certain times. This assumption allows for exchange of average of process parameters over statistical ensemble (many realizations, as in many recordings of earthquakes) is the same as an average of process parameters over time.

While ergodic assumptions is frequently used, there are issues that need to be addressed when it is applied to earthquake motion records. For example, earthquake records from different geological settings are used to develop GMP equations for specific geologic settings (again, different from those where recordings were made) at the location of interest.

3C (6C) versus 1C Records/Motions. Recordings of earthquakes around the world show that earthquakes are almost always featuring all three components (E-W, N-S, U-D). There are very few known recorded events where one of the components was not present or is present in much smaller magnitude. Presence of two horizontal components (E-W, N-S) of similar amplitude and appearing at about the same time is quite expectable. The four cardinal directions (North, East, South and West) which humans use to orient recorded motions have little to do with the earthquakes mechanics. The third direction, Up-Down is different. Presence of the vertical motions before main horizontal motions appear signify arrival of Primary (P) waves (hence the name). In addition, presence of vertical motions at about the same time when horizontal motions appear, signifies Rayleigh surface waves. On the other hand, lack (or very reduced amplitude) of vertical motions at about the same time when horizontal motions are present signifies that Rayleigh surface waves are not present. This is a very rare event, that the combination of source, path and local site conditions produce a plane shear (S) waves that surfaces (almost) vertically. One such example (again, very rare) is a recording LSST07 from Lotung recording array in Taiwan (Tseng et al., 1991). Figure 502.3 shows three directional recording of earthquake LSST07 that occurred on

May 20th, 1986, at the SMART-1 Array at Lotung, Taiwan.

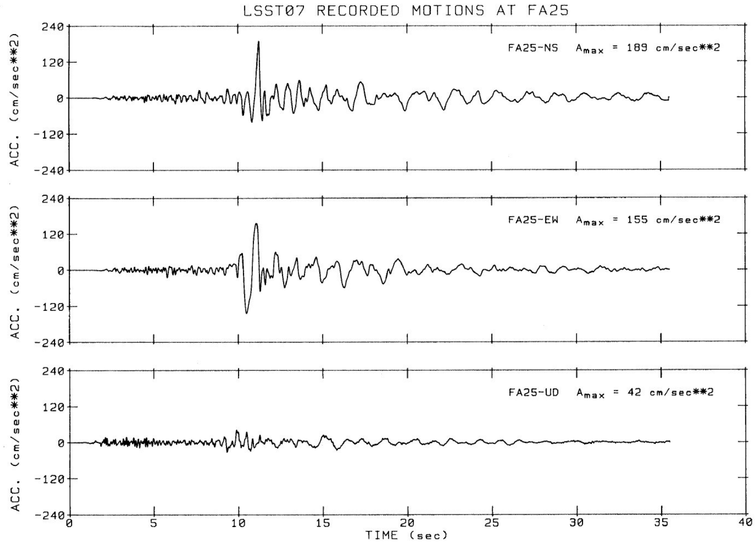


Figure 502.3: Acceleration time history LSST07 recorded at SMART-1 Array at Lotung, Taiwan, on May 20th, 1986. This recording was at location FA25. Note the (almost complete) absence of vertical motions. signifying absence of Rayleigh waves. Figure from Tseng et al. (1991).

Note almost complete lack of vertical motions at around the time of occurrence of two components of horizontal motions, signifies absence of Rayleigh surface waves. In other words, a plane shear wave front was propagating vertically and surfaced as a plane shear wave front. Other recordings, at locations FA15 and FA35 for event LSST07 reveal almost identical earthquake shear wave front surfacing at the same time (Tseng et al., 1991).

On the other hand, recording at the very same location, for a different earthquake (different source, different path) (LSST12, occurring on July 30th 1986) revels quit different wave field at the surface, as shown in Figure 502.4.

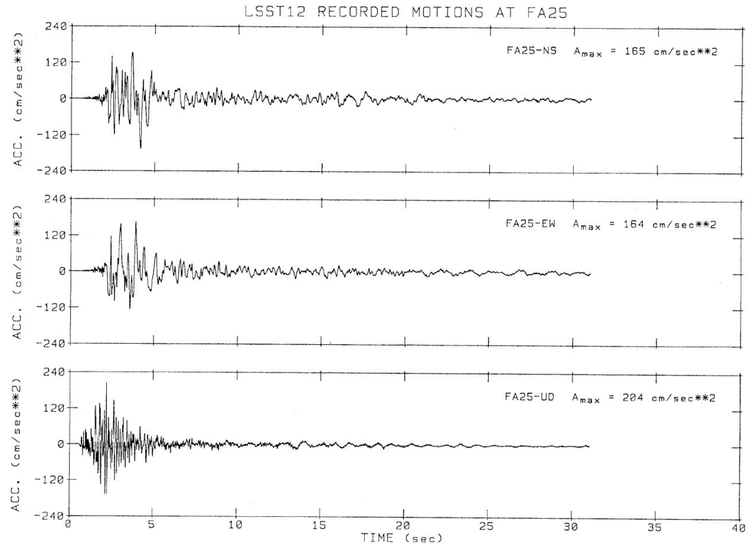


Figure 502.4: Acceleration time history LSST12 recorded at SMART-1 Array at Lotung, Taiwan, on July 30th, 1986. This recording was at location FA25. Figure from Tseng et al. (1991).

502.2.2 Multi-Directional and Seismic Input Coming in at Inclined Angle

Both multi-directional and seismic input coming at an inclined angle is possible by using (and in fact it is inherent to) the DRM, described in previous section. It is important to note that both seismic body waves (P and S) and seismic surface waves (Rayleigh and Love) are present in all situations, and do contribute to multi-directional and inclined seismic input.

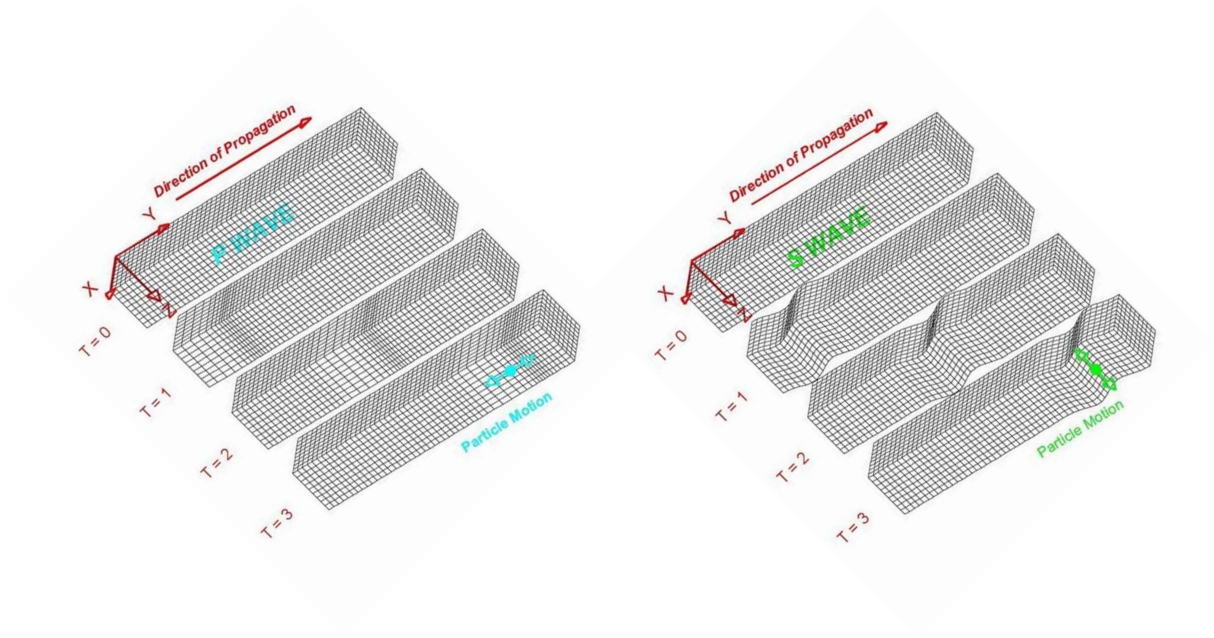


Figure 502.5: Illustration of seismic body waves, namely the P (primary) and S (secondary) waves (illustration from MTU web site).

Importance of surface waves. It has long been recognized that surface waves are responsible for majority of destruction and the seismic energy at some distance from the epicenter during earthquakes (Kramer (1996a), Semblat and Pecker (2009)). In a perfectly linear elastic half-space, Rayleigh waves become significant source of seismic motions at distance $R = h/(\sqrt{(v_p/v_R)^2 - 1})$ where h is hypocenter depth (for uniform half-space) and v_p is the compressional (primary) wave velocity, while v_R is the Rayleigh wave velocity (Kramer (1996a)). For example, if $v_R \approx 0.93 v_S = 0.93 \times 0.6 v_p = 0.56 v_p$ it follows that $R \approx 1.5h$ where v_S is the shear wave velocity (Semblat and Pecker (2009)). Of course for a case with realistic geology, the pattern of occurrence of Rayleigh waves is much more complex. However, since Rayleigh waves represent the result of interaction of body waves (P and S) with the free surface, it is safe to conclude that they are always present.

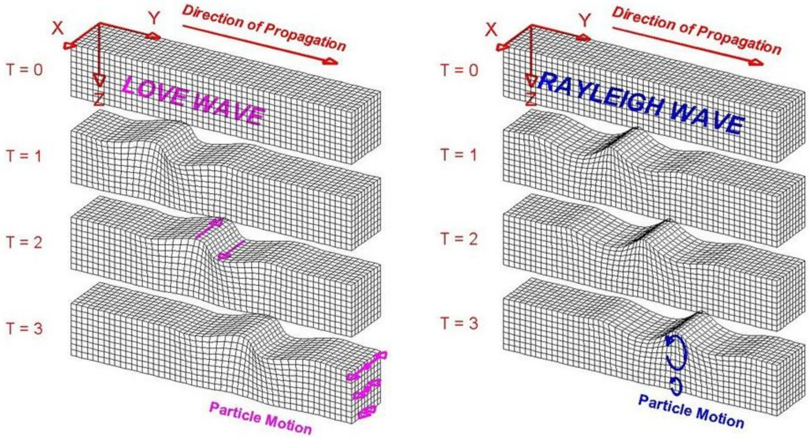


Figure 502.6: Illustration of seismic surface waves, namely the Rayleigh and Love waves (illustration from MTU web site).

502.2.3 Free Field Motion Development

For verification of and for full analysis of ESSI using the DRM, we develop seismic motions using a number of approaches

- Closed form solution,
  - One Component (1C) of three times one components (3 × 1C) using deconvolution (recommended) or convolution (not recommended, due to possible and negative interference of seismic waves at depth, see examples with the SMR)
  - Three Components (3C), plane waves, body and surface (Thomson, 1950; Haskell, 1953)
- Integration equation, Green's functions (frequency – wavenumber method, fk)
- FEM (fault slip model using Real-ESSI)
- Finite Difference models, SW4

502.2.3.1 Details of Free Field Motion Development

Closed Form Solution Analytic solutions used can impose simplified motions (not all components):

- Monochromatic harmonic wave

  - Body wave: P, SV, SH
  - Surface wave: Rayleigh, Love
  - Incident, reflected, transmitted wave components

- Ormsby wavelet
  - Body wave: P, SV, SH
  - Surface wave: Rayleigh, Love
  - Incident, reflected, transmitted wave components

- Ricker wavelet
  - Body wave: P, SV, SH
  - Surface wave: Rayleigh, Love
  - Incident, reflected, transmitted wave components

Definitions for Ormsby and Ricker wavelet are given in chapter 311.2 on page 1786.

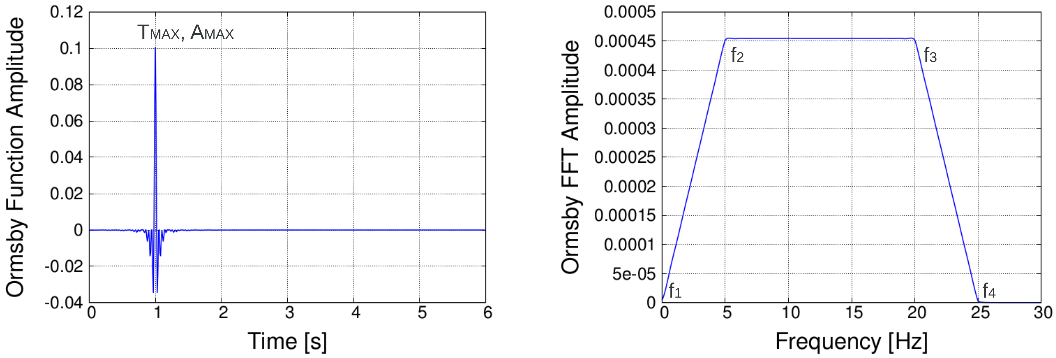


Figure 502.7: Ormsby wavelet.

Closed Form Solution – Ormsby Wavelet?

- Ormsby wavelet is a broad band signal
- Idea: "Shake" the model (or components) with a broad frequency signal to evaluate effects of different frequencies
- Verifying models, propagation of waves of different frequencies

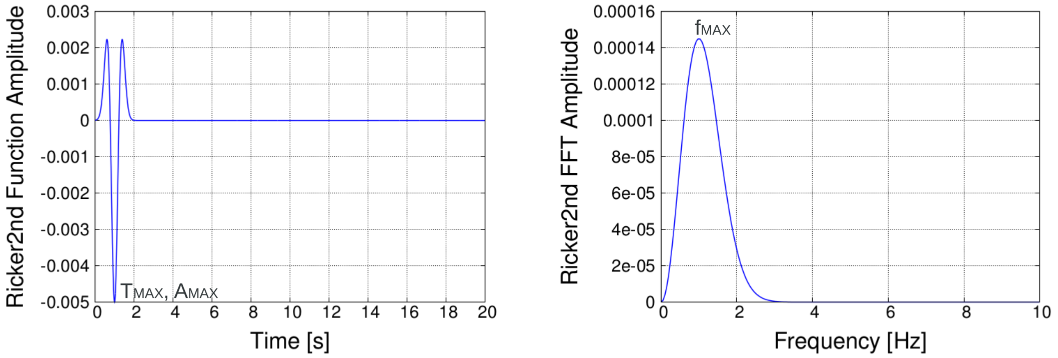


Figure 502.8: Ricker wavelet.

Closed Form Solution – Ricker Wavelet?

- Ricker wavelet is a narrow band signal
- Idea: "Shake" the model (or components) with a narrow frequency signal to evaluate its effects
- Verifying models, propagation of waves of this particular frequency

Frequency-Wavenumber Method (fk method)

- fk method developed originally by Haskell (1964), Wang and Herrmann (1980)
- Based on Green's functions
- Current fk program developed by Zhu and Rivera (2002)
- available at `http://www.eas.slu.edu/People/LZhu`
- fk program is widely used in seismology and geophysics

Fault Slip Model (FEM)

- Develop large FEM model (5km × 10km
- One element (or more) used to initiate the fault rupture
- Fault rupture can be a realistic stress drop, or Ormsby or Ricker wavelets
- Use initial set of motions before the reflections of boundaries interfere (but can use those too!)

FE Model

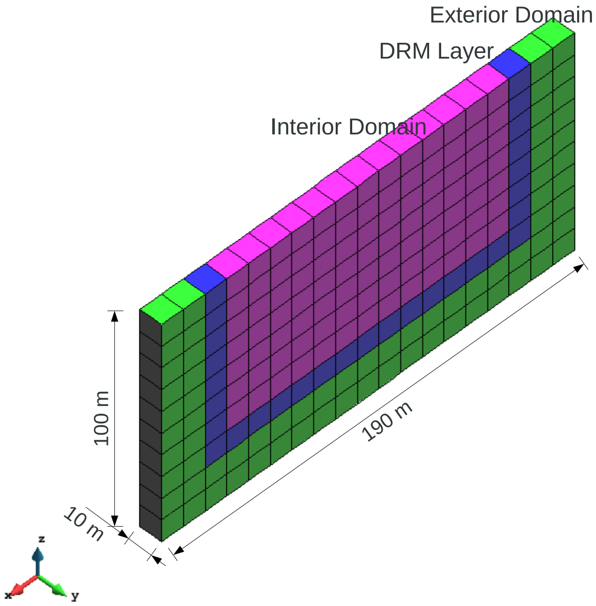


Figure 502.9: 2D model.

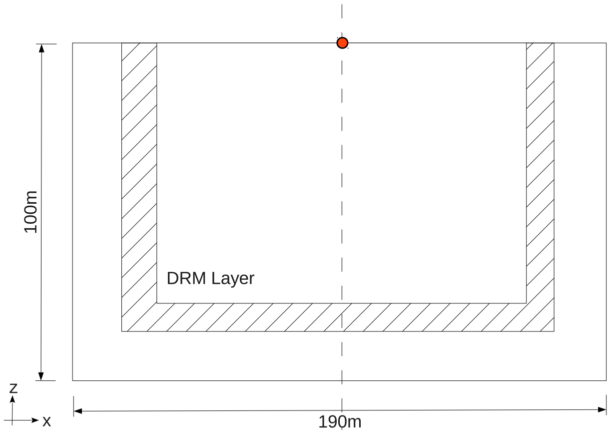


Figure 502.10: 2D model, with observation point location.

FE Model

Properties

- Model Properties
    - 190 m × 100 m × 10 m dimension
    - Vs = 100 m/s

  - Poisson's ratio = 0.3
  - Density = 2000 kg/m^3
- Input Wave Properties
  - Ricker Wavelet (dominant frequency 1 Hz)
  - Body (SH, SV), Surface (Rayleigh)
  - 0°, 30° inclination from the vertical
  - Evaluated closed-form solution (incident, reflected waves)

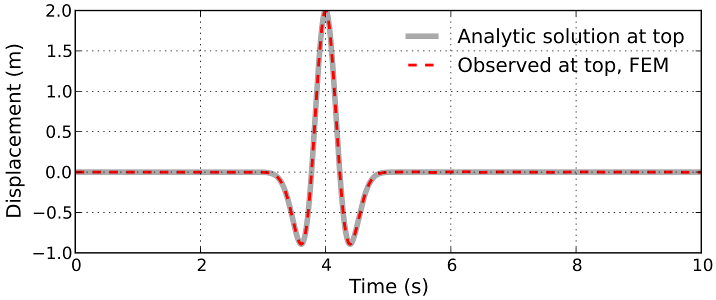


Figure 502.11: Results at the observation point, vertical incident SV wave, horizontal displacements.

SH Ricker Wavelet Input, 0 deg, y Component

502.2.4 Free Field Ground Motions Development: Closed Form Solution

502.2.4.1 Three Component, 3C Motion Development

Thomson (1950), Haskell (1953)

502.2.4.2 One Component, 1C Motion Development

Note on Viscous Damping for 1C Deconvolution and Convolution. Viscous damping used in 1C analytic solution needs to be properly calibrated. For deconvolution, using analytic/closed form solution, propagation of surface record back in time to the depth the viscous damping model used is Voigt damping. Voigt viscous damping features constant damping ratio over all frequencies. For propagation of that same wave back to the surface using FEM solution, Rayleigh viscous damping is used. Rayleigh viscous damping is not constant over frequencies. Rayleigh damping parameters α and β ($C = \alpha * M + beta * K$) have to be calibrated. Calibration of Rayleigh viscous damping parameters α and β has to be done in such a way so that Rayleigh viscous damping can approximate behavior of Voigt viscous damping over main frequency range of seismic motion,

It is suggested that for deconvolution and convolution, propagation of motions upward, damping parameters first be set as zero (0.0) for both de-convolution and FEM propagation. This is done to verify the model and make sure that everything else is correct and that surface free field response can be recovered. This is important in order to make sure that surface motions can be successfully recovered from prescribed motions, through deconvolution and convolution process, that rely on different methods, and without influence of damping.

Only after model is verified with no viscous damping, one should proceed to calibrate viscous damping parameters. The common values that work well for a number of examples, using Real-ESSI, are:

```
a_0 = 10/s; // calibrated parameters
a_1 = 0.06*s; //calibrated parameters

soil_damping = 0.02; // change this value accordingly
add damping # 1 type Rayleigh with a0 = a_0*soil_damping a1 = a_1*soil_damping ↩
    stiffness_to_use = Initial_Stiffness;
```

502.2.5 Free Field Ground Motions Development: Frequency Wave Number Method (Green's functions) (fk)

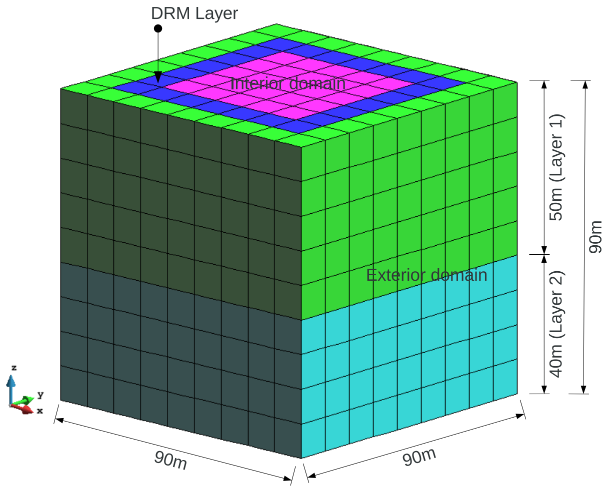


Figure 502.12: 3D model.

FE Model

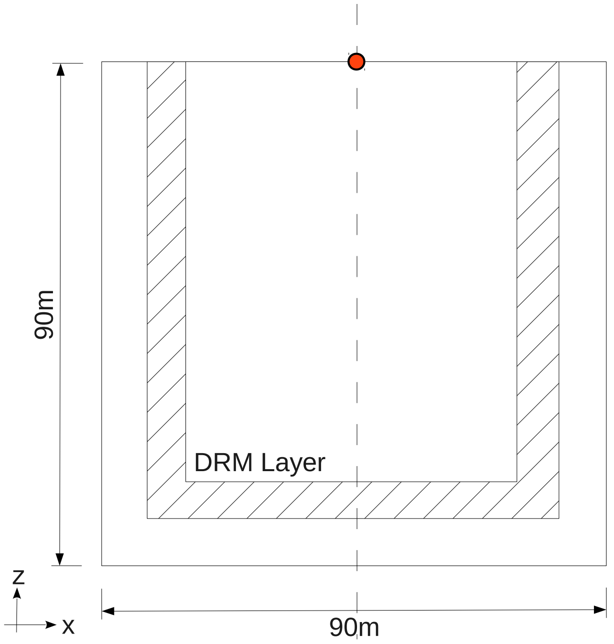


Figure 502.13: 3D model, with observation point location.

FE Model

Properties

- Model Properties
  - 90 m × 90 m × 90 m dimension
  - Vs1 = 300 m/s, Vs2 = 400 m/s
  - Poisson's ratio1 = 0.25, Poisson's ratio2 = 0.25
  - Density1 = 940 kg/m^3, Density2 = 990 kg/m^3
- Input Wave Properties
  - Generated using fk program
  - Variables are chosen to simulate Northridge Earthquake

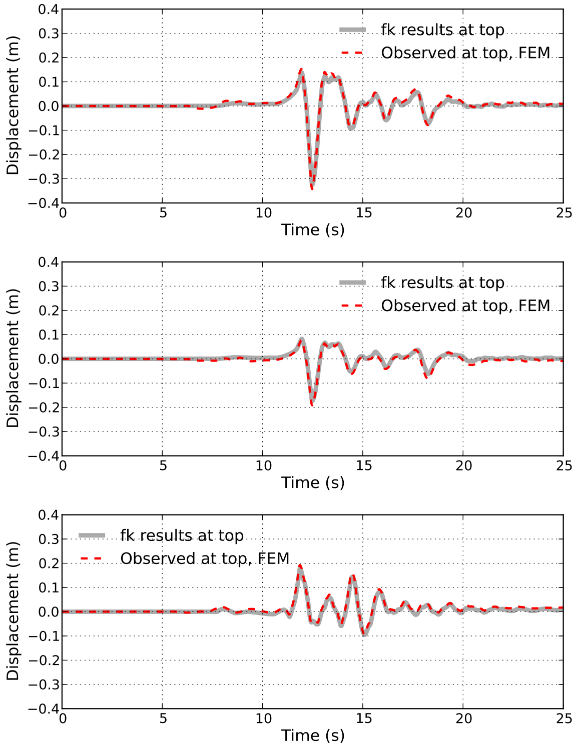


Figure 502.14: Observation point displacements (top - EW, middle - NS, bottom UD) comparison between *fk* motions and DRM motions (that were developed using *fk* motions as input.

FEM Results, EW, NS, UD Components

Artificial 1D Downhole Array

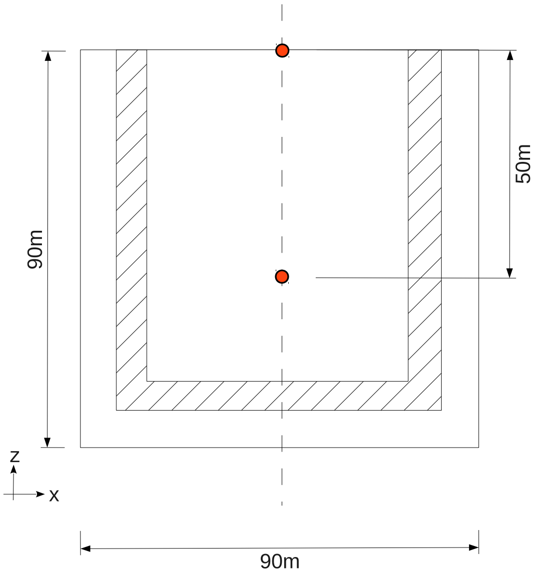


Figure 502.15: Artificial 1D Downhole Array

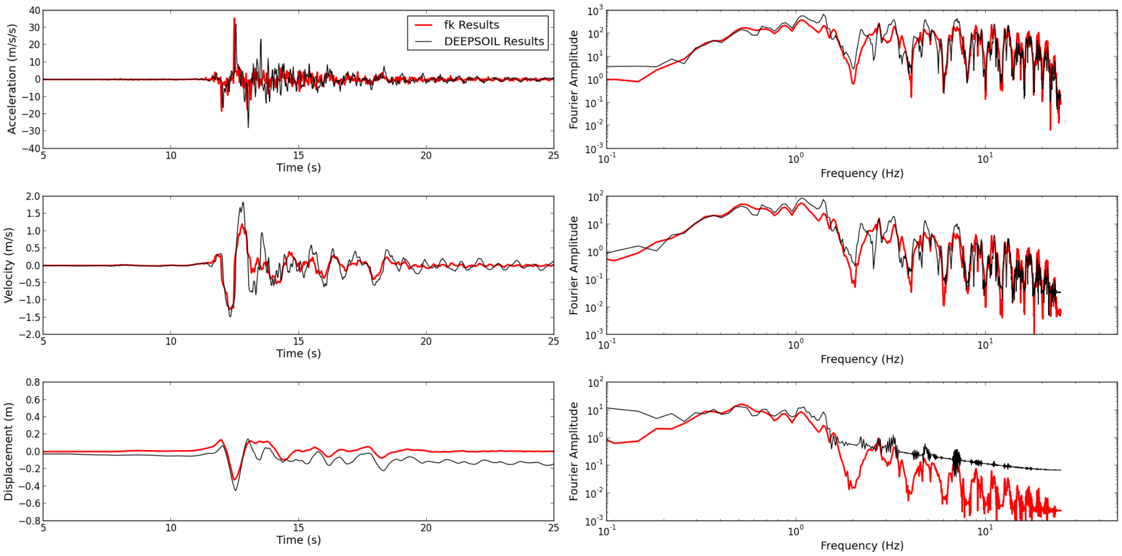


Figure 502.16: EW Component, Station Depth = 0 m

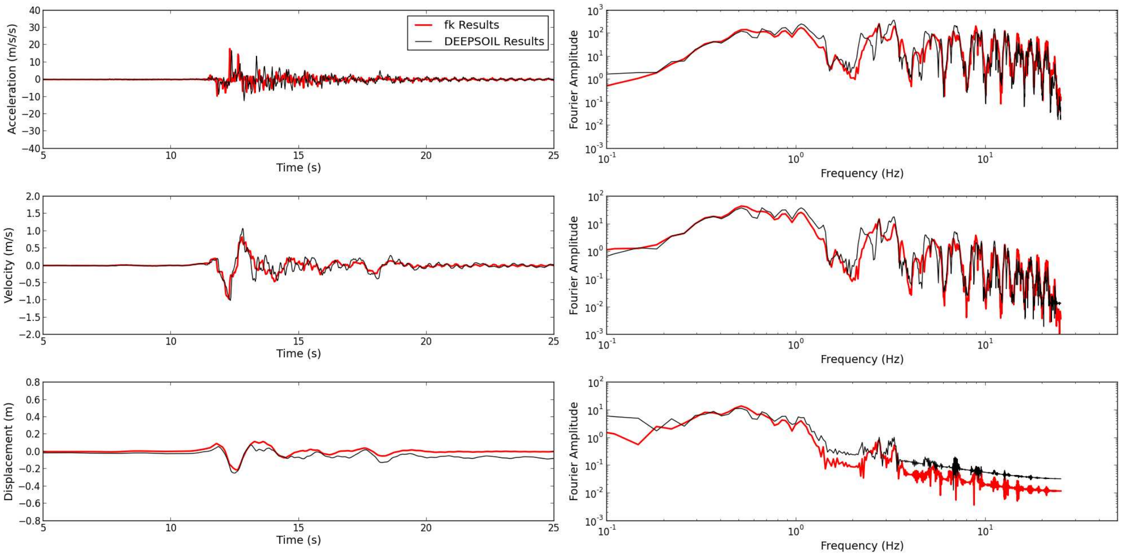


Figure 502.17: Component, Station Depth = 50 m

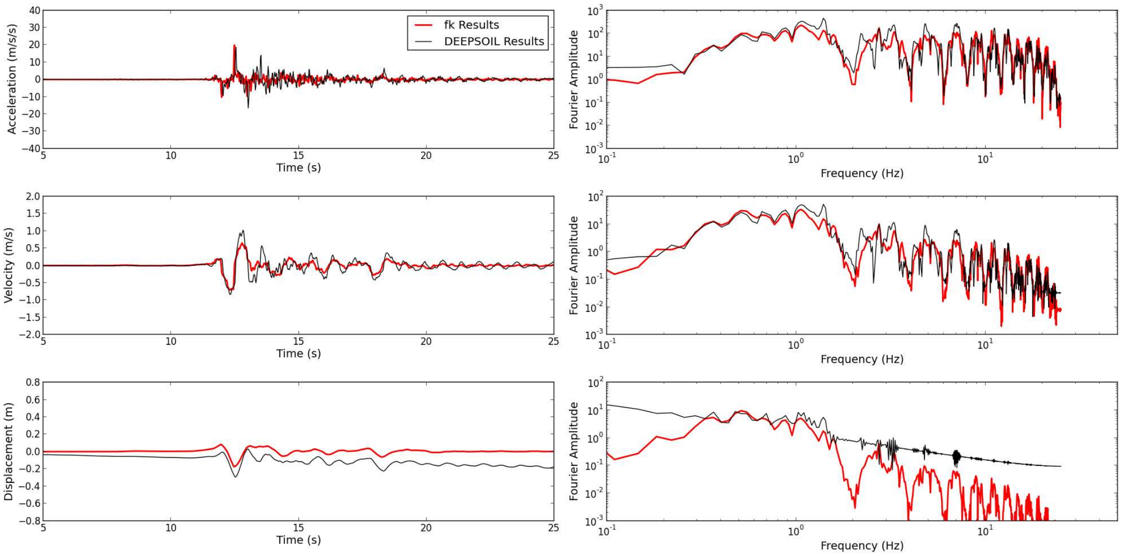


Figure 502.18: Component, Station Depth = 0 m

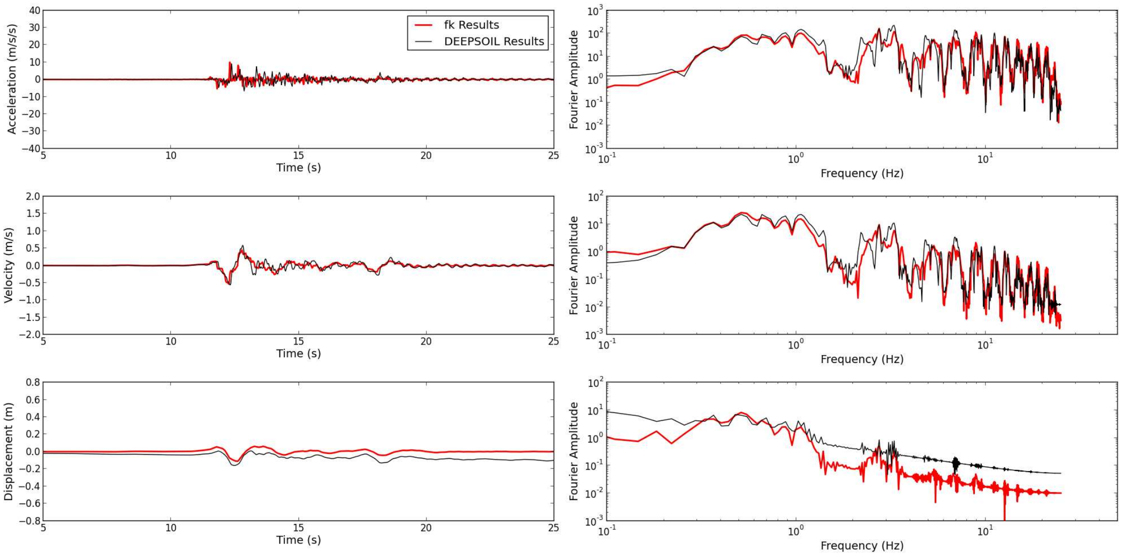


Figure 502.19: Component, Station Depth = 50 m

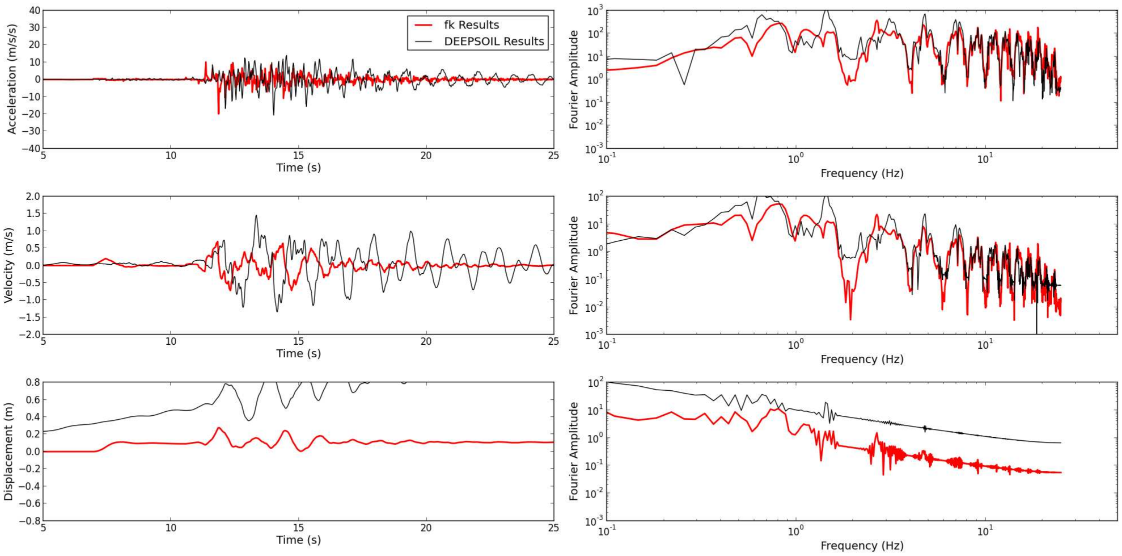


Figure 502.20: Component, Station Depth = 0 m

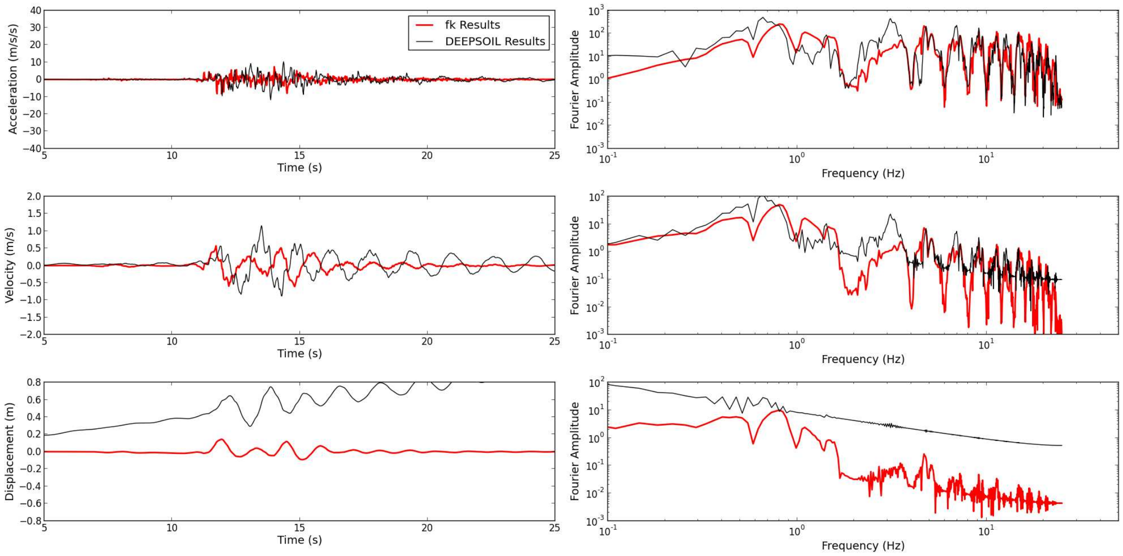


Figure 502.21: Component, Station Depth = 50 m

502.2.6 Free Field Ground Motions Development: Fault Slip Model

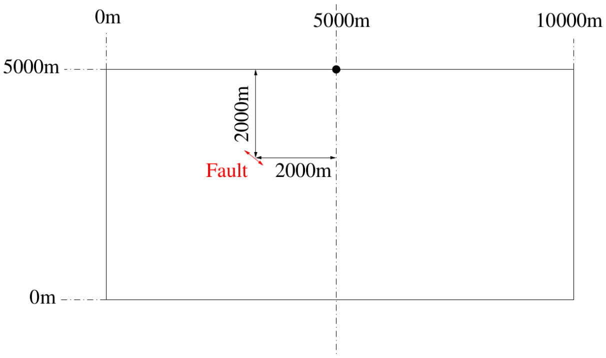


Figure 502.22: Fault Slip FE Model at -2km -2km.

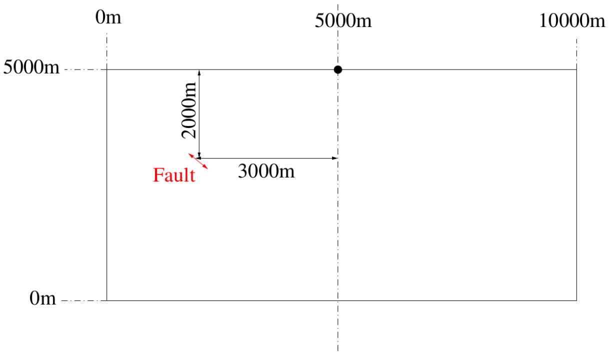


Figure 502.23: Fault Slip FE Model at -2km -3km.

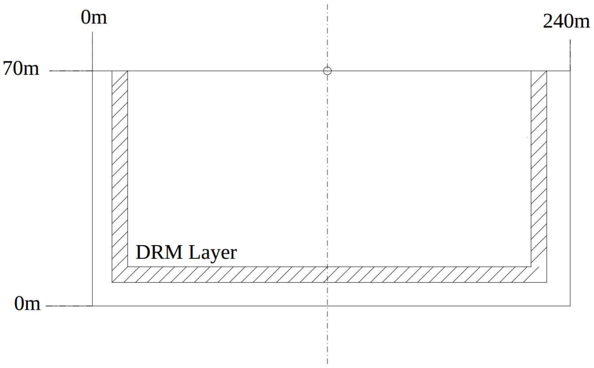


Figure 502.24: Fault Slip DRM Model.

Properties

- Model Properties
  - Vs = 700 m/s
  - Poisson's ratio1 = 0.1
  - Density = 1800 kg/m^3
- Input Wave Properties
  - Generated using 2 different 'Fault Slip Model' (2 km × 2 km, 2 km × 3 km)
  - Ricker Wavelet (dominant frequency of 1 Hz)

Figure 502.25: Fault Slip Model, -2km -2km, (left) X and (right) Z displacements at middle top.

Figure 502.26: Fault Slip Model, -2km -3km, (left) X and (right) Z displacements at middle top.

Plane Wave Model

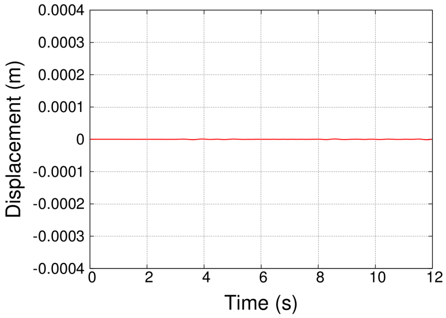


Figure 502.27: Fault Slip Model, -2km -2km, X displacements outside of Γ (DRM) domain.

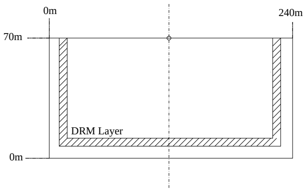


Figure 502.28: Plane wave model.

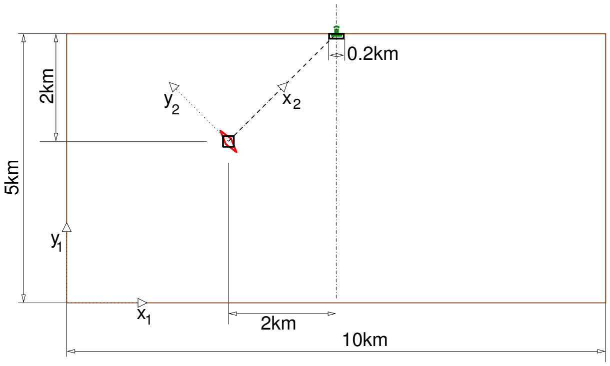


Figure 502.29: 2D fault slip model disposition.

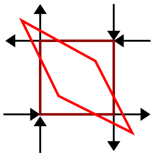


Figure 502.30: Seismic source mechanics. stress drop, Ormsby wavelet.

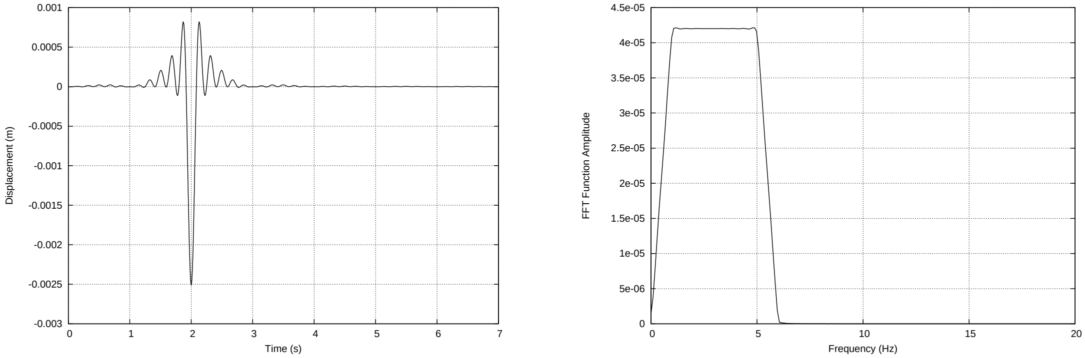


Figure 502.31: (left) Ormsby wavelet displacement and (right) Fourier transform. Used for stress drop.

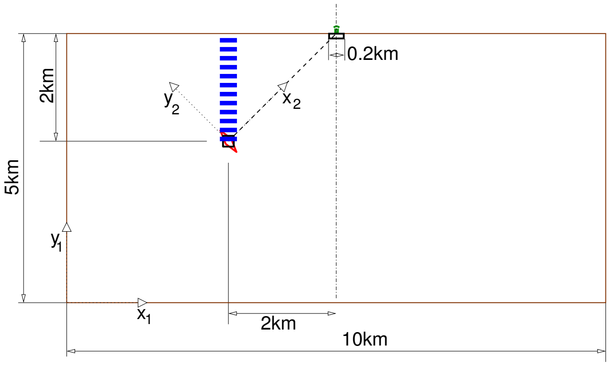


Figure 502.32: Location of a measuring array, vertical, above source.

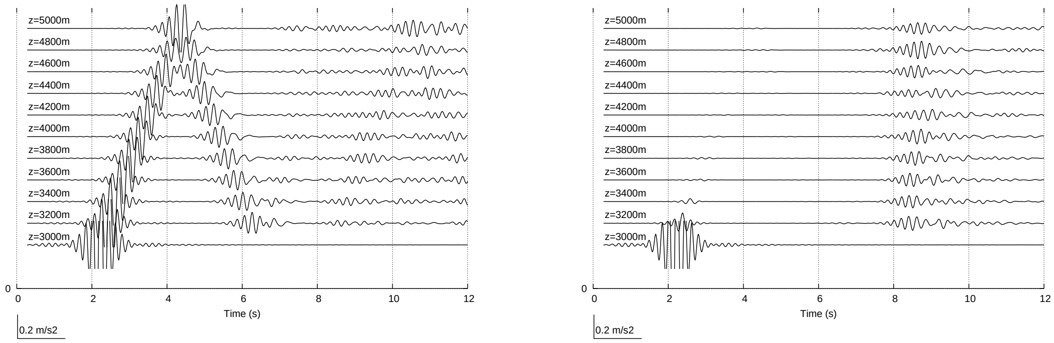


Figure 502.33: Accelerations along the measuring array above source: (left) horizontal, and (right) vertical.

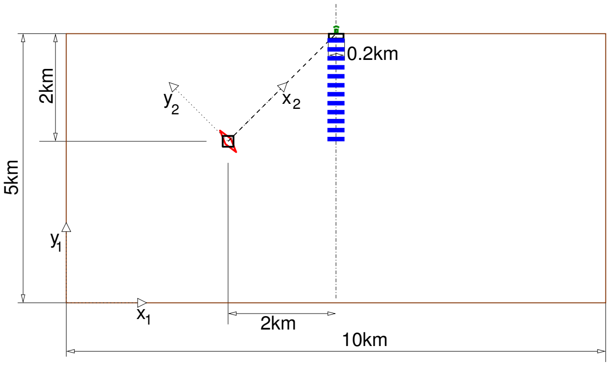


Figure 502.34: Location of a measuring array, vertical, middle.

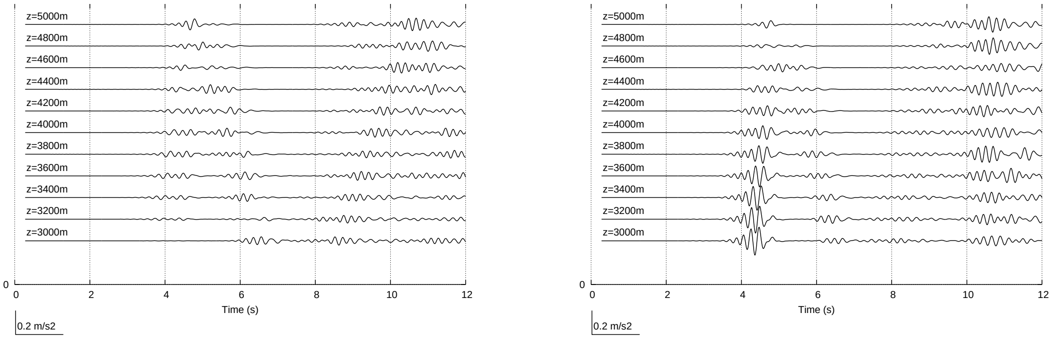


Figure 502.35: Accelerations along the measuring array in the middle: (left) horizontal, and (right) vertical.

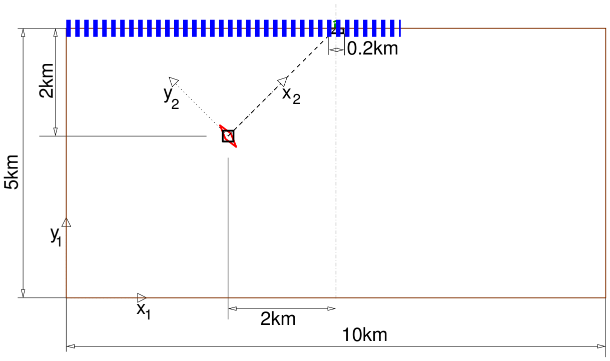


Figure 502.36: Location of a measuring array, horizontal, surface.

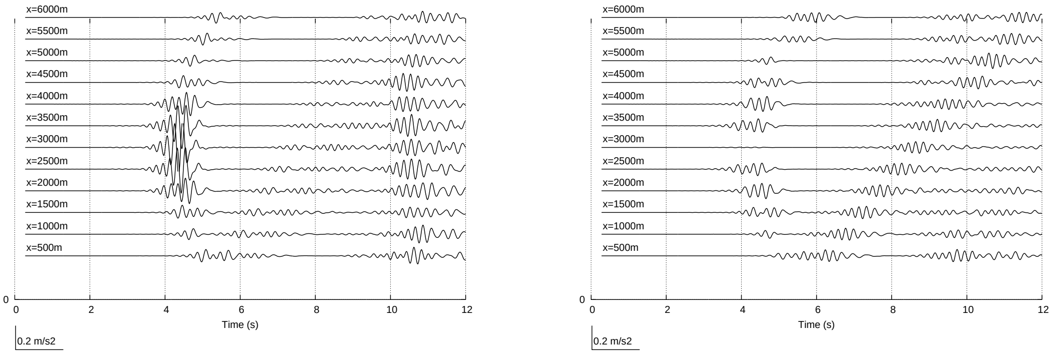


Figure 502.37: Accelerations along the measuring array at the surface: (left) horizontal, and (right) vertical.

502.2.6.1 Input Motion

The point fault (element) is located at X=3000m and Z=3000m as shown in Fig. 502.113 (that is 2000m under and 2000m to the left of surface location of interest). The source is at the angle of 45^{deg}, from the site at the middle top of the model. Pure shear is applied on the fault element in order to generate both S wave and P wave as a double couple source. In the actual analysis, equivalent nodal forces are used, and were obtained from constant surface tractions. Equivalent nodal forces for 27 nodes brick are shown in Fig. 502.38.

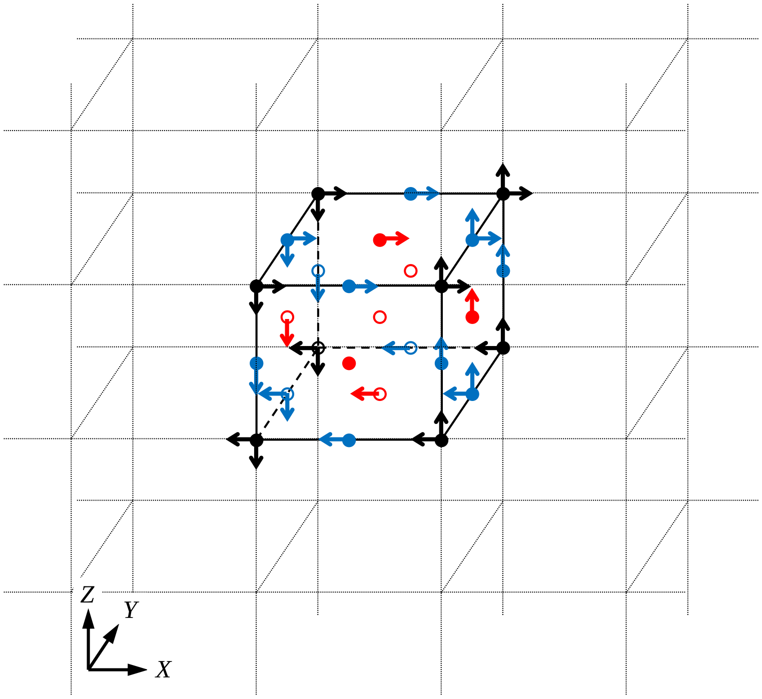


Figure 502.38: Equivalent nodal forces to apply pure shear on 27 node brick element (Loading factors: 1/36 for black, 4/36 for blue, 16/36 for red)

Ormsby wavelet is used as the time history of the shear force amplitude. The time history and its Fourier amplitude are shown in Figs. 502.39 and 502.40.

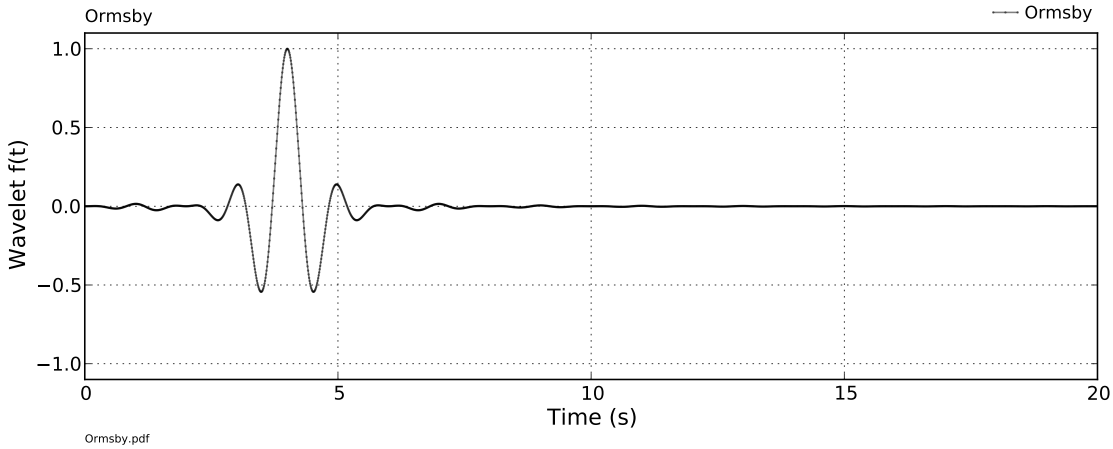


Figure 502.39: Time history of the Ormsby wavelet ($f_1 = 0$ [Hz], $f_2 = 1$ [Hz], $f_3 = 1$ [Hz], $f_4 = 1.5$ [Hz])

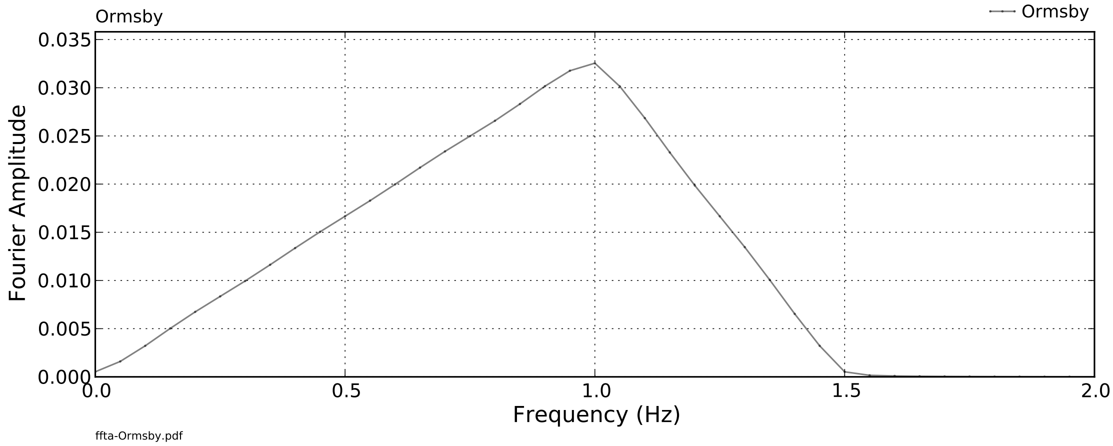


Figure 502.40: Fourier amplitude of the Ormsby wavelet ($f_1 = 0$ [Hz], $f_2 = 1$ [Hz], $f_3 = 1$ [Hz], $f_4 = 1.5$ [Hz])

502.2.6.2 Select Seismic Motions, Displacement Array Traces

The horizontal and vertical displacements, and particle motions in XZ plane at the site are observed in Fig. 502.41. Particle motions of every 1.2 sec are shown in the figure as well. Various color schemes were used for time section, to aid visualization of particle motions.

The point seismic source, as used in this case, a single finite element with cycles of pure shear, creates both primary (P, compressional) and secondary (S, shear). Initially only the P wave reaches the point of interest at the surface in the middle of the model, because of the radiation pattern and the site location (directly, at 45^{deg} away from the point source). On the other hand S wave propagates in the horizontal and vertical directions most strongly while there are traces of this wave away from these main propagation directions. With the P wave velocity $V_p = 4899 \text{m/s}$ (assumed of isotropic material) the first arrival time is 0.58 sec. Once the body waves hit the surface, surface (Rayleigh) wave is generated and propagates horizontally. Therefore, the obvious P wave particle motion can be seen only for the first 1.2 sec, while the ellipse orbit, which is the typical Rayleigh characteristic, is observed after 1.2 sec.

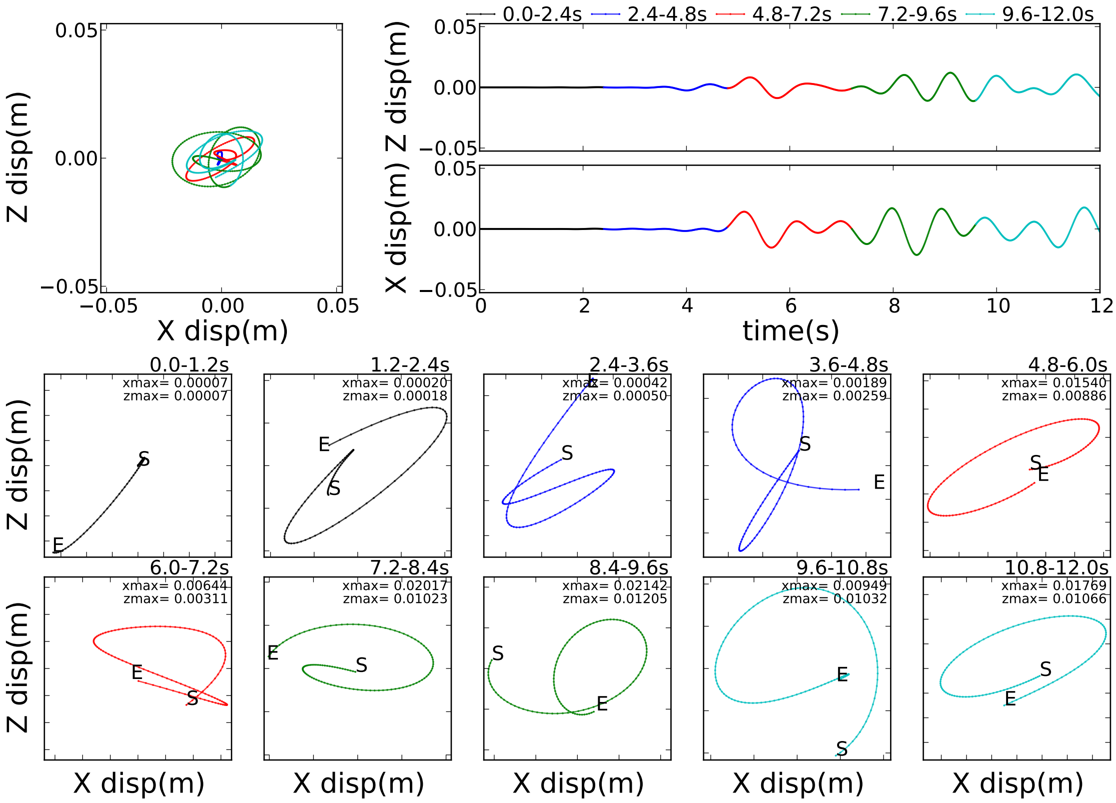


Figure 502.41: Displacement and particle motions at the site. (right top) Displacement time history, (let top) particle motion of the whole analysis, (bottom) particle motions of every 1.2 sec, S: Start, E: End

To understand wave propagation better, displacement time history and particle motions are plotted on five observatory arrays.

- Array1: diagonal array from the fault to the site (Fig. 502.42)
- Array2: horizontal array from the source (Fig. 502.65)
- Array3: vertical array upon the fault (Fig. 502.77)
- Array4: horizontal array on the surface (Fig. 502.89)
- Array5: vertical array below the site (Fig. 502.101)

Array1: Displacement time history and particle motion on Array1 are shown in Fig. 502.43 to Fig. 502.53. Fig. 502.43 and Fig. 502.53. show the resultant motions at the site and the fault respectively. It is expected that the direct wave propagating along this array is only P wave, not S wave, because of the radiation pattern and the homogeneous medium. Nodes moves in both horizontal and vertical direction together with the direct P wave. Dominant P wave can be seen at the beginning of the motion at every single nodes on this array. It is also observed that direct wave amplitude decreases when the observation point is far from the fault, a so called geometrical damping effect. Since direct S wave does not exist in this array, all non-diagonal displacements are caused by reflection wave and Rayleigh wave. Particle motion of Rayleigh wave is observed as ellipse in XZ plane. It is obviously confirmed that the effect of the Rayleigh wave is more significant as the observation point is near the surface. In order to make it easy to distinguish P and S wave propagation along the diagonal array, these resultant waves are converted into radial/transverse coordinate from vertical/horizontal coordinate (RT conversion). Converted waves of Array1 are shown in Fig. 502.54 to Fig. 502.64. It is now easy to observe P wave motion in radial direction and no S wave in Transverse direction.

Array2 and Array3: Results of Array2 are shown in Fig. 502.66 to Fig. 502.76 and results of Array3 are shown in Fig. 502.78 to Fig. 502.88. Since S wave is dominant in vertical and horizontal direction from the source while P wave is dominant in diagonal direction, only horizontal motion and vertical motion are observed well for the first several seconds until reflected wave and surface wave reach on these two arrays. Geometry damping effect is observed as well as P wave propagation in Array1. Rayleigh wave effects are propagating in depth. We can still see Rayleigh wave motion on these arrays, even though the effect of surface wave is less significant than Array1.

Array 4: Results of Array4 are shown in Fig. 502.90 to Fig. 502.100. Array4 is located on the surface and affected by surface wave effects the most.

Array 5: Results of Array5 are shown in Fig. 502.102 to Fig. 502.112. It is sometimes assumed (by others) that wave propagate in vertical direction near the surface in case of horizontally layered ground model. In this case, the incident angle is 45 degree and the vertical wave propagation assumption is obviously not valid.

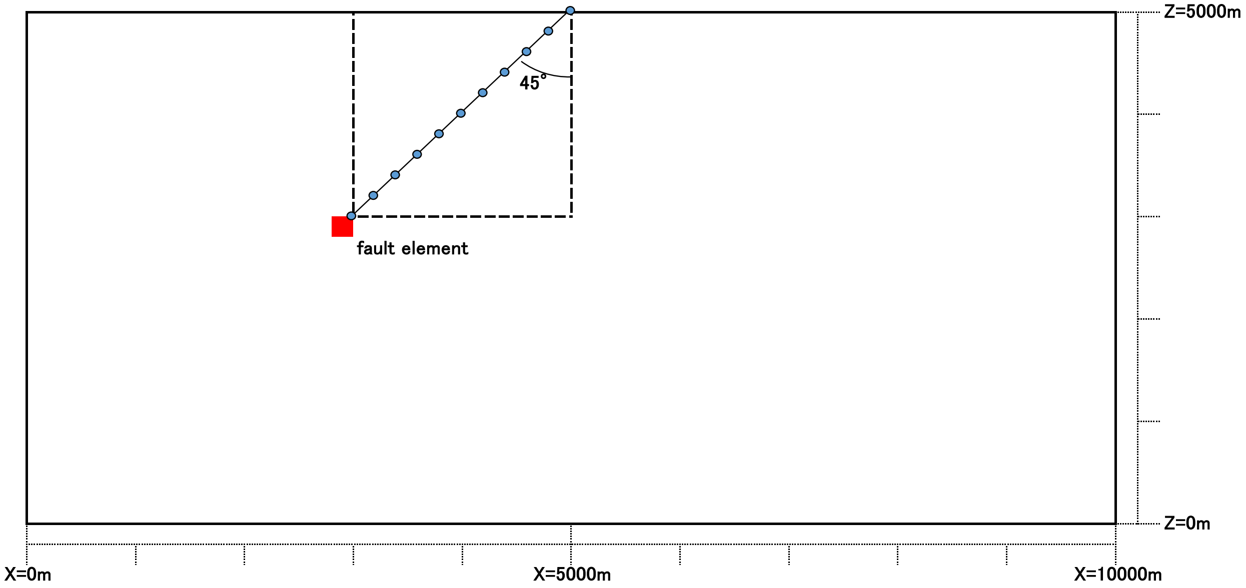


Figure 502.42: Observation array of Array1

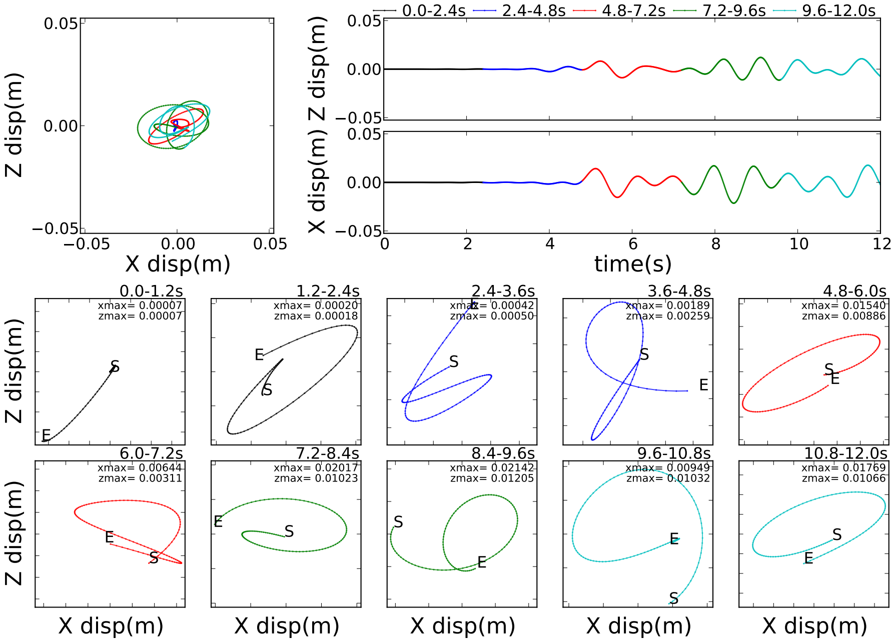


Figure 502.43: diagonal array X=5000m, Z=5000m

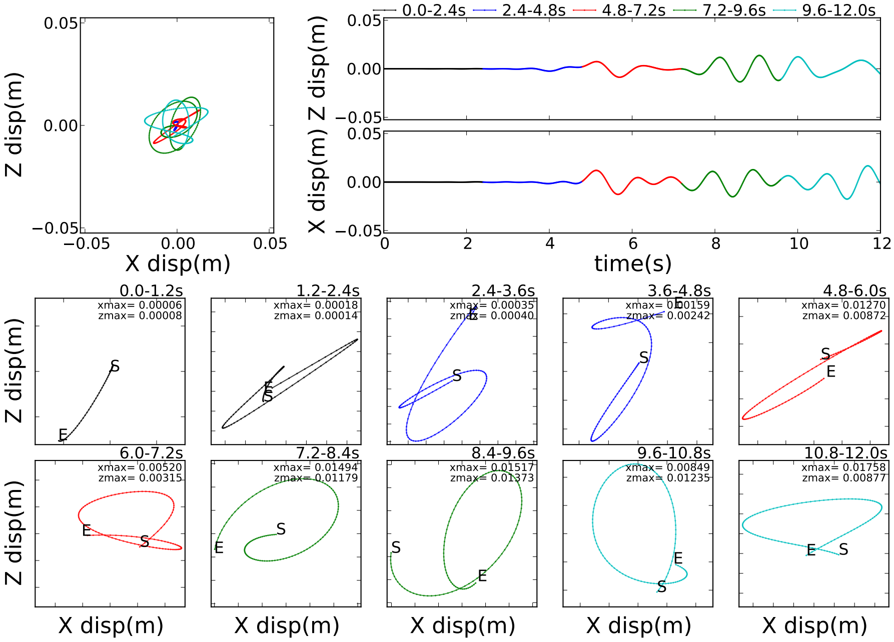


Figure 502.44: diagonal array X=4800m, Z=4800m

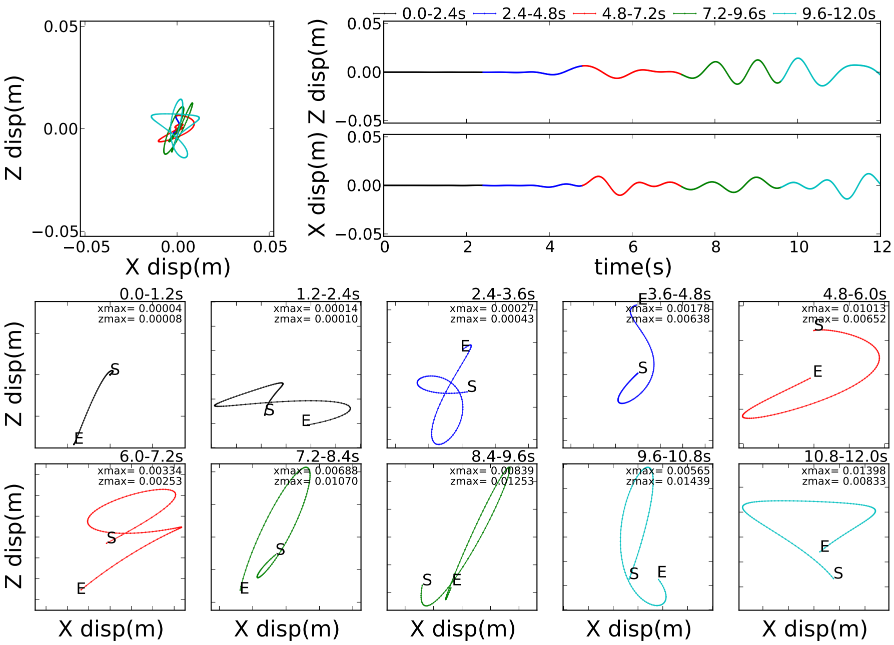


Figure 502.45: diagonal array X=4600m, Z=4600m

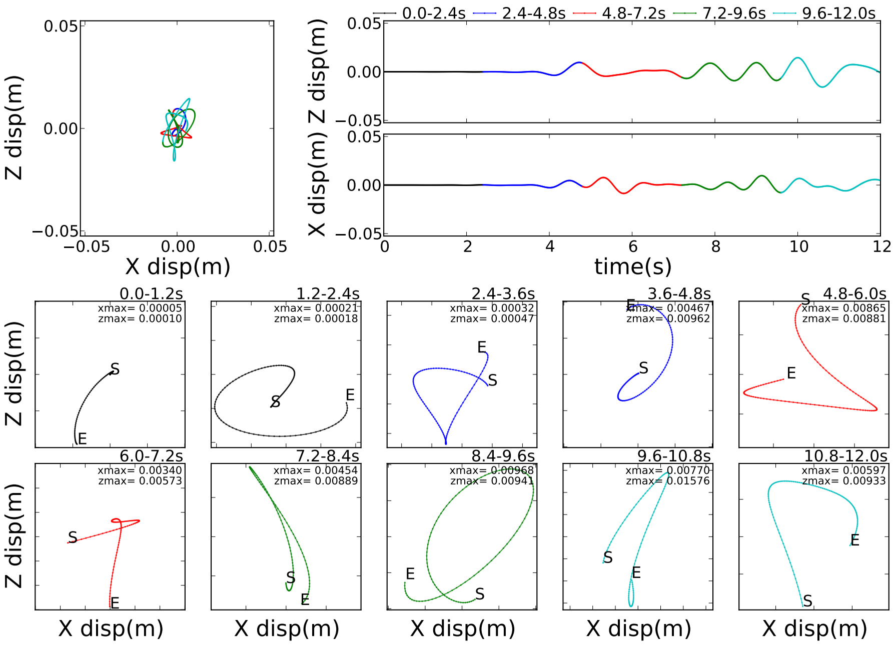


Figure 502.46: diagonal array X=4400m, Z=4400m

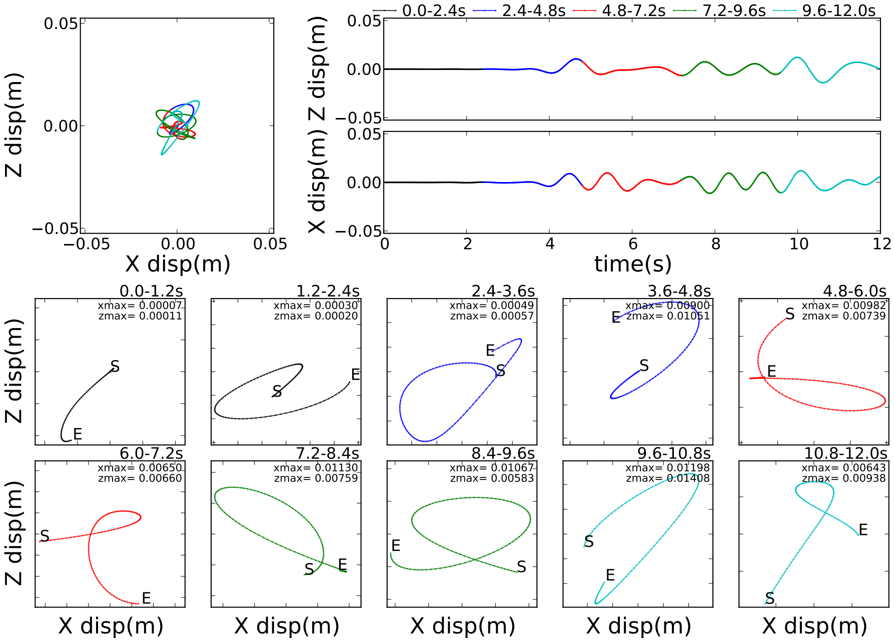


Figure 502.47: diagonal array X=4200m, Z=4200m

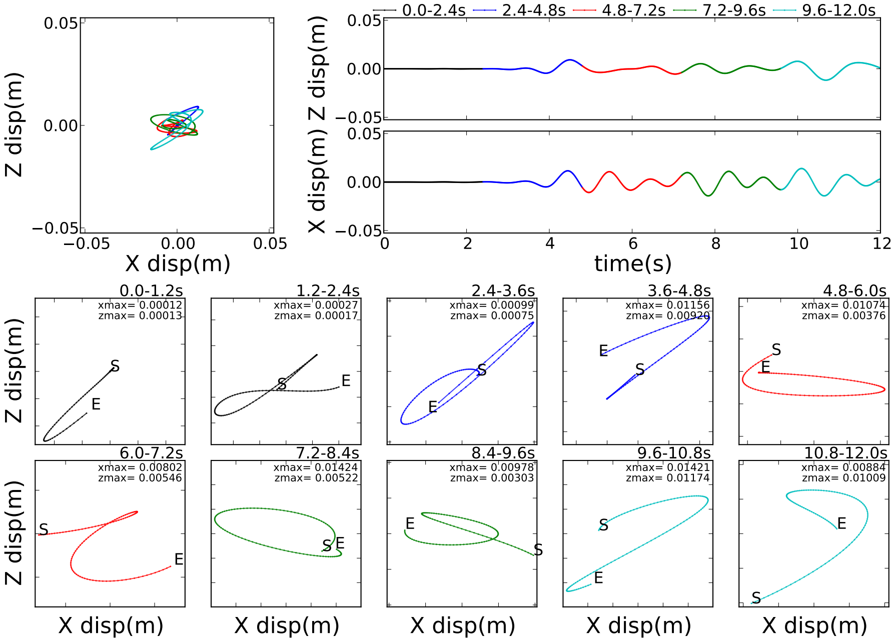


Figure 502.48: diagonal array X=4000m, Z=4000m

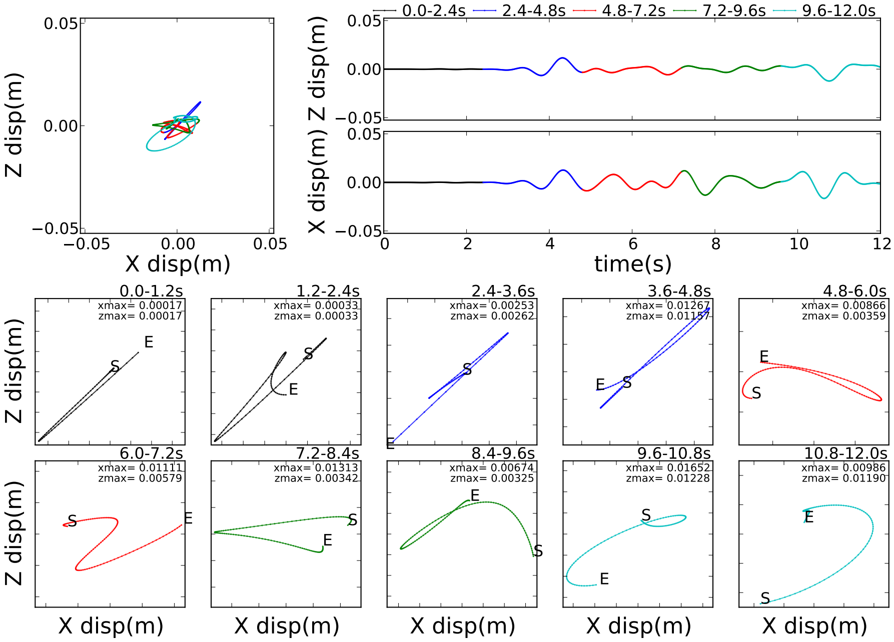


Figure 502.49: diagonal array X=3800m, Z=3800m

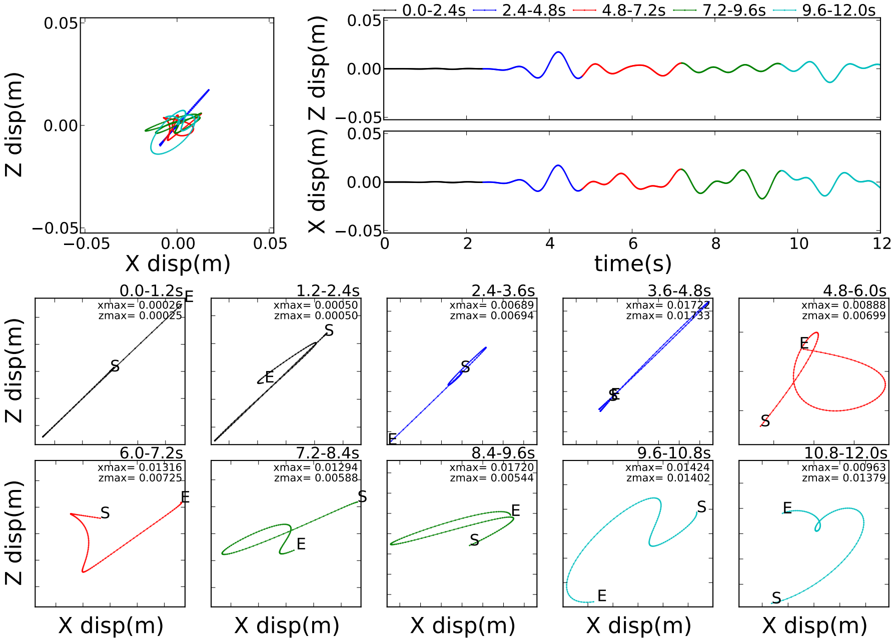


Figure 502.50: diagonal array X=3600m, Z=3600m

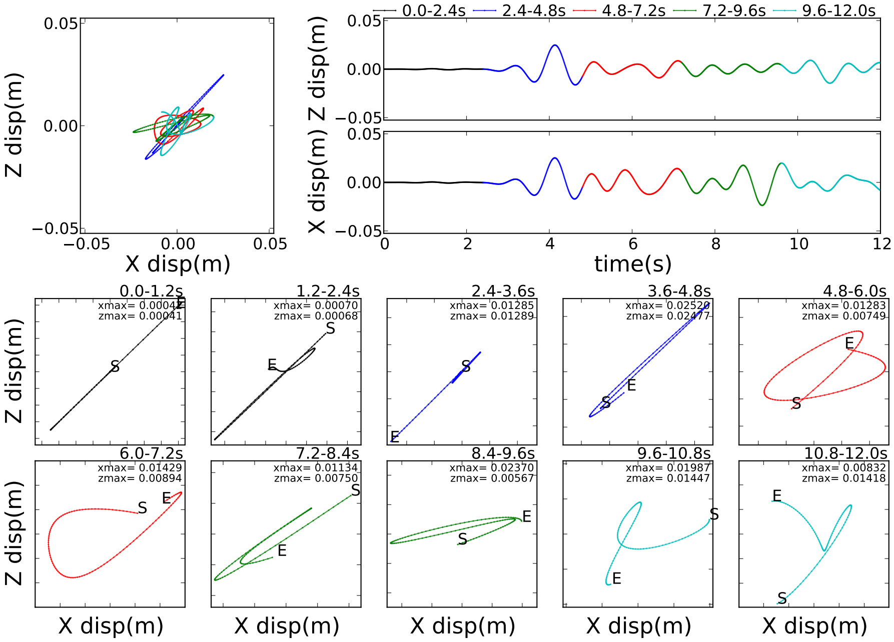


Figure 502.51: diagonal array X=3400m, Z=3400m

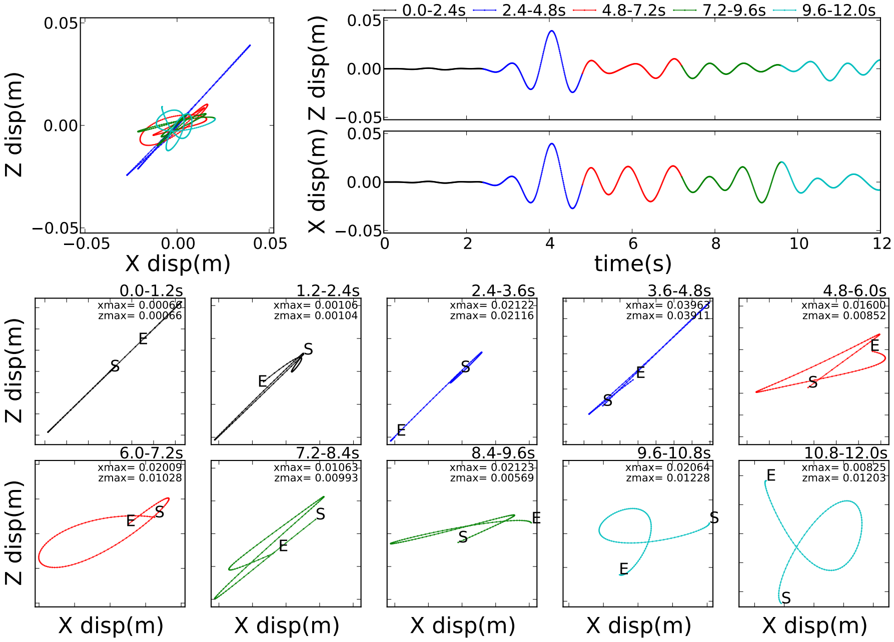


Figure 502.52: diagonal array X=3200m, Z=3200m

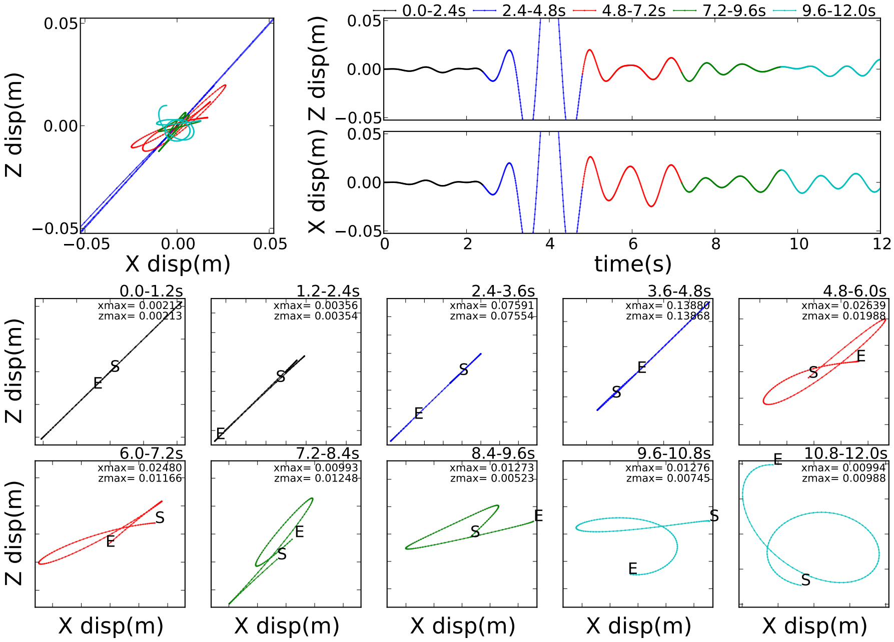


Figure 502.53: diagonal array X=3000m, Z=3000m

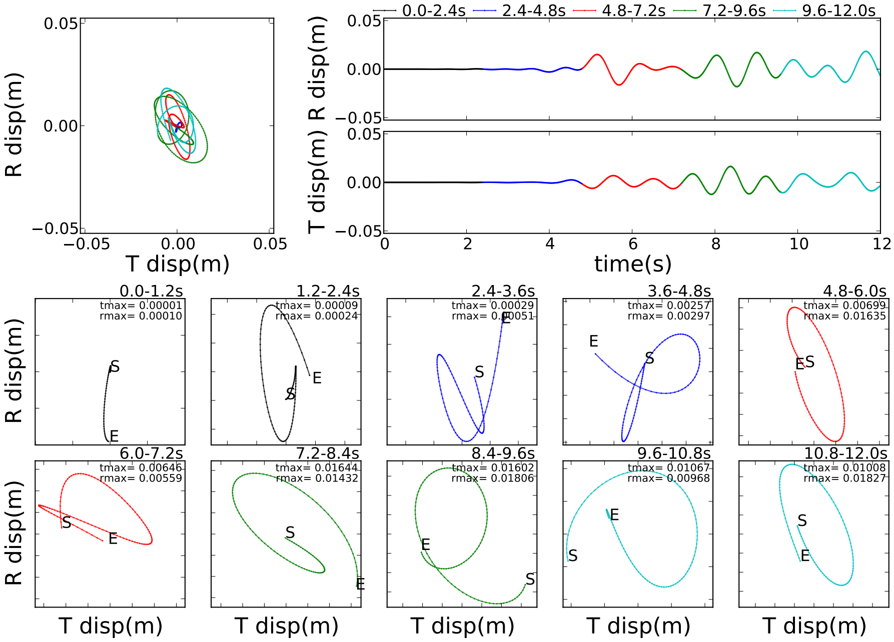


Figure 502.54: diagonal array X=5000m, Z=5000m

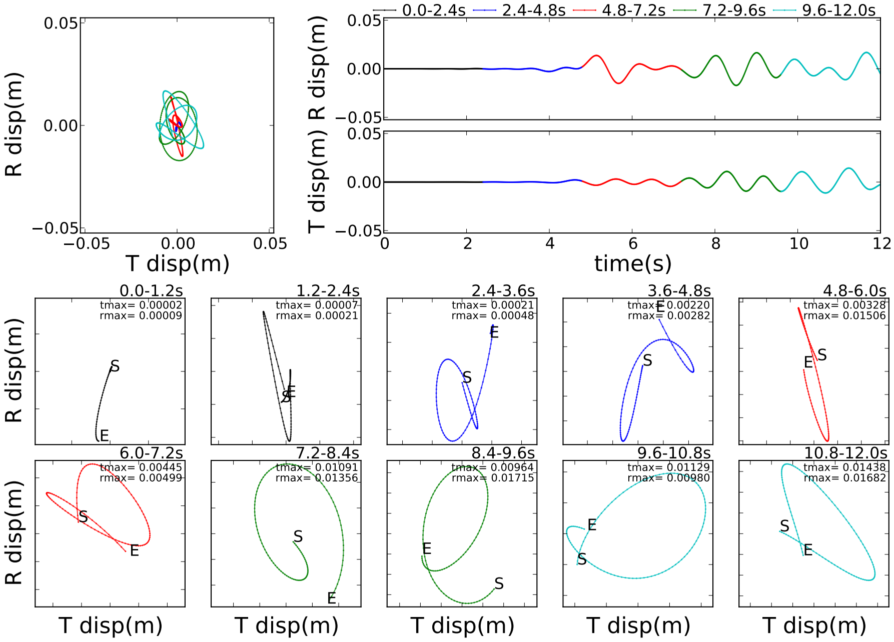


Figure 502.55: diagonal array X=4800m, Z=4800m

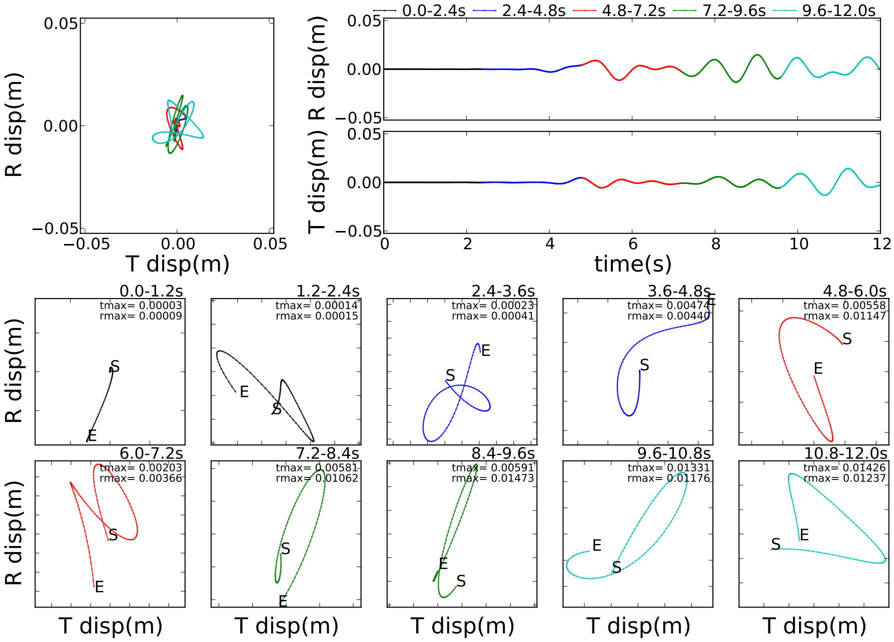


Figure 502.56: diagonal array X=4600m, Z=4600m

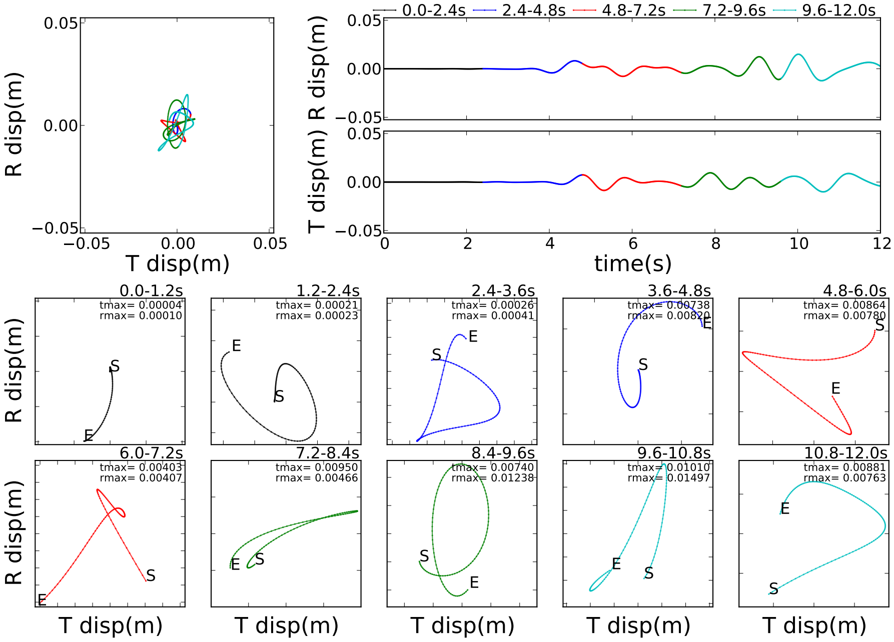


Figure 502.57: diagonal array X=4400m, Z=4400m

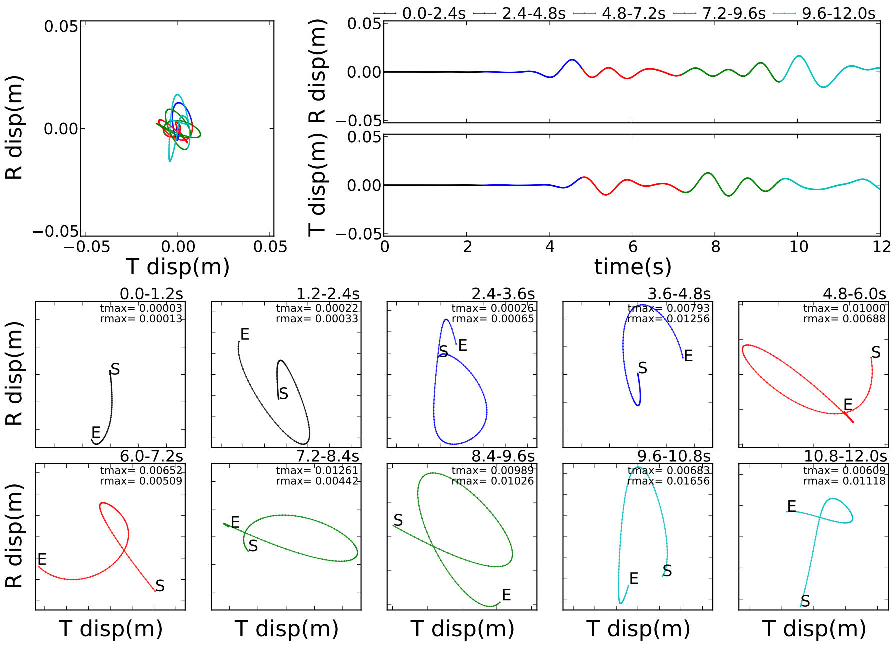


Figure 502.58: diagonal array X=4200m, Z=4200m

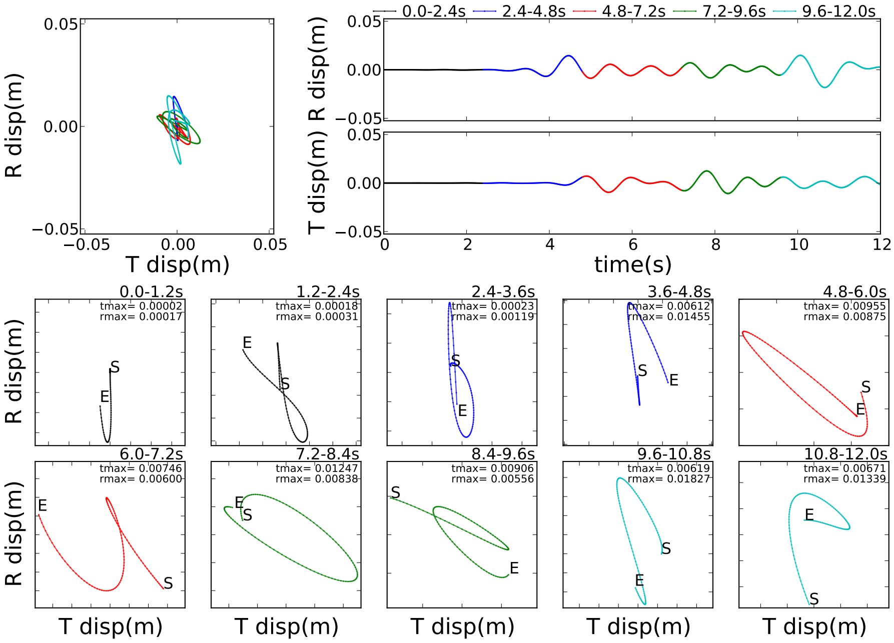


Figure 502.59: diagonal array X=4000m, Z=4000m

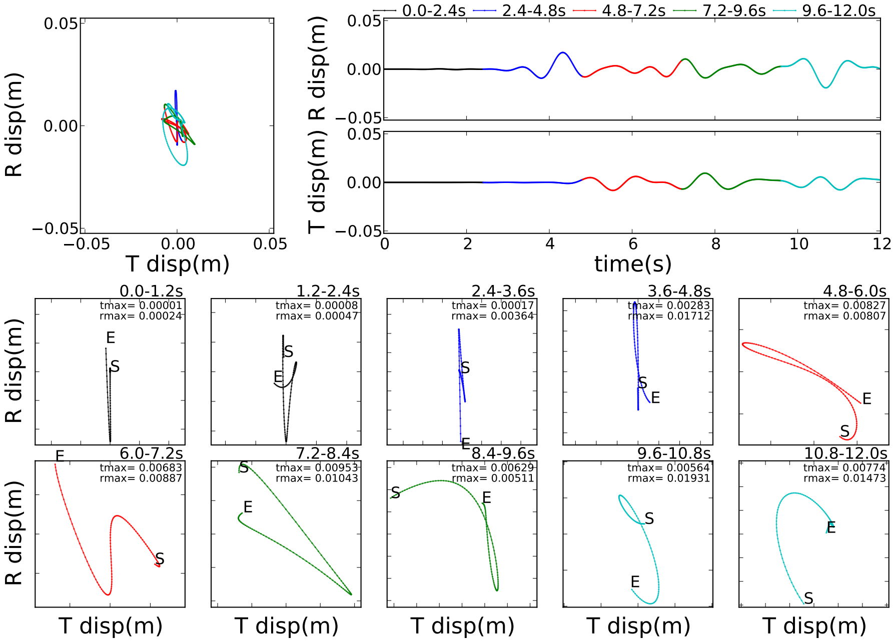


Figure 502.60: diagonal array X=3800m, Z=3800m

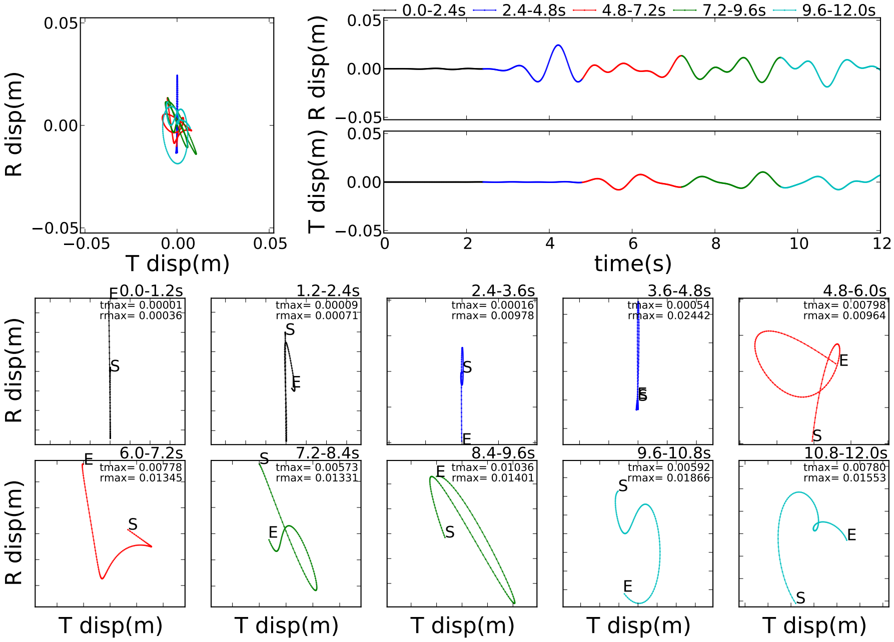


Figure 502.61: diagonal array X=3600m, Z=3600m

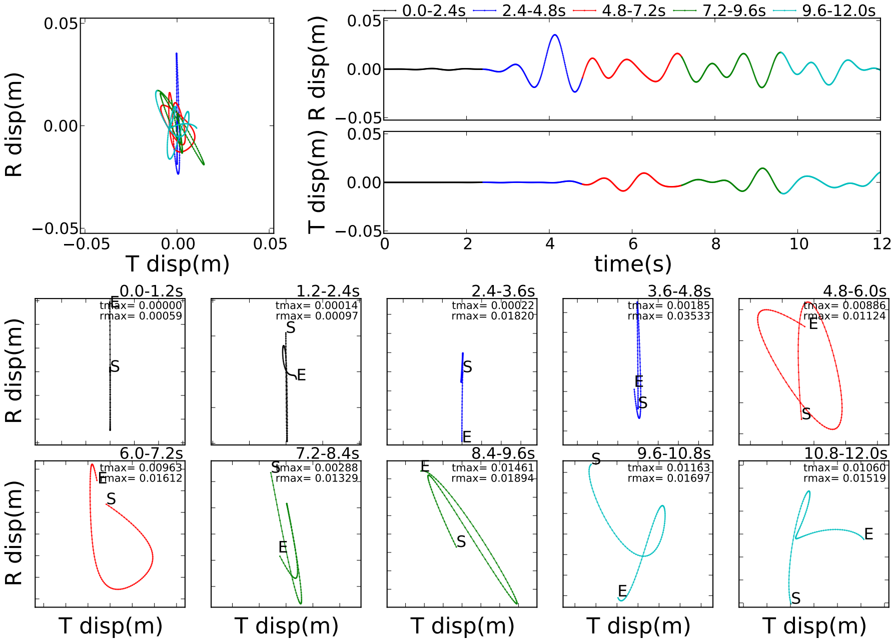


Figure 502.62: diagonal array X=3400m, Z=3400m

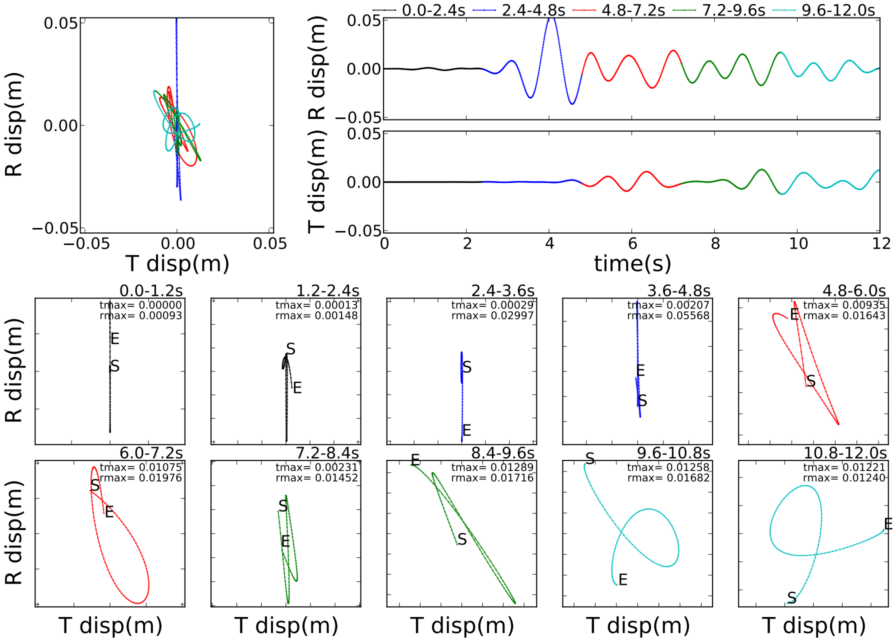


Figure 502.63: diagonal array X=3200m, Z=3200m

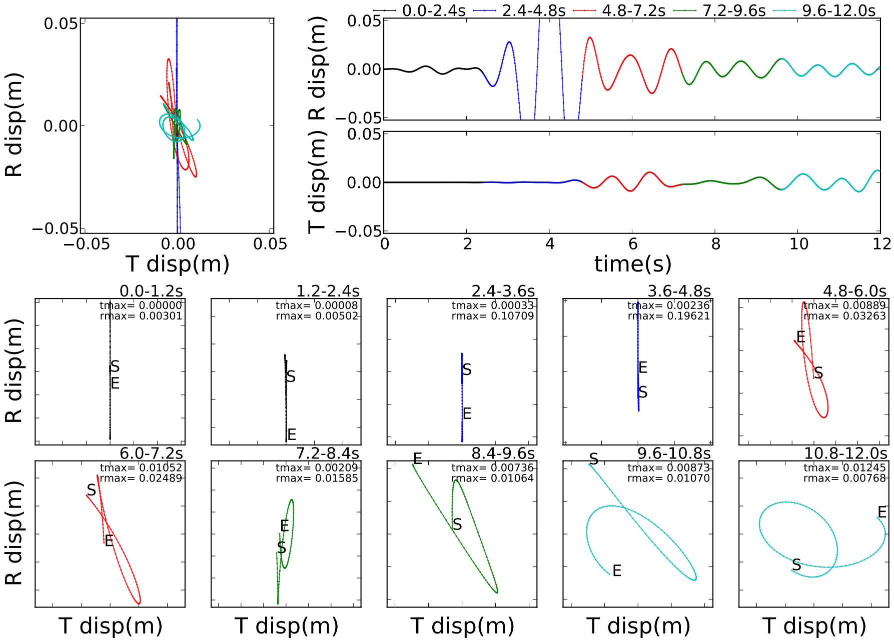


Figure 502.64: diagonal array X=3000m, Z=3000m

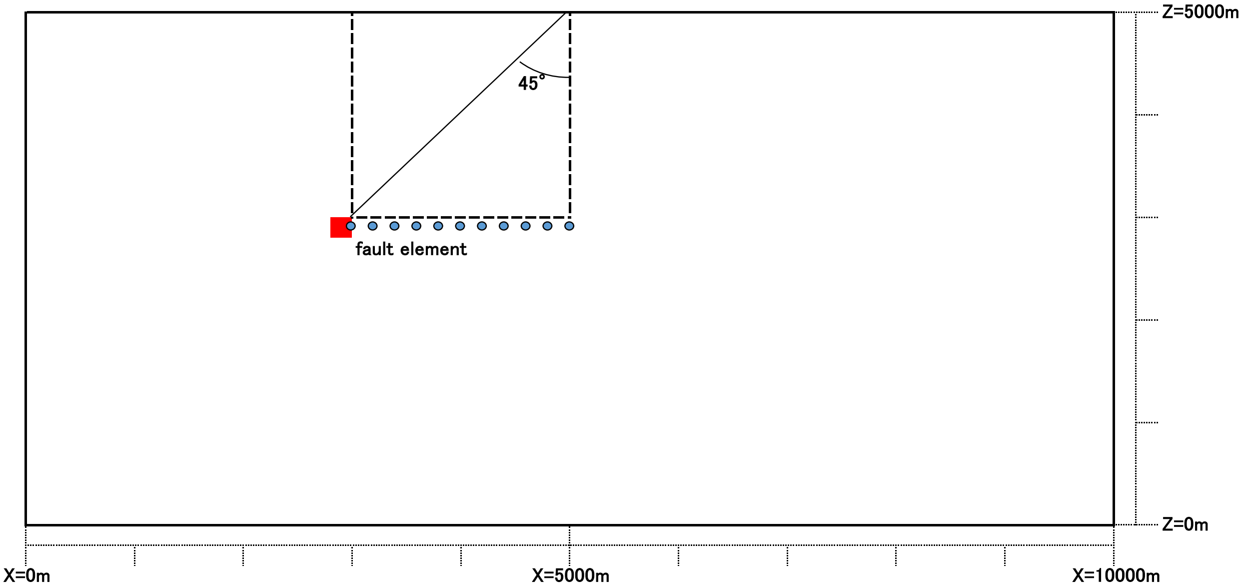


Figure 502.65: Observation array of Array2

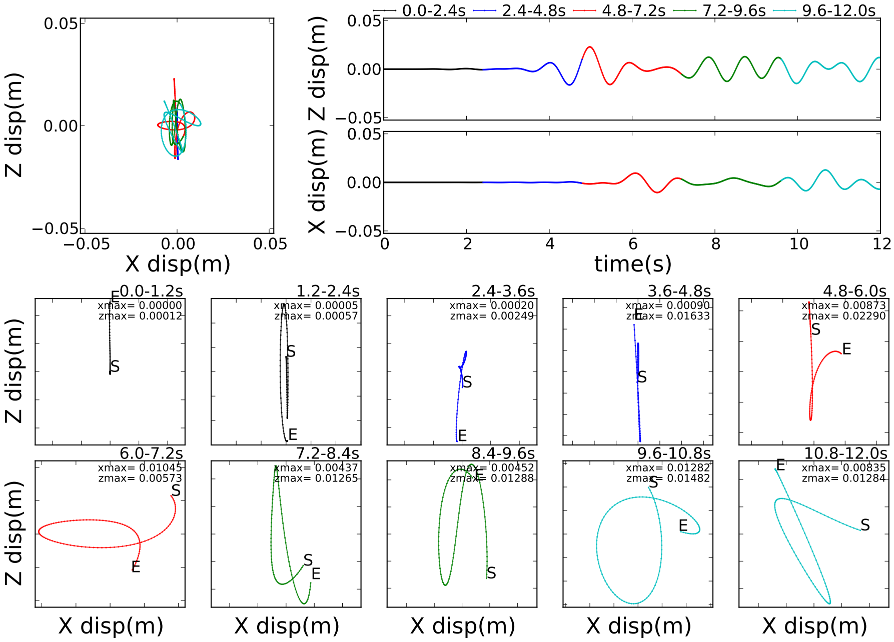


Figure 502.66: X array2 X=5000m, Z=2950m

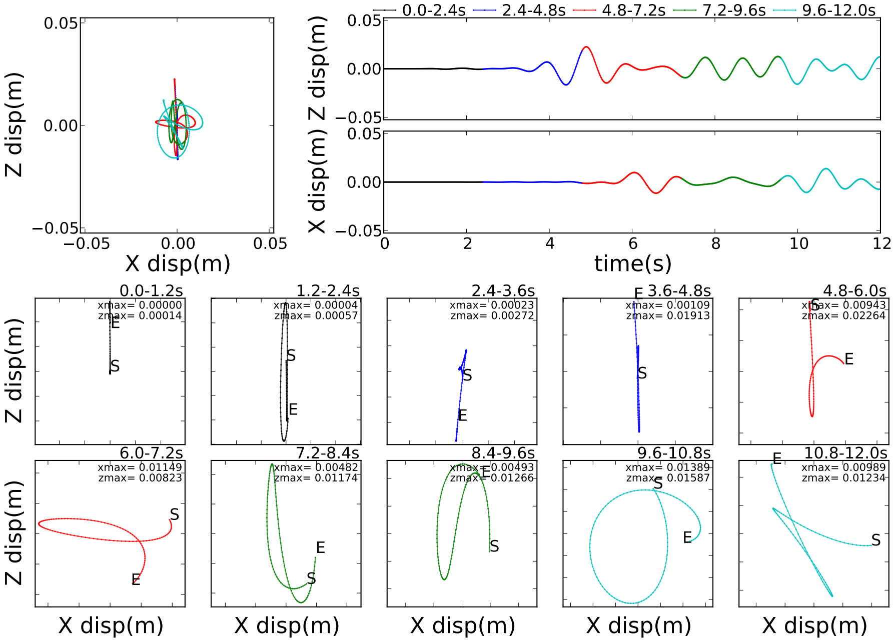


Figure 502.67: X array2 X=4800m, Z=2950m

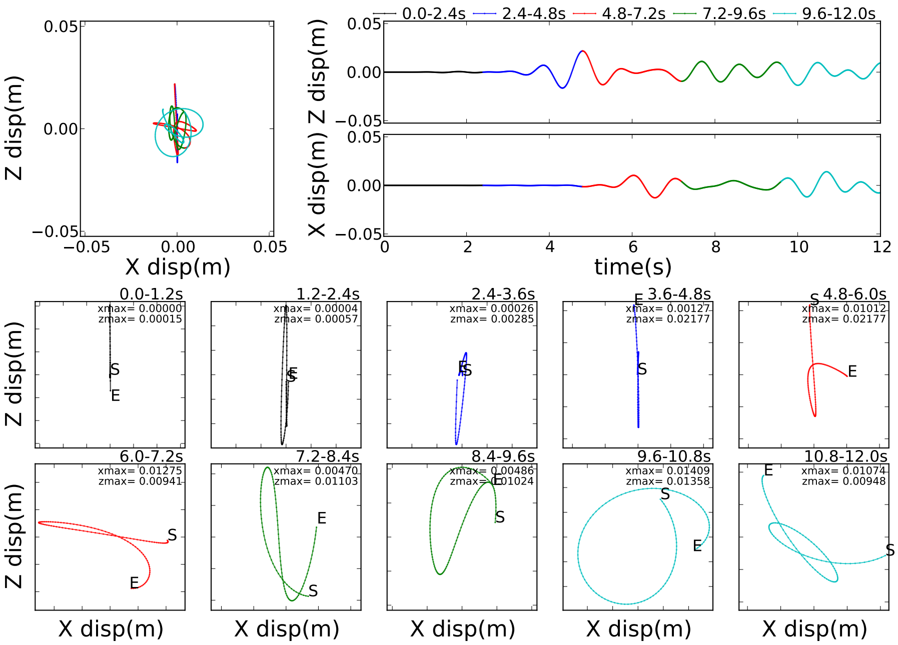


Figure 502.68: X array2 X=4600m, Z=2950m

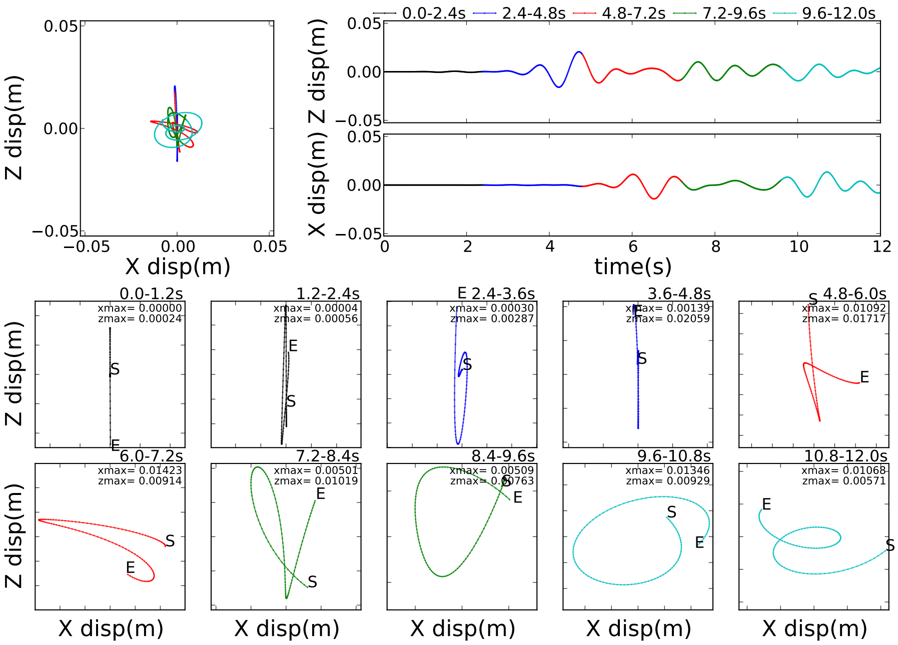


Figure 502.69: X array2 X=4400m, Z=2950m

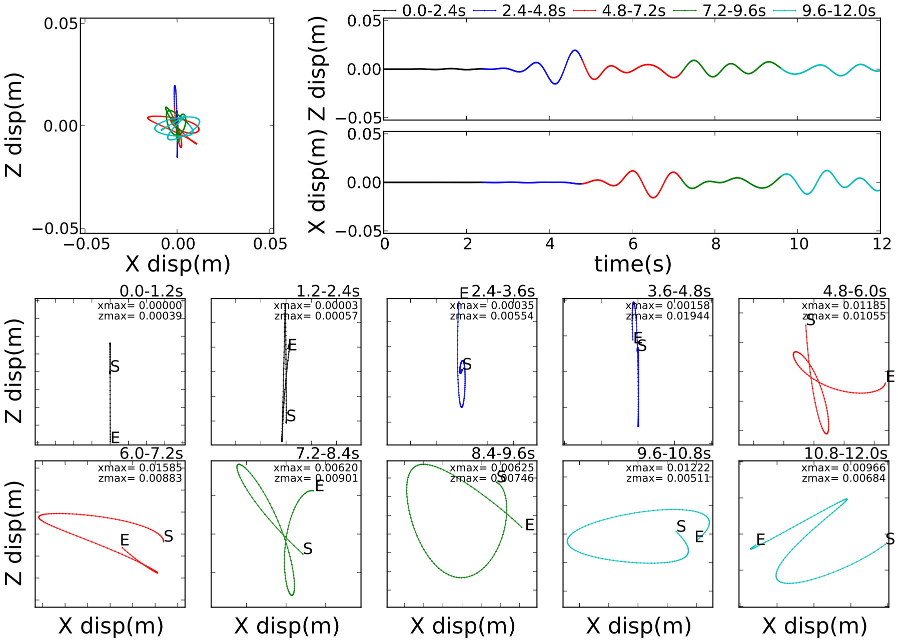


Figure 502.70: X array2 X=4200m, Z=2950m

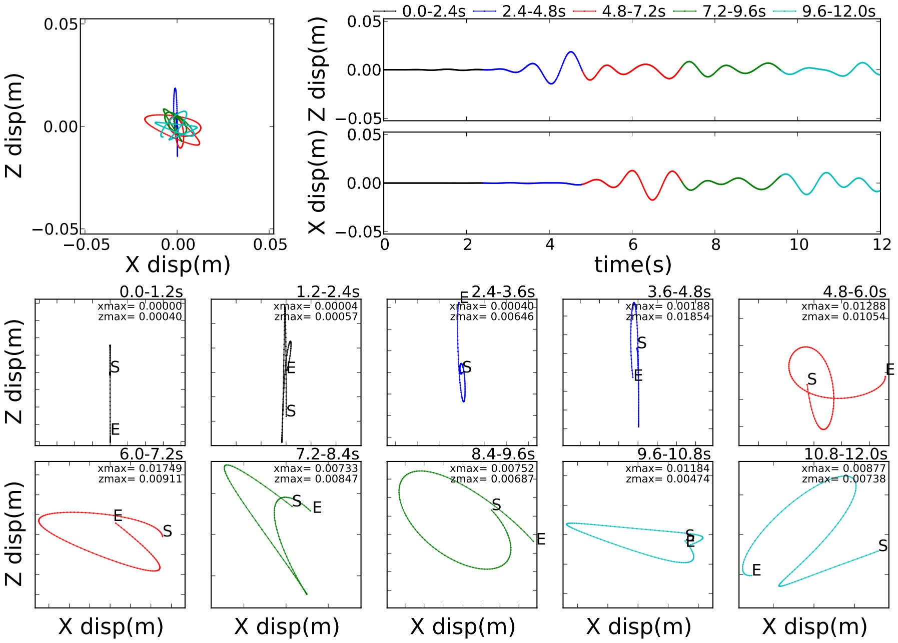


Figure 502.71: X array2 X=4000m, Z=2950m

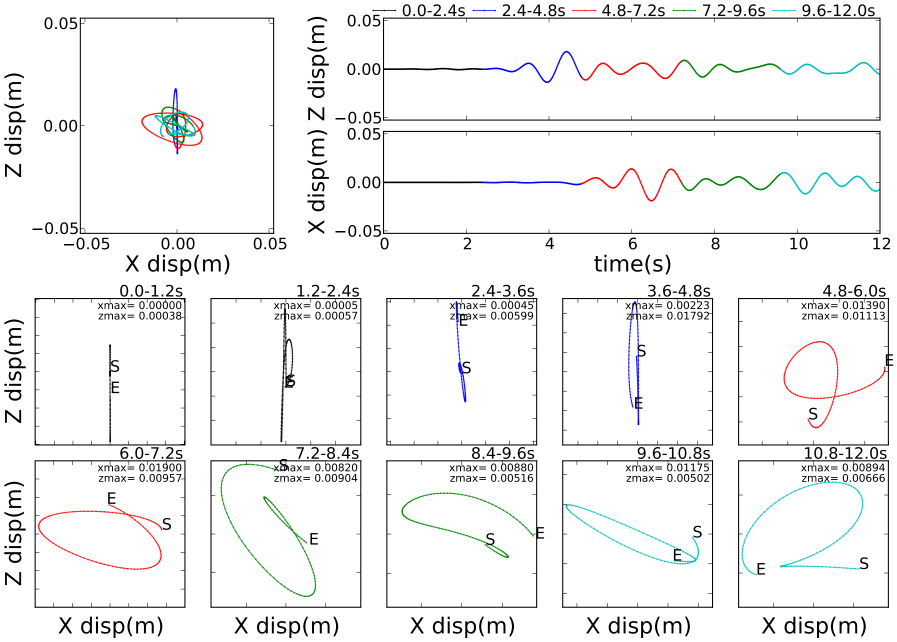


Figure 502.72: X array2 X=3800m, Z=2950m

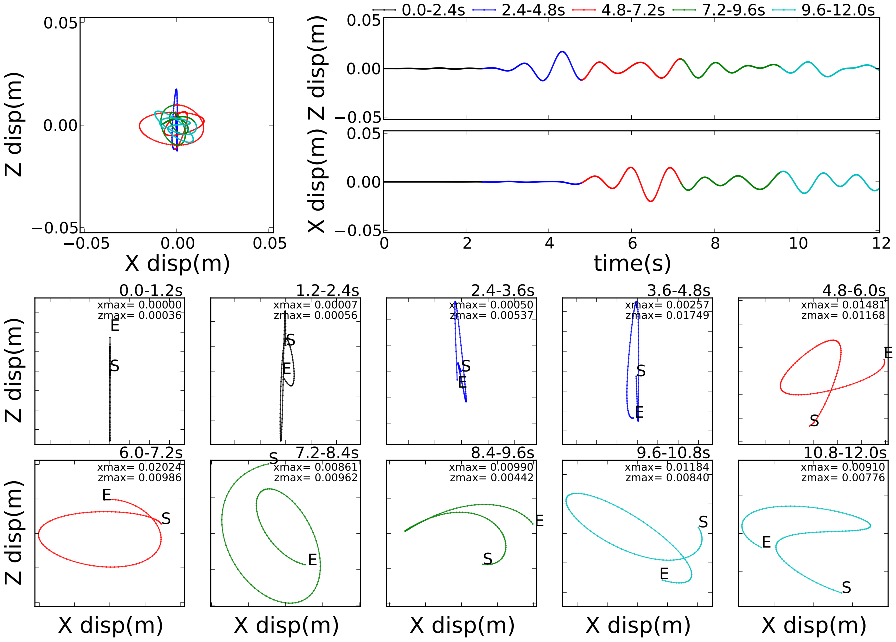


Figure 502.73: X array2 X=3600m, Z=2950m

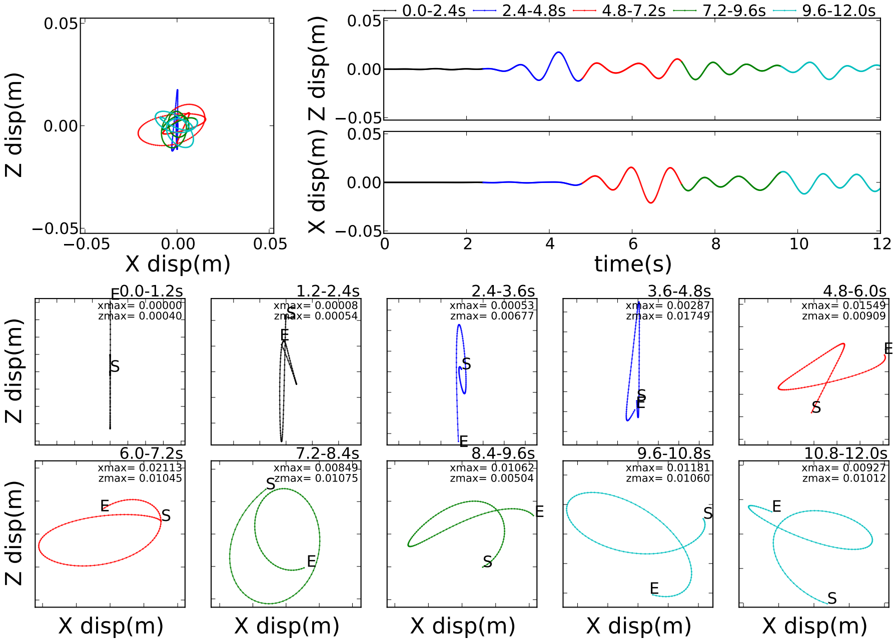


Figure 502.74: X array2 X=3400m, Z=2950m

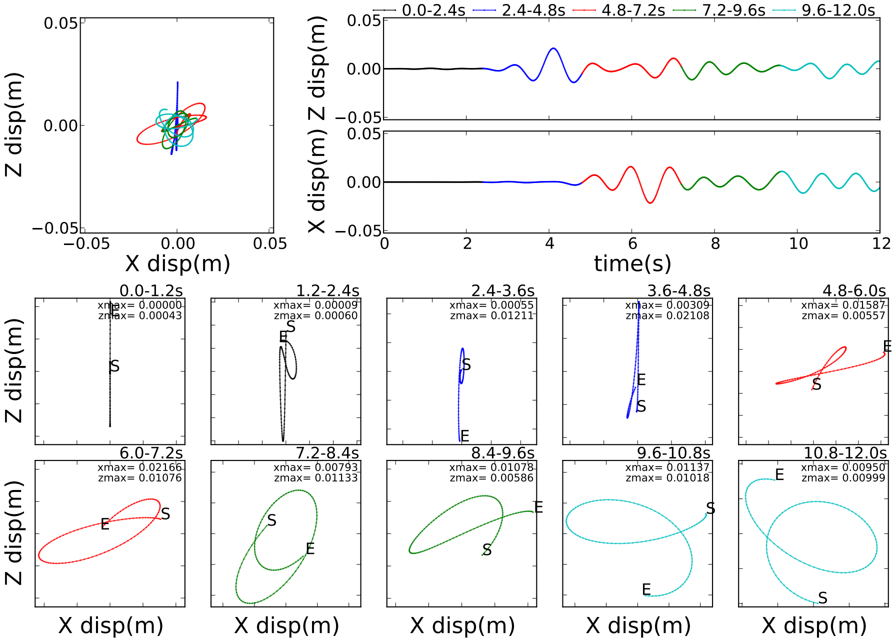


Figure 502.75: X array2 X=3200m, Z=2950m

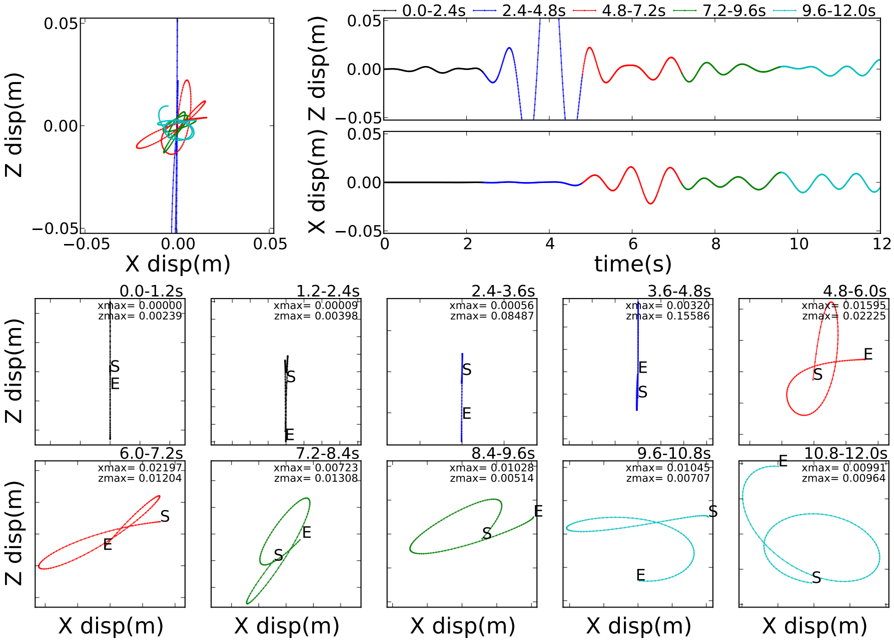


Figure 502.76: X array2 X=3000m, Z=2950m

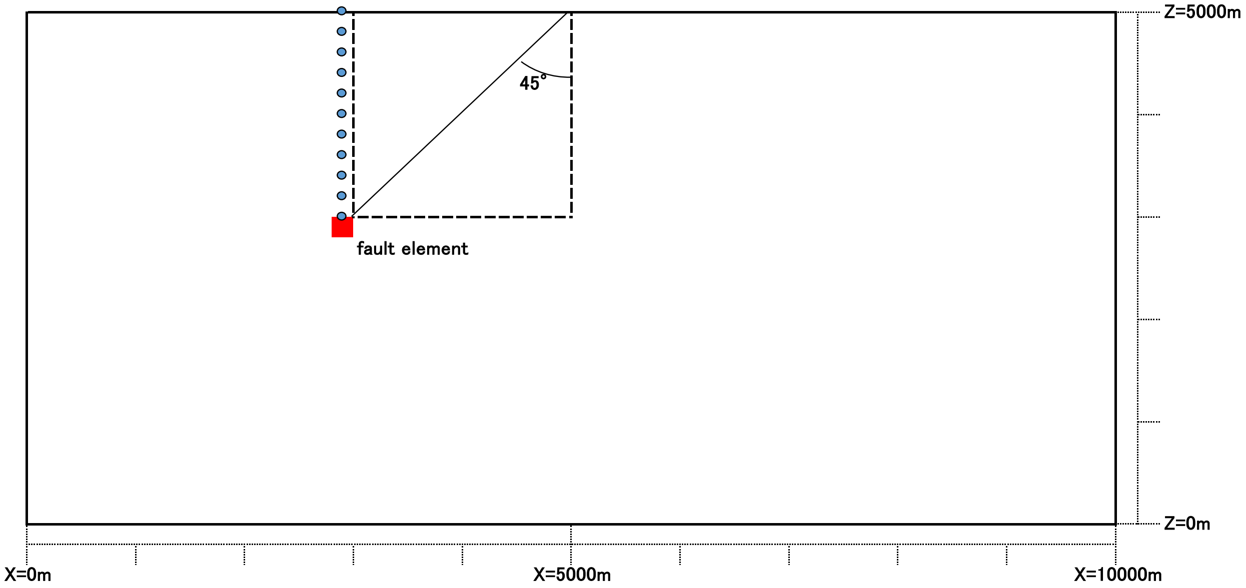


Figure 502.77: Observation array of Array3

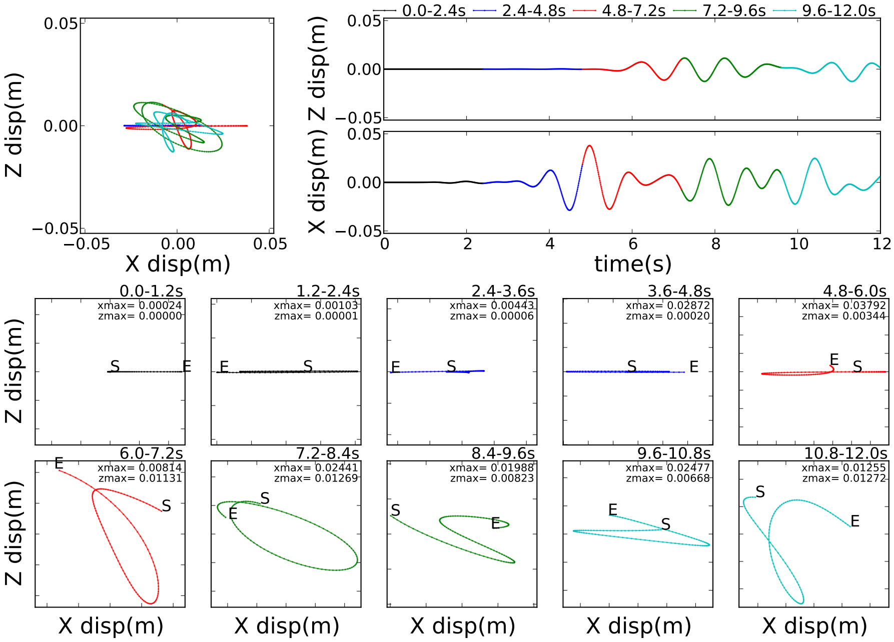


Figure 502.78: Z array2 X=2950m, Z=5000m

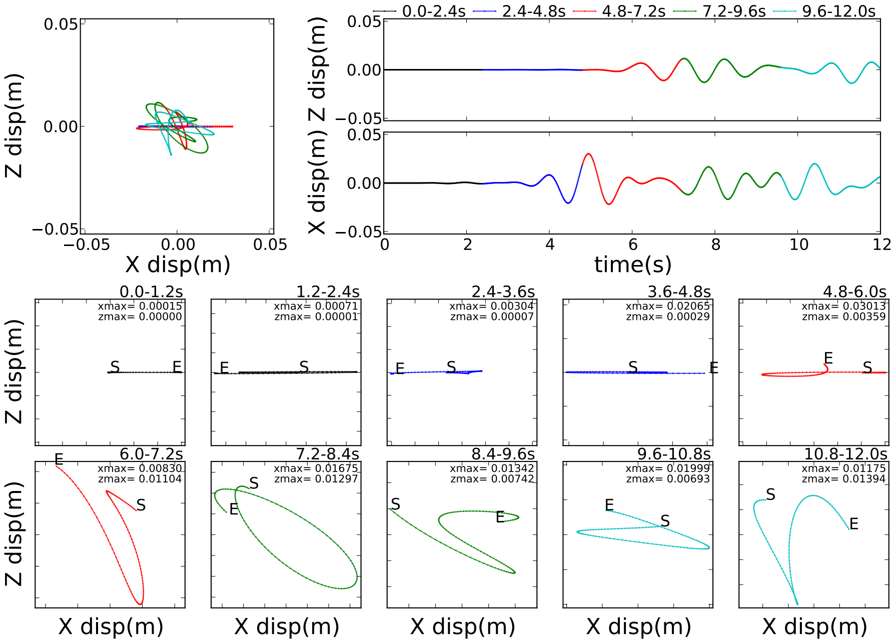


Figure 502.79: Z array2 X=2950m, Z=4800m

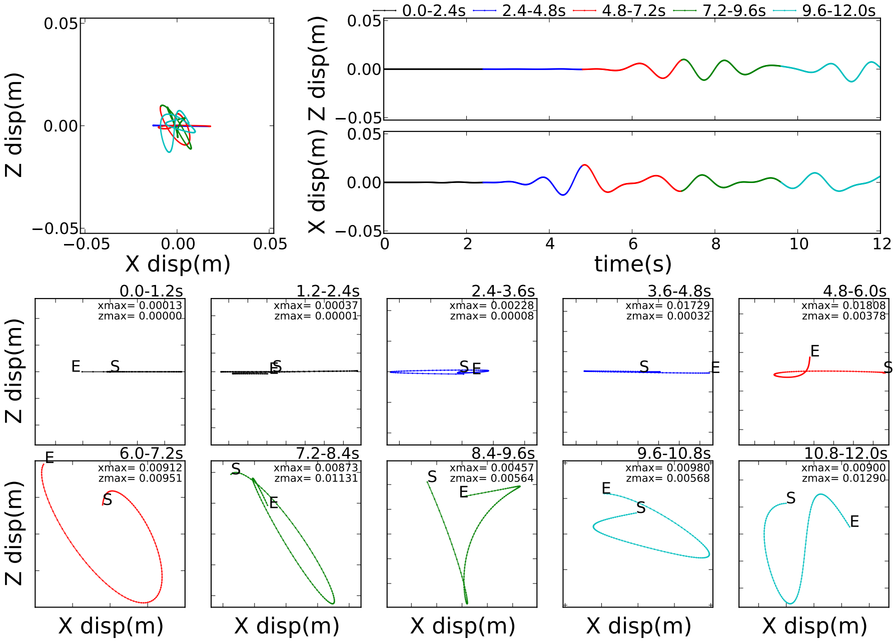


Figure 502.80: Z array2 X=2950m, Z=4600m

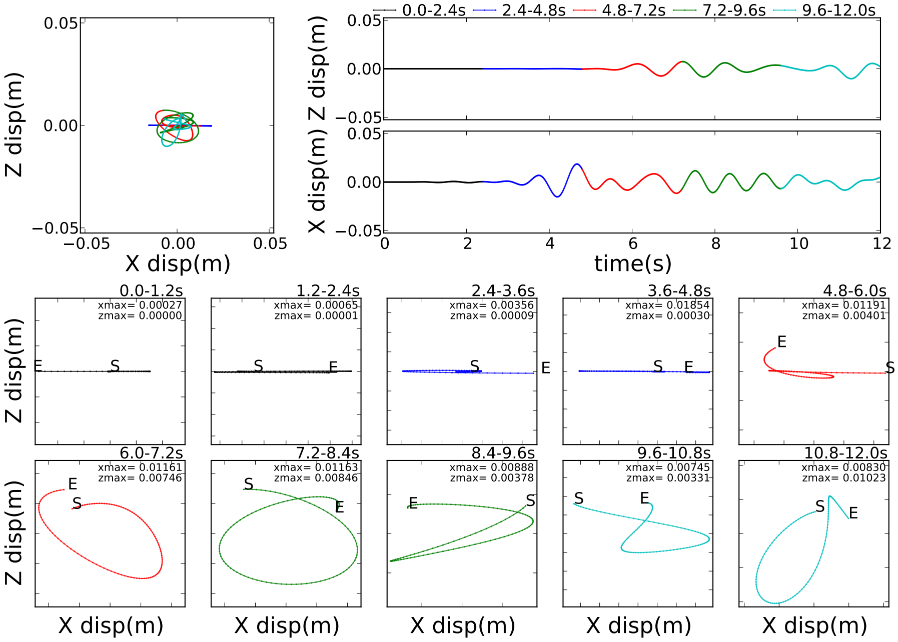


Figure 502.81: Z array2 X=2950m, Z=4400m

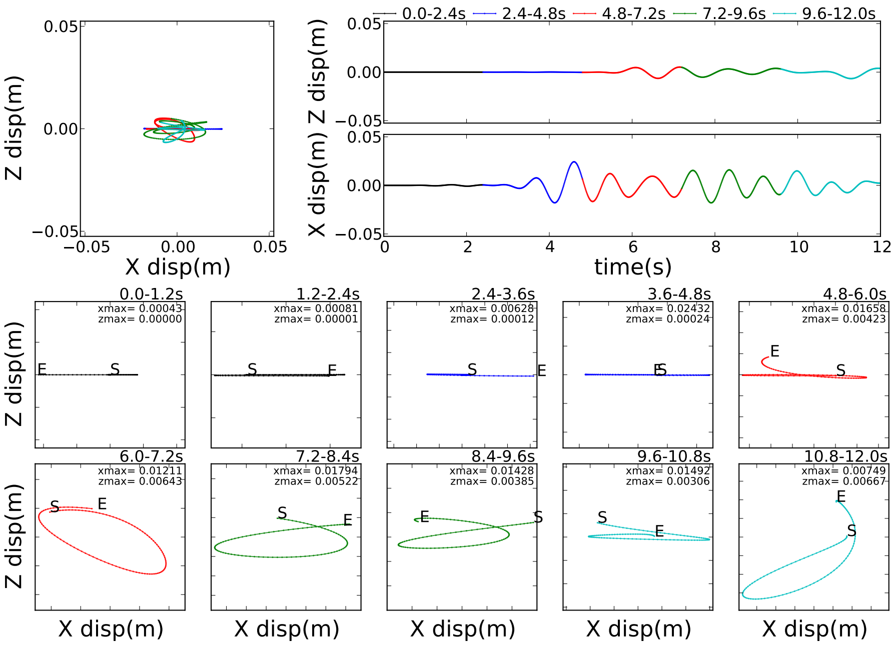


Figure 502.82: Z array2 X=2950m, Z=4200m

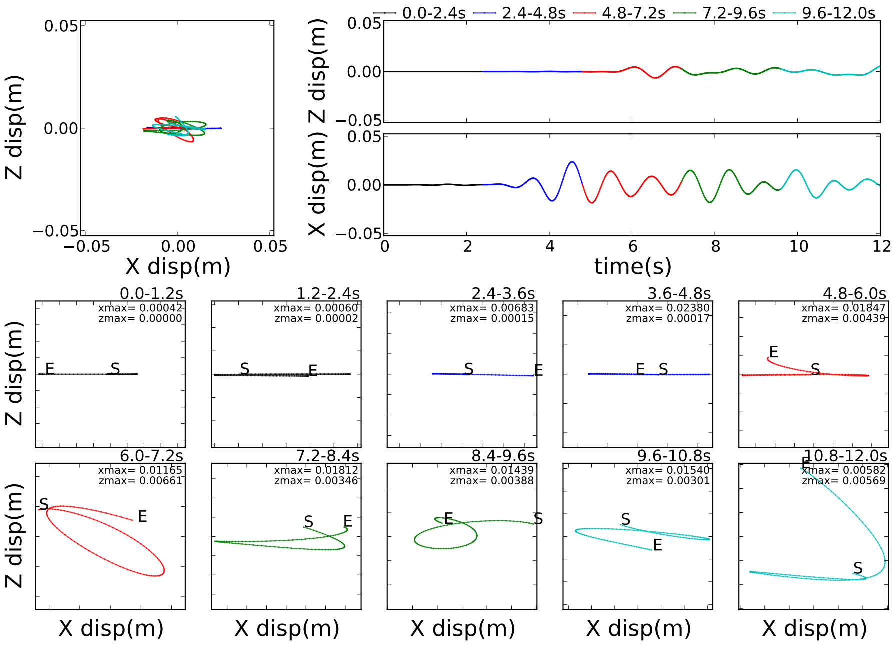


Figure 502.83: Z array2 X=2950m, Z=4000m

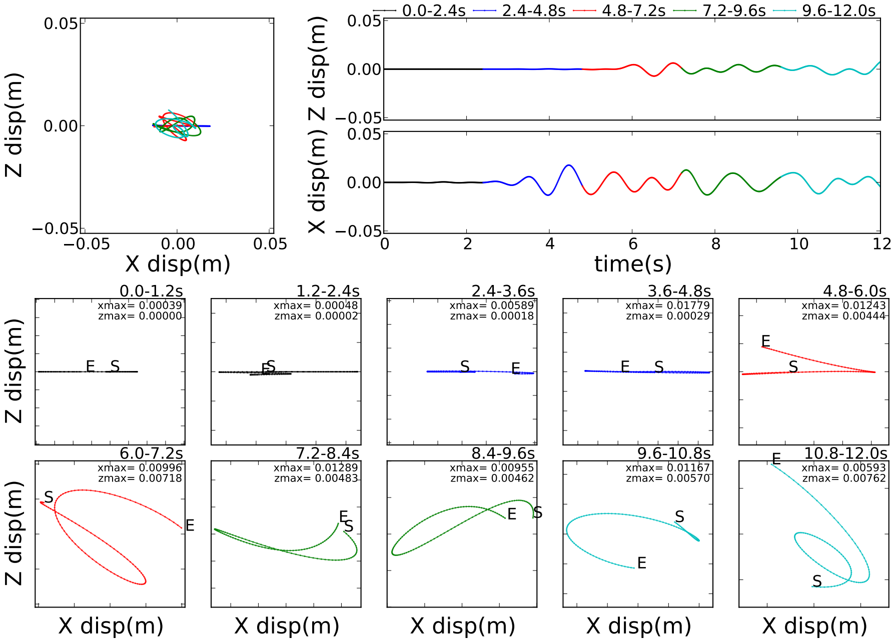


Figure 502.84: Z array2 X=2950m, Z=3800m

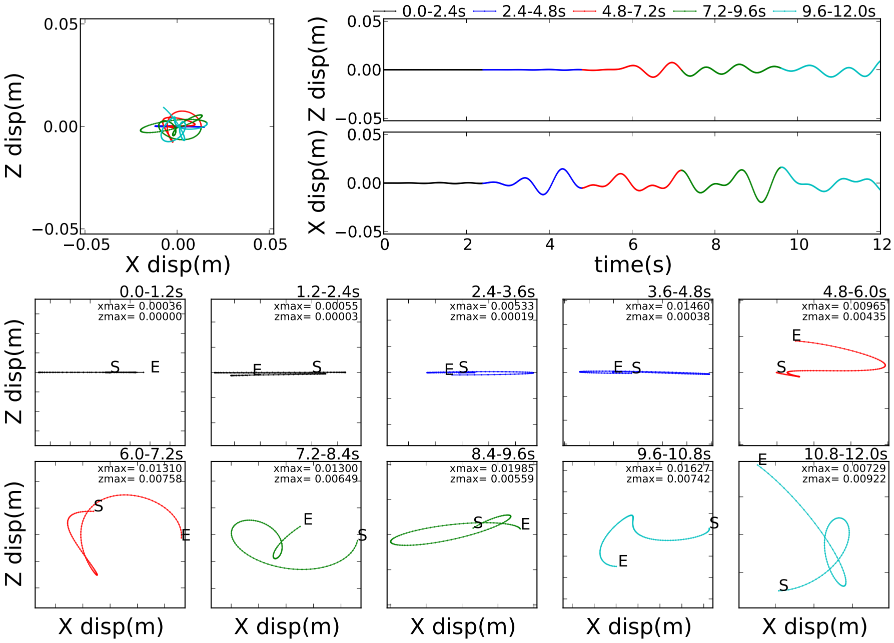


Figure 502.85: Z array2 X=2950m, Z=3600m

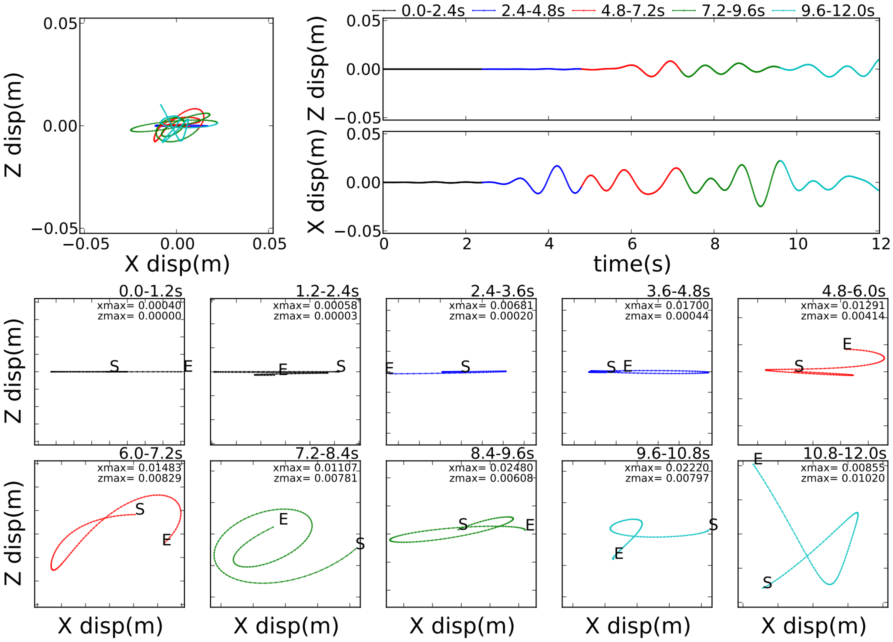


Figure 502.86: Z array2 X=2950m, Z=3400m

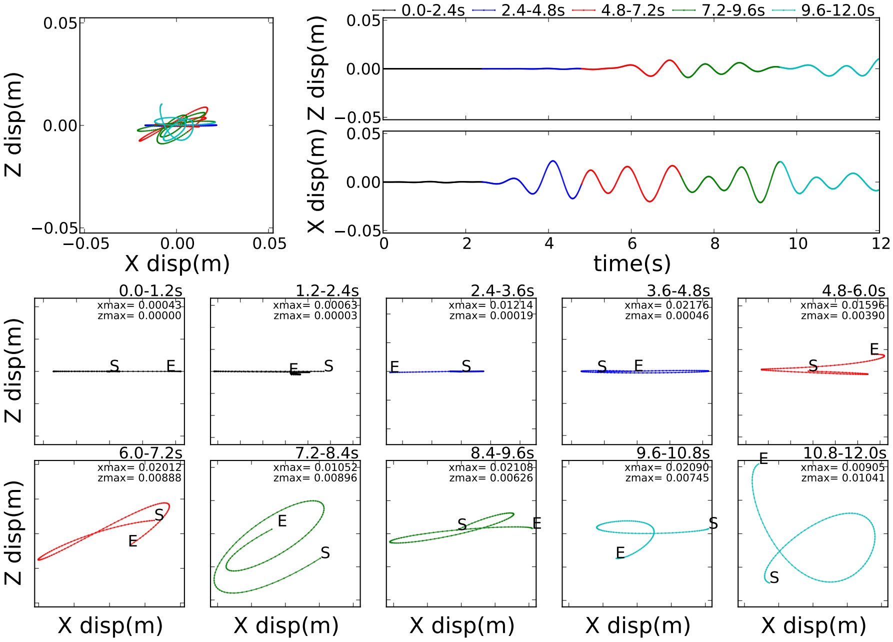


Figure 502.87: Z array2 X=2950m, Z=3200m

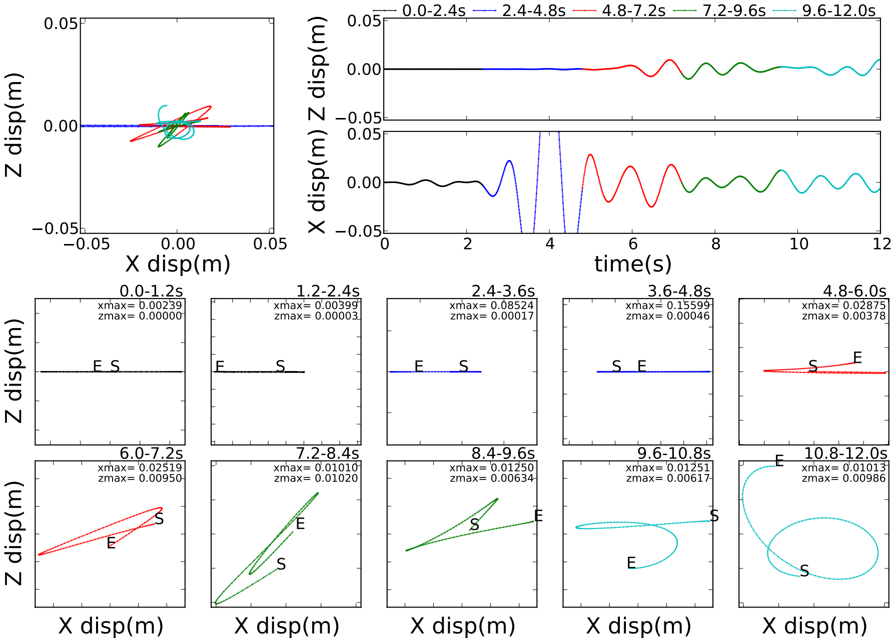


Figure 502.88: Z array2 X=2950m, Z=3000m

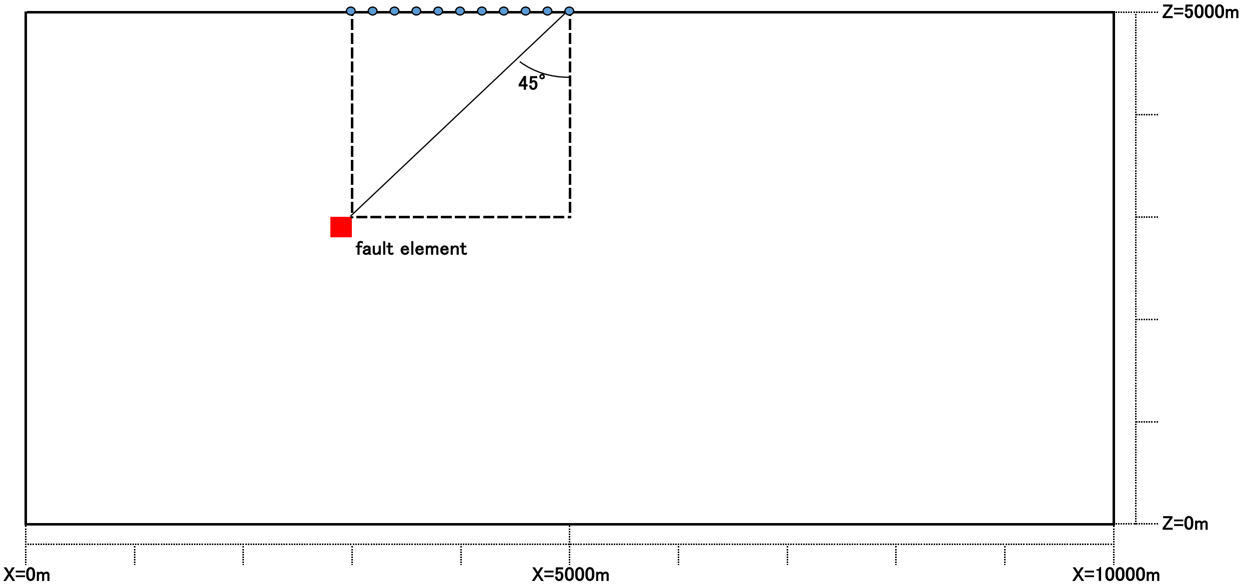


Figure 502.89: Observation array of Array4

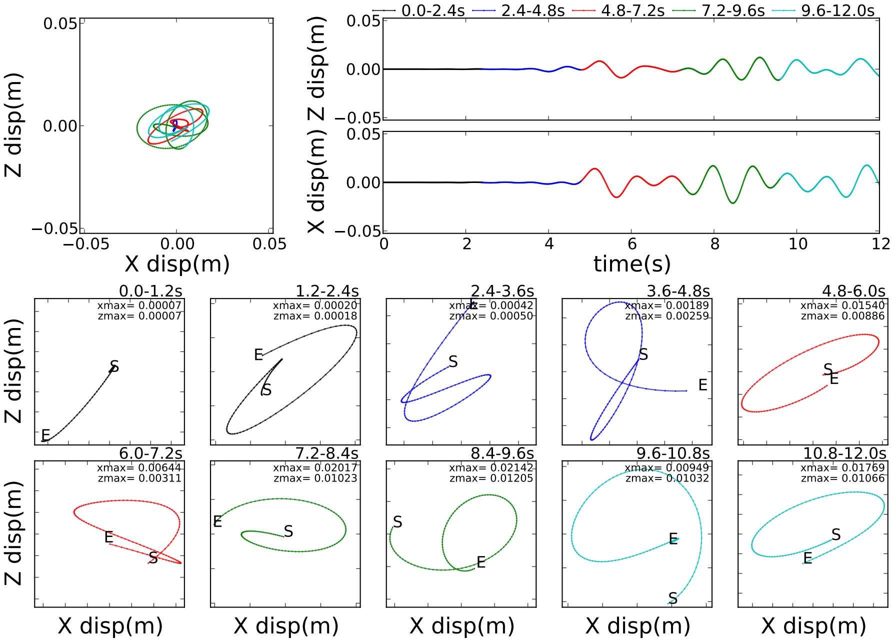


Figure 502.90: X array1 X=5000m, Z=5000m

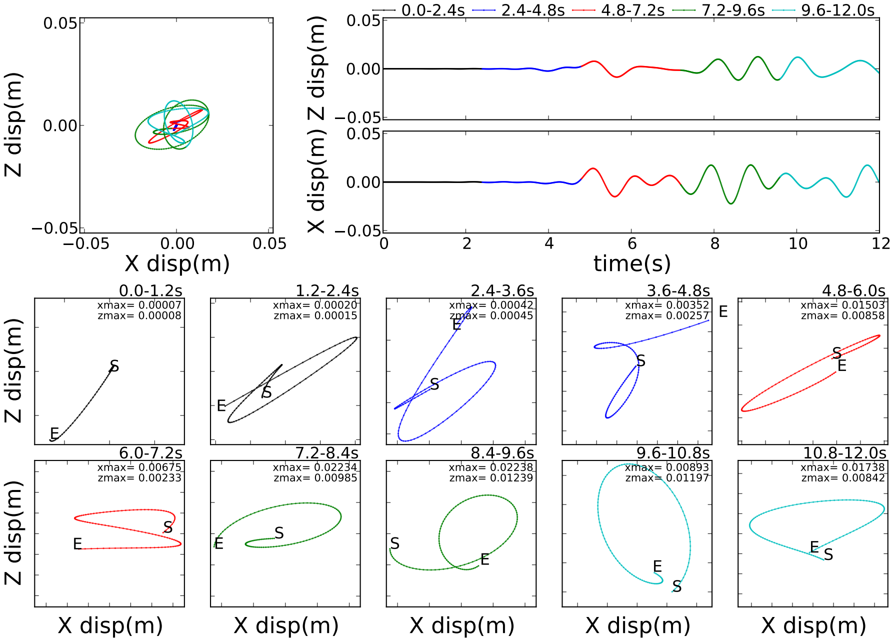


Figure 502.91: X array1 X=4800m, Z=5000m

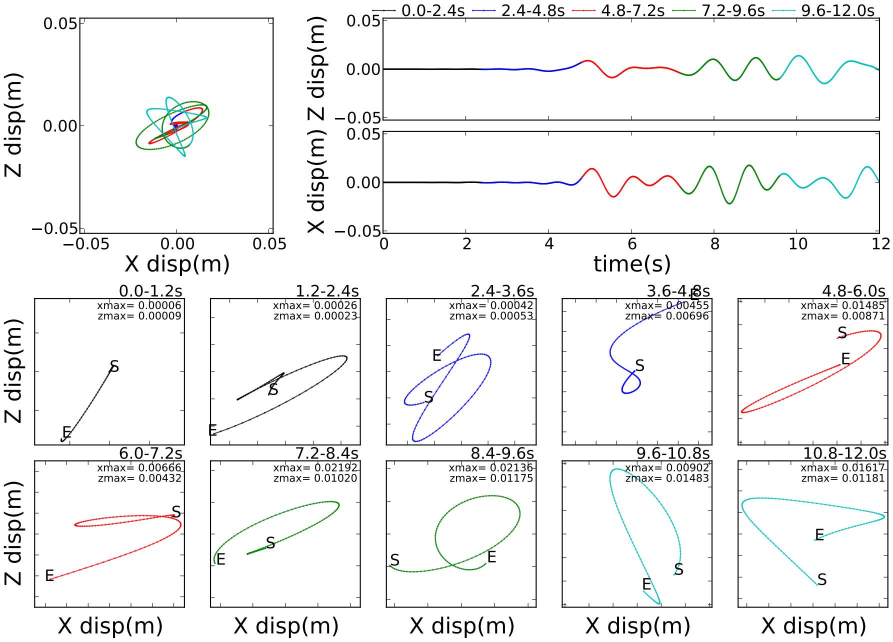


Figure 502.92: X array1 X=4600m, Z=5000m

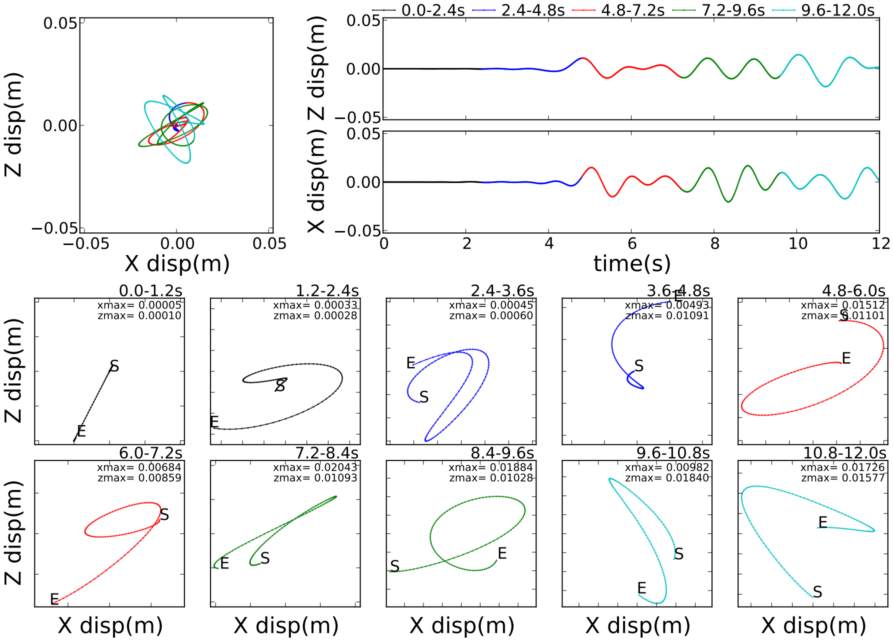


Figure 502.93: X array1 X=4400m, Z=5000m

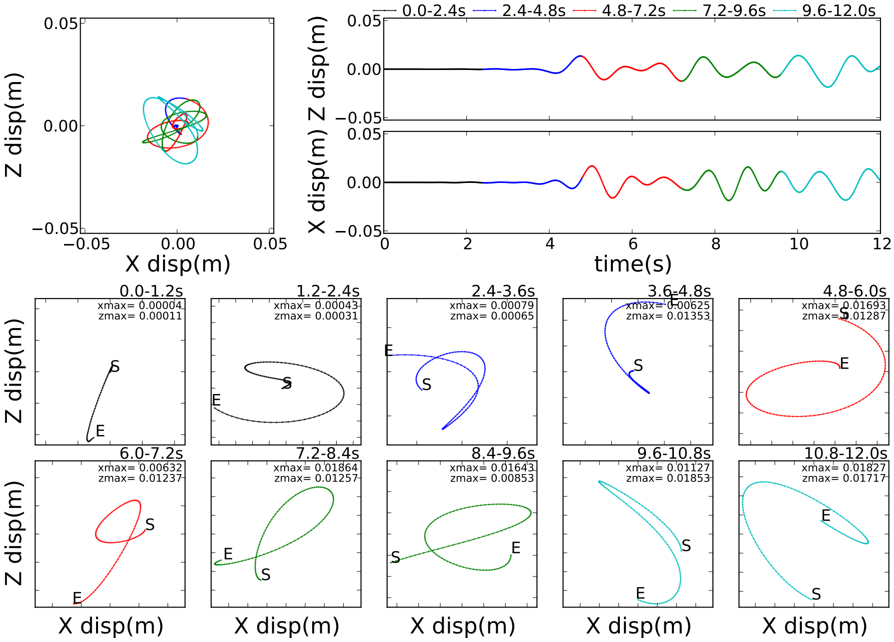


Figure 502.94: X array1 X=4200m, Z=5000m

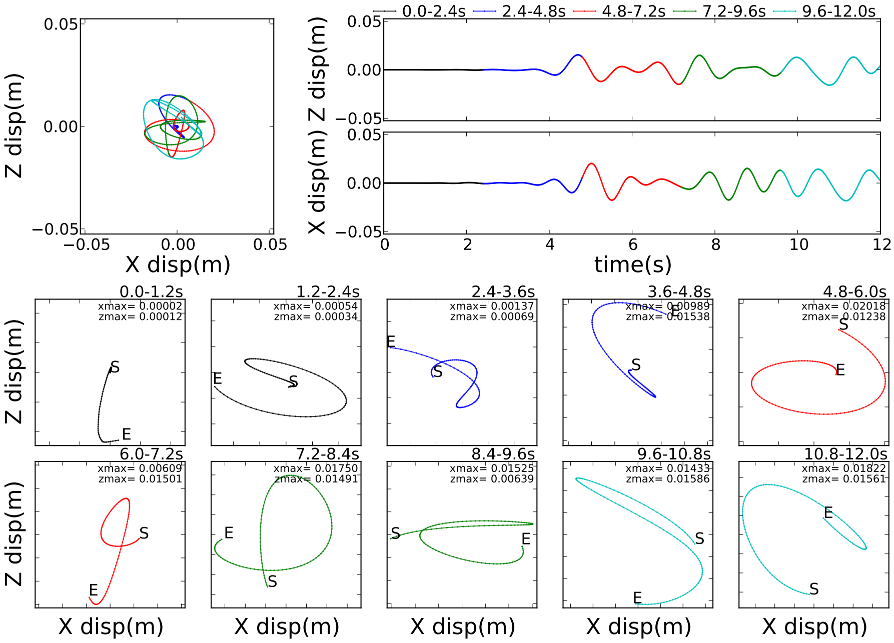


Figure 502.95: X array1 X=4000m, Z=5000m

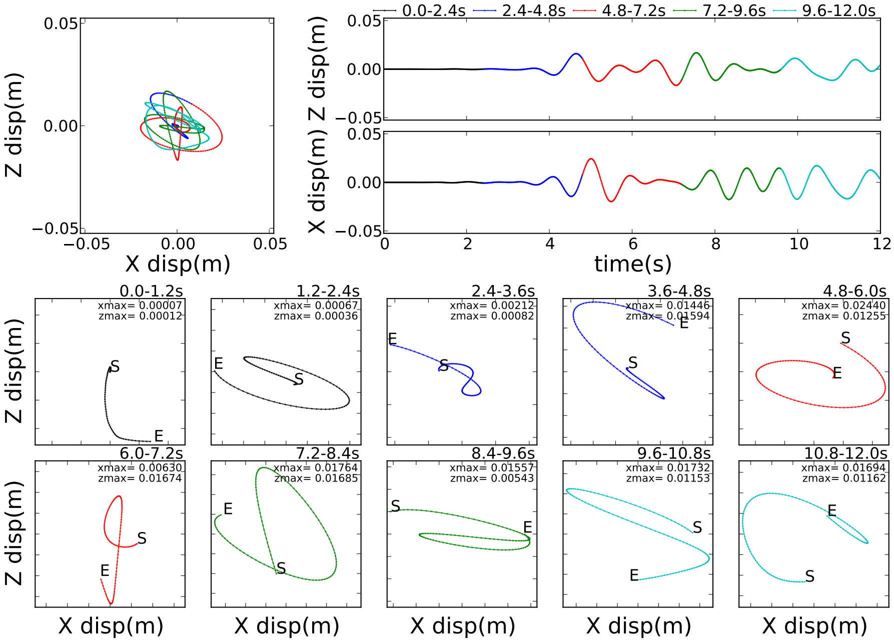


Figure 502.96: X array1 X=3800m, Z=5000m

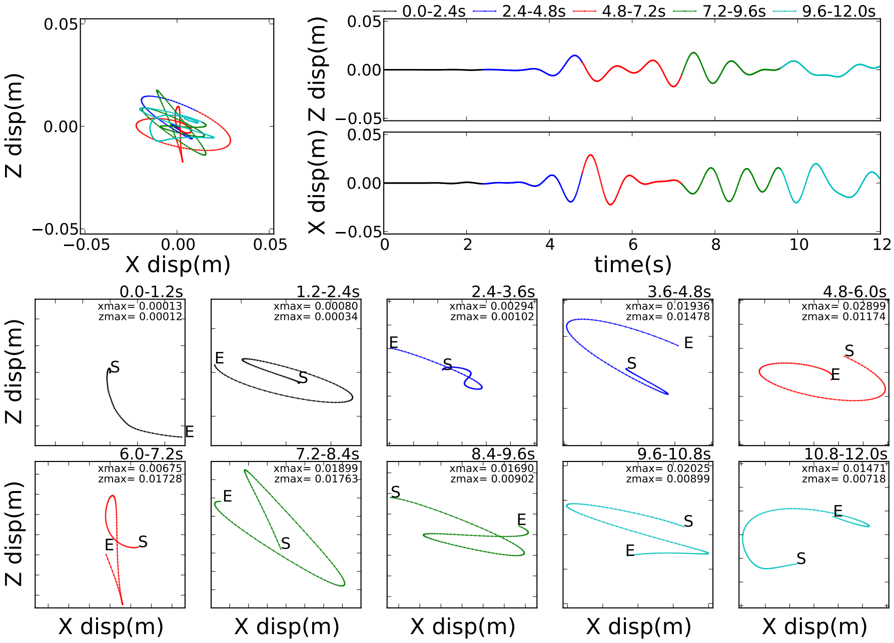


Figure 502.97: X array1 X=3600m, Z=5000m

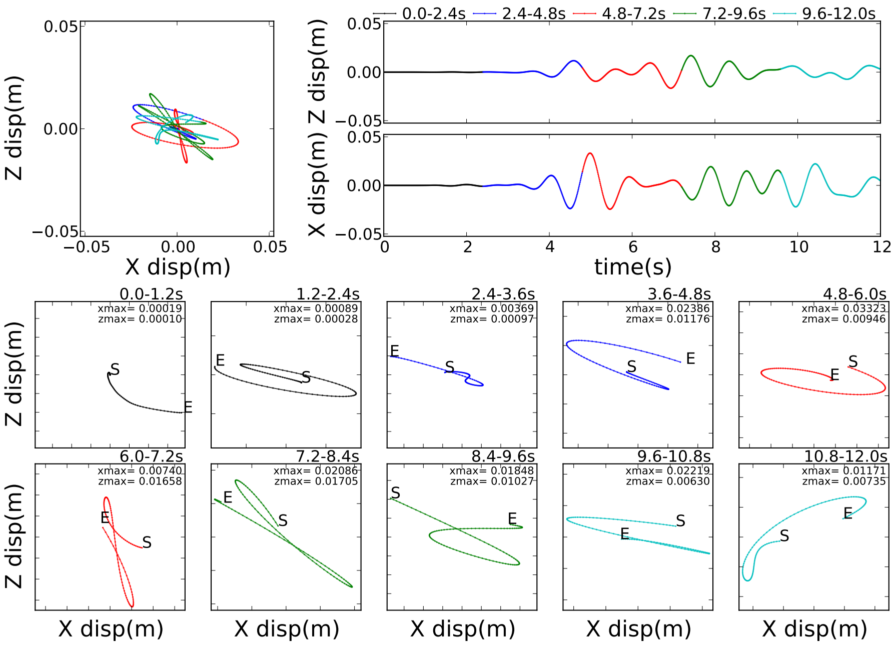


Figure 502.98: X array1 X=3400m, Z=5000m

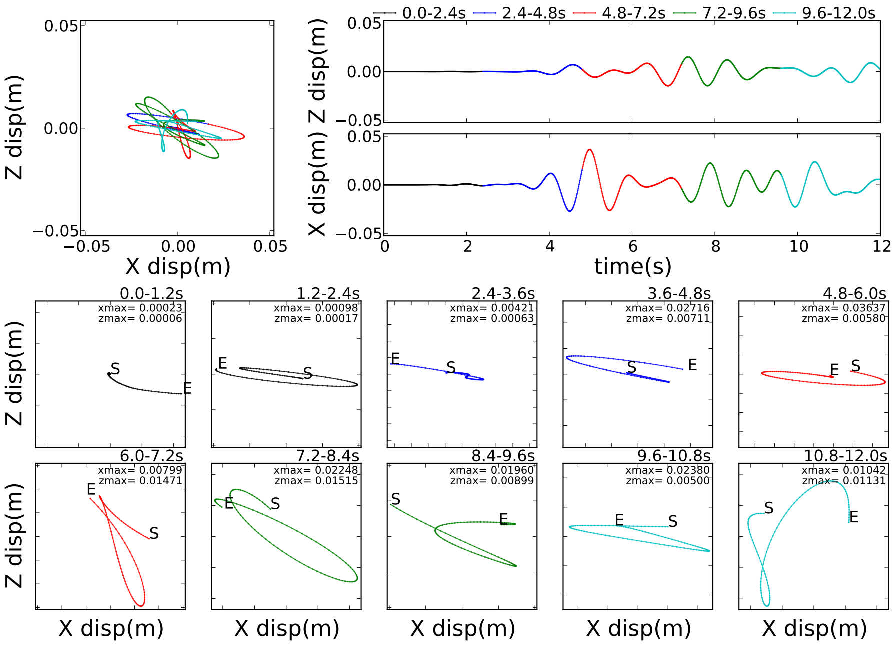


Figure 502.99: X array1 X=3200m, Z=5000m

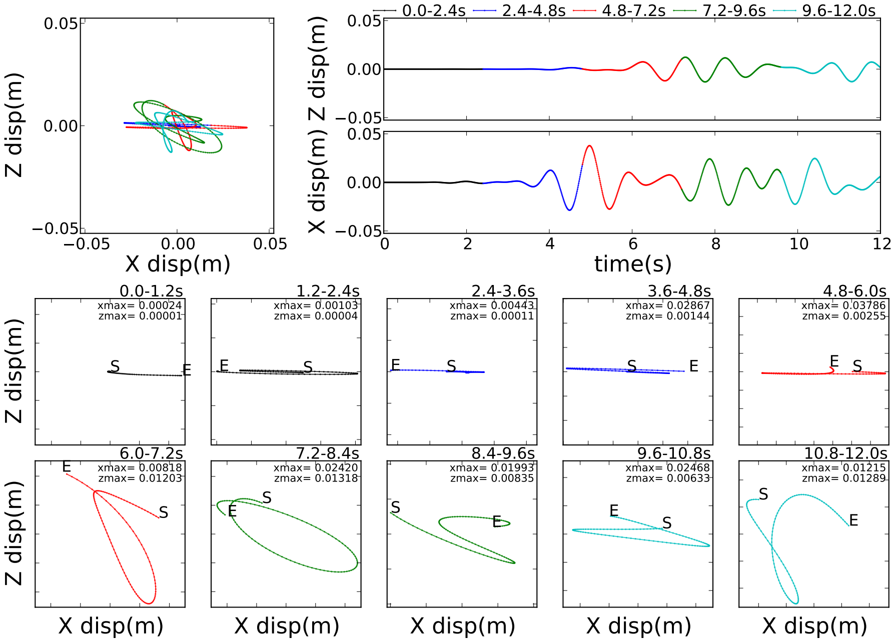


Figure 502.100: X array1 X=3000m, Z=5000m

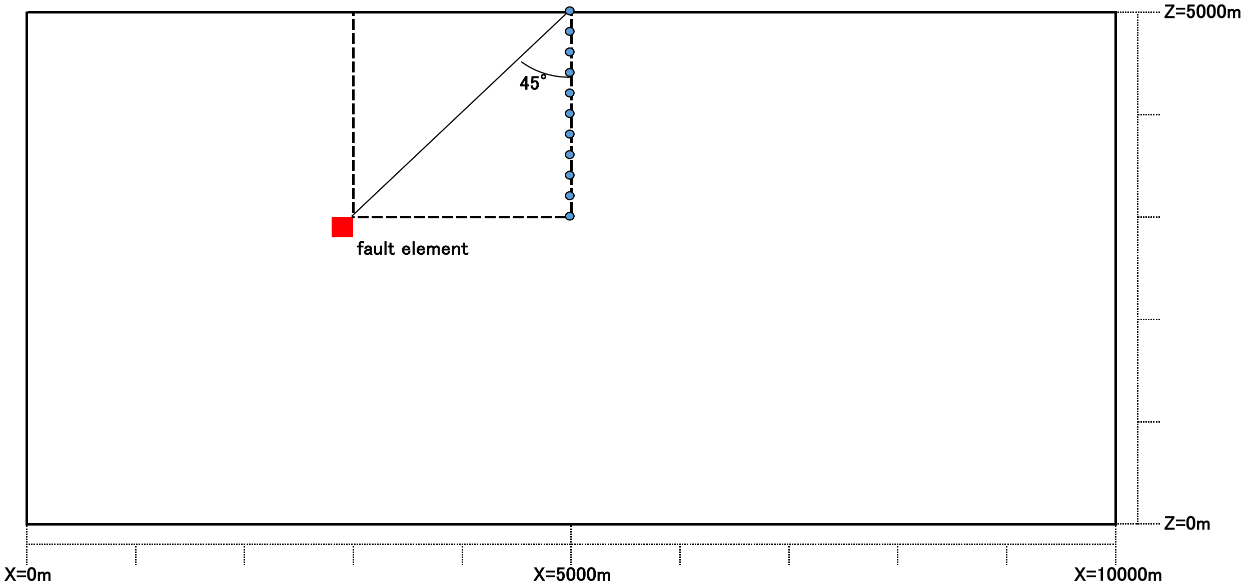


Figure 502.101: Observation array of Array5

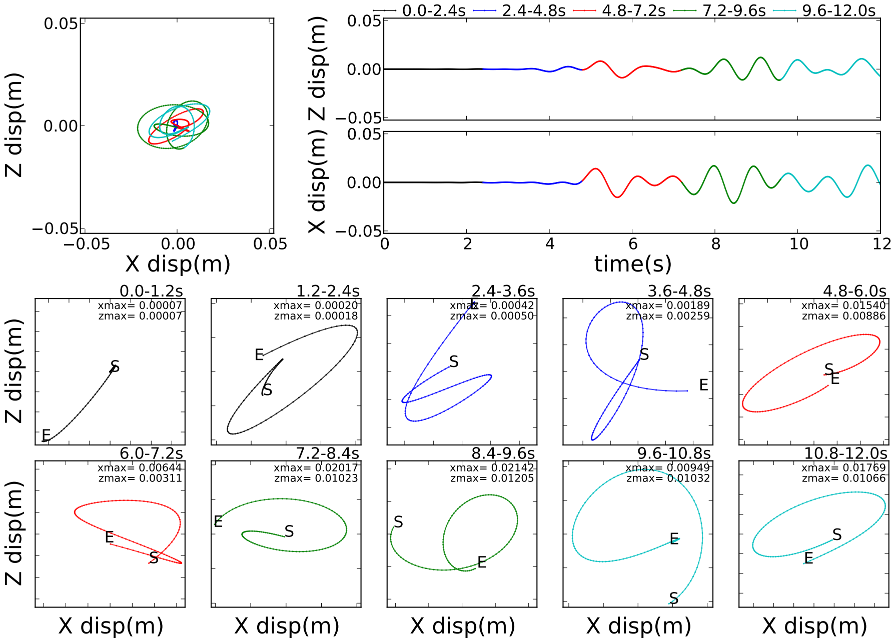


Figure 502.102: Z array1 X=5000m, Z=5000m

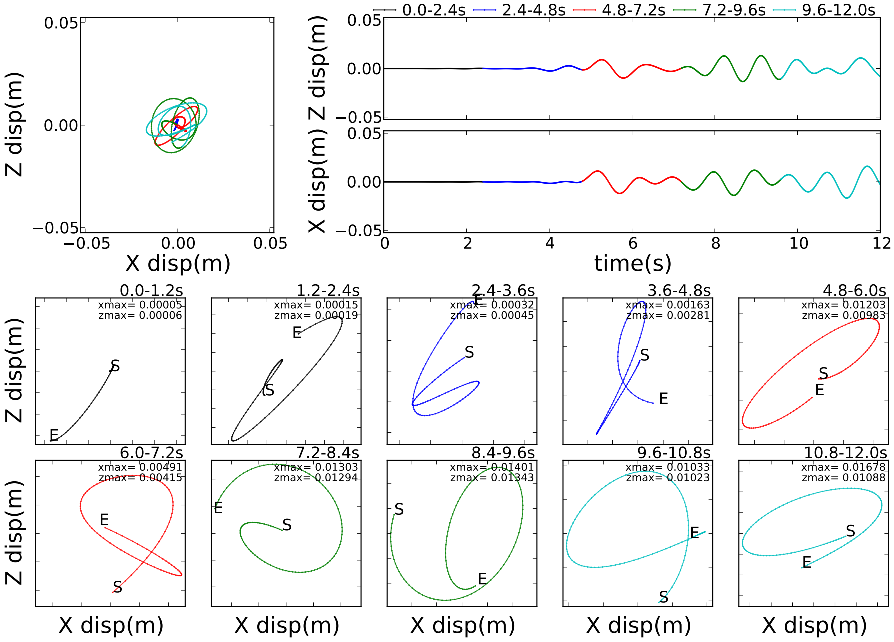


Figure 502.103: Z array1 X=5000m, Z=4800m

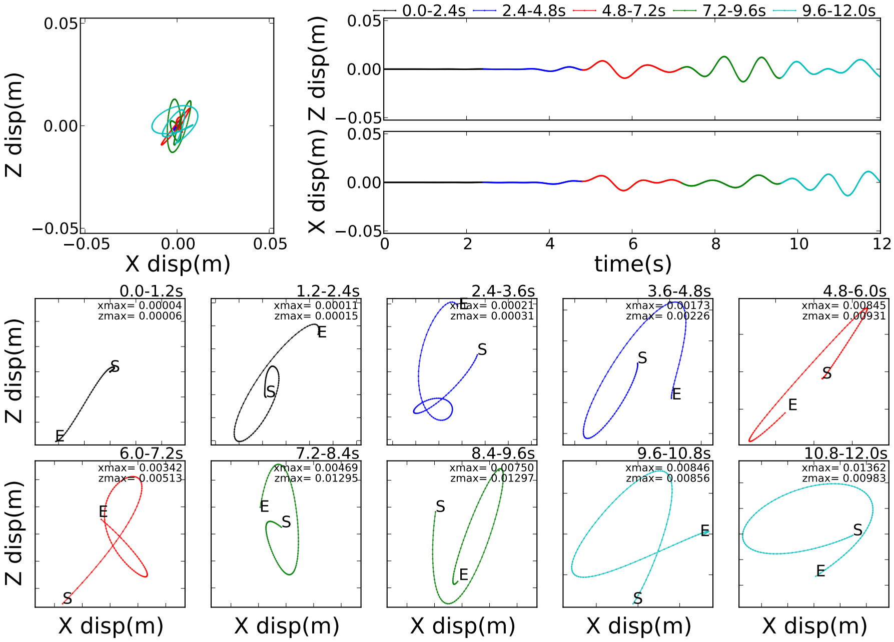


Figure 502.104: Z array1 X=5000m, Z=4600m

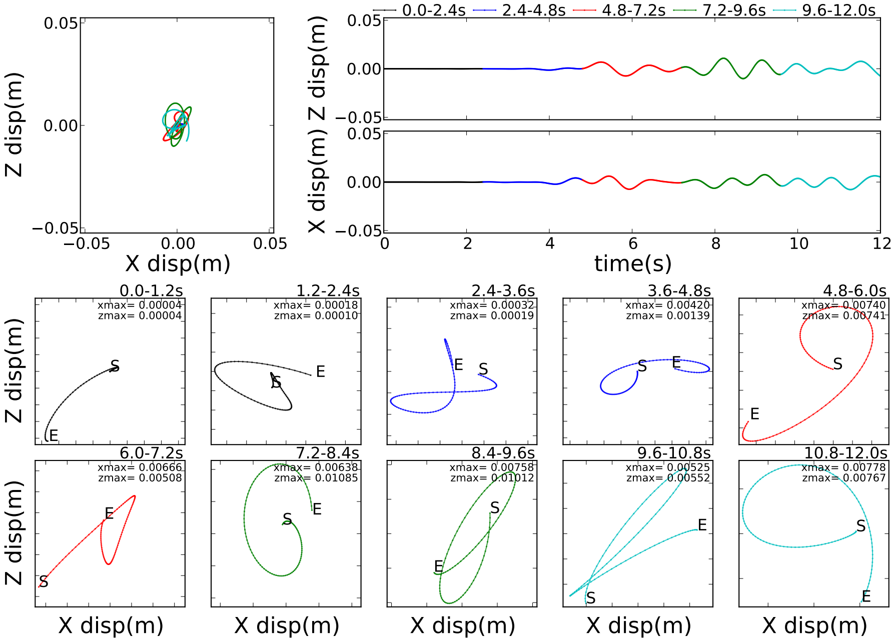


Figure 502.105: Z array1 X=5000m, Z=4400m

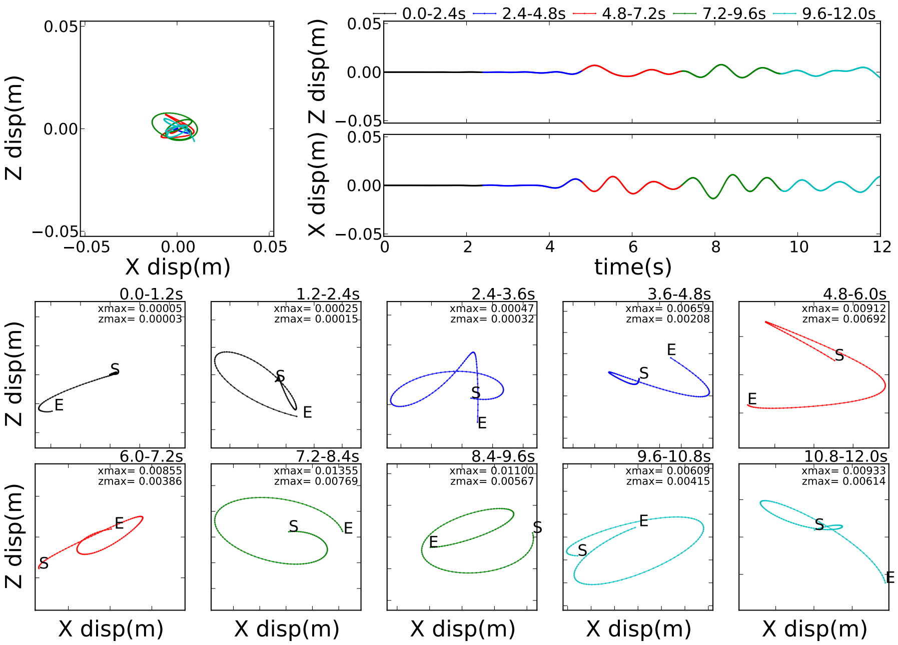


Figure 502.106: Z array1 X=5000m, Z=4200m

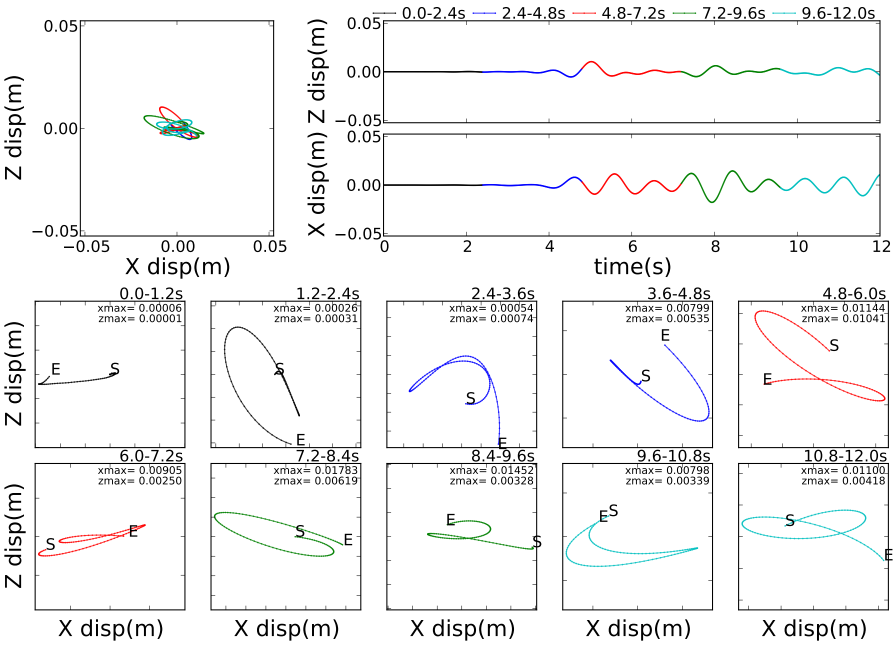


Figure 502.107: Z array1 X=5000m, Z=4000m

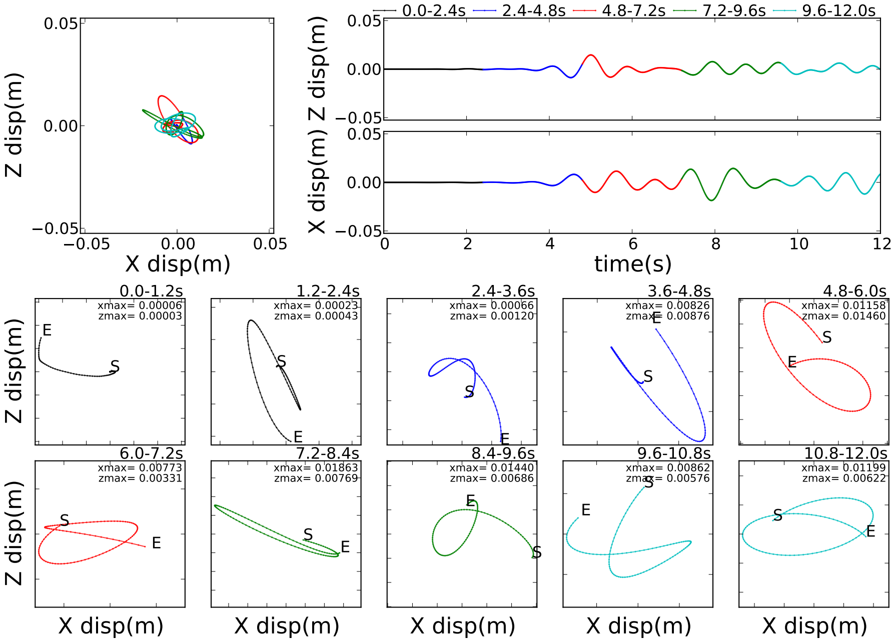


Figure 502.108: Z array1 X=5000m, Z=3800m

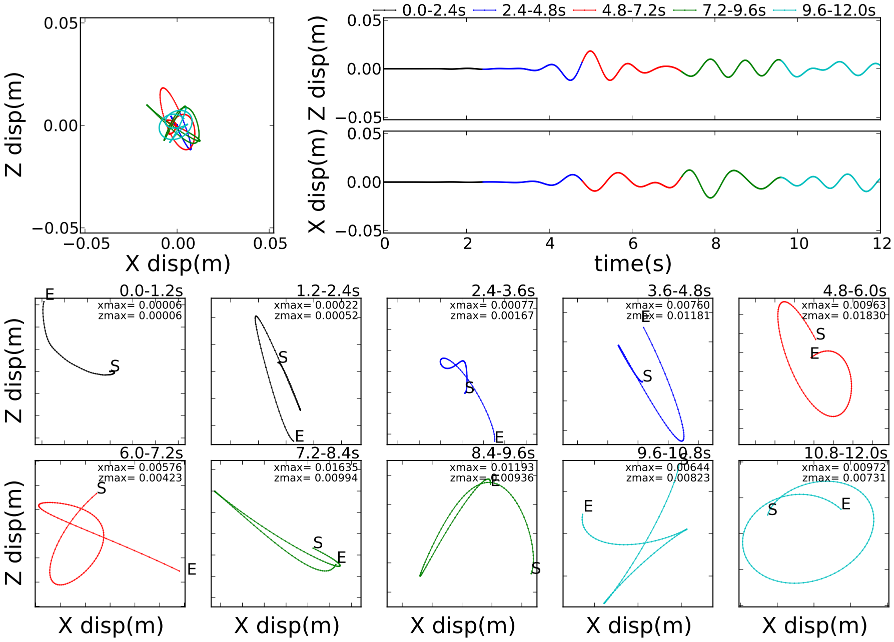


Figure 502.109: Z array1 X=5000m, Z=3600m

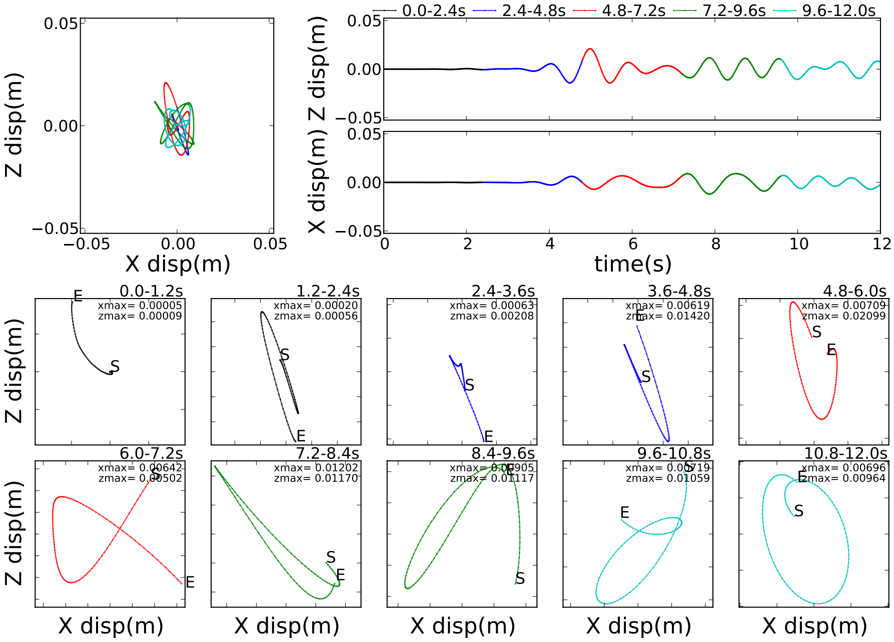


Figure 502.110: Z array1 X=5000m, Z=3400m

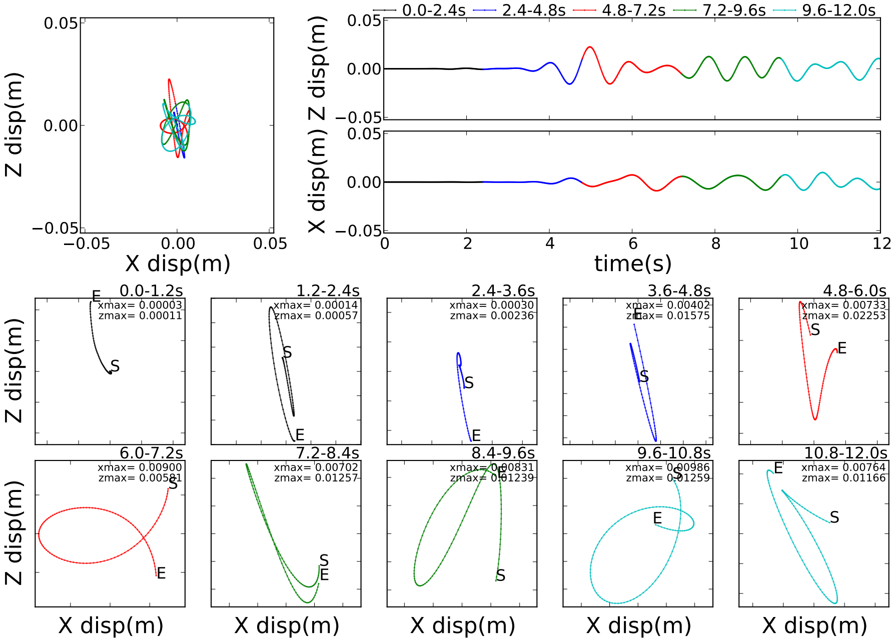


Figure 502.111: Z array1 X=5000m, Z=3200m

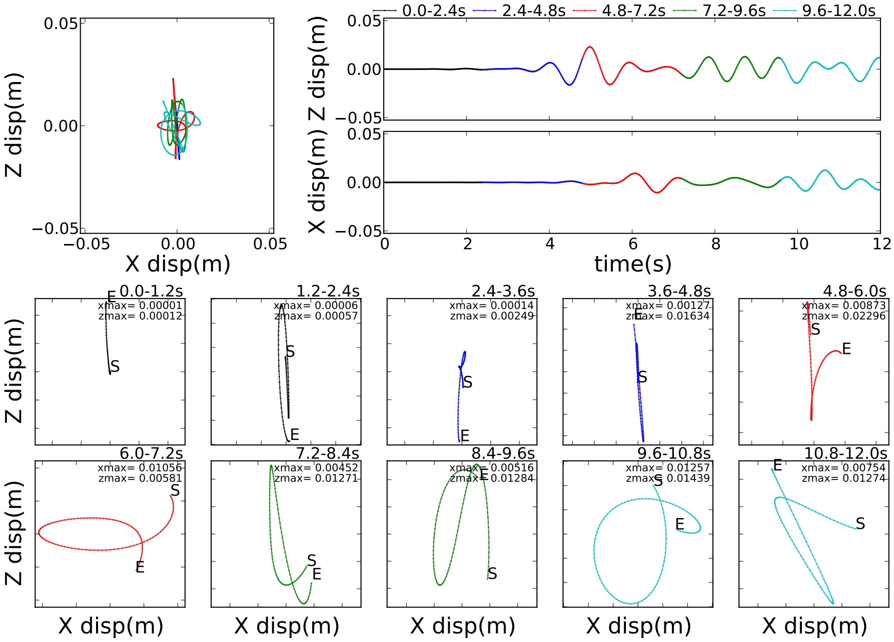


Figure 502.112: Z array1 X=5000m, Z=3000m

502.2.6.3 Animations of Fault Slip Motions

All the motions are developed from a point source at a depth of –2km while the distance from the center of the model (where the observation point, or object of interest is) is 1km ($27^{\rm deg}$ off vertical), 2km ($45^{\rm deg}$ off vertical) and 3km ($56^{\rm deg}$ off vertical). There are three surface soil cases, homogeneous, single layer of soft soil and two layers of soft soil. Stress drop (input, pure shear at the one element) is defined by an Ormsby wavelet.

- Homogeneous soil/rock, $56^{\rm deg}$ off vertical) (link to a movie, 39MB)
- Homogeneous soil/rock, $45^{\rm deg}$ off vertical) (link to a movie, 32MB)
- Homogeneous soil/rock, $27^{\rm deg}$ off vertical) (link to a movie, 37MB)
- Single layer soft soil with homogeneous soil/rock, $56^{\rm deg}$ off vertical) (link to a movie, 30MB)
- Single layer soft soil with homogeneous soil/rock, $45^{\rm deg}$ off vertical) (link to a movie, 34MB)
- Single layer soft soil with homogeneous soil/rock, $27^{\rm deg}$ off vertical) (link to a movie, 32MB)
- Two layers of soft soil with homogeneous soil/rock, $56^{\rm deg}$ off vertical) (link to a movie, 30MB)
- Two layers of soft soil with homogeneous soil/rock, $45^{\rm deg}$ off vertical) (link to a movie, 31MB)
- Two layers of soft soil homogeneous soil/rock, $27^{\rm deg}$ off vertical) (link to a movie, 32MB)

Details motions at the top of the model:

- Homogeneous soil/rock, $45^{\rm deg}$ off vertical) motions at the top 2km × 2km (link to a movie, 30MB)
- Homogeneous soil/rock, $45^{\rm deg}$ off vertical) motions at the very top, location of observation point and/or structure (link to a movie, 134MB)

502.2.6.4 Point Fault Slip Motions, Arrays and Particle Motions

The FEM model used is shown in Fig. 502.113. The brief description of FEM model is as follows:

- Model size: $10000\text{m} \times 20\text{m} \times 5000\text{m}$
- Mesh size: 10m
- Element size: 20m (27 node brick element is used)
- Elastic parameters:
  - Poisson ratio $\nu = 0.4$
  - Shear wave velocity $V_s = 2000\text{m/s}$
- Fixed boundary at $X = 0\text{m}$, $X = 10000\text{m}$, and $Z = 0\text{m}$
- Free boundary at $Z = 5000\text{m}$
- Plane strain condition in y-direction (all nodes on $y = 0\text{m}$, $y = 20\text{m}$ are fully fixed).

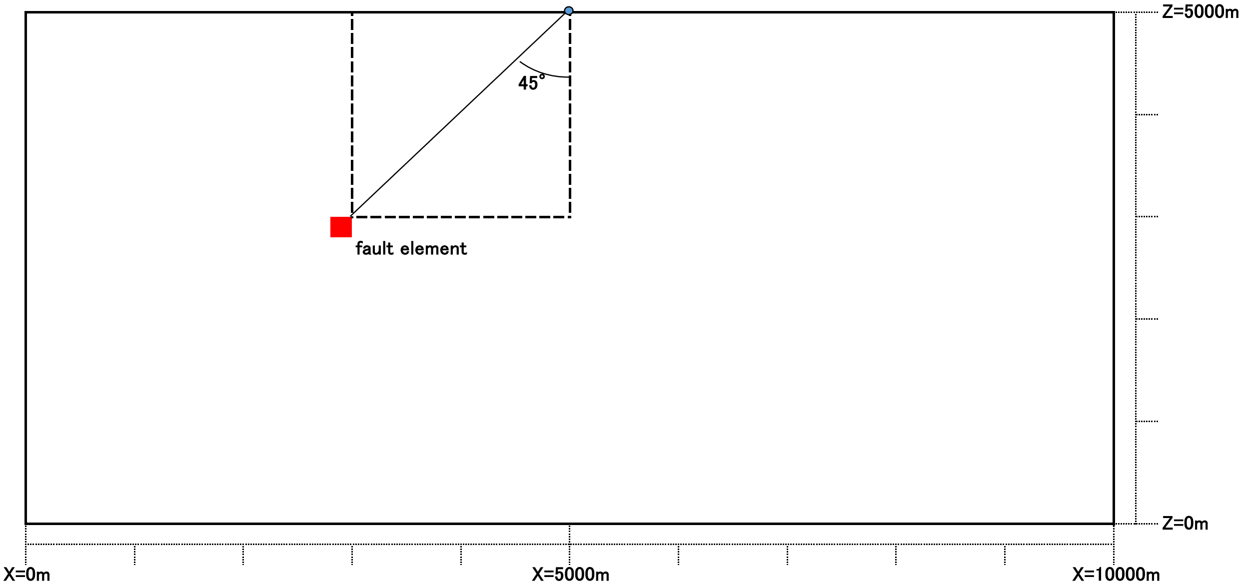


Figure 502.113: FEM model (red square:fault element, blue circle: observation site)

502.3 Dynamic Soil-Foundation-Structure Interaction

Theoretical details for this section are given in Section 109.3 on page 566.

502.3.1 Animation of the DRM on a 1D Stack of Elements

(link to a movie, 8.9MB)

502.3.2 Using External Finite-Diference Seismic Code for DRM Motions

Introduction This section explores the numerical conditions under which a high-performance fourth-order finite difference code for seismic modeling, henceforth the 'seismic' code, can be coupled successfully with a general purpose parallel non-linear finite element simulator, the FEM simulator, through the use of the domain reduction method (DRM). The approach taken consists in modeling a simple homogeneous half-space subjected to a single double-couple point-source to generate motions. DRM will be used to input these motions into the FEM simulator for an equivalent model of the domain, and the response compared at a control point on the surface. The seismic code used is SW4 (Sjögreen and Petersson, 2011) developed at Lawrence Livermore National labs, while the FEM simulator will be the Real-ESSI Simulator developed at the University of California Davis. Both are high-performance parallel programs highly regarded in their respective domains of application.

As originally proposed by Bielak et al. (2003a), the DRM input motions can be generated using a different method to compute the seismic wave field than the FEM code used to model site and structure. The rationale being that both methods will be approximating the same equations of elastodynamics and should both converge to the same solution as grid spacing tends to get smaller. What was implicit in that seminal work, but not explored or tested, is the effect of using different methods with possibly different orders of convergence, and how this affects convergence of the overall method.

The advantage of using different codes is that it is possible to choose a 'seismic modeling' code which is better suited and optimized for simulating earthquakes and then use the DRM to input the resulting motions into a code which is more suitable for modeling of non-linear soil and structural behavior in the chosen site. For example, SW4 has very convenient features for the input of double couple sources and also for extended sources, a task which would be much harder to achieve in a civil-engineering oriented code such as Real-ESSI Simulator. It is important to use the proper tools for modeling task, both for efficiency and credibility of the results used for design.

When solving the elastodynamic equations, it is expected that different solution schemes will yield different solutions for the same problem. Even if the seismic code's finite difference grid points coincide spatially with the FE nodes, the different mathematical transformations involved in advancing the solution and the different orders of approximation will no-doubt lead to some degree of disagreement on the nodal values. When using the seismic code as DRM input into the FEM code it is expected that this disagreement will manifest itself in two ways: first, that the solution at a common control point within the DRM domain and on the seismic domain will differ and, second, consequently there will be a portion of the wave-field that will not be absorbed at the DRM boundary leading to outgoing motions which need to be damped out. Again, it is expected that both the difference in motions and the outgoing

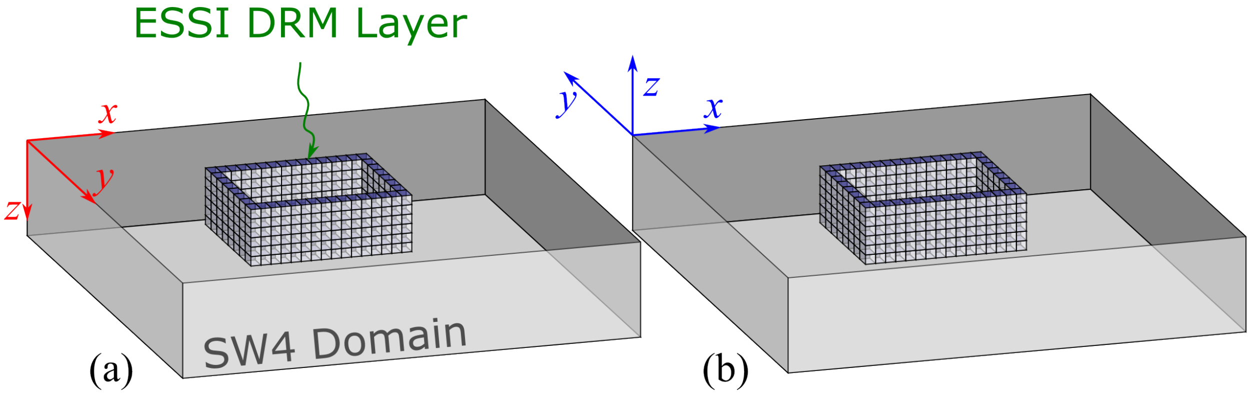


Figure 502.114: Model layout and coordinate system origin and orientation for (a) SW4 (b) Real-ESSI Simulator

wave-field will diminish with decreasing spatial and temporal discretization.

Format for Node Coordinates, for Direct Extraction of Motions from SW4 This is based on Rodgers (2017).

A list of DRM nodes need to be supplied to SW4 modeler.

This list should be a simple 4-column asci file with node-name (6-10 characters), x, y, z location relative to origin (top, center) of ESSI domain.

Example Figure 502.115 illustrates the general layout of the DRM boundary elements with respect to the SW4 model for the domain used in this study. The seismic model consists on a single material elastic box of size $8\,\mathrm{km} \times 4\,\mathrm{km} \times 4\,\mathrm{km}$, various values of the grid size are chosen ($h = 10\,\mathrm{m}$ or $20\,\mathrm{m}$), a point double-couple source is placed at $(2\,\mathrm{km},\ 2\,\mathrm{km}, 2\,\mathrm{km})$ such that it represents a reverse fault with a 45° dip. The center of both models coincide. SW4 uses a right-handed coordinate system with z-axis positive downwards, while the FE meshes developed[1] for Real-ESSI simulator use a right-handed system with z-axis upwards. Both models x-axis' and origins are made to coincide. Thus, the transformation from SW4 coordinates to Real-ESSI Simulator coordinates is:

$$x_{\mathrm{SW4}} = x_{\mathrm{ESSI}} \qquad y_{\mathrm{SW4}} = -y_{\mathrm{ESSI}} \qquad z_{\mathrm{SW4}} = -z_{\mathrm{ESSI}} \tag{502.1}$$

Figure 502.115 shows the construction of the DRM model used in this study. It consists of an internal domain (within the DRM boundary) of dimensions $200\,\mathrm{m} \times 200\,\mathrm{m} \times 40\,\mathrm{m}$ with discretization $\mathrm{d}x = \mathrm{d}y = \mathrm{d}z = h = 5\,\mathrm{m},\ 10\,\mathrm{m}$ or $20\,\mathrm{m}$. The elastic properties throughout both SW4 and Real-ESSI

---

[1] This is for convenience when using the meshing program gmsh to develop the mesh.

Table 502.1: Mesh information for DRM models used with Real-ESSI simulator.

h, m	Number of Nodes	Number of elements
20.0	28,577	3,200
10.0	78,141	9,000
5.0	295,829	35,000

Simulator domains are such that the speed of P-waves is $V_p = 2000\,\mathrm{m/s}$, the speed of S-waves is $V_s = 1000\,\mathrm{m/s}$ and the density is $\rho = 2000\,\mathrm{kg/m^3}$.

For the absorbent layer 4, 8 and 16 elements are used with Rayleigh damping ratios of $\xi = 0, 0.05, 0.1$. The Real-ESSI meshes are made up of second-order 27-node bricks with 27 Gauss-integration points. Table 502.1 summarizes the number of elements and nodes for the Real-ESSI Simulator meshes used.

The Real-ESSI meshes are generated using gmsh (Geuzaine and Remacle, 2009) and results are visualized using the custom visualizer plugin for ParaView (`https://www.paraview.org/`) (Ayachit, 2015).

Figure 502.116 shows the visualization of the displaced Real-ESSI mesh for $h = 20\,\mathrm{m}$ (left) and $h = 10\,\mathrm{m}$ (right) at time point $t = 3.40\,\mathrm{s}$. No damping was used at the absorbing boundary at this point. This instant in time corresponds to the end of the arrival of the P-wave and, it can be observed, that DRM is working to eliminate the outgoing motions as the absorbing boundary has little or no displacement. Furthermore, the $h = 10\,\mathrm{m}$ mesh seems to be doing a better job at absorbing the out-going motions. Figure 502.117 shows the same situation for time point $t = 3.93\,\mathrm{s}$. At this time the S-wave is coming into the domain. It can be seen that the $h = 20\,\mathrm{m}$ domain is producing large out-going motions while the finer domain is still handling the out-going motions.

Figures 502.116 shows the trace plots for the control point at the center of the domain for both SW4 results as well as the results with DRM when using a DRM domain with discretization $h = 20\,\mathrm{m}$ and $h = 10\,\mathrm{m}$ respectively.

P-waves are faster than S-waves, in this case, by a factor of 2. Since wave resolvability for a given time-step size is proportional to wave speed, P-waves will be better resolved than S-waves. This manifests itself in the fact that initially, for P-wave arrivals, DRM works at capturing out-going motions and later, for S-waves, some out-going motions escape the boundary. These out-going motions are not absorbed by damping since damping is not applied at this point, therefore, reflecting off the model boundaries back into the domain. This why there is oscillatory motion, corresponding to energy trapped in the system, observed after $t = 4\,\mathrm{s}$. Note that the oscillatory motion is mainly seen in the x-component and not the others.

To mitigate these trapped waves, the absorbent boundary is assigned Rayleigh damping of varying intensities. Rayleigh damping can be applied in many ways. From (Tafazzoli, 2012) several lessons can be drawn regarding how to design the absorbent boundary for maximum efficiency. The following is a summary of these lessons:

- Increasing in-absorbent-boundary damping has the general effect of reducing the amplitude of waves reflected at the domain boundary back into the internal domain.
- Sharp damping contrasts produce additional reflected waves at the DRM/absorbing boundary interface.
- Gradually increasing the damping ratio with increasing distance to the DRM boundary alleviates the reflection issue while retaining a similar damping efficiency.
- Increasing the thickness in elements of the absorbent boundary also increases the damping efficiency.
- When selecting two frequencies to provide for specification of Rayleigh damping, the best frequency is not related to the 'natural' frequency of the soil stratum as is commonly assumed in practice and some research. The best frequency is related, instead, to the frequency of the out-going waves.

In this study a uniform value for the damping ratio assigned to the Rayleigh damping coefficients is used. Figure 502.119 shows the effect that this additional damping has on DRM models with different sizes. The beneficial effect of this damping in capturing the energy leaking out from the DRM boundary is apparent.

Perfect matching of the motions obtained with SW4 and Real-ESSI simulation with DRM modeling was not achieved. A key reason for this is that the SW4 simulations were done at $h = 20\,\mathrm{m}$ while varying the mesh size for the Real-ESSI simulations. This means that SW4 motions had to be interpolated between grid spaces when the grids did not match. SW4 provides only 'nearest' neighbor interpolation for requested output stations, so an improvement on this is needed if better matching is expected. Alternatively, it is possible to achieve better agreement by decreasing the discretization on both domains.

From these experiments it is possible to extract the following design considerations when seeking to couple DRM-based finite element simulations with other forms of seismic modeling:

- Size of the DRM domain is irrelevant for the 'free-field' problem.
- Relative order of accuracy of the Finite-element mesh and the code that produces seismic input is important when designing mesh sizes for both the finite-element and the seismic simulation.

- Matching motions perfectly might result in expensive computations, beyond what is needed due to physical and numerical constraints for the propagation problem alone.
- Out-going motions due to mismatch need to be absorbed outside the DRM domain by some method.

Ultimately, the purpose of using the DRM is as the enabling technology allowing the rational modeling of perturbations of the free-field model in order to reduce the cost of jointly modeling complex site and structure response along with the seismic wave propagation problems. In such a case, out-going motions will be unavoidable and have to be dealt with efficiently. In any such study it will be very important to demonstrate that the DRM motions and forces developed agree with the free-field model as a basic way of showing the adequacy of the numerical implementations involved in the modeling effort.

Practical considerations Using SW4 to generate motions for DRM in Real-ESSI requires the following steps:

1. Generate a large-scale geologic model in SW4.
2. Generate a FEM mesh for Real-ESSI, generate the following lists:
    - Coordinates and numbers of nodes in DRM layer.
    - A flag indicating whether a particular node is internal or external.
    - List of elements in DRM layer.
3. Within the SW4 input file write recorder lines (USGS format) for all DRM nodes. Example:

```
 rec x=4037.5 y=2050.0 z=40.0 file=node000734b sacformat=0 usgsformat=1
```

4. Run SW4.
5. Collect the results for each recorded node in one DRM HDF5 input file.

Some notes to complement the above steps.

- Nodes in SW4 input file must have coordinates in SW4 coordinates. Remember that, in SW4, down is positive. The best is to develop FEM model centered around origin of a coordinate system, so that determining the position in SW4 coordinates is a simple translation and flipping of the signs for both Z and Y coordinates.

- Use the SW4 output file name in a meaninful way to determine which node number it belongs to. In the example `nodeXXXY` where `XXX` has the node number and `Y` is either `e` or `b` depending if it is an internal (boundary) node or external.
- Remember to reverse the sign of Y and Z results when writing the DRM HDF5 input file from the SW4 results.
- It is recommended to use chunking and compression in HDF5 dataset to speed up loading times and optimize storage usage.

Template code for generating HDF5 input suitable for DRM analysis in Real-ESSI.

need to move the code here and not have it in Figure-files!!!!

```
import h5py
import scipy as sp
import time

#Global parameters
Ntimesteps = 1000
dt = 0.0001
Nnodes = 10000
Nelements = 4000

#Create HDF5 file - note filename
h5file = h5py.File("earthquake.h5.drminput","w")


#Initialize memory
u = sp.zeros((3*Nnodes, Ntimesteps), dtype = sp.double) #Will hold dispalcements
a = sp.zeros((3*Nnodes, Ntimesteps), dtype = sp.double) #Will hold accelerations

nodelist = sp.zeros(Nnodes,dtype=sp.int32)
is_boundary_node = sp.zeros(Nnodes,dtype=sp.bool_)
elements = sp.zeros(Nelements,dtype=sp.int32)

#Time vector
t = sp.linspace(0,dt*(Ntimesteps-1),dt)

# ========================
# ========================
# ========================
# * Read in u from SW4 output, determine v and a (differentiate).
#
# * Remember to flip signs for Y and Z components.
#
# * Also read in:
# - is_boundary_node (boolean vector, see above)
# - nodelist (integer vector, see above)
```

```
# - elements (integer vector, see above)
#
# This is problem and formatting specific and likely to change depending on many
# factors.
# ========================
# ========================
# ========================

# Count numbers of nodes (DRM input file needs this)
Nb = 0
Ne = 0
for i in range(Nnodes):
    if is_boundary_node[i] == True:
        Nb += 1
    else:
        Ne += 1


#Write out HDF5 file.

h_acc = h5file.create_dataset("Accelerations", (3*Nnodes,Ntimesteps), ↩
    dtype=sp.double, data=a)
h_dis = h5file.create_dataset("Displacements", (3*Nnodes,Ntimesteps), ↩
    dtype=sp.double, data=u)

h5file.create_dataset("Time", data=t)
h5file.create_dataset("Elements", data=elements)
h5file.create_dataset("DRM Nodes", data=nodelist)
h5file.create_dataset("Is Boundary Node", data=is_boundary_node, ↩
    dtype=sp.int32) #This array has 1 if the node at the corresponding position ↩
    in "DRM nodes" array is a boundary node and zero if not
h5file.create_dataset("Number of Exterior Nodes", data=Ne)
h5file.create_dataset("Number of Boundary Nodes", data=Nb)

#For big cases, it is better to do this one record at a time.

# #Write timestamp (time format used is that of c "asctime" Www Mmm dd hh:mm:ss ↩
    yyyy example: Tue Jan 13 10:17:09 2009)
localtime = time.asctime( time.localtime(time.time()) )
h5file.create_dataset("Created",data=str(localtime))

#Close HDF5
h5file.close()
```

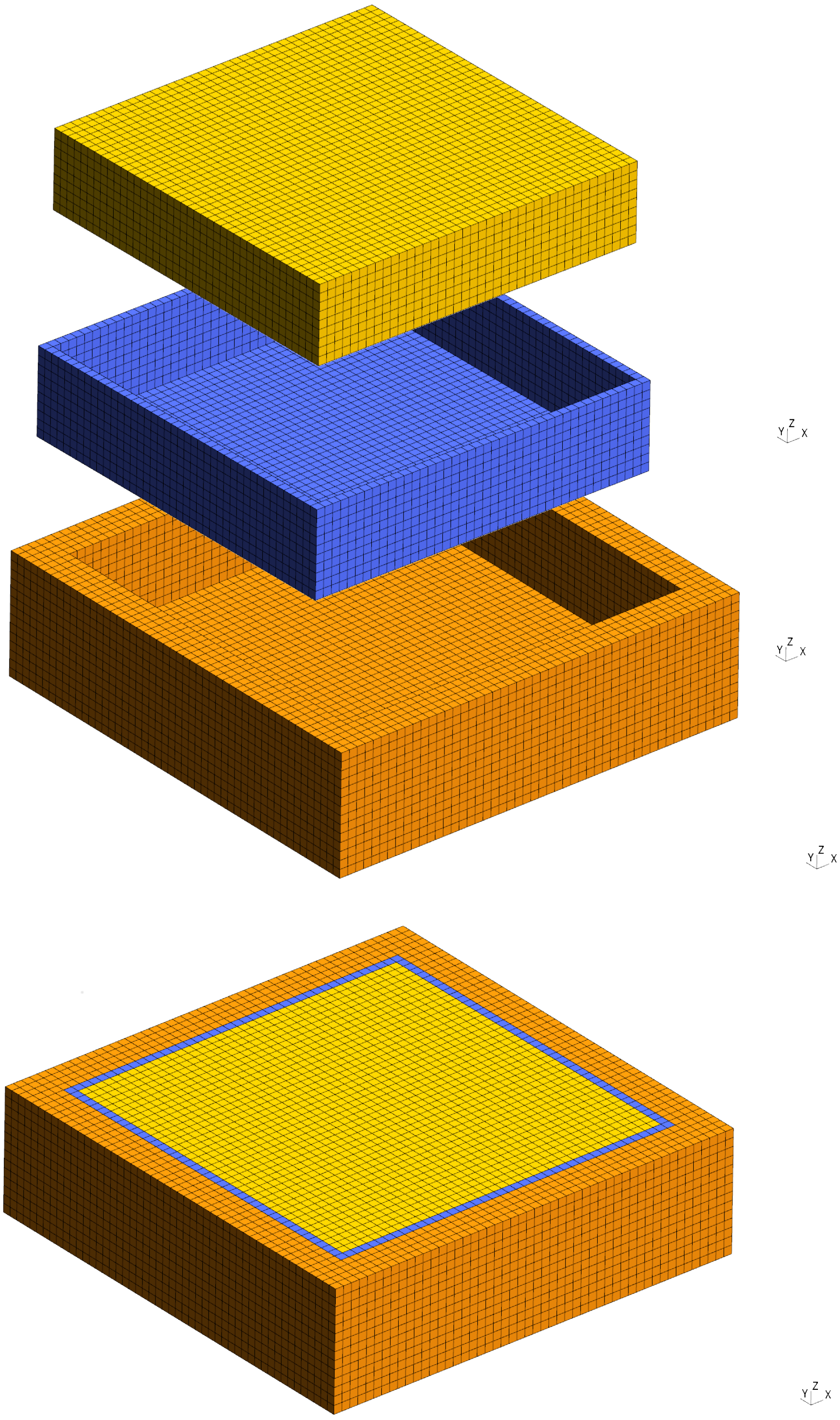


Figure 502.115: Free field DRM model for verification of SW4 and Real-ESSI Simulator coupling. From top, internal domain, DRM boundary, absorbent element layer, complete model.

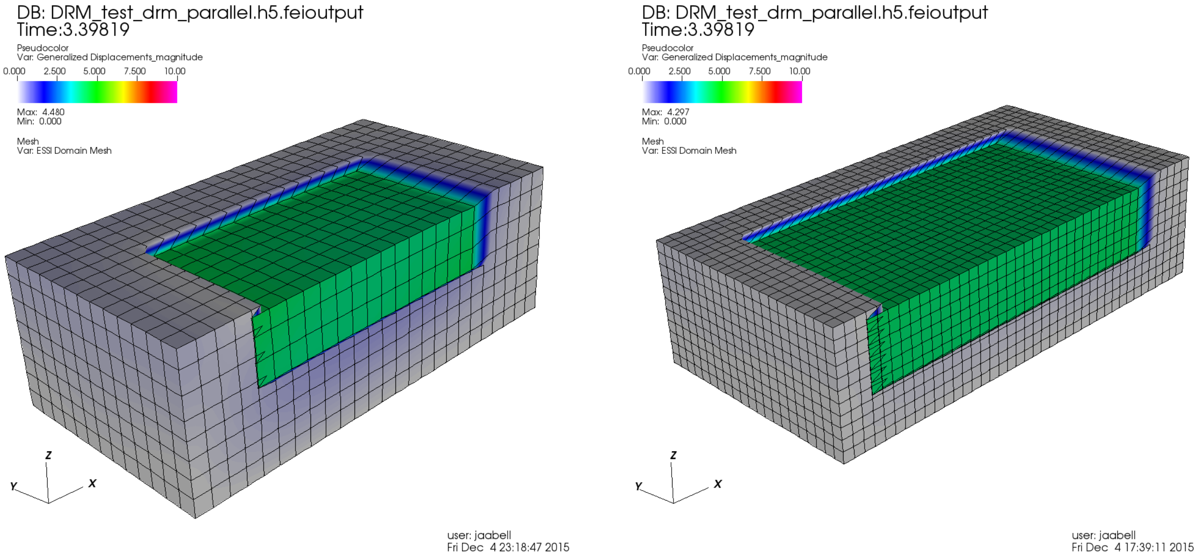


Figure 502.116: Visualization of the DRM solution at time $t = 3.40\,\mathrm{s}$ for DRM mesh sizes of $h = 20\,\mathrm{m}$ (left) $10\,\mathrm{m}$ (right), SW4 mesh is $h = 20\,\mathrm{m}$ for both cases. Color shows magnitude of displacement vector. Arrival of P wave is correctly resolved on both meshses.

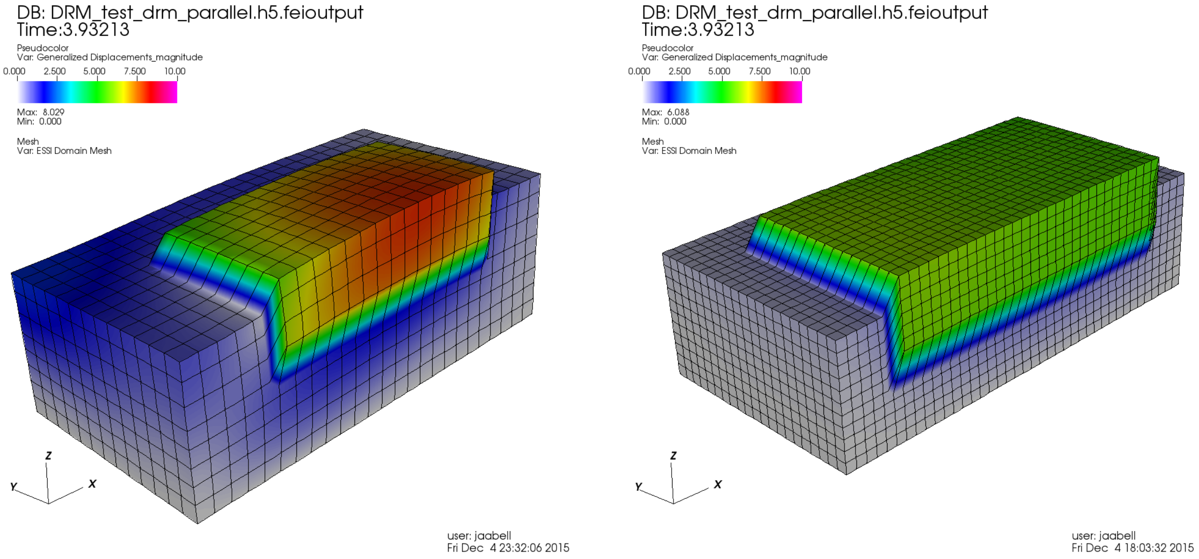
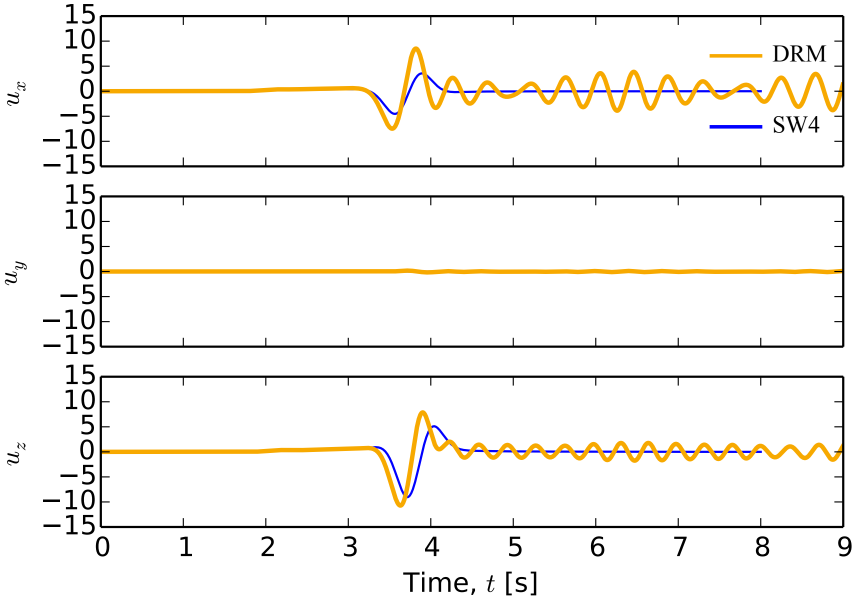
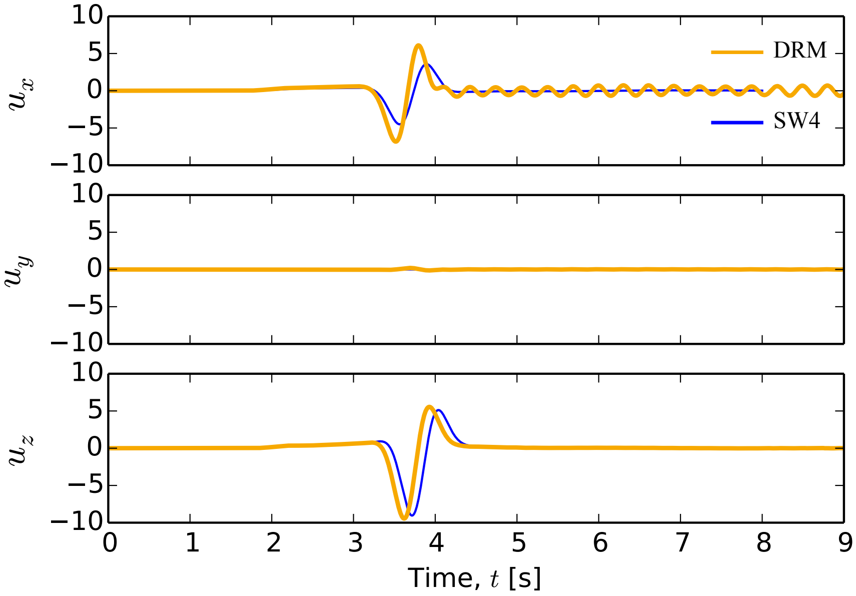


Figure 502.117: Visualization of the DRM solution at time $t = 3.93\,\mathrm{s}$ for same setup as Figure 502.116, with color showing magnitude of the displacement vector. Arrival of S wave is better resolved on the $h = 10\,\mathrm{m}$ mesh but not on the $h = 20\,\mathrm{m}$ mesh as can be seen by looking at the out-going motions. SW4 mesh is $h = 20\,\mathrm{m}$ for both cases



$20m$ grid spacing



$10m$ grid spacing

Figure 502.118: Generated motions at the center node ($x = 4000m$, $y = 2000$, $z = 0$), blue blue shows SW4 motions generated with an $h = 2\,\mathrm{m}$ grid, and myorange orange shows motions obtained with Real-ESSI using DRM and variable mesh size.

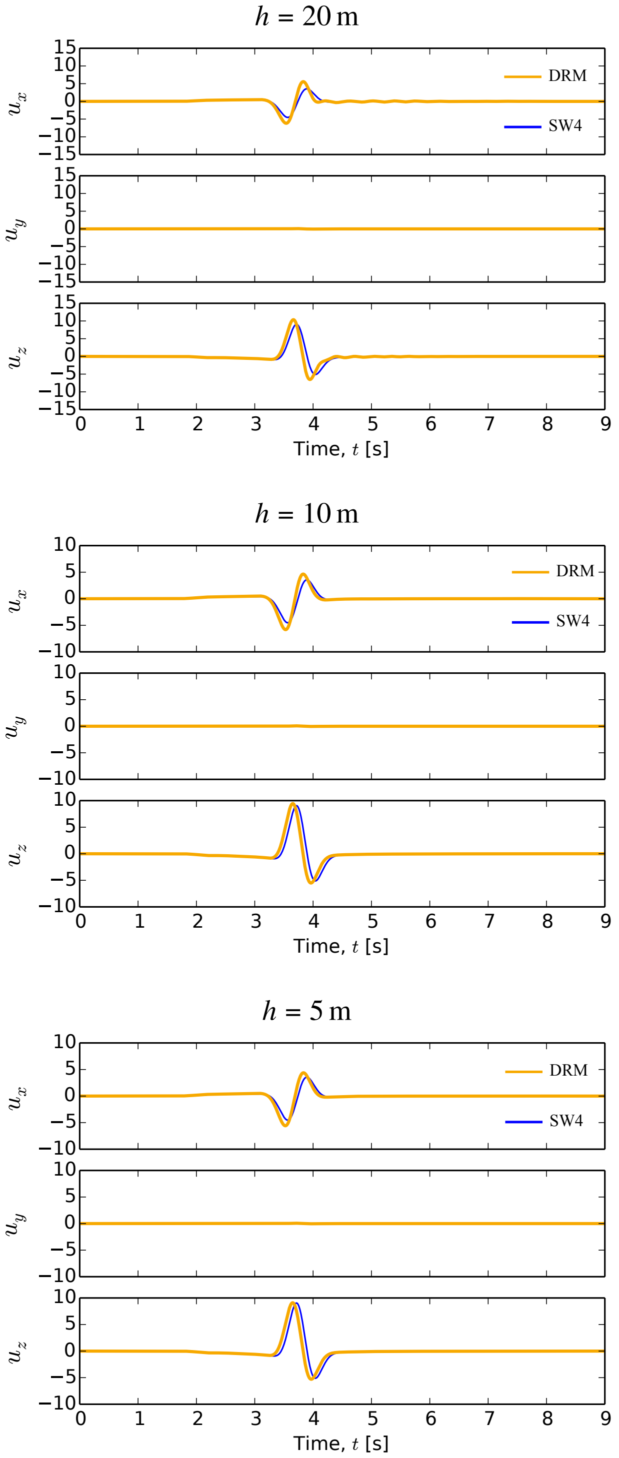


Figure 502.119: Results from computing motions with Real-ESSI Simulator within the DRM domain for different ESSI mesh sizes h and Rayleigh damping with $\xi = 0.1$. All SW4 motions were computed for $h = 20\,\mathrm{m}$

502.3.3 Seismic Wave Propagation Modeling and Simulation: Numerical Accuracy and Stability

The accuracy of a numerical simulation of dynamic SFSI is controlled by two main parameters: a) the spacing of the nodes of the finite element model (Δh) and b) the length of the time step Δt. Assuming that the numerical method converges toward the exact solution as Δt and Δh go toward zero the desired accuracy of the solution can be obtained as long as sufficient computational resources are available.

502.3.3.1 Grid Spacing Δh

In order to represent a traveling wave of a given frequency accurately about 10 nodes per wavelength λ are required for finite element with linear displacement interpolations between nodes[2] (Bathe and Wilson, 1976; Hughes, 1987; Argyris and Mlejnek, 1991). Fewer than 10 nodes can lead to numerical damping as the discretization misses certain peaks of the wave. In order to determine the appropriate maximum grid spacing the highest relevant frequency f_{max} that is present in the model needs to be found by performing a Fourier analysis of the input motion. Typically, for seismic analysis f_{max} is about 10 Hz. By choosing the wavelength $\lambda_{min} = v/f_{max}$, where v is the wave velocity, to be represented by 10 nodes the smallest wavelength that can still be captured partially is $\lambda = 2\Delta h$, corresponding to a frequency of $5f_{max}$. This is true for finite elements with linear interpolation of displacements.

In general, for finite elements with different interpolation of displacements the following conclusion can be made about the number of elements used for proper wave propagation:

- Linear interpolation finite elements (1D 2-node truss, 2D 4-node quad, 3D 8-node brick): $h^{LE} \leq v/(10 f_{max})$
- Quadratic interpolation finite elements (1D 3-node truss, 2D 9-node quad, 3D 27-node brick): $h^{QE} \leq v/(2 f_{max})$
- Structural elements (beams, shells) have at least quadratic interpolation functions (Euler-Bernoulli beam uses cubic Hermite polynomials) and since material is very stiff, and wave propagation speed is very high, these elements usually satisfy wave propagation criteria a priori.

For example, element size for propagating $f_{max} = 20$ Hz, for for linear interpolation element $\Delta h^{LE} \leq v/200$ Hz while for quadratic interpolation element $\Delta h^{QE} \leq v/40$ Hz. When material plastifies, element size needs reduction, depending on the reduction in (shear) wave velocity.

---

[2] If quadratic finite elements are use, for example a 27 node brick than only two elements are needed per wave length as they can properly represent the single wave (as they have quadratic interpolation for displacements, and feature 5 along the wave).

Nonlinear, Inelastic, Elastic-Plastic Material: It is very important to note that recommended grid spacing changes if material stiffness changes. In other words, for softer material grid spacing is to be be reduced. This means that for elastic-plastic material, when material plastifies the element size should be reduced. Detailed analysis if grid spacing, finite element size for elastic-plastic material is provided by Watanabe et al. (2017).

502.3.3.2 Time Step Length Δt

The time step Δt used for numerically solving nonlinear vibration or wave propagation problems has to be limited for two reasons. The stability requirement depends on the numerical procedure in use and is usually formulated in the form $\Delta t / T_n <$ value. T_n denotes the smallest fundamental period of the system. Similar to the spatial discretization T_n needs to be represented by about 10 time steps. While the accuracy requirement provides a measure on which higher modes of vibration are represented with sufficient accuracy, the stability criterion needs to be satisfied for *all* modes. If the stability criterion is not satisfied for all modes of vibration, then the solution may diverge. In many cases it is necessary to provide an upper bound to the frequencies that are present in a system by including frequency dependent damping to the model.

The second stability criterion results from the nature of the finite element method. As a wave front progresses in space it reaches one point after the other. If the time step in the finite element analysis is too large the wave front can reach two consecutive elements at the same moment. This would violate a fundamental property of wave propagation and can lead to instability. The time step therefore needs to be limited to

$$\Delta t \; < \; \frac{\Delta h}{v} \tag{502.2}$$

where v is the highest wave velocity.

502.3.3.3 Nonlinear Material Models

If nonlinear material models are used the considerations for stability and accuracy as stated above don't necessarily remain valid. Especially modal considerations need to be examined further for these cases. It is however save to assume that the natural frequencies decrease as plastic deformations occur. The minimum time step required to represent the natural frequencies of the dynamic system can therefore taken to be the same as in an elastic analysis.

A high frequency component is introduced due to plastic slip and counter balancing of the resulting displacement. This is especially true if a linear algorithm with no iterations within one time step is used. Figure 502.120 shows a part of an acceleration time history from an analysis involving elastic-plastic

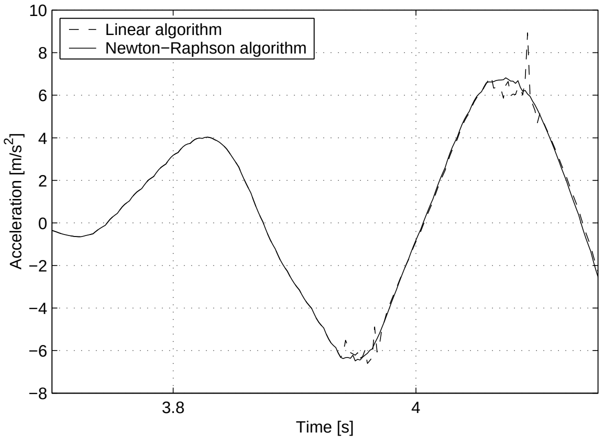


Figure 502.120: Resulting acceleration using Linear and Newton-Raphson algorithms

material. It can be seen that the out-of-balance forces at the end of a time step can be quite large if a linear algorithm is used. While the Newton-Raphson algorithm minimizes out-of-balance forces within one time step the linear algorithm requires several time steps to return to a stable equilibrium path.

The frequencies corresponding to these peaks are typically of the order of $1/(\text{a few}\,\Delta t)$. Normally the time step is small enough so that these frequencies don't interfere with the input motion. They can be prevented from propagating through the model by an appropriate choice of algorithmic or material damping.

For stability the time step used in a nonlinear analysis needs to be smaller than in a linear elastic analysis. By how much it has to be reduced is difficult to predict as this depends on many factors such as the material model, the applied loading or the numerical method itself. Argyris and Mlejnek (1991) suggest the time step to be reduced by 60% or more compared to the time step used in an elastic analysis. The best way to determine whether the time step is appropriate for a given analysis consists in running a second analysis with a reduced time step.

502.3.4 Seismic Wave Propagation Modeling and Simulation: Domain Boundaries

One of the biggest problems in dynamic SFSI in infinite media is related to the modeling of domain boundaries. Because of limited computational resources the computational domain needs to be kept small enough so that it can be analyzed in a reasonable amount of time. By limiting the domain however an artificial boundary is introduced. As an accurate representation of the soil-structure system this boundary has to absorb all outgoing waves and reflect no waves back into the computational domain. The most commonly used types of domain boundaries are presented in the following:

- Fixed or free

  By fixing all degrees of freedom on the domain boundaries any radiation of energy away from the structure is made impossible. Waves are fully reflected and resonance frequencies can appear that don't exist in reality. The same happens if the degrees of freedom on a boundary are left 'free', as at the surface of the soil.

  A combination of free and fully fixed boundaries should be chosen only if the entire model is large enough and if material damping of the soil prevents reflected waves to propagate back to the structure.

- Absorbing Lysmer Boundaries

  A way to eliminate waves propagating outward from the structure is to use Lysmer boundaries. This method is relatively easy to implement in a finite element code as it consists of simply connecting dash pots to all degrees of freedom of the boundary nodes and fixing them on the other end (Figure 502.121).

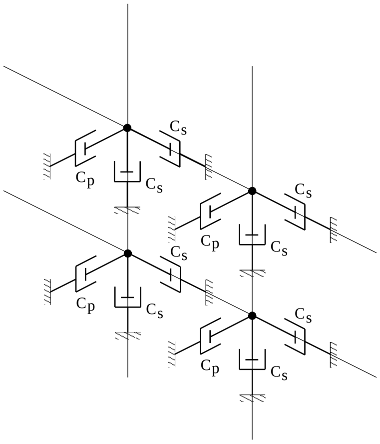


Figure 502.121: Absorbing boundary consisting of dash pots connected to each degree of freedom of a boundary node

Lysmer boundaries are derived for an elastic wave propagation problem in a one-dimensional semi-infinite bar. It can be shown that in this case a dash pot specified appropriately has the same dynamic properties as the bar extending to infinity (Wolf, 1988). The damping coefficient C of the dash pot equals

$$C = A \rho c \tag{502.3}$$

where A is the section of the bar, ρ is the mass density and c the wave velocity that has to be selected according to the type of wave that has to be absorbed (shear wave velocity c_s or compressional wave velocity c_p).

In a 3d or 2d model the angle of incidence of a wave reaching a boundary can vary from almost 0° up to nearly 180°. The Lysmer boundary is able to absorb completely only those under an angle of incidence of 90°. Even with this type of absorbing boundary a large number of reflected waves are still present in the domain. By increasing the size of the computational domain the angles of incidence on the boundary can be brought closer to 90° and the amount of energy reflected can be reduced.

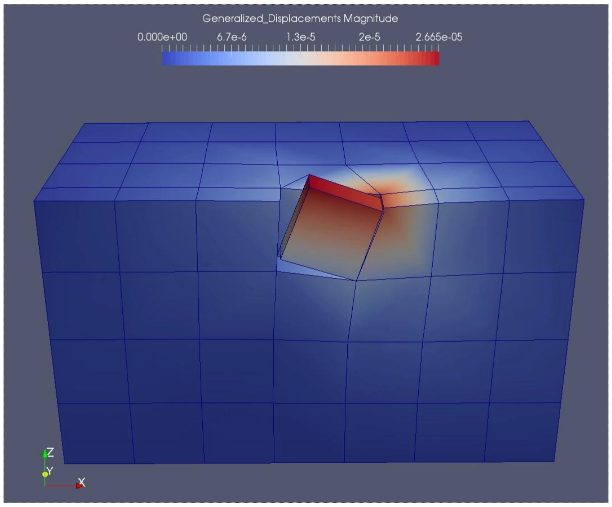
- Infinite elements
- More sophisticated boundaries modeling wave propagation toward infinity (boundary elements)

For a spherical cavity involving only waves propagating in radial direction a closed form solution for radiation toward infinity, analogous to the Lysmer boundary for wave propagation in a prismatic rod, exists (Sections 3.1.2 and 3.1.3 in Wolf (1988)). Since this solution, in contrast to the Lysmer boundary, includes radiation damping it can be thought of as an efficient way of eliminating reflections on a semi-spherical boundary surrounding the computational domain.

More generality in terms of absorption properties and geometry of the boundary are provided by the various boundary element methods (BEM) available in the literature.

502.3.5 Soil/Rock Modeling and Simulation

502.3.6 Soil/Rock – Foundation Contact (Slipping and Gaping) Modeling and Simulation



502.3.7 Buoyancy Modeling and Simulation

For self weight and other static loads, and if we assume infinitely stiff structures, buoyant force B can be calculated and applied as a single resultant force directed upward. Calculation of static buoyant force is based on the Archimedes principle: "Any object immersed in water is buoyed up by a force equal to the weight of water displaced by the object", and therefor such a buoyancy force is given as:

$$B = \rho_w g V \tag{502.4}$$

where $\rho_w = 999.972\ \mathrm{kg/m^3}$ is the mass density of water (at temperature of $+4^o$C with small changes of less than 1% up to $+40^o$C), $g = 9.81\ \mathrm{m/s^2}$ is the gravitational acceleration, and V is the volume of displaced fluid.

For seismic and other dynamic loading, and for structures which have finite stiffness (real stiffness), effects of buoyant (pressure) force will have to be modeled and simulated using methods that are more sophisticated than the simple static approach noted above.

Two main approaches to buoyancy modeling are described below:

- Sharp contrast in permeability (naturally occurring), between soil/rock and foundation concrete can be used to model buoyant pressures (and consequentially buoyant forces). This approach has been used before (Cheng and Jeremić, 2009b) and it works quite satisfactory if gap is not expected to form between soil/rock and concrete. Using sharp contrast approach, the buoyant forces are created by providing physical permeabilities for soil/rock (permeable rock, which can have high permeability) and for concrete which is quite impermeable (but not absolutely impermeable, permeability of concrete is couple of orders of magnitude lower than that of soil/rock). When the water tries to move (natural process due to pore water pressure gradient), it is restricted by the low permeability of the concrete, and thus forms a region of pressure (hydrostatic for static loading, or a different, dynamic pressure that results from dynamic behavior of soil/rock). This pressure is actually acting as a buoyant force on the concrete foundation. However, this approach only works well when there is no gap opening. In addition to that, there are modeling problems, with high pressure gradient close to the boundary between soil/rock and concrete, modeler needs to carefully mesh that region, to overcome too large pressure gradients in single layer of finite elements. A better approach, with or without gap opening is to use special coupled contact/interface finite element described below.

- Special coupled contact/interface finite element explicitly models water displacements and pressures and allows for explicit gap opening, filling of gap with water, slipping (frictional) when the gap is closed, and pumping of water as gap opens and closes. This contact/interface element incorporates the pore water pressure information, as well as the information about the displacement (movement) of pore water within a gap. It is based on a previous version of the contact/interface element with two important features in addition to features available to the dry contact element:

- Pore water pressure values (physical values obtained from simulation) from one side of the special coupled contact/interface element (side in contact/interface with saturated soil/rock) will be communicated (directly transferred) to the other side of this element. With this water pressure information available, the applied water pressure will be acting on the foundation finite elements. Foundation finite element will integrate pressure field on a given face (or faces) of the element and will create a buoyant force. Integration of pressure on an element face into buoyant forces is done using standard finite element procedure for calculating nodal forces (these are the buoyant forces in our case) from face pressures (Jeremić et al., 1989-2025):

  $$F_{Ia}^{buoyant} = \int_{S^m} f_a^{buoyant} \; H_I \; dS^m \tag{502.5}$$

  where $F_{Ia}^{buoyant}$ is the buoyant force at each node of the foundation finite element, $f_a^{buoyant}$ is the distribution of buoyant pressures on a face the foundation finite element, and H_I is a standard shape function of the foundation finite element (in our case, linear for 8 node brick, or quadratic for a 27 node brick). The integration is performed over a surface area S^m of each finite element face where buoyant pressures are present. With buoyant forces acting at the bottom (or sides) of a foundation, a proper reduction (change) of contact/interface pressures (forces) will be calculated. This means that the slipping (frictional) criteria of the contact/interface element will have all the necessary information about the normal forces (now reduced because of buoyant forces) and will determine is the contact/interface will slip and remain attached (at the location of that contact/interface element).

- Gap opening (physical values obtained from simulations) will be used to create suction and compression pore/gap water field. This will be achieved by connecting the displacements of the pore fluid from soil/rock finite elements (we will be using u-p-U finite elements for soil/rock modeling (Jeremić et al., 2008)), to the contact/interface element node on the opposite side of the special coupled contact element. This way, if the gap opens, and the contact/interface element now features an opening, the water displacements from the soil/rock side of the contact/interface element will be "pulled" to follow the uplifting foundation. This water movement will create pressure gradients in the soil/rock elements beneath, which will be accurately modeled using fully coupled u-p-U finite elements for soil/rock (Jeremić et al., 2008; Jeremić and Cheng, 2009).

Using above described approach to modeling will provide for high fidelity modeling and simulation of the buoyant pressures/forces, which will resolve all the difficulties related to this modeling.

502.3.8 Structural Foundations Modeling and Simulation

27 node solid bricks

502.3.9 Seismic Isolator Modeling and Simulation

Latex Rubber

Neoprene Rubber

Rubber with lead core

Frictional Pendulum

502.3.10 Structural Components Modeling and Simulation

Shells

Thick shells

502.3.11 Nonlinear Time Domain Analysis Progress and Example

502.3.11.1 Model Development

Mesh Development

Material Model Development

Loading Stages Development

502.3.11.2 Simulation Development

Sequential versus Parallel

Simulation Progress and Control

502.3.11.3 Seismic Motions

Full 3C Seismic Wave Field

- 1C wave field (deconvolution)
- 3C, inclined or vertical (body and surface waves) wave field, using 2D FEM fault slip model, see section 502.2.6 on page 2278
- 3C, inclined (body and surface waves) wave field, using fk (see section 706.1 on page 2926
- 3C, inclined (body and surface waves) wave field, using analytic solutions (Kausel, 2006)

Seismic Input Using DRM Theory, see section 109.4.1.1 on page 569.

Input, see section section 205.3 on page 827

502.4 Step by Step, Hierarchical Inelastic ESSI Analysis

502.4.1 ESSI Model Verification

ESSI Model Verification consists of gradual, hierarchical development of the large, detailed ESSI model, with verification of model response as sophistication level increases.

Recommended nonlinear/inelastic modeling and simulation phases for an SSI system are shown in Figure 502.122.

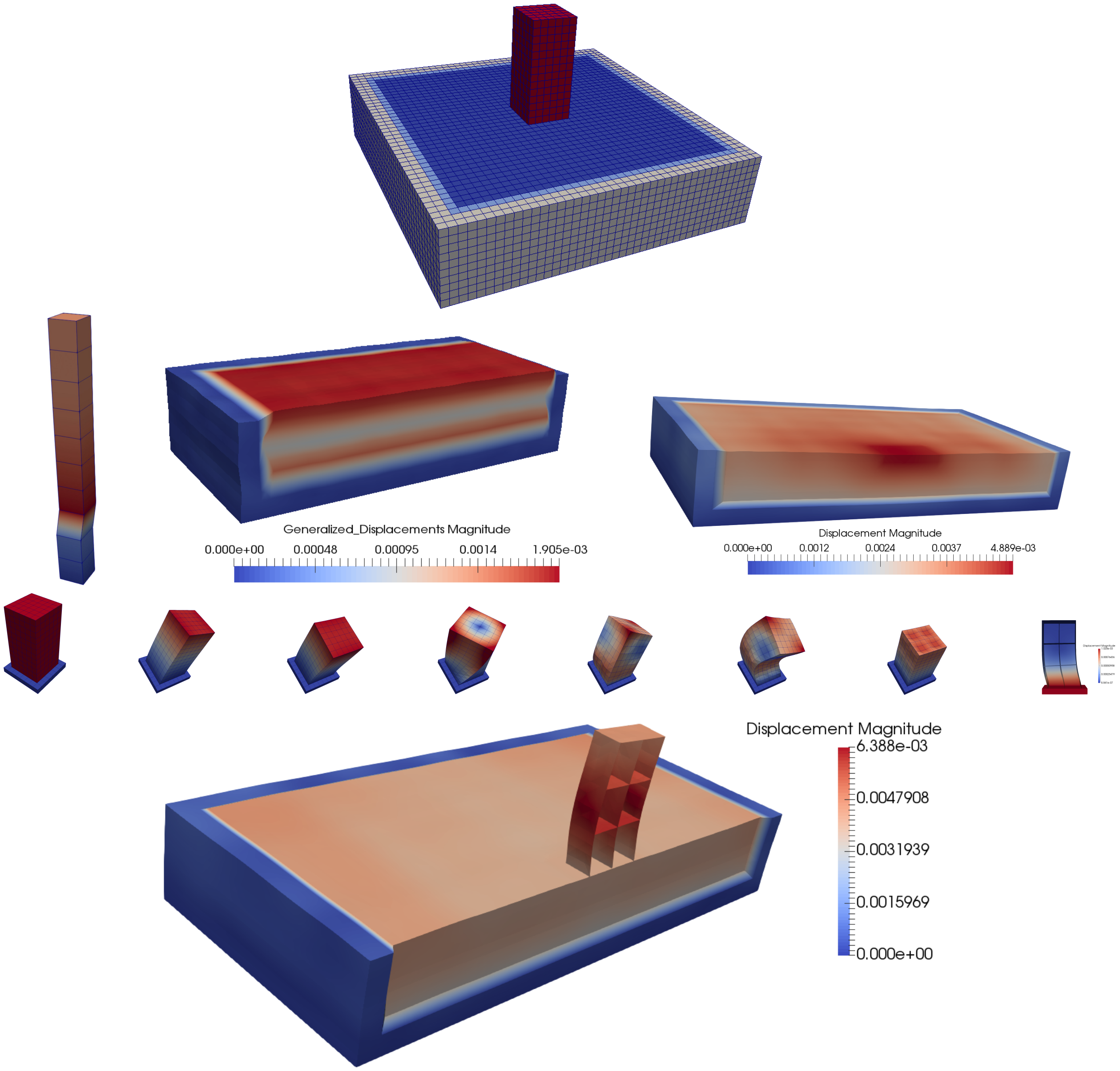


Figure 502.122: Step by step, hierarchical inelastic ESSI modeling and simulations.

It is highly recommended to proceed in phases, from simpler to more sophisticated, while employing sound engineering judgement at the end of each phase in order to understand static and dynamic response of the system and its components. For example, for an SSI system shown in Figure 502.122, advice is to follow these steps:

- start with a 1D model for one component (1C) shear wave, SV free field wave propagation, linear elastic soil, simple wavelet motions (Ricker, Ormsby)
- For a 1D-1C free field wave propagation, linear elastic soil, use more realistic motions, using

deconvolution of surface motions and convolution of rock motions from depth

- For a 1D model, apply 1 component (1C) vertical motions, a free field P wave
- For a 1D model, apply 2 component (2C), shear wave, SV, motions, for a free field wave propagation,
- For a 1D model, apply 3 component (3×1C) motions, for a free field wave
- For a 1D-1C, 1D-2C and 1D-3×1C free field wave propagation, update soil model to mild nonlinear/inelastic and perform wavelet and seismic motions, as noted above
- For a 1D-1C, 1D-2C and 3×1C models, update soil model to more realistic nonlinear/inelastic model and perform wavelet and seismic motions, as noted above
- Develop a 3D model of a free field and follow above steps, 3D-1C, 3D-2C, 3D-3×1C, elastic, inelastic, wavelet, realistic seismic input
- For a 3D model develop full 3C motions, inclined waves, and test your system, using phased modeling of material (elastic-inelastic), and motions, wavelets, then realistic seismic motions
- For a 3D-1C, 3D-2C and 3D-3×1C, models, add foundation slab, follow steps from above, elastic, inelastic soil, wavelets, realistic motions.
- Develop model of a structure, fully fixed DoFs at the bottom and perform eigen-analysis.
- For a fixed base structural model, apply motions from above,
- Progression in modeling sophistication from simpler material models, linear elastic, to more sophisticated, inelastic material models, should be followed for each component of the model, soils, structures, special elements, &c.
- Finally you might be able to develop a full 3D model, first linear then slowly nonlinear/inelastic, first simple motions, wavelets, then more realistic 1C, 2C, 3×1C and 3C motions

Progression in modeling sophistication from simpler material models, linear elastic, to more sophisticated, inelastic material models, as suggested above, should be followed by progression of simulation sophistication, as noted below:

- Start with linear elastic material models for all components, including bonded contact/interface/joints.

- Proceed with explicit, non convergence check simulation at both constitutive and global levels:
  - On constitutive level, use Forward Euler algorithm. Theory for the Forward Euler, explicit constitutive algorithm is described in Section 104.3, on page 206, in Jeremić et al. (1989-2025). Commands for constitutive level Forward Euler algorithm are given in Section 205.3.5.15, on page 1117, in Jeremić et al. (1989-2025).
  - With Forward Euler, explicit computations on the constitutive level, it is only appropriate to use global, finite element algorithm with no convergence check, that is also known as the explicit global algorithm. This algorithm is described in some detail in Section 107.3, on page 523, in Jeremić et al. (1989-2025). Commands for constitutive level Forward Euler algorithm are given in Section 205.3.5.14, on page 1116, in Jeremić et al. (1989-2025).
- For fully implicit algorithm, with enforcement of equilibrium on constitutive and global, finite element levels, proceed with Backward Euler (or one of the variants) on constitutive level and Newton algorithm (or one of variants) on global level:
  - On constitutive level, use Backward Euler algorithm. Theory for the Backward Euler, implicit constitutive algorithm is described in Section 104.4, on page 207, in Jeremić et al. (1989-2025). Commands for constitutive level Forward Euler algorithm are given in Section 205.3.5.15, on page 1117, in Jeremić et al. (1989-2025).
  - With Backward Euler, implicit computations on the constitutive level, it is appropriate to use global, finite element algorithm with or without convergence check, that is, one can use either explicit or implicit global algorithm. Thesealgorithm is described in some detail in Section 107.3, on page 523, in Jeremić et al. (1989-2025). Commands for constitutive level Backward Euler algorithm are given in Section 205.3.5.14, on page 1116, in Jeremić et al. (1989-2025).

A section from IAEA TECODC chapter will be used for this section ...

Pecker et al. (2022)

Input files for all the examples are available online at this LINK. All the examples can run directly at the Amazon Web Services, through Real-ESSI image.

502.4.2 ESSI Model Validation

Quality assurance and confidence in aalysis results is developed through full program verification, all the available model validation, as described in in chapters in part 300 on page 1436. For a particular model that is analyzed, step by step analysis procedures are decribed in section 502.4 on page 2354.

It is higtly advisable to perform ESSI model validation as well. ESSI model validation increases confidence in validity of modeling results. ESSI model validation consists of developing models for all the components of the large detailed model, and performing analysis of each of these components using material parameters that are used in large detailed model. While this activity does not strictly represent validation, as results do not have to be directly compared to available tests results, ESSI model validation provides engineers with a way to crudely, approximately assess simple mechanical behavior model components.

For example, a number of elastic and inelastice models might be used to model behavior of soil and rock beneath the structure. Single finite element models are to be developed, models that replicate simple behavior, for example simple shear, pure shear, triaxial conditions, etc. Similarly, structural model components, fibers, beams, shells, are developed and tested using simple loads, 1D uniaxial loads, pure bending, etc. Interface, contact joint models shouls also be developed.

These simple models are to be numerically analyzed, tested, using sets of material parameters/properties that are used in a large, detailed model. Engineering analyst is to inspect simple mechanial responses, and, using her/his experience and engineering judgement, assess approximately if material parameters used for modeling provide reasonable mechanical response for model components.

502.5 Metamaterials and ESSI

Brillouin (1953)

Carta et al. (2017), Basone et al. (2019), Casablanca et al. (2018), Colombi et al. (2020), Mu et al. (2020), Brûlé et al. (2014), Cacciola et al. (2020), Colombi et al. (2016a), Colombi et al. (2016b), Krödel et al. (2015), Miniaci et al. (2016), Palermo et al. (2016), Palermo et al. (2018), Palermo and Marzani (2018), Zaccherini et al. (2020), AL-Shudeifat et al. (2013), Shen et al. (2021), AL-Shudeifat (2014), Wang et al. (2020b), Wenzel et al. (2020), Chen et al. (2020), Antoniadis et al. (2017), Yuksei and Yilmaz (2020), Chondrogiannis et al. (2020), Chondrogiannis et al. (2021), Banerjee et al. (2018), Cai et al. (2020), Wehmeyer et al. (2019), Fiore et al. (2020), Dertimanis et al. (2016), Cheng and Shi (2018), Kacin et al. (2021), Kanellopoulos et al. (2022),

Chapter 503

Earthquake-Soil-Structure Interaction, Bridge Structures

(2003-2007-2011-)

(In collaboration with Dr. Guanzhou Jie)

503.1 Chapter Summary and Highlights

503.2 Case History: Earthquake-Soil-Structure Interaction for a Bridge System

503.2.1 Prototype Bridge Model Simulation

The final objective this work is to improve current modeling techniques through the comparative study between numerical and experimental components. In this work, a whole prototype bridge model has been built using finite element techniques demonstrated in previous sections. Domain Reduction Method (Bielak et al., 2003b; Yoshimura et al., 2003b) (DRM) has been used to reduce the model size while still preserves the accuracy of the ground motion analysis.

503.2.1.1 Soil Model

Capitol Aggregates, a local quarry located in the south of Austin, has been selected to be the test site for this project. Site characterization has been preformed to collect information on the soil (Kurtulus et al., 2005). Based on the only triaxial test data available, a nonlinear soil constitutive model is developed in this work for prototype finite element analysis.

Undrained triaxial compression test has been carried out on one, $1.5in$ diameter triaxial test specimen trimmed from an undisturbed soil sample obtained from borehole at an approximate depth of $10.6ft$. The initial size and index properties of the soil specimen are given in Table 503.1 (Kurtulus et al., 2005).

Table 503.1: Index Properties of the Undisturbed Triaxial Test Specimen

Soil Index Property	Inital	After Consolidation	Failure
Diameter D $(inch)$	1.50	1.48	1.56
Height H $(inch)$	3.00	2.87	2.56
Total Unit Weight γ_t (pcf)	107.3	111.1	112.8
Water Content w (%)	18	18	18
Dry Unit Weight γ_d (pcf)	90.9	94.3	95.7
Void Ratio e[1]	0.84	0.77	0.75
Degree of Saturation S_r[1](%)	57	62	64

[1] Specific Gravity G_s is assumed to be 2.68.

In the triaxial cell, the specimen was allowed to come into equilibrium (compress/consolidate with drainage lines open) under an isotropic pressure equal to the assumed in-situ mean total stress, which is about $5.6psi$. Upon equilibrating, the specimen was sheared under undrained conditions with a strain rate of $\%1$ per hour. No pore pressure readings were taken since the specimen was unsaturated. The resulting stress-strain curve is presented in Figure 503.1. An estimate of the undrained shear strength in terms of total stresses was measured as $13.41psi$ $(1931psf)$ at about 9% strain. The specimen failed in a bulging mode. The index properties of the specimen at failure are presented in Table 503.1.

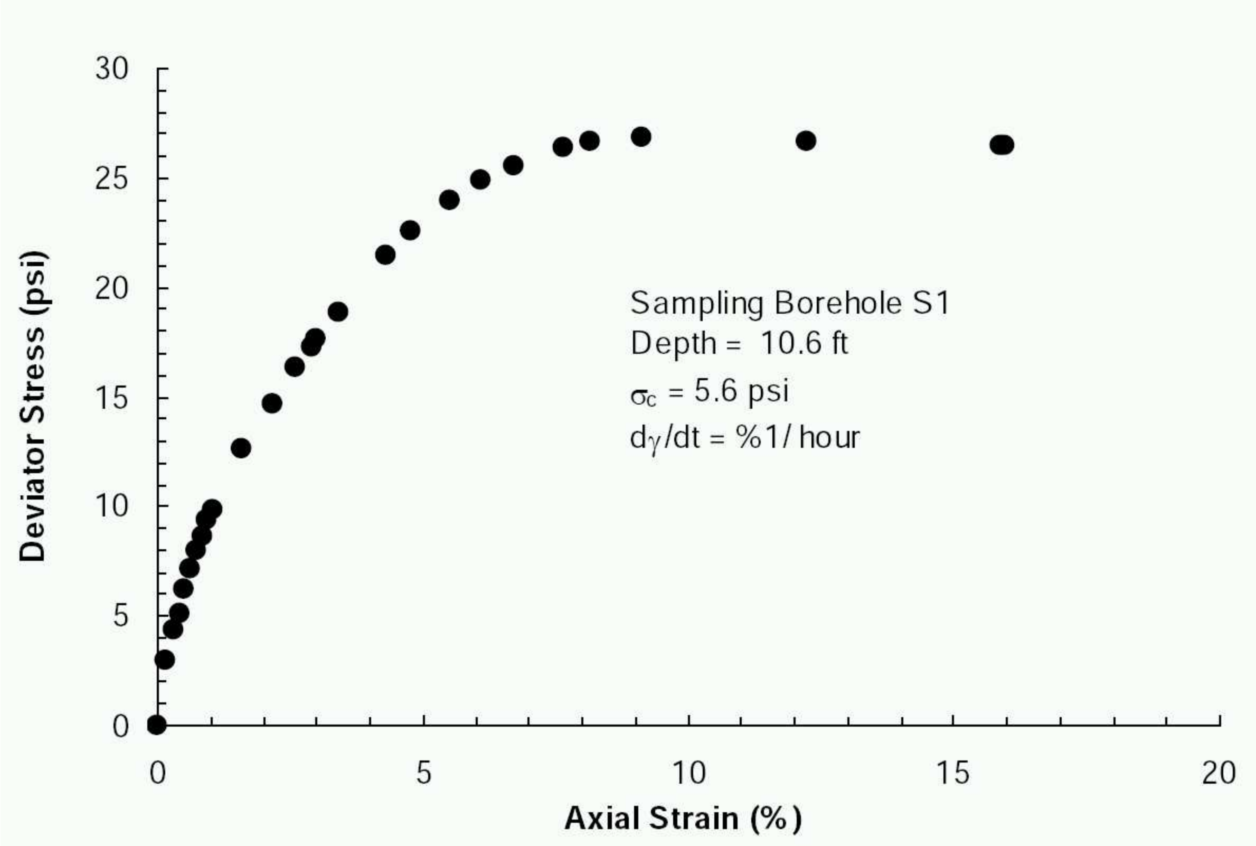


Figure 503.1: Total Stress Strain Curve Determined from Undrained Triaxial Compression Test (Undisturbed Sample from Depth $10.6ft$)

Based on the laboratory triaxial test data, a nonlinear elastic-plastic soil model has been developed to calibrate the finite element simulation. Associative Drucker-Prager plasticity model, combined with nonlinear Armstrong-Frederick kinematic hardening rule, yields good match between laboratory data and numerical results, as shown in Figure 503.2. The same model has been exposed to various confinements to test robustness of the model for soils at different depths.

503.2.1.2 Element Size Determination

The accuracy of a numerical simulation of dynamic SFSI (Soil-Structure-Foundation-Interaction) problems is controlled by two main parameters (Preisig, 2005):

1. The spacing of the nodes of the finite element model Δh

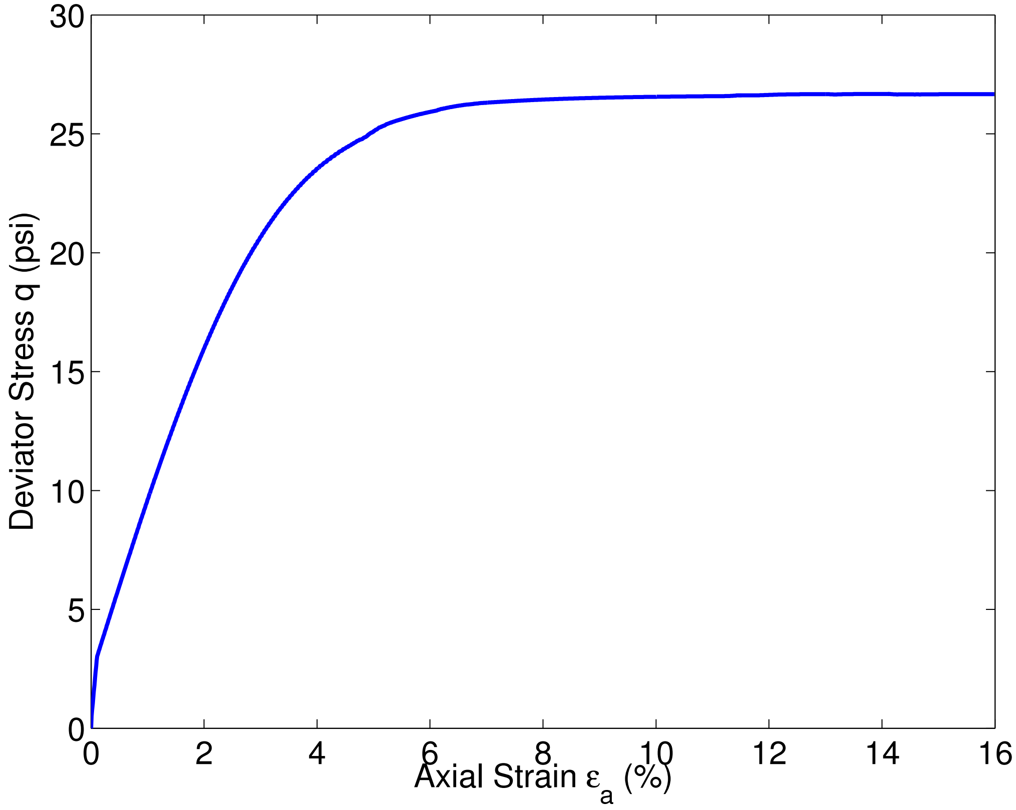


Figure 503.2: Total stress strain curve obtained using simulation of triaxial test (from depth $10.6 ft$)

2. The length of the time step Δt.

Assuming that the numerical method converges toward the exact solution as Δt and Δh go toward zero the desired accuracy of the solution can be obtained as long as sufficient computational resources are available.

As presented in Preisig (2005), in order to represent a traveling wave of a given frequency accurately about 10 nodes per wavelength are required. Fewer than 10 nodes can lead to numerical damping as the discretization misses certain peaks of the wave. In order to determine the appropriate maximum grid spacing the highest relevant frequency f_{max} that is present in the model needs to be found by performing a Fourier analysis of the input motion. Typically, for seismic analyses f_{max} is about $10 Hz$. By choosing the wavelength $\lambda_{min} = v/f_{max}$, where v is the wave velocity, to be represented by 10 nodes the smallest wavelength that can still be captured partially is $\lambda = 2\Delta h$, corresponding to a frequency of $5 f_{max}$.

The maximum grid spacing should not exceed

$$\Delta h \leq \frac{\lambda_{min}}{10} = \frac{v}{10 f_{max}} \tag{503.1}$$

where v is the smallest wave velocity that is of interest in the simulation.

Table 503.2: Maximum Element Size Determination ($f_{max} = 10HZ$)

Depth (ft)	Thickness (ft)	v_{shear} (fps)	Δh_{max} (ft)	h_{max} (m)
0	1	320	3.2	0.98
1	1.5	420	4.2	1.28
2.5	4.5	540	5.4	1.65
7	7	660	6.6	2.01
14	7.5	700	7.0	2.13
21.5	17	750	7.5	2.29
38.5	half-space	2200	22.0	6.7

In this work, the prototype site chosen is Capitol Aggregates, a local quarry located in the south of Austin. According to the site characterization report Kurtulus et al. (2005), we obtain Table 503.2 for element size determination.

Mechanical properties of soil changes with cyclic loadings. In order to predict more accurately the dynamic behaviors of soil subject to earthquake loadings, various laboratory and in situ tests have been performed to examine the degradation of dynamic soil properties. Equivalent linear model has been used extensively in practice (Kramer, 1996b). Moduli reduction curve (G/G_{max}) and damping ratio relationship have been obtained for prototype soil at the site of Capitol Aggregates as shown in Figure 503.3.

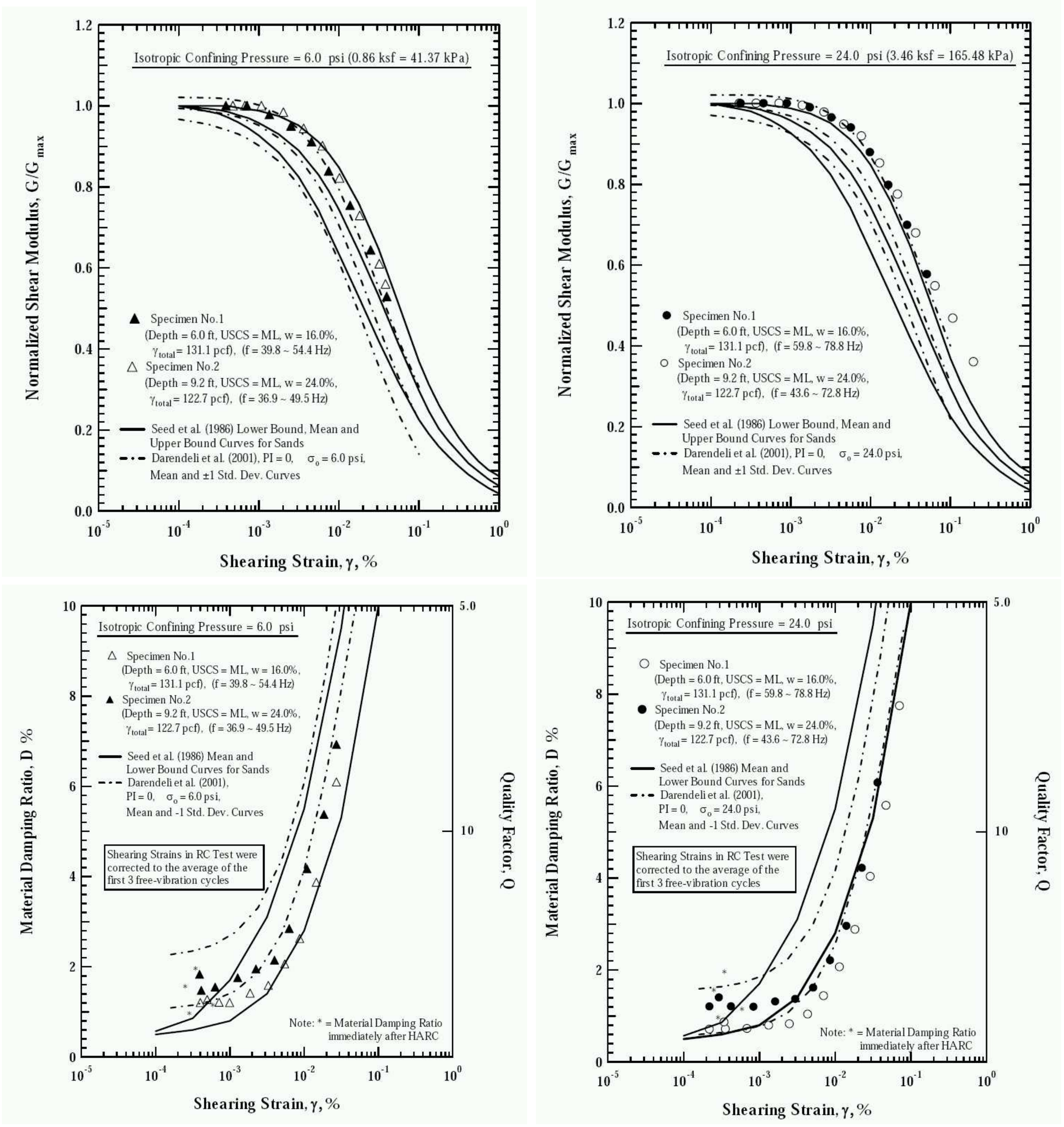


Figure 503.3: Comparison of the Variation in Normalized Shear Modulus and Damping Ratio with Shearing Strain from the Resonant Column Tests with Modulus Reduction Curves proposed by Seed et al. (1986) and Darendeli (2001) (Kurtulus et al., 2005)

Table 503.3: Element Size Determination after Degradation of $G(v_s)$ ($f_{max} = 10HZ$)

Depth (ft)	Thickness (ft)	v_s (fps)	Min G/G_{max}[2]	Min v_s	Δh_{max} (ft)	h_{max} (m)
0	1	320	0.36	192	1.92	0.59
1	1.5	420	0.36	252	2.52	0.77
2.5	4.5	540	0.36	324	3.24	0.99
7	7	660	0.36	396	3.96	1.21
14	7.5	700	0.36	420	4.20	1.28
21.5	17	750	0.36	450	4.50	1.37
38.5	half-space	2200	0.36	1320	13.20	4.02

[2] The value is obtained from the lab test data, which corresponds to 0.2% strain level.

The degradation of dynamic soil properties as observed in experiments has to be considered in finite element analysis in order to capture more accurate behaviors. As the shear wave velocity correlates with shear modulus by Equation 503.2,

$$v_{shear} = \sqrt{\frac{G}{\rho}} \tag{503.2}$$

we can readily obtain the dynamic degradation of wave velocities. This leads to smaller element size required for detailed simulation of wave propagation. The newly calculated element sizes are listed in Table 503.3. A three bent prototype finite element model has been developed with element size $\Delta h = 0.6$m as shown in Figure 503.4

503.2.1.3 Time Step Length Requirement

As stated in Preisig (2005), the time step Δt used for numerically solving nonlinear vibration or wave propagation problems has to be limited for two reasons. The stability requirement depends on the numerical procedure in use and is usually formulated in the form $\Delta t = T_n <$ value. T_n denotes the smallest fundamental period of the system. Similar to the spatial discretization T_n needs to be represented by about 10 time steps. While the accuracy requirement provides a measure on which higher modes of vibration are represented with sufficient accuracy, the stability criterion needs to be satisfied for all modes. If the stability criterion is not satisfied for all modes of vibration, then the solution may diverge. In many cases it is necessary to provide an upper bound to the frequencies that are present in a system by including frequency dependent damping to the model.

The second stability criterion results from the nature of the finite element method. As a wave front progresses in space it reaches one point after the other. If the time step in the finite element analysis is

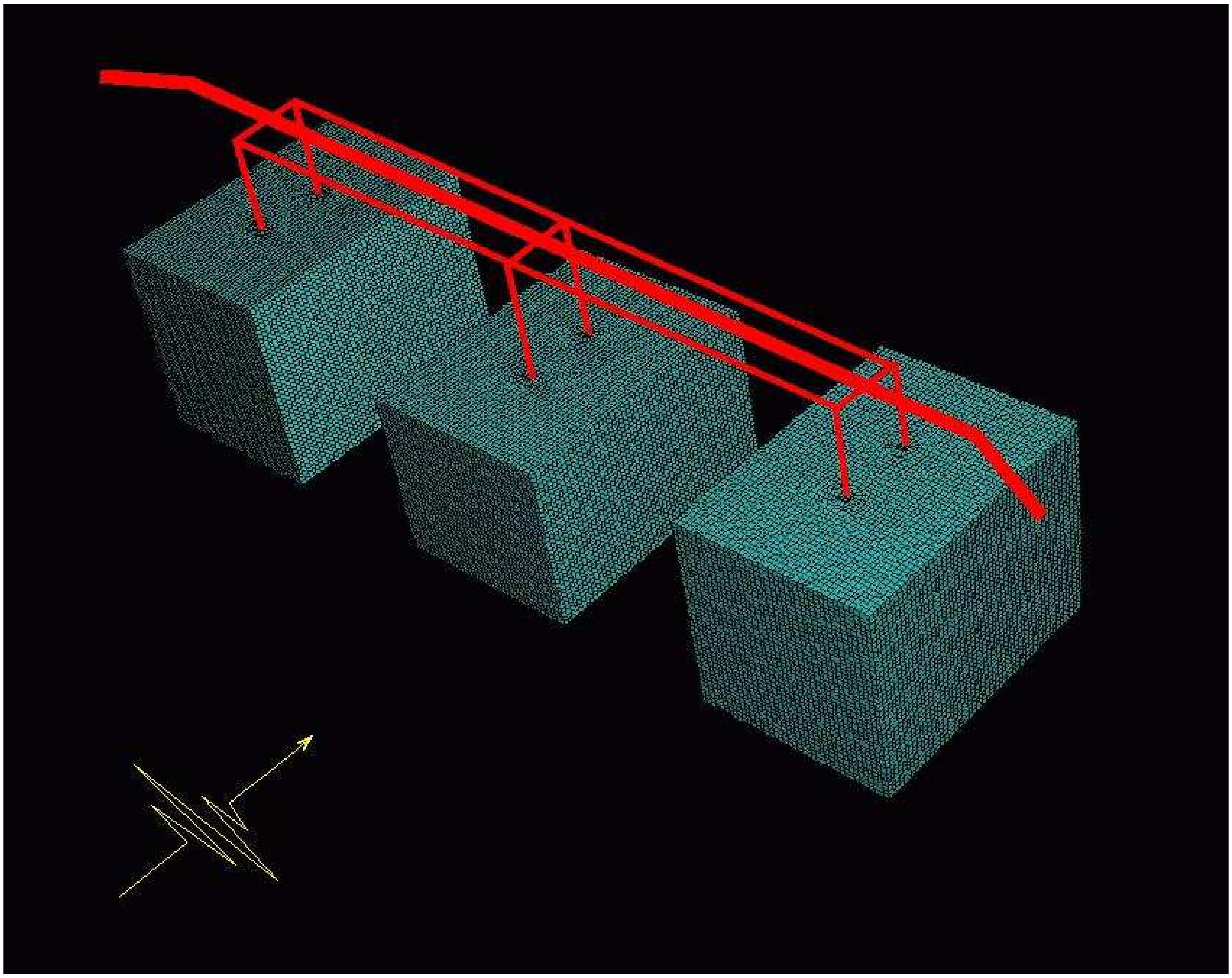


Figure 503.4: Detailed Three Bent Prototype SFSI Finite Element Model - 3 Bent SFSI, 484,104 DOFs, 151,264 Elements, Frequency Cutoff 10Hz, Element Size 0.3m, Minimum G/G_{max} 0.08, Maximum Shear Strain γ 1%

too large the wave front can reach two consecutive elements at the same moment. This would violate a fundamental property of wave propagation and can lead to instability. The time step therefore needs to be limited to

$$\Delta t < \frac{\Delta h}{v} \tag{503.3}$$

where v is the highest wave velocity.

According to Table 503.3, the time step requirement can be obtained as

$$\Delta t < \frac{\Delta h}{v} = 0.00256 \tag{503.4}$$

in seconds.

503.2.1.4 Domain Reduction Method

Domain reduction method was originally proposed in Bielak et al. (2003b); Yoshimura et al. (2003b) The theory aims at reducing the size of simulation domain by means of variable interchange. This method features a two-stage strategy for complicated three dimensional earthquake engineering simulations. The first is an auxiliary problem that simulates the earthquake source and propagation path effects with a model that encompasses the source and a background structure from which the localized feature has been removed. The second problem models local site effects. Its input is a set of equivalent localized forces derived from the first step. These forces act only within a single layer of elements adjacent to the interface between the exterior region and the geological feature of interest. The beauty of this theory comes from the fact that we can use established numerical and/or experimental approaches to solve the first-stage wave propagation problem. With the outcome of the first phase solution, we greatly reduce the size of the problem and then efforts can be focused on the second phase to deliver more accurate simulation on local responses. This approach can be successfully used in soil-foundation-structure-interaction finite element modeling without the need to incorporate unnecessary far-field motion simulations.

503.2.1.5 Structural Model

The nonlinear structure model developed in this work is a joint effort of UCB and UCD. Experimental data has been collected from UNR shaking table tests to calibrate the structural models. The effort in this work has been focused on how to integrate advanced structure model with geotechnical model to enable full-scale prototype simulations. The assumption that the plastic hinge forms either on the top of column or at the fixed bottom does not hold for SFSI problems. This restriction has been removed as the geotechnical and structural models are connected together.

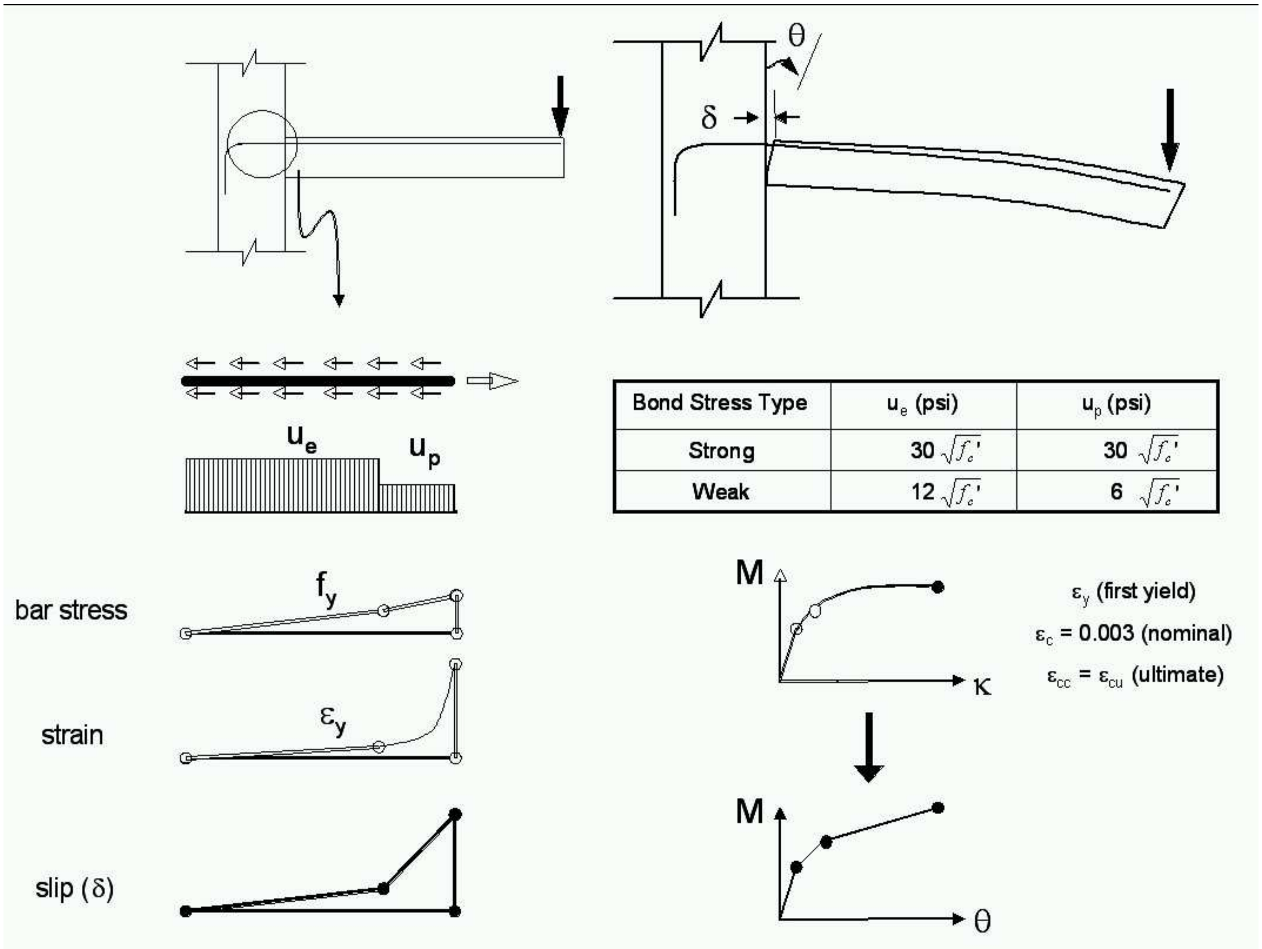


Figure 503.5: Simplified Hinge Model Developed for SFSI Prototype Simulations (Dryden, 2005)

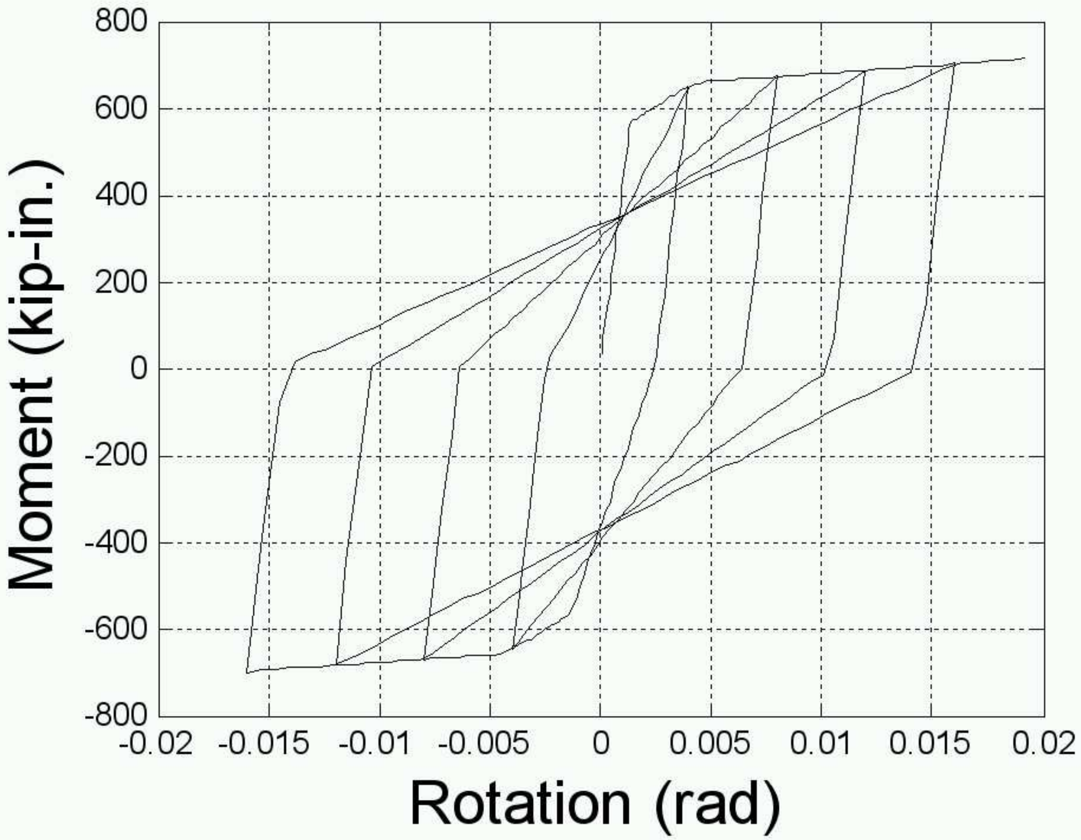


Figure 503.6: Moment-Rotation Relationship of Structural Hinge Model (Dryden, 2005)

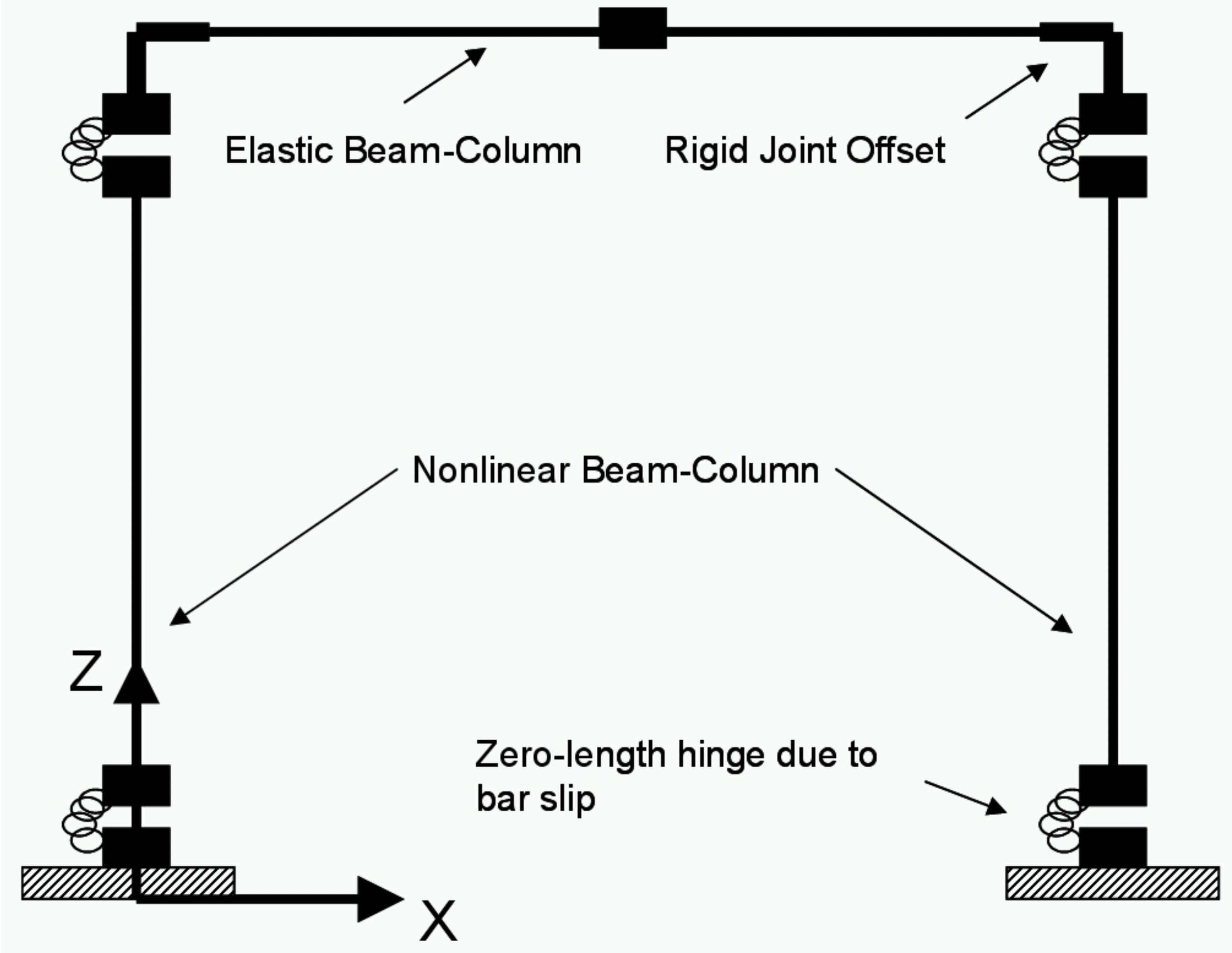


Figure 503.7: Developed structural model (Dryden, 2005).

503.2.1.6 Simulation Scenarios

In order to fully investigate how the relative strength of soil to the structure can affect the system behaviors, a thorough parametric study has been performed. The prototype model used in this work is a 4-span bridge structure with 3 bents. The supporting soil foundation can be varied according to different site conditions. The mesh of the prototype finite element model is shown in the Figure 503.8. The underlying soil of the bridge can be soft bay mud or stiff sand. In order to fully investigate the SFSI response, various scenarios are simulated as shown in Table 503.4 and results are analyzed.

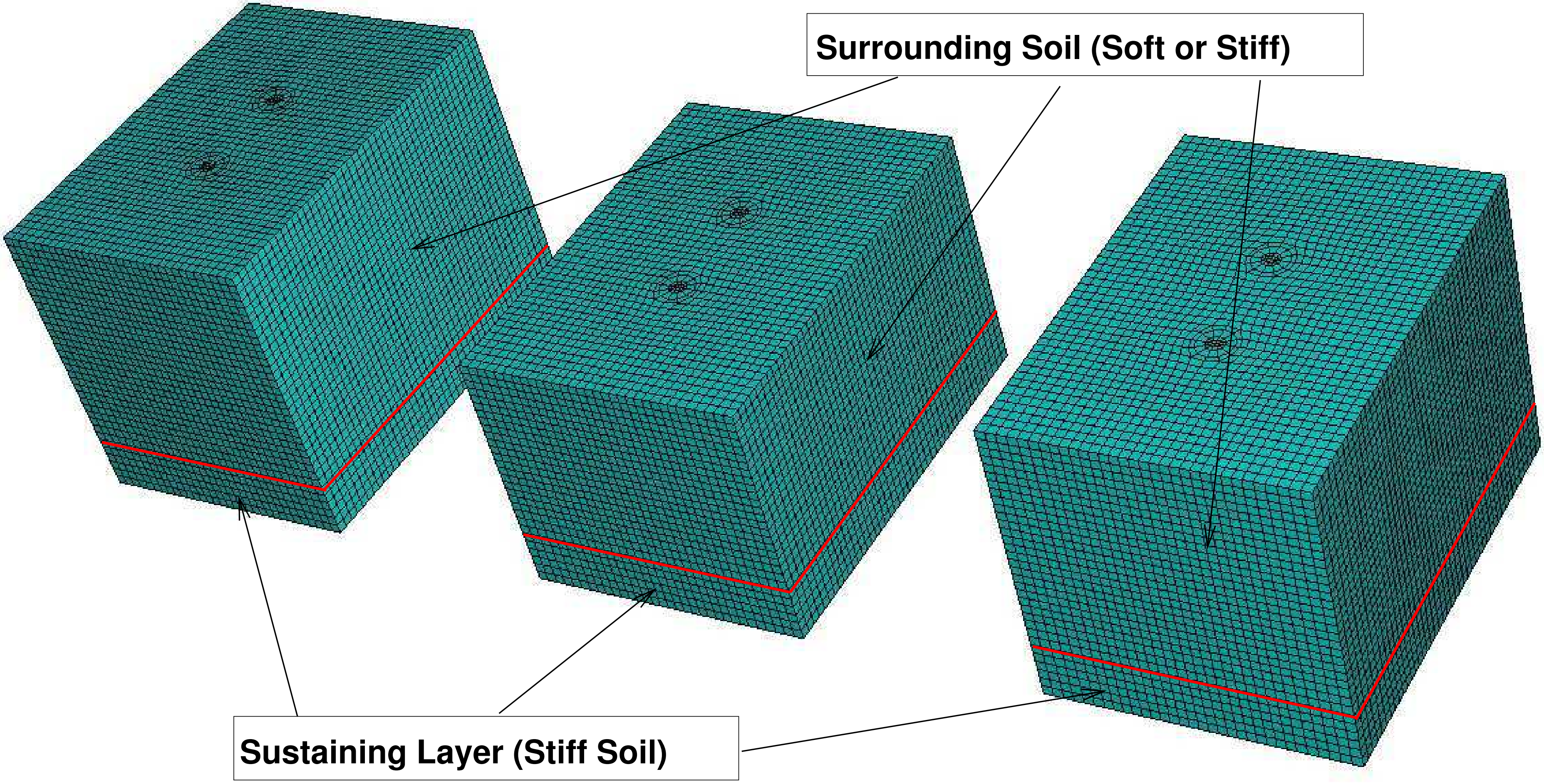


Figure 503.8: Finite Element Model for 3 Bent Prototype Bridge System

Table 503.4: Simulation Scenarios for Prototype SFSI Studies

Simulation Cases	Soil Block 1	Soil Block 2	Soil Block 3
Case 1	Stiff Sand	Stiff Sand	Stiff Sand
Case 2	Stiff Sand	Stiff Sand	Soft Clay
Case 3	Stiff Sand	Soft Clay	Stiff Sand
Case 4	Stiff Sand	Soft Clay	Soft Clay
Case 5	Soft Clay	Stiff Sand	Stiff Sand
Case 6	Soft Clay	Stiff Sand	Soft Clay
Case 7	Soft Clay	Soft Clay	Stiff Sand
Case 8	Soft Clay	Soft Clay	Soft Clay

Constitutive Modeling of Stiff Sand For the stiff sand, the constitutive model developed in previous sections, as shown in Table 503.1 will be used.

Constitutive Modeling of Soft Clay This soil model aims at simulating in-situ undrained behavior of soft bay mud. Undrained shear strength can be easily determined and a simple von Mises model is used in this research.

503.2.2 Earthquake Simulations - 1994 Northridge

Starting from this section, detailed numerical simulation results will be presented to show how the finite element simulation techniques can be used in prototype earthquake simulations. The results are presented here, and discussions will follow. Figure 503.9 shows the input motion recorded from 1994 Northridge which contains lots of high frequency contents. There are totally two motions are selected for this work, one with primary short period (high frequency) contents, and the other with primary long period (low frequency) contents. The purpose is to study every single component of the SFSI system trying to expose how each affects the SFSI system response.

503.2.2.1 Input Motion

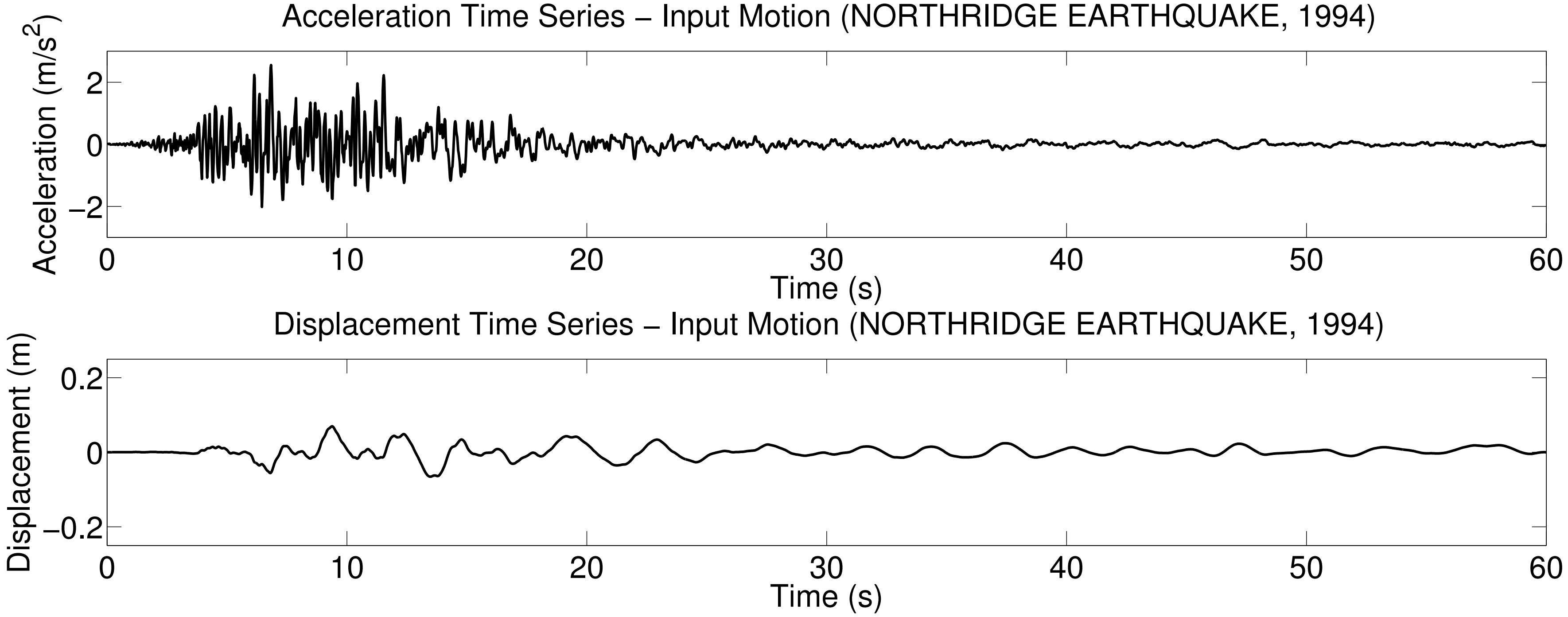


Figure 503.9: Input Motion - Century City, Northridge Earthquake 1994

503.2.2.2 Displacement Response

This section presents the displacement results from numerical simulations.

There are a couple of interesting things that deserve attention. Firstly, in the 1994 Northridge earthquake which contains much high frequency content, the structural response from softer soil actually is smaller than those on top of stiff soil. This is interesting because it basically controdicts the common notion that stiffer the soil, stabler the structure. That is the case for static design. But for earthquake design, we are presenting different stories. Secondly, we see the soil displacement near structure is largely affected by the SFSI. So the question if it is valid to apply outcrop motion directly to fixity point to excite the structure, just as people commonly do, might need a revisit.

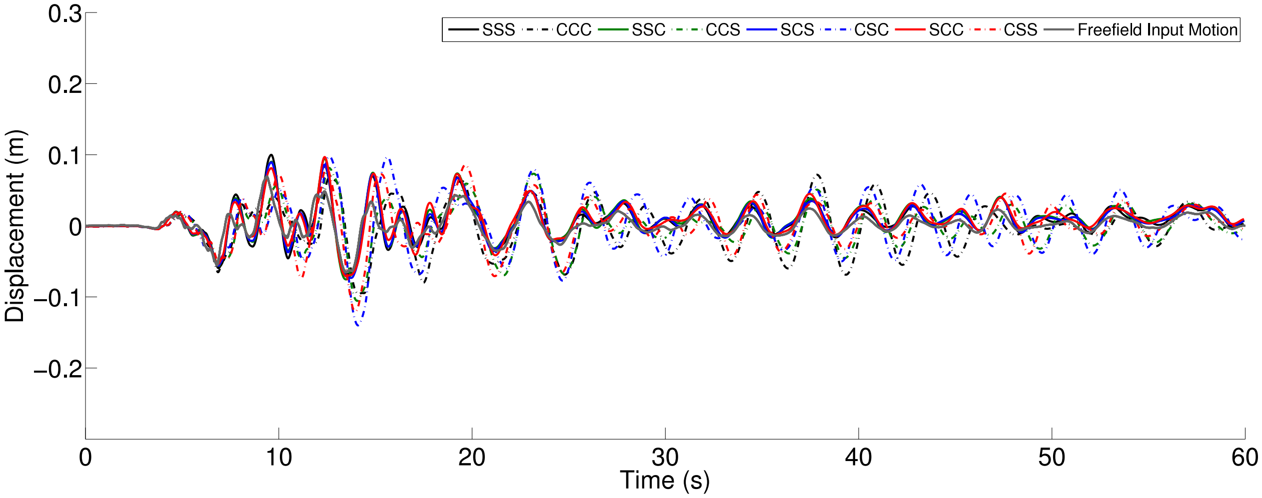


Figure 503.10: Simulated Displacement Time Series, Northridge 1994, Century City, Comparison of Eight Cases with Free Field Motions (Soil Block 1)

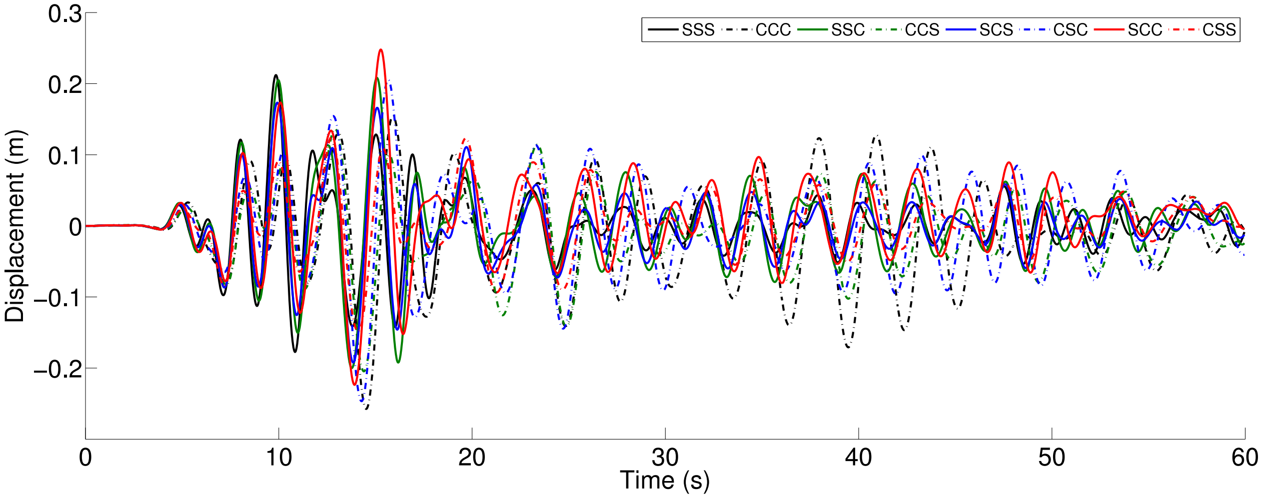


Figure 503.11: Simulated Displacement Time Series, Northridge 1994, Century City, Comparison of Eight Cases (Structure Bent 1)

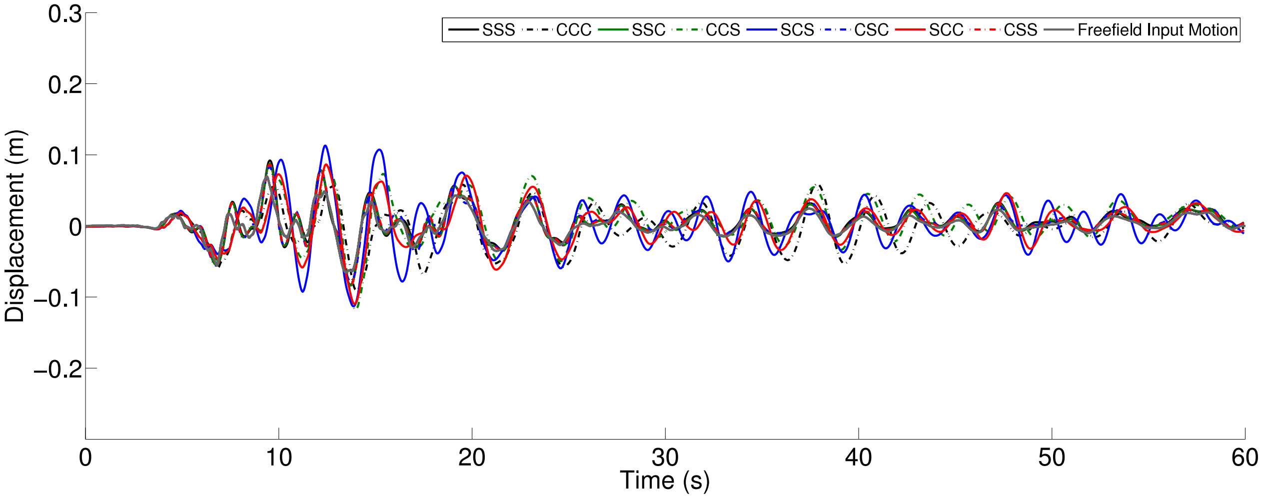


Figure 503.12: Simulated Displacement Time Series, Northridge 1994, Century City, Comparison of Eight Cases with Free Field Motions (Soil Block 2)

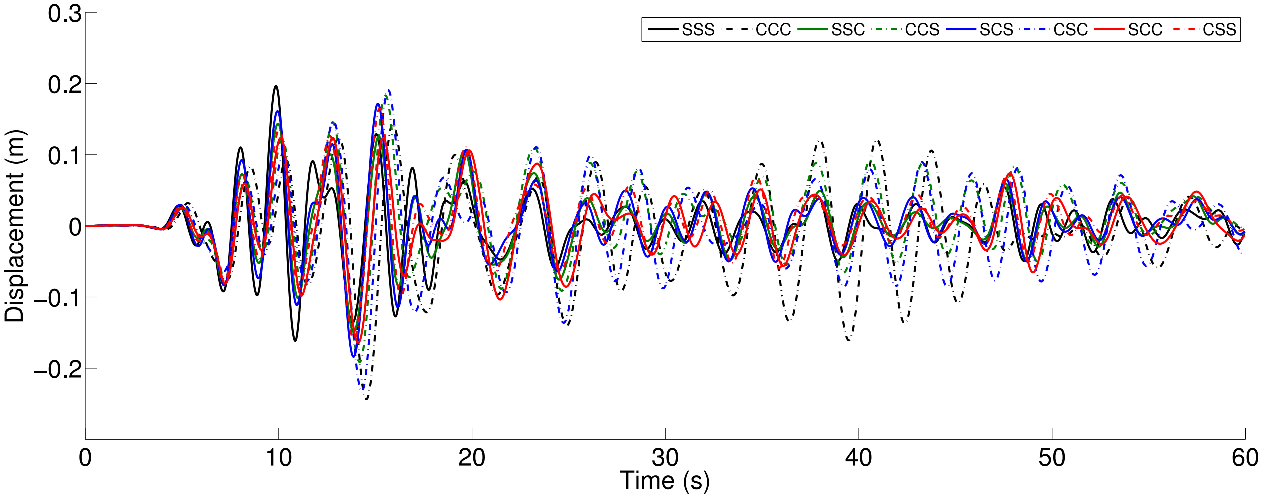


Figure 503.13: Simulated Displacement Time Series, Northridge 1994, Century City, Comparison of Eight Cases (Structure Bent 2)

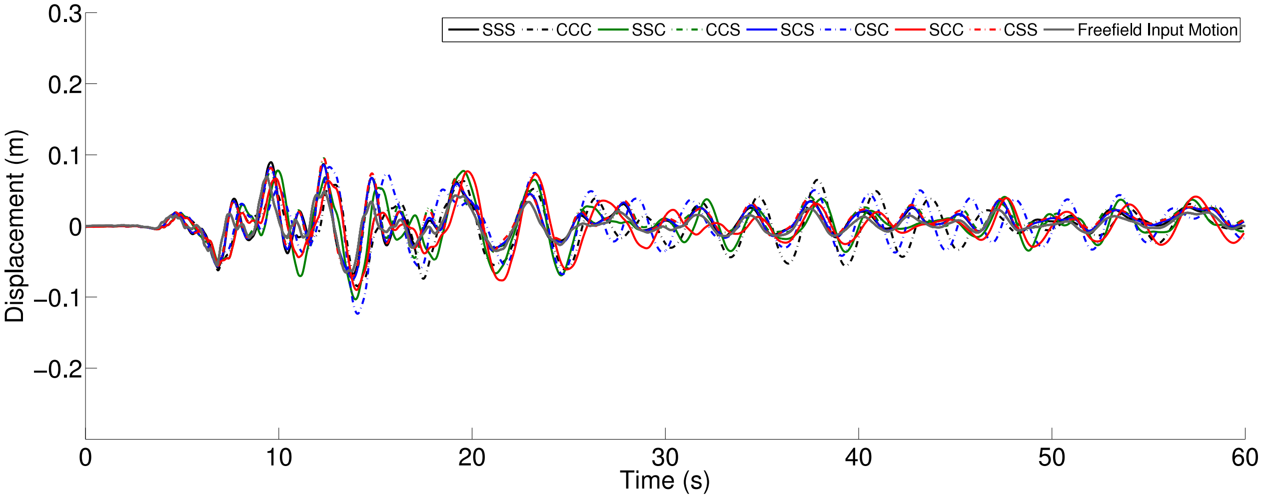


Figure 503.14: Simulated Displacement Time Series, Northridge 1994, Century City, Comparison of Eight Cases with Free Field Motions (Soil Block 3)

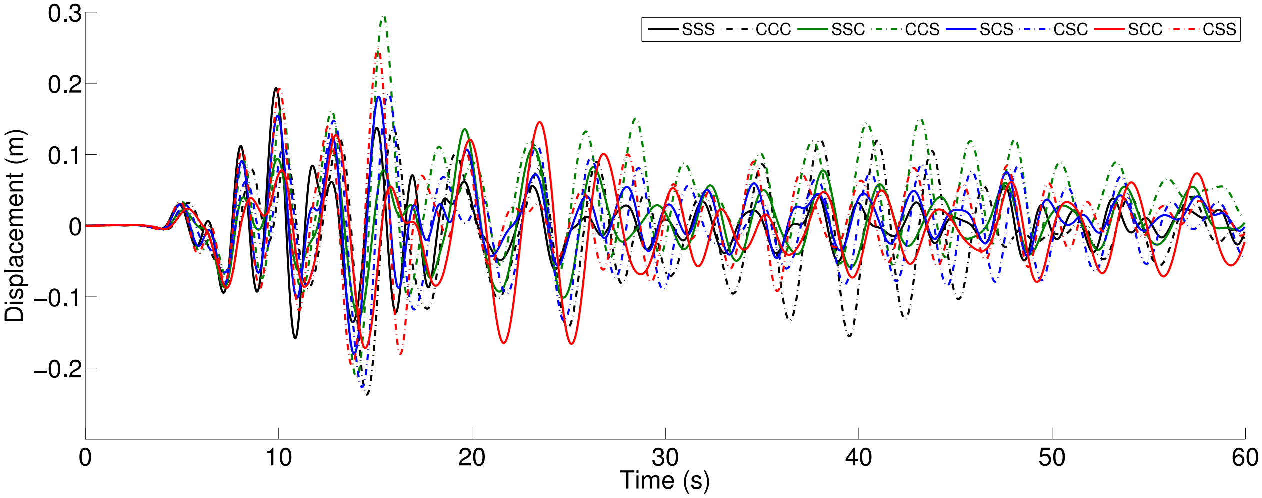


Figure 503.15: Simulated Displacement Time Series, Northridge 1994, Century City, Comparison of Eight Cases (Structure Bent 3)

503.2.2.3 Acceleration Response

The acceleration results are shown in this section, which also supports the observation that stiffer soil might not necessarily enhance the stability of the structure. Acceleration time series consistently show that the stiff soil will excite larger amplification for structures.

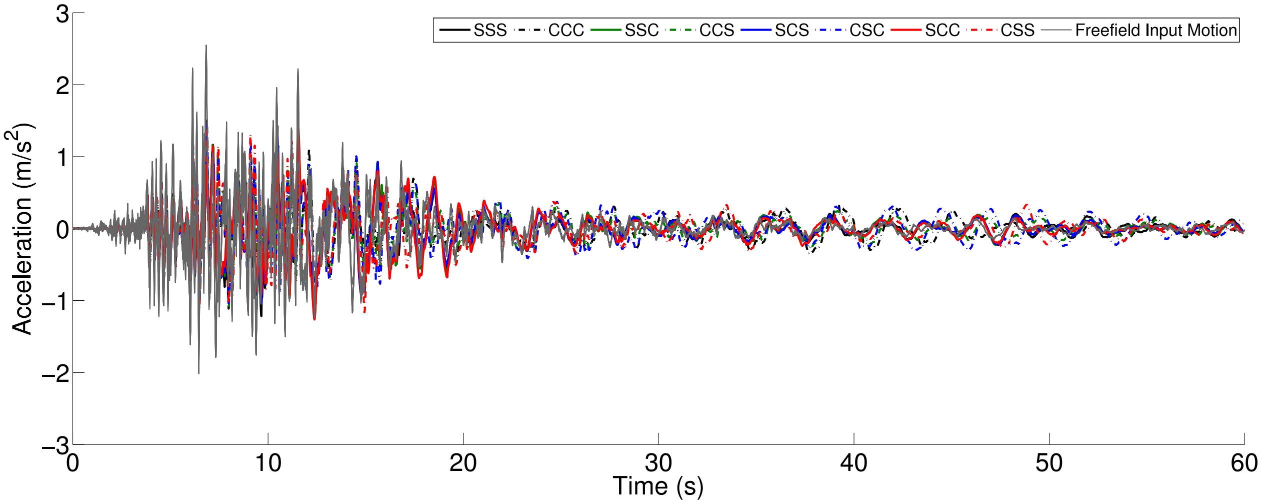


Figure 503.16: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Eight Cases with Free Field Motions (Soil Block 1)

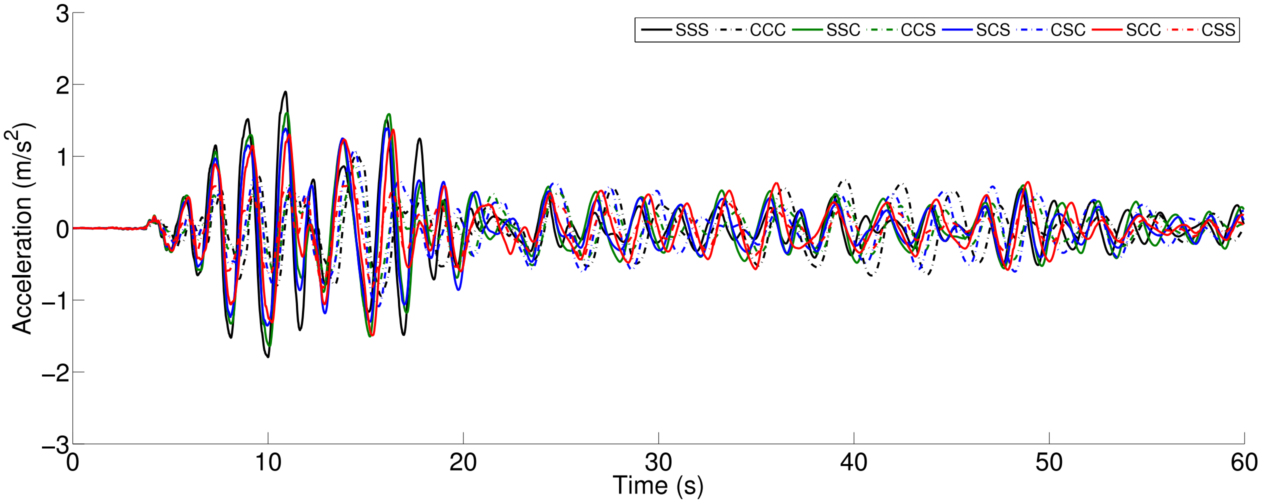


Figure 503.17: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Eight Cases (Structure Bent 1)

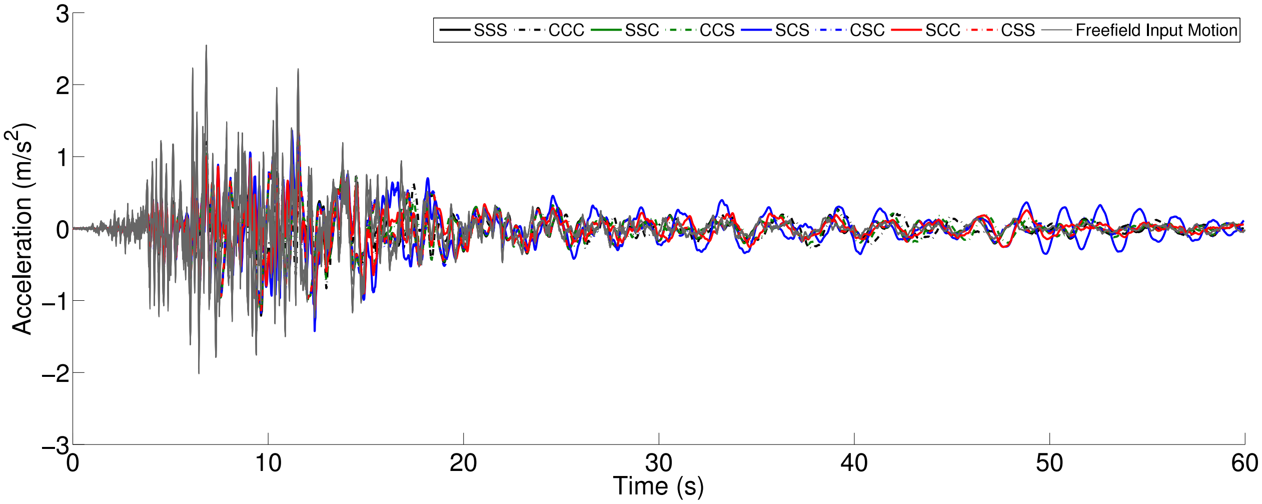


Figure 503.18: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Eight Cases with Free Field Motions (Soil Block 2)

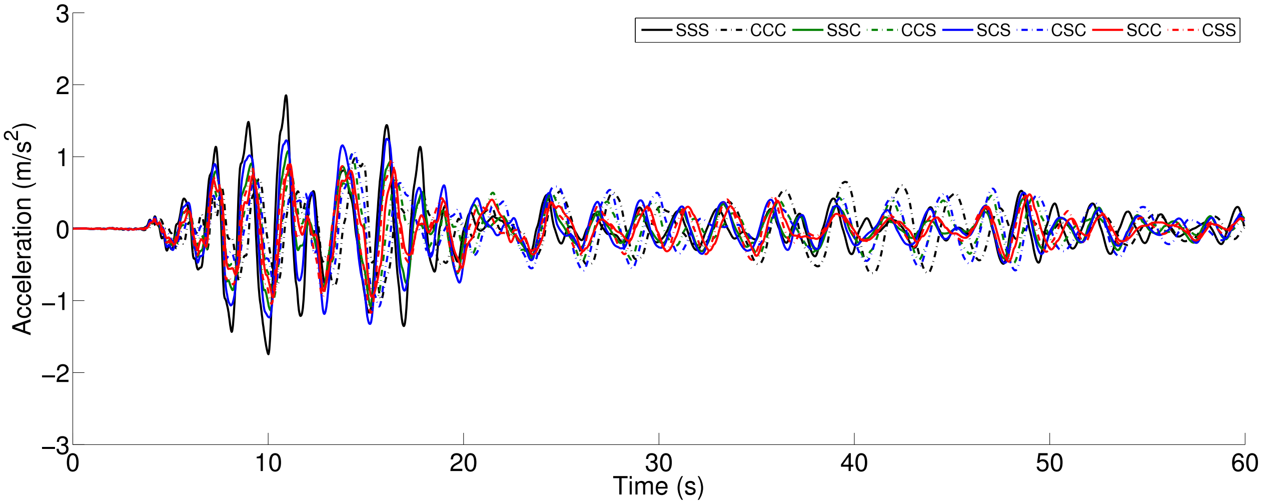


Figure 503.19: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Eight Cases (Structure Bent 2)

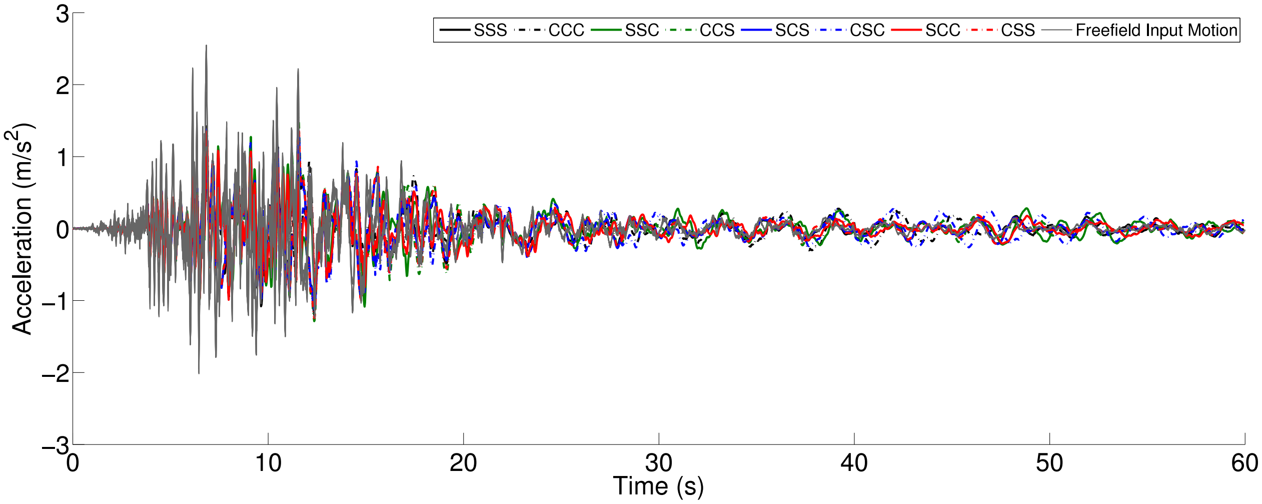


Figure 503.20: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Eight Cases with Free Field Motions (Soil Block 3)

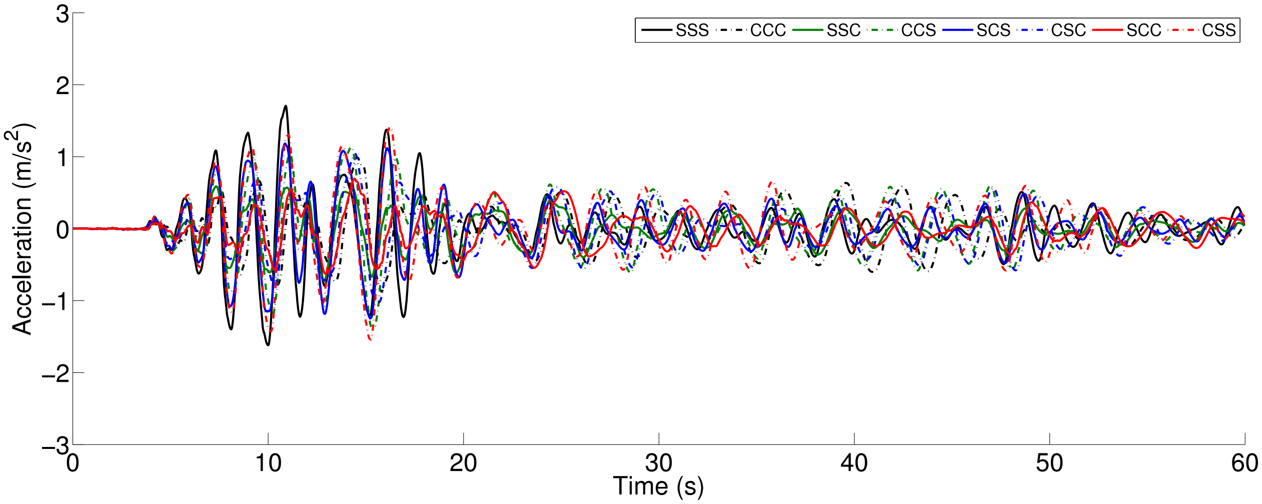


Figure 503.21: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Eight Cases (Structure Bent 3)

503.2.2.4 Displacement Response Spectra

It will be very interesting to look into the frequency domain what is going on. The structure is always stiffer than underlying soils. So if the underlying soil is stronger, it implies the natural frequency of the stiff soil will be closer to the structure on top of it.

If, the input motion contains much high frequency content, it will directly excite the stiffer soil so the structure on top will receive very large amplification. This conclusion is supported by following plots.

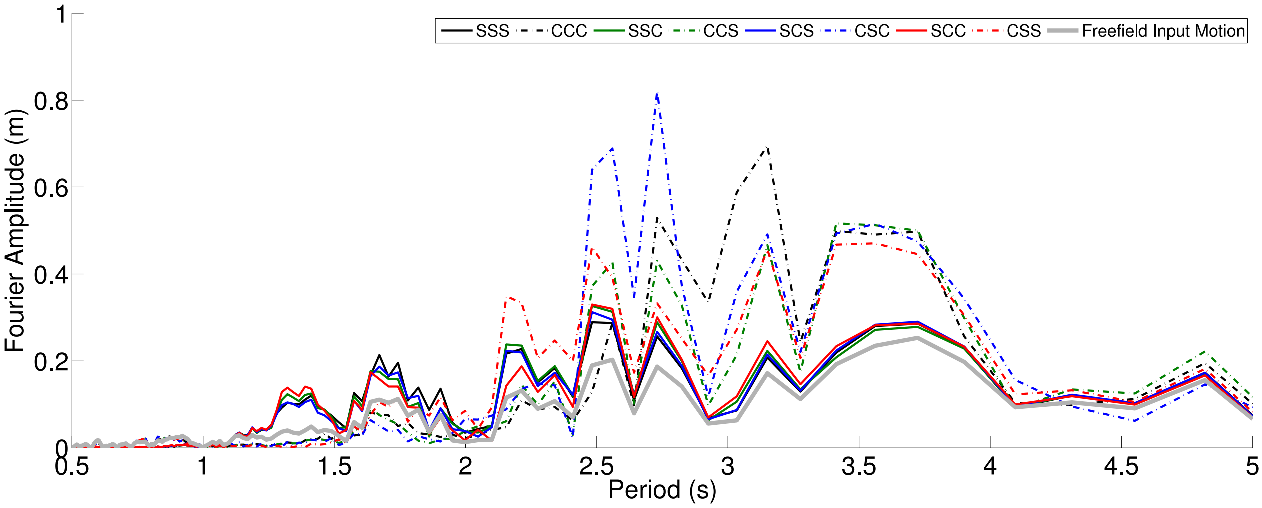


Figure 503.22: Simulated Displacement Response Spectra, Northridge 1994, Century City, Comparison of Eight Cases with Free Field Motions (Soil Block 1)

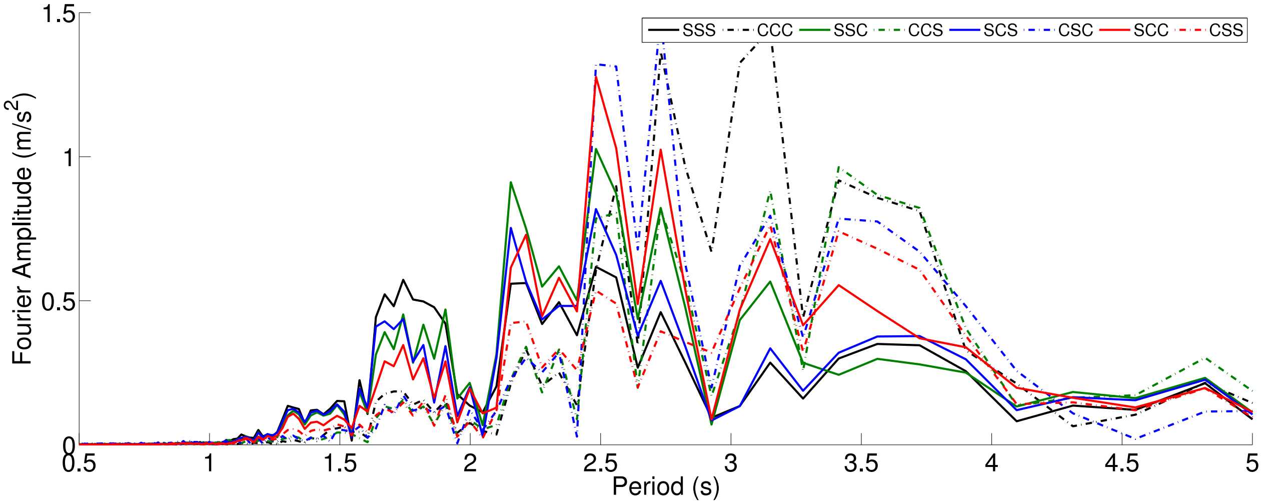


Figure 503.23: Simulated Displacement Response Spectra, Northridge 1994, Century City, Comparison of Eight Cases (Structure Bent 1)

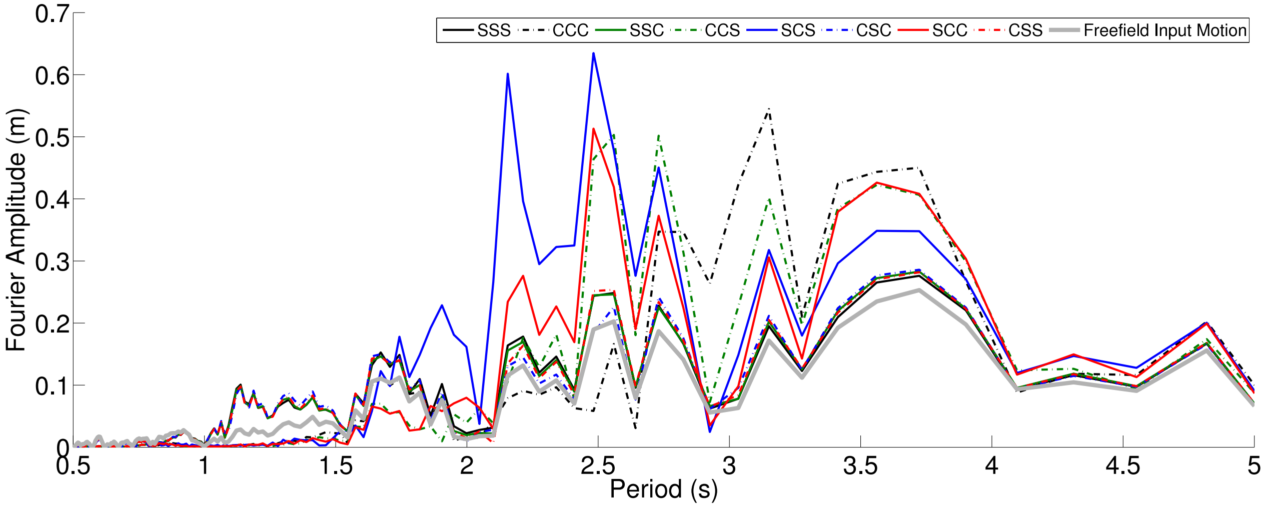


Figure 503.24: Simulated Displacement Response Spectra, Northridge 1994, Century City, Comparison of Eight Cases with Free Field Motions (Soil Block 2)

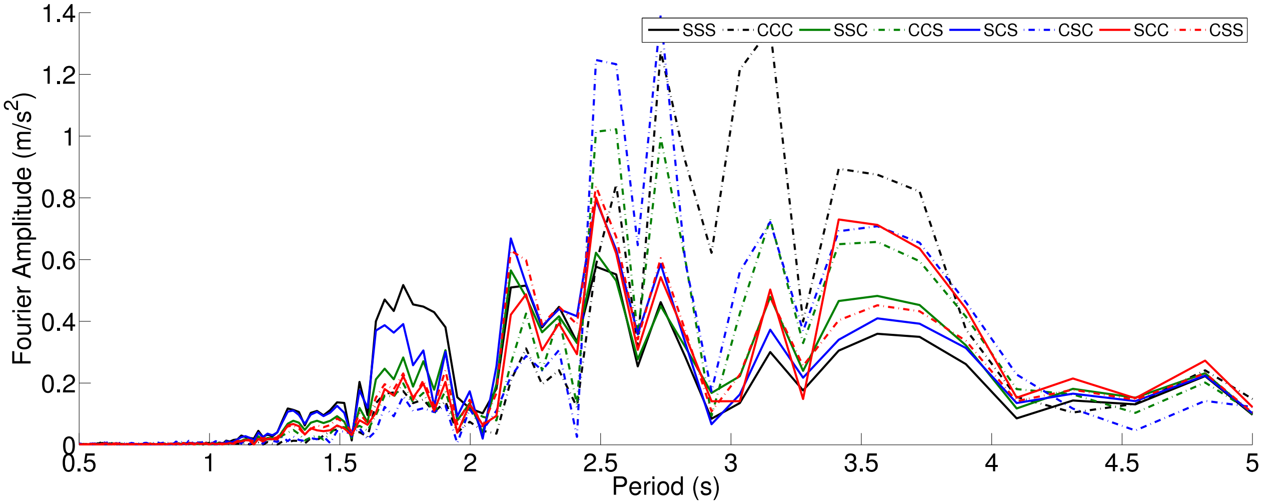


Figure 503.25: Simulated Displacement Response Spectra, Northridge 1994, Century City, Comparison of Eight Cases (Structure Bent 2)

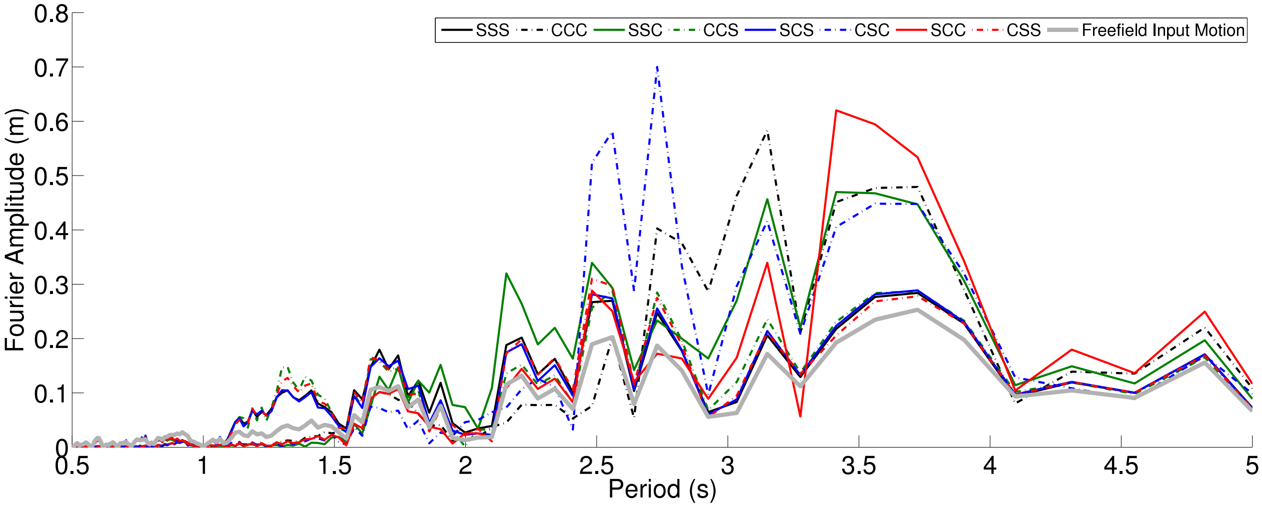


Figure 503.26: Simulated Displacement Response Spectra, Northridge 1994, Century City, Comparison of Eight Cases with Free Field Motions (Soil Block 3)

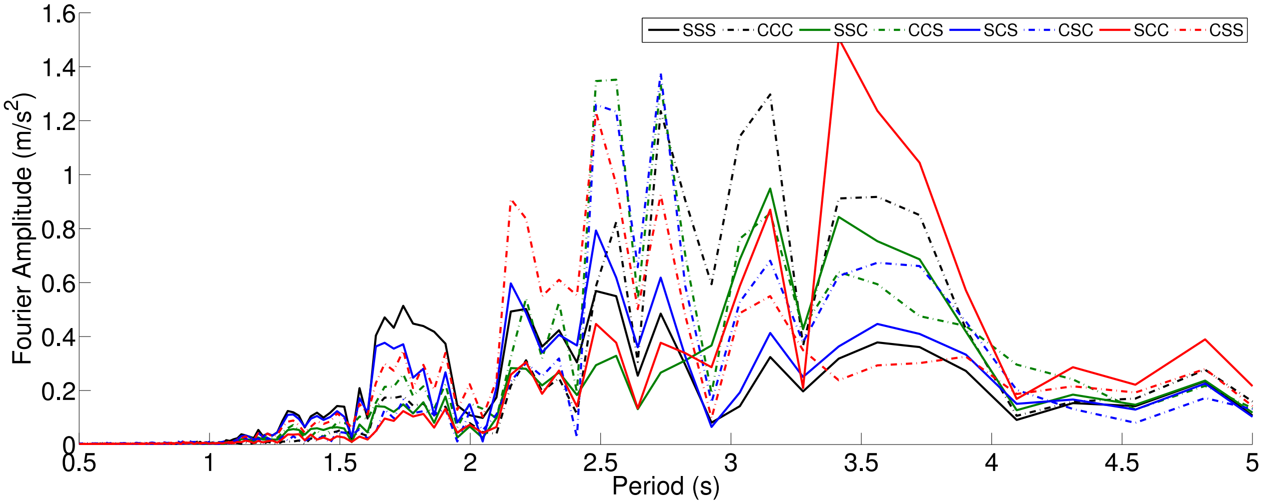


Figure 503.27: Simulated Displacement Response Spectra, Northridge 1994, Century City, Comparison of Eight Cases (Structure Bent 3)

503.2.2.5 Acceleration Response Spectra

The acceleration spectra have also been plotted to support the observation we made before. The consistent discovery is that stiffer soil will have amplification concentrated to the lower period side. If the input motion also have lower period contents, those will amplify the response the structure can see.

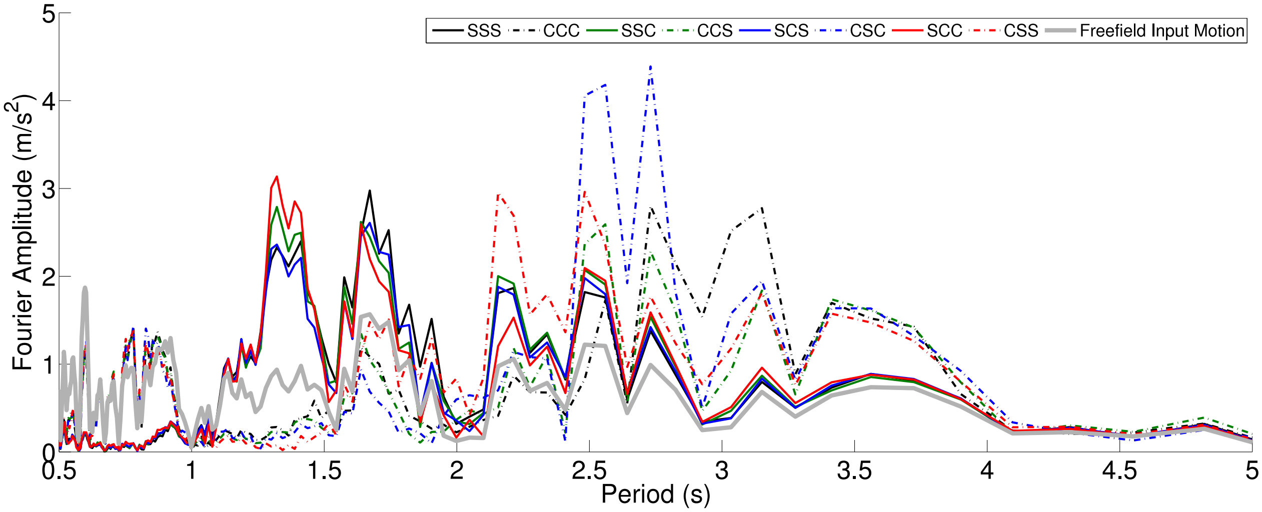


Figure 503.28: Simulated Acceleration Response Spectra, Northridge 1994, Century City, Comparison of Eight Cases with Free Field Motions (Soil Block 1)

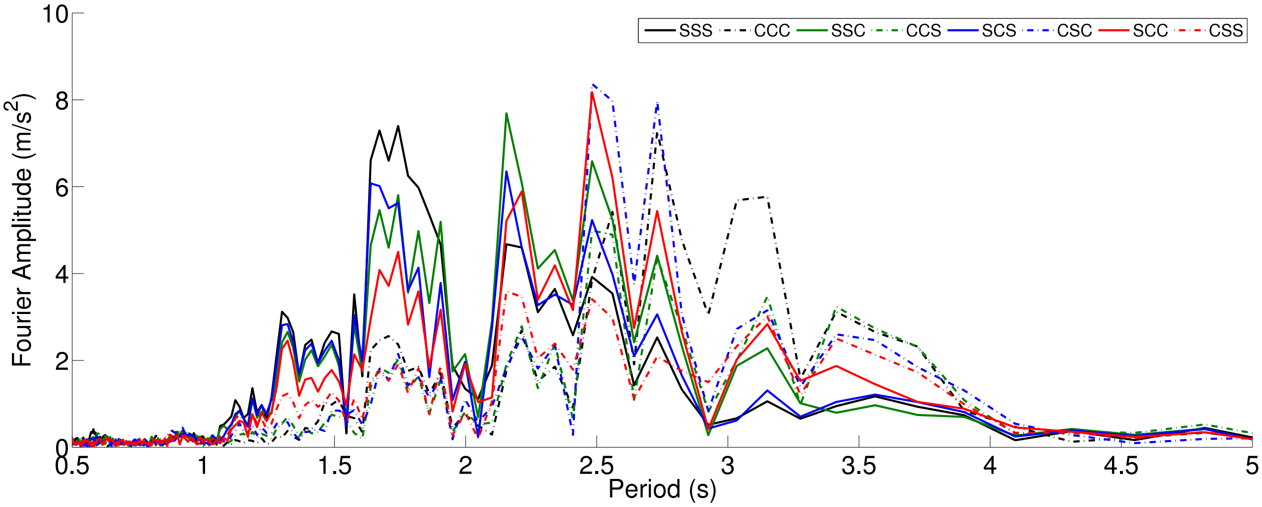


Figure 503.29: Simulated Acceleration Response Spectra, Northridge 1994, Century City, Comparison of Eight Cases (Structure Bent 1)

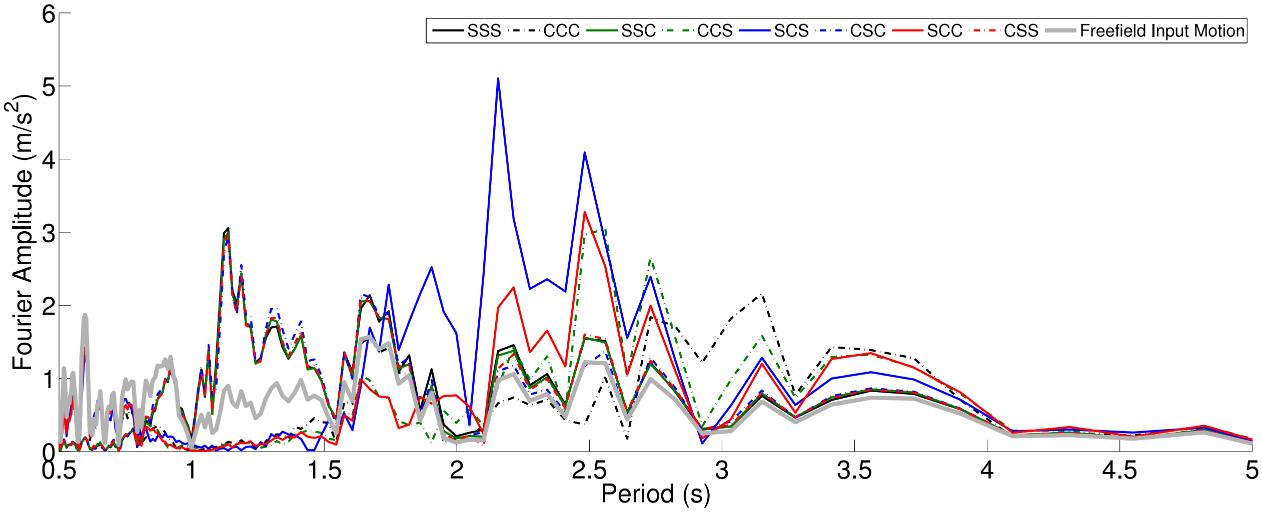


Figure 503.30: Simulated Acceleration Response Spectra, Northridge 1994, Century City, Comparison of Eight Cases with Free Field Motions (Soil Block 2)

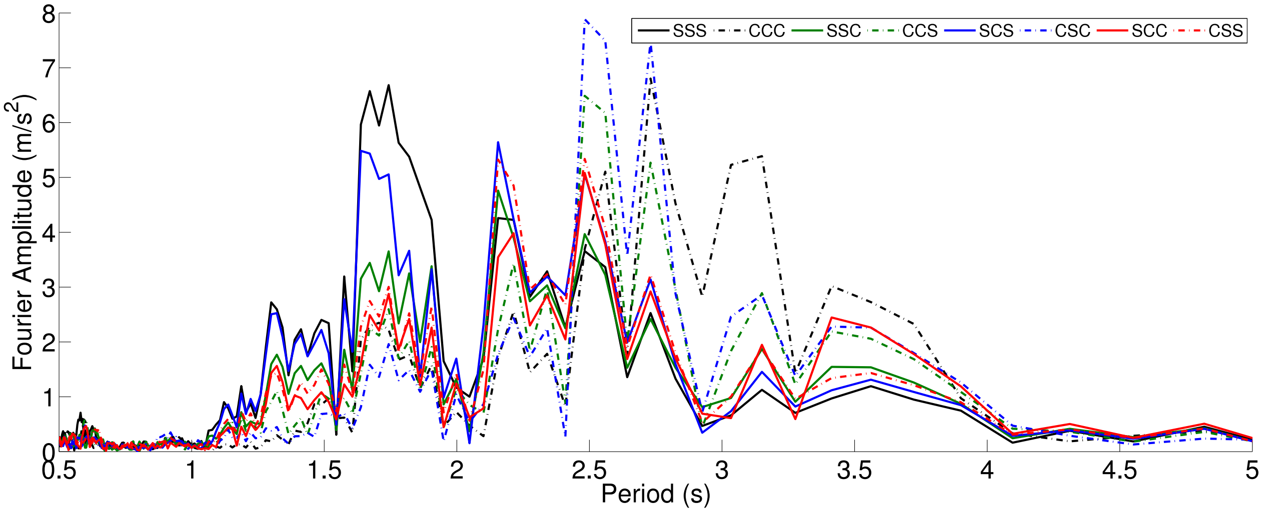


Figure 503.31: Simulated Acceleration Response Spectra, Northridge 1994, Century City, Comparison of Eight Cases (Structure Bent 2)

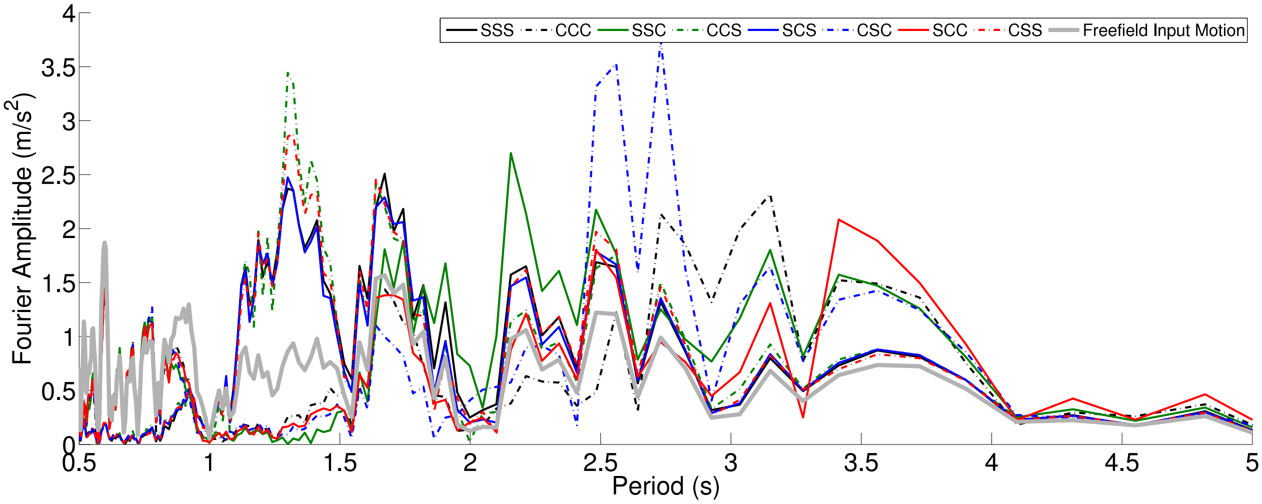


Figure 503.32: Simulated Acceleration Response Spectra, Northridge 1994, Century City, Comparison of Eight Cases with Free Field Motions (Soil Block 3)

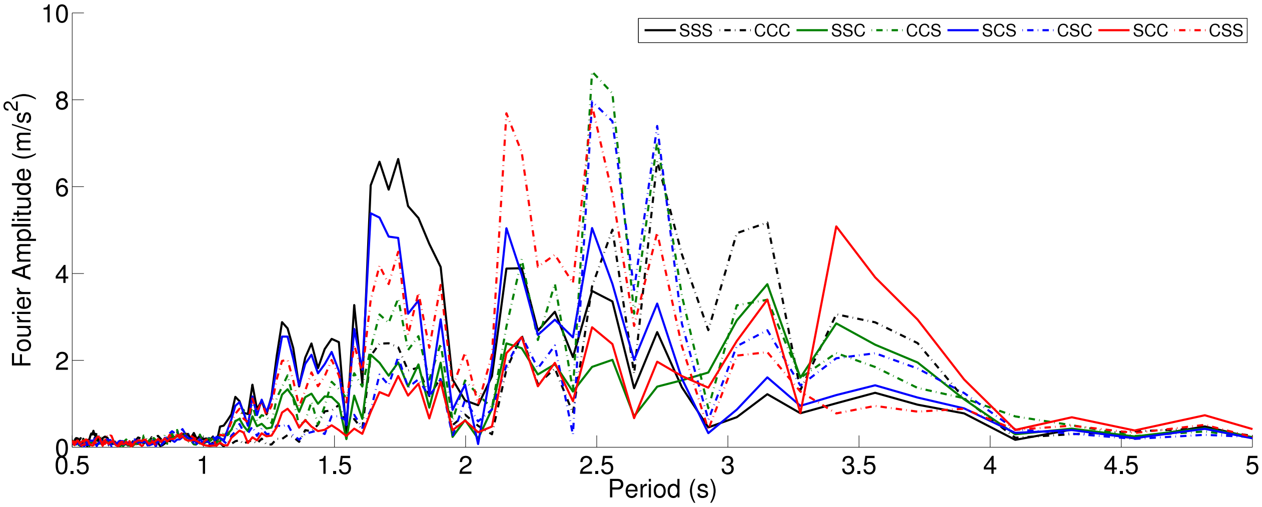


Figure 503.33: Simulated Acceleration Response Spectra, Northridge 1994, Century City, Comparison of Eight Cases (Structure Bent 3)

503.2.2.6 Structural Response

Finally we come to the point that we can see exactly how the structure responds to excitation mechanically. The moment time series shown here tells some important stories during dynamic shaking. Firstly, the structure on top of stiffer soil will yield much faster than those in soft soils. This makes perfect sense after the observations we made in previous sections. The input motion contains very similar frequency content as the stiff soil so stiff soil and the structure on top of it are excited much more than the soft-soil-structure system. Secondly, The structure on top of soft soil will see larger residual response than the stiffer soil. This exactly tells the story that the soft soil will respond much largely to the long period content of input motions which is much closer to the natural period of the soft soil.

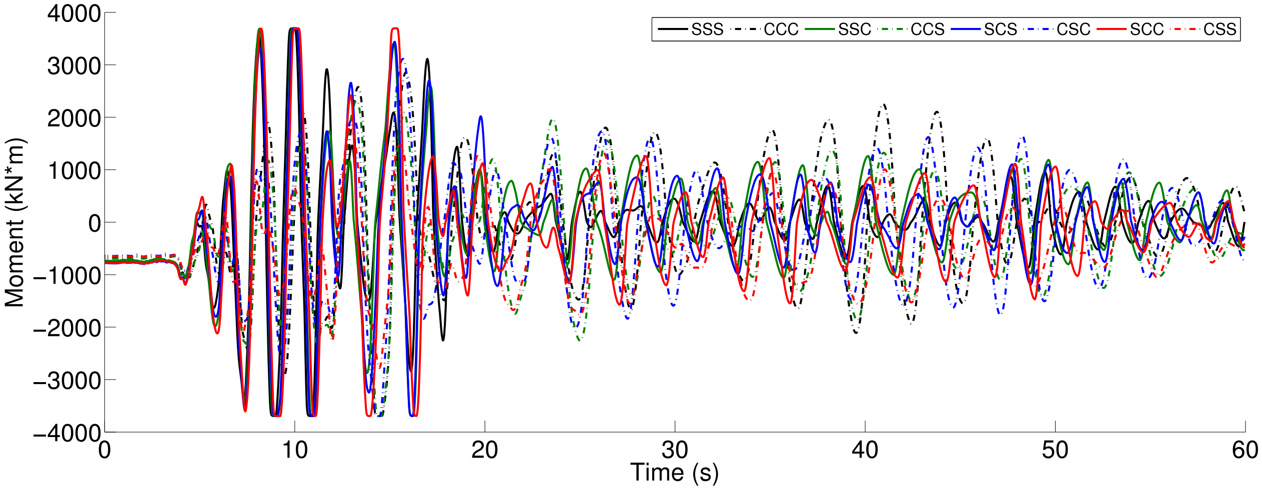


Figure 503.34: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Eight Cases (Structure Bent 1 Pile 1)

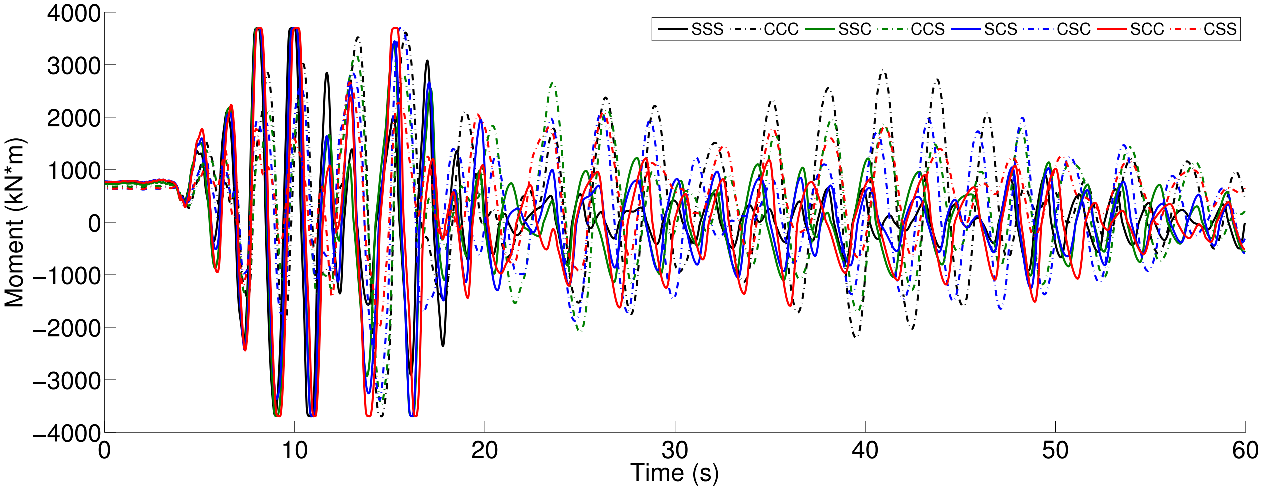


Figure 503.35: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Eight Cases (Structure Bent 1 Pile 2)

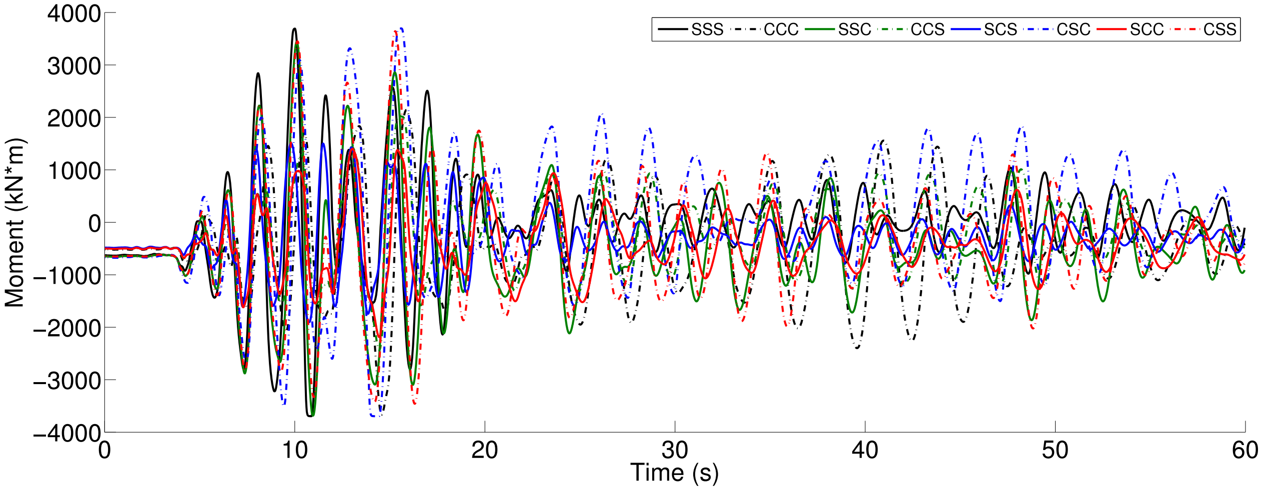


Figure 503.36: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Eight Cases (Structure Bent 2 Pile 1)

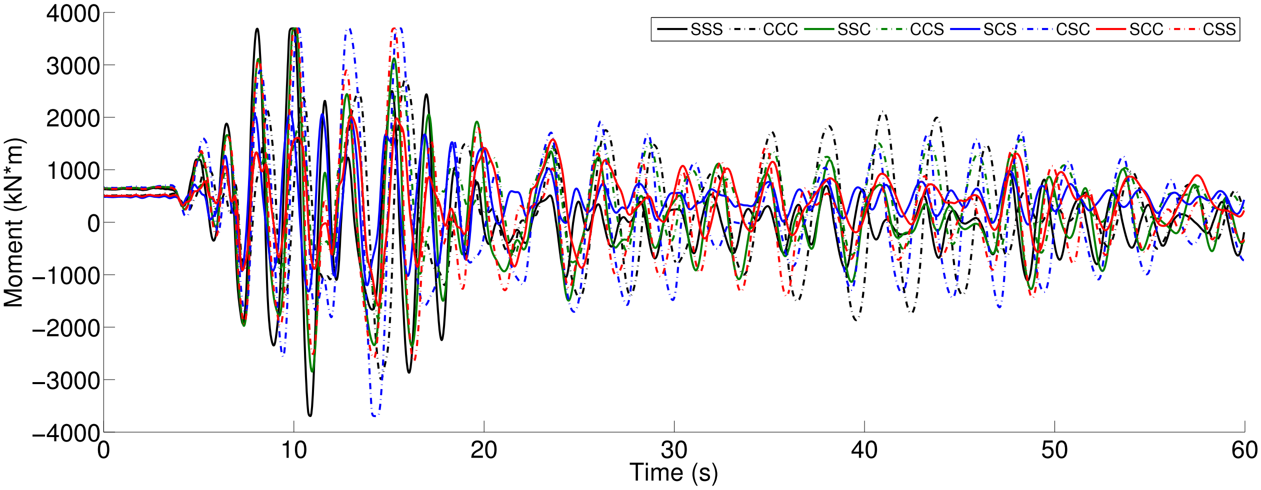


Figure 503.37: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Eight Cases (Structure Bent 2 Pile 2)

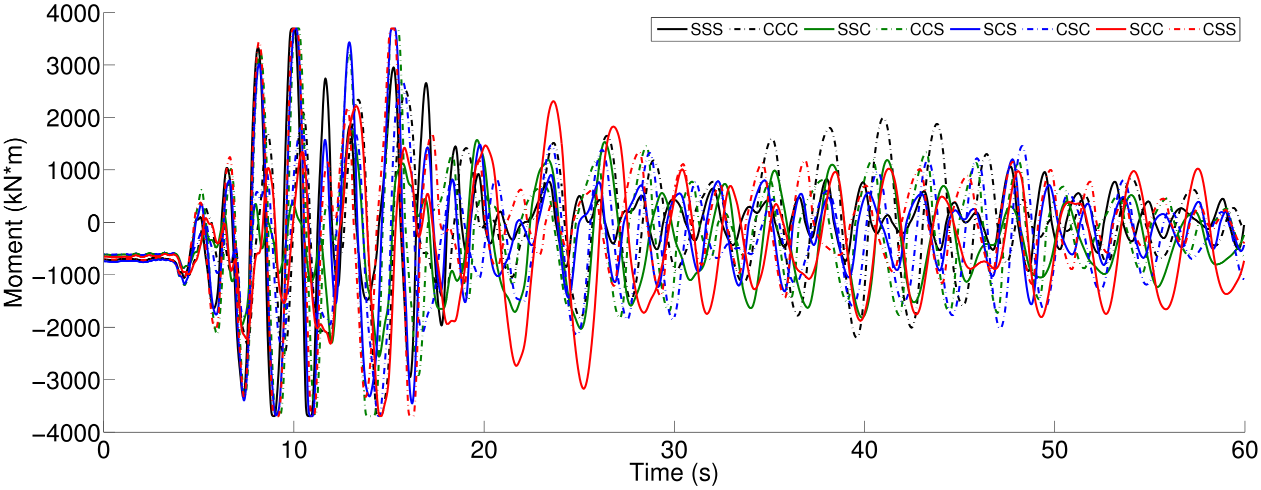


Figure 503.38: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Eight Cases (Structure Bent 3 Pile 1)

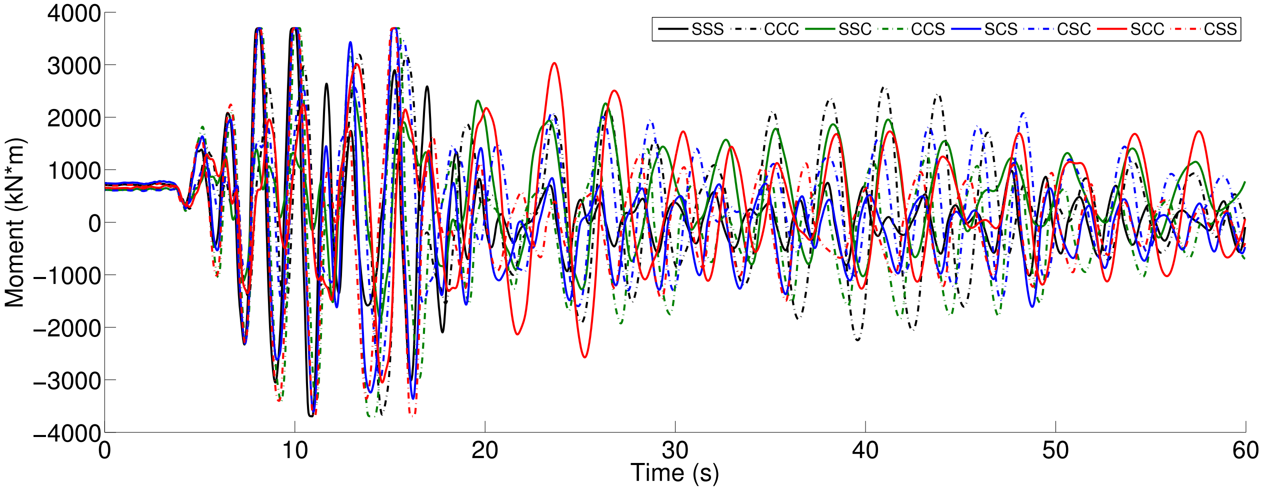


Figure 503.39: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Eight Cases (Structure Bent 3 Pile 2)

503.2.3 Earthquake Simulations - 1999 Turkey Kocaeli

It has been well observed that the characteristics of input ground motion also affects the SFSI system response. From the perspective of soil dynamics, stiffer soil will have shorted natural period and thus higher natural frequency. One can argue that the conclusion that has been established in Section 503.2.4.1 might exactly reflect the case that for earthquake input motions containing much high frequency content, the stiff soil will always receive much stronger shaking. The 1994 Northridge earthquake has been known to contain very high frequency component as shown in Figure 503.40.

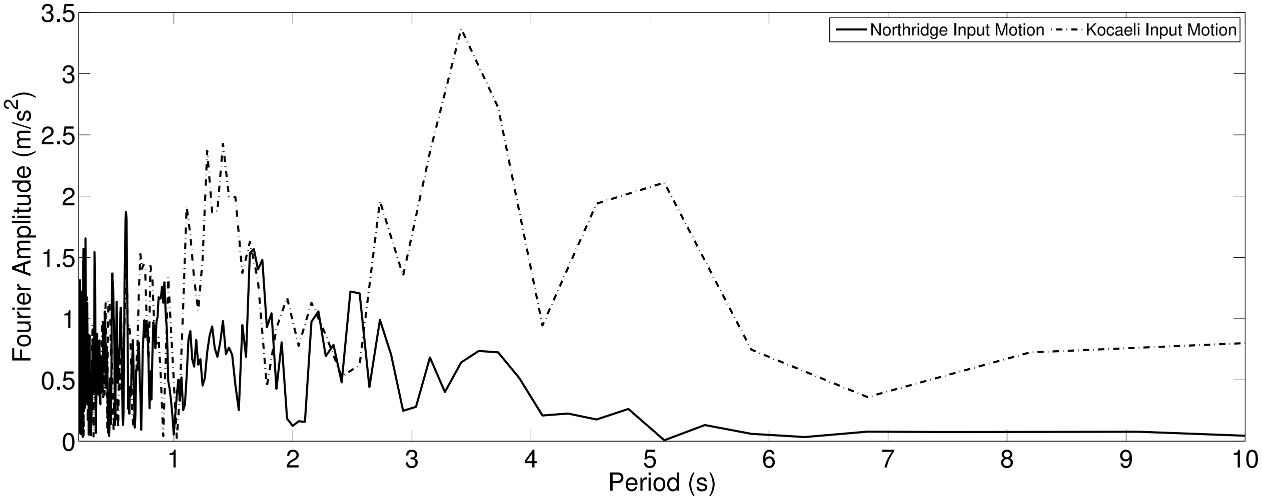


Figure 503.40: Frequency Contents of Ground Motions - Acceleration Time Series

So it will be also legitimate to question that if the ground motion has much long frequency (long period) content, the structure supported by soft soil might be the one to be exposed.

In this work, the question has been studied using the exact finite element models that we created in the previous sections.

The 1999 Turkey Kocaeli earthquake motion recorded at station Yarimca (YPT330) has been used as the target long period motion to study the SFSI behavior with different soil profiles.

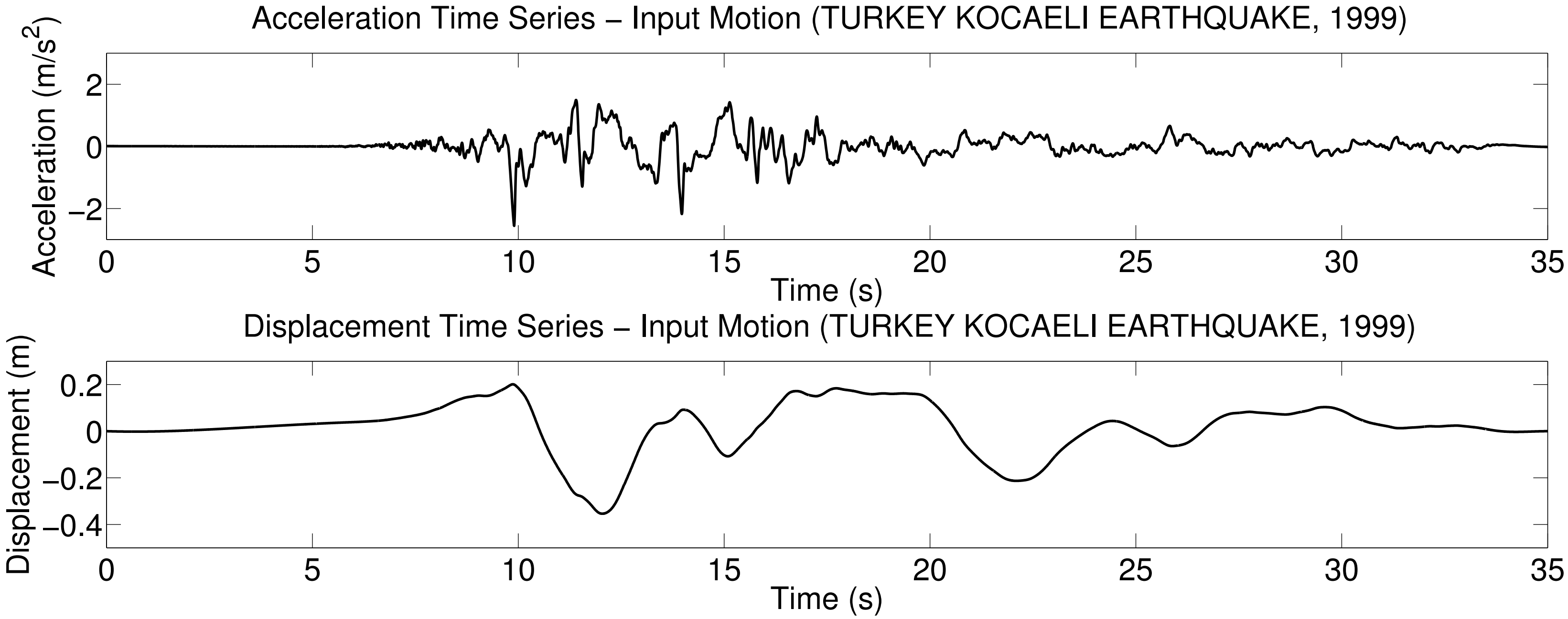


Figure 503.41: Input Motion - Turkey Kocaeli Earthquake 1999

It has been shown in this work that long period (low frequency) motion will excites stronger response from softer clay soil layers which has a lower natural frequency as opposed to the conclusion we draw before in Section 503.2.4.1.

503.2.3.1 Displacement Response

This section is designed specifically to expose the effect of input motion on SFSI system responses. From the pictures shown below, you can see that now the structure on top of soft soil will show much larger response from the shaking. The story behind is obvious to explain. Now the input motion from Turkey Kocaeli earthquake contains primary long period content, which is similar to the natural frequency content of the soft soil. During shaking, this underlying resonance excites the response of the whole SFSI to a larger degree. While on the other hand, the stiff soil now is further away from the primary frequency of the input motion. The consequence is that now the stiff soil will not see much excitation, neither will the structure on top of it.

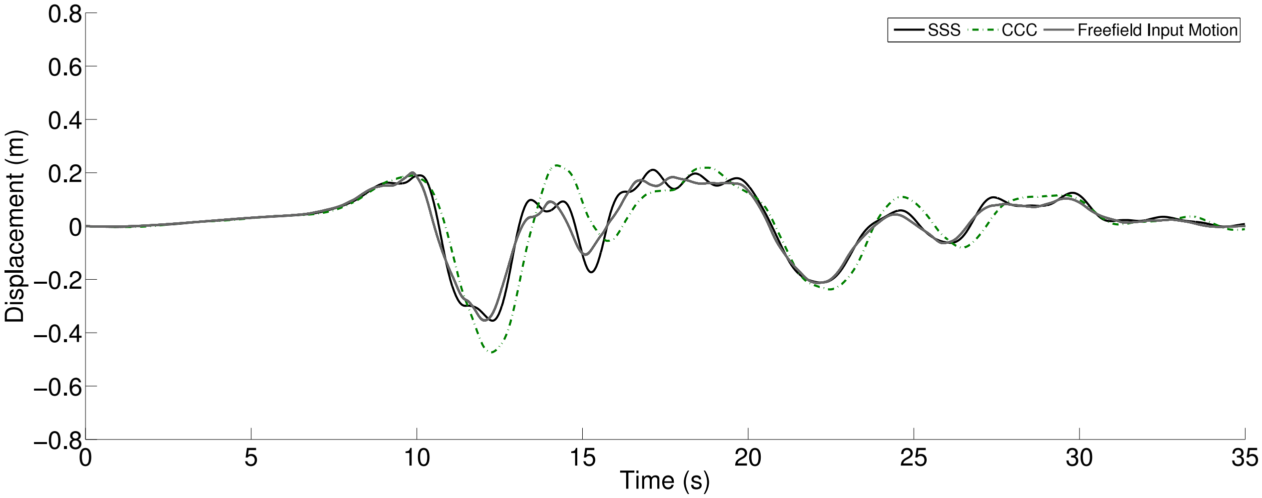


Figure 503.42: Simulated Displacement Time Series, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases with Free Field Motions (Soil Block 1)

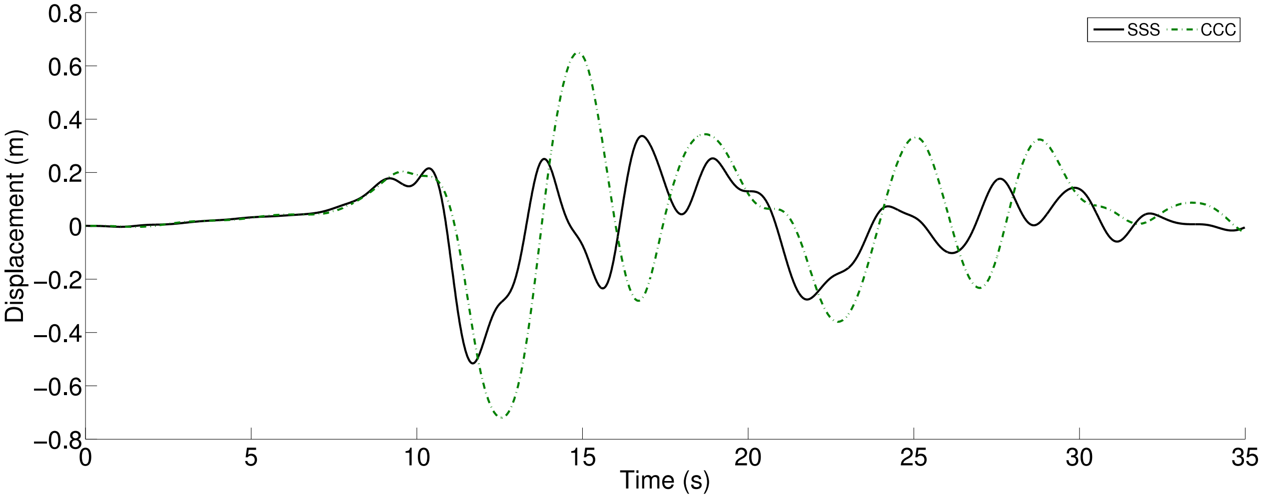


Figure 503.43: Simulated Displacement Time Series, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases (Structure Bent 1)

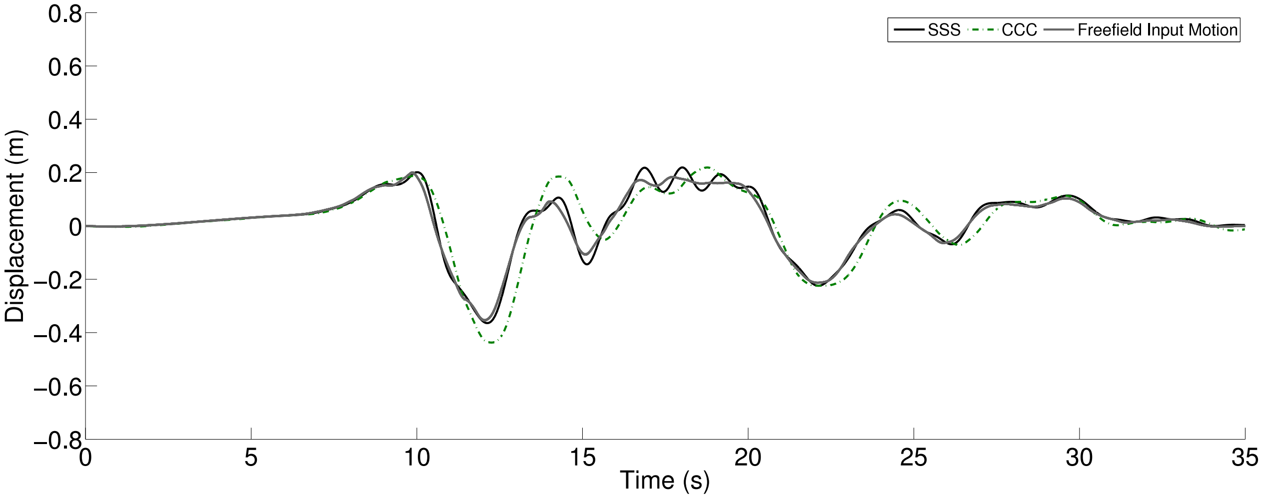


Figure 503.44: Simulated Displacement Time Series, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases with Free Field Motions (Soil Block 2)

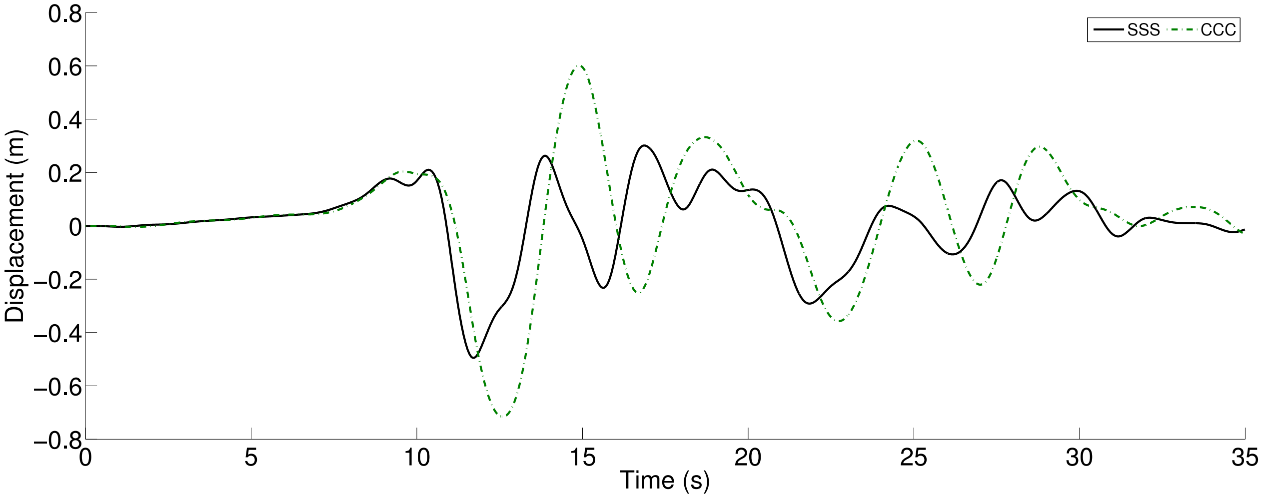


Figure 503.45: Simulated Displacement Time Series, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases (Structure Bent 2)

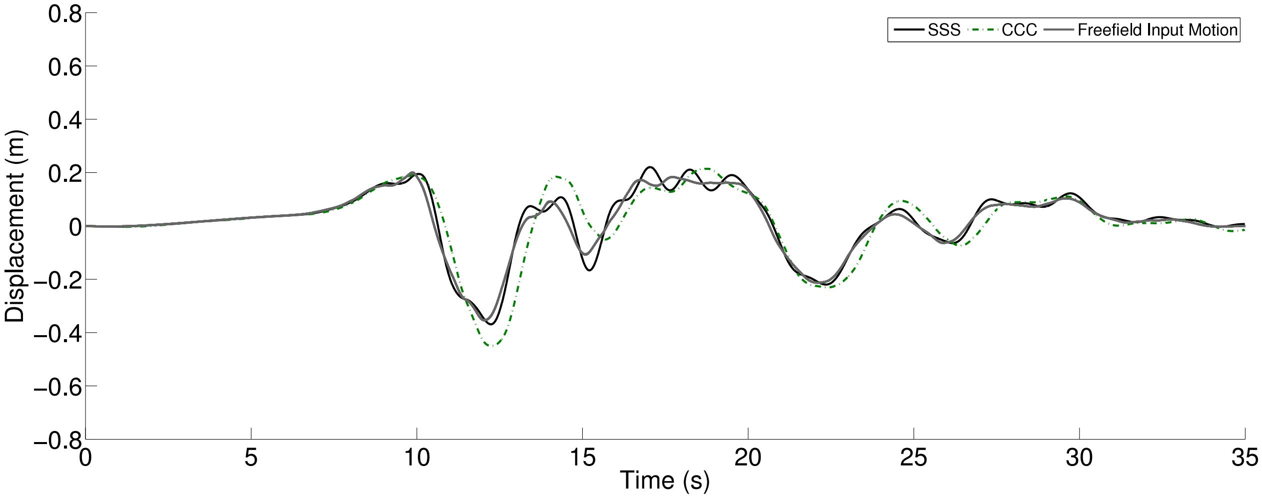


Figure 503.46: Simulated Displacement Time Series, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases with Free Field Motions (Soil Block 3)

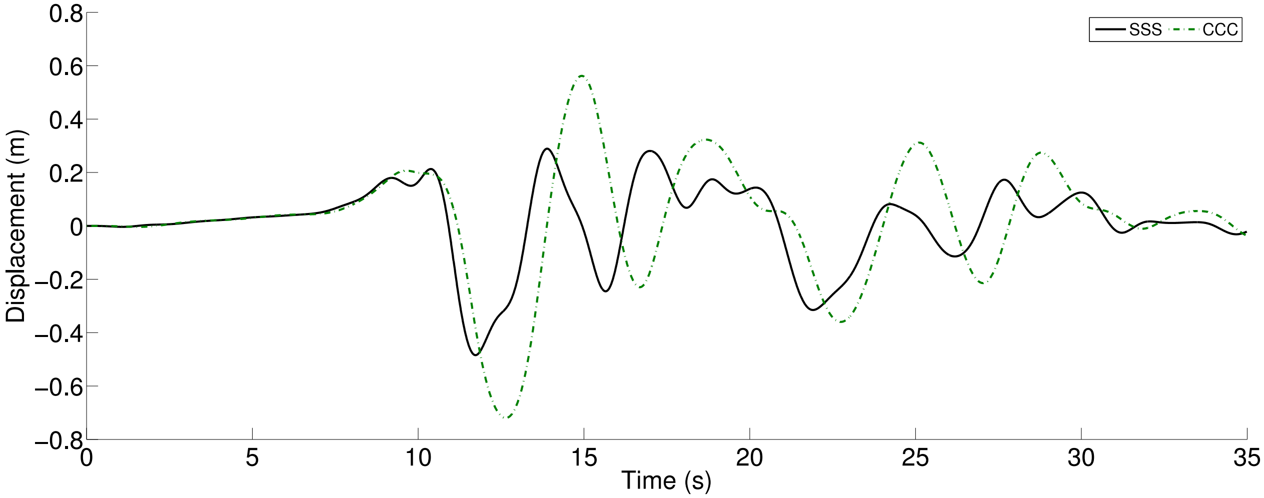


Figure 503.47: Simulated Displacement Time Series, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases (Structure Bent 3)

503.2.3.2 Acceleration Response

Acceleration plots might not show as clearly as those in displacement plots due to the reason that acceleration is derivative of displacement, so acceleration is more responsive to higher frequency contents than lower frequency contents. This effect actually can be observed from any recorded displacement and acceleration spectra records. The acceleration spectra will shift to the low period or high frequency side. This observation will be further explained in later sections when we discuss the moment time series.

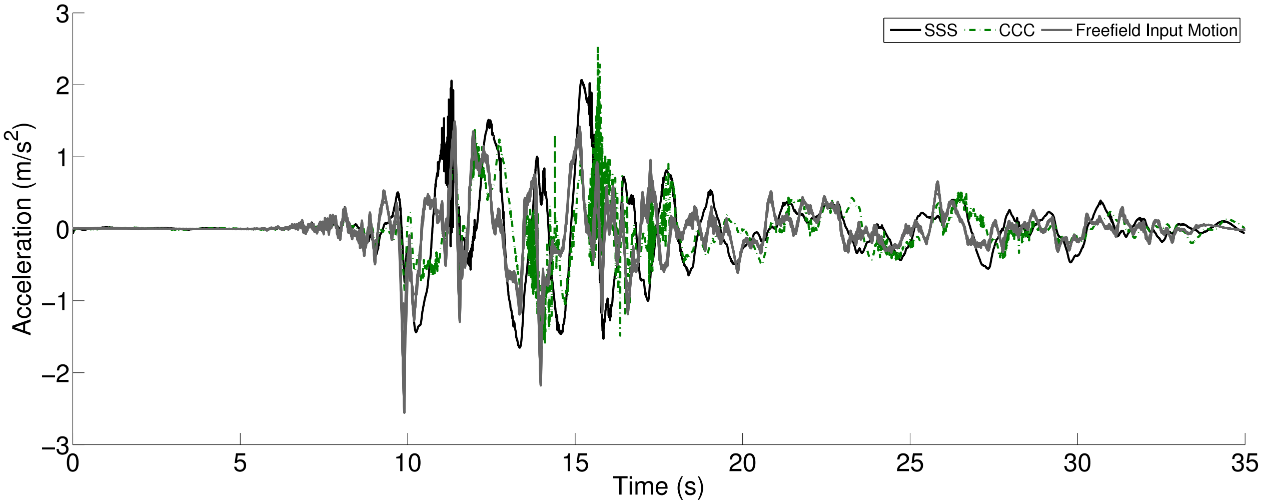


Figure 503.48: Simulated Acceleration Time Series, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases with Free Field Motions (Soil Block 1)

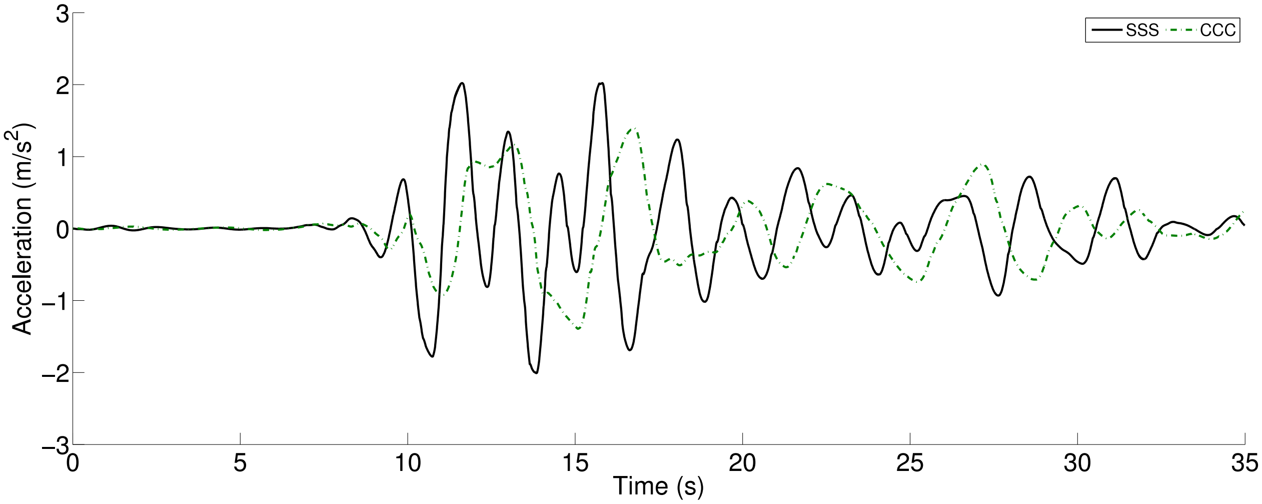


Figure 503.49: Simulated Acceleration Time Series, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases (Structure Bent 1)

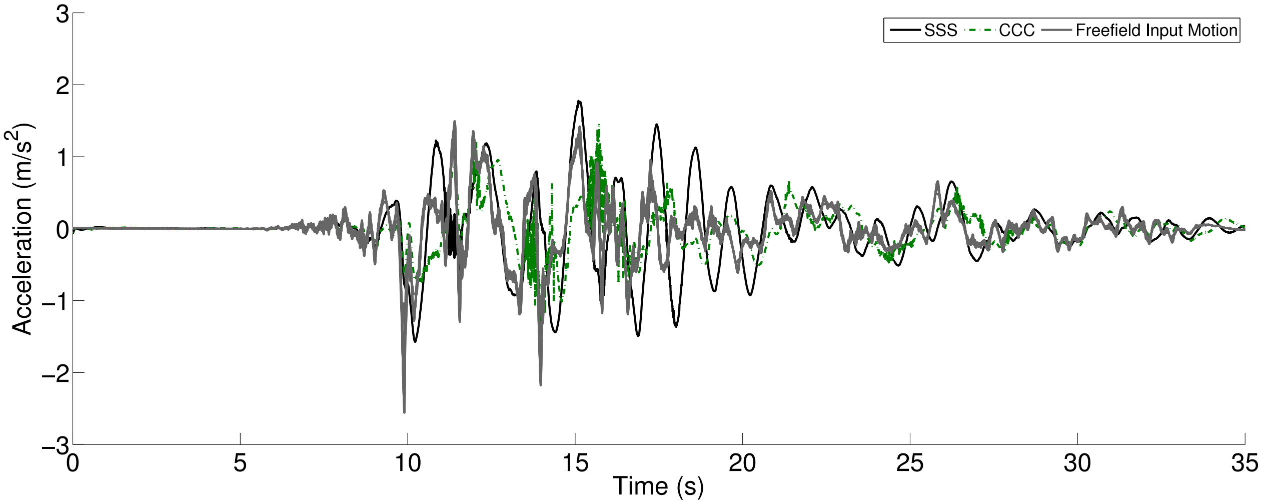


Figure 503.50: Simulated Acceleration Time Series, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases with Free Field Motions (Soil Block 2)

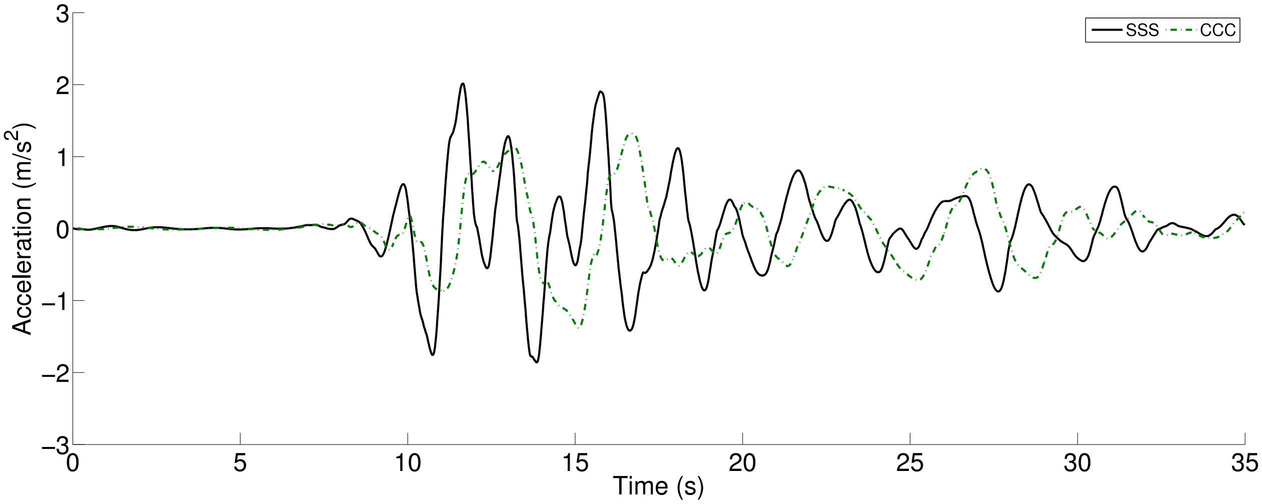


Figure 503.51: Simulated Acceleration Time Series, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases (Structure Bent 2)

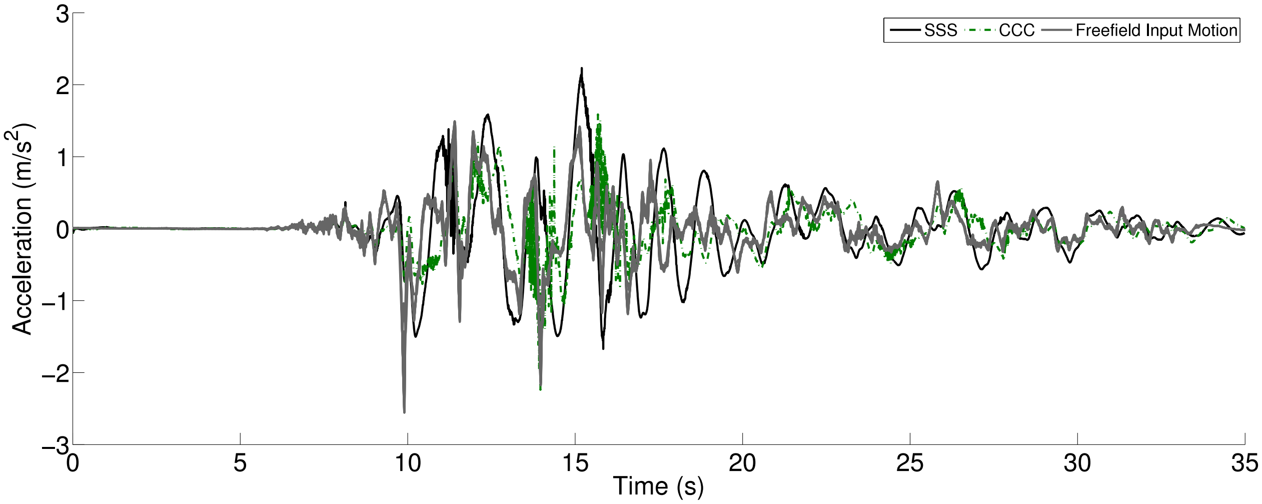


Figure 503.52: Simulated Acceleration Time Series, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases with Free Field Motions (Soil Block 3)

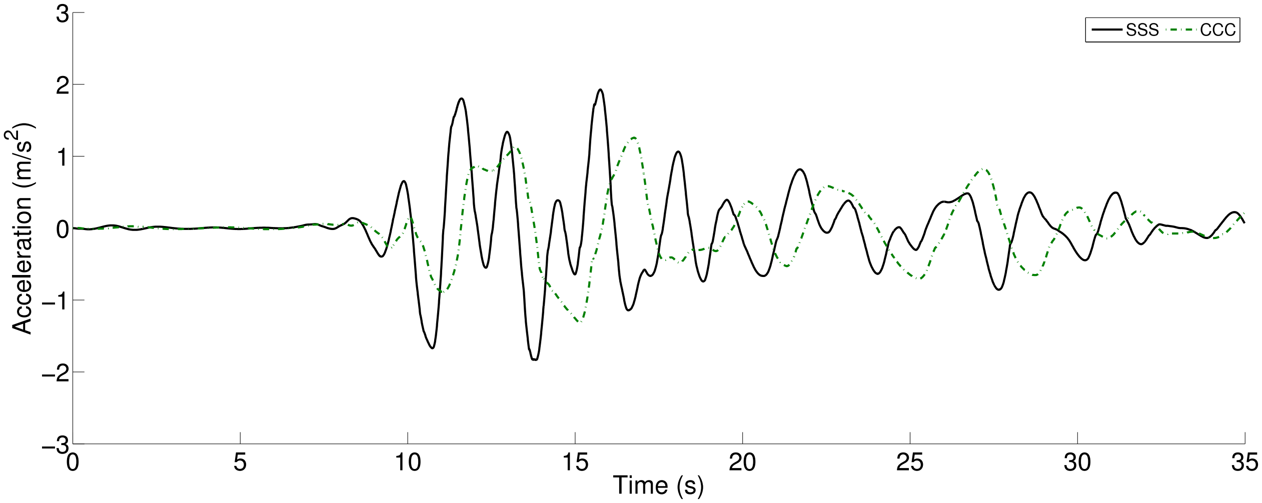


Figure 503.53: Simulated Acceleration Time Series, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases (Structure Bent 3)

503.2.3.3 Displacement Response Spectra

The response spectra are useful tools to look deep into the frequency domain. As we explained before, the Turkey Kocaeli shaking was picked for this study because it contains much longer period contents that were not present in the 1999 Northridge earthquake. In this section, we can clearly see how the soil foundation can affect the structures on top of it. Firstly, structures supported by stiff soil will only see amplification from those low period, high frequency components. Correspondingly, structures supported by soft soil tend to respond much more to long period, low frequency components of the shaking motion. Secondly, this plots exactly show how important the SFSI analysis is. It will not make much sense if one wants to analyze the structure without acknowledging the characteristics of the underlying soil foudation and the input motion. How the structure behaves is the combination of answers to many questions such as the stiffness of soil and the frequency contents of the motions.

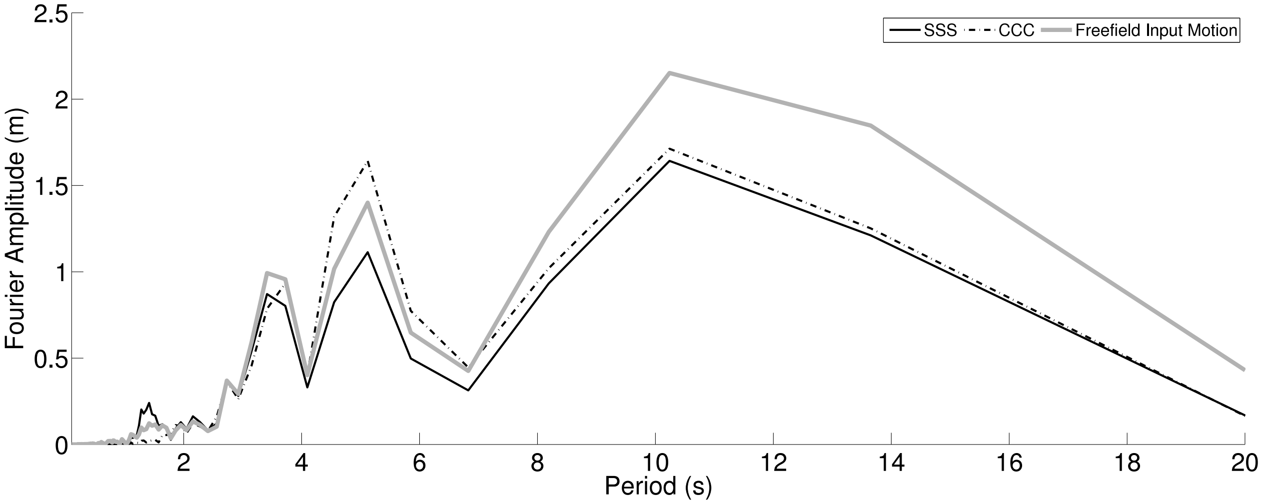


Figure 503.54: Simulated Displacement Response Spectra, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases with Free Field Motions (Soil Block 1)

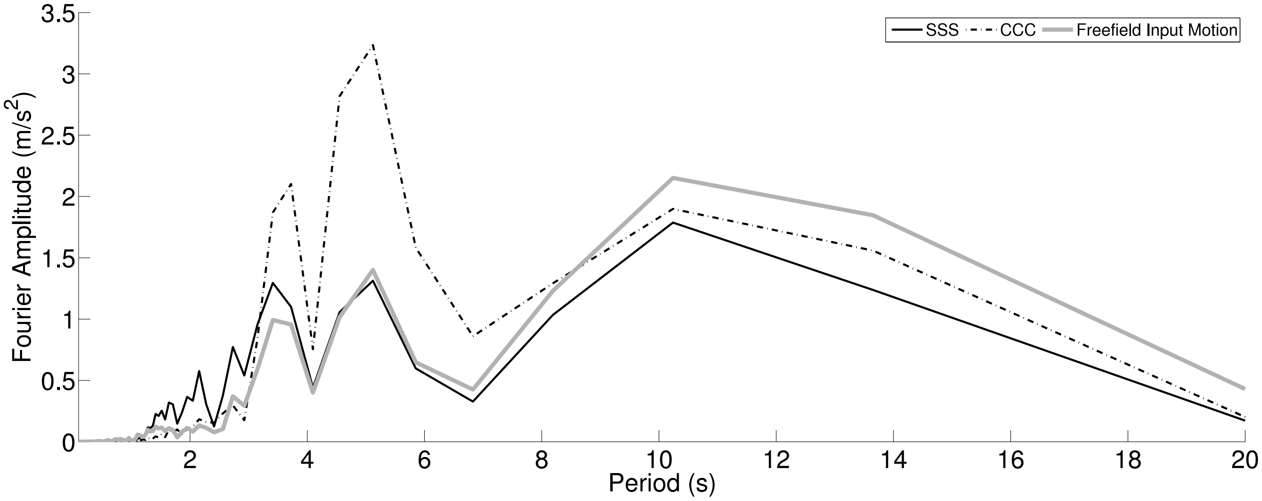


Figure 503.55: Simulated Displacement Response Spectra, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases (Structure Bent 1)

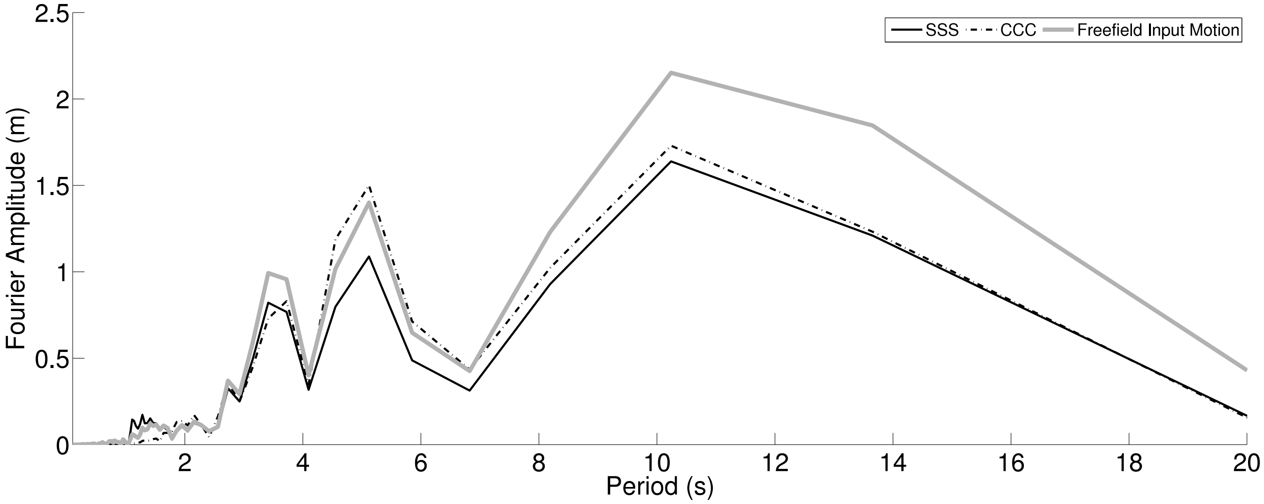


Figure 503.56: Simulated Displacement Response Spectra, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases with Free Field Motions (Soil Block 2)

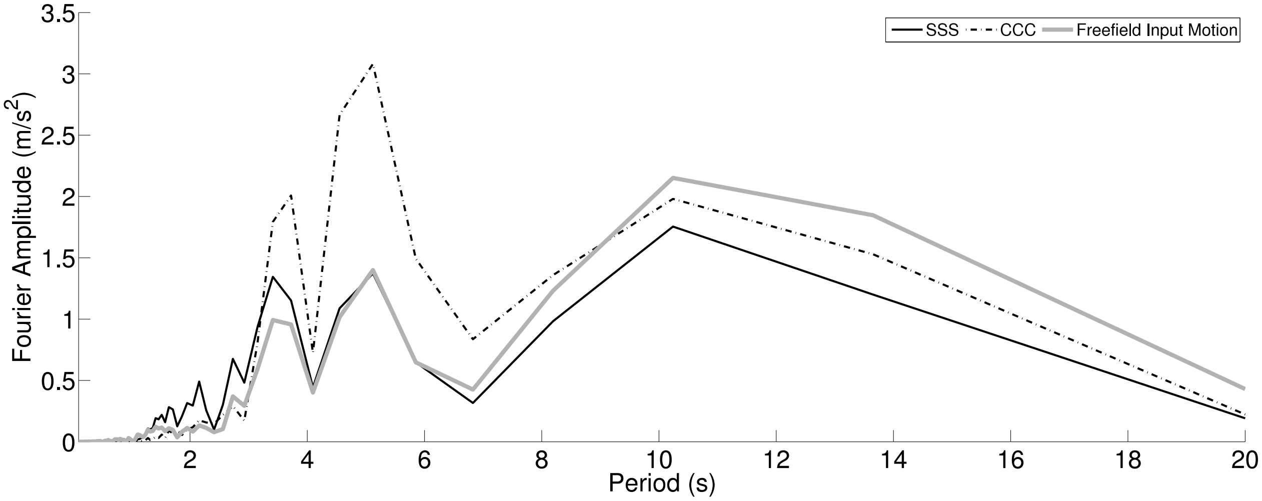


Figure 503.57: Simulated Displacement Response Spectra, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases (Structure Bent 2)

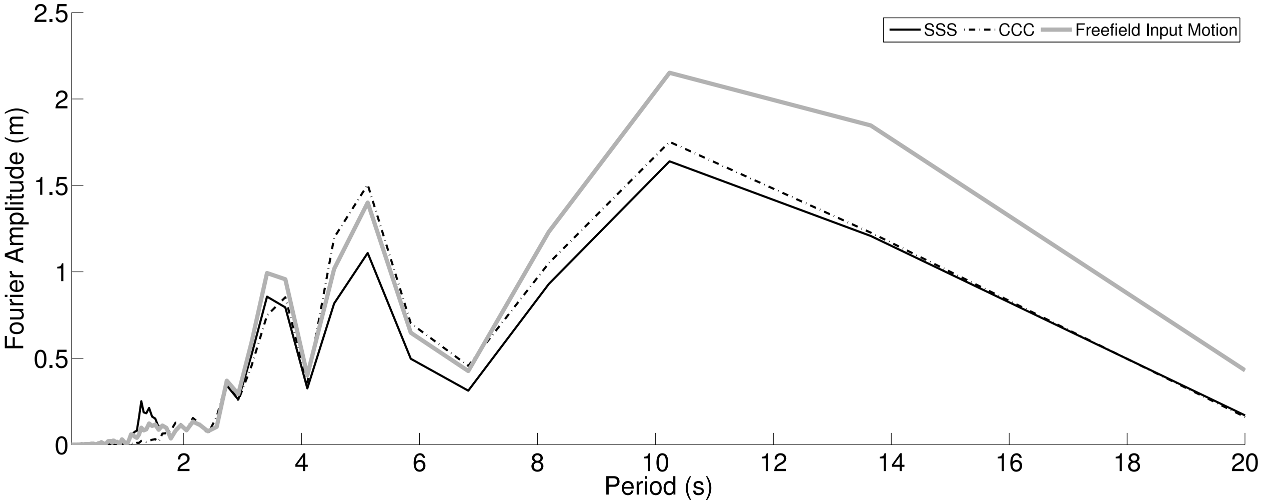


Figure 503.58: Simulated Displacement Response Spectra, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases with Free Field Motions (Soil Block 3)

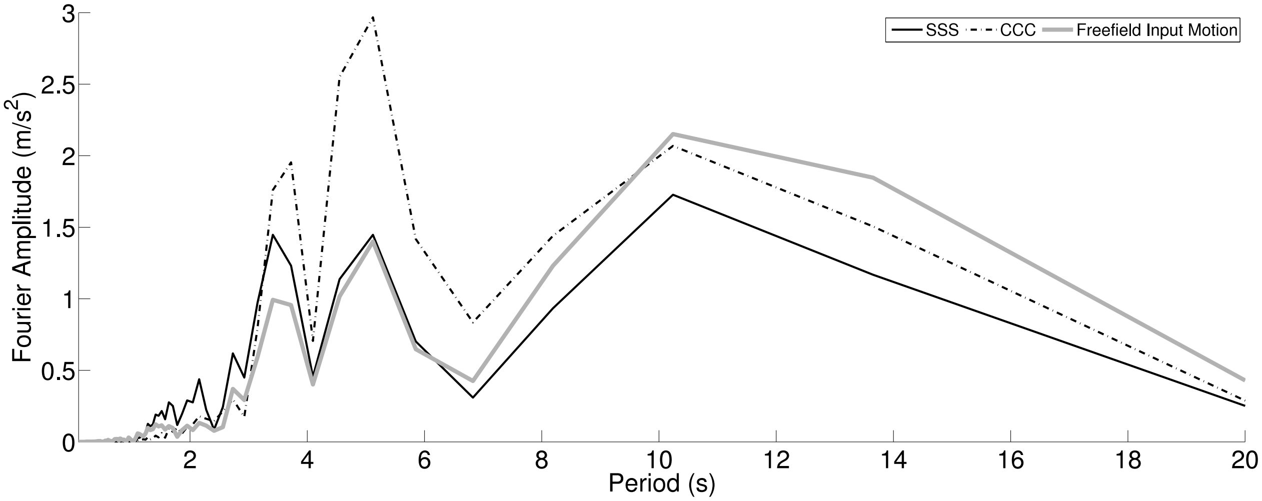


Figure 503.59: Simulated Displacement Response Spectra, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases (Structure Bent 3)

503.2.3.4 Acceleration Response Spectra

As we discussed before, the acceleration response shows much focused content to the high frequency (low period) side as acceleration is the derivative of displacement. But it is still clearly shows the difference between the structures supported by stiff soil and those supported by soft soil. SFSI is crucial in the sense that the structure response must be determined by both short period and long period components. The overall response is determined by the primary periods of both underlying soil and the input motion.

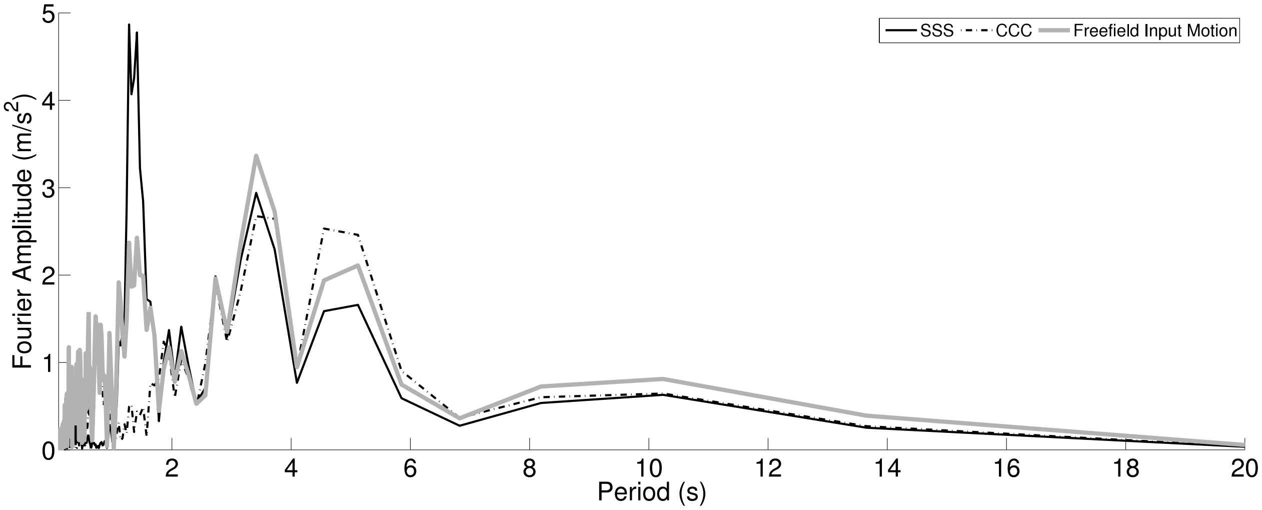


Figure 503.60: Simulated Acceleration Response Spectra, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases with Free Field Motions (Soil Block 1)

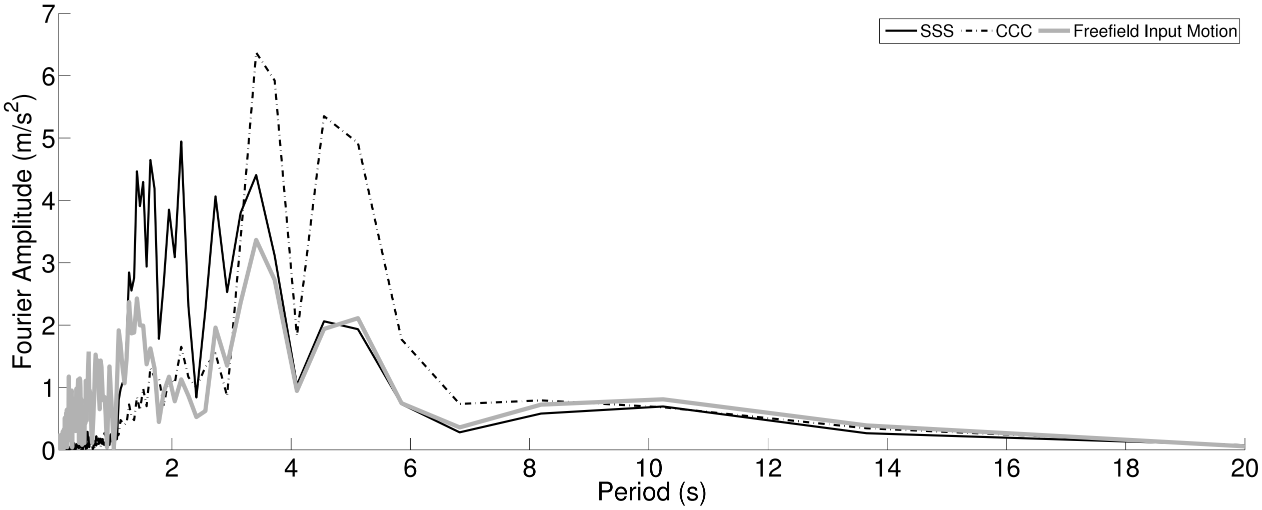


Figure 503.61: Simulated Acceleration Response Spectra, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases (Structure Bent 1)

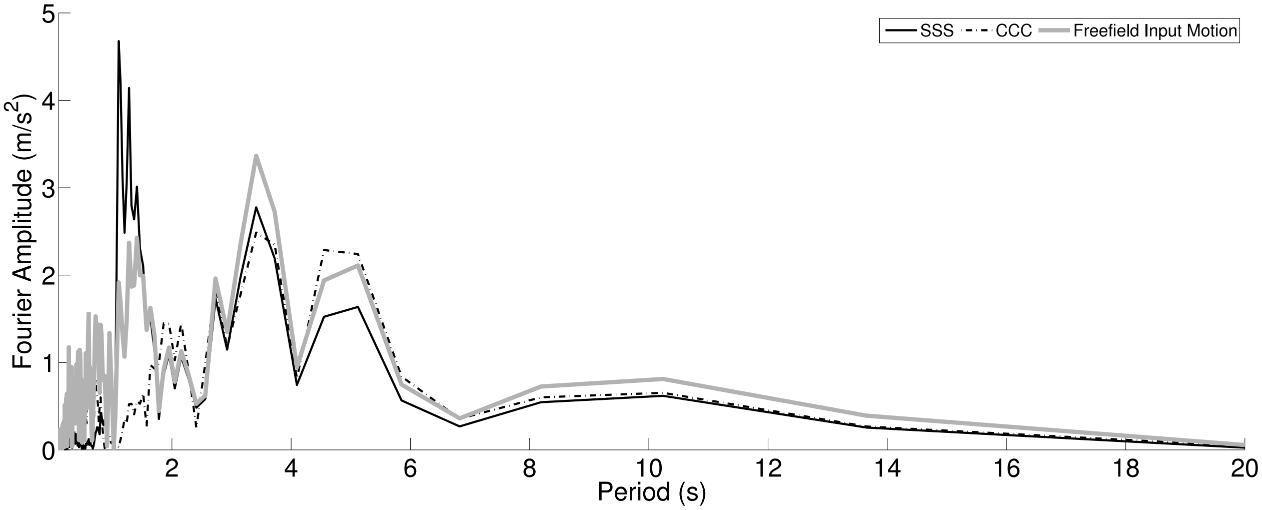


Figure 503.62: Simulated Acceleration Response Spectra, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases with Free Field Motions (Soil Block 2)

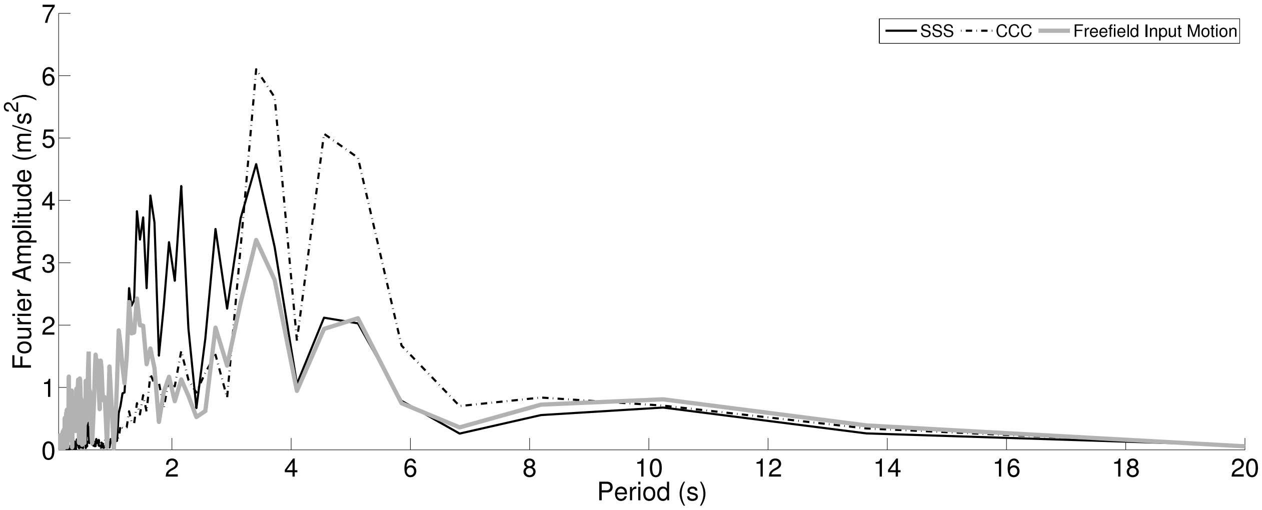


Figure 503.63: Simulated Acceleration Response Spectra, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases (Structure Bent 2)

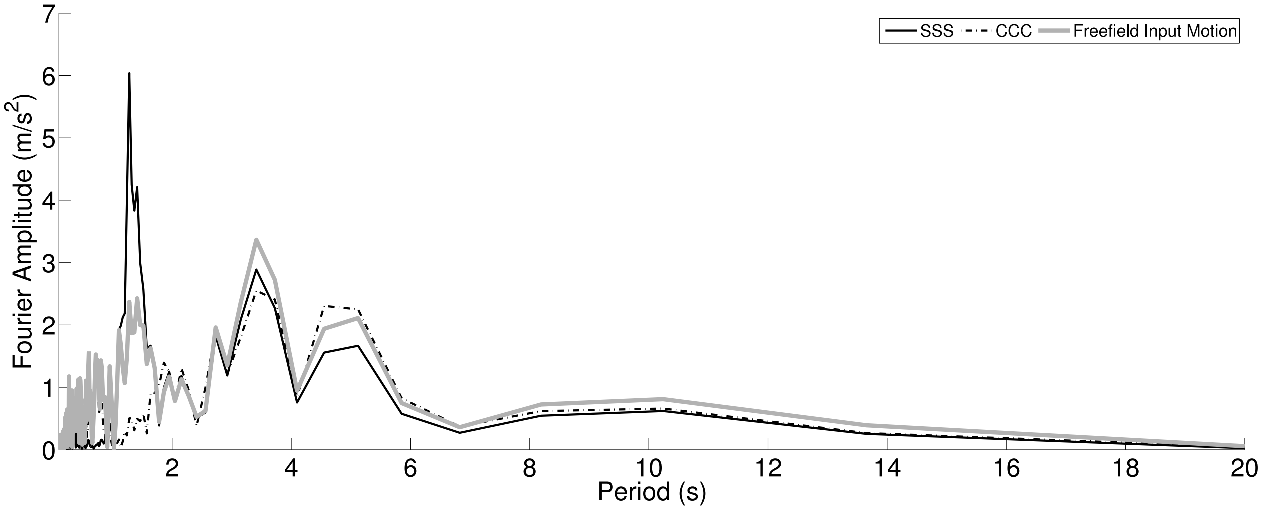


Figure 503.64: Simulated Acceleration Response Spectra, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases with Free Field Motions (Soil Block 3)

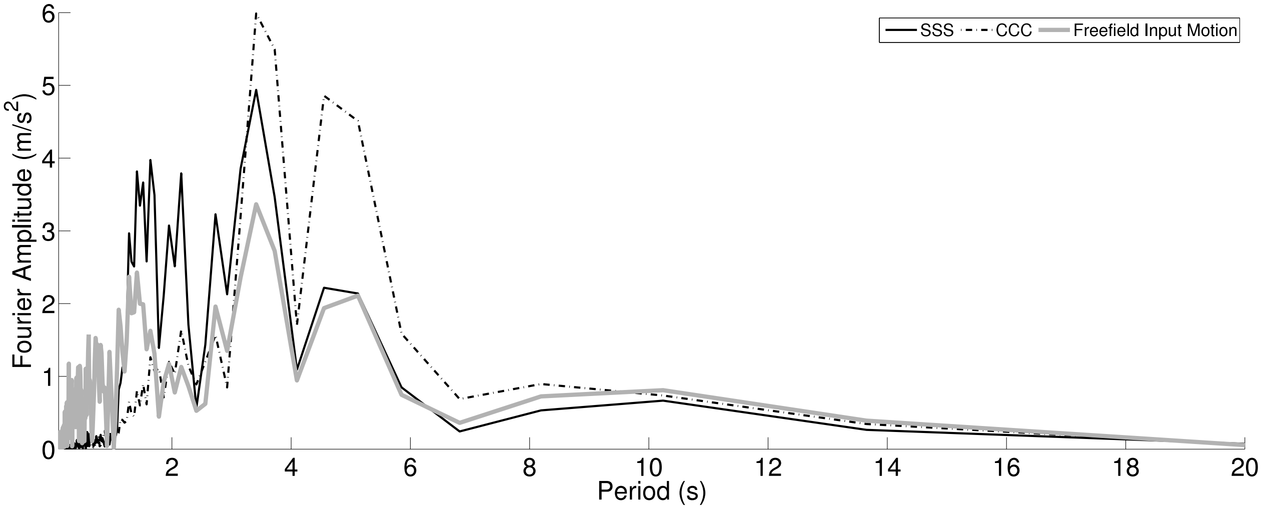


Figure 503.65: Simulated Acceleration Response Spectra, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases (Structure Bent 3)

503.2.3.5 Structural Response

The structure response now directly shows how differently the structure can react to the same input motion, given different soil conditions. We can again see the structure on top of soft soil exhibits much response to long period content especially for the fact that the structure on top of the soft soil reaches maximum moment for about 3 seconds, which is not present in the stiff soil case. This is also consistent with the observation that the displacement of the structure supported by soft soil is much larger due to the much longer time for plastic slip when the plastic moment is reached.

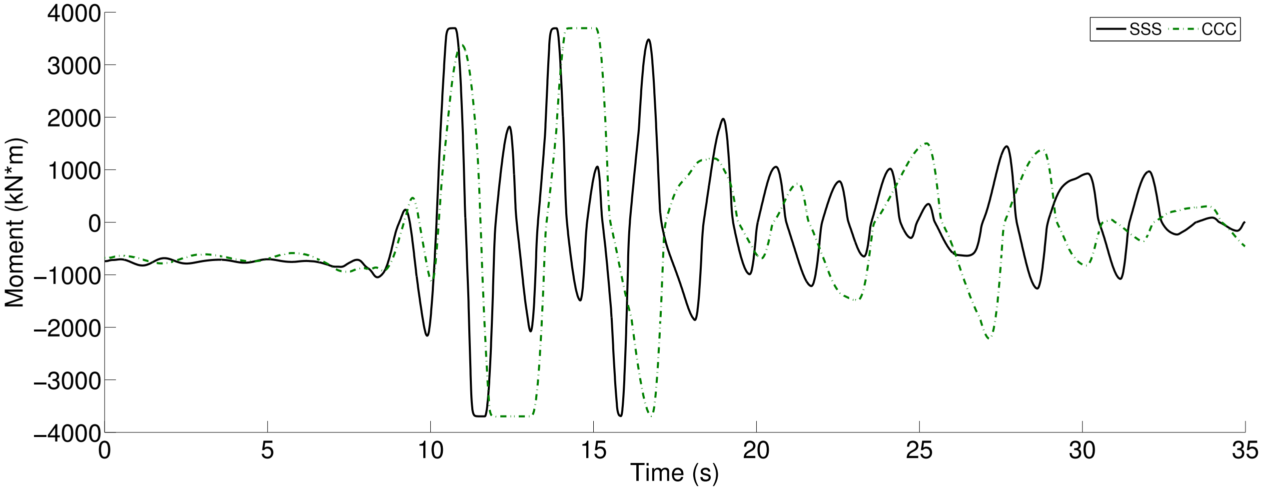


Figure 503.66: Simulated Maximum Moment Time Series, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases (Structure Bent 1 Pile 1)

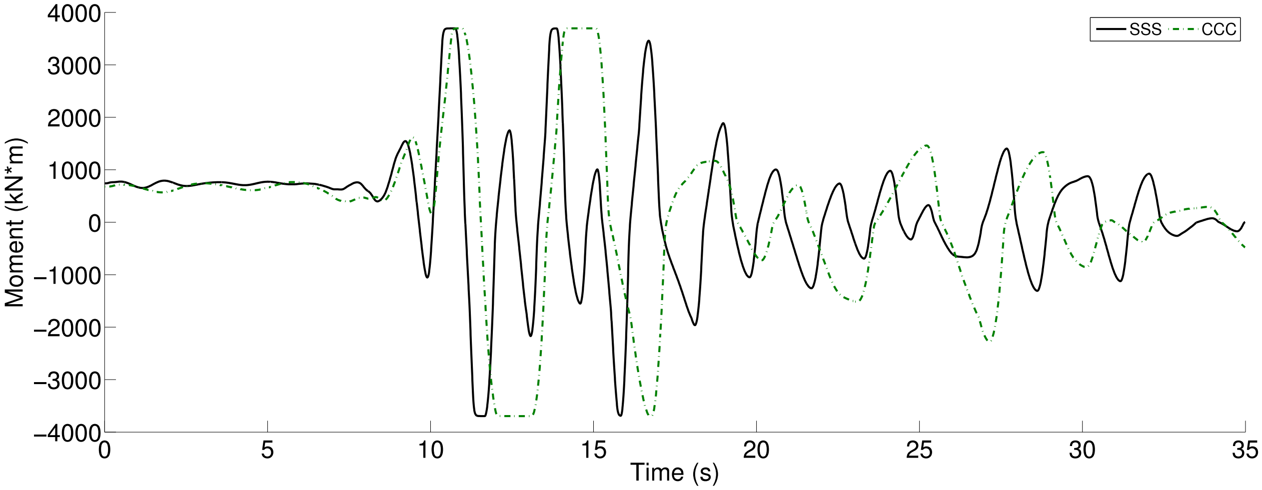


Figure 503.67: Simulated Maximum Moment Time Series, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases (Structure Bent 1 Pile 2)

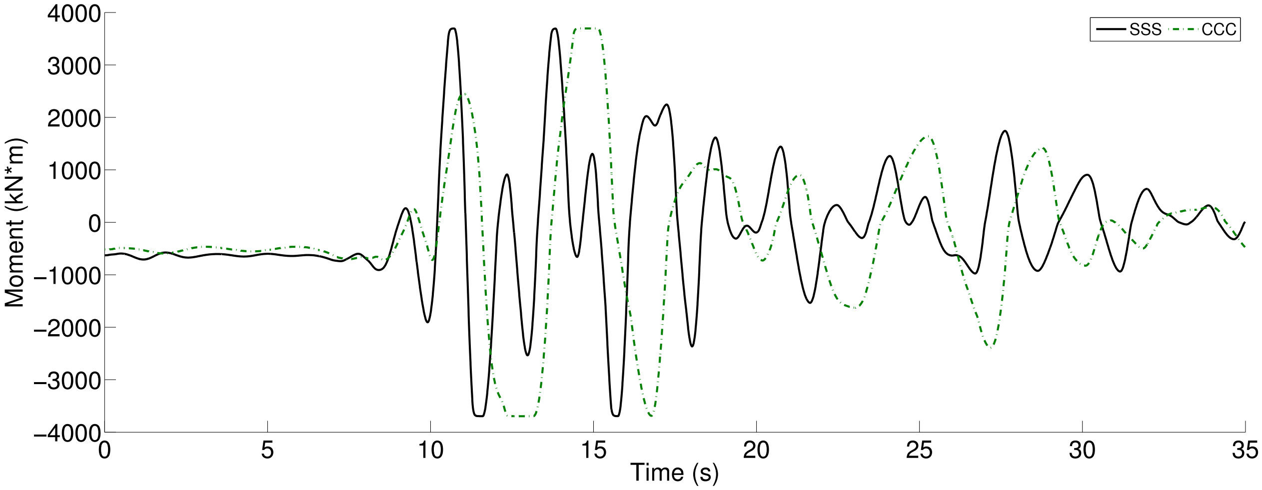


Figure 503.68: Simulated Maximum Moment Time Series, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases (Structure Bent 2 Pile 1)

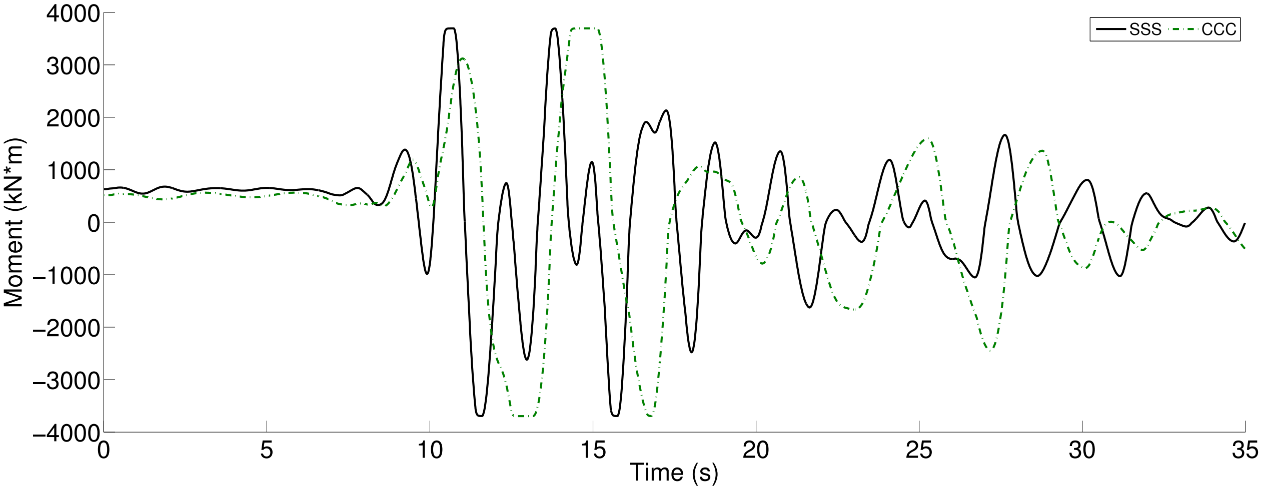


Figure 503.69: Simulated Maximum Moment Time Series, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases (Structure Bent 2 Pile 2)

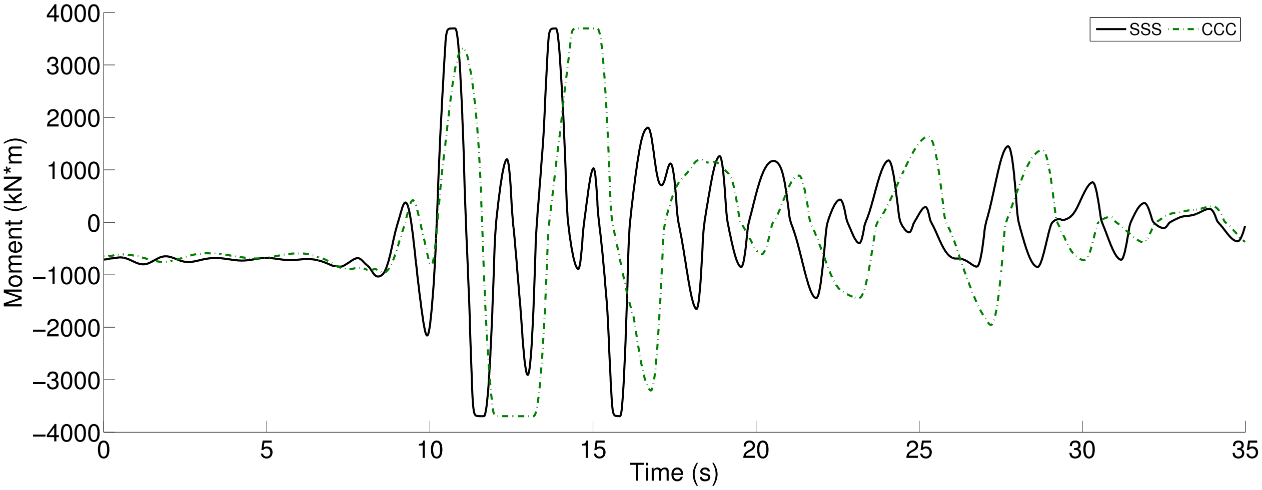


Figure 503.70: Simulated Maximum Moment Time Series, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases (Structure Bent 3 Pile 1)

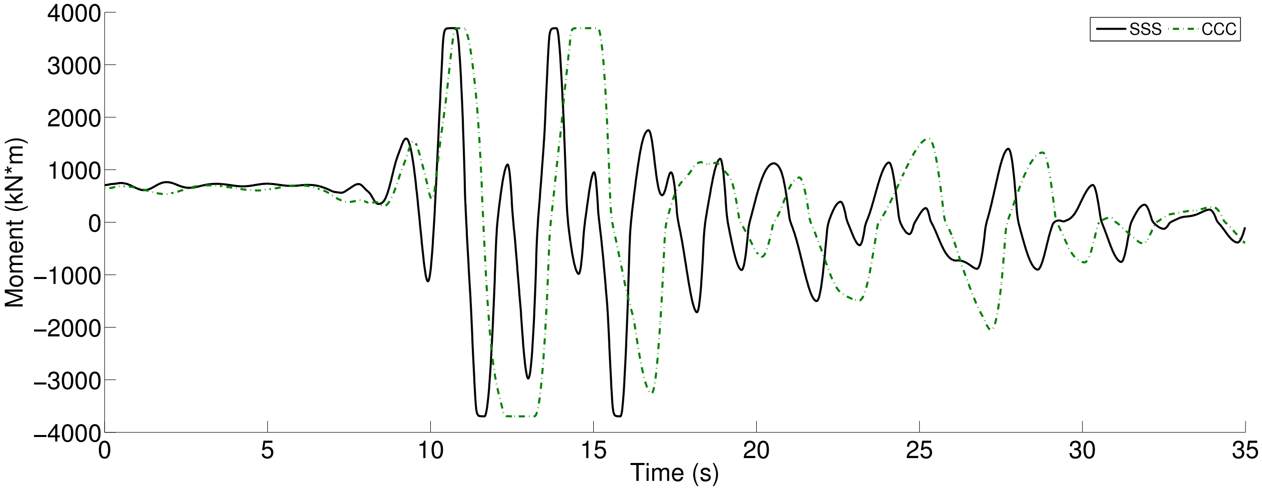


Figure 503.71: Simulated Maximum Moment Time Series, Turkey Kocaeli 1999, Yarimca, Comparison of Two Cases (Structure Bent 3 Pile 2)

Table 503.5: Simulation Scenarios for Prototype ESSI Studies

Simulation Cases	Soil Block 1	Soil Block 2	Soil Block 3
Case 1	Stiff Sand	Stiff Sand	Stiff Sand
Case 8	Soft Clay	Soft Clay	Soft Clay

503.2.4 Earthquake Soil Structure Interaction Effects

In the following sections we analyze various Earthquake Soil Structure Interaction (ESSI) effects.

503.2.4.1 How Strength of Soil Foundations Affects ESSI

It has been well known that stiffer soil layer will provide higher bearing capacity of structures so site improvements will always be preferred in engineering practice when one talks about foundation design.

While for dynamic cases, this widely-held impression will not be valid anymore. In order to see how stiffness of soil can affect the response of the whole ESSI system during earthquake shaking, two distinct scenarios listed in Table 503.5, Case 1 with all stiff soil foundations, and Case 8 with all soft soil foundations have been extracted to show the dynamic system response.

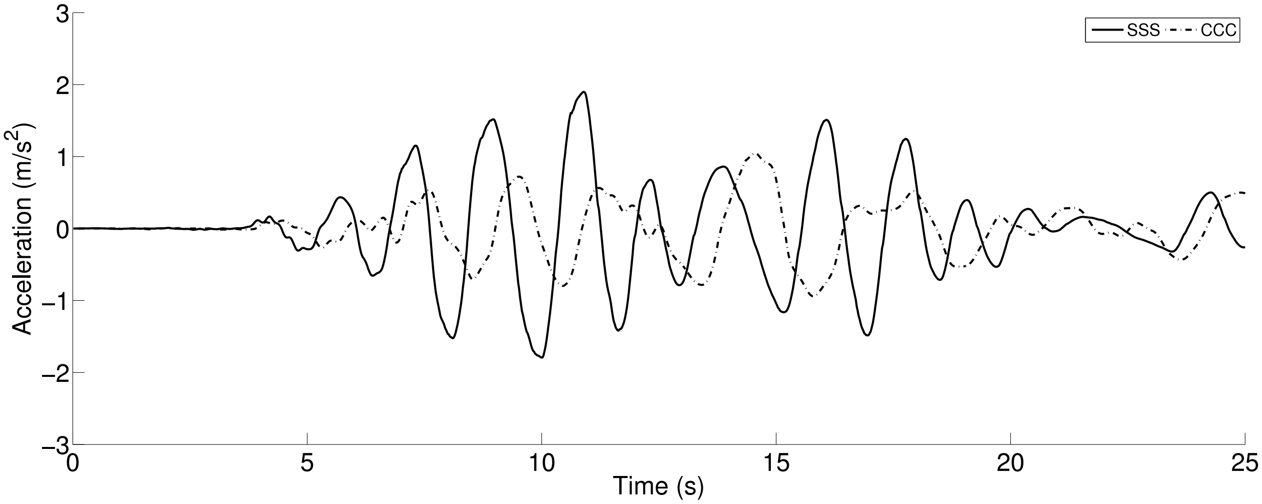


Figure 503.72: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Two Cases - First 25s (Structure Bent 1)

Case 1 and Case 8

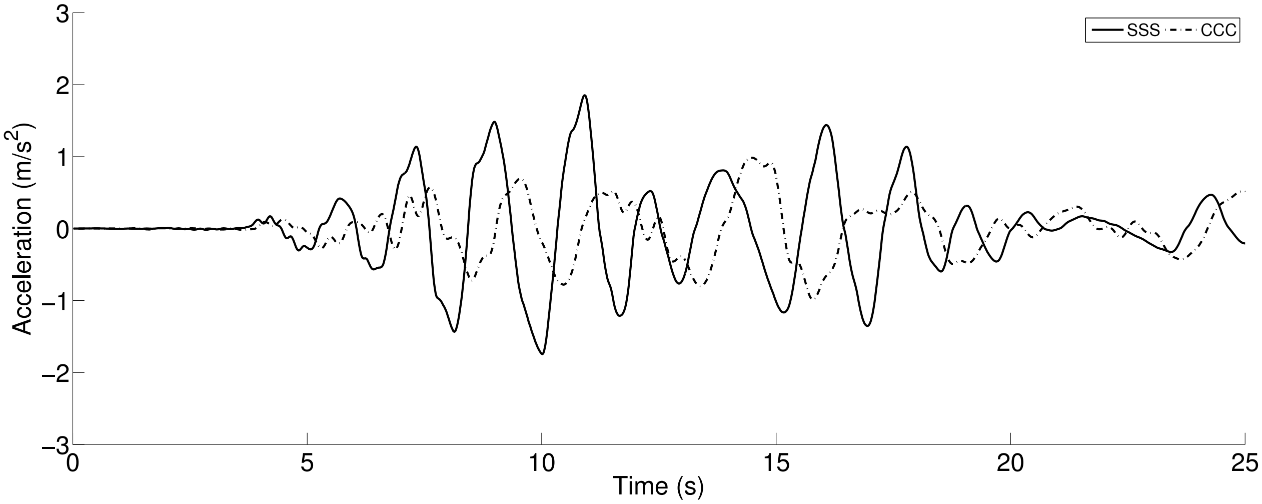


Figure 503.73: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Two Cases - First 25s (Structure Bent 2)

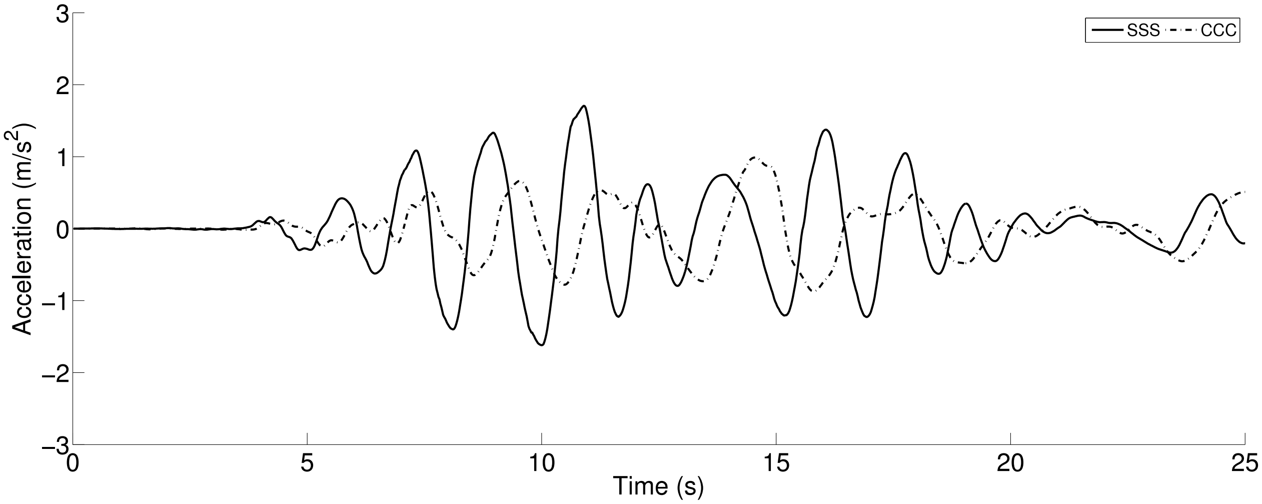


Figure 503.74: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Two Cases - First 25s (Structure Bent 3)

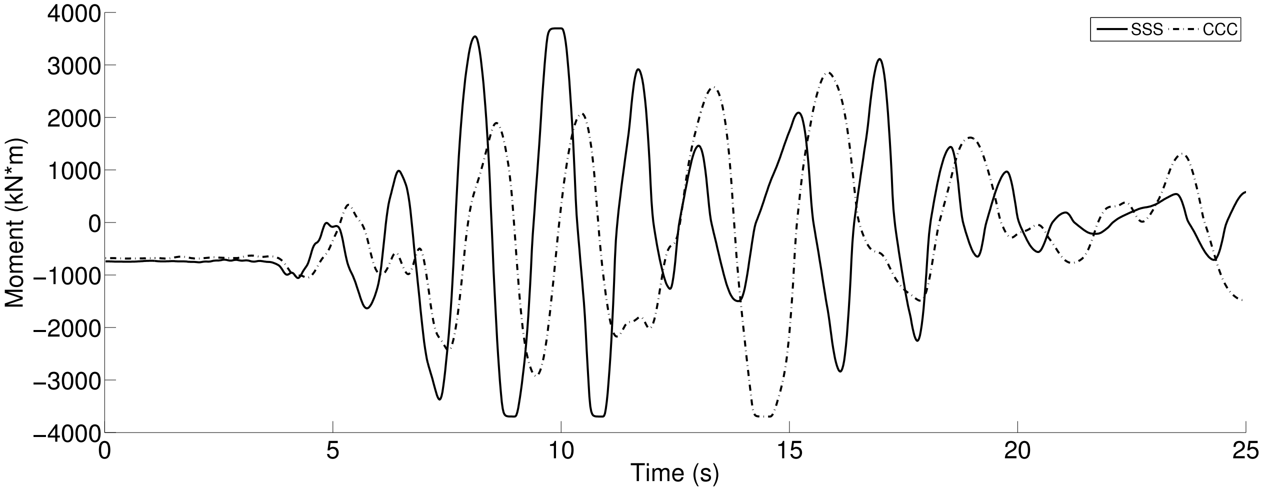


Figure 503.75: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Two Cases - First 25s (Structure Bent 1)

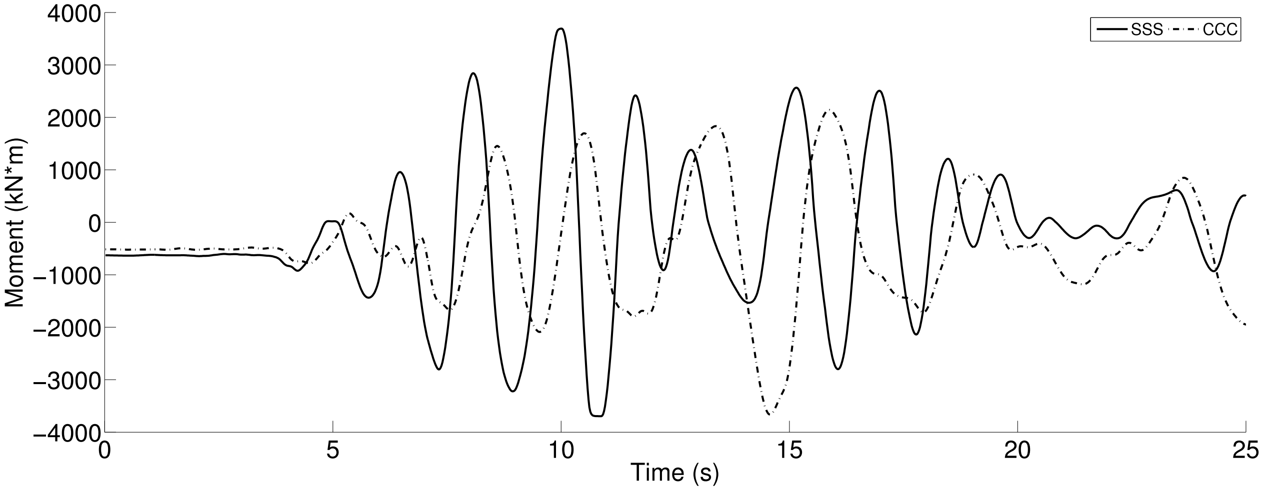


Figure 503.76: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Two Cases - First 25s (Structure Bent 2)

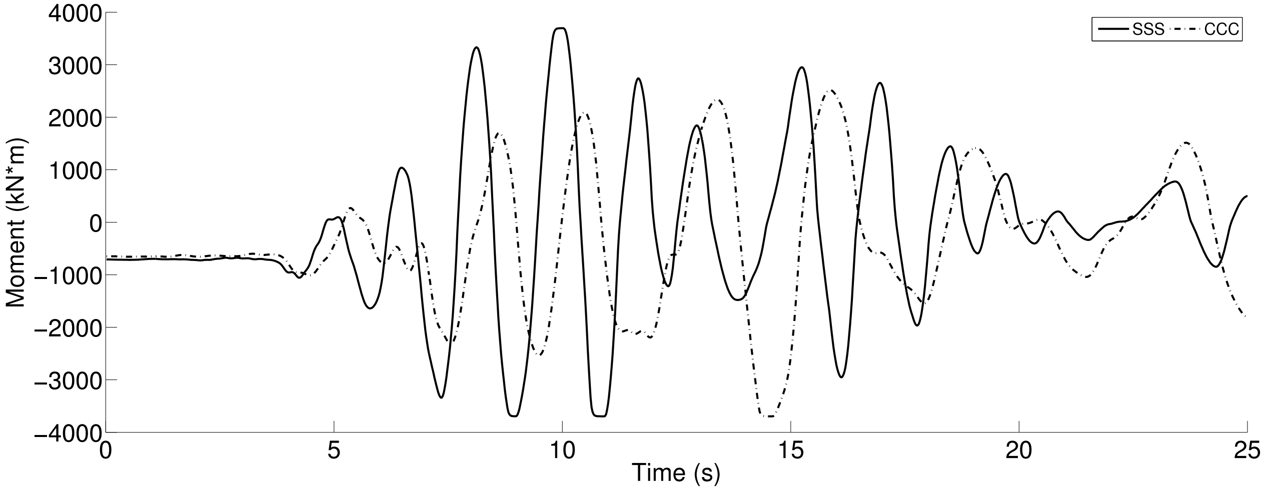


Figure 503.77: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Two Cases - First 25s (Structure Bent 3)

It has been shown in this research that if the soil layer is stiffer, structures will see much more amplification from earthquake shaking. Figures 503.72 to 503.74 show the acceleration response from ESSI systems with stiff sand and soft clay, which clearly shows that stiff sand delivers much stronger energy input to superstructures. The superstructures on top of stiffer soil also yield sooner than softer soil layers as shown in maximum moment time series Figures 503.75 to 503.77. This observation proves one interesting point that in order to improve structural stability, site improvement is not necessarily improving the dynamic resistance of the ESSI system. In later section of this work, further observation will be made to correlate this conclusion with the characteristics of the input motion.

As for the soil side, near-structure soil motion will be also affected by ESSI. So the traditional way of assuming that recorded ground motion can be used as input motion to ESSI analysis should be revisited. According to Figures 503.78 503.79 503.80, we can see that stiff soil also shows stronger surface motion records during earthquake shaking.

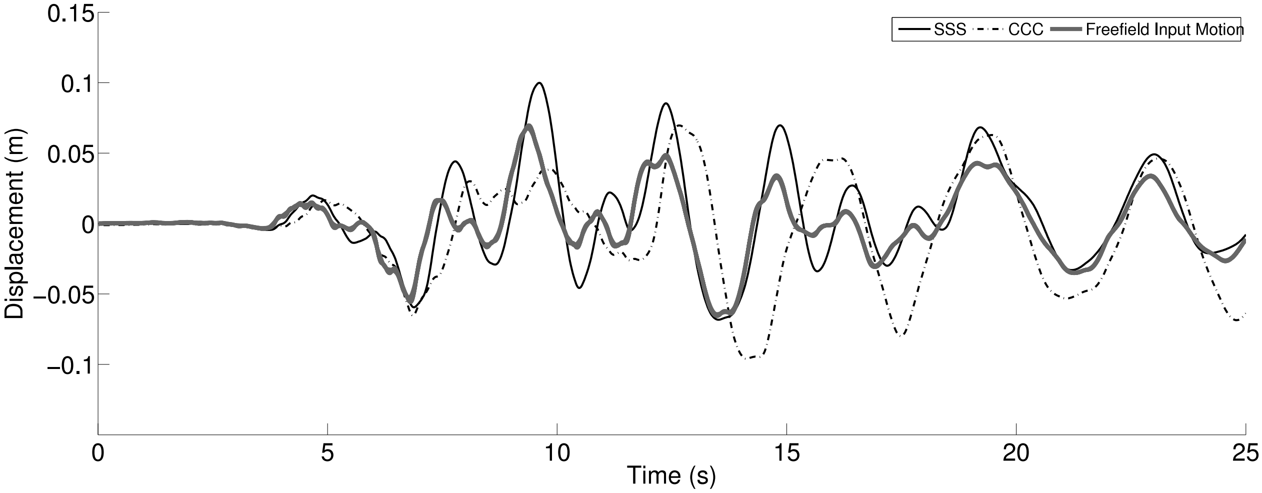


Figure 503.78: Simulated Displacement Time Series, Northridge 1994, Century City, Comparison of Two Cases - First 25s (Soil Block 1)

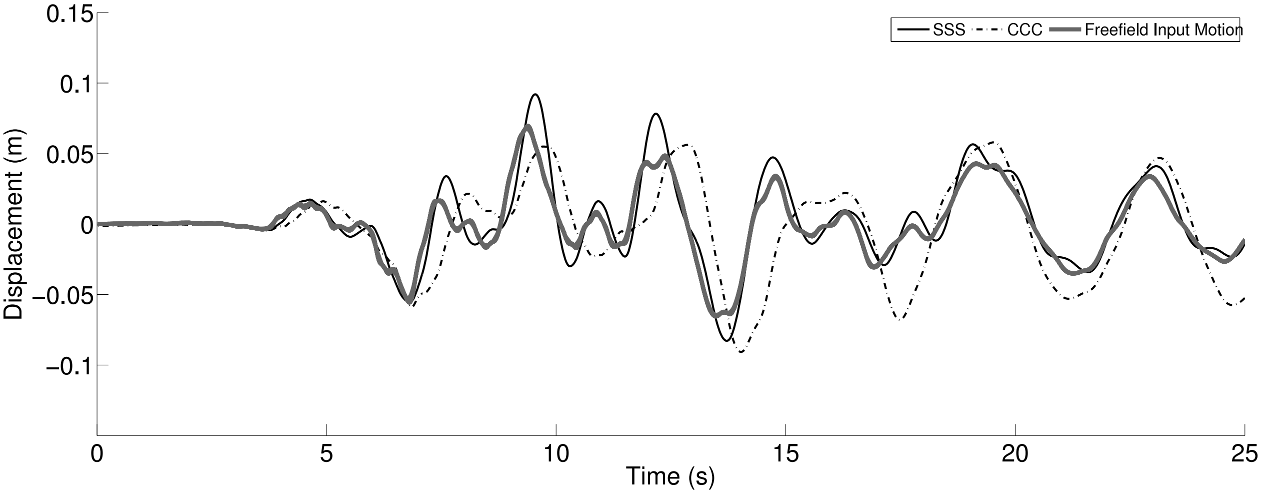


Figure 503.79: Simulated Displacement Time Series, Northridge 1994, Century City, Comparison of Two Cases - First 25s (Soil Block 2)

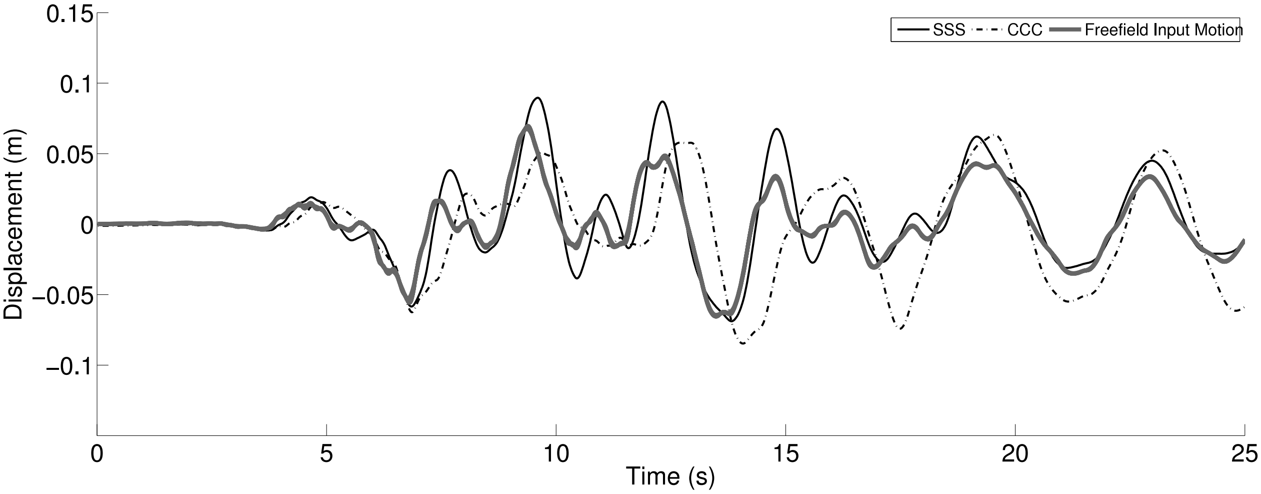


Figure 503.80: Simulated Displacement Time Series, Northridge 1994, Century City, Comparison of Two Cases - First 25s (Soil Block 3)

Table 503.6: Simulation Scenarios for Prototype ESSI Studies

Simulation Cases	Soil Block 1	Soil Block 2	Soil Block 3
Case 3	Stiff Sand	Soft Clay	Stiff Sand

503.2.4.2 How Site Non-Uniformity Affects ESSI

Bridge is always constructed over gulfs or bays. It is a common case that for multi-span bridge, different bents will inevitably sit on soil foundations with totally different strength. This site nonuniformity complicates design because individual structure response might be largely varied.

In order to study the effects of site non-uniformity on the dynamic response of ESSI system, Case 3 listed in Table 503.6 has been selected as the test bed of our simulation. This scenario corresponds to the case that a 4-span bridge sits on solid abutments but with much softer bay-mud type foundation in the middle of a bay.

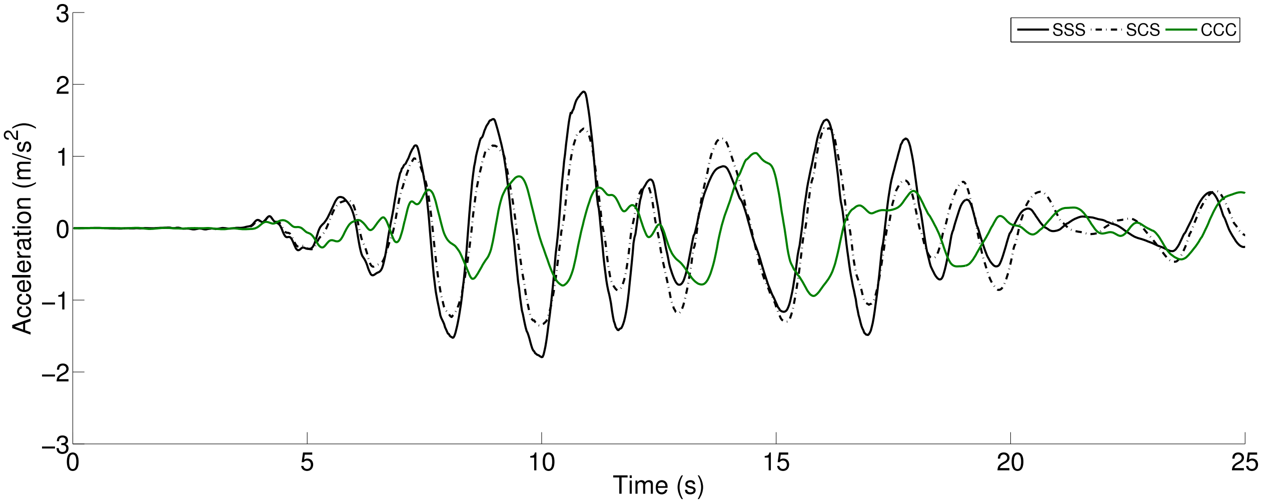


Figure 503.81: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 1)

Case 3

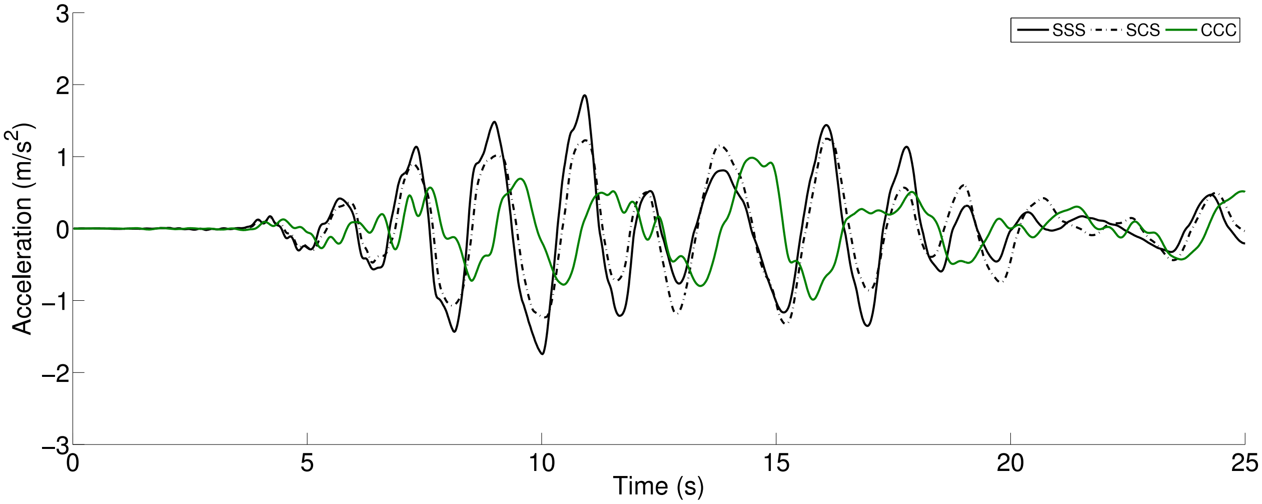


Figure 503.82: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 2)

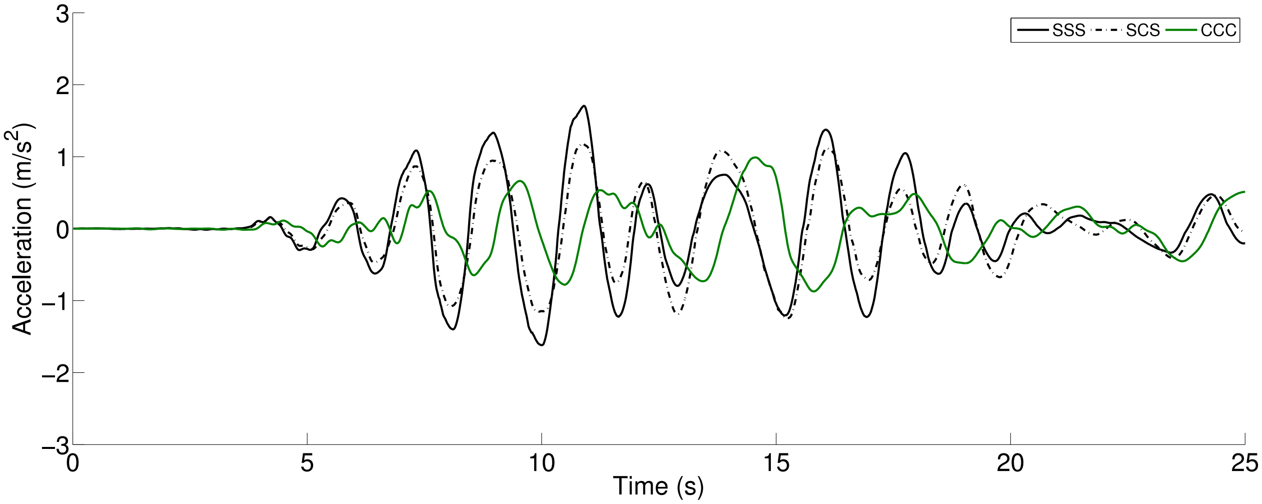


Figure 503.83: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 3)

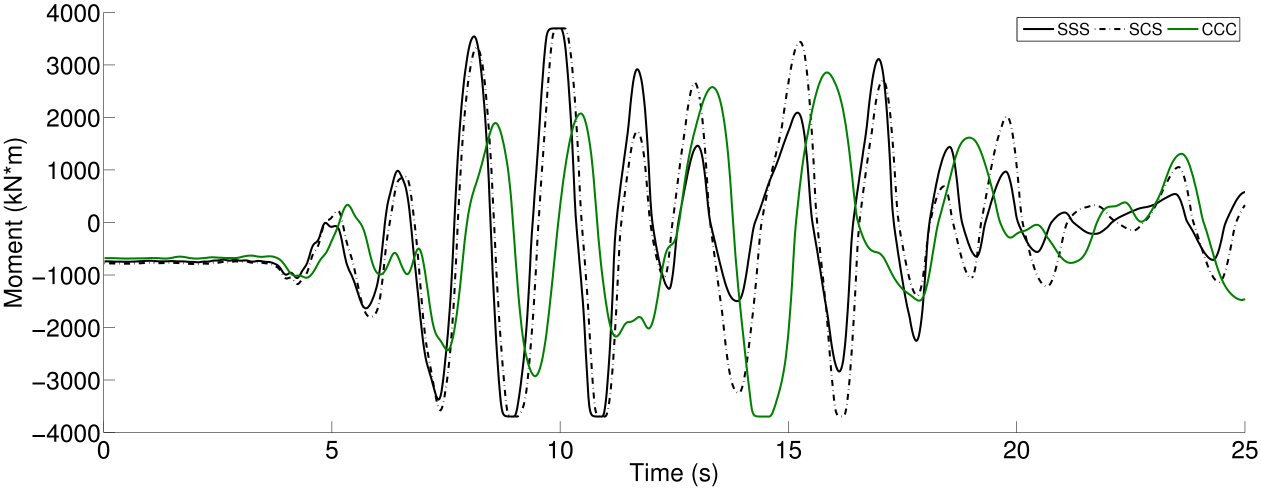


Figure 503.84: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 1)

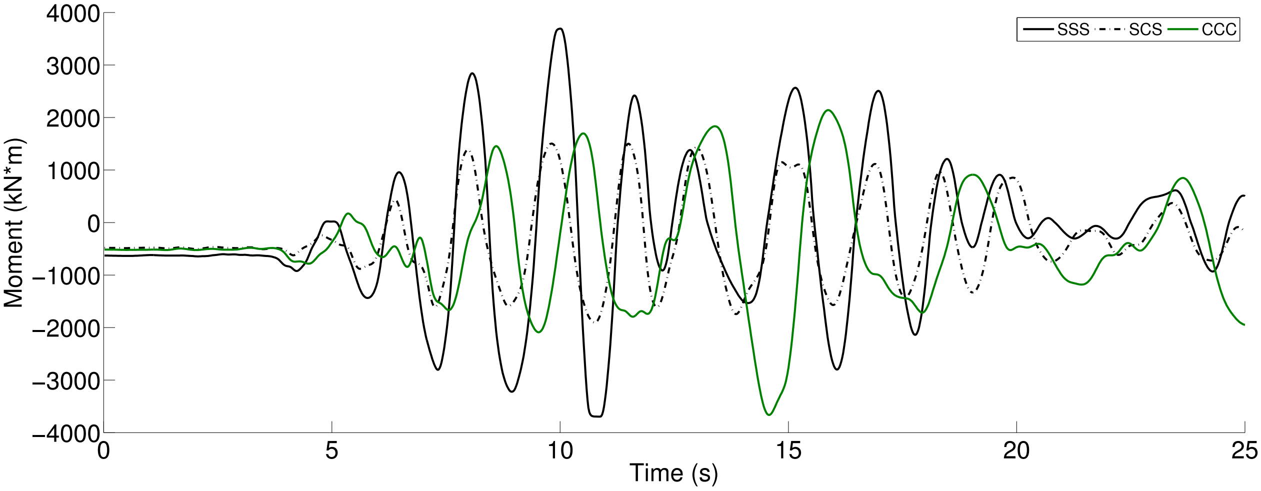


Figure 503.85: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 2)

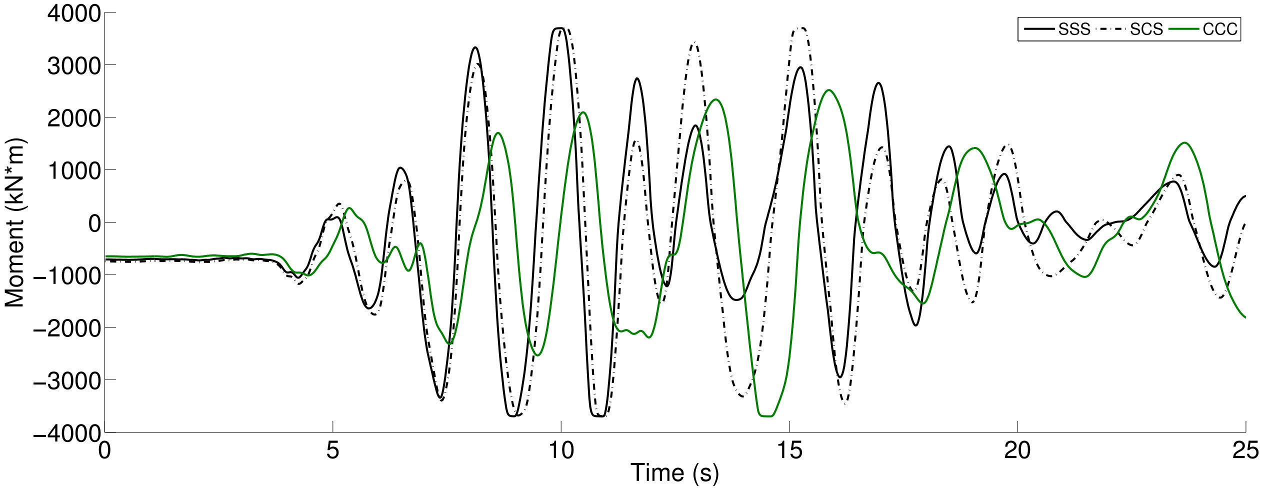


Figure 503.86: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 3)

Table 503.7: Simulation Scenarios for Prototype ESSI Studies

Simulation Cases	Soil Block 1	Soil Block 2	Soil Block 3
Case 2	Stiff Sand	Stiff Sand	Soft Clay
Case 5	Soft Clay	Stiff Sand	Stiff Sand

From acceleration Figures 503.81, 503.82, 503.83, and moment Figures 503.84, 503.85 and 503.86, we can clearly see that due to the presence of a soft soil block, all structures show smaller magnitude in response. This is especially true for the superstructure that directly sits on top of the softer soil foundation. For Case 3 listed in Table 503.6, the superstructure (bent 2) gets much smaller response because it is right on top of clay (bay mud) foundation. As a matter of fact, the middle bent (bent 2) does not yield at all.

We want to extend this observation to other cases as listed in Table 503.7. Case 2 shows the scenario that the soil foundation supporting bent 3 is soft bay mud, while Case 5 shows it for bent 1.

The same reasoning can be applied to these similar cases. Figures 503.87 to 503.92 show the results of Case 2 and Figures 503.93 to 503.98 for Case 5.

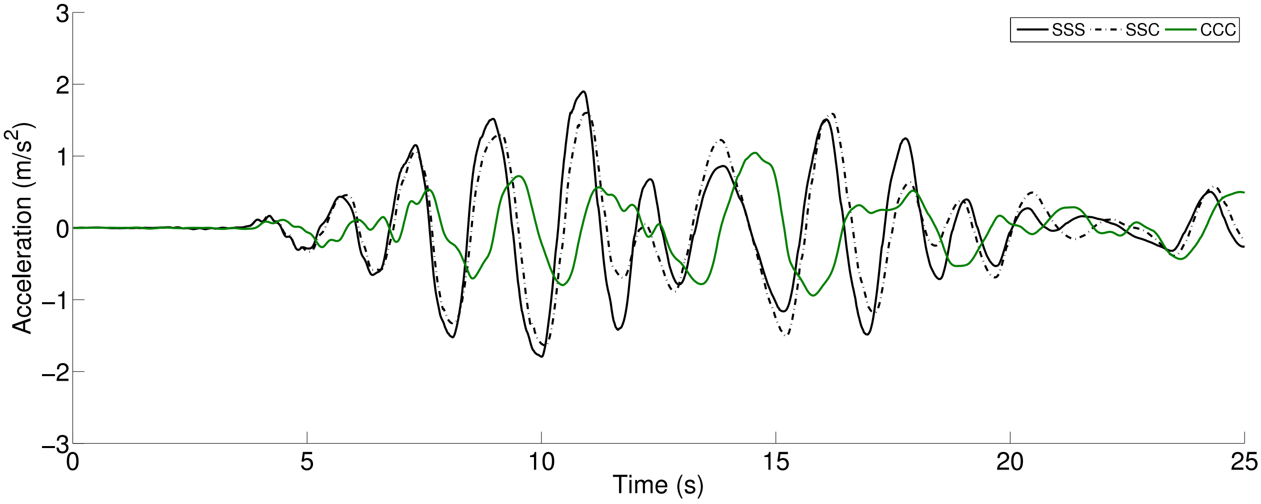


Figure 503.87: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 1)

Case 2

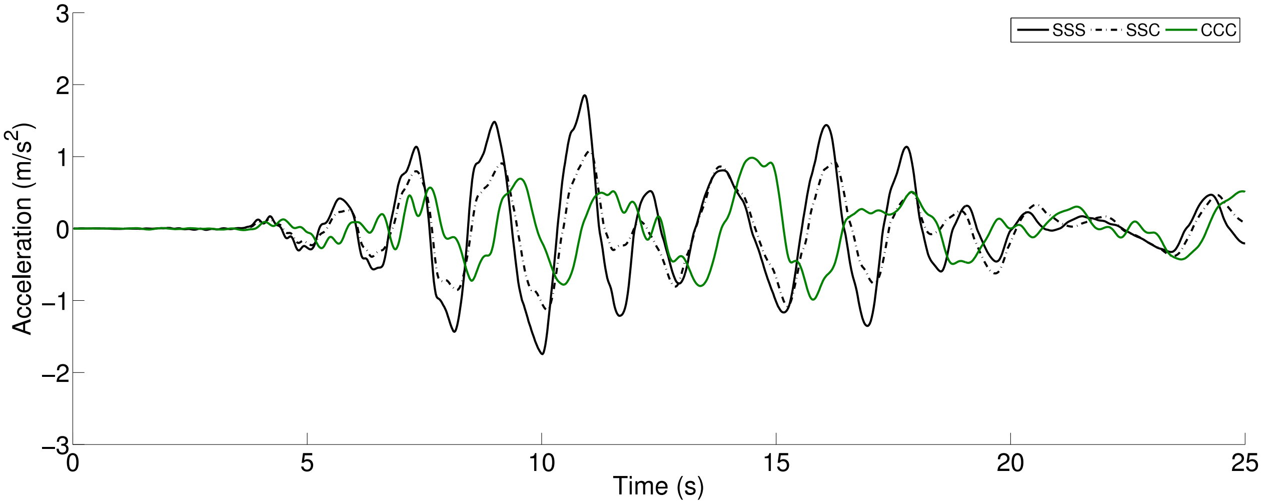


Figure 503.88: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 2)

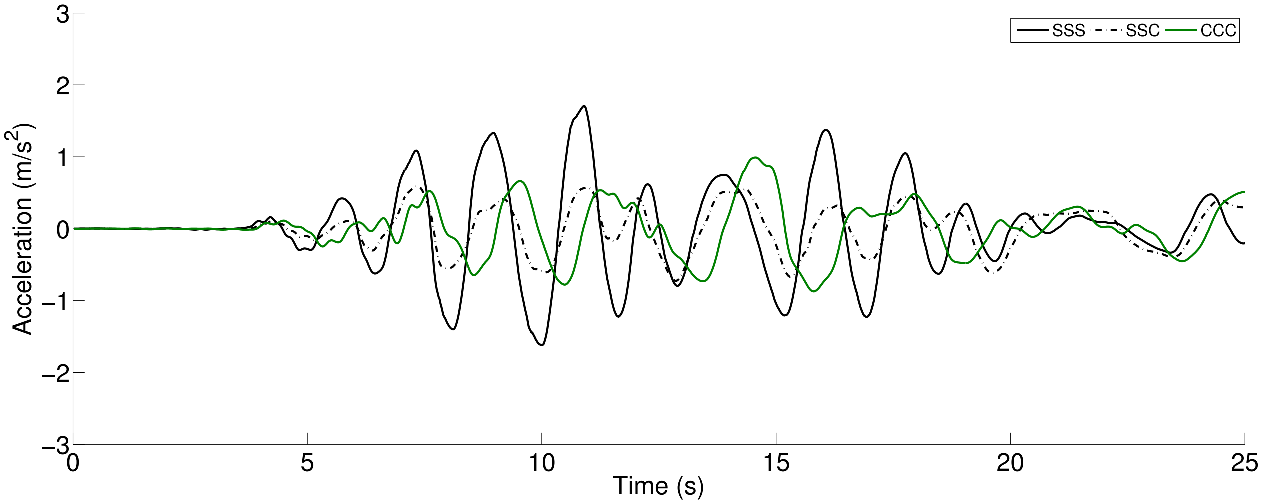


Figure 503.89: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 3)

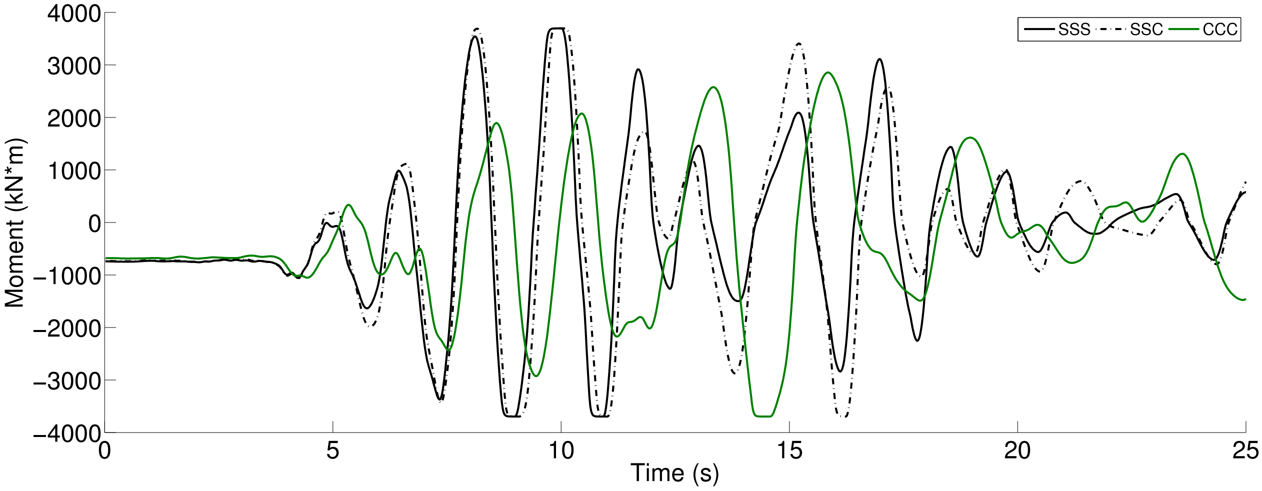


Figure 503.90: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 1)

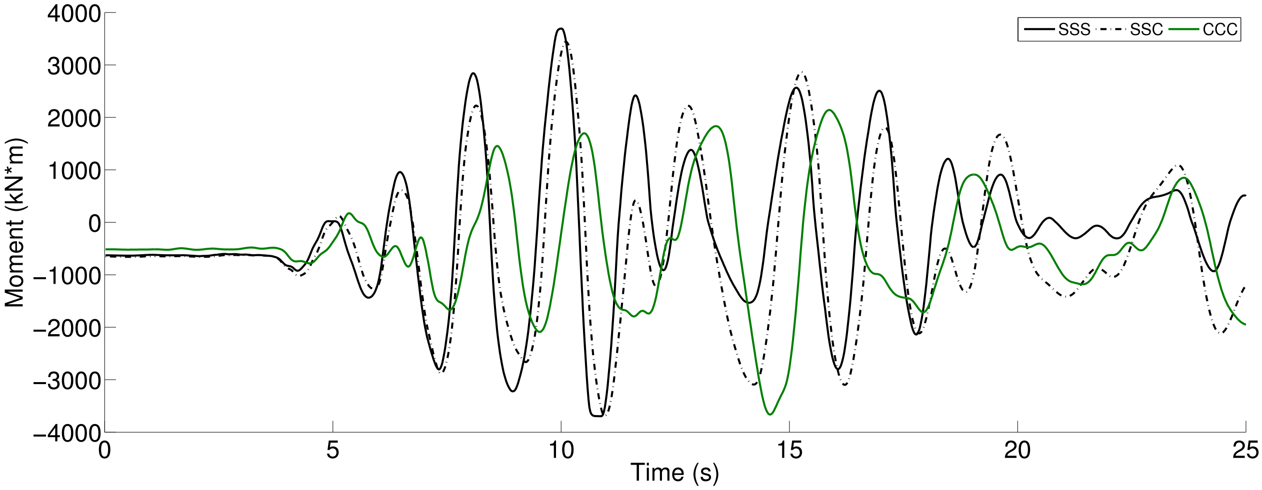


Figure 503.91: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 2)

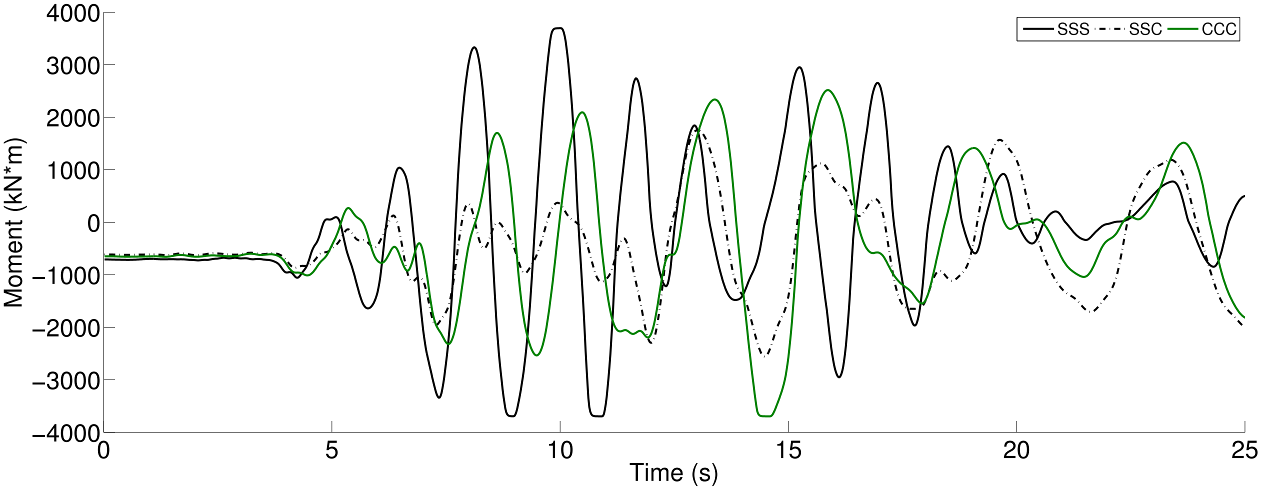


Figure 503.92: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 3)

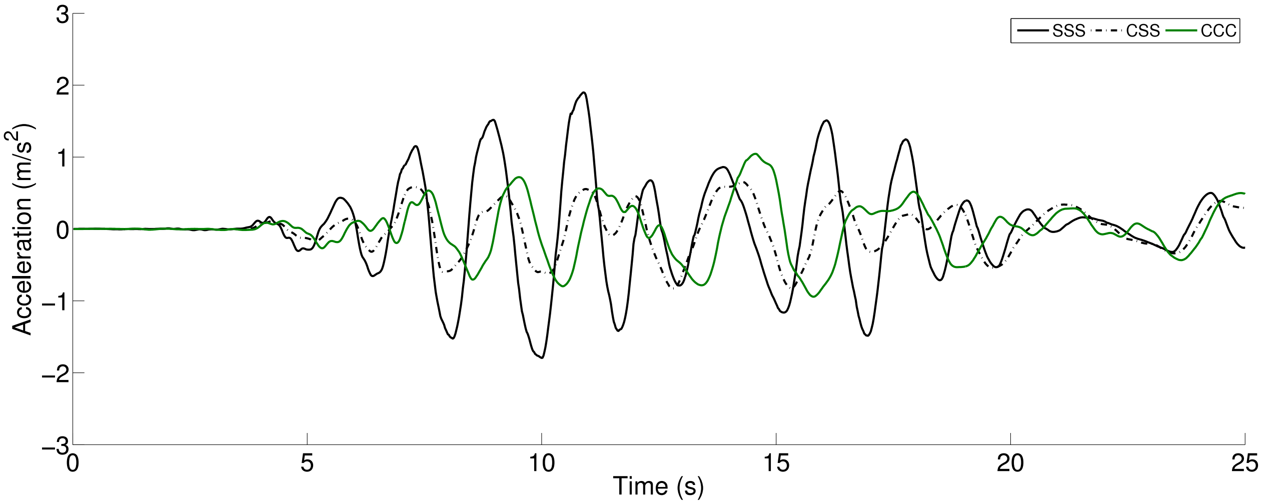


Figure 503.93: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 1)

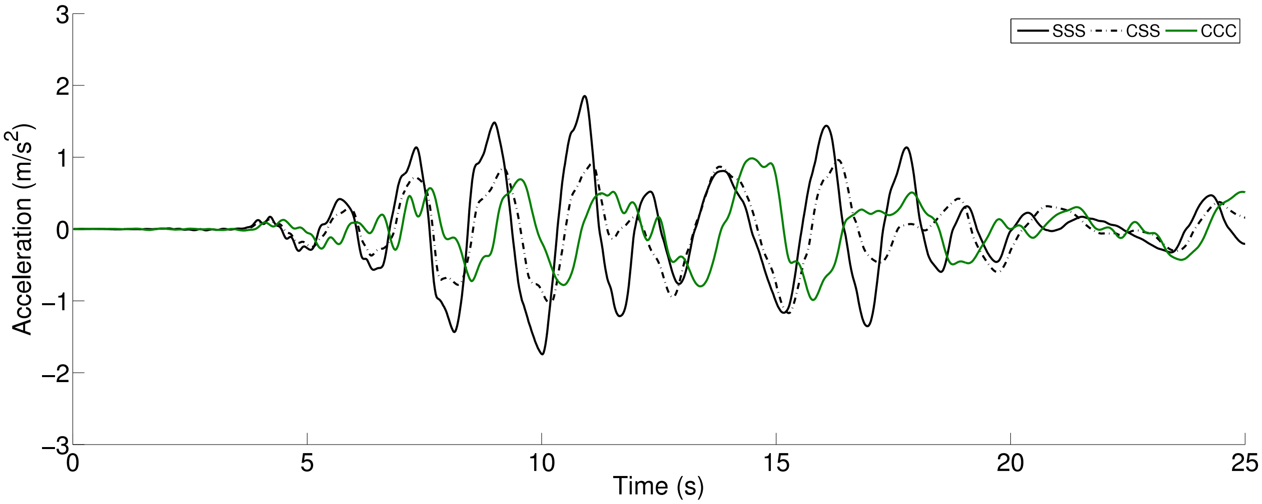


Figure 503.94: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 2)

Case 5

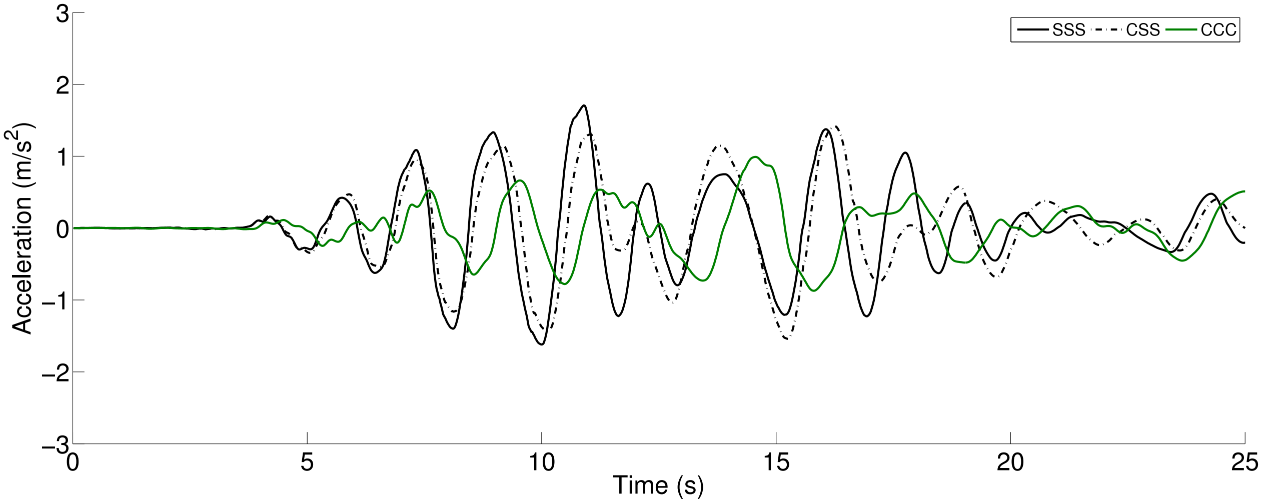


Figure 503.95: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 3)

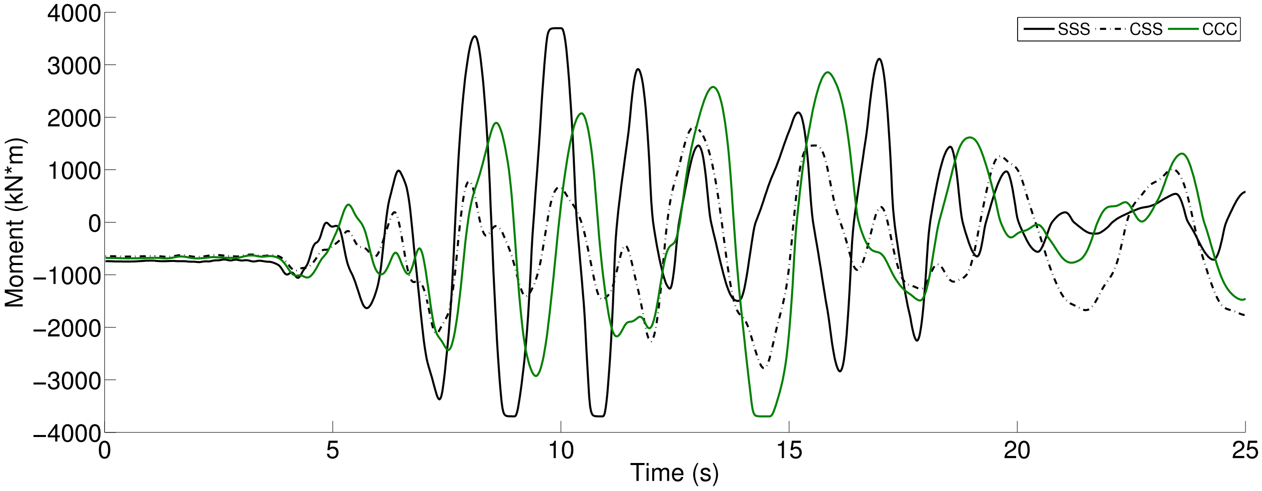


Figure 503.96: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 1)

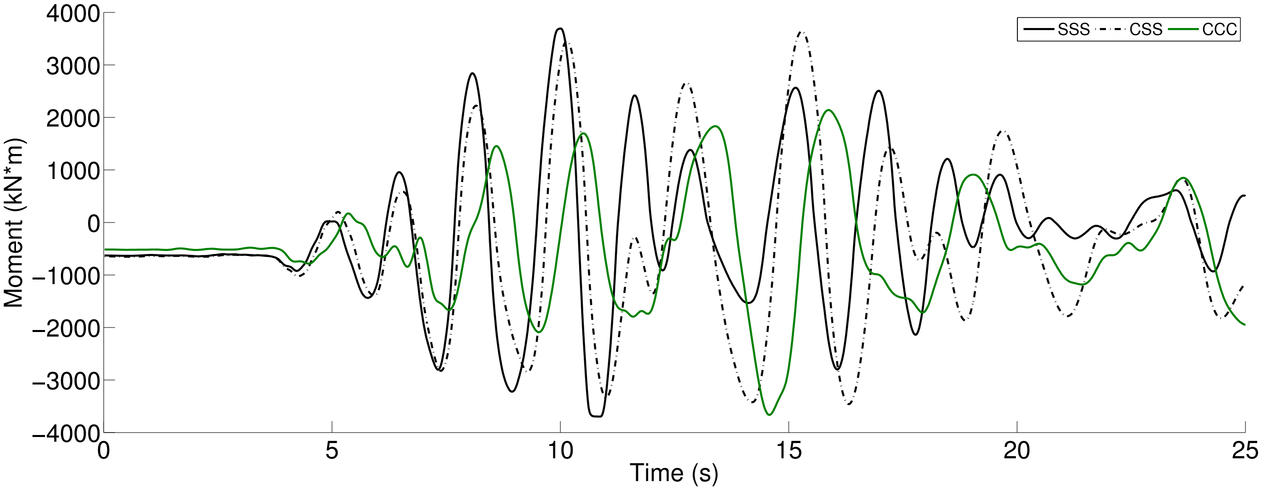


Figure 503.97: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 2)

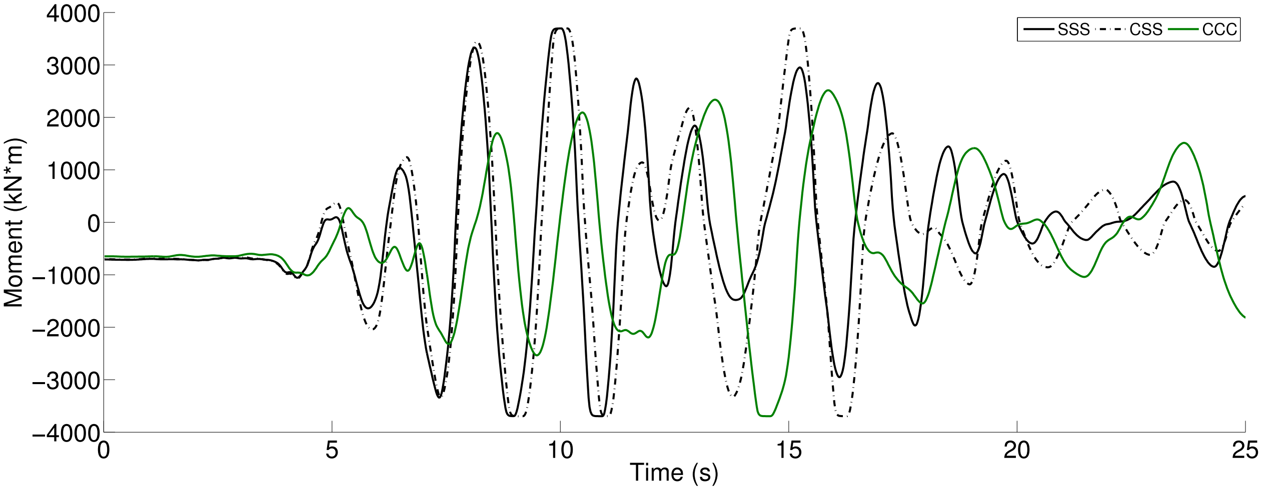


Figure 503.98: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 3)

Table 503.8: Simulation Scenarios for Prototype ESSI Studies

Simulation Cases	Soil Block 1	Soil Block 2	Soil Block 3
Case 4	Stiff Sand	Soft Clay	Soft Clay
Case 6	Soft Clay	Stiff Sand	Soft Clay
Case 7	Soft Clay	Soft Clay	Stiff Sand

As a final conclusion, if there exists one substantially weaker soil layer for the ESSI system, the dynamic response for the whole ESSI system will be attenuated. The structure that directly sits on top of the soft soil block will receive significantly smaller excitation during earthquake shakings.

The other side of story can be also formulated using results obtained in this work. What if there is a substantially stronger soil foundation in the ESSI system? It would be worthwhile exploring the other side of reasoning. In this work, three other cases have been used to observe how the ESSI system will behave for the cases that one soil block is much stronger that the other blocks as listed in Table 503.8.

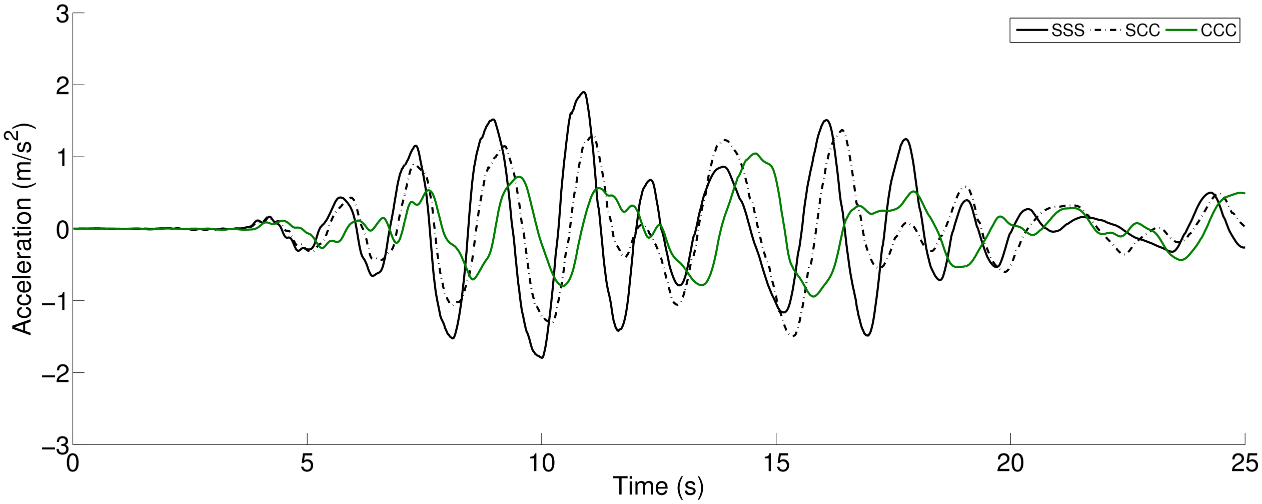


Figure 503.99: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 1)

Case 4

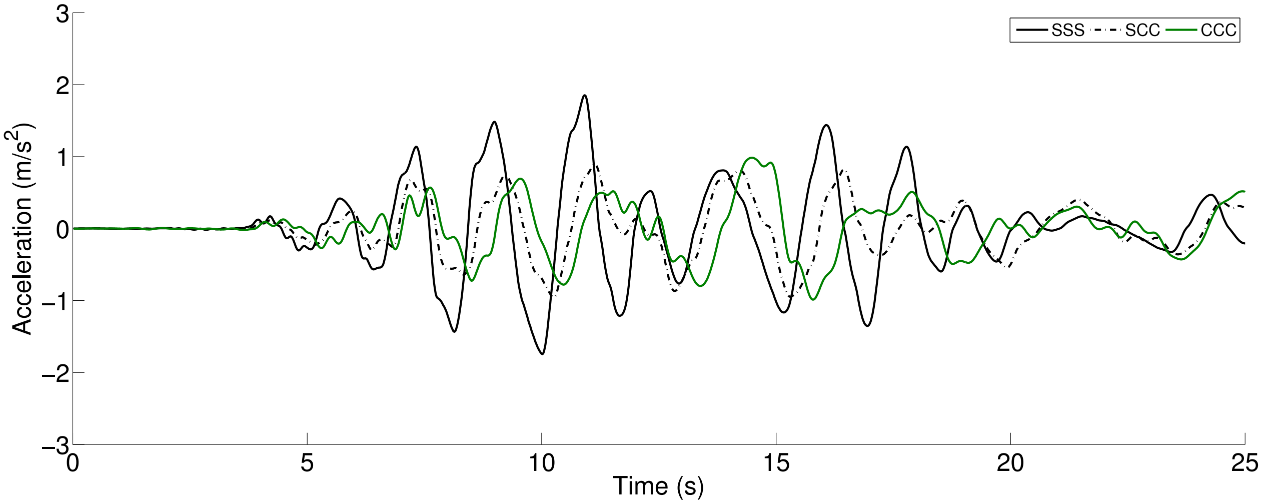


Figure 503.100: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 2)

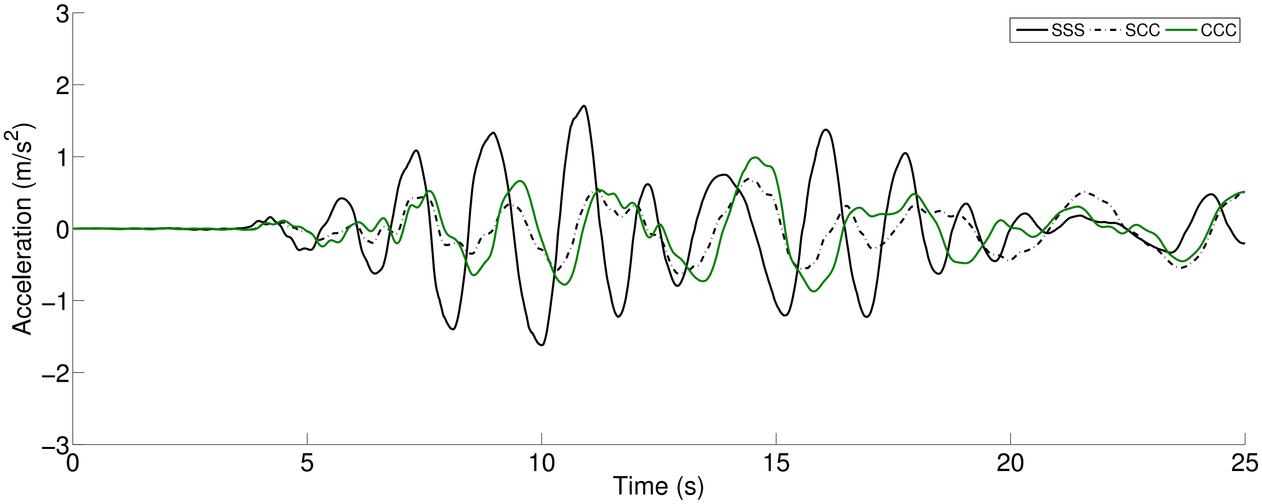


Figure 503.101: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 3)

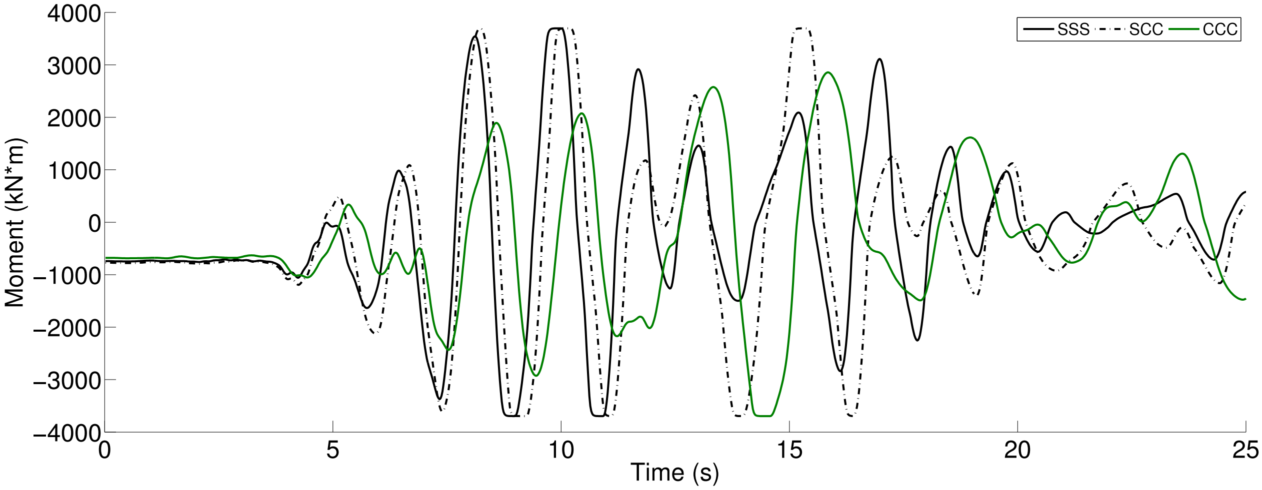


Figure 503.102: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 1)

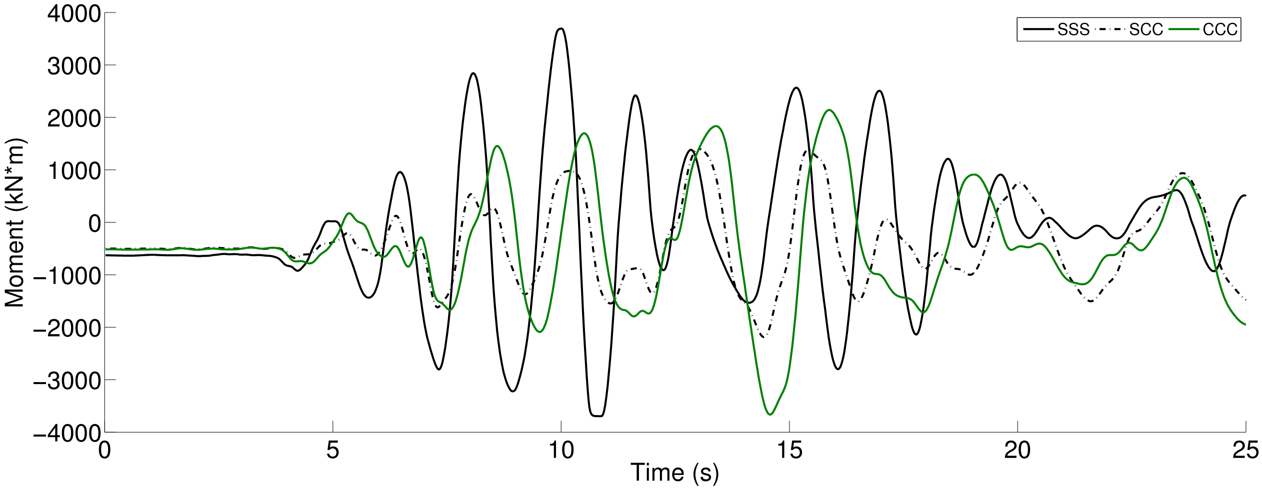


Figure 503.103: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 2)

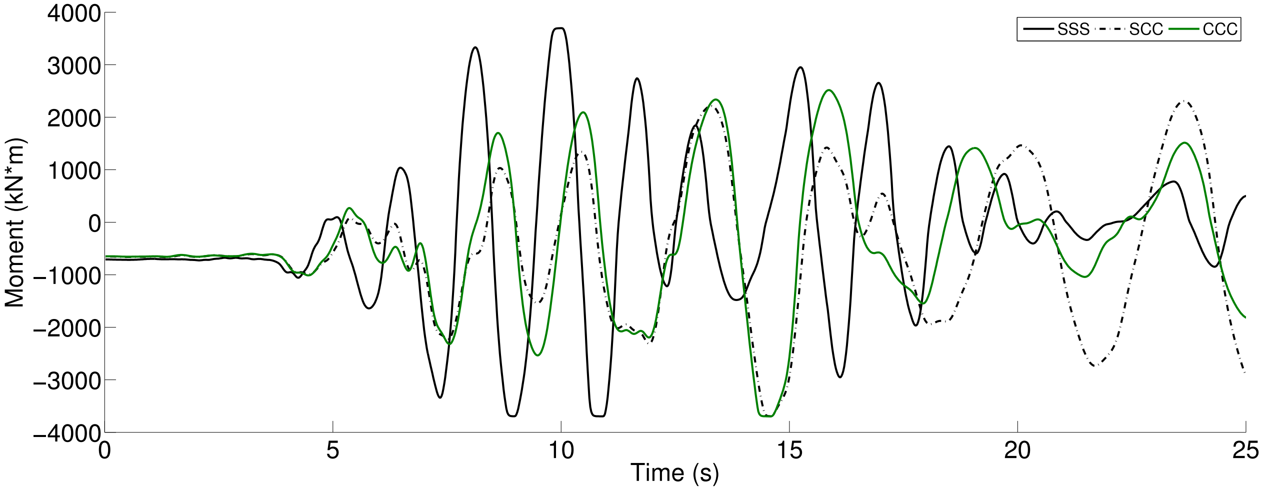


Figure 503.104: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 3)

It can easily be seen from Figures 503.99 to 503.104 that the structure on top of the strong soil block will exhibit much bigger response and while the other structures sitting on weaker soil blocks will not get much shaking at all.

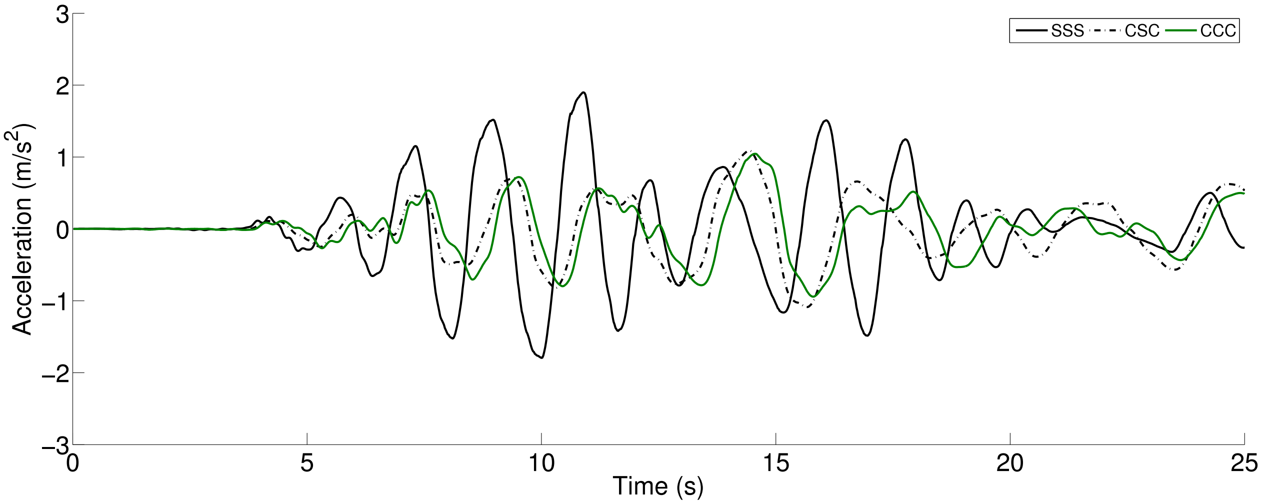


Figure 503.105: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 1)

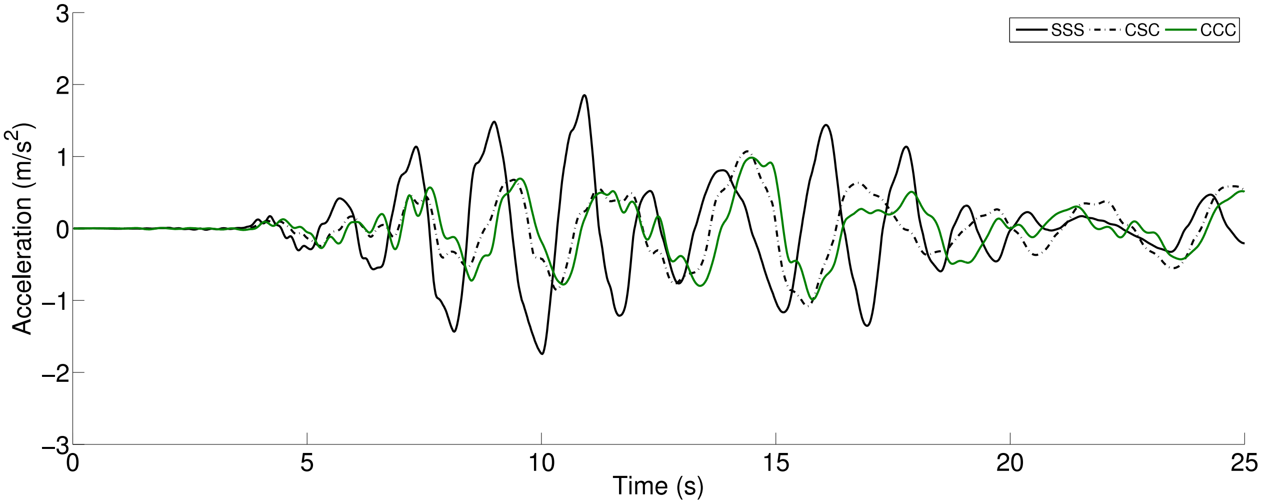


Figure 503.106: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 2)

Case 6

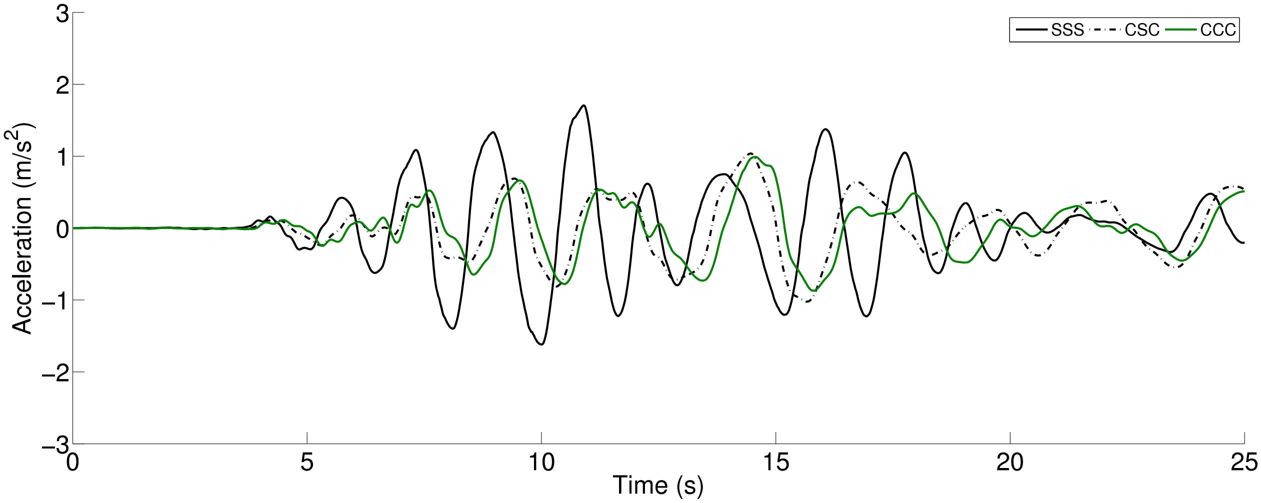


Figure 503.107: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 3)

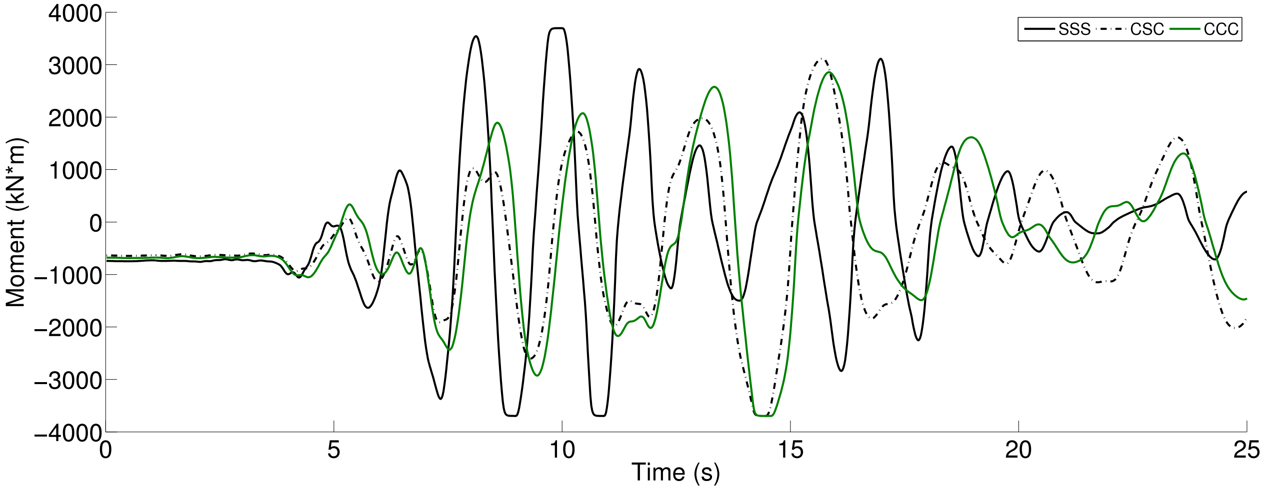


Figure 503.108: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 1)

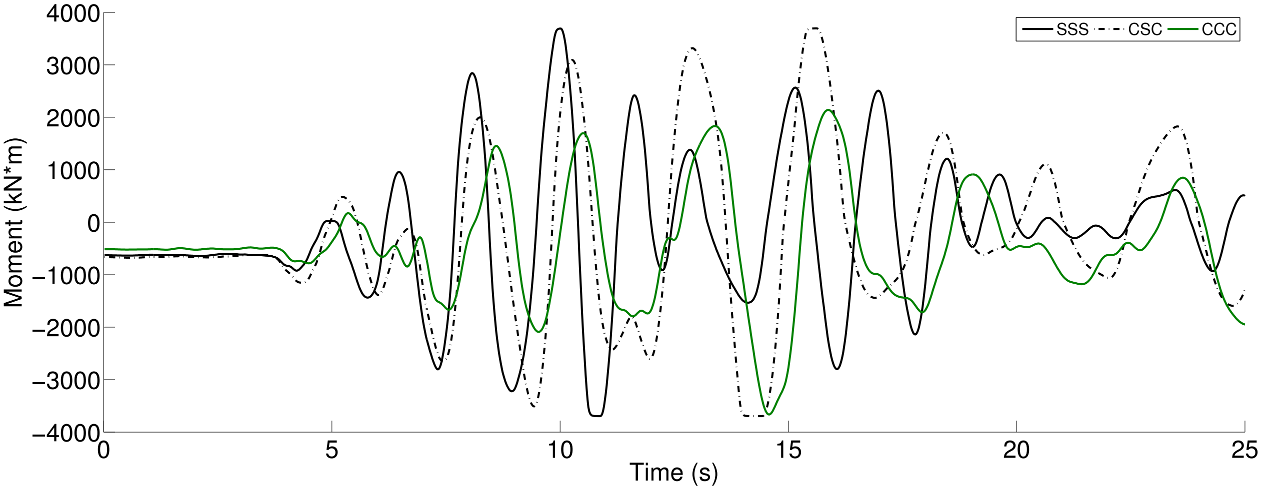


Figure 503.109: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 2)

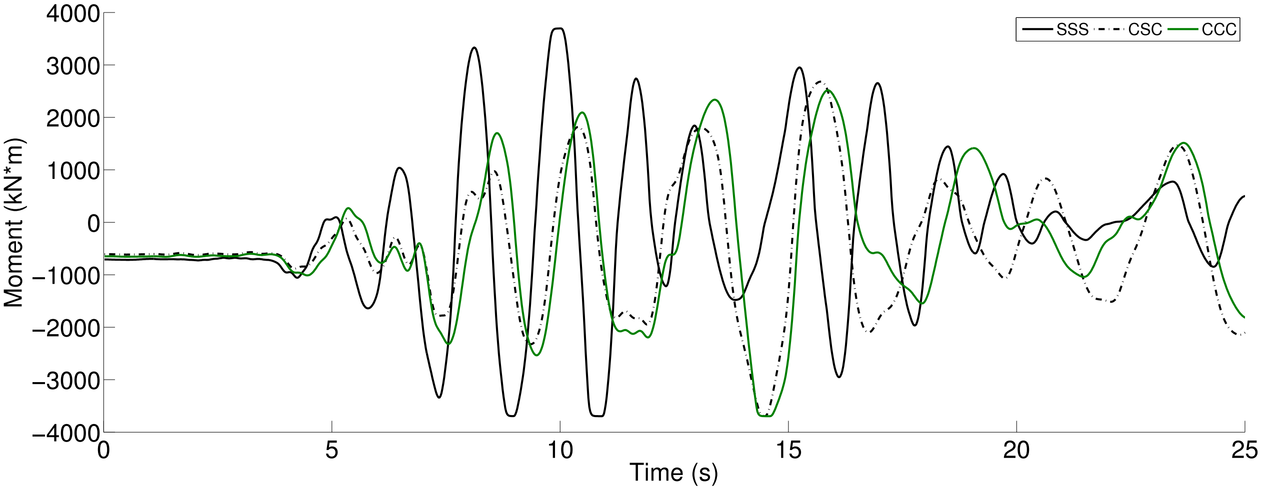


Figure 503.110: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 3)

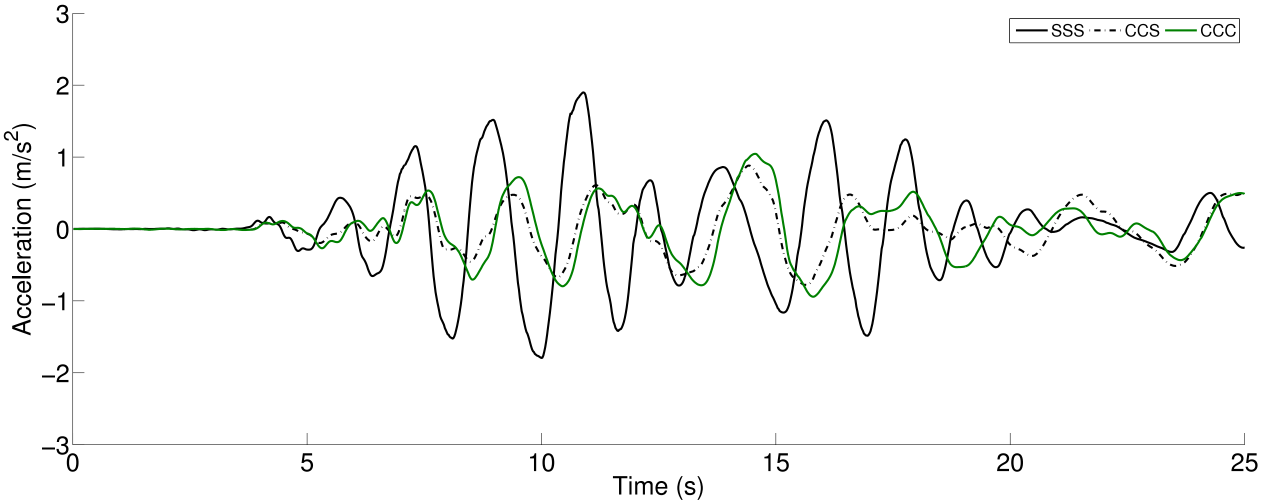


Figure 503.111: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 1)

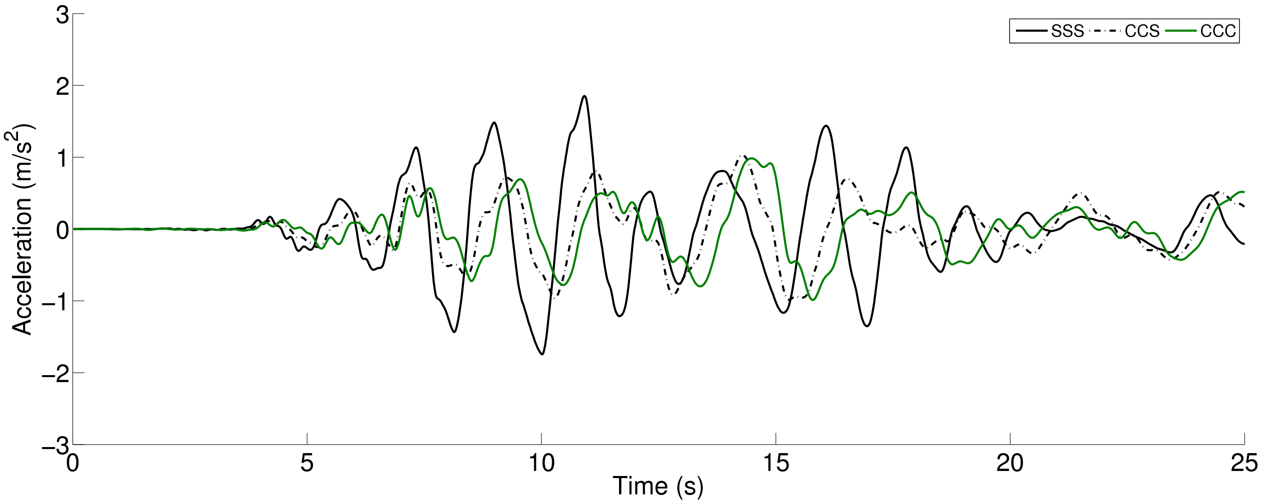


Figure 503.112: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 2)

Case 7

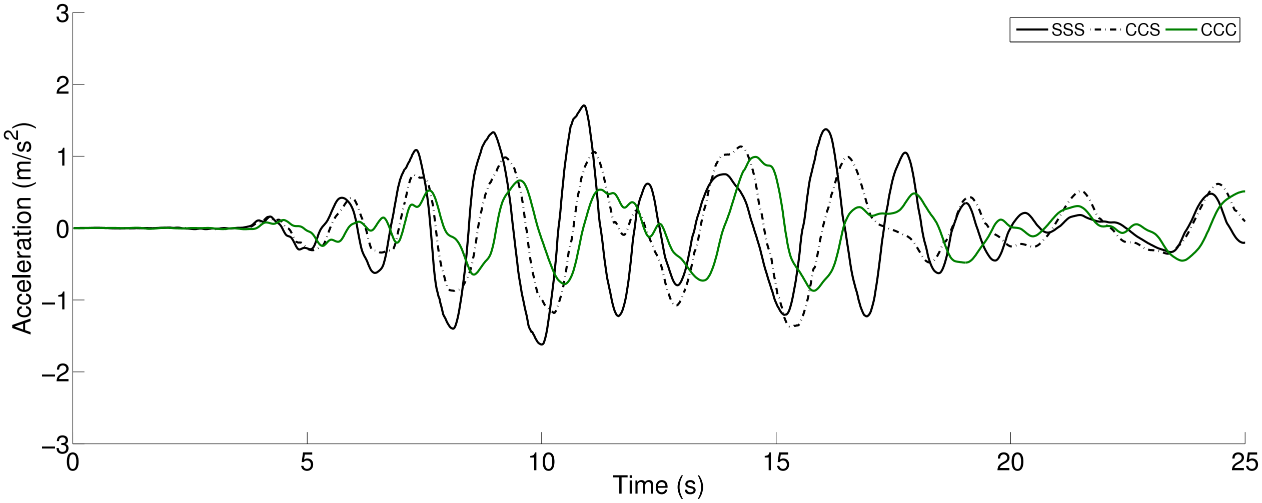


Figure 503.113: Simulated Acceleration Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 3)

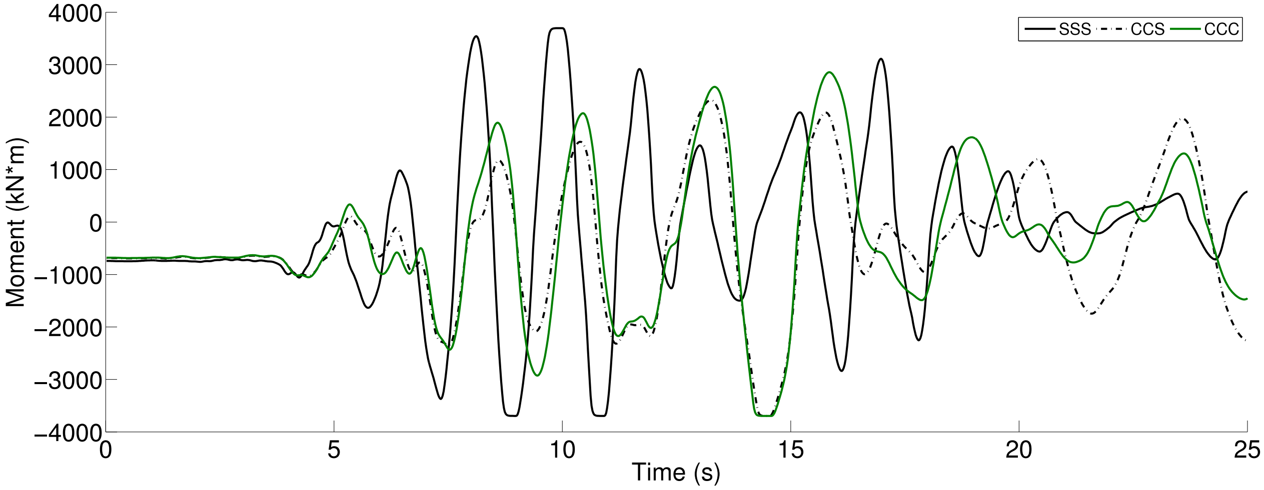


Figure 503.114: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 1)

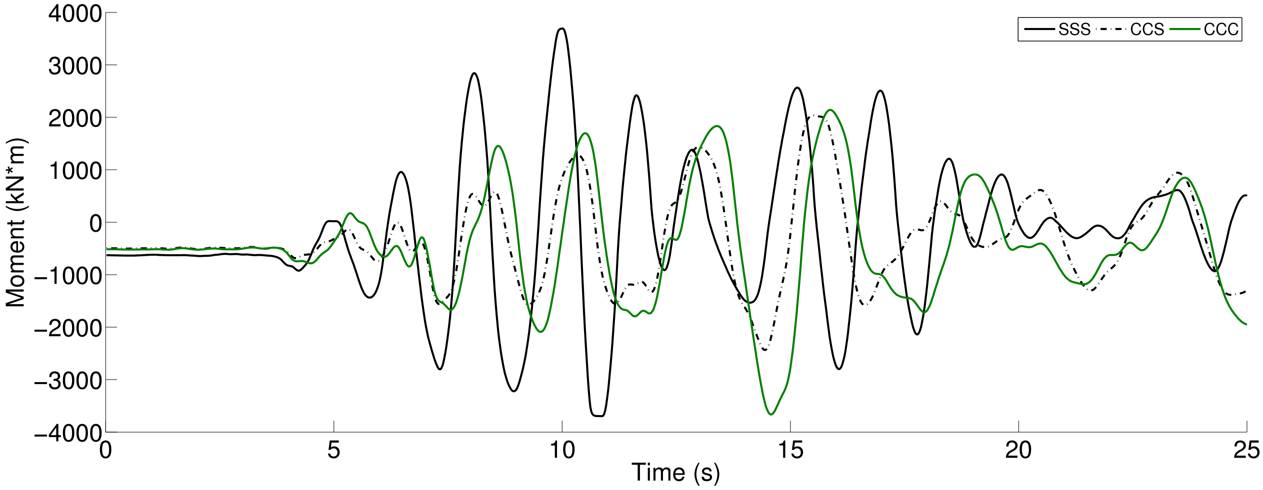


Figure 503.115: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 2)

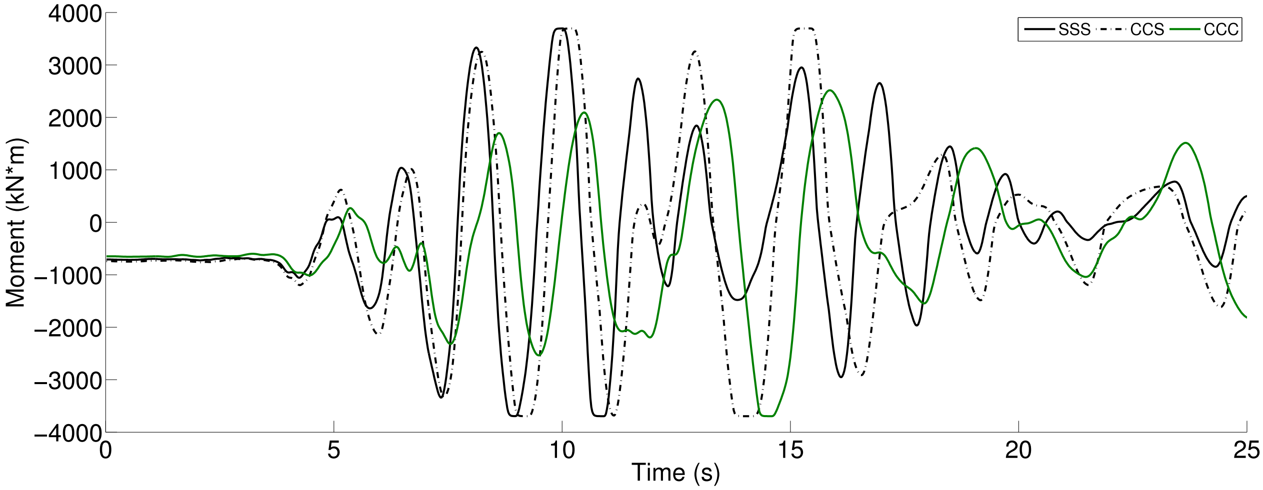


Figure 503.116: Simulated Maximum Moment Time Series, Northridge 1994, Century City, Comparison of Three Cases - First 25s (Structure Bent 3)

As a conclusion, dynamic behaviors of the ESSI system on non-uniform sites can be much more complicated than the uniform case. Generally speaking, the existence of soft soil foundation will attenuate the dynamic response of the whole ESSI system. The structure supported by soft soil block will see far less shakings than the ones supported by stiff soil. From the design perspective, the structure on top of strong soil foundations should be designed to higher safety with larger margin.

503.2.4.3 How Input Motion Affects ESSI

All the results have been shown in previous section, the purpose is to investigate how the whole ESSI system will respond to excitations with different predominant frequency.

From Figure 503.117 and 503.118, we can clearly see the difference in frequency contents of the two ground motions studied. THe 1999 Turkey Kocaeli earthquake contains more long period components that are not present at the 1994 Northridge site.

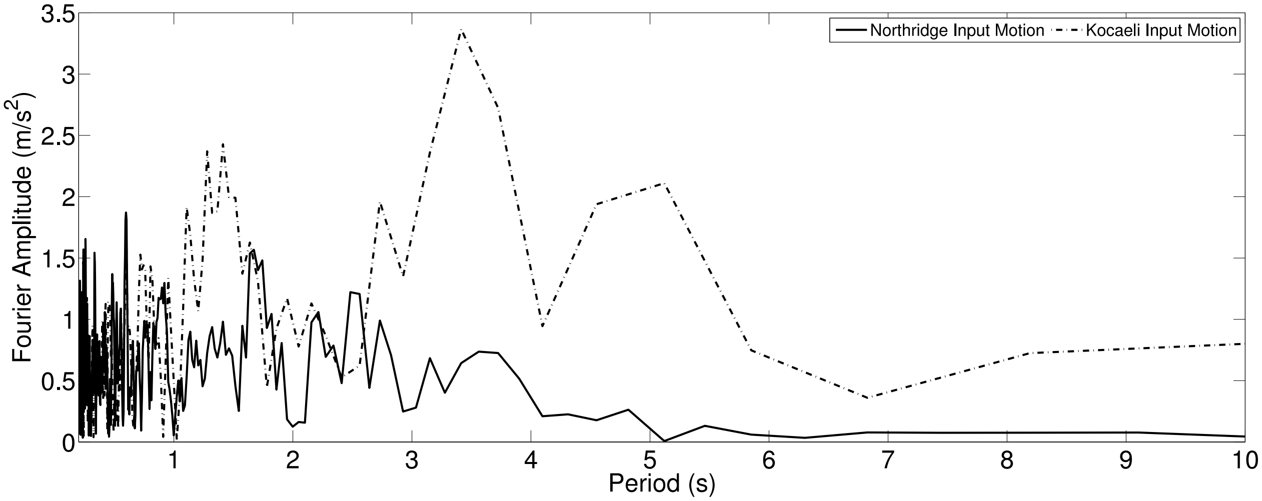


Figure 503.117: Frequency Contents of Ground Motions - Acceleration Time Series

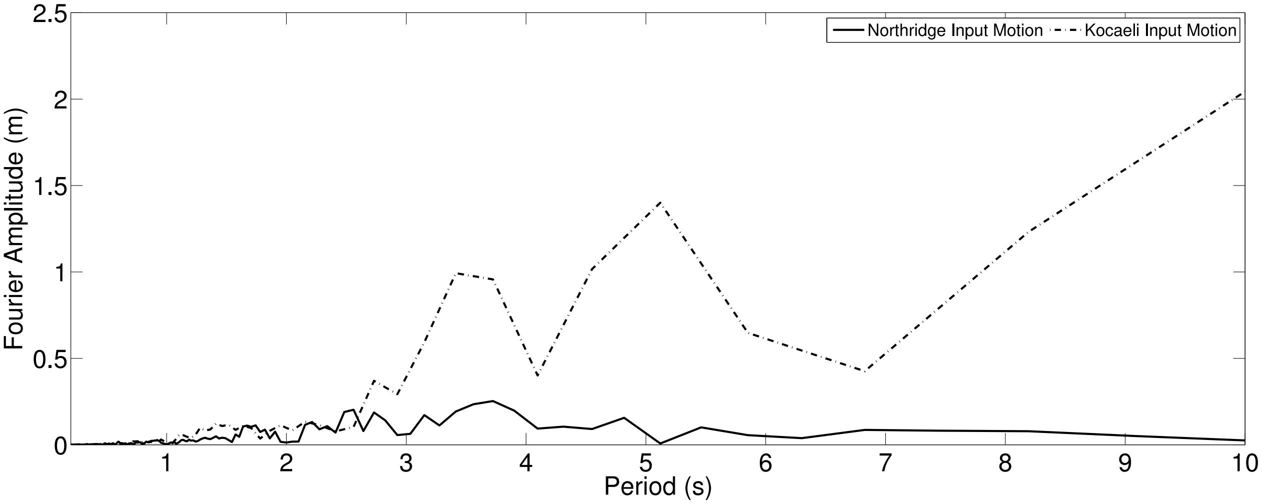


Figure 503.118: Frequency Contents of Ground Motions - Displacement Time Series

Depending on different site conditions, ESSI system will exhibit varied response to excitations with different predomiant frequencies. As we can see from Figure 503.42 to Figure 503.47, the structure sitting on softer soil foundations now exhibits much larger structural response when it is subject to long period excitations. This conclusion has been supported by every single plot that records the response spetra of structures such as Figure 503.61, Figure 503.63 and Figure 503.65. For the long perdominant periods present in the Kocaeli earthquake, the structures supported by soft clays exhibit much larger amplification than those supported by stiff soils.

It is also worthwhile to look deeper into the transfer function that tells some other aspects of the simulation scenarios. We can see from Figure 503.119 to Figure 503.130 that in terms of transfer function, which defines the amplification from soil surface to top of the structure, the structure is able to pick up much high frequency content that is not present within the soil layers. The visual representation of this is that the transfer function has much larger value at the high frequency end (low period end). This can be explained that typically people construct pile foundation inside the stiff soil (or bedrock). In our simulation, we also have a very stiff sustaining layer to provide pile tip resistance. The results shown in transfer functions make sense and also verifies our numerical model. Acceleration results are also shown as following.

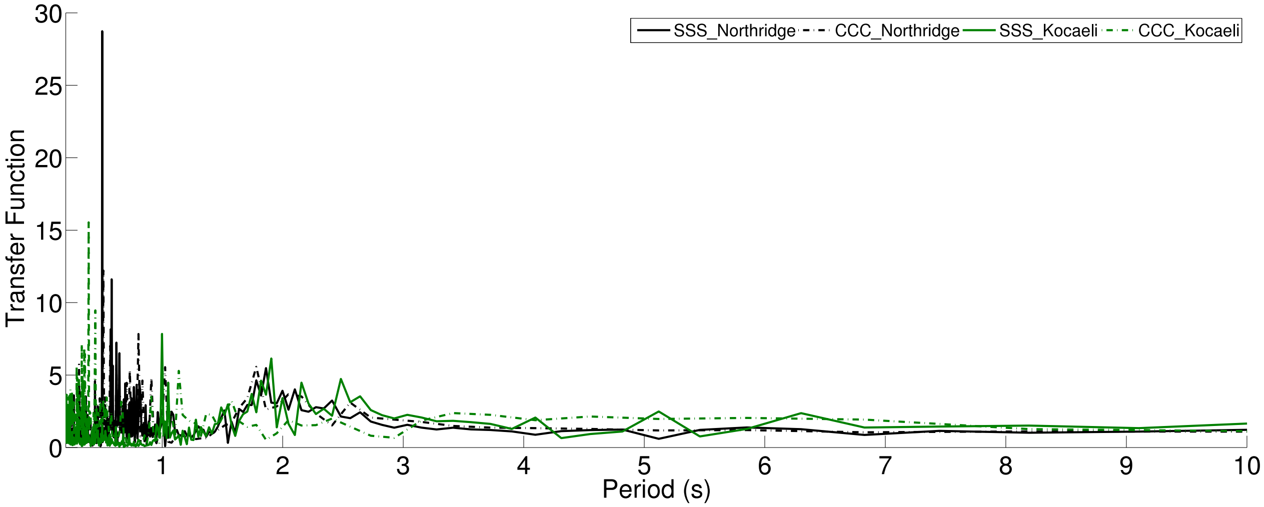


Figure 503.119: Transfer Function of Simulated Displacement Time Series for Both Long Period and Short Period Motions (Structure Bent 1, from Soil Surface to Top of Bent)

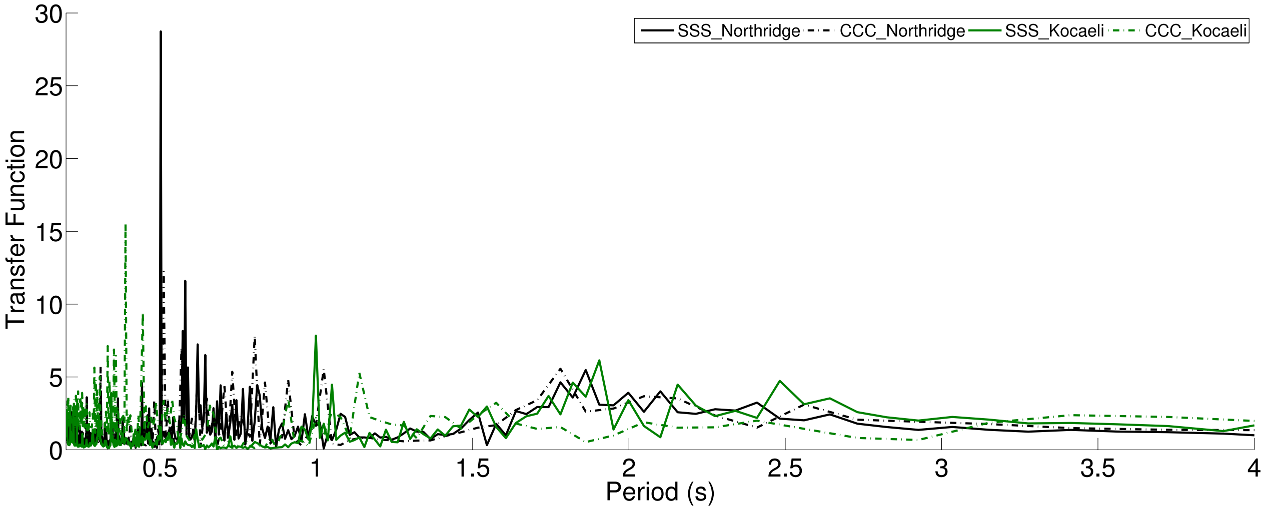


Figure 503.120: Zoomed View: Transfer Function of Simulated Displacement Time Series for Both Long Period and Short Period Motions (Structure Bent 1, from Soil Surface to Top of Bent)

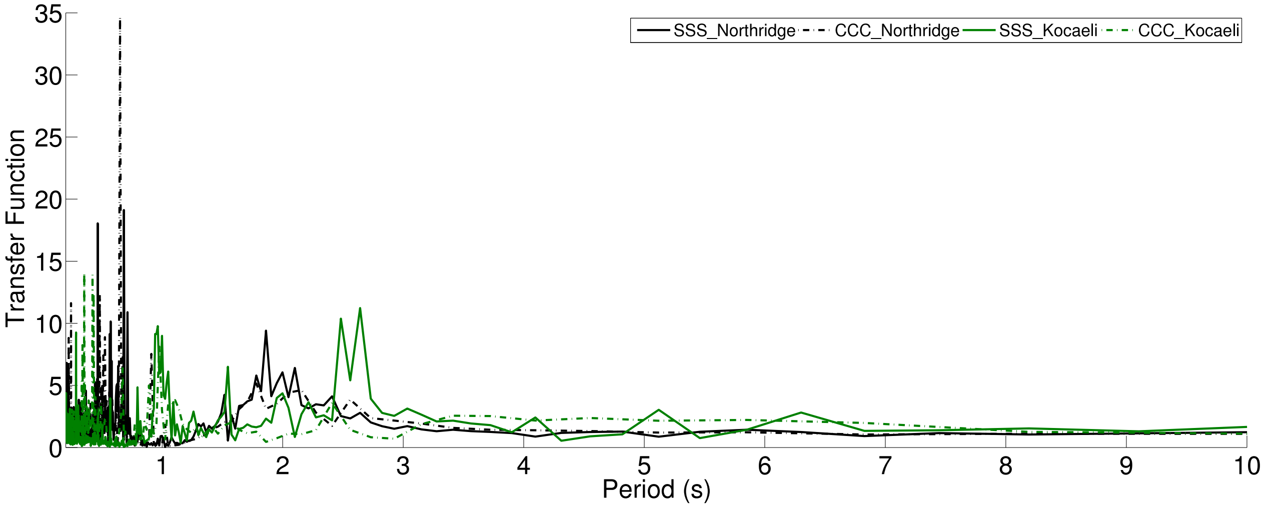


Figure 503.121: Transfer Function of Simulated Displacement Time Series for Both Long Period and Short Period Motions (Structure Bent 2, from Soil Surface to Top of Bent)

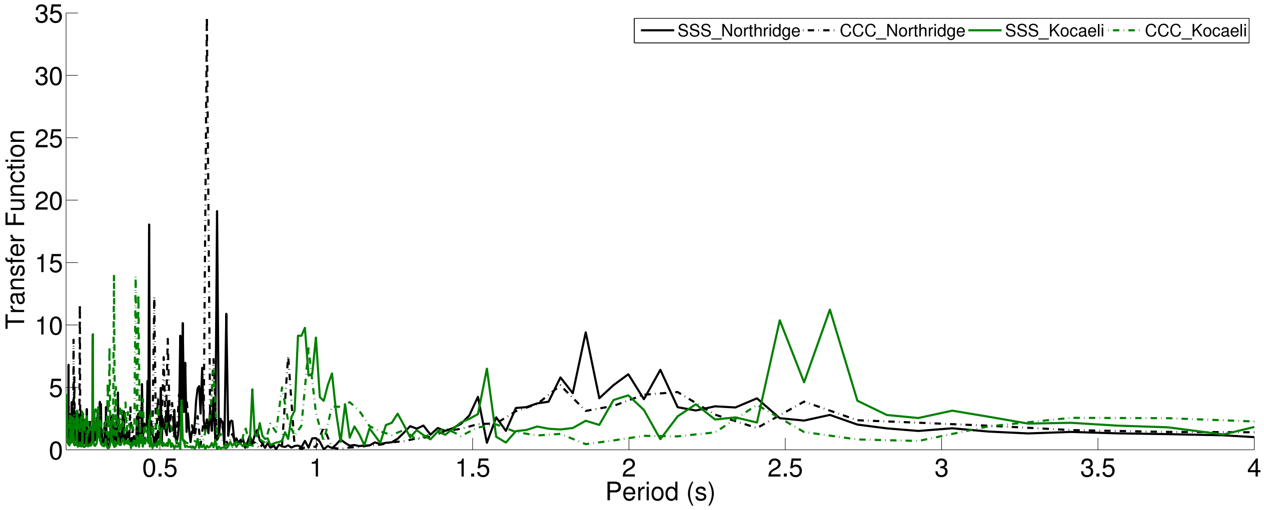


Figure 503.122: Zoomed View: Transfer Function of Simulated Displacement Time Series for Both Long Period and Short Period Motions (Structure Bent 2, from Soil Surface to Top of Bent)

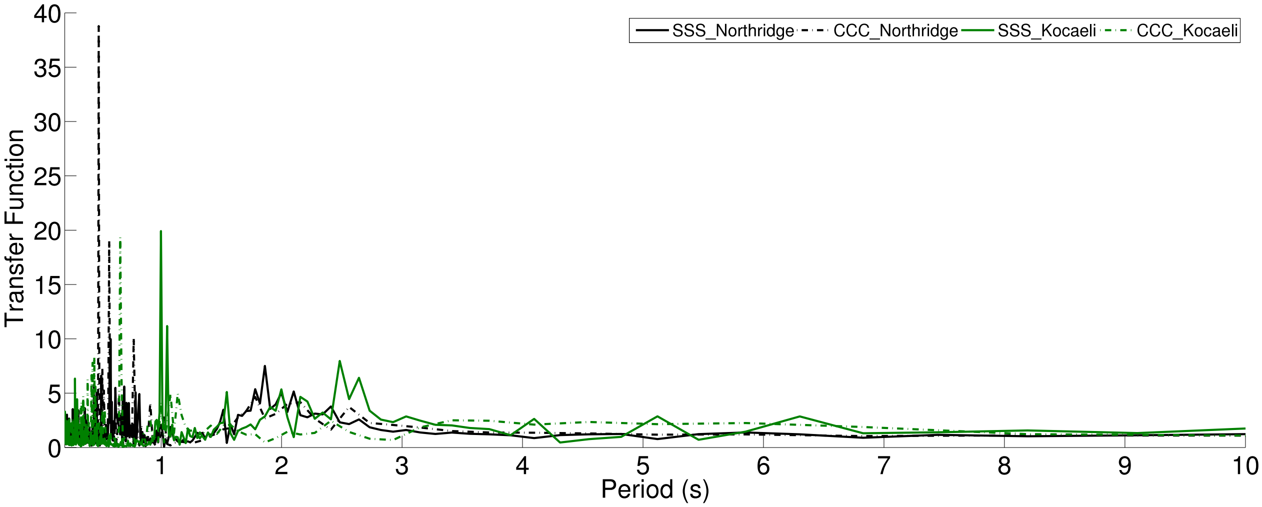


Figure 503.123: Transfer Function of Simulated Displacement Time Series for Both Long Period and Short Period Motions (Structure Bent 3, from Soil Surface to Top of Bent)

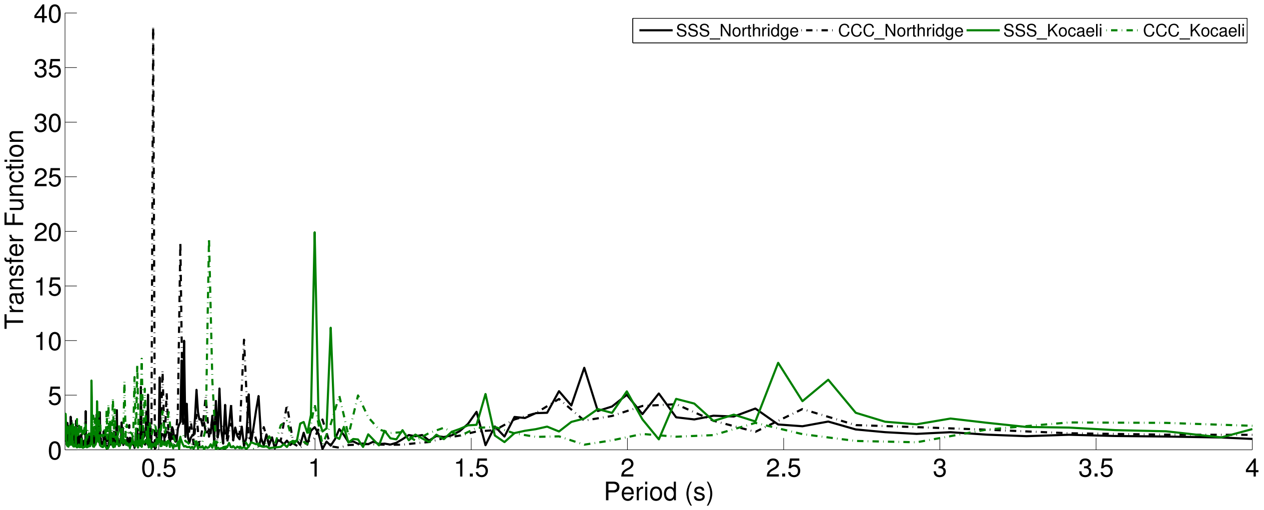


Figure 503.124: Zoomed View: Transfer Function of Simulated Displacement Time Series for Both Long Period and Short Period Motions (Structure Bent 3, from Soil Surface to Top of Bent)

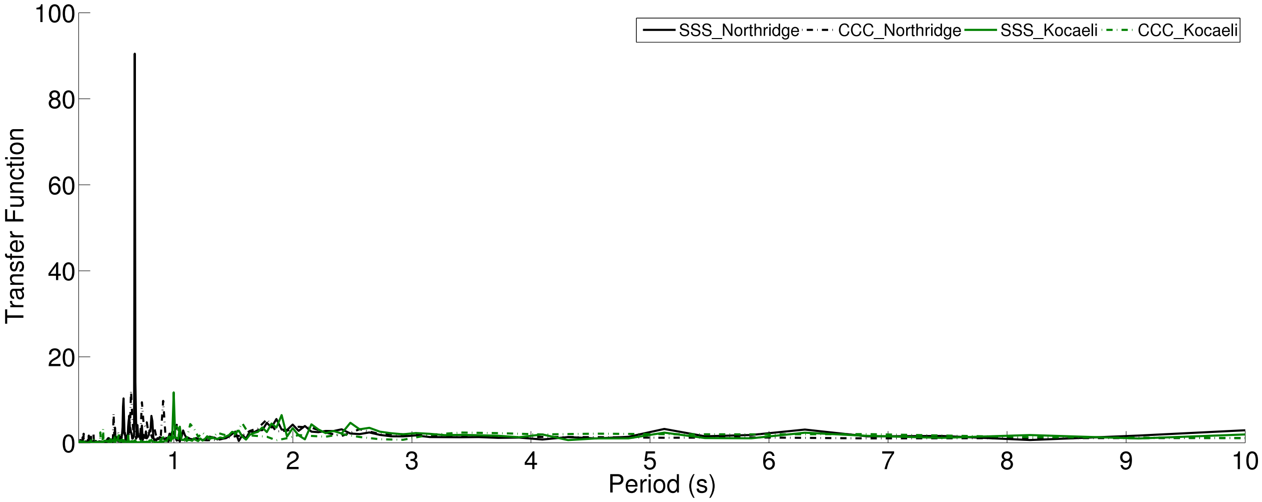


Figure 503.125: Transfer Function of Simulated Acceleration Time Series for Both Long Period and Short Period Motions (Structure Bent 1, from Soil Surface to Top of Bent)

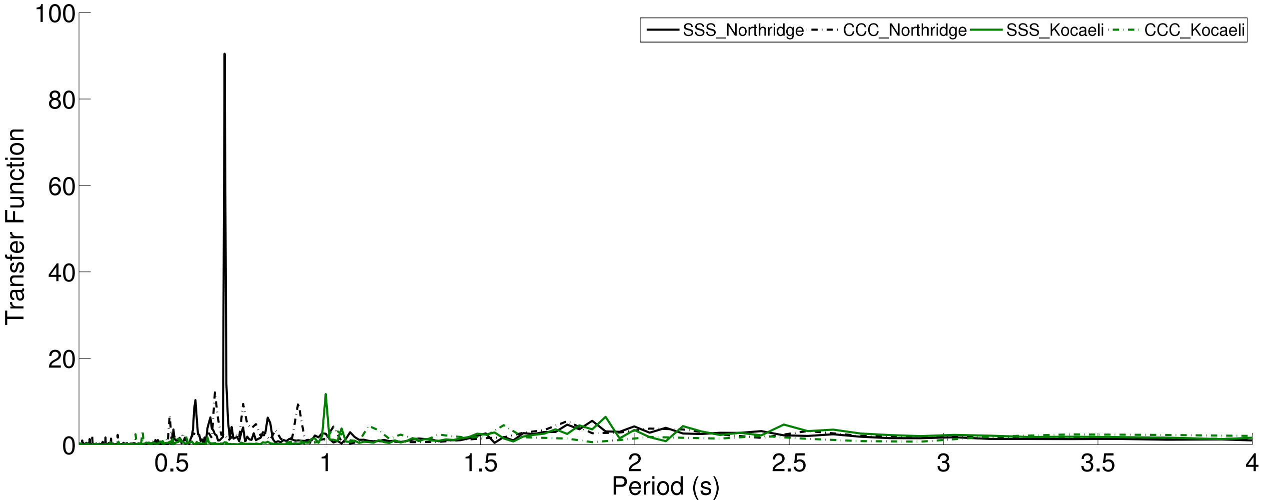


Figure 503.126: Zoomed View: Transfer Function of Simulated Acceleration Time Series for Both Long Period and Short Period Motions (Structure Bent 1, from Soil Surface to Top of Bent)

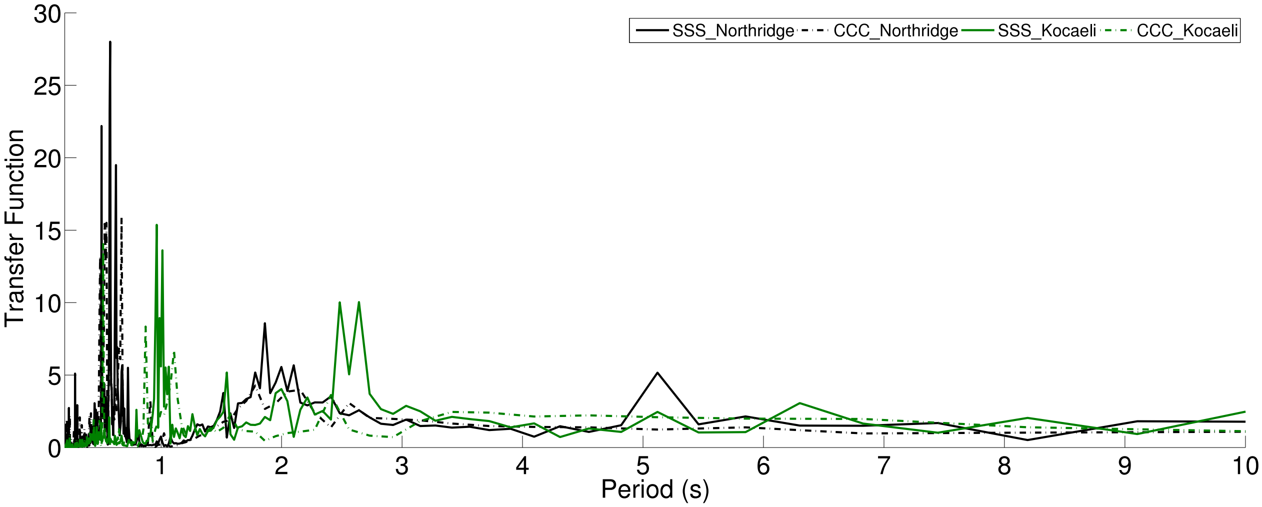


Figure 503.127: Transfer Function of Simulated Acceleration Time Series for Both Long Period and Short Period Motions (Structure Bent 2, from Soil Surface to Top of Bent)

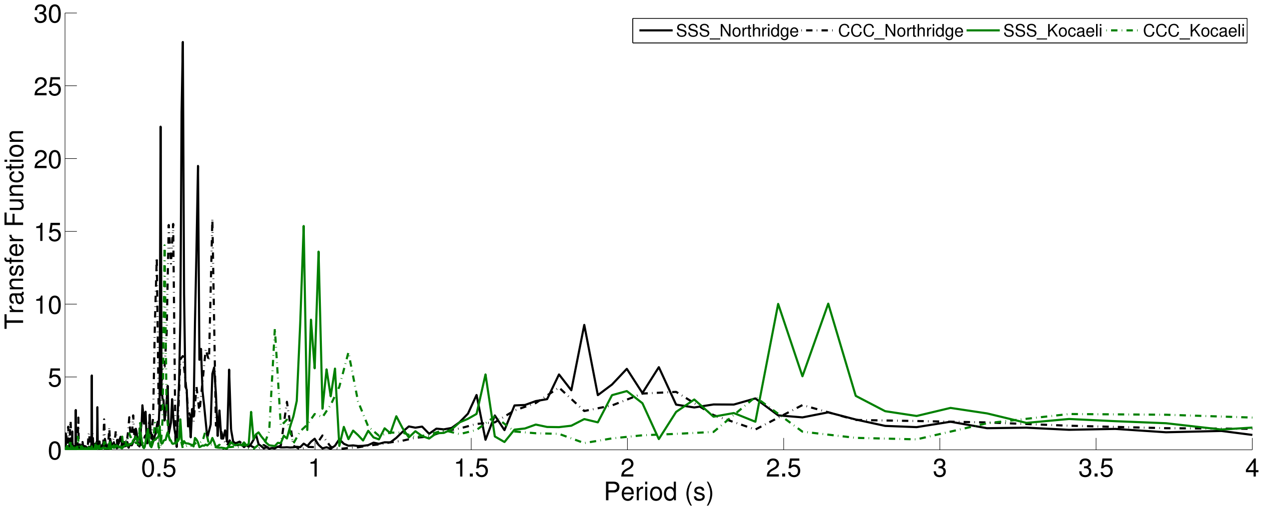


Figure 503.128: Zoomed View: Transfer Function of Simulated Acceleration Time Series for Both Long Period and Short Period Motions (Structure Bent 2, from Soil Surface to Top of Bent)

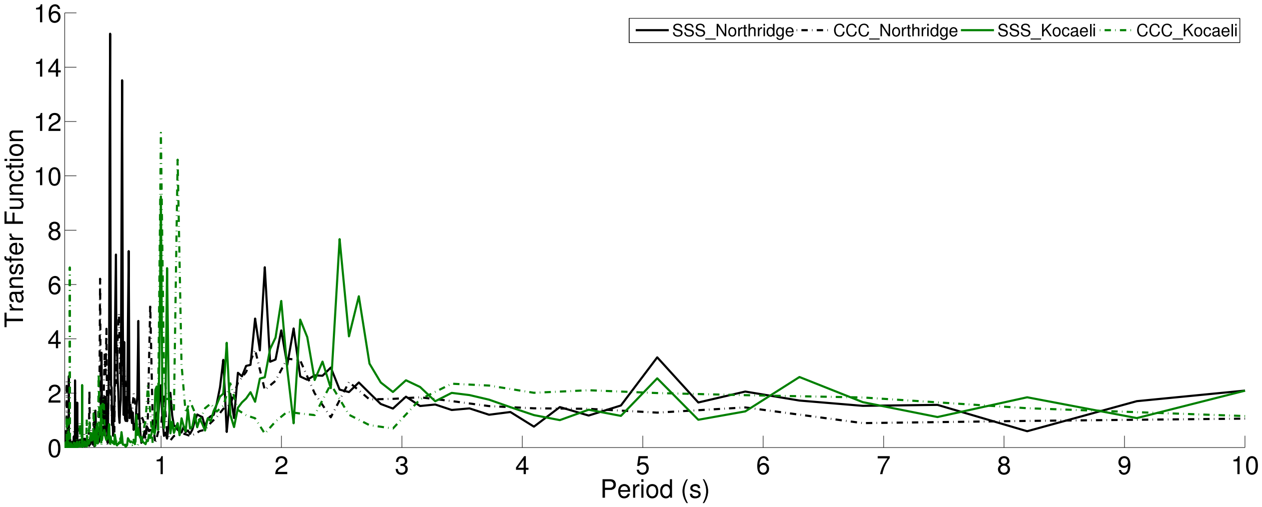


Figure 503.129: Transfer Function of Simulated Acceleration Time Series for Both Long Period and Short Period Motions (Structure Bent 3, from Soil Surface to Top of Bent)

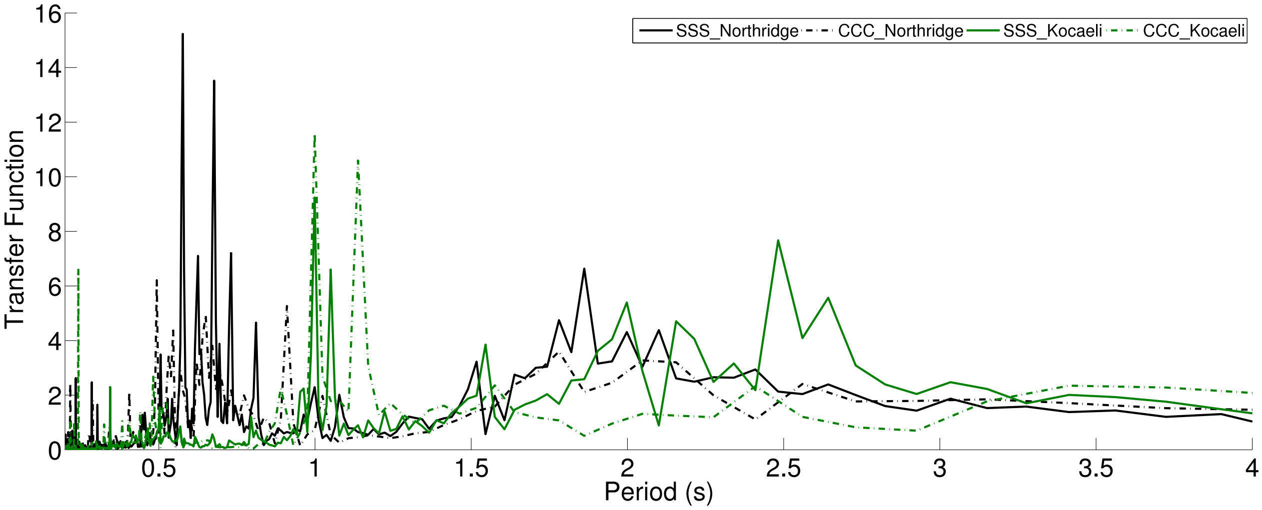


Figure 503.130: Zoomed View: Transfer Function of Simulated Acceleration Time Series for Both Long Period and Short Period Motions (Structure Bent 3, from Soil Surface to Top of Bent)

Chapter 504

Earthquake-Soil-Structure Interaction, Nuclear Power Plants

(2010-2011-2012-2017-2018-2019-2020-2021-2023-)

(In collaboration with Dr. Nima Tafazzoli, Prof. José Abell, Dr. Yuan Feng, Prof. Sumeet Kumar Sinha, Prof. Han Yang and Dr. Hexiang Wang, Dr. Katarzyna Staszewska)

504.1 Stick/Solid Finite Element Model

The 3D finite element model is shown in Figure (504.1). The finite element model is created using both 8 node brick elements for modeling the soil part and 27 node brick elements for modeling the foundation part, and displacement beam element for structural components.

The size of the model is $140m$ along the X and Y directions, and $50m$ along the Z direction (the height of the soil layer). Shear wave velocity of the soil is considered to be $700m/s$ with density of $2200kg/m^3$. Foundation has the height of $5m$ embedded in the soil layer, with size of $90m$ in each horizontal direction. Concrete is chosen to have shear wave velocity of $2000m/s$ with density of $2400kg/m^3$. The size of the elements are $5m$ in each direction for both foundation and soil elements. Structural beam is composed of 12 displacement beam elements attached to each other with different stiffness and mass properties. Domain Reduction Method is used to apply the input effective forces. Both 1C and 3C wave propagation cases are using the same finite element model for second stage of DRM analysis.

Frictional contact/interface elements are placed at the interface of foundation and soil layer. The contact/interface element used here has the same normal and tangential stiffness with magnitude of $10^8 N/m$. Friction ratio and cohesion of the contact/interface element are $\mu = 0.4$ and $c = 0.0$ respectively. Contact elements are oriented along the Z axis of global model.

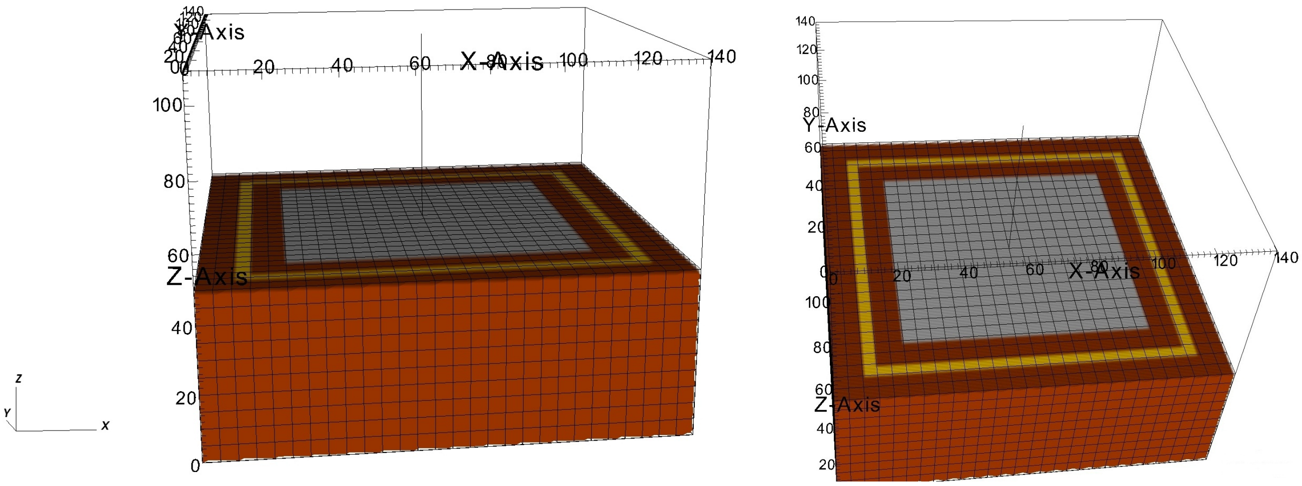


Figure 504.1: 3D finite element SFSI model considering slipping behavior at the interface of foundation and soil layer

In general it is accepted that in order to represent a traveling wave of a given frequency, the size of the finite elements have to have about 10 nodes per wavelength λ (Hughes (1987); Argyris and Mlejnek

(1991)). Using fewer than 10 nodes per wave length λ specially for linear elements leads to numerical damping of higher frequencies as such element discretization misses certain frequencies of the wave. In order to satisfy this requirement, the size of the mesh should satisfy Equation (504.1) which is a function of the maximum frequency (f_{max}) of the input motion and also the shear wave velocity of the media $V_s = \sqrt{G/\rho}$ where G is the shear modulus and ρ is density of the soil.

$$\Delta h \leq \lambda/10 = V_s/(10 f_{max}) \qquad (504.1)$$

In this model, the size of the elements are chosen to be $5m$ in each direction. Considering that the shear wave velocity of the soil is $700m/s$, the maximum frequency able to be propagated through this model is $14Hz$.

504.1.1 Slipping behavior of SFSI models by considering 1C wave propagation

In this section, slipping behavior of SFSI systems is studied under 1C seismic wave propagation assumption to come up with the input motion to be used for the model prepared for second stage of DRM. Morgan Hill earthquake and Ricker wavelets are used as the source of motion.

Domain Reduction Method is used to apply the input effective forces. In order to obtain the displacement and acceleration time histories used for calculating the DRM layer effective forces, a finite element soil column with shear behavior is considered representing the 1C wave propagation. The motions are applied at the base of the 1C finite element model and propagated through the soil column using multiple support excitation pattern.

504.1.1.1 Morgan Hill earthquake

Figures (504.2) to (504.3) show the acceleration and displacement time histories and FFT of the Morgan Hill earthquake at the base of the 1D model. Figure (504.4) shows the acceleration time histories recorded at the bottom and top of the structure in X and Z directions.

As it is observed, the acceleration time history of slipping and no-slipping model has differences in amplitude and the phase in X direction of the model. However, the differences in phases of the recorded motions seems to be more along the Z direction. It shows that considering the slipping behavior has caused a lag in phase of the response. Comparison of displacement time histories of the same models is shown in Figure (504.5) along X and Z directions. As it is shown, displacement time histories along X direction are quite similar in this case while along Z direction they have different amplitudes due to initial settlement and different phases. FFT of the accelerations do not have much of a difference in terms of frequency content (shown in Figure (504.6) but they are different in amplitude which has been observed in acceleration time histories as well. In general it is observed that in this earthquake the the magnitude of accelerations and displacements are less at top of the structure comparing to the ones recorded at the bottom of the structure for both slipping and no-slipping behavior cases.

Figure (504.7) shows the distribution of sliding at the interface of foundation and the soil layer at 9 different time steps of analysis from 0.5 to 1.3 seconds. As it is observed, location of maximum sliding is changed along the foundation in time while magnitude and direction of the applied motion changes. Slipping happens at the specific steps and parts of the interface zone in which the applied force is more than the resistant one.

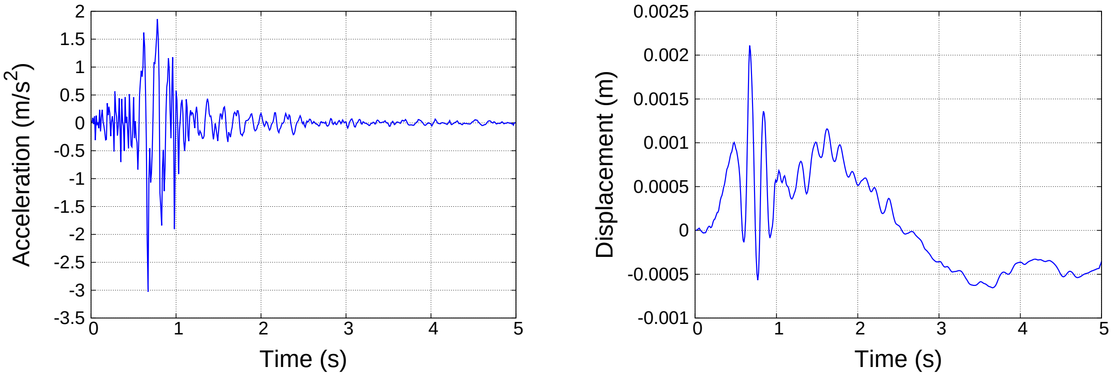


Figure 504.2: Acceleration and displacement time histories of Morgan Hill earthquake

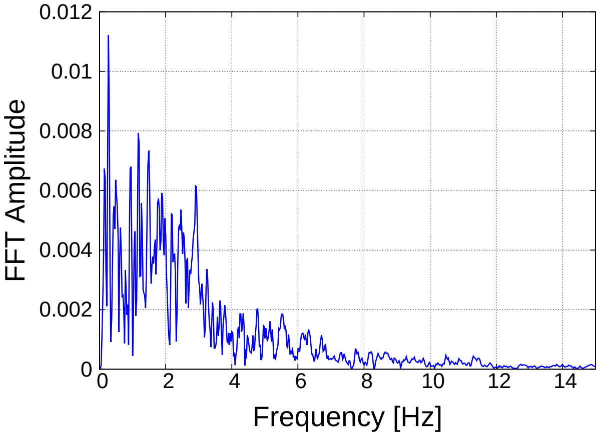


Figure 504.3: FFT of Morgan Hill earthquake

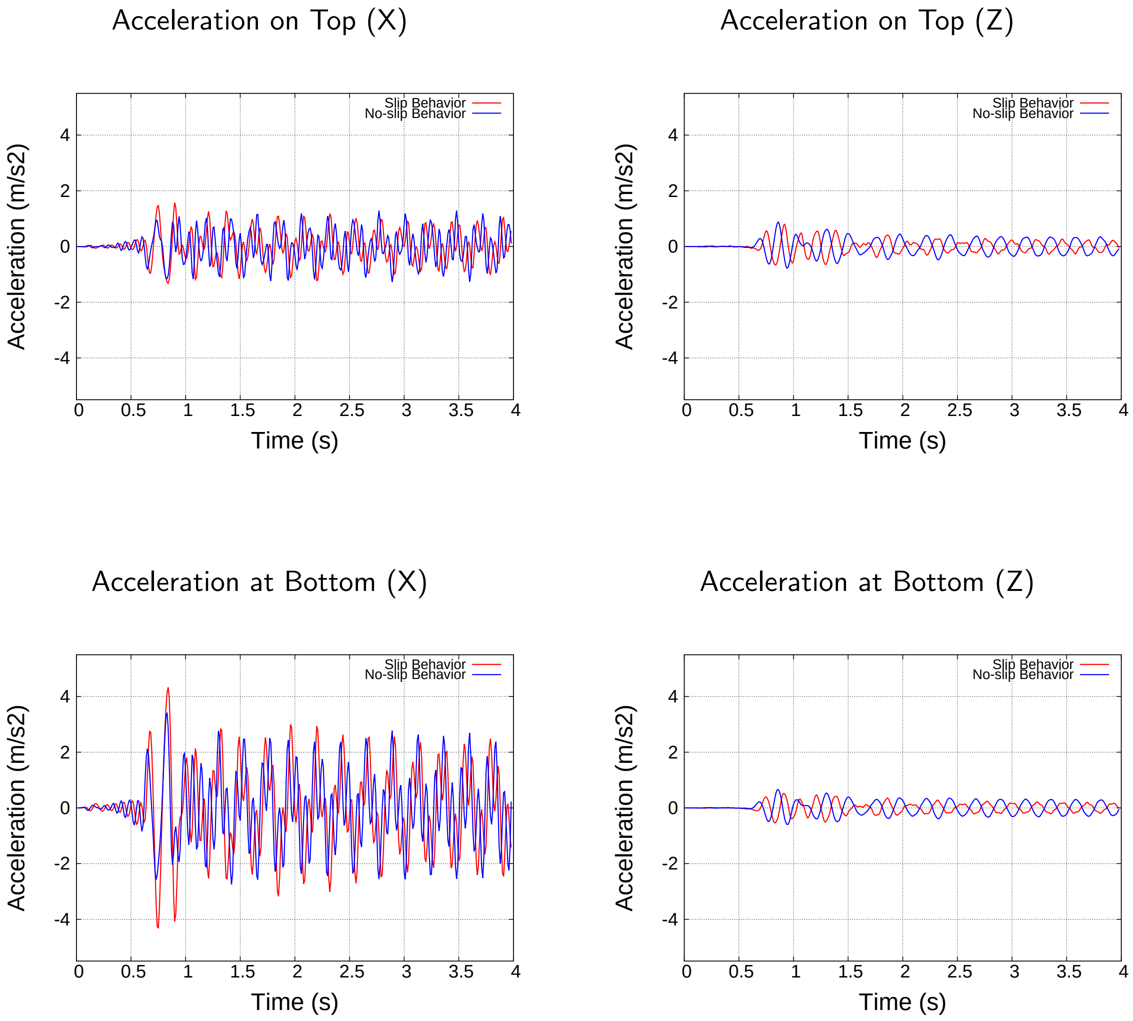


Figure 504.4: Comparison of acceleration time histories of the structure between slipping and no-slipping models for Morgan Hill earthquake

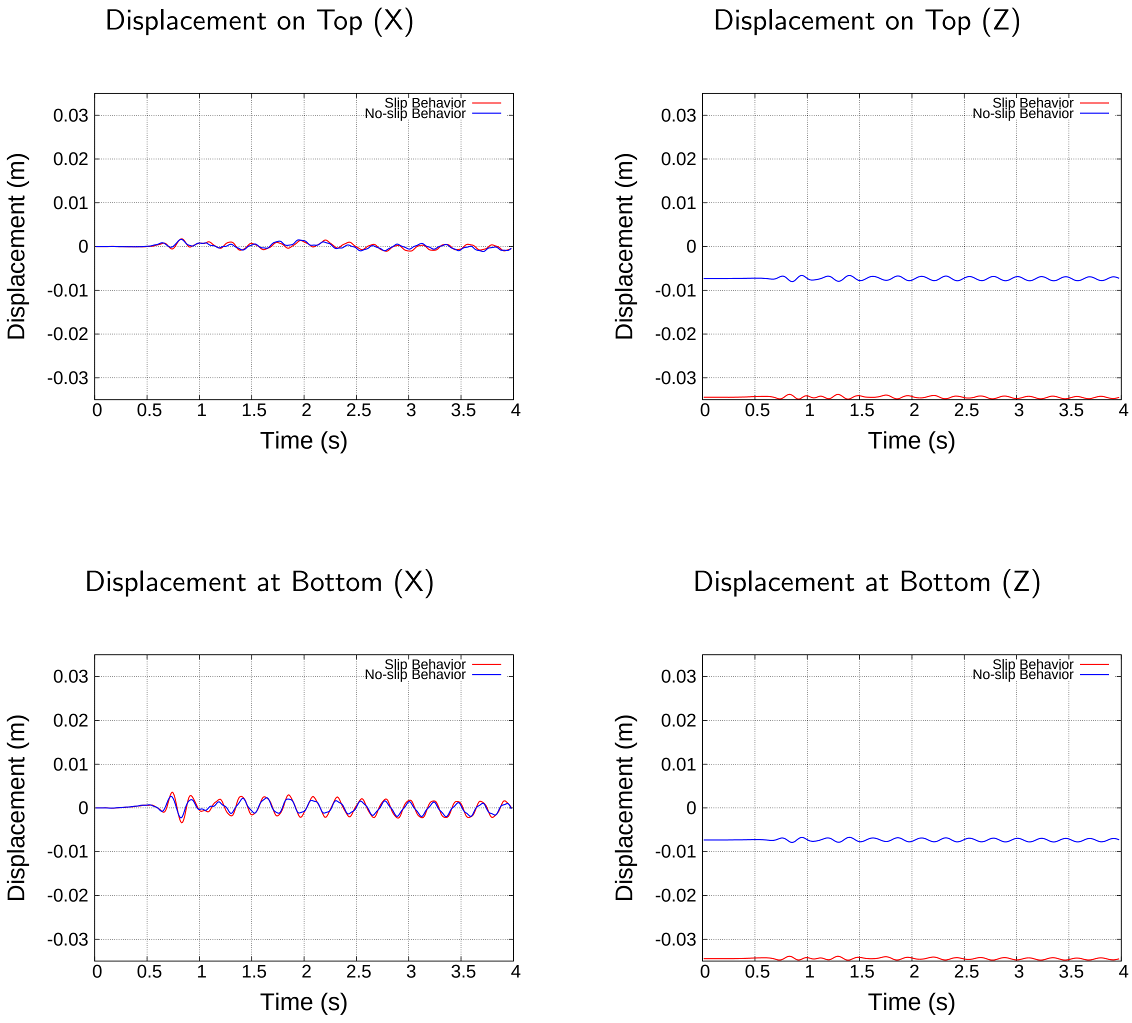
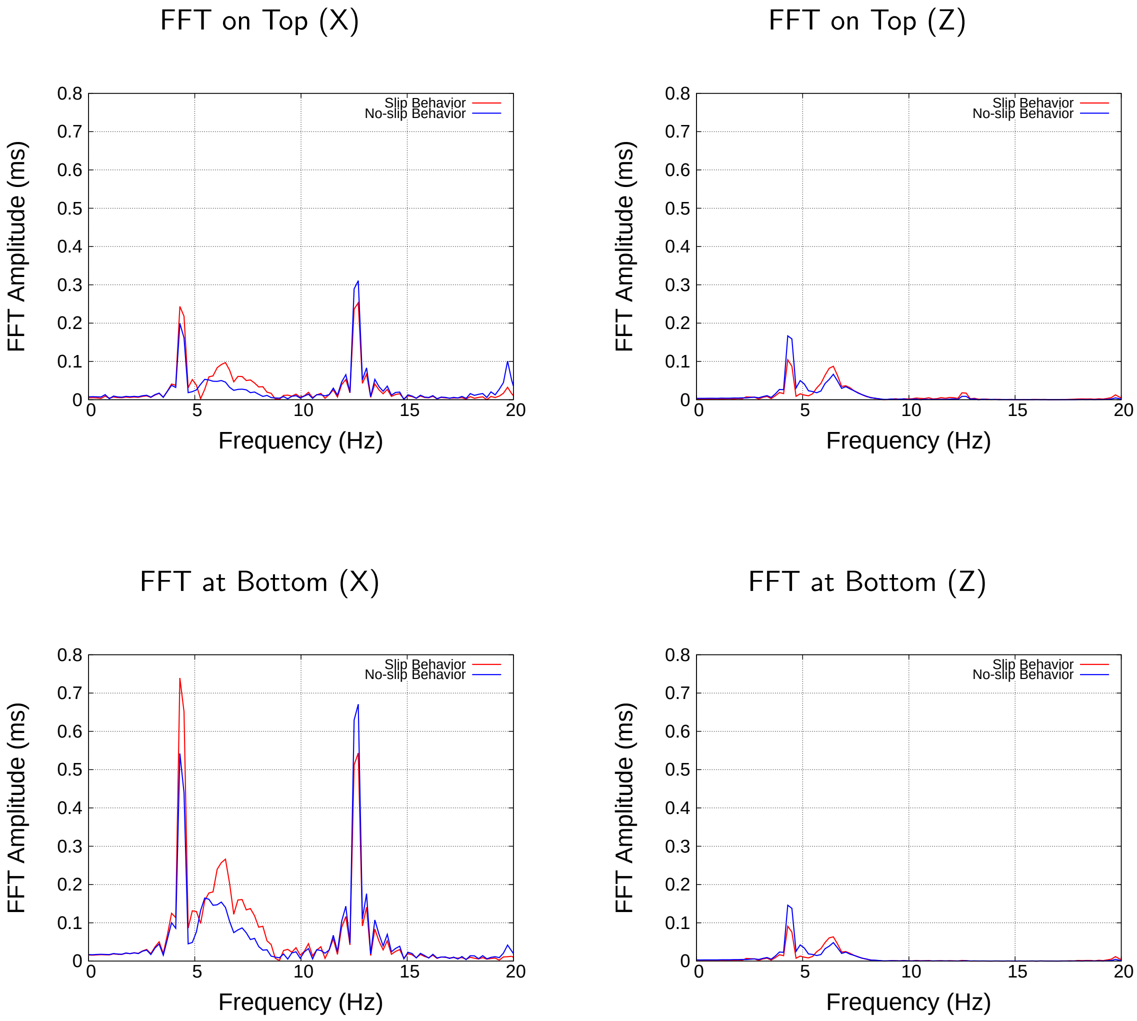


Figure 504.5: Comparison of displacement time histories of the structure between slipping and no-slipping models for Morgan Hill earthquake



FFT on Top (X)

FFT on Top (Z)

FFT at Bottom (X)

FFT at Bottom (Z)

Figure 504.6: Comparison of FFT of the acceleration of the structure between slipping and no-slipping models for Morgan Hill earthquake

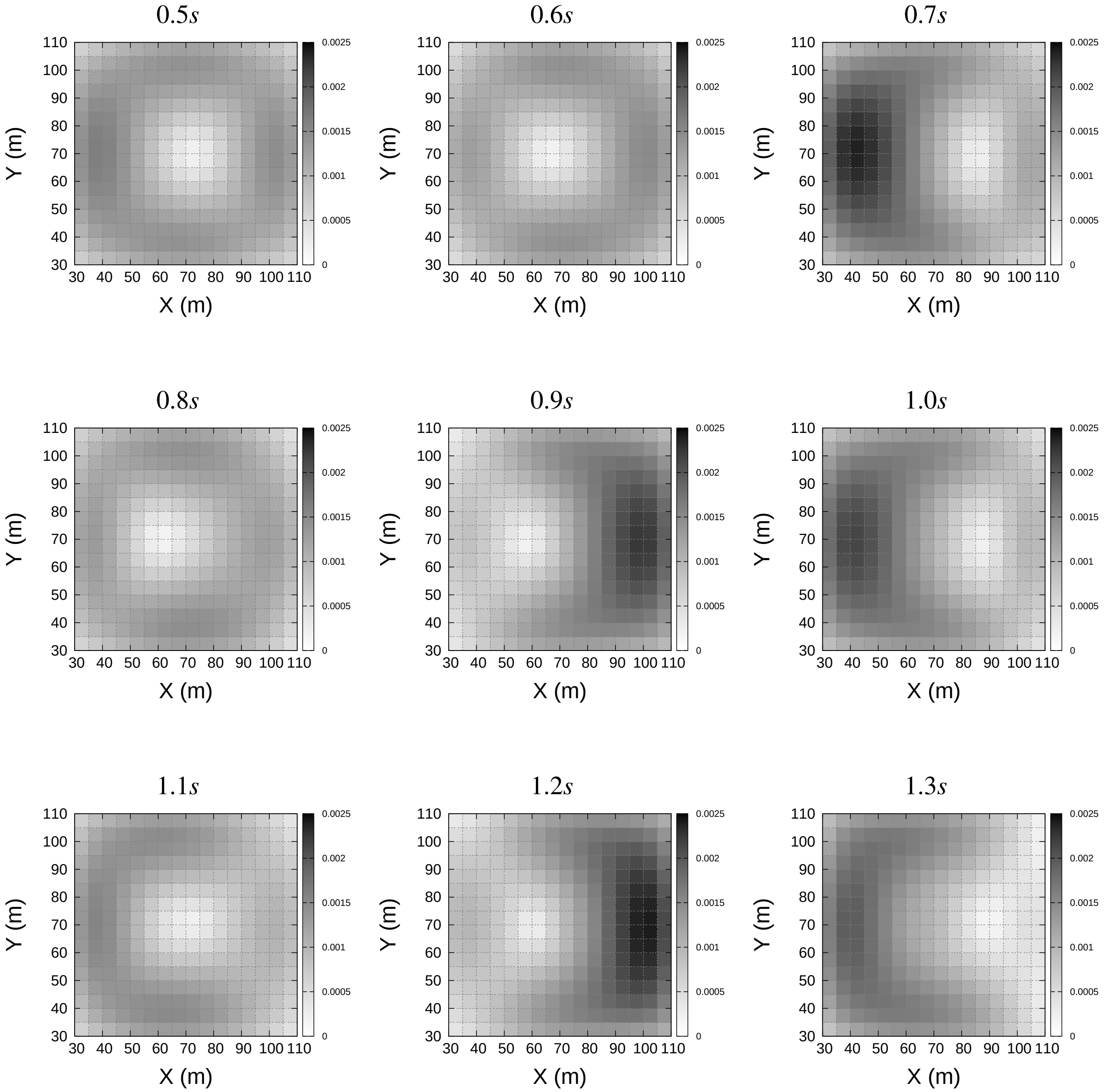


Figure 504.7: Distribution of sliding along the contact/interface for Morgan Hill earthquake (gray scale given in meters)

504.1.1.2 Ricker wave

In order to investigate more, the Ricker wave with dominant frequency of $1Hz$ is used for analysis of the same model. The maximum of this function happens at 1 second. Figures (504.8) and (504.9) show the acceleration and displacement time histories as well as the FFT of the Ricker wave respectively.

As shown in Figure (504.10), displacement time history of top of the structure along X direction is higher when slipping behavior is not considered, while along Z direction it is higher considering the slipping behavior. Since structural components have rotational degrees of freedom, the slipping behavior and gaps in the model will lead to have more rotational movement and rocking of the foundation. There is also a phase lag in response of the structure in slipping behavior model. The displacement time histories at the bottom of the structure are slightly different in X direction. It can also be observed that along Z direction, the response of the structure in slipping model is higher comparing to the no-slipping one.

In this analysis slipping behavior make the movement to be less along the X direction. This might be due to the fact that while the foundation and structure are moving, there are gaps created at some parts of the interface which will affect the rocking movement of the foundation.

Figure (504.11) shows the FFT of the acceleration at the bottom and top of the structure. As it is observed, there is a slight shift in predominant frequency of the response between the slipping and no-slipping behavior. By considering the slipping behavior, the dominant frequency of the motion is decreased along the X direction while is increased along the Z direction. This means that the system gets softer along X direction and stiffer along Z direction. This is due to the gap openings and slidings at the foundation and soil layer interface. Since sliding happens along the X direction, and also considering the gap openings occur along the Z direction which can lead to the rocking of the foundation, it makes the system softer along the X direction. In general this shows the fact that the natural frequency of the system can be changed due to the slipping behavior.

Figures (504.12) and (504.13) show the distribution of gap openings and slidings at the interface of foundation and the soil layer at 9 different time steps of analysis from 0.5 to 1.3 seconds. It can be observed from Figure (504.12) how the location of gap openings are changed on different parts of the contact/interface as the dynamic motion is applied. The location of gap openings are changed from one side to the other while the magnitude and direction of the applied motion changes. The maximum of the gap openings at these time steps happen at 1.3 second with maximum amount of $0.1m$ on the right side of the interface. Maximum sliding at the shown range of time steps happens at 1.0 second with maximum value of $0.07m$ on the left side of the interface.

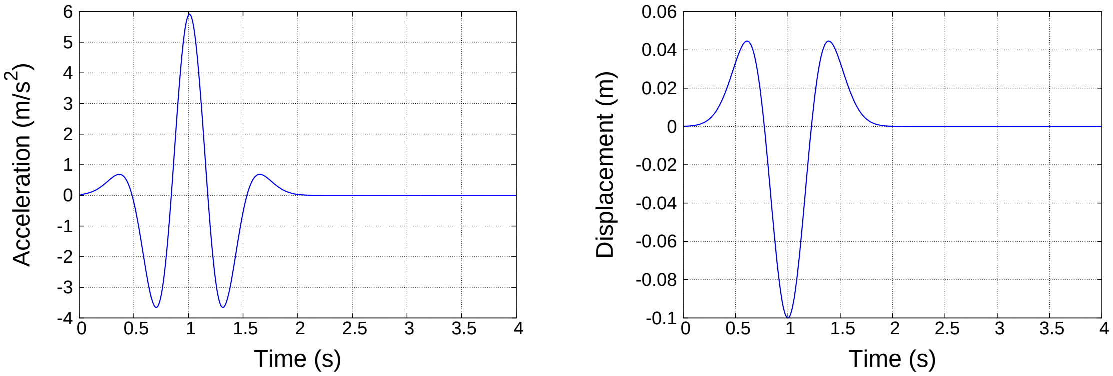


Figure 504.8: Acceleration and displacement time histories of Ricker wave with dominant frequency of $1 Hz$

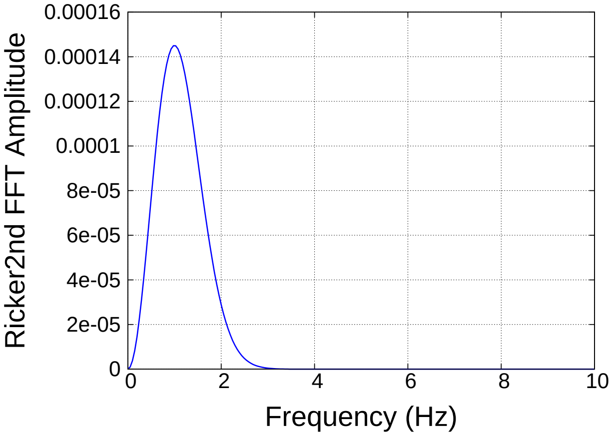


Figure 504.9: FFT of Ricker wave with dominant frequency of $1 Hz$

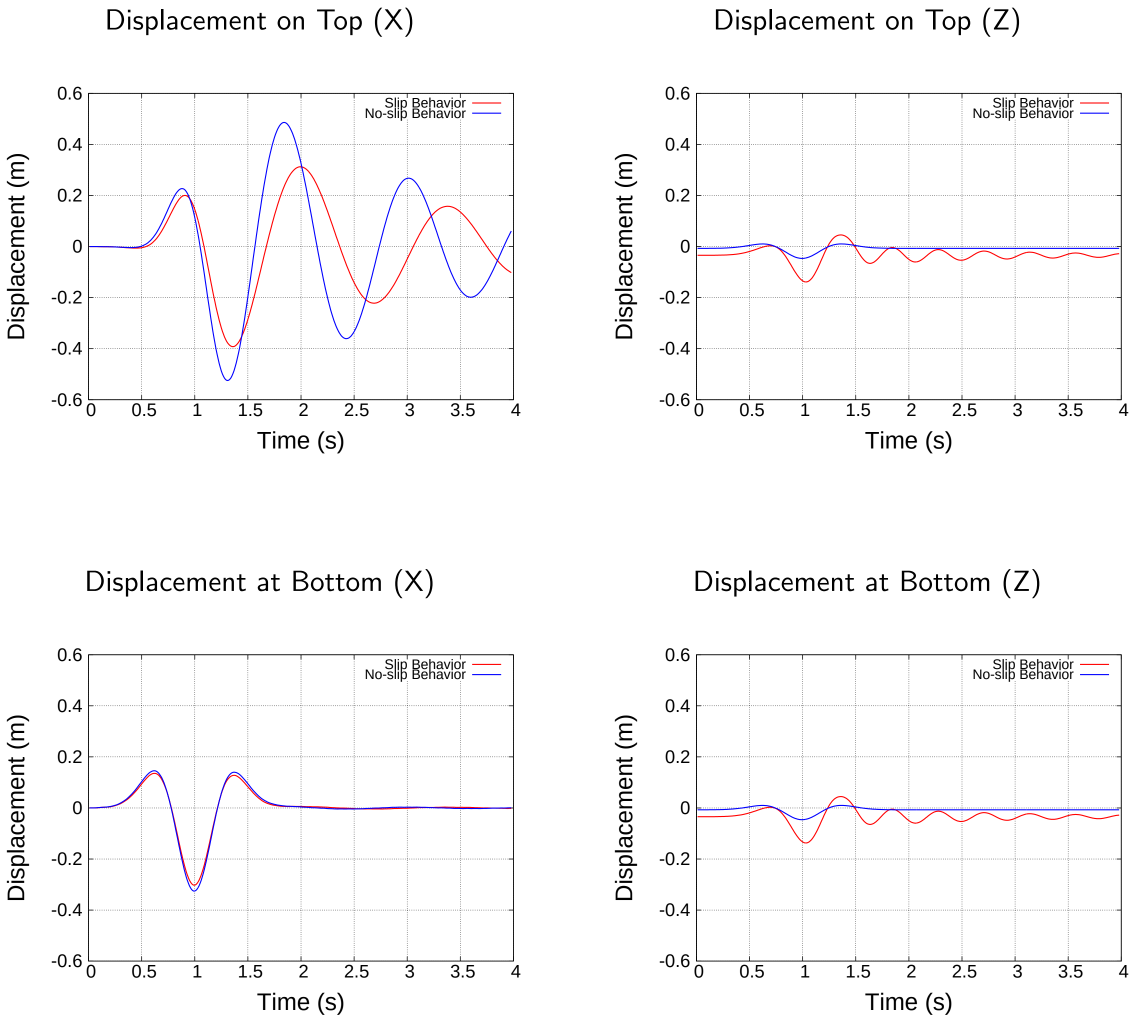
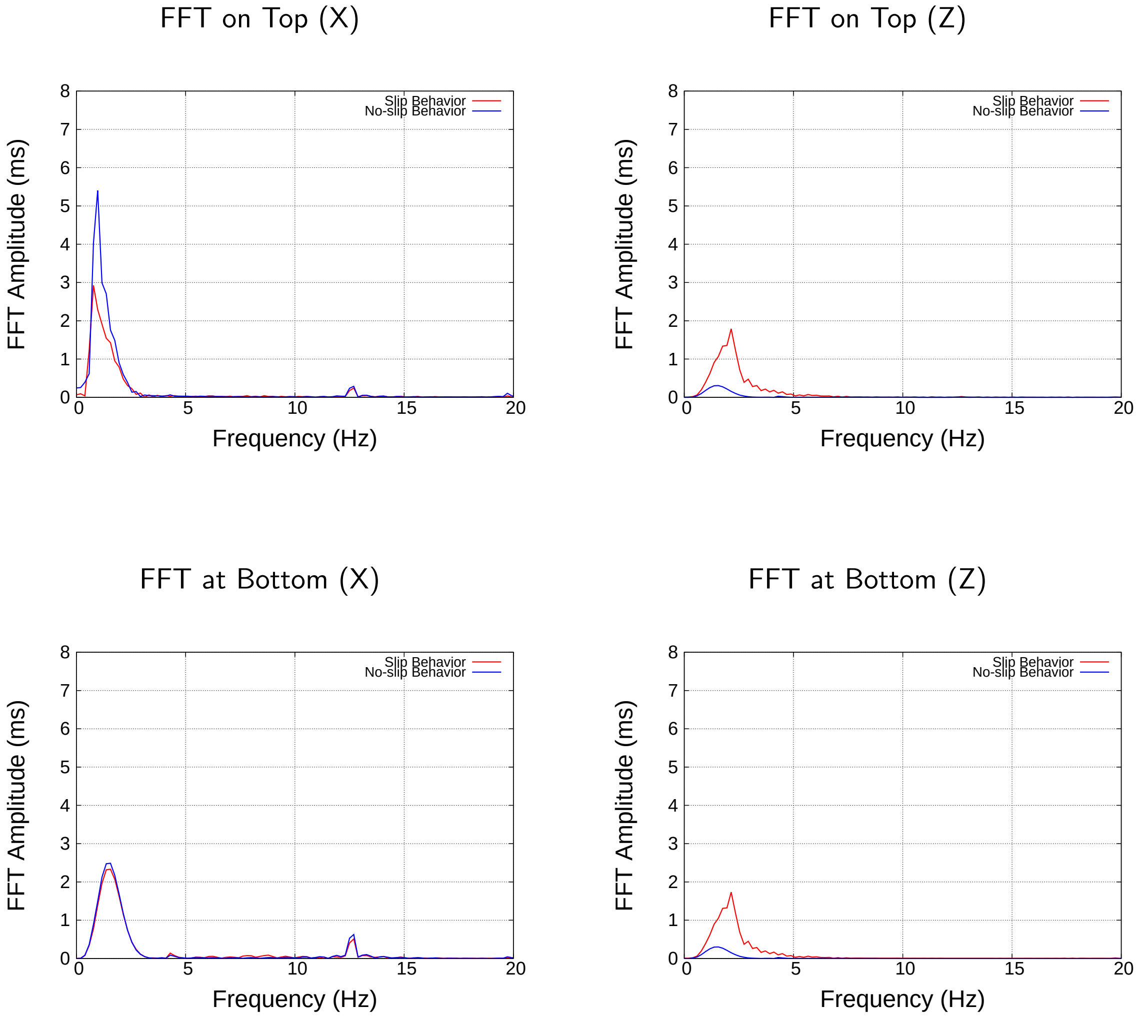


Figure 504.10: Comparison of displacement time histories of the structure between slipping and no-slipping models for Ricker wave



FFT on Top (X)

FFT on Top (Z)

FFT at Bottom (X)

FFT at Bottom (Z)

Figure 504.11: Comparison of FFT of the acceleration of the structure between slipping and no-slipping models for Ricker wave

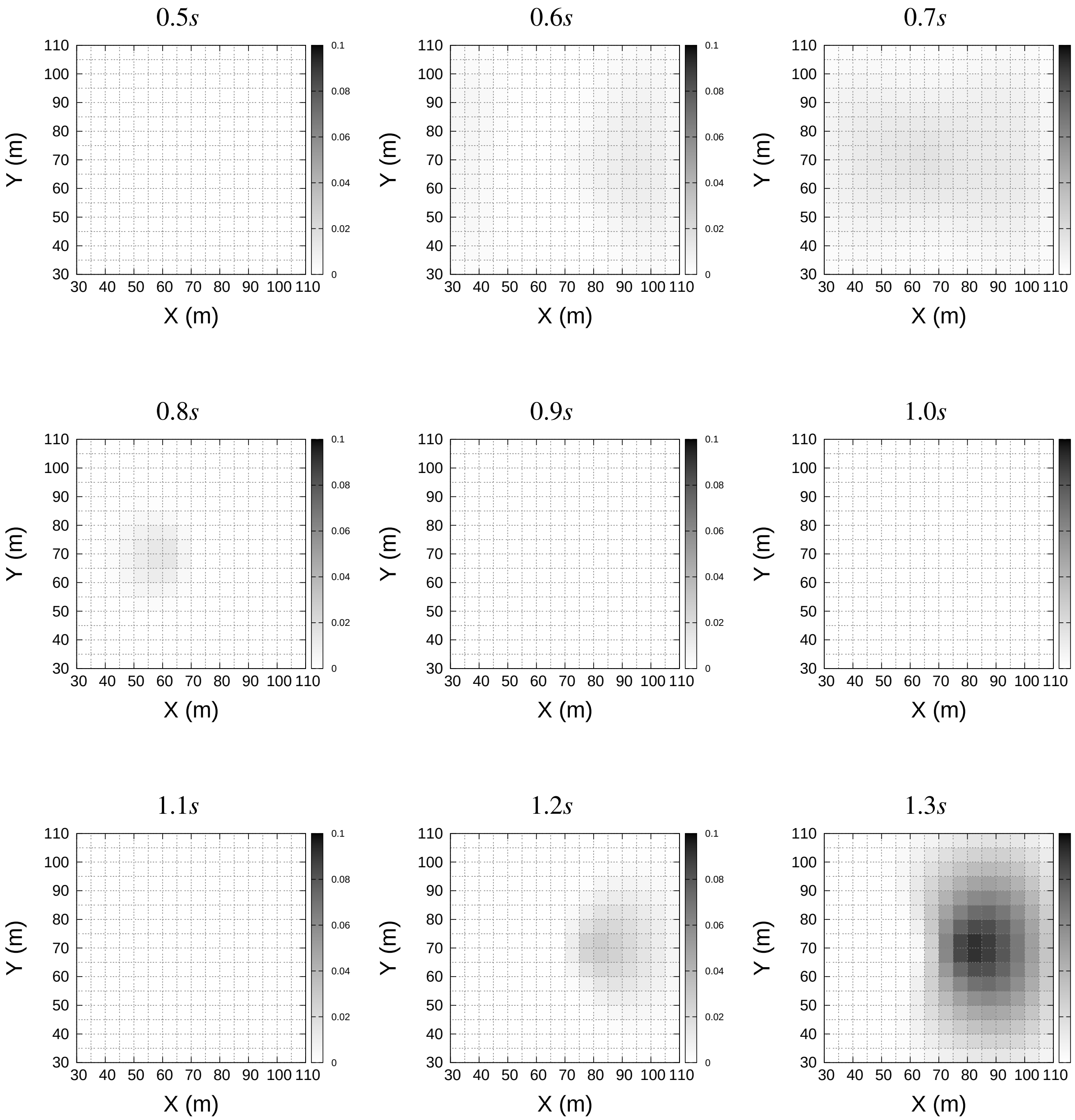
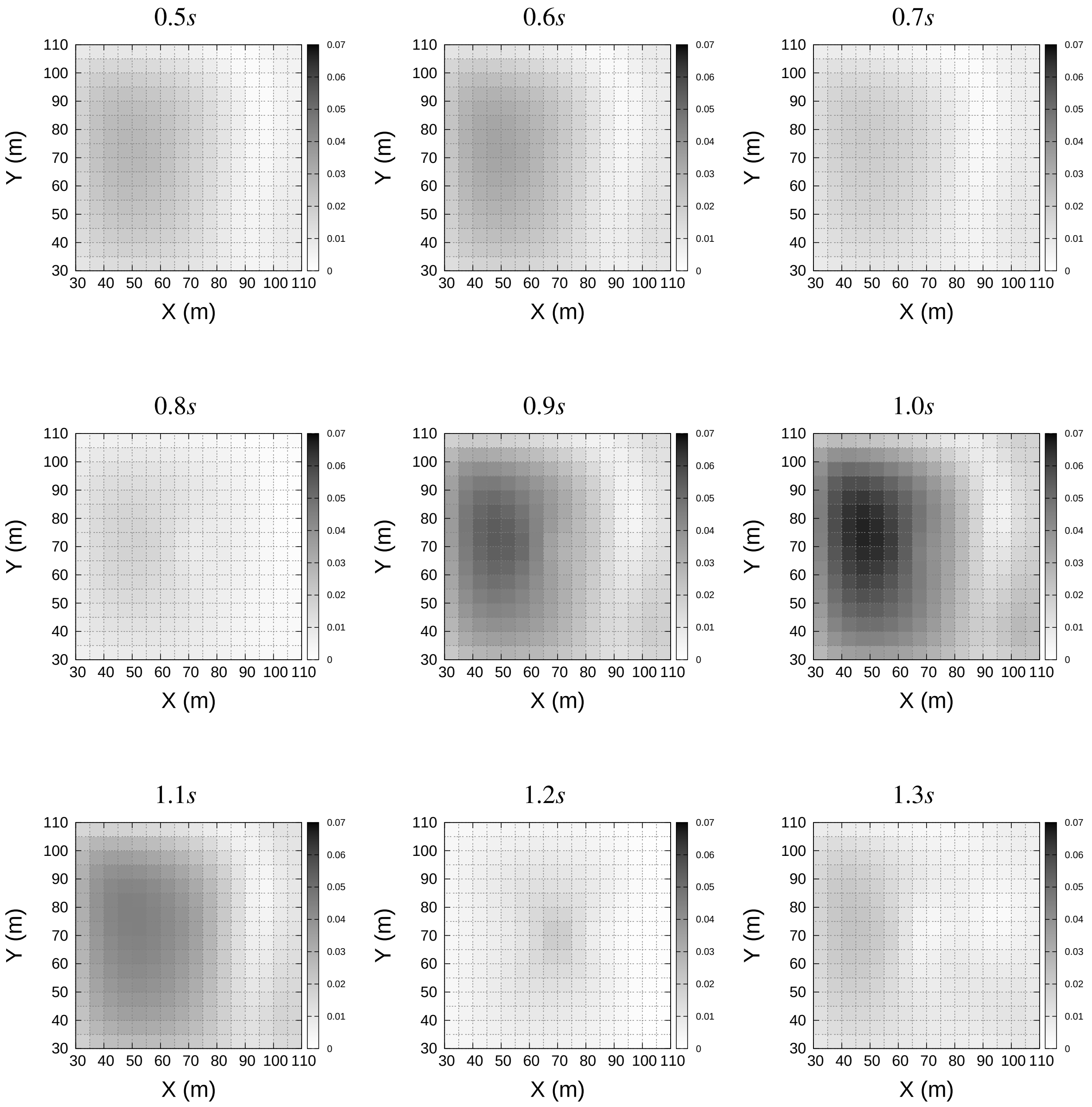


Figure 504.12: Distribution of gap openings along the contact/interface for Ricker wave (gray scale given in meters)



0.5s 0.6s 0.7s 0.8s 0.9s 1.0s 1.1s 1.2s 1.3s

Figure 504.13: Distribution of sliding along the contact/interface for Ricker wave (gray scale given in meters)

504.1.2 Slipping behavior of SFSI models by considering 3C wave propagation

In this section, the same finite element SFSI model as previous section is used with 3C wave propagated motions as input motions for simulations. The input motion used here is Ricker wave (as shown in Figure (504.8)).

In order to study the slipping behavior of SFSI considering 3C wave propagation for first stage of DRM simulation, a finite element model with dimensions of $10000m \times 50m \times 5000m$ is considered. Two cases are studied here with the source of motion (fault) to be located at $(x = 3000m, y = 0, z = 3000m)$ and $(x = 3000m, y = 0, z = 3000m)$. Figures (504.14) and (504.15) show these two cases respectively.

The size of the elements is chosen to be $50m$ in all directions for both cases in order to reduce the computational time. The soil parameters are: shear wave velocity of $700m/s$ and density of $2200kg/m^3$. Analyses for the fault slip model are done by applying the motion at the nodes of one element. This is done in order to represent the the wave propagation starting from the fault using multiple support excitation pattern. This is representing the first stage of analysis of DRM in which a big model including the fault is considered for free field case in order to obtain the required motions for DRM layer.

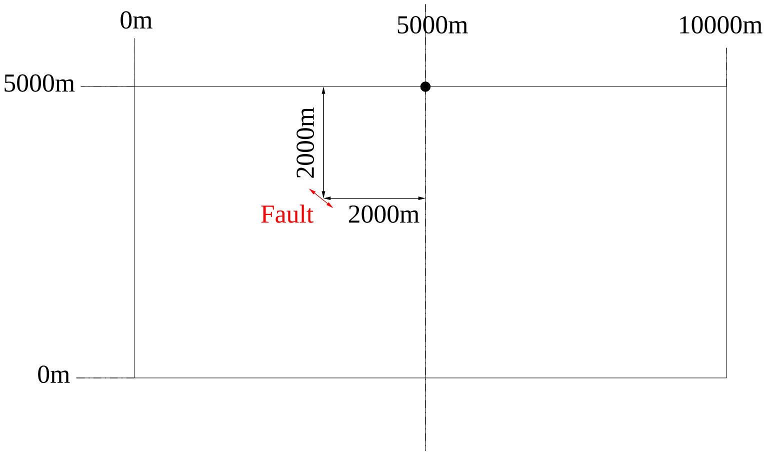


Figure 504.14: Domain to be analyzed for the 1^{st} stage of DRM with fault located at an angle of 45° with respect to the top middle point of the model

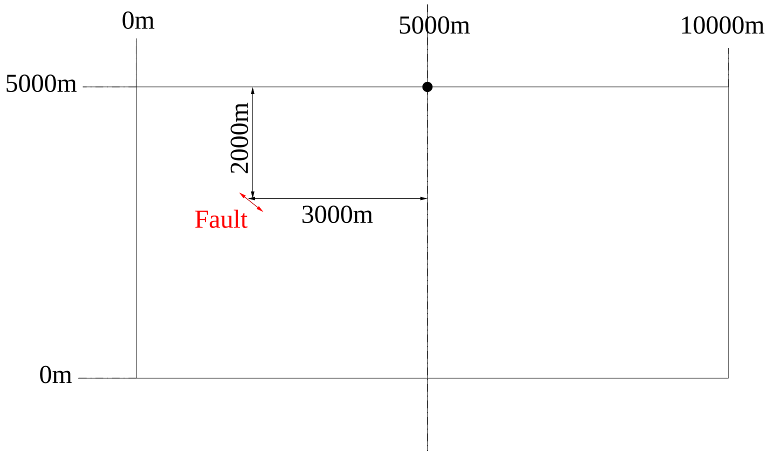


Figure 504.15: Domain to be analyzed for the 1^{st} stage of DRM with fault located at an angle of 34° with respect to the top middle point of the model

504.1.2.1 Ricker wave, with fault located at 45° towards the top middle point of the model

For the first case to be studied here, Ricker wavelet is considered as an input motion with dominant frequency of $1Hz$. The fault is located with angle of 45° toward the top middle point of the model (Figure (504.14)).

Displacement time histories of the structure along X and Z directions are shown in Figure (504.16). It is observed that along the X direction, the amplitude of displacement is little less at the beginning of the shaking of the slipping behavior model and a slight phase lag can be observed as well. However, when the actual pick of the motion is gone, it is shown that the motion for no-slipping model will damp out sooner. So the magnitude of displacements are higher at this time range for slipping model which could be because of the gap/sliding at the interface zone. The displacement time histories do not have a significant difference at the bottom of the structure. Displacement time histories observed along the Z direction shows the fact that magnitude of displacements for slipping model is higher and will damp out later comparing to the no-slipping model.

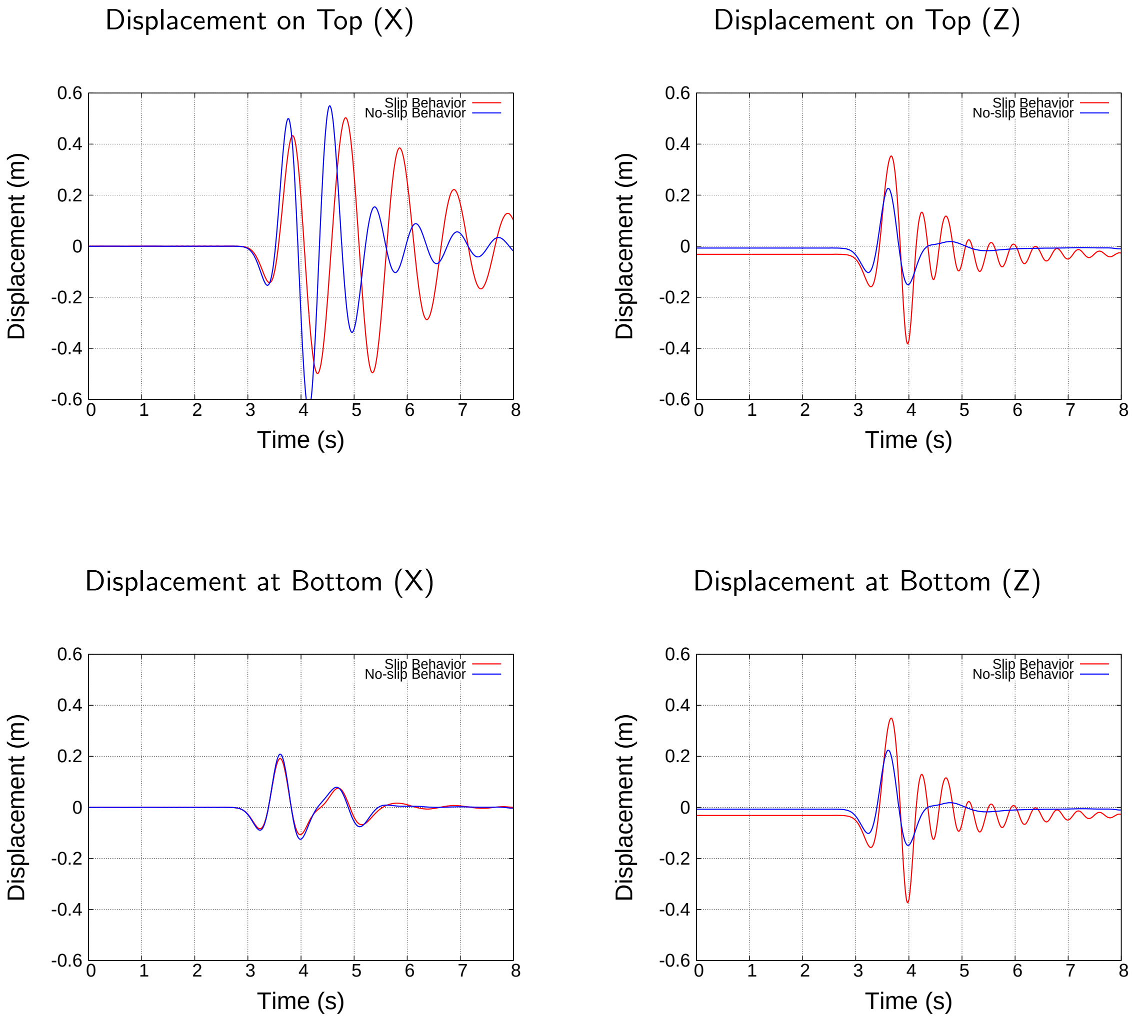
Acceleration time histories along X and Z directions at the bottom and top of the structure are shown in Figure (504.17). It is observed that along the X direction the amplitude of acceleration is less considering the slipping behavior and also there is a phase lag in the observed motion. However, the amplitude of the acceleration along Z direction is much higher in case of considering the slipping case. This is the same behavior observed in 1C wave propagation as well but with higher difference in time histories amplitudes.

As shown in Figure (504.18), the dominant frequency of the response in case of slipping is less than the one observed in no-slipping case along X direction while it is higher along Z direction.

Figures (504.19) and (504.20) show the distribution of sliding at the interface of foundation and the soil layer at 9 different time steps of analysis from 4.5 to 5.3 seconds. It can be observed from Figure (504.19) that maximum gap opening of $0.12m$ is occured at 4.7 seconds while the location of the openings are changed during the analysis. In addition, maximum sliding at the interface zone in this case happens at 4.8 seconds with magnitude of $0.03m$. In both gap and slide distribution plots, it can be observed that the place of maximum is close to the middle of the foundation which is where the structure is located.

Distribution of cumulative dissipated energy due to sliding of the foundation and soil layer contact/interface zone is shown in Figure (504.21). By modeling the slipping behavior at the interface zone, part of the seismic energy is dissipated through the sliding and rocking of the foundation and therefore, less amount will be transferred to the structural components. Figure (504.22) shows how energy can be dissipated during the analysis for the point at the middle of the interface zone (location of the structure).

When sliding happens, some part of the energy is dissipated as shown while there will be no change in dissipated energy if the foundation and soil are sticking to each other.



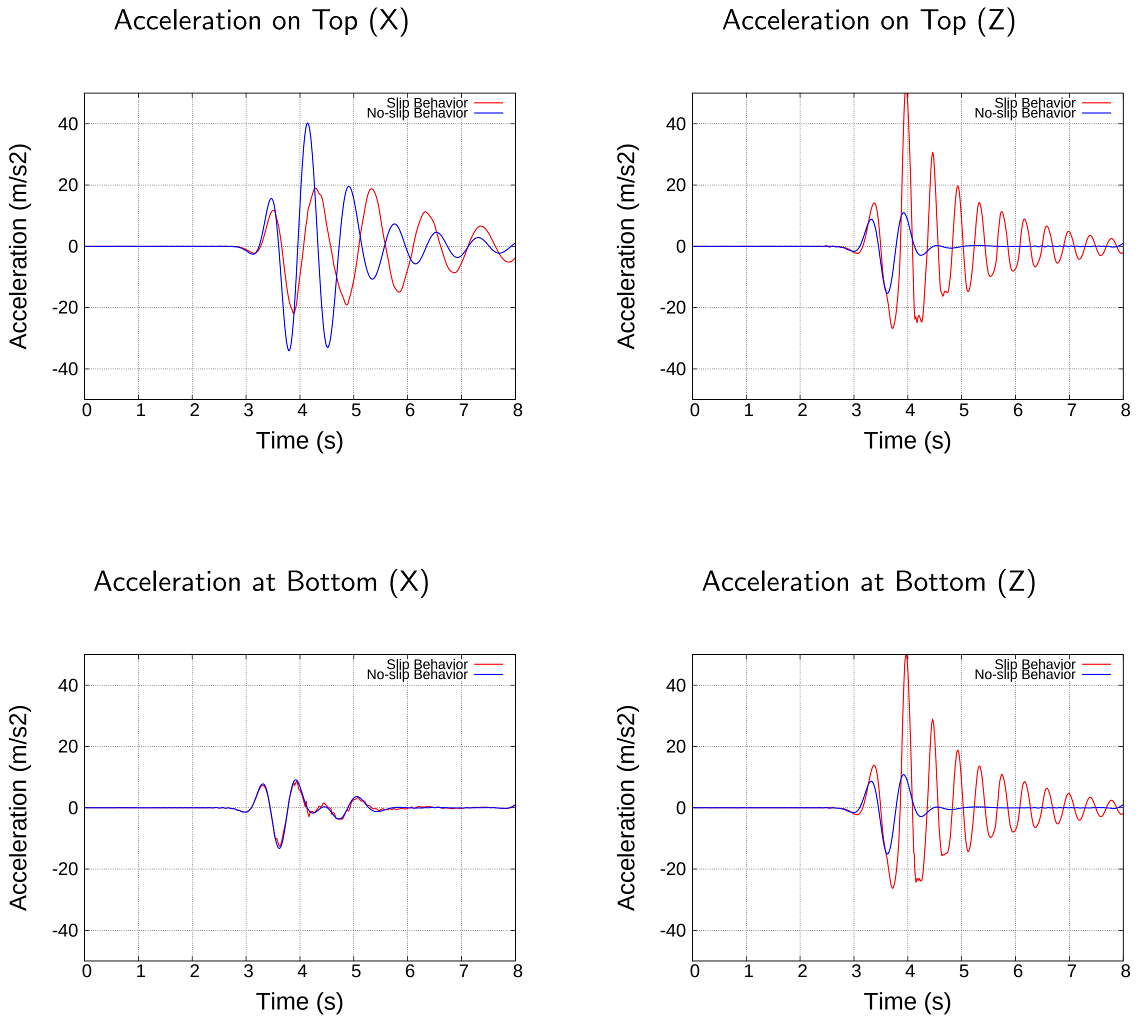
Displacement on Top (X)

Displacement on Top (Z)

Displacement at Bottom (X)

Displacement at Bottom (Z)

Figure 504.16: Comparison of displacement time histories of the structure between slipping and no-slipping models for Ricker wave



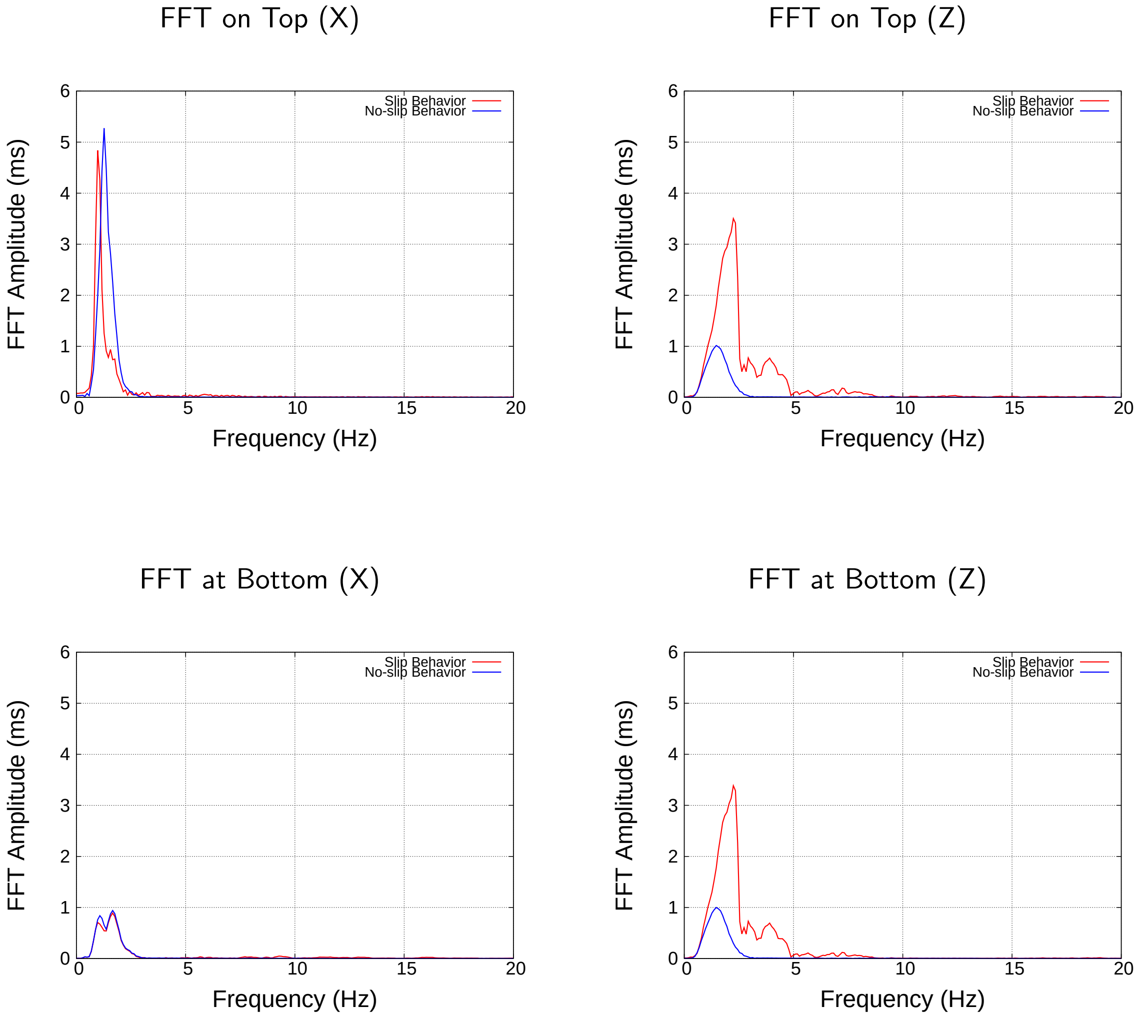
Acceleration on Top (X)

Acceleration on Top (Z)

Acceleration at Bottom (X)

Acceleration at Bottom (Z)

Figure 504.17: Comparison of acceleration time histories of the structure between slipping and no-slipping models for Ricker wave



FFT on Top (X)

FFT on Top (Z)

FFT at Bottom (X)

FFT at Bottom (Z)

Figure 504.18: Comparison of FFT of the acceleration of the structure between slipping and no-slipping models for Ricker wave

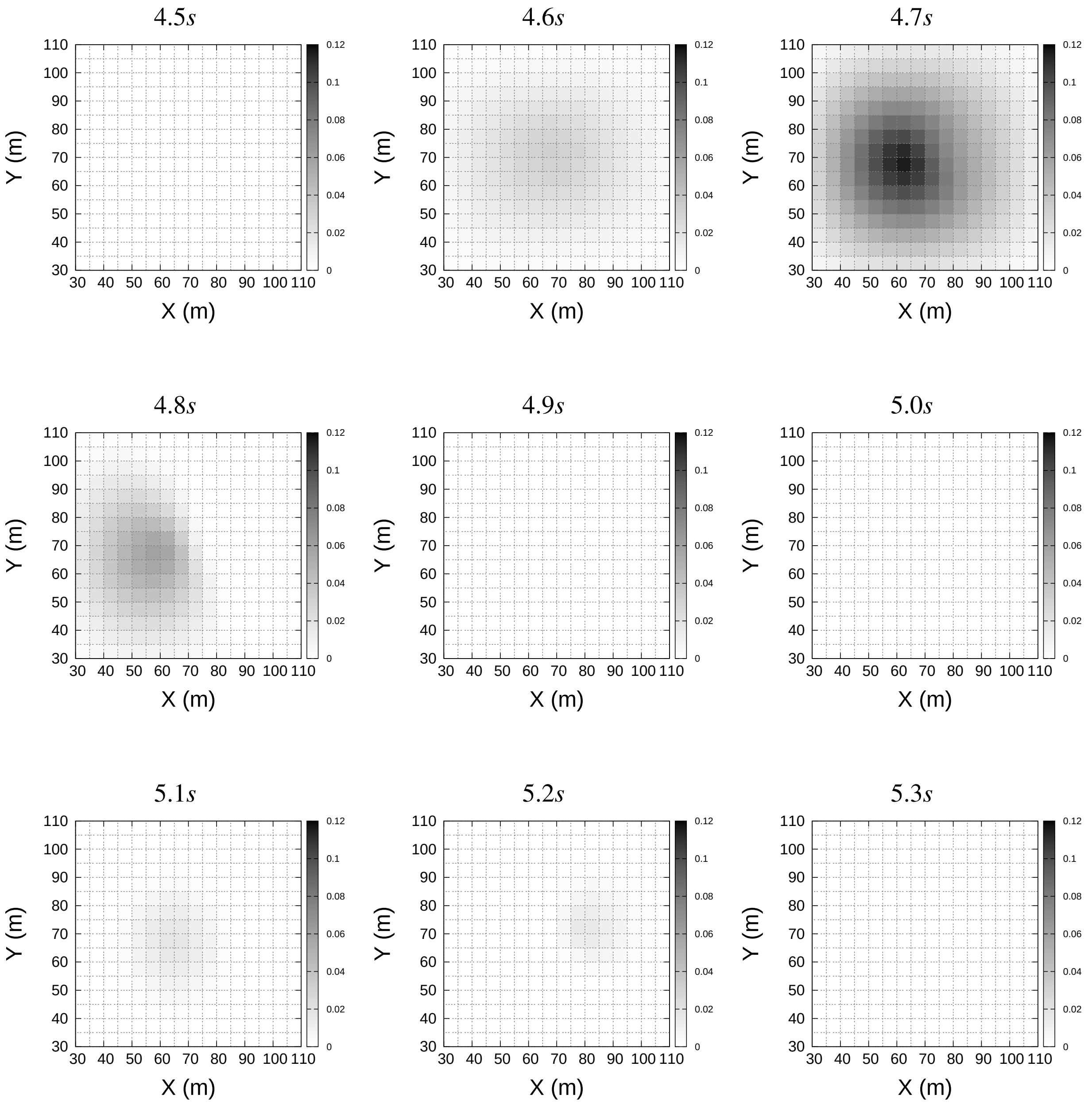


Figure 504.19: Distribution of gap openings along the contact/interface interface for Ricker wave (gray scale given in meters).

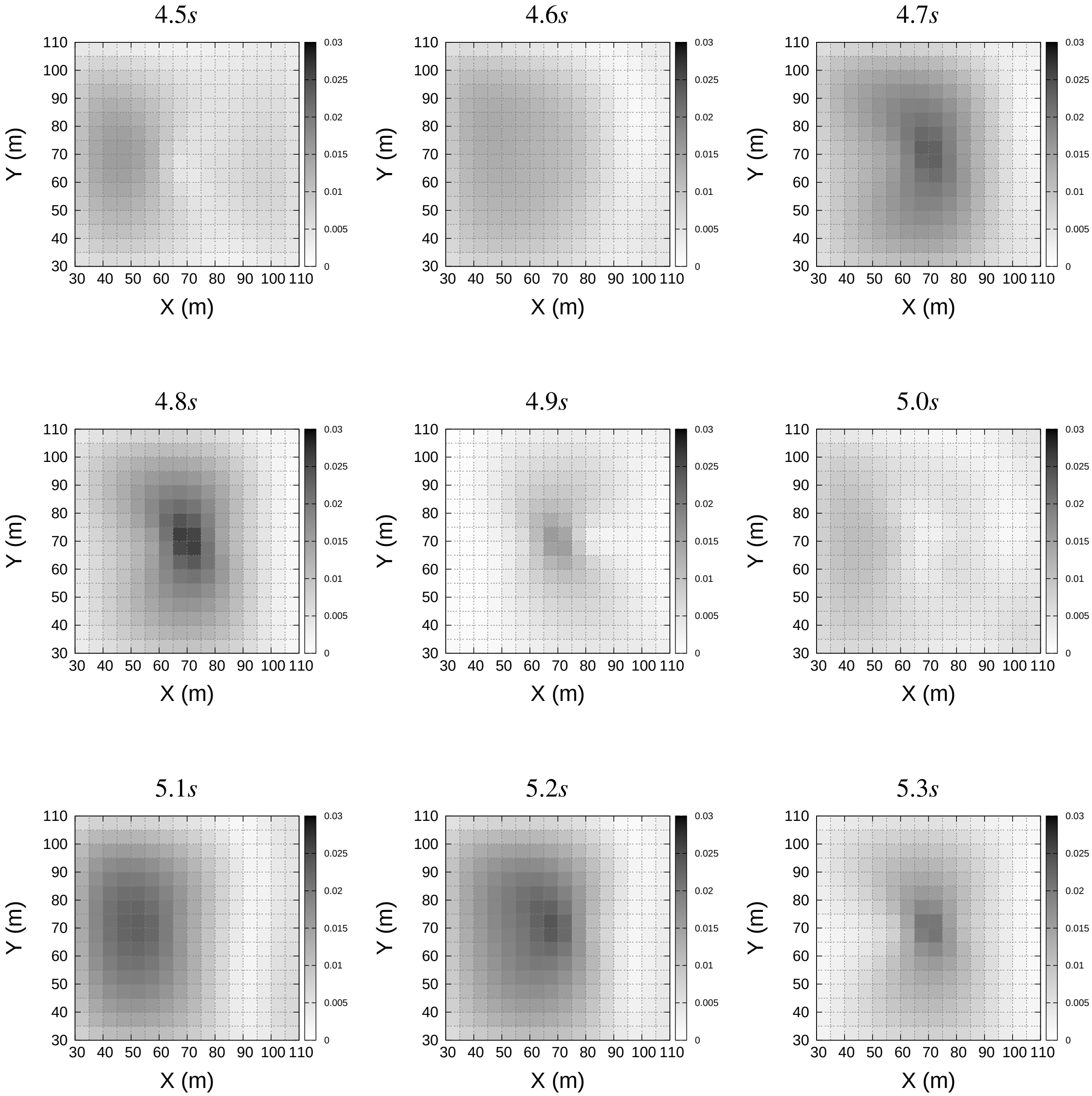


Figure 504.20: Distribution of sliding along the contact/interface for Ricker wave (gray scale given in meters).

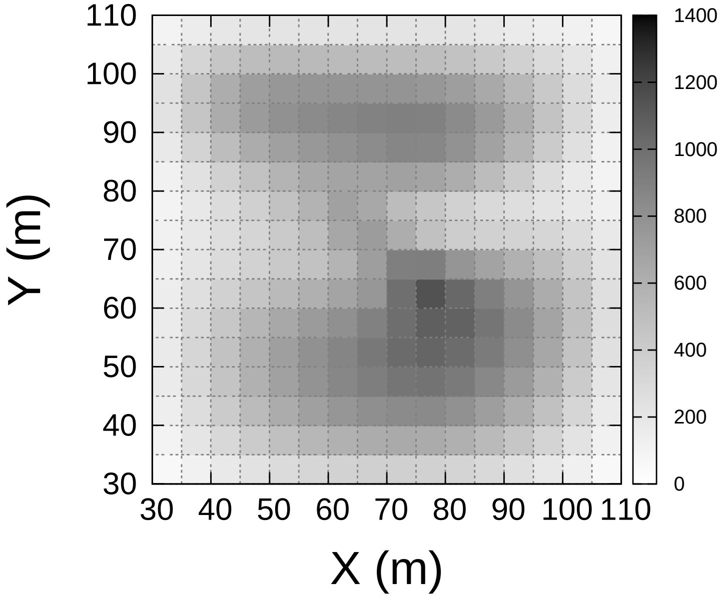


Figure 504.21: Distribution of cumulative dissipated energy due to sliding along the contact/interface for Ricker wave (gray scale given in kJ).

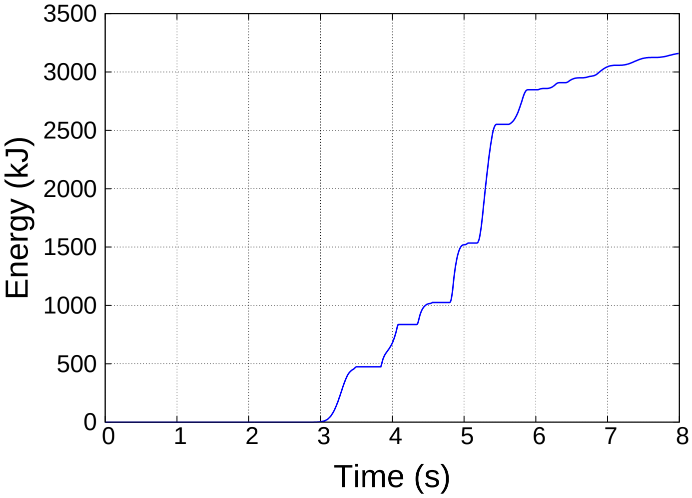


Figure 504.22: Cumulative dissipated energy time history due to sliding at the mid-center of the contact/interface for Ricker wave (gray scale given in kJ).

504.1.2.2 Ricker wave, with fault located at 34° towards the top middle point of the model

The last simulation to be studied here is the case which the location of the fault has an angle of 34° with respect to the top middle point of the model (Figure (504.15)). Same Ricker wave is used as an input motion to be propagated through the model built for first stage of DRM analysis. In this case, since the source of the motion is farther from the interested domain (comparing to previous case with the angle of 45°), motions will arrive later at the interface zone, and structure start shaking later.

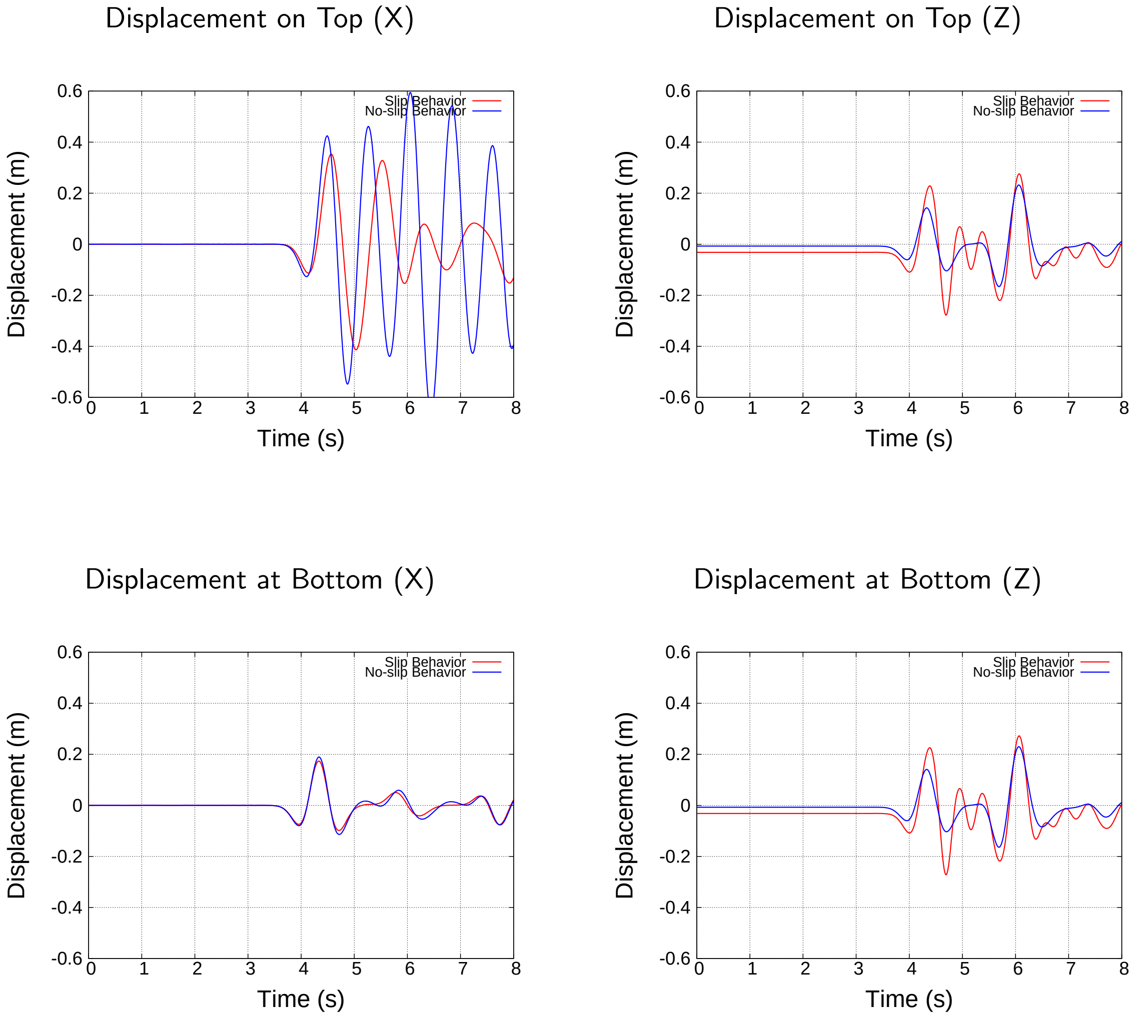
Displacement time histories at the bottom and top of the structure are recorded and shown in Figure (504.23) along X and Z directions. Comparing to the previous case along X direction, at the beginning of the shaking the magnitude of the displacement for slipping behavior model is a bit smaller while it damps out sooner as well. Along the X direction, the trend of the displacement time histories are pretty much the same with a slight difference in magnitudes in time.

Acceleration time histories of the structure are shown in Figure (504.24) along X and Z directions. The same behavior is observed here as previous case such that the amplitude of the acceleration is significantly higher for case of no-slipping along the X direction while it is less along the Z direction. The seismic wave damps out sooner along X direction in case of considering the slipping behavior.

The frequency content change is shown in Figure (504.25) by comparing the FFT of the accelerations obtained from the model considering the slipping behavior and the one with no-slipping behavior. It can be observed that there is a slight change in frequency content and predominant frequencies along both X and Z directions.

Figure (504.26) shows the distribution of gap openings along the soil-foundation contact/interface zone for 9 time steps from 4.5 to 5.3 seconds. The maximum gap opening happens at 4.5 seconds with magnitude of $0.12m$. On the other hand, the sliding at the contact/interface zone is shown in Figure (504.27) with maximum magnitude of $0.0035m$ happening at 4.5 seconds. In this case, the maximum sliding happens around the location of the structure.

Figures (504.28) and (504.29) show the distribution of the dissipated seismic energy at the interface zone and dissipated energy time history at the middle of the contact/interface zone respectively.


Displacement on Top (X)

Displacement on Top (Z)

Displacement at Bottom (X)

Displacement at Bottom (Z)

Figure 504.23: Comparison of acceleration time histories of the structure between slipping and no-slipping models for Ricker wave

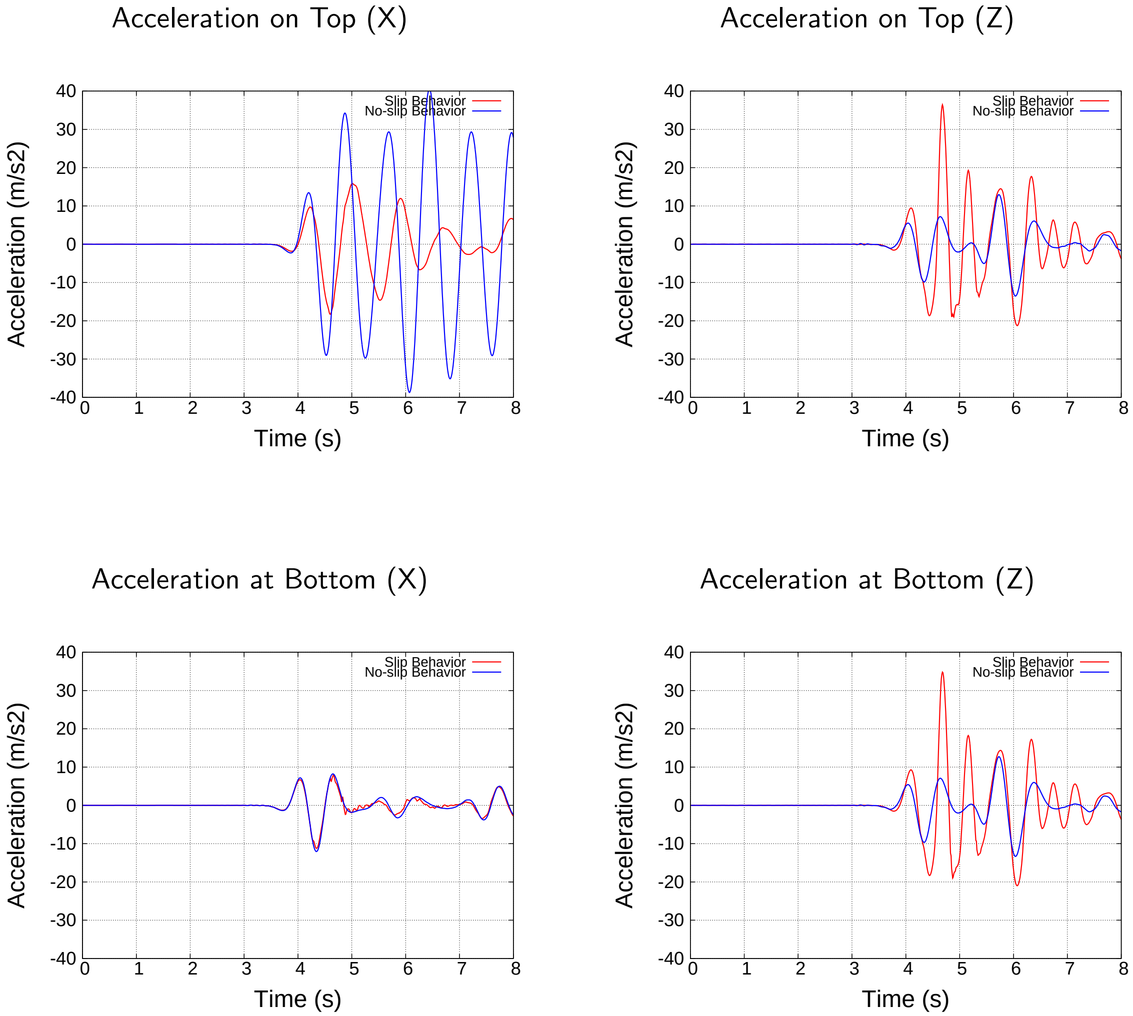


Figure 504.24: Comparison of acceleration time histories of the structure between slipping and no-slipping models for Ricker wave

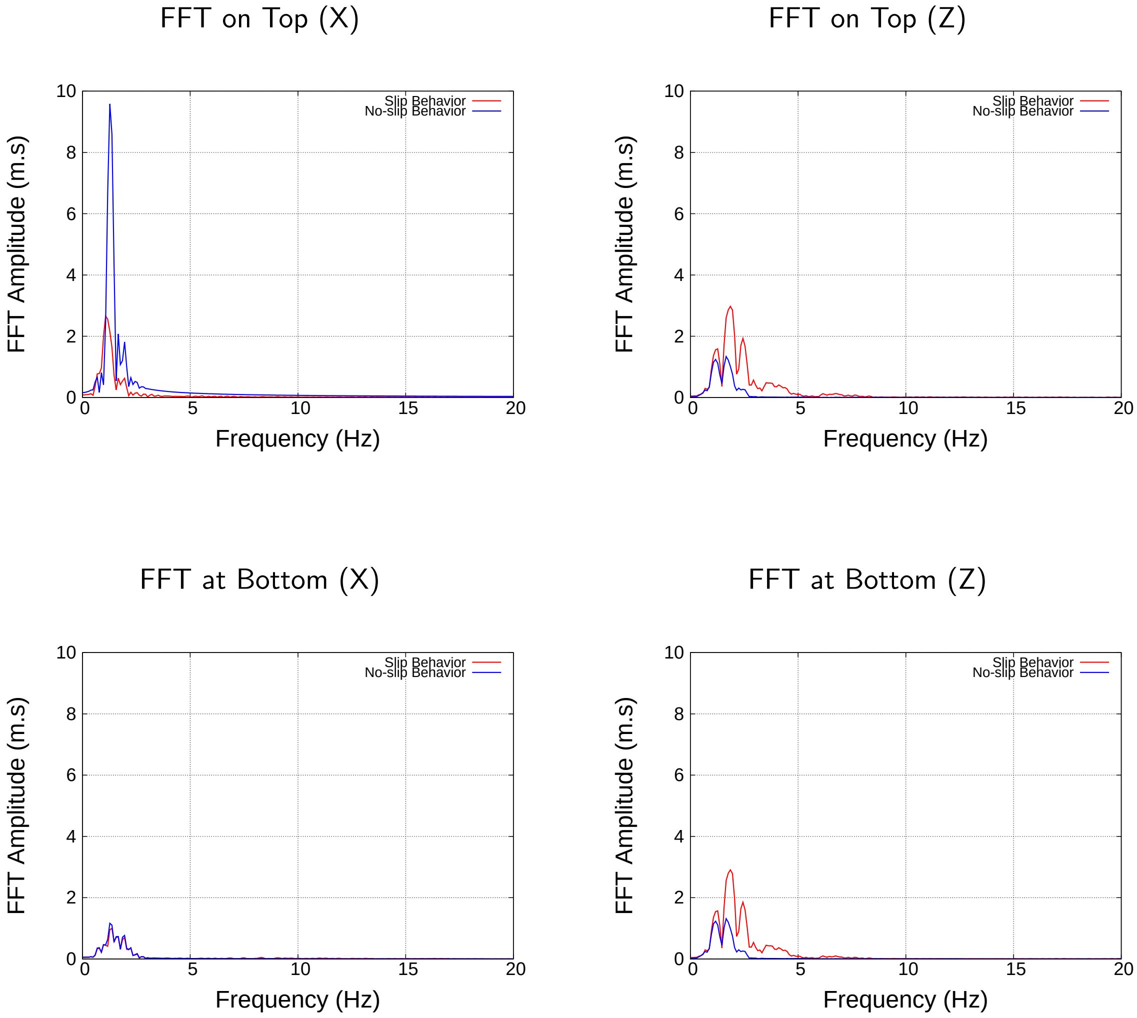


Figure 504.25: Comparison of FFT of the acceleration of the structure between slipping and no-slipping models for Ricker wave

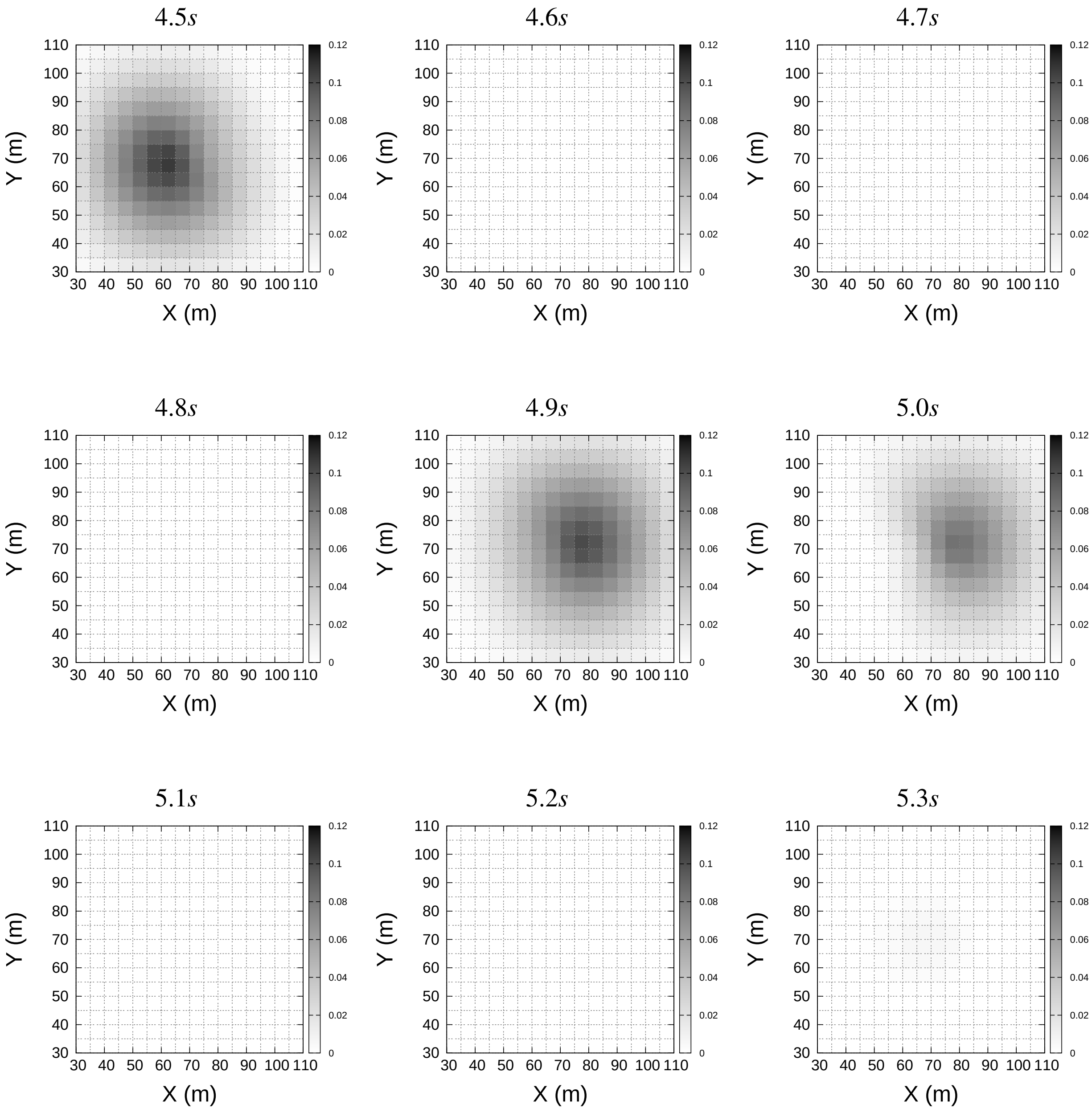


Figure 504.26: Distribution of gap openings along the contact/interface for Ricker wave (gray scale given in meters)

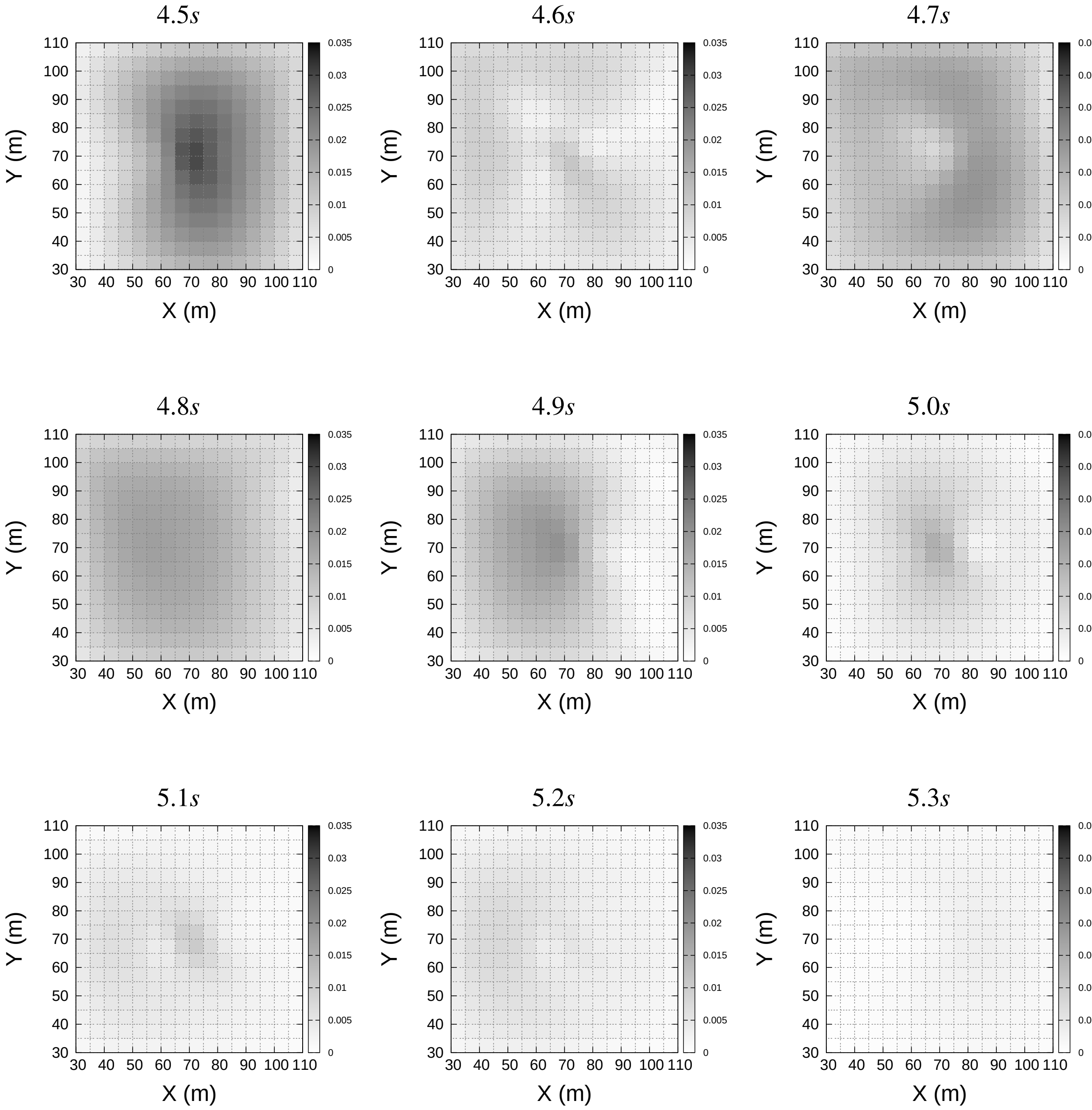


Figure 504.27: Distribution of sliding along the contact/interface for Ricker wave (gray scale given in meters)

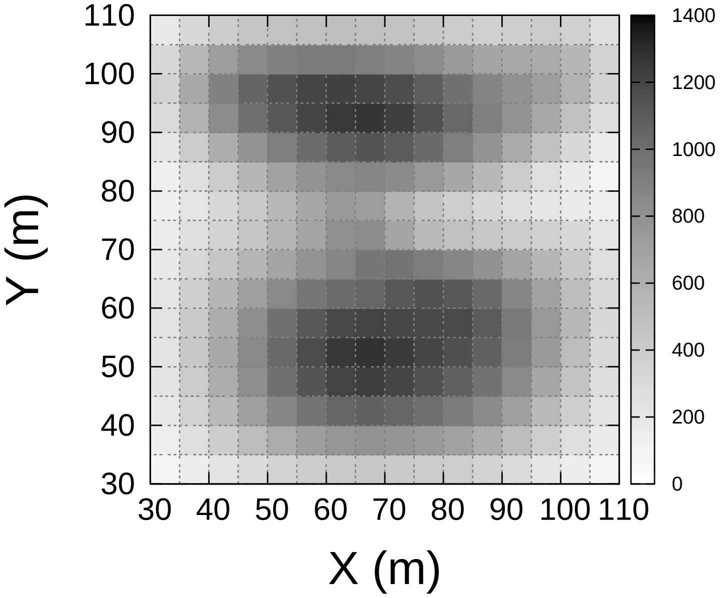


Figure 504.28: Distribution of cumulative dissipated energy due to sliding along the contact/interface for Ricker wave (gray scale given in kJ)

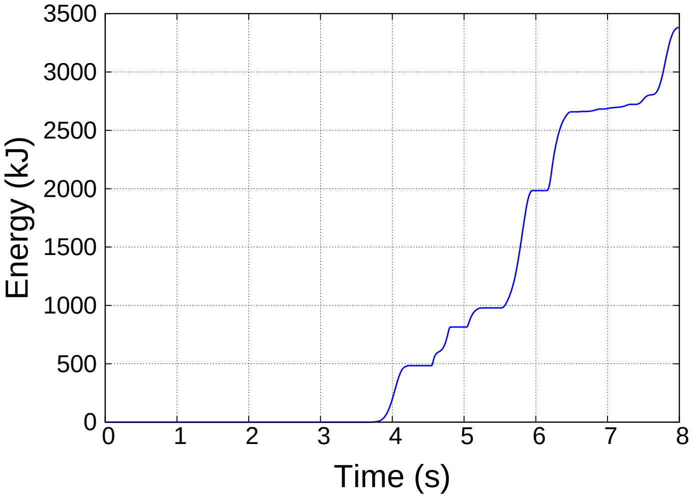


Figure 504.29: Cumulative dissipated energy time history due to sliding at the mid-center of the contact/interface for Ricker wave (gray scale given in kJ)

504.2 Three Dimensional (3C) Seismic Wave Fields and Behavior of Nuclear Power Plants (NPPs)

504.2.1 Development of Seismic Motions: Large Scale Free Field Model

- Large scale seismic free field
- Close up for large scale seismic free field

504.2.2 NPP Response, Model #01

- Free field at NPP location
- NPP response
- NPP response, cut-out of the model, inside response

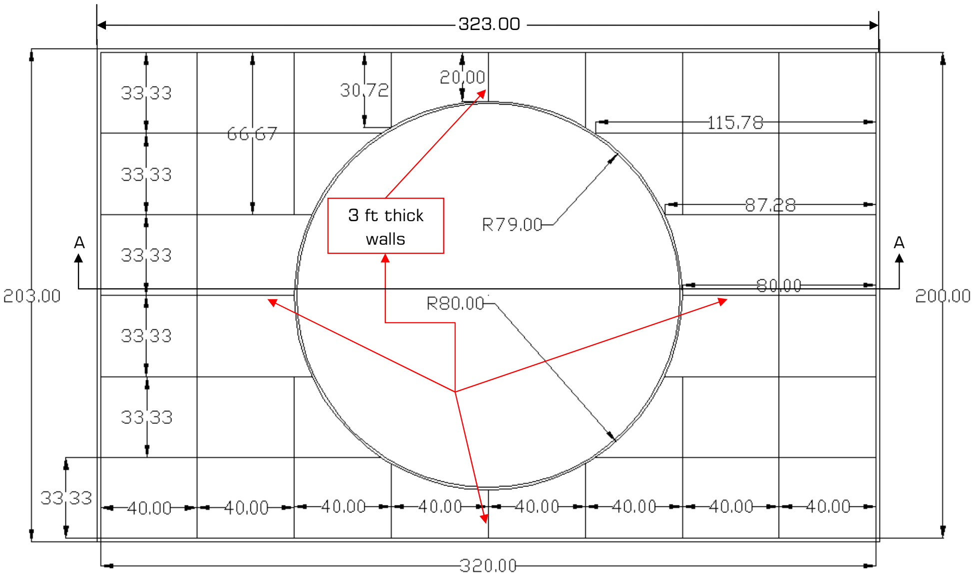
504.3 3D Representative NPP Structure Model(s)

504.3.1 Model #01, Single NPP

Total number degrees of freedom (DOFs, unknowns) for Model 01 is 681,648. Table 504.3.1, gives other basic statistics for this model, while Figures 504.3.1 to 504.3.1 show disposition and views of the finite element mesh.

Table 504.1: Model 01 Statistics

Components	Number of
3 DOF Nodes	205875
6 DOF Nodes	14413
27 node Bricks	23916
ANDES Shells	15627
Contact elements	2124
9 DOF Beams	583



PLAN VIEW (distances given in ft)

Notes for Auxiliary Building:

- Interior walls except those highlighted above have a thickness = 2 ft.
- Exterior walls have a thickness of 3 ft.

Figure 504.30: Model01: Plan view (dimensions in feet).

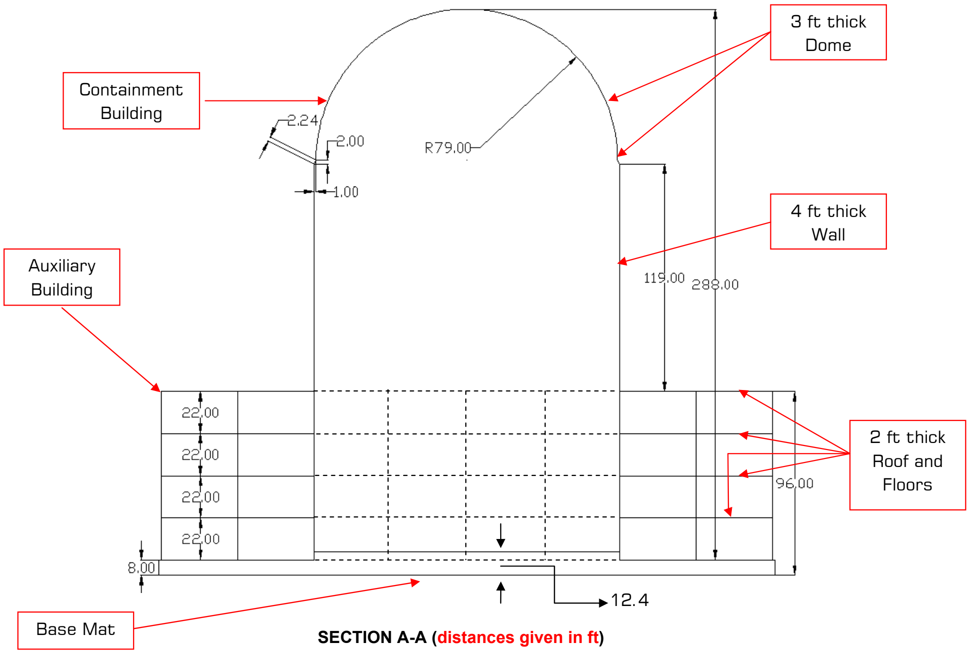


Figure 504.31: Model01: Section view (dimensions in feet).

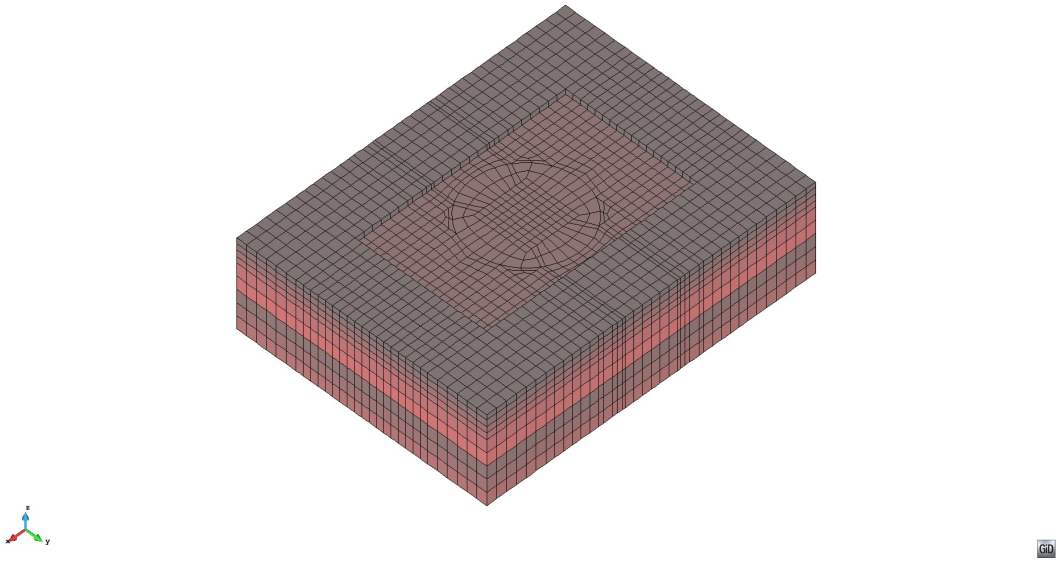


Figure 504.32: Model01: Soil layers.

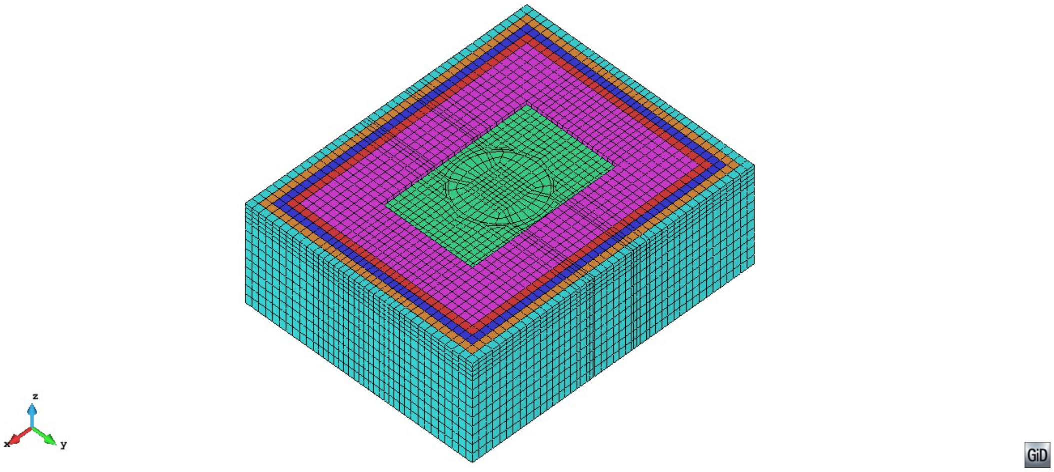


Figure 504.33: Model01: Soil with the DRM (no slab).

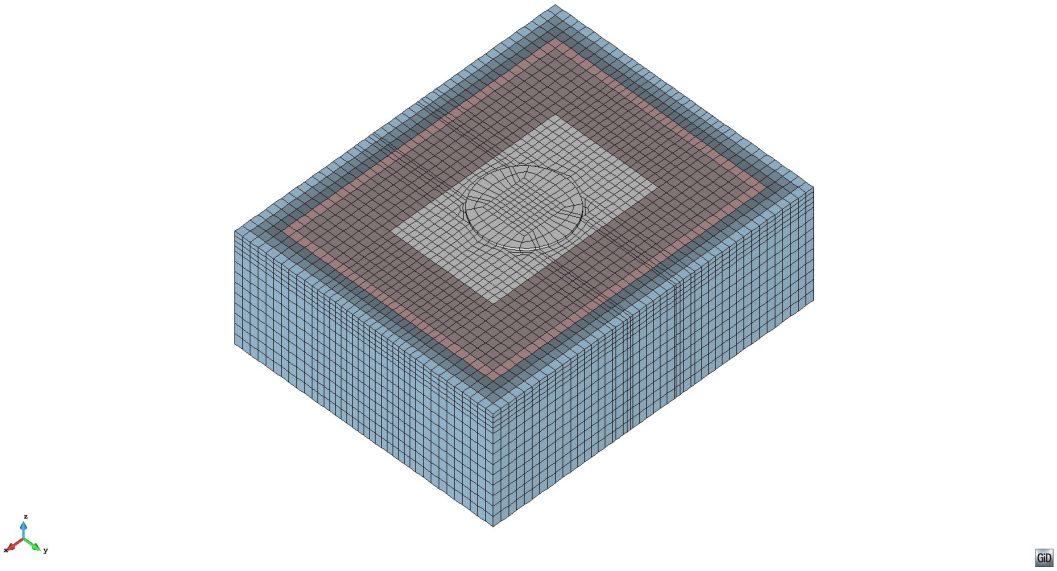


Figure 504.34: Model01: Soil, with the slab and the DRM.

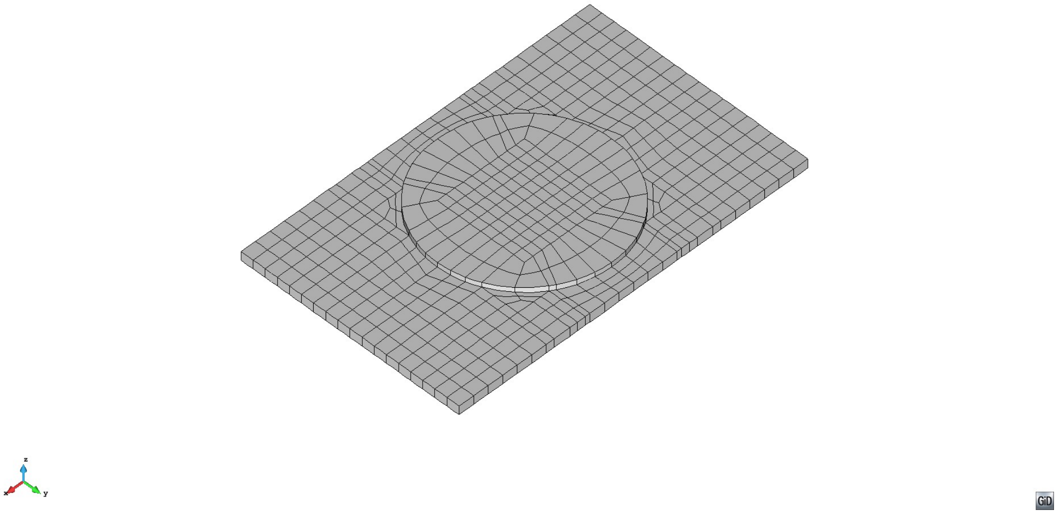


Figure 504.35: Model01: Slab only.

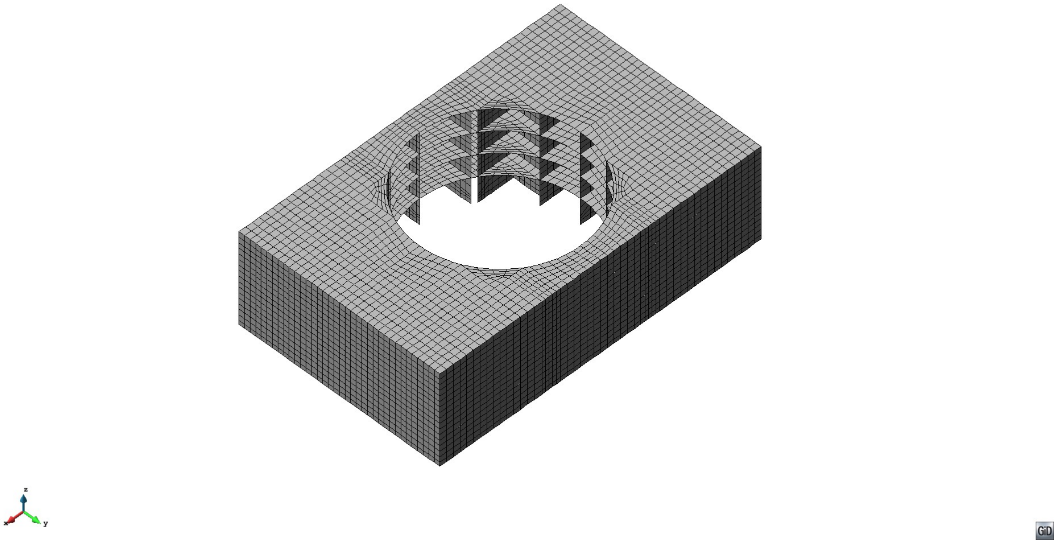


Figure 504.36: Model01: Auxillary building

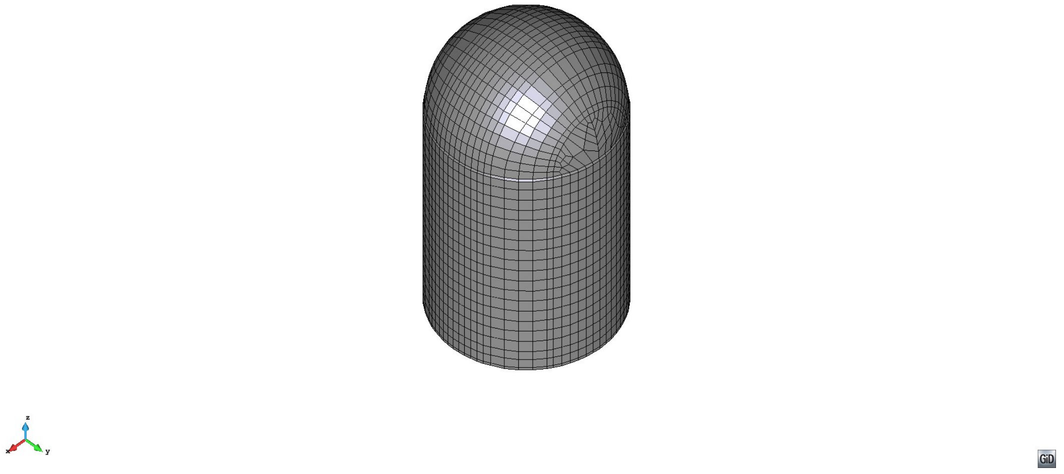


Figure 504.37: Model01: Containment building.

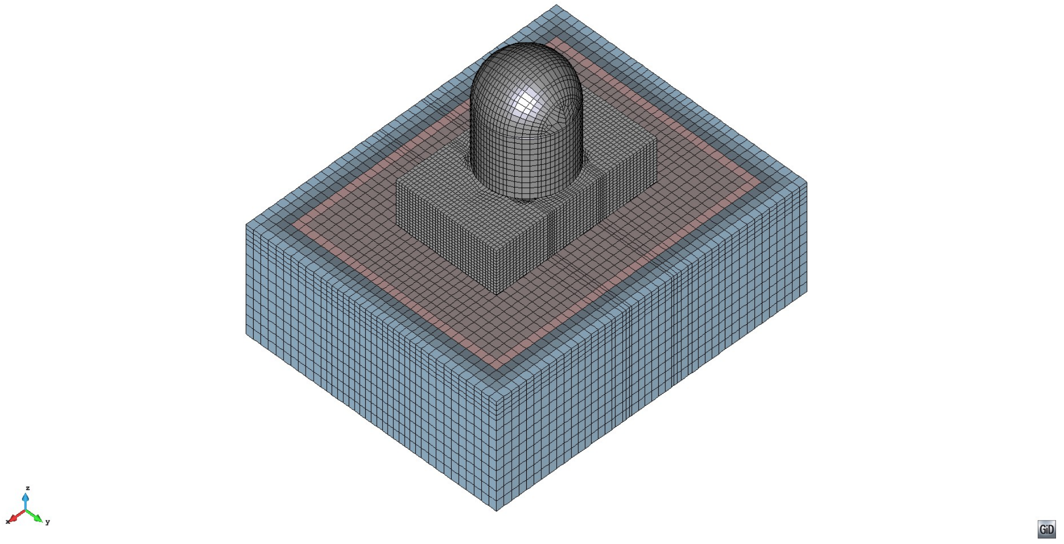


Figure 504.38: Model01: Full view.

504.3.2 Model #02, Single NPP

Simplified Representative 3D Model. Total number degrees of freedom (DOFs, unknowns) for Model 02 is 260,883. Table 504.2, gives other basic statistics for this model, while Figures 504.3.1 to 504.3.2 show disposition and views of the finite element mesh.

Table 504.2: Model 02 Statistics

Components	Number of
3 DOF Nodes	84820
6 DOF Nodes	3263
27 node Bricks	9576
ANDES Shells	8384
Contact elements	1249
9 Node Beams	490

Figure 504.3.2 shows a general disposition of a representative 3D NPP model #02.

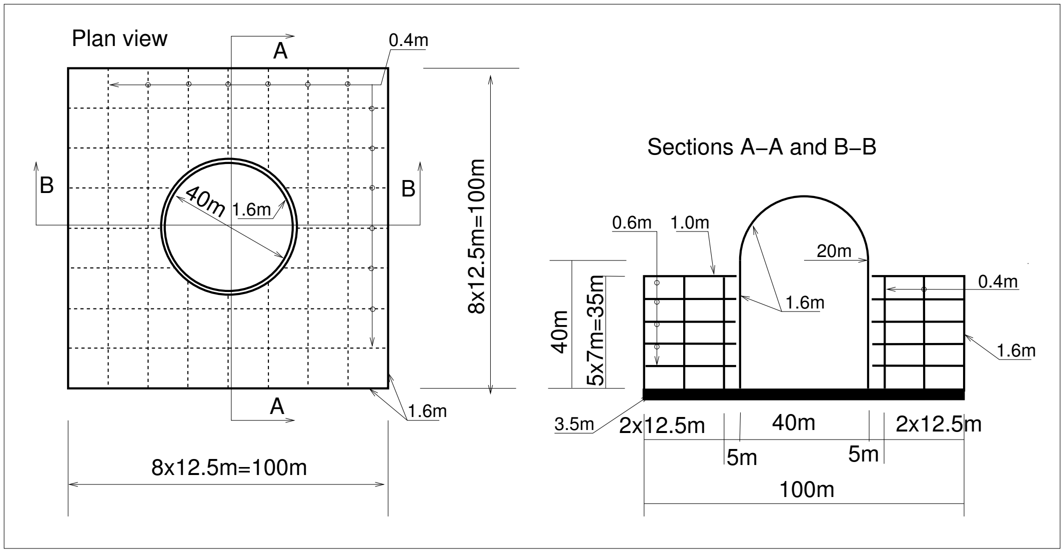


Figure 504.39: General disposition of a representative 3D NPP model (model #02).

The model was developed as a cylindrical containment with dome top, while the auxiliary building surrounds the containment. There is no contact/interface between containment and the auxiliary building and current model has a gap of 0.2m while a new model (currently under development) will reduce this gap space, while still maintaining independence of two structural systems.

Finite element mesh for components and the complete system is shown in Figures 504.3.2 to 504.3.2.

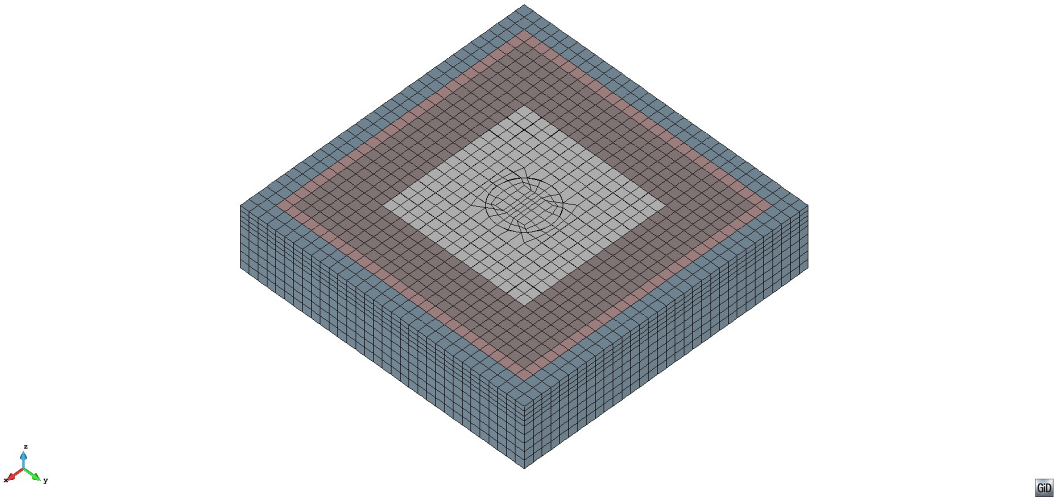


Figure 504.40: Model02: Soil with the fondation slab and the DRM layers.

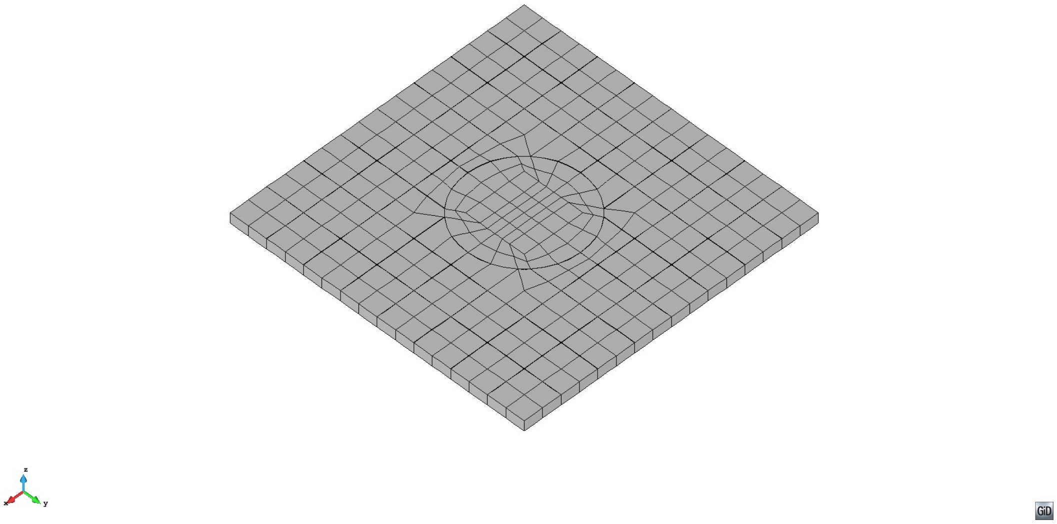


Figure 504.41: Model02: Foundation slab only.

Both structures (containment and auxiliary building) were placed on a slab foundation and then placed on a soil/rock base. Model is flexible enough that soil/rock properties and geology can be

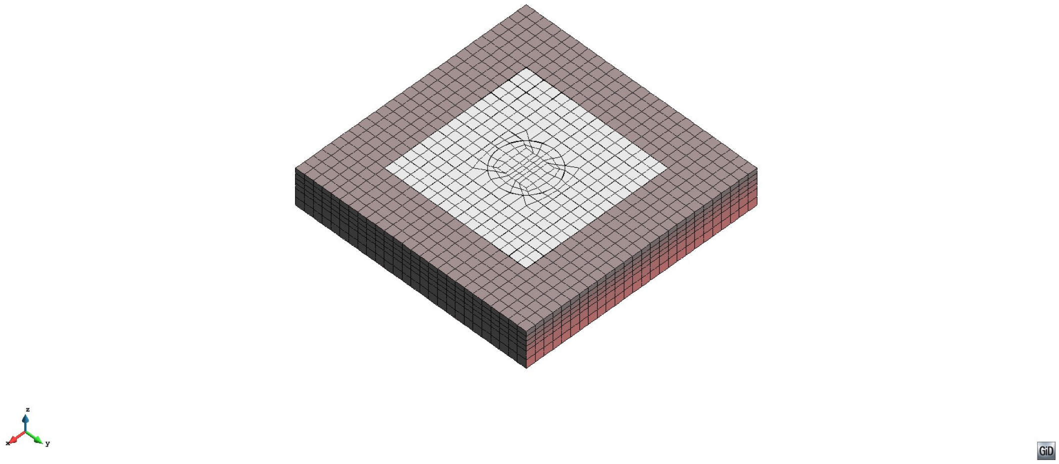


Figure 504.42: Model02: Layered soil.

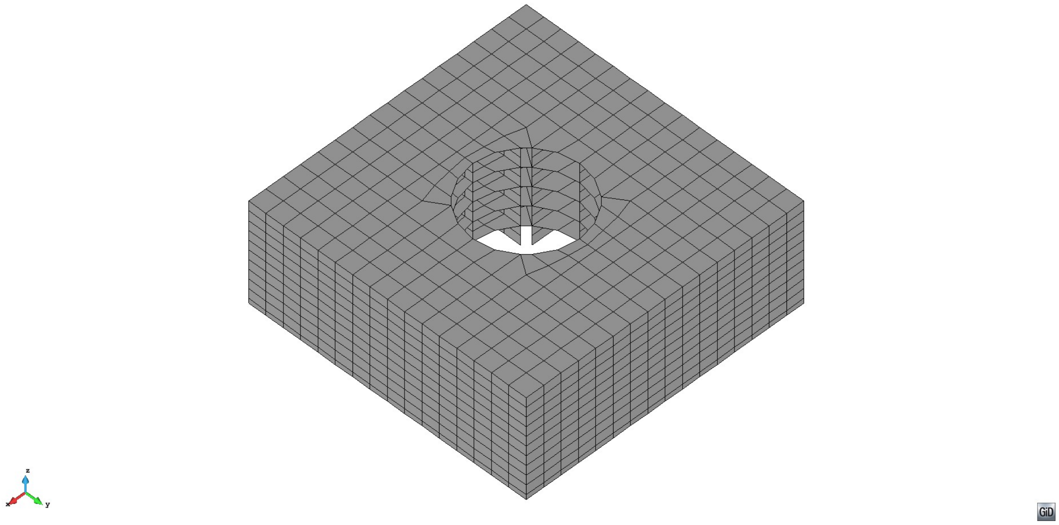


Figure 504.43: Model02: Auxillary building.

varied. The sub-base model also includes a layer of elements for the DRM motion input, as well as two layers of elements outside the DRM layer for damping any outgoing waves. Models with a single NPP (Figure 504.3.2) is developed and used in analysis.

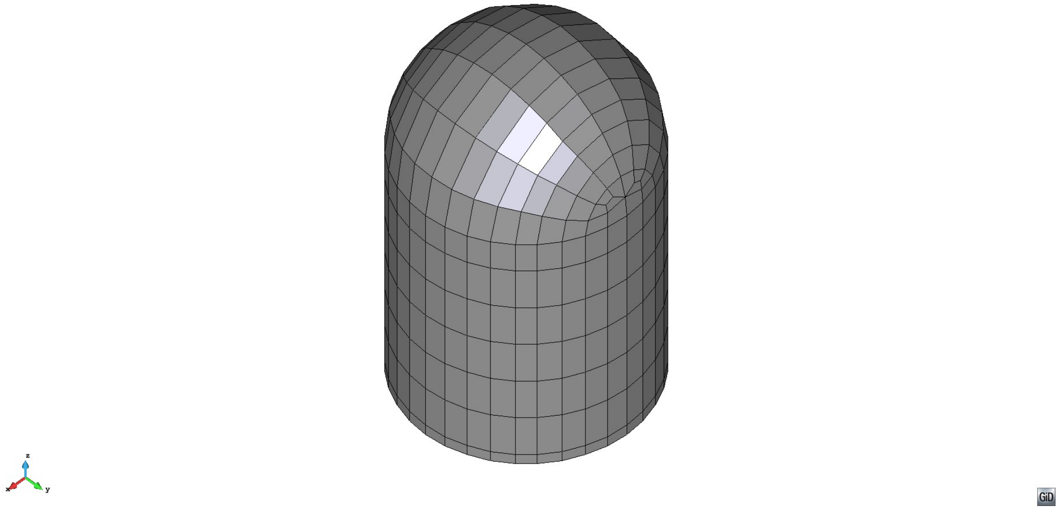


Figure 504.44: Model02: Containment building.

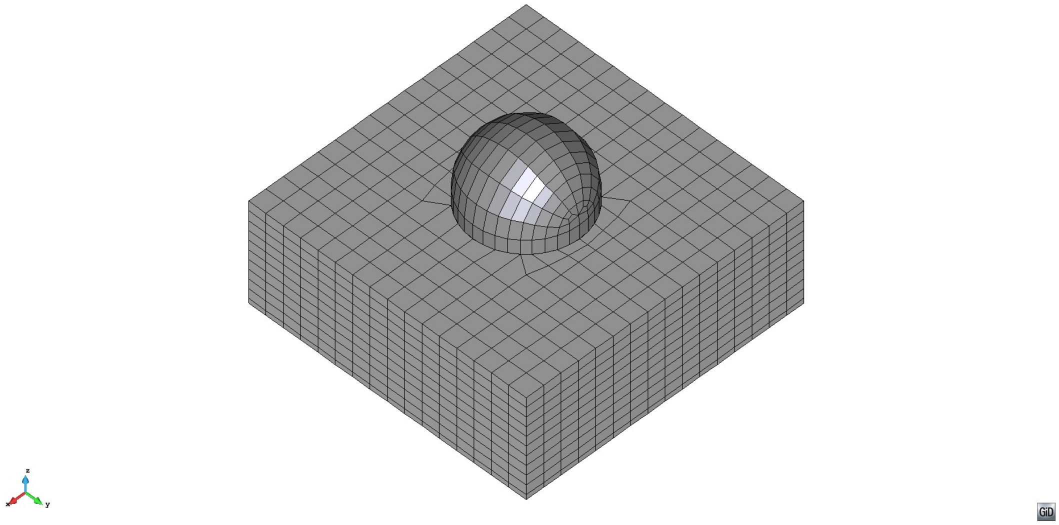


Figure 504.45: Model02: Auxillary and containment buildings.

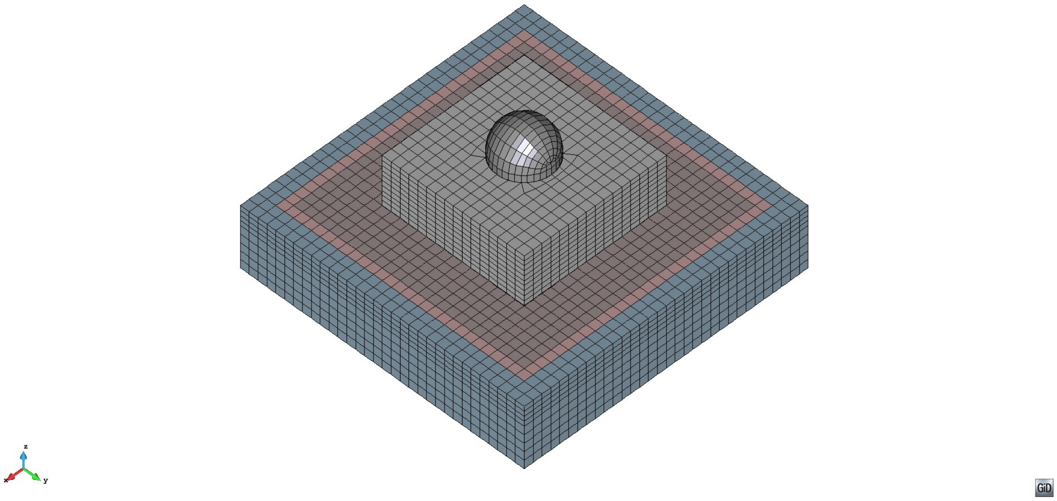


Figure 504.46: Model02: Full view, Finite element model for a single NPP with the containment and auxiliary buildings on a common base mat, as well as the soil/rock sub-base, DRM layer for seismic motions input and the layers outside of DRM for damping out outgoing waves.

504.3.3 Model #03, Double NPP, Soil-Structure-Soil-Structure Intaraction

Total number degrees of freedom (DOFs, unknowns) for Model 03 is 518,472. Table 504.3, gives other basic statistics for this model, while Figures 504.3.3 to 504.3.3 show finite element mesh.

Table 504.3: Model III Basic Statistics

Components	Number of
3 DOF Nodes	168405
6 DOF Nodes	6526
27 node Bricks	19152
ANDES Shells	16768
Contact elements	2498
9 Node Beams	950

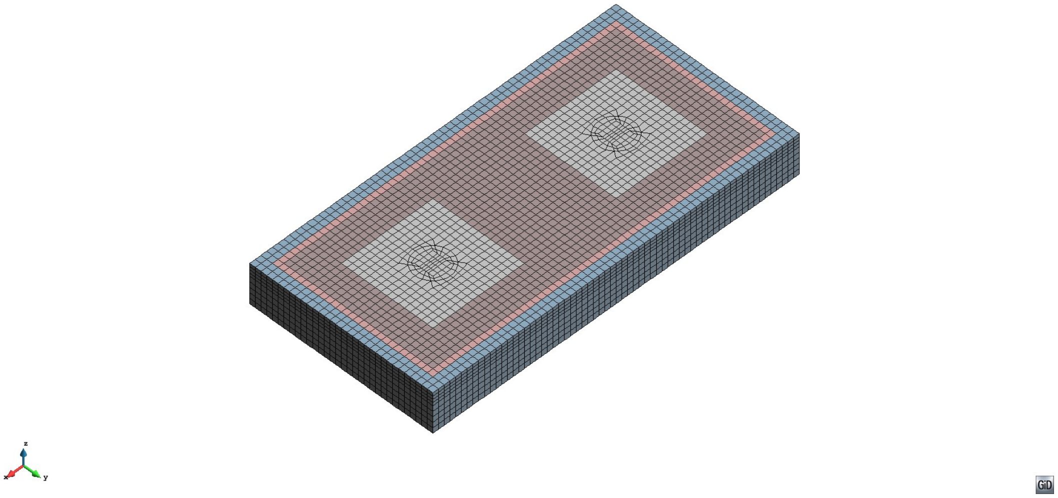


Figure 504.47: Model03: Soil, foundation slabs and the DRM layers.

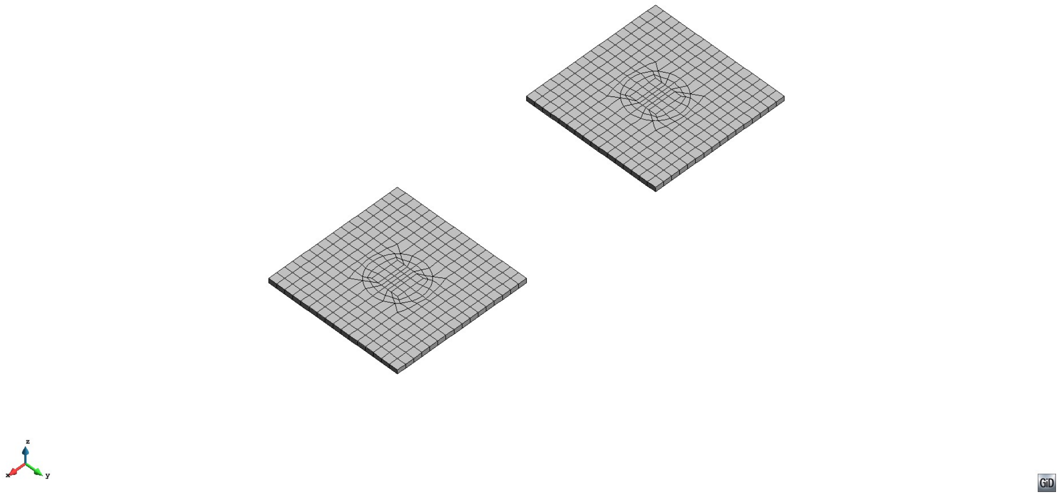


Figure 504.48: Model03: Foundation slabs only.

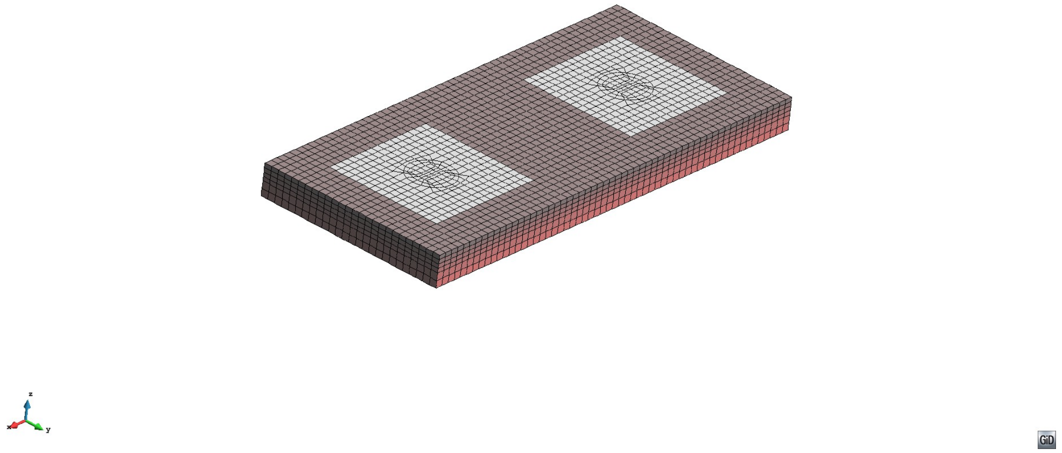


Figure 504.49: Model03: Layered soil.

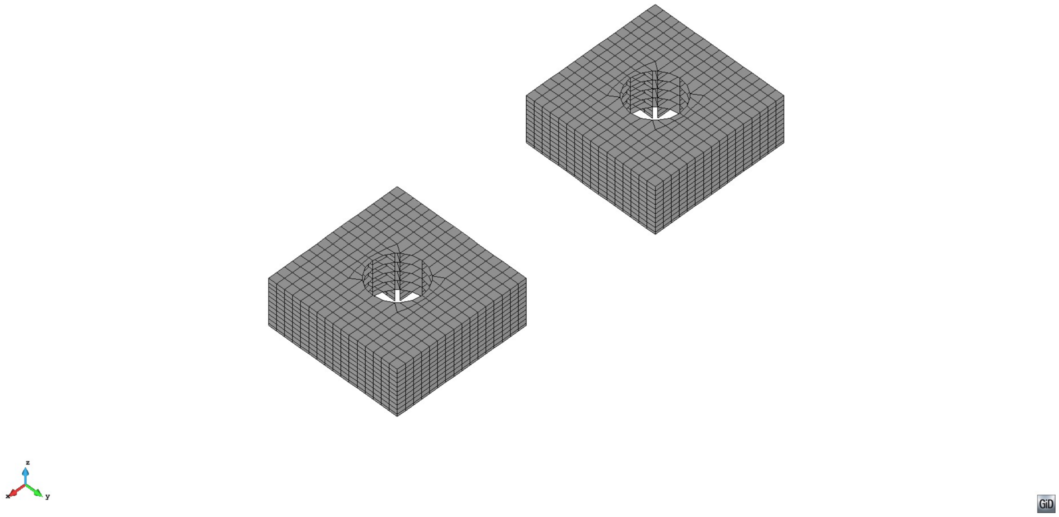


Figure 504.50: Model03: Auxillary buildings.

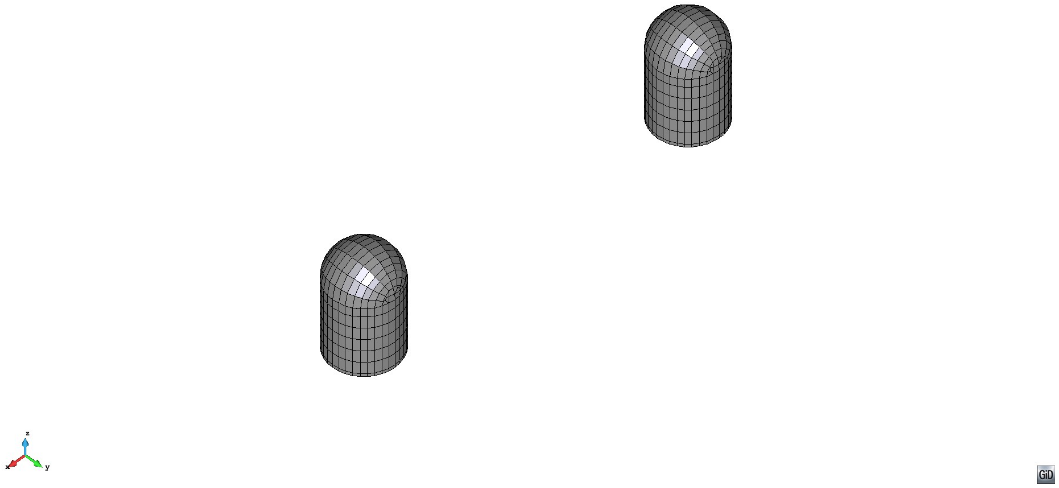


Figure 504.51: Model03: Containment buildings.

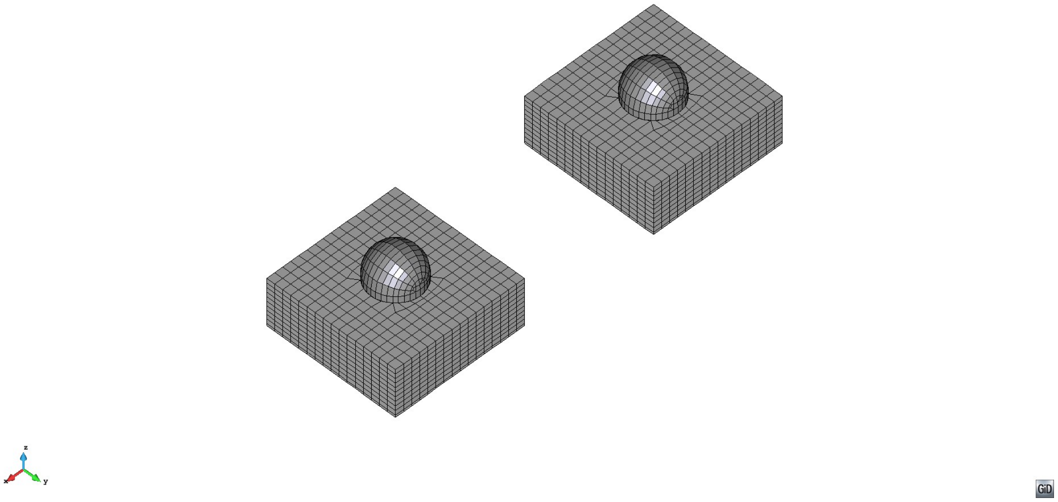


Figure 504.52: Model03: Auxillary and containment buildings.

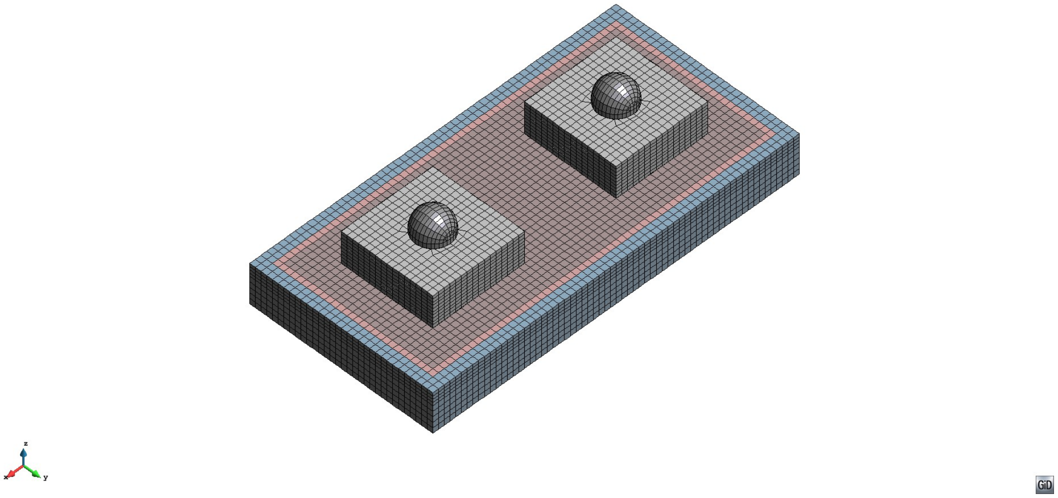


Figure 504.53: Model03: Full view.

Soil-Structure-Soil-Structure Interaction. Soil-Structure-Soil-Structure Interaction (SSSSI) need to be taken into account sometimes, as it might contribute to higher levels of seismic shaking for NPPs. It seems that in the case of making an assumption of elastic soil and rock beneath the NPP foundation, the SSSSI will have a larger effect most of the time, while with the inclusion of elastic-plastic models for soil and rock under the NPP foundation, those SSSSI influences will be reduced most of the time. This is illustrated in one of the models that was developed for analyzing SSSSI. While SSSSI for two NPPs (of similar stiffness and mass) conforms to the above observations, it is noted that SSSSI influences of smaller structures on larger (NPPs) can be mostly neglected, the SSSSI influence of larger structures (NPPs) on smaller structures Probably cannot be neglected.

There are a number of ways to model SSSSI.

- Direct Models. The simplest and most accurate is to develop a direct model of both (two or more) structures on subsurface soil and rock, to develop input seismic motions and analyze results. While this approach is the most involved, it is also the most accurate, as it allows for proper modeling of all the structure, foundation and soil/rock geometries and material without making any unnecessary simplifying assumptions.

- Symmetry and Anti-Symmetry Models. These models are sometimes used in order to reduce complexity and sophistication of the direct model (see recent paper by Roy et al. (2013) for example). However, there are a number of concerns regarding simplifying assumptions that need to be made in order for these models to work. These models have to make an assumption of a vertically propagating shear waves and as such do not take into account surface waves (Rayleigh, Love, etc) that carry significant amount of seismic energy. These surface waves will additionally excite NPP for rocking and twisting motions, which will then be transferred to adjacent NPP by means of additional surface waves. If only vertically propagating waves are used for input (as is the case for symmetry and anti-symmetry models) energy of input surface waves is neglected. It is noted that depending on the surface wave length and the distance between adjacent structures, a simple analysis can be performed to determine if particular surface waves, emitted/radiated from one structure toward the other one (and in the opposite direction) can influence adjacent structures. It is noted that the wave length can be determined using a classical equation $\lambda = v/f$ where λ is the length of the (surface) wave, v is the wave speed[1] and f is the wave frequency of interest. Table504.4 below gives Rayleigh wave lengths for four different wave frequencies $(1, 5, 10, 20$ Hz and for three different Rayleigh (very close to shear) wave velocities $(300, 1000, 2500$ m/s):

---

[1] For Rayleigh surface waves, their speed is just slightly below the shear wave speed (within 10%, depending on elastic properties of material), so a shear wave speed can be used for making these Rayleigh wave length estimates.

Table 504.4: Rayleigh wave length as a function of wave speed [m/s] and wave frequency [Hz].

	1.0Hz	5.0Hz	10.0Hz	20Hz
300m/s	300m	60m	30m	15m
1000m/s	1000m	200m	100m	50m
2500m/s	2500m	500m	250m	125m

It is apparent that for given separation between NPP buildings, different surface wave (frequencies) will be differently transmitted with different effects. For example, for an NPP building that has a basic linear dimension (length along the main rocking direction) of 100m, the surface wave the low frequency waves (1Hz) in soft soil ($v_s \approx 300$m/s) will be able to encompass a complete building within a single wave length, while for the same soil stiffness, the high frequency (20Hz) will produce waves that are too short to efficiently propagate through such NPP structure. On the other hand, for higher rock stiffness ($v_s \approx 2500$m/s), waves with frequencies all the way up to approximately 5Hz and maybe even 10Hz, have an extent that can easily be affecting a building with a 100m dimension.

Further comments on symmetric and antisymmetric models:

- Symmetry: motions of two NPPs are out phase and this represents an unrealistic case, unless the wave length of surface wave created by one NPP (toward the other NPP) is so large that half wave length will encompass both NPPs. This type of motions (symmetry) is illustrated in figure 504.3.3 below

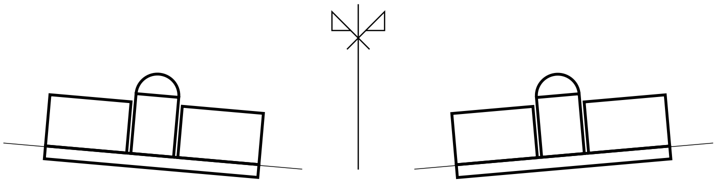


Figure 504.54: Symmetric mode of deformation for two NPPs near each other.

- Antisymmetry: motions of two NPPs are in phase and while that is more realistic than the symmetry case, still requires perfect matching of 1C input motions and the soil/rock conditions beneath, and as such is not realistic. This type of SSSSI is illustrated in figure 504.3.3 below

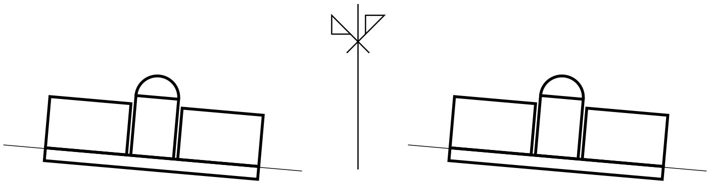


Figure 504.55: Anti-symmetric mode of deformation for two NPPs near each other.

504.3.4 Model #04, Small Modular Reactor (SMR)

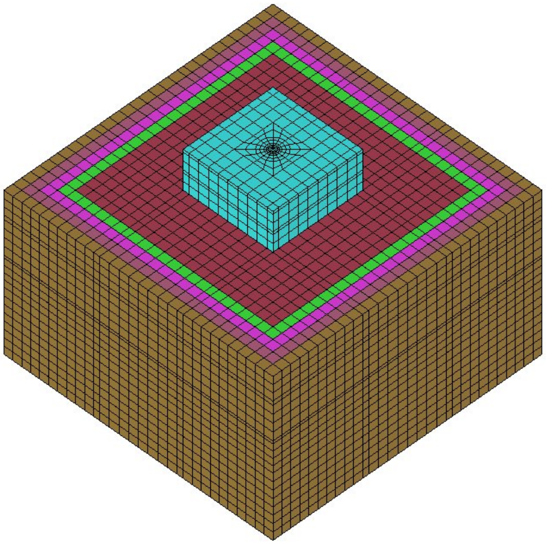


Figure 504.56: Small Modular Reactor model, top (surface) view.

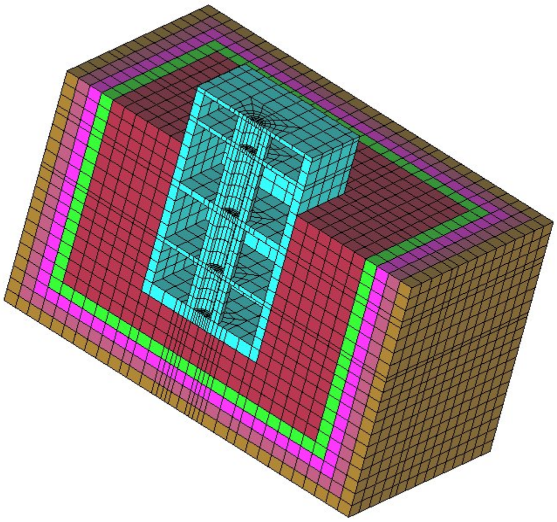


Figure 504.57: Small Modular Reactor model, cross section (half model) view.

504.4 3C (6C) vs 1C Seismic Motions

Realistic seismic wave fields are fully three dimensional, consisting of a body and surface waves, and featuring translational and rotational components of motions at and near the surface. However, current focus of various seismic analysis standards is on 1C motions. These 1C motions are representing one component of a full 3C wave field. More recently, use of 3×1C wave fields have been advocated, where 1C wave motions are replicated in other horizontal and a vertical direction with certain scaling factors. Use of 3×1C motions makes an implicit assumption that vertical motions are resulting from 1C compressional waves. This assumption is usually not true, as most of the vertical motions are a result of surface wave motions.

504.4.1 Appropriate Use of 3C and 3×1C and 1C Seismic Motions

We start by pointing out one of the biggest simplifying assumptions made, is that of a presence and use of 1C seismic waves. As pointed out in section 4.2.1 above, worldwide records do not show evidence of 1C seismic waves. It must be noted, that an assumption of neglecting full 3C seismic wave field and replacing it with a 1C wave field can sometimes be appropriate. However, such assumption should be carefully made, taking into account possible intended and unintended consequences.

A brief discussion on 1C, 3×1C and 3C seismic wave modelling and effects on SSI is provided below:

- 1C modelling of seismic waves is possible if material modelling for soil is linear or equivalent linear elastic. In this case, 1C motions from different directions (horizontal) can be combined, as superposition principle applies for linear elastic systems (soil in this case). Modeling of vertical motions using 1C approach is abit different as an analysis needs to be performed to decide if the vertical wave is a compressional wave (primary, P wave) or if vertical motions are a consequence of vertical components of surface waves. More on those options is provided below in 3×1C modelling option.

- 3×1C modeling of seismic waves is possible, similarly to the above case, if soil material is linear or equivalent linear elastic. SAs noted above, superposition principle can be applied and motions from each direction can be superimposed to obtained 3C motions at the surface. Since most of the time vertical motions are a results (consequence) of Rayleigh surface waves, it is important to analyze vertical motions and decide if modelling motions as 1C is appropriate. To this end, a wave length of surface wave plays an important role. If the Rayleigh surface wave length (which features both horizontal and vertical components) is longer than 12 times the dimension of the object (NPP), than object rotations, due to differential vertical displacements at object ends, are indeed fairly small and object does move up and down as if excited with a vertical wave. This is shown in Figure 504.58 as the upper case. On the other hand, if the wave is long less than 12 object dimensions, then vertical motions are gradually replaced by object rotations, while vertical motions are reduced. Case in the lower left corner of Figure 504.58 shows a limiting case where seismic wave is 4 times longer than object dimension, which results in minimal vertical motions of the object, and maximum rotations, due to differential motions of object ends. For shorter surface waves, as shown in Figure 504.58, lower right case, waves might not even be exciting any significant dynamic behavior of the object (except local deformation) as their wave lengths are shorter than twice object length.

- 3C modelling, when done properly will capture all the body and surface wave effects for SSI analysis of NPPs.

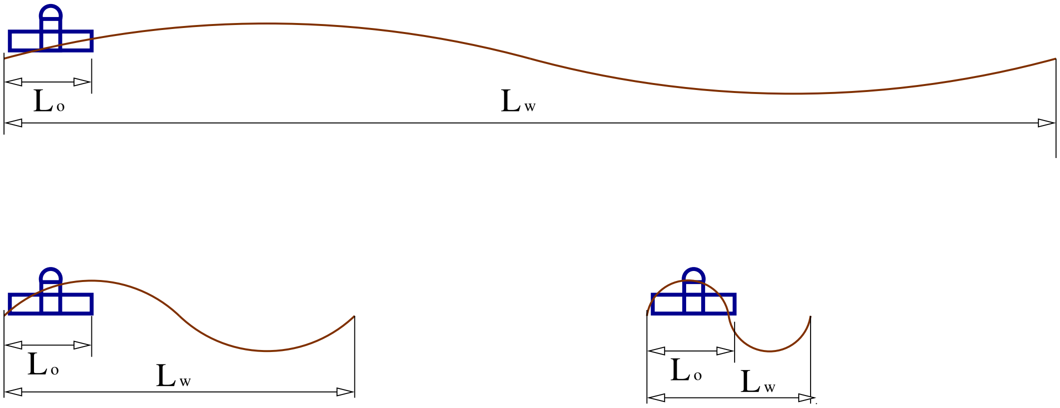


Figure 504.58: 1C vs 3×1C vs 3C. Three different cases of surface wave wave length. Upper case is where the surface wave length is 12 or more times longer than the object (NPP) dimension). Lower left case is where the surface wave length is only four times longer than the wave length, and lower right case is where the surface wave length is only two times longer than the object length.

504.4.2 Illustration of Use of 3C and 1C Seismic Motions

A simple example can be used to illustrate differences in 1C vs 3C seismic motions. Assume that a full 3C (6C, 3 translational components and 3 rotational components) motions at the surface are only recorded in one horizontal direction. From a 1C recorded component one can develop a vertically propagating shear wave in 1C, that exactly models 1C recorded motion. This is usually done using de-convolution Kramer (1996a). Figure 504.59 illustrates the idea of using a full 3C seismic wave field to develop a reduced, 1C wave field

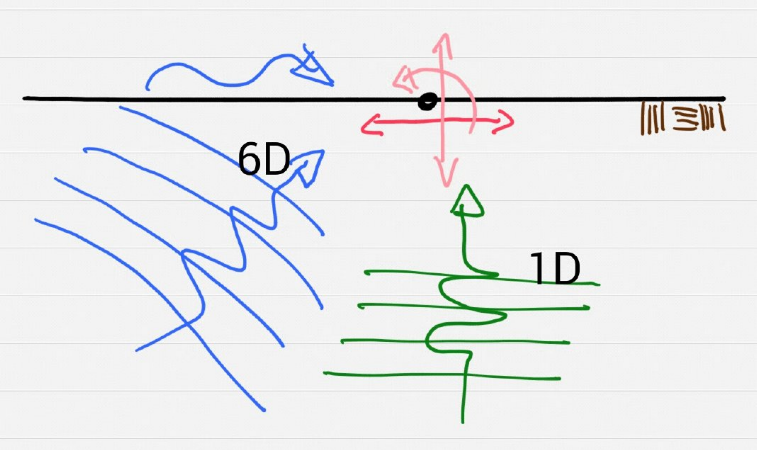


Figure 504.59: Illustration of the idea of using a full 3C (6C) seismic motion field to develop a 1C seismic motion field.

Two seismic wave fields, the original wave field and a subset 1C wave field now exist. The original wave field includes body and surface waves, and features translational and rotational motions. On the other hand, subset 1C wave field only has one component of motions, usually an SV component (vertically polarized component of S (Secondary) body waves).

Figure 504.61 shows a snapshot of a full 3C wave field, resulting from a large scale regional simulation, from a point source (simplified), propagating P and S waves through layers.

Figure 504.62 shows a snapshot of the same wave field as in Figure 504.61, now focused on an immediate viscinity of location of interest, where blue stick is positioned.

It should be noted that regional simulation model shown in Figures 504.62 and 504.61 is rather simple, consisting of a point source at shallow depth in a 3 layer elastic media. Waves propagate, refract at layer boundaries (turn more "vertical") and, upon hitting the surface, create surface waves (in this case, Rayleigh waves). In our case (as shown), out of plane translations and out of plane rotations are not developed, however this simplification will not affect conclusions that will be drawn. A seismic wave field with full 3 translations and 3 rotations (6C) will only emphasize differences that will be shown later.

Figures 504.63 and 504.64 show local free field model with 3C and 1C wave fields respectively.

Please note that seismic motions are input in an exact way, using the Domain Reduction Method Bielak et al. (2003a); Yoshimura et al. (2003a) and how there are no waves leaving the model out of DRM element layer (4th layer from side and lower boundaries). It is also important to note that horizontal motions in one direction at the location of interest (in the middle of the model) are exactly the same for both 3C motions case and for a 1C motions case.

Figure 504.65 shows a snapshot of an animation (available through a link within a figure) of difference

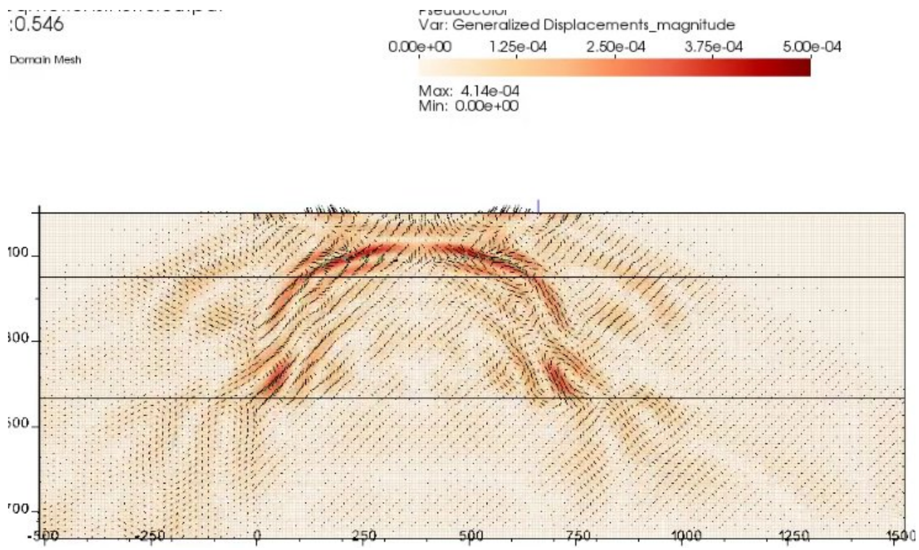


Figure 504.60: Snapshot of a full 3C wave field

Figure 504.61: Snapshot of a full 3C wave field, with body and surface waves, resulting from a point source at 45^o at depth, down-left. This is a regional scale model of a (simplified) point source (fault) with soil layers. Figure is a link to an animation of a full wave propagation.

in response of an NPP excited with full 3C (6C) seismic wave field, and a response of the same NPP to 1C seismic wave field.

Figures 504.66 and 504.67 show displacement and acceleration response on top of containment building for both 3C and 1C seismic wave fields.

Figures ?? and ?? show displacement and acceleration response on top of containment building for both 3C and 1C seismic wave fields.

A number of remarks can be made:

- Accelerations and displacements (motions, NPP response) of 6C and 1C cases are quite different. In some cases 1C case gives bigger influences, while in other, 6C case gives bigger influences.
- Differences are particularly obvious in vertical direction, which are much bigger in 6C case.
- Some accelerations of 6C case are larger that those of a 1C case. On the other hand, some displacements of 1C case are larger than those of a 6C case. This just happens to be the case for given source motions (a Ricker wavelet), for given geologic layering and for a given wave speed (and length). There might (will) be cases (combinations of model parameters) where 1C motions model will produce larger influences than 6C motions model, however motions will certainly again

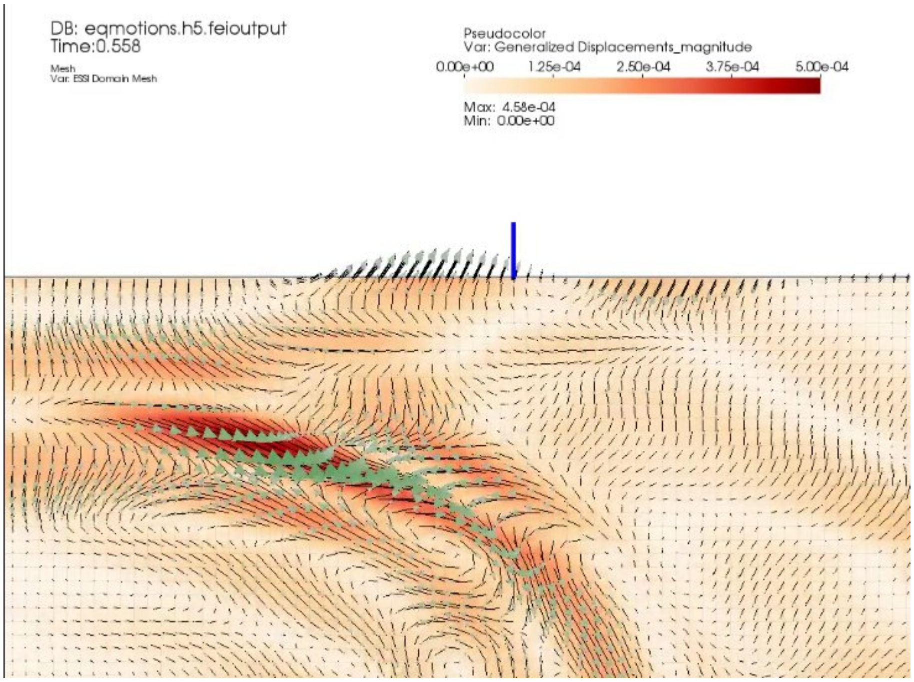


Figure 504.62: Snapshot of a full 3C wave field, with body and surface waves, resulting from a point source at 45^o at depth, down-left. This is a large regional scale model of a (simplified) fault with soil layers. Figure is a link to an animation of a full wave propagation.

be quite different. There will also be cases where 6C motions will produce larger influences than 1C motions. These differences will have to be analyzed on a case by case basis.

In conclusion, response of an NPP will be quite different when realistic 3C (6C) seismic motions are used, as opposed to a case when 1C, simplified seismic motions are used.

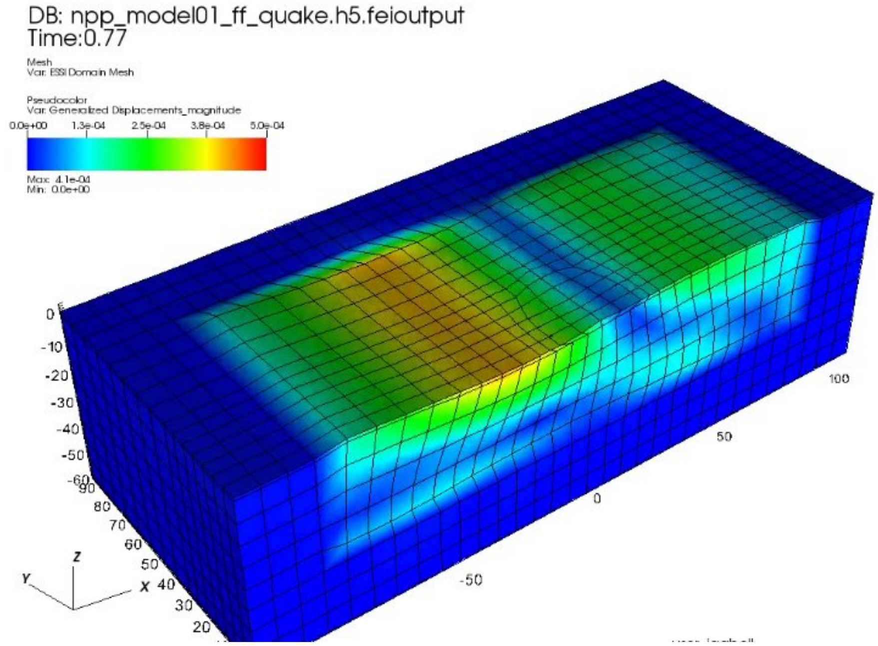


Figure 504.63: Snapshot of a full 3C wave field at the location of interest (where an NPP will be founded), featuring body and surface waves.

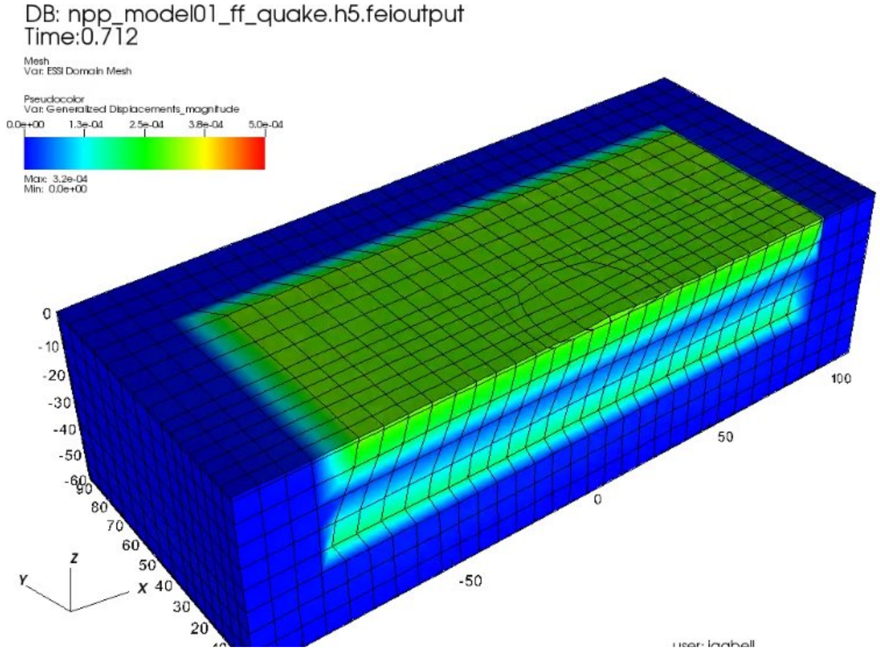


Figure 504.64: Snapshot of a reduced 1C wave field at the location of interest (where an NPP will be founded), featuring just SV body waves.

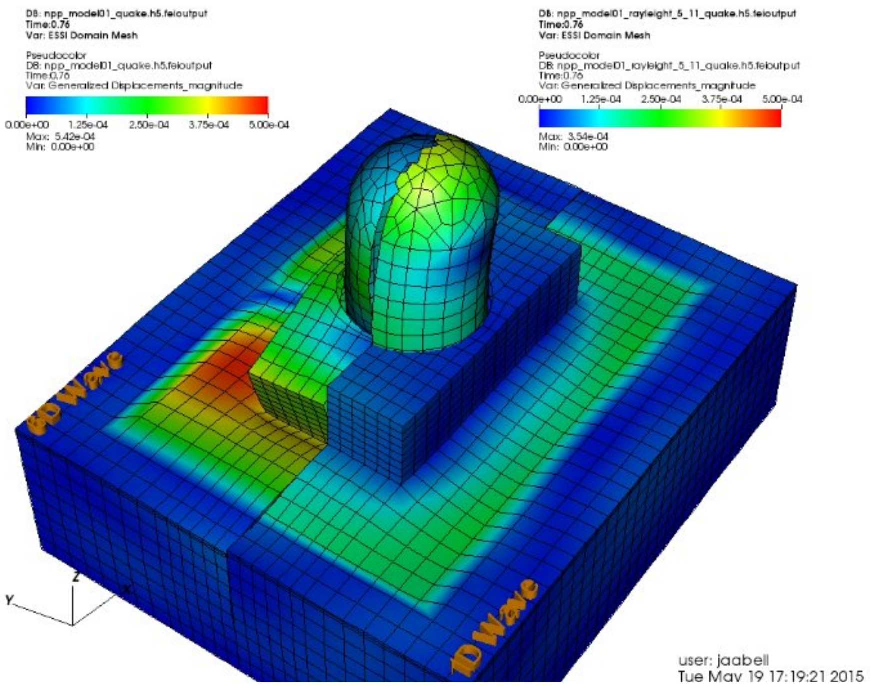


Figure 504.65: Snapshot of a 3C (6C) vs 1C response of an NPP, upper left side is the response of the NPP to full 3C wave field, lower right side is a response of an NPP to 1C wave field.

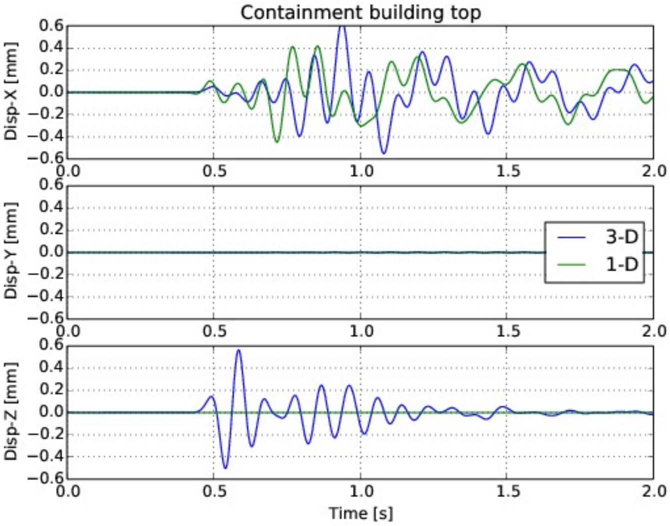


Figure 504.66: Displacements response on top of a containment building for 3C and 1C seismic input.

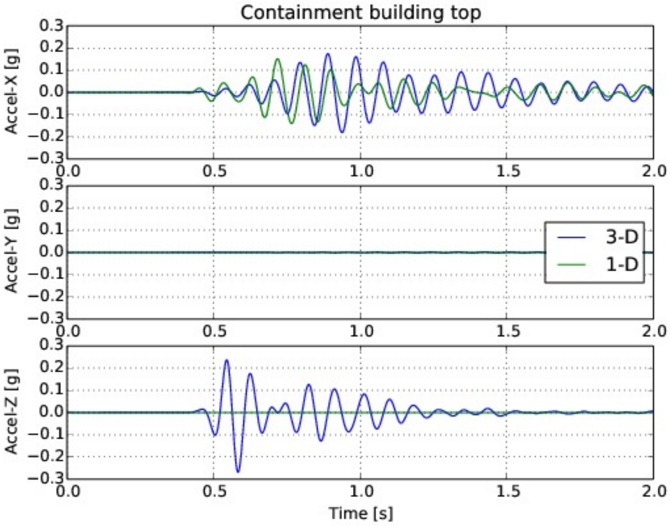


Figure 504.67: Acceleration response on top of a containment building for 3C and 1C seismic input.

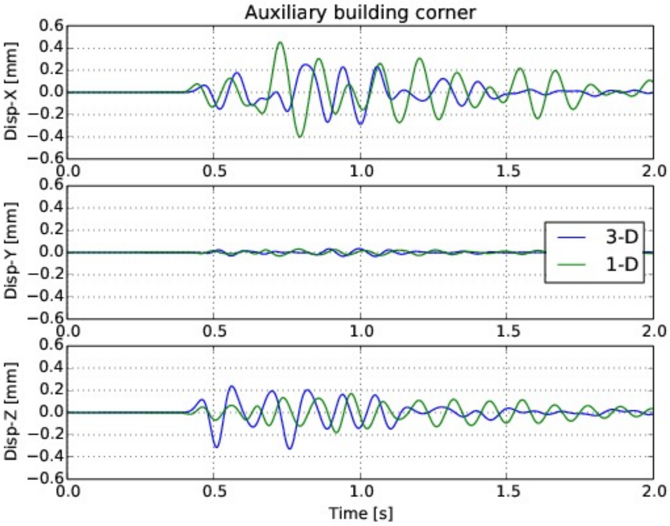


Figure 504.68: Displacements response at the top corner of auxiliary for 3C and 1C seismic input.

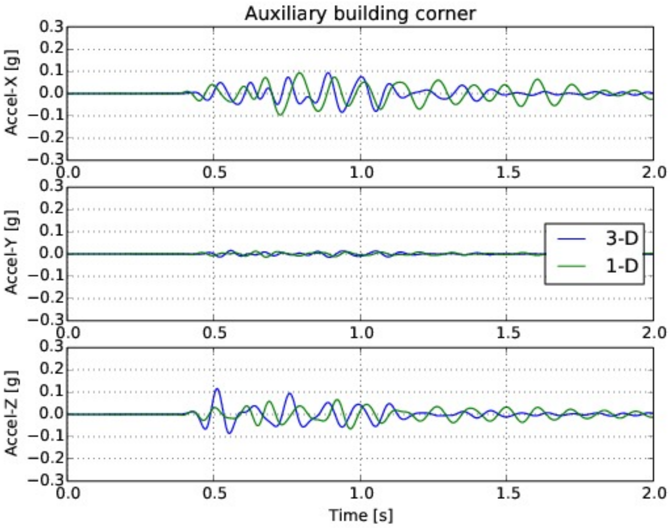


Figure 504.69: Acceleration response at the top corner of auxiliary for 3C and 1C seismic input.

504.5 3C (6C) vs 3 × 1C vs 1C Seismic Motions

This section is from our paper (Abell et al., 2018).

504.6 3D Nonlinear Modeling for Nuclear Power Plants

This section is based on Sinha et al. (2017).

It noted that input files for these models are available at this LINK, and can be directly simulated using Real-ESSI Simulator, `http://real-essi.us/`, that is available on Amazon Web Services, `https://aws.amazon.com/`.

504.6.1 Introduction

Seismic simulations to structures are often done by 1-D input excitations defined from a family of damped response spectra. These input motions are applied uniformly to the entire base of the structure regardless of its dimension and dynamic characteristics of the soil, foundation and motion itself. This not only ignores the foundation and its contact/interface with soil, soil-structure interaction (SSI) but also the 3C nature and variability of seismic waves.

Interest to study SSI effects has grown significantly in recent years. However Tyapin (2007) and Lou et al. (2011) note that even after four decades of extensive SSI research, there still exists a large gap. Lou et al. (2011) notes that spatial analysis of full model in 3D is hardly done. To reduce the amount of calculations, many existing publications simplify extremely the super-structure to spring mass damper model or consider only limited interaction. Elgamal et al Elgamal et al. (2008) performed a 3D analysis of a full soil–bridge system, focusing on interaction of liquefied soil in foundation and bridge structure. Jeremić et al. (2009) showed a full 3D soil-structure interaction of a prototype bridge, devised as a part of grand challenge, pre-NEESR project.

Investigations of SSI have shown that the dynamic response of a structure supported on elastic-plastic soil may differ significantly from the response of the same structure when supported on a rigid base Chopra and Gutierrez (1974); Bielak (1978). The difference comes because of the dissipation of part of the vibrational energy (seismic energy) by hysteresis action of the soil or structure itself. This results in damping of high frequency components, which could potentially prove quite useful for equipment that are prone to damage from high frequencies. On the other hand Jeremić et al. (2004) found that SSI can have detrimental effects on structural behavior as well and is dependent on the dynamic characteristics of the earthquake motion, the foundation soil and the structure.

Dissipation of energy during seismic events is another important factor to consider in design for its safety and economy. Dissipating energy in structure can lead to material degradation and damage. It is desired to dissipate most of the energy in soil with acceptable level of deformations in structure. A common neglect of plastic free energy has been observed in many publications, which results in clear violation of the second law of thermodynamics. A thermomechanical framework that can correctly evaluate energy transformation and dissipation in dynamic SSI simulation was presented by Yang et al. (2018, 2019a) based on works of Rosakis et al. (2000); Dafalias et al. (2002). This framework is applied to the prototype NPP model that is being analyzed in this paper. Locations with high possibility of damage are identified and insights on design improvement are discussed.

Only a few full 3D SSI interactions have been studied that too mainly focusing on bridges or small soil-foundation system. However, as per author's knowledge a full 3D non-linear analysis for a structure with soil-foundation-structure and contact/interface effects have not been investigated. Purpose of this paper here is to present a methodology for high fidelity modeling of seismic soil–foundation-structure (SFSI) interaction for a prototype of Nuclear Power Plant (NPP) with surface (shallow) foundation. Presented methodology employs the currently best available models and simulation procedures. In addition to presenting such state-of-the-art modeling, simulation results are used to illustrate non-linear-effects on seismic response of a prototype NPP model.

504.6.2 Model Development and Simulation Details

The Nuclear Power Plant (NPP) modeled here is a symmetric structure with shallow foundation of thickness $3.5m$ and size $100m$. Figure 504.70 shows a slice view of the model in normal y direction (perpendicular to plane of the paper). Solid brick elements were used to model soil and foundation. The NPP structure was modelled by elastic shell elements. This section describes the material and modeling parameters summarized in Table 504.5, foundation, structure, and contact. Given also is a brief description of staged loading and seismic force application using domain reduction method (DRM).

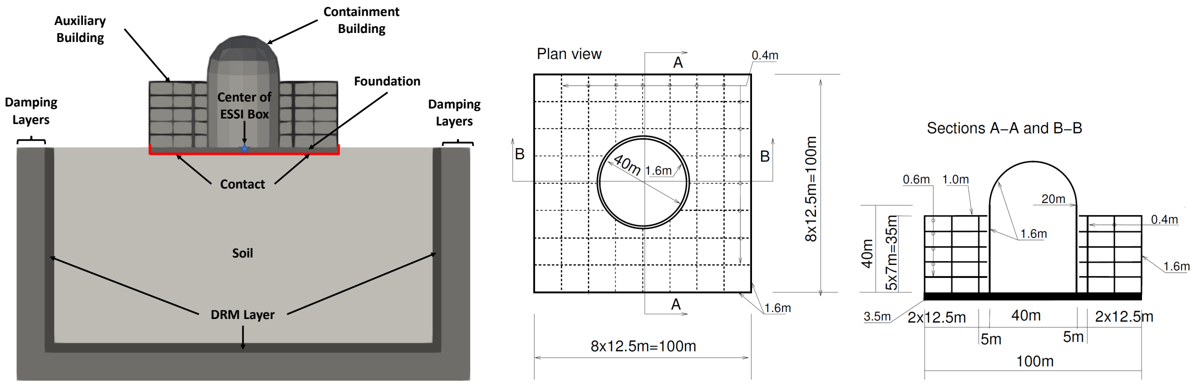


Figure 504.70: Nuclear power plant model with shallow foundation.

Soil	Unit weight , γ $[kPa]$	21.4
	Shear velocity , Vs $[m/s]$	500
	Young's modulus , E $[GPa]$	1.3
	Poisson's ratio , ν	0.25
	Shear strength , S_u $[kPa]$	650
	von Mises radius , k $[kPa]$	60
	linear kinematic hardening , h_a $[Mpa]$	30
	non-linear kinematic hardening , c_r	25
Structure	Unit weight , γ $[kPa]$	24.0
	Young's modulus , E $[GPa]$	20
	Poisson's ratio , ν	0.21

Structure	Unit weight , γ $[kPa]$	24.0
	Young's modulus , E $[GPa]$	20
	Poisson's ratio , ν	0.21
Contact	Initial normal stiffness , k_n^{init} $[N/m]$	1e9
	Stiffening rate , S_r $[/m]$	1000
	Maximum normal stiffness , k_n^{max} $[N/m]$	1e12
	Tangential stiffness , k_t $[N/m]$	1e7
	Normal damping , c_n $[Ns/m]$	100
	Tangential damping , c_t $[Ns/m]$	100
	Friction ratio , μ	0.25

Figure 504.71: Modeling parameters.

504.6.2.1 Structure Model

The NPP structure consists of auxiliary building, containment building and shallow foundation as shown in Figure ??. The auxiliary building consists of 4 floors of $0.6m$ thickness, ceiling floor of $1m$ thickness, exterior wall of $1.6m$ thickness and interior walls of $0.4m$ thickness. The exterior and interior walls are embedded down to the depth of the foundation. The containment building is a cylinder of diameter $20m$ and height $40m$ with wall thickness of $1.6m$. There is a gap of $0.2m$ between the containment and auxiliary building. Top of the containment building is covered by semi-spherical dome of radius $20m$. The foundation is square shallow footing of size $100m$ and thickness $3.5m$. The containment building and the auxiliary building were modelled as shell elements and foundation as linear brick elements, both having the properties of concrete of elastic Young's modulus $20GPa$, poison's ratio 0.21 and density $2400kg/m^3$. The containment building which is more flexible than the auxiliary building had its first mode as bending with fundamental frequency at $4Hz$.

504.6.2.2 Soil Model

The depth of the soil modelled below the foundation was 120 m, which is also the depth of DRM layer Sec 504.6.2.5. It is assumed that within this range the soil will plastify because of its self-weight, structure and seismic motions. The soil is assumed to be a stiff saturated-clay with undrained behavior having shear velocity of 500 m/s, unit weight of 21.4 kPa and Poisson's ratio of 0.25. To represent the travelling wave accurately for a given frequency, about 10 nodes per wavelength i.e. about 10 linear or 3 quadratic brick elements are required. Here, the seismic waves are analyzed up to $f_{max} = 10Hz$. The smallest wavelength λ_{min} to be captured thus, can be estimated as

$$\lambda_{min} = v/f_{max} \qquad (504.2)$$

where, v is the smallest shear wave velocity of interest. For $v = 500m/s$ and $f_{max} = 10Hz$ the minimum wavelength λ_{min} would be $(500m/s)(10/s) = 50m$. Choosing 10 nodes/elements per wavelength the element size would be 5m. Jeremić et al. (2009); Watanabe et al. (2017) state that even by choosing mesh size $\Delta h = \lambda_{min}/10$, smallest wavelength that can be captured with confidence is $\lambda = 2\Delta h$ i.e. a frequency corresponding to $5f_{max}$. Based on the above analysis, soil was modeled as linear 8-node brick elements with grid spacing of $\Delta h = 5m$.

Because of the complex plastic-behavior of the soil many sophisticated models Yang et al. (2003); Dafalias and Manzari (2004b) have been developed to capture the non-linear response of soil. Wair et al. (2012) provides an empirical correlation to predict the shear strength of soil for given shear velocity V_s. Dickenson (1994) proposed the following relationship Eq 504.3 between V_s and undrained strength S_u for cohesive soils in San Francisco Bay Area.

$$V_s[m/s] = 23(S_u[kPa])^{0.475} \tag{504.3}$$

Thus, for $V_s = 500m/s$, the undrained strength S_u would be $650kPa$. Here, two scenarios of soil properties were considered in analysis. One linear elastic and the other as von-Mises with non-linear kinematic hardening of Armstrong – Frederick type. For S_u of $650kPa$ and $E = 1.3Gpa$, the non-linear inelastic model was calibrated for yield strength achieved at 0.01% shear strain with linear kinematic hardening rate (h_a) as $30MPa$ and non-linear hardening rate (c_r) as 25. The soil properties is summarized in Table 504.5. The stress-strain response for the non-linear material model is shown in Figure ??

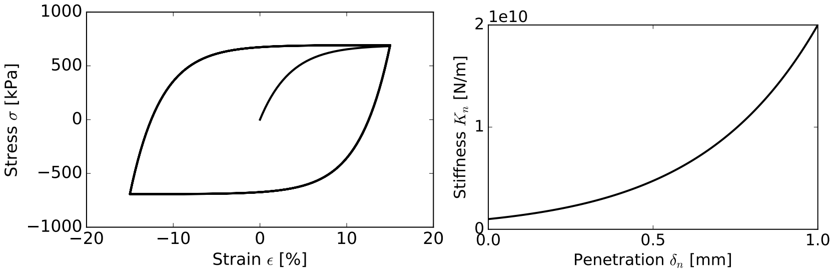


Figure 504.72: Soil and contact/interface modeling.

504.6.2.3 Interface/Contact Modeling

Node-to-node penalty based soft contact/interface element Sinha and Jeremić (2017) was used to model the interaction between foundation and soil as they are not one continuum material. In soft contact, normal contact/interface force F_n from soft-soil is assumed to increases exponential with penetration δ_n

as shown in Eq 504.4. The normal force F_n and stiffness K_n in defined as

$$\begin{aligned} F_n &= k_n^{init} * exp(S_r * \delta_n) * \delta_n \\ K_n &= max(k_n^{init} * exp(S_r * \delta_n) * (1 + k_n^{init} * \delta_n), k_n^{max}) \end{aligned} \tag{504.4}$$

where δ_n refers to the relative displacement between contact/interface node pairs in normal contact direction, k_n^{init} refers to the normal stiffness in normal contact/interface direction, S_r refers to the stiffening rate in normal contact direction. k_n^{Max} refers to maximum normal stiffness and provides a cap on exponentially increasing stiffness to make the solution numerically stable. The soft contact/interface was implemented to capture the phenomenon of increasing stiffness of soil with increasing penetration. Figure ?? shows the stiffness curve with penetration for the chosen contact/interface parameters also shown in Table 504.5.

Contact elements were applied all around the foundation connecting to the soil as shown in Figure ?? in red color zone. To ensure the stability of the numerical solution, the penalty stiffness in normal direction was chosen $2-3$ order magnitude greater than the stiffness of the soil. The Coulomb's friction coefficient μ between the soil and the foundation was chosen as 0.25. Viscous damping of $100 Ns/m$ in normal and tangential damping was provided to model viscous damping arising from water.

504.6.2.4 Seismic Motions

3C seismic motions were developed by Rodgers (2017) using SW4 (Serpentine Wave Propagation of 4^{th} order) Petersson and Sjögreen (2018) for an earthquake of magnitude (M_w) of 5.5 modelled with a point source on a fault of dimension $5.5 \times 5.6 km$ with up-dip rupture slip model. The ESSI (Earthquake Soil Structure Interaction) box to capture the free-field motion was located on the foot-wall of the reverse thrust fault. The generated motion had a directivity effect as the fault slips and propagates in x-direction. Also, since the ESSI box was located perpendicular to the fault, strong motions in y-direction was expected.

Acceleration and displacement time-series of the motion at the center of ESSI box is shown in Figure 504.86. The peak ground acceleration (PGA) in x and y direction is about $0.5g$. Significant amount of vertical motions PGA of $0.2g$ can be observed which is neglected in many conventional seismic simulations. Since, the fault is located at foot-wall side of reverse thrust fault, there is permanent subsidence of about $50mm$ in z-direction at the end of shaking event. Fourier transform and response spectrum of the motions are shown in Figure 504.87. The frequency range of the motion is within $20Hz$. Response spectrum plot shows amplification for natural frequency greater than $2Hz$. Since, many equipments in nuclear industry operate at high frequencies, determination of high frequency excitation of NPP building is critical for design.

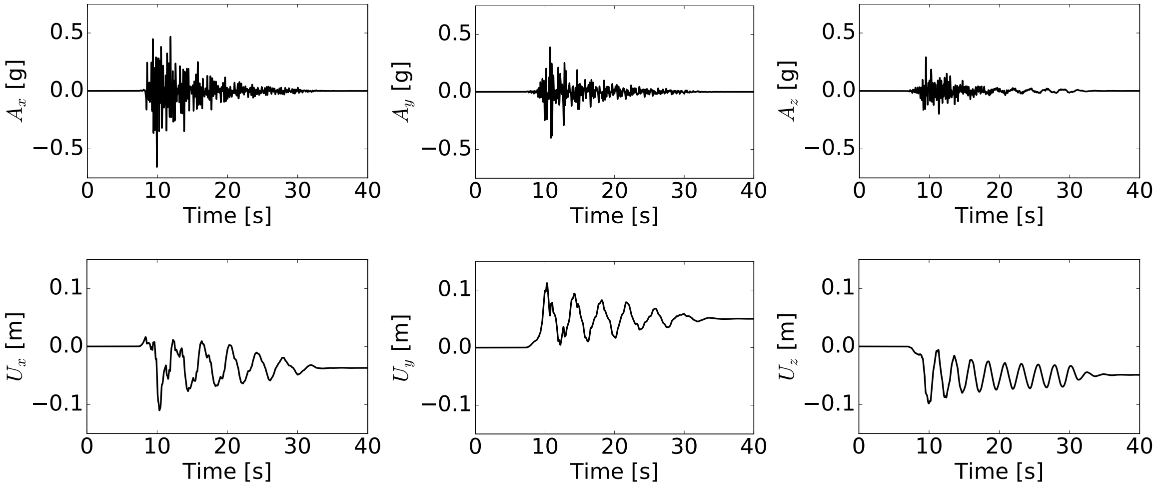


Figure 504.73: Acceleration and displacement time series.

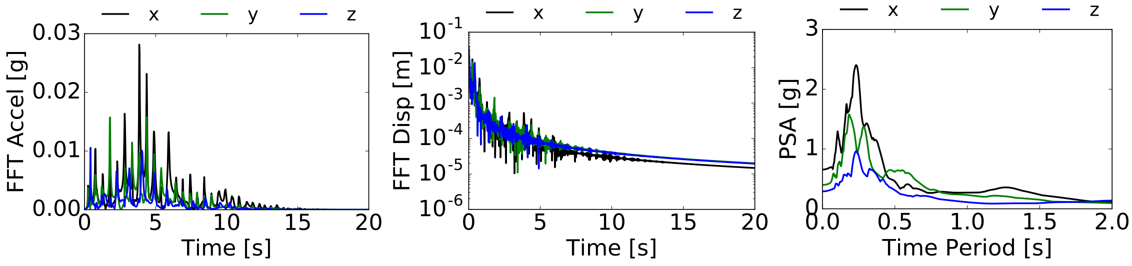


Figure 504.74: Strong motion Fourier transform and response spectrum.

504.6.2.5 Domain Reduction Method

Domain Reduction Method (DRM) Bielak et al. (2003a) was used to apply 3C seismic motions generated from SW4 all around the model as shown in Figure ??. DRM is one of the best methods that can apply free field 3C ground motions to a finite-element model. It features a two-stage strategy for a complex, realistic 3D earthquake engineering simulation. First, is the generation of free field model with correct geology and second is the application of the generated free-field to the structure of interest. The DRM layer here is modeled as a single layer of elastic soil. Three damping (absorbing) layers adjacent to DRM layer were modeled to prevent incoming of reflected waves. For this analysis, 60% Rayleigh damping was applied in each of the damping and DRM layers. The Rayleigh damping was applied in the frequency range of 1-5Hz.

504.6.2.6 Staged Simulation

The whole analysis was simulated with two loading stages. First stage was static self-weight to get the initial stress state of the soil and contact/interface elements. In second stage, seismic motion was applied using DRM method. For each stage, equilibrium was achieved using full Newton-Raphson method with a small tolerance of $1e^{-4}N$ on second-norm of unbalanced force. For dynamic analysis, Newmark integration method with numerical damping $\gamma = 0.7$ was used. Rayleigh damping of 2% in structure and 30% in soil was applied. The time step considered here was 0.02 seconds with simulation running in total for 40 seconds.

The analysis was run in parallel in Real-ESSI Simulator Jeremić et al. (1988-2025), `http://real-essi.info`. on eight CPUs. The model consisted of about $300k$ degrees of freedoms (dofs). Four scenarios (a) elastic no contact/interface (b) elastic with contact (c) elastic-plastic no contact and (d) elastic-plastic with contact/interface were performed. In this paper, unless specified *elastic* means elastic without contact/interface and *inelastic* means elastic-plastic with contact.

504.6.3 Simulation Results

Due to the space restriction, only few locations are selected to study the non-linear effects on NPP structure. The selected locations are shown in Figure 504.75. Since the containment building is more flexible than auxiliary building, location (D) in Figure 504.75 located on the top of the containment building is naturally the point of interest as it describes maximum drift during shaking. Three locations (A), (B) and (C) located at center of foundation is also selected to study the slip at interface during shaking.

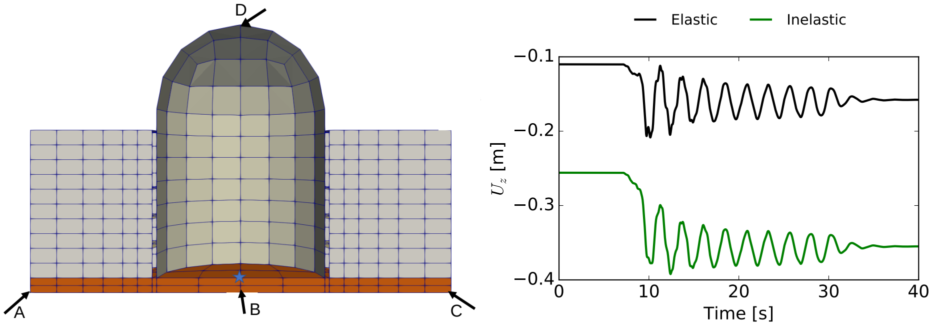


Figure 504.75: Locations selected to study non-linear effects and plot of total displacement at center of model *Elastic* (elastic with contact) and *Inelastic* (Elastic-Plastic with contact).

Since the site is located on the foot wall, during seismic shaking the whole structure along with soil subsides down by about $50mm$ in elastic and $100mm$ in inelastic case. Overall, if self-weight stage is also included, the soil settles by $150mm$ in elastic and $350mm$ in inelastic case as shown in Figure ??.

It is important to predict the development of high frequency excitation during shaking because it can prove to be alarming (when close to fundamental frequency) for nuclear-equipment. These high frequencies are thus, important to be monitored, predicted during earthquakes, for design of nuclear building to ensure the safety of equipment. Figure 504.76 plots the acceleration and its Fourier amplitude for the location (D). It is interesting to observe, the elastic-plastic analysis kills high frequency excitations in the structure which are persistent in elastic analysis. Elastic-plastic soil shows natural damping to some high frequencies because of dissipation of energy in form of heat by hysteresis loop.This can prove to be bigly useful for safe operation of nuclear equipments even at strong seismic events. The effects of contacts coupled with elastic-plastic material leads to huge dissipation of energy reducing the high frequency modes. In Z-direction, very little significant excitation was observed.

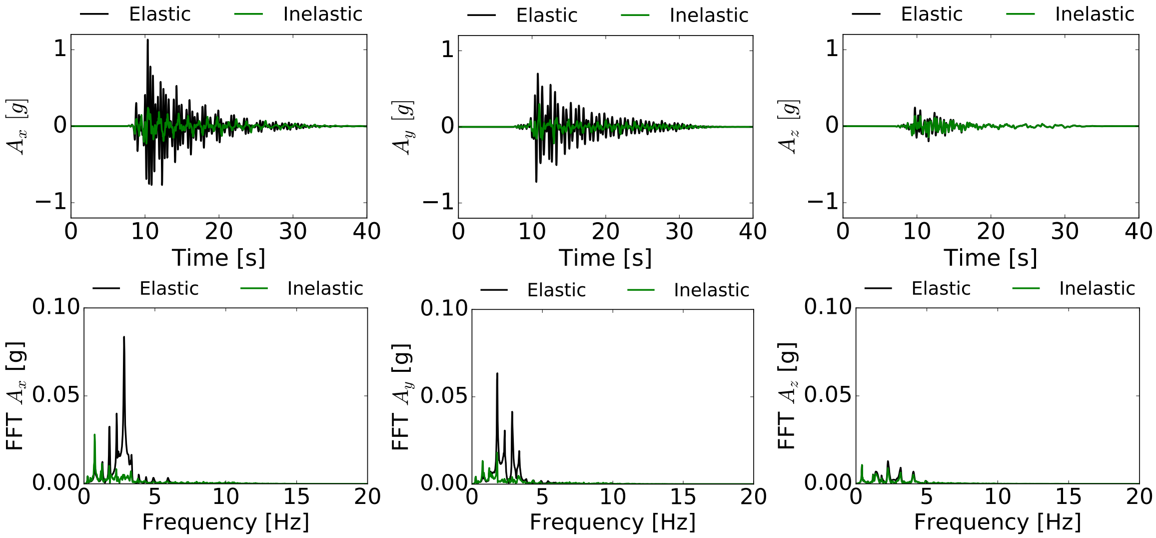


Figure 504.76: Seismic response at top of containment building *Elastic* (elastic without contact) and *Inelastic* (elastic-plastic with contact).

The introduction of contact/interface can result in opening and closing of gaps at the soil-foundation interface for stronger earthquakes. However, here for the considered seismic motion for both elastic and inelastic case with contact, no uplift was observed. Figure 504.77 shows the relative displacement of NPP structure for elastic and inelastic analysis at 11 seconds. In elastic case, the structure drifts a lot while the deformation in soil remains small. Whereas, in the inelastic case, the soil deforms and plastify in z-direction keeping the structure deformation small. Thus the elasto-plastic soil acts as a natural base isolators material. This demonstrates that for the considered earthquake motions, the elastic-plastic soil can prove to quite beneficial because of small deformation and excitation in structure.

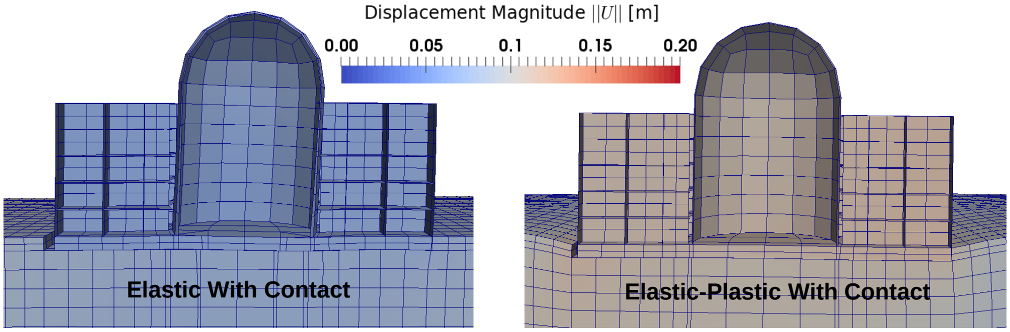


Figure 504.77: Deformation of the NPP structure at 11 seconds (scaled 100 times).

Figure 504.78 plots the inter-slip of foundation with respect to soil at location (A), (B) and (C) for elastic and elastic-plastic case with contact. It can be observed that point (A) and (B) slips both relatively towards each other describing the presence of surface waves. The center of the foundation (B) bends and slides comparatively less than the exterior ends. This also strongly shows the directivity effects of the motion coming from the $-x$ to $+x$ direction. The directivity effect is more pronounced in inelastic analysis resulting in comparatively more slip and permanent deformation. Careful observation of sliding in x-direction, shows a permanent slip of $18mm$ for elastic-plastic soil. Although not shown in figure, the whole NPP structure show tendency of rotation about its center of mass during the DRM stage.

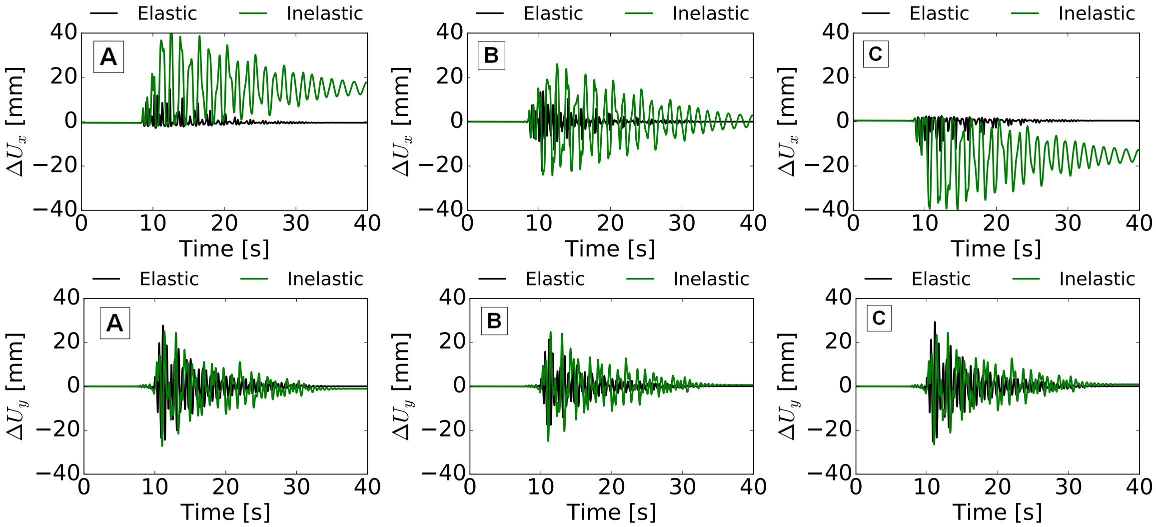


Figure 504.78: Slip of foundation with respect to soil beneath it in x-y slice plane for *Elastic* (elastic with contact) and *Inelastic* (Elastic-Plastic with contact).

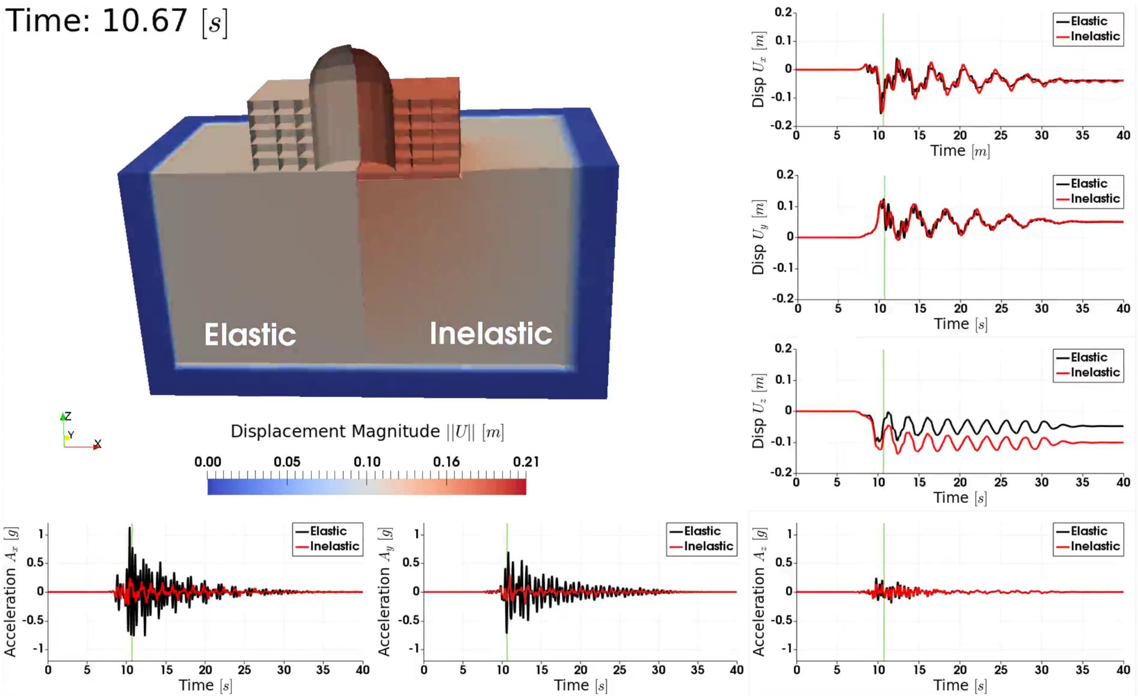


Figure 504.79: Animation of a linear elastic vs inelastic response of an NPP.

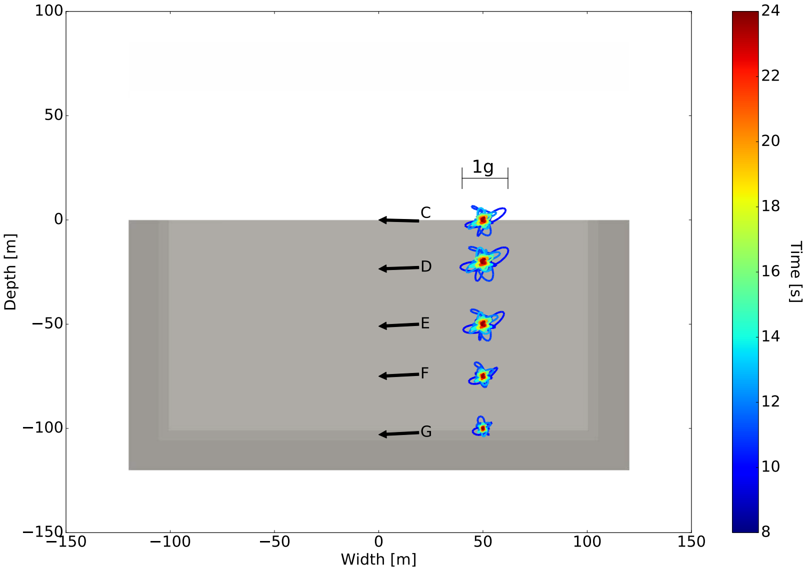


Figure 504.80: Free Field.

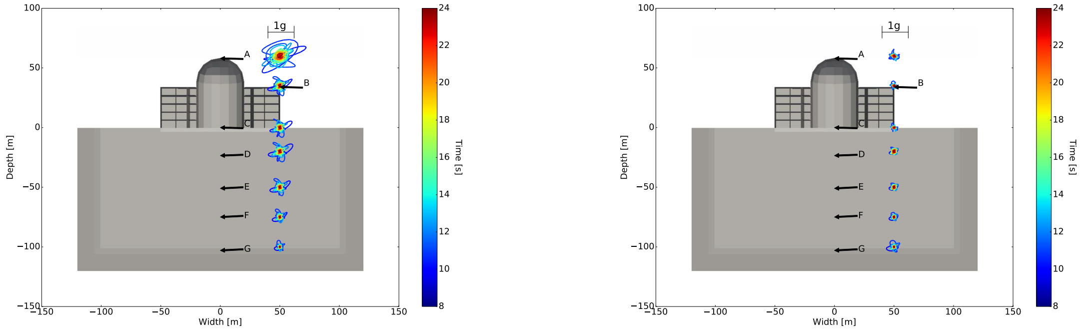


Figure 504.81: Acceleration traces, elastic vs inelastic.

504.6.4 Energy Dissipation

The energy dissipation in decoupled elastic plastic material under isothermal condition is given by Yang et al. (2019a):

$$\Phi = \sigma_{ij}\dot{\epsilon}_{ij} - \sigma_{ij}\dot{\epsilon}_{ij}^{el} - \rho\dot{\psi}_{pl} \geq 0 \tag{504.5}$$

where Φ is the rate of change of energy dissipation per unit volume (or dissipation density), σ_{ij} and ϵ_{ij} are the stress and strain tensors respectively, ϵ_{ij}^{el} is the elastic part of the strain tensor, ρ is the mass density of the material, and ψ_{pl} is the plastic free energy per unit volume (or plastic free energy density). Note that Equation 504.8 is derived from the first and second laws of thermodynamics, which indicate the conditions of energy balance and nonnegative rate of energy dissipation, respectively. Considering all possible forms of energy inside SSI system, the energy balance between input mechanical work W_{Input} and the combination of internal energy storage E_{Stored} and energy dissipation $E_{Dissipated}$ can be expressed as:

$$W_{Input} = E_{Stored} + E_{Dissipated} = KE + SE + PF + PD \tag{504.6}$$

where KE is the kinetic energy, SE is the elastic strain energy, PF is the plastic free energy, and PD is the energy dissipation due to material plasticity. Equations for each energy component can be found in Yang et al. (2018). Note that the plastic dissipation term PD includes energy dissipated in both elastic plastic solids (soil) and contact/interface elements.

Figure ?? shows the accumulated plastic dissipation density field of the NPP model at the end of seismic event. The super-structure does not dissipate energy since it is modeled as a linear elastic material. Significant amount of seismic energy is dissipated in the contact/interface zone between the structure and underlying soil, especially at regions around the corners and edges of the foundation.

An arch-shaped elastic region is formed under the structure, where the soil moves together with the foundation and dissipates little energy. Such observation is consistent with classic bearing capacity analysis, which also indicates the formation of a relatively undeformed "active zone" beneath foundation.

As can be observed in Figure ??, the plastic dissipation density at location (A) is the highest. From Figure ??, it can be observed that more than 80% of the total input work is dissipated due to material plasticity or contact slipping. About 70% of the energy dissipation happens due to contact/interface slipping, which indicates that the property and behavior of the interface between foundation and soil is crucial in SSI system. It is worth pointing out that there is about 10% of the input work transformed into plastic free energy, which falls in the typical range reported by Taylor and Quinney (1934).

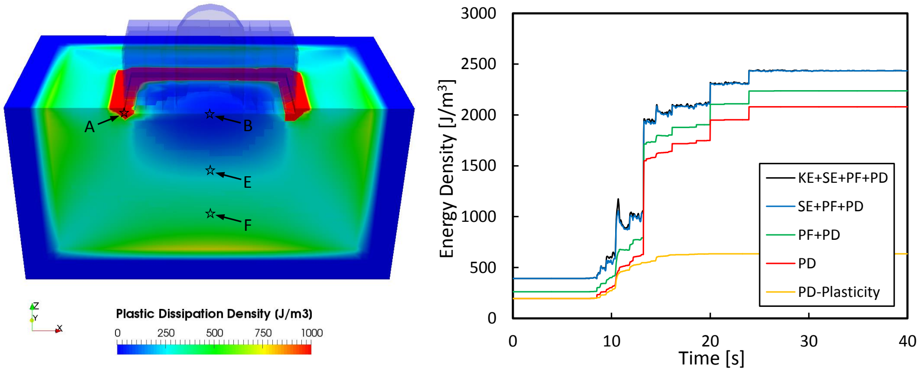


Figure 504.82: Energy dissipation in SMR model for inelastic (elastic-plastic soil with contact).

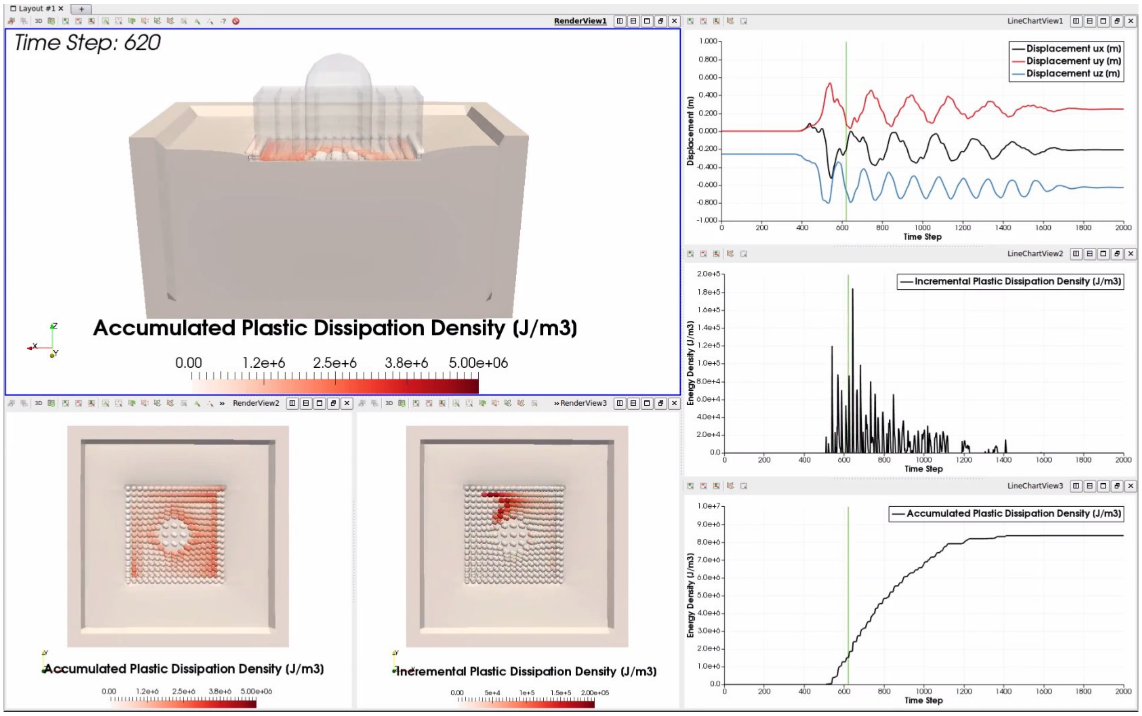


Figure 504.83: Animation of energy dissipation for an NPP.

504.6.5 Conclusion

Presented was a high fidelity seismic simulation methodology for investigating non-linear SSI effects on NPP structures. The site being in foot-wall of the reverse thrust fault, results in permanent subsidence in vertical direction. Due to plastification, elastic-plastic soil produces comparatively more vertical settlement than linear elastic soil. It also leads to damping of some higher frequency waves. This effect of non-linear material could be beneficial to the machines which are fatal to high frequency waves. It was found that with elastic-plastic soil, there was comparatively less seismic excitation and deformation in the NPP structure for the considered seismic motion. This illustrates that the stiff soil does not necessarily help in seismic behavior of structure. This also emphasizes the fact that linear elastic modeling of soil can lead to wrong conclusions resulting in huge capital loss. With the advancement of super-computers, uncertainty in modeling can be significantly reduced by following the high-fidelity modelling techniques discussed in this paper.

The non-linear effect studied here is with respect to the specific motion (Mw 5.5 up-dip slip fault). The non-linear effect cannot be fully described using this motion itself. More similar kind of research studies using different motions and geology needs to be carried out to find out the overall non-linear and geology effects on soil-structure interaction. The author also feels that new quantities needs to be formulated to study and compare different models to categories and unify the nonlinear effects on Soil Structure Interaction (SSI) effects. Energy dissipation analysis showed that the soil close to the corners and edges of the NPP structure dissipates large amount of seismic energy. An arch-shaped elastic region was identified where design can be improved so that soil strength at these locations can contribute to the overall safety of the SSI system.

504.7 3D Nonlinear Modeling for Small Modular Reactors (SMRs)

Figure 504.84 shows a generic model (half is only shown, full model is simulated) of an SMR. It is important to note extensive contact/interface zone of SMR walls with surrounding soil. This brings forward a number of modeling and simulation issues for SMRs:

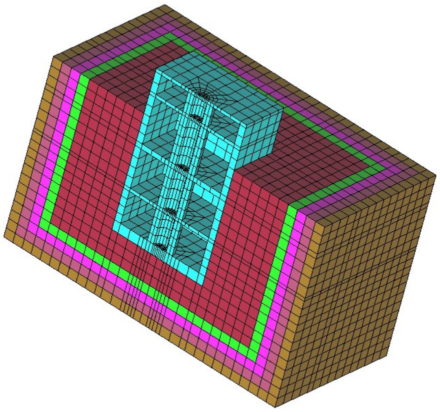


Figure 504.84: Generic model of a Small Modular Reactor (SMR).

- Seismic Motions: There will be a difference in seismic wave fields at the surface and at depth. Surface waves do extend somewhat into depth (about two wave lengths at most Aki and Richards (2002)), so different motion frequencies, wave lengths, and depending on soil stiffness, SMR will experience very different motions at tha surface and at the base. This seismic motion incoherence will affect seismic response of an SRM.
- Nonlinear/Inelastic Contact: Large contact/interface zone, with its nonlinear/inelastic behavior will have significant effect on dynamic response of a deeply embedded SMR.
- Nonlinear/Inelastic Soil Behavior: With deep embeddement, dynamic behavior an SMR is significantly influenced by the nonlinear/inelastic behavior of adjacent soil.
- Buoyant Forces: With deep embeddement, and (a possible) presence of underground water (water table that is within depth of embeddement), water pressure on walls of SMR will create buoyant forces. During earthquake shaking, those forces will change dynamically, with possibility of cyclic mobility and liquefaction (even for dense soil, due to water pumping during shaking).
- Uncertainty in Motions and Material: Due to large contact/interface area and significant embeddement, significant uncertainty and variability (incoherence) in seismic motions will be present.

Moreover, uncertainties in properties of soil material surrounding SMR will add to uncertainty of the response.

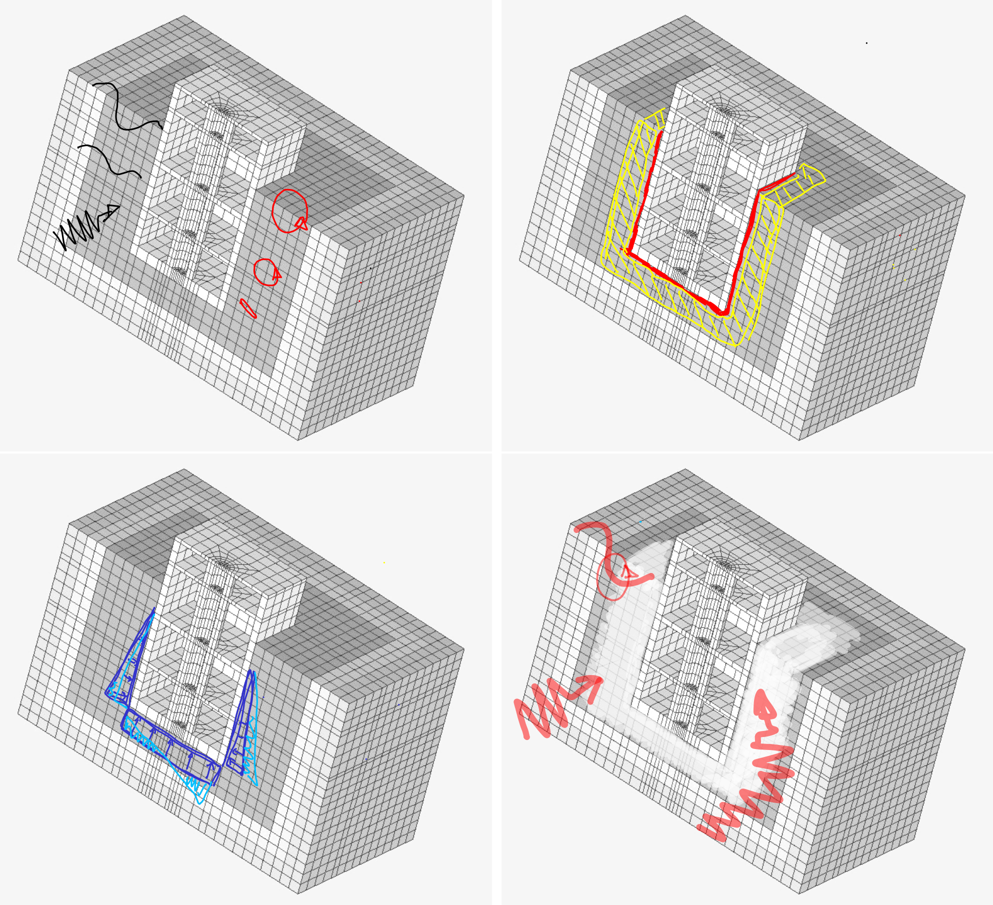


Figure 504.85: Four main issues for realistic modeling of Earthquake Soil Structure Interaction of SMRs: variable weave field at depth and surface, inelastic behavior of contact/interface and adjacent soil, dynamic buoyant forces, and uncertain seismic motions and material.

Writeup that follows is based on Wang et al. (2017).

It noted that input files for this model are available at this LINK, and can be directly simulated using Real-ESSI Simulator, `http://real-essi.info/`, that is available on Amazon Web Services, `https://aws.amazon.com/`.

504.7.1 Introduction

Seismic performance of nuclear facilities is carefully analyzed considering the significant problems that damage of such structures can bring. The structure investigated here is a deeply embedded Small

Modular Reactor (SMR). Model SMR structure analyzed here is deeply embedded (36 meters) with only 14 meters of structure above ground.

In recent years, many researchers (Spyrakos et al. (1989); El Ganainy and El Naggar (2009); Iida (2012)) made efforts to perform realistic modeling of dynamic SSI system and seismic response of underground structure. Romero et al. (2013) coupled FEM and BEM method to model wave propagation in elastic foundation and corresponding dynamic response of the structure. Fatahi and Tabatabaiefar (2013) investigated the seismic performance of mid-rise buildings on soft soils using existing earthquake records. An elastoplastic SSI analysis was conducted by Shahrour et al. (2010) to explore the seismic response of tunnels in soft soils. However, some inherent modeling uncertainties still existed in these previous studies and were not well addressed:

- A very important modeling uncertainty comes from the ground motion. For surface structures, it is common to use historical earthquake records and simplified 1C seismic wave propagation models (Paolucci et al. (2008)). Vertical ground motion are usually neglected. However, Oprsal and Fäh (2007) has emphasized the necessity to use 3C ground motion by showing the big difference between 1C and 3C computation result. The modeling uncertainty of input motion for seismic modeling of underground structure is even higher. Due to the lack of ground motion observations along the depth, deconvolution method is usually adopted in many studies (Elgamal et al. (2008)) to get the excitation motion at certain depth. The deconvolution procedure represents a 1C linear inverse analysis. This inverse analysis is seemingly simple but it can introduce considerable confusion and uncertainties to the modeling system (Mejia and Dawson (2006)).

- Another uncertainty comes from the method that is used to input seismic motion into SSI system. Usually free field motion are directly imposed to the structure without considering ESSI effects. This is especially common for underground structures where simplified static loads are directly imposed and these structures are simply designed to accommodate the estimated free field deformation(Hashash et al. (2001)).

- Nonlinear ESSI effects are also important factor that is neglected or simplified in many existing studies. There are three sources of significant nonlinearity in an ESSI system: (a) Inelastic (elastic-plastic) behavior of soil, (b) inelastic (elastic-plastic) behavior of the contact/interface zone, and (c) inelastic (elastic-damage-plastic) behavior of the structure. Early works found that structural response can be quite different when elastoplasticity of surrounding soil is considered (Bielak (1978); Iguchi and Luco (1981)). In addition to that Jeremić et al. (2004) reported that ESSI behavior can have both beneficial and detrimental effects on structural behavior. The nonlinear contact (interface) was analyzed by Hu and Pu (2004) and it was shown that its accurate modeling is a key part to realistic modeling of ESSI systems.

Due to computational limitations and complicated nature of ESSI problems mentioned above, there exist only few high-fidelity ESSI simulations, for bridges (Jeremić et al. (2009)) and tunnels (Corigliano et al. (2011)). To the Author's knowledge, there is no high fidelity (realistic 3C motions, realistic elastic plastic soil modeling, and realistic contact/interface modeling) for a deeply embedded SMR structure. In this paper, we present high fidelity modeling of SMR using state-of-the-art ESSI methodology. Realistic 3C free field seismic motions are modeled using regional scale wave propagation models. Developed free field motions are then input into ESSI system using Domain Reduction Method (Bielak et al. (2003a)). Modeling description section presents inelastic/nonlinear modeling details for elastic-plastic models of surrounding soils and nonlinear interface/contact behavior. The inelastic/nonlinear modeling result are compared with linear elastic in Simulation Results section.

In addition to accurate modeling of 3C motions and nonlinear effects, energy propagation through the model is also accurately modeled. Energy dissipation is a widely used indicator of material damage in elastic plastic materials. A common misconception does exist, however, on the meaning of plastic work and plastic energy dissipation, as observed in a number of publications. Correct evaluation of energy dissipation should follow the principles of thermodynamics that incorporated plastic free energy (Rosakis et al., 2000; Dafalias et al., 2002). The thermodynamics framework presented by Yang et al. (2018, 2019a) is implemented in the Real-ESSI Simulator Jeremić et al. (1988-2025), and is used to perform energy analysis on the SMR model in this paper. Energy dissipation in the SMR model is discussed in some detail.

504.7.2 Domain Reduction method

Input seismic motions into finite element model is an indispensable step for the simulation of soil structure interaction. The method we used here is called Domain Reduction Method, developed by Bielak et al. (2003a). It is a modular, two-step dynamic procedure aimed at reducing the large computation domain to a more manageable size. Firstly, large scale regional free field model is developed encompassing causative fault and location of SMR structure (however SMR structure is not present, it is a free field model). Time series of free field motions (displacements and accelerations) are recorded at locations of DRM elements, a single layer of finite elements encompassing soil structure SMR model. Those motions are then used in the second step, to develop effective forces that are used to input free field motions into ESSI SMR model.

504.7.3 3C Free Field Motions

Development of free field motions was done using a fourth order finite difference program SW4 (Petersson and Sjögreen (2018) developed at LLNL. Modeled was propagation of fault rupture in a model with dimensions $9km \times 6km \times 20km$. The magnitude of simulated earthquake is 5.5. The shear wave velocity of soils in surface layer (500 meters thick) is V_s = 500m/s. Motions were recorded in a box with dimensions $300m \times 300m \times 200m$

The characteristic ground motions recorded by ESSI nodes are plotted in Figure 504.86. The peak ground acceleration (PGA) in x and y direction is about 1g. Apart from that, significant amount of vertical motions with PGA $0.5g$ is also observed. The peak ground displacement (PGD) is about 0.1m in horizontal direction. Since ESSI box is located in the foot wall of the reverse fault, the permanent ground subsidence of about 6cm is recorded. Fourier transformation and response spectrum of the motions are shown in figure 504.87. The frequency range of the motion is within 15Hz. The dominant frequency of the motion is around 5 Hz. In response spectrum, we also see significant resonance effects for structure whose fundamental period is around 0.2s corresponding to 5 Hz fundamental frequency.

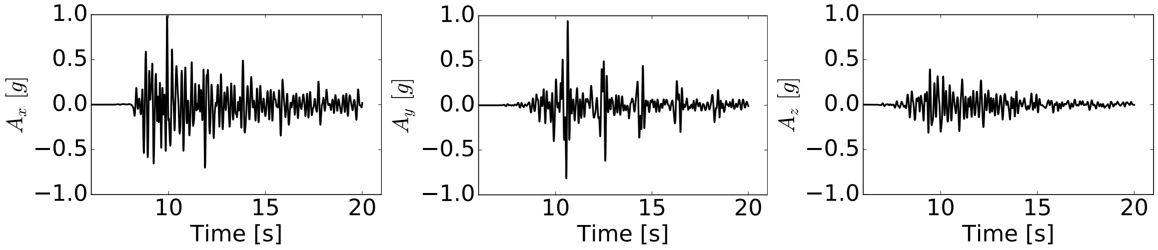


Figure 504.86: Acceleration and Displacement Time Series of Motion

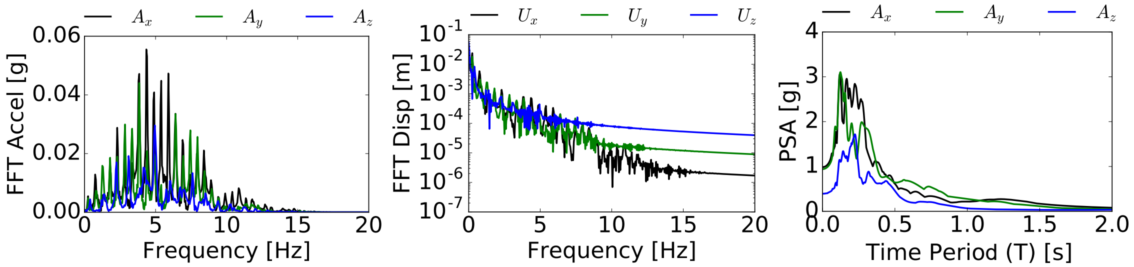


Figure 504.87: Strong Motion Fourier Transform and Response Spectrum

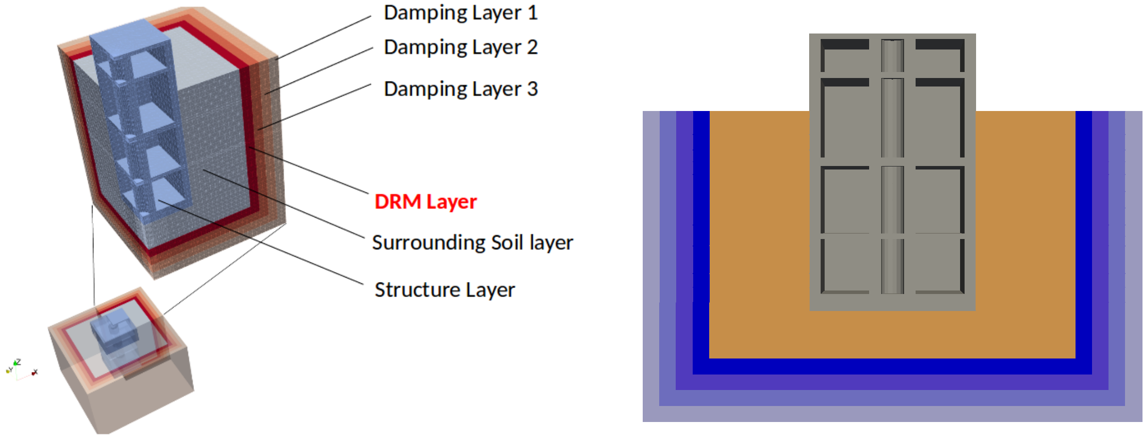


Figure 504.88: FEM model for an SMR.

504.7.4 Model Description

In order to reduce model size using DRM method, we develop our target model with 6 layers. As shown in figure 504.88, the innermost part is a structure layer, which is surrounded by a soil layer. Following that, there is a DRM layer used to apply equivalent earthquake force. Outside DRM layer, there are three damping layers. These damping layers are designed to add high Rayleigh damping so that the outgoing wave (vibrations of structure, radiation damping) can be adsorbed. Table 504.5 shows the material used. The size of whole FEM model is 72m×72m×56m. There are 177,806 nodes, 20172 27-node brick elements, 3,177 contact/interface elements (modeling the interface between soil and embedded structure), with a total of over 533 thousand degrees of freedom (DoFs)l. The average mesh size is 3 meters. Newmark time integration method is used in this study with parameters γ =0.7 and β=0.36 addingl numerical damping that reduces high frequency motions that were introduced by SW4. In order to capture the wave propagation in FEM model, mesh size should be controlled so that there is no artificial filtering to motions above certain frequency (Watanabe et al., 2017). As pointed out by Hughes (1987), 10 linear interpolation finite elements and 2 quadratic interpolation elements are needed per wave wavelength. Since second order 27 node brick element are used here, the minimum wave length captured is 6 meters. Considering shear wave velocity $v_s = 500 m/s$, the maximum frequency calculated by equation 504.7 is 83 Hz. Even when material plasticifies (becomes softer), model is still propagating high frequencies of up to required $f_{max} \leq$ 15Hz.

$$f_{max} = v_s/\lambda_{min} \tag{504.7}$$

504.7.4.1 Embedded Nuclear Structure

Small Modular Reactor (SMR) analyzed here is a 4 storied reinforced concrete structure with total height 50 meters and 36 meters embedded in the ground. The length and width of the structure is 30 meters. The whole structure is modeled using 27-node solid brick element with linear elastic material. The Young's modulus is selected as $E = 30\text{GPa}$ and Poisson's ratio $\nu = 0.2$. Single layer of 27 node bricks can accurately model (beam and plate) bending, and is hence chosen for structural model.

504.7.4.2 Soil Model

The depth of the model surrounding the structure is 45m. The soil is assumed to be saturated soil with undrained behavior during the earthquake. In order to considering nonlinear site effects, the soil is modeled with elastoplastic material. In the past 20 years, many 3D constitutive models Yang et al. (2003); Dafalias and Manzari (2004b); Park and Byrne (2004); Pisanò and Jeremić (2014) have been developed. Undrained behavior can be approximately modeled using von Mises material model (Yang and Jeremić, 2003). Elastic plastic von Mises material model with linear kinematic hardening rule is used here. The material parameters are presented in table 504.5. Backward Euler implicit algorithm (Jeremić and Sture (1997)) is used for the equilibrium iterations at constitutive level.

504.7.4.3 Soft Contact Element

Model for contact/interface (axial contact, gap opening and closing and slip behavior) of the interface between structure and surrounding soil, relies on a node-to-node soft contact/interface element (Sinha and Jeremić (2017)). In soft contact, the normal stiffness exponentially grows as the relative displacement between two contact/interface nodes increases and finally reaches maximum normal stiffness. 3,177 contact elements are placed at the soil-structure interface. Contact/Interface parameters are shown in Table 504.5.

504.7.4.4 Simulation Procedure

The nonlinear ESSI analysis was conducted using Real-ESSI Simulator (Jeremić et al. (1988-2025)) developed at UC Davis and LBNL. Two SMR simulation models were simulated. First model uses linear elastic soil without contact element and second model uses inelastic/nonlinear soil with inelastic/nonlinear contact. In both cases, two loading stages were modeled: First loading stage is a self weight, developed by adding a uniform gravity field. This is a necessary stage for inelastic analysis in order to develop initial stress state of structure and surrounding soil before earthquake comes. Self weight was also applied to the elastic model, just so that we have comparable displacement results. Then second loading stage

Table 504.5: Modeling parameters

Material parameters	shear wave velocity [m/s]	500
	Young's modulus [GPa]	1.25
	Poisson ratio	0.25
	von Mises radius [kPa]	60
	kinematic hardening rate [MPa]	0
Contact parameters	initial normal stiffness [N/m]	1e9
	hardening rate [/m]	1000
	maximum normal stiffness [N/m]	1e12
	tangential stiffness [N/m]	1e7
	normal damping [N/(m/s)]	100
	tangential damping [N/(m/s)]	100
	friction ratio	0.25
Damping parameters	structure layer	5%
	surrounding soil	15%
	DRM layer	20%
	outside layer 1	20%
	outside layer 2	40%
	outside layer 3	60%

is an earthquake load. Simulations were performed on a local parallel computer with parallel version of the Real-ESSI Simulator using 10 CPUs. It is noted that Real-ESSI Simulator is also available on Amazon and Google cloud parallel computers as well as on large national parallel compters at the LBNLf (EDISON and CORI). Local parallel computer (a workstation) was used in order to illustrate versatility of Real-ESSI Simulator and to show that high fidelity parallel computations do not need to require high price parallel computers.

504.7.5 Simulation Results

Figure 504.89 shows time series acceleration response of top center of SMR.

The elastic results represent simulation case where the surrounding soil is modeled using linear elastic material and no contact elements in soil-structure interface. The inelastic results represent simulation case where the surrounding soil is modeled using inelastic soil material and an inelastic contact. Signifi-

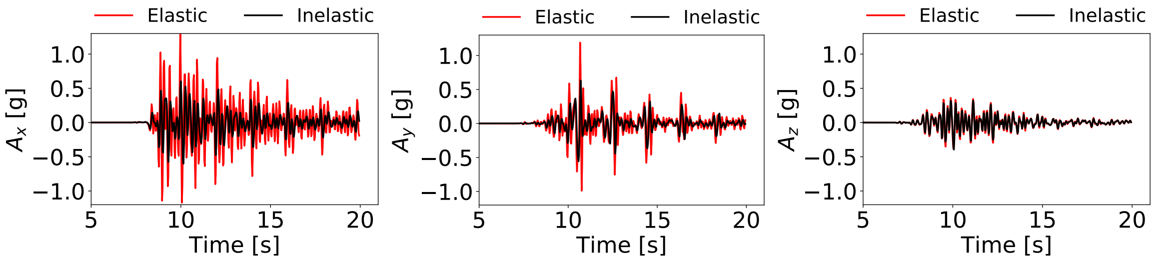


Figure 504.89: Time Series Acceleration Response.

cant acceleration decreases can be seen in the inelastic case. The horizontal peak acceleration values is reduced by almost 40%. This is due to plastification of soil and inelastic response of the contact/interface zone. Moreover significant seismic energy is dissipated, so that inelastic soil and inelastic contact/interface act as seismic dissipators and isolators during seismic event. The acceleration difference in vertical, z direction is less significant than horizontal direction.

Figure 504.90 shows Fourier magnitude that high frequency component of horizontal acceleration was significant decreased in inelastic case.

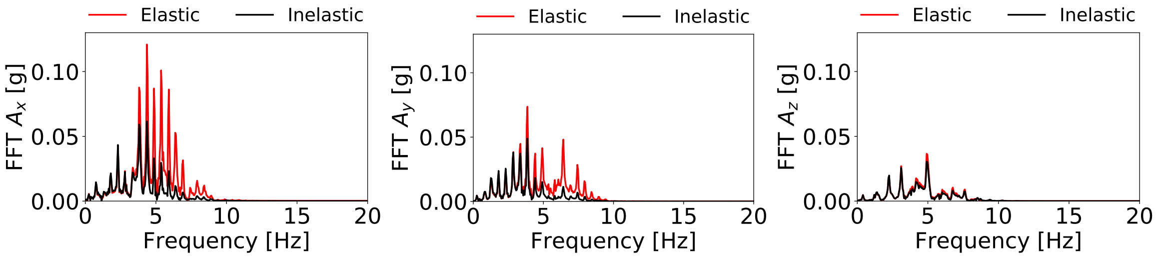


Figure 504.90: Acceleration Response in frequency domain

Material Modeling Influence on Results is Shown in Figure 504.91.

SMR: Variation of Acceleration with Depth is Shown in Figures 504.92 and 504.93.

SMR: PGA and PGD Depth Variation is Shown in Figure 504.94.

- The PGA & PGD of SSI systems are (very) different from free field motions,

- Material nonlinearity has significant effect on acceleration response.

SMR: Elastic vs Inelastic response is shown in Figure 504.95.

Figure 504.96 shows the distribution of plastic strain in surrounding soil. There are two main plastic zones near two bottom corners of the structure. Also the plastic strain at the soil-structure interface is higher than adjacent area. It is interesting to note that there is an elastic zone beneath structure. The

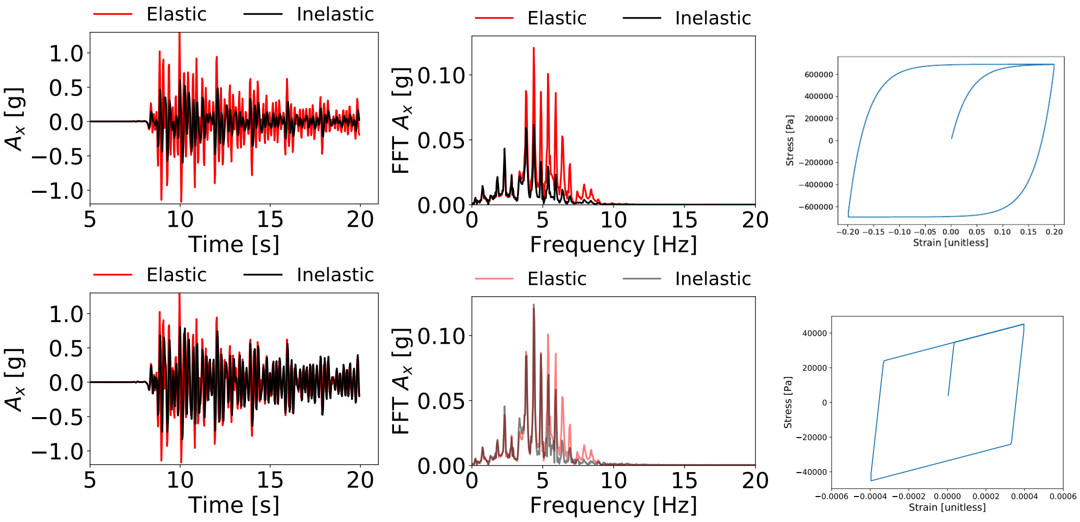


Figure 504.91: Upper: Material A: nonlinear, vM - AF; Lower: Material B: Bilinear.

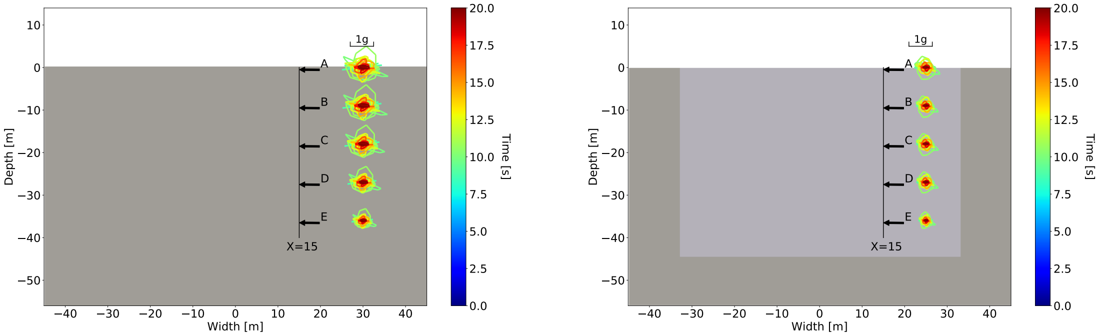


Figure 504.92: Free field response, left: elastic, right, inelastic.

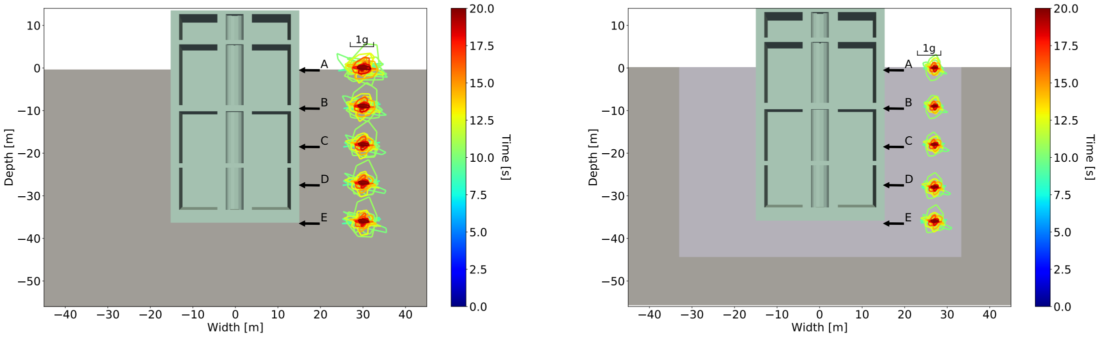


Figure 504.93: ESSI for an SMR response, left: elastic, right, inelastic.

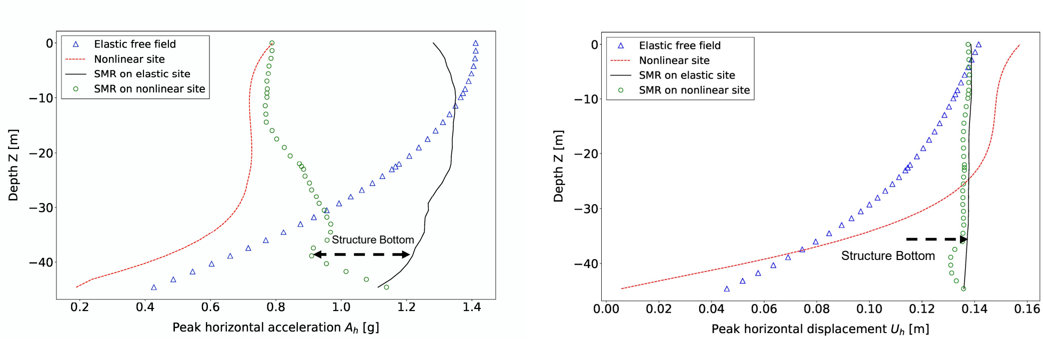


Figure 504.94: SMR: PGA and PGD variation with depth.

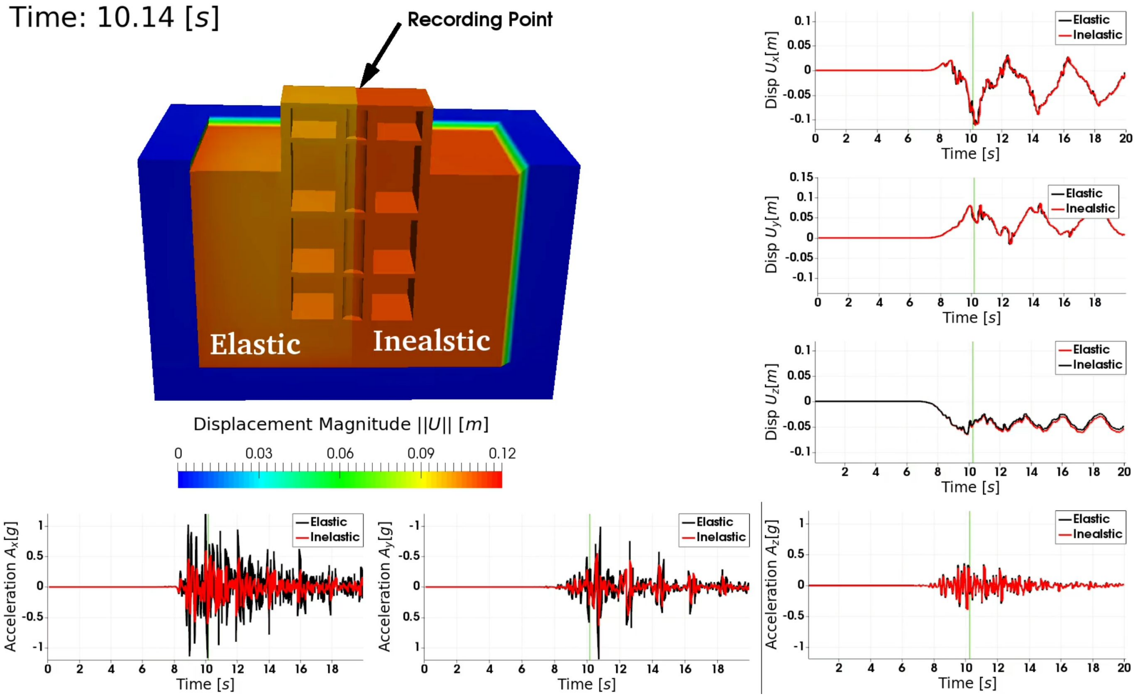


Figure 504.95: Animation of a linear elastic vs inelastic response of an SMR.

shape of the elastic zone resembles a bulb and is due to self weight and stiffness of the structure that compresses soil beneath

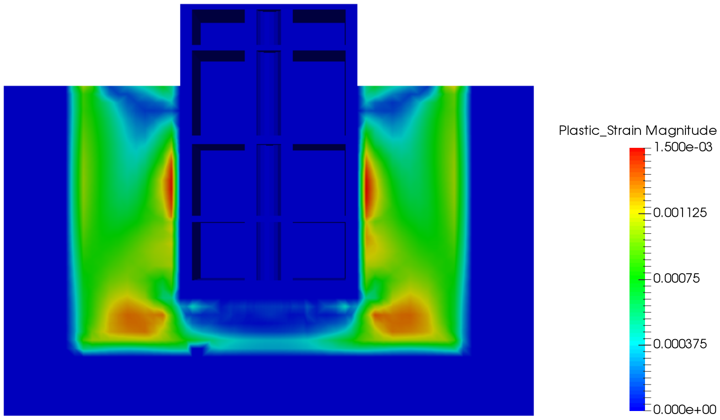


Figure 504.96: Distribution of the magnitude of plastic strain

504.7.6 Energy Dissipation

According to the thermodynamics framework presented by Yang et al. (2019a), the energy dissipation in any decoupled material undergoing isothermal process can be expressed as:

$$\Phi = \sigma_{ij}\dot{\epsilon}_{ij} - \sigma_{ij}\dot{\epsilon}^{el}_{ij} - \rho\dot{\psi}_{pl} \geq 0 \tag{504.8}$$

where Φ is the rate of change of energy dissipation per unit volume (or dissipation density), σ_{ij} and ϵ_{ij} are the stress and strain tensors respectively, ϵ^{el}_{ij} is the elastic part of the strain tensor, ρ is the mass density of the material, and ψ_{pl} is the plastic free energy per unit volume (or plastic free energy density). Equation 504.8 ensures the energy balance and nonnegative energy dissipation conditions that correspond to the first and second law of thermodynamics.

With Equation 504.8, the energy balance of a SSI system is simply given by:

$$W_{Input} = E_{Stored} + E_{Dissipated} = KE + SE + PF + PD \tag{504.9}$$

where W_{Input} is the input work due to external loading, KE is the kinetic energy, SE is the elastic strain energy, PF is the plastic free energy, and PD is the energy dissipation due to material plasticity. Formulation for each energy component can be found in Yang et al. (2018). Note that in Equation 504.9, it is assumed that no other forms of energy dissipation exists in the system.

Figure 504.97 (a) shows the distribution of plastic dissipation density in the SMR model at the end of simulation. The case presented in this section is elastic plastic soil without contact/interface element. Note that the structure is modeled with elastic material, so they do not dissipate any energy. As expected, more seismic energy is dissipated around the corners and edges of the structure due to stress concentration. It can be observed that there are several elastic regions around the boundaries of the structure, which means that the soil there does not plastify much and moves together with the structure. Economy of the design can be improved by better utilizing the strength of soil around these locations.

Figure 504.97 (b) show the evolution of energy components at location A. It can be observed that the amount of plastic energy dissipation is much larger than the other forms of energy, indicating that the nonlinear effect is quite significant in deeply embedded structure. Another interesting observation is the small amount of plastic free energy whose quantity largely depends on material hardening parameters and loading conditions. It should be pointed out that even if it is small, plastic free energy should never be neglected so that the condition of nonnegative incremental energy dissipation can be upheld Rosakis et al. (2000); Taylor and Quinney (1934).

SMR: Energy Dissipation for an SMR, Figure 504.98.

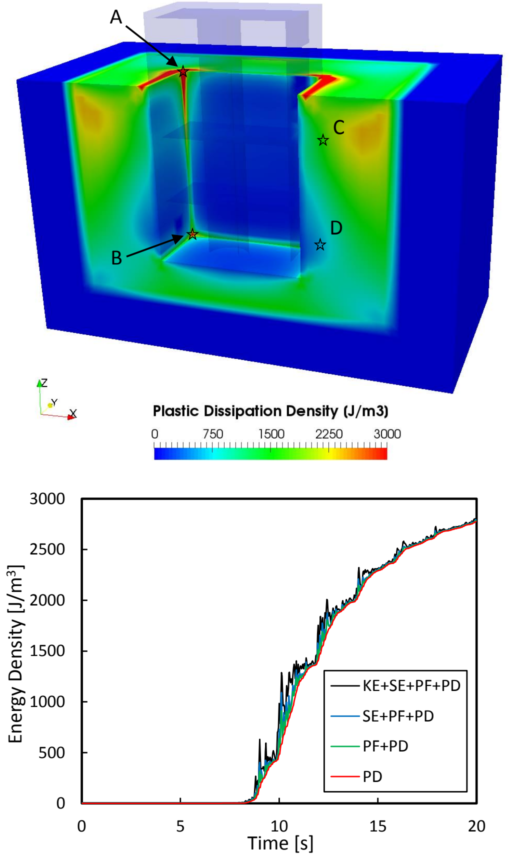


Figure 504.97: Energy dissipation in SMR model: (a) Plastic dissipation density field at the end of simulation; (b) Evolution of energy components at location A.

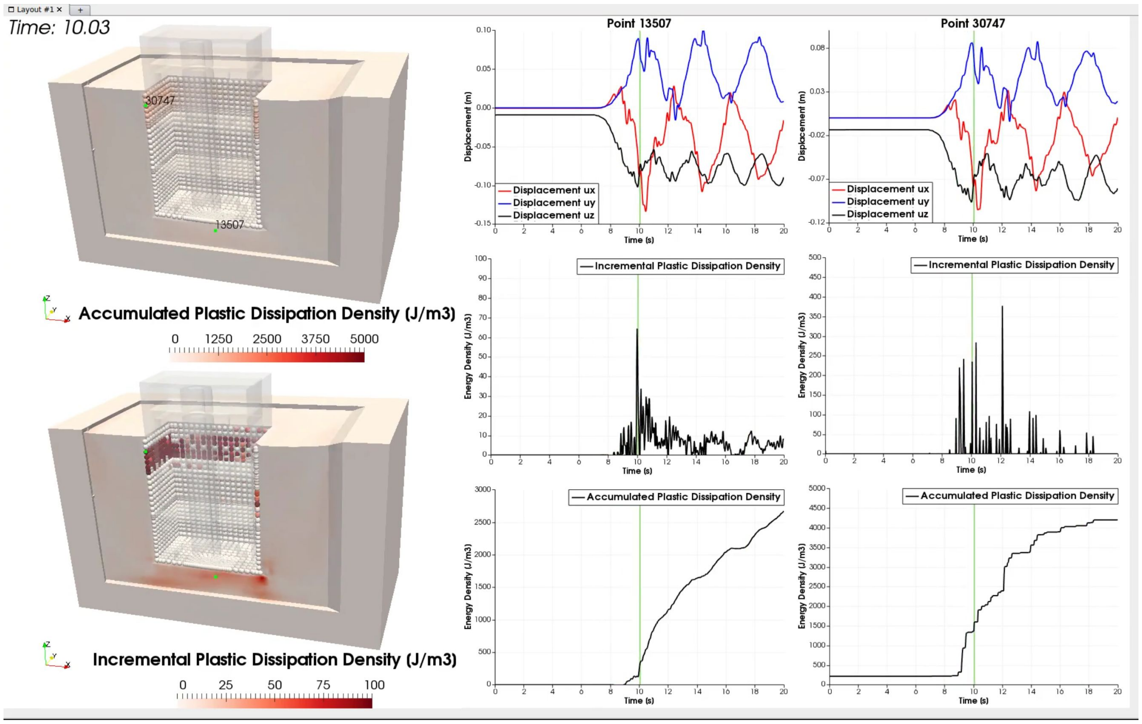


Figure 504.98: Animation of energy dissipation for an SMR.

504.7.7 SMR Inelastic Modeling Conclusions

The seismic response of an embedded SMR has been modeled with high fidelity. Using state-of-the-art nonlinear ESSI simulation techniques, many modeling uncertainties have been eliminated. The methodology shown here is also applicable to many other ESSI problems (buildings, bridges, dams, etc.). The simulation result of SMR shows that the acceleration response of the structure decreases with nonlinear effects properly modeled. In addition, the high frequency component of acceleration is significantly damped out in inelastic case due to soil plastification.

Energy dissipation analysis shows that the soil close to the edge of the SMR structure dissipates large amount of seismic energy during shaking. Such observation also indicates significant nonlinear effect when elastoplastic material is used for soil modeling. Several elastic regions are identified where design can be improved so that soil strength at these locations can contribute to the safety of the SSI system.

504.8 Inclined Waves, Free Field and SMR Modeling

Methodology developed by Wang et al. (2020a) and presented in section 109.2.5 on page 557 is illustrated here. This is really a repeat of results from Wang et al. (2020a).

Input files for these models are available HERE.

Presented WPF-DRM method is implemented in the Real-ESSI Simulator (Jeremić et al., 1988-2025). Described examples and publicly available executables for the Real ESSI sequential and parallel programs are available through Real ESSI Simulator web site `http://real-essi.info/`. All numerical examples presented here are analyzed using Real-ESSI Simulator version 20.01, in parallel computing mode, on UC Davis and Amazon Web Services parallel computers.

504.8.1 Free Field Modeling and Verification

Free field response of layered ground excited by an inclined incident seismic wave is used to illustrate and verify developed methodology. Analytic solutions based on Thomson-Haskell propagation matrix technique (Thomson, 1950; Haskell, 1953; Silva, 1976) are used for verification.

A finite element model for the free field, that is 300m wide and 200m deep, consisting of three layers, as described in Table 504.6, is used.

It is noted that dimension of analyzed model is 300m × 200m, however there exist additional finite elements outside this domain: DRM layer is a single layer of finite elements that surround the interior domain. Beside the DRM layer, there are absorbing layers consisting of multiple layers of finite elements with high viscous damping. These damping layers should be thick enough to absorb the outgoing waves.

Table 504.6: Properties of layers: thickness d, density ρ, shear wave velocity V_s, compressional wave velocity V_p and Poisson's ratio ν.

Layer	d [m]	ρ [kg/m^3]	V_s [m/s]	V_p [m/s]	ν
1	50	2100	500	816.5	0.2
2	100	2300	750	1403.1	0.3
3	∞	2500	1000	2081.7	0.35

The thickness and damping parameters of these absorbing layers are determined such that the response of exterior damping layer given by earthquake soil structure interaction analysis is negligible compared to the inner part. A fixed boundary condition is applied to the outer boundary. It is also noted that theoretically there should be no waves propagating outside of the DRM layer for a free field response. Additional damping layers are added in order to accommodate further, non-free field model expansions and additions. Finite element size is set to 5m, and with 10 finite elements per wave length, this mesh can accurately propagate waves of up to $f = 10\text{Hz}$, for surface soil with shear wave velocity of $V_s = 500\text{m/s}^2$, as per Lysmer and Kuhlemeyer (1969); Watanabe et al. (2017).

A number of monochromatic, single frequency plane SV wave, represented by a cosine function, with variable inclinations $\theta = 10^o, 45^o, 60^o, 80^o$ and variable frequencies, $f = 1.0, 2.5, 5.0, 10.0\text{Hz}$, are applied to the layered ground model using developed methodology. The incident SV wave magnitude from the depth is 0.06m and is kept the same for all the analyzed cases. It is noted that inclination angle θ is measured between a wave propagation direction vector and vertical axes. The wave inclination θ depends on many factors, e.g., source focal mechanism and radiation pattern, wave propagation path and local site geology and topography. The typical range of inclination is $0^\circ \sim 40^\circ$ (Tabatabaie et al., 1986; Sigaki et al., 2000). For example, Tabatabaie et al. (1986) estimated that the incidence angle of shear waves at the SMART-1 array site is around 20 degrees using the recorded motions from 1981 Taiwan earthquake. For far-field, flat engineering site with large impedance contrast (e.g., soft soil overlying stiff bedrock), the assumption of vertical wave propagation can be adopted due to very small inclination of incident seismic waves. However, for near-field, hard rock site with low impedance contrast or engineering site with significant topography, incidence angle of seismic waves tends to be large and inclined wave propagation should be carefully modeled.

Free field motions are developed and introduced into the model through WPF-DRM. Figure 504.99 shows snapshots of wave displacements in the model, for a wave frequency of $f = 5\text{Hz}$, for different input plane wave inclinations, $\theta = 10^o, 45^o, 60^o, 80^o$. Figure 504.100 shows snapshots of wave displacements in the model, for a wave that is inclined at $\theta = 60^o$, for variable input plane wave frequencies $f = 1.0, 2.5, 5.0, 10.0\text{Hz}$.

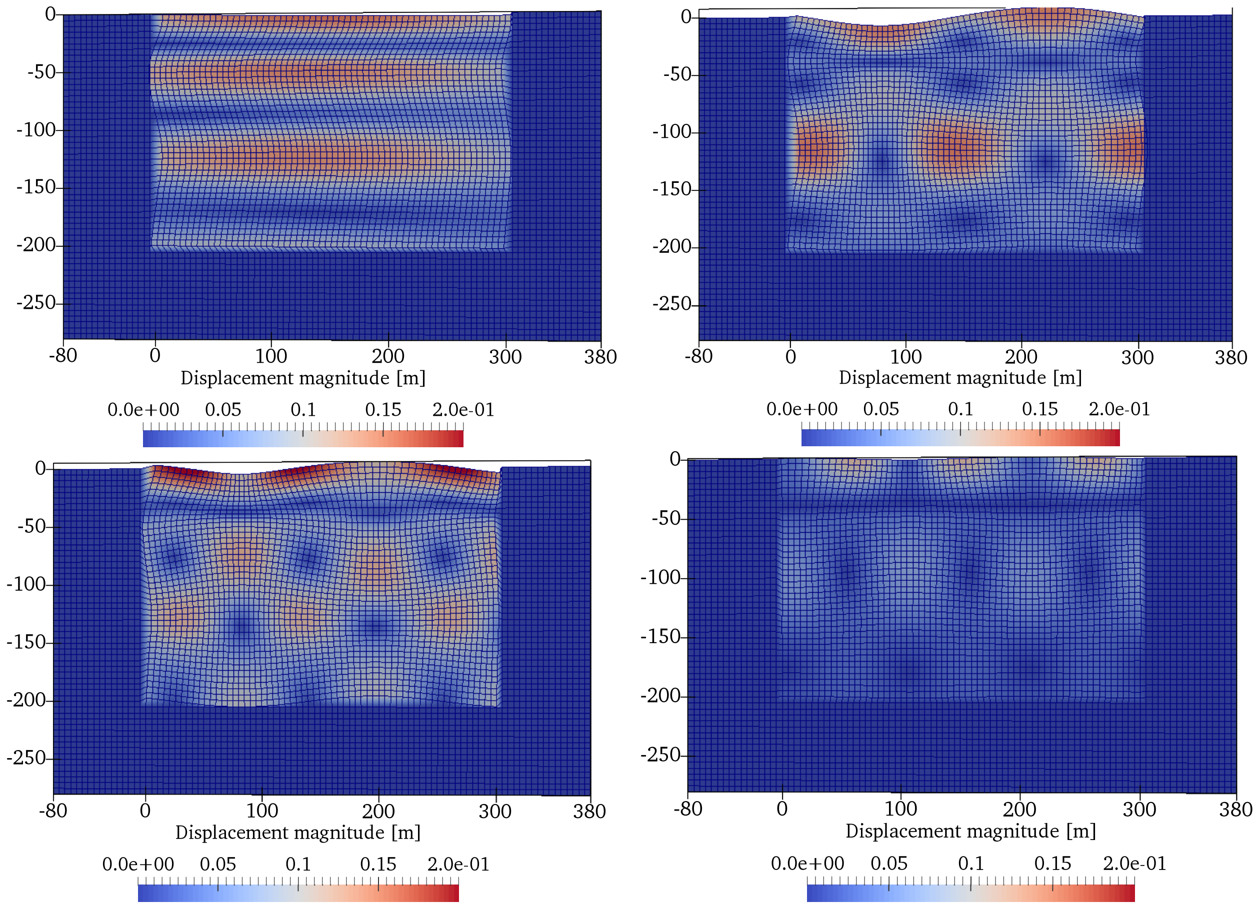


Figure 504.99: Displacement magnitudes for a free field response under incident SV wave, frequency $f = 5$Hz, with different incident wave inclinations: (a) $\theta = 10^\circ$ (b) $\theta = 45^\circ$ (c) $\theta = 60^\circ$ (d) incident angle $\theta = 80^\circ$.

Few notes are in order upon visual inspection of results in Figures 504.99 and 504.100. The outgoing waves in exterior region, outside DRM layer, are negligibly small, almost zero for all the cases. This is indeed expected, as it follows from the theory of the domain reduction method (Bielak et al., 2003a; Yoshimura et al., 2003a), whereby the so called residual field (w_e) should be non-existent for free field motions, that were used to develop effective DRM forces.

Comparing free field responses for SV wave with different incident angles, Figure 504.99, the $\theta = 10^\circ$ case behaves very similar to 1D vertically propagating motion field that is commonly used in engineering practice. It is noted, however that there are still vertical motions at the surface due to such almost vertical SV wave interacting with the free surface. For cases where wave inclination is more significant, for $\theta = 45^\circ$ and $\theta = 60^\circ$, significant surface motions are observed, with pronounced vertical and horizontal motions. When the incident wave inclination is $\theta = 80^\circ$, seismic wave propagates almost horizontally

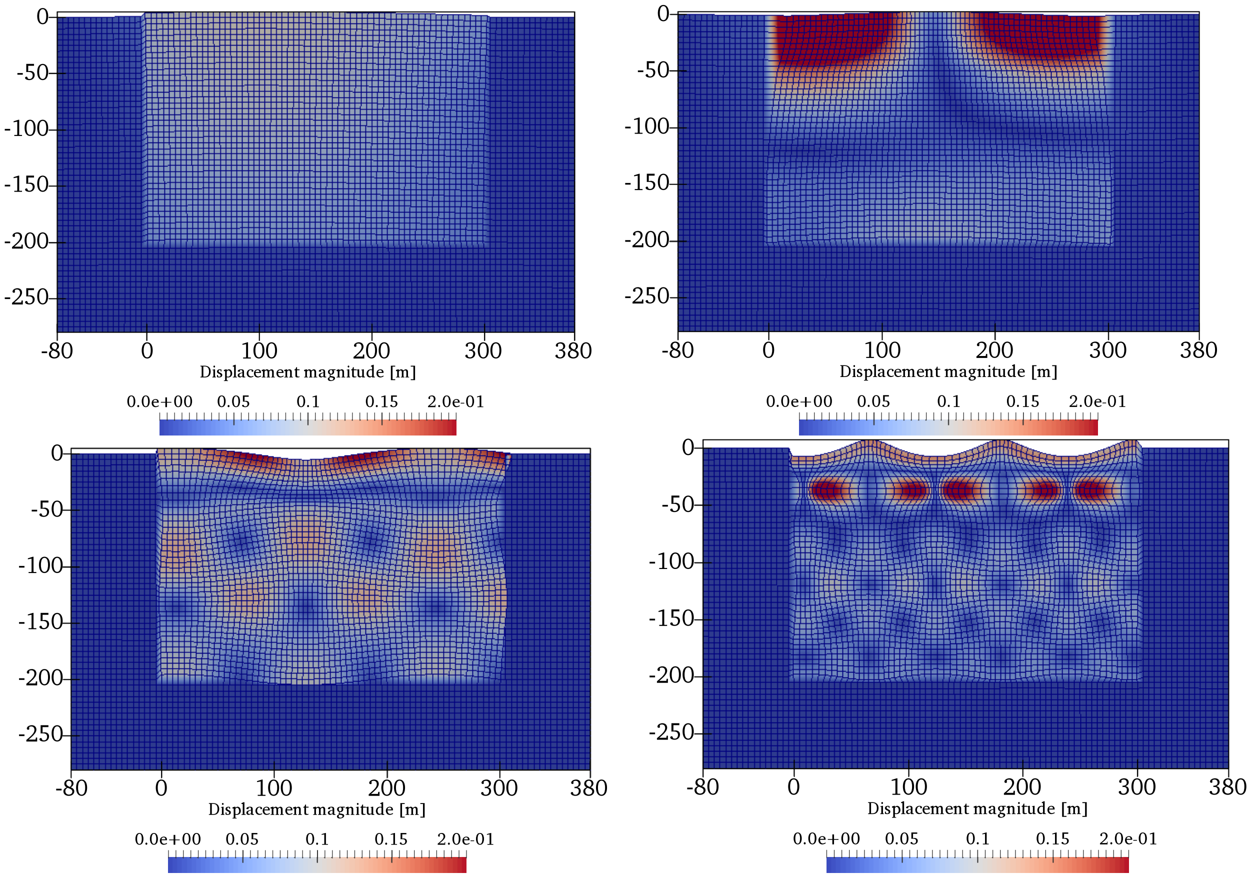


Figure 504.100: Displacement magnitudes for a free field response under incident SV wave at an angle of $\theta = 60^o$, with different frequencies: (a) $f = 1.0$Hz (b) $f = 2.5$Hz (c) $f = 5.0$Hz (d) $f = 10.0$Hz.

without generating significant surface motions. It is also noted that the displacement magnitude of the seismic wave field for wave inclination case $\theta = 80^{\circ}$ is much smaller than for the other cases. This is reasonable considering the site amplification for other free field cases comes, in part, from the impedance contrast of vertical wave propagation.

Results, snapshots of displacement field magnitudes for wave fields of different frequencies are shown in Figure 504.100 for seismic motion inclined SV wave field at $\theta = 60^{\circ}$. It is noted that layer boundaries, impedance contrasts, are at -50m, and at -150m. Those layer boundaries can be visually identified from distribution of waves through model depth with positive and negative interference reflected and refracted waves within different layers of the domain.

Figures 504.101 and 504.102 compare simulated free field horizontal and vertical displacement magnitudes against corresponding analytical solutions along the depth. It is noted that acceleration magnitudes can be obtained by multiplying displacement magnitudes with w^2. Very good agreement is observed

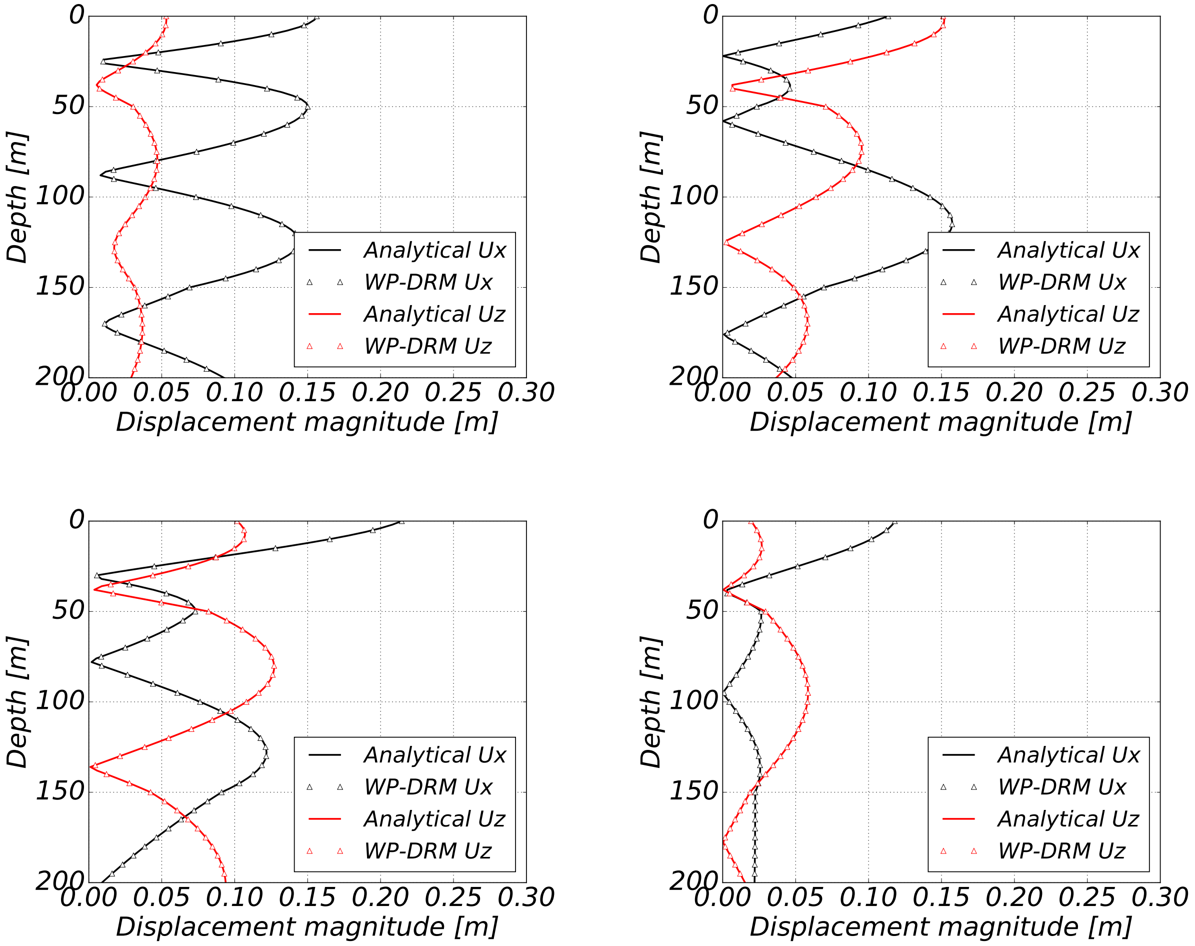


Figure 504.101: Verification of free field modeling under incident SV wave $f = 5Hz$ with different incident angles θ: (a) $\theta = 10°$, (b) $\theta = 45°$, (c) $\theta = 60°$ (d) $\theta = 80°$.

between results given by WPF-DRM simulation and analytical solutions. Several interesting observations can also be made:

1. Along with the increase in frequencies, the vertical wave length becomes shorter, and that results in more wave interferences along the depth.
2. The existence of layers and interfaces at $z = -50$m and $z = -150$m complicates the spatial variation of wave field along the depth, especially for higher frequencies, $f = 5Hz$ and $10Hz$. The response curves at depths $0 \sim 50$m and $50 \sim 150$m are quite different in both amplitude and variation pattern.

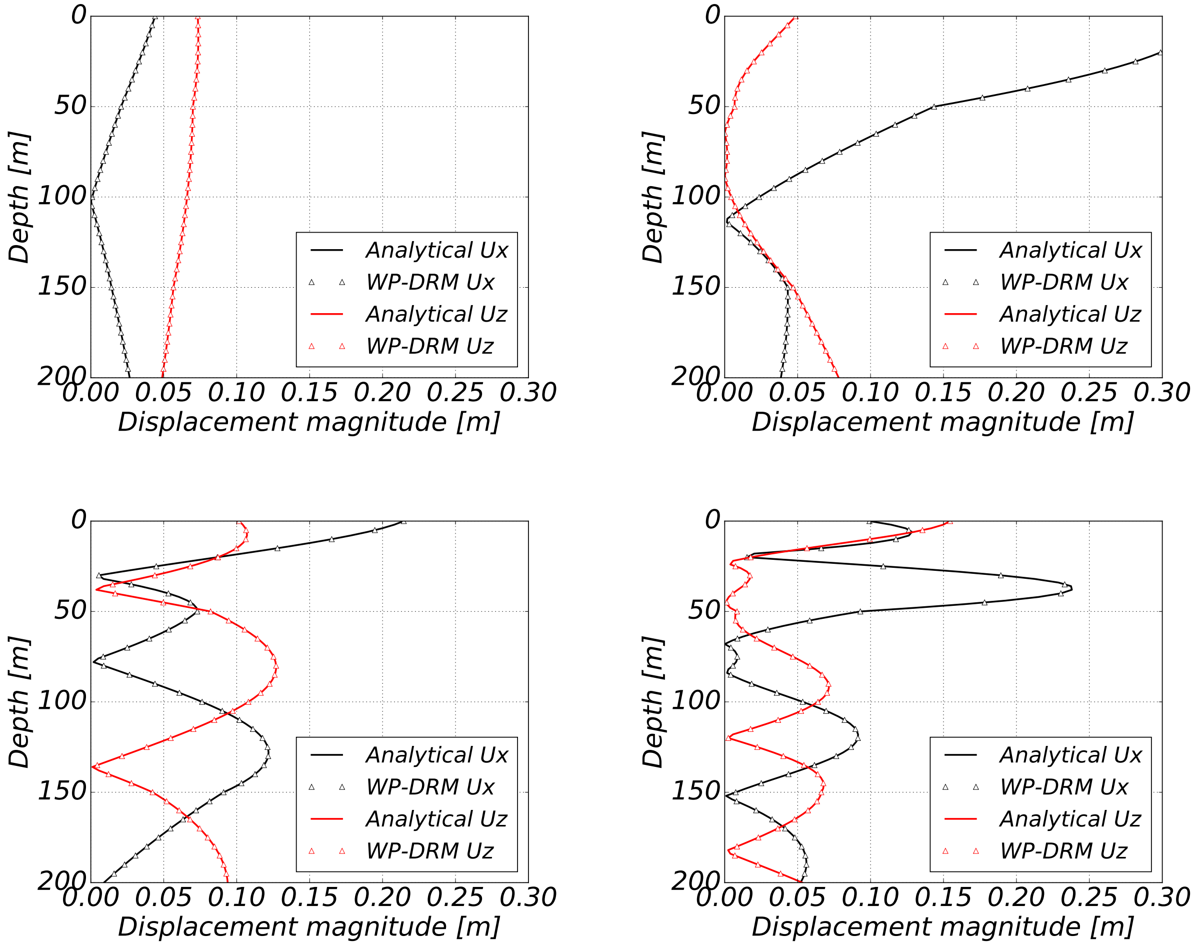


Figure 504.102: Verification of free field modeling under incident $\theta = 60^o$ SV wave with different frequencies f: (a) $f = 1.0$Hz, (b) $f = 2.5$Hz, (c) $f = 5.0$Hz, (d) $f = 10.0$Hz.

3. From Fig. 504.101, it can be seen that inclination angle of input SV wave also plays a crucial role in the interference characteristic of inclined wave field. Periodic peaks and troughs shown in the case of 10° inclination are typical interference characteristics of 1D homogeneous, vertically propagating wave field. However, the interference characteristics given by other wave inclinations show significant differences. These different variation patterns along the depth, that might not make much difference for shallow founded surface structures, can result in very different seismic response for deeply embedded structures.

504.8.2 Deeply Embedded Soil-Structure Model

Deeply embedded structural model, a model of a Small Modular Reactor (SMR) is analyzed and used to illustrate developed methodology. The FEM model of an SMR structure embedded in layered ground is shown in Figure 504.103(a). The embedment depth is 36m, while the height of SMR structure above ground is 14m. The structure width is 30m and the whole model width of SSI system is 150m. It is noted that the lateral extent of soil domain should be large enough such that the outgoing waves passing through DRM layer are insignificant. These waves can then be damped out through the absorbing layer and would have negligible influence on the dynamic response of SMR. Key factors to determine the model width of SSI system include structural width, intensity of seismic excitations, etc. Detailed discussions regarding the required lateral extent of soil domain for dynamic SSI analysis can be found in Sharma et al. (2020). Eleven representative points, point A to point K in Figure 504.103(b), are selected to monitor the dynamic response of SMR. The layered ground parameters are the same as those used in free field study given in Table 504.6.

To proper model wave propagation, the finite element size and time step should be carefully controlled to reduce discretization errors. For linear displacement approximation within finite element, in this case eight-node brick elements, at least 10 nodes per wavelength should be used (Watanabe et al., 2017). The time step length Δt is limited by Courant-Friedrichs-Lewy condition (Courant and Hilbert, 1989) for stability. In addition, following requirement needs to be met to accurately capture the propagation of wave front (Jeremić et al., 2009), where Δh is the mesh size and v is the highest wave velocity.

$$\Delta t < \frac{\Delta h}{v} \tag{504.10}$$

In this study, eight-node brick element with 4m mesh size is used for spatial discretization. The maximum frequency the model can propagate is about $12.5Hz$ considering the minimum elastic shear wave velocity $500m/s$. Time step is chosen as $\Delta t = 0.005$s. Newmark time integration method (Newmark, 1959) is used with Newmark parameters $\gamma = 0.505$ and $\beta = 0.25(0.5 + \gamma)^2$. Since parameter $\gamma > 0.5$, a small amount of numerical, algorithmic damping is introduced to damp out unrealistic high frequency responses from spatial discretization (Argyris and Mlejnek, 1991). See Yang et al. (2019b,c) for more information about the proper selection of Newmark parameters for dynamic analysis. Gradually increasing Rayleigh damping (7%, 15% and 30%) is assigned to the inner, middle and exterior part of the absorbing layers, outside of the DRM layer, to prevent reflection of radiated outgoing waves (Jeremić et al., 2009; Abell et al., 2018). These damping values are determined such that after dynamic SSI analysis the response of the exterior absorbing layer is negligible compared to the inner part.

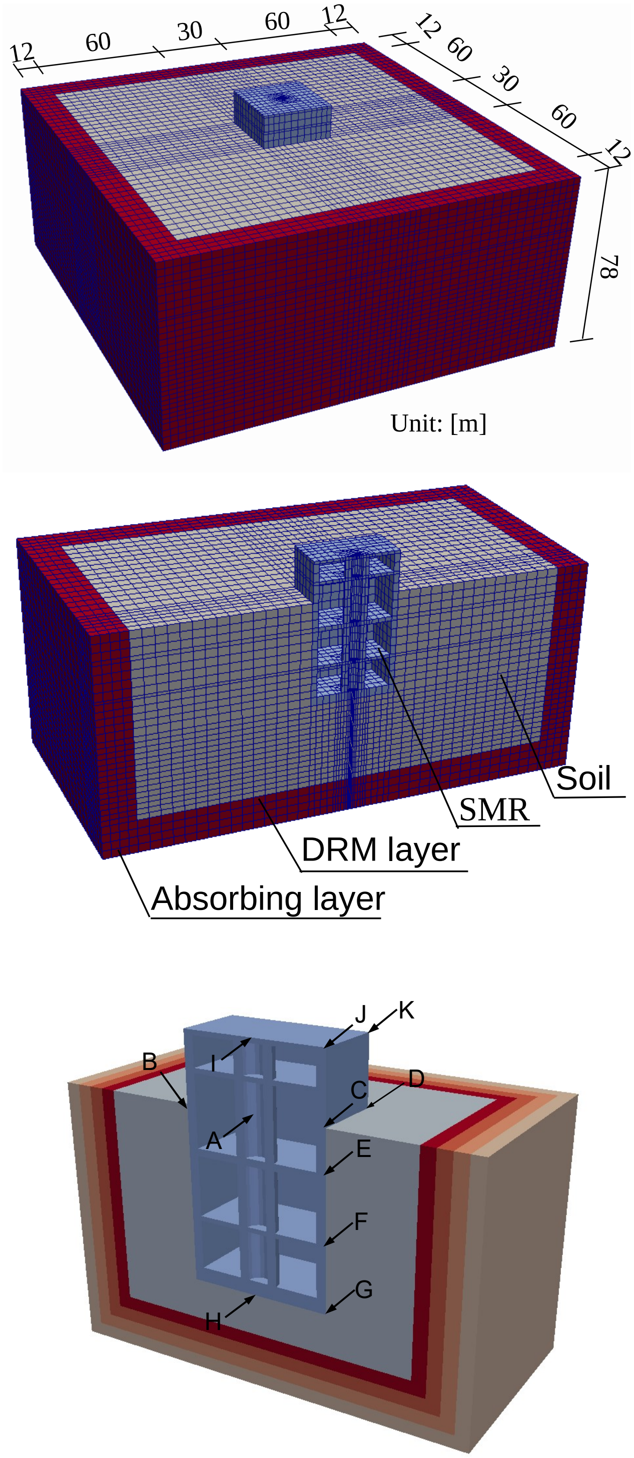


Figure 504.103: FEM model of embedded SMR and representative points.

504.8.3 SMR Excited with Inclined SV Waves

Deeply embedded SMR structure is excited with inclined plane waves, at inclination angles of $\theta = 10^\circ, 45^\circ, 60^\circ$ and 80°. Seismic wave frequency used for this set of numerical test was set at $f = 5Hz$. As described in table 504.6 on page 2544, shear wave velocities of top 50m layer is $V_s = 500$m/s while the lower layer is 100m think and has a shear wave velocity of $V_s = 750$m/s. Due to presence of layers, seismic wave field close to the surface is made up Rayleigh and Stoneley waves (Aki and Richards, 2002; Semblat and Pecker, 2009). It might thus be difficult to separate influence of these different surface waves the response of the SMR. For example, in Figure 504.99 on page 2545, that shows displacement magnitudes at certain time, for different inclination of incident plane wave, Stoneley wave is apparent close to depth of 50m. In addition, Rayleigh wave is also apparent close to free field surface. Those wave fields, when applied to the SMR SSI system, produce response, at location of point A[2] on SMR structure, as shown in Figures 504.104 and 504.105.

It is noted that corresponding free field motions at the same location are also plotted for comparison. Variations of displacement magnitudes caused by different inclinations of incident SV wave are quite noticeable for vertical displacements and accelerations, while influence on horizontal displacements and accelerations is much less significant. The reduction of vertical displacement and accelerations that is observed in all the four cases, is consistent with the concept of "base averaging", "ironing out" of seismic motions by Housner (1957). The most significant reduction occurs for the case of incident wave at an angle $\theta = 45^\circ$ while little reduction is seen in the case of $\theta = 80^\circ$.

The deformed shapes of SMR at $t = 0.4s$ for four scenarios are shown in Figure 504.106. In the cases of seismic waves at inclinations $\theta = 45^\circ$ and $\theta = 60^\circ$, rocking responses of SMR are quite evident when compared with the cases of almost vertical wave propagation ($\theta = 10^\circ$) and almost horizontal wave propagation ($\theta = 80^\circ$).

---

[2] Location of point A is in the middle of SMR structure, where center of the free field model would be, please see Figure 504.103 on page 2550.

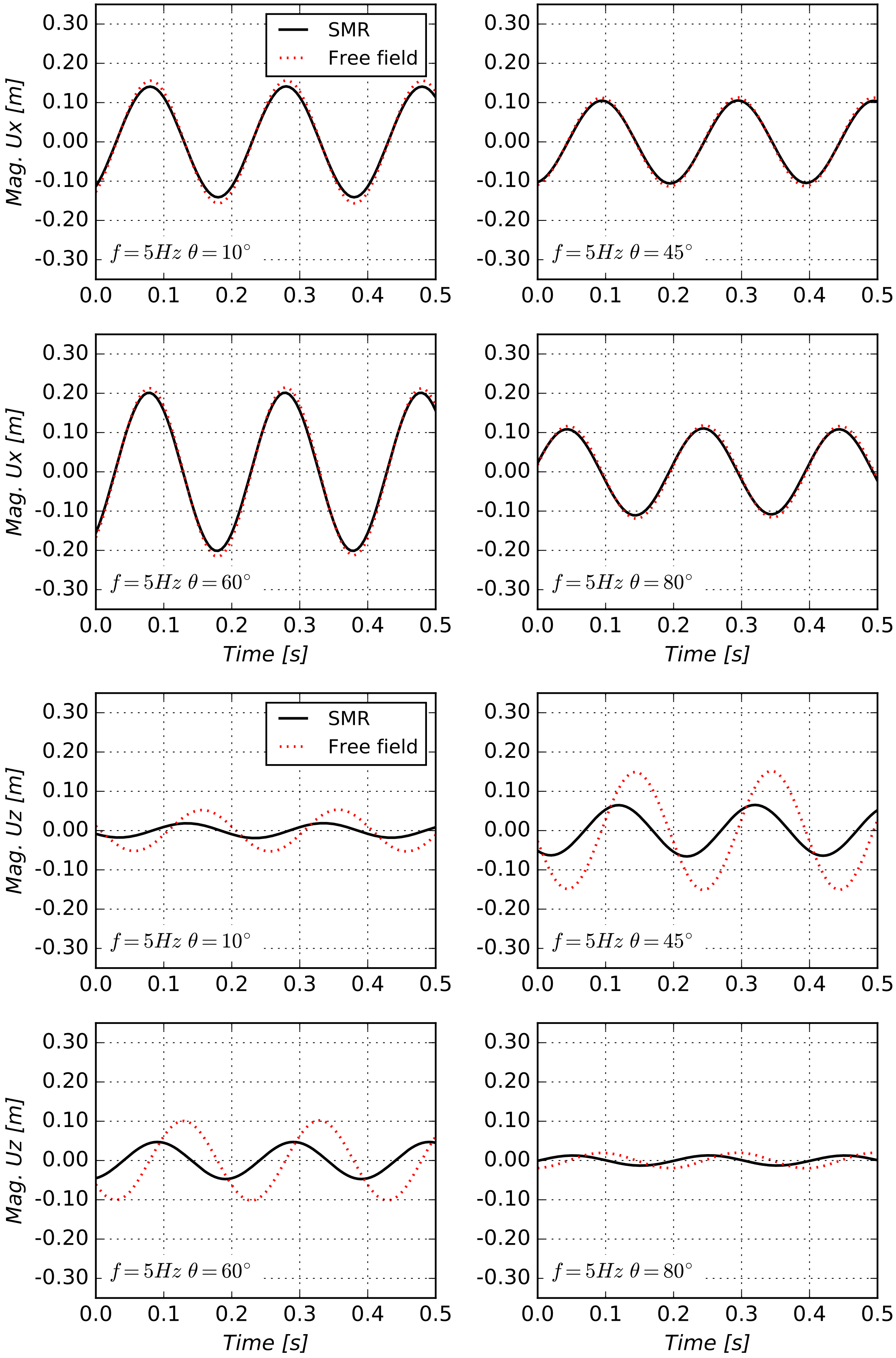


Figure 504.104: Displacement response of point A within embedded SMR, excited by an inclined SV wave with $f = 5Hz$ and different inclination angles, $\theta = 10^\circ, 45^\circ, 60^\circ$ and 80°: (a) horizontal displacement (b) vertical displacement.

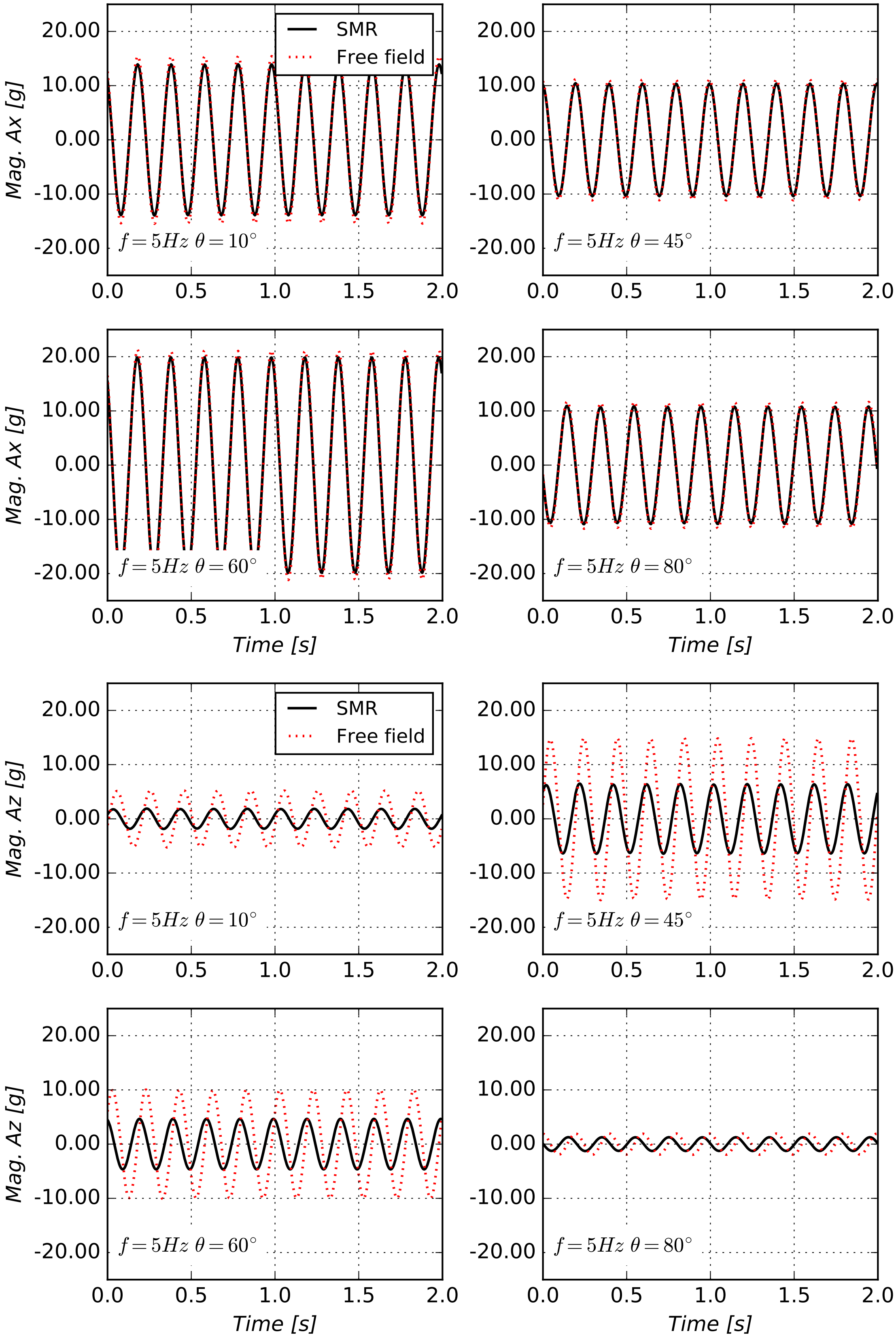


Figure 504.105: Acceleration response of point A within embedded SMR, excited by an inclined SV wave with $f = 5Hz$ and different inclination angles, $\theta = 10°, 45°, 60°$ and $80°$: (a) horizontal acceleration (b) vertical acceleration.

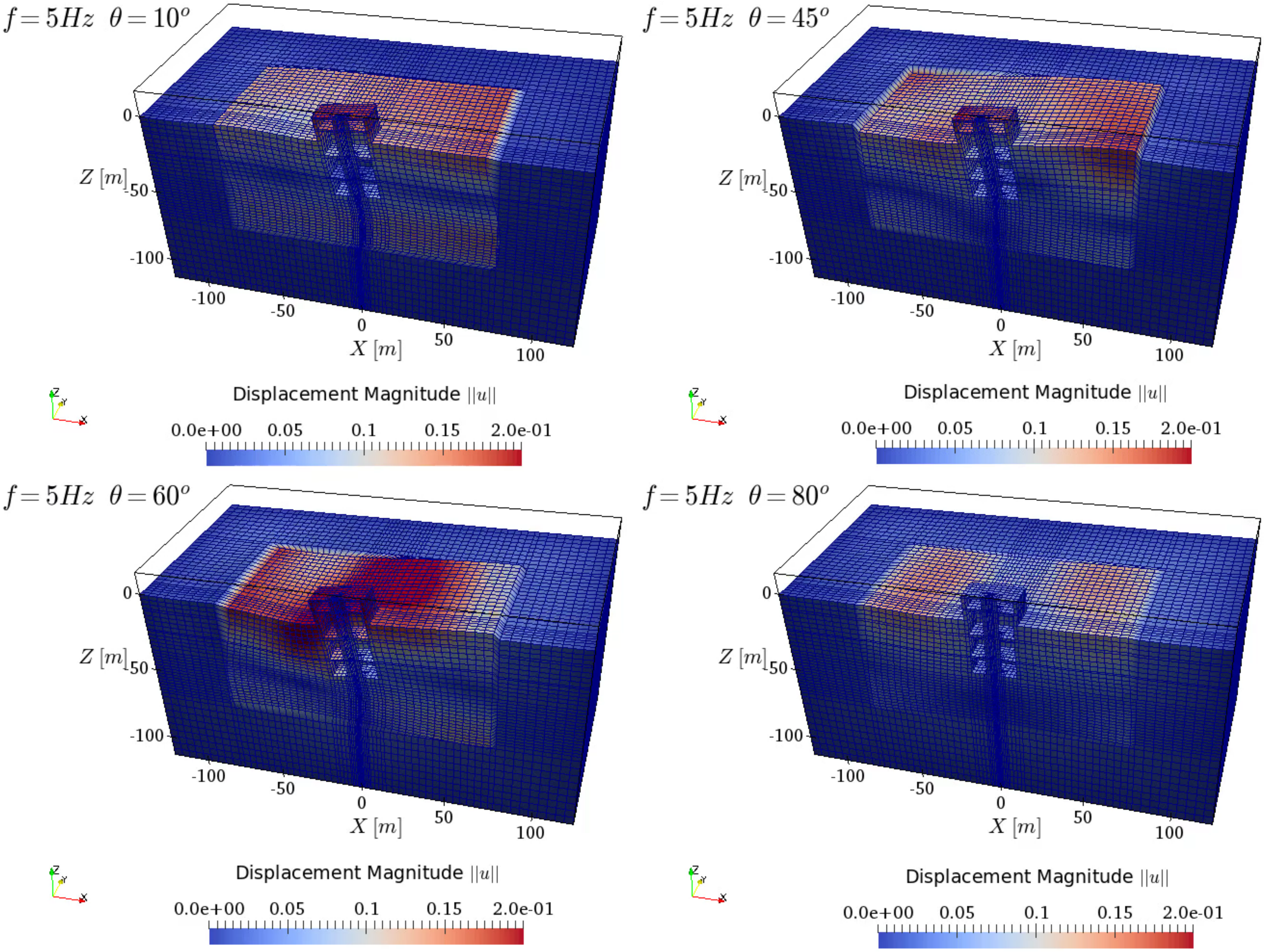


Figure 504.106: The deformed shapes of SMR at $t = 0.4s$ for incident SV wave at different inclinations $\theta = 10^\circ, 45^\circ, 60^\circ$ and 80°.

504.8.4 SMR Excited with Variable Frequency Inclined SV Waves

Keeping incidence angle constant, at $\theta = 60^\circ$, dynamic responses of an SMR under different frequencies of SV wave ($f = 1Hz, 2.5Hz, 5Hz$ and $10Hz$) is investigated next. Figures 504.107 and 504.108, show displacement and acceleration responses at point A of SMR model.

It is noted that, again, free field response at the location of point A is also shown for comparison purposes. Significantly variation in displacement and acceleration responses are produced by incident SV wave at different frequencies. The largest horizontal displacement magnitude 0.30m is observed for the case of frequency of $f = 2.5Hz$ while the smallest horizontal magnitude of 0.047m for $f = 1Hz$. The vertical displacement responses varies from 0.02m for $f = 2.5Hz$ to 0.085m for $f = 10Hz$. SSI effects are almost negligible in the case of $f = 1Hz$ due to long horizontal wave length of 1154m. This observation follows similar observation made many years ago by Housner (1957) for large stiff buildings. Both horizontal and vertical displacements of SMR overlap with corresponding free field response for $f = 1Hz$. Along with the increase of incident frequency, SSI effects become more significant, especially for the vertical components of displacement and acceleration. In the cases of $f = 2.5Hz$ and $f = 5Hz$, horizontal response of SMR is still very close to its free field counterpart, for both displacements and accelerations, however the reduction of vertical response of SMR becomes more significant for frequency of $f = 5Hz$, For relatively high frequency of $f = 10Hz$, both horizontal and vertical response of SMR are significantly different from free field modeling in both displacements and accelerations.

The spatial variation of displacements at the surface of free field model and at the same location within SMR model, along the horizontal line through SMR (i.e. $x \in [-75\text{m}, 75\text{m}], y = 0m, z = 0\text{m}$), at $t = 3.5$s are shown in Figure 504.109. It is noted that SMR structure occupies space for $x \in [-15m, 15m]$, where flat trace of displacements within a stiff structure is observed. The base slab averaging is observed for higher frequency, shorter wave length cases of $f = 5$Hz and $f = 10$Hz, while it is almost negligible for incident waves at frequencies of $f = 1$Hz or $f = 2.5$Hz due to the wavelength being longer that object size for those low frequencies.

Similar spatial variation of displacement along the transverse axis (i.e. $x = 0m, y \in [-75m, 75m], z = 0$m) is shown in Figure 504.110. Since the incident SV wave propagates within the XZ plane, uniform distribution of both horizontal and vertical free field response along the transverse axis (Y axis) is expected and presented in Figure 504.110. However, the existence of SMR alters the original uniform distribution, and a wave field in this, out plane of polarization direction. Significant wave field disturbance effects can be observed within the structure part ($y \in [-15m, 15m]$) in the cases of medium ($f = 5Hz$) to high frequency ($f = 10Hz$). In other words, 3C dynamic response of soils surrounding the structure has been induced from 2C excitation by an SV wave due to SSI and transverse wave field disturbance effects.

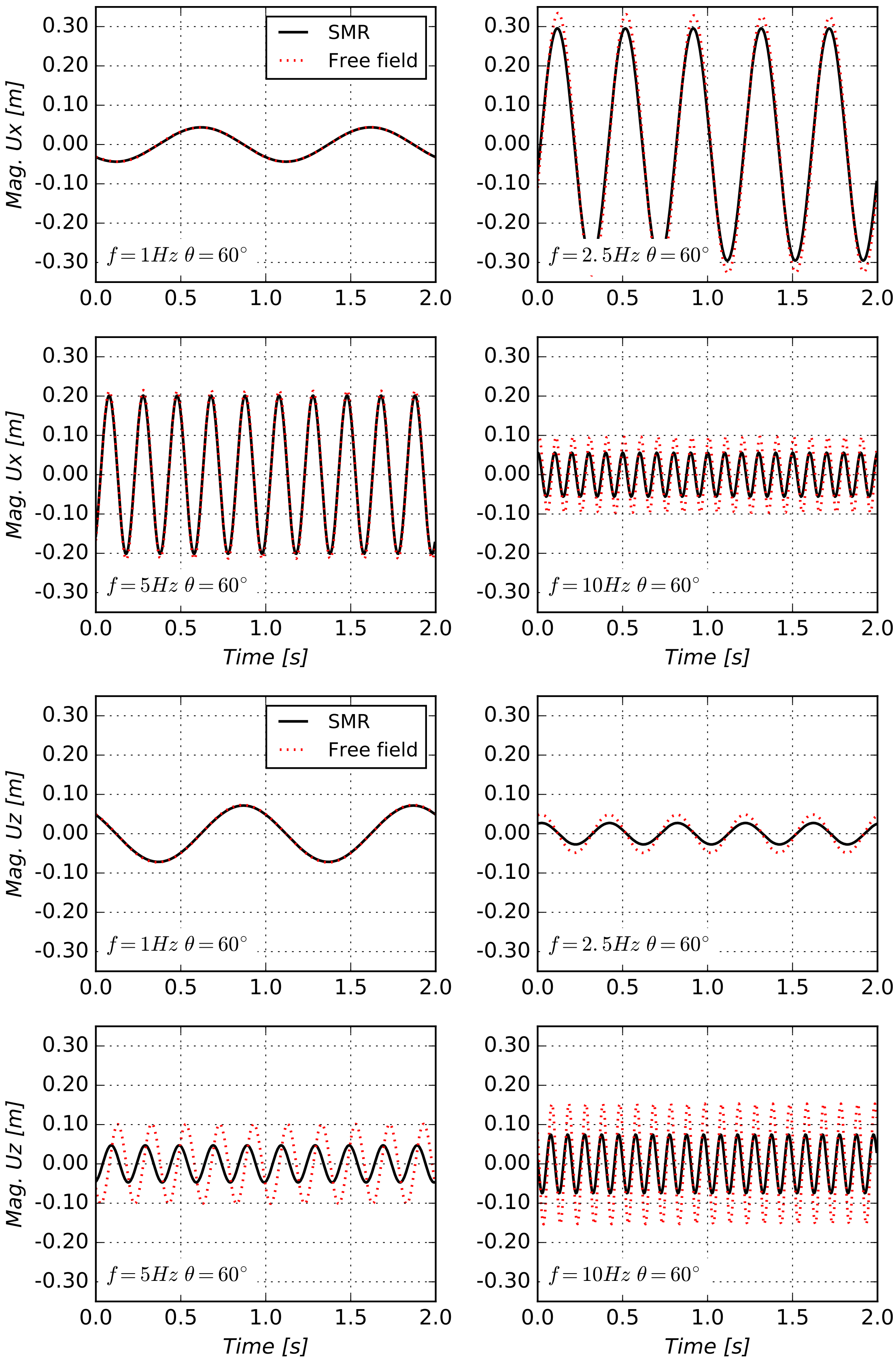


Figure 504.107: Displacement response of point A for scenarios with different frequencies of incident SV wave: (a) Horizontal displacement (b) Vertical displacement.

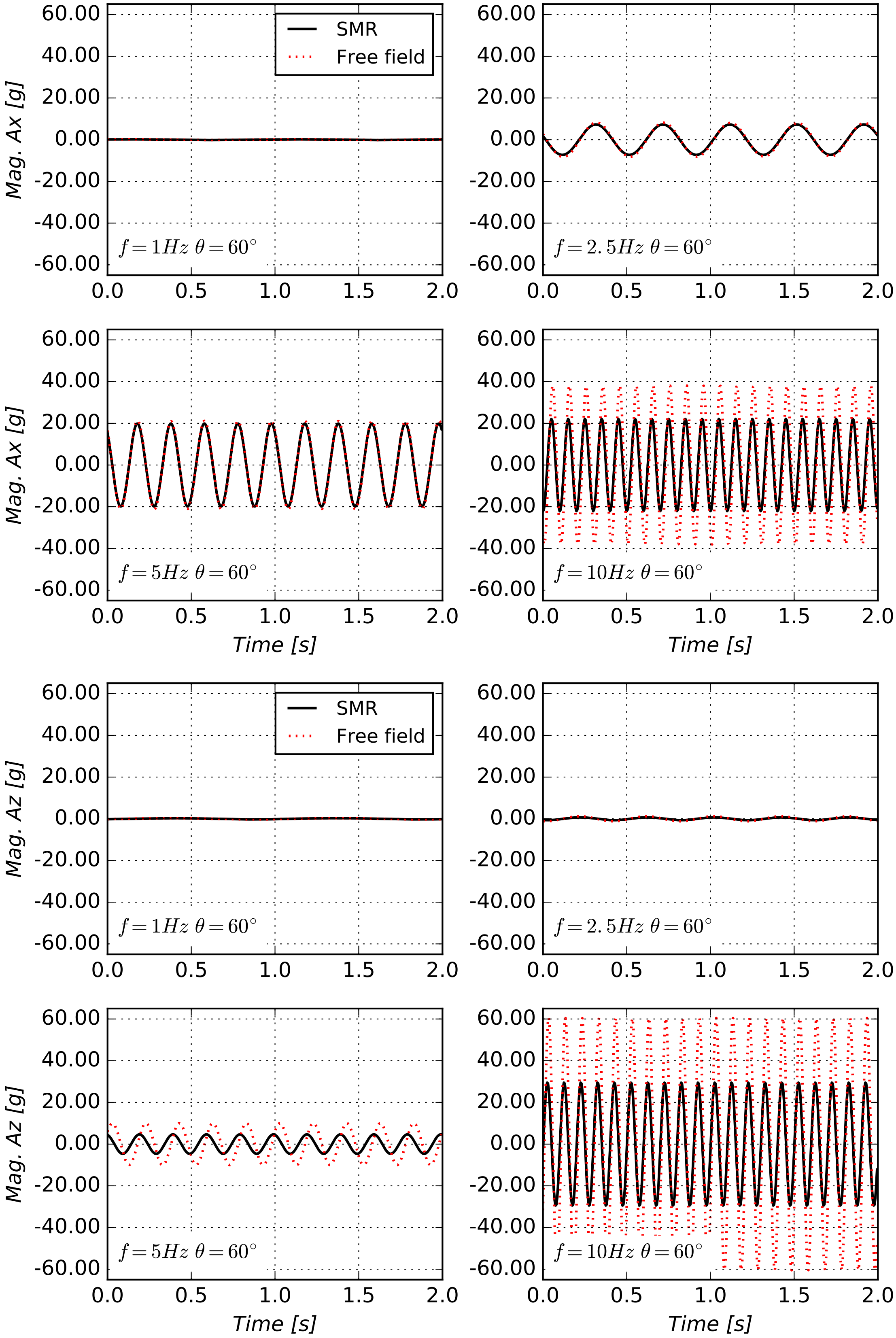


Figure 504.108: Acceleration response of point A for scenarios with different frequencies of incident SV wave: (a) Horizontal acceleration (b) Vertical acceleration.

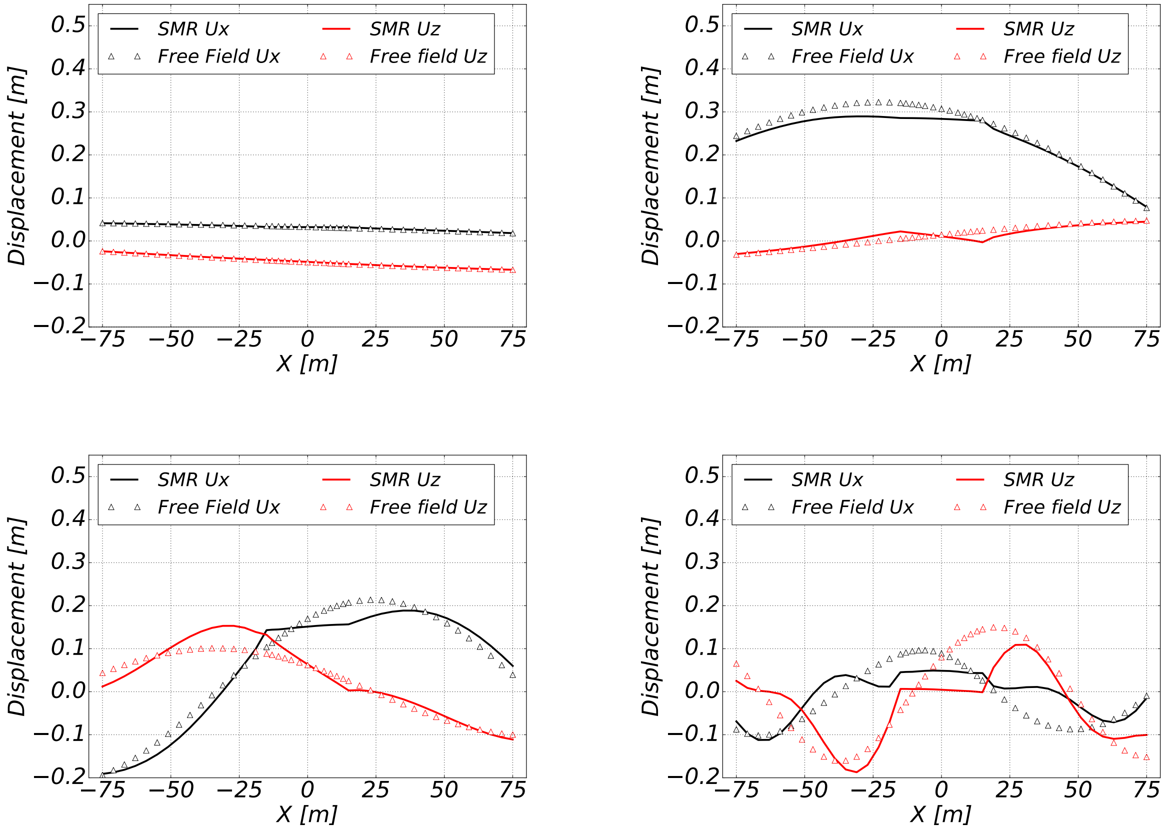


Figure 504.109: Spatial variation of displacement along the horizontal axis at $t = 3.5s$ for a wave at $\theta = 60^o$ different incident wave frequencies (a) f= $1Hz$ (b) f= $2.5Hz$ (c) f= $5Hz$ (d) f= $10Hz$.

Another important observation from Fig. 504.110(d) is that, although the reduction of displacement amplitude is observed within the structure, in locations where $y \in [-15m, 15m]$, near field motions close to the structure can be amplified, for example, motion within region $y \in \pm[25m, 50m]$ in this case. This implies that there are potentially significant structure-soil-structure dynamic effects for closely spaced structures.

The deformed shapes of SMR for four frequency scenarios at $t = 0.3s$ with different frequencies are shown in Fig. 504.111. The aforementioned wave field disturbance effects are clearly visible for the low wave length, high frequency case of $f = 10Hz$. The existence of local structure has significantly altered the near field seismic wave due to strong SSI effect, since wave lengths are shorter than the dominant dimension of the structure.

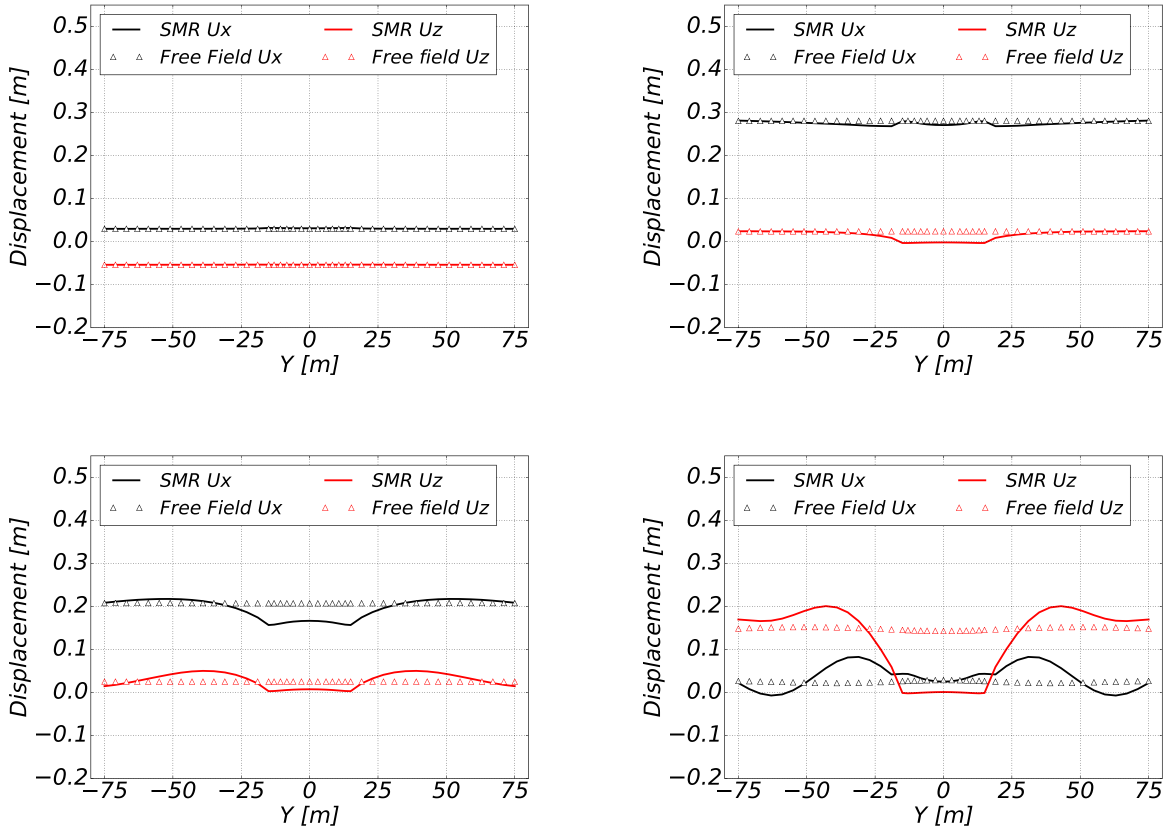


Figure 504.110: Spatial variation of displacement along the transverse axis at $t = 3.5s$ for for a wave at $\theta = 60^o$ at different incident wave frequency (a) f= $1Hz$ (b) f= $2.5Hz$ (c) f= $5Hz$ (d) f= $10Hz$.

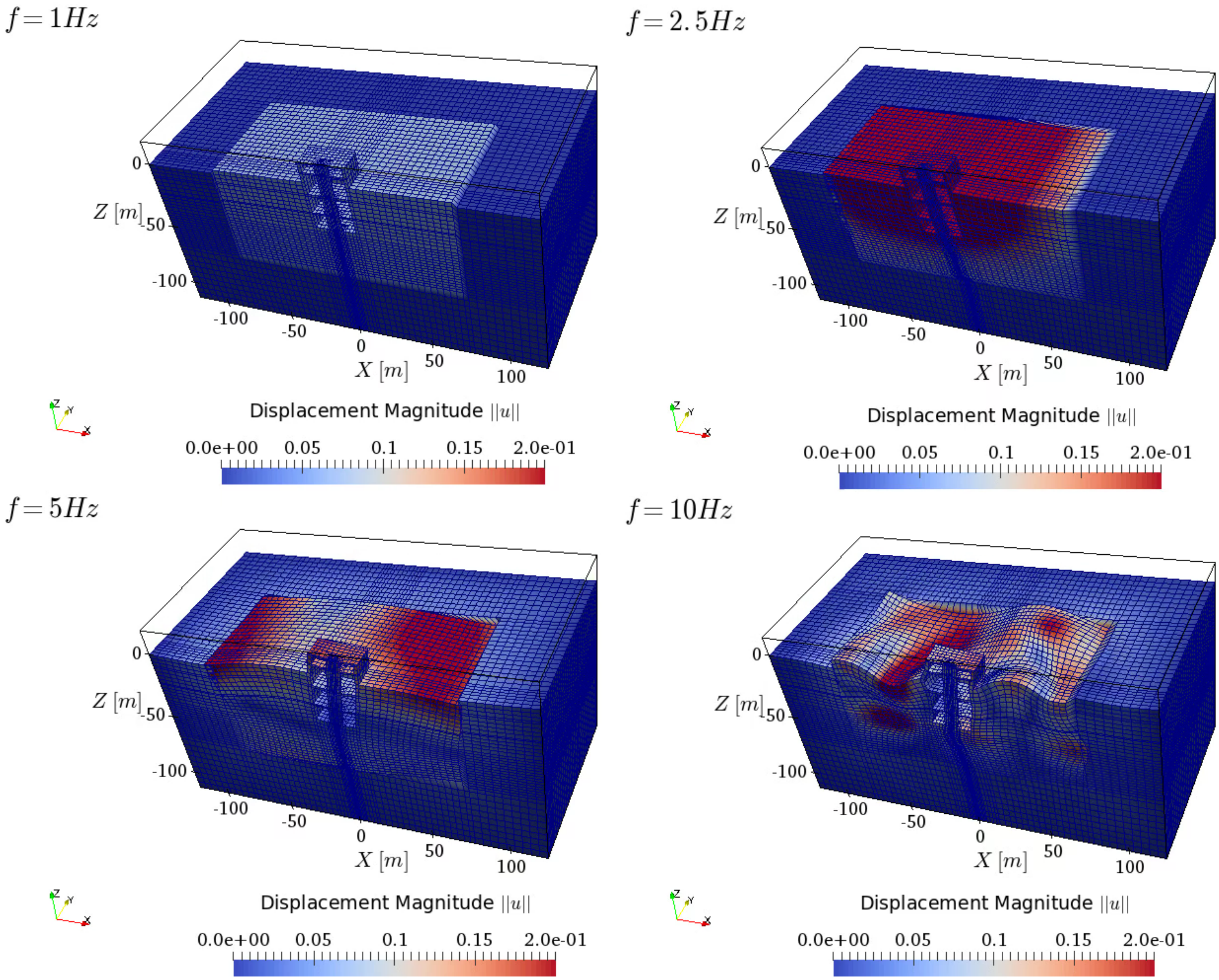


Figure 504.111: The deformed shapes of SMR at $t = 0.3s$ for four scenarios.

504.9 Three Dimensional (3D) Inelastic Modeling for Structure Soil Structure Interaction

504.10 Case Study of Cruas Nuclear Power Plant under Seismic Load from Le Teil Earthquake

504.10.1 Introduction

Seismic hazard in different regions of Central Europe has been recently revised, see, e.g., (Renault et al., 2013). The studies have been based on the historical evidence, like the earthquake in Basel, Switzerland, in 1356 (Mayer-Rosa and Cadiot, 1979). It turns out that quite a few nuclear facilities are located in areas where a weak to moderate seismic activity has been recorded.

Much effort has lately been put into the assessment of existing nuclear stations and into the design of new ones in these earthquake-prone areas. Hence the knowledge on the mechanical behavior of the so-called *earthquake-soil-structure interaction* (ESSI) systems is currently of significant importance in structural and geotechnical engineering.

The nuclear power plant (NPP) in Cruas is located in the south of France, on the right bank of the Rhône River. This NPP was built in the 1970s and it is seismically isolated at the base. On November 11, 2019, the NPP was excited by an earthquake near Le Teil (Viallet et al., 2022). The resulting ground motions were recorded at different locations, both, near Le Teil and at Cruas NPP. The structural vibrations of the NPP were also measured.

Here, simulation of the ground motions and of the mechanical behavior of the NPP under the earthquake load is conducted. For this purpose, the Real-ESSI Simulator (Jeremić et al., 1988-2025) is used. It is a finite element (FE) system developed to reproduce the behavior of soils and structures, and their interaction, under static and dynamic loads. The results obtained in this research are evaluated by comparison with the available field data. In this way, the current design and assessment of ESSI systems can be validated and possibly improved.

This study is conducted within the *SMATCH benchmark*[3] organized by the Institut de Radioprotection et de Sûreté Nucléaire (IRSN), Électricité de France (EDF) and Organisation for Economic Co-operation and Development-Nuclear Energy Agency (OECD-NEA).

---

[3] `https://www.smatch-benchmark.org/`

504.10.2 Notation

Bold-face letters, like $\boldsymbol{\sigma}$, are vectors or second rank tensors. Sans-serif letters, like E, are fourth rank tensors. Gibbs notation, like $\dot{\boldsymbol{\sigma}} = \mathsf{E} : \dot{\boldsymbol{\varepsilon}}$, or index notation, $\dot{\sigma}_{ij} = E_{ijkl}\dot{\varepsilon}_{kl}$, in the Cartesian coordinate system with usual summation over repeated (dummy) indices is used. The mechanical sign convention is applied to stress, $\boldsymbol{\sigma}$, and strain, $\boldsymbol{\varepsilon}$, with compression negative.

The basic Latin variables are explained below.

E	Young modulus
f	frequency
$\mathbf{u}$	displacement
$\mathbf{x}$	location
V_p	primary, compressional, wave velocity
V_s	secondary, shear, wave velocity

The basic Greek variables are given below.

ε	strain
λ, μ	Lamé constants
ν	Poisson ratio
ρ	mass density
$\boldsymbol{\sigma}$	stress, tension positive

The following notation is used.

$\dot{\sqcup}$	material rate of $\sqcup$, $\dot{\sqcup} = \partial \sqcup / \partial t = \sqcup_{,t}$
$\ddot{\sqcup}$	material acceleration of $\sqcup$, $\partial^2 \sqcup / \partial t^2 = \sqcup_{,tt}$

The essential abbreviations are listed below.

DRM	Domain Reduction Method
ESSI	Earthquake-Soil-Structure Interaction
FE	Finite Element
NPP	Nuclear Power Plant
THMM	Thomson-Haskell Matrix Method

504.10.3 Le Teil Earthquake

Some basic knowledge about Le Teil earthquake is indispensable to determine the seismic load in the simulation of the mechanical behavior of Cruas NPP. After a brief introduction of the necessary seismological definitions, a short description of the earthquake in Le Teil is given. It will allow to establish the *seismic input* in Section 504.10.7.

504.10.3.1 A Short Résumé on the Seismological Description of Earthquakes

Origins of Earthquakes Tectonic plates, pieces of the Earth's litosphere, move, due to, i.a., the convection in the Earth's mantel (Andel, 2008; Bragg, 2022), and interact with each other at *interfaces*. An interface, also known as a *fault*, between two tectonic plates is rough, jigsaw-like, and the movement of the plates is hindered. This is similar to shearing along a fault in a rock (Wittke, 2014).

During shearing along an interface, the jigsaw-like boundaries of the plates undergo the plastic deformation. Stress at the interface is being increased, which is known as *hardening*, and the *plastic work* is being done at the interface. A part of the plastic work, the so-called *frozen elastic energy*, also known as the *plastic free Helmholtz energy*, is being accumulated at the fault.

When the stress limit is reached, the peak friction is mobilized, at the fault, a sudden, *dynamic*, brittle *slip* occurs and causes an *earthquake*. A slip is a movement of the plates along the interface. A ductile slip is also possible but it is *aseismic* (Kramer, 1996a) and hence it is not considered here. A brittle slip is also called a *rupture*.
The strain is localized along the fault and the stress decreases, which is known as *softening*. The frozen elastic energy accumulated at the interface is transformed into other forms of energy, like *acoustic energy* or heat. In seismology, the origins of earthquakes are explained with the so-called *elastic-rebound theory* (Kramer, 1996a; Bragg, 2022). It is based on the assumption of repeatable accumulation and release of the frozen elastic energy at a fault.

However, an earthquake may also result from various, natural or anthropogenic, events within a single tectonic plate (Kramer, 1996a; Semblat and Pecker, 2009), for example, from a large underground explosion, a volcanic eruption or a slip triggered by *fracking*. A slip may generally occur along, e.g., an interface between two rock masses and not necessarily along an interface between two tectonic plates.

Here, a shallow earthquake, triggered at a depth of less than 5 km, is examined. It was caused by a slip along a fault in a rock.

Types of Faults A slip is described by jump in displacement $[\![\mathbf{u}]\!]$ between two rock masses, or two tectonic plates. This $[\![\mathbf{u}]\!]$ is measured along the slip plane, π. At a fault, discontinuity, in a rock, $\|[\![\mathbf{u}]\!]\|$ can achieve several kilometers (Wittke, 2014). Two rock masses, or two tectonic plates, can (Wittke, 2014; Bragg, 2022; Andel, 2008):

- slide past each other at a *transform fault*, particularly:
    - in horizontal direction at a *strike-slip fault*, inclined or vertical, Figure 504.112(a)
    - in vertical direction at a *dip-slip fault*, vertical, Figure 504.112(b)
    - in both, horizontal and vertical, directions at an *oblique-slip fault*, vertical, Figure 504.112(c)
- move towards each other, collide, at a *reverse fault*, also called *thrust fault*, inclined or horizontal, Figure 504.112(d)
- move apart from each other at a *normal fault*, inclined or horizontal, Figure 504.112(e).

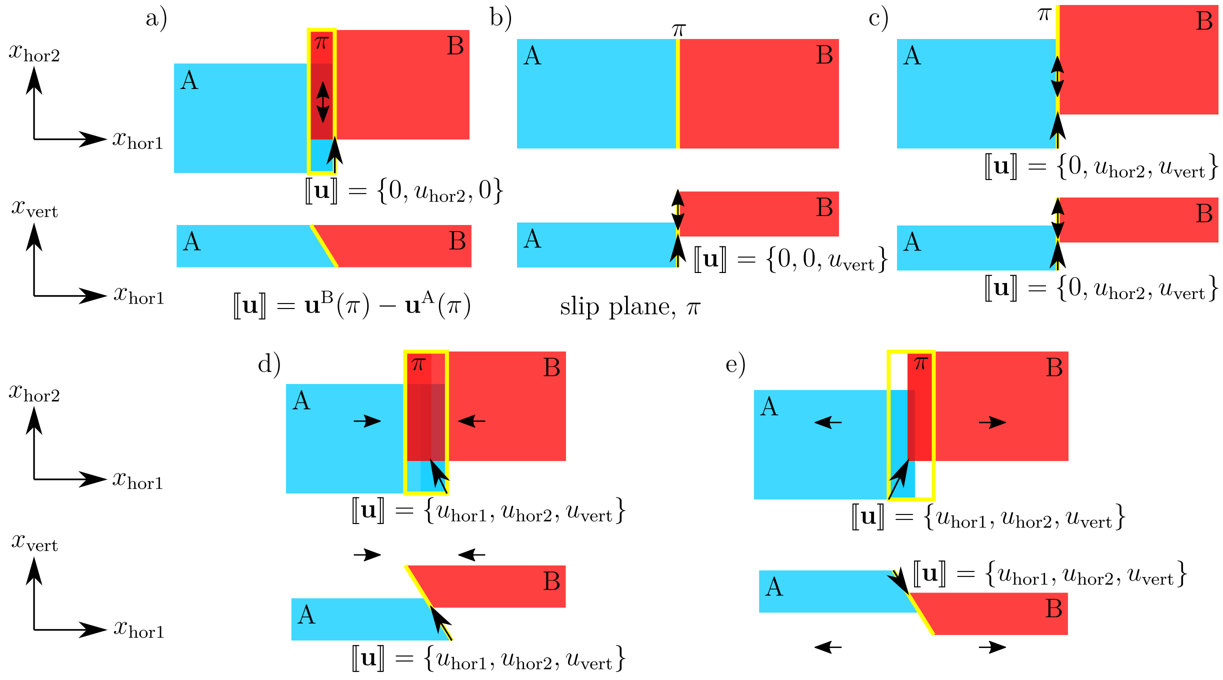


Figure 504.112: Types of faults: a) strike-slip fault, here, inclined; b) dip-slip fault; c) oblique-slip fault; d) reverse fault, here, inclined; e) normal fault, here, inclined.

Mechanism and Orientation of a Slip in a Cartesian Coordinate System Mechanism and orientation of a slip are commonly described in a Cartesian coordinate system using the so-called *focal mechanism*. It is a graphic representation of the slip and it is known in the seismological community as the “beach ball”[4]. A focal mechanism is defined by *strike angle*, *dip angle* and *rake angle* (Aki and Richards, 2002; Wittke, 2014), see Figure 504.113.

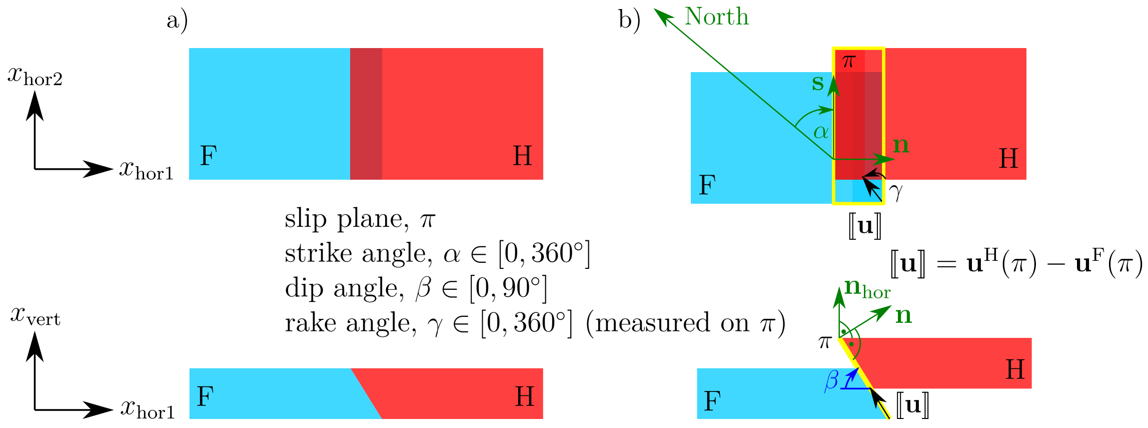


Figure 504.113: a) Inclined fault before a slip; b) Inclined fault after a reverse slip: the slip plane, π, is defined by the unit normal vector, $\mathbf{n}$, that points upwards; $\mathbf{n}_{\mathrm{hor}}$ is a unit vector that is normal to the horizontal plane and points upwards; $\mathbf{s} = (\mathbf{n}_{\mathrm{hor}} \times \mathbf{n})^{\rightarrow}$; strike angle, $\alpha \in \left[0, 360°\right]$, is the azimuth of $\mathbf{s}$; dip angle, $\beta \in \left[0, 90°\right]$, describes the inclination of π; rake angle, $\gamma \in \left[0, 360°\right]$, is measured anticlock-wise from $\mathbf{s}$ to the jump in displacement, $[\![\mathbf{u}]\!]$, on π. F is the so-called *foot wall* and H is the *hanging wall*.

Strike angle, dip angle and rake angle can be found in MATHEMATICA for a slip plane, π, defined by the unit normal vector, $\mathbf{n}$, with given displacement of the foot wall, $\mathbf{u}^{\mathrm{F}}(\pi)$, and of the hanging wall, $\mathbf{u}^{\mathrm{H}}(\pi)$, along π.

```
n = Normalize[{1, 1, 1}]; nhor = {0, 0, 1}; s = Normalize[Cross[nhor, n]]; north = {0, 1, 0};
rotclockwise = {{Cos[a], Sin[a], 0}, {-Sin[a], Cos[a], 0}, {0, 0, 1}};
solua = Solve[rotclockwise.north == s, a][[1]] /. C[1] -> 0; (* solve for strike angle *)
rotanticlockwise = {{Cos[a], -Sin[a], 0}, {Cos[a], Sin[a], 0}, {0, 0, 1}}; nprime = (rotanticlockwise.n) /. solua;
solub = Solve[Tan[b] == nprime[[3]]/nprime[[1]], b][[1]] /. C[1] -> 0; (* solve for dip angle *)
uF = {0, 0, 0}; uH = {1, 1, -2}; jump = uH - uF;
solug = Solve[((rotclockwise /. a -> g).jump/Norm[jump])[[1]] == 1, g][[1]] /. C[1] -> 0; (* solve for rake angle *)
```

---

[4] Focal mechanism is briefly explained under `https://www.iris.edu/hq/inclass/animation/focal_mechanisms_explained`. Knowing the strike, dip and slip angle, one can easily plot the focal mechanism in MATHEMATICA using the code by Prof. Scherbaum from the University of Potsdam, available on the Internet, `https://demonstrations.wolfram.com/EarthquakeFocalMechanism/`.

```
{a //. solua, b /. solub, Re[g /. solug]} // N (* strike angle, dip angle, rake angle *)
```

504.10.3.2 Slip along Rouvière Fault near Le Teil

Le Teil earthquake was recorded on November 11, 2019 (Marconato et al., 2022; Ritz et al., 2020; Viallet et al., 2022). This earthquake resulted from a slip along Rouvière fault within the existing les Cévennes faults system in Southern France, near Le Teil, Figure 504.114. The slip might have been caused by a reduction of the effective stress at Rouvière fault either due to the increase in the ground water level, caused by a heavy rain, (Burnol et al., 2023) or due to the progressive unloading of the ground in the nearby quarry (Novellis et al., 2020).

Basic characteristics of the slip that caused shallow Le Teil earthquake are given in Table 504.7 and explained in Figure 504.113. The location of the source of the earthquake is shown in Figure 504.114.

Table 504.7: Basic characteristics of the slip along Rouvière fault

Latitude	Longitude	Depth	Mw	Dip angle	Strike angle	Rake angle	Velocity
°	°	km	-	°	°	°	m/s
44.5188 ± 0.01	4.6694 ± 0.01	1 ± 0.5	4.9 ± 0.1	55 ± 5	45 ± 5	90 ± 10	1800

504.10.3.3 Site Description

A number of free-field and in-structure instruments were used to record the seismic motions at Cruas NPP, the local site, and near Le Teil. The data are accessed via the SMATCH project. Unfortunately, the elastic soil and rock parameters are available at the epicenter of the earthquake and at two locations at the local site only.

Locations of the Free-field Recordings Locations of the free-field instruments are given in Table 504.8 and shown in Figure 504.114. In-structure recordings are discussed in Section 504.10.6.

Elastic Soil and Rock Parameters The elastic soil and rock parameters at station 1 and station 2, at Cruas NPP, and at the epicenter of the earthquake are given in Table 504.9. The elevations at station 1 and 2 are given in Table 504.8. The elevation of the epicenter, ≈ 260 m, can be estimated using, e.g., an online map with elevation contours.

Table 504.8: Locations of the free-field instruments at Cruas NPP and near Le Teil

Station	Latitude °	Longitude °	Elevation m
1	44.636253	4.758796	77
2	44.630001	4.753816	80.5
3	44.37408	4.76974	90
4	44.307	4.689	46
5	44.3561	4.8572	141.2
6	44.324346	4.73236	68

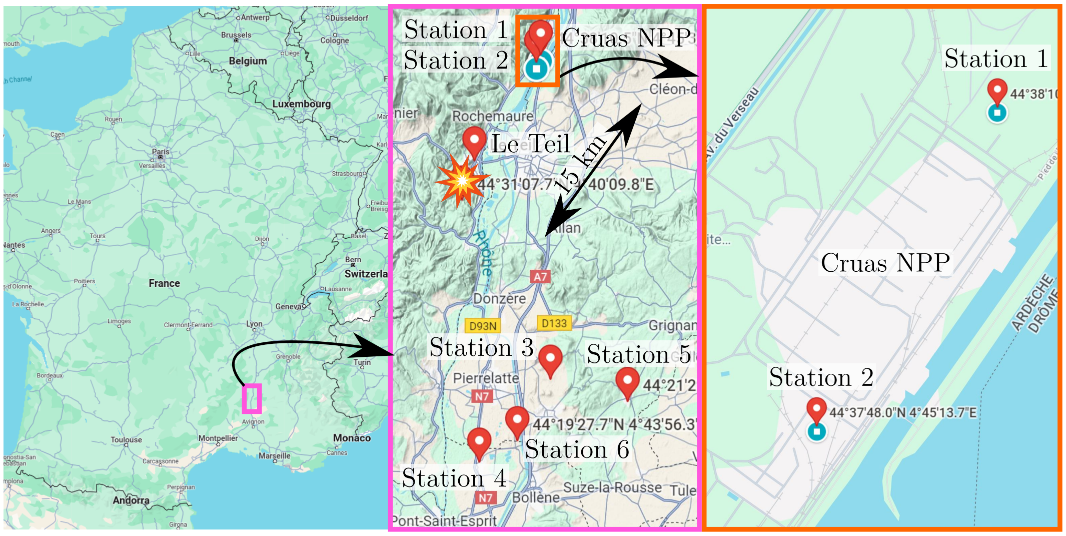


Figure 504.114: Locations of the free-field instruments near Le Teil and at Cruas NPP and location of the source of the earthquake

Table 504.9: Elastic soil and rock parameters at station 1 and station 2, at Cruas NPP, and at the epicenter of the earthquake

Station 1				Station 2				Epicenter of the earthquake			
z	ρ	V_s	ν	z	ρ	V_s	ν	z	ρ	V_s	ν
m	kg/m^3	m/s	-	m	kg/m^3	m/s	-	m	kg/m^3	m/s	-
0	1988	300	0.48	0	2039	300	0.48	0	2500	2047	0.21
1	1978	160	0.45								
3	2141	450	0.45								
				6.5	1937	180	0.45				
7	2350	800	0.45								
				7.5	2141	450	0.45				
				12.5	2500	1600	0.4				
17	2500	1600	0.4								
62	2500	$60 + 23.7z$	0.4								
				65.5	2500	$60 + 23.7z$	0.4				
100	2500	2500	0.4								
				103.5	2500	2500	0.4				
								628	2600	3645	0.21
								1197	2300	1200	0.20
								1416	2500	2291	0.22
								2026	2500	2314	0.23
								2194	2600	3457	0.23
								5956	2600	3616	0.21

504.10.4 FE Simulation of Ground Motions from Le Teil Earthquake

Ground motions from Le Teil earthquake had been intensively studied within the SIGMA2 project[5] (Viallet et al., 2022). Two kinds of methods for the simulation of seismic motions had been used, that is, the empirical ones and the physically based ones. The latter include both 1D (Fasan, 2016; Fasan et al., 2016; Hassan et al., 2020; Magrin et al., 2016; Magrin, 2012; Panza et al., 2012) and 3D (Mazzieri et al., 2013; Paolucci et al., 2020, 2021) simulations. Some recent 3D numerical calculations of seismic motions from Le Teil earthquake are reported in (Lehmann et al., 2023; Smerzini et al., 2023).

Elastic soil and rock parameters are available at three locations only, see Table 504.9. This scarcity of data alone precludes a realistic FE reproduction of Le Teil earthquake. Different assumptions about the regional geology could be made, of course. Then, a parametric back-analysis of the earthquake could be conducted to match the measured ground motions. However, such analysis would require a very large FE domain to avoid the corruption of the obtained results by an unreal behavior at the model boundaries, which generally cannot be avoided, see, e.g., (Baffet et al., 2012). It is well-known that in dynamics a parametric FE calculation with a large spatial domain involves a considerable computation time.

Here, we aim to save the computation time. For this purpose, we make a few physically justified simplifying assumptions about the ground motions from Le Teil earthquake. Based on these assumptions, the seismic input will be determined and used in a computationally efficient simulation of the mechanical behavior of Cruas NPP, Section 504.10.7. To establish the necessary assumptions, we first conduct a preliminary FE calculation of the seismic wave field from Le Teil earthquake using the Real-ESSI Simulator (Jeremić et al., 1988-2025).

504.10.4.1 Preliminary 2D Simulation of Le Teil Earthquake

In the case of a shallow earthquake, the incident angle larger than the critical angle, $\alpha_i > \alpha_{ic}$, see (504.13) and (504.14), should be expected at the ground surface. Such $\alpha_i > \alpha_{ic}$ results in a *Rayleigh wave*, see, e.g., (Nowacki, 1974).

A preliminary simulation of the earthquake in Le Teil is performed and used to examine the possible presence of a Rayleigh wave. The calculated ground motions will be validated by a qualitative comparison with the field measurements.

---

[5] `https://www.sigma-2.net/`

The geometry, geology, material description and mechanism of the slip are simplified. The FE model represents $2000 \times 1 \times 300$ m^3 ground with a 280 m thick rock that is overlayed by a 20 m thick soil, see Figure 504.115, cf. Table 504.9.

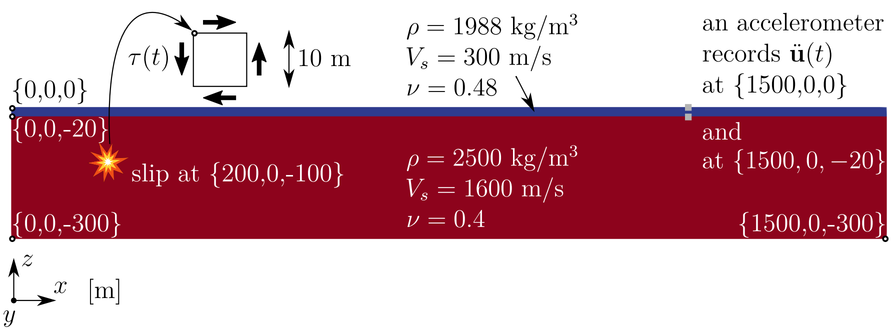


Figure 504.115: FE model used in the preliminary 2D simulation of Le Teil earthquake

Both materials are isotropic linear elastic with parameters given in Table 504.10 (with Young modulus $E = 2\rho V_s^2(1+\nu)$).

Table 504.10: Material properties in preliminary 2D simulation of Le Teil earthquake

Material	z m	ρ kg/m^3	V_s m/s	ν
Soil	$[0, -20]$	1988	300	0.48
Rock	$[-20, -300]$	2500	1600	0.4

The ground is discretized into a single layer (in y direction, Figure 504.115) of 6000 `27NodeBrick` elements. The size of the element is 10 m.

The slip is simulated as the *pure shear* of an element at $\boldsymbol{x} = \{200, 0, -100\}$. The dynamic disturbance is introduced onto faces of the element in the form of a shear load, $\tau(t) = \tau^{\text{ampl}} \sin(2\pi f t)$ with $\tau^{\text{ampl}} = -60$ GPa and the frequency $f = 1$ Hz.

`Newmark` integration procedure is used in search of the dynamic equilibrium. Numerical damping is introduced using parameters $\gamma = 0.6$ and $\beta = 0.3025$ in the `Newmark` algorithm.

Time functions of the horizontal displacement, $u_x(t)$, and the vertical displacement, $u_z(t)$, at the ground surface, $\boldsymbol{x} = \{1500, 0, 0\}$ m, and at the top of the rock layer, $\boldsymbol{x} = \{1500, 0, -20\}$ m, are plotted in Figure 504.116(a) and in Figure 504.117(a), respectively. Additionally, the corresponding hodographs, $u_z(u_x)$, are shown in Figure 504.116(b) and in Figure 504.117(b). The horizontal ground acceleration, $\ddot{u}_x(t)$, and the vertical ground acceleration, $\ddot{u}_z(t)$, at $\boldsymbol{x} = \{1500, 0, 0\}$ m and at $\boldsymbol{x} = \{1500, 0, -20\}$ m are plotted in Figure 504.118(a) and in Figure 504.119(a). The corresponding diagrams $\ddot{u}_z(\ddot{u}_x)$ are shown in Figure 504.118(b) and in Figure 504.119(b).

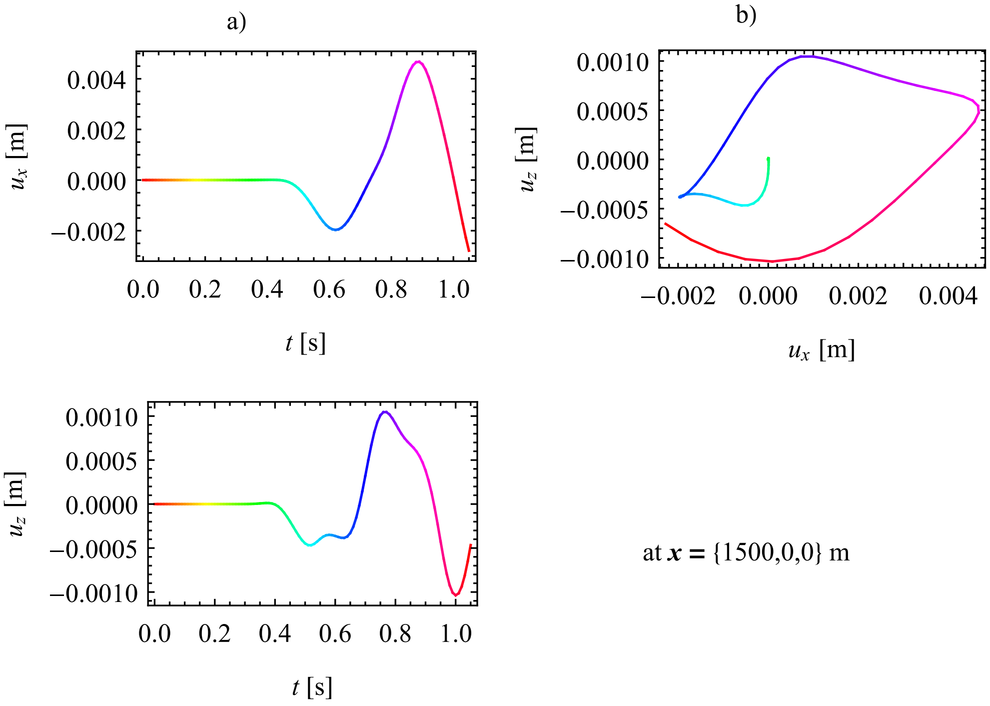


Figure 504.116: Preliminary 2D simulation of Le Teil earthquake: a) $u_x(t)$ and $u_z(t)$, and b) $u_z(u_x)$ at the ground surface, $\boldsymbol{x} = \{1500, 0, 0\}$ m

Judging by the hodograph from Figure 504.116(b), the motions at the ground surface can be interpreted as a Rayleigh wave, a surface wave. It is probably overlapped by the Stoneley wave from Figure 504.117(b) which propagates along the interface between the soil and the rock.

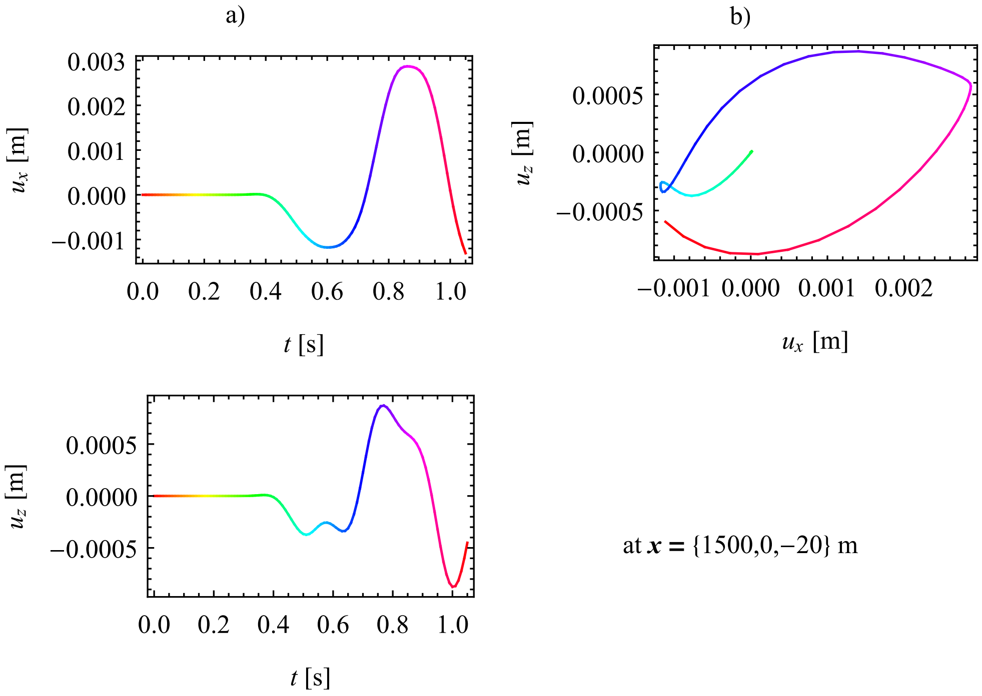


Figure 504.117: Preliminary 2D simulation of Le Teil earthquake: a) $u_x(t)$ and $u_z(t)$, and b) $u_z(u_x)$ at the top of the rock layer, $\boldsymbol{x} = \{1500, 0, -20\}$ m

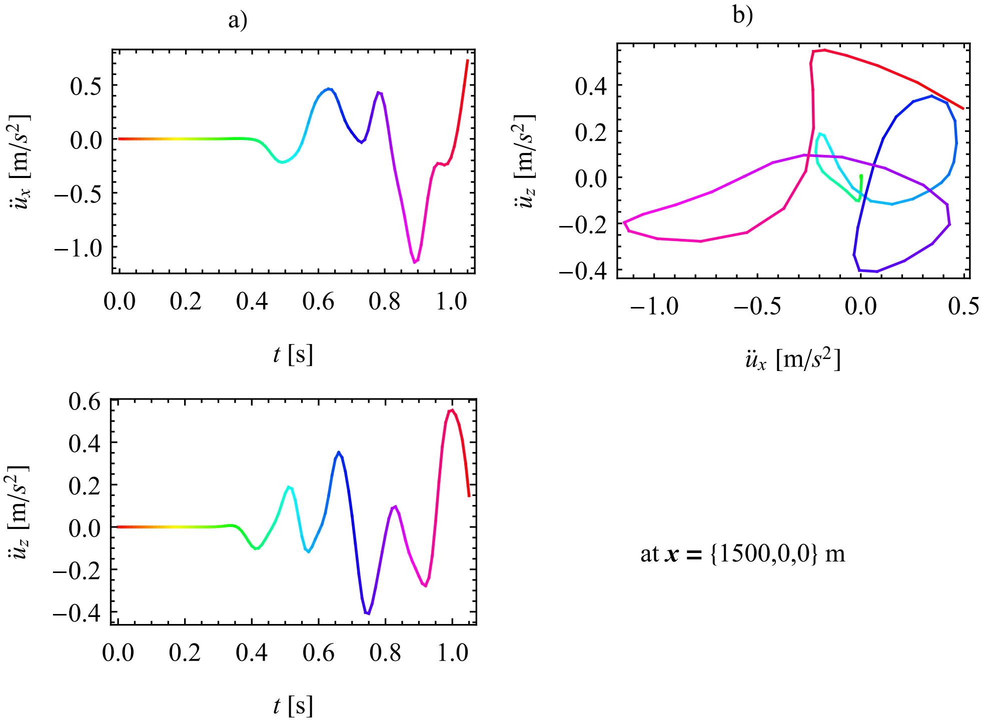


Figure 504.118: Preliminary 2D simulation of Le Teil earthquake: a) $\ddot{u}_x(t)$ and $\ddot{u}_z(t)$, and b) $\ddot{u}_z(\ddot{u}_x)$ at the ground surface, $\boldsymbol{x} = \{1500, 0, 0\}$ m

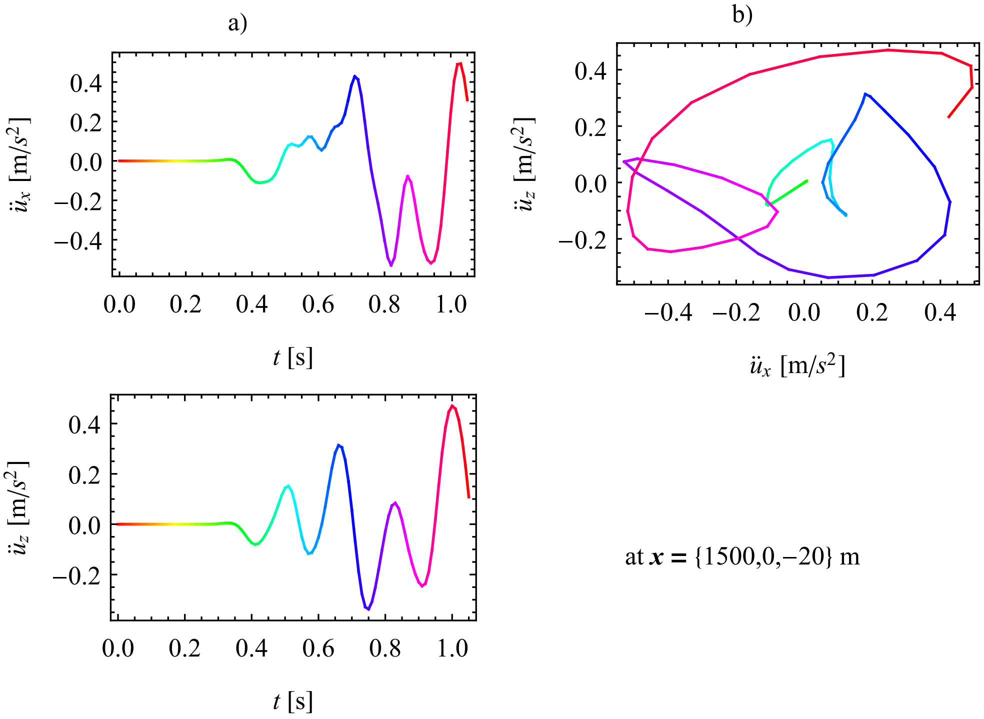
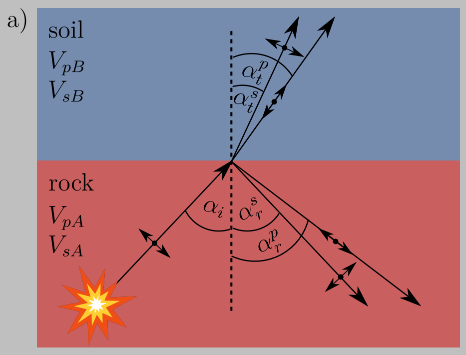
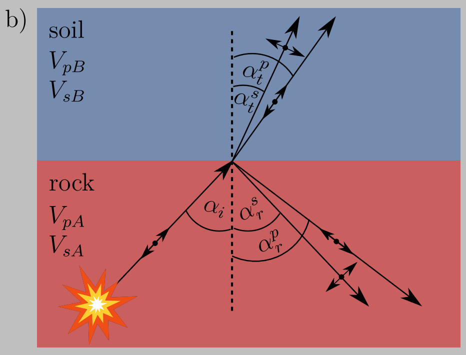


Figure 504.119: Preliminary 2D simulation of Le Teil earthquake: a) $\ddot{u}_x(t)$ and $\ddot{u}_z(t)$, and b) $\ddot{u}_z(\ddot{u}_x)$ at the top of the rock layer, $\boldsymbol{x} = \{1500, 0, -20\}$ m

Some "unreal" *reflections* and *refractions* of the waves, at the bottom and side boundaries of the model, are present in the simulation due to the very small size of the model. Reflection and refraction of a wave is governed by the *Snell law*. It is assumed that the "unreal" waves do not corrupt the preliminary results significantly.

Let us consider a 2D S-wave, see Figure (a) below, or a 2D P-wave, Figure (b) below, that travels from the source, through the rock.





The incident wave arrives at the interface between the rock and the soil at an incident angle, $\alpha_i \in [0, 90^\circ)$, and it is *reflected* and *refracted*. Component of the wave that is normal to the interface is reflected back into the rock at α_r^p and transmitted, refracted, into the soil at α_t^p as a P-wave. Component tangent to the interface is reflected back into the rock at α_r^s and transmitted into the soil at α_t^s as a S-wave.

According to the Snell law, the following holds

$$\begin{aligned}
\frac{\sin\alpha_i}{V_{sA}} = \frac{\sin\alpha_r^p}{V_{pA}} = \frac{\sin\alpha_r^s}{V_{sA}} = \frac{\sin\alpha_t^p}{V_{pB}} = \frac{\sin\alpha_t^s}{V_{sB}} \quad &\text{for the incident wave being a S-wave and} \\
\frac{\sin\alpha_i}{V_{pA}} = \frac{\sin\alpha_r^p}{V_{pA}} = \frac{\sin\alpha_r^s}{V_{sA}} = \frac{\sin\alpha_t^p}{V_{pB}} = \frac{\sin\alpha_t^s}{V_{sB}} \quad &\text{for the incident wave being a P-wave}\,,
\end{aligned} \tag{504.11}$$

wherein

$$V_p = \sqrt{\frac{\lambda}{\rho} + 2V_s^2} \quad \text{with} \quad \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \tag{504.12}$$

and V_{pA}, V_{sA} denote the wave velocities in material from which a wave arrives at the interface and V_{pB}, V_{sB} denote the wave velocities in material into which the wave is transmitted.

Additionally, one defines the so-called *critical angle*, α_{ic}. It is the incident angle for which the incident wave is transmitted or reflected parallel to the interface between two layers, here, rock and soil,

$$\begin{aligned}
\sin\alpha_{ic}^{srp} &= \frac{V_{sA}}{V_{pA}} \quad \text{with} \quad \alpha_r^p = 90^\circ \quad \text{and} \quad V_{sA} < V_{pA} \quad \text{for an incident S-wave reflected as the P-wave} \\
\sin\alpha_{ic}^{stp} &= \frac{V_{sA}}{V_{pB}} \quad \text{with} \quad \alpha_t^p = 90^\circ \quad \text{and} \quad V_{sA} \leq V_{pB} \quad \text{for an incident S-wave transmitted as the P-wave} \\
\sin\alpha_{ic}^{sts} &= \frac{V_{sA}}{V_{sB}} \quad \text{with} \quad \alpha_t^s = 90^\circ \quad \text{and} \quad V_{sA} \leq V_{sB} \quad \text{for an incident S-wave transmitted as the S-wave}
\end{aligned} \tag{504.13}$$

and

$$\begin{aligned}
\sin\alpha_{ic}^{ptp} &= \frac{V_{pA}}{V_{pB}} \quad \text{with} \quad \alpha_t^p = 90^\circ \quad \text{and} \quad V_{pA} \leq V_{pB} \quad \text{for an incident P-wave transmitted as the P-wave} \\
\sin\alpha_{ic}^{pts} &= \frac{V_{pA}}{V_{sB}} \quad \text{with} \quad \alpha_t^s = 90^\circ \quad \text{and} \quad V_{pA} \leq V_{sB} \quad \text{for an incident P-wave transmitted as the S-wave.}
\end{aligned} \tag{504.14}$$

If the incident angle, α_i, exceeds the critical angle, α_{ic}, it follows that

$$\sin\alpha_i > \sin\alpha_{ic} \quad \text{with} \quad V_A \leq V_B \qquad \frac{V_A}{V_B}\sin\alpha_t > \frac{V_A}{V_B} \qquad \sin\alpha_t > 1 \tag{504.15}$$

for transmission of the incident wave from layer A to layer B or, analogously,

$$\sin\alpha_i > \sin\alpha_{ic} \quad \text{with} \quad V_{sA} < V_{pA} \qquad \frac{V_{sA}}{V_{pA}}\sin\alpha_r^p > \frac{V_{sA}}{V_{pA}} \qquad \sin\alpha_r^p > 1 \tag{504.16}$$

for reflection of the incident S-wave as the P-wave in layer A. If (504.15) or (504.16) holds, then an interface wave is present, e.g., the *Rayleigh wave*.

504.10.5 Validation of the Preliminary 2D Simulation of Le Teil Earthquake

Ground accelerations at station 1, at Cruas NPP, see Table 504.8 and Figure 504.114, are plotted as time functions, $\ddot{u}_{\mathrm{NS}}(t), \ddot{u}_{\mathrm{EW}}(t), \ddot{u}_{\mathrm{UD}}(t)$, in Figure 504.120(a). Indices NS, EW and UD denote the North-South, East-West and up-down components of $\ddot{\mathbf{u}}$, respectively. Accelerations $\ddot{u}_{\mathrm{NS}}(t), \ddot{u}_{\mathrm{EW}}(t), \ddot{u}_{\mathrm{UD}}(t)$ were integrated over time t into displacements $u_{\mathrm{NS}}, u_{\mathrm{EW}}, u_{\mathrm{UD}}$. Functions $u_{\mathrm{NS}}(t), u_{\mathrm{EW}}(t), u_{\mathrm{UD}}(t)$ are given in Figure 504.120(b).

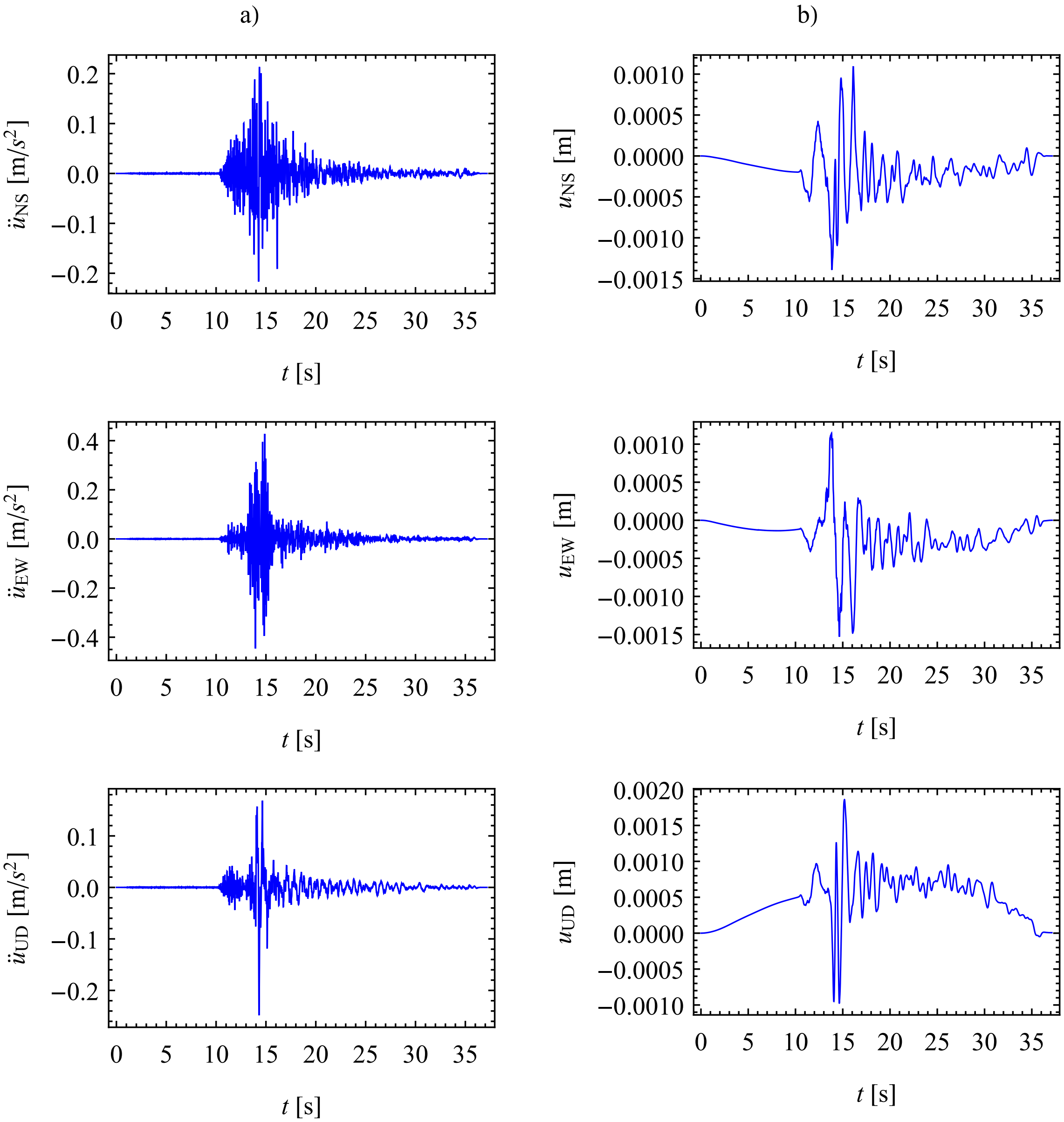


Figure 504.120: Ground motions from Le Teil earthquake recorded at station 1, at Cruas NPP: a) accelerations $\ddot{u}_{\rm NS}(t), \ddot{u}_{\rm EW}(t), \ddot{u}_{\rm UD}(t)$ and b) displacements $u_{\rm NS}(t), u_{\rm EW}(t), u_{\rm UD}(t)$.

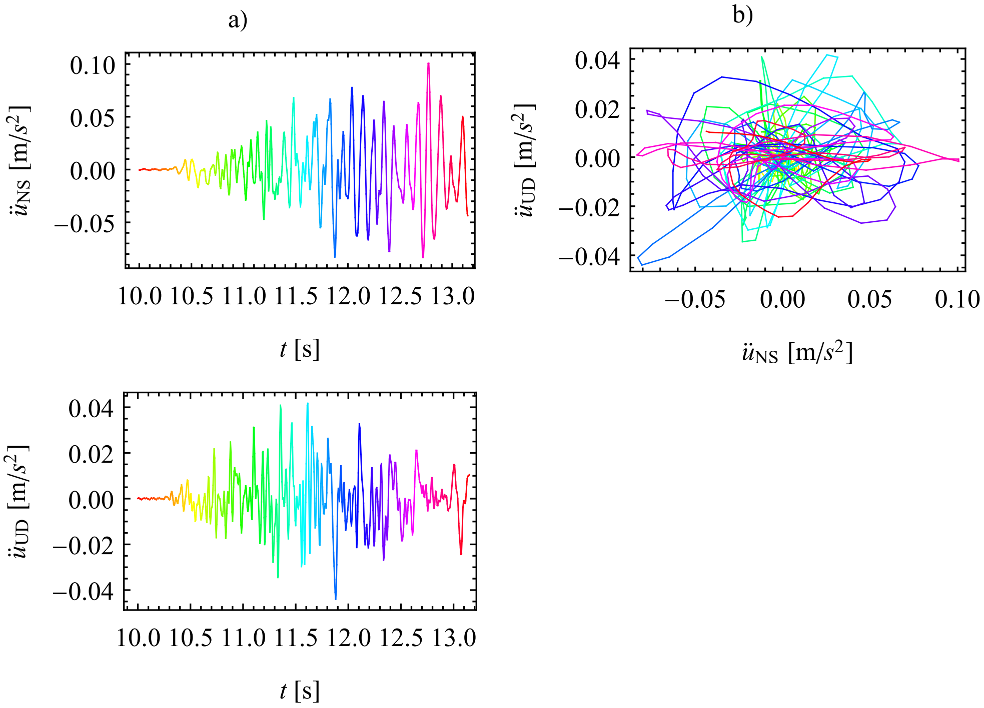


Figure 504.121: Ground accelerations from Le Teil earthquake recorded at station 1, at Cruas NPP, for $t \in [10, 13.15]$ s: a) $\ddot{u}_{NS}(t), \ddot{u}_{UD}(t)$ and b) $\ddot{u}_{UD}(\ddot{u}_{NS})$.

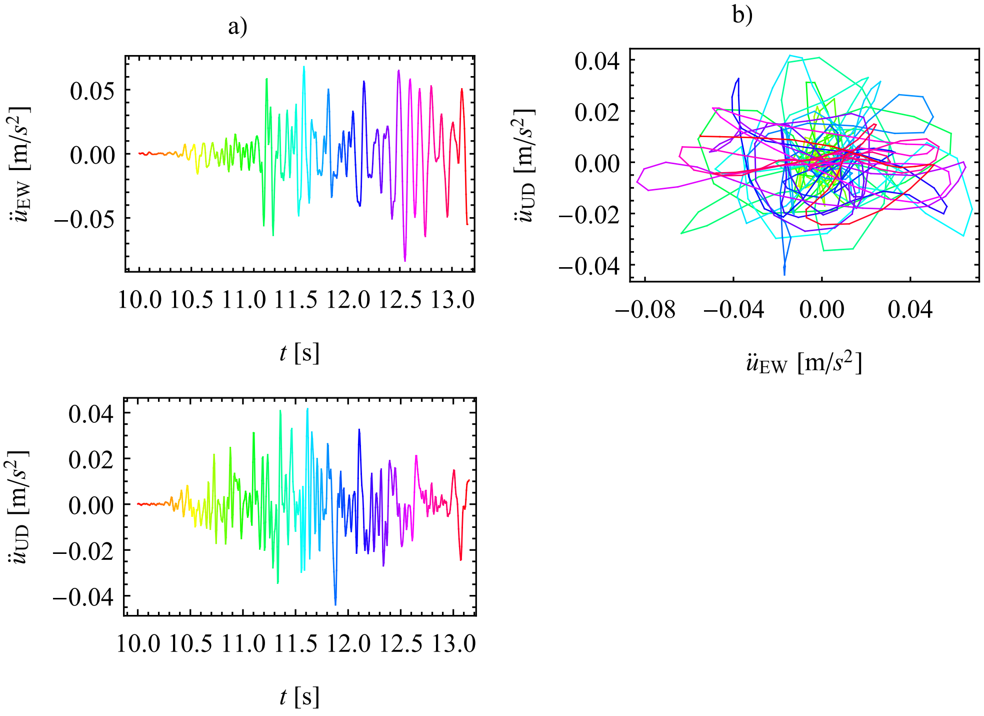


Figure 504.122: Ground accelerations from Le Teil earthquake recorded at station 1, at Cruas NPP, for $t \in [10, 13.15]$ s: a) $\ddot{u}_{\mathrm{EW}}(t), \ddot{u}_{\mathrm{UD}}(t)$ and b) $\ddot{u}_{\mathrm{UD}}(\ddot{u}_{\mathrm{EW}})$.

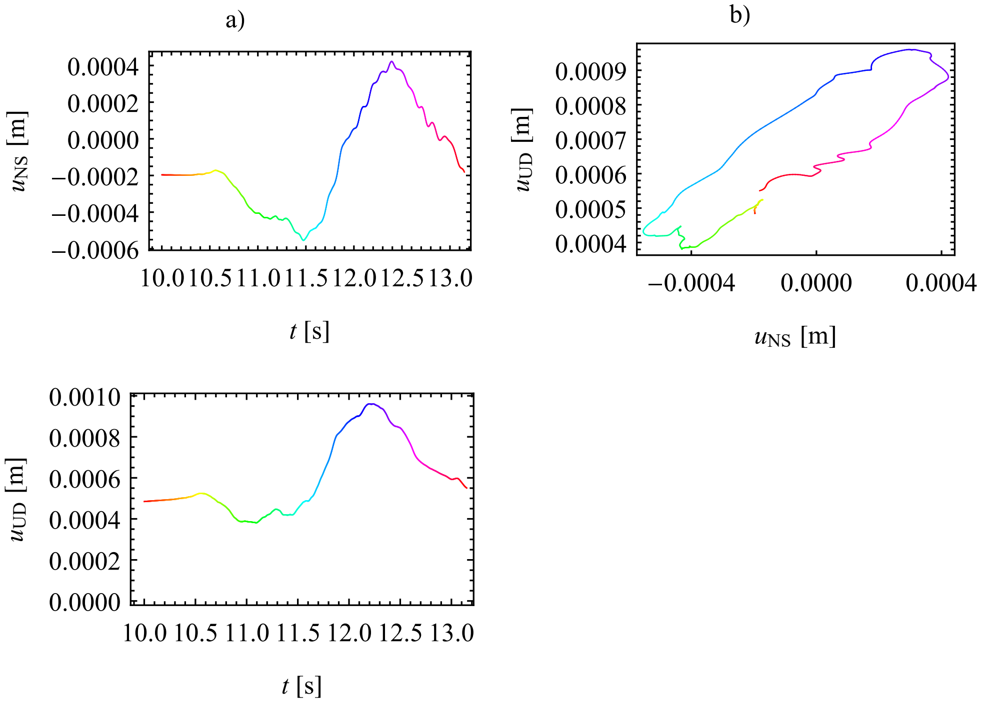


Figure 504.123: Ground displacements from Le Teil earthquake recorded at station 1, at Cruas NPP, for $t \in [10, 13.15]$ s: a) $u_{\mathrm{NS}}(t), u_{\mathrm{UD}}(t)$ and b) $u_{\mathrm{UD}}(u_{\mathrm{NS}})$.

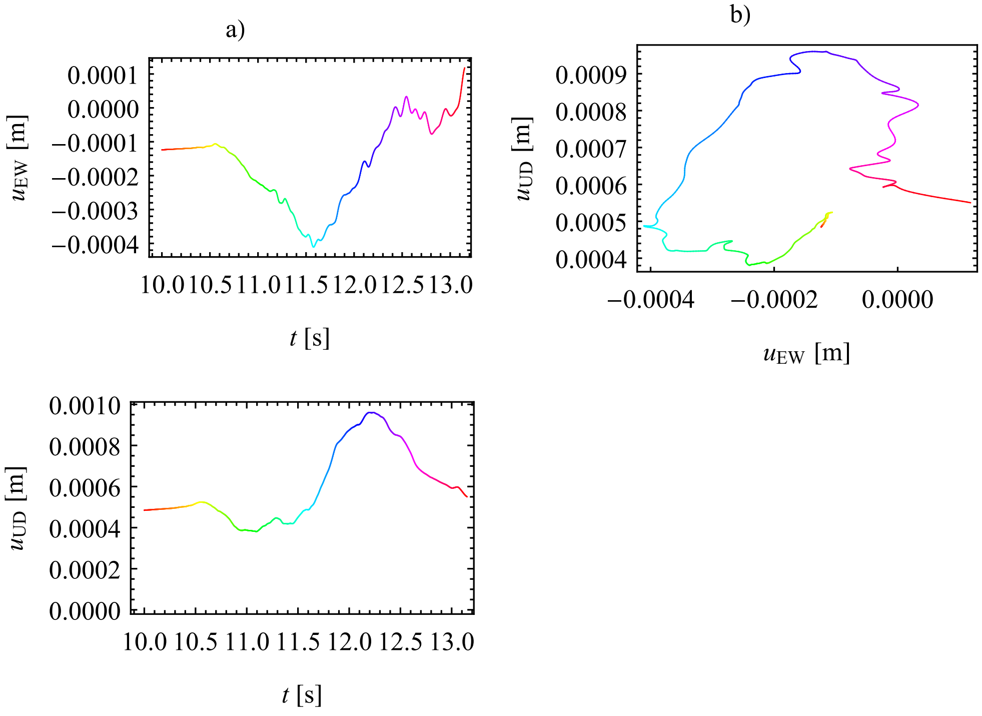


Figure 504.124: Ground displacements from Le Teil earthquake recorded at station 1, at Cruas NPP, for $t \in [10, 13.15]$ s: a) $u_{\mathrm{EW}}(t), u_{\mathrm{UD}}(t)$ and b) $u_{\mathrm{UD}}(u_{\mathrm{EW}})$.

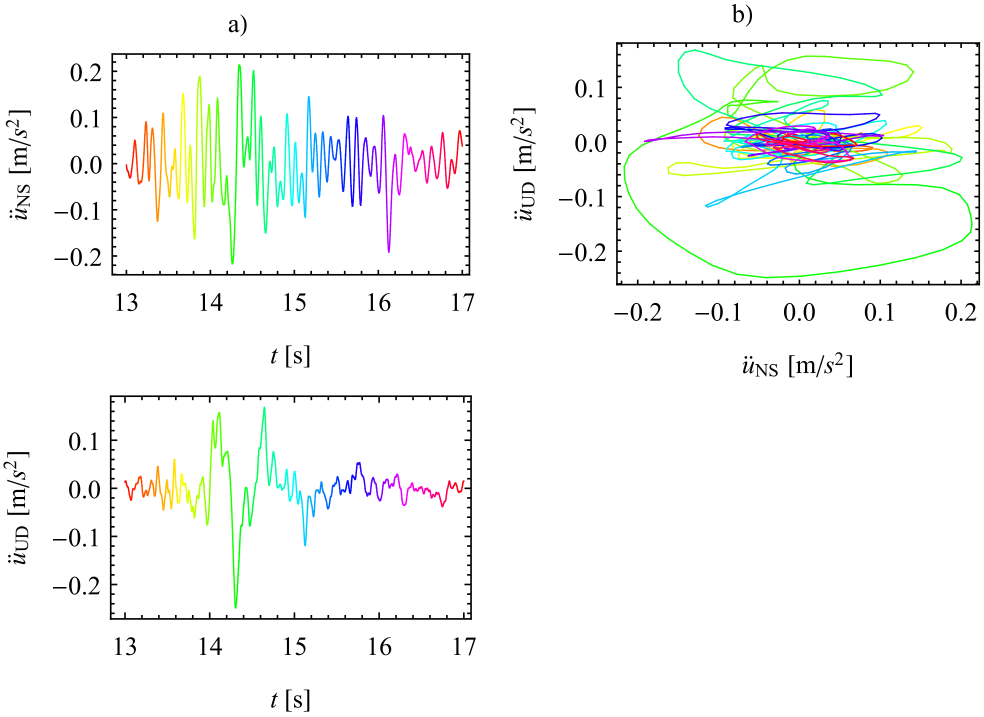


Figure 504.125: Ground accelerations from Le Teil earthquake recorded at station 1, at Cruas NPP, for $t \in [13, 17]$ s: a) $\ddot{u}_{\mathrm{NS}}(t), \ddot{u}_{\mathrm{UD}}(t)$ and b) $\ddot{u}_{\mathrm{UD}}(\ddot{u}_{\mathrm{NS}})$.

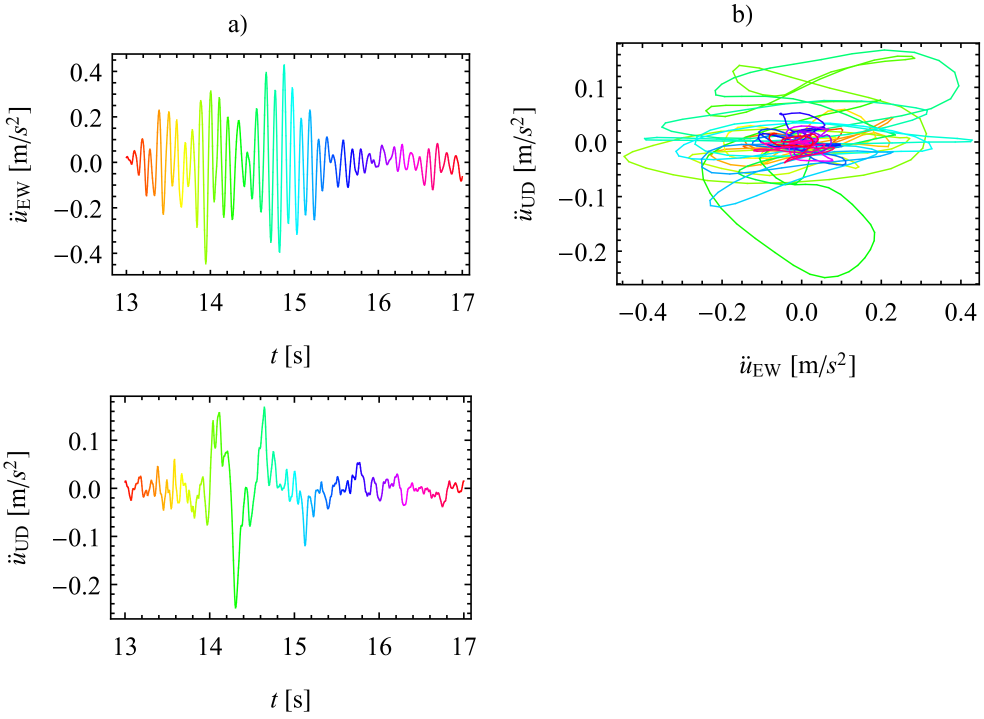


Figure 504.126: Ground accelerations from Le Teil earthquake recorded at station 1, at Cruas NPP, for $t \in [13, 17]$ s: a) $\ddot{u}_{\mathrm{EW}}(t), \ddot{u}_{\mathrm{UD}}(t)$ and b) $\ddot{u}_{\mathrm{UD}}(\ddot{u}_{\mathrm{EW}})$.

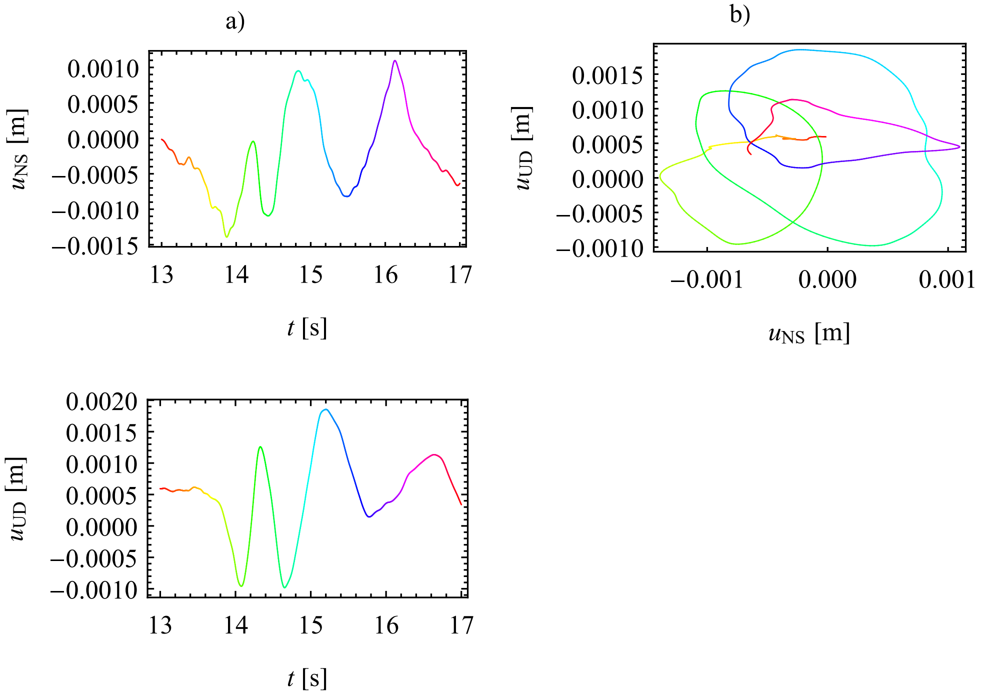


Figure 504.127: Ground displacements from Le Teil earthquake recorded at station 1, at Cruas NPP, for $t \in [13, 17]$ s: a) $u_{\mathrm{NS}}(t), u_{\mathrm{UD}}(t)$ and b) $u_{\mathrm{UD}}(u_{\mathrm{NS}})$.

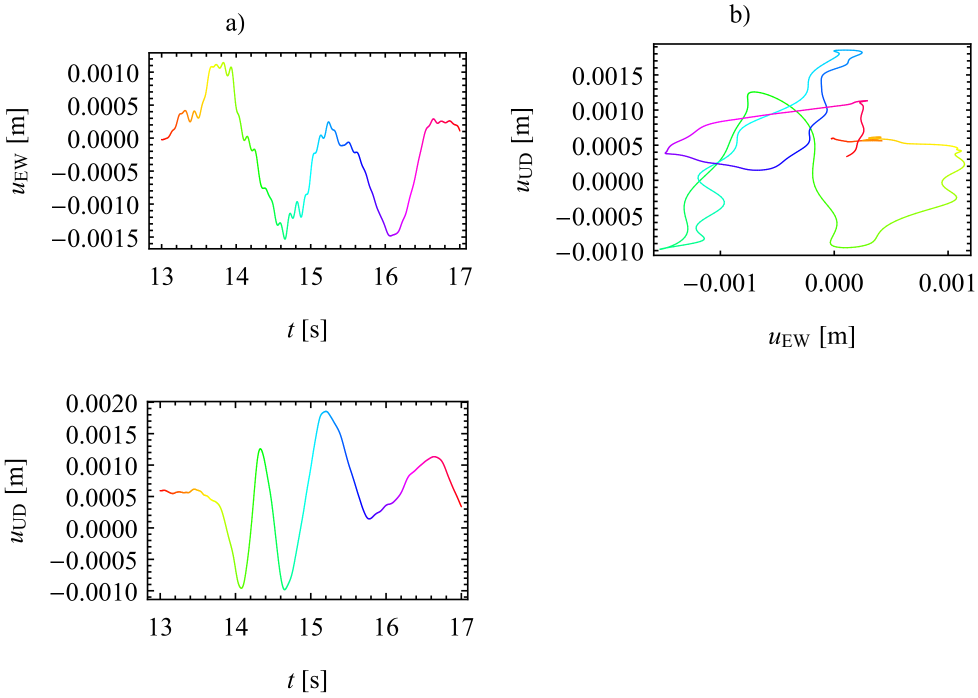


Figure 504.128: Ground displacements from Le Teil earthquake recorded at station 1, at Cruas NPP, for $t \in [13, 17]$ s: a) $u_{\mathrm{EW}}(t), u_{\mathrm{UD}}(t)$ and b) $u_{\mathrm{UD}}(u_{\mathrm{EW}})$.

504.10.5.1 Estimation of the Seismic Input

Seismic loads on ESSI systems are generally uncertain. Different earthquake scenarios and regional geology are probable. A parametric FE simulation of the mechanical behavior of an ESSI system using models at the regional scale would be required. It is usually quite problematic due to large computation time, however. Apart from that, unrealistic reflections of the waves at the model boundaries cannot be avoided (Baffet et al., 2012). Even larger FE models, and computation time, would be needed to ensure that the obtained results are not corrupted by the resulting so-called spurious waves. Hence it is advantageous to test ESSI systems parametrically using models at the local scale.

Different simplified approaches have been used in the literature (Semblat and Pecker, 2009; Zienkiewicz et al., 1999b) in order to shorten the computation time in FE solutions of the seismic wave propagation in ESSI systems. These methods rely on reduction of the size of the model and replacement of the distant earthquake load with a *proxy seismic load* applied at the local scale using some simplifying assumptions. Here, the *domain reduction method* (DRM) (Bielak et al., 2003a) is adapted. The proxy seismic input, also called the *DRM load*, will be used to "stress"-test Cruas NPP efficiently and fairly realistically.

An earthquake load usually arrives at the local site in the form of a plane seismic wave. This is due to the large distance between the source and the local site. The inclination of this incident plane wave has been shown, e.g., in (Hori, 2006; Teisseyre et al., 2006; Trifunac, 1982), to have significant influence on the mechanical behavior of ESSI systems. For this reason, Cruas NPP will be tested under an inclined plane wave, an incoherent wave. The DRM load will be obtained using the *Thomson-Haskell matrix method* (THMM) (Haskell, 1953). It is a closed-form solution of the propagation of an inclined monochromatic plane wave in a 3D linear elastic medium with horizontal isotropic layers.

Domain Reduction Method (DRM) The DRM (Bielak et al., 2003a; Yoshimura et al., 2003a) allows to significantly shorten the computation time in a FE solution of the seismic wave propagation in an ESSI system. However, some simplifying assumptions are needed. It is assumed that the real seismic ground motions do not differ significantly from the approximate ground motions. These approximate seismic ground motions correspond to:

- the *far field* simplified into the linear elastic medium and
- the *local site* simplified into the linear elastic *free field*, i.e., without the structure.

The earthquake load at the source is replaced by a proxy load at the local site. This proxy load is also called the DRM load. It can be calculated analytically, possibly with additional simplifying assumptions. Here, a plane wave is assumed to arrive at Cruas NPP, the local site, and the THMM is used.

It will be shown sufficient to apply the proxy load to solely a single layer of elements around the FE domain at the local scale. In this way, the regional domain can be remarkably reduced. The single layer of elements is called the *DRM layer*. The formulation of the DRM is given in what follows.

Global System of Equations We recall the global system of equations for dynamics, here, without an artificial damping,

$$K_{\alpha i\beta j}\, U_{\beta j} + M_{\alpha i\beta j}\, \ddot{U}_{\beta j} = F^{\text{ext}}_{\alpha i} \quad \text{at given } t\,. \tag{504.17}$$

Eq. (504.17) describes the discretized, in space only, weak form of the dynamic equilibrium equation (EE) and it is solved numerically in a FE calculation for a given time, t. The global tangent stiffness matrix is $K_{\alpha i\beta j}$. It stores the contributions from all elements and, generally, it can be either elastic, $K^{\text{el}}_{\alpha i\beta j}$, or elasto-plastic, $K^{\text{el-pl}}_{\alpha i\beta j}$. The Greek indices, $\alpha, \beta = 1, 2, \ldots, N$, correspond to the global node numbers and the Latin indices, $i, j = 1, 2, 3$, denote the degrees of freedom (DOFs). The displacement at node β in direction j is $U_{\beta j}$. Analogously, $M_{\alpha i\beta j}$ is the constant global mass matrix that stores the contributions from all elements. The acceleration at node β in direction j is $\ddot{U}_{\beta j}$. The external force at node α in direction i is $F^{\text{ext}}_{\alpha i}$.

Global System of Equations in the Regional Model The regional domain includes the source of the earthquake and it is denoted as R. The local-site domain, with the structure, is denoted as L, Fig. 504.129(a). Let us write (504.17) for R, L and the boundary, Γ, between R and L,

$$\begin{bmatrix} K^{\text{el}}_{\alpha i\beta j} & K^{\text{el}}_{\alpha i\mathsf{b}j} & 0_{\alpha i\mathsf{B}j} \\ K^{\text{el}}_{\mathsf{a}i\beta j} & K^{\text{R el}}_{\mathsf{a}i\mathsf{b}j} + K^{\text{L el--pl}}_{\mathsf{a}i\mathsf{b}j} & K^{\text{el--pl}}_{\mathsf{a}i\mathsf{B}j} \\ 0_{\mathsf{A}i\beta j} & K^{\text{el--pl}}_{\mathsf{A}i\mathsf{b}j} & K^{\text{el--pl}}_{\mathsf{A}i\mathsf{B}j} \end{bmatrix} \cdot \begin{Bmatrix} U_{\beta j} \\ U_{\mathsf{b}j} \\ U_{\mathsf{B}j} \end{Bmatrix} + \begin{bmatrix} M_{\alpha i\beta j} & M_{\alpha i\mathsf{b}j} & 0_{\alpha i\mathsf{B}j} \\ M_{\mathsf{a}i\beta j} & M^{\text{R}}_{\mathsf{a}i\mathsf{b}j} + M^{\text{L}}_{\mathsf{a}i\mathsf{b}j} & M_{\mathsf{a}i\mathsf{B}j} \\ 0_{\mathsf{A}i\beta j} & M_{\mathsf{A}i\mathsf{b}j} & M_{\mathsf{A}i\mathsf{B}j} \end{bmatrix} \cdot \begin{Bmatrix} \ddot{U}_{\beta j} \\ \ddot{U}_{\mathsf{b}j} \\ \ddot{U}_{\mathsf{B}j} \end{Bmatrix} = \begin{Bmatrix} F^{\text{ext}}_{\alpha i} \\ 0_{\mathsf{a}i} \\ 0_{\mathsf{A}i} \end{Bmatrix}, \tag{504.18}$$

wherein $K^{\text{R el}}_{\mathsf{a}i\mathsf{b}j}, M^{\text{R}}_{\mathsf{a}i\mathsf{b}j}$ correspond to the elements in R and $K^{\text{L el--pl}}_{\mathsf{a}i\mathsf{b}j}, M^{\text{L}}_{\mathsf{a}i\mathsf{b}j}$ correspond to the elements in L. The Greek indices, $\alpha, \beta = 1, 2, \ldots, N$, denote the nodes in R. The lower-case sans-serif letters, $\mathsf{a}, \mathsf{b} = 1, 2, \ldots, N$, denote the nodes at Γ and the upper-case ones, $\mathsf{A}, \mathsf{B} = 1, 2, \ldots, N$, denote the nodes in L. Domain R is much, say, $10^3\times$, larger than domain L. The Neumann BC, $F^{\text{ext}}_{\alpha i}$, is prescribed to the nodes in R only. The materials in R are assumed to be linear elastic. The material behavior in L may be strongly non-linear. Hence L may require some sophisticated constitutive description and a fine discretization, both, in space and time. Computational cost of this regional model is mainly related to the size of domain R and to the material non-linearity inside domain L.

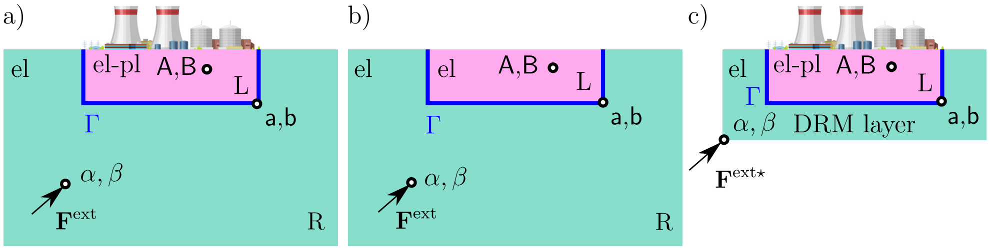


Figure 504.129: DRM: a) regional model, b) regional model with the simplified local site and c) reduced model. L is the local-site domain, R is the regional domain and Γ is the boundary between L and R. The BC, $\mathbf{F}^{\text{ext}}$, is replaced by the proxy load, $\mathbf{F}^{\text{ext}\star}$, applied to the DRM layer only.

Global System of Equations in the Regional Model with the Simplified Local Site Let us temporarily remove the structure from domain L, Fig. 504.129(b), and assume the linear elastic behavior within L. We rewrite (504.18) using index $\sqcup^{\star}$ for a quantity, $\sqcup$, that corresponds to the regional model with the simplified local site. The following holds.

$$\begin{bmatrix} K^{\text{el}}_{\alpha i\beta j} & K^{\text{el}}_{\alpha i\mathsf{b}j} & 0_{\alpha i\mathsf{B}j} \\ K^{\text{el}}_{\mathsf{a}i\beta j} & K^{\text{R el}}_{\mathsf{a}i\mathsf{b}j}+K^{\text{L el}}_{\mathsf{a}i\mathsf{b}j} & K^{\text{el}}_{\mathsf{a}i\mathsf{B}j} \\ 0_{\mathsf{A}i\beta j} & K^{\text{el}}_{\mathsf{A}i\mathsf{b}j} & K^{\text{el}}_{\mathsf{A}i\mathsf{B}j} \end{bmatrix} \cdot \begin{Bmatrix} U^{\star}_{\beta j} \\ U^{\star}_{\mathsf{b}j} \\ U^{\star}_{\mathsf{B}j} \end{Bmatrix} + \begin{bmatrix} M_{\alpha i\beta j} & M_{\alpha i\mathsf{b}j} & 0_{\alpha i\mathsf{B}j} \\ M_{\mathsf{a}i\beta j} & M^{\text{R}}_{\mathsf{a}i\mathsf{b}j}+M^{\text{L}\star}_{\mathsf{a}i\mathsf{b}j} & M^{\star}_{\mathsf{a}i\mathsf{B}j} \\ 0_{\mathsf{A}i\beta j} & M^{\star}_{\mathsf{A}i\mathsf{b}j} & M^{\star}_{\mathsf{A}i\mathsf{B}j} \end{bmatrix} \cdot \begin{Bmatrix} \ddot{U}^{\star}_{\beta j} \\ \ddot{U}^{\star}_{\mathsf{b}j} \\ \ddot{U}^{\star}_{\mathsf{B}j} \end{Bmatrix} = \begin{Bmatrix} F^{\text{ext}}_{\alpha i} \\ 0_{\mathsf{a}i} \\ 0_{\mathsf{A}i} \end{Bmatrix} \tag{504.19}$$

The simplification of L does not influence either the stiffness matrices or the mass matrices corresponding to the elements in R. Neither the BC, $F^{\text{ext}}_{\alpha i}$, is affected.

Material in the far field is assumed linear elastic independently of the temporary simplification of L, so the displacement and its time derivates are additive, that is,

$$\Delta U_{\beta j} = U_{\beta j} - U^{\star}_{\beta j} \quad \text{and} \quad \Delta \ddot{U}_{\beta j} = \ddot{U}_{\beta j} - \ddot{U}^{\star}_{\beta j} \tag{504.20}$$

hold in R.

Global System of Equations in the Reduced Model The force

$$F^{\text{ext}}_{\alpha i} = K^{\text{el}}_{\alpha i\beta j} U^{\star}_{\beta j} + K^{\text{el}}_{\alpha i\mathsf{b}j} U^{\star}_{\mathsf{b}j} + M_{\alpha i\beta j} \ddot{U}^{\star}_{\beta j} + M_{\alpha i\mathsf{b}j} \ddot{U}^{\star}_{\mathsf{b}j} \tag{504.21}$$

from (504.19) can be substituted together with (504.20) into (504.18) and one obtains the following system of equations.

$$\begin{bmatrix} K^{\mathrm{el}}_{\alpha i\beta j} & K^{\mathrm{el}}_{\alpha i\mathrm{b}j} & 0_{\alpha i\mathrm{B}j} \\ K^{\mathrm{el}}_{\mathrm{a}i\beta j} & K^{\mathrm{R\,el}}_{\mathrm{a}i\mathrm{b}j} + K^{\mathrm{L\,el-pl}}_{\mathrm{a}i\mathrm{b}j} & K^{\mathrm{el-pl}}_{\mathrm{a}i\mathrm{B}j} \\ 0_{\mathrm{A}i\beta j} & K^{\mathrm{el-pl}}_{\mathrm{A}i\mathrm{b}j} & K^{\mathrm{el-pl}}_{\mathrm{A}i\mathrm{B}j} \end{bmatrix} \cdot \begin{Bmatrix} \Delta U_{\beta j} \\ U_{\mathrm{b}j} \\ U_{\mathrm{B}j} \end{Bmatrix} + \begin{bmatrix} M_{\alpha i\beta j} & M_{\alpha i\mathrm{b}j} & 0_{\alpha i\mathrm{B}j} \\ M_{\mathrm{a}i\beta j} & M^{\mathrm{R}}_{\mathrm{a}i\mathrm{b}j} + M^{\mathrm{L}}_{\mathrm{a}i\mathrm{b}j} & M_{\mathrm{a}i\mathrm{B}j} \\ 0_{\mathrm{A}i\beta j} & M_{\mathrm{A}i\mathrm{b}j} & M_{\mathrm{A}i\mathrm{B}j} \end{bmatrix} \cdot \begin{Bmatrix} \Delta \ddot{U}_{\beta j} \\ \ddot{U}_{\mathrm{b}j} \\ \ddot{U}_{\mathrm{B}j} \end{Bmatrix} = \begin{Bmatrix} K^{\mathrm{el}}_{\alpha i\mathrm{b}j} U^{\star}_{\mathrm{b}j} + M_{\alpha i\mathrm{b}j} \ddot{U}^{\star}_{\mathrm{b}j} \\ -K^{\mathrm{el}}_{\mathrm{a}i\beta j} U^{\star}_{\beta j} - M_{\mathrm{a}i\beta j} \ddot{U}^{\star}_{\beta j} \\ 0_{\mathrm{A}i} \end{Bmatrix} \tag{504.22}$$

The RHS of (504.22),

$$\begin{Bmatrix} F^{\mathrm{ext}\star}_{\alpha i} \\ F^{\mathrm{ext}\star}_{\mathrm{a}i} \\ 0_{\mathrm{A}i} \end{Bmatrix} = \begin{Bmatrix} K^{\mathrm{el}}_{\alpha i\mathrm{b}j} U^{\star}_{\mathrm{b}j} + M_{\alpha i\mathrm{b}j} \ddot{U}^{\star}_{\mathrm{b}j} \\ -K^{\mathrm{el}}_{\mathrm{a}i\beta j} U^{\star}_{\beta j} - M_{\mathrm{a}i\beta j} \ddot{U}^{\star}_{\beta j} \\ 0_{\mathrm{A}i} \end{Bmatrix}, \tag{504.23}$$

can be used as the BC instead of $\mathbf{F}^{\mathrm{ext}}$ from (504.18).

It is evident in (504.23) that the proxy forces, $\mathbf{F}^{\mathrm{ext}\star}$, take on non-zero values for elements with nodes a, b, in domain R, or both, a, b, in domain R, and α, β, at boundary Γ, only. This means that $\mathbf{F}^{\mathrm{ext}\star}$ can be applied to solely the single layer of elements adjacent to Γ, namely, the DRM layer, and hence the model can be reduced, Fig. 504.129(c).

The discrete proxy load, $\mathbf{F}^{\mathrm{ext}\star}$, is only a rough estimation of the real seismic load because it corresponds to the simplified local site. Hence using the DRM, one saves the computation time at the cost of the reliability of the obtained results.

This $\mathbf{F}^{\mathrm{ext}\star}$ is obtained in Real-ESSI Simulator from an inclined plane wave in horizontally layered soil using the THMM.

Thomson-Haskell Matrix Method (THMM) The propagation of an inclined plane wave in a linear elastic ground with horizontal isotropic layers is solved analytically using the Thomson-Haskell matrix method (THMM) (Haskell, 1953; Thomson, 1950). In this way, reflections and refractions of the wave at interfaces between the layers are captured. Also, the Rayleigh wave is taken into account. The use of a closed-form solution eliminates the problem of unrealistic reflections of the wave at the boundaries of the FE model.

The THMM holds for monochromatic waves only. If the wave is non-monochromatic, it can be decomposed into harmonic components with the Fourier analysis. The THMM is then used for each monochromatic component and the solutions are synthesized using, again, the Fourier analysis.

The analytical solution of the wave propagation is generally established from the boundary conditions at the interfaces between the layers and at the surface. The Helmholtz decomposition theorem is incorporated and the wave field is separated into two parts, i.e., the part pertaining to the purely volumetric deformation and the part related to the pure rotation of material particles (Nowacki, 1974). The Snell law describes reflections and refractions of the waves at the interfaces between the layers and at the surface. In the THMM, the governing equations are conveniently rewritten into a matrix form to ease the computation of the wave propagation.

A comprehensive description of the implementation of the THMM into Real-ESSI Simulator is given in (Wang et al., 2021).

504.10.5.2 Preliminary 2D Simulation of Le Teil Earthquake using DRM with THMM

Rock is isotropic, linear elastic, E, ν with $E = 2\rho V_s^2(1+\nu)$, Table 504.11.

Table 504.11: Material properties in preliminary 2D simulation of the earthquake in Le Teil using DRM with THMM

Material	z m	ρ kg/m^3	V_s m/s	ν
Rock	$[0, -\infty]$	2500	1600	0.4

Time functions of the horizontal ground acceleration, $\ddot{u}_x(t)$, and the vertical ground acceleration, $\ddot{u}_z(t)$, at $\boldsymbol{x} = \{140, 0, 0\}$ m is plotted in Figure 504.130(a). The corresponding hodograph, $u_z(u_x)$, is shown in Figure 504.130(b).

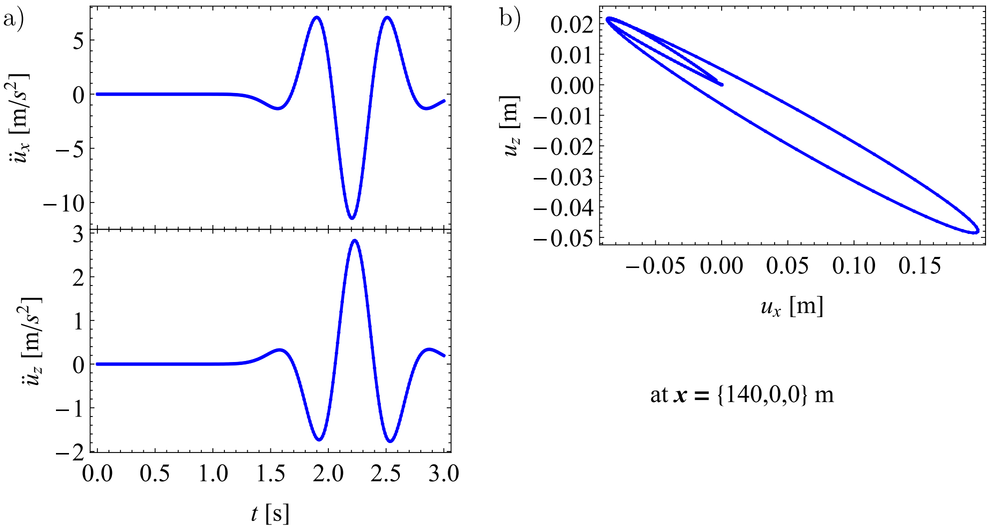


Figure 504.130: Preliminary 2D simulation of the earthquake in Le Teil using DRM with THMM: a) $\ddot{u}_x(t)$ and $\ddot{u}_z(t)$, and b) $u_z(u_x)$ at $\boldsymbol{x} = \{140, 0, 0\}$ m.

504.10.6 Cruas Nuclear Power Plant (NPP)

Base isolators are located at approximately -9.0 to -12.0 m. This means that the NPP is founded on rock, see the soil profile from Table 504.9 on page 2570.

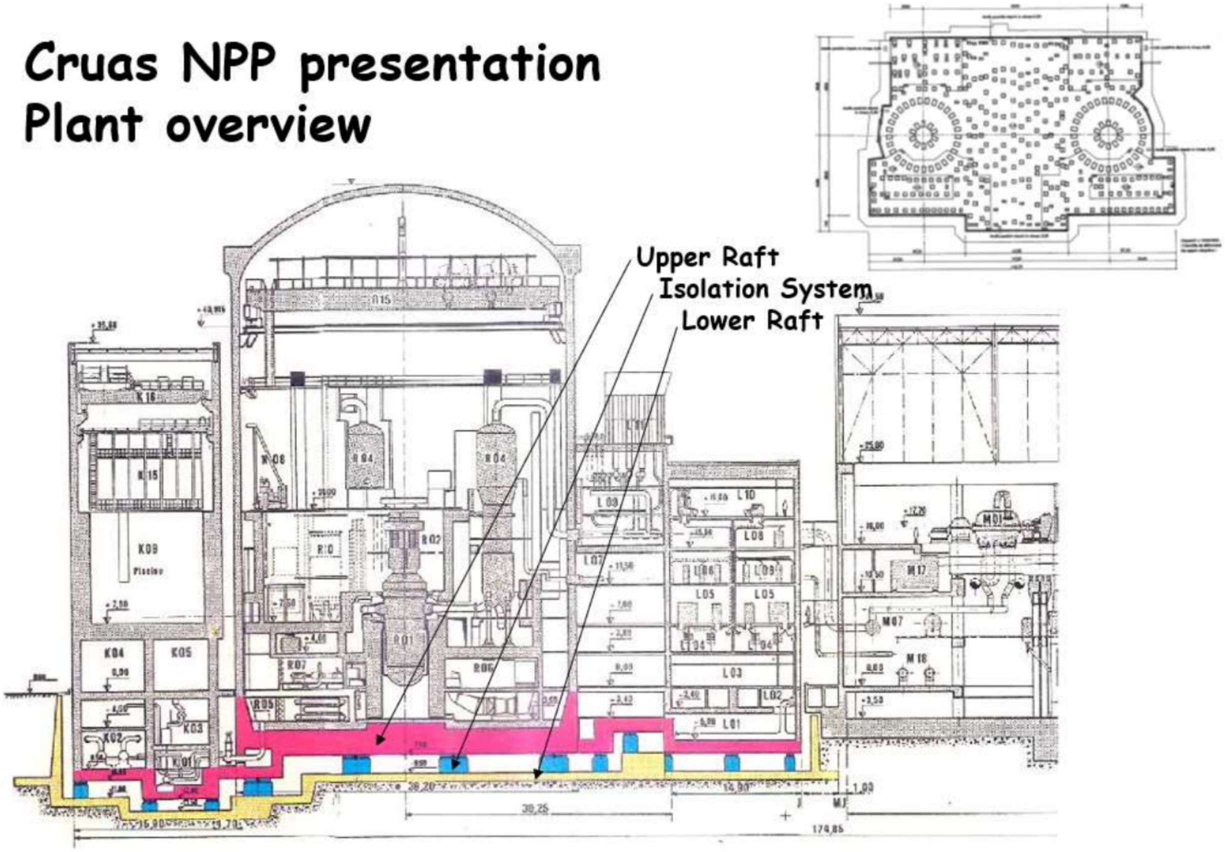


Figure 504.131: Cruas NPP, section 01, according to Viallet et al. (2022).

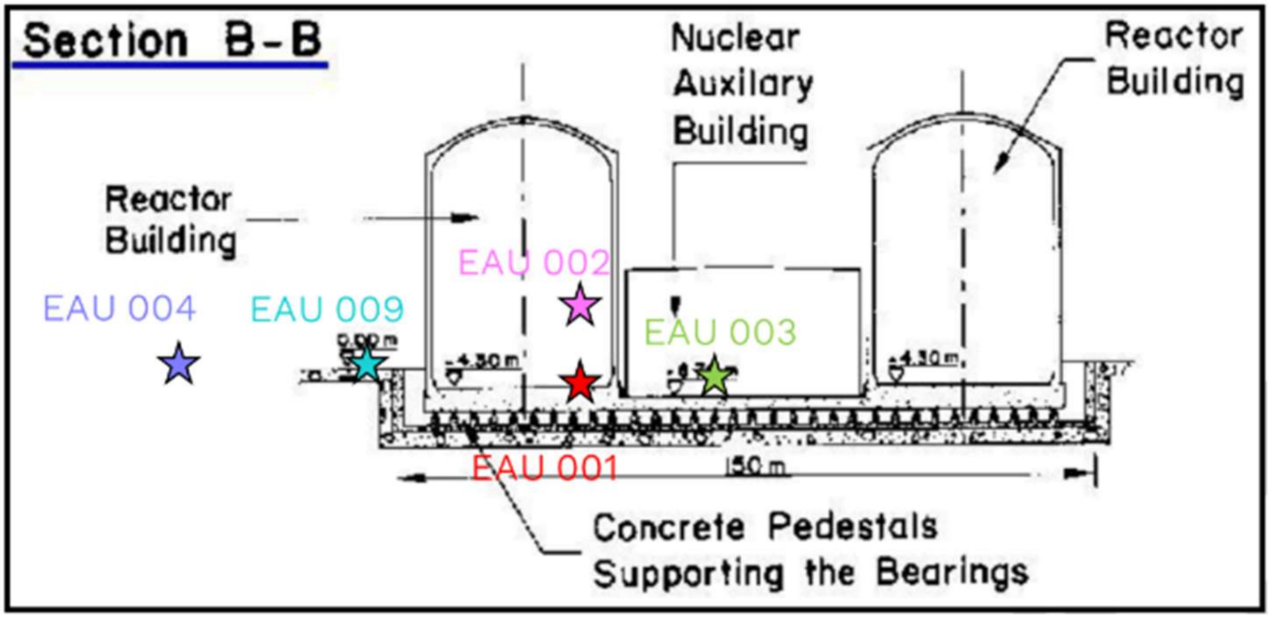


Figure 504.132: Cruas NPP, section 02, according to Viallet et al. (2022).

504.10.7 FE Simulation of Cruas NPP under Seismic Load from Le Teil Earthquake

...

504.10.7.1 Verification of the Simulation

FE simulation of an ESSI system requires one to specify:

- the domain of simulation (both, in space and in time)
- the boundary conditions (BCs)
- the initial conditions (ICs)
- the material laws
- the material parameters
- the discretization (in space and in time) and
- the numerical procedures used in the calculation,

wisely. A parametric study using a simple FE model may be helpful in understanding the influence of the above aspects on the results of the simulation. This influence can be additionally quantified in a sensitivity study (Sobol, 2001). Also, in the case of a simple model, the numerical solution can be compared with the corresponding analytical one. In this way, a complex FE model can be verified.

Element Tests The mechanical behavior of chosen critical parts of Cruas NPP (being an ESSI system) is investigated in suitable element tests. They are conducted on different structural materials and on different soils with various:

- BCs (displacement, seismic load)
- ICs (initial stress and strain)
- material laws
- material parameters.

The resulting evolution of the stress, $\boldsymbol{\sigma}(\mathbf{x},t)$, and strain, $\varepsilon(\mathbf{x},t)$, fields is examined.

Beam Element Tests on Reinforced Concrete A cantilever beam is tested on reinforced concrete with different material laws for concrete and for steel in:

- uniaxial compression or extension, Figure ...
- uniaxial bending, Figure ...
- biaxial bending, Figure ...
- torsion, Figure ....

A (displacement-based) *fiber beam-column element* similar to the one from (Taucer et al., 1991) is used. It is called `BeamColumnDispFiber3d_Corotational`. The co-rotational formulation by (Crisfield, 1990) is implemented. The local coordinate system co-rotates with the element during a rigid body rotation and hence `BeamColumnDispFiber3d_Corotational` is applicable also in the case of the *geometric non-linearity*, i.e., when *large deformations* are involved. Otherwise, during a large deformation with a significant rigid body rotation, an excessive stiffness might be obtained in such beam element.
The element has two nodes and the user defines the number of Gauß points along the element.

`BeamColumnDispFiber3d_Corotational` may represent multiple 1D so-called *fibers*. They are simply connected in parallel, Figure 504.133. A fiber is defined by its cross-section, A, and location in the element cross-section, $\{y, z\}$. The cross-section is constant along the beam.

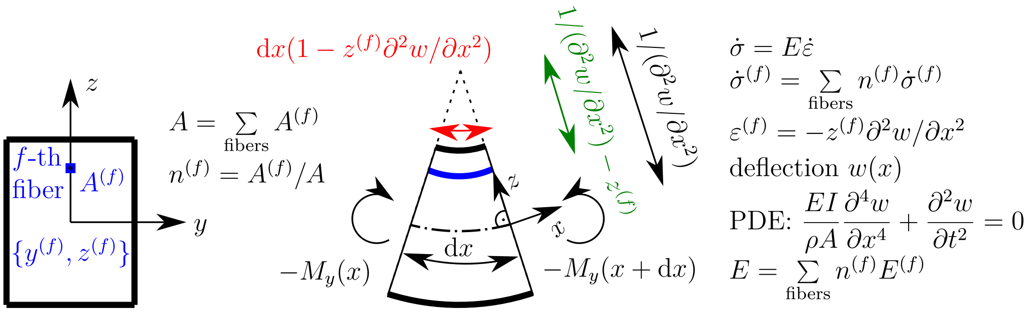


Figure 504.133: Fiber beam-column element in uniaxial bending: The second moment of area, I, pertains to the whole cross-section and the density, ρ, describes the average density of the material.

Each fiber has a prescribed material law. Different material laws can be prescribed to different fibers. The strain in a single fiber is obtained assuming that the cross-section remains plane and normal to the beam axis during the deformation. This means that `BeamColumnDispFiber3d_Corotational` is an *Euler-Bernoulli beam*. The shear stress is neglected which is allowed in beams for which the cross-sectional dimensions are much smaller than the length, see, e.g., (Nowacki, 1974).

However, the cross-section is additionally (independently of the fibers) described by the product, GJ, of the shear modulus, G, and the torsional constant, J. Hence the element has four nodal degrees of freedom (DOFs) altogether, i.e., three translations and the angle of twist due to the torsional moment, M_x.

Shell Element Tests on Concrete ...

Solid Element Tests on Concrete Different constitutive laws for concrete are tested in:

- triaxial compression, Figure 504.134(a),
- simple shear, Figure 504.134(b),
- pure shear (here, understood as isochoric shear with no rotation of the principal stress and strain directions), Figure 504.134(c).

The 3D solid brick element, `8NodeBrick`, is used.

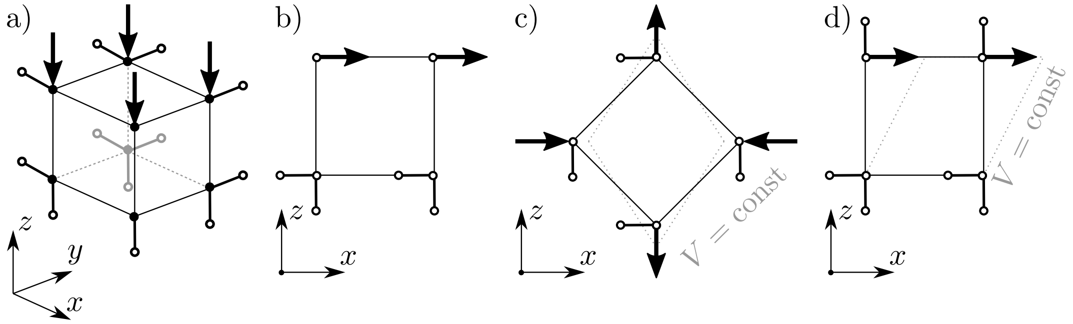


Figure 504.134: Solid element tests: a) triaxial compression, b) simple shear, c) pure shear (isochoric with no rotation of the principle stress and strain directions), d) pure shear (in soil mechanics, commonly understood as isochoric shear).

Solid Element Tests on Soil Similarly as in the case of the solid element tests on concrete, different constitutive laws for soils are tested in:

- triaxial compression, Figure 504.134(a),
- simple shear, Figure 504.134(b),

- pure shear (here, understood as isochoric shear with no rotation of the principal stress and strain directions), Figure 504.134(c).

The 3D solid brick element, `8NodeBrick`, is used.

Interface Element Tests …

Seismic Isolator and Dissipator Element Tests …

Report for Phase 2 of Stage 1, Simulation of the Ground Motions

Ground motions from the Le Teil earthquake had been intensively studied within the SIGMA2 project[6] (Viallet et al., 2022). Two types of methods for the simulation of seismic motions had been used, the empirical methods and the physically based methods. The latter include both, 1D (Fasan, 2016; Fasan et al., 2016; Hassan et al., 2020; Magrin et al., 2016; Magrin, 2012; Panza et al., 2012) and 3D (Mazzieri et al., 2013; Paolucci et al., 2020, 2021) simulations. Some recent 3D numerical calculations of seismic motions from the Le Teil earthquake are reported in (Lehmann et al., 2023; Smerzini et al., 2023).

Seismic input due to the Le Teil earthquake is needed to simulate the inelastic behavior of the Cruas NPP. A FE simulation of the wave field from the earthquake is conducted using the Real-ESSI Simulator (Jeremić et al., 1988-2025). Unfortunately, the knowledge about the local and regional geology is very limited. Soil and rock profiles are available at three locations only. This scarcity of data precludes the preparation of a realistic model at the regional scale. Hence an accurate reproduction of the seismic wave field from the earthquake in Le Teil is impossible.

Based on a preliminary simplified FE simulation and the recent work by Viallet et al. (2022), some important observations are made about the studied seismic wave field.

- The preliminary simulation using a simple 2D model with a shallow point source reveals a possible occurence of the Stoneley wave and the Rayleigh wave. This should, indeed, be expected from a shallow earthquake, particularly with the rupture that achieves the surface.
- Based on Viallet et al. (2022), the horizontal and vertical components of ground motions at the Cruas NPP occur almost simultaneously. We interprete these motions as the overlapping Rayleigh and Stoneley waves.
- Earthquake loads are usually simulated as one dimensional (1D), one component (1C) waves, possibly superposed, wherein 1D pertains to the direction of propagation, either vertical or horizontal, while 1C corresponds to the direction of polarization, analogously, either vertical or horizontal. The probable presence of the Rayleigh wave and the Stoneley wave excludes this common approach to modelling of seismic loads. In the case of the surface and interface waves, the simulated seismic wave field is shown to be significantly different than the one obtained using the common approach. The seismic response of a structure will also be different depending on the seismic input. The difference is most evident in the case of shorter wave lengths, that is, for higher wave frequencies.

---

[6] `https://www.sigma-2.net/`

These and other observations will be presented during the next SMATCH benchmark meeting in November 2023.

504.10.8 Finite Element Models

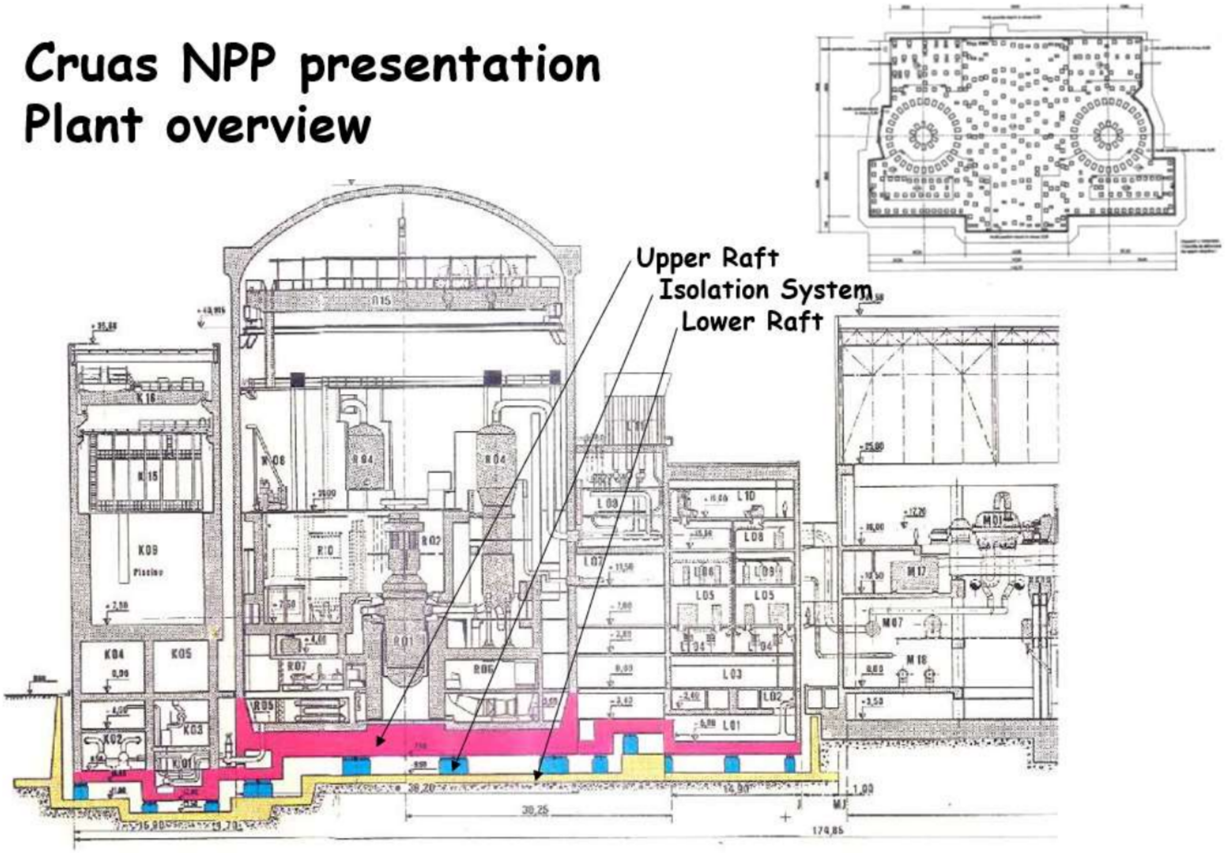


Figure 504.135: Cruas NPP, section 01, according to Viallet et al. (2022).

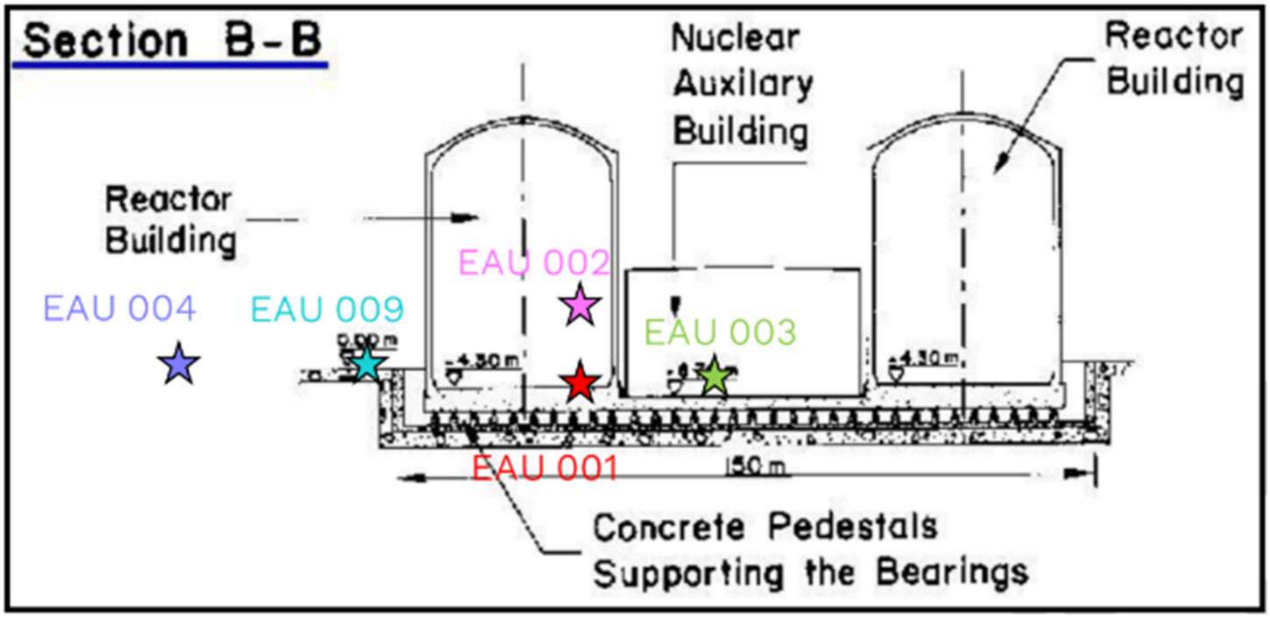
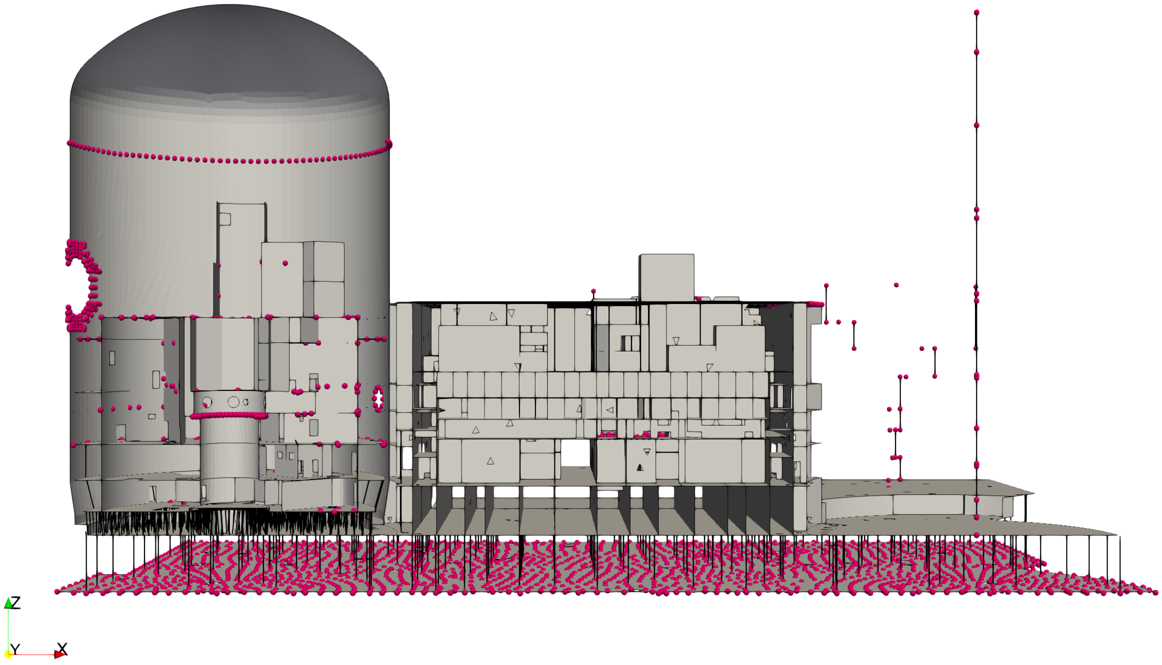
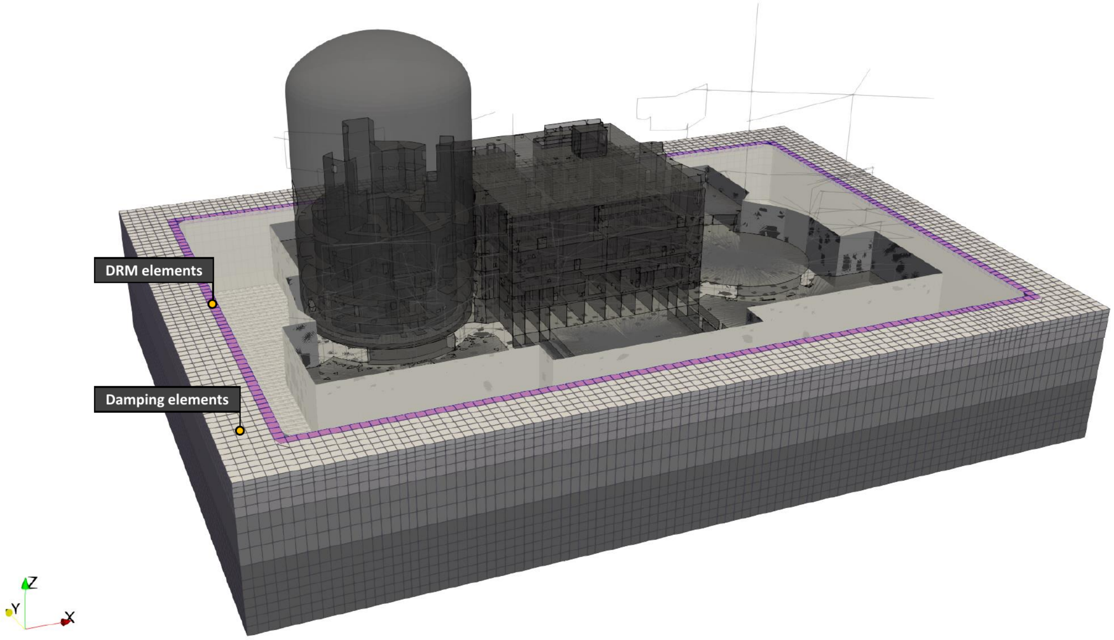


Figure 504.136: Cruas NPP, section 02, according to Viallet et al. (2022).




DRM elements
Damping elements
Z
Y
X

Chapter 505

Liquefaction and Cyclic Mobility

(2002-2006-2009-2021-)

(In collaboration with Dr. Zhao Cheng, and Dr. Panagiota Tasiopoulou)

505.1 Chapter Summary and Highlights

505.2 Introduction

Liquefaction is one of the most complex phenomena in earthquake engineering. Liquefaction also represents one of the biggest contributors to damage of constructed facilities during earthquakes (Kramer, 1996a). Prediction of behaviors of liquefiable soils is difficult but achievable. There are number of methods that can be utilized to predict such behaviors. Methods currently used can have varying prediction accuracy and certainty. Of particular interest in this paper is the description of verified and validated numerical simulation methodology based on rational mechanics that is used to model, simulate and predict behavior of a single pile in liquefiable soil subjected to seismic loading. Both level and sloping ground pile systems are modeled and simulated. Detailed description of background theory, formulation and implementation were recently given by Cheng et al. (2007) and Jeremić et al. (2008).

It should be noted that presented development does show great promise in analyzing a myriad of liquefaction related problems in geotechnical and structural engineering. The effectiveness and power of numerical simulation tools for analyzing liquefaction problems becomes even more important and prominent in the light of potential disadvantages of models used in experimental simulations. These disadvantages, related to proper scaling (Wood, 2004) and problems in maintaining appropriate similarities (Harris and Sabnis, 1999) for first order important phenomena, can render scaled models ineffective, when used for physical simulations (under one-step or multiple-step gravity loading).

In what follows, a brief literature review is provided. The literature review comprises sections on observations of liquefaction behavior in case studies, non–continuum modeling efforts, review of redistribution of voids and pore fluid volume/pressures phenomena and continuum modeling efforts.

Observation of Behavior. Liquefaction behavior was observed during a number of earthquakes in the past. During Alaskan Earthquake (1964), liquefaction was the main cause of severe damage to 92 highway bridges, moderate to light damage to another 49 highway bridges, and moderate to sever damage to 75 railroad bridges (Youd and Bartlett, 1989). During Niigata Earthquake (1964) liquefaction induced damage to foundation piles under Yachiya bridge (Hamada, 1992). During that same earthquake, girders of Showa Bridge toppled as the support structure and piles moved excessively due to liquefaction (Japanese Society of Civil Engineers, 1966). During Kobe Earthquake (1995), liquefaction was the primary cause of damage to many pile supported or caisson supported bridges and structures. For example, Shin–Shukugawa bridge was subjected to excessive pile foundation movement due to liquefaction (Yokoyama et al., 1997).

Opposed to these failures and collapses, there were a number of bridges with pile foundations that did not suffer much or even minor damage even though there was liquefaction around foundations. For example, pile foundations of the Landing Road Bridge in New Zealand performed quite well during Edgecumbe earthquake (1987) even with a significant liquefaction recorded (Berril et al., 1997; Dobry and Abdoun, 2001). In addition to that, Second Maya Bridge piles (large steel pipes) were not damaged during Kobe earthquake despite significant liquefaction in surrounding soils (Yokoyama et al., 1997).

Non–Continuum Modeling Efforts. Modeling and simulation of piles in liquefied grounds has been focus of a number of recent studies. The simple approach, based on scaling of p-y springs has been suggested early by Japanese Road Association (1980), Architectural Institutive of Japan (1988), Liu and Dobry (1995), Miura et al. (1989) and O'Rourke (1991). However, large inconsistencies with material parameter selection are present when p-y spring approach is used for piles in liquefied soils. Since p-y methodology for liquefied soils is not based on rational mechanics, appropriate choice of material parameters is primarily based on empirical observations of behaviors of piles in liquefied soils in experimental studies. A number of experimental studies have carefully examined pile behaviors in liquefiable soils. We mention Tokida et al. (1992), Liu and Dobry (1995), Abdoun et al. (1997), Horikoshi et al. (1998) and Boulanger and Tokimatsu (2006). Studies using physical model can be used to obtain very high quality data on behavior of piles in liquefied soils, provided that similarity of important physical phenomena is maintained (Wood, 2004; Harris and Sabnis, 1999). Some of the recent papers that discussed use of these models and gave recommendations about parameter choices are listed for reference: Tokimatsu and Asaka (1998), Martin et al. (2002), Dobry et al. (2003), Liyanapathirana and Poulos (2005), Rollins et al. (2005), Čubrinovski and Ishihara (2006), Brandenberg et al. (2007).

Redistribution of Voids and Pore Fluid Volume/Pressures. Mechanics of pile behavior in liquefiable grounds is based on the concept of redistribution of voids and pore fluid volume/pressures (RVPFVP). It should be emphasized that geomechanics phenomena of redistribution of voids – pore fluid volume/pressure is used here in purely mechanistic way. That is, RVPFVP is a phenomena that occurs in saturated soils and that phenomena is responsible for (is manifested in) liquefaction related soil behaviors with or without piles. This is noted as in some recent publications, RVPFVP terminology is explicitly used for problems of liquefaction induced failures of sloping grounds without piles. Our understanding of the RVPFVP phenomena is that RVFVP is responsible for many more facets of behavior of liquefied soils, rather than only failure of liquefied slopes.

The early investigation of the RVPFVP phenomena was related to the behavior of infinite slopes. For example, loss of shear strength in infinite slopes is one of the early understood manifestations of RVPFVP (Whitman, 1985; National Research Council, 1985; Malvick et al., 2006). Laboratory investigation of sand was also used to observe the RVPFVP phenomena (Casagrande and Rendon, 1978; Gilbert, 1984)

Continuum Modeling Efforts. Continuum based formulations for modeling liquefaction problems have been present for over two decades. In a landmark paper, Zienkiewicz and Shiomi (1984) presented three possible coupled formulations for modeling of soil skeleton – pore fluid problems. The most general and complete one is the so called u-p-U formulations while the other two, the u-p and the u-U have a number of restrictions on the domain of application. Here, the unknowns are the soil skeleton displacements u; the pore fluid (water) pressure p; and the pore fluid (water) displacements U. The u-p formulation captures the movements of the soil skeleton and the change of the pore pressure, and is the most simplistic one of the three mentioned above. This formulation neglects the differential accelerations of the pore fluid (it does account for acceleration of pore fluid together with soil skeleton, but not the differential one if it exists), and in one version neglects the compressibility of the fluid (assuming complete incompressibility of the pore fluid). In the case of incompressible pore fluid, the formulation requires special treatment of the approximation function (shape function) for pore fluid to prevent the volumetric locking (Zienkiewicz and Taylor, 2000). The majority of the currently available implementations are based on this formulation. For example Elgamal et al. (2002) and Elgamal et al. (2003) developed an implementation of the u-p formulation with the multi-surface plasticity model by Prevost (1985b), while Chan (1988) and Zienkiewicz et al. (1999a) used generalized theory of plasticity Pastor et al. (1990).

The u-U formulations tracks the movements of both the soil skeleton and the pore fluid. This formulation is complete in the sense of basic variables, but might still experience numerical problems (volumetric locking) if the difference in volumetric compressibility of the pore fluid and the solid skeleton is large.

The u-p-U formulation resolves the issues of volumetric locking by including the displacements of both the solid skeleton and the pore fluid, and the pore fluid pressure as well. This formulation uses additional dependent unknown field of pore fluid pressures to stabilize the solution of the coupled system. The pore fluid pressures are connected to (dependent on) displacements of pore fluid. With known (given) volumetric compressibility of the pore fluid, pore fluid pressure can be calculated. Despite it's power, the u-p-U formulation has rarely been implemented into finite element code, and has never (at least to our knowledge) been used to analyze pile – liquefied soil interaction. This can be attributed in part to a sophistication of implementation that is required, and to a sizable increase in computational cost for u-p-U elements.

505.3 Liquefaction of Level and Sloping Grounds

Material presented here is from Cheng et al. (2007); Jeremić et al. (2008).

Liquefaction of level and sloping grounds represents a very common behavior during earthquakes. Of interest is to estimate settlement for level ground, and horizontal movements for sloping grounds. In next few sections, presented are results for a 1D, vertical (level ground) and sloping ground cases for dense and loose sand behavior during seismic shaking.

505.3.1 Model Description

Vertical soil column consists of a multiple-elements subjected to an earthquake shaking. The soil is assumed to be Toyoura sand and the calibrated parameters are from Dafalias and Manzari (2004a), and are given in the Table (505.1).

Table 505.1: Material parameters of Dafalias-Manzari model.

material parameter		value	material parameter		value
Elasticity	G_0	125 kPa	Plastic modulus	h_0	7.05
	v	0.05		c_h	0.968
Critical sate	M	1.25		n_b	1.1
	c	0.712	Dilatancy	A_0	0.704
	λ_c	0.019		n_d	3.5
	ξ	0.7	Fabric-dilatancy	z_{max}	4.0
	e_r	0.934		c_z	600.0
Yield surface	m	0.01			

The other parameters, related to the boundary value problem are given in table (505.2).

For tracking convenience, the mesh elements are labeled from E01 (bottom) to E10 (surface) and nodes at each layers are labeled from A (bottom) to K (surface).

A static application of gravity analysis is performed before seismic excitation. The resulting fluid hydrostatic pressures and soil stress states along the soil column serve as initial conditions for the subsequent dynamic analysis.

It should be noted that the self weight loading is performed on an initially zero stress (unloaded) soil column and that the material model and numerical integration algorithms are powerful enough to follow through this early loading with proper evolution. The boundary conditions are such that the soil and water displacement degree of freedom (DOF) at the bottom surface are fixed, while the pore pressure

Table 505.2: Additional parameters used in boundary value problem simulations (other than material parameters from the Table (505.1)).

Parameter	Symbol	Value
Solid density	ρ_s	$2700\ kg/m^3$
Fluid density	ρ_f	$1000\ kg/m^3$
Solid particle bulk modulus	K_s	$3.6 \times 10^7\ kN/m^2$
Fluid bulk modulus	K_f	$2.2 \times 10^6\ kN/m^2$
permeability	k	$5.0 \times 10^{-4}\ m/s$
HHT parameter	α	-0.2

DOFs are free; the soil and water displacement DOFs at the upper surface are free upwards to simulate the upward drainage. The pore pressure DOFs are fixed at surface thus setting pore pressure to zero. On the sides, soil skeleton and water are prevented from moving in horizontal directions while vertical movement of both is free. It is emphasized that those displacements (of soil skeleton and pore fluid) are different. In order to simulate the 1D behavior, all DOFs at the same depth level are connected in a master–slave fashion. Modeling of sloping ground is done by creating a constant horizontal load, sine of inclination angle, multiplied by the self weight of soil column, to mimic sloping ground. In addition to that, for a sloping ground, there should be a constant flow (slow) downhill, however this is neglected in our modeling. The permeability is assumed to be isotropic $k = 5.0{\times}10^{-4}$ m/s. The input acceleration time history (Figure (505.1)) is taken from the recorded horizontal acceleration of Model No.1 of VELACS project Arulanandan and Scott (1993) by Rensselaer Polytechnic Institute, `http://geoinfo.usc.edu/gees/velacs/`. The magnitude of the motion is close to 0.2 g, while main shaking lasts for about 12 seconds (from 1 s to 13 s). For the sloping ground model a slope of % 3 was considered.

It should be emphasized that the soil parameters are related to Toyoura sand, not Nevada sand which is used in VELACS project. The purpose of presented simulation is to show the predictive performance using verified and validated formulation, algorithms, implementation and models.

505.3.2 Behavior of Saturated Level Ground

Figure (505.2) describes the response of the sample with loose sand $e_0 = 0.85$. This figure shows the typical mechanism of cyclic decrease in effective vertical stress due to pore pressure build up as expected for the looser than critical granular material. The lower layers show only the reduction of effective vertical stress from the beginning. Once the effective vertical (and therefore confining) stress

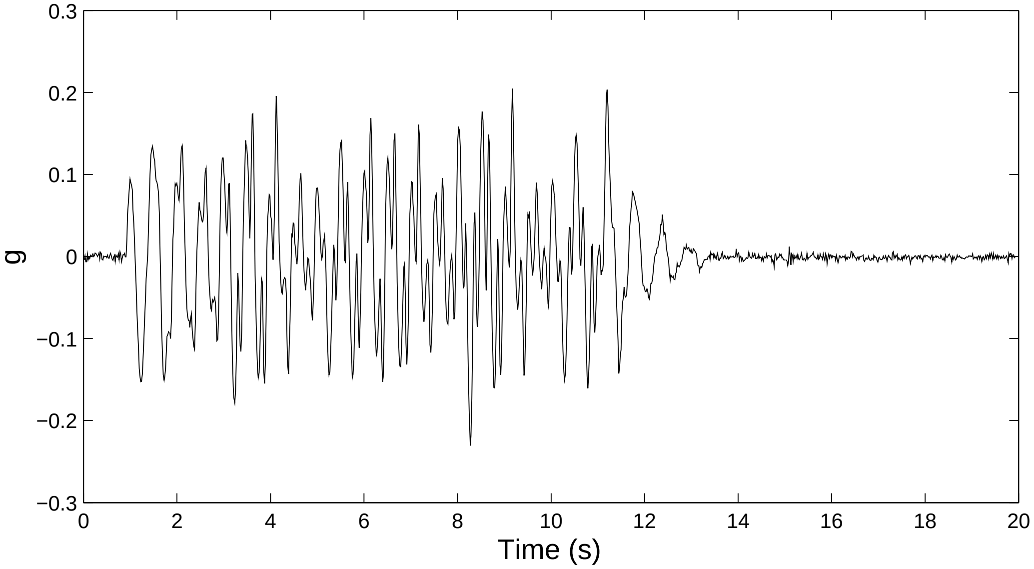


Figure 505.1: Input earthquake ground motion for the soil column.

approaches the smaller values, signs of the so-called butterfly shape can be observes in the stress path. Similar observation can made in the upper layers which have the smaller confining pressure comparing to the lower layers from the very beginning due to lower surcharge. The upper layers have lower confining pressure (lower surcharge) at the beginning of the shaking, hence less contractive response is expected in these layer; however, soon after the initiation of the shaking these top layers start showing the liquefaction state and that type of response continues even after the end of the shaking. The top section of the model has remained liquefied well past the end of shaking. This is explained by the large supply of pore fluid from lower layers, for which the dissipation starts earlier. For example, for the lowest layer, the observable drop in excess pore pressure start as soon as the shaking ends, while, the upper layers then receive this dissipated pore fluid from lower layers and do liquefy (or continue being liquefied) well past end of shaking (which happens at approximately 13 seconds). It is very important to note the significance of this incoming pore water flux on the pore water pressure of the top layers. Despite the less contractive response of soil skeleton at the top elements, the transient pore water flux, that enters these elements from the bottom, forces those to a liquefaction state. In other words, the top elements have not liquefied only due to their loose state but also because of the water flow coming from the bottom layers. The maximum horizontal strains can be observed in the middle layers due to liquefaction and prevents upper layers from experiencing larger strains. The displacements of water and soil are presented in the last column. It shows that in all layers the upward displacement of water is larger than the downward displacement of soil. This behavior reflects soil densification during shaking.

Figure (505.3) describes the response of the sample with dense sand ($e_0 = 0.75$). This figure also shows the typical mechanism of cyclic decrease in effective vertical stress. However, in case of this dense sample the decreasing rate of the effective confining pressure is much smaller than what was observed in the loose sample. Signs of the partial butterfly shape in the effective stress path can be observed from early stages of shaking. The butterfly is more evident in the upper layers with the lower confining pressure, i.e. more dilative response. In later stages of the shaking, i.e. when the confining pressure reduces to smaller value the butterfly shape of the stress path gets more pronounced due to having more dilative response in the lower confining pressure based on CSSM concept. In comparing this dense case to the case of shaking the loose sand column, the current case does not show any major sign of liquefaction (when stress ratio $r_u = 1$). This is due to the less contractive (more dilative) response of the sand in this case, which is coming from the the denser state of the sample. Because of having partial segments of dilative response, the whole column of the sand has not loosed its strength to the extent that happened for the case of loose sand and therefore smaller values of horizontal strains has been observed in the results. The absolute values of soil and water vertical displacements are also smaller than the case of loose sand which can be again referred to the less overall contractive response in this case.

Overall, it can be noted that the response in the case of loose sand ($e_0 = 0.85$) is mainly below the dilatancy surface (phase transformation surface) while the denser sand sample with $e_0 = 0.75$ shows partially dilative response referring to the denser than critical state.

505.3.3 Behavior of Saturated Sloping Ground

Figures (505.4) and (505.5) present the result of the numerical simulations for shaking the inclined soil columns (toward right) with loose and dense sand samples, respectively. The inclination of the soil column results in presence of the offset shear stress to the right side. This essentially poses asymmetric horizontal shear stresses (toward the direction of inclination) during cycles of shaking. On one hand, this offset shear stress makes the sample more dilative in the parts of shaking toward the right side (think about the state distance from the phase transformation line or dilatancy line in the $p-q$ space). As a result asymmetric butterfly loops will be induced causing the soil to regain its stiffness and strength (p) in the dilative parts of the corresponding cycles, therefore only instantaneous spikes of $r_u = 1$ can be observed in case of the sloped columns of soil. There is also a permanent liquefaction in terms of having stationary portions of $r_u = 1$ in this case. On the other hand, the offset shear stress results generation of more horizontal strains in the portions of loading which are directed toward the right side than those which are directed back toward the left side. As a result the horizontal shear strains will accumulative toward the right side and create larger permanent horizontal displacement comparing

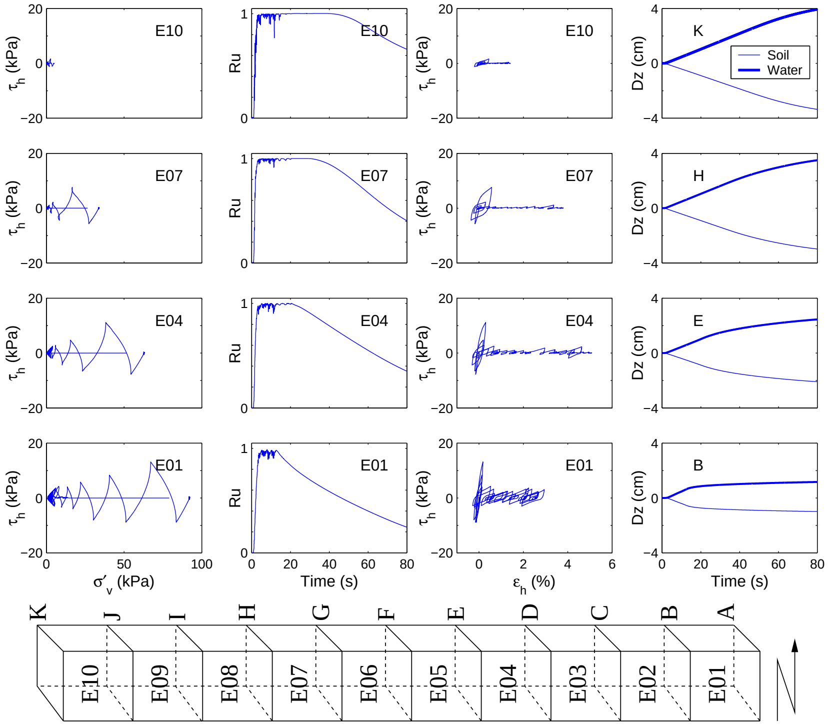


Figure 505.2: Seismic results for (loose sand) soil column in level ground ($e_0 = 0.85$).

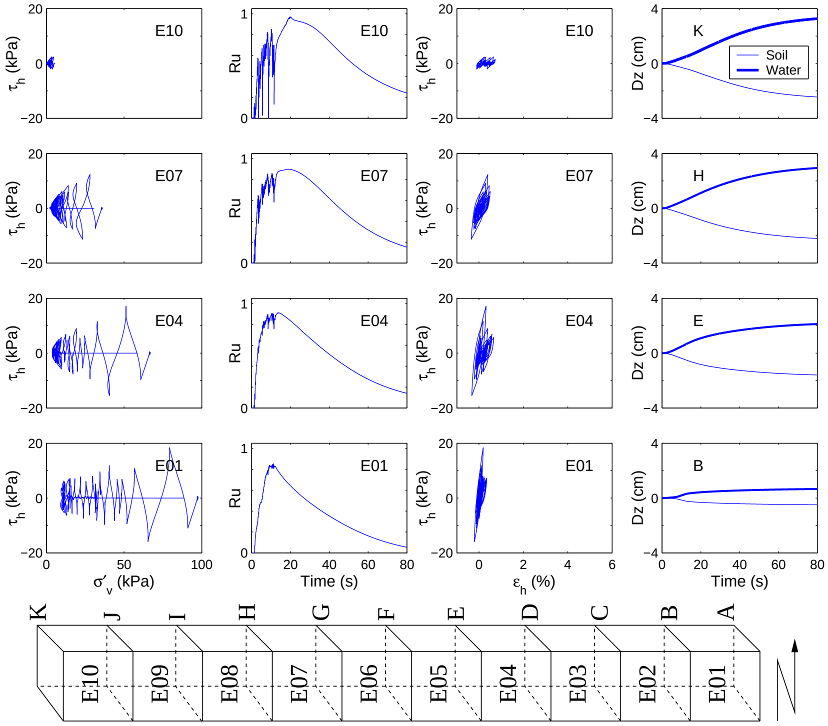


Figure 505.3: Seismic results for (dense sand) soil column in level ground ($e_0 = 0.75$).

to the case of level ground soil column. Since the overall dilative response of the dense sample, i.e. Figure (505.5), is larger than that of the loose sample, i.e. Figure (505.4), the dense sample shows stiffer response and therefore less accumulative horizontal shear strains than the loose sample. The difference in predicted horizontal displacements is almost three times, that is, for the dense sample the final, maximum horizontal displacement is approx. 0.5 m, while for the loose sand sample, it almost 1.5 m,

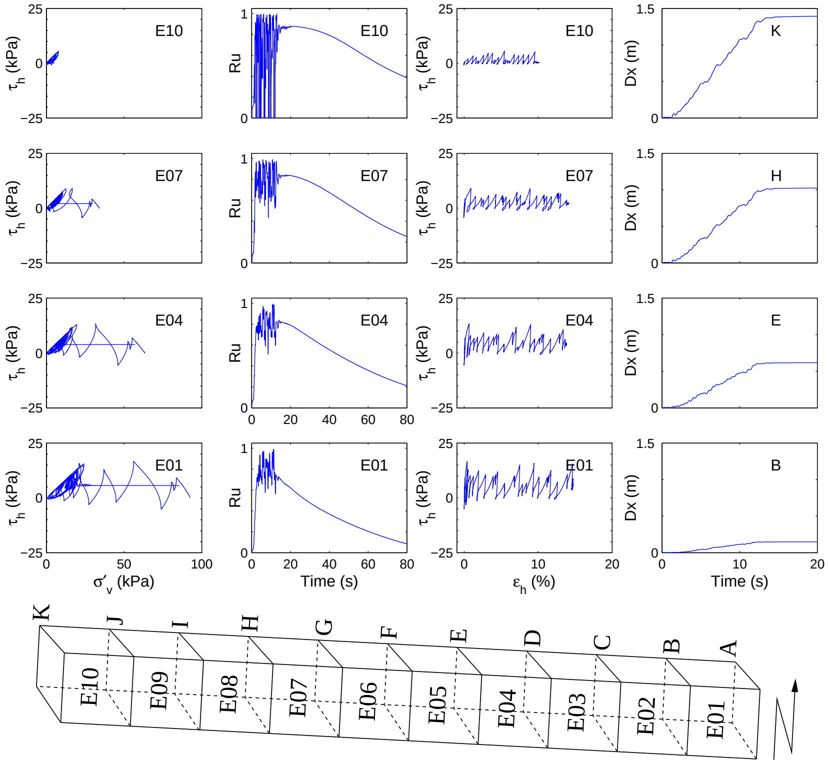


Figure 505.4: Seismic results for (loose sand) soil column in sloping ground ($e_0 = 0.85$).

505.4 Pile in Liquefied Ground, Staged Simulation Model Development

Material presented here is from Cheng and Jeremić (2009a,b) Model development for a pile in the liquefiable soil follows physics (mechanics) of the problem as close as possible. Numerical simulation of such problems in geomechanics is usually based on stages of loading and increments within those stages.

All load stages are applied to a series of finite element models, all of which share features of an initial soil model. This initial soil model consists of a soil block with dimension of $12 \times 12 \times 15$ m (length × width × depth). Due to the symmetry of the model, only half of the block is modeled. Symmetry assumptions is based on assumption that all the loads, dynamic shaking and other influences are symmetric with

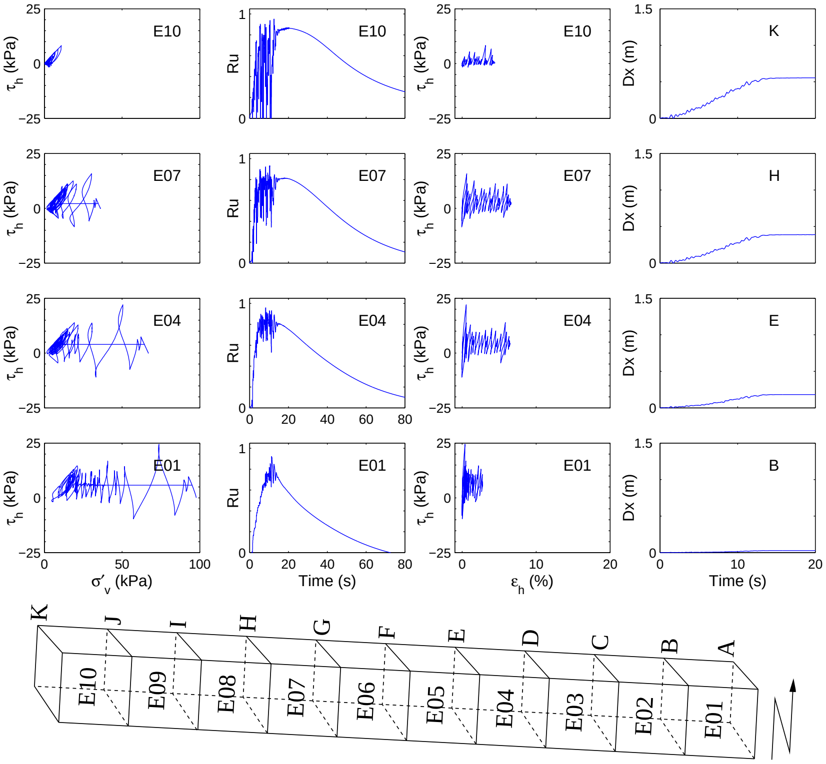


Figure 505.5: Seismic results for (dense sand) soil column in sloping ground ($e_0 = 0.75$).

respect to the plane of symmetry. This specialization to symmetric model reduces model generality (for example this use of symmetry will preclude analysis of dynamic shaking perpendicular to sloping ground dip). However, as our goal is to present a methodology of analyzing behavior of piles in liquefying ground, this potential drawback is not deemed significant in this study. Finite element mesh for the model is presented in Figure (505.6). The initial mesh consists of 160 eight node u-p-U elements.

Each node of the mesh has 7 degrees of freedom, three for soil skeleton displacements (u_i), one for pore water pressure (p), and three for pore water displacement (U_i). While it can be argued that the mesh is somewhat coarse, it is well refined around the pile, yet to be installed, in place of gray region in the middle.

A single set of parameters is used with the Dafalias-Manzari material model. Soil is modeled as Toyoura sand and material parameters (summarized in Table 505.4) are calibrated using tests by Verdugo and Ishihara (1996), while initial void ration was set to $e_0 = 0.80$. It is very important to emphasize that the state of stress and internal variables from initial state (zero for stress and given value for void ratio and fabric) will evolve through all stages of loading by proper modeling and algorithms, by using single set of material parameters. Table 505.4 presents additional parameters, other than material parameters

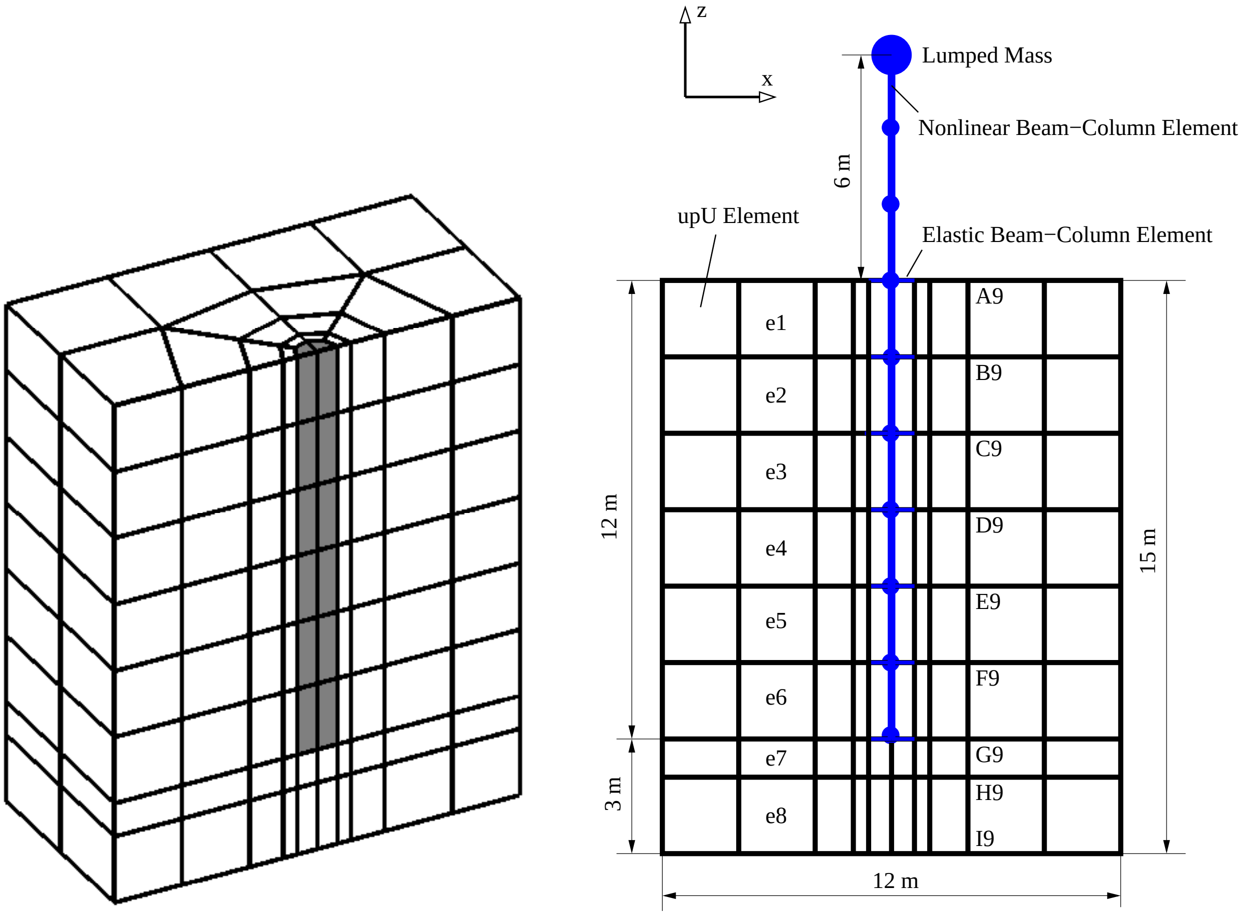


Figure 505.6: Left: Three dimensional finite element mesh featuring initial soil setup, where all the soil elements are present. The gray region of elements is excavated (numerically) and replaced by a pile during later stages of loading; Right: Side view of the pile-soil model with some element and node annotation, used to visualize results.

presented in Table 505.4, used for numerical simulations.

505.4.1 First Loading Stage: Self Weight

The initial stage of loading is represented by the application of self weight on soil, including both the soil skeleton and the pore water. Initial state in soil before application of self weight is of a zero stress and strain while void ratio and fabric are given initial values. The state of stress/strain, void ratio and fabric will evolve upon application of self weight. At the end of self weight loading stage, soil is under appropriate state of stress (K_0 stress), the void ratio corresponds to the void ratio after self weight (redistributed such that soil is denser at lower layers), while soil fabric has evolved with respect to stress

Table 505.3: Material parameters used for Dafalias-Manzari elastic–plastic model.

Material Parameter		Value	Material Parameter		Value
Elasticity	G_0	125 kPa	Plastic modulus	h_0	7.05
	v	0.05		c_h	0.968
Critical sate	M	1.25		n_b	1.1
	c	0.8	Dilatancy	A_0	0.704
	λ_c	0.019		n_d	3.5
	ξ	0.7	Fabric-dilatancy	z_{max}	4.0
	e_r	0.934		c_z	600.0
Yield surface	m	0.02			

Table 505.4: Additional parameters used in FEM simulations.

Parameter		Value
Solid density	ρ_s	$2800\ kg/m^3$
Fluid density	ρ_f	$1000\ kg/m^3$
Solid particle bulk modulus	K_s	$1.0 \times 10^{12}\ kN/m^2$
Pore fluid bulk modulus	K_f	$2.2 \times 10^{6}\ kN/m^2$
permeability	k	$1.0 \times 10^{-4}\ m/s$
Gravity	g	10 m/s^2

induced anisotropy. All of these changes are modeled using Dafalias–Manzari material model and using constitutive and finite element level integration algorithms developed within UC Davis Computational Geomechanics group in recent years.

Boundary conditions (BC) for self weight stage of loading are set in the following way:

- Soil skeleton displacements (u_i), are fixed in all three directions at the bottom of the model. At the side planes, nodes move only vertically to mimic self-weight effect. All other nodes are free to move in any direction.

- Pore water pressures (p), are free to develop at the bottom plane and at all levels of the models except at the top level at soil surface where they are fixed (set to zero, replicating drained condition),

- Pore water displacements (U_i), are fixed in all three directions at the bottom, are free to move only vertically at four sides of the model and are free to move in any direction at all other nodes.

These boundary condition are consistent with initial self-weighting deformation condition for soil and pore water at the site.

For the case of sloping ground, an additional load sub-stage is applied after self weight loading, in order to mimic self weight of inclined (sloping) ground. This is effectively achieved by applying a resultant of total self weight of the soil skeleton times the sine of the inclination angle at uphill side of the model. This load is applied only to the solid skeleton DOFs, and not on the water DOFs. Physically it would be correct to consider the sloping ground effects on the pore water as well. This will create a constant flow field of the water downstream, which, while physically accurate, is small enough that it does not have any real effect on modeling and simulations performed here.

505.4.2 Second Loading Stage: Pile–Column Installation

After the first loading stage, comprising self weight applications (for level or sloping ground, as discussed above), second loading stage includes installation (construction) of the pile–column. Modeling changes performed during loading stage included:

- Excavation of soil occupying space where the pile will be installed. This was done by removing elements, nodes and loads on elements shown in gray in Figure (505.6).
- These elements were replaced by very soft set of elements with small stiffness, low permeability. This was done in order to prevent water from rushing into the newly opened hole in the ground after original soil elements (used in the first loading stage) are removed.
- Installation of a pile in the ground and a superstructure (column) above the ground. Nonlinear bean–column elements were used for both pile and column together with addition of appropriate nodal masses at each beam-column node, and with the addition of a larger mass at the top representing lumped mass of a bridge superstructure. Pile beam-column elements were connected with soil skeleton part of soil elements using a specially devised technique.

As mentioned earlier, the volume that would be physically occupied by the pile in the pile hole, is "excavated" during this loading stage. Beam–column elements, representing piles, are then placed in the middle of this opening. Pile (beam–column) elements are then connected to the surrounding soil elements by means of stiff elastic beam–column elements. These "connection" beam–column elements extend from each pile node to surrounding nodes of soil elements. The connectivity of nodes to soil skeleton nodes is done only for three beam–column translational DOFs, while the three rotational DOFs from the beam–column element are left unconnected. These three DOFs from the beam–column side are connected to first three DOFs of the u-p-U soil elements, representing displacements of the soil skeleton

(u_i). Water displacements (U_i) and pore water pressures (p) are not connected in any way. Rather, these two sets of DOFs representing pore water behave in a physical manner (cannot enter newly created hole around pile beam–column elements) because of the addition of a soft, but very impermeable set of u-p-U elements, replacing excavated soil elements. By using this method, both solid phase (pile with soil skeleton) and the water phase (pore water within the soil) are appropriately modeled. Figure (505.7) shows in some detail schematics of coupling between the pile and soil skeleton part finite elements.

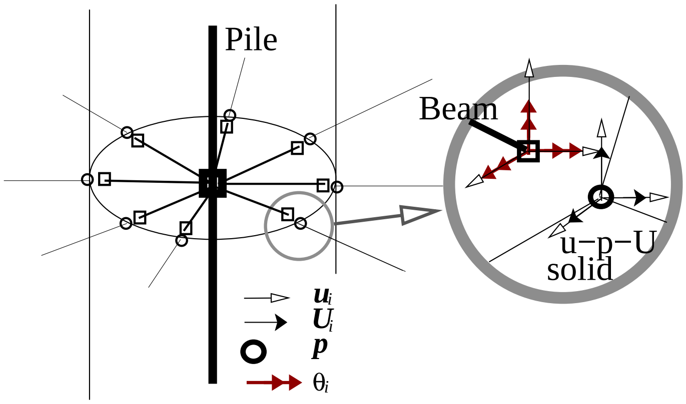


Figure 505.7: Schematic description of coupling of displacement DOFs (u_i) of beam-column element (pile) with displacement DOFs (u_i) of u-p-U elements (soil).

Nonlinear force based beam–column elements (Spacone et al., 1996a,b) were used for modeling the pile–column. Pile was assumed to be made of aluminum. This was done in order to be able to validate simulations with centrifuge experiments (when they become available). Presented models were all done in prototype scale, while for (possible future) validation, select results will be carefully scaled and compared with appropriate centrifuge modeling. Pile and the column were assumed to have a diameter of $d = 1.0$ m, with Young's modulus of $E = 68.5$ GPa, yield strength $f_y = 255$ kPa, and the density $\rho = 2.7\ \text{kg/m}^3$. Wall thickness of prototype pile–column is $t = 0.05$ m. Lumped mass of pile and column was distributed along the beam–column nodes, while an additional mass was added on top $(m = 1200\ \text{kg})$ that represents (small) part of the superstructure mass. This particular mass $(m = 1200\ \text{kg})$ comes from a standard (scaled up in our case) centrifuge model for pile–column–mass used at UCD.

Figure (505.6) (right side) shows side view of the column-pile-soil model after second stage of loading.

505.4.3 Third Loading Stage: Seismic Shaking

After the application of self weight on the uniform soil profile, excavation and construction of the single pile with column and super structure mass on top and application of their self weight, the model is at the appropriate initial state for further application of loading. In this case, this additional loading comprises seismic shaking. For this stage, fixed horizontal DOFs used on the side planes during the first stage are removed (set free).

The input acceleration time history, shown in Figure (505.8) was taken from the recorded horizontal acceleration of Model No.1 of VELACS project Arulanandan and Scott (1993) by Rensselaer Polytechnic Institute, `http::/geoinfo.usc.edu/gees/velacs/`. The magnitude of the motion is close to 0.2 g, while main shaking lasts for about 12 seconds (from 1 s to 13 s). Although the input earthquake motions lasts until approx. 13 seconds, simulations are continued until 120 seconds so that both liquefaction (dynamic) and pore water dissipation (slow transient) can be appropriately simulated during and after earthquake shaking (Jeremić et al., 2008).

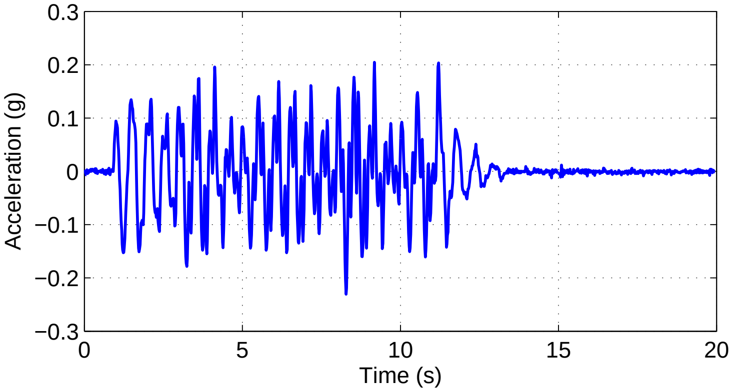
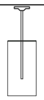
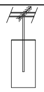
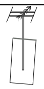


Figure 505.8: Input earthquake ground motions.

505.4.4 Free Field, Lateral and Longitudinal Models

Six models were developed during the course of this study. First three models (model numbers I, II and III) were for level ground, while last three models (model numbers IV, V, and VI) were for sloping ground. First in each series of models (model I for level ground and model IV for sloping ground) were left without the second loading stage, without a pile–column system. Other four models (numbers II, III, V and VI) were analyzed for all three loading stages. Second in each series of models (models number II and V) had all displacements and rotations of pile–column top (where additional mass representing superstructure was placed) left free, without restraints. Thus, these two models represent lateral behavior of a bridge. Third in each series of models (model numbers III and VI) had rotations in y directions fixed at the pile–column top, thus representing longitudinal behavior of a bridge. Modeling longitudinal behavior of a bridge by restraining rotations perpendicular to the bridge superstructure is appropriate if the stiffness of a bridge superstructure is large enough, which in this case it was, as it was assumed to be a post–tensioned concrete box girder, so that realistically, the top of a column does not rotate (much) during application of loads. Table 505.5 summarizes models described above.

Table 505.5: Cases descriptions.

Case	Model sketch	Descriptions
I		horizontal ground, no pile
II		horizontal ground, single pile, free column head
III		horizontal ground, single pile, no rotation at column head
IV		sloping ground, no pile
V		sloping ground, single pile, free column head
VI		sloping ground, single pile, no rotation at column head

505.5 Simulation Results

505.5.1 Pore Fluid Migration

Figures (505.9) through (505.11) show the Ru time history for up to 30 seconds, for elements (at one of Gauss point) e1, e3, e5 and e7 (refer to right side of Figure (505.6)). It is important to note that R_u is defined as the ratio of the difference of initial mean and current mean effective stresses over the initial mean effective stress:

$$R_u = \frac{p'_{initial} - p'_{current}}{p'_{initial}}$$

where mean effective stress is defined as $p' = \sigma'_{kk}/3$. This is different from traditional definition for R_u, that uses ratio of excess pore pressure over the initial mean effective stress ($p'_{initial}$). However, these two definitions are essentially equivalent, as soil is in the state of liquefaction for $R_u = 1$ (so that $p'_{current} = 0$), while there is no excess pore pressure for $R_u = 0$ (so that $p'_{initial} = p'_{current}$). However, the former definition is advocated here as it avoids the interpolation of pore pressure or extrapolation of the stresses (as the latter definition requires), since for the u-p-U element, stresses are available at Gauss points while pore pressures are available element nodes. In particular, Figure 505.9 shows R_u time histories for four points for models I (level ground without pile) and model IV (sloping ground without pile). It is noted that differences are fairly small. It is interesting to observe that lower layers do not liquefy as supply of pore fluid for initial void ratio of $e_0 = 0.8$ is too small, and the pore fluid dissipation upward seems to be to rapid. On the other hand, the upper soil layers do reach close to or liquefaction state ($R_u = 1$). This is primarily due to the propagation of pore fluid pressure/volume from lower layers upward (pumping effect) and, in addition to that, to a local excess pore fluid production. These results can also be contrasted with those of Jeremić et al. (2008), where similar pumping scenario has been observed. The main difference between soil used by Jeremić et al. (2008) and here is in the coefficient of permeability. Namely, here $k = 1.0 \times 10^{-4}$ m/s was used (Čubrinovski et al., 2008; Uzuoka et al., 2008) while Jeremić et al. (2008) used $k = 5.0 \times 10^{-4}$ m/s. It is important to note that other values of permeability for Toyoura sand have also been reported (Sakemi et al., 1995), but current value was

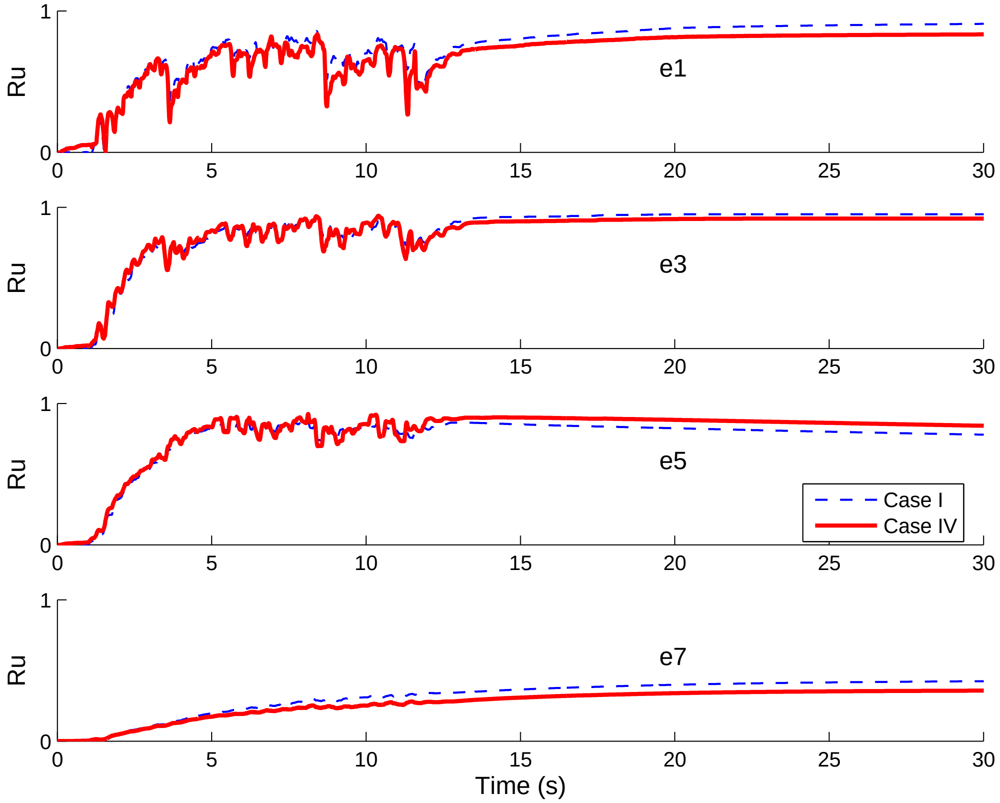


Figure 505.9: R_u times histories for elements e1 (top element), e3, e5, and e7 (bottom element) Gauss point) for Cases I (level ground, no pile) and IV (sloping ground, no pile).

chosen based on Čubrinovski (2007 –).

In addition to that, similar to Jeremić et al. (2008), sloping ground case shows larger R_u spikes, since there is static shear force (stress) that is always present from gravity load on a slope. This static gravity on a slope will result in an asymmetric horizontal shear stresses in the down–slope direction during cycles of shaking. This asymmetric shear stress induces a more dilative response for down–slope shaking which will help soil regain its stiffness in the dilative parts of the loading cycles. This observation can be used to explain smaller R_u spikes for the sloping ground case. Of course, this asymmetry in loading will result in larger accumulation of down–slope deformation.

While R_u ratios for level and sloping ground cases are fairly similar along the depth of the model, the response changes when the pile is present. Figure (505.10) shows R_u responses at four different points (along the depth) approximately midway between the pile and the model boundary, in the plane of shaking (see location of those elements in Figure (505.6) on page 2616). In comparison to behavior without the pile (Figure (505.9)), it is immediately obvious that addition of a pile with a mass on top reduces R_u during shaking for the top element (e1). This is to be expected as presence of a pile–column–mass (PCM) system changes the dynamics of the top layers of soil significantly enough to reduce total amount of shear. This is particularly true for the top layers of soil as effects of column–mass tend to create compressive and extensive movements (compression when the PCM system moves toward soil and extension, and possibly even tension, when PCM system moves away from soil). However, this extension, or possible tension, is not directly observable in presented plots since array of elements where we follow R_u (e1, e3, e5, e7) is some distance away from the pile–soil interface. Middle layers (e3 and e5), on the other hand, display similar response to that of Cases I and IV, as shown in Figure (505.9). It is noted that in a case with of sloping ground with pile, the R_u measurements are always larger that those for level ground (this is also observed for Cases III and VI, as shown in Figure (505.11)). This is expected as presence of a pile in loose sand, and particularly the dynamic movement of a pile during seismic shaking, create an additional shearing deformation field (in the soil adjacent to the pile) that provides for additional (faster) compression of soil skeleton and thus creates additional volume of pore fluid, that is then distributed to adjacent soil (adjacent to the pile).

Particularly interesting are R_u results for soil element e7, which is located below pile tip level (see Figure (505.6)). Observed R_u for Case V in element e7 is significantly larger than for the same element for Case II. Similarly, simulated R_u is larger than what was observed in cases without a pile (see bottom of Figure (505.9)). This increase in R_u for Case V (sloping ground with pile) is explained by noting that the pile "reinforces" upper soil layers and thus prevents excess shear deformation in the upper 12.0 m of soil (above pile tip). The reduction of deformation in upper layers of soil (top 12.0 meters) results in transfer of excessive soil deformation to soil layers below pile tip (where element e7 is located). This,

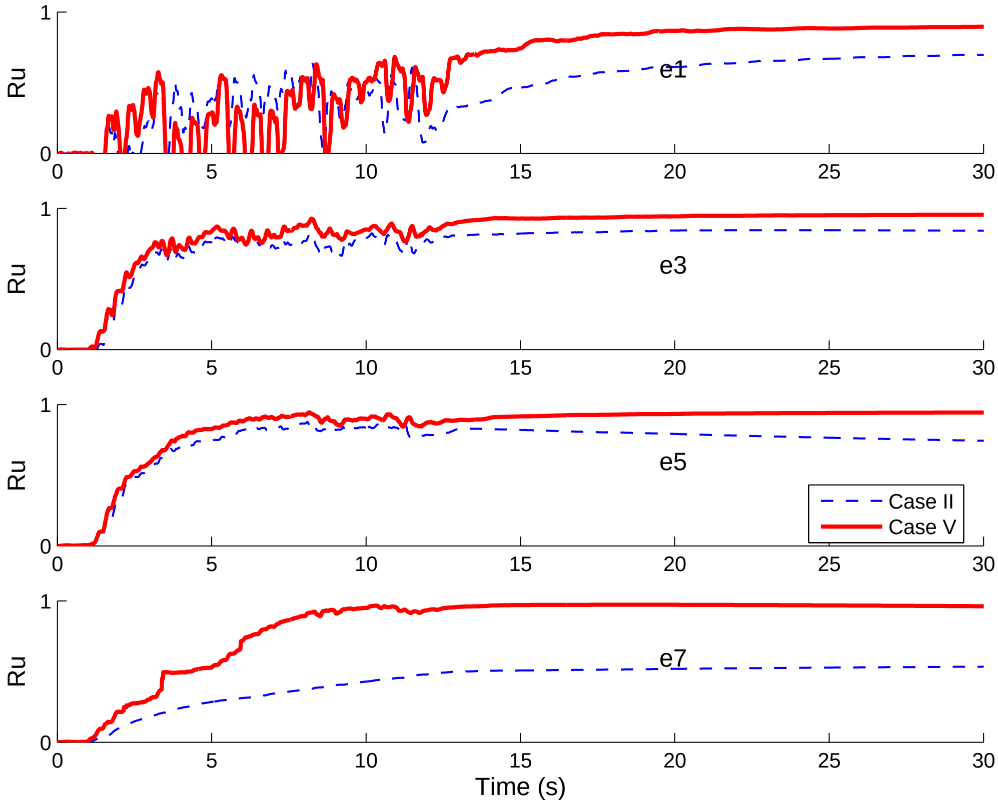


Figure 505.10: R_u times histories for elements e1, e3, e5, and e7 (upper Gauss point) for Cases II (level ground, with pile–column, free column head) and V (sloping ground, with pile–column, free column head).

in turn, results in a much larger and faster shearing of those lower loose soil layers. This significantly larger shearing results in a much higher R_u. Deformed shape, shown in Figure (505.12) for Case V, reinforces this explanation, showing much large shearing deformation in lower soil layers, below pile tip. Same observation can be made for Case VI, shown in Figure (505.12).

Observation similar to the above, for Cases II and V can be made for Cases III and VI, results for which are shown in Figure (505.11). One noticeable difference in R_u results between cases with free column head (Cases II and V) and cases with fixed rotation column head (Cases III and VI) is in significantly larger (and faster) development of R_u close to soil surface for a stiffer, no rotation column cases (Cases III and VI). This much larger R_u observed in a “stiffer” PCM system setup, is due to larger shearing deformation that develops in soils adjacent to the pile during shaking. The stiffer PCM system can displace less (because of additional no rotation condition on column top) while the soil beneath is undergoing shaking (same demand in all cases), thus resulting in larger relative shearing of soil, which then results in larger and faster pore pressure development close to the soil surface, where the column

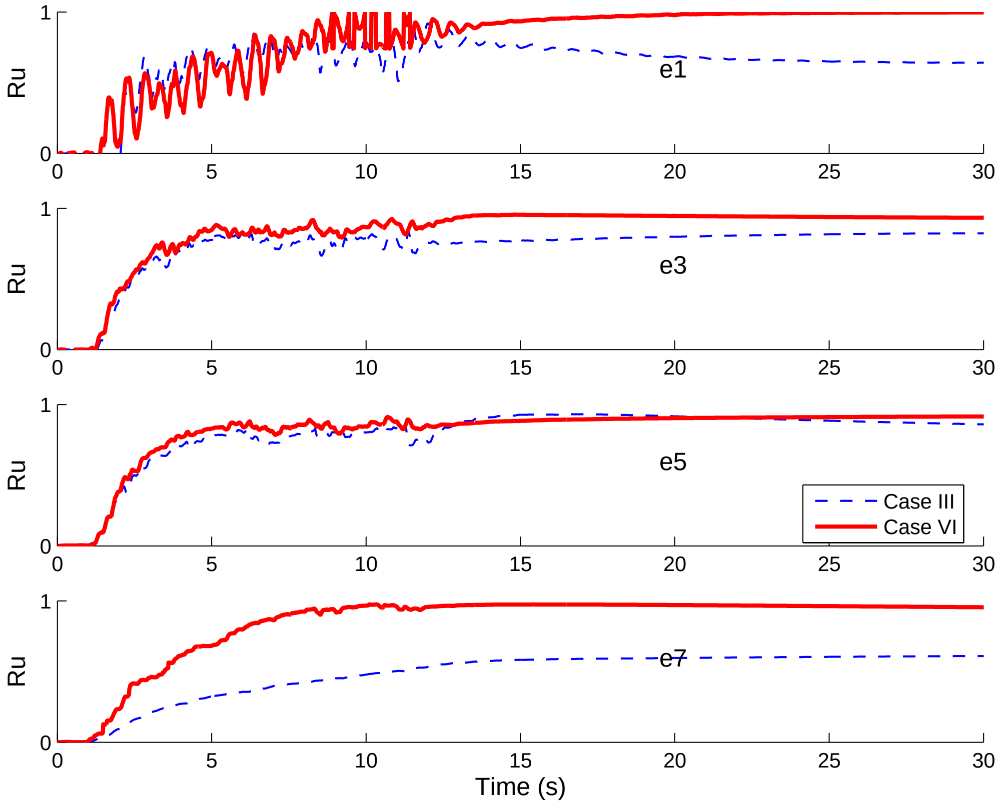


Figure 505.11: R_u times histories for elements e1, e3, e5, and e7 (upper Gauss point) for Cases III (level ground, with pile, no rotation of pile head) and VI (sloping ground, with pile-column, no rotation of column head).

no rotation effect is most pronounced.

505.5.2 Soil Skeleton Deformation

A number of deformation modes is observed for both level and sloping ground cases, with or without PCM system. Figure (505.12) shows deformation patterns and excess pore pressures in symmetry plane for all six cases over a period of eighty seconds. A number of observation can be made on both deformation patterns, excess pore fluid patterns and their close coupling.

Level Ground without Pile (Case I). Excess pore pressures and deformations in symmetry plane for level ground without a pile are shown in Figure (505.12) (I). At the very beginning (at $t = 2$ s) there is initial development of excess pore fluid pressure in the middle soil layers. This expected, as the self weight loading stage has densified lower soil layers enough so that their response is not initially contractive enough to produce excess pore pressure. Top soil layers, on the other hand, have a drainage boundary

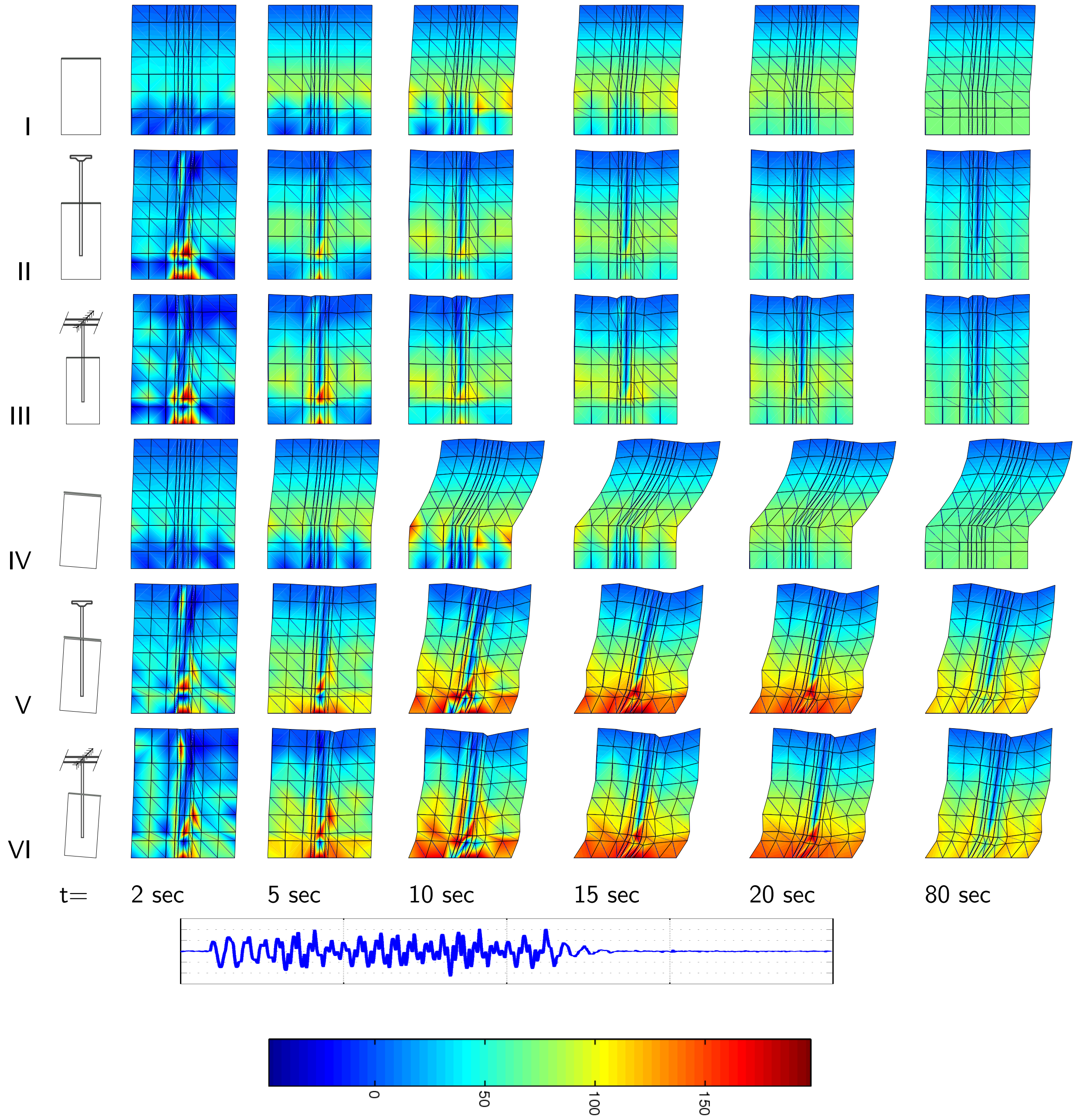


Figure 505.12: Time sequence of deformed shapes and excess pore pressure in symmetry plane of a soil system. Deformation is exaggerated 15 times; Color scale for excess pore pressures (above) is in kN/m^2. Graph of ground motions used (also shown in Figure (505.8)) is placed below appropriate time snapshots and is matching for $t = 2, 5, 10, 15, 20$ seconds while at $t = 80$ seconds there is no seismic shaking.

(top surface) too close to develop any significant excess pore pressures. As seismic shaking progresses (for $t = 5, 10$ s), the excess pore pressure increases, and starts developing in lower soil layers as well. It should be noted that a small non-uniformity in results is present. For example, zones of variable, nonuniform excess pore pressures on the lower mid and right side for Case I at $t = 10$ s develop. Nonuniform mesh (many small, long elements in the middle, large elements outside this middle zone) may introduce small numerical errors in results which can be observed by slightly nonuniform results at $t = 10$ s and $t = 15$ s. It should be noted that results for excess pore pressure shown for first 13 seconds (during shaking) in Figure (505.12) (I) are transient in nature, that is, seismic waves are traveling throughout the domain (model) during shaking (first 13 seconds) and slight oscillations in vertical stresses are possible. This oscillations will contribute to the (small) non-uniformity of excess pore pressure results. After the shaking (after 15 seconds) resulting excess pore pressure field is quite uniform.

Level Ground with Pile (Cases II and III). Excess pore pressures and deformations in symmetry plane for models with PCM system and with two different boundary conditions at top of column (see model description in section 505.4.4) in level ground are shown in Figures (505.12) (II and III). One of the interesting observations is significant shearing and excess pore pressure generation adjacent to the pile tip. The reason for this is that pile is too short, that is, pile tip has significant horizontal displacements during shaking. Those pile tip displacements shear the soil, resulting in excess pore pressure generation. As soon as there is time for dissipation, this localized excess pore pressure dissipates to adjacent soil, and then, after shaking has ceased (at $t = 13$ s and later), it slowly dissipates upward. Addition of pile into the model (construction), with a highly impermeable elements (that mimic permeability of concrete) is apparent as there is a low excess pore pressure region in the middle of model, where pile is located.

Sloping Ground without Pile (Case IV). Excess pore pressures and deformation in symmetry plane for sloping ground without pile is shown in Figures (505.12) (IV). It is noted that initially the excess pore pressure starts developing in middle soil layers,similar to the Case I above. Bottom layers start developing excess pore pressure only after significant shear deformation occurs (at $t = 10$ s) at approximately $2/3$ of the model depth. Lower layers have densified enough during self weight stage of loading that initial shaking is not strong enough to create excess pore water pressure, rather, those layers are fed by the excess pore pressure from above. Lower soil layers also do not develop much deformation, while middle and upper layers together develop excessive horizontal deformation.

Sloping Ground with Pile (Cases V and VI). Excess pore pressures and deformation in symmetry plane for sloping ground with PCM system are shown in Figures (505.12) (V and VI). Similar to the above cases

(II and III), pile is too short and there is again excessive shearing of soil at the pile tip, suggesting large movement of that pile tip. In addition to that, pile introduces significant stiffness to upper 12 meters of soil (along the length of pile) and helps reduce deformation of those upper soil layers. Down-slope gravity load is thus transferred to lower soil layers (below pile tip) which exhibit most of the deformation. It should be noted that soil in middle and upper layers (adjacent to pile) does deform, just not as much as the soil below pile tip. The predominant mode of deformation of middle soil layers is shearing in horizontal plane, around the pile. Deformation in horizontal plane is not significant as the pile is short in this examples (as mentioned above) and does not have enough horizontal support at the bottom. The deformation pattern of a soil – pile system is such that pile experiences significant rotation, and deforms with the soil that moves down-slope. If the pile was longer, and if it had significant horizontal support at the bottom, the middle and upper soil layers would have showed more significant flow around the pile in horizontal planes.

Upper layers undergo significant settlement, as seen in Figure (505.13). This settlement is mainly caused by the above mentioned rotation of pile–soil system, where soil in general settles (compacts) but also undergoes differential settlement, between left (up–slope from pile) and right (down–slope from pile) side of the model. As significant shearing with excess pore pressure generation develops in lower soil layers, below pile tip, those lower layers contribute to most of down–slope horizontal deformation. In a sense, all the demand from down-slope gravity forces and seismic shaking is now responded to by lower soil layers, which contribute to most of the excess pore pressure generation and consequently, to most of the soil deformation. Soil surface horizontal deformation is thus strongly influenced by significant shearing of the bottom layers and by rotation of the middle and upper soil layers with the pile. It is interesting to note that the largest settlement is observed just down-slope from pile for Cases V and VI.

505.5.3 Pile Response

Figure (505.14) shows bending moment envelops for pile–column–mass (PCM) system for all four cases (II, III, V and VI). It should be noted that bending moment diagrams are plotted on compression side of the beam–column. A number of observations can be made about bending moment envelopes. For cases with free pile head (shaking transverse to the bridge main axes, Cases II and V) the maximum moments are attained in soil, at depths of approximately $0.6D - 1.2D$, where D (= 1.0 m in this case) is the pile diameter. Opposed to that are cases for PCM systems with restricted rotations at the pile top which (Cases III and VI), which, of course feature largest moment at the column top. Maximum bending moments for section of PCM system in soil (pile) in these two cases are now attained at the depth of approximately $1.8D - 2.0D$.

It is noted that bending moment envelopes are mostly symmetric. Slight non–symmetry is introduced

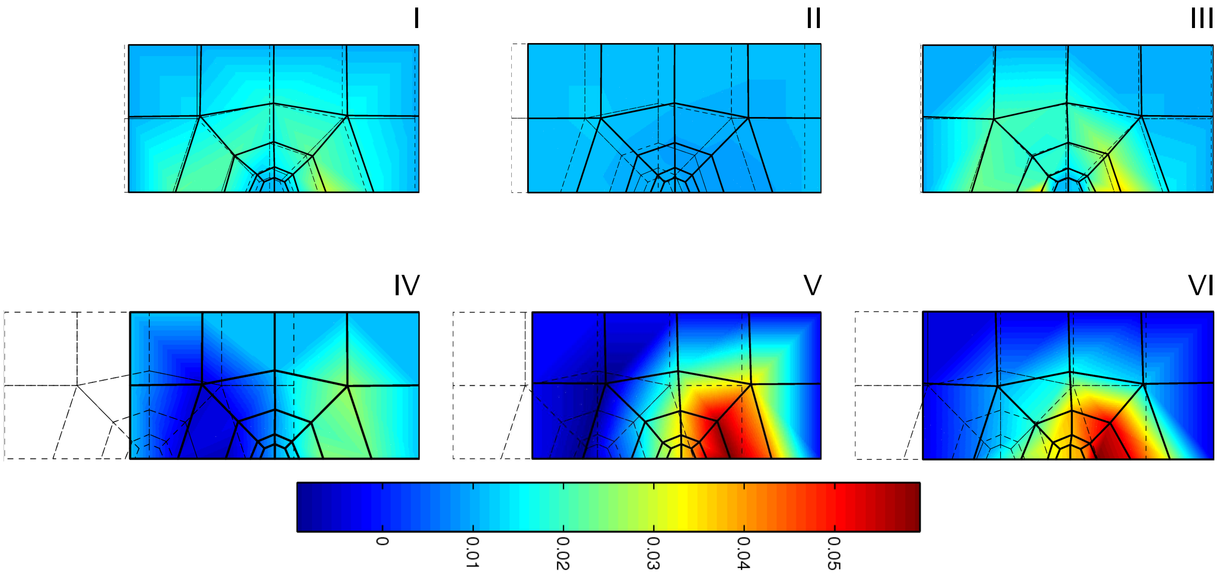


Figure 505.13: Soil surface settlements at 120 s for all six cases. Color scale given in meters

for cases on sloping ground (Case V and VI). It is also noted that moments do exist (are not zero) all the say to the bottom of the pile. Theoretically, moments should be zero at the pile tip, but since physical volume of the pile is considered (see note on that in section 505.4.2 and Figure (505.7)), differential pressure on pile bottom from soil will produce small (non–zero) moments even at the pile tip. More importantly, non–zero moments at the bottom and along the lower part of the pile show that pile is indeed too short, and thus changing curvatures are present along the whole length of the pile.

505.5.4 Pile Pinning Effects

Piles in sloping liquefying ground can also be used to resists movement of soil (all liquefied or liquefied with hard crust on top) down–slope. For models developed in this paper, pile pinning effect can be investigated for Cases IV, V and VI. In particular, deformation of sloping ground without the pile (Case IV) can be compared with either of the cases of piles in sloping ground, Cases V and VI. It is very important to note, again, that models developed here had relatively short pile, and that major soil shearing developed below the pile tip. This apparent shortcoming of a short pile results in reduced pile pinning capacity, thus reducing the down–slope movement by only approximately half, from 0.35 m (Case IV) to 0.22 m (Case V) and to 0.18 m (Case VI) as seen in Figure (505.15). It would have been expected that, had the pile been longer and had it penetrated in deeper, non-liquefiable layers, it would have reduced down–slope movement of the soil to a much larger extent. However, had the pile been longer and had it penetrated non-liquefiable layers, it would have had a much firmer horizontal support

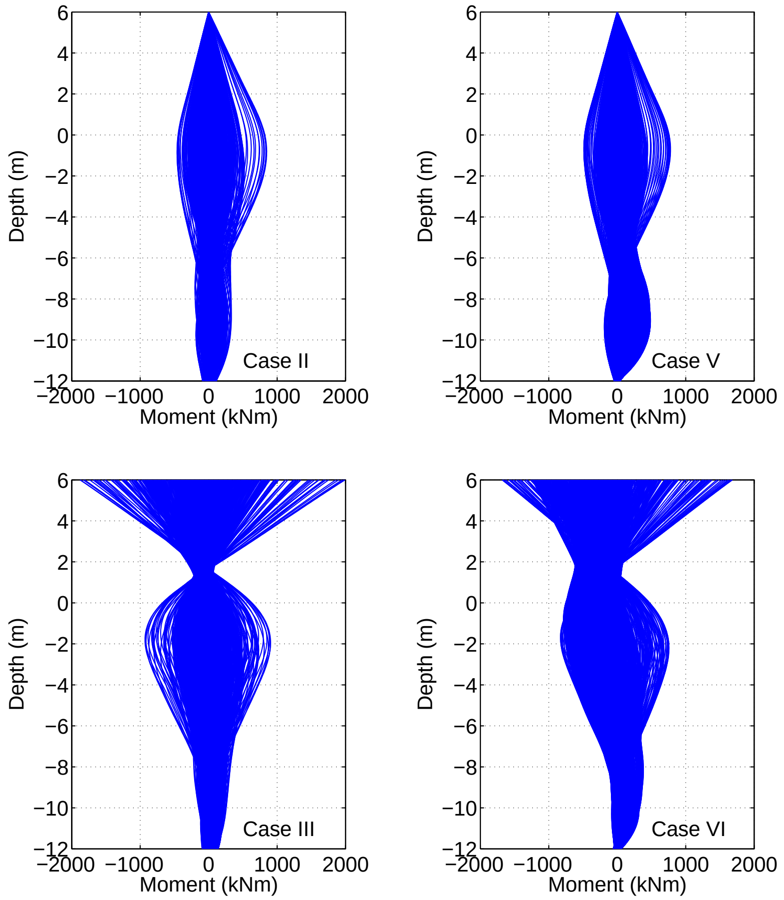


Figure 505.14: Envelope of bending moments for pile–column system for Cases II, III, V and VI.

at the bottom and would have thus attracted much larger forces too, potentially leading to pile damage and yielding.

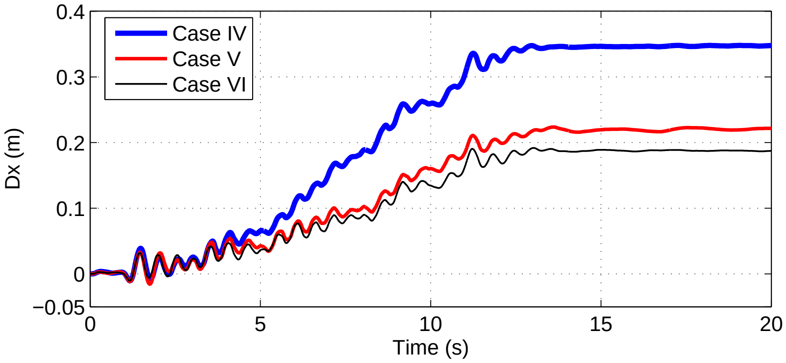


Figure 505.15: Down–slope movement at the ground surface (model center) for Cases IV (no pile), V and VI (with pile–mass system).

Chapter 506

Slope Stability in 2D and 3D

(1999-2010-)

(In collaboration with Dr. Nima Tafazzoli)

506.1 Chapter Summary and Highlights

506.2 Introduction

Presented here is a brief overview of a work on determining factors of safety for a 3D slope for one of the prominent dams in the midwest, for different saturation and water level conditions.

506.3 Dam Section Geometry

Figures 506.1 and 506.2 shows the satellite picture of location

Figure 506.1: Dam satellite picture. Upstream (south facing) three dimensional slope at connection of the embankment and concrete gravity dam is evaluated for potential stability problem.

Figure 506.3 shows the sonar data for water depth measurements at the location of a 3D slope in question.

An interesting topographical feature resembling a valley should be noted at the toe of (beginning of) wrap around section of upstream section of the embankment. That topographical feature is also apparent in a photo taken during Wold Creek Dam construction in 1948, show in Figure 506.4. Figure 506.5 shows rock surface under the embankment and alluvial fill, that serves as base rock foundation for both the alluvial fill and the embankment above. Figures 506.6, 506.7 and 506.8 show sections of the curved, 3D slope of the upstream embankment next to the concrete dam section. In particular, Figure 506.6 shows a dam section perpendicular to the dam axes. Note a significant extent of the alluvium that was left in place during dam construction. Similarly, the alluvium is present in both section inclined at 45^o to dam axes (Figure 506.7) and a section parallel to the dam axes (Figure 506.8).

Analyzed 3D, curved section of the upstream embankment is also shown in a photograph in Figure 506.9, taken on morning of April 21st, 2010. Note a significant length/extent of the two sheet pile walls (one running along the length of work platform and the other one at the end of work platform, next to

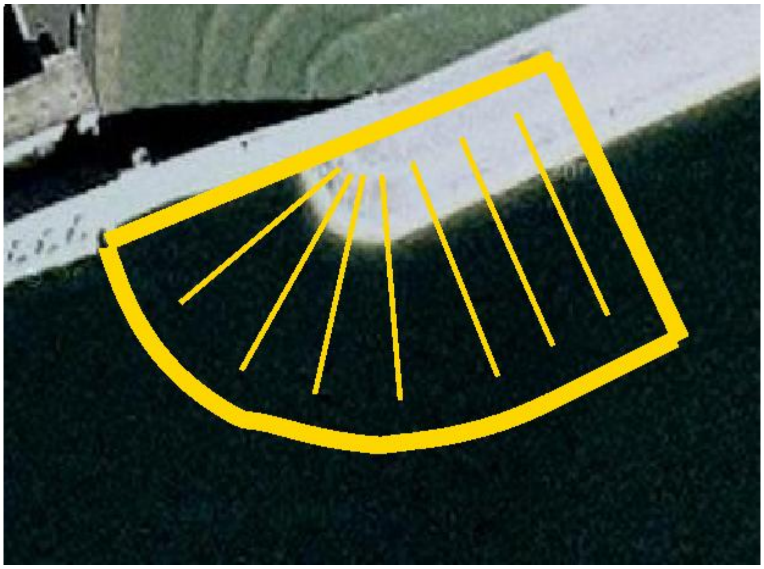


Figure 506.2: Dam satellite picture with location of 3D slope under consideration for potential stability problem.

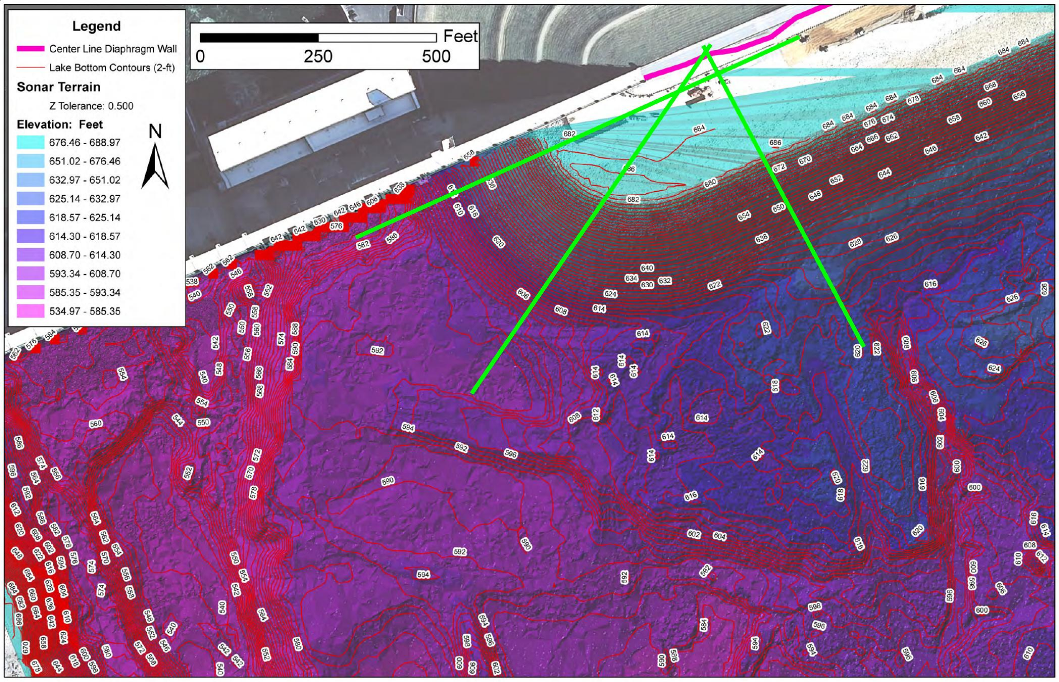
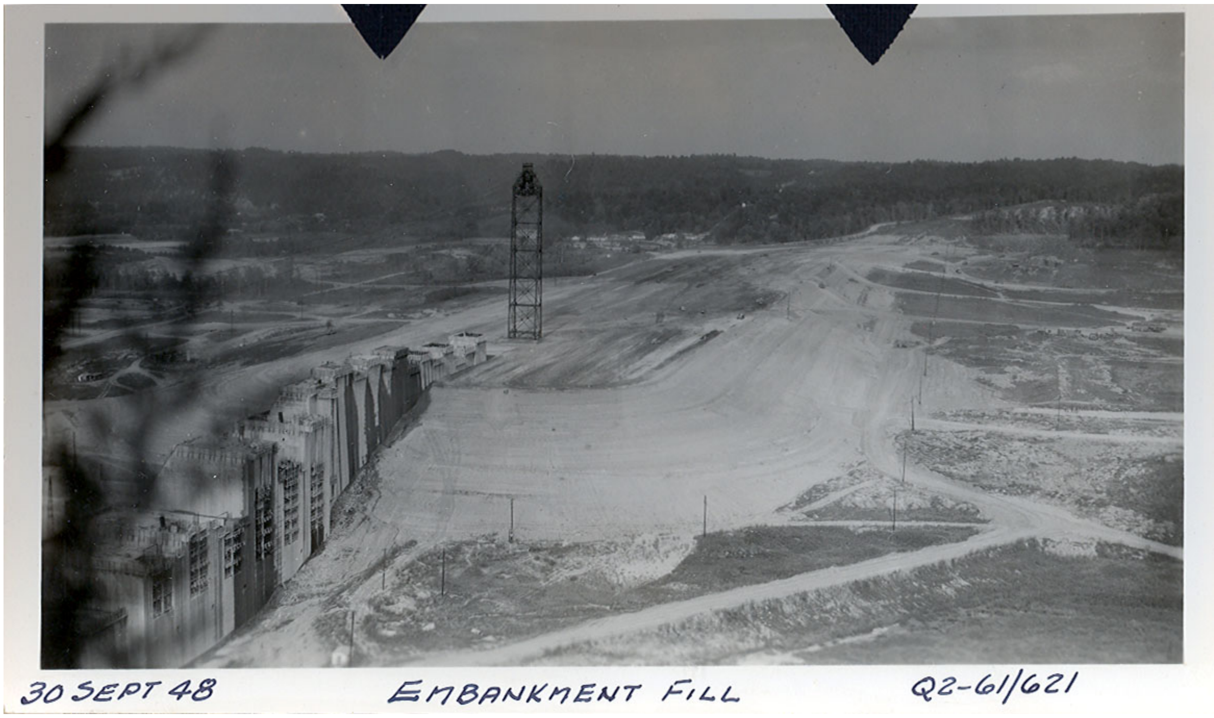


Figure 506.3: Dam sonar data for water depth for the analyzed 3D slope.

concrete dam section).



30 SEPT 48 EMBANKMENT FILL Q2-61/621

Figure 506.4: Dam during construction, 1948. View toward the upstream curved slope. Note shallow valley where the straight section of pole line ends, approximately at the end or straight slope, toward the beginning of curves slope section (Photo courtesy of Mike Zoccola, USACE).

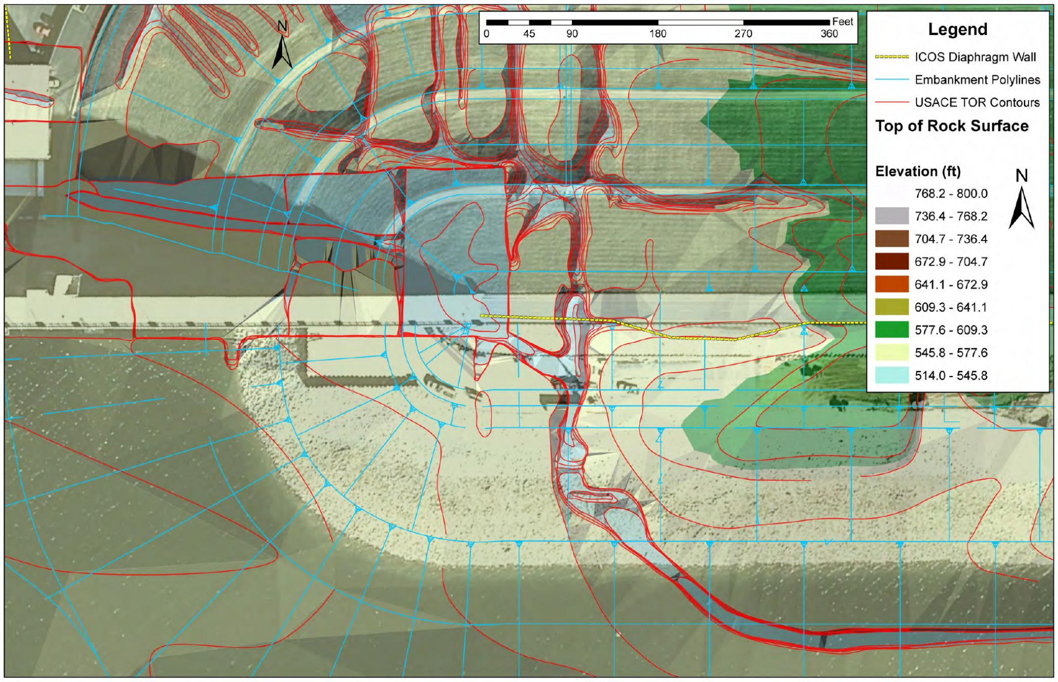


Figure 506.5: Dam rock surface data under the embankment and alluvial fill (under analyzed 3D slope), used for defining FEM model boundary conditions.

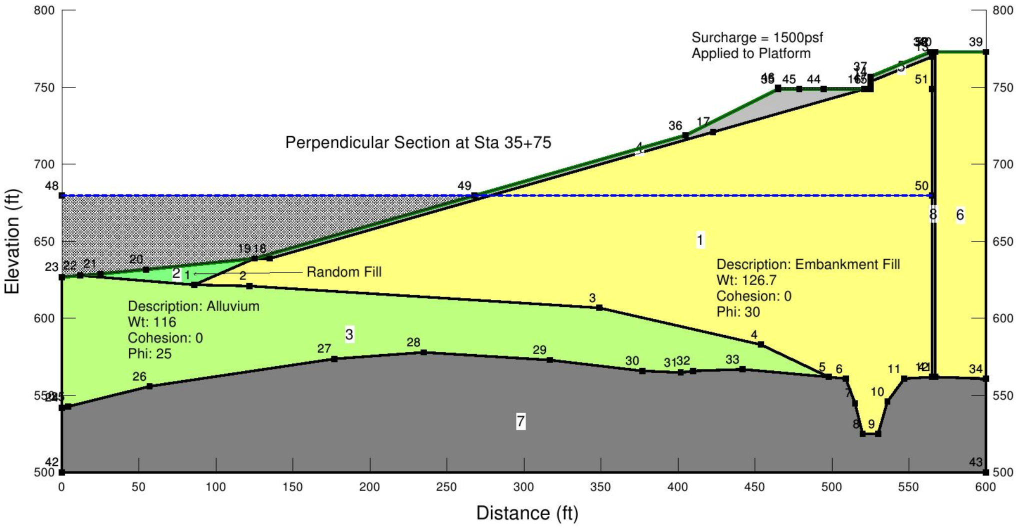


Figure 506.6: Dam section perpendicular to the dam axes for analyzed 3D slope.

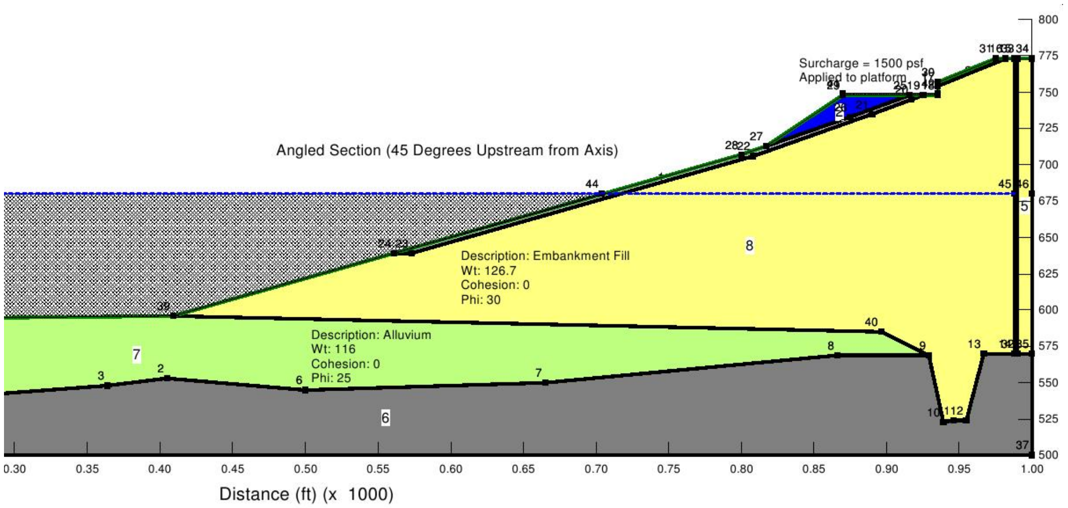


Figure 506.7: Dam section inclined at 45^o to dam axes for analyzed 3D slope.

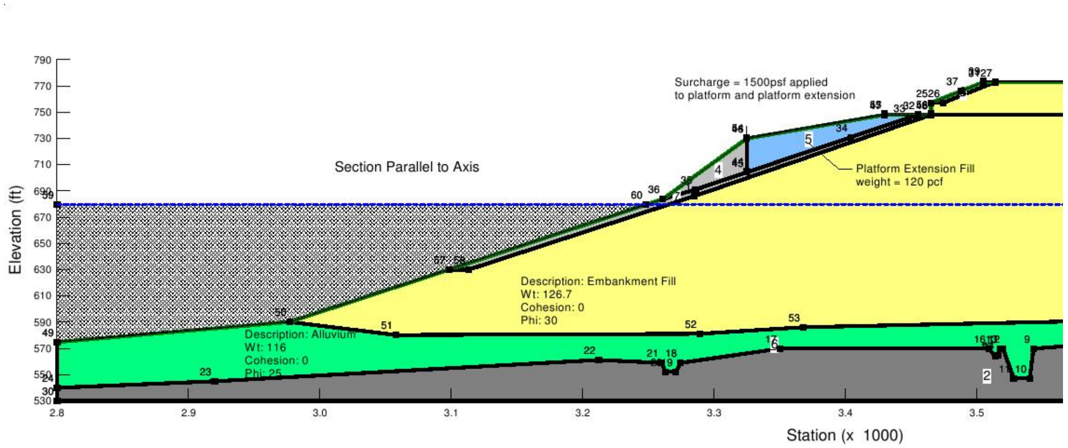


Figure 506.8: Dam section parallel to the dam axes for analyzed 3D slope.

Figure 506.9: Dam view toward the curved, "wrap" section, upstream side.

506.4 Finite Element Modeling

Analysis of 2D and 3D slope stability problem was performed using finite element method. In particular, the strength reduction method was used in conjunction with 2D and 3D finite element models to assess factors of safety. Main focus was on determining the difference between 2D and 3D factors of safety.

506.4.1 Material Models

During initial teleconference, in early April, it was decided to perform two dimensional and three dimensional analysis using finite element method for two limiting cases of material behavior:

- fully drained material behavior, with material defined by a friction angle and no cohesion,
- fully undrained material behavior, with material defined by an undrained shear strength.

Parameters for drained material modeling are given in table 506.1 below. In addition to that table 506.2 below, defines parameters for undrained material modeling. Those material parameters were agreed upon after few and a discussion. It should be noted that USACE has initiated a testing program that will determine material parameters of the embankment and the alluvium with higher certainty.

Table 506.1: Drained strength material parameters.

Material	Unit Weight [pcf]	ϕ [deg]	S_u [psf]
Embankment Fill	126.7	30	0
Alluvial Soil	116	25	0
Random Fill	120	20	0
Riprap	110	40	0
Shot Rock Fill	145	45	0
Platform Extension Fill	120	20	0

Two material models were used for analysis. For drained analysis, a Drucker Prager material model was used. This model is described in some detail in Section 3.5.7 of my lecture notes Jeremić et al. (1989-2025) (available online through my web site). It is important to note that single value of friction angle was used, thus rendering friction angle for both compression and extension the same. While there might be an influence of the difference of friction angle for compression and extension, such difference was not analyzed here. For undrained analysis a total stress, von Mises model was used. This model is also described in some details in Section 3.5.6 of my lecture notes Jeremić et al. (1989-2025). Both models used here feature perfectly plastic behavior after yielding, with associated plastic flow.

Table 506.2: Revised undrained strength material parameters.

Material	Unit Weight [pcf]	ϕ [deg]	S_u [psf]
Embankment Fill	126.7	0	2000
Alluvial Soil	116	0	1000-1500
Random Fill	120	0	500
Riprap	110	40	0
Shot Rock Fill	145	45	0
Platform Extension Fill	120	20	0

506.4.2 Two Dimensional Models

Three two dimensional (2D) models were developed in order to test the effects mesh size has on quality of simulations. Developed 2D mesh was based on perpendicular cross section, shown in Figure 506.6. Mesh for a 2D model, shown in Figure 506.10, represents a very coarse mesh, which, if higher order elements are used (with displacement interpolation higher than linear) might actually work quite well. However, with linear interpolating displacements, this mesh is clearly of low quality. However, this mesh was used as a first iteration, and to gain initial insight into 2D behavior.

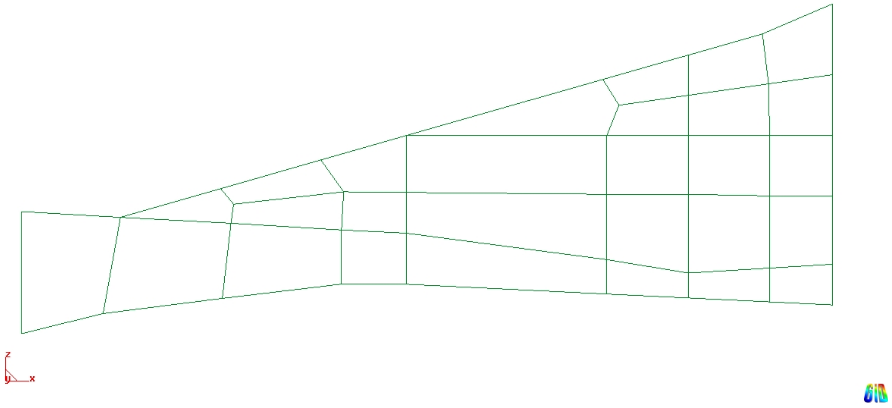


Figure 506.10: Coarse mesh model used for the 2D upstream section of the Dam.

Meshes shown in Figures 506.11 and 506.12, represent a refinement of the first mesh. Models using both meshes were tested for both drained and undrained material models. Results for both fine mesh (Fig. 506.11) and finest mesh (Fig. 506.12) were the same for all practical purposes (differences were negligible for the purpose of analysis), hence it was decided to use fine mesh (Fig. 506.11) for all

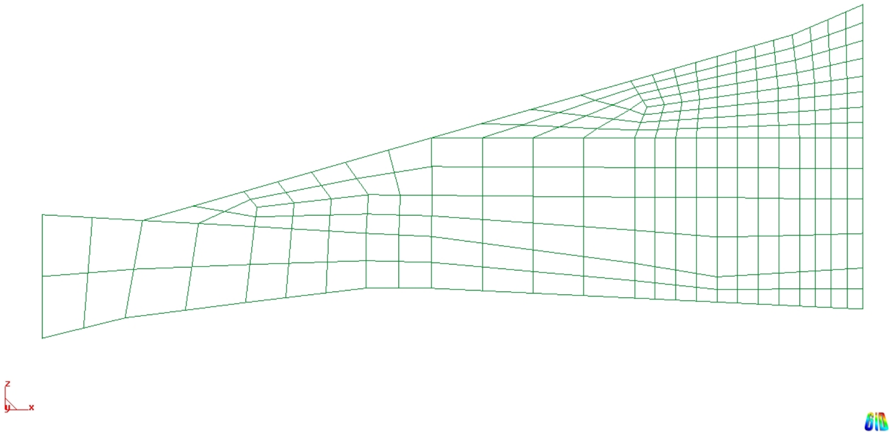


Figure 506.11: Fine mesh model used for the 2D upstream section of the Dam. This mesh was chosen as appropriate enough for all subsequent 2D analysis.

subsequent analysis in 2D.

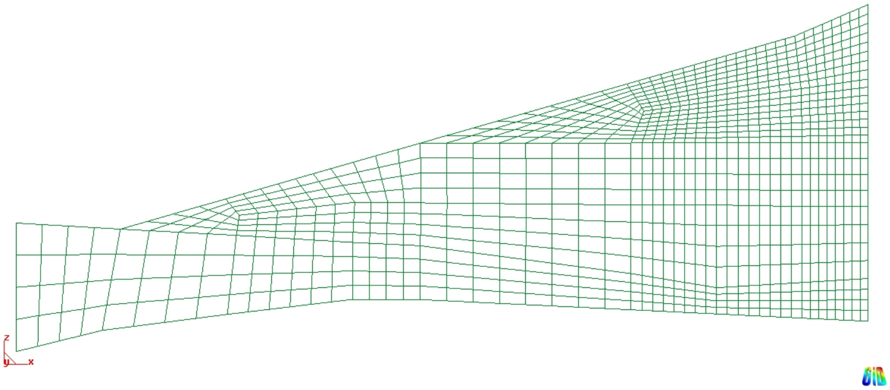


Figure 506.12: Much finer mesh model used for the 2D upstream section of Dam. Results obtained this mesh were the same as previous mesh (shown in Figure 506.11), leading to the choice of the previous mesh (Figure 506.11) for all 2D analysis.

It is important to note that all 2D meshes were actually developed using 3D, brick elements (in this case a linear interpolation, 8 node bricks). Appropriate boundary conditions were used to prevent out of plane displacements. While this approach uses extra resources (use of 3D element for a 2D problem) it allowed us to use the very same element and material models for both 2D and 3D problems, thus removing modeling uncertainty and emphasizing accuracy of determination of differences between 2D and 3D slope stability. The finite element used for analysis in 2D (and later in 3D) was based on an eight node brick element, described in some more details in Section 2.3 of my lecture notes Jeremić et al. (1989-2025).

Mesh in Figure 506.11 has 250 brick finite elements while finest mesh in Figure 506.12 has 1000 brick finite elements. Boundary conditions for the 2D mesh were such that right vertical boundary (at the current Barrier wall) was allowed to move down but not horizontally. Lower mesh boundary (contact with base rock) was fully supported, while left vertical boundary (far into alluvium) was also allowed to move vertically but not horizontally. Work platform was modeled using loads, instead of extending the mesh to include the shape of the platform. This allowed for additional modeling flexibility, without the loss of accuracy. Loading tests with and without the work platform show that its influence on calculated factors of safety is negligible. Figure 506.13 shows a 2D model with location of work platform, water level at 680 ft, boundary conditions and the extent of alluvium layer.

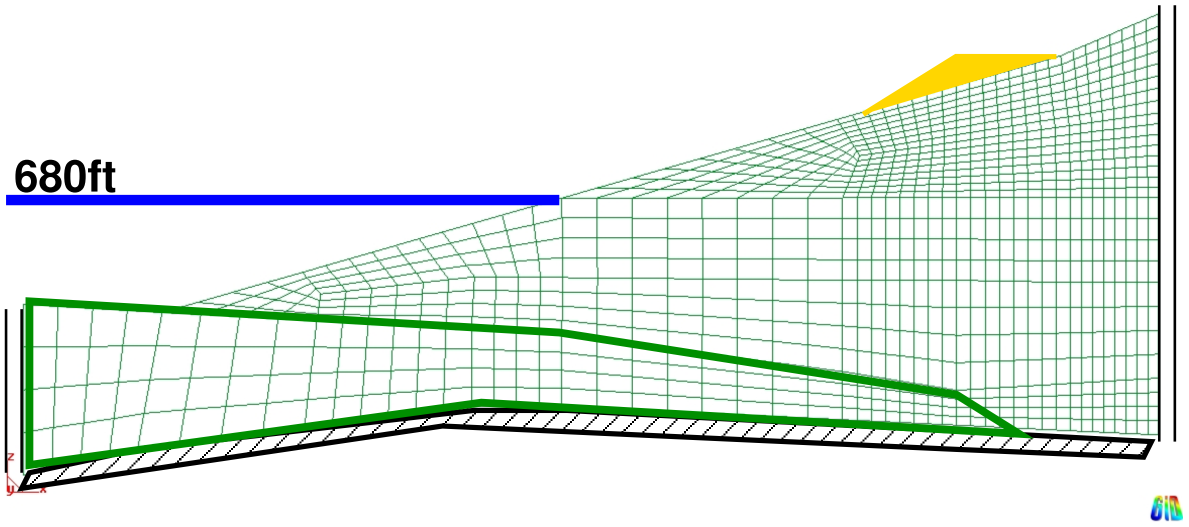


Figure 506.13: A 2D model with boundary conditions, water level at 680 ft and the work platform as load.

506.4.3 Three Dimensional Models

The Dam is a full three dimensional solid, consisting of foundation rock, concrete section and soil embankment section with barrier wall(s) (old and new one when finished). Ideally we would model the complete foundation-concrete-embankment-wall system with porous solid (soil skeleton), pore fluid (water in fully and partially saturated pores), structural components (barrier walls) as well as solid concrete section.

Particular focus on assessing 2D versus 3D slope stability for the upstream, curved (wrap around) section, as well as the very short time frame for this project, dictated development of small part of the complete model. The most important feature of the 3D model was to use optimal model size (extent) so that all important features are properly captured. This was done by developing 3D model for upstream curved, wrap slope in two stages. First stage comprised development of the conical portion of mesh, which was then extended, using a straight second stage, extending the model into the embankment by 200 ft (60 meters).

Conical Section of a 3D Slope Three sections shown in Figures 506.6, 506.7 and 506.7 were used to develop three dimensional mesh. Conical wrap around part of the mesh features 1344 brick elements and is shown in Figure 506.14.

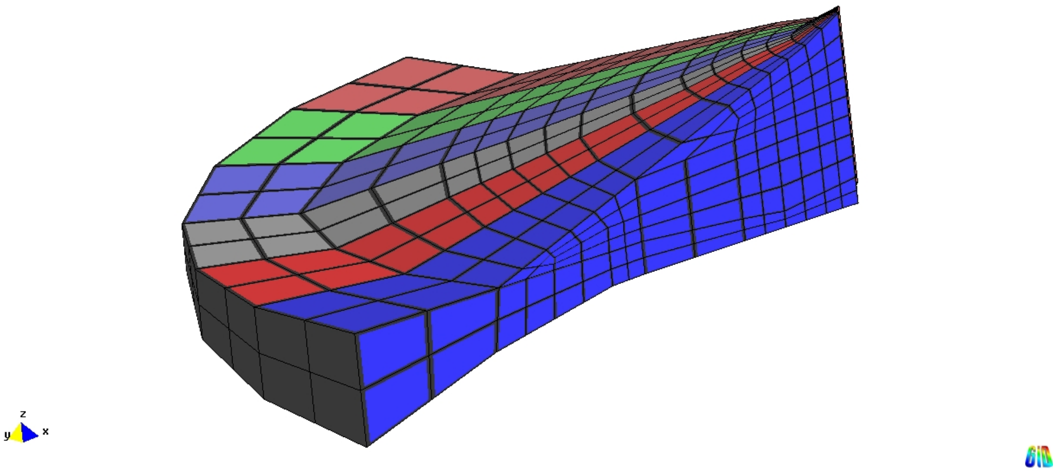


Figure 506.14: Conical section of the 3D upstream slope of the Dam, 3D mesh.

Boundary conditions are such that two vertical sections (one adjacent to the concrete monolith and the one adjacent to straight embankment sections have free vertical and in plane horizontal displacements while no out of plane displacement. A small quarter cylinder boundary was also developed at the top of the curved, wrap around section. This small part of boundary (radius of 2 meters) was necessary to properly mesh radial directions of the 3D model. Boundary conditions at that location are such that they prevent radial deformation while allowing vertical and tangential movement. This is achieved using short, stiff truss elements.

At the lower end of the model (well into alluvium), curved, radial section of the model is supported in such a way that radial deformation is prevented while vertical and horizontal tangential movement is allowed, again achieved using short, stiff truss elements.

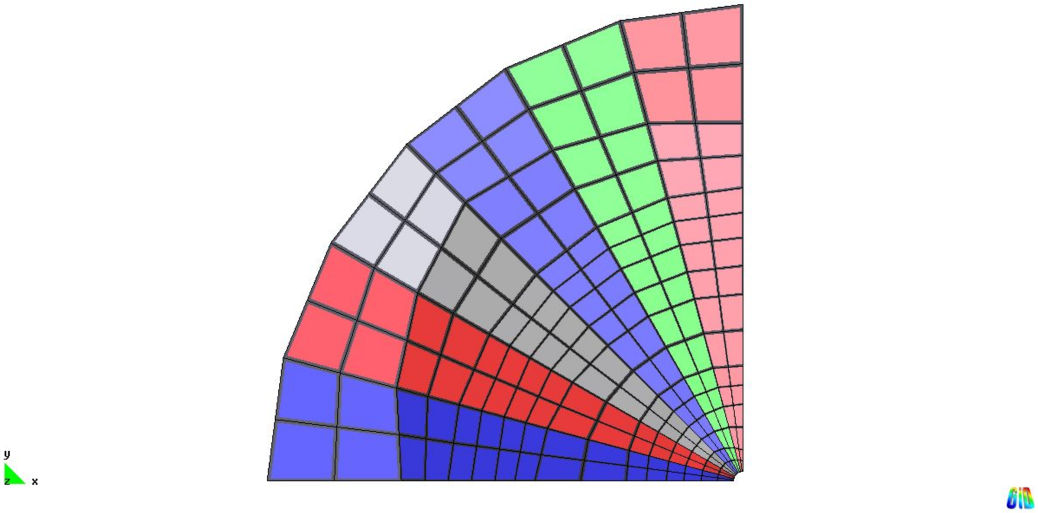


Figure 506.15: Curved section of the 3D upstream slope of the Dam, 3D mesh, top view. Note the difference in extent of the actual embankment (slightly different color of elements, alluvium is represented by last two layers of elements on the outskirts of model. Also note a small cylindrical section close to center of conical slope, that was used to control meshing.

Conical and Straight Section of a 3D Slope While conical section of the slope represent accurately the curved section of the dam, influence of the straight section cannot be neglected. A section (200 ft, 60 meters) of a straight embankment was added to a curved section in order to have a more realistic modeling of the complete upstream 3D embankment. Figures 506.16 and 506.17 show extended 3D mesh for curved and part of straight section of the upstream slope.

Boundary conditions were similar to the previous case for conical part of mesh, except where mesh extension was applied. Extended mesh was fully supported at the bottom, while vertical faces were supported in such a way not to have out of plane displacements while in plane (horizontal and vertical) displacements were left free. Loads from work platform and from water in the lake were applied as appropriate nodal forces.

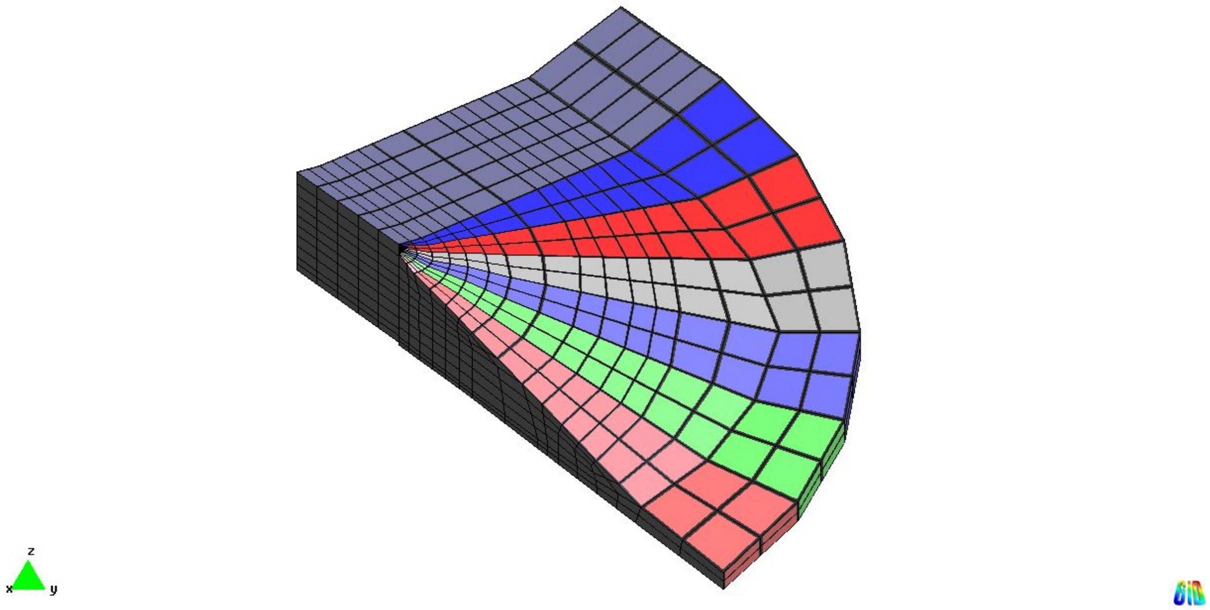


Figure 506.16: Curved and straight section of the 3D upstream slope of the Dam, 3D mesh.

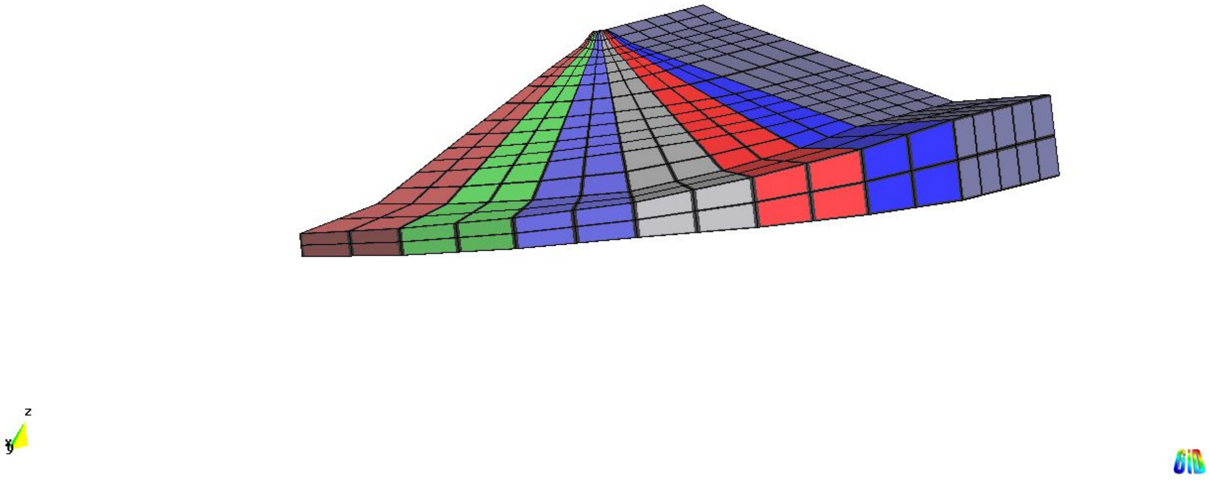


Figure 506.17: Curved and straight section of the 3D upstream slope of the Dam, 3D mesh.

506.4.4 Modeling Issues

Mesh Size Multiple mesh sizes were tested to alleviate any mesh dependency of the computed solution close to stability loss (close to but not into the development of localized deformation).

Following the Equilibrium Path Numerical integration of elastic-plastic constitutive equations was performed using both explicit and implicit algorithms (see Jeremić et al. (1989-2025), Sections 3.3 and 3.4). This was necessary in order to verify the solution accuracy. For explicit constitutive computations, explicit global solution scheme was used (see more detailed description of equilibrium following path algorithms in my lecture notes Jeremić et al. (1989-2025), Chapter 7). For implicit constitutive integrations, Newton method (cf. Dennis and Schnabel (1983), and Jeremić et al. (1989-2025), Chapter 7) was used on both constitutive and global solution levels. Global solution advancement was constrained using both load control (appropriate for factor of safety stability simulations) as well as displacement control (in order to validate load control solutions). Detailed discussion on both load and displacement control algorithms is given in lecture notes Jeremić et al. (1989-2025), Sections 7.3. and 7.4.

Strength Reduction Method Strength reduction method is often used with finite elements to assess factors of safety for slopes, foundations and other problems in geotechnical engineering where elastic-perfectly plastic failure is expected Duncan (1996) Griffiths and Lane (1999). Strength reduction method can be performed in two ways:

- Apply self weight to a model where strength parameters are reduced by Factor of Safety (FS). Perform analysis number of times for larger and larger FS, until the finite element system is not in equilibrium, that is external forces cannot be balanced by internal forces (see Section 7.1 in Jeremić et al. (1989-2025)). This approach has the benefit of not being mesh dependent, that is, for elastic – perfectly plastic material models (without softening), dependence of solution on mesh refinment does not become an issue (Lu et al., 2009). One potential drawback is that since there is no softening involved, deformation will not localize into localized zone, rather failure mode is somewhat diffuse, however still following proper displacement of blocks of material as system becomes unbalanced.
- Apply self weight to a model with full strength material parameters, and then gradually start softening the material parameters, by dividing them with FS. Perform reduction in parameters, until there is loss of equilibrium, that is until external forces cannot be balanced by internal forces (again see Section 7.1 in Jeremić et al. (1989-2025)). This approach might have a potential problem in that the mesh size/refinment will affect the solution, as material models involved are

softening, and different meshes will create different localization patterns and hence lead to different solutions. The benefit of this approach is that deformation will indeed localize in a very thin band, thus resembling the limit equilibrium approach that is popular for slope stability problems, however, such deformation patterns are mesh size/refinment dependent and thus not unique.

In this work, the first approach is used, that is self weight computations are performed for a number of models where strength parameters are reduced by FS, until such FS for which equilibrium between external and internal forces cannot be achieved.

Variable Water Levels Comparison of 2D vs 3D stability for two water elevation was originally suggested. Discussions during a meeting at a Dam site, further clarified conceptual problems with two water elevation, namely lake at 680 ft and water table in the embankment at 720 ft. For any water table that is above the lake level, there will exist a water flow (seepage) within the embankment toward the (lower level) lake water. Such seepage will create a seepage force $f_s = \gamma_w (dh)/(dl) = \gamma_w i$ where, γ_w is the unit weight of water, and $i = (dh)/(dl)$ is the hydraulic gradient. Clearly, with seepage from embankment toward the lake, such force destabilizes the embankment (slope). Finite Element formulation and tools that can model and simulate such fully coupled system (transient analysis of deforming porous soil with moving pore fluid) are available (Cheng et al., 2007; Jeremić et al., 2008; Cheng and Jeremić, 2009a,b), however material parameters for soil permeability as well as the extent of phreatic surface need to be determined. Due to unavailable test data for permeability of embankment soil and due to (high) uncertainty in the extent of phreatic surface (number of piesometric measurements placing the water table in the embankment at 720 ft were highly questionable, as discussed by the panel), only single phase soil (either effective stress or total stress) analysis were performed. This decision has no effect on analysis for cases where both lake and phreatic line in the embankment are at 680 ft. However, such decisions affects cases with different water levels (lake at 680 ft, phreatic line at):

- Fully undrained case with lake level at 680 ft, embankment phreatic level at 720 ft, was analyzed and in fact provides (based on theory of undrained soil behavor) the same factors of safety as no seepage case, that is, same water level in lake and embankment. This is due to the fact that for total stress analysis, pore fluid pressure does not influence the shear strength. In addition to that, with the assumption of fully undrained conditions, coefficient of permeability is nonexistent, there is no seepage, and hence there is no seepage force. In reality coefficient of permeability for soil is never really zero, however this case was treated as an extreme case, used purely for checking differences in factors of safety between 2D and 3D slope stability.
- Fully drained case with lake level at 680 ft, embankment phreatic level at 720 ft, was not analyzed

as neglecting seepage force would place calculated factors of safety on the *unsafe* side. While such analysis can be run using either buoyant weight approach (however somewhat inconsistent as noted by Duncan (1996)) or by mixing total stress and effective stress approach (more inconsistent) it was decided not to simulate this case.

506.5 Results: Factors of Safety

A large number of computations on a number of models were performed.

Factors of Safety for 2D Models

- Drained, Lake/Embankment at 680 ft,
  $FS_{2D} = 1.89$
- Undrained, Lake/Embankment at 680 ft, alluvium weak of most likely $(S_u = 1000 \text{ psf})$,
  $FS_{2D} = 2.22$
- Undrained, Lake/Embankment at 680 ft, alluvium at most likely value $(S_u = 1500 \text{ psf})$,
  $FS_{2D} = 2.50$

Factors of Safety for Extended 3D Models

- Drained, Lake/Embankment at 680 ft,
  $FS_{3D} = 1.78$
- Undrained, Lake/Embankment at 680 ft, (alluvium weak of most likely $S_u = 1000\text{psf}$),
  $FS = 2.0$

Comparison of 2D vs 3D Factors of Safety

- Drained, lake/water in embankment at 680 ft,
  $FS_{2D} = 1.89$ vs $FS_{3D} = 1.78$
  FS reduced by 5.8 %.
- Undrained, Lake at 680 ft, embankment at 680 ft
  (alluvium weak of most likely $S_u = 1000\text{psf}$),
  $FS_{2D} = 2.22$ vs $FS_{3D} = 2.00$
  FS reduced by 9.91 %.

- Undrained, Lake at 680 ft, embankment at 720 ft
  (alluvium weak of most likely $S_u = 1000\mathrm{psf}$),
  $FS_{2D} = 2.22$ vs $FS_{3D} = 2.00$
  FS reduced by 9.91 %.

506.6 Uncertainty of Results

Full probabilistic analysis for large scale, elastic-plastic finite element models is currently within reach (Sett et al., 2011a) and will probably become a standard simulation tool within next 5 to 10 years. However, for such an analysis, a(n) (extensive) site characterization program is necessary, if probabilistic simulations are to be useful (Sett and Jeremić, 2009). Alternatively, one can use a simplified methodology, described in detail by Duncan (2000a), that can give sensible estimates of mean and standard deviation of common results obtained in soil mechanics. However, in view of lack of consistent material properties for the embankment further work on estimating uncertainty of factors of safety was discontinued. It is important to note that a program was initiated by the USACE to perform extensive testing of embankment soil and upon completion of that program, it will be possible to perform a simplified (and even a more accurate, full probabilistic analysis, as mentioned above) estimation of influence of material uncertainty on obtained factors of safety.

506.7 Conclusion

The main purpose of this study was to investigate changes in factors of safety for failure between 2D and 3D slope problems for curved part of the upstream section of a Dam embankment. Detailed models (both 2D and 3D) were developed for the upstream section of Wold Creek Dam and were analyzed. Strength reduction method was used to assess factors of safety in both 2D and 3D. It was shown that for a number of different cases (drained or undrained soil, lake at 680 ft and water table in the embankment either at 680 ft or at 720 ft for undrained case) the factor of safety is reduced in 3D when compared to 2D. Such reduction, however was not significant (up to approx. 10 %).

Of particular importance for this factor of safety comparison (2D vs 3D) was the robustness of simulations and a number of methods were used to ensure that obtained factors of safety, determined as failure to converge upon strength reduction, were due to loss of equilibrium, and not due to numerical problems leading to loss of convergence.

Appendix given below, contains two additional sections, describing numerical tool used in factor of safety computations (available in public domain) and deformation patterns close to failure for select cases.

506.8 Displacement Patterns

Hypothetical displacement patterns are presented below in Figures 506.18, 506.19 and 506.20. It should be noted that displacements that results from simulations of stability by strength reduction method are not necessarily the failure modes, rather, they are just a side-product of simulation. However, it is instructive to inspect those displacements just before equilibrium is lost in order to gain a better understanding of potential failure patterns.

Figure 506.18 shows a 2D displacement pattern for a drained case with lake and embankment water level at 680 ft where $FS_{2D} = 1.89$. Noted is a clear rotating pattern of the slope just before loss of equilibrium.

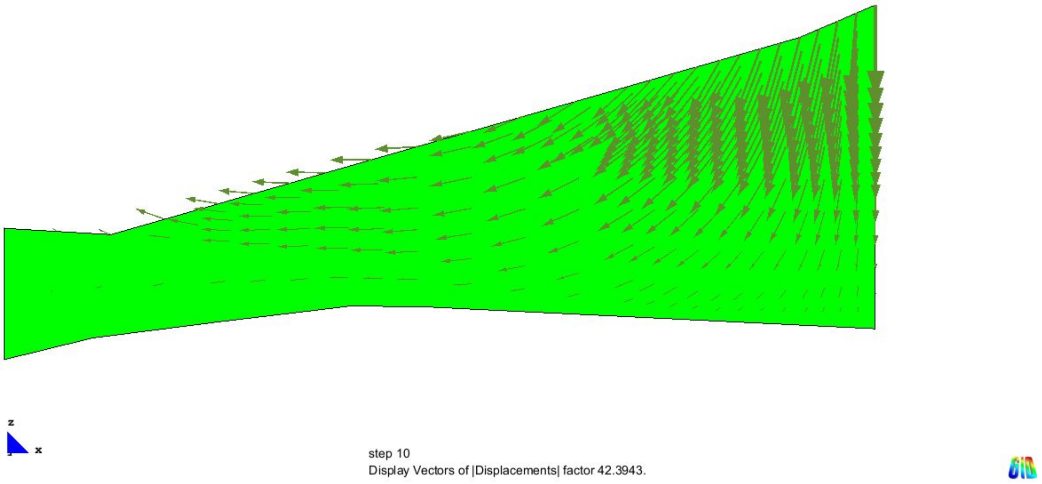


Figure 506.18: Drained case, lake and embankment water lever 680 ft, $FS_{2D} = 1.89$.

Figure 506.19 shows a 2D displacement pattern for an undrained case with more likely (stronger) alluvium ($S_u = 1000 \text{psf}$). Note that while displacement pattern is somewhat similar to the previous case, the block movement on top of alluvium is much reduced, and main failure is through rotation mechanism.

Figure 506.20 shows a plan view of a 3D displacement pattern (vectors) for an undrained case with limiting weak alluvium and lake at 680 ft. Noted is the dominant 3D deformation pattern by which the curved and straight sections of the slope rotate around vertical axes and move toward the lake with direction biased toward dam axes. It is important to note that there is no single 2D section of the 3D

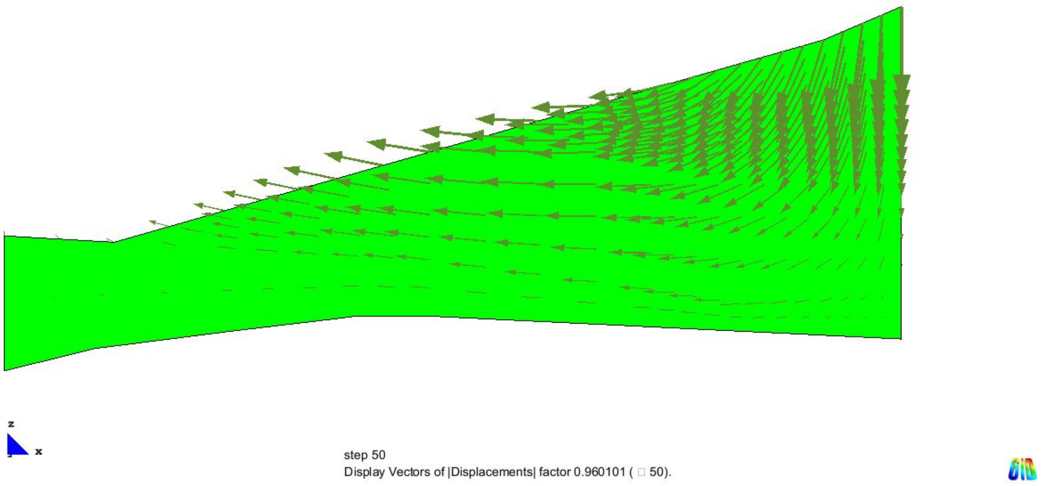


Figure 506.19: Undrained, L680 E680, Most Likely to Weak Alluvium, $FS_{2D} = 2.22$.

slope that can be used to model such failure as failure mode is fully three dimensional.

Figure 506.20: Plan view of a 3D undrained case, displacement vectors, lake and embankment at 680 ft, $FS_{3D} = 2.00$.

Chapter 507

Concrete Structures

(1989-2017-2018-2019-2020-2021-)

(In collaboration with Prof. Han Yang and Dr. Hexiang Wang)

507.1 Chapter Summary and Highlights

507.2 Concrete Wall/Membrane

507.2.1 Introduction

OECD organized project to investigate modeling of concrete walls that have alkali - silica reaction (ASR) has been going on for few years. Our modeling and simulation group, based at the University of California at Davis, in California, USA, and at the Lawrence Berkeley National Laboratory at Berkeley, in California, USA, has joined this effort only at the beginning of 2018. We have gotten project description and already existing reports from participants, in January/February of 2018. We managed to quickly make a model of the wall and to calibrate parameters for a concrete model that we use in our modeling and simulation program.

We rely on and used our program, called the Real-ESSI Simulator (Realistic Modeling and Simulation of Earthquakes, and Soils, and Structures and their Interaction), that is also known as the MS ESSI Simulator[1] (Jeremić et al., 1988-2025) (`http://real-essi.info/`). Real-ESSI Simulator is a software, hardware and documentation system for high fidelity, high performance, time domain, nonlinear/inelastic, deterministic or probabilistic, 3D, finite element modeling and simulation of (a) statics and dynamics of soil, (b) statics and dynamics of rock, (c) statics and dynamics of structures, (d) statics of soil-structure systems, and (e) dynamics of earthquake-soil-structure system interaction.

The Real-ESSI Simulator systems is used for the design and assessment of static and dynamic behavior of infrastructure objects, including buildings, bridges, dams, nuclear installations, tunnels, etc. For design, multiple linear elastic load cases can be combined and design quantities, sectional forces exported for design and cross section dimensioning. For assessment, realistic inelastic load staged analysis is performed, with all the inelastic components properly modeled, as listed below, and with all the simulation, algorithmic features available, as listed below. Analysis is performed in order to assess safety and economy of objects. The work on Real-ESSI Simulator is based on a philosophy that aims to develop modeling and simulations that inform and predict rather than (force) fit.

507.2.1.1 Motivation

The alkali - silica reaction (ASR), is a reaction that occurs over time in concrete, between the alkaline cement paste and the silica found in many common aggregates, triggered by the presence of moisture. The ASR does volumetric expansion of interface between cement paste and the aggregate, thus resulting

---

[1] The Real-ESSI Simulator was developed in collaboration and with financial support from the US-DOE, US-NRC, US-NSF, CNSC-CCSN, Caltrans, etc.

in potentially damaging tensile stress within concrete. A comprehensive study of this effect on concrete properties has been done at University of Toronto, funded by Canadian Nuclear Safety Commission (CNSC). Five shear walls, two regular (REG A, REG B) and three ASR (ASR A1, ASR B1, ASR B2) along with their control specimens were tested as a part of Phase 1, Phase 2, and Phase 3 of the investigation program [ref Prof. Sheikh presentation].

It was proven that ASR causes changes in mechanical properties of the concrete, introducing inaccuracies if conventional analysis is used, without taking into account ASR. Beyond a certain age, modulus of elasticity and tensile strength of ASR concrete decrease, in contrast to regular concrete. Observed is degradation of the ductility in ASR specimens. While concrete expands, the bonding strength between concrete and rebars decreases causing drop in ductility. However, despite these changes in concrete, tested ASR shear wall showed higher peak shear strength compared to regular concrete.

This newly understood behavior of ASR concrete necessitates development of reliable numerical simulation, to be used for the design of new structures as well as to predict behavior of existing ones.

In order to model behavior of the examined wall specimens, Real-ESSI Simulator, (Jeremić et al., 1988-2025) was used.

Plane stress finite element with an inelastic plane stress material model for concrete was used for modeling of shear wall specimens. All inelastic material models and finite elements inside the Real-ESSI Simulator feature accurate energy dissipation modeling and calculation (Yang et al., 2018, 2019a).

Results from ASCET benchmark tests are used for validation of shear behavior for plane stress inelastic reinforced concrete wall tests.

507.2.2 Model Availability

The Real-ESSI input files for the ASR concrete wall example are available HERE.

507.2.3 Model Development

507.2.3.1 Model Mesh

Finite element model includes all relevant parts of the experimental setup, as shown in Figures 507.3 and 507.2.

Beam slabs and steel plates are modeled using 27NodeBrick element, while steel bolt is represented as a single truss element. The shear wall is modeled using nonlinear layered plane stress elements. For the web part of the wall, the elements have a horizontal rebar layer, a vertical rebar layer, and an unconfined concrete layer. For the flanges (columns) of the shear wall, the elements have an additional layer of confined concrete. It is emphasized that the main wall is really made from unconfined concrete, and the

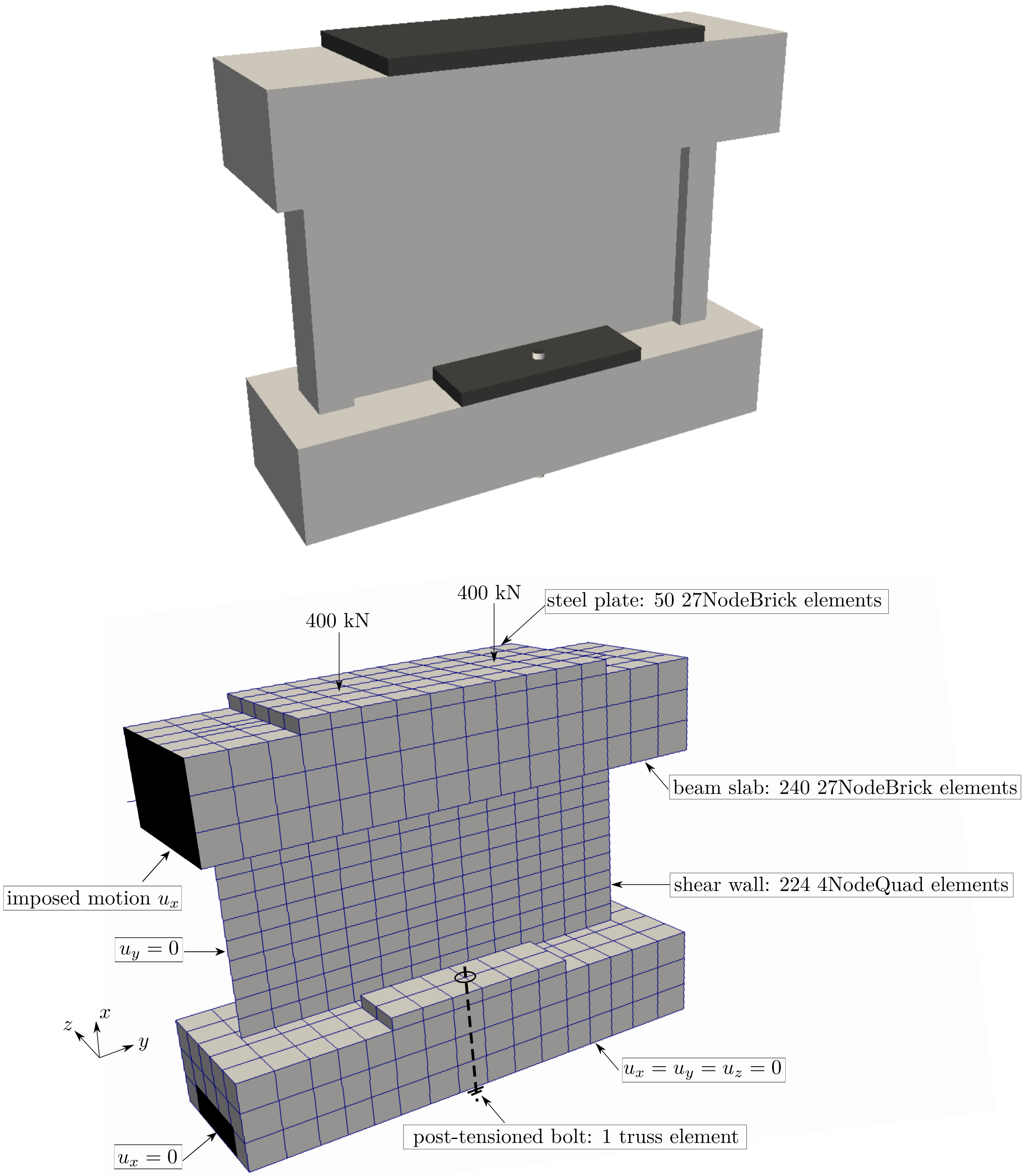


Figure 507.1: 3D rendered view and finite element mesh of the reinforced concrete shear wall model.

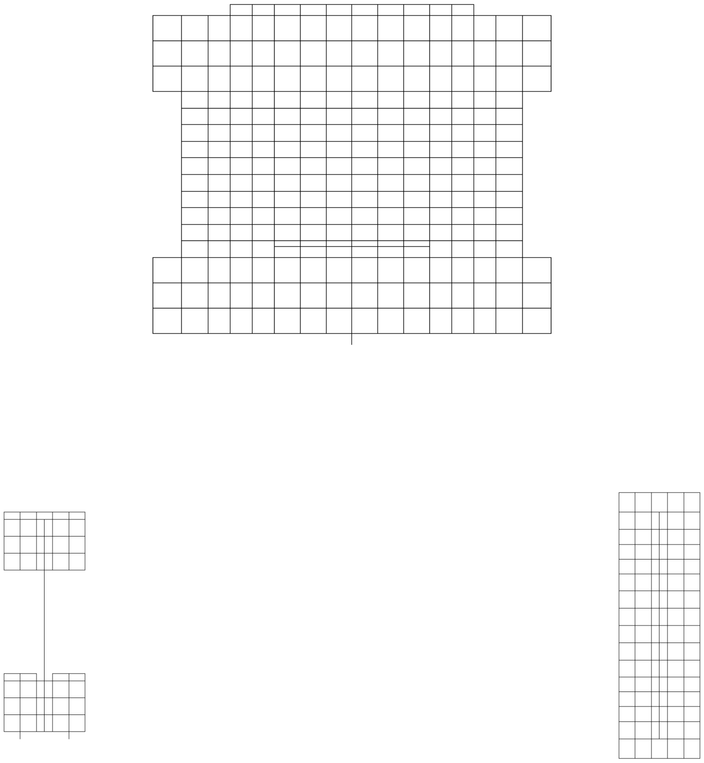


Figure 507.2: Different views of the finite element mesh.

only actually confined concrete is within concrete flanges. Detailed rebar plan and model dimensions are shown in Fig 507.3.

The bottom of the model is restrained in all directions, while the lateral sides of the bottom beam slab are restrained in direction of imposed motion. Since the shear wall consists of 2D plane stress elements the out-of plane displacement is also precluded. The sides of the top beam slab are also restrained to have the same displacement, which is important to represent the boundary conditions of the physical experiment. Initial model included inelastic contact/interface elements (stick-slip and gap open and close) at the bottom boundary. However it was concluded that there will be no slip and there is no gap opening so these elements were removed in order to speed up computations.

To correctly simulate the loading process of the experiment, four loading stages are applied:

1. Self-weight loading is applied to the whole model.

2. To represent post-tensioned force in the truss (bolt) element, the truss is stretched so that an adequate force is obtained and after that the bottom of the bolt is fixed.

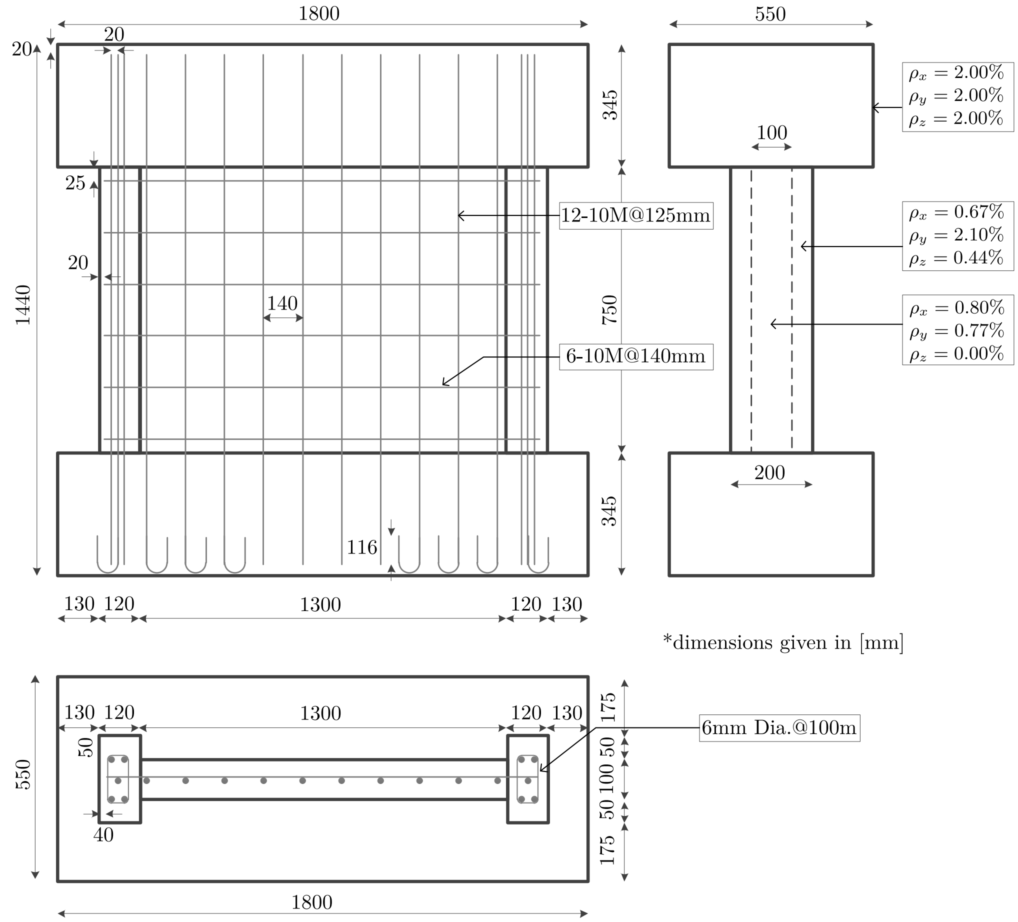


Figure 507.3: Schematics for the rebar plan of the shear wall model.

3. Two-point vertical loading is applied to the top steel plate.

4. Cyclic horizontal loading is applied (using displacement control) to the sides of the top beam slab.

507.2.3.2 Plastic Damage Concrete Material Model

Details for this model are given in Section 104.9.

507.2.3.3 Uniaxial Steel Material Model

Details for this model are given in Section 104.9.

507.2.3.4 Material Model Parameters

There are two approaches to determine the ASR affected concrete model parameters. The first choice is to obtain concrete samples directly from the existing structure. Those samples can be tested in the laboratory to determine mechanical properties of the concrete material. This is a preferred way to obtain material properties.

In this project, the material properties for ASR affected concrete were obtained using laboratory, for concrete aged to 260, 610, and 995 days, as shown in Table 507.1. These values are used in the material models for finite element simulations.

Table 507.1: Mechanical properties for the ASR concrete at 260, 610, and 995 days.

Wall	Age (Days)	Compressive Strength (MPa)	Tensile Strength (MPa)	Elastic Modulus (MPa)	Expansion (%)
ASR A1	260	63.7	3.24	35750	0.190
ASR B1	610	67.1	N/A	32600	0.215
ASR B2	995	63.0	3.18	28100	0.223

The second approach is to estimate change of material properties, future material properties, based on currently measured material properties and empirical correlations. In what follows, it is demonstrated how to predict the mechanical properties of the ASR concrete at 610 and 995 days using the values at 260 days.

First, according to data provided for ASCET II workshop, the maximum ASR expansions of the concrete prisms in longitudinal and transverse directions are approximately 0.25% and 0.28%. For simplicity, an average value of 0.265% is chosen.

Estimations of ASR expansion rates at 260, 610, and 995 days are needed. According to various ASR models (Charlwood et al., 1992; Saouma and Perotti, 2006), the rate of expansion after about 365 days is very small. So it is reasonable to use the maximum value of expansion (0.265%) for estimating mechanical properties of ASR concrete at 610 and 995 days. At 260 days, the expansion is estimated to be 0.23%.

The correlation between normalized mechanical properties of ASR concrete and rate of expansion/swelling from Capra and Sellier (2003) is used. At 0.265% swelling, at 365 days, the normalized compressive strength is 0.60 while the normalized tensile strength is 0.55. The normalized elastic modulus ranges from 0.60 to 0.70.

On the other hand, at 0.23% swelling, at 260 days, the normalized compressive strength is 0.65 while the normalized tensile strength is 0.60. The normalized elastic modulus is approximately 0.70. Therefor, from 260 days to 365 and consequently at 610 and 995 days, the compressive and tensile strengths both decrease approximately 92%. Similarly, the elastic modulus decreases approximately 85%.

Using these values, for ASR concrete older than 365 days , the compressive strength is 58.6MPa, the tensile strength is 2.98MPa, and elastic modulus 30387MPa. Note that these values match well with the experimental results shown in Table 507.1. It is important to note that concrete is a composite material and that expansion of mass of concrete, within confines of reinforcing, can produce increase in confinement, which can have beneficial effects on concrete behavior.

507.2.4 Modeling of Energy Storage and Dissipation

Described in this section is modeling and calculation of energy storage and dissipation within concrete and steel. This section is based on recent work by Yang et al. (2018, 2019a).

507.2.4.1 Plastic Damage Concrete Material

The Helmholtz free energy potential postulated for the plastic damage concrete material has the form:

$$\psi(\epsilon_{ij}, \epsilon^p_{ij}, d^+, d^-) = (1 - d^+)\psi^+_0(\epsilon_{ij}, \epsilon^p_{ij}) + (1 - d^-)\psi^-_0(\epsilon_{ij}, \epsilon^p_{ij}) \tag{507.1}$$

where ψ^+_0 and ψ^-_0 are tensile and compressive parts of the elastic free energy (strain energy), that are defined as:

$$\psi^+_0(\bar{\sigma}_{ij}(\epsilon_{ij}, \epsilon^p_{ij})) = \frac{1}{2}\bar{\sigma}^+_{ij} D^{-1}_{ijkl} \bar{\sigma}_{kl} = \frac{1}{2}\bar{\sigma}^+_{ij}(\epsilon_{ij} - \epsilon^p_{ij}) \tag{507.2}$$

$$\psi^-_0(\bar{\sigma}_{ij}(\epsilon_{ij}, \epsilon^p_{ij})) = \frac{1}{2}\bar{\sigma}^-_{ij} D^{-1}_{ijkl} \bar{\sigma}_{kl} = \frac{1}{2}\bar{\sigma}^-_{ij}(\epsilon_{ij} - \epsilon^p_{ij}) \tag{507.3}$$

The rate of strain energy can be calculated from:

$$\dot{E}_S = \dot{\psi}^+_0 + \dot{\psi}^-_0 = \sigma_{ij}(\dot{\epsilon}_{ij} - \dot{\epsilon}^p_{ij}) \tag{507.4}$$

Taking the time derivative of Equation 507.1 gives the rate form of Helmholtz free energy:

$$\dot{\psi} = \frac{\partial \psi}{\partial \epsilon_{ij}}\dot{\epsilon}_{ij} + \frac{\partial \psi}{\partial \epsilon^p_{ij}}\dot{\epsilon}^p_{ij} + \frac{\partial \psi}{\partial d^+}\dot{d}^+ + \frac{\partial \psi}{\partial d^-}\dot{d}^- = (\sigma_{ij}\dot{\epsilon}_{ij} - \sigma_{ij}\dot{\epsilon}^p_{ij}) + (-\psi^+_0\dot{d}^+ - \psi^-_0\dot{d}^-) \tag{507.5}$$

where the first term is the rate of strain energy given by Equation 507.4.

For a decoupled material model, the Helmholtz free energy can be decomposed into elastic and plastic parts (Collins and Houlsby, 1997). Subtracting the elastic part of the free energy (strain energy) from Equation 507.5 gives the rate form of the plastic free energy:

$$\dot{E}_P = \dot{\psi} - \dot{E}_S = -\psi^+_0\dot{d}^+ - \psi^-_0\dot{d}^- = -\frac{1}{2}(\bar{\sigma}^+_{ij}\dot{d}^+ + \bar{\sigma}^-_{ij}\dot{d}^-)(\epsilon_{ij} - \epsilon^p_{ij}) \tag{507.6}$$

The plastic energy dissipation D_P is defined as the difference between plastic work and plastic free energy (Farren and Taylor, 1925; Taylor and Quinney, 1934). The rate of plastic dissipation can be expressed as:

$$\dot{D}_P = \sigma_{ij}\dot{\epsilon}^p_{ij} - \dot{E}_P = \sigma_{ij}\dot{\epsilon}^p_{ij} + \frac{1}{2}(\bar{\sigma}^+_{ij}\dot{d}^+ + \bar{\sigma}^-_{ij}\dot{d}^-)(\epsilon_{ij} - \epsilon^p_{ij}) \tag{507.7}$$

It can be proven that the rate of plastic dissipation given by Equation 507.7 is always non-negative. This means that the concrete material model used in this study satisfies the Clausius-Duhem inequality, which represents a form of the second law of thermodynamics.

Combining Equation 507.4, 507.6, and 507.7, the rate form of energy balance in this concrete material model is achieved:

$$\dot{E}_S + \dot{E}_P + \dot{D}_P = \sigma_{ij}\dot{\epsilon}_{ij} \tag{507.8}$$

507.2.5 Uniaxial Steel Material

The energy computation procedure for the uniaxial steel model is shown in Figure 507.4. Note that the only difference between the monotonic loading branch (Figure 507.4(a)) and the cyclic loading branch (Figure 507.4(b)) is that the strain reversal point c is at the origin o in the monotonic case. So the following explanation of the proposed energy computation method applies to both monotonic and cyclic loading scenarios.

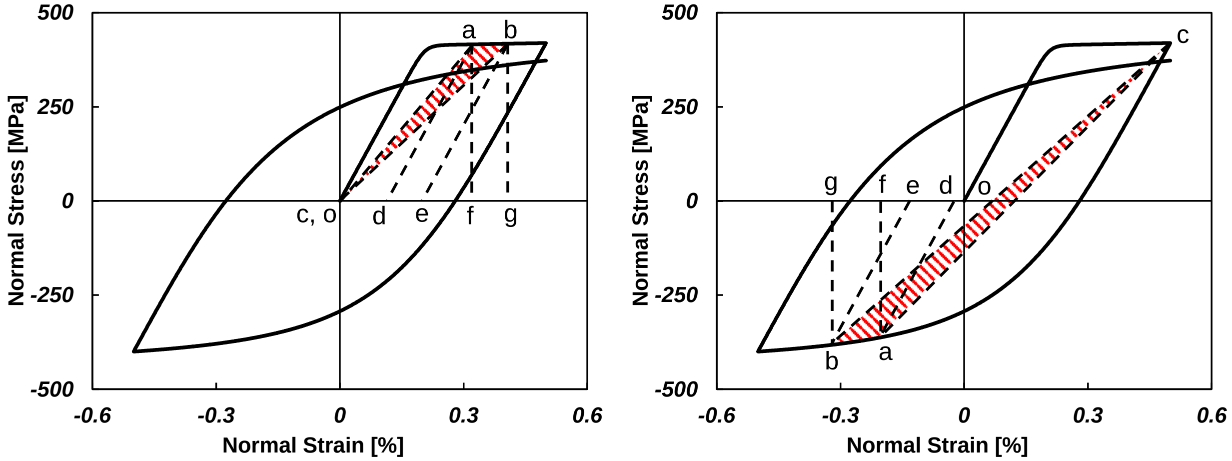


Figure 507.4: Energy computation of uniaxial steel fiber: (a) Monotonic loading branch; (b) Cyclic loading branch.

Firstly, the elastic strain energy density E_S is defined in accordance with the classic assumption that it is only a function of current stress state of the material, which yields:

$$E_S = E_S(\sigma) = \frac{1}{2E_0}\sigma^2 \tag{507.9}$$

where E_0 is the initial stiffness of the material.

Graphically, the elastic strain energy density of the material shown in Figure 507.4 at states a and b are the triangular areas afd and bge. Then the incremental form of Equation 507.9 is simply:

$$dE_S = \frac{1}{E_0}\sigma d\sigma \tag{507.10}$$

Next, the incremental plastic dissipation density D_P from state a to b is assumed to be the triangular area abc:

$$dD_P = \frac{1}{2}[(\sigma - \sigma_r)d\epsilon - (\epsilon - \epsilon_r)d\sigma] \tag{507.11}$$

This assumption ensures that the incremental plastic dissipation is non-negative, and that ensures that the second law of thermodynamics is satisfied.

For general case where the material does exhibit cyclic softening, plastic free energy density E_P is graphically described as the areas $adoc$ and $beoc$ at states a and b, respectively. The formulation for plastic free energy density E_P representing this assumption is given by:

$$E_P = \frac{1}{2}\left[\sigma\left(\epsilon - \frac{\sigma}{E_0} - \epsilon_r\right) + \sigma_r\epsilon\right] \tag{507.12}$$

The incremental form of Equation 507.12 is:

$$dE_P = \frac{1}{2}\left[(\sigma + \sigma_r)\, d\epsilon + \left(\epsilon - \frac{1}{E_0}\sigma - \epsilon_r\right) d\sigma\right] \tag{507.13}$$

Adding Equation 507.10, 507.11, and 507.13, the incremental form of energy balance is achieved:

$$dE_S + dE_P + dD_P = \sigma d\epsilon \tag{507.14}$$

where the increment of three energy components add up to the increment of stress power during any loading step.

507.2.6 Modeling and Simulation Results

In this chapter, the simulation results are presented and compared with corresponding experimental results. Three sets of FEM simulations (Reg A, Reg B, and ASR A1) using the Real-ESSI Simulator system (Jeremić et al., 1988-2025) (`http://real-essi.info/`) are performed. In the following sections, the force–displacement responses of these three sets of simulations are shown and discussed. The strain and stress distributions, that share similar pattern in all three simulations, are also presented. In order to investigate the level of damage in the shear wall, the evolution of concrete damage index as well as the plastic dissipation density are plotted and discussed.

507.2.6.1 Force–Displacement Response

Figures 507.5 and 507.6 present the force–displacement responses of the simulation and experiment results for regular concrete (Reg A and Reg B) and ASR concrete.

The ESSI simulation curves show good matching of experimental results. The differences in the envelopes of the cyclic loading curves, between the numerical and experimental results are within 10%.

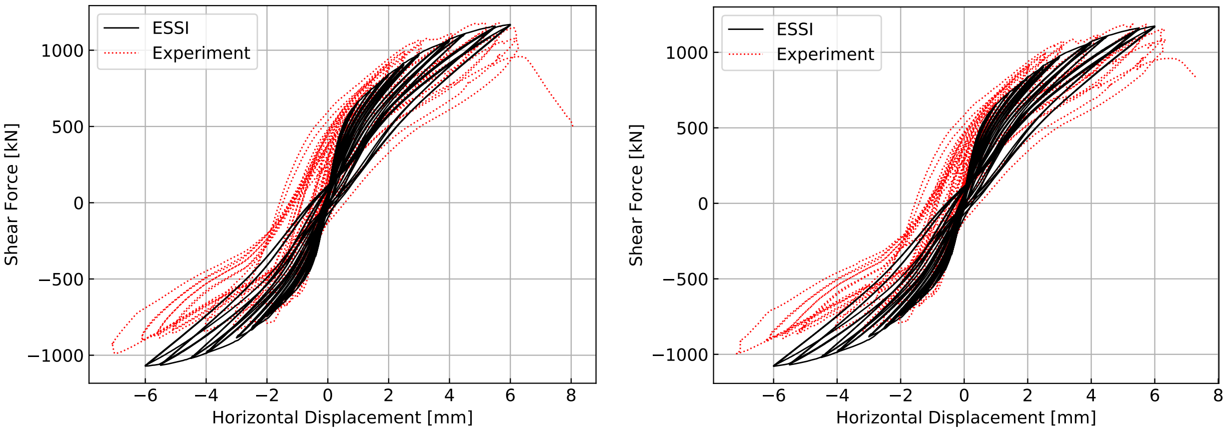


Figure 507.5: Force–Displacement responses of regular concrete tests: (a) Reg A; (b) Reg B.

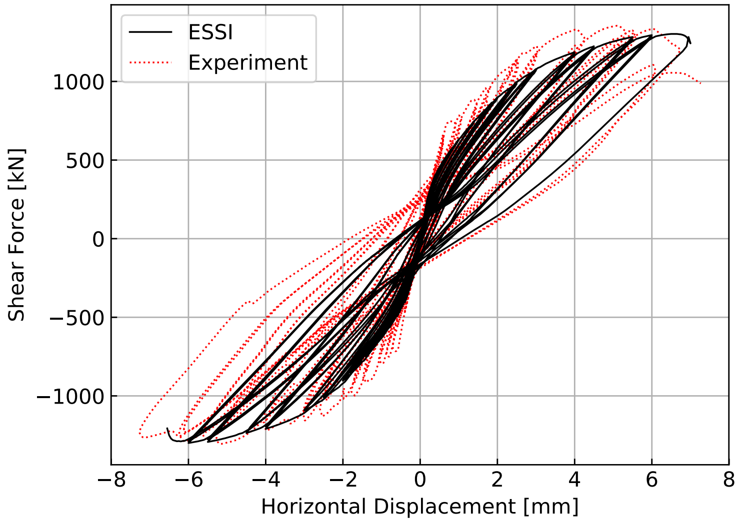


Figure 507.6: Force–Displacement response of ASR concrete test (ASR A1).

The shear strengths and failure loads/displacements given by ESSI simulations match well with the values determined by physical experiments. Note that this particular case with ASR concrete has a much larger unloading-reloading cyclic area, which means that ASR concrete has the capability of dissipating more input energy. It is important to note that this conclusion dos not hold for other ASR concrete walls that were tested. This might indicate that for some structures with the ASR concrete, it is possible to dissipate more seismic energy if the structure is under earthquake cyclic loading. On the other hand, for some other structure with the ASR concrete, such conclusion might not hold as other test data suggests reduction of seismic energy dissipation capacity. This leads to the conclusion that variability of ASR concrete quality and material behavior can be significant.

507.2.6.2 Strain and Stress Distribution

Due to the fact that the stress, strain, and damage patterns in all three cases are very similar, only plots for the Reg A case is presented and discussed in the following sections.

Figure 507.7 shows the distribution of displacement components in the shear wall model at $u_y = 6$ mm. The vertical displacement distribution is almost symmetric. The left part of the shear wall is compressed down while the right part is extended up. It is noted that the upper support beam is thus not remaining horizontal, rather, it is applying bending load to the shear wall. The horizontal displacement in the top beam slab is almost uniform.

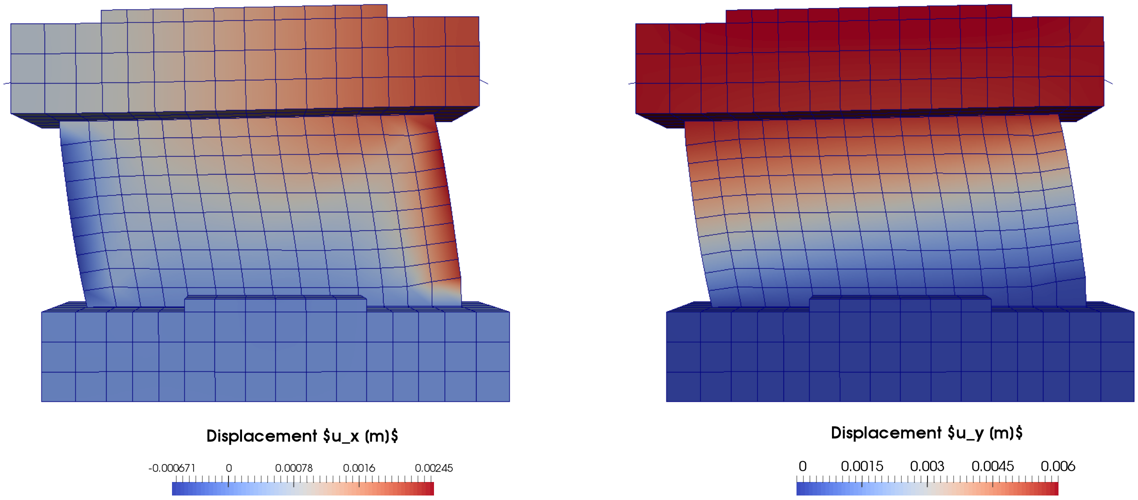


Figure 507.7: Distribution of displacement components in the shear wall model at $u_y = 6$ mm: (a) Vertical displacement u_x; (b) Horizontal displacement u_y.

Figure 507.8 shows the distribution of strain components in the shear wall model at $u_y = 6$ mm. The dominant components of the strains are the vertical normal strain ϵ_{xx} due to the vertical loading, and the shear strain ϵ_{xy} due to the horizontal loading. As can be seen in the distribution of ϵ_{xx}, the maximum tensile strain is much larger than the maximum compressive strain. Tensile strain actually means that there is a crack that opened at the location. This is expected since the compressive strength of concrete is higher than the tensile strength.

A 45^o shear zone can be observed in the shear wall, with significant amounts of tensile and shear strains. In addition, large tensile strains are developing around the top-left and bottom-right corners. As will be shown in the following section, the concrete in these areas is significantly damaged during the test.

Figure 507.9 shows the distribution of stress components in the shear wall model at $u_y = 6$ mm. To

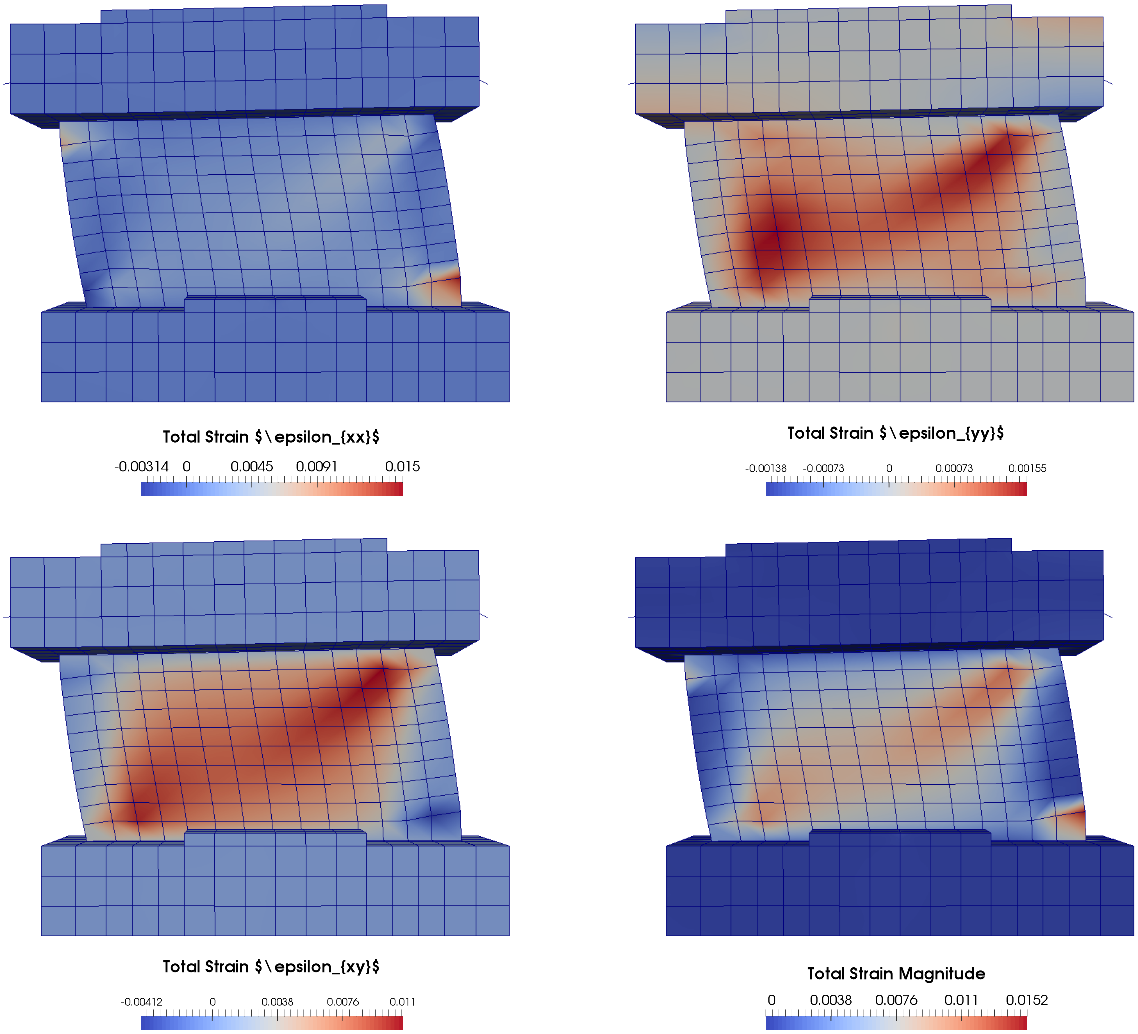


Figure 507.8: Distribution of strain components in the shear wall model at $u_y = 6$ mm: (a) Vertical normal strain ϵ_{xx}; (b) Horizontal normal strain ϵ_{yy}; (c) Shear strain ϵ_{xy}; (d) Strain magnitude $|\epsilon|$.

make the plots more clear, only stresses in the shear wall is plotted. Again, a 45^o shear zone can be observed, that is consistent with the pattern in the strain plots.

Large compressive normal stresses are developed around the top-right and bottom-left corners, while large tensile normal stresses are observed around the top-left and bottom-right corners. The shear stresses at all four corners are significant.

Stress/force and strain/deformation distributions are not enough to directly quantify the level of damage in structures, especially in the case of cyclic loading. In order to analyze damage conditions, the concrete damage index and plastic energy dissipation are calculated and plotted, as shown in the next section.

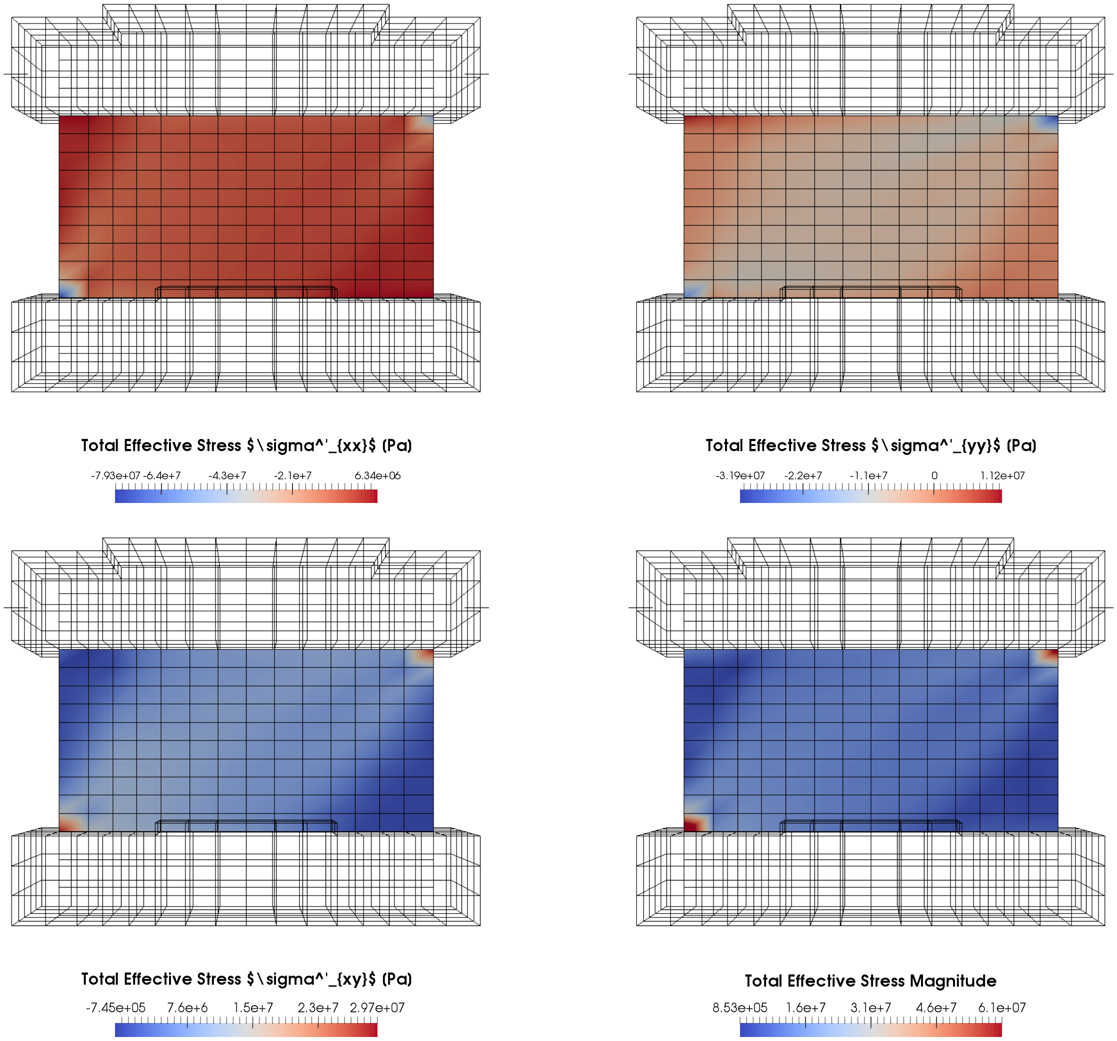


Figure 507.9: Distribution of stress components in the shear wall model at $u_y = 6$ mm: (a) Vertical normal stress σ_{xx}; (b) Horizontal normal stress σ_{yy}; (c) Shear stress σ_{xy}; (d) Stress magnitude $|\sigma|$.

507.2.6.3 Concrete Damage and Energy Dissipation

Figure 507.10 shows the evolution of shear wall concrete damage index d^+, defined in Equation 104.622, at different levels of deformation during cyclic loading. Only tensile damage index is presented in Figure 507.10 as no compressive damage occurs in the wall. It is noted that, according to damage evolution from Figure 507.10, three damage/cracks zones are observed in the wall for load/displacement cycles of $u_y = \pm 1.4$mm. Two 45^o tensile/shear zones with opposite directions as well as a tensile failure zone along the bottom of the wall are developing. Note that the flanges experience damage at the bottom due to tension, which indicates that wall-flange system experiences bending, and not pure shear as intended.

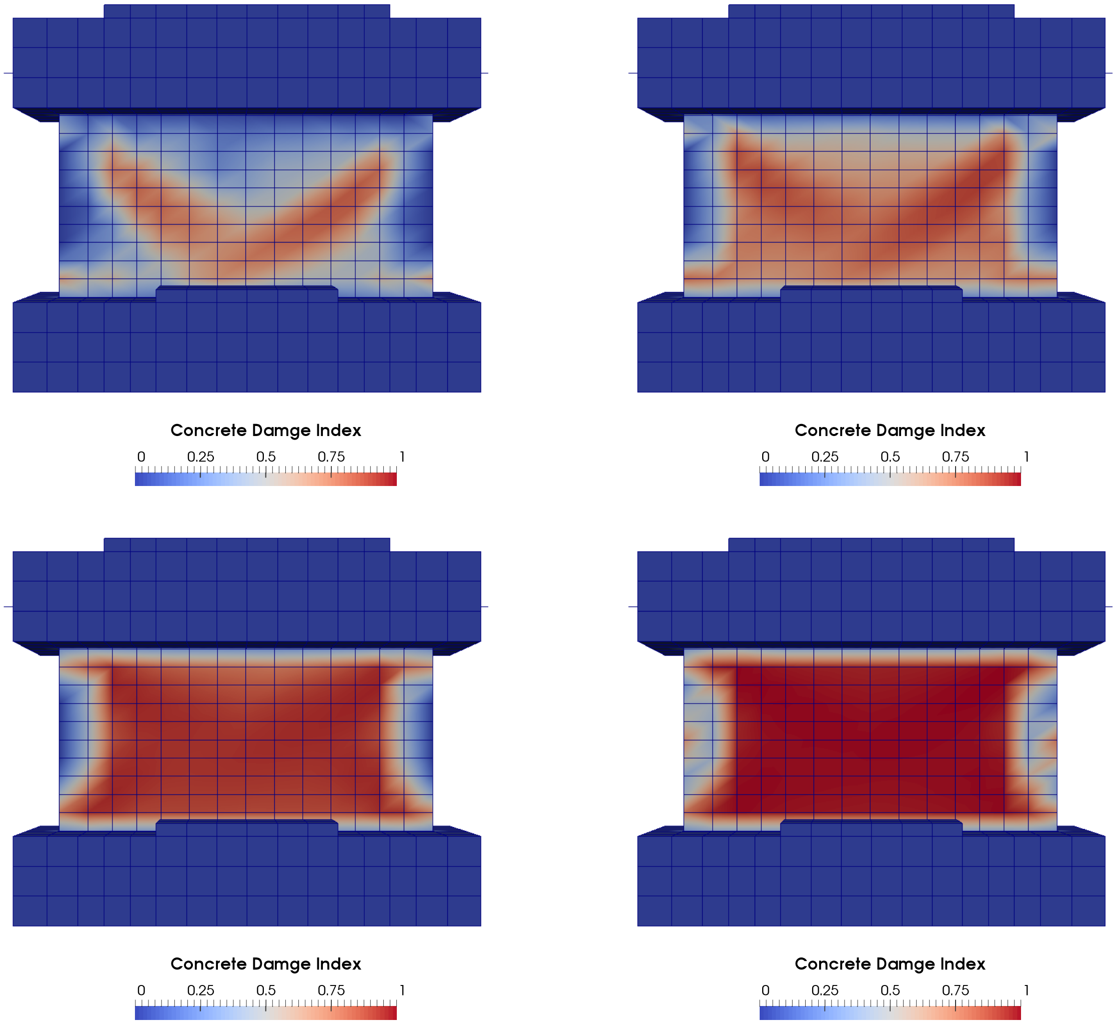


Figure 507.10: Evolution of concrete damage index in the shear wall model for different levels of cyclic loading: (a) At $u_y = \pm 1.4$mm; (b) At $u_y = \pm 1.8$mm; (c) At $u_y = \pm 3.0$mm; (d) At $u_y = \pm 5.0$mm.

As the loading/deformation cycles increase, the damage zones extends and the level of damage increases. At the end, the entire wall is damaged. In reality, due to variability of material properties, some wall regions will experience localization of deformation, as observed in experiments. Modeling of localization of deformation suffers from mesh dependency effects. It is noted that localization of deformation zones, as seen in Figures 507.10 and 507.11 are about the size of single finite element. It is likely that upon mesh refinement these zones would change (Lu et al., 2009), however this type of sensitivity study was not done here due to time constraints. We also note that a more sound approach to modeling localized deformation behavior would be through the use of Cosserat continuum (Cosserat, 1909; de Borst, 1987). Cosserat continuum finite element and micropolar elastic and elastic-plastic models are available within the Real-ESSI Simulator, however calibration of material parameters for those models from current test results is beyond the scope and time frame of this study, and was not done.

Other than damage index, plastic energy dissipation density can also be used to illustrate damage in structures. The main difference between damage index and plastic energy dissipation is that the plastic energy dissipation increases even after an element is completely damaged, that is when $d^+ = 1.0$. [2]

Figure 507.11 shows the evolution of plastic energy dissipation density in the shear wall model at different stages of loading. At the beginning of the test, Figure 507.11(a), the distribution of plastic energy dissipation density is very similar to the distribution of a damage index (d^+). However, as the load cycles progress and displacement increase, Figure 507.11(b), significant amount of energy is dissipated at bottom corners of both flanges. Most of that plastic dissipation happens due to opening of gaps, tensile cracking, and some shear. Plastic dissipation due to tensile cracking can happen only once, as cracks will not "heal", hence there is a redistribution of loads and deformation within flange-wall system. Once flange cannot dissipate any more energy, only wall is left to pick up loads and dissipate energy due to tension and shear. It is noted again that there was no observed development of compression damage, hence there is no plastic dissipation in compressed concrete. Toward the end of simulation, Figure 507.11(c), a failure zone that is X shaped, with distinct 45^o tensile plastic dissipation zones in the wall, and the plastic dissipation zone at the corners, bottom and top of flanges, is observed. At the end of the simulation, Figure 507.11(d), a failure zone due to tension and shear, encompasses flange corners and most of the wall.

Figure 507.12 shows animation of plastic dissipation development in the wall/membrane.

---

[2] Plastic dissipation is also very important for following seismic energy as it propagates through the soil structure system during earthquakes (Sinha et al., 2017; Wang et al., 2017). If seismic energy, and plastic dissipation can be accurately followed and its path even directed, during earthquake soil structure interaction (ESSI), soil structure systems can be optimized for safety and economy.

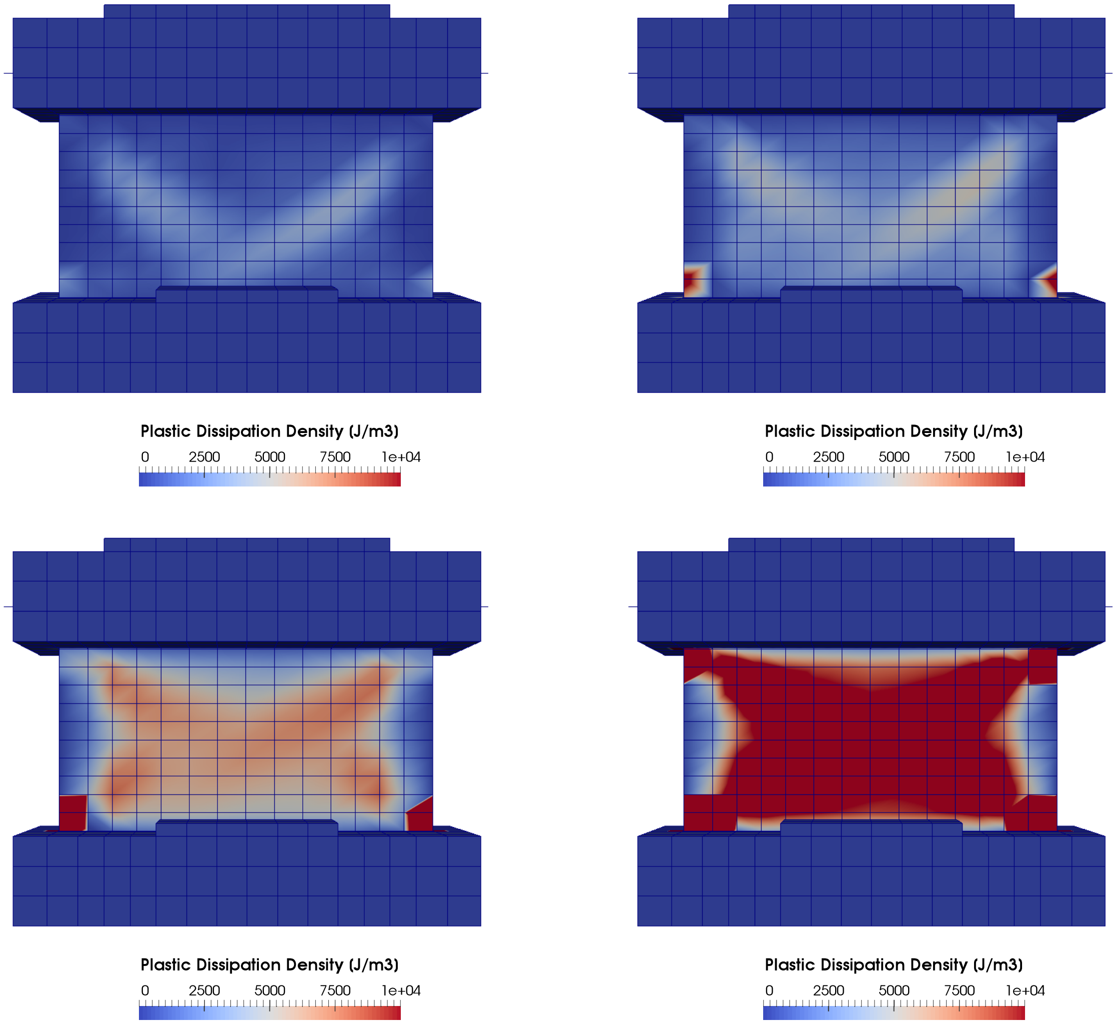


Figure 507.11: Evolution of plastic energy dissipation density in the shear wall model: (a) At $u_y = \pm 1.4\text{mm}$; (b) At $u_y = \pm 1.8\text{mm}$; (c) At $u_y = \pm 3.0\text{mm}$; (d) At $u_y = \pm 5.0\text{mm}$.

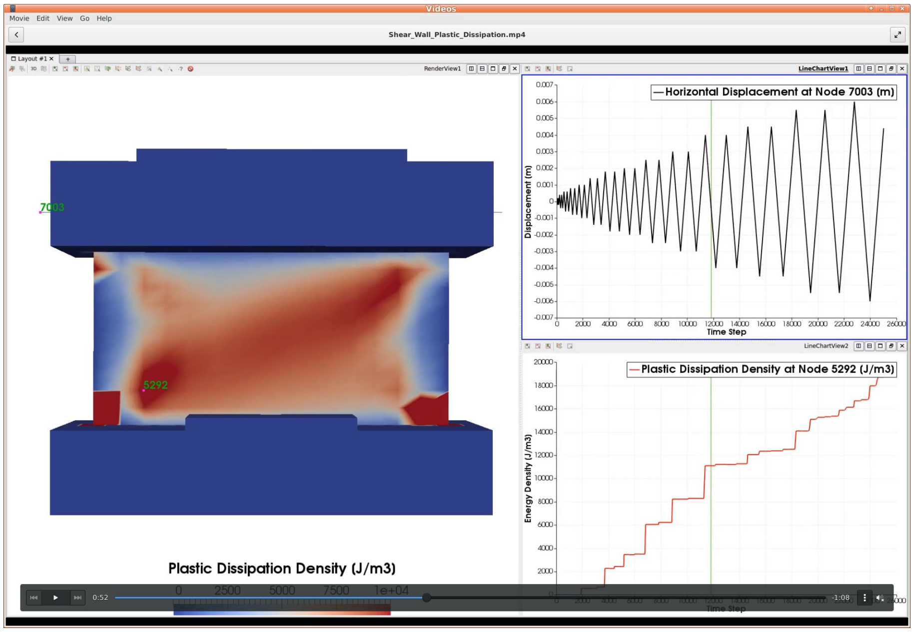


Figure 507.12: Animation of plastic dissipation in the wall/membrane.

Chapter 508

ESSI for Concrete Dams

(2019-2020-2021-)

(In collaboration with Prof. Han Yang and Dr. Hexiang Wang)

508.1 Chapter Summary and Highlights

508.2 Pine Flat Dam

Chapter 509

ESSI for Buildings

(2018-2019-2020-2021-)

(In collaboration with Prof. Han Yang, and Dr. Hexiang Wang)

509.1 2D Frame with Energy Dissipation

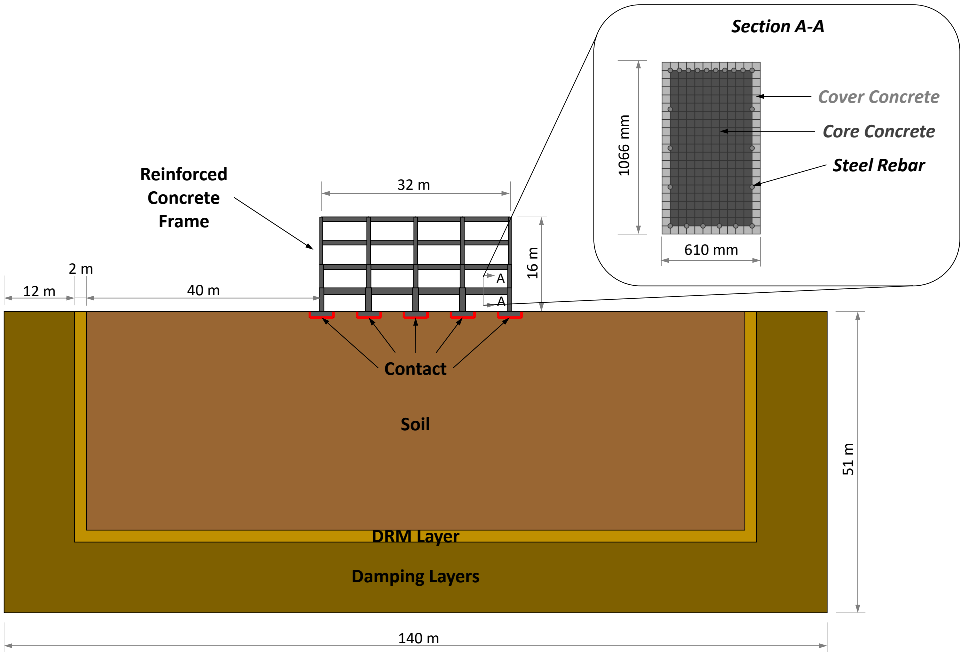


Figure 509.1: 2D frame model.

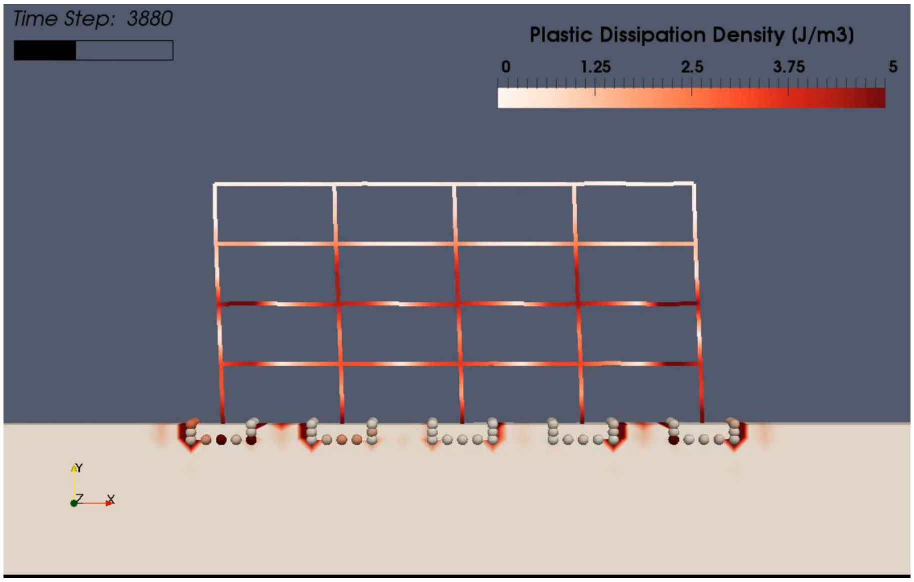


Figure 509.2: Energy Dissipation for a frame on spread foundations.

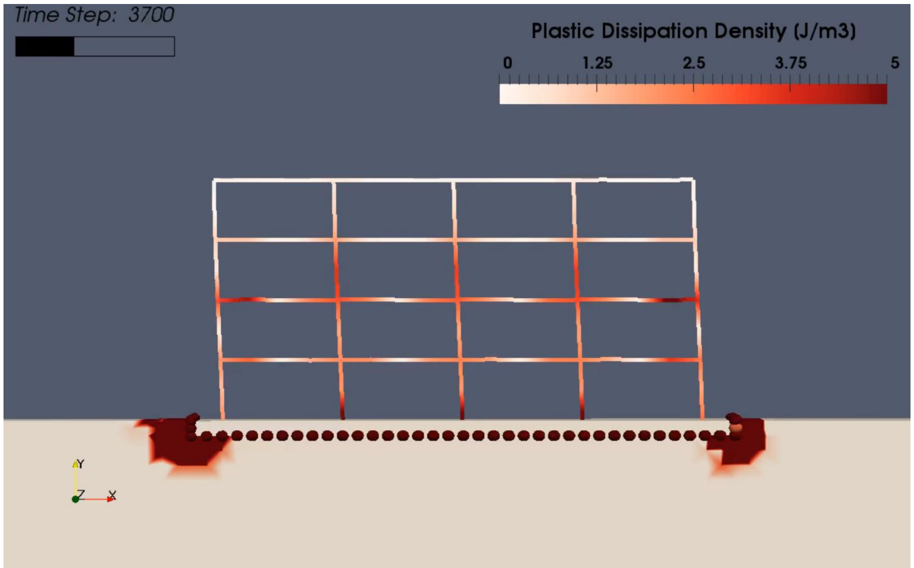


Figure 509.3: Energy Dissipation for a frame on slab foundation.

Finite element model, input files for the Real-ESSI simulator for Concrete Frame model are available:

- All model files, one by one,
- Model archive (tar xz)

For uncompressing and un-taring the model archive file, please use

```
tar -xvf Concrete_frame_PEER_model.txz.
```

509.2 Ventura Hotel

509.2.1 Finite Element Model

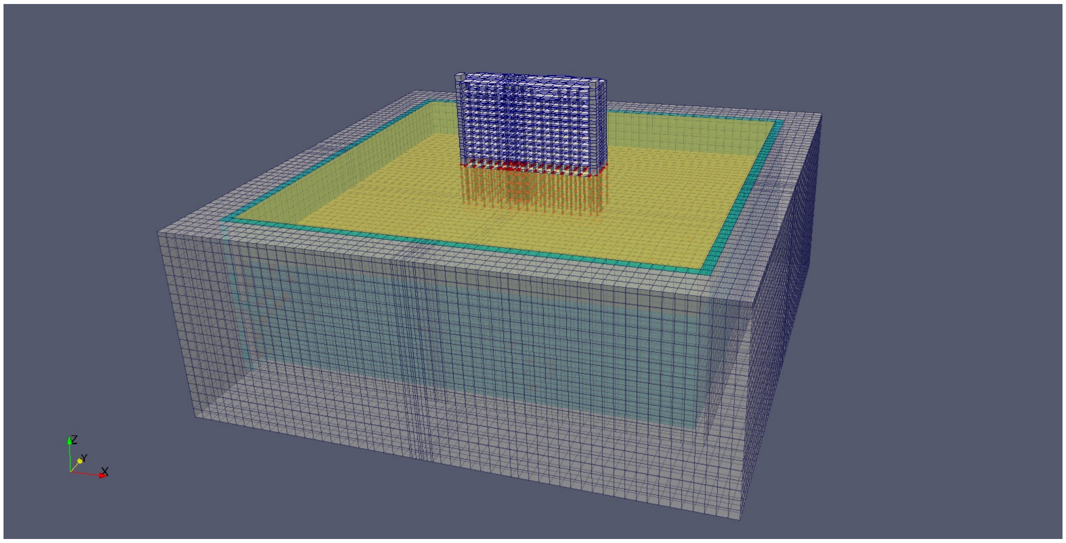


Figure 509.4: Ventura hotel model.

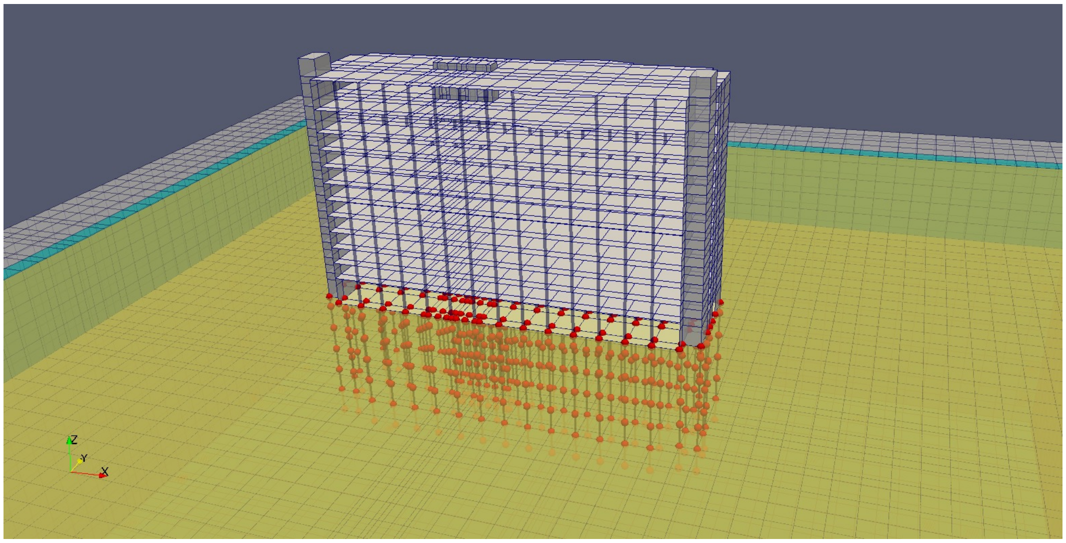


Figure 509.5: Ventura hotel model.

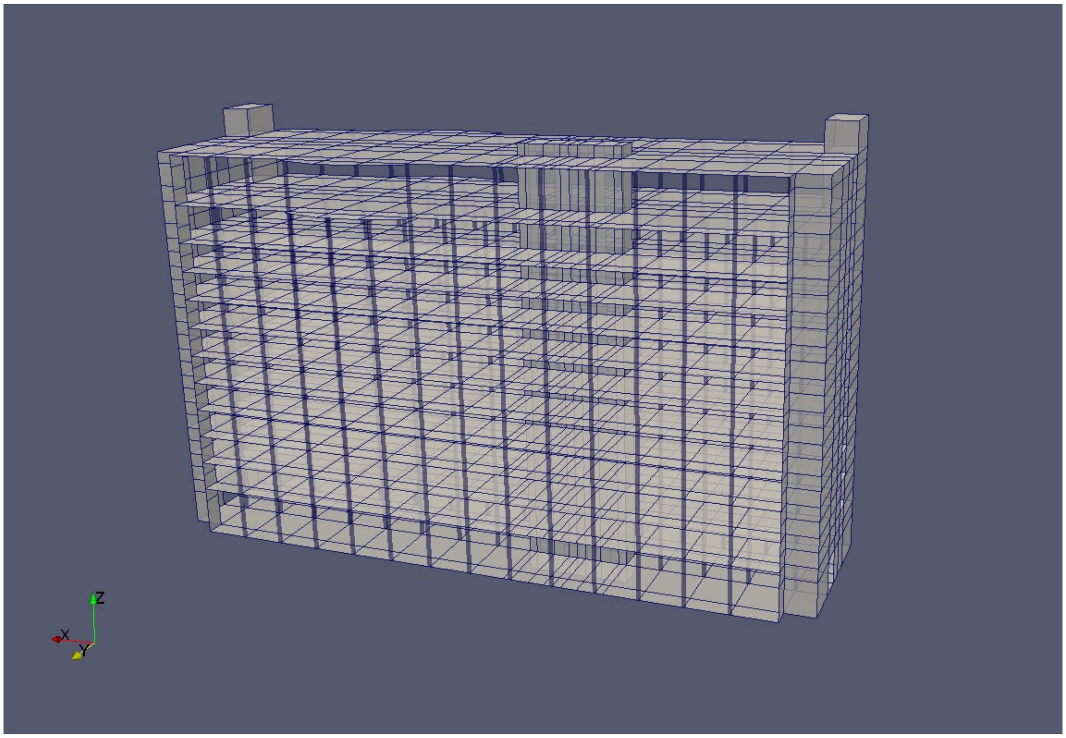


Figure 509.6: Ventura hotel model.

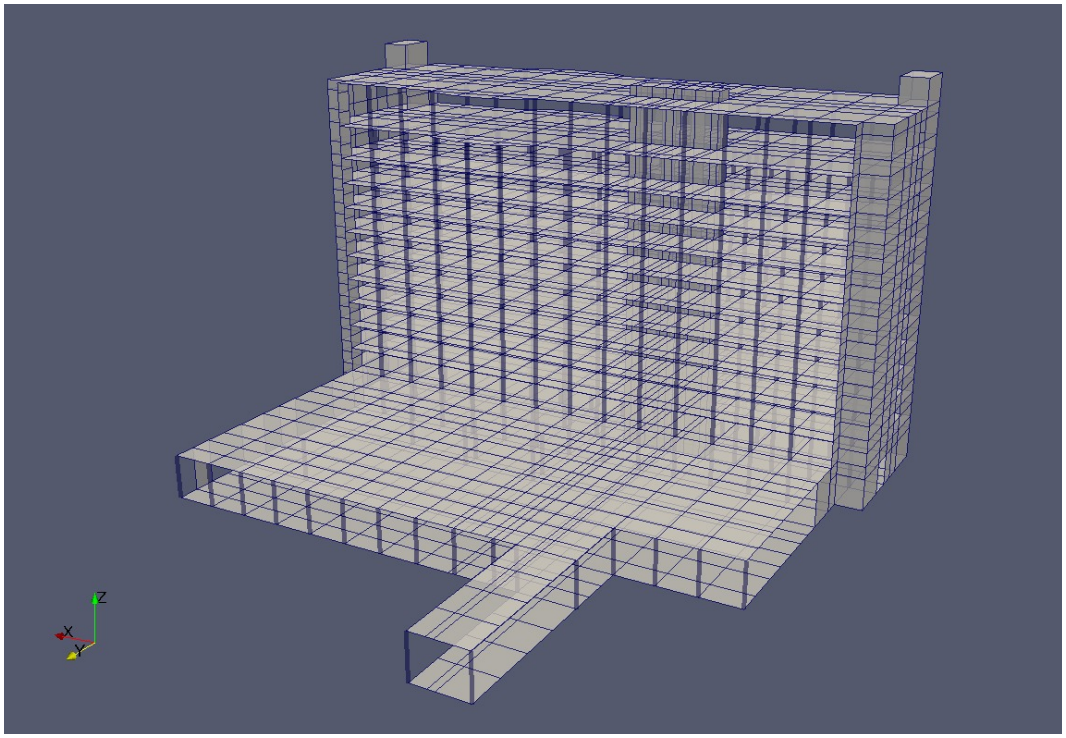


Figure 509.7: Ventura hotel model.

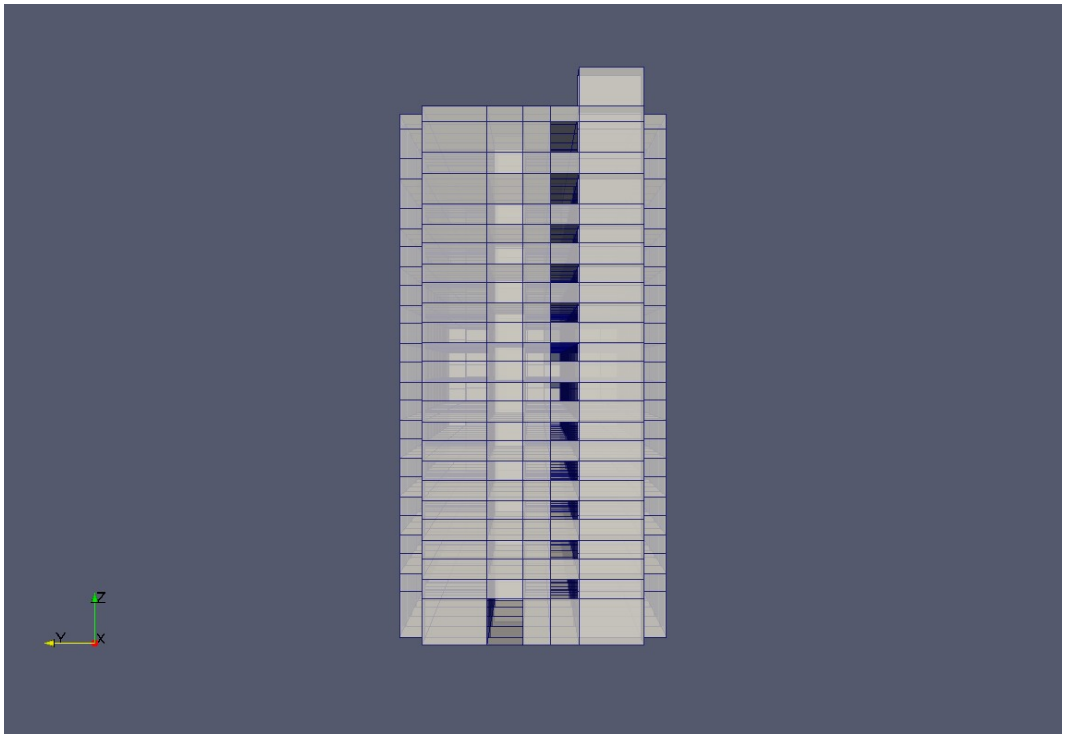


Figure 509.8: Ventura hotel model, view in X direction.

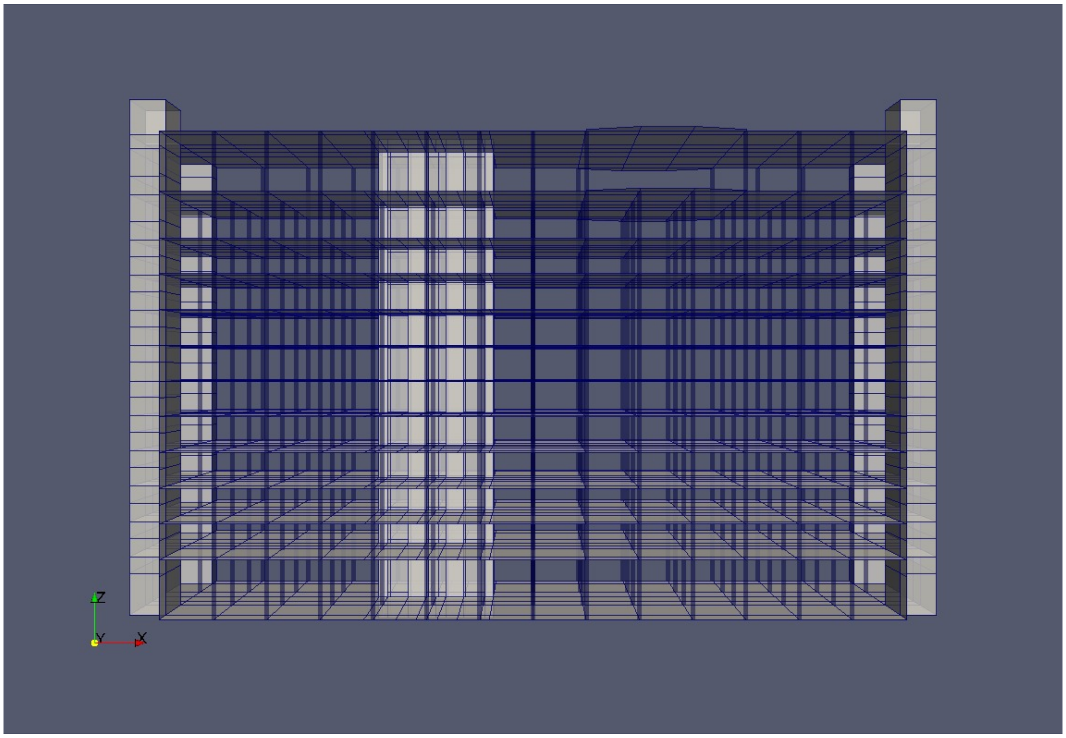


Figure 509.9: Ventura hotel model, view in Y direction.

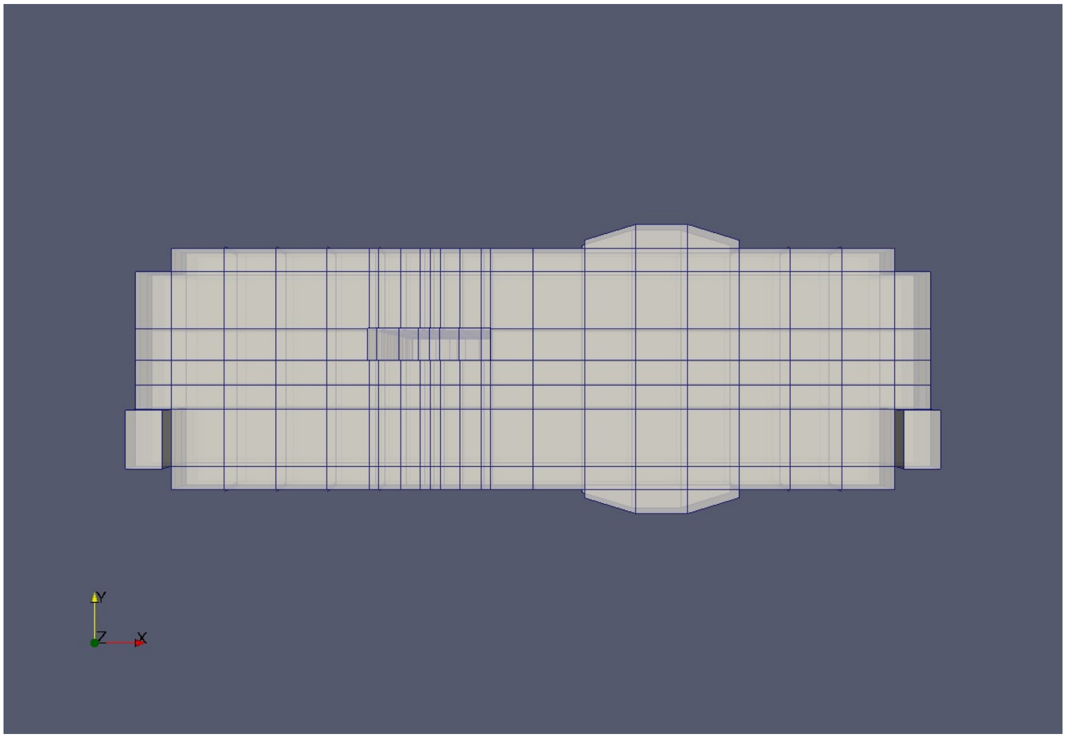


Figure 509.10: Ventura hotel model, view in Z direction.

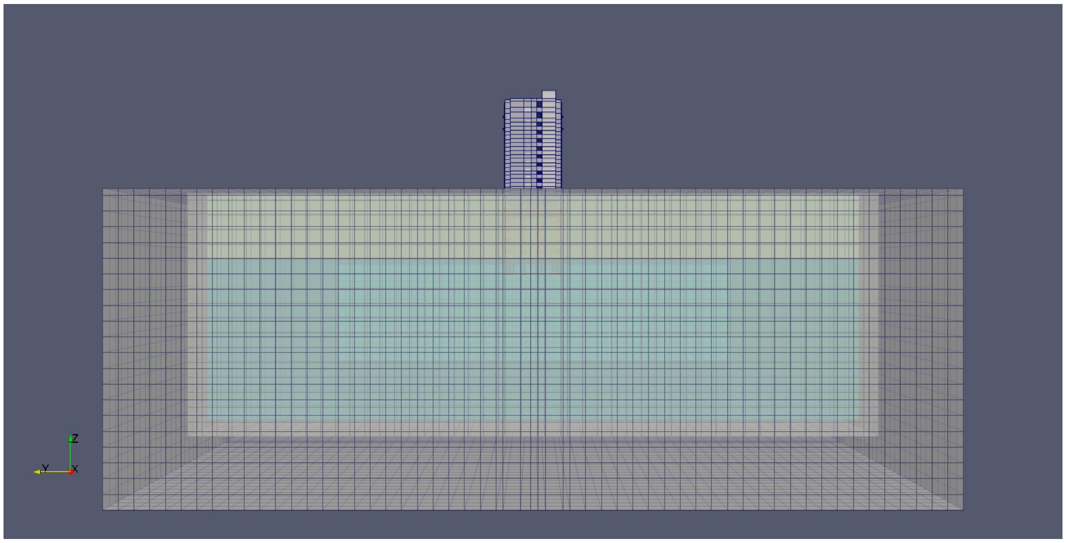


Figure 509.11: Ventura hotel model, view along X direction.

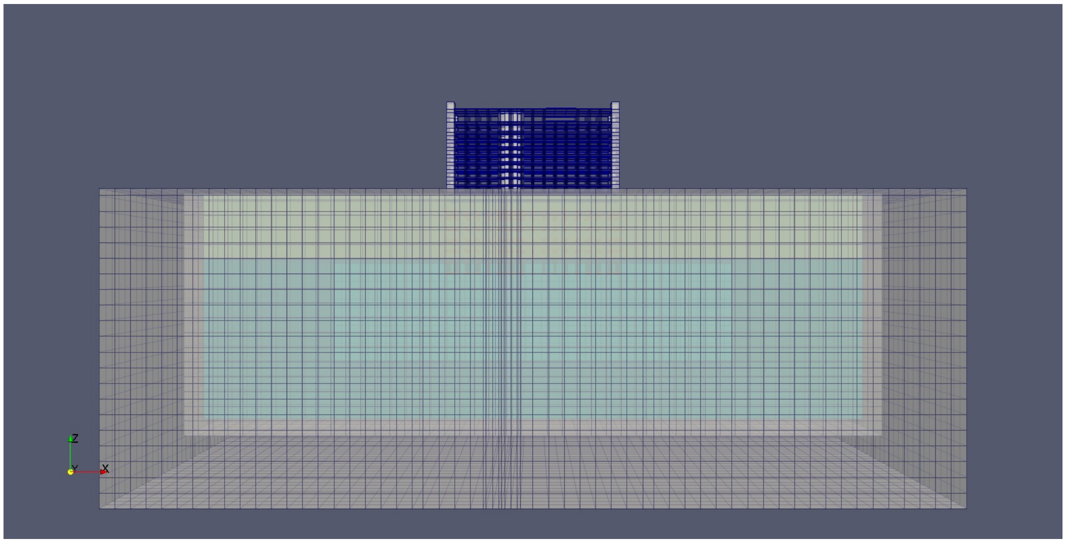


Figure 509.12: Ventura hotel model, view along Y direction.

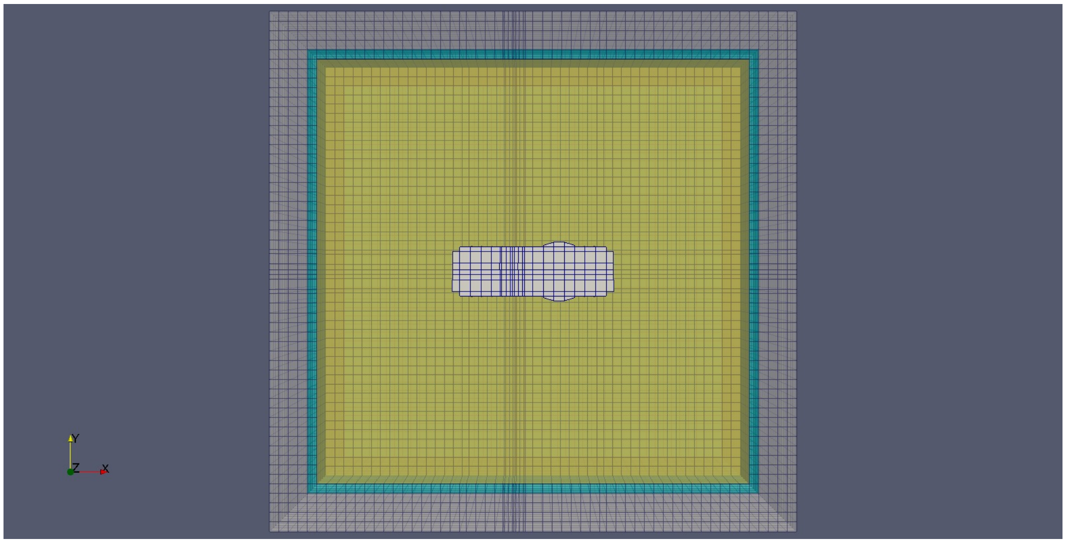


Figure 509.13: Ventura hotel model, view along Z direction.

Finite element model, input file for the Real-ESSI simulator for Ventura hotel model with and without SSI are available:

- Full SSI model, (7.8MB)
- Structure only, non-SSI model, (84kB)

NOTE: while full SSI model is only 11MB in compressed format, using `xz` compressor, fully uncompressed model files are over 3.7GB large, that is 3,770MB, or 3,868,356Kb.

For uncompressing and un-taring, please use

`tar -xvf _Ventura_hotel_Full_SSI_Model_.tar.xz`,

and

`tar -xvf _Ventura_hotel_non_SSI_Model_.tar.xz`.

509.3 Loma Linda Hospital

509.3.1 Finite Element Model

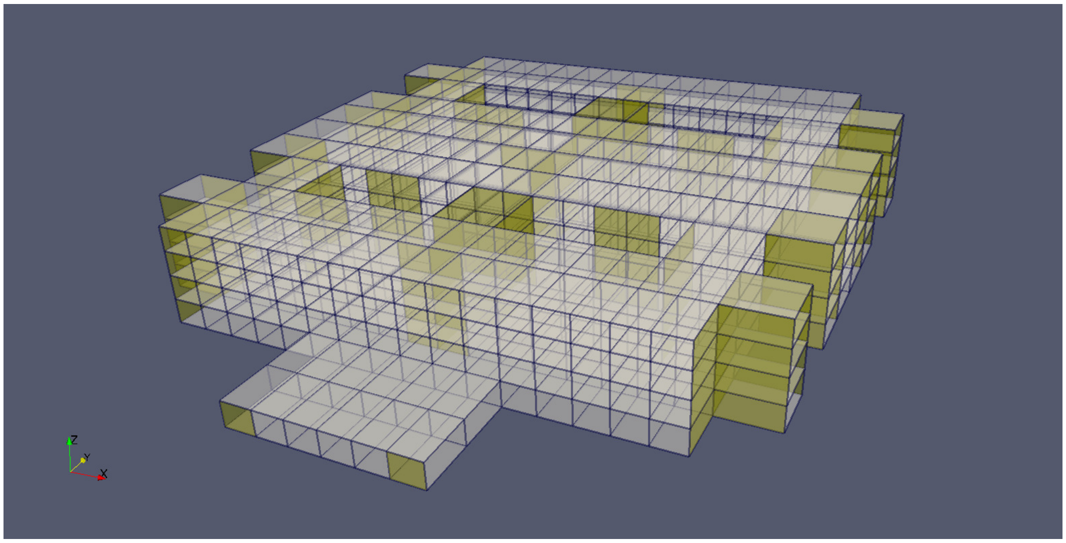


Figure 509.14: Loma Linda hospital 3D view.

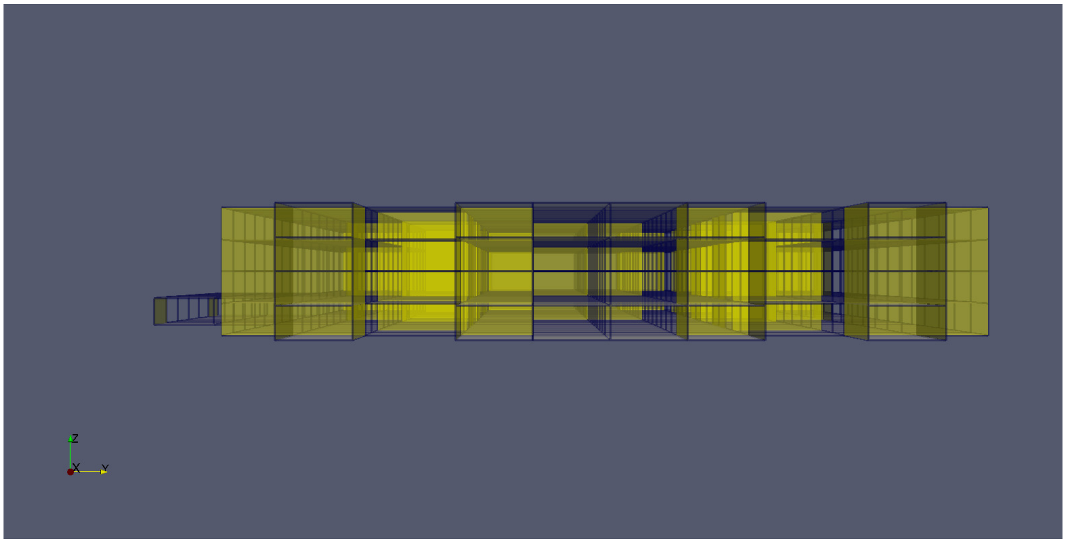


Figure 509.15: Loma Linda hospital view in X direction.

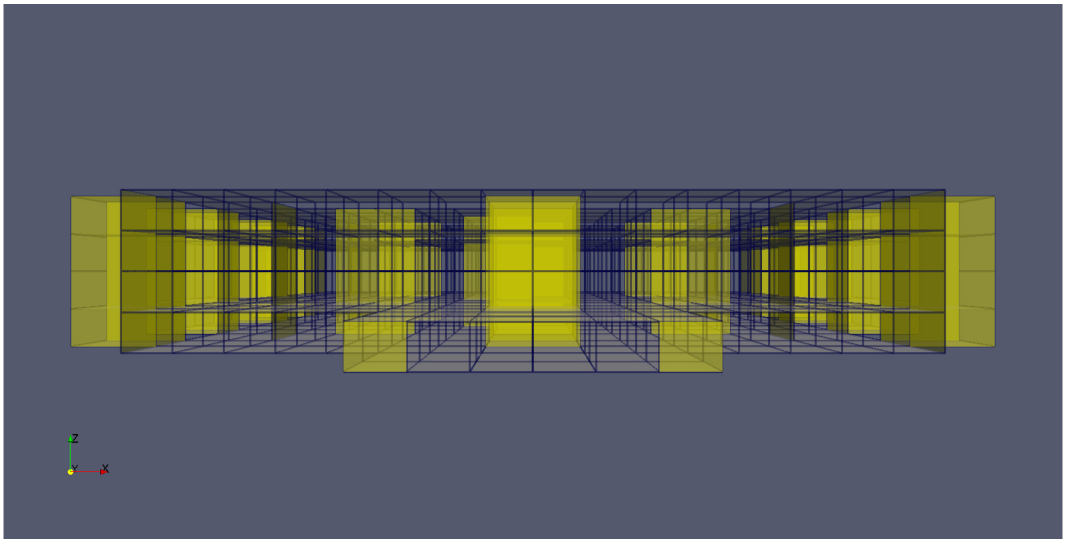


Figure 509.16: Loma Linda hospital view in Y direction.

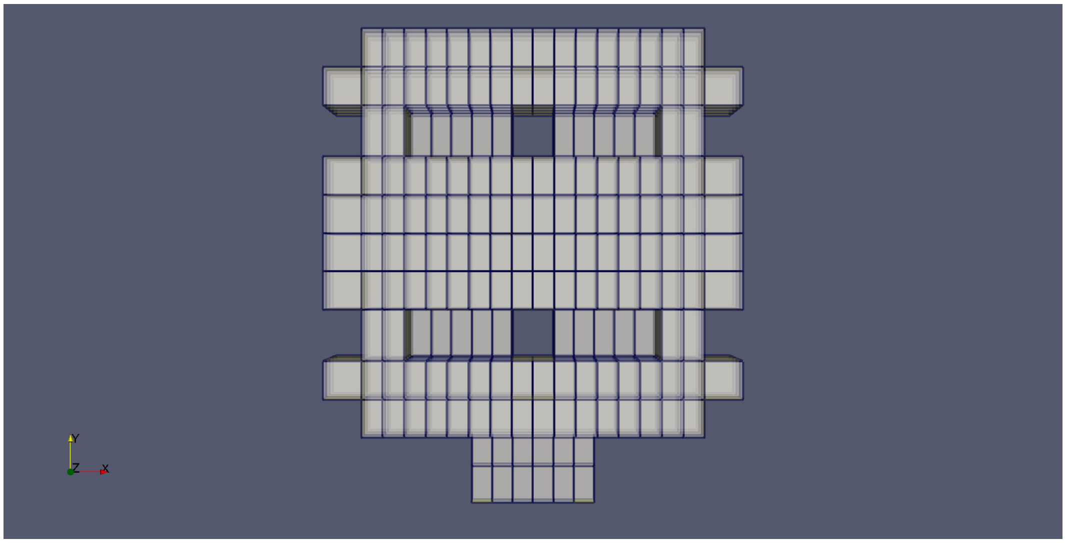


Figure 509.17: Loma Linda hospital view in Z direction.

509.4 ASCE-7 Model, Low, Steel Building

509.4.1 Finite Element Model

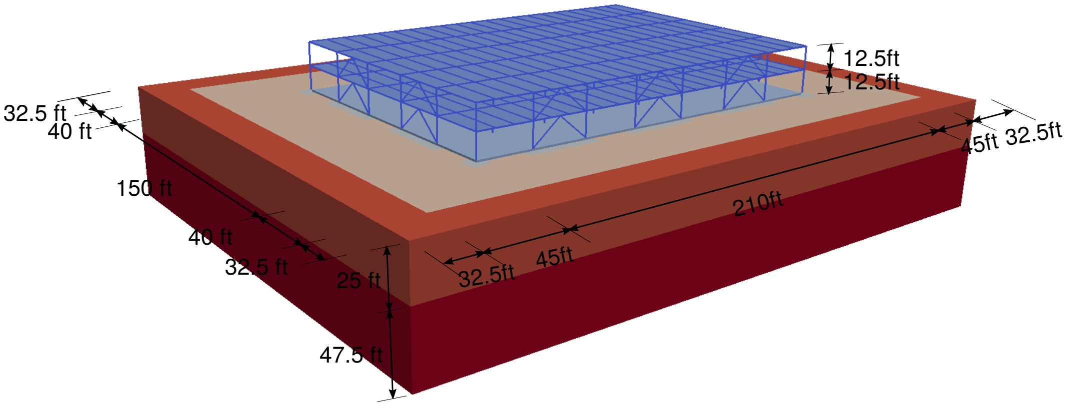


Figure 509.18: Low steel building model, 3D view.

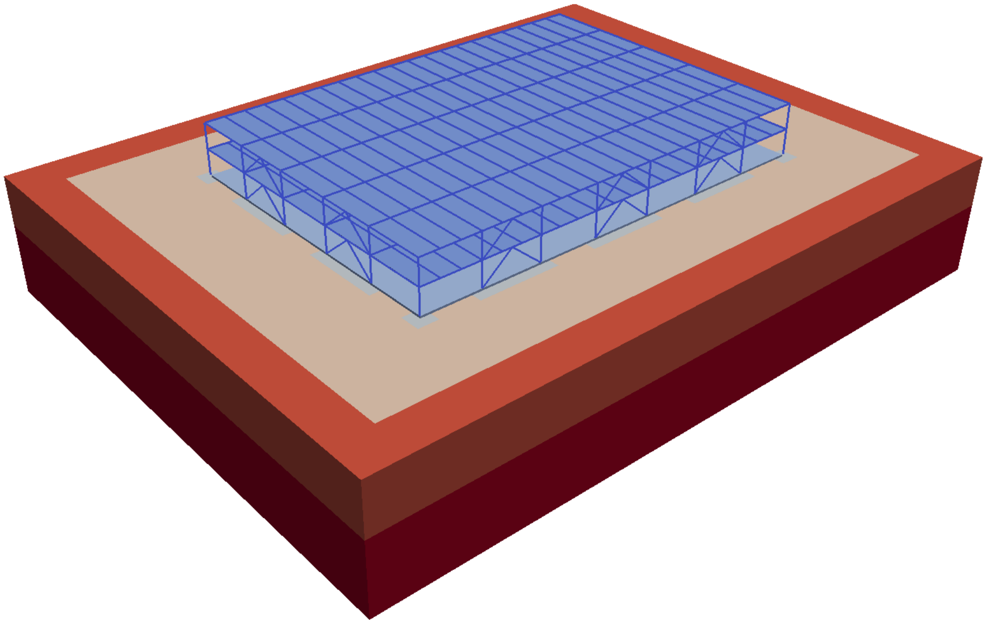


Figure 509.19: Low steel building model, 3D view.

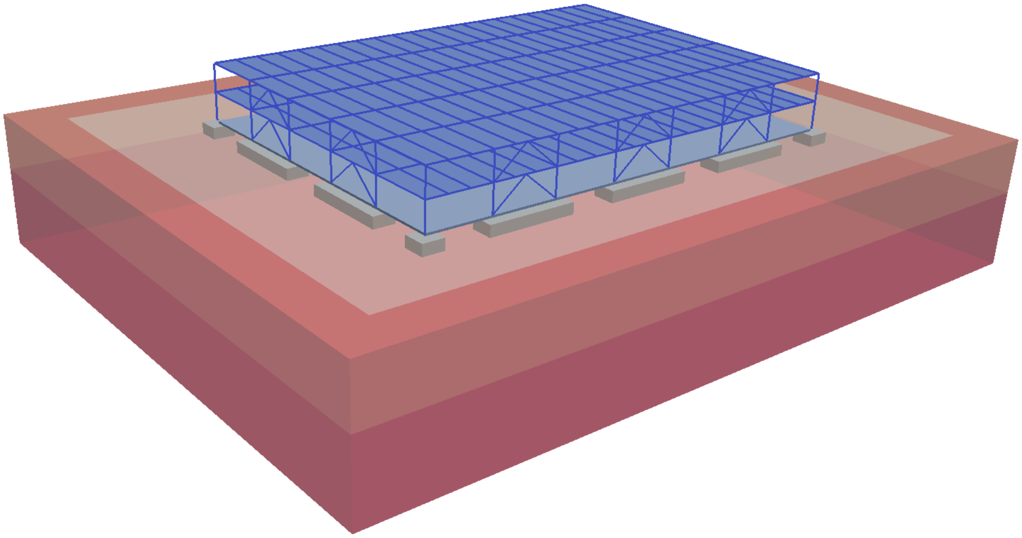


Figure 509.20: Low steel building model, 3D view.

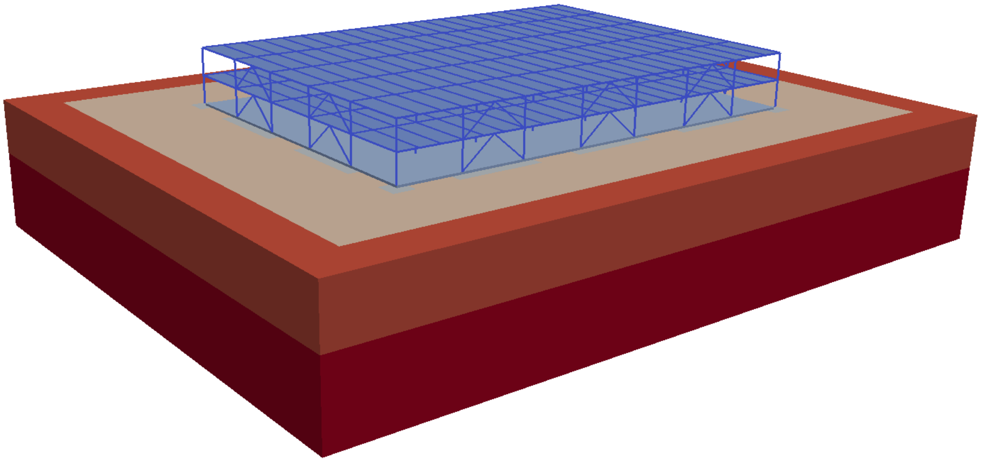


Figure 509.21: Low steel building model, 3D view.

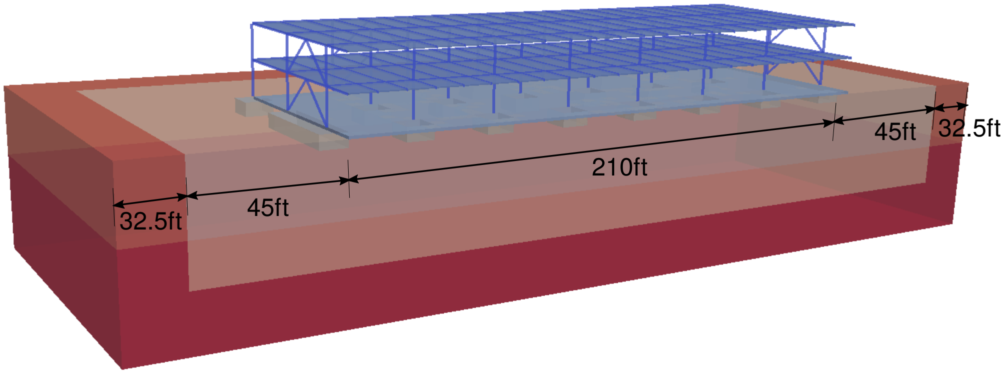


Figure 509.22: Low steel building model, 3D view.

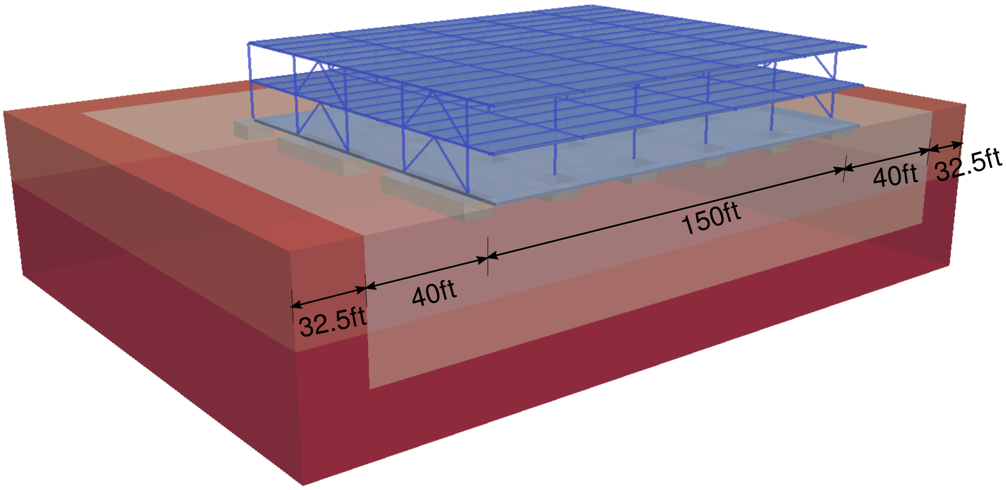


Figure 509.23: Low steel building model, 3D view.

Finite element model, input file for the Real-ESSI simulator for full SSI ASCE-7 model, low, steel building are available HERE.

NOTE: This model is 1.1GB (1,102MB, 1,127,928Kb) in size even when compressed, due to large file describing incoherent motions. When uncompressed
(use
`tar -xvf _ASCE-7_low_steel_building_Model_.tar.xz`), models files grow to 3.0GB, that is 2,977MB, or 3,047,532Kb, so please be aware of disk space requirements for the model alone.

509.5 ASCE-7 Model, High, Concrete Building

509.5.1 Finite Element Model

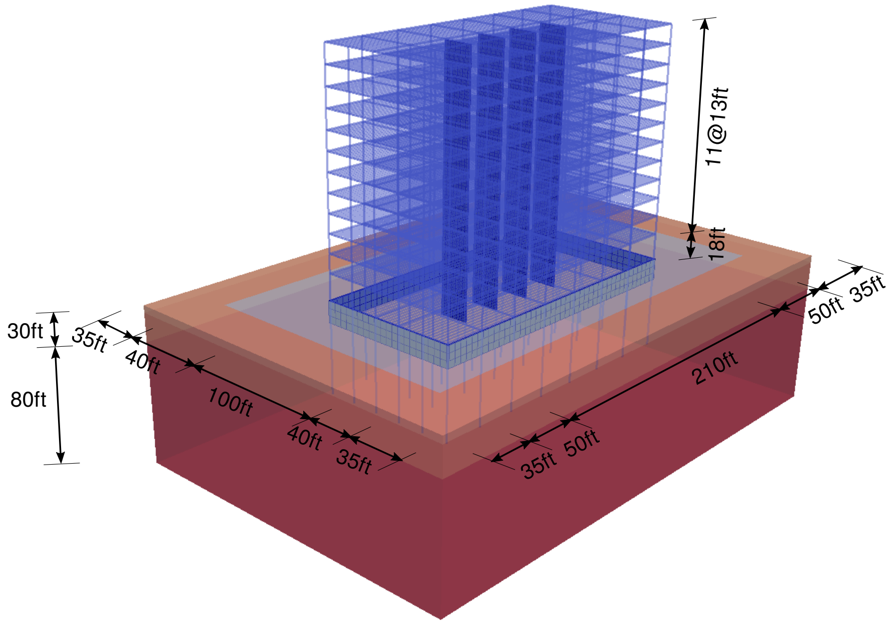


Figure 509.24: High concrete building model, 3D view.

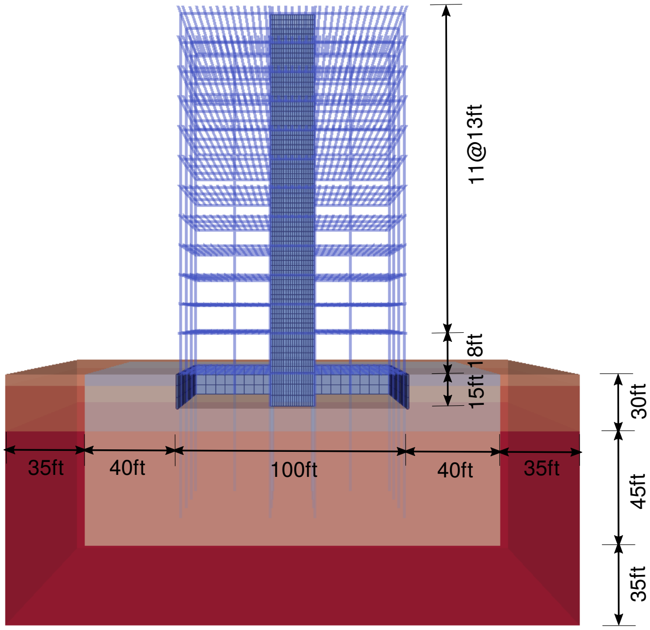


Figure 509.25: High concrete building model, XZ plane view.

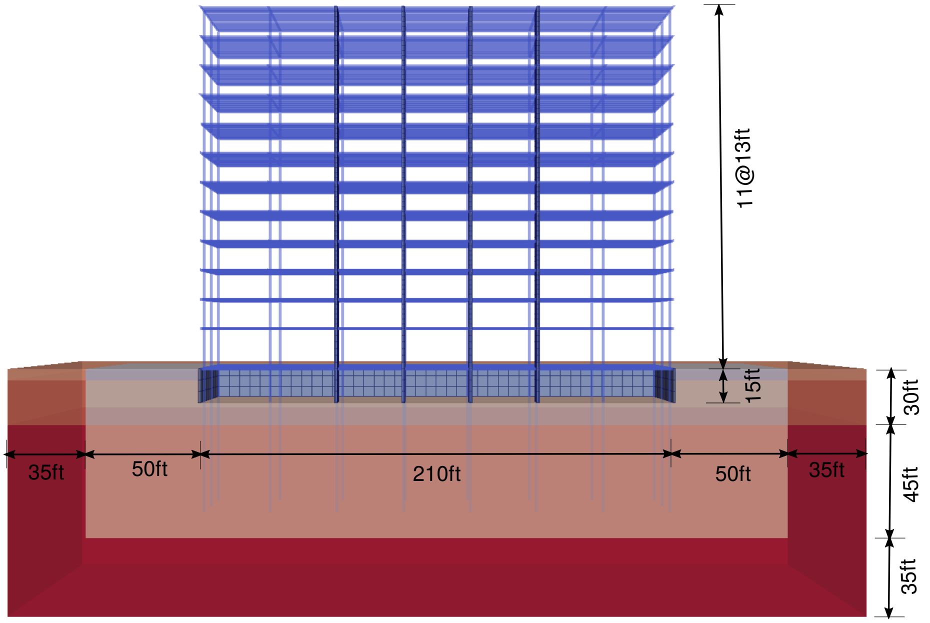


Figure 509.26: High concrete building model, YZ plane view.

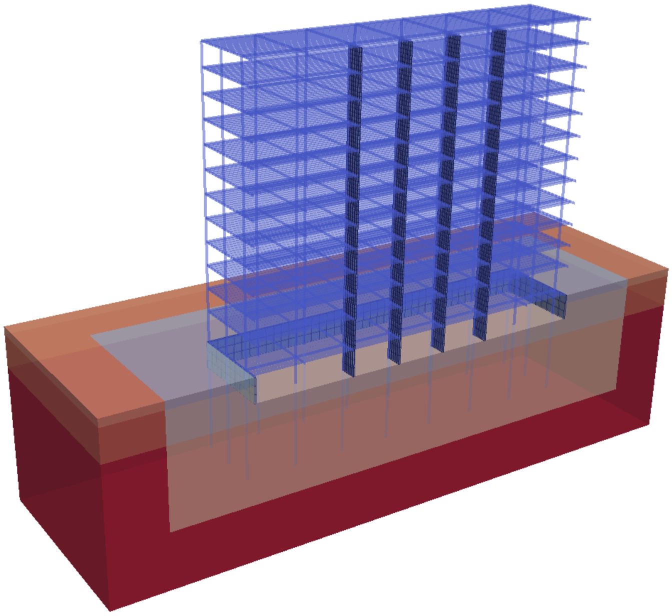


Figure 509.27: High concrete building model, cut through model view.

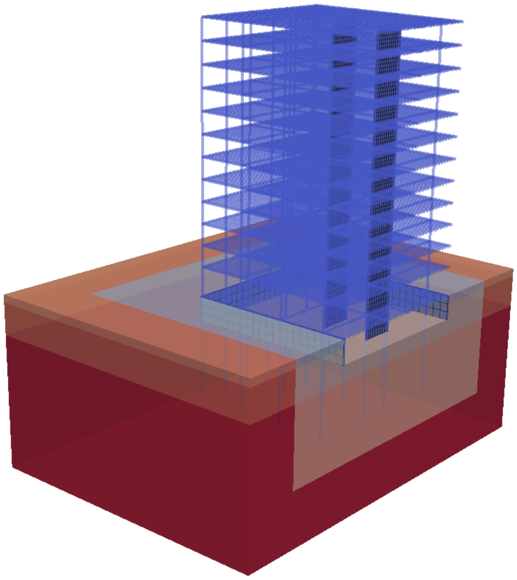


Figure 509.28: High concrete building model, cut through model view.

Finite element model, input file for the Real-ESSI simulator for full SSI ASCE-7 model, high concrete building are available HERE.

NOTE: This model is only 11MB large when compressed. When uncompressed (use `tar -xvf _ASCE-7_tall_concrete_building_Model_.tar.xz`), models files grow to 1.9GB, that is 1,867MB, or 1,911,524Kb, so please be aware of disk space requirements for the model alone.

Chapter 510

Guidebook: Modeling and Simulation of Earthquake-Soil-Structure Interaction for Nuclear Energy Installations, Dams, Buildings, Bridges, Tunnels, &c.

(2016-2017-2018-2019-2020-2021-)

510.1 Motivation: Modeling and Simulation of Earthquake Soil Structure Interaction

Main motivation of this write-up (chapter, guidebook) is to provide a clear, practical, up to date guide on how to perform linear elastic and nonlinear, inelastic Earthquake Soil Structure Interaction (ESSI) modeling and simulations for infrastructure objects, including buildings, dams, bridges, nuclear installations, etc. This is particularly important at this time as a number of endeavors are underway to perform realistic ESSI analysis for a number of important soil, rock – structure systems, including dams, nuclear installations, bridges, buildings, etc.

This write-up is further motivated by the modeling and simulation challenges that are part of any soil, rock – structure system. These challenges are illustrated in Figure 510.1 for a number of soil, rock and soil/rock-structure systems.

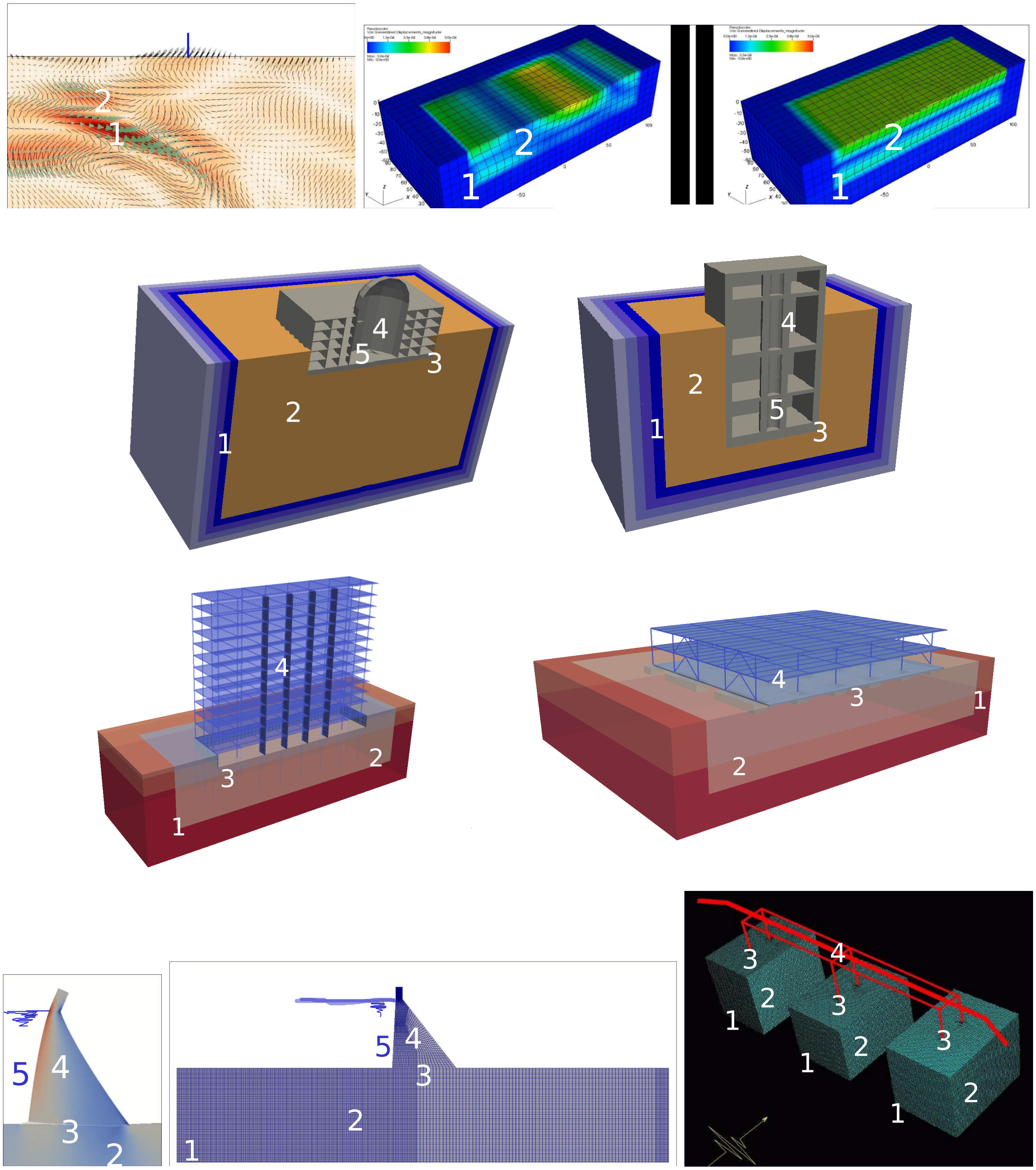


Figure 510.1: ESSI modeling and simulation challenges: Free field motions, 3C/6C vs 3×1C; Nuclear Power Plant structure – soil/rock system, Small Modular Reactor structure – soil/rock system; Low and High Building-foundation-soil system; Dam-Foundation-Fluid system; Bridge-soil system; Aspects of modeling: 1) Seismic motions, 2) Inelastic soil and rock, 3) Inelastic interface/contact/joints, foundation with soil/rock and interfaces/contacts/joints within structure, 4) Inelastic structure, systems and components, 5) Solid, Structure – Fluid interaction, external (reservoirs, fluid pools...) and internal (fully saturated and partially, (un-)saturated soil, rock and concrete).

Challenges for Modeling of Mechanics of Earthquake – Soil/Rock – Structure – Interaction. Presented are challenges related to the modeling of mechanics of ESSI problems.

1 Seismic motions: use of 1C, 3×1C and 3C motions, seismic motions input and radiation damping.

2 Inelastic, elastic plastic modeling of soil and rock, dry and/or partially or fully saturated, and energy dissipation in those soil/rock – structure components,

3 Inelastic, elastic plastic modeling of foundation concrete – soil/rock contacts/interfaces/joints that may be dry and/or partially or fully saturated, and energy dissipation in those parts of soil/rock – structure system,

4 Inelastic, elastic damage plastic modeling of structure, systems and components (SSCs). SSCs: beams, walls, plates, shell made of steel and reinforced concrete, base isolators and dissipators, systems and etc. and energy dissipation in those soil/rock – structure components,

5 Interaction of soil/rock – structure systems with internal, within structure or in pores of porous materials (soil, rock, concrete) and external fluids, reservoirs, pools, etc.

Challenges for Numerical Simulations of Earthquake – Soil/Rock – Structure – Interaction. Presented are challenges related to the numerical simulation of ESSI problems.

A Inelastic simulations on constitutive level, stress-strain, constitutive problem solutions

B Inelastic simulations on finite element level, nonlinear system of equations solutions

C Time marching algorithms, numerical damping

D High performance, parallel computing

510.2 Introduction

Focus is on modeling and simulation of linear elastic and nonlinear, inelastic, elastic-plastic behavior of soil/rock – structure systems during earthquakes.

It is assumed that earthquake motions, earthquake field is known. Earthquake motion or earthquake field, can be given as a simple 1C (1 Component) vertically propagating shear wave, that is obtained from a de-convolution of a given (1C) surface motion, using, for example SHAKE type analysis. Earthquake motion or earthquake field can also be given as a full 3C (3 Component) wave field that is obtained from analytic wave propagation solutions or from a regional geophysical model simulations, using, for example SW4 type analysis. In addition, earthquake motions can be defined in a probabilistic way...

510.3 Seismic Energy Input and Dissipation

Recorded lectures, together with slides, about these topics are available at
`http://sokocalo.engr.ucdavis.edu/~jeremic/Real_ESSI_Simulator/OnlineLectures`

More details about these topics are given in Section 109.2, on page 547 in Jeremić et al. (1989-2025).

510.3.1 Seismic Energy Input

Seismic energy input flux

510.3.2 Seismic Energy Dissipation

510.3.2.1 Seismic Energy Dissipation, Wave Reflection and Wave Radiation

Wave reflection and radiation damping

510.3.2.2 Seismic Energy Dissipation, Viscous Coupling

velocity proportional, viscous damping

510.3.2.3 Seismic Energy Dissipation, Material Inelasticity

Elastic-plastic energy dissipation of material, Displacement proportional

510.3.2.4 Seismic Energy Dissipation, Numerical, Algorithmic Positive and Negative Damping

510.4 Modeling: Seismic Motions

Recorded lectures, together with slides, about these topics are available at

`http://sokocalo.engr.ucdavis.edu/~jeremic/Real_ESSI_Simulator/OnlineLectures`

More details about these topics are given in Section 502.2, on page 2260 in Jeremić et al. (1989-2025).

510.4.1 Seismic Motions: Available Data

Details are given in Section 502.2.1, on page 2262 in Jeremić et al. (1989-2025).

510.4.2 Seismic Motion Development

Details are given in Section 502.2.3, on page 2266 in Jeremić et al. (1989-2025).

510.4.2.1 Seismic Motions from Empirical Models

Details are given in Section 109.2.3, on page 553 in Jeremić et al. (1989-2025).

510.4.2.2 Seismic Motions from Geophysical Models

Details are given in Section 109.2.6, on page 561 in Jeremić et al. (1989-2025).

Small Scale Geophysical Models.

Large Scale Regional Geophysical Models

510.4.2.3 Seismic Motions from 3D/3C Analytic Models

Details are given in Section 109.2.5, on page 557 in Jeremić et al. (1989-2025).

510.4.2.4 Seismic Motions from Full Waveform Inversion

This is based on recent work by Guidio (2020); Guidio et al. (2022b); Guidio and Jeong (2021); Guidio et al. (2022a).

510.4.3 6C vs 3C vs 3×1C vs 1C Seismic Motions

Details are given in Section 504.4.1, on page 2504 in Jeremić et al. (1989-2025).

510.4.4 Incoherent Seismic Motions

Details are given in Section 109.2.8, on page 562 in Jeremić et al. (1989-2025).

510.4.5 Seismic Motion Input into FEM Models

Details are given in Section 109.4.1, on page 568 in Jeremić et al. (1989-2025).

510.5 Modeling: Inelastic, Nonlinear Material Modeling for Solids and Structures

Recorded lectures, together with slides, about these topics are available at
`http://sokocalo.engr.ucdavis.edu/~jeremic/Real_ESSI_Simulator/OnlineLectures`

510.5.1 Inelastic Material Modeling of Rock

Details are given in Section 104.8, on page 329 in Jeremić et al. (1989-2025).

510.5.1.1 Calibration of Inelastic Material Model Parameters for Rock

Details are given in Section 104.10, on page 368 in Jeremić et al. (1989-2025).

510.5.2 Inelastic Material Modeling of Soil

510.5.2.1 Dry Soil

Details are given in Section 104.6, on page 220 in Jeremić et al. (1989-2025).

510.5.2.2 Fully Saturated Soil

Details are given in Section 102.12, on page 137 in Jeremić et al. (1989-2025).

510.5.2.3 Partially Saturated, Unsaturated Soil

Details are given in Section 102.12.2, on page 152 in Jeremić et al. (1989-2025).

510.5.2.4 Calibration of Inelastic Material Model Parameters for Soil

Details are given in Section 104.10, on page 368 in Jeremić et al. (1989-2025).

510.5.3 Inelastic Material Modeling of Steel

Details are given in Section 104.9, on page 363 in Jeremić et al. (1989-2025).

510.5.3.1 Calibration of Inelastic Material Model Parameters for Steel

Details are given in Section 104.10, on page 368 in Jeremić et al. (1989-2025).

510.5.4 Inelastic Material Modeling of Concrete

Solids, Beams, Plates, Walls and Shells

Details are given in Section 104.9, on page 363 in Jeremić et al. (1989-2025).

510.5.4.1 Calibration of Inelastic Material Model Parameters for Concrete

Details are given in Section 104.10, on page 368 in Jeremić et al. (1989-2025).

510.6 Modeling: Inelastic, Nonlinear Material Modeling for Contacts, Interfaces, and Joints

Recorded lectures, together with slides, about these topics are available at
`http://sokocalo.engr.ucdavis.edu/~jeremic/Real_ESSI_Simulator/OnlineLectures`

510.6.1 Material Modeling of Dry Contacts, Interfaces, and Joints (Concrete, Steel – Soil, Rock)

Details are given in Section 104.7, on page 306 in Jeremić et al. (1989-2025).

510.6.1.1 Calibration of Inelastic Material Model Parameters for Dry Contacts, Interfaces, and Joints (Concrete, Steel – Soil, Rock)

510.6.2 Material Modeling of Saturated Contacts, Interfaces, and Joints (Concrete, Steel – Soil, Rock)

Details are given in Section 104.7, on page 306 in Jeremić et al. (1989-2025).

510.6.2.1 Calibration of Inelastic Material Model Parameters for Saturated Contacts, Interfaces, and Joints (Concrete, Steel – Soil, Rock)

510.7 Modeling: Buoyancy

Details are given in Section 502.3.7, on page 2349 in Jeremić et al. (1989-2025).

510.8 Modeling: Base Isolator and Base Dissipator Systems

510.8.1 Base Isolator Systems

Details are given in Section 102.11, on page 136, in Jeremić et al. (1989-2025).

510.8.1.1 Calibration of Elastic/Inelastic Material Model Parameters for Base Isolator Systems

510.8.2 Base Dissipator Systems

Details are given in Section 102.11, on page 136, in Jeremić et al. (1989-2025).

510.8.2.1 Calibration of Elastic/Inelastic Material Model Parameters for Base Dissipator Systems

510.9 Modelling: Finite Element System

510.9.1 Mass Matrix

Details are given in Section 102.4, on page 114, Section 102.6, on page 126, Section 102.7, on page 126, Section 102.8, on page 129, Section 102.9, on page 135, Section 102.10, on page 135, Section 102.11, on page 136, and Section 102.12, on page 137, in Jeremić et al. (1989-2025).

510.9.1.1 Consistent Mass Matrix

510.9.1.2 Lumped Mass Matrix

510.9.2 Viscous Damping Matrix

Details are given in Section 108.4, on page 540, in Jeremić et al. (1989-2025).

510.9.2.1 Rayleigh Damping

510.9.2.2 Caughey Damping

510.9.3 Stiffness Matrix

Details are given in Section 102.4, on page 114, Section 102.6, on page 126, Section 102.7, on page 126, Section 102.8, on page 129, Section 102.9, on page 135, Section 102.10, on page 135, Section 102.11, on page 136, and Section 102.12, on page 137, in Jeremić et al. (1989-2025).

510.9.3.1 Tangent Stiffness Matrix

510.9.3.2 Consistent Stiffness Matrix

510.10 Modeling: Solid, Structure – Fluid Interaction Modeling

Details of OpenFOAM – Real-ESSI Simulator coupling are available starting with Section 111.2, on page 685, in Jeremić et al. (1989-2025).

510.11 Simulation: Nonlinear Finite Elements

Details are given in Section 102.2, on page 100 in Jeremić et al. (1989-2025).

510.11.1 Time Marching Algorithms for Solution of Nonlinear Equations of Motion

Details are given in Section 108.3, on page 538 in Jeremić et al. (1989-2025).

510.11.1.1 Newmark Algorithm

Details are given in Section 108.3.1, on page 538 in Jeremić et al. (1989-2025).

510.11.1.2 Hilber Hughes Taylor α Algorithm

Details are given in Section 108.3.2, on page 539 in Jeremić et al. (1989-2025).

510.11.2 Solution of Elastic-Plastic Constitutive Equations

Details are given in Section 104.2.2, on page 182 in Jeremić et al. (1989-2025).

510.11.2.1 Explicit Integration of Elastic-Plastic Constitutive Equations

Details are given in Section 104.3, on page 206 in Jeremić et al. (1989-2025).

Error accumulation.

510.11.2.2 Implicit Integration of Elastic-Plastic Constitutive Equations

Details are given in Section 104.4, on page 207 in Jeremić et al. (1989-2025).

Iterations and tolerance issues.

510.12 Modelling Guide for ESSI

Recorded lectures, together with slides, about these topics are available at

`http://sokocalo.engr.ucdavis.edu/~jeremic/Real_ESSI_Simulator/OnlineLectures`

More details about these topics are given in Section 502.3, on page 2332 in Jeremić et al. (1989-2025).

510.12.1 Buildings and NPPs on Shallow Foundations, Models

Details are given in Section 504.6, on page 2514 in Jeremić et al. (1989-2025).

510.12.2 Buildings and NPPs on Deeply Embedded Foundation (SMRs), Models

Details are given in Section 504.7, on page 2528 in Jeremić et al. (1989-2025).

510.12.3 Buildings and NPPs on Piles and Pile Group Foundations, Models

Details are given in Section 504.6, on page 2514 in Jeremić et al. (1989-2025).

510.12.4 Structure – Soil – Structure Interaction, Models

Details are given in Section 504.9, on page 2561 in Jeremić et al. (1989-2025).

510.13 Practical Steps for Inelastic ESSI Analysis

Recorded lectures, together with slides, about these topics are available at
`http://sokocalo.engr.ucdavis.edu/~jeremic/Real_ESSI_Simulator/OnlineLectures`

See Model Development section in Pecker et al. (2022).

510.13.1 Model Development for ESSI

Details are given in Section 502.4, on page 2354 in Jeremić et al. (1989-2025).

510.13.2 Earthquake Soil Structure Interaction: Model Analysis

510.13.3 Earthquake Soil Structure Interaction: Results Postprocessing

Details are given in Section 208.2, on page 1289 in Jeremić et al. (1989-2025).

510.14 Quality Assurance Procedures for ESSI Modeling and Simulation

Details of the quality assurance are given in Sections 314.3, on page 1893 and 314.4, on page 1896, in Jeremić et al. (1989-2025).

Moreover, verification procedures for ESSI modeling is given in part 300, on page 1436 in Jeremić et al. (1989-2025).

See also Verification and Validation section in Pecker et al. (2022).

510.14.1 Verification

510.14.2 Validation

510.15 Practical Examples, Nonlinear, Inelastic ESSI

Recorded lectures, together with slides, about these topics are available at

http://sokocalo.engr.ucdavis.edu/~jeremic/Real_ESSI_Simulator/OnlineLectures

510.15.1 Nuclear Power Plant, Inelastic Structure, Inelastic Soil, Inelastic Contact/Interface, 6C/3C/3×1C/1C Seismic Motions

510.15.2 Nuclear Power Plant on Piles, Inelastic Structure, Inelastic Soil, Inelastic Contact/Interface, 6C/3C/3×1C/1C Seismic Motions

510.15.3 Nuclear Power Plant, High Water Table, Inelastic Structure, Inelastic Soil, Cyclic Mobility and Liquefaction, Inelastic Saturated Contact/Interface, Buoyant Pressures, 6C/3C/3×1C/1C Seismic Motions

510.15.4 Small Modular Reactor, Deeply Embedded, Inelastic Structure, Inelastic Soil, Inelastic Contact/Interface, 6C/3C/3×1C/1C Seismic Motions

510.15.5 Small Modular Reactor, Deeply Embedded, High Water Table, Inelastic Structure, Inelastic Soil (Cyclic Mobility and Liquefaction), Inelastic Saturated Contact/Interface (Buoyant Pressures), 6C/3C/3×1C/1C Seismic Motions

510.15.6 Multiple Buildings and Nuclear Power Plants (Structure-Soil-Structure Interaction), Inelastic Structure, Inelastic Soil, Inelastic Contact/Interface, 6C/3C/3×1C/1C Seismic Motions

510.15.7 Multiple Small Modular Reactors (Structure-Soil-Structure Interaction), Deeply Embedded, High Water Table, Inelastic Structure, Inelastic Soil, Cyclic Mobility and Liquefaction, Inelastic Saturated Contact/Interface, Buoyant Pressures, 3C Seismic Motions

Chapter 511

ASCE-4, Chapter on Nonlinear ESSI analysis

(2016-2020-2021-)

511.1 Motivation: Modeling and Simulation of Earthquake Soil Structure Interaction

Main motivation of this write-up, chapter is to provide a clear, practical, up to date guide on how to perform nonlinear, inelastic Earthquake Soil Structure Interaction (ESSI) modeling and simulations for nuclear installations. This is particularly important at this time as a number of endeavors are underway to perform realistic ESSI analysis for a number of nuclear installations.

This write-up is further motivated by the modeling and simulation challenges that are part of any soil, rock – structure system. These challenges are illustrated in Figure 511.1 for a number of soil, rock and soil/rock-structure systems.

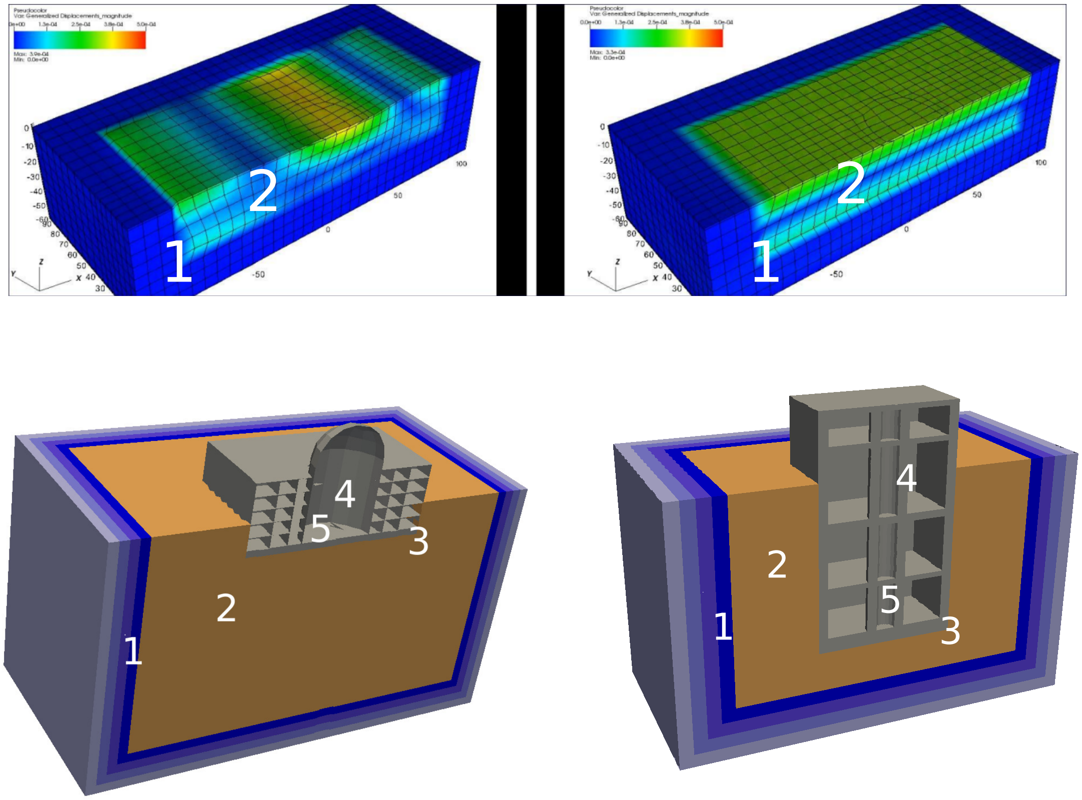


Figure 511.1: ESSI modeling and simulation challenges: Free field motions, 3C vs 3×1C; Nuclear Power Plant structure – soil/rock system, Small Modular Reactor structure – soil/rock system. Aspects of modeling: 1) Seismic motions, 2) Inelastic soil and rock, 3) Inelastic interface/contact/joints, foundation with soil/rock and interfaces/contacts/joints within structure, 4) Inelastic structure, systems and components, 5) Interaction with external (reservoirs, fluid pools...) and internal (saturated and un-saturated soil/rock and concrete.

Challenges for Modeling of Mechanics of Earthquake – Soil/Rock – Structure – Interaction. Presented are challenges related to the modeling of mechanics of ESSI problems.

1 Seismic motions: use of 1C, 3×1C and 3C motions, seismic motions input and radiation damping.

2 Inelastic, elastic plastic modeling of soil and rock, dry and/or partially or fully saturated, and energy dissipation in those soil/rock – structure components,

3 Inelastic, elastic plastic modeling of foundation concrete – soil/rock contacts/interfaces/joints that may be dry and/or partially or fully saturated, and energy dissipation in those parts of soil/rock – structure system,

4 Inelastic, elastic damage plastic modeling of structure, systems and components (SSCs). SSCs: beams, walls, plates, shell made of steel and reinforced concrete, base isolators and dissipators, systems and etc. and energy dissipation in those soil/rock – structure components,

5 Interaction of soil/rock – structure systems with internal, within structure or in pores of porous materials (soil, rock, concrete) and external fluids, reservoirs, pools, etc.

Challenges for Numerical Simulations of Earthquake – Soil/Rock – Structure – Interaction. Presented are challenges related to the numerical simulation of ESSI problems.

A Inelastic simulations on constitutive level, stress-strain, constitutive problem solutions

B Inelastic simulations on finite element level, nonlinear system of equations solutions

C Time marching algorithms, numerical damping

D High performance, parallel computing

511.2 Introduction

Focus is on modeling and simulation of linear elastic and nonlinear, inelastic, elastic-plastic behavior of soil/rock – structure systems during earthquakes.

It is assumed that earthquake motions, earthquake field is known. Earthquake motion or earthquake field, can be given as a simple 1C (1 Component) vertically propagating shear wave, that is obtained from a de-convolution of a given (1C) surface motion, using, for example SHAKE type analysis. Earthquake motion or earthquake field can also be given as a full 3C (3 Component) wave field that is obtained from analytic wave propagation solutions or from a regional geophysical model simulations, using, for example SW4 type analysis. In addition, earthquake motions can be defined in a probabilistic way...

Pecker et al. (2022).

511.3 Seismic Energy Input and Dissipation

511.3.1 Seismic Energy Input

Seismic energy input flux

511.3.2 Seismic Energy Dissipation

511.3.2.1 Seismic Energy Dissipation, Wave Reflection and Wave Radiation

Wave reflection and radiation damping

511.3.2.2 Seismic Energy Dissipation, Viscous Coupling

velocity proportional, viscous damping

511.3.2.3 Seismic Energy Dissipation, Material Inelasticity

Elastic-plastic energy dissipation of material, Displacement proportional

511.3.2.4 Seismic Energy Dissipation, Numerical, Algorithmic Positive and Negative Damping

511.4 Modeling: Seismic Motions

511.4.1 Seismic Motions: Available Data

511.4.2 Seismic Motion Development

511.4.2.1 Seismic Motions from Empirical Models

511.4.2.2 Seismic Motions from Geophysical Models

Small Scale Geophysical Models.

Large Scale Regional Geophysical Models

511.4.2.3 Seismic Motions from 3D/3C Analytic Models

511.4.3 6C vs 3C vs 3×1C vs 1C Seismic Motions

511.4.4 Incoherent Seismic Motions

511.4.5 Seismic Motion Input into FEM Models

511.5 Modeling: Inelastic, Nonlinear Material Modeling for Solids and Structures

511.5.1 Inelastic Material Modeling of Rock

511.5.1.1 Calibration of Inelastic Material Model Parameters for Rock

511.5.2 Inelastic Material Modeling of Soil

511.5.2.1 Dry Soil

511.5.2.2 Fully Saturated Soil

511.5.2.3 Partially Saturated, Unsaturated Soil

511.5.2.4 Calibration of Inelastic Material Model Parameters for Soil

511.5.3 Inelastic Material Modeling of Steel

511.5.3.1 Calibration of Inelastic Material Model Parameters for Steel

511.5.4 Inelastic Material Modeling of Concrete

Solids, Beams, Plates, Walls and Shells

511.5.4.1 Calibration of Inelastic Material Model Parameters for Concrete

511.6 Modeling: Inelastic, Nonlinear Material Modeling for Contacts, Interfaces, and Joints

511.6.1 Material Modeling of Dry Contacts, Interfaces, and Joints (Concrete, Steel – Soil, Rock)

511.6.1.1 Calibration of Inelastic Material Model Parameters for Dry Contacts, Interfaces, and Joints (Concrete, Steel – Soil, Rock)

511.6.2 Material Modeling of Saturated Contacts, Interfaces, and Joints (Concrete, Steel – Soil, Rock)

511.6.2.1 Calibration of Inelastic Material Model Parameters for Saturated Contacts, Interfaces, and Joints (Concrete, Steel – Soil, Rock)

511.7 Modeling: Base Isolator and Base Dissipator Systems

511.7.1 Base Isolator Systems

511.7.1.1 Calibration of Elastic/Inelastic Material Model Parameters for Base Isolator Systems

511.7.2 Base Dissipator Systems

511.7.2.1 Calibration of Elastic/Inelastic Material Model Parameters for Base Dissipator Systems

511.8 Modeling: Buried Pipes and Conduits

511.9 Modeling: Buoyancy

511.10 Modelling: Finite Element System

511.10.1 Mass Matrix

511.10.1.1 Consistent Mass Matrix

511.10.1.2 Lumped Mass Matrix

511.10.2 Viscous Damping Matrix

511.10.2.1 Rayleigh Damping

511.10.2.2 Caughey Damping

511.10.3 Stiffness Matrix

511.10.3.1 Tangent Stiffness Matrix

511.10.3.2 Consistent Stiffness Matrix

511.11 Modeling: Solid, Structure – Fluid Interaction Modeling

511.12 Simulation: Nonlinear Finite Elements

511.12.1 Time Marching Algorithms for Solution of Nonlinear Equations of Motion

511.12.1.1 Newmark Algorithm

511.12.1.2 Hilber Hughes Taylor α Algorithm

511.12.2 Solution of Elastic-Plastic Constitutive Equations

511.12.2.1 Explicit Integration of Elastic-Plastic Constitutive Equations

511.12.2.2 Implicit Integration of Elastic-Plastic Constitutive Equations

511.13 Modelling Guide for ESSI

511.13.1 Buildings and NPPs on Shallow Foundations, Models

511.13.2 Buildings and NPPs on Deeply Embedded Foundation (SMRs), Models

511.13.3 Buildings and NPPs on Piles and Pile Group Foundations, Models

511.13.4 Structure – Soil – Structure Interaction, Models

511.14 Practical Steps for Inelastic ESSI Analysis

511.14.1 Model Development for ESSI

511.14.2 Earthquake Soil Structure Interaction: Model Analysis

511.14.3 Earthquake Soil Structure Interaction: Results Postprocessing

511.15 Quality Assurance Procedures for ESSI Modeling and Simulation

511.15.1 Verification

511.15.2 Validation

511.16 Standard for Nonlinear/Inelastic Earthquake-Soil-Structure Analysis

511.16.1 Standard for Solids Analysis

511.16.2 Standard for Structure Analysis

511.16.3 Standard for Elastic-Plastic Analysis

511.17 Practical Examples, Nonlinear, Inelastic ESSI

511.17.1 Nuclear Power Plant, Inelastic Structure, Inelastic Soil, Inelastic Contact/Interface, 6C/3C/3×1C/1C Seismic Motions

511.17.2 Nuclear Power Plant on Piles, Inelastic Structure, Inelastic Soil, Inelastic Contact/Interface, 6C/3C/3×1C/1C Seismic Motions

511.17.3 Nuclear Power Plant, High Water Table, Inelastic Structure, Inelastic Soil, Cyclic Mobility and Liquefaction, Inelastic Saturated Contact/Interface, Buoyant Pressures, 6C/3C/3×1C/1C Seismic Motions

511.17.4 Small Modular Reactor, Deeply Embedded, Inelastic Structure, Inelastic Soil, Inelastic Contact/Interface, 6C/3C/3×1C/1C Seismic Motions

511.17.5 Small Modular Reactor, Deeply Embedded, High Water Table, Inelastic Structure, Inelastic Soil (Cyclic Mobility and Liquefaction), Inelastic Saturated Contact/Interface (Buoyant Pressures), 6C/3C/3×1C/1C Seismic Motions

511.17.6 Multiple Buildings and Nuclear Power Plants (Structure-Soil-Structure Interaction), Inelastic Structure, Inelastic Soil, Inelastic Contact/Interface, 6C/3C/3×1C/1C Seismic Motions

511.17.7 Multiple Small Modular Reactors (Structure-Soil-Structure Interaction), Deeply Embedded, High Water Table, Inelastic Structure, Inelastic Soil, Cyclic Mobility and Liquefaction, Inelastic Saturated Contact/Interface, Buoyant Pressures, 3C Seismic Motions

Chapter 512

Earthquake-Soil-Structure Interaction, Core Functionality

(2017-2018-2019-2021-)

(In collaboration with Dr. Yuan Feng, Prof. Han Yang and Dr. Hexiang Wang)

512.1 Core Functionality for ESSI Analysis of Nuclear Installations

Presented here are models that represent core functionality for elastic and inelastic analysis of infrastructure objects, including nuclear installations. There exist a number of other models, with different sophistication levels, that can be used, depending on the amount of data available, about the soil, rock, concrete, contacts/interfaces and seismic motions (Jeremić et al., 1989-2025). However, in order to begin to use of inelastic/nonlinear analysis, and assess inelastic/nonlinear effects on a dynamic response of soil structure systems, a set of initial models and analysis parameters are needed. Provided below is a set of models and materials parameters that are recommended for initial use of inelastic/nonlinear analysis of soil structure systems, using the Real-ESSI Simulator system. (`http://real-essi.info/`). It is noted that a detailed description of examples, commands and the Real-ESSI Simulator system is provided by Jeremić et al. (1989-2025, 1988-2025) and is also available at the Real-ESSI Simulator web site `http://real-essi.info/`. In addition, preprocessing, model development and postprocessing, results visualization for the Real-ESSI Simulator system is also described in detail pre and post processing documents that are available at `http://real-essi.info/`.

512.2 Model Setup

Each model has to be named:

```
model name "model_name_string";
```

In addition to that, there are a number of other considerations to be aware of:

- Each command line has to end with a semicolon ";"
- Comment on a line begins with either "//" or "!" and last until the end of current line.
- Units are required (see more below) for all quantities and variables.
- Include statements allow splitting source into several files
- All variables are double precision (i.e. floats) with a unit attached.
- All standard arithmetic operations are implemented, and are unit sensitive.
- Internally, all units are represented in the base SI units (m - s - kg).
- The syntax ignores extra white spaces, tabulations and newlines. Wherever they appear, they are there for code readability only. (This is why all commands need to end with a semicolon).

512.3 Linear Elastic Modeling

for single stage linear elastic modeling, one stage of loading has to be defined

```
new loading stage "self weight loading stage";
```

512.4 Nonlinear/Inelastic Modeling

For inelastic modeling, stages of loading have to be defined in proper sequence.

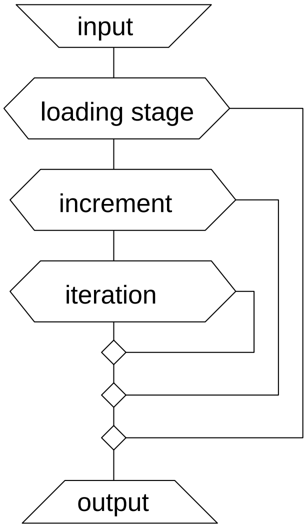


Figure 512.1:

```
new loading stage "self weight loading stage";
```

. . .

```
new loading stage "Seismic Loading";
```

. . .

512.5 Model Domain

Finite element model is developed by defining the finite element mesh which is made of nodes, finite elements, the material, and the loads.

512.5.1 Nodes

For example:

```
add node No 1 at (1.0*m, 2.5*m, 3.33*m) with 3 dofs;
```

adds a node number 1 at coordinates $x = 1.0m$, $y = 2.5m$ and $z = 3.33m$ with 3 dofs. The nodes can be of 3dofs $[u_x, u_y, u_z]$, 4dofs $[]u_x, u_y, u_z, p]$ (u-p elements) , 6dofs $[u_x, u_y, u_z, r_x, r_y, r_z]$ (beams and shells) and 7 dofs $[u_x, u_y, u_z, p, U_x, U_y, U_z]$ (upU element) types.

512.5.2 Boundary Conditions

Example fix translation x and y for node #3 `fix node # 3 dofs ux uy;`

Example fix all appropriate DOFs for node #7. `fix node # 7 dofs all;`

512.5.3 Static Acceleration Field

Example adding acceleration induced loading field for (some) elements

```
add acceleration field # 1
ax = 0*m/s^2
ay = 0*m/s^2
az = -9.81*m/s^2;
```

512.5.4 Dynamic Acceleration Field, Earthquake

One Example of add DRM load from wave fields:

```
add load # 1 type DRM from wave field
  # 1 in direction ux
  # 2 in direction uy
  soil_surface at z = 60.0*m
  hdf5_file = "input.hdf5" ;
```

512.5.5 Super Element

Super element is defined by providing mass and stiffness matrix, together with nodes and DOF numbering. It is assumed that the Super Element is a linear elastic element that is made up of a number of other finite elements. Super Element represents a part of model (structure, solid) that is linear elastic, and that has stiffness and mass matrix already defined using other finite element programs. Other finite element programs export stiffness and mass matrix. In addition, information about Super Element node numbers and degrees of freedom (DOFs) needs to be supplied as well.

512.6 Structural Modeling

Presented in this section are models that are used for modeling and simulation of structural behavior. Following the usually made assumption that structural components will remain linear elastic, only linear elastic material is used for structural modeling. It is noted, that fully nonlinear (inelastic, elastic-damage-plastic) models are also available for modeling of structural components (Jeremić et al., 1989-2025, 1988-2025). However, for the purpose of presenting core functionality features, those models are not covered here.

It is noted that a complete structural model can be replaced with one linear elastic super element, as described in section 512.5.5 on page 2769.

512.6.1 Truss

Truss element represents a 3D two node linear geometry truss member. Real-ESSI command for truss element is given in detail in section ??.

```
add element # 1 type truss
  with nodes (1,2)
  use material # 1
  cross_section = 1*m^2
  mass_density = 2000*kg/m^3;
```

512.6.2 Beam

Beam finite element represents a 3D linear geometry, two node Bernoulli beam member, with 6 DOFs per node. Real-ESSI command for beam element is given in detail in section ??.

```
add element # 1 type beam_elastic
    with nodes (1, 2)
    cross_section = 1*m^2
    elastic_modulus = 2e8*Pa
    shear_modulus = 1e8*Pa
    torsion_Jx = 0.33*m^4
    bending_Iy = 1.0/12*m^4
    bending_Iz = 1.0/12*m^4
    mass_density = 2000*kg/m^3
    xz_plane_vector = (1, 0, 1 )
    joint_1_offset = (0*m, 0*m, 0*m )
    joint_2_offset = (0*m, 0*m, 0*m );
```

512.6.3 Shell

Shell finite element represents a 3D linear elastic geometry, 4 node ANDES shell member with 6DOFs per node, including drilling DOFs (in plane twist). Real-ESSI command for shell element is given in detail in section ??.

```
add element # 1 type 4NodeShell_ANDES
   with nodes (1,2,3,4)
   use material # 1
   thickness = 1*m ;
```

512.7 Solid Modeling

Presented in this section are models that are used for modeling and simulation of soils, using solid and contact/interface elements for interface of foundations and soil. Models for soil can be linear elastic, while they can also be nonlinear/inelastic, mimicking simple G/G_{max} behavior. Models for contact/interface can represent bonded contact, where no slip or gapping is allowed, and also a frictional slip and gapping contact/interface.

It is noted, that a number of more or less sophisticated material models for soil and for contact/interface are also available (Jeremić et al., 1989-2025, 1988-2025). However, for the purpose of presenting core functionality features, those models are not covered here.

512.7.1 Solid Brick

Solid brick finite element with 8 nodes, linear interpolation of displacements between nodes, and three DOFs per node is available. This element is very good for modeling soil volume close to and far away from the structural. Real-ESSI command for 8 node solid brick is given in detail in section ??.

```
add element # 1 type 8NodeBrick
   using 2 Gauss points each direction
   with nodes (1, 2, 3, 4, 5, 6, 7, 8)
   use material # 1;
```

512.7.2 Contact, Interfaces, Joints

```
add element # 1 type StressBasedSoftContact_NonLinHardShear
   with nodes (1, 2)
   initial_axial_stiffness = 5*MPa
   stiffening_rate = 100
   max_axial_stiffness = 800*MPa
```

```
    initial_shear_stiffness = 800*kPa
    axial_viscous_damping = 50*Pa*s
    shear_viscous_damping = 50*Pa*s
    residual_friction_coefficient = 0.68
    shear_zone_thickness = 5e-3*m
    contact_plane_vector = (0, 0, 1 );
```

512.8 Core Material Modeling Parameters for Soil, Rock, Concrete, and Steel

512.8.1 Linear and Nonlinear Elastic Soil, Rock, Concrete, and Steel Modeling

512.8.2 Inelastic/Nonlinear Soil Modeling

Simple modeling of soil can be done using the so called stiffness degradation curves, or G/G_{max} curves, and damping curves, as developed by Seed and Idriss (1970a).

As an example, an elastic plastic material model based on von Mises yield surface with isotropic hardening or softening and Armstrong Frederick nonlinear kinematic hardening can be used to develop such curves. Model parameters are given below:

```
add material # 1 type vonMisesArmstrongFrederick
    mass_density = 2500*kg/m^3
    elastic_modulus = 30 * MPa
    poisson_ratio = 0.3
    von_mises_radius = 300 * Pa
    armstrong_frederick_ha = 150 * MPa
    armstrong_frederick_cr = 25000
    isotropic_hardening_rate = 0*Pa;
```

while the corresponding G/G_{max} and damping curves are given in Figure 512.2.

It is noted that von Mises Armstrong-Frederick Nonlinear Kinematic Hardening material model is a full 3D elastic plastilc material model, that is capable of modeling G/G_{max} and damping behavior, defined in 1D shear testing, fairly well in full 3D.

The command is

```
add material # <.> type vonMisesArmstrongFrederick
    mass_density = <M/L^3>
    elastic_modulus = <F/L^2>
    poisson_ratio = <.>
    von_mises_radius = <.>
    armstrong_frederick_ha = <F/L^2>
    armstrong_frederick_cr = <.>
    isotropic_hardening_rate = <F/L^2> ;
```

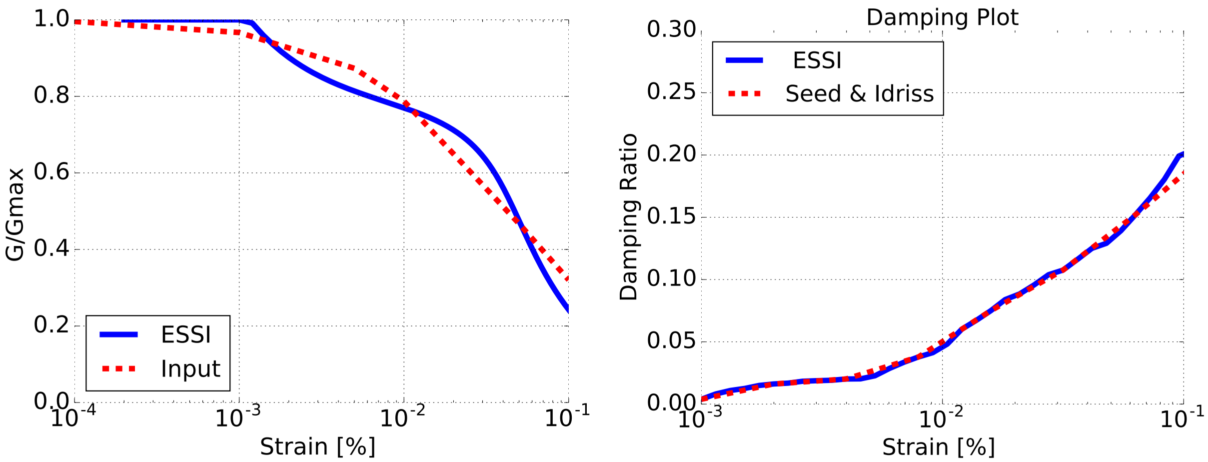


Figure 512.2: Stiffness degradation (G/G_{max}) and damping curves developed using von Mises Armstrong-Frederick Nonlinear Kinematic Hardening material model.

512.8.3 Inelastic/Nonlinear Rock Modeling

512.8.4 Inelastic/Nonlinear Concrete Modeling

512.8.5 Inelastic/Nonlinear Steel Modeling

512.9 Core Material Modeling Parameters for Contacts, Interfaces and Joints

The command for stress based dry soft nonlinear hardening is:

```
add element # 1 type StressBasedSoftContact_NonLinHardShear
   with nodes (1, 2)
   initial_axial_stiffness = 5*MPa
   stiffening_rate = 100
   max_axial_stiffness = 800*MPa
   initial_shear_stiffness = 800*kPa
   axial_viscous_damping = 50*Pa*s
   shear_viscous_damping = 50*Pa*s
   residual_friction_coefficient = 0.68
   shear_zone_thickness = 5e-3*m
   contact_plane_vector = (0, 0, 1 );
```

512.9.1 Mass Concrete Against Silt, Sand, Gravel and Clay

A set of initial recommended material parameters for frictional contact/interface are given in Tables 512.1 for contact between mass concrete and sand/gravel. Frictional properties given below are recommended by NAVFAC (1986).

Table 512.1: Friction coefficients for contact/interface of dissimilar materials, mass concrete against soil.

Mass concrete on soil	Friction coefficient ($\tan\phi$)	Friction angle (ϕ)
Clean sound rock	0.70	35^o
Clean gravel, gravel sand mixture, coarse sand	$0.55 – 0.60$	$29^o – 31^o$
Clean fine to medium sand, silty medium to coarse sand	$0.45 – 0.55$	$24^o – 29^o$
Fine sandy silt, nonplastic silt	$0.35 – 0.45$	$19^o – 24^o$
Very stiff clay	$0.40 – 0.50$	$22^o – 27^o$

Example command for a contact/interface element for mass concrete against clean sand, silty sand-gravel mix, single size rock fill (friction coefficient 0.30) is given below:

```
add element # 1 type StressBasedSoftContact_NonLinHardShear
   with nodes ( 1, 2)
   initial_axial_stiffness = 10 * MPa
   stiffening_rate = 100
   max_axial_stiffness = 50 * MPa
   initial_shear_stiffness = 40 * kPa
   axial_viscous_damping = 100 * Pa * s
   shear_viscous_damping = 100 * Pa * s
   residual_friction_coefficient = 0.30
   shear_zone_thickness = 5e-3*m
```

```
    contact_plane_vector = (0, 0, 1);
```

Another example, for contact/interface element for mass concrete against clean gravel, gravel sand mixture, coarse sand (friction coefficient 0.55-0.60) is given below:

```
add element # 1 type StressBasedSoftContact_NonLinHardShear
    with nodes ( 1, 2)
    initial_axial_stiffness = 20 * MPa
    stiffening_rate = 100
    max_axial_stiffness = 100 * MPa
    initial_shear_stiffness = 80 * kPa
    axial_viscous_damping = 200 * Pa * s
    shear_viscous_damping = 200 * Pa * s
    residual_friction_coefficient = 0.55
    shear_zone_tkness = 1e-2*m
    contact_plane_vector = (0, 0, 1);
```

512.9.2 Steel Sheet Against Sand, Gravel and Rockfill

Recommended material parameters for frictional contact/interface of steel sheets against sand and gravel are given in Tables 512.2. Frictional properties given below are recommended by NAVFAC (1986).

Table 512.2: Friction coefficients for contact/interface of dissimilar materials, steel sheet piles against soil.

Steel sheets against soil	Friction coefficient ($\tan\phi$)	Friction angle (ϕ)
Clean gravel, sand-gravel mix, well graded rock fill	0.40	22^o
Clean sand, silty sand-gravel mix, single size rock fill	0.30	17^o
Fine sandy silt, nonplastic silt	0.20	11^o

Example commands for contact/interface element for steel sheets against clean sand, silty sand-gravel mix, single size rock fill (friction coefficient 0.30) is given below:

```
add element # 1 type StressBasedSoftContact_NonLinHardShear
    with nodes ( 1, 2)
    initial_axial_stiffness = 1000 * MPa
    stiffening_rate = 100
    max_axial_stiffness = 5 * GPa
    initial_shear_stiffness = 4 * MPa
    axial_viscous_damping = 100 * Pa * s
    shear_viscous_damping = 100 * Pa * s
    residual_friction_coefficient = 0.30
```

```
   shear_zone_thickness = 5e-3*m
   contact_plane_vector = (0, 0, 1);
```

and for steel sheets against clean gravel, sand-gravel mix, well graded rock fill, with friction coefficient 0.40, command is:

```
add element # 1 type StressBasedSoftContact_NonLinHardShear
   with nodes ( 1, 2)
   initial_axial_stiffness = 2000 * MPa
   stiffening_rate = 100
   max_axial_stiffness = 10 * GPa
   initial_shear_stiffness = 8 * MPa
   axial_viscous_damping = 100 * Pa * s
   shear_viscous_damping = 100 * Pa * s
   residual_friction_coefficient = 0.40
   shear_zone_thickness = 5e-3*m
   contact_plane_vector = (0, 0, 1);
```

512.9.3 Formed Concrete Against Sand, Gravel and Rockfill

Recommended material parameters for frictional contact/interface of formed concrete against sand and gravel are given in Tables 512.3. Frictional properties given below are recommended by NAVFAC (1986).

Table 512.3: Friction coefficients for contact/interface of dissimilar materials, formed concrete against soil.

Formed concrete against soil	Friction coefficient ($\tan\phi$)	Friction angle (ϕ)
Clean gravel, sand-gravel mix, well graded rock fill	$0.40 - 0.50$	$22^o - 27^o$
Clean sand, silty sand-gravel mix, single size rock fill	$0.30 - 0.40$	$17^o - 22^o$
Silty sand, gravel or sand mixed with silt and clay	0.30	17^o
Fine sandy silt, nonplastic silt	0.25	14^o

Example command for contact/interface element for formed concrete against clean gravel, sand-gravel mix, well graded rock fill (friction coefficient 0.40-0.50) is given below:

```
add element # 1 type StressBasedSoftContact_NonLinHardShear
   with nodes ( 1, 2)
   initial_axial_stiffness = 30 * MPa
   stiffening_rate = 100
   max_axial_stiffness = 150 * MPa
   initial_shear_stiffness = 120 * kPa
   axial_viscous_damping = 100 * Pa * s
```

```
    shear_viscous_damping = 100 * Pa * s
    residual_friction_coefficient = 0.40
    shear_zone_thickness = 5e-3*m
    contact_plane_vector = (0, 0, 1);
```

512.9.4 Rock or Concrete on Rock or Concrete

More recently, Lei and Barton (2022) presented a very nice set of experiments with data for proper choice of interface parameters for rock on rock interface, that can also be used for concrete as well.

512.10 Earthquake Motion Modeling

512.10.1 One Component (1C) Seismic Motions Defined at Surface or at Depth

DRM...

One can add DRM loading directly, where `input.hdf5` specifies the DRM motions to all DRM nodes.

```
add load # 1 type DRM
    hdf5_file = "input.hdf5"
    scale_factor = 1.0 ;
```

Since the direct specification of DRM motions to all DRM nodes is complicated, alternatively, user is able to specify DRM motion using a surface motion. Internally, wave deconvolution is conducted to specify the DRM motions to all DRM nodes.

```
add wave field # 1 with
    acceleration_filename = "acceleration.txt"
    unit_of_acceleration = 1 * m/s^2
    displacement_filename = "displacement.txt"
    unit_of_displacement = 1 * m
    add_compensation_time = 0.0 * s
    motion_depth = 0 * m
    monitoring_location = within_soil_layer
    soil_profile_filename = "soil_profile.txt"
    unit_of_Vs = 1 * m/s
    unit_of_rho = 1 * kg/m^3
    unit_of_damping = absolute
    unit_of_thickness = 1*m
    ;
```

```
add load # 1 type DRM from wave field # 1 in direction ux
    soil_surface at z = 0.0*m
    hdf5_file = "input.hdf5" ;
```

where `input.hdf5` specifies the HDF5 file which contain the information about the DRM elements and DRM nodes.

512.10.2 3 × 1C Seismic Motions Defined at Surface or at Depth

One Example of add DRM load from wave fields:

```
add load # 1 type DRM from wave field
  # 1 in direction ux
  # 2 in direction uy
  # 3 in direction uz
  soil_surface at z = 0.0*m
  hdf5_file = "input.hdf5" ;
```

512.10.3 Seismic Motions Imposed at Model Base

```
add load # 1 type imposed motion to node # 1 dof ux
    time_step = 0.01*s
    displacement_scale_unit = 1*m
    displacement_file = "displacement.txt"
    velocity_scale_unit = 1*m/s
    velocity_file = "velocity.txt"
    acceleration_scale_unit = 1*m/s^2
    acceleration_file = "acceleration.txt";
```

512.10.4 Eigen Analysis

For structural model alone.

```
simulate using eigen algorithm
    number_of_modes = 3;
```

512.11 Core Modeling and Simulation Commands: Simulation Parameters

Developed model, using core functionality, as described above, numerically simulated using core functionality simulation controls.

Finite element system of equations can be solved in sequential processing mode, for smaller models, on sequential, single CPU computers (laptops, desktops, single CPU Amazon Web Services computers, etc.):

```
define solver sequential umfpack;
```

For larger models, parallel processing mode, on parallel computers ((multi CPU laptops, multi CPU desktops, clusters of PCs, Amazon Web Services parallel computers, Supercomputers, etc.):

Command Example for a direct solver:

```
define solver parallel petsc "-pc_type lu -pc_factor_mat_solver_package mumps" ;
```

For selfweight phase of loading, static solution algorithm is used:

```
 simulate 100 steps using static algorithm;
```

For static loading, for example self weight as described above, load application and the siulation process is controlled trough load control:

```
define load factor increment 0.01;
```

For dynamic loads, simulation process is controlled using Newmark time integration method:

```
define dynamic integrator Newmark with gamma = 0.6000 beta = 0.3025;
```

The dynamic simulation process is performed in a number of steps:

```
 simulate 2000 steps using transient algorithm time_step = 0.01*s;
```

For proper integration of constitutive equations on the integration point (Gauss point) level, within each finite element, constitutive algorithm needs to be defined:

```
define NDMaterial constitutive integration algorithm Forward_Euler;
```

For the finite element level, analysis of nonlinear systems require definition of nonlinear iteration algorithm:

```
define algorithm With_no_convergence_check;
```

Part 600

References

Bibliography

A.Angabini. Anisotropy of rock elasticity behavior and of gas migration in a Variscan Carboniferous rock mass in the South Limburg, The Netherlands. *Engineering Geology*, 67(3-4):353–372, 2003.

T. Abdoun, R. Dobry, and T. D. O'Rourke. Centrifuge and numerical modeling of soil–pile interaction during earthquake induced soil liquefaction and lateral spreading. In *Geotechnical Special Publications 64*, pages 76–90. ASCE, 1997.

J. A. Abell, N. Orbović, D. B. McCallen, and B. Jeremić. Earthquake soil structure interaction of nuclear power plants, differences in response to 3-D, 3×1-D, and 1-D excitations. *Earthquake Engineering and Structural Dynamics*, 47(6):1478–1495, May 2018. doi: 10.1002/eqe.3026. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/eqe.3026`.

R. Abraham, J. Marsden, and T. Ratiu. *Manifolds, Tensor Analysis, and Applications*, volume 75 of *Applied Mathematical Sciences*. Springer Verlag, second edition, 1988.

N. Abrahamson. Spatial variation of earthquake ground motion for application to soil-structure interaction. *EPRI Report No. TR-100463, March.*, 1992a.

N. Abrahamson. Generation of spatially incoherent strong motion time histories. In A. Bernal, editor, *Earthquake Engineering, Tenth World Conference*, pages 845–850. Balkema, Rotterdam, 1992b. ISBN 90 5410 060 5.

N. Abrahamson. Updated coherency model. *Report Prepared for Bechtel Corporation, April*, 2005.

N. A. Abrahamson, J. F. Schneider, and J. C. Stepp. Empirical spatial coherency functions for applications to soil–structure interaction analysis. *Earthquake Spectra*, 7(1):1–27, 1991.

A. N. Afanasenkov, V. M. Bogomolov, and I. M. Voskoboinikov. Generalized shock Hugoniot of condensed substances. *Journal of Applied Mechanics and Technical Physics*, 10(4):660–664, July 1969.

S. AG. Verification, ve56 interface element. Technical report, SOFISTIK AG, 2020.

G. Agha. Actors: A Model of Concurrent Computation in Distributed Systems. MIT Press, 1984.

K. Aki and P. G. Richards. Quantitative Seismology. University Science Books, 2^{nd} edition, 2002.

M. A. AL-Shudeifat. Highly efficient nonlinear energy sink. Nonlinear Dynamics, 76:1905–1920, January 2014. URL https://link.springer.com/article/10.1007/s11071-014-1256-x#citeas.

M. A. AL-Shudeifat, N. Wierschem, D. D. Quinn, A. F. Vakakis, L. A. Bergman, and B. F. Spencer. Numerical and experimental investigation of a highly effective single-sided vibro-impact non-linear energy sink for shock mitigation. International Journal of Non-Linear Mechanics, 52:96–109, 2013. ISSN 0020-7462. doi: https://doi.org/10.1016/j.ijnonlinmec.2013.02.004. URL https://www.sciencedirect.com/science/article/pii/S0020746213000322.

K. Alvin, H. M. de la Fuente, B. Haugen, and C. A. Felippa. Membrane triangles with corner drilling freedoms – I. the EFF element. Finite Elements in Analysis and Design, 12:163–187, 1992.

B. Amadei. Rock anisotropy and the theory of stress measurements. Lecture notes in engineering. Springer-Verlag, 1983.

B. Amadei and R. E. Goodman. The influence of rock anisotropy on stress measurements by overcoring techniques. Rock Mechanics, 15:167–180, December 1982. 10.1007/BF01240588.

T. v. Andel. New Views on an Old Planet. Cambridge University Press, Cambridge, 2 edition, 2008.

M. Anders and M. Hori. Stochastic finite element method for elasto-plastic body. International Journal for Numerical Methods in Engineering, 46:1897–1916, 1999.

M. Anders and M. Hori. Three-dimensional stochastic finite element method for elasto-plastic bodies. International Journal for Numerical Methods in Engineering, 51:449–478, 2000.

Working Paper for Draft Proposed International Standard for Information Systems–Programming Language C++. ANSI/ISO, Washington DC, April 1995. Doc. No. ANSI X3J16/95-0087 ISO WG21/N0687.

I. Antoniadis, E. Chatzi, D. Chronopoulos, A. Paradeisiotis, I. Sapountzakis, and S. Konstantopoulos. Low-frequency wide band-gap elastic/acoustic meta-materials using the k-damping concept, 2017.

T. H. Antoun, I. N. Lomov, and L. A. Glenn. Simulation of the penetration of a sequence of bombs into granitic rock. International Journal of Impact Engineering, 29(1-10):81–94, December 2003.

R. J. Apsel and J. E. Luco. Impedance functions for foundations embedded in a layered medium: An integral equation approach. Earthquake Engineering & Structural Dynamics, 15(2):213–231, 1987.

Y. Araki, T. Asai, and T. Masui. Vertical vibration isolator having piecewise-constant restoring force. Earthquake Engineering & Structural Dynamics, 38(13):1505–1523, 2009.

G. C. Archer. Object Oriented Finite Analysis. PhD thesis, University of California, Berkeley, 1996.

G. C. Archer, G. Fenves, and C. Thewalt. A new object–oriented finite element analysis program architecture. Computers and Structures, 70(1):63–75, 1999.

Architectural Institutite of Japan. Recomendation for design of building foundations, 1988.

G. B. Arfken and H. J. Weber. Mathematical methods for physicists. AAPT, 1999.

J. Argyris and H.-P. Mlejnek. Dynamics of Structures. North Holland in USA Elsevier, 1991.

P. Armstrong and C. Frederick. A mathematical representation of the multiaxial Bauschinger effect. Technical Report RD/B/N/ 731,, C.E.G.B., 1966.

K. Arulanandan and R. F. Scott, editors. Verification of Numerical Procedures for the Analysis of Soil Liquefaction Problems. A. A. Balkema, 1993.

C. Ashcraft. Ordering sparse matrices and transforming front trees. Technical report, Boeing Shared Service Group, 1999.

C. Ashcraft, D. Pierce, D. K. Wah, and J. Wu. The Reference Manual for SPOOLES, Release 2.2: An Object Oriented Software Library for Solving Sparse Linear Systems of Equations. Boeing Shared Services Group, Seattle, Washington, 1999.

ASME-VV-10. Standard for verification and validation in computational solid mechanics, V&V 10 - 2019. Technical report, American Society of Mechanical Engineers, 2019. ISBN: 9780791873168.

ASME-VV-20. Standard for verification and validation in computational fluid dynamics and heat transfer V&V 20 - 2009(r2016). Technical report, American Society of Mechanical Engineers, 2009. ISBN: 9780791832097.

ASME-VV-40. Assessing credibility of computational modeling through verification and validation: Application to medical devices V&V 40 - 2018. Technical report, American Society of Mechanical Engineers, 2018. ISBN: 9780791872048.

D. Aubry, A. Modaressi, and H. Modaressi. A constitutive model for cyclic behaviour of interfaces with variable dilatancy. Computers and Geotechnics, 9(1-2):47–58, 1990.

U. Ayachit. The ParaView Guide: A Parallel Visualization Application. Kitware, Inc., USA, 2015. ISBN 1930934300, 9781930934306.

I. Babuska and P. Chatzipantelidis. On solving elliptic stochastic partial differential equations. Computer Methods in Applied Mechanics and Engineering, 191:4093–4122, 2002.

I. Babuška and J. T. Oden. Verification and validation in computational engineering and science: basic concepts. Computer Methods in Applied Mechanics and Engineering, 193(36-38):4057–4066, Sept 2004.

I. Babuska, F. Nobile, J. Oden, and R. Tempone. Reliability, uncertainty estimates, validation and verification. Technical Report 04-05, ICES, 2004.

D. Baffet, J. Bielak, D. Givoli, T. Hagstrom, and D. Rabinovich. Long-time stable high-order absorbing boundary conditions for elastodynamics. Computer Methods in Applied Mechanics and Engineering, 241-244(0):20 – 37, 2012. ISSN 0045-7825. doi: http://dx.doi.org/10.1016/j.cma.2012.05.007. URL http://www.sciencedirect.com/science/article/pii/S0045782512001557.

Z. Bai. Class notes on large scale scientific computing. ECS 231, Department of Computer Science, UC Davis, 2007.

S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, Modern Software Tools in Scientific Computing, pages 163–202. Birkhäuser Press, 1997.

S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc Web page, 2001. http://www.mcs.anl.gov/petsc.

S. Balay, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, 2004.

A. Baltov and A. Sawczuk. A rule of anisotropic hardening. Acta Mechanica, I(2):81–92, 1965.

S. Bandis, A. Lumsden, and N. Barton". Fundamentals of rock joint deformation. International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts, 20(6):249 – 268, 1983.

A. Banerjee, E. Calius, and R. Das. Impact based wideband nonlinear resonating metamaterial chain. International Journal of Non-Linear Mechanics, 103:138–144, 2018. ISSN 0020-7462. doi: https://doi.org/10.1016/j.ijnonlinmec.2018.04.011. URL https://www.sciencedirect.com/science/article/pii/S0020746218300775.

J. P. Bardet, K. Ichii, and C. H. Lin. Eera, a computer program for equivalent-linear earthquake site response analyses of layered soil deposits. Technical report, University of Southern California, 2000.

M. Bart, J. F. Shao, and D. Lydzba. Poroelastic behaviour of saturated brittle rock with anisotropic damage. International Journal for Numerical and Analytical Methods in Geomechanics, 24(15):1139–1154, 2000.

F. Basone, M. Wenzel, O. S. Bursi, and M. Fossetti. Finite locally resonant metafoundations for the seismic protection of fuel storage tanks. Earthquake Engineering & Structural Dynamics, 48(2):232–252, 2019. doi: https://doi.org/10.1002/eqe.3134. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/eqe.3134.

U. Basu. Explicit finite element perfectly matched layer for transient three-dimensional elastic waves. International Journal for Numerical Methods in Engineering, 77(2):151–176, 2009.

K.-J. Bathe. Finite Element Procedures in Engineering Analysis. Prentice Hall Inc., 1982.

K.-J. Bathe. Finite Element Procedures in Engineering Analysis. Prentice Hall Inc., 1996. ISBN 0-13-301458-4.

K.-J. Bathe and E. Dvorkin. On the automatic solution of nonlinear finite element equations. Computers & Structures, 17(5-6):871–879, 1983.

K.-J. Bathe and L. Wilson, Edward. Numerical Methods in Finite Element Analysis. Prentice Hall Inc., 1976.

J. Batoz and G. Dhatt. Incremental displacement algorithms for nonlinear problems. International Journal for Numerical Methods in Engineering, 14:1262–1267, 1979. Short Communications.

H. Bavestrello, P. Avery, and C. Farhat. Incorporation of linear multipoint constraints in domain-decomposition-based iterative solvers - Part II: Blending FETI-DP and mortar methods and assembling floating substructures. Computer Methods in Applied Mechanics and Engineering, 196(8):1347–1368, January 2007.

K. Been and M. Jefferies. A state parameter for sands. Geotechnique, 35(2):99–112, 1985.

T. Belytschko, B. Moran, and M. Kulkarni. On the crucial role of imperfections in quasi-static viscoplastic solutions. Journal of Applied Mechanics, 58:658–665, 1991.

T. Belytschko, W. K. Liu, and B. Moran. Nonlinear finite elements for continua and structures. John Wiley & Sons, 2000.

T. Belytschko, W. K. Liu, and B. Moran, editors. Nonlinear Finite Element for Continua and Structures. John Wiley & Sons Ltd., New York, 2001.

Y. Ben-Haim and I. Elishakoff. Convex Models of Uncertainty in Applied Mechanics. Elsevier, Amsterdam, 1990.

T. Benz and R. Schwabb. A quantitative comparison of six rock failure criteria. International Journal of Rock Mechanics and Mining Sciences, 45(7):1176–1186, October 2008.

M. Benzi. Preconditioning techniques for large linear systems: A survey. Journal of Computational Physics, 182:418–477, 2002.

M. Benzi and M. Tůma. A robust incomplete factorization preconditioner for positive definite matrices. Numerical Linear Algebra with Applications, 10:385–400, 2003.

M. Benzi, C. D. Meyer, and M. Tůma. A sparse approximate inverse preconditioner for the conjugate gradient method. SIAM Journal on Scientific Computing, 17(5):1135–1149, 1996.

M. Benzi, J. K. Cullum, and M. Tůma. Robust approximate inverse preconditioning for the conjugate gradient method. SIAM Journal on Scientific Computing, 22(4):1318–1332, 2000.

P. G. Bergan and C. A. Felippa. A triangular membrane element with rotational degrees of freedom. Computer Methods in Applied Mechanics and Engineering, 50:25–69, 1985.

J. B. Berril, S. A. Christensen, R. J. Keenan, W. Okada, and J. K. Pettinga. Lateral–spreading loads on a piled bridge foundations. In S. E. Pinto, editor, Seismic Behavior of Ground and Geotechnical Structures, pages 173–183, 1997.

J. F. Besseling and E. Van Der Giessen. Mathematical modeling of inelastic deformation, volume 5. CRC Press, 1994.

P. N. Bićanić. Exact evaluation of contact stress state in computational elasto plasticity. Engineering Computations, 6:67–73, 1989.

J. Bielak. Dynamic response of non-linear building-foundation systems. Earthquake Engineering & Structural Dynamics, 6(1):17–30, 1978.

J. Bielak, K. Loukakis, Y. Hisada, and C. Yoshimura. Domain reduction method for three–dimensional earthquake modeling in localized regions. part I: Theory. Bulletin of the Seismological Society of America, 93(2):817–824, 2003a.

J. Bielak, K. Loukakis, Y. Hisada, and C. Yoshimura. Domain reduction method for three dimensional earthquake modeling in localized regions. part i: Theory. Bulletin of the Seismological Society of America, 93(2):817–824, 2003b.

A. Bilbao, R. Aviles, J. Agirrebeitia, and G. Ajuria. Proportional damping approximation for structures with added viscoelastic dampers. Finite Elements in Analysis and Design, 42:492–502, 2006.

B. A. Bilby, L. R. T. Gardner, and A. N. Stroh. Continuous distributions of dislocations and the theory of plasticity. In IX^e Congrès International de Mèchanique Appliquèe, volume VIII, pages 35–44, Universitè de Bruxelles, 50. Avènue Franklin Roosevelt, 1957.

M. Biot. Mechanics of deformation and acoustic propagation in porous media. Journal of Applied Physics, 33(4):1482–1498, April 1962.

M. A. Biot. Theory of finite deformations of porous solids. Indiana University Mathematical Journal, 21 (7):597–620, January 1972.

J. Bishop and R. Hill. A theory of the plastic distortion of a polycrystalline aggregate under combined stresses. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 42 (327):414–427, 1951.

F. Black and M. Scholes. The pricing of options and corporate liabilities. Journal of Political Economy, 81(3):637–654, 1973.

J. Bloch. How to design a good api and why it matters. In D. Musser, editor, Library-Centric Software Design LCSD'05. Object-Oriented Programming, Systems, Languages and Applications, October 2005. Presentation Slides.

W. Boggs and M. Boggs. Mastering UML with Rational Rose 2002. Sybex, Alameda CA 94501, 2002.

M. Bollhöfer and Y. Saad. On the relations between ILUs and factored approximate inverses. Technical Report UMSI-2001-67, Department of Computer Science and Engineering, University of Minnesota, 2001.

M. Bolton. The strength and dilatancy of sands. Geotechnique, 36(1):65–78, 1986.

G. Booch. Object Oriented Analysis and Design with Applications. Series in Object–Oriented Software Engineering. Benjamin Cummings, second edition, 1994.

R. Borja and A. Amies. Multiaxial cyclic plasticity model for clays. Journal of Geotechnical Engineering, 120(6):1051–1070, 1994.

R. Borja, C. Lin, K. Sama, and G. Masada. Modelling non–linear ground response of non–liquiefiable soils. Earthquake Engineering and Structural Dynamics, 29:63–83, 2000.

R. I. Borja. Conditions for instabilities in collapsible solids including volume implosion and compaction banding. Acta Geotechnica, 1:107–122, 2006.

R. I. Borja, H. Yih Chao, F. J. Montána, and C. Hua Lin. Nonlinear ground response at Lotung LSST site. ASCE Journal of Geotechnical and Geoenvironmental Engineering, 125(3):187–197, March 1999.

R. W. Boulanger and K. Tokimatsu, editors. Seismic Performance and Simulation of Pile Foundations in Liquefied and Laterally Spreading Ground. Geotechnical Special Publication No 145. ASCE Press, Reston, VA, 2006. 321 pp.

M. Boulon. Basic features of soil structure interface behaviour. Computers and Geotechnics, 7(1-2): 115–131, 1989.

M. Boulon and R. Nova. Modelling of soil-structure interface behaviour a comparison between elastoplastic and rate type laws. Computers and Geotechnics, 9(1-2):21–46, 1990.

M. Bouteca and Y. Gueguen. Mechanical properties of rocks: Pore pressure and scale effects. Oil & Gas Science and Technology, 54(6):703–714, 1999. URL `RockPapers/R12.pdf`.

B. L. Boyce, S. L. B. Kramer, H. E. Fang, T. E. Cordova, M. K. Neilsen, K. Dion, A. K. Kaczmarowski, E. Karasz, L. Xue, A. J. Gross, A. Ghahremaninezhad, K. Ravi-Chandar, S.-P. Lin, S.-W. Chi, J. S. Chen, E. Yreux, M. Rüter, D. Qian, Z. Zhou, S. Bhamare, D. T. O'Connor, S. Tang, K. I. Elkhodary, J. Zhao, J. D. Hochhalter, A. R. Cerrone, A. R. Ingraffea, P. A. Wawrzynek, B. J. Carter, J. M. Emery, M. G. Veilleux, P. Yang, Y. Gan, X. Zhang, Z. Chen, E. Madenci, B. Kilic, T. Zhang, E. Fang, P. Liu, J. Lua, K. Nahshon, M. Miraglia, J. Cruce, R. DeFrese, E. T. Moyer, S. Brinckmann, L. Quinkert, K. Pack, M. Luo, and T. Wierzbicki. The Sandia fracture challenge: blind round robin predictions of ductile tearing. International Journal of Fracture, 186:5–68, January 2014. doi: 10.1007/s10704-013-9904-6. URL `https://doi.org/10.1007/s10704-013-9904-6`.

M. Bragg. In Our Time, Seismology. BBC Podcast, March 2022. `https://www.bbc.co.uk/sounds/play/m00154gh`.

S. J. Brandenberg, R. W. Boulanger, B. L. Kutter, and D. Chang. Static pushover analyses of pile groups in liquefied and laterally spreading ground in centrifuge tests. Journal of Geotechnical and Geoenvironmental Engineering, 133(9):1055–1066, 2007.

H. J. Braudel, M. Abouaf, and J. L. Chenot. An implicit and incremental formulation for the solution of elastoplastic problems , by the finite element method. Computers Structures, 22(5):801–814, 1986.

L. Brillouin. Wave Propagation in Periodic Structures. McGraw-Hill Book Company, Inc, (1946) and Dover Publications, Inc. (1953), 1953.

D. A. Brown and C.-F. Shie. Three dimensional finite element model of laterally loaded piles. Computers and Geotechnics, 10:59–79, 1990a.

D. A. Brown and C.-F. Shie. Numerical experiments into group effects on the response of piles to lateral loading. Computers and Geotechnics, 10:211–230, 1990b.

D. A. Brown and C.-F. Shie. Some numerical experiments with a three dimensional finite element model of a laterally loaded pile. Computers and Geotechnics, 12:149–162, 1991.

D. A. Brown, C. Morrison, and L. C. Reese. Lateral loaded behavior of pile group in sand. Journal of Geotechnical Engineering, 114(11):1261–1277, November 1988.

S. Brûlé, E. H. Javelaud, S. Enoch, and S. Guenneau. Experiments on seismic metamaterials: Molding surface waves. Phys. Rev. Lett., 112:133901, Mar 2014. doi: 10.1103/PhysRevLett.112.133901. URL https://link.aps.org/doi/10.1103/PhysRevLett.112.133901.

W. Brumund and G. Leonards. Experimental study of static and dynamic friction between sand and typical constuction materials. Journal of Testing and Evaluation, 1(2):162–165, 1973.

A. Burnol, A. A. L. Landes, D. Raucoules, M. Foumelis, C. Allanic, F. Paquet, J. Maury, H. Aochi, T. Guillon, M. Delatre, P. Dominique, A. Bitri, S. Lopez, P. P. Pébaÿ, and B. Bazargan-Sabet. Impacts of Water and Stress Transfers from Ground Surface on the Shallow Earthquake of 11 November 2019 at Le Teil (France). Remote Sens., 2023. doi: 10.3390/rs15092270.

P. Cacciola, A. Tombari, and A. Giaralis. An inerter-equipped vibrating barrier for noninvasive motion control of seismically excited structures. Structural Control and Health Monitoring, 27(3):e2474, 2020. doi: https://doi.org/10.1002/stc.2474. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/stc.2474. e2474 STC-18-0444.R2.

C. Cai, J. Zhou, L. Wu, K. Wang, D. Xu, and H. Ouyang. Design and numerical validation of quasi-zero-stiffness metamaterials for very low-frequency band gaps. Composite Structures, 236:111862, 2020. ISSN 0263-8223. doi: https://doi.org/10.1016/j.compstruct.2020.111862. URL https://www.sciencedirect.com/science/article/pii/S0263822319331459.

B. Capra and A. Sellier. Orthotropic modelling of alkali-aggregate reaction in concrete structures: numerical simulations. Mechanics of materials, 35(8):817–830, 2003.

A. Cardona, I. Klapka, and M. Geradin. Design of a new finite element programming environment. Engineering Computations, 11:365–381, 1994.

I. Carol, E. Rizzi, and K. Willam. Current issues in elastic degradation and damage. In S. Sture, editor, Proceedings of 10th Conference, pages 521–524. Engineering Mechanics Division of the American Society of Civil Engineers, May 1995.

I. Carol, E. Rizzi, and K. William. On the formulation of anisotropic elastic degradadation. I. theory based on a pseudo–logarithmic damage tensor rate. International Journal of Solids and Structures, 38:491–518, 2001a.

I. Carol, E. Rizzi, and K. William. On the formulation of anisotropic elastic degradadation. II. generalized pseudo–Rankine model for tensile damage. International Journal of Solids and Structures, 38:519–546, 2001b.

G. Carta, G. Giaccu, and M. Brun. A phononic band gap model for long bridges. the ˇ2018brabauˇ2019 bridge case. Engineering Structures, 140:66–76, 2017. ISSN 0141-0296. doi: https://doi.org/10.1016/j.engstruct.2017.01.064. URL https://www.sciencedirect.com/science/article/pii/S0141029617303103.

W. T. Carter, T. L. Sham, and K. H. Law. A Parallel Finite Element Method and It's Prototype Implementation on a Hypercube. Computers and Structures, 31(6):921–934, 1989.

O. Casablanca, G. Ventura, F. Garescì, B. Azzerboni, B. Chiaia, M. Chiappini, and G. Finocchio. Seismic isolation of buildings using composite foundations based on metamaterials. Journal of Applied Physics, 123(17):174903, 2018. doi: 10.1063/1.5018005. URL https://doi.org/10.1063/1.5018005.

A. Casagrande and F. Rendon. Gyratory shear apparatus; design, testing procedures. Technical Report S–78–15, U.S. Army Corps of Engineers, Waterways Experiment Station, Vicksburg, Miss., 1978.

A. H.-C. Chan. A Unified Finite Element Solution to Static and Dynamic Problems in Geomecahnics. PhD thesis, Department of Civil Engineering, University College of Swansea, February 1988.

G. Chang and J. B. Mander. Seismic energy based fatigue damage analysis of bridge columns: Part I-Evaluation of seismic capacity. National Center for Earthquake Engineering Research Buffalo, NY, 1994.

R. G. Charlwood, S. V. Solymar, and D. D. Curtis. A review of alkali aggregate reactions in hydroelectric plants and dams. In Proceedings of the International Conference of Alkali-Aggregate Reactions in Hydroelectric Plants and Dams, Fredericton, Canada, pages 129–135, 1992.

L. Chen, J. Shao, and H. Huang. Coupled elastoplastic damage modeling of anisotropic rocks. Computer and Geotechnics, 37:187–194, 2010.

L. Chen, J. Zang, A. Hillis, G. Morgan, and A. Plummer. Numerical investigation of wave–structure interaction using openfoam. Ocean Engineering, 88:91–109, 2014.

S. Chen, B. Wang, S. Zhu, X. Tan, J. Hu, X. Lian, L. Wang, and L. Wu. A novel composite negative stiffness structure for recoverable trapping energy. Composites Part A: Applied Science and Manufacturing, 129:105697, 2020. ISSN 1359-835X. doi: https://doi.org/10.1016/j.compositesa.2019.105697. URL https://www.sciencedirect.com/science/article/pii/S1359835X19304464.

W. F. Chen and D. J. Han. Plasticity for Structural Engineers. Springer Verlag, 1988a.

W. F. Chen and D. J. Han. Plasticity for Structural Engineers. Springer-Verlag, 1988b.

X. Chen and K. Phoon. Some numerical experiences on convergence criteria for iterative finite element solvers. Computers and Geotechnics, 36(8):1272–1284, October 2009.

Z. Cheng and B. Jeremić. Numerical modeling and simulation of soil lateral spreading against piles. In In proceedings of the GeoOrlando, Geo Institute Annual Conference, Orlando, Florida, March 2009a.

Z. Cheng and B. Jeremić. Numerical modeling and simulations of piles in liquefiable soil. Soil Dynamics and Earthquake Engineering, 29:1405–1416, 2009b.

Z. Cheng and Z. Shi. Composite periodic foundation and its application for seismic isolation. Earthquake Engineering & Structural Dynamics, 47(4):925–944, 2018. doi: https://doi.org/10.1002/eqe.2999. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/eqe.2999.

Z. Cheng, M. Taiebat, B. Jeremić, and Y. Dafalias. Modeling and simulation of saturated geomaterials. In In proceedings of the GeoDenver conference, 2007.

S. Chiriţă, C. Galeş, and I. Ghiba. On spatial behavior of the harmonic vibrations in kelvin-voigt materials. Journal of Elasticity, 93(1):81–92, 2008.

C. H. Choi. Physical and Mathematical Modeling of Coarse-Grained Soils. PhD thesis, Department of Civil and Environmental Engineering, University of Washington, Seattle, Washington, 2004.

K. A. Chondrogiannis, V. Dertimanis, B. Jeremić, and E. Chatzi. On the vibration attenuation properties of metamaterial design using negative stiffness elements. In W. Lacarbonara, B. Balachandran, M. J. Leamy, J. Ma, J. T. Machado, and G. Stepan, editors, Advances in Nonlinear Dynamics - Proceedings of the Second International Nonlinear Dynamics Conference (NODYCON 2021), volume 3, Sapienza, Università di Roma, 16-19 February 2021. Springer Nature.

K. A. I. Chondrogiannis, V. K. Dertimanis, S. F. Masri, and E. N. Chatzi. Vibration absorption performance of metamaterial lattices consisting of impact dampers. In M. Papadrakakis, M. Fragiadakis, and C. Papadimitriou, editors, EURODYN 2020, XI International Conference on Structural Dynamics, Athens, Greece, 23-26 November 2020 2020. URL `https://doi.org/10.3929/ethz-b-000451611`.

A. K. Chopra. Dynamics of Structures, Theory and Application to Earthquake Engineering. Prentice Hall, second edition, 2000. ISBN 0-13-086973-2.

A. K. Chopra and J. A. Gutierrez. Earthquake response analysis of multistorey buildings including foundation interaction. Earthquake Engineering & Structural Dynamics, 3(1):65–77, 1974.

R. Chudoba and Z. Bittnar. Explicit Finite Element Computation: An Object-Oriented Approach. In Computing in Civil and Building Engineering: Proceedings of the Sixth International Conference on Computing in Civil and Building Engineering, Berlin, Germany, July 12-15 1995.

R. Clifton, J. Duffy, K. Hartley, and T. Shawki. On critical conditions for shear band formation at high strain rates. Scripta Metallurgica, 18(5):443–448, 1984.

I. Collins. Associated and non-associated aspects of the constitutive laws for coupled elastic/plastic materials. International Journal of Geomechanics, 2(2):259–267, 2002.

I. Collins. A systematic procedure for constructing critical state models in three dimensions. International Journal of Solids and Structures, 40(17):4379–4397, 2003.

I. Collins and P. Kelly. A thermomechanical analysis of a family of soil models. Geotechnique, 52(7): 507–518, 2002.

I. F. Collins and G. T. Houlsby. Application of thermomechanical principles to the modelling of geotechnical materials. Procedings of Royal Society London, 453:1975–2001, 1997.

A. Colombi, D. Colquitt, P. Roux, S. Guenneau, and R. V. Craster. A seismic metamaterial: The resonant metawedge. Scientific Reports, 6(27717):1–6, June 2016a. URL `https://www.nature.com/articles/srep27717#citeas`.

A. Colombi, D. Colquitt, P. Roux, S. Guenneau, and R. V. Craster. Forests as a natural seismic metamaterial: Rayleigh wave bandgaps induced by local resonances. Scientific Reports, 6(19238):1–7, January 2016b. URL https://www.nature.com/articles/srep19238.

A. Colombi, R. Zaccherini, G. Aguzzi, A. Palermo, and E. Chatzi. Mitigation of seismic waves: Metabarriers and metafoundations bench tested. Journal of Sound and Vibration, 485:115537, 2020. ISSN 0022-460X. doi: https://doi.org/10.1016/j.jsv.2020.115537. URL https://www.sciencedirect.com/science/article/pii/S0022460X20303692.

R. D. Cook, D. S. Malkus, M. E. Plesha, and R. J. Witt. Concepts and Applications of Finite Element Analysis. John Wiley & Sons, 2002.

J. O. Coplien. Advanced C++, Programming Styles and Idioms. Addison – Wesley Publishing Company, 1992.

M. Corigliano, L. Scandella, C. G. Lai, and R. Paolucci. Seismic analysis of deep tunnels in near fault conditions: a case study in southern Italy. Bulletin of Earthquake Engineering, 9(4):975–995, 2011.

E. Cosenza, G. Manfredi, and R. Ramasco. The use of damage functionals in earthquake engineering: a comparison between different methods. Earthquake engineering & structural dynamics, 22(10): 855–868, 1993.

E. Cosserat. Théorie des Corps Déformables. Éditions Jacques Gabay (2008), 151 bis rue Saint-Jacques, 75005 Paris, France, 1909. ISBN 978-2-87647-301-0. (originally published in 1909, by Librairie Scientifique A. Herman et Fils, 6, rue de la Sorbonne, 6, Paris).

R. Courant and D. Hilbert. Methods of Mathematical Physics. Wiley, 1989. ISBN 979-0-471-50447-4.

O. Coussy. Mechanics of Porous Continua. John Wiley and Sons, 1995. ISBN 471 95267 2.

O. Coussy. Poromechanics. John Wiley & Sons, 2004.

M. A. Crisfield. Consistent schemes for plasticity computation with the Newton Raphson method. Computational Plasticity Models, Software, and Applications, 1:136–160, 1987.

M. A. Crisfield. A consistent co-rotational formulation for non-linear, three-dimensional, beam-elements. Computer methods in applied mechanics and engineering, 81(2):131–150, 1990.

M. A. Crisfield. Non–Linear Finite Element Analysis of Solids and Structures Volume 1: Essentials. John Wiley and Sons, Inc. New York, 605 Third Avenue, New York, NY 10158–0012, USA, 1991.

M. A. Crisfield. Non–Linear Finite Element Analysis of Solids and Structures Volume 2: Advanced Topics. John Wiley and Sons, Inc. New York, 605 Third Avenue, New York, NY 10158–0012, USA, 1997a.

M. A. Crisfield. Non-linear Finite Element Analysis of Solids and Structures. John Wiley & Sons, 1997b.

L. Crivelli and C. Farhat. Implicit transient finite element structural computations on mimd systems: Feti v.s. direct solvers. In 34th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, La Jolla, CA, USA, April 19-22 1993.

M. Čubrinovski. Private communications. ..., 2007 –.

M. Čubrinovski and K. Ishihara. Assessment of pile group response to lateral spreading by single pile analysis. In Seismic Performance and Simulation of Pile Foundations in Liquefied and Laterally Spreading Ground, Geotechnical Special Publication No. 145, pages 242–254. ASCE, 2006.

M. Čubrinovski, R. Uzuoka, H. Sugita, K. Tokimatsu, M. Sato, K. Ishihara, Y. Tsukamoto, and T. Kamata. Prediction of pile response to lateral spreading by 3-d soil–coupled dynamic analysis: Shaking in the direction of ground flow. Soil Dynamics and Earthquake Engineering, 28(6):421–435, June 2008.

D. Curran. A pressure-induced strength transition in water-saturated geologic materials. In International Conference on Mechanical and Behaviour of Materials under Dynamic Loading / Congrès international sur le comportemant mécanique et physique des matériaux sous sollicitations dynamiques, volume J. Phys. IV France 04 (1994) C8-243-C8-247, 1994. URL `http://hal.archives-ouvertes.fr/docs/00/25/33/91/PDF/ajp-jp4199404C836.pdf`. DOI: 10.1051/jp4:1994836.

W. J. N. D. Erskine and S. T. Weir. Shock wave profile study of tuff from the nevada test site. Journal of Geophysical Research, 99(B8):15,529–15,537, 1994.

Y. Dafalias. A model for soil behavior under monotonic and cyclic loading conditions. In Proceedings of the 5th international conference on SMiRT, volume K 1/8, 1979.

Y. Dafalias and E. Popov. A model of nonlinearly hardening materials for complex loading. Acta mechanica, 21(3):173–192, 1975.

Y. Dafalias and E. Popov. Cyclic loading for materials with a vanishing elastic region. Nuclear Engineering and Design, 41(2):293–302, 1977.

Y. Dafalias, D. Schick, C. Tsakmakis, K. Hutter, and H. Baaser. A simple model for describing yield surface evolution. In Lecture notes in applied and computational mechanics, pages 169–201. Springer, 2002.

Y. F. Dafalias. Il'iushin's postulate and resulting thermodynamic conditions on elastic–plastic coupling. International Journal of Solids and Structures, 13:239–251, 1977.

Y. F. Dafalias. Bounding surface plasticity. I: Mathematical foundations and hypoplasticity. ASCE Journal of Engineering Mechanics, 112(9):966–987, September 1986.

Y. F. Dafalias and M. T. Manzari. Simple plasticity sand model accounting for fabric change effects. Journal of Engineering Mechanics, 130(6):622–634, 2004a.

Y. F. Dafalias and M. T. Manzari. Simple plasticity sand model accounting for fabric change effects. ASCE Journal of Engineering Mechanics, 130(6):622–634, June 2004b.

Y. F. Dafalias, M. T. Manzari, and A. G. Papadimitriou. SANICLAY: simple anisotropic clay plasticity model. International Journal for Numerical and Analytical Methods in Geomechanics, 30(12):1231–1257, 2006.

J. D'Alembert. Traité de Dynamique. Éditions Jacques Gabay, 151 bis rue Saint-Jacques, 75005 Paris, France, 1758. ISBN 2-87647-064-0. (originally published in 1758), This edition published in 1990.

S. K. Das and P. K. Basudhar. Comparison of intact rock failure criteria using various statistical methods. Acta Geotechnica, 4(3):223–231, September 2009.

M. T. Davisson and H. L. Gill. Laterally loaded piles in a layered soil system. Journal of the Soil Mechanics and Foundations Division, 89(SM3):63–94, May 1963. Paper 3509.

S. M. Day. Finite Element Analysis of Seismic Scattering Problems. PhD thesis, University of California at San Diego, 1977.

R. de Boer, W. Ehlers, and Z. Liu. One-dimensional transient wave propagation in fluid-saturated incompressible porous media. Archive of Applied Mechanics, 63(1):59–72, January 1993.

R. de Borst. Computation of post–biffurcation and post–failure behavior of strain–softening solids. Computers & Structures, 25(2):211–224, 1987.

R. de Borst. Smeared cracking, plasticity, creep, and thermal loading - a unified approach. Computer Methods in Applied Mechanics and Engineering, 62:89–110, 1987.

R. de Borst. Simulation of strain localization: a reappraisal of the Cosserat continuum. Engineering Computations, 8(4):317–332, 1991. doi: 10.1108/eb023842.

R. de Borst. A generalisation of J2-flow theory for polar continua. Computer Methods in Applied Mechanics and Engineering, 103(3):347–362, 1993. ISSN 00457825. doi: 10.1016/0045-7825(93)90127-J.

R. de Borst and P. H. Feenstra. Studies in anysotropic plasticity with reference to the hill criterion. International Journal for Numerical Methods in Engineering, 29:315–336, 1990.

B. S. L. P. De Lima, E. C. Teixeira, and N. F. F. Ebecken. Probabilistic and possibilistic methods for the elastoplastic analysis of soils. Advances in Engineering Software, 132:569–585, 2001.

M. K. Deb, I. M. Babuska, and J. T. Oden. Solution of stochastic partial differential equations using Galerkin finite element techniques. Computer Methods in Applied Mechanics and Engineering, 190:6359–6372, 2001.

B. J. Debusschere, H. N. Najm, A. Matta, O. M. Knio, and R. G. Ghanem. Protein labeling reactions in electrochemical microchannel flow: Numerical simulation and uncertainty propagation. Physics of Fluids, 15:2238–2250, 2003.

D. J. DeGroot and G. B. Baecher. Estimating autocovariance of in-situ soil properties. Journal of Geotechnical Engineering, 119(1):147–166, January 1993.

A. R. Dehkordi. 3D Finite Element Cosserat Continuum Simulation of Layered Geomaterials. PhD thesis, University of Toronto, 2008.

J. T. DeJong and Z. J. Westgate. Role of initial state, material properties, and confinement condition on local and global soil-structure interface behavior. ASCE Journal of Geotechnical and Geoenvironmental Engineering, 135(11):1646–1660, November 2009.

J. T. Dejong, D. J. White, and M. F. Randolph. Microscale observation and modeling of soil-structure interface behavior using particle image velocimetry. Soils and Foundations, 46(1):15–28, 2006.

I. Demirdžić and M. Perić. Space conservation law in finite volume calculations of fluid flow. International journal for numerical methods in fluids, 8(9):1037–1050, 1988.

J. W. Demmel. Applied Numerical Linear Algebra. SIAM, Philadelphia, 1997.

J. W. Demmel, M. T. Heath, and H. A. van der Vorst. Parallel numerical linear algebra. Technical report, LAPACK Working Note 60, UT CS–93–192, 1993.

J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu. A supernodal approach to sparse partial pivoting. SIAM J. Matrix Analysis and Applications, 20(3):720–755, 1999a.

J. W. Demmel, J. R. Gilbert, and X. S. Li. An asynchronous parallel supernodal algorithm for sparse gaussian elimination. SIAM J. Matrix Analysis and Applications, 20(4):915–952, 1999b.

J. W. Demmel, J. R. Gilbert, and X. S. Li. SuperLU Users' Guide, 2003.

D. Deniz, J. Song, and J. F. Hajjar. Energy-based seismic collapse criterion for ductile planar structural frames. Engineering Structures, 141:1–13, 2017.

J. E. Dennis, Jr. and R. B. Schnabel. Numerical Methods for Unconstrained Optimization and Nonlinear Equations. Prentice Hall , Engelwood Cliffs, New Jersey 07632., 1983.

A. Der Kiureghian and B. J. Ke. The stochastic finite element method in structural reliability. Journal of Probabilistic Engineering Mechanics, 3(2):83–91, 1988.

V. K. Dertimanis, I. A. Antoniadis, and E. N. Chatzi. Feasibility analysis on the attenuation of strong ground motions using finite periodic lattices of mass-in-mass barriers. Journal of Engineering Mechanics, 142(9):04016060, 2016. doi: 10.1061/(ASCE)EM.1943-7889.0001120. URL https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29EM.1943-7889.0001120.

C. Desai. Behavior of interfaces between structural and geologic media. Technical report, University of Missouri–Rolla, 1981.

C. Desai and B. Nagaraj. Modeling for cyclic normal and shear behavior of interfaces. Journal of engineering mechanics, 114(7):1198–1217, 1988.

S. S. Deshpande, L. Anumolu, and M. F. Trujillo. Evaluating the performance of the two-phase flow solver interfoam. Computational Science & Discovery, 5(1):014016, 2012.

C. di Prisco and F. Pisanò. An exercise on slope stability and perfect elasto-plasticity. Géotechnique, 61 (11):923–934, 2011.

C. di Prisco and D. Wood. Mechanical Behaviour of Soils Under Environmentally-Induced Cyclic Loads. Springer, 2012.

C. di Prisco, M. Pastor, and F. Pisanò. Shear wave propagation along infinite slopes: A theoretically based numerical study. International Journal for Numerical and Analytical Methods in Geomechanics, 36(5):619–642, 2012. ISSN 1096-9853. doi: 10.1002/nag.1020. URL http://dx.doi.org/10.1002/nag.1020.

S. E. Dickenson. Dynamic response of soft and deep cohesive soils during the Loma Prieta earthquake of October 17, 1989. PhD thesis, University of California, Berkeley, 1994.

S. Dmitriev. Language oriented programming: The next programming paradigm. published online http://www.onboard.jetbrains.com/is1/articles/04/10/lop/, November 2004.

R. Dobry and T. Abdoun. Recent studies on seismic centrifuge modeling of liquefaction and its effects on deep foundations. In S. Prakash, editor, Proceedings of Fourth International Conference on Recent Advances in Geotechnical Earthquake Engineering and Soil Dynamics, San Diego, March 26-31 2001.

R. Dobry, T. Abdoun, T. D. O'Rourke, and S. H. Goh. Single piles in lateral spreads: Field bending moment evaluation. Journal of Geotechnical and Geoenvironmental Engineering, 129(10):879–889, 2003.

C. R. Dohrmann. A preconditioner for substructuring based on constrained energy minimization. SIAM Journal of Scientific Computing, 25(1):246–258, September/October 2003.

M. Dolinski, D. Rittel, and A. Dorogoy. Modeling adiabatic shear failure from energy considerations. Journal of the Mechanics and Physics of Solids, 58(11):1759–1775, 2010.

J. Donea, S. Giuliani, and J.-P. Halleux. An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions. Computer methods in applied mechanics and engineering, 33(1-3):689–723, 1982.

P. Donescu and T. A. Laursen. A generalized object–oriented approach to solving ordinary and partial differential equations using finite elements. Finite Elements in Analysis and Design, 22:93–107, 1996.

J. Dongarra, I. Foster, G. Fox, W. Gropp, K. Kennedy, L. Torczon, and A. White. Source Book of Parallel Computing. Morgan Kaufmann Publishers, 2003.

J. J. Dongarra, I. S. Duff, D. C. Sorensen, and H. A. van der Vorst. Numerical Linear Algebra for High Performance Computers. Prentice Hall, New Jersey, 1996.

D. C. Drucker. On uniqueness in the theory of plasticity. Quarterly of Applied Mathematics, pages 35–42, 1956.

D. C. Drucker. A definition of stable inelastic material. Technical report, DTIC Document, 1957.

M. Dryden. Nees sfsi demonstration project. In NEES Project Meeting, Austin, TX, August 2005.

Y. Dubois-Pèlerein and T. Zimmermann. Object–oriented finite element programing: Iii. an efficient implementation in c++. Computer Methods in Applied Mechanics and Engineering, 108:165–183, 1993.

Y. Dubois-Pélerin and P. Pegon. Object-oriented programming in nonlinear finite element analysis. Computers & Structures, 67(4):225–241, 1998.

Y. Dubois-Pélerin and T. Zimmermann. Object Oriented Finite Element Programming: Theory and C++ Implementation for FEM_Objects C++ 01. Elmepress International, P.O.Box 2 CH 1015 Lausanne 15, Switzerland, 1992.

Y. Dubois-Pelerin and T. Zimmermann. Object-Oriented Finite Element Programming: III. An Efficient Implementation in C++. Computer Methods in Applied Mechanics and Engineering, 108(1-2):165–183, 1993.

Y. Dubois-Pelerin, T. Zimmermann, and P. Bomme. Object-Oriented Finite Element Programming: II. A Prototype Program in Smalltalk. Computer Methods in Applied Mechanics and Engineering, 98(3): 361–397, 1992.

Y.-D. Dubois-Pelerin. Object Oriented Finite Elements: Programming Concepts and Implementation. PhD thesis, Ecole Polytechnique Federale de Lausanne, 1992.

J. M. Duncan. State of the art: Limit equilibrium and finite-element analysis of slopes. ASCE Journal of Geotechnical and Geoenvironmental Engineering, 122(7):577–596, July 1996.

J. M. Duncan. Factors of safety and reliability in geotechnical engineering. ASCE Journal of Geotechnical and Geoenvironmental Engineering,, 126(4):307–316, April 2000a.

J. M. Duncan. Factors of safety and reliability in geotechnical engineering. ASCE Journal of Geotechnical and Geoenvironmental Engineering, 126(4):307–316, April 2000b.

J. M. Duncan and C.-Y. Chang. Nonlinear analysis of stress and strain in soils. Journal of Soil Mechanics and Foundations Division, 96:1629–1653, 1970.

Š. Dunica and B. Kolundžija. Nelinarna Analiza Konstrukcija. Gradjevinski fakultet Univerziteta u Beogradu, Bulevar revolucije 73 i IRO "Naučna knjiga", Beograd, Uzun–Mirkova 5, 1986. Nonlinear Analysis of Structures, In Serbian.

B. Eckel. Using C++. Osborne McGraw – Hill, 1989.

H. El Ganainy and M. El Naggar. Seismic performance of three-dimensional frame structures with underground stories. Soil Dynamics and Earthquake Engineering, 29(9):1249–1261, 2009.

A. Elgamal, Z. Yang, and E. Parra. Computational modeling of cyclic mobility and post–liquefaction site response. Soil Dynamics and Earthquake Engineering, 22:259–271, 2002.

A. Elgamal, Z. Yang, E. Parra, and A. Ragheb. Modeling of cyclic mobility in saturated cohesionless soils. International Journal of Plasticity, 19(6):883–905, 2003.

A. Elgamal, L. Yan, Z. Yang, and J. P. Conte. Three-dimensional seismic response of humboldt bay bridge-foundation-ground system. ASCE Journal of Structural Engineering, 134(7):1165–1176, July 2008.

M. A. Ellis and B. Stroustrup. The Annotated C++ Reference Manual. AT&T Bell Laboratories, Murray Hill, New Jersey and Addison - Wesley Publishing Company, 1990.

A. C. Eringen. Microcontinuum field theories: I. Foundations and solids. Springer Science & Business Media, 2012.

E. Evgin and K. Fakharian. Effect of stress paths on the behaviour of sand steel interfaces. Canadian geotechnical journal, 33(6):853–865, 1997.

D. Eyheramendy. Object–Oriented Finite Element Programming: symbolic Derivations and Automatic Programming. PhD thesis, Ecole Polytechniques Fédéral de Lausanne, 1997.

D. Eyheramendy and T. Zimmermann. Object–oriented finite elements II. a symbolic environment for automatic programming. Computer Methods in Applied Mechanics and Engineering, 132:277–304, 1996.

D. Eyheramendy and T. Zimmermann. Object-oriented finite elements. iv. symbolic derivations and automatic programming of nonlinear formulations. Computer Methods in Applied Mechanics and Engineering, 190(22-23):2729–2751, 2001.

F. Fadi and M. C. Constantinou. Evaluation of simplified methods of analysis for structures with triple friction pendulum isolators. Earthquake Engineering & Structural Dynamics, 39(1):5–22, January 2010.

C. Fairhurst. Stress estimation in rock: a brief history and review. International Journal of Rock Mechanics and Mining Sciences, 40(7-8):957–973, October – December 2003.

K. Fakharian. Three-dimensional monotonic and cyclic behaviour of sand-steel interfaces: Testing and modelling. University of Ottawa (Canada), 1996.

K. Fakharian and E. Evgin. Simple shear versus direct shear tests on interfaces during cyclic loading. In Proceedings of International Conference on Recent Advances in geotechnical Earthquake Engineering and Soil Dynamics, volume 1. University of Missouri–Rolla, 1995.

K. Fakharian and E. Evgin. An automated apparatus for three-dimensional monotonic and cyclic testing of interfaces. Geotechnical Testing Journal, 19(1):22–31, 1996.

K. Fakharian and E. Evgin. Cyclic simple-shear behavior of sand-steel interfaces under constant normal stiffness condition. Journal of Geotechnical and Geoenvironmental Engineering, 123(12):1096–1105, 1997.

K. Fakharian, E. Evgin, et al. A comprehensive experimental study of sand-steel interfaces subjected to various monotonic and cyclic stress paths. In The Twelfth International Offshore and Polar Engineering Conference. International Society of Offshore and Polar Engineers, 2002.

C. Farhat. Muliprocessors in Computational Mechanics. PhD thesis, University of California, Berkeley, 1987.

C. Farhat. Saddle-point principle domain decomposition method for the solution of solid mechanics problems. In Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, Norfolk, VA, USA, May 6-8 1991.

C. Farhat and L. Crivelli. A transient FETI methodology for large–scale parallel implicit comutations in structural mechanics. International Journal for Numerical Methods in Engineering, 37:1945–1975, 1994.

C. Farhat and M. Geradin. Using a reduced number of lagrange multipliers for assembling parallel incomplete field finite element approximations. Computer Methods in Applied Mechanics and Engineering, 97(3):333–354, June 1992.

C. Farhat and F. Roux. Method of finite element tearing and interconnecting and its parallel solution algorithm. International Journal for Numerical Methods in Engineering, 32(6):1205–1227, October 1991a.

C. Farhat and F. X. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. International Journal for Numerical Methods in Engineering, 32(6):1205–1227, 1991b.

C. Farhat, M. Lesoinne, and K. Pierson. A scalable dual-primal domain decomposition method. Numerical Linear Algebra with Applications, 7(7–8):687–714, 2000.

C. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, and D. Rixen. FETI-DP: a dual-primal unified FETI method - Part I: a faster alternative to the two-level FETI method. International Journal for Numerical Methods in Engineering, 50(7):1523–1544, 2001.

R. Faria, J. Oliver, and M. Cervera. A strain-based plastic viscous-damage model for massive concrete structures. International Journal for Solids and Structures, 35(14):1533–1558, 1998.

W. Farren and G. Taylor. The heat developed during plastic extension of metals. Proceedings of the royal society of London A: mathematical, physical and engineering sciences, 107(743):422–451, 1925.

M. Fasan. ADVANCED SEISMOLOGICAL AND ENGINEERING ANALYSIS FOR STRUCTURAL SEISMIC DESIGN. PhD thesis, UNIVERSITÀ DEGLI STUDI DI TRIESTE, 2016.

M. Fasan, A. Magrin, C. Amadio, F. Romanelli, F. Vaccari, and G. F. Panza. A seismological and engineering perspective on the 2016 Central Italy earthquakes. Int. J. Earthquake and Impact Engineering, 1(4):395–420, 2016. doi: 10.1504/IJEIE.2016.10004076.

B. Fatahi and S. H. R. Tabatabaiefar. Fully nonlinear versus equivalent linear computation method for seismic analysis of midrise buildings on soft soils. International Journal of Geomechanics, 14(4): 04014016, 2013.

Federal Highway Administration. Geotechnical engineering circular No 5. U.S. Department of Transportation, Publication No. FHWA-IF-02-034, April 2002.

H. P. Feigenbaum and Y. F. Dafalias. Directional distortional hardening in metal plasticity within thermodynamics. International Journal of Solids and Structures, 44(22-23):7526–7542, 2007.

C. A. Felippa. Dynamic relaxation under general incremental control. In W. K. Liu, T. Belytschko, and K. C. Park, editors, Innovative Methods for Nonlinear Problems, pages 103–133. Pineridge Press, Swansea U.K., 1984.

C. A. Felippa. Introduction to linear finite element methods, lecture notes, i and ii. Technical report, University of Colorado at Boulder, 1989. Report No. CU-CSSC-89-19 Septembar 1989.

C. A. Felippa. Object oriented finite element programing. Lecture Notes at CU Boulder, august - decembar 1992a.

C. A. Felippa. Object oriented finite element programing. Lecture Notes at CU Boulder, august - decembar 1992b.

C. A. Felippa. Nonlinear finite element methods. Lecture Notes at CU Boulder, 1993.

C. A. Felippa and S. Alexander. Membrane triangles with corner drilling freedoms – III. implementation and performance evaluation. Finite Elements in Analysis and Design, 12:203–239, 1992.

C. A. Felippa and C. Militello. Membrane triangles with corner drilling freedoms – II. the ANDES element. Finite Elements in Analysis and Design, 12:189–201, 1992.

C. A. Felippa and K. Park. Advanced finite element methods. Lecture Notes at CU Boulder, January-May 1995.

Y. Feng, H. Wang, H. Yang, and B. Jeremić. Hardware aware plastic domain decomposition for nonlinear finite element simulation. Advances in Engineering Software, to be submitted, 2024.

G. A. Fenton. Estimation of stochastic soil models. Journal of Geotechnical and Geoenvironmental Engineering, 125(6):470–485, June 1999a.

G. A. Fenton. Random field modeling of CPT data. Journal of Geotechnical and Geoenvironmental Engineering, 125(6):486–498, June 1999b.

G. A. Fenton and D. V. Griffiths. Probabilistic foundation settlement on spatially random soil. Journal of Geotechnical and Geoenvironmental Engineering, 128(5):381–390, May 2002.

G. A. Fenton and D. V. Griffiths. Bearing capacity prediction of spatially random $c-\phi$ soil. Canadian Geotechnical Journal, 40:54–65, 2003.

G. A. Fenton and D. V. Griffiths. Three-dimensional probabilistic foundation settlement. Journal of Geotechnical and Geoenvironmental Engineering, 131(2):232–239, February 2005.

G. L. Fenves. Object –oriented programming for engineering software development. Engineering with Computers, 6:1–15, 1990.

P. A. Ferrer. Elastoplatic characterization of granular materials. Master's thesis, University of Puerto Rico, 1992.

G. Festa and S. Nielsen. Pml absorbing boundaries. Bulletin of the Seismological Society of America, 93(2):891–903, 2003.

F. C. Filippou, V. V. Bertero, and E. P. Popov. Effects of bond deterioration on hysteretic behavior of reinforced concrete joints. Technical report, Earthquake Engineering Research Center, University of California, Berkeley, 1983.

S. Fiore, G. Finocchio, R. Zivieri, M. Chiappini, and Garesc?F. Wave amplitude decay driven by anharmonic potential in nonlinear mass-in-mass systems. Applied Physics Letters, 117(12):124101, 2020. doi: 10.1063/5.0020486. URL https://doi.org/10.1063/5.0020486.

B. W. R. Forde, R. O. Foschi, and S. F. Steimer. Object – oriented finite element analysis. Computers and Structures, 34(3):355–374, 1990.

L. Fox, H. D. Huskey, and J. H. Wilkinson. Notes on the solution of algebraic linear simultaneous equations. Quarterly Journal of Mechanics and Applied Mathematics, 1:149–173, 1948.

N. Fox. On the continuum theories of dislocations and plasticity. Quarterly Journal of Mechanics and Applied Mathematics, XXI(1):67–75, 1968.

R. E. Fulton and P. S. Su. Parallel substructure approach for massively parallel computers. Computers in Engineering, 2:75–82, 1992.

H. H. G. W. Ma and Y. X. Zhou. Modeling of wave propagation induced by underground explosion. Computers and Geotechnics, 22(3/4):283–303, 1998.

S. Gajan and D. S. Saravanathiiban. Modeling of energy dissipation in structural devices and foundation soil during seismic loading. Soil Dynamics and Earthquake Engineering, 31(8):1106 – 1122, 2011. ISSN 0267-7261. doi: 10.1016/j.soildyn.2011.02.006. URL http://www.sciencedirect.com/science/article/pii/S0267726111000923.

A. Gajo. Influence of viscous coupling in propagation of elastic waves in saturated soil. ASCE Journal of Geotechnical Engineering, 121(9):636–644, September 1995.

A. Gajo and L. Mongiovi. An analytical solution for the transient response of saturated linear elastic porous media. International Journal for Numerical and Analytical Methods in Geomechanics, 19(6): 399–413, 1995.

A. Gajo, A. Saetta, and R. Vitaliani. Evaluation of three– and two–field finite element methods for the dynamic response of saturated soil. International Journal for Numerical Methods in Engineering, 37: 1231–1247, 1994.

E. Gamma, R. Helm, R. Johnson, and J. Vlissides. Design Patterns. Elements of Reusable Object–Oriented Software. Professional Computing Series. Addison–Wesley, 1995. ISBN 0-201-63361-2.

C. W. Gardiner. Handbook of Stochastic Methods for Physics, Chemistry and the Natural Science. Springer:Complexity. Springer-Verlag, Berlin Heidelberg, third edition, 2004.

S. K. Garg, H. Nayfeh, and A. J. Good. Compressional waves in fluid-saturated elastic porous media. Journal of Applied Physics, 45(5):1968–1974, 1974.

G. C. Gazetas and J. M. Roesset. Vertical vibration of machine foundations. Journal of Geotechnical Engineering, 105(12):1435–1454, 1979.

A. Gens, I. Carol, and E. Alonso. A constitutive model for rock joints formulation and numerical implementation. Computers and Geotechnics, 9(1-2):3–20, 1990.

M. Georgiadis. Development of p-y curves for layered soils. In Geotechnical Practice in Offshore Engineering, pages 536–545. Americal Society of Civil Engineers, April 1983.

C. Geuzaine and J.-F. Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. International Journal for Numerical Methods in Engineering, 79(11):1309–1331, 2009.

R. Ghanem and R. M. Kruger. Numerical solution of spectral stochastic finite element systems. Computer Methods in Applied Mechanics and Engineering, 129(3):289 – 303, 1996. doi: http://dx.doi.org/10.1016/0045-7825(95)00909-4.

R. G. Ghanem and P. D. Spanos. Stochastic Finite Elements, A Spectral Approach. Dover Publications Inc., revised edition edition, 1991.

P. A. Gilbert. Investigation of density variation in triaxial test specimens of cohesionless soil subjected to cyclic and monotonic loadin. Technical Report GL–84–10, U.S. Army Corps of Engineers, Waterways Experiment Station, Vicksburg, Miss., 1984.

D. Givoli. Non-reflecting boundary conditions. Journal of computational physics, 94:1–29, 1991.

L. A. Glenn. The influence of rock material models on seismic discrimination of underground nuclear explosions. In 1995 APS Topical Conference on Shock Compression of Condensed Matter, Seattle, WA, August 13-18 1995. Lawrence Livermore National Laboratory. URL `http://www.osti.gov/bridge/servlets/purl/93963-W1J2gq/webviewable/93963.pdf`. UCRL-JC-120689.

P. Goldstein and A. Snoke. Sac availability for the iris community. Incorporated Institutions for Seismology Data Management Center Electronic Newsletter, 7, 2005.

P. Grammenoudis and C. Tsakmakis. Finite element implementation of large deformation micropolar plasticity exhibiting isotropic and kinematic hardening effects. International Journal for Numerical Methods in Engineering, 62(12):1691 – 1720, 2005.

G. Green. On the laws of the reflexion and refraction of light at the common surface of two non-crystallized media. Transactions of the Cambridge Philosophical Society, 7:1, 1848.

D. Griffiths and P. Lane. Slope stability analysis by finite elements. Geotechnique, 49(3):387–403, 1999.

D. V. Griffiths, G. A. Fenton, and N. Manoharan. Bearing capacity of rough rigid strip footing on cohesive soil: Probabilistic study. Journal of Geotechnical and Geoenvironmental Engineering, 128(9): 743–755, 2002.

T. H. Group. HDF5. https://www.hdfgroup.org/HDF5/, 2020.

B. Guidio and C. Jeong. Full-waveform inversion of incoherent dynamic traction in a bounded 2D domain of scalar wave motions. ASCE Journal of Engineering Mechanics, 147(4):04021010, 2021.

B. Guidio, B. Jeremić, L. Guidio, and C. Jeong. Passive-seismic inversion of sh-wave input motions in a domain truncated by wave-absorbing boundary conditions. Soil Dynamics and Earthquake Engineering, 2022a. In Print.

B. Guidio, B. Jeremić, L. Guidio, and C. Jeonga. Full-waveform inversion of sh-wave input motions in a domain truncated by wave-absorbing boundary conditions. Structural Dynamics and Earthquake Engineering, 2022b. in print.

B. P. Guidio. Full-waveform Inversion of Seismic Input Motions in a Near-surface Domain. PhD thesis, The Catholic University of America, 2020.

M. A. Gutierrez and R. De Borst. Numerical analysis of localization using a viscoplastic: Influence of stochastic material defects. International Journal for Numerical Methods in Engineering, 44:1823–1841, 1999.

D. Hadley and H. Kanamori. Seismic structure of the transverse ranges, california. Geological Society of America Bulletin, 88(10):1469–1478, 1977.

J. F. Hajjar and J. F. Abel. Parallel processing for transient nonlinear structural dynamics of three–dimensional framed structures using domain decomposition. Computers & Structures, 30(6):1237–1254, 1988.

J. F. Hall. Problems encountered from the use (or misuse) of Rayleigh damping. Earthquake Engineering & Structural Dynamics, 35(5):525–545, 2006.

M. Hamada. Large ground deformations and their effects on lifelines: 1964 Niigata earthquake. In Hamada and O'Rourke, editors, Case Studies of Liquefaction and Lifeline Performance During Past Earthquakes, Vol 1: Japanese Case Studies, Chapter 3, pages 3:1 – 3:123, 1992.

Y. Hamiel, V. Lyakhovsky, and A. Agnon. Rock dilation, nonlinear deformation, and pore pressure change under shear. Earth and Planetary Science Letters, 237:577–589, 2005.

B. O. Hardin. The nature of stress–strain behavior of soils. In Proceedings of the Specialty Conference on Earthquake Engineering and Soil Dynamics, volume 1, pages 3–90, Pasadena, 1978.

H. G. Harris and G. M. Sabnis. Structural Modeling and Experimental Techniques. CRC Press, 1999. ISBN 0849324696.

J. P. D. Hartog. Advanced Strength of Materials. Dover Publications, Inc., 1952.

Y. Hashash and D. Park. Non-linear one-dimensional wave propagation in the Mississipi embayment. Engineering Geology, 62(1–3):185–206, 2001.

Y. Hashash and D. Park. Viscous damping formulation and high frequency motion in non–linear site response analysis. Soil Dynamics and Earthquake Engineering, 22:611–624, 2002.

Y. M. Hashash, J. J. Hook, B. Schmidt, I. John, and C. Yao. Seismic design and analysis of underground structures. Tunnelling and underground space technology, 16(4):247–293, 2001.

Z. Hashin. Analysis of composite materials. Journal of Applied Mechanics, 50:481–505, 1983.

N. Haskell. Radiation pattern of surface waves from point sources in a multi-layered medium. Bulletin of the Seismological Society of America, 54(1):377–393, 1964.

N. A. Haskell. The dispersion of surface waves on multilayered media. Bulletin of the Seismological Society of America, 43(1):17–34, 1953.

H. M. Hassan, M. Fasan, M. A. Sayed, F. Romanelli, M. N. ElGabry, F. Vaccari, and A. Hamed. Site-specific ground motion modeling for a historical Cairo site as a step towards computation of seismic input at cultural heritage sites. Engineering Geology, 2020. doi: 10.1016/j.enggeo.2020.105524.

L. Hatton. The T experiments: Errors in scientific software. IEEE Computational Science and Engineering, 4(2):27–38, April–June 1997.

S. Helgason. Differential Geometry, Lie Groups, and Symmetric Spaces. Pure and Applied Mathematics. Academic Press, 1978.

B. Hendrickson and R. Leland. A multilevel algorithm for partitioning graphs. Proceedings Supercomputing '95, 1995.

M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. Journal of Research of the National Bureau of Standards, 49:409–436, 1952.

C. Hewitt, P. Bishop, and R. Steiger. A Universal Modular ACTOR Formalism for Artificial Intelligence. In Proceedings of the 3rd International Joint Conference on Artificial Intelligence, Stanford, CA, August 1973.

K. Hijikata, M. Takahashi, T. Aoyagi, and M. Mashimo. Behavior of a base-isolated building at fukushima dai-ichi nuclear power plant during the great east japan earthquake. In Proceedings of the International Symposium on Engineering Lessons Learned from the 2011 Great East Japan Earthquake, Tokyo, Japan, March 1-4 2012.

H. M. Hilber, T. J. R. Hughes, and R. L. Taylor. Improved numerical dissipation for time integration algorithms in structural dynamics. Earthquake Engineering and Structure Dynamics, 5(3):283–292, 1977.

R. Hill. The Mathematical Theory of Plasticity. The Oxford Engineering Science Series. Oxford at the Clarendon Press, 1st. edition, 1950.

R. Hill. A general theory of uniqueness and stability in elastic-plastic solids. Journal of the Mechanics and Physics of Solids, 6(3):236–249, 1958.

R. Hill. Aspects of invariance in solid mechanics. In C.-S. Yih, editor, Advances in Applied Mechanics, volume 18, pages 1–72. Academic Press, 1978.

M. Hiltl, C. R. Hagelberg, W. J. Nellis, T. C. Carney, and R. P. Swift. Dynamic response of berea sandstone shock-loaded under dry, wet and water-pressurized conditions. In Proceedings of International Conference on High Pressure Science and Technology, Honolulu, HI (USA), 1999. Preprint: UCRL-JC-135647.

A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward. SUNDIALS: SUite of Nonlinear and DIfferential/ALgebraic equation Solvers. ACM Transactions on Mathematical Software, 31(3):363–396, 2005.

C. Hird and D. Russell. A benchmark for soil-structure interface elements. Computers and Geotechnics, 10(2):139 – 147, 1990. ISSN 0266-352X. doi: https://doi.org/10.1016/0266-352X(90)90003-E. URL http://www.sciencedirect.com/science/article/pii/0266352X9090003E.

M. S. Hiremath, R. S. Sandhu, L. W. Morland, and W. E. Wolfe. Analysis of one-dimensional wave propagation in a fluid-saturated finite soil column. International Journal for Numerical and Analytical Methods in Geomechanics, 12:121–139, 1988.

C. Hirt, A. A. Amsden, and J. Cook. An arbitrary lagrangian-eulerian computing method for all flow speeds. Journal of computational physics, 14(3):227–253, 1974.

C. W. Hirt and B. D. Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. Journal of computational physics, 39(1):201–225, 1981.

Y. Hisada. Broadband strong motion simulation in layered half–space using stochastic green's function technique. Journal of Seismology, 12(2):265–279, 2008.

K. Hjelmstad. Fundamentals of Structural Mechanics. Prentice–Hall, 1997. ISBN 0-13-485236-2.

J. Hodowany, G. Ravichandran, A. Rosakis, and P. Rosakis. Partition of plastic work into heat and stored energy in metals. Experimental mechanics, 40(2):113–123, 2000.

E. Hoek, C. Carranza-Torres, and B. Corkum. Hoek-Brown failure criterion: 2002 edition. In 5th North American Rock Mechanics Symposium and 17th Tunneling Association of Canada Conference: NARMS-TAC, pages 267–271, 2002.

C. Holyoke and T. Rushmer. An experimental study of grain scale melt segregation mechanisms in two common crustal rock types. Journal of Metamorphic Geology, 20(5):493–512, 2002.

E. Hopf. Statistical hydromechanics and functional calculus. Journal of rational Mechanics and Analysis, 1:87–123, June 1952.

M. Hori. Introduction to Computational Earthquake Engineering. Imperial College Press, 2006. ISBN ISBN-10: 1848163983; ISBN-13: 978-1848163980.

K. Horikoshi, A. Tateishi, and T. Fujiwara. Centrifuge modeling of single pile subjected to liquefaction-induced lateral spreading. Soils and Foundations, Special Issue(2):193–208, 1998.

G. Houlsby and A. Puzrin. A thermomechanical framework for constitutive models for rate-independent dissipative materials. International Journal of Plasticity, 16(9):1017–1047, 2000.

G. W. Housner. Interaction of building and ground during an earthquake. Bulletin of the Seismological Society of America, 47(3):179–186, 1957.

http://graal.ens-lyon.fr/MUMPS/. MUltifrontal Massively Parallel Solver (MUMPS Version 4.6.2) Users' Guide, 2006.

http://www.sdsc.edu/user_services/datastar/. Ibm datastar user guide. San Diego Supercomputer Center at UCSD, 2020.

L. Hu and J. Pu. Testing and modeling of soil-structure interface. Journal of Geotechnical and Geoenvironmental Engineering, 130(8):851–860, 2004.

C. Hua. An inverse transformation for quadrilateral isoparametric elements: analysis and application. Finite elements in analysis and design, 7(2):159–166, 1990.

Y.-N. Huang, A. S. Whittaker, and N. Luco. Seismic performance assessment of base-isolated safety-related nuclear structures. Earthquake Engineering and Structures Dynamics, Early View: 20 SEP 2010 — DOI: 10.1002/eqe.1038:1–22, 2010.

T. Hughes. The Finite Element Method ; Linear Static and Dynamic Finite Element Analysis. Prentice Hall Inc., 1987.

T. Hughes and K. Pister. Consistent linearization in mechanics of solids and structures. Computers and Structures, 8:391–397, 1978.

T. J. R. Hughes and W. K. Liu. Implicit-explicit finite elements in transient analaysis: implementation and numerical examples. Journal of Applied Mechanics, pages 375–378, 1978a.

T. J. R. Hughes and W. K. Liu. Implicit-explicit finite elements in transient analaysis: stability theory. Journal of Applied Mechanics, pages 371–374, 1978b.

F.-N. Hwang and X.-C. Cai. A class of parallel two-level nonlinear Schwarz preconditioned inexact Newton algorithms . Computer Methods in Applied Mechanics and Engineering, 196(8):1603–1611, January 2007.

I. M. Idriss and J. I. Sun. SHAKE91: A Computer Program for Equivalent Linear Seismic Response Analyses of Horizontally Layered Soil Deposits. Center for Geotechnical Modeling Department of Civil & Environmental Engineering University of California Davis, California, 1992.

M. Iguchi and J. E. Luco. Dynamic response of flexible rectangular foundations on an elastic half-space. Earthquake Engineering and Structural Dynamics, 9(3):239–249, 1 1981.

M. Iida. Three-dimensional finite-element method for soil-building interaction based on an input wave field. International Journal of Geomechanics, 13(4):430–440, 2012.

A. Il'Iushin. On the postulate of plasticity. Journal of Applied Mathematics and Mechanics, 25(3): 746–752, 1961.

Y. Inada, T. Okawa, H. Mashimo, and Y. Kokudo. Failure characteristics of rock under compression at high and low temperatures. In J. Hudson, editor, Rock Characterization, pages 63–68. Thomas Telford Ltd., 1992.

K. Ishihara. Soil behaviour in earthquake geotechnics. Clarendon Press, Oxford University Press, 1996.

ISO-90003. ISO/IEC/IEEE 90003, software engineering – guidelines for the application of ISO 9001:2015 to computer software. Technical Report ISO/IEC/IEEE 90003:2018(E), ISO, 2018.

ISO/IEC/IEEE 90003 Developers, International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), and Institute of Electrical and Electronics Engineers (IEEE). International Standard ISO/IES/IEEE 90003, Software Engineering - Guidelines for applicaiton of ISO 9001:2015 to computer software. ISO/IEC/IEEE, first edition 2018-11 edition, November 2018.

K. A. Issen and J. W. Rudnicki. Theory of compaction bands in porous rock. Physics and Chemistry of the Earth, Part A: Solid Earth and Geodesy, 26(1–2):95–100, 2001.

J. Jaeger, N. Cook, and R. Zimmerman. Fundamentals of rock mechanics. Blackwell Pub., 2007. ISBN 9780632057597.

N. Janbu. Soil compressibility as determined by odometer and triaxial tests. In Proceedings of European Conference on Soil Mechanics and Foundation Engineering, pages 19–25, 1963.

Japanese Road Association. Specification for highway bridges, 1980.

Japanese Society of Civil Engineers. The report of damage investigation in the 1964 Niigata Earthquake. Japanese Soc. of Civil Engineers, Tokyo, 1966. in Japanese.

H. Jasak. Error analysis and estimation for finite volume method with applications to fluid flow. PhD thesis, Imperial College of Science, Technology and Medicine, 1996.

H. Jasak and Z. Tukovic. Automatic mesh motion for the unstructured finite volume method. Transactions of FAMENA, 30(2):1–20, 2006.

S. Y. Je, Y.-S. Chang, and S.-S. Kang. Dynamic characteristics assessment of reactor vessel internals with fluid-structure interaction. Nuclear Engineering and Technology, 2017.

B. Jeremić. Dynamic analysis of axisymmetric solids subjected to non-symmetric loading by the finite element method. Diploma thesis,, Faculty of Civil Engineering, Belgrade Universtity, july 1989. in Serbian. (Борис Јеремић Динамичка Анализа Ротационо Симетричних Тела оптерећених

Несиметричним Оптерећењима Методом Коначних Елемената, *Дипломски Рад* Грађевински Факултет Универзитета у Београду.

B. Jeremić. nDarray programming tool. Object Oriented Approach to Numerical Computations in Elastoplactity, Reference Manual University of Colorado at Boulder, December 1993.

B. Jeremić. Implicit integration rules in plasticity: Theory and implementation. Master's thesis, University of Colorado at Boulder, May 1994.

B. Jeremić. Finite Deformation Hyperelasto–Plasticity of Geomaterials. PhD thesis, University of Colorado at Boulder, July 1997.

B. Jeremić. Line search techniques in elastic–plastic finite element computations in geomechanics. Communications in Numerical Methods in Engineering, 17(2):115–125, January 2001.

B. Jeremić. Lectures notes on computational geomechanics: Inelastic finite elements for pressure sensitive materials. –, April 2004.

B. Jeremić. High fidelity modeling and simulation of sfs interaction: energy dissipation by design. In R. P. Orense, N. Chouw, and M. Pender, editors, Soil-Foundation-Structure Interaction, pages 125–132. CRC Press, Taylor & Francis Group, 2010. ISBN 978-0-415-60040-8 (Hbk); 978-0-203-83820-4 (eBook).

B. Jeremić and Z. Cheng. Numerical modeling and simulations of piles in liquefied soils. Soil Dynamics and Earthquake Engineering, in Print, 2009.

B. Jeremić and M. Preisig. Seismic soil - foundation - structure interaction: Numerical modeling issues. In SEI Structures Congress, 2005.

B. Jeremić and K. Sett. The influence of uncertain material parameters on stress-strain response. In P. V. Lade and T. Nakai, editors, Geomechanics II: Testing, Modeling, and Simulation (Proceedings of the Second-U.S. Workshop on Testing, Modeling, and Simulation in Geomechanics, held in Kyoto, Japan from September 8-10, 2005), Geotechnical Special Publication No. 156, pages 132–147. American Society for Civil Engineers, August 2006.

B. Jeremić and K. Sett. Uncertain soil properties and elastic-plastic simulations in geomechanics. In GeoDenver, 2007.

B. Jeremić and K. Sett. On probabilistic yielding of materials. Communications in Numerical Methods in Engineering, 25(3):291–300, 2009a.

B. Jeremić and K. Sett. On probabilistic yielding of materials. Communications in Numerical Methods in Engineering, 25(3):291–300, 2009b.

B. Jeremić and K. Sett. Uncertain seismic wave propagation in 1D. Soil Dynamics and Earthquake Engineering, 2010. in Review.

B. Jeremić and S. Sture. Implicit integrations in elasto–plastic geotechnics. International Journal of Mechanics of Cohesive–Frictional Materials, 2:165–183, 1997.

B. Jeremić and S. Sture. Implicit integrations in elastoplastic geotechnics. Mechanics of Cohesive-Frictional Materials, 2(2):165–183, 1997.

B. Jeremić and S. Sture. Tensor data objects in finite element programming. International Journal for Numerical Methods in Engineering, 41(1):113–126, 1998.

B. Jeremić and C. Xenophontos. Application of the p-version of the finite element method to elastoplasticity with localization of deformation. Communications in Numerical Methods in Engineering, 15 (12):867–876, December 1999.

B. Jeremić and Z. Yang. Template elastic–plastic computations in geomechanics. International Journal for Numerical and Analytical Methods in Geomechanics, 26(14):1407–1427, 2002.

B. Jeremić and Z. Yang. Template elastic-plastic computations in geomechanics. International Journal for Numerical and Analytical Methods in Geomechanics, 26(14):1407–1427, December 2002.

B. Jeremić, G. Jie, Z. Cheng, N. Tafazzoli, P. Tasiopoulou, F. Pisanò, J. A. Abell, K. Watanabe, Y. Feng, S. K. Sinha, F. Behbehani, H. Yang, H. Wang, and K. D. Staszewska. The Real-ESSI Simulator System. University of California, Davis, 1988-2025. http://real-essi.us/.

B. Jeremić, Z. Yang, Z. Cheng, G. Jie, N. Tafazzoli, M. Preisig, P. Tasiopoulou, F. Pisanò, J. Abell, K. Watanabe, Y. Feng, S. K. Sinha, F. Behbehani, H. Yang, H. Wang, and K. D. Staszewska. Nonlinear Finite Elements: Modeling and Simulation of Earthquakes, Soils, Structures and their Interaction. Self-Published-Online, University of California, Davis, CA, USA, 1989-2025. ISBN 978-0-692-19875-9. URL: http://sokocalo.engr.ucdavis.edu/~jeremic/LectureNotes/.

B. Jeremić, K. Runesson, and S. Sture. Large deformation constitutive integration algorithm. In Murakami and Luco, editors, Proceedings of the 12th Conference, pages 1029–1032, La Jolla, California, May 1998. Engineering Mechanics Division of the American Society of Civil Engineers.

B. Jeremić, N. Straz, M. Akers, and K. Makles. Beowulf class parallel computer "NorthCountry": Design, construction and testing. Report CEE-98-01, Clarkson University, Department of Civil and Environmental Engineering, May 1998. `http://sokocalo.sc.clarkson.edu/`

B. Jeremić, K. Runesson, and S. Sture. A model for elastic–plastic pressure sensitive materials subjected to large deformations. International Journal of Solids and Structures, 36(31/32):4901–4918, 1999.

B. Jeremić, Z. Yang, and M. Olton. Beowulf class parallel computer "GeoWulf": Design, construction and testing. Progress report, University of California, Davis, Department of Civil and Environmental Engineering, 1999. `http://sokocalo.engr.ucdavis.edu/~jeremic/GeoWulf`

B. Jeremić, S. Kunnath, and F. Xiong. Influence of soil–foundation–structure interaction on seismic response of the i-880 viaduct. Engineering Structures, 26(3):391–402, 2004.

B. Jeremić, K. Sett, and M. L. Kavvas. Probabilistic elasto-plasticity: formulation in 1D. Acta Geotechnica, 2(3):197–210, October 2007a.

B. Jeremić, K. Sett, and M. L. Kavvas. Probabilistic elasto-plasticity: Formulation in 1–D. Acta Geotechnica, 2(3):197–210, September 2007b.

B. Jeremić, Z. Cheng, M. Taiebat, and Y. F. Dafalias. Numerical simulation of fully saturated porous materials. International Journal for Numerical and Analytical Methods in Geomechanics, 32(13): 1635–1660, 2008.

B. Jeremić, G. Jie, M. Preisig, and N. Tafazzoli. Time domain simulation of soil–foundation–structure interaction in non–uniform soils. Earthquake Engineering and Structural Dynamics, 38(5):699–718, 2009.

P. Jetteur. Implicit integration algorithm for elastoplasticity in plane stress analysis. Engineering Computations, 3:251–253, 1986.

H. Jin, Y. Liu, and H.-J. Li. Experimental study on sloshing in a tank with an inner horizontal perforated plate. Ocean Engineering, 82:75–84, 2014.

S. Johnson. Objecting the objects. WTEC'94 Proceedings of the USENIX Winter 1994 Technical Conference, USENIX Winter 1994 Technical Conference, september 1994.

S. C. Johnson. Yacc: Yet another compiler-compiler. Technical report, AT&T Bell Laboratories, Murray Hill, New Jersey 07974, 1975. URL `http://dinosaur.compilertools.net/`.

S. Kacin, M. Ozturk, U. K. Sevim, B. A. Mert, Z. Ozer, O. Akgol, E. Unal, and M. Karaaslan. Seismic metamaterials for low-frequency mechanical wave attenuation. Natural Hazards, 107:213–229, May 2021.

E. Kalkan and S. Kunnath. Effective cyclic energy as a measure of seismic demand. Journal of Earthquake Engineering, 11(5):725–751, 2007.

E. Kalkan and S. Kunnath. Relevance of absolute and relative energy content in seismic evaluation of structures. Advances in Structural Engineering, 11(1):17–34, 2008.

L. F. Kallivokas, J. Bielak, and R. C. MacCamy. A simple impedance–infinite element for the finite element solution of the three–dimensional wave equation in unbound domains. Computer Methods in Applied Mechanics and Engineering, 147:235–262, 1997.

C. Kanellopoulos, N. Psycharis, H. Yang, B. Jeremić, I. Anastasopoulos, and B. Stojadinović. Seismic resonant metamaterials for the protection of an elastic-plastic sdof system against vertically propagating seismic shear waves (SH) in nonlinear soil. Soil Dynamics and Earthquake Engineering, 162:107366, 2022. ISSN 0267-7261. doi: https://doi.org/10.1016/j.soildyn.2022.107366. URL `https://www.sciencedirect.com/science/article/pii/S0267726122002159`.

K. Karhunen. Über lineare methoden in der wahrscheinlichkeitsrechnung. Ann. Acad. Sci. Fennicae. Ser. A. I. Math.-Phys., 37:1–79, 1947.

W. Karush. Minima of functions of several variables with inequalities as side constraints. Master's thesis, University of Chicago, Chicago, IL., 1939.

G. Karypis and V. Kumar. Multilevel k-way partitioning scheme for irregular graphs. Journal of Parallel and Distributed Computing, 48(1), 1998a.

G. Karypis and V. Kumar. A fast and highly quality multilevel scheme for partitioning irregular graphs. SIAM Journal on Scientific Computing, 1998b. A short version appears in Intl. Conf. on Parallel Processing 1995.

G. Karypis and V. Kumar. METIS A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices Version 4.0. Army HPC Research Center, Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, September 1998c.

G. Karypis and V. Kumar. METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices, Version 4.0. Army HPC Research

Center, Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, September 1998d.

G. Karypis, K. Schloegel, and V. Kumar. PARMETIS Parallel Graph Partitioning and Sparse Matrix Ordering Library Version 3.1. Army HPC Research Center, Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, August 2003.

E. Kausel. Fundamental Solutions in Elastodynamics, A Compendium. Cambridge University Press, The Edinburgh Building, Cambridge CB2 2RU, UK, 2006. ISBN ISBN-10: 0521855705; ISBN-13: 978-0521855709.

E. Kausel and J. M. Roesset. Dynamic stiffness of circular foundations. Journal of the Engineering Mechanics Division , ASCE, 101(6):771–785, 1975.

M. L. Kavvas. Nonlinear hydrologic processes: Conservation equations for determining their means and probability distributions. Journal of Hydrologic Engineering, 8(2):44–53, March 2003.

M. L. Kavvas and A. Karakas. On the stochastic theory of solute transport by unsteady and steady groundwater flow in heterogeneous aquifers. Journal of Hydrology, 179:321–351, 1996.

G. Kaypis and V. Kumar. Multilevel k-way partitioning scheme for irregular graphs. Journal of Parallel and Distributed Computing, 48(1):96–129, 1998.

A. Keese. A review of recent developments in the numerical solution of stochastic partial differential equations (stochastic finite elements). Scientific Computing 2003-06, Deapartment of Mathematics and Computer Science, Technical University of Braunschweig, Brunswick, Germany, 2003.

A. Keese and H. G. Matthies. Efficient solvers for nonlinear stochastic problem. In H. A. Mang, F. G. Rammmerstorfer, and J. Eberhardsteiner, editors, Proceedings of the Fifth World Congress on Computational Mechanics, July 7-12, 2002, Vienna, Austria, http://wccm.tuwien.ac.at/publications/Papers/fp81007.pdf, 2002.

J. Kelly and S. Hodder. Experimental study of lead and elastomeric dampers for base isolation systems in laminated neoprene bearings. Bulletin of the New Zealand National Society for Earthquake Engineering,, 15(2):53–67, 1982.

J. M. Kelly. A long-period isolation system using low-damping isolators for nuclear facilities at soft soil sites. Technical Report UCB/EERC-91/03, Earthquake Engineering Research Center, University of California at Berkeley, March 1991a.

J. M. Kelly. Dynamic and failure characteristics of Bridgestone isolation bearings. Technical Report UCB/EERC-91/04, Earthquake Engineering Research Center, University of California at Berkeley, March 1991b.

D. Kent and R. Park. Flexural members with confined concrete. ASCE Journal of Structural Division, 97:1969–1990, 1971.

M. Kimura, T. Adachi, H. Kamei, and F. Zhang. 3-D finite element analyses of the ultimate behavior of laterally loaded cast-in-place concrete piles. In G. N. Pande and S. Pietruszczak, editors, Proceedings of the Fifth International Symposium on Numerical Models in Geomechanics, NUMOG V, pages 589–594. A. A. Balkema, September 1995.

H. Kishida and M. Uesugi. Tests of the interface between sand and steel in the simple shear apparatus. Geotechnique, 37(1):45–52, 1987.

O. Klaas, M. Kreienmeyer, and E. Stein. Elastoplastic finite element analysis on a MIMD parallel–computer. Engineering Computations, 11:347–355, 1994.

M. Kleiber and T. D. Hien. The Stochastic Finite Element Method: Basic Perturbation Technique and Computer Implementation. John Wiley & Sons, Baffins Lane, Chichester, West Sussex PO19 1UD , England, 1992.

C. Knott. Reflection and refraction of elastic waves with seismological applications. Philosophical Magazine, 48:64–97, 1899.

D. E. Knuth. Literate programming. Computer Journal (British Computer Society), 27(2):97–111, 1984.

A. Koenig. C++ columns. Journal of Object Oriented Programming, 1989 - 1993.

W. T. Koiter. Stress - strain relations, uniqueness and variational theorems for elastic - plastic materials with singular yield surface. Quarterly of Applied Mathematics, 2:350–354, 1953.

W. T. Koiter. General theorems for elastic-plastic solids. In I. N. Sneddon and R. Hill, editors, Progress in Solid Mechanics, pages 165–221. North Holland, 1960.

M. Kojić. Computational Procedures in Inelastic Analysis of Solids and Structures. Center for Scientific Research of Srbian Academy of Sciences and Arts and University of Kragujevac and Faculty of Mechanical Engineering University of Kragujevac, 1997. ISBN 86-82607-02-6.

Милош. Којић. . M. Kojić. Општи концепт имплицитне интеграције конститутивних релација при нееластичном *Монографија Центра за научна истраживања Српске академије наука и уметности и Универзитета у Крагујевцу* , 1993. (Miloš Kojić, A General Concept of Implicit Integration of Constitutive Relations for Inelastic Material Deformation, in Serbian).

K. Kolozvari, K. Orakcal, and J. Wallace. Shear-flexure interaction modeling for reinforced concrete structural walls and columns under reversed cyclic loading. Technical Report PEER Report No. 2015/12, Pacific Earthquake Engineering Research Center, Pacific Earthquake Engineering Research Center, Richmod, CA, December 2015.

S. Koutsourelakis, J. H. Prevost, and G. Deodatis. Risk assesment of an interacting structure-soil system due to liquefaction. Earthquake Engineering and Structural Dynamics, 31:851–879, 2002.

S. L. Kramer. Geotechnical Earthquake Engineering. Prentice Hall, Inc, Upper Saddle River, New Jersey, 1996a.

S. L. Kramer. Geotechnical Earthquake Engineering. Prentice Hall, New Jersey, 1996b.

S. Krödel, N. Thomé, and C. Daraio. Wide band-gap seismic metastructures. Extreme Mechanics Letters, 4:111–117, 2015. ISSN 2352-4316. doi: https://doi.org/10.1016/j.eml.2015.05.004. URL https://www.sciencedirect.com/science/article/pii/S2352431615000772.

E. Kröner. Algemeine kontinuumstheorie der versetzungen und eigenspanungen. Archive for Rational Mechanics and Analysis, 4(4):273–334, 1960.

P. Krysl and Z. Bittnar. Parallel explicit finite element solid dynamics with domain decomposition and message passing: dual partitioning scalability. Computers and Structures, 79:345–360, 2001.

R. Kubo. Stochastic Liouville equations. J. Math. Phys., 4(2):174–183, 1963.

H. W. Kuhn and A. W. Tucker. Nonlinear programming. In J. Neyman, editor, Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, pages 481 – 492. University of California Press, July 31 – August 12 1950 1951.

M. Kumar, A. S. Whittaker, and M. C. Constantinou. An advanced numerical model of elastomeric seismic isolation bearings. Earthquake Engineering & Structural Dynamics, 43(13):1955–1974, 2014. ISSN 1096-9845. doi: 10.1002/eqe.2431. URL http://dx.doi.org/10.1002/eqe.2431.

H. Kupfer, H. K. Hilsdorf, and H. Rusch. Behavior of concrete under biaxial stresses. In Journal Proceedings, volume 66/8, pages 656–666, 1969.

A. Kurtulus, J. J. Lee, and K. H. Stokoe. Summary report — site characterization of capital aggregates test site. Technical report, Department of Civil Engineering, University of Texas at Austin, 2005.

W.-P. Kwan and S. Billington. Simulation of structural concrete under cyclic load. Journal of Structural Engineering, 127(12):1391–1401, 2001.

J. Labuz and L. Biolzi. Experiments with rock: Remarks on strength and stability issues. International Journal of Rock Mechanics and Mining Sciences, 44(4):525–537, June 2007.

S. Lacasse and F. Nadim. Uncertainties in characterizing soil properties. In C. D. Shackelford and P. P. Nelson, editors, Uncertainty in Geologic Environment: From Theory to Practice, Proceedings of Uncertainty 1996, July 31-August 3, 1996, Madison, Wisconsin, volume 1 of Geotechnical Special Publication No. 58, pages 49–75. ASCE, New York, 1996.

P. V. Lade. Elastoplastic stress strain theory for cohesionless soil with curved yield surfaces. International Journal of Solids and Structures, 13:1019–1035, 1977.

P. V. Lade. Model and parameters for the elastic behavior of soils. In Swoboda, editor, Numerical Methods in Geomechanics, pages 359–364, Innsbruck, 1988a. Balkema, Rotterdam.

P. V. Lade. Effects of voids and volume changes on the behavior of frictional materials. International Journal for Numerical and Analytical Methods in Geomechanics, 12:351–370, 1988b.

P. V. Lade. Instability, shear banding, and failure in granular materials. International Journal of Solids and Structures, 39(13):3337–3357, 2002.

P. V. Lade and R. B. Nelson. Modeling the elastic behavior of granular materials. International Journal for Numerical and Analytical Methods in Geomechanics, 4, 1987.

I. P. Lam and H. K. Law. Soil–foundation–structure-interaction analytical considerations by empirical p–y methods. In The fourth Caltrans Seismic Research Workshop, Sacramento, CA, July 1996. California Dept. of Transportation, Engineering Service Center.

H. Lamb. Hydrodynamics. Cambridge university press, 1932.

W. T. Lambe and R. V. Whitman. Soil Mechanics, SI Version. John Wiley and Sons, 1979.

H. Langtangen. A general numerical solution method for Fokker-Planck equations with application to structural reliability. Probabilistic Engineering Mechanics, 6(1):33–48, 1991.

D. B. Larson and G. D. Anderson. Plane shock wave studies of porous geologic media. Journal of Geophysical Research, 84(B9):357–363, 1979.

D. B. Larson and G. D. Anderson. Plane shock wave studies of westerly granite and nugget sandstone. International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts, 17(6): 357–363, 1980.

P. Le Tallec and J. Mouro. Fluid structure interaction with large structural displacements. Computer methods in applied mechanics and engineering, 190(24):3039–3067, 2001.

R. Lebrun and A. Dutfoy. An innovating analysis of the Nataf transformation from the copula viewpoint. Probabilistic Engineering Mechanics, 24(3):312–320, 2009.

E. H. Lee. Elastic–plastic deformation at finite strains. Journal of Applied Mechanics, 36(1):1–6, 1969.

E. H. Lee and D. T. Liu. Finite–strain elastic–plastic theory with application to plane–wave analysis. Journal of Applied Physics, 38(1):19–27, January 1967.

J. Lee and G. L. Fenves. Plastic-damage model for cyclic loading of concrete structures. Journal of engineering mechanics, 124(8):892–900, 1998.

L. C. Lee. Wave propagation in a random medium: A complete set of the moment equations with different wavenumbers. Journal of Mathematical physics, 15(9):1431–1435, September 1974.

S. L. Lee and G. P. Karunaratne. Laterally loaded piles in layered soil. Soils and Foundations, 27(4): 1–10, Dec. 1987.

P. Léger and S. Dussault. Seismic-energy dissipation in mdof structures. Journal of Structural Engineering, 118(5):1251–1269, 1992.

F. Lehmann, F. Gatti, M. Bertin, and D. Clouteau. FOURIER NEURAL OPERATOR SURROGATE MODEL TO PREDICT 3D SEISMIC WAVES PROPAGATION. arXiv - PHYS - Geophysics, 2023. doi: 10.48550/arXiv.2304.10242.

Q. Lei and N. Barton. On the selection of joint constitutive models for geomechanics simulation of fractured rocks. Computers and Geotechnics, 145:104707, 2022. ISSN 0266-352X. doi: https://doi.org/10.1016/j.compgeo.2022.104707. URL https://www.sciencedirect.com/science/article/pii/S0266352X22000714.

J. Lemaitre and J. Chaboche. Mechanics of Solid Materials. Cambridge University Press, 1990. ISBN 0 521 47758 1 ; TA405.L3813 1990.

M. Lesk and E. Schmidt. Lex - a lexical analyzer generator. Technical Report UNIX TIME-SHARING SYSTEM:UNIX PROGRAMMER'S MANUAL, Seventh Edition, Volume 2B, AT&T Bell Laboratories, Murray Hill, New Jersey 07974, 1975. URL `http://dinosaur.compilertools.net/`.

J. Li and J. Bing Chen. Dynamic response and reliability analysis of structures with uncertain parameters. International Journal for Numerical Methods in Engineering, 62(2):289–315, 2005.

J. Li and O. B. Widlund. On the use of inexact subdomain solvers for BDDC algorithms . Computer Methods in Applied Mechanics and Engineering, 196(8):1415–1428, January 2007.

X. S. Li and J. W. Demmel. Making sparse Gaussian elimination scalable by static pivoting. In Proceedings of SC98: High Performance Networking and Computing Conference, Orlando, Florida, November 7–13 1998.

X. S. Li and J. W. Demmel. SuperLU_DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems. ACM Trans. Mathematical Software, 29(2):110–140, June 2003.

X. S. Li and Y. Wang. Linear representation of steady-state line for sand. Journal of Geotechnical and Geoenvironmental Engineering, 124(12):1215–1217, 1998.

S. Liao and A. Zerva. Physically compliant, conditionally simulated spatially variable seismic ground motions for performance–based design. Earthquake Engineering & Structural Dynamics, 35:891–919, 2006.

C. M. Linton and P. McIver. Handbook of mathematical techniques for wave/structure interactions. CRC Press, 2001.

I. Littleton. An experimental study of the adhesion between clay and steel. Journal of terramechanics, 13(3):141–152, 1976.

J. W. H. Liu. Modification of the minimum degree algorithm by multiple elimination. ACM Transactions on Mathematical Software, 11(2):141–153, June 1985.

L. Liu and R. Dobry. Effect of liquefaction on lateral response of piles by centrifuge model tests. In NCEER Bulletin, volume 9:1, pages 7–11. National Center for Earthquake Engineering Research, January 1995.

P. L. Liu and A. Der Kiureghian. A finite element reliability of geometrically nonlinear uncertain structures. Journal of Engineering Mechanics, 117(8):1806–1825, 1991.

D. S. Liyanapathirana and H. G. Poulos. Pseudostatic approach for seismic analysis of piles in liquefying soil. Journal of Geotechnical and Geoenvironmental Engineering, 131(12):1480–1487, 2005.

K. Lo, T. Yung, and B. Lukajic. A field method for the determination of rock-mass modulus. Canadian Geotechnical Journal,, 24:406–413, 1987.

D. A. Lockner and S. A. Stanchits. Undrained poroelastic response of sandstones to deviatoric stress change. JOURNAL OF GEOPHYSICAL RESEARCH, 107(B12):ETG 13–1 – ETG 13–14, 2002. doi:10.1029/2001JB001460.

M. Loève. Fonctions aleatories du second ordre. Supplement to P. Levy, Processus Stochastic et Mouvement Brownien, Gauthier-Villars, Paris, 1948.

I. N. Lomov, M. Hiltl, O. Y. Vorobiev, and L. A. Glenn. Dynamic behavior of berea sandstone for dry and water-saturated conditions. International Journal of Impact Engineering, 26(1-10):465–474, December 2001.

B. Loret and E. Rizzi. Anisotropic stiffness degradation triggers onset of strain localization. International Journal of Plasticity, 13(5):447–459, 1997.

M. Lou, H. Wang, X. Chen, and Y. Zhai. Structure–soil–structure interaction: literature review. Soil Dynamics and Earthquake Engineering, 31(12):1724–1731, 2011.

A. Love. Some problems of geodynamics. Some Problems of Geodynamics Publisher: Cambridge University Press, Cambridge, 1911, 1, 1911.

X. Lu, J. P. Bardet, and M. Huang. Numerical solutions of strain localization with nonlocal softening plasticity. Computer Methods Applied Mechanics and Engineering, 198:3702–3711, 2009.

J. Lubliner. On the thermodynamic foundations of non-linear solid mechanics. International Journal of Non-Linear Mechanics, 7:237–254, 1972.

J. Lubliner. Plasticity Theory. Macmillan Publishing Company, New York., 1990.

J. E. Luco. Impedance functions for a rigid foundation on a layered medium. Nuclear Engineering and Design, 31:204–217, 1974.

J. E. Luco, H. L. Wong, and F. C. P. D. Barros. Three-dimensional response of a cylindrical canyon in a layered half-space. Earthquake Engineering & Structural Dynamics, 19(6):799–817, 1990.

P. Lumb. The variability of natural soils. Canadian Geotechnical Journal, 3:74–97, 1966.

J. Lysmer. SASSI: A computer program for dynamic soil structure interaction analysis. Report UBC/GT81-02. University of California, Berkeley, CA, USA., 1988.

J. Lysmer and R. L. Kuhlemeyer. Finite dynamic model for infinite media. Journal of Engineering Mechanics Division, ASCE, 95(EM4):859–877, 1969.

P. C. Lysne. A comparison of calculated and measured low-stress Hugoniots and release adiabats of dry and water-saturated tuff. Journal of Geophysical Research, 75(23):4375–4386, 1970.

M.A.Biot. Theory of propagation of elastic waves in a fluid-saturated porous solid. low-frequency range. The journal of acoustical society of America, 28(2):168–178, March 1956.

R. I. Mackie. Object-Oriented Methods - Finite Element Programming and Engineering Software Design. In Computing in Civil and Building Engineering: Proceedings of the Sixth International Conference on Computing in Civil and Building Engineering, Berlin, Germany, July 12-15 1995.

A. Magrin. MULTI-SCALE SEISMIC HAZARD SCENARIOS. PhD thesis, UNIVERSITÀ DEGLI STUDI DI TRIESTE, 2012.

A. Magrin, A. A. Gusev, F. Romanelli, F. Vaccari, and G. F. Panza. Broadband NDSHA computations and earthquake ground motion observations for the Italian territory. Int. J. Earthquake and Impact Engineering, 1(1/2):131–158, 2016.

L. E. Malvern. Introduction to the Mechanics of a Continuous Medium. In Engineering of the Physical Sciences. Prentice Hall Inc., 1969.

E. J. Malvick, B. L. Kutter, R. W. Boulanger, and R. Kulasingam. Shear localization due to liquefaction–induced void redistribution in a layered infinite slope. ASCE Journal of Geotechnical and Geoenvironmental Engineering, 132(10):1293–1303, 2006.

J. Mandel and C. R. Dohrmann. Convergence of a balancing domain decomposition by constraints and energy minimization. Numerical Linear Algebra with Applications, 10(7):639–659, 2003.

J. B. Mander, M. J. Priestley, and R. Park. Theoretical stress-strain model for confined concrete. Journal of structural engineering, 114(8):1804–1826, 1988.

T. Manteuffel. An incomplete factorization technique for posititive definite linear systems. Mathematics of Computation, 34:473–497, 1980.

M. T. Manzari and Y. F. Dafalias. A critical state two–surface plasticity model for sands. Géotechnique, 47(2):255–272, 1997.

M. T. Manzari and R. Prachathananukit. On integartion of a cyclic soil plasticity model. International Journal for Numerical and Analytical Methods in Geomechanics, 25:525–549, 2001.

L. Marconato, P. Leloup, C. Lasserre, R. Jolivet, S. Caritg, R. Grandin, M. Métois, O. Cavalié, and L. Audin. Insights on fault reactivation during the 2019 November 11, Mw 4.9 Le Teil earthquake in southeastern France, from a joint 3-D geological model and InSAR time-series analysis. Geophys. J. Int., 229:758–775, 2022. doi: 10.1093/gji/ggab498.

K. T. Marosi and D. R. Hiltunen. Characterization of spectral analysis of surface waves shear wave velocity measurement uncertainty. Journal of Geotechnical and Geoenvironmental Engineering, 130 (10):1034–1041, October 2004.

J. M. M. C. Marques. Stress computations in elastoplasticity. Engineering Computations, 1:42–51, 1984.

J. E. Marsden and T. J. R. Hughes. Mathematical Foundations of Elasticity. Prentice Hall Inc,, 1983. local CM65; 4QA 931.M42 ; ISBN 0-13-561076-1.

G. R. Martin, M. L. March, D. G. Anderson, R. L. Mayes, and M. S. Power. Recommended design approach for liquefaction induced lateral spreads. In Proceedings of the 3rd National Seismic Conf and Workshop on Bridges and highways, MCEER–02–SP04, Buffalo, NY, 2002.

J. Martin and W. Moyce. An experimental study of the collapse of liquid columns on a rigid horizontal plane. Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences, pages 312–324, 1952.

A. Martinez, J. D. Frost, and G. L. Hebeler. Experimental study of shear zones formed at sand/steel interfaces in axial and torsional axisymmetric tests. Geotechnical Testing Journal, 38(4):409–426, 2015.

J. Mason, A. Rosakis, and G. Ravichandran. On the strain and strain rate dependence of the fraction of plastic work converted to heat: an experimental study using high speed infrared detectors and the kolsky bar. Mechanics of Materials, 17(2-3):135–145, 1994.

A. Masud and L. A. Bergman. Application of multi-scale finite element methods to the solution of the Fokker-Planck equation. Computer Methods in Applied Mechanics and Engineering, 194(1):1513–1526, April 2005.

H. Matsuoka and T. Nakai. Stress, deformation and strength characteristics under three different principal stresses. Proceedings of Japanese Society of Civil Engineers, 232:59–70, 1974.

H. G. Matthies, C. E. Brenner, C. G. Bucher, and C. G. Soares. Uncertainties in probabilistic numerical analysis of structures and soilds - stochastic finite elements. Structural Safety, 19(3):283–336, 1997.

G. P. Mavroeidis and A. S. Papageorgiou. A mathematical representation of near-fault ground motions. Bulletin of the Seismological Society of America, 93(3):1099–1131, June 2003.

D. Mayer-Rosa and B. Cadiot. A review of the 1356 Basel earthquake: Basic data,. Tectonophysics, 53(3):325–333, 1979. doi: https://doi.org/10.1016/0040-1951(79)90077-5. URL https://www.sciencedirect.com/science/article/pii/0040195179900775. Proceedings of the 16th General Assemble of the European Seismological Commission.

G. G. Mayerhoff. Development of geotechnical limit state design. In Proceedingd of International Symposium on Limit State Design in Geotechnical Engineering, Cpenhagen, pages 1–12. ASCE, ASCE, New York, 1993.

I. Mazzieri, M. Stupazzini, R. Guidotti, and C. Smerzini. SPEED: SPectral Elements in Elastodynamics with Discontinuous Galerkin: a non-conforming approach for 3D multi-scale problems. Int. J. Numer. Meth. Engng, 2013. doi: 10.1002/nme.4532.

S. Mazzoni, F. McKenna, M. H. Scott, G. L. Fenves, and B. Jeremić. Open System for Earthquake Engineers Simulation (OpenSees) : User Manual. Pacific Earthquake Engineering Research Center, Richmond, December 2002.

F. T. McKenna. Object Oriented Finite Element Programming: Framework for Analysis, Algorithms and Parallel Computing. PhD thesis, University of California, Berkeley, 1997.

F. T. McKenna. Object-Oriented Finite Element Programming: Frameworks for Analysis, Algorithms and Parallel Computing. PhD thesis, University of California, Berkeley, 1997.

M. McVay, R. Casper, , and T.-I. Shang. Lateral response of three-row groups in loose to dense sands at 3D and 5D pile spacing. Journal of Geotechnical Engineering, 121(5):436–441, May 1995.

M. McVay, L. Zhang, T. Molnit, and P. Lai. Centrifuge testing of large laterally loaded pile groups in sands. Journal of Geotechnical and Geoenvironmental Engineering, 124(10):1016–1026, October 1998.

J. A. Meijerink and H. A. van der Vorst. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. Mathematics of Computation, 31:148–162, 1977.

L. Mejia and E. Dawson. Earthquake deconvolution for flac. In 4th International FLAC Symposium on Numerical Modeling in Geomechanics, pages 04–10, 2006.

R. Mellah, G. Auvinet, and F. Masrouri. Stochastic finite element method applied to non-linear analysis of embankments. Probabilistic Engineering Mechanics, 15:251–259, 2000.

M. Menegotto and P. E. Pinto. Method of analysis for cyclically loaded reinforced clncrete plane frames including changes in geometry and non-elastic behaviour of elements under combined normal force and bending. In Proceedings of IABSE Symposium, pages 15–22, 1973.

P. Menéntrey and T. Zimmermann. Object–oriented non–linear finite element analysis: Application to J2 plasticity. Computers and Structures, 49(5):767–77, 1993.

M. Mernik, J. Heering, and A. M. Sloane. When and how to develop domain-specific languages. ACM Computing Surveys, 37(4):316–344, December 2005. `http://doi.acm.org/10.1145/1118890.1118892`.

M. G. Mezgebo and E. M. Lui. A new methodology for energy-based seismic design of steel moment frames. Earthquake Engineering and Engineering Vibration, 16(1):131–152, 2017.

C. Militello and C. A. Felippa. The first ANDES elements: 9-dof plate bending triangles. Computer Methods in Applied Mechanics and Engineering, 93:217–246, 1991.

G. R. Miller. An object – oriented approach to structural analysis and design. Computers and Structures, 40(1):75–82, 1991.

G. R. Miller and M. D. Rucki. A Program Architecture for Interactive Nonlinear Dynamic Analysis of Structures. In Computing in Civil and Building Engineering: Proceedings of the Fifth International Conference V_ICCCBE, Anaheim, CA, June 7–9 1993.

R. D. Mindlin and H. Deresiewicz. Elastic spheres in contact under varying oblique forces. ASME Journal of Applied Mechanics, 53(APM–14):327–344, September 1953.

M. Miniaci, A. Krushynska, F. Bosia, and N. Pugno. Large scale mechanical metamaterials as seismic shields. New Journal of Physics, 18:083041, 08 2016. doi: 10.1088/1367-2630/18/8/083041.

G. P. Mitchell and D. R. J. Owen. Numerical solution for elasto - plastic problems. Engineering Computations, 5:274–284, 1988.

F. Miura, H. E. Stewart, and T. D. O'Rourke. Nolinear analysis of piles subjected to liquefaction induced large ground deformation. In Proceedings of the Second U.S.–Japan Workshop on Liquefaction , Large Ground Deformations and Their Effect on Lifelines, number 0032 in NCEER-89, Sept. 26-29 1989.

K. Mogi. Experimental Rock Mechanics. Taylro and Francis, 2006. ISBN ISBN-10: 0415394430.

B. Monien, R. Preis, and R. Diekmann. Quality matching and local improvement for multilevel graph-partitioning. Technical report, University of Paderborn, 1999.

D. C. Montgomery and G. C. Runger. Applied Statistics and Probability for Engineers. John Wiley & Sons, 605 Third Avenue, New York, NY 10158, third edition, 2003.

R. Moore. Methods and Applications of Interval Analysis. SIAM, Philadelphia, 1979.

K. N. Morman. The generalized strain meassures with application to nonhomogeneous deformation in rubber–like solids. Journal of Applied Mechanics, 53:726–728, 1986.

G. Mortara, A. Mangiola, and V. N. Ghionna. Cyclic shear stress degradation and post-cyclic behaviour from sand–steel interface direct shear tests. Canadian Geotechnical Journal, 44(7):739–752, 2007.

Y. E. Mostafa and M. H. El Naggar. Dynamic analysis of laterally loaded pile groups in sand and clay. Canadian Geotechnical Journal, 39(6):1358–1383, 2002.

F. Moukalled, L. Mangani, M. Darwish, et al. The finite volume method in computational fluid dynamics. Springer, 2016.

A. Moustafa. Damage-based design earthquake loads for single-degree-of-freedom inelastic structures. Journal of Structural Engineering, 137(3):456–467, 2011.

A. Moustafa and S. Mahmoud. Damage assessment of adjacent buildings under earthquake loads. Engineering Structures, 61:153–165, 2014.

Z. Mróz, V. A. Norris, and O. C. Zienkiewicz. An anisotropic hardening model for soils and its application to cyclic loading. International Journal for Numerical and Analytical Methods in Geomechanics, 2: 203–221, 1978.

D. Mu, H. Shu, L. Zhao, and S. An. A review of research on seismic metamaterials. Advanced Engineering Materials, 22(4):1901148, 2020. doi: https://doi.org/10.1002/adem.201901148. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/adem.201901148.

D. Muir Wood. Soil Behaviour and Critical State Soil Mechanics. Cambridge University Press, 1990.

A. Muqtadir and C. S. Desai. Three dimensional analysis of a pile-group foundation. International journal for numerical and analysis methods in geomechanics, 10:41–58, 1986.

C. M. Murea and S. Sy. Updated lagrangian/arbitrary lagrangian–eulerian framework for interaction between a compressible neo-hookean structure and an incompressible fluid. International Journal for Numerical Methods in Engineering, 109(8):1067–1084, 2017.

NASA. Standard for models and simulations, NASA-STD-7009, July 2008. URL https://standards.nasa.gov/standard/nasa/nasa-std-7009.

NASA. Standard for models and simulations, NASA-STD-7009A-Change-1, December 2016. URL https://standards.nasa.gov/standard/nasa/nasa-std-7009.

National Research Council. Liquefaction of soils during earthquakes. National Academy Press, Washington, D.C., 1985.

NAVFAC. Foundations & Earth Structures, DESIGN MANUAL 7.02. Naval Facilities Engineering Command, 200 Stovall Street, Alexandria, Virginia 22332-2300, USA, revalidated by change 1 september 1986 edition, 1986.

G. C. Nayak and O. C. Zienkiewicz. Elasto - plastic stress analysis a generalization for various constitutive relations including strain softening. International Journal for Numerical Methods in Engineering, 5: 113–135, 1972.

M. Nehdi, M. S. Alam, and M. Youssef. Development of corrosion-free concrete beam–column joint with adequate seismic energy dissipation. Engineering Structures, 32(9):2518–2528, 2010.

S. Nemat–Nasser. On finite plastic flow of crystalline solids and geomaterials. Journal of Applied Mechanics, 50:1114–1126, 1983.

M. Nenning and M. Schanz. Infinite elements in a poroelastodynamic fem. INTERNATIONAL JOURNAL FOR NUMERICAL AND ANALYTICAL METHODS IN GEOMECHANICS, Early View DOI: 10.1002/nag.980, 2010.

N. M. Newmark. A method of computation for structural dynamics. ASCE Journal of the Engineering Mechanics Division, 85:67–94, July 1959.

C. W. W. Ng, L. Zhang, and D. C. N. Nip. Response of laterally loaded large-diameter bored pile groups. Journal of Geotechnical and Geoenvironmental Engineering, 127(8):658–669, Aug. 2001.

E. Ng and B. Peyton. Block sparse cholesky algorithms on advanced uniprocessor computers. SIAM Journal on Scientific and Statistical Computing, 14(5):1034–1056, September 1993.

E. Niebler. xpressive: Dual-mode dsel library design. In D. Musser, editor, Library-Centric Software Design LCSD'05. Object-Oriented Programming, Systems, Languages and Applications, October 2005.

A. Niemunis. Private correspondence, 2015 –.

E. Nikbakht, K. Rashid, F. Hejazi, and S. A. Osman. A numerical study on seismic response of self-centring precast segmental columns at different post-tensioning forces. Latin American Journal of solids and structures, 11(5):864–883, 2014.

V. N. Nikolaevskiy, S. M. Kapustyanskiy, M. Thiercelin, and A. G. Zhilenkov. Explosion dynamics in saturated rocks and solids. Transport in Porous Media, 65:485–504, 2006. URL `http://www.springerlink.com/index/R221288861444442.pdf`. DOI 10.1007/s11242-006-6752-0.

A. Nobahar. Effects of Soil Spatial Variability on Soil-Structure Interaction. Doctoral dissertation, Memorial University, St. John's, NL, 2003.

A. K. Noor, A. Kamel, and R. E. Fulton. Substructuring techniques – status and projections. Computers & Structures, 8:628–632, 1978.

R. Nova and D. M. Wood. A constitutive model for sand in triaxial compression. International Journal for Numerical and Analytical Methods in Geomechanics, 3:255–278, 1979.

V. D. Novellis, V. Convertito, S. Valkaniotis, F. Casu, R. Lanari, M. F. M. Tobar, and N. A. Pino. Coincident locations of rupture nucleation during the 2019 Le Teil earthquake, France and maximum stress change from local cement quarrying. Communications Earth & Environment, 1(20), 2020. doi: https://doi.org/10.1038/s43247-020-00021-6.

W. Nowacki. Baudynamik. Ingenieurbauten. Springer, Vienna, 1974.

W. Oberkampf. Material from the short course on verification and validation in computational mechanics. Albuquerque, New Mexico, July 2003.

W. L. Oberkampf and M. Pilch. Simulation Verification and Validation for Managers. NAFEMS, 2017. ISBN 978-1-910643-33-4.

W. L. Oberkampf and C. J. Roy. Verification and Validation in Scientific Computing. Cambridge University Press, 2010. ISBN 978-0-521-11360-1.

W. L. Oberkampf and T. G. Trucano. Verification and validation benchmarks. Nuclear engineering and Design, 238(3):716–743, 2008. URL `http://www.sciencedirect.com/science/article/pii/S0029549307003548`.

W. L. Oberkampf, T. G. Trucano, and C. Hirsch. Verification, validation and predictive capability in computational engineering and physics. In Proceedings of the Foundations for Verification and Validation on the 21st Century Workshop, pages 1–74, Laurel, Maryland, October 22-23 2002. Johns Hopkins University / Applied Physics Laboratory.

W. L. Oberkampf, M. M. Pilch, and T. G. Trucano. Predictive capability maturity model for computational modeling and simulation. OSTI, (SAND2007-5948, TRN: US20100910 2007. doi: 10.2172/976951. URL https://www.osti.gov/biblio/976951.

J. T. Oden, I. Babuška, F. Nobile, Y. Feng, and R. Tempone. Theory and methodology for estimation and control of errors due to modeling, approximation, and uncertainty. Computer Methods in Applied Mechanics and Engineering, 194(2-5):195–204, February 2005.

T. Oden, R. Moser, and O. Ghattas. Computer predictions with quantified uncertainty, part i. SIAM News,, 43(9), November 2010a.

T. Oden, R. Moser, and O. Ghattas. Computer predictions with quantified uncertainty, part ii. SIAM News,, 43(10), December 2010b.

R. W. Ogden. Non–Linear Elastic Deformations. Series in mathematics and its applications. Ellis Horwood Limited, Market Cross House, Cooper Street, Chichester, West Sussex, PO19 1EB, England, 1984.

W. A. Olsson. Theoretical and experimental investigation compaction of bands in porous rock. JOURNAL OF GEOPHYSICAL RESEARCH,, 104(B4):7219–7228, April 1999.

W. A. Olsson. Quasistatic propagation of compaction fronts in porous rock. Mechanics of Materials, 33 (11):659–668, 2001.

OpenCFD Ltd. OpenFOAM User Guide, v1906 edition, 2019.

OpenSees Development Team (Open Source Project). OpenSees: open system for earthquake engineering simulations. http://opensees.berkeley.edu/, 2000-2006.

I. Oprsal and D. Fäh. 1D vs 3D strong ground motion hybrid modelling of site, and pronounced topography effects at augusta raurica, switzerland-earthquakes or battles. In Proceedings of 4th International Conference on Earthquake Geotechnical Engineering June 25–28, 2007, Greece, 2007.

F. M. T. D. O'Rourke. Lateral spreading effects on pile foundations. In Proceedings of the Third U.S.–Japan Workshop on Earthquake Resistant Design of Lifeline Facilities and Countermeasures for Soil Liquefaction, number 0001 in NCEER-91, Feb. 1 1991.

M. Ortiz and E. P. Popov. Accuracy and stability of integration algorithms for elastoplastic constitutive relations. International Journal for Numerical Methods in Engineering, 21:1561–1576, 1985.

S. Osovski, D. Rittel, and A. Venkert. The respective influence of microstructural and thermal softening on adiabatic shear localization. Mechanics of Materials, 56:11–22, 2013.

F. Ostadan. Sassi2000 a system for analysis of soil-structure interaction, version 3, user's manual. Program Manual, April 2007.

W. Ostermann, W. Wunderlich, and H. Cramer. Object-Oriented Tools for the Development of User Interface for Interactive Teachware. In Computing in Civil and Building Engineering: Proceedings of the Sixth International Conference on Computing in Civil and Building Engineering, Berlin, Germany, July 12-15 1995.

J. Padovan. Applications of 3-d finite element procedures to static and dynamic problems in micropolar elasticity. Computers & Structures, 8(2):231–236, 1978. ISSN 00457949. doi: 10.1016/0045-7949(78)90027-5.

G. M. Paice, D. V. Griffiths, and G. A. Fenton. Finite element modeling of settlement on spatially random soil. Journal of Geotechnical Engineering, 122(9):777–779, 1996.

A. L. Pais and E. Kausel. On rigid foundations subjected to seismic waves. Earthquake Engineering & Structural Dynamics, 18(4):475–489, 1989.

A. Palermo and A. Marzani. Control of love waves by resonant metasurfaces. Scientific Reports, 8, 05 2018. doi: 10.1038/s41598-018-25503-8.

A. Palermo, S. Krödel, A. Marzani, and C. Daraio. Engineered metabarrier as shield from seismic surface waves. Scientific Reports, 6:39356, 12 2016. doi: 10.1038/srep39356.

A. Palermo, S. Krödel, K. H. Matlack, R. Zaccherini, V. K. Dertimanis, E. N. Chatzi, A. Marzani, and C. Daraio. Hybridization of guided surface acoustic modes in unconsolidated granular media by a resonant metasurface. Phys. Rev. Applied, 9:054026, May 2018. doi: 10.1103/PhysRevApplied.9.054026. URL https://link.aps.org/doi/10.1103/PhysRevApplied.9.054026.

J. L. Pan, A. T. C. Goh, K. S. Wong, and A. R. Selby. Three–dimensional analysis of single pile response to lateral soil movements. International Journal for Numerical and Analytical Methods in Geomechanics, 26:747–758, 2002.

G. F. Panza, C. L. Mura, A. Peresan, F. Romanelli, and F. Vaccari. Advances in Geophysics, volume 53, chapter Seismic Hazard Scenarios as Preventive Tools for a Disaster Resilient Society, pages 93–165. Elsevier Inc., 2012. ISSN 0065-2687.

R. Paolucci, M. Shirato, and M. T. Yilmaz. Seismic behaviour of shallow foundations: Shaking table experiments vs numerical modelling. Earthquake Engineering & Structural Dynamics, 37(4):577–595, 2008.

R. Paolucci, I. Mazzieri, G. Piunno, C. Smerzini, M. Vanini, and A. Özcebe. Earthquake ground motion modeling of induced seismicity in the Groningen gas field. Earthquake Engineering & Structural Dynamics, n/a(n/a):20, 2020. doi: 10.1002/eqe.3367. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/eqe.3367.

R. Paolucci, C. Smerzini, and M. Vanini. BB-SPEEDset: A Validated Dataset of Broadband Near-Source Earthquake Ground Motions from 3D Physics-Based Numerical Simulations. Bulletin of the Seismological Society of America, 111(5):2527–2545, October 2021. doi: 10.1785/0120210089.

A. G. Papadimitriou and G. D. Bouckovalas. Plasticity model for sand under small and large cyclic strains: A multiaxial formulation. Soil Dynamics and Earthquake Engineering, 22(3):191–204, 2002.

A. S. Papageorgiou and D. Pei. A discrete wavenumber boundary element method for study of the 3-d response 2-d scatterers. Earthquake Engineering & Structural Dynamics, 27(6):619–638, 1998.

E. Papamichos, I. Vardoulakis, and H.-B. Mühlhaus. Buckling of layered elastic media: A Cosserat-continuum approach and its validation. International Journal for Numerical and Analytical Methods in Geomechanics, 14:473–498, 1990.

W. G. Pariseau. Design Analysis in Rock Mechanics. Taylor & Francis, 1st edition, October 2006. ISBN 978-0415413817.

J. Park. A brief review of tensor operations for students of continuum mechanics. Journal of Applied Engineering Mathematics, 5:1–4, December 2018.

J.-B. Park, Y. Choi, S.-J. Lee, N.-C. Park, K.-S. Park, Y.-P. Park, and C.-I. Park. Modal characteristic analysis of the apr1400 nuclear reactor internals for seismic analysis. Nuclear Engineering and Technology, 46(5):689–698, 2014.

S. Park and P. Byrne. Practical constitutive model for soil liquefaction. In Proc., 9th Int. Symp. on Numerical Models in Geomechanics (NUMOG IX), pages 181–186. CRC Press, Boca Raton, FL, 2004.

M. Pastor, O. C. Zienkiewicz, and A. H. C. Chan. Generalized plasticity and the modeling of soil behaviour. International Journal for Numerical and Analytical Methods in Geomechanics, 14:151–190, 1990.

L. F. Pavarino. BDDC and FETI-DP preconditioners for spectral element discretizations. Computer Methods in Applied Mechanics and Engineering, 196(8):1380–1388, January 2007.

C. E. Pearson, editor. Handbook of Applied Mathematics. Van Nostrand Reinhold Company, 1974.

A. Pecker, J. J. Johnson, and B. Jeremić. Methodologies for Seismic Soil-Structure Interaction Analysis in the Design and Assessment of Nuclear Installations. Number IAEA-TECDOC-1990. United Nations, International Atomic Energy Agency, UN-IAEA, External Events Safety Section, IAEA, VIC, PO Box 100, 1400 Vienna, Austria, February 2022. ISBN 978-92-0-143021-2. IAEAL 21-01471.

A. Pérez-Foguet and A. Huerta. Plastic flow potential for the cone region of the MRS–Lade model, 1997.

A. Pérez-Foguet, A. Rodríguez-Ferran, and A. Huerta. Numerical differentiation for non-trivial consistent tangent matrices: an application to the mrs-lade model. International Journal for Numerical Methods in Engineering, 48:159–184, 2000.

A. Pérez-Foguet, A. Rodríguez-Ferran, and A. Huerta. Consistent tangent matrices for substepping schemes. Computer Methods in Applied Mechanics and Engineering, 190(35-36):4627 – 4647, 2001. ISSN 0045-7825. doi: http://dx.doi.org/10.1016/S0045-7825(00)00336-4. URL http://www.sciencedirect.com/science/article/pii/S0045782500003364.

D. Perić. Localized Deformation and Failure Analysis of Pressure Sensitive Granular Material. PhD thesis, University of Colorado at Boulder, 1991.

N. A. Petersson and B. Sjögreen. High order accurate finite difference modeling of seismo-acoustic wave propagation in a moving atmosphere and a heterogeneous earth model coupled across a realistic topography. Journal of Scientific Computing, –(14):290–323, 2018.

K.-K. Phoon and F. H. Kulhawy. Characterization of geotechnical variability. Canadian Geotechnical Journal, 36:612–624, 1999a.

K.-K. Phoon and F. H. Kulhawy. Evaluation of geotechnical property variability. Canadian Geotechnical Journal, 36:625–639, 1999b.

R. M. V. Pidaparti and A. V. Hudli. Dynamic analysis of structures using object–oriented techniques. Computers and Structures, 49(1):149–156, 1993.

F. Pisanò and B. Jeremić. Simulating stiffness degradation and damping in soils via a simple visco-elastic-plastic model. Soil Dynamics and Geotechnical Earthquake Engineering, 63:98–109, 2014.

R. Popescu. Stochastic variability of soil properties: data analysis, digital simulation, effects on system behavior. Doctoral dissertation, Princeton University, Princeton, NJ, 1995.

R. Popescu, J. H. Prevost, and G. Deodatis. Effects of spatial variability on soil liquefaction: Some design recommendations. Geotechnique, 47(5):1019–1036, 1997.

R. Popescu, J. H. Prevost, and G. Deodatis. Spatial variability of soil properties: Two case studies. In P. Dakoulas and M. Yegian, editors, Geotechnical Earthquake Engineering and Soil Dynamics III: Proceedings of a Speciality Conference held in Seattle, Washington, August 3-6, 1998, Geotechnical Special Publication No. 75, pages 568–579. Geo-Institute of ASCE, ASCE, Reston, VA, 1998.

S. Popovics. A numerical approach to the complete stress-strain curve of concrete. Cement and concrete research, 3(5):583–599, 1973.

D. Post. The coming crisis in computational science. Technical Report LA-UR-04-0388, 2004. Submitted to: Keynote Address, Proceedings of the IEEE International Conference on High Performance Computer Architecture: Workshop on Productivity and Performance in High-End Computing, Madrid, Spain, February 14, 2004.

A. Pothier and J. Rencis. Three-dimensional finite element formulation for microelastic solids. Computers & Structures, 51(1):1–21, 1994. ISSN 00457949. doi: 10.1016/0045-7949(94)90031-0.

J. G. Potyondy. Skin friction between various soils and construction materials. Geotechnique, 11(4): 339–353, 1961.

M. Preisig. Nonlinear finite element analysis of dynamic Soil - Foundation - Structure interaction. Master's thesis, University of California, Davis, 2005.

W. H. Press, B. P. Flannery, S. A. teukolsky, and W. T. Vetterling. Numerical Recipes in C. Cambridge University Press, Cambridge, 1988a.

W. H. Press, B. P. Flannery, S. A. Teukolvsky, and W. T. Vetterling. Numerical Recipes in C, The Art of Scientific Computing. Cambridge University Press, 1988b.

J. S. Pressley and H. G. Poulos. Finite element analysis of mechanisms of pile group behavior. International Journal for Numerical and Analytical Methods in Geomechanics, 10:213–221, 1986.

J. H. Prevost. A simple plasticity theory for frictional cohesionless soils. International Journal of Soil Dynamics and Earthquake Engineering, 4(1):9 – 17, 1985a. ISSN 0261-7277. doi: http://dx.doi.org/10.1016/0261-7277(85)90030-0. URL http://www.sciencedirect.com/science/article/pii/0261727785900300.

J. H. Prevost. Wave propagation in fluid-saturated porous media: an efficient finite element procedure. Soil Dynamics and Earthquake Engineering, 4(4):183 –202, 1985b.

J. H. Prevost. DYNA1D: A computer program for nonlinear site response analysis, technical documentation. technical report no. nceer-89-0025. Technical report, National Center for Earthquake Engineering Research, State University of New York at Buffalo, 1989.

J. H. Prevost and R. Popescu. Constitutive relations for soil materials. Electronic Journal of Geotechnical Engineering, October 1996. available at http://139.78.66.61/ejge/.

J. S. Przemieniecki. Theory of Matrix Structural Analysis. McGraw Hill, New York, 1985.

R. Rackwitz. Reviewing probabilistic soil modelling. Computers and Geotechnics, 26:199–223, 2000.

R. Ramey. Making a boost library. In D. Musser, editor, Library-Centric Software Design LCSD'05. Object-Oriented Programming, Systems, Languages and Applications, October 2005.

E. Ramm. Strategies for tracing the nonlinear response near limit points. In W. Wunderlich, E. Stein, and K.-J. Bathe, editors, Nonlinear Finite Element Analysis in Structural Mechanics, pages 63–89. Springer–Werlag Berlini Heidelberg New York, 1981.

E. Ramm. The Riks/Wempner approach – an extension of the displacement control method in nonlinear analysis. In E. Hinton, D. Owen, and C. Taylor, editors, Recent Advances in Non–Linear Computational Mechanics, chapter 3, pages 63–86. Pineridge Press, Swansea U.K., 1982.

E. Ramm and A. Matzenmiller. Consistent linearization in elasto - plastic shell analysis. Engineering Computations, 5:289–299, 1988.

B. Raphael and C. S. Krishnamoorthy. Automating finite element development using object oriented techniques. Engineering Computations, 10:267–278, 1993.

G. Ravichandran, A. J. Rosakis, J. Hodowany, P. Rosakis, M. D. Furnish, N. N. Thadhani, and Y. Horie. On the conversion of plastic work into heat during high-strain-rate deformation. In AIP conference proceedings, volume 620/1, pages 557–562. AIP, 2002.

L. Rayleigh. On waves propagated along the plane surface of an elastic solid. Proceedings of the London Mathematical Society, 1(1):4, 1885.

L. C. Reese, J. D. Allen, and J. Q. Hargrove. Laterally loaded piles in layered soils. In P. C. of X IC-SMFE, editor, Proceedings of the Tenth International Conference on Soil Mechanics and Foundations Engineering, volume 2, pages 819–822, Stockholm, June 1981. A. A. Balkema.

L. C. Reese, S. T. Wang, W. M. Isenhower, and J. A. Arrellaga. LPILE plus 4.0 Technical Manual. ENSOFT, INC., Austin, TX, version 4.0 edition, Oct. 2000a.

L. C. Reese, S. T. Wang, I. W. M., A. J. A., and H. J. LPILE plus 4.0 User Guide. ENSOFT, INC., Austin, TX, version 4.0 edition, Oct. 2000b.

B. Ren and S. Li. Meshfree simulations of plugging failures in high-speed impacts. Computers & structures, 88(15):909–923, 2010.

P. L. Renault, N. A. Abrahamson, K. J. Coppersmith, M. Koller, P. Roth, and A. Hölker. Probabilistic Seismic Hazard Analysis for Swiss Nuclear Power Plant Sites - PEGASOS Refinement Project. Final Report, Vol. 1-5 . Technical report, Swissnuclear, 2013.

C. Reschke, T. Sterling, D. Ridge, D. Savarese, D. J. Becker, and P. Merkey. A design study of alternative network topologies for the Beowulf parallel workstation. In Proceedings, High Performance and Distributed Computing, 1996. `http://cesdis.gsfc.nasa.gov/beowulf/papers.html`

A. Riahi and J. H. Curran. Full 3D finite element Cosserat formulation with application in layered structures. Applied Mathematical Modelling, 33(8):3450–3464, 2009. ISSN 0307904X. doi: 10.1016/j.apm.2008.11.022.

F. E. Richart, J. J. R. Hall, and R. D. Woods. Vibration of Soils and Foundations, International Series in Theoretical and Applied Mechanics. Prentice-Hall, Englewood Cliffs, N.J., 1970.

D. Ridge, D. J. Becker, P. Merkey, and T. Sterling. BEOWULF: Harnesing the power of parallelism in a Pile–of–PCs. In Proceedings, IEEE Aerospace, 1997. `http://cesdis.gsfc.nasa.gov/beowulf/papers.html`

E. Riks. The application of Newton's method to the problems of elastic stability. Journal of Applied Mechanics, 39:1060–1066, December 1972.

E. Riks. An incremental approach to the solution of snapping and buckling problems. International Journal for Solids and Structures, 15:529–551, 1979.

H. Risken. The Fokker-Planck Equation: Methods of Solutions and Applications. Springer Series in Synergetics. Springer-Verlag, Berlin Heidelberg, second edition, 1989.

D. Rittel. An investigation of the heat generated during cyclic loading of two glassy polymers. part i: Experimental. Mechanics of Materials, 32(3):131–147, 2000.

D. Rittel and Y. Rabin. An investigation of the heat generated during cyclic loading of two glassy polymers. part ii: Thermal analysis. Mechanics of Materials, 32(3):149–159, 2000.

D. Rittel, N. Eliash, and J. Halary. Hysteretic heating of modified poly (methylmethacrylate). Polymer, 44(9):2817–2822, 2003.

J.-F. Ritz, S. Baize, M. Ferry, C. Larroqu, L. Audin, B. Delouis, and E. Mathot. Surface rupture and shallow fault reactivation during the 2019 Mw 4.9 Le Teil earthquake, France. COMMUNICATIONS EARTH & ENVIRONMENT, 1(10), 2020. doi: 10.1038/s43247-020-0012-z.

D. Rixena and F. Magoulès. Domain decomposition methods: Recent advances and new challenges in engineering. Computer Methods in Applied Mechanics and Engineering, 196(8):1345–1346, January 2007.

E. Rizzi. Localization analysis of damaged materials. Master's thesis, University of Colorado at Boulder, 1993.

P. J. Roache. Verification and Validation in Computational Science and Engineering. Hermosa Publishers, Albuquerque, New Mexico, 1998. ISBN 0-913478-08-3.

Y. Robert. Regular incomplete factorizations of real positive definite matrices. Linear Algebra and Its Applications, 48:105–117, 1982.

A. D. Robison. C++ gets faster for scientific computing. Computers in Physics, 10(5):458–462, Sept/Oct 1996.

C. Roblee, W. Silva, G. Toro, and N. Abrahamson. Variability in site-specific seismic ground-motion predictions. In Proceedings: Uncertainty in the Geologic Environment, page 21p, Madison WI., August 1-2 1996. American Society of Civil Engineers.

A. Rodgers. Private communications. SW4 – Real-ESSI connection, 2017.

K. M. Rollins, K. T. Peterson, and T. J. Weaver. Lateral load behavior of full scale pile group in clay. Journal of geotechnical and geoenvironmental engineering, 124(6):468–478, June 1997.

K. M. Rollins, T. M. Gerber, J. D. Lane, and S. A. Ashford. Lateral resistance of a full–scale pile group in liquefied sand. Journal of Geotechnical and Geoenvironmental Engineering, 131(1):115–125, 2005.

A. Romero, P. Galvín, and J. Domínguez. 3D non-linear time domain fem–bem approach to soil–structure interaction problems. Engineering Analysis with Boundary Elements, 37(3):501–512, 2013.

P. Rosakis, A. Rosakis, G. Ravichandran, and J. Hodowany. A thermodynamic internal variable model for the partition of plastic work into heat and stored energy in metals. Journal of the Mechanics and Physics of Solids, 48(3):581–607, 2000.

K. H. Roscoe and J. B. Burland. On the generalized stress-strain behaviour of "wet" clay. In Engineering plasticity, pages 553–609. Cambridge University Press, 1968.

K. H. Roscoe, A. N. Schofield, and C. P. Wroth. On the yielding of soils. Geotechnique, 8(1):22–53, 1958.

K. H. Roscoe, A. N. Schofield, and A. Thurairajah. Yielding of clays in states wetter than critical. Géotechnique, 13(3):211–240, 1963.

C. Rose and M. D. Smith. Mathematical Statistics with Mathematica. Springer Texts in Statistics. Springer-Verlag, New York, 2002.

T. J. Ross, L. R. Wagner, and G. F. Luger. Object-Oriented Programming for Scientific Codes. II: Examples in C++. Journal of Computing in Civil Engineering, 6(4):497–514, 1992.

P. W. Rowe. The stress–dilatancy relation for static equilibrium of an assembly of particles in contact. Proceedings of the Royal Society, 269(1339):500–527, 9 October 1962. Series A Mathematical and Physical Sciences.

C. Roy, A. J. Roffel, S. Bolourchi, L. Todorovski, and M. Khoncarly. Study of seismic structure-soil-structure interaction between two heavy structures. In Transactions, SMiRT-22, pages 1–7, San Francisco, August 2013. IASMiRT.

C. J. Roy and W. L. Oberkampf. A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing. Computer Methods in Applied Mechanics and Engineering, 200 (25-28):2131 – 2144, 2011. ISSN 0045-7825. doi: 10.1016/j.cma.2011.03.016. URL http://www.sciencedirect.com/science/article/pii/S0045782511001290.

M. D. Rucki and G. R. Miller. An Algorithmic Framework for Flexible Finite Element-Based Structural Modeling. Computer Methods in Applied Mechanics and Engineering, 136(3–4):363–384, 1996.

J. W. Rudnicki and J. R. Rice. Conditions for the localization of deformation in pressure–sensitive dilatant materials. Journal of the Mechanics and Physics of Solids, 23:371 to 394, 1975.

P. F. Ruesta and F. C. Townsend. Evaluation of laterally loaded pile group at Roosevelt bridge. Journal of Geotechnical and Geoenvironmental Engineering, 123(12):1153–1161, December 1997.

J. Rumbaugh, M. Blaha, W. Premerhani, F. Eddy, and W. Lorensen. Object-Oriented Modeling and Design. Prentice Hall, Englewood Cliffs, New Jersey, 1991.

K. Runesson. Implicit integration of elastoplastic relations with reference to soils. International Journal for Numerical and Analytical Methods in Geomechanics, 11:315–321, 1987.

K. Runesson. Constitutive theory and computational technique for dissipative materials with emphasis on plasticity, viscoplasticity and damage: Part III. Lecture Notes, Chalmers Technical University, Göteborg, Sweden, September 1996.

K. Runesson and Z. Mrŏz. A note on nonassociated plastic flow rules. International Journal of Plasticity, 5:639–658, 1989.

K. Runesson and A. Samuelsson. Aspects on numerical techniques in small deformation plasticity. In G. N. P. J. Middleton, editor, NUMETA 85 Numerical Methods in Engineering, Theory and Applications, pages 337–347. AA.Balkema., 1985.

H. Ryan. Ricker, Ormsby, Klander, Butterworth - a choice of wavelets. Canadian Society of Exploration Geophysicists Recorder, 19(7):8–9, September 1994. URL http://www.cseg.ca/publications/recorder/1994/09sep/sep94-choice-of-wavelets.pdf.

Y. Saad. Iterative Methods for Sparse Linear Systems. SIAM, Philadelphia, second edition, 2003.

S. Sakamoto and R. Ghanem. Polynomial chaos decomposition for the simulation of non-gaussian nonstationary stochastic processes. Journal of Engineering Mechanics, 128(2):190–201, February 2002.

T. Sakemi, M. Tanaka, Y. Higuchi, K. Kawasaki, and K. Nagura. Permeability of pore fluids in the centrifugal fields. In 10th Asian Regional Conference on Soil Mechanics and Foundation Engineering (10ARC), Beijing, China, August 29 - Sept 2 1995.

M. D. Salas. The curious events leading to the theory of shock waves. In 17th Shock Interaction Symposium, Rome, Italy, 4-8 September 2006. URL http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20060047586_2006228914.pdf.

V. Saouma and L. Perotti. Constitutive model for alkali-aggregate reactions. Materials Journal, 103(3): 194–202, 2006.

V. E. Saouma. Prof. Saouma's lecture notes, manuals, &c. (available through his web page: http://civil.colorado.edu/~saouma/), 1992-2013.

R. Z. Sarica and M. S. Rahman. A wavelet analysis of ground motion characteristics. In U.S.-Taiwan Workshop on Soil Liquefaction, 2003.

M. Sarkis and D. B. Szyld. Optimal left and right additive Schwarz preconditioning for minimal residual methods with Euclidean and energy norms. Computer Methods in Applied Mechanics and Engineering, 196(8):1612–1621, January 2007.

R. Sause and J. Song. Object-Oriented Structural Analysis with Substructures. In Computing in Civil Engineering: Proceedings of the First Conference held in Conjunction with with A/E/C Systems' 94, Washington, D.C., June 20–22 1994.

J. C. J. Schellekens and H. Parisch. On finite deformation elasticity. Technical Report ISD Nr. 94/3, Institute for Statics and Dynamics of Aerospace Structures, Pfaffenwaldring 27, 70550 Stutgart, 1994.

K. Schloegel, G. Kaypis, and V. Kumar. Graph partitioning for high performance scientific simulations. Technical report, Army HPC Research Center, Department of Computer Science and Engineering, University of Minnesota, 1999.

K. Schloegel, G. Karypis, and V. Kumar. A unified algorithm for load-balancing adaptive scientific simulations. Technical report, Army HPC Research Center, Department of Computer Science and Engineering, University of Minnesota, 2000.

P. Schnabel, J. Lysmer, and H. B. Seed. Shake – a computer program for equation response analysis of horizontally layered sites. report eerc 72-12. Technical report, University of California Berkeley, 1972.

A. Schofield and C. Wroth. Critical state soil mechanics. McGraw–Hill, London, 1968.

S.-P. Scholz. Elements of an object – oriented FEM++ program in C++. Computers and Structures, 43(3):517–529, 1992.

G. I. Schüeller. A state-of-the-art report on computational stochastic mechanics. Probabilistic Engineering Mechanics, 12(4):197–321, 1997.

B. Scott, R. Park, and M. Priestley. Fiber element modeling for seismic performance of bridge columns made of concrete-filled frp tubes. Journal of the American Concrete Institute, 79(1):13–27, 1982.

H. B. Seed and I. M. Idriss. Soil moduli and damping factors for dynamic response analyses. report eerc 70-10. Technical report, University of California Berkeley, 1970a.

H. B. Seed and I. M. Idriss. Soil moduli and damping factors for dynamic response analyses. report eerc 70-10. Technical report, University of California Berkeley, 1970b.

J. F. Semblat. Rheological interpretation of rayleigh damping. Journal of Sound and Vibration, 206(5): 741–744, 1997.

J.-F. Semblat and A. Pecker. Waves and Vibrations in Soils: Earthquakes, Traffic, Shocks, Construction works. IUSS Press, first edition, 2009. ISBN ISBN-10: 8861980309; ISBN-13: 978-8861980303.

J.-F. Semblat, L. Lenti, and A. Gandomzadeh. A simple multi-directional absorbing layer method to simulate elastic wave propagation in unbounded domains. International Journal for Numerical Methods in Engineering, doi: 10.1002/nme.3035, 2010. URL `http://dx.doi.org/10.1002/nme.3035`.

S. J. Semnani, J. A. White, and R. I. Borja. Thermoplasticity and strain localization in transversely isotropic materials based on anisotropic critical state plasticity. International Journal for Numerical and Analytical Methods in Geomechanics, 40(18):2423–2449, 2016.

K. Sett. Probabilistic elasto–plasticity and its application in finite element simulations of stochastic elastic–plastic boundary value problems. Doctoral Dissertation, University of California, Davis, CA, September 2007.

K. Sett and B. Jeremić. Uncertain soil properties and elastic–plastic simulations in geomechanics. In K. K. Phoon, G. A. Fenton, E. F. Glynn, C. H. Juang, T. F. Griffiths, T. F. Wolff, and L. Zhang, editors, Probabilistic Applications in Geotechnical Engineering (Proceedings of Geo-Denver 2007: New Peaks in Geotechnics, held in Denver, CO from February 18-21, 2007), Geotechnical Special Publication No. 170, pages 1–11. American Society for Civil Engineers, 2007.

K. Sett and B. Jeremić. Forward and backward probabilistic simulations in geotechnical engineering. In M. Iskander, D. F. Laefer, and M. H. Hussein, editors, Contemporary Topics in Insitu Testing, Analysis, and Reliability of Foundations (Selected Papers from the 2009 International Foundation Congress and Equipment Expo, held in Orlando, FL, from March 15-19, 2009), volume 186 of Geotechnical Special Publications No. 186, pages 1–11. American Society for Civil Engineers, 2009.

K. Sett and B. Jeremić. Probabilistic yielding and cyclic behavior of geomaterials. International Journal for Numerical and Analytical Methods in Geomechanics, 34(15):1541–1559, 2010. 10.1002/nag.870 (first published online March 11, 2010).

K. Sett, B. Jeremić, and M. L. Kavvas. Probabilistic elasto-plasticity: Solution and verification in 1D. Acta Geotechnica, 2(3):211–220, October 2007a.

K. Sett, B. Jeremić, and M. L. Kavvas. The role of nonlinear hardening in probabilistic elasto-plasticity. International Journal for Numerical and Analytical Methods in Geomechanics, 31(7):953–975, June 2007b.

K. Sett, B. Jeremić, and M. L. Kavvas. Probabilistic elasto-plasticity: Solution and verification in 1–D. Acta Geotechnica, 2(3):211–220, September 2007c.

K. Sett, B. Jeremić, and M. L. Kavvas. The role of nonlinear hardening/softening in probabilistic elasto–plasticity. International Journal for Numerical and Analytical Methods in Geomechanics, 31(7): 953–975, June 2007d.

K. Sett, B. Unutmaz, K. O. Çetin, S. Koprivica, and B. Jeremić. Soil uncertainty and its influence on simulated G/G_{max} and damping behaviors. Journal of Geotechnical and Geoenvironmental Engineering, 2008. In review.

K. Sett, B. Jeremić, and M. L. Kavvas. Stochastic elastic-plastic finite elements. Computer Methods in Applied Mechanics and Engineering, 200(9-12):997–1007, February 2011a. ISSN 0045-7825. doi: DOI:10.1016/j.cma.2010.11.021.

K. Sett, B. Unutmaz, K. Önder Çetin, S. Koprivica, and B. Jeremić. Soil uncertainty and its influence on simulated G/G_{max} and damping behavior. ASCE Journal of Geotechnical and Geoenvironmental Engineering, 137(3):218–226, 2011b. 10.1061/(ASCE)GT.1943-5606.0000420 (July 29, 2010).

I. Shahrour and F. Rezaie. An elastoplastic constitutive relation for the soil-structure interface under cyclic loading. Computers and Geotechnics, 21(1):21–39, 1997.

I. Shahrour, F. Khoshnoudian, M. Sadek, and H. Mroueh. Elastoplastic analysis of the seismic response of tunnels in soft soils. Tunnelling and underground space technology, 25(4):478–482, 2010.

N. Sharma, K. Dasgupta, and A. Dey. Optimum lateral extent of soil domain for dynamic ssi analysis of rc framed buildings on pile foundations. Frontiers of Structural and Civil Engineering, 14(1):62–81, 2020.

B. Shen, W. Xu, J. Wang, Y. Chen, W. Yan, J. Huang, and Z. Tang. Seismic control of super high-rise structures with double-layer tuned particle damper. Earthquake Engineering & Structural Dynamics, 50(3):791–810, 2021. doi: https://doi.org/10.1002/eqe.3372. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/eqe.3372.

F. Shipman, V. Gregson, and A. Jones. A shock wave study of coconino sandstone. Technical Report NASA CR-1842, NASA, May 1971.

T. Sigaki, K. Kiyohara, Y. Sono, D. Kinosita, T. Masao, R. Tamura, C. Yoshimura, and T. Ugata. Estimation of earthquake motion incident angle at rock site. In Proceedings of 12th world conference on earthquake engineering, pages 1–8, 2000.

J.-F. Sigrist, D. Broc, and C. Lainé. Dynamic analysis of a nuclear reactor with fluid–structure interaction: Part i: Seismic loading, fluid added mass and added stiffness effects. Nuclear engineering and design, 236(23):2431–2443, 2006.

W. Silva. Body waves in a layered anelastic solid. Bulletin of the Seismological Society of America, 66 (5):1539–1554, 1976.

J. Simo and J. Ju. Strain- and stress-based continuum damage models: Formulation. International Journal of Solids and Structures, 23(7):821 – 840, 1987. ISSN 0020-7683. doi: https://doi.org/10.1016/0020-7683(87)90083-7. URL http://www.sciencedirect.com/science/article/pii/0020768387900837.

J. C. Simo. A framework for finite strain elastoplasticity based on maximum plastic dissipation and the multiplicative decomposition: Part i. continuum formulation. Computer Methods in Applied Mechanics and Engineering, 66:199–219, 1988. TA345. C6425.

J. C. Simo. Algorithms for static and dynamic multiplicative plasticity that preserve the classical return mapping schemes of the infinitesimal theory. Computer Methods in Applied Mechanics and Engineering, 99:61–112, 1992.

J. C. Simo and R. L. fTaylor. Quasi–incompressible finite elasticity in principal stretches. continuum basis and numerical algorithms. Computer Methods in Applied Mechanics and Engineering, 85:273–310, 1991.

J. C. Simo and S. Govindjee. Exact closed - form solution of the return mapping algorith in plane stress elasto - viscoplasticity. Engineering Computations, 5:254–258, 1988.

J. C. Simo and T. J. R. Hughes, editors. Computational Inelasticity. Springer-verlag, New York, 1998.

J. C. Simo and C. Miehe. Associative coupled thermoplasticity at finite strains: Formulation, numerical analysis and implementation. Computer Methods in Applied Mechanics and Engineering, 98:41–104, 1992.

J. C. Simo and K. S. Pister. Remarks on rate constitutive equations for finite deformations problems: Computational implications. Computer Methods in Appliied Mechanics and Engineering, 46:201–215, 1984.

J. C. Simo and R. L. Taylor. Consistent tangent operators for rate-independent elastoplasticity. Computer Methods in Applied Mechanics and Engineering, 48:101–118, 1985.

J. C. Simo and R. L. Taylor. A returning mapping algorithm for plane stress elastoplasticity. International Journal for Numerical Methods in Engineering, 22:649–670, 1986.

S. K. Sinha and B. Jeremić. Modeling of dry and saturated soil-foundation contact. Technical Report UCD–CompGeoMech–01–2017, University of California, Davis, August 2017.

S. K. Sinha, Y. Feng, H. Yang, H. Wang, N. Orbović, D. B. McCallen, and B. Jeremić. 3-D non-linear modeling and its effects in earthquake soil-structure interaction. In Proceedings of the 24th International Conference on Structural Mechanics in Reactor Technology (SMiRT 24), Busan, South Korea, August 20-25 2017.

B. Sjögreen and N. A. Petersson. A Fourth Order Accurate Finite Difference Scheme for the Elastic Wave Equation in Second Order Formulation. J. Sci. Comput., 52(1):17–48, 2011. ISSN 0885-7474. doi: 10.1007/s10915-011-9531-1. URL http://link.springer.com/10.1007/s10915-011-9531-1$\backslash$nhttps://computation-rnd.llnl.gov/serpentine/pubs/wpp4th2011.pdf$\backslash$nhttps://computation-rnd.llnl.gov/serpentine/publications.html.

A. Skempton. The pore-pressure coefficients a and b. Geotechnique, 4:143–147, 1954.

C. Smerzini, M. Vanini, R. Paolucci, and P. Traversa. Regional physics-based simulation of ground motion within the Rhône Valley, France, during the Mw 4.9 2019 Le Teil earthquake. Bulletin of Earthquake Engineering, 21:1747–1774, 2023. doi: 10.1007/s10518-022-01591-w.

I. Sobol. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. Mathematics and Computers in Simulation, 55(1):271 – 280, 2001. ISSN 0378-4754. doi: https://doi.org/10.1016/S0378-4754(00)00270-6. URL http://www.sciencedirect.com/science/article/pii/S0378475400002706. The Second IMACS Seminar on Monte Carlo Methods.

C. Soize. The Fokker-Planck Equation for stochastic dynamical systems and its explicit steady state solutions. World Scientific, Singapore, 1994.

T. Soong and B. Spencer. Supplemental energy dissipation: state-of-the-art and state-of-the-practice. Engineering Structures, 24(3):243–259, 2002.

M. Souli and J. Zolesio. Arbitrary lagrangian–eulerian and free surface methods in fluid mechanics. Computer Methods in Applied Mechanics and Engineering, 191(3):451–466, 2001.

E. Spacone, F. C. Filippou, and F. F. Taucer. Fibre beam-column model for non-linear analysis of r/c frames: Part i. formulation. Earthquake Engineering & Structural Dynamics, 25(7):711–725, July 1996a.

E. Spacone, F. C. Filippou, and F. F. Taucer. Fibre beam-column model for non-linear analysis of r/c frames: Part ii. applications. Earthquake Engineering & Structural Dynamics, 25(7):727–742, July 1996b.

A. Spencer. Continuum Mechanics. Longman Mathematical Texts. Longman Group Limited, 1980.

C. Spyrakos, P. Patel, and F. Kokkinos. Assessment of computational practices in dynamic soil-structure interaction. Journal of computing in civil engineering, 3(2):143–157, 1989.

A. Stavrogin, B. Tarasov, and e. C. Fairhurst. Experimental physics and rock mechanics. A.A. Balkema, Lisse ; Exton PA, 2001. ISBN 9058092135.

A. N. Stavrogin and E. D. Pevzner. Critical states for rocks at changing deformation rates. Journal of Mining Science, 19(5):359–366, 1983.

A. N. Stavrogin and A. G. Protosenya. Rock plasticity in conditions of variable deformation rates. Journal of Mining Science, 19(4):245–255, 1983.

T. Sterling, D. J. Becker, D. Savarese, J. E. Dorband, U. A. Ranawake, and C. V. Parker. BEOWULF: A parallel workstation for scientific computations. In Proceedings of the International Conference on Parallel on Parallel Processing, 1995. `http://cesdis.gsfc.nasa.gov/beowulf/papers.html`

T. Sterling, T. Cwik, D. Becker, J. Salmon, M. Warren, and B. Nitzberg. An assessment of Beowulf-class computing for NASA requirements: Initial findings from the first NASA workshop on beowulf-class clustered computing. In Proceedings, IEEE Aerospace, 1998. http://www.beowulf.org/papers/index.html.

T. L. Sterling, J. Salmon, D. J. Becker, and D. F. Savarese. How to Build a Beowulf: A Guide to the Implementation and Application of PC Clusters. Scientific and Engineering Computations Series. The MIT Press, 1999. ISBN 0-262-69218-X ; QA 76.58.S854 1998.

J. B. Stevens and J. M. E. Audibert. Re-examination of p-y curve formulations. In Eleventh Annual Offshore Technology Conference, volume I, pages 397–403, Dallas, TX, April 1979. Americal Society of Civil Engineers.

S.Timoshenko and D.H.Young. Engineering Mechanics. McGraw-Hill Book Company, London, 1940.

K. H. Stokoe II, R. B. Darendeli, R. B. Gilbert, F.-Y. Menq, and W. K. Choi. Development of a new family of normalized modulus reduction and material damping curves. In International Workshop on Uncertainties in Nonlinear Soil Properties and their Impact on Modeling Dynamic Soil Response, PEER Headquarters, UC Berkeley, March 18-19, 2004, Plenary Paper, http://peer.berkeley.edu/lifelines/Workshop304/pdf/Stokoe_PlenaryPaper.pdf, 2004. PEER.

O. O. Storaasli and P. Bergan. Nonlinear substructuring method for concurrent processing computers. AIAA Journal, 25(6):871–876, 1987.

S. Stošić. Private communications, 1984-2022.

B. Stroustrup. The C++ Programming Language. Series in Computer Sciences. Addison – Wesley, reprint 1987 edition, 1986.

B. Stroustrup. The Design and Evolution of C++. Addison–Wesley Publishing Company, 1994.

B. Stroustrup. A rationale for semantically enhanced library languages. In D. Musser, editor, Library-Centric Software Design LCSD'05. Object-Oriented Programming, Systems, Languages and Applications, October 2005.

S. Sture. Engineering properties of soils. Lecture Notes at CU Boulder, January - May 1993.

H. H. Stutz. Hypoplastic Models for Soil-Structure Interfaces-Modelling and Implementation. PhD thesis, Christian-Albrechts Universität Kiel, 2016.

B. Sudret. Global sensitivity analysis using polynomial chaos expansions. Reliability engineering & system safety, 93(7):964–979, 2008.

B. Sudret and A. Der Kiureghian. Stochastic finite element methods and reliability: A state of the art report. Technical Report UCB/SEMM-2000/08, University of California, Berkeley, 2000.

M. Symans and M. Constantinou. Passive fluid viscous damping systems for seismic energy dissipation. ISET Journal of Earthquake Technology, 35(4):185–206, 1998.

M. Symans, F. Charney, A. Whittaker, M. Constantinou, C. Kircher, M. Johnson, and R. McNamara. Energy dissipation systems for seismic applications: current practice and recent developments. Journal of structural engineering, 134(1):3–21, 2008.

B. Szabó and R. Actis. Simulation governance: Technical requirements for mechanical design. Computer Methods in Applied Mechanics and Engineering, 249-252:158–168, 2012. doi: 10.1016/j.cma.2012.02.008.

B. A. Szabó and R. L. Actis. Simulation governance: New technical requirements for software tools in computational solid mechanics. Presentation at the: International Workshop on Verification and Validation in Computational Science University of Notre Dame, 17-19 October 2011.

M. Tabatabaie, N. Abrahamson, and J. Singh. Effect of seismic wave inclination on structural response. In Dynamic Response of Structures, pages 613–620. ASCE, 1986.

N. Tafazzoli. Methods, Computational Platform, Verification, and Application of Earthquake-Soil-Structure-Interaction Modeling and Simulation. PhD thesis, University of California at Davis, 2012.

M. Taiebat and Y. F. Dafalias. SANISAND: Simple anisotropic sand plasticity model. International Journal for Numerical and Analytical Methods in Geomechanics, 2008. (in print, available in earlyview).

F. Taucer, E. Spacone, and F. C. Filippou. A fiber beam-column element for seismic response analysis of reinforced concrete structures. Technical Report Report No. UCB/EERC-91/17, Earthquake Engineering Research Center, College of Engineering, University of California, Berkeley, December 1991.

G. I. Taylor and H. Quinney. The latent energy remaining in a metal after cold working. Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character, 143(849):307–326, 1934.

R. Taylor, J. Simo, O. Zienkiewicz, and A. Chen. The patch test – a condition for assessing FEM convergence. International Journal for Numerical Methods in Engineering, 22:39–62, 1986.

R. Teisseyre, T. Minoru, and E. Majewski. Earthquake source asymmetry, structural media and rotation effects. Springer, 2006.

J. Tejchman and W. Wu. Experimental and numerical study of sand–steel interfaces. International journal for numerical and analytical methods in geomechanics, 19(8):513–536, 1995.

R. Temam. Mathematical Problems in Plasticity. Gauthier–Villars, 1985.

W. T. Thomson. Transmission of elastic waves through a stratified solid medium. Journal of Applied Physics, 21(2):89–93, 1950.

S. S. Timoshenko. History of Strength of Materials. McGraw–Hill, Book Company, Inc., 1953.

T. C. T. Ting. Determination of $C^{1/2}$, $C^{-1/2}$ and more general isotropic tensor functions of C. Journal of Elasticity, 15:319–323, 1985.

K. Tokida, H. Matsumoto, and H. Iwasaki. Experimental study of drag action of piles in ground flowing by liquefaction. In Proceedings of Fourth Japan–U.S. Workshop on Earthquake Resistant Design of Lifeline Facilities and Countermeasures for Soil Liquefaction, volume 1 of NCEER92-0019, SUNY Buffalo, N. Y., 1992.

K. Tokimatsu and Y. Asaka. Effects of liquefaction–induced ground displacements on pile performance in the 1995 hyogoken–nambu earthquake. Special Issue of Soils and Foundations, Japanese Geotechnical Society, pages 163–177, 1998.

H. Toopchi-Nezhad, M. J. Tait, and R. G. Drysdale. Lateral response evaluation of fiber-reinforced neoprene seismic isolators utilized in an unbonded application. ASCE Journal of Structural Engineering, 134(10):1627–1637, October 2008.

M. Trifunac. A note on rotational components of earthquake motions on ground surface for incident body waves. International Journal of Soil Dynamics and Earthquake Engineering, 1(1):11–19, 1982.

A. M. Trochanis, J. Bielak, and P. Christiano. Three-dimensional nonlinear study of piles. Journal of Geotechnical Engineering, 117(3):429–447, March 1991.

W. Tseng, K. Lilhanand, F. Ostadan, and S. Tuann. Post-earthquake analysis and data correlation for the 1/4-scale containment model of the Lotung experiment. Technical Report NP-7305-SL, EPRI, Electric Power Research Institute, 3412 Hillview Avenue, Palo Alto, California, 94304, 1991.

A. Tyapin. The frequency–dependent elements in the code SASSI: A bridge between civil engineers and the soil–structure interaction specialists. Nuclear Engineering and Design, 237:1300–1306, 2007.

C.-M. Uang and V. V. Bertero. Evaluation of seismic energy in structures. Earthquake Engineering & Structural Dynamics, 19(1):77–90, 1990.

M. Uesugi and H. Kishida. Influential factors of friction between steel and dry sands. Soils and foundations, 26(2):33–46, 1986a.

M. Uesugi and H. Kishida. Frictional resistance at yield between dry sand and mild steel. Soils and foundations, 26(4):139–149, 1986b.

M. Uesugi, H. Kishida, and Y. Tsubakihara. Friction between sand and steel under repeated loading. Soils and foundations, 29(3):127–137, 1989.

M. Uesugi, H. Kishida, and Y. Uchikawa. Friction between dry sand and concrete under monotonic and repeated loading. Soils and Foundations, 30(1):115–128, 1990.

S. Utku, R. Melosh, M. Islam, and M. Salama. On nonlinear finite element analysis in single– multi– and parallel processors. Computers & Structures, 15(1):39–47, 1982.

R. Uzuoka, M. Čubrinovski, H. Sugita, M. Sato, K. Tokimatsu, N. Sento, M. Kazama, F. Zhang, A. Yashima, and F. Oka. Prediction of pile response to lateral spreading by 3-d soil–coupled dynamic analysis: Shaking in the direction perpendicular to ground flow. Soil Dynamics and Earthquake Engineering, 28(6):436–452, June 2008.

N. G. Van Kampen. Stochastic differential equations. Phys. Rep., 24:171–228, 1976.

R. Van Loon, P. Anderson, F. Van de Vosse, and S. Sherwin. Comparison of various fluid–structure interaction methods for deformable bodies. Computers & structures, 85(11):833–843, 2007.

E. Vanmarcke. Random Fields: Analysis and Synthesis. The MIT Press, Cambridge, Massachusetts, 1983.

E. H. Vanmarcke. Probabilistic modeling of soil profiles. Journal of Geotechnical Engineering Division, 103(11):1227–1246, 1977.

I. Vardoulakis. Shear-banding and liquefaction in granular materials on the basis of a Cosserat continuum theory. Ingenieur Archiv, 59:106–113, 1989.

I. Vardoulakis and J. Sulem. Bifurcation Analysis in Geomechanics. Blackie Academic & Professional, 1995. ISBN 0-7514-0214-1.

R. S. Varga, E. B. Saff, and V. Mehrmann. Incomplete factorizations of matrices and connections with H-matrices. SIAM Journal on Numerical Analysis, 17:787–793, 1980.

Various Authors. The C++ report: Columns on C++, 1991-.

A. V. Vasilev, E. E. Lovetskii, and V. I. Selyakov. Injection effect in a contained explosion in a liquid-saturated medium. translated from Zhurnal Prikladnoi Mekhaniki i Tekhnicheskoi Fiziki, 4:107–112, July – August 1980. URL http://www.springerlink.com/index/N24126788T374535.pdf.

M. F. Vassiliou, A. Tsiavos, and B. Stojadinović. Dynamics of inelastic base-isolated structures subjected to analytical pulse ground motions. EARTHQUAKE ENGINEERING & STRUCTURAL DYNAMICS, 42:2043–2060, 2013.

T. Veldhuizen. Using C++ template metaprograms. C++ Report, 7(4):36–43, May 1995a.

T. Veldhuizen. Expression templates. C++ Report, 7(5):26–31, June 1995b.

T. Veldhuizen. Rapid linear algebra in C++. Dr. Dobb's Journal, August 1996.

T. Veldhuizen. Software libraries and their reuse: Entropy, kolmogorov complexity, and zipf's law. In D. Musser, editor, Library-Centric Software Design LCSD'05. Object-Oriented Programming, Systems, Languages and Applications, October 2005.

T. L. Veldhuizen and M. E. Jernigan. Will C++ be faster than FORTRAN. In J. Hull, editor, Proceedings of the first Internationa conference Scientific Computing in Object-Oriented Parallel Environments (ISCOPE) Conference. Springer Verlag, 1997.

C. M. Venier, C. I. Pairetti, S. M. Damian, and N. M. Nigro. On the stability analysis of the piso algorithm on collocated grids. Computers & Fluids, 147:25–40, 2017.

R. Verdugo and K. Ishihara. The steady state of sandy soils. Soils and foundations, 36(2):81–91, 1996.

A. Verruijt. Soil Mechanics. Verruijt Self Published, 2012. URL http://geo.verruijt.net/.

E. Veveakis, I. Vardoulakis, and G. Di Toro. Thermoporomechanics of creeping landslides: The 1963 vaiont slide, northern Italy. Journal of Geophysical Research: Earth Surface, 112(F3), 2007.

E. Veveakis, J. Sulem, and I. Stefanou. Modeling of fault gouges with Cosserat continuum mechanics: Influence of thermal pressurization and chemical decomposition as coseismic weakening mechanisms. Journal of Structural Geology, 38:254–264, 2012.

E. Viallet, J. Berger, P. Traversa, E. E. Haber, E. Hervé-Secourgeon, G. Hervé-Secourgeon, L. Zuchowski, and G. Dupuy. 2019-11-11 Le Teil earthquake - the ultimate missing piece of experience feedback related to a nuclear power plant built on seismic base isolation: A real earthquake. In Transactions, SMiRT-26. SMiRT, July 2022.

M. Vucetic and R. Dobry. Degradation of marine clays under cyclic loading. Journal of Geotechnical Engineering, 114(2):133–149, February 1988.

M. Vucetic and R. Dobry. Effect of soil plasticity on cyclic response. Journal of Geotechnical Engineering, 117(1):89–107, January 1991.

B. R. Wair, J. T. de Jong, and T. Shantz. Guidelines for estimation of shear wave velocity profiles. Technical Report 8, PEER, 2012. 95 pages.

A. Wakai, S. Gose, and K. Ugai. 3-d elasto-plastic finite element analysis of pile foundations subjected to lateral loading. Soil and Foundations, 39(1):97–111, Feb. 1999.

D. Wald, T. Heaton, and K. Hudnut. The slip history of the 1994 northridge, california, earthquake determined from strong-motion, teleseismic, gps, and leveling data. Bulletin of the Seismological Society of America, 86(1B):S49–S70, 1996.

M. A. Walling. Non-Ergodic Probabilistic Seismic Hazard Analysis and Spatial Simulation of Variation in Ground Motion. PhD thesis, University of California at Berkeley, Spring 2009. Under guidance of Professor Norman Abrahamson.

C. Walshaw and M. Cross. Parallel optimisation algorithms for multilevel mesh partitioning. Technical Report 99/IM/44, University of Greenwich, London, UK, 1999.

C. Wang and R. Herrmann. A numerical study of p-, sv-, and sh-wave generation in a plane layered medium. Bulletin of the Seismological Society of America, 70(4):1015–1036, 1980.

H. Wang, H. Yang, S. K. Sinha, Y. Feng, C. Luo, D. B. McCallen, and B. Jeremić. 3-D non-linear earthquake soil-structure interaction modeling of embedded small modular reactor (SMR). In Proceedings of the 24th International Conference on Structural Mechanics in Reactor Technology (SMiRT 24), Busan, South Korea, August 20-25 2017.

H. Wang, H. Yang, Y. Feng, F. Wang, and B. Jeremić. Wave potential-domain reduction method for 3D earthquake soil structure interaction. Soil Dynamics and Earthquake Engineering, 2020a. In review.

H. Wang, H. Yang, Y. Feng, and B. Jeremić. Modeling and simulation of earthquake soil structure interaction excited by inclined seismic waves. Soil Dynamics and Earthquake Engineering, 146:106720, 2021. ISSN 0267-7261. doi: https://doi.org/10.1016/j.soildyn.2021.106720. URL https://www.sciencedirect.com/science/article/pii/S0267726121001421.

M. Wang, S. Nagarajaiah, and F.-F. Sun. Dynamic characteristics and responses of damped outrigger tall buildings using negative stiffness. Journal of Structural Engineering, 146(12):04020273, 2020b. doi: 10.1061/(ASCE)ST.1943-541X.0002846. URL https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29ST.1943-541X.0002846.

Y. Wang, Z. Y. Bu, and L. Hu. Seismic behavior of precast segmental bridge columns with carbon fibre reinforcement as energy dissipation bars. In Applied Mechanics and Materials, volume 157, pages 1148–1152. Trans Tech Publ, 2012.

M. P. Ward. Language oriented programming. Computer Science Department, Science Labs, South Rd, Durham, DH1 3LE, http://www.cse.dmu.ac.uk/ mward/martin/papers/middle-out-t.pdf,, January 17 2003.

M. S. Warren, T. C. Germann, P. S. Lomdahl, D. M. Beazley, and J. K. Salmon. Avalon: an Alpha/Linux cluster achieves 10 Gflops for $ 150K. Gordon Bell Price/Performance Prize Entry, 1998.

C. Warshaw. Parallel JOSTLE User Guide. University of Greenwich, London, 1998.

K. Watanabe, F. Pisanò, and B. Jeremić. Discretization effects in the finite element simulation of seismic waves in elastic and elastic-plastic media. Engineering with Computers, 33(3):519–545, Jul 2017. ISSN 1435-5663. doi: 10.1007/s00366-016-0488-4. URL http://dx.doi.org/10.1007/s00366-016-0488-4.

J. D. Waugh. Nonlinear analysis of T-shaped concrete walls subjected to multi-directional displacements. Iowa State University, 2009.

S. Wehmeyer, F. W. Zok, C. Eberl, P. Gumbsch, N. Cohen, R. M. McMeeking, and M. R. Begley. Post-buckling and dynamic response of angled struts in elastic lattices. Journal of the Mechanics and Physics of Solids, 133:103693, 2019. ISSN 0022-5096. doi: https://doi.org/10.1016/j.jmps.2019.103693. URL https://www.sciencedirect.com/science/article/pii/S0022509619303758.

X. Wei, Z. Zhao, and J. Gu. Numerical simulations of rock mass damage induced by underground explosion. International Journal of Rock Mechanics and Mining Sciences, 46(7):1206–1213, October 2009.

G. A. Wempner. Discrete approximations related to nonlinear theories of solids. International Journal for Solids and Structures, 7:1581–1599, 1971.

M. Wenzel, O. S. Bursi, and I. Antoniadis. Optimal finite locally resonant metafoundations enhanced with nonlinear negative stiffness elements for seismic protection of large storage tanks. Journal of Sound and Vibration, 483:115488, 2020. ISSN 0022-460X. doi: https://doi.org/10.1016/j.jsv.2020.115488. URL https://www.sciencedirect.com/science/article/pii/S0022460X20303205.

R. V. Whitman. On liquefaction. In Proceedings of the 11th Int. Conf. on Soil Mechanics and Foundation Engineering, pages 1923–1926, San Francisco, 1985. Balkema, Rotterdam, The Netherlands.

E. Wiechert and K. Zoeppritz. Über Erdbebenwellen, volume 1. Nachrichten der K. Gesellschaft der Wissenschaften zu Göttingen, 1907.

N. Wiener. The homogeneous chaos. American Journal of Mathematics, 60:897–936, 1938.

M. L. Wilkins. Computer Simulation of Dynamic Phenomena. Scientific Computation. Springer, 1999. ISBN ISBN-10: 3540630708 ; ISBN-13: 978-3540630708.

K. Willam. Plasticity and elastic degradation. Lecture Notes at CU Boulder, august - decembar 1993.

K. J. Willam and E. P. Warnke. Constitutive model for the triaxial behaviour of concrete. In Proceedings IABSE Seminar on Concrete Bergamo. ISMES, 1974.

W. Wittke. Rock mechanics based on an Anisotropic Jointed Rock Model AJRM. Ernst & Sohn, Berlin, 2014.

J. P. Wolf. Dynamic Soil-Structure-Interaction. Prentice-Hall, Englewood Cliffs (NJ), 1985.

J. P. Wolf. Soil-Structure-Interaction Analysis in Time Domain. Prentice-Hall, Englewood Cliffs (NJ), 1988.

J. P. Wolf. Foundation Vibration Analysis Using Simple Physical Models. Prentice-Hall, Englewood Cliffs (NJ), 1994.

J. P. Wolf and M. Preisig. Dynamic stiffness of foundation embedded in layered halfspace based on wave propagation in cones. Earthquake Engineering and Structural Dynamics, 32(7):1075–1098, JUN 2003.

J. P. Wolf and C. M. Song. Some cornerstones of dynamic soil-structure interaction. Engineering Structures, 24(1):13–28, 2002.

K. Wong. Seismic energy dissipation of inelastic structures with tuned mass dampers. Journal of engineering mechanics, 134(2):163–172, 2008.

L. H. Wong and J. E. Luco. Tables of impedance functions and input motions for rectangular foundations. Technical report, Department of Civil Engineering, University of Southern California, Los Angeles, California, Report CE-78-15, 1978.

L. H. Wong and J. E. Luco. Tables of impedance functions for square foundations on layered media. International Journal of Soil Dynamics and Earthquake Engineering, 4(2):64–81, 1985.

D. Wood, K. Belkheir, and D. Liu. Strain softening and state parameter for sand modelling. Geotechnique, 44(3):335–339, 1994.

D. M. Wood. Geotechnical Modeling. Spoon Press, 2004. ISBN 0-415–34304.

P. Wriggers. Computational Contact Mechanics. John Wiley & Sons, 2002.

T. Wu Tai. The effect of inclusion shape on the elastic moduli of a two - phase material. International Journal of Solids and Structures, 2:1–8, 1966.

D. Xiu. Numerical Methods for Stochastic Computations. Princeton University Press, 2010.

D. Xiu and G. E. Karniadakis. The wiener–askey polynomial chaos for stochastic differential equations. SIAM Journal on Scientific Computing, 24(2):619–644, 2002. doi: 10.1137/S1064827501387826.

D. Xiu and G. E. Karniadakis. A new stochastic approach to transient heat conduction modeling with uncertainty. International Journal of Heat and Mass Transfer, 46:4681–4693, 2003.

H. Yang, S. K. Sinha, Y. Feng, D. B. McCallen, and B. Jeremić. Energy dissipation analysis of elastic-plastic materials. Computer Methods in Applied Mechanics and Engineering, 331:309–326, 2018.

H. Yang, Y. Feng, H. Wang, and B. Jeremić. Energy dissipation analysis for inelastic reinforced concrete and steel beam-columns. Engineering Structures, 197:109431, 2019a. ISSN 0141-0296. doi: https://doi.org/10.1016/j.engstruct.2019.109431. URL http://www.sciencedirect.com/science/article/pii/S0141029618323459.

H. Yang, H. Wang, Y. Feng, F. Wang, and B. Jeremić. Energy dissipation in solids due to material inelasticity, viscous coupling, and algorithmic damping. ASCE Journal of Engineering Mechanics, 145 (9), 2019b.

H. Yang, H. Wang, B. Jeremić, and J. Salamon. Earthquake soil structure interaction analysis of a gravity dam. In G. MAZZA, editor, Proceedings of the 15th International Benchmark Workshop on Numerical Analysis of Dams. ICOLD, PoLiMi, September 2019c.

Z. Yang. Three Dimensional Nonlinear Finite Element Analysis of Soil–Foundation–Structure Interaction. PhD thesis, University of California at Davis, Davis, California, September 2002.

Z. Yang and B. Jeremić. Numerical analysis of pile behaviour under lateral loads in layered elastic-plastic soils. International Journal for Numerical and Analytical Methods in Geomechanics, 26(14): 1385–1406, 2002.

Z. Yang and B. Jeremić. Numerical study of the effective stiffness for pile groups. International Journal for Numerical and Analytical Methods in Geomechanics, 27(15):1255–1276, Dec 2003.

Z. Yang and B. Jeremić. Soil layering effects on lateral pile behavior. ASCE Journal of Geotechnical and Geoenvironmental Engineering, 131(6):762–770, June 2005.

Z. Yang, A. Elgamal, and E. Parra. Computational model for cyclic mobility and associated shear deformation. Journal of Geotechnical and Geoenvironmental Engineering, 129(12):1119–1127, 2003.

M. H. M. Yassin. Nonlinear analysis of prestressed concrete sructures under monotonic and cyclic loads. PhD thesis, University of California, Berkeley, 1994.

K. Yokoyama, K. Tamura, and O. Matsuo. Design methods of bridge foundations against soil liquefaction and liquefaction induced ground flow. In 2nd Italy–Japan Workshop on Seismic Design and Retrofit of Bridges, page 23 pages, Rome, Italy, 27-28 Feb. 1997.

Y. Yoshimi and T. Kishida. A ring torsion apparatus for evaluating friction between soil and metal surfaces. Geotechnical Testing Journal, 4(4):145–152, 1981.

C. Yoshimura, J. Bielak, and Y. Hisada. Domain reduction method for three–dimensional earthquake modeling in localized regions. part II: Verification and examples. Bulletin of the Seismological Society of America, 93(2):825–840, 2003a.

C. Yoshimura, J. Bielak, and Y. Hisada. Domain reduction method for three dimensional earthquake modeling in localized regions. part ii: Verification and examples. Bulletin of the Seismological Society of America, 93(2):825–840, 2003b.

T. L. Youd and S. F. Bartlett. Case histories of lateral spreads from the 1964 Alaskan earthquake. In Proceedings of the Third Japan–U.S. Workshop on Earth Resistant Design of Lifeline Facilities and Countermeasures for Soils Liquefaction, number 91-0001 in NCEER, Feb 1. 1989.

O. Yuksei and C. Yilmaz. Design of s broadband elastic metamaterial via topologically optimized inertial amplification mechanisms. In M. Papadrakakis, M. Fragiadakis, and C. Papadimitriou, editors, EURODYN 2020, XI International Conference on Structural Dynamics, Athens, Greece, 23-26 November 2020 2020. https://doi.org/10.47964/1120.9337.19454.

R. Zaccherini, A. Colombi, A. Palermo, V. K. Dertimanis, A. Marzani, H. R. Thomsen, B. Stojadinović, and E. N. Chatzi. Locally resonant metasurfaces for shear waves in granular media. Phys. Rev. Applied, 13:034055, Mar 2020. doi: 10.1103/PhysRevApplied.13.034055. URL https://link.aps.org/doi/10.1103/PhysRevApplied.13.034055.

J. Zadeh. The role of fuzzy logic in the management of uncertainty in expert systems. Fuzzy Sets and Systems, 11:199–228, 1983.

W. Zahlten, P. Demmert, and W. B. Kratzig. An Object-Oriented Approach to Physically Nonlinear Problems in Computational Mechanics. In Computing in Civil and Building Engineering: Proceedings of the Sixth International Conference on Computing in Civil and Building Engineering, Berlin, Germany, July 12-15 1995.

G. W. Zeglinski, R. S. Han, and P. Aitchison. Object oriented matrix classes for use in a finite element code using C++. International Journal for Numerical Methods in Engineering, 37:3921–3937, 1994.

A. Zerva. Spatial variation of seismic ground motions: modeling and engineering applications. CRC Press, 2009. ISBN ISBN-10: 0849399297; ISBN-13: 978-0849399299.

A. Zerva and V. Zervas. Spatial variation of seismic ground motions: An overview. ASME Applied Mechanics Revue, 55(3):271–297, May 2002.

L. Zhang and A. K. Chopra. Three-dimensional analysis of spatially varying ground motions around a uniform canyon in a homogeneous half-space. Earthquake Engineering & Structural Dynamics, 20 (10):911–926, 1991.

L. Zhang, M. McVay, and P. Lai. Centrifuge testing of vertically loaded battered pile groups in sand. Geotechnical Testing Journal, 21(4):281–288, 1998. No paper yet.

L. Zhang, M. McVay, and P. Lai. Numerical analysis of laterally loaded 3x3 to 7x3 pile groups in sands. Journal of Geotechnical and Geoenvironmental Engineering, 125(11):936–946, Nov. 1999.

Y.-y. Zhang, K. A. Harries, and W.-c. Yuan. Experimental and numerical investigation of the seismic performance of hollow rectangular bridge piers constructed with and without steel fiber reinforced concrete. Engineering Structures, 48:255–265, 2013.

J. Zhao, H. Li, M. Wu, and T. Li. Dynamic uniaxial compression tests on a granite. International Journal of Rock Mechanics and Mining Sciences, 36(2):273–277, February 1999.

M. Zhou, G. Ravichandran, and A. Rosakis. Dynamically propagating shear bands in impact-loaded prenotched plates-ii. numerical simulations. Journal of the Mechanics and Physics of Solids, 44(6): 1007–1032, 1996.

L. Zhu and L. A. Rivera. A note on the dynamic and static displacements from a point source in multilayered media. Geophysical Journal International, 148(3):619–627, 2002.

Z. Zhu, I. Ahmad, and A. Mirmiran. Fiber element modeling for seismic performance of bridge columns made of concrete-filled frp tubes. Engineering structures, 28(14):2023–2035, 2006.

H. Ziegler. Discussion of some objections to thermomechanical orthogonality. Ingenieur-Archiv, 50(3): 149–164, 1981.

H. Ziegler and C. Wehrli. The derivation of constitutive relations from the free energy and the dissipation function. Advances in Applied Mechanics, 25:183–238, 1987.

O. Zienkiewicz, A. Chan, M. Pastor, B. Schrefler, and T. Shiomi. Computational Geomechanics–with Special Reference to Earthquake Engineering. John Wiley and Sons Ltd., Baffins Lane, Chichester, West Sussex PO19 IUD, England, 1999a.

O. C. Zienkiewicz. The Finite Element Method. McGraw - Hill Book Company, 3rd edition, 1977. ISBN 0-07-084072-5.

O. C. Zienkiewicz and T. Shiomi. Dynamic behaviour of saturated porous media; the generalized Biot formulation and its numerical solution. International Journal for Numerical and Analytical Methods in Geomechanics, 8:71–96, 1984.

O. C. Zienkiewicz and R. L. Taylor. The Finite Element Method, volume 1. McGraw - Hill Book Company, fourth edition, 1991a.

O. C. Zienkiewicz and R. L. Taylor. The Finite Element Method, volume 2. McGraw - Hill Book Company, Fourth edition, 1991b.

O. C. Zienkiewicz and R. L. Taylor. The Finite Element Method, Volume 1, The Basis, 5th Edition. Butterworth Heinemann, London, 2000.

O. C. Zienkiewicz, A. H. C. Chan, M. Pastor, B. A. Schrefler, and T. Shiomi. Computational Geomechanics with Special Reference to Earthquake Engineering. John Wiley and Sons., 1999b. ISBN 0-471-98285-7.

T. Zimmermann and D. Eyheramendy. Object–oriented finite elements I. principles of symbolic derivations and automatic programming. Computer Methods in Applied Mechanics and Engineering, 132:259–276, 1996.

T. Zimmermann, Y. Dubois-Pèlerin, and P. Bomme. Object oriented finite element programming: I. governing principles. Computer Methods in Applied Mechanics and Engineering, 98:291–303, 1992a.

T. Zimmermann, Y. Dubois-Pèlerin, and P. Bomme. Object Oriented Finite Element Programming: Theory and Smalltalk V Implementation for FEM_Objects PC 01. Elmepress International, P.O.Box 2 CH 1015 Lausanne 15, Switzerland, 1992b.

Part 700

Appendix

Appendix 701

Useful Formulae

(1985-1989-1993-2021-)

701.1 Chapter Summary and Highlights

701.2 Stress and Strain

This section reviews small deformation stress and strain measures used in this report.

701.2.1 Stress

In this work, the tensile stress is assumed positive, and in general we follow classical strength of materials (mechanics of materials) conventions for stress and strain. The stress tensor σ_{ij} is defined as

$$\sigma_{ij} = \lim_{A_i \to 0} \frac{F_j}{A_i} \tag{701.1}$$

where F_j is a traction (force) in the j direction and A_i is an infinitesimal surface area with normal in i direction. Cauchy stress tensor has a total of nine components, six of which are independent (symmetry $\sigma_{ij} = \sigma_{ji}$):

$$\boldsymbol{\sigma} = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{pmatrix} = \begin{pmatrix} \sigma_{x} & \sigma_{xy} & \sigma_{zx} \\ \sigma_{xy} & \sigma_{y} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{yz} & \sigma_{z} \end{pmatrix} \tag{701.2}$$

In small deformation theory, this stress is symmetric, that is, $\sigma_{xy} = \sigma_{yx}$, $\sigma_{yz} = \sigma_{zy}$, and $\sigma_{zx} = \sigma_{xz}$. There are only six independent components and sometimes the stress can be expressed in the vector form

$$\boldsymbol{\sigma} = \left\{\sigma_{xx},\ \sigma_{yy},\ \sigma_{zz}, \sigma_{xy},\ \sigma_{yz},\ \sigma_{zx}\right\} \tag{701.3}$$

The principle stresses σ_1, σ_2, and σ_3 ($\sigma_1 \geq \sigma_2 \geq \sigma_3$) are the eigenvalues of the symmetric tensor σ_{ij} in Equation 701.2 and can be obtained by solving the equation

$$\begin{vmatrix} \sigma_{xx} - \sigma & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} - \sigma & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} - \sigma \end{vmatrix} = \mathbf{0} \tag{701.4}$$

or in alternative form

$$\sigma^3 - I_1\sigma^2 - I_2\sigma - I_3 = 0 \tag{701.5}$$

The three first-type stress invariants are then

$$\begin{aligned} I_1 &= \sigma_{ii} \\ &= \sigma_{xx} + \sigma_{yy} + \sigma_{zz} \\ &= \sigma_1 + \sigma_2 + \sigma_3 \end{aligned} \tag{701.6}$$

$$
\begin{aligned}
I_2 &= \frac{1}{2}\sigma_{ij}\sigma_{ji} \\
&= -(\sigma_{xx}\sigma_{yy} + \sigma_{yy}\sigma_{zz} + \sigma_{zz}\sigma_{xx}) + (\sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{zx}^2) \\
&= -(\sigma_1\sigma_2 + \sigma_2\sigma_3 + \sigma_3\sigma_1)
\end{aligned}
\tag{701.7}
$$

$$
\begin{aligned}
I_3 &= \frac{1}{3}\sigma_{ij}\sigma_{jk}\sigma_{ki} = \det(\sigma_{ij}) \\
&= \sigma_{xx}\sigma_{yy}\sigma_{zz} + 2\sigma_{xy}\sigma_{yz}\sigma_{zx} - (\sigma_{xx}\sigma_{yz}^2 + \sigma_{yz}\sigma_{zx}^2 + \sigma_{zz}\sigma_{xy}^2) \\
&= \sigma_1\sigma_2\sigma_3
\end{aligned}
\tag{701.8}
$$

The stress σ_{ij} can be decomposed into the hydrostatic stress $\sigma_m\delta_{ij}$ and deviatoric stress s_{ij} as $\sigma_{ij} = \sigma_m\delta_{ij} + s_{ij}$, with the definitions

$$
\sigma_m = \frac{1}{3}I_1, \quad s_{ij} = \sigma_{ij} - \frac{1}{3}\sigma_{kk}\delta_{ij}
\tag{701.9}
$$

where δ_{ij} is the Kronecker operator such that $\delta_{ij} = 1$ for $i = j$ and $\delta_{ij} = 0$ for $i \neq j$.

Since both hydrostatic and deviatoric stresses are stress tensors, they have their own coordinate-independent stress invariants respectively. The three invariants of the hydrostatic stress are

$$
I_1 = 3\sigma_m = I_1, \quad I_2 = -3\sigma_m^2 = -\frac{1}{3}I_1^2, \quad I_3 = \sigma_m^3 = \frac{1}{27}I_1^3
\tag{701.10}
$$

Since I_1, I_2 and I_3 are all simple functions of I_1, the hydrostatic stress state can therefore be represented by only one variable I_1.

The three eigenvalues of the deviatoric stresses s_{ij} are called principal deviatoric stresses, with the order $s_1 \geq s_2 \geq s_3$. The three invariants of the deviatoric stress are

$$
J_1 = s_{ii} = 0
\tag{701.11}
$$

$$
\begin{aligned}
J_2 &= \frac{1}{2}s_{ij}s_{ji} \\
&= \frac{1}{3}I_1^2 + I_2 \\
&= \frac{1}{6}[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2] \\
&= -(s_{xx}s_{yy} + s_{yy}s_{zz} + s_{zz}s_{xx}) + (s_{xy}^2 + s_{yz}^2 + s_{zx}^2) \\
&= \frac{1}{2}(s_1^2 + s_2^2 + s_3^2) = -(s_1s_2 + s_2s_3 + s_3s_1)
\end{aligned}
\tag{701.12}
$$

$$
\begin{aligned}
J_3 &= \frac{1}{3}s_{ij}s_{jk}s_{ki} = \det(s_{ij}) \\
&= I_3 + \frac{1}{3}I_1I_2 + \frac{2}{27}I_1^3 = I_3 - \frac{1}{3}I_1J_2 - \frac{1}{27}I_1^3 \\
&= \frac{1}{27}(2\sigma_1 - \sigma_2 - \sigma_3)(2\sigma_2 - \sigma_3 - \sigma_1)(2\sigma_3 - \sigma_1 - \sigma_2) \\
&= s_{xx}s_{yy}s_{zz} + 2s_{xy}s_{yz}s_{zx} - (s_{xx}s_{yz}^2 + s_{yy}s_{zx}^2 + s_{zz}s_{xy}^2) \\
&= s_1s_2s_3
\end{aligned}
\tag{701.13}
$$

The deviatoric stress state can therefore be represented by only two variables J_2 and J_3.

Combining hydrostatic and deviatoric stress, we can conclude that the stress state can be be represented by three variables I_1, J_2 and J_3. Using the three invariants (I_1, J_2, J_3) or its equivalents instead of the nine components of σ_{ij} is widely used in geomechanics.

The stress state may also be described in three dimensional space (p, q, θ_σ), defined as

$$p = -\frac{1}{3} I_1 \tag{701.14}$$

$$q = \sqrt{3J_2} \tag{701.15}$$

$$\theta_\sigma = \frac{1}{3} \arccos\left(\frac{3\sqrt{3}}{2} \frac{J_3}{\sqrt{J_2^3}}\right) \tag{701.16}$$

where $\theta_{\sigma_{ij}}$ is the stress Lode's angle $(0 \leq \theta_{\sigma_{ij}} \leq \pi/3)$. A stress state with $\theta_\sigma = 0$ corresponds to the meridian of conventional triaxial compression (CTC), while $\theta_\sigma = \pi/3$ to the meridian of conventional triaxial extension (CTE). The relationship between $(\sigma_1, \sigma_2, \sigma_3)$ and (p, q, θ_σ) is

$$\begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{pmatrix} = -p + \frac{2}{3} q \begin{pmatrix} \cos\theta_\sigma \\ \cos(\theta_\sigma - \frac{2}{3}\pi) \\ \cos(\theta_\sigma + \frac{2}{3}\pi) \end{pmatrix} \tag{701.17}$$

The line of the principal stress space diagonal is called hydrostatic axis. Any plane perpendicular to the hydrostatic axis is an deviatoric plane, or π plane. The Haigh-Westergaard three dimensional stress coordinate system $(\xi, \rho, \theta_\sigma)$ Chen and Han (1988a), is defined as

$$\xi = \frac{I_1}{\sqrt{3}} = -\sqrt{3} p \tag{701.18}$$

$$\rho = \sqrt{2J_2} = \sqrt{\frac{2}{3}} q \tag{701.19}$$

The Haigh-Westergaard invariants have physical meanings. ξ is the distance of the deviatoric plane to the origin of the Haigh-Westergaard coordinates, and ρ is the distance of a stress point to the hydrostatic line and represents the magnitude of the deviatoric stress. The projections of the axes σ_1, σ_2 and σ_3 on the deviatoric plane are assumed σ'_1, σ'_2 and σ'_3 respectively. (ρ, θ_σ) is the polar coordinate system in the deviatoric plane with the σ'_1 the polar axis and θ_σ the polar angle. The relationship between $(\sigma_1, \sigma_2, \sigma_3)$ and $(\xi, \rho, \theta_\sigma)$ is

$$\begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{pmatrix} = \frac{1}{\sqrt{3}} \xi + \sqrt{\frac{2}{3}} \rho \begin{pmatrix} \cos\theta_\sigma \\ \cos(\theta_\sigma - \frac{2}{3}\pi) \\ \cos(\theta_\sigma + \frac{2}{3}\pi) \end{pmatrix} \tag{701.20}$$

701.2.2 Strain

Point $P(x_i)$ and nearby point $Q(x_i + \mathrm{d}x_i)$ displace due to applied loading to new positions $P(x_i + U_i)$ and $Q(u_i + (\partial u_i/\partial x_j)\mathrm{d}x_j)$. We can define a displacement gradient tensor $u_{i,j}$ as

$$u_{i,j} = \frac{\partial u_i}{\partial x_j} \tag{701.21}$$

Matrix form of the displacement gradient can decomposed into the symmetric and antisymmetric parts

$$\begin{pmatrix} u_{1,1} & u_{1,2} & u_{1,3} \\ u_{2,1} & u_{2,2} & u_{2,3} \\ u_{3,1} & u_{3,2} & u_{3,3} \end{pmatrix} = \begin{pmatrix} u_{1,1} & \frac{1}{2}(u_{1,2}+u_{2,1}) & \frac{1}{2}(u_{1,3}+u_{3,1}) \\ \frac{1}{2}(u_{2,1}+u_{1,2}) & u_{2,2} & \frac{1}{2}(u_{2,3}+u_{3,2}) \\ \frac{1}{2}(u_{3,1}+u_{1,3}) & \frac{1}{2}(u_{3,2}+u_{2,3}) & u_{3,3} \end{pmatrix} + \begin{pmatrix} 0 & \frac{1}{2}(u_{1,2}-u_{2,1}) & \frac{1}{2}(u_{1,3}-u_{3,1}) \\ \frac{1}{2}(u_{2,1}-u_{1,2}) & 0 & \frac{1}{2}(u_{2,3}-u_{3,2}) \\ \frac{1}{2}(u_{3,1}-u_{1,3}) & \frac{1}{2}(u_{3,2}-u_{2,3}) & 0 \end{pmatrix} \tag{701.22}$$

or

$$u_{i,j} = \epsilon_{ij} + w_{ij} \tag{701.23}$$

where

$$\epsilon_{ij} = \frac{1}{2}\left(u_{i,j} + u_{j,i}\right) \tag{701.24}$$

$$w_{ij} = \frac{1}{2}\left(u_{i,j} - u_{j,i}\right) \tag{701.25}$$

The symmetric part of the deformation gradient tensor, ϵ_{ij}, is the small deformation strain tensor [1], while the antisymmetric part of the deformation gradient tensor, w_{ij}, is the rotation motion tensor. The matrix form of the strain ϵ_{ij} is

$$\boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_{xx} & \epsilon_{xy} & \epsilon_{xz} \\ \epsilon_{yx} & \epsilon_{yy} & \epsilon_{yz} \\ \epsilon_{zx} & \epsilon_{zy} & \epsilon_{zz} \end{pmatrix} = \begin{pmatrix} \epsilon_x & \frac{1}{2}\gamma_{xy} & \frac{1}{2}\gamma_{xz} \\ \frac{1}{2}\gamma_{yx} & \epsilon_y & \frac{1}{2}\gamma_{yz} \\ \frac{1}{2}\gamma_{zx} & \frac{1}{2}\gamma_{zy} & \epsilon_z \end{pmatrix} \tag{701.26}$$

The engineering strain is usually expressed in the vector form

$$\boldsymbol{\epsilon} = \left\{\epsilon_x,\ \epsilon_y,\ \epsilon_z, \gamma_{xy},\ \gamma_{yz},\ \gamma_{zx}\right\}^T \tag{701.27}$$

Note that the engineering shear strain γ_{ij} is the double of the corresponding strain component ϵ_{ij}.

---

[1] Here the second and higher order derivative terms are neglected due to the small deformation assumption.

Similar to the stress tensor, the strain tensor also has three principle strains $\epsilon_i (\epsilon_1 \geq \epsilon_2 \geq \epsilon_3)$, and three strain invariants I_1', I_2', and I_3', defined as

$$\begin{aligned}
I_1' &= \epsilon_{ii} \\
&= \epsilon_v \\
&= \epsilon_{xx} + \epsilon_{yy} + \epsilon_{zz} \\
&= \epsilon_1 + \epsilon_2 + \epsilon_3 \qquad (701.28)
\end{aligned}$$

$$\begin{aligned}
I_2' &= \frac{1}{2}\epsilon_{ij}\epsilon_{ji} \\
&= -(\epsilon_{xx}\epsilon_{yy} + \epsilon_{yy}\epsilon_{zz} + \epsilon_{zz}\epsilon_{xx}) + (\epsilon_{xy}^2 + \epsilon_{yz}^2 + \epsilon_{zx}^2) \\
&= -(\epsilon_1\epsilon_2 + \epsilon_2\epsilon_3 + \epsilon_3\epsilon_1) \qquad (701.29)
\end{aligned}$$

$$\begin{aligned}
I_3' &= \frac{1}{3}\epsilon_{ij}\epsilon_{jk}\epsilon_{ki} = \det(\epsilon_{ij}) \\
&= \epsilon_{xx}\epsilon_{yy}\epsilon_{zz} + 2\epsilon_{xy}\epsilon_{yz}\epsilon_{zx} - (\epsilon_{xx}\epsilon_{yz}^2 + \epsilon_{yz}\epsilon_{zx}^2 + \epsilon_{zz}\epsilon_{xy}^2) \\
&= \epsilon_1\epsilon_2\epsilon_3 \qquad (701.30)
\end{aligned}$$

The first strain invariant is also called the volumetric strain ϵ_v.

The strain ϵ_{ij} can be decomposed into the hydrostatic strain $\epsilon_m\delta_{ij}$ and deviatoric strain e_{ij} through $\epsilon_{ij} = \epsilon_m\delta_{ij} + e_{ij}$ where:

$$\epsilon_m = \frac{1}{3}I_1', \quad e_{ij} = \epsilon_{ij} - \frac{1}{3}\epsilon_{kk}\delta_{ij} \tag{701.31}$$

Since both hydrostatic and deviatoric strains are strain tensors, they have their own strain invariants respectively. The three invariants of the hydrostatic strain are

$$I_1' = 3\epsilon_m = I_1', \quad I_2' = -3\epsilon_m^2 = -\frac{1}{3}(I_1')^2, \quad I_3' = \epsilon_m^3 = \frac{1}{27}(I_1')^3 \tag{701.32}$$

The hydrostatic strain state can therefore be represented by only one variable I_1'.

The three eigenvalues of the deviatoric strains e_{ij} are called principal deviatoric strains, with the order $e_1 \geq e_2 \geq e_3$. The three invariants of the deviatoric strain are

$$J_1' = e_{ii} = 0 \tag{701.33}$$

$$\begin{aligned}
J_2' &= \frac{1}{2}e_{ij}e_{ji} \\
&= \frac{1}{3}(I_1')^2 + I_2' \\
&= \frac{1}{6}[(\epsilon_1 - \epsilon_2)^2 + (\epsilon_2 - \epsilon_3)^2 + (\epsilon_3 - \epsilon_1)^2] \\
&= -(e_{xx}e_{yy} + e_{yy}e_{zz} + e_{zz}e_{xx}) + (e_{xy}^2 + e_{yz}^2 + e_{zx}^2) \\
&= \frac{1}{2}(e_1^2 + e_2^2 + e_3^2) = -(e_1e_2 + e_2e_3 + e_3e_1) \qquad (701.34)
\end{aligned}$$

$$\begin{aligned}
J'_3 &= \frac{1}{3}e_{ij}e_{jk}e_{ki} = \det(e_{ij}) \\
&= I'_3 + \frac{1}{3}I'_1I'_2 + \frac{2}{27}(I'_1)^3 = I_3 - \frac{1}{3}I'_1J'_2 - \frac{1}{27}(I'_1)^3 \\
&= \frac{1}{27}(2\epsilon_1 - \epsilon_2 - \epsilon_3)(2\epsilon_2 - \epsilon_3 - \epsilon_1)(2\epsilon_3 - \epsilon_1 - \epsilon_2) \\
&= e_{xx}e_{yy}e_{zz} + 2e_{xy}e_{yz}e_{zx} - (e_{xx}e^2_{yz} + e_{yy}s^2_{zx} + e_{zz}e^2_{xy}) \\
&= e_1e_2e_3 \qquad (701.35)
\end{aligned}$$

The deviatoric strain state can therefore be represented by only two variables J'_2 and J'_3.

Combining the hydrostatic and deviatoric strain, we can conclude that the strain state can be be represented by three variables I'_1, J'_2 and J'_3.

Strain state may also be represented with another three invariant $(\epsilon_p, \epsilon_q, \theta_\epsilon)$, defined as

$$\epsilon_p = -I'_1 = -\epsilon_v \qquad (701.36)$$

$$\epsilon_q = 2\sqrt{\frac{J'_2}{3}} \qquad (701.37)$$

$$\theta_\epsilon = \frac{1}{3}\arccos\left(\frac{3\sqrt{3}}{2}\frac{J'_3}{\sqrt{(J'_2)^3}}\right) \qquad (701.38)$$

where θ_ϵ is the strain Lode's angle and $0 \leq \theta_\epsilon \leq \pi/3$. The relationship between $(\epsilon_1, \epsilon_2, \epsilon_3)$ and $(\epsilon_p, \epsilon_q, \theta_\epsilon)$ is

$$\begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{pmatrix} = -\frac{1}{3}\epsilon_p + \sqrt{\frac{3}{2}}\epsilon_q \begin{pmatrix} \cos\theta_\epsilon \\ \cos(\theta_\epsilon - \frac{2}{3}\pi) \\ \cos(\theta_\epsilon + \frac{2}{3}\pi) \end{pmatrix} \qquad (701.39)$$

701.3 Derivatives of Stress Invariants

In this part of the Appendix, we shall derive some useful formulae, that are rarely found[2] in texts treating elasto–plastic problems in mechanics of solid continua.

First derivative of I_1 with respect to stress tensor σ_{ij}:

$$\frac{\partial I_1}{\partial \sigma_{ij}} = \frac{\partial \sigma_{kk}}{\partial \sigma_{ij}} = \delta_{ij}$$

First derivative of J_{2D} with respect to stress tensor σ_{ij}:

---

[2] if found at all.

$$
\begin{aligned}
\frac{\partial J_{2D}}{\partial \sigma_{ij}} = \frac{\partial(\frac{1}{2}s_{mn}s_{nm})}{\partial \sigma_{ij}} = \frac{1}{2}\frac{\partial s_{mn}}{\partial \sigma_{ij}}s_{nm} + \frac{1}{2}\frac{\partial s_{nm}}{\partial \sigma_{ij}}s_{mn} = \\
= \frac{\partial s_{nm}}{\partial \sigma_{ij}}s_{mn} = \frac{\partial(\sigma_{nm} - \frac{1}{3}\sigma_{kk}\delta_{nm})}{\partial \sigma_{ij}}s_{mn} = (\delta_{ni}\delta_{jm} - \frac{1}{3}\delta_{nm}\delta_{ki}\delta_{jk})s_{mn} = \\
= (\delta_{ni}\delta_{jm} - \frac{1}{3}\delta_{nm}\delta_{ij})s_{mn} = \delta_{ni}\delta_{jm}s_{nm} - \frac{1}{3}\delta_{nm}\delta_{ij}s_{mn} = s_{ij}{}^{\dagger}
\end{aligned}
$$

First derivative of J_{3D} with respect to stress tensor σ_{pq}:

$$
\begin{aligned}
\frac{\partial J_{3D}}{\partial \sigma_{pq}} = \frac{\partial(\frac{1}{3}s_{ij}s_{jk}s_{ki})}{\partial \sigma_{pq}} = \frac{1}{3}\frac{\partial s_{ij}}{\partial \sigma_{pq}}s_{jk}s_{ki} + \frac{1}{3}\frac{\partial s_{jk}}{\partial \sigma_{pq}}s_{ij}s_{ki} + \frac{1}{3}\frac{\partial s_{ki}}{\partial \sigma_{pq}}s_{ij}s_{jk} = \\
= \frac{\partial s_{ij}}{\partial \sigma_{pq}}s_{jk}s_{ki} = \frac{\partial(\sigma_{ij} - \frac{1}{3}\sigma_{kk}\delta_{ij})}{\partial \sigma_{pq}}s_{jk}s_{ki} = (\delta_{ip}\delta_{qj} - \frac{1}{3}\delta_{ij}\delta_{kp}\delta_{qk})s_{jk}s_{ki} = \\
= \delta_{ip}\delta_{qj}s_{jk}s_{ki} - \frac{1}{3}\delta_{ij}\delta_{kp}\delta_{qk}s_{jk}s_{ki} = s_{qk}s_{kp} - \frac{2}{3}\delta_{pq}J_{2D} = t_{pq}{}^{\ddagger}
\end{aligned}
$$

First derivative of s_{pq} with respect to stress tensor σ_{mn}, or second derivative of J_{2D} with respect to stress tensors σ_{pq} and σ_{mn}:

$$
\begin{aligned}
\frac{\partial s_{pq}}{\partial \sigma_{mn}} = \frac{\partial\left(\sigma_{pq} - \frac{1}{3}\delta_{pq}\sigma_{kk}\right)}{\partial \sigma_{mn}} = \frac{\partial\left(\left(\delta_{mp}\delta_{nq} - \frac{1}{3}\delta_{pq}\delta_{mn}\right)\sigma_{mn}\right)}{\partial \sigma_{mn}} = \\
= \left(\delta_{mp}\delta_{nq} - \frac{1}{3}\delta_{pq}\delta_{mn}\right) = p_{pqmn}
\end{aligned}
$$

Second derivative of J_{3D} with respect to stress tensors σ_{pq} and σ_{mn}:

$$
\begin{aligned}
\frac{\partial t_{pq}}{\partial \sigma_{mn}} = \frac{\partial\left(s_{qk}s_{kp} - \frac{2}{3}\delta_{pq}J_{2D}\right)}{\partial \sigma_{mn}} = \frac{\partial\left(s_{qk}s_{kp}\right)}{\partial \sigma_{mn}} - \frac{\partial\left(\frac{2}{3}\delta_{pq}J_{2D}\right)}{\partial \sigma_{mn}} = \\
= \frac{\partial\left(s_{qk}s_{kp}\right)}{\partial \sigma_{mn}} - \frac{2}{3}\delta_{pq}\frac{\partial J_{2D}}{\partial \sigma_{mn}} = \frac{\partial s_{qk}}{\partial \sigma_{mn}}s_{kp} + s_{qk}\frac{\partial s_{kp}}{\partial \sigma_{mn}} - \frac{2}{3}\delta_{pq}s_{mn} = \\
= \left(\delta_{qm}\delta_{nk} - \frac{1}{3}\delta_{qk}\delta_{nm}\right)s_{kp} + s_{qk}\left(\delta_{km}\delta_{np} - \frac{1}{3}\delta_{kp}\delta_{nm}\right) - \frac{2}{3}\delta_{pq}s_{mn} = \\
= \left(\delta_{qm}s_{np} - \frac{1}{3}s_{qp}\delta_{nm}\right) + \left(s_{qm}\delta_{np} - \frac{1}{3}s_{qp}\delta_{nm}\right) - \frac{2}{3}\delta_{pq}s_{mn} = \\
= s_{np}\delta_{qm} + s_{qm}\delta_{np} - \frac{2}{3}s_{qp}\delta_{nm} - \frac{2}{3}\delta_{pq}s_{mn} = w_{pqmn}
\end{aligned}
$$

Multiplying stiffness tensor E_{ijkl} with compliance tensor D_{klpq}:

[†] because $\delta_{nm}\delta_{ij}s_{mn} \equiv 0$

[‡] since $\frac{1}{3}\delta_{ij}\delta_{kp}\delta_{qk}s_{jk}s_{ki} = \frac{1}{3}\delta_{kp}\delta_{qk}s_{ik}s_{ki} = \frac{1}{3}\delta_{qp}s_{ik}s_{ki} = \frac{2}{3}\delta_{pq}J_{2D}$ see also Chen and Han (1988a) page 222

$$
\begin{aligned}
E_{ijkl} D_{klpq} & = \\
\frac{E}{2(1+\nu)} \frac{1+\nu}{2E} \left(\frac{2\nu}{1-2\nu} \delta_{ij} \delta_{kl} + \delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk} \right) \left(\frac{-2\nu}{1+\nu} \delta_{kl} \delta_{pq} + \delta_{kp} \delta_{lq} + \delta_{kq} \delta_{lp} \right) & = \\
\frac{1}{4} \left(\delta_{ik} \delta_{jl} \delta_{kp} \delta_{lq} + \delta_{il} \delta_{jk} \delta_{kp} \delta_{lq} + \delta_{ik} \delta_{jl} \delta_{kq} \delta_{lp} + \delta_{il} \delta_{jk} \delta_{kq} \delta_{lp} \right) & + \\
+ \frac{\nu}{2(1-2\nu)} \left(\delta_{ij} \delta_{kl} \delta_{kp} \delta_{lq} + \delta_{ij} \delta_{kl} \delta_{kq} \delta_{lp} \right) - \frac{\nu}{2(1+\nu)} \left(\delta_{ik} \delta_{jl} \delta_{kl} \delta_{pq} + \delta_{il} \delta_{jk} \delta_{kl} \delta_{pq} \right) & - \\
- \frac{\nu^2}{(1-2\nu)(1+\nu)} \delta_{ij} \delta_{kl} \delta_{kl} \delta_{pq} & = \\
\frac{1}{2} \left(\delta_{ip} \delta_{jq} + \delta_{iq} \delta_{jp} \right) & + \\
+ \frac{\nu}{2(1-2\nu)} \left(\delta_{ij} \delta_{kq} \delta_{kp} + \delta_{ij} \delta_{kp} \delta_{kq} \right) - \frac{\nu}{2(1+\nu)} \left(\delta_{il} \delta_{jl} \delta_{pq} + \delta_{il} \delta_{jl} \delta_{pq} \right) & - \\
- \frac{3\nu^2}{(1-2\nu)(1+\nu)} \delta_{ij} \delta_{pq} & = \\
\frac{1}{2} \left(\delta_{ip} \delta_{jq} + \delta_{iq} \delta_{jp} \right) & + \\
+ \frac{\nu}{2(1-2\nu)} \left(\delta_{ij} \delta_{pq} + \delta_{ij} \delta_{pq} \right) - \frac{\nu}{2(1+\nu)} \left(\delta_{ij} \delta_{pq} + \delta_{ij} \delta_{pq} \right) & - \\
- \frac{3\nu^2}{(1-2\nu)(1+\nu)} \delta_{ij} \delta_{pq} & = \\
\frac{1}{2} \left(\delta_{ip} \delta_{jq} + \delta_{iq} \delta_{jp} \right) & + \\
+ \frac{\nu}{(1-2\nu)} \delta_{ij} \delta_{pq} - \frac{\nu}{2(1+\nu)} \delta_{ij} \delta_{pq} & - \\
- \frac{3\nu^2}{(1-2\nu)(1+\nu)} \delta_{ij} \delta_{pq} & = \\
\frac{1}{2} \left(\delta_{ip} \delta_{jq} + \delta_{iq} \delta_{jp} \right) & + \\
+ \frac{\nu(1+\nu) - \nu(1-2\nu) - 3\nu^2}{(1-2\nu)(1+\nu)} \delta_{ij} \delta_{pq} & = \\
\frac{1}{2} \left(\delta_{ip} \delta_{jq} + \delta_{iq} \delta_{jp} \right) & = I^{sym}_{ijpq}
\end{aligned}
$$

Appendix 702

The nDarray Programming Tool

(1993-1995-1996-1999-)

702.1 Chapter Summary and Highlights

Material in this chapter is based on the following publications Jeremić (1993); Jeremić and Sture (1998).

This section describes a programming tool, nDarray, which is designed using an Object Oriented Paradigm (OOP) and implemented in the C++ programming language. Finite element equations, represented in terms of multidimensional tensors are easily manipulated and programmed. The usual matrix form of the finite element equations are traditionally coded in FORTRAN, which makes it difficult to build and maintain complex program systems. Multidimensional data systems and their implementation details are seldom transparent and thus not easily dealt with and usually avoided. On the other hand, OOP together with efficient programming in C++ allows building new concrete data types, namely tensors of any order, thus hiding the lower level implementation details. These concrete data types prove to be quite useful in implementing complicated tensorial formulae associated with the numerical solution of various elastic and elastoplastic problems in solid mechanics. They permit implementing complex nonlinear continuum mechanics theories in an orderly manner. Ease of use and the immediacy of the nDarray programming tool in constitutive driver programming and in building finite element classes will be shown.

702.2 Introduction

In implementing complex programming systems for finite element computations, the analyst is usually faced with the challenge of transforming complicated tensorial formulae to a matrix form. Considerable amount of time in solving problems by the finite element method is often devoted to the actual implementation process. If one decides to use FORTRAN, a number of finite element and numerical libraries are readily available. Although quick results can be produced in solving simpler problems, when implementing complex small deformation elastoplastic or large deformation elastic and elastoplastic algorithms, C++ provides clear benefits.

Some of the improvements C++ provides over C and FORTRAN are classes for encapsulating abstractions, the possibility of building user–defined concrete data types and operator overloading for expressing complex formulae in a natural way. In the following we shall show that the nDarray tool will allow analysts to be a step closer to the problem space and a step further away from the underlying machine.

As most analysts know, the intention (Stroustrup, 1994) behind C++ was not to replace C. Instead, C was extended with far more freedom given to the program designer and implementor. In C and FORTRAN, large applications become collections of programs and functions, order and the structure are left to the programmer. The C++ programming language embodies the OOP, which can be used to simplify and organize complex programs. One can build a hierarchy of derived classes and nest classes inside other classes. A concern in C and FORTRAN programming languages is handling data type conflicts and data which are being operated on or passed. The C++ programming language extends the definition of type to include abstract data types. With abstract data types, data can be encapsulated with the methods that operate on it. The C++ programming language offers structure and mechanisms to handle larger, more complex programming systems. Object Oriented technology, with function and operator overloading, inheritance and other features, provides means of attacking a problem in a natural way. Once basic classes are implemented, one can concentrate on the physics of a problem. By building further abstract data types one can describe the physics of a problem rather that spend time on the lower level programming issues. One should keep in mind the adage, credited to the original designer and implementor of C++ programming language, Bjarne Stroustrup: "C makes it easy to shoot yourself in the foot, C++ makes it harder, but when you do, it blows away your whole leg".

Rather than attempting here to give a summary of Object Oriented technology we will suggest useful references for readers who wish to explore the subject in greater depth (Booch, 1994). The current language definition is given in the *Working Paper for Draft Proposed International Standard for Information Systems–Programming Language C++* (ANS, 1995). Detailed description of language evolution and main design decisions are given by Stroustrup (1994). Useful sets of techniques, explanations and directions for designing and implementing robust C++ code are given in books (Coplien, 1992) (Eckel, 1989) and journal articles (Koenig, 1989 - 1993) (Various Authors, 1991-).

Increased interest in using Object Oriented techniques for finite element programming has resulted in a number (Donescu and Laursen, 1996) (Eyheramendy and Zimmermann, 1996) (Forde et al., 1990) (Miller, 1991) (Pidaparti and Hudli, 1993) (Scholz, 1992) (Zeglinski et al., 1994) of experimental developments and implementations. Programming techniques used in some of the papers are influenced by the FORTRAN programming style. Examples provided in some of the above mentioned papers are readable by C++ experts only. It appears that none of the authors have used Object Oriented techniques for complex elastoplasticity computations.

702.3 nDarray Programming Tool

702.3.1 Introduction to the nDarray Programming Tool

The nDarray programming tool is a set of classes written in the C++ programming language. The main purpose of the package is to facilitate algebraic manipulations with matrices, vectors and tensors that are often found in computer codes for solving engineering problems. The package is designed and implemented using the Object Oriented philosophy. Great care has been given to the problem of cross–platform and cross–compiler portability. Currently, the nDarray set of classes has been tested and running under the following C++ compilers:

- Sun CC on SunOS and Solaris platforms,
- IBM xlC on AIX RISC/6000 platforms,
- Borland C++ and Microsoft C++ on DOS/Windows platforms,
- CodeWarrior C++ on Power Macintosh platform,
- GNU g++ on SunOS, SOLARIS, LINUX, AIX, HPUX and AMIGA platforms.

702.3.2 Abstraction Levels

nDarray tool has the following simple class hierarchy:

```
nDarray_rep, nDarray
    matrix
        vector
    tensor
```

Indentation of class names implies the inheritance level. For example, class vector is derived from class matrix, which, in turn is derived from classes nDarray and nDarray_rep. The idea is to subdivide classes into levels of abstraction, and hide the implementation from end users. This means that the end user can use the nDarray tool on various levels.

- At the highest level of abstraction, one can use tensor, matrix and vector objects without knowing anything about the implementation and the inner workings. They are all designed and implemented as concrete data types. In spite of the very powerful code that can be built using Object Oriented technology, it would be unwise to expect proficiency in Object Oriented techniques and the C++ programming language from end users. It was our aim to provide power programming with multidimensional data types to users with basic knowledge of C.
- At a lower abstraction level, users can address the task of the actual implementation of operators and functions for vector, matrix and tensor classes. A number of improvements can be made, especially in optimizing some of the operators.
- The lowest level of abstraction is associated with nDarray and nDarray_rep classes. Arithmetic operators[1] are implemented at this level.

---

[1] Like addition and subtraction.

Next, classes are described from the base and down the inheritance tree. Later we focus our attention on nDarray usage examples. Our goal is to provide a useful programming tool, rather than to teach OOP or to show C++ implementation. For readers interested in actual implementation details, source code, examples and makefiles are available at `http://sokocalo.engr.ucdavis.edu/~jeremic`.

702.3.2.1 nDarray_rep class

The nDarray_rep class is a data holder and represents an n dimensional array object. A simple memory manager, implemented with the *reference counting idiom* (Coplien, 1992) is used. The memory manager uses rather inefficient built–in C memory allocation functions. Performance can be improved if one designs and implements specially tailored allocation functions for fast heap manipulations. Another possible improvement is in using memory resources other than heap memory. Sophisticated memory management introduced by the reference counting is best explained by Coplien (1992). The nDarray_rep class is not intended for stand–alone use. It is closely associated with the nDarray class.

The data structure of nDarray_rep introduces a minimal amount of information about a multidimensional array object. The actual data are stored as a one–dimensional array of double numbers. Rank, total number of elements, and array of dimensions are all that is needed to represent an multidimensional object. The data structure is allocated dynamically from the heap, and memory is reclaimed by the system after the object has gone out of scope.

702.3.2.2 nDarray class

The nDarray class together with the nDarray_rep class represents the abstract base for derived multidimensional data types: matrices, vectors and tensors. Objects derived from the nDarray class are generated dynamically by constructor functions at the first appearance of an object and are destroyed at the end of the block in which the object is referenced. The reference counting idiom provides for the object's life continuation after the end of the block where it was defined. To extend an object's life, a standard C++ compiler would by default call constructor functions, thus making the entire process of returning large objects from functions quite inefficient. By using reference counting idiom, destructor and constructor functions manipulate reference counter which results in a simple copying of a pointer to nDarray_rep object. By using this technique, copying of large objects is made very efficient.

Objects can be created from an array of values, or from a single scalar value, as shown in Table 702.1. Some of the frequently used multidimensional arrays are predefined and can be constructed by sending the proper flag to the constructor function. For example by sending the "`I`" flag one creates Kronecker delta δ_{ij} and by sending "e" flag, one creates a rank 3 Levi-Civita permutation tensor e_{ijk}. Functions and operators common to multidimensional data types are defined in the nDarray

constructor function	description
`nDarray(int rank_of_nD=1, double initval=0.0)`	default
`nDarray(int rank_of_nD, const int *pdim, double *val)`	from array
`nDarray(int rank_of_nD, const int *pdim, double initval)`	from scalar value
`nDarray(const char *flag, int rank_of_nD, const int *pdim)`	unit nDarrays
`nDarray(const nDarray & x)`	copy-initializer
`nDarray(int rank_of_nD, int rows, int cols, double *val)`	special for matrix
`nDarray(int rank_of_nD, int rows, int cols, double initval)`	special for matrix

Table 702.1: nDarray constructor functions.

class, as described in Table 702.2. These common operators and functions are inherited by derived classes. Occasionally, some of the functions will be redefined, overloaded in derived classes. In tensor multiplications we need additional information about indices. For example $C_{il} = (A_{ijk} + B_{ijk}) * D_{jkl} \overset{\text{coded}}{\longrightarrow}$ `C=(A("ijk")+B("ijk"))*D("jkl")`, the temporary in brackets will receive ijk indices, to be used for multiplication with D_{jkl}. It is interesting to note (Koenig, 1989 - 1993) that operator += is defined as a member and + is defined as an inline function in terms of += operator.

702.3.2.3 Matrix and Vector Classes

The matrix class is derived from the nDarray class through the public construct. It inherits common operators and functions from the base nDarray class, but it also adds its own set of functions and operators. Table (702.3) summarizes some of the more important additional functions and operators for the matrix class. The vector class defines vector objects and is derived and inherits most operators and data members from the matrix class. Some functions, like copy constructor, are overloaded in order to handle specifics of a vector object.

702.3.2.4 Tensor Class

The main goal of the tensor class development was to provide the implementing analyst with the ability to write the following equation directly into a computer program:

$$d\sigma_{mn} = -{}^{old}r_{ij} T^{-1}_{ijmn} - d\lambda\ E_{ijkl}\ {}^{n+1}m_{kl} T^{-1}_{ijmn}$$

as:

```
dsigma = -(r("ij")*Tinv("ijmn")) - dlambda*((E("ijkl")*dQods("kl"))*Tinv("ijmn"));
```

operator or function	left value	right value	description
`=`	nDarray	nDarray	nDarray assignment
`+`	nDarray	nDarray	nDarray addition
`+=`	nDarray	nDarray	nDarray addition
`-`		nDarray	unary minus
`-`	nDarray	nDarray	nDarray subtraction
`-=`	nDarray	nDarray	nDarray subtraction
`*`	double	nDarray	scalar multiplication (from left)
`*`	nDarray	double	scalar multiplication (from right)
`==`	nDarray	nDarray	nDarray comparison
`val(...)`	nDarray		reference to members of nDarray
`cval(...)`	nDarray		members of nDarray
`trace()`	nDarray		trace of square nDarray
`eigenvalues()`	nDarray		eigenvalues of rank 2 square nDarray
`eigenvectors()`	nDarray		eigenvectors of rank 2 square nDarray
`General_norm()`	nDarray		general p-th norm of nDarray
`nDsqrt()`	nDarray		square root of nDarray
`print(...)`	nDarray		generic print function

Table 702.2: Public functions and operators for nDarray class.

operator or function	left value	right value	description
`=`	matrix	matrix	matrix assignment
`*`	matrix	matrix	matrix multiplication
`transpose()`	matrix		matrix transposition
`determinant()`	matrix		determinant of a matrix
`inverse()`	matrix		matrix inversion

Table 702.3: Matrix class functions and operators (added on nDarray class definitions).

Instead of developing theory in terms of indicial notation, then converting everything to matrix notation and then implementing it, we were able to copy formulae directly from their indicial form to the C++ source code.

In addition to the definitions in the base nDarray class, the tensor class adds some specific functions and operators. Table 702.4 summarizes some of the main new functions and operators. The most significant addition is the tensor multiplication operator. With the help of a simple indicial parser, the multiplication operator contracts or expands indices and yields a resulting tensor of the correct rank. The resulting tensor receives proper indices, and can be used in further calculations on the same code statement.

operator or function	left value	right value	description
+	tensor	tensor	tensor addition
-	tensor	tensor	tensor subtraction
*	tensor	tensor	tensor multiplication
`transpose0110()`	tensor		$A_{ijkl} \rightarrow A_{ikjl}$
`transpose0101()`	tensor		$A_{ijkl} \rightarrow A_{ilkj}$
`transpose0111()`	tensor		$A_{ijkl} \rightarrow A_{iljk}$
`transpose1100()`	tensor		$A_{ijkl} \rightarrow A_{jikl}$
`transpose0011()`	tensor		$A_{ijkl} \rightarrow A_{ijlk}$
`transpose1001()`	tensor		$A_{ijkl} \rightarrow A_{ljki}$
`transpose11()`	tensor		$a_{ij} \rightarrow a_{ji}$
`symmetrize11()`	tensor		symmetrize second order tensor
`determinant()`	tensor		determinant of 2nd order tensor
`inverse()`	tensor		tensor inversion (2nd, 4th order)

Table 702.4: Additional and overloaded functions and operators for tensor class.

702.4 Finite Element Classes

702.4.1 Stress, Strain and Elastoplastic State Classes

The next step in our development was to use the nDarray tool classes for constitutive level computations. The simple extension was design and implementation of infinitesimal stress and strain tensor classes, namely stresstensor and straintensor. Both classes are quite similar, they inherit all the functions from the tensor class and we add some tools that are specific to them. Both stress and strain tensors are implemented as full second order 3×3 tensors. Symmetry of stress and strain tensor was not used to save storage space. Table 702.5 summarizes some of the main functions added on for the stresstensor class.

operator or function	description
`Iinvariant1()`	first stress invariant I1
`Iinvariant2()`	second stress invariant I2
`Iinvariant3()`	third stress invariant I3
`Jinvariant2()`	second deviatoric stress invariant J2
`Jinvariant3()`	third deviatoric stress invariant J3
`deviator()`	stress deviator
`principal()`	principal stresses on diagonal
`sigma_octahedral()`	octahedral mean stress
`tau_octahedral()`	octahedral shear stress
`xi()`	Haigh–Westergard coordinate ξ
`rho()`	Haigh–Westergard coordinate ρ
`p_hydrostatic()`	hydrostatic stress invariant
`q_deviatoric()`	deviatoric stress invariant
`theta()`	θ stress invariant (Lode's angle)

Table 702.5: Additional methods for stress tensor class.

Further on, we defined an elastoplastic state, which according to incremental theory of elastoplasticity with internal variables, is completely defined with the stress tensor and a set of internal variables. This definition led us to define an elastoplastic state termed class ep_state. Objects of type ep_state contain a stress tensor and a set of scalar or tensorial internal variables[2].

702.4.2 Material Model Classes

With all the previous developments, the design and implementation of various elastoplastic material models was not a difficult task. A generic class Material_Model defines techniques that form a framework for small deformation elastoplastic computations. Table 702.6 summarizes some of the main methods defined for the Material_Model class in terms of yield (F) and potential (Q) functions.

---

[2] Internal variables can be characterized as tensors of even order, where, for example, zero tensor is a scalar internal variable associated with isotropic hardening and second order tensors can be associated with kinematic hardening.

operator or function	description
`F`	F Yield function value
`dFods`	$\partial F / \partial \sigma_{ij}$
`dQods`	$\partial Q / \partial \sigma_{ij}$
`d2Qods2`	$\partial^2 Q / \partial \sigma_{ij} \partial \sigma_{kl}$
`dpoverds`	$\partial p / \partial \sigma_{ij}$
`dqoverds`	$\partial q / \partial \sigma_{ij}$
`dthetaoverds`	$\partial \theta / \partial \sigma_{ij}$
`d2poverds2`	$\partial^2 p / \partial \sigma_{ij} \partial \sigma_{kl}$
`d2qoverds2`	$\partial^2 q / \partial \sigma_{ij} \partial \sigma_{kl}$
`d2thetaoverds2`	$\partial^2 \theta / \partial \sigma_{ij} \partial \sigma_{kl}$
`ForwardPredictorEPState`	Explicit predictor elastoplastic state
`BackwardEulerEPState`	Implicit return elastoplastic state
`ForwardEulerEPState`	Explicit return elastoplastic state
`BackwardEulerCTensor`	Algorithmic tangent stiffness tensor
`ForwardEulerCTensor`	Continuum tangent stiffness tensor

Table 702.6: Some of the methods in material model class.

It is important to note that all the material model dependent functions are defined as virtual functions. Integration algorithms are designed and implemented using template algorithms, and each implemented material model appends its own yield and potential functions and appropriate derivatives. Implementation of additional material models requires coding of yield and potential functions and respective derivative functions.

702.4.3 Stiffness Matrix Class

Starting from the incremental equilibrium of the stationary body, the principle of virtual displacements and with the finite element approximation of the displacement field $u \approx \hat{u}_a = H_I \bar{u}_{Ia}$, the weak form of equilibrium can be expressed as (Zienkiewicz and Taylor, 1991a)

$$\bigcup_m \int_{V^m} H_{I,b}\, E_{abcd}\, H_{J,d}\, dV^m\, \bar{u}_{Jc} = \bigcup_m \int_{V^m} f_a\, H_I\, dV^m \quad \text{or} \quad (f_{Ia}(\bar{u}_{Jc}))_{int} = \lambda\, (f_{Ia})_{ext}$$

where E_{abcd} is the constitutive tangent stiffness tensor[3]. The element stiffness tensor is recognized as

$$k^e_{aIcJ} = \int_{V^m} H_{I,b} \; {}^{tan}E_{abcd} \; H_{J,d} \; dV^m$$

This generic form for the finite element stiffness tensor is easily programmed with the help of the nDarray tool. A simple implementation example is provided later. It should be noted that the element stiffness tensor in this case is a four–dimensional tensor. It is the task of the assembly function to collect proper terms for addition in a global stiffness matrix.

702.5 Examples

702.5.1 Tensor Examples

Some of the basic tensorial calculations with tensors are presented. Tensors have a default constructor that creates a first order tensor with one element initialized to 0.0:

```
tensor t1;
```

Tensors can be constructed and initialized from a given set of numbers:

```
static double t2values[] = {  1,2,3,
                              4,5,6,
                              7,8,9  };
tensor t2( 2, DefDim2, t2values); // order 2; 3x3 tensor (like matrix)
```

Here, `DefDim2`, `DefDim3` and `DefDim4` are arrays of dimensions for the second, third and fourth order tensor[4]. A fourth order tensor with 0.0 value assignment and dimension 3 in each order $(3 \times 3 \times 3 \times 3)$ is constructed in the following way:

```
tensor ZERO(4,DefDim4,0.0);
```

Tensors can be multiplied using indicial notation. The following example will do a tensorial multiplication of previously defined tensors `t2` and `t4` so that $tst1 = t2_{ij} t4_{ijkl} t4_{klpq} t2_{pq}$. Note that the memory is dynamically allocated to accept the proper tensor dimensions that will result from the multiplication[5]

```
tensor tst1 = t2("ij")*t4("ijkl")*t4("klpq")*t2("pq");
```

Inversion of tensors is possible. It is defined for 2 and 4 order tensors only. The fourth order tensor inversion is done by converting it to matrix, inverting that matrix and finally converting matrix back to tensor.

```
tensor t4inv_2 = t4.inverse();
```

---

[3] Which may be continuum or algorithmic (Jeremić and Sture, 1997) tangent stiffness tensor

[4] In this case dimensions are 3 in every order.

[5] In this case it will be zero dimensional tensor with one element.

There are two built–in tensor types, *Levi-Civita permutation tensor* e_{ijk} and *Kronecker delta tensor* δ_{ij}

```
tensor e("e",3,DefDim3);     // Levi-Civita permutation tensor
tensor I2("I", 2, DefDim2);  // Kronecker delta tensor
```

Trace and determinant functions for tensors are used as follows

```
double deltatrace = I2.trace();
double deltadet = I2.determinant();
```

Tensors can be compared to within a square root of *machine epsilon*[6] tolerance

```
tensor I2again = I2;
if ( I2again == I2 )
  printf("I2again == I2  TRUE (OK)");
else
  printf("I2again == I2  NOTTRUE");
```

702.5.2 Fourth Order Isotropic Tensors

Some of the fourth order tensors used in continuum mechanics are built quite readily. The most general representation of the fourth order isotropic tensor includes the following fourth order unit isotropic tensors[7]

```
tensor I_ijkl = I2("ij")*I2("kl");
```

The resulting tensor `I_ijkl` will have the correct indices, $I\_ijkl_{ijkl} = I2_{ij} I2_{kl}$. Note that `I_ijkl` is just a name for the tensor, and the `_ijkl` part reminds the implementor what that tensor is representing.

The real indices, $*_{ijkl}$ in this case, are stored in the tensor object, and can be used further or changed appropriately. The next tensor that is needed is a fourth order unit tensor obtained by transposing the previous one in the minor indices,

```
tensor I_ikjl = I_ijkl.transpose0110();
```

while the third tensor needed for representation of general isotropic tensor is constructed by using similar transpose function

```
tensor I_iljk = I_ijkl.transpose0111();
```

The inversion function can be checked for fourth order tensors:

---

[6] Machine epsilon ($macheps$) is defined as the smallest distinguishable positive number (in a given precision, i.e. float (32 bits), double (64 bits) or long double (80 bits), such that $1.0 + macheps > 1.0$ yields true on the given computer platform. For example, double precision arithmetics (64 bits), on the `Intel 80x86` platform yields `macheps=1.08E-19` while on the `SUN SPARCstation` and DEC platforms `macheps=2.22E-16`.

[7] Remember that `I2` was constructed as the Kronecker delta tensor δ_{ij}.

```
tensor I_ikjl_inv_2 = I_ikjl.inverse();
if ( I_ikjl == I_ikjl_inv_2 )
  printf(" I_ikjl == I_ikjl_inv_2 (OK) !");
else
  printf(" I_ikjl != I_ikjl_inv_2 !");
```

Creating a symmetric and skew symmetric unit fourth order tensors gets to be quite simple by using tensor addition and scalar multiplication

```
tensor I4s  = (1./2.)*(I_ikjl+I_iljk);
tensor I4sk = (1./2.)*(I_ikjl-I_iljk);
```

Another interesting example is a numerical check of the $e - \delta$ identity (Lubliner, 1990) ($e_{ijm}e_{klm} = \delta_{ik}\delta_{jl} - \delta_{il}\delta_{jk}$)

```
tensor id = e("ijm")*e("klm") - (I_ikjl - I_iljk);
if ( id == ZERO )
  printf(" e-delta identity HOLDS !! ");
```

702.5.3 Elastic Isotropic Stiffness and Compliance Tensors

The linear isotropic elasticity tensor E_{ijkl} can be built from Young's modulus E and Poisson's ratio ν

```
double Ey = 20000; // Young's modulus of elasticity
double nu = 0.2;   // Poisson's Ratio
tensor E = ((2.*Ey*nu)/(2.*(1.+nu)*(1-2.*nu)))*I_ijkl + (Ey/(1.+nu))*I4s;
```

Similarly, the compliance tensor is

```
tensor D = (-nu/Ey)*I_ijkl + ((1.0+nu)/Ey)*I4s;
```

One can multiply the two and check if the result is equal to the symmetric fourth order unit tensor

```
tensor test = E("ijkl")*D("klpq");
if ( test == I4s )
  printf(" test == I4s  TRUE  (OK up to sqrt(macheps)) ");
else
  printf(" test == I4s  NOTTRUE   ");
```

The linear isotropic elasticity and compliance tensors can be obtained in a different way, by using Lamé constants λ and μ

```
double lambda = nu * Ey / (1. + nu) / (1. - 2. * nu);
double mu = Ey / (2. * (1. + nu));
tensor E = lambda*I_ijkl + (2.*mu)*I4s;         // stiffness tensor
tensor D = (-nu/Ey)*I_ijkl  + (1./(2.*mu))*I4s; // compliance tensor
```

702.5.4 Second Derivative of θ Stress Invariant

As an extended example of nDarray tool usage, the implementation for the second derivative of the stress invariant θ (Lode angle) is presented. The derivative is used for implicit constitutive integration schemes applied to three invariant material models. The original equation reads:

$$
\begin{aligned}
\frac{\partial^2\theta}{\partial\sigma_{pq}\partial\sigma_{mn}} &= \\
&- \left(\frac{9}{2}\frac{\cos 3\theta}{q^4\sin(3\theta)} + \frac{27}{4}\frac{\cos 3\theta}{q^4\sin^3 3\theta}\right) s_{pq}\, s_{mn} + \frac{81}{4}\frac{1}{q^5\sin^3 3\theta}\, s_{pq}\, t_{mn} + \\
&+ \left(\frac{81}{4}\frac{1}{q^5\sin 3\theta} + \frac{81}{4}\frac{\cos^2 3\theta}{q^5\sin^3 3\theta}\right) t_{pq}\, s_{mn} - \frac{243}{4}\frac{\cos 3\theta}{q^6\sin^3 3\theta}\, t_{pq}\, t_{mn} + \\
&+ \frac{3}{2}\frac{\cos(3\theta)}{q^2\sin(3\theta)}\, p_{pqmn} - \frac{9}{2}\frac{1}{q^3\sin(3\theta)}\, w_{pqmn}
\end{aligned}
$$

where:

$$
q = \sqrt{\frac{3}{2}s_{ij}s_{ij}} \quad ; \quad \cos 3\theta = \frac{3\sqrt{3}}{2}\frac{\frac{1}{3}s_{ij}s_{jk}s_{ki}}{\sqrt{(\frac{1}{2}s_{ij}s_{ij})^3}} \quad ; \quad s_{ij} = \sigma_{ij} - \frac{1}{3}\sigma_{kk}\delta_{ij}
$$

$$
w_{pqmn} = s_{np}\delta_{qm} + s_{qm}\delta_{np} - \frac{2}{3}s_{qp}\delta_{nm} - \frac{2}{3}\delta_{pq}s_{mn} \quad ; \quad p_{pqmn} = \delta_{mp}\delta_{nq} - \frac{1}{3}\delta_{pq}\delta_{mn}
$$

and the implementation follows:

```
tensor Yield_Criteria::d2thetaoverds2(stresstensor & stress)
 {
  tensor ret( 4, DefDim4, 0.0);
  tensor I2("I", 2, DefDim2);
  tensor I_pqmn = I2("pq")*I2("mn");
  tensor I_pmqn = I_pqmn.transpose0110();
  double J2D = stress.Jinvariant2();
  tensor s = stress.deviator();
  tensor t = s("qk")*s("kp") - I2*(J2D*(2.0/3.0));
  double theta = stress.theta();
  double q_dev = stress.q_deviatoric();
//setting up some constants
  double c3t    = cos(3*theta);
  double s3t    = sin(3*theta);
  double s3t3   = s3t*s3t*s3t;
  double q3     = q_dev * q_dev * q_dev;
  double q4     = q3 * q_dev;
  double q5     = q4 * q_dev;
```

```
  double q6     = q5 * q_dev;
  double tempss = -(9.0/2.0)*(c3t)/(q4*s3t)-(27.0/4.0)*(c3t/(s3t3*q4));
  double tempst = +(81.0/4.0)*(1.0)/(s3t3*q5);
  double tempts = +(81.0/4.0)*(1.0/(s3t*q5))+(81.0/4.0)*(c3t*c3t)/(s3t3*q5);
  double temptt = -(243.0/4.0)*(c3t/(s3t3*q6));
  double tempp  = +(3.0/2.0)*(c3t/(s3t*q_dev*q_dev));
  double tempw  = -(9.0/2.0)*(1.0/(s3t*q3));
  tensor s_pq_d_mn = s("pq")*I2("mn");
  tensor s_pn_d_mq = s_pq_d_mn.transpose0101();
  tensor d_pq_s_mn = I2("pq")*s("mn");
  tensor d_pn_s_mq = d_pq_s_mn.transpose0101();
  tensor p = I_pmqn  - I_pqmn*(1.0/3.0);
  tensor w = s_pn_d_mq+d_pn_s_mq - s_pq_d_mn*(2.0/3.0)-d_pq_s_mn*(2.0/3.0);
// finally
  ret = (s("pq")*s("mn")*tempss + s("pq")*t("mn")*tempst +
         t("pq")*s("mn")*tempts + t("pq")*t("mn")*temptt +
         p*tempp + w*tempw );
   return ret;
 }
```

702.5.5 Application to Computations in Elastoplasticity

A useful application of the previously described classes is for elastoplastic computations. If the Newton iterative scheme is used at the global equilibrium level, then in order to preserve a quadratic rate, a consistent, algorithmic tangent stiffness (ATS) tensor should be used. For a general class of three–invariant, non–associated, hardening or softening material models, ATS is defined (Jeremić and Sture, 1997) as:

$$
{}^{cons}E^{ep}_{pqmn} = R_{pqmn} - \frac{R_{pqkl}\,{}^{n+1}H_{kl}\,{}^{n+1}n_{ij}R_{ijmn}}{{}^{n+1}n_{ot}R_{otpq}\;{}^{n+1}H_{pq} + {}^{n+1}\xi_*\;h_*}
$$

where

$$
m_{kl} = \frac{\partial Q}{\partial \sigma_{kl}} \quad ; \quad n_{kl} = \frac{\partial F}{\partial \sigma_{kl}} \quad ; \quad \xi_* = \frac{\partial F}{\partial q_*} \quad ; \quad T_{ijmn} = \delta_{im}\delta_{nj} + \Delta\lambda\; E_{ijkl}\,\frac{\partial m_{kl}}{\partial \sigma_{mn}}
$$

$$
H_{kl} = {}^{n+1}m_{kl} + \Delta\lambda \frac{\partial m_{kl}}{\partial q_*} h_* \quad ; \quad R_{mnkl} = \left({}^{n+1}T_{ijmn}\right)^{-1} E_{ijkl}
$$

A straightforward implementation of the above tensorial formula follows:

```
    double Ey = Criterion.E();
    double nu = Criterion.nu();
    tensor Eel = StiffnessTensorE(Ey,nu);
```

```
    tensor I2("I", 2, DefDim2);
    tensor I_ijkl = I2("ij")*I2("kl");
    tensor I_ikjl = I_ijkl.transpose0110();
    tensor m = Criterion.dQods(final_stress);
    tensor n = Criterion.dFods(final_stress);
    double lambda = current_lambda_get();
    tensor d2Qoverds2 = Criterion.d2Qods2(final_stress);
    tensor T = I_ikjl + Eel("ijkl")*d2Qoverds2("klmn")*lambda;
    tensor Tinv = T.inverse();
    tensor R = Tinv("ijmn")*Eel("ijkl");
    double h_  =  h(final_stress);
    double xi_ = xi(final_stress);
    double hardMod_ = h_ * xi_;
    tensor d2Qodqast2 = d2Qoverdqast2(final_stress);
    tensor H = m + d2Qodqast2 * lambda * h_;
//
    tensor upper = R("pqkl")*H("kl")*n("ij")*R("ijmn");
    double lower = (n("ot")*R("otpq"))*H("pq")).trace();
    lower = lower + hardMod_;
    tensor Ep = upper*(1./lower);
    tensor Eep =  R - Ep; // elastoplastic ATS constitutive tensor
```

This ATS tensor can be used further in building finite element stiffness tensors, as will be shown in our next example.

702.5.6 Stiffness Matrix Example

By applying a numerical integration technique to the stiffness matrix equation

$$k^e_{aIcJ} = \int_{V^m} H_{I,b}\, E_{abcd}\, H_{J,d}\, dV^m$$

individual contributions are summed into the element stiffness tensor. This process can be implemented on a integration point level by using the nDarray tool as

```
  K  =  K  +  H("Ib") * E("abcd") * H("Jd") * weight ;
```

It is interesting to note the lack of loops at this level of implementation. However, there exists a loop over integration points which contributes stiffness to the element tensor.

702.6 Performance Issues

In the course of developing the nDarray tool, execution speed was not a priority or issue that we tried to perfect. The benefit of being able to implement and test various numerical algorithms in a straightforward manner was the main concern. The efficiency of the nDarray tool when compared with FORTRAN or C was never assessed. In all honesty, some of the formulae implemented in C++ with the help of the nDarray tool would be difficult to implement in FORTRAN or C. The entire question of efficiency of the nDarray as compared to FORTRAN or C codes might thus remain unanswered for the time being.

The efficiency of C++ for numerical computations has been under consideration (Robison, 1996) for some time now. Poor efficiency and possible remedies for improving efficiency of C++ computations has been reported in literature (Robison, 1996) (Veldhuizen, 1995b) (Veldhuizen, 1996). Novel techniques, such as *Template Expressions* (Veldhuizen, 1995b) can be used to achieve and sometimes surpass the performance of hand–tuned FORTRAN or C codes.

702.7 Summary and Future Directions

A novel programming tool, named nDarray, has been presented which facilitates implementation of tensorial formulae. It was shown how OOP and efficient programming in C++ allows building of new concrete data types, in this case tensors of any order. In a number of examples these new data types were shown to be useful in implementing tensorial formulae associated with the numerical solution of various elastic and elastoplastic problems with the finite element method. The nDarray tool is been used in developing of the FEMtools tools library. The FEMtools tools library includes a set of finite elements, various solvers, solution procedures for non–linear finite element system of equations and other useful functions.

Appendix 703

Closed Form Gradients to the Plastic Potential Function

(1993-1994-)

703.1 Chapter Summary and Highlights

A complete derivation of gradients to the Potential and Yield function follows. The yield function F and potential function Q are functions of the stress tensor σ_{ij} and plastic internal variable tensor q_*. Only derivatives with respect to the stress tensor σ_{ij} are given here. It is assumed that any stress state can be represented with three stress invariants p, q and θ given in the following form:

$$p = -\frac{1}{3} I_1 \quad q = \sqrt{3 J_{2D}} \quad \cos 3\theta = \frac{3\sqrt{3}}{2} \frac{J_{3D}}{\sqrt{(J_{2D})^3}} \tag{703.1}$$

$$I_1 = \sigma_{kk} \quad J_{2D} = \frac{1}{2} s_{ij} s_{ij} \quad J_{3D} = \frac{1}{3} s_{ij} s_{jk} s_{ki} \quad s_{ij} = \sigma_{ij} - \frac{1}{3} \sigma_{kk} \delta_{ij} \tag{703.2}$$

and stresses are chosen as positive in tension. One can write the Potential Function in the following form:

$$Q = Q(p, q, \theta) \tag{703.3}$$

and the derivation follows. Hopefully the pace of derivation is rather slow, thus little explanation will be given until the end of the derivation. Chain rule of differentiation yields:

$$\frac{\partial Q}{\partial \sigma_{ij}} = \frac{\partial Q}{\partial p} \frac{\partial p}{\partial \sigma_{ij}} + \frac{\partial Q}{\partial q} \frac{\partial q}{\partial \sigma_{ij}} + \frac{\partial Q}{\partial \theta} \frac{\partial \theta}{\partial \sigma_{ij}} \tag{703.4}$$

and the intermediate derivatives are:

$$\frac{\partial p}{\partial \sigma_{ij}} = \frac{\partial(-\frac{1}{3}\sigma_{kk})}{\partial \sigma_{ij}} = -\frac{1}{3}\,\delta_{ij} \tag{703.5}$$

$$\frac{\partial q}{\partial \sigma_{ij}} = \frac{\partial \sqrt{3J_{2D}}}{\partial \sigma_{ij}} = \frac{\sqrt{3}}{2}\frac{1}{\sqrt{J_{2D}}}\frac{\partial J_{2D}}{\partial \sigma_{ij}} = \frac{\sqrt{3}}{2}\frac{1}{\sqrt{J_{2D}}}\,s_{ij} = \frac{3}{2}\frac{1}{q}s_{ij} \tag{703.6}$$

$$\frac{\partial \theta}{\partial \sigma_{ij}} = \tag{703.7}$$

$$= \quad \frac{1}{3}\frac{-1}{\sqrt{1-(\frac{3\sqrt{3}}{2}\frac{J_{3D}}{J_{2D}^{3/2}})^2}}\frac{3\sqrt{3}}{2}\left(\frac{\partial J_{3D}}{\partial \sigma_{ij}}\frac{1}{\sqrt{(J_{2D})^3}} - \frac{3}{2}J_{3D}\frac{\partial J_{2D}}{\partial \sigma_{ij}}\frac{1}{\sqrt{(J_{2D})^5}}\right) =$$

$$= \quad \frac{1}{3\sqrt{1-\left(\frac{3\sqrt{3}}{2}\frac{J_{3D}}{J_{2D}^{3/2}}\right)^2}}\frac{3\sqrt{3}}{2}\left(-t_{ij}\frac{1}{\sqrt{(J_{2D})^3}} + \frac{3}{2}J_{3D}s_{ij}\frac{1}{\sqrt{(J_{2D})^5}}\right) =$$

$$= \quad \frac{1}{\sin 3\theta}\frac{\sqrt{3}}{2}\left(\frac{3}{2}J_{3D}\frac{1}{\sqrt{(J_{2D})^5}}\;s_{ij} - \frac{1}{\sqrt{(J_{2D})^3}}\;t_{ij}\right) =$$

$$= \quad \frac{\sqrt{3}}{2}\frac{1}{\sin(3\theta)}\left(\frac{\sqrt{3}\cos(3\theta)}{q^2}\;s_{ij} - \frac{3\sqrt{3}}{q^3}\;t_{ij}\right) =$$

$$= \quad \frac{3}{2}\frac{\cos(3\theta)}{q^2\sin(3\theta)}s_{ij} - \frac{9}{2}\frac{1}{q^3\sin(3\theta)}t_{ij} \tag{703.8}$$

Second derivatives of the potential function Q using again the chain rule of differentiation are as follows:

$$
\begin{aligned}
&\frac{\partial^2 Q}{\partial \sigma_{pq} \partial \sigma_{mn}} = \frac{\partial \left(\frac{\partial Q}{\partial \sigma_{pq}} \right)}{\partial \sigma_{mn}} = \\
&\frac{\partial \left(\frac{\partial Q}{\partial p} \frac{\partial p}{\partial \sigma_{pq}} + \frac{\partial Q}{\partial q} \frac{\partial q}{\partial \sigma_{pq}} + \frac{\partial Q}{\partial \theta} \frac{\partial \theta}{\partial \sigma_{pq}} \right)}{\partial \sigma_{mn}} = \\
&\frac{\partial \left(\frac{\partial Q}{\partial p} \right)}{\partial \sigma_{mn}} \frac{\partial p}{\partial \sigma_{pq}} + \frac{\partial Q}{\partial p} \frac{\partial^2 p}{\partial \sigma_{pq} \partial \sigma_{mn}} + \\
+ \quad &\frac{\partial \left(\frac{\partial Q}{\partial q} \right)}{\partial \sigma_{mn}} \frac{\partial q}{\partial \sigma_{pq}} + \frac{\partial Q}{\partial q} \frac{\partial^2 q}{\partial \sigma_{pq} \partial \sigma_{mn}} + \\
+ \quad &\frac{\partial \left(\frac{\partial Q}{\partial \theta} \right)}{\partial \sigma_{mn}} \frac{\partial \theta}{\partial \sigma_{pq}} + \frac{\partial Q}{\partial \theta} \frac{\partial^2 \theta}{\partial \sigma_{pq} \partial \sigma_{mn}} = \\
&\left(\frac{\partial^2 Q}{\partial p^2} \frac{\partial p}{\partial \sigma_{mn}} + \frac{\partial^2 Q}{\partial p \partial q} \frac{\partial q}{\partial \sigma_{mn}} + \frac{\partial^2 Q}{\partial p \partial \theta} \frac{\partial \theta}{\partial \sigma_{mn}} \right) \frac{\partial p}{\partial \sigma_{pq}} + \frac{\partial Q}{\partial p} \frac{\partial^2 p}{\partial \sigma_{pq} \partial \sigma_{mn}} + \\
+ \quad &\left(\frac{\partial^2 Q}{\partial q \partial p} \frac{\partial p}{\partial \sigma_{mn}} + \frac{\partial^2 Q}{\partial q^2} \frac{\partial q}{\partial \sigma_{mn}} + \frac{\partial^2 Q}{\partial q \partial \theta} \frac{\partial \theta}{\partial \sigma_{mn}} \right) \frac{\partial q}{\partial \sigma_{pq}} + \frac{\partial Q}{\partial q} \frac{\partial^2 q}{\partial \sigma_{pq} \partial \sigma_{mn}} + \\
+ \quad &\left(\frac{\partial^2 Q}{\partial \theta \partial p} \frac{\partial p}{\partial \sigma_{mn}} + \frac{\partial^2 Q}{\partial \theta \partial q} \frac{\partial q}{\partial \sigma_{mn}} + \frac{\partial^2 Q}{\partial \theta^2} \frac{\partial \theta}{\partial \sigma_{mn}} \right) \frac{\partial \theta}{\partial \sigma_{pq}} + \frac{\partial Q}{\partial \theta} \frac{\partial^2 \theta}{\partial \sigma_{pq} \partial \sigma_{mn}}
\end{aligned}
$$

and the intermediate derivatives are as follows:

$$
\frac{\partial^2 p}{\partial \sigma_{pq} \partial \sigma_{mn}} = \frac{\partial^2 \left(-\frac{1}{3} \sigma_{kk} \right)}{\partial \sigma_{pq} \partial \sigma_{mn}} = \frac{\partial \left(-\frac{1}{3} \delta_{kp} \delta_{qk} \right)}{\partial \sigma_{mn}} = \emptyset
$$

$$
\begin{aligned}
\frac{\partial^2 q}{\partial \sigma_{pq} \partial \sigma_{mn}} &= \frac{\partial \left(\frac{\partial q}{\partial \sigma_{pq}} \right)}{\partial \sigma_{mn}} = \\
\frac{\partial \left(\frac{\sqrt{3}}{2} \frac{1}{\sqrt{J_{2D}}} s_{pq} \right)}{\partial \sigma_{mn}} = \frac{\sqrt{3}}{2} \frac{1}{\sqrt{J_{2D}}} \frac{\partial s_{pq}}{\partial \sigma_{mn}} + \frac{\sqrt{3}}{2} \frac{\partial \frac{1}{\sqrt{J_{2D}}}}{\partial \sigma_{mn}} s_{pq} &= \\
\frac{\sqrt{3}}{2} \frac{1}{\sqrt{J_{2D}}} \left(\delta_{pm} \delta_{nq} - \frac{1}{3} \delta_{pq} \delta_{km} \delta_{nk} \right) + \frac{\sqrt{3}}{2} \left(\frac{-1}{2} \left(\frac{1}{\left(\sqrt{J_{2D}} \right)^3} \right) s_{mn} \right) s_{pq} &= \\
\frac{\sqrt{3}}{2} \frac{1}{\sqrt{J_{2D}}} \left(\delta_{pm} \delta_{nq} - \frac{1}{3} \delta_{pq} \delta_{nm} \right) - \frac{\sqrt{3}}{4} \left(\frac{1}{\sqrt{J_{2D}}} \right)^3 s_{mn} s_{pq} &= \\
\frac{3}{2} \frac{1}{q} \left(\delta_{pm} \delta_{nq} - \frac{1}{3} \delta_{pq} \delta_{nm} \right) - \frac{9}{4} \frac{1}{q^3} \; s_{mn} s_{pq}
\end{aligned}
$$

Let us introduce a slightly different form for the equation $\frac{\partial^2 \theta}{\partial \sigma_{pq} \partial \sigma_{mn}}$ in order to simplify writing:

$$\frac{\partial \theta}{\partial \sigma_{pq}} = \frac{3}{2}\frac{\cos(3\theta)}{q^2 \sin(3\theta)} s_{pq} - \frac{9}{2}\frac{1}{q^3 \sin(3\theta)} t_{pq} =$$

$$= AS\ s_{pq} + AT\ t_{pq}$$

where:

$$AS = \frac{3}{2}\frac{\cos(3\theta)}{q^2 \sin(3\theta)}$$

$$AT = -\frac{9}{2}\frac{1}{q^3 \sin(3\theta)}$$

Now the problem will be separated in two smaller problems, namely:

$$\frac{\partial^2 \theta}{\partial \sigma_{pq} \partial \sigma_{mn}} = \frac{\partial \frac{\partial \theta}{\partial \sigma_{pq}}}{\partial \sigma_{mn}} =$$

$$\frac{\partial \left(\frac{3}{2}\frac{\cos(3\theta)}{q^2 \sin(3\theta)} s_{pq} - \frac{9}{2}\frac{1}{q^3 \sin(3\theta)} t_{pq} \right)}{\partial \sigma_{mn}} =$$

$$\frac{\partial \left(AS\ s_{pq} + AT\ t_{pq} \right)}{\partial \sigma_{mn}} =$$

$$\frac{\partial \left(AS\ s_{pq} \right)}{\partial \sigma_{mn}} + \frac{\partial \left(AT\ t_{pq} \right)}{\partial \sigma_{mn}}$$

Now let us take a look at $\frac{\partial (AS\ s_{pq})}{\partial \sigma_{mn}}$. Since:

$$
\begin{aligned}
&\frac{\partial \left(AS \; s_{pq} \right)}{\partial \sigma_{mn}} = \\
&\frac{\partial AS}{\partial \sigma_{mn}} \; s_{pq} + AS \; \frac{\partial s_{pq}}{\partial \sigma_{mn}} = \\
&\left(\frac{\partial AS}{\partial q} \frac{\partial q}{\partial \sigma_{mn}} + \frac{\partial AS}{\partial \theta} \frac{\partial \theta}{\partial \sigma_{mn}} \right) \; s_{pq} + AS \; \frac{\partial s_{pq}}{\partial \sigma_{mn}} = \\
&\Bigg(\frac{-3. \; \cot(3\,\theta)}{q^3} \; \frac{3}{2} \frac{1}{q} \; s_{mn} \; + \\
&+ \frac{-4.5 \csc(3\,\theta)^2}{q^2} \left(\frac{3}{2} \frac{\cos{(3\theta)}}{q^2 \sin{(3\theta)}} s_{mn} - \frac{9}{2} \frac{1}{q^3 \sin{(3\theta)}} t_{mn} \right) \Bigg) \; s_{pq} \; + \\
&\frac{3}{2} \frac{\cos{(3\theta)}}{q^2 \sin{(3\theta)}} \; p_{pqmn} = \\
&-\frac{9}{2} \frac{\cos 3\theta}{q^4 \sin{(3\theta)}} \; s_{pq} \; s_{mn} - \frac{27}{4} \frac{\cos 3\theta}{q^4 \sin^3 3\theta} \; s_{pq} \; s_{mn} + \frac{81}{4} \frac{1}{q^5 \sin^3 3\theta} \; s_{pq} \; t_{mn} \; + \\
&\frac{3}{2} \frac{\cos{(3\theta)}}{q^2 \sin{(3\theta)}} \; p_{pqmn} = \\
&-\left(\frac{9}{2} \frac{\cos 3\theta}{q^4 \sin{(3\theta)}} + \frac{27}{4} \frac{\cos 3\theta}{q^4 \sin^3 3\theta} \right) \; s_{pq} \; s_{mn} + \frac{81}{4} \frac{1}{q^5 \sin^3 3\theta} \; s_{pq} \; t_{mn} \; + \\
&\frac{3}{2} \frac{\cos{(3\theta)}}{q^2 \sin{(3\theta)}} \; p_{pqmn}
\end{aligned}
$$

where:

$$
p_{pqmn} = \frac{\partial s_{pq}}{\partial \sigma_{mn}} = \left(\delta_{mp} \delta_{nq} - \frac{1}{3} \delta_{pq} \delta_{mn} \right)
$$

is the projection tensor and:

$$
\frac{\partial AS}{\partial q} = \frac{-3. \; \cot(3\,\theta)}{q^3}
$$

$$
\frac{\partial AS}{\partial \theta} = \frac{-4.5 \csc(3\,\theta)^2}{q^2}
$$

The second member is $\frac{\partial (AT \; t_{pq})}{\partial \sigma_{mn}}$:

$$
\begin{aligned}
&\frac{\partial\left(AT\ t_{pq}\right)}{\partial\sigma_{mn}} = \\
&\frac{\partial AT}{\partial\sigma_{mn}}\ t_{pq} + AT\ \frac{\partial t_{pq}}{\partial\sigma_{mn}} = \\
&\left(\frac{\partial AT}{\partial q}\frac{\partial q}{\partial\sigma_{mn}} + \frac{\partial AT}{\partial\theta}\frac{\partial\theta}{\partial\sigma_{mn}}\right)\ t_{pq} + AT\ \frac{\partial t_{pq}}{\partial\sigma_{mn}} =
\end{aligned}
$$

$$
\begin{aligned}
&\left(\frac{13.5\ \csc(3\,\theta)}{q^4}\frac{3}{2}\frac{1}{q}\ s_{mn}+\right.\\
&\left.+\ \frac{13.5\ \cot(3\,\theta)\ \csc(3\,\theta)}{q^3}\left(\frac{3}{2}\frac{\cos(3\theta)}{q^2\sin(3\theta)}s_{mn} - \frac{9}{2}\frac{1}{q^3\sin(3\theta)}t_{mn}\right)\right)\ t_{pq} + \\
&+ - \frac{9}{2}\frac{1}{q^3\sin(3\theta)}\ w_{pqmn} =
\end{aligned}
$$

$$
\begin{aligned}
&\frac{81}{4}\frac{1}{q^5\sin 3\theta}\ t_{pq}\ s_{mn} + \frac{81}{4}\frac{\cos^2 3\theta}{q^5\sin^3 3\theta}\ t_{pq}\ s_{mn} - \frac{243}{4}\frac{\cos 3\theta}{q^6\sin^3 3\theta}\ t_{pq}\ t_{mn} - \\
&-\frac{9}{2}\frac{1}{q^3\sin(3\theta)}\ w_{pqmn} =
\end{aligned}
$$

$$
\begin{aligned}
&\left(\frac{81}{4}\frac{1}{q^5\sin 3\theta} + \frac{81}{4}\frac{\cos^2 3\theta}{q^5\sin^3 3\theta}\right)\ t_{pq}\ s_{mn} - \frac{243}{4}\frac{\cos 3\theta}{q^6\sin^3 3\theta}\ t_{pq}\ t_{mn} - \\
&-\frac{9}{2}\frac{1}{q^3\sin(3\theta)}\ w_{pqmn}
\end{aligned}
$$

where:

$$w_{pqmn} = \frac{\partial t_{pq}}{\partial \sigma_{mn}} = s_{np}\delta_{qm} + s_{qm}\delta_{np} - \frac{2}{3} s_{qp}\delta_{nm} - \frac{2}{3}\delta_{pq}s_{mn}$$

$$\frac{\partial AT}{\partial q} = \frac{13.5\ \csc(3\,\theta)}{q^4}$$

$$\frac{\partial AT}{\partial \theta} = \frac{13.5\ \cot(3\,\theta)\ \csc(3\,\theta)}{q^3}$$

Then finally by collecting terms back again we have:

$$\begin{aligned}
\frac{\partial^2 \theta}{\partial \sigma_{pq} \partial \sigma_{mn}} &= \\
\frac{\partial \left(AS\ s_{pq}\right)}{\partial \sigma_{mn}} + \frac{\partial \left(AT\ t_{pq}\right)}{\partial \sigma_{mn}} &= \\
-\left(\frac{9}{2}\frac{\cos 3\theta}{q^4 \sin(3\theta)} + \frac{27}{4}\frac{\cos 3\theta}{q^4 \sin^3 3\theta}\right)\ s_{pq}\ s_{mn} + \frac{81}{4}\frac{1}{q^5 \sin^3 3\theta}\ s_{pq}\ t_{mn} &+ \\
+\left(\frac{81}{4}\frac{1}{q^5 \sin 3\theta} + \frac{81}{4}\frac{\cos^2 3\theta}{q^5 \sin^3 3\theta}\right)\ t_{pq}\ s_{mn} - \frac{243}{4}\frac{\cos 3\theta}{q^6 \sin^3 3\theta}\ t_{pq}\ t_{mn} &+ \\
+\frac{3}{2}\frac{\cos(3\theta)}{q^2 \sin(3\theta)}\ p_{pqmn} - \frac{9}{2}\frac{1}{q^3 \sin(3\theta)}\ w_{pqmn}
\end{aligned}$$

Appendix 704

Hyperelasticity, Detailed Derivations

(1995-1996-)

704.1 Chapter Summary and Highlights

704.2 Simo–Serrin's Formula

In order to derive the analytical gradient of the fourth order tensor

$$\mathcal{M}_{IJKL} = \frac{\partial M_{IJ}}{\partial C_{KL}} \tag{704.1}$$

we shall proceed by using the third equation in (106.86).

$$\begin{aligned}
\frac{\partial M_{IJ}}{\partial C_{KL}} &= \frac{1}{D_{(A)}} \left(I_{IKJL} - \frac{\partial I_1}{\partial \, C_{KL}} \delta_{IJ} + 2\lambda_{(A)} \, \frac{\partial \lambda_{(A)}}{\partial C_{KL}} \, \delta_{IJ} + \right. \\
&\left. + \frac{\partial I_3}{\partial C_{KL}} \, \lambda_{(A)}^{-2} \, (C^{-1})_{IJ} - 2\lambda_{(A)}^{-3} \, \frac{\partial \lambda_{(A)}}{\partial C_{KL}} \, I_3 \, (C^{-1})_{IJ} + \frac{\partial (C^{-1})_{IJ}}{\partial C_{KL}} \, \lambda_{(A)}^{-2} \, I_3 \right) - \\
&- \frac{1}{D_{(A)}^2} \, \frac{\partial D_{(A)}}{\partial C_{KL}} \left(C_{IJ} - \left(I_1 - \lambda_{(A)}^2 \right) \delta_{IJ} + I_3 \, \lambda_{(A)}^{-2} \, (C^{-1})_{IJ} \right)
\end{aligned} \tag{704.2}$$

where it was used that

$$\frac{\partial C_{IJ}}{\partial C_{KL}} = I_{IKJL} \tag{704.3}$$

Derivatives $\partial \lambda_{(A)} / \partial C_{KL}$ can be found by starting from equation for C_{IJ} (106.63) and differentiating it

$$dC_{IJ} = 2\lambda_A d\lambda_{(A)} \, \left(N_I^{(A)} N_J^{(A)} \right)_A + \lambda_A^2 \, \left(dN_I^{(A)} \, N_J^{(A)} \right)_A + \lambda_A^2 \, \left(N_I^{(A)} \, dN_J^{(A)} \right)_A \tag{704.4}$$

By premultiplying previous equation with $N_J^{(A)}$ and post-multiplying with $N_I^{(A)}$, and by noting that

$$N_I^{(A)} \, dN_I^{(A)} \equiv 0 \qquad ; \qquad \| N_I^{(A)} \| \equiv 1 \tag{704.5}$$

we get

$$N_J^{(A)} \, dC_{IJ} \, N_I^{(A)} = 2\lambda_A d\lambda_{(A)} \tag{704.6}$$

or

$$dC_{IJ} N_I^{(A)} \, N_J^{(A)} = dC_{IJ} \lambda_{(A)} \, M_{IJ}^{(A)} = 2\lambda_A d\lambda_{(A)} \quad \Rightarrow \quad \frac{\partial \lambda_A}{\partial C_{KL}} = \frac{1}{2} \, \lambda_{(A)} \, (M_{KL}^{(A)})_A \tag{704.7}$$

It can be proved[1] that

$$\frac{\partial I_1}{\partial \, C_{KL}} = \delta_{IJ} \quad ; \quad \frac{\partial I_2}{\partial \, C_{KL}} = I_1 \, \delta_{KL} - C_{KL} \quad ; \quad \frac{\partial I_3}{\partial \, C_{KL}} = I_3 \, (C^{-1})_{KL} \tag{704.8}$$

and since $I_3 = J^2$

$$\frac{\partial J}{\partial \, C_{KL}} = \frac{1}{2} \, J \, (C^{-1})_{KL} \tag{704.9}$$

---

[1] See Marsden and Hughes (1983)

With this in mind, equation (704.2) can be rewritten as:

$$\begin{aligned}
\frac{\partial M_{IJ}}{\partial C_{KL}} \; = \; & \frac{1}{D_{(A)}} \left(I_{IKJL} - \delta_{KL}\, \delta_{IJ} + 2\lambda_{(A)}^{2}\, \frac{1}{2}\, M_{KL}^{(A)}\, \delta_{IJ} + \right. \\
& + I_3\, \lambda_{(A)}^{-2}\, (C^{-1})_{IJ}\, (C^{-1})_{KL} - \lambda_{(A)}^{-2}\, I_3\, (C^{-1})_{IJ}\, M_{KL}^{(A)} + \\
& \left. + \frac{1}{2} \left((C^{-1})_{IK} (C^{-1})_{JL} + (C^{-1})_{IL} (C^{-1})_{JK} \right)\, \lambda_{(A)}^{-2}\, I_3 \right) - \\
& - \frac{1}{D_{(A)}}\, \frac{\partial D_{(A)}}{\partial C_{KL}} M_{IJ}
\end{aligned} \tag{704.10}$$

where the definition of M_{IJ} from equation (106.86) was used and also:

$$\frac{\partial (C^{-1})_{IJ}}{\partial C_{KL}} = -\frac{1}{2} \left((C^{-1})_{IK} (C^{-1})_{JL} + (C^{-1})_{IL} (C^{-1})_{JK} \right) = I_{IJKL}^{(C^{-1})} \tag{704.11}$$

Relation (704.11) can be obtained if one starts from the identity:

$$C_{IJ}\, (C^{-1})_{JK} = \delta_{IK} \tag{704.12}$$

which after differentiation reads:

$$\begin{aligned}
dC_{IJ}\, (C^{-1})_{JK} + C_{IJ}\, d(C^{-1})_{JK} = 0 \Rightarrow \\
\Rightarrow d(C^{-1})_{JK} = -(C^{-1})_{JM}\, dC_{MN}\, (C^{-1})_{NK} = \\
= -\frac{1}{2} \left((C^{-1})_{JM}\, (C^{-1})_{KN} + (C^{-1})_{JN}\, (C^{-1})_{KM} \right)\, dC_{MN} \Rightarrow \\
\Rightarrow \frac{\partial (C^{-1})_{JK}}{\partial C_{MN}} = -\frac{1}{2} \left((C^{-1})_{JM} (C^{-1})_{KN} + (C^{-1})_{JN} (C^{-1})_{KM} \right)
\end{aligned} \tag{704.13}$$

The derivative of $D_{(A)}$, that was defined in equation (106.77)as

$$D_{(A)} = 2\lambda_{(A)}^{4} - I_1 \lambda_{(A)}^{2} + I_3 \lambda_{(A)}^{-2} \tag{704.14}$$

is given by:

$$\begin{aligned}
\frac{\partial D_{(A)}}{\partial C_{KL}} &= 8\lambda_{(A)}^{3}\frac{\partial \lambda_{(A)}}{\partial C_{KL}} - \frac{\partial I_1}{\partial C_{KL}}\ \lambda_{(A)}^{2} - 2\lambda_{(A)} I_1\ \frac{\partial \lambda_{(A)}}{\partial C_{KL}} + \frac{\partial I_3}{\partial C_{KL}}\ \lambda_{(A)}^{-2} - 2\lambda_{(A)}^{-3} I_3\ \frac{\partial \lambda_{(A)}}{\partial C_{KL}} \\
&= 4\lambda_{(A)}^{4}\ M_{KL}^{(A)} - \delta_{KL}\ \lambda_{(A)}^{2} - \lambda_{(A)}^{2} I_1\ M_{KL}^{(A)} + I_3 (C^{-1})_{KL}\ \lambda_{(A)}^{-2} - \lambda_{(A)}^{-2} I_3\ M_{KL}^{(A)} \\
&= \left(4\lambda_{(A)}^{4} - \lambda_{(A)}^{2} I_1 - \lambda_{(A)}^{-2} I_3\right) M_{KL}^{(A)} - \delta_{KL}\ \lambda_{(A)}^{2} + I_3 (C^{-1})_{KL}\ \lambda_{(A)}^{-2} \\
&= D'_{(A)}\ M_{KL}^{(A)} - \delta_{KL}\ \lambda_{(A)}^{2} + I_3 (C^{-1})_{KL}\ \lambda_{(A)}^{-2}
\end{aligned} \tag{704.15}$$

where $D'_{(A)} = 4\lambda_{(A)}^{4} - \lambda_{(A)}^{2} I_1 - \lambda_{(A)}^{-2} I_3$. With the previous derivations, equation (704.10) can be written in expanded form as:

$$\begin{aligned}
\frac{\partial M_{IJ}}{\partial C_{KL}} = &\frac{1}{D_{(A)}} \Bigg(I_{IKJL} - \delta_{KL}\ \delta_{IJ} + 2\lambda_{(A)}^{2}\ \frac{1}{2}\ M_{KL}^{(A)}\ \delta_{IJ} + \\
&+ I_3\ \lambda_{(A)}^{-2}\ (C^{-1})_{IJ}\ (C^{-1})_{KL} - \lambda_{(A)}^{-2}\ I_3\ (C^{-1})_{IJ}\ M_{KL}^{(A)}\ + \\
&+ \frac{1}{2}\left((C^{-1})_{IK}(C^{-1})_{JL} + (C^{-1})_{IL}(C^{-1})_{JK}\right)\ \lambda_{(A)}^{-2}\ I_3\ - \\
&- \left(D'_{(A)}\ M_{KL}^{(A)} - \delta_{KL}\ \lambda_{(A)}^{2} + I_3 (C^{-1})_{KL}\ \lambda_{(A)}^{-2}\right) M_{IJ} \Bigg)
\end{aligned} \tag{704.16}$$

If one collects similar terms, equation (704.16), also known as *Simo–Serrin's formula* can be written in the final form as:

$$\begin{aligned}
\frac{\partial M_{IJ}}{\partial C_{KL}} = &\ \mathcal{M}_{IJKL} = \\
&\frac{1}{D_{(A)}}\Bigg(I_{IKJL} - \delta_{KL}\ \delta_{IJ} + \lambda_{(A)}^{2}\left(\delta_{IJ}\ M_{KL}^{(A)} + M_{IJ}^{(A)}\ \delta_{KL}\right) + \\
&+ I_3\ \lambda_{(A)}^{-2}\left((C^{-1})_{IJ}\ (C^{-1})_{KL} + \frac{1}{2}\left((C^{-1})_{IK}(C^{-1})_{JL} + (C^{-1})_{IL}(C^{-1})_{JK}\right)\right) - \\
&- \lambda_{(A)}^{-2}\ I_3\ \left((C^{-1})_{IJ}\ M_{KL}^{(A)} + M_{IJ}^{(A)}\ (C^{-1})_{KL}\right) - D'_{(A)}\ M_{IJ}^{(A)}\ M_{KL}^{(A)}\Bigg)
\end{aligned} \tag{704.17}$$

704.3 Derivation of $\partial^{2vol}W/(\partial C_{IJ}\ \partial C_{KL})$

The volumetric part $\partial^{2vol}W/(\partial C_{IJ}\ \partial C_{KL})$ can be derived by starting from the equation (106.96):

$$
\begin{aligned}
&\frac{\partial^{2\,vol}W}{\partial C_{IJ}\;\partial C_{KL}} = \\
&\frac{\partial\left(\frac{1}{2}\,\frac{\partial^{vol}W}{\partial J}\;J\;(C^{-1})_{IJ}\right)}{\partial C_{KL}} = \\
&\frac{1}{2}\,\frac{\partial\left(\frac{\partial^{vol}W}{\partial J}\right)}{\partial C_{KL}}\;J\;(C^{-1})_{IJ} + \frac{1}{2}\,\frac{\partial^{vol}W}{\partial J}\,\frac{\partial\,(J)}{\partial C_{KL}}\;(C^{-1})_{IJ} + \frac{1}{2}\,\frac{\partial^{vol}W}{\partial J}\;J\;\frac{\partial\left((C^{-1})_{IJ}\right)}{\partial C_{KL}} \\
&= \\
&\frac{1}{2}\,\frac{\partial^2\,\left({}^{vol}W\right)}{\partial J\;\partial J}\,\frac{\partial J}{\partial C_{KL}}J\;(C^{-1})_{IJ} + \frac{1}{2}\,\frac{\partial\;{}^{vol}W}{\partial J}\;\frac{1}{2}\;J\;(C^{-1})_{KL}\;(C^{-1})_{IJ} + \frac{1}{2}\,\frac{\partial^{vol}W}{\partial J}\;J\;I^{(C^{-1})}_{IJKL} \\
&= \\
&\frac{1}{4}\;J^2\;\frac{\partial^2\;{}^{vol}W}{\partial J\;\partial J}\;(C^{-1})_{KL}(C^{-1})_{IJ} + \frac{1}{4}\;J\frac{\partial^{vol}W}{\partial J}\;(C^{-1})_{KL}\;(C^{-1})_{IJ} + \frac{1}{2}\;J\;\frac{\partial^{vol}W}{\partial J}\;I^{(C^{-1})}_{IJKL} \\
&= \\
&\frac{1}{4}\left(J^2\;\frac{\partial^2\;{}^{vol}W}{\partial J\;\partial J} + J\frac{\partial^{vol}W}{\partial J}\right)(C^{-1})_{KL}\;(C^{-1})_{IJ} + \frac{1}{2}\;J\;\frac{\partial^{vol}W}{\partial J}\;I^{(C^{-1})}_{IJKL}
\end{aligned}
\tag{704.18}
$$

where equations (704.11) and (704.9) were used.

704.4 Derivation of $\partial^{2iso}W/(\partial C_{IJ}\;\partial C_{KL})$

The isochoric part $\partial^{2iso}W/(\partial C_{IJ}\;\partial C_{KL})$ can be derived by starting from equation (106.97)

$$
\begin{aligned}
\frac{\partial^{2\,iso}W(\lambda_{(A)})}{\partial C_{IJ}\partial C_{KL}} &= \\
\frac{1}{2}\,\frac{\partial\left(w_A\;(M^{(A)}_{IJ})_A\right)}{\partial C_{KL}} &= \\
\frac{1}{2}\,\frac{\partial w_A}{\partial C_{KL}}\;(M^{(A)}_{IJ})_A + \frac{1}{2}\;w_A\;\frac{\partial (M^{(A)}_{IJ})_A}{\partial C_{KL}} &= \\
\frac{1}{2}\,\frac{\partial w_A}{\partial \lambda_B}\frac{\partial \lambda_B}{\partial C_{KL}}\;(M^{(A)}_{IJ})_A + \frac{1}{2}\;w_A\;(\mathcal{M}^{(A)}_{IJKL})_A &= \\
\frac{1}{2}\,\frac{\partial w_A}{\partial \lambda_B}\frac{1}{2}\;\lambda_{(B)}\;(M^{(B)}_{KL})_B(M^{(A)}_{IJ})_A + \frac{1}{2}\;w_A\;(\mathcal{M}^{(A)}_{IJKL})_A &= \\
\frac{1}{4}\;Y_{AB}\;(M^{(B)}_{KL})_B\;(M^{(A)}_{IJ})_A + \frac{1}{2}\;w_A\;(\mathcal{M}^{(A)}_{IJKL})_A &
\end{aligned}
\tag{704.19}
$$

where equation (704.7) was used and tensor Y_{AB} is defined as:

$$
Y_{AB} = \frac{\partial w_A}{\partial \lambda_B}\;\lambda_{(B)}
\tag{704.20}
$$

704.5 Derivation of w_A

$$
w_A = \frac{\partial^{iso}W}{\partial \lambda_{(A)}}\;\lambda_A \;=\; \frac{\partial^{iso}W}{\partial \tilde{\lambda}_B}\;\frac{\partial \tilde{\lambda}_B}{\partial \lambda_{(A)}}\;\lambda_A
\tag{704.21}
$$

where $\tilde{\lambda}_B$ is the isochoric part of the stretch defined as

$$
\tilde{\lambda}_B = J^{-\frac{1}{3}}\;\lambda_B
\tag{704.22}
$$

From the definition of $\tilde{\lambda}_B$ in equation (704.22) it follows

$$
\frac{\partial \tilde{\lambda}_B}{\partial \lambda_{(A)}} \;=\; \frac{\partial J^{-\frac{1}{3}}}{\partial \lambda_{(A)}}\;\lambda_B + J^{-\frac{1}{3}}\;\frac{\partial \lambda_B}{\partial \lambda_{(A)}} \;= -\frac{1}{3}\;J^{-\frac{1}{3}}\;\lambda^{-1}_{(A)}\;\lambda_B + J^{-\frac{1}{3}}\;\delta_{B(A)}
\tag{704.23}
$$

since

$$
\frac{\partial J^{-\frac{1}{3}}}{\partial \lambda_{(A)}} = -\frac{1}{3}\;J^{-\frac{4}{3}}\;\;\frac{\partial \lambda_1\lambda_2\lambda_3}{\partial \lambda_{(A)}} = -\frac{1}{3}\;J^{-\frac{4}{3}}\;\;J\;\lambda^{-1}_{(A)} = -\frac{1}{3}\;J^{-\frac{1}{3}}\;\lambda^{-1}_{(A)}
\tag{704.24}
$$

then

$$
\begin{aligned}
w_A \;&=\; -\frac{1}{3}\;J^{-\frac{1}{3}}\;\frac{\partial^{iso}W}{\partial \tilde{\lambda}_B}\;\lambda^{-1}_{(A)}\;\lambda_B\;\lambda_{(A)} + J^{-\frac{1}{3}}\;\frac{\partial^{iso}W}{\partial \tilde{\lambda}_B}\;\delta_{B(A)}\;\lambda_{(A)} \\
&=\; -\frac{1}{3}\;\frac{\partial^{iso}W}{\partial \tilde{\lambda}_B}\;\tilde{\lambda}_B \;+\; \frac{\partial^{iso}W}{\partial \tilde{\lambda}_{(A)}}\;\tilde{\lambda}_{(A)}
\end{aligned}
\tag{704.25}
$$

704.6 Derivation of Y_{AB}

By starting from equation 704.20

$$Y_{AB} = \frac{\partial w_A}{\partial \lambda_B} \; \lambda_{(B)} \tag{704.26}$$

and by using equation 704.25

$$w_A = -\frac{1}{3} \; \frac{\partial^{iso}W}{\partial \tilde{\lambda}_C} \; \tilde{\lambda}_C \; + \frac{\partial^{iso}W}{\partial \tilde{\lambda}_{(A)}} \; \tilde{\lambda}_{(A)} \tag{704.27}$$

we can write:

$$Y_{AB} = \frac{\partial w_A}{\partial \tilde{\lambda}_D} \; \frac{\partial \tilde{\lambda}_D}{\partial \lambda_B} \; \lambda_{(B)} \tag{704.28}$$

By first considering $\partial w_A / \partial \tilde{\lambda}_D$ we get:

$$\begin{aligned}
\frac{\partial w_A}{\partial \tilde{\lambda}_D} \; &= \; \frac{\partial \left(-\frac{1}{3} \; \frac{\partial^{iso}W}{\partial \tilde{\lambda}_C} \; \tilde{\lambda}_C \; + \frac{\partial^{iso}W}{\partial \tilde{\lambda}_{(A)}} \; \tilde{\lambda}_{(A)} \right)}{\partial \tilde{\lambda}_D} \\
&= \; -\frac{1}{3} \; \frac{\partial^{2\,iso}W}{\partial \tilde{\lambda}_C \partial \tilde{\lambda}_D} \; \tilde{\lambda}_C - \frac{1}{3} \; \frac{\partial^{iso}W}{\partial \tilde{\lambda}_C} \; \frac{\partial \tilde{\lambda}_C}{\partial \tilde{\lambda}_D} + \frac{\partial^{2\,iso}W}{\partial \tilde{\lambda}_{(A)} \partial \tilde{\lambda}_D} \; \tilde{\lambda}_{(A)} + \frac{\partial^{iso}W}{\partial \tilde{\lambda}_{(A)}} \; \frac{\partial \tilde{\lambda}_{(A)}}{\partial \tilde{\lambda}_D} \\
&= \; -\frac{1}{3} \; \frac{\partial^{2\,iso}W}{\partial \tilde{\lambda}_C \partial \tilde{\lambda}_D} \; \tilde{\lambda}_C - \frac{1}{3} \; \frac{\partial^{iso}W}{\partial \tilde{\lambda}_C} \; \delta_{CD} + \frac{\partial^{2\,iso}W}{\partial \tilde{\lambda}_{(A)} \partial \tilde{\lambda}_D} \; \tilde{\lambda}_{(A)} + \frac{\partial^{iso}W}{\partial \tilde{\lambda}_{(A)}} \; \delta_{(A)D} \\
&= \; -\frac{1}{3} \; \frac{\partial^{2\,iso}W}{\partial \tilde{\lambda}_C \partial \tilde{\lambda}_D} \; \tilde{\lambda}_C - \frac{1}{3} \; \frac{\partial^{iso}W}{\partial \tilde{\lambda}_D} \; + \frac{\partial^{2\,iso}W}{\partial \tilde{\lambda}_{(A)} \partial \tilde{\lambda}_D} \; \tilde{\lambda}_{(A)} + \frac{\partial^{iso}W}{\partial \tilde{\lambda}_{(A)}} \; \delta_{(A)D}
\end{aligned} \tag{704.29}$$

Next, from equation 704.23, we have that

$$\frac{\partial \tilde{\lambda}_D}{\partial \lambda_{(B)}} = -\frac{1}{3} \; J^{-\frac{1}{3}} \; \lambda_{(B)}^{-1} \; \lambda_D + J^{-\frac{1}{3}} \; \delta_{D(B)} \tag{704.30}$$

and by multiplying the result for $\partial w_A / \partial \tilde{\lambda}_D$ from equation 704.29 and the result for $\partial \tilde{\lambda}_D / \partial \lambda_{(B)}$ from equation 704.30 we obtain:

$$\begin{aligned}
\frac{\partial w_A}{\partial \tilde{\lambda}_D} \; \frac{\partial \tilde{\lambda}_D}{\partial \lambda_{(B)}} = \; & + \; \frac{1}{9} \; \frac{\partial^{2\,iso}W}{\partial \tilde{\lambda}_C \partial \tilde{\lambda}_D} \; \tilde{\lambda}_C \; J^{-\frac{1}{3}} \; \lambda_{(B)}^{-1} \; \lambda_D \; - \; \frac{1}{3} \; \frac{\partial^{2\,iso}W}{\partial \tilde{\lambda}_C \partial \tilde{\lambda}_D} \; \tilde{\lambda}_C J^{-\frac{1}{3}} \; \delta_{D(B)} \\
& + \; \frac{1}{9} \; \frac{\partial^{iso}W}{\partial \tilde{\lambda}_D} \; J^{-\frac{1}{3}} \; \lambda_{(B)}^{-1} \; \lambda_D \; - \; \frac{1}{3} \; \frac{\partial^{iso}W}{\partial \tilde{\lambda}_D} \; J^{-\frac{1}{3}} \; \delta_{D(B)} \\
& - \; \frac{1}{3} \; \frac{\partial^{2\,iso}W}{\partial \tilde{\lambda}_{(A)} \partial \tilde{\lambda}_D} \; \tilde{\lambda}_{(A)} J^{-\frac{1}{3}} \; \lambda_{(B)}^{-1} \; \lambda_D \; + \; \frac{\partial^{2\,iso}W}{\partial \tilde{\lambda}_{(A)} \partial \tilde{\lambda}_D} \; \tilde{\lambda}_{(A)} J^{-\frac{1}{3}} \; \delta_{D(B)} \\
& - \; \frac{1}{3} \; \frac{\partial^{iso}W}{\partial \tilde{\lambda}_{(A)}} \; \delta_{(A)D} J^{-\frac{1}{3}} \; \lambda_{(B)}^{-1} \; \lambda_D \; + \; \frac{\partial^{iso}W}{\partial \tilde{\lambda}_{(A)}} \; \delta_{(A)D} J^{-\frac{1}{3}} \; \delta_{D(B)} \\
\\
= \; & + \; \frac{1}{9} \; \frac{\partial^{2\,iso}W}{\partial \tilde{\lambda}_C \partial \tilde{\lambda}_D} \; \tilde{\lambda}_C \; \lambda_{(B)}^{-1} \; \tilde{\lambda}_D \; - \; \frac{1}{3} \; \frac{\partial^{2\,iso}W}{\partial \tilde{\lambda}_C \partial \tilde{\lambda}_{(B)}} \; \tilde{\lambda}_C \; J^{-\frac{1}{3}} \\
& + \; \frac{1}{9} \; \frac{\partial^{iso}W}{\partial \tilde{\lambda}_D} \; \lambda_{(B)}^{-1} \; \tilde{\lambda}_D \; - \; \frac{1}{3} \; \frac{\partial^{iso}W}{\partial \tilde{\lambda}_{(B)}} \; J^{-\frac{1}{3}} \\
& - \; \frac{1}{3} \; \frac{\partial^{2\,iso}W}{\partial \tilde{\lambda}_{(A)} \partial \tilde{\lambda}_D} \; \tilde{\lambda}_{(A)} \lambda_{(B)}^{-1} \; \tilde{\lambda}_D \; + \; \frac{\partial^{2\,iso}W}{\partial \tilde{\lambda}_{(A)} \partial \tilde{\lambda}_{(B)}} \; \tilde{\lambda}_{(A)} \; J^{-\frac{1}{3}} \\
& - \; \frac{1}{3} \; \frac{\partial^{iso}W}{\partial \tilde{\lambda}_{(A)}} \; \lambda_{(B)}^{-1} \; \tilde{\lambda}_{(A)} \; + \; \frac{\partial^{iso}W}{\partial \tilde{\lambda}_{(A)}} \; \delta_{(A)(B)} \; J^{-\frac{1}{3}}
\end{aligned} \tag{704.31}$$

where equation 704.22 was used. The final form for Y_{AB} is obtained by multiplying equation 704.31 with $\tilde{\lambda}_{(B)}$ to obtain:

$$
\begin{aligned}
Y_{AB} = \quad & \frac{\partial w_A}{\partial \tilde{\lambda}_D} \frac{\partial \tilde{\lambda}_D}{\partial \lambda_B} \lambda_{(B)} = \\
+ \quad & \frac{1}{9} \frac{\partial^{2\,iso} W}{\partial \tilde{\lambda}_C \partial \tilde{\lambda}_D} \tilde{\lambda}_C \, \lambda^{-1}_{(B)} \, \tilde{\lambda}_D \, \lambda_{(B)} \quad - \quad \frac{1}{3} \frac{\partial^{2\,iso} W}{\partial \tilde{\lambda}_C \partial \tilde{\lambda}_{(B)}} \tilde{\lambda}_C \, \tilde{\lambda}_{(B)} \\
+ \quad & \frac{1}{9} \frac{\partial^{iso} W}{\partial \tilde{\lambda}_D} \lambda^{-1}_{(B)} \, \tilde{\lambda}_D \, \lambda_{(B)} \quad - \quad \frac{1}{3} \frac{\partial^{iso} W}{\partial \tilde{\lambda}_{(B)}} \tilde{\lambda}_{(B)} \\
- \quad & \frac{1}{3} \frac{\partial^{2\,iso} W}{\partial \tilde{\lambda}_{(A)} \partial \tilde{\lambda}_D} \tilde{\lambda}_{(A)} \lambda^{-1}_{(B)} \, \tilde{\lambda}_D \, \lambda_{(B)} \quad + \quad \frac{\partial^{2\,iso} W}{\partial \tilde{\lambda}_{(A)} \partial \tilde{\lambda}_{(B)}} \tilde{\lambda}_{(A)} \, \tilde{\lambda}_{(B)} \\
- \quad & \frac{1}{3} \frac{\partial^{iso} W}{\partial \tilde{\lambda}_{(A)}} \, \lambda^{-1}_{(B)} \, \tilde{\lambda}_{(A)} \, \lambda_{(B)} \quad + \quad \frac{\partial^{iso} W}{\partial \tilde{\lambda}_{(A)}} \delta_{(A)(B)} \, \tilde{\lambda}_{(B)}
\end{aligned}
\tag{704.32}
$$

By recognizing that $\lambda^{-1}_{(B)} \lambda_{(B)} \equiv 1$ and after rearranging elements, we can finally write the equation for Y_{AB} as:

$$
\begin{aligned}
Y_{AB} = \quad & \frac{\partial^{iso} W}{\partial \tilde{\lambda}_{(A)}} \delta_{(A)(B)} \, \tilde{\lambda}_{(B)} + \frac{\partial^{2\,iso} W}{\partial \tilde{\lambda}_{(A)} \partial \tilde{\lambda}_{(B)}} \tilde{\lambda}_{(A)} \, \tilde{\lambda}_{(B)} \\
- \quad & \frac{1}{3} \left(\frac{\partial^{2\,iso} W}{\partial \tilde{\lambda}_C \partial \tilde{\lambda}_{(B)}} \tilde{\lambda}_C \, \tilde{\lambda}_{(B)} + \frac{\partial^{iso} W}{\partial \tilde{\lambda}_{(B)}} \tilde{\lambda}_{(B)} + \frac{\partial^{2\,iso} W}{\partial \tilde{\lambda}_{(A)} \partial \tilde{\lambda}_D} \tilde{\lambda}_{(A)} \tilde{\lambda}_D + \frac{\partial^{iso} W}{\partial \tilde{\lambda}_{(A)}} \tilde{\lambda}_{(A)} \right) \\
+ \quad & \frac{1}{9} \left(\frac{\partial^{2\,iso} W}{\partial \tilde{\lambda}_C \partial \tilde{\lambda}_D} \tilde{\lambda}_C \, \tilde{\lambda}_D + \frac{\partial^{iso} W}{\partial \tilde{\lambda}_D} \tilde{\lambda}_D \right)
\end{aligned}
\tag{704.33}
$$

Appendix 705

Body and Surface Wave Analytic Solutions

(2005-2001-2010-2011-2018-2019-2021-)

(In collaboration with Dr. Nima Tafazzoli, Mr. Chang-Gyun Jeong, and Dr. Hexiang Wang)

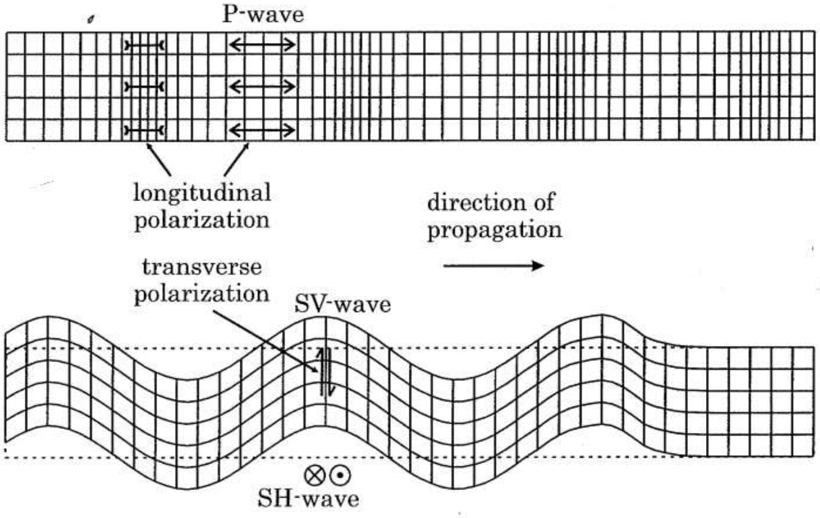


Figure 705.1: Motion due to plane pressure waves (P-wave) and shear waves (S-wave) (Semblat and Pecker, 2009)

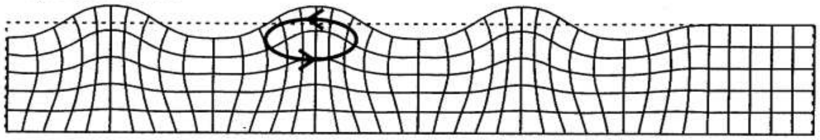


Figure 705.2: Particle movement due to surface Rayleigh waves near free surface (Semblat and Pecker, 2009)

705.1 3D Seismic Wave Field: Analytic Solution

In this chapter, wave field generation methods using analytic solution, and frequency wavenumber integration method are introduced. Theoretical background and examples are presented for each method.

705.1.1 Analytic solution

Seismic waves can be categorized as body waves and surface waves. The seismic body waves are traveling through the interior of the earth whereas the surface waves are traveling through the surface of the earth.

There are two different body waves, such as the pressure wave (also called as P wave, Figure 705.1 top) and the shear waves (also called as S wave, Figure 705.1 bottom). The shear waves which have the same velocity V_S can be polarized along its plane location to the direction of propagation (vertical plane - SV and horizontal plane - SH, Figure 705.1 bottom).

Surface waves are mainly categorized as Rayliegh waves (Rayleigh, 1885) and Love waves (Love,

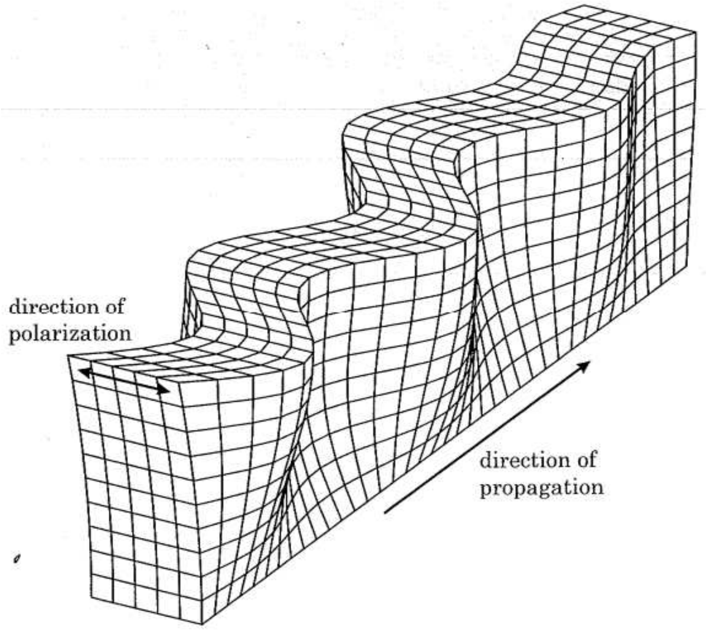


Figure 705.3: Displacements due to surface Love waves (Semblat and Pecker, 2009)

1911). The Rayleigh waves induce elliptical ground movement near the surface (Figure 705.2) whereas the Love waves induce shearing movement (Figure 705.3).

Following sections introduce full three dimensional exact-solution and its examples for plane body and surface waves in homogeneous media. The chapter does not include full derivation of equations since it's beyond the scope. Instead, final equations for body and surface waves are presented.

The original works for those problems are done by Green (1848), Knott (1899), and Wiechert and Zoeppritz (1907). The notations and equations hereafter are mainly based on Semblat and Pecker (2009) and Aki and Richard's work (Aki and Richards, 2002).

705.1.1.1 Wave equations for body waves

Hereafter, the reflection and refraction coefficients are indicated by using its wave component symbols. An acute and grave accents are adapted to explain the direction of propagation. The acute accent indicates an upcoming wave, and the grave accent indicates a down-going wave (e.g. $\acute{P}$, $\grave{P}$). For example, if the upcoming incident wave type is P and down-going reflected wave type is S, then reflected wave will be indicated as $\acute{P}\grave{S}$.

Reflected and transmitted waves arising from incident SH wave In the case of SH incident wave on the interface between two half-spaces, reflected wave is SH wave (Figure 705.4). The vector displacements

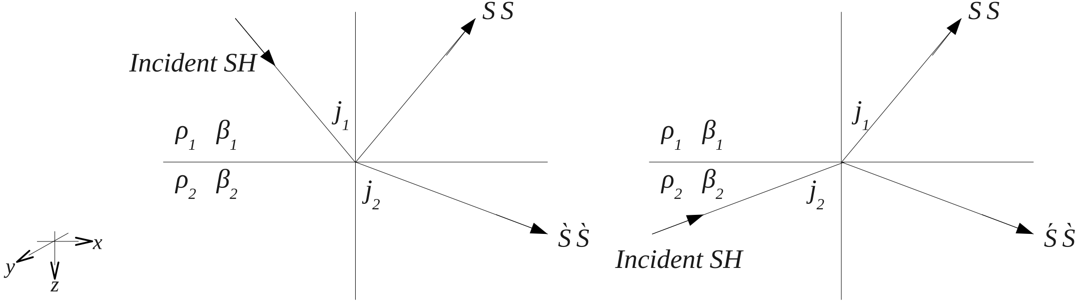


Figure 705.4: Schematic cartoon to show all possible coefficients of reflection and transmission with SH incident wave

for the downgoing and upgoing incident SH waves can be calculated as below equation (705.1) and (705.2), respectively (Aki and Richards, 2002).

$$
\begin{aligned}
\text{(Downgoing SH)} &= S(0, S, 0) exp \left[i\omega \left(px + \frac{\cos j_1}{\beta_1} z - t \right) \right] \\
\text{(Upgoing SH)} &= S(0, S, 0) \grave{S}\acute{S} exp \left[i\omega \left(px - \frac{\cos j_1}{\beta_1} z - t \right) \right] \\
\text{(Downgoing SH)} &= S(0, S, 0) \grave{S}\grave{S} exp \left[i\omega \left(px + \frac{\cos j_2}{\beta_2} z - t \right) \right]
\end{aligned}
\tag{705.1}
$$

$$
\begin{aligned}
\text{(Upgoing SH)} &= S(0, S, 0) exp \left[i\omega \left(px - \frac{\cos j_2}{\beta_2} z - t \right) \right] \\
\text{(Upgoing SH)} &= S(0, S, 0) \acute{S}\acute{S} exp \left[i\omega \left(px - \frac{\cos j_1}{\beta_1} z - t \right) \right] \\
\text{(downgoing SH)} &= S(0, S, 0) \acute{S}\grave{S} exp \left[i\omega \left(px + \frac{\cos j_2}{\beta_2} z - t \right) \right]
\end{aligned}
\tag{705.2}
$$

where,

$$
\begin{aligned}
\grave{S}\acute{S} &= \frac{\rho_1 \beta_1 \cos j_1 - \rho_2 \beta_2 \cos j_2}{\Delta} \\
\acute{S}\acute{S} &= \frac{2 \rho_2 \beta_2 \cos j_2}{\Delta} \\
\grave{S}\grave{S} &= \frac{2 \rho_1 \beta_1 \cos j_1}{\Delta} \\
\acute{S}\grave{S} &= -\grave{S}\acute{S} \\
\Delta &= \rho_1 \beta_1 \cos j_1 + \rho_2 \beta_2 \cos j_2
\end{aligned}
\tag{705.3}
$$

where α is the P wave velocity, β is the S wave velocity, ρ is the density, $p = (\sin i)/\alpha = (\sin j)/\beta$ is the ray parameter, and S is the amplitude of the incident wave.

Figure 705.4 shows all possible reflection and transmission coefficients with incident SH waves. Equation (705.4) is a ‘scattering matrix’ which includes every possible reflection and transmission coefficients

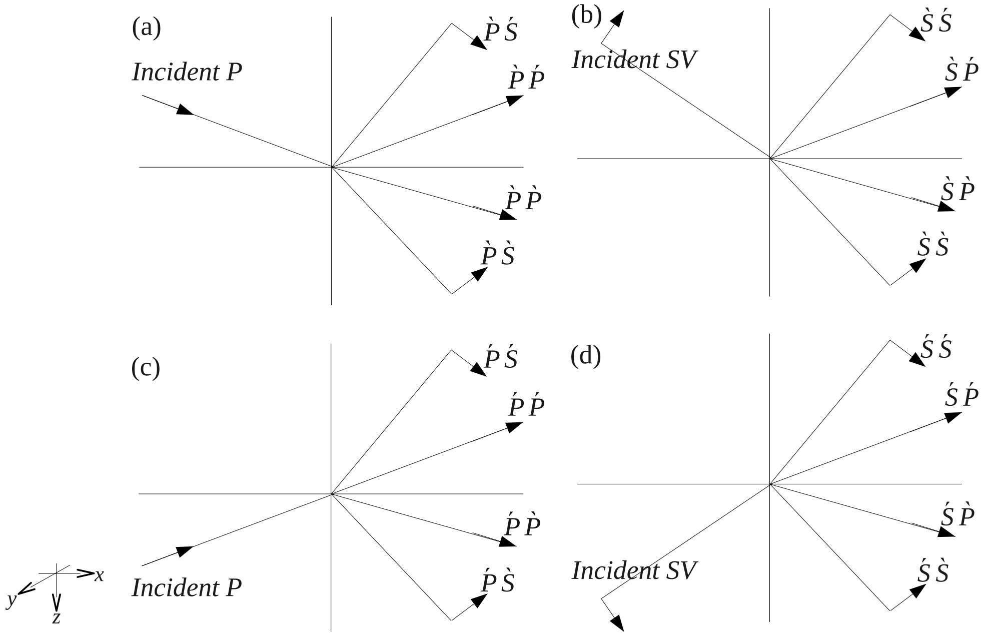


Figure 705.5: Schematic cartoon to show all possible coefficients of reflection and transmission with P/SV incident wave

for the problem. The matrix components have one to one relation with Figure 705.4.

$$\begin{pmatrix} \grave{S}\acute{S} & \acute{S}\acute{S} \\ \grave{S}\acute{S} & \acute{S}\grave{S} \end{pmatrix} \tag{705.4}$$

Reflected and transmitted waves arising from incident P/SV wave The displacements generated by the downgoing incident P/SV and upgoing incident P/SV waves can be calculated as below equation (705.5), (705.6), (705.7), and (705.8), respectively (Figure 705.5) (Aki and Richards, 2002).

$$\begin{aligned}
\text{(Downgoing P)} &= S(\sin i_1, 0, \cos i_1) exp\left[i\omega\left(px + \frac{\cos i_1}{\alpha_1}z - t\right)\right] \\
\text{(Upgoing P)} &= S(\sin i_1, 0, -\cos i_1)\grave{P}\acute{P} exp\left[i\omega\left(px - \frac{\cos i_1}{\alpha_1}z - t\right)\right] \\
\text{(Upgoing SV)} &= S(\cos j_1, 0, \sin j_1)\grave{P}\acute{S} exp\left[i\omega\left(px - \frac{\cos j_1}{\beta_1}z - t\right)\right] \\
\text{(Downgoing P)} &= S(\sin i_2, 0, \cos i_2)\grave{P}\grave{P} exp\left[i\omega\left(px + \frac{\cos i_2}{\alpha_2}z - t\right)\right] \\
\text{(Downgoing SV)} &= S(\cos j_2, 0, -\sin j_2)\grave{P}\grave{S} exp\left[i\omega\left(px + \frac{\cos j_2}{\beta_2}z - t\right)\right]
\end{aligned} \tag{705.5}$$

$$
\begin{aligned}
\text{(Downgoing SV)} &= S(\cos j_1, 0, -\sin j_1) exp \left[i\omega \left(px + \frac{\cos j_1}{\beta_1} z - t \right) \right] \\
\text{(Upgoing P)} &= S(\sin i_1, 0, -\cos i_1) \grave{S}\acute{P} exp \left[i\omega \left(px - \frac{\cos i_1}{\alpha_1} z - t \right) \right] \\
\text{(Upgoing SV)} &= S(\cos j_1, 0, \sin j_1) \grave{S}\acute{S} exp \left[i\omega \left(px - \frac{\cos j_1}{\beta_1} z - t \right) \right] \\
\text{(Downgoing P)} &= S(\sin i_2, 0, \cos i_2) \grave{S}\grave{P} exp \left[i\omega \left(px + \frac{\cos i_2}{\alpha_2} z - t \right) \right] \\
\text{(Downgoing SV)} &= S(\cos j_2, 0, -\sin j_2) \grave{S}\grave{S} exp \left[i\omega \left(px + \frac{\cos j_2}{\beta_2} z - t \right) \right]
\end{aligned}
\tag{705.6}
$$

$$
\begin{aligned}
\text{(Upgoing P)} &= S(\sin i_2, 0, -\cos i_2) exp \left[i\omega \left(px - \frac{\cos i_2}{\alpha_2} z - t \right) \right] \\
\text{(Upgoing P)} &= S(\sin i_1, 0, -\cos i_1) \acute{P}\acute{P} exp \left[i\omega \left(px - \frac{\cos i_1}{\alpha_1} z - t \right) \right] \\
\text{(Upgoing SV)} &= S(\cos j_1, 0, \sin j_1) \acute{P}\acute{S} exp \left[i\omega \left(px - \frac{\cos j_1}{\beta_1} z - t \right) \right] \\
\text{(Downgoing P)} &= S(\sin i_2, 0, \cos i_2) \acute{P}\grave{P} exp \left[i\omega \left(px + \frac{\cos i_2}{\alpha_2} z - t \right) \right] \\
\text{(Downgoing SV)} &= S(\cos j_2, 0, -\sin j_2) \acute{P}\grave{S} exp \left[i\omega \left(px + \frac{\cos j_2}{\beta_2} z - t \right) \right]
\end{aligned}
\tag{705.7}
$$

$$
\begin{aligned}
\text{(Upgoing SV)} &= S(\cos j_2, 0, \sin j_2) exp \left[i\omega \left(px - \frac{\cos j_2}{\beta_2} z - t \right) \right] \\
\text{(Upgoing P)} &= S(\sin i_1, 0, -\cos i_1) \acute{S}\acute{P} exp \left[i\omega \left(px - \frac{\cos i_1}{\alpha_1} z - t \right) \right] \\
\text{(Upgoing SV)} &= S(\cos j_1, 0, \sin j_1) \acute{S}\acute{S} exp \left[i\omega \left(px - \frac{\cos j_1}{\beta_1} z - t \right) \right] \\
\text{(Downgoing P)} &= S(\sin i_2, 0, \cos i_2) \acute{S}\grave{P} exp \left[i\omega \left(px + \frac{\cos i_2}{\alpha_2} z - t \right) \right] \\
\text{(Downgoing SV)} &= S(\cos j_2, 0, -\sin j_2) \acute{S}\grave{S} exp \left[i\omega \left(px + \frac{\cos j_2}{\beta_2} z - t \right) \right]
\end{aligned}
\tag{705.8}
$$

where,

$$
\begin{aligned}
\grave{P}\acute{P} &= \left[\left(b\frac{\cos i_1}{\alpha_1} - c\frac{\cos i_2}{\alpha_2} \right) F - \left(a + d\frac{\cos i_1}{\alpha_1}\frac{\cos j_2}{\beta_2} \right) Hp^2 \right] /\mathbf{D} \\
\grave{P}\acute{S} &= -2\frac{\cos i_1}{\alpha_1} \left(ab + cd\frac{\cos i_2}{\alpha_2}\frac{\cos j_2}{\beta_2} \right) p\alpha_1/(\beta_1 \mathbf{D}) \\
\grave{P}\grave{P} &= 2\rho_1 \frac{\cos i_1}{\alpha_1} F\alpha_1/(\alpha_2 \mathbf{D}) \\
\grave{P}\grave{S} &= 2\rho_1 \frac{\cos i_1}{\alpha_1} Hp\alpha_1/(\beta_2 \mathbf{D}) \\
\grave{S}\acute{P} &= -2\frac{\cos j_1}{\beta_1} \left(ab + cd\frac{\cos i_2}{\alpha_2}\frac{\cos j_2}{\beta_2} \right) p\beta_1/(\alpha_1 \mathbf{D}) \\
\grave{S}\acute{S} &= -\left[\left(b\frac{\cos j_1}{\beta_1} - c\frac{\cos j_2}{\beta_2} \right) E - \left(a + d\frac{\cos i_2}{\alpha_2}\frac{\cos j_1}{\beta_1} \right) Gp^2 \right] /\mathbf{D} \\
\grave{S}\grave{P} &= -2\rho_1 \frac{\cos j_1}{\beta_1} Gp\beta_1/(\alpha_2 \mathbf{D}) \\
\grave{S}\grave{S} &= 2\rho_1 \frac{\cos j_1}{\beta_1} E\beta_1/(\beta_2 \mathbf{D}) \\
\acute{P}\acute{P} &= 2\rho_2 \frac{cosi_2}{\alpha_2} F\alpha_2 \ (\alpha_1 \mathbf{D}) \\
\\
\acute{P}\acute{S} &= -2\rho_2 \frac{\cos i_2}{\alpha_2} Gp\alpha_2/(\alpha_2 \mathbf{D}) \\
\acute{P}\grave{P} &= -\left[\left(b\frac{\cos i_1}{\alpha_1} - c\frac{\cos i_2}{\alpha_2} \right) F + \left(a + d\frac{\cos i_2}{\alpha_2}\frac{\cos j_1}{\beta_1} \right) Gp^2 \right] /\mathbf{D} \\
\acute{P}\grave{S} &= 2\frac{\cos i_2}{\alpha_2} \left(ac + bd\frac{\cos i_1}{\alpha_1}\frac{\cos j_1}{\beta_1} \right) p\alpha_2/(\beta_2 \mathbf{D}) \\
\acute{S}\acute{P} &= 2\rho_2 \frac{\cos j_2}{\beta_2} Hp\beta_2/(\alpha_1 \mathbf{D}) \\
\acute{S}\acute{S} &= 2\rho_2 \frac{\cos j_2}{\beta_2} E\beta_2/(\beta_1 \mathbf{D}) \\
\acute{S}\grave{P} &= 2\frac{\cos j_2}{beta_2} \left(ac + bd\frac{\cos i_1}{\alpha_1}\frac{\cos j_1}{\beta_1} \right) p\beta_2/(\alpha_2 \mathbf{D}) \\
\acute{S}\grave{S} &= \left[\left(b\frac{\cos j_1}{\beta_1} - c\frac{\cos j_2}{\beta_2} \right) E + \left(a + d\frac{\cos i_1}{\alpha_1}\frac{\cos j_2}{\beta_2} \right) Hp^2 \right] /\mathbf{D}
\end{aligned}
\tag{705.9}
$$

$$
\begin{aligned}
a &= \rho_2(1 - 2\beta_2^2 p^2) - \rho_1(1 - 2\beta_1^2 p^2) \\
b &= \rho_2(1 - 2\beta_2^2 p^2) + 2\rho_1 \beta_1^2 p^2 \\
c &= \rho_1(1 - 2\beta_1^2 p^2) + 2\rho - 2\beta_2^2 p^2 \\
d &= 2(\rho_2 \beta_2^2 - \rho_1 \beta_1^2)
\end{aligned}
\tag{705.10}
$$

$$
\begin{aligned}
E &= b\frac{\cos i_1}{\alpha_1} + c\frac{\cos i_2}{\alpha_2} \\
F &= b\frac{\cos j_1}{\beta_1} + c\frac{\cos j_2}{\beta_2} \\
G &= a - d\frac{\cos i_1}{\alpha_1}\frac{\cos j_2}{\beta_2} \\
H &= a - d\frac{\cos i_2}{\alpha_2}\frac{\cos j_1}{\beta_1} \\
\mathbf{D} &= EF + GHp^2 = (\det \mathbf{M})/(\alpha_1\alpha_2\beta_1\beta_2)
\end{aligned}
\tag{705.11}
$$

$$
\mathbf{M} = \begin{pmatrix}
-\alpha_1 p & -\cos j_1 & \alpha_2 p & \cos j_2 \\
\cos i_1 & -\beta_1 p & \cos i_2 & -\beta_2 p \\
2\rho_1\beta_1^2 p\cos i_1 & \rho_1\beta_1(1-2\beta_1^2p^2) & 2\rho_2\beta_2^2 p\cos i_2 & \rho_2\beta_2(1-2\beta_2^2p^2) \\
-\rho_1\alpha_1(1-2\beta_1^2p^2) & 2\rho_1\beta_1^2 p\cos j_1 & \rho_2\alpha_2(1-2\beta_2^2p^2) & -2\rho_2\beta_2^2 p\cos j_2
\end{pmatrix}
\tag{705.12}
$$

where α is the P wave velocity, β is the S wave velocity, ρ is the mass density, $p = (\sin i)/\alpha = (\sin j)/\beta$ is the ray parameter, and S is the amplitude of the incident wave.

Similar as the incident SH wave case, Figure 705.5 shows all possible reflection and transmission coefficients. Below equation (705.13) is a 'scattering matrix' which includes every possible reflection and transmission coefficients for the problem. The matrix components have one to one relation with Figure 705.5.

$$
\begin{pmatrix}
\grave{P}\acute{P} & \grave{S}\acute{P} & \acute{P}\acute{P} & \acute{S}\acute{P} \\
\grave{P}\acute{S} & \grave{S}\acute{S} & \acute{P}\acute{S} & \acute{S}\acute{S} \\
\grave{P}\grave{P} & \grave{S}\grave{P} & \acute{P}\grave{P} & \acute{S}\grave{P} \\
\grave{P}\grave{S} & \grave{S}\grave{S} & \acute{P}\grave{S} & \acute{S}\grave{S}
\end{pmatrix}
\tag{705.13}
$$

705.1.1.2 Wave equations for surface waves

Surface wave with incident SH wave – Love wave The displacements of surface wave due to the incident SH wave can be obtained by solving wave equations under free-surface condition (zero traction on surface). Below equation (705.14) shows the time history displacement with incident SH wave

(Semblat and Pecker, 2009).

$$
\begin{aligned}
u_y^i &= A_{SH} exp\left[\frac{i\omega}{V_S}\left(x\sin\theta_i^{SH} + z\cos\theta_i^{SH} - V_S t\right)\right] \\
u_y^R &= R_{SH} exp\left[\frac{i\omega}{V_S}\left(x\sin\theta_R^{SH} - z\cos\theta_R^{SH} - V_S t\right)\right] \\
u_y &= u_y^i + u_y^R = 2A_{SH}\cos\left(\frac{\omega z\cos\theta_i^{SH}}{V_S}\right) exp\left[\frac{i\omega}{V_S}\left(x\sin\theta_i - V_S t\right)\right]
\end{aligned}
\tag{705.14}
$$

As shown in equation (705.14) displacements of waves induced by incident SH waves can be calculated by summing incident and reflected waves on the ground surface. The particle movement of a Love wave is a perpendicular to the propagation plane.

Surface wave with incident P/SV wave – Rayleigh wave. Equation (705.15) shows the displacements of surface wave due to the incident P/SV waves (Semblat and Pecker, 2009).

$$
\begin{aligned}
u_x &= \frac{i\omega}{V_R} A\left(e^{az} - \frac{2ab}{b^2 + \omega^2/V_R^2} e^{bz}\right) exp\left[\frac{i\omega}{V_R}\left(x - V_R t\right)\right] \\
u_z &= aA\left(e^{az} - \frac{2\omega^2/V_R^2}{b^2 + \omega^2/V_R^2} e^{bz}\right) exp\left[\frac{i\omega}{V_R}\left(x - V_R t\right)\right]
\end{aligned}
\tag{705.15}
$$

where, $a^2 = \frac{\omega^2}{V_R^2} - \frac{\omega^2}{V_P^2}$, and $b^2 = \frac{\omega^2}{V_R^2} - \frac{\omega^2}{V_S^2}$.

Imaginary term on u_x shows that the components has a 90° phase shift from u_z. The particle movement of the wave is ellipses in x, z plane.

705.2 Matlab code – body wave solution

Listing 705.1: Example MATLAB code for surface waves

```
% ==========================================================
% body.m: inclined wave propagation closed—form solution for layered ground
%
% ref. Waves and Vibrations in Soils (Semblat and Pecker)
%
% Chang—Gyun Jeong
% Last update: 05/24/2011
% ==========================================================

clear all; clc;

%% initial condition
% wave type 'P' / 'SV' / 'SH'
wave = 'P';

% time
```

```
t_low = 0; t_up = 1;
dt = 0.01;
t_no = (t_up − t_low) / dt;

% x and z
min_x = 0; max_x = 100;
min_z = 0; max_z = 100;
delta_x = 2; delta_z = 2;

x = min_x : delta_x : max_x; x = x';
z = min_z : delta_z : max_z; z = z';

% frequency
freq = 10;
omega = freq ∗ 2 ∗ pi;


% incident angle and amplitude
th_P_i = 20; A_P = 1;
th_SV_i = 20; A_SV = 1;
th_SH_i = 20; A_SH = 1;

% velocity and shear modulus
V_S1 = 1000; V_S2 = 300;

nu = 0.3;
gamma = 20000; % (N/m^3)

V_P1 = V_S1 ∗ sqrt((2 − 2 ∗ nu) / (1 − 2 ∗ nu));
V_P2 = V_S2 ∗ sqrt((2 − 2 ∗ nu) / (1 − 2 ∗ nu));

G_1 = gamma ∗ V_S1^2; G_2 = gamma ∗ V_S2^2;

X_S = V_S2 / V_S1;
X_1 = V_P1 / V_S1;
X_2 = V_P2 / V_S2;

%% calculation
switch upper(wave)
    case {'P'}
        % angle and amplitude
        th_P_P_r = th_P_i;
        th_P_P_t = asind(sind(th_P_i) ∗ V_P2 / V_P1);
        th_SV_P_r = asind(sind(th_P_i) ∗ V_S1 / V_P1);
        th_SV_P_t = asind(sind(th_P_i) ∗ V_S2 / V_P1);

        temp_left = [−sind(th_P_i), −cosd(th_SV_P_r), sind(th_P_P_t), −cos(th_SV_P_t);...
            cosd(th_P_i), −sind(th_SV_P_r), cosd(th_P_P_t), sind(th_SV_P_t);...
            sind(2 ∗ th_P_i), X_1 ∗ cosd(2 ∗ th_SV_P_r), X_1 / X_2 ∗ X_S...
            ∗ sind(2 ∗ th_P_P_t), −X_1 ∗ X_S ∗ cosd(2 ∗ th_SV_P_t);...
            −X_1 ∗ cosd(2 ∗ th_SV_P_r), sind(2 ∗ th_SV_P_r), X_2 ∗ X_S...
            ∗ cosd(2 ∗ th_SV_P_t), X_S ∗ sind(2 ∗ th_SV_P_t);];

        temp_right = A_P ∗ [sind(th_P_i); cosd(th_P_i); sind(2 ∗ th_P_i); X_1 ∗...
            cosd(2 ∗ th_SV_P_r)];

        temp = temp_left \ temp_right;

        R_SV_P = temp(1, 1);
        R_P_P = temp(2, 1);
        T_SV_P = temp(3, 1);
        T_P_P = temp(4, 1);

        % initialize matrix
```

```
        ux_P_i = zeros(max(size(z)), max(size(x)));
        uz_P_i = zeros(max(size(z)), max(size(x)));
        ux_P_P_r = zeros(max(size(z)), max(size(x)));
        uz_P_P_r = zeros(max(size(z)), max(size(x)));
        ux_SV_P_r = zeros(max(size(z)), max(size(x)));
        uz_SV_P_r = zeros(max(size(z)), max(size(x)));
        ux_P_P_t = zeros(max(size(z)), max(size(x)));
        uz_P_P_t = zeros(max(size(z)), max(size(x)));
        ux_SV_P_t = zeros(max(size(z)), max(size(x)));
        uz_SV_P_t = zeros(max(size(z)), max(size(x)));

        % calculate incident, reflected, and transmitted wave
        for t = t_low : dt : t_up
            for i = 1 : max(size(x))
                for j = 1 : max(size(z))
                    % incident P
                    ux_P_i(i, j) = real(A_P * sind(th_P_i) * (exp((1i * omega / V_P1) ...
                        * (x(i, 1) * sind(th_P_i) + z(j, 1) * cosd(th_P_i) - V_P1 * t))));
                    uz_P_i(i, j) = real(A_P * cosd(th_P_i) * (exp((1i * omega / V_P1) ...
                        * (x(i, 1) * sind(th_P_i) + z(j, 1) * cosd(th_P_i) - V_P1 * t))));
                    % reflected P
                    ux_P_P_r(i, j) = real(R_P_P * sind(th_P_P_r) * (exp((1i * omega / V_P1) ...
                        * (x(i, 1) * sind(th_P_P_r) - z(j, 1) * cosd(th_P_P_r) - V_P1 * t))));
                    uz_P_P_r(i, j) = real(-R_P_P * cosd(th_P_P_r) * (exp((1i * omega / V_P1) ...
                        * (x(i, 1) * sind(th_P_P_r) - z(j, 1) * cosd(th_P_P_r) - V_P1 * t))));
                    % reflected SV
                    ux_SV_P_r(i, j) = real(R_SV_P * cosd(th_SV_P_r) * (exp((1i * omega / V_S1) ...
                        * (x(i, 1) * sind(th_SV_P_r) - z(j, 1) * cosd(th_SV_P_r) - V_S1 * t))));
                    uz_SV_P_r(i, j) = real(R_SV_P * sind(th_SV_P_r) * (exp((1i * omega / V_S1) ...
                        * (x(i, 1) * sind(th_SV_P_r) - z(j, 1) * cosd(th_SV_P_r) - V_S1 * t))));
                    % transmitted P
                    ux_P_P_t(i, j) = real(T_P_P * sind(th_P_P_t) * (exp((1i * omega / V_P2) ...
                        * (x(i, 1) * sind(th_P_P_t) + z(j, 1) * cosd(th_P_P_t) - V_P2 * t))));
                    uz_P_P_t(i, j) = real(T_P_P * cosd(th_P_P_t) * (exp((1i * omega / V_P2) ...
                        * (x(i, 1) * sind(th_P_P_t) + z(j, 1) * cosd(th_P_P_t) - V_P2 * t))));
                    % transmitted SV
                    ux_SV_P_t(i, j) = real(-T_SV_P * cosd(th_SV_P_t) * (exp((1i * omega / V_S2) ...
                        * (x(i, 1) * sind(th_SV_P_t) + z(j, 1) * cosd(th_SV_P_t) - V_S2 * t))));
                    uz_SV_P_t(i, j) = real(T_SV_P * sind(th_SV_P_t) * (exp((1i * omega / V_S2) ...
                        * (x(i, 1) * sind(th_SV_P_t) + z(j, 1) * cosd(th_SV_P_t) - V_S2 * t))));
                end
            end

            % plot
            subplot(3,4,1)
            surf(x, z, ux_P_i,'EdgeColor','none'); axis([0 100 0 100 -4 4])
            xlabel('x direction'); ylabel('z direction'); view(3)
            title('Incident P x comp.')
            subplot(3,4,2)
            surf(x, z, uz_P_i,'EdgeColor','none'); axis([0 100 0 100 -4 4])
            xlabel('x direction'); ylabel('z direction'); view(3)
            title('Incident P z comp.')

            subplot(3,4,5)
            surf(x, z, ux_P_P_r,'EdgeColor','none'); axis([0 100 0 100 -4 4])
            xlabel('x direction'); ylabel('z direction'); view(3)
            title('Reflected P x comp.')
            subplot(3,4,6)
            surf(x, z, uz_P_P_r,'EdgeColor','none'); axis([0 100 0 100 -4 4])
            xlabel('x direction'); ylabel('z direction'); view(3)
            title('Reflected P z comp.')
            subplot(3,4,7)
            surf(x, z, ux_SV_P_r,'EdgeColor','none'); axis([0 100 0 100 -4 4])
            xlabel('x direction'); ylabel('z direction'); view(3)
```

```
            title('Reflected SV x comp.')
            subplot(3,4,8)
            surf(x, z, uz_SV_P_r,'EdgeColor','none'); axis([0 100 0 100 −4 4])
            xlabel('x direction'); ylabel('z direction'); view(3)
            title('Reflected SV z comp.')

            subplot(3,4,9)
            surf(x, z, ux_P_P_t,'EdgeColor','none'); axis([0 100 0 100 −4 4])
            xlabel('x direction'); ylabel('z direction'); view(3)
            title('Refracted P x comp.')
            subplot(3,4,10)
            surf(x, z, uz_P_P_t,'EdgeColor','none'); axis([0 100 0 100 −4 4])
            xlabel('x direction'); ylabel('z direction'); view(3)
            title('Refracted P z comp.')
            subplot(3,4,11)
            surf(x, z, ux_SV_P_t,'EdgeColor','none'); axis([0 100 0 100 −4 4])
            title('Refracted SV x comp.')
            subplot(3,4,12)
            surf(x, z, uz_SV_P_t,'EdgeColor','none'); axis([0 100 0 100 −4 4])
            title('Refracted SV z comp.')

            drawnow
        end

    case {'SV'}
        % angle and amplitude
        th_P_SV_r = asind(sind(th_SV_i) ∗ V_P1 / V_S1);
        th_P_SV_t = asind(sind(th_SV_i) ∗ V_P2 / V_S1);
        th_SV_SV_r = th_SV_i;
        th_SV_SV_t = asind(sind(th_SV_i) ∗ V_S2 / V_S1);

        temp_left = [cosd(th_SV_i), sind(th_P_SV_r), cosd(th_SV_SV_t), sin(th_P_SV_t);...
            sind(th_SV_i), −cosd(th_P_SV_r), −sind(th_SV_SV_t), cosd(th_P_SV_t);...
            −cosd(2 ∗ th_SV_i), (−1 / X_1) ∗ sind(2 ∗ th_P_SV_r), X_S...
            ∗ cosd(2 ∗ th_SV_SV_t), (−X_S ∗ X_2) ∗ sind(2 ∗ th_P_SV_t);...
            −sind(2 ∗ th_SV_i), X_1 ∗ cosd(2 ∗ th_SV_SV_r), −X_S...
            ∗ sind(2 ∗ th_SV_SV_t), −X_2 ∗ X_S ∗ cosd(2 ∗ th_SV_SV_t);];

        temp_right = A_SV ∗ [cosd(th_SV_i); −sind(th_SV_i); cosd(2 ∗ th_SV_i); ...
            −sind(2 ∗ th_SV_i)];

        temp = temp_left \ temp_right;

        R_P_SV = temp(1, 1);
        R_SV_SV = temp(2, 1);
        T_P_SV = temp(3, 1);
        T_SV_SV = temp(4, 1);

        % initialize matrix
        ux_SV_i = zeros(max(size(z)), max(size(x)));
        uz_SV_i = zeros(max(size(z)), max(size(x)));
        ux_P_SV_r = zeros(max(size(z)), max(size(x)));
        uz_P_SV_r = zeros(max(size(z)), max(size(x)));
        ux_SV_SV_r = zeros(max(size(z)), max(size(x)));
        uz_SV_SV_r = zeros(max(size(z)), max(size(x)));
        ux_P_SV_t = zeros(max(size(z)), max(size(x)));
        uz_P_SV_t = zeros(max(size(z)), max(size(x)));
        ux_SV_SV_t = zeros(max(size(z)), max(size(x)));
        uz_SV_SV_t = zeros(max(size(z)), max(size(x)));

        % calculate incident, reflected, and transmitted wave
        for t = t_low : dt : t_up
        for i = 1 : max(size(x))
            for j = 1 : max(size(z))
```

```
            % incident SV
            ux_SV_i(i, j) = real(A_SV * cosd(th_SV_i) * (exp((1i * omega / V_S1) ...
                * (x(i, 1) * sind(th_SV_i) + z(j, 1) * cosd(th_SV_i) - V_S1 * t))));
            uz_SV_i(i, j) = real(-A_SV * sind(th_SV_i) * (exp((1i * omega / V_S1) ...
                * (x(i, 1) * sind(th_SV_i) + z(j, 1) * cosd(th_SV_i) - V_S1 * t))));
            % reflected P
            ux_P_SV_r(i, j) = real(R_P_SV * cosd(th_P_SV_r) * (exp((1i * omega / V_P1) ...
                * (x(i, 1) * sind(th_P_SV_r) - z(j, 1) * cosd(th_P_SV_r) - V_P1 * t))));
            uz_P_SV_r(i, j) = real(R_P_SV * sind(th_P_SV_r) * (exp((1i * omega / V_P1) ...
                * (x(i, 1) * sind(th_P_SV_r) - z(j, 1) * cosd(th_P_SV_r) - V_P1 * t))));
            % reflected SV
            ux_SV_SV_r(i, j) = real(-R_SV_SV * sind(th_SV_SV_r) * (exp((1i * omega / V_S1) ...
                * (x(i, 1) * sind(th_SV_SV_r) - z(j, 1) * cosd(th_SV_SV_r) - V_S1 * t))));
            uz_SV_SV_r(i, j) = real(R_SV_SV * cosd(th_SV_SV_r) * (exp((1i * omega / V_S1) ...
                * (x(i, 1) * sind(th_SV_SV_r) - z(j, 1) * cosd(th_SV_SV_r) - V_S1 * t))));
            % transmitted P
            ux_P_SV_t(i, j) = real(T_P_SV * cosd(th_P_SV_t) * (exp((1i * omega / V_P2) ...
                * (x(i, 1) * sind(th_P_SV_t) + z(j, 1) * cosd(th_P_SV_t) - V_P2 * t))));
            uz_P_SV_t(i, j) = real(-T_P_SV * sind(th_P_SV_t) * (exp((1i * omega / V_P2) ...
                * (x(i, 1) * sind(th_P_SV_t) + z(j, 1) * cosd(th_P_SV_t) - V_P2 * t))));
            % transmitted SV
            ux_SV_SV_t(i, j) = real(T_SV_SV * sind(th_SV_SV_t) * (exp((1i * omega / V_S2) ...
                * (x(i, 1) * sind(th_SV_SV_t) + z(j, 1) * cosd(th_SV_SV_t) - V_S2 * t))));
            uz_SV_SV_t(i, j) = real(-T_SV_SV * cosd(th_SV_SV_t) * (exp((1i * omega / V_S2) ...
                * (x(i, 1) * sind(th_SV_SV_t) + z(j, 1) * cosd(th_SV_SV_t) - V_S2 * t))));
        end
    end

        % plot
        subplot(3,4,1)
        surf(x, z, ux_SV_i,'EdgeColor','none'); axis([0 100 0 100 -4 4])
        xlabel('x direction'); ylabel('z direction'); view(3)
        title('Incident SV x comp.')

        subplot(3,4,2)
        surf(x, z, uz_SV_i,'EdgeColor','none'); axis([0 100 0 100 -4 4])
        xlabel('x direction'); ylabel('z direction'); view(3)
        title('Incident SV z comp.')

        subplot(3,4,5)
        surf(x, z, ux_P_SV_r,'EdgeColor','none'); axis([0 100 0 100 -4 4])
        xlabel('x direction'); ylabel('z direction'); view(3)
        title('Reflected P x comp.')

        subplot(3,4,6)
        surf(x, z, uz_P_SV_r,'EdgeColor','none'); axis([0 100 0 100 -4 4])
        xlabel('x direction'); ylabel('z direction'); view(3)
        title('Reflected P z comp.')

        subplot(3,4,7)
        surf(x, z, ux_SV_SV_r,'EdgeColor','none'); axis([0 100 0 100 -4 4])
        xlabel('x direction'); ylabel('z direction'); view(3)
        title('Reflected SV x comp.')

        subplot(3,4,8)
        surf(x, z, uz_SV_SV_r,'EdgeColor','none'); axis([0 100 0 100 -4 4])
        xlabel('x direction'); ylabel('z direction'); view(3)
        title('Reflected SV z comp.')

        subplot(3,4,9)
        surf(x, z, ux_P_SV_t,'EdgeColor','none'); axis([0 100 0 100 -4 4])
        xlabel('x direction'); ylabel('z direction'); view(3)
        title('Refracted P x comp.')

```

```
            subplot(3,4,10)
            surf(x, z, uz_P_SV_t,'EdgeColor','none'); axis([0 100 0 100 −4 4])
            xlabel('x direction'); ylabel('z direction'); view(3)
            title('Refracted P z comp.')

            subplot(3,4,11)
            surf(x, z, ux_SV_SV_t,'EdgeColor','none'); axis([0 100 0 100 −4 4])
            xlabel('x direction'); ylabel('z direction'); view(3)
            title('Refracted SV x comp.')

            subplot(3,4,12)
            surf(x, z, uz_SV_SV_t,'EdgeColor','none'); axis([0 100 0 100 −4 4])
            xlabel('x direction'); ylabel('z direction'); view(3)
            title('Refracted SV z comp.')

            drawnow
        end

    case {'SH'}
        % angle and amplitude
        th_SH_r = th_SH_i;
        th_SH_t = asind(sind(th_SH_i) * V_S2 / V_S1);

        R_SH = A_SH * (G_1 * V_S2 * cosd(th_SH_i) − G_2 * V_S1 * cosd(th_SH_t))...
            / (G_1 * V_S2 * cosd(th_SH_i) + G_2 * V_S1 * cosd(th_SH_t));
        T_SH = A_SH * (2 * G_1 * V_S2 * cosd(th_SH_i) / (G_1 * V_S2 * cosd(th_SH_i)...
            + G_2 * V_S1 * cosd(th_SH_t)));

        % initialize matrix
        uy_SH_i = zeros(max(size(z)), max(size(x)));
        uy_SH_r = zeros(max(size(z)), max(size(x)));
        uy_SH_t = zeros(max(size(z)), max(size(x)));

        % calculate incident, reflected, and transmitted wave
        for t = t_low : dt : t_up
            for i = 1 : max(size(x))
                for j = 1 : max(size(z))
                    % incident SH
                    uy_SH_i(i, j) = real(A_SH *(exp((1i * omega / V_S1) * (x(i, 1) ...
                        * sind(th_SH_i) + z(j, 1) * cosd(th_SH_i) − V_S1 * t))));
                    % reflected SH
                    uy_SH_r(i, j) = real(R_SH *(exp((1i * omega / V_S1) * (x(i, 1) ...
                        * sind(th_SH_r) − z(j, 1) * cosd(th_SH_r) − V_S1 * t))));
                    % refracted SH
                    uy_SH_t(i, j) = real(T_SH *(exp((1i * omega / V_S2) * (x(i, 1) ...
                        * sind(th_SH_t) + z(j, 1) * cosd(th_SH_t) − V_S2 * t))));
                    % adding incident and reflected SH
                    adding_SH(i,j) = (uy_SH_i(i, j) − uy_SH_r(i, j)) / 2;

                end
            end

             subplot(3,1,1)
             surf(x, z, uy_SH_i,'EdgeColor','none'); axis([0 100 0 100 −4 4])
             xlabel('x direction'); ylabel('z direction'); view(3)
             title('Incident')
             subplot(3,1,2)
             surf(x, z, uy_SH_r,'EdgeColor','none'); axis([0 100 0 100 −4 4])
             xlabel('x direction'); ylabel('z direction'); view(3)
             title('Reflected')
             subplot(3,1,3)
             surf(x, z, uy_SH_t,'EdgeColor','none'); axis([0 100 0 100 −4 4])
             xlabel('x direction'); ylabel('z direction'); view(3)
             title('Refracted')
```

```
% surf(x, z, adding_SH,'EdgeColor','none'); xlabel('x direction'); ylabel('z direction'); view(3)

            drawnow
        end
end
```

705.3 Matlab code – surface wave solution

Listing 705.2: Example MATLAB code for surface waves

```

% =======================================================
% surf.m: inclined wave propagation closed-form exact solution for surface waves
%
% ref. Waves and Vibrations in Soils (Semblat and Pecker, 2009)
%
% Chang-Gyun Jeong
% Last update: 03/27/2011
% =======================================================

clear all; clc;

% coordinates to see waves
x_cord = 0; z_cord = 0;

% time limit for time history
t_low = 0; t_up = 10; dt = 0.01;

% wave frequency and initial time for phase
freq = 10; % natural frequency of wave
t = 0.0; % initial time for phase
omega = freq * 2 * pi; % omega

% incident angle and amplitude
A_P = 1; % incident amp. (P wave)
A_SH = 1; % incident amp. (SH wave)
th_SH_i = 20; % incident angle. (SH wave)

% velocity and shear modulus (r is assumed to 20 kN/m^3)
v = 0.3; % Poisson's ratio
V_S1 = 50; % Vs for layer 1

V_R = V_S1 * (0.862 + 1.14 * v) / (1 + v); % Rayleigh wave velocity
V_P1 = V_S1 * sqrt((2 - 2 * v) / (1 - 2 * v)); % Vp for layer 1

% calculate Rayleigh waves
a = -1*abs(sqrt(omega^2 / V_R^2 - omega^2 / V_P1^2));
b = -1*abs(sqrt(omega^2 / V_R^2 - omega^2 / V_S1^2));

fq_u_x_R = A_P * 1i * omega / V_R * ...
    (exp(a * z_cord) - ((2 * a * b / (b^2 + omega^2 / V_R^2))* exp(b * z_cord))) ...
    * exp((1i * omega / V_R) * (x_cord - V_R * t));
fq_u_z_R = A_P * a * ...
    (exp(a * z_cord) - ((2 * omega^2 / V_R^2) / (b^2 + omega^2 / V_R^2) * exp(b * z_cord))) ...
    * exp((1i * omega / V_R) * (x_cord - V_R * t));

% calculate Love waves
fq_u_y_L = 2 * A_SH * cosd(omega * z_cord * cosd(th_SH_i) / V_S1 ) ...
    * (exp((1i * omega / V_S1) * (x_cord * sind(th_SH_i) - V_S1 * t)));

% calculate phase angle and amplutude of harmonic motion
time_ang_x=angle(fq_u_x_R); % phase angle (R wave, x)
time_abs_x=abs(fq_u_x_R); % amplitude (R wave, x)
time_ang_z=angle(fq_u_z_R); % phase angle (R wave, z)
time_abs_z=abs(fq_u_z_R); % amplitude (R wave, z)
time_ang_y=angle(fq_u_y_L); % phase angle (L wave, y)
time_abs_y=abs(fq_u_y_L); % amplitude angle (L wave, y)

% calculate time history Rayleigh wave
```

```
time = t_low : dt : t_up;

u_x_R = time_abs_x * cos(2*freq*pi*time+time_ang_x) ;
u_z_R = time_abs_z * cos(2*freq*pi*time+time_ang_z) ;

% calculate time history Love wave
u_y_L = time_abs_y * cos(2*freq*pi*time+time_ang_y) ;

% plot
subplot(1,4,1)
plot(u_x_R, u_z_R)
xlabel('x'); ylabel('z');
axis([-4 4 -4 4])

subplot(1,4,2)
plot(u_y_L, u_z_R)
xlabel('y'); ylabel('z');
axis([-4 4 -4 4])

subplot(1,4,3)
plot(u_x_R, u_y_L)
xlabel('x'); ylabel('y');
axis([-4 4 -4 4])

subplot(1,4,4)
plot3(u_x_R, u_y_L, u_z_R)
axis([-4 4 -4 4 -4 4])
xlabel('x'); ylabel('y'); zlabel('z');
```

705.4 Matlab code – Ricker wavelet as an input motion

Listing 705.3: Example MATLAB code for Ricker wavelet as an input motion

```
% ======================================
% Ricker.m: Ricker wavelet propagation on 3D space closed−form exact solution
%
% ref. Waves and Vibrations in Soils (Semblat and Pecker, 2009)
%
% Chang−Gyun Jeong
% Last update: 01. 05. 2012.
% ======================================

clf; clc; clear all;

%% Initial condition
% max frequency, amplitude, and input angle of ricker wavelet
f_max = 1;
amplitude = 0.005;
th = 40;

% amplitude ratio
% ar_PP = 1;
% arR = 1;

% coordinate for extracting waves
xxx = 0;
zzz = 900;

dx = 100;
x_min = 0;
x_max = xxx;

dz = 100;
z_min = 0;
z_max = zzz;

nx = x_max / dx + 1;
nz = z_max / dz + 1;

% Poisson's ratio and wave velocity of ground
v = 0.1;
Vs = 700;
Vp = Vs * sqrt((2 − 2 * v) / (1 − 2 * v));
Vr = Vs * (0.862 + 1.14 * v) / (1 + v);

% time step, peak time, max and min time
dt = 0.01;
t_min = 0;
t_max = 20;
t_peak = 4;

% parameters for plotting
t_min_pl = 0;
t_max_pl = 10;
d_min_pl = −0.02;
d_max_pl = 0.02;

f_min_pl = 0;
f_max_pl = 5;
f_amp_min_pl = 0;
f_amp_max_pl = 0.0006;

```

```
% amplitude and angle calculation for reducing computation time
thrp = asind(sind(th) * Vp / Vs);
thrs = asind(sind(th) * Vs / Vp);

ss = sind(th) / Vs;
cc = cosd(th) / Vs;
ssrp = sind(thrp) / Vp;
ccrp = cosd(thrp) / Vp;
ssp = sind(th) / Vp;
ccp = cosd(th) / Vp;
ssrs = sind(thrs) / Vs;
ccrs = cosd(thrs) / Vs;

as = amplitude * sind(th);
ac = amplitude * cosd(th);

%% Ricker wavelet
k = 1;
for t = t_min:dt:t_max
    j = 1;
    for z = z_min:dz:z_max
        i = 1;
        for x = x_min:dx:x_max
            SV_i_x(i, j) = ac * ...
                ((1 - 2 * pi^2 * f_max^2 * (t - t_peak + x * ss + z * cc)^2) * ...
                exp(-pi^2 * f_max^2 * ...
                (t - t_peak + x * ss + z * cc)^2));
            SV_i_z(i, j) = as * ...
                ((1 - 2 * pi^2 * f_max^2 * (t - t_peak + x * ss + z * cc)^2) * ...
                exp(-pi^2 * f_max^2 * ...
                (t - t_peak + x * ss + z * cc)^2));

            SVP_r_x(i, j) = as * sind(thrp) * ...
                ((1 - 2 * pi^2 * f_max^2 * (t - t_peak + x * ssrp - z * ccrp)^2) * ...
                exp(-pi^2 * f_max^2 * ...
                (t - t_peak + x * ssrp - z * ccrp)^2));
            SVP_r_z(i, j) = as * cos(thrp) *...
                ((1 - 2 * pi^2 * f_max^2 * (t - t_peak + x * ssrp + z * ccrp)^2) * ...
                exp(-pi^2 * f_max^2 * ...
                (t - t_peak + x * ssrp + z * ccrp)^2));

            SVSV_r_x(i, j) = ac * cosd(th) * ...
                ((1 - 2 * pi^2 * f_max^2 * (t - t_peak + x * ss - z * cc)^2) * ...
                exp(-pi^2 * f_max^2 * ...
                (t - t_peak + x * ss - z * cc)^2));
            SVSV_r_z(i, j) = as * sind(th) * ...
                ((1 - 2 * pi^2 * f_max^2 * (t - t_peak + x * ss - z * cc)^2) * ...
                exp(-pi^2 * f_max^2 * ...
                (t - t_peak + x * ss - z * cc)^2));

            P_i_x(i, j) = ac * ...
                ((1 - 2 * pi^2 * f_max^2 * (t + 1.35 - t_peak + x * ssp + z * ccp)^2) * ...
                exp(-pi^2 * f_max^2 * ...
                (t + 1.35 - t_peak + x * ssp + z * ccp)^2));
            P_i_z(i, j) = as * ...
                ((1 - 2 * pi^2 * f_max^2 * (t + 1.35 - t_peak + x * ssp + z * ccp)^2) * ...
                exp(-pi^2 * f_max^2 * ...
                (t + 1.35 - t_peak + x * ssp + z * ccp)^2));

            PP_r_x(i, j) = as * sind(th) * ...
                ((1 - 2 * pi^2 * f_max^2 * (t + 1.35 - t_peak + x * ssp - z * ccp)^2) * ...
                exp(-pi^2 * f_max^2 * ...
                (t + 1.35 - t_peak + x * ssp - z * ccp)^2));
            PP_r_z(i, j) = as * cos(th) *...
```

```
                ((1 - 2 * pi^2 * f_max^2 * (t + 1.35 - t_peak + x * ssp + z * ccp)^2) * ...
                exp(-pi^2 * f_max^2 * ...
                (t + 1.35 - t_peak + x * ssp + z * ccp)^2));

            PSV_r_x(i, j) = ac * cosd(thrs) * ...
                ((1 - 2 * pi^2 * f_max^2 * (t + 1.35 - t_peak + x * ssrs - z * ccrs)^2) * ...
                exp(-pi^2 * f_max^2 * ...
                (t + 1.35 - t_peak + x * ssrs - z * ccrs)^2));
            PSV_r_z(i, j) = as * sind(thrs) * ...
                ((1 - 2 * pi^2 * f_max^2 * (t + 1.35 - t_peak + x * ssrs - z * ccrs)^2) * ...
                exp(-pi^2 * f_max^2 * ...
                (t +1.35 - t_peak + x * ssrs - z * ccrs)^2));


            total_x(i, j) = SV_i_x(i, j) + SVP_r_x(i, j) + SVSV_r_x(i, j) + ...
                            P_i_x(i, j) + PP_r_x(i, j) + PSV_r_x(i, j);
            total_z(i, j) = SV_i_z(i, j) + SVP_r_z(i, j) + SVSV_r_z(i, j) + ...
                            P_i_z(i, j) + PP_r_z(i, j) + PSV_r_z(i, j);

            i = i + 1;
        end
        j = j + 1;
    end

    disp_1(k) = SV_i_x(nx, nz) + P_i_x(nx, nz);
    disp_2(k) = SV_i_z(nx, nz) + P_i_z(nx, nz);
    disp_3(k) = SVP_r_x(nx, nz) + PP_r_x(nx, nz);
    disp_4(k) = SVP_r_z(nx, nz) + PP_r_z(nx, nz);
    disp_5(k) = SVSV_r_x(nx, nz) + PSV_r_x(nx, nz);
    disp_6(k) = SVSV_r_z(nx, nz) + PSV_r_z(nx, nz);

    time_hist(k) = t;

    k = k + 1;
end

% Fourier transform
Fs = 1 / dt;
NFFT = 2^nextpow2(((t_max - t_min) / dt));
F_1 = fft(disp_1, NFFT) / ((t_max - t_min) / dt);
F_2 = fft(disp_2, NFFT) / ((t_max - t_min) / dt);
F_3 = fft(disp_3, NFFT) / ((t_max - t_min) / dt);
F_4 = fft(disp_4, NFFT) / ((t_max - t_min) / dt);
F_5 = fft(disp_5, NFFT) / ((t_max - t_min) / dt);
F_6 = fft(disp_6, NFFT) / ((t_max - t_min) / dt);
F_7 = fft(disp_1 + disp_3 + disp_5, NFFT) / ((t_max - t_min) / dt);
F_8 = fft(disp_2 + disp_4 + disp_6, NFFT) / ((t_max - t_min) / dt);
fq = (Fs / 2) * linspace(0, 1, NFFT / 2 + 1);

%% Rayleigh wave
z = 0;
x = 0;
t = 0;

omega = fq * 2 * pi();
Ax=2 * abs(F_1(1:NFFT/2+1));
Az=2 * abs(F_2(1:NFFT/2+1));

omega(1) = 0.000001;

for i = 1 : max(size(fq))
    a = 1*abs(sqrt(omega(i)^2 / Vr^2 - omega(i)^2 / Vp^2));
    b = 1*abs(sqrt(omega(i)^2 / Vr^2 - omega(i)^2 / Vs^2));

```

```
    surf_x(i) = (1i * omega(i) * Ax(i) / Vr) * (exp(a * z) - (2 * a * b * exp(b * z)) / ...
        (b^2 + omega(i)^2 / Vr^2)) * exp(1i * omega(i) * (x - Vr * t) / Vr);
    surf_z(i) = a * Az(i) * (exp(a * z) - (2 * omega(i)^2 * exp(b * z) / Vr^2) / ...
        (b^2 + omega(i)^2 / Vr^2)) * exp(1i * omega(i) * (x - Vr * t) / Vr);
end

% calculate phase angle and amplitude of R wave
time_ang_x=angle(surf_x);
time_abs_x=abs(surf_x);
time_ang_z=angle(surf_z);
time_abs_z=abs(surf_z);

for i = 1:max(size(fq))
    j = 1;
    for time = t_min:dt:t_max;
        u_x_R(i, j) = time_abs_x(i) * cos(omega(i) * (time - t_peak) - time_ang_x(i));
        u_z_R(i, j) = time_abs_z(i) * cos(omega(i) * (time - t_peak) - time_ang_z(i));
        j = j + 1;
    end
    i = i + 1;
end

u_x = sum(u_x_R);
u_z = sum(u_z_R);

% get max amplitude of R wave to scale it
max_R_x = max(u_x);
max_R_z = max(u_z);

% R wave will be recalculated on the point of interest
z = xxx;
x = zzz;

for i = 1 : max(size(fq))
    a = 1*abs(sqrt(omega(i)^2 / Vr^2 - omega(i)^2 / Vp^2));
    b = 1*abs(sqrt(omega(i)^2 / Vr^2 - omega(i)^2 / Vs^2));

    surf_x(i) = (1i * omega(i) * Ax(i) / Vr) * (exp(a * z) - (2 * a * b * exp(b * z)) / ...
        (b^2 + omega(i)^2 / Vr^2)) * exp(1i * omega(i) * (x - Vr * t) / Vr);
    surf_z(i) = a * Az(i) * (exp(a * z) - (2 * omega(i)^2 * exp(b * z) / Vr^2) / ...
        (b^2 + omega(i)^2 / Vr^2)) * exp(1i * omega(i) * (x - Vr * t) / Vr);
end

% calculate phase angle and amplitude of R wave
time_ang_x=angle(surf_x);
time_abs_x=abs(surf_x);
time_ang_z=angle(surf_z);
time_abs_z=abs(surf_z);

for i = 1:max(size(fq))
    j = 1;
    for time = t_min:dt:t_max;
        u_x_R(i, j) = time_abs_x(i) * cos(omega(i) * (time - t_peak) - time_ang_x(i));
        u_z_R(i, j) = time_abs_z(i) * cos(omega(i) * (time - t_peak) - time_ang_z(i));
        j = j + 1;
    end
    i = i + 1;
end

u_x = sum(u_x_R);
u_z = sum(u_z_R);

% scale R wave
u_x = u_x * amplitude / max_R_x;
```

```
u_z = u_z * amplitude / max_R_x;

% set amplitude of R wave as zero
% u_x = 0;
% u_z = 0;

% add all displacement (R + input + reflect)
u_xx = u_x + disp_1 + disp_3 + disp_5;
u_zz = u_z + disp_2 + disp_4 + disp_6;

% Fourier transform
F_9 = fft(u_x, NFFT) / ((t_max - t_min) / dt);
F_10 = fft(u_z, NFFT) / ((t_max - t_min) / dt);
F_11 = fft(u_xx, NFFT) / ((t_max - t_min) / dt);
F_12 = fft(u_zz, NFFT) / ((t_max - t_min) / dt);

% make final output matrix
disp_out(:, 1) = time_hist';
disp_out(:, 2) = u_xx';
disp_out(:, 4) = u_zz';

%% plot

% figure 1: all components
figure(1)
subplot(6, 2, 1)
plot(time_hist, disp_1, time_hist, disp_2)
ylim([d_min_pl d_max_pl])
xlim([t_min_pl t_max_pl])
xlabel('Time (s)')
ylabel('Displacement (m)')
title('Input SV + P')

subplot(6, 2, 3)
plot(time_hist, disp_3, time_hist, disp_4)
ylim([d_min_pl d_max_pl])
xlim([t_min_pl t_max_pl])
xlabel('Time (s)')
ylabel('Displacement (m)')
title('Reflected P')

subplot(6, 2, 5)
plot(time_hist, disp_5, time_hist, disp_6)
ylim([d_min_pl d_max_pl])
xlim([t_min_pl t_max_pl])
xlabel('Time (s)')
ylabel('Displacement (m)')
title('Reflected SV')

subplot(6, 2, 7)
plot(time_hist, disp_1 + disp_3 + disp_5, time_hist, disp_2 + disp_4 + disp_6)
ylim([d_min_pl d_max_pl])
xlim([t_min_pl t_max_pl])
xlabel('Time (s)')
ylabel('Displacement (m)')
title('Body total')

subplot(6,2,9)
plot(time_hist, u_x, time_hist, u_z)
ylim([d_min_pl d_max_pl])
xlim([t_min_pl t_max_pl])
xlabel('Time (s)')
ylabel('Displacement (m)')
title('Surf. Rayleigh')
```

```

subplot(6,2,11)
plot(time_hist, u_xx, time_hist, u_zz)
ylim([d_min_pl d_max_pl])
xlim([t_min_pl t_max_pl])
xlabel('Time (s)')
ylabel('Displacement (m)')
title('Body and Surface Total')

subplot(6,2,2)
plot(fq, 2 * abs(F_1(1:NFFT/2+1)), fq, 2 * abs(F_2(1:NFFT/2+1)))
ylim([f_amp_min_pl f_amp_max_pl])
xlim([f_min_pl f_max_pl])
xlabel('Frequency (Hz)')
ylabel('Fourier Amplitude')
title('Input SV + P')

subplot(6,2,4)
plot(fq, 2 * abs(F_3(1:NFFT/2+1)), fq, 2 * abs(F_4(1:NFFT/2+1)))
ylim([f_amp_min_pl f_amp_max_pl])
xlim([f_min_pl f_max_pl])
xlabel('Frequency (Hz)')
ylabel('Fourier Amplitude')
title('Reflected P')

subplot(6,2,6)
plot(fq, 2 * abs(F_5(1:NFFT/2+1)), fq, 2 * abs(F_6(1:NFFT/2+1)))
ylim([f_amp_min_pl f_amp_max_pl])
xlim([f_min_pl f_max_pl])
xlabel('Frequency (Hz)')
ylabel('Fourier Amplitude')
title('Reflected SV')

subplot(6,2,8)
plot(fq, 2 * abs(F_7(1:NFFT/2+1)), fq, 2 * abs(F_8(1:NFFT/2+1)))
ylim([f_amp_min_pl f_amp_max_pl])
xlim([f_min_pl f_max_pl])
xlabel('Frequency (Hz)')
ylabel('Fourier Amplitude')
title('Body Total')

subplot(6,2,10)
plot(fq, 2 * abs(F_9(1:NFFT/2+1)), fq, 2 * abs(F_10(1:NFFT/2+1)))
ylim([f_amp_min_pl f_amp_max_pl])
xlim([f_min_pl f_max_pl])
xlabel('Frequency (Hz)')
ylabel('Fourier Amplitude')
title('Surf. Rayleigh')

subplot(6,2,12)
plot(fq, 2 * abs(F_11(1:NFFT/2+1)), fq, 2 * abs(F_12(1:NFFT/2+1)))
ylim([f_amp_min_pl f_amp_max_pl])
xlim([f_min_pl f_max_pl])
xlabel('Frequency (Hz)')
ylabel('Fourier Amplitude')
title('Body and Surface Total')
```

705.5 Wave Potential Formulation – Domain Reduction Method

Methodology presented here is from Wang et al. (2021).

Presented is a methodology developed to investigate influence of inclined body and surface seismic wave on linear or nonlinear earthquake soil structure interaction (ESSI) behavior of soil-structure systems. Methodology is based on Wave Potential Formulation (WPF) (Thomson, 1950; Haskell, 1953) as well as Domain Reduction Method (DRM) (Bielak et al., 2003a).

Presented WPF-DRM methodology consists of three main steps:

1. Analytic development of free field ground motions for a layered half space, excited by an incident, inclined plane wave. Development of this seismic wave field is relying on wave potential formulation in frequency-wave number domain. Time domain spatially varying ground motions are then synthesized through inverse Fourier transformation.
2. Development of the Effective Earthquake Forces, from DRM formulation, is then performed using free field seismic motions developed in the previous step.
3. Earthquake Soil Structure Interaction (ESSI) analysis of the soil-structure system is then performed using effective earthquake forces that are applied to a single layer of finite elements surrounding soil-structure system, so called DRM layer. The only waves that are radiated from the soil-structure system and exit the DRM layer are due to oscillations of the structure. These outgoing waves are absorbed by damping layers.

Sections ?? and ?? below provide details of Wave Potential Formulation and Domain Reduction Method, respectively.

Appendix 706

Body and Surface Wave Numerical Modeling

(2010-2012-2018-2019-2021-)

(In collaboration with Dr. Nima Tafazzoli, Mr. Chang-Gyun Jeong and Dr. Hexiang Wang)

706.1 Integral equations

Second method to generate wave fields is frequency-wavenumber integration method. The fk package is a frequency-wavenumber integration (Haskell, 1964; Wang and Herrmann, 1980) code developed by Zhu and Rivera (2002). The fk code package can be downloaded from `http://www.eas.slu.edu/People/LZhu`. In this section, using fk code package, wave propagation is simulated. The main modules of the fk code package and its function are shown as followings.

1. fk – compute Green's functions
2. syn – compute synthetic seismogram
3. fk.pl – PERL script to simplify the use of fk

First of all, Green's function is computed by fk with ground model properties (model layer dimension, shear and P wave velocity, density, Q and so on) and source, station (receiver) properties (source depth, epicentral distance, wave propagation direction, and receiver depth and so on). Then, Using syn with calculated Green's function, seismograph is synthesized with given variables such as magnitude, fault strike/dip/rake, and station azimuth. Synthesized seismograph (by syn) are stored as binary / Seismic Analysis Code (SAC) form (Goldstein and Snoke, 2005). Thus, in addition to fk package, Python code to run fk package iteratively, convert results from binary to ASCII text, and make plot is developed (plot.py). All necessary variables to run fk and syn can be adjusted in plot.py. Variables are explained in next sections and can also be found in source codes (fk.f, syn,c, fk.pl, and plot.py)

706.1.1 fk3.0 package

In this section, fk3.0 package is briefly introduced. All source code can be downloaded from `http://www.eas.slu.edu/People/LZhu`.

706.1.1.1 fk and 'sample_input'

Program fk can be run by using input text file or using PERL wrapper fk.pl as shown in next section. Example input file 'sample_input' is used to run fk. 'sample_input' is included in fk package. Total layer number, source layer number, source type, and receiver layer number are defined in the first row of input file. Layer properties are defined from second to fourth row (layer thickness / Vp / Vs / density / Qp / Qs values are defined from the first column to sixth column, respectively). Then, sigma, number of sampling points, sampling interval, tapering factor, high / low pass filter frequencies, slowness limit, epicentral distances are defined. In example, variables can be set as following.

Line1: 3 2 2 1 0

3 – total number of layers

2 – source is located at the top of 2nd layer

2 – double coupled source (0: explosion, 1: single coupled)

1 – receiver is located at the top of 1st layer

0 – consider both up and down going wave (1: down going wave, -1: up going wave)

Line2: 10.0000 6.3000 3.5000 2.7860 1000.00 500.00

10.0 – depth (km)

6.3 – Vp (km/s)

3.5 – Vs (km/s)

2.7860 – density (g/cm^3)

1000 – Qp

500 – Qs

Line3: 25.0000 6.3000 3.5000 2.7860 1000.00 500.00

ref. Line2

Line4: 0.0000 8.1000 4.7000 3.3620 1600.00 800.00

ref. Line2

Line5: 2 512 0.2 0.5 25 2 1 1

2 – sigma in 1/trace length, the small imaginary frequency (2 – 3)

512 – number of points in the time domain

0.2 – time step (sec, dt)

0.5 – tapering factor to suppress high frequencies

25 – number of points to be saved before t0

2 – smooth factor to increase the sampling rate

1 1 – high pass filter (wc1, wc2)

Line6: 0. 1 0.3 15

0. 1 – minimum and maximum slowness, to sepcify the window for the wavenumber integration (pmin, pmax)

0.3 – wavenumber sampling interval

15 – the maximum wavenumber at zero frequency

Line7: 1

1 – number of distance ranges

Line8: 200.00 20.00 200.grn.

200.00 – distance (km)

20.00 – t0

200.grn. – output file name

706.1.1.2 fk.pl

fk also can be run by PERL wrapper. Using the PERL wrapper fk.pl is strongly recommended by Zhu (README file of fk package). Since most of variables those are introduced on prior section are set to default values and also can be adjusted on the command line, it is much easier to use fk.pl than run fk.

706.1.1.3 syn

Using syn with calculated Green's function, seismogram is computed. Direction of fault (strike / dip / rake) and recording station (azimuth), magnitude have to be defined to run syn.

706.1.1.4 plot.py

As mentioned above, output seismogram is in SAC form (binary). PERL script plot.py is coded to convert binary to ASCII, run fk / syn repetitively, perform Fourier transform, and plot figures. Necessary options to run fk and syn can be adjusted in plot.py.

706.1.2 3D seismic wave field generation using integral equation

Using fk package, 3D seismic wave field is generated. Analysis for the most common fault mechanism (strike-slip, dip-slip, and normal fault) are presented. A real earthquake example (Northridge) is also given toward the end of the section.

706.1.2.1 Case 1: strike-slip fault / single layer ground

The first example is strike slip fault case. Model is as shown in Figure 706.1. Ground, fault, and wave properties are shown as below.

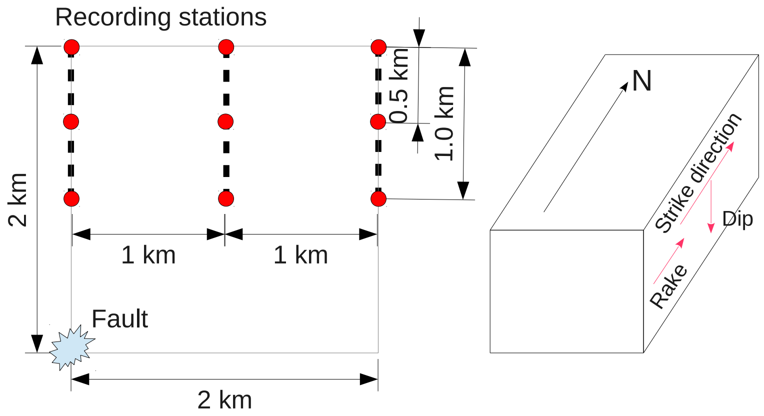


Figure 706.1: Ground and fault model used for analysis, results are captured on circles

- Ground properties
  - $V_S = 1$ km/s
  - $V_P/V_S = 1.73$
  - Poisson's ratio = 0.25
  - Density = 1.32 g/cm^3
  - Shear modulus = 1.32 GPa
  - Elastic modulus = 3.31 GPa
- Fault properties
  - Moment magnitude = 3.5
  - Strike = 0°
  - Dip = 90°
  - Rake = 0°
  - Double - coupled source
  - Triangular source time function
- Wave properties
  - dt = 0.1 s (Max available freq. = 5 Hz, Nyquist freq.)

As indicated above, single layer ground (Vs = 1 km/s) is modeled. Fault is located at 2 km depth, 2 km away from the recording points (stations). Double coupled fault source is assumed and triangular source time function is used (Aki and Richards, 2002). Nine recording points are set as recording stations (Figure 706.1). Direction of the fault is aligned parallel to the north (strike = 0°) and recording station azimuth is set to 0°, 45°, and 90°.

Figure 706.2 – 706.10 show analyses results for the example. Legends on figures mean 'component (epicentral distance, receiver depth)'. EW, NS, and UD components mean East - West, North - South, and Up - Down, respectively (those terms are used for all seismograms, hereafter).

Only EW components are predicted on the stations at 0° azimuth (Figure 706.2 – 706.4) and NS components are showed up on the station at 90° azimuth (Figure 706.8 – 706.10). On the station at 45° azimuth, all components (EW, NS, and UD) are observed.

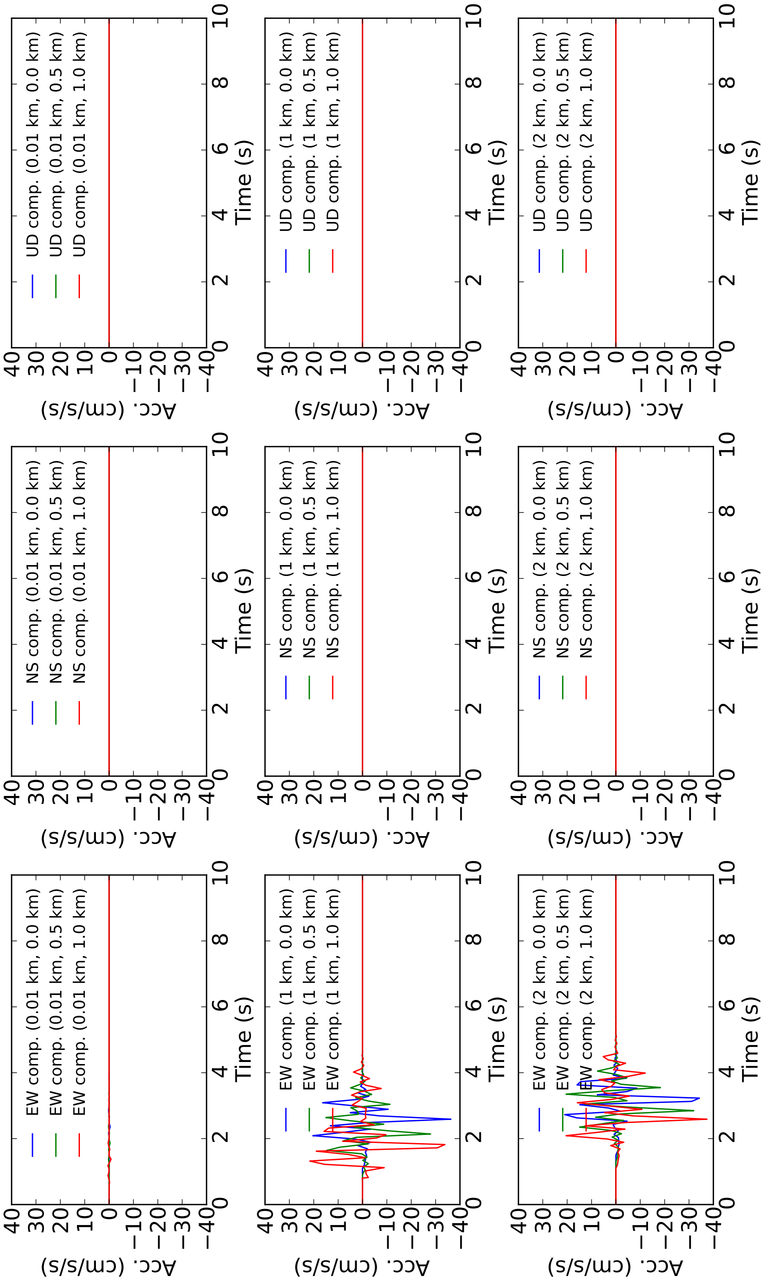


Figure 706.2: Calculated time history acceleration, station azimuth $= 0^{\circ}$

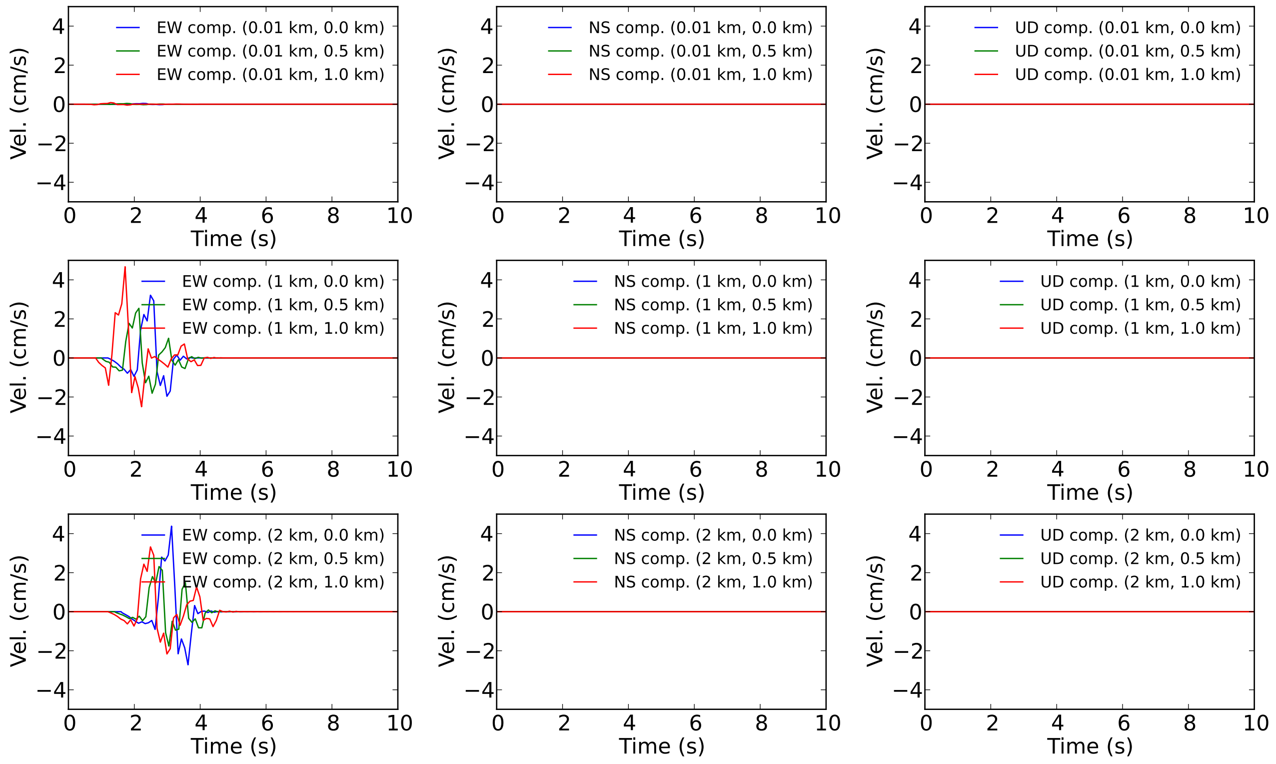


Figure 706.3: Calculated time history velocity, station azimuth $= 0^{\circ}$

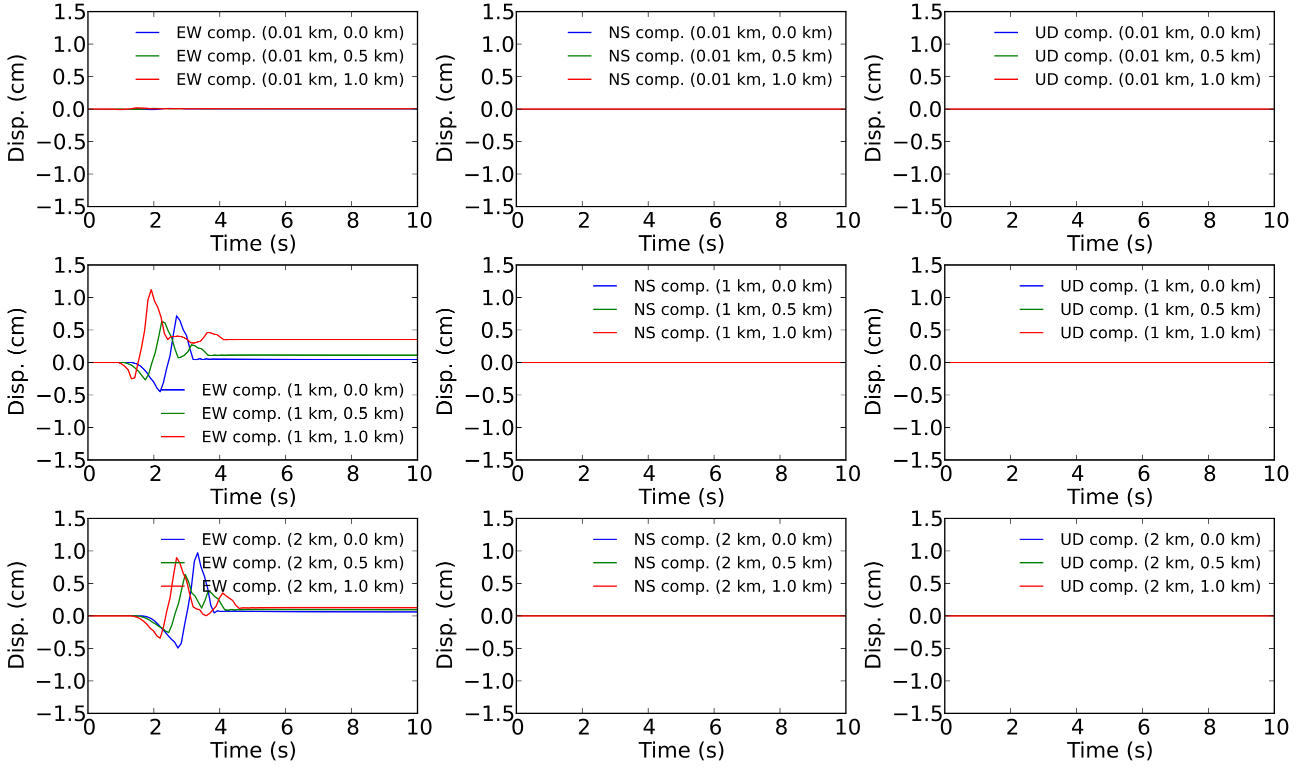


Figure 706.4: Calculated time history displacement, station azimuth $= 0^\circ$

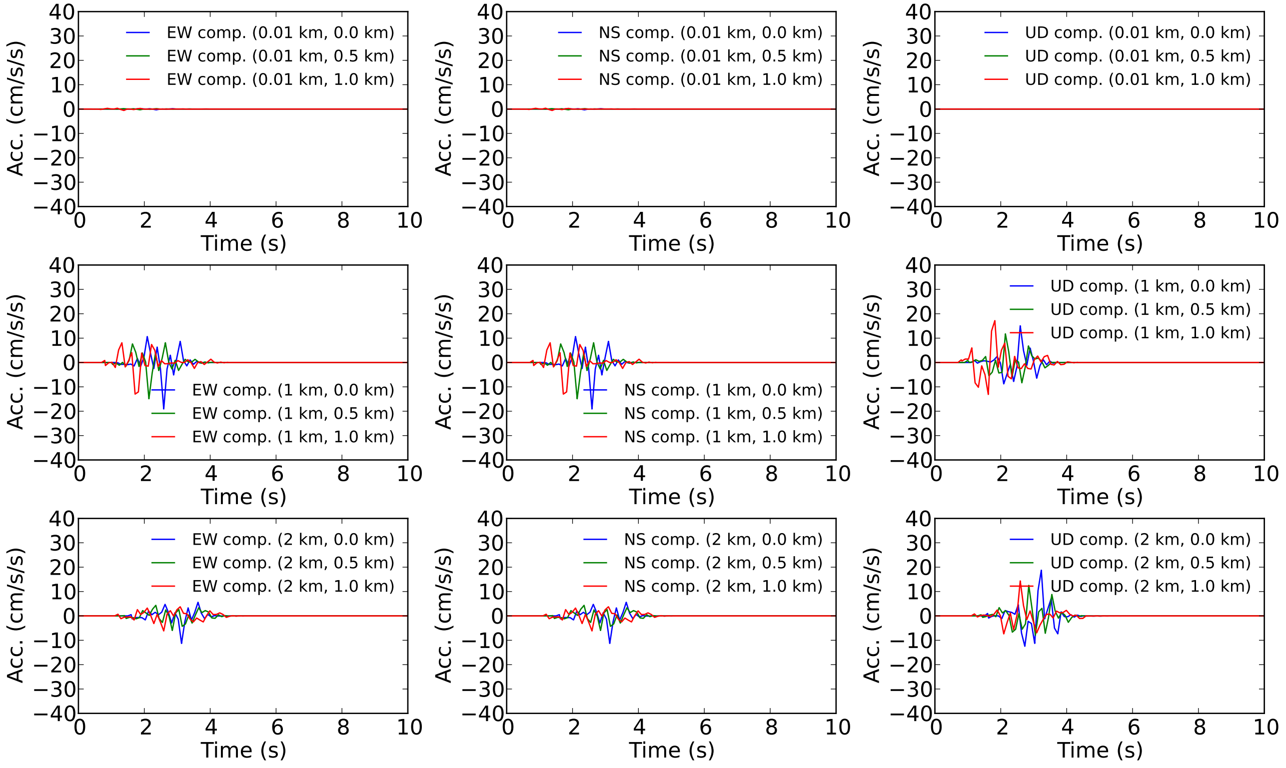


Figure 706.5: Calculated time history acceleration, station azimuth $= 45^{\circ}$

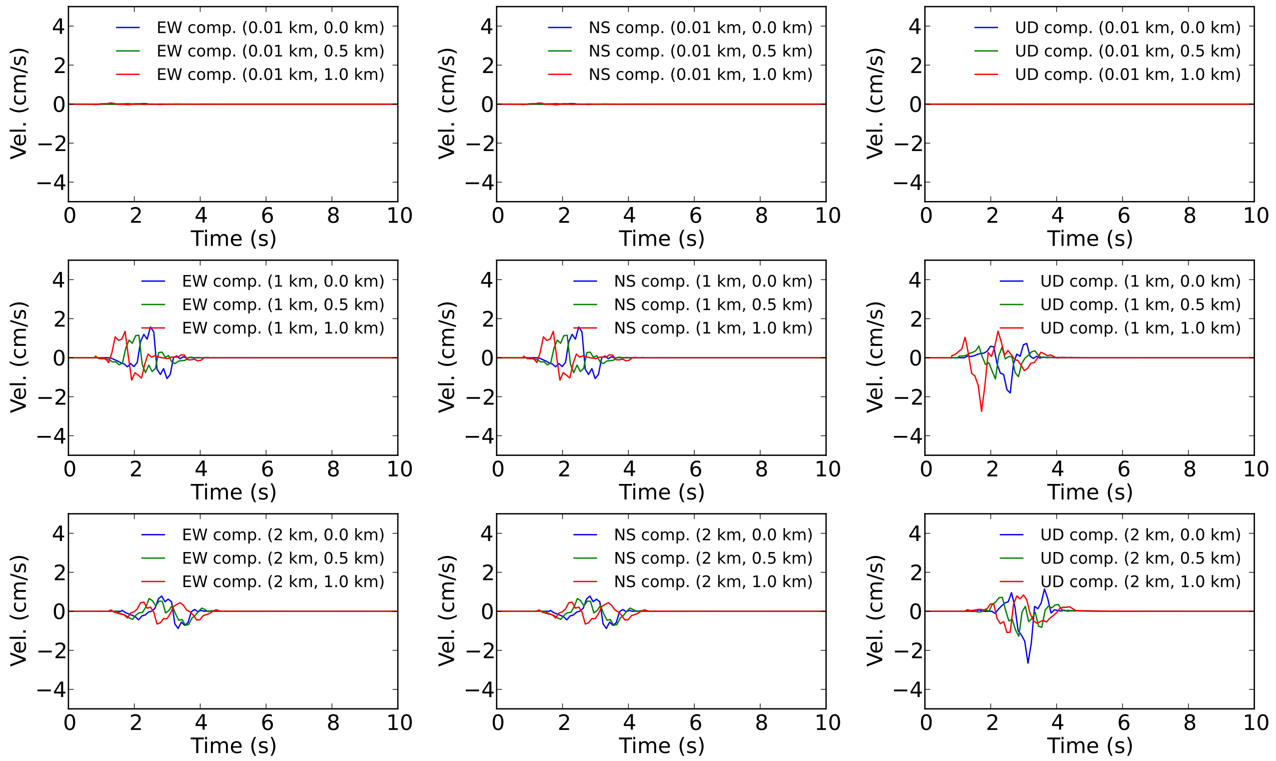


Figure 706.6: Calculated time history velocity, station azimuth = 45°

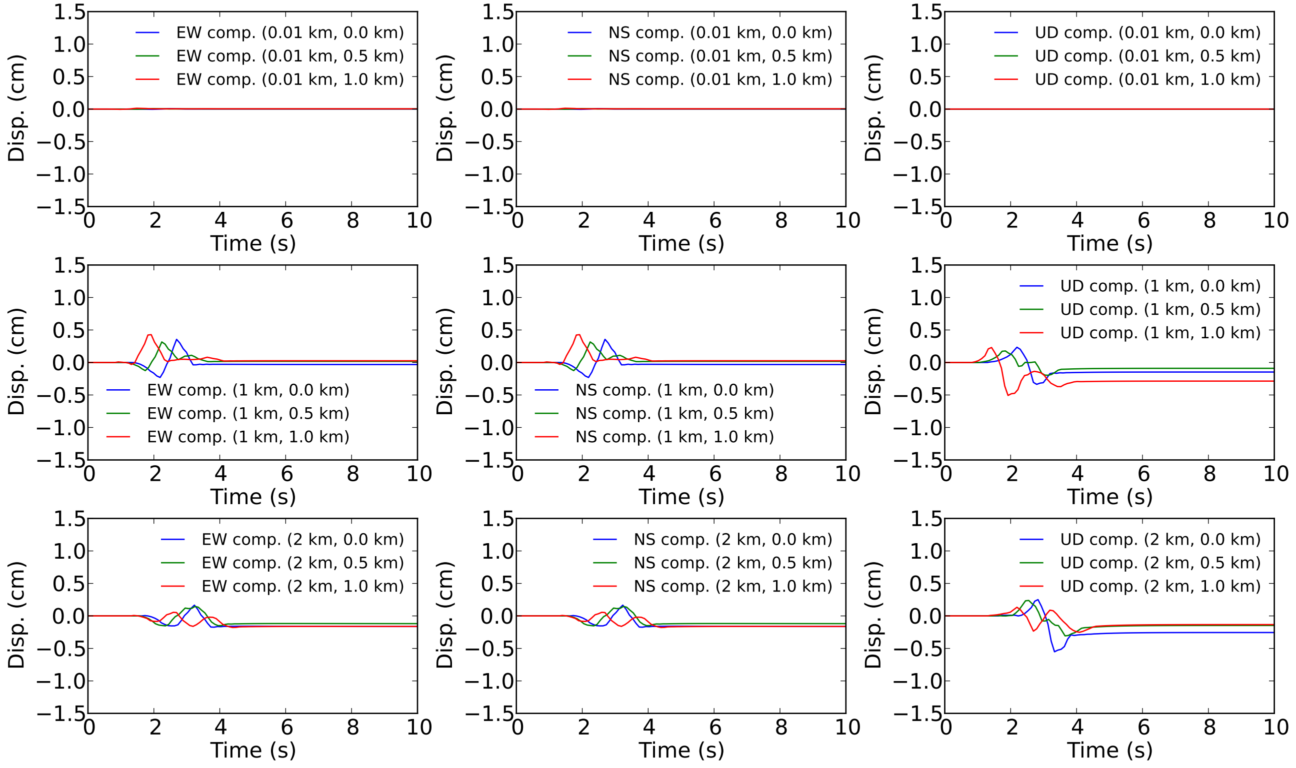


Figure 706.7: Calculated time history displacement, station azimuth = 45°

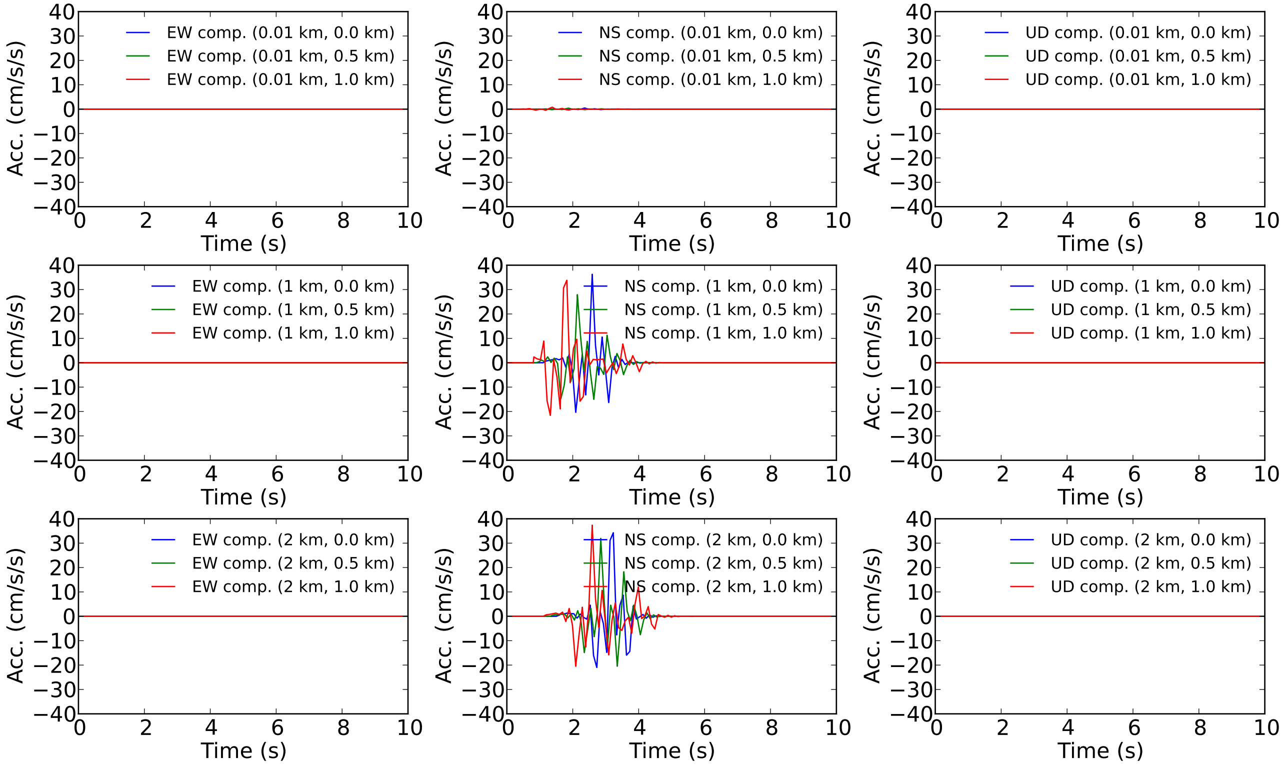


Figure 706.8: Calculated time history acceleration, station azimuth = 90°

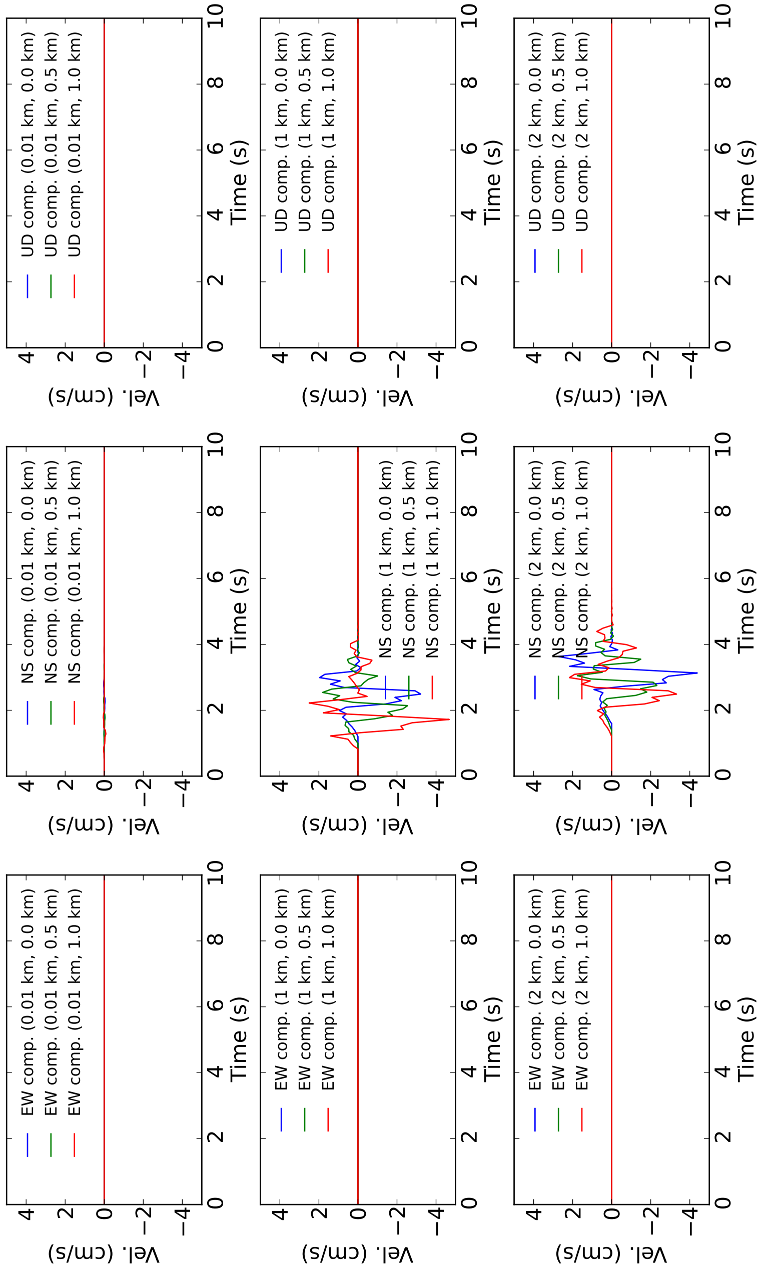


Figure 706.9: Calculated time history velocity, station azimuth = 90°

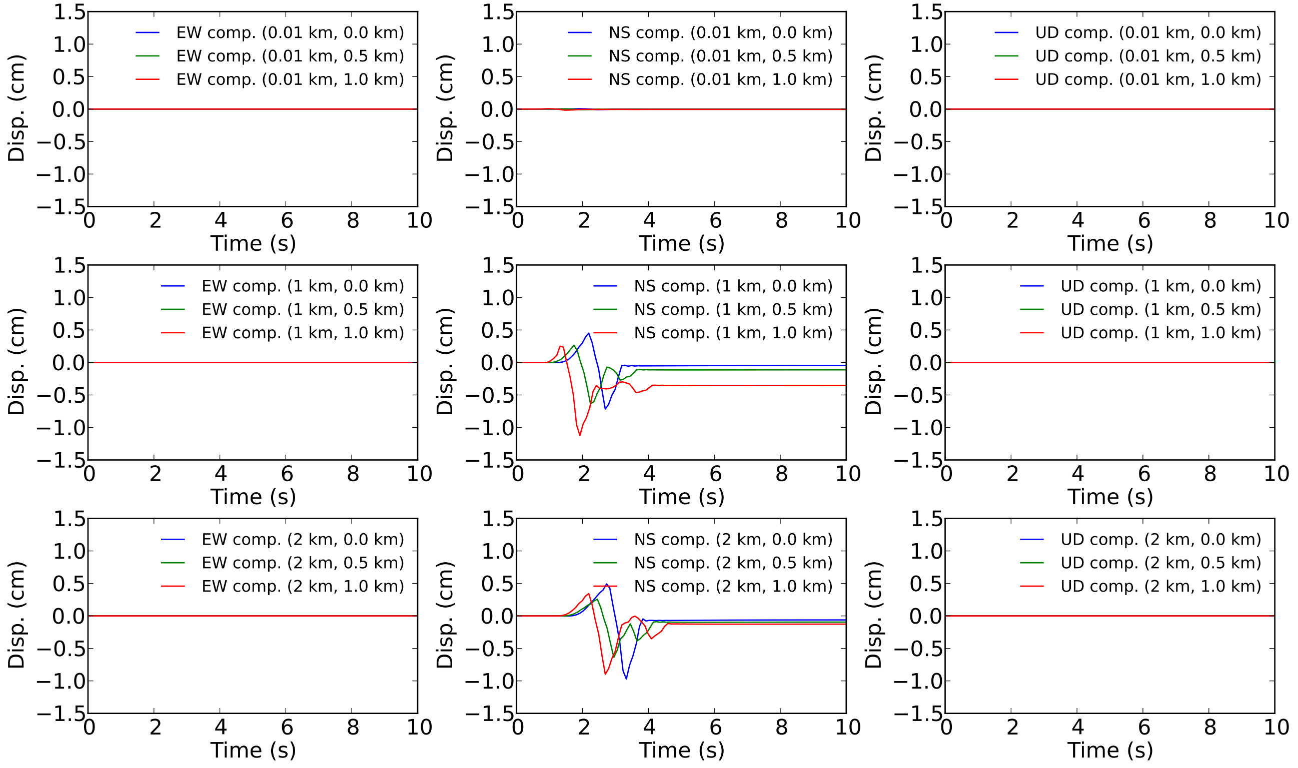


Figure 706.10: Calculated time history displacement, station azimuth = 90°

706.1.2.2 Case 2: dip-slip fault / single layer ground

Similar fault is tested for vertical (dip) slip (rake = 90°) fault case. Figure 706.11 shows model used for the analysis. Ground, fault, and wave properties are shown as below.

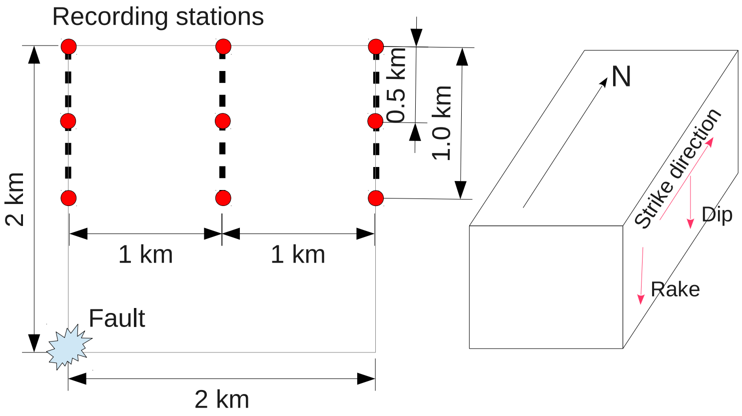


Figure 706.11: Ground and fault model used for analysis, results are captured on circles

- Ground properties
  - V_S = 1 km/s
  - V_P/V_S = 1.73
  - Poisson's ratio = 0.25
  - Density = 1.32 g/cm^3
  - Shear modulus = 1.32 GPa
  - Elastic modulus = 3.31 GPa
- Fault properties
  - Moment magnitude = 3.5
  - Strike = 0°
  - Dip = 90°
  - Rake = 90°
  - Double - coupled source
  - Triangular source time function
- Wave properties

- dt = 0.1 s (Max available freq. = 5 Hz, Nyquist freq.)

Similar to the strike slip example, single layer ground (Vs = 1 km/s) is modeled. Fault is located at 2 km depth, 2 km away from the recording stations. Double coupled fault source is assumed and triangular source time function is used (Aki and Richards, 2002). Nine recording points are set as recording stations (Figure 706.11). Direction of the fault is aligned parallel to the north (strike = 0°) and rake is 90°. Station azimuth is set to 0°, 45°, and 90°.

Figure 706.12 – 706.20 show analyses results for this example. Since it's dip slip case, permanent deformation on UD components are observed (Figure 706.17 and 706.20).

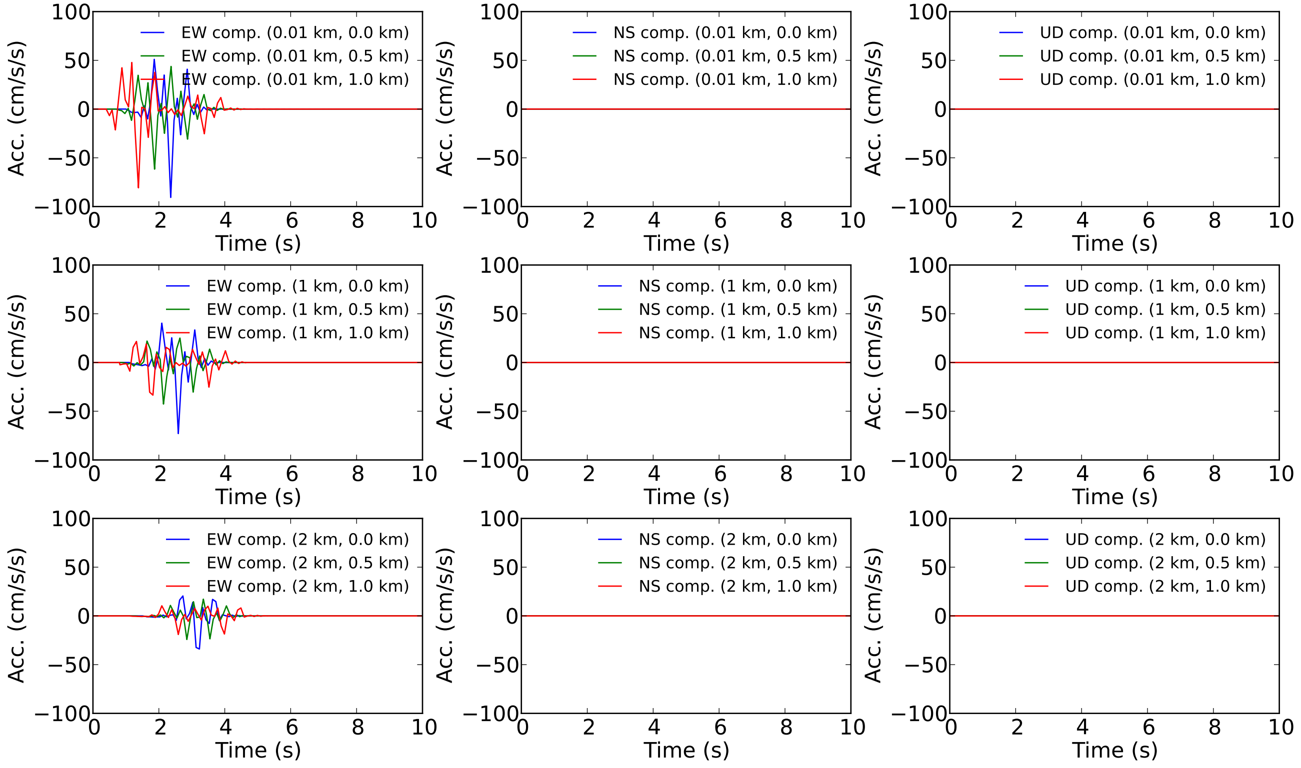


Figure 706.12: Calculated time history acceleration, station azimuth $= 0^\circ$

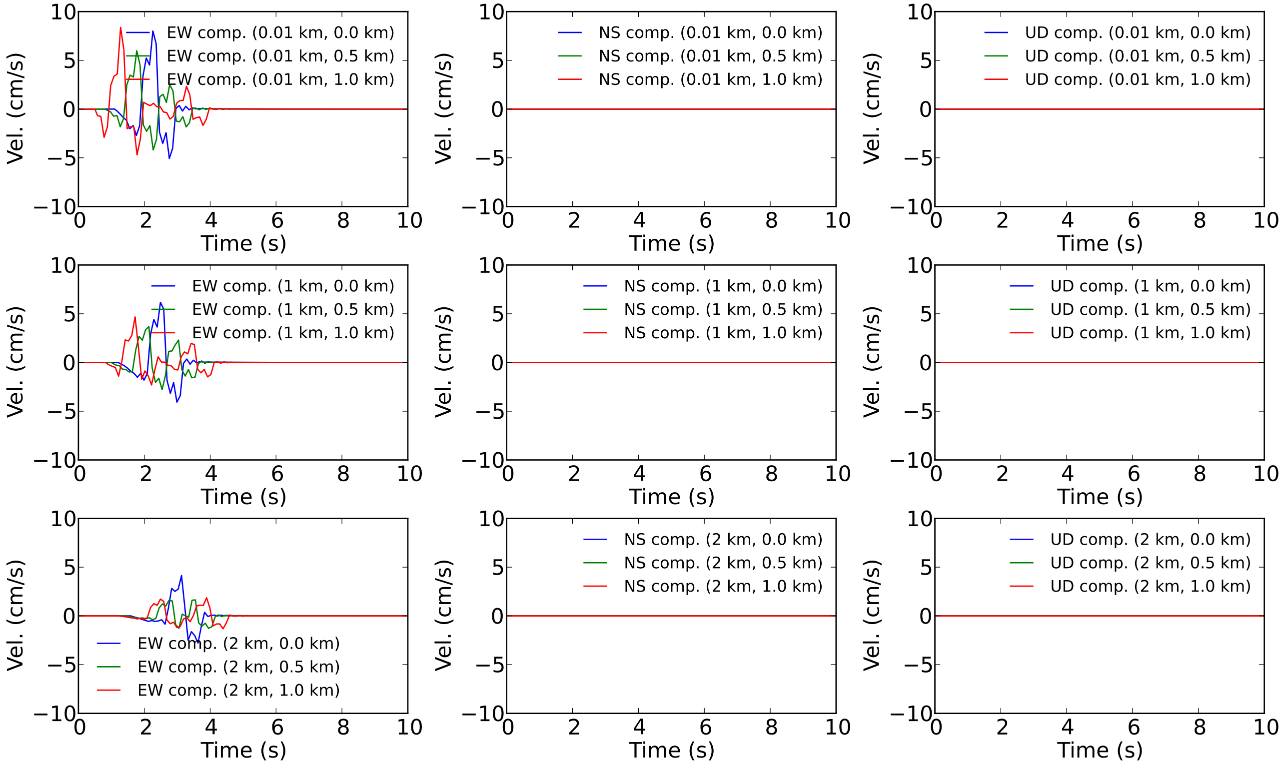


Figure 706.13: Calculated time history velocity, station azimuth $= 0^\circ$

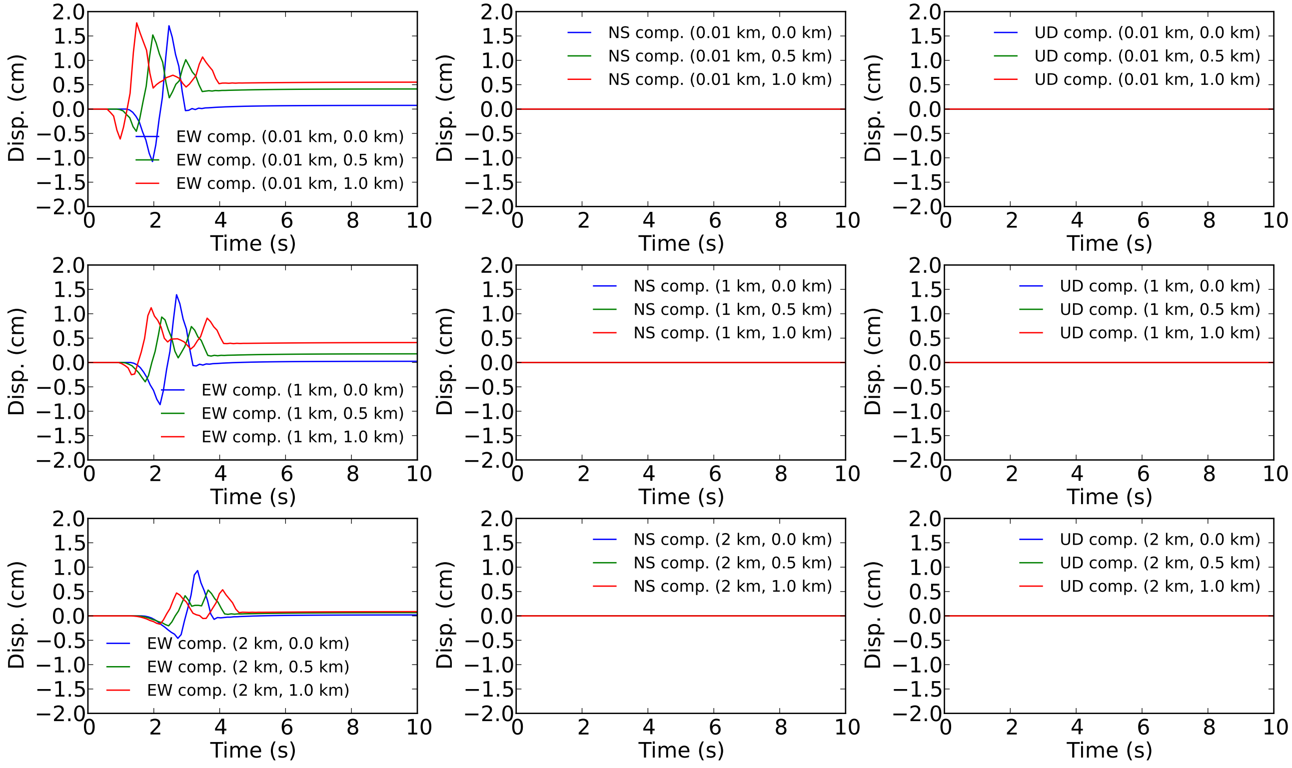


Figure 706.14: Calculated time history displacement, station azimuth = 0°

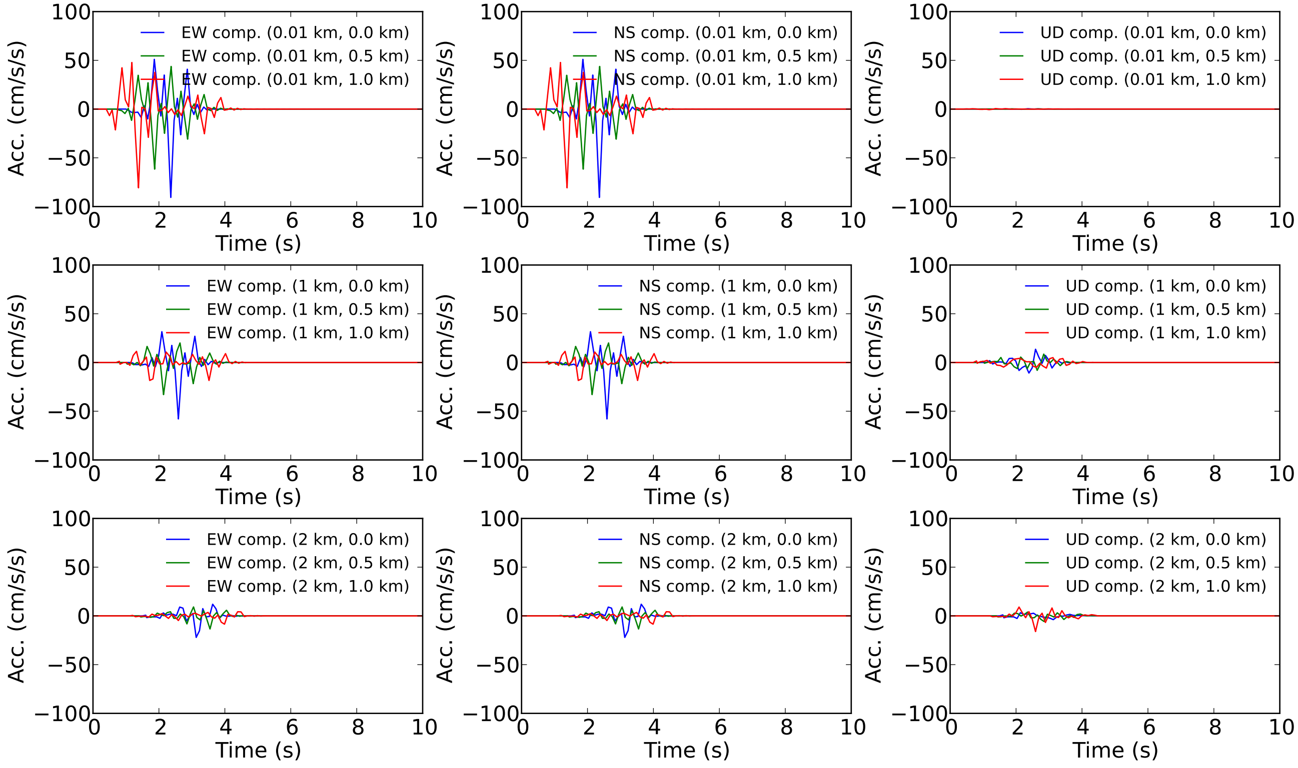


Figure 706.15: Calculated time history acceleration, station azimuth = 45°

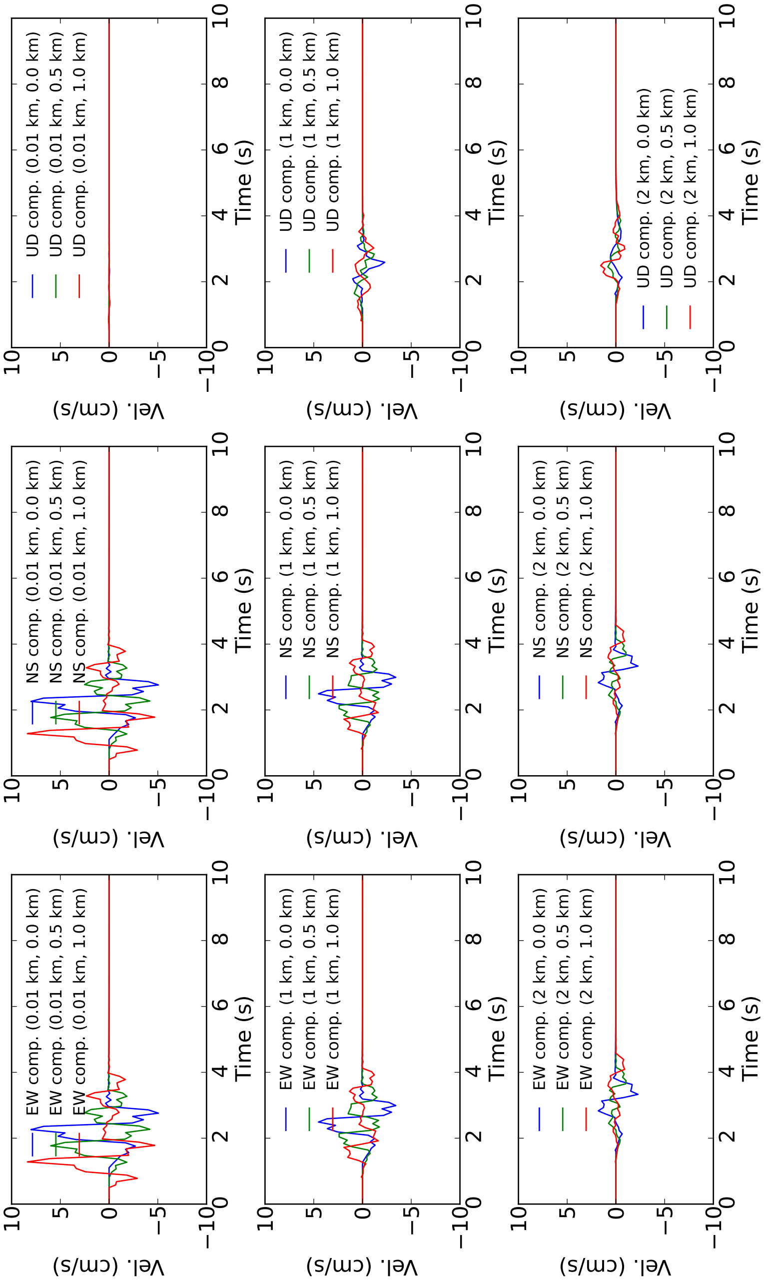


Figure 706.16: Calculated time history velocity, station azimuth $= 45^\circ$

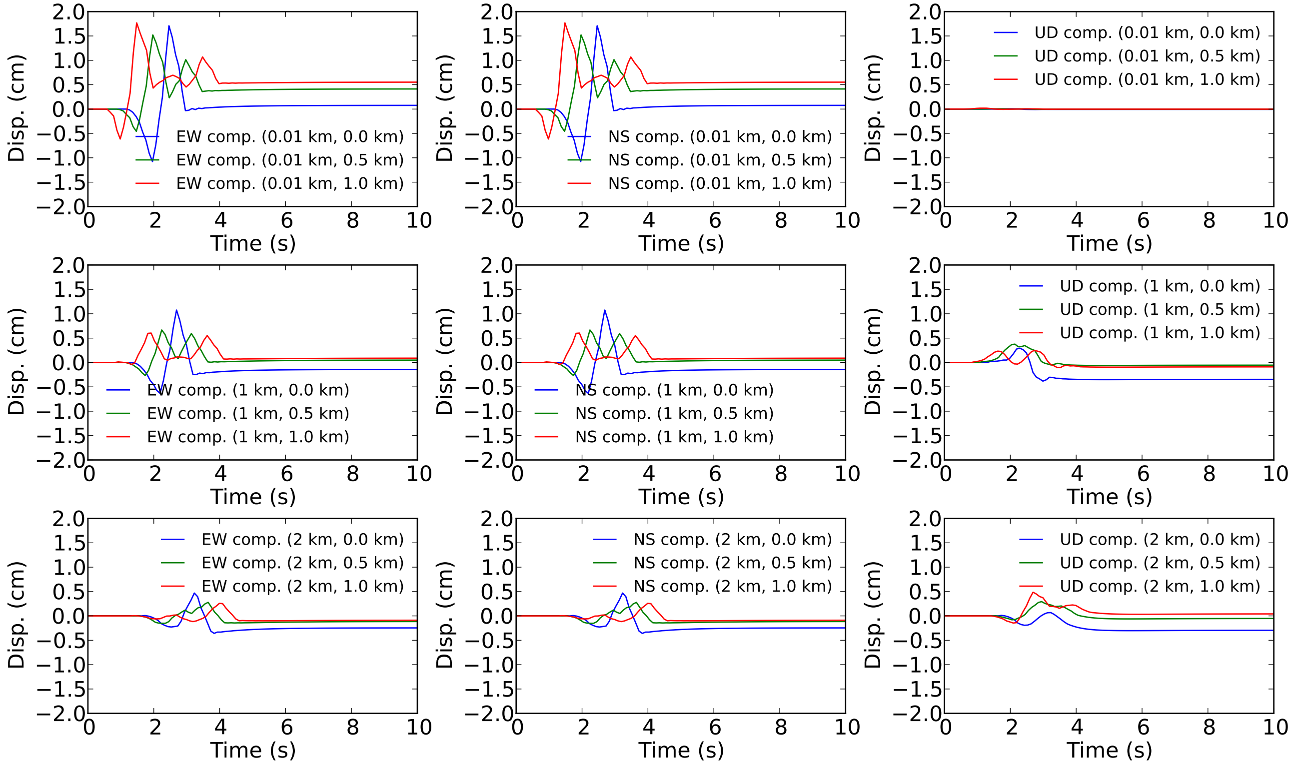


Figure 706.17: Calculated time history displacement, station azimuth = 45°

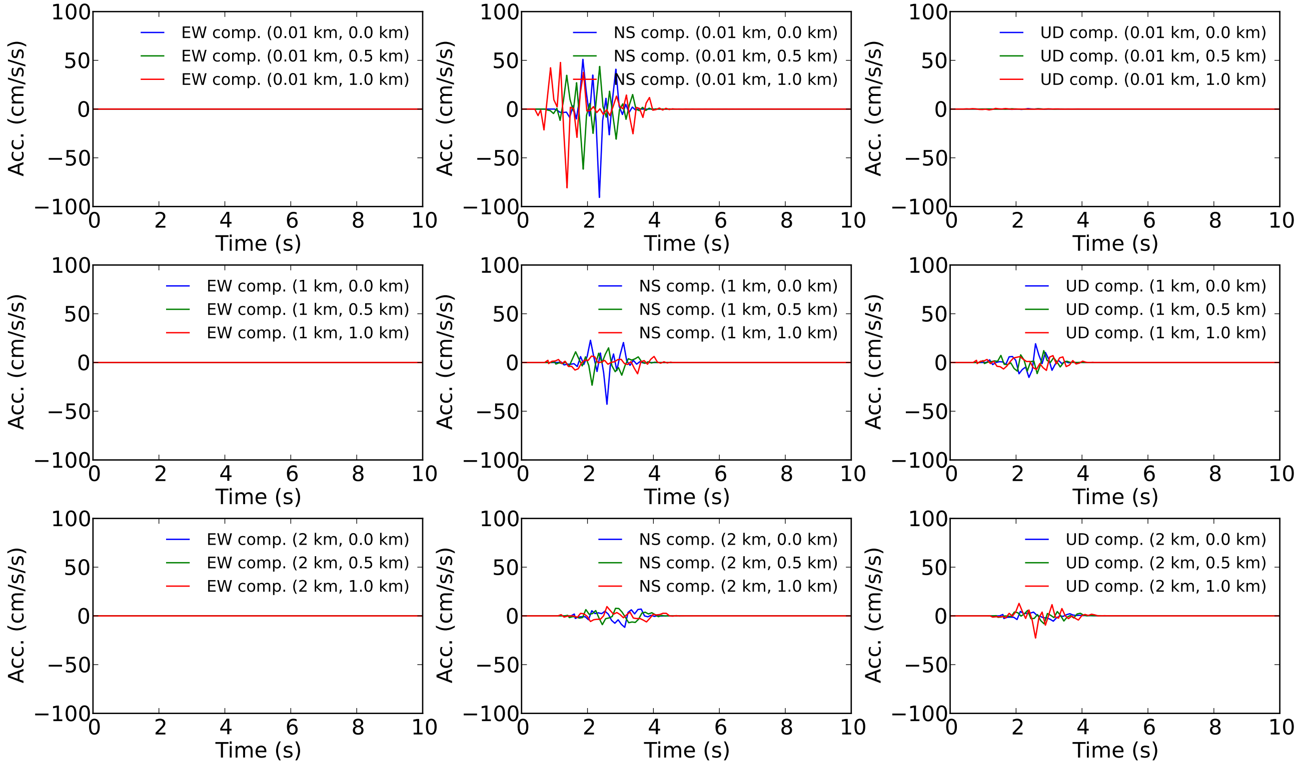


Figure 706.18: Calculated time history acceleration, station azimuth $= 90^{\circ}$

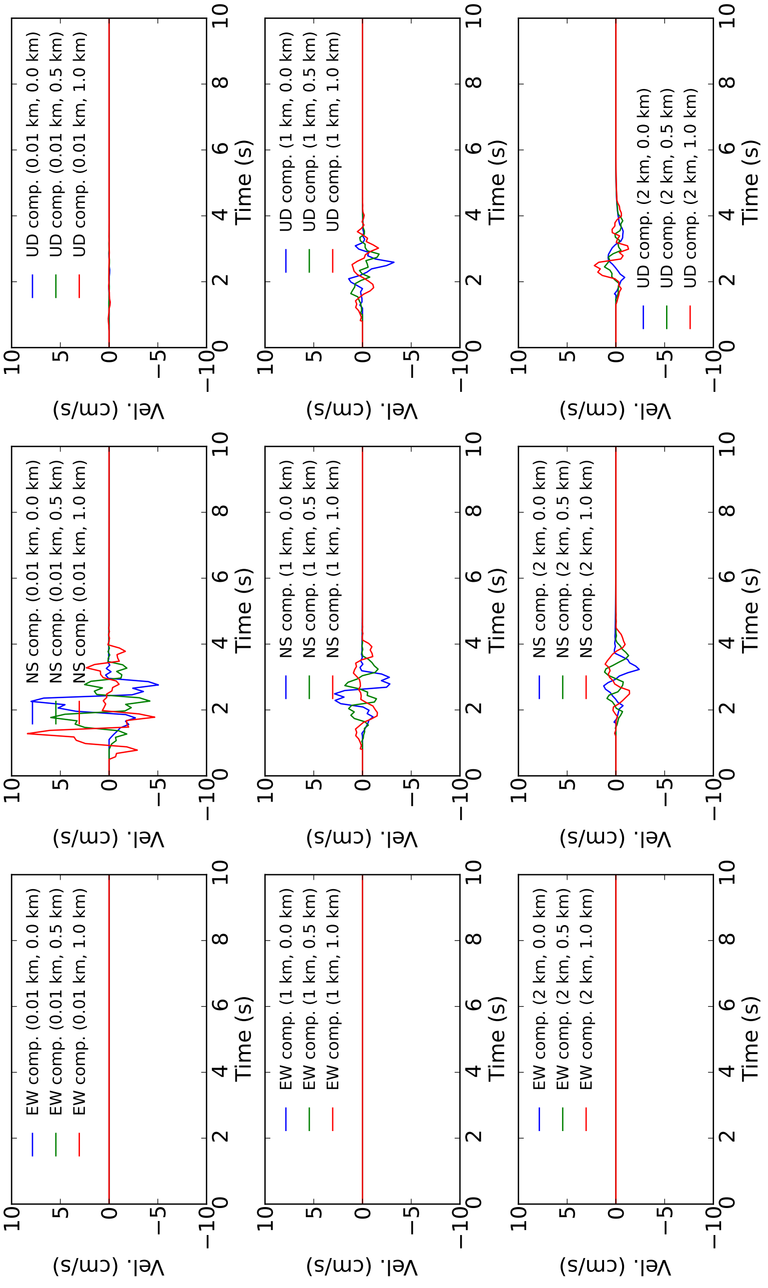


Figure 706.19: Calculated time history velocity, station azimuth = 90°

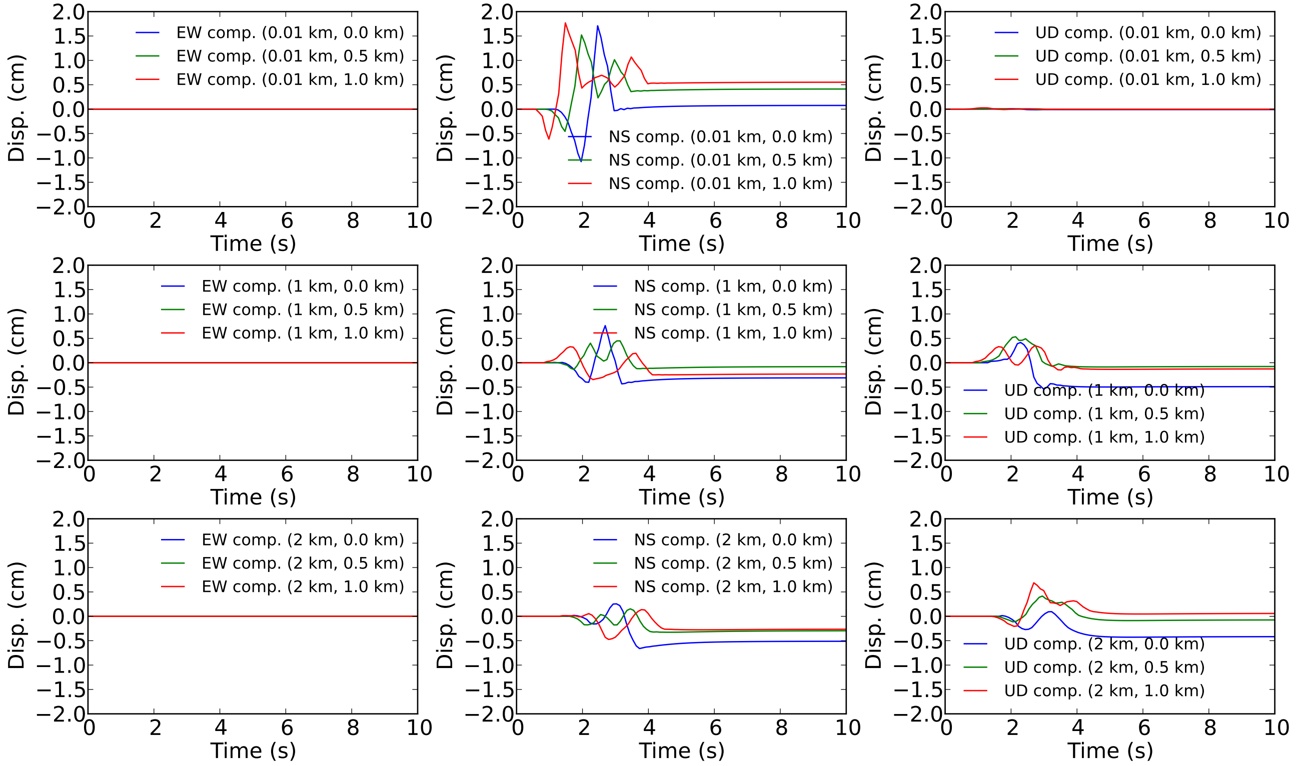


Figure 706.20: Calculated time history displacement, station azimuth $= 90^\circ$

706.1.2.3 Case 3: normal fault / single layer ground

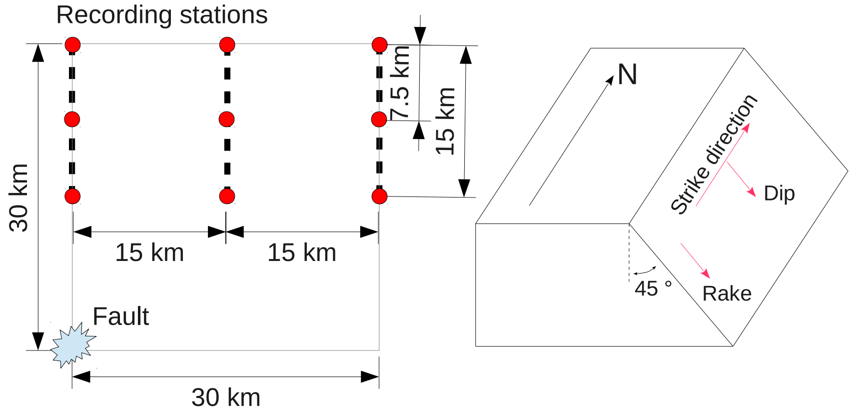


Figure 706.21: Ground and fault model used for analysis, results are captured on circles

Normal fault is tested. Figure 706.21 shows model used for the analysis. Wave is propagated through the single layer ground (Vs = 1 km/s). Properties are shown as below.

- Ground properties
  - V_S = 1 km/s
  - V_P/V_S = 1.73
  - Poisson's ratio = 0.25
  - Density = 1.32 g/cm^3
  - Shear modulus = 1.32 GPa
  - Elastic modulus = 3.31 GPa
- Fault properties
  - Moment magnitude = 5.0
  - Strike = 0°
  - Dip = 45°
  - Rake = 90°
  - Double - coupled source
  - Triangular source time function
- Wave properties

  - dt = 0.1 s (Max available freq. = 5 Hz, Nyquist freq.)

In this example, the distance between the fault and the station is increased and magnitude is changed also (Mw = 5.0). Fault is located at 30 km depth, 30 km away from the recording stations (Figure 706.21). Double coupled fault source is assumed and triangular source time function is used (Aki and Richards, 2002). Recording points are similar as prior examples (total 9 stations). Azimuth of recording station is set to 0°, 45°, and 90°.

Figure 706.22 – 706.30 show analyses results. Since the distance between fault and station is increased to 30 km and waves are propagated through the ground with relatively low shear wave velocity, arrival of propagating and reflecting waves can be observed easily (the first arrival of P wave followed by S wave). Permanent displacements by the fault movement are observed as desired at all stations (0°, 45°, and 90°).

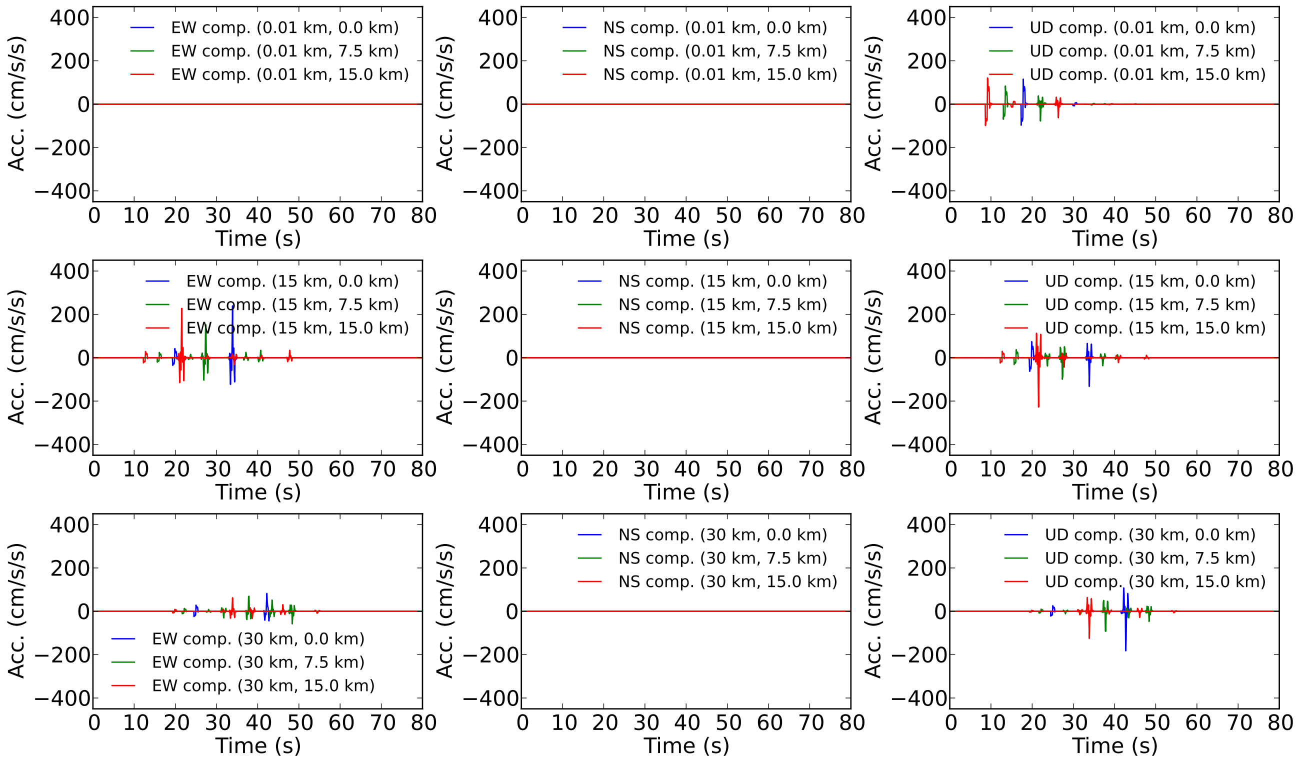


Figure 706.22: Calculated time history acceleration, station azimuth $= 0^\circ$

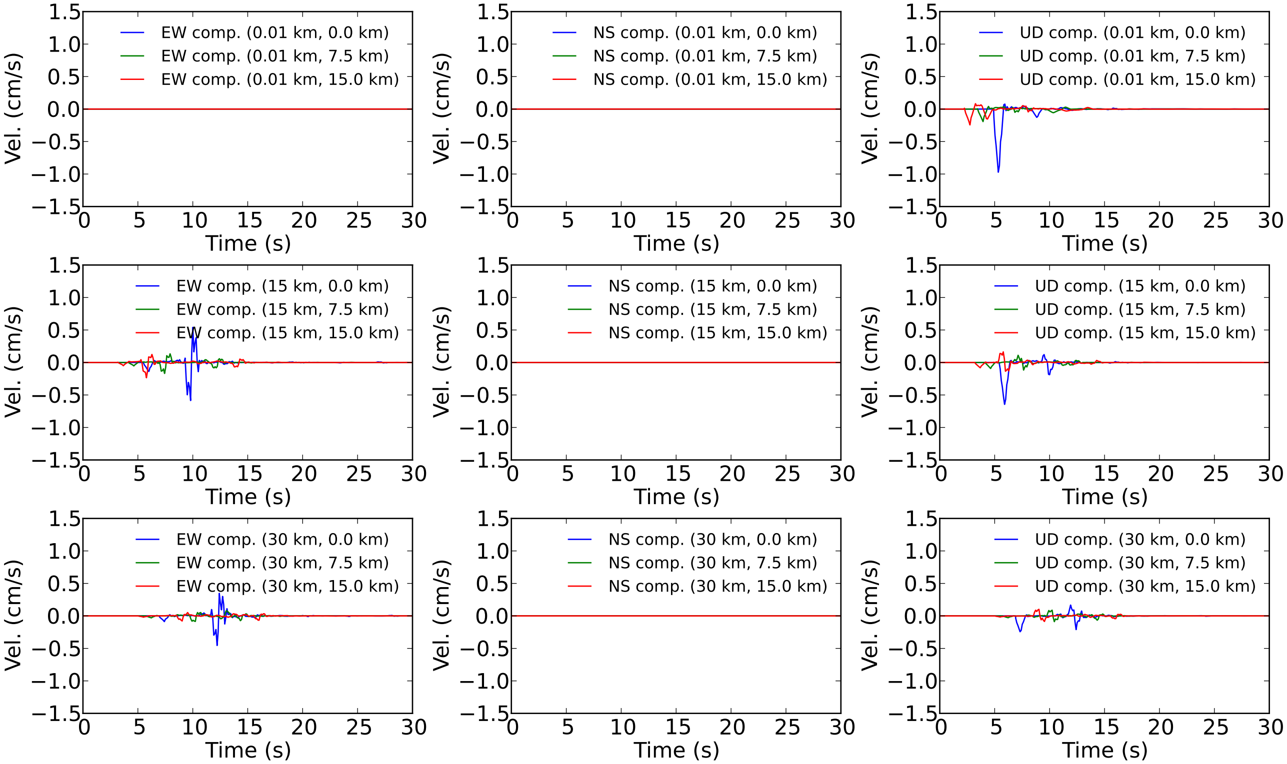


Figure 706.23: Calculated time history velocity, station azimuth $= 0^{\circ}$

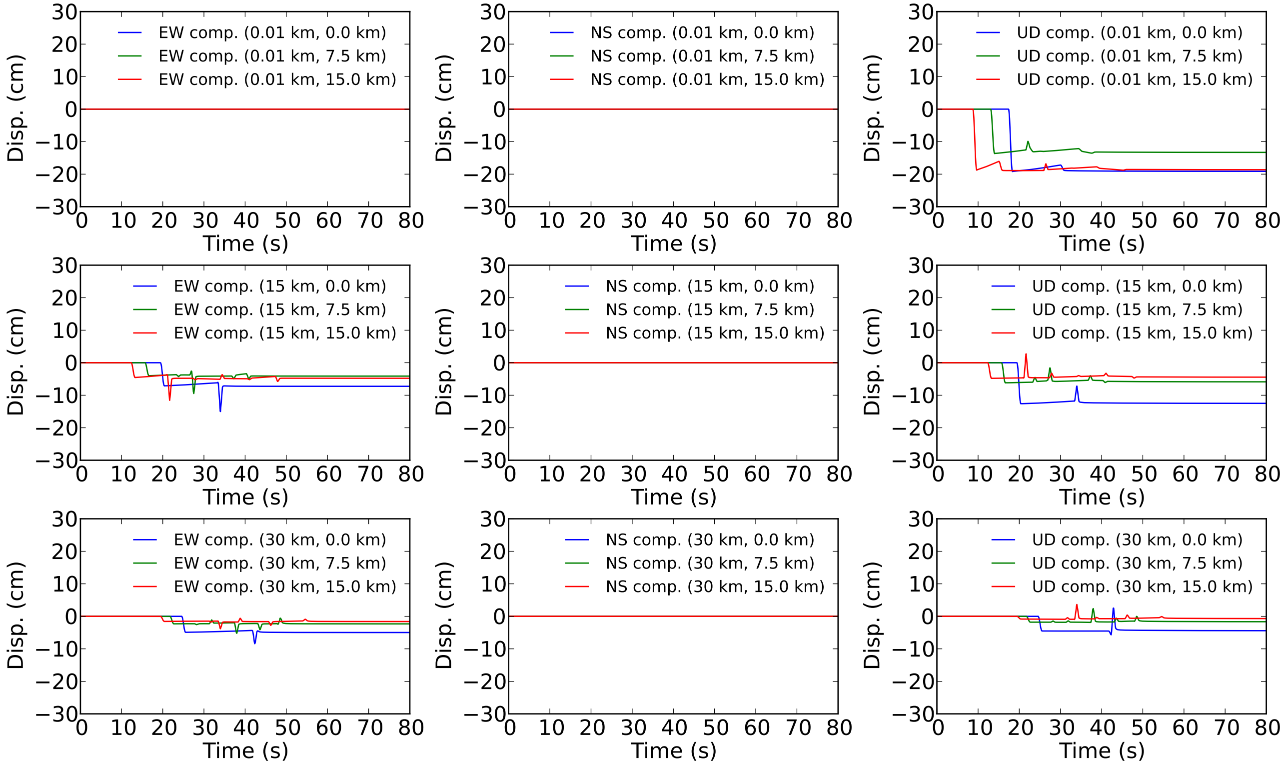


Figure 706.24: Calculated time history displacement, station azimuth $= 0^\circ$

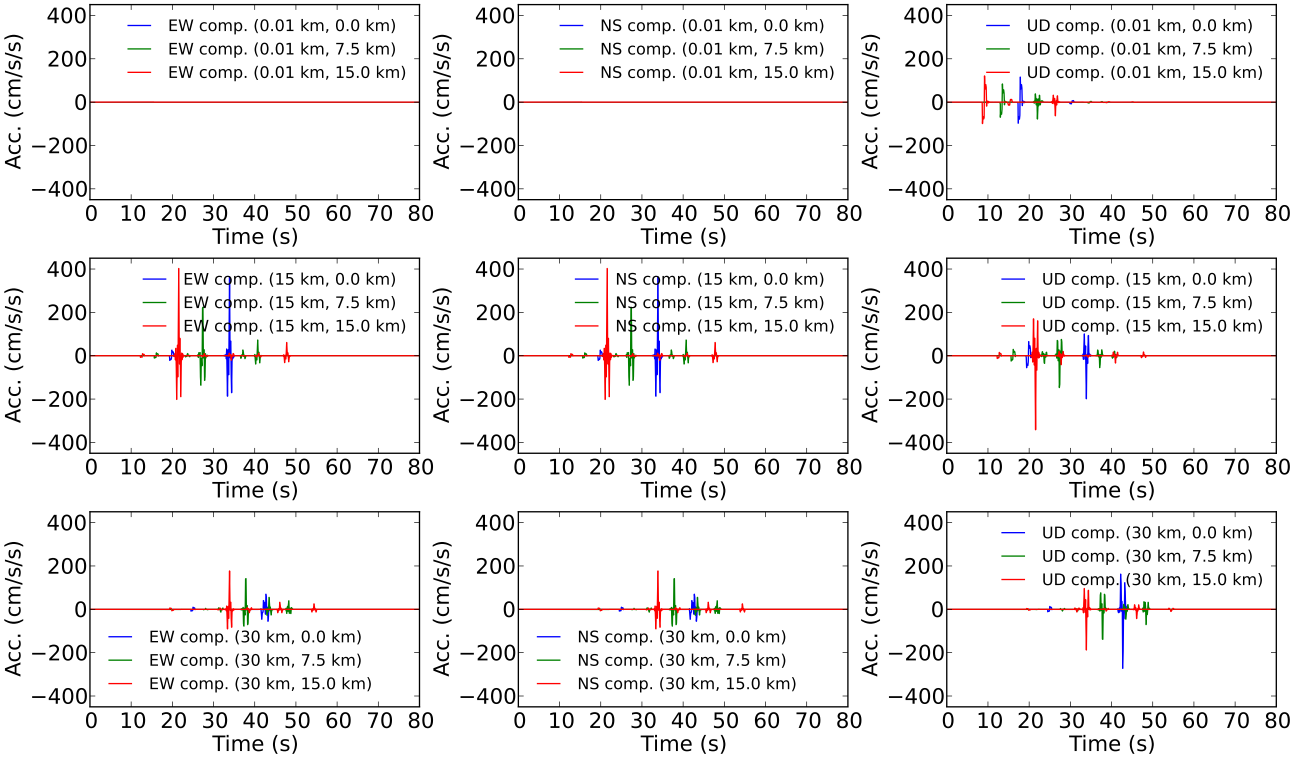


Figure 706.25: Calculated time history acceleration, station azimuth $= 45^{\circ}$

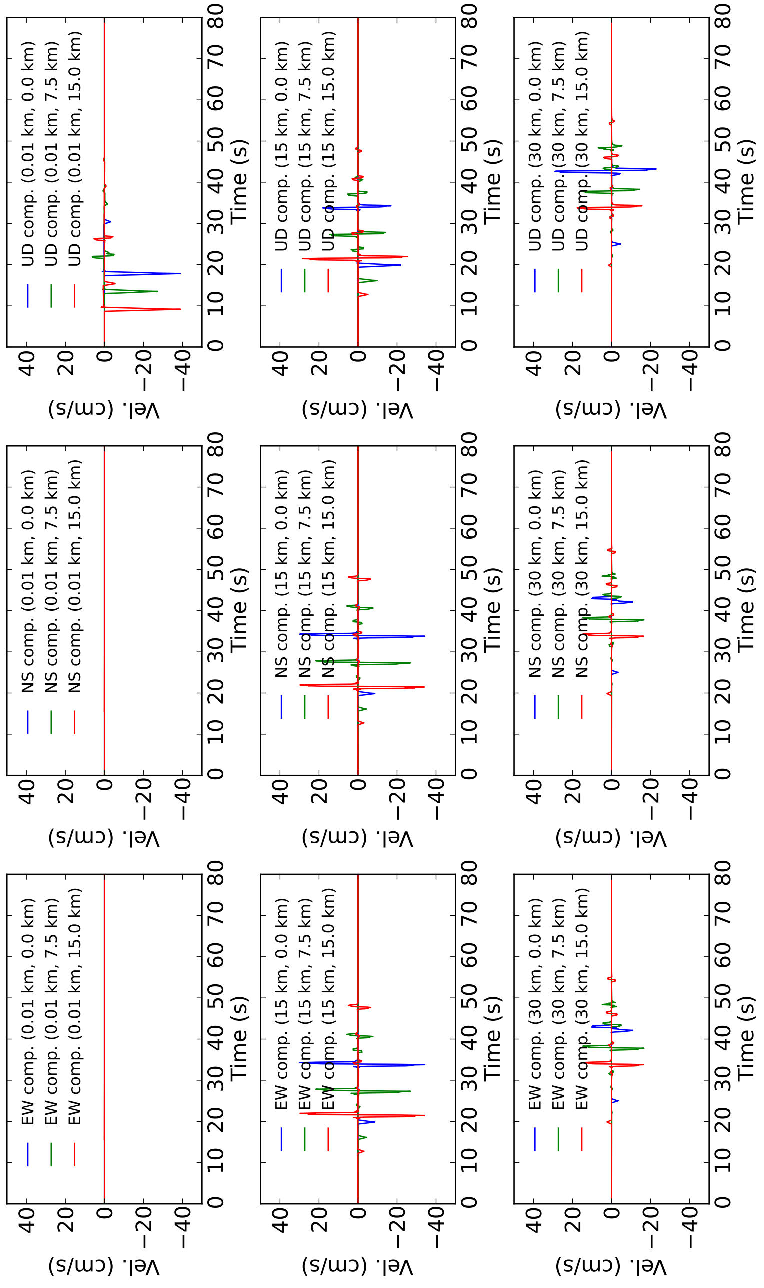


Figure 706.26: Calculated time history velocity, station azimuth = 45°

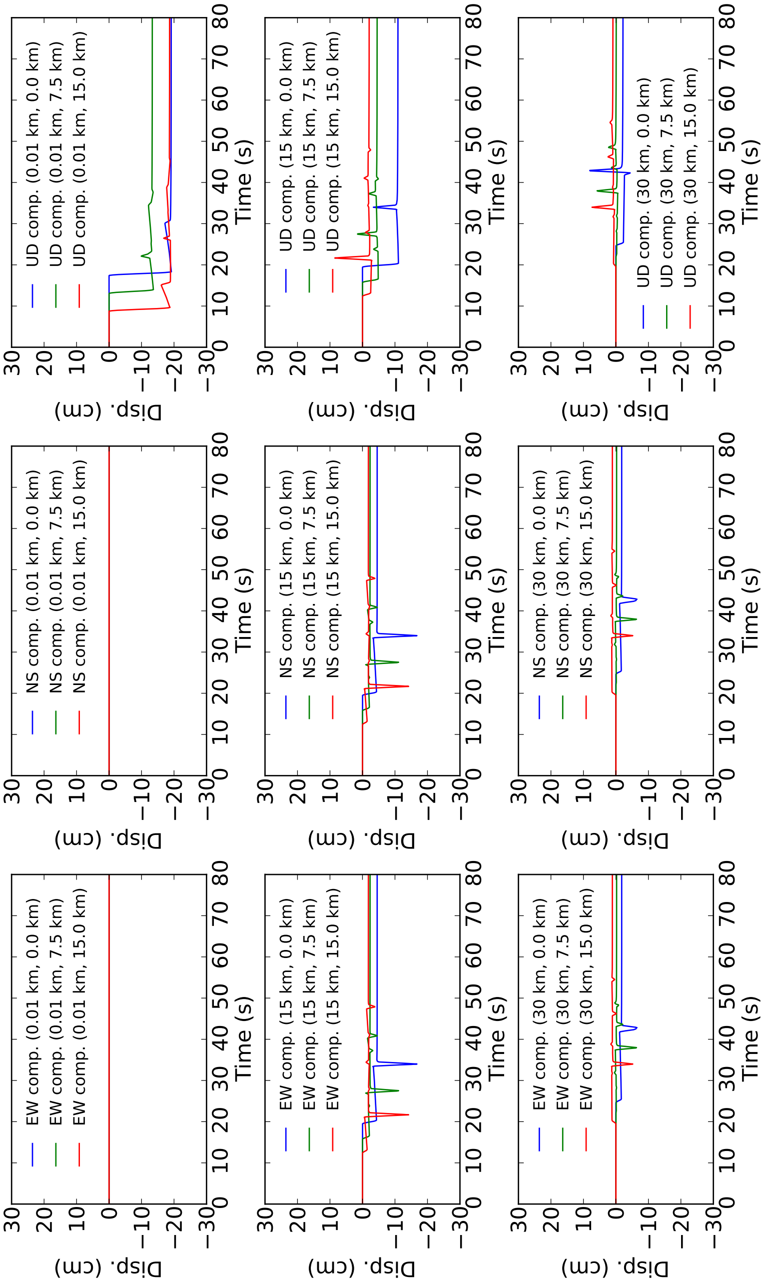


Figure 706.27: Calculated time history displacement, station azimuth = 45°

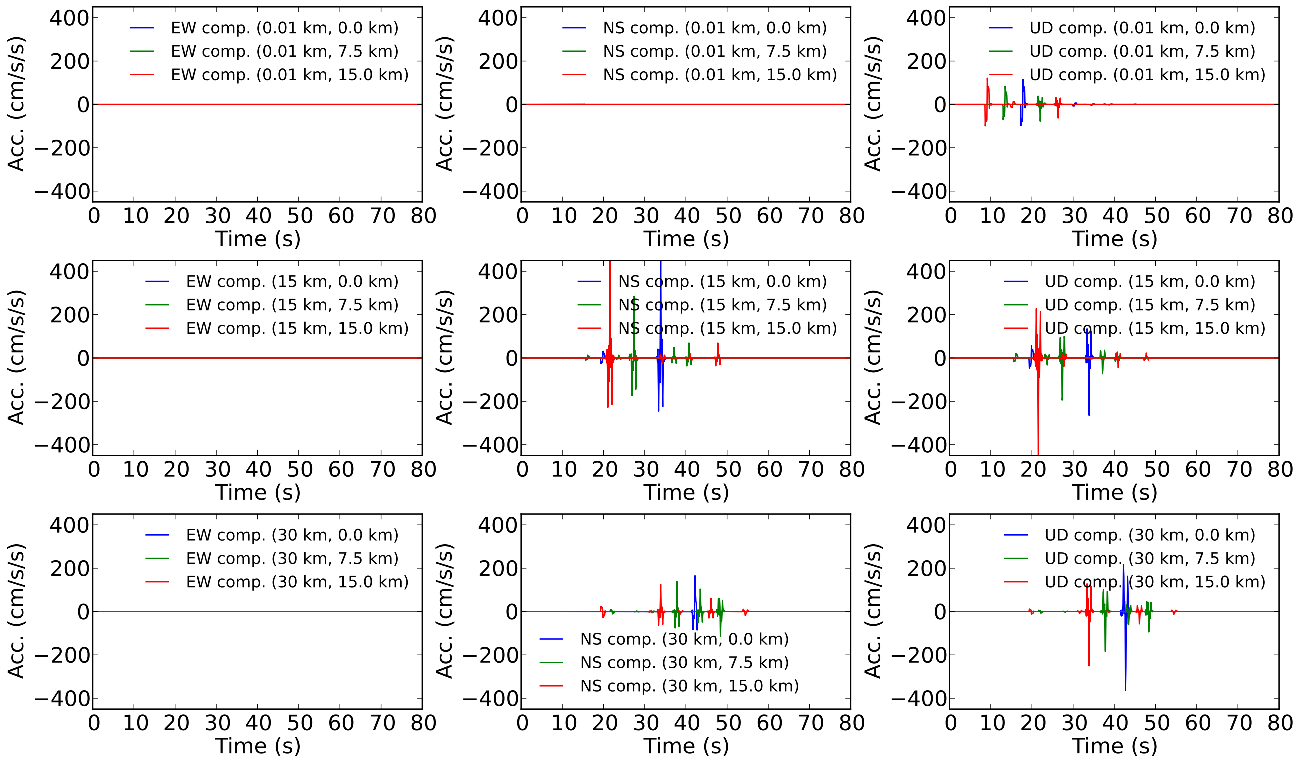


Figure 706.28: Calculated time history acceleration, station azimuth $= 90^{\circ}$

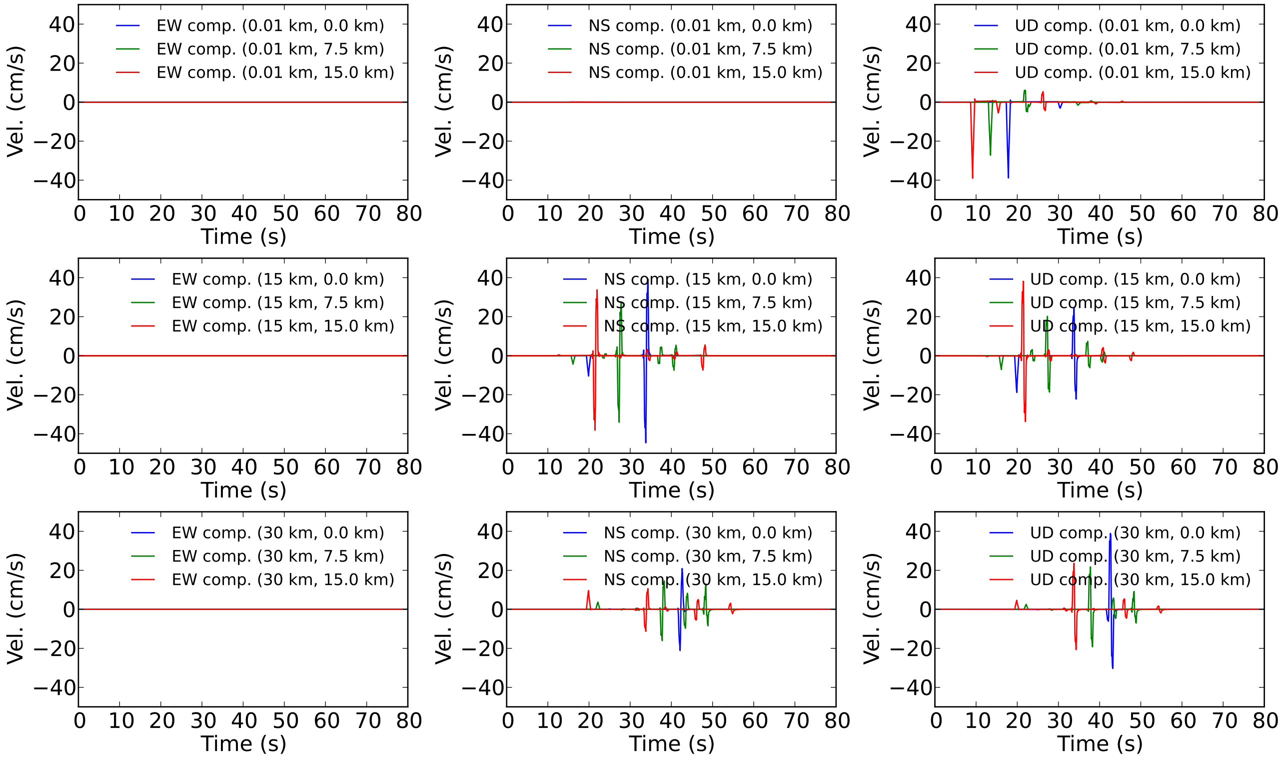


Figure 706.29: Calculated time history velocity, station azimuth = 90°

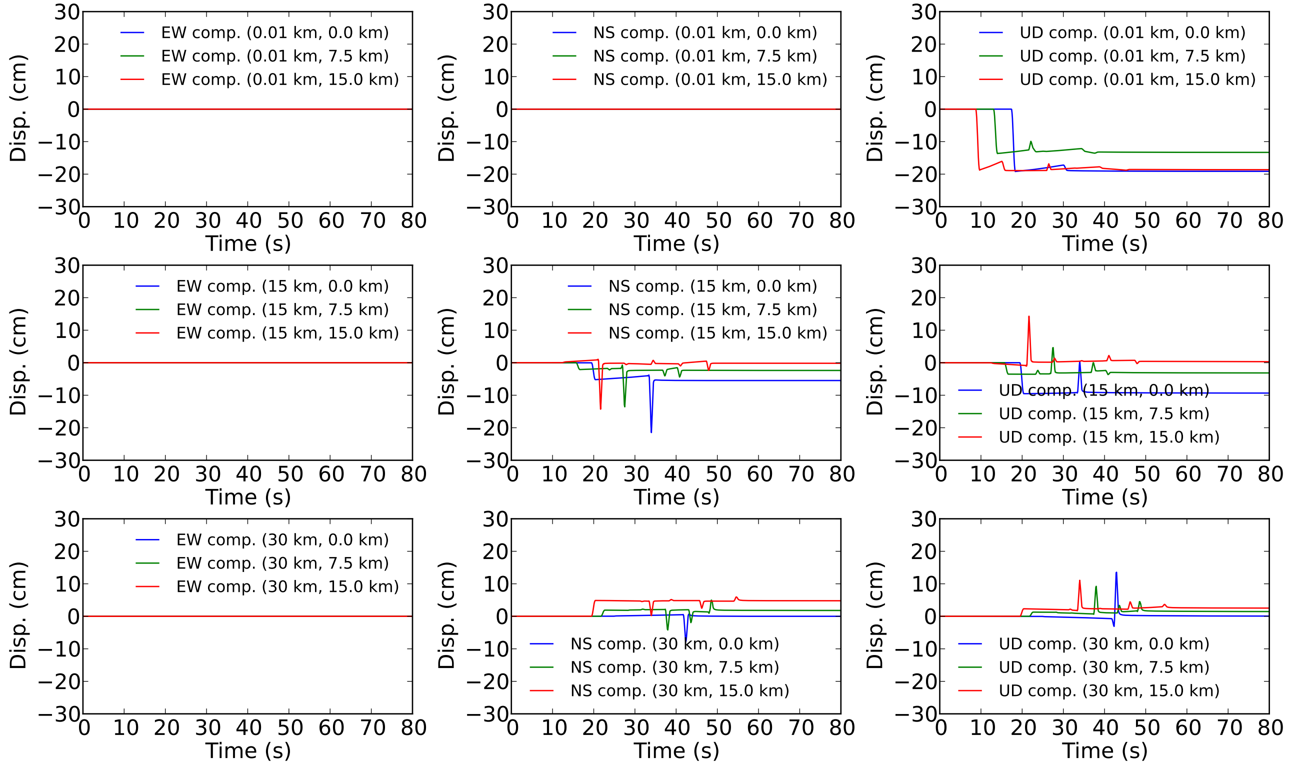


Figure 706.30: Calculated time history displacement, station azimuth = 90°

706.1.2.4 Case 4: normal fault / layered ground

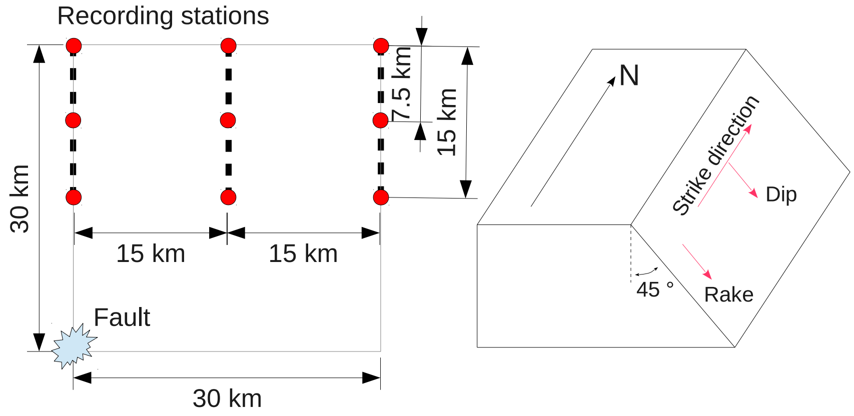


Figure 706.31: Ground and fault model used for analysis, results are captured on circles

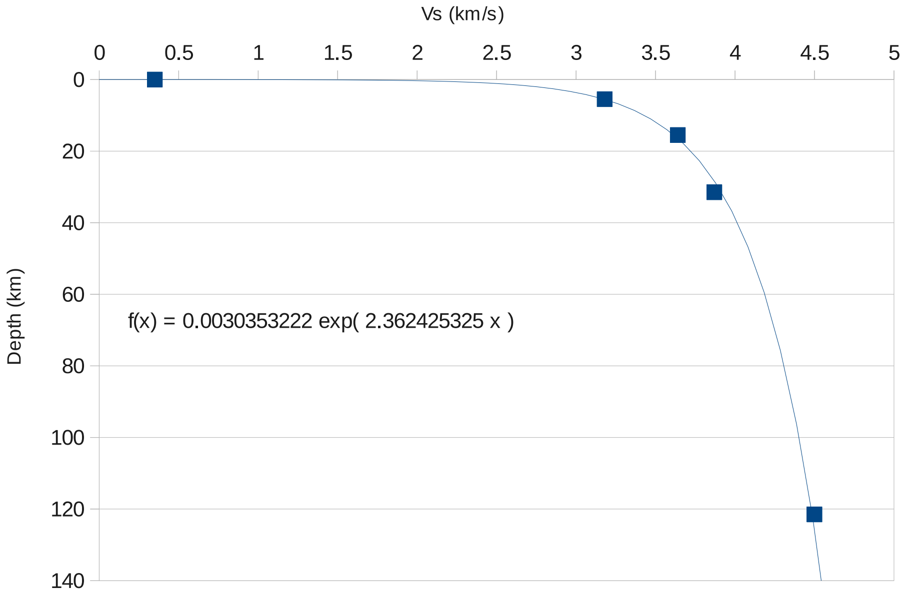


Figure 706.32: Vs profile, square points are hk model; blue line is trend line based on hk model

Normal fault within layered ground is modeled. Model is as shown Figure 706.31 and 706.32. Properties are shown below.

- Ground properties: see Figure 706.32 and Table 706.1
- Fault properties

  - Moment magnitude = 5.0
  - Strike = 0°
  - Dip = 45°
  - Rake = 90°
  - Double - coupled source
  - Triangular source time function

- Wave properties
  - dt = 0.1 s (Max available freq. = 5 Hz, Nyquist freq.)

Fault is located at depth of 30 km, 30 km away from the recording point (station) at surface. Double coupled fault source is assumed and triangular source time function is used (Aki and Richards, 2002). Recording points are similar as prior examples (total 9 stations). The 1D standard southern California model (Hadley and Kanamori (1977), hk model hereafter) is used for ground layering. As shown in Figure 706.32, hk model is interpolated and divided to define ground layer (Table 706.1).

Results are shown on Figure 706.33 – 706.35. Since wave is propagated through the layered ground, compared to prior examples, more realistic waves are observed.

Table 706.1: Ground properties for the example

Depth (km)	Thickness (km)	VS (km/s)	VP/VS	QB	VP (km/s)	Poisson's R	Density (g/cm3)	G (GPa)	E (GPa)
0.01	0.01	0.50	1.730	500	0.87	0.25	1.05	0.27	0.67
0.02	0.01	0.80	1.730	500	1.38	0.25	1.21	0.77	1.93
0.03	0.01	0.97	1.730	500	1.68	0.25	1.31	1.23	3.07
0.04	0.01	1.09	1.730	500	1.89	0.25	1.37	1.64	4.09
0.05	0.01	1.19	1.730	500	2.05	0.25	1.43	2.01	5.01
0.06	0.01	1.26	1.730	500	2.19	0.25	1.47	2.34	5.86
0.07	0.01	1.33	1.730	500	2.30	0.25	1.51	2.66	6.64
0.08	0.01	1.38	1.730	500	2.40	0.25	1.54	2.95	7.36
0.09	0.01	1.43	1.730	500	2.48	0.25	1.56	3.22	8.04
0.1	0.01	1.48	1.730	500	2.56	0.25	1.59	3.48	8.69
0.2	0.10	1.77	1.730	500	3.07	0.25	1.75	5.50	13.75
0.3	0.10	1.94	1.730	500	3.36	0.25	1.85	6.98	17.44
0.4	0.10	2.07	1.730	500	3.57	0.25	1.91	8.17	20.41
0.5	0.10	2.16	1.730	500	3.74	0.25	1.97	9.18	22.93
0.6	0.10	2.24	1.730	500	3.87	0.25	2.01	10.06	25.13
0.7	0.10	2.30	1.730	500	3.98	0.25	2.04	10.85	27.10
0.8	0.10	2.36	1.730	500	4.08	0.25	2.08	11.56	28.88
0.9	0.10	2.41	1.730	500	4.17	0.25	2.10	12.21	30.51
1	0.10	2.45	1.730	500	4.25	0.25	2.13	12.82	32.02
1.5	0.50	2.63	1.730	500	4.54	0.25	2.22	15.33	38.30
2.5	1.00	2.84	1.730	500	4.92	0.25	2.34	18.92	47.28
3.5	1.00	2.98	1.730	500	5.16	0.25	2.42	21.57	53.89
4.5	1.00	3.09	1.730	500	5.35	0.25	2.48	23.70	59.21
5.5	1.00	3.18	1.730	600	5.50	0.25	2.53	25.59	63.93
16	10.50	3.64	1.731	600	6.30	0.25	2.79	36.92	92.26

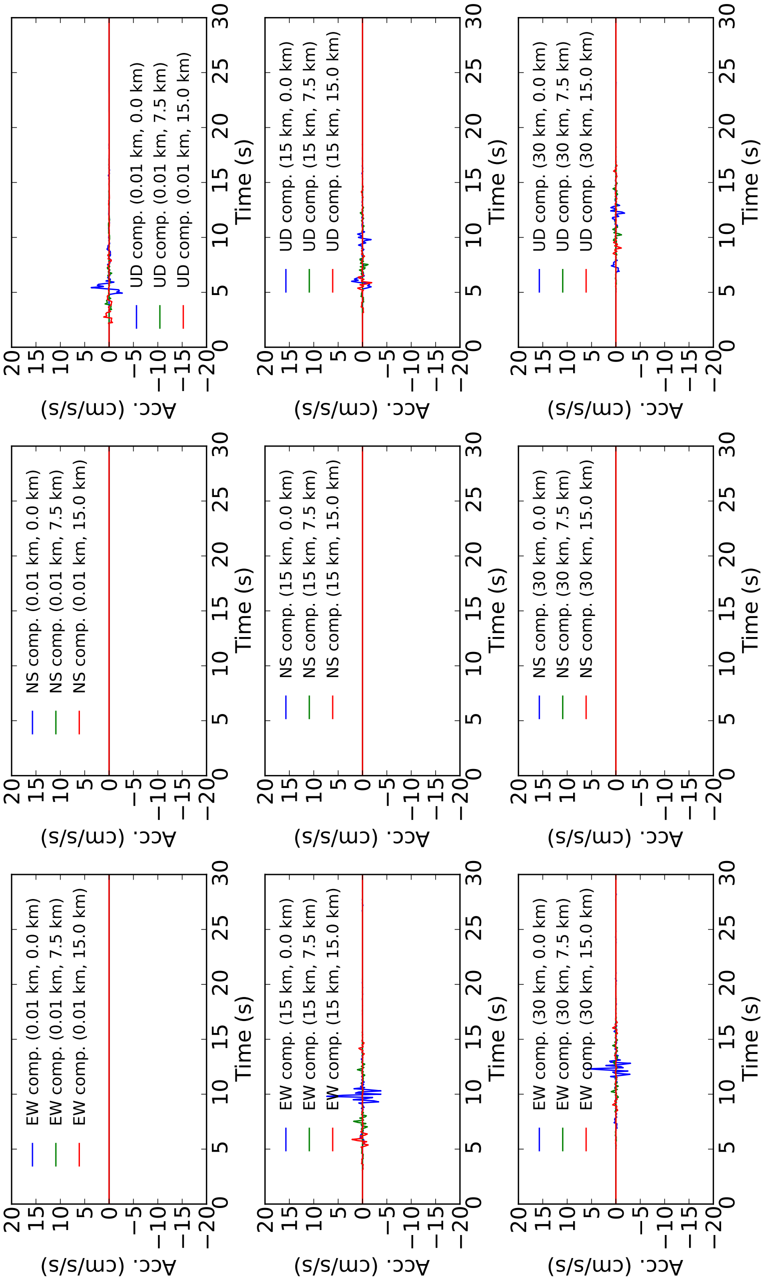


Figure 706.33: Calculated time history acceleration, station azimuth = 0°

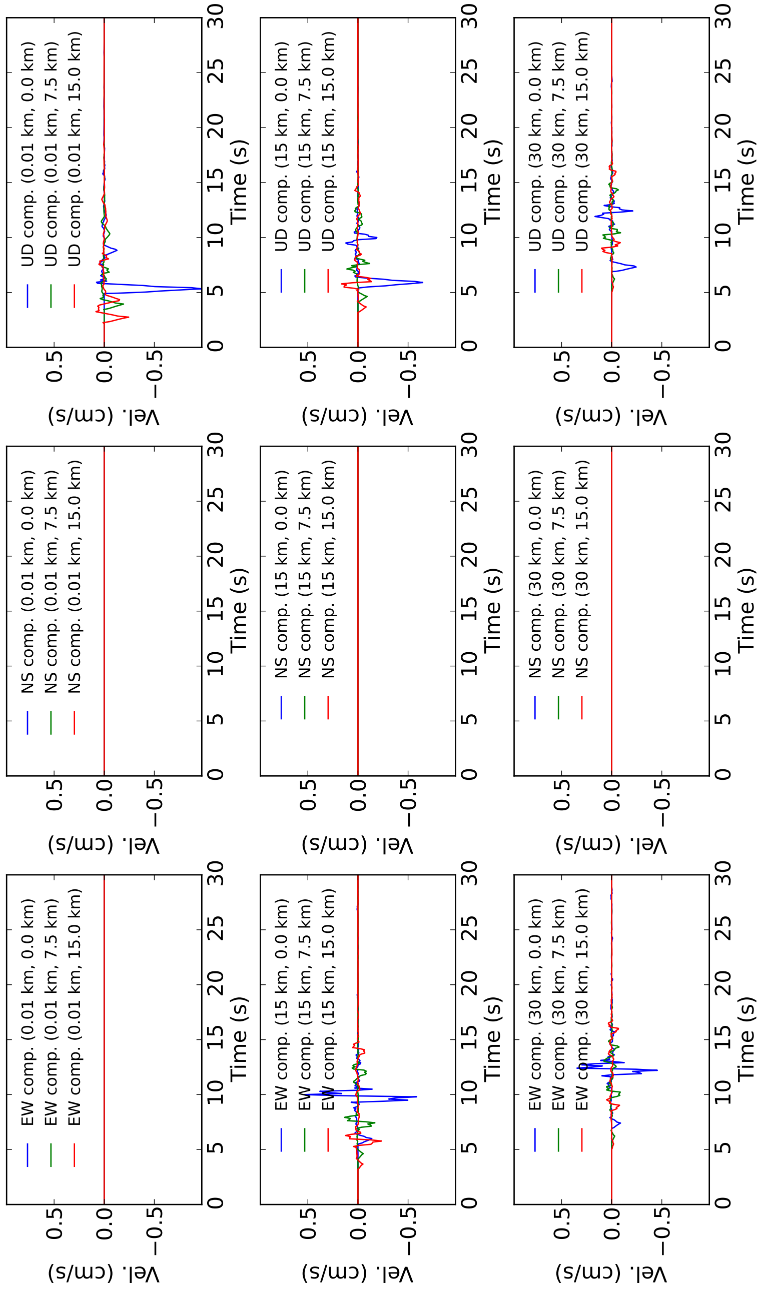


Figure 706.34: Calculated time history velocity, station azimuth = 0°

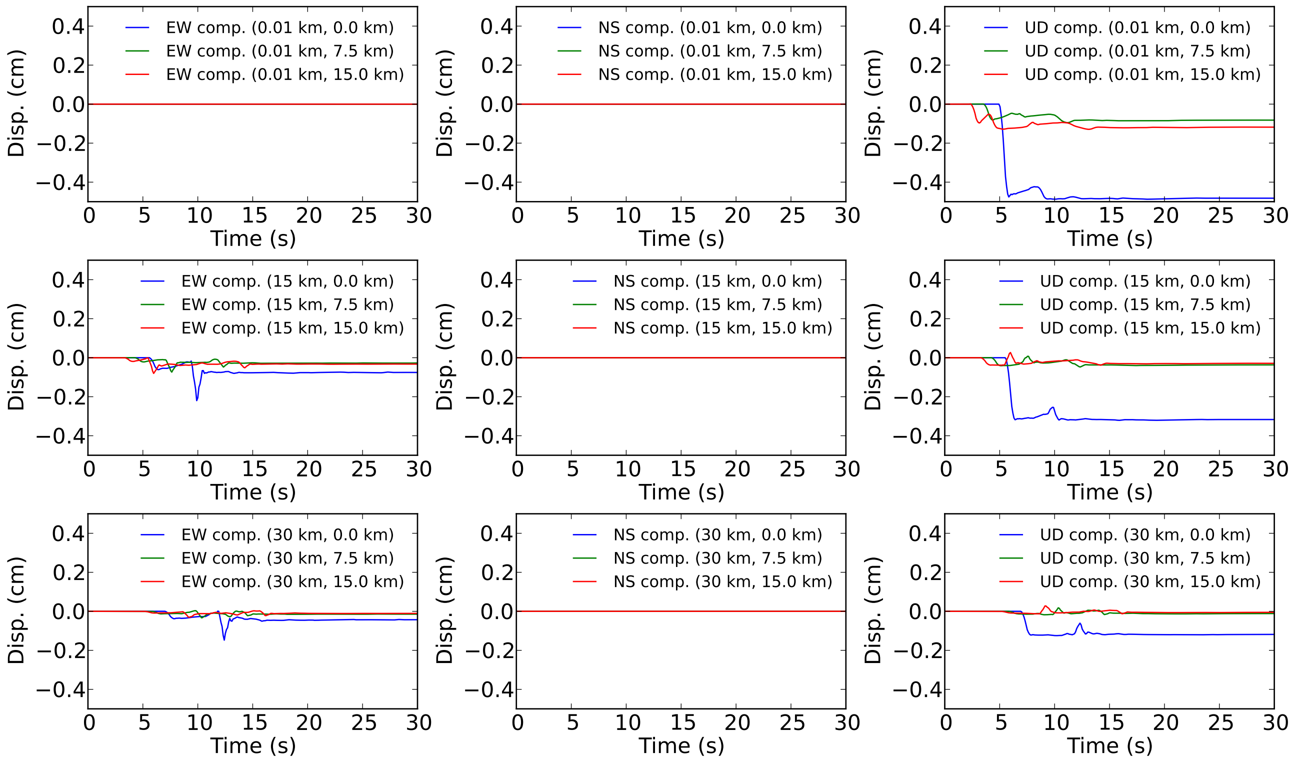


Figure 706.35: Calculated time history displacement, station azimuth $= 0^{\circ}$

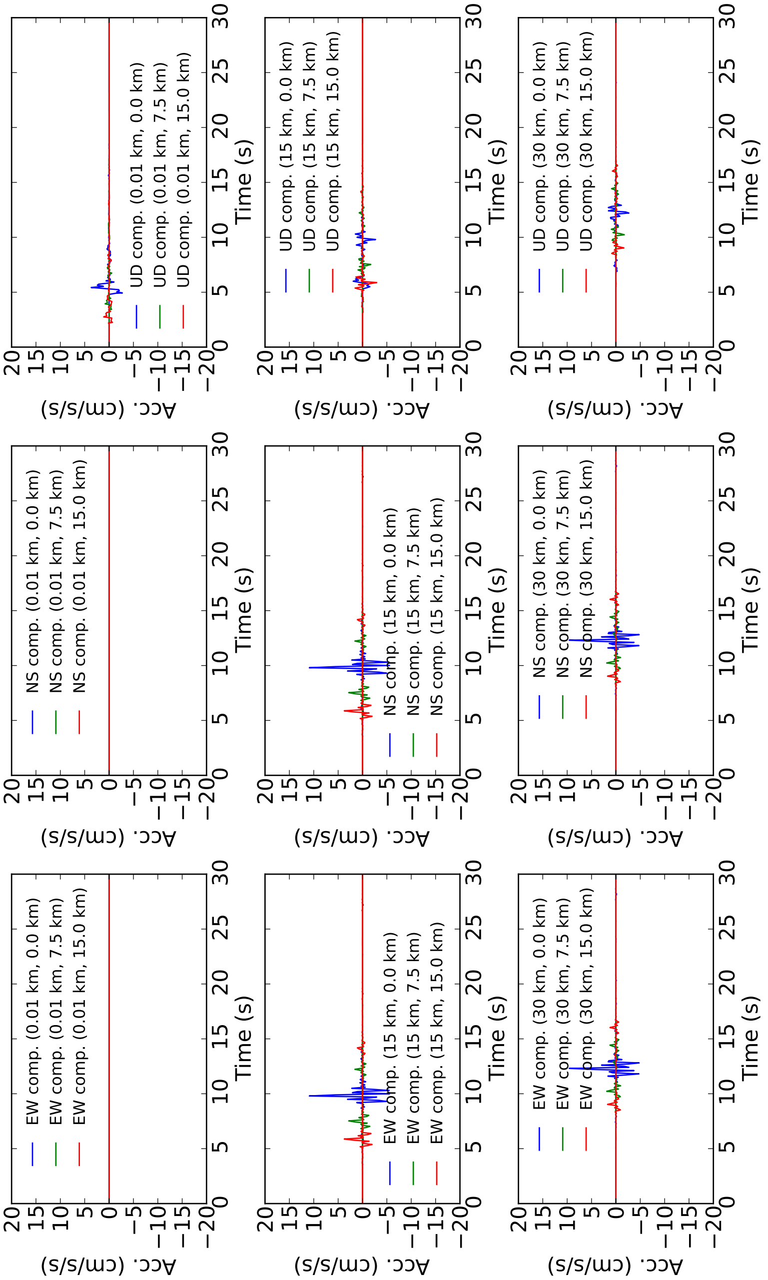


Figure 706.36: Calculated time history acceleration, station azimuth = 45°

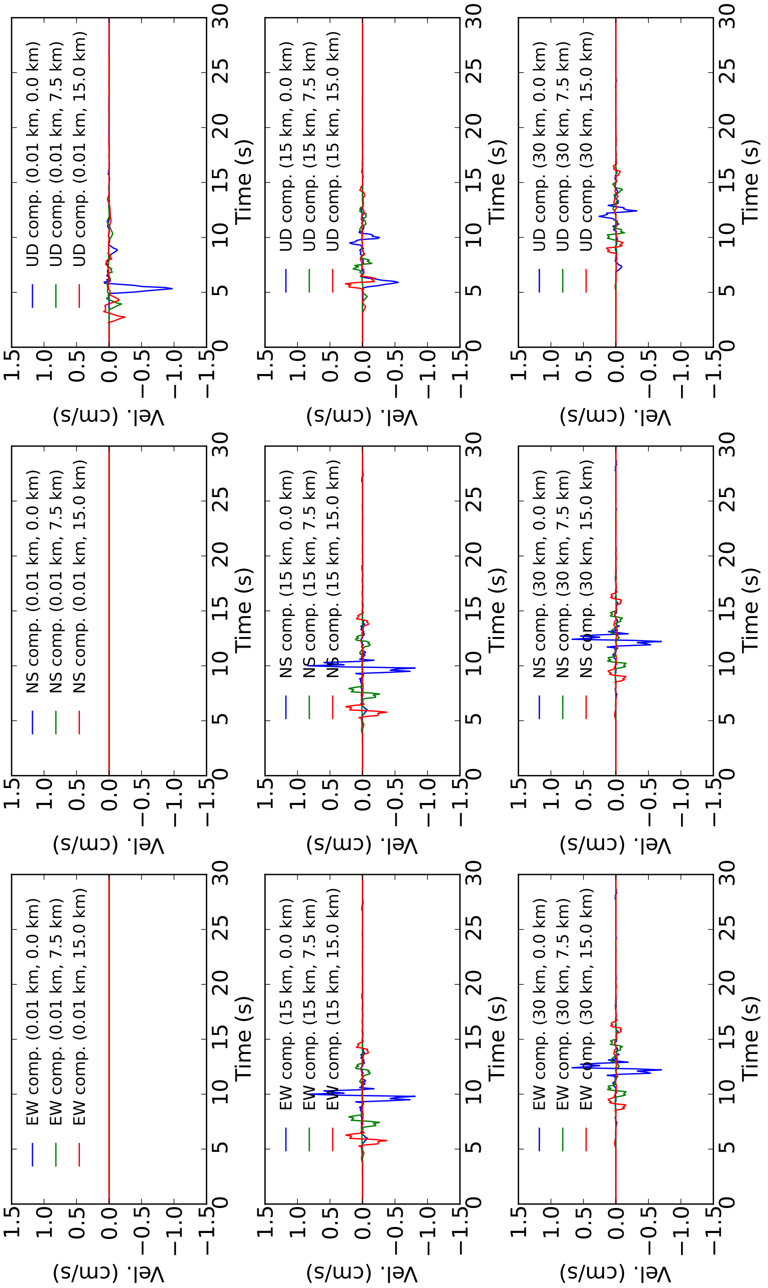


Figure 706.37: Calculated time history velocity, station azimuth = 45°

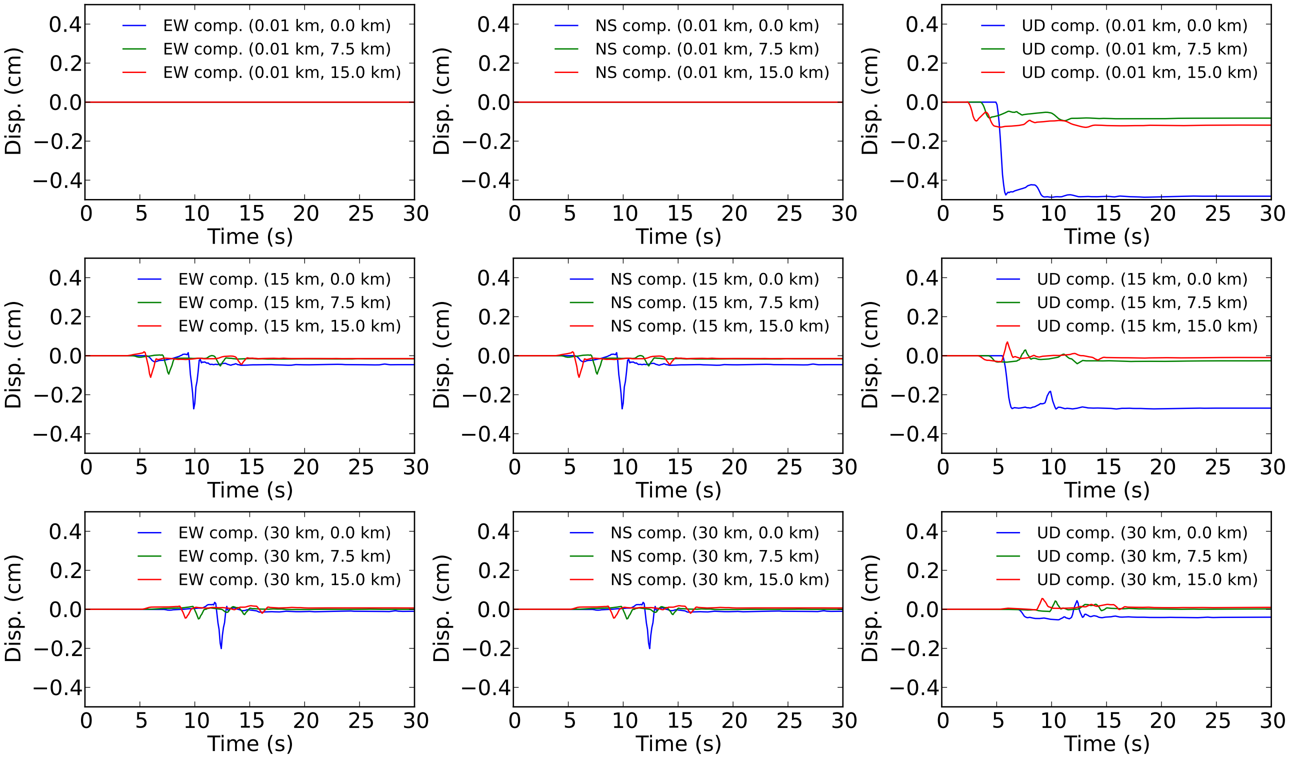


Figure 706.38: Calculated time history displacement, station azimuth $= 45^\circ$

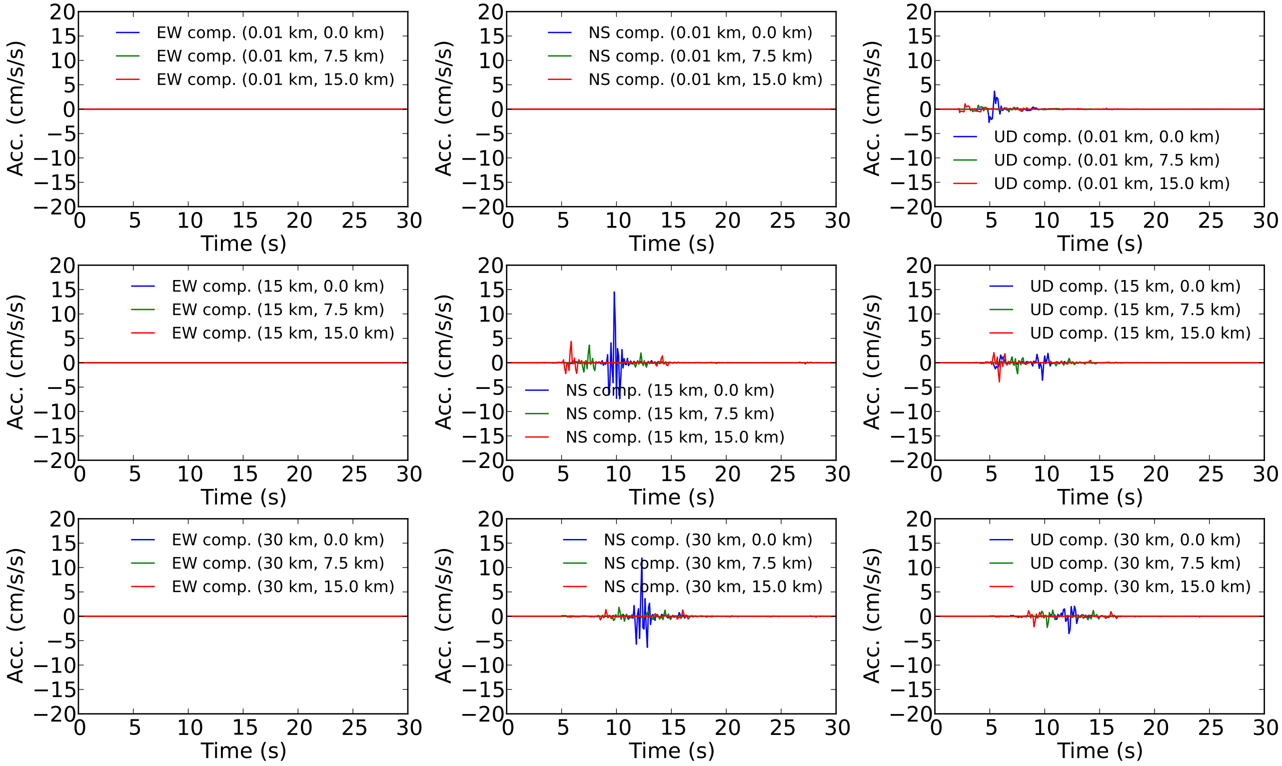


Figure 706.39: Calculated time history acceleration, station azimuth = 90°

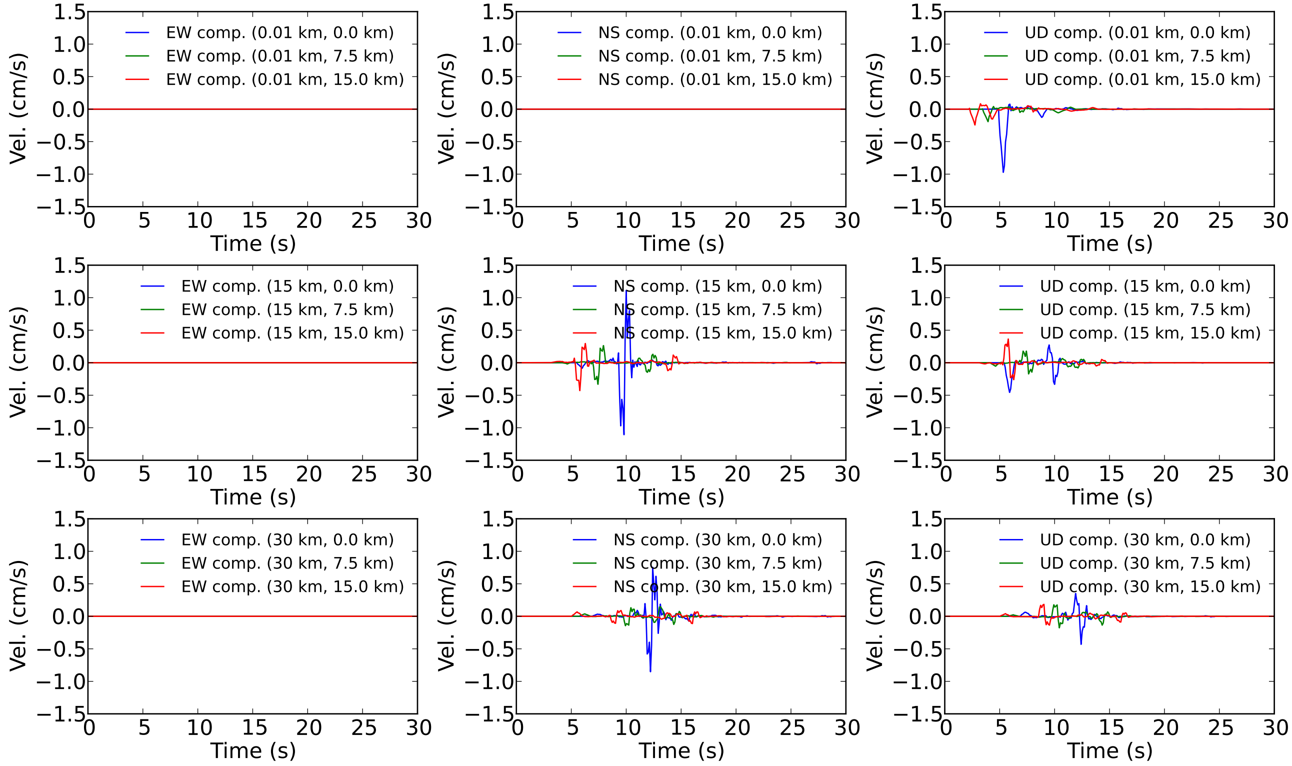


Figure 706.40: Calculated time history velocity, station azimuth $= 90^\circ$

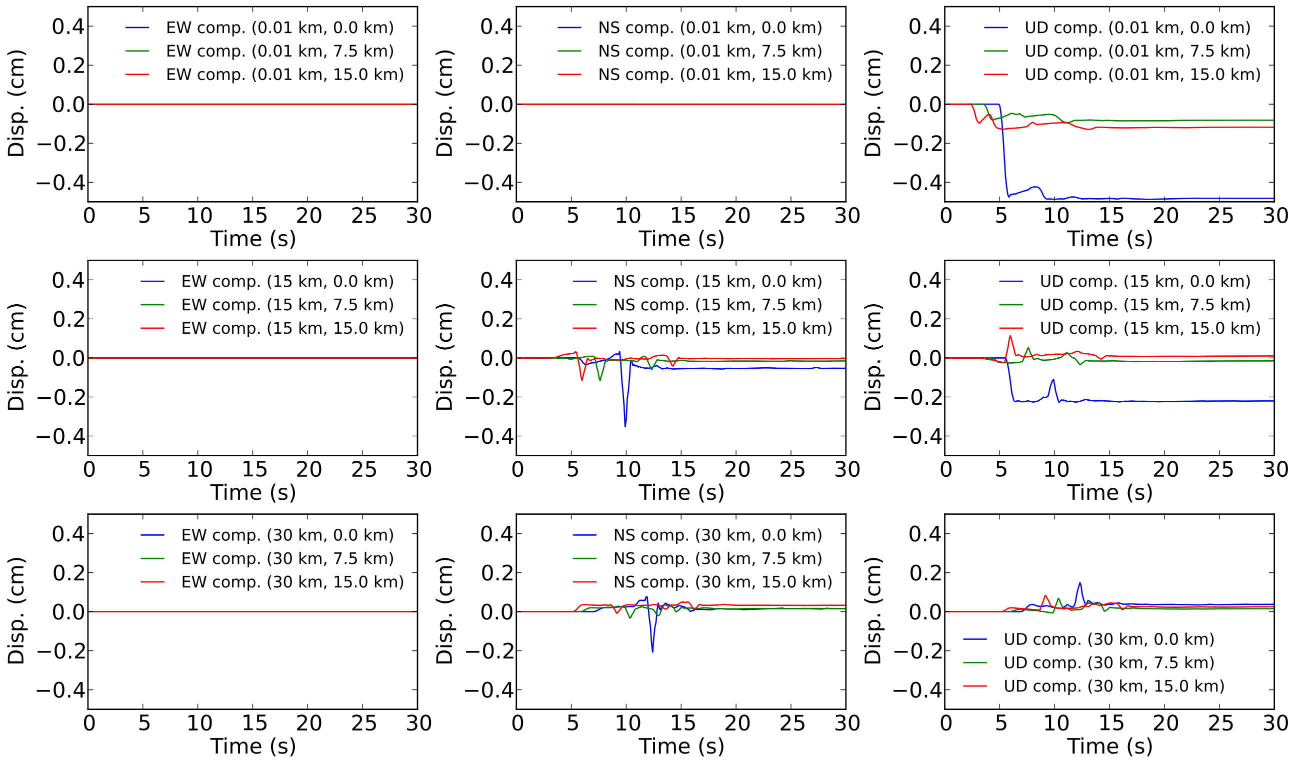


Figure 706.41: Calculated time history displacement, station azimuth $= 90^{\circ}$

706.1.2.5 Case 5: Northridge earthquake / layered ground

In this example, Northridge earthquakes are simulated. Northridge earthquake has occurred on January, 1994 with a moment magnitude of 6.7. Properties are shown as below and partly adapted from Hisada (2008) and Wald et al. (1996).

- Ground properties: see Table 706.2
- Fault properties
  - Moment magnitude = 6.7
  - Strike = 122°
  - Dip = 40°
  - Rake = 140°
  - Double - coupled source
  - Triangular source time function
- Wave properties
  - dt = 0.05 s (Max available freq. = 10 Hz, Nyquist freq.)

Source depth is set as 25 km and epicentral distance from the fault to the station is set as 30 km. Station is located on the ground surface. As shown in Table 706.2, ground is divided into 9 layers (Hisada, 2008).

Figure 706.42 shows analyses results. Computed results are compared with measured one. Measured records are obtained from cosmos virtual data center `http://db.cosmos-eq.org/`. As shown in Figure 706.42, predicted seismogram agrees well with measured ones considering the fk package assumes a single point source and simplified ground.pdf.

Table 706.2: Ground properties for the example

Depth (km)	Thickness (km)	VS (km/s)	VP/VS	QB	VP (km/s)	Poisson's R	Density (g/cm3)	G (GPa)	E (GPa)
0.05	0.05	0.30	1.730	600	0.52	0.25	0.94	0.08	0.21
0.1	0.05	0.40	1.730	600	0.69	0.25	0.99	0.16	0.40
0.2	0.10	0.50	1.730	600	0.87	0.25	1.05	0.26	0.65
0.3	0.10	0.75	1.730	600	1.30	0.25	1.19	0.67	1.67
0.5	0.20	1.00	1.730	600	1.73	0.25	1.32	1.32	3.31
1.5	1.00	2.00	1.730	600	3.46	0.25	1.88	7.51	18.76
4	2.50	3.20	1.730	600	5.54	0.25	2.54	26.03	65.02
27	23.00	3.60	1.730	600	6.23	0.25	2.76	35.81	89.46
40	13.00	3.90	1.730	900	6.75	0.25	2.93	44.55	111.30

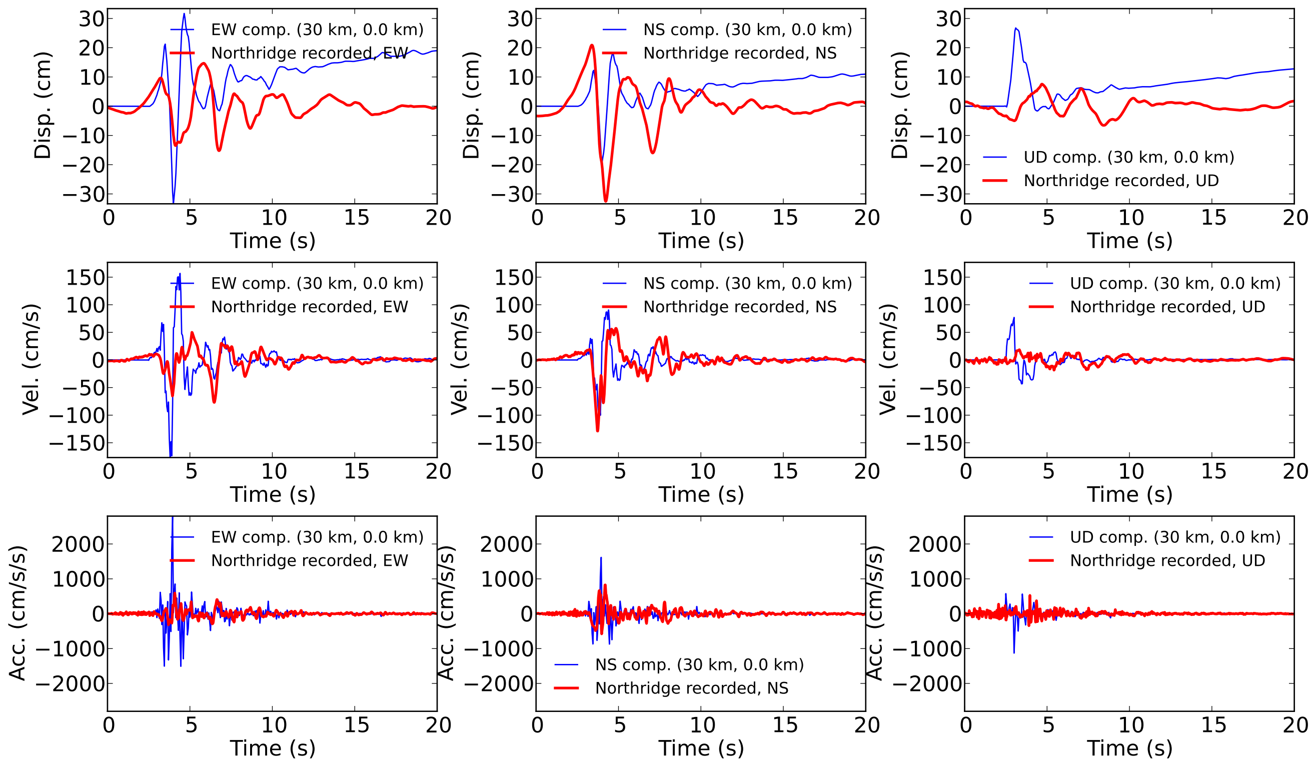


Figure 706.42: Analysis result, thick line is observed (SYL station, `http://db.cosmos-eq.org/`); results presented by the thin line are calculated using fk/syn

Appendix 707

Real-ESSI Illustrative Examples

(2015-2016-2017-2018-2019-2021-)

(In collaboration with Dr. Yuan Feng, Prof. José Abell, Prof. Sumeet Kumar Sinha, Prof. Han Yang and Dr. Hexiang Wang)

This chapter presents a number of illustrative examples. The main aim is simple: present Real-ESSI Simulator features (available elements, algorithms, domain specific language (DSL), &c.) through a number of simple examples. It is noted that all presented elements and algorithms work in sequential and parallel mode. However, presented examples are very small, and parallel mode will not bring any benefits.

707.1 Elastic Beam Element Under Static Loading

This is a simple beam example under static loading in three directions. The diagram below shows the loading in one bending direction.

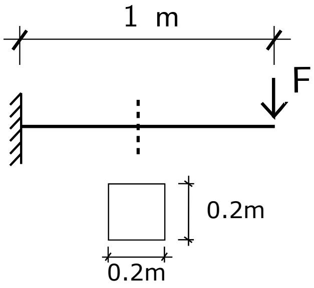


Figure 707.1: The cantilever model

ESSI model fei/DSL file:

```
model name "beam_1element" ;
// define the node coordinates
add node # 1 at ( 0.0*m , 0.0*m, 0.0*m) with 6 dofs;
add node # 2 at ( 1.0*m , 0.0*m, 0.0*m) with 6 dofs;
// Geometry: width and height. Help the beam definition.
b=0.2*m;
h=0.2*m;
I=b*h^3/12.0;
// define the beam element
add element # 1 type beam_elastic with nodes (1,2)
  cross_section = b*h
  elastic_modulus = 1e9*N/m^2
  shear_modulus = 5e8*N/m^2
  torsion_Jx = 0.33*b*h^3
  bending_Iy = I
  bending_Iz = I
  mass_density = 0*kg/m^3
  xz_plane_vector = ( 1, 0, 1)
  joint_1_offset = (0*m, 0*m, 0*m)
  joint_2_offset = (0*m, 0*m, 0*m);
// add boundary condition
fix node # 1 dofs all;
// axial loading
new loading stage "axial";
add load # 1 to node # 2 type linear Fx = 1*N;
define load factor increment 1;
define algorithm With_no_convergence_check ;
define solver ProfileSPD;
simulate 1 steps using static algorithm;
```

```
// bending in one direction
new loading stage "bending1";
remove load # 1;
add load # 2 to node # 2 type linear Fy = 1*N;
define load factor increment 1;
define algorithm With_no_convergence_check ;
define solver ProfileSPD;
simulate 1 steps using static algorithm;
// bending in the other direction
new loading stage "bending2";
remove load # 2;
add load # 3 to node # 2 type linear Fz = 1*N;
define load factor increment 1;
define algorithm With_no_convergence_check ;
define solver ProfileSPD;
simulate 1 steps using static algorithm;

bye;
```

The ESSI model fei/DSL files for this example can be downloaded here.

707.2 Elastic Beam Element under Dynamic Loading

Problem description:

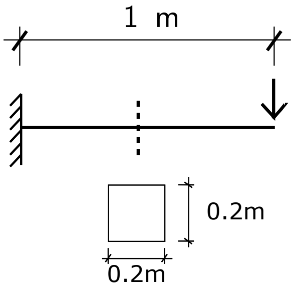


Figure 707.2: The cantilever model.

ESSI model fei/DSL file:

```
model name "beam_1element" ;

// add node
add node # 1 at ( 0.0*m , 0.0*m, 0.0*m) with 6 dofs;
add node # 2 at ( 1.0*m , 0.0*m, 0.0*m) with 6 dofs;
  // Geometry: width and height
b=0.2*m;
h=0.2*m;
// Materials: properties
natural_period = 1*s;
natural_frequency = 2*pi/natural_period;
elastic_constant = 1e9*N/m^2;
I=b*h^3/12.0;
A=b*h;
L=1*m;
rho = (1.8751)^4*elastic_constant*I/(natural_frequency^2*L^4*A);
possion_ratio=0.3;
// add elements
add element # 1 type beam_elastic with nodes (1,2)
  cross_section = b*h
  elastic_modulus = elastic_constant
  shear_modulus = elastic_constant/2/(1+possion_ratio)
  torsion_Jx = 0.33*b*h^3
  bending_Iy = b*h^3/12
  bending_Iz = b*h^3/12
  mass_density = rho
  xz_plane_vector = ( 1, 0, 1)
  joint_1_offset = (0*m, 0*m, 0*m)
```

```
  joint_2_offset = (0*m, 0*m, 0*m);

// add boundary condition
fix node # 1 dofs all;

// // ----------------------------------------------------------------------------
// // --slowLoading---------------------------------------------------------------
// // add load in 180 seconds. (Slow)
// // ----------------------------------------------------------------------------
// new loading stage "slowLoading";
// add load # 1 to node # 2 type path_time_series
// Fz = 1.*N
// series_file = "slowLoading.txt" ;
// define dynamic integrator Newmark with gamma = 0.5 beta = 0.25;
// define algorithm With_no_convergence_check ;
// define solver ProfileSPD;
// simulate 2000 steps using transient algorithm
// time_step = 0.1*s;

// // ----------------------------------------------------------------------------
// // --fastLoading---------------------------------------------------------------
// // add load in 0.6 seconds (Fast)
// // ----------------------------------------------------------------------------
// remove load # 1;
// new loading stage "fastLoading";
// add load # 2 to node # 2 type path_time_series
// Fz = 1.*N
// series_file = "fastLoading.txt" ;
// define dynamic integrator Newmark with gamma = 0.5 beta = 0.25;
// define algorithm With_no_convergence_check ;
// define solver ProfileSPD;
// simulate 1000 steps using transient algorithm
// time_step = 0.01*s;

// // ----------------------------------------------------------------------------
// // --freeVibration-------------------------------------------------------------
// // add a load and then release to free vibration
// // ----------------------------------------------------------------------------
// remove load # 2;
new loading stage "freeVibration";
add load # 3 to node # 2 type path_time_series
  Fz = 1.*N
  series_file = "freeVibration.txt" ;
define dynamic integrator Newmark with gamma = 0.5 beta = 0.25;
define algorithm With_no_convergence_check ;
define solver ProfileSPD;
simulate 2000 steps using transient algorithm
  time_step = 0.01*s;

bye;
```

Displacement Results

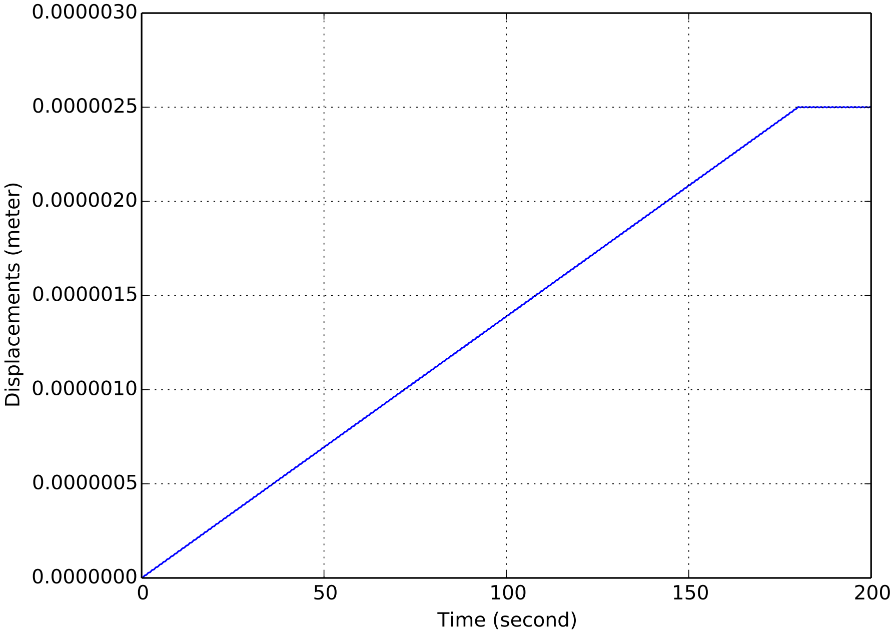


Figure 707.3: Slow loading condition, vertical displacements or the cantilever tip.

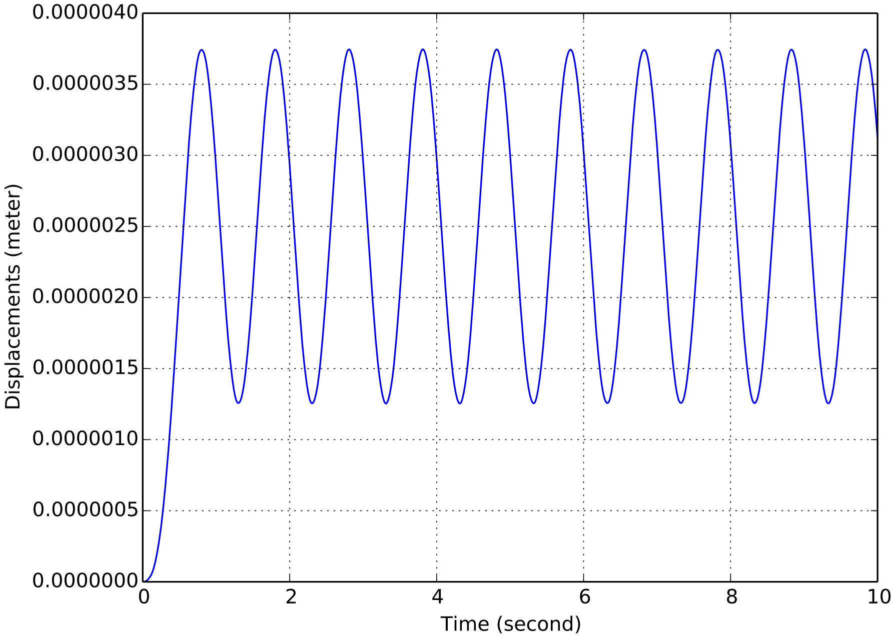


Figure 707.4: Fast loading condition, vertical displacements of the cantilever tip.

The ESSI model fei/DSL files for this example can be downloaded here.

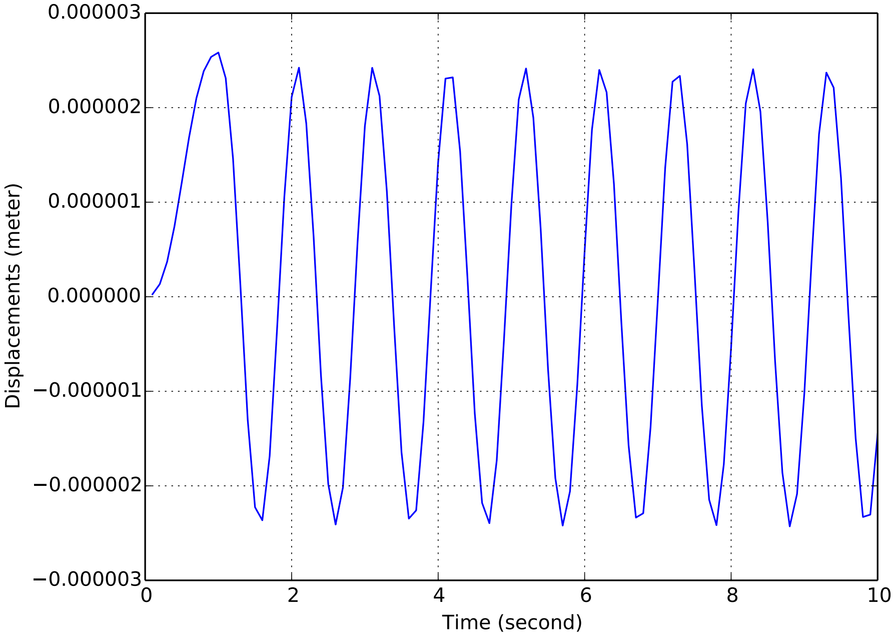


Figure 707.5: Free vibration, vertical displacements of the cantilever tip.

707.3 Cantilever, 5 Elastic Beam Elements

Problem description:

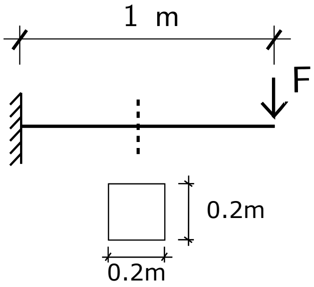


Figure 707.6: The cantilever model.

ESSI model fei/DSL file:

```
model name "beam_5element" ;

// add node
add node # 1 at ( 0.0*m , 0.0*m, 0.0*m) with 6 dofs;
add node # 2 at ( 0.2*m , 0.0*m, 0.0*m) with 6 dofs;
add node # 3 at ( 0.4*m , 0.0*m, 0.0*m) with 6 dofs;
add node # 4 at ( 0.6*m , 0.0*m, 0.0*m) with 6 dofs;
add node # 5 at ( 0.8*m , 0.0*m, 0.0*m) with 6 dofs;
add node # 6 at ( 1.0*m , 0.0*m, 0.0*m) with 6 dofs;

// Geometry: width and height
b=0.2*m;
h=0.2*m;

// Materials: properties
natural_period = 1*s;
natural_frequency = 2*pi/natural_period;
elastic_constant = 1e9*N/m^2;
I=b*h^3/12.0;
A=b*h;
L=1*m;
rho = (1.8751)^4*elastic_constant*I/(natural_frequency^2*L^4*A);
possion_ratio=0.3;

// Cross section geometry: width and height
b=0.2*m;
h=0.2*m;

// add elements
ii=1;
while (ii<6) {
```

```
  add element # ii type beam_elastic with nodes (ii,ii+1)
    cross_section = b*h
    elastic_modulus = elastic_constant
    shear_modulus = elastic_constant/2/(1+possion_ratio)
    torsion_Jx = 0.33*b*h^3
    bending_Iy = b*h^3/12
    bending_Iz = b*h^3/12
    mass_density = rho
    xz_plane_vector = ( 1, 0, 1)
    joint_1_offset = (0*m, 0*m, 0*m)
    joint_2_offset = (0*m, 0*m, 0*m);
  ii+=1;
}

// add boundary condition
fix node # 1 dofs all;

// // ----------------------------------------------------------------------------
// // --slowLoading---------------------------------------------------------------
// // add load in 180 seconds.
// // ----------------------------------------------------------------------------
// new loading stage "slowLoading";
// add load # 1 to node # 6 type path_time_series
// Fz = 1.*N
// series_file = "slowLoading.txt" ;
// define dynamic integrator Newmark with gamma = 0.5 beta = 0.25;
// define algorithm With_no_convergence_check ;
// define solver ProfileSPD;
// simulate 2000 steps using transient algorithm
// time_step = 0.1*s;

// // ----------------------------------------------------------------------------
// // --fastLoading---------------------------------------------------------------
// // add load in 0.6 seconds.
// // ----------------------------------------------------------------------------
// remove load # 1;
// new loading stage "fastLoading";
// add load # 2 to node # 6 type path_time_series
// Fz = 1.*N
// series_file = "fastLoading.txt" ;
// define dynamic integrator Newmark with gamma = 0.5 beta = 0.25;
// define algorithm With_no_convergence_check ;
// define solver ProfileSPD;
// simulate 1000 steps using transient algorithm
// time_step = 0.01*s;

// // ----------------------------------------------------------------------------
// // --freeVibration-------------------------------------------------------------
// // add a load and then release for free vibration
// // ----------------------------------------------------------------------------
// remove load # 2;
```

```
new loading stage "freeVibration";
add load # 3 to node # 6 type path_time_series
  Fz = 1.*N
  series_file = "freeVibration.txt" ;
define dynamic integrator Newmark with gamma = 0.5 beta = 0.25;
define algorithm With_no_convergence_check ;
define solver ProfileSPD;
simulate 100 steps using transient algorithm
  time_step = 0.1*s;

bye;
```

Displacement results

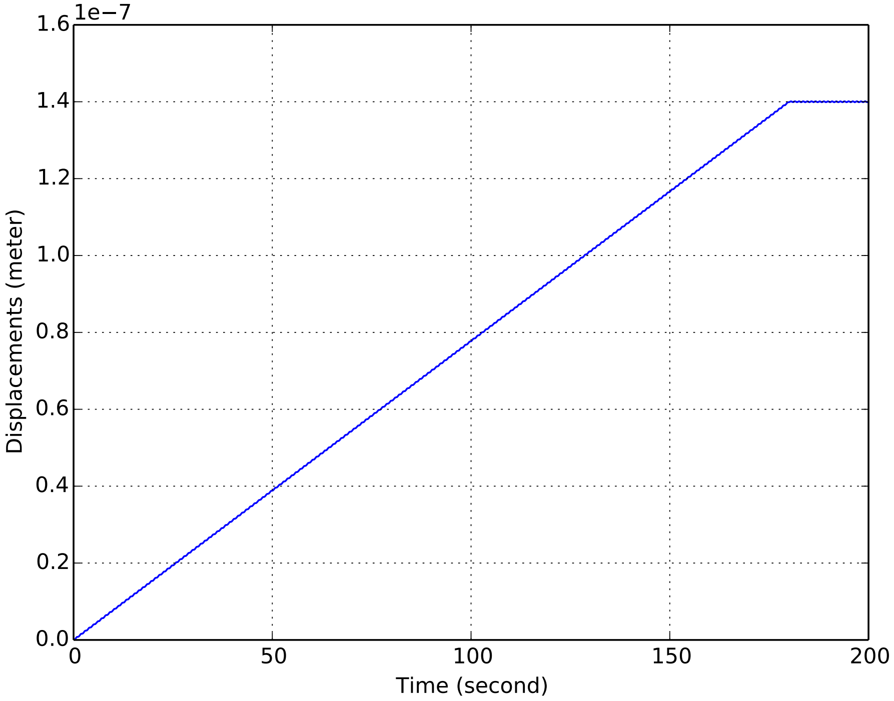


Figure 707.7: Slow loading condition, vertical displacements of the cantilever tip.

The ESSI model fei/DSL files for this example can be downloaded here.

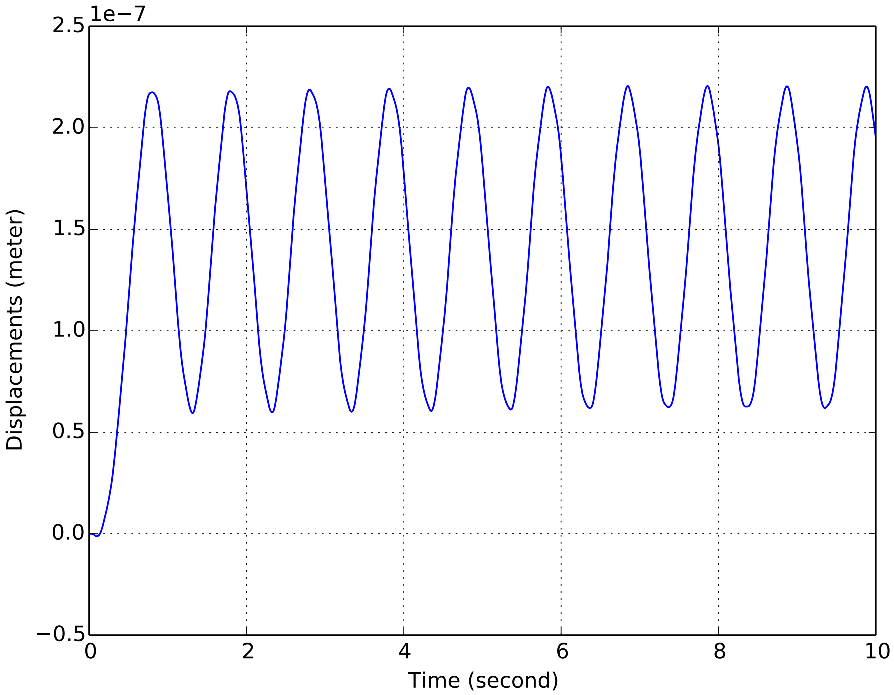


Figure 707.8: Fast loading condition, vertical displacements of the cantilever tip.

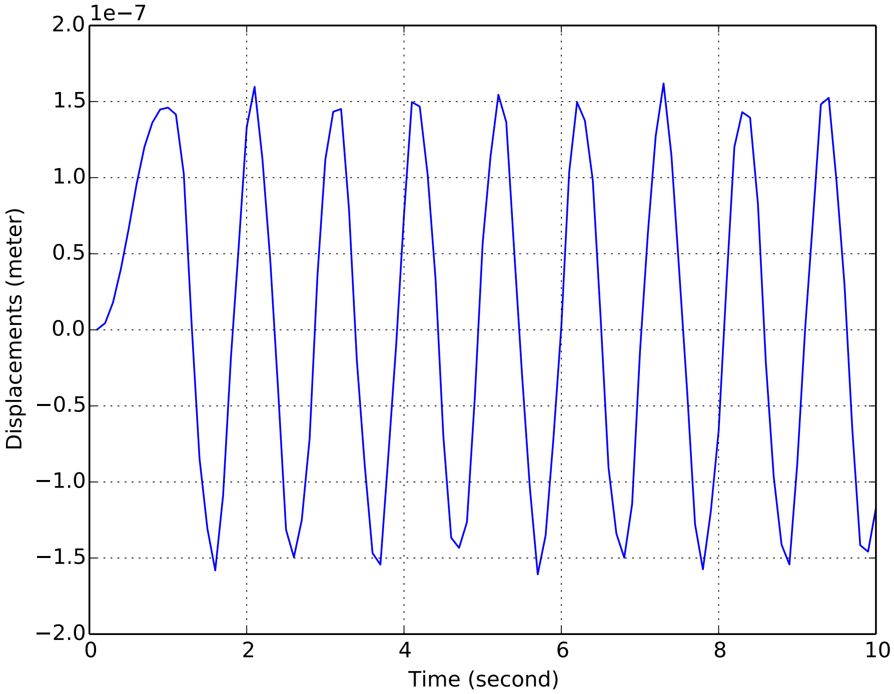


Figure 707.9: Free vibration condition, vertical displacements of the cantilever tip.

707.4 Cantilever, One 27 Node Brick Element, Dynamic Loading

Problem description:

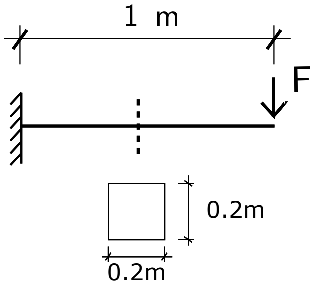


Figure 707.10: The cantilever model.

ESSI model fei/DSL file:

```
model name "brick_1element" ;

// Geometry: width and height
b=0.2*m;
h=0.2*m;

// Materials: properties
natural_period = 1*s;
natural_frequency = 2*pi/natural_period;
elastic_constant = 1e9*N/m^2;
I=b*h^3/12.0;
A=b*h;
L=1*m;
rho = (1.8751)^4*elastic_constant*I/(natural_frequency^2*L^4*A);
possion_ratio=0.3;


add material # 1 type linear_elastic_isotropic_3d_LT
  mass_density = rho
  elastic_modulus = elastic_constant
  poisson_ratio = possion_ratio;

add node # 1 at ( 0.0000 *m, 0.2000 *m, 0.0000 *m) with 3 dofs;
add node # 2 at ( 0.0000 *m, 0.0000 *m, 0.0000 *m) with 3 dofs;
add node # 3 at ( 1.0000 *m, 0.2000 *m, 0.0000 *m) with 3 dofs;
add node # 4 at ( 1.0000 *m, 0.0000 *m, 0.0000 *m) with 3 dofs;
add node # 5 at ( 0.0000 *m, 0.0000 *m, 0.2000 *m) with 3 dofs;
add node # 6 at ( 1.0000 *m, 0.0000 *m, 0.2000 *m) with 3 dofs;
add node # 7 at ( 1.0000 *m, 0.2000 *m, 0.2000 *m) with 3 dofs;
add node # 8 at ( 0.0000 *m, 0.2000 *m, 0.2000 *m) with 3 dofs;
add node # 9 at ( 0.0000 *m, 0.1000 *m, 0.0000 *m) with 3 dofs;
```

```
add node # 10 at ( 0.5000 *m, 0.2000 *m, 0.0000 *m) with 3 dofs;
add node # 11 at ( 1.0000 *m, 0.1000 *m, 0.0000 *m) with 3 dofs;
add node # 12 at ( 0.5000 *m, 0.0000 *m, 0.0000 *m) with 3 dofs;
add node # 13 at ( 0.0000 *m, 0.1000 *m, 0.2000 *m) with 3 dofs;
add node # 14 at ( 0.5000 *m, 0.2000 *m, 0.2000 *m) with 3 dofs;
add node # 15 at ( 1.0000 *m, 0.1000 *m, 0.2000 *m) with 3 dofs;
add node # 16 at ( 0.5000 *m, 0.0000 *m, 0.2000 *m) with 3 dofs;
add node # 17 at ( 0.0000 *m, 0.0000 *m, 0.1000 *m) with 3 dofs;
add node # 18 at ( 0.0000 *m, 0.2000 *m, 0.1000 *m) with 3 dofs;
add node # 19 at ( 1.0000 *m, 0.2000 *m, 0.1000 *m) with 3 dofs;
add node # 20 at ( 1.0000 *m, 0.0000 *m, 0.1000 *m) with 3 dofs;
add node # 21 at ( 0.5000 *m, 0.1000 *m, 0.1000 *m) with 3 dofs;
add node # 22 at ( 0.0000 *m, 0.1000 *m, 0.1000 *m) with 3 dofs;
add node # 23 at ( 0.5000 *m, 0.2000 *m, 0.1000 *m) with 3 dofs;
add node # 24 at ( 1.0000 *m, 0.1000 *m, 0.1000 *m) with 3 dofs;
add node # 25 at ( 0.5000 *m, 0.0000 *m, 0.1000 *m) with 3 dofs;
add node # 26 at ( 0.5000 *m, 0.1000 *m, 0.0000 *m) with 3 dofs;
add node # 27 at ( 0.5000 *m, 0.1000 *m, 0.2000 *m) with 3 dofs;

add element # 1 type 27NodeBrickLT with nodes( 2, 1, 3, 4, 5, 8, 7, 6, 9, 10, ↩
    11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27) use ↩
    material # 1;

fix node # 1 dofs all;
fix node # 2 dofs all;
fix node # 5 dofs all;
fix node # 8 dofs all;
fix node # 9 dofs all;
fix node # 13 dofs all;
fix node # 17 dofs all;
fix node # 18 dofs all;
fix node # 22 dofs all;


// // ----------------------------------------------------------------------------
// // --slowLoading---------------------------------------------------------------
// // ----------------------------------------------------------------------------
// new loading stage "slowLoading";
// add load # 1 to node # 4 type path_time_series Fz=1/36.0*N series_file = ↩
    "slowLoading.txt" ;
// add load # 2 to node # 6 type path_time_series Fz=1/36.0*N series_file = ↩
    "slowLoading.txt" ;
// add load # 3 to node # 3 type path_time_series Fz=1/36.0*N series_file = ↩
    "slowLoading.txt" ;
// add load # 4 to node # 7 type path_time_series Fz=1/36.0*N series_file = ↩
    "slowLoading.txt" ;
// add load # 5 to node # 20 type path_time_series Fz=1/9.0*N series_file = ↩
    "slowLoading.txt" ;
// add load # 6 to node # 11 type path_time_series Fz=1/9.0*N series_file = ↩
    "slowLoading.txt" ;
// add load # 7 to node # 15 type path_time_series Fz=1/9.0*N series_file = ↩
```

```
    "slowLoading.txt" ;
// add load # 8 to node # 19 type path_time_series Fz=1/9.0*N series_file = ↩
    "slowLoading.txt" ;
// add load # 9 to node # 24 type path_time_series Fz=4/9.0*N series_file = ↩
    "slowLoading.txt" ;
// // add algorithm and solver
// define dynamic integrator Newmark with gamma = 0.5 beta = 0.25;
// define algorithm With_no_convergence_check ;
// define solver ProfileSPD;
// simulate 2000 steps using transient algorithm
// time_step = 0.1*s;

// // ----------------------------------------------------------------------------
// // --fastLoading---------------------------------------------------------------
// // ----------------------------------------------------------------------------
// new loading stage "fastLoading";
// add load # 101 to node # 4 type path_time_series Fz=1/36.0*N series_file = ↩
    "fastLoading.txt" ;
// add load # 102 to node # 6 type path_time_series Fz=1/36.0*N series_file = ↩
    "fastLoading.txt" ;
// add load # 103 to node # 3 type path_time_series Fz=1/36.0*N series_file = ↩
    "fastLoading.txt" ;
// add load # 104 to node # 7 type path_time_series Fz=1/36.0*N series_file = ↩
    "fastLoading.txt" ;
// add load # 105 to node # 20 type path_time_series Fz=1/9.0*N series_file = ↩
    "fastLoading.txt" ;
// add load # 106 to node # 11 type path_time_series Fz=1/9.0*N series_file = ↩
    "fastLoading.txt" ;
// add load # 107 to node # 15 type path_time_series Fz=1/9.0*N series_file = ↩
    "fastLoading.txt" ;
// add load # 108 to node # 19 type path_time_series Fz=1/9.0*N series_file = ↩
    "fastLoading.txt" ;
// add load # 109 to node # 24 type path_time_series Fz=4/9.0*N series_file = ↩
    "fastLoading.txt" ;
// // add algorithm and solver
// define dynamic integrator Newmark with gamma = 0.5 beta = 0.25;
// define algorithm With_no_convergence_check ;
// define solver ProfileSPD;
// simulate 1000 steps using transient algorithm
// time_step = 0.01*s;

// // ----------------------------------------------------------------------------
// // ↩
    --freeVibration---------------------------------------------------------------
// // ----------------------------------------------------------------------------
new loading stage "freeVibration";
add load # 201 to node # 4 type path_time_series Fz=1/36.0*N series_file = ↩
    "freeVibration.txt" ;
add load # 202 to node # 6 type path_time_series Fz=1/36.0*N series_file = ↩
    "freeVibration.txt" ;
add load # 203 to node # 3 type path_time_series Fz=1/36.0*N series_file = ↩
```

```
    "freeVibration.txt" ;
add load # 204 to node # 7 type path_time_series Fz=1/36.0*N series_file = ↩
    "freeVibration.txt" ;
add load # 205 to node # 20 type path_time_series Fz=1/9.0*N series_file = ↩
    "freeVibration.txt" ;
add load # 206 to node # 11 type path_time_series Fz=1/9.0*N series_file = ↩
    "freeVibration.txt" ;
add load # 207 to node # 15 type path_time_series Fz=1/9.0*N series_file = ↩
    "freeVibration.txt" ;
add load # 208 to node # 19 type path_time_series Fz=1/9.0*N series_file = ↩
    "freeVibration.txt" ;
add load # 209 to node # 24 type path_time_series Fz=4/9.0*N series_file = ↩
    "freeVibration.txt" ;
// add algorithm and solver
define dynamic integrator Newmark with gamma = 0.5 beta = 0.25;
define algorithm With_no_convergence_check ;
define solver ProfileSPD;
simulate 10000 steps using transient algorithm
  time_step = 0.001*s;

// end
bye;
```

Displacement results against time series

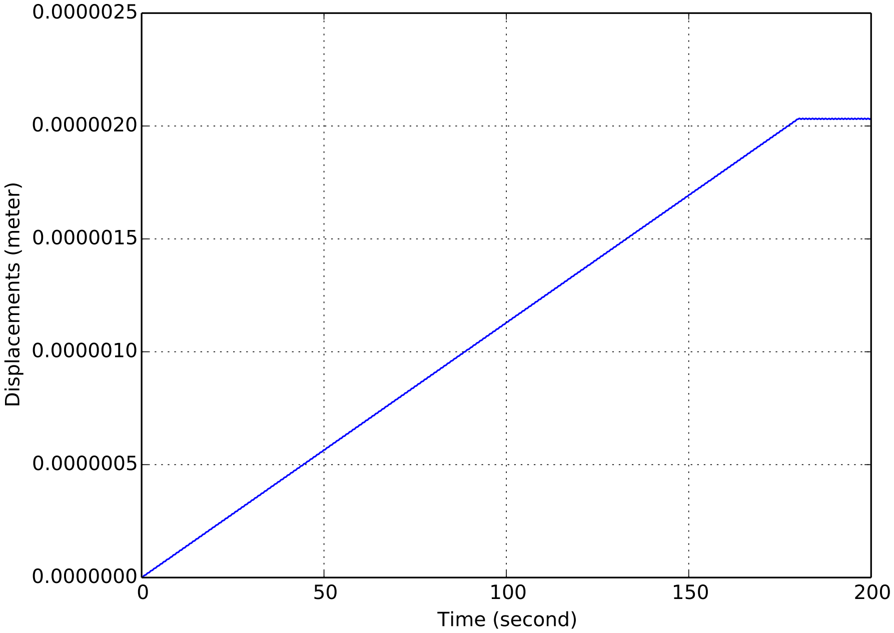


Figure 707.11: Slow loading condition, vertical displacements of the cantilever tip.

The ESSI model fei/DSL files for this example can be downloaded here.

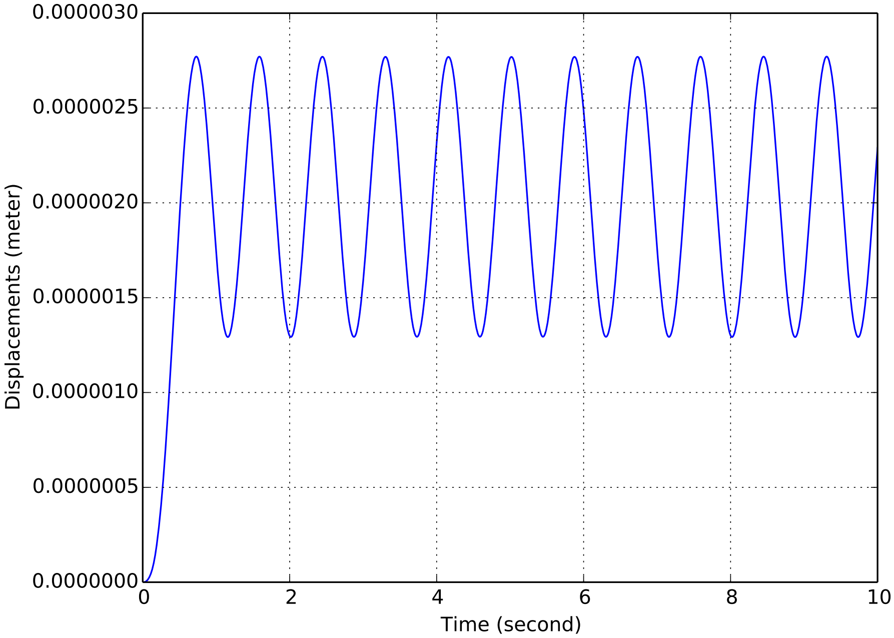


Figure 707.12: Fast loading condition, vertical displacements of the cantilever tip.

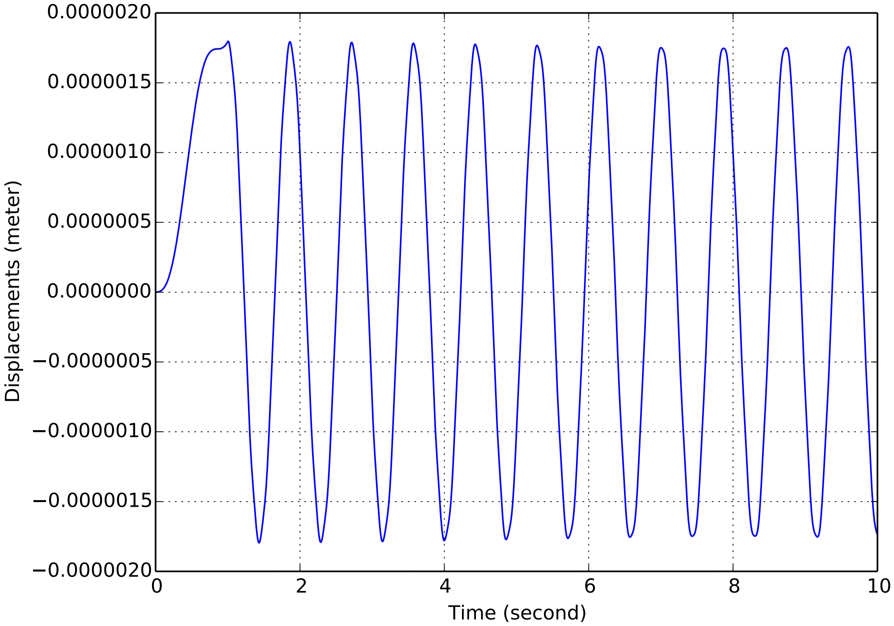


Figure 707.13: Free vibration condition, vertical displacements of the cantilever tip.

707.5 Simulate Cantilever Using Five 27 Node Brick Elements

Problem description:

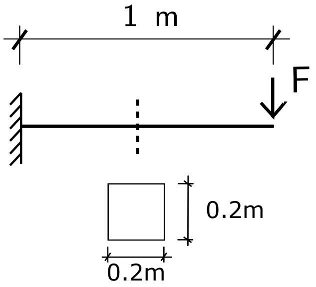


Figure 707.14: The cantilever model.

ESSI model fei/DSL file:

```
model name "brick_5element" ;

// Geometry: width and height
b=0.2*m;
h=0.2*m;

// Materials: properties
natural_period = 1*s;
natural_frequency = 2*pi/natural_period;
elastic_constant = 1e9*N/m^2;
I=b*h^3/12.0;
A=b*h;
L=1*m;
rho = (1.8751)^4*elastic_constant*I/(natural_frequency^2*L^4*A);
possion_ratio=0.3;


add material # 1 type linear_elastic_isotropic_3d_LT
  mass_density = rho
  elastic_modulus = elastic_constant
  poisson_ratio = possion_ratio;

add node # 1 at (0.0*m, 0.0*m , 0.0*m) with 3 dofs;
add node # 2 at (0.1*m, 0.0*m , 0.0*m) with 3 dofs;
add node # 3 at (0.2*m, 0.0*m , 0.0*m) with 3 dofs;
add node # 4 at (0.0*m, 0.1*m , 0.0*m) with 3 dofs;
add node # 5 at (0.1*m, 0.1*m , 0.0*m) with 3 dofs;
add node # 6 at (0.2*m, 0.1*m , 0.0*m) with 3 dofs;
add node # 7 at (0.0*m, 0.2*m , 0.0*m) with 3 dofs;
add node # 8 at (0.1*m, 0.2*m , 0.0*m) with 3 dofs;
add node # 9 at (0.2*m, 0.2*m , 0.0*m) with 3 dofs;
```

```

fix node No 1 dofs ux uy uz;
fix node No 2 dofs ux uy uz;
fix node No 3 dofs ux uy uz;
fix node No 4 dofs ux uy uz;
fix node No 5 dofs ux uy uz;
fix node No 6 dofs ux uy uz;
fix node No 7 dofs ux uy uz;
fix node No 8 dofs ux uy uz;
fix node No 9 dofs ux uy uz;
e = 0;
hh = 0*m;
NBricks=5;
dz = 0.2*m;
while ( e < NBricks)
{
  hh += dz;
  add node # 10+18*e at (0.0*m, 0.0*m , hh - 0.5*dz) with 3 dofs;
  add node # 11+18*e at (0.1*m, 0.0*m , hh - 0.5*dz) with 3 dofs;
  add node # 12+18*e at (0.2*m, 0.0*m , hh - 0.5*dz) with 3 dofs;
  add node # 13+18*e at (0.0*m, 0.1*m , hh - 0.5*dz) with 3 dofs;
  add node # 14+18*e at (0.1*m, 0.1*m , hh - 0.5*dz) with 3 dofs;
  add node # 15+18*e at (0.2*m, 0.1*m , hh - 0.5*dz) with 3 dofs;
  add node # 16+18*e at (0.0*m, 0.2*m , hh - 0.5*dz) with 3 dofs;
  add node # 17+18*e at (0.1*m, 0.2*m , hh - 0.5*dz) with 3 dofs;
  add node # 18+18*e at (0.2*m, 0.2*m , hh - 0.5*dz) with 3 dofs;

  add node # 19+18*e at (0.0*m, 0.0*m , hh) with 3 dofs;
  add node # 20+18*e at (0.1*m, 0.0*m , hh) with 3 dofs;
  add node # 21+18*e at (0.2*m, 0.0*m , hh) with 3 dofs;
  add node # 22+18*e at (0.0*m, 0.1*m , hh) with 3 dofs;
  add node # 23+18*e at (0.1*m, 0.1*m , hh) with 3 dofs;
  add node # 24+18*e at (0.2*m, 0.1*m , hh) with 3 dofs;
  add node # 25+18*e at (0.0*m, 0.2*m , hh) with 3 dofs;
  add node # 26+18*e at (0.1*m, 0.2*m , hh) with 3 dofs;
  add node # 27+18*e at (0.2*m, 0.2*m , hh) with 3 dofs;

  add element # e+1 type 27NodeBrickLT with nodes
    (
      21+18*e,
      27+18*e,
      25+18*e,
      19+18*e,

       3+18*e,
       9+18*e,
       7+18*e,
       1+18*e,

      24+18*e,
      26+18*e,
```

```
     22+18*e,
     20+18*e,

      6+18*e,
      8+18*e,
      4+18*e,
      2+18*e,

     12+18*e,
     18+18*e,
     16+18*e,
     10+18*e,

     14+18*e,
     15+18*e,
     17+18*e,
     13+18*e,
     11+18*e,
     23+18*e,
      5+18*e
    )
    use material # 1;

  e += 1;
};


e = e -1;


// // ------------------------------------------------------------------------------
// // --slowLoading-------------------------------------------------------------------
// // add the 1 Newton load in 180 seconds.
// // ------------------------------------------------------------------------------
// new loading stage "slowLoading";
// add load # 1 to node # (19+18*e) type path_time_series Fx=1/36.0*N ↩
    series_file = "slowLoading.txt";
// add load # 2 to node # (20+18*e) type path_time_series Fx=1/9.0*N ↩
    series_file = "slowLoading.txt";
// add load # 3 to node # (21+18*e) type path_time_series Fx=1/36.0*N ↩
    series_file = "slowLoading.txt";
// add load # 4 to node # (22+18*e) type path_time_series Fx=1/9.0*N ↩
    series_file = "slowLoading.txt";
// add load # 5 to node # (23+18*e) type path_time_series Fx=4/9.0*N ↩
    series_file = "slowLoading.txt";
// add load # 6 to node # (24+18*e) type path_time_series Fx=1/9.0*N ↩
    series_file = "slowLoading.txt";
// add load # 7 to node # (25+18*e) type path_time_series Fx=1/36.0*N ↩
    series_file = "slowLoading.txt";
// add load # 8 to node # (26+18*e) type path_time_series Fx=1/9.0*N ↩
    series_file = "slowLoading.txt";
```

```
// add load # 9 to node # (27+18*e) type path_time_series Fx=1/36.0*N series_file = "slowLoading.txt";
// // add algorithm and solver
// define dynamic integrator Newmark with gamma = 0.5 beta = 0.25;
// define algorithm With_no_convergence_check ;
// define solver ProfileSPD;
// simulate 2000 steps using transient algorithm
// time_step = 0.1*s;

// // -----------------------------------------------------------------------------
// // --fastLoading----------------------------------------------------------------
// // add the 1 Newton load in 0.6 seconds.
// // -----------------------------------------------------------------------------
// new loading stage "fastLoading";
// add load # 101 to node # (19+18*e) type path_time_series Fx=1/36.0*N series_file = "fastLoading.txt" ;
// add load # 102 to node # (20+18*e) type path_time_series Fx=1/9.0*N series_file = "fastLoading.txt" ;
// add load # 103 to node # (21+18*e) type path_time_series Fx=1/36.0*N series_file = "fastLoading.txt" ;
// add load # 104 to node # (22+18*e) type path_time_series Fx=1/9.0*N series_file = "fastLoading.txt" ;
// add load # 105 to node # (23+18*e) type path_time_series Fx=4/9.0*N series_file = "fastLoading.txt" ;
// add load # 106 to node # (24+18*e) type path_time_series Fx=1/9.0*N series_file = "fastLoading.txt" ;
// add load # 107 to node # (25+18*e) type path_time_series Fx=1/36.0*N series_file = "fastLoading.txt" ;
// add load # 108 to node # (26+18*e) type path_time_series Fx=1/9.0*N series_file = "fastLoading.txt" ;
// add load # 109 to node # (27+18*e) type path_time_series Fx=1/36.0*N series_file = "fastLoading.txt" ;
// // add algorithm and solver
// define dynamic integrator Newmark with gamma = 0.5 beta = 0.25;
// define algorithm With_no_convergence_check ;
// define solver ProfileSPD;
// simulate 1000 steps using transient algorithm
// time_step = 0.01*s;

// // -----------------------------------------------------------------------------
// // --freeVibration----------------------------------------------------------------
// // add a load and then release to free vibration
// // -----------------------------------------------------------------------------
new loading stage "freeVibration";
add load # 201 to node # (19+18*e) type path_time_series Fx=1/36.0*N series_file = "freeVibration.txt" ;
add load # 202 to node # (20+18*e) type path_time_series Fx=1/9.0*N series_file = "freeVibration.txt" ;
add load # 203 to node # (21+18*e) type path_time_series Fx=1/36.0*N series_file = "freeVibration.txt" ;
```

```
add load # 204 to node # (22+18*e) type path_time_series Fx=1/9.0*N series_file ↩
    = "freeVibration.txt" ;
add load # 205 to node # (23+18*e) type path_time_series Fx=4/9.0*N series_file ↩
    = "freeVibration.txt" ;
add load # 206 to node # (24+18*e) type path_time_series Fx=1/9.0*N series_file ↩
    = "freeVibration.txt" ;
add load # 207 to node # (25+18*e) type path_time_series Fx=1/36.0*N ↩
    series_file = "freeVibration.txt" ;
add load # 208 to node # (26+18*e) type path_time_series Fx=1/9.0*N series_file ↩
    = "freeVibration.txt" ;
add load # 209 to node # (27+18*e) type path_time_series Fx=1/36.0*N ↩
    series_file = "freeVibration.txt" ;
// add algorithm and solver
define dynamic integrator Newmark with gamma = 0.5 beta = 0.25;
define algorithm With_no_convergence_check ;
define solver ProfileSPD;
simulate 100 steps using transient algorithm
  time_step = 0.1*s;

// end
bye;
```

Displacement Results.

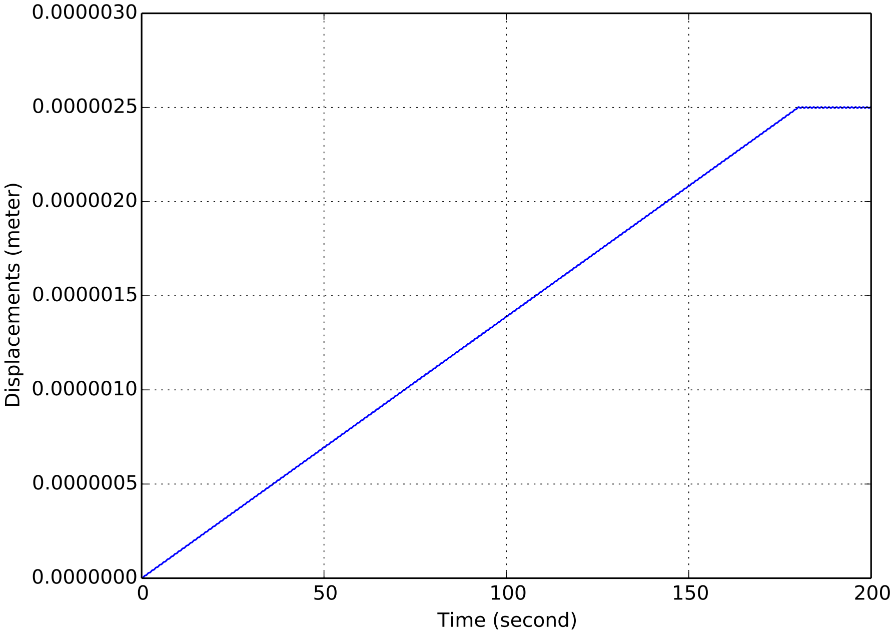


Figure 707.15: Slow loading condition, vertical displacements of the cantilever tip.

The ESSI model fei/DSL files for this example can be downloaded here.

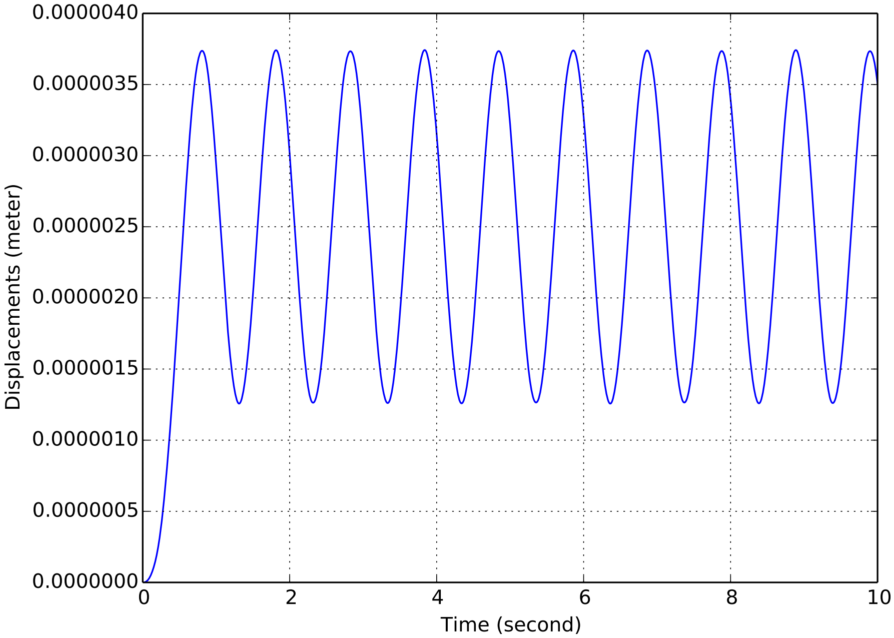


Figure 707.16: Fast loading condition, vertical displacements of the cantilever tip.

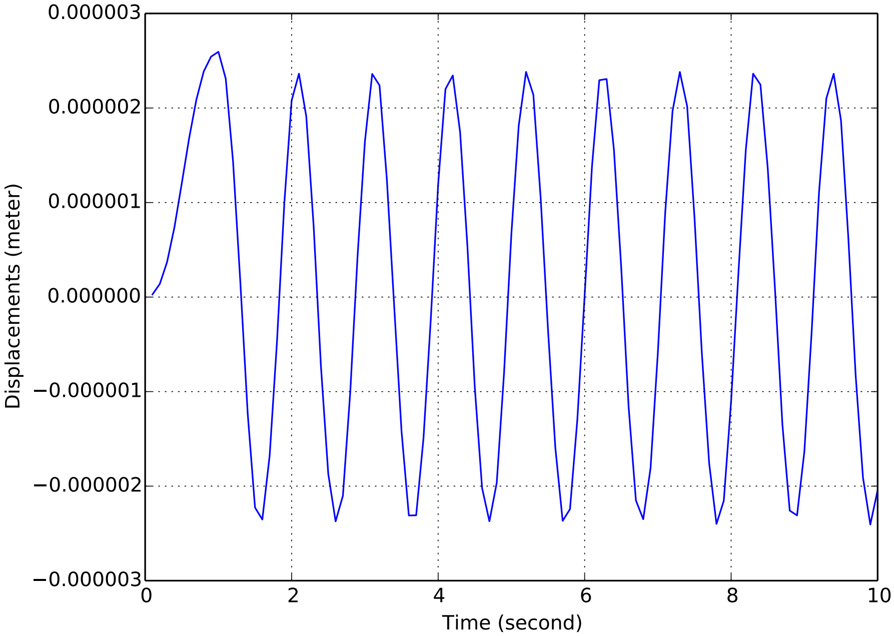


Figure 707.17: Free vibration condition, vertical displacements of the cantilever tip.

707.6 Elastic Beam Element under Dynamic Loading with concentrated mass

Problem description:

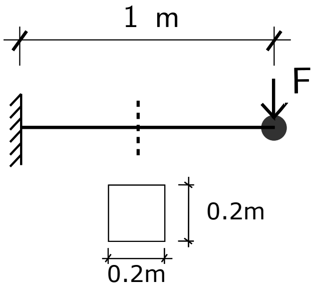


Figure 707.18: The cantilever-mass model.

ESSI model fei/DSL file:

```
model name "beam-mass_1element" ;

// add node
add node # 1 at ( 0.0*m , 0.0*m, 0.0*m) with 6 dofs;
add node # 2 at ( 1.0*m , 0.0*m, 0.0*m) with 6 dofs;

// Geometry: width and height
b=0.2*m;
h=0.2*m;

// Materials: properties
natural_period = 1*s;
natural_frequency = 2*pi/natural_period;
elastic_constant = 1e9*N/m^2;
I=b*h^3/12.0;
A=b*h;
L=1*m;
rho = (1.8751)^4*elastic_constant*I/(natural_frequency^2*L^4*A);
possion_ratio=0.3;

// add elements
add element # 1 type beam_elastic with nodes (1,2)
  cross_section = b*h
  elastic_modulus = elastic_constant
  shear_modulus = elastic_constant/2/(1+possion_ratio)
  torsion_Jx = 0.33*b*h^3
  bending_Iy = b*h^3/12
  bending_Iz = b*h^3/12
```

```
  mass_density = rho
  xz_plane_vector = ( 1, 0, 1)
  joint_1_offset = (0*m, 0*m, 0*m)
  joint_2_offset = (0*m, 0*m, 0*m);

// add boundary condition
fix node # 1 dofs all;

// add mass
beamMass=rho*A*L;
add mass to node # 2
  mx = beamMass
  my = beamMass
  mz = beamMass
  Imx = 0*beamMass*L^2
  Imy = 0*beamMass*L^2
  Imz = 0*beamMass*L^2;

// // ----------------------------------------------------------------------------
// // --slowLoading---------------------------------------------------------------
// // ----------------------------------------------------------------------------
// new loading stage "slowLoading";
// add load # 1 to node # 2 type path_time_series
// Fz = 1.*N
// series_file = "slowLoading.txt" ;
// define dynamic integrator Newmark with gamma = 0.5 beta = 0.25;
// define algorithm With_no_convergence_check ;
// define solver ProfileSPD;
// simulate 2000 steps using transient algorithm
// time_step = 0.1*s;

// // ----------------------------------------------------------------------------
// // --fastLoading---------------------------------------------------------------
// // ----------------------------------------------------------------------------
// remove load # 1;
// new loading stage "fastLoading";
// add load # 2 to node # 2 type path_time_series
// Fz = 1.*N
// series_file = "fastLoading.txt" ;
// define dynamic integrator Newmark with gamma = 0.5 beta = 0.25;
// define algorithm With_no_convergence_check ;
// define solver ProfileSPD;
// simulate 1000 steps using transient algorithm
// time_step = 0.01*s;

// // ----------------------------------------------------------------------------
// // --freeVibration-------------------------------------------------------------
// // ----------------------------------------------------------------------------
// remove load # 2;
new loading stage "freeVibration";
add load # 3 to node # 2 type path_time_series
```

```
  Fz = 1.*N
  series_file = "freeVibration.txt" ;
define dynamic integrator Newmark with gamma = 0.5 beta = 0.25;
define algorithm With_no_convergence_check ;
define solver ProfileSPD;
simulate 1000 steps using transient algorithm
  time_step = 0.01*s;

bye;
```

Displacement results against time series

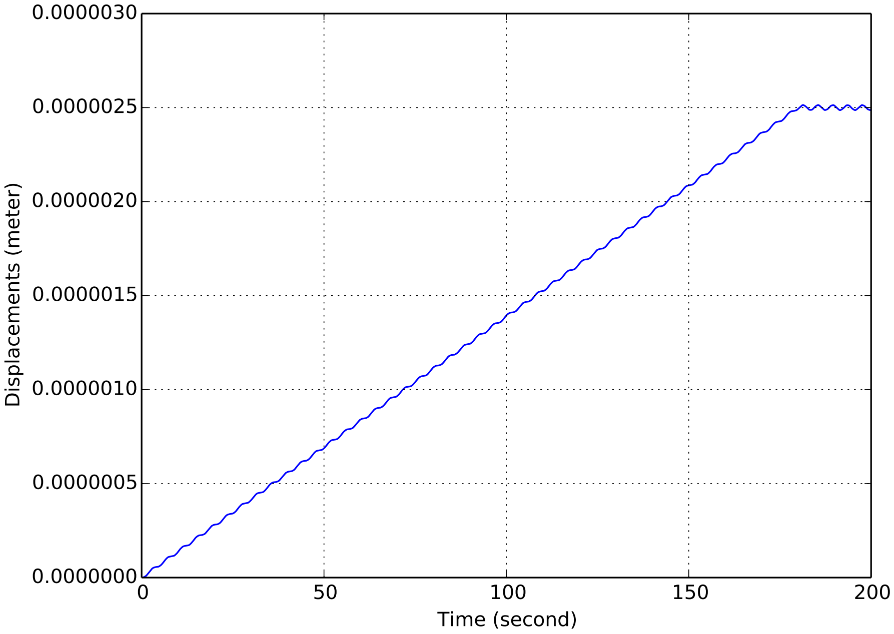


Figure 707.19: Slow loading condition, vertical displacements of the cantilever tip.

The ESSI model fei/DSL files for this example can be downloaded here.

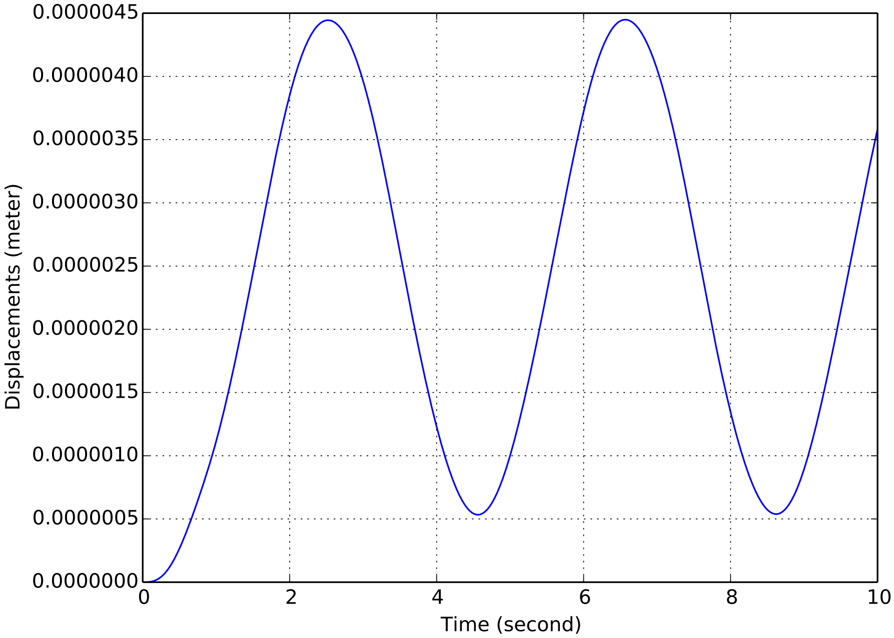


Figure 707.20: Fast loading condition, vertical displacements of the cantilever tip.

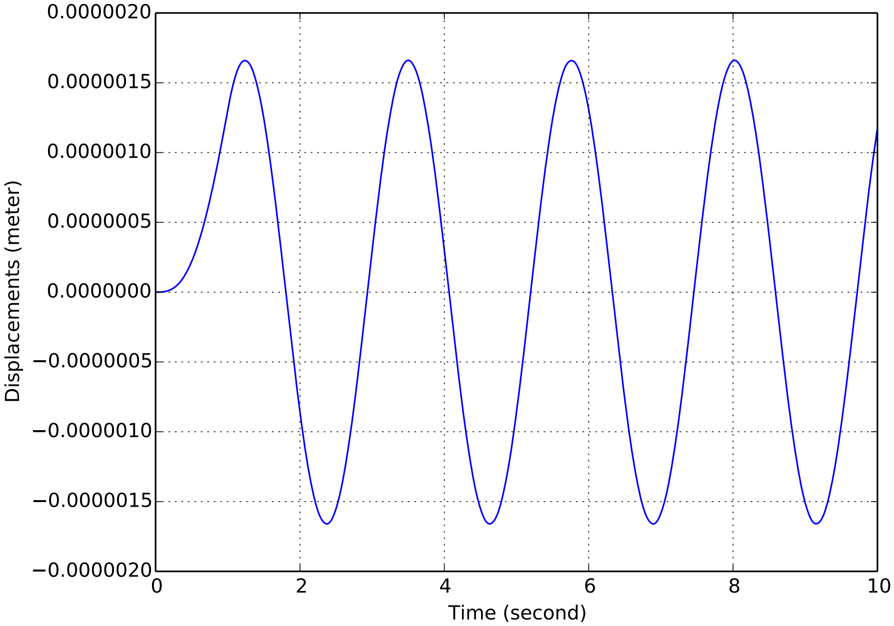


Figure 707.21: Free vibration condition, vertical displacements of the cantilever tip.

707.7 Elastic Beam, 27 Node Brick Model With Concentrated Mass

Problem description:

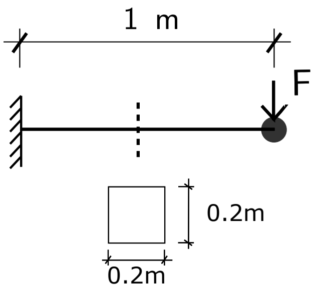


Figure 707.22: The cantilever-mass model.

ESSI model fei/DSL file:

```
model name "brick-mass_1element" ;

// Geometry: width and height
b=0.2*m;
h=0.2*m;

// Materials: properties
natural_period = 1*s;
natural_frequency = 2*pi/natural_period;
elastic_constant = 1e9*N/m^2;
I=b*h^3/12.0;
A=b*h;
L=1*m;
rho = (1.8751)^4*elastic_constant*I/(natural_frequency^2*L^4*A);
possion_ratio=0.3;

add material # 1 type linear_elastic_isotropic_3d_LT
  mass_density = rho
  elastic_modulus = elastic_constant
  poisson_ratio = possion_ratio;

add node # 1 at ( 0.0000 *m, 0.2000 *m, 0.0000 *m) with 3 dofs;
add node # 2 at ( 0.0000 *m, 0.0000 *m, 0.0000 *m) with 3 dofs;
add node # 3 at ( 1.0000 *m, 0.2000 *m, 0.0000 *m) with 3 dofs;
add node # 4 at ( 1.0000 *m, 0.0000 *m, 0.0000 *m) with 3 dofs;
add node # 5 at ( 0.0000 *m, 0.0000 *m, 0.2000 *m) with 3 dofs;
add node # 6 at ( 1.0000 *m, 0.0000 *m, 0.2000 *m) with 3 dofs;
add node # 7 at ( 1.0000 *m, 0.2000 *m, 0.2000 *m) with 3 dofs;
add node # 8 at ( 0.0000 *m, 0.2000 *m, 0.2000 *m) with 3 dofs;
add node # 9 at ( 0.0000 *m, 0.1000 *m, 0.0000 *m) with 3 dofs;
add node # 10 at ( 0.5000 *m, 0.2000 *m, 0.0000 *m) with 3 dofs;
```

```
add node # 11 at ( 1.0000 *m, 0.1000 *m, 0.0000 *m) with 3 dofs;
add node # 12 at ( 0.5000 *m, 0.0000 *m, 0.0000 *m) with 3 dofs;
add node # 13 at ( 0.0000 *m, 0.1000 *m, 0.2000 *m) with 3 dofs;
add node # 14 at ( 0.5000 *m, 0.2000 *m, 0.2000 *m) with 3 dofs;
add node # 15 at ( 1.0000 *m, 0.1000 *m, 0.2000 *m) with 3 dofs;
add node # 16 at ( 0.5000 *m, 0.0000 *m, 0.2000 *m) with 3 dofs;
add node # 17 at ( 0.0000 *m, 0.0000 *m, 0.1000 *m) with 3 dofs;
add node # 18 at ( 0.0000 *m, 0.2000 *m, 0.1000 *m) with 3 dofs;
add node # 19 at ( 1.0000 *m, 0.2000 *m, 0.1000 *m) with 3 dofs;
add node # 20 at ( 1.0000 *m, 0.0000 *m, 0.1000 *m) with 3 dofs;
add node # 21 at ( 0.5000 *m, 0.1000 *m, 0.1000 *m) with 3 dofs;
add node # 22 at ( 0.0000 *m, 0.1000 *m, 0.1000 *m) with 3 dofs;
add node # 23 at ( 0.5000 *m, 0.2000 *m, 0.1000 *m) with 3 dofs;
add node # 24 at ( 1.0000 *m, 0.1000 *m, 0.1000 *m) with 3 dofs;
add node # 25 at ( 0.5000 *m, 0.0000 *m, 0.1000 *m) with 3 dofs;
add node # 26 at ( 0.5000 *m, 0.1000 *m, 0.0000 *m) with 3 dofs;
add node # 27 at ( 0.5000 *m, 0.1000 *m, 0.2000 *m) with 3 dofs;

add element # 1 type 27NodeBrickLT with nodes( 2, 1, 3, 4, 5, 8, 7, 6, 9, 10, ↩
    11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27) use ↩
    material # 1;

fix node # 1 dofs all;
fix node # 2 dofs all;
fix node # 5 dofs all;
fix node # 8 dofs all;
fix node # 9 dofs all;
fix node # 13 dofs all;
fix node # 17 dofs all;
fix node # 18 dofs all;
fix node # 22 dofs all;


// Mapping from 3 dofs to 6 dofs.
add node # 1003 at ( 1.0000 *m, 0.2000 *m, 0.0000 *m) with 6 dofs;
add node # 1004 at ( 1.0000 *m, 0.0000 *m, 0.0000 *m) with 6 dofs;
add node # 1006 at ( 1.0000 *m, 0.0000 *m, 0.2000 *m) with 6 dofs;
add node # 1007 at ( 1.0000 *m, 0.2000 *m, 0.2000 *m) with 6 dofs;
// And connect the nodes at the same location.
add constraint equal dof with master node # 3 and slave node # 1003 dof to ↩
    constrain ux uy uz;
add constraint equal dof with master node # 4 and slave node # 1004 dof to ↩
    constrain ux uy uz;
add constraint equal dof with master node # 6 and slave node # 1006 dof to ↩
    constrain ux uy uz;
add constraint equal dof with master node # 7 and slave node # 1007 dof to ↩
    constrain ux uy uz;

add mass to node # 24 mx = rho*A*L my = rho*A*L mz = rho*A*L;

// add 6 beams to connect the mass
```

```
smallb=0.01*m;
smallh=0.01*m;
smallE = 1e9*N/m^2;
smallnu=0.3;
smallrho=0*kg/m^3;
smallI=smallb*smallh^3/12.0;
add element # 11 type beam_elastic with nodes (1003,1004)
  cross_section = smallb*smallh
  elastic_modulus = smallE
  shear_modulus = smallE/2/(1+smallnu)
  torsion_Jx = 0.33*smallb*smallh^3
  bending_Iy = smallI
  bending_Iz = smallI
  mass_density = smallrho
  xz_plane_vector = ( 1, 0, 1)
  joint_1_offset = (0*m, 0*m, 0*m)
  joint_2_offset = (0*m, 0*m, 0*m);
add element # 12 type beam_elastic with nodes (1003,1006)
  cross_section = smallb*smallh
  elastic_modulus = smallE
  shear_modulus = smallE/2/(1+smallnu)
  torsion_Jx = 0.33*smallb*smallh^3
  bending_Iy = smallI
  bending_Iz = smallI
  mass_density = smallrho
  xz_plane_vector = ( 1, 0, 1)
  joint_1_offset = (0*m, 0*m, 0*m)
  joint_2_offset = (0*m, 0*m, 0*m);
add element # 13 type beam_elastic with nodes (1003,1007)
  cross_section = smallb*smallh
  elastic_modulus = smallE
  shear_modulus = smallE/2/(1+smallnu)
  torsion_Jx = 0.33*smallb*smallh^3
  bending_Iy = smallI
  bending_Iz = smallI
  mass_density = smallrho
  xz_plane_vector = ( 1, 0, 1)
  joint_1_offset = (0*m, 0*m, 0*m)
  joint_2_offset = (0*m, 0*m, 0*m);
add element # 14 type beam_elastic with nodes (1004,1006)
  cross_section = smallb*smallh
  elastic_modulus = smallE
  shear_modulus = smallE/2/(1+smallnu)
  torsion_Jx = 0.33*smallb*smallh^3
  bending_Iy = smallI
  bending_Iz = smallI
  mass_density = smallrho
  xz_plane_vector = ( 1, 0, 1)
  joint_1_offset = (0*m, 0*m, 0*m)
  joint_2_offset = (0*m, 0*m, 0*m);
add element # 15 type beam_elastic with nodes (1004,1007)
```

```
  cross_section = smallb*smallh
  elastic_modulus = smallE
  shear_modulus = smallE/2/(1+smallnu)
  torsion_Jx = 0.33*smallb*smallh^3
  bending_Iy = smallI
  bending_Iz = smallI
  mass_density = smallrho
  xz_plane_vector = ( 1, 0, 1)
  joint_1_offset = (0*m, 0*m, 0*m)
  joint_2_offset = (0*m, 0*m, 0*m);
add element # 16 type beam_elastic with nodes (1006,1007)
  cross_section = smallb*smallh
  elastic_modulus = smallE
  shear_modulus = smallE/2/(1+smallnu)
  torsion_Jx = 0.33*smallb*smallh^3
  bending_Iy = smallI
  bending_Iz = smallI
  mass_density = smallrho
  xz_plane_vector = ( 1, 0, 1)
  joint_1_offset = (0*m, 0*m, 0*m)
  joint_2_offset = (0*m, 0*m, 0*m);


// // ------------------------------------------------------------------------------
// // --slowLoading------------------------------------------------------------------
// // add the 1 Newton load in 180 seconds.
// // ------------------------------------------------------------------------------
// new loading stage "slowLoading";
// add load # 1 to node # 4 type path_time_series Fz=1/36.0*N series_file = "slowLoading.txt" ;
// add load # 2 to node # 6 type path_time_series Fz=1/36.0*N series_file = "slowLoading.txt" ;
// add load # 3 to node # 3 type path_time_series Fz=1/36.0*N series_file = "slowLoading.txt" ;
// add load # 4 to node # 7 type path_time_series Fz=1/36.0*N series_file = "slowLoading.txt" ;
// add load # 5 to node # 20 type path_time_series Fz=1/9.0*N series_file = "slowLoading.txt" ;
// add load # 6 to node # 11 type path_time_series Fz=1/9.0*N series_file = "slowLoading.txt" ;
// add load # 7 to node # 15 type path_time_series Fz=1/9.0*N series_file = "slowLoading.txt" ;
// add load # 8 to node # 19 type path_time_series Fz=1/9.0*N series_file = "slowLoading.txt" ;
// add load # 9 to node # 24 type path_time_series Fz=4/9.0*N series_file = "slowLoading.txt" ;
// // add algorithm and solver
// define dynamic integrator Newmark with gamma = 0.5 beta = 0.25;
// define algorithm With_no_convergence_check ;
// define solver ProfileSPD;
// simulate 2000 steps using transient algorithm
```

```
// time_step = 0.1*s;

// // ----------------------------------------------------------------------------
// // --fastLoading---------------------------------------------------------------
// // add the 1 Newton load in 0.6 seconds.
// // ----------------------------------------------------------------------------
// new loading stage "fastLoading";
// add load # 101 to node # 4 type path_time_series Fz=1/36.0*N series_file = ↩
    "fastLoading.txt" ;
// add load # 102 to node # 6 type path_time_series Fz=1/36.0*N series_file = ↩
    "fastLoading.txt" ;
// add load # 103 to node # 3 type path_time_series Fz=1/36.0*N series_file = ↩
    "fastLoading.txt" ;
// add load # 104 to node # 7 type path_time_series Fz=1/36.0*N series_file = ↩
    "fastLoading.txt" ;
// add load # 105 to node # 20 type path_time_series Fz=1/9.0*N series_file = ↩
    "fastLoading.txt" ;
// add load # 106 to node # 11 type path_time_series Fz=1/9.0*N series_file = ↩
    "fastLoading.txt" ;
// add load # 107 to node # 15 type path_time_series Fz=1/9.0*N series_file = ↩
    "fastLoading.txt" ;
// add load # 108 to node # 19 type path_time_series Fz=1/9.0*N series_file = ↩
    "fastLoading.txt" ;
// add load # 109 to node # 24 type path_time_series Fz=4/9.0*N series_file = ↩
    "fastLoading.txt" ;
// // add algorithm and solver
// define dynamic integrator Newmark with gamma = 0.5 beta = 0.25;
// define algorithm With_no_convergence_check ;
// define solver ProfileSPD;
// simulate 1000 steps using transient algorithm
// time_step = 0.01*s;

// // ----------------------------------------------------------------------------
// // ↩
    --freeVibration---------------------------------------------------------------
// // ----------------------------------------------------------------------------
new loading stage "freeVibration";
add load # 201 to node # 4 type path_time_series Fz=1/36.0*N series_file = ↩
    "freeVibration.txt" ;
add load # 202 to node # 6 type path_time_series Fz=1/36.0*N series_file = ↩
    "freeVibration.txt" ;
add load # 203 to node # 3 type path_time_series Fz=1/36.0*N series_file = ↩
    "freeVibration.txt" ;
add load # 204 to node # 7 type path_time_series Fz=1/36.0*N series_file = ↩
    "freeVibration.txt" ;
add load # 205 to node # 20 type path_time_series Fz=1/9.0*N series_file = ↩
    "freeVibration.txt" ;
add load # 206 to node # 11 type path_time_series Fz=1/9.0*N series_file = ↩
    "freeVibration.txt" ;
add load # 207 to node # 15 type path_time_series Fz=1/9.0*N series_file = ↩
    "freeVibration.txt" ;
```

```
add load # 208 to node # 19 type path_time_series Fz=1/9.0*N series_file = ↩
    "freeVibration.txt" ;
add load # 209 to node # 24 type path_time_series Fz=4/9.0*N series_file = ↩
    "freeVibration.txt" ;
// add algorithm and solver
define dynamic integrator Newmark with gamma = 0.5 beta = 0.25;
define algorithm With_no_convergence_check ;
define solver ProfileSPD;
simulate 100 steps using transient algorithm
  time_step = 0.1*s;

// end
bye;
```

Displacement Results.

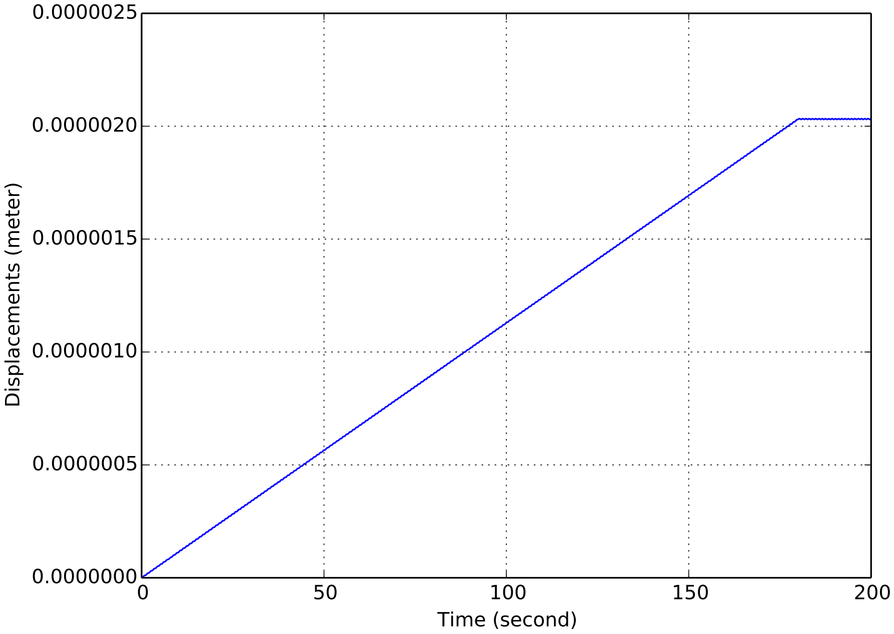


Figure 707.23: Slow loading condition, vertical displacements of the cantilever tip.

The ESSI model fei/DSL files for this example can be downloaded here.

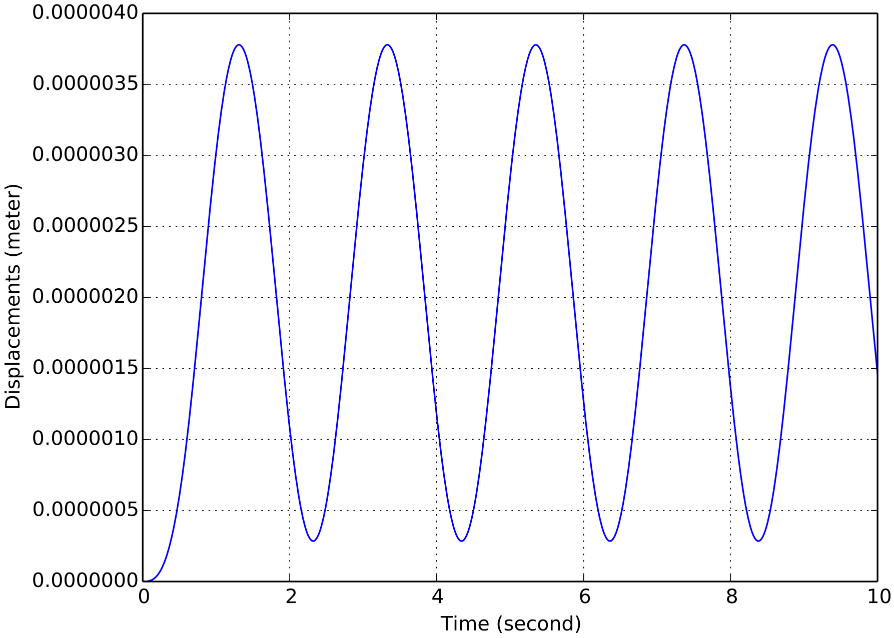


Figure 707.24: Fast loading condition, vertical displacements of the cantilever tip.

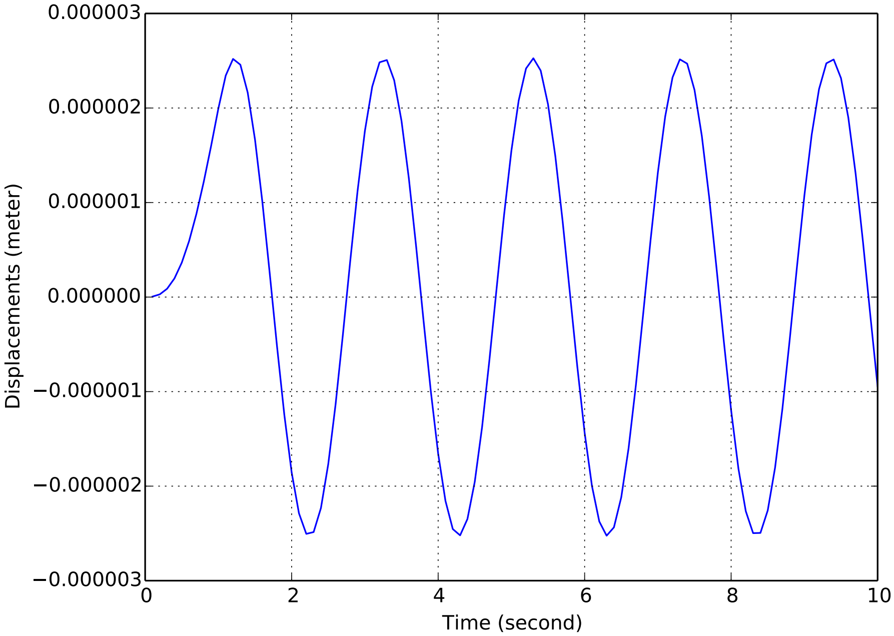


Figure 707.25: Free vibration condition, vertical displacements of the cantilever tip.

707.8 Elastic Beam Element, Dynamic Loading, Viscous (Rayleigh/Caughey) and Numerical (Newmark/HHT) Damping

Problem description:

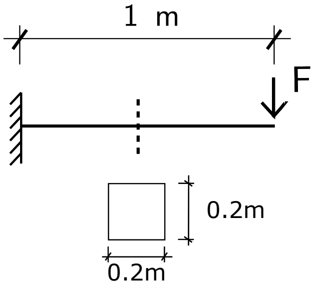


Figure 707.26: The cantilever-mass model.

ESSI model fei/DSL file:

```
model name "beam_1element" ;

// add node
add node # 1 at ( 0.0*m , 0.0*m, 0.0*m) with 6 dofs;
add node # 2 at ( 1.0*m , 0.0*m, 0.0*m) with 6 dofs;

// Geometry: width and height
b=0.2*m;
h=0.2*m;

// Materials: properties
natural_period = 1*s;
natural_frequency = 2*pi/natural_period;
elastic_constant = 1e9*N/m^2;
I=b*h^3/12.0;
A=b*h;
L=1*m;
rho = (1.8751)^4*elastic_constant*I/(natural_frequency^2*L^4*A);
possion_ratio=0.3;

// add elements
add element # 1 type beam_elastic with nodes (1,2)
  cross_section = b*h
  elastic_modulus = elastic_constant
  shear_modulus = elastic_constant/2/(1+possion_ratio)
  torsion_Jx = 0.33*b*h^3
  bending_Iy = b*h^3/12
  bending_Iz = b*h^3/12
  mass_density = rho
```

```
  xz_plane_vector = ( 1, 0, 1)
  joint_1_offset = (0*m, 0*m, 0*m)
  joint_2_offset = (0*m, 0*m, 0*m);

// add boundary condition
fix node # 1 dofs all;

// // ----------------------------------------------------------------------------
// // --no-damping-------------------------------------------------------------
// // ----------------------------------------------------------------------------
// new loading stage "no-damping";
// add load # 1 to node # 2 type path_time_series
// Fz = 1.*N
// series_file = "freeVibration.txt" ;
// define dynamic integrator Newmark with gamma = 0.5 beta = 0.25;
// define algorithm With_no_convergence_check ;
// define solver ProfileSPD;
// simulate 100 steps using transient algorithm
// time_step = 0.1*s;

// // ----------------------------------------------------------------------------
// // ↩
   --Newmark-damping-------------------------------------------------------------
// // ----------------------------------------------------------------------------
// remove load # 2;
// new loading stage "Newmark-damping";
// add load # 3 to node # 2 type path_time_series
// Fz = 1.*N
// series_file = "freeVibration.txt" ;
// define dynamic integrator Newmark with gamma = 0.6 beta = 0.3025;
// define algorithm With_no_convergence_check ;
// define solver ProfileSPD;
// simulate 100 steps using transient algorithm
// time_step = 0.1*s;
// // ----------------------------------------------------------------------------
// // --HHT-damping-------------------------------------------------------------
// // ----------------------------------------------------------------------------
// remove load # 3;
// new loading stage "HHT-damping";
// add load # 4 to node # 6 type path_time_series
// Fz = 1.*kN
// series_file = "freeVibration.txt" ;
// define dynamic integrator Hilber_Hughes_Taylor with alpha = -0.20;
// define algorithm With_no_convergence_check ;
// define solver ProfileSPD;
// simulate 300 steps using transient algorithm
// time_step = 0.1*s;
// // ----------------------------------------------------------------------------
// // ↩
   --Rayleigh-damping-------------------------------------------------------------
// // ----------------------------------------------------------------------------
```

```
// remove load # 4;
// simulate using eigen algorithm number_of_modes = 2;
f1=0.996807/s;
f2=0.996807/s;
w1 = 2*pi*f1;
w2 = 2*pi*f2;
xi=0.05;
rayl_a1 = 2*xi/(w1 + w2);
rayl_a0 = rayl_a1*w1*w2;

add damping # 1 type Rayleigh with
  a0 = rayl_a0
  a1 = rayl_a1
  stiffness_to_use = Initial_Stiffness;
add damping # 1 to element # 1;

new loading stage "Rayleigh-damping";
add load # 5 to node # 2 type path_time_series
  Fz = 1.*N
  series_file = "freeVibration.txt" ;
define dynamic integrator Newmark with gamma = 0.5 beta = 0.25;
define algorithm With_no_convergence_check ;
define solver ProfileSPD;
simulate 100 steps using transient algorithm
  time_step = 0.1*s;

// // ---------------------------------------------------------------------------
// // --Caughey3rd-damping--------------------------------------------------------
// // ---------------------------------------------------------------------------
// add damping # 2 type Caughey3rd with
// a0 = 0.560523/s
// a1 = 0.0730746*s
// a2 = 0.000361559*s^3
// stiffness_to_use = Last_Committed_Stiffness;
// kk=1;
// while (kk<6) {
// add damping # 2 to element # kk;
// kk+=1;
// }
// new loading stage "Caughey3rd-damping";
// add load # 6 to node # 6 type path_time_series
// Fz = 10.*kN
// series_file = "freeVibration.txt" ;
// For Caughey3rd damping, we have to add some Newmark damping,
// Otherwise, there will be some high frequency noise.
// define dynamic integrator Newmark with gamma = 0.6 beta = 0.3025;
// define algorithm With_no_convergence_check ;
// define solver ProfileSPD;
// simulate 100 steps using transient algorithm
// time_step = 0.2*s;

```

```

// // -----------------------------------------------------------------------------
// // --Caughey4th-damping---------------------------------------------------------
// // -----------------------------------------------------------------------------
// add damping # 2 type Caughey4th with
// a0 = 0.560523/s
// a1 = 0.0756472*s
// a2 = 0.000517195*s^3
// a3 = 1.20005*10^(-6)*s^5
// stiffness_to_use = Last_Committed_Stiffness;
// kk=1;
// while (kk<6) {
// add damping # 2 to element # kk;
// kk+=1;
// }
// new loading stage "Caughey4th-damping";
// add load # 6 to node # 6 type path_time_series
// Fz = 10.*kN
// series_file = "freeVibration.txt" ;
// For Caughey4th damping, we have to add some Newmark damping,
// Otherwise, there will be some high frequency noise.
// define dynamic integrator Newmark with gamma = 0.6 beta = 0.3025;
// define algorithm With_no_convergence_check ;
// define solver ProfileSPD;
// simulate 100 steps using transient algorithm
// time_step = 0.2*s;

bye;
```

Displacement results against time series

The ESSI model fei/DSL files for this example can be downloaded here.

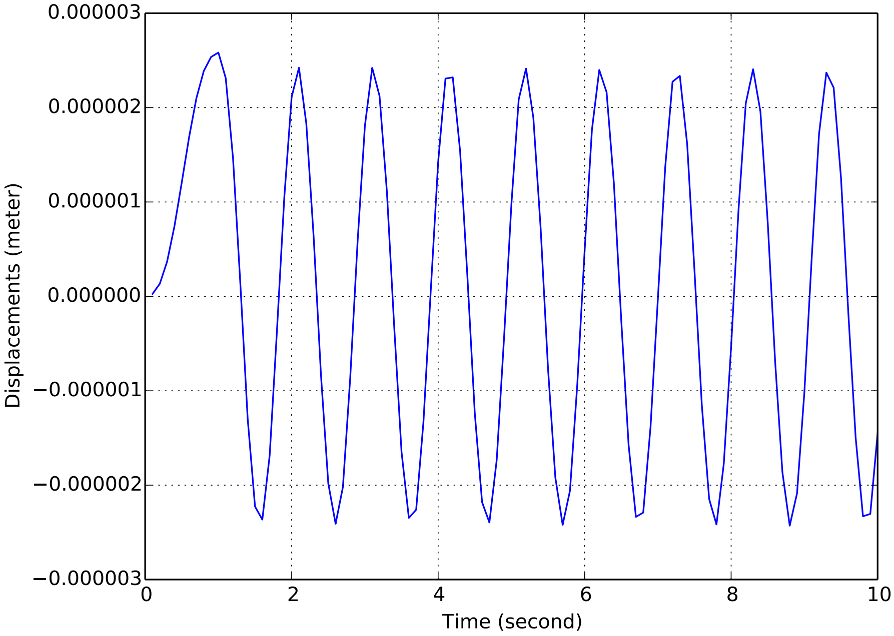


Figure 707.27: Free vibration condition, no damping, vertical displacements of the cantilever tip.

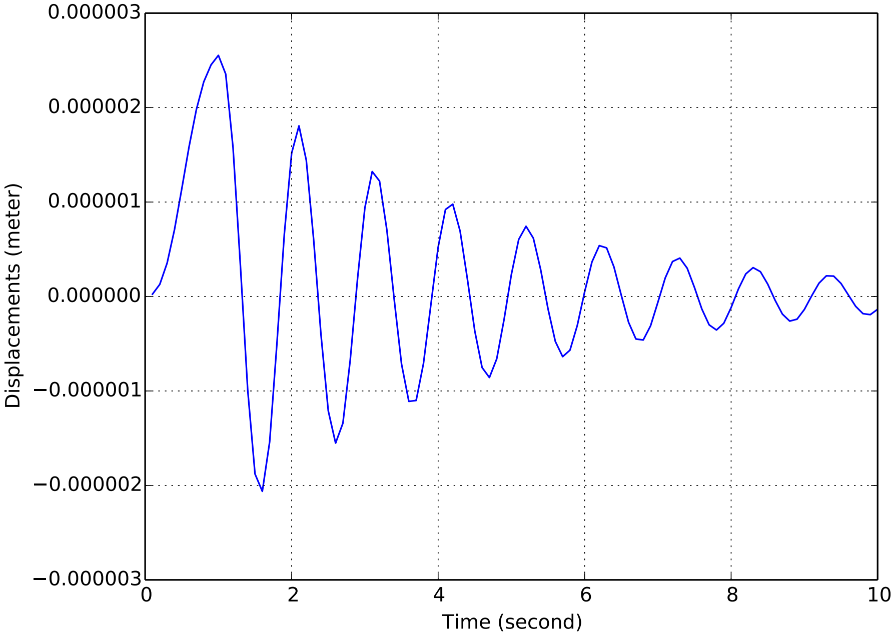


Figure 707.28: Free vibration condition, viscous (Rayleigh) damping, vertical displacements of the cantilever tip.

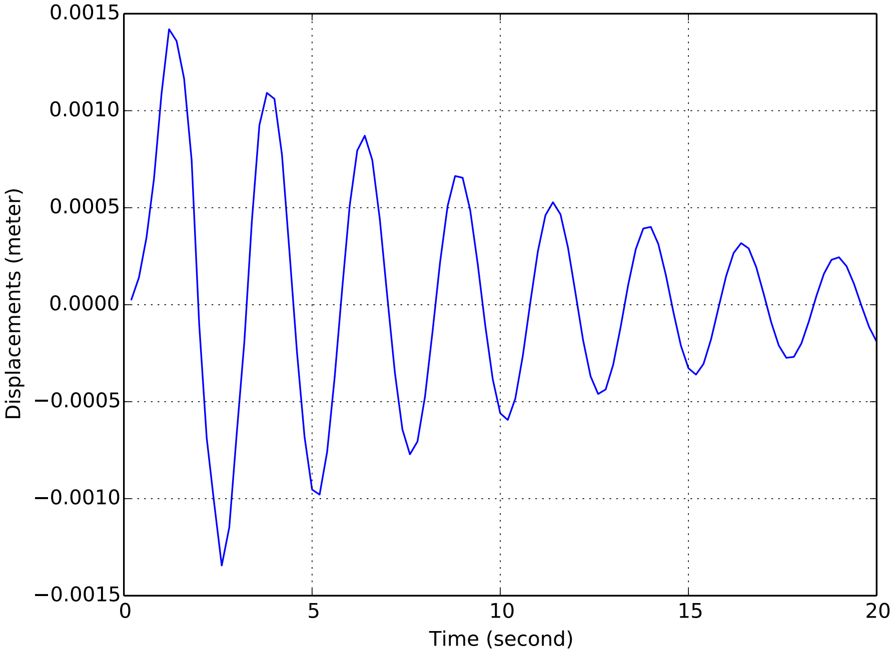


Figure 707.29: Free vibration condition, viscous (Caughey3rd) damping, vertical displacements of the cantilever tip.

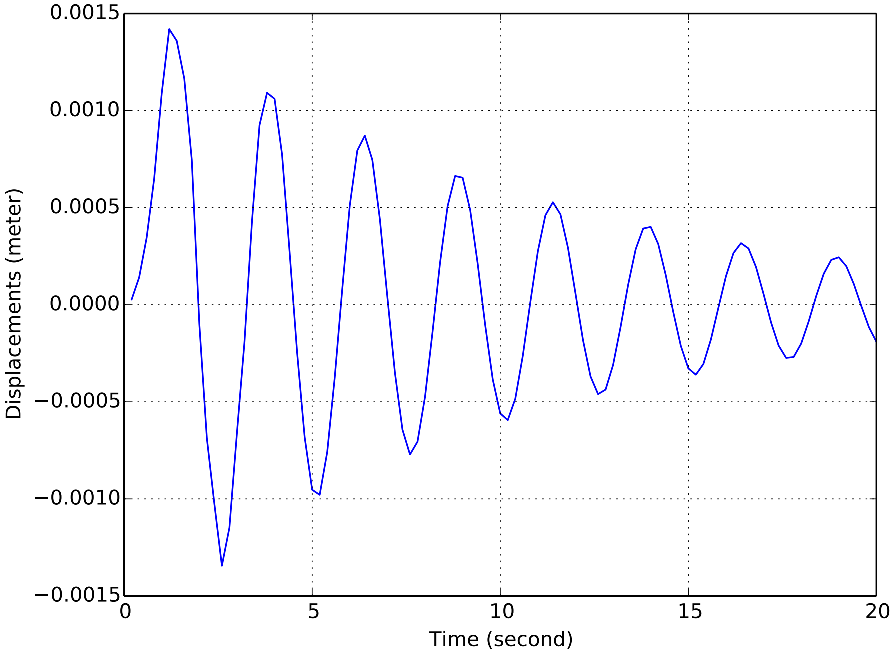


Figure 707.30: Free vibration condition, viscous (Caughey4th) damping, vertical displacements of the cantilever tip.

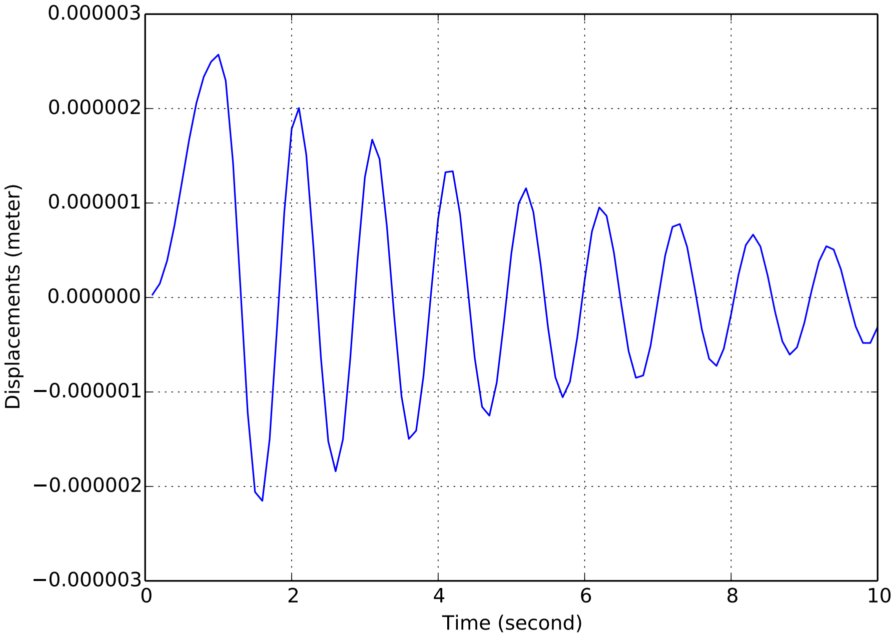


Figure 707.31: Free vibration condition, numerical (Newmark) damping, vertical displacements of the cantilever tip.

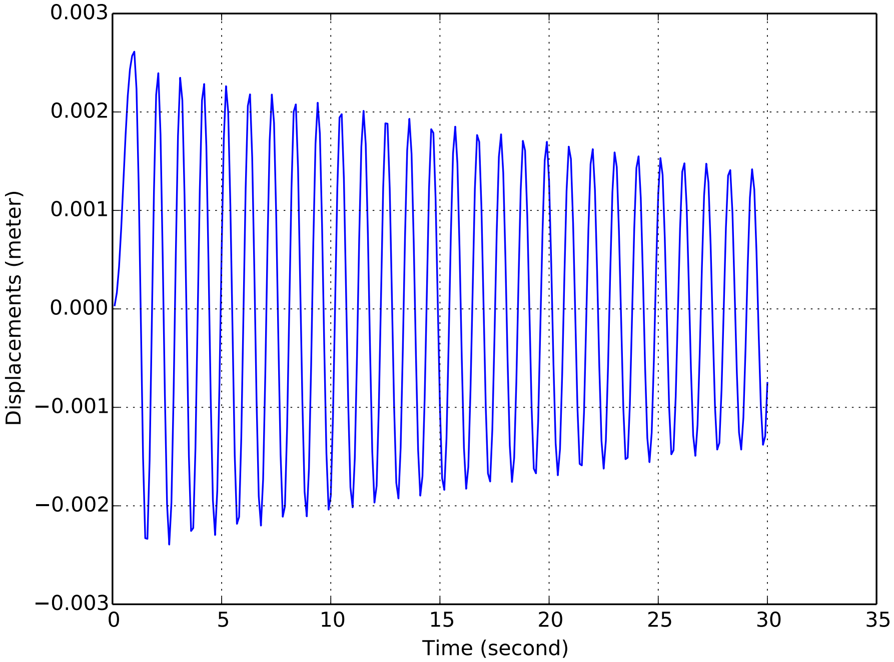


Figure 707.32: Free vibration condition, numerical (HHT) damping, vertical displacements of the cantilever tip.

707.9 Elastic Beam Element for a Simple Frame Structure

Problem Description

- Dimensions: hidth=$6m$, height=$6m$, force=$100N$
- Element dimensions: length=6m, cross section width=1m, cross section height=1m, mass density $\rho = 0.0\mathrm{kN/m}^3$, Young's modulus $E = 1E8$ Pa, Poisson's ratio $\nu = 0.0$.

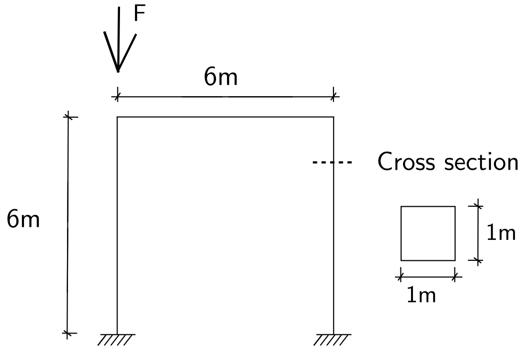


Figure 707.33: Elastic frame with beam_elastic elements.

ESSI model fei/DSL file:

```
model name "beam_element_presentation" ;

add node # 1 at ( 0.00*m, 0.00*m, 0.00*m) with 6 dofs;
add node # 2 at ( 0.00*m, 0.00*m, 6.00*m) with 6 dofs;
add node # 3 at ( 6.00*m, 0.00*m, 6.00*m) with 6 dofs;
add node # 4 at ( 6.00*m, 0.00*m, 0.00*m) with 6 dofs;

elastic_constant = 1e8*N/m^2;
b=1*m;
h=1*m;
rho = 0*kg/m^3; // Mass density

add element # 1 type beam_elastic with nodes (1, 2)
 cross_section = b*h elastic_modulus = elastic_constant
 shear_modulus = elastic_constant/2
 torsion_Jx = 0.33*b*h^3 bending_Iy = b*h^3/12 bending_Iz = h*b^3/12
 mass_density = rho xz_plane_vector = (1, 0, 1 )
   joint_1_offset = (0*m, 0*m, 0*m ) joint_2_offset = (0*m, 0*m, 0*m );

add element # 2 type beam_elastic with nodes (2,3)
```

```
 cross_section = b*h elastic_modulus = elastic_constant
 shear_modulus = elastic_constant/2
 torsion_Jx = 0.33*b*h^3 bending_Iy = b*h^3/12 bending_Iz = h*b^3/12
 mass_density = rho xz_plane_vector = (1, 0, 1 )
   joint_1_offset = (0*m, 0*m, 0*m ) joint_2_offset = (0*m, 0*m, 0*m );

add element # 3 type beam_elastic with nodes (3,4)
 cross_section = b*h elastic_modulus = elastic_constant
 shear_modulus = elastic_constant/2
 torsion_Jx = 0.33*b*h^3 bending_Iy = b*h^3/12 bending_Iz = h*b^3/12
 mass_density = rho xz_plane_vector = (1, 0, 1 )
   joint_1_offset = (0*m, 0*m, 0*m ) joint_2_offset = (0*m, 0*m, 0*m );

fix node #1 dofs all;
fix node #4 dofs all;

new loading stage "Fz";

add load # 1 to node # 2 type linear Fz=50*N;

define algorithm With_no_convergence_check;
define solver ProfileSPD;
define load factor increment 1;
simulate 1 steps using static algorithm;

bye;
```

The ESSI model fei/DSL files for this example can be downloaded here.

707.10 27NodeBrick Cantilever Beam, Static Load

Problem description:

Length=6m, Width=1m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.0$. The force direction is shown in Figure (707.34).

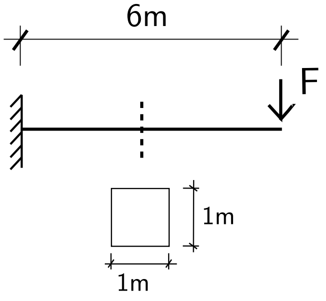


Figure 707.34: Problem description for cantilever beam.

Numerical model:

The 27NodeBrick elements for cantilever beams is shown in Figure (707.35):

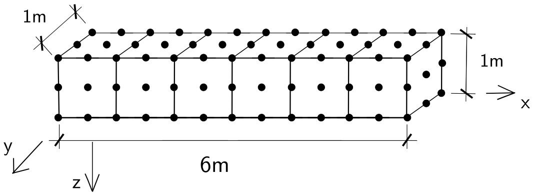


Figure 707.35: 27NodeBrick elements for cantilever beams made of solid elements.

ESSI model fei/DSL file:

```
model name "6meter_cantilever_27brick" ;

add material # 1 type linear_elastic_isotropic_3d
  mass_density = 0*kg/m^3
  elastic_modulus = 1e8*N/m^2
  poisson_ratio = 0.0;

add node # 1 at ( 0.00 *m, 1.00 *m, 0.00 *m) with 3 dofs;
add node # 2 at ( 0.00 *m, 0.00 *m, 0.00 *m) with 3 dofs;
```

```
add node # 3 at ( 6.00 *m, 1.00 *m, 0.00 *m) with 3 dofs;
add node # 4 at ( 5.00 *m, 1.00 *m, 0.00 *m) with 3 dofs;
add node # 5 at ( 4.00 *m, 1.00 *m, 0.00 *m) with 3 dofs;
add node # 6 at ( 3.00 *m, 1.00 *m, 0.00 *m) with 3 dofs;
...
...
add node #117 at ( 5.50 *m, 0.50 *m, 1.00 *m) with 3 dofs;

add element # 1 type 27NodeBrickLT with nodes( 2, 10, 8, 1, 15, 17, 28, 23, 29, ↩
    30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47) use ↩
    material # 1;
add element # 2 type 27NodeBrickLT with nodes( 10, 11, 7, 8, 17, 18, 27, 28, ↩
    48, 49, 50, 30, 51, 52, 53, 34, 38, 54, 55, 39, 56, 57, 58, 59, 43, 60, 61) ↩
    use material # 1;
add element # 3 type 27NodeBrickLT with nodes( 11, 12, 6, 7, 18, 19, 26, 27, ↩
    62, 63, 64, 49, 65, 66, 67, 52, 54, 68, 69, 55, 70, 71, 72, 73, 58, 74, 75) ↩
    use material # 1;
add element # 4 type 27NodeBrickLT with nodes( 12, 13, 5, 6, 19, 20, 25, 26, ↩
    76, 77, 78, 63, 79, 80, 81, 66, 68, 82, 83, 69, 84, 85, 86, 87, 72, 88, 89) ↩
    use material # 1;
add element # 5 type 27NodeBrickLT with nodes( 13, 14, 4, 5, 20, 21, 24, 25, ↩
    90, 91, 92, 77, 93, 94, 95, 80, 82, 96, 97, 83, 98, 99, 100, 101, 86, 102, ↩
    103) use material # 1;
add element # 6 type 27NodeBrickLT with nodes( 14, 9, 3, 4, 21, 16, 22, 24, ↩
    104, 105, 106, 91, 107, 108, 109, 94, 96, 110, 111, 97, 112, 113, 114, 115, ↩
    100, 116, 117) use material # 1;

fix node # 1 dofs all;
fix node # 2 dofs all;
fix node # 15 dofs all;
fix node # 23 dofs all;
fix node # 32 dofs all;
fix node # 36 dofs all;
fix node # 37 dofs all;
fix node # 40 dofs all;
fix node # 45 dofs all;

new loading stage "Fz";
add load # 1 to node # 13 type linear Fz=2.777778*N;
add load # 2 to node # 24 type linear Fz=2.777778*N;
add load # 3 to node # 3 type linear Fz=2.777778*N;
add load # 4 to node # 34 type linear Fz=2.777778*N;
add load # 5 to node # 182 type linear Fz=11.111111*N;
add load # 6 to node # 177 type linear Fz=11.111111*N;
add load # 7 to node # 180 type linear Fz=11.111111*N;
add load # 8 to node # 183 type linear Fz=11.111111*N;
add load # 9 to node # 186 type linear Fz=44.444444*N;

define algorithm With_no_convergence_check ;
define solver UMFPack;
define load factor increment 1;
```

```
simulate 1 steps using static algorithm;

bye;
```

The ESSI model fei/DSL files for this example can be downloaded here.

707.11 4NodeANDES Cantilever Beam, Force Perpendicular to Plane

Problem description:

Length=6m, Width=1m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.0$.

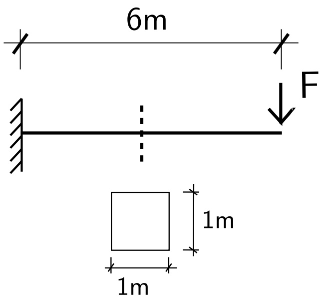


Figure 707.36: Cantilever beams

Numerical model:

For a force direction perpendicular to the plane, only the bending deformation is present.

The model is shown in Figure (707.37).

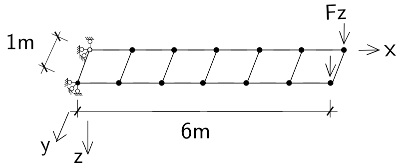


Figure 707.37: 4NodeANDES elements for cantilever beams under force perpendicular to plane.

ESSI model fei/DSL file:

```
 model name "6meter_cantilever_4NodeANDES" ;

add material # 1 type linear_elastic_isotropic_3d
  mass_density = 0*kg/m^3
  elastic_modulus = 1e8*N/m^2
  poisson_ratio = 0.0;

add node # 1 at ( 0.0*m, 0.0*m, 0.0*m) with 6 dofs;
add node # 2 at ( 6.0*m, 0.0*m, 0.0*m) with 6 dofs;
add node # 3 at ( 1.0*m, 0.0*m, 0.0*m) with 6 dofs;
```

```
add node # 4 at ( 2.0*m, 0.0*m, 0.0*m) with 6 dofs;
add node # 5 at ( 3.0*m, 0.0*m, 0.0*m) with 6 dofs;
add node # 6 at ( 4.0*m, 0.0*m, 0.0*m) with 6 dofs;
add node # 7 at ( 5.0*m, 0.0*m, 0.0*m) with 6 dofs;
add node # 8 at ( 6.0*m, 1.0*m, 0.0*m) with 6 dofs;
add node # 9 at ( 0.0*m, 1.0*m, 0.0*m) with 6 dofs;
add node # 10 at ( 5.0*m, 1.0*m, 0.0*m) with 6 dofs;
add node # 11 at ( 4.0*m, 1.0*m, 0.0*m) with 6 dofs;
add node # 12 at ( 3.0*m, 1.0*m, 0.0*m) with 6 dofs;
add node # 13 at ( 2.0*m, 1.0*m, 0.0*m) with 6 dofs;
add node # 14 at ( 1.0*m, 1.0*m, 0.0*m) with 6 dofs;

h = 1*m;
add element # 1 type 4NodeShell_ANDES with nodes (1,3,14,9) use material # 1 ↩
    thickness = h ;
add element # 2 type 4NodeShell_ANDES with nodes (3,4,13,14) use material # 1 ↩
    thickness = h ;
add element # 3 type 4NodeShell_ANDES with nodes (4,5,12,13) use material # 1 ↩
    thickness = h ;
add element # 4 type 4NodeShell_ANDES with nodes (5,6,11,12) use material # 1 ↩
    thickness = h ;
add element # 5 type 4NodeShell_ANDES with nodes (6,7,10,11) use material # 1 ↩
    thickness = h ;
add element # 6 type 4NodeShell_ANDES with nodes (7,2,8,10) use material # 1 ↩
    thickness = h ;

fix node # 1 dofs all ;
fix node # 9 dofs all ;

new loading stage "Fz";
add load # 1 to node # 8 type linear Fz=50*N;
add load # 2 to node # 2 type linear Fz=50*N;

define algorithm With_no_convergence_check ;
define solver ProfileSPD;
define load factor increment 1;
simulate 1 steps using static algorithm;

bye;
```

The ESSI model fei/DSL files for this example can be downloaded here.

707.12 4NodeANDES Cantilever Beams, In-Plane Force

Problem description:

Length=6m, Width=1m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.0$.

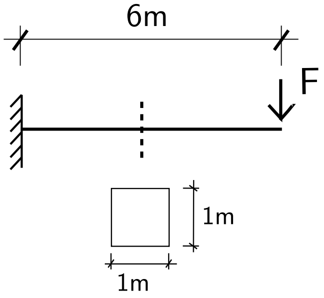


Figure 707.38: Problem description for cantilever beams with in plane force

Numerical model:

The 4NodeANDES elements under in-plane force is shown in Figure (707.39).

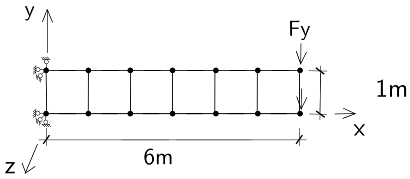


Figure 707.39: 4NodeANDES elements for cantilever beams under in-plane force

ESSI model fei/DSL file:

```
model name "6meter_cantilever_4NodeANDES" ;

add material # 1 type linear_elastic_isotropic_3d
  mass_density = 0*kg/m^3
  elastic_modulus = 1e8*N/m^2
  poisson_ratio = 0.0;

add node # 1 at ( 0.00*m, 0.00*m, 0.00*m) with 6 dofs;
add node # 2 at ( 6.00*m, 0.00*m, 0.00*m) with 6 dofs;
add node # 3 at ( 1.00*m, 0.00*m, 0.00*m) with 6 dofs;
add node # 4 at ( 2.00*m, 0.00*m, 0.00*m) with 6 dofs;
```

```
add node # 5 at ( 3.00*m, 0.00*m, 0.00*m) with 6 dofs;
add node # 6 at ( 4.00*m, 0.00*m, 0.00*m) with 6 dofs;
add node # 7 at ( 5.00*m, 0.00*m, 0.00*m) with 6 dofs;
add node # 8 at ( 6.00*m, 1.00*m, 0.00*m) with 6 dofs;
add node # 9 at ( 0.00*m, 1.00*m, 0.00*m) with 6 dofs;
add node # 10 at ( 5.00*m, 1.00*m, 0.00*m) with 6 dofs;
add node # 11 at ( 4.00*m, 1.00*m, 0.00*m) with 6 dofs;
add node # 12 at ( 3.00*m, 1.00*m, 0.00*m) with 6 dofs;
add node # 13 at ( 2.00*m, 1.00*m, 0.00*m) with 6 dofs;
add node # 14 at ( 1.00*m, 1.00*m, 0.00*m) with 6 dofs;

h = 1*m;
add element # 1 type 4NodeShell_ANDES with nodes (1,3,14,9) use material # 1 ↩
    thickness = h ;
add element # 2 type 4NodeShell_ANDES with nodes (3,4,13,14) use material # 1 ↩
    thickness = h ;
add element # 3 type 4NodeShell_ANDES with nodes (4,5,12,13) use material # 1 ↩
    thickness = h ;
add element # 4 type 4NodeShell_ANDES with nodes (5,6,11,12) use material # 1 ↩
    thickness = h ;
add element # 5 type 4NodeShell_ANDES with nodes (6,7,10,11) use material # 1 ↩
    thickness = h ;
add element # 6 type 4NodeShell_ANDES with nodes (7,2,8,10) use material # 1 ↩
    thickness = h ;

fix node # 1 dofs all;
fix node # 9 dofs all;

new loading stage "Fy";
add load # 1 to node # 8 type linear Fy=50*N;
add load # 2 to node # 2 type linear Fy=50*N;

define algorithm With_no_convergence_check ;
define solver ProfileSPD;
define load factor increment 1;
simulate 1 steps using static algorithm;

bye;
```

The ESSI model fei/DSL files for this example can be downloaded here.

707.13 27NodeBrick Cantilever Beams, Dynamic Input

Problem description:

Length=20m, Width=1m, Height=1m, E=504MPa, $\nu = 0.4$.

All degree of freedoms at the bottom nodes are fixed.

The load is a self weight with a dynamic displacement of supports.

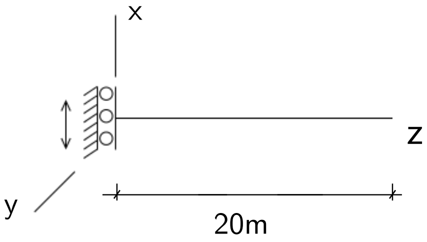


Figure 707.40: Problem description for one simple dynamic example

Numerical model:

The numerical model applied 27NodeBrick to simulate the 1C (1 component) motion.

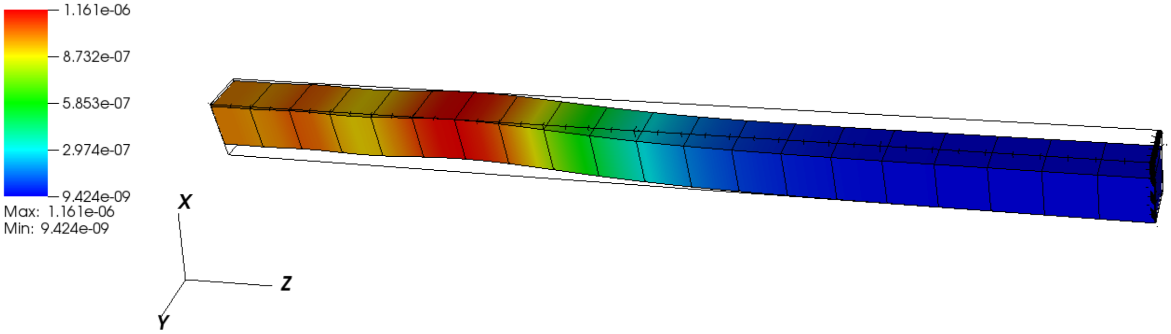


Figure 707.41: Numerical model for one simple dynamic example

ESSI model fei/DSL file:

```
model name "dynamic_example";

add material # 1 type linear_elastic_isotropic_3d_LT
 mass_density = 2000*kg/m^3
```

```
 elastic_modulus = 504000000.00*Pa
 poisson_ratio = 0.4;

add node No 1 at (0*m, 0*m, 0*m) with 3 dofs;
add node No 2 at (0*m, 0.5*m, 0*m) with 3 dofs;
add node No 3 at (0*m, 1*m, 0*m) with 3 dofs;
add node No 4 at (0.5*m, 0*m, 0*m) with 3 dofs;
add node No 5 at (0.5*m, 0.5*m, 0*m) with 3 dofs;
add node No 6 at (0.5*m, 1*m, 0*m) with 3 dofs;
...
...
add node No 369 at (1*m, 1*m, 20*m) with 3 dofs;

add element # 1 type 27NodeBrickLT with nodes ↩
    (27,21,19,25,9,3,1,7,24,20,22,26,6,2,4,8,18,12,10,16,14,15,11,13,17,23,5) ↩
    use material # 1 ;
add element # 2 type 27NodeBrickLT with nodes ↩
    (45,39,37,43,27,21,19,25,42,38,40,44,24,20,22,26,36,30,28,34,32,33,29,31,35,41,23) ↩
    use material # 1 ;
add element # 3 type 27NodeBrickLT with nodes ↩
    (63,57,55,61,45,39,37,43,60,56,58,62,42,38,40,44,54,48,46,52,50,51,47,49,53,59,41) ↩
    use material # 1 ;
add element # 4 type 27NodeBrickLT with nodes ↩
    (81,75,73,79,63,57,55,61,78,74,76,80,60,56,58,62,72,66,64,70,68,69,65,67,71,77,59) ↩
    use material # 1 ;
add element # 5 type 27NodeBrickLT with nodes ↩
    (99,93,91,97,81,75,73,79,96,92,94,98,78,74,76,80,90,84,82,88,86,87,83,85,89,95,77) ↩
    use material # 1 ;
...
...
add element # 20 type 27NodeBrickLT with nodes ↩
    (369,363,361,367,351,345,343,349,366,362,364,368,348,
 344,346,350,360,354,352,358,356,357,353,355,359,365,347) use material # 1 ;

add acceleration field # 1 ax = 0*g ay = 0*g az = -1*g ;
add load # 1 to element # 1 type self_weight use acceleration field # 1;
add load # 2 to element # 2 type self_weight use acceleration field # 1;
add load # 3 to element # 3 type self_weight use acceleration field # 1;
add load # 4 to element # 4 type self_weight use acceleration field # 1;
add load # 5 to element # 5 type self_weight use acceleration field # 1;
add load # 6 to element # 6 type self_weight use acceleration field # 1;
...
...
add load # 20 to element # 20 type self_weight use acceleration field # 1;

fix node No 1 dofs uy uz;
fix node No 2 dofs uy uz;
fix node No 3 dofs uy uz;
fix node No 4 dofs uy uz;
fix node No 5 dofs uy uz;
fix node No 6 dofs uy uz;
```

```
...
...
fix node No 369 dofs uy uz;

zeta = 0.0166667;
fq1 = 3.75;
fq2 = 11.25;
omega1 = 2*pi*fq1;
omega2 = 2*pi*fq2;
zeta1 = zeta;
zeta2 = zeta;
alpha1 = ↩
    2*omega1*omega2*(zeta1*omega2-zeta2*omega1)/(omega2*omega2-omega1*omega1);
beta1 = 2* (zeta2*omega2-zeta1*omega1)/(omega2*omega2-omega1*omega1);
add damping # 1
   type Rayleigh
   with
   a0 = alpha1/s
   a1 = beta1*s
   stiffness_to_use = Initial_Stiffness;

add damping # 1 to element # 1;
add damping # 1 to element # 2;
add damping # 1 to element # 3;
add damping # 1 to element # 4;
add damping # 1 to element # 5;
add damping # 1 to element # 6;
...
...
add damping # 1 to element # 20;

new loading stage "impose_motion";

add imposed motion # 1001 to node # 1 dof ux
 displacement_scale_unit = 1*m displacement_file = "dis.txt"
 velocity_scale_unit = 1*m/s velocity_file = "vel.txt"
 acceleration_scale_unit = 1*m/s^2 acceleration_file = "acc.txt";

add imposed motion # 1002 to node # 2 dof ux
 displacement_scale_unit = 1*m displacement_file = "dis.txt"
 velocity_scale_unit = 1*m/s velocity_file = "vel.txt"
 acceleration_scale_unit = 1*m/s^2 acceleration_file = "acc.txt";

add imposed motion # 1003 to node # 3 dof ux
 displacement_scale_unit = 1*m displacement_file = "dis.txt"
 velocity_scale_unit = 1*m/s velocity_file = "vel.txt"
 acceleration_scale_unit = 1*m/s^2 acceleration_file = "acc.txt";
...
...
add imposed motion # 1009 to node # 9 dof ux
 displacement_scale_unit = 1*m displacement_file = "dis.txt"
```

```
 velocity_scale_unit = 1*m/s velocity_file = "vel.txt"
 acceleration_scale_unit = 1*m/s^2 acceleration_file = "acc.txt";

define dynamic integrator Newmark with gamma = 0.5 beta = 0.25;
define algorithm With_no_convergence_check;
define solver ProfileSPD;
simulate 50 steps using transient algorithm time_step = 0.005*s;

bye;
```

The ESSI model fei/DSL files for this example can be downloaded here.

707.14 4NodeANDES Square Plate, Four Edges Clamped

Problem description:

Length=20m, Width=20m, Height=1m, Force=100N, E=1E8Pa, $\nu = 0.3$.

The four edges are clamped.

The load is a self weight.

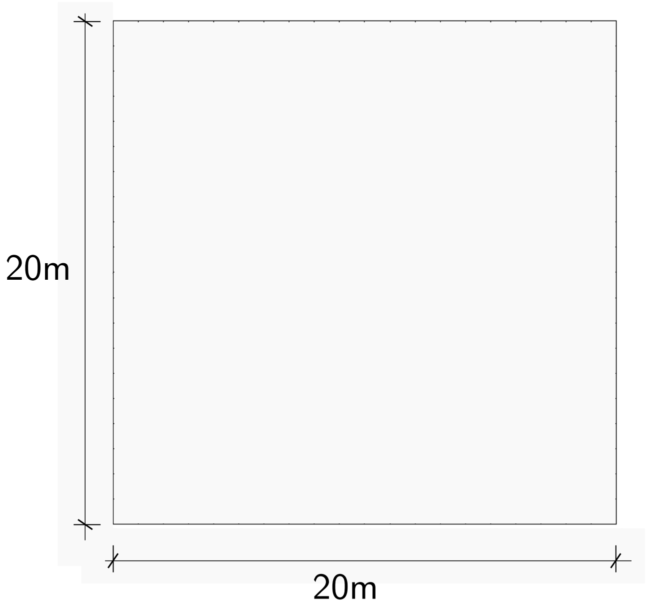


Figure 707.42: Square plate with four edges clamped

Numerical model:

The element side length is 1 meter.

ESSI model fei/DSL file:

```
model name "square_plate" ;

add material # 1 type linear_elastic_isotropic_3d
  mass_density = 1e2*kg/m^3 elastic_modulus = 1e8*N/m^2 poisson_ratio = 0.3;

add node # 1 at ( 0.00*m, 0.00*m, 0.00*m) with 6 dofs;
add node # 2 at ( 20.00*m, 0.00*m, 0.00*m) with 6 dofs;
add node # 3 at ( 1.00*m, 0.00*m, 0.00*m) with 6 dofs;
add node # 4 at ( 2.00*m, 0.00*m, 0.00*m) with 6 dofs;
add node # 5 at ( 3.00*m, 0.00*m, 0.00*m) with 6 dofs;
```

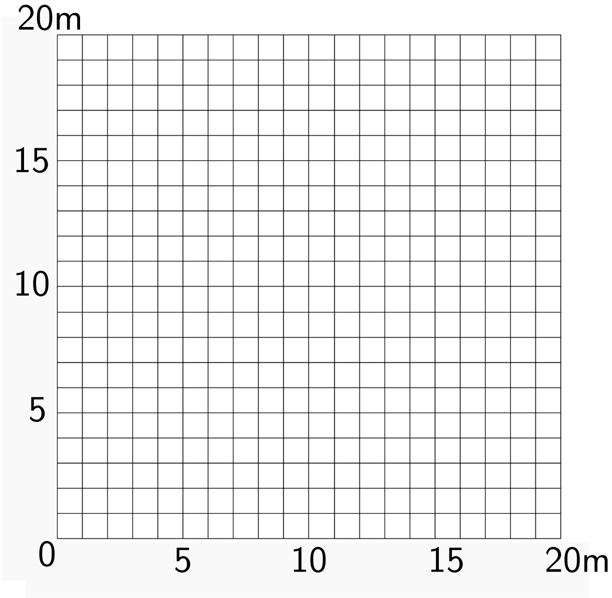


Figure 707.43: 4NodeANDES edge clamped square plate with element side length 1m

```
add node # 6 at ( 4.00*m, 0.00*m, 0.00*m) with 6 dofs;
...
...
add node # 441 at ( 19.00*m, 19.00*m, 0.00*m) with 6 dofs;

h = 1*m;
add element # 1 type 4NodeShell_ANDES with nodes( 1, 3, 81, 80) use material # ↩
    1 thickness=h;
add element # 2 type 4NodeShell_ANDES with nodes( 3, 4, 100, 81) use material # ↩
    1 thickness=h;
add element # 3 type 4NodeShell_ANDES with nodes( 4, 5, 119, 100) use material ↩
    # 1 thickness=h;
add element # 4 type 4NodeShell_ANDES with nodes( 5, 6, 138, 119) use material ↩
    # 1 thickness=h;
add element # 5 type 4NodeShell_ANDES with nodes( 6, 7, 157, 138) use material ↩
    # 1 thickness=h;
add element # 6 type 4NodeShell_ANDES with nodes( 7, 8, 176, 157) use material ↩
    # 1 thickness=h;
...
...
add element # 400 type 4NodeShell_ANDES with nodes( 441, 41, 22, 43) use ↩
    material # 1 thickness=h;


fix node # 1 dofs all ;
fix node # 2 dofs all ;
```

```
fix node # 3 dofs all ;
fix node # 4 dofs all ;
fix node # 5 dofs all ;
fix node # 6 dofs all ;
...
...
fix node # 80 dofs all ;


new loading stage "self_weight";
add acceleration field # 1 ax = 0*g ay = 0*g az = 1*m/s^2;
add load # 1 to element # 1 type self_weight use acceleration field # 1;
add load # 2 to element # 2 type self_weight use acceleration field # 1;
add load # 3 to element # 3 type self_weight use acceleration field # 1;
add load # 4 to element # 4 type self_weight use acceleration field # 1;
add load # 5 to element # 5 type self_weight use acceleration field # 1;
add load # 6 to element # 6 type self_weight use acceleration field # 1;
...
...
add load # 400 to element # 400 type self_weight use acceleration field # 1;


define algorithm With_no_convergence_check ;
define solver ProfileSPD;
define load factor increment 1;
simulate 1 steps using static algorithm;

bye;
```

The ESSI model fei/DSL files for this example can be downloaded here.

707.15 One Dimensional DRM Model

Problem description:

A simple 1D DRM model is shown in Fig.(707.44). The "DRM element", "Exterior node" and "Boundary node" are required to be designated in the DRM HDF5 input. The format and script for the HDF5 input is available in DSL/input manual.

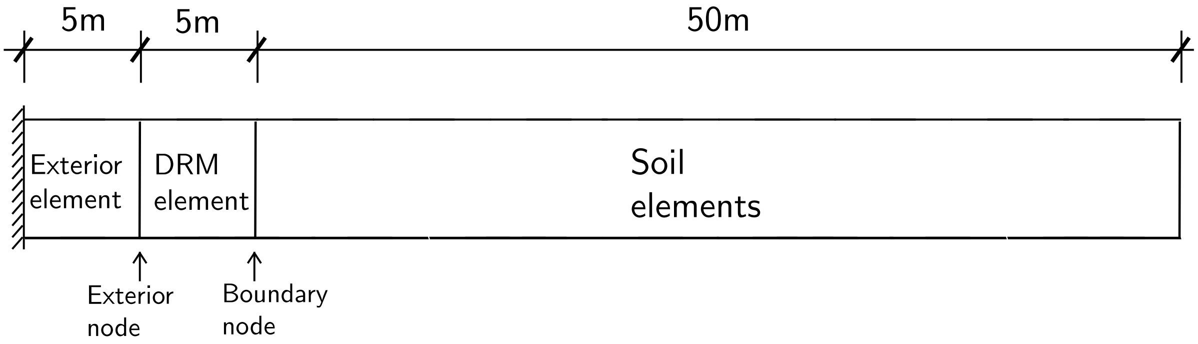


Figure 707.44: 1D DRM model.

Numerical model:

ESSI model fei/DSL file:

```
model name "DRM" ;

//Material for soil
add material # 1 type linear_elastic_isotropic_3d_LT
  mass_density = 2000*kg/m^3
  elastic_modulus = 1300*MPa
  poisson_ratio = 0.3;

//Material for DRM layer
add material # 2 type linear_elastic_isotropic_3d_LT
  mass_density = 2000*kg/m^3
  elastic_modulus = 1300*MPa
  poisson_ratio = 0.3;

//Material for exterior layer
add material # 3 type linear_elastic_isotropic_3d_LT
  mass_density = 2000*kg/m^3
  elastic_modulus = 1300*MPa
  poisson_ratio = 0.3;
//
```

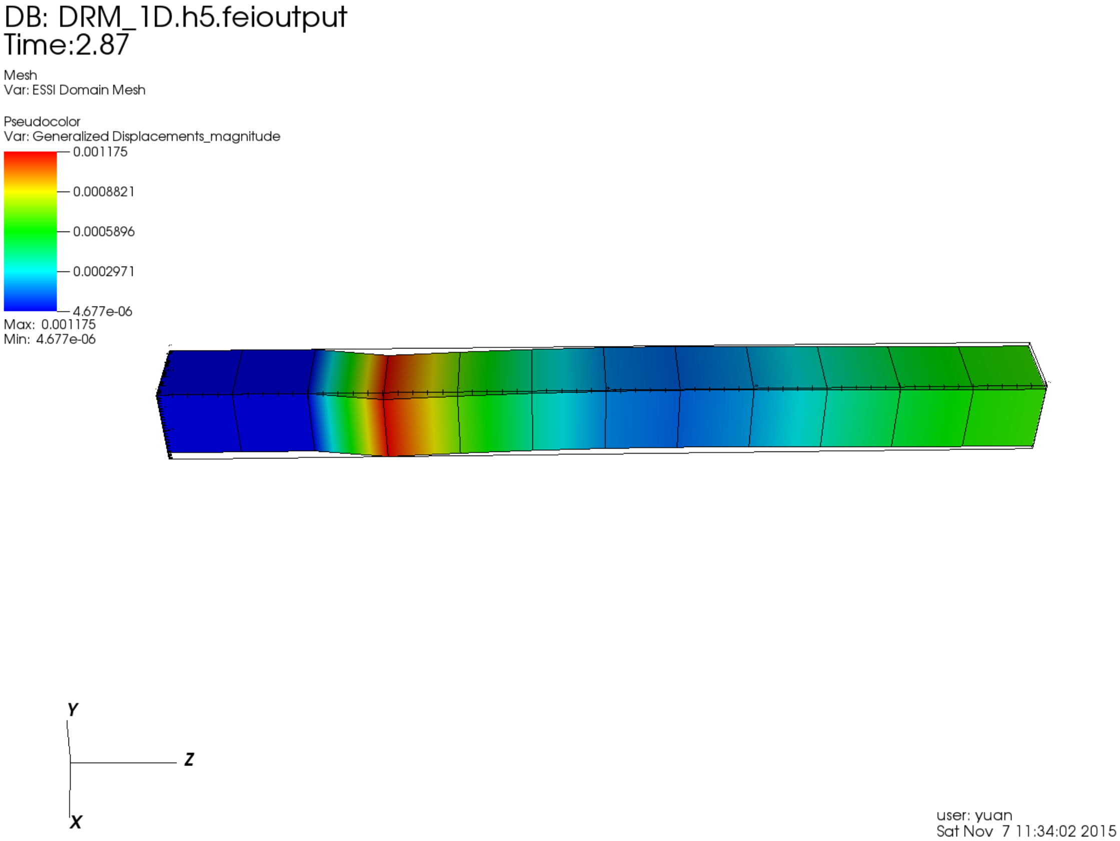


Figure 707.45: 1D DRM model.

```
add node # 1 at ( 0.00*m, 0.00*m, 0.00*m) with 3 dofs;
add node # 2 at ( 5.00*m, 0.00*m, 0.00*m) with 3 dofs;
add node # 3 at ( 5.00*m, 5.00*m, 0.00*m) with 3 dofs;
add node # 4 at ( 0.00*m, 5.00*m, 0.00*m) with 3 dofs;
add node # 5 at ( 5.00*m, 0.00*m, 50.00*m) with 3 dofs;
add node # 6 at ( 5.00*m, 0.00*m, 5.00*m) with 3 dofs;
...
...
add node # 52 at ( 0.00*m, 5.00*m, -5.00*m) with 3 dofs;

//
add element # 1 type 8NodeBrickLT with nodes( 1, 4, 3, 2, 24, 44, 34, 6) use material # 1;
add element # 2 type 8NodeBrickLT with nodes( 24, 44, 34, 6, 23, 43, 33, 7) use material # 1;
...
add element # 12 type 8NodeBrickLT with nodes( 48, 47, 45, 46, 52, 51, 49, 50) use material # 3;

//
fix node # 1 dofs uy ;
fix node # 1 dofs uz ;
```

```
fix node # 2 dofs uy ;
fix node # 2 dofs uz ;
fix node # 3 dofs uy ;
fix node # 3 dofs uz ;
fix node # 4 dofs uy ;
fix node # 4 dofs uz ;
...
fix node # 51 dofs ux ;


new loading stage "1D";
add domain reduction method loading # 1
  hdf5_file = "input.hdf5";

define algorithm With_no_convergence_check ;
define solver ProfileSPD;
define dynamic integrator Newmark with gamma = 0.5 beta = 0.25;
simulate 999 steps using transient algorithm time_step = 0.01*s;

bye;
```

The ESSI model fei/DSL files for this example can be downloaded here.

The same model for this example with 27NodeBrickLT can be downloaded here.

Long 1D DRM model 1000:1

To show the wave propagation explicitly, a long 1D model (1000:1) similar to the 1D DRM model above was made in this section.

The model description is same to Fig.(707.44) except this model use far more soil elements.

The general view is shown in Fig.(707.46) below.

There is still now outgoing waves at the exterior layers, which is shown in Fig(707.47).

The ESSI model fei/DSL files for this example can be downloaded here.

The results can also be seen in this animation.

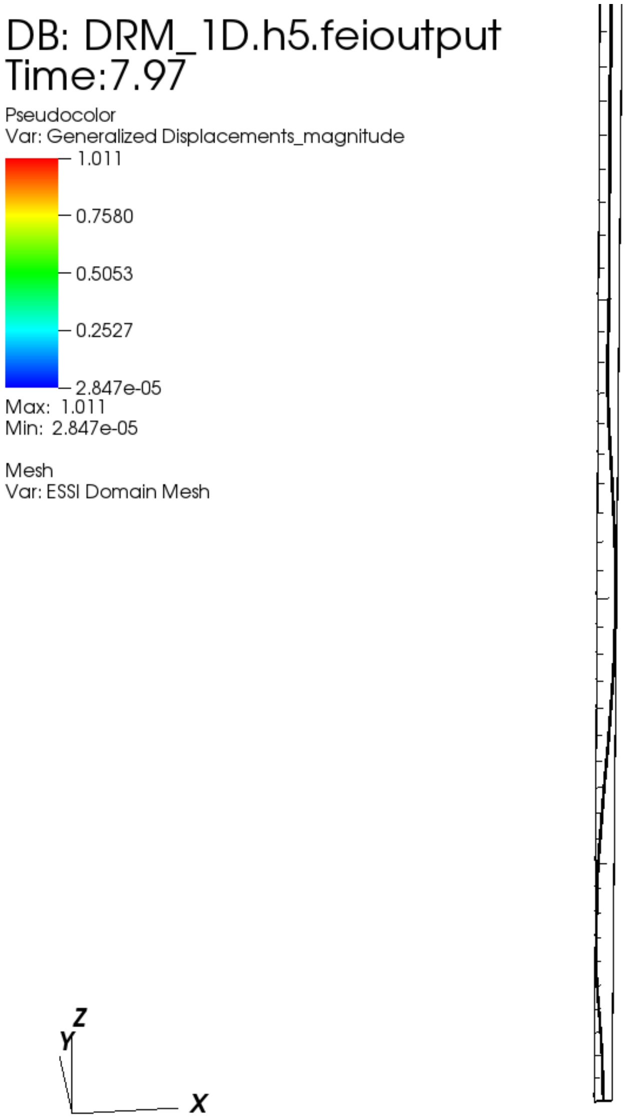


Figure 707.46: Long 1D DRM model

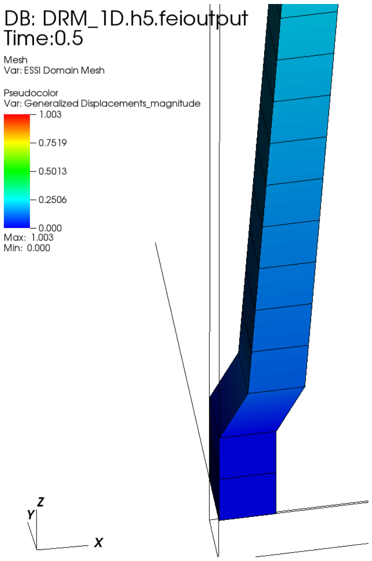


Figure 707.47: Long 1D DRM model: exterior layer

707.16 Three Dimensional DRM Model

Problem description:

As shown in Fig.(707.48), the DRM layer is used to add the earthquake motion.

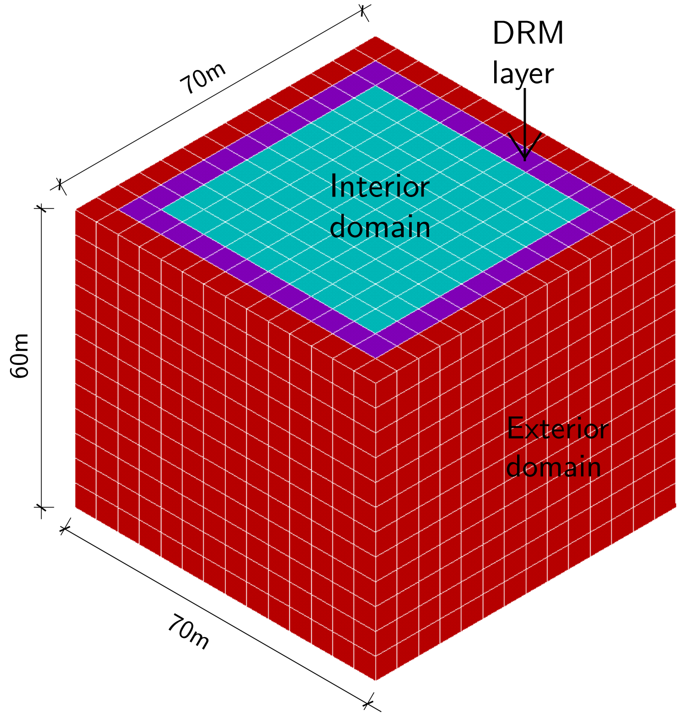


Figure 707.48: The diagram for 3D Domain Reduction Method example.

Numerical result:

ESSI model fei/DSL file:

```
model name "DRM" ;

//Material for soil
add material # 1 type linear_elastic_isotropic_3d_LT
  mass_density = 2000*kg/m^3
  elastic_modulus = 1300*MPa
  poisson_ratio = 0.3;

//Material for DRM layer
add material # 2 type linear_elastic_isotropic_3d_LT
  mass_density = 2000*kg/m^3
  elastic_modulus = 1300*MPa
  poisson_ratio = 0.3;
```

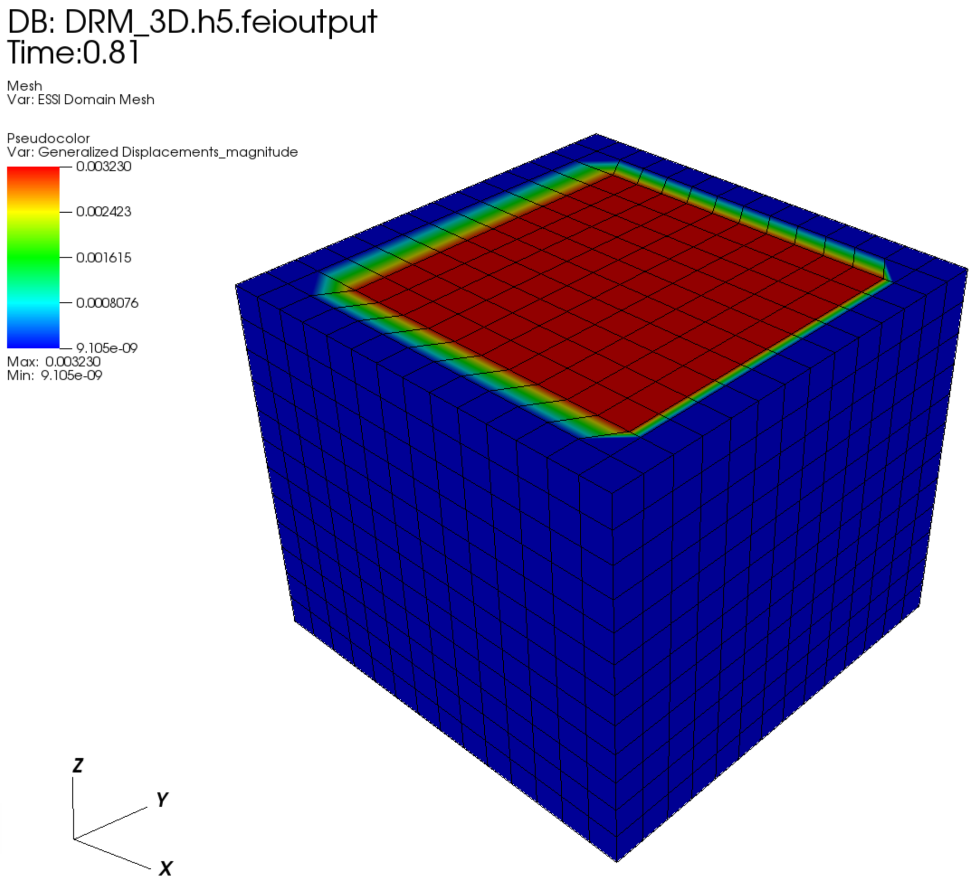


Figure 707.49: Diagram for the 3D DRM model.

```

//Material for exterior layer
add material # 3 type linear_elastic_isotropic_3d_LT
  mass_density = 2000*kg/m^3
  elastic_modulus = 1300*MPa
  poisson_ratio = 0.3;

//
add node # 1 at ( 0.00*m, 0.00*m, 0.00*m) with 3 dofs;
add node # 2 at ( 50.00*m, 0.00*m, 0.00*m) with 3 dofs;
add node # 3 at ( 5.00*m, 0.00*m, 0.00*m) with 3 dofs;
add node # 4 at ( 10.00*m, 0.00*m, 0.00*m) with 3 dofs;
add node # 5 at ( 15.00*m, 0.00*m, 0.00*m) with 3 dofs;
add node # 6 at ( 20.00*m, 0.00*m, 0.00*m) with 3 dofs;
add node # 7 at ( 25.00*m, 0.00*m, 0.00*m) with 3 dofs;
...
...
add node # 2925 at ( 55.00*m, 55.00*m, -5.00*m) with 3 dofs;

//
add element # 1 type 8NodeBrickLT with nodes( 1, 40, 41, 3, 150, 441, 603, 151) ↩
```

```
    use material # 1;
add element # 2 type 8NodeBrickLT with nodes( 3, 41, 50, 4, 151, 603, 684, 160) ↩
    use material # 1;
...
add element # 2352 type 8NodeBrickLT with nodes( 2925, 2924, 2922, 2923, 2921, ↩
    2920, 2918, 2919) use material # 3;

//
fix node # 1332 dofs all ;
fix node # 1334 dofs all ;
...
...
fix node # 2924 dofs all ;

new loading stage "3D";
add domain reduction method loading # 1
  hdf5_file = "input.hdf5";

define algorithm With_no_convergence_check ;
define solver ProfileSPD;
define dynamic integrator Newmark with gamma = 0.5 beta = 0.25;

simulate 999 steps using transient algorithm time_step = 0.01*s;

bye;
```

The ESSI model fei/DSL files for this example can be downloaded here.

The same model for this example with 27NodeBrickLT can be downloaded here.

707.17 ShearBeam Element, Pisano Material

Problem description:

In the element type "ShearBeamLT", only one Gauss point exists. ShearBeamLT element was used here to test the Pisanó material model.

Vertical force F_z was used to apply confinement to the element. Then, cyclic force F_x is used to load. point.

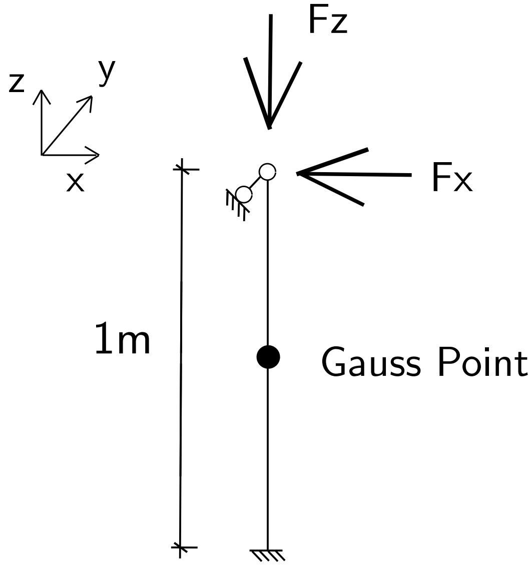


Figure 707.50: ShearBeam element.

Results

Resulting stress-strain relationship is shown in Fig.(707.51).

ESSI model fei/DSL file:

```
model name "pisanoLT";

add node # 1 at (0*m,0*m,0*m) with 3 dofs;
add node # 2 at (0*m,0*m,1*m) with 3 dofs;

fix node # 1 dofs all;
```

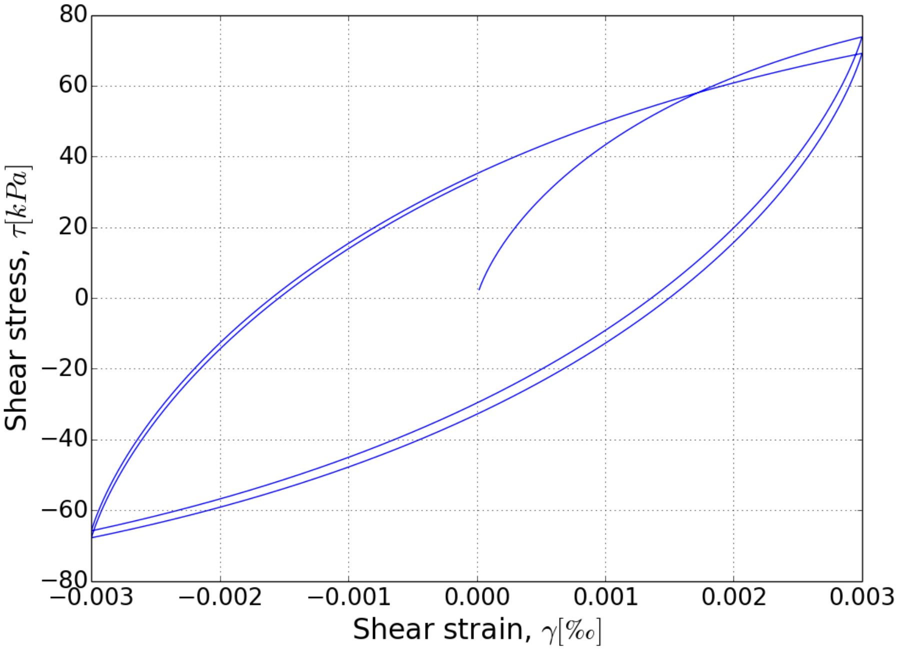


Figure 707.51: Shear stress-strain response.

```
fix node # 2 dofs uy;

add material # 1 type New_PisanoLT
 mass_density = 2000*kg/m^3
 elastic_modulus_1atm = 325*MPa poisson_ratio = 0.3
 M_in = 1.4 kd_in = 0.0 xi_in = 0.0 h_in = 700 m_in = 0.7
 initial_confining_stress = 0*kPa n_in = 0 a_in = 0.0 eplcum_cr_in = 1e-6;

add element # 1 type ShearBeamLT with nodes (1, 2) \
     cross_section = 1*m^2 use material # 1;

new loading stage "confinement";

add load # 1 to node # 2 type linear Fz = -200*kN;
define load factor increment 0.01;
define algorithm With_no_convergence_check ;
define solver UMFPack;
simulate 100 steps using static algorithm;

new loading stage "test01";
gamma_max = 3e-3;
add imposed motion # 2 to node # 2 dof ux
 displacement_scale_unit = gamma_max*m displacement_file = "input_sine.txt"
 velocity_scale_unit = gamma_max*m/s velocity_file = "input_sine.txt"
 acceleration_scale_unit = gamma_max*m/s^2 acceleration_file = "input_sine.txt";

define load factor increment 0.0005;
```

```
define algorithm With_no_convergence_check;
define solver UMFPack;
simulate 2000 steps using static algorithm;

bye;
```

The ESSI model fei/DSL files for this example can be downloaded here.

707.18 8NodeBrickLT Element, Drucker-Prager Material, Armstrong-Frederick Rotational Kinematic Hardening

Problem description:

This example is used to test the materials properties, such as G/Gmax against strains. The element type is 8NodeBrickLT. And there are two stages of loading. The first loading stage is confinement and the second loading stage is shearing.

The boundary condition is specially designed such that each Gauss point has the same stress state.

Results

Resulting stress-strain relationship is shown in Fig.(707.52).

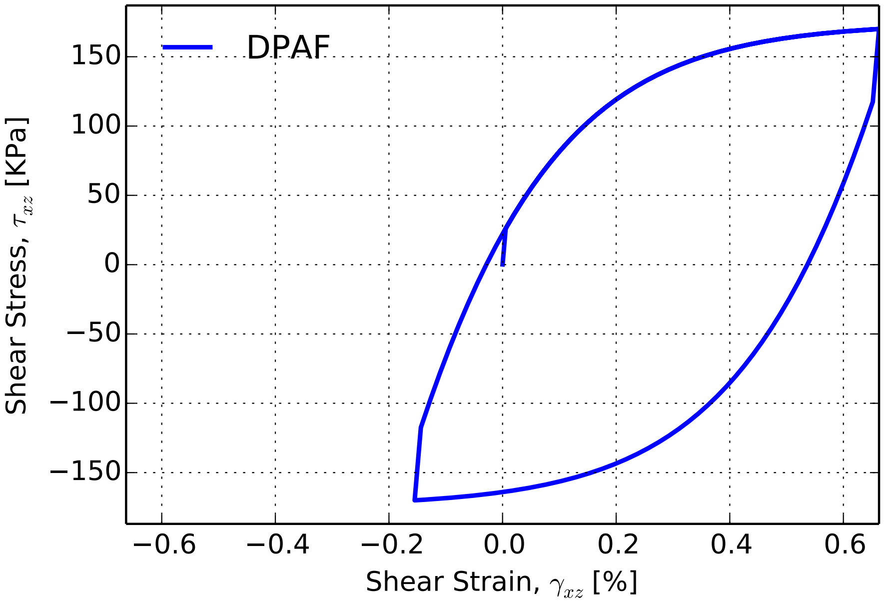


Figure 707.52: Shear stress-strain response.

ESSI model fei/DSL file:

```
// Drucker Prager Armstrong Frederick
// This model is created by Jose.
model name "druckeraf";

// Parameters:
phi = 5;
ha = 1000;
cr = 973;
```

```
gam = 0.01;
Ncyc = 5;
Nsteps = 1000;
H=1;
vp=1000*m/s;
vs=500*m/s;
rho=2000*kg/m^3;
p0 = 250*kPa;
G = rho*vs^2;
M = rho*vp^2;

E = G*(3*M-4*G)/(M-G);
nu = (M-2*G)/(2*M-2*G);

K0 = 1.0;
phirad = pi*phi/180;
M = 6*sin(phirad)/(3-sin(phirad));

// Define the material:
add material # 1 type DruckerPragerArmstrongFrederickLT
    mass_density = 0*kg/m^3
    elastic_modulus = E
    poisson_ratio = nu
    druckerprager_k = M
    armstrong_frederick_ha = ha*Pa
    armstrong_frederick_cr = cr*Pa
    isotropic_hardening_rate = 0*E
    initial_confining_stress = 1*Pa;

// define the node:
add node # 1 at (0*m,0*m,1*m) with 3 dofs;
add node # 2 at (1*m,0*m,1*m) with 3 dofs;
add node # 3 at (1*m,1*m,1*m) with 3 dofs;
add node # 4 at (0*m,1*m,1*m) with 3 dofs;

add node # 5 at (0*m,0*m,0*m) with 3 dofs;
add node # 6 at (1*m,0*m,0*m) with 3 dofs;
add node # 7 at (1*m,1*m,0*m) with 3 dofs;
add node # 8 at (0*m,1*m,0*m) with 3 dofs;

// add equal degree of freedom in three directions
add constraint equal dof with master node # 2 and slave node # 3 dof to ↩
    constrain ux;
add constraint equal dof with master node # 2 and slave node # 6 dof to ↩
    constrain ux;
add constraint equal dof with master node # 2 and slave node # 7 dof to ↩
    constrain ux;

add constraint equal dof with master node # 3 and slave node # 4 dof to ↩
    constrain uy;
add constraint equal dof with master node # 3 and slave node # 8 dof to ↩
```

```
    constrain uy;
add constraint equal dof with master node # 3 and slave node # 7 dof to ↩
    constrain uy;

add constraint equal dof with master node # 1 and slave node # 2 dof to ↩
    constrain uz;
add constraint equal dof with master node # 1 and slave node # 3 dof to ↩
    constrain uz;
add constraint equal dof with master node # 1 and slave node # 4 dof to ↩
    constrain uz;

// Define the element.
add element # 1 type 8NodeBrickLT with nodes (1, 2,3 , 4, 5, 6,7, 8) use ↩
    material # 1;

new loading stage "confinement";
fix node # 1 dofs ux uy;
fix node # 2 dofs uy;
fix node # 4 dofs ux;

fix node # 5 dofs ux uy uz;
fix node # 6 dofs uy uz;
fix node # 7 dofs uz;
fix node # 8 dofs ux uz;

sigma_z = -3*p0/(1+2*K0);
sigma_x = K0*sigma_z;
sigma_y = K0*sigma_z;

//Z-face
add load # 1 to node # 1 type linear Fz = sigma_z*m^2/4;
add load # 2 to node # 2 type linear Fz = sigma_z*m^2/4;
add load # 3 to node # 3 type linear Fz = sigma_z*m^2/4;
add load # 4 to node # 4 type linear Fz = sigma_z*m^2/4;

//X-face
add load # 5 to node # 2 type linear Fx = sigma_x*m^2/4;
add load # 6 to node # 6 type linear Fx = sigma_x*m^2/4;
add load # 7 to node # 7 type linear Fx = sigma_x*m^2/4;
add load # 8 to node # 3 type linear Fx = sigma_x*m^2/4;

add load # 9 to node # 3 type linear Fy = sigma_y*m^2/4;
add load # 10 to node # 7 type linear Fy = sigma_y*m^2/4;
add load # 11 to node # 8 type linear Fy = sigma_y*m^2/4;
add load # 12 to node # 4 type linear Fy = sigma_y*m^2/4;

Nsteps_static=100;
define load factor increment 1/Nsteps_static;

define solver UMFPack;
define convergence test Norm_Displacement_Increment
```

```
    tolerance = 1e-6
    maximum_iterations = 100
    verbose_level = 4;
define algorithm Newton ;

define NDMaterialLT constitutive integration algorithm Euler_One_Step
    yield_function_relative_tolerance = 0.002
    stress_relative_tolerance = 0.002
    maximum_iterations = 1000;

simulate Nsteps_static steps using static algorithm;


new loading stage "shearing";
compute reaction forces;
add load # 13 to node # 1 type from_reactions;
add load # 14 to node # 4 type from_reactions;

free node # 1 dofs ux;
free node # 4 dofs ux;
fix node # 3 dofs uy;
fix node # 6 dofs ux;
fix node # 7 dofs ux uy;
fix node # 8 dofs uy;

add constraint equal dof with master node # 1 and slave node # 3 dof to ↩
    constrain ux;
add constraint equal dof with master node # 1 and slave node # 4 dof to ↩
    constrain ux;
add constraint equal dof with master node # 1 and slave node # 2 dof to ↩
    constrain ux;
remove constraint equaldof node # 6;
remove constraint equaldof node # 7;
remove constraint equaldof node # 8;

n = 1;
while(n<=1)
{
    add load # 14+n to node # n type path_time_series
     Fx = 170.*kN
     series_file = "path.txt";
    n+=1;
}

define load factor increment 1/Nsteps;

define solver UMFPack;
define convergence test Norm_Displacement_Increment
    tolerance = 1e-5
    maximum_iterations = 100
    verbose_level = 4;
```

```
define algorithm Newton ;

define NDMaterialLT constitutive integration algorithm Euler_One_Step
    yield_function_relative_tolerance = 0.0002
    stress_relative_tolerance = 0.002
    maximum_iterations = 1000;

simulate Ncyc*Nsteps steps using static algorithm;

bye;
```

The ESSI model fei/DSL files for this example can be downloaded here.

707.19 Contact Element Under Static Loading

Two Bar Normal Contact Problem Under Monotonic Loading.

This is an example of normal monotonic loading on a 1-D contact/interface between two bars separated by an initial gap of 0.1 unit. An illustrative diagram of the problem statement is shown below.

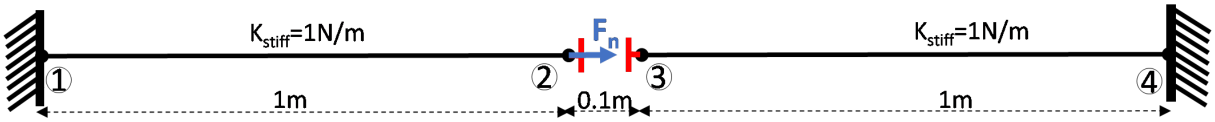


Figure 707.53: Illustration of Two Bar Normal Contact Problem under monotonic loading with intial gap

ESSI model fei/DSL file:

```
model name "Two_Bar_Contact_Under_Normal_Monotonic_Loading" ;

// Adding material
add material #1 type uniaxial_elastic elastic_modulus = 1*Pa ↩
    viscoelastic_modulus = 0*Pa*s;

// Adding Nodes
add node #1 at (0*m,0*m,0*m) with 3 dofs;
add node #2 at (1*m,0*m,0*m) with 3 dofs;
add node #3 at (1.1*m,0*m,0*m) with 3 dofs;
add node #4 at (2.1*m,0*m,0*m) with 3 dofs;

// Adding Fixities
fix node #1 dofs ux uy uz;
fix node #4 dofs ux uy uz;
fix node #2 dofs uy uz ;
fix node #3 dofs uy uz ;

// Adding Truss Elements
add element #1 type truss with nodes (1,2) use material # 1 cross_section = ↩
    1*m^2 mass_density = 1*kg/m^3;
add element #2 type truss with nodes (3,4) use material # 1 cross_section = ↩
    1*m^2 mass_density = 1*kg/m^3;

// Adding Contact Element
add element #3 type FrictionalPenaltyContact with nodes (2,3)
 normal_stiffness =1e10*N/m
 tangential_stiffness = 1e10*Pa*m
 normal_damping = 0*kN/m*s
 tangential_damping = 0*kN/m*s
 friction_ratio = 0.3
```

```
  contact_plane_vector = (1,0,0);

new loading stage "Adding_Normal_Load";

  add load #1 to node #2 type linear Fx = 0.3*N;

  Nsteps = 10;

  tol = 5e-12;
  define convergence test Norm_Displacement_Increment
    tolerance = tol
    maximum_iterations = 10
    verbose_level = 4;

  define algorithm Newton;
  define solver UMFPack;

  define load factor increment 1/Nsteps;
  simulate Nsteps steps using static algorithm;

bye;
```

The displacement output of *Node 2 and Node 3* are shown below.

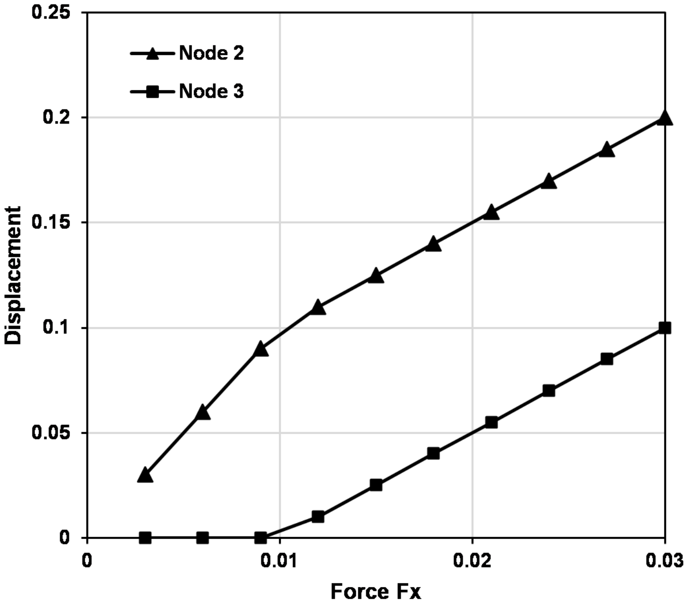


Figure 707.54: Displacemnet of Nodes 2 and 3

The ESSI model fei/DSL files for this example can be downloaded here.

707.20 Four Bar Contact Problem With Normal and Shear Force Under Monotonic Loading

This is an example to show the normal and tangential behaviour (stick and slip case) of contacts/interfaces using four bars in 2-D plane. The bars in x-directions are in contact (initial gap=0).

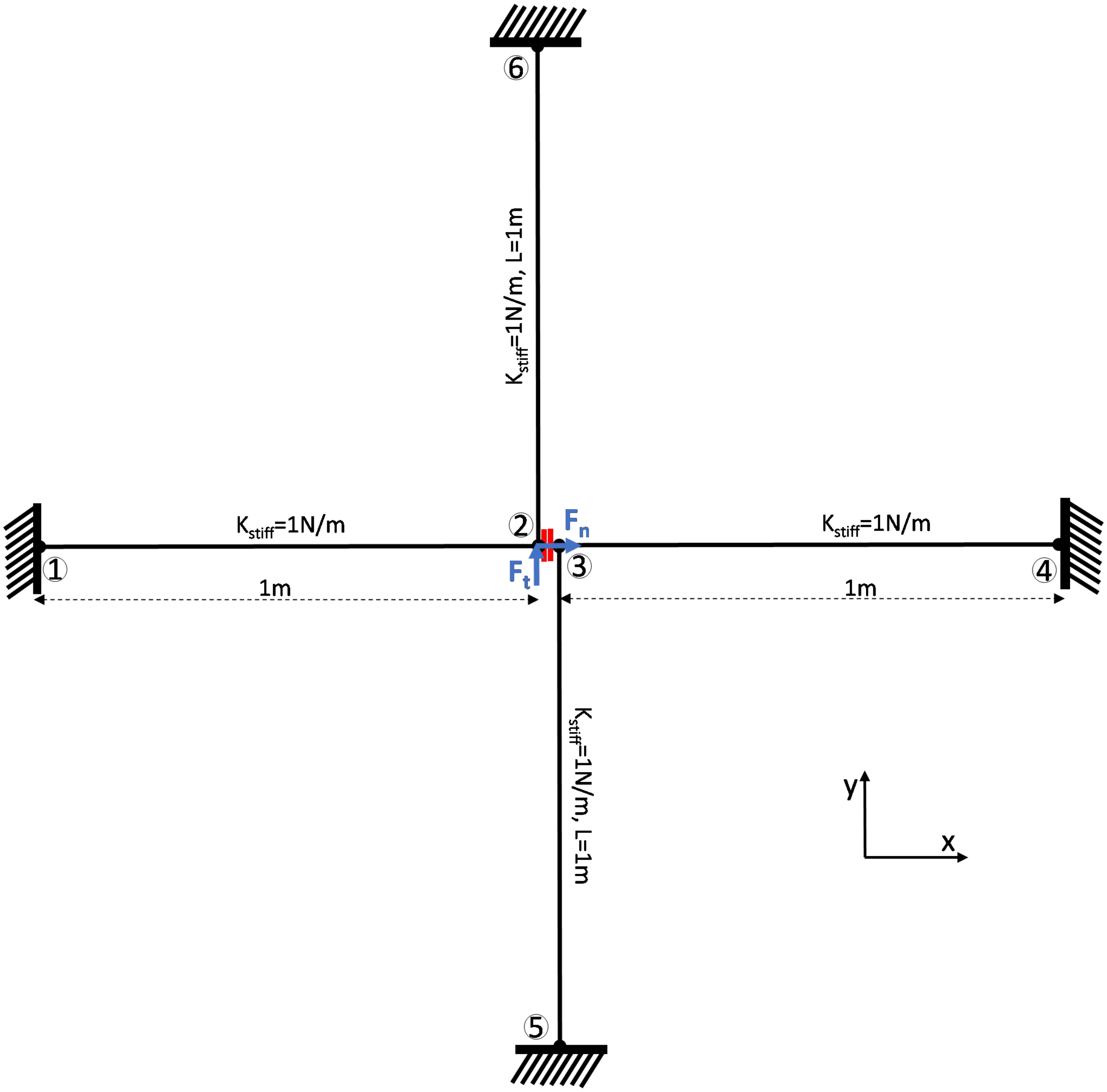


Figure 707.55: Illustration of Four Bar Normal Contact Problem With Normal and Shear Force Under Monotonic Loading with no initial gap

ESSI model fei/DSL file:

```
model name "Four_Bar_Contact_Under_Monotonic_Normal_and_Shear_Loading";

// Adding material
add material #1 type uniaxial_elastic elastic_modulus = 1*Pa ↩
    viscoelastic_modulus = 0*Pa*s;

// Adding Nodes
add node #1 at (0*m,0*m,0*m) with 3 dofs;
add node #2 at (1*m,0*m,0*m) with 3 dofs;
add node #3 at (1*m,0*m,0*m) with 3 dofs;
add node #4 at (2*m,0*m,0*m) with 3 dofs;
add node #5 at (1*m,-1*m,0*m) with 3 dofs;
add node #6 at (1*m,1*m,0*m) with 3 dofs;

// Adding Truss Elements
add element #1 type truss with nodes (1,2) use material # 1 cross_section = ↩
    1*m^2 mass_density = 1*kg/m^3;
add element #2 type truss with nodes (3,4) use material # 1 cross_section = ↩
    1*m^2 mass_density = 1*kg/m^3;
add element #3 type truss with nodes (3,5) use material # 1 cross_section = ↩
    1*m^2 mass_density = 1*kg/m^3;
add element #4 type truss with nodes (2,6) use material # 1 cross_section = ↩
    1*m^2 mass_density = 1*kg/m^3;

// Adding Contact Element
add element #5 type FrictionalPenaltyContact with nodes (2,3)
 normal_stiffness = 1e12*N/m
 tangential_stiffness = 1e12*N/m
 normal_damping = 0*N/m*s
 tangential_damping = 0*N/m*s
 friction_ratio = 0.4
 contact_plane_vector = (1,0,0);

// Adding Fixities
fix node #1 dofs ux uy uz ;
fix node #4 dofs ux uy uz ;
fix node #5 dofs ux uy uz ;
fix node #6 dofs ux uy uz ;
fix node #2 dofs uz ;
fix node #3 dofs uz ;

new loading stage "Normal_Loading";

 add load #1 to node #2 type linear Fx = 0.1*N;

 tol = 1e-10;
 define convergence test Norm_Displacement_Increment
   tolerance = tol
   maximum_iterations = 10
   verbose_level = 4;
```

```

 define algorithm Newton;

 Nsteps= 10;
 define solver UMFPack;
 define load factor increment 1/Nsteps;
 simulate Nsteps steps using static algorithm;

new loading stage "Shear_Loading";

 add load #2 to node #2 type linear Fy = 0.2*N;

 tol = 1e-10;
 define convergence test Norm_Displacement_Increment
   tolerance = tol
   maximum_iterations = 10
   verbose_level = 4;

 define algorithm Newton;

 Nsteps= 100;
 define solver UMFPack;
 define load factor increment 1/Nsteps;
 simulate Nsteps steps using static algorithm;

bye;
```

The displacement output of *Node 2 and Node 3* are shown below.

The ESSI model fei/DSL files for this example can be downloaded here.

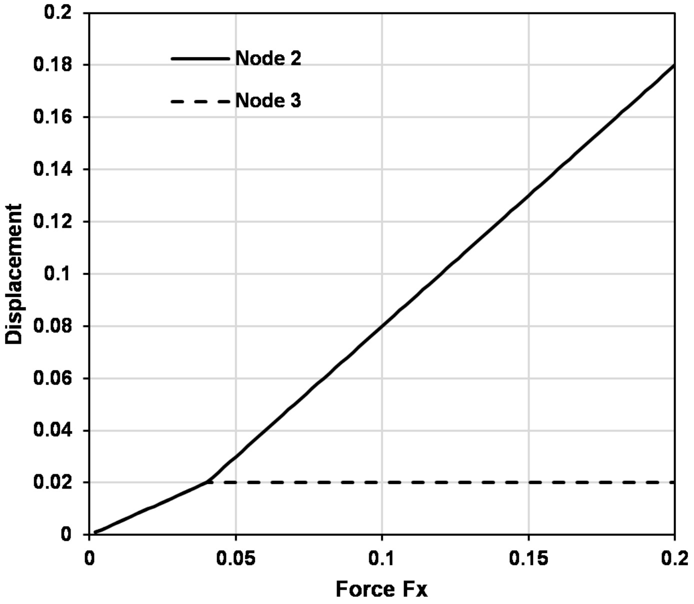


Figure 707.56: Displacemnet of Nodes 2 and 3 along y direction

707.21 3-D Truss example with normal confinement and Shear Loading

A simple 3-D truss example with Normal confinement in z-direction of $F_N = 0.5N$, friction coefficient $\mu = 0.2$ and shear loading of magnitude $F_s = 0.5N$. Figure 707.57 below, shows the description of the problem.

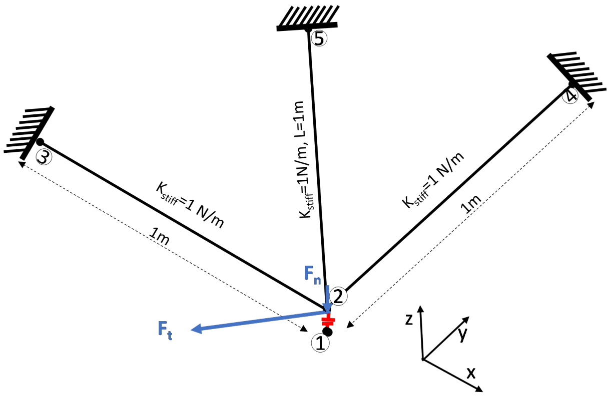


Figure 707.57: Illustration of 3-D Truss Problem with confinement loading in z-direction of 0.5N and then shear loading of 0.5N in x-y plane

ESSI model fei/DSL file:

```
model name "3-D_Contact_Under_Normal_And_Tangential_Loading" ;

// Adding material
add material #1 type uniaxial_elastic elastic_modulus = 1*Pa ↩
    viscoelastic_modulus = 0*Pa*s;

// Adding Nodes
add node #1 at (0*m,0*m,0*m) with 3 dofs;
add node #2 at (0*m,0*m,0*m) with 3 dofs;
add node #3 at (-1*m,0*m,0*m) with 3 dofs;
add node #4 at (0*m,1*m,0*m) with 3 dofs;
add node #5 at (0*m,0*m,1*m) with 3 dofs;

// Adding Fixities
fix node #1 dofs ux uy uz;
fix node #3 dofs ux uy uz;
fix node #4 dofs ux uy uz;
fix node #5 dofs ux uy uz;

// Adding Truss Elements
add element #1 type truss with nodes (2,3) use material # 1 cross_section = ↩
    1*m^2 mass_density = 1*kg/m^3;
add element #2 type truss with nodes (2,4) use material # 1 cross_section = ↩
```

```
        1*m^2 mass_density = 1*kg/m^3;
    add element #3 type truss with nodes (2,5) use material # 1 cross_section = ↩
        1*m^2 mass_density = 1*kg/m^3;

    // Adding Contact Element
    add element #4 type FrictionalPenaltyContact with nodes (1,2)
    normal_stiffness =1e10*N/m
    tangential_stiffness = 1e10*Pa*m
    normal_damping = 0*kN/m*s
    tangential_damping = 0*kN/m*s
    friction_ratio = 0.2
    contact_plane_vector = (0,0,1);

    new loading stage "Adding_Normal_Load";

    add load #1 to node #2 type linear Fz = -0.5*N;

    Nsteps = 1;

    tol = 1e-10;
    define convergence test Norm_Displacement_Increment
      tolerance = tol
      maximum_iterations = 1
      verbose_level = 4;

    define algorithm Newton;
    define solver UMFPack;

    define load factor increment 1/Nsteps;
    simulate Nsteps steps using static algorithm;

    new loading stage "Shear_Loading";

    add load #2 to node #2 type linear Fx = 0.4;
    add load #3 to node #2 type linear Fy = 0.3;

    tol = 1e-12;
    define convergence test Norm_Displacement_Increment
      tolerance = tol
      maximum_iterations = 10
      verbose_level = 4;

    define algorithm Newton;

    Nsteps= 20;
    define solver UMFPack;
    define load factor increment 1/Nsteps;
    simulate Nsteps steps using static algorithm;

    bye;
```

The generalized displacement response of the tangential loading stage is shown below.

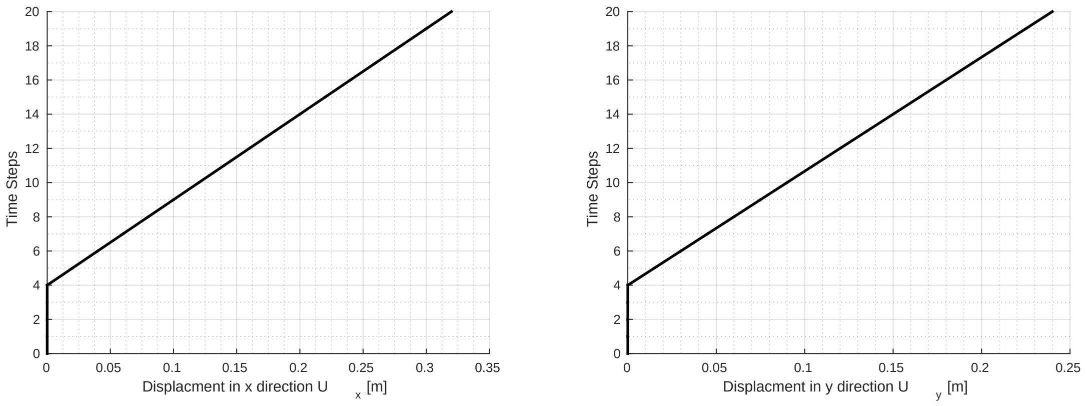


Figure 707.58: Displacements of Node 2 with applied shear tangential load step.

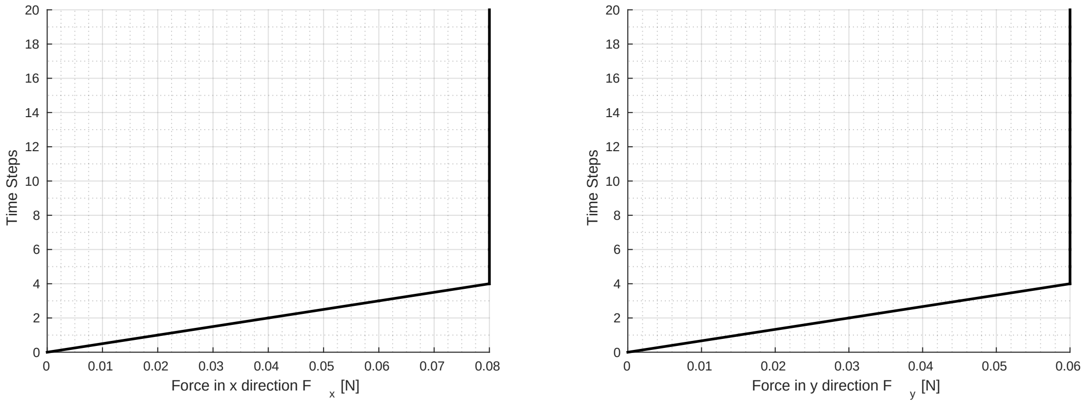


Figure 707.59: Resisting force by the contact/interface element with applied shear tangential load step.

The ESSI model fei/DSL files for this example can be downloaded here.

707.22 Six Solid Blocks Example With Contact

This is a 3-D solid block example withi initial normal and then tangential load on different surfaces as shown below.

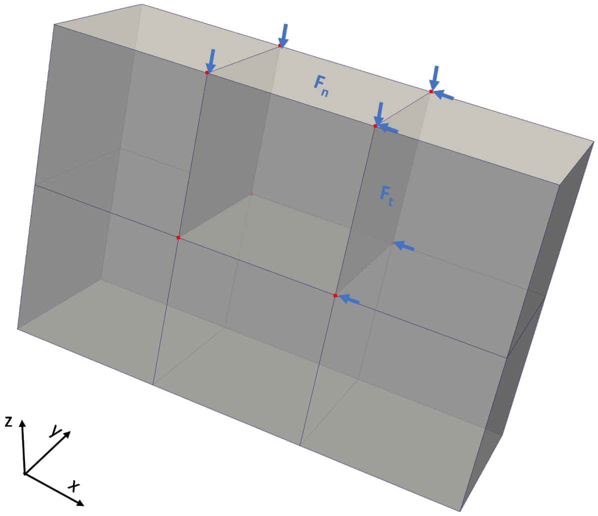


Figure 707.60: Illustration of Six Solid Blocks Example with Contact having first normal and then tangential loading stages.

ESSI model fei/DSL file:

```

model name "Six_Solid_Blocks_Example_With_Contact";

// Adding material
 add material #1 type linear_elastic_isotropic_3d_LT mass_density=2000*kg/m^3 elastic_modulus=200*MPa poisson_ratio=0.3;

// Adding Nodes
 add node # 1 at (-1.500000*m,-0.500000*m,0.000000*m) with 3 dofs;
 add node # 2 at (-1.500000*m,0.500000*m,0.000000*m) with 3 dofs;
 add node # 3 at (1.500000*m,-0.500000*m,0.000000*m) with 3 dofs;
 add node # 4 at (1.500000*m,0.500000*m,0.000000*m) with 3 dofs;
 add node # 5 at (-1.500000*m,-0.500000*m,-2.000000*m) with 3 dofs;
 add node # 6 at (-1.500000*m,0.500000*m,-2.000000*m) with 3 dofs;
 add node # 7 at (1.500000*m,0.500000*m,-2.000000*m) with 3 dofs;
 add node # 8 at (1.500000*m,-0.500000*m,-2.000000*m) with 3 dofs;
 add node # 9 at (-0.500000*m,-0.500000*m,0.000000*m) with 3 dofs;
 add node # 10 at (0.500000*m,-0.500000*m,0.000000*m) with 3 dofs;
 add node # 11 at (-0.500000*m,0.500000*m,0.000000*m) with 3 dofs;
 add node # 12 at (0.500000*m,0.500000*m,0.000000*m) with 3 dofs;
 add node # 13 at (-0.500000*m,0.500000*m,-2.000000*m) with 3 dofs;
```

```
 add node # 14 at (0.500000*m,0.500000*m,-2.000000*m) with 3 dofs;
 add node # 15 at (0.500000*m,-0.500000*m,-2.000000*m) with 3 dofs;
 add node # 16 at (-0.500000*m,-0.500000*m,-2.000000*m) with 3 dofs;
 add node # 17 at (-1.500000*m,-0.500000*m,-1.000000*m) with 3 dofs;
 add node # 18 at (-1.500000*m,0.500000*m,-1.000000*m) with 3 dofs;
 add node # 19 at (1.500000*m,0.500000*m,-1.000000*m) with 3 dofs;
 add node # 20 at (1.500000*m,-0.500000*m,-1.000000*m) with 3 dofs;
 add node # 21 at (-0.500000*m,0.500000*m,-1.000000*m) with 3 dofs;
 add node # 22 at (0.500000*m,0.500000*m,-1.000000*m) with 3 dofs;
 add node # 23 at (-0.500000*m,-0.500000*m,-1.000000*m) with 3 dofs;
 add node # 24 at (0.500000*m,-0.500000*m,-1.000000*m) with 3 dofs;

 add node # 25 at (-0.500000*m,-0.500000*m,0.000000*m) with 3 dofs;
 add node # 26 at (0.500000*m,-0.500000*m,0.000000*m) with 3 dofs;
 add node # 27 at (-0.500000*m,0.500000*m,0.000000*m) with 3 dofs;
 add node # 28 at (0.500000*m,0.500000*m,0.000000*m) with 3 dofs;

 add node # 29 at (-0.500000*m,0.500000*m,-1.000000*m) with 3 dofs;
 add node # 30 at (0.500000*m,0.500000*m,-1.000000*m) with 3 dofs;
 add node # 31 at (-0.500000*m,-0.500000*m,-1.000000*m) with 3 dofs;
 add node # 32 at (0.500000*m,-0.500000*m,-1.000000*m) with 3 dofs;

// Adding Solid 8 Node Brick Elements
 add element #1 type 8NodeBrickLT with nodes (21,23,17,18,11,9,1,2) use ↩
     material #1;
 add element #2 type 8NodeBrickLT with nodes (13,16,5,6,21,23,17,18) use ↩
     material #1;
 add element #3 type 8NodeBrickLT with nodes (30,32,31,29,28,26,25,27) use ↩
     material #1;
 add element #4 type 8NodeBrickLT with nodes (14,15,16,13,22,24,23,21) use ↩
     material #1;
 add element #5 type 8NodeBrickLT with nodes (19,20,24,22,4,3,10,12) use ↩
     material #1;
 add element #6 type 8NodeBrickLT with nodes (7,8,15,14,19,20,24,22) use ↩
     material #1;

//Adding some variables
 Kn = 1e12*N/m; // normal penalty stiffness
 Kt = 1e12*N/m; // tangential penalty stiffness
 Cn = 0*N/m*s; // normal penalty damping
 Ct = 0*N/m*s; // tangential penalty damping
 nu = 0.4; // friction ratio

// Adding Contact Element
 add element #7 type FrictionalPenaltyContact with nodes (9,25)
  normal_stiffness = Kn
  tangential_stiffness = Kt
  normal_damping = Cn
  tangential_damping = Ct
  friction_ratio = nu
  contact_plane_vector = (1,0,0);
```

```

add element #8 type FrictionalPenaltyContact with nodes (10,26)
 normal_stiffness = Kn
 tangential_stiffness = Kt
 normal_damping = Cn
 tangential_damping = Ct
 friction_ratio = nu
 contact_plane_vector = (-1,0,0);

add element #9 type FrictionalPenaltyContact with nodes (11,27)
 normal_stiffness = Kn
 tangential_stiffness = Kt
 normal_damping = Cn
 tangential_damping = Ct
 friction_ratio = nu
 contact_plane_vector = (1,0,0);

add element #10 type FrictionalPenaltyContact with nodes (12,28)
 normal_stiffness = Kn
 tangential_stiffness = Kt
 normal_damping = Cn
 tangential_damping = Ct
 friction_ratio = nu
 contact_plane_vector = (-1,0,0);

add element #11 type FrictionalPenaltyContact with nodes (21,29)
 normal_stiffness = Kn
 tangential_stiffness = Kt
 normal_damping = Cn
 tangential_damping = Ct
 friction_ratio = nu
 contact_plane_vector = (1,0,0);

add element #12 type FrictionalPenaltyContact with nodes (22,30)
 normal_stiffness = Kn
 tangential_stiffness = Kt
 normal_damping = Cn
 tangential_damping = Ct
 friction_ratio = nu
 contact_plane_vector = (-1,0,0);

add element #13 type FrictionalPenaltyContact with nodes (23,31)
 normal_stiffness = Kn
 tangential_stiffness = Kt
 normal_damping = Cn
 tangential_damping = Ct
 friction_ratio = nu
 contact_plane_vector = (1,0,0);

add element #14 type FrictionalPenaltyContact with nodes (24,32)
 normal_stiffness = Kn
```

```
  tangential_stiffness = Kt
  normal_damping = Cn
  tangential_damping = Ct
  friction_ratio = nu
  contact_plane_vector = (-1,0,0);

 add element #15 type FrictionalPenaltyContact with nodes (21,29)
  normal_stiffness = Kn
  tangential_stiffness = Kt
  normal_damping = Cn
  tangential_damping = Ct
  friction_ratio = nu
  contact_plane_vector = (0,0,1);

 add element #16 type FrictionalPenaltyContact with nodes (22,30)
  normal_stiffness = Kn
  tangential_stiffness = Kt
  normal_damping = Cn
  tangential_damping = Ct
  friction_ratio = nu
  contact_plane_vector = (0,0,1);

 add element #17 type FrictionalPenaltyContact with nodes (23,31)
  normal_stiffness = Kn
  tangential_stiffness = Kt
  normal_damping = Cn
  tangential_damping = Ct
  friction_ratio = nu
  contact_plane_vector = (0,0,1);

 add element #18 type FrictionalPenaltyContact with nodes (24,32)
  normal_stiffness = Kn
  tangential_stiffness = Kt
  normal_damping = Cn
  tangential_damping = Ct
  friction_ratio = nu
  contact_plane_vector = (0,0,1);

// Adding Fixities
 fix node #5 dofs ux uy uz;
 fix node #6 dofs ux uy uz;
 fix node #13 dofs ux uy uz;
 fix node #16 dofs ux uy uz;
 fix node #15 dofs ux uy uz;
 fix node #14 dofs ux uy uz;
 fix node #7 dofs ux uy uz;
 fix node #8 dofs ux uy uz;
 fix node #17 dofs ux uy;
 fix node #18 dofs ux uy;
 fix node #1 dofs ux uy;
 fix node #2 dofs ux uy;
```

```
 fix node #20 dofs ux uy;
 fix node #19 dofs ux uy;
 fix node #3 dofs ux uy;
 fix node #4 dofs ux uy;
 fix node #9 dofs uy;
 fix node #10 dofs uy;
 fix node #23 dofs uy;
 fix node #24 dofs uy;
 fix node #11 dofs uy;
 fix node #21 dofs uy;
 fix node #12 dofs uy;
 fix node #22 dofs uy;
 fix node #25 dofs uy;
 fix node #26 dofs uy;
 fix node #27 dofs uy;
 fix node #28 dofs uy;
 fix node #29 dofs uy;
 fix node #30 dofs uy;
 fix node #31 dofs uy;
 fix node #32 dofs uy;

new loading stage "Normal_Loading";

 add load #1 to element #3 type surface at nodes (25,26,27,28) with magnitude ↩
    (-1*Pa);

 tol = 1e-12;
 define convergence test Norm_Displacement_Increment
   tolerance = tol
   maximum_iterations = 100
   verbose_level = 4;

 define algorithm Newton;

 Nsteps= 10;
 define solver UMFPack;
 define load factor increment 1/Nsteps;
 simulate Nsteps steps using static algorithm;

new loading stage "Shear_Loading";

 add load #2 to element #3 type surface at nodes (26,28,30,32) with magnitude ↩
    (-1*Pa);

 tol = 1e-12;
 define convergence test Norm_Displacement_Increment
   tolerance = tol
   maximum_iterations = 100
   verbose_level = 4;

 define algorithm Newton;
```

```

 Nsteps= 10;
 define solver UMFPack;
 define load factor increment 1/Nsteps;
 simulate Nsteps steps using static algorithm;

bye;
```

The generalized displacement field of the two loading stages normal loading and tangentiual loading is shown below..

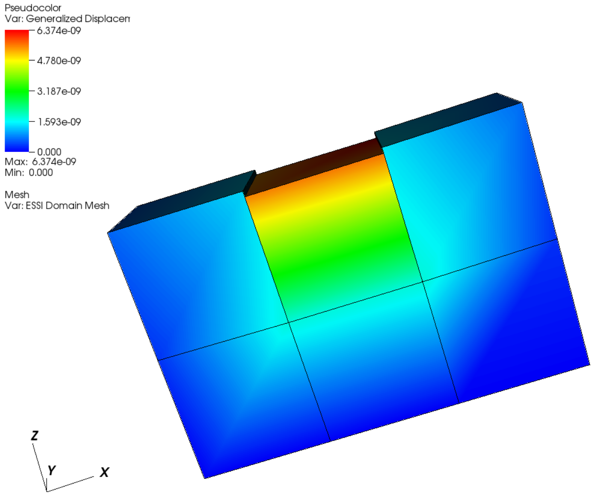


Figure 707.61: Generalized displacement magnitude visualization of normal loading

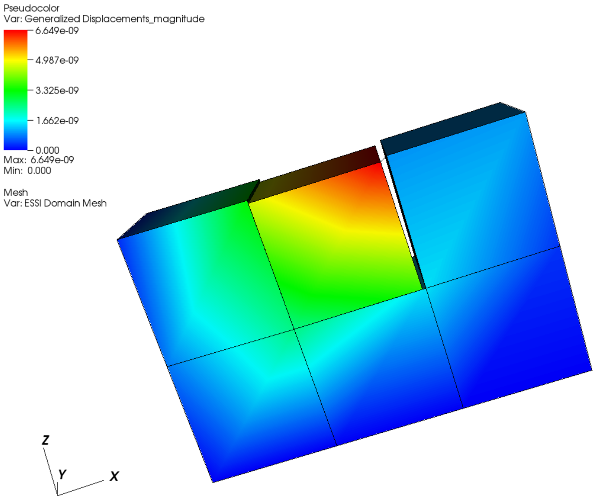


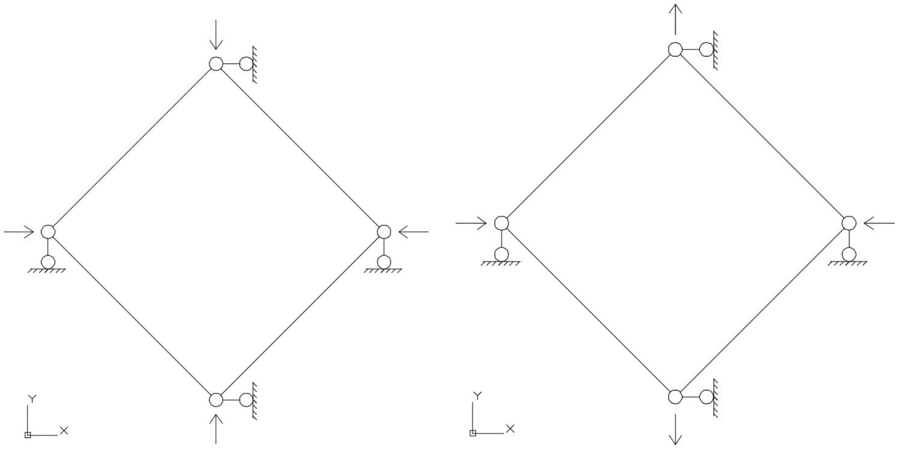
Figure 707.62: Generalized displacement magnitude visualization of tangential loading

The ESSI model fei/DSL files for this example can be downloaded here.

707.23 Pure shear model for G/Gmax plot

Problem description:

The pure shear model for G/Gmax plot



(a) (b)

Figure 707.63: The pure shear model for (a) confinment and (b) shearing

ESSI model fei/DSL file:

```
model name "GGmax" ;
// Parameters:
phi = 0.0135713590083;
ha = 2.94767923453;
cr = 1854.31984573;

rho=1922.5 ;
depth=0.1524/2;
confinstress=9.8*depth*rho;
G=12388.33;

p0 = confinstress*Pa;
phirad = pi*phi/180;
M = 6*sin(phirad)/(3-sin(phirad));
nu=0.3;
add material # 1 type DruckerPragerArmstrongFrederickLT
    mass_density = rho*kg/m^3
    elastic_modulus = 2*G*(1+nu)*Pa
    poisson_ratio = nu
    druckerprager_k = M
    armstrong_frederick_ha = ha*Pa
    armstrong_frederick_cr = cr*Pa
```

```
    isotropic_hardening_rate = 0*Pa
    initial_confining_stress = 10*Pa;
add node # 1 at ( 1.0000 *m, 0.0000 *m, 0.0000 *m) with 3 dofs;
add node # 2 at ( 0.0000 *m, 1.0000 *m, 0.0000 *m) with 3 dofs;
add node # 3 at ( 1.0000 *m, 2.0000 *m, 0.0000 *m) with 3 dofs;
add node # 4 at ( 2.0000 *m, 1.0000 *m, 0.0000 *m) with 3 dofs;
add node # 5 at ( 1.0000 *m, 0.0000 *m, 1.0000 *m) with 3 dofs;
add node # 6 at ( 0.0000 *m, 1.0000 *m, 1.0000 *m) with 3 dofs;
add node # 7 at ( 1.0000 *m, 2.0000 *m, 1.0000 *m) with 3 dofs;
add node # 8 at ( 2.0000 *m, 1.0000 *m, 1.0000 *m) with 3 dofs;
add element # 1 type 8NodeBrickLT with nodes(1,2,3,4,5,6,7,8) use material # 1;

// fix the y direction for node 2,4,6,8
fix node # 2 dofs uy ;
fix node # 4 dofs uy ;
fix node # 6 dofs uy ;
fix node # 8 dofs uy ;
// fix the x direction for node 1,3,5,7
fix node # 1 dofs ux ;
fix node # 3 dofs ux ;
fix node # 5 dofs ux ;
fix node # 7 dofs ux ;
// Stage 1: confinement
new loading stage "confinement";
add load # 1 to node # 1 type linear Fy= p0*m^2;
add load # 2 to node # 3 type linear Fy= - p0*m^2;
add load # 3 to node # 5 type linear Fy= p0*m^2;
add load # 4 to node # 7 type linear Fy= - p0*m^2;

add load # 5 to node # 2 type linear Fx= p0*m^2;
add load # 6 to node # 4 type linear Fx= - p0*m^2;
add load # 7 to node # 6 type linear Fx= p0*m^2;
add load # 8 to node # 8 type linear Fx= - p0*m^2;

// confinement at z direction
add load # 101 to node # 1 type linear Fz= p0*m^2;
add load # 102 to node # 2 type linear Fz= p0*m^2;
add load # 103 to node # 3 type linear Fz= p0*m^2;
add load # 104 to node # 4 type linear Fz= p0*m^2;

add load # 105 to node # 5 type linear Fz= - p0*m^2;
add load # 106 to node # 6 type linear Fz= - p0*m^2;
add load # 107 to node # 7 type linear Fz= - p0*m^2;
add load # 108 to node # 8 type linear Fz= - p0*m^2;

// add algorithm and solver
Nsteps=100;
define load factor increment 1/Nsteps;
define solver ProfileSPD;
define convergence test Norm_Displacement_Increment
    tolerance = 1e-5
```

```
    maximum_iterations = 100
    verbose_level = 4;
// define algorithm With_no_convergence_check ;
define algorithm Newton ;
define NDMaterialLT constitutive integration algorithm Euler_One_Step
    yield_function_relative_tolerance = 0.00002
    stress_relative_tolerance = 0.0002
    maximum_iterations = 1000;
simulate Nsteps steps using static algorithm;
// ---------------------------------------------------------------------
// Stage 2: shear
new loading stage "shear";
// fix all the uz, since we want plane strain.
i=1;
while (i<9) {
    remove load # 100+i ;
    fix node # i dofs uz;
    i=i+1;
};
shearforce=1.6*kN;


add load # 9 to node # 1 type linear Fy= shearforce;// series_file = "path.txt" ;
add load # 10 to node # 3 type linear Fy=-shearforce;// series_file = ↩
    "path.txt" ;
add load # 11 to node # 5 type linear Fy= shearforce;// series_file = ↩
    "path.txt" ;
add load # 12 to node # 7 type linear Fy=-shearforce;// series_file = ↩
    "path.txt" ;

add load # 13 to node # 2 type linear Fx=-shearforce;// series_file = ↩
    "path.txt" ;
add load # 14 to node # 4 type linear Fx= shearforce;// series_file = ↩
    "path.txt" ;
add load # 15 to node # 6 type linear Fx=-shearforce;// series_file = ↩
    "path.txt" ;
add load # 16 to node # 8 type linear Fx= shearforce;// series_file = ↩
    "path.txt" ;

// add algorithm and solver
Nsteps=1e4 ;
define static integrator displacement_control using node # 1 dof uy increment ↩
    1e-2/Nsteps*m;
define convergence test Norm_Displacement_Increment tolerance = 0.000001 ↩
    maximum_iterations = 100 verbose_level = 0;
define solver ProfileSPD;
define algorithm Newton ;
define NDMaterialLT constitutive integration algorithm Euler_One_Step
    yield_function_relative_tolerance = 0.00002
    stress_relative_tolerance = 0.0002
    maximum_iterations = 1000;
```

```

simulate Nsteps steps using static algorithm;
bye;
```

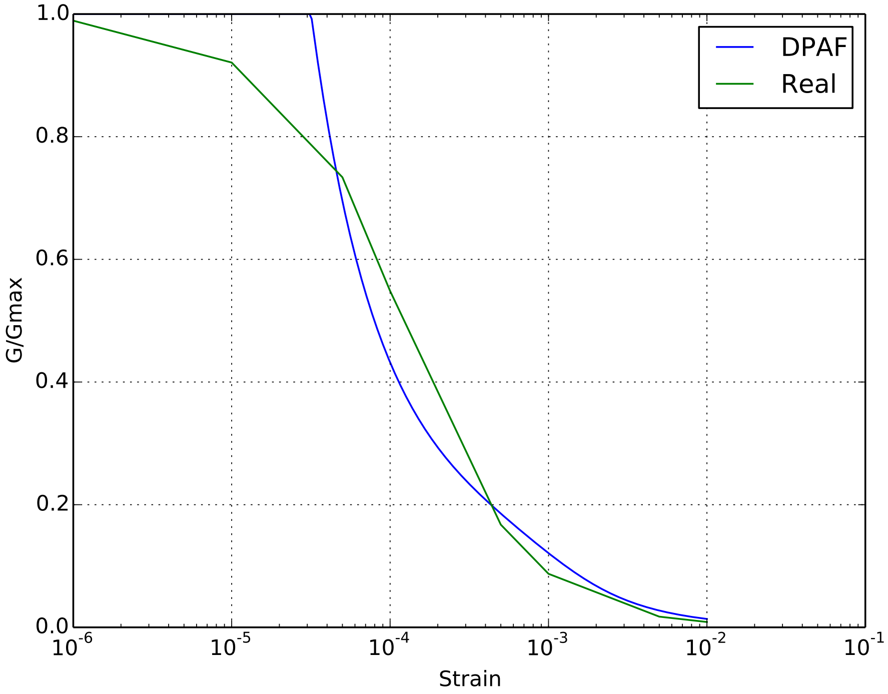


Figure 707.64: The G/Gmax results

The ESSI model fei/DSL files for this example can be downloaded here.

707.24 Multi-yield-surface von-Mises for G/Gmax plot

Problem description:

This model illustates the G/Gmax input to multi-yield-surface von-Mises material. This example is based on one Gauss-point with multi-yield-surface von-Mises material. The G/Gmax is converted to material modelingn parameters (yield-surface size and hardening parameter) inside the DSL.

ESSI model fei/DSL file:

```
model name "GGmax";
add material # 1 type vonMisesMultipleYieldSurfaceGoverGmax
  mass_density = 0.0*kg/m^3
  initial_shear_modulus = 3E8 * Pa
  poisson_ratio = 0.0
  total_number_of_shear_modulus = 9
  GoverGmax =
  "1,0.995,0.966,0.873,0.787,0.467,0.320,0.109,0.063"
  ShearStrainGamma =
  "0,1E-6,1E-5,5E-5,1E-4, 0.0005, 0.001, 0.005, 0.01"
  ;

incr_size = 0.000001 ;
max_strain= 0.005 ;
num_of_increm = max_strain/incr_size -1 ;
simulate constitutive testing strain control pure shear use material # 1
  confinement_strain = 0.0
  strain_increment_size = incr_size
  maximum_strain = max_strain
  number_of_increment = num_of_increm;
bye;
```

Computed G/Gmax curve exactly matches the one used for input at control points.

The difference in G/Gmax between control points can be reduced by using more than just 9 control points as in this example.

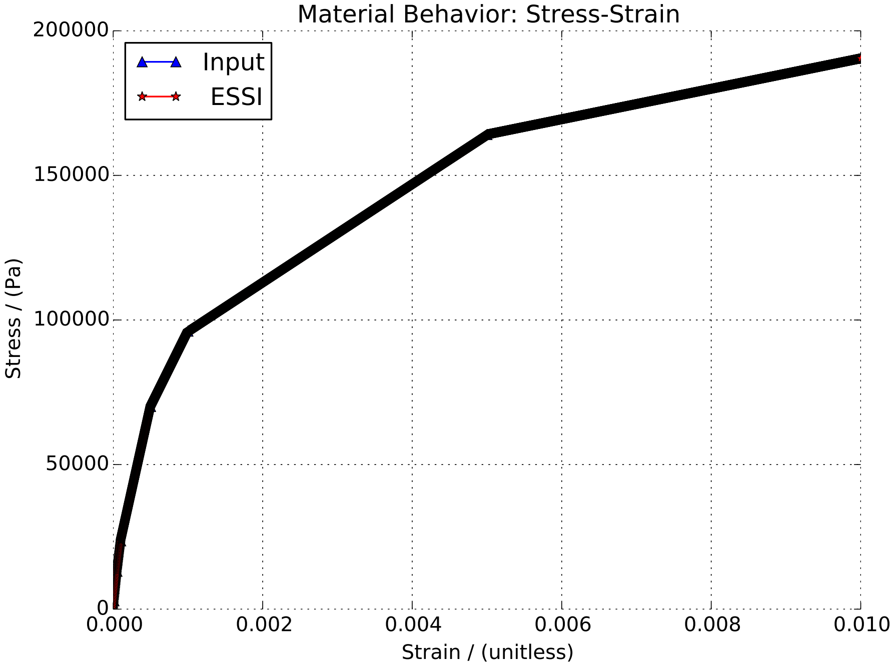


Figure 707.65: Stress-Strain Relationship

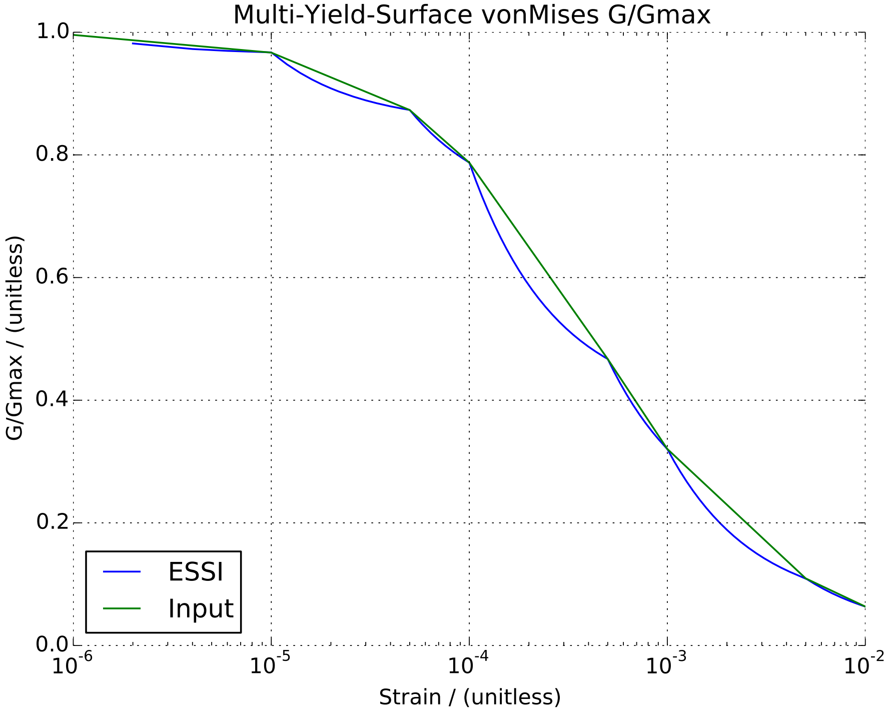


Figure 707.66: The G/Gmax results.

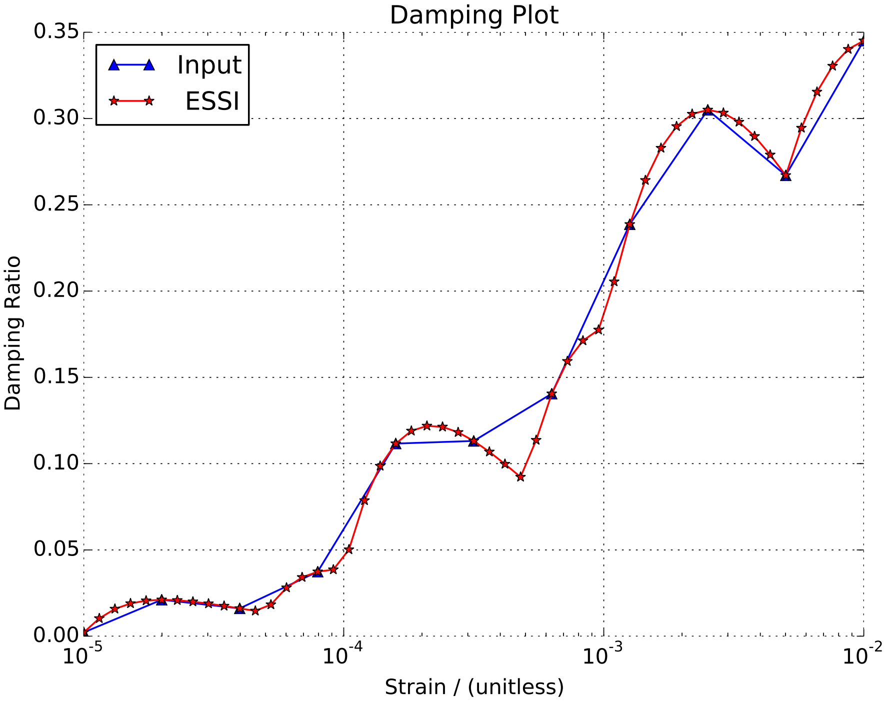


Figure 707.67: Damping Ratio Plot

707.25 Multi-yield-surface Drucker-Prager for G/Gmax plot

Problem description:

This model illustrates the G/Gmax input to multi-yield-surface Drucker-Prager material. Purely deviatoric plastic flow is used in this material, which means that the parameter dilation_scale is set to zero. If user wants to model change of volume (dilation or compression) for this material, then G/Gmax curve need to be iterated upon manually by changing yield surface size directly, which is done using different DruckerPragerMultipleYieldSurface command. This example is based on one Gauss-point which use multi-yield-surface Drucker-Prager material. The G/Gmax is converted to the yield-surface size and hardening parameter inside the DSL.

ESSI model fei/DSL file:

```
model name "GGmax";
add material # 1 type DruckerPragerMultipleYieldSurfaceGoverGmax
  mass_density = 0.0*kg/m^3
  initial_shear_modulus = 3E8 * Pa
  poisson_ratio = 0.0
  initial_confining_stress = 1E5 * Pa
  reference_pressure = 1E5 * Pa
  pressure_exponential_n = 0.5
  cohesion = 0. * Pa
  dilation_angle_eta =1.0
  dilation_scale = 0.0
  total_number_of_shear_modulus = 9
  GoverGmax =
  "1,0.995,0.966,0.873,0.787,0.467,0.320,0.109,0.063"
  ShearStrainGamma =
  "0,1E-6,1E-5,5E-5,1E-4, 0.0005, 0.001, 0.005, 0.01"
  ;

incr_size = 0.000001 ;
max_strain= 0.005 ;
num_of_increm = max_strain/incr_size -1 ;
simulate constitutive testing strain control pure shear use material # 1
  confinement_strain = 0.0
  strain_increment_size = incr_size
  maximum_strain = max_strain
  number_of_increment = num_of_increm;
bye;
```

Inside the DSL, the yield surface radius is calculated as $\sqrt{3}\sigma_y$, where σ_y is the yield stress of the corresponding yield surface. Then, the radius is divided by the confinement to obtain the slope (opening angle).

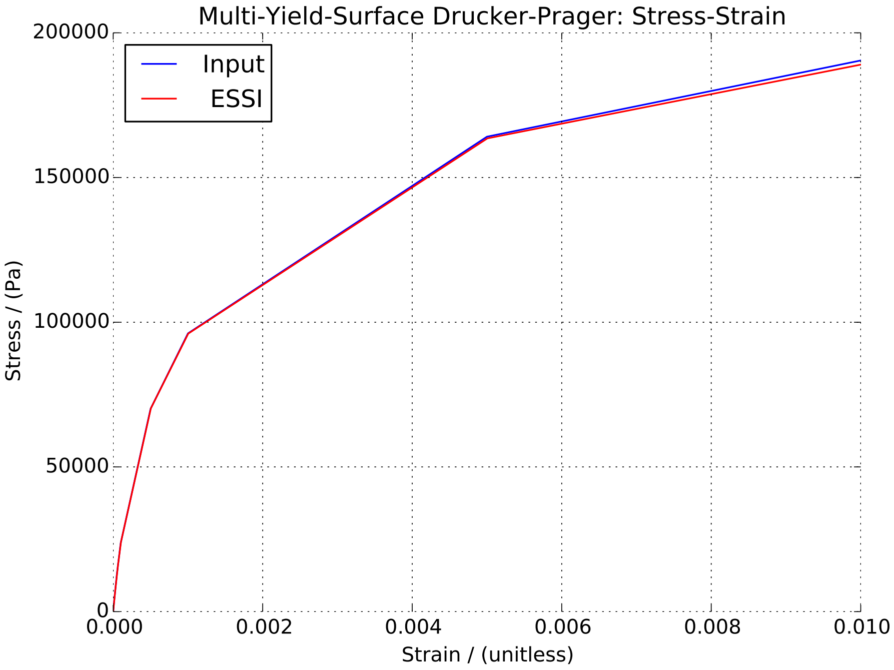


Figure 707.68: Nested-Yield-Surface Drucker-Prager Stress-Strain Relationship

The hardening parameter is calculated as

$$\frac{1}{H_i'} = \frac{1}{H_i} - \frac{1}{2G} \tag{707.1}$$

where H_i' is the current hardening parameter corresponding to yield surface i. H_i is the current tangent shear modulus to surface i, namely, $H_i = 2(\frac{\tau_{i+1}-\tau_i}{\gamma_{i+1}-\gamma_i})$. And G is the initial shear modulus.

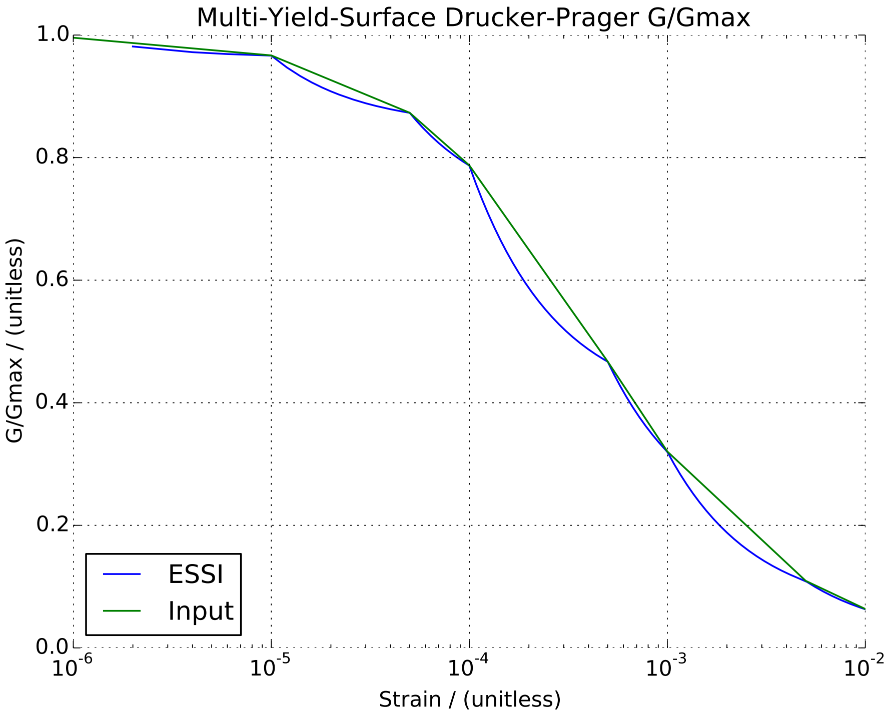


Figure 707.69: Nested-Yield-Surface Drucker-Prager G/Gmax results

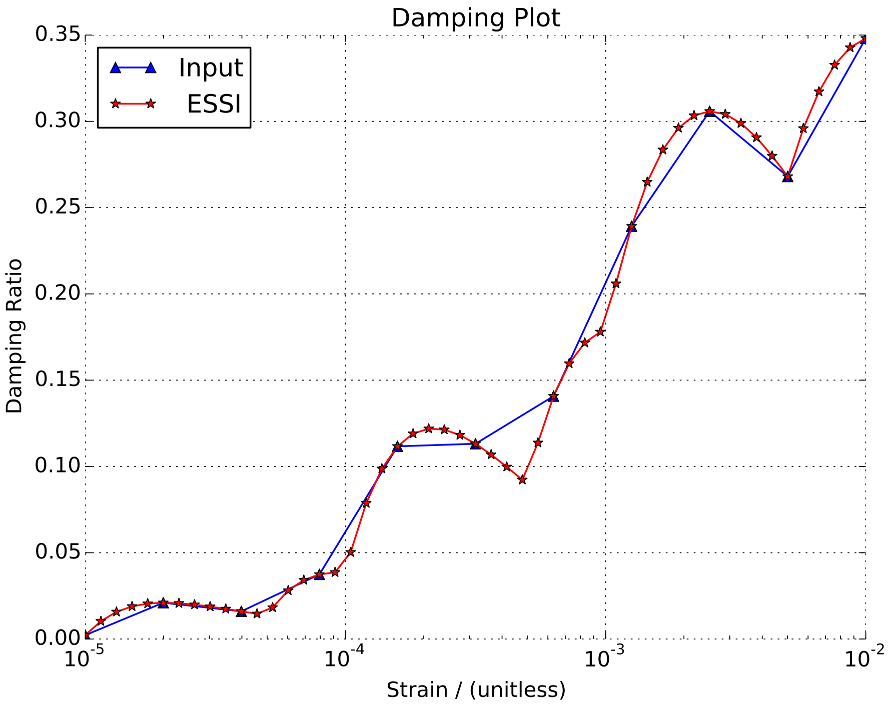


Figure 707.70: Damping Ratio Plot

Appendix 708

Brief History of the Real-ESSI Simulator Development

(1986-)

This section briefly describes history of the development of the Finite Element Interpreted, FEI, that is currently represented by the Real-ESSI Simulator system. Developments are presented chronologically, with very brief description of capabilities, and with references to further reading and documents with more information.

1986-1988: Development of the `FRAME_and_GRID` program, in 2D, using BASIC programming language, on SHARP 1500, CASIO 1000 (48KB RAM) and ZX Spectrum (128KB RAM), by Boris Jeremić, undergraduate student at the University of Belgrade.

1988-1989: Development of the Earthquake Soil Structure Interaction (ESSI) Program in time domain for axisymmetric solids with general 3D loads, using higher modes of response in circumferential direction, expanded in Fourier series, so that any general 3D loading and deformation can be modeled, earthquake shaking applied through "heavy" rock at the bottom of the model, using FORTRAN programming language, on PC-DOS, x286+287, 640KB+384KB RAM, by Boris Jeremić, undergraduate student at the University of Belgrade, as part of his Diploma Thesis (Jeremić, 1989).

1989-1992: Development of the Finite Element Interpreter (FEI), a general purpose static and dynamic, elastic and elastic-plastic finite element program for solids (3D), rudimentary parser for a simple Domain Specific Language (DSL), using C Programming language, on PC-DOS, x286+287, 640KB+384KB RAM, by Boris Jeremić, a staff engineer at (a) Energoprojekt-Hidroinžinjering Company in Belgrade, Yugoslavia, at (b) Bekhme Dam Project site in Iraq, and at (c) Gasser&Scepan Design Bureau in Baar, Switzerland.

1992-1997: Development of the program FEM, featuring small and large deformation (large strain, large displacements/rotations), elasto-plasticity, solids (bricks with 8, 20 and 27 nodes), solution advancement control (hyperspherical/arc length control), using C++ Programming language, on Sun-SparcStation 5, Solaris, 256MB RAM, and on PC-DOS x386, x486 and on PC-Linux-TurboRedHat, by Boris Jeremić, a graduate student at the University of Colorado at Boulder, as part of his Master Thesis (Jeremić, 1994) and PhD Dissertation (Jeremić, 1997).

1997-2000: Continued development of the program FEM, addition of dynamics from ESSI, structural elements from `FRAME_and_GRID`, Parallel version, MPI based, linking with FEI, using C++ Programming language, on PC-Linux, and PC-Linux cluster: NorthCountry, 4 nodes + master, 100based T network, by Professor Boris Jeremić, at Clarkson University and at the University of California at Davis.

2000-2006: Developments continued with introduction of all the previous and new developments from FEM into G3 Framework, later renamed OpenSEES. at PEER, using C++ Programming language, on PC-Linux, by Professor Boris Jeremić and co-workers at the University of California at Davis, CA, USA, see Final Report Presentation.

2006-Present: Development of the Real-ESSI Simulator System (aka Real-ESSI, MS-ESSI, NRC-ESSI), using C++, FORTRAN, FEI-DSL, Python Programming languages, on PC-Linux, by Professor Boris Jeremić and co-workers at UCD. For details see main Real-ESSI Simulator web site or/and real-essi.us or/and real-essi.info (they all point to the same URL),

Appendix 709

Computer Programs for ESSI Analysis

(2019-)

This section lists a number of available computer programs, commercial, Open Source, Public Domain and Open Use, that can be used and are used for performing Earthquake Soil-Structure Interaction (ESSI) analysis. Focus is on presenting information about programs without much critical assessment of programs capabilities for ESSI analysis.

709.1 Overview of Available ESSI Analysis Programs

This section is based in part on material from Pecker et al. (2022).

709.1.1 Program Distribution Methods

Briefly described here are method for distribution of programs, in source code form or in executable form

- Commercial programs (CP): Distributed, sold, made available by commercial companies. It is very important to note that these companies need to earn funds to support company, staff, etc. Commercial programs usually have features and capabilities that are defined by a commercial license. Commercial license content is usually controlled and written by company lawyers. Commercial programs usually guaranty good accuracy of examples provided in the manual. These example usually show good, nice comparison or results with some, carefully chosen analytic solution. Publicly available, accessible Verification and Validation (V&V) for commercial programs is usually not available. One of the reasons for this, as privately noted by one of principal engineers from one of big software companies, is that verification will document level of error for approximate, numerical methods, however these errors are small, for elements, algorithms. These errors of implemented numerical approximation methods are not deemed good for business. It is reasonable to assume that commercial programs do have a significant V&V effort and documentation...
- Open Source programs (OSP): Distributed online by developers, covered by one of the open source licenses (OSL): (a) General Public License (GPL), (b) Lesser General Public License (LGPL), (c) Creative Commons (CC)) The OSL guarantees that software source code and derivative source code will be always available through similar OSL. The OSL does not even attempt to provide any quality assurance for the quality of program due to legal reasons, liability. The quality assurance (QA) for a given program, is usually a separate effort. It is noted that QA for OSL programs is almost impossible, as anyone can obtain a source code for a program, make changes to program sources, that can possibly destroy any previous QA and V&V effort and present results as using the same program...

- Restricted Source programs (RSP): Distributed to select developers, users, using a restricted version of an open source license. The difference is that developers and program owners can restrict source code distribution, mostly due to intellectual property reasons. A version of OSL is used, usually a revised version of CC license. Quality Assurance with restricted source programs is easier, as the main developer, program owner, quality assurance maintainer controls program sources distribution and can, therefor, main control of the QA process.
- Open Use programs (OUS): Distributed are executable versions of the program. The program owner can place limitations on use of the program. The quality assurance (QA) is controlled by the program owner, distributor. The QA, if it exists, is easily mainteined.
- Public Domain programs (PDP): Distributed are source code and/or executable without any restrictions for any future use. Original developer and owner of the program releases all the rights to the program sources and executables for any future use.

709.1.2 Available Programs

Provided is an incomplete list of programs that can be used and are used for ESSI analysis, or part of the ESSI analysis. These programs are available using one of the distribution methods as noted in previous section 709.1.1 on page 3080.

- Commercial Programs:
  - ABAQUS (`http://www.3ds.com`)
  - ADINA (`http://www.adina.com`)
  - ANSYS (`http://www.ansys.com/`)
  - CLASSI ()
  - GT STRUDL (`https://hexagonppm.com/offerings/products/gt-strudl`)
  - LS-DYNA (`http://www.lstc.com`)
  - NASTRAN (`http://www.mscsoftware.com`)
  - RIGID ()
  - SAP2000 (`https://www.csiamerica.com`)
  - SASSI 2010 (`http://sassi2000.net`)
  - ACS SASSI (`http://www.ghiocel-tech.com`)

  - SMACS ()
  - STARDYNE (`ftp://ftp.cray.com`)
  - SOFISTIK (`http://www.sofistik.com`)
  - PLAXIS (`http://www.plaxis.nl/`)
  - FLAC (`http://www.itascacg.com`)
  - DYNAFLOW (`https://blogs.princeton.edu/prevost/dynaflow/`)
  - Zsoil (`http://www.zsoil.com`)
  - Real-ESSI (`http://real-essi.us/`, `http://essi-consultants.com`)

- Open Source programs, Restricted Source programs, and Open Use programs:
  - FEAP (`http://www.ce.berkeley.edu`)
  - DEEPSOIL (`http://deepsoil.cee.illinois.edu/`)
  - SIMQKE1 (`http://nisee.berkeley.edu/`)
  - OpenSees (`http://opensees.berkeley.edu/`)
  - Code_ASTER (`http://www.code_astair.org`)
  - Real-ESSI (`http://real-essi.us/`)

- Public Domain programs:
  - SHAKE91 (`http://nisee.berkeley.edu/`)
  - EERA and NEERA (`http://www.ce.memphis.edu/`)
  - DESRA-2 ()
  - SUMDES ()
  - D-MOD ()
  - TESS ()
  - OpenSees (`http://opensees.berkeley.edu/`)

Appendix 710

Work Organization

(1989-)

This section describes in some detail work organization related to the development of FEI modeling and computational system.

710.1 Communication

Tablets, smart phones, laptops and computers, using `https://zoom.us/` as it works on linux and all other OSs.

710.2 Writing (Notes, Code, &c.) Version Control

710.2.1 Source Code

Memory Leaks Memory leaks are best discovered by running Valgrind (`http://valgrind.org/`). There are a number of tools that can be used with Valgrind. Mentioned are some of the most important ones, with example commands[1]

use of `tcsh` is assumed, with a time stamp (used in commands below) set as: `` set TIMESTAMP = `date +%h_%d_%Y_%Hh_%Mm_%Ss__%A` ``

-
  ```
  (time valgrind --tool=cachegrind $argv[1] >! $argv[1].cachegrind.$TIMESTAMP.out)>&! ↩
  $argv[1].cachegrind.$TIMESTAMP.err
  ```

-
  ```
  (time valgrind --tool=callgrind $argv[1] >! $argv[1].callgrind.$TIMESTAMP.out)>&! ↩
  $argv[1].callgrind.$TIMESTAMP.err
  ```

-
  ```
  (time valgrind --tool=massif $argv[1] >! $argv[1].massif.$TIMESTAMP.out)>&! ↩
  $argv[1].massif.$TIMESTAMP.err
  ```

-
  ```
  (time valgrind --tool=memcheck --leak-check=full --show-reachable=yes --freelist-vol=10000
  $argv[1] >! $argv[1].memcheck.$TIMESTAMP.out)>&! $argv[1].memcheck.$TIMESTAMP.err
  ```

-
  ```
  valgrind -v --leak-check=yes --show-reachable=yes --num-callers=32 --trace-malloc=yes ↩
  --error-limit=no --tool=massif $argv[1]
  ```

---

[1] Examples use synthax from few years ago, so should be proper synthax should be verified using excellent Valgrind documentation.

710.2.2 Verification of Real-ESSI

The aim is to run the verification procedure for Real-ESSI as automatically as possible. The verification of Real-ESSI is based on the verification of C++ libraries by https://www.boost.org/.

The verification is divided into 3 parts:

1. verification of `essi.sequential`, run by calling bash script

   ```
   ESSI_VERIFICATION_run_all_verification_SEQUENTIAL.sh
   ```

2. verification of `essi.parallel`, run by calling bash script

   ```
   ESSI_VERIFICATION_run_all_verification_PARALLEL.sh
   ```

3. check of the code stability, run by calling bash script

   `ESSI_VERIFICATION_run_CODE_STABILITY.sh`.

710.2.2.1 Update of the verification procedure from 2019

The following was done in `.../oofep/Rad_na_cml04/GLOBAL_RELEASE/Real-ESSI-Examples`.

1. In `*.fei` files, variable `Gamma` was replaced by `GammaParam` because `Gamma` is a keyword. The following was used

   ```
   grep -rl --include \*.fei ’Gamma’ * | xargs -i@ sed -i ’s/Gamma/GammaParam/g’ @
   ```

   and then

   ```
   grep -rl --include $\backslash$*.fei 'ShearStrainGammaParam' * $\vert$ xargs ↩
   -i@ sed -i 's/ShearStrainGammaParam/ShearStrainGamma/g' @
   ```

2. `Beta` was replaced by `BetaParam` using

   ```
   grep -rl --include \*.fei ’Beta’ * | xargs -i@ sed -i ’s/Beta/BetaParam/g’ @
   ```

3. 2TO3 converter was used to convert the `*.py` files from PYTHON2 to PYTHON3 using

   ```
   cd .../oofep/Rad_na_cml04/GLOBAL_RELEASE/Real-ESSI-Examples/
   ```

   and then

   ```
   2to3 -w .
   ```

   Before that, 2TO3 was installed as follows

   ```
   sudo apt install 2to3
   sudo apt install python3-lib2to3
   sudo apt install python3-toolz
   ```

4. During the evaluation of dynamic examples in `../Real-ESSI-Examples/dynamic_test`, warning:

   ```
   DeprecationWarning: Please use ‘fftfreq‘ from the ‘scipy.fftpack‘ namespace,
   the ‘scipy.fftpack.helper‘ namespace is deprecated.
   ```

   was returned, so in `../Real-ESSI-Examples/dynamic_test`, the following was done:

   ```
   grep -rl --include \*.py 'scipy.fftpack.helper' * | xargs -i@ sed -i 's/scipy.fftpack.helper/scipy.fftpack/g'
   @
   ```

5. During the evaluation of dynamic examples in `../Real-ESSI-Examples/dynamic_test`, an error was returned:

   ```
   xi, fs, Ys = measure_damping(f[0:N/2], abs(D[0:N/2]))
   TypeError: slice indices must be integers or None or have an __index__ method Solution
   ```

   so in `../Real-ESSI-Examples/dynamic_test`, the following was done:

   ```
   grep -rl --include \*.py 'N/2]' * | xargs -i@ sed -i 's#N/2]#N//2]#g' @
   ```

6. During the evaluation of dynamic examples in `../Real-ESSI-Examples/dynamic_test`, an error was returned:

   ```
   runall.sh: line 28: cd: */: No such file or directory
   ```

   Examples in all subfolders are evaluated by `runall.sh`. The error pertains to the folder __pycache__. I added `__pycache__.fei` (with just `bye;` inside), in folder __pycache__.

7. `ESSI_VERIFICATION_run_all_verification_SEQUENTIAL.sh` and `ESSI_VERIFICATION_run_all_verification_PARALLEL.sh` were modified.

710.2.3 Lecture Notes

Maintain lecture notes using git on https://github.com/.

Checking all http links in lecture notes using script `ESSI_check_URLs_in_lecture_notes.sh` in bin.

710.2.4 Bibliography

Bibliography List.

Papers of interest are organized in bibtex files (managed through *git* version control.

A list of those paper is compiled and available at:

http://sokocalo.engr.ucdavis.edu/~jeremic/research/Jeremic_et_al_bibliography_mechanics.pdf

http://sokocalo.engr.ucdavis.edu/~jeremic/research/Jeremic_et_al_bibliography_computers.pdf

http://sokocalo.engr.ucdavis.edu/~jeremic/research/Jeremic_et_al_bibliography_education.pdf

Bibliography Repository.

Most listed papers are available at: http://sokocalo.engr.ucdavis.edu/~jeremic/PAPERSlocalREPO/. This site is only accessible to members of the Computational Mechanics group at University of California at Davis, and few other collaborating entities.

710.3 Backup

710.4 Calendar

710.5 Useful Programs and Scripts

710.5.1 Backup Scripts

710.5.2 Domain Reduction Method Processing Programs and Scripts

DRM Node Extraction for fk.

fk Output Processing for DRM.

710.5.3 Pre Processing Programs and Scripts

710.5.4 Post Processing Programs and Scripts

710.5.5 Parallel Computer Architecture

http://www.open-mpi.org/projects/hwloc/

Appendix 711

Collected Bibliography

Compilation of all collected bibliography, over years, not necessarily cited in this book.

Bibliography

by:

Jeremić CompMech Group

Department of Civil and Environmental Engineering

University of California, Davis

Bibliography

[A.A03] A.Angabini. Anisotropy of rock elasticity behavior and of gas migration in a Variscan Carboniferous rock mass in the South Limburg, The Netherlands. *Engineering Geology*, 67(3-4):353–372, 2003.

[AA06] Khalid A. Alshibli and Bashar A. Alramahi. Microscopic evaluation of strain distribution in granular materials during shear. *Journal of Geotechnical and Geoenvironmental Engineering*, 132(1):80–91, 2006.

[AA14] Stephen D Antolovich and Ronald W Armstrong. Plastic strain localization in metals: origins and consequences. *Progress in Materials Science*, 59:1–160, 2014.

[AA19] Carlos A Arteta and Norman A Abrahamson. Conditional scenario spectra (CSS) for hazard-consistent analysis of engineering systems. *Earthquake Spectra*, 35(2):737–757, 2019.

[AAB^{+}10] Linda Al Atik, Norman Abrahamson, Julian J Bommer, Frank Scherbaum, Fabrice Cotton, and Nicolas Kuehn. The variability of ground-motion prediction models and its components. *Seismological Research Letters*, 81(5):794–801, 2010.

[AAdV07] E. Artiolib, F. Auricchiob, and L. Beirão da Veiga. Second-order accurate integration algorithms for von-Mises plasticity with a nonlinear kinematic hardening mechanism. *Computer Methods in Applied Mechanics and Engineering*, 196(9-12):1827–1846, February 2007.

[AAGW11] Julian M. Allwood, Michael F. Ashby, Timothy G. Gutowski, and Ernst Worrell. Material efficiency: A white paper. *Resources, Conservation and Recycling*, 55(3):362–381, 2011.

[AAM09] Yoshikazu Araki, Takehiko Asai, and Takeshi Masui. Vertical vibration isolator having piecewise-constant restoring force. *Earthquake Engineering & Structural Dynamics*, 38(13):1505–1523, 2009.

[AB78] J. H. Atkinson and P. L. Bransby. *The Mechanics of Soils, An Introduction to Critical Stat Soil Mechanics*. McGraw - Hill Book Company, 1978.

[AB95] Gail M Atkinson and David M Boore. Ground-motion relations for eastern north america. *Bulletin of the Seismological Society of America*, 85(1):17–30, 1995.

[AB99] John G Anderson and James N Brune. Probabilistic seismic hazard analysis without the ergodic assumption. *Seismological Research Letters*, 70(1):19–28, 1999.

[AB05] Norman A Abrahamson and Julian J Bommer. Probability and uncertainty in seismic hazard analysis. *Earthquake spectra*, 21(2):603–607, 2005.

[AB06] José E. Andrade and Ronaldo I. Borja. Capturing strain localization in dense sands with random density. *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, 67(11):1531–1564, 2006.

[ABB^{+}99] Edward Anderson, Zhaojun Bai, Christian Bischof, L Susan Blackford, James Demmel, Jack Dongarra, Jeremy Du Croz, Anne Greenbaum, Sven Hammarling, Alan McKenney, et al. *LAPACK Users' guide*. SIAM, 1999.

[ABD^{+}90] Edward Anderson, Zhaojun Bai, Jack Dongarra, Anne Greenbaum, Alan McKenney, Jeremy Du Croz, Sven Hammarling, James Demmel, C Bischof, and Danny Sorensen. Lapack: A portable linear algebra library for high-performance computers. In *Proceedings of the 1990 ACM/IEEE conference on Supercomputing*, pages 2–11. IEEE Computer Society Press, 1990.

[ABdL16a] A. Amorosi, D. Boldini, and A. di Lernia. Seismic ground response at lotung: Hysteretic elasto-plastic-based 3D analyses. *Soil Dynamics and Earthquake Engineering*, 85:44 – 61, 2016.

[ABdL16b] A. Amorosi, D. Boldini, and A. di Lernia. Seismic ground response at lotung: Hysteretic elasto-plastic-based 3D analyses. *Soil Dynamics and Earthquake Engineering*, 85:44 – 61, 2016.

[ABKS99] Rajendram Arulnathan, Ross W. Boulanger, Bruce L. Kutter, and Bill Sluis. A new tool for V_s measurments in model tests. *Submitted to the ASTM Journal*, 1999.

[Abr65] J. Abram. *Tensor Calculus through Differential Geometry*. Butterworths & Co. (Publishers) Ltd., London 88 Kingsway, W.C.2, 1965.

[Abr85] Norman Alan Abrahamson. Estimation of seismic wave coherency and rupture velocity using the smart 1 strong-motion array recordings. Technical Report EERC-85-02, Earthquake Engineering Research Center, University of California, Berkeley, 1985.

[Abr92a] N. Abrahamson. Spatial variation of earthquake ground motion for application to soil-structure interaction. *EPRI Report No. TR-100463, March.*, 1992.

[Abr92b] N.A. Abrahamson. Generation of spatially incoherent strong motion time histories. In Alberto Bernal, editor, *Earthquake Engineering, Tenth World Conference*, pages 845–850. Balkema, Rotterdam, 1992. ISBN 90 5410 060 5.

[Abr93] N.A. Abrahamson. Spatial variation of multiple support inputs. In *Proceedings of the First U.S. Seminar, Seismic Evaluation and Retrofit of Steel Bridges*, San Francisco, CA, October 1993. UCB and CalTrans.

[ABR98] Rajendram Arulnathan, Ross W. Boulanger, and Michael F. Riemer. Analysis of bender element tests. *Geotechnical Testing Journal*, 21(2):120–131, 1998.

[Abr05] N. Abrahamson. Updated coherency model. *Report Prepared for Bechtel Corporation, April*, 2005.

[Abr06a] N. A. Abrahamson. Program on technology innovation: Spatial coherency models for soil-structure interaction. Technical report, EPRI, Palo Alto, CA, and U.S. Department of Energy, Washington, DC: 2006. 1012968, 2006.

[Abr06b] NA Abrahamson. Seismic hazard assessment: problems with current practice and future developments. In *First European conference on earthquake engineering and seismology*, pages 3–8, 2006.

[Abr07a] Norman A. Abrahamson. Hard rock coherency functions based on Pinyon Flat data. unpublished data report, 2007.

[Abr07b] Norman A. Abrahamson. Hard-rock coherency functions based on the pinyon flat array data. Technical report, EPRI, 05Jul 2007.

[Abr10] Norm Abrahamson. Sigma components: Notation & initial action items. In *Proceedings of NGA-East "Sigma" Workshop*, University of California, Berkeley, February 2010. Pacific Earthquake Engineering Research Center. `http://peer.berkeley.edu/ngaeast/2010/02/sigma-workshop/`.

[Abr18] Norman A Abrahamson. What changes to expect in seismic hazard analyses in the next 5 years, 2018. Plenary talk at the 11th U.S. National Conference on Earthquake Engineering, Los Angeles, United States.

[ABV69] A. N. Afanasenkov, V. M. Bogomolov, and I. M. Voskoboinikov. Generalized shock Hugoniot of condensed substances. *Journal of Applied Mechanics and Technical Physics*, 10(4):660–664, July 1969.

[AC96] A. Anandarajah and J. Chen. Macroscopic constitutivebehavior of clays from microscopic considerations. In Y. K. Lin and T. C. Su, editors, *Proceedings of 11th Conference*, pages 709–712. Engineering Mechanics Division of the American Society of Civil Engineers, May 1996.

[AC20] Guido Andreotti and Gian Michele Calvi. Design of laterally loaded pile-columns considering ssi effects: Strengths and weaknesses of 3d, 2d, and 1D nonlinear analysis. *Earthquake Engineering & Structural Dynamics*, n/a(n/a):26, 2020.

[ACC+17] I. Antoniadis, E. Chatzi, D. Chronopoulos, A. Paradeisiotis, I. Sapountzakis, and S. Konstantopoulos. Low-frequency wide band-gap elastic/acoustic meta-materials using the k-damping concept, 2017.

[ACD+] Per-Erik Austrell, H/' akan Carlsson, Ola Dahlblom, Jonas Lindemann, Karl-Gunnar Olsson, Anders Peterson, Hans Petersson, Matti Ristinmaa, and Göran Sandberg. Computer added learning of the finite element method CALFEM. `http://www.byggmek.lth.se/Calfem/` Division of Structural Mechanics and Division of Solid Mechanics Lund University, Sweden.

[Ach73] J. D. Achenbach. *Wave Propagation in Elastic Solids*. Applied Mathematics and Mechanics. North Holland Publishing Company, 1973.

[ACLF13] Mathieu Acher, Philippe Collet, Philippe Lahire, and Robert B. France. Familiar: A domain-specific language for large scale management of feature models. *Science of Computer Programming*, 78(6):657 – 681, 2013. Special section: The Programming Languages track at the 26th ACM Symposium on Applied Computing (SAC 2011) & Special section on Agent-oriented Design Methods and Programming Techniques for Distributed Computing in Dynamic and Complex Environments.

[AD87] A. Anandarajah and Yannis F. Dafalias. Bounding surface plasticity. III: Application to anisotropic cohesive soils. *ASCE Journal of Engineering Mechanics*, 112(12):1292–1318, December 1987.

[AD12] Sinan Acikgoz and Matthew J. DeJong. The interaction of elasticity and rocking in flexible structures allowed to uplift. *Earthquake Engineering & Structural Dynamics*, 41(15):2177–2194, 2012.

[AD19a] Mohamad Essa Alyounis and Chandrakant S. Desai. Testing and modeling of saturated interfaces with effect of surface roughness. I: Test behavior. *International Journal of Geomechanics*, 19(8):04019096, 2019.

[AD19b] Mohamad Essa Alyounis and Chandrakant S. Desai. Testing and modeling of saturated interfaces with effect of surface roughness. II: Modeling and validations. *International Journal of Geomechanics*, 19(8):04019097, 2019.

[Ada95a] G. G. Adams. Self–excited oscillations in sliding with a constant coefficient of friction – a simple model. *ASME Journal of Tribology*, 118:819–823, 1995.

[Ada95b] G. G. Adams. Self–excited oscillations of two elastic half–spaces sliding with a constant coefficient of friction. *ASME Journal of Applied Mechanics*, 62:867–872, 1995.

[Ada97] G. G. Adams. Dynamics instabilites in the sliding of two layered elastic half–spaces. *ASME Journal of Tribology*, 120:289–295, 1997.

[Add95] Daniel Averbuch, Patrick de Buhan, and Gianmmarco de Felice. Finite elements for limit analysis of homogenized block masonry walls. In Stein Sture, editor, *Proceedings of 10th Conference*, pages 891–894. Engineering Mechanics Division of the American Society of Civil Engineers, May 1995.

[ADD17] Thibaut Abergel, Brian Dean, and John Dulac. Towards a zero-emission, efficient, and resilient buildings and construction sector, global status report 2017. Technical report, United Nations, Environment Programme, 2017.

[ADF$^+$01] G. Allen, T. Dramlitsch, I. Foster, N. Karonis, M. Ripeanu, E. Seidel, and B. Toonen. Supporting efficient execution in heterogeneous distributed computing environments with cactus and globus. In *Proceedings of Super Computing 2001*, November 10-16 2001.

[AdH99] Harm Askes, René de Borst, and Otto Heeres. Conditions for locking–free elasto–plastic analysis in the element–free galerkin method. *International Journal for Computer Methods in Applied Mechanics and Engineering*, 159:99–109, 1999.

[AdHF92] Ken Alvin, Horacio M. de la Fuente, Bjørn Haugen, and Carlos A. Felippa. Membrane triangles with corner drilling freedoms – I. the EFF element. *Finite Elements in Analysis and Design*, 12:163–187, 1992.

[AE23] Mohamed Abouyoussef and Mohamed Ezzeldin. Fragility and economic evaluations of high-strength reinforced concrete shear walls in nuclear power plants. *Journal of Structural Engineering*, 149(5):04023035, 2023.

[AEM98] Korhan Adalier, Ahmed-W Elgamal, and Goeffrey R. Martin. Foundation liquefaction countermeasures for earth embankments. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 124(6):500–517, June 1998.

[AF66] P.J. Armstrong and C.O. Frederick. A mathematical representation of the multiaxial Bauschinger effect. Technical Report RD/B/N/ 731,, C.E.G.B., 1966.

[AF23] Faisal As'ad and Charbel Farhat. A mechanics-informed deep learning framework for data-driven nonlinear viscoelasticity. *Computer Methods in Applied Mechanics and Engineering*, 417:116463, 2023.

[AFT99] G. C. Archer, G. Fenves, and C. Thewalt. A new object–oriented finite element analysis program architecture. *Computers and Structures*, 70(1):63–75, 1999.

[AG82] B. Amadei and R. E. Goodman. The influence of rock anisotropy on stress measurements by overcoring techniques. *Rock Mechanics*, 15:167–180, December 1982. 10.1007/BF01240588.

[AG96a] F. Armero and K. Garikipati. An analysis of strong–discontinuities in inelastic solids with application to the finite element simulation of strain localization problems. In Y. K. Lin and T. C. Su, editors, *Proceedings of 11th Conference*, pages 136–139. Engineering Mechanics Division of the American Society of Civil Engineers, May 1996.

[AG96b] F. Armero and K. Garikipati. An analysis of strong discontinuity in multiplicative finite strain plasticity and their relation with the numerical simulation of strain localization in solids. *International Journal of Solids and Structures*, 33(20-22):2863–2885, 1996.

[AG99] Cleve Ashcraft and Roger G Grimes. Spooles: An object-oriented sparse matrix library. In *PPSC*, 1999.

[AG04] Sanjay R. Arwade and Mircea Grigoriu. Probabilistic model for polycrystalline microstructures with application to intergranular fracture. *ASCE Journal of Engineering Mechanics*, 130(9):997–1123, 2004.

[AG05] David Abrahams and Aleksey Gurtovoy. *C++ Template Metaprogramming: Concepts, Tools, and Techniques from Boost and Beyond*. C++ in Depth Series. Addison-Wesley., 2005.

[AG12] M. Arnst and R. Ghanem. A variational-inequality approach to stochastic boundary value problems with inequality constraints and its application to contact and elastoplasticity. *International Journal for Numerical Methods in Engineering*, 89(13):1665–1690, 2012.

[AG20] SOFiSTiK AG. Verification, ve56 interface element. Technical report, SOFISTIK AG, 2020.

[Agh85] Gul Agha. *ACTORS: A Model of Concurrent Computation in Distributed Systems*. PhD thesis, MIT, 1985. http://dspace.mit.edu/handle/1721.1/6952.

[AGP^{+}15] Grigorios Antonellis, Andreas G. Gavras, Marios Panagiotou, Bruce L. Kutter, Gabriele Guerrini, Andrew C. Sander, and Patrick J. Fox. Shake table test of large-scale bridge columns supported on rocking shallow foundations. *Journal of Geotechnical and Geoenvironmental Engineering*, 141(5):04015009, 2015.

[AGR^{+}10] Brad T. Aagaard, Robert W. Graves, Arthur Rodgers, Thomas M. Brocher, Robert W. Simpson, Douglas Dreger, N. Anders Petersson, Shawn C. Larsen, Shuo Ma, and Robert C. Jachens. Ground-motion modeling of hayward fault scenario earthquakes, part II: Simulation of long-period and broadband ground motions. *Bulletin of the Seismological Society of America*, 100(6):2945–2977, December 2010.

[AH84] John G Anderson and Susan E Hough. A model for the shape of the fourier amplitude spectrum of acceleration at high frequencies. *Bulletin of the Seismological Society of America*, 74(5):1969–1993, 1984.

[AH99] Maciej Anders and Muneo Hori. Stochastic finite element method for elasto–plastic body. *International Journal for Numerical Methods in Engineering*, 46:1897–1916, 1999.

[AH01] Maciej Anders and Muneo Hori. Tree-dimensional stochastic finite element method for elasto-plastic bodies. *International Journal for Numerical Methods in Engineering*, 51:449–478, 2001.

[AH06] M Arnold and I Herle. Hypoplastic description of the frictional behaviour of contacts. *Numerical methods in geotechnical engineering*, pages 101–6, 2006.

[AHL12] S. Atamturktur, F.M. Hemez, and J.A. Laman. Uncertainty quantification in model verification and validation as applied to large scale historic masonry monuments. *Engineering Structures*, 43:221–234, 2012.

[AHLM82] D Aubry, JC Hujeux, F Lassoudiere, and Y Meimon. A double memory model with multiple mechanisms for cyclic soil behaviour. In *Proceedings of the Int. Symp. Num. Mod. Geomech*, pages 3–13, 1982.

[AHM^{+}24] Filip P. Adamus, David Healy, Philip G. Meredith, Thomas M. Mitchell, and Ashley Stanton-Yonge. Multi-porous extension of anisotropic poroelasticity: Consolidation and related coefficients. *International Journal for Numerical and Analytical Methods in Geomechanics*, n/a(n/a), 2024.

[AHMM24] Filip P. Adamus, David Healy, Philip G. Meredith, and Thomas M. Mitchell. Multi-porous extension of anisotropic poroelasticity: Linkage with micromechanics. *International Journal for Numerical and Analytical Methods in Geomechanics*, n/a(n/a), 2024.

[AHZH24] Karim AlKhatib, Youssef M. A. Hashash, Katerina Ziotopoulou, and James Heins. Centrifuge and numerical modeling of the seismic response of buried water supply reservoirs. *Journal of Geotechnical and Geoenvironmental Engineering*, 150(3):04023141, 2024.

[AHZM23] Karim AlKhatib, Youssef M. A. Hashash, Katerina Ziotopoulou, and Brian Morales. Hydrodynamic pressures on rigid walls subjected to cyclic and seismic ground motions. *Earthquake Engineering & Structural Dynamics*, n/a(n/a):24pages, 2023.

[AI02] Fadel Alameddine and Roy A. Imbsen. Rocking of bridge piers under earthquake loading. In *Proceedings of the Third National Seismic Conference & Workshop on Bridges and Higways*, 2002.

[AJ22] A. H. Amjadi and A. Johari. Stochastic nonlinear ground response analysis considering existing boreholes locations by the geostatistical method. *Bulletin of Earthquake Engineering*, 20(5):2285–2327, 2022.

[AK07] Dominic Assimaki and Eduardo Kausel. Modified topographic amplification factors for a single-faced slope due to kinematic soil-structure interaction. *ASCE Journal Geotechnical and Geoenvironmental Engineering*, 133(11):1414–1431, November 2007.

[AK14] J.D. Allmond and B.L. Kutter. Fluid effects on rocking foundations in difficult soil. In *Tenth U.S. National Conference on Earthquake EngineeringFrontiers of Earthquake Engineering*, July 2014.

[AKL18] Marreddy Ambati, Josef Kiendl, and Laura De Lorenzis. Isogeometric Kirchhoff-Love shell formulation for elasto-plasticity. *Computer Methods in Applied Mechanics and Engineering*, 340:320 – 339, 2018.

[AKW$^+$22] Giulia Aguzzi, Constantinos Kanellopoulos, Richard Wiltshaw, Richard Craster, Eleni Chatzi, and Andrea Colombi. Octet lattice-based plate for elastic wave control. *Scientific Reports*, 12, 01 2022.

[Al-95] Khalid Al–Shibli. *Localized Deformation in Granular Materials*. PhD thesis, University of Colorado at Boulder, 1995.

[AL95] T. J. Ahrens and C. Liu. Loss of high frequency upon propagation through shock-damaged rock. Technical report, Seismological Laboratory, California Institute of Technology, Pasadena, 1995.

[AL03] Andrei V. Abelev and Poul V. Lade. Effects of cross anisotropy on three–dimensional behavior of sand. : I stress– strain behavior and shear banding. *ASCE Journal of Engineering Mechanics*, 129(2):160–166, February 2003.

[AL04] Andrei V. Abelev and Poul V. Lade. Characterization of failure in cross-anisotropic soils. *ASCE Journal of Engineering Mechanics*, 130(5):599–606, 2004.

[Ala90] Husssain A. S. Alawaji. *Formulation and Integration of Constitutive Relations in Soil Plasticity Under Mixed Control for Drained and Undrained Control*. PhD thesis, University of Colorado ant Boulder, Campus Box 428, Boulder, CO, 80309, 1990.

[Ala] Fadel Alameddine. Private communications. ..., 2003 –.

[ALB99] Bernt S. Aadnoy, Kenneth Larsen, and Per C. Berg. Analysis of stuck pipe in deviated boreholes. In *Proceedings - SPE Annual Technical Conference and Exhibition*, volume 2, pages 35–49, Richardson, TX, USA., 1999. Soc Pet Eng (SPE).

[Ale06] Sergei Alexandrov. Steady penetration of a rigid cone into pressure-dependent plastic material. *International Journal of Solids and Structures*, 43(2):193–205, January 2006.

[ALG03] Tarabay H. Antoun, Ilya N. Lomov, and Lewis A. Glenn. Simulation of the penetration of a sequence of bombs into granitic rock. *International Journal of Impact Engineering*, 29(1-10):81–94, December 2003.

[All85] Junius Allen. *$p-y$ Curves in Layered Soils*. PhD thesis, The University of Texas at Austin, May 1985.

[ALM$^+$09] Martin Sandve Alnaes, Anders Logg, Kent-Andre Mardal, Ola Skavhaug, and Hans Petter Langtangen. Unified framework for finite element assembly. *International Journal of Computational Science and Engineering*, 4(4):231–244, 2009.

[ALMO87] A. Agah–Tehrani, E. H. Lee, R. L. Mallet, and E. T. Onat. The theory of elastic plastic deformation at finite strain with induced anisotropy modeled as combined isotropic–kinematic hardening. *Journal of Mechanics and Physics of Solids*, 35(5):519–539, 1987.

[ALO$^+$14] Martin S. Alnæs, Anders Logg, Kristian B. Olgaard, Marie E. Rognes, and Garth N. Wells. Unified form language: A domain-specific language for weak formulations of partial differential equations. *ACM Trans. Math. Softw.*, 40(2):9:1–9:37, mar 2014.

[ALS$^+$03] Ralph J Archuleta, Pengcheng Liu, Jamison H Steidl, L Fabian Bonilla, Daniel Lavallée, and Francois Heuze. Finite-fault site-specific acceleration time histories that include nonlinear soil response. *Physics of the Earth and Planetary Interiors*, 137(1-4):153–181, 2003.

[AM91] John Argyris and Hans-Peter Mlejnek. *Dynamics of Structures*. North Holland in USA Elsevier, 1991.

[AM96] Pedro Arduino and Emir J. Macari. Multiphase flow in deforming porous media by the finite element method. In Y. K. Lin and T. C. Su, editors, *Proceedings of 11th Conference*, pages 420–423. Engineering Mechanics Division of the American Society of Civil Engineers, May 1996.

[AM01a] Pedro Arduino and Emir Jose Macari. Implementation of porous media formulation for geomaterials. *ASCE Journal of Engineering Mechanics*, 127(2):157–166, 2001.

[AM01b] Pedro Arduino and Emir Jose Macari. Numerical analysis of geomaterials within theory of porous media. *ASCE Journal of Engineering Mechanics*, 127(2):167–175, 2001.

[AM06] I.E. Avramidis and K. Morfidis. Bending of beams on three-parameter elastic foundation. *International Journal of Solids and Structures*, 43(2):357–375, January 2006.

[AM12] Martin Sandve Alnaes and Kent-Andre Mardal. Syfi and sfc: Symbolic finite elements and form compilation. In *Automated Solution of Differential Equations by the Finite Element Method*, pages 273–282. Springer, 2012.

[AM18] Domniki Asimaki and Kami Mohammadi. On the complexity of seismic waves trapped in irregular topographies. *Soil Dynamics and Earthquake Engineering*, 114:424 – 437, 2018.

[Ama83] Bernard Amadei. *Rock anisotropy and the theory of stress measurements*. Lecture notes in engineering. Springer-Verlag, 1983.

[Ame13] American Society of Civil Engineers. *Minimum design loads for buildings and other structures, ASCE/SEI 7-10*, volume 7. American Society of Civil Engineers, 2013.

[Amh67] Gene M. Amhdal. Validity of the single processor approach to achieving large scale computing capabilities. In *AFIPS Conference Proceedings*, volume 30, pages 483–485, 1967.

[AMKRWH25] Mojtaba Aliasghar-Mamaghani, Ioannis Koutromanos, Carin Roberts-Wollmann, and Matthew Hebdon. Multiphysics modeling of chloride-induced corrosion damage in concrete structures. *Computers & Structures*, 308:107643, 2025.

[AMM90] D Aubry, A Modaressi, and H Modaressi. A constitutive model for cyclic behaviour of interfaces with variable dilatancy. *Computers and Geotechnics*, 9(1-2):47–58, 1990.

[AMR88] Ralph Abraham, J.E. Marsden, and Tudor Ratiu. *Manifolds, Tensor Analysis, and Applications*, volume 75 of *Applied Mathematical Sciences*. Springer Verlag, second edition, 1988.

[AMT$^+$21] G. Abbiati, S. Marelli, N. Tsokanas, B. Sudret, and B. Stojadinović. A global sensitivity analysis framework for hybrid simulation. *Mechanical Systems and Signal Processing*, 146:106997, 2021.

[AMY97] Kandiah Arulanandan, Kanthasamy K. Muraleetharan, and Chelvarajah Yogachandran. Sesimic response of soil deposits in San Fransisco marina district. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 123(10):965–974, October 1997.

[AN00] M. Ashour and G. Norris. Modeling lateral soil-pile response based on soil-pile interaction. *Journal of Geotechnical and Geoenvironmental Engineering*, 126(5):420–428, May 2000.

[AN03] Mohamed Ashour and Gary Norris. Lateral loaded pile response in liquefiable soil. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 129(6):404–414, June 2003.

[AN05] Anoosh Shamsabadi 1, Mohamed Ashour, and Gary Norris. Bridge abutment nonlinear force-displacement-capacity prediction for seismic design. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 131(2):141–272, 2005.

[Anc10] Timothy David Ancheta. *Engineering characterization of spatially variable earthquake ground motions*. PhD thesis, University of California at Los Angeles, 2010.

[And03] Donald L. Anderson. Effect of foundation rocking on the seismic response of shear walls. *Canadian Civil Engineering Journal*, 2003.

[And04] John G. Anderson. Quantitative measure of the goodness-of-fit of synthetic seismograms. In *13th World Conference on Earthquake Engineering*, Vancouver, B.C., Canada, August 1-6 2004.

[And08] T.H. van Andel. *New Views on an Old Planet*. Cambridge University Press, Cambridge, 2 edition, 2008.

[And10] John G. Anderson. Source and site characteristics of earthquakes that have caused exceptional ground accelerations and velocities. *Bulletin of the Seismological Society of America*, 100(1):1–36, 02 2010.

[ANS95] ANSI/ISO, Washington DC. *Working Paper for Draft Proposed International Standard for Information Systems–Programming Language C++*, April 1995. Doc. No. ANSI X3J16/95-0087 ISO WG21/N0687.

[ANS03] ANSI C++ Standard Committee. *Information Technology - Programming Languages - C++*. ANSI ISO/IEC 14882-2003, 2003. `http://reality.sgi.com/austern_mti/std-c++/faq.html`

[AO82] T. Adachi and F. Oka. Constitutive equations for sands and over–consolidated clays, and assigned works for sand. In *Results of the International Workshop on Constitutive Relations for Soils*, pages 141–157, Grenoble, 1982.

[AOMJ18] José A. Abell, Nebojša Orbović, David B. McCallen, and Boris Jeremić. Earthquake soil structure interaction of nuclear power plants, differences in response to 3-D, 3×1-D, and 1-D excitations. *Earthquake Engineering and Structural Dynamics*, 47(6):1478–1495, May 2018.

[AP04] Farzad Abedzadeh and Ronald Y. S. Pak. Continuum mechanics of lateral soil-pile interaction. *ASCE Journal of Engineering Mechanics*, 130(11):1309–1318, November 2004.

[AP19] Sebastian Andersen and Peter Noe Poulsen. A Taylor basis for kinematic nonlinear real-time simulations. Part I: The complete modal derivatives. *Earthquake Engineering & Structural Dynamics*, 0(0), 2019.

[APDdBL13] D. Akçay Perdahcioğlu, M. Doreille, A. de Boer, and T. Ludwig. Coupling of non-conforming meshes in a component mode synthesis method. *International Journal for Numerical Methods in Engineering*, pages n/a–n/a, 2013.

[APDMD12] Wahib Arairo, Florent Prunier, Irini Djéran-Maigre, and Félix Darve. A new insight into modelling the behaviour of unsaturated soils. *International Journal for Numerical and Analytical Methods in Geomechanics*, pages n/a–n/a, 2012.

[APF02] F. Armero and A. Pérez-Foguet. On the formulation of closest-point projection algorithms in elastoplasticity-part i: The variational structure. *International Journal for Numerical Methods in Engineering*, 53(2):297–329, 2002.

[APLN15] Vladimir Andjelković, Nenad Pavlović, Žarko Lazarević, and Velimir Nedović. Modelling of shear characteristics at the concrete-rock mass interface. *International Journal of Rock Mechanics and Mining Sciences*, 76:222 – 236, 2015.

[APN04] M. Ashour, P. Pilling, and G. Norris. Lateral behavior of pile groups in layered soils. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 130(6):580–592, June 2004.

[App18] Applied Technology Council . Examples application guide for ASCE/SEI 41-13 seismic evaluation and retrofit of existing buildings with additional commentary for ASCE/SEI 41-17. Technical report, Federal Emergency Management Agency, June 2018.

[APPP03] Dominic Assimaki, Alain Pecker, Radu Popescu, and Jean Prevost. Effects of spatial variability of soil properties on surface ground motion. *Journal of Earthquake Engineering*, 7(Special Issue No. 1):1–41, 2003.

[APRT$^+$22] Norman Abrahamson, Camilo Pinilla-Ramos, Payman Tehrani, Michael Perez, and Alex Krimotat. Modeling of vertical component ground motion for soil-structure-interaction analyses. In *SMiRT-26*, 2022.

[AR02] Keiiti Aki and Paul G. Richards. *Quantitative Seismology*. University Science Books, 2^{nd} edition, 2002.

[AR08] I. Agnolin and J.-N. Roux. On the elastic moduli of three-dimensional assemblies of spheres: Characterization and modeling of fluctuations in the particle displacement and rotation. *International Journal of Solids and Structures*, 45(3-4):1101–1123, 2008.

[Ara87] N. Aravas. On the numerical integration of a class of pressure–dependent plasticity models. *International Joournal for Numerical Methods in Engineering*, 24:1395–1416, 1987.

[Arc96] Graham Charles Archer. *Object Oriented Finite Analysis*. PhD thesis, University of California, Berkeley, 1996.

[Ari70] Arturo Arias. A measure of earthquake intensity. In Robert J. Hansen, editor, *Seismic Design for Nuclear Power Plants*, pages 438–483. The M.I.T. Press, Cambridge, Massachusetts, 1970.

[Arm99] F. Armero. Formulation and finite element implementation of a multiplicative model of coupled poro–plasticity at finite strains under fully saturated conditions. *Computer Methods in Applied Mechanics and Engineering*, 171:205–241, 1999.

[ARS^{+}] K. Axelsson, K. Runesson, S. Sture, Y. Yu, and H. Alawaji. Characteristics and integration of undrained response of silty soils. ——, pages 195–203, —-.

[ARSA92] H. Alawaji, K. Runesson, S. Sture, and K. Axelsson. Implicit integration in soil plasticity under mixed control for drained and, undrained response. *International Journal for Numerical Methods in Geomechanics*, 16:737–756, 1992.

[ARTH20] O. Ashraf, Andrei Rykhlevskii, G.V. Tikhomirov, and Kathryn D. Huff. Whole core analysis of the single-fluid double-zone thorium molten salt reactor (sd-tmsr). *Annals of Nuclear Energy*, 137:107115, 2020.

[AS93] Kandiah Arulanandan and Ronald F. Scott, editors. *Verification of Numerical Procedures for the Analysis of Soil Liquefaction Problems*. A. A. Balkema, 1993.

[AS95] A.J. Abbo and S.W. Sloan. A smooth hyperbolic approximation to the mohr-coulomb yield criterion. *Computers and Structures*, 54(3):427 – 441, 1995.

[AS96] Ricardo S. Avila and Lisa M. Sobierajski. A haptic interactive method for volume visualization. *IEEE Visualization*, ISBN 0-7803-3707(7/96), 1996.

[AS97a] NA Abrahamson and Walter J Silva. Empirical response spectral attenuation relations for shallow crustal earthquakes. *Seismological research letters*, 68(1):94–127, 1997.

[AS97b] Gail M Atkinson and Walt Silva. An empirical study of earthquake source spectra for california earthquakes. *Bulletin of the Seismological Society of America*, 87(1):97–113, 1997.

[AS00a] Khalid Alshibli and Stein Sture. Shear band formation in plane strain experiments of sand. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 126(6):495–503, 2000.

[AS00b] Gail M. Atkinson and Walter Silva. Stochastic modeling of california ground motions. *Bulletin of the Seismological Society of America*, 90(2):255–274, 2000.

[AS06] Giulio Alfano and Elio Sacco. Combining interface damage and friction in a cohesive-zone model. *International Journal for Numerical Methods in Engineering*, 68(5):542–582, 2006.

[AS08] Haydar Arslan and Stein Sture. Finite element simulation of localization in granular materials by micropolar continuum approach. *Computers and Geotechnics*, 35(4):548–562, July 2008.

[AS10] Linda Al Atik and Nicholas Sitar. Seismic earth pressures on cantilever retaining structures. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 136(10):1324–1333, October 2010.

[AS14] Mohammad A. AL-Shudeifat. Highly efficient nonlinear energy sink. *Nonlinear Dynamics*, 76:1905–1920, January 2014.

[AS16a] Kioumars Afshari and Jonathan P Stewart. Validation of duration parameters from SCEC broadband platform simulated ground motions. *Seismological Research Letters*, 87(6):1355–1362, 2016.

[AS16b] FS Alıcı and Halûk Sucuoğlu. Prediction of input energy spectrum: attenuation models and velocity spectrum scaling. *Earthquake Engineering & Structural Dynamics*, 45(13):2137–2161, 2016.

[AS18] F Soner Alıcı and HalÛk Sucuoğlu. Elastic and inelastic near-fault input energy spectra. *Earthquake Spectra*, 34(2):611–637, 2018.

[AS20] Philip Andrews-Speed. South Korea's nuclear power industry: recovering from scandal. *Journal of World Energy Law and Business*, 13:47–57. 2020.

[Asa83] Robert J. Asaro. Micromechanics of crystals and polycrystals. In John W. Hutchinson and Wu Theodore Y, editors, *Advances in Applied Mechanics*, volume 23, pages 1–115. Academic Press, 1983.

[ASA11] Timothy D Ancheta, Jonathan P Stewart, and Norman A Abrahamson. Engineering characterization of earthquake ground motion coherency and amplitude variability. In *4th IASPEI/IAEE International Symposium*, 2011.

[ASA12] TD Ancheta, JP Stewart, and NA Abrahamson. Frequency dependent windowing: A non-stationary method for simulating spatially variable earthquake ground motions. *Earthquake Spectra*, 2012. in review.

[ASA20] Athanasios Agalianos, Max Sieber, and Ioannis Anastasopoulos. Cost-effective analysis technique for the design of bridges against strike-slip faulting. *Earthquake Engineering & Structural Dynamics*, eqe.3282:1–21, 2020.

[ASC16] ASCE-4. *ASCE 4 Standard for Seismic Analysis of Safety-Related Nuclear Structures*. ASCE, American Society of Civil Engineers, 2016.

[ASG96] Jan-Olov Aidanpää, Hayley H. Shen, and Ram. B. Gupta. Experimental and numerical studies of shear layers in granular shear cell. *ASCE Journal of Engineering Mechanics*, 122(3):187–196, 1996.

[ASJ04] Ronald D. Andrus, Kenneth H. Stokoe, and C. Hsein Juang. Guide for shear-wave-based liquefaction potential evalualtion. *Earthquake Spectra*, 20(2):285–308, May 2004.

[ASJ15] José Antonio Abell, Sumeet Kumar Sinha, and Boris Jeremić. Wavelet based synthetic earthquake sources for path and soil structure interaction modeling: Stress testing of nuclear power plants. In Y. Fukushima and L. Dalguer, editors, *Best Practices in Physics-based Fault Rupture Models for Seismic Hazard Assessment of Nuclear Installations*. IAEA, 2015.

[ASK14] Norman A Abrahamson, Walter J Silva, and Ronnie Kamai. Summary of the ASK14 ground motion relation for active crustal regions. *Earthquake Spectra*, 30(3):1025–1055, 2014.

[ASM09a] V&V Committee ASME. Standard for verification and validation in computational fluid dynamics and heat transfer. *American Society of Mechanical Engineers, New York*, 2009.

[ASM09b] ASME-VV-20. Standard for verification and validation in computational fluid dynamics and heat transfer V&V 20 - 2009(r2016). Technical report, American Society of Mechanical Engineers, 2009. ISBN: 9780791832097.

[ASM18] ASME-VV-40. Assessing credibility of computational modeling through verification and validation: Application to medical devices V&V 40 - 2018. Technical report, American Society of Mechanical Engineers, 2018. ISBN: 9780791872048.

[ASM19] ASME-VV-10. Standard for verification and validation in computational solid mechanics, V&V 10 - 2019. Technical report, American Society of Mechanical Engineers, 2019. ISBN: 9780791873168.

[ASRAK^{+}24] Safwan Al-Subaihawi, James Ricles, Qasim Abu-Kassab, Muhannad Suleiman, Richard Sause, and Thomas Marullo. Coupled aero-hydro-geotech real-time hybrid simulation of offshore wind turbine monopile structures. *Engineering Structures*, 303:117463, 2024.

[Ass71] John E. Blume & Associates. Holiday inn (29). 359-393 1971, vol. 1, part a, National Oceanographic and Aviation Administration, Washington, DC, 1971.

[ASS83] Javier Avilés and Francisco J. Sánchez-Sesma. Piles as barriers for elastic waves. *Journal of Geotechnical Engineering*, 109(9):1133–1146, 1983.

[ASS91a] N. A. Abrahamson, J. F. Schneider, and J. C. Stepp. Empirical spatial coherency functions for applications to soil–structure interaction analysis. *Earthquake Spectra*, 7(1):1–27, 1991.

[ASS91b] N. A. Abrahamson, J. F. Schneider, and J. C. Stepp. Spatial coherency of shear waves from the Lotung, Taiwan large-scale seismic test. *Structural Safety*, 10:145–162, 1991.

[ASS97] GEB Archer, Andrea Saltelli, and IM Sobol. Sensitivity measures, anova-like techniques and the use of bootstrap. *Journal of Statistical Computation and Simulation*, 58(2):99–120, 1997.

[ASS19] Kioumars Afshari, Jonathan P. Stewart, and Jamison H. Steidl. California ground motion vertical array database. *Earthquake Spectra*, 35(4):2003–2015, November 2019.

[ASWQ^{+}13] Mohammad A. AL-Shudeifat, Nicholas Wierschem, D. Dane Quinn, Alexander F. Vakakis, Lawrence A. Bergman, and Billie F. Spencer. Numerical and experimental investigation of a highly effective single-sided vibro-impact non-linear energy sink for shock mitigation. *International Journal of Non-Linear Mechanics*, 52:96–109, 2013.

[ATA^{+}09] John Anderson, Ileana Tibuleac, Abdolrasool Anooshehpoor, G. Biasi, Kenneth Smith, and David Seggern. Exceptional ground motions recorded during the 26 april 2008 Mw 5.0 earthquake in Mogul, Nevada. *Bulletin of The Seismological Society of America - BULL SEISMOL SOC AMER*, 99:3475–3486, 2009.

[ATG03] J. B. Anderson, R. C. Townsend, and B. Grajales. Case history evaluation of laterally loaded piles. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 129(3):187–196, March 2003.

[Atk93] John Atkinson. *An Introduction to the Mechanics of Soils and Foundations*. Series in Civil Engineering. McGraww–Hill, 1993. ISBN 0-07-707713-X.

[Atk06] Gail M. Atkinson. Single-station sigma. *Bulletin of the Seismological Society of America*, 96(2):446–455, April 2006.

[A.TXX] L. A.Taber. Application of shell theory to cardiac mechanics. *XX*, XX.

[AW93] Kevin Amaratunga and John R. Williams. Wavelet based green's function approach to 2D PDEs. *Engineering Computations*, 10:349–367, 1993.

[AW99] George B Arfken and Hans J Weber. *Mathematical methods for physicists*. AAPT, 1999.

[AWC^{+}92] William A. Arnold, William R. Wilcox, Frederick Carlson, Arnon Chait, and Liya L. Regel. Transport modes during crystal growth in a centrifuge. *Journal of Crystal Growth*, 119:24–40, 1992.

[AWC98] Schott A. Ashford, Thomas J. Weaver, and Ronaldo L Carreon. Seismic response of the pantabangan dam complex in the 1990 phillipine earthquake. In Panos Dakoulas, Mishac Yegian, and Robert D. Holtz, editors, *Proceedings of a Specialty Conference: Geotechnical Earthwuake Engineering and Soil Dynamics III*, Geotechnical Special Publication No. 75, pages 962–973. ASCE, August 1998. 1998.

[Aya15] Utkarsh Ayachit. *The ParaView Guide: A Parallel Visualization Application*. Kitware, Inc., USA, 2015.

[AZ67] A. D. Aleksandrov and V. A. Zaglaller. *Intrinsinic Geometry of Surfaces*. American Mathematical Society, 1967. Translation from Russian.

[AZ05] Swagato Acharjee and Nicholas Zabaras. Uncertainty propagation in finite deformations – a spectral stochastic lagrangian approach. *Computer Methods in Applied Mechanics and Engineering*, 2005. Preprint.

[AZ06] Swagato Acharjee and Nicholas Zabaras. Uncertainty propagation in finite deformations – a spectral stochastic Lagrangian approach. *Computer Methods in Applied Mechanics and Engineering*, 195(19-22):2289–2312, April 2006.

[AZY^{+}23] Mohamed Abdelmeguid, Chunhui Zhao, Esref Yalcinkaya, George Gazetas, Ahmed Elbanna, and Ares Rosakis. Revealing the dynamics of the feb 6th 2023 M7.8 Kahramanmaraş/Pazarcik earthquake: near-field records and dynamic rupture modeling. EarthArXiv: https://doi.org/10.31223/X5066R, April 2023.

[BA82] Michel Bouchon and Keiiti Aki. Strain, tilt, and rotation associated with strong ground motion in the vicinity of earthquake faults. *Bulletin of the Seismological Society of America*, 72(5):1717–1738, 10 1982.

[BA95] Ronaldo I. Borja and Enruiqe Alarcón. A mathematical framework for finite strain elastoplastic consolidation part 1: Balance laws, variational formulation, and linearization. *Computer Methods in Applied Mechanics and Engineering*, 122:145–171, 1995.

[BA97] Igor A Beresnev and Gail M Atkinson. Modeling finite-fault radiation from the ω n spectrum. *Bulletin of the Seismological Society of America*, 87(1):67–84, 1997.

[BA08] D. M. Boore and G. M. Atkinson. Ground-motion prediction equations for the average horizontal component of pga, pgv, and 5 %-damped psa at spectral periods between 0.01 s and 10.0 s. *Earthquake Spectra*, March 2008.

[BA18a] Jeff Bayless and Norman A Abrahamson. Evaluation of the interperiod correlation of ground-motion simulations. *Bulletin of the Seismological Society of America*, 108(6):3413–3430, 2018.

[BA18b] Jeff Bayless and Norman A Abrahamson. Implications of the inter-period correlation of strong ground motions on structural risk. In *Proceedings of the 11th U.S. National Conference on Earthquake Engineering*, 2018.

[BA19a] Jeff Bayless and Norman A Abrahamson. An empirical model for the interfrequency correlation of epsilon for fourier amplitude spectra. *Bulletin of the Seismological Society of America*, 109(3):1058–1070, 2019.

[BA19b] Jeff Bayless and Norman A Abrahamson. Summary of the ba18 ground-motion model for fourier amplitude spectra for crustal earthquakes in california. *Bulletin of the Seismological Society of America*, 109(5):2088–2105, 2019.

[BA19c] Milad Bybordiani and Yalin Arici. Structure-soil-structure interaction of adjacent buildings subjected to seismic loading. *Earthquake Engineering & Structural Dynamics*, 0(0), 2019.

[Bab95a] Marijan Babic. A connection between continuum and discrete models of particulate materials. In Stein Sture, editor, *Proceedings of 10th Conference*, pages 790–793. Engineering Mechanics Division of the American Society of Civil Engineers, May 1995.

[BAB^{+}95b] Robert J. Budnitz, George Apostolakis, David M. Boore, Lloyd S. Cluff, Kevin J. Coppersmith, C. Allin Cornell, and Peter A. Morris. Recommendation for probabilistic seismic hazard analysis: Guidance on uncertainty and use of experts, volume 1, main report. Prepared by the Senior Seismic Hazard Analysis Committee (SSHAC). NUREG-CR-6372, Lawrence Livermore National Laboratory, August 1995.

[BAB^{+}95c] Robert J. Budnitz, George Apostolakis, David M. Boore, Lloyd S. Cluff, Kevin J. Coppersmith, C. Allin Cornell, and Peter A. Morris. Recommendation for probabilistic seismic hazard analysis: Guidance on uncertainty and use of experts, volume 2, appendices. Prepared by the Senior Seismic Hazard Analysis Committee (SSHAC). NUREG-CR-6372, Lawrence Livermore National Laboratory, August 1995.

[BAB^{+}98] Robert J. Budnitz, George Apostolakis, David M. Boore, Lloyd S. Cluff, Kevin J. Coppersmith, C. Allin Cornell, and Peter A. Morris. Use of technical expert panels: Applications to probabilistic seismic hazard analysis. *Risk Analysis*, 18(4):463–469, 1998.

[BAB^{+}05] Gerald Baumgartner, Alexander Auer, David E Bernholdt, Alina Bibireata, Venkatesh Choppella, Daniel Cociorva, Xiaoyang Gao, Robert J Harrison, So Hirata, Sriram Krishnamoorthy, et al. Synthesis of high-performance parallel programs for a class of ab initio quantum chemistry models. *Proceedings of the IEEE*, 93(2):276–292, 2005.

[BAC86] H. J. Braudel, M. Abouaf, and J. L. Chenot. An implicit and incremental formulation for the solution of elastoplastic problems , by the finite element method. *Computers Structures*, 22(5):801–814, 1986.

[Bac97] George Backus. *Continuum Mechanics*. Samizdat Press `http://samizdat.mines.edu`, 1997.

[BAC05] Jack W Baker and C Allin Cornell. A vector-valued ground motion intensity measure consisting of spectral acceleration and epsilon. *Earthquake Engineering & Structural Dynamics*, 34(10):1193–1217, 2005.

[BAC06] Jack W Baker and C Allin Cornell. Spectral shape, epsilon and record selection. *Earthquake Engineering & Structural Dynamics*, 35(9):1077–1095, 2006.

[BAD13] Timothy A. Burkhart, David M. Andrews, and Cynthia E. Dunning. Finite element modeling mesh quality, energy balance and validation methods: A review with recommendations associated with the modeling of bone tissue. *Journal of Biomechanics*, 46(9):1477 – 1488, 2013.

[Bae23] Gregory B. Baecher. 2021 terzaghi lecture: Geotechnical systems, uncertainty, and risk. *Journal of Geotechnical and Geoenvironmental Engineering*, 149(3):03023001, 2023.

[Bag17] Marco Gaetano Baglio. *Stochastic ground motion method combining a Fourier amplitude spectrum model from a response spectrum with application of phase derivatives distribution prediction*. PhD thesis, Politecnico di Torino, 2017.

[BAGN93] R Betti, AM Abdel-Ghaffar, and AS Niazy. Kinematic soil–structure interaction for long-span cable-supported bridges. *Earthquake engineering & structural dynamics*, 22(5):415–430, 1993.

[BAJF98] Igor A Beresnev, Gail M Atkinson, Paul A Johnson, and Edward H Field. Stochastic finite-fault modeling of ground motions from the 1994 northridge, california, earthquake. ii. widespread nonlinear response at soil sites. *Bulletin of the Seismological Society of America*, 88(6):1402–1410, 1998.

[Bak07] Jack W Baker. Probabilistic structural response assessment using vector-valued intensity measures. *Earthquake Engineering & Structural Dynamics*, 36(13):1861–1883, 2007.

[Bak13] Jack W. Baker. An introduction to probabilistic seismic hazard analysis (psha). Technical Report Version 2.0.1, Stanford University, 2013.

[Ban93] Prasnata Kumar Banerjee. *The Boundary Element Methods in Engineering*. McGraw Hill Book Company, 1993.

[Bar89] N. S. Bardell. The application of symbolic computing to the hierarchical finite element method. *International Journal for Numerical Methods in Engineering*, 28:1181–1204, 1989.

[Bar90] J. P. Bardet. Hypoplastic model for sands. *ASCE Journal of Engineering Mechanics*, 116(9):1973–1994, September 1990.

[Bar97] A. A. Barhorst. Symbolic equation processing utilizing vector / dyan notation. *Journal of Sound and Vibration*, 208(5):823–839, 1997.

[Bar07] Michele Barbato. *FINITE ELEMENT RESPONSE SENSITIVITY, PROBABILISTIC RESPONSE AND RELIABILITY ANALYSES OF STRUCTURAL SYSTEMS WITH APPLICATIONS TO EARTHQUAKE ENGINEERING*. PhD thesis, Universiy of California, San Diego, 2007.

[Bas09] Ushnish Basu. Explicit finite element perfectly matched layer for transient three-dimensional elastic waves. *International Journal for Numerical Methods in Engineering*, 77(2):151–176, 2009.

[Bat82] Klaus-Jürgen Bathe. *Finite Element Procedures in Engineering Analysis*. Prentice Hall Inc., 1982.

[Bat91] K. J. Bathe. Some remarks and references on recent developments in finite element analysis procedures. *Computers & Structures*, 40(2):201–202, 1991.

[Bat96] Klaus-Jürgen Bathe. *Finite Element Procedures in Engineering Analysis*. Prentice Hall Inc., 1996. ISBN 0-13-301458-4.

[Bat07] Klaus-Jürgen Bathe. Conserving energy and momentum in nonlinear dynamics: a simple implicit time integration scheme. *Computers & Structures*, 85(7):437–445, 2007.

[Baz88] Zdenek Bazant, editor. *Mathematical Modeling of Creep and Shrinkage of Concrete Simposium*. Northwestern University, John Wiley and Sons, august 26–29 1988.

[Baž93] Zdeněk P. Bažant. Scaling laws in mechanics of failure. *ASCE Journal of Engineering Mechanics*, 119(9):1828–1844, 1993.

[Baž96] Zdeněk Bažant. Finite strain generalization of small–strain constitutive relations for any finite strain tensor and additive volumetric–deviatoric split. *International Journal of Solids and Structures*, 33(20-22):2887–2897, 1996.

[BB63] Alan W Bishop and GE Blight. Some aspects of effective stress in saturated and partly saturated soils. *Geotechnique*, 13(3):177–197, 1963.

[BB84] David M Boore and John Boatwright. Average body-wave radiation coefficients. *Bulletin of the Seismological Society of America*, 74(5):1615–1621, 1984.

[BB04a] Y. Bozorgnia and V.V. Bertero, editors. *Earthquake Engineering: From Engineering Seismology to Performance–Based Engineering*. CRC Press, 2004.

[BB04b] B.H.G. Brady and E.T. Brown. *Rock mechanics for underground mining*. Kluwer Academic Publishers, 2004, Dordrecht ; Boston, 3rd edition, 2004.

[BB23a] Patrick C. Bassal and Ross W. Boulanger. System response of an interlayered deposit with a localized graben deformation in the Northridge earthquake. *Soil Dynamics and Earthquake Engineering*, 165:107668, 2023.

[BB23b] Gitanjali Bhattacharjee and Jack W. Baker. Using global variance-based sensitivity analysis to prioritise bridge retrofits in a regional road network subject to seismic hazard. *Structure and Infrastructure Engineering*, 19(2):164–177, 2023.

[BB23c] Henry V. Burton and Jack W. Baker. Evaluating the effectiveness of ground motion intensity measures through the lens of causal inference. *Earthquake Engineering & Structural Dynamics*, n/a(n/a):1–23, 2023.

[BB23d] Henry V. Burton and Jack W. Baker. Evaluating the effectiveness of ground motion intensity measures through the lens of causal inference. *Earthquake Engineering & Structural Dynamics*, 52(15):4842–4864, 2023.

[BBBF06] J. Baroth, L. Bodé, Ph. Bressolette, and M. Fogli. Sfe method using hermite polynomials: An approach for solving nonlinear mechanical problems with uncertain parameters. *Computer Methods in Applied Mechanics and Engineering*, 195(44-47):6479–6501, September 2006.

[BBC^{+}] Georgios Baltzopoulos, Roberto Baraschino, Eugenio Chioccarelli, Pasquale Cito, Antonio Vitale, and Iunio Iervolino. Near-source ground motion in the M7.8 Gaziantep (Turkey) earthquake. *Earthquake Engineering & Structural Dynamics*, n/a(n/a).

[BBC^{+}17] Jack Baker, Jonathan Bray, C.B. Crouse, Gregory Deierlein, Ronald O. Hamburger, John Hooper, Marshall Lew, Joe Maffei, Stephen Mahin, James O. Malley, Jack P. Moehle, Farzad Naeim, Jonathan P. Stewart, and John Wallace. Guidelines for performance-based seismic design of tall buildings. Technical Report PEER Report No. 2017/05, Pacific Earthquake Engineering Research Center, Richmod, CA, May 2017.

[BBCF07] J. Barotha, Ph. Bressolettea, C. Chauvièreb, and M. Fogli. An efficient sfe method using lagrange polynomials: Application to nonlinear mechanical problems with uncertain parameters. *Computer Methods in Applied Mechanics and Engineering*, 196(45-48):4419–4429, September 2007.

[BBD^{+}20] D.E.-M. Bouhjiti, J. Baroth, F. Dufour, S. Michel-Ponnelle, and B. Masson. Stochastic finite elements analysis of large concrete structures2̆019 serviceability under thermo-hydro-mechanical loads 2̆013 case of nuclear containment buildings. *Nuclear Engineering and Design*, 370:110800, 2020.

[BBE^{+}04] Satish Balay, Kris Buschelman, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, 2004.

[BBG^{+}96] Hensheng Bao, Jacobo Bielak, Omar Ghattas, Loukas F. Kallivokas, David R. O'Hallaron, Jonathan Richard Shewchuk, and Jifeng Xu. Earthquake ground motion modeling on parallel computers. In *Supercomputing '96*, 1996.

[BBG^{+}98] Hesheng Bao, Jacobo Bielak, Omar Ghattas, Loukas F. Kallivokas, David R. O'Hallaron, Jonathan R. Shewchuk, and Jifeng Xu. Large-scale simulation of elastic wave propagation in heterogeneous media on parallel computers. *Computer Methods in Applied Mechanics and Engineering*, 152(1-2):85–102, January 1998.

[BBG^{+}01] Satish Balay, Kris Buschelman, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. PETSc Web page, 2001. `http://www.mcs.anl.gov/petsc`.

[BBG^{+}12] Daniel Baffet, Jacobo Bielak, Dan Givoli, Thomas Hagstrom, and Daniel Rabinovich. Long-time stable high-order absorbing boundary conditions for elastodynamics. *Computer Methods in Applied Mechanics and Engineering*, 241-244(0):20 – 37, 2012.

[BBI19] Georgios Baltzopoulos, Roberto Baraschino, and Iunio Iervolino. On the number of records for structural risk estimation in PBEE. *Earthquake Engineering & Structural Dynamics*, 48(5):489–506, 2019.

[BBKC07] Scott J. Brandenberg, Ross W. Boulanger, Bruce L. Kutter, and Dongdong Chang. Liquefaction-induced softening of load transfer between pile groups and laterally spreading crusts. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 133(1):91–103, January 2007.

[BBRS24] Nikola Blagojević, Lukas Bodenmann, Yves Reuland, and Božidar Stojadinović. Regional recovery modeling and postdisaster model updating: The case of the 2010 Kraljevo earthquake in Serbia. *Journal of Structural Engineering*, 150(2):05023006, 2024.

[BC64] R H Brooks and A T Corey. Hydraulic properties of porous media and their relation to drainage design. *Trans. ASAE*, 7(1):26–0028, 1964.

[BC80] Klaus Jürgen Bathe and Arthur P Cimento. Some practical procedures for the solution of nonlinear finite element equations. *Computer Methods in Applied Mechanics and Engineering*, 22(1):59–85, 1980.

[BC84] Jacobo Bielak and Paul Christiano. On the effective seismic input for non-linear soil-structure interaction systems. *Earthquake Engineering & Structural Dynamics*, 12(1):107–119, 1984.

[BC91] J. P. Bardet and W. Choucair. A linearized integration technique for incremental constitutive equations. *International Journal for Numerical and Analytical Methods in Geomechanics*, 15(1):1–19, 1991.

[BC02] P Bazzurro and CA Cornell. Vector-valued probabilistic seismic hazard analysis (VPSHA). In *Proceedings of the 7th US national conference on earthquake engineering*, pages 21–25, 2002.

[BC03a] Gregory B. Baecher and John T. Christian. *Reliability and Statistics in Geotechnical Engineering*. John Wiley & Sons Ltd., The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, 2003. ISBN 0-471-49833-5.

[BC03b] Ushnish Basu and Anil K. Chopra. Perfectly matched layers for time-harmonic elastodynamics of unbounded domains: theory and finite-element implementation. *Computer Methods in Applied Mechanics and Engineering*, 192(11-12):1337–1375, March 2003.

[BC04a] U. Basu and A. K. Chopra. Perfectly matched layers for transient elastodynamics of unbounded domains. *International Journal for Numerical Methods in Engineering*, 59(8):1039–1074, February 2004. Erratum: Ibid. 2004; **61**(1):156–157.

[BC04b] Paolo Bazzurro and C. Allin Cornell. Nonlinear soil-site effects in probabilistic seismic-hazard analysis. *Bulletin of the Seismological Society of America*, 94(6):2110–2123, Dec 2004.

[BC04c] Yousef Bozorgnia and Kenneth W Campbell. The vertical-to-horizontal response spectral ratio and tentative procedures for developing simplified v/h and vertical design spectra. *Journal of Earthquake Engineering*, 8(02):175–207, 2004.

[BC04d] Jules Thomas Browaeys and Sébastien Chevrot. Decomposition of the elastic tensor and geophysical applications. *Geophysical Journal International*, 159(2):667–668, 2004.

[BC05] B. Biondi and S. Caddemi. Closed form solutions of euler-bernoulli beams with singularities. *International Journal of Solids and Structures*, 42(9-10):3027–3044, 2005.

[BCC+15] Julian J. Bommer, Kevin J. Coppersmith, Ryan T. Coppersmith, Kathryn L. Hanson, Azangi Mangongolo, Johann Neveling, Ellen M. Rathje, Adrian Rodriguez-Marek, Frank Scherbaum, Refilwe Shelembe, Peter J. Stafford, and Fleur O. Strasser. A SSHAC level 3 probabilistic seismic hazard analysis for a new-build nuclear site in South Africa. *Earthquake Spectra*, 31(2):661–698, 2015.

[BCCA23] Marco Baglio, Alessandro Cardoni, Gian Paolo Cimellaro, and Norman Abrahamson. Generating ground motions using the fourier amplitude spectrum. *Earthquake Engineering & Structural Dynamics*, 52(15):4884–4899, 2023.

[BCD18] A. Banerjee, E.P. Calius, and R. Das. Impact based wideband nonlinear resonating metamaterial chain. *International Journal of Non-Linear Mechanics*, 103:138–144, 2018.

[BCG+05] Peter Brown, Aaron Collier, Keith Grant, Alan Hindmarsh, Steven Lee, Dan Reynolds, Radu Serban, Dan Shumaker, and Carol Woodward. SUNDIALS (SUite of Nonlinear and DIfferential/ALgebraic equation Solvers). `http://www.llnl.gov/CASC/sundials/`, March 2005. Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, California, USA.

[BCJ+88] Alain Bamberger, Bruno Chalindar, Patrick Joly, Jean Elizabeth Roberts, and Jean Luc Teron. Absorbing boundary conditions for rayleigh waves. *SIAM Journal on Scientific and Statistical Computing*, 9(6):1016–1049, Novermber 1988.

[BCK+99] Ross W Boulanger, Christina J Curras, Bruce L Kutter, Daniel W Wilson, and Abbas Abghari. Seismic soil-pile-structure interaction experiments and analyses. *Journal of Geotechnical and Geoenvironmental Engineering*, 125(9):750–759, 1999.

[BCL09] Luc Buatois, Guillaume Caumon, and Bruno Lévy. Concurrent number cruncher - a GPU implementation of a general sparse linear solver. *International Journal of Parallel, Emergent and Distributed Systems*, 2009. to appear, available at `http://alice.loria.fr/index.php/publications.html?Paper=CNC@2008`.

[BCLABG10] A Benavent-Climent, Francisco López-Almansa, and Diego Andrés Bravo-González. Design energy input spectra for moderate-to-high seismicity regions based on colombian earthquakes. *Soil dynamics and earthquake engineering*, 30(11):1129–1148. 2010.

[BCPLA02] A Benavent-Climent, LIG Pujades, and F Lopez-Almansa. Design energy input spectra for moderate-seismicity regions. *Earthquake engineering & structural dynamics*, 31(5):1151–1172, 2002.

[BCS^{+}17] Sanjay Singh Bora, Fabrice Cotton, Frank Scherbaum, Benjamin Edwards, and Paola Traversa. Stochastic source, path and site attenuation parameters and associated variabilities for shallow crustal European earthquakes. *Bulletin of Earthquake Engineering*, 15(11):4531–4561, 2017.

[BCS18] Sanjay Singh Bora, Fabrice Cotton, and Frank Scherbaum. NGA-West2 empirical fourier and duration models to generate adjustable response spectra. *Earthquake Spectra*, page 2, 2018.

[BCSC98] Paolo Bazzurro, C Allin Cornell, Nilesh Shome, and Jorge E Carballo. Three proposals for characterizing MDOF nonlinear seismic response. *Journal of Structural Engineering*, 124(11):1281–1289, 1998.

[BCSS93] R. W. Boulanger, C. K. Chan, H. B. Seed, and R. B. Seed. A low-compliance bi-directional cyclic simple shear apparatus. *Geotech. Testing J.,*, 16(1):36–45, 1993.

[BCW10] D. F. Boutt, B. K. Cook, and J. R. Williams. A coupled fluid-solid model for problems in geomechanics: Application to sand production. *International Journal for Numerical and Analytical Methods in Geomechanics*, Early View, 2010.

[BD79] Jean–Louis Batoz and Gouri Dhatt. Incremental displacement algorithms for nonlinear problems. *International Journal for Numerical Methods in Engineering*, 14:1262–1267, 1979. Short Communications.

[BD83] Klaus-Jürgen Bathe and Eduardo Dvorkin. On the automatic solution of nonlinear finite element equations. *Computers & Structures*, 17(5-6):871–879, 1983.

[BD96] B. Birgisson and A. Drescher. A model for flow liquefaction in saturated loose sand. In *Nordiskt Geoteknikermöte*, 1996.

[BDAA14] David M Boore, Carola Di Alessandro, and Norman A Abrahamson. A generalization of the double-corner-frequency source spectral model and its use in the SCEC bbp validation exercise. *Bulletin of the Seismological Society of America*, 104(5):2387–2398, 2014.

[BdBG^{+}93] Nenad Bićanić, René de Borst, Walter Gerstle, Dave W. Murray, Gilles Pijaudier-Cabot, Victor Saouma, Kaspar J. Willam, and Jun Yamazaki. Computational aspect of finite element analysis of reinforced concrete structures. Structural Engineering and Structural Mechanics Research Series Report CU/SR-93/3, Department of CEAE, University of Colorado at Boulder, February 1993.

[BDD04] Matt Bournonville, Jason Dahnke, and David Darwin. Statistical analysis of the mechanical properties and weight of reinforcing bars. Technical report, The University of Kansas, Structural Engineering and Materials Laboratory, Kansas, 2004.

[BDdP11] Giuseppe Buscarnera, Giuseppe Dattola, and Claudio di Prisco. Controllability, uniqueness and existence of the incremental response: A mathematical criterion for elastoplastic constitutive laws. *International Journal of Solids and Structures*, 48(13):1867 – 1878, 2011.

[BDH^{+}] Erik Boman, Karen Devine, Robert Heaphy, Bruce Hendrickson, William F. Mitchell, Matthew St. John, and Courtenay Vaughan. *Zoltan: Data-Management Services for Parallel Applications*. Sandia National Laboratories and National Institute of Standards and Technology. `http://www.cs.sandia.gov/Zoltan/`

[BDP06] Zdeněk Bažant and Sze Dai Pang. Mechanics-based statistics of failure risk of quasibrittle structures and size effect on safety factors. *Proceedings of the National Academy of Sciences*, 103(25):9434–9439, 2006.

[BDSS20] Íhsan Engin Bal, Dimitris Dais, Eleni Smyrou, and Vasilis Sarhosis. Monitoring of a historical masonry structure in case of induced seismicity. *International Journal of Architectural Heritage*, 0(0):1–18, 2020.

[BDVS15] Edson Borin, Philippe R.B. Devloo, Gilvan S. Vieira, and Nathan Shauer. Accelerating engineering software on modern multi-core processors. *Advances in Engineering Software*, 84:77–84, June 2015.

[BE89] I. Babuška and H. C. Elman. Some aspects of parallel implementation of the finite–element method on message passing architecture. *Journal of Comutational and Applied Mathematics*, 27:157–187, 1989.

[Bea06] Robert Bea. Reliability and human factors in geotechnical engineering. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 132(5):631–643, May 2006.

[BEF$^+$56] B. S. Bloom, M. D. Engelhart, E. J. Furst, W. H. Hill, and D. R. Krathwohl. *Taxonomy of educational objectives; the classification of educational goals*, volume Handbook I: Cognitive Domain. New York, D. McKay Co., Inc., 1956.

[Beh17] Fathemah Behbehani. Theoretical and numerical modeling of unsaturated soil using coupled elements. Master's thesis, University of California Davis, 2017.

[BEM92] I. Babuška, H. C. Elman, and K. Markley. Parallel implementation of the *hp*-version of the finite element method on a shared–memory. *SIAM Journal of Scientific and Statistical Computing*, 13(6):1433–1459, November 1992.

[Ber82] Pål Bergan. Automated incremental – iterative solution methods in structural mechanics. In E. Hinton, D.R.J. Owen, and C. Taylor, editors, *Recent Advances in Non–Linear Computational Mechanics*, chapter 2, pages 41–62. Pineridge Press, Swansea U.K., 1982.

[Ber91] Lars Bernspång. *Iterative and Adaptive Solution Techniques in Computational Plasticity*. PhD thesis, Department of Structural Mechanics, Chalmers University of Technology, Göteborg, Sweden, May 1991. Publication 91:6.

[Ber95] Pål G. Bergan. Challenges in computational mechanics applied to offshore engineering. In Stein Sture, editor, *Proceedings of 10th Conference*, pages 29–44. Engineering Mechanics Division of the American Society of Civil Engineers, May 1995.

[Bes58] J. F. Besseling. A theory of elastic, plastic, and creep deformations of an initially isotropic material showing anisotropic strain–hardening, creep recovery, and secondary creep. *ASME Journal of Applied Mechanics*, pages 529–536, December 1958.

[Bes85] J. F. Besseling. Models of metal plasticity: Theory and experiment. In A. Sawczuk and G. Bianchi, editors, *Plasticity Today: Modelling, Methods and Applications*, pages 97–113. Elsevier Applied Sciences Publishers, 1985.

[Bes89] Dimitri E. Beskos. Dynamics of saturated rocks. i: Equations of motion. *ASCE Journal of Engineering Mechanics*, 115(5):982–995, May 1989.

[BF85] P. G. Bergan and C. A. Felippa. A triangular membrane element with rotational degrees of freedom. *Computer Methods in Applied Mechanics and Engineering*, 50:25–69, 1985.

[BF89] Ted Belytschko and Jacob Fish. Embedded hinge lines for plate elements. *Computer Methods in Applied Mechanics and Engineering*, 76:67–86, 1989.

[BF02] P. Bernard and G. Fleury. Stochastic Newmark scheme. *Probabilistic Engineering Mechanics*, 17(1):45–61, January 2002.

[BF08] Jack W. Baker and Michael H. Faber. Liquefaction risk assessment using geostatistics to account for soil spatial variability. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 1:14–23, 2008.

[BFE88] Ted Belytschko, Jacob Fish, and Bruce E. Engelmann. A finite element with embedded localization zones. *Computer Methods in Applied Mechanics and Engineering*, 70:59–89, 1988.

[BFT07] I. Babuška, Nobile F, and R. Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM J. Numer. Anal.*, 45(3):1005–1034, 2007.

[BG94] L. Badea and P. Gilormini. Application of a domain decomposition method to elastoplastic problems. *International Journal of Solids and Structures*, 31(5):643–656, 1994.

[BG98] F.E. Benth and J. Gjerde. Convergence rates for finite element approximations for stochastic partial differential equations. *Stoch. Stoch. Rep.*, 63:313–326, 1998.

[BG99] M. Bouteca and Y. Gueguen. Mechanical properties of rocks: Pore pressure and scale effects. *Oil & Gas Science and Technology*, 54(6):703–714, 1999.

[BG13a] Holly M. Bik and Miriam C. Goldstein. An introduction to social media for scientists. *PLoS Biol*, 11(4):e1001535, 04 2013.

[BG13b] Holly M. Bik and Miriam C. Goldstein. An introduction to social media for scientists. *PLOS Biology*, 2013.

[BG17] Zhenning Ba and Xi Gao. Soil-structure interaction in transversely isotropic layered media subjected to incident plane sh waves. *Shock and Vibration*, 2017, 2017.

[BGD03] Laurie G. Baise, Steven D. Glazer, and Douglas Dreger. Site response at treasure and yerba buena islands, california. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 129(6):415–426, June 2003.

[BGM03] M Boulon, Vito Nicola Ghionna, and G Mortara. A strain-hardening elastoplastic model for sand-structure interface under monotonic and cyclic loading. *Mathematical and computer modelling*, 37(5-6):623–630, 2003.

[BGMS97] Satish Balay, William D. Gropp, Lois Curfman McInnes, and Barry F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997.

[BGO$^+$10] Jacobo Bielak, Robert W. Graves, Kim B. Olsen, Ricardo Taborda, Leonardo Ramírez-Guzmán, Steven M. Day, Geoffrey P. Ely, Daniel Roten, Thomas H. Jordan, Philip J. Maechling, John Urbanic, Yifeng Cui, and Gideon Juve. The shakeout earthquake scenario: Verification of three simulation sets. *Geophysical Journal International*, 180(1):375–404, 2010.

[BGP89] I. Babuška, M. Griebel, and J. Pitkåranta. The problem of selecting the shape functions for a p–type finite element. *International Journal for Numerical Methods in Engineering*, 28:1891–1908, 1989.

[BGP03] Paul E. Barbone, Dan Givoli, and Igor Patlashenko. Optimal modal reduction of vibrating substructures. *International Journal for Numerical Methods in Engineering*, 57:341–369, 2003.

[BGS57] B. A. Bilby, L. R. T. Gardner, and A. N. Stroh. Continuous distributions of dislocations and the theory of plasticity. In IX^e *Congrès International de Mèchanique Appliquèe*, volume VIII, pages 35–44, Universitè de Bruxelles, 50. Avènue Franklin Roosevelt, 1957.

[BGT85] A Bayliss, C.I Goldstein, and E Turkel. On accuracy conditions for the numerical computation of waves. *Journal of Computational Physics*, 59(3):396 – 404, 1985.

[BH51] JFW Bishop and Rodney Hill. A theory of the plastic distortion of a polycrystalline aggregate under combined stresses. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 42(327):414–427, 1951.

[BH76] P. G. Bergan and L. Hanssen. A new approach for deriving "good" element stiffness matrices. In J. R. Whiteman, editor, *MAFELAP II conference, The Mathematics of Finite Elements and Applications*, volume 2, pages 483–497. Academmic Press, London, 1976.

[BH82] A. Brooks and T.J. Hughes. Streamline upwind/petrov - galerkin formulations for convection-dominated flows with particular emphasis on the incompressible navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32:199 – 259, 1982.

[BH08] Kevin Boudreau and Andrei Hagiu. Platform rules: Multi-sided platforms as regulators. *Working Paper Series, Social Science Research Network*, pages 1–29, 2008. (`http://papers.ssrn.com/sol3/cf_dev/AbsByAuth.cfm?per_id=110372`).

[BH10] Kaiming Bi and Hong Hao. Influence of irregular topography and random soil properties on coherency loss of spatial seismic ground motions. *EARTHQUAKE ENGINEERING AND STRUCTURAL DYNAMICS*, Early View DOI: 10.1002/eqe.1077, 2010.

[Bha93] Sudip S. Bhattacharjee. Finite element modelling of the tensile strain softening behaviour of plane concrete structures. *Engineering Computations*, 10:205–221, 1993.

[BHA17] Annemarie S Baltay, Thomas C Hanks, and Norm A Abrahamson. Uncertainty, variability, and earthquake physics in ground-motion prediction equations. *Bulletin of the Seismological Society of America*, 107(4):1754–1772, 2017.

[BHB$^+$00] Jacobo Bielak, Yoshiaki Hisada, Hesheng Bao, Jifeng Xu, and Omar Ghattas. One- vs two- or three-dimensional effects in sedimentary valleys. In *Proceedings of the 12th WCEE*, Auckland, New Zealand, 2000.

[Bić89] Pankaj Nenad Bićanić. Exact evaluation of contact stress state in computational elasto plasticity. *Engineering Computations*, 6:67–73, 1989.

[Bie78] Jacobo Bielak. Dynamic response of non-linear building-foundation systems. *Earthquake Engineering & Structural Dynamics*, 6(1):17–30, 1978.

[BIK+20] V.A. Bratov, A.V. Ilyanshenko, S.V. Kuznetsov, T.-K. Lin, and N.F. Morozov. Homogeneous horizontal and vertical seismic barriers: Mathematical foundations and dimensional analysis. *Materials Physics and Mechanics*, 44:61–66, 2020.

[Bio41] M.A. Biot. General theory of three-dimensional consolidation. *Journal of Applied Physics*, 12:155–164, February 1941.

[Bio62] M.A. Biot. Mechanics of deformation and acoustic propagation in porous media. *Journal of Applied Physics*, 33(4):1482–1498, April 1962.

[Bio72] M. A. Biot. Theory of finite deformations of porous solids. *Indiana University Mathematical Journal*, 21(7):597–620, January 1972.

[BIPJ78] David M Boore, Adolph A. Oliver III, Robert A. Page, and William B. Joyner. Estimation of grond motion parameters. Open File Report 78-509, US-NRC, 1978.

[Bir03] Peter Bird. An updated digital model of plate boundaries. *Geochemistry, Geophysics, Geosystems*, 4(3), 2003.

[Bis59] Alan W Bishop. The principle of effective stress. *Teknisk ukeblad*, 39:859–863, 1959.

[BJ78] N. Bićanić and K. H. Johnson. Who was '–Raphson'? __, pages 148–152, 1978. Short Communications.

[BJ95] Stacy J. Bartoletti and James O. Jirsa. Effects of epoxy coating on anchorage and development of welded wire fabric. *ACI Journal*, 92(6):757–764, November-December 1995.

[BJ97] David M Boore and William B Joyner. Site amplifications for generic rock sites. *Bulletin of the seismological society of America*, 87(2):327–341, 1997.

[BJEG14] S. Brûlé, E. H. Javelaud, S. Enoch, and S. Guenneau. Experiments on seismic metamaterials: Molding surface waves. *Phys. Rev. Lett.*, 112:133901, Mar 2014.

[BJR05] E. Bécache, P. Joly, and J. Rodríguez. Space–time mesh refinement for elastodynamics. numerical results. *Computer Methods in Applied Mechanics and Engineering*, 194(2-5):355–366, February 2005.

[BK62] Henri Bader and Daisuke Kuroiwa. *The physics and mechanics of snow as a material*. Cold regions science and engineering. Part II. Cold Regions Research and Engineering Laboratory (U.S.), Engineer Research and Development Center (U.S.), 1962.

[BKD24] Seung-Hun Baek, Tae-Hyuk Kwon, and Jason T. DeJong. Reductions in hydraulic conductivity of sands caused by microbially induced calcium carbonate precipitation. *Journal of Geotechnical and Geoenvironmental Engineering*, 150(2):04023134, 2024.

[BKF+14] B. L. Boyce, S. L. B. Kramer, H. E. Fang, T. E. Cordova, M. K. Neilsen, K. Dion, A. K. Kaczmarowski, E. Karasz, L. Xue, A. J. Gross, A. Ghahremaninezhad, K. Ravi-Chandar, S.-P. Lin, S.-W. Chi, J. S. Chen, E. Yreux, M. Rüter, D. Qian, Z. Zhou, S. Bhamare, D. T. O'Connor, S. Tang, K. I. Elkhodary, J. Zhao, J. D. Hochhalter, A. R. Cerrone, A. R. Ingraffea, P. A. Wawrzynek, B. J. Carter, J. M. Emery, M. G. Veilleux, P. Yang, Y. Gan, X. Zhang, Z. Chen, E. Madenci, B. Kilic, T. Zhang, E. Fang, P. Liu, J. Lua, K. Nahshon, M. Miraglia, J. Cruce, R. DeFrese, E. T. Moyer, S. Brinckmann, L. Quinkert, K. Pack, M. Luo, and T. Wierzbicki. The Sandia fracture challenge: blind round robin predictions of ductile tearing. *International Journal of Fracture*, 186:5–68, January 2014.

[BKW87] K.-J. Bathe, M. Kojić, and J. Walczak. Some developments in methods for large strain elasto–plastic analysis. In E. Oñate et al., editor, *Computational Plasticity*, pages 263–279. Pineridge Press, 1987.

[BKZ15] Robert J. Budnitz, Dimitrios Konstandinidis, and Zhiguang Zhou. Evaluations of NRC seismic-structural regulations and regulatory guidance, and simulation-evaluation tools for applicability to small modular reactors (SMRs). NUREG/CR 7193, United States Nuclear Regulatory Commission, Division of Engineering, Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, Washington, DC 20555-0001, December 2015.

[BKZ21] P.A. Bońkowski, J. Kuś, and Z. Zembaty. Seismic rocking effects on a mine tower under induced and natural earthquakes. *Archives of Civil and Mechanical Engineering*, 21(65):1–11, April 2021.

[BL73] WF Brumund and GA Leonards. Experimental study of static and dynamic friction between sand and typical constuction materials. *Journal of Testing and Evaluation*, 1(2):162–165, 1973.

[BL90] Ronaldo I. Borja and Seung R. Lee. Cam–clay plasticity, part I: Implicit integration of elasto–plastic constitutive relations. *Computer Methods In Applied Mechanics and Engineering*, 78:49–72, 1990.

[BL93] Zdeněk P. Bažant and Zhengzhi Li. Modulus of rupture: Size effect due to fracture initition in boundary layer. *ASCE Journal of Structural Engineering*, 121(4):739–746, 1993.

[BL94] D.M. Beazley and P.S. Lomdahl. Message-passing multi-cell molecular dynamics on the connection machine 5. *Parallel Computations*, 20:173–195, 1994.

[BL98] A. Barak and O. La'adan. The MOSIX multicomputer operating system for high performance cluster computing. *Journal of Future Generation Computer Systems*, April 1998.

[BL06] Paolo Bazzurro and Nicolas Luco. Do scaled and spectrum-matched near-source records produce biased nonlinear structural responses. In *Proceedings of the 8th National Conference on Earthquake Engineering*, 2006.

[BL09] Rebecca M. Brannon and Seubpong Leelavanichkul. Survey of four damage models for concrete. Technical Report SAN2009-5544, Sandia National Laboratory, 2009.

[Bla08] Edgar F Black. Dynamic allocation of ragged arrays using C++: Potentiality for finite-element method software. *ASCE Journal of Computing in Civil Engineering*, 22(5):303–311, September/October 2008.

[Bla09] G. Blatman. *Adaptive sparse polynomial chaos expansions for uncertainty propagation and sensitivity analysis*. PhD thesis, Université Blaise Pascal, Clermont-Ferrand, France, 2009.

[BLB83] "S.C. Bandis, A.C. Lumsden, and N.R. Barton". Fundamentals of rock joint deformation. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, 20(6):249 – 268, 1983.

[BLD18] Nenad Bijelić, Ting Lin, and Greg G. Deierlein. Validation of the SCEC broadband platform simulations for tall building risk assessments considering spectral shape and duration of the ground motion. *Earthquake Engineering & Structural Dynamics*, DOI: 10.1002/eqe.3066(0), 2018.

[BLD19] Nenad Bijelić, Ting Lin, and Gregory G Deierlein. Evaluation of building collapse risk and drift demands by nonlinear structural analyses using conventional hazard analysis versus direct simulation with cybershake seismograms evaluation of building collapse risk and drift demands. *Bulletin of the Seismological Society of America*, 109(5):1812–1828, 2019.

[BLHY03] Jacobo Bielak, Kostas Loukakis, Yoshiaki Hisada, and Chiaki Yoshimura. Domain reduction method for three–dimensional earthquake modeling in localized regions. part I: Theory. *Bulletin of the Seismological Society of America*, 93(2):817–824, 2003.

[BLLH17] Zhenning Ba, Jianwen Liang, Vincent W Lee, and Liming Hu. IBEM for impedance functions of an embedded strip foundation in a multi-layered transversely isotropic half-space. *Journal of Earthquake Engineering*, pages 1–32, 2017. DOI: 10.1080/13632469.2017.1286621.

[BLO87] Ted Belytschko, W. K. Liu, and J. S.-J. Ong. Mixed variational priciples and stabilization of spurious modes in the 9–node element. *Computer Methods in Applied Mechanics and Engineering*, 62:275–292, 1987.

[Blo00] Frederic J Blom. Considerations on the spring analogy. *International journal for numerical methods in fluids*, 32(6):647–668, 2000.

[Blo05] Joshua Bloch. How to design a good api and why it matters. In David Musser, editor, *Library-Centric Software Design LCSD'05*. Object-Oriented Programming, Systems, Languages and Applications, October 2005. Presentation Slides.

[BLR^{+}23] André Burnol, Antoine Armandine Les Landes, Daniel Raucoules, Michael Foumelis, Cécile Allanic, Fabien Paquet, Julie Maury, Hideo Aochi, Théophile Guillon, Mickael Delatre, Pascal Dominique, Adnand Bitri, Simon Lopez, Philippe P. Pébaÿ, and Behrooz Bazargan-Sabet. Impacts of Water and Stress Transfers from Ground Surface on the Shallow Earthquake of 11 November 2019 at Le Teil (France). *Remote Sens.*, 2023.

[BM81] B.A. Bolt and H.F. Morrison. Modification by trench barriers of the seismic input to nuclear power plants. Technical Report NUREG/CR-1777; UCB-ENG-4723, Department of Engineering Geosciences, UC Berkeley, 1981.

[BM83] D. Bogard and H. Matlock. Procedures for analysis of laterally loaded pile groups in soft clay. In *Geotechnical Practice in Offshore Engineering*, pages 499–535. the American Society of Civil Engineers, April 1983.

[BM93] Gerald Van Belle and Donald C. Martin. Sample size as a function of coefficient of variation and ratio of means. *The American Statistician*, 47(3):165–167, 1993.

[BM01] T. Belytschko and K. Mish. Computability in non–linear mechanics. *International for Numerical Methods in Engineering*, 52:3–21, 2001.

[BM08] Jacques Betbeder-Matibet. *Seismic Engineering*. ISTE Ltd and John Wiley & Sons, Inc., 2008.

[BM16] Robert J. Budnitz and Michael W. Mieler. Toward a more risk-informed and performance-based framework for the regulation of the seismic safety of nuclear power plants. NUREG/CR 7214, United States Nuclear Regulatory Commission, Division of Engineering, Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, Washington, DC 20555-0001, May 2016.

[BM20] Jad I. Boksmati and Gopal S.P. Madabhushi. Centrifuge modelling of structures with oil dampers under seismic loading. *Earthquake Engineering & Structural Dynamics*, 49(4):356–374, 2020.

[BMK91] T Belytschko, B Moran, and M Kulkarni. On the crucial role of imperfections in quasi-static viscoplastic solutions. *Journal of Applied Mechanics*, 58:658–665, 1991.

[BMLT03] Ivo Babuška, Kang Man Liu, and Raùl Tempone. Solving stochastic partial differential equations based on the experimental data. *Mathematical Models & Methods in Applied Sciences*, 13:415–444, 2003.

[BMR88] Dan A. Brown, Clark Morrison, and Lymon C. Reese. Lateral loaded behavior of pile group in sand. *Journal of Geotechnical Engineering*, 114(11):1261–1277, November 1988.

[BMUP01] T. Belytschko, N. Moës, S. Usui, and C. Parimi. Arbirtary discontinuity in finite elements. *International Journal for Numerical Methods in Engineering*, 50:993–1013, 2001.

[BN84] P. G. Bergan and M. K. Nygård. Finite elements with increased freedom in choosing shape functions. *International Journal for Numerical Methods in Engineering*, 20:643–663, 1984.

[BN90] M Boulon and R Nova. Modelling of soil-structure interface behaviour a comparison between elastoplastic and rate type laws. *Computers and Geotechnics*, 9(1-2):21–46, 1990.

[BN97] H. J. Bunge and I. Nielsen. Experimental determination of plastic spin in polycrystalline materials. *International Journal of Plasticity*, 13(5):435–446, 1997.

[BNEH05] D Breysse, H Niandou, S Elachachi, and L Houy. A generic approach to soil–structure interaction considering the effects of soil heterogeneity. *Geotechnique*, 55(2):143–150, 2005.

[BNOT04] I. Babuska, F. Nobile, J.T. Oden, and R. Tempone. Reliability, uncertainty estimates, validation and verification. Technical Report 04-05, ICES, 2004.

[BNT07] Ivo Babuška, Fabio Nobile, and Raul Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 45(3):1005–1034, 2007.

[BNY87] Richard Byrd, Jorge Nocedal, and Ya Xiang Yuan. Global convergence of a class of quasi – newton methods on convex problems. *SIAM Journal On Numerical Analysis*, 24(5):1171 – 1190, 1987.

[BO76] Bernard Budiansky and Richard J. O'Connell. Elastic moduli of a cracked solids. *International Journal of Solids and Structures*, 12:81–97, 1976.

[BO04] Ivo Babuška and J. Tinsley Oden. Verification and validation in computational engineering and science: basic concepts. *Computer Methods in Applied Mechanics and Engineering*, 193(36-38):4057–4066, Sept 2004.

[BOK07] Ryan B. Bond, Curtis C. Ober, and Patrick M. Knupp. Measuring progress in Premo order-verification. *Engineering with Computers.*, 23(4):283–294, 2007.

[BOKar] T. Belytschko, D. Organ, and Y. Krongauz. A coupled finite element–element-free galerkin method. *Computational Mechanics*, 1995 (to appear).

[Bol86] MD Bolton. The strength and dilatancy of sands. *Geotechnique*, 36(1):65–78, 1986.

[Bol89] Bruno A. Boley. A short history of SMiRT, a personal view, August 1989.

[Bom98] Patricia Bomme. *Intelligent Objects in Object–Oriented Engineering Environments*. PhD thesis, École Polytechnique Fédéderal the Lausanne, 1998.

[Bom10] Julian J Bommer. Sigma: What is it, why it matters and what can we do with it. In *Proceedings of NGA-East "Sigma" Workshop*, University of California, Berkeley, February 2010. Pacific Earthquake Engineering Research Center. `http://peer.berkeley.edu/ngaeast/2010/02/sigma-workshop/`.

[Boo83] David M Boore. Stochastic simulation of high-frequency ground motions based on seismological models of the radiated spectra. *Bulletin of the Seismological Society of America*, 73(6A):1865–1894, 1983.

[Boo94] Grady Booch. *Object Oriented Analysis and Design with Applications*. Series in Object–Oriented Software Engineering. Benjamin Cummings, second edition, 1994.

[Boo03a] David M Boore. Phase derivatives and simulation of strong ground motions. *Bulletin of the Seismological Society of America*, 93(3):1132–1143, 2003.

[Boo03b] David M. Boore. Simulation of ground motion using the stochastic method. *Pure and Applied Geophysics*, 160:635–676, 2003.

[Boo05] David M Boore. *SMSIM: Fortran programs for simulating ground motions from earthquakes: Version 2.3*. Citeseer, 2005.

[Boo09] David M Boore. Comparing stochastic point-source and finite-source ground-motion simulations: Smsim and exsim. *Bulletin of the Seismological Society of America*, 99(6):3202–3216, 2009.

[Boo13] David M Boore. The uses and limitations of the square-root-impedance method for computing site amplification. *Bulletin of the Seismological Society of America*, 103(4):2356–2368, 2013.

[Boo16] David M Boore. Determining generic velocity and density models for crustal amplification calculations, with an update of the generic site amplification for. *Bulletin of the Seismological Society of America*, 106(1):313–317, 2016.

[Bor91] Ronaldo I. Borja. Cam – clay plasticity, part II: Implicit integration of constitutive equation based on a nonlinear elastic stress predictor. *Computer Methods in Applied Mechanics and Engineering*, 88:225–240, 1991.

[Bor98] Joseph Bordogna. Tomorrow's civil systems engineers – the master intergrators. *ASCE Journal of Professional Issues in Engineering Education and Practice*, 124(2):48, 1998.

[Bor04] Ronaldo I. Borja. Cam-clay plasticity. Part V: A mathematical framework for three-phase deformation and strain localization analyses of partially saturated porous media. *Computer Methods in Applied Mechanics and Engineering*, 193(48-51):5301–5338, December 2004.

[Bor06a] Ronaldo I. Borja. Conditions for instabilities in collapsible solids including volume implosion and compaction banding. *Acta Geotechnica*, 1:107–122, 2006.

[Bor06b] Ronaldo I. Borja. On the mechanical energy and effective stress in saturated and unsaturated porous continua. *International Journal of Solids and Structures*, 43(6):1764–1786, 2006.

[Boris Jeremić04] Boris Jeremić. A brief overview of neesgrid simulation platform opensees: Application to the soil–foundation–structure interaction problems. In Maria Todorovska and Mehmet Chelebi, editors, *Proceedings of the Third United States–Japan Natural Resources Workshop on Soil-Structure Interaction*, Vallombrosa Center, Menlo Park, California, USA, March 29-30 2004.

[Bou62] R. C. Bourret. Propagation of randomly perturbed fields. *Canadian Journal of Physics*, 40:782–790, 1962.

[Bou89] M Boulon. Basic features of soil structure interface behaviour. *Computers and Geotechnics*, 7(1-2):115–131, 1989.

[Bou06] Kevin Boudreau. Does opening a platform stimulate innovation? effects on modular and systemic innovation. *MIT Sloan Research Paper No. 4611-06, Working Paper Series, Social Science Research Network*, (http://papers.ssrn.com/sol3/papers.cfm?abstract_id=913402):1–33, 2006.

[Bou08] Kevin Boudreau. Opening the platform vs. opening the complementary good? the effect on product innovation in handheld computing. *Working Paper Series, Social Science Research Network*, (http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1251167):1–36, 2008.

[Bount] Ross Boulanger. Personal communications, 1999-present.

[BP88] G. Bianchini and P. Puccini. Response to questionaire. In A. Saada and G. Bianchini, editors, *Constitutive Equations for Granular Non–Cohesive Soils*, pages 63–79. A. A. Balkema, July 1988.

[BP98] Ed Boring and Alex Pang. Interactive deformation from tensor fields. In D. Ebert and H. Hagen adn H. Rushmeier, editors, *Processing of IEEE Visualization 98*, pages 297–304. Computer Society Press, 1998.

[BP06] Peter J. Basser and Siniša Pajević. Spectral decomposition of a 4th-order covariance tensor: Applications to diffusion tensor mri. *Signal Processing*, 87:220–236, 2006.

[BP09] M. N. Bagde and V. Petros. Fatigue and dynamic energy behaviour of rock subjected to cyclical loading. *International Journal of Rock Mechanics and Mining Sciences*, 46(1):200–209, 2009.

[BP11] T. Blanc and M. Pastor. A stabilized smoothed particle hydrodynamics, taylor2̆013galerkin algorithm for soil dynamics problems. *International Journal for Numerical and Analytical Methods in Geomechanics*, pages n/a–n/a, 2011.

[BP21] Vinicius Beber and Diogo Pitz. Machine learning and finite element analysis: An integrated approach for fatigue lifetime prediction of adhesively bonded joints. *Fatigue & Fracture of Engineering Materials & Structures*, 08 2021.

[BPCA17] Sifeng Bi, Saurabh Prabhu, Scott Cogan, and Sez Atamturktur. Uncertainty quantification metrics with varying statistical information in model calibration and validation. *AIAA Journal*, 55(10):3570–3583, 2017.

[BPDH19] Jin Whan Bae, Joshua L. Peterson-Droogh, and Kathryn D. Huff. Standardized verification of the cyclus fuel cycle simulator. *Annals of Nuclear Energy*, 128:288–291, 2019.

[BPE88] E. D. Booth, J. W. Pappin, and J. J. B. Evans. Computer aided analysis methods for the design of earthquake resistant structures: a review. *Proc. Instn Civ. Engrs, Part 1*, 84:671–691, August 1988.

[BPE$^+$14] Lauren L. Beghini, Anderson Pereira, Rodrigo Espinha, Ivan F.M. Menezes, Waldemar Celes, and Glaucio H. Paulino. An object-oriented framework for finite element analysis based on a compact topological data structure. *Advances in Engineering Software*, 68(0):40 – 48, 2014.

[BPJJ23] Siddharth Banerjee, Colin M. Potts, Arnav H. Jhala, and Edward J. Jaselskis. Developing a construction domain-specific artificial intelligence language model for NCDOT's CLEAR program to promote organizational innovation and institutional knowledge. *Journal of Computing in Civil Engineering*, 37(3):04023007, 2023.

[BPW89] Dimitri E. Beskos, Constantine N. Papadakis, and Hyo Seop Woo. Dynamics of saturated rocks. i: Rayleigh waves. *ASCE Journal of Engineering Mechanics*, 115(5):1017–1034, May 1989.

[Bra96] M. F. Bransby. Differences between load–transfer relationships for laterally loaded pil groups: Active $p-y$ or passive $p-\delta$. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 122(12):1015–1018, December 1996.

[Bra97] Dietrich Braess. *Finite Elements*. Cambridge University Press, 1997.

[Bra99] M. F. Bransby. Selection of p-y curves for the design of single laterally loaded piles. *International journal for numerical and analysis methods in geomechanics*, 23:1909–1926, 1999.

[Bra10] Brendon A Bradley. A generalized conditional intensity measure approach and holistic ground-motion selection. *Earthquake Engineering & Structural Dynamics*, 39(12):1321–1342, 2010.

[Bra11] Brendon A. Bradley. A framework for validation of seismic response analyses using seismometer array recordings. *Soil Dynamics and Earthquake Engineering*, 31(3):512 – 520, 2011.

[Bra12] Brendon A. Bradley. The seismic demand hazard and importance of the conditioning intensity measure. *Earthquake Engineering & Structural Dynamics*, 41(11):1417–1437, 2012.

[Bra22] M. Bragg. In Our Time, Seismology. BBC Podcast, March 2022. `https://www.bbc.co.uk/sounds/play/m00154gh`.

[BRD03] D Batista, P Royis, and T Doanh. Time-integration of a hypoplastic constitutive equation in finite element modelling. *Mathematical and computer modelling*, 37(5):615–621, 2003.

[Bre80] Leonid Maksimovich Brekhovskikh. *Waves in Layered Media*. Academic Press, 1980. Translated by Robert T. Beyer; ISBN: 9783540655923.

[Bre90] John E. Breen. Prestressed concrete: The state of the art in north america. *PCI Journal*, 35(6):62–67, November/December 1990.

[Bre13] Carlos A. Brebbia, editor. *Finite Element Systems, A Handbook, Edition 2*. Springer-Verlag Berlin Heidelberg GmbH, 2013.

[Bre21] Bret Lizundia. Private Communications. RUTHERFORD + CHEKENE, San Francisco, California. 2019-2021.

[Bri53] Léon Brillouin. *Wave Propagation in Periodic Structures*. McGraw-Hill Book Company, Inc, (1946) and Dover Publications, Inc. (1953), 1953.

[Bri94] Ronald Bastiaan Johan Brinkreve. *Geomaterial Models and Numerical Analysis of Softening*. PhD thesis, Delft Technical University, May 1994.

[BRL00] Ronaldo I. Borja, Richard A. Regueiro, and Timothy Y. Li. FE modeling of strain localization in soft rock. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 126(4):335–343, 2000.

[BRMB19] Mahdi Bahrampouri, Adrian Rodriguez-Marek, and Julian J. Bommer. Mapping the uncertainty in modulus reduction and damping curves onto the uncertainty of site amplification functions. *Soil Dynamics and Earthquake Engineering*, 126:105091, 2019.

[BRRC$^+$23] Edén Bojórquez, Sonia E. Ruiz, Ali Rodriguez-Castellanos, Miguel A. Orellana, Alfredo Reyes-Salazar, and Juan Bojórquez. Bayesian analysis-based ground motion prediction equations for earthquake input energy. *Soil Dynamics and Earthquake Engineering*, 173:108115, 2023.

[BRS76] J. R. Booker, M. S. Rashman, and H. Bolton Seed. GADFLEA: A computer program for the analysis of pore pressure generation and dissipation during cyclic or earthquake loading. Technical Report EERC 76-24, University of California, Berkeley, October 1976.

[BRSTGR10] E Bojórquez, A Reyes-Salazar, A Terán-Gilmore, and SE Ruiz. Energy-based damage index for steel structures. *Steel and Composite Structures*, 10(4):331–348, 2010.

[BRT93] Samir Bougacha, José Roësset, and John L. Tassoulas. Dynamics stiffness of foundations on fluid–filled poroelastic stratum. *ASCE Journal of Engineering Mechanics*, 119(8):1649–, 1993. Missing last part.

[Bru70] James N Brune. Tectonic stress and the spectra of seismic shear waves from earthquakes. *Journal of geophysical research*, 75(26):4997–5009, 1970.

[BS65] A. Baltov and A. Sawczuk. A rule of anisotropic hardening. *Acta Mechanica*, I(2):81–92, 1965.

[BS73] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.

[BS75a] J.R Booker and J.C. Small. An investigation of the stability of numerical solutions of biot's equations of consolidation. *International Journal for Solids and Structures*, 11:907–917, 1975.

[BS75b] Peter L. Bransby and Ian A. A. Smith. Side friction in model retaining-wall experiment. *ASCE Journal of the Geotechnical Engineering Division*, 101(7):615–632, July 1975.

[BS90a] H.P. Bader and B. Salm. On the mechanics of snow slab release. *Cold Regions Science and Technology*, 17(3):287–300, 1990.

[BS90b] Dan A. Brown and Chine-Feng Shie. Numerical experiments into group effects on the response of piles to lateral loading. *Computers and Geotechnics*, 10:211–230, 1990.

[BS90c] Dan A. Brown and Chine-Feng Shie. Three dimensional finite element model of laterally loaded piles. *Computers and Geotechnics*, 10:59–79, 1990.

[BS91a] I. Babuška and B. Szabó. *Finite element analysis*. John Wiley & Sons Inc., 1991.

[BS91b] Dan A. Brown and Chine-Feng Shie. Some numerical experiments with a three dimensional finite element model of a laterally loaded pile. *Computers and Geotechnics*, 12:149–162, 1991.

[BS94a] Ivo Babuška and Manil Suri. The p and h–p versions of the finite element method, basic principles and properties. *SIAM Review*, 36(4):578–632, 1994.

[BS94b] Mikaël Bourges-Sévenier. Réalisation d'une bibliothéque C de fonctions ondolettes. Publication Interne 864, IRISA Institute de Recherche en Informatique et Systémes Aléatoires, Campus Universitaire de Beaulieu, 35042 Rennes Cedex, France, Septembre 1994.

[BS96a] Louis Baker and Bradley J. Smith. *Parallel Programming*. Series in Computing. McGraw–Hill, 1996. ISBN 0-07-912259-0.

[BS96b] M. F. Bransby and S. M. Springman. 3-d finite element modeling of pile groups adjacent to surcharge loads. *Computers and Geotechnics*, 19(4):301–324, 1996.

[BS98] I. Babuška and P. Shi. Regularity of solutions to a one dimensional plasticity model. *Mathematical Modeling and Numerical Analysis*, 32(5):521–537, 1998.

[BS08] Thomas Benz and Radu Schwabb. A quantitative comparison of six rock failure criteria. *International Journal of Rock Mechanics and Mining Sciences*, 45(7):1176–1186, October 2008.

[BS10] Géraud Blatman and Bruno Sudret. An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis. *Probabilistic Engineering Mechanics*, 25:183–197, 2010.

[BS16] Christopher G Burton and Vitor Silva. Assessing integrated earthquake risk in openquake with an application to mainland portugal. *Earthquake Spectra*, 32(3):1383–1403, 2016.

[BS22] Nikola Blagojević and Božidar Stojadinović. A demand-supply framework for evaluating the effect of resource and service constraints on community disaster resilience. *Resilient Cities and Structures*, 1(1):13–32, 2022.

[BSA12] André R. Brodtkorb, Martin L. Sætra, and Mustafa Altinakar. Efficient shallow water simulations on gpus: Implementation, visualization, verification, and validation. *Computers and Fluids*, 55:1 – 12, 2012.

[BSB12] B. Bassa, F. Sabourin, and M. Brunet. A new nine-node solid-shell finite element using complete 3D constitutive laws. *International Journal for Numerical Methods in Engineering*, pages n/a–n/a, 2012.

[BSH19] Jin Whan Bae, Clifford E. Singer, and Kathryn D. Huff. Synergistic spent nuclear fuel dynamics within the european union. *Progress in Nuclear Energy*, 114:1–12, 2019.

[BSK85] K.J. Bathe, R. Slavković, and M. Kojić. On large strain elasto–plastic and creep analysis. In Wunderlich Bergan, Bathe, editor, *Finite Element Methods for Nonlinear Problems, Europe–US Symposium, Trondheim, Norway*, pages 176–190, 1985.

[BSK$^+$15] Sanjay Singh Bora, Frank Scherbaum, Nicolas Kuehn, Peter Stafford, and Benjamin Edwards. Development of a response spectral ground-motion prediction equation (GMPE) for seismic-hazard analysis from empirical fourier spectral and duration models. *Bulletin of the Seismological Society of America*, 105(4):2192–2218, 2015.

[BSKS16] Sanjay Singh Bora, Frank Scherbaum, Nicolas Kuehn, and Peter Stafford. On the relationship between fourier and response spectra: Implications for the adjustment of empirical ground-motion prediction equations (GMPEs). *Bulletin of the Seismological Society of America*, 106(3):1235–1253, 2016.

[BSL00] M. Bart, J. F. Shao, and D. Lydzba. Poroelastic behaviour of saturated brittle rock with anisotropic damage. *International Journal for Numerical and Analytical Methods in Geomechanics*, 24(15):1139–1154, 2000.

[BSL06] M. Berveiller, B. Sudret, and M. Lemaire. Stochastic finite element: A non-intrusive approach by regression. *European Journal of Computational Mechanics*, 15:81–92, 2006.

[BSM^{+}23] Reza Boushehri, Kevin Stanton, Ramin Motamed, Kirk Ellison, and Ibbi Almufti. Nonlinear time-domain soil-structure interaction analysis of a mid-rise building benchmarked against earthquake recordings: A case study of the BRI's annex building. *Soil Dynamics and Earthquake Engineering*, 172:108031, 2023.

[BSMF21] Gabriella Bolzon, Donatella Sterpi, Guido Mazzà, and Antonella Frigerio, editors. *Numerical Analysis of Dams*, volume 91 of *Lecture Notes in Civil Engineering*. Springer, 2021.

[BSpBP14] S. Barani, D. Spallarossa, p. Bazzurro, and F. Pelli. The multiple facets of probabilistic seismic hazard analysis: a review of probabilistic approaches to the assessment of the different hazards caused by earthquakes. *Bollettino di Geofisica Teorica ed Applicata*, 55(1):17–40, March 2014.

[BSS^{+}16] Ivo Babuška, Zaid Sawlan, Marco Scavino, Barna Szabó, and Raúl Tempone. Bayesian inference and model comparison for metallic fatigue data. *Computer Methods in Applied Mechanics and Engineering*, 304:171 – 196, 2016.

[BSSA14] David M Boore, Jonathan P Stewart, Emel Seyhan, and Gail M Atkinson. NGA-West2 equations for predicting pga, pgv, and 5% damped psa for shallow crustal earthquakes. *Earthquake Spectra*, 30(3):1057–1085, 2014.

[BT96] Ronaldo I. Borja and Claudio Tamagnini. Critical state model at finite strains. In Y. K. Lin and T. C. Su, editors, *Proceedings of 11th Conference*, pages 148–151. Engineering Mechanics Division of the American Society of Civil Engineers, May 1996.

[BT98] Ronaldo I. Borja and Claudio Tamagnini. Cam–clay plasticity part iii: Extension of the infinitesimal model to include finite strains. *International Journal for Computer Methods in Applied Mechanics and Engineering*, 155:73–95, 1998.

[BT15] David M Boore and Eric M Thompson. Revisions to some parameters used in stochastic-method simulations of ground motion. *Bulletin of the Seismological Society of America*, 105(2A):1029–1041, 2015.

[BTA98] Ronaldo I. Borja, Claudio Tamagnini, and Enrique Alarcón. Elastoplastic consolidation at finite strain part 2: Finite element implementation and numerical examples. *International Journal for Computer Methods in Applied Mechanics and Engineering*, 159:103–122, 1998.

[BTP51] Frank P Bowden, D Tabor, and Frederic Palmer. The friction and lubrication of solids. *American Journal of Physics*, 19(7):428–429, 1951.

[BTR93] Samir Bougacha, John L. Tassoulas, and José Roësset. Analysis of foundations on fluid–filled poroelastic stratum. *ASCE Journal of Engineering Mechanics*, 119(8):1632–1648, 1993.

[BTSL^{+}17] Catherine Berge-Thierry, Angkeara Svay, Aurore Laurendeau, Thomas Chartier, Vincent Perron, Cédric Guyonnet-Benaize, Ejona Kishta, Régis Cottereau, Fernando Lopez-Caballero, Fabrice Hollender, et al. Toward an integrated seismic risk assessment for nuclear safety improving current french methodologies through the SINAPS@ research project. *Nuclear Engineering and Design*, 323:185–201, 2017.

[BTVR^{+}20] C. Berge-Thierry, F. Voldoire, F. Ragueneau, F. Lopez-Caballero, and A. Le Maoult. Main achievements of the multidisciplinary SINAPS research project: Towards an integrated approach to perform seismic safety analysis of nuclear facilities. *Pure and Applied Geophysics*, 177(5):2299–2351, 2020.

[BTZ05] Ivo Babuška, Raúl Tempone, and Georios E. Zouraris. Solving elliptic boundary value problems with uncertain coefficients by the finite element method: the stochastic formulation. *Computational Methods in Applied Mechanics and Engineering*, 194(1):1251–1294, April 2005.

[Buc14] E. Buckingham. On physically similar systems; illustrations of the use of dimensional equations. *Physical Review*, 4:345–376, October 1914.

[Bud65] B. Budiansky. On the elastic moduli of some heterogeneous materials. *Journal of Mechanics and Physics of Solids*, 13:223–227, 1965.

[Bul32] Bulletin of the Seismological Society of America. Kyoji suyehiro. *Bulletin of the Seismological Society of America*, 22(2):180, June 1932.

[BV98] I. N. Basuroychowdhury and G. Z. Voyiadjis. A mutliaxial cyclic plasticity model for non–proportional loading cases. *International Journal of Plasticity*, 14(9):855–870, 1998.

[BVDG94] Johannes F Besseling and Erik Van Der Giessen. *Mathematical modeling of inelastic deformation*, volume 5. CRC Press, 1994.

[BVP89] Dimitri E. Beskos, Irene Vgenopoulou, and Constantine P. Providakis. Dynamics of saturated rocks. i: Body waves. *ASCE Journal of Engineering Mechanics*, 115(5):996–1016, May 1989.

[BW76] Klaus-Jürgen Bathe and L. Wilson, Edward. *Numerical Methods in Finite Element Analysis*. Prentice Hall Inc., 1976.

[BW96] Igor A Beresnev and Kuo-Liang Wen. Nonlinear soil response, a reality? *Bulletin of the Seismological Society of America*, 86(6):1964–1978, 1996.

[BW15] Chandrakanth Bolisetti and Andrew S. Whittaker. Site response, soil-structure interaction and structure-soil-structure interaction for performance assessment of buildings and nuclear structures. Technical Report MCEER-15-0002, MCEER, 2015.

[BWBF19] Francesco Basone, Moritz Wenzel, Oreste S. Bursi, and Marinella Fossetti. Finite locally resonant metafoundations for the seismic protection of fuel storage tanks. *Earthquake Engineering & Structural Dynamics*, 48(2):232–252, 2019.

[BWC12] Dhiman Basu, Andrew S. Whittaker, and Michael C. Constantinou. Estimating rotational components of ground motion using data recorded at a single station. *ASCE Journal of Engineering Mechanics*, 138(9):1141–1156, September 2012.

[BWC18] Chandrakanth Bolisetti, Andrew S Whittaker, and Justin L Coleman. Linear and nonlinear soil-structure interaction analysis of buildings and safety-related nuclear structures. *Soil Dynamics and Earthquake Engineering*, 107:218–233, 2018.

[BWGT08] Sven Buijssen, Hilmar Wobker, Dominik Göddeke, and Stefan Turek. FEASTSolid and feastflow: FEM applications exploiting feast‘s HPC technologies. In Wolfgang Nagel, , Dietmar Kröner, and Michael Resch, editors, *High Performance Computing in Science and Engineering ‘08*, Transactions of the High Performance Computing Center Stuttgart (HLRS) 2008, pages 425–440. Springer, Berlin, December 2008.

[BWM+14] Chandrakanth Bolisetti, Andrew S Whittaker, H Benjamin Mason, Ibrahim Almufti, and Michael Willford. Equivalent linear and nonlinear site response analysis for design and risk assessment of safety-related nuclear structures. *Nuclear Engineering and Design*, 275:107–121, 2014.

[BWY95] Igor A Beresnev, Kuo-Liang Wen, and Yeong Tein Yeh. Seismological evidence for nonlinear elastic ground behavior during large earthquakes. *Soil Dynamics and Earthquake Engineering*, 14(2):103–114, 1995.

[BXG99] Jacobo Bielak, Jifeng Xu, and Omar Ghattas. Earthquake ground motion and structural response in alluvial valleys. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 125(5):413–423, 1999.

[BYCMHL99] Ronaldo I. Borja, Heng Yih Chao, Francisco J. Montána, and Chao Hua Lin. Nonlinear ground response at Lotung LSST site. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 125(3):187–197, March 1999.

[BYCP94] Ted Belytschko, Huai Yang Chaing, and Edward Plakacz. High resolution two–dimensional shear band computations: Imperfections and mesh dependence. *Computer Methods in Applied Mechanics and Engineering*, 119:1–15, 1994.

[Byr93] Richard Byrd. Numerical methods in unconstrained optimization and systems of equations. Lecture Notes at CU Boulder, august - decembar 1993.

[BZ02] Zdeněnek P. Bažant and Yong Zhou. Why did the world trade center collapse? – simple analysis. *ASCE Journal of Engineering Mechanics*, 128(1):2–6, 2002.

[BZ13] RW Boulanger and K Ziotopoulou. Formulation of a sand plasticity plane-strain model for earthquake engineering applications. *Soil Dynamics and Earthquake Engineering*, 53:254–267, 2013.

[BZM19] Piotr Adam Bońkowski, Zbigniew Zembaty, and Maciej Yan Minch. Engineering analysis of strong ground rocking and its effect on tall structures. *Soil Dynamics and Earthquake Engineering*, 116:358 – 370, 2019.

[CA15] Jorge GF Crempien and Ralph J Archuleta. UCSB method for simulation of broadband ground motion from kinematic earthquake sources. *Seismological Research Letters*, 86(1):61–67, 2015.

[Cac05] Dan G Cacuci. *Sensitivity and uncertainty analysis, volume II: applications to large-scale systems*, volume 2. CRC press, 2005.

[Cal04] Caltrans. *Caltrans Seismic Design Criteria version 1.3*. California Department of Transportation, February 2004.

[CAL09] Roger S. Crouch, Harm Askes, and Tianbai Li. Analytical cpp in energy-mapped stress space: Application to a modified drucker-prager yield surface. *Computer Methods in Applied Mechanics and Engineering*, 198(5-8):853 – 859, 2009.

[Cal18] Gian Michele Calvi. Revisiting design earthquake spectra. *Earthquake Engineering & Structural Dynamics*, 10.1002/eqe.3101(0):1–17, 2018.

[Cam03] Kenneth W. Campbell. Prediction of strong ground motion using the hybrid empirical method and its use in the development of ground-motion (attenuation) relations in eastern north america. *Bulletin of the Seismological Society of America*, 93(3):1012–1033, June 2003.

[CAM11] Mark A. Carruth, Julian M. Allwood, and Muiris C. Moynihan. The technical potential for reducing metal requirements through lightweight product design. *Resources, Conservation and Recycling*, 57:48–60, 2011.

[Car05] Carsten Carstensen. Ten remarks on nonconvex minimisation for phase transition simulations. *Computer Methods in Applied Mechanics and Engineering*, 194(2-5):169–193, February 2005.

[Car11] Carl J. Costantino and Associates. Computational concerns using the SASSI computer code for SSI analyses. Report/Letter FYI, 2011.

[Cau60] T.K. Caughey. Classical normal modes in damped linear systems. *ASME Journal of Applied Mechanics*, 27:269–271, 1960.

[CB88] Hui Sing Chang and Ted Belytschko. Multi-scattering and boundary effects in soil-structure interaction. *Nuclear Engineering and Design*, 106(1):9 – 17, 1988.

[CB91] M Carter and S Bentley. Swiss standard sn 670 010b, characteristic coefficients of soils. Technical report, Association of Swiss Road and Traffic Engineers, 1991.

[CB08] Kenneth W Campbell and Yousef Bozorgnia. NGA ground motion model for the geometric mean horizontal component of PGA, PGV, PGD and 5% damped linear elastic response spectra for periods ranging from 0.01 to 10 s. *Earthquake Spectra*, 24(1):139–171, 2008.

[CB14] Kenneth W Campbell and Yousef Bozorgnia. NGA-West2 ground motion model for the average horizontal components of pga, pgv, and 5% damped linear acceleration response spectra. *Earthquake Spectra*, 30(3):1087–1115, 2014.

[CB22] Cyrille Couture and Pierre Bésuelle. A true triaxial experimental study on porous Vosges sandstone: from strain localization precursors to failure using full-field measurements. *International Journal of Rock Mechanics and Mining Sciences*, 153:105031, 2022.

[CBR10] Diego Cecilio, Lira, Philippe Remy Bernard Devloo, and Erick Slis Raggio Santos. On an object oriented implementation of plasticity models. In Eduardo Dvorkin, Marcela Goldschmit, and Mario Storti, editors, *Mecánica Comput.*, volume XXIX, pages 4235–4245, Buenos Aires, Argentina, Noviembre 2010.

[CBS79] J.P. Carter, J.R. Booker, and J.C. Small. The analysis of finite elasto-plastic consolidation. *International Journal for Numerical and Analytical Methods in Geomechanics*, 3:107–129, 1979.

[CBW$^+$09a] K. Onder Cetin, H. Tolga Bilge, Jiaer Wu, Annie M. Kammerer, and Raymond B. Seed. Probabilistic model for the assessment of cyclically induced reconsolidation (volumetric) settlements. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 135(3):387–398, March 2009.

[CBW$^+$09b] K. Onder Cetin, H. Tolga Bilge, Jiaer Wu, Annie M. Kammerer, and Raymond B. Seed. Probabilistic models for cyclic straining of saturated clean sands. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 135(3):371–386, March 2009.

[CBW15] Justin L. Coleman, Chandrakanth Bolisetti, and Andrew S. Whittaker. Time-domain soil-structure interaction analysis of nuclear facilities. *Nuclear Engineering and Design*, 2015.

[CC08] Fehmi Cirak and Julian C. Cummings. Generic programming techniques for parallelizing and extending procedural finite element programs. *Engineering with Computers*, 24:1–16, 2008.

[CC19] Jianbing Chen and Jianpeng Chan. Error estimate of point selection in uncertainty quantification of nonlinear structures involving multiple nonuniformly distributed parameters. *International Journal for Numerical Methods in Engineering*, 0(0), 2019.

[CC24] Elife Cakir and Kemal Onder Cetin. Liquefaction triggering and induced ground deformations at a metallurgical facility in Dörtyol-Hatay after the february 6 Kahramanmaraş earthquake sequence. *Soil Dynamics and Earthquake Engineering*, 178:108465, 2024.

[CCB58] D Croney, J D Coleman, and W P M Black. Movement and distribution of water in soil in relation to highway design and performance, in water and its conduction in soils. Technical report, Highway Research Board, 1958.

[CCB88] Marco G. Cremonini, Paul Christiano, and Jacobo Bielak. Implementation of effective seismic input for soil-structure interaction systems. *Earthquake Engineering & Structural Dynamics*, 16(4):615–625, 1988.

[CCC^{+}16] Carl J. Costantino, Michael C Costantino, Isabel Cuesta, Thomas W. Houston, Andrew S. Maham, E. Mertz Greg, James J. Johnson, Brent J. Gutierrez, and Debra Sparkman. Verificationa and validation of SASSI. Technical Report 4300070839-VDE-6.0-0012, CJC-SVV-C-012, US-DOE, 2016.

[CCKP92] R. A. Cook, D. M. Collins, R. E. Klingner, and D. Polyzois. Load–deflection behavior of cast–in–place and retrofit anchors. *ACI Structural Journal*, 89(6):639–649, November-December 1992.

[CCM^{+}21] Mathieu Causse, Cécile Cornou, Emeline Maufroy, Jean-Robert Grasso, Laurent Baillet, and Elias El Haber. Exceptional ground motion during the shallow Mw 4.9 2019 Le Teil earthquake, France. *COMMUNICATIONS EARTH & ENVIRONMENT*, 2(14), 2021.

[CCP77] Ma Chi Chen and J. Penzien. Nonlinear soil–structure interaction of skew highway bridges. Technical Report UCB/EERC-77/24, Earthquake Engineering Research Center, University of California, Berkeley, August 1977.

[CCR^{+}16a] Andrea Colombi, Daniel Colquitt, Philippe Roux, Sebastien Guenneau, and Richard V. Craster. Forests as a natural seismic metamaterial: Rayleigh wave bandgaps induced by local resonances. *Scientific Reports*, 6(19238):1–7, January 2016.

[CCR^{+}16b] Andrea Colombi, Daniel Colquitt, Philippe Roux, Sebastien Guenneau, and Richard V. Craster. A seismic metamaterial: The resonant metawedge. *Scientific Reports*, 6(27717):1–6, June 2016.

[CD96] Han Wook Cho and Yannis F. Dafalias. Distortinal and orientational hardening at large viscoplastic deformations. *International Journal of Plasticity*, 12(7):903–925, 1996.

[CD02] Xavier Chateau and Luc Dormieux. Micromechanics of saturated and unsaturated porous media. *International Journal for Numerical and Analytical Methods in Geomechanics*, 26:831–844, 2002.

[CDAA^{+}] Serena Cattari, Stefania Degli Abbati, Sara Alfano, Andrea Brunelli, Filippo Lorenzoni, and Francesca da Porto. Dynamic calibration and seismic validation of numerical models of URM buildings through permanent monitoring data. *Earthquake Engineering & Structural Dynamics*, n/a(n/a).

[CDGS08] Brian Chiou, Robert Darragh, Nick Gregor, and Walter Silva. NGA project strong-motion database. *Earthquake Spectra*, 24(1):23–44, 2008.

[CDHS84] RJ Clifton, J Duffy, KA Hartley, and TG Shawki. On critical conditions for shear band formation at high strain rates. *Scripta Metallurgica*, 18(5):443–448, 1984.

[CDJC21] Kyriakos Alexandros Chondrogiannis, Vasilis Dertimanis, Boris Jeremić, and Eleni Chatzi. On the vibration attenuation properties of metamaterial design using negative stiffness elements. In Walter Lacarbonara, Balakumar Balachandran, Michael J. Leamy, Jun Ma, J.A. Tenreiro Machado, and Gabor Stepan, editors, *Advances in Nonlinear Dynamics - Proceedings of the Second International Nonlinear Dynamics Conference (NODYCON 2021)*, volume 3, Sapienza, Università di Roma, 16-19 February 2021. Springer Nature.

[CDJC23] Kyriakos Alexandros Chondrogiannis, Vasilis Dertimanis, Boris Jeremić, and Eleni Chatzi. Design of the negative stiffness negsv mechanism for structural vibration attenuation exploiting resonance. *International Journal of Mechanical Sciences*, 260:108640, 2023.

[CDMC20] Kyriakos Alexandros I. Chondrogiannis, Vasilis K. Dertimanis, Sami F. Masri, and Eleni N. Chatzi. Vibration absorption performance of metamaterial lattices consisting of impact dampers. In Manolis Papadrakakis, Michalis Fragiadakis, and Costas Papadimitriou, editors, *EURODYN 2020, XI International Conference on Structural Dynamics.* Athens. Greece. 23-26 November 2020 2020.

[CDS06] Gye-Chun Cho, Jake Dodds, and J. Carlos Santamarina. Particle shape effects on packing density, stiffness, and strength: Natural and crushed sands. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 132(5):591–602, May 2006.

[CDS16] R. Capillon, C. Desceliers, and C. Soize. Uncertainty quantification in computational linear structural dynamics for viscoelastic composite structures. *Computer Methods in Applied Mechanics and Engineering*, 305:154 – 172, 2016.

[CE01] C. J. Collier and A. S. Elnashai. A procedure for combining vertical and horizontal seismic action effects. *Journal of Earthquake Engineering*, 5(4):521–539, 2001.

[CEA$^+$94] Y. P. Chien, A. Ecer, H. U. Akay, F. Carpenter, and R. A. Blech. Dynamic load balancing on a network of workstations for solving computational fluid dynamics problems. *Computer Methods in Applied Mechanics and Engineering*, 119:17–33, 1994.

[CEH61] A. D. Cox, G. Eason, and H. G. Hopkins. Axially symmetric plastic deformations in soils. *Philosophical Transactions of the Royal Society of London*, A254:1–45, 1961.

[Cel96] Christian C. Celigoj. Coupled thermomechanical problems and generalized standard materials. Seminar given at CU Boulder, July 1996.

[Çel23] Mehmet Çelebi. The new self-anchored suspension bridge of the San Francisco Bay Bridge system: A preliminary study of its response and behavior during a small earthquake. *Journal of Structural Engineering*, 149(6):05023003, 2023.

[Cer95] John N. Cernica. *Geotechnical Engineering Soil Mechanics*. John Wiley & Sons, Inc., 1995.

[CF93] Juan W. Chavez and Gregory L. Fenves. Earthquake analysis and response of concrete gravity dams including base sliding. Technical Report 93/07, EERC/UCB, December 1993.

[CFF10] É. Chamberland, A. Fortin, and M. Fortin. Comparison of the performance of some finite element discretizations for large deformation elasticity problems. *Computers & Structures*, 88(11-12):664 – 673, 2010.

[CFK$^+$98] K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith, and S. Tuecke. A resource management architecture for metacomputing systems. In *Proc. IPPS/SPDP '98 Workshop on Job Scheduling Strategies for Parallel Processing*, 1998.

[CG74] Anil K Chopra and Jorge A Gutierrez. Earthquake response analysis of multistorey buildings including foundation interaction. *Earthquake Engineering & Structural Dynamics*, 3(1):65–77, 1974.

[CGB17] G. Carta, G.F. Giaccu, and M. Brun. A phononic band gap model for long bridges. the 2̌018brabau2̌019 bridge case. *Engineering Structures*, 140:66–76, 2017.

[CGdB05] Doo Bo Chung, Miguel A. Gutiérrez, and René de Borst. Object-oriented stochastic finite element analysis of fibre metal laminates. *Computational Methods in Applied Mechanics and Engineering*, 194(1):1427–1446, April 2005.

[CGG08] S Chiriță, C Galeș, and ID Ghiba. On spatial behavior of the harmonic vibrations in kelvin-voigt materials. *Journal of Elasticity*, 93(1):81–92, 2008.

[CH88] W. F. Chen and D. J. Han. *Plasticity for Structural Engineers*. Springer Verlag, 1988.

[CH89] R. Courant and D. Hilbert. *Methods of Mathematical Physics*. Wiley, 1989.

[CH90] Murray J. Clarke and Gregory J. Hancock. A study of incremental–iterative strategies for non–linear analysis. *International Journal for Numerical Methods in Engineering*, 29:1365–1391, 1990.

[CH92] Peter A. Cundall and Roger D. Hart. Numerical modelling of discontinua. *Engineering Computations*, 9:101–113, 1992.

[CH93] Jintai Chung and GM Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method. *Journal of Applied Mechanics*, 60(2):371–375, 1993.

[CH97] I. F. Collins and G. T. Houlsby. Application of thermomechanical principles to the modelling of geotechnical materials. *Procedings of Royal Society London*, 453:1975–2001, 1997.

[CH03] Ian F. Collins and Tamsyn Hilder. A theoretical framework for constructing elastic/plastic constitutive models from triaxial tests. *International Journal for Numerical an Analytical Methods in Geomechanics*, 26:1313–1347, 2003.

[CH05] N. Chouw and H. Hao. Study of ssi and non–uniform ground motion effect on pounding between bridge girders. *Soil Dyanmics and Earthquake Engineering*, 25:717–728, 2005.

[CH06] Samit Ray Chaudhuri and Tara C. Hutchinson. Fragility of bench-mounted equipment considering uncertain parameters. *ASCE Journal of Structural Engineering*, 132(6):884–898, June 2006.

[CH07] Hsoung-Wei Chou and Jong-Shin Huang. Effects of cyclic compression and thermal aging on dynamic properties of neoprene rubber bearings. *Journal of Applied Polymer Science*, 107:1635–1641, 2007.

[Cha85] H. W. Chandler. A plasticity theory without Drucker's postulate, sujitable for granular materials. *Journal of Mechanics and Physics of Solids*, 33(3):215–226, 1985.

[Cha88a] Andrew Hin-Cheong Chan. *A Unified Finite Element Solution to Static and Dynamic Problems in Geomecahnics*. PhD thesis, Department of Civil Engineering, University College of Swansea, February 1988.

[Cha88b] H. W. Chandler. A variational principle for granular materials. *International Journal for Numerical and Analytical Methods in Geomechanics*, 12:371–378, 1988.

[Cha90] H. W. Chandler. Homogeneous and localized deformation in granular materials: A mechanistic model. *International Journal of Engineering Sciences*, 28(8):719–734, 1990.

[Cha95] Ching S. Chang. Microstructural constitutive modelling for granular material. In Stein Sture, editor, *Proceedings of 10th Conference*, pages 1304–1307. Engineering Mechanics Division of the American Society of Civil Engineers, May 1995.

[Cha96] Ching S. Chang. Micromechanical modelling for granular materials. In Y. K. Lin and T. C. Su, editors, *Proceedings of 11th Conference*, pages 551–554. Engineering Mechanics Division of the American Society of Civil Engineers, May 1996.

[Cha99] Martin C Chapman. On the use of elastic input energy for seismic hazard analysis. *Earthquake Spectra*, 15(4):607–635, 1999.

[Cha06] Noël Challamel. On the comparison of timoshenko and shear models in beam dynamics. *ASCE Journal of Engineering Mechanics*, 132(10):1141–1145, October 2006.

[Cha07] J.L. Chaboche. My side of the story for "A mathematical representation of the multiaxial Bauschinger effect" by Armstrong and Frederick. *Materials at High Temperature*, 24(1):1–26, 2007.

[Che88] W. F. Chen. Evaluation of constitutive models in soil mechanics. In A. Saada and G. Bianchini, editors, *Constitutive Equations for Granular Non–Cohesive Soils*, pages 687–693. A. A. Balkema, July 1988.

[Che94] Wai–Fah Chen. *Constitutive Equations for Engineering Materials: Plasticity and Modeling*, volume 2 of *Studies in applied mechanics 37B*. Elsevier, ELSEVIER SCIENCE B.V. Sara Burgerharstraat 25, P.O.Box 211, 1000 AE Amsterdam, The Netherlands, 1994. In collaboration with W. O. McCarron, AMOCO production Company, Tulsa, OK, USA and E, Yamaguchi, University of Tokyo, Japan.

[Che95] Z. Chen. A stochastic approach with moving jump conditions for localization. In Stein Sture, editor, *Proceedings of 10th Conference*, pages 309–312. Engineering Mechanics Division of the American Society of Civil Engineers, May 1995.

[Che12] J.C. Chen. Seismic response of reactor, turbine and facility buildings of Fukushima power plant from Pacific ocean Tohoku earthquake, march 11, 2011. LLNL-PRES-557237, May 2012.

[Chi99] C. C. Chin. *Substructure subtraction method and dynamic analysis of pile foundations*. PhD thesis, University of California, Berkeley, 1999.

[CHK23] Ross B. Corotis, John D. Hooper, and Ronald Klemencic. Design live loads for office gathering spaces. *Journal of Structural Engineering*, 149(12):04023170, 2023.

[CHKK88] R.D. Campbell, B.F. Henley, R.P. Kennedy, and R.P. Kassawara. Walkdown criteria for evaluating seismic margin in nuclear power plants. *Nuclear Engineering and Design*, 107(1-2):83 – 93, 1988.

[CHLW90] C. B. Crouse, Behnam Hushmand, J. Enrique Luco, and H. L. Wong. Foundation impedance functions: Theory versus experiment. *Journal of Geotechnical Engineering*, 116(3):432–449, March 1990.

[Cho86] Y. K. Chow. Analysis of vertically loaded pile groups. *International journal for numerical and analysis methods in geomechanics*, 10:59–72, 1986.

[Cho00] Anil K. Chopra. *Dynamics of Structures, Theory and Application to Earthquake Engineering*. Prentice Hall, second edition, 2000. ISBN 0-13-086973-2.

[Cho17] Anil K. Chopra. In memoriam: Ray W. Clough. *Earthquake Engineering and Structural Dynamics*, 46:3–4, 2017.

[Cho20] Anil K. Chopra. Modal combination rules in response spectrum analysis: Early history. *Earthquake Engineering & Structural Dynamics*, n/a(n/a), 2020.

[CHP20] Jianye Ching, Wen-Han Huang, and Kok-Kwang Phoon. 3D probabilistic site characterization by sparse Bayesian learning. *ASCE Journal of Engineering Mechanics*, 146(12):04020134, 2020.

[Chr04] John T. Christian. Geotechnical engineering reliability: How well do we know what we are doing? *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 130(10):985–1003, October 2004.

[Chu73] S.L. Chu. Analysis and design capabilities of STRUDL program. In STEVEN J. FENVES, NICHOLAS PERRONE, ARTHUR R. ROBINSON, and WILLIAM C. SCHNOBRICH, editors, *Numerical and Computer Methods in Structural Mechanics*, pages 229 – 246. Academic Press, 1973.

[Chu03] Ashok K. Chugh. On the boundary conditions in slope stability analysis. *International Journal for Numerical an Analytical Methods in Geomechanics*, 27:905–926, 2003.

[Chu09] Glenda Chui. Shaking up earthquake theory. *Nature*, 461(7266):870–873, October 2009.

[Chu14] Ashok K. Chugh. Influence of valley geometry on stability of an earth dam. *Canadian Geotechnical Journal*, 51:1207–1217, 2014.

[CI04] Sara Casciati and Ronaldo I.Borja. Dynamic fe analysis of south memnon colossus including 3D soil–foundation–structure interaction. *Computers and Structures*, 82(20-21):1719 1736, August 2004.

[CI24] Pasquale Cito and Iunio Iervolino. Drivers to seismic hazard curve slope. *Earthquake Engineering & Structural Dynamics*, 53(15):4497–4510, 2024.

[CIBN05] Dan G. Cacuci, Mihaela Ionescu-Bujor, and Ionel Michael Navon. *Sensitivity and Uncertainty Analysis, Volume II*. CRC Press., May 2005.

[Cic11] Pietro Cicotti. *Tarragon: a programming model for latency-hiding scientific computations*. PhD thesis, UC San Diego, 2011.

[Cim21] Robert Cimrman. Fast evaluation of finite element weak forms using python tensor contraction packages. *Advances in Engineering Software*, 159:103033, 2021.

[CJ09a] Zhao Cheng and Boris Jeremić. Constitutive algorithm for hyperelasto-placiticity in the intermediate configuration. *Communications in Numerical Methods in Engineering*, 2009. in review.

[CJ09b] Zhao Cheng and Boris Jeremić. Large deformation, fully coupled modeling and simulation of geomaterials. *International Journal for Numerical Methods in Engineering*, 2009. in review.

[CJ09c] Zhao Cheng and Boris Jeremić. Numerical modeling and simulation of soil lateral spreading against piles. In *In proceedings of the GeoOrlando, Geo Institute Annual Conference*, Orlando, Florida, March 2009.

[CJ09d] Zhao Cheng and Boris Jeremić. Numerical modeling and simulations of piles in liquefiable soil. *Soil Dynamics and Earthquake Engineering*, 29:1405–1416, 2009.

[CJHF02] C. Allin Cornell, Fatemeh Jalayer, Ronald O. Hamburger, and Douglas A. Foutch. Probabilistic basis for 2000 sac federal emergency management agency steel moment frame guidelines. *ASCE Journal of Structural Engineering*, 128(4):526–533, April 2002.

[CJS87] N. C. Costes, V. C. Janoo, and S. Sture. Microgravity experiements on granular materials. In R. H. Doremus and P. C. Nordine, editors, *Material Research Society Symposium Proceedings*, volume 87, pages 203–212, 1987.

[CJW95a] A. H. C. Chan, L. Jendele, and D. Muir Wood. Numerical modelling of moisture, heat and pollutant transport in partially saturated porous media using object oriented programming. Technical report, Department of Civil Engineering, University of Glasgow, Scotland G128QQ, 1995.

[CJW95b] A. H. C. Chan, L. Jendele, and D. Muir Wood. Numerical modelling of moisture, heat and pollutant transport in partially saturated porous media using object oriented programming. Technical report, Department of Civil Engineering, University of Glasgow, Scotland G128QQ, 1995.

[CK00] C. Allin Cornell and Helmut Krawinkler. Progress and challenges in seismic performance assessment. *PEER newsletter*, 2000. https://apps.peer.berkeley.edu/news/2000spring/performance.html Accessed 1 August 2018.

[CK02] IF Collins and PA Kelly. A thermomechanical analysis of a family of soil models. *Geotechnique*, 52(7):507–518, 2002.

[CK05a] Ching S. Chang and Matthew R. Kuhn. On virtual work and stress in granular media. *International Journal of Solids and Structures*, 42(13):3773–3793, 2005.

[CK05b] S. H. Cho and K. Kaneko. Influence of the applied pressure waveform on the dynamic fracture processes in rock. *International Journal of Japanese Comittee for Rock Mechanics*, 1(1):25–28, 2005.

[CK09a] Mesut Cayar and M. Levent Kavvas. Symmetry in nonlinear hydrologic dynamics under uncertainty: A modeling approach. *ASCE Journal of Hydrologic Engineering*, 14(10):001–010, October 2009.

[CK09b] Mesut Cayar and M. Levent Kavvas. Symmetry in nonlinear hydrologic dynamics under uncertainty: Ensemble modeling of 2D Boussinesq equation for unsteady flow in heterogeneous aquifers. *ASCE Journal of Hydrologic Engineering*, 14(10):001–012, October 2009.

[CKAG19] M. Chatzimanolakis, K.-D. Kantarakias, V.G. Asouti, and K.C. Giannakoglou. A painless intrusive polynomial chaos method with rans-based applications. *Computer Methods in Applied Mechanics and Engineering*, 348:207 – 221, 2019.

[CKG94] A Cardona, I. Klapka, and M. Geradin. Design of a new finite element programming environment. *Engineering Computations*, 11:365–381, 1994.

[CKL17] Ali Charara, David Elliot Keyes, and Hatem Ltaief. Batched triangular dense linear algebra kernels for very small matrix sizes on gpus. *ACM Transactions on Mathematical Software*, 2017.

[CKS02] K. Onder Cetin, Armen Der Kiureghian, and Raymond B. Seed. Probabilistic models for the initiation of seismic soil liquefaction. *Structural Safety*, 24(1):67–82, January 2002.

[CL10] Wei Che and Qifeng Luo. Time-frequency response spectrum of rotational ground motion and its application. *Earthquake Science*, 23(1):71–77, 2010.

[CLC17] Yuanzhi Chen, Tam Larkin, and Nawawi Chouw. Experimental assessment of contact forces on a rigid base following footing uplift. *Earthquake Engineering & Structural Dynamics*, 46(11):1835–1854, 2017. EQE-15-0471.R3.

[CLCP16] Jianye Ching, Guan-Hong Lin, Jie-Ru Chen, and Kok-Kwang Phoon. Transformation models for effective friction angle and relative density calibrated based on generic database of coarse-grained soils. *Canadian Geotechnical Journal*, 54(4):481–501, 2016.

[CLD10] Cris Cecka, Adrian J. Lew, and E. Darve. Assembly of finite element methods on graphics processors. *International Journal for Numerical Methods in Engineering*, 85(5):537–561, 2010.

[CLM20] Yin Cheng, Andrea Lucchini, and Fabrizio Mollaioli. Ground-motion prediction equations for constant-strength and constant-ductility input energy spectra. *Bulletin of Earthquake Engineering*, 18(1):37–55, 2020.

[CLT18] Yiwei Cai, Vincent W Lee, and Mihailo D Trifunac. In-plane soil-structure interaction excited by incident plane SV waves. *Soil Dynamics and Earthquake Engineering*, 105:224–230, 2018.

[CM44] R.H. Cameron and W.T. Martin. Transformations of wiener integrals under translations. *The Annals of Mathematics*, 45(2):386–396, 1944.

[CM69] E. Cuthill and J. McKee. Reducing the bandwidth of sparse symmetric matrices. In *ACM Annual Conference/Annual Meeting; Proceedings of the 1969 24th national conference*, pages 157–172, 1969.

[CM94] GA Chang and John B Mander. *Seismic energy based fatigue damage analysis of bridge columns: Part I-Evaluation of seismic capacity*. National Center for Earthquake Engineering Research Buffalo, NY, 1994.

[CM98] Jason A. Crawford and Clark M. Mobarry. HRUNTING: A distributed shared memory system for the Beowulf parallel workstation. In *Proceedings, IEEE Aerospace*, 1998. `http://www.beowulf.org/papers/index.html`

[CM01] C.B. Crouse and Jeff McGuire. Energy dissipation in soil-structure interaction. *Earthquake Spectra*, 17(2):235–259, May 2001.

[CM03] IF Collins and B Muhunthan. On the relationship between stress–dilatancy, anisotropy, and plastic dissipation for granular materials. *Geotechnique*, 53(7):611–618, 2003.

[CM16] Anil K. Chopra and Frank McKenna. Modeling viscous damping in nonlinear response history analysis of buildings for earthquake excitation. *Earthquake Engineering & Structural Dynamics*, 45(2):193–211, 2016.

[CMDÁ21] Yin Cheng, Fabrizio Mollaioli, and Jesús Donaire-Ávila. Characterization of dissipated energy demand. *Soil Dynamics and Earthquake Engineering*, 147:1–14, 2021.

[CMJL95] M. A. Crisfield, G. F. Moita, G. Jelenić, and L. P. R. Lyons. Enhanced lower–order element formulation for large strains. *Computational Mechanics*, 17:62–73, 1995.

[CMO$^+$09] Yifeng Cui, Reagan Moore, Kim Olsen, Amit Chourasia, Philip Maechling, Bernard Minster, Steven Day, Yuanfang Hu, Jing Zhu, and Thomas Jordan. Toward petascale earthquake simulations. *Acta Geotechnica*, 4(2):79–93, 2009.

[CMPW01] Robert D. Cook, David S. Malkus, Michael E. Plesha, and Robert J. Witt. *Concepts and Applications of Finite Element Analysis*. Wiley, 4th edition, 2001.

[CMR93] Edoardo Cosenza, Gaetano Manfredi, and Roberto Ramasco. The use of damage functionals in earthquake engineering: a comparison between different methods. *Earthquake engineering & structural dynamics*, 22(10):855–868, 1993.

[CMR08] Silvia Castellaro, Francesco Mulargia, and Piermaria Luigi Rossi. VS30: Proxy for seismic amplification? *Seismological Research Letters*, 79(4):540–543, 2008.

[CMS95] Peter C. Chang, Richard H. McCuen, and Jayanta K. Sircar. Multimedia–based instruction in engineering education: Strategy. *ASCE Journal of Professional Issues in Engineering Education and Practice*, 121(4):216–219, October 1995.

[CNBC13] Charisis T. Chatzigogos, Pierre-Alain Nazé, Pauline Billion, and Matthieu Caudron. Uplift evaluation of structures under seismic loading: assessment of different calculation methods and design guidelines for the nuclear industry. In *1st Conference on Technical Innovation in Nuclear Civil Engineering – TINCE 2013*, Paris, France, 28-31 October 2013.

[CO67] E.U. Condon and Hugh Odishaw, editors. *Handbook of Physics*. McGraw hill Bokk Company, 1967. QC21C7.

[CO92] A. Cuitiño and M. Ortiz. A material – independent method for extending stress update algorithms from small–strain plasticity to finite plasticity with multiplicative kinematics. *Engineering Computations*, 9:437–451, 1992.

[Cod99a] Donald P. Coduto. *Foundation Design: Principle and Practice*. Prentice Hall, 1999.

[Cod99b] Donald P. Coduto. *Geotechnical Engineering: Principle and Practice*. Prentice Hall, 1999.

[ÇOK$^+$12] Mehmet Çelebi, Izuru Okawa, Toshidate Kashima, Shin Koyama, and Masanori Iiba. Response of a tall building far from the epicenter of the 11 march 2011 M 9.0 Great East Japan earthquake and aftershocks. *THE STRUCTURAL DESIGN OF TALL AND SPECIAL BUILDINGS*, 2012.

[Col02] IF Collins. Associated and non-associated aspects of the constitutive laws for coupled elastic/plastic materials. *International Journal of Geomechanics*, 2(2):259–267, 2002.

[Col03] IF Collins. A systematic procedure for constructing critical state models in three dimensions. *International Journal of Solids and Structures*, 40(17):4379–4397, 2003.

[Com18] Committee on Nuclear Quality Assurance. ASME NQA-1-2017 quality assurance requirements for nuclear facility applications. ASME, Two Park Avenue. New York, NY, 10016 USA, January 2018.

[Con23] Gemma Conroy. How generative AI could disrupt scientific publishing. *Nature*, 622:234–236, 12 October 2023.

[Cop92] James O. Coplien. *Advanced C++, Programming Styles and Idioms*. Addison – Wesley Publishing Company, 1992.

[Cor68] C. Allin Cornell. Engineering seismic risk analysis. *Bulletin of the seismological society of America*, 58(5):1583–1606, 1968.

[Cor86] Ivan Cormeau. Bruce Irons: A non-conforming engineering scientist to be remembered and rediscovered. *International Journal for Numerical Methods in Engineering*, 22:1–10, 1986.

[Cor94] A. Corigliano. Numerical analysis of discretized elastoplastic systems using the generalized mid-point time integration. *Engineering Computations*, 11:389–411, 1994.

[Cos09] Eugéne Cosserat and François Cosserat. *Théorie des Corps Déformables*. Éditions Jacques Gabay (2008), 151 bis rue Saint-Jacques, 75005 Paris, France, 1909. (originally published in 1909, by Librairie Scientifique A. Herman et Fils, 6, rue de la Sorbonne, 6, Paris).

[Cos69] Carl J. Costantino. Two dimensional wave through nonlinear media. *Journal of Computational Physics*, 4:147–170, 1969.

[Cou95] Olivier Coussy. *Mechanics of Porous Continua*. John Wiley and Sons, 1995. ISBN 471 95267 2.

[Cou04] O Coussy. *Poromechanics*. John Wiley & Sons, 2004.

[Cow66] G. R. Cowper. The Shear Coefficient in Timoshenko's Beam Theory. *Journal of Applied Mechanics*, 33(2):335–340, 06 1966.

[CP03] Giuseppe Cocchetti and Umberto Perego. A rigorous bound on error in backward-difference elastoplastic time-integration. *Computer Methods in Applied Mechanics and Engineering*, 192:4909–4927, 2003.

[CP05] Alberto Carpinteri and Marco Paggi. Size-scale effects on the friction coefficient. *International Journal of Solids and Structures*, 42(9-10):2901–2910, 2005.

[CP09] X. Chen and K.K. Phoon. Some numerical experiences on convergence criteria for iterative finite element solvers. *Computers and Geotechnics*, 36(8):1272–1284, October 2009.

[CPACGAM18] A. Castanheira-Pinto, P. Alves-Costa, L. Godinho, and P. Amado-Mendes. On the application of continuous buried periodic inclusions on the filtering of traffic vibrations: A numerical study. *Soil Dynamics and Earthquake Engineering*, 113:391 – 405, 2018.

[CPB$^+$09] C. R. I. Clayton, J. A. Priest, M. Bui, A. Zervos, and S. G. Kim. The Stokoe resonant column apparatus: effects of stiffness, mass and specimen fixity. *Géotechnique*, 59(5):429–438, June 2009.

[CPD01] C. Cremer, A. Pecker, and L. Davenne. Cyclic macro-element of soil structure interaction: Material and geometrical nonlinearities. *International Journal for Numerical and Analytical Methods in Geomechanics*, 25(13):1257–1284, 2001.

[CPS09] C.T. Chatzigogos, A. Pecker, and J. Salençon. Macroelement modeling of shallow foundations. *Soil Dynamics and Earthquake Engineering*, 29(5):765–781, May 2009.

[CPW16] Jianye Ching, Kok Kwang Phoon, and Shih Hsuan Wu. Impact of statistical uncertainty on geotechnical reliability estimation. *Journal of Engineering Mechanics*, 142(6):04016027, 2016.

[CPZ$^+$18] Wenjie Cui, David M. Potts, Lidija Zdravković, Klementyna A. Gawecka, David M.G. Taborda, and Aikaterini Tsiampousi. A coupled thermo-hydro-mechanical finite element formulation for curved beams in two-dimensions. *Computers and Geotechnics*, 103:103 – 114, 2018.

[CR83a] J. L. Chaboche and G. Rousselier. On the plastic and viscoplastic constitutive equations – Part I: Rules developed with internal variable concept. *Journal of Pressure Vessel Technology*, 105:153–158, May 1983.

[CR83b] J. L. Chaboche and G. Rousselier. On the plastic and viscoplastic constitutive equations – Part II: Applications of internal variable concepts to the 316 stainless steel. *Journal of Pressure Vessel Technology*, 105:159–164, May 1983.

[CR19] T. S. Charlton and M. Rouainia. Uncertainty quantification of offshore anchoring systems in spatially variable soil using sparse polynomial chaos expansions. *International Journal for Numerical Methods in Engineering*, 0(0):20, 2019.

[Cri81] M. A. Crisfield. A fast incremental / iterative solution procedure that handles snap - through. *Computers & Structures*, 13:55–62, 1981.

[Cri83] M. A. Crisfield. An arc – length method including line searches and accelerations. *International Journal for Numerical Methods in Engineering*, 19:1296–1289, 1983.

[Cri84] M. A. Crisfield. Accelerating and dumping the modified Newton–Raphson method. *Computers & Structures*, 18(3):395–407, 1984.

[Cri86] M. A. Crisfield. Snap-through and snap-back response in concrete structures and the dangers of underintegration. *International Journal for Numerical Methods in Engineering*, 22:751–767, 1986.

[Cri87] M. A. Crisfield. Consistent schemes for plasticity computation with the Newton Raphson method. *Computational Plasticity Models, Software, and Applications*, 1:136–160, 1987.

[Cri90] Michael A Crisfield. A consistent co-rotational formulation for non-linear, three-dimensional, beam-elements. *Computer methods in applied mechanics and engineering*, 81(2):131–150, 1990.

[Cri91] M. A. Crisfield. *Non–Linear Finite Element Analysis of Solids and Structures Volume 1: Essentials*. John Wiley and Sons, Inc. New York, 605 Third Avenue, New York, NY 10158–0012, USA, 1991.

[Cri97] M. A. Crisfield. *Non–Linear Finite Element Analysis of Solids and Structures Volume 2: Advanced Topics*. John Wiley and Sons, Inc. New York, 605 Third Avenue, New York, NY 10158–0012, USA, 1997.

[Cro11] C.B. Crouse. Effects of surface geology on seismic motion. In *4th IASPEI/IAEE International Symposium*, UCSB, 23-26 August 2011.

[CRW95] Ignacio Carol, Egidio Rizzi, and Kaspar Willam. Current issues in elastic degradation and damage. In Stein Sture, editor, *Proceedings of 10th Conference*, pages 521–524. Engineering Mechanics Division of the American Society of Civil Engineers, May 1995.

[CRW01a] Ignacio Carol, Egidio Rizzi, and Kaspar William. On the formulation of anisotropic elastic degradadation. I. theory based on a pseudo–logarithmic damage tensor rate. *International Journal of Solids and Structures*, 38:491–518, 2001.

[CRW01b] Ignacio Carol, Egidio Rizzi, and Kaspar William. On the formulation of anisotropic elastic degradadation. II. generalized pseudo–Rankine model for tensile damage. *International Journal of Solids and Structures*, 38:519–546, 2001.

[CS79] Peter A Cundall and Otto DL Strack. A discrete numerical model for granular assemblies. *Geotechnique*, 29(1):47–65, 1979.

[CS92] S. L. Crouch and S. Selcuk. Two-dimensional direct boundary integral method for multilayered elastic media. *International Journal for Rock Mechanics and Mining Sciences*, 29(5):491–501, 1992.

[CS97] M. F. Coughlin and D. Stamenović. A tensegrity structure with buckling compression elements: Application to cell mechanics. *Transactions of ASME Journal of Applied Mechanics*, 64:480–486, 1997.

[CS03a] Bruno Capra and Alain Sellier. Orthotropic modelling of alkali-aggregate reaction in concrete structures: numerical simulations. *Mechanics of materials*, 35(8):817–830, 2003.

[CS03b] Y.H. Chai and S.T. Song. Assessment of seismic performance of extended pile–shafts. *Earthquake Engineering & Structural Dynamics*, 32:1937–1954, 2003.

[CS06] Kuang-Tsung Chang and Stein Sture. Microplane modeling of sand behavior under non-proportional loading. *Computers and Geotechnics*, 33(3):177–187, April 2006.

[CS08] IEEE Computer Society. Ieee standard for floating-point arithmetic. Technical Report IEEE Std 754-2008, Institute of Electrical and Electronics Engineers, 2008.

[CS10a] T. Chakraborty and R. Salgado. Dilatancy and shear strength of sand at low confining pressures. *ASCE Journal of Geotechnics and Geoenvironmental Engineering*, 136(3):532, 2010.

[CS10b] H.W. Chandler and C.M. Sands. Including friction in the mathematics of classical plasticity. *INTERNATIONAL JOURNAL FOR NUMERICAL AND ANALYTICAL METHODS IN GEOMECHANICS*, 34:53–72, 2010.

[CS18] Z.B. Cheng and Z.F. Shi. Composite periodic foundation and its application for seismic isolation. *Earthquake Engineering & Structural Dynamics*, 47(4):925–944, 2018.

[CSC92] R G Charlwood, S V Solymar, and D D Curtis. A review of alkali aggregate reactions in hydroelectric plants and dams. In *Proceedings of the International Conference of Alkali-Aggregate Reactions in Hydroelectric Plants and Dams, Fredericton, Canada*, pages 129–135, 1992.

[CSCB18] E. Catalano, R. Stucchi, R. Crapp, and R. Basso. Analysis of tensile stresses arising in the concrete slab of CFRD in seismic areas. In *Hydro 2018 conference proceedings*, Gdansk, Poland, 2018.

[CSD98] M. C. Constantinou, T. T. Soong, and G. F. Dargush. *Passive Energy Dissipation Systems for Structural Retrofit and Design*. Multidisciplinary Center for Earthquake Engineering Research (MCEER), 1998.

[CSF+12] Kevin J. Coppersmith, Lawrence A. Salomone, Chris W. Fuller, Laura L. Glaser, Kathryn L. Hanson, Ross D. Hartleb, William R. Lettis, Scott C. Lindvall, Stephen M. McDuffie, Robin K. McGuire, Gerry L. Stirewalt, Gabriel R. Toro, Davis L. Youngs, Robert R. amd Slayter, Sarkan B. Bozkurt, Valentina Cumbest, Randolph J. Falero Montaldo, Roseanne C. Perman, Allison M. Shumway, Frank H. Syms, and Martitia P. Tuttle. Central and eastern United States (CEUS) seismic source characterization (SSC) for nuclear facilities. Technical report, Electric Power Research Institute, United States., 2012.

[CSG12] Alberto Castellani, Marco Stupazzini, and Roberto Guidotti. Free-field rotations during earthquakes: Relevance on buildings. *Earthquake Engineering & Structural Dynamics*, 41(5):875–891, 2012.

[CSH10] L. Chen, J.F. Shao, and H.W. Huang. Coupled elastoplastic damage modeling of anisotropic rocks. *Computer and Geotechnics*, 37:187–194, 2010.

[CSK+04] K. Onder Cetin, Raymond B. Seed, Armen Der Kiureghian, Kohji Tokimatsu, Leslie F. Harder Jr., Robert E. Kayen, and Robert E. S. Moss. Standard penetration test-based probabilistic and deterministic assessment of seismic soil liquefaction potential. *ASCE Journal of Geotechnics and Geoenvironmental Engineering*, 130(12):1314–1340, December 2004.

[CSLP11] Mirko Corigliano, Laura Scandella, Carlo G Lai, and Roberto Paolucci. Seismic analysis of deep tunnels in near fault conditions: a case study in southern Italy. *Bulletin of Earthquake Engineering*, 9(4):975–995, 2011.

[CT] Giovanni Ciardi and Claudio Tamagnini. Numerical study on the thermo-hydro-mechanical behavior of energy micropiles in hypoplastic soft clay under cyclic thermal loading. *International Journal for Numerical and Analytical Methods in Geomechanics*, n/a(n/a).

[CT11] Luigi Cucci and Andrea Tertulliani. Clues for a relation between rotational effects induced by the 2009 M_w 6.3 L'Aquila (central Italy) earthquake and site and source effects. *Bulletin of the Siesmological Society of America*, 101(3):1109–1120, June 2011.

[CT13] Luigi Cucci and Andrea Tertulliani. The earthquake-rotated objects induced by the 2012 Emilia (northern Italy) seismic sequence: Relation with seismological and geomorphological factors. *Seismological Research Letters*, 84(6):973–981, 2013.

[CTC11] Luigi Cucci, Andrea Tertulliani, and Corrado Castellano. The photographic dataset of the rotational effects produced by the 2009 L'aquila earthquake. *Miscellanea INGV*, 2011.

[CTC+24] Lichiel Cruz, Maria Todorovska, Mingyang Chen, Mihailo Trifunac, Alimu Aihemaiti, Guoliang Lin, and Jianwen Cui. For how large soil shear wave velocity the soil-structure interaction effects on a tall building can be neglected? – a case study, 2024.

[ÇTD09] Ahmet Ş. Çakmak, Rabun M. Taylor, and Eser Durukal. The structural configuration of the first dome of Justinian's Hagia Sophia (A.D. 537–558): An investigation based on structural and literary analysis. *Soil Dynamics and Earthquake Engineering*, 29(4):693–698, April 2009.

[CTG20] Pierfrancesco Cacciola, Alessandro Tombari, and Agathoklis Giaralis. An inerter-equipped vibrating barrier for noninvasive motion control of seismically excited structures. *Structural Control and Health Monitoring*, 27(3):e2474, 2020. e2474 STC-18-0444.R2.

[CTH^{+}13] K. Chockalingam, M. R. Tonks, J. D. Hales, D. R. Gaston, P. C. Millett, and Liangzhe Zhang. Crystal plasticity with jacobian-free newton-krylov. *Computational Mechanics*, 51:617–627, 2013.

[CTJD07] Zhao Cheng, Mahdi Taiebat, Boris Jeremić, and Yannis Dafalias. Modeling and simulation of saturated geomaterials. In *In proceedings of the GeoDenver conference*, 2007.

[CU00] Chung-Che Chou and Chia-Ming Uang. Establishing absorbed energy spectra-an attenuation approach. *Earthquake Engineering & Structural Dynamics*, 29(10):1441–1455, 2000.

[CU03] Chung-Che Chou and Chia-Ming Uang. A procedure for evaluating seismic energy demand of framed structures. *Earthquake Engineering & Structural Dynamics*, 32(2):229–244, 2003.

[Čub] Miško Čubrinovski. Private communications. ..., 2007 –.

[CUJ12] K. Onder Cetin, Berna Unutmaz, and Boris Jeremic. Assessment of seismic soil liquefaction triggering beneath building foundation systems. *Soil Dynamics and Earthquake Engineering*, 43(0):160 – 173, 2012. THIS paper was NOT WRITTEN by me (Boris Jeremic), I discoved it online in June 2015!

[Cun94] Howard G. Cunningham. Wikiwikiweb, a web–system for web pages that anyone can edit. `http://http://c2.com/cgi/wiki?WikiWikiWeb/`, 1994.

[Cun99] Peter Cundall. Private communications, 1998-1999.

[Cur94] D.R. Curran. A pressure-induced strength transition in water-saturated geologic materials. In *International Conference on Mechanical and Behaviour of Materials under Dynamic Loading / Congrès international sur le comportemant mécanique et physique des matériaux sous sollicitations dynamiques*, volume J. Phys. IV France 04 (1994) C8-243-C8-247, 1994. DOI: 10.1051/jp4:1994836.

[ČUS^{+}08] M. Čubrinovski, R. Uzuoka, H. Sugita, K. Tokimatsu, M. Sato, K. Ishihara, Y. Tsukamoto, and T. Kamata. Prediction of pile response to lateral spreading by 3-d soil–coupled dynamic analysis: Shaking in the direction of ground flow. *Soil Dynamics and Earthquake Engineering*, 28(6):421–435, June 2008.

[CVG^{+}18] O. Casablanca, G. Ventura, F. Garescì, B. Azzerboni, B. Chiaia, M. Chiappini, and G. Finocchio. Seismic isolation of buildings using composite foundations based on metamaterials. *Journal of Applied Physics*, 123(17):174903, 2018.

[CVM83] M. P. Collins, F. Vecchio, and G. Melhorn. An international competition to predict the response of reinforced concrete panels. Technical report, Department of Civil Engineering, University of Toronto, Toronto Canada; Civil Research Department, Ontario Hydro, Toronto, Canada Institüt für Massivbau, TH–Darmstadt, Darmstadt, West Germany, 1983.

[CW62] R.W. Clough and E.L. Wilson. Stress analysis of a gravity dam by the finite element method. In *Bulletin RILEM*, 1962.

[CW88] M.A. Crisfield and J. Wills. Solution strategies and softening materials. *Computer Methods in Applied Mechanics and Engineering*, 66:267–289, 1988.

[CW97] Ignacio Carol and Kaspar Willam. Application of analytical solutions in elasto–plasticity to localization analysis of damage models. In *COMPLAS 5*, 17-20 March 1997.

[CWZ^{+}20] Shuai Chen, Bing Wang, Shaowei Zhu, Xiaojun Tan, Jiqiang Hu, Xu Lian, Lianchao Wang, and Linzhi Wu. A novel composite negative stiffness structure for recoverable trapping energy. *Composites Part A: Applied Science and Manufacturing*, 129:105697, 2020.

[CY14] Brian S-J Chiou and Robert R Youngs. Update of the chiou and youngs NGA model for the average horizontal component of peak ground motion and response spectra. *Earthquake Spectra*, 30(3):1117–1153, 2014.

[CYL^{+}23] Guan Chen, Jiashu Yang, Yong Liu, Takeshi Kitahara, and Michael Beer. An energy-frequency parameter for earthquake ground motion intensity measure. *Earthquake Engineering & Structural Dynamics*, n/a(n/a):1–14, 2023.

[CYWL12] Y. Cai, H.-S. Yu, D. Wanatowski, and X. Li. Non-coaxial behavior of sand under various stress paths. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, in print(doi:10.1061/(ASCE)GT.1943-5606.0000854), 2012.

[CZ16] Asskar Janalizadeh Choobbasti and Ali Zahmatkesh. Computation of degradation factors of p-y curves in liquefiable soils for analysis of piles using three-dimensional finite-element model. *Soil Dynamics and Earthquake Engineering*, 89:61 – 74, 2016.

[CZA^{+}20] Andrea Colombi, Rachele Zaccherini, Giulia Aguzzi, Antonio Palermo, and Eleni Chatzi. Mitigation of seismic waves: Metabarriers and metafoundations bench tested. *Journal of Sound and Vibration*, 485:115537, 2020.

[CZH^{+}14] LF Chen, J Zang, AJ Hillis, GCJ Morgan, and AR Plummer. Numerical investigation of wave–structure interaction using openfoam. *Ocean Engineering*, 88:91–109, 2014.

[CZW^{+}20] Changqi Cai, Jiaxi Zhou, Linchao Wu, Kai Wang, Daolin Xu, and Huajiang Ouyang. Design and numerical validation of quasi-zero-stiffness metamaterials for very low-frequency band gaps. *Composite Structures*, 236:111862, 2020.

[D'A58] Jean D'Alembert. *Traité de Dynamique*. Éditions Jacques Gabay, 151 bis rue Saint-Jacques, 75005 Paris, France, 1758. (originally published in 1758), This edition published in 1990.

[DA90] Y. F. Dafalias and E. C. Aifantis. On the microscopic origin of the plastic spin. *Acta Mechanica*, 82(1-2):31–48, 1990.

[DA98] Ricardo Dobry and Tarek Abdoun. Post–triggering response of liquefied sand in the free field and near foundations. In Panos Dakoulas, Mishac Yegian, and Robert D. Holtz, editors, *Proceedings of a Specialty Conference: Geotechnical Earthwuake Engineering and Soil Dynamics III*, Geotechnical Special Publication No. 75, pages 270–300. ASCE, August 1998. 1998.

[DA01] Ricardo Dobry and Tarek Abdoun. Recent studies on seismic centrifuge modeling of liquefaction and its effects on deep foundations. In S. Prakash, editor, *Proceedings of Fourth International Conference on Recent Advances in Geotechnical Earthquake Engineering and Soil Dynamics*, San Diego, March 26-31 2001.

[DAC16] Vasilis K. Dertimanis, Ioannis A. Antoniadis, and Eleni N. Chatzi. Feasibility analysis on the attenuation of strong ground motions using finite periodic lattices of mass-in-mass barriers. *Journal of Engineering Mechanics*, 142(9):04016060, 2016.

[Daf77a] Yannis F. Dafalias. Author's closure on the paper: Il'iushin's postulate and resulting thermodynamic conditions on elastic–plastic coupling. *International Journal of Solids and Structures*, 13:1305–1307, 1977.

[Daf77b] Yannis F. Dafalias. Elasto–plastic coupling within a thermodynamic strain space formulation of plasticity. *International Journal of Non–Linear Mechanics*, 12:327–337, 1977.

[Daf77c] Yannis F. Dafalias. Il'iushin's postulate and resulting thermodynamic conditions on elastic–plastic coupling. *International Journal of Solids and Structures*, 13:239–251, 1977.

[Daf85a] Y. F. Dafalias. The plastic spin. *Journal of Applied Mechanics*, 52:865–871, December 1985.

[Daf85b] Yanis F. Dafalias. A missing link in the macroscopic constitutive formulation of large plastic deformations. In A. Sawczuk and G. Bianchi, editors, *Plasticity Today: Modelling, Methods and Applications*, pages 135–151. Elsevier Applied Sciences Publishers, 1985.

[Daf86a] Y. F. Dafalias. An anisotropic critical state soil plasticity model. *Mechanics Research Communications*, 13(6):341–347, 1986.

[Daf86b] Yannis F. Dafalias. Bounding surface plasticity. I: Mathematical foundations and hypoplasticity. *ASCE Journal of Engineering Mechanics*, 112(9):966–987, September 1986.

[Daf87] Y. F. Dafalias. Issues on the constitutive formulation at large elastoplastic deformation part1: Kinematics. *Acta Mechanica*, 69:119–138, 1987.

[Daf88] Y. F. Dafalias. Issues on the constitutive formulation at large elastoplastic deformation part2: Kinetics. *Acta Mechanica*, 73:121–146, 1988.

[Daf93] Y. F. Dafalias. On multiple spins and texture development. case study: Kinematic and orthotropic hardening. *Acta Mechanica*, 100:171–194, 1993.

[Daf98] Yannis F. Dafalias. Plastic spin: Necessity or redundancy ? *International Journal of Plasticity*, 14(9):909–931, 1998.

[DAO$^+$08] Kent T. Danielson, Stephen A. Akers, James L. O'Daniel, Mark D. Adley, and Sharon B. Garner. Large-scale parallel computation methodologies for highly nonlinear concrete and soil applications. *ASCE Journal of Computing in Civil Engineering*, 22(2):140–146, March/April 2008.

[Dar01] Mehmet Baris Darendeli. *Development of a New Family of Normalized Modulus Reduction And Material Damping Curves*. PhD thesis, The University of Texas at Austin, 2001.

[Das86] P. A. Dashner. Invariance considerations in large strain elasto–plasticity. *Journal of Applied Mechanics*, 53:55–60, 1986.

[DAT12] Suraj S Deshpande, Lakshman Anumolu, and Mario F Trujillo. Evaluating the performance of the two-phase flow solver interfoam. *Computational Science & Discovery*, 5(1):014016, 2012.

[Dav04a] T.A. Davis. Algorithm 832: Umfpack, an unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software*, 30(2):196–199, 2004.

[Dav04b] T.A. Davis. A column pre-ordering strategy for the unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software*, 30(2):165–195, 2004.

[Day77] Steven Milton Day. *Finite Element Analysis of Seismic Scattering Problems*. PhD thesis, University of California at San Diego, 1977.

[DB73] E. H. Davis and J. R. Brooker. Some adaptations of classical plasticity theory for soil stability problems. In *Proceedings of the Symposium on the Role of Plasticity in Soil Mechanics*, pages 24–, 1973.

[dB86] René de Borst. *Non - Linear Analysis of frictional Materials*. PhD thesis, Delft University of Technology, April 1986.

[dB87] R. de Borst. Smeared cracking, plasticity, creep, and thermal loading - a unified approach. *Computer Methods in Applied Mechanics and Engineering*, 62:89–110, 1987.

[dB91] René de Borst. Simulation of strain localization: a reappraisal of the Cosserat continuum. *Engineering Computations*, 8(4):317–332, 1991.

[dB93a] René de Borst. A generalisation of J2-flow theory for polar continua. *Computer Methods in Applied Mechanics and Engineering*, 103(3):347–362, 1993.

[DB93b] Don J. DeGroot and Gregory B. Baecher. Estimating autocovariance of in–situ soil properties. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 119(1):147–166, January 1993.

[dB96] Reint de Boer. Highlights in the historical developments of the porous media theory: Toward a consistent macroscopic theory. *ASME Applied Mechanics Review*, 49(4):201–262, 1996.

[DB09] Sarat Kumar Das and Prabir Kumar Basudhar. Comparison of intact rock failure criteria using various statistical methods. *Acta Geotechnica*, 4(3):223–231, September 2009.

[DB12] Shideh Dashti and Jonathan D. Bray. Numerical simulation of building response on liquefiable sand. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, in print(doi:10.1061/(ASCE)GT.1943-5606.0000853), 2012.

[dBE86] R. de Boer and W. Ehlers. On the problem of fluid– and gass–filled elasto–plastic solids. *International Journal of Solids and Structures*, 22(11):1231–1242, 1986.

[dBEL93] Reint de Boer, Wolfgang Ehlers, and Zhangfang Liu. One-dimensional transient wave propagation in fluid-saturated incompressible porous media. *Archive of Applied Mechanics*, 63(1):59–72, January 1993.

[dBF90] René de Borst and Peter H. Feenstra. Studies in anysotropic plasticity with reference to the hill criterion. *International Journal for Numerical Methods in Engineering*, 29:315–336, 1990.

[dBK95] R. de Boer and S. J. Kowalski. Thermodynamics of fluid–saturated porous medial with a phase change. *Acta Mechanica*, 109:167–189, 1995.

[DBL95a] FCP De Barros and JE Luco. Dynamic response of a two-dimensional semi-circular foundation embedded in a layered viscoelastic half-space. *Soil Dynamics and Earthquake Engineering*, 14(1):45–57, 1995.

[dBL95b] Francisco C. P. de Barros and J. Enrique Luco. Identification of foundation impedance functions and soil properties from vibration tests of the Hualien containment model. *Soil Dynamics and Earthquake Engineering*, 14:229–248, 1995.

[DBM$^+$] Somayajulu L. N. Dhulipala, Chandrakanth Bolisetti, Lynn B. Munday, William M. Hoffman, Ching-Ching Yu, Faizan Ul Haq Mir, Fande Kong, Alexander D. Lindsay, and Andrew S. Whittaker. Development, verification, and validation of comprehensive acoustic fluid-structure interaction capabilities in an open-source computational platform. *Earthquake Engineering & Structural Dynamics*, 51(10):2188–2219.

[DBO01] Manas K Deb, Ivo M Babuška, and J Tinsley Oden. Solution of stochastic partial differential equations using galerkin finite element techniques. *Computer Methods in Applied Mechanics and Engineering*, 190(48):6359–6372, 2001.

[DBS95] A. Drescher, B. Birgisson, and K. Shah. A model for water–saturated loose sand. In G. N. Pande and S. Pietruszczak, editors, *Numerical Models in Geomechanics*, volume V, pages 109–112. Balkema, Rotterdam, 1995.

[dBSG96] R. de Boer, R. L. Schiffman, and R. E. Gibson. The origins of the theory of consolidation: The Terzaghi–Fillunger dispute. *Géotechnique*, 46(2):175–186, 1996.

[dBSMP93] R. de Borst, L. J. Sluys, H.-B. Mühlhaus, and J. Pamin. Fundamental issues in finite element analysis of localization of deformation. *Engineering Computations*, 10:99–121, 1993.

[dBV84] R. de Borst and P. A. Vermeer. Possibilities and limitations of finite elements for limit analysis. *Geotechnique*, 34(2):199–210, 1984.

[DC91] Emmanuel Detournay and Alexander H.-D. Cheng. Plane strain analysis of a stationary hydraulic fracture in poroelastic media: Stationary fracture. *International Journal of Solids and Structures*, 27(13):1645–1662, 1991.

[dCD00] Eduardo Gomes Dutra do Carmo and André Vicinius Duarte. A discontinuous finite element–based domain decomposition method. *Computer Methods in Applied Mechanics and Engineering*, 190:825–843, 2000.

[DCG90] Eduardo N. Dvorkin, Alberto M. Cuitiño, and Gustavo Gioia. Finite elements with displacement interpolated embedded localization lines insensitive to mesh size and distortions. *International Journal for Numerical Methods in Engineering*, 30:541–564, 1990.

[DCH] William Dearholt, Steven Castillo, and Gary Hennigan. *Parallel Sparse Irregular System Solvers mp_solve*. New Mexico State University. `http://emlab2.nmsu.edu/mp_solve/index.html`

[DCH89] P. G. Duxbury, M. A. Crisfield, and G. W. Hunt. Benchtests for geometric nonlinearities. *Computers and Structures*, 33(1):21–29, 1989.

[DCKP99] Mandar M. Dewoolkar, A. H. C. Chan, Hon-Yim Ko, and Ronald Y. S. Pak. Finite element simulations of seisimic effects on retaining walls with liquefiable backfills. *Submitted for publication to the: International Journal for Numerical and Analytical Methods in Geomechanics*, 1999.

[DCR] Angshuman Deb, Joel P. Conte, and José I. Restrepo. Comprehensive treatment of uncertainties in risk-targeted performance-based seismic design and assessment of bridges. *Earthquake Engineering & Structural Dynamics*, n/a(n/a).

[DCR$^+$22] B. Dasgupta, N. Chokshi, M.K. Ravindra, R.J. Budnitz, J. Stamatakos, and O. Pensado. Risk-informed performance-based seismic design approach for advanced reactors. In *Transactions, SMiRT-26*, Berlin/Potsdam, Germany, July 10-15 2022. Division VII.

[DCY90] Timothy A. Davis and Pen Cung Yew. A nondeterministic parallel algorithm for general unsymmetric sparse LU factorization. *SIAM Journal of Matrix Analysis and Applications*, 11(3):383–402, 1990.

[DD92] E. Detournay and P. Defourny. A phenomenological model for the drilling action of drag bits. *International Journal for Rock Mechanics and Mining Sciences*, 29(1):13–23, 1992.

[DD93] A. Drescher and E. Detournay. Limit load in translational failure mechanisms for associative and non-associative materials. *Geotechnique*, 43(3):443–456, 1993.

[DD97a] Timothy A. Davis and Iain S. Duff. A combined unifrontal/multifrontal method for unsymmetric sparse matrices. Technical Report TR97-016 (UofF) and TR-97-046 (RAL), University of Florida and Rutherford Appleton Laboratory, 1997.

[DD97b] Timothy A. Davis and Iain S. Duff. An unsymmetric–pattern mutlifrontal method for sparse LU factorization. *SIAM Journal of Matrix Analysis and Applications*, 18(1):140–158, 1997.

[DD99] T.A. Davis and I.S. Duff. A combined unifrontal/multifrontal method for unsymmetric sparse matrices. *ACM Transactions on Mathematical Software*, 25(1):1–19, 1999.

[DD04] James P. Doherty and Andrew J. Deeks. Semi-analytical far field model for three-dimensional finite-element analysis. *International Journal for Numerical and Analytical Methods in Geomechanics*, 28:1121–1140, 2004.

[DDG$^+$16] Jack Dongarra, Iain Duff, Mark Gates, Azzam Haidar, Sven Hammarling, Nicholas J Higham, Jonathon Hogg, Pedro Valero-Lara, Samuel D Relton, Stanimire Tomov, et al. A proposed api for batched basic linear algebra subprograms. 2016.

[de 87] R. de Borst. Computation of post–biffurcation and post–failure behavior of strain–softening solids. *Computers & Structures*, 25(2):211–224, 1987.

[DE56] T. C. Doyle and J. L. Ericksen. Nonlinear elasticity. In H. L. Dryden and Th. von Kármán, editors, *Advances in Applied Mechanics*, volume IV, pages 53–115. Academic Press, 1956.

[DE98] P. Dakoulas and A. Eltaher. Nonlinear seismic effective stress dam–foundation interaction. In Panos Dakoulas, Mishac Yegian, and Robert D. Holtz, editors, *Proceedings of a Specialty Conference: Geotechnical Earthwuake Engineering and Soil Dynamics III*, Geotechnical Special Publication No. 75, pages 866–877. ASCE, August 1998. 1998.

[Dee91] Dick P Dee. Prescribed solution forcing method for model verification. In *Hydraulic Engineering, Proc. 1991 National Conf. of Hydr. Engrg*, pages 734–739. Citeseer, 1991.

[DEG$^+$99] James W. Demmel, Stanley C. Eisenstat, John R. Gilbert, Xiaoye S. Li, and Joseph W. H. Liu. A supernodal approach to sparse partial pivoting. *SIAM J. Matrix Analysis and Applications*, 20(3):720–755, 1999.

[Deh08] Azadeh Riahi Dehkordi. *3D Finite Element Cosserat Continuum Simulation of Layered Geomaterials*. PhD thesis, University of Toronto, 2008.

[DEH$^+$15] Iain Dinwoodie, Ole-Erik V. Endrerud, Matthias Hofmann, Rebecca Martin, and Iver Bakken Sperstad. Reference cases for verification of operation and maintenance simulation models for offshore wind farms. *Wind Engineering*, 39(1):1–14, 2015.

[Dem69] B. P. Demidovich. *Problems in Mathematical Analysis*. Russian Monographs and Texts on Advanced Mathematics and Physics. Gordon and Breach, Science Publishers, 1969. (Б. П. Демидович). Translated from Russian by G. Yankovsky.

[Dep18] USA Department of Defense. DoD instruction 5000.61. online, October 2018.

[Des81] CS Desai. Behavior of interfaces between structural and geologic media. Technical report, University of Missouri–Rolla, 1981.

[DEW94] W. J. Nellis D. Erskine and S. T. Weir. Shock wave profile study of tuff from the nevada test site. *Journal of Geophysical Research*, 99(B8):15,529–15,537, 1994.

[DF] Spyridon Diamantopoulos and Michalis Fragiadakis. Modeling of rocking frames under seismic loading. *Earthquake Engineering & Structural Dynamics*, n/a(n/a):1–21.

[DF08] M. Dryden and G.L. Fenves. Validation of numerical simulations of a two-span reinforced concrete bridge. In *Proceedings of the 14th World Conference on Earthquake Engineering*, October 2008.

[dG02] Joel de Guzman. MAJOR BREAK-THROUGH !!! Look Ma, no virtual functions! `http://sourceforge.net/mailarchive/forum.php?thread_id=529112&forum_id=1595`, March 2002.

[DG08] P. Dakoulas and G. Gazetas. Insight into seismic earth and water pressures against caisson quay walls. *Géotechnique*, 58(2):95–111, 2008.

[DGF02] V De Gennaro and R Frank. Elasto-plastic analysis of the interface behaviour between granular media and structure. *Computers and Geotechnics*, 29(7):547–572, 2002.

[DGG12] Vasileios A Drosos, Nikos Gerolymos, and George Gazetas. Constitutive model for soil amplification of ground shaking: Parameter calibration, comparisons, validation. *Soil Dynamics and Earthquake Engineering*, 42:255–274, 2012.

[DGH82] Jean Donea, S Giuliani, and Jean-Pierre Halleux. An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions. *Computer methods in applied mechanics and engineering*, 33(1-3):689–723, 1982.

[DGL94] Don J. DeGroot, John T. Germain, and Charles C. Ladd. Effect of nonuniform stresses on measured dss stress–strain behavior. *ASCE Journal of Geothechnical Engineering*, 120(5):892–912, 1994.

[DGL99] James W. Demmel, John R. Gilbert, and Xiaoye S. Li. An asynchronous parallel supernodal algorithm for sparse gaussian elimination. *SIAM J. Matrix Analysis and Applications*, 20(4):915–952, 1999.

[DGL^{+}12] V. Drosos, T. Georgarakos, M. Loli, I. Anastasopoulos, O. Zarzouras, and G. Gazetas. Soil-foundation-structure interaction with mobilization of bearing capacity: Experimental study on sand. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 138(11):1369–1386, November 2012.

[DH80] Yannis F. Dafalias and Leonard R. Herrmann. A bounding surface soils plasticity model. In G. N. Pande and O. C. Zienkiewicz, editors, *Internatinoal Symposium on Soils under Cyclic and Transient Loadings*, pages 335–345, Swansea, January 1980. A. A. Balkema.

[DH82] Y. F. Dafalis and L. R. Herrmann. Bounding surface formulation of soil plasticity. In G. N. Pande and O. C. Zienkiewicz, editors, *Soil Mechanics – Transient and Cyclic Loads*, pages 253–282. John Wiley and Sons Ltd., 1982.

[DH85] Yannis F. Dafalias and Leonard R. Herrmann. Bounding surface plasticity. II: Application to isotropic cohesive soils. *ASCE Journal or Engineering Mechanics*, 112(12):1263–1291, 1985.

[DH94] Thierry Delmarcelle and Lambertus Hesselink. The topology of symmetric, second-order tensor fields. In *Proceedings Visualization '94,*, pages 140–147, Washington, DC, USA, 17-21 Oct. 1994 1994. Los Alamitos, CA: IEEE Computer Society Press.

[DHL^{+}16] Tingxing Dong, Azzam Haidar, Piotr Luszczek, Stanimire Tomov, Ahmad Abdelfattah, and Jack Dongarra. Magma batched: A batched blas approach for small matrix factorizations and applications on gpus. Technical report, Innovative Computing Laboratory, University of Tennessee, Knowville, 2016.

[DHR^{+}15] Douglas S. Dreger, Mong-Han Huang, Arthur Rodgers, Taka'aki Taira, and Kathryn Wooddell. Kinematic finite-source model for the 24 August 2014 South Napa, California, earthquake from joint inversion of seismic, GPS, and InSAR data. *Seismological Research Letters*, 86(2A):327–334, March/April 2015.

[Dic94] Stephen Eugene Dickenson. *Dynamic response of soft and deep cohesive soils during the Loma Prieta earthquake of October 17, 1989.* PhD thesis, University of California, Berkeley, 1994.

[Div98] Dejan Divac. Prvonek dam, numerical model of filtration. Technical report, Institute for Water Management "Jaroslav Černi, Belgrade, Yugoslavia, 1998. In Serbian.

[DJB84] Eduardo N. Dvorkin and Klaus Jürgen Bathe. A continuum mechanics based four-node shell element for general nonlinear analysis. *Engineering Computations*, 1:77–88, March 1984.

[DJH13] M Davoodi, MK Jafari, and N Hadiani. Seismic response of embankment dams under near-fault and far-field ground motion excitation. *Engineering Geology*, 158:66–76, 2013.

[DK86] Šerif Dunica and Branislav Kolundžija. *Nelinarna Analiza Konstrukcija.* Gradjevinski fakultet Univerziteta u Beogradu, Bulevar revolucije 73 i IRO "Naučna knjiga", Beograd, Uzun–Mirkova 5, 1986. Nonlinear Analysis of Structures, In Serbian.

[DKN93] Andrew Drescher, Jong R. Kim, and David E. Newcomb. Permanent deformation in asphalt concrete. *ASCE Journal of Materials and Engineering*, 5(1):112–128, February 1993.

[DL96] Pompiliu Donescu and Tod A. Laursen. A generalized object–oriented approach to solving ordinary and partial differential equations using finite elements. *Finite Elements in Analysis and Design*, 22:93–107, 1996.

[DLMV24] Maria-Eleni Dasiou, Christos G. Lachanas, Vasileios E. Melissianos, and Dimitrios Vamvatsikos. Seismic performance of the temple of Aphaia in Aegina island, Greece. *Earthquake Engineering & Structural Dynamics*, 53(2):573–592, 2024.

[dLS12] Maria Paola Santisi d'Avila, Luca Lenti, and Jean-François Semblat. Modelling strong seismic ground motion: three-dimensional loading path versus wavefield polarization. *Geophysical Journal International*, 190:1607–1624, 2012.

[DM77] J. E. Dennis, Jr. and Jorge J. Moré. Quasi-newton methods, motivation and theory. *SIAM Review*, 19:46–89, 1977.

[DM98] Luis D Decanini and Fabrizio Mollaioli. Formulation of elastic earthquake input energy spectra. *Earthquake engineering & structural dynamics*, 27(12):1503–1522, 1998.

[DM01] Luis D Decanini and Fabrizio Mollaioli. An energy-based methodology for the assessment of seismic demand. *Soil Dynamics and Earthquake Engineering*, 21(2):113–137, 2001.

[DM04] Yannis F. Dafalias and Majid T. Manzari. Simple plasticity sand model accounting for fabric change effects. *ASCE Journal of Engineering Mechanics*, 130(6):622–634, June 2004.

[DM05] J. Dvorkin and G. Mavko. P-wave attenuation in reservoir and non-reservoir rock. In *67th EAGE Conference & Exhibition*. SPE, EAGE, June 2005.

[dM10] Florent de Martin. *Influence of the Nonlinear Behavior of Soft Soils on Strong Ground Motions*. PhD thesis, Ecole Centrale Paris, 2010.

[DM14] M. Dehghan and V. Mohammadi. The numerical solution of Fokker-Planck equation with radial basis functions (rbfs) based on the meshless technique of Kansa's approach and Galerkin method. *Engineering Analysis with Boundary Elements*, 47(0):38 – 63, 2014.

[DMAP24] Wulf G. Dettmer, Eugenio J. Muttio, Reem Alhayki, and Djordje Perić. A framework for neural network based constitutive modelling of inelastic materials. *Computer Methods in Applied Mechanics and Engineering*, 420:116672, 2024.

[DMFRDSLC11a] Sofia Costa D'Aguiar, Arezou Modaressi-Farahmand-Razavi, Jaime Alberto Dos Santos, and Fernando Lopez-Caballero. Elastoplastic constitutive modelling of soil–structure interfaces under monotonic and cyclic loading. *Computers and Geotechnics*, 38(4):430–447, 2011.

[DMFRdSLC11b] Sofia Costa D'Aguiar, Arezou Modaressi-Farahmand-Razavi, Jaime Alberto dos Santos, and Fernando Lopez-Caballero. Elastoplastic constitutive modelling of soil-structure interfaces under monotonic and cyclic loading. *Computers and Geotechnics*, 38:430–447, 2011.

[Dmi04] Sergey Dmitriev. Language oriented programming: The next programming paradigm. published online `http://www.onboard.jetbrains.com/is1/articles/04/10/lop/`, November 2004.

[DMP06] Yannis F. Dafalias, Majid T. Manzari, and Achilleas G. Papadimitriou. SANICLAY: simple anisotropic clay plasticity model. *International Journal for Numerical and Analytical Methods in Geomechanics*, 30(12):1231–1257, 2006.

[DN88] CS Desai and BK Nagaraj. Modeling for cyclic normal and shear behavior of interfaces. *Journal of engineering mechanics*, 114(7):1198–1217, 1988.

[DN92] Ricardo Dobry and T–T Ng. Discrete modelling of stress–strain behavior of media at small and large strain. *Engineering Computations*, 9:129–143, 1992.

[DNH99] Andrew Drescher, David Newcomb, and Thor Heimdahl. Deformability of shredded tires. Technical Report MN/RC - 1999-13, University of Minnesota, 1999.

[DNP+04] Bert J. Debusschere, Habib N. Najim, Philippe P. Pebay, Omar M. Knio, Roger G. Ghanem, and Oliver P. Maitre. Numerical challenges in the use of polynomial chaos representations for stochastic processes. *SIAM Journal on Scientific Computing*, 26(2):698–719, 2004.

[DO04] A. Dorfmann and R.W. Ogden. A constitutive model for the mullins effect with permanent set in particle-reinforced rubber. *International Journal of Solids and Structures*, 41:1855–1878, 2004.

[Dow93] Kevin Dowd. *High Performance Computing*. A Nutshell Handbook. O'Reilly & Associates, Inc, 103 Morris Street, Suite A, Sebastopol, CA 95472, first edition, June 1993.

[DP52] Daniel Charles Drucker and William Prager. Soil mechanics and plastic analysis or limit design. *Quarterly of applied mathematics*, 10(2):157–165, 1952.

[DP75] YF Dafalias and EP Popov. A model of nonlinearly hardening materials for complex loading. *Acta mechanica*, 21(3):173–192, 1975.

[DP86] R. Delpak and V. Peshkam. A study of the influnece of hierachical nodes on the performance of selected parametric elements. *International Journal for Numerical Methods in Engineering*, 22:153–171, 1986.

[DP88] I Demirdžić and M Perić. Space conservation law in finite volume calculations of fluid flow. *International journal for numerical methods in fluids*, 8(9):1037–1050, 1988.

[DP92a] Yves-Dominique Dubois-Pelerin. *Object Oriented Finite Elements: Programming Concepts and Implementation*. PhD thesis, Ecole Polytechnique Federale de Lausanne, 1992.

[DP92b] Yves-Dominique Dubois-Pelerin. *Object Oriented Finite Elements: Programming Concepts and Implementation*. PhD thesis, Ecole Polytechnique Federale de Lausanne, 1992.

[DP94] RA Day and DM Potts. Zero thickness interface elements-numerical stability and application. *International Journal for numerical and analytical methods in geomechanics*, 18(10):689–708, 1994.

[DP97] David Durban and Panos Papanastasiou. Elastoplastic reponse of pressure sensitive solids. *International Journal for Numerical and Analytical Methods in Geomechanics*, 21:423–441, 1997.

[DPB06] Colin T. Davie, Chris J. Pearce, and Nenad Bićanić. Coupled heat and moisture transport in concrete at elevated temperatures - effects of capillary pressure and adsorbed water. *Numerical Heat Transfer, Part A: Applications*, 49(8):733–763, 2006.

[DPO93] Martin Dutko, Djordje Perić, and D. R. J. Owen. Universal anisotropic yield criterion based on superquadratic functional representation: Part1. algorithmic issues and accuray analysis. *Computer methods in applied mechanics and engineering*, 109:73–93, 1993.

[DPP98] Yves Dubois-Pélerin and Pierre Pegon. Object-oriented programming in nonlinear finite element analysis. *Computers & Structures*, 67(4):225–241, 1998.

[dPPP12] C. di Prisco, M. Pastor, and F. Pisanò. Shear wave propagation along infinite slopes: A theoretically based numerical study. *International Journal for Numerical and Analytical Methods in Geomechanics*, 36(5):619–642, 2012.

[DPR94] Eduardo N. Dvorkin, Daniel Pantuso, and Eduardo A. Repetto. A finite element formulation for finite strain elasto – plastic analysis based on mixed interpolation of tensorial components. *Computer Methods in Applied Mechanics and Engineering*, 114:35–54, 1994.

[dPSZ07] C. di Prisco, M. Stupazzini, and C. Zambelli. Nonlinear sem numerical analyses of dry dense sand specimens under rapid and dynamic loading. *International Journal for Numerical and Analytical Methods in Geomechanics*, 31(6):757–788, 2007.

[DPZ92] Yves Dubois-Pélerin and Thomas Zimmermann. *Object Oriented Finite Element Programming: Theory and C++ Implementation for FEM_Objects C++ 01*. Elmepress International, P.O.Box 2 CH 1015 Lausanne 15, Switzerland, 1992.

[DPZ93a] Yves Dubois-Pèlerein and Thomas Zimmermann. Object–oriented finite element programing: Iii. an efficient implementation in c++. *Computer Methods in Applied Mechanics and Engineering*, 108:165–183, 1993.

[DPZ93b] Yves Dubois-Pèlerein and Thomas Zimmermann. Object–oriented finite element programing: Iii. an efficient implementation in c++. *Computer Methods in Applied Mechanics and Engineering*, 108:165–183, 1993.

[DR87] Jr. Dodds and H. Robert. Numerical techniques for plasticity computations in finite element analysis. *Computers & Structures*, 26(5):767–779, 1987.

[DR89] Y. F. Dafalias and M. M. Rashid. The effect of plastic spin on anysotropic material behavior. *International Journal of Plasticity*, 5:227–246, 1989.

[DR02] A. Düster and E. Rank. A p-version finite element approach for two– and three– dimensional problems of j_2 flow theory of non–linear isotropic hardening. *International Journal for Numerical Methods in Engineering*, 53:49–63, 2002.

[DR04] Wulf Dettmer and Stefanie Reese. On the theoretical and numerical modelling of Armstrong-Frederick kinematic hardening in the finite strain regime. *Computer Methods in Applied Mechanics and Engineering*, 193(1-2):87–116, 2004.

[DR05] M.K. Dash D. Roy. Explorations of a family of stochastic Newmark methods in engineering dynamics. *Comput. Methods Appl. Mech. Engrg*, 194(45-47):4758–4796, November 2005.

[DRD10] M Dolinski, D Rittel, and A Dorogoy. Modeling adiabatic shear failure from energy considerations. *Journal of the Mechanics and Physics of Solids*, 58(11):1759–1775, 2010.

[dRFI19] Zachary del Rosario, Richard W. Fenrich, and Gianluca Iaccarino. Cutting the double loop: Theory and algorithms for reliability-based design optimization with parametric uncertainty. *International Journal for Numerical Methods in Engineering*, 118(12):718–740, 2019.

[dRR^{+}25] Wanderson F. dos Santos, Alina S.L. Rodrigues, Igor A. Rodrigues Lopes, Francisco M. Andrade Pires, Sergio P.B. Proença, and Zilda C. Silveira. Analysis of a novel 3D-printed mechanical metamaterial with tension-induced undulation: Experimental and numerical investigations. *International Journal of Solids and Structures*, 317:113402, 2025.

[Dru53] Daniel Charles Drucker. Coulomb friction, plasticity, and limit loads. Technical report, BROWN UNIV PROVIDENCE RI DIV OF APPLIED MATHEMATICS, 1953.

[Dru56] Daniel Charles Drucker. On uniqueness in the theory of plasticity. *Quarterly of Applied Mathematics*, pages 35–42, 1956.

[Dru57] Daniel Charles Drucker. A definition of stable inelastic material. Technical report, DTIC Document, 1957.

[Dru85] D. C. Drucker. Appropriate simple idealizations for finite plasticity. In A. Sawczuk and G. Bianchi, editors, *Plasticity Today: Modelling, Methods and Applications*, pages 47–59. Elsevier Applied Sciences Publishers, 1985.

[Dru88] D. C. Drucker. Comments on the modeling of the behaviour of sand. In A. Saada and G. Bianchini, editors, *Constitutive Equations for Granular Non–Cohesive Soils*, pages 695–697. A. A. Balkema, July 1988.

[Dry06] M. Dryden. Validation of simulations of a two–span reinforced concrete bridge. Submitted in partial satisfaction of requirements for a Ph.D. degree at the University of California at Berkeley, 2006.

[DS83] J. E. Dennis, Jr. and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice Hall , Engelwood Cliffs, New Jersey 07632., 1983.

[DS84] Chandakant S. Desai and Hema J. Siriwardane. *Constitutive Laws for Engineering Materials With Emphasis on Geologic Materials*. Prentice–Hall, Inc. Englewood Cliffs, NJ 07632, 1984.

[DS11] M Dysli and W Steiner. *Foundation Engineering*. McGraw Hill Book Company, 2011.

[DSF86] C. S. Desai, S. Somasundaram, and G. Frantziskonis. A hierarchical approach for constitutive modelling of geological materials. *International Journal for Numerical and Analytical Methods in Geomechanics*, 10:225–257, 1986.

[DSH17] Derya Deniz, Junho Song, and Jerome F Hajjar. Energy-based seismic collapse criterion for ductile planar structural frames. *Engineering Structures*, 141:1–13, 2017.

[DSH18] Derya Deniz, Junho Song, and Jerome F Hajjar. Energy-based sidesway collapse fragilities for ductile structural frames under earthquake loadings. *Engineering Structures*, 174:282–294, 2018.

[dSKB16] Ketson R. M. dos Santos, Ioannis A. Kougioumtzoglou, and André T. Beck. Incremental dynamic analysis: A nonlinear stochastic dynamics perspective. *ASCE Journal of Engineering Mechanics*, 142(10):X1–X7, October 2016.

[dSL13] Maria Paola Santisi d'Avila, Jean-François Semblat, and Luca Lenti. Strong ground motion in the 2011 Tohoku earthquake: A one-directional three-component modeling. *Bulletin of the Seismological Society of America*, 103(2B):1394–1410, May 2013.

[dSNP96] E. A. de Souza Neto and Djordje Perić. A computational framework for a class of fully coupled models for elastoplastic damage at finite strains with reference to the linearization aspects. *Computer Methods in Applied Mechanics and Engineering*, 130:179–193, 1996.

[dSNPDO95] E. A. de Souza Neto, D. Perić, M. Dutko, and D. R. J. Owen. Finite strain implementation of an elastoplastic model for crushable foam. In N.-E. Wiberg, editor, *Advances in Finite Element Technology*, pages 174–188, Barcelona, 1995. CIMNE.

[dSNPO94a] E. A. de Souza Neto, Djordje Perić, and D. R. J. Owen. A model for elastoplastic damage at finite strains: Algorithmic issues and applications. *Engineering Computations*, 11:257–281, 1994.

[dSNPO94b] E. A. de Souza Neto, Djordje Perić, and D. R. J. Owen. A phenomenological three–dimensional rate–independent continuum damage model for highly filled polymers: Formualation and computational aspects. *Journal of Mechanics and Physics of Solids*, 42(10):1533–1550, 1994.

[dSNPO95] E. A. de Souza Neto, Djordje Perić, and D. R. J. Owen. Finite elasticity in spatial description: Linearization aspects with 3-D membrane applications. *International Journal for Numerical Methods in Engineering*, 38:3365–3381, 1995.

[dSNPO11] Eduardo A de Souza Neto, Djordje Perić, and David Roger Jones Owen. *Computational methods for plasticity: theory and applications*. John Wiley & Sons, 2011.

[DST^{+}02] YF Dafalias, D Schick, C Tsakmakis, K Hutter, and H Baaser. A simple model for describing yield surface evolution. In *Lecture notes in applied and computational mechanics*, pages 169–201. Springer, 2002.

[DSW93] Andreas Dietsche, Paul Steinmann, and Kaspar Willam. Micropolar elastoplasticity and its role in localization. *International Journal of Plasticity*, 9(7):813–831, 1993.

[DSW^{+}23] Anna Dalklint, Filip Sjövall, Mathias Wallin, Seth Watts, and Daniel Tortorelli. Computational design of metamaterials with self contact. *Computer Methods in Applied Mechanics and Engineering*, 417:116424, 2023.

[DSXD05] Patrick Dangla, Jean-François Semblat, Haihong Xiao, and Nicolas Delépine. A simple and efficient regularization method for 3D bem: application to frequency-domain elastodynamics. *Bulletin of the Seismological Society of America*, 95(5):1916–1927, 2005.

[DTBP25] Daniel M. Dowden, Arman Tatar, Jeffrey W. Berman, and Shiling Pei. Shake-table test of a seismically resilient 10-story mass timber building with supplemental uplift friction dampers. *Journal of Structural Engineering*, 151(1):04024199, 2025.

[Dun95] Šerif Dunica. *Otpornost materijala, uvod u mehaniku deformabilnog tela, opšta teorija, teorija greda, 107 rešenih zadataka*. Gradjevinska Knjiga i Grosknjiga D.O.O., 1995. ISBN 86-485-0020-6, (Strength of Materials, Introduction to the Mechanics of Deformable Bodies, General Theory, Beam Theory, 107 Solved Problems, in Serbian).

[Dun96] James Michael Duncan. State of the art: Limit equilibrium and finite-element analysis of slopes. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 122(7):577–596, July 1996.

[Dun00] J. Michael Duncan. Factors of safety and reliability in geotechnical engineering. *ASCE Journal of Geotechnical and Geoenvironmental Engineering,*, 126(4):307–316, April 2000.

[DV04] Jacques Desrues and Gioacchino Viggiani. Strain localization in sand: an overview of the experimental results obtained in Grenoble using stereophotogrametry. *International Journal for Numerical and Analytical Methods in Geomechanics*, 28:279–321, 2004.

[DVH90] Andrew Drescher, Ioannis Vardoulakis, and Chunhya Han. A biaxial apparatus for testing soils. *Geotechnical Testing Journal*, 13(3):226–234, September 1990.

[DVTPH13] Cecile Daversin, Stephane Veys, Christophe Trophime, and Christophe Prud Homme. A reduced basis framework: Application to large scale non-linear multi-physics problems. In *ESAIM: Proceedings*, volume 43, pages 225–254. EDP Sciences, 2013.

[DW92] Andreas Dietsche and Kaspar J. Willam. Localization analysis of elasto-plastic Cosserat continua. In J. W. Wu and K. C. Valanis, editors, *Damage Mechanics and Localization*, volume AMD-142, MD-34, pages 109 – 123, The 345 East 47th street New York, N.Y. 10017, November 1992. The American Society of Mechanical Engineers.

[DW09] Jason T. DeJong and Zachary J. Westgate. Role of initial state, material properties, and confinement condition on local and global soil-structure interface behavior. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 135(11):1646–1660, November 2009.

[DWR06] Jason T Dejong, David J White, and Mark F Randolph. Microscale observation and modeling of soil-structure interface behavior using particle image velocimetry. *Soils and Foundations*, 46(1):15–28, 2006.

[DWS12] Sheng Dai, Frank Wuttke, and J. Carlos Santamarina. Coda wave analysis to monitor processes in soils. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, publihsed online(November 28, 2012. doi:10.1061/(ASCE)GT.1943-5606.0000872), 2012.

[DWS13] Sheng Dai, Frank Wuttke, and J. Carlos Santamarina. Coda wave analysis to monitor processes in soils. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 139(9):1504–1511., September 2013.

[DYY^{+}15] Ahmet Anıl Dindar, Cem Yalçın, Ercan Yüksel, Hasan Özkaynak, and Oral Büyüköztürk. Development of earthquake energy demand spectra. *Earthquake Spectra*, 31(3):1667–1689, 2015.

[DZLS84] CS Desai, MM Zaman, JG Lightner, and HJ Siriwardane. Thin-layer element for interfaces and joints. *International Journal for Numerical and Analytical Methods in Geomechanics*, 8(1):19–43, 1984.

[DZW06] Xiuli Du, Mi Zhao, and Jinting Wang. A stress artificial boundary in FEA for near-field wave problem (in chinese). *Chinese Journal of Theoretical and Applied Mechanics*, 38(1):49, 2006.

[ea19] David B. McCallen et al. Large-scale laminar soil box (LLSB). Technical Report 201911, University of Nevada, Reno, 2019.

[EACB19] Hamed Ebrahimian, Rodrigo Astroza, Joel P. Conte, and Robert R. Bitmead. Information-theoretic approach for identifiability assessment of nonlinear structural finite-element models. *Journal of Engineering Mechanics*, 145(7):04019039, 2019.

[EB90] Adrian Luis Eterovic and Klaus-Jürgen Bathe. A hyperelastic – based large strain elasto – plastic constitutive formulation with combined isotropic – kinematic hardening using the logarithmic stress and strain measures. *International Journal for Numerical Methods in Engineering*, 30:1099–1114, 1990.

[EB96] David Elata and James G. Berryman. Contact force-displacement laws and the mechancial behavior of random packs of identical spheres. *Mechanics of Materials*, 24(3):229–240, November 1996.

[EBF^{+}22] Simona Esposito, Alessio Botta, Melania De Falco, Adriana Pacifico, Eugenio Chioccarelli, Antonio Pescapè, Antonio Santoc, and Iunio Iervolino. Seismic risk analysis of a data communication network. *Sustainable and Resilient Infrastructure*, 0(0):1–18, 2022.

[EC05] Itai Einav and Mark J. Cassidy. A framework for modelling rigid footing behaviour based on energy principles. *Computers and Geotechnics*, 32(7):491–504, October 2005.

[ECJ^{+}21] E. El Haber, C. Cornou, D. Jongmans, F. Lopez-Caballero, D. Youssef Abdelmassih, and T. Al-Bittar. Impact of spatial variability of shear wave velocity on the lagged coherency of synthetic surface ground motions. *Soil Dynamics and Earthquake Engineering*, 145:106689, 2021.

[Eck89] Bruce Eckel. *Using C++*. Osborne McGraw – Hill, 1989.

[ECR^{+}21] Lelli Van Den Einde, Joel P. Conte, José I. Restrepo, Ricardo Bustamante, Marty Halvorson, Tara C. Hutchinson, Chin-Ta Lai, Koorosh Lotfizadeh, J. Enrique Luco, Machel L. Morrison, Gilberto Mosqueda, Mike Nemeth, Ozgur Ozcelik, Sebastian Restrepo, Andrés Rodriguez, P. Benson Shing, Brad Thoen, and Georgios Tsampras. NHERI@UC San Diego 6-DOF large high-performance outdoor shake table facility. *Frontiers in Build Environment*, 6:1–21, January 2021.

[EE98] Bahaa M. Elaidi and Mohsen A. Eissa. Soil-structure interaction in fuel handling building. *Nuclear Engineering and Design*, 181(1-3):145 – 156, 1998.

[EEJH96] Kenneth Eriksson, Don Estep, Claes Johnson, and Peter Hansbo. *Computational Differential Equations*. Cambridge University Press, 1996.

[EEKAI15] Magdi El-Emam, Zahid Khan, Jamal Abdalla, and Muhammad Irfan. Local site effects on seismic ground response of major cities in UAE. *Natural Hazard*, (ISSN 0921-030X, DOI 10.1007/s11069-015-1873-9), 2015.

[EEOS05] Howard C. Elman, Oliver G. Ernst, Dianne P. O'Leary, and Michael Stewart. Efficient iterative algorithms for the stochastic finite element method with application to acoustic scattering. *Computer Methods in Applied Mechanics and Engineering*, 194(9-11):1037–1055, 18 March 2005.

[EF97] Erman Evgin and Kazem Fakharian. Effect of stress paths on the behaviour of sand steel interfaces. *Canadian geotechnical journal*, 33(6):853–865, 1997.

[EFK+00] D. Engert, I. Foster, C. Kesselman, S. Tuecke, J. Volmer, and V. Welch. A national-scale authentication infrastructure. *IEEE Computer*, 33(12):60–66, 2000.

[EGEN09] H El Ganainy and MH El Naggar. Seismic performance of three-dimensional frame structures with underground stories. *Soil Dynamics and Earthquake Engineering*, 29(9):1249–1261, 2009.

[EH93] M. D. Engelhardt and A. S. Husain. Cyclic-loading performance of welded flange-web connections. *ASCE Journal of Structural Engineering*, 119(12):3537–3550, December 1993.

[EH04] Ahmed Elgamal and Liangcai He. Vertical earthquake ground motion records: An overview. *Journal of Earthquake Engineering*, 8(5):663–697, 2004.

[EH14] M Esmaeilbeigi and M.M. Hosseini. A new approach based on genetic algorithm for finding a good shape parameter in solving partial differential equations by kansa's method. *Applied Mathematics and Computation*, 249(0):419–428, 2014.

[EH19] Martin Roberts Enders and Nadja Hofback. Dimensions of digital twin applications - a literature review. In *Twenty-fifth Americas Conference on Information Systems*, pages 1–10, Cancun, 2019.

[EHN07] I. Einav, G.T. Houlsby, and G.D. Nguyen. Coupled damage and plasticity models derived from energy and dissipation potentials. *International Journal of Solids and Structures*, 44(7-8):2487–2508, April 2007.

[EHS05] David S. Evans, Andrei Hagiu, and Richard Schmalensee. A survey of the economic role of software platforms in computer–based industries. *CESifo Economic Studies, Vol. 51, No. 2-3, 2005*, pages 1–37, 2005.

[EHS06] David S. Evans, Andrei Hagiu, and Richard Schmalensee. *Invisible Engines: How Software Platforms Drive Innovation and Transform Industries*. The MIT Press, 2006. ISBN-13: 978-0262550680.

[EI65] A. Cemal Eringen and John D. Ingram. A continuum theory of chemically reacting media – I. *International Journal of Engineering Science*, 3:197–212, 1965.

[EIJ24] Sara Grbčić Erdelj, Adnan Ibrahimbegović, and Gordan Jelenić. Incompatible-mode geometrically nonlinear finite element for micropolar elasticity. *International Journal of Solids and Structures*, 289:112647, 2024.

[Ein12] Itai Einav. The unification of hypo-plastic and elasto-plastic theories. *International Journal of Solids and Structures*, 49(11-12):1305–1315, June 2012.

[EKC+15] Benjamin Edwards, Olga-Joan Ktenidou, Fabrice Cotton, Norman Abrahamson, Chris Van Houtte, and Donat Fäh. Epistemic uncertainty and limitations of the $\kappa 0$ model for near-surface attenuation at hard rock sites. *Geophysical Journal International*, 202(3):1627–1645, 2015.

[EKK05] I. Etsion, Y. Kligerman, and Y. Kadin. Unloading of an elastic–plastic loaded spherical contact. *International Journal of Solids and Structures*, 42(13):3716–3729, 2005.

[EKT14] Mojtaba E-Kan and Hossein A. Taiebat. On implementation of bounding surface plasticity models with no overshooting effect in solving boundary value problems. *Computers and Geotechnics*, 55(0):103 – 116, 2014.

[EL95] Howard C. Elman and Dennis K.–Y. Lee. Use of linear algebra kernels to build and efficient finite element solver. *Parallel Computing*, 21:161–173, 1995.

[ELF09] Ahmed Elgamal, Jinchi Lu, and Davide Forcellini. Mitigation of liquefaction-induced lateral deformation in a sloping stratum: Three-dimensional numerical simulation. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 135(11):1672–1682, November 2009.

[EML15] Laura Eads, Eduardo Miranda, and Dimitrios G Lignos. Average spectral acceleration as an intensity measure for collapse risk assessment. *Earthquake Engineering & Structural Dynamics*, 44(12):2057–2073, 2015.

[EMPT11] Howard C Elman, Christopher W Miller, Eric T Phipps, and Raymond S Tuminaro. Assessment of collocation and Galerkin approaches to linear diffusion equations with random data. *International Journal for Uncertainty Quantification*, 1(1), 2011.

[EPF21] Angelo Di Egidio, Stefano Pagliaro, and Cristiano Fabrizio. Combined use of rocking walls and inerters to improvethe seismic response of frame structuresl. *ASCE Journal of Engineering Mechanics*, 147(5), 2021.

[EPM20] Mohamed A. ElGhoraiby, Hanna Park, and Majid T. Manzari. Stress-strain behavior and liquefaction strength characteristics of Ottawa F65 sand. *Soil Dynamics and Earthquake Engineering*, 138:106292, 2020.

[ER94] R. A. Eve and B. D. Reddy. The variational formulation and solution of problems of finite–strain elastoplasticity based on the use of dissipation function. *International Journal for Numerical Methods in Engineering*, 37:1673–1695, 1994.

[Eri65] A Cemal Eringen. Theory of micropolar fluids. Technical report, DTIC Document, 1965.

[Eri88] Anders Eriksson. On some path–related measures for non–linear structural f. e. problems. *International Journal for Numerical Methods in Engineering*, 26:1791–1803, 1988.

[Eri89] Anders Eriksson. On linear constraints for Newton–Raphson corrections and critical point searches in structural f. e. problems. *International Journal for Numerical Methods in Engineering*, 28:1317–1334, 1989.

[Eri12] A Cemal Eringen. *Microcontinuum field theories: I. Foundations and solids*. Springer Science & Business Media, 2012.

[ES90] Margaret A. Ellis and Bjarne Stroustrup. *The Annotated C++ Reference Manual*. AT&T Bell Laboratories, Murray Hill, New Jersey and Addison - Wesley Publishing Company, 1990.

[ES01] E.A. Ellis and S.M. Springman. Modelling of soil-structure interaction for a piled bridge abutment in plane strain fem analyses. *Computers and Geotechnics*, 28:79–98, 2001.

[ES23] Fady A. Elshazly and Elnaz Seylabi. On seismic isolation of soil-meta-foundation-structure systems. *Computers and Geotechnics*, 161:105561, 2023.

[Esh57] J. D. Eshelby. The determination of the elastic field of an ellipsoidal inclusion, and related problems. *Proceedings of the Royal Society London*, A:376–296, 1957.

[ESI$^+$98] Masayoshi Esashi, Susumu Sugiyama, Kyoichi Ikeda, Yuelin Wang, and Haruzo Miyashita. Vacuum–sealed silicon micromachined pressure sensors. In Kensall D. Wise, editor, *Proceedings or the IEEE, Special Issue: Integrated Sensors, Microactuators & Microsystems (MEMS)*, pages 1627–1639, August 1998.

[Ets92] Guillermo Etse. *Theoretische un numerische Untersuchung zum diffusen und lokalisiert Versagen in Beton*. PhD thesis, Universtity of Karlsruhe, 1992. in German.

[Ets10] Izhak Etsion. Revisiting the Cattaneo–Mindlin concept of interfacial slip in tangentially loaded compliant bodies. *Journal of Tribology*, 132(2):020801, 2010.

[EU14] Roland Ewald and Adelinde M. Uhrmacher. Sessl: A domain-specific language for simulation experiments. *ACM Trans. Model. Comput. Simul.*, 24(2):11:1–11:25, feb 2014.

[EW93] Guillermo Etse and Kaspar Willam. A fracture energy – based constitutive formulation for inelastic behavior of plain concrete. Technical Report CU/SR-93/13, Universtity of Colorado, Department of Civil, Environmental & Architectural Engineering, December 1993.

[EWN05] Magnus Ekman, Fredrik Warg, and Jim Nilsson. An in-depth look at computer performance growth. *ACM SIGARCH Computer Architecture News*, 33(1):144–147, March 2005. ISSN:0163-5964.

[Eyh97] Dominique Eyheramendy. *Object–Oriented Finite Element Programming: symbolic Derivations and Automatic Programming*. PhD thesis, Ecole Polytechniques Fédéral de Lausanne, 1997.

[EYP02] Ahmed Elgamal, Zhaohui Yang, and Ender Parra. Computational modeling of cyclic mobility and post–liquefaction site response. *Soil Dynamics and Earthquake Engineering*, 22:259–271, 2002.

[EYYC08] Ahmed Elgamal, Linjun Yan, Zhaohui Yang, and Joel P. Conte. Three-dimensional seismic response of humboldt bay bridge-foundation-ground system. *ASCE Journal of Structural Engineering*, 134(7):1165–1176, July 2008.

[EYZP95] G. W. Ellis, C. Yao, R. Zhao, and D. Penumadu. Stress–strain modeling of sands using artificial neural networks. *ASCE Journal of Geotechnical Engineering*, 121(5):429–435, 1995.

[EZ96] D. Eyheramendy and Th. Zimmermann. Object–oriented finite elements II. a symbolic environment for automatic programming. *Computer Methods in Applied Mechanics and Engineering*, 132:277–304, 1996.

[EZ01] D. Eyheramendy and Th. Zimmermann. Object-oriented finite elements. iv. symbolic derivations and automatic programming of nonlinear formulations. *Computer Methods in Applied Mechanics and Engineering*, 190(22-23):2729–2751, 2001.

[FA92] Carlos A. Felippa and Scott Alexander. Membrane triangles with corner drilling freedoms – III. implementation and performance evaluation. *Finite Elements in Analysis and Design*, 12:203–239, 1992.

[Fai03] C. Fairhurst. Stress estimation in rock: a brief history and review. *International Journal of Rock Mechanics and Mining Sciences*, 40(7-8):957–973, October – December 2003.

[Faj18] Peter Fajfar. Analysis in seismic provisions for buildings: past, present and future. In *European Conference on Earthquake Engineering Thessaloniki, Greece*, pages 1–49. Springer, 2018.

[Fak] Fakespace, Inc. Immersive WorkBench. `http://www.fakescape.com`

[Fak96] Kazem Fakharian. *Three-dimensional monotonic and cyclic behaviour of sand-steel interfaces: Testing and modelling.* University of Ottawa (Canada), 1996.

[Far88] Charbel Farhat. A simple and efficient automatic FEM domain decomposer. *Computers & Structures*, 28(5):579–602, 1988.

[Far89] C. Farhat. Which parallel finite element algorithm for which architecture and which problems? In R.V. Grandhi, W.J. Stroud, and V.B. Venkayya, editors, *Proceedings of the Winter Annual Meeting of the American Society of Mechanical Engineers*, volume 16 of *AD*, pages 35–43. New York, N.Y. : American Society of Mechanical Engineers, 1989.

[Far18] Michael N. Fardis. Capacity design: Early history. *Earthquake Engineering & Structural Dynamics*, 10.1002/eqe.3110:1–10, 2018.

[Far24] Michael N. Fardis. Model- versus data-uncertainty for concrete members and connections in cyclic loading. *Earthquake Engineering & Structural Dynamics*, n/a(n/a), 2024.

[Fas16] Marco Fasan. *ADVANCED SEISMOLOGICAL AND ENGINEERING ANALYSIS FOR STRUCTURAL SEISMIC DESIGN*. PhD thesis, UNIVERSITÀ DEGLI STUDI DI TRIESTE, 2016.

[FAY^{+}24] Yuan Feng, José Antonio Abell, Han Yang, Hexiang Want, and Boris Jeremić. Domain specific language for finite element modeling and simulation. *Advances in Engineering Software*, 193(103666):1–17, 2024.

[FB87] C. A. Felippa and P. G. Bergan. A triangular bending element based of an energy-orthogonal free formualtion. *Computer Methods in Applied Mechanics and Engineering*, 61:129–160, 1987.

[FB96] R. M. Felder and R. Brent. Navigating the bumpy road to student–centered instructions. *College Teaching*, 44(2):43–47, 1996.

[FB99] Richard Felder and Rebecca Brent. ExCEEd effective college teaching seminar: Course material. 1999 Civil Engineering Conference and Exposition, Charlotte, North Carolina, October 16-17 1999.

[FB05] C.A. Felippa and B.Haugen. A unified formulation of small-strain corotational finite elements:i.theory. *Computer Methods in Applied Mechanics and Engineering*, 194:2285 2335, 2005.

[FB09] S Frydman and R Baker. Theoretical soil-water characteristic curves based on adsorption, cavitation, and a double porosity model. *International journal of geomechanics*, 9(6):250–257, 2009.

[FBCP97] R. Foerch, J. Besson, G Cailletaud, and P. Pilvin. Polymorphic constitutive equations in finite element codes. *Computer methods in applied mechanics and engineering*, 141:355–372, 1997.

[FBD97] George M. Filz, Ronald D. Boyer, and Richard R. Davidson. Bentonite–water slurry rheology and cutoff wall trench stability. In *Proceedings of the ASCE Conference on in Situ Remediation of the Geoenvironment*, 1997.

[FBM06] M. Fortin F. Brezzi and L.D. Marini. Error analysis of piecewise constant pressure approximations of darcy's law. *Computer Methods in Applied Mechanics and Engineering*, 195(13-16):1547–1559, February 2006.

[FBP83] Filip C Filippou, Vitelmo Victorio Bertero, and Egor P Popov. Effects of bond deterioration on hysteretic behavior of reinforced concrete joints. Technical report, Earthquake Engineering Research Center, University of California, Berkeley, 1983.

[FC84] Gregory Fenves and Anil K. Chopra. EAGD-84 a computer program for earthquake analysis concrete gravity dams. Technical Report 84/11, EERC/UCB, August 1984.

[FC94] Charbel Farhat and Luis Crivelli. A transient FETI methodology for large–scale parallel implicit comutations in structural mechanics. *International Journal for Numerical Methods in Engineering*, 37:1945–1975, 1994.

[FC10] Fabio Fadi and Michael C. Constantinou. Evaluation of simplified methods of analysis for structures with triple friction pendulum isolators. *Earthquake Engineering & Structural Dynamics*, 39(1):5–22, January 2010.

[FC14] A Fernando Concha. *Solid-liquid separation in the mining industry*. Springer, 2014.

[FCDL^{+}21] Gabriele Fiorentino, Cihan Cengiz, Flavia De Luca, George Mylonakis, Dimitris Karamitros, Matt Dietz, Luiza Dihoru, Davide Lavorato, Bruno Briseghella, Tatjana Isaković, Christos Vrettos, Antonio Topa Gomes, Anastasios Sextos, and Camillo Nuti. Integral abutment bridges: Investigation of seismic soil-structure interaction effects by shaking table testing. *Earthquake Engineering & Structural Dynamics*, 50(6):1517–1538, 2021.

[FCMT23] De-Cheng Feng, Xin Chen, Frank McKenna, and Ertugrul Taciroglu. Consistent nonlocal integral and gradient formulations for force-based Timoshenko elements with material and geometric nonlinearities. *Journal of Structural Engineering*, 149(4):04023018, 2023.

[Fd95] P. H. Feenstra and R. de Borst. A constitutive model for reinforced concrete based on stress decomposition. In Stein Sture, editor, *Proceedings of 10th Conference*, pages 643–646. Engineering Mechanics Division of the American Society of Civil Engineers, May 1995.

[FD97] George M. Filz and J. Michael Duncan. Vertical shear loads on nonmoving walls. I: Theory. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 123(9):856–862, September 1997.

[FD05] Gregory Fenves and Mathew Dryden. Nees sfsi demonstration project. NEES project meeting, TX, Austin, August 2005.

[FD07] Heidi P. Feigenbaum and Yannis F. Dafalias. Directional distortional hardening in metal plasticity within thermodynamics. *International Journal of Solids and Structures*, 44(22-23):7526–7542, 2007.

[FD10] Pengcheng Fu and Yannis Dafalias. Fabric evolution within shear bands of granular materials and its relation to critical state theory. *INTERNATIONAL JOURNAL FOR NUMERICAL AND ANALYTICAL METHODS IN GEOMECHANICS*, Early View DOI: 10.1002/nag.988, 2010.

[FD11] Pengcheng Fu and Yannis F. Dafalias. Study of anisotropic shear strength of granular materials using dem simulation. *International Journal for Numerical and Analytical Methods in Geomechanics*, 35(10):1098–1126, 2011.

[FDE97] George M. Filz, J. Michael Duncan, and Robert M. Ebeling. Vertical chear loads on nonmoving walls. II: Applications. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 123(9):863–873, September 1997.

[FDRDL^{+}24a] Gabriele Fiorentino, Raffaele De Risi, Flavia De Luca, George Mylonakis, Bruno Briseghella, Camillo Nuti, and Anastasios Sextos. Ssi-induced seismic earth pressures on an integral abutment bridge model: Experimental measurements versus numerical simulations and code provisions. *Earthquake Engineering & Structural Dynamics*, n/a(n/a), 2024.

[FDRDL^{+}24b] Gabriele Fiorentino, Raffaele De Risi, Flavia De Luca, George Mylonakis, Bruno Briseghella, Camillo Nuti, and Anastasios Sextos. SSI-induced seismic earth pressures on an integral abutment bridge model: Experimental measurements versus numerical simulations and code provisions. *Earthquake Engineering & Structural Dynamics*, 53(15):4830–4852, 2024.

[FDTHS10] Fabio Ferri, Giulio Di Toro, Takehiro Hirose, and Toshihiko Shimamoto. Evidence of thermal pressurization in high-velocity friction experiments on smectite-rich gouges. *Terra Nova*, 22(5):347–353, 2010.

[FDWC02] B. Fornberg, T.A. Driscoll, G. Wright, and R. Charles. Observations on the behavior of radial basis function approximations near boundaries. *Computers & Mathematics with Applications*, 43(3-5):473 – 490, 2002.

[FE95] K Fakharian and E Evgin. Simple shear versus direct shear tests on interfaces during cyclic loading. In *Proceedings of International Conference on Recent Advances in geotechnical Earthquake Engineering and Soil Dynamics*, volume 1. University of Missouri–Rolla, 1995.

[FE96] KAZEM Fakharian and E Evgin. An automated apparatus for three-dimensional monotonic and cyclic testing of interfaces. *Geotechnical Testing Journal*, 19(1):22–31, 1996.

[FE97] Kazem Fakharian and Erman Evgin. Cyclic simple-shear behavior of sand-steel interfaces under constant normal stiffness condition. *Journal of Geotechnical and Geoenvironmental Engineering*, 123(12):1096–1105, 1997.

[FE98] G.L. Fenves and M. Ellery. Behavior and failure analysis of a multiple–frame highway bridge in the 1994 northridge earthquake. Technical Report 98/08, PEER Center, December 1998.

[FE00] Kazem Fakharian and Erman Evgin. Elasto-plastic modelling of stress-path-dependent behaviour of interfaces. *International Journal for Numerical and Analytical Methods in Geomechanics*, 24(2):183–199, 2000.

[FE$^+$02] Kazem Fakharian, Erman Evgin, et al. A comprehensive experimental study of sand-steel interfaces subjected to various monotonic and cyclic stress paths. In *The Twelfth International Offshore and Polar Engineering Conference*. International Society of Offshore and Polar Engineers, 2002.

[FEB95] Werner Fuchs, Rolf Eligehausen, and John E. Breen. Concrete capacity design (CCD) approach for fastening to concrete. *ACI Journal*, 92(1):73–94, January/Febryary 1995.

[Fee93] Peter H. Feenstra. *Computational aspects of biaxial stress in plain and reinforced concrete*. PhD thesis, Delft University of Technology, November 1993.

[FEG$^+$24] Tony Fierro, Stefano Ercolessi, Davide Noè Gorini, Giovanni Fabbrocino, and Filippo Santucci de Magistris. Implementation of an advanced bounding surface constitutive model in opensees. *Computers and Geotechnics*, 166:106030, 2024.

[Fel77a] Carlos A. Felippa. Error analysis of penalty function techniques for constraint definition in linear algebraic systems. *International Journal for Numerical Methods in Engineering*, 11:709–728, 1977.

[Fel77b] Carlos A. Felippa. Error analysis of penalty functions techniques for constraint definition in linear algebraic systems. *International Journal for Numerical Methods in Engineering*, 11:709–728, 1977.

[Fel81] Carlos A. Felippa. Architecture of a distributed analysis network for computational mechanics. *Computers & Structures*, 13:405–413, 1981.

[Fel84] Carlos A. Felippa. Dynamic relaxation under general incremental control. In W. K. Liu, T. Belytschko, and K. C. Park, editors, *Innovative Methods for Nonlinear Problems*, pages 103–133. Pineridge Press, Swansea U.K., 1984.

[Fel87a] Richard M. Felder. On creating creative engineers. *Engineering Education*, pages 222–227, January 1987.

[Fel87b] Carlos A. Felippa. Traversing critical points with penalty springs. Technical Report CU–CSSC–87–02, University of Colorado at Boulder, 1987.

[Fel89a] Carlos A. Felippa. Introduction to linear finite element methods, lecture notes, i and ii. Technical report, University of Colorado at Boulder, 1989. Report No. CU-CSSC-89-19 Septembar 1989.

[Fel89b] Carlos A. Felippa. Parametrized multifield variational principles in elasticity: II. hybrid functionals and the free formulation. *Communications in Applied Numerical Methods*, 5:89–98, 1989.

[Fel89c] Carlos A. Felippa. Parametrized mutlifield variational principles in elasticity: I. mixed functionals. *Comunications in Applied Numerical Methods*, 5:79–88, 1989.

[Fel92a] Carlos A. Felippa. Object oriented finite element programing. Lecture Notes at CU Boulder, august - decembar 1992.

[Fel92b] Carlos A. Felippa. Object oriented finite element programing. Lecture Notes at CU Boulder, august - decembar 1992.

[Fel93] Carlos A. Felippa. Nonlinear finite element methods. Lecture Notes at CU Boulder, 1993.

[Fel94] Carlos A. Felippa. A survey of parametrized variational principles and applications to computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 113:109–139, 1994.

[Fel95] R. M Felder. A longitudinal study of engineering student performance and retention. IV instructional methods. *Journal of Engineering Education*, 82(1):361–367, 1995.

[Fel06] Carlos A. Felippa. Supernatural QUAD4: A template formulation. *Computer Methods in Applied Mechanics and Engineering*, 195(41-43):5316–5342, August 2006.

[Fel12] Carlos A. Felippa. CIMNE and the butterfly effect. Technical report, University of Colorado at Boulder, July, November 2012.

[FEM00] FEMA-365. Prestandard and commentary for the seismic rehabilitation of buildings. Technical report, Federal Emergency Management Agency, Washington DC., 2000.

[Fen90] Gregory L. Fenves. Object –oriented programming for engineering software development. *Engineering with Computers*, 6:1–15, 1990.

[Fen99a] Gordon A. Fenton. Estimation for stochastic soil models. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 125(6):470–485, June 1999.

[Fen99b] Gordon A. Fenton. Random field modeling of cpt data. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 125(6):486–498, June 1999.

[Fen09] Gregory L. Fenves. Editorial for the special issue. *Earthquake Engineering and Structural Dynamics*, 38(5):537–539, 2009.

[Fer92] Pedro Arduino Ferrer. Elastoplatic characterization of granular materials. Master's thesis, University of Puerto Rico, 1992.

[FFBW$^+$93] R. M Felder, K. D. Forrest, L. Baker-Ward, E. J. Dietz, and P. H. Mohr. A longitudinal study of engineering student performance and retention. I success and failure of introductory courses. *Journal of Engineering Education*, 82(1):15–21, 1993.

[FFD98] R.M. Felder, G.N. Felder, and E.J. Dietz. A longitudinal study of engineering student performance and retention. V. comparisons with traditionally-taught students. *Journal of Engineering Education*, 87(4):469–480, 1998.

[FFK02] A.I. Fedoseyev, M.J. Friedman, and E. Kansa. Improved multiquadric method for elliptic partial differential equations via pde collocation on the boundary. *Comput. Math. Appl.*, 43(3-5):439–355, 2002.

[FFM$^+$95] R. M Felder, G. N. Felder, M. Mauney, C. E. Hamrin Jr., and E. J. Dietz. A longitudinal study of engineering student performance and retention. III gender differences in student performance and atitudes. *Journal of Engineering Education*, 84(2):151–163, 1995.

[FFS90] Bruce W. R. Forde, Ricardo O. Foschi, and Siegfried F. Steimer. Object – oriented finite element analysis. *Computers and Structures*, 34(3):355–374, 1990.

[FFZ$^+$20] S. Fiore, G. Finocchio, R. Zivieri, M. Chiappini, and Garesc?F. Wave amplitude decay driven by anharmonic potential in nonlinear mass-in-mass systems. *Applied Physics Letters*, 117(12):124101, 2020.

[FG96] Peter Fajfar and Peter Gašperšič. The N2 method for the seismic damage analysis of rc buildings. *Earthquake Engineering & Structural Dynamics*, 25(1):31–46, 1996.

[FG02] Gordon A. Fenton and D. V. Griffiths. Probabilistic foundation settlement on spatially random soil. *Journal of Geotechnical and Geoenvironmental Engineering, ASCE*, 128(5):381–390, May 2002.

[FG03] G.A. Fenton and D.V. Griffiths. Bearing-capacity prediction of spatially random $c--\phi$ soils. *Canadian Geotechnical Journal*, 40(1):54–65, 2003.

[FG04] R.V. Field and M. Grigoriu. On the accuracy of the polynomial chaos approximation. *Probabilistic Engineering Mechanics*, 19(1-2):65–80, 2004.

[FG05] Gordon A. Fenton and D. V. Griffiths. Three-dimensional probabilistic foundation settlement. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 131(2):232–239, 2005.

[FG08] G.A. Fenton and D.V. Griffiths. *Risk Assessment in Geotechnical Engineering*. John Wiley & Sons, 2008.

[FGH82] Roger Frank, A Guenot, and P Humbert. Numerical analysis of contacts in geomechanics. In *4th Int. Conf. Num. in Geomechs*, volume 1, pages 37–45, 1982.

[FGH$^+$19] Anca C. Ferche, Bishnu Gautam, Farhad Habibi, Daman K. Panesar, Shamim A. Sheikh, Frank J. Vecchio, and Nebojsa Orbovic. Material, structural and modelling aspects of alkali aggregate reaction in concrete. *Nuclear Engineering and Design*, 351:87–93, 2019.

[FH95] R.M. Felder and E.R. Henriques. Learning and teaching styles in foreign and second language education. *Foreign Language Annals*, 28(1):21–31, 1995.

[FHF$^+$20] M. Fadaee, K. Hashemi, F. Farzaneganpour, I. Anastasopoulos, and G. Gazetas. 3-storey building subjected to reverse faulting: Analysis and experiments. *Soil Dynamics and Earthquake Engineering*, 138:106297, 2020.

[FHM95] Carlos A. Felippa, Bjørn Haugen, and Carmelo Militello. From the individual element test to finite element templates: Evolution of the patch test. *International Journal for Numerical Methods in Engineering*, 38:199–229, 1995.

[Fil96] George M. Filz. Consolidation stresses in soil–bentonite backfilled trenches. In Masashi kamon, editor, *Environmental Geotechnics*, pages 497–502. Balkema, 1996.

[Fin98] W. D. Liam Finn. Seismic safety of embankment dams developments in research and practice 1988-1998. In Panos Dakoulas, Mishac Yegian, and Robert D. Holtz, editors, *Proceedings of a Specialty Conference: Geotechnical Earthwuake Engineering and Soil Dynamics III*, Geotechnical Special Publication No. 75, pages 812–853. ASCE, August 1998. 1998.

[Fin00] W. D. L. Finn. State–of–the–art of geotechnical earthquake engineering practice. *Soil Dynamics and Earthquake Engineering*, 20:1–15, 2000.

[Fin18] W.D. Liam Finn. Performance based design in geotechnical earthquake engineering. *Soil Dynamics and Earthquake Engineering*, 114:326 – 332, 2018.

[FJBZ97] Edward H Field, Paul A Johnson, Igor A Beresnev, and Yuehua Zeng. Nonlinear ground-motion amplification by sediments during the 1994 northridge earthquake. *Nature*, 390(6660):599, 1997.

[FJC03] Edward H Field, Thomas H Jordan, and C Allin Cornell. OpenSHA: A developing community-modeling environment for seismic hazard analysis. *Seismological Research Letters*, 74(4):406–419, 2003.

[FJP11] Massimiliano Ferronato, Carlo Janna, and Giorgio Pini. Parallel solution to ill-conditioned fe geomechanical problems. *International Journal for Numerical and Analytical Methods in Geomechanics*, pages n/a–n/a, 2011.

[FJP$^+$17] Edward H. Field, Thomas H. Jordan, Morgan T. Page, Kevin R. Milner, Bruce E. Shaw, Timothy E. Dawson, Glenn P. Biasi, Tom Parsons, Jeanne L. Hardebeck, Andrew J. Michael, II Ray J. Weldon, Peter M. Powers, Kaj M. Johnson, Yuehua Zeng, Karen R. Felzer, Nicholas van der Elst, Christopher Madden, Ramon Arrowsmith, Maximilian J. Werner, and Wayne R. Thatcher. A synoptic view of the third Uniform California Earthquake Rupture Forecast (UCERF3). *Seismological Research Letters*, 88(5):1259–1267, 2017.

[FK] Ian Foster and Carl Kesselman. The globus project. `http://www.globus.org/`.

[FK95] C. Ben Farros and Richard E. Klingner. Tensile capacity of anchors with partial or overlapping failure surfaces: Evaluation of existing formulae on an LRFD basis. *ACI Journal*, 92(6):698–710, November/December 1995.

[FK99] Ian Foster and Carl Kesselman, editors. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers, Inc., San Francisco, California, 1999.

[FKKY08] V.R. Feldgun, A.V. Kochetkov, Y.S. Karinski, and D.Z. Yankelevsky. Blast response of a lined cavity in a porous saturated soil. *International Journal of Impact Engineering*, 35(9):953–966, 2008.

[FKNT02] I. Foster, C. Kesselman, J. Nick, and S. Tuecke. Grid services for distributed system integration. *IEEE Computer*, 35(6):37–46, June 2002.

[FKT01] Ian Foster, Carl Kesselman, and Steven Tuecke. The anatomy of the grid. *International Journal for Suprcomputing Applications*, 2001. `http://www.globus.org/research/papers.html`.

[FKTT98] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke. A security architecture for computational grids. In *Proc. 5th ACM Conference on Computer and Communications Security Conference,*, pages 83–92, 1998.

[FKYK16] V.R. Feldgun, Y.S. Karinski, D.Z. Yankelevsky, and A.V. Kochetkov. A new analytical approach to reconstruct the acceleration time history at the bedrock base from the free surface signal records. *Soil Dynamics and Earthquake Engineering*, 85:19 – 30, 2016.

[FL93] Charbel Farhat and Michel Lesoinne. Automatic partitioning of unstructured meshes for the parallel solution of problems in computational mechanics. *International Journal for Numerical Methods in Engineering*, 36:745–764, 1993.

[FLB86] W.D.L. Finn, R.H. Ledbetter, and L.L. Beratan. Seismic soil-structure interaction: Analysis and centrifuge model studies. *Nuclear Engineering and Design*, 94(1):53 – 66, 1986.

[Fle87] R. Fletcher. *Practical Methods of Optimization*. John Wiley and Sons., 1987.

[FLF11] B. Fornberg, E. Larsson, and N. Flyer. Stable computations with gaussian radial basis functions. *SIAM J. Sci. Comput.*, 33(2):869–892, 2011.

[FLH17] Jia Fu, Jianwen Liang, and Bin Han. Impedance functions of three-dimensional rectangular foundations embedded in multi-layered half-space. *Soil Dynamics and Earthquake Engineering*, 103:118–122, 2017.

[FLO76] William N. Findley, James S. Lai, and Kasif Onaran. *Creep abd Relaxation of Viscoelastic Materials*. Series in Applied Mathematics and Mechanics. North–Holland Publishing Company, 1976.

[Flo86] Floating Point Systems Ltd. Notes on supercomputers. *Engineering Computations*, 3:254–258, 1986.

[FM77] D G Fredlund and N R Morgenstern. Stress state variables for unsaturated soils. *Journal of Geotechnical and Geoenvironmental Engineering*, 103, 1977.

[FM92] Carlos A. Felippa and Carmelo Militello. Membrane triangles with corner drilling freedoms – II. the ANDES element. *Finite Elements in Analysis and Design*, 12:189–201, 1992.

[FM99] Gregory Fenves and Francis McKenna. An object–oriented design for structural analyses. *to be submitted*, 1999.

[FM12] G. E. Fasshauer and M.J. McCourt. Stable evaluation of gaussian radial basis function interpolants. *SIAM J. Sci. Comput.*, 34(2):737–762, 2012.

[FMA$^+$16] Marco Fasan, Andrea Magrin, Claudio Amadio, Fabio Romanelli, Franco Vaccari, and Giuliano F. Panza. A seismological and engineering perspective on the 2016 Central Italy earthquakes. *Int. J. Earthquake and Impact Engineering*, 1(4):395–420, 2016.

[FMDBW94] R. M Felder, P. H Mohr, E. J. Dietz, and L. Baker-Ward. A longitudinal study of engineering student performance and retention. II differences between students from rural and urban backgrounds. *Journal of Engineering Education*, 83(3):15–21, 1994.

[FMdS04] J. Fortin, O. Millet, and G. de Saxcé. Numerical simulation of granular materials by an improved discrete element method. *International Journal for Numerical Methods in Engineering*, 62(5):639 – 663, 2004.

[FO07] Magnus Fredriksson and Niels Saabye Ottosen. Accurate eight-node hexahedral element. *International Journal for Numerical Methods in Engineering*, 72(6):631–657, 2007.

[FOC98] R. Faria, J. Oliver, and M. Cervera. A strain-based plastic viscous-damage model for massive concrete structures. *International Journal for Solids and Structures*, 35(14):1533–1558, 1998.

[FOT$^+$21] James W. Foulk? III, James T. Ostien, Brandon Talamini, Michael R. Tupek, Nathan K. Crane, Alejandro Mota, and Michael G. Veilleux. Extending a 10-node composite tetrahedral finite element for solid mechanics. *International Journal for Numerical Methods in Engineering*, 122(15):3845–3875, 2021.

[Fow05] Martin Fowler. Language workbenches: The killer-app for domain specific languages? published online `http://martinfowler.com/articles/languageWorkbench.html`, June 2005.

[Fow11] Martin Fowler. *Domain-Specific Languages*. Addison-Wesley Signature Series, 2011.

[Fox68] N. Fox. On the continuum theories of dislocations and plasticity. *Quarterly Journal of Mechanics and Applied Mathematics*, XXI(1):67–75, 1968.

[Foy95] H. C. Foyle. Interactive learning in the higher education classroom: Cooperative, collaborative, and active learning strategies. In Harvey C. Foyle, editor, _, page 237. National Education Association, Washington D.C., 1995.

[FP94] C. M. Famiglietti and J. H. Prevost. Solution of the slump test using a finite deformation elasto–plastic drucker–prager model. *International Journal for Numerical Methods in Engineering*, 37:3869–3903, 1994.

[FP95] Carlos A. Felippa and K.C. Park. Advanced finite element methods. Lecture Notes at CU Boulder, January-May 1995.

[FPF01] Carlos A. Felippa, K.C. Park, and Charbel Farhat. Partitioned analysis of coupled mechanical systems. *Computer Methods in Applied Mechanics and Engineering*, 190(24-25):3247 – 3270, 2001. Advances in Computational Methods for Fluid-Structure Interaction.

[FPF$^+$24] Salah A. Faroughi, Nikhil M. Pawar, Célio Fernandes, Maziar Raissi, Subasish Das, Nima K. Kalantari, and Seyed Kourosh Mahjour. Physics-guided, physics-informed, and physics-encoded neural networks and operators in scientific computing: Fluid and solid mechanics. *Journal of Computing and Information Science in Engineering*, 24(4):040802, 01 2024.

[FPK15] Arash Fathi, Babak Poursartip, and Loukas F. Kallivokas. Time-domain hybrid formulations for wave simulations in three-dimensional pml-truncated heterogeneous media. *International Journal for Numerical Methods in Engineering*, 101(3):165–198, 2015.

[FPO96] Y. T. Feng, D. Perić, and D. R. J. Owen. A new criterion for determination of initial loading parameter in arc–length methods. *Computers and Structures*, 58(3):479–485, 1996.

[FR64] Reeves Fletcher and Colin M Reeves. Function minimization by conjugate gradients. *The Computer Journal*, 7(2):149–154, 1964.

[FR91] Charbel Farhat and Francois–Xavier Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineers*, 32:1205–1227, 1991.

[FR93] D G Fredlund and H Rahardjo. *Soil mechanics for unsaturated soils*. John Wiley & Sons, 1993.

[Fra04] Paolo Franchin. Reliability of uncertain inelastic structures under earthquake excitation. *ASCE Journal of Engineering Mechanics*, 130(2):180–191, February 2004.

[Fre03] Roland W. Freund. Model reduction methods based on Krylov subspaces. *Acta Numerical*, pages 267–319, 2003.

[Fre07] Free Software Foundation, Inc. Gnu lesser general public license, ver. 3. published online: http://www.gnu.org/copyleft/lesser.html, 29th. June 2007.

[Fre10] Margus Freudenthal. Domain-specific languages in a customs information system. *IEEE Software*, 27:65–71, 2010.

[FRF12] D G Fredlund, H Rahardjo, and M D Fredlund. *Unsaturated soil mechanics in engineering practice*. John Wiley & Sons, 2012.

[FRF$^+$22] Giuseppe Andrea Ferro, Luciana Restuccia, Devid Falliano, Achille Devitofranceschi, and Angelo Gemelli. Collapse of existing bridges: From the lesson of La Reale viaduct to the definition of a partial safety coefficient of variable traffic loads. *Journal of Structural Engineering*, 148(11):04022181, 2022.

[Fri00] Peter Fritz. *RHEO-STAUB, User Manual*. Eidgenössische Technische Hochschule Zürich, Swiss Federal Institute of Technology in Zurich, 2000.

[FS85] C. A. Felippa and G. M. Stanley. NICE: A utility architecture for computational mechanics. In Wunderlich Bergan, Bathe, editor, *Finite Element Methods for Nonlinear Problems, Europe–US Symposium, Trondheim, Norway*, pages 447–463, 1985.

[FS87] Bruce W. R. Forde and Siegfried F. Stiemer. Improved arc length orthogonality methods for nonlinear finite element analysis. *Computers & Structures*, 27(5):625–630, 1987.

[FS96] C. Freischläger and K. Schweizerhof. On a systematic development of trilinear three–dimensional solid elements bsed on Simo's enhanced strain formulation. *International Journal of Solids and Structures*, 33(20-22):2993–3017, 1996.

[FST05] Philipp Frauenfelder, Christoph Schwab, and Radu Alexandru Todor. Finite elements for elliptic problems with stochastic coefficients. *Computer Methods in Applied Mechanics and Engineering*, 194(2-5):205–228, February 2005.

[FT25] WS Farren and GI Taylor. The heat developed during plastic extension of metals. *Proceedings of the royal society of London A: mathematical, physical and engineering sciences*, 107(743):422–451, 1925.

[FT13] Behzad Fatahi and S Hamid Reza Tabatabaiefar. Fully nonlinear versus equivalent linear computation method for seismic analysis of midrise buildings on soft soils. *International Journal of Geomechanics*, 14(4):04014016, 2013.

[FTBK90] I. K. Fang, C. K. T. Tsui, N. H. Burns, and R. E. Klingner. Fatigue behavior of cast–in–place and precast panel bridge decks with isotropic reinforcement. *PCI Journal*, 35(3):28–39, May–June 1990.

[FTC$^+$00] Issai Fujishiro, Yuriko Takeshima, Li Chen, Hiroko Nakamura, and Yasuko Suzuki. Parallel visualization of gigabyte datasets in geofem. In *2nd ACES Workshop Extended Abstracts*. APEC Cooperation for Earthquake Simulation (ACES), `http://www.tokyo.rist.or.jp/ACES_WS2/`, 2000.

[FTV06] F. Froiio, G. Tomassetti, and I. Vardoulakis. Mechanics of granular materials: The discrete and the continuum descriptions juxtaposed. *International Journal of Solids and Structures*, 43(25-26):7684–7720, December 2006.

[FW87] Charbel Farhat and Edward Wilson. A new finite element concurrent computer program architecture. *International Journal for Numerical Methods in Engineering*, 24:1771–1792, 1987.

[FW88] Charbel Farhat and Edward Wilson. A parallel active column equation solver. *Computers & Structures*, 28(2):289–304, 1988.

[FW20] Cheng Fang and Wei Wang. *Shape Memory Allows for Seismic Resilience*. Springer, 2020.

[FWPN24] Mojtaba Farahnak, Richard Wan, Mehdi Pouragha, and François Nicot. A multiscale bifurcation analysis using micromechanical-based constitutive tensor for granular material. *International Journal of Solids and Structures*, 298:112866, 2024.

[FWSR00] R.M. Felder, D.R. Woods, J.E. Stice, and A. Rugarcia. The future of engineering education. ii. teaching methods that work. *Chemical Engineering Education*, 34(1):26–39, 2000.

[FWYJ24] Yuan Feng, Hexiang Wang, Han Yang, and Boris Jeremić. Hardware aware plastic domain decomposition for nonlinear finite element simulation. *Advances in Engineering Software*, to be submitted, 2024.

[FXH94] D G Fredlund, A Xing, and S Huang. Predicting the permeability function for unsaturated soils using the soil-water characteristic curve. *Canadian Geotechnical Journal*, 31(4):533–546, 1994.

[FZ75] A Francavilla and OC Zienkiewicz. A note on numerical computation of elastic contact problems. *International Journal for Numerical Methods in Engineering*, 9(4):913–924, 1975.

[FZ07a] G.E. Fasshauer and J.G. Zhang. Iterated approximate moving least squares approximation. *Advances in Meshfree Techniques*, pages 221–240, 2007.

[FZ07b] B. Fornberg and J. Zuev. The runge phenomenon and spatially variable shape parameters in {RBF} interpolation. *Computers & Mathematics with Applications*, 54(3):379 – 398, 2007.

[FZH18] Da-Kuo Feng, Jian-Min Zhang, and Wen-Jun Hou. Three-dimensional direct-shear behaviors of a gravel-structure interface. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 144(12):04018095–1–11, 2018.

[FZK21] Nan Feng, Guodong Zhang, and Kapil Khandelwal. On the performance evaluation of stochastic finite elements in linear and nonlinear problems. *Computers & Structures*, 243:106408, 2021.

[FZY$^+$19] Yuan Feng, Kaveh Zamani, Han Yang, Hexiang Wang, Fangbo Wang, and Boris Jeremić. Procedures to build trust in nonlinear elastoplastic integration algorithm: solution and code verification. *Engineering with Computers*, pages 1–14, 2019.

[GA77] Rex V. Gibbons and Thomas J. Ahrens. Effects of shock pressures on calcic plagioclase. *Physics and Chemistry of Minerals*, 1(1):95–107, 1977.

[GAA03] G. Gazetas, M Apostolou, and J. Anastopoulos. Seismic uplifting of foundations on soft soils, with examples from adapazari (izmit 1999 earthquake). In *Proceedings of the International Conference on Foundations*, Dundee, Sept. 2003.

[GAA$^+$14] Nick Gregor, Norman A Abrahamson, Gail M Atkinson, David M Boore, Yousef Bozorgnia, Kenneth W Campbell, Brian S-J Chiou, IM Idriss, Ronnie Kamai, Emel Seyhan, et al. Comparison of NGA-West2 GMPEs. *Earthquake Spectra*, 30(3):1179–1197, 2014.

[GAB$^+$80] Kurt H. Gerstle, Helmut Aschl, Roberto Bellotti, Paolo Bertacchi, Michale D. Kotsovos, Hon-Yim Ko, Diethlem Linse, John B. Newman, Pio Rosi, Gerald Schickert, Michale A. Taylor, Leonard A. Traina, Helmut Winkler, and Roger M. Zimmerman. Behaviour of concrete under multiaxial stress states. *Journal of Engineering Mechanics Division*, 106(EM6):1383–1403, December 1980.

[GAF09] D. Ghosh, P. Avery, and C. Farhat. A feti-preconditioned conjugate gradient method for large-scale stochastic finite element problems. *International Journal for Numerical Methods in Engineering*, 80(6-7):914–931, 2009.

[GAG20] Evangelia Garini, Ioannis Anastasopoulos, and George Gazetas. Soil, basin and soil-building-soil interaction effects on motions of mexico city during seven earthquakes. *Géotechnique*, 70(7):581–607, 2020.

[Gaj95] A. Gajo. Influence of viscous coupling in propagation of elastic waves in saturated soil. *ASCE Journal of Geotechnical Engineering*, 121(9):636–644, September 1995.

[Gaj10] A. Gajo. Hyperelastic modelling of small-strain stiffness anisotropy of cyclically loaded sand. *International Journal for Numerical and Analytical Methods in Geomechanics*, 34(2):111–134, 2010.

[GAJ$^+$18] S.F. Ghahari, F. Abazarsa, C. Jeong, A. Kurtulus, and E. Taciroglu. Blind identification of site effects and bedrock motion from surface response signals. *Soil Dynamics and Earthquake Engineering*, 107:322 – 331, 2018.

[Gan18] Buntara S Gan. *An Isogeometric Approach to Beam Structures*. Springer, 2018.

[Gar96] Igor A. Garagash. Localization of inelastic deformation in elasto–plastic pore solids saturated by liquid. In Y. K. Lin and T. C. Su, editors, *Proceedings of 11th Conference*, pages 931–934. Engineering Mechanics Division of the American Society of Civil Engineers, May 1996.

[GASW15] Christine A Goulet, Norman A Abrahamson, Paul G Somerville, and Katie E Wooddell. The SCEC broadband platform validation exercise: Methodology for code validation in the context of seismic-hazard analyses. *Seismological Research Letters*, 86(1):17–26, 2015.

[GB95] G. Gabriel and K. J. Bathe. Some computational issues in large strain elasto–plastic analysis. *Computers and Structures*, 56(2/3):249–267, 1995.

[GB99] M. A. Gutierrez and R. de Borst. Numerical analysis of localization using a viscoplastic regularization: Influence of stochastic material defects. *International Journal for Numerical Methods in Engineering*, 44:1823–1841, 1999.

[GBC$^+$12] Quan Gu, Michele Barbato, Joel P. Conte, Philip E. Gill, and Frank McKenna. OpenSees-SNOPT framework for finite-element-based optimization of structural and geotechnical systems. *ASCE Journal of Structural Engineering*, 138(6):822–834, June 2012.

[GBD$^+$] Al Geist, Adam Beguelin, Jack Dongarra, Qeicheng Jiang, Robert Manchek, and Vaidy Sunderam. The parallel virtual machine (pvm). set of documents on the WWW. `http://www.epm.ornl.gov/pvm/pvm_home.html`

[GBD16] Beliz U Gokkaya, Jack W Baker, and Greg G Deierlein. Quantifying the impacts of modeling uncertainties on the seismic drift demands and collapse risk of buildings with implications on seismic design checks. *Earthquake Engineering & Structural Dynamics*, 45(10):1661–1683, 2016.

[GBT06] Dominik Göddeke, Christian Becker, and Stefan Turek. Integrating GPUs as fast co-processors into the parallel FE package FEAST. In Markus Becker and Helena Szczerbicka, editors, *Proceedings of the 19th Symposium on Simulation Technique*, pages 277–282, Septembar 2006.

[GBvH13] P. Gousseau, B. Blocken, and G.J.F. van Heijst. Quality assessment of large-eddy simulation of wind flow around a high-rise building: Validation and solution verification. *Computers and Fluids*, 79:120 – 133, 2013.

[GBWT09] Dominik Göddeke, Sven H.M. Buijssen, Hilmar Wobker, and Stefan Turek. GPU acceleration of an unmodified parallel finite element Navier-Stokes solver. In Waleed W. Smari and John P. McIntire, editors, *High Performance Computing & Simulation 2009*, pages 12–21, June 2009.

[GC78] Jorge A Gutierrez and Anil K Chopra. A substructure method for earthquake analysis of structures including structure-soil interaction. *Earthquake Engineering & Structural Dynamics*, 6(1):51–69, 1978.

[GC08] Rakesh K. Goel and Anil K. Chopra. Role of shear keys in seismic behavior of bridges crossing fault-rupture zones. *ASCE Journal of Bridge Engineering*, 13(4):398–408, July 2008.

[GCA88] A. Gens, I. Carol, and E. E. Alonso. An interface element formulation for the analysis of soil–reinforcement interactions. *Computers and Geotechnics*, 7:133–151, 1988.

[GCA90] A. Gens, I. Carol, and E.E. Alonso. A constitutive model for rock joints formulation and numerical implementation. *Computers and Geotechnics*, 9(1-2):3–20, 1990.

[GCR16] Shawn C. Griffiths, Brady R. Cox, and Ellen M. Rathje. Challenges associated with site response analyses for soft soils subjected to high-intensity input ground motions. *Soil Dynamics and Earthquake Engineering*, 85:1 – 10, 2016.

[GD91] George Gazetas and Panos Dakoulas. Aspects of seismic analysis and design of rockfill dams. In *International Conference on Recent Advances in Geotechnical Earthquake Engineering and Soil Dynamics*, pages 1851–1888, 1991.

[GD10] A. Gajo and R. Denzer. Finite element modelling of saturated porous media at finite strains under dynamic conditions with compressible constituents. *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, Early View: DOI: 10.1002/nme.3051:1–38, 2010.

[GD11] A. Gajo and R. Denzer. Finite element modelling of saturated porous media at finite strains under dynamic conditions with compressible constituents. *International Journal for Numerical Methods in Engineering*, 85(13):1705–1736, 2011.

[GDA03] M. Grigoriu, O. Ditlevsen, and S.R. Arwade. A monte carlo simulation model for stationary non-gaussian processes. *Probabilistic Engineering Mechanics*, 18(1):87–95, 2003.

[GDC+22] Xiangfeng Guo, Daniel Dias, Claudio Carvajal, Laurent Peyras, and Pierre Breul. Modelling and comparison of different types of random fields: case of a real earth dam. *Engineering with Computers*, 38(5):4529–4543, 2022.

[Gea71] William C. Gear. *Numerical Inital Value Problems in Ordinary Differential Equations*. Series in Automatic Computation. Prentice–Hall, Inc, 1971.

[GEG24] Farzaneh Ghalamzan Esfahani and Alessandro Gajo. A zero-thickness interface element incorporating hydro-chemo-mechanical coupling and rate-dependency. *Acta Geotechnica*, 19:197–220, January 2024.

[Geo83] M. Georgiadis. Development of p-y curves for layered soils. In *Geotechnical Practice in Offshore Engineering*, pages 536–545. Americal Society of Civil Engineers, April 1983.

[Ger76] P. Germain. Duality and convection in continuum mechanics. In G. Fichera, editor, *Trends in Applications of Pure Mathematics to Mechanics*, pages 107–128. Pitman Publishing, 1976.

[Gey92] Charles J Geyer. Practical Markov chain Monte Carlo. *Statistical science*, pages 473–483, 1992.

[GF98] D. V. Griffiths and Gordon A. Fenton. Probabilistic analysis of exit gradients due to steady seepage. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 124(9):789–797, September 1998 1998.

[GF08] Debraj Ghosh and Sharbel Farhat. Strain and stress computations in stochastic finite element methods. *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, 74:1219–1239, 2008.

[GF09] D. V. Griffiths and Gordon A. Fenton. Probabilistic settlement analysis by stochastic and random finite-element methods. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 135(11):1629–1637, November 2009.

[GFB03] Bojan B. Guzina, Sylvain Nintcheu Fata, and Marc Bonnet. On the stress-wave imaging of cavities in a semi-infinite solid. *International Journal of Solids and Structures*, 40(6):1505 – 1523, 2003.

[GFB19] Sofia Grammatikou, Michael N. Fardis, and Dionysis Biskinis. Energy dissipation models for RC members and structures. *Earthquake Engineering & Structural Dynamics*, 48(3):287–305, 2019.

[GFB22] Sophia Grammatikou, Michael N. Fardis, and Dionysis Biskinis. Energy dissipation in reinforced concrete members before or after yielding. *Earthquake Engineering & Structural Dynamics*, 51(4):974–997, 2022.

[GG96] Lennart Gustafsson and Peter Gustafsson. Studying mixed granular flows by image analysis. In Y. K. Lin and T. C. Su, editors, *Proceedings of 11th Conference*, pages 100–103. Engineering Mechanics Division of the American Society of Civil Engineers, May 1996.

[GG02] Dan M. Ghiocel and Roger G. Ghanem. Stochastic finite–element analysis of seismic soil–structure interaction. *ASCE Journal of Engineering Mechanics*, 128(1):66–77, 2002.

[GG07] Nikos Gerolymos and George Gazetas. A model for grain-crushing-induced landslides – application to Nikawa, Kobe 1995. *Soil Dynamics and Earthquake Engineering*, 27(9):803–817, September 2007.

[GG13] Dan M. Ghiocel and Mircea Grigoriu. Efficient probabilistic seismic soil-structure interaction (ssi) analysis for nuclear structures using a reduced-order modeling in probabilistic space. In *Transactions SMiRT-22*, San Francisco, California, USA, 2013.

[GGAG09] G. Gazetas, E. Garini, I. Anastasopoulos, and T. Georgarakos. Effects of near-fault ground shaking on sliding systems. *ASCE Journal of Geotechnics and Geoenvironmental Engineering*, 135(12):1906–1921, December 2009.

[GGHM05] Nico Galoppo, Naga Govindaraju, Michael Henson, and Dinesh Manocha. LU-GPU: Algorithms for dense linear systems on graphics hardware. In *Proc. of ACM SuperComputing*, 2005.

[GGJ23] Bruno Guidio, Heedong Goh, and Chanseok Jeong. Effective seismic force retrieval from surface measurement for sh-wave reconstruction. *Soil Dynamics and Earthquake Engineering*, 165:107682, 2023.

[GGKJ24] Bruno Guidio, Heedong Goh, Loukas F. Kallivokas, and Chanseok Jeong. On the reconstruction of the near-surface seismic motion. *Soil Dynamics and Earthquake Engineering*, 177:108414, 2024.

[GH91] Tinsley A. Galyean and John F. Hughes. Sculpting: An interactive volumetric modeling technique. *ACM*, ISBN 0-89791-436(8/91):267–274, 1991.

[Gha99] Roger Ghanem. Ingredients for a general purpose stochastic finite elements implementation. *Computer Methods in Applied Mechanics and Engineering*, 168(1-4):19–34, January 1999.

[GHBD] Francisco A. Galvis, Anne M. Hulsey, Jack W. Baker, and Gregory G. Deierlein. Simulation-based methodology to identify damage indicators and safety thresholds for post-earthquake evaluation of structures. *Earthquake Engineering & Structural Dynamics*, n/a(n/a):1–22.

[GHF09] D. V. Griffiths, Jinsong Huang, and Gordon A. Fenton. Influence of spatial variability on slope reliability using 2-d random fields. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 135(10):1367–1378, October 2009.

[GHJV95] Erich Gamma, Riachard Helm, Ralph Johnson, and John Vlissides. *Design Patterns. Elements of Reusable Object–Oriented Software*. Professional Computing Series. Addison–Wesley, 1995. ISBN 0-201-63361-2.

[Gho01] Ahmed Ghobarah. Performance-based design in earthquake engineering: state of development. *Engineering structures*, 23(8):878–884, 2001.

[GI07] D. Ghosh and G. Iaccarino. Applicability of the spectral stochastic finite element method in time-dependent uncertain problems. Annual brief,, Center for Turbulence Research, Stanford University, 2007.

[Gib97] Sarah F. Gibson. 3D ChainMail: a fast algorithm for deforming volumetric objects. In Andy van Dam, editor, *Proceedings Symposium on Interactive 3D Graphics*, pages 149–154, April 1997.

[Gil92] Daniel Gilly. *UNIX in a Nutshell*. A Nutshell Handbook. O'Reilly & Associates, Inc, 103 Morris Street, Suite A, Sebastopol, CA 95472, second edition, September 1992.

[Gir59] Karl Girkmann. *Flächentragwerke; Einführung in die Elastostatik der Scheiben, Platten, Schalen und Faltwerke*. Springer-Verlag, Wien, 1959.

[Giv21] Dan Givoli. A tutorial on the adjoint method for inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 380:113810, 2021.

[GJ06] Peter Grassl and Milan Jirásek. Damage-plastic model for concrete failure. *International Journal of Solids and Structures*, 43(22-23):7166–7196, 2006.

[GJ$^+$10] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

[GJ13] R.J. Gollan and P.A. Jacobs. About the formulation, verification and validation of the hypersonic flow solver eilmer. *International Journal for Numerical Methods in Fluids*, 73(1):19–57, 9 2013.

[GJ21a] B. Guidio and C. Jeong. Full-waveform inversion of incoherent dynamic traction in a bounded 2D domain of scalar wave motions. *ASCE Journal of Engineering Mechanics*, 147(4):04021010, 2021.

[GJ21b] Bruno Guidio and Chanseok Jeong. Reconstruction of sh-wave responses in a domain surrounded by a domain reduction method layer. *Computational Mechanics*, 2021. in review.

[GJC$^+$11] Robert Graves, Thomas Jordan, Scott Callaghan, Ewa Deelman, Edward Field, Gideon Juve, Carl Kesselman, Philip Maechling, Gaurang Mehta, Kevin Milner, David Okaya, Patrick Small, and Karan Vahi. Cybershake: A physics-based seismic hazard model for southern california. *Pure and Applied Geophysics*, 168(3):367–381, 2011.

[GJGJ22a] B. Guidio, B. Jeremić, L. Guidio, and C. Jeong. Passive-seismic inversion of sh-wave input motions in a domain truncated by wave-absorbing boundary conditions. *Soil Dynamics and Earthquake Engineering*, 2022. In Print.

[GJGJ22b] B. Guidio, B. Jeremić, L. Guidio, and C. Jeonga. Full-waveform inversion of sh-wave input motions in a domain truncated by wave-absorbing boundary conditions. *Structural Dynamics and Earthquake Engineering*, 2022. in print.

[GK89] Dan Givoli and Joseph B. Keller. A finite element method for large domains. *Computer Methods in Applied Mechanics and Engineering*, 76(1):41–66, November 1989.

[GK96] R. Ghanem and R. M. Kruger. Numerical solution of spectral stochastic finite element systems. *Computer Methods in Applied Mechanics and Engineering*, 129(3):289 – 303, 1996.

[GK08] Vladimir Grazier and Erol Kalkan. Response of pendulums to complex input ground motion. *Soil Dynamics and Earthquake Engineering*, 28:621–631, 2008.

[GKA$^+$10] F Gelagoti, R Kourkoulis, I Anastasopoulos, T Tazoh, and G Gazetas. Seismic wave propagation in a very soft alluvial valley: Sensitivity to ground-motion details and soil nonlinearity, and generation of a parasitic vertical component. *Bulletin of the Seismological Society of America*, 100(6):3035–3054, December 2010.

[GKAG12] F Gelagoti, R Kourkoulis, I Anastasopoulos, and G Gazetas. Nonlinear dimensional analysis of trapezoidal valleys subjected to vertically propagating SV waves. *Bulletin of the Seismological Society of America*, 102(3):999–1017, June 2012.

[GKHK77] W. Gellert, H. Küstner, M. Hellwich, and H. Kästner, editors. *The VNR Concise Encyclopedia of Mathematics*. Van Nostrand Reinhold Company, 1977. ISBN 0-442-22646-2 ; QA40.V18.1977.

[GKM08] Steéphane Grange, Panagiotis Kotronis, and Jacky Mazars. A macro-element for a shallow foundation to simulate soil–structure interaction considering uplift. *Comptes Rendu Mecanique*, 336:856–862, 2008.

[GKM09] S. Grange, P. Kotronis, and J. Mazars. A macro-element for a circular foundation to simulate 3D soil–structure interaction. *INTERNATIONAL JOURNAL FOR NUMERICAL AND ANALYTICAL METHODS IN GEOMECHANICS*, 46(20):3651–3663, 2009.

[GL99] D.V. Griffiths and P.A. Lane. Slope stability analysis by finite elements. *Geotechnique*, 49(3):387–403, 1999.

[GL06] Murthy N Guddati and Keng-Wit Lim. Continued fraction absorbing boundary conditions for convex polygonal domains. *International Journal for Numerical Methods in Engineering*, 66(6):949–977, 2006.

[GL09] Sivapalan Gajan and Bruce L.Kutter. Contact interface model for shallow foundations subjected to combined cyclic loading. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 135(3):407–419, March 2009.

[GLCCP18] F. Gatti, F. Lopez-Caballero, D Clouteau, and R. Paolucci. On the effect of the 3-D regional geology on the seismic design of critical structures: the case of the Kashiwazaki-Kariwa nuclear power plant. *Geophysical Journal International*, 2018.

[Gle95] Lewis A. Glenn. The influence of rock material models on seismic discrimination of underground nuclear explosions. In *1995 APS Topical Conference on Shock Compression of Condensed Matter*, Seattle, WA, August 13-18 1995. Lawrence Livermore National Laboratory. UCRL-JC-120689.

[GLL+14] L. Giraldi, A. Litvinenko, D. Liu, H.G. Matthies, and A. Nouy. To be or not to be intrusive? the solution of parametric and stochastic equations—the "plain vanilla" Galerkin case. *SIAM J. Sci. Comput.*, 36(6):A2720–A2744, 2014.

[GM79] Jamshid Ghaboussi and Hassan Momen. Plasticity model for cyclic behavior of sand. In W. Wittke, editor, *Third International Conference on Numerical Methods in Geomechanics, Aachen*, pages 423–434. A. A. Balkema, 1979.

[GM84] Jamshid Ghaboussi and Hassan Momen. Plasticity model for inherently anisotropic behavior of sands. *International Journal for Numerical and Analytical Methods in Geomechanics*, 8:1–17, 1984.

[GM95a] A. Gajo and L. Mongiovi. An analytical solution for the transient response of saturated linear elastic porous media. *International Journal for Numerical and Analytical Methods in Geomechanics*, 19(6):399–413, 1995.

[GM95b] Lennart Gustafsson and Olov Marklund. Image analysis of fine-grain granular flow: Conditions for high quality measurements results. In Stein Sture, editor, *Proceedings of 10th Conference*, pages 614–617. Engineering Mechanics Division of the American Society of Civil Engineers, May 1995.

[GM98] George Gazetas and George Mylonakis. Seismic soil–structure interaction: New evidence and emerging issues. In Panos Dakoulas, Mishac Yegian, and Robert D. Holtz, editors, *Proceedings of a Specialty Conference: Geotechnical Earthquake Engineering and Soil Dynamics III*, Geotechnical Special Publication No. 75, pages 1119–1174. ASCE, August 1998. 1998.

[GM07] B. Ghosh and S.P.G. Madabhushi. Centrifuge modelling of seismic soil structure interaction effects. *Nuclear Engineering and Design*, 237(8):887 – 896, 2007.

[GM09] G. Gudehus and D. Mašin. Graphical representation of constitutive equations. *Géotechnique*, 59(2):147–151, March 2009.

[GM13] Selim Günay and Khalid M Mosalam. PEER performance-based earthquake engineering methodology, revisited. *Journal of Earthquake Engineering*, 17(6):829–858, 2013.

[GMT+21] Dewald Z. Gravett, Christos Mourlas, Vicky-Lee Taljaard, Nikolaos Bakas, George Markou, and Manolis Papadrakakis. New fundamental period formulae for soil-reinforced concrete structures interaction using machine learning algorithms and anns. *Soil Dynamics and Earthquake Engineering*, 144:106656, 2021.

[GN65] A.E. Green and P.M. Naghdi. A dynamical theory of interacting continua. *International Journal of Engineering Science*, 3:231–241, 1965.

[GN05] Behrouz Gatmiri and Khoa Van Nguyen. Time 2d fundamental solution for saturated porous media with incompressible fluid. *Communications in Numerical Methods in Engineering*, 21(3):119–132, 2005.

[GN14] Irene Guiamatsia and Giang D Nguyen. A thermodynamics-based cohesive model for interface debonding and friction. *International Journal of Solids and Structures*, 51(3):647–659, 2014.

[Gon00] Oscar Gonzalez. Exact energy and momentum conserving algorithms for general models in nonlinear elasticity. *Computer Methods in Applied Mechanics and Engineering*, 190(13-14):1763–1783, 2000.

[GP88] D. V. Griffiths and J. H. Prevost. The properties of anisotropic conical failure surface in relation to the Mohr–Coulomb criterion. *International Journal for Numerical and Analytical Methods in Geomechanics*, 12:497–504, 1988.

[GP96] L. N. B. Gummadi and A. N. Palazotto. Nonlinear finite element analysis of beams and arches using parallel processors. *Computers & Structures*, 63(3):413–428, May 1996.

[GP10] Robert W Graves and Arben Pitarka. Broadband ground-motion simulation using a hybrid approach. *Bulletin of the Seismological Society of America*, 100(5A):2095–2123, 2010.

[GP14] Robert Graves and Arben Pitarka. Refinements to the graves and pitarka (2010) broadband ground-motion simulation method. *Seismological Research Letters*, 86(1):75–80, 2014.

[GP16] Robert Graves and Arben Pitarka. Kinematic ground-motion simulations on rough faults including effects of 3D stochastic velocity perturbations. *Bulletin of the Seismological Society of America*, 106(5):2136–2153, 2016.

[GPGW10] Markus Gitterle, Alexander Popp, Michael W. Gee, and Wolfgang A. Wall. Finite deformation frictional mortar contact using a semi-smooth Newton method with consistent linearization. *International Journal for Numerical Methods in Engineering*, 84(5):543–571, October 2010.

[GPK$^+$04] S. Gajan, J. D. Phalen, B. L. Kutter, T. C. Hutchinson, and G. R. Martin. Centrifuge modeling of nonlinear cyclic load-deformation behavior of shallow foundations. In Toyoaki Nogami and Raymond B. Seed, editors, *Proceedings of the 11th International Conference on Soil Dynamics & Earthquake Engineering*, January 2004.

[GPS13] B. Goller, H.J. Pradlwarter, and G.I. Schuëller. Reliability assessment in structural dynamics. *Journal of Sound and Vibration*, 332:2488–2499, 2013.

[GPZH96] Jamshid Ghaboussi, David A. Pecknold, Ming-Fu Zhang, and Rami M. HajAli. Neural network constitutive models determined from structural tests. In Y. K. Lin and T. C. Su, editors, *Proceedings of 11th Conference*, pages 701–704. Engineering Mechanics Division of the American Society of Civil Engineers, May 1996.

[GR09] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.

[Gra79] Vladimir M. Graizer. Determination of the true ground displacement by using strong motion records. *Izvestiya, Earth Physics*, 15(12):875–885, 1979.

[Gra98] D.E. Grady. Shock-wave compression of brittle solids. *Mechanics of Materials*, 29:181–203, 1998.

[Gra04] V.M Graizer. Effect of tilt on strong motion data processing. *Soil Dynamics and Earthquake Engineering*, 25:197–204, 2004.

[Gra06] Vladimir Grazier. Tilts in strong ground motions. *Bulletin of Seismological Society of America*, 96(6):2090–2102, December 2006. doi: 10.1785/0120060065.

[Gra14] Vladimit Graizer. Comment on "comparison of time series and random-vibration theory site-response methods" by Albert R. Kottke and Ellen M. Rathje. *Bulletin of the Seismological Society of America*, 104(1):540–546, February 2014.

[Gri] Grid Infrastructure Group. Teragrid: open scientific discovery infrastructure. `http://www.teragrid.org/`.

[Gri95] Neil S. Grigg. Case method for teaching water– resources management. *ASCE Journal of Professional Issues in Engineering Education and Practice*, 121(1):30–36, January 1995.

[Gri06] M. Griebel. Sparse grids and related approximation schemes for higher dimensional problems. *Foundations of Computational Mathematics*, pages 106–161, 2006.

[Gri11] Mircea Grigoriu. To scale or not to scale seismic ground-acceleration records. *ASCE Journal of Engineering Mechanics*, 137(4):284–293, April 2011.

[Gri16] M Grigoriu. Do seismic intensity measures (ims) measure up? *Probabilistic Engineering Mechanics*, 46:80–93, 2016.

[Gro20] The HDF Group. HDF5. `https://www.hdfgroup.org/HDF5/`, 2020.

[GS91] Roger G. Ghanem and Pol D. Spanos. *Stochastic Finite Elements, A Spectral Approach*. Dover Publications Inc., revised edition edition, 1991.

[GS97] Shailesh R. Gandhi and S. Selvam. Group effect on driven piles under lateral load. *Journal of geotechnical and geoenvironmental engineering*, 123(8):702–709, August 1997.

[GS10] Johann Guilleminot and Christian Soize. A stochastic model for elasticity tensors with uncertain material symmetries. *International Journal of Solids and Structures*, 47(22-23):3121 – 3130, 2010.

[GS11] Sivapalan Gajan and Duraisamy S. Saravanathiiban. Modeling of energy dissipation in structural devices and foundation soil during seismic loading. *Soil Dynamics and Earthquake Engineering*, 31(8):1106 – 1122, 2011.

[GS13] J Goggins and S Salawdeh. Validation of nonlinear time history analysis models for single-storey concentrically braced frames using full-scale shake table tests. *Earthquake engineering & Structural dynamics*, 42(8):1151–1170, 2013.

[GSA20a] Joaquin Garcia-Suarez and Domniki Asimaki. Exact seismic response of smooth rigid retaining walls resting on stiff soil. *International Journal for Numerical and Analytical Methods in Geomechanics*, n/a(n/a), 2020.

[GSA20b] Joaquin Garcia-Suarez and Domniki Asimaki. Exact seismic response of smooth rigid retaining walls resting on stiff soil. *International Journal for Numerical and Analytical Methods in Geomechanics*, 44(13):1750–1769, 2020.

[GSMY$^+$07] Dominik Göddeke, Robert Strzodka, Jamaludin Mohd-Yusof, Patrick McCormick, Sven H.M. Buijssen, Matthias Grajewski, and Stefan Turek. Exploring weak scalability for FEM calculations on a GPU-enhanced cluster. *Parallel Computing*, 33(10–11):685–699, 2007.

[GSMY$^+$08] Dominik Göddeke, Robert Strzodka, Jamal Mohd-Yusof, Patrick McCormick, Hilmar Wobker, Christian Becker, and Stefan Turek. Using GPUs to improve multigrid solver performance on a cluster. *International Journal of Computational Science and Engineering*, 4(1):36–55, 2008.

[GSS23] Amir H. Gandomi, Christian Soize, and James R. Stewart. AI in computational mechanics and engineering sciences. *Computer Methods in Applied Mechanics and Engineering*, 407:115935, 2023.

[GSSS15] Michele Godio, Ioannis Stefanou, Karam Sab, and Jean Sulem. Dynamic finite element formulation for Cosserat elastic plates. *International Journal for Numerical Methods in Engineering*, 101(13):992–1018, 2015.

[GSSS16] Michele Godio, Ioannis Stefanou, Karam Sab, and Jean Sulem. Multisurface plasticity for Cosserat materials: Plate element implementation and validation. *International Journal for Numerical Methods in Engineering*, 108(5):456–484, 2016. nme.5219.

[GST07] Dominik Göddeke, Robert Strzodka, and Stefan Turek. Performance and accuracy of hardware-oriented native-, emulated- and mixed-precision solvers in FEM simulations. *International Journal of Parallel, Emergent and Distributed Systems*, 22(4):221–256, 2007. Special Issue: Applied Parallel Computing.

[GSV94a] A. Gajo, A. Saetta, and R. Vitaliani. Evaluation of three– and two–field finite element methods for the dynamic response of saturated soil. *International Journal for Numerical Methods in Engineering*, 37:1231–1247, 1994.

[GSV94b] A. Gajo, A. Saetta, and R. Vitaliani. Evaluation of three- and two-field finite element methods for the dynamic response of saturated soil. *International Journal for Numerical Methods in Engineering*, 37(7):1231–1247, 1994.

[GT96] F. Ghrib and R. Tinawi. Time integration algorithms for seismic response of softening structures. *11th WCEE*, 1889, 1996.

[GT05] P. Grammenoudis and Ch. Tsakmakis. Finite element implementation of large deformation micropolar plasticity exhibiting isotropic and kinematic hardening effects. *International Journal for Numerical Methods in Engineering*, 62(12):1691 – 1720, 2005.

[GT07] Michael Grottke and Kishor S Trivedi. Fighting bugs: Remove, retry, replicate, and rejuvenate. *Computer*, 40(2), 2007.

[GT12a] Vlado Gičev and Mihailo D Trifunac. Energy dissipation by nonlinear soil strains during soil–structure interaction excited by sh pulse. *Soil Dynamics and Earthquake Engineering*, 43:261–270, 2012.

[GT12b] Vlado Gičev and Mihailo D. Trifunac. A note on predetermined earthquake damage scenarios for structural health monitoring. *Structural Control and Health Monitoring*, 19(8):746–757, 2012.

[GTB68] Richard E Goodman, Robert L Taylor, and Tor L Brekke. A model for the mechanics of jointed rocks. *Journal of Soil Mechanics & Foundations Div*, 1968.

[GTLC^{+}18] F Gatti, Sara Touhami, Fernando Lopez-Caballero, Roberto Paolucci, D Clouteau, V Alves Fernandes, M Kham, and F Voldoire. Broad-band 3-D earthquake simulation at nuclear site by an all-embracing source-to-structure approach. *Soil Dynamics and Earthquake Engineering*, 115:263–280, 2018.

[GTM^{+}13] James-A. Goulet, Marie Texier, Clotaire Michel, Ian F. C. Smith, and Luc Chouinard. Quantifying the effects of modeling simplifications for structural identification of bridges. *ASCE Journal of Bridge Engineering*, doi:10.1061/(ASCE)BE.1943-5592.0000510:1–44, 2013.

[GTO15] Vlado Gičev, Mihailo D Trifunac, and Nebojša Orbović. Translation, torsion, and wave excitation of a building during soil–structure interaction excited by an earthquake SH pulse. *Soil Dynamics and Earthquake Engineering*, 77:391–401, 2015.

[GTO16a] Vlado Gičev, Mihailo D. Trifunac, and Nebojša Orbović. Two-dimensional translation, rocking, and waves in a building during soil-structure interaction excited by a plane earthquake P-wave pulse. *Soil Dynamics and Earthquake Engineering*, 90:454 – 466, 2016.

[GTO16b] Vlado Gičev, Mihailo D Trifunac, and Nebojša Orbović. Two-dimensional translation, rocking, and waves in a building during soil-structure interaction excited by a plane earthquake P-wave pulse. *Soil Dynamics and Earthquake Engineering*, 90:454–466, 2016.

[GTO16c] Vlado Gičev, Mihailo D Trifunac, and Nebojša Orbović. Two-dimensional translation, rocking, and waves in a building during soil-structure interaction excited by a plane earthquake SV-wave pulse. *Soil Dynamics and Earthquake Engineering*, 88:76–91, 2016.

[GTT21] Vlado Gičev, Mihailo D. Trifunac, and Maria I. Todorovska. Reduction of peak ground velocity by nonlinear soil response - iii: Excitation by an sv-wave pulse. *Soil Dynamics and Earthquake Engineering*, 145:106535, 2021.

[Gud21] Gerd Gudehus. Implications of the principle of effective stress. *Acta Geotechnica*, 16(6):1939–1947, 2021.

[Gui20] Bruno Peruqui Guidio. *Full-waveform Inversion of Seismic Input Motions in a Near-surface Domain*. PhD thesis, The Catholic University of America, 2020.

[Gus98] John Gustafson. Computational verifiability and feasibility of the ASCI program. *IEEE Computational Science and Engineering*, 5(1):36–45, January - March 1998.

[Gut06] Miguel A. Gutiérrez. The random dimension in computational failure mechanics. *Probabilistic Engineering Mechanics*, 21(3):257–266, 2006.

[Gut11a] B Gutierrez. Us department of energy, soil-structure interaction report. Technical report, Savannah River Operations Offices, DOE, July, 2011.

[Gut11b] Marte Gutierrez. Effects of constitutive parameters on strain localization in sands. *International Journal for Numerical and Analytical Methods in Geomechanics*, 35(2):161–178, 2011.

[Guy65] Robert J. Guyan. Reduction of stiffness and mass matrices. *AIAA Journal*, 3(2):380–380, 1965.

[GvdSVK03] Marc Gerritsma, Jan-Bart van der Steen, Peter Vos, and George Karniadakis. A new stochastic approach to transient heat conduction modeling with uncertainty. *International Journal of Heat and Mass Transfer*, 46:4681–4693, May 2003.

[GvdSVK10a] Marc Gerritsma, Jan-Bart van der Steen, Peter Vos, and George Karniadakis. Time-dependent generalized polynomial chaos. *Journal of Computational Physics*, 229:8333–8363, August 2010.

[GvdSVK10b] Marc Gerritsma, Jan-Bart van der Steen, Peter Vos, and George Karniadakis. Time-dependent generalized polynomial chaos. *Journal of Computational Physics*, 229(22):8333 – 8363, 2010.

[GVL12] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[GW01] F. Gruttmann and W Wagner. Shear correction factors in Timoshenko's beam theory for arbitrary shaped cross-sections. *Computational Mechanics*, 27:199–207, March 2001.

[GWB07] A. Gajo, D. Muir Wood, and D. Bigoni. On certain critical material and testing characteristics affecting shear band development in sand. *Géotechnique*, 57(5):449–463, 2007.

[GWC20] Davide Noè Gorini, Andrew John Whittle, and Luigi Callisto. Ultimate limit states of bridge abutments. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 146(7):04020054, 2020.

[GWGY13] M. Gao, Y. Wang, G.Y. Gao, and J. Yang. An analytical solution for the transient response of a cylindrical lined cavity in a poroelastic medium. *Soil Dynamics and Earthquake Engineering*, 46(0):30 – 40, 2013.

[GWI73] Jamshid Ghaboussi, Edward L Wilson, and Jeremy Isenberg. Finite element for rock joints and interfaces. *Journal of Soil Mechanics & Foundations Div*, 99(Proc Paper 10095), 1973.

[GWL+16] J. Guo, J.J. Wang, Y. Li, W.G. Zhao, and Y.L. Du. Three dimensional extension for Park and Ang damage model. *Structures*, 7:184–194, 2016.

[GWMZ98] H. Hao G. W. Ma and Y. X. Zhou. Modeling of wave propagation induced by underground explosion. *Computers and Geotechnics*, 22(3/4):283–303, 1998.

[GX05] Mao-sheng Gong and Li-li Xie. Study on comparison between absolute and relative input energy spectra and effects of ductility factor. *Acta Seismologica Sinica*, 18(6):717–726, 2005.

[GY89] Clayford T. Grimm and Joseph A. Yura. Shelf angles for masonry veneer. *ASCE Journal of Structural Engineering*, 115(3):509–525, March 1989.

[GZ92] Albert Edward Green and Wolfgang Zerna. *Theoretical Elasticity*. Dover Publications, Inc., 1992.

[GZGG19] P.A. Gourgiotis, Th. Zisis, A.E. Giannakopoulos, and H.G. Georgiadis. The Hertz contact problem in couple-stress elasticity. *International Journal of Solids and Structures*, 168:228 – 237, 2019.

[HA20] Mohammad Amin Hariri-Ardebili. Uncertainty quantification of heterogeneous mass concrete in macro-scale. *Soil Dynamics and Earthquake Engineering*, 137:106137, 2020.

[HA24] Mohammad Amin Hariri-Ardebili. Quantifying modeling uncertainties in seismic analysis of dams: Insights from an international benchmark study. *Earthquake Engineering & Structural Dynamics*, 53(3):1168–1194, 2024.

[HAB+04] F Heuze, R Archuleta, F Bonilla, S Day, M Doroudian, A Elgamal, S Gonzales, M Hoehler, T Lai, D Lavallee, et al. Estimating site-specific strong earthquake motions. *Soil dynamics and earthquake Engineering*, 24(3):199–223, 2004.

[HAB18] Christie Hale, Norman Abrahamson, and Yousef Bozorgnia. Probabilistic seismic hazard analysis code verification. Technical Report PEER 2018/03, Pacific Earthquake Engineering Research Center, Headquarters at the University of California, Berkeley, 2018.

[HAC74] CW Hirt, Anthony A Amsden, and JL Cook. An arbitrary lagrangian-eulerian computing method for all flow speeds. *Journal of computational physics*, 14(3):227–253, 1974.

[Had45] Jacques Hadamard. *An Essay on The Psychology of Invention in the Mathematical Field*. Princeton University Press, and Dover Publishing, Inc., 1945.

[Had02] Asadour Hadjian. A general framework for risk-consistent seismic design. *Earthquake Engineering and Structural Dynamics*, 31:601–626, 2002.

[Haf95] P. K. Haff. Constitutive laws and prediciton in granular systems. In Stein Sture, editor, *Proceedings of 10th Conference*, pages 786–789. Engineering Mechanics Division of the American Society of Civil Engineers, May 1995.

[Hag07] Andrei Hagiu. Proprietary vs. open two-sided platforms and social efficiency. *HBS Working Paper Number*, 07-095:1–37, 2007. http://hbswk.hbs.edu/item/5705.html.

[Hah93] Harley Hahn. *A Students Guide to UNIX*. McGraw – Hill Inc., 1993.

[Hal06] John F Hall. Problems encountered from the use (or misuse) of Rayleigh damping. *Earthquake Engineering & Structural Dynamics*, 35(5):525–545, 2006.

[Hal08] Tom R. Halfhill. Parallel processing with CUDA. *MicroprocessoR Report*, MPR 01/28/2008, 2008.

[Hal18] John F. Hall. Performance of viscous damping in inelastic seismic analysis of moment-frame buildings. *Earthquake Engineering & Structural Dynamics*, DOI: 10.1002/eqe.3104(0), 2018.

[Hanar] Peter Hansbo. A new approach to quadrature for finite elements incorporating hourglass control as a special case. *Computer Methods in Applied Mechanics and Engineering*, to appear.

[Har52] J. P. Den Hartog. *Advanced Strength of Materials*. Dover Publications, Inc., 1952.

[Has53] Norman A Haskell. The dispersion of surface waves on multilayered media. *Bulletin of the Seismological Society of America*, 43(1):17–34, 1953.

[Has62] Zvi Hashin. The elastic moduli of heterogeneous materials. *Journal of Applied Mechanics*, pages 143–150, March 1962.

[Has79] K. Hashiguchi. Constitutive equations of granular media with an anisotropic hardening. In W. Wittke, editor, *Third International Conference on Numerical Methods in Geomechanics, Aachen*, pages 435–439. A. A. Balkema, 1979.

[Has81] K. Hashiguchi. Constitutive equations of elastoplastic materials with anisotropic hardening and elasto–plastic transition. *ASME Journal of Applied Mechanics*, 48:297–301, 1981.

[Has83] Z. Hashin. Analysis of composite materials. *Journal of Applied Mechanics*, 50:481–505, 1983.

[Has88] K. Hashiguchi. Theoretical assessments on basic structures of elastoplastic constitutive models (panel report: International workshop on constitutive equations for granular non–cohesive soils, clevelend, 1987. In A. Saada and G. Bianchini, editors, *Constitutive Equations for Granular Non–Cohesive Soils*, pages 699–715. A. A. Balkema, July 1988.

[Hat97] Les Hatton. The T experiments: Errors in scientific software. *IEEE Computational Science and Engineering*, 4(2):27–38, April–June 1997.

[Hau94] Bjørn Haugen. *Buckling and Stability Problems for Thin Shell Structures Using High Performance Finite Elements*. PhD thesis, University of Colorado at Boulder, april 1994.

[HAU10] François Hemez, H. Sezer Atamturktur, and Cetin Unal. Defining predictive maturity for validated numerical simulations. *Computers & Structures*, 88(7-8):497 – 505, 2010.

[Hau24] Terje Haukaas. Exact sensitivity of nonlinear dynamic response with modal and rayleigh damping formulated with the tangent stiffness. *Journal of Structural Engineering*, 150(3):04024012, 2024.

[HAYa] Leonard R. Herrmann and Zaynab Al-Yassin. Finite element analysis of reinforced earth walls.

[HAYb] Leonard R. Herrmann and Zaynab Al-Yassin. Numerical analysis of reinforced soil systems.

[HB89] Thomas JR Hughes and F Brezzi. On drilling degrees of freedom. *Computer Methods in Applied Mechanics and Engineering*, 72(1):105–121, 1989.

[HB03] Yoshiaki Hisada and Jacobo Bielak. A theoretical method for computing near–fault ground motions in layered half–spaces considering static offset due to surface faulting, with a physical interpretation of fling step and rupture directivity. *Bulletin of the Seismological Society of America*, 93(3):1154–1168, 2003.

[HC] Thomas C. Hanks and C. Allin Cornell. Probabilistic seismic hazard analysis: A beginner's guide. A never published manuscript.

[HC98] K. Hashiguchi and Z.-P. Chen. Elastoplastic constitutive equation of soils with the subloading surface and the rotational hardening. *International Journal for Numerical and Analytical Methods in Geomechanics*, 22:197–227, 1998.

[HC10] Tara C. Hutchinson and Samit Ray Chaudhuri. Simplified expression for seismic fragility estimation of sliding-dominated equipment and contents. *Earthquake Spectra*, 22(3):709–732, August 2010.

[HCC09] Thomas W. Houston, Michael C. Costantino, and Carl J. Costantino. Hazard consistent structural demands and in-structure design response spectra. In *Transactions, SMiRT 20, Espoo, Finland, August 9-14 2009*, number SMiRT 20 - Division 4, Paper 1717 in ., pages 1–9, 2009.

[HCTC02] E. Hoek, C. Carranza-Torres, and B. Corkum. Hoek-Brown failure criterion: 2002 edition. In *5th North American Rock Mechanics Symposium and 17th Tunneling Association of Canada Conference: NARMS-TAC*, pages 267–271, 2002.

[HD93] Chunhua Han and Andrew Drescher. Shear bands in biaxial tests on dry coarse sand. *Soils and Foundations*, 33(1):118–132, 1993.

[HD99] Thor C. Heimdahl and Andrew Drescher. Elastic anisotropy of tire shreds. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 125(5):383–389, 1999.

[HD03] Jon C Helton and Freddie Joe Davis. Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. *Reliability Engineering & System Safety*, 81(1):23–69, 2003.

[HDJZ16] Jing-Qi Huang, Xiu-Li Du, Liu Jin, and Mi Zhao. Impact of incident angles of P waves on the dynamic responses of long lined tunnels. *Earthquake Engineering & Structural Dynamics*, 45(15):2435–2454, 2016.

[HDL75] Gary C. Hart, M Dijulio, and Marshall Lew. Torsional response of high-rise buildings. *Journal of the Structural Division*, 101:397–414, 1975.

[Hel78] Sigurdur Helgason. *Differential Geometry, Lie Groups, and Symmetric Spaces*. Pure and Applied Mathematics. Academic Press, 1978.

[Hel01] Peter Helnwein. Some remarks on the compressed matrix representation of symmetric second–order and fourth order tensors. *Computers Methods in Applied Mechanics and Engineering*, 190:2753–2770, 2001.

[Her82] Heinrich Hertz. Ueber dir berührung fester elastischer körper. *Journal für die reine and angewandte Mathematik*, pages 156–171, 1882. Berlin.

[Her78] Leonard R Herrmann. Finite element analysis of contact problems. *Journal of the Engineering Mechanics Division*, 104(5):1043–1057, 1978.

[Her96] Richard W. Hertzberg. *Deformation and fracture mechanics of engineering materials*. J. Wiley & Sons, 1996.

[Her99] Leonard R. Herrmann. Nonlinear finite element analysis for a class of steady–state problems in geotechnical engineering. Technical report, University of California at Davis, February 1999.

[HFdS03] M. Hjiaj, J. Fortin, and G. de Saxcé. A complete stress update algorithm for the non-associated Drucker-Prager model including treatment of the apex. *International Journal of Engineering Science*, 41(10):1109 – 1143, 2003.

[HFH^{+}04] Ingrid Hotz, Louis Feng, Hans Hagen, Bernd Hamann, Boris Jeremić, and Kenneth Joy. Physically based methods for tensor field visualization. In *Proceedings of the IEEE Visualization 2004 Conference (Vis04)*, Austin, Texas, October 10-15 2004.

[HFJ99] Chaojie Huang, Kenneth L. Fishman, and Rowland Richards Jr. Seismic plastic deformation in the free field. *International Journal for Numerical and Analytical Methods in Geomechanics*, 23:45–60, 1999.

[HFS^{+}20] Hany M. Hassan, Marco Fasan, Mohamed A. Sayed, Fabio Romanelli, Mohamed N. ElGabry, Franco Vaccari, and Ayman Hamed. Site-specific ground motion modeling for a historical Cairo site as a step towards computation of seismic input at cultural heritage sites. *Engineering Geology*, 2020.

[HG05] Randall J. Hickman and Marte Gutierrez. Influence of implicit integration scheme on prediction of shear band formation. *ASCE JOURNAL OF ENGINEERING MECHANICS*, 131(8):791–800, August 2005.

[HG11] T. Haukaas and P. Gardoni. Model uncertainty in finite-element analysis: Bayesian finite elements. *ASCE Journal of Engineering Mechanics*, 137(8):519–526, August 2011.

[HGJ11] Qing He, Houle Gan, and Dan Jiao. An explicit time-domain finite-element method that is unconditionally stable. Purdue e-Pubs; ECE Technical Reports 421, Purdue University, 2011.

[HGMR^{+}08] Curt B Haselton, Christine A Goulet, Judith Mitrani-Reiser, James L Beck, Gregory G Deierlein, Keith A Porter, Jonathan P Stewart, and Ertugrul Taciroglu. An assessment to benchmark the seismic performance of a code-conforming reinforced-concrete moment-frame building. Technical Report 2007/1, Pacific Earthquake Engineering Research Center, 2008.

[HGR^{+}23] John M. Harmon, Vahe Gabuchian, Ares J. Rosakis, Joel P. Conte, José I. Restrepo, Andrés Rodriguez, Arpit Nema, Andrea R. Pedretti, and José E. Andrade. Predicting the seismic behavior of multiblock tower structures using the level set discrete element method. *Earthquake Engineering & Structural Dynamics*, 52(9):2577–2596, 2023.

[HGSP18] Fei Han, Eshan Ganju, Rodrigo Salgado, and Monica Prezzi. Effects of interface roughness, particle geometry, and gradation on the sand-steel interface friction angle. *Journal of Geotechnical and Geoenvironmental Engineering*, 144(12):04018096–1–12, 2018.

[HHFL99] Stephen Hartzell, Stephen Harmsen, Arthur Frankel, and Shawn Larsen. Calculation of broadband time histories of ground motion: Comparison of methods and validation using strong-ground motion from the 1994 northridge earthquake. *Bulletin of the Seismological Society of America*, 89(6):1484–1504, 1999.

[HHHP16] Alexander Heinecke, Greg Henry, Maxwell Hutchinson, and Hans Pabst. Libxsmm: accelerating small matrix multiplications by runtime code generation. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, page 84. IEEE Press, 2016.

[HHL] E. Hinton, T. K. Hellen, and L. P. R. Lyons. On elasto-plastic benchmark philosophies. ––, pages 389–407, ––.

[HHN+99] M. Hiltl, C. R. Hagelberg, W. J. Nellis, T. C. Carney, and R. P. Swift. Dynamic response of berea sandstone shock-loaded under dry, wet and water-pressurized conditions. In *Proceedings of International Conference on High Pressure Science and Technology*, Honolulu, HI (USA), 1999. Preprint: UCRL-JC-135647.

[HHPA95] Shang Hsieh Hsieh, Glaucio H. Paulino, and John F. Abel. Recursive spectral algorithms for automatic domain partitioning in parallel finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 121:137–162, 1995.

[HHPA97] Shang Hsien Hsieh, Glaucio H. Paulino, and John F. Abel. Evaluation of automatic domain partioning algorithms for parallel finite element analysis. *International Journal for Numerical Methods in Engineering*, 40:1025–1051, 1997.

[HHS+01] Youssef MA Hashash, Jeffrey J Hook, Birger Schmidt, I John, and Chiang Yao. Seismic design and analysis of underground structures. *Tunnelling and underground space technology*, 16(4):247–293, 2001.

[HHT77] Hans M. Hilber, Thomas J. R. Hughes, and Robert L. Taylor. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Engineering and Structure Dynamics*, 5(3):283–292, 1977.

[Hib99] Hibbitt, Karlsson and Sorensen, Inc. ABAQUS, 1999. `http://www.hks.com`

[Hil50] R. Hill. *The Mathematical Theory of Plasticity*. The Oxford Engineering Science Series. Oxford at the Clarendon Press, 1st. edition, 1950.

[Hil58] R. Hill. A general theory of uniqueness and stability in elastic-plastic solids. *Journal of the Mechanics and Physics of Solids*, 6(3):236–249, 1958.

[Hil61] R. Hill. Discontinuity relations in mechanics of solids. In Sneddon I N and Hill R, editors, *Progress in Solid Mechanics*, pages 247–276. North Holland, 1961.

[Hil65] R. Hill. A self-consistent mechanics of composite materials. *Journal of Mechanics and Physics of Solids*, 13:213–222, 1965.

[Hil78] Rodney Hill. Aspects of invariance in solid mechanics. In Chia-Shun Yih, editor, *Advances in Applied Mechanics*, volume 18, pages 1–72. Academic Press, 1978.

[Hin11] Klaus-G. Hinzen. Rotation of vertically oriented objects during earthquakes. published via email to rotation@lists.geophysik.uni-muenchen.de group, July 2011.

[Hir03] So Hirata. Tensor contraction engine: Abstraction and automated parallel implementation of configuration-interaction, coupled-cluster, and many-body perturbation theories. *The Journal of Physical Chemistry A*, 107(46):9887–9897, 2003.

[His94] Yoshiaki Hisada. An efficient method for computing Green's functions for a layered half-space with sources and receivers at close depths. *Bulletin of the Seismological Society of America*, 84(5):1456–1472, 1994.

[His08] Yoshiaki Hisada. Broadband strong motion simulation in layered half–space using stochastic green's function technique. *Journal of Seismology*, 12(2):265–279, 2008.

[HISE88] K. Hanada, T. Iwatate, Y. Sawada, and H. Ezure. Soil-structure interaction of "jpdr" based on earthquake observations and forced vibration tests. *Nuclear Engineering and Design*, 105(2):173 – 183, 1988.

[HJ93] B. S. Hamad and J. O. Jirsa. Strength of epoxy–coated reinforcement bar splices confined with transverse reinforcement. *ACI Journal*, 90(1):77–88, January/Febryary 1993.

[HJ11] T.J. Holmquist and G.R. Johnson. A computational constitutive model for glass subjected to large strains, high strain rates and high pressures. *Journal of Applied Mechanics*, 78(5), September 2011.

[HJC95] E. E. Hellawell, L. Jendele, and A. H. C. Chan. Comparison of software modelling the transportation of pollutants through saturated soil. Technical report, Department of Civil Engineering, University of Glasgow, Scotland G128QQ, 1995.

[Hje97] Keith Hjelmstad. *Fundamentals of Structural Mechanics*. Prentice–Hall, 1997. ISBN 0-13-485236-2.

[HJG$^+$95] T.J. Holmquist, G.R. Johnson, D.E. Grady, C.M. Lopatin, and E.S. Hertel Jr. High strain rate properties and constitutive modeling of glass. Technical report, Alliant Techsystems Inc., Sandia National Laboratory, Alliant Techsystems Inc. 600 2nd St. NE., Hopkins, MN, USA, Sandia National Laboratory, P.O. Box 5800, Albuquerque, NM, USA, May 1995.

[HJG17] T.J. Holmquist, G.R. Johnson, and C.A. Gerlach. An improved computational constitutive model for glass. *Philosophical Transactions of the Royal Society A*, 375, January 2017.

[HK81] Robert D. Holtz and William D. Kovacs. *An Introduction to Geotechnical Engineering*. Prentice Hall, 1981.

[HK07] Terje Haukaas and Armen Der Kiureghian. Methods and object-oriented software for fe reliability and sensitivity analysis with application to a bridge structure. *ASCE Journal of Computing in Civil Engineering*, 21(3):151–163, May/June 2007.

[HK08] F. M. Hemez and J. R. Kamm. A brief overview of the state-of-the-practice and current challenges of solution verification. In Frank Graziani, editor, *Computational Methods in Transport: Verification and Validation*, pages 229–250. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[HKBK98] A.L. Halbritter, N.J. Krutzik, Z. Boyadjiev, and T. Katona. Dynamic analysis of vver type nuclear power plants using different procedures for consideration of soil-structure interaction effects. *Nuclear Engineering and Design*, 182(1):73 – 92, 1998.

[HKBT17] Xu Huang, Oh-Sung Kwon, Evan Bentz, and Julia Tcherner. Method for evaluation of concrete containment structure subjected to earthquake excitation and internal pressure increase. *Earthquake Engineering & Structural Dynamics*, pages n/a–n/a, 2017. EQE-17-0330.R3.

[HKL97] Lambertus Hesselink, Yuvai Kevy, and Yinmei Lavin. The topology of symmetric second–order 3D tensor fields. *IEEE Transaction on Visualizations and Computers Graphics*, 3(1):1–11, 1997.

[HKL06] F. Darve H.D.V. Khoa, I.O. Georgopoulos and F. Laouafa. Diffuse failure in geomaterials: Experiments and modelling. *Computers and Geotechnics*, 33(1):1–14, January 2006.

[HKL$^+$21] John M. Harmon, Konstantinos Karapiperis, Liuchi Li, Scott Moreland, and José E. Andrade. Modeling connected granular media: Particle bonding within the level set discrete element method. *Computer Methods in Applied Mechanics and Engineering*, 373:113486, 2021.

[HKP07] Tara C. Hutchinson, Falko Kuester, and Mark E. Phair. Sketching finite-element models within a unified two-dimensional framework. *ASCE Journal of Computing in Civil Engineering*, 21(3):175–186, May/June 2007.

[HKS94a] Hibbitt, Karlsson, and Sorensen. *ABAQUS Theory Manual*. Hibbitt, Karlsson & Sorensen Inc., 1080 Main Street, Pawtucket, RI 02860-4847, 5.4 edition, 1994.

[HKS94b] Hibbitt, Karlsson, and Sorensen. *ABAQUS/Explicit User's Manual*. Hibbitt, Karlsson & Sorensen Inc., 1080 Main Street, Pawtucket, RI 02860-4847, 5.4 edition, 1994.

[HKS94c] Hibbitt, Karlsson, and Sorensen. *ABAQUS/Standard User's Manual*. Hibbitt, Karlsson & Sorensen Inc., 1080 Main Street, Pawtucket, RI 02860-4847, 5.4 edition, 1994.

[HKS$^+$14] Youssef MA Hashash, Albert R Kottke, Jonathan P Stewart, Kenneth W Campbell, Byungmin Kim, Cheryl Moss, Sissy Nikolaou, Ellen M Rathje, and Walter J Silva. Reference rock site condition for central and eastern North America. *Bulletin of the Seismological Society of America*, 104(2):684–701, 2014.

[HL] Bruce Hendrickson and Robert Leland. *CHACO: Algorithms and Software for Partitioning Meshes*. Computation, Computers, and Math Center, Sandia National Laboratories. `http://www.cs.sandia.gov/CRF/chac.html`

[HL78a] T. J. R. Hughes and W. K. Liu. Implicit–explicit finite elements in transient analysis: Implementation and numerical examples. *ASME Journal of Applied Mechanics*, 45:375–378, June 1978.

[HL78b] T. J. R. Hughes and W. K. Liu. Implicit–explicit finite elements in transient analysis: Stability theory. *ASME Journal of Applied Mechanics*, 45:371–374, June 1978.

[HLA05] Yariv Hamiel, Vladimir Lyakhovsky, and Amotz Agnon. Rock dilation, nonlinear deformation, and pore pressure change under shear. *Earth and Planetary Science Letters*, 237:577–589, 2005.

[HLF$^+$02] Stephen Hartzell, Alena Leeds, Arthur Frankel, Robert A Williams, Jack Odum, William Stephenson, and Walter Silva. Simulation of broadband ground motion including nonlinear soil effects for a magnitude 6.5 earthquake on the seattle fault, seattle, washington. *Bulletin of the Seismological Society of America*, 92(2):831–853, 2002.

[HLK03] Chao Huang, Orion Lawlor, and Laxmikant V Kale. Adaptive mpi. In *International workshop on languages and compilers for parallel computing*, pages 306–322. Springer, 2003.

[HM81] Thomas C Hanks and Robin K McGuire. The character of high-frequency strong ground motion. *Bulletin of the Seismological Society of America*, 71(6):2071–2095, 1981.

[HM07] F.E. Heuze and J.P. Morris. Insights into ground shock in jointed rocks and the response of structures there-in. *International Journal of Rock Mechanics and Mining Sciences*, 44(5):647–676, July 2007.

[HM09] B.C.P. Heng and R.I. Mackie. Using design patterns in object-oriented finite element programming. *Computers & Structures*, 87(15-16):952–961, 2009.

[HM24] Junfei Huang and David McCallen. Applicability of 1D site response analysis to shallow sedimentary basins: A critical evaluation through physics-based 3D ground motion simulations. *Earthquake Engineering & Structural Dynamics*, 53(9):2876–2907, 2024.

[HM25] Junfei Huang and David McCallen. Evaluation of the impact of incident wavefield modeling on soil-structure interaction of buildings using broadband physics-based 3d earthquake simulations. *Earthquake Engineering & Structural Dynamics*, 54(9):2339–2362, 2025.

[HMKB95] Azez Hindi, Robert MacGregor, Michael Kreger, and John E. Breen. Enhancing strength and ductility of post–tensioned segmental box girder bridges. *ACI Journal*, 92(1):33–44, January/Febryary 1995.

[HMKR24] Ahmad S. Hassan, Arka Maity, Amit M. Kanvinde, and Paul W. Richards. Seismic response of block-out column-base plate connections under axial compression and flexure. *Journal of Structural Engineering*, 150(1):04023196, 2024.

[HMO$^+$16] John T. Harvey, Joep Meijer, Hasan Ozer, Imad L. Al-Qadi, Arash Saboori, and Alissa Kendal. Pavement life-cycle assessment framework. Technical Report FHWA-HIF-16-014, Applied Pavement Technology, Inc.; Federal Highway Administration, July 2016.

[HMR70] H. D. Hibbitt, P. V. Marcal, and J. R. Rice. A finite element formulation for problems of large strain and large displacements. *International Journal of Solids and Structures*, 6:1069–1086, 1970.

[HMvdG18] Jianyu Huang, Devin A Matthews, and Robert A van de Geijn. Strassen's algorithm for tensor contraction. *SIAM Journal on Scientific Computing*, 40(3):C305–C326, 2018.

[HN81] Cyril W Hirt and Billy D Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of computational physics*, 39(1):201–225, 1981.

[HN92] Mats Henricson and Erik Nyquist. Programming in c++, rules and recomendations. ftp-able postscript file, Ellemtel Telecomunication Systems Laboratories, Box 1505, 125 25 Ä;vsjö, Sweden (tel: int + 46 8 727 30 00), April 1992.

[HN25] Kazuhiro Hayashi and Masayoshi Nakashima. Insights from large-scale shake table testing: Key criticisms and potential solutions. *Earthquake Engineering & Structural Dynamics*, 54(2):604–617, 2025.

[HNGH24] Paul Hofer, Matthias Neuner, Peter Gamnitzer, and Günter Hofstetter. Revisiting strain localization analysis for elastoplastic constitutive models in geomechanics. *International Journal for Numerical Methods in Engineering*, n/a(n/a):e7579, 2024.

[HNK96] Shengyang Huang, Shoichi Nakai, and Hiroshi Katukura. An object–oriented architecture for a finite element method knowledge–based system. *International Journal for Numerical Methods in Engineering*, 39:3497–3517, 1996.

[HNW^{+}12] J. D. Hales, S. R. Novascone, R. L. Williamson, D. R. Gaston, and M. R. Tonks. Solving nonlinear solid mechanics problems with the jacobian-free newton krylov method. *Computer Modeling in Engineering and Science*, 84(22):123–152, 2012.

[HOCM23] O. M. Hunt, K. B. O'Hara, Y. Chen, and A. Martinez. Numerical and physical modeling of the effect of the cone apex angle on the penetration resistance in coarse-grained soils. *International Journal of Geomechanics*, 23(2):04022273, 2023.

[Hod84] G. Philip Jr. Hodge. Simple examples of complex phenomena in plasticity. In Richard T Shield George J Dvorak, editor, *Mechanics of Material Behavior, The Daniel C. Drucker Anniversary Volume*, pages 147–173. Elsevier, 1984.

[Hoe05] Evert Hoek. *Rock Engineering*. Vulcanhammer.net, 2005.

[Hog87] Anne Hoger. The stress conjugate to logarithmic strain. *International Journal of Solids and Structures*, 23(12):1645–1656, 1987.

[HOJF05] Yosuke Higo, Fusao Oka, Mingjing Jiang, and Yuji Fujita. Effects of transport of pore water and material heterogeneity on strain localization of fluid-saturated gradient-dependent viscoplastic geomaterial. *International Journal for Numerical and Analytical Methods in Geomechanics*, 29:495–523, 2005.

[HOL11] Joachim Berdal Haga, Harald Osnes, and Hans Petter Langtangen. On the causes of pressure oscillations in low-permeable and low-compressible porous media. *International Journal for Numerical and Analytical Methods in Geomechanics*, pages n/a–n/a, 2011.

[HOM^{+}14] K Harada, J Ohbayashi, J Matsumoto, H Yoshitomi, S Yasuda, and R Orense. Verification of effectiveness of liquefaction countermeasures during past large scale earthquakes in japan. *Soil Liquefaction during Recent Large-Scale Earthquakes*, 181, 2014.

[Hor96] John S. Horvath. Geofoam geosynthetic: Past, present and future. *Electronic Journal of Geotechnical Engineering*, October 1996. available at http://139.78.66.61/ejge/.

[Hor06] Muneo Hori. *Introduction to Computational Earthquake Engineering*. Imperial College Press, 2006.

[Hor22] Muneo Hori. Integrated earthquake simulation. *IACM Expressions*, (51):2–5, December 2022.

[Hou57] George W Housner. Interaction of building and ground during an earthquake. *Bulletin of the Seismological Society of America*, 47(3):179–186, 1957.

[HP78] Thomas J. R. Hughes and Karl S. Pister. Consistent linearization in mechanics of solids and structures. *Computers and Structures*, 8:391–397, 1978.

[HP96] John L. Hennessy and David A. Patterson. *Computer Architecture A Quantitative Approach*. Morgan Kauffmann Publishers, Inc., second edition, 1996. ISBN 1-55860-329-8.

[HP00] GT Houlsby and AM Puzrin. A thermomechanical framework for constitutive models for rate-independent dissipative materials. *International Journal of Plasticity*, 16(9):1017–1047, 2000.

[HP04] Liming Hu and Jialiu Pu. Testing and modeling of soil-structure interface. *Journal of Geotechnical and Geoenvironmental Engineering*, 130(8):851–860, 2004.

[HPG06] I. Hlaváček, J. Plešek, and D. Gabriel. Validation and sensitivity study of an elastoplastic problem using the worst scenario method. *Computer Methods in Applied Mechanics and Engineering*, 195(7-8):763–774, January 2006.

[HQP01] S. P. Huang, S. T. Quek, and K. K. Phoon. Convergence study of the truncated karhunen–Loève expansion for simulation of stochastic processes. *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, 52:1029–1043, 2001.

[HR90] C.C. Hird and D. Russell. A benchmark for soil-structure interface elements. *Computers and Geotechnics*, 10(2):139 – 147, 1990.

[HR94] Les Hatton and Andy Roberts. How accurate is scientific software? *IEEE Transaction on Software Engineering*, 20(10):185–797, October 1994.

[HR02] C.W. Holyoke and T. Rushmer. An experimental study of grain scale melt segregation mechanisms in two common crustal rock types. *Journal of Metamorphic Geology*, 20(5):493–512, 2002.

[HRRR00] J Hodowany, G Ravichandran, AJ Rosakis, and P Rosakis. Partition of plastic work into heat and stored energy in metals. *Experimental mechanics*, 40(2):113–123, 2000.

[HRS97] Weimin Han, B. Daya Reddy, and Gregory C. Schroeder. Qualitative and numerical analysis of quasi–static problems in elastoplasticity. *SIAM Journal of Numerical Analysis*, 34(1):143–177, 1997.

[HS75] Bernard Halphen and Nguyen Quoc Son. Sur les matériaux standards généralisés. *Journal De Méchanique*, 14(1):39–63, 1975.

[HSMW88] Mahantesh S. Hiremath, Ranbir S. Sandhu, Leslie W. Morland, and William E. Wolfe. Analysis of one-dimesional wave propagation in a fluid-saturated finite soil column. *International Journal for Numerical and Analytical Methods in Geomechanics*, 12(2):121–139, 1988.

[HSO10] Youssef M. A. Hashash, Hwayeon Song, and Abdolreza Osouli. Three-dimensional inverse analyses of a deep excavation in chicago clays. *International Journal for Numerical and Analytical Methods in Geomechanics*, Early View, 2010.

[HSP10] R. Holtzman, D. B. Silin, and T. W. Patzek. Frictional granular mechanics: A variational approach. *International Journal for Numerical Methods in Engineering*, 81(10):1259–1280, 2010.

[HT96] Eqramul Hoque and Fumio Tatsuoka. Elasto–plasticity of sand deformation. In Y. K. Lin and T. C. Su, editors, *Proceedings of 11th Conference*, pages 547–550. Engineering Mechanics Division of the American Society of Civil Engineers, May 1996.

[HT00] K. D. Hjelmstad and E. Taciroglu. Analysis and implementation of resilient modulus models for granular soils. *ASCE Journal of Engineering Mechanics*, 126(8):821–830, August 2000.

[HTAM12] Katsuichirou Hijikata, Makoto Takahashi, Takayuki Aoyagi, and Mitsugu Mashimo. Behavior of a base-isolated building at fukushima dai-ichi nuclear power plant during the great east japan earthquake. In *Proceedings of the International Symposium on Engineering Lessons Learned from the 2011 Great East Japan Earthquake*, Tokyo, Japan, March 1-4 2012.

[HTL23] Ray Harran, Dimitrios Terzis, and Lyesse Laloui. Mechanics, modeling, and upscaling of biocemented soils: A review of breakthroughs and challenges. *International Journal of Geomechanics*, 23(9):03123004, 2023.

[HTT01] Abdul Hayir, Maria I Todorovska, and Mihailo D Trifunac. Antiplane response of a dike with flexible soil-structure interface to incident SH waves. *Soil Dynamics and Earthquake Engineering*, 21(7):603–613, 2001.

[Hua90] Chongyu Hua. An inverse transformation for quadrilateral isoparametric elements: analysis and application. *Finite elements in analysis and design*, 7(2):159–166, 1990.

[Hua03] Bor-Shouh Huang. Ground rotational motions of the 1999 Chi-Chi, Taiwan earthquake as inferred from dense array observations. *Geophysical Research Letters*, 30(6), 2003.

[Hud96] Paul Hudak. Building domain-specific embedded languages. *ACM Computing Surveys (CSUR)*, 28(4es):196, 1996.

[Hug80] Thomas J. R. Hughes. Generalization of selective integration procedures to anisotropic and nonlinear media. *International Journal for Numerical Methods in Engineering*, pages 1413–1418, 1980.

[Hug87] Thomas J. R. Hughes. *The Finite Element Method ; Linear Static and Dynamic Finite Element Analysis*. Prentice Hall Inc., 1987.

[Hum99] Dana N. Humphrey. Civil engineering applications of tire shreds. Technical report, University of Maine, 1999.

[HV86] R.S. Harichandran and E.H. Vanmarcke. Stochastic variation of earthquake ground motion in space and time. *ASCE Journal of Engineering Mechanics*, 112:154–174, 1986.

[HW96] Y.M.A. Hashash and A.J. Whittle. Ground movement prediction for deep excavations in soft clay. *ASCE Journal of Geotechnical Engineering*, 122(6):474–486, 1996.

[HW00] A. Haraldsson and P. Wriggers. A strategy for numerical testing of frictional laws with application to contact between soil and concrete. *Computer Methods in Applied Mechanics and Engineering*, 190:963–977, 2000.

[HWL10] Yin-Nan Huang, Andrew S. Whittaker, and Nicolas Luco. Seismic performance assessment of base-isolated safety-related nuclear structures. *Earthquake Engineering and Structures Dynamics*, Early View: 20 SEP 2010 — DOI: 10.1002/eqe.1038:1–22, 2010.

[HWLH09] Yin-Nan Huang, Andrew S Whittaker, Nicolas Luco, and Ronald O Hamburger. Scaling earthquake ground motions for performance-based assessment of buildings. *Journal of Structural Engineering*, 137(3):311–321, 2009.

[HWPC09] Zhi Hua Wang, Jean H. Prévost, and Olivier Coussy. Bending of fluid-saturated linear poroelastic beams with compressible constituents. *International Journal for Numerical and Analytical Methods in Geomechanics*, 33(4):425–447, 2009.

[HWW$^+$98] Dana N. Humphrey, Nathan Whetten, James Weaver, Kenneth Recker, and Tricia A. Cosgrove. Tire shreds as lightweight fill for embankments and retaining walls. In C. Vipulanandan and David J. Elton, editors, *Proceedings of the Geo–Congress 98*, pages 51–65. Geo Institute, ASCE, 18-21 October 1998.

[HWZM21] Hsuan Wen Huang, Jiaji Wang, Chunfeng Zhao, and Y. L. Mo. Two-dimensional finite-element simulation of periodic barriers. *Journal of Engineering Mechanics*, 147(2):04020150, 2021.

[HY96] Stefan M. Holzer and Zohar Yosibash. The *p*–version of the finite element method in incremental elasto–plastic analysis. *International Journal for Numerical Method in Engineering*, 39:1859–1878, 1996.

[HYTZ04] Maosong Huang, Zhong Qi Yue, L. G. Tham, and O. C. Zienkiewicz. On the stable finite element procedures for dynamic problems of saturated porous media. *International Journal for Numerical Methods in Engineering*, 61(9):1421–1450, Sep. 2004.

[HZD17] Jingqi Huang, Mi Zhao, and Xiuli Du. Non-linear seismic responses of tunnels within normal fault ground under obliquely incident P waves. *Tunnelling and Underground Space Technology*, 61:26–39, 2017.

[HZL$^+$24] Xinyi Hou, Yanjie Zhao, Yue Liu, Zhou Yang, Kailong Wang, Li Li, Xiapu Luo, David Lo, John Grundy, and Haoyu Wang. Large language models for software engineering: A systematic literature review. *ACM Trans. Softw. Eng. Methodol.*, 33(8), December 2024.

[IAE22] International Atomic Energy Agency IAEA. Seismic hazards in site evaluation for nuclear installations. Technical Report SSG-9, Rev 1, International Atomic Energy Agency IAEA, Vianna, Austria, 2022.

[Iai98] Susumu Iai. Seismic analysis and performance of retaining structures. In Panos Dakoulas, Mishac Yegian, and Robert D. Holtz, editors, *Proceedings of a Specialty Conference: Geotechnical Earthwuake Engineering and Soil Dynamics III*, Geotechnical Special Publication No. 75, pages 1020–1044. ASCE, August 1998. 1998.

[IB95] Frank Ihlenburg and Ivo Babuška. Dispersion analysis and error estimation of Galerkin finite element methods for the Helmholtz equation. *International Journal for Numerical Methods in Engineering*, 38:3745–3774, 1995.

[IB06] IM Idriss and RW Boulanger. Semi-empirical procedures for evaluating liquefaction potential during earthquakes. *Soil Dynamics and Earthquake Engineering*, 26(2):115–130, 2006.

[Ibr93] Adnan Ibrahimbegović. Quadrilateral finite elements for analysis of thick and thin plates. *Computer Methods in Applied Mechanics and Engineering*, 110:195–209, 1993.

[Ibr94] Adnan Ibrahimbegoviffl. Equivalent spatial and material descriptions of finite deformation elastoplasticity in principal axes. *International Journal of Solids and Structures*, 31(22):3027–3040, 1994.

[IČ98] Kenji Ishihara and Miško Čubrinovski. Problems associated with liquefaction and lateral spreading during earthquake. In Panos Dakoulas, Mishac Yegian, and Robert D. Holtz, editors, *Proceedings of a Specialty Conference: Geotechnical Earthwuake Engineering and Soil Dynamics III*, Geotechnical Special Publication No. 75, pages 301–312. ASCE, August 1998. 1998.

[ICO94] ICOLD. Computer software for dams. validation. Technical Report Bulletin 94, Commission Internationale des Grands Barrages, International Committee on Large Dams, 61, avenue Kléber, 75116 Paris, France, 1994.

[ICO13] ICOLD. Guidelines for use of numerical models in dam engineering. Technical Report Bulletin 155, Commission Internationale des Grands Barrages, International Committee on Large Dams, 61, avenue Kléber, 75116 Paris, France, 2013.

[IdK85] Takeru Igusa and Armen der Kiureghian. Generation of floor response spectra including oscillator-structure interaction. *Earthquake Engineering and Structural Dynamics*, 13:661–676, 1985.

[IDLC10] Iunio Iervolino, Flavia De Luca, and Edoardo Cosenza. Spectral shape-based assessment of SDOF nonlinear response to real, adjusted and artificial accelerograms. *Engineering Structures*, 32(9):2776–2792, 2010.

[Idr14] IM Idriss. An NGA-West2 empirical model for estimating the horizontal spectral values generated by shallow crustal earthquakes. *Earthquake Spectra*, 30(3):1155–1177, 2014.

[IE67] John D. Ingram and A. Cemal Eringen. A continuum theory of chemically reacting media – II constitutive equations of reacting fluid mixtures. *International Journal of Engineering Science*, 5:289–322, 1967.

[IEE04] IEEE-SA Standards Board. IEEE standard for software verification and validation, IEEE std. 1012-2004, December 2004.

[IF93] Adnan Ibrahimbegović and François Frey. Stress resultant finite element analysis of reinforced concrete plates. *Engineering Computations*, 10:15–30, 1993.

[IG99] Adnan Ibrahimbegovic and Fadi Gharzeddine. Finite deformation plasticity in principal axes: from a manifold to the euclidean setting. *International Journal for Computer Methods in Applied Mechanics and Engineering*, 159:341–369, 1999.

[Ige17] Heiner Igel. *Computational Seismology: A Practical Introduction*. Oxford University Press, 2017.

[IH13] Daniel Ingraham and Ray Hixon. External verification analysis: A code-independent verification technique for unsteady pde codes. *Journal of Computational Physics*, 243:46–57, 2013.

[IHBD14] D. T. Ingersoll, Z. J. Houghton, R. Bromm, and C. Desportes. Integration of NuScale SMR with desalination technologies. In *Proceddings of the ASME 2014 Small Modular Reactor Symposium, SMR104*, Washington D.C. USA, April 15-17 2014.

[IHGC12] Darrel C. Ince, Leslie Hatton, and John Graham-Cumming. The case for open computer programs. *Nature*, 482:485–488, February 2012.

[IHQ$^+$11] T. Ichimura, M. Hori, P. E. Quinay, M. L. L. Wijerathne, T. Suzuki, and S. Noguchi. Comprehensive numerical analysis of fault-structure systems – computation of the large-scale seismic structural response to a given earthquake scenario –. *Earthquake Engineering & Structural Dynamics*, Early View(25 AUG 2011 — DOI: 10.1002/eqe.1158):17, 2011.

[II15] Barak Hussein Obama II. Executive order 13693 – planning for federal sustainability in the next decade. USA administration, 19Mar 2015.

[Iid12] Masahiro Iida. Three-dimensional finite-element method for soil-building interaction based on an input wave field. *International Journal of Geomechanics*, 13(4):430–440, 2012.

[III03] W. David Carrier III. Goodby, Hazen; hello Kozeny–Carman. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 129(11):1054–1056, 2003.

[IIII18] ISO/IEC/IEEE 90003 Developers, International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), and Institute of Electrical and Electronics Engineers (IEEE). *International Standard ISO/IES/IEEE 90003, Software Engineering - Guidelines for applicaiton of ISO 9001:2015 to computer software*. ISO/IEC/IEEE, first edition 2018-11 edition, November 2018.

[IK90] L. Imre and T Környey. Computer simulation of salami drying. *International Journal for Numerical Methods in Engineering*, 30:767–777, 1990.

[Il'61] AA Il'Iushin. On the postulate of plasticity. *Journal of Applied Mathematics and Mechanics*, 25(3):746–752, 1961.

[IL81] M. Iguchi and J. E. Luco. Dynamic response of flexible rectangular foundations on an elastic half-space. *Earthquake Engineering and Structural Dynamics*, 9(3):239–249, 1 1981.

[IL97] Sinan Inel and Poul V. Lade. Rotational kinematic hardening model for sand. part II characteristic work hardening law and predictions. *Computers and Geotechnics*, 21(3):217–234, 1997.

[IMGT$^+$13] T. Ibn-Mohammed, R. Greenough, S. Taylor, L. Ozawa-Meida, and A. Acquaye. Operational vs. embodied emissions in buildings - a review of current trends. *Energy and Buildings*, 66:232–245, 2013.

[Inc] Wolfram Research, Inc. Mathematica, Version 12.0. Champaign, IL, 2024.

[Inc02] MapQuest Inc. `http://www.mapquest.com/`. Online version, 2002.

[Ine92] Sinan Inel. *Kinematic Hardening Model for Sand Behavior During Large Stress Reversals*. PhD thesis, Universtity of California, Los Angeles, 1992.

[INM$^+$21] Yoshitaka Ichihara, Naohiro Nakamura, Hiroshi Moritani, Byunghyun Choi, and Akemi Nishida. 3D FEM soil-structure interaction analysis for Kashiwazaki-Kariwa nuclear power plant considering soil separation and sliding. *Frontiers in Built Environment*, 7:94, 2021.

[Int] International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC). VRML standard iso/iec-14772-1:1997.

[Int15] International Code Council, Inc. *2015 International Building Code*. International Code Council, Inc., 2015.

[Int16] International Energy Agency. Energy technology perspectives 2016, towards sustainable urban energy systems. Technical report, IEA International Energy Agency, 9 rue de la Fédération, 75739 Paris Cedex 15, France, www.iea.org, 2016.

[IOMD04] James R. Martin II, C. Guney Olgun, James K. Mitchell, and H. Turan Durgunoglu. High-modulus columns for liquefaction mitigation. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 130(6):561–571, 2004.

[IPST$^+$25] Nazanin Irani, Luis Felipe Prada-Sarmiento, Merita Tafili, Mohammad Salimi, Torsten Wichtmann, and Theodoros Triantafyllidis. Assessment of free energy functions for sand. *International Journal for Numerical and Analytical Methods in Geomechanics*, 49(1):132–150, 2025.

[IR01] K. A. Issen and J. W. Rudnicki. Theory of compaction bands in porous rock. *Physics and Chemistry of the Earth, Part A: Solid Earth and Geodesy*, 26(1–2):95–100, 2001.

[Iro71] Bruce M. Irons. Quadrature rules for brick based finite elements. *International Journal for Numerical Methods in Engineering*, 3(2):293–294, April/June 1971.

[IS92] I. M. Idriss and J. I. Sun. *Excerpts from USER'S Manual for SHAKE91: A Computer Program for Conducting Equivalent Linear Seismic Response Analyses of Horizontally Layered Soil Deposits*. Center for Geotechnical Modeling Department of Civil & Environmental Engineering University of California Davis, California, 1992.

[ISO18] ISO-90003. ISO/IEC/IEEE 90003, software engineering – guidelines for the application of ISO 9001:2015 to computer software. Technical Report ISO/IEC/IEEE 90003:2018(E), ISO, 2018.

[ISS21] ISSMGE-TC304. State-of-the-art review of inherent variability and uncertainty in geotechnical properties and models. Technical report, TC304, ISSMGE, 2021. Editors: Jianye Ching and Timo Schweckendiek.

[Ita] Itasca Consulting Group, Inc. Particle Flow Code 3D (pfc3d).

[ITB15] Yigit Isbiliroglu, Ricardo Taborda, and Jacobo Bielak. Coupled soil-structure interaction effects of building clusters during earthquakes. *Earthquake Spectra*, 31(1):463–500, 2015.

[Its00] Mikhail Itskov. On the theory of fourth-order tensors and their applications in computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 189:419–438, 2000.

[ITW90] Adnan Ibrahimbegović, Robert L. Taylor, and Edward L. Wilson. A robust quadrilateral membrane finite element with drilling degrees of freedom. *International Journal for Numerical Methods in Engineering*, 30:445–457, 1990.

[IUF03] Masahiro Ishida, Takeshi Umebara, and Jiro Fukui. Joint study on seismic retrofitting technologies for existing foundations. In *The 6th International Workshop on Micropiles*, 2003.

[IUU17] Abidemi Olujide Ilori, Ndifreke Edem Udoh, and Joseph Ignatius Umenge. Determination of soil shear properties on a soil to concrete interface using a direct shear box apparatus. *International Journal of Geo-Engineering*, 8(1):17, 2017.

[IW95] M-M. Iordache and K. Willam. Elastoplastic bifurcation in Cosserat continua. In Stein Sture, editor, *Proceedings of 10th Conference*, pages 525–528. Engineering Mechanics Division of the American Society of Civil Engineers, May 1995.

[IYF^{+}19] Kahori Iiyama, Atsushi Yoshiyuki, Kohei Fujita, Tsuyoshi Ichimura, Hitoshi Morikawa, and Muneo Hori. A point-estimate based method for soil amplification estimation using high resolution model under uncertainty of stratum boundary geometry. *Soil Dynamics and Earthquake Engineering*, 121:480 – 490, 2019.

[JA16] Alex X. Jerves and José E. Andrade. A micro-mechanical study of peak strength and critical state. *International Journal for Numerical and Analytical Methods in Geomechanics*, 40(8):1184–1202, 2016. nag.2478.

[JAH21] A. Johari, A.H. Amjadi, and A. Heidari. Stochastic nonlinear ground response analysis: A case study site in Shiraz, Iran. *Scientia Iranica*, 28(4):2070–2086, 2021.

[Jan05a] Robert Jankowski. Impact force spectrum for damage assessment of earthquake-induced structural pounding. *Key Engineering Materials*, 293-294:711–718, 2005.

[Jan05b] Robert Jankowski. Non-linear viscoelastic modelling of earthquake-induced structural pounding. *Earthquake Engineering and Structural Dynamics*, 34:595–611, 2005.

[Jan06] Robert Jankowski. Analytical expression between the impact damping ratio and the coefficient of restitution in the non-linear viscoelastic model of structural pounding. *Earthquake Engineering and Structural Dynamics*, 35:517–524, 2006.

[Jan07] Robert Jankowski. Theoretical and experimental assessment of parameters for the non-linear viscoelastic model of structural pounding. *Journal of Theoretical and Applied Mechanics*, 45(4):931–942, 2007.

[Jas96] Hrvoje Jasak. *Error analysis and estimation for finite volume method with applications to fluid flow*. PhD thesis, Imperial College of Science, Technology and Medicine, 1996.

[JASZ^{+}94] Boris Jeremić, Khalid Al-Shibli, Runing Zhang, Roy Swanson, and Stein Sture. Static and dynamic testing of MGM triaxial specimens. Report to NASA Marshall Space Flight Center, Contract: NAS8-38779, University of Colorado at Boulder, February 1994.

[JB01] M. Jirasek and Z. P. Bazant. General Elastoplastic Constitutive Models. *Inelast. Anal. Struct.*, 2001.

[JC97] V. Jovičić and M. R. Coop. Stiffness of coarse–grained soils at small strains. *Gétechnique*, 3:545–561, 1997.

[JC05] Boris Jeremić and Zhao Cheng. Significance of equal principal stretches in computational hyperelasticity. *Communications in Numerical Methods in Engineering*, 21(9):477–486, September 2005.

[JC09a] Boris Jeremić and Zhao Cheng. Numerical modeling and simulations of piles in liquefied soils. *Soil Dynamics and Earthquake Engineering*, in Print, 2009.

[JC09b] Boris Jeremić and Zhao Cheng. On large deformation hyperelasto–plasticity of anisotropic materials. *Communications in Numerical Methods in Engineering*, 25(4):391–400, 2009.

[JCK17] Sang Yun Je, Yoon-Suk Chang, and Sung-Sik Kang. Dynamic characteristics assessment of reactor vessel internals with fluid-structure interaction. *Nuclear Engineering and Technology*, 2017.

[JCT09] Boris Jeremić, Zhao Cheng, and Nima Tafazzoli. Application Programming Interface (API) for UCD CompGeoMech libraries. http://sokocalo.engr.ucdavis.edu/~jeremic/research/index.html#Sources_and_API, 2009.

[JCTD08] Boris Jeremić, Zhao Cheng, Mahdi Taiebat, and Yannis F. Dafalias. Numerical simulation of fully saturated porous materials. *International Journal for Numerical and Analytical Methods in Geomechanics*, 32(13):1635–1660, 2008.

[JCW12] Boris Jeremić, Justin Coleman, and Andrew Whittaker. *Seismic Analysis of Safety-Related Nuclear Structures*, chapter Nonlinear Time-Domain Soil-Structure-Interaction Analysis. ASCE-4-2012. ASCE, American Society of Civil Engineers, 2012.

[JCZ07] J.C. Jaeger, N.G.W. Cook, and R.W. Zimmerman. *Fundamentals of rock mechanics*. Blackwell Pub., 2007.

[JD81] D J Janssen and B J Dempsey. Soil-moisture properties of subgrade soils. *Transportation Research Record*, 790:61–66, 1981.

[JDJ19] Guillermo A. López Jiménez, Daniel Dias, and Orianne Jenck. Effect of the soil-pile-structure interaction in seismic analysis: case of liquefiable soils. *Acta Geotechnica*, 14:1509–1525, October 2019.

[Jea24] Boris Jeremić and et al. *Nonlinear Finite Elements: Modeling and Simulation of Earthquakes, Soils, Structures and their Interaction*. Self Published, University of California, Davis, CA, USA, 1989-2024. URL: http://sokocalo.engr.ucdavis.edu/~jeremic/LectureNotes/.

[Jef97] Mike Jefferies. Plastic work and isotropic softening in unloading. *Géotechnique*, 47(5), 1997.

[Jer89] Boris Jeremić. Dynamic analysis of axisymmetric solids subjected to non-symmetric loading by the finite element method. Diploma thesis,, Faculty of Civil Engineering, Belgrade Universtity, july 1989. in Serbian. (Борис Јеремић Динамичка Анализа Ротационо Симетричних Тела оптерећених Несиметричним Оптерећењима Методом Коначних Елемената, *Дипломски Рад* Грађевински Факултет Универзитета у Београду.

[Jer92] Boris Jeremić. Nonlinear effects in structures. Report to PAK group, Kragujevac, in Serbian. (Борис Јеремић Нелинеарни Ефекти у Конструкцијама: Извештај Групи ПАК, Крагујевац), May 1992.

[Jer93] Boris Jeremić. **nDarray** programming tool. Object Oriented Approach to Numerical Computations in Elastoplactity, Reference Manual University of Colorado at Boulder, December 1993.

[Jer94] Boris Jeremić. Implicit integration rules in plasticity: Theory and implementation. Master's thesis, University of Colorado at Boulder, May 1994.

[Jer96] Boris Jeremić. Object oriented numerical computations: Applications in continuum mechanics. Lecture for the Geotechnical Engineering seminar series, Universtity of Colorado at Boulder, October 1996.

[Jer97] Boris Jeremić. *Finite Deformation Hyperelasto–Plasticity of Geomaterials*. PhD thesis, University of Colorado at Boulder, July 1997.

[Jer00] Boris Jeremić. Finite element methods for 3D slope stability analysis. In Gordon A. Fenton D. V. Griffiths and Timothy R. Martin, editors, *Geotechnical Special Publications, No. 101*, volume Slope Stability 2000, pages 224–238. Geo Institute, 2000.

[Jer01a] Boris Jeremić. Geotechnical capabilities in opensees. Technical Report PEER – 2132000-3, University of California, Pacific Earthquake Engineering Research Center, Richmod, CA, September 2001. available at `http://sokocalo.engr.ucdavis.edu/OpenSees/report01.pdf`.

[Jer01b] Boris Jeremić. Line search techniques in elastic–plastic finite element computations in geomechanics. *Communications in Numerical Methods in Engineering*, 17(2):115–125, January 2001.

[Jer01c] Boris Jeremić. Recent developments in computer simulations and visualization for for geotechnical earthquake engineering problems. In Xiangwu Zeng, editor, *Proceedings of the International Workshop on Earthquake Simulations in Geotechnical Engineering*. Case Western Reserve University, 2001.

[Jer10] Boris Jeremić. High fidelity modeling and simulation of sfs interaction: energy dissipation by design. In Rolando P. Orense, Nawawi Chouw, and Michael Pender, editors, *Soil-Foundation-Structure Interaction*, pages 125–132. CRC Press, Taylor & Francis Group, 2010.

[Jer16] Boris Jeremić. Development of analytical tools for soil-structure analysis. Technical Report R444.2, Canadian Nuclear Safety Commission – Comission canadiene de sûreté nucléaire, Ottawa, Canada, 2016.

[Jer20] Boris Jeremić. Report to the US-DOE for project: A modern computational framework for the nonlinear seismic analysis of nuclear facilities and systems; the Real-ESSI Simulator system development and analysis of nuclear facilities and systems, 2015-2020. Technical Report UCD-CompMech-Oct2020, UCDavis, October 2020.

[Jet86] Philippe Jetteur. Implicit integration algorithm for elastoplasticity in plane stress analysis. *Engineering Computations*, 3:251–253, 1986.

[JFKN11] Xiaodong Ji, Gregory L Fenves, Kouichi Kajiwara, and Masayoshi Nakashima. Seismic damage detection of a full-scale shaking table test structure. *Journal of Structural Engineering*, 137(1):14–21, 2011.

[JFN$^+$19] John D. Jakeman, Fabian Franzelin, Akil Narayan, Michael Eldred, and Dirk Plfüger. Polynomial chaos expansions for dependent random variables. *Computer Methods in Applied Mechanics and Engineering*, 351:643 – 666, 2019.

[JGB08] B.-C. Jung, P. Gardoni, and G. Biscontin. Probabilistic soil identification based on cone penetration tests. *Géotechnique*, 58(7):581–590, 2008.

[JGJ02] Boris Jeremić and Niels Grønbech-Jensen. Shearing materials of spatially extended grains. In Benjamin Cook and Richard Jensen, editors, *3rd International Conference on Discrete Element Methods*, 2002.

[JH73] D E Jones and W G Holtz. *Expansive Soils-The Hidden Disaster*. National Emergency Training Center, 1973.

[JH07] G.R. Johnson and T.J. Holmquist. Determination of simple constitutive models for borosilicate glass using penetration - velocity data from ballistic experiments. In *AIP Conference Proceedings*, volume 995, December 2007.

[Jib93] Randall W. Jibson. Predicting earthquake-induced landslide displacements using newmark's sliding block analysis. Technical report, Transportation Research Record 1411, National Research Council, 1993.

[Jim19] Guillermo Alfonso López Jiménez. *Static and dynamic behavior of pile supported structures in soft soil*. PhD thesis, Université Grenoble Alpes, 2019.

[Jir94] James O. Jirsa. Divergent issues in rehabilitation of existing buildings. *Earthquake Spectra*, 10(1):95–113, 1994.

[Jir02a] Milan Jirásek. Numerical modeling of strong discontinuities. *Revue française de génie civil*, 6:1133–1146, 2002.

[Jir02b] Milan Jirásek. Objective modeling of strain localization. *Revue française de génie civil*, 6:1119–1132, 2002.

[Jir07] Milan Jirásek. Mathematical analysis of strain localization. *Revue française de génie civil*, 11:977–991, 2007.

[JJ04] Ritu Jain and Boris Jeremić. The plastic domain decomposition method in parallel computational geomechanics. Technical Report UCD-CGM 02-2004, University of California, Davis, February 2004.

[JJ05] Boris Jeremić and Guanzhou Jie. Plastic domain decomposition parallel opensees. |http://sokocalo.engr.ucdavis.edu/ jeremic/PDD/—, November 2005.

[JJ07a] Boris Jeremić and Guanzhou Jie. Plastic domain decomposition method for parallel elastic–plastic finite element computations in gomechanics. Technical Report UCD–CompGeoMech–03–07, University of California, Davis, 2007. available online: `http://sokocalo.engr.ucdavis.edu/~jeremic/wwwpublications/CV-R24.pdf`.

[JJ07b] Boris Jeremić and Guanzhou Jie. Short report: Parallel finite element computations for soil–foundation—structure interaction problems. Technical Report UCD–CompGeoMech–02–07, University of California, Davis, 2007. available online: `http://sokocalo.engr.ucdavis.edu/~jeremic/wwwpublications/CV-R23.pdf`.

[JJ08] Boris Jeremić and Guanzhou Jie. Parallel soil–foundation–structure computations. In N.D. Lagaros M. Papadrakakis, D.C. Charmpis and Y. Tsompanakis, editors, *Progress in Computational Dynamics and Earthquake Engineering*. Taylor and Francis Publishers, 2008.

[JJC+25] Boris Jeremić, Guanzhou Jie, Zhao Cheng, Nima Tafazzoli, Panagiota Tasiopoulou, Federico Pisanò, José Antonio Abell, Kohei Watanabe, Yuan Feng, Sumeet Kumar Sinha, Fatemah Behbehani, Han Yang, Hexiang Wang, and Katarzyna D Staszewska. *The Real-ESSI Simulator System*. University of California, Davis, 1988-2025. `http://real-essi.us/`.

[JJP07] Boris Jeremić, Guanzhou Jie, and Matthias Preisig. Benefits and detriments of soil foundation structure interaction. In *GeoDenver*, 2007.

[JJPT09] Boris Jeremić, Guanzhou Jie, Matthias Preisig, and Nima Tafazzoli. Time domain simulation of soil–foundation–structure interaction in non–uniform soils. *Earthquake Engineering and Structural Dynamics*, 38(5):699–718, 2009.

[JJS98] D. W. Johnson, R. T. Johnson, and K. A. Smith. *Active Learning: Cooperation in the College Classroom*. Interaction Book Co., Edina, MN, 2 edition, 1998.

[JKA02] Allen L. Jones, Steven L. Kramer, and Pedro Arduino. Estimation of uncertainty in geotechnical properties for performance-based earthquake engineering. Technical Report PEER 2002/16, Pacific Earthquake Engineering Research Center, Dec. 2002.

[JKA11] Boris Jeremić, Sashi Kunnath, and Timothy D. Ancheta. Assessment of seismic input and soil structure interaction for deeply embedded, large foundations. Technical report, Canadian Nuclear Safety Commission – Comission canadiene de sûreté nucléaire, Ottawa, Canada, 2011.

[JKK$^+$98a] Mahesh Joshi, George Karypis, Vipin Kumar, Anshul Gupta, and Fred Gustavson. *METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill–Reduced Ordering of Sparse Matrices*. University of Minnesota and IBM Thomas J. Watson Research Center, 1998.

[JKK$^+$98b] Mahesh Joshi, George Karypis, Vipin Kumar, Anshul Gupta, and Fred Gustavson. *PSPASES: Scalable Parallel Direct Solver Library for Sparse Symmetric Positivie Definite Linear Systems; User Manual (version 1.0)*. Univesity of Minnesota and IBM Thomas J. Watson Research Center, 1998.

[JKO98] R. Jordan, D. Kinderlehrer, and F. Otto. The variational formulation of the fokker–planck equation. *SIAM Journal on Mathematical Analysis*, 29(1):1 – 17, 1998.

[JKP06] Boris Jeremić, Sashi Kunnath, and Matthias Preisig. On benefits/detriments of soil–foundation–structure interaction to dynamic behavior of simple structures. *International Journal for Engineering Structures*, In review, 2006.

[JKTK11] B. Jeremić, A. Kammerer, N. Tafazzoli, and B. Kamrani. The NRC ESSI Simulator. In B.K.Dutta, editor, *Proceedings of the 21st SMiRT (Structural Mechanics in Reactor Technology) Conference*, 2011. Paper # 550.

[JKX04a] Boris Jeremić, Sashi Kunnath, and Feng Xiong. Influence of soil–foundation–structure interaction on seismic response of the i-880 viaduct. *Engineering Structures*, 26(3):391–402, 2004.

[JKX04b] Boris Jeremić, Sashi Kunnath, and Feng Xiong. Influence of soil–structure interaction on seismic response of bridges. International Journal for Engineering Structures, 26(3):391–402, February 2004.

[JL69] III John Lowe. Stability analysis of embankments. In *Stability and Performance of Slopes and Embankments*, pages 1–35. Soil Mechanics and Foundations Division, American Society of Civil Engineering, August 1969.

[JL04] Boris Jeremić and Jinxiu Liao. Domain reduction method for soil-foundation-structure interaction analysis. Technical Report UCD-CGM 01-2004, University of California, Davis, January 2004.

[JLC18] Jeena R. Jayamon, Philip Line, and Finley A. Charney. State-of-the-art review on damping in wood-frame shear wall structures. *Journal of Structural Engineering*, 144(12):03118003, 2018.

[JLL14] Heng Jin, Yong Liu, and Hua-Jun Li. Experimental study on sloshing in a tank with an inner horizontal perforated plate. *Ocean Engineering*, 82:75–84, 2014.

[JLW04] Boris Jeremić, Qing Liu, and Xiaoyan Wu. Fully coupled, solid–fluid formulation and implementation. Technical Report UCD-CGM 02-2004, University of California, Davis, March 2004.

[JM97] J-M Jazequel and Bertrand Meyer. Design by contract: The lessons of ariane. *Computer*, 30(1):129–130, 1997.

[JM00] Boris Jeremić and Kevin Makles. Performance of cluster parallel computers in computational geomechanics. *ASCE Journal of Computing in Civil Engineering*, 2000. To be submitted, August 2000.

[JM15] A. Johari and M. Momeni. Stochastic analysis of ground response using non-recursive algorithm. *Soil Dynamics and Earthquake Engineering*, 69:57–82, 2015.

[JMP$^+$15] Luis A.G. Bitencourt Jr., Osvaldo L. Manzoli, Plinio G.C. Prazeres, Eduardo A. Rodrigues, and Tulio N. Bittencourt. A coupling technique for non-matching finite element meshes. *Computer Methods in Applied Mechanics and Engineering*, 290(0):19 – 44, 2015.

[Joh75] Stephen C. Johnson. Yacc: Yet another compiler-compiler. Technical report, AT&T Bell Laboratories, Murray Hill, New Jersey 07974, 1975.

[Joh77] C. Johnson. A mixed finite element method for plasticity problems with hardening. *SIAM Journal of Numerical Analysis*, 14(4):575–583, September 1977.

[Joh87] Kenneth Langstreth Johnson. *Contact mechanics*. Cambridge university press, 1987.

[Joh94] Steve Johnson. Objecting the objects. WTEC'94 Proceedings of the USENIX Winter 1994 Technical Conference, USENIX Winter 1994 Technical Conference, september 1994.

[Jon98] Reese Edward Jones. *A yield-limited Lagrange multiplier formulation for frictional contact*. PhD thesis, University of California, Berkeley, 1998.

[Jos93] Hans Peter Jostad. *Bifurcation Analysis of Frictional Materials*. PhD thesis, The University of Trondheim, The Norwegian Institute of Technology, Geotechnical Division, August 1993.

[Jov97] Vojkan Jovičić. *The Measurement and Interpretation of Small Strain Stiffness of Soils*. PhD thesis, The City University of London, February 1997.

[JP82] David Linton Johnson and Thomas J. Plona. Acoustic slow waves and the consolidation transition. *The Journal of the Acoustical Society of America*, 72(2):556–565, 1982.

[JP05] Boris Jeremić and Matthias Preisig. Seismic soil–foundation–structure interaction: Numerical modeling issues. In *ASCE Structures Congress*, New York, NY, U.S.A., April 20-24 2005.

[JP06] Boris Jeremić and Matthias Preisig. Inelastic soil–foundation–structure interaction modeling and simulations. *Soil Dynamics and Earthquake Engineering*, In Review, 2006.

[JPF97] Manoel R. Justino JR., K. C. Park, and Carlos A. Felippa. An algebraically partitioned FETI method for parallel structural analysis: Performance evaluation. *International Journal for Numerical Methods in Engineering*, 40:2739–2758, 1997.

[JPFB86] R Jl Jardine, DM Potts, AB Fourie, and JB Burland. Studies of the influence of non-linear stress–strain characteristics in soil–structure interaction. *Geotechnique*, 36(3):377–396, 1986.

[JPY$^+$04] Boris Jeremić, James Putnam, Zhaohui Yang, Kallol Sett, Jinxiu Liao, and Guanzhou Jie. Earthquake response of bridge abutment backfills constructed with tire shreds. Technical report, University of California, Davis, May 2004.

[JRP$^+$91] Jack Benjamin and Associates Inc., RPK Structural Mechanics Consulting, Pickard, Lowe and Garrick, I.M. Idriss, and Southern Company Services Inc. A methodology for assessment of nuclear power plant seismic margin (revision 1). Technical Report EPRI NP-6041-SL, Electric Power Research Institute (EPRI), August 1991.

[JRS97a] Boris Jeremić, Kenneth Runesson, and Stein Sture. Aspects of implementation of large deformation hyperelasto–plasticity in finite element analysis. In Preparation for the *International Journal for Numerical Methods in Engineering*, 1997.

[JRS97b] Boris Jeremić, Kenneth Runesson, and Stein Sture. Elastoplastic analysis of pressure sensitive materials subjected to large deformations. In *Proceedings of the 1997 Joint Summer Meeting of the American Society of Mechanical Engineers, American Society of Civil Engineers and the Society of Engineering Science*, 1997. Invited Paper.

[JRS97c] Boris Jeremić, Kenneth Runesson, and Stein Sture. Elastoplastic analysis of pressure sensitive materials subjected to large deformations. *International Journal of Solids and Structures*, 1997. Accepted for publication DUPLICATE SEE CM539.

[JRS97d] Boris Jeremić, Kenneth Runesson, and Stein Sture. Finite deformation constitutive integration algorithm for non–isotropic materials. *to be submitted to: International Journal for Numerical Methods in Engineering*, 1997. In preparation.

[JRS97e] Boris Jeremić, Kenneth Runesson, and Stein Sture. Finite deformation hyperelasto–plasticity of coupled solid–fluid systems. *In preparation*, 1997.

[JRS97f] Boris Jeremić, Kenneth Runesson, and Stein Sture. Large deformation elastoplastic analysis of geomaterials: From experiments to numerical predictions. In Jian-Xin Yuan, editor, *proceedings of the Ninth International Conference of The Association for Computer Methods and Advances in Geomechanics*, page 6 pages. A. A. Balkema Publishers, November 1997.

[JRS98a] Boris Jeremić, Kenneth Runesson, and Stein Sture. Advanced data objects in finite element programming. In *Modern Software Tools for Scientific Computing SciTools'98*. Birkhauser, 1998. To be presented.

[JRS98b] Boris Jeremić, Kenneth Runesson, and Stein Sture. Finite deformation hyperelasto–plasticity of anisotropic hardening geomaterials solids. In *Proceedings of the Thirtheen U.S. National Congress of Applied Mechanics*. University of Florida, Gainsville, June 21–26 1998.

[JRS98c] Boris Jeremić, Kenneth Runesson, and Stein Sture. Isotropic finite deformation constitutive integration algorithm. *to be submitted to: ASCE Journal of Engineering Mechanics*, 1998. In preparation.

[JRS98d] Boris Jeremić, Kenneth Runesson, and Stein Sture. Large deformation constitutive integration algorithm. In Murakami and Luco, editors, *Proceedings of the 12th Conference*, pages 1029–1032, La Jolla, California, May 1998. Engineering Mechanics Division of the American Society of Civil Engineers.

[JRS99a] Boris Jeremić, Kenneth Runesson, and Stein Sture. A model for elastic–plastic pressure sensitive materials subjected to large deformations. *International Journal of Solids and Structures*, 36(31/32):4901–4918, 1999.

[JRS99b] Boris Jeremić, Kenneth Runesson, and Stein Sture. Object oriented approach to hyperelasticity. *International Journal of Engineering with Computers*, 15(1):2–12, 1999. ISSN 0177-0667.

[JRS01] Boris Jeremić, Kenneth Runesson, and Stein Sture. Finite deformation analysis of geomaterials. *International Journal for Numerical and Analytical Methods in Geomechanics including International Journal for Mechanics of Cohesive–Frictional Materials*, 25(8):809–840, 2001.

[JS81] C. Johnson and R. Scott. A finite element method for problems in perfect plasticity using discontinuous trial functions. In W. Wunderlich, E. Stein, and K.-J. Bathe, editors, *Nonlinear Finite Element Analysis in Structural Mechanics*, pages 307–324. Springer Berlin, 1981.

[JS94] Boris Jeremić and Stein Sture. Implicit integration rules in plasticity: Theory and implementation. Report to NASA Marshall Space Flight Center, Contract: NAS8-38779, University of Colorado at Boulder, June 1994.

[JS95a] Boris Jeremić and Stein Sture. Finite element implementation of elasto plastic material model. Report to: NASA, Marshall Space Flight Center, Contract: NAS8-38779, University of Colorado at Boulder, May 1995.

[JS95b] Boris Jeremić and Stein Sture. Implicit integrations in geoplasticity. In Stein Sture, editor, *Proceedings of 10th Conference*, pages 1099–1102, Boulder, Colorado, May 1995. Engineering Mechanics Division of the American Society of Civil Engineers.

[JS96a] Boris Jeremić and Stein Sture. Refined finite element analysis of geomaterials. In Y. K. Lin and T. C. Su, editors, *Proceedings of 11th Engineering Mechanics Conference*, pages 555–558, Fort Lauderdale, Florida, May 1996. Engineering Mechanics Division of the American Society of Civil Engineers.

[JS96b] Boris Jeremić and Stein Sture. Refined solution procedures for finite element analysis in geotechnics. presentation at the CAMM seminar 96/2, Center for Acustics, Mechanics and Materials, Universtity of Colorado, October 1996.

[JS97a] Boris Jeremić and Stein Sture. Globally convergent modification of the implicit integration schemes in soil elastoplasticity. In *Proceedings of the 1997 Joint Summer Meeting of the American Society of Mechanical Engineers, American Society of Civil Engineers and the Society of Engineering Science*, 1997.

[JS97b] Boris Jeremić and Stein Sture. Implicit integrations in elasto–plastic geotechnics. *International Journal of Mechanics of Cohesive–Frictional Materials*, 2:165–183, 1997.

[JS98a] Boris Jeremić and Stein Sture. Digital image analysis technique for automatically following displacements of deforming specimens. *to be submitted to the: ASCE Journal on Computing in Civil Engineering Special Issue on Imaging Technologies in Civil and Environmental Engineering*, 1998.

[JS98b] Boris Jeremić and Stein Sture. Tensor data objects in finite element programming. *International Journal for Numerical Methods in Engineering*, 41(1):113–126, 1998.

[JS06] Boris Jeremić and Kallol Sett. The influence of uncertain material parameters on stress-strain response. In Poul V. Lade and Teruo Nakai, editors, *Geomechanics II: Testing, Modeling, and Simulation (Proceedings of the Second-U.S. Workshop on Testing, Modeling, and Simulation in Geomechanics, Kyoto, Japan, September 8-10, 2005)*, Geotechnical Special Publication No, 156, pages 132–147. American Society for Civil Engineers, August 2006.

[JS07] Boris Jeremić and Kallol Sett. Uncertain soil properties and elastic-plastic simulations in geomechanics. In *GeoDenver*, 2007.

[JS09] Boris Jeremić and Kallol Sett. On probabilistic yielding of materials. *Communications in Numerical Methods in Engineering*, 25(3):291–300, 2009.

[JS10] Boris Jeremić and Kallol Sett. Uncertain seismic wave propagation in 1D. *Soil Dynamics and Earthquake Engineering*, 2010. in Review.

[JS18] C. Jeong and E. Esmaeilzadeh Seylabi. Seismic input motion identification in a heterogeneous halfspace. *Journal of Engineering Mechanics*, 144(8):04018070, 2018.

[JS24] Jan Jaśkowiec and N. Sukumar. Penalty-free discontinuous galerkin method. *International Journal for Numerical Methods in Engineering*, 125(12):e7472, 2024.

[JSAM98] Boris Jeremić, Nathan Straz, Michael Akers, and Kevin Makles. Beowulf class parallel computer "North-Country": Design, construction and testing. Report CEE-98-01, Clarkson University, Department of Civil and Environmental Engineering, May 1998. `http://sokocalo.sc.clarkson.edu/`

[JSF$^+$02] Boris Jeremić, Gerik Scheuermann, Jan Frey, Zhaohui Yang, Bernd Hamann, Kenneth I. Joy, and Hans Hagen. Tensor visualizations in computational geomechanics. *International Journal for Numerical and Analytical Methods in Geomechanics*, 26(10):925–944, 2002.

[JSH07] James J. Johnson, Stephen A. Short, and Gregory S. Hardy. Modeling seismic incoherence effects on NPP structures: Unifying CLASSI and SASSI approaches. In *Transactions, SMiRT 19, Toronto, August 2007*, number Paper # K05/5 in ., pages 1–8, 2007.

[JSI16] Seungmin Jin, Dongwoo Sohn, and Seyoung Im. Node-to-node scheme for three-dimensional contact mechanics using polyhedral type variable-node elements. *Computer Methods in Applied Mechanics and Engineering*, 304:217 – 242, 2016.

[JSK07] Boris Jeremić, Kallol Sett, and M. Levent Kavvas. Probabilistic elasto-plasticity: formulation in 1D. *Acta Geotechnica*, 2(3):197–210, October 2007.

[JSS$^+$94] Boris Jeremić, Roy Swanson, Stein Sture, Khalid Al-Shibli, and Runing Zhang. Automation of digitization process for recording grid displacement. Report to NASA Marshall Space Flight Center, Contract: NAS8-38779, University of Colorado at Boulder, September 1994.

[JSV12] Issam Jassim, Dieter Stolle, and Pieter Vermeer. Two-phase dynamic analysis by material point method. *International Journal for Numerical and Analytical Methods in Geomechanics*, pages n/a–n/a, 2012.

[JT91] J. E. Dennis Jr. and Virginia Torzcon. Direct search methods on parallel machines. *SIAM Journal On Optimization*, 1(4):448–474, 1991.

[JT06] Hrvoje Jasak and Zeljko Tukovic. Automatic mesh motion for the unstructured finite volume method. *Transactions of FAMENA*, 30(2):1–20, 2006.

[JTA$^+$13] B Jeremić, N Tafazzoli, T Ancheta, N Orbović, and A Blahoianu. Seismic behavior of NPP structures subjected to realistic 3d, inclined seismic motions, in variable layered soil/rock, on surface or embedded foundations. *Nuclear Engineering and Design*, 265:85–94, 2013.

[JTK$^+$10] B. Jeremić, N. Tafazzoli, B. Kamrani, Y.C. Chao, C.G. Jeong, P. Tasiopoulou, K. Sett, A. Kammerer, N. Orbović, and A. Blahoianu. On seismic soil structure interaction simulations for nuclear power plants. In *Proceedings of the OECD – NEA – IAGE – ISSC Workshop on Soil Structure Interaction Knowledge and Effect on the Seismic Assessment of NPPs Structures and Components*, 2010.

[JTK$^+$12] Boris Jeremić, Nima Tafazzoli, Babak Kamranimoghaddam, Chang-Gyun Jeong, José Antonio Abell, and Federico Pisanò. The NRC ESSI Notes (DRAFT). Technical report, University of California, Davis CA, and Lawrence Livermore National Laboratory, Berkeley, CA, 2012.

[JTOB11] B. Jeremić, N. Tafazzoli, N. Orbović, and A. Blahoianu. 3D analysis of the influence of varying rock/soil profiles on seismic NPP response. In B.K.Dutta, editor, *Proceedings of the 21st SMiRT (Structural Mechanics in Reactor Technology) Conference*, 2011. Paper # 553.

[Jum79] Alfreds R. Jumikis. *Rock mechanics.* Series on rock and soil mechanics ;v. 3:5. Trans Tech Publications,, Rockport, MA, 1979.

[JVG21] A. Johari, B. Vali, and H. Golkarfard. System reliability analysis of ground response based on peak ground acceleration considering soil layers cross-correlation. *Soil Dynamics and Earthquake Engineering*, 141:106475, 2021.

[JW01] Boris Jeremić and Xiaoyan Wu. Aspects of large deformation formulation for fully coupled analysis of saturated soils. *To be Submitted to ASCE Journal of Engineering Mechanics*, 2001. Draft available at: `sokocalo.engr.ucdavis.edu/~jeremic/UCDavis/BJPapers.html`.

[JWRY00] Boris Jeremić, Dan W. Wilson, Key Rosebrook, and Zhaohui Yang. Centrifuge characterization and numerical modeling of the dynamic properties of tire shreds for use as bridge abutment backfill. Technical Report UCD CGM–00/01, University of California, Center for Geotechnical Modeling Report, Davis, May 2000.

[JX98] Boris Jeremić and Christos Xenophontos. Application of the p–version of the finite element method to elasto–plasticity with localization of deformation. Technical report, Department of Civil and Environmental Engineering (CEE98-18), Department of Mathematics and Computer Science (TR98-03), Clarkson University, 1998.

[JX99] Boris Jeremić and Christos Xenophontos. Application of the p–version of the finite element method to elasto–plasticity with localization of deformation. *Communications in Numerical Methods in Engineering*, 15:867–976, 1999.

[JXS99] Boris Jeremić, Christos Xenophontos, and Stein Sture. Modeling of continuous localization of deformation. In Roger Ghanem Nick Jones, editor, *Proceedings of 13th Engineering Mechanics Specialty Conference*, page 4 pages, Baltimore, Maryland, June 1999. Engineering Mechanics Division of the American Society of Civil Engineers.

[JY99] Boris Jeremić and Jerry Yamamuro. Anisotropic plasticity in geomechanics. In *Proceedings of the Fourth International Conference on Constitutive Laws for Engineering Materials: Experiment, Theory, Computation and Applications*, Rensselaer Polytechnic Institute, Troy, NY, USA, July 27 – 30 1999.

[JY02] Boris Jeremić and Zhaohui Yang. Template elastic–plastic computations in geomechanics. *International Journal for Numerical and Analytical Methods in Geomechanics*, 26(14):1407–1427, 2002.

[JYC$^+$25] Boris Jeremić, Zhaohui Yang, Zhao Cheng, Guanzhou Jie, Nima Tafazzoli, Matthias Preisig, Panagiota Tasiopoulou, Federico Pisanò, José Abell, Kohei Watanabe, Yuan Feng, Sumeet Kumar Sinha, Fatemah Behbehani, Han Yang, Hexiang Wang, and Katarzyna D Staszewska. *Nonlinear Finite Elements: Modeling and Simulation of Earthquakes, Soils, Structures and their Interaction*. Self-Published-Online, University of California, Davis, CA, USA, 1989-2025. URL: `http://sokocalo.engr.ucdavis.edu/~jeremic/LectureNotes/`.

[JYD$^+$98] Boris Jeremić, Zhaohui Yang, Oleg Dulin, Michael Akers, and Kevin Makles. Large scale finite element computations on parallel computers: Application to elasto–plastic problems in geomechanics. Report CEE-98-02, Clarkson University, 1998. `http://sokocalo.sc.clarkson.edu/`

[JYL00] Boris Jeremić, Zhaohui Yang, and Tiejun Li. Large scale, 3D finite element analysis of dynamic soil-foundation-structure interaction. In John Tassoulas, editor, *Proceedings of 13th Engineering Mechanics Conference: CD-ROM*, page 6 pages, Austin, Texas, May 2000. Engineering Mechanics Division of the American Society of Civil Engineers.

[JYO99] Boris Jeremić, Zhaohui Yang, and Mark Olton. Beowulf class parallel computer "GeoWulf": Design, construction and testing. Progress report, University of California, Davis, Department of Civil and Environmental Engineering, 1999. `http://sokocalo.engr.ucdavis.edu/~jeremic/GeoWulf`

[JYS$^+$99] Boris Jeremić, Zhaohui Yang, Nathan Straz, Michael Akers, Kevin Makles, and Oleg Dulin. Direct, time domain, large deformation, elastic-plastic 3D finite element analysis of soil-foundation-structure interaction during earthquakes. In *Proceedings of the 51st Earthquakes Engineering Research Institute (EERI) Annual Meeting*. EERI, February 3-6 1999.

[JYS04] Boris Jeremić, Zhaohui Yang, and Stein Sture. Influence of imperfections on constitutive behavior of geomaterials. *ASCE Journal of Engineering Mechanics*, 130(6), June 2004.

[JYW22] Boris Jeremić, Han Yang, and Hexiang Wang. Deterministic and/or probabilistic, time domain, nonlinear, inelastic earthquake soil structure interaction (ESSI) modeling and simulation the Real-ESSI Simulator. In *Transactions, SMiRT-26*. SMiRT, July 2022.

[JYWL20] Boris Jeremić, Han Yang, Hexiang Wang, and Bret Lizundia. Direct analysis of soil structure interaction case studies for ATC-144 project. Technical Report UCD-CompMech-Sep2020, UCDavis and Rutherford+Chekene, 2020.

[JYY05] C. Hsein Juang, Susan Hui Yang, and Haiming Yuan. Model uncertainty of shear wave velocity-based method for liquefaction potential evaluation. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 131(10):1274–1282, October 2005.

[JZD90] T. Wong J. Zhang and D. M. Davis. Micromechanics of pressure-induced grain crushing in porous rocks. *Journal of Geophysical Research*, 95(B1):341–352, 1990.

[JZJ00] Boris Jeremić, Runing Zhang, and Vojkan Jovičić. Discrete and finite element analysis of bender element tests. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 2000. In Preparation.

[KA12] Annie M. Kammerer and Jon P. Ake. Practical implementation guidelines for SSHAC level 3 and 4 hazard studies. NUREG 2117, United States Nuclear Regulatory Commission, 2012.

[KA13] Abhishek Kundu and Sondipon Adhikari. Transient response of structural dynamic systems with parametric uncertainty. *ASCE Journal of Engineering Mechanics*, doi:10.1061/(ASCE)EM.1943-7889.0000643:1–50, 2013.

[KA14] Abhishek Kundu and Sondipon Adhikari. Transient response of structural dynamic systems with parametric uncertainty. *ASCE Journal of Engineering Mechanics*, 140(2):315–331, February 2014.

[KA19] Nicolas M. Kuehn and Norman A. Abrahamson. Spatial correlations of ground motion for non-ergodic seismic hazard analysis. *Earthquake Engineering & Structural Dynamics*, 0(0):1–19, 2019.

[KAA22] Konstantinos Kassas, Orestis Adamidis, and Ioannis Anastasopoulos. Structure-soil-structure interaction (sssi) of adjacent buildings with shallow foundations on liquefiable soil. *Earthquake Engineering & Structural Dynamics*, 51(10):2315–2334, 2022.

[KAF14] A. Kundu, S. Adhikari, and M. I. Friswell. Stochastic finite elements of discretely parameterized random systems on domains with boundary uncertainty. *International Journal for Numerical Methods in Engineering*, 100(3):183–221, 2014.

[Kag84] Takaaki Kagawa. Lateral load behavior of full scale pile group in clay. *Journal of geotechnical engineering*, 109(10):1267–1285, Oct. 1984.

[KAGG10] R. Kourkoulis, I. Anastasopoulos, F. Gelagoti, and G. Gazetas. Interaction of foundation-structure systems with seismically precarious slopes: Numerical analysis with strain softening constitutive model. *Soil Dynamics and Earthquake Engineering*, 30(12):1430 – 1445, 2010.

[KAKZ95] M. Kimura, T. Adachi, H. Kamei, and F. Zhang. 3-D finite element analyses of the ultimate behavior of laterally loaded cast-in-place concrete piles. In G. N. Pande and S. Pietruszczak, editors, *Proceedings of the Fifth International Symposium on Numerical Models in Geomechanics, NUMOG V*, pages 589–594. A. A. Balkema, September 1995.

[Kal92] V. N. Kaliakin. A simple coordiante determination scheme for two–dimensional mesh generation. *Computers and Structures*, 43(3):505–516, 1992.

[Kal96] V. N. Kaliakin. Formulation and implementation of improved zero–thickness interface elements. In Y. K. Lin and T. C. Su, editors, *Proceedings of 11th Conference*, pages 285–289. Engineering Mechanics Division of the American Society of Civil Engineers, May 1996.

[Kam99] Marcin Kamiński. Probabilistic characterization of porous plasticity in solids. *Mechanics Research Communications*, 26(1):99–106, 1999.

[Kam02] Ann Marie Kammerer. *Undrained Response of Monterey 0/30 Sand Under Multidirectional Cyclic Simple Shear Loading Conditions*. PhD thesis, University of California, Berkeley, Fall 2002.

[Kam21] Marcin M. Kamiński. On Shannon entropy computations in selected plasticity problems. *International Journal for Numerical Methods in Engineering*, 2021.

[Kar] George Karypis. Private communications.

[Kar39] W. Karush. Minima of functions of several variables with inequalities as side constraints. Master's thesis, University of Chicago, Chicago, IL.. 1939.

[Kar67] Krishnamurty Karamcheti. *Vector Analysis and Cartesian Tensors*. Series in Mathematical Physics. Holden–Day, 1967.

[Kar97] Ozgur Karatekin. Numerical experiments on application of Richardson extrapolation with nonuniform grids. *Journal of Fluids Engineering*, 1997.

[KAS$^+$] George Karypis, Rajat Aggarwal, Kirk Schloegel, Vipin Kumar, and Shashi Shekhar. METIS family of multilevel partitioning algorithms. `http://www-users.cs.umn.edu/~karypis/metis/`

[KAS14] Ronnie Kamai, Norman A Abrahamson, and Walter J Silva. Nonlinear horizontal site amplification for constraining the nga-west2 gmpes. *Earthquake Spectra*, 30(3):1223–1240, 2014.

[KASK88] M. Klisinski, M. M. Alawi, S. Sture, and H.-Y. Ko. Elasto–plastic model for sand based on fuzzy–sets. In A. Saada and G. Bianchini, editors, *Constitutive Equations for Granular Non–Cohesive Soils*, pages 325–347. A. A. Balkema, July 1988.

[Kau94] E. Kausel. Thin-layer method: Formulation in time domain. *International Journal for Numerical Methods in Engineering*, 37:927–941, 1994.

[Kau06] Eduardo Kausel. *Fundamental Solutions in Elastodynamics, A Compendium*. Cambridge University Press, The Edinburgh Building, Cambridge CB2 2RU, UK, 2006.

[Kau07] Eduardo Kausel. Comments on soil-structure interaction. In *Proceedings of the 4th US-Japan Workshop on Soil-Structure Interaction*, pages 1–9, Tsukuba, Japan, March 2007.

[Kau10] Eduardo Kausel. Early history of soil-structure interaction. *Soil Dynamics and Earthquake Engineering*, 30(9):822 – 832, 2010. Special Issue in honour of Prof. Anestis Veletsos.

[Kau12] Eduardo Kausel. Lamb's problem at its simplest. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 2012.

[Kau17] Eduardo Kausel. *Advanced Structural Dynamics*. Cambridge University Press, 2017.

[Kav99] M. Levent Kavvas. On the coarse–graining od hydrological processes with increasing scales. *Journal of Hydrology*, 217:191–202, 1999.

[Kav03] M. Levent Kavvas. Nonlinear hydrologic processes: Conservation equations for determining their means and probability distributions. *ASCE Journal of Hydrologic Engineering*, 8(2):44–53, March 2003.

[KB87a] Miloš Kojić and Klaus-Jürgen Bathe. The "effective stress function" algorithm for thermo elasto plasticity and, creep. *International Journal for Numerical Methods in Engineering*, 24:1509–1532, 1987.

[KB87b] Miloš Kojić and Klaus-Jürgen Bathe. Studies of finite element procedures – stress solution of a closed elastic strain path with stretching and shearing using the updated lagrangian jaumann formulation. *Computers and Structures*, 26(1/2):175–179, 1987.

[KB93] Loukas F. Kallivokas and Jacobo Bielak. Time–domain analysis of transient structural acoustics problems based on the finite element method and novel absorbing boundary element. *Journal of Acoustic Society of America*, 94(6):3480–3492, 1993.

[KB01a] Petr Krysl and Zdeněk Bittnar. Parallel explicit finite element solid dynamics with domain decomposition and message passing: dual partitioning scalability. *Computers & Structures*, 79(3):345–360, January 2001.

[KB01b] W-P Kwan and SL Billington. Simulation of structural concrete under cyclic load. *Journal of Structural Engineering*, 127(12):1391–1401, 2001.

[KB09] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[KB12] S. C. Kaethner and J. A. Burridge. Embodied CO_2 of structural frames. *theStructuralEngineer*, May 2012.

[KB19] Nitin Kumar and Michele Barbato. New constitutive model for interface elements in finite-element modeling of masonry. *Journal of Engineering Mechanics*, 145(5):04019022, 2019.

[KBA$^+$12] Debi Kilb, Glenn Biasi, John Anderson, James Brune, Zhigang Peng, and Frank L Vernon. A comparison of spectral parameter kappa from small and moderate earthquakes using southern California ANZA seismic network data. *Bulletin of the Seismological Society of America*, 102(1):284–300, 2012.

[KBB89] Michael E. Kreger, Patrick M Bachman, and John E. Breen. An exploratory study of shear fatigue behavior of prestressed concrete girders. *PCI Journal*, 34(4):104–125, July–August 1989.

[KBI99] R. Kulasingam, R. W. Boulanger, and I. M. Idriss. Evaluation of CPT analysis methods against inclinometer data from moss landing. In *U.S–Japan Workshop*, 1999.

[KBM91] Loukas F. Kallivokas, Jacobo Bielak, and Richard MacCamy. Symmetric local absorbing boundaries in time and space. *ASCE Journal of Engineering Mechanics*, 117(9):2027–2048, 1991.

[KBM97] Loukas F. Kallivokas, Jacobo Bielak, and Richard C. MacCamy. A simple impedance–infinite element for the finite element solution of the three–dimensional wave equation in unbound domains. *Computer Methods in Applied Mechanics and Engineering*, 147:235–262, 1997.

[KBRLM22] Nitin Kumar, Michele Barbato, Erika L. Rengifo-López, and Fabio Matta. Capabilities and limitations of existing finite element simplified micro-modeling techniques for unreinforced masonry. *Research on Engeineering Structures and Materials*, 8(8):463–490, 2022.

[KBSG13] K. Kamojjala, R. Brannon, A. Sadeghirad, and J. Guilkey. Verification tests in solid mechanics. *Engineering with Computers*, 29(4), November 2013.

[KBV16a] Mohsen Kohrangi, Paolo Bazzurro, and Dimitrios Vamvatsikos. Vector and scalar IMs in structural response estimation, part I: Hazard analysis. *Earthquake Spectra*, 32(3):1507–1524, 2016.

[KBV16b] Mohsen Kohrangi, Paolo Bazzurro, and Dimitrios Vamvatsikos. Vector and scalar IMs in structural response estimation, part II: building demand assessment. *Earthquake Spectra*, 32(3):1525–1543, 2016.

[KBV16c] Malte Krack, Lawrence A. Bergman, and Alexander F. Vakakis. On the efficacy of friction damping in the presence of nonlinear modal interactions. *Journal of Sound and Vibration*, 370:209 – 220, 2016.

[KBVS17] Mohsen Kohrangi, Paolo Bazzurro, Dimitrios Vamvatsikos, and Andrea Spillatura. Conditional spectrum-based ground motion record selection using average spectral acceleration. *Earthquake Engineering & Structural Dynamics*, 46(10):1667–1685, 2017.

[KC] Shoma Kitayama and Michael C. Constantinou. Performance of seismically isolated and non-isolated steel-framed buildings: Sensitivity to amount and form of inherent damping. *Earthquake Engineering & Structural Dynamics*, n/a(n/a).

[KC81] Matthew R. Kuhn and Ching S. Chang. Stability, bifurcation, and softening in discrete systems: A conceptual approach for granular materials. *International Journal of Solids and Structures*, 43(20):6026–6051, 1981.

[KC04] SK Kunnath and YH Chai. Cumulative damage-based inelastic cyclic demand spectrum. *Earthquake engineering & structural dynamics*, 33(4):499–520, 2004.

[KC10] Erol Kalkan and Anil K. Chopra. Practical guidelines to select and scale earthquake records for nonlinear response history analysis of structures. Open-file report 2010, U.S. Department of the Interior, U.S. Geological Survey, 2010.

[KCAA14] Olga-Joan Ktenidou, Fabrice Cotton, Norman A Abrahamson, and John G Anderson. Taxonomy of κ: A review of definitions and estimation approaches targeted to applications. *Seismological Research Letters*, 85(1):135–146, 2014.

[KCH$^+$18a] Annie M. Kammerer, Justin L. Coleman, Youssef M. A. Hashash, James J. Johnson, Robert P. Kennedy, Andrew S. Whittaker, and Ching-Ching Yu. Nonlinear soil-structure-interaction analysis in support of seismic design and probabilistic risk assessment of nuclear facilities. Technical report, INL, 2018.

[KCH$^+$18b] Bruce L Kutter, Trevor J Carey, Takuma Hashimoto, Mourad Zeghal, Tarek Abdoun, Panagiota Kokkali, Gopal Madabhushi, Stuart K Haigh, Francesca Burali d'Arezzo, Srikanth Madabhushi, et al. Leap-gwu-2015 experiment specifications, results, and comparisons. *Soil Dynamics and Earthquake Engineering*, 113:616–628, 2018.

[KCVP19] Evangelos Kementzetzidis, Simone Corciulo, Willem G. Versteijlen, and Federico Pisanò. Geotechnical aspects of offshore wind turbine dynamics from 3D non-linear soil-structure simulations. *Soil Dynamics and Earthquake Engineering*, 120:181 – 199, 2019.

[KCW10] Ioannis V. Kalpakidis, Michael C. Constantinou, and Andrew S. Whittaker. Modeling strength degradation in lead-rubber bearings under earthquake shaking. *Earthquake Engineering and Structural Dynamics*, Early View, 2010.

[KCY$^+$03] K.K. Koo, K.T. Chau, X. Yang, S.S. Lam, and Y.L. Wong. Soil–pile–structure interaction under SH wave excitation. *Earthquake Engineering & Structural Dynamics*, 32:395–415, 2003.

[KD93] Helmut Kopka and Patrick W. Daly. *A Guide To LaTeX*. Addison – Wesley, 1993.

[KD98] Mark A. Koelling and Stephen E. Dickenson. Ground improvement case histories for liquefaction mitigation of port and near–shore structures. In Panos Dakoulas, Mishac Yegian, and Robert D. Holtz, editors, *Proceedings of a Specialty Conference: Geotechnical Earthwuake Engineering and Soil Dynamics III*, Geotechnical Special Publication No. 75, pages 614–626. ASCE, August 1998. 1998.

[KDK11] Katerina Konakli and Armen Der Kiureghian. Simulation of spatially varying ground motions including incoherence, wave-passage and differential site-response effects. *Earthquake Engineering & Structural Dynamics*, pages n/a–n/a, 2011.

[KDKD13] Katerina Konakli, Armen Der Kiureghian, and Douglas Dreger. Coherency analysis of accelerograms recorded by the upsar array during the 2004 parkfield earthquake. *Earthquake Engineering & Structural Dynamics*, pages n/a–n/a, 2013.

[KdOB11] Eduardo Kausel and João Manuel de Oliveira Barbosa. Pmls: A direct approach. *International Journal for Numerical Methods in Engineering*, pages n/a–n/a, 2011.

[Kea02] M. A. Keavey. A canonical form return mapping algorithm for rate independent plasticity. *International Journal for Numerical Methods in Engineering*, 53:1491–1510, 2002.

[Kee03] Andreas Keese. A review of recent developments in the numerical solution of stochastic partial differential equations (stochastic finite elements). Technical Report Informatikbericht Nr.: 2003-06, Institute of Scientific Computing, Technical University Braunschweig, D-38092 Braunschweig, Germany, October 2003.

[Kel91a] James M. Kelly. Dynamic and failure characteristics of Bridgestone isolation bearings. Technical Report UCB/EERC-91/04, Earthquake Engineering Research Center, University of California at Berkeley, March 1991.

[Kel91b] James M. Kelly. A long-period isolation system using low-damping isolators for nuclear facilities at soft soil sites. Technical Report UCB/EERC-91/03, Earthquake Engineering Research Center, University of California at Berkeley, March 1991.

[Kel00] L. Kellezi. Local transmitting boundaries for transient elastic analysis. *Soil Dynamics and Earthquake Engineering*, 19(7):533–547, 2000.

[Ken85] Robert P Kennedy. Various types of reported margins and their use. In *Proceedings of the EPRI/NRC Workshop on Nuclear Power Plant Reevaluation to Quatify Seismic Margins*, pages R–85–R–100. EPRI and NRC, 1985.

[Ken98a] B. L. N. Kenneth. *Introduction to Continuum Mechanics*. Samizdat Press `http://samizdat.mines.edu`, 1998.

[Ken98b] David N. Kenwright. Automatic detection of open and closed separation and attachment lines. In *Proceedings Visualization '98,*, pages 151–158. Los Alamitos, CA: IEEE Computer Society Press, Oct. 1998 1998.

[Ken11] Robert P. Kennedy. Performance-goal based (risk informed) approach for establishing the SSE site specific response spectrum for future nuclear power plants. *Nuclear Engineering and Design*, 241(3):648–656, 2011. The International Conference on Structural Mechanics in Reactor Technology (SMiRT19) Special Section.

[Ken18] Robert P. Kennedy. Private communications. RPK Structural Mechanics Consulting, 2009-2018.

[KEW05] P. Kettil, G. Engström, and N.-E. Wiberg. Coupled hydro-mechanical wave propagation in road structures. *Computers & Structures*, 83:1719–1729, August 2005.

[KEW07] P. Kettil, G. Engström, and N.-E. Wiberg. Coupled simulation of wave propagation and water pumping phenomenon in driven concrete piles. *Computers & Structures*, 85(3-4):170–178, February 2007.

[KF96] Chun-Yi Kuo and J. David Frost. Uniformity evaluation of cohesionless specimen using digital image analysis. *ASCE Journal of Geotechnical Engineering*, 122(5):209–215, may 1996.

[KFG97] Praveen Kumar and Efi Foufoula-Georgiou. Wavelet analysis for geophysical applications. *Reviews of Geophysics*, 35(4):385–412, November 1997. Paper Number 97RG00427.

[KFM97] Miloš Kojić, Nebojša Filipović, and Srboljub Mijailović. A general formulation for finite element anlysis of flow through a porous deformable medium. *Theoretical and Appled Mechanics*, 23:67–81, 1997.

[KFVM97] Miloš Kojić, Nebojša Filipović, Snežana Vulović, and Srboljub Mijailović. A finite element solution procedure for porous medium with fluid flow and electromechanical coupling. *Submitted for publication*, 1997.

[KG73] Helmut B. Kupfer and Kurt H. Gerstle. Behaviour of concrete under biaxial stresses. *Journal of Engineering Mechanics Division*, 99, EM4:853–866, August 1973.

[KG07a] Erol Kalkan and Vladimir Graizer. Coupled tilt and translational ground motion response spectra. *ASCE Journal of Structural Engineering*, 133(5):609–619, May 2007.

[KG07b] Erol Kalkan and Vladimir Grazier. Multi-component ground motion response spectra for coupled horizontal, vertical, angular accelerations, and tilt. *ISET Journal of Earthquake Technology*, 44(1):259–284, March 2007. Paper No.485.

[KGB13] O-J Ktenidou, Céline Gélis, and L-F Bonilla. A study on the variability of kappa (κ) in a borehole: Implications of the computation process. *Bulletin of the Seismological Society of America*, 103(2A):1048–1068, 2013.

[KGE$^+$21] Christopher P. Kohar, Lars Greve, Tom K. Eller, Daniel S. Connolly, and Kaan Inal. A machine learning framework for accelerating the design process using CAE simulations: An application to finite element analysis in structural crashworthiness. *Computer Methods in Applied Mechanics and Engineering*, 385:114008, 2021.

[KGK$^+$03] Payam Khashaee, John L Gross, Payam Khashaee, Hai Sang Lew, Bijan Mohraz, and Fahim Sadek. *Distribution of earthquake input energy in structures*. Diane Publishing Company, 2003.

[KGK23] Seungbum Koo, Heedong Goh, and Loukas F. Kallivokas. Wave-focusing to subsurface targets using a switching time-reversal mirror. *Soil Dynamics and Earthquake Engineering*, 166:107736, 2023.

[KGM95] R. D. Kriz, E. H. Glaessgen, and J. D. Macrae. Eigenvalue-eigenvector glyphs: Visualizing zeroth, second, fourth and higher order tensors in a continuum. In *NCSA-NIST–IMM, Workshop on Modeling the Development of Residual Stresses During Thermoset Composites Curing*, University of Illinois, Urbana-Champaign, September 15-16 1995. Institute for Mechanics and Materials, University of California, San Diego, 9500 Gilman Dr., La Jolla, California 92093-0404, Report No. 95-19, 1995.

[KGSA20] Danilo S. Kusanovic, Joaquin Garcia-Suarez, and Domniki Asimaki. Dimensional analysis: Overview and applications to problems of soil-structure interaction. ..., 2020.

[KH82] J.M Kelly and S.B. Hodder. Experimental study of lead and elastomeric dampers for base isolation systems in laminated neoprene bearings. *Bulletin of the New Zealand National Society for Earthquake Engineering,*, 15(2):53–67, 1982.

[KH03] Kazuhiko Kawashima and Keisuke Hosoiri. Rocking isolation of bridge columns on direct foundations. Technical report, Tokyo Institute of Technology, 2003. `seismic.cv.titech.ac.jp/common/PDF/publication/2003/fib2003_118.pdf`.

[KHI99] Yoshio Kitada, Tsutomu Hirotani, and Michio Iguchi. Models test on dynamic structure-structure interaction of nuclear power plant buildings. *Nuclear Engineering and Design*, 192(2-3):205 – 216, 1999.

[KHR69] Helmut Kupfer, Hubert K Hilsdorf, and Hubert Rusch. Behavior of concrete under biaxial stresses. In *Journal Proceedings*, volume 66/8, pages 656–666, 1969.

[KHVW95] D Kolymbas, I Herle, and PA Von Wolffersdorff. Hypoplastic constitutive equation with internal variables. *International Journal for Numerical and Analytical Methods in Geomechanics*, 19(6):415–436, 1995.

[KI04] H. M. Kim and J. Inoue. A spectral stochastic element free galerkin method for the problems with random material parameter. *International Journal for Numerical Methods in Engineering*, 61(11):1957 – 1975, October 2004.

[Kit03] Yoshio Kitada. On a test to resolve issues related to earthquake response of nuclear structures and the ground motions used for the test. In *Proceedings of an OECD workshop of the Relations Between Seismological DATA and Seismic Engineering*, pages 301–310, Istanbul, Turkey, 16-18 October 2003.

[KIT$^+$12] Kiyoshi Kurokawa, Katsuhiko Ishinashi, Koichi Tanaka, Kenzo Oshima, Mitsushiko Tanaka, Hisako Sakiyama, Shuya Nomura, Reiko Hachisuka, Masafuni Sakurai, and Yochinori Yokoyama. The official report of the Fukushima nuclear accident independent investigation commission. Technical report, The National Diet of Japan, 2012.

[Kiu05] Armen Der Kiureghian. Non–ergodicity and PEER's framework formula. *EARTHQUAKE ENGINEERING AND STRUCTURAL DYNAMICS*, 34:1643–1652, 2005.

[KJAS20] Constantinos Kanellopoulos, Boris Jeremić, Ioannis Anastasopoulos, and Božidar Stojadinović. Use of the domain reduction method to simulate the seismic response of an existing structure protected by resonating unit cell metamaterials. In M. Papadrakakis, M. Fragiadakis, and C. Papadimitriou, editors, *Proceedings of the EURODYN 2020, XI International Conference on Structural Dynamics,*, pages 2926–2938, Athens, Greece, November 2020. EASD Procedia.

[KK77] R. D. Krieg and D. B. Krieg. Accuracies of numerical solution methods for the elastic - perfectly plastic model. *Journal of Pressure Vessel Technology*, pages 510–515, November 1977.

[KK95a] George Karypis and Vipin Kumar. Analysis of multilevel graph partitioning. Technical Report 95-037, Univesity of Minnesota and Army HPC Research Center, 1995.

[KK95b] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 1995.

[KK95c] George Karypis and Vipin Kumar. Multilevel *k*–way partitioning scheme for irreguler graphs. *Journal of Parallel and Distributed Computing*, 1995.

[KK96a] George Karypis and Vipin Kumar. Parallel multilevel *k*–way partitioning scheme for irregular graphs. In *Supercomputing '96*, 1996.

[KK96b] M. L. Kavvas and A. Karakas. On the stochastic theory of solute transport by unsteady and steady groundwater flow in heterogeneous aquifiers. *Journal of Hydrology*, 179:321–351, 1996.

[KK97] George Karypis and Vipin Kumar. A coarse–grain parallel formulation of multilevel *k*–way graph partitioning algorithm. In *Proceedings of the 8th SIAM Conference on Parallel Processing for Scientific Computing*, 1997. `http://www.cs.umn.edu/~metis`

[KK98a] George Karypis and Vipin Kumar. Multilevel algorithms for multi–constraint graph partitioning. Technical Report 98-019, Univesity of Minnesota and Army HPC Research Center, 1998.

[KK98b] George Karypis and Vipin Kumar. Multilevel *k*–way hypergraph partitioning. Technical Report 98-036, Univesity of Minnesota, 1998.

[KK99] Piotr Kowalczyk and Michal Kleiber. Shape sensitivity in elasto–plastic computations. *International Journal for Computer Methods in Applied Mechanics and Engineering*, 159:371–386, 1999.

[KK01] S. H. Kim and K. J. Kim. Three-dimensional dynamic response of underground openings in saturated rock masses. *Earthquake Engineering and Structural Dynamics*, 30(5):765–782, 2001.

[KK07] Erol Kalkan and Sashi Kunnath. Effective cyclic energy as a measure of seismic demand. *Journal of Earthquake Engineering*, 11(5):725–751, 2007.

[KK08] Erol Kalkan and Sashi Kunnath. Relevance of absolute and relative energy content in seismic evaluation of structures. *Advances in Structural Engineering*, 11(1):17–34, 2008.

[KK10] Jun Won Kang and Loukas F Kallivokas. The inverse medium problem in 1d pml-truncated heterogeneous semi-infinite domains. *Inverse Problems in Science and Engineering*, 18(6):759–786, 2010.

[KK11] S. Kucukcoban and L.F. Kallivokas. Mixed perfectly-matched-layers for direct transient analysis in 2D elastic heterogeneous media. *Computer Methods in Applied Mechanics and Engineering*, 200(1-4):57–76, January 2011.

[KK18] T. Kokusho and Y. Kaneko. Energy evaluation for liquefaction-induced strain of loose sands by harmonic and irregular loading tests. *Soil Dynamics and Earthquake Engineering*, 114:362 – 377, 2018.

[KK24] Satyam Kumar and Manish Kumar. Rocking induced axial response and performance of lead-rubber bearings in seismically isolated nuclear plants. *Earthquake Engineering & Structural Dynamics*, 53(2):717–738, 2024.

[KKB18] Mohsen Kohrangi, Sreeram Reddy Kotha, and Paolo Bazzurro. Ground-motion models for average spectral acceleration in a period range: direct and indirect methods. *Bulletin of Earthquake Engineering*, 16(1):45–65, 2018.

[KKK$^+$24] Daehyun Kim, Taegu Kim, Yejin Kim, Yong-Hoon Byun, and Tae Sup Yun. A ChatGPT-MATLAB framework for numerical modeling in geotechnical engineering applications. *Computers and Geotechnics*, 169:106237, 2024.

[KKKK18] Maha Kenawy, Sashi Kunnath, Subodh Kolwankar, and Amit Kanvinde. Fiber-based nonlocal formulation for simulating softening in reinforced concrete beam-columns. *Journal of Structural Engineering*, 144(12):04018217, 2018.

[KKMD06] Miriam Kristeková, Jozef Kristek, Peter Moczo, and Steven M. Day. Misfit criteria for quantitative comparison of seismograms. *Bulletin of the Seismological Society of America*, 96(5):1836 – 1850, 2006.

[KKS0s] A Ketcham, Stephen, Hon-Yim Ko, and Stein Sture. An electrohydraulic earthquake motion simulator for centrifuge testing. Technical report, University of Colorado at Boulder, 80's.

[KKT$^+$23] Jae-Do Kang, Koichi Kajiwara, Yusuke Tosauchi, Eiji Sato, Takahito Inoue, Toshimi Kabeyasawa, Hitoshi Shiohara, Takuya Nagae, Toshikazu Kabeyasawa, Hiroshi Fukuyama, and Tomohisa Mukai. Shaking-table test results of a full-scale free-standing building with base sliding and rocking. *Earthquake Engineering & Structural Dynamics*, n/a(n/a), 2023.

[KL73] Roger L. Kuhlemeyer and John Lysmer. Finite element method accuracy for wave propagation problems. *Journal of Soil Mechanics and Foundations Division ASCE*, 99(SM5):421–426, May 1973.

[KL85] Moonkyum Kim and Poul V. Lade. A study of constitutive models for frictional materials. Report to Natinal Science Foundation under Grant No. CEE 8211159 UCLA–ENG–8505, Universtity of California, Los Angeles, School of Engineering and Applied Sciences, January 1985.

[KL88] M. K. Kim and P. V. Lade. Single hardening constitutive model for frictional materials I. plastic potential function. *Computers and Geotechnics*, 5:307–324, 1988.

[KL95] V. N. Kaliakin and J. Li. Insight into deficiencies associated with commonly used zero–thickness interface elements. *Computers and Geotechnics*, 17:225–252, 1995.

[KLC21] C. Khalil and F. Lopez-Caballero. Survival analysis of a liquefiable embankment subjected to sequential earthquakes. *Soil Dynamics and Earthquake Engineering*, 140:106436, 2021.

[KLM09] B.N. Khoromskij, A. Litvinenko, and H.G. Matthies. Application of hierarchical matrices for computing the Karhunen-Loève expansion. *Computing*, 84(1-2):49–67, 2009.

[KM02] Andreas Keese and Hermann G. Matthies. Efficient solvers for nonlinear stochastic problem. In H.A. Mang, F.G. Rammerstorfer, and J. Eberhardsteiner, editors, *Proceedings of the WCCM V, Fifth World Congress on Computational Mechanics*, July 7-12 2002.

[KM05] Andreas Keese and Hermann G. Matthies. Hierarchical parallelisation for the solution of stochastic finite element equations. *Computers & Structures*, 83:1033 1047, 2005.

[KMB19] Xenia Karatzia, George Mylonakis, and George Bouckovalas. Seismic isolation of surface foundations exploiting the properties of natural liquefiable soil. *Soil Dynamics and Earthquake Engineering*, 121:233 – 251, 2019.

[KMBK04] R. Kulasingam, Erik J. Malvick, Ross W. Boulanger, and Bruce L. Kutter. Strength loss and localization at silt interlayers in slopes of liquefied sand. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 130(11):1192–1202. November 2004.

[KMF18] A. Kundu, H.G. Matthies, and M.I. Friswell. Probabilistic optimization of engineering system with prescribed target design in a reduced parameter space. *Computer Methods in Applied Mechanics and Engineering*, 337:281 – 304, 2018.

[KMG+14] V. Keryvin, J.-X. Meng, S. Gicquel, J.-P. Guin, L. Charleux, J.-C. Sanglebœ, P. Pilvin, T. Rouxel, and G. Le Quilliec. Constitutive modeling of the densification process in silica glass under hydrostatic compression. *Acta Materialia*, 62:250–257, January 2014.

[KMM02] A. J. Kappos, G. D. Manolis, and I. F. Moschonas. Seismic assessment and design of R/C bridges with irregular configuration, including SSI effects. *International Journal of Engineering Structures*, 24:1337–1348, 2002.

[KMMK0] H. Kato, T. Mori, N. Murota, and M. Kikuchi. Analytical model for elastoplastic and creep-like behavior of high-damping rubber bearings. *ASCE Journal of Structural Engineering*, 0(0):04014213, 0.

[KMR92] M. Klisinski, Z. Mroz, and K. Runesson. Structure of constitutive equations in plasticity for different choices of state variables. *International Journal of Plasticity*, 8:221–243, 1992.

[KMVP20] Evangelos Kementzetzidis, Andrei V. Metrikine, Willem G. Versteijlen, and Federico Pisanò. Frequency effects in the dynamic lateral stiffness of monopiles in sand: insight from field tests and 3D FE modelling. *Géotechnique*, 00:1–15, 2020.

[KMZ97a] Miloš Kojić, Srboljub Mijailović, and Nebojša Zdravković. Modeling of muscle behavior by finite element method using Hill's three–element model. *submitted for publication*, 1997.

[KMZ97b] Miloš Kojić, Srboljub Mijailović, and Nebojša Zdravković. A numerical algorithm for stress integration of a fiber–fiber kinetics model with coulomb friction for connective tissue. *submitted for publication*, 1997.

[KMZ98] M. Kojić, S. Mijailović, and N. Zdravković. Modelling of muscle behavior by the finite element method using Hill's three–element model. *International Journal for Numerical Methods in Engineering*, 43:941–953, 1998.

[Knu84] Donald E. Knuth. Literate programming. *Computer Journal (British Computer Society)*, 27(2):97–111, 1984.

[KO72] Heinz-Otto Kreiss and Joseph Oliger. Comparison of accurate methods for the integration of hyperbolic equations. *Tellus*, 24(3):199–215, 1972.

[KO88] N. Kikuchi and J. T. Oden. *Contact Problems in Elasticity*. Studies in Applied Mathematics, 1988.

[Ko92] Hon-Yim Ko. Mechanics of soils. Lecture Notes at CU Boulder, august - decembar 1992.

[KO16] T. Kirchdoerfer and M. Ortiz. Data-driven computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 304:81 – 101, 2016.

[KOB07] Patrick Knupp, Curtis C. Ober, and Ryan B. Bond. Measuring progress in order-verification within software development projects. *Engineering with Computers.*, 23(4):271–282, 2007.

[Koe93] Andrew Koenig. C++ columns. *Journal of Object Oriented Programming*, 1989 - 1993.

[Koi53] W. T. Koiter. Stress - strain relations, uniqueness and variational theorems for elastic - plastic materials with singular yield surface. *Quarterly of Applied Mathematics*, 2:350–354, 1953.

[Koi60] W. T. Koiter. General theorems for elastic-plastic solids. In I. N. Sneddon and R. Hill, editors, *Progress in Solid Mechanics*, pages 165–221. North Holland, 1960.

[Koi76] W. T. Koiter. On the complementary energy theorem in non – linear elasticity theory. In G. Fichera, editor, *Trends in Applications of Pure Mathematics to Mechanics*, pages 207–232. Pitman Publishing, 1976.

[Koj93] Милош Којић ; Miloš Kojić. Општи концепт имплицитне интеграције конститутивних релација при нееластичном деформисању материјала. *Монографија Центра за научна истраживања Српске академије наука и уметности и Универзитета у Крагујевцу* , 1993. (Miloš Kojić, A General Concept of Implicit Integration of Constitutive Relations for Inelastic Material Deformation, in Serbian).

[Koj97] Miloš Kojić. *Computational Procedures in Inelastic Analysis of Solids and Structures*. Center for Scientific Research of Srbian Academy of Sciences and Arts and University of Kragujevac and Faculty of Mechanical Engineering University of Kragujevac, 1997. ISBN 86-82607-02-6.

[KOK20] Hiroyuki Kyokawa, Shintaro Ohno, and Ichizou Kobayashi. A method for extending a general constitutive model to consider the electro-chemo-mechanical phenomena of mineral crystals in expansive soils. *International Journal for Numerical and Analytical Methods in Geomechanics*, 44(6):749–771, 2020.

[KOKA07] T. Kawasato, T. Okutani, O. Kurimoto, and M. Akimoto. A study on evaluation of seismic response considering basemat uplift for soil-building system using 3D fem. In *Transactions on the 19th International Conference on Structural Mechanics in Reactor Technology*, August 2007.

[Kol68] Andrei N. Kolmogorov. Logical basis for information theory and probability theory. *IEEE Transactions on Information Theory*, IT-14(5):662–664, September 1968.

[Kol77] D Kolymbas. A rate-dependent constitutive equation for soils. *Mechanics Research Communications*, 4(6):367–372, 1977.

[Kol85] D. Kolymbas. A generalized hypoplastic constitutive law. In *Proceedings of the Eleventh International Conference on Soil Mechanics and Foundation Engineering*, page 2626, San Francisco, USA, 12-16 August 1985.

[Kol91] D. Kolymbas. An outline of hypoplasticity. *Archive of Applied Mechanics*, 61(3):143–151, 1991.

[Kol03] Herbert Kolsky. *Stress Waves in Solids*. Dover Pheonix, 2003. ISBN0-486-49534-5.

[Kol09] Dimitrios Kolymbas. Kinematics of shear bands. *Acta Geotechnica*, 4:315–318, 2009.

[Kon05] A. W. A. Konter. Advanced finite element contact benchmarks. Technical report, Netherlands Institute for Metals Research, 2005.

[Kor33] Alfred Korzybski. *Science and Sanity: An Introduction to Non-Aristotelian Systems and General Semantics*. International non-Aristotelian library. International Non-Aristotelian Library Publishing Company, 1933.

[Kor97] Jovze Korelc. Automatic generation of finite-element code by simultaneous optimization of expressions. *Theoretical Computer Science*, 187(1-2):231–248, 1997.

[KOS$^+$21] Selcuk Kacin, Murat Ozturk, Umur Korkut Sevim, Bayram Ali Mert, Zafer Ozer, Oguzhan Akgol, Emin Unal, and Muharrem Karaaslan. Seismic metamaterials for low-frequency mechanical wave attenuation. *Natural Hazards*, 107:213–229, May 2021.

[KOT95] Marek Klisinski, Thomas Olofsson, and Robert Tano. Modelling of cracking of concrete with mixed mode inner softening band. In Stein Sture, editor, *Proceedings of 10th Conference*, pages 1095–1098. Engineering Mechanics Division of the American Society of Civil Engineers, May 1995.

[KOW15] Kristijan Kolozvari, Kutay Orakcal, and John Wallace. Shear-flexure interaction modeling for reinforced concrete structural walls and columns under reversed cyclic loading. Technical Report PEER Report No. 2015/12, Pacific Earthquake Engineering Research Center, Pacific Earthquake Engineering Research Center, Richmod, CA, December 2015.

[KOWK02] Lovre Krstulović-Opara, Peter Wriggers, and Jože Korelc. Ac 1-continuous formulation for 3D finite deformation frictional contact. *Computational Mechanics*, 29(1):27–42, 2002.

[Koz09] Jan T Kozák. Tutorial on earthquake rotational effects: Historical examples. *Bulletin of the Seismological Society of America*, 99(2B):998–1010, 2009.

[KP71] D.C. Kent and R. Park. Flexural members with confined concrete. *ASCE Journal of Structural Division*, 97:1969–1990, 1971.

[KP93] S. Kacou and I. D. Parsons. A parallel multigrid method for history–dependent elastoplasticity computations. *Computer Methods in Applied Mechanics and Engineering*, 108:1–21, 1993.

[KP07] A. Kotsoglou and S. Pantazopoulou. Bridge–embankment interaction under transverse ground excitation. *Earthquake Engineering & Structural Dynamics*, 36:1719–1740, 2007.

[KPY⁺22] Constantinos Kanellopoulos, Nikolaos Psycharis, Han Yang, Boris Jeremić, Ioannis Anastasopoulos, and Božidar Stojadinović. Seismic resonant metamaterials for the protection of an elastic-plastic sdof system against vertically propagating seismic shear waves (SH) in nonlinear soil. *Soil Dynamics and Earthquake Engineering*, 162:107366, 2022.

[KR74] E Kausel and JM Roesset. Soil structure interaction problems for nuclear containment structures. In *Electric power and the civil engineer*. EPRI, 1974.

[KR85] M. Kleiber and B. Raniecki. Elastic – plastic materials at finite strains. In A. Sawczuk and G. Bianchi, editors, *Plasticity Today: Modelling, Methods and Applications*, pages 3–46. Elsevier Applied Sciences Publishers, 1985.

[KR93] Marek Klisinski and Kenneth Runesson. Improved symmetric and non–symmetric solvers for fe calculations. *Advances in Engineering Software*, 18:41–51, 1993.

[KR04] Yong-Seok Kim and Jose M Roesset. Effect of nonlinear soil behavior on inelastic seismic response of a structure. *International Journal of Geomechanics*, 4(2):104–114, 2004.

[KR08] Albert R Kottke and Ellen M Rathje. *Technical manual for Strata*. Pacific Earthquake Engineering Research Center, 2008.

[KR09] Albert R. Kottke and Ellen M. Rathje. Technical manual for strata. Technical report, Pacific Earthquake Engineering Research Center Berkeley, California, 2009.

[KR14] Chinmoy Kolay and James M. Ricles. Development of a family of unconditionally stable explicit direct integration algorithms with controllable numerical energy dissipation. *Earthquake Engineering & Structural Dynamics*, pages n/a–n/a, 2014.

[Kra96] Steven L. Kramer. *Geotechnical Earthquake Engineering*. Prentice Hall, Inc, Upper Saddle River, New Jersey, 1996.

[Kra09] K. Krabenhøft. A variational principle of elastoplasticity and its application to the modeling of frictional materials. *International Journal for Solids and Structures*, 46:464–479, 2009.

[KRB21] Sai Chowdeswara Rao Korlapati, Rahul Raman, and Michel Bruneau. Modeling and test data uncertainty factors used in prior FEMA P695 studies. *Journal of Structural Engineering*, 147(2):06020009, 2021.

[Kre91] Erwin Kreyszig. *Differential Geometry*. Dover Publications, Inc., 1991.

[Kre96] Steen Krenk. Family of invariant stress surfaces. *ASCE Journal of Engineering Mechanics*, 122(3):201–208, 1996.

[Kre06a] Steen Krenk. Energy conservation in Newmark based time integration algorithms. *Computer Methods in Applied Mechanics and Engineering*, 195(44-47):6110–6124, September 2006.

[Kre06b] Steen Krenk. State-space time integration with energy control and fourth-order accuracy for linear dynamic systems. *International Journal for Numerical Methods in Engineering*, 65(5):595–619, 2006.

[Kre09] Steen Krenk. *Non-linear modeling and analysis of solids and structures*. Cambridge University Press, 2009.

[Kre14] Steen Krenk. Global format for energy-momentum based time integration in nonlinear dynamics. *International Journal for Numerical Methods in Engineering*, 100(6):458–476, 2014.

[Kri75] R. D. Krieg. A practical two surface plasticity theory. *ASME Journal of Applied Mechanics*, pages 641–646, September 1975.

[KRJ⁺23] Constantinos Kanellopoulos, Peter Rangelow, Boris Jeremić, Ioannis Anastasopoulos, and Božidar Stojadinović. Linear and nonlinear dynamic structure-soil-structure interaction for nuclear power plants. *Soil Dynamics and Earthquake Engineering*, in Review, 2023.

[KRJ⁺24] Constantinos Kanellopoulos, Peter Rangelow, Boris Jeremić, Ioannis Anastasopoulos, and Božidar Stojadinović. Dynamic structure-soil-structure interaction for nuclear power plants. 2024. https://ssrn.com/abstract=4683938 or http://dx.doi.org/10.2139/ssrn.4683938.

[KRLP11] A. Karrech, K. Regenauer-Lieb, and T. Poulet. Frame indifferent elastoplasticity of frictional materials at finite strain. *International Journal of Solids and Structures*, 48(3-4):397 – 407, 2011.

[Krö60] Ekkehart Kröner. Algemeine kontinuumstheorie der versetzungen und eigenspanungen. *Archive for Rational Mechanics and Analysis*, 4(4):273–334, 1960.

[Krö92] E. Kröner. The internal mechanical state of solids with defects. *Internal Journal of Solids and Structures*, 29(14/15):1849–1852, 1992.

[Krö95] E. Kröner. Displocation in crystals and in continua: A confrontation. *International Journal of Engineering Science*, 33(15):2127–2135, 1995.

[KRP12] S. Kazem, J.A. Rad, and K. Parand. Radial basis functions methods for solving fokker-planck equation. *Engineering Analysis with Boundary Elements*, 36(2):181 – 189, 2012.

[KRS91] Marek Klisinski, Kenneth Runesson, and Stein Sture. Finite element with inner softening band. *Journal of Engineering Mechanics*, 117(3):575–587, March 1991.

[KRS03] Christian Kurmann, Felix Rauch, and Thomas M. Stricker:. Cost/performance tradeoffs in network interconnects for clusters of commodity pcs. In *Proceedings of Workshop on Communication Architecture for Clusters*, Nice, France, April 22 2003.

[Krz93] Jacek Krzypek. *Plasticity and Creep*. Begell House, 1993. ISBN 0-8493-9936-X.

[KS92] B. L. Kutter and N. Sathialingam. Elasti–viscoplastic modelling of the rate–dependent behavior of clays. *Géotechnique*, 42(3):427–441, 1992.

[KS95] Igor Kaljević and Sunil Saigal. Stochastic boundary elements for two–dimensional potential flow in non–homogeneous media. *Computer Methods in Applied Mechanics and Engineering*, 121:211–230, 1995.

[KS97] Igor Kaljević and Sunil Saigal. An improved element free galerkin formulation. *International Journal for Numerical Methods in Engineering*, 40:2953–2974, 1997.

[KS01] Juraj Kralik and Miroslav Simonovic. Earthquake response analysis including site effects of NPP buildings with reactor wwer 440. In *Transactions, SMiRT 16, Washington DC*, August 2001.

[KS05a] A. Der Kiureghian and J. L. Sackman. Tangent geometric stiffness of inclined cables under self-weight. *ASCE Journal of Structural Engineering*, 131(6):941–945, 2005.

[KS05b] Frances Y Kuo and Ian H Sloan. Lifting the curse of dimensionality. *Notices of the AMS*, 52(11):1320–1328, 2005.

[KS08] H.-K. Kim and J.C. Santamarina. Spatial variability: drained and undrained deviatoric load response. *Géotechnique*, 58(10):805–814, 2008.

[KS12] Alexander Konyukhov and Karl Schweizerhof. *Computational Contact Mechanics: geometrically exact theory for arbitrary shaped bodies*, volume 67. Springer Science & Business Media, 2012.

[KS19] Kaan Kaatsiz and Halûk Sucuoĝu. The role of overstrength on the seismic performance of asymmetric-plan structures. *Earthquake Engineering & Structural Dynamics*, 48(4):412–431, 2019.

[KSA20] Danilo Kusanovic, Elnaz Seylabi, and Domniki Asimaki. SEISMO-VLAB: A parallel, object oriented virtual lab for mesoscale seismic wave propagation problems. In *USSD 2020 Annual Conference*, 2020.

[KSA$^+$23] Danilo Kusanovic, Elnaz Seylabi, Peyman Ayoubi, Kien Nguyen, Joaquin Garcia-Suarez, Albert Kottke, and Domniki Asimaki. Seismo-VLAB: An open-source software for soil-structure interaction analyses. *Mathematics*, 11:4530, 11 2023.

[KSBD06] Marc Kham, Jean-François Semblat, Pierre-Yves Bard, and Patrick Dangla. Seismic site–city interaction: main governing phenomena through simplified numerical models. *Bulletin of the Seismological Society of America*, 96(5):1934–1951, October 2006.

[KSH$^+$07] Annie O. L. Kwok, Jonathan P. Stewart, Youssef M. A. Hashash, Neven Matasovic, Robert Pyke, Zhiliang Wang, and Zhaohui Yang. Use of exact solutions of wave propagation problems to guide implementation of nonlinear seismic ground response analysis procedures. *ASCE Journal Geotechnical and Geoenvironmental Engineering*, 133(11):1385–1398, November 2007.

[KSK98] George Karypis, Kirk Schloegel, and Vipin Kumar. *ParMETIS: Parallel Graph Partitioning and Sparse Matrix Ordering Library*. University of Minnesota, 1998.

[KSKJ16] Konstantinos Karapiperis, Kallol Sett, M. Levent Kavvas, and Boris Jeremić. Fokker-planck linearization for non-gaussian stochastic elastoplastic finite elements. *Computer Methods in Applied Mechanics and Engineering*, 307:451–469, 2016.

[KT51] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In Jerzy Neyman, editor, *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481 – 492. University of California Press, July 31 – August 12 1950 1951.

[KT91a] Slobodan B. Kojić and Mihailo D. Trifunac. Earthquake stresses in arch dams. i: Theory and antiplane excitation. *Journal of Engineering Mechanics*, 117(3):532–552, March 1991.

[KT91b] Slobodan B. Kojić and Mihailo D. Trifunac. Earthquake stresses in arch dams. ii: Excitation by SV-, P- and rayleigh waves. *Journal of Engineering Mechanics*, 117(3):553–574, March 1991.

[KT95a] Spyros A. Karamanos and John L. Tassoulas. Effects of external pressure on capacity of tubular beam–columns. *ASCE Journal of Structural Engineering*, 121(11):1620–1628, 1995.

[KT95b] Spyros A. Karamanos and John L. Tassoulas. Tension effects on pressure capacity of tubular members. *ASCE Journal of Structural Engineering*, 121(6):955–963, 1995.

[KT99] Dimitri Komatitsch and Jeroen Tromp. Introduction to the spectral element method for three-dimensional seismic wave propagation. *Geophysical journal international*, 139(3):806–822, 1999.

[KT02] Dimitri Komatitsch and Jeroen Tromp. Spectral-element simulations of global seismic wave propagation-i. validation. *Geophysical Journal International*, 149(2):390–412, 2002.

[KT04] James M. Kelly and Shakhzod M. Takhirov. Analytical and numerical stucy on buckling of elastomeric bearings with various shape factors. Technical Report EERC 2004-03, Earthquake Engineering Research Center, University of California, Berkeley, December 2004.

[KT11] A R Khoei and Mohammadnejad T. Numerical modeling of multiphase fluid flow in deforming porous media: a comparison between two-and three-phase models for seismic analysis of earth and rockfill dams. *Computers and Geotechnics*, 38(2):142–166, 2011.

[KTB97] Loukas F. Kallivokas, Aggelos Tsikas, and Jacobo Bielak. On transient three–dimensional absorbing boundary conditions for the modeling of acoustic scattering from near–surface obstacles. *Journal of Computational Acoustics*, 5(1):117–136, 1997.

[KTD15] S. Krödel, N. Thomé, and C. Daraio. Wide band-gap seismic metastructures. *Extreme Mechanics Letters*, 4:111–117, 2015.

[KU87] H Kishida and M Uesugi. Tests of the interface between sand and steel in the simple shear apparatus. *Geotechnique*, 37(1):45–52, 1987.

[Kuh96] Matthew R. Kuhn. Experimantal measurement of strain gradient effects in granular materials. In Y. K. Lin and T. C. Su, editors, *Proceedings of 11th Conference*, pages 881–884. Engineering Mechanics Division of the American Society of Civil Engineers, May 1996.

[Kuh99] Matthew R. Kuhn. Fabric and deformation in granular materials. In N.P. Jones and Roger G. Ghanem, editors, *the Proceedings of the 13th ASCE Engineering Mechanics Division Specialty Conference*, Johns Hopkins University, Baltimore, June 13-16 1999. CD-ROM.

[Kum24] Krishna Kumar. Geotechnical Parrot Tales (GPT): Harnessing large language models in geotechnical engineering. *Journal of Geotechnical and Geoenvironmental Engineering*, 150(1):02523001, 2024.

[KV08] D. Kolymbas and A. Verrujt. Discussion on: Hydrostatic paradox of saturated media by El Tani. *Géotechnique*, 58(10):835–837, 2008.

[KVB20] Mohsen Kohrangi, Dimitrios Vamvatsikos, and Paolo Bazzurro. Multi-level conditional spectrum-based record selection for ida. *Earthquake Spectra*, 36(4):1976–1994, 2020.

[KvDR$^+$17] Pauline P. Kruiver, Ewoud van Dedem, Remco Romijn, Ger de Lange, Mandy Korff, Jan Stafleu, Jan L. Gunnink, Adrian Rodriguez-Marek, Julian J. Bommer, Jan van Elk, and Dirk Doornhof. An integrated shear-wave velocity model for the Groningen gas field, the Netherlands. *Bulletin of Earthquake Engineerin*, 15(9):3555–3580, 2017.

[KvHRR18] Anssi T. Karttunen, Raimo von Hertzen, J.N. Reddy, and Jani Romanoff. Shear deformable plate elements based on exact elasticity solution. *Computers & Structures*, 200:21 – 31, 2018.

[KW93] D. Kolymbas and W. Wu. Introduction of hypoplasticity. *Modern approaches to plasticity*, pages 213–223, 1993.

[KW96] Hong D. Kang and Kaspar J. Willam. Finite element analysis of discontinuities in concrete. In Y. K. Lin and T. C. Su, editors, *Proceedings of 11th Conference*, pages 1054–1057. Engineering Mechanics Division of the American Society of Civil Engineers, May 1996.

[KW99] Bruce L. Kutter and Daniel W. Wilson. De–liquefaction shock waves. In *U.S–Japan Workshop*, 1999.

[KWC14] Manish Kumar, Andrew S. Whittaker, and Michael C. Constantinou. An advanced numerical model of elastomeric seismic isolation bearings. *Earthquake Engineering & Structural Dynamics*, 43(13):1955–1974, 2014.

[KWC17] Manish Kumar, Andrew S. Whittaker, and Michael C. Constantinou. Extreme earthquake response of nuclear power plants isolated using sliding bearings. *Nuclear Engineering and Design*, 316:9 – 25, 2017.

[KWR^{+}04a] Annie Kammerer, Jiaer Wu, Michael Riemer, Juan Pestana, and Raymond Seed. A new multi-directional direct simple shear testing database. In *13th World Conference on Earthquake Engineering*, number 2083 in ., Vancouver, B.C., Canada, August 2004.

[KWR^{+}04b] Annie Kammerer, Jiaer Wu, Michael Riemer, Juan Pestana, and Raymond Seed. Shear strain development in liquefiable soil under bi-directional loading conditions. In *13th World Conference on Earthquake Engineering*, number 2081 in ., Vancouver, B.C., Canada, August 2004.

[KWRP04] A Kammerer, J Wu, M. F. Riemer, and J. Pestana. Pore pressure development in liquefiable soils under bi-directional loading conditions. In *Proceedings of the 11th International Conference on Soil Dynamics and Earthquake Engineering and the 3rd International Conference on Earthquake Geotechnical Engineering,*, volume 2, pages 697–, 2004.

[KY83] A. J. Kinloch and R. J. Young. *Fracture Behavior of Polymers*. Applied Science Publishers, 1983.

[KYS25] Taegu Kim, Tae Sup Yun, and Hyoung Suk Suh. Can ChatGPT implement finite element models for geotechnical engineering applications? *arXiv,*, 2025. https://arxiv.org/abs/2501.02199.

[KŻ96] T. Kurtyka and M. Żysczkovski. Evolution equations for distortional plastic hardening. *International Journal of Plasticity*, 12(2):191–213, 1996.

[KŽK95] Miloš Kojić, Miroslav Živković, and Aleksandar Kojić. Elastic–plastic analysis of orthotropic multilayered beam. *Computers & Structures*, 57(2):205–211, 1995.

[KZP08] Stavroula Kontoe, Lidija Zdravković, and David M. Potts. An assessment of time integration schemes for dynamic geotechnical problems. *Computers and Geotechnics*, 35(2):253–264, 2008.

[KZP09] Stavroula Kontoe, Lidija Zdravković, and David M. Potts. An assessment of the domain reduction method as an advanced boundary condition and some pitfalls in the use of conventional absorbing boundaries. *International Journal for Numerical and Analytical Methods in Geomechanics*, 33(3):309–330, 2009.

[KZPM11] S Kontoe, L Zdravković, DM Potts, and CO Menkiti. On the relative merits of simple and advanced constitutive models in dynamic analysis of tunnels. *Géotechnique*, 61(10):815–829, 2011.

[LA79] D. B. Larson and G. D. Anderson. Plane shock wave studies of porous geologic media. *Journal of Geophysical Research*, 84(B9):357–363, 1979.

[LA80] D. B. Larson and G. D. Anderson. Plane shock wave studies of westerly granite and nugget sandstone. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, 17(6):357–363, 1980.

[LA01] L. Lehmann and H. Antes. Dynamic structure – soil – structure interaction applying the symmetric galerkin boundary element method (sgbem). *Mechanics Research Communications*, 28(3):297 – 304, 2001.

[LA03] Poul V. Lade and Andrei V. Abelev. Effects of cross anisotropy on three–dimensional behavior of sand. : II volume change behavior and failure. *ASCE Journal of Engineering Mechanics*, 129(2):167–174, February 2003.

[LA20] Maxime Lacour and Norman A. Abrahamson. Stochastic constitutive modeling of elastic-plastic materials with uncertain properties. *Computers and Geotechnics*, 125:103642, 2020.

[Lad88a] Poul V. Lade. Double hardening constitutive model for soils, parameter determination and predictions for two sands. In A. Saada and G. Bianchini, editors, *Constitutive Equations for Granular Non–Cohesive Soils*, pages 367–382. A. A. Balkema, July 1988.

[Lad88b] Poul V. Lade. Effects of voids and volume changes on the behavior of frictional materials. *International Journal for Numerical and Analytical Methods in Geomechanics*, 12:351–370, 1988.

[Lad88c] Poul. V. Lade. Model and parameters for the elastic behavior of soils. In Swoboda, editor, *Numerical Methods in Geomechanics*, pages 359–364, Innsbruck, 1988. Balkema, Rotterdam.

[Lad89] P. V. Lade. Experimental observations of stability, and shear planes in granular materials. *Ingenieur Archiv*, 59:114–123, 1989.

[Lad90] Poul V. Lade. Single–hardening model with application to NC clay. *ASCE Journal of Geotechnical Engineering*, 116(3):394–414, 1990.

[Lad02] Poul V Lade. Instability, shear banding, and failure in granular materials. *International Journal of Solids and Structures*, 39(13):3337–3357, 2002.

[Lad14] Poul V. Lade. Estimating parameters from a single test for the three-dimensional failure criterion for frictional materials. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 140(8), August 2014.

[LADP18] Hao Yuan Liu, Jose Antonio Abell, Andrea Diambra, and Federico Pisano. Modelling the cyclic ratcheting of sands through memory-enhanced bounding surface plasticity. *Geotechnique*, https://doi.org/10.1680/jgeot.17.P.307, 2018.

[Lag88] Joseph-Louis Lagrange. *Œuvres, Tome XI, Mécanique Analytique*, volume 1. Éditions Jacques Gabay, 151 bis rue Saint-Jacques, 75005 Paris, France, quatriéme edition, avec notes de Joseph Bertand et Gaston Darboux edition, 1888. (originally published in 1888, by Gauthier-Villars et Fils, Imprimeurs-Librairies du Bureau des Longitudes at de L'École Polytechnique, Quai des Grands-Augustines, 55, Paris), This edition published in 2006.

[Lag89] Joseph-Louis Lagrange. *Œuvres, Tome XII, Mécanique Analytique*, volume 2. Éditions Jacques Gabay, 151 bis rue Saint-Jacques, 75005 Paris, France, quatriéme edition, avec notes de Joseph Bertand et Gaston Darboux edition, 1889. (originally published in 1889, by Gauthier-Villars et Fils, Imprimeurs-Librairies du Bureau des Longitudes at de L'École Polytechnique, Quai des Grands-Augustines, 55, Paris) This edition published in 2006.

[LAH06] Pengcheng Liu, Ralph J Archuleta, and Stephen H Hartzell. Prediction of broadband ground-motion time histories: Hybrid low/high-frequency method with correlated random source parameters. *Bulletin of the Seismological Society of America*, 96(6):2118–2130, 2006.

[Lam52] Gabriel Lamé. *Leçons sur la Théorie Mathématique de L'Élasticité des Corps Solides*. Éditions Jacques Gabay, 151 bis rue Saint-Jacques, 75005 Paris, France, 1852. (originally published in 1852, by Bachelier, Imprimeur-Libraire du Bureau des Longitudes at de L'École Polytechnique, Quai des Augustines, 55, Paris), This edition published in 2006.

[Lam32] Horace Lamb. *Hydrodynamics*. Cambridge university press, 1932.

[Lam70] T. William Lambe. Braced excavations. In *State–of–the–Art Papers Presented at 1970 Specialty Conference: Lateral Stress in the Ground and Design of Earth–Retaining Structures*, pages 149–218. American Society of Civil Engineers, 1970.

[Lan85] R.S. Langley. A finite element method for the statistics of non-linear random vibration. *Journal of Sound and Vibration*, 101(1):41 – 54, 1985.

[Lan91] H.P. Langtangen. A general numerical solution method for Fokker–Planck equations with applications to structural reliability. *Probabilistic Engineering Mechanics*, 6(1):33–48, 1991.

[Lan01] Walter Landry. Implementing a high performance tensor library. In *Proceedings of the Second Workshop on C++ Template Programming*, 2001. available online: `http://superbeast.ucsd.edu/~landry/FTensor/index.html`.

[Lan03] Walter Landry. Implementing a high performance tensor library. *Scientific Programming*, 11(4):273–290, 2003.

[LANS09] Bibiana Luccioni, Daniel Ambrosinib, Gerald Nurickc, and Izak Snyman. Craters produced by underground explosions. *Computers & Structures*, 87(21-22):1366–1373, November 2009.

[Lar90] Ragnar Larsson. *Numerical Simulation of Plastic Localization*. PhD thesis, Department of Structural Mechanics, Chalmers University of Technology, Göteborg, Sweden, April 1990. Publication 90:5.

[Lau02] TA Laursen. *Computational contact and impact mechanics. 2002*. Springer, Berlin, 2002.

[LB] Jay R. Lund and Joseph P. Byrne. Da Vinci's wire tensile tests: Implications for the discovery of engineering mechanics. Unpublished Paper.

[LB97] Yiwei Li and Ivo Babuška. A convergence analysis of an h–version finite–element method with high–order elements for two–dimensional elasto–plastic problems. *SIAM Journal of Numerical Analysis*, 34(3):998–1036, 1997.

[LB02] Vlado A. Lubarda and David J. Benson. On the numerical algorithm for isotropic-kinematic hardening with the Armstrong-Frederick evolution of the back stress. *Comput. Methods Appl. Mech. Eng.*, 191(33):3583–3596, 2002.

[LB07a] J.F. Labuz and L. Biolzi. Experiments with rock: Remarks on strength and stability issues. *International Journal of Rock Mechanics and Mining Sciences*, 44(4):525–537, June 2007.

[LB07b] Nicolas Luco and Paolo Bazzurro. Does amplitude scaling of ground motion records result in biased nonlinear structural drift responses? *Earthquake Engineering & Structural Dynamics*, 36(13):1813–1835, 2007.

[LB22] Qinghua Lei and Nick Barton. On the selection of joint constitutive models for geomechanics simulation of fractured rocks. *Computers and Geotechnics*, 145:104707, 2022.

[LBA20] Maxime Lacour, Guillaume Bal, and Norman Abrahamson. Dynamic stochastic finite element method using time-dependent generalized polynomial chaos. *International Journal for Numerical and Analytical Methods in Geomechanics*, n/a(n/a):1–14, 2020.

[LBA21] Maxime Lacour, Guillaume Bal, and Norman Abrahamson. Dynamic stochastic finite element method using time-dependent generalized polynomial chaos. *International Journal for Numerical and Analytical Methods in Geomechanics*, 45(3):293–306, 2021.

[LBC05] Jie Li and Jian Bing Chen. Dynamic response and reliability analysis of structures with uncertain parameters. *International Journal for Numerical Methods in Engineering*, 62(2):289–315, 2005.

[LBDMK21] Sijia Li, Michael Brun, Irini Djeran-Maigre, and Sergey Kuznetsov. Benchmark for three-dimensional explicit asynchronous absorbing layers for ground wave propagation and wave barriers. *Computers and Geotechnics*, 131:103808, 2021.

[LBH09] X. Lu, J. P. Bardet, and M. Huang. Numerical solutions of strain localization with nonlocal softening plasticity. *Computer Methods Applied Mechanics and Engineering*, 198:3702–3711, 2009.

[LBH14] Thanh-Nam Le, Jean-Marc Battini, and Mohammed Hjiaj. A consistent 3D corotational beam element for nonlinear dynamic analysis of flexible structures. *Computer Methods in Applied Mechanics and Engineering*, 269:538–565, 2014.

[LBM86] Wing Kam Liu, Ted Belytschko, and A Mani. Probabilistic finite elements for nonlinear structural dynamics. *Computer Methods in Applied Mechanics and Engineering*, 56(1):61–81, 1986.

[LBR04] Chao Li, Ronaldo I. Borja, and Richard A. Regueiro. Dynamics of porous media at finite strain. *Computational Methods in Applied Mechanics and Engineering*, 193:3837–3870, 2004.

[LC87] C. F. Leung and Y. K. Chow. Response of pile groups subjected to lateral loads. *International journal for numerical and analysis methods in geomechanics*, 11:307–314, 1987.

[LC89] Vincent W Lee and Hong Cao. Diffraction of SV waves by circular canyons of various depths. *Journal of engineering mechanics*, 115(9):2035–2056, 1989.

[LC90] Jean Lemaitre and Jean–Louis Chaboche. *Mechanics of Solid Materials*. Cambridge University Press, 1990. ISBN 0 521 47758 1 ; TA405.L3813 1990.

[LC06] Jie Li and Jian-Bing Chen. The probability density evolution method for dynamic response analysis of non-linear stochastic structures. *International Journal for Numerical Methods in Engineering*, 65(6):882–903, 2006.

[LC17a] Arnkjell Løkke and Anil K. Chopra. Direct finite element method for nonlinear analysis of semi-unbounded dam-water-foundation rock systems. *Earthquake Engineering & Structural Dynamics*, 46(8):1267–1285, 2017. EQE-16-0361.R1.

[LC17b] Arnkjell Løkke and Anil K. Chopra. Direct finite element method for nonlinear analysis of semi-unbounded dam-water-foundation rock systems. *Earthquake Engineering & Structural Dynamics*, 46(8):1267–1285, 2017. EQE-16-0361.R1.

[LC18a] Yong Li and Joel P. Conte. Probabilistic performance-based optimum design of seismic isolation for a California high-speed rail prototype bridge. *Earthquake Engineering & Structural Dynamics*, 47(2):497–514, 2018.

[LC18b] Arnkjell Løkke and Anil K. Chopra. Direct finite element method for nonlinear earthquake analysis of 3-dimensional semi-unbounded dam-water-foundation rock systems. *Earthquake Engineering & Structural Dynamics*, 47(5):1309–1328, 2018.

[LC19a] Arnkjell Løkke and Anil K. Chopra. Direct-finite-element method for nonlinear earthquake analysis of concrete dams including dam-water-foundation rock interaction. Technical Report PEER 2019/02, Pacific Earthquake Engineering Research Center, March 2019.

[LC19b] Arnkjell Løkke and Anil K. Chopra. Direct finite element method for nonlinear earthquake analysis of concrete dams: Simplification, modeling, and practical application. *Earthquake Engineering & Structural Dynamics*, 48(7):818–842, 2019.

[LC24] Chin-Ta Lai and Joel P. Conte. Dynamic model of the uc san diego nheri six-degree-of-freedom large high-performance outdoor shake table. *Earthquake Engineering & Structural Dynamics*, n/a(n/a), 2024.

[LCCB88] Wing Kam Liu, Herman Chang, Juin-Shyan Chen, and Ted Belytchko. Arbitrary Langrangian – Eulerian Petrov – Galerkin finite elements for nonlinear continua. *Computer Methods in Applied Mechanics and Engineering*, 68:259–310, 1988. TA345. C6425.

[LCFR08] Fernando Lopez-Caballero and Arezou Modaressi Farahmand-Razavi. Numerical simulation of liquefaction effects on seismic ssi. *Soil Dynamics and Earthquake Engineering*, 28(2):85–98, February 2008.

[LCH$^+$20] Bret Lizundia, C.B. Crouse, Steve Harris, Boris Jeremić, Jonathan P. Stewart, and Michael Valley. A practical guide to soil-structure interaction, FEMA P-2091. Technical Report FEMA P-2091, Applied Technology Council, Redwood City, CA, USA, 2020.

[LCMFR10] F. Lopez-Caballero and A. Modaressi-Farahmand-Razavi. Assessment of variability and uncertainties effects on the seismic response of a liquefiable soil profile. *Soil Dynamics and Earthquake Engineering*, 30(7):600–613, July 2010.

[LCRM07] Fernando Lopez-Caballero, Arezou Modaressi-Farahmand Razavi, and Hormoz Modaressi. Nonlinear numerical method for earthquake site response analysis i-elastoplastic cyclic model and parameter identification strategy. *Bulletin of Earthquake Engineering*, 5(3):303, 2007.

[LCS22] Martí Lloret-Cabot and Daichao Sheng. Assessing the accuracy and efficiency of different order implicit and explicit integration schemes. *Computers and Geotechnics*, 141:104531, 2022.

[LD92] Pierre Léger and Serge Dussault. Seismic-energy dissipation in mdof structures. *Journal of Structural Engineering*, 118(5):1251–1269, 1992.

[LD03] Xiaoye S. Li and James W. Demmel. SuperLU_DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems. *ACM Trans. Mathematical Software*, 29(2):110–140, June 2003.

[LD09] Regis Lebrun and Anne Dutfoy. An innovating analysis of the Nataf transformation from the copula viewpoint. *Probabilistic Engineering Mechanics*, 24(3):312–320, 2009.

[LDD$^+$06] Jingbo Liu, Yixin Du, Xiuli Du, Zhenyu Wang, and Jun Wu. 3D viscous-spring artificial boundary in time domain. *Earthquake Engineering and Engineering Vibration*, 5(1):93–102, 2006.

[LdGV^{+}16] V. Lucas, J.-C. de Golinval, R. C Voicu, M. Danila, R. Gavrila, R. Müller, A. Dinescu, L. Noels, and L. Wu. Propagation of material and surface profile uncertainties on MEMS micro-resonators using a stochastic second-order computational multi-scale approach. *International Journal for Numerical Methods in Engineering*, pages n/a–n/a, 2016. nme.5452.

[LDP96] J. F. Labuz, S. T. Dai, and E. Papamichos. Plane–strain compression of rock–like materials. *International Journal of Rock Mechanics, Mining Sciences & Geomechanics Abstracts*, 33(6):573–584, 1996.

[Lee69] E. H. Lee. Elastic–plastic deformation at finite strains. *Journal of Applied Mechanics*, 36(1):1–6, 1969.

[Lee81] E. H. Lee. Some comments on elasto–plastic analysis. *International Journal of Solids and Structures*, 17:859–872, 1981.

[Lee85] E. H. Lee. Finite deformation effects in plasticity analysis. In A. Sawczuk and G. Bianchi, editors, *Plasticity Today: Modelling, Methods and Applications*, pages 61–74. Elsevier Applied Sciences Publishers, 1985.

[Leg79] Robert F. Legget. Geology and geotechnical engineering. *ASCE Journal of Geotechnical Engineering Division*, 105(GT3):342–391, March 1979.

[Leh85] Th. Lehmann. On a generalized constitutive law in thermoplasticity. In A. Sawczuk and G. Bianchi, editors, *Plasticity Today: Modelling, Methods and Applications*, pages 115–134. Elsevier Applied Sciences Publishers, 1985.

[Leo62] G A Leonards. *Correlations of soil properties*. Penetech Press Publishers, London., 1962.

[Leo10] Mark Leonard. Earthquake fault scaling: Self-consistent relating of rupture length, width, average displacement, and moment release. *Bulletin of the Seismological Society of America*, 100(5A):1971–1988, 2010.

[LF95] Chien–Hung Lin and Richard W. Furlong. Longitudintal steel limits for concrete columns. *ACI Journal*, 92(3):282–287, May–June 1995.

[LF98a] Jeeho Lee and Gregory L. Fenves. A plastic–damage concrete model for earthquake analysis of dams. *Earthquake Engineering and Structural Dynamics*, 27:937–956, 1998.

[LF98b] Jeeho Lee and Gregory L. Fenves. Plastic–damage model for cyclic loading of concrete structures. *ASCE Journal of Engineering Mechanics*, 124(8):892–900, 1998.

[LF98c] Jeeho Lee and Gregory L Fenves. Plastic-damage model for cyclic loading of concrete structures. *Journal of engineering mechanics*, 124(8):892–900, 1998.

[LF01] Jeeho Lee and Gregory L. Fenves. A return–mapping algorithm for plastic–damage models: 3–D and plane stress formulation. *International Journal for Numerical Methods in Engineering*, 50:487–506, 2001.

[LF03] E. Larsson and B. Fornberg. A numerical study of some radial basis function based solution methods for elliptic {PDEs}. *Computers & Mathematics with Applications*, 46(5-6):891 – 902, 2003.

[LF08] A.C. Limache and P.S. Rojas Fredini. A tensor library for scientific computing. In Alberto Cardona, Mario Storti, and Carlos Zuppa, editors, *Proceedings of the Mecánica Computacional*, volume XXVII, pages 2907–2925. Asociación Argentina de Mecánical Computacional, 10-13 November 2008.

[LFG^{+}23] Soledad Le Clainche, Esteban Ferrer, Sam Gibson, Elisabeth Cross, Alessandro Parente, and Ricardo Vinuesa. Improving aircraft performance using machine learning: A review. *Aerospace Science and Technology*, 138:108354, 2023.

[LFO06] C.F. Li, Y.T. Feng, and D.R.J. Owen. Explicit solution to the stochastic system of linear algebraic equations $(\alpha_1 A_1 + \alpha_2 A_2 + \cdots + \alpha_m A_m)x = b$. *Computer Methods in Applied Mechanics and Engineering*, 195(44-47):6560–6576, September 2006.

[LFO^{+}08] C. F. Li, Y. T. Feng, D. R. J. Owen, D. F. Li, and I. M. Davis. A Fourier-Karhunen-Loève discretization scheme for stationary random material properties in SFEM. *International Journal for Numerical Methods in Engineering*, 73(13):1942–1965, 2008.

[LFPM05] A. Lupoi, P. Franchin, P.E. Pinto, and G. Monti. Seismic design of bridges accounting for spatial variability of ground motion. *Earthquake Engineering and Structural Dynamics*, 34:327 348, January 2005.

[LFTT13a] Jianwen Liang, Jia Fu, Maria I Todorovska, and Mihailo D Trifunac. Effects of site dynamic characteristics on soil–structure interaction (ii): Incident p and SV waves. *Soil Dynamics and Earthquake Engineering*, 51:58–76, 2013.

[LFTT13b] Jianwen Liang, Jia Fu, Maria I. Todorovska, and Mihailo D. Trifunac. Effects of the site dynamic characteristics on soil-structure interaction (i): Incident SH-waves. *Soil Dynamics and Earthquake Engineering*, 44(0):27 – 37, 2013.

[LG99] A. Lorenzana and J. A. Garrido. Analysis of the elasto–plastic problem involving finite plastic strain using the boundary element method. *Computers and Structures*, 73:147–159, 1999.

[LG11] Pierre Labbé and Antonio Godoy. Non-linear response to a type of seismic input motion. Technical Report IAEA-TECDOC-1655, UN-IAEA, Vienna, 2011.

[LGBC23] Fanny Lehmann, Filippo Gatti, Michaël Bertin, and Didier Clouteau. FOURIER NEURAL OPERATOR SURROGATE MODEL TO PREDICT 3D SEISMIC WAVES PROPAGATION. *arXiv - PHYS - Geophysics*, 2023.

[LGP25] Sangwoo Lee, Abhinav Gupta, and Giorgio T. Proestos. Simulating experimentally observed nonlinear response of large-scale concrete structure to understand the selection of damping: A case of minor nonlinearities. *Nuclear Engineering and Design*, 439:114098, 2025.

[LGYR05] Roland W. Lewis, David T. Gethin, Xinshe S. Yang, and Ray C. Rowe. A combined finite-discrete element method for simulating pharmaceutical powder tableting. *International Journal for Numerical Methods in Engineering*, 62(7):853–869, 2005.

[LH18] Alexander Lindsay and Kathryn Huff. Moltres: finite element based simulation of molten salt reactors. *Journal of Open Source Software*, 3(21):298, 2018.

[LHKK79] Chuck L Lawson, Richard J. Hanson, David R Kincaid, and Fred T. Krogh. Basic linear algebra subprograms for fortran usage. *ACM Transactions on Mathematical Software (TOMS)*, 5(3):308–323, 1979.

[LHL^{+}09] W.H.K. Lee, B.S. Huang, C.A. Langston, C.J. Lin, C.C. Liu, T.C. Shin, T.L. Teng, and C.F. Wu. Review: progress in rotational ground-motion observations from explosions and local earthquakes in Taiwan. *Bulletin of the Seismological Society of America*, 99(2B):958, 2009.

[LHLL09] Chun-Shi Liu, Bor-Shouh Huang, William H. K. Lee, and Chin-Jen Lin. Observing rotational and translational ground motions at the hgsd station in taiwan from 2007 to 2008. *Bulletin of the Seismological of America*, 99(2B):1228–1236, May 2009.

[LHTT18] Jianwen Liang, Bing Han, Maria I. Todorovska, and Mihailo D. Trifunac. 2D dynamic structure-soil-structure interaction for twin buildings in layered half-space ii: Incident sv-waves. *Soil Dynamics and Earthquake Engineering*, 113:356 – 390, 2018.

[LHVG01] I. N. Lomov, M. Hiltl, O. Yu. Vorobiev, and L. A. Glenn. Dynamic behavior of berea sandstone for dry and water-saturated conditions. *International Journal of Impact Engineering*, 26(1-10):465–474, December 2001.

[LHY96] Meei-Ling Lin, Tsan-Hwei Huang, and Jeng-Chie You. The effects of frequency on damping properties of sand. *Soil Dynamics and Earthquake Engineering*, 15(4):269 – 278, 1996.

[LI97a] Poul V. Lade and Sinan Inel. Rotational kinematic hardening model for sand. In Pietruszczak and Pande, editors, *Numerical Methods in Geomechanics*, pages 9–14. Balkema, Rotterdam, 1997.

[LI97b] Poul V. Lade and Sinan Inel. Rotational kinematic hardening model for sand. part I concept of rotating yield and plastic potential surfaces. *Computers and Geotechnics*, 21(3):183–216, 1997.

[Lin99] Frederik Jan Lingen. A generalized conjugate residual method for the solution of non–symmetric systems of equations with multiple right–hand sides. *International Journal for Numerical Methods in Engineering*, 44:641–656, 1999.

[Lis95] Tadeusz J. Liszka. An introduction to h-p –adaptive finite element method. Technical report, Computational Mechanics Co., Austin, TX, january 31st 1995.

[Lit76] I Littleton. An experimental study of the adhesion between clay and steel. *Journal of terramechanics*, 13(3):141–152, 1976.

[LJ84] A.N. Lin and P.C. Jennings. Effect of embedment on foundation-soil impedances. *Journal of Engineerings Mechanics*, 110(7):1060–1075, July 1984.

[LJ18] SF Lloyd and C Jeong. Adjoint equation-based inverse-source modeling to reconstruct moving acoustic sources in a one-dimensional heterogeneous solid. *Journal of Engineering Mechanics*, 144(9):04018089, 2018.

[LJ24] Stephen Lloyd and Chanseok Jeong. Discretize-then-optimize modeling for dynamic force inversion based on runge-kutta explicit time integration. *Journal of Engineering Mechanics*, 150(4):04024012, 2024.

[LJGP13] W Leng, L Ju, M Gunzburger, and S Price. Manufactured solutions and the verification of three-dimensional stokes ice-sheet models. *The Cryosphere*, 7(1):19–29, 2013.

[LJP95] Poul V. Lade and Marc Joachim Prabucki. Softening and preshearing effects in sand. *Soils and Foundations, Journal of Japanese Geotechnical Society*, 35(4):93–104, 1995.

[LJTT16] Jianwen Liang, Liguo Jin, Maria I. Todorovska, and Mihailo D. Trifunac. Soil-structure interaction for a SDOF oscillator supported by a flexible foundation embedded in a half-space: Closed-form solution for incident plane SH-waves. *Soil Dynamics and Earthquake Engineering*, 90:287 – 298, 2016.

[LJW99] B E Larock, R W Jeppson, and G Z Watters. *Hydraulics of pipeline systems*. CRC press, 1999.

[LJY91] William B. Lamport, James O. Jirsa, and Joseph A. Yura. Strength and behavior of grouted pile–to–sleeve connections. *ASCE Journal of Structural Engineering*, 117(8):2477–2498, August 1991.

[LK69] John Lysmer and Roger L. Kuhlemeyer. Finite dynamic model for infinite media. *Journal of Engineering Mechanics Division, ASCE*, 95(EM4):859–877, 1969.

[LK88a] P. V. Lade and M. K. Kim. Single hardening constitutive model for frictional materials II. yield criterion and plastic work contours. *Computers and Geotechnics*, 6:13–29, 1988.

[LK88b] P. V. Lade and M. K. Kim. Single hardening constitutive model for frictional materials III. comparisons with experiemental data. *Computers and Geotechnics*, 6:31–47, 1988.

[LK92] VW Lee and J Karl. Diffraction of SV waves by underground, circular, cylindrical cavities. *Soil Dynamics and Earthquake Engineering*, 11(8):445–456, 1992.

[LK95] Pould V. Lade and Moon K. Kim. Single hardening constitutive model for soil, rock and concrete. *International Journal of Solids and Structures*, 32(14):1963–1995, 1995.

[LK98] Roland W. Lewis and Amir R. Khoei. Numerical modeling of large deformation in metal powder forming. *International Journal for Computer Methods in Applied Mechanics and Engineering*, 159:291–328, 1998.

[LK04] L. Ling and E. J. Kansa. Preconditioning for radial basis functions with domain decomposition methods. *Math. Comput. Model.*, 40(13):1413–1427, 2004.

[LK15] N Lu and M Khorshidi. Mechanisms for soil-water retention and hysteresis at high suction range. *Journal of Geotechnical and Geoenvironmental Engineering*, 141(8):04015032, 2015.

[LKB$^+$14] Marianna Loli, Jonathan A. Knappett, Michael J. Brown, Ioannis Anastasopoulos, and George Gazetas. Centrifuge modeling of rocking-isolated inelastic rc bridge piers. *Earthquake Engineering & Structural Dynamics*, pages n/a–n/a, 2014.

[LKBG16] M. Luo, C. G. Koh, W. Bai, and M. Gao. A particle method for two-phase flows with compressible air pocket. *International Journal for Numerical Methods in Engineering*, 108(7):695–721, November 2016.

[LKK14] Jin Ho Lee, Jae Kwan Kim, and Jung Han Kim. Nonlinear analysis of soil-structure interaction using perfectly matched discrete layers. *Computers & Structures*, 142(0):28 – 44, 2014.

[LKS$^+$22] Grigorios Lavrentiadis, Nicolas Kuehn, Yousef Bozorgnia Elnaz Seylabi, Xiaofeng Meng, Christine Goulet, and Albert Kottke. Non-ergodic methodology and modeling tools. Technical Report GIRS-2022-04 ; DOI: 10.34948/N35P4Z, Natural Hazards Risk & Resiliency Research Center; The B. John Garrick Institute for the Risk Sciences, UCLA, University of California, Los Angeles, August 2022.

[LKvBW10] Kevin Long, Robert Kirby, and Bart van Bloemen Waanders. Unified embedded parallel finite element computations via software-based frechet differentiation. *SIAM Journal on Scientific Computing*, 32(6):3323–3351, 2010.

[LL67] E. H. Lee and D. T. Liu. Finite–strain elastic–plastic theory with application to plane–wave analysis. *Journal of Applied Physics*, 38(1):19–27, January 1967.

[LL81] V. A. Lubarda and E. H. Lee. A correct definition of elastic and plastic deformation and its computational significance. *Journal of Applied Mechanics*, 48:35–40, 1981.

[LL01] Hubert K. Law and Ignatius P. Lam. Application of periodic boundary for large pile group. *Journal of Geotechnical and Geoenvironmental Engineering*, 127(10):889–892, Oct. 2001.

[LL04a] Jonas Larsson and Ragnar Larsson. Non-linear analysis of nearly saturated porous media: theoretical and numerical formulation. *Computational Methods in Applied Mechanics and Engineering*, 191:3885–3907, 2004.

[LL04b] Ning Lu and William J. Likos. *Unsaturated Soil Mechanics*. John Wiley & Sons, May 2004.

[LL08] Huabei Liu and Hoe I Ling. Constitutive description of interface behavior including cyclic loading and particle breakage within the framework of critical state soil mechanics. *International journal for numerical and analytical methods in geomechanics*, 32(12):1495–1514, 2008.

[LL16] Jie Li and Yung Y. Liu. Thermal modeling of a vertical dry storage cask for used nuclear fuel. *Nuclear Engineering and Design*, 301:74–88, 2016.

[LL17] J. Enrique Luco and Armando Lanzi. Optimal Caughey series representation of classical damping matrices. *Soil Dynamics and Earthquake Engineering*, 92:253 – 265, 2017.

[LL21] B.T. Lester and K.N. Long. A constitutive model for glass-ceramic materials. *Mechanics of Materials*, 158, April 2021.

[LLGW19] Chao Luo, Menglin Lou, Guoqing Gui, and Hao Wang. A modified domain reduction method for numerical simulation of wave propagation in localized regions. *Earthquake Engineering and Engineering Vibra*, 18:35–52, 2019.

[LLL16] B. Lindsey, M. Leslie, and W. Luk. A domain specific language for accelerated multilevel monte carlo simulations. In *2016 IEEE 27th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, pages 99–106, July 2016.

[LLLL14] Qipeng Liu, Xiaoyu Liu, Xikui Li, and Shihai Li. Micro-macro homogenization of granular materials based on the average-field theory of Cosserat continuum. *Advanced Powder Technology*, 25(1):436 – 449, 2014.

[LLLO21] Ning Lu, William J. Likos, Shengmin Luo, and Hyunjun Oh. Is the conventional pore water pressure concept adequate for fine-grained soils in geotechnical and geoenvironmental engineering? *Journal of Geotechnical and Geoenvironmental Engineering*, 147(10):02521001, 2021.

[LLMS18] Liang Liang, Minliang Liu, Caitlin Martin, and Wei Sun. A deep learning approach to estimate stress distribution: a fast and accurate surrogate of finite-element analysis. *Journal of The Royal Society Interface*, 15(138):20170844, 2018.

[LLP16] Zhangjun Liu, Wei Liu, and Yongbo Peng. Random function based spectral representation of stationary and non-stationary stochastic processes. *Probabilistic Engineering Mechanics*, 45:115–126, 2016.

[LLP17] Zhangjun Liu, Zixin Liu, and Yongbo Peng. Dimension reduction of Karhunen-Loève expansion for simulation of stochastic processes. *Journal of Sound and Vibration*, 408:168–189, 2017.

[LLP23] Wing Kam Liu, Shaofan Li, and Harold S. Park. Eighty years of the finite element method: Birth, evolution, and future. *Archives of Computational Methods in Engineering*, 30(5):4431–4453, June 2023.

[LLRZ18] Zixin Liu, Zhangjun Liu, Xinxin Ruan, and Qi Zhang. Spectral representation-based dimension reduction for simulating multivariate non-stationary ground motions. *Soil Dynamics and Earthquake Engineering*, 114:313–325, 2018.

[LLTO14a] VW Lee, WY Liu, MD Trifunac, and N Orbović. Scattering and diffraction of earthquake motions in irregular elastic layers, i: Love and SH waves. *Soil Dynamics and Earthquake Engineering*, 66:125–134, 2014.

[LLTO14b] VW Lee, WY Liu, MD Trifunac, and N Orbović. Scattering and diffraction of earthquake motions in irregular, elastic layers, ii: Rayleigh and body P and SV waves. *Soil Dynamics and Earthquake Engineering*, 66:220–230, 2014.

[LLV16] P Li, T Li, and S K Vanapalli. Influence of environmental factors on the wetting front depth: A case study in the loess plateau. *Engineering Geology*, 214:1–10, 2016.

[LLZL20] Shutong Liu, Peizhen Li, Wenyang Zhang, and Zheng Lu. Experimental study and numerical simulation on dynamic soil-structure interaction under earthquake excitations. *Soil Dynamics and Earthquake Engineering*, 138:106333, 2020.

[LM98a] Edward A. Lee and David G. Messerschmitt. Engineering an education for the future. *IEEE Computer*, 31(1):77–85, 1998.

[LM98b] D.E. Lehman and J.P. Moehle. Seismic performance of well–confined concrete bridge columns. Technical report, Pacific Earthquake Engineering Research Center, 1998.

[LM99] Edward A. Lee and David G. Messerschmitt. A highest education in the year 2049. *Proceedings of the IEEE*, 87(9):1685–1691, September 1999.

[LM01] Christopher M Linton and Philip McIver. *Handbook of mathematical techniques for wave/structure interactions*. CRC Press, 2001.

[LMA20] Maxime Lacour, Jorge Macedo, and Norman A. Abrahamson. Stochastic finite element method for non-linear material models. *Computers and Geotechnics*, 125:103641, 2020.

[LMK96] V. A. Lubarda, S. Mastilovic, and J. Knapp. Some comments on plasticity postulates and non–associative flow rules. *International Journal of Mechanical Sciences*, 38(3):247–258, 1996.

[LMTS96] R. W. Lewis, K. Morgan, H. R. Thomas, and K. N. Seetharamu. *The Finite Element Method in Heat Transfer Analysis*. John Wiley, 1996. ISBN 0 471 93424 0.

[LMW83] E. H. Lee, R. L. Mallet, and T. B. Wertheimer. Stress analysis for anysotropic hardening in finite–deformation plasticity. *ASME Journal of Applied Mechanics*, 50:554–560, September 1983.

[LMX+18] Jie Liu, Xianghua Meng, Can Xu, Dequan Zhang, and Chao Jiang. Forward and inverse structural uncertainty propagations under stochastic variables with arbitrary probability distributions. *Computer Methods in Applied Mechanics and Engineering*, 342:287 – 320, 2018.

[LN84] Poul V. Lade and Richard B. Nelson. Incrementalization procedure for elasto–plastic constitutive model with multiple, intersecting yield surfaces. *International Journal for Numerical and Analytical Methods in Geomechanics*, 8:311–323, 1984.

[LN95] Xiaoshan Lin and Tang–Tat Ng. Contact detection algorithms for three–dimensional ellipsoids in discrete element modeling. *International Journal for Numerical and Analytical Methods in Geomechanics*, 19:653–659, 1995.

[LN97] X. Lin and T.-T. Ng. A three dimensional discrete element model using arrays of ellipsoids. *Géotechnique*, 47(2):319–329, 1997.

[LNN94] M. L. Lings, C. W. W. Ng, and D. F. T. Nash. The lateral pressure of wet concrete in diaphragm wall panels cast under bentonite. *Proceedings of Institution of Civil Engineers, Geotechnical Engineering*, 107:163–172, July 1994.

[LOC24] Burigede Liu, Michael Ortiz, and Fehmi Cirak. Towards quantum computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 432:117403, 2024.

[LOO08] L.J. Lucas, H. Owhadi, and M. Ortiz. Rigorous verification, validation, uncertainty quantification and certification through concentration-of-measure inequalities. *Computer Methods in Applied Mechanics and Engineering*, 197(51-52):4591–5609, October 2008.

[Lor88] B. Loret. Some effects of microstructure on the overall behaviour of granular materials. In A. Saada and G. Bianchini, editors, *Constitutive Equations for Granular Non–Cohesive Soils*, pages 717–721. A. A. Balkema, July 1988.

[Lor08] E Lorentz. A mixed interface finite element for cohesive zone models. *Computer Methods in Applied Mechanics and Engineering*, 198(2):302–317, 2008.

[LORW12] Anders Logg, Kristian B Olgaard, Marie E Rognes, and Garth N Wells. Ffc: the fenics form compiler. *Automated Solution of Differential Equations by the Finite Element Method*, pages 227–238, 2012.

[LOT$^+$88] J. Lysmer, F. Ostadan, M. Tabatabaie, S. Vahdani, and F. Tajirian. *SASSI - A System for Analysis of Soil-Structure Interaction, Theoretical Manual*. University of California, Berkeley, 1988.

[Lov44] A. E. H. Love. *A Treatise of the Mathematical Theory of Elasticity*. Dover Publications, Inc., 1944.

[Lov71] David Lovelock. The Einstein tensor and its generalizations. *Journal of Mathematical Physics*, 12(3):498–501, 1971.

[LP91] Par J. Leray and A. Pecker. Calcul explicite due déplacement ou de la tension du demi-plan élastique isotrope et honogêne soumis a un choc en son bord. *J. Math. et appl.*, 70:489–511, 1991.

[LP04] Jia Lua and Panayiotis Papadopoulos. A covariant formulation of anisotropic finite plasticity: theoretical developments. *Computer Methods in Applied Mechanics and Engineering*, 193(48-51):5339–5358, December 2004.

[LPD$^+$10] F. Laouafa, F. Prunier, A. Daouadji, H. Al Gali, and F. Darve. Stability in geomechanics, experimental and numerical analyses. *International Journal for Numerical and Analysis Methods in Geomechanics*, Early View DOI: 10.1002/nag.996, 2010.

[LPDR05] P. Ladevéze, G. Puel, A. Deraemaeker, and T. Romeuf. Validation of structural dynamics models containing uncertainties. *Computer Methods in Applied Mechanics and Engineering*, 195(4-6):373–393, 2005.

[LPL] Choonghyun Lee, Duhee Park, and Yong-Gook Lee. Numerical estimation of seismic earth pressure on walls of basements underlain by bedrock. *Earthquake Engineering & Structural Dynamics*, n/a(n/a).

[LPM05] Frédéric Légeron, Patrick Paultre, and Jacky Mazars. Damage mechanics modeling of nonlinear seismic behavior of concrete structures. *ASCE Journal of Structural Engineering*, 131(6):946–955, 2005.

[LPM$^+$15] Pierre Labbé, Pierre Pegon, Javier Molina, Christian Gallois, and Daniéle Chauvel. The SAFE experimental research on the frequency dependence of shear wall seismic design margins. *Journal of Earthquake Engineering*, pages 1–28, 2015. DOI: 10.1080/13632469.2015.1038370.

[LQ99] Zhao Lei and Chen Qiu. Neumann dynamic stochastic finite element method of vibration for structures with stochastic parameters to random excitation. *Computers and Structures*, 77:651–657, December 1999.

[LR56] Peter D Lax and Robert D Richtmyer. Survey of the stability of linear finite difference equations. *Communications on pure and applied mathematics*, 9(2):267–293, 1956.

[LR93] Ragnar Larsson and Kenneth Runesson. Discontinuous displacement approximation for capturing plastic localization. *International Journal for Numerical Methods in Engineering*, 36:2087–2105, 1993.

[LR96a] Ragnar Larsson and Kenneth Runesson. Element embedded localization band based on regularized displacement discontinuity. *ASCE, Journal of Engineering Mechanics*, 122(5):402–411, may 1996.

[LR96b] Ragnar Larsson and Kenneth Runesson. Implicit integration and consistent linearization for yield criteria of the Mohr–Coulomb type. *International Journal of Mechanics of Cohesive-Frictional Materials*, 1(4):367–383, 1996.

[LR97a] B. Loret and E. Rizzi. Anisotropic stiffness degradation triggers onset of strain localization. *International Journal of Plasticity*, 13(5):447–459, 1997.

[LR97b] B. Loret and E. Rizzi. Qualitative analysis of strain localization part II: Transversely isotropic elasticity and plasticity. *International Journal of Plasticity*, 13(5):501–519, 1997.

[LR04] Fredrik Larsson and Kenneth Runesson. Modeling and discretization errors in hyperelasto-(visco-)plasticity with a view to hierarchical modeling. *Computer Methods in Applied Mechanics and Engineering*, 193(48-51):5283–5300, December 2004.

[LR07] Luis M. Lacomaa and Ignacio Romero. Error estimation for the HHT method in non-linear solid dynamics. *Computers & Structures*, 85(3-4):158–169, February 2007.

[LRK78] W. Michael Lai, David Rubin, and Erhard Krempl. *Introduction to Continuum Mechanics*, volume 17 of *Pergamon Unified Engineering Series*. Pergamon Press, 1978.

[LRO$^+$13] Antonia Larese, Riccardo Rossi, Eugenio Oñate, Miguel Àngel Toledo, Rafael Moránn, and Hibber Campos. Numerical and experimental study of overtopping and failure of rockfill dams. *International Journal of Geomechanics*, doi:10.1061/(ASCE)GM.1943-5622.0000345, 2013.

[LRS96] R. Larsson, K. Runesson, and S. Sture. Embedded localization band in undrained soil based on regularized strong discontinuity–theory and FE–analysis. *International Journal of Solids and Structures*, 33(20-22):3081–3101, 1996.

[LS67] Kenneth L Lee and H Bolton Seed. Drained strength characteristics of sands. *Journal of Soil Mechanics & Foundations Div*, 1967.

[LS75] M.E. Lesk and E. Schmidt. Lex - a lexical analyzer generator. Technical Report UNIX TIME-SHARING SYSTEM:UNIX PROGRAMMER'S MANUAL, Seventh Edition, Volume 2B, AT&T Bell Laboratories, Murray Hill, New Jersey 07974, 1975.

[LS93a] TA Laursen and JC Simo. Algorithmic symmetrization of coulomb frictional problems using augmented lagrangians. *Computer methods in applied mechanics and engineering*, 108(1-2):133–146, 1993.

[LS93b] TA Laursen and JC Simo. A continuum-based finite element formulation for the implicit solution of multibody, large deformation-frictional contact problems. *International Journal for numerical methods in engineering*, 36(20):3451–3485, 1993.

[LS94] V. A. Lubarda and C. F. Shih. Plastic spin and related issues in phenomenological plasticity. *Transactions of ASME Journal of Applied Mechanics*, 61:524–529, 1994.

[LS99] Roland W. Lewis and B. A. Schrefler. *The Finite Element Method in the Static and Dynamic Deformation and Consolidation of Porous Media*. John Wiley & Sons, 2nd edition edition, January 1999.

[LS02a] SH Liu and DA Sun. Simulating the collapse of unsaturated soil by dem. *International journal for numerical and analytical methods in geomechanics*, 26(6):633–646, 2002.

[LS02b] David A. Lockner and Sergei A. Stanchits. Undrained poroelastic response of sandstones to deviatoric stress change. *JOURNAL OF GEOPHYSICAL RESEARCH*, 107(B12):ETG 13–1 – ETG 13–14, 2002. doi:10.1029/2001JB001460.

[LS06] Suchart Limkatanyu and Enrico Spacone. Frame element with lateral deformable supports: Formulations and numerical validation. *Computers & Structures*, 84(13-14):942–954, May 2006.

[LS25] Shijin Li and Alister Smith. Pipeline-soil interaction behavior: Acoustic emission and energy dissipation. *Journal of Geotechnical and Geoenvironmental Engineering*, 151(2):04024157, 2025.

[LSBL99] J. H. Lee, R. Salgado, A. Bernal, and C. W. Lovell. Shredded tires and rubber–sand as lightweight backfill. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 125(2):132–141, February 1999.

[LSJ23] Stephen Lloyd, Christoph Schaal, and Chanseok Jeong. Inverse modeling and experimental validation for reconstructing wave sources on a 2D solid from surficial measurement. *Ultrasonics*, 128:106880, 2023.

[LSK04] D. Lucor, C.-H. Su, and G. E. Karniadakis. Generalized polynomial chaos and random oscillators. *Int. J. Numer. Meth. Engng*, 60(0):571–596, August 2004.

[LSL06] Huabei Liu, Erxiang Song, and Hoe I Ling. Constitutive modeling of soil-structure interface through the concept of critical state soil mechanics. *Mechanics Research Communications*, 33(4):515–531, 2006.

[LSW03] SH Liu, DA Sun, and Yisen Wang. Numerical study of soil collapse behavior by discrete element modelling. *Computers and Geotechnics*, 30(5):399–408, 2003.

[LT05] Xikui Li and Hongxiang Tang. A consistent return mapping algorithm for pressure-dependent elastoplastic Cosserat continua and modelling of strain localisation. *Computers and Structures*, 83(1):1 – 10, 2005.

[LT10] Vincent W. Lee and Mihailo D. Trifunac. Should average shear-wave velocity in the top 30 m of soil be used to describe seismic amplification? *Soil Dynamics and Earthquake Engineering*, 30(11):1250–1258, November 2010.

[LTC21] Hao Liu, Zhenyun Tang, and Yanjiang Chen. Earthquake input energy prediction considering structural transient response. *Soil Dynamics and Earthquake Engineering*, 144:106661, 2021.

[LTM01] Patrick Le Tallec and Jean Mouro. Fluid structure interaction with large structural displacements. *Computer methods in applied mechanics and engineering*, 190(24):3039–3067, 2001.

[Lu16] N Lu. Generalized soil water retention equation for adsorption and capillarity. *Journal of Geotechnical and Geoenvironmental Engineering*, 142(10):04016051, 2016.

[Lub72a] J. Lubliner. On the thermodynamic foundations of non–linear solid mechanics. *International Journal of Non–Linear Mechanics*, 7:237–254, 1972.

[Lub72b] J. Lubliner. On the thermodynamic foundations of non-linear solid mechanics. *International Journal of Non-Linear Mechanics*, 7:237–254, 1972.

[Lub90] Jacob Lubliner. *Plasticity Theory*. Macmillan Publishing Company, New York., 1990.

[Lub91] V. A. Lubarda. Constitutive analysis of large elasto–plastic deformation based on the multiplicative decomposition of deformation gradient. *International Journal of Solids and Structures*, 27(7):885–895, 1991.

[Lub94] V. A. Lubarda. An analysis of large–strain damage elastoplasticity. *International Journal of Solids and Structures*, 31(21):2951–2964, 1994.

[Luc74] J. Enrique Luco. Impedance functions for a rigid foundation on a layered medium. *Nuclear Engineering and Design*, 31:204–217, 1974.

[Luc76] JE Luco. Torsional response of structures for sh waves: the case of hemispherical foundations. *Bulletin of the Seismological Society of America*, 66(1):109–123, 1976.

[Lue84] David G. Luenberger. *Linear and Nonlinear Programming*. Addison – Wesley Publishing Company, second edition, 1984.

[Luo86] MINH PHONG Luong. Characteristic threshold and infrared vibrothermography of sand. *Geotechnical Testing Journal*, 9(2):80–86, 1986.

[LW72] John Lysmer and Günter Waas. Shear waves in plane infinite structures. *Journal of the Engineering Mechanics Division*, 98(1):85–105, 1972.

[LW73] T.H. Lee and D.A. Wesley. Soil-structure interaction of nuclear reactor structures considering through-soil coupling between adjacent structures. *Nuclear Engineering and Design*, 24(3):374 – 387, 1973.

[LW79] William T. Lambe and Robert V. Whitman. *Soil Mechanics, SI Version*. John Wiley and Sons, 1979.

[LW82] JE Luco and HL Wong. Response of structures to nonvertically incident seismic waves. *Bulletin of the Seismological Society of America*, 72(1):275–302, 1982.

[LW87] JE Luco and HL Wong. Seismic response of foundations embedded in a layered half-space. *Earthquake engineering & structural dynamics*, 15(2):233–247, 1987.

[LW09] Danuta Lesniewska and David Muir Wood. Observations of stresses and strains in a granular material. *ASCE Journal of Engineering Mechanics*, 135(9):1038–1054, September 2009.

[LW10] Anders Logg and Garth N Wells. Dolfin: Automated finite element computing. *ACM Transactions on Mathematical Software (TOMS)*, 37(2):20, 2010.

[LWCZ11] Menglin Lou, Huaifeng Wang, Xi Chen, and Yongmei Zhai. Structure–soil–structure interaction: literature review. *Soil Dynamics and Earthquake Engineering*, 31(12):1724–1731, 2011.

[LWM99] C. H. Liu, J. Y. Wong, and H. A. Mang. Large strain finite element analysis of sand: model, algorithm and application to numerical simulations of tire–sand interaction. *Computers and Structures*, 159:253–265, 1999.

[LWV^{+}] Kaivalya M. Lal, Andrew S. Whittaker, Shahriar Vahdani, Benjamin D. Kosbab, and Koroush Shirvan. Seismically isolated nuclear power plants: Is soil-structure-interaction analysis needed? *Earthquake Engineering & Structural Dynamics*, n/a(n/a).

[LXG^{+}15] Xinzheng Lu, Linlin Xie, Hong Guan, Yuli Huang, and Xiao Lu. A shear wall element for nonlinear seismic analysis of super-tall buildings using opensees. *Finite Elements in Analysis and Design*, 98:14–25, 2015.

[LY88] C.H. Loh and Y.T. Yeh. Spatial variation and stochastic modeling of seismic differential ground movement. *Earthquake Engineering and Structural Dynamics*, 16:583–596, 1988.

[LY92] Pould V. Lade and Jerry A. Yamamuro. Stability of granular materials in postpeak softening regime. *ASCE Journal of Engineering Mechanics*, 119:128–144, 1992.

[LY96] Poul V. Lade and Jerry A. Yamamuro. Undrained sand behavior in axisymmetric tests at high pressures. *ASCE Journal of Geotechnical Engineering*, 122(2):120–129, 1996.

[LY97] Poul V. Lade and Jerry A. Yamamuro. Effects of nonplastic fines on static liquefaction of sands. *Canadian Geotechnical Journal*, 34:918–928, 1997.

[LY98] Haowu Liu and Zhaohui Yang. Distributed optical fiber sensing of cracks in concrete. *SPIE Journal*, 1998. to appear.

[LYB96] Poul V. Lade, Jerry A. Yamamuro, and Paul A. Bopp. Significance of particle crushing in granular materials. *ASCE Journal of Geotechnical Engineering*, 122(4):309–316, 1996.

[Lys70] P. C. Lysne. A comparison of calculated and measured low-stress Hugoniots and release adiabats of dry and water-saturated tuff. *Journal of Geophysical Research*, 75(23):4375–4386, 1970.

[Lys88] J. Lysmer. SASSI: A computer program for dynamic soil structure interaction analysis. *Report UBC/GT81-02. University of California, Berkeley, CA, USA.*, 1988.

[LZ06] S. Liao and A. Zerva. Physically compliant, conditionally simulated spatially variable seismic ground motions for performance–based design. *Earthquake Engineering & Structural Dynamics*, 35:891–919, 2006.

[LZ19] Ning Lu and Chao Zhang. Soil sorptive potential: Concept, theory, and verification. *Journal of Geotechnical and Geoenvironmental Engineering*, 145(4):04019006, 2019.

[LZaWS23] Yuchen Liao, Ruiyang Zhang, G ang Wu, and Hao Sun. A frequency-based ground motion clustering approach for data-driven surrogate modeling of bridges. *Journal of Engineering Mechanics*, 149(9):04023069, 2023.

[LZF$^+$21] Jian-Ping Li, Xiao-Lei Zhang, Shi-Jin Feng, Zhang-Long Chen, and Yi-Cheng Li. Numerical investigation of ground-borne vibration mitigation by infilled trenches in a poroelastic half-space considering the moving water table. *International Journal of Geomechanics*, 21(10):04021187, 2021.

[LZK14] Jingmao Liu, Degao Zou, and Xianjing Kong. A three-dimensional state-dependent model of soil–structure interface for monotonic and cyclic loadings. *Computers and Geotechnics*, 61:166–177, 2014.

[LZKJ19a] Cuihua Li, Changhai Zhai, Sashi Kunnath, and Duofa Ji. Methodology for selection of the most damaging ground motions for nuclear power plant structures. *Soil Dynamics and Earthquake Engineering*, 116:345 – 357, 2019.

[LZKJ19b] Cuihua Li, Changhai Zhai, Sashi Kunnath, and Duofa Ji. Methodology for selection of the most damaging ground motions for nuclear power plant structures. *Soil Dynamics and Earthquake Engineering*, 116:345–357, 2019.

[LZL$^+$20] Yadong Li, Sen Zheng, Weili Luo, Jie Cui, and Qianran Chen. Design and performance of a laminar shear container for shaking table tests. *Soil Dynamics and Earthquake Engineering*, 135:1–6, 2020.

[M.93] SOBOL I. M. Sensitivity analysis for non-linear mathematical models. *Math. Modeling Comput. Exp.*, 1, 1993.

[M.A56] M.A.Biot. Theory of propagation of elastic waves in a fluid-saturated porous solid. low-frequency range. *The journal of acoustical society of America*, 28(2):168–178, March 1956.

[MA04a] K. J. McManus and D. Alabaster. Constant force shaking of a group of four drilled shafts. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 130(2):123–128, 2004.

[MA04b] Christian Miehe and Nikolas Apel. Anisotropic elastic-plastic analysis of shells at large strains. a comparison of multiplicative and additive approaches to enhanced finite element design and constitutive modelling. *International Journal for Numerical Methods in Engineering*, 61(12):2067 – 2113, 2004.

[MA06] E Miranda and SD Akkar. Generalized interstory drift spectrum. *Journal of Structural Engineering*, 132(6):840–852, 2006.

[MA14] Muiris C. Moynihan and Julian M. Allwood. Utilization of structural steel in buildings. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 470(2168):20140170, 2014.

[MÁA$^+$19] Cristina Medina, Guillermo M. Álamo, Juan J. Aznárez, Luis A. Padrón, and Orlando Maeso. Variations in the dynamic response of structures founded on piles induced by obliquely incident sv waves. *Earthquake Engineering & Structural Dynamics*, 0(0):1–18, 2019.

[Mac91] R. I. Mackie. Object – oriented programming and numerical methods. *Microcomputers in CIvil Engineering*, 6:123–128, 1991.

[Mac92] Jaroslav Mackerle. Finite and boundary element methods in biomechanics: a bibliography (1976-1991). *Engineering Computations*, 9:403–435, 1992.

[Mac95] Emir J. Macari–Pasqualino. Closure: Response predictions of granular materials at low effective stresses. *Journal of Geotechnical Engineering*, 121(9):680, September 1995.

[Mac00] Jaroslav Mackerle. Object–oriented techniques in FEM and BEM a bibliography (1996-1999). *Finite Elements in Analysis and Design*, 36:189–196, 2000.

[MAC$^+$20] R. Martineau, D. Andrs, R. Carlsen, D. Gaston, J. Hansel, F. Kong, A. Lindsay, C. Permann, A. Slaughter, E. Merzari, Rui Hu, A. Novak, and R. Slaybaugh. Multiphysics for nuclear energy applications using a cohesive computational framework. *Nuclear Engineering and Design*, 367:110751, 2020.

[MAD$^+$21] Panagiotis Martakis, Giulia Aguzzi, Vasilis K. Dertimanis, Eleni N. Chatzi, and Andrea Colombi. Nonlinear periodic foundations for seismic protection: Practical design, realistic evaluation and stability considerations. *Soil Dynamics and Earthquake Engineering*, 150:106934, 2021.

[MAEA$^+$14] T. Mourabit, K. M. Abou Elenean, A. Ayadi, D. Benouar, A. Ben Suleman, M. Bezzeghoud, A. Cheddadi, M. Chourak, M. N. ElGabry, A. Harbi, M. Hfaiedh, H. M. Hussein, J. Kacem, A. Ksentini, N. Jabour, A. Magrin, S. Maouche, M. Meghraoui, F. Ousadou, G. F. Panza, A. Peresan, N. Romdhane, F. Vaccari, and E. Zuccolo. Neo-deterministic seismic hazard assessment in North Africa. *Journal of Seismology*, 18(2):301–318, Apr 2014.

[Mag12] Andrea Magrin. *MULTI-SCALE SEISMIC HAZARD SCENARIOS*. PhD thesis, UNIVERSITÀ DEGLI STUDI DI TRIESTE, 2012.

[Mah95] R. Mahnken. A newton–multigrid algorithm for elasto-plastic / viscoplastic problems. *Computational Mechanics*, 15(5):408–425, 1995.

[Mah99] Rolf Mahnken. A comprehensive study of a multiplicative elastoplasticity moel coupled to damage including parameter identification. *Computers and Structures*, 73:179–200, 1999.

[MAJC03] G.R. Miller, P. Arduino, J. Jang, and C. Choi. Localized tensor-based solvers for interactive finite element applications using C++ and Java. *Computers and Structures*, 81:423–437, 2003.

[Mak96a] M. Maksimović. The shear strength components of a rough rock joint. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, 33(8):769–783, 1996.

[Mak96b] Milan Maksimović. A family of nonlinear failure envelopes. *Electronic Journal of Geotechnical Engineering*, October 1996. available at http://139.78.66.61/ejge/.

[Mak18] Nicos Makris. Seismic isolation: Early history. *Earthquake Engineering & Structural Dynamics*, 0(0):1–16, 2018. early view, https://doi.org/10.1002/eqe.3124.

[Mak19] Nicos Makris. Seismic isolation: Early history. *Earthquake Engineering & Structural Dynamics*, 48(2):269–283, 2019.

[Mal69] Lawrence E. Malvern. *Introduction to the Mechanics of a Continuous Medium*. In Engineering of the Physical Sciences. Prentice Hall Inc., 1969.

[MAL02] C. Miehe, N. Apel, and M. Lambrecht. Anisotropic additive plasticity in the logarithmic strain space: modular kinematic formulation and implementation based on incremental minimization principles for standard materials. *Computer Methods in Applied Mechanics and Engineering*, 191:5383–5425, 2002.

[Mal21] Praveen K. Malhotra. *Seismic Analysis of Structures and Equipment*. Springer, 2021. https://doi.org/10.1007/978-3-030-57858-9.

[MAMH11] Christopher R. McGann, Pedro Arduino, and Peter Mackenzie-Helnwein. Applicability of conventional p-y relations to the analysis of piles in laterally spreading soil. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 137(6):557–567, 2011.

[MAMH12] Christopher R. McGann, Pedro Arduino, and Peter Mackenzie-Helnwein. Stabilized single-point 4-node quadrilateral element for dynamic analysis of fluid saturated porous media. *Acta Geotechnica*, 7:297–311, 2012.

[Man65] J. Mandel. Generalisation de la theorie de plasticité de W. T. Koiter. *International Journal for Solids and Structures*, 1:273–295, 1965. in French.

[Man01a] Gaetano Manfredi. Evaluation of seismic energy demand. *Earthquake Engineering & Structural Dynamics*, 30(4):485–499, 2001.

[Man01b] Gaetano Manfredi. Evaluation of seismic energy demand. *Earthquake Engineering & Structural Dynamics*, 30(4):485–499, 2001.

[Man02] George D. Manolis. Stochastic soil dynamics. *Soil Dynamics and Earthquake Engineering*, 22:3–15, 2002.

[Man04] Majid T. Manzari. Application of micropolar plasticity to post failure analysis in geomechanics. *International Journal for Numerical and Analytical Methods in Geomechanics*, 28(10):1011–1032, 2004.

[Mao05] Jianqing Mao. A finite element approach to solve contact problems in geotechnical engineering. *International Journal for Numerical and Analytical Methods in Geomechanics*, 29:525–550, 2005.

[Mar73] J. Marsden. On product formulas for nonlinear semigroups. *Journal of Functional Analysis*, 13:51 – 72, 1973.

[Mar84] J. M. M. C. Marques. Stress computations in elastoplasticity. *Engineering Computations*, 1:42–51, 1984.

[MAS93] P. E. McHugh, R. J. Asaro, and C. F. Shih. Computational modeling of metal matrix composite materials – I: Isothermal deformation patterns in ideal microstructures. *Acta Metallurgica and Materialia*, 41(5):1461–1476, 1993.

[Mas98] Thomas Massie. A tangible goal for 3D modeling. *IEEE Computer Graphics and Applications*, 18:62–65, 1998.

[Maš05] D Mašín. A hypoplastic constitutive model for clays. *International Journal for Numerical and Analytical Methods in Geomechanics*, 29(4):311–336, 2005.

[Mat89] Hermann Matthies. A decomposition method for the integration of the elastic-plastic rate problem. *International Journal for Numerical Methods in Engineering*, 28:1–11, 1989.

[Mat04] H.G. Matthies. *Uncertainty Quantification with Stochastic Finite Elements*. Wiley, 2004.

[Mat18] Devin A Matthews. High-performance tensor contraction without transposition. *SIAM Journal on Scientific Computing*, 40(1):C1–C24, 2018.

[MAT^{+}19] J.M. Mayoral, D. Asimaki, S. Tepalcapa, C. Wood, A. Roman de la Sancha, T. Hutchinson, K. Franke, and G. Montalva. Site effects in mexico city basin: Past and present. *Soil Dynamics and Earthquake Engineering*, 121:369 – 382, 2019.

[May07] Paul W. Mayne. Cone penetration testing state-of-practice. Technical Report NCHRP Project 20-05, Topic 37-14, Georgia Institute of Technology, 2007.

[MB88] J. B. Martin and W. W. Bird. Integration along the path of loading in elastic - plastic problems. *Engineering Computations*, 5:217–223, 1988.

[MB94] L. K. Michaelson and R. H. Black. *Building learning teams: the key to harnessing the power of small groups in higher education*, volume 2 of *Collaborative learning: A Sourcebook for Higer Education*. National Center for Teaching, Learning & Asesment, State College, PA., 1994.

[MB99] Hermann G. Matthies and Christian Bucher. Finite elements for stochastic media problems. *Computer methods in applied mechanics and engineering*, 168:3–17, 1999.

[MB04] Arif Masud and Lawrence A. Bergman. Application of multi-scale finite element methods to the solution of the Fokker–Planck equation. *Computer Methods in Applied Mechanics and Engineering*, In press, 2004.

[MB05] Arif Masud and Lawrence A. Bergman. Application of multi-scale finite element methods to the solution of the Fokker–Planck equation. *Computational Methods in Applied Mechanics and Engineering*, 194(1):1513–1526, April 2005.

[MB09] J. Mosler and O.T. Bruhns. Towards variational constitutive updates for non-associative plasticity models at finite strain: Models based on a volumetric-deviatoric split. *International Journal of Solids and Structures*, 46(7-8):1676–1684, 2009.

[MB16] Olivier Mesnard and Lorena A. Barba. Reproducible and replicable CFD: it's harder than you think. *arXiv*, .(1605.04339v2):1–11, May 2016.

[MBB^{+}21] Eduardo Miranda, Svetlana Brzev, Nenad Bijelić, Željko Arbanas, Marko Bartolac, Vedran Jagodnik, Damir Lazarević, Snježana Mihalić Arbanas, Sonja Zlatović, Andres Acosta Vera, Jorge Archbold, James Bantis, Nikola Blagojević, Jovana Borozan, Ivana Božulić, Cristian Cruz, Héctor Dávalos, Erica Fischer, Selim Gunay, Marijana Hadzima-Nyarko, Pablo Heresi, Dimitrios Lignos, Ting Lin, Marko Marinković, Sebastian Miranda, Armando Messina, Alan Poulos, Giulia Scagliotti, Ingrid Tomac, Igor Tomić, Katerina Ziotopoulou, Željko Žugić, Ian Robertson, and Dimitrios Lignos. StEER-EERI: Petrinja, Croatia december 29, 2020, Mw 6.4 earthquake. Technical Report PRJ-2959, EERI, StEER, 2021.

[MBBS04] Hermann G. Matthies, Christoph E. Brenner, Christian G. Bucher, and C. Guedes Soares. Uncertainties in probabilistic numerical analysis of structures and solids – stochastic finite elements. *Structural Safety*, 19(3):283–336, 2004.

[MBDG94] N. Makris, D. Badoni, E. Delis, and G. Gazetas. Prediction of observed bridge response with soil–pile–structure interaction. *ASCE Journal of Structural Engineering*, 120(10):2992–3011, October 1994.

[MBG02] Giuseppe Mortara, Marc Boulon, and Vito Nicola Ghionna. A 2-d constitutive model for cyclic interface behaviour. *International journal for numerical and analytical methods in geomechanics*, 26(11):1071–1096, 2002.

[MBH21] Ruisheng Ma, Kaiming Bi, and Hong Hao. Inerter-based structural vibration control: A state-of-the-art review. *Engineering Structures*, 243:112655, 2021.

[MBV^{+}00] Paul W. Mayne, Dan Brown, James Vinson, James A. Schneider, and Kimberly A. Finke. Site characterization of piedmont residual soils at the NGES, Opelika, Alabama. In Jean Benoit and Alan Lutenegger, editors, *ASCE Geotechnical Special Publication (GSP) No. 93, National Geotechnical Experimentation Sites (NGES),*, pages 160–185, ASCE Reston, Virginia, 2000.

[MC83] E. Mizuno and W. F. Chen. Plasticity analysis of slope with different flow rules. *Computers & Structures*, 17(3):375–388, 1983.

[MC95] Richard H. McCuen and Peter C. Chang. Multimedia–based instruction in engineering education: Evaluation. *ASCE Journal of Professional Issues in Engineering Education and Practice*, 121(4):220–224, October 1995.

[MC01] Christian Miehe and Jörg Cshröder. A comparative study of stress update algorithm for rate-independent and rate-dependent crystal plasticity. *International Journal for Numerical Methods in Engineering*, 50:273–298, 2001.

[MC05] G.R. Martin and C.-Y. Chen. Response of piles due to lateral slope movement. *Computers & Structures*, 83(8-9):588–598, March 2005.

[MCB^{+}18] A. Mouyeaux, C. Carvajal, P. Bressolette, L. Peyras, P. Breul, and C. Bacconnet. Probabilistic stability analysis of an earth dam by stochastic finite element method based on field data. *Computers and Geotechnics*, 101:34–47, 2018.

[MCC^{+}11] Masoud Moghaddasi, Misko Cubrinovski, J. Geoff Chase, Stefano Pampanin, and Athol Carr. Probabilistic evaluation of soil-foundation-structure interaction effects on seismic structural response. *Earthquake Engineering & Structural Dynamics*, 40(2):135–154, 2011.

[McF88] Jerome J. McFadden. Experimental response of sand during principal stress rotations. Master of Science thesis, University od Colorado at Boulder, December 6 1988.

[McG95] Robin K McGuire. Probabilistic seismic hazard analysis and design earthquakes: closing the loop. *Bulletin of the Seismological Society of America*, 85(5):1275–1284, 1995.

[McG01] Robin K McGuire. Deterministic vs. probabilistic earthquake hazards and risks. *Soil Dynamics and Earthquake Engineering*, 21(5):377–384, 2001.

[McG04] Robin K McGuire. *Seismic hazard and risk analysis*. Earthquake Engineering Research Institute, 2004.

[MCH$^+$15] E. Maufroy, E. Chaljub, F. Hollender, J. Kristek, P. Moczo, P. Klin, E. Priolo, A. Iwaki, T. Iwata, V. Etienne, F. De Martin, N. P. Theodoulidis, M. Manakou, C. Guyonnet-Benaize, K. Pitilakis, and P.Y. Bard. Earthquake ground motion in the mygdonian basin, greece: The e2vp verification and validation of 3D numerical simulation up to 4 hz. *Bulletin of the Seismological Society of America*, 2015.

[McK95] Michael D McKay. Evaluating prediction uncertainty. Technical report, Nuclear Regulatory Commission, 1995.

[McK97a] Francis Thomas McKenna. *Object Oriented Finite Element Programming: Framework for Analysis, Algorithms and Parallel Computing*. PhD thesis, University of California, Berkeley, 1997.

[McK97b] Francis Thomas McKenna. Object oriented finite element programming: Framework for analysis, algorithms and parallel computing. Technical report, University of California, Berkeley, 1997.

[McK11] Frank McKenna. OpenSees: A framework for earthquake engineering simulations. *Computing in Sciences & Engineering, IEEE*, 13(4):58–66, Jul-Aug 2011.

[McK21] Francis McKenna. Private communications, 1999-2021.

[MCM25] Ignasi Mundó, Ferhun C. Caner, and Antonio Mateo. Micromechanical modelling of the elastoplasticity and damage in ductile metals. *International Journal of Solids and Structures*, 317:113437, 2025.

[MCS95] Michael McVay, Robert Casper, , and Te-I Shang. Lateral response of three-row groups in loose to dense sands at 3D and 5D pile spacing. *Journal of Geotechnical Engineering*, 121(5):436–441, May 1995.

[MCS98] James K. Mitchell, Harry G. Cooke, and Jennifer A. Schaeffer. Design considerations in ground improvement for seismic risk mitigation. In Panos Dakoulas, Mishac Yegian, and Robert D. Holtz, editors, *Proceedings of a Specialty Conference: Geotechnical Earthquake Engineering and Soil Dynamics III*, Geotechnical Special Publication No. 75, pages 580–613. ASCE, August 1998. 1998.

[MD53] R. D. Mindlin and H. Deresiewicz. Elastic spheres in contact under varying oblique forces. *ASME Journal of Applied Mechanics*, 53(APM–14):327–344, September 1953.

[MD69] Zenon Mróz and Andrzej Drescher. Limit plasticity approach to some cases of flow of bulk solids. *Journal of Engineering for Industry*, 91(2):357–364, 1969.

[MD86] A. Muqtadir and C. S. Desai. Three dimensional analysis of a pile-group foundation. *International journal for numerical and analysis methods in geomechanics*, 10:41–58, 1986.

[MD97] M. T. Manzari and Y. F. Dafalias. A critical state two–surface plasticity model for sands. *Géotechnique*, 47(2):255–272, 1997.

[MD04] Jack Moehle and Gregory G Deierlein. A framework methodology for performance-based earthquake engineering. In *13th world conference on earthquake engineering*, volume 679, 2004.

[MD06] LH Mejia and EM Dawson. Earthquake deconvolution for flac. In *4th International FLAC Symposium on Numerical Modeling in Geomechanics*, pages 04–10, 2006.

[MD07] Lelio H. Mejia and Ethan M Dawson. Analysis of seismic response of seven oaks dam. In Moh Huang, editor, *Proceedings of SMIP 2007 Seminar on Utilization of Strong-Motion Data*, pages 21–40. California Strong Motion Instrumentation Program, 2007.

[MdB05] Sami Montassar and Patrick de Buhan. Minimum principle and related numerical scheme for simulating initial flow and subsequent propagation of liquefied ground. *International Journal for Numerical and Analytical Methods in Geomechanics*, 29(11):1065–1086, 2005.

[MDMS07] John B Mander, Rajesh P Dhakal, Naoto Mashiko, and Kevin M Solberg. Incremental dynamic analysis applied to seismic financial risk assessment of bridges. *Engineering structures*, 29(10):2662–2672, 2007.

[MDP04] G. P. Mavroeidis, G. Dong, and A. S. Papageorgiou. Near-fault ground motions, and the response of elastic and inelastic single-degree-of-freedom (sdof) systems. *Earthquake Engineering & Structural Dynamics*, 33(9):1023–1049, 2004.

[MDTC01] N. Mai-Duy and T. Tran-Cong. Numerical solution of differential equations using multiquadric rbfs. *Neural Networks*, 14:185–199, 2001.

[MDZ$^+$07] Philip Maechling, Ewa Deelman, Li Zhao, Robert Graves, Gaurang Mehta, Nitin Gupta, John Mehringer, Carl Kesselman, Scott Callaghan, David Okaya, et al. SCEC cybershake workflow-automating probabilistic seismic hazard analysis calculations. In *Workflows for e-Science*, pages 143–163. Springer, 2007.

[MDZP$^+$18] V. Masson-Delmotte, P. Zhai, H.-O. Pörtner, D. Roberts, J. Skea, P.R. Shukla, A. Pirani, W. Moufouma-Okia, C. Péan, R. Pidcock, S. Connors, J.B.R. Matthews, Y. Chen, X. Zhou, M.I. Gomis, E. Lonnoy, T. Maycock, M. Tignor, and T. Waterfield, editors. *IPCC, 2018: Global Warming of 1.5^oC. An IPCC Special Report on the impacts of global warming of 1.5^oC above pre-industrial levels and related global greenhouse gas emission pathways, in the context of strengthening the global response to the threat of climate change, sustainable development, and efforts to eradicate poverty* . Cambridge University Press, Cambridge, UK and New York, NY,, 2018. 616pp.

[MEB95] Wahib Meftab, Pierre Evesque, and Jean Biarez. Deformation of oredered 2–D packing of grains, role of rotations. In Stein Sture, editor, *Proceedings of 10th Conference*, pages 1284–1287. Engineering Mechanics Division of the American Society of Civil Engineers, May 1995.

[MEHS$^+$13] Nicholas Malaya, Kemelli C Estacio-Hiroms, Roy H Stogner, Karl W Schulz, Paul T Bauman, and Graham F Carey. Masa: a library for verification using manufactured and analytical solutions. *Engineering with Computers*, 29(4):487–496, 2013.

[Mei86] Leonard Meirovitch. *Elements of Vibration Analysis*. McGraw - Hill Inc., 1986.

[Mel65] Mirko J. Melentijevich. The analysis of range with output linearly dependent upon storage. Technical Report 11, Colorado State University, Fort Collings, Colorado, USA, September 1965.

[Men96] Philippe Menetrey. Analytical computation of the punching strength of reinforced concrete. *ACI Structural Journal*, 93(5):503–511, September–October 1996.

[Men03] Jiewu Meng. *The Influence of Loading Frequency on Dynamic Soil Properties*. PhD thesis, Georgia Institute of Technology, 2003.

[Mer04] Panagiotis Mergos. Rocking isolation of bridge piers resting on spreading foundations. Master's thesis, ROSE School, European School for Advanced Studies in Reduction of Seismic Risk, May 2004. `www.roseschool.it/docs/Dissertation2004-Mergos.pdf`.

[MESZA17] Vicente Mercado, Waleed El-Sekelly, Mourad Zeghal, and Tarek Abdoun. Identification of soil dynamic properties of sites subjected to bi-directional excitation. *Soil Dynamics and Earthquake Engineering*, 92:215 – 228, 2017.

[MF91] Carmelo Militello and Carlos A. Felippa. The first ANDES elements: 9-dof plate bending triangles. *Computer Methods in Applied Mechanics and Engineering*, 93:217–246, 1991.

[MFH15] Alejandro Martinez, J David Frost, and Gregory L Hebeler. Experimental study of shear zones formed at sand/steel interfaces in axial and torsional axisymmetric tests. *Geotechnical Testing Journal*, 38(4):409–426, 2015.

[MFK97] L. K. Michaelsen, L. D. Fink, and A. Knight. Designing effective group activities: Lessons for classroom teaching and faculty development. *To Improve Academy*, 16:373–398, 1997.

[MFP18] Kristel C. Meza Fajardo and Apostolos S. Papageorgiou. Response of tall buildings to base rocking induced by rayleigh waves. *Earthquake Engineering & Structural Dynamics*, 47(8):1755–1773, 2018.

[MFP23] Kristel C. Meza-Fajardo and Apostolos S. Papageorgiou. Quantitative analysis of surface-wave propagation in the Mexico City valley. *Earthquake Engineering & Structural Dynamics*, 52(9):2755–2771, 2023.

[MFPS15] Kristel C. Meza-Fajardo, Apostolos S. Papageorgiou, and Jean-François Semblat. Identification and extraction of surface waves from three-component seismograms based on the normalized inner product. *Bulletin of the Seismological Society of America*, 105(1):1–20, February 2015.

[MFS04] S. Mazzoni, G.L. Fenves, and J.B. Smith. Effects of local deformations on lateral response of bridge frames. Technical report, University of California, Berkeley, April 2004.

[MFSCL16] Kristel C. Meza-Fajardo, Jean-François Semblat, Stéphanie Chaillat, and Luca Lenti. Seismic-wave amplification in 3D alluvial basins: 3D/1D amplification ratios from fast multipole BEM simulations. *Bulletin of the Seismological Society of America*, 106(3):(1–15 (tentative)), June 2016.

[MFVL$^+$19] Kristel C. Meza-Fajardo, Chiara Varone, Luca Lenti, Salvatore Martino, and Jean-François Semblat. Surface wave quantification in a highly heterogeneous alluvial basin: Case study of the Fosso di Vallerano valley, Rome, Italy. *Soil Dynamics and Earthquake Engineering*, 120:292 – 300, 2019.

[MG99] George Myionakis and George Gazetas. Lateral vibration and internal forces of grouped piles in layered soil. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 125(1):16–25, 1999.

[MG00] George Mylonakis and George Gazetas. Seismic soil-structure interaction: beneficial or detrimental? *Journal of Earthquake Engineering*, 4(03):277–301, 2000.

[MG19] George Markou and Filippo Genco. Seismic assessment of small modular reactors: Nuscale case study for the 8.82̌02fmw earthquake in chile. *Nuclear Engineering and Design*, 342:176 – 204, 2019.

[MGEA$^+$10] Clotaire Michel, Philippe Guéguen, Saber El Arem, Jacky Mazars, and Panagiotis Kotronis. Full-scale dynamic response of an rc building under weak seismic motions using earthquake recordings, ambient vibrations and modelling. *Earthquake Engineering & Structural Dynamics*, 39(4):419–441, 2010.

[MGG$^+$23] Andrea Marchi, Domenico Gallese, Davide Noè Gorini, Paolo Franchin, and Luigi Callisto. On the seismic performance of straight integral abutment bridges: From advanced numerical modelling to a practice-oriented analysis method. *Earthquake Engineering & Structural Dynamics*, 52(1):164–182, 2023.

[MGH$^+$11] Jack P Moehle, Tony Ghodsi, John D Hooper, David C Fields, and Rajnikanth Gedhada. Seismic Design of Cast-in-Place Concrete Special Structural Walls and Coupling Beams: A Guide for Practicing Engineers. Technical Report NEHRP Seismic Design Technical Brief No.6, National Institute of Standards and Technology (NIST), 2011.

[MGR14] Divya S.K. Mana, Susan Gourvenec, and Mark F. Randolph. Numerical modelling of seepage beneath skirted foundations subjected to vertical uplift. *Computers and Geotechnics*, 55(0):150 – 157, 2014.

[MGR$^+$16] Andrea Magrin, Alexander A. Gusev, Fabio Romanelli, Franco Vaccari, and Giuliano F. Panza. Broadband NDSHA computations and earthquake ground motion observations for the Italian territory. *Int. J. Earthquake and Impact Engineering*, 1(1/2):131–158, 2016.

[MH83] Jerrold E. Marsden and Thomas J. R. Hughes. *Mathematical Foundations of Elasticity*. Prentice Hall Inc,, 1983. local CM65; 4QA 931.M42 ; ISBN 0-13-561076-1.

[MH89] SL McCabe and WJ Hall. Assessment of seismic structural damage. *Journal of Structural Engineering*, 115(9):2166–2183, 1989.

[MH03] Yun Mei Hsiung. Theoretical elastic–plastic solution for laterally loaded piles. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 129(6):475–480, June 2003.

[MH04] Karen T. Marosi and Dennis R. Hiltunen. Characterization of spectral analysis of surface waves shear wave velocity measurement uncertainty. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 130(10):1034–1041, October 2004.

[MHS05] Marjan Mernik, Jan Heering, and Anthony M. Sloane. When and how to develop domain-specific languages. *ACM Computing Surveys*, 37(4):316–344, December 2005. `http://doi.acm.org/10.1145/1118890.1118892`.

[MHSCCC06] Yun Mei Hsiung, Shi Shuenn Chen, and Yi Chuan Chou. Analytical solution for piles supporting combined lateral loads. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 132(10):1315–1324, October 2006.

[MHZ98] G. W. Ma, H. Hao, and Y. X. Zhou. Modeling of wave propagation induced by underground explosion. *Computers and Geotechnics*, 22(3-4):283–303, 1998.

[Mic78] R Michalowski. Associated and non-associated sliding rules in contact friction problems. *Archiv. Mech.*, 30:259–276, 1978.

[Mic98] R. L. Michalowski. Kinematic hardening of fiber–reinforced granular composites. In Murakami and Luco, editors, *Proceedings of the 12th Conference*, pages 1343–1346, La Jolla, California, May 1998. Engineering Mechanics Division of the American Society of Civil Engineers.

[Mic06] Paul Michaels. Relating damping to soil permeability. *International Journal for Geomechanics*, 6(3):158–165, May/June 2006.

[Mie94a] C. Miehe. Aspects of the formulation and finite element implementation of large strain isotropic elasticity. *International Journal for Numerical Methods in Engineering*, 37:1981–2004, 1994.

[Mie94b] Christian Miehe. On the representation of Prandtl–Reuss tensors within the framework of multiplicative elastoplasticity. *International Journal of Plasticity*, 10(6):609–621, 1994.

[Mie95a] C. Miehe. Entropic thermoelasticity at finite strains. aspects of the formulation and numerical implementation. *Computer Methods in Applied Mechanics and Engineering*, 120:243–269, 1995.

[Mie95b] C. Miehe. A theory of large–strain isotropic thermoelasticity based on metric transformation tensors. *Archive of Applied Mechanics*, 66:45–64, 1995.

[Mie96] C. Miehe. Multisurface thermoelasticity for single crystals at large strains in terms of eulerian vector updates. *International Journal of Solids and Structures*, 33(20-22):3103–3130, 1996.

[Mie98] Christian Miehe. A formulation of finite elastoplasticity based on dual co– and contra–variant eigenvectors triads normalized with respect to a plastic metric. *International Journal for Computer Methods in Applied Mechanics and Engineering*, 159:223–260, 1998.

[MII03] Hiroe Miyake, Tomotaka Iwata, and Kojiro Irikura. Source characterization for broadband ground-motion simulation: Kinematic heterogeneous source model and strong motion generation area. *Bulletin of the Seismological Society of America*, 93(6):2531–2545, 2003.

[Mil91a] Carmelo Militello. *Application of Parametrized Variational Principles in the Finite Element Method*. PhD thesis, University of Colorado at Boulder, august 1991.

[Mil91b] G. R. Miller. An object – oriented approach to structural analysis and design. *Computers and Structures*, 40(1):75–82, 1991.

[Mil06] Greg Miller. A scientist's nightmare: software problem leads to five retractions. *Science*, 314(5807):1856–1857, 2006.

[Min63] R. D. Mindlin. Influence of couple-stresses on stress concentrations. *Experimental Mechanics*, 3(1):1–7, 1963.

[Min65] R.D. Mindlin. Stress functions for a Cosserat continuum. *International Journal of Solids and Structures*, 1(3):265 – 271, 1965.

[Mir93] Eduardo Miranda. Evaluation of site–dependent inelastic sesimic design spectra. *ASCE Journal of Structural Engineering*, 119(5):1319–1338, 1993.

[MJ94a] James G. Malone and Nancy L. Johnson. A parallel finite element contact/impact algorithm for non–linear explicit transient analysis: Part I – the search algorithm and contact mechanics. *International Journal for Numerical Methods in Engineering*, 37:559–590, 1994.

[MJ94b] James G. Malone and Nancy L. Johnson. A parallel finite element contact/impact algorithm for non–linear explicit transient analysis: Part II – parallel implementation. *International Journal for Numerical Methods in Engineering*, 37:591–603, 1994.

[MJB05] Francisco Javier Montáns and Klaus Jürgen Bathe. Computational issues in large strain elasto-plasticity: an algorithm for mixed hardening and plastic spin. *International Journal for Numerical Methods in Engineering*, 63(2):159–196, 2005.

[MJJaY05a] D. Harris M. J. Jiang and and H. S. Yu. Kinematic models for non-coaxial granular materials. Part I: theory. *International Journal for Numerical and Analytical Methods in Geomechanics*, 29(7):643–661, April 2005.

[MJJaY05b] D. Harris M. J. Jiang and and H. S. Yu. Kinematic models for non-coaxial granular materials. Part II: evaluation. *International Journal for Numerical and Analytical Methods in Geomechanics*, 29(7):663–689, April 2005.

[MK0] A. N. Moysidis and V. K. Koumousis. Hysteretic plate finite element. *Journal of Engineering Mechanics*, 0(0):04015039, 0.

[MK98] Jeffrey S. Mulliken and Dimitris L. Karabalis. Discrete model for dynamic through-the-soil coupling of 3-d foundations and structures. *Earthquake Engineering & Structural Dynamics*, 27(7):687–710, 1998.

[MK05] Hermann G. Matthies and Andreas Keese. Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations. *Computational Methods in Applied Mechanics and Engineering*, 194(1):1295–1331, April 2005.

[MK06] Irene Moulitsas and George Karypis. Architecture aware partitioning algorithms. Technical report, Department of Computer Science, University of Minnesota, 2006. UMN CSE TR #06–001.

[MK08] Morteza Jiryaei Sharahi Mohsen Kamalian, Behrouz Gatmiri. Time domain 3D fundamental solutions for saturated poroelastic media with incompressible constituents. *Communications in Numerical Methods in Engineering*, 24(9):749–759, 2008.

[MK10] O. Le Maitre and O. M. Knio. *Spectral methods for uncertainty quantification with applications to computational fluid dynamics*. Springer, Dordrecht, 2010.

[MK14] Teja Melink and Jože Korelc. Stability of Karhunen-Loève expansion for the simulation of Gaussian stochastic fields using Galerkin scheme. *Probabilistic Engineering Mechanics*, 37:7–15, 2014.

[MKBP16] Marco Miniaci, Anastasiia Krushynska, Federico Bosia, and Nicola Pugno. Large scale mechanical metamaterials as seismic shields. *New Journal of Physics*, 18:083041, 08 2016.

[MKG17] G Michaloudis, A Konyukhov, and N Gebbeken. An interface finite element based on a frictional contact formulation with an associative plasticity model for the tangential interaction. *International Journal for Numerical Methods in Engineering*, 2017.

[MKGB18] Ioannis P Mitseas, Ioannis A Kougioumtzoglou, Agathoklis Giaralis, and Michael Beer. A novel stochastic linearization framework for seismic demand estimation of hysteretic mdof systems subject to linear response spectra. *Structural Safety*, 72:84–98, 2018.

[MKL$^+$22] Ilhwan Moon, Doyeon Kim, Kyungkoo Lee, Jaemin Kim, and Heekyun Kim. Seismic analysis of nuclear power plant structures under beyond-design basis earthquake excitation. In *Transactions, SMiRT-26*. SMiRT, July 2022.

[MKTZ03] V Montaldo, AS Kiremidjian, H Thrainsson, and G Zonno. Simulation of the fourier phase spectrum for the generation of synthetic accelerograms. *Journal of Earthquake Engineering*, 7(03):427–445, 2003.

[ML94] MÀrcio A. Murad and Abimael F. D. Loula. On stability and convergence of finite element approximations of Biot's consolidation problem. *International Journal for Numerical Methods in Engineering*, 37:645–667, 1994.

[ML99] G. Meschke and W. N. Liu. A re–formulation of the exponential algorithm for finite strain plasticity in terms of Cauchy stresses. *International Journal for Computer Methods in Applied Mechanics and Engineering*, 159:167–187, 1999.

[ML14] Rui Marques and Paulo B Lourenço. Unreinforced and confined masonry buildings in seismic regions: Validation of macro-element models and cost analysis. *Engineering Structures*, 64:52–67, 2014.

[ML17] Mebrahtom Gebrekirstos Mezgebo and Eric M Lui. A new methodology for energy-based seismic design of steel moment frames. *Earthquake Engineering and Engineering Vibration*, 16(1):131–152, 2017.

[MLA94] Hormoz Modaressi, Lyesse Laloui, and Denis Aubry. Thermodynamical approach for Camclay–family models with Roscoe–type dilatancy rules. *International Journal for Numerical and Analytical Methods in Geomechanics*, 18:133–138, 1994.

[MLA20] Jorge Macedo, Maxime Lacour, and Norman Abrahamson. Epistemic uncertainty treatment in seismically induced slope displacements using polynomial chaos. *Journal of Geotechnical and Geoenvironmental Engineering*, 146(10):04020111, 2020.

[MLL$^+$22] L. Marconato, P.H. Leloup, C. Lasserre, R. Jolivet, S. Caritg, R. Grandin, M. Métois, O. Cavalié, and L. Audin. Insights on fault reactivation during the 2019 November 11, Mw 4.9 Le Teil earthquake in southeastern France, from a joint 3-D geological model and InSAR time-series analysis. *Geophys. J. Int.*, 229:758–775, 2022.

[MLP^{+}12] H. G. Matthies, A. Litvinenko, O. Pajonk, B. V. Rosić, and E. Zander. Parametric and uncertainty computations with tensor product representations, in uncertainty quantification in scientific computing. *IFIP Advances in Information and Communication Technology*, 377:139–150, 2012.

[MLS^{+}23] Mohammad J. A. Moein, Cornelius Langenbruch, Ryan Schultz, Francesco Grigoli, William L. Ellsworth, Ruijia Wang, Antonio Pio Rinaldi, and Serge Shapiro. The physical mechanisms of induced earthquakes. *Nature Reviews Earth & Environment*, 4:847–863, December 2023.

[MM52] JC Martin and WJ Moyce. An experimental study of the collapse of liquid columns on a rigid horizontal plane. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, pages 312–324, 1952.

[MM96] M. P. Miller and D. L. McDowwel. Modeling large strain multiaxial effects in FCC polycrystals. *International Journal of Plasticity*, 12(7):875–902, 1996.

[MM14] Abbas Moustafa and Sayed Mahmoud. Damage assessment of adjacent buildings under earthquake loads. *Engineering Structures*, 61:153–165, 2014.

[MM21] Fatemeh Soleiman Meigooni and Fabrizio Mollaioli. Simulation of seismic collapse of simple structures with energy-based procedures. *Soil Dynamics and Earthquake Engineering*, 145:106733, 2021.

[MMB^{+}20] Thomas P. Murphy, Lee Marsh, Stuart Bennion, Ian G. Buckle, Nicolas Luco, Donald Anderson, Mervyn Kowalsky, and Jose Restrepo. Proposed AASHTO guidelines for performance-based seismic bridge design. Technical Report ISBN 978-0-309-48177-9 ; DOI 10.17226/25913, The National Academies Press., 2020.

[MMD^{+}16] F Moukalled, L Mangani, M Darwish, et al. *The finite volume method in computational fluid dynamics*. Springer, 2016.

[MMG07] Giuseppe Mortara, Antonio Mangiola, and Vito Nicola Ghionna. Cyclic shear stress degradation and post-cyclic behaviour from sand–steel interface direct shear tests. *Canadian Geotechnical Journal*, 44(7):739–752, 2007.

[MMS^{+}02] Silvia Mazzoni, Frank McKenna, Michael H. Scott, Gregory L. Fenves, and Boris Jeremić. *Open System for Earthquake Engineers Simulation (OpenSees) : User Manual*. Pacific Earthquake Engineering Research Center, Richmond, December 2002.

[MN82] Z. Mroz and V. A. Norris. Elastoplastic and viscoplastic constitutive models for soils with application to cyclic loadings. In G. N. Pande and O. C. Zienkiewicz, editors, *Soil Mechanics – Transient and Cyclic Loads*, pages 173–217. John Wiley and Sons Ltd., 1982.

[MN84] G. Maier and A. Nappi. On the unified framework provided by mathematical programming to plasticity. In Richard T. Shield George J. Dvorak, editor, *Mechanics of Material Behavior, The Daniel C. Drucker Anniversary Volume*, pages 253–273. Elsevier, 1984.

[MN88] W.R. Madych and S.A. Nelson. Multivariate interpolation and conditionally positive definite functions. *Approx. Theory Appl.*, 4:77–89, 1988.

[MN97] G. Mylonakis and A. Nikolaou. Soil–pile–bridge seismic interaction: Kinematic and inertial effects. part i: Soft soil. *Earthquake Engineering & Structural Dynamics*, 26:337–359, 1997.

[MNC^{+}19] Lluís Monforte, Pedro Navas, Josep Maria Carbonell, Marcos Arroyo, and Antonio Gens. Low-order stabilized finite element for the full Biot formulation in soil mechanics at finite strain. *International Journal for Numerical and Analytical Methods in Geomechanics*, 43(7):1488–1515, 2019.

[MNG06] G. Mylonakis, S. Nikolaou, and G. Gazetas. Footings under seismic loading: Analysis and design issues with emphasis on bridge foundations. *Soil Dyanmics and Earthquake Engineering*, 26:824–853, 2006.

[MNKS91] Klisinski M., Abifadel N., Runesson K., and Sture S. Modelling of the behaviour of dry sand by an elasto–plastic "fuzzy – set" model. *Computers and Geotechnics*, 11:229–261, 1991.

[MNN83] M. M. Mehrabadi and S. Nemat-Nasser. Stress, dilatancy and fabric in granular materials. *Mechanics of Materials*, 2:155–161, 1983.

[MNN87] M. M. Mehrabadi and S. Nemat-Nasser. Some basic kinematical relations for finite deformations of continua. *Mechanics of Materials*, 6:127–138, 1987.

[MNZ79] Z. Mróz, V. A. Norris, and O. C. Zienkiewicz. Application of an anisotropic hardening model in the analysis of elasto–plastic deformation of soils. *Géotechnique*, 29(1):1–34, 1979.

[MO84] J. M. M. C. Marques and D. R. J. Owen. Some reflections on elastoplastic stress calculations in finite element analysis. *Computers & Structures*, 18(6):1135–1139, 1984.

[MO88] G. P. Mitchell and D. R. J. Owen. Numerical solution for elasto - plastic problems. *Engineering Computations*, 5:274–284, 1988.

[Moa93] Torgeir Moan. Reliability and risk analysis for design and operations planning of offshore structures. In *Sixt ICOSSAR, Innsbruck, 9-13 August*. A. A. Balkema Publishers, Rotterdam, 1993.

[Mog06] Kiyoo Mogi. *Experimental Rock Mechanics*. Taylro and Francis, 2006.

[Moo65a] Gordon E. Moore. Cramming more components onto integrated circuits. *Electronics Magazine*, 38(8), April 1965.

[Moo65b] Gordon E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8):33–35, 1965.

[Mor19] J. J. Moreau. Application of convex analysis to the treatment of elastoplastic systems. In __, pages 56–89,. __, 19__.

[Mor65] JoDean Morrow. Cyclic plastic strain energy and fatigue of metals. In *Cyclic Internal friction, damping, and cyclic plasticity,*, pages 45–87, West Conshohocken, PA, 1965. ASTM.

[Mor86] K. N. Morman. The generalized strain meassures with application to nonhomogeneous deformation in rubber–like solids. *Journal of Applied Mechanics*, 53:726–728, 1986.

[MOS90] B. Moran, M. Ortiz, and C. F. Shih. Formulation of implicit finite element methods for multiplicative finite deformation plasticity. *International Journal for Numerical Methods in Engineering*, 29:483–514, 1990.

[Mos05] J. Mosler. Numerical analyses of discontinuous material bifurcation: strong and weak discontinuities. *Computer Methods in Applied Mechanics and Engineering*, 194(9-11):979–1000, 18 March 2005.

[MOSC$^+$07] L Moratto, B Orlecka-Sikora, G Costa, P Suhadolc, Ch Papaioannou, and CB Papazachos. A deterministic seismic hazard analysis for shallow earthquakes in Greece. *Tectonophysics*, 442(1-4):66–82, 2007.

[Mou01] CT. Mouat. *Fast algorithms and preconditioning techniques for fitting radial basis function*. Doctoral dissertation, Mathematics Dept., University of Canterbury, Christchurch, NZ, 2001.

[Mou11] Abbas Moustafa. Damage-based design earthquake loads for single-degree-of-freedom inelastic structures. *Journal of Structural Engineering*, 137(3):456–467, 2011.

[MP54] GF Miller and H. Pursey. The field and radiation impedance of mechanical radiators on the free surface of a semi-infinite isotropic solid. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 223(1155):521, 1954.

[MP55] GF Miller and H. Pursey. On the partition of energy between elastic waves in a semi-infinite solid. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 233(1192):55, 1955.

[MP73] Marco Menegotto and Paolo Emilio Pinto. Method of analysis for cyclically loaded reinforced clncrete plane frames including changes in geometry and non-elastic behaviour of elements under combined normal force and bending. In *Proceedings of IABSE Symposium*, pages 15–22, 1973.

[MP89] José Emir Macari-Pasqualino. *Behavior of Granular Material in a Reduced Gravity Environment and Under Low Effective Stresses*. PhD thesis, University of Colorado at Boulder, 1989.

[MP03] G. P. Mavroeidis and A. S. Papageorgiou. A mathematical representation of near-fault ground motions. *Bulletin of the Seismological Society of America*, 93(3):1099–1131, June 2003.

[MP16] Mohammad Maleki and Hassan Pouyan. A kinematic hardening based model for unsaturated soils considering different hydraulic conditions. *International journal for Numerical and Analytical Methods in Geomechanics*, 40(16):2271–2290, December 2016.

[MPA$^+$20a] Gaetano Miraglia, Miloš Petrović, Giuseppe Abbiati, Nebojša Mojsilović, and Bo vzidar Stojadinovi 'c. A model-order reduction framework for hybrid simulation based on component-mode synthesis. *Earthquake Engineering & Structural Dynamics*, 49(8):737–753, 2020.

[MPA$^+$20b] Gaetano Miraglia, Miloš Petrović, Giuseppe Abbiati, Nebojša Mojsilović, and Božidar Stojadinović. A model-order reduction framework for hybrid simulation based on component-mode synthesis. *Earthquake Engineering & Structural Dynamics*, n/a(n/a), 2020.

[MPC95] Jack Mazars and Gilles Pijaudier-Cabot. From damage to fracture mechanics and conversely a combined approach. In Stein Sture, editor, *Proceedings of 10th Conference*, pages 231–234. Engineering Mechanics Division of the American Society of Civil Engineers, May 1995.

[MPD92] Richard A. Millet, Jean-Yves Perez, and Richard R. Davidson. USA practice slurry wall specifications 10 years later. In David B. Paul, Richard R. Davidson, and Nicholas J. Cavalli, editors, *Slurry Walls: Design, Construction and Quality Control*, ASTM Publication Code Number (PCN) 04-011290-38, pages 42–66. ASTM, ASTM, 1916 Race Street, Philadelphia, PA 19103, 1992.

[MPI] MPI Forum. The message passing interface (MPI). set of documents on WWW, . `http://www-c.mcs.anl.gov/mpi/index.html`

[MPP88] John B Mander, Michael JN Priestley, and R Park. Theoretical stress-strain model for confined concrete. *Journal of structural engineering*, 114(8):1804–1826, 1988.

[MPP$^+$15] Morgan P. Moschetti, Peter M. Powers, Mark D. Petersen, Oliver S. Boyd, Rui Chen, Edward H. Field, Arthur D. Frankel, Kathleen M. Haller, Stephen C. Harmsen, Charles S. Mueller, Russel L. Wheeler, and Yuehua Zeng. Seismic source characterization for the 2014 update of the US national seismic hazard model. *Earthquake Spectra*, 31(S1):S31–S57, 2015.

[MR75] R. M. McMeeking and J. R. Rice. Finite – element formulations for problems of large elastic – plastic deformation. *International Jopurnal of SOlids and Structures*, 11:601–616, 1975.

[MR94] D. B. McCallen and K. M. Romstadt. Analysis of a skewed short span, box girder overpass. *Earthquake Spectra*, 10(4):729–755, 1994.

[MR95] Mayne and Rix. Correlations between shear wave velocity and cone tip resistance in natural clays. *SOILS AND FOUNDATIONS*, 35(2):107–110, 1995.

[MRC79] D. Mayer-Rosa and B. Cadiot. A review of the 1356 Basel earthquake: Basic data,. *Tectonophysics*, 53(3):325–333, 1979. Proceedings of the 16th General Assemble of the European Seismological Commission.

[MRH$^+$16] R. Membarth, O. Reiche, F. Hannig, J. Teich, M. Kärner, and W. Eckert. Hipacc: A domain-specific language and compiler for image processing. *IEEE Transactions on Parallel and Distributed Systems*, 27(1):210–224, Jan 2016.

[Mró88] Zenon Mróz. On proper selection of identification and verification tests. In A. Saada and G. Bianchini, editors, *Constitutive Equations for Granular Non–Cohesive Soils*, pages 721–722. A. A. Balkema, July 1988.

[MRR94] JJ Mason, AJ Rosakis, and G Ravichandran. On the strain and strain rate dependence of the fraction of plastic work converted to heat: an experimental study using high speed infrared detectors and the kolsky bar. *Mechanics of Materials*, 17(2-3):135–145, 1994.

[MRS94] Emir J. Macari–Pasqualino, Kenneth Runesson, and Stein Sture. Response predictions of granular materials at low effective stresses. *Journal of Geotechnical Engineering*, 120(7):1252–1268, July 1994.

[MRS$^+$14] Richard Membarth, Oliver Reiche, Christian Schmitt, Frank Hannig, Jürgen Teich, Markus Sürmer, and Harald Köstler. Towards a performance-portable description of geometric multigrid algorithms using a domain-specific language. *Journal of Parallel and Distributed Computing*, 74(12):3191 – 3201, 2014. Domain-Specific Languages and High-Level Frameworks for High-Performance Computing.

[MS83] Jorge J. Moré and D. C. Sorensen. Computing a trust region step. *SIAM Journal On Scientific And Statistical Computing*, 4(3):553–572, 1983.

[MS94] Thomas H. Massie and J. Kenneth Salisbary. The PHANToM haptic interface: a device for probing virtual objects. *ASME Dynamic Systems and Control*, 55(1):295–301, 1994.

[MS96] G. D. Manolis and R. P. Shaw. Harmonic wave propagation through viscoelastic heterogeneous media exhibiting mild stochasticity: I. Fundamental solution. *Soil Dynamic and Earthquake Engineering*, 15:119–127, 1996.

[MS02] S. Modak and E. D. Sotelino. An object-oriented programming framework for the parallel dynamic analysis of structures. *Computers & Structures*, 80(1):77–84, January 2002.

[MS05] James K. Mitchell and J. Carlos Santamarina. Biological considerations in geotechnical engineering. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 131(10):1222–1233, October 2005.

[MS17] Cornel Marius Murea and Soyibou Sy. Updated lagrangian/arbitrary lagrangian–eulerian framework for interaction between a compressible neo-hookean structure and an incompressible fluid. *International Journal for Numerical Methods in Engineering*, 109(8):1067–1084, 2017.

[MS25] Kevin R. Mackie and Michael H. Scott. Modeling by altering parameter values during time-dependent finite-element simulations. *Journal of Structural Engineering*, 151(5):04025049, 2025.

[MSC01] R.K. McGuire, W.J. Silva, and C.J. Costantino. Technical basis for revision of regulatory guidance on design ground motions: Hazard- and risk-consistent ground motion spectra guidelines. Prepared for Division of Engineering Technology Office of Nuclear Regulatory Research U.S. Nuclear Regulatory Commission Washington, DC 20555-0001 NRC Job Code W6248 NUREG/CR6728, Risk Engineering, Inc., 2001.

[MSF10] Frank McKenna, Michael H. Scott, and Gregory L. Fenves. Nonlinear finite-element analysis software architecture using object composition. *ASCE Journal of Computing in Civil Engineering*, 24(1):95–107, January/February 2010.

[MSGS13] I. Mazzieri, M. Stupazzini, R. Guidotti, and C. Smerzini. SPEED: SPectral Elements in Elastodynamics with Discontinuous Galerkin: a non-conforming approach for 3D multi-scale problems. *Int. J. Numer. Meth. Engng*, 2013.

[MSGT06] George Mylonakis, Costis Syngros, George Gazetas, and Takashi Tazoh. The role of soil in the collapse of 18 piers of Hanshin expressway in the Kobe eqarthquake. *Earthquake Engineering and Structural Dynamics*, 35:547–575, 2006.

[MSH$^+$13] GR Markall, A Slemmer, DA Ham, PHJ Kelly, CD Cantwell, and SJ Sherwin. Finite element assembly strategies on multi-core and many-core architectures. *International Journal for Numerical Methods in Fluids*, 71(1):80–97, 2013.

[MSH16] Greg Mertz, Robert Spears, and Thomas Houston. The effects of discretization errors on the high frequency content of in-structure response spectra. In *Proceedings of the ASME 2016 Pressure Vessels and Piping Conference, PVP2016*, volume PVP2016-63679. ASME, July 17-21 2016.

[MSH$^+$21] H. Motoyama, M. Sawada, W. Hotta, K. Haba, Y. Otsuka, H. Akiba, and M. Hori. Development of a general-purpose parallel finite element method for analyzing earthquake engineering problems. *Earthquake Engineering & Structural Dynamics*, 50(15):4180–4198, 2021.

[MSN$^+$13] Akira Murakami, Takayuki Shuku, Shin-ichi Nishimura, Kazunori Fujisawa, and Kazuyuki Nakamura. Data assimilation using the particle filter for identifying the elasto-plastic material properties of geomaterials. *International Journal for Numerical and Analytical Methods in Geomechanics*, 37(11):1642–1669, 2013.

[MSS99] Christian Miehe, Jörg Scröder, and Jan Schotte. Computational homogenization analysis in finite plasticity: Simulation of texture development in polychristalline materials. *International Journal for Computer Methods in Applied Mechanics and Engineering*, 159:387–418, 1999.

[MSSS] Oswamu Matsuo, Yukiko Saito, Tetsuya Sasaki, and Takashi Sato. Dynamic centrifuge tests on flow failure of fills and infinite slopes.

[MSU$^+$00] Osamu Matsuo, Takao Shimazu, Ryosuke Uzuoka, Masaya Mihara, and Kunio Nishi. Numerical analysis os seismic behavior of embankments founded on liquefiable soils. *Soils and Foundations*, 40(2):21–39, 2000.

[MSW94] C. Miehe, E. Stein, and W. Wagner. Associative multiplicative elasto–plasticity: Formulation and aspects of the numerical implementation including stability analysis. *Computers & Structures*, 52(5):969–978, 1994.

[MSY21] Ryo Morita, Kiyoshi Saito, and Ayumi Yuyama. Development and analysis of seismic experience database of structures, systems and components in nuclear power plants based on investigation reports and maintenance records. *Nuclear Engineering and Design*, 375:111078, 2021.

[MSZA20] Di Mu, Haisheng Shu, Lei Zhao, and Shuowei An. A review of research on seismic metamaterials. *Advanced Engineering Materials*, 22(4):1901148, 2020.

[MT62] RD Mindlin and HF Tiersten. Effects of couple-stresses in linear elasticity. *Archive for Rational Mechanics and analysis*, 11(1):415–448, 1962.

[MT94] Mounir E. Mabsout and John L. Tassoulas. A finite element model for the simulation of pile driving. *International Journal for Numerical Methods in Engineering*, 37:257–278, 1994.

[MT97] Ghiath N. Mansour and John L. Tassoulas. Crossover of integral–ring buckle arrestor: Computational results. *ASCE Journal of Engineering Mechanichs*, 123(4):359–366, April 1997.

[MT05] Eduardo Miranda and Shahram Taghavi. Approximate floor acceleration demands in multistory buildings. i: Formulation. *Journal of structural engineering*, 131(2):203–211, 2005.

[MTB12] Arif Masud, Timothy J. Truster, and Lawrence A. Bergman. A unified formulation for interface coupling and frictional contact modeling with embedded error estimation. *International Journal for Numerical Methods in Engineering*, 92(2):141–177, 2012.

[MTCL17] Panagiotis Martakis, Damoun Taeseri, Eleni Chatzi, and Jan Laue. A centrifuge-based experimental verification of soil-structure interaction effects. *Soil Dynamics and Earthquake Engineering*, 103(Supplement C):1 – 14, 2017.

[MTNG04] B.K. Maheswwari, K.Z. Truman, M.H. El Naggar, and P.L. Gould. Three–dimensional nonlineal analysis for seismic soil–pile–structure interaction. *Soil Dynamics and Earthquake Engineering*, 24:343–356, 2004.

[MU49] Nicholas Metropolis and Stanislaw Ulam. The Monte Carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.

[Mua6a] Y Mualem. A new model for predicting the hydraulic conductivity of unsaturated porous media. *Water Resources Research*, 12(3):513–522, 1976a.

[MUCL97] I. Masters, A. S. Usmani, J. T. Cross, and R. W. Lewis. Domain decomposition with BEM and FEM. *International Journal for Numerical Methods in Engineering*, 40:2891–2909, 1997.

[Mur05] Grigori Muravskii. On description of hysteretic behaviour of materials. *International Journal of Solids and Structures*, 42(9-10):2625–2644, 2005.

[Mus12] Roger Musson. On the nature of logic trees in probabilistic seismic hazard assessment. *Earthquake Spectra*, 28(3):1291–1296, 2012.

[Mut08] Aurelio Muttoni. Punching shear strength of reinforced concrete slabs without transverse reinforcement. *ACI Structural Journal*, –(Title no. 105-S42):440–450, July-August 2008.

[MV87] H B Mühlhaus and I Vardoulakis. The thickness of shear bands in granular materials. *Géotechnique*, 37(3):271–283, 1987.

[MV14] Nicos Makris and Michalis F. Vassiliou. Are some top-heavy structures more stable? *ASCE Journal of Structural Engineering*, DOI: 10.1061/(ASCE)ST.1943-541X.0000933., 2014.

[MVAS89] P. E. McHugh, A. G. Varias, R. J. Asaro, and C. F. Shih. Computational modeling of microstructures. *Future Generation Computer Systems*, 5(2):295–318, 1989.

[MVG$^+$02] G. Marckmann, E. Verron, L. Gornet, G. Chagnon, P. Charrier, and P. Fort. A theory of network alteration for the mullins effect. *Journal of the Mechanics and Physics of Solids*, 50:2011–2028, 2002.

[MW90] David Muir Wood. *Soil Behaviour and Critical State Soil Mechanics*. Cambridge University Press, 1990.

[MW95] Ph. Menetrey and K. J. Willam. Triaxial failure criterion for concrete and its generalization. *ACI Structural Journal*, 92(3):311–318, May–June 1995.

[MW01] J. E. Marsden and M. West. Discrete mechanics and variational integrators. *Acta Numerica*, pages 357–514, 2001.

[MWFM04] Gilberto Mosqueda, Andrew S. Whittaker, Gregory L. Fenves, and Stephen A. Mahin. Experimental and analytical studies of the friction pendulum system for the seismic protection of simple bridges. Technical Report EERC 2004-01, Earthquake Engineering Research Center, University of California, Berkeley, August 2004.

[MWKN23] Faizan Ul Haq Mir, Andrew S. Whittaker, Benjamin D. Kosbab, and Nam Nguyen. Characterizing the seismic response of a molten salt nuclear reactor. *Earthquake Engineering & Structural Dynamics*, 52(7):2025–2046, 2023.

[MWS95] Shuke Miao, Ming L. Wang, and Howard L. Schreyer. Constitutive models for healing of materials. In Stein Sture, editor, *Proceedings of 10th Conference*, pages 1135–1138. Engineering Mechanics Division of the American Society of Civil Engineers, May 1995.

[MYP09] M.A. Millá, Y.L. Young, and J.H. Prévost. Seismic response of intake towers including dam-tower interaction. *Earthquake Engineering & Structural Dynamics*, 38(3):307–329, 2009.

[MYT$^+$] Faizan Ul Haq Mir, Ching-Ching Yu, Mohamed M. Talaat, Benjamin M. Carmichael, Brandon M. Chisholm, and Andrew S. Whittaker. Risk-informed, performance-based design of a seismic isolation system for a nuclear power plant. *Earthquake Engineering & Structural Dynamics*, n/a(n/a).

[MYT$^+$25] Faizan Ul Haq Mir, Ching-Ching Yu, Mohamed M. Talaat, Benjamin M. Carmichael, Brandon M. Chisholm, and Andrew S. Whittaker. Risk-informed, performance-based design of a seismic isolation system for a nuclear power plant. *Earthquake Engineering & Structural Dynamics*, 54(9):2231–2245, 2025.

[MZ93a] Ph. Menéntrey and Th. Zimmermann. Object–oriented non–linear finite element analysis: Application to J2 plasticity. *Computers and Structures*, 49(5):767–777, 1993.

[MZ93b] Ph. Menéntrey and Th. Zimmermann. Object–oriented non–linear finite element analysis: Application to J2 plasticity. *Computers and Structures*, 49(5):767–77, 1993.

[MZ00] Micos Makris and Jian Zhang. Time-domain viscoelastic analysis of earth structures. *Earthquake Engineering and Structural Dynamics*, 29:745–768, 2000.

[MZL$^+$12] Z.S. Ma, Y.C. Zhou, S.G. Long, X.L. Zhong, and C. Lu. Characterization of stress-strain relationships of elastoplastic materials: An improved method with conical and pyramidal indenters. *Mechanics of Materials*, 54:113 – 123, 2012.

[MZML98] Michael McVay, Linmin Zhang, Thomas Molnit, and Peter Lai. Centrifuge testing of large laterally loaded pile groups in sands. *Journal of Geotechnical and Geoenvironmental Engineering*, 124(10):1016–1026, October 1998.

[NA16] National Aeronautics NASA and Space Administration. Standard for models and simulations. Technical Report NASA-STD-7009A, NASA, National Aeronautics and Space Administration, 2016.

[Nae95] Farzad Naeim. On seismic design implications of the 1994 northridge earthquake records. *Earthquake Spectra*, 11(1):91–108, 1995.

[Nag82] Joop C. Nagtegaal. On the implementation of inelastic constitutive equations with special, reference to large deformation problems. *Computer Methods in Applied Mechanics and Engineering*, 33:469–484, 1982.

[Nag85a] P. M. Naghdi. Recent developments in finite deformation plasticity. In A. Sawczuk and G. Bianchi, editors, *Plasticity Today: Modelling, Methods and Applications*, pages 75–83. Elsevier Applied Sciences Publishers, 1985.

[Nag85b] Joop C. Nagtegaal. Discussion of: On the computational significance of the intermediate configuration and hyperelastic stress relations in finite deformation elastoplasticity. *Mechanics of Materials*, 4:453–455, 1985.

[Nak07] Naohiro Nakamura. Practical causal hysteretic damping. *Earthquake Engineering & Structural Dynamics*, 36(5):597–617, 2007.

[NAS08] NASA. Standard for models and simulations, NASA-STD-7009, July 2008.

[NAS16] NASA. Standard for models and simulations, NASA-STD-7009A-Change-1, December 2016.

[Nata] National Center for Supercomputing Applications. The Integrated George E. Brown, Jr. Network for Earthquake Engineering Simulation (NEES) Network (NEESgrid). `http://www.neesgrid.org`.

[Natb] National Partnership for Advanced Computational Infrastructure (NPACI). NPACI HotPage Grid Computing Portal. `https://hotpage.npaci.edu/`.

[Nat15] National Earthquake Hazard Reduction Program (NEHRP). 2015 NEHRP Recommended Seismic Provisions: Design Examples. Technical Report FEMA P-1051, Federal Emergency Management Agency (FEMA), 2015.

[NAV86] NAVFAC. *Foundations & Earth Structures, DESIGN MANUAL 7.02*. Naval Facilities Engineering Command, 200 Stovall Street, Alexandria, Virginia 22332-2300, USA, revalidated by change 1 september 1986 edition, 1986.

[NAY10] M Nehdi, M Shahria Alam, and MA Youssef. Development of corrosion-free concrete beam–column joint with adequate seismic energy dissipation. *Engineering Structures*, 32(9):2518–2528, 2010.

[NBA96] Yacoub M. Najjar, Imad A. Basheer, and Hossam A. Ali. Modeling stress–strain response of clay using nerual nets. In Y. K. Lin and T. C. Su, editors, *Proceedings of 11th Conference*, pages 697–700. Engineering Mechanics Division of the American Society of Civil Engineers, May 1996.

[NBB$^+$01] D. Niemeier, R. W. Boulanger, P. V. Bayly, S. R. Schmid, K. K. Muraleetharan, and A. Barros. Integration of engineering education and research: Perspectives from the NSF Civil and Mechanical Systems 1998 CAREER workshop. *Journal of Engineering Education*, April 2001.

[NBHA08] Jinsuo Nie, Joseph I. Braverman, Charles H. Hofmayer, and Syed A. Ali. Evaluation of simplified methods for estimating shear capacity using jnes/nupec low-rise concrete shear wall cyclic test data. In *Proceedings of PVP2008*. ASME, July 2008.

[NC10] A. Nouy and A. Cléent. eXtended stochastic finite element method for the numerical simulation of heterogeneous materials with random material interfaces. *International Journal for Numerical Methods in Engineering*, Published online in Wiley Online Library (wileyonlinelibrary.com)(DOI: 10.1002/nme.2865):1–33, 2010.

[NC13] Naoki Nakatani and Garnet Kin-Lic Chan. Efficient tree tensor network states (TTNS) for quantum chemistry: Generalizations of the density matrix renormalization group algorithm. *The Journal of chemical physics*, 138(13):134113, 2013.

[NCA09] M. Nazem, J.P. Carter, and D.W. Airey. Arbitrary lagrangian–eulerian method for dynamic analysis of geotechnical problems. *Computers and Geotechnics*, 36(4):549–557, May 2009.

[NCAB16] Nicola A. Nodargi, Federica Caselli, Edoardo Artioli, and Paolo Bisegna. A mixed tetrahedral element with nodal rotations for large-displacement analysis of inelastic structures. *International Journal for Numerical Methods in Engineering*, 108(7):722–749, 2016. nme.5232.

[NCGW12] K. Nissen, J.C. Christian, V. Gravemeier, and A.W. Wolfgang. Information-flux method: a meshfree maximum-entropy petrov-galerkin method including stabilised finite element methods. *Computer Methods in Applied Mechanics and Engineering*, 241-244(0):225 – 237, 2012.

[NCL$^+$11] François Nicot, Noel Challamel, Jean Lerbet, Florent Prunier, and Félix Darve. Bifurcation and generalized mixed loading conditions in geomaterials. *International Journal for Numerical and Analytical Methods in Geomechanics*, 35(13):1409–1431, 2011.

[NCMW07] Patrizio Neff, Krzysztof Chełmiński, Wolfgang Müller, and Christian Wieners. A numerical solution method for an infinitesimal elasto-plastic Cosserat model. *Mathematical Models and Methods in Applied Sciences*, 17(08):1211–1239, aug 2007.

[NCV$^+$20] Vincenzo De Novellis, Vincenzo Convertito, Sotiris Valkaniotis, Francesco Casu, Riccardo Lanari, Mario Fernando Monterroso Tobar, and Nicola Alessandro Pino. Coincident locations of rupture nucleation during the 2019 Le Teil earthquake, France and maximum stress change from local cement quarrying. *Communications Earth & Environment*, 1(20), 2020.

[ND94a] David E. Newcomb and Andrew Drescher. Engineering properties of shredded tires in lightweight fill applications. *Transportaion Research Record*, 1437:1–7, 1994.

[ND94b] Tang–Tat Ng and Ricardo Dobry. Numerical simulations of monotonic and cyclic loading of granular soil. *ASCE Journal of Geotechnical Engineering*, 120(2):388–403, February 1994.

[ND06] François Nicot and Félix Darve. Micro-mechanical investigation of material instability in granular assemblies. *International Journal of Solids and Structures*, 43(11-12):3569–3595, June 2006.

[ND07a] François Nicot and Félix Darve. Micro-mechanical bases of some salient constitutive features of granular materials. *International Journal of Solids and Structures*, 44(22-23):7420–7443, 2007.

[ND07b] François Nicot and Félix Darve. A micro-mechanical investigation of bifurcation in granular materials. *International Journal of Solids and Structures*, 44(20):6630–6652, 2007.

[NDH19a] Kamaljyoti Nath, Anjan Dutta, and Budhaditya Hazra. An iterative polynomial chaos approach toward stochastic elastostatic structural analysis with non-gaussian randomness. *International Journal for Numerical Methods in Engineering*, 0(0), 2019.

[NDH19b] Kamaljyoti Nath, Anjan Dutta, and Budhaditya Hazra. An iterative polynomial chaos approach towards stochastic elastostatic structural analysis with non-gaussian randomness. *International Journal for Numerical Methods in Engineering*, 0, 2019.

[NDIN18a] Mijo Nikolić, Xuan Nam Do, Adnan Ibrahimbegovic, and Željana Nikolić. Crack propagation in dynamics by embedded strong discontinuity approach: Enhanced solid versus discrete lattice model. *Computer Methods in Applied Mechanics and Engineering*, 2018.

[NDIN18b] Mijo Nikolić, Xuan Nam Do, Adnan Ibrahimbegović, and Željana Nikolić. Crack propagation in dynamics by embedded strong discontinuity approach: Enhanced solid versus discrete lattice model. *Computer Methods in Applied Mechanics and Engineering*, 340:480 – 499, 2018.

[NdJ81] J. C. Nagtegaal and J. E. de Jong. Some aspects of non–isotropic workhardening in finite strain plasticity. In E. H. Lee and R. L. Mallet, editors, *Plasticity of Metals at finite strains*, pages 65–106. Stanford University, 1981.

[NDK92] N Navayogarajah, CS Desai, and PD Kiousis. Hierarchical single-surface model for static and cyclic behavior of interfaces. *Journal of engineering mechanics*, 118(5):990–1011, 1992.

[Nee88] A. Needleman. Material rate dependence and mesh sensitivity in localization problems. *Computer Methods in Applied Mechanics and Engineering*, 67:69–85, 1988.

[Nee90] A Needleman. An analysis of decohesion along an imperfect interface. In *Non-Linear Fracture*, pages 21–40. Springer, 1990.

[Nee94] A. Needleman. Computational modeling of material failure. *Applied Mechanics Review*, 47(6):s32–s42, 1994. part 2.

[NEH12] NEHRP Consultants Joint Venture. Soil structure interaction for building structures. Technical Report NIST GCR 12-917-21, National Institute of Standards and Technology, 2012.

[NEH15] National Earthquake Hazard Reduction Program NEHRP. Recommended seismic provisions for new buildings and other structures, fema p-1050-1. Technical Report 2015 Edition, National Earthquake Hazard Reduction Program, 2015.

[Nem82] S. Nemat–Nasser. On finite deformation elasto–plasticity. *International Journal of Solids and Structures*, 18(10):857–872, 1982.

[Nem83] S. Nemat–Nasser. On finite plastic flow of crystalline solids and geomaterials. *Journal of Applied Mechanics*, 50:1114–1126, 1983.

[New59] Nathan. M. Newmark. A method of computation for structural dynamics. *ASCE Journal of the Engineering Mechanics Division*, 85:67–94, July 1959.

[Ng99] Tang-Tat Ng. Compaction process of granular particles. In N.P. Jones and Roger G. Ghanem, editors, *the Proceedings of the 13th ASCE Engineering Mechanics Division Specialty Conference*, Johns Hopkins University, Baltimore, June 13-16 1999. CD-ROM.

[Ng00] Tang-Tat Ng. Private communications, 1999, 2000.

[Ng04] Tang-Tat Ng. Behavior of ellipsoids of two sizes. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 130(10):1077–1083, October 2004.

[NGCM97] Kuo Neng G. Chang and Jay N. Meegoda. Micromechanical simulations of hot mix asphalt. *ASCE Journal of Engineering Mechanics*, 123(5):495–503, 1997.

[Ngu77a] Quoc Son Nguyen. On the elastic plastic inital–boundary value problem and its numerical integration. *International Journal for Numerical Methods in Engineering*, 11:817–832, 1977.

[Ngu77b] Quoc Son Nguyen. On the elastic plastic initial - boundary value problem and it numerical integration. *International Journal for Numerical Methods in Engineering*, 11:817–832, 1977.

[Nie05] Eric Niebler. xpressive: Dual-mode dsel library design. In David Musser, editor, *Library-Centric Software Design LCSD'05*. Object-Oriented Programming, Systems, Languages and Applications, October 2005.

[Nie] Andrzej Niemunis. Private correspondence, 2015 –.

[Nik96a] Victor N. Nikolaevskiy. Inelastic strains of porous saturated media. In Y. K. Lin and T. C. Su, editors, *Proceedings of 11th Conference*, pages 927–930. Engineering Mechanics Division of the American Society of Civil Engineers, May 1996.

[Nik96b] Viktor Nikolaevich Nikolaevskiy. *Geomechanics and fluidodynamics: with applications to reservoir engineering*. Kluwer Academic Publishers, Dodrecht, 1996.

[Nik08] V. N. Nikolaevskiy. Non-linear evolution of p-waves in viscous-elastic granular saturated media. *Transport in Porous Media*, 73(2):125–140, 2008.

[NIM10] T. Nishikawa, H Inoue, and S. Motohashi. Ssi effects of kashiwazaki-kariwa NPP at nco (niigataken chuetsu-oki) earthquake in 2007. In *Proceedings of the OECD – NEA – IAGE – ISSC Workshop on Soil Structure Interaction Knowledge and Effect on the Seismic Assessment of NPPs Structures and Components*, 2010.

[NK06] Hyuk-Chun Noh and Hyo-Gyoung Kwak. Response variability due to randomness in poisson's ratio for plane-strain and plane-stress states. *International Journal of Solids and Structures*, 43(5):1093–1116, 2006.

[NKA] Kien T. Nguyen, Danilo S. Kusanović, and Domniki Asimaki. Three-dimensional nonlinear soil-structure interaction for Rayleigh wave incidence in layered soils. *Earthquake Engineering & Structural Dynamics*, 51(11):2752–2770.

[NKM$^+$98] G. P. Nikishkova, M. Kawkaa, A. Makinouchia, G. Yagawab, and S. Yoshimura. Porting an industrial sheet metal forming code to a distributed memory parallel computer. *Computers & Structures*, 67(6):439–449, June 1998.

[NKTZ06] V. N. Nikolaevskiy, S. M. Kapustyanskiy, M. Thiercelin, and A. G. Zhilenkov. Explosion dynamics in saturated rocks and solids. *Transport in Porous Media*, 65:485–504, 2006. DOI 10.1007/s11242-006-6752-0.

[NLhLK21] Van-Linh Ngo, Changho Lee, Eun haeng Lee, and Jae-Min Kim. Semi-automated procedure to estimate nonlinear kinematic hardening model to simulate the nonlinear dynamic properties of soil and rock. *Applied Science*, 11(8611):1–16, 2021.

[NLSN95] C. W. W. Ng, M. L. Lings, B. Simpson, and D. F. T. Nash. An approximate analysis of the three–dimensional effects of diaphragm wall installation. *Géotechnique*, 45(3):497–507, 1995.

[NLVC05] C. Nouguier-Lehon, E. Vincens, and B. Cambou. Structural changes in granular materials: The case of irregular polygonal particles. *International Journal of Solids and Structures*, 42(24-25):6356–6375, December 2005.

[NM00] A. Naess and V. Moe. Efficient path integration methods for nonlinear dynamic systems. *Probabilistic Engineering Mechanics*, 15(2):221 – 231, 2000.

[NN82] S. Nemat-Nasser. On finite deformation elasto–plasticity. *International Journal for Solids and Structures*, 18(10):857–872, 1982.

[NN85] S. Nemat-Nasser. Micromechanically based finite plasticity. In A. Sawczuk and G. Bianchi, editors, *Plasticity Today: Modelling, Methods and Applications*, pages 85–95. Elsevier Applied Sciences Publishers, 1985.

[NN90] Sia Nemat-Nasser. Certain basic issues in finite–deformation continuum plasticity. *Meccanica*, 25(4):223–229, 1990.

[NNO95] Sia Nemat-Nasser and Naoyuki Okada. Direct observation of deformation of granular materials through X-ray photographs. In Stein Sture, editor, *Proceedings of 10th Conference*, pages 605–608. Engineering Mechanics Division of the American Society of Civil Engineers, May 1995.

[NNO98] S. Nemat-Nasser and N. Okada. Strain localization in particulate media. In H. Murakami and J. E. Luco, editors, *Proceedings of the 12th Engineering Mechanics Conference*. ASCE, May 17-20 1998.

[Nog16] Silvana Montoya Noguera. *Assessment and mitigation of liquefaction seismic risk : numerical modeling of their effects on SSI*. PhD thesis, Université Paris-Saclay, Paris, France, 2016.

[Noh06] Hyuk Chun Noh. Effect of multiple uncertain material properties on the response variability of in-plane and plate structures. *Computer Methods in Applied Mechanics and Engineering*, 195(19-22):2697–2718, April 2006.

[NOKC92] Toyoaki Nogami, Jun Otani, Kazuo Konagai, and Hsiao-Lian Chen. Nonlinear soil-pile interaction model for dynamic lateral motion. *Journal of Geotechnical Engineering*, 118(1):89–106, 1992.

[Nor20] Mark Norris. *How to get started with simulation data management*. NAFEMS, 2020.

[Nov87] Milos Novak. Discussion of "dynamic response of arbitrarily shaped foundations: Experimental verification". *Journal of Geotechnical Engineering*, 113:1410–1412, 1987.

[Nov88] Roberto Nova. "simfonietta classica": An exercise on classical soil modelling. In A. Saada and G. Bianchini, editors, *Constitutive Equations for Granular Non–Cohesive Soils*, pages 501–519. A. A. Balkema, July 1988.

[Nov99] Valentin Valentinovich Novozhilov. *Foundations of the Nonlinear Theory of Elasticity*. Dover Publications, Inc., 1999.

[Nov22] Lukáš Novák. On distribution-based global sensitivity analysis by polynomial chaos expansion. *Computers & Structures*, 267:106808, 2022.

[Now74] Witold Nowacki. *Baudynamik*. Ingenieurbauten. Springer, Vienna, 1974.

[NPF15] Yuri P. Nazarov, Elena Poznyak, and Anton V. Filimonov. A brief theory and computing of seismic ground rotations for structural analyses. *Soil Dynamics and Earthquake Engineering*, 71(0):31 – 41, 2015.

[NPO94] E. A. De Souza Neto, Djordje Perić, and D. R. J. Owen. A model for elastoplastic damage at finite strains: Algorithmic issues and applications. *Engineering Computations*, 11:257–281, 1994.

[NPR74] J. C. Nagtegaal, D. M. Parks, and J. R. Rice. On numerically accurate finite element solution in the fully plastic range. *Computer Methods in Applied Mechanics and Engineering*, 4:153–177, 1974.

[NPT$^+$23] Vicente Navarro, Arianna Pucci, Erik Tengblad, Francesca Casini, and Laura Asensio. Symbolic algebra integration of soil elastoplastic models. *Computers and Geotechnics*, 164:105834, 2023.

[NR19] Quoc Son Nquyen and Dragoš Radenković. Stability of equilibrium in elastic-plastic solids. In __, pages 403–414,. __, 19__.

[NRHO14] Ehsan Nikbakht, Khalim Rashid, Farzad Hejazi, and Siti A Osman. A numerical study on seismic response of self-centring precast segmental columns at different post-tensioning forces. *Latin American Journal of solids and structures*, 11(5):864–883, 2014.

[NS92] M. K. Neilsen and H. L. Schreyer. Bifurcation in elastic-damaging materials. In J. W. Wu and K. C. Valanis, editors, *Damage Mechanics and Localization*, volume AMD-142, MD-34, pages 109 – 123, Th345 East 47th street New York, N.Y. 10017, November 1992. The American Society of Mechanical Engineers.

[NS93] M. K. Neilsen and H. L. Schreyer. Bifurcations in elastic–plastic materials. *International Journal for Solids and Structures*, 30(4):521–544, 1993.

[NS97] André T. Noêl and Barna Szabó. Formulation of geometrically non–linear problems in the spatial reference frame. *International Journal for Numerical Methods in Engineering*, 40:1263–1280, 1997.

[NS10] M. Nenning and M. Schanz. Infinite elements in a poroelastodynamic fem. *INTERNATIONAL JOURNAL FOR NUMERICAL AND ANALYTICAL METHODS IN GEOMECHANICS*, Early View DOI: 10.1002/nag.980, 2010.

[NS18] Lan Nguyen and Jerzy Salamon. Guide for analysis of concrete dam structures using finite element methods. Technical Report DSO-2018-09, US Bureau of Reclamation, Denver, Colorado, USA, 2018.

[NSD09] François Nicot, Luc Sibille, and Félix Darve. Bifurcation in granular materials: An attempt for a unified framework. *International Journal of Solids and Structures*, 46(22-23):3938–3947, November 2009.

[nSR04] H. Xu nd S. Rahman. A generalized dimension-reduction method for multidimensional integration in stochastic mechanics. *International Journal for Numerical Methods in Engineering*, 61(12):1992 – 2019, 2004.

[NSW19] Alexandros Nikellis, Kallol Sett, and Andrew S. Whittaker. Multihazard design and cost-benefit analysis of buildings with special moment-resisting steel frames. *ASCE Journal of Structural Engineering*, 145(5):001–013, 2019.

[NSZA25] Yu Ni, Zhifei Shi, Antonis Zervos, and Ioannis Antoniadis. Periodic wave barriers for the mitigation of surface waves due to moving loads in layered soils. *Soil Dynamics and Earthquake Engineering*, 190:109128, 2025.

[NT94] André C. Nogueira and John L. Tassoulas. Buckle propagation: Steady–state finite–element analysis. *ASCE Journal of Engineering Mechanics*, 120(9):1931–1944, 1994.

[NVI07a] NVIDIA Corporation. CUDA BLAS library. Technical Report PG-00000-002 V1.0, ., June 2007.

[NVI07b] NVIDIA Corporation. CUDA zone. GPGPU compiler, 2007.

[NVM$^+$07] F. Nucera, A. F. Vakakis, D. M. McFarland, L. A. Bergman, and G. Kerschen. Targeted energy transfers in vibro-impact oscillators for seismic mitigation. *Nonlinear Dynamics*, 50:651–677, 2007.

[NW79] R. Nova and D. M. Wood. A constitutive model for sand in triaxial compression. *International Journal for Numerical and Analytical Methods in Geomechanics*, 3:255–278, 1979.

[NW99] Tang-Tat Ng and Changming Wang. Numerical study of arrays of ellipsoids. In N.P. Jones and Roger G. Ghanem, editors, *the Proceedings of the 13th ASCE Engineering Mechanics Division Specialty Conference*, Johns Hopkins University, Baltimore, June 13-16 1999. CD-ROM.

[NXC06] J. Nie, J. Xu, and C. Costantino. P-CARES: Probabilistic computer analysis for rapid evaluation of structures. Technical Report NUREG/CR-6922; BNL-NUREG-77338-2006, Brookhaven National Lab, 2006.

[NY98] Charles W. W. Ng and Ryan W. M. Yan. Stress transfer and deformation mechanisms around a diaphragm wall panel. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 124(7):638–648, July 1998.

[NYS$^+$08] Naohiro Nakamura, Naoto Yabushita, Takuya Suzuki, Jun Yamada, Naohiko Tsunashima, and Tomio Nakano. Analyses of reactor building by 3D nonlinear FEM models considering basemat uplift for simultaneous horizontal and vertical ground motions. *Nuclear Engineering and Design*, 238:3551–3560, 2008.

[NZ72] G. C. Nayak and O. C. Zienkiewicz. Elasto - plastic stress analysis a generalization for various constitutive relations including strain softening. *International Journal for Numerical Methods in Engineering*, 5:113–135, 1972.

[oAE94] American Institute of Aeronautics and Astronautics (AIAA) Editors. AIAA editorial policy statement on numerical accuracy and experimental uncertainty. *AIAA Journal*, 32(1):3–3, 1994.

[OB85] Lorraine G Olson and Klaus Jürgen Bathe. An infinite element for analysis of transient fluid-structure interactions. *Engineering Computations*, 2(4):319–329, 1985.

[OB89] Carlos S. Oliveira and Bruce A. Bolt. Rotational components of strong surface ground motions. *Earthquake Engineering and Structural Dynamics*, 18:517–526, 1989.

[OB23] Tyler J. Oathes and Ross W. Boulanger. Effect of viscoplasticity on localization in saturated clays and plastic silts. *Journal of Geotechnical and Geoenvironmental Engineering*, 149(4):04023016, 2023.

[Obe98] William L. Oberkampf. Bibliography for verificatioon and validation in computational simulations. Technical Report SAND98-2041, Sandia National Laboratory, September 1998.

[Obe03] William Oberkampf. Material from the short course on verification and validation in computational mechanics. Albuquerque, New Mexico. July 2003.

[OBF93] O. A. Olukoko, A. A. Becker, and R. T. Fenner. Three benchmark examples for frictional contact modelling using finite element and boundary element methods. *Journal of Strain Analysis*, 28(4):293–301, 1993.

[OBI17] John Orr, Ana Bras, and Tim Ibell. Effectiveness of design codes for life cycle energy optimisation. *Energy and Buildings*, 140:61–67, 2017.

[OBL03] Catherine O'Sullivan, Jonathan D. Bray, and Shaofan Li. A new approach for calculating strain for particulate media. *International Journal for Numerical an Analytical Methods in Geomechanics*, 27:859–877, 2003.

[OBN^{+}05] J. Tinsley Oden, Ivo Babuška, Fabio Nobile, Yusheng Feng, and Raul Tempone. Theory and methodology for estimation and control of errors due to modeling, approximation, and uncertainty. *Computer Methods in Applied Mechanics and Engineering*, 194(2-5):195–204, February 2005.

[OBPDG11] Adrien Oth, Dino Bindi, Stefano Parolai, and Domenico Di Giacomo. Spectral analysis of K-NET and KiK-net data in Japan, Part II: On attenuation characteristics, source spectra, and site response of borehole and surface stations. *Bulletin of the Seismological Society of America*, 101(2):667–687, 2011.

[OBR02] Catherine O'Sallivan, Jonathan D. Bray, and Michael F. Riemer. Influnce of particle shape and surface friction variability on response of rod–shaped particulate media. *ASCE Journal of Engineering Mechanics*, 128(11):1182–1192, 2002.

[OBR04] Catherine O'Sullivan, Jonathan D. Bray, and Michael Riemer. Examination of the response of regularly packed specimens of spherical particles using physical tests and discrete element simulations. *ASCE Journal of Engineering Mechanics*, 130(10):1140–1150, October 2004.

[OC19] Gerard J. O'Reilly and Gian Michele Calvi. Conceptual seismic design in performance-based earthquake engineering. *Earthquake Engineering & Structural Dynamics*, 48(4):389–411, 2019.

[oCEP03] Working Group on California Earthquake Probabilities. Earthquake probabilities in the San Francisco bay region: 2002–2031. Technical report, U.S. Geological Survey, 2003.

[OCL21] Ozgur Ozcelik, Joel P. Conte, and J. Enrique Luco. Comprehensive mechanics-based virtual model of NHERI@UCSD shake table?uniaxial configuration and bare table condition. *Earathquake Engineering and Structural Dynamics*, 50:3288–3310, 2021.

[ODGM08] Farhang Ostadan, Nan Deng, Orhan Gurbuz, and Sanjeev Malushte. Seismic soil-structure interaction analysis including ground motion incoherency effects. In *The 14th World Conference on Earthquake Engineering*, 2008.

[ODK05] F. Ostadan, N. Deng, and R. Kennedy. Soil structure interaction analysis including ground motion incoherency effects. *18th International Conference on Structural Mechanics in Reactor Technology (SMiRT 18), Beijing, China, SMiRT18-K04-7*, August 2005.

[ODR^{+}02] William L Oberkampf, Sharon M DeLand, Brian M Rutherford, Kathleen V Diegert, and Kenneth F Alvin. Error and uncertainty in modeling and simulation. *Reliability Engineering & System Safety*, 75(3):333–357, 2002.

[ODR04] Farhang Ostadan, Nan Deng, and Jose M. Roesset. Estimating total system damping for soil-structure interaction systems. In M. Celebi (USGS), M.I. Todorovska (USC), I. Okawa (BRI), and M. Iiba (NILIM), editors, *Proceedings Third UJNR Workshop on Soil-Structure Interaction,*, 2004.

[OE82] M. Ortiz and Popov E.P. A statistical-theory of polycrytalline plasticity. *Proceedings of the Royal Society, London, A Materials*, 379:430–458, 1982.

[OF07] Ivo Oprsal and Donat Fäh. 1D vs 3D strong ground motion hybrid modelling of site, and pronounced topography effects at augusta raurica, switzerland-earthquakes or battles. In *Proceedings of 4th International Conference on Earthquake Geotechnical Engineering June 25–28, 2007, Greece*, 2007.

[OG78] Yutaka Ohta and Noritoshi Goto. Empirical shear wave velocity equations in terms of characteristic soil indexes. *Earthquake Engineering and Structural Dynamics*, 6(2):167–187, 1978.

[Ogd84] R. W. Ogden. *Non–Linear Elastic Deformations*. Series in mathematics and its applications. Ellis Horwood Limited, Market Cross House, Cooper Street, Chichester, West Sussex, PO19 1EB, England, 1984.

[OH80] D.R.J. Owen and E. Hinton. *Finite Elements in Plasticity: Theory and Practice*. Pineridge Press, 1980.

[OH81] Miguel Ortiz Herrera. *Topics in Constitutive Theory for Inelastic Solids.* PhD thesis, University of California at Berkeley, 1981.

[O'H06] A. O'Hagan. Bayesian analysis of computer code outputs: A tutorial. *Reliability Engineering and System Safety*, 91(10):1290–1300, 2006.

[O'H14] Anthony O'Hagan. Eliciting and using expert knowledge in metrology. *Metrologia*, 51(4):S237, 2014.

[OHL$^+$08] J.D. Owens, M. Houston, D. Luebke, S. Green, J.E. Stone, and J.C. Phillips. GPU computing. *Proceedings of the IEEE*, 96(5):879–899, May 2008.

[OHP91] Carlso S. Oliveira, Hong Hao, and J. Penzien. Ground motion modeling for multiple–input structural analysis. *Structural Safety*, 10:79–93, 1991.

[Ohs79] Y Ohsaki. On the significance of phase content in earthquake ground motions. *Earthquake Engineering & Structural Dynamics*, 7(5):427–439, 1979.

[OIZT96] E. Onate, S. Idelshon, O.C. Zienkiewicz, and R.L Taylor. A finite point method in computational mechanics. applications to convective transport and fluid flow. *International Journal for Numerical Methods in Engineering*, 39(22):3839 – 3866, 1996.

[oJ12] The National Diet of Japan. Fukushima nuclear accident independent investigation commission. Technical report, The National Diet of Japan, 2012.

[OJA$^+$15] Nebojša Orbović, Boris Jeremić, José Antonio Abell, Chao Luo, Robert P. Kennedy, and Andrei Blahoianu. Use of nonlinear, time domain analysis for NPP design. In *Transactions, SMiRT-23*, Manchester, United Kingdom, August 10-14 2015.

[OL12] Michael O'Rourke and Jack X. Liu. Seismic design of buried and offshore pipelines. Technical Report MCEER-12-MN04, MCEER, 2012.

[OLG$^+$07] John D. Owens, David Luebke, Naga Govindaraju, Mark Harris, Jens Krüger, Aaron E. Lefohn, and Timothy J. Purcell. A survey of general-purpose computation on graphics hardware. *Computer Graphics Forum*, 26(1):80–113, 2007.

[OLN87] Michael Ortiz, Yves Leroy, and Alan Needleman. A finite element method for localized failure analysis. *Computer Methods in Applied Mechanics and Engineering*, 61:189–214, 1987.

[Ols98a] Roy E. Olson. Settlement of embankments on soft clays. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 124(8):659–669, 1998. The Thirty–First Terzaghi Lecture.

[Ols98b] Anders Olsson. An object-oriented implementation of structural path-following. *Computer Methods in Applied Mechanics and Engineering*, 161(1-2):19–47, 1998.

[Ols99] William A. Olsson. Theoretical and experimental investigation compaction of bands in porous rock. *JOURNAL OF GEOPHYSICAL RESEARCH,*, 104(B4):7219–7228, April 1999.

[Ols01] W. A. Olsson. Quasistatic propagation of compaction fronts in porous rock. *Mechanics of Materials*, 33(11):659–668, 2001.

[OM85] JT Oden and JAC Martins. Models and computational methods for dynamic friction phenomena. *Computer methods in applied mechanics and engineering*, 52(1-3):527–634, 1985.

[OM89] Michael Ortiz and John B. Martin. Symmetry–preserving return mapping algorithms and incrementally extremal paths: A unification of concept. *International Journal for Numerical Methods in Engineering*, 28:1839–1853, 1989.

[OM10] Kim B. Olsen and John E. Mayhew. Goodness-of-fit criteria for broadband synthetic seismograms, with application to the 2008 Mw 5.4 Chino Hills, California, earthquake. *Seismological ResearchLetters*, 81(5):715–723, 2010.

[OME$^+$24] Pelin Ozener, M. Murat Monkul, Ece Eseller Bayat, Abdulmuttalip Ari, and Kemal Onder Cetin. Liquefaction and performance of foundation systems in Iskenderun during 2023 Kahramanmaras-Turkiye earthquake sequence. *Soil Dynamics and Earthquake Engineering*, 178:108433, 2024.

[OMG10a] Tinsley Oden, Robert Moser, and Omar Ghattas. Computer predictions with quantified uncertainty, part i. *SIAM News,*, 43(9), November 2010.

[OMG10b] Tinsley Oden, Robert Moser, and Omar Ghattas. Computer predictions with quantified uncertainty, part ii. *SIAM News,*, 43(10), December 2010.

[OMK+09] Makoto Ohsaki, Tomoshi Miyamura, Masayuki Kohiyama, Muneo Hori, Hirohisa Noguchi, Hiroshi Akiba, Koichi Kajiwara, and Tatsuhiko Ine. High-precision finite element analysis of elastoplastic dynamic responses of super-high-rise steel frames. *Earthquake Engineering and Structural Dynamics*, 38(5):635–654, 2009.

[OML+16] Jonathan J. F Ortney, Mark S. Marley, Gregory Laughlin, Nadine Nettelmann, Caroline V. Morley, Roxana E. Lupu, Channon Visscher, Pavle Jeremić, Wade Khadder, and Mason Hargrave. The hunt for planet nine: Atmosphere, spectra, evolution, and detectability. *TBD*, 2016.

[ON12] S. Oladyshkin and W. Nowak. Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion. *Reliability Engineering & System Safety*, 106(0):179 – 190, 2012.

[ONI+03] Hiroshi Okuda, Kengo Nakajima, Mikio Iizuka, Li Chen, and Hisashi Nakamura. Parallel finite element analysis platform for the earth simulator : GeoFEM. Technical report, University of Tokyo, Research Organization for Information Science & Technology, January 21th, 2003.

[ONN94] N Okada and S Nemat-Nasser. Energy dissipation in inelastic flow of saturated cohesionless granular media. *Geotechnique*, 44(1):1–19, 1994.

[OP85] M. Ortiz and E. P. Popov. Accuracy and stability of integration algorithms for elastoplastic constitutive relations. *International Journal for Numerical Methods in Engineering*, 21:1561–1576, 1985.

[OP11] J. Tinsley Oden and Serge Prudhomme. Control of modeling error in calibration and validation processes for predictive stochastic models. *International Journal for Numerical Methods in Engineering*, 87(1-5):262–272, 2011.

[OP17] William L. Oberkampf and Martin Pilch. *Simulation Verification and Validation for Managers*. NAFEMS, 2017.

[Ope06] OpenSees Development Team (Open Source Project). OpenSees: open system for earthquake engineering simulations. `http://opensees.berkeley.edu/`, 2000-2006.

[Ope19] OpenCFD Ltd. *OpenFOAM User Guide*, v1906 edition, 2019.

[OPT98] Miguel Ortiz, Peter M. Pinsky, and Robert L. Taylor. Operator split methods for the numerical solution of the elastoplastic dynamic problem. *Computer Methods in Applied Mechanics and Engineering*, 39:137–157, 198.

[OPT07] William Louis Oberkampf, Martin M Pilch, and Timothy Guy Trucano. Predictive capability maturity model for computational modeling and simulation. *OSTI*, (SAND2007-5948, TRN: US201009

[OR91] Niels Saabye Ottosen and Keneth Runesson. Properties of discontinuous bifurcation solutions in elasto–plasticity. *International Journal of Solids and Structures*, 27(4):401–421, 1991.

[OR96] Niels Saabye Ottosen and Matti Ristinmaa. Corners in plasticity – Koiter's theory revisited. *International Journal of Solids and Structures*, 33(25):3697–3721, 1996.

[OR10] William L. Oberkampf and Christopher J. Roy. *Verification and Validation in Scientific Computing*. Cambridge University Press, 2010.

[Ord00] Gustavo A Ordonez. *SHAKE2000: A computer program for the 1D analysis of geotechnical earthquake engineering problems*. Geomotions, LLC, USA, 2000.

[Ort85] Michael Ortiz. A constitutive theory for the inelastic behavior of concrete. *Mechanics of Materials*, 4:67–93, 1985.

[Ort96] Miguel Ortiz. Obituary to Juan Carlos Simo 1952–1994. *International Journal of Solids and Structures*, 33(20-22):2859–2861, 1996.

[ORV13] S Osovski, D Rittel, and A Venkert. The respective influence of microstructural and thermal softening on adiabatic shear localization. *Mechanics of Materials*, 56:11–22, 2013.

[OS86] M. Ortiz and J. C. Simo. An analysis of a new class of integration algorithms for elastoplastic, constitutive relations. *International Journal for Numerical Methods in Engineering*, 23:353–366, 1986.

[OS96] David R. O'Hallaron and Jonathan Richard Shewchuk. Properties of a family of parallel finite element simulations. Technical Report CMU-CS-96-141, School of Computer Science, Carnegie Mellon University, 23 December 1996.

[Osi] Vladimir A. Osinov. The u-p approximation versus the exact dynamic equations for anisotropic fluid-saturated solids. ii. harmonic waves. *International Journal for Numerical and Analytical Methods in Geomechanics*, n/a(n/a).

[Ost05] Farhang Ostadan. Seismic soil pressure for building walls: An updated approach. *Soil Dynamics and Earthquake Engineerings*, 25:785–793, 2005.

[Ost07a] Farhand Ostadan. Sassi2000 a system for analysis of soil-structure interaction, version 3, theoretical manual. Commercial Program, April 2007.

[Ost07b] Farhand Ostadan. Sassi2000 a system for analysis of soil-structure interaction, version 3, user's manual. Program Manual, April 2007.

[Ost17] Farhang Ostadan. Advanced nuclear technology: High-frequency seismic loading evaluation for standard nuclear power plants. Technical report, Bechtel Corporation, 2017.

[OSW99] M. Ostoja-Starzewski and X. Wang. Stochastic finite elements as a bridge between random material microstructure and global response. *International Journal for Computer Methods in Applied Mechanics and Engineering*, 168:35–49, 1999.

[OT02] William L Oberkampf and Timothy G Trucano. Verification and validation in computational fluid dynamics. *Progress in Aerospace Sciences*, 38(3):209–272, 2002.

[OT07] William L Oberkampf and Timothy G Trucano. Verification and validation benchmarks. Technical Report SAN2007-0853, Sandia National Laboratory, Albuquerque, New Mexico 87185 and Livermore, California 94550, February 2007.

[OT08] William L Oberkampf and Timothy G Trucano. Verification and validation benchmarks. *Nuclear engineering and Design*, 238(3):716–743, 2008.

[OT17] John Ousterhout and Tcl/Tk Consortium. Tool command language (Tcl), tool kit (Tk) gui. `https://www.tcl.tk/`, 2017.

[OTH02] William L. Oberkampf, Timothy G. Trucano, and Charles Hirsch. Verification, validation and predictive capability in computational engineering and physics. In *Proceedings of the Foundations for Verification and Validation on the 21st Century Workshop*, pages 1–74, Laurel, Maryland, October 22-23 2002. Johns Hopkins University / Applied Physics Laboratory.

[OTL87] F Ostadan, Wen S Tseng, and K Lilhanand. Application of the flexible volume method to soil-structure interaction analysis of flexible and embedded foundations. In *Structural mechanics in reactor technology*. SMiRT, 1987.

[OTR$^+$15] Nebojša Orbović, Francois Tarallo, Jean-Matthieu Rembach, Genadijs Segals, and Andrei Blahoianu. IRIS 2012 OECD/NEA/CSNI benchmark: Numerical simulations of structural impact. *Nuclear Engineering and Design*, 2015.

[otSESC98] IEEE Life Cycle Data Harmonization Working Group of the Software Engineering Standards Committee. IEEE standard for software maintenance, IEEE std. 1219-1998, 25June 1998.

[OTST05] Richard S. Orr, Leonardo Tunon-Sanjur, and Sener Tinic. Finite element modeling of the ap1000 nuclear island for seismic analyses at generic soil and rock sites. In *Proceedings of the 18th International Conference on Structural Mechanics in Reactor Technology (SMiRT 18)*, volume SMiRT18 -J12-1, pages 2856 – 2867, Beijing, China,, August 7-12 2005.

[OV10] Won Taek Oh and Sai K. Vanapalli. Influence of rain infiltration on the stability of compacted soil slopes. *Computers and Geotechnics*, 37(5):649 – 657, 2010.

[OW96] Ruaidhri O'Connor and John R. Williams. Portable multibody simulations using MPI. With John R. Williams, IESL–MIT, 1996.

[OW98] Farhang Ostadan and William H. White. Lateral seismic soil pressure an updated approach. In *US-Japan SSI Workshop*, September 1998.

[OYKS92] F. Oka, A. Yashima, M. Kato, and K. Sekiguchi. A constitutive model for sand based on the nonlinear hardening rule and its applications. In *Earthquake Engineering, 10th World Conference*, pages 2529–2534, Spain, 1992.

[PA04] B. Puig and J.L. Akian. Non-gaussian simulation using hermite polynomial expansion and maximum entropy principle. *Probabilistic Engineering Mechanics*, 19(4):293–305, 2004.

[PAAP+15] Arben Pitarka, Abdullah Al-Amri, Michael E. Pasyanos, Arthur J. Rodgers, and Robert J. Mellors. Long-period ground motion in the Arabian Gulf from earthquakes in the Zagros mountains thrust belt. *Pure and Applied Geophysics*, 172:2517–2532, 2015.

[Pad78] Joseph Padovan. Applications of 3-d finite element procedures to static and dynamic problems in micropolar elasticity. *Computers & Structures*, 8(2):231–236, 1978.

[Pai94] Geoffrey Miles Paice. *Finite Element Analysis of Stochastic Soils*. PhD thesis, University of Mancheter, 1994.

[Pak94] Ronald Pak. Elasticity and continuum mechanics. Lecture Notes at CU Boulder, august - decembar 1994.

[PAK20] Alexios Papasotiriou, Asimina Athanatopoulou, and Konstantinos Kostinakis. Investigation on engineering demand parameters describing the seismic damage of masonry infilled R/C frames. *Bulletin of Earthquake Engineering*, 18(13):6075–6115, 2020.

[Pal67] Andrew C Palmer. Stress-strain relations for clays: an energy theory. *Géotechnique*, 17(4):348–358, 1967.

[PALJ14] Federico Pisanò, José Abell, Chao Luo, and Boris Jeremić. Effects of soil volume changes on the nonlinear, site response for dry soils. *Geophysical Journal International*, 2014. In Review.

[PAM08] L.A. Padrón, J.J. Aznárez, and O. Maeso. Dynamic analysis of piled foundations in stratified soils by a BEM–FEM model. *Soil Dynamics and Earthquake Engineering*, 28(5):333 – 346, 2008.

[PAM+21] Floriana Petrone, Norman Abrahamson, David McCallen, Arben Pitarka, and Arthur Rodgers. Engineering evaluation of the EQSIM simulated ground-motion database: The San Francisco bay area region. *Earthquake Engineering & Structural Dynamics*, 50(15):3939–3961, 2021.

[PAMM20] Floriana Petrone, Norman Abrahamson, David McCallen, and Mamun Miah. Validation of (not-historical) large-event near-fault ground-motion simulations for use in civil engineering applications. *Earthquake Engineering & Structural Dynamics*, n/a(n/a):19, 2020.

[Pap90] M. Papadrakakis. A truncated Newton–Lanczos method for overcoming limit and bifurcation points. *International Journal for Numerical Methods in Engineering*, 29:1065–1077, 1990.

[Pap97] M. Papadrakakis, editor. *Parallel Solution Methods in Computational Mechanics*. Wiley series in Solving Large–Scale Problems in Mechanics. John Wiley and Sons, 1997. ISBN 0 471 95696 1.

[Pap04] E. Papamichos. Plasticity model for stress-release induced damage. *ASCE International Journal of Geomechanics*, 4(1):13–18, March 2004.

[Par83] K.C. Park. Stabilization of partioned solution procedure for pore fluid-soil interaction analysis. *International Journal for Numerical Methods in Engineering*, 19:1669–1673, 1983.

[Par06] William G. Pariseau. *Design Analysis in Rock Mechanics*. Taylor & Francis, 1st edition, October 2006.

[Par18] Jared Park. A brief review of tensor operations for students of continuum mechanics. *Journal of Applied Engineering Mathematics*, 5:1–4, December 2018.

[Pav14] G. Pavliotis. *Stochastic Processes and Applications*. Springer, 2014.

[PAW85] Young-Ji Park, Alfredo H-S Ang, and Yi Kwei Wen. Seismic damage analysis of reinforced concrete buildings. *Journal of Structural Engineering*, 111(4):740–757, 1985.

[PB02] Achilleas G. Papadimitriou and George D. Bouckovalas. Plasticity model for sand under small and large cyclic strains: A multiaxial formulation. *Soil Dynamics and Earthquake Engineering*, 22(3):191–204, 2002.

[PB04] SS Park and PM Byrne. Practical constitutive model for soil liquefaction. In *Proc., 9th Int. Symp. on Numerical Models in Geomechanics (NUMOG IX)*, pages 181–186. CRC Press, Boca Raton, FL, 2004.

[PBD01] A. G. Papadimitriou, G. D. Bouckovalas, and Y. F. Dafalias. Plasticity model for sand under small and large cyclic strains. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 127(11):973–983, 2001.

[PBNN] Quoc T. Phan, Ha H. Bui, Giang D. Nguyen, and François Nicot. Strain localization in the standard triaxial tests of granular materials: Insights into meso- and macro-scale behaviours. *International Journal for Numerical and Analytical Methods in Geomechanics*, n/a(n/a).

[PC04] D. Perić and A.J.L. Crook. Computational strategies for predictive geology with reference to salt tectonics. *Computer Methods in Applied Mechanics and Engineering*, 193(48-51):5195–5222, December 2004.

[PCC^{+}24] Cosimo Pellecchia, Alessandro Cardoni, Gian Paolo Cimellaro, Marco Domaneschi, Farhad Ansari, and Ahmed Amir Khalil. Progressive collapse analysis of the Champlain Towers South in Surfside, Florida. *Journal of Structural Engineering*, 150(1):04023211, 2024.

[PCL^{+}14] Jong-Beom Park, Youngin Choi, Sang-Jeong Lee, No-Cheol Park, Kyoung-Su Park, Young-Pil Park, and Chan-Il Park. Modal characteristic analysis of the apr1400 nuclear reactor internals for seismic analysis. *Nuclear Engineering and Technology*, 46(5):689–698, 2014.

[PCM^{+}11] M. Pastor, A. H. C. Chan, P. Mira, D. Manzanal, J. A. Fernández Merodo, and T. Blanc. Computational geomechanics: The heritage of Olek Zienkiewicz. *International Journal for Numerical Methods in Engineering*, pages n/a–n/a, 2011.

[PD06] Vissarion Papadopoulos and George Deodatis. Response variability of stochastic frame structures using evolutionary field theory. *Computer Methods in Applied Mechanics and Engineering*, 195(9-12):1050–1074, January 2006.

[PDAG] Daniele Pietrosanti, Maurizio De Angelis, and Agathoklis Giaralis. Experimental seismic performance assessment and numerical modelling of nonlinear inerter vibration absorber (iva)-equipped base isolated structures tested on shaking table. *Earthquake Engineering & Structural Dynamics*, n/a(n/a).

[PDD14] N. Prime, F. Dufour, and F. Darve. Unified model for geomaterial solid/fluid states and the transition in between. *ASCE Journal of Engineering Mechanics*, 140(6), June 2014.

[PDN05] Radu Popescu, George Deodatis, and Arash Nobahar. Effects of random heterogeneity of soil properties on bearing capacity. *Probabilistic Engineering Mechanics*, 20:324–341, 2005.

[PdP15] F. Pisanò and C. di Prisco. A stability criterion for elasto-viscoplastic constitutive relationships. *International Journal for Numerical and Analytical Methods in Geomechanics*, pages n/a–n/a, 2015.

[PdSN98] D. Perić and E. A. de Souza Neto. A new computational model for Tresca plasticity at finite strains with an optimal parametrization in the principal space. *International Journal for Computer Methods in Applied Mechanics and Engineering*, 159:463–489, 1998.

[PE96] AJ Papazoglou and AS Elnashai. Analytical and field evidence of the damaging effect of vertical earthquake ground motion. *Earthquake Engineering & Structural Dynamics*, 25(10):1109–1137, 1996.

[PE07] C. E. Powell and H. C. Elman. Block-diagonal preconditioning for spectral stochastic finite element systems. Technical report, University of Maryland, 2007.

[Pea74] Carl E. Pearson, editor. *Handbook of Applied Mathematics*. Van Nostrand Reinhold Company, 1974.

[PEC78] M.J.N. Priestly, R.J. Evison, and A.J. Carr. Seismic response of structures free to rock on their fondations. *Bulletin of the New Zealand National Society for Earthquake Engineering*, 11(3):141–150, September 1978.

[Pec17] Alain Pecker. Private communications. ..., 2016-2017.

[Pec18] Alain Pecker. Seismic design of foundations in difficult soil conditions: Examples of solutions. In *Recent Advances in Earthquake Engineering in Europe: 16th European Conference on Earthquake Engineering-Thessaloniki 2018*, pages 323–337. Springer, 2018.

[Ped] Dorival M. Pedroso. Caveats of three direct linear solvers for finite element analyses. *International Journal for Numerical Methods in Engineering*, n/a(n/a):e7545.

[Pel12] François Pellegrini. *Scotch and Libscotch 6.0 User's guide*, 2012.

[Per91] Dunja Perić. *Localized Deformation and Failure Analysis of Pressure Sensitive Granular Material*. PhD thesis, University of Colorado at Boulder, 1991.

[Per92] Djordje Perić. On consistent stress rates in solid mechanics: Computational implications. *International Journal for Numerical Methods in Engineering*, 33:799–817, 1992.

[Per93] Djordje Perić. On a class of constitutive equations in viscoplasticity: Formulation and computational issues. *International Journal for Numerical Methods in Engineering*, 36:1365–1393, 1993.

[Pes96] Juan M. Pestana. A unified description of soil behavior. In Y. K. Lin and T. C. Su, editors, *Proceedings of 11th Conference*, pages 281–284. Engineering Mechanics Division of the American Society of Civil Engineers, May 1996.

[PF10] Georgios Petropoulos and Gregory L. Fenves. Interprocessor communication for high performance, explicit time integration. *Engineering with Computers.*, 26(2):149–157, 2010.

[PF24] Claudio M. Perez and Filip C. Filippou. On nonlinear geometric transformations of finite elements. *International Journal for Numerical Methods in Engineering*, 125(17):e7506, 2024.

[PFA02] A. Pérez-Foguet and F. Armero. On the formulation of closest-point projection algorithms in elastoplasticity-part ii: Globally convergent schemes. *International Journal for Numerical Methods in Engineering*, 53(2):331–374, 2002.

[PFC18] Alberto Pavese, Marco Furinghetti, and Chiara Casarotti. Experimental assessment of the cyclic response of friction-based isolators under bidirectional motions. *Soil Dynamics and Earthquake Engineering*, 114:1 – 11, 2018.

[PFG$^+$17] Alain Pecker, Ezio Faccioli, Aybars Gurpinar, Christoph Martin, and Philippe Renault. *An Overview of the SIGMA Research Project, A European Approach to Seismic Hazarad Analysis*, volume 42 of *Geotechnical, Geological and Earthquake Engineering*. Springer, 2017.

[PFH97] A. Pérez-Foguet and A. Huerta. Plastic flow potential for the cone region of the MRS–Lade model, 1997.

[PFH$^+$08] Mark D. Petersen, Arthur D. Frankel, Stephen C. Harmsen, Charles S. Mueller, Kathleen M. Haller, Russell L. Wheeler, Robert L. Wesson, Yuehua Zeng, Oliver S. Boyd, David M. Perkins, Nicolas Luco, Edward H. Field, Chris J. Wills, and Kenneth S. Rukstales. Documentation for the 2008 update of the united states national seismic hazard maps. Technical report, U.S. Geological Survey, 2008.

[PFR01] K. C. Park, C. A. Felippa, and G. Rebel. Interfacing nonmatching FEM meshes: the zero moment rule. Technical Report CU-CAS-01-01, College of Engineering, University of Colorado, Campus Box 429, Boulder, Colorado 80309, January 2001.

[PFRFH00] Agusti Pérez-Foguet, Antonio Rodríguez-Ferran, and Antonio Huerta. Numerical differentiation for non-trivial consistent tangent matrices: an application to the MRS–Lade model. *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, 48(2):159 – 184, 2000.

[PFRFH01] Agustí Pérez-Foguet, Antonio Rodríguez-Ferran, and Antonio Huerta. Consistent tangent matrices for substepping schemes. *Computer Methods in Applied Mechanics and Engineering*, 190(35-36):4627 – 4647, 2001.

[PFRM20] Federico Passeri, Sebastiano Foti, and Adrian Rodriguez-Marek. A new geostatistical model for shear wave velocity profiles. *Soil Dynamics and Earthquake Engineering*, 136:106247, 2020.

[PFTV88] William H. Press, Brian P. Flannery, Seul A. Teukolvsky, and William T. Vetterling. *Numerical Recipes in C, The Art of Scientific Computing*. Cambridge University Press, 1988.

[PG89] C. A. Pekau and V. Gocevski. Elasto–plastiuc model for cemented and pure sand deposits. *Computers and Geotechnics*, 7:155–187, 1989.

[PGE$^+$16] Arben Pitarka, Rengin Gok, Gurban Y Etirmishli, Saida Ismayilova, and Robert Mellors. Ground motion modeling in the eastern caucasus. *Pure and Applied Geophysics*, online:X1–X11, 2016.

[PGF96] G. M. Paice, D. V. Griffiths, and G. A. Fenton. Finite element modeling of settlements on spatially random soil. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 122(9):777–779, September 1996 1996.

[PGH85] D. A. Pecknold, J. Ghaboussi, and T. J. Healey. Snap–through and bifurcation in a simple structures. *ASCE Journal of Engineering Mechanichs*, 111(7):909–922, 1985.

[PGP$^+$] K. C. Park, J. A. González, Y. H. Park, S. J. Shin, J. G. Kim, K. K. Maute, C. Farhat, and C. A. Felippa. Displacement-based partitioned equations of motion for structures: Formulation and proof-of-concept applications. *International Journal for Numerical Methods in Engineering*, n/a(n/a).

[PGWS02] J. L. Pan, A. T. C. Goh, K. S. Wong, and A. R. Selby. Three–dimensional analysis of single pile response to lateral soil movements. *International Journal for Numerical and Analytical Methods in Geomechanics*, 26:747–758, 2002.

[PH93] R. M. V. Pidaparti and A. V. Hudli. Dynamic analysis of structures using object–oriented techniques. *Computers and Structures*, 49(1):149–156, 1993.

[PHCD$^+$12] Christophe Prud Homme, Vincent Chabannes, Vincent Doyeux, Mourad Ismail, Abdoulaye Samake, Goncalo Pena, Cecile Daversin, and Christophe Trophime. Advances in feel++: a domain specific embedded language in c++ for partial differential equations. In *Eccomas' 12-European Congress on Computational Methods in Applied Sciences and Engineering*, 2012.

[Phi86] Aris Philips. A review of quasistatic experimental plasticity and viscoplasticity. *International Journal of Plasticity*, 2:315–322, 1986.

[PHQ02] K.K. Phoon, S.P. Huang, and S.T. Quek. Implementation of Karhunen–Loève expansion for simulation using a wavelet–Galerkin scheme. *Probabilistic Engineering Mechanics*, 17:293–303, 2002.

[PHQ05] K.K. Phoon, H.W. Huang, and S.T. Quek. Simulation of strongly non-gaussian process using Karhunen-Loève expansion. *Probabilistic Engineering Mechanics*, 20:188–198, June 2005.

[PHS09] Christos Papageorgiou, Neil E. Houghton, and Malcolm C. Smith. Experimental testing and analysis of inerter devices. *ASME Journal of Dynamic Systems, Measurement, and Control*, 131(1):1–11, January 2009.

[PI96] Joy M. Pauschke and Anthony R. Ingraffea. Recent innovations in undergraduate civil engineering curriculums. *ASCE Journal of Professional Issues in Engineering Education and Practice*, 122(3):123–133, July 1996.

[PJ14] Federico Pisanò and Boris Jeremić. Simulating stiffness degradation and damping in soils via a simple visco-elastic-plastic model. *Soil Dynamics and Geotechnical Earthquake Engineering*, 63:98–109, 2014.

[PJB19] O. Penner, Yathon J, and B. Bergman. Best practice for seismic response analysis of concrete dams: Proposed industry guidelines. In *United States Society on Dams, Conference, Chicago*, 2019.

[PJF97] K. C. Park, Manoel R. Justino JR., and Carlos A. Felippa. An algebraically partitioned FETI method for parallel structural analysis: Algorithm description. *International Journal for Numerical Methods in Engineering*, 40:2717–2737, 1997.

[PJJ22] Alain Pecker, James J. Johnson, and Boris Jeremić. *Methodologies for Seismic Soil-Structure Interaction Analysis in the Design and Assessment of Nuclear Installations*. Number IAEA-TECDOC-1990. United Nations, International Atomic Energy Agency, UN-IAEA, External Events Safety Section, IAEA, VIC, PO Box 100, 1400 Vienna, Austria, February 2022. IAEAL 21-01471.

[PJY$^+$95] D. Perić, B. Jeremić, T-F. Yang, S. Sture, H-Y. Ko, and Y. Atsushi. The elasto plastic material model: Model description and numerical predictions. Report to: VELACS extension project for the M.I.T. meeting, october 30-31, 1995., University of Colorado at Boulder, 1995.

[PK99a] Kok-Kwang Phoon and Fred H. Kulhawy. Characterization of geotechnical variability. *Canadian Geotechnical Journal*, 36:612–624, 1999.

[PK99b] Kok-Kwang Phoon and Fred H. Kulhawy. Evaluation of geotechnical property variability. *Canadian Geotechnical Journal*, 36:625–639, 1999.

[PK02] E Providas and MA Kattis. Finite element method in plane Cosserat elasticity. *Computers & structures*, 80(27-30):2059–2069, 2002.

[PKG05] P.N. Psarropoulos, G. Klonaris, and G. Gazetas. Seismic earth pressures on rigid and flexible retaining walls. *Soil Dynamics and Earthquake Engineering*, 25(7-10):795 – 809, 2005. 11th International Conference on Soil Dynamics and Earthquake Engineering (ICSDEE): Part 1.

[PKHR12] Camilo Phillips, Albert R. Kottke, Youssef M.A. Hashash, and Ellen M. Rathje. Significance of ground motion time step in one dimensional site response analysis. *Soil Dynamics and Earthquake Engineering*, 43(0):202 – 217, 2012.

[PKLB22] Christos Psarras, Lars Karlsson, Jiajia Li, and Paolo Bientinesi. The landscape of software for tensor computations. *arXiv*, 2103.13756v3:1–16, 29Jun2022 2022.

[PKM+18] Antonio Palermo, Sebastian Krödel, Kathryn H. Matlack, Rachele Zaccherini, Vasilis K. Dertimanis, Eleni N. Chatzi, Alessandro Marzani, and Chiara Daraio. Hybridization of guided surface acoustic modes in unconsolidated granular media by a resonant metasurface. *Phys. Rev. Applied*, 9:054026, May 2018.

[PKMD16] Antonio Palermo, Sebastian Krödel, Alessandro Marzani, and Chiara Daraio. Engineered metabarrier as shield from seismic surface waves. *Scientific Reports*, 6:39356, 12 2016.

[PKS25] Diego Pizarro, Milan Kovarbašić, and Božidar Stojadinović. Hybrid simulation tests of real-scale squat reinforced concrete shear wall specimens. *Journal of Structural Engineering*, 151(4):04025017, 2025.

[PKZ15] Loizos Pelecanos, Stavroula Kontoe, and Lidija Zdravković. A case study on the seismic performance of earth dams. *Géotechnique*, 65(11):923–935, 2015.

[PL98a] Panayaiotis Papadopoulos and Jia Lu. A general framework for the numerical solution of problems in finite elasto-plasticity. *International Journal for Computer Methods in Applied Mechanics and Engineering*, 159:1–18, 1998.

[PL98b] Panayiotis Papadopoulos and Jia Lu. A general framework for the numerical solution of problems in finite elasto-plasticity. *Computer Methods in Applied Mechanics and Engineering*, 159:1–18, 1998.

[PL22] Andrea Panteghini and Rocco Lagioia. An implicit integration algorithm based on invariants for isotropic elasto-plastic models of the Cosserat continuum. *International Journal for Numerical and Analytical Methods in Geomechanics*, 46(12):2233–2267, 2022.

[PLA] PLAXIS. Finite element code for soil and rock analyses. `http://www.plaxis.nl` Delftechpark 26, NL-2628 XH DELFT, The Netherlands.

[PLAM22] F. J. Pinto, C. Ledezma, R. Astroza, and J. A. Abell Mena. Modeling the loss of vibration energy in buildings to elastic-waves using high-fidelity fe modeling and absorbent exterior boundaries. *Journal of Earthquake Engineering*, 26(13):6567–6584, 2022.

[PLC+11] Darrel Palke, Zhongzang Lin, Guoning Chen, Harry Yeh, Paul Vincent, Robert Laramee, and Eugene Zhang. Asymmetric tensor field visualization for surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1979–1988, 2011.

[PLLD09] F. Prunier, F. Laouafa, S. Lignon, and F. Darve. Bifurcation modeling in geomaterials: From the second-order work criterion to spectral analyses. *International Journal for Numerical and Analytical Methods in Geomechanics*, 33(9):1169–1202, 2009.

[PLLE04] Jun Peng, Jinchi Lu, Kincho H. Law, and Ahmed Elgamal. ParCYCLIC: finite element modelling of earthquake liquefaction response on parallel computers. *International Journal for Numerical and Analytical Methods in Geomechanics*, 28:1207–1232, 2004.

[PLLZ99] M. Pastor, T. Li, X. Liu, and O. C. Zienkiewicz. Stabilized low–order finite elements for failure and localization problems in undrained soils and foundations. *International Journal for Computer Methods in Applied Mechanics and Engineering*, 159:219–234, 1999.

[PLV94] E. Papamichos, J.F. Labuz, and I. Vardulakis. A surface instability detection apparatus. *Rock Mechanics and Rock Engineering*, 27(1):37–56, 1994.

[PM18] Antonio Palermo and Alessandro Marzani. Control of love waves by resonant metasurfaces. *Scientific Reports*, 8, 05 2018.

[PM81] St. Pietruszczak and Z. Mróz. Finite element analysis of deformation of strain–softening materials. *International Journal for Numerical Methods in Engineering*, 17:327–334, 81.

[PMMD] Iason Pelekis, Frank McKenna, Gopal S. P. Madabhushi, and Matthew J. DeJong. Finite element modeling of buildings with structural and foundation rocking on dry sand. *Earthquake Engineering & Structural Dynamics*, n/a(n/a).

[PMO91] D. Perić, G. P. Mitchell, and D. R. J. Owen. Some computational issues in strain localization problems for rate independent solids. *International Journal for Numerical Methods in Engineering*, 32:79–101, 1991.

[PMP$^+$12] Giuliano F. Panza, Cristina La Mura, Antonella Peresan, Fabio Romanelli, and Franco Vaccari. *Advances in Geophysics*, volume 53, chapter Seismic Hazard Scenarios as Preventive Tools for a Disaster Resilient Society, pages 93–165. Elsevier Inc., 2012. ISSN 0065-2687.

[PMP$^+$20a] R. Paolucci, I. Mazzieri, G. Piunno, C. Smerzini, M. Vanini, and A.G. Özcebe. Earthquake ground motion modeling of induced seismicity in the Groningen gas field. *Earthquake Engineering & Structural Dynamics*, n/a(n/a):20, 2020.

[PMP$^+$20b] R. Paolucci, I. Mazzieri, G. Piunno, C. Smerzini, M. Vanini, and A.G. Özcebe. Earthquake ground motion modelling of induced seismicity in the Groningen gas field. *Earthquake Engng Struct Dyn.*, pages 135–154, 2020.

[PMS11] S. Dal Pont, F. Meftah, and B.A. Schrefler. Modeling concrete under severe conditions as a multiphase material. *Nuclear Engineering and Design*, 241(3):562 – 572, 2011. The International Conference on Structural Mechanics in Reactor Technology (SMiRT19) Special Section.

[PN96] Kenneth J. Perano and Paul E. Nielsen. SMARTARRAY: A C++ class template for self–describing, resizable, error–resistent arrays. Technical Report SAND95-8207.UC-405, Sandia National Laboratories, Livermore, California, 94551, April 1996.

[POH92] Djordje Perić, D. R. J. Owen, and M. E. Honnor. A model for finite strain elasto–plasticity based on logarithmic strains: Computational issues. *Computer Methods in Applied Mechanics and Engineering*, 94:35–61, 1992.

[Pon99] Maurice J. Ponte. A day in the life of a student in the year 2012 a.d. *Proceedings of the IEEE*, 87(9):1682–1684, 1999.

[Pop73] Sandor Popovics. A numerical approach to the complete stress-strain curve of concrete. *Cement and concrete research*, 3(5):583–599, 1973.

[Pos04] Douglas Post. The coming crisis in computational science. Technical Report LA-UR-04-0388, 2004. Submitted to: Keynote Address, Proceedings of the IEEE International Conference on High Performance Computer Architecture: Workshop on Productivity and Performance in High-End Computing, Madrid, Spain, February 14, 2004.

[Pot61] J Gl Potyondy. Skin friction between various soils and construction materials. *Geotechnique*, 11(4):339–353, 1961.

[POT83] Peter M. Pinsky, Miguel Ortiz, and Robert L. Taylor. Operator split methods in the numerical solution of the finite deformation elastoplastic dynamic problem. *Computers & Structures*, 17(3):345–359, 1983.

[Pou99] Harry G. Poulos. Approximate computer analysis of pile groups subjected to loads and ground movements. *International journal for numerical and analysis methods in geomechanics*, 23:1021–1041, 1999.

[Pou06] Harry G Poulos. Raked piles – virtues and drawbacks. *ASCE Journal of Geotechnics and Geoenvironmental Engineering*, 132(6):795–803, June 2006.

[PP86] J. S. Pressley and H. G. Poulos. Finite element analysis of mechanisms of pile group behavior. *International Journal for Numerical and Analytical Methods in Geomechanics*, 10:213–221, 1986.

[PP96] Jean H. Prevost and Radu Popescu. Constitutive relations for soil materials. *Electronic Journal of Geotechnical Engineering*, October 1996. available at http://139.78.66.61/ejge/.

[PP98] Apostolos S. Papageorgiou and Duoli Pei. A discrete wavenumber boundary element method for study of the 3-d response 2-d scatterers. *Earthquake Engineering & Structural Dynamics*, 27(6):619–638, 1998.

[PP10] Matthias Preisig and Jean H. Prévost. Stabilization procedures in coupled poromechanics problems: A critical assessment. *International Journal for Numerical and Analytical Methods in Geomechanics*, Early View(DOI: 10.1002/nag.951):1–19, 2010.

[PP11] Matthias Preisig and Jean H. Prévost. Stabilization procedures in coupled poromechanics problems: A critical assessment. *International Journal for Numerical and Analytical Methods in Geomechanics*, 35(11):1207–1225, 2011.

[PPC94] V. Prakash, G. Powell, and S. Cambell. *DRAIN-3DX: Base Program Description and User Guide*. University of California, Berkeley, technical report no. ucb/semm-94/07 edition, August 1994.

[PPDC06] Radu Popescu, Jean H. Prevost, George Deodatis, and Pradipta Chakrabortty. Dynamics of nonlinear porous media with applications to soil liquefaction. *Soil Dynamics and Earthquake Engineering*, 26:648–665, 2006.

[PPM$^+$18] Slavomír Parma, Jiří Plešek, René Marek, Zbyněk Hrubý, Heidi P. Feigenbaum, and Yannis F. Dafalias. Calibration of a simple directional distortional hardening model for metal plasticity. *International Journal of Solids and Structures*, 143:113 – 124, 2018.

[PPP17] George Papazafeiropoulos, Vagelis Plevris, and Manolis Papadrakakis. A new energy-based structural design optimization concept under seismic actions. *Frontiers in Built Environment*, 3:44, 2017.

[PPS02] B. Puig, F. Poirion, and C. Soize. Non-gaussian simulation using hermite polynomial expansion: convergences and algorithms. *Probabilistic Engineering Mechanics*, 17(3):253–264, July 2002.

[PPSC89] K. C. Park, E. Pramono, G. M. Stanley, and H. A. Cabiness. The ANS shell elements: Earlier developments and recent improvements. In Ahmed K. Noor, Ted Belytschko, and Juan C. Simo, editors, *Analytical and Computational Models of Shells*, pages 217 – 239. American Society of Mechanical Engineers, New York, N.Y. :, December 1989.

[PQ05] Nicola Parolinia and Alfio Quarteroni. Mathematical models and numerical simulations for the america's cup. *Computer Methods in Applied Mechanics and Engineering*, 194(9-11):1001–1026, 18 March 2005.

[PQLD20] Qiujing Pan, Xingru Qu, Leilei Liu, and Daniel Dias. A sequential sparse polynomial chaos expansion using Bayesian regression for geotechnical reliability estimations. *International Journal for Numerical and Analytical Methods in Geomechanics*, 44(6):874–889, 2020.

[PR94] A. Pothier and J.J. Rencis. Three-dimensional finite element formulation for microelastic solids. *Computers & Structures*, 51(1):1–21, 1994.

[Pra21] L. Prandtl. Über die härte plasticher körper. *Nachrichten von der Keongl. Gesellschaft der Wissenschaften zu Geottingen, Matematisch–physikalische Klasse aus dem Jahre*, pages 74–85, 1921.

[PRB$^+$24] Shiling Pei, Keri L. Ryan, Jeffrey W. Berman, John W. van de Lindt, Steve Pryor, Da Huang, Sarah Wichman, Aleesha Busch, William Roser, Sir Lathan Wynn, Yi en Ji, Tara Hutchinson, Shokrullah Sorosh, Reid B. Zimmerman, and James Dolan. Shake-table testing of a full-scale 10-story resilient mass timber building. *Journal of Structural Engineering*, 150(12):04024183, 2024.

[PRC10] J.-R. Peng, M. Rouainia, and B.G. Clarke. Finite element analysis of laterally loaded fin piles. *Computers & Structures*, 88(21-22):1239 – 1247, 2010.

[Pre81] J.H. Prevost. Consolidation of aneleastic porous media. *Journal of the Engineering Mechanics Division*, 107:169–186, February 1981.

[Pre85a] Jean H. Prevost. A simple plasticity theory for frictional cohesionless soils. *International Journal of Soil Dynamics and Earthquake Engineering*, 4(1):9 – 17, 1985.

[Pre85b] Jean H. Prevost. Wave propagation in fluid-saturated porous media: an efficient finite element procedure. *Soil Dynamics and Earthquake Engineering*, 4(4):183 –202, 1985.

[Pre89] Jean H Prevost. *DYNA1D: a computer program for nonlinear seismic site response analysis technical documentation*. National Center for Earthquake Engineering Research, 1989.

[PRF91] K. Perktold, M. Resch, and H. Florian. Pulsatile non–newtonian flow characteristics in a three–dimensional human carotid bifurcation model. *ASME Journal of Biomechanical Engineering*, 113:464–475, 1991.

[PRS93] Dunja Perić, Kenneth Runesson, and Stein Sture. Prediction of plastic localisation using MRS-Lade model. *Journal of Geothechnical Engineering*, 119(4):639–661, April 1993.

[PRVV20] Anthony Perez-Rivera and Aidcer L. Vidot-Vega. Effects of high-frequency ground-motion spectra in the seismic response of squat RC walls. *ASCE Journal of Structural Engineering*, 146(8):05020002, 2020.

[Prz85] J. S. Przemieniecki. *Theory of Matrix Structural Analysis*. McGraw Hill, New York, 1985.

[Prz00] Jaroslaw Przewlócki. Two–dimensional random field of mechanical soil properties. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 126(4):373–377, 2000.

[PS] N Anders Petersson and Björn Sjögreen. *SW4, version 2.0*. Computational Infrastructure for Geodynamics,. doi: 10.5281/zenodo.1045297, https://doi.org/10.5281/zenodo.1045297.

[PS75] C. Paige and M. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal.*, 12(0):617–629, 1975.

[PS86] K. C. Park and G. M. Stanley. A curved C^0 element based on assumed natural-coordinate strains. *Transactions of ASME*, 53:278–290, June 1986.

[PS87] S. Pietruszak and D. F. F. Stolle. Modeling of sand behaviour under earthquake excitation. *International Journal for Numerical and Analytical Methods in Geomechanics*, 11:221–240, 1987.

[PS97] H.J. Pradlwarter and G.I. Schüller. On advanced monte carlo simulation procedures in stochastic structural dynamics. *International Journal of Non-Linear Mechanics*, 32(4):735 – 744, 1997.

[PS12] N Anders Petersson and Björn Sjögreen. Stable and efficient modeling of anelastic attenuation in seismic wave propagation. *Communications in Computational Physics*, 12(1):193–225, 2012.

[PS14] N Anders Petersson and Björn Sjögreen. Super-grid modeling of the elastic wave equation in semi-bounded domains. *Communications in Computational Physics*, 16(4):913–955, 2014.

[PS15] N Anders Petersson and Björn Sjögreen. Wave propagation in anisotropic elastic materials and curvilinear coordinates using a summation-by-parts finite difference method. *Journal of Computational Physics*, 299:820–841, 2015.

[PS18] N Anders Petersson and Björn Sjögreen. High order accurate finite difference modeling of seismo-acoustic wave propagation in a moving atmosphere and a heterogeneous earth model coupled across a realistic topography. *Journal of Scientific Computing*, –(14):290–323, 2018.

[PSCA09] G. Palmieri, M. Sasso, G. Chiappini, and D. Amodio. Mullins effect characterization of elastomers by multi-axial cyclic tests and optical experimental methods. *Mechanics of Materials*, 41:1059–1067, 2009.

[PSK11] M. Papadrakakis, G. Stavroulakis, and A. Karatarakis. A new era in scientific computing: Domain decomposition methods in hybrid cpu-gpu architectures. *Computer Methods in Applied Mechanics and Engineering*, 200(13-16):1490 – 1508, 2011.

[PSK20] Floriana Petrone, Li Shan, and Sashi Kunnath. Assessment of building robustness against disproportionate collapse. *ASCE Journal of Structural Engineering*, 146(12):04020272, 2020.

[PSP$^+$0] Mark D Petersen, Allison M Shumway, Peter M Powers, Edward H Field, Morgan P Moschetti, Kishor S Jaiswal, Kevin R Milner, Sanaz Rezaeian, Arthur D Frankel, Andrea L Llenos, Andrew J Michael, Jason M Altekruse, Sean K Ahdi, Kyle B Withers, Charles S Mueller, Yuehua Zeng, Robert E Chase, Leah M Salditch, Nicolas Luco, Kenneth S Rukstales, Julie A Herrick, Demi L Girot, Brad T Aagaard, Adrian M Bender, Michael L Blanpied, Richard W Briggs, Oliver S Boyd, Brandon S Clayton, Christopher B DuRoss, Eileen L Evans, Peter J Haeussler, Alexandra E Hatem, Kirstie L Haynie, Elizabeth H Hearn, Kaj M Johnson, Zachary A Kortum, N Simon Kwong, Andrew J Makdisi, H Benjamin Mason, Daniel E McNamara, Devin F McPhillips, Paul G Okubo, Morgan T Page, Fred F Pollitz, Justin L Rubinstein, Bruce E Shaw, Zheng-Kang Shen, Brian R Shiro, James A Smith, William J Stephenson, Eric M Thompson, Jessica A Thompson Jobe, Erin A Wirth, and Robert C Witter. The 2023 US 50-state national seismic hazard model: Overview and implications. *Earthquake Spectra*, 0(0):87552930231215428, 0.

[PSRdS15] Larissa A. Petri, Patricia Sartori, Josuel K. Rogenski, and Leandro F. de Souza. Verification and validation of a direct numerical simulation code. *Computer Methods in Applied Mechanics and Engineering*, 291(0):266 – 279, 2015.

[PSS00] David Pfitzner, John Salmon, and Thomas Sterling. Halo world: Tools for parallel cluster finding in astrophysical n-body simulations. *Journal of Data Mining and Knowledge Discovery special issue on Scalable High-Performance Computing for KDD*, to Appear, 2000.

[PSS18] Siddharth S. Parida, Kallol Sett, and Puneet Singla. An efficient pde-constrained stochastic inverse algorithm for probabilistic geotechnical site characterization using geophysical measurements. *Soil Dynamics and Earthquake Engineering*, 109:132 – 149, 2018.

[PSV21] Roberto Paolucci, Chiara Smerzini, and Manuela Vanini. BB-SPEEDset: A Validated Dataset of Broadband Near-Source Earthquake Ground Motions from 3D Physics-Based Numerical Simulations. *Bulletin of the Seismological Society of America*, 111(5):2527–2545, October 2021.

[PSY08] Roberto Paolucci, Masahiro Shirato, and M Tolga Yilmaz. Seismic behaviour of shallow foundations: Shaking table experiments vs numerical modelling. *Earthquake Engineering & Structural Dynamics*, 37(4):577–595, 2008.

[PTG+22] Maozhu Peng, Yinghui Tian, Christophe Gaudin, Lihai Zhang, and Daichao Sheng. Application of a coupled hydro-mechanical interface model in simulating uplifting problems. *International Journal for Numerical and Analytical Methods in Geomechanics*, 46(17):3256–3280, 2022.

[PTS89] Tej B.S. Pradhan, Fumio Tatsuoka, and Yasuhiko Sato. Experimental stress-dilatancy relations of sand subjected to cyclic loading. *Soils and Foundations*, 29(1):45–64, 1989.

[PTSB13] Arben Pitarka, Hong Kie Thio, Paul Somerville, and Luis Fabian Bonilla. Broadband ground-motion simulation of an intraslab earthquake and nonlinear site response: 2010 Ferndale, California, earthquake case study. *Seismological Research Letters*, 84(5):785–795, September/October 2013.

[PTVF07] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 2007.

[PV95] E. Papamichos and I. Vardoulakis. Shear band formation in sand according to non–coaxial plasticity model. *Géotechnique*, 45(4):649–661, 1995.

[PVM90] E. Papamichos, I. Vardoulakis, and H-B Mühlhaus. Buckling of layered elastic media: A Cosserat-continuum approach and its validation. *International Journal for Numerical and Analytical Methods in Geomechanics*, 14:473–498, 1990.

[PW] Sai Sharath Parsi and Andrew S. Whittaker. Numerical simulations of rocking, keyed graphite blocks in the core of a high-temperature gas reactor. *Earthquake Engineering & Structural Dynamics*, n/a(n/a).

[PW89] Eddy Pramono and Kaspar Willam. Implicit integration of composite yield surfaces with corners. *Engineering Computations*, 6:186–197, 1989.

[PW95] J. M. Pestana and A. J. Whittle. Compression model for cohesionless soils. *Géotechnique*, 45(4):611–631, 1995.

[PW25] Sai Sharath Parsi and Andrew S. Whittaker. Numerical simulations of rocking, keyed graphite blocks in the core of a high-temperature gas reactor. *Earthquake Engineering & Structural Dynamics*, 54(9):2212–2230, 2025.

[PWL79] Aris Phillips and Chong Won Lee. Yield surfaces and loading surfaces. experiments and recommendations. *International Journal of Solids and Structures*, 15:715–729, 1979.

[PYO94] Djordje Perić, Jiangou Yu, and D. R. J. Owen. On error estimates and adaptivity in elastoplastic solids: Applications to the numerical simulation of strain localization in classical and Cosserat continua. *International Journal for Numerical Methods in Engineering*, 37:1351–1379, 1994.

[PZ86] M. Pastor and O. C. Zienkiewicz. A generalized plasticity, hierachical model for sands under monotonic and cyclic loading. In *Proceedings of the 2nd International Symposium on Numerical Models in Geotechnics*, pages 131–150, Ghent, 1986.

[PZ23] Manolis Papadrakakis and Tarek Zohdi. Laudatio for editor thomas j.r. hughes. *Computer Methods in Applied Mechanics and Engineering*, 417:116331, 2023. A Special Issue in Honor of the Lifetime Achievements of T. J. R. Hughes.

[PZAC23] Madura Pathirage, Boqin Zhang, Mohammed Alnaggar, and Gianluca Cusatis. Confinement and alkali-silica reaction in concrete: Review and numerical investigation. *International Journal of Solids and Structures*, 277-278:112341, 2023.

[PZC88] M. Pastor, O. C. Zienkiewicz, and A. H. C. Chan. Generalized plasticity model for three–dimensional sand behaviour. In A. Saada and G. Bianchini, editors, *Constitutive Equations for Granular Non–Cohesive Soils*, pages 535–549. A. A. Balkema, July 1988.

[PZC90] M Pastor, OC Zienkiewicz, and AHC Chan. Generalized plasticity and the modelling of soil behaviour. *International Journal for Numerical and Analvtical Methods in Geomechanics*, 14(3):151–190, 1990.

[PZD21] Renmin Pretell, Katerina Ziotopoulou, and Craig A. Davis. Liquefaction and cyclic softening at balboa boulevard during the 1994 Northridge earthquake. *Journal of Geotechnical and Geoenvironmental Engineering*, 147(2):05020014, 2021.

[QB95] J. Qian and D. E. Beskos. Dynamic interaction between 3-d rigid surface foundations and comparison with the ATC-3 provisions. *Earthquake Engineering & Structural Dynamics*, 24(3):419–437, 1995.

[QCM20] Xin Qian, Anil K. Chopra, and Frank McKenna. Modeling viscous damping in nonlinear response history analysis of steel moment-frame buildings: Design-plus ground motions. Technical Report 2020/01, Pacific Earthquake Engineering Research Center, June 2020.

[QCX19] Dapeng Qiu, Jianyun Chen, and Qiang Xu. Dynamic responses and damage forms analysis of underground large scale frame structures under oblique SV seismic waves. *Soil Dynamics and Earthquake Engineering*, 117:216–220, 2019.

[QEA$^+$20] Zhijian Qiu, Ahmed Ebeido, Abdullah Almutairi, Jinchi Lu, Ahmed Elgamal, P. Benson Shing, and Geoffrey Martin. Aspects of bridge-ground seismic response and liquefaction-induced deformations. *Earthquake Engineering & Structural Dynamics*, n/a(n/a), 2020.

[QH12] T. Qiu and Y Huang. Energy dissipation in nearly saturated poroviscoelastic soil columns during quasi-static compressional excitations. *ASCE Journal of Engineering Mechanics*, 138(10):1263–1274, October 2012.

[QIH$^+$13] Pher Errol Balde Quinay, Tsuyoshi Ichimura, Muneo Hori, Akemi Nishida, and Shinobu Yoshimura. Seismic structural response estimates of a fault-structure system model with fine resolution using multiscale analysis with parallel simulation of seismic-wave propagation. *Bulletin of the Seismological Society of America*, 103(3):2094–2110, 2013.

[QOIQOv08] Gregorio Quintana-Ortí, Francisco D. Igual, Enrique S. Quintana-Ortí, and Robert van de Geijn. Solving dense linear systems on platforms with multiple hardware accelerators. FLAME Working Note # 32 Technical Report TR-08-22., The University of Texas at Austin, Department of Computer Sciences, May 9 2008.

[Qua11] Guiseppe Quaranta. Finite element analysis with uncertain probabilities. *Computer Methods in Applied Mechanics and Engineering*, 200(1-4):114–129, January 2011.

[Qui94] Michael Jay Quinn. *Parallel Computing, Theory and Practice*. McGraw–Hill, 1994. ISBN 0-07-051294-9.

[QZF25] Tongming Qu, Jidong Zhao, and Y.T. Feng. Artificial intelligence for computational granular media. *Computers and Geotechnics*, 185:107310, 2025.

[QZPK25] Hongning Qi, Jian Zhou, Kang Peng, and Manoj Khandelwal. Knowledge structure and research progress in earthquake-induced liquefaction assessment from 2000 to 2023: A scientometric analysis incorporating domain knowledge. *Soil Dynamics and Earthquake Engineering*, 188:109075, 2025.

[RA59] W C Rasmussen and G E Andreasen. Hydrologic budget of the beaverdam creek basin, maryland. Technical report, USGPO, 1959.

[Rac00] R. Rackwitz. Reviewing probabilistic soils modelling. *Computers and Geotechnics*, 26:199–223, 2000.

[RAC$^+$13] Philippe L.A. Renault, Norman A. Abrahamson, Kevin J. Coppersmith, Martin Koller, Philippe Roth, and Andreas Hölker. Probabilistic Seismic Hazard Analysis for Swiss Nuclear Power Plant Sites - PEGASOS Refinement Project. Final Report, Vol. 1-5 . Technical report, Swissnuclear, 2013.

[RACO11] Amy L. Rechenmacher, Sara Abedi, Olivier Chupin, and Andrés D. Orlando. Characterization of mesoscale instabilities in localized granular shear using digital image correlation. *Acta Geotechnica*, 6(4):205–217, 2011.

[RAE23] Ares Rosakis, Mohamed Abdelmeguid, and Ahmed Elbanna. Evidence of early supershear transition in the Mw 7.8 Kahramanmaraş earthquake from near-field records. arXiv, 2023.

[RAK92] K. Runesson, K. Axelsson, and M. Klisinski. Characteristics of constitutive relations in soil plasticity for undrained behavior. *International Journal of Solids and Structures*, 29(3):363–380, 1992.

[Ram81] E. Ramm. Strategies for tracing the nonlinear response near limit points. In W. Wunderlich, E. Stein, and K-J Bathe, editors, *Nonlinear Finite Element Analysis in Structural Mechanics*, pages 63–89. Springer–Werlag Berlini Heidelberg New York. 1981.

[Ram82] Ekkehard Ramm. The Riks/Wempner approach – an extension of the displacement control method in nonlinear analysis. In E. Hinton, D.R.J. Owen, and C. Taylor, editors, *Recent Advances in Non–Linear Computational Mechanics*, chapter 3, pages 63–86. Pineridge Press, Swansea U.K., 1982.

[Ram05] Robert Ramey. Making a boost library. In David Musser, editor, *Library-Centric Software Design LCSD'05*. Object-Oriented Programming, Systems, Languages and Applications, October 2005.

[Ran06] T. Ranf. Opensees model calibration and assessment. Written communication, 2006.

[RAT19] Andrés Reyes, James Adinata, and Mahdi Taiebat. Impact of bidirectional seismic shearing on the volumetric response of sand deposits. *Soil Dynamics and Earthquake Engineering*, 125:105665, 2019.

[Ray98] Eric S. Raymond. The cathedral and the bazaar. *WWW*, 1998. available at: `http://sagan.earthspace.net/~esr/writings/cathedral-bazaar/`

[RB94] Bruce W. Russell and Ned. H. Burns. Fatigue tests on prestressed concrete beams made with debonded strands. *PCI Journal*, 39(6):70–88, July-August 1994.

[RB14] Doriam Restrepo and Jacobo Bielak. Virtual topography: A fictitious domain approach for analyzing free-surface irregularities in large-scale earthquake ground motion simulation. *International Journal for Numerical Methods in Engineering*, pages n/a–n/a, 2014.

[RBB$^+$16] Julie Régnier, Luis-Fabian Bonilla, Pierre-Yves Bard, Etienne Bertrand, Fabrice Hollender, Hiroshi Kawase, Deborah Sicilia, Pedro Arduino, Angelo Amorosi, Domniki Asimaki, Daniela Boldini, Long Chen, Anna Chiaradonna, Florent DeMartin, Marco Ebrille, Ahmed Elgamal, Gaetano Falcone, Evelyne Foerster, Sebastiano Foti, Evangelia Garini, George Gazetas, Céline Gélis, Alborz Ghofrani, Amalia Giannakou, James R. Gingery, Nathalie Glinsky, Joseph Harmon, Youssef Hashash, Susumu Iai, Boris Jeremić, Steve Kramer, Stavroula Kontoe, Jozef Kristek, Giuseppe Lanzo, Annamaria di Lernia, Fernando Lopez-Caballero, Marianne Marot, Graeme McAllister, E. Diego Mercerat, Peter Moczo, Silvana Montoya-Noguera, Michael Musgrove, Alex Nieto-Ferro, Alessandro Pagliaroli, Federico Pisanò, Aneta Richterova, Suwal Sajana, Maria Paola Santisi d'Avila, Jian Shi, Francesco Silvestri, Mahdi Taiebat, Giuseppe Tropeano, Luca Verrucci, and Kohei Watanabe. International benchmark on numerical simulations for 1d, nonlinear site response (PRENOLIN): Verification phase based on canonical cases. *Bulletin of the Seismological Society of America*, 106(5):2112–2135, 2016.

[RBB$^+$18] Philippe Roux, Dino Bindi, Tobias Boxberger, Andrea Colombi, Fabrice Cotton, Isabelle Douste-Bacque, Stéphane Garambois, Philippe Gueguen, Gregor Hillers, Dan Hollis, Thomas Lecocq, and Ildut Pondaven. Toward seismic metamaterials: The METAFORET project. *Seismological Research Letters*, 89(2A):582–593, 01 2018.

[RBC$^+$18] Jenny Ramirez, Andres R. Barrero, Long Chen, Shideh Dashti, Alborz Ghofrani, Mahdi Taiebat, and Pedro Arduino. Site response in a layered liquefiable deposit: Evaluation of different numerical tools and methodologies with centrifuge experimental results. *Journal of Geotechnical and Geoenvironmental Engineering*, 144(10):04018073, 2018.

[RBF96] S. Ramakrishnan, Muniram Budhu, and George Frantziskonis. Constitutive behavior of granular media using a lattice model. In Y. K. Lin and T. C. Su, editors, *Proceedings of 11th Conference*, pages 713–716. Engineering Mechanics Division of the American Society of Civil Engineers, May 1996.

[RBF$^+$20] Jean-François Ritz, Stéphane Baize, Matthieu Ferry, Christophe Larroqu, Laurence Audin, Bertrand Delouis, and Emmanuel Mathot. Surface rupture and shallow fault reactivation during the 2019 Mw 4.9 Le Teil earthquake, France. *COMMUNICATIONS EARTH & ENVIRONMENT*, 1(10), 2020.

[RBH19] Andrei Rykhlevskii, Jin Whan Bae, and Kathryn D. Huff. Modeling and simulation of online reprocessing in the thorium-fueled molten salt breeder reactor. *Annals of Nuclear Energy*, 128:366–379, 2019.

[RBM21] Sylvain Renaud, Najib Bouaanani, and Benjamin Miquel. Experimental, analytical, and finite element assessment of the shear strength of concrete-rock interfaces at different scales. *International Journal for Numerical and Analytical Methods in Geomechanics*, 45(9):1238–1259, 2021.

[RBMS97] Daniel Ridge, Donald J. Becker, Phillip Merkey, and Thomas Sterling. BEOWULF: Harnesing the power of parallelism in a Pile–of–PCs. In *Proceedings, IEEE Aerospace*, 1997. `http://cesdis.gsfc.nasa.gov/beowulf/papers.html`

[RBRCL] Andres Rodriguez-Burneo, José I. Restrepo, Joel P. Conte, and Carlo G. Lai. Continuum soil-structure-interaction model of the LHPOST6 shaking table reaction mass at uc san diego. *Earthquake Engineering & Structural Dynamics*, n/a(n/a).

[RBS^{+}24] Leonardo G. Rodrigues, Andre R. Barbosa, Arijit Sinha, Christopher Higgins, Scott Breneman, Reid B. Zimmerman, Shiling Pei, John W. van de Lindt, Jeffrey W. Berman, Jorge M. Branco, and Luís C. Neves. Analytical and numerical models for wind and seismic design and assessment of mass timber diaphragms. *Journal of Structural Engineering*, 150(2):04023229, 2024.

[RC05] F. L. B. Ribeiro and A. L. G. A. Coutinho. Comparison between element, edge and compressed storage schemes for iterative solutions in finite element analyses. *International Journal for Numerical Methods in Engineering*, 63(4):569–588, 2005.

[RC06] Kyle M. Rollins and Ryan T. Cole. Cyclic lateral load behavior of a pile cap and backfill. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 132(9):1143–1153, September 2006.

[RC09] Azadeh Riahi and John H. Curran. Full 3D finite element Cosserat formulation with application in layered structures. *Applied Mathematical Modelling*, 33(8):3450–3464, 2009.

[RCMG07] Jean-François Remacle, Nicolas Chevaugeon, Émilie Marchandise, and Christophe Geuzaine. Efficient visualization of high-order finite elements. *International Journal for Numerical Methods in Engineering*, 69(4):750–771, 2007.

[RCP22] Mukesh K. Ramancha, Joel P. Conte, and Matthew D. Parno. Accounting for model form uncertainty in Bayesian calibration of linear dynamic systems. *Mechanical Systems and Signal Processing*, 171:108871, 2022.

[RD15] R. Regueiro and Z. Duan. Static and dynamic micropolar linear elastic beam finite element formulation, implementation, and analysis. *Journal of Engineering Mechanics*, 0(0):04015026, 2015.

[RDI06] Frédéric Ragueneau, Norberto Dominguez, and Adnan Ibrahimbegovic. Thermodynamic-based interface model for cohesive brittle materials: application to bond slip in rc structures. *Computer Methods in Applied Mechanics and Engineering*, 195(52):7249–7263, 2006.

[RDP^{+}17] Ellen M. Rathje, Clint Dawson, Jamie E. Padgett, Jean-Paul Pinelli, Dan Stanzione, Ashley Adair, Pedro Arduino, Scott J. Brandenberg, Tim Cockerill, Charlie Dey, Maria Esteva, Fred L. Haan, Matthew Hanlon, Ahsan Kareem, Laura Lowes, Stephen Mock, and Gilberto Mosqueda. DesignSafe: New cyberinfrastructure for natural hazards engineering. *Natural Hazards Review*, 18(3):06017001, 2017.

[Rea90] James Reason. The contribution of latent human failures to the breakdown of complex systems. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, 327(1241):475–484, 1990.

[Red93] Junuthula N. Reddy. *An Introduction to the Finite Element Method*. Series in Mechanical Engineering. McGraww–Hill, 2nd edition, 1993.

[REG23] David Riley, Itai Einav, and François Guillard. A constitutive model for porous media with recurring stress drops: From snow to foams and cereals. *International Journal of Solids and Structures*, 262-263:112044, 2023.

[REH03] D Rittel, N Eliash, and JL Halary. Hysteretic heating of modified poly (methylmethacrylate). *Polymer*, 44(9):2817–2822, 2003.

[Rei99] Sebastian Reich. Backward error analysis for numerical integrators. *SIAM Journal on Numerical Analysis*, 36(5):1549–1570, 1999.

[Rei08] Robert Reitherman. International aspects of the history of earthquake engineering. Technical report, Earthquake Engineering Research Institute, Oakland, California, U.S.A., 2008.

[Res93a] Research Systems, Inc., Boulder. *IDL Reference Guide*, January 1993.

[Res93b] Research Systems, Inc., Boulder. *IDL User's Guide*, January 1993.

[RFB13] D. Roten, D. Fäh, and L.F. Bonilla. High-frequency ground motion amplification during the 2011 tohoku earthquake explained by soil dilatancy. *Geophysical Journal International*, 193:898–904, 2013.

[RFPH97] Antonio Rodriguez-Ferran, Pierre Pegon, and Antonio Huerta. Two stress update algorithms for large strains: Accuracy and numerical implementation. *International Journal for Numerical Methods in Engineering*, 30:4363–4404, 1997.

[RFWS00] A. Rugarcia, R.M. Felder, D.R. Woods, and J.E. Stice. The future of engineering education. i. a vision for a new century. *Chemical Engineering Education*, 34(1):16–25, 2000.

[RG01] Rafael Riddell and Jaime E Garcia. Hysteretic energy spectrum and damage control. *Earthquake Engineering & Structural Dynamics*, 30(12):1791–1816, 2001.

[RGD13] A Romero, Pedro Galvín, and J Domínguez. 3D non-linear time domain fem–bem approach to soil–structure interaction problems. *Engineering Analysis with Boundary Elements*, 37(3):501–512, 2013.

[RGH97] M. Rassem, A. Ghobarah, and A.C. Heidebrecht. Engineering perspective for the seismic site response of alluvial valleys. *Earthquake Engineering & Structural Dynamics*, 26:477–493, 1997.

[RH09] Prishati Raychowdhury and Tara C. Hutchinson. Performance evaluation of a nonlinear Winkler-based shallow foundation model using centrifuge test results. *Earthquake Engineering and Structural Dynamics*, 38(5):679–698, 2009.

[RHPRH19] Eduardo Rojas, Jaime Horta, Mar? L. Pérez-Rea, and Christian E. Hernández. A fully coupled simple model for unsaturated soils. *International Journal for Numerical and Analytical Methods in Geomechanics*, 43(6):1143–1161, 2019.

[RHR19] Milad Roohi, Eric M Hernandez, and David Rosowsky. Nonlinear seismic response reconstruction and performance assessment of instrumented wood-frame buildings-validation using neeswood capstone full-scale tests. *Structural Control and Health Monitoring*, 26(9):e2373, 2019.

[RHR21] Milad Roohi, Eric M. Hernandez, and David Rosowsky. Reconstructing element-by-element dissipated hysteretic energy in instrumented buildings: Application to the van nuys hotel testbed. *Journal of Engineering Mechanics*, 147(1):04020141, 2021.

[RI00] Lymon C Reese and William M Isenhower. Deep foundations in the future. *ASCE Civil Engineering Magazine*, pages A2–A9, February 2000.

[RIBM01] Chao-Hua Lin Ronaldo I. Borja and Francisco J. Montáns. Cam-clay plasticity, part iv: Implicit integration of anisotropic bounding surface model with nonlinear hyperelasticity and ellipsoidal loading function. *Computer Methods in Applied Mechanics and Engineering*, 190(26-27):3293–3323, 2001.

[Ric11] Lewis Fry Richardson. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 210:307–357, 1911.

[Ric13] Richard H. Langdon, Jr. Technical work plan and the project plan for the Department of Energy (DOE) project to develop verification and validation (V&V) test problems and solutions for the system for analysis of soil-structure interaction (SASSI) computer code, 3 January 2013.

[Rie10] Jorge D. Riera. Considerations on model uncertainty and analyst qualifications in soil-structure interaction studies. In *Proceedings of the OECD – NEA – IAGE – ISSC Workshop on Soil Structure Interaction Knowledge and Effect on the Seismic Assessment of NPPs Structures and Components*, 2010.

[Rik72] E. Riks. The application of Newton's method to the problems of elastic stability. *Journal of Applied Mechanics*, 39:1060–1066, December 1972.

[Rik79] E. Riks. An incremental approach to the solution of snapping and buckling problems. *International Journal for Solids and Structures*, 15:529–551, 1979.

[Rit00] D Rittel. An investigation of the heat generated during cyclic loading of two glassy polymers. part i: Experimental. *Mechanics of Materials*, 32(3):131–147, 2000.

[Riv76] R. S. Rivlin. The application of the theory of invariants to the study of constitutive equations. In G. Fichera, editor, *Trends in Applications of Pure Mathematics to Mechanics*, pages 299–310. Pitman Publishing, 1976.

[Riz93] Egidio Rizzi. Localization analysis of damaged materials. Master's thesis, University of Colorado at Boulder, 1993.

[RJL$^+$99] Debin Ren, Weiping Jia, Shouxin Li, Zhongguang Wang, and Zhaoxing Peng. Finite element analysis of elastic stress and the resolved shear stress in the primary slip system of a copper tricrystal. *Physica Status Solidi A-Applied Research*, 171(2):453–466, Feb. 1999.

[RJMR96] Farhang Radjai, Michel Jean, Jean-Jacques Moreau, and Stéphane Roux. Force distributions in dense two-dimensional granular systems. *Phvsical review letters*, 77(2):274, 1996.

[RK93a] Benny Raphael and C. S. Krishnamoorthy. Automating finite element development using object oriented techniques. *Engineering Computations*, 10:267–278, 1993.

[RK93b] Benny Raphael and C. S. Krishnamoorthy. Automating finite element development using object oriented techniques. *Engineering Computations*, 10:267–278, 1993.

[RK93c] Patrick J Roache and Patrick M Knupp. Completed Richardson extrapolation. *International Journal for Numerical Methods in Biomedical Engineering*, 9(5):365–374, 1993.

[RK08] Ellen Rathje and Albert Kottke. Procedures for random vibration theory based seismic site response analyses. A White Paper Report Prepared for the Nuclear Regulatory Commission. Geotechnical Engineering Report GR08-09., The University of Texas., 2008.

[RK14] Ellen Rathje and Albert Kottke. Reply to comment on "comparison of time series and random-vibration theory site-response methods" by V. Graizer. *Bulletin of the Seismological Society of America*, 104(1):547–550, February 2014.

[RKL93] Kenneth Runesson, Marek Klisinski, and Ragnar Larsson. Formulation and implementation of conditions for frictional contact. *Engineering Computations*, 10:3–14, 1993.

[RKM$^+$95] Karl Romstadt, Bruce Kutter, Brian Maroney, Eric Vanderbilt, Matt Griggs, and Yuk Hon Chai. Experimental measurements of bridge abutment behavior. Technical Report UCD-STR-1, University of California, Davis, September 1995.

[RKT10] Ellen M. Rathje, Albert R. Kottke, and Whitney L. Trent. Influence of input motion and site property variabilities on seismic site response analysis. *ASCE Journal of Geotechnics and Geoenvironmental Engineering*, 136(4):607–619, 2010.

[RKW10] William J. Rider, James R. Kamm, and V. Gregory Weirs. Verification, validation and uncertainty quantification workflow in CASL. Technical Report Sand2010-234P, Sandia National Laboratory, December 2010.

[RL93] Kenneth Runesson and Ragnar Larsson. Properties of incremental solutions for dissipative material. *Journal of Engineering Mechanics*, 119:647–667, 1993.

[RL97] E. Rizzi and B. Loret. Qualitative analysis of strain localization part I: Transversely isotropic elasticity and isotropic plasticity. *International Journal of Plasticity*, 13(5):461–499, 1997.

[RL10] Bo Ren and Shaofan Li. Meshfree simulations of plugging failures in high-speed impacts. *Computers & structures*, 88(15):909–923, 2010.

[RL11] Raymond A. Ryckman and Adrian J. Lew. An explicit asynchronous contact algorithm for elastic body-rigid wall interaction. *International Journal for Numerical Methods in Engineering*, Early View(DOI: 10.1002/nme.3266):28, 2011.

[RM88] E. Ramm and A. Matzenmiller. Consistent linearization in elasto - plastic shell analysis. *Engineering Computations*, 5:289–299, 1988.

[RM89] Kenneth Runesson and Zenon Mrŏz. A note on nonassociated plastic flow rules. *International Journal of Plasticity*, 5:639–658, 1989.

[RM14] Bojana V. Rosić and Hermann G. Matthies. Variational theory and computations in stochastic plasticity. *Archives of Computational Methods in Engineering*, pages 1–53, June 2014.

[RMC02] Manuel L. Romero, Pedro F. Miguel, and Juan J. Cano. A parallel procedure for nonlinear analysis of reinforced concrete three-dimensional frames. *Computers & Structures*, 80(16-17):1334–1350, July 2002.

[RMC07] Amy L. Rechenmacher and Zenon Medina-Cetina. Calibration of soil constitutive models with spatially varying parameters. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 133(12):1567–1576, December 2007.

[RMH21] Muhammad Rizwan Riaz, Hiroki Motoyama, and Muneo Hori. Review of soil-structure interaction based on continuum mechanics theory and use of high performance computing. *Geosciences*, 11(2), 2021.

[RMMCB11] Adrian Rodriguez-Marek, Gonzalo A. Montalva, Fabrice Cotton, and Fabian Bonilla. Analysis of single-station standard deviation using the KiK-net data. *Bulletin of the Seismological Society of America*, 101(3):1242–1258, June 2011.

[RMW96] Egidio Rizzi, Giulio Maier, and Kaspar Willam. On failure indicators in multi–dissipative materials. *International Journal of Solids and Structures*, 33(20-22):3187–3214, 1996.

[RNR21] Tonatiuh Rodriguez-Nikl and Mario E. Rodriguez. Effect of displacement and hysteretic energy on earthquake damage in reinforced concrete structures. *Journal of Structural Engineering*, 147(7):04021083, 2021.

[RO11] Christopher J. Roy and William L. Oberkampf. A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing. *Computer Methods in Applied Mechanics and Engineering*, 200(25-28):2131 – 2144, 2011.

[Roa97] Patrick J Roache. Quantification of uncertainty in computational fluid dynamics. *Annual review of fluid Mechanics*, 29(1):123–160, 1997.

[Roa98a] Patrick J. Roache. *Verification and Validation in Computational Science and Engineering*. Hermosa Publishers, Albuquerque, New Mexico, 1998. ISBN 0-913478-08-3.

[Roa98b] Patrick J Roache. Verification of codes and calculations. *AIAA journal*, 36(5):696–702, 1998.

[Roa02] Patrick J Roache. Code verification by the method of manufactured solutions. *Journal of Fluids Engineering*, 124(1):4–10, 2002.

[Rob96] Arch D. Robison. C++ gets faster for scientific computing. *Computers in Physics*, 10(5):458–462, Sept/Oct 1996.

[Rod17] Artie Rodgers. Private communications. SW4 – Real-ESSI connection, 2017.

[Roe98] J. M. Roesset. Seismic design of nuclear power plants-where are we now? *Nuclear Engineering and Design*, 182(1):3–15, 1998.

[ROP91] Keneth Runesson, Niels Saabye Ottosen, and Dunja Perić. Discontinuous bifurcations of elastic-plastic solutions at plane stress and plane strain. *International Journal of Plasticity*, 7:99–121, 1991.

[Ros09] Bojana Rosić. A review of computational stochastic elastoplasticity. Technical Report Informatikbericht Nr. 2008-08, Technische Universität Braunschweig, 2009.

[Row62] P. W. Rowe. The stress–dilatancy relation for static equilibrium of an assembly of particles in contact. *Proceedings of the Royal Society*, 269(1339):500–527, 9 October 1962. Series A Mathematical and Physical Sciences.

[Roy05] Christopher J. Roy. Review of code and solution verification procedures for computational simulation. *Journal of Computational Physics*, 205(1):131 – 156, 2005.

[RPM^{+}13] Richard Regueiro, Ronald Pak, John McCartney, Stein Sture, Beichuan Yan, Zheng Duan, Jenna Svoboda, WoongJu Mun, Oleg Vasilyev, Nurlybek Kasimov, Eric Brown-Dymkoski, Curt Hansen, Shaofan Li, Bo Ren, Khalid Alshibli, Andrew Druckrey, Hongbing Lu, Huiyang Luo, Rebecca Brannon, Carlos Bonifasi-Lista, Asghar Yarahmadi, Emad Ghodrati, and James Colovos. ONR MURI project on soil blast modeling and simulation. In Bo Song, Dan Casem, and Jamie Kimberley, editors, *Conference Proceedings of the Society for Experimental Mechanics Series, Dynamic Behavior of Materials*, volume 1, pages 341–353. Springer, 2013.

[RPM^{+}14] Sanaz Rezaeian, Mark D Petersen, Morgan P Moschetti, Peter Powers, Stephen C Harmsen, and Arthur D Frankel. Implementation of NGA-West2 ground motion models in the 2014 US national seismic hazard maps. *Earthquake Spectra*, 30(3):1319–1333, 2014.

[RPN^{+}08] Arthur Rodgers, N. Anders Petersson, Stefan Nilsson, Björn Sjögreen, and Kathleen McCandless. Broadband waveform modeling of moderate earthquakes in the San Francisco bay area and preliminary assessment of the USGS 3D seismic velocity model. *Bulletin of the Seismological Society of America*, 98(2):969–988, April 2008.

[RPP^{+}18] A. Rodgers, A. Pitarka, N.A. Petersson, B. Sjögreen, and D. McCallen. Broadband (0-4 hz) ground motions for a magnitude 7.0 Hayward fault earthquake with 3D structure and topography. *Geophys. Res. Lett.*, 45, 2018. doi: 10.1002/2017GL076505.

[RPP^{+}20] A. Rodgers, A. Pitarka, R. Pankajakshan, B. Sjögreen, and N. A. Petersson. Regional-Scale 3D Ground Motion Simulations of Mw 7 Earthquakes on the Hayward Fault, Northern California Resolving Frequencies 0-10 Hz and Including Site Response Corrections. *Bulletin of the Seismological Society of America*, 110(6):2862–2881, 2020.

[RPSA89] K. Runesson, E. Pramono, S. Sture, and K. Axelsson. Assessment of a new class of implicit integration scheme for a cone-cap plasticity models. UC at Boulder, 1989.

[RPT^{+}23] Marisol Salva Ramirez, Junghee Park, Marco Terzariol, Jiming Jiang, and J. Carlos Santamarina. Shallow seafloor sediments: Density and shear wave velocity. *Journal of Geotechnical and Geoenvironmental Engineering*, 149(5):04023022, 2023.

[RPW97] Kyle M. Rollins, Kris T. Peterson, and Thomas J. Weaver. Lateral load behavior of full scale pile group in clay. *Journal of geotechnical and geoenvironmental engineering*, 124(6):468–478, June 1997.

[RPW98] Kyle M. Rollins, Kris T. Peterson, and Thomas J. Weaver. Lateral load behavior of full–scale pile group in clay. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 124(6):468–478, June 1998.

[RR75] J. W. Rudnicki and J. R. Rice. Conditions for the localization of deformation in pressure–sensitive dilatant materials. *Journal of the Mechanics and Physics of Solids*, 23:371 to 394, 1975.

[RR98] K. Ranjith and James R. Rice. Stability of quasi–static slip in single degree of freedom elastic system with rate and state dependent friction. *Journal of the Mechanics and Physics of Solids*, 1998. Submitted for publication.

[RR00] D Rittel and Y Rabin. An investigation of the heat generated during cyclic loading of two glassy polymers. part ii: Thermal analysis. *Mechanics of Materials*, 32(3):149–159, 2000.

[RR10] Hwasung Roh and Andrei M Reinhorn. Modeling and seismic response of structures with concrete rocking columns and viscous dampers. *Engineering Structures*, 32(8):2096–2107, 2010.

[RRB^{+}13] Christine Roy, Aaron J. Roffel, Said Bolourchi, Luben Todorovski, and Mahmoud Khoncarly. Study of seismic structure-soil-structure interaction between two heavy structures. In *Transactions, SMiRT-22*, pages 1–7, San Francisco, August 2013. IASMiRT.

[RRH^{+}02] Guruswami Ravichandran, Ares J Rosakis, Jon Hodowany, Phoebus Rosakis, Michael D Furnish, Naresh N Thadhani, and Yasuyuki Horie. On the conversion of plastic work into heat during high-strain-rate deformation. In *AIP conference proceedings*, volume 620/1, pages 557–562. AIP, 2002.

[RRH07] Christopher J Roy, Anil Raju, and Matthew M Hopkins. Estimation of discretization errors using the method of nearby problems. *AIAA journal*, 45(6):1232–1243, 2007.

[RRRH00] P Rosakis, AJ Rosakis, G Ravichandran, and J Hodowany. A thermodynamic internal variable model for the partition of plastic work into heat and stored energy in metals. *Journal of the Mechanics and Physics of Solids*, 48(3):581–607, 2000.

[RRV^{+}09] J.C. Ruegg, A. Rudloff, C. Vigny, R. Madariaga, J.B. de Chabalier, J. Campos, E. Kausel, S. Barrientos, and D. Dimitrov. Interseismic strain accumulation measured by GPS in the seismic gap between Constitución and Concepción in Chile. *Physics of the Earth and Planetary Interiors*, 175:78–85, 2009.

[RS85] K. Runesson and A. Samuelsson. Aspects on numerical techniques in small deformation plasticity. In G. N. Pande J. Middleton, editor, *NUMETA 85 Numerical Methods in Engineering, Theory and Applications*, pages 337–347. AA.Balkema., 1985.

[RS86] J. B. Roberts and P. D. Spanos. Stochastic averaging: An approximate method of solving random vibration problems. *International Journal of Non-Linear Mechanics*, 21(2):111–134, 1986.

[RS87] R. S. Rivlin and G. F. Smith. A note of material frame indifference. *International Journal of Solids and Structures*, 23(12):1639–1643, 1987.

[RS89] K. Runesson and S. Sture. Stability of frictional materials. *Journal of Engineering Mechanics*, 115(8):1828–1833, 1989.

[RS96] P. Rahulkumar and S. Saigal. Implicit integration procedures and consistent tangent operators for bounding surface plasticity models. In Y. K. Lin and T. C. Su, editors, *Proceedings of 11th Conference*, pages 140–143. Engineering Mechanics Division of the American Society of Civil Engineers, May 1996.

[RS09] S. Ruiz and G.R. Saragoni. Free vibration of soils during large earthquakes. *Soil Dynamics and Earthquake Engineering*, 29(1):1–16, January 2009.

[RSB86] Kenneth Runesson, Alf Samuelsson, and Lars Bernspång. Numerical technique in plasticity including solution advancement control. *International Journal for Numerical Methods in Engineering*, 22:769–788, 1986.

[RSC99] A. J. Rosakis, O. Samudrala, and D. Coker. Cracks faster than the shear wave speed. *Science*, 284(5418):1337–1340, 1999.

[RSKY10] Jeong-Soo Ryu, Choon-Gyo Seo, Jae-Min Kim, and Chung-Bang Yun. Seismic response analysis of soil-structure interactive system using a coupled three-dimensional fe-ie method. *Nuclear Engineering and Design*, 240(8):1949 – 1966, 2010.

[RSR$^+$96] Chance Reschke, Thomas Sterling, Daniel Ridge, Daniel Savarese, Donald J. Becker, and Phillip Merkey. A design study of alternative network topologies for the Beowulf parallel workstation. In *Proceedings, High Performance and Distributed Computing*, 1996. `http://cesdis.gsfc.nasa.gov/beowulf/papers.html`

[RSTA96] C.J. Roblee, W.J. Silva, G.R. Toro, and N. Abrahamson. Variability in site-specific seismic ground-motion predictions. In *Proceedings: Uncertainty in the Geologic Environment*, page 21p, Madison WI., August 1-2 1996. American Society of Civil Engineers.

[RSW88] Kenneth Runesson, Stein Sture, and Kaspar Willam. Integration in computational plasticity. *Computers & Structures*, 30(1/2):119–130, 1988.

[RT66] Ernest Rabinowicz and RI Tanner. Friction and wear of materials. *Journal of Applied Mechanics*, 33:479, 1966.

[RT97] Pedro F. Ruesta and Frank C. Townsend. Evaluation of laterally loaded pile group at Roosevelt bridge. *Journal of Geotechnical and Geoenvironmental Engineering*, 123(12):1153–1161, December 1997.

[RT12] Brittani R. Russell and Ashley P. Thrall. Portable and rapidly deployable bridges: Historical perspective and recent technology developments. *ASCE Journal of Bridge Engineering*, 2012.

[RTA23] Ahmad Rahmzadeh, Robert Tremblay, and M. Shahria Alam. Cyclic response of buckling-restrained stainless steel energy dissipating bars. ii: Finite element investigations. *Journal of Engineering Mechanics*, 149(4):04023017, 2023.

[RTFS13] S Hamid Reza Tabatabaiefar, Behzad Fatahi, and Bijan Samali. Seismic behavior of building frames considering dynamic soil-structure interaction. *International Journal of Geomechanics*, 13(4):409–420, 2013.

[RTKM02] J. B. Rundle, K. F. Tiampo, W. Klein, and J. S. Sá Martins. Self–ogranization in leaky threshold systems: The influence of near–mean field dynamics and its implications for earthquakes, neurobilogy, and forecasting. In *Proc. Natl. Acad. Sci. USA*, volume 99, supl 1, pages 2514–2521, February 19, 2002.

[Rud96] J. W. Rudnicki. Development of localization in undrained deformation. In Y. K. Lin and T. C. Su, editors, *Proceedings of 11th Conference*, pages 939–942. Engineering Mechanics Division of the American Society of Civil Engineers, May 1996.

[Run] Kenneth Runesson. Private correspondence, .

[Run78] Kenneth Runesson. *On Non–Linear Consolidation of Soft Clay, a Numerical Approach with Special Emphasis on Plasticity and Creep*. PhD thesis, Department of Structural Mechanics, Chalmers University of Technology, Göteborg, Sweden, January 1978. Publication 78:1.

[Run87] Kenneth Runesson. Implicit integration of elastoplastic relations with reference to soils. *International Journal for Numerical and Analytical Methods in Geomechanics*, 11:315–321, 1987.

[Run94] Kenneth Runesson. Constitutive theory and computational technique for dissipative materials with emphasis on plasticity, viscoplasticity and damage: Part II. Lecture Notes, Chalmers Technical University, Göteborg, Sweden, Novembar 1994.

[Run96] Kenneth Runesson. Constitutive theory and computational technique for dissipative materials with emphasis on plasticity, viscoplasticity and damage: Part III. Lecture Notes, Chalmers Technical University, Göteborg, Sweden, September 1996.

[RW95] Egidio Rizzi and Kaspar Willam. Constitutive singularities of combined elastic degradation and plasticity. In Stein Sture, editor, *Proceedings of 10th Conference*, pages 529–532. Engineering Mechanics Division of the American Society of Civil Engineers, May 1995.

[RWBK95] Carin L. Roberts-Wollman, John E. Breen, and Michael E. Kreger. Temperature induced deformation in match cast segments. *PCI Journal*, 40(4):62–71, July-August 1995.

[RWIA00] L. C. Reese, S. T. Wang, W. M. Isenhower, and J. A. Arrellaga. *LPILE plus 4.0 Technical Manual*. ENSOFT, INC., Austin, TX, version 4.0 edition, Oct. 2000.

[RWJK01] K. Rosebrook, D. W. Wilson, B. Jeremić, and B. Kutter. Centrifuge characterization and numerical modeling of the dynamic properties of tire shreds for use as bridge abutment backfill. In *Fourth International Conference On Recent Advances In Geotechnical Earthquake Engineering And Soil Dynamics*, March 26-31 2001.

[RWM^{+}00] L. C. Reese, S. T. Wang, Isenhower W. M., Arréllaga J. A., and Hendrix J. *LPILE plus 4.0 User Guide*. ENSOFT, INC., Austin, TX, version 4.0 edition, Oct. 2000.

[RX05] S. Rahman and H. Xu. A meshless method for computational stochastic mechanics. *Int. J. Comput. Meth. Eng. Sc. Mech.*, 6(1):41–58, 2005.

[RY96] M. S. Rahman and C. H. Yeh. Variability of seismic response of soils using stochastic finite element method. *Soil Dynamic and Earthquake Engineering*, 18:229–245, 1996.

[Rya94] Harold Ryan. Ricker, Ormsby, Klander, Butterworth - a choice of wavelets. *Canadian Society of Exploration Geophysicists Recorder*, 19(7):8–9, September 1994.

[SA79] J. B. Stevens and J. M. E. Audibert. Re-examination of p-y curve formulations. In *Eleventh Annual Offshore Technology Conference*, volume I, pages 397–403, Dallas, TX, April 1979. Americal Society of Civil Engineers.

[SA11] Barna A Szabó and Ricardo L. Actis. Simulation governance: New technical requirements for software tools in computational solid mechanics. Presentation at the: International Workshop on Verification and Validation in Computational Science University of Notre Dame, 17-19 October 2011.

[SA12] Barna Szabó and Ricardo Actis. Simulation governance: Technical requirements for mechanical design. *Computer Methods in Applied Mechanics and Engineering*, 249-252:158–168, 2012.

[SA25] Max Sieber and Ioannis Anastasopoulos. Seismic performance of a rocking pile group supporting a bridge pier. *Journal of Geotechnical and Geoenvironmental Engineering*, 151(1):04024145, 2025.

[Saa03] Yousef Saad. *Iterative methods for sparse linear systems*, volume 82. siam, 2003.

[SAA^{+}10] Andrea Saltelli, Paola Annoni, Ivano Azzini, Francesca Campolongo, Marco Ratto, and Stefano Tarantola. Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index. *Computer Physics Communications*, 181(2):259 – 270, 2010.

[Sab97] David A. Sabatini. Teaching and research synergism: the undergraduate research experience. *ASCE Journal of Professional Issues in Engineering Education and Practice*, 123(3):98–102, July 1997.

[SAC^{+}23] Luca Sironi, Marco Andreini, Cristiana Colloca, Michael Poehler, Davide Bolognini, Filippo Dacarro, Pierino Lestuzzi, Frédéic Dubois, Ziran Zhou, and José E. Andrade. Shaking table tests for seismic stability of stacked concrete blocks used for radiation shielding. *Engineering Structures*, 283:115895, 2023.

[SAH93] Barna A. Szabó, Ricardo L. Actis, and Stefan M. Holzer. Solution of elastic–plastic stress analysis problems by the p–version of the finite element method. In J. E. Flaherty et al. I. Babuška, editor, *The IMA Volumes in Mathematics and its Application*, volume 75, pages 395–416. University of Minnesota, 1993.

[SAK19] Miad Saberi, Charles-Darwin Annan, and Jean-Marie Konrad. Implementation of a soil-structure interface constitutive model for application in geo-structures. *Soil Dynamics and Earthquake Engineering*, 116:714 – 731, 2019.

[Sal06] Manuel D. Salas. The curious events leading to the theory of shock waves. In *17th Shock Interaction Symposium*, Rome, Italy, 4-8 September 2006.

[Sal08] Rodrigo Salgado. *The Engineering of Foundations*. McGraw Hill, 2008.

[Sal11] Jerzy W. Salamon. Seismic induced loads on spillway gates phase i - literature review. Technical Report DSO-11-06, Uunited States Bureau of Reclamation, Denver, Colorado, 2011.

[SAL12] J Schmedes, Ralph J Archuleta, and Daniel Lavallée. A kinematic rupture model generator incorporating spatial interdependency of earthquake source parameters. *Geophysical Journal International*, 192(3):1116–1131, 2012.

[San] Sandia National Labs. SIERRA: Software environment for developing complex multiphysics applications. `http://www.cfd.sandia.gov/sierra.html`.

[San24] Luis Santos. Deep and physics-informed neural networks as a substitute for finite element analysis. In *Proceedings of the 2024 9th International Conference on Machine Learning Technologies, Oslo, Norway*, ICMLT '24, pages 84–90. Association for Computing Machinery, 2024.

[Sao13] Victor E. Saouma. Prof. Saouma's lecture notes, manuals, &c. (available through his web page: `http://civil.colorado.edu/~saouma/`), 1992-2013.

[Sar06] S.A. Sarra. Integrated multiquadric radial basis function approximation methods. *Computers & Mathematics with Applications*, 51(8):1283 – 1296, 2006. Radial Basis Functions and Related Multivariate Meshfree Approximation Methods: Theory and Applications.

[SAS11] Vincenzo Silvestri and Ghassan Abou-Samra. Application of the exact constitutive relationship of modified Cam clay to the undrained expansion of a spherical cavity. *INTERNATIONAL JOURNAL FOR NUMERICAL AND ANALYTICAL METHODS IN GEOMECHANICS*, 35(1):53–66, January 2011.

[SASF$^+$98] Roy Swanson, Khalid AL-Shibli, Melissa Frank, Nicholas Costes, Stein Sture, Susan Batiste, Mark Lankton, and Boris Jeremić. Mechanics of granular materials in microgravity at low effective stresses. In *Proceedings of the Spring Meeting of the American Geophysical Union*, 1998.

[SAT93] J. C. Simo, F. Armero, and R. L. Taylor. Improved versions of assumed enhanced strain tri–linear elements for 3D finite deformation problems. *Computer Methods in Applied Mechanics and Engineering*, 110:359–386, 1993.

[Sat13] Tadanobu Sato. Fractal characteristics of phase spectrum of earthquake motion. *Journal of Earthquake and Tsunami*, 7(02):1350010, 2013.

[SATC96] Walter Silva, Norman Abrahamson, Giulio Toro, and Carl Costantino. Description and validation of the stochastic ground motion model. Technical Report PE&A 94PJ20, Associated Universities, Inc., November 1996. Brookhaven National Laboratory.

[Sav03] E. Savin. Infuence of free field variability on linear soil–structure interaction (ssi) by indirect integral representation. *Earthquake Engineering & Structural Dynamics*, 32:49–69, 2003.

[SB81] Mark D. Snyder and Klaus-Jürgen Bathe. A solution procedure for thermo-elastic-plastic and creep problems. *Nuclear Engineering and Design*, 64:49–80, 1981. Invited paper, presented at the 5th International Conference on Structural Mechanics in Reactor Technology, Berilin (West), Avgust 13-17 1979.

[SB89] Lásló Szabó and Mihály Balla. Comparison of some stress rates. *International Journal of Solids and Structures*, 25(3):279–297, 1989.

[SB93] Jr. Spencer, B.F. and L.A. Bergman. On the numerical solution of the fokker-planck equation for nonlinear stochastic systems. *Nonlinear Dynamics*, 4(4):357–372, 1993.

[SB96a] L. J. Sluys and A. H. Berends. Embedded localization band elements for mode–I and mode–II failure. In Y. K. Lin and T. C. Su, editors, *Proceedings of 11th Conference*, pages 1181–1184. Engineering Mechanics Division of the American Society of Civil Engineers, May 1996.

[SB96b] AA Stamos and DE Beskos. 3-d seismic response analysis of long lined tunnels in half-space. *Soil Dynamics and Earthquake Engineering*, 15(2):111–118, 1996.

[SB00] Jean-FranÇois Semblat and J. J. Brioist. Efficiency of higher order finite elements for the analysis of seismic wave propagation. *Journal of Sound and Vibration*, 231(2):460–467, 2000.

[SB10] Peter J Stafford and Julian J Bommer. Theoretical consistency of common record selection strategies in performance-based earthquake engineering. In *Advances in Performance-Based Earthquake Engineering*, pages 49–58. Springer, 2010.

[SB16] AE Seifried and JW Baker. Spectral variability and its relationship to structural response estimated from scaled and spectrum-matched ground motions. *Earthquake Spectra*, 32(4):2191–2205, 2016.

[SB20] Carl Sisemore and Vít Babuška. Energy spectra methods. In *The Science and Engineering of Mechanical Shock*, pages 331–344. Springer, 2020.

[SBD76] J.C. Small, J.R. Booker, and E.H. Davis. Elasto-plastic consolidation of soils. *International Journal for Solids and Structures*, 12:431–448, 1976.

[SBJ17] Sumeet K. Sinha, Fatemah Behbehani, and Boris Jeremić. Modelling of buoyant forces in earthquake soil-structure interaction. In *Proceedings of the 15th International Conference of the International Association for Computer Methods and Advances in Geomechanics (IACMAG 15)*, Wuhan, China, October 19-23 2017.

[SBL84] Nakamura Sachio, Robert Benedict, and Roderic Lakes. Finite element method for orthotropic micropolar elasticity. *International Journal of Engineering Science*, 22(3):319–330, 1984.

[SBL06] Jean-François Sigrist, Daniel Broc, and Christian Lainé. Dynamic analysis of a nuclear reactor with fluid–structure interaction: Part i: Seismic loading, fluid added mass and added stiffness effects. *Nuclear engineering and design*, 236(23):2431–2443, 2006.

[SBPN10] Barna A. Szabó, Ivo Babuška, Juhani Pitkäranta, and Sebastian Nervi. The problem of verification with reference to the Girkmann problem. *Engineering with Computers*, 26:171–183, 2010.

[SBR$^+$23] Travis A. Shoemaker, Charbel Beaino, Dylan M. Centella R., Wendi Zhao, Carine Tanissa, Jack Lawrence, and Youssef M. A. Hashash. Generative AI: The new geotechnical assistant? *Journal of Geotechnical and Geoenvironmental Engineering*, 149(10):02823004, 2023.

[SBS$^+$95] Thomas Sterling, Donald J. Becker, Daniel Savarese, John E. Dorband, Udaya A. Ranawake, and Charles V. Parker. BEOWULF: A parallel workstation for scientific computations. In *Proceedings of the International Conference on Parallel on Parallel Processing*, 1995. `http://cesdis.gsfc.nasa.gov/beowulf/papers.html`

[SBTB21] A. Spagnoli, G. Beccarelli, M. Terzano, and J.R. Barber. A numerical study on frictional shakedown in large-scale three-dimensional conforming elastic contacts. *International Journal of Solids and Structures*, 217-218:1–14, 2021.

[SC93] E. Siebrits and S. L. Crouch. Geotechnical applications of a two-dimensional elastodynamic displacement discontinuity method. *International Journal for Rock Mechanics and Mining Sciences*, 30(7):1387–1393, 1993.

[SC94] E. Siebrits and S. L. Crouch. Two-dimensional elastodynamic displacement discontinuity method. *International Journal for Numerical Methods in Engineering*, 37:3229–3250, 1994.

[SC95] K. J. Shou and S. L. Crouch. A higher order displacement discontinuity method for analysis of crack problems. *International Journal for Rock Mechanics and Mining Sciences*, 32(1):49–55, 1995.

[SC98a] Jayant N. Sheth and Ronald L. Coker. Innovative use of slurry walls at dam number 2 hydropower project. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 124(6):518–522, June 1998.

[SC98b] MD Symans and MC Constantinou. Passive fluid viscous damping systems for seismic energy dissipation. *ISET Journal of Earthquake Technology*, 35(4):185–206, 1998.

[SC04] J. M. Segura and I. Carol. On zero-thickness interface elements for diffusion problems. *International Journal for Numerical and Analytical Methods in Geomechanics*, 28(9):947–962, 2004.

[SC08a] J.M. Segura and I. Carol. Coupled HM analysis using zero-thickness interface elements with double nodes. part i: Theoretical model. *International Journal for Numerical and Analytical Methods in Geomechanics*, 32:2083–2101, 2008.

[SC08b] J.M. Segura and I. Carol. Coupled HM analysis using zero-thickness interface elements with double nodes. part ii: Verification and application. *International Journal for Numerical and Analytical Methods in Geomechanics*, 32:2103–2123, 2008.

[SC10] J.M. Segura and I. Carol. Numerical modelling of pressurized fracture evolution in concrete using zero-thickness interface elements. *Engineering Fracture Mechanics*, 77:1386–1399, 2010.

[SCB$^+$98a] Thomas Sterling, Tom Cwik, Don Becker, John Salmon, Mike Warren, and Bill Nitzberg. An assessment of Beowulf-class computing for NASA requirements: Initial findings from the first NASA workshop on beowulf-class clustered computing. In *Proceedings, IEEE Aerospace*, 1998. http://www.beowulf.org/papers/index.html.

[SCB^{+}98b] Stein Sture, Nicholas Costes, Susan Batiste, Mark Lankton, Khalid AL-Shibli, Boris Jeremić, Roy Swanson, and Melissa Frank. Mechanics of granular materials at low effective stresses. *ASCE Journal of Aerospace Engineering*, 11(3):67–72, July 1998.

[SCB^{+}02] Jonathan P Stewart, Shyh-Jeng Chiou, Jonathan D Bray, Robert W Graves, Paul G Somerville, and Norman A Abrahamson. Ground motion evaluation procedures for performance-based design. *Soil dynamics and earthquake engineering*, 22(9-12):765–772, 2002.

[SCB06] S.T. Song, Y.H. Chai, and A.M. Budek. Methodology for preliminary seismic design of extended pile–shafts for bridge structures. *Earthquake Engineering & Structural Dynamics*, 35:1721–1738, 2006.

[SCG^{+}79] Stewart Schlesinger, Roy E. Crosbie, Roland E. Gagné, George S. Innis, C.S. Lalwani, Joseph Loch, Richard J. Sylvester, Richard D. Wright, Naim Kheir, and Dale Bartos. Terminology for model credibility. *Simulation*, 32(3):103–104, 1979.

[Sch67] H. Schaefer. Das Cosserat Kontinuum. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift f?r Angewandte Mathematik und Mechanik*, 47(8):485–498, 1 1967.

[Sch88] Ronald R. Schmeck. *Learning strategies and learning styles*, chapter 2 and 7. Plenum Press, 1988. LB1060 L4246 1988.

[Sch91] Herbert Schildt. *C++, the Complete Reference*. Osborne McGraw – Hill, 1991.

[Sch92] S.-P. Scholz. Elements of an object – oriented FEM++ program in C++. *Computers and Structures*, 43(3):517–529, 1992.

[Sch95] H. L. Schreyer. Continuum damage based on elastic projection operators. In Stein Sture, editor, *Proceedings of 10th Conference*, pages 6980–701. Engineering Mechanics Division of the American Society of Civil Engineers, May 1995.

[Sch98a] Andrew N. Schofield. Coulomb's 1773 essay. Technical Report 305, Cambridge University Engineering Department, Division of Soil Mechanics, 1998. `http://www2.eng.cam.ac.uk/~ans/habibx.html`.

[Sch98b] Andrew N. Schofield. Geotechnical centrifuge development can correct soil mechanics erros. In *Proceedings of Centrifuge 98*, volume II. Tokyo Conference of TC2, September 1998.

[Sch01] G. I. Schüeller. Computational stochastic mechanics - recent advances. *Computers and Structures*, 79(22-25):2225 – 2234, 2001.

[Sch07a] L.E. Schwer. An overview of the PTC 60/V&V 10: guide for verification and validation in computational solid mechanics. *Engineering with Computers.*, 23(4):245–252, 2007.

[Sch07b] Leonard E. Schwer. Validation metrics for response histories: perspectives and case studies. *Engineering with Computers.*, 23(4):295–309, 2007.

[Sch09] Martin Schanz. Poroelastodynamics: Linear models, analytical solutions, and numerical methods. *ASME Applied Mechanics Review*, 62:030803–1 – 030803–15, 2009.

[SCH^{+}12] Jonathan P. Stewart, C. B. Crouse, Tara Hutchinson, Bret Lizundia, Farzad Naeim, and Farhang Ostadan. Soil-structure interaction for building structures. Technical Report NIST GCR 12-917-21, National Institute of Standards and Technology, NIST, September 2012.

[Sch17] W.M. Scherzinger. A return mapping algorithm for isotropic and anisotropic plasticity models using a line search method. *Computer Methods in Applied Mechanics and Engineering*, 317:526 – 553, 2017.

[Sch18a] R.A. Schapery. Elastomeric bearing sizing analysis part 1: Spherical bearing. *International Journal of Solids and Structures*, 152-153:118 – 139, 2018.

[Sch18b] R.A. Schapery. Elastomeric bearing sizing analysis part 2: Flat and cylindrical bearings. *International Journal of Solids and Structures*, 152-153:140 – 150, 2018.

[SCM04] P. D. Spanos, P. Cacciola, and G. Muscolino. Stochastic averaging of preisach hysteretic systems. *ASCE Journal of Engineering Mechanics*, 130(11):1257–1267, November 2004.

[Sco63] Ronald F. Scott. *Principles of Soil Mechanics*. Addison - Wesley Publishing Company Inc., 1963.

[Sco88] Ronald F. Scott. Constitutive relations for soils: Present and future. In A. Saada and G. Bianchini, editors, *Constitutive Equations for Granular Non–Cohesive Soils*, pages 723–725. A. A. Balkema, July 1988.

[Sco96] R. F. Scott. Geotechnical earthquake engineering: Present dilemma and future directions. Presented at the Workshop: Application of Numerical Procedures in Geotechnical Earthquake Engineering, Universtity of California, Davis, October 28-30 1996.

[Sco06] Jennifer A. Scott. A frontal solver for the 21st century. *Communications in Numerical Methods in Engineering*, 22(10):1015–1029, October 2006.

[SCS06] Frank Scherbaum, Fabrice Cotton, and Helmut Staedtke. The estimation of minimum-misfit stochastic models from empirical ground-motion prediction equations. *Bulletin of the Seismological Society of America*, 96(2):427–445, April 2006.

[SCW^{+}08] MD Symans, FA Charney, AS Whittaker, MC Constantinou, CA Kircher, MW Johnson, and RJ McNamara. Energy dissipation systems for seismic applications: current practice and recent developments. *Journal of structural engineering*, 134(1):3–21, 2008.

[SCY97] W. Y. Shen, Y. K. Chow, and K. Y. Yong. A variational approach for vertical deformation analysis of pile group. *International Journal for Numerical and Analytical Methods in Geomechanics*, 21:741–752, 1997.

[SD40] S.Timoshenko and D.H.Young. *Engineering Mechanics*. McGraw-Hill Book Company, London, 1940.

[SD20] Jeongeun Son and Yuncheng Du. Comparison of intrusive and nonintrusive polynomial chaos expansion-based approaches for high dimensional parametric uncertainty quantification and propagation. *Computers & Chemical Engineering*, 134:106685, 2020.

[SdB93] J.C. J. Schellekens and R. de Borst. Interface elements. *International Journal for Numerical Methods in Engineering*, 36:43–66, 1993.

[SdC07] Giulio Sciarra, Francesco dell'Isola, and Olivier Coussy. Second gradient poromechanics. *International Journal of Solids and Structures*, 44(20):6607–6629, 2007.

[SDD20] Nishant Sharma, Kaustubh Dasgupta, and Arindam Dey. Optimum lateral extent of soil domain for dynamic ssi analysis of rc framed buildings on pile foundations. *Frontiers of Structural and Civil Engineering*, 14(1):62–81, 2020.

[SDG^{+}98] Walter J Silva, Robert Darragh, Nicholas Gregor, Geoff Martin, Norm Abrahamson, and Charles Kircher. Reassessment of site coefficients and near-fault factors for building code provisions. Technical report, Technical Report Program Element II: 98-HQGR-1010, Pacific Engineering and Analysis, El Cerrito, U.S.A., 1998.

[SdlPS^{+}09] M. Stupazzini, J. de la Puente, C. Smerzini, M. Kaser, H. Igel, and A. Castellani. Study of rotational ground motion in the near-field region. *Bulletin of the Seismological Society of America*, 99(2B):1271, 2009.

[SdOG04] Fernando Schnaid, Luis Artur Kratz de Oliveira, and Wai Ying Yuk Gehling. Unsaturated constitutive surfaces from pressuremeter tests. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 130(2):174–185, 2004.

[SE90] Robert B. Schnabel and Elizabeth Eskow. A new modified Cholesky factorization. *SIAM Journal On Scientific And Statistical Computing*, 11(6):1136–1158, 1990.

[Sem97] J. F. Semblat. Rheological interpretation of rayleigh damping. *Journal of Sound and Vibration*, 206(5):741–744, 1997.

[Sen] SensAble Technologies, Inc. PHANTOM haptic device. `http://www.sensable.com`

[Sew87] Michael J. Sewell. *Maximum and Minimum Principles*. Cambridge Texts in Applied Mathematics. Cambridge University Press, i edition, 1987.

[Sf91a] Juan. C. Simo and Robert L. fTaylor. Quasi–incompressible finite elasticity in principal stretches. continuum basis and numerical algorithms. *Computer Methods in Applied Mechanics and Engineering*, 85:273–310, 1991.

[SF91b] James M. Stallings and Karl H. Frank. Stay–cable fatigue behavior. *ASCE Journal of Structural Engineering*, 117(3):936–950, December 1991.

[SF96] William C. Shallenberger and George M. Filz. Interface strength determination using a large displacement shear box. In Masashi kamon, editor, *Environmental Geotechnics*, pages 147–152. Balkema, 1996.

[SF06] M.H. Scott and G.L. Fenves. Plastic hinge integration methods for force–based beam–column elements. *ASCE Journal of Structural Engineering*, 132:244–252, 2006.

[SF19] C. Soize and C. Farhat. Probabilistic learning for modeling and quantifying model-form uncertainties in nonlinear computational mechanics. *International Journal for Numerical Methods in Engineering*, 117(7):819–843, 2019.

[SFAO08] N.A. Sakharova, J.V. Fernandes, J.M. Antunes, and M.C. Oliveira. Comparison between berkovich, vickers and conical indentation tests: A three-dimensional numerical simulation study. *International Journal of Solids and Structures*, 46(5):1095–1104, 2008.

[SFB00] Gerik Scheuermann, Jan Frey, and Tom Bobach. Field analysis using topology methods (fantom) visualization program. `http://daddi.informatik.uni-kl.de`, 2000.

[SFC96] Kevin G. Sutterer, J. David Frost, and Jean–Lou A. Chameau. Polymer impregnation to assist undisturbed sampling of cohesionless soils. *ASCE Journal of Geotechnical Engineering*, 122(3):209–215, march 1996.

[SFD10] Mattias Schevenels, Stijn François, and Geert Degrande. *EDT, Elastodynamics Toolbox for Matlab*. Faculty of Engineering, Department of Civil Engineering, Structural Mechanics, Katholieke Universiteit, Leuven, Kasteelpark Arenberg 40 B-3001 Leuven, edt version 2.2 build 20 edition, July 2010.

[SFFF04] Michael H. Scott, Paolo Franchin, Gregory L. Fenves, and Filip C. Filippou. Response sensitivity for nonlinear beam-column elements. *ASCE Journal of Structural Engineering*, 130(9):1281–1288, 2004.

[SFH^{+}00] Gerik Scheuermann, Jan Frey, Bernd Hamann, Zhaohui Yang, and Boris Jeremić. Tensor visualizations for computational geomechanics. *Computers and Graphics*, 2000. In preparation.

[SFH^{+}01] Gerik Scheuermann, Jan Frey, Hans Hagen, Bernd Hamann, **B. Jeremić**, and Kenneth I. Joy. Case study: Visualization of seismic soils structure interaction simulations. In *Proceedings of the Joint Eurographics - IEEE TCVG Symposium on Visualization 2001*, page 10 pages, Ascona, Switzerland,, May 28-30 2001.

[SFLN00] Masanobu Shinozuka, Maria Q Feng, Jongheon Lee, and Toshihiko Naganuma. Statistical analysis of fragility curves. *Journal of engineering mechanics*, 126(12):1224–1231, 2000.

[SFMF08] Michael H. Scott, Gregory L. Fenves, Frank McKenna, and Filip C. Filippou. Software patterns for nonlinear beam-column models. *ASCE JOURNAL OF STRUCTURAL ENGINEERING*, 134(4):562–571, April 2008.

[SFS99a] Jonathan P. Stewart, Gregory L. Fenves, and Raymond B. Seed. Seismic soil–structure interaction in buildings. I: Analytical methods. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 125(1):26–37, 1999.

[SFS99b] Jonathan P. Stewart, Gregory L. Fenves, and Raymond B. Seed. Seismic soil–structure interaction in buildings. II: Empirical findings. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 125(1):38–48, 1999.

[SFS99c] Jonathan P Stewart, Gregory L Fenves, and Raymond B Seed. Seismic soil-structure interaction in buildings. i: Analytical methods. *Journal of Geotechnical and Geoenvironmental Engineering*, 125(1):26–37, 1999.

[SFT96a] E. Spacone, F. C. Filippou, and F. F. Taucer. Fibre beam-column model for non-linear analysis of r/c frames: Part i. formulation. *Earthquake Engineering & Structural Dynamics*, 25(7):711–725, July 1996.

[SFT96b] E. Spacone, F. C. Filippou, and F. F. Taucer. Fibre beam-column model for non-linear analysis of r/c frames: Part ii. applications. *Earthquake Engineering & Structural Dynamics*, 25(7):727–742, July 1996.

[SFT96c] E. Spacone, F.C. Filippou, and F.F. Taucer. Fibre beam–column model for non–linear analysis of r/c frames 1. formulation. *Earthquake Engineering and Structural Dynamics*, 25:711–725, 1996.

[SFWR00] J.E. Stice, R.M. Felder, D.R. Woods, and A. Rugarcia. The future of engineering education. iv. learning how to teach. *Chemical Engineering Education*, 34(2):118–127, 2000.

[SFY^{+}17] Sumeet K. Sinha, Yuan Feng, Han Yang, Hexiang Wang, Nebojša Orbović, David B. McCallen, and Boris Jeremić. 3-D non-linear modeling and its effects in earthquake soil-structure interaction. In *Proceedings of the 24th International Conference on Structural Mechanics in Reactor Technology (SMiRT 24)*, Busan, South Korea, August 20-25 2017.

[SG] D. Thomas Seidl and Brian N. Granzow. Calibration of elastoplastic constitutive model parameters from full-field data with automatic differentiation-based sensitivities. *International Journal for Numerical Methods in Engineering*, n/a(n/a).

[SG88] J. C. Simo and S. Govindjee. Exact closed - form solution of the return mapping algorith in plane stress elasto - viscoplasticity. *Engineering Computations*, 5:254–258, 1988.

[SG94] Jonathan Richard Shewchuk and Omar Ghattas. A compiler for parallel finite element methods with domain-decomposed unstructured meshes. *Contemporary Mathematics*, 180:445–445, 1994.

[SG99] M. Shirkhande and V.K. Gupta. Dynamic soil structure interaction effects on the seismic response of suspension bridges. *Earthquake Engineering & Structural Dynamics*, 28:1383–1403, 1999.

[SG02] S. Sakamoto and R. Ghanem. Polynomial chaos decomposition for the simulation of non-gaussian non-stationary stochastic processes. *Journal of Engineering Mechanics*, 128(2):190–201, February 2002.

[SG22a] Daniel P. Shahraki and Bojan B. Guzina. From d'alembert to bloch and back: A semi-analytical solution of 1d boundary value problems governed by the wave equation in periodic media. *International Journal of Solids and Structures*, 234-235:111239, 2022.

[SG22b] Christian Soize and Roger Ghanem. Probabilistic learning on manifolds (PLoM) with partition. *International Journal for Numerical Methods in Engineering*, 123(1):268–290, 2022.

[SG24] Christian Soize and Roger Ghanem. Probabilistic-learning-based stochastic surrogate model from small incomplete datasets for nonlinear dynamical systems. *Computer Methods in Applied Mechanics and Engineering*, 418:116498, 2024.

[SGD20] Qiangqiang Sun, Xiangfeng Guo, and Daniel Dias. Evaluation of the seismic site response in randomized velocity profiles using a statistical model with monte carlo simulations. *Computers and Geotechnics*, 120:103442, 2020.

[SGFS08] Daichao Sheng, Antonio Gens, Delwyn G Fredlund, and Scott W Sloan. Unsaturated soils: from constitutive modelling to numerical algorithms. *Computers and Geotechnics*, 35(6):810–824, 2008.

[SGJ71] F.H. Shipman, V.G. Gregson, and A.H. Jones. A shock wave study of coconino sandstone. Technical Report NASA CR-1842, NASA, May 1971.

[SGL12] Seth Stein, Robert J Geller, and Mian Liu. Why earthquake hazard maps often fail and what to do about it. *Tectonophysics*, 562:1–25, 2012.

[SGR09] A Spada, G Giambanco, and P Rizzo. Damage and plasticity at the interfaces in composite materials and structures. *Computer Methods in Applied Mechanics and Engineering*, 198(49):3884–3901, 2009.

[SH86] J. C. Simo and T. J. R. Hughes. On the variational foundations of assumed strain methods. *Journal of Applied Mechanics*, 53:51–54, March 1986.

[SH90a] J.C. Simo and T. Honein. Variational formulation, discrete conservation laws and path - domain independent integrals for elasto - viscoplasticity. *Journal of Applied Mechanics*, 57:488–497, 1990.

[SH90b] J.Q. Sun and C.S. Hsu. The generalized cell mapping method in nonlinear random vibration based upon short-time gaussian approximation. *Journal of Applied Mechanics*, 57(4):1018 – 1025, 1990.

[SH06] M.W. Schraad and F.H. Harlow. A stochastic constitutive model for disordered cellular materials: Finite-strain uni-axial compression. *International Journal of Solids and Structures*, 43(11-12):3542–3568, June 2006.

[SH08a] M. H. Scott and O. M. Hamutçuoğlu. Numerically consistent regularization of force-based frame elements. *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, 76:1612–1631, 2008.

[SH08b] Michael H. Scott and Terje Haukaas. Software framework for parameter updating and finite-element response sensitivity analysis. *ASCE Journal of Computing in Civil Engineering*, 22(5):281–291, September/October 2008.

[Sha] S. P. Shah. RILEM draft recommendations, TC 89–FMT fracture mechanics of concrete – test methods. In: Materials and Structures, 23, pp 461-465.

[Sha37] Claude Elwood Shannon. A symbolic analysis of relay and switching circuits. Master's thesis, MIT, 10Aug1937 1937.

[Sha40] Claude Elwood Shannon. *An Algebra for Theoretical Genetics*. PhD thesis, MIT, 15April1940 1940.

[Sha50] Claude E. Shannon. Programming a computer for playing chess. *Philosophical Magazine*, 7, 41(314):1–18, 1950.

[SHAGB20] Victor E. Saouma, Mohammad Amin Hariri-Ardebili, and Lori Graham-Brady. Stochastic analysis of concrete dams with alkali aggregate reaction. *Cement and Concrete Research*, 132:106032, 2020.

[SHB$^+$13] Jerome M. Solberg, Quazi Hossain, James A. Blink, Steven R. Bohlen, George Mseis, Harris R. Greenberg, and Robert M. Ferencz. Development of a generalized methodology for soil-structure interaction analysis using nonlinear time-domain techniques. Technical Report LLNL-TR-635762, Lawrence Livermore National Laboratory, 2013.

[She94] Jonathan Richard Shewchuk. An introduction fo the conjugate gradient method without the agonizing pain. Technical Report Edition $1\frac{1}{4}$, School of Computer Sciences, Carnegie Mellon University, Pittsburgh, PA,15213, August 1994.

[Shi72] M. Shinozuka. Monte carlo solution of structural dynamics. *Computers and Structures*, 2:855–874, 1972.

[SHK11] G. Sobhaninejad, M. Hori, and T. Kabeyasawa. Enhancing integrated earthquake simulation with high performance computing. *Advances in Engineering Software*, 42(5):286 – 292, 2011. ¡ce:title¿PARENG 2009¡/ce:title¿.

[SHM16] Jerome M. Solberg, Quazi Hossain, and George Mseis. Nonlinear time-domain soil-structure interaction analysis of embedded reactor structures subjected to earthquake loads. *Nuclear Engineering and Design*, 304:100 – 124, 2016.

[SHN$^+$09] L. Scholtès, P.-Y. Hicher, F. Nicot, B. Chareyre, and F. Darve. On the capillary stress tensor in wet granular materials. *International Journal for Numerical and Analytical Methods in Geomechanics*, 33(10):1289–1313, 2009.

[SHO$^+$15] T. A. Stern, S. A. Henrys, D. Okaya, J. N. Louie, M. K. Savage, S. Lamb, H. Sato, R. Sutherland, and T. and Iwasaki. A seismic reflection image for the base of a tectonic plate. *Nature*, 518:85–88, February 2015.

[SHU00] A. Yashima S. Hadush and R. Uzuoka. Importance of viscous fluid characteristics in liquefaction induced lateral spreading analysis. *Computers and Geotechnics*, 27(3):199–224, 2000.

[SI70] H. B. Seed and I. M. Idriss. Soil moduli and damping factors for dynamic response analyses. report eerc 70-10. Technical report, University of California Berkeley, 1970.

[SIG$^+$99] Paul Somerville, Kojiro Irikura, Robert Graves, Sumio Sawada, David Wald, Norman Abrahamson, Yoshinori Iwasaki, Takao Kagawa, Nancy Smith, and Akira Kowada. Characterizing crustal earthquake slip models for the prediction of strong ground motion. *Seismological Research Letters*, 70(1):59–80, 1999.

[Sil76] W Silva. Body waves in a layered anelastic solid. *Bulletin of the Seismological Society of America*, 66(5):1539–1554, 1976.

[Sil93] W. J. Silva. Factors controlling strong ground motions and their associated uncertainties. In *Seismic and Dynamic Analysis and Design Considerations for High Level Nuclear Waste Repositories*, pages 132–161. American Society of Civil Engineers, 1993.

[Sim85] J. C. Simo. On the computational significance of the intermediate configuration and hyperelastic stress relations in finite deformation elastoplasticity. *Mechanics of Materials*, 4:439–451, 1985.

[Sim88a] Juan C. Simo. A framework for finite strain elastoplasticity based on maximum plastic dissipation and the multiplicative decomposition: Part i. continuum formulation. *Computer Methods in Applied Mechanics and Engineering*, 66:199–219, 1988. TA345. C6425.

[Sim88b] Juan C. Simo. A framework for finite strain elastoplasticity based on maximum plastic dissipation and the multiplicative decomposition: Part ii. computational aspects. *Computer Methods in Applied Mechanics and Engineering*, 68:1–31, 1988. TA345. C6425.

[Sim92] J. C. Simo. Algorithms for static and dynamic multiplicative plasticity that preserve the classical return mapping schemes of the infinitesimal theory. *Computer Methods in Applied Mechanics and Engineering*, 99:61–112, 1992.

[Sin17] Sumeet Kumar Sinha. Modeling of dry and saturated soil-foundation interfaces. Master's thesis, University of California Davis, 2017.

[SJ87] J.C. Simo and J.W. Ju. Strain- and stress-based continuum damage models: Formulation. *International Journal of Solids and Structures*, 23(7):821 – 840, 1987.

[SJ07] Kallol Sett and Boris Jeremić. Uncertain soil properties and elastic–plastic simulations in geomechanics. In K. K. Phoon, G. A. Fenton, E. F. Glynn, C. H. Juang, T. F. Griffiths, T. F. Wolff, and L. Zhang, editors, *Probabilistic Applications in Geotechnical Engineering (Proceedings of Geo-Denver 2007: New Peaks in Geotechnics, Denver, Colorado, February 18-21, 2007)*, Geotechnical Special Publication No. 170, pages 1–11. American Society for Civil Engineers, 2007.

[SJ08] B.B. Soneji and R.S. Jangid. Influence of soil–structure interaction on the response of seismically isolated cable–stayed bridge. *Soil Dynamics and Earthquake Engineering*, 28:245–257, 2008.

[SJ10] Kallol Sett and Boris Jeremić. Probabilistic yielding and cyclic behavior of geomaterials. *International Journal for Numerical and Analytical Methods in Geomechanics*, 34(15):1541–1559, 2010. 10.1002/nag.870 (first published online March 11, 2010).

[SJ17] Sumeet Kumar Sinha and Boris Jeremić. Modeling of dry and saturated soil-foundation contact. Technical Report UCD–CompGeoMech–01–2017, University of California, Davis, August 2017.

[SJK96] Niclas Strömberg, Lars Johansson, and Anders Klarbring. Derivation and analysis of a generalized standard model for contact, friction and wear. *International Journal of Solids and Structures*, 33(13):1817–1836, 1996.

[SJK07a] Kallol Sett, Boris Jeremić, and M. Levent Kavvas. Probabilistic elasto-plasticity: Solution and verification in 1D. *Acta Geotechnica*, 2(3):211–220, October 2007.

[SJK07b] Kallol Sett, Boris Jeremić, and M. Levent Kavvas. The role of nonlinear hardening in probabilistic elasto-plasticity. *International Journal for Numerical and Analytical Methods in Geomechanics*, 31(7):953–975, June 2007.

[SJK11] Kallol Sett, Boris Jeremić, and M. Levent Kavvas. Stochastic elastic-plastic finite elements. *Computer Methods in Applied Mechanics and Engineering*, 200(9-12):997–1007, February 2011.

[SJSB97] Lewis H. Shapiro, Jerome B. Johnson, Matthew Sturm, and George L. Blaisdell. *Snow Mechanics, Review of the State of Knowledge and Applications*. Number 97-3. Cold Regions Research and Engineering Laboratory, 1997.

[SJW95] Dan R. Stoppenhagen, James O. Jirsa, and Loring A. Wyllie. Seismic repair and strengthening of a severly damaged concrete frame. *ACI Journal*, 92(2):177– 187, March/April 1995.

[SK35a] Katsutada Sezawa and Kiyoshi Kanai. Decay in the seismic vibrations of a simple or tall structure by dissipation of their energy into the ground. *Bulletin of the Earthquake Research Institute, Japan*, 13:682–697, 1935.

[SK35b] Katsutada Sezawa and Kiyoshi Kanai. Energy dissipation in seismic vibration of actual buildings. *Bulletin of the Earthquake Research Institute, Japan*, 13:925–941, 1935.

[SK35c] Katsutada Sezawa and Kiyoshi Kanai. Energy dissipation in seismic vibrations of a framed structure. *Bulletin of the Earthquake Research Institute, Japan*, 13:698–714, 1935.

[SK86] Gilbert Strang and Robert V. Kohn. Optimal design in elasticity and plasticity. *International Journal for Numerical Methods in Engineering*, 22:183–188, 1986.

[SK98] Jonathan P. Stewart and Seunghyuan Kim. Empirical verification of soil–structure interaction provisions in building codes. In Panos Dakoulas, Mishac Yegian, and Robert D. Holtz, editors, *Proceedings of a Specialty Conference: Geotechnical Earthquake Engineering and Soil Dynamics III*, Geotechnical Special Publication No. 75, pages 1259–1270. ASCE, August 1998. 1998.

[SK00a] Kambiz Salari and Patrick Knupp. Code verification by the method of manufactured solutions. Technical report, Sandia National Labs., Albuquerque, NM (US); Sandia National Labs., Livermore, CA (US), 2000.

[SK00b] Bruno Sudret and Armen Der Kiureghian. Stochastic finite element methods and reliability: A state–of–the–art report. Technical Report UCB/SEMM-2000/08, University of California at Berkeley, November 2000.

[SK05] S. Sharma and M. L. Kavvas. Modeling noncohesive suspended sediment transport in stream channels using an ensemble-averaged conservation equation. *ASCE JOURNAL OF HYDRAULIC ENGINEERING*, 131(5):380–389, May 2005.

[SK12] László Szabó and Attila Kossa. A new exact integration method for the drucker-prager elastoplastic model with linear isotropic hardening. *International Journal of Solids and Structures*, 49(1):170 – 190, 2012.

[SKAH12a] Armin W. Stuedlein, Steven L. Kramer, Pedro Arduino, and Robert D. Holtz. Geotechnical characterization and random field modeling of desiccated clay. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 138(11):1301–1313, November 2012.

[SKAH12b] Armin W. Stuedlein, Steven L. Kramer, Pedro Arduino, and Robert D. Holtz. Reliability of spread footing performance in desiccated clay. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 138(11):1314–1325, November 2012.

[SKB08] Jean-François Semblat, Marc Kham, and Pierre-Yves Bard. Seismic-wave propagation in alluvial basins and influence of site-city interaction. *Bulletin of the Seismological Society of America*, 98(6):2665–2678, December 2008.

[SKBV21] Andrea Spillatura, Mohsen Kohrangi, Paolo Bazzurro, and Dimitrios Vamvatsikos. Conditional spectrum record selection faithful to causative earthquake parameter distributions. *Earthquake Engineering & Structural Dynamics*, 2021.

[Ske07] R.P. Skelton. Editorial for "A mathematical representation of the multiaxial Bauschinger effect" by Armstrong and Frederick. *Materials at High Temperature*, 24(1):1–26, 2007.

[SKE08] Oh Sung Kwon and Amr S. Elnashai. Seismic analysis of meloland road overcrossing using multiplatform simulation software including ssi. *ASCE Journal of Structural Engineering*, 134(4):651–660, 2008.

[SKG88] J. C. Simo, J. G. Kennedy, and S. Govindjee. Non - smooth multisurface plasticity and viscoplasticity. loading / unloading, conditions and numerical algorithms. *International Journal for Numerical Methods in Engineering*, 26:2161–2185, 1988.

[SKG19] Mohammad R Salami, Mohammad M Kashani, and Katsuichiro Goda. Influence of advanced structural modeling technique, mainshock-aftershock sequences, and ground-motion types on seismic fragility of low-rise rc structures. *Soil Dynamics and Earthquake Engineering*, 117:263–279, 2019.

[SKH$^+$14] Christian Schmitt, Sebastian Kuckuk, Frank Hannig, Harald Köstler, and Jürgen Teich. Exaslang: A domain-specific language for highly scalable multigrid solvers. In *Proceedings of the Fourth International Workshop on Domain-Specific Languages and High-Level Frameworks for High Performance Computing*, WOLFHPC '14, pages 42–51, Piscataway, NJ, USA, 2014. IEEE Press.

[SKK79] H. L. Schreyer, R. F. Kulak, and J. M. Kramer. Accurate numerical solutions for elastic - plastic models. *Journal of Pressure Vessel Technology*, 101:226–234, 1979.

[SKK97a] Kirk Schloegel, George Karypis, and Vipin Kumar. Multilevel diffusion schemes for repartitioning of adaptive meshes. Technical Report 97-013, Univesity of Minnesota, June 1997.

[SKK97b] Kirk Schloegel, George Karypis, and Vipin Kumar. Parallel multilevel diffusion schemes for repartitioning of adaptive meshes. Technical Report 97-014, Univesity of Minnesota, 1997.

[SKK98a] Kirk Schloegel, George Karypis, and Vipin Kumar. Wavefront diffucion and LMSR: Algorithms for dynamic repartitioning of adaptive meshes. Technical Report 98-034, Univesity of Minnesota, 1998.

[SKK$^+$98b] Kirk Schloegel, George Karypis, Vipin Kumar, Rupak Biswas, and Leonid Oliker. A performance study of diffusive vs. remapped load–balancing schemes. Technical Report 98-018, Univesity of Minnesota and Army HPC Research Center and NASA Ames Research Center, 1998.

[SKMR98] Gerik Scheuermann, Heinz Krüger, Martin Menzel, and Alyn P. Rockwood. Visualizing nonlinear vector field topology. *IEEE Transactions on Visualization and Computer Graphics*, 4(2):109–116, 1998.

[SKS$^+$00] Takahiro Sigaki, Kazuhiko Kiyohara, Yoichi Sono, Dai Kinosita, Toru Masao, Ryoichi Tamura, Chiaki Yoshimura, and Takeshi Ugata. Estimation of earthquake motion incident angle at rock site. In *Proceedings of 12th world conference on earthquake engineering*, pages 1–8, 2000.

[SKSM10] I Shahrour, F Khoshnoudian, M Sadek, and H Mroueh. Elastoplastic analysis of the seismic response of tunnels in soft soils. *Tunnelling and underground space technology*, 25(4):478–482, 2010.

[SKVA20] Max Sieber, Sebastian Klar, Michalis F. Vassiliou, and Ioannis Anastasopoulos. Robustness of simplified analysis methods for rocking structures on compliant soil. *Earthquake Engineering & Structural Dynamics*, n/a(n/a), 2020.

[SL06] A Szymkiewicz and J Lewandowska. Unified macroscopic model for unsaturated water flow in soils of bimodal porosity. *Hydrological sciences journal*, 51(6):1106–1124, 2006.

[SLA78] Genevieve Segol, Peter C.Y. Lee, and John F. Abel. Amplitude reduction of surface waves by trenches. *Journal of the Engineering Mechanics Division*, 104(3):621–641, 1978.

[Sla02] William S Slaughter. Linearized elasticity problems. In *The Linearized Theory of Elasticity*, pages 221–254. Springer, 2002.

[SLC97] Hung Tao Shen, Shunan Lu, and Randy D. Crissman. Numerical simulation of ice transport over the lake Erie–Niagara river ice boom. *Cold Regions Science and Technology*, 26:17–33, 1997.

[SLDLW99] Xiang Song Li, Yannis F. Dafalias, and Zhi Liang Wang. State dependent dilatancy in critical state constitutive modelling of sand. *Canadian Geotechnical Journal*, In Print, 1999.

[SLEA17] Lei Su, Jinchi Lu, Ahmed Elgamal, and Arul K. Arulmoli. Seismic performance of a pile-supported wharf: Three-dimensional finite element simulation. *Soil Dynamics and Earthquake Engineering*, 95:167 – 179, 2017.

[SLG10] Jean-François Semblat, Luca Lenti, and Ali Gandomzadeh. A simple multi-directional absorbing layer method to simulate elastic wave propagation in unbounded domains. *International Journal for Numerical Methods in Engineering*, doi: 10.1002/nme.3035, 2010.

[SLM08] Francisco Silva, T. William Lambe, and W. Allen Marr. Probability and risk of slope failure. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 134(12):1691–1699, December 2008.

[SLR97] P. Steinman, R. Larsson, and K. Runesson. On the localized properties of multiplicative hyperelasto–plastic continuaa with strong discontinuities. *International Journal of Solids and Structures*, 34(8):969–990, 1997.

[Slu92] Lambertus Johannes Sluys. *Wave Propagation, Localization and Dispersion in Softening Solids*. PhD thesis, Civil Engineering Department of Delft University of Technology, June 1992.

[SM50] Jack Sherman and Winifried J. Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics*, XXI:124–127, 1950.

[SM57] K.V. Steinbrugge and D.F Moran. Engineering aspects of the Dixie Valley-Fairview Peak earthquakes. *Bulletin of the Seismological Society of America*, 47(4):335–348, October 1957.

[SM84a] J. C. Simo and J. E. Marsden. Stress tensors, riemannian metrics and the alternative descriptions in elasticity. In P.G. Ciarlet and M. Roseau, editors, *Trends and Applications of Pure Mathematics to Mechanics*, volume 195 of *Lecture Notes in Physics*, pages 369–383. Springer-Verlag, 1984.

[SM84b] J. C. Simo and J. E. Marsden. Stress tensors, riemannian metrics and the alternative descriptions in plasticity. In *Proc. Symp. Trends in Application of Mathematics to Mechanics, Paris 1983*, pages 367–383. Springer Verlag, 1984.

[SM92] J. C. Simo and C. Miehe. Associative coupled thermoplasticity at finite strains: Formulation, numerical analysis and implementation. *Computer Methods in Applied Mechanics and Engineering*, 98:41–104, 1992.

[SM93] J. C. Simo and G. Meschke. A new class of algorithms for classical plasticity extended to finite strain. application to geomaterials. *Computational Mechanics*, 11:253–278, 1993.

[SM99] V. V. R. N. Sastry and G. G. Meyerhof. Flexible piles in layered soil under eccentric and included loads. *Soils and Foundations*, 39(1):11–20. Feb. 1999.

[SM00] Daniel Sorid and Samuel K. Moore. The virtual surgeon. *IEEE Spectrum*, pages 26–31, July 2000.

[SM09] Andrzej Sawicki and Jacek Mierczyński. On the behaviour of liquefied soil. *Computers and Geotechnics*, 36(4):531–536, May 2009.

[SM10] Jonathan P. Stewart and George Mylonakis. Baseline knowledge report on soil-foundation-structure interaction of building structures. Technical Report ATC 83 Project. Task 10, Applied Technology Council, September 2010.

[SM19] P. Staubach and J. Machaček. Influence of relative acceleration in saturated sand: Analytical approach and simulation of vibratory pile driving tests. *Computers and Geotechnics*, 112:173 – 184, 2019.

[SM23] Bernd Schulze and Cameron Millar. Graphic statics and symmetry. *International Journal of Solids and Structures*, 283:112492, 2023.

[Sma01] J.C. Small. Practical solutions to soil–structure interaction problems. *Progress in Structural Engineering and Materials*, 3:305–314, 2001.

[SMD05] Nicholas Sitar, Mary M. MacLaughlin, and David M. Doolin. Influence of kinematics on landslide mobility and failure mode. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 131(6):716–728, June 2005.

[Sme05] D. M. J. Smeulders. Experimental evidence for slow compressional waves. *ASCE Journal of Engineering Mechanics*, 131(9):908–917, September 2005.

[Smi02] Malcolm C. Smith. Synthesis of mechanical networks: The inerter. *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*, 47(10):1648–1662, October 2002.

[Smi14] Ralph C. Smith. *Uncertainty Quantification*. Computational Science & Engineering. SIAM, 2014.

[Smi23] D. Smith. Physics II. Lecture notes. Embry-Riddle Aeronautical University, Prescott, `https://physicsx.erau.edu/`, June 2023.

[Smo63] S.A. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Soviet Math. Dokl.*, 4:240–243, 1963.

[SMR05] M. Sarrazin, o. Moroni, and J.M. Roesset. Evaluation of dynamic response characteristics of seismically isolated bridges in chile. *Earthquake Engineering & Structural Dynamics*, 34:425–448, 2005.

[SMS94] P. Steinmann, C. Miehe, and E. Stein. Comparison of different finite deformation inelastic damage models within multiplicative elastoplasticity for ductile materials. *Computational Mechanics*, 13(6):458–474, 1994.

[SMSW17] H Stutz, D Mašín, AS Sattari, and Frank Wuttke. A general approach to model interfaces using existing soil constitutive models application to hypoplasticity. *Computers and Geotechnics*, 87:115–127, 2017.

[SMW16] H Stutz, D Mašín, and Frank Wuttke. Enhancement of a hypoplastic model for granular soil–structure interface behaviour. *Acta Geotechnica*, 11(6):1249–1261, 2016.

[SN95] Haluk Sucuoğlu and Alphan Nurtuğ. Earthquake ground motion characteristics and seismic energy dissipation. *Earthquake Engineering & Structural Dynamics*, 24(9):1195–1213, 1995.

[SN96] H. L. Schreyer and M. K. Nielsen. Analytical and numerical tests for loss of material stability. *International Journal for Numerical Methods in Engineering*, 39:1721–1736, 1996.

[SN06] E Sharbati and R Naghdabadi. Computational aspects of the Cosserat finite element analysis of localization phenomena. *Computational materials science*, 38(2):303–315, 2006.

[SNCS14] H. Sabetamal, M. Nazem, J.P. Carter, and S.W. Sloan. Large deformation dynamic analysis of saturated porous media with applications to penetration problems. *Computers and Geotechnics*, 55(0):117 – 131, 2014.

[Sni98] Roel Snieder. *A Guided Tour of Mathematical Physics*. Samizdat Press `http://samizdat.mines.edu`, 1998.

[SNK06a] Sachin K. Sachdeva, Prasanth B. Nair, and Andy J. Keane. Comparative study of projection schemes for stochastic finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 195(19-22):2371–2392, April 2006.

[SNK06b] Sachin K. Sachdeva, Prasanth B. Nair, and Andy J. Keane. Hybridization of stochastic reduced basis methods with polynomial chaos expansions. *Probabilistic Engineering Mechanics*, 21:182–192, 2006.

[SNK^{+}18] Arfon M. Smith, Kyle E. Niemeyer, Daniel S. Katz, Lorena A. Barba, George Githinji, Melissa Gymrek, Kathryn D. Huff, Christopher R. Madan, Abigail Cabunoc Mayes, Kevin M. Moerman, Pjotr Prins, Karthik Ram, Ariel Rokem, Tracy K. Teal, Roman Valls Guimera, and Jacob T. Vanderplas. Journal of open source software (JOSS): design and first-year review. *PeerJ Computer Science*, 4:e147, February 2018.

[SO85] J. C. Simo and M. Ortiz. A unified approach to finite deformation elastoplastic analysis based on the use of hyperelastic constitutive equations. *Computer Methods in Applied Mechanics and Engineering*, 49:221–245, 1985.

[SO90] T Strouboulis and JT Oden. A posteriori error estimation of finite element approximations in fluid mechanics. *Computer methods in applied mechanics and engineering*, 78(2):201–242, 1990.

[Sob91] K. Sobczyk. *Stochastic Differential Equations With Applications to Physics and Engineering*. Mathematics and its Applications. Springer Dordrecht, 1991.

[Sob01] I.M Sobol. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulation*, 55(1):271 – 280, 2001. The Second IMACS Seminar on Monte Carlo Methods.

[Soi94] C. Soize. *The Fokker-Planck Equation for stochastic dynamical systems and its explicit steady state solutions*. World Scientific, Singapore, 1994.

[Soi10] Christian Soize. Generalized probabilistic approach of uncertainties in computational dynamics using random matrices and polynomial chaos decompositions. *International Journal for Numerical Methods in Engineering*, 81(8):939–970, 2010.

[Sol14] Edgar Solomonik. Cyclops tensor framework, 2014.

[Som98] Paul Somerville. Emerging art: Earthquake ground motion. In Panos Dakoulas, Mishac Yegian, and Robert D. Holtz, editors, *Proceedings of a Specialty Conference: Geotechnical Earthquake Engineering and Soil Dynamics III*, Geotechnical Special Publication No. 75, pages 1–38. ASCE, August 1998. 1998.

[Sou23] Eduardo Souza. Is it possible to grow cement? Prometheus Materials and the transformation of concrete. online, on https://www.archdaily.com/1007630/, October 2023.

[SP83a] A. N. Stavrogin and E. D. Pevzner. Critical states for rocks at changing deformation rates. *Journal of Mining Science*, 19(5):359–366, 1983.

[SP83b] A. N. Stavrogin and A. G. Protosenya. Rock plasticity in conditions of variable deformation rates. *Journal of Mining Science*, 19(4):245–255, 1983.

[SP84] J. C. Simo and K. S. Pister. Remarks on rate constitutive equations for finite deformations problems: Computational implications. *Computer Methods in Appliied Mechanics and Engineering*, 46:201–215, 1984.

[SP94a] J. C. J. Schellekens and H. Parisch. On finite deformation elasticity. Technical Report ISD Nr. 94/3, Institute for Statics and Dynamics of Aerospace Structures, Pfaffenwaldring 27, 70550 Stutgart, 1994.

[SP94b] J. C. J. Schellekens and H. Parisch. On finite deformation elasto–plasticity. Technical Report ISD Nr. 94/5, Stutgart Institute for Statics and Dynamics of Aerospace Structures, Pfaffenwaldring 27, 70550 Stutgart, 1994.

[SP97] I. M. Smith and M. A. Pettipher. Finite elements and parallel computations in geomechanics. In Pietruszczak and Pande, editors, *Numerical Methods in Geomechanics*, pages 421–425. Balkema, Rotterdam, 1997.

[SP05] H.F. Schweiger and G.M. Peschl. Reliability analysis in geotechnics with the random set finite element method. *Computers and Geotechnics*, 32(6):422–435, September 2005.

[SP06] Victor Saouma and Luigi Perotti. Constitutive model for alkali-aggregate reactions. *Materials Journal*, 103(3):194–202, 2006.

[SP09] Jean-François Semblat and Alain Pecker. *Waves and Vibrations in Soils: Earthquakes, Traffic, Shocks, Construction works*. IUSS Press, first edition, 2009.

[SP11] Björn Sjögreen and N. Anders Petersson. A Fourth Order Accurate Finite Difference Scheme for the Elastic Wave Equation in Second Order Formulation. *J. Sci. Comput.*, 52(1):17–48, 2011.

[SP12] Navjeev Saxena and D.K. Paul. Effects of embedment including slip and separation on seismic ssi response of a nuclear reactor building. *Nuclear Engineering and Design*, 247(0):23 – 33, 2012.

[SP21] Takayuki Shuku and Kok-Kwang Phoon. Three-dimensional subsurface modeling using geotechnical lasso. *Computers and Geotechnics*, 133:104068, 2021.

[SPB19] M. Shahbazi, O. Penner, and B. Bergman. Seismic input and topographic effects: A rigorous approach to simulating 3D dam-foundation interaction in LS-DYNA. In *United States Society on Dams, Conference, Chicago*, 2019.

[SPC^{+}13] M.B. Syed, L. Patisson, M. Curtido, B. Slee, and S. Diaz. The challenging requirements of the {ITER} anti seismic bearings. *Nuclear Engineering and Design*, 0:–, 2013.

[Spe80] A. J. M. Spencer. *Continuum Mechanics*. Longman Mathematical Texts. Longman Group Limited, 1980.

[Spi01] Diomidis Spinellis. Notable design patterns for domain-specific languages. *Journal of Systems and Software*, 56(1):91–99, February 2001. `http://dx.doi.org/10.1016/S0164-1212(00)00089-3`.

[SPK89] CC Spyrakos, PN Patel, and FT Kokkinos. Assessment of computational practices in dynamic soil-structure interaction. *Journal of computing in civil engineering*, 3(2):143–157, 1989.

[SPK11] Navjeev Saxena, D.K. Paul, and Ram Kumar. Effects of slip and separation on seismic ssi response of nuclear reactor building. *Nuclear Engineering and Design*, 241(1):12 – 17, 2011.

[SPM^{+}15] Szilárd Szalay, Max Pfeffer, Valentin Murg, Gergely Barcza, Frank Verstraete, Reinhold Schneider, and Örs Legeza. Tensor product methods and entanglement optimization for ab initio quantum chemistry. *International Journal of Quantum Chemistry*, 115(19):1342–1391, 2015.

[SPM24] Bartosz Sobczyk, Lukasz Pyrzowski, and Mikotaj Miśkiewicz. Computational modelling of historic masonry railroad arch bridges. *Computers & Structures*, 291:107214, 2024.

[SPP82] BD Scott, R Park, and MJN Priestley. Fiber element modeling for seismic performance of bridge columns made of concrete-filled frp tubes. *Journal of the American Concrete Institute*, 79(1):13–27, 1982.

[SPR18] Jonathan Salvi, Fabio Pioldi, and Egidio Rizzi. Optimum tuned mass dampers under seismic soil-structure interaction. *Soil Dynamics and Earthquake Engineering*, 114:576 – 597, 2018.

[SPTJ12] Hadi Shahir, Ali Pak, Mahdi Taiebat, and Boris Jeremić. Evaluation of variation of permeability in liquefiable soil under earthquake loading. *Computers and Geotechnics*, 40:74–88, 2012.

[SPU98] Qin Shen, Alex Pang, and Sam Uselton. Data level comparison of wind tunnel and computational fluid dynamics data. In *IEEE Visualization '98*, pages 415–418, 557, 1998.

[SR81] TD Sachdeva and CV Ramakrishnan. A finite element solution for the two-dimensional elastic contact problems with friction. *International Journal for Numerical Methods in Engineering*, 17(8):1257–1271, 1981.

[SR90] J. C. Simo and M. S. Rifai. A class of mixed assumed strain methods and the method of incompatible models. *International Journal for Numerical Methods in Engineering*, 29:1595–11638, 1990.

[SR92] M. J. Saran and K. Runesson. A generalized closest–point–projection method for deformation–neutralized formualtion in finite strain plasticity. *Engineering Computations*, 9:359–370, 1992.

[SR97] I Shahrour and F Rezaie. An elastoplastic constitutive relation for the soil-structure interface under cyclic loading. *Computers and Geotechnics*, 21(1):21–39, 1997.

[SR01] Natarajan Sukumar and Mark Rashid. Finite element procedures in applied mechanics, April 2001. ECI212A, Lecture Notes, UC Davis.

[SR02] Mettupalayam V. Sivaselvan and Andrei M. Reinhorn. Collapse analysis: Large inelastic deformations analysis of planar frames. *ASCE Journal of Structural Engineering*, 128(12):1575–1583, December 2002.

[SR08] E.H. Stehmeyer and D.C. Rizos. Considering dynamic soil structure interaction (ssi) effects on seismic isolation retrofit efficiency and the importance of natural frequency ratio. *Soil Dynamics and Earthquake Engineering*, 28:468–479, 2008.

[SRA⁺07] Andrea Saltelli, Marco Ratto, Terry Andres, Francesca Campolongo, Jessica Cariboni, Debora Gatelli, Michaela Saisana, and Stefano Tarantola. *Global Sensitivity Analysis. The Primer*. John Wiley and Sons, 2007.

[SRM22] Sharana Kumar Shivanand, Bojana Rosić, and Hermann G. Matthies. Stochastic modelling of symmetric positive definite material tensors. *arXiv*, 10.48550/arXiv.2109.07962, 2022.

[SRME⁺17] Peter J Stafford, Adrian Rodriguez-Marek, Benjamin Edwards, Pauline P Kruiver, and Julian J Bommer. Scenario dependence of linear site-effect factors for short-period response spectral ordinates scenario dependence of linear site-effect factors for short-period response spectral ordinates. *Bulletin of the Seismological Society of America*, 107(6):2859–2872, 2017.

[SRMP89] S. Sture, K. Runesson, and E. J. Macari-Pasqualino. Analysis and calibration of a three invariant plasticity model for granular materials. *Ingenieur Archiv*, 59:253–266, 1989.

[SS86] Youcef Saad and Martin H. Schultz. Gmres: A generalized minimal residual algorithm for solving non-symmetric linear systems. *SIAM Journal on Scientific Computating*, 7(3):856–869, 1986.

[SS88] D. A. Shuttle and I. M. Smith. Numerical simulation of shear band formation in soils. *International Journal for Numerical and Analytical Methods in Geomechanics*, 12:611–626, 1988.

[SS90] Michal J. Saran and Alf Samuelsson. Elastic - viscoplastic implicit formulation for finite element simulation of complex sheet forming processes. *International Journal for Numerical Methods in Engineering*, 30:1675–1697, 1990.

[SS95] Paul Steinman and Erwin Stein. Shearband localization and instability phenomena in crystal plasticity. In Stein Sture, editor, *Proceedings of 10th Conference*, pages 1022–1025. Engineering Mechanics Division of the American Society of Civil Engineers, May 1995.

[ŠS96] D. Šumarac and S. Stošić. The Preisach model for the cyclic bending of elasto–plastic beams. *European Journal of Mechanics A/Solids*, 15(1):155–172, 1996.

[SS97] Jonathan P. Stewart and Alisa F. Stewart. Analysis of soil-structure interaction effects on building response from earthquake strong motion recordings at 58 sites. Technical Report UCB/EERC-97/01, University of California at Berkeley, 1997.

[SS01] Bernhard A. Schrefler and Roberto Scotta. A fully coupled dynamic model for two-phase fluid flow in deformable porous media. *Computer Methods in Applied Mechanics and Engineering*, 190(24-25):3223–3246, 2001.

[SS02] TT Soong and BF Spencer. Supplemental energy dissipation: state-of-the-art and state-of-the-practice. *Engineering Structures*, 24(3):243–259, 2002.

[SS04] Machhour Sadek and Isam Shahrour. A three dimensional embedded beam element for reinforced geomaterials. *International Journal for Numerical and Analytical Methods in Geomechanics*, 28(9):931–946, 2004.

[SS05] V. A. Salomoni and B. A. Schrefler. A cbs-type stabilizing algorithm for the consolidation of saturated porous media. *International Journal for Numerical Methods in Engineering*, 63(4):502–527, 2005.

[SS14] Seth Stein and Jerome Stein. *Playing against nature: integrating science and economics to mitigate natural hazards in an uncertain world*. John Wiley & Sons, 2014.

[SS22] Hyoung Suk Suh and WaiChing Sun. Multi-phase-field microporomechanics model for simulating ice-lens growth in frozen soil. *International Journal for Numerical and Analytical Methods in Geomechanics*, 46(12):2307–2336, 2022.

[SSB85] Gerald A. Shultz, Robert B. Scnabel, and Richard H. Byrd. A familly of trust – region – based algorithms for unconstrained minimization with strong global convergence properties. *SIAM Journal On Numerical Analysis*, 22(1):47–67, 1985.

[SSBS99] Thomas L. Sterling, John Salmon, Donald J. Becker, and Daniel F. Savarese. *How to Build a Beowulf: A Guide to the Implementation and Application of PC Clusters*. Scientific and Engineering Computations Series. The MIT Press, 1999. ISBN 0-262-69218-X ; QA 76.58.S854 1998.

[SSH90] Ranbir S. Sandhu, H.L. Shaw, and S.J. Hong. A three-field finite element procedure for analysis of elastic wave propagation through fluid-saturated soils. *Soil Dynamics and Earthquake Engineering*, 9:58–65, 1990.

[SSJ87] Jörg Schlaich, Kurt Schäfer, and Mattias Jennewein. Toward a consistent design of structural concrete. *PCI Journal*, 32(3):77–150, May/June 1987.

[SSM95] E. Stein, P. Steinmann, and C. Miehe. Instability phenomena in plasticity: Modeling and computation. *Computational Mechanics*, 17:74–87, 1995.

[SSO19] Steve WaiChing Sun, George Spanos, and OTHERS. Verification & validation of computational models associated with the mechanics of materials. Technical report, TMS, The Minerals, Metals & Metarials Society, 2019.

[SSR82] Ignacio Sanchez-Salinero and Jose M. Roesset. Static and dynamic stiffness of single piles. Technical Report Geotechnical Engineering Report GR82-31, Geotechnical Engineering Center, Civil Engineering Department, The University of Texas at Austin, 1982.

[SSRT83] Ignacio Sanchez-Salinero, Jose M. Roesset, and John L. Tassoulas. Dynamic stiffness of pile groupes: Approximate solutions. Technical Report Geotechnical Engineering Report GR83-5, Geotechnical Engineering Center, Civil Engineering Department, The University of Texas at Austin, 1983.

[SSSM17] Benshun Shao, Andreas Schellenberg, Matthew Schoettler, and Stephen Mahin. Preliminary studies on the dynamic response of a seismically isolated prototype gen-iv sodium-cooled fast reactor (pgsfr). Technical Report 2017/11, Pacific Earthquake Research Center, PEER, 2017. see refined model in OpenSees, p53, p58fig4.4.

[SSW96] Y. S. Suh, F. I. Saunders, and R. H. Wagoner. Anisotropic yield functions with plastic–strain–induced anisotropy. *International Journal of Plasticity*, 12(2):417–438, 1996.

[SSWB04] Jonathan P. Stewart, Patrick M. Smith, Daniel H. Whang, and Jonathan D. Bray. Seismic compression of two compacted earth fills shaken by the 1994 Northridge earthquake. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 130(5):461–476, May 2004.

[SSX98] C. Schwab, M. Suri, and C. Xenophontos. The *hp* finite element methods for problems in mechanics with boundary layers. *Computer methods in applied mechanics and engineering*, 157:311–333, 1998.

[SSY99] D. Sheng, S. W. Sloan, and H. S. Yu. Aspects of finite element implementation of critical state models. Researh report 176.02.1999, The University of Newcastle, 1999. ISBN 0 7259 1070 4.

[ST81] A. S. Saada and F. C. Townsend. Strength laboratory testing of soils. *ASTM*, STP 740:7–77, June 25th 1981. A State of the Art paper presented at the ASTM Symposium on the Shear Strength of Soils, June 25, 1980,.

[ST85] J. C. Simo and R. L. Taylor. Consistent tangent operators for rate-independent elastoplasticity. *Computer Methods in Applied Mechanics and Engineering*, 48:101–118, 1985.

[ST86] J. C. Simo and R. L. Taylor. A returning mapping algorithm for plane stress elastoplasticity. *International Journal for Numerical Methods in Engineering*, 22:649–670, 1986.

[ST97] Eckart Schnack and Karsten Türke. Domain decomposition with BEM and FEM. *International Journal for Numerical Methods in Engineering*, 40:2593–2610, 1997.

[ST01] S.S.Rajashree and T.G.Sitharam. Nonlinear finite element modeling of batter piles under lateral load. *Journal of Geotechnical and Geoenvironmental Engineering*, 127(7):604–612, July 2001.

[Sta92] Bjørn Stavtrup. A proposal regarding invisible logic for object-oriented languages. *Journal of object oriented programming*, 5(1):63–65, March/April 1992.

[Sta17] Peter J Stafford. Interfrequency correlations among fourier spectral ordinates and implications for stochastic ground-motion simulationinterfrequency correlations among fourier spectral ordinates and implications. *Bulletin of the Seismological Society of America*, 107(6):2774–2791, 2017.

[STC99] Andrea Saltelli, Stefano Tarantola, and KP-S Chan. A quantitative model-independent method for global sensitivity analysis of model output. *Technometrics*, 41(1):39–56, 1999.

[STC+25] Osman Sivrikaya, Emel Türker, Evrim Cüre, Esin Ertürk Atmaca, Zekai Angin, Hasan Basri Baçağa, and Ahmet Can Altunişik. Impact of soil conditions and seismic codes on collapsed structures during the 2023 Kahramanmaraş earthquakes: An in-depth study of 400 reinforced concrete buildings. *Soil Dynamics and Earthquake Engineering*, 190:109119, 2025.

[Ste09] George Stefanou. The stochastic finite element method: Past, present and future. *Computer Methods in Applied Mechanics and Engineering*, 198:1031–1051, 2009.

[Ste13] John D. Stevenson. Summary of the historical development of seismic design of nuclear power plants in japan and the u.s. *Nuclear Engineering and Design*, 0:–, 2013.

[STF01] A.N. Stavrogin, B.G. Tarasov, and (edited by) Charles Fairhurst. *Experimental physics and rock mechanics*. A.A. Balkema, Lisse ; Exton PA, 2001.

[STH+95] T. Sakemi, M. Tanaka, Y. Higuchi, K. Kawasaki, and K. Nagura. Permeability of pore fluids in the centrifugal fields. In *10th Asian Regional Conference on Soil Mechanics and Foundation Engineering (10ARC)*, Beijing, China, August 29 - Sept 2 1995.

[Sto92] Saša Stosić. Primena prajzakvog (preisach) modela u elastoplastičnoj analizi nozača izloženih civkličnom opterećenju (application of preisach model for elastoplastic analysis of cyclicaly loaded structural elements. Master's thesis, Gradjevinski Fakultet Univerziteta u Beogradu (Faculty of Civil Engineering, University of Belgrade), 1992.

[Sto98] Bertil Storåkers. Contact mechanics of particles and powders. In B. F. Carroll, editor, *Proceedings of the Thirteenth U.S. National Congress of Applied Mechanics*, page WK5. University of Florida, June 1998. ISBN 0-9652609.

[Sto22] Saša Stošić. Private communications, 1984-2022.

[Str77] Gilbert Strang. A homework exercise in finite elements. *International Journal for Numerical Methods in Engineering*, 11:411–417, 1977.

[Str86] Bjarne Stroustrup. *The C++ Programming Language*. Series in Computer Sciences. Addison – Wesley, reprint 1987 edition, 1986.

[Str94] Bjarne Stroustrup. *The Design and Evolution of C++*. Addison–Wesley Publishing Company, 1994.

[Str05] Bjarne Stroustrup. A rationale for semantically enhanced library languages. In David Musser, editor, *Library-Centric Software Design LCSD'05*. Object-Oriented Programming, Systems, Languages and Applications, October 2005.

[Str12] Stress Test Peer Review Board. Stress test performed on european nuclear power plants. Technical report, European Nuclear Safety Regulators Group (ENSREG), 2012.

[Stu86] Stein Sture. Hollow cylinder apparatuses, directional shear cells, and induced anisotropy in soils. Technical Report 59076-1, Norwegian Geotechnical Institute, P.O.Box 40 Täsen, Oslo, Norway, May 1986.

[Stu93] Stein Sture. Engineering properties of soils. Lecture Notes at CU Boulder, January - May 1993.

[Stu16] Hans Henning Stutz. *Hypoplastic Models for Soil-Structure Interfaces-Modelling and Implementation*. PhD thesis, Christian-Albrechts Universität Kiel, 2016.

[STU20] Anurag Sahare, Yoshikazu Tanaka, and Kyohei Ueda. Numerical study on the effect of rotation radius of geotechnical centrifuge on the dynamic behavior of liquefiable sloping ground. *Soil Dynamics and Earthquake Engineering*, 138:106339, 2020.

[Sud08a] Bruno Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability engineering & system safety*, 93(7):964–979, 2008.

[Sud08b] Bruno Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety*, 93(7):964 – 979, 2008. Bayesian Networks in Dependability.

[Sud14] Bruno Sudret. *Risk and Reliability in Geotechnical Engineering*, chapter Polynomial chaos expansions and stochastic finite element methods, pages 265–300. CRC Press, 12 2014.

[SUL76] H. Bolton Seed, Celso Ugas, and John Lysmer. Site–dependent spectra for earthquake–resistant design. *Bullettin of Seismological Society of America*, 66(1):221–243, 1976.

[Sun94] Keming Sun. Laterally loaded piles in elastic media. *Journal of Geotechnical Engineering*, 120(8):1324–1344, 1994.

[SUOc$^+$11] Kallol Sett, Berna Unutmaz, Kemal Önder Çetin, Suzana Koprivica, and Boris Jeremić. Soil uncertainty and its influence on simulated G/G_{max} and damping behavior. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 137(3):218–226, 2011. 10.1061/(ASCE)GT.1943-5606.0000420 (July 29, 2010).

[SUP03] Qin Shen, Sam Uselton, and Alex Pang. Comparison of wind tunnel experiments and computational fluid dynamics simulations. *Journal of Visualization*, 6(1):31–19, 2003.

[Sur96] Manil Suri. Analytic and computational assesment of locking in the *hp* finite element method. *Computer methods in applied mechanics and engineering*, 133(3/4):347–, 1996.

[Suy32] Kyoji Suyehiro. Engineering seismology notes on American lectures. In *Proceedings of the American Society of Civil Engineers*, volume 58, 4, pages 2–110. American Society of Civil Engineers, May 1932.

[SV21] Kadir C. Sener and Amit H. Varma. Steel-plate composite walls with different types of out-of-plane shear reinforcement: Behavior, analysis, and design. *Journal of Structural Engineering*, 147(2):04020329, 2021.

[SVC97] RG Selby, FJ Vecchio, and MP Collins. The failure of an offshore platform. *Concrete International*, 19(8):28–35, 1997.

[Sve14] Aram Arutiunovich Sveshnikov. *Applied methods of the theory of random functions*, volume 89. Elsevier, 2014.

[SVPT23] C. Smerzini, M. Vanini, R. Paolucci, and P. Traversa. Regional physics-based simulation of ground motion within the Rhône Valley, France, during the Mw 4.9 2019 Le Teil earthquake. *Bulletin of Earthquake Engineering*, 21:1747–1774, 2023.

[SW68] Andrew Schofield and Peter Wroth. *Critical state soil mechanics*, volume 310. McGraw-Hill London, 1968.

[SW70] H. Bolton Seed and Robert T. Whitman. Design of retaining structures for dynamic loads. In *State–of–the–Art Papers Presented at 1970 Specialty Conference: Lateral Stress in the Ground and Design of Earth–Retaining Structures*, pages 103–147. American Society of Civil Engineers, 1970.

[SW86] K. H. Schweizerhof and P. Wriggers. Consistent linearization for path following methods in finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 59:261–279, 1986.

[SW91] Juan Carlos Simo and Kachung Kevin Wong. Unconditionally stable algorithms for rigid body dynamics that exactly preserve energy and momentum. *International Journal for Numerical Methods in Engineering*, 31(1):19–52, 1991.

[SW95] Chadchart Sittipunt and Sharon L. Wood. Influence of web reinforcement on the cyclic response of structural walls. *ACI Journal*, 92(6):745–756, November-December 1995.

[SW97] John Salmon and Michael S. Warren. Parallel, out–of–core methods for N–body simulations. In M. Heath, V. Torczon, and et al., editors, *Proceedings of the Eight Conference on Parallel Processing for Scientific Computing*. SIAM, 1997.

[SW04] Malcolm C. Smith and Fu-Cheng Wang. Performance benefits in passive vehicle suspensions employing inerters. *Vehicle System Dynamics*, 42(4):235–257, 2004.

[SW07] Purnendu Singh and Wesley W. Wallender. Effecitve stress from force balance on submerged granular particles. *ASCE International Journal of Geomechanics*, 7(3):186–193, May/June 2007.

[Swa98] Travis Swatson. 3D data visualization and modeling with a haptic interface. UCDavis seminar slides, May 1998.

[SWB86] A. Samuelsson, N-E Wiberg, and L. Bernspång. A study of the efficiency of iterative methods for linear problems in structural mechanics. *International Journal for Numerical Methods in Engineering*, 22:209–218, 1986.

[SWB16] Shabnam J Semnani, Joshua A White, and Ronaldo I Borja. Thermoplasticity and strain localization in transversely isotropic materials based on anisotropic critical state plasticity. *International Journal for Numerical and Analytical Methods in Geomechanics*, 40(18):2423–2449, 2016.

[SWHA+19] J.W. Salamon, C. Wood, M.A. Hariri-Ardebili, R. Malm, and G. Faggiani. Seismic analysis of pine flat concrete dam: Formulation and synthesis of results. In G MAZZA, editor, *Proceedings of the 15th International Benchmark Workshop on Numerical Analysis of Dams*, pages 3–97. ICOLD, PoLiMi, September 2019.

[SWK05] D. Sakellaraki, G. Watanabe, and K. Kawashima. Experimental rocking response of direct foundations of bridges. In *Second International Conference on Urban Earthquake Engineering,*, pages 497–504, Tokyo, Japan, March 7-8 2005. Tokyo Institute of Technology.

[SWM+23] Jonathan P. Stewart, Nathaniel Wagner, Debra Murphy, Jeremy Butkovich, Micaela Largent, Hamid Nouri, Hannah Curran, Darcie Maffioli, and John A. Egan. Foundation settlement and tilt of millennium tower in san francisco, california. *Journal of Geotechnical and Geoenvironmental Engineering*, 149(6):05023002, 2023.

[SWS07] Daichao Sheng, Peter Wriggers, and Scott W. Sloan. Application of frictional contact in geotechnical engineering. *ASCE International Journal of Geomechanics*, 7(3):176–185, May/June 2007.

[SWW+01] J. Carl Stepp, Ivan Wong, John Whitney, Richard Quittmeyer, Norman Abrahamson, Gabriel Toro, Robert Youngs, Kevin Coppersmith, Jean Savy, and Tim Sullivan. Probabilistic seismic hazard analysis for ground motions and fault displacement at Yucca Mountain, Nevada. *Earthquake Spectra*, 17:113–151, 2001.

[SWZP84] B. R. Simon, J. S.–S. Wu, O. C. Zienkiewicz, and D. K. Paul. Evaluation of $u - w$ and $u - \pi$ finite element methods for the dynamic response of saturated porous media using one–dimensional models. *International Journal for Numerical and analytical methods in Geomechanics*, 10:461–482, 1984.

[SXW+21] Botan Shen, Weibing Xu, Jin Wang, Yanjiang Chen, Weiming Yan, Jianhui Huang, and Zhenyun Tang. Seismic control of super high-rise structures with double-layer tuned particle damper. *Earthquake Engineering & Structural Dynamics*, 50(3):791–810, 2021.

[SY05] Ryuji Shioya and Genki Yagawa. Large-scale parallel finite-element analysis using the internet: a performance study. *International Journal for Numerical Methods in Engineering*, 63(2):218–230, 2005.

[SYM08] M.A. Saadeghvaziri and A.R. Yazdani-Motlagh. Seismic behavior and capacity/demand analyses of three multi–span simply supported bridges. *Engineering Structures*, 30:54–66, 2008.

[Sym17] Ian Symington. The big issues. NAFEMS web site, July 2017.

[SYMR00] M.A. Saadeghvaziri, A.R. Yazdani-Motlagh, and S. Rashidi. Effects of soil-structure interaction on longitudinal seismic response of msss bridges. *Soil Dyanmics and Earthquake Engineering*, 20:231–242, 2000.

[SZ01] M Souli and JP Zolesio. Arbitrary lagrangian–eulerian and free surface methods in fluid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 191(3):451–466, 2001.

[SZ06] A. Samuelsson and O. C. Zienkiewicz. History of the stiffness method. *International Journal for Numerical Methods in Engineering*, 67:149–157, 2006.

[Sza09] Lásló Szabó. A semi-analytical integration method for J2 flow theory of plasticity with linear isotropic hardening. *Computational Methods in Applied Mechanics and Engineering*, 198(27-29):2151–2166, May 1009.

[SZP84] B. R. Simon, O. C. Zienkiewicz, and D. K. Paul. An analytical solution for the transient response of saturated porous elastic solids. *International Journal for Numerical and analytical methods in Geomechanics*, 8:381–398, 1984.

[Taf12] Nima Tafazzoli. *Methods, Computational Platform, Verification, and Application of Earthquake-Soil-Structure-Interaction Modeling and Simulation*. PhD thesis, University of California at Davis, 2012.

[Tak98] M. Takeo. Ground rotational motions recorded in near-source region of earthquakes. *Geophysical research letters*, 25(6):789–792, 1998.

[Tak13] Haruo Takizawa. *Continuing Lessons for Dynamics of Structures*. Hokkaido University Press, Sapporo, Japan, 2013.

[Tak07] Izuru Takewaki. Closed-form sensitivity of earthquake input energy to soil-structure interaction system. *ASCE Journal of Engineering Mechanics*, 133(4):389–399, 207.

[Tan07] M. El Tani. Hydrostatic paradox of saturated media. *Géotechnique*, 57(9):773–777, 2007.

[TAS86] M Tabatabaie, N Abrahamson, and JP Singh. Effect of seismic wave inclination on structural response. In *Dynamic Response of Structures*, pages 613–620. ASCE, 1986.

[TAS97] Gabriel R. Toro, Norman A. Abrahamson, and John F. Schneider. Model of strong ground motions from earthquakes in Central and Eastern North America: Best estimates and uncertainties. *Seismological Research Letters*, 68(1):41–57, January/February 1997.

[Tat95] Fumio Tatsuoka. Discussions: Response predictions of granular materials at low effective stresses. *Journal of Geotechnical Engineering*, 121(9):678–680, September 1995.

[Tay85] Robert L. Taylor. Solution of linear equations by a profile solver. *Engineering Computations*, 2:344 – 350, December 1985.

[TB99] Patricia A. Thomas and Jonathan D. Bray. Capturing nonspherical shape of granular media with disk clusters. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 125(3):169–178, March 1999.

[TB11] Ricardo Taborda and Jacobo Bielak. Large-scale earthquake simulation: Computational seismology and complex engineering system. *Computing in Science and Engineering*, 13(4):14–27, Sept. 2011.

[TB13] Ricardo Taborda and Jacobo Bielak. Ground-motion simulation and validation of the 2008 Chino Hills, California, earthquake. *Bulletin of the Seismological Society of America*, 103(1):131–156, February 2013.

[TB19] Ganyu Teng and Jack Baker. Evaluation of SCEC cybershake ground motions for engineering practice. *Earthquake Spectra*, 35(3):1311–1328, 2019.

[TBC91] A. M. Trochanis, J. Bielak, and P. Christiano. Three-dimensional nonlinear study of piles. *Journal of Geotechnical Engineering*, 117(3):429–447, March 1991.

[TBKG09] Eric M Thompson, Laurie G Baise, Robert E Kayen, and Bojan B Guzina. Impediments to predicting site response: Seismic property estimation and modeling simplifications. *Bulletin of the Seismological Society of America*, 99(5):2927–2949, 2009.

[TBN20] Anteneh Biru Tsegaye, Thomas Benz, and Steinar Nordal. Formulation of non-coaxial plastic dissipation and stress-dilatancy relations for geomaterials. *Acta Geotechnica*, 15(10):2727–2739, 2020.

[TBW76] Robert L. Taylor, Peter J. Berensford, and Edward L. Wilson. A non–conforming element for stress analysis. *International Journal for Numerical Methods in Engineering*, 10:1211–1219, 1976.

[TC07] Chin Tung Cheng. Energy dissipation in rocking bridge piers under free vibration tests. *Earthquake Engineering & Structural Dynamics*, 36(4):503–518, 2007.

[TCC10] B. Tegeler, S. Chakrabarti, and M. Chakravorty. U.s. regulatory lessons learned from new nuclear power plant applications on evaluating soil-structure interaction. In *Proceedings of the OECD – NEA – IAGE – ISSC Workshop on Soil Structure Interaction Knowledge and Effect on the Seismic Assessment of NPPs Structures and Components*, 2010.

[TCMT56] M.J. Turner, R.W. Clough, H.C. Martin, and L.J. Topp. Stiffness and deflection analysis of complex structures. *Journal of Aeronautical Sciences*, 23(9):805–823, 1956.

[TD08] Mahdi Taiebat and Yannis F. Dafalias. SANISAND: Simple anisotropic sand plasticity model. *International Journal for Numerical and Analytical Methods in Geomechanics*, 2008. (in print, available in earlyview).

[TDG$^+$18] M. Tomasin, M. Domaneschi, C. Guerini, L. Martinelli, and F. Perotti. A comprehensive approach to small and large-scale effects of earthquake motion variability. *Computers & Structures*, 207:155 – 170, 2018. CIVIL-COMP 2017.

[TDI91] M. Nafi Toksöz, Anton M. Dainty, and E. E. Charrette III. Spatial variation of ground motion due to lateral heterogeneity. *Structural Safety*, 10:53–77, 1991.

[TDP10] Mahdi Taiebat, Yannis F. Dafalias, and Ralf Peek. A destructuration theory and its application to SANICLAY model. *International Journal for Numerical and Analytical Methods in Geomechanics*, 34(10):1009–1040, 2010. DOI: 10.1002/nag.841.

[TDT$^+$21] Payman Khalili Tehrani, Natalie Doulgerakis, Iman Talebinejad, Benjamin Kosbab, Michael Cohen, and Andrew Whittaker. Software verification and validation guidelines for non-linear soil-structure interaction analysis. Technical report, SC Solutions Inc., 1261 Oakmead Pkwy, Sunnyvale, CA 94085, November 2021. Project: DE-NE0008857.

[TDV$^+$06] Ko-Foa Tchon, Julien Dompierre, Marie-Gabrielle Vallet, François Guibault, and Ricardo Camarero. Two-dimensional metric tensor visualization using pseudo-meshes. *Engineering with Computers*, 22(2):121–131, September 2006.

[TE03] Paulos B Tekie and Bruce R Ellingwood. Seismic fragility assessment of concrete gravity dams. *Earthquake engineering & structural dynamics*, 32(14):2221–2240, 2003.

[Tem85] Roger Temam. *Mathematical Problems in Plasticity*. Gauthier–Villars, 1985.

[Ter] TeraScale, LLC. Terascale mechanics software development company. `http://www.eterascale.com/`.

[Ter43] K Terzaghi. *Theory of consolidation*. Wiley Online Library, 1943.

[TFPC05] Elisa Tinti, Eiichi Fukuyama, Alessio Piatanesi, and Massimo Cocco. A kinematic source-time function compatible with earthquake dynamics. *Bulletin of the Seismological Society of America*, 95(4):1211–1223, 2005.

[TGB$^+$08] Stefan Turek, Dominik Gëddeke, Christian Becker, Sven Buijssen, and Hilmar Wobker. Uchpc - unconventional high performance computing for finite element simulations. In *Proceedings of the International Supercomputing Conference*, Dresden, June 2008.

[TGN$^+$23] K. Trevlopoulos, P. Gehl, C. Negulescu, H. Crowley, and L. Danciu. Testing the 2020 European seismic hazard and risk models using data from the 2019 Le Teil (France) earthquake. *EGUsphere*, 2023:1–21, 2023.

[TH02] E. Taciroglu and K. D. Hjelmstad. Simple nonlinear model for elastic response of cohesionless granular materials. *ASCE Journal of Engineering Mechanics*, 128(9):821–830, September 2002.

[The98] The Mathworks. Guide, matlab user's. *Inc., Natick, MA*, 5:333, 1998.

[The05] The Mathworks Company. Matlab. `http://www.mathworks.com`, 2005.

[THH$^+$06] Naoki Takahashi, Masayuki Hyodo, Adrian F. L. Hyde, Yoichi Yamamoto, and Shinya Kimura. Online earthquake response test for stratified layers of clay and sand. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 132(5):611–621, May 2006.

[Tho50] William T Thomson. Transmission of elastic waves through a stratified solid medium. *Journal of Applied Physics*, 21(2):89–93, 1950.

[Tho00] Colin Thornton. Numerical simulations of deviatoric shear deformation of granular media. *Géotechnique*, 50(1):43–53, 2000.

[THS98a] J. J. Tweedie, D. N. Humphrey, and T. S. Sanford. Tire shreds as lightweight retaining wall backfill: Active conditions. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 124(11):1061–1070, 1998.

[THS98b] Jeffrey J. Tweedie, Dana M. Humphrey, and Thomas C. Sanford. Full–scale field trials of tire shreds as lightweight retaining wall backfill under at–rest conditions. *Transportaition Research Record*, 1619:64–71, 1998.

[THT01] M.D. Trifunac, T.Y. Hao, and M.I. Todorovska. On energy flow in earthquake response. Technical Report CE 01-03, University of Southern California, Los Angeles, California, 2001.

[Thu61] Alagiah Thurairajah. *Some properties of kaolin and of sand*. PhD thesis, Cambridge University, 1961.

[Tim40] Stephen Timoshenko. *Strength of Materials. Part I: Elementary Theory and Problems*. D. Van Nostrand Company, Inc., second edition – tenth printing edition, 1940.

[Tim53] Stephen S. Timoshenko. *History of Strength of Materials*. McGraw–Hill, Book Company, Inc., 1953.

[Tin85] T. C. T. Ting. Determination of $C^{1/2}$, $C^{-1/2}$ and more general isotropic tensor functions of C. *Journal of Elasticity*, 15:319–323, 1985.

[TIT^{+}99] MD Trifunac, SS Ivanovć, MI Todorovska, EI Novikova, and AA Gladkov. Experimental evidence for flexibility of a building foundation supported by concrete friction piles. *Soil Dynamics and Earthquake Engineering*, 18(3):169–187, 1999.

[TJ03] N.P. Tongaonkar and R.S. Jangid. Seismic response of isolated bridges with soil–structure interaction. *Soil Dyanmics and Earthquake Engineering*, 23:287–302, 2003.

[TJD10a] M. Taiebat, B. Jeremić, and Y. F. Dafalias. Prediction of seismically induced voids and pore fluid volume/pressure redistribution in geotechnical earthquake engineering. In *Proceedings of Sixty Third Canadian Geotechnical Conference & Sixth Canadian Permafrost Conference*, pages 233–237, Calgary, AB, Canada, September 12–16 2010.

[TJD^{+}10b] Mahdi Taiebat, Boris Jeremić, Yannis F. Dafalias, Amir M. Kaynia, and Zhao Cheng. Propagation of seismic waves through liquefied soils. *Soil Dynamics and Earthquake Engineering*, 30(4):236–257, 2010.

[TJDK09] Mahdi Taiebat, Boris Jeremić, Yannis F. Dafalias, and Amir M. Kaynia. Earthquake–induced shear deformation of slopes for performance–based engineering. In Kokusho, Tsukamoto, and Yoshimine, editors, *Performance-Based Design in Earthquake Geotechnical Engineering: From Case History to Practice*, pages 907–914, Tsukuba, Japan, 2009. Taylor & Francis Group, London.

[TJK09] Mahdi Taiebat, Boris Jeremić, and Amir M. Kaynia. Propagation of seismic waves through liquefied soils. In *In proceedings of the GeoOrlando, Geo Institute Annual Conference*, Orlando, Florida, March 2009.

[TK02] Hjörtur Thráinsson and Anne S Kiremidjian. Simulation of digital earthquake accelerograms using the inverse discrete fourier transform. *Earthquake engineering & structural dynamics*, 31(12):2023–2048, 2002.

[TKMR93] J.M. Ting, M. Khwaja, L.R. Meachum, and J.D. Rowell. An ellipse-based discrete element model for granular materials. *International Journal for Numerical and Analytical Methods in Geomechanics*, 17:603–623, 1993.

[TKW00] Hjörtur Thráinsson, Anne Setian Kiremidjian, and Steven R Winterstein. Modeling of earthquake ground motion in the frequency domain. Technical report, John A. Blume Earthquake Engineering Center, 2000.

[TL09] Robert L. Taylor and Roland W. Lewis. Olgierd C. Zienkiewicz (18 may 1921-2 january 2009). *International Journal for Numerical Methods in Engineering*, 80(1):2009, 2009.

[TLOT91] W.S. Tseng, K. Lilhanand, F. Ostadan, and S.Y. Tuann. Post-earthquake analysis and data correlation for the 1/4-scale containment model of the Lotung experiment. Technical Report NP-7305-SL, EPRI, Electric Power Research Institute, 3412 Hillview Avenue, Palo Alto, California, 94304, 1991.

[TM] Inc. The MathWorks. Matlab. `http://www.mathworks.com`

[TMGJ04] Horacio Tapia-McClung and Niels Grønbech-Jensen. Non–iterative and exact method for constraining particles in a linear geometry. *Journal of Polymer Science B: Polymer Physics*, 2004. In Review.

[TMM06] Roman Teisseyre, Takeo Minoru, and Eugeniusz Majewski. *Earthquake source asymmetry, structural media and rotation effects*. Springer, 2006.

[TMM08] Rabin Tuladhar, Takeshi Maki, and Hiroshi Mutsuyoshi. Cyclic behavior of laterally loaded concrete piles embedded into cohesive soil. *Earthquake Engineering & Structural Dynamics*, 37(1):43–59, January 2008.

[TN87] Jawahar M Tembulkar and James M Nau. Inelastic modeling and seismic energy dissipation. *Journal of Structural Engineering*, 113(6):1373–1377, 1987.

[TN04] Tang Tat Ng. Triaxial test simulations with discrete element method and hydrostatic boundaries. *ASCE Journal of Engineering Mechanics*, 130(10):1188–1194, October 2004.

[TNTD08] Hamid Toopchi-Nezhad, Michael J. Tait, and Robert G. Drysdale. Lateral response evaluation of fiber-reinforced neoprene seismic isolators utilized in an unbonded application. *ASCE Journal of Structural Engineering*, 134(10):1627–1637, October 2008.

[To92] C.W.S. To. A stochastic version of the Newmark family of algorithms for discretized dynamic systems. *Computers & Structures*, 44(3):667 – 673, 1992.

[TO04] P. Thoutireddy and M. Ortiz. A variational r-adaption and shape-optimization method for finite-deformation elasticity. *International Journal for Numerical Methods in Engineering*, 61(1):1–21, June 2004.

[TO05] Markus Tuller and Dani Or. Water films and scaling of soil characteristic curves at low water contents. *Water Resources Research*, 41(9), 2005.

[Tod93] Maria I Todorovska. Effects of the wave passage and the embedment depth for in-plane building-soil interaction. *Soil Dynamics and Earthquake Engineering*, 12(6):343–355, 1993.

[Tol96] David G. Toll. Artificial intelligence applications in geotechnical engineering. *Electronic Journal of Geotechnical Engineering*, October 1996. available at http://139.78.66.61/ejge/.

[Tom94] Yoshihiro Tomita. Simulations of plastic instabilites in solid mechanics. *Applied Mechanics Review*, 47(6):s32–s42, 1994. part 2.

[Tou64] R. A. Toupin. Theories of elasticity with couple-stress. *Archive for Rational Mechanics and Analysis*, 17(2):85–112, 1964.

[Toz16] Christopher Tozzi. Linux at 25. *IEEE Spectrum*, pages 48–57, April 2016.

[TP19] Hing-Ho Tsang and Kyriazis Pitilakis. Mechanism of geotechnical seismic isolation system: Analytical modeling. *Soil Dynamics and Earthquake Engineering*, 122:171 – 184, 2019.

[TPdPM15] F. Tradigo, F. Pisanò, C. di Prisco, and A. Mussi. Non-linear soil-structure interaction in disconnected piled raft foundations. *Computers and Geotechnics*, 63(0):121 – 134, 2015.

[TPM96] Karl Terzaghi, Ralph B. Peck, and Gholamreza Mesri. *Soil Mechanics in Engineering Practice.* John Wiley & Sons, Inc., third edition, 1996.

[TPSH92] M. Thiriet, C. Pares, E. Saltel, and F. Hecht. Numerical simulation of steady flow in a model of the aortic bifurcation. *ASME Journal of Biomechanical Engineering*, 114:40–49, 1992.

[TPW$^+$14] Antoinette Tordesillas, Sebastian Pucilowski, David M. Walker, John F. Peters, and Laura E. Walizer. Micromechanics of vortices in granular media: connection to shear bands and implications for continuum modelling of failure in geomaterials. *International Journal for Numerical and Analytical Methods in Geomechanics*, 38(12):1247–1275, 2014.

[TPZ16] D.M.G. Taborda, D.M. Potts, and L. Zdravković. On the assessment of energy dissipated through hysteresis in finite element analysis. *Computers and Geotechnics*, 71:180 – 194, 2016.

[TQ31] Geoffrey Ingram Taylor and H. Quinney. The plastic distortion of metals. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 230(681-693):323–362, 1931.

[TQ34] Geoffrey Ingram Taylor and H Quinney. The latent energy remaining in a metal after cold working. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 143(849):307–326, 1934.

[TR14] Hooman Torabi and Mohammad T Rayhani. Three dimensional finite element modeling of seismic soil–structure interaction in soft soil. *Computers and Geotechnics*, 60:9–19, 2014.

[TRB95] Andrew W. Taylor, Randall B. Rowell, and John E. Breen. Behavior of thin–walled concrete box piers. *ACI Structural Journal*, 92(3):319–333, May–June 1995.

[Tri82] MD Trifunac. A note on rotational components of earthquake motions on ground surface for incident body waves. *International Journal of Soil Dynamics and Earthquake Engineering*, 1(1):11–19, 1982.

[Tri94] M.D. Trifunac. Q and high-frequency strong motion spectra. *Soil Dynamics and Earthquake Engineering*, 13:149–161, 1994.

[Tri05] M.D. Trifunac. Power design method. In *Proceedings Earthquake Engineering in the 21st Century*, Skopje, Ohrid, Macedonia, August 27 - September 1 2005. IZIIS.

[Tri08a] M.D. Trifunac. Energy of strong motion at earthquake source. *Soil Dynamics and Earthquake Engineering*, 28(1):1–6, January 2008.

[Tri08b] Mihailo D. Trifunac. Early history of the response spectrum method. *Soil Dynamics and Earthquake Engineering*, 28(9):676–685, September 2008.

[Tri09] Mihailo D. Trifunac. 75th anniversary of strong motion observation: A historical review. *Soil Dynamics and Earthquake Engineering*, 29(4):591–606, April 2009.

[Tri12] Mihailo D. Trifunac. Earthquake response spectra for performance based design – critical review. *Soil Dynamics and Earthquake Engineering*, 37(0):73 – 83, 2012.

[Tru55] C Truesdell. Hypo-elasticity. *Journal of Rational Mechanics and Analysis*, 4:83–1020, 1955.

[Tru63] C Truesdell. Remarks on hypo-elasticity. *Journal of Research of the National Bureau of Standards-B. Mathematics and Mathematical Physica*, 67(3):141–143, 1963.

[Tru25a] Donald J. Trump. ORDERING THE REFORM OF THE NUCLEAR REGULATORY COMMISSION. https://www.whitehouse.gov/presidential-actions/2025/05/ordering-the-reform-of-the-nuclear-regulatory-commission/, 23 May 2025. Presidential Executive Order.

[Tru25b] Donald J. Trump. REFORMING NUCLEAR REACTOR TESTING AT THE DEPARTMENT OF ENERGY. https://www.whitehouse.gov/presidential-actions/2025/05/reforming-nuclear-reactor-testing-at-the-department-of-energy/, 23 May 2025. Presidential Executive Order.

[Tru25c] Donald J. Trump. Unleashing American Energy. https://www.whitehouse.gov/presidential-actions/2025/01/unleashing-american-energy/, 20 January 2025. Presidential Executive Order.

[TS85] T.–S. Tan and R. F. Scott. Centrifuge scaling considerations for fluid – particle systems. *Géotechnique*, 35(4):461–470, 1985.

[TSB07] M. Tabatabaie, B. Sumodobila, and T. Ballard. Time-domain ssi analysis of typical reactor building using frequency-dependent foundation impedance derived from SASSI. In *Transactions on the 19th International Conference on Structural Mechanics in Reactor Technology*, August 2007.

[TSF91] Fabio Taucer, Enrico Spacone, and Filip C Filippou. A fiber beam-column element for seismic response analysis of reinforced concrete structures. Technical Report Report No. UCB/EERC-91/17, Earthquake Engineering Research Center, College of Engineering, University of California, Berkeley, December 1991.

[TSMB21] Denny Thaler, Marcus Stoffel, Bernd Markert, and Franz Bamer. Machine-learning-enhanced tail end prediction of structural response statistics in earthquake engineering. *Earthquake Engineering & Structural Dynamics*, 50(8):2098–2114, 2021.

[TSN11] Salih Tileylioglu, Jonathon P. Stewart, and Rober L. Nigbor. Dynamic stiffness and damping of a shallow foundation from forced vibration of a field test structure. *Journal of Geotechnical and Geoenvironmental Engineering*, 137(4):344–353, April 2011.

[TSN$^+$25] Alessandro Tombari, Luciano Stefanini, Giovanni Li Destri Nicosia, Liam M.J. Holland, and Marcus Dobbs. Geotechnical data-driven possibility reliability assessment. *Computers and Geotechnics*, 185:107311, 2025.

[TSOTR07a] Leonardo Tuñón-Sanjur, Richard S. Orr, Sener Tinic, and Diego Peña Riuz. Finite element modeling of the ap1000 nuclear island for seismic analyses at generic soil and rock sites. *Nuclear Engineering and Design*, 237(12-13):1474–1485, July 2007.

[TSOTR07b] Leonardo Tunon-Sanjura, Richard S. Orr, Sener Tinic, and Diego Pena Ruiz. Finite element modeling of the ap1000 nuclear island for seismic analyses at generic soil and rock sites. *Nuclear Engineering and Design*, 237(12-13):1474–1485, July 2007.

[TSP07] Mahdi Taiebat, Hadi Shahir, and Ali Pak. Study of pore pressure variation during liquefaction using two constitutive models for sand. *Soil Dynamics and Earthquake Engineering*, 27:60–72, 2007.

[Tsy98] Semyon V Tsynkov. Numerical solution of problems on unbounded domains. a review. *Applied Numerical Mathematics*, 27(4):465–532, 1998.

[TSZC86] R.L. Taylor, J.C. Simo, O.C. Zienkiewicz, and A.C.H. Chen. The patch test – a condition for assessing FEM convergence. *International Journal for Numerical Methods in Engineering*, 22:39–62, 1986.

[TT89] MI Todorovska and MD Trifunac. Antiplane earthquake waves in long structures. *Journal of engineering mechanics*, 115(12):2687–2708, 1989.

[TT90] MI Todorovska and MD Trifunac. Note on excitation of long structures by ground waves. *Journal of Engineering Mechanics*, 116(4):952–964, 1990.

[TT92] MI Todorovska and MD Trifunac. The system damping, the system frequency and the system response peak amplitudes during in-plane building-soil interaction. *Earthquake engineering & structural dynamics*, 21(2):127–144, 1992.

[TT98] M.D. Trifunac and M.I. Todorovska. Nonlinear soil response as a natural passive isolation mechanism - the 1994 Northridge, California, earthquake. *Soil Dynamics and Earthquake Engineering*, 17(1):41–51, January 1998.

[TT99] M.D. Trifunac and M.I. Todorovska. Reduction of structural damage by nonlinear soil response. *Journal of Structural Engineering*, 125(1):89–97, January 1999.

[TT13a] M. Tannenbaum and M. Tulay. Guideline for the acceptance of commercial-grade design and analysis computer programs used in nuclear safety-related applications. Technical Report 3002002289, 2013, Revision 1 of 1025243, Electric Power Research Institute (EPRI), 420 Hillview Avenue, Palo Alto, CA 94304-1338,USA, December 2013.

[TT13b] M. Tannenbaum and M. Tulay. Plant engineering: Guideline for the acceptance of commercial-grade design and analysis computer programs used in nuclear safety-related applications. Technical Report 3002002289, Electric Power Research Institute, EPRI, 3420 Hillview Ave, Palo Alto, CA, 943003-1338, USA, December 2013.

[TT13c] Mihailo D Trifunac and Maria I Todorovska. A note on energy of strong ground motion during northridge, california, earthquake of january 17, 1994. *Soil Dynamics and Earthquake Engineering*, 47:175–184, 2013.

[TTDO15] Maria I. Todorovska, Mihailo D. Trifunac, Haiping Ding, and Nebojša Orbović. Coherency of dispersed synthetic earthquake ground motion at small separation distances: Dependence on site conditions. *Soil Dynamics and Earthquake Engineering*, 79, Part A:253 – 264, 2015.

[TTLO13] Maria I. Todorovska, Mihailo D. Trifunac, Vincent W. Lee, and Nebojša Orbović. Synthetic earthquake ground motions on an array. *Soil Dynamics and Earthquake Engineering*, 48(0):234 – 251, 2013.

[TTM^{+}24] Matteo Torzoni, Marco Tezzele, Stefano Mariani, Andrea Manzoni, and Karen E. Willcox. A digital twin framework for civil engineering structures. *Computer Methods in Applied Mechanics and Engineering*, 418:116584, 2024.

[TTS^{+}89] H.T Tang, Y.K Tang, J.C Stepp, I.B Wall, E Lin, S.C Cheng, and S.K Lee. A large-scale soil-structure interaction experiment: Design and construction. *Nuclear Engineering and Design*, 111(3):371 – 379, 1989.

[TTS90] H.T Tang, Y.K Tang, and J.C Stepp. Lotung large-scale seismic experiment and soil-structure interaction method validation. *Nuclear Engineering and Design*, 123(2-133):397 – 412, 1990.

[TTTJ15a] Panagiota Tasiopoulou, Mahdi Taiebat, Nima Tafazzoli, and Boris Jeremić. On validation of fully coupled behavior of porous media using centrifuge test results. *Coupled Systems Mechanics Journal*, 4(1):37–65, 2015. DOI: http://dx.doi.org/10.12989/csm.2015.4.1.037.

[TTTJ15b] Panagiota Tasiopoulou, Mahdi Taiebat, Nima Tafazzoli, and Boris Jeremić. Solution verification procedures for modeling and simulation of fully coupled porous media: Static and dynamic behavior. *Coupled Systems Mechanics Journal*, 4(1):67–98, 2015. DOI: http://dx.doi.org/10.12989/csm.2015.4.1.067.

[TVCD00] Claudio Tamagnini, Gioacchino Viggiani, René Chambon, and Jacques Desrues. Evaluation of different strategies for the integration of hypoplastic constitutive equations: Application to the cloe model. *Mechanics of Cohesive-frictional Materials*, 5(4):263–289, 2000.

[TW93] J Tejchman and W Wu. Numerical study on patterning of shear bands in a Cosserat continuum. *Acta Mechanica*, 99(1-4):61–74, 1993.

[TW95] Jacek Tejchman and Wei Wu. Experimental and numerical study of sand–steel interfaces. *International journal for numerical and analytical methods in geomechanics*, 19(8):513–536, 1995.

[TWC10] Jin Ting Wang and Anil K. Chopra. Linear analysis of concrete arch dams including dam-water-foundation rock interaction considering spatially varying ground motions. *Earthquake Engineering & Structural Dynamics*, 39(7):731–750, 2010.

[TWK59] Stephen P Timoshenko and Sergius Woinowsky-Krieger. *Theory of plates and shells*. McGraw-hill, 1959.

[TWR98] Shin Tower Wang and Lymon C. Reese. Design of pile foundations in liquefied soils. In Panos Dakoulas, Mishac Yegian, and Robert D. Holtz, editors, *Proceedings of a Specialty Conference: Geotechnical Earthwuake Engineering and Soil Dynamics III*, Geotechnical Special Publication No. 75, pages 1331–1343. ASCE, August 1998. 1998.

[TWZ$^+$14] Patricia Thomas, Ivon Wong, Judith Zachariasen, Robert Darragh, and Walt Silva. 2013 update to the site-specific seismic hazard analysis and development of seismic design ground motions. Technical report, URS Corporation, Oakland, CA, 2014.

[Tya07] A. Tyapin. The frequency–dependent elements in the code SASSI: A bridge between civil engineers and the soil–structure interaction specialists. *Nuclear Engineering and Design*, 237:1300–1306, 2007.

[TYHT01] Mihailo D. Trifunac, Tzong Ying Hao, and Maria I. Todorovska. Energy of earthquake as a design tool. In *Proceedings of 13th Mexican Conference on Earthquake Engineering, XIII MCEE*, pages 1–49, Guadalajara, Mexico, 31th October – 3rd November 2001.

[TZ06] Andrzej Truty and Thomas Zimmermann. Stabilized mixed finite element formulations for materially nonlinear partially saturated two-phase media. *Computer Methods in Applied Mechanics and Engineering*, 195(13-16):1517–1546, February 2006.

[UB90] Chia-Ming Uang and Vitelmo V Bertero. Evaluation of seismic energy in structures. *Earthquake Engineering & Structural Dynamics*, 19(1):77–90, 1990.

[UC92] Tzou-Shin Ueng and Jian-Chu Chen. Computational procedures for determining parameters in ramberg-osgood elastoplastic model based on modulus and damping versus strain. Technical Report UCRL-ID-111487, Lawrence Livermore National Laboratory, 1992.

[UC07] Berna Unutmaz and K Onder Cetin. Effects of soil-structure-earthquake-interaction on seismic soil liquefaction triggering. In *Proceedings of fourth international conference on earthquake geotechnical engineering*, 2007.

[UČS$^+$08] R. Uzuoka, M. Čubrinovski, H. Sugita, M. Sato, K. Tokimatsu, N. Sento, M. Kazama, F. Zhang, A. Yashima, and F. Oka. Prediction of pile response to lateral spreading by 3-d soil–coupled dynamic analysis: Shaking in the direction perpendicular to ground flow. *Soil Dynamics and Earthquake Engineering*, 28(6):436–452, June 2008.

[UK86a] Morimichi Uesugi and Hideaki Kishida. Frictional resistance at yield between dry sand and mild steel. *Soils and foundations*, 26(4):139–149, 1986.

[UK86b] Morimichi Uesugi and Hideaki Kishida. Influential factors of friction between steel and dry sands. *Soils and foundations*, 26(2):33–46, 1986.

[ÜKN89] Kahraman Ünlü, M Levent Kavvas, and Donald R Nielsen. Stochastic analysis of field measured unsaturated hydraulic conductivity. *Water Resources Research*, 25(12):2511–2519, December 1989.

[UKOY97] Teruyuki Ueshima, Takeji Kokusho, Toshiro Okamoto, and Hiroshi Yajima. Seismic response analysis of embedded structure at hualien, taiwan. *Nuclear Engineering and Design*, 172(3):289 – 295, 1997.

[UKT89] Morimichi Uesugi, Hideaki Kishida, and Yasunori Tsubakihara. Friction between sand and steel under repeated loading. *Soils and foundations*, 29(3):127–137, 1989.

[UKU90] Morimichi Uesugi, Hideaki Kishida, and Yuichiro Uchikawa. Friction between dry sand and concrete under monotonic and repeated loading. *Soils and Foundations*, 30(1):115–128, 1990.

[Ull08] E. Ullmann. *Solution strategies for stochastic finite element discretizations*. Doctoral dissertation, TU Bergakademie Freiberg, 2008.

[ULNP06] Marco Uzielli, Suzanne Lacasse, Farrokh Nadim, and Kok-Kwang Phoon. Soil variability analysis for geotechnical practice. In *Proceedings of the 2nd International Workshop on Characterisation and Engineering Properties of Natural Soils*, volume 3, 12 2006.

[UNN13] US-NRC-NUREG-1824. Verification and validation of selected fire models for nuclear power plant applications. NUREG 1824, United States Nuclear Regulatory Commission and, 2013.

[UR09] M.B.C. Ulker and M.S. Rahman. Response of saturated and nearly saturated porous media: Different formulations and their applicability. *International Journal for Numerical and Analytical Methods in Geomechanics*, 33(5):633–664, 2009.

[uRLWZ20] Zia ur Rehman, Fangyue Luo, Teng Wang, and Ga Zhang. Large-scale test study on the three-dimensional behavior of the gravel-concrete interface of a CFR dam. *International Journal of Geomechanics*, 20(6):04020046, 2020.

[U.S08] U.S. Nuclear Regulatory Commission. Interim staff guidance on seismic issues of high frequency ground motion. Technical report, DC/COL-ISG-1, Rockville, Maryland, USA, 2008.

[USA16] USA Department of Energy. DOE STANDARD, natural phenomena hazards analysis and design criteria for DOE facilities. Technical Report DOE-STD-1020-2016, USA Department of Energy, Washington, D.C. 20585, 2016.

[USG19] USGBC. LEED v4 for building design and construction. Technical Report v4, U.S. Green Building Council, 2019.

[USK$^+$07] R. Uzuokaa, N. Sentoa, M. Kazamaa, F. Zhangb, A. Yashimac, and F. Oka. Three-dimensional numerical simulation of earthquake damage to group-piles in a liquefied ground. *Soil Dynamics and Earthquake Engineering*, 27(5):395–413, May 2007.

[UUSK13] T Unno, R Uzuoka, N Sento, and M Kazama. Pore air pressure effect on cyclic shear behavior of undrained sandy soil. *Proc. JSCE*, 69:386–403, 2013.

[VA91] I. Vardoulakis and E. C. Aifantis. A gradient flow theory of plasticity for granular materials. *Acta Mechanica*, 87:197–217, 1991.

[VA14] Apostolos Vrakas and Georgios Anagnostou. A finite strain closed-form solution for the elastoplastic ground response curve in tunnelling. *International Journal for Numerical and Analytical Methods in Geomechanics*, pages n/a–n/a, 2014.

[VA15] Manuela Villani and Norman A. Abrahamson. Repeatable site and path effects on the ground-motion sigma based on empirical data from southern california and simulated waveforms from the cybershake platform. *Bulletin of the Seismological Society of America*, 105(5):2681–2695, October 2015.

[VA19] Felipe Vicencio and Nicholas A. Alexander. A parametric study on the effect of rotational ground motions on building structural responses. *Soil Dynamics and Earthquake Engineering*, 118:191 – 206, 2019.

[Vam14] Dimitrios Vamvatsikos. Seismic performance uncertainty estimation via ida with progressive accelerogram-wise latin hypercube sampling. *Journal of Structural Engineering*, 140(8):A4014015, 2014.

[VAMC24] Felipe Vicencio, Nicholas A. Alexander, and Christian Málaga-Chuquitaype. Seismic structure-soil-structure interaction between inelastic structures. *Earthquake Engineering & Structural Dynamics*, n/a(n/a), 2024.

[Van94] Eric H. Vanmarcke. Stochastic finite elements and experimental measurements. *Probabilistic Engineering Mechanics*, 9:103–114, 1994.

[Var81] I. Vardoulakis. Bifurcation analysis of the plane rectilinear deformation on dry sand samples. *International Journal of Solids and Structures*, 17(11):1085–1101, 1981.

[Var89] I. Vardoulakis. Shear-banding and liquefaction in granular materials on the basis of a Cosserat continuum theory. *Ingenieur Archiv*, 59:106–113, 1989.

[Var96a] I. Vardoulakis. Deformation of water-saturated sand: I. uniform undrained deformation and shear banding. *Geotechnique*, 46(3):441–456, 1996.

[Var96b] I. Vardoulakis. Deformation of water-saturated sand: Ii. effect of pore water flow and shear banding. *Geotechnique*, 46(3):457–472, 1996.

[Var04a] I. Vardoulakis. Fluidisation in artesian flow conditions: Hydromechanically stable granular media. *Géotechnique*, 54(2):117–130, January 2004.

[Var04b] I. Vardoulakis. Fluidisation in artesian flow conditions: Hydromechanically unstable granular media. *Géotechnique*, 54(3):165–178, April 2004.

[Var1] Various Authors. The C++ report: Columns on C++, 1991-.

[VAV12] Jasmina Vujić, Dragoljub P. Antić, and Zorka Vukmirović. Environmental impact and cost analysis of coal versus nuclear power: The U.S. case. *Energy*, doi:10.1016/j.energy.2012.02.011:–, 2012.

[VB92] I. Vgenopoulou and D. E. Beskos. Dynamics of saturated rocks. iv: Column and borehole problems. *Journal of Engineering Mechanics*, 118(9):1795–1813, 1992.

[vB96] S. van Baars. Discrete element modeling of granular materials. *Heron*, 41(2):139–157, 1996.

[VB22] Ricardo Vinuesa and Steven L. Brunton. Enhancing computational fluid dynamics with machine learning. *Nature Computational Science*, 2(6):358–366, June 2022.

[VBB$^+$25] Gianmarco Vallero, Monica Barbero, Fabrizio Barpi, Mauro Borri-Brunetto, and Valerio De Biagi. An elasto-visco-plastic constitutive model for snow: Theory and finite element implementation. *Computer Methods in Applied Mechanics and Engineering*, 433:117465, 2025.

[VBM95] Stevan D. Vidić, George H. Beckwith, and Paul W. Mayne. Profiling mine tailings with CPT. In *Proceedings, International Symposium on Cone Penetration Testing CPT ´95*, Linköping, Sweden, October 1995.

[vBM98] Gerald van Belle and Steven P. Millard. STRUTS: statistical rules of thumb. Technical report, University of Washington, 1998.

[VBT$^+$22] Emmanuel Viallet, Julien Berger, Paola Traversa, Elias El Haber, Estelle Hervé-Secourgeon, Guillaume Hervé-Secourgeon, Loic Zuchowski, and Guillaume Dupuy. 2019-11-11 Le Teil earthquake - the ultimate missing piece of experience feedback related to a nuclear power plant built on seismic base isolation: A real earthquake. In *Transactions, SMiRT-26*. SMiRT, July 2022.

[VC02] Dimitrios Vamvatsikos and C. Allin Cornell. Incremental dynamic analysis. *Earthquake Engineering & Structural Dynamics*, 31(3):491–514, 2002.

[VC22] Joseph P. Vantassel and Brady R. Cox. SWprocess: a workflow for developing robust estimates of surface wave dispersion uncertainty. *Journal of Seismology*, 26(4):731–756, 2022.

[VCHY22] Joseph P. Vantassel, Brady R. Cox, Peter G. Hubbard, and Michael Yust. Extracting high-resolution, multi-mode surface wave dispersion data from distributed acoustic sensing measurements using the multichannel analysis of surface waves. *Journal of Applied Geophysics*, 205:104776, 2022.

[VD91] Mladen Vučetić and Ricardo Dobry. Effect of soil plasticity on cyclic response. *ASCE Journal of Geotechnical Engineering*, 117(1):89–107, January 1991.

[vDKV00] A. van Deursen, P. Klint, and J. Visser. Domain-specific languages: An annotated bibliography. *ACM SIGPLAN Notices*, 35(6):26–36, June 2000.

[Vel95a] Todd Veldhuizen. Expression templates. *C++ Report*, 7(5):26–31, June 1995.

[Vel95b] Todd Veldhuizen. Using C++ template metaprograms. *C++ Report*, 7(4):36–43, May 1995.

[Vel96] Todd Veldhuizen. Rapid linear algebra in C++. *Dr. Dobb's Journal*, August 1996.

[Vel98] Todd L Veldhuizen. Arrays in blitz++. In *International Symposium on Computing in Object-Oriented Parallel Environments*, pages 223–230. Springer, 1998.

[Vel05] Todd Veldhuizen. Software libraries and their reuse: Entropy, kolmogorov complexity, and zipf's law. In David Musser, editor, *Library-Centric Software Design LCSD'05*. Object-Oriented Programming, Systems, Languages and Applications, October 2005.

[Ver12] Arnold Verruijt. *Soil Mechanics*. Verruijt Self Published, 2012.

[VF85] K. C. Valanis and Jighong Fan. Experimental verification of endochronic plasticity in spatially varying strain fields. In A. Sawczuk and G. Bianchi, editors, *Plasticity Today: Modelling, Methods and Applications*, pages 153–174. Elsevier Applied Sciences Publishers, 1985.

[VF00] S K Vanapalli and D G Fredlund. Comparison of different procedures to predict unsaturated soil shear strength. In *Advances in Unsaturated Geotechnics*, pages 195–209, 2000.

[VFdBAF97] F. Venancio-Filho, F.C.P. de Barros, M.C.F. Almeida, and W.G. Ferreira. Soil-structure interaction analysis of NPP containments: substructure and frequency domain methods. *Nuclear Engineering and Design*, 174(2):165 – 176, 1997.

[VFSP13] Viktor Vlaski, Adam Fila, Oliver Schneider, and Demetre Papandreou. Reduction of external hazard (fast impact) induced vibrations. In *1st CONFERENCE ON TECHNICAL INNOVATION IN NUCLEAR CIVIL ENGINEERING*, 2013.

[VG80] M Th Van Genuchten. A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil science society of America journal*, 44(5):892–898, 1980.

[VGB20] Pragya Vaishanav, Abhinav Gupta, and Saran Srikanth Bodda. Limitations of traditional tools for beyond design basis external hazard pra. *Nuclear Engineering and Design*, 370:110899, 2020.

[VGG78] I. Vardoulakis, M. Goldscheider, and G. Gudehus. Formation of shear bands in sand bodies as a bifurcation problem. *International Journal for Numerical and Analytical Methods in Geomechanics*, 2:99–128, 1978.

[vH09] Dimitri van Heesch. Doxygen. Technical report, Private Consultant, `http://www.stack.nl/~dimitri/doxygen/index.html`, 2009.

[VHDC11] Chris Van Houtte, Stéphane Drouet, and Fabrice Cotton. Analysis of the origins of κ (kappa) to compute hard rock to rock adjustment factors for GMPEs. *Bulletin of the Seismological Society of America*, 101(6):2926–2941, 2011.

[VI96] Ramon Verdugo and Kenji Ishihara. The steady state of sandy soils. *Soils and foundations*, 36(2):81–91, 1996.

[VitBC] Marcus Pollio Vitruvius. *De architectura, libri decem, The Ten Books on Architecture, Book-I*. old, 30BC.

[VJ88] Božidar D. Vujanović and S. E. Jones. *Variational Methods in Nonconservative Phenomena*. Academic Press Inc., 1988.

[VJ97] Todd L. Veldhuizen and M. Ed Jernigan. Will C++ be faster than FORTRAN. In Jan Hull, editor, *Proceedings of the first Internationa conference Scientific Computing in Object-Oriented Parallel Environments (ISCOPE) Conference*. Springer Verlag, 1997.

[VLAVdVS07] R Van Loon, PD Anderson, FN Van de Vosse, and SJ Sherwin. Comparison of various fluid–structure interaction methods for deformable bodies. *Computers & structures*, 85(11):833–843, 2007.

[VLS80] A. V. Vasilev, E. E. Lovetskii, and V. I. Selyakov. Injection effect in a contained explosion in a liquid-saturated medium. *translated from Zhurnal Prikladnoi Mekhaniki i Tekhnicheskoi Fiziki*, 4:107–112, July – August 1980.

[vLV90] H. van Langen and P. A. Vermeer. Automatic step size correction for non–associated plasticity problems. *International Journal for Numerical Methods in Engineering*, 29:579–598, 1990.

[VM74] Anestis S. Veletsos and Jethro W. Meek. Dynamic behaviour of building-foundation systems. *Earthquake Engineering & Structural Dynamics*, 3(2):121–138, 1974.

[VMC$^+$21] Jan Vorel, Marco Marcon, Gianluca Cusatis, Ferhun Caner, Giovanni Di Luzio, and Roman Wan-Wendner. A comparison of the state of the art models for constitutive modelling of concrete. *Computers & Structures*, 244:106426, 2021.

[VMP$^+$23] Carlos E. Ventura, Mehrtash Motamedi, Yuxin Pan, Solomon Tesfamariam, and Haibei Xiong. Drift- and energy-based seismic performance assessment of retrofitted wood frame shear wall buildings: Shake table tests. *Earthquake Engineering & Structural Dynamics*, n/a(n/a):1–17, 2023.

[vMPRM91] Jan van Mier, Arjan Pruijssers, Hans Reinhardt, and Theo Monnier. Load time response of colliding concrete bodies. *Journal of Structural Engineering*, 117(2):354–374, 1991.

[VMS17] Michalis F. Vassiliou, Kevin R. Mackie, and Božidar Stojadinović. A finite element model for seismic response analysis of deformable rocking frames. *Earthquake Engineering & Structural Dynamics*, 46(3):447–466, 2017. EQE-16-0047.R1.

[VOD02] Ph. Volant, N. Orbovic, and F. Dunand. Seismic evaluation of existing nuclear facility using ambient vibration test to characterize dynamic behavior of the structure and microtremor measurements to characterize the soil: a case study. *Soil Dynamics and Earthquake Engineering*, 22(9ˇ201312):1159 – 1167, 2002.

[Vor08] Oleg Vorobiev. Generic strength model for dry jointed rock masses. *International Journal of Plasticity*, 24(12):2221 – 2247, 2008.

[VPDN17] Cesar M Venier, Cesar I Pairetti, Santiago Marquez Damian, and Norberto M Nigro. On the stability analysis of the piso algorithm on collocated grids. *Computers & Fluids*, 147:25–40, 2017.

[VPPW11] Francisco J. Valdés-Parada, Mark L. Porter, and Brian D. Wood. The role of tortuosity in upscaling. *Transport in Porous Media*, 88(1):1–30, 2011.

[VPW19] Antonios Vytiniotis, Andriani-Ioanna Panagiotidou, and Andrew J. Whittle. Analysis of seismic damage mitigation for a pile-supported wharf structure. *Soil Dynamics and Earthquake Engineering*, 119:21 – 35, 2019.

[VRM] VRML Consortium. Virtual reality modeling language. `http://www.vrml.org/`

[vRP17] Guido van Rossum and Python Software Foundation. Python. `https://www.python.org/`, 2017.

[vRWH96] B. van Rietbergen, H. Weinans, and R. Huiskes. Computational strategies for iterative solutions of large FEM applications employing voxel data. *International Journal for Numerical Methods in Engineering*, 39:2743–2767, 1996.

[VS95] I. Vardoulakis and J. Sulem. *Bifurcation Analysis in Geomechanics*. Blackie Academic & Professional, 1995. ISBN 0-7514-0214-1.

[VS21] Nikolaos N. Vlassis and WaiChing Sun. Sobolev training of thermodynamic-informed neural networks for interpretable elasto-plasticity models with level set hardening. *Computer Methods in Applied Mechanics and Engineering*, 377:113695, 2021.

[VSC19] Swetha Veeraraghavan, Robert E. Spears, and Justin L. Coleman. High frequency content in nonlinear soil response: A numerical artifact or a reality? *Soil Dynamics and Earthquake Engineering*, 116:185 – 191, 2019.

[VSS12] Emmanuil Veveakis, Jean Sulem, and Ioannis Stefanou. Modeling of fault gouges with Cosserat continuum mechanics: Influence of thermal pressurization and chemical decomposition as coseismic weakening mechanisms. *Journal of Structural Geology*, 38:254–264, 2012.

[VT96] George Z. Voyiadjis and Ganesh Thiagarajan. A cyclic anisotropic–plasticity model for metal matrix composites. *International Journal of Plasticity*, 12(2):69–91, 1996.

[VTC$^+$25] Alberto Hurtado Valdss, Eduardo Torres, Guido Camata, Massimo Petracca, Jorge G. F. Crempien, and Jose A. Abell. Impact of soil-structure interaction modeling simplifications and structural nonlinearity on uncertainty in EDPs: A case study on an existing RC building in Santiago. *Earthquake Engineering & Structural Dynamics*, n/a(n/a), 2025.

[VTS13] Michalis F. Vassiliou, Anastasios Tsiavos, and Božidar Stojadinović. Dynamics of inelastic base-isolated structures subjected to analytical pulse ground motions. *EARTHQUAKE ENGINEERING & STRUCTURAL DYNAMICS*, 42:2043–2060, 2013.

[VTS15] Michalis F. Vassiliou, Rico Truniger, and Božidar Stojadinović. An analytical model of a deformable cantilever structure rocking on a rigid surface: development and verification. *Earthquake Engineering & Structural Dynamics*, 44(15):2775–2794, 2015.

[vTSC13] G. Žerovnik, A. Trkov, D.L. Smith, and R. Capote. Transformation of correlation coefficients between normal and lognormal distribution and implications for nuclear applications. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 727:33–39, 2013.

[VV88] I. Vardoulakis and Ch. Vrettos. Dispersion law of Rayleigh–type wave in a compressible Gibson half–space. *International Journal for Numerical and Analytical Methods in Geomechanics*, 12:639–655, 1988.

[VVDT07] Emmanuil Veveakis, Ioannis Vardoulakis, and Giulio Di Toro. Thermoporomechanics of creeping landslides: The 1963 vaiont slide, northern Italy. *Journal of Geophysical Research: Earth Surface*, 112(F3), 2007.

[vW96] P-A von Wolffersdorff. A hypoplastic relation for granular materials with a predefined limit state surface. *Mechanics of Cohesive-frictional Materials*, 1(3):251–271, 1996.

[VY97] Anestis S. Veletsos and Adel H. Younan. Dynamic response of cantilever retaining walls. *Journal of Geotechnical and Geoenvironmental Engineering*, 123(2):161–172, 1997.

[VZM$^+$22] Nikolaos N. Vlassis, Puhan Zhao, Ran Ma, Tommy Sewell, and WaiChing Sun. Molecular dynamics inferred transfer learning models for finite-strain hyperelasticity of monoclinic crystals: Sobolev training and validations against physical constraints. *International Journal for Numerical Methods in Engineering*, 123(17):3922–3949, 2022.

[VZSM$^+$16] Field G Van Zee, Tyler M Smith, Bryan Marker, Tze Meng Low, Robert A van de Geijn, Francisco D Igual, Mikhail Smelyanskiy, Xianyi Zhang, Michael Kistler, Vernon Austel, et al. The blis framework: Experiments in portability. *ACM Transactions on Mathematical Software (TOMS)*, 42(2):12, 2016.

[VZvdG12] Field G Van Zee and Robert A van de Geijn. BLIS: A framework for generating BLAS-like libraries. *The University of Texas at Austin, Department of Computer Sciences, Technical Report TR-12-30*, 2012.

[WA90] Gustavo Weber and Lalit Anand. Finite deformation constitutive equations and a time integration procedure for isotropic, hyperelastic – viscoplastic solids. *Computer Methods In Applied Mechanics and Engineering*, 79:173–202, 1990.

[WA94] John R. Williams and Kevin Amaratunga. Introduction to wavelets in engineering. *International Journal for Numerical Methods in Engineering*, 37:2365–2388, 1994.

[Wag87] E. Wagenknecht. Response of a NPP reactor building under seismic action with regard to different soil properties. *Nuclear Engineering and Design*, 104(2):187 – 195, 1987.

[Wal08] Edward Walker. Benchmarking Amazon EC2 for high-performance scientific computing. *;login: the USENIX Magazine*, 33(5):18–23, 2008.

[Wal09] Melanie Anne Walling. *Non-Ergodic Probabilistic Seismic Hazard Analysis and Spatial Simulation of Variation in Ground Motion*. PhD thesis, University of California at Berkeley, Spring 2009. Under guidance of Professor Norman Abrahamson.

[Wal17] Larry Wall. The perl programming language. http://www.perl.org/, 2017.

[Wal19] Bruce Walsh. Lecture notes in Introduction to Mixed Models, Infinite-dimensional/Function-valued Traits: Covariance Functions and Random Regressions , July 2019. SISG (Module 12) .

[Wan86] Paul S Wang. Finger: A symbolic system for automatic generation of numerical programs in finite element analysis. *Journal of Symbolic Computation*, 2(3):305–316, 1986.

[Wan09] Zhenming Wang. comment on sigma issues insights and challenges. *International journal for Numerical and Analytical Methods in Geomechanics*, 80(3):491–493, May/June 2009.

[WAR95] John R. Williams, Kevin Amaratunga, and Nabha Rege. An element – free boundary point method based on wavelet radial basis functions. In Stein Sture, editor, *Proceedings of 10th Conference*, pages 1050–1053. Engineering Mechanics Division of the American Society of Civil Engineers, May 1995.

[War03] M. P. Ward. Language oriented programming. Computer Science Department, Science Labs, South Rd, Durham, DH1 3LE, `http://www.cse.dmu.ac.uk/ mward/martin/papers/middle-out-t.pdf`,, January 17 2003.

[Was82] Kyuishiro Washizu. *Variational Methods in Elasticity and Plasticity*. K. Washizu, 3rd. edition edition, 1982.

[Wat90] Osamu Watanabe. Variational principles of elastoplasticity in finite deformation. *International Journal of Japan Society of Mechanical Engineering*, 33(4):480–489, 1990. Series I.

[Wau09] Jonathan David Waugh. *Nonlinear analysis of T-shaped concrete walls subjected to multi-directional displacements*. Iowa State University, 2009.

[WB69] Robert V. Whitman and William A. Bailey. Use of computers for slope stability analysis. In *Stability and Performance of Slopes and Embankments*, pages 519–548. Soil Mechanics and Foundations Division, American Society of Civil Engineering, August 1969.

[WB87] F. J. Wall and C. G. Bucher. Sensitivity of expected exceedance rate of SDOF- system response to statistical uncertainties of loading and system parameters. *Probabilistic Engineering Mechanics*, 2(3), 1987.

[WBA20] Moritz Wenzel, Oreste S. Bursi, and Ioannis Antoniadis. Optimal finite locally resonant metafoundations enhanced with nonlinear negative stiffness elements for seismic protection of large storage tanks. *Journal of Sound and Vibration*, 483:115488, 2020.

[WBH12] Ya Wang, Zhan Yu Bu, and Lei Hu. Seismic behavior of precast segmental bridge columns with carbon fibre reinforcement as energy dissipation bars. In *Applied Mechanics and Materials*, volume 157, pages 1148–1152. Trans Tech Publ, 2012.

[WBL22] Mengtao Wu, Zhenning Ba, and Jianwen Liang. A procedure for 3d simulation of seismic wave propagation considering source-path-site effects: Theory, verification and application. *Earthquake Engineering & Structural Dynamics*, 51(12):2925–2955, 2022.

[WBM^{+}15] X Wei, J Braverman, M Miranda, ME Rosario, and CJ Costantino. Depth-dependent vertical-to-horizontal (v/h) ratios of free-field ground motion response spectra for deeply embedded nuclear structures. Technical report, Brookhaven National Lab.(BNL), Upton, NY (United States), 2015.

[WC06] Wills and Clahan. Developing a map of geologically defined site-condition categories for california. *Bulletin of the Seismological Society of America*, 96(4), 2006.

[WCV11] Stéfan van der Walt, S Chris Colbert, and Gaël Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

[WCX19] Xiaowei Wang, Juntao Chen, and Ming Xiao. Seismic responses of an underground powerhouse structure subjected to oblique incidence SV and P waves. *Soil Dynamics and Earthquake Engineering*, 119:130 – 143, 2019.

[WD23] Zachary J. Westgate and Jason T. DeJong. Role of initial state, material properties, and confinement condition on local and global soil-structure interface behavior during cyclic shear. *Journal of Geotechnical and Geoenvironmental Engineering*, 149(10):04023088, 2023.

[WdS12] Bernard R Wair, Jason T de Jong, and Thomas Shantz. Guidelines for estimation of shear wave velocity profiles. Technical Report 8, PEER, 2012. 95 pages.

[Wea98] Craig S. Weaver. No longer clueless in seattle: Current assessment of earthquake hazards. In Panos Dakoulas, Mishac Yegian, and Robert D. Holtz, editors, *Proceedings of a Specialty Conference: Geotechnical Earthquake Engineering and Soil Dynamics III*, Geotechnical Special Publication No. 75, pages 39–53. ASCE, August 1998. 1998.

[Web04] Steven Weber. *The Success of Open Source*. Harvard University Press, 2004. ISBN 0-674-01292-5.

[WEHM20] S. Wu, E. Eckert, J. Huang, and D. McCallen. Evaluation of the domain reduction method applied to broad-band, near-fault earthquake ground motions with inter-code comparisons. Technical Report CCEER 20-04, Center for Civil Engineering Earthquake Research, University of Nevada, Reno,, 2020.

[Wei89] Stefan Weihe. Implicit integration schemes for multi surface yield criteria subjected to hardening / softening behavior. Master's thesis, University of Colorado at Boulder, 1989.

[Wel99] Brent Welch. *Pratical programming in Tcl and Tk*. Pearson Education, 1999.

[Wem71] Gerald A. Wempner. Discrete approximations related to nonlinear theories of solids. *International Journal for Solids and Structures*, 7:1581–1599, 1971.

[Wes33] H.M. Westergaard. Water pressure on dams during earthquakes. *Transactions of the American Society of Civil Engineers*, 98(2):418–433, January 1933.

[WF70] W.R. Wawersik and C. Fairhurst. A study of brittle rock fracture in laboratory compression experiments. *International Journal of Rock Mechanics and Mining Sciences - Geomechanics Abstracts*, 7(5):561 – 575, 1970.

[WF03] Y Wang and D G Freeland. Towards a better understanding of the role of the contractile skin. In *Proceedings of the 2nd Asian Conference on Unsaturated Soils, April*, pages 15–17, 2003.

[WFRS00] D.R. Woods, R.M. Felder, A. Rugarcia, and J.E. Stice. The future of engineering education. iii. developing critical skills. *Chemical Engineering Education*, 34(2):108–117, 2000.

[WG99] Richard G. Wan and Pei J. Guo. Description of macroscopic stress-dilatancy in a granular assembly with microstructure. In N.P. Jones and Roger G. Ghanem, editors, *the Proceedings of the 13th ASCE Engineering Mechanics Division Specialty Conference*, Johns Hopkins University, Baltimore, June 13-16 1999. CD-ROM.

[WG21] Zhiheng Wang and Roger Ghanem. An extended polynomial chaos expansion for pdf characterization and variation with aleatory and epistemic uncertainties. *Computer Methods in Applied Mechanics and Engineering*, 382:113854, 2021.

[WGG20] Shengzhe Wang, Maria Garlock, and Branko Glišić. Hydrostatic response of deployable hyperbolic-paraboloid umbrellas as coastal armor. *ASCE Journal of Structural Engineering*, 146(6):04020096, 2020.

[WGL$^+$98] Michael S. Warren, Timothy C. Germann, Peter S. Lomdahl, David M. Beazley, and John K. Salmon. Avalon: an Alpha/Linux cluster achieves 10 Gflops for $ 150K. *Gordon Bell Price/Performance Prize Entry*, 1998.

[WGS$^+$] Henry Weller, Chris Greenshields, Bruno Santos, Will Bainbridge, Mattijs Janssens, and OTHERS. *OpenFOAM*. The OpenFOAM Foundation. https://openfoam.org/.

[WGU99] Akihiko Wakai, Shingo Gose, and Keizo Ugai. 3-d elasto-plastic finite element analysis of pile foundations subjected to lateral loading. *Soil and Foundations*, 39(1):97–111, Feb. 1999.

[WGY$^+$21] Weifeng Wu, Shiping Ge, Yong Yuan, Wenqi Ding, and Ioannis Anastasopoulos. Seismic response of a cross interchange metro station in soft soil: Physical and numerical modeling. *Earthquake Engineering & Structural Dynamics*, 2021.

[WGZZ18] Yongxin Wu, Yufeng Gao, Ning Zhang, and Fei Zhang. Simulation of spatially varying non-Gaussian and nonstationary seismic ground motions by the spectral representation method. *Journal of Engineering Mechanics*, 144(1):04017143, 2018.

[WH88] M. De Wet and F. Hugo. Evaluating the design of an earth dam by finite element analysis. *International Journal for Numerical and Analytical Methods in Geomechanics*, 12:573–578, 1988.

[WH89] J Wolf and William Hall. Soil-structure-interaction analysis in time domain. *Nuclear Engineering and Design*, 111(3):381 – 393, 1989.

[Whi00] Robert V. Whitman. Organizing and evaluating uncertainity in geotechnical engineering. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 126(7):583–593, July 2000.

[WHL11] F-C Wang, M-F Hong, and T-C Lin. Designing and testing a hydraulic inerter. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 225(1):66–72, 2011.

[WHS13] John A Wise, V David Hopkin, and Paul Stager. *Verification and validation of complex systems: Human factors issues*, volume 110. Springer Science & Business Media, 2013.

[WHZ14] Xi-Bin Li Wan-Huan Zhou, Lin-Shuang Zhao. A simple analytical solution to one-dimensional consolidation for unsaturated soils. *International journal for Numerical and Analytical Methods in Geomechanics*, 38(8):794–810, August 2014.

[Wik06] Wikipedia Community. Wikipedia, a free-content encyclopedia that anyone can edit. http://wikipedia.org/, 2006.

[Wil63] Edward L. Wilson. *Finite Element Analysis of Two-Dimensional Structures*. PhD thesis, University of California, Berkeley, June 1963.

[Wil69] Kaspar Jodok Willam. *Finite Element Analysis of Cellular Structures*. PhD thesis, University of California at Berkeley, December 1969.

[Wil78] K. J. Willam. Numerical solution of inelastic rate processes. *Computers & Structures*, 8:511–531, 1978.

[Wil89] J. Kaspar Willam. Recent issues in computational plasticity. In *COMPLAS*, pages 1353–1377, 1989.

[Wil93] Kaspar Willam. Plasticity and elastic degradation. Lecture Notes at CU Boulder, august - decembar 1993.

[Wil98] Daniel Wilson. *Soil–Pile–Superstructure interaction at soft and liquefying soils sites*. PhD thesis, University of California, Davis, Caifornia, 1998.

[Wil02] Edward L. Wilson. *Three-Dimensional Static and Dynamic Analysis of Structures A Physical Approach With Emphasis on Earthquake Engineering*. Computers and Structures Inc., third edition, 2002. ISBN 0-923907-00-9.

[Wir71] Niklaus Wirth. Program development by stepwise refinement. *Communications of the ACM*, 14(4):221–227, April 1971.

[Wir95a] Niklaus Wirth. A plea for lean software. *Computer*, 28(2):64–68, 1995.

[Wir95b] Niklaus Wirth. A plea for lean software. *Computer*, 28:64–68, 1995.

[Wit14] W. Wittke. *Rock mechanics based on an Anisotropic Jointed Rock Model AJRM*. Ernst & Sohn, Berlin, 2014.

[WJ98] K. P. Wang and J. C. Bruch Jr. Finite element adaptive mesh analysis using a cluster of workstations. *International Journal for Numerical Methods in Fluids*, 27:179–192, 1998.

[WJ18] Hexiang Wang and Boris Jeremić. Beneficial and detrimental mechansims for simplification of 3D motion field. *to be submitted to: Soil Dynamics and Earthquake Engineering*, 2018. In review.

[WKC$^+$98] Shaomin Wang, Bruce L Kutter, M Jacob Chacko, Daniel W Wilson, Ross W Boulanger, and Abbas Abghari. Nonlinear seismic soil-pile structure interaction. *Earthquake spectra*, 14(2):377–396, 1998.

[WKR$^+$04] Jiaer Wu, Annie Kammerer, Michael Riemer, Raymond Seed, and Juan Pestana. Laboratory study of liquefaction triggering criteria. In *13th World Conference on Earthquake Engineering*, number 2580 in ., page 14pp, Vancouver, B.C., Canada, August 2004.

[WL85] L. H. Wong and J. Enrique Luco. Tables of impedance functions for square foundations on layered media. *International Journal of Soil Dynamics and Earthquake Engineering*, 4(2):64–81, 1985.

[WLH04] Zhongqi Wang, Yong Lu, and Hong Hao. Numerical investigation of effects of water saturation on blast wave propagation in soil mass. *ASCE Journal of Engineering Mechanics*, 130(5):551–561, 2004.

[WLLZ22] Pan Wang, Chunyu Li, Fuchao Liu, and Hanyuan Zhou. Global sensitivity analysis of failure probability of structures with uncertainties of random variable and their distribution parameters. *Engineering with Computers*, 38(5):4367–4385, 2022.

[WLN24] Xiaoxiao Wang, Yang Liu, and François Nicot. Energy processes and phase transition in granular assemblies. *International Journal of Solids and Structures*, 289:112634, 2024.

[WLWT77] Stuart D Werner, LC Lee, HL Wong, and MD Trifunac. An evaluation of the effects of traveling seismic waves on the three-dimensional response of structures. Technical Report NASA STI No. R-7720-4514, NASA, Agbabian Associates, El Segundo, CA, 1977.

[WM06] Changfu Wei and Kanthasamy K. Muraleetharan. Acoustical characterization of fluid-saturated porous media with local heterogeneity: Theory and application. *International Journal of Solids and Structures*, 43(5):982–1008, 2006.

[WM16] Chris Wise and Ed McCann. MAGIC, MArginal Gains In Cconstruction: UK industry strategy to deliver more value with less cost. Draft for discussion, Expedition Engineering, London, July 2016.

[WM18] Chong Wang and Hermann G. Matthies. Evidence theory-based reliability optimization design using polynomial chaos expansion. *Computer Methods in Applied Mechanics and Engineering*, 341:640–657, 2018.

[WM19] Chong Wang and Hermann G. Matthies. Epistemic uncertainty-based reliability analysis for engineering system with hybrid evidence and fuzzy variables. *Computer Methods in Applied Mechanics and Engineering*, 355:438 – 455, 2019.

[WMAM24] Kenneth K. Walsh, Claudia Marin-Artieda, and Keanu McElroy. Adaptive passive seismic isolation system for mitigating the acceleration response of floor-mounted equipment. *Journal of Structural Engineering*, 150(1):04023199, 2024.

[WMXL18] Chong Wang, Hermann G. Matthies, Menghui Xu, and Yunlong Li. Epistemic uncertainty-based model validation via interval propagation and parameter calibration. *Computer Methods in Applied Mechanics and Engineering*, 342:161 – 176, 2018.

[WN96] B. C. Watson and A. K. Noor. Nonlinear structural analysis on distributed-memory computers. *Computers & Structures*, 58(2):233–247, January 1996.

[WNS20] Meng Wang, Satish Nagarajaiah, and Fei-Fei Sun. Dynamic characteristics and responses of damped outrigger tall buildings using negative stiffness. *Journal of Structural Engineering*, 146(12):04020273, 2020.

[WO93] Phillip C. Wankat and Frank S. Oreovicz. *Teaching engineering*. New York : McGraw-Hill, 1993.

[Wol85] JP Wolf. *Dynamic Soil-Structure Interaction*. Prentice-Hall Inc, New Jersey, 1985.

[Wol88] J. P. Wolf. *Soil-Structure-Interaction Analysis in Time Domain*. Prentice-Hall, Englewood Cliffs (NJ), 1988.

[Wol89] John P. Wolf. Soil-structure-interaction analysis in time domain. *Nuclear Engineering and Design*, 111(3):381 – 393, 1989.

[Wol91] Stephen Wolfram. *Mathematica, A System for Doing Mathematics by Computer*. Addison – Wesley Publishing Company, second edition, 1991.

[Wol05] Wolfram Research Inc. *Mathematica*. Wolfram Research Inc., Champaign, Illinois, version 5.0 edition, 2005.

[Won08] KK Wong. Seismic energy dissipation of inelastic structures with tuned mass dampers. *Journal of engineering mechanics*, 134(2):163–172, 2008.

[Woo68] Richard D. Woods. Screening of surface waves in soils. *ASCE Journal of the Soil Mechanic and Foundations Division*, 94(4):951–979, July 1968.

[Woo92] Sharon L. Wood. Seismic response of r/c frames with irregular profiles. *ASCE Journal of Structural Engineering*, 118(2):545–566, February 1992.

[Woo04] David Muir Wood. *Geotechnical Modeling*. Spoon Press, 2004. ISBN 0-415–34304.

[Wor] World Wide Web Consortium. eXtensible Markup Language (XML). `http://www.w3.org/TR/1998/REC-xml-19980210`.

[WOR96] John R. Williams, Rory O'Connor, and Nabha Rege. Discrete element analysis and granular vortex formation. *Electronic Journal of Geotechnical Engineering*, October 1996. http://www.ejge.com/1996/Ppr9603/Ppr9603.zip.

[Wor00] World Wide Web Consortium. XML Schema, w3c working draft. `http://www.w3.org/TR/xmlschema-0/`, February 25th 2000.

[WOW83] John P. Wolf, Pius Obernhuber, and Benedikt Weber. Response of a nuclear power plant on aseismic bearings to horizontally propagating waves. *Earthquake Engineering & Structural Dynamics*, 11(4):483–499, 1983.

[WP03] John P. Wolf and Matthias Preisig. Dynamic stiffness of foundations embedded in layered halfspace based on wave propagation in cones. *Earthquake Engineering and Structural Dynamics*, 32:1075–1098, 2003.

[WPD00] R. Clint Whaley, Antoine Petitet, and Jack J. Dongarra. Automated empirical optimization of software and the ATLAS project. Technical Report UT-CS-00-448, University of Tenessee, September 2000.

[WPJ17] Kohei Watanabe, Federico Pisanò, and Boris Jeremić. Discretization effects in the finite element simulation of seismic waves in elastic and elastic-plastic media. *Engineering with Computers*, 33(3):519–545, Jul 2017.

[WPO96] Z. Wei, D. Perić, and D. R. J. Owen. Consistent linearization for the exact stress update of Prandtl – Reuss non–hardening elastoplastic models. *International Journal for Numerical Methods in Engineering*, 39:1219–1235, 1996.

[WPP04] Bin Wang, Radu Popescu, and Jean H. Prevost. Effects of foundation conditions and partial drainage on cyclic simple shear test results – a numerical study. *International Journal for Numerical and Analytical Methods in Geomechanics*, 28:1057–1082, 2004.

[WR97a] John R. Williams and Nabha Rege. Coherent vortex structures in deforming granular materials. *MECHANICS OF COHESIVE-FRICTIONAL MATERIALS*, 2:223–236, 1997.

[WR97b] John R. Williams and Nabha Rege. The development of circulation cell structures in granular materials undergoing compression. *Powder Technology*, 90(3):187 – 194, 1997.

[WR15] Hua-Ping Wan and Wei-Xin Ren. Parameter selection in finite-element-model updating by global sensitivity analysis using gaussian process metamodel. *Journal of Structural Engineering*, 141(6):04014164, 2015.

[Wri02] Peter Wriggers. *Computational Contact Mechanics*. John Wiley & Sons, 2002.

[WRP98] Thomas J Weaver, Kyle M. Rollins, and Kris T. Peterson. Lateral statnamic load testing and analysis of a pile group. In Panos Dakoulas, Mishac Yegian, and Robert D. Holtz, editors, *Proceedings of a Specialty Conference: Geotechnical Earthwuake Engineering and Soil Dynamics III*, Geotechnical Special Publication No. 75, pages 1319–1330. ASCE, August 1998. 1998.

[WS90] P. Wriggers and J. C. Simo. A general procedure for the direct computation of turning and bifurcation points. *International Journal for Numerical Methods in Engineering*, 30:155–176, 1990.

[WS99] John P. Wolf and Chongmin Song. The guts of dynamic soil–structure interaction. In *International Symposium of Earthquake Engineering*, Budva, Montenegro, Yugoslavia, September 1999.

[WS16] Fangbo Wang and Kallol Sett. Time-domain stochastic finite element simulation of uncertain seismic wave propagation through uncertain heterogeneous solids. *Soil Dynamics and Earthquake Engineering*, 88:369 – 385, 2016.

[WS19] Fangbo Wang and Kallol Sett. Time domain stochastic finite element simulation towards probabilistic seismic soil-structure interaction analysis. *Soil Dynamics and Earthquake Engineering*, 116:460 – 475, 2019.

[WSG91] Sharon L. Wood, Roberto Stark, and Scott A. Greer. Collapse of eight–story RC building during 1985 chile earthquake. *ASCE Journal of Structural Engineering*, 117(2):600–619, February 1991.

[WT74] HL Wong and MD Trifunac. Interaction of a shear wall with the soil for incident plane sh waves: elliptical rigid foundation. *Bulletin of the Seismological Society of America*, 64(6):1825–1842, 1974.

[WT75] HL Wong and MD Trifunac. Two-dimensional, antiplane, building-soil-building interaction for two or more buildings and for incident planet SH waves. *Bulletin of the Seismological Society of America*, 65(6):1863–1885, 1975.

[WTJF98] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics*, 12(6):620–631, Nov/Dec 1998.

[WTL88] H. L. Wong, M. D. Trifunac, and J. E. Luco. A comparison of soil-structure interaction calculations with results of full-scale forced vibration tests. *Soil Dynamics and Earthquake Engineering*, 7(1):22–31, July 1988.

[WTM04] W.Ehlers, T.Graf, and M.Ammann. Deformation and localization analysis of partially saturated soil. *Computational Methods in Applied Mechanics and Engineering*, 193:2885–2910, 2004.

[Wu 66] Te Wu Tai. The effect of inclusion shape on the elastic moduli of a two - phase material. *International Journal of Solids and Structures*, 2:1–8, 1966.

[Wu92] Z. Wu. Hermite-birkhoff interpolation of scattered data by radial basis functions. *Approx. Theory Appl.*, 8:1–10, 1992.

[WU96] Andrew J. Whittle and Boonchai Ukritchon. Applicattion of numerical limit analysis for shallow foundations on clay. In Y. K. Lin and T. C. Su, editors, *Proceedings of 11th Conference*, pages 132–135. Engineering Mechanics Division of the American Society of Civil Engineers, May 1996.

[WVS90] Peter Wriggers, T Vu Van, and Erwin Stein. Finite element formulation of large deformation impact-contact problems with friction. *Computers & Structures*, 37(3):319–331, 1990.

[WW74] K. J. Willam and E. P. Warnke. Constitutive model for the triaxial behaviour of concrete. In *Proceedings IABSE Seminar on Concrete Bergamo*. ISMES, 1974.

[WWB$^+$19] Hexiang Wang, Fangbo Wang, Jeff Bayless, Han Yang, Yuan Feng, Norman A. Abrahamson, and Boris Jeremić. Time domain intrusive stochastic seismic risk analysis using ground motion prediction equations of Fourier amplitude spectra. *to be submitted to: Earthquake Spectra*, 2019.

[WWB$^+$20] Hexiang Wang, Fangbo Wang, Jeff Bayless, Han Yang, Marco Baglio, Norman A. Abrahamson, and Boris Jeremić. Time domain intrusive stochastic seismic risk analysis using ground motion prediction equations of fourier amplitude spectra. *Soil Dynamics and Earthquake Engineering*, in Review, 2020.

[WWJ$^+$11] Qiang Wang, Jin-Ting Wang, Feng Jin, Fu-Dong Chi, and Chu-Han Zhang. Real-time dynamic hybrid testing for soil-structure interaction analysis. *Soil Dynamics and Earthquake Engineering*, 31:1690–1702, 2011.

[WWY$^+$19] Hexiang Wang, Fangbo Wang, Han Yang, Yuan Feng, Jeff Bayless, Norman A. Abrahamson, and Boris Jeremić. Time domain seismic risk analysis framework for nuclear installations. In *Proceedings of the 25th International Conference on Structural Mechanics in Reactor Technology (SMiRT 25), Charlotte, NC, USA*, page 8, 2019.

[WWY$^+$20a] Fangbo Wang, Hexiang Wang, Han Yang, Yuan Feng, and Boris Jeremić. A modular methodology for time-domain intrusive stochastic seismic wave propagation. *Computers and Geotechnics*, (139), 2020.

[WWY$^+$20b] Hexiang Wang, Fangbo Wang, Han Yang, Yuan Feng, Jeff Bayless, Norman A. Abrahamson, and Boris Jeremić. Time domain intrusive probabilistic seismic risk analysis of nonlinear shear frame structure. *Soil Dynamics and Earthquake Engineering*, 136:106201, 2020.

[WWY$^+$20c] Hexiang Wang, Fangbo Wang, Han Yang, Yuan Feng, Jeff Bayless, Norman A. Abrahamson, and Boris Jeremić. Time domain intrusive probabilistic seismic risk analysis of nonlinear shear frame structure. *Soil Dynamics and Earthquake Engineering*, 136:106201, 2020.

[WWY$^+$20d] Hexiang Wang, Fangbo Wang, Han Yang, Yuan Feng, Jeff Bayless, Norman A. Abrahamson, and Boris Jeremić. Time domain intrusive probabilistic seismic risk analysis of nonlinear shear frame structure. *Soil Dynamics and Earthquake Engineering*, 136:106201, 2020.

[WWY$^+$25] Hexiang Wang, Fangbo Wang, Han Yang, Katarzyna Staszewska, and Boris Jeremić. Sobol' sensitivity analysis of a 1D stochastic elasto-plastic seismic wave propagation. *Soil Dynamics and Earthquake Engineering*, 191:109283, 2025.

[WWYJ21] Hexiang Wang, Fangbo Wang, Han Yang, and Boris Jeremić. Site response analysis: Uncertain motions propagating through uncertain elastoplastic soil. *Nuclear Engineering and Design*, 2021. In Review.

[WWZL21] Jingjing Wang, Bin Wang, Chao Zhang, and Zhibin Liu. Effectiveness and robustness of an asymmetric nonlinear energy sink-inerter for dynamic response mitigation. *Earthquake Engineering & Structural Dynamics*, Early View Alert(n/a), 2021.

[WYF$^+$20] Hexiang Wang, Han Yang, Yuan Feng, Fangbo Wang, and Boris Jeremić. Wave potential-domain reduction method for 3D earthquake soil structure interaction. *Soil Dynamics and Earthquake Engineering*, 2020. In review.

[WYFJ21] Hexiang Wang, Han Yang, Yuan Feng, and Boris Jeremić. Modeling and simulation of earthquake soil structure interaction excited by inclined seismic waves. *Soil Dynamics and Earthquake Engineering*, 146:106720, 2021.

[WYGJ24] Hai-Lin Wang, Zhen-Yu Yin, Xiao-Qiang Gu, and Yin-Fu Jin. Evaluation of soil-structure interface models considering cyclic loading effect. *International Journal for Numerical and Analytical Methods in Geomechanics*, n/a(n/a), 2024.

[WYL$^+$24] Juntian Wang, Yunming Yang, Xiang Li, Enlong Liu, and Hai-Sui Yu. Energy dissipation in monotonic and cyclic simple shear tests considering initial static shear stress. *Soil Dynamics and Earthquake Engineering*, 180:108576, 2024.

[WYS$^+$17] Hexiang Wang, Han Yang, Sumeet K. Sinha, Yuan Feng, Chao Luo, David B. McCallen, and Boris Jeremić. 3-D non-linear earthquake soil-structure interaction modeling of embedded small modular reactor (SMR). In *Proceedings of the 24th International Conference on Structural Mechanics in Reactor Technology (SMiRT 24)*, Busan, South Korea, August 20-25 2017.

[WYWJ22] Hexiang Wang, Han Yang, Fangbo Wang, and Boris Jeremić. Time domain, intrusive probabilistic seismic risk analysis of nonlinear earthquake soil structure interaction systems. In *Transactions, SMiRT-26*. SMiRT, July 2022.

[WZE$^+$19] Steven Wehmeyer, Frank W. Zok, Christopher Eberl, Peter Gumbsch, Noy Cohen, Robert M. McMeeking, and Matthew R. Begley. Post-buckling and dynamic response of angled struts in elastic lattices. *Journal of the Mechanics and Physics of Solids*, 133:103693, 2019.

[WZG09] X.Y. Wei, Z.Y. Zhao, and J. Gu. Numerical simulations of rock mass damage induced by underground explosion. *International Journal of Rock Mechanics and Mining Sciences*, 46(7):1206–1213, October 2009.

[WZKH97] G.W. Wei, D.S. Zhang, D.J. Kouri, and D.K. Hoffman. Distributed approximating functional approach to the Fokker–Planck equation: Time propagation. *The Journal of Chemical Physics*, 107(8):3239–3246, August 1997.

[WZW14] Rui Wang, Jian-Min Zhang, and Gang Wang. A unified plasticity model for large post-liquefaction shear deformation of sand. *Computers and Geotechnics*, 59:54–66, 2014.

[XB10] Liang Xue and Ted Belytschko. Fast methods for determining instabilities of elastic-plastic damage models through closed-form expressions. *International Journal for Numerical Methods in Engineering*, 84(12):1490–1518, December 2010.

[XBGW02] Jifeng Xu, Jacobo Bielak, Omar Ghattas, and Jianlin Wang. Three-dimensioanl seismic ground motion modeling in inelastic basins,. *Physics of the Earth and Planetary Interiors*, To Appear, 2002.

[XBGW03] Jifeng Xu, Jacobo Bielak, Omar Ghattas, and Jianlin Wang. Three-dimensional nonlinear seismic ground motion modeling in basins. *Physics of the earth and planetary interiors*, 137(1-4):81–95, 2003.

[XCH03] J. Xu, C. Costantino, and C. Hofmayer. Collaborative study of nupec seismic field test data for NPP structures. Technical Report NUREG/CR-6822, U.S. Nuclear Regulatory Commission, Washington, DC, June 2003. Also BNL-NUREG-71355-2003.

[XCHG06] J. Xu, C. Costantino, C. Hofmayer, and H. Graves. Finite element models for computing seismic induced soil pressures on deeply embedded nuclear power plant structures. *ASME Pressure Vessels and Piping Division Conference, Vancouver, BC, Canada, BNL NUREG-76748-2006-CP*, July 2006.

[Xen98a] Christos A. Xenophontos. Finite element computations for the Reissner–Mindlin plate model. *Communications in Numerical Methods in Engineering*, 14:1119–1131, 1998.

[Xen98b] Christos A. Xenophontos. The *hp* finite element method for singularly perturbed problems in smooth domains. *Mathematical Models and Methods in Applied Sciences*, 8(2):299–326, 1998.

[Xen99] Christos A. Xenophontos. The *hp* finite element method for singularly perturbed problems in nonsmooth domains. *Numerical Methods for PDEs*, 15(1):63–90, 1999.

[XF18] Jun Xu and De-Cheng Feng. Seismic response analysis of nonlinear structures with uncertain parameters under stochastic ground motions. *Soil Dynamics and Earthquake Engineering*, 111:149–159, 2018.

[XF19] Jun Xu and De-Cheng Feng. Stochastic dynamic response analysis and reliability assessment of non-linear structures under fully non-stationary ground motions. *Structural Safety*, 79:94–106, 2019.

[XGB05] X.F. Xu and L. Graham-Brady. A stochastic computational method for evaluation of global and local behavior of random elastic media. *Computer Methods in Applied Mechanics and Engineering*, 194(42-44):4362–4385, 2005.

[XH05] Dongbin Xiu and J. S. Hesthaven. High-order collocation methods for differential equations with random inputs. *SIAM Journal on Scientific Computing*, 27(3):1118–1139, 2005.

[XHZG06] H. Xia, Y. Han, N. Zhang, and W. Guo. Dynamic analysis of train–bridge system subjected to non–uniform seismic excitations. *Earthquake Engineering & Structural Dynamics*, 35:1563–1579, 2006.

[Xiu08] D. Xiu. Fast numerical methods for stochastic computations. *Commun. Comput. Physics*, 5(2-4):242–272, 2008.

[Xiu10] Dongbin Xiu. *Numerical Methods for Stochastic Computations*. Princeton University Press, 2010.

[XJ99] Christos Xenophontos and Boris Jeremić. *p*–version finite elements in elasto–plasticity. In *Proceedings of the* Fourth International Congress of Industrial and Applied Mathematics, Edinburgh, Scotland, 5-9 July 1999. SIAM.

[XJJP01] Jianxin Xiong, Jeremy Johnson, Robert Johnson, and David Padua. Spl: A language and compiler for dsp algorithms. In *ACM SIGPLAN Notices*, volume 36 / 5, pages 298–308. ACM, 2001.

[XK02] D. Xiu and G. E. Karniadakis. The wiener–askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644, 2002.

[XK03] Dongbin Xiu and George Em Karniadakis. Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of computational physics*, 187(1):137–167, 2003.

[XK04] Dongbin Xiu and George E. Karniadakis. Supersensitivity due to uncertain boundary conditions. *International Journal for Numerical Methods in Engineering*, 61(12):2114–2138, 2004.

[XMCH06] J. Xu, C. Miller, C. Costantino, and C. Hofmayer. Assessment of seismic analysis methodologies for deeply embedded nuclear power plant structures. Technical Report NUREG/CR-6896, BNL-NUREG-75410-2006, Brookhaven National Laboratory, 2006.

[XNCG08] J. Xu, J. Nie, C. Costantino, and H. Graves. Correlation analysis of jnes seismic wall pressure data for abwr model structures. Technical Report NUREG/CR-6957, US. NRC, 2008.

[XRK04] K. Xia, A.J. Rosakis, and H. Kanamori. Laboratory earthquakes: The sub-rayleigh-to-supershear rupture transition. *Science*, 303(5665):1859–1861, 2004.

[XRKR05] K. Xia, A.J. Rosakis, H. Kanamori, and J.R Rice. Laboratory earthquakes along inhomogeneous faults: Directionality and supershear. *Science*, 303(5722):681–684, 2005.

[XS09] Jim Xu and Sujit Samaddar. Case study: Effect of soil-structure interaction and ground motion incoherency on nuclear power plant structures. In *ASME 2009 Pressure Vessels and Piping Conference*, pages 369–377. American Society of Mechanical Engineers, 2009.

[XS10] Tao Xing and Frederick Stern. Factors of safety for Richardson extrapolation. *Journal of Fluids Engineering*, 132(6):061403, 2010.

[XUU24] Jiawei Xu, Ryosuke Uzuoka, and Kyohei Ueda. Coupled finite element analysis of the dynamics of poroelastic media considering the relative fluid acceleration. *International Journal for Numerical and Analytical Methods in Geomechanics*, 48(14):3561–3592, 2024.

[XW13] Bin Xu and Ron C.K. Wong. Coupled finite-element simulation of injection well testing in unconsolidated oil sands reservoir. *International Journal for Numerical and Analytical Methods in Geomechanics*, 37(18):3131–3149, 2013.

[XXA^{+}20] Wang Xiaohui, Li Xiaojun, Liu Aiwen, He Qiumei, and Hou Chunlin. Seismic analysis of soil-structure system of nuclear power plant on non-rock site via shaking table test. *Soil Dynamics and Earthquake Engineering*, 136:1–11, 2020.

[XZS16] Jun Xu, Wangxi Zhang, and Rui Sun. Efficient reliability assessment of structural dynamic systems with unequal weighted quasi-monte carlo simulation. *Computers & Structures*, 175:37–51, 2016.

[XZYG25] Hao-Ruo Xu, Ning Zhang, Zhen-Yu Yin, and Pierre Guy Atangana Njock. GeoLLM: A specialized large language model framework for intelligent geotechnical design. *Computers and Geotechnics*, 177:106849, 2025.

[YA92] Y. Yu and K. Axelsson. A plasticity model for silt. *Numerical Models in Geomechanics*, pages 37–45, 1992.

[YA03] M. Yao and A. Anandarajah. Three–dimensional discrete element method of analysis of clays. *ASCE Journal of Engineering Mechanics*, 129(6):585–596, June 2003.

[YA15] Emrah Yenier and Gail M Atkinson. An equivalent point-source model for stochastic simulation of earthquake ground motions in california. *Bulletin of the Seismological Society of America*, 105(3):1435–1455, 2015.

[YAF04] Junji Yoshida, Masato Abe, and Yozo Fujino. Constitutive model of high-damping rubber materials. *ASCE Journal of Engineering Mechanics*, 130(2):129–141, February 2004.

[YAN98] Navid Yazdi, Farrokh Ayazi, and Khalil Najafi. Micromachined inertial sensors. In Kensall D. Wise, editor, *Proceedings or the IEEE, Special Issue: Integrated Sensors, Microactuators & Microsystems (MEMS)*, pages 1640–1659, August 1998.

[Yan02a] Zhaohui Yang. *Development of Geotechnical Capabilities into OpenSees Platform and their Applications in Soil-Foundation-Structure Interaction Analyses*. PhD thesis, University of California, Davis, Davis, CA, September 2002.

[Yan02b] Zhaohui Yang. *Three Dimensional Nonlinear Finite Element Analysis of Soil–Foundation–Structure Interaction*. PhD thesis, University of California at Davis, Davis, California, September 2002.

[Yas94] Mohd Hisham Mohd Yassin. *Nonlinear analysis of prestressed concrete sructures under monotonic and cyclic loads*. PhD thesis, University of California, Berkeley, 1994.

[Yas12] Marh Yashinsky. Lessons for caltrans from the 2011 Great East Japan earthquake and tsunami. In *Proceedings of the International Symposium on Engineering Lessons Learned from the 2011 Great East Japan Earthquake*, Tokyo, Japan, March 1-4 2012.

[YAT^{+}20] T.Y. Yang, Jeremy Atkinson, Lisa Tobber, Dorian P. Tung, and Bob Neville. Seismic design of outrigger systems using equivalent energy design procedure. *The Structural Design of Tall and Special Buildings*, 29(10):e1743, 2020. e1743 TAL-19-0268.R1.

[YAVS22] Qing Yin, Edward Andò, Gioacchino Viggiani, and WaiChing Sun. Freezing-induced stiffness and strength anisotropy in freezing clayey soil: Theory, numerical modeling, and experimental validation. *International Journal for Numerical and Analytical Methods in Geomechanics*, 46(11):2087–2114, 2022.

[YBH03] Chaiki Yoshimura, Jacobo Bielak, and Yoshiaki Hisada. Domain reduction method for three–dimensional earthquake modeling in localized regions. part II: Verification and examples. *Bulletin of the Seismological Society of America*, 93(2):825–840, 2003.

[YBL96] Jerry A. Yamamuro, Paul A. Bopp, and Pould V. Lade. One–dimensional compression of sands at high pressures. *ASCE Journal of Geotechnical Engineering*, 122(2):147–154, Febryary 1996.

[YC85] Robert R Youngs and Kevin J Coppersmith. Implications of fault slip rates and earthquake recurrence models to probabilistic seismic hazard estimates. *Bulletin of the Seismological society of America*, 75(4):939–964, 1985.

[YCK25] Karin L. Yu, Eleni Chatzi, and Georgios Kissas. Grammar-based ordinary differential equation discovery, 2025.

[YCL98] Jerry A. Yamamuro, Kelly M. Covert, and Poul V. Lade. Static and dynamic liquefaction of silty sand. In Poul V. Lade and Jerry A. Yamamuro, editors, *Physics and Mechanics of Soil Liquefaction*, pages 55–65, Baltimore, Maryland, 10-11 September 1998. A. A. Balkema.

[YDC^{+}04] Z. Yang, U. Dutta, M. Celebi, H. Liu, N. Biswas, and T. Kono. Seismic instrumentation of a building/downhole array system and analysis of preliminary results. In *Proceedings of the 17th ASCE Engineering Mechanics Conference*, pages 1–5, Newark, DE, June 2004. Engineering Mechanics Division of the American Society of Civil Engineers. CD-ROM.

[YDX^{+}11] Zhaohui (Joey) Yang, Utpal Dutta, Gang Xu, Kenan Hazirbaba, and Elmer E. Marx. Numerical analysis of permafrost effects on the seismic site response. *Soil Dynamics and Earthquake Engineering*, 31(3):282 – 290, 2011.

[YE03] Zhaohui Yang and Ahmed Elgamal. Application of unconstrained optimization and sensitivity analysis to calibration of a soil constitutive model. *International Journal for Numerical and Analytical Methods in Geomechanics*, 27(15):1277–1297, 2003.

[YEAS04] Zhaohui Yang, Ahmed Elgamal, Korhan Adalier, and Michael Sharp. Earth dam on liquefiable foundations and remediation: Numerical simulation of centrifuge experiments. *ASCE Journal of Engineering Mechanics*, 130(10):1168–1176, October 2004.

[YEP03] Zhaohui Yang, Ahmed Elgamal, and Ender Parra. Computational model for cyclic mobility and associated shear deformation. *Journal of Geotechnical and Geoenvironmental Engineering*, 129(12):1119–1127, 2003.

[YF20] H.D. Young and R.A. Freedman. *University Physics with Modern Physics*. Pearson, 15th edition, 2020.

[YFLY23] Yong Yuan, Zexu Fan, Fang Liu, and Yusheng Yang. A drm-smm framework for wave propagation in layered saturated ground under inclined P1-SV waves. *Computers and Geotechnics*, 162:105658, 2023.

[YFWJ19] Han Yang, Yuan Feng, Hexiang Wang, and Boris Jeremić. Energy dissipation analysis for inelastic reinforced concrete and steel beam-columns. *Engineering Structures*, 197:109431, 2019.

[YHB99] H. S. Yu, L. R. Herrmann, and R. W. Boulanger. Analysis of steady cone penetration in clay. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, In Print, 1999.

[YHF82] Joseph A. Yura, Mikkel A. Hansen, and Karl H. Frank. Bolted splice connections with underdeveloped fillers. *ASCE Journal of the Structural Division*, 108(ST12):2837–2849, 1982.

[YJ02] Zhaohui Yang and Boris Jeremić. Numerical analysis of pile behaviour under lateral loads in layered elastic-plastic soils. *International Journal for Numerical and Analytical Methods in Geomechanics*, 26(14):1385–1406, 2002.

[YJ03] Zhaohui Yang and Boris Jeremić. Numerical study of the effective stiffness for pile groups. *International Journal for Numerical and Analytical Methods in Geomechanics*, 27(15):1255–1276, Dec 2003.

[YJ05a] Zhaohui Yang and Boris Jeremić. Soil layering effects on lateral pile behavior. ASCE Journal of Geotechnical and Geoenvironmental Engineering, 131(6):762–770, June 2005.

[YJ05b] Zhaohui Yang and Boris Jeremić. Study of soil layering effects on lateral loading behavior of piles. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 131(6):762–770, June 2005.

[YJWZ13] Jianhong Ye, Dongsheng Jeng, Ren Wang, and Changqi Zhu. A 3-d semi-coupled numerical model for fluid-structures-seabed-interaction (FSSI-CAS 3D): Model and verification. *Journal of Fluids and Structures*, 40:148–162, 2013.

[YK81] Yoshiaki Yoshimi and Takao Kishida. A ring torsion apparatus for evaluating friction between soil and metal surfaces. *Geotechnical Testing Journal*, 4(4):145–152, 1981.

[Y.K87] Y.K.Chow. 3-dimensional analysis of pile groups. *Journal of Geotechnical engineering*, 113(6):637–651, June 1987.

[YKS09] Louie L. Yaw, Sashi K. Kunnath, and N. Sukumar. Meshfree method for inelastic frame analysis. *ASCE Journal of Structural Engineering*, 135(6):676–684, June 2009.

[YL93a] Jerry A. Yamamuro and Poul V. Lade. Effects of strain rate on instability of granular soils. *Geotechnical Testing Journal GTJODJ*, 16(3):304–313, 1993.

[YL93b] Jerry A. Yamamuro and Pould V. Lade. B-value measurement for granular materials for high confining pressures. *Geotechnical Testing Journal GTJODJ*, 16(2):165–171, 1993.

[YL95] Jerry A. Yamamuro and Poul V. Lade. Strain localization in extension tests on granular materials. *Journal of Engineering Mechanics*, 121(7), July 1995.

[YL96] Jerry A. Yamamuro and Poul V. Lade. Drained sand behavior in axisymmetric tests at high pressures. *ASCE Journal of Geotechnical Engineering*, 122(2):109–119, 1996.

[YL97a] Jerry A. Yamamuro and Poul V. Lade. Behavior and modeling of static liquefaction of silty sands. In S. Pietruszak and G. N. Pande, editors, *Proceedings of the Sixth International Symposium on Numerical Models in Geomechanics, NUMOG VI*, pages 27–32. A. A. Balkema, 1997.

[YL97b] Jerry A. Yamamuro and Poul V. Lade. Effects of nonplastic fines on static liquefaction of sands. *Canadian Geotechnical Journal*, 34:905–917, 1997.

[YL97c] Jerry A. Yamamuro and Poul V. Lade. Experiments and modeling of silty sands susceptible to static liquefaction. *International Journal of Mechanics of Cohesive–Frictional Materials*, 1997. Submitted for publication.

[YL97d] Jerry A. Yamamuro and Poul V. Lade. Instability of granular materials at high pressures. *Soils and Foundations, Japanese Geotechnical Society*, 37(1):41–52, March 1997.

[YL97e] Jerry A. Yamamuro and Poul V. Lade. Prediction of instability conditions for sand. In Publications Committee of XIV ICSMFE, editor, *Proceedings of the Fourtheenth International Conference on Soil Mechanics and Foundations Engineering*, pages 435–438, Hamburg, 1997. A. A. Balkema.

[YL98] Jerry A. Yamamuro and Poul V. Lade. Steady state concepts and static liquefaction of sitly sands. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 124(9):868–877, 1998. Submitted for publication.

[YL04a] J. Yang and X. S. Li. State-dependent strength of sands from the perspective of unified modeling. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 130(2):186–198, 2004.

[YL04b] J.S. Yu and Y.K. Lin. Numerical path integration of a non-homogeneous markov process. *International Journal of Non-Linear Mechanics*, 39(9):1493 – 1500, 2004. Themes in Non-linear Stochastic Dynamics.

[YMH91] N. Yasufuku, H. Murata, and M. Hyodo. Yield characteristics of anisotropically consolidated sand under low and high stresses. *Soils and Foundations*, 31(95-109), 19791.

[YMQB22] T.Y. Yang, Muhib Muazzam, Musab Aied Qissab Al-Janabi, and Svetlana Brzev. Quantification of energy dissipation demand for buckling-restrained braces. *Soil Dynamics and Earthquake Engineering*, 155:107190, 2022.

[YMU20] Shuo Yang, George P. Mavroeidis, and Alper Ucak. Analysis of bridge structures crossing strike-slip fault rupture zones: A simple method for generating across-fault seismic ground motions. *Earthquake Engineering & Structural Dynamics*, 49(13):1281–1307, 2020.

[YNCS16] Jianming Yin, Robert L Nigbor, Qingjun Chen, and Jamison Steidl. Engineering analysis of measured rotational ground motions at GVDA. *Soil Dynamics and Earthquake Engineering*, 87:125–137, 2016.

[YPU97] Yuzhen Yu, Jialiu Pu, and Keizo Ugai. Study of mechanical properties of soil–cement mixture for a cutoff wall. *Soils and Foundations*, 37(4):93–103, 1997.

[YSF$^+$18] Han Yang, Sumeet Kumar Sinha, Yuan Feng, David B McCallen, and Boris Jeremić. Energy dissipation analysis of elastic-plastic materials. *Computer Methods in Applied Mechanics and Engineering*, 331:309–326, 2018.

[YTL18] TY Yang, Dorian P Tung, and Yuanjie Li. Equivalent energy design procedure for earthquake resilient fused structures. *Earthquake Spectra*, 34(2):795–815, 2018.

[YW20] Ching-Ching Yu and Andrew S. Whittaker. Analytical and numerical studies of seismic fluid-structure interaction in liquid-filled vessels. Technical Report MCEER-20-0003, University of Buffalo, MCEER, August 2020.

[YWF$^+$19] Han Yang, Hexiang Wang, Yuan Feng, Fangbo Wang, and Boris Jeremić. Energy dissipation in solids due to material inelasticity, viscous coupling, and algorithmic damping. *ASCE Journal of Engineering Mechanics*, 145(9), 2019.

[YWFJ19] Han Yang, Hexiang Wang, Yuan Feng, and Boris Jeremić. Plastic energy dissipation in pressure-dependent materials. *ASCE Journal of Engineering Mechanics*, 2019. In Print.

[YWJ21] Han Yang, Hexiang Wang, and Boris Jeremić. Numerical modeling and validation of earthquake soil structure interaction: A 12-story building in Ventura, California. *Engineering Structures*, 2021. In Review.

[YWJ22] Han Yang, Hexiang Wang, and Boris Jeremić. Seismic energy flow and balance in earthquake soil structure interaction systems. In *Transactions, SMiRT-26*. SMiRT, July 2022.

[YWJ24] Han Yang, Hexiang Wang, and Boris Jeremić. Energy balance for earthquake soil structure interaction systems, analysis of input and dissipated energy. *Engineering Structures*, 2024. In Print.

[YWJS19] Han Yang, Hexiang Wang, Boris Jeremić, and Jerzy Salamon. Earthquake soil structure interaction analysis of a gravity dam. In G MAZZA, editor, *Proceedings of the 15th International Benchmark Workshop on Numerical Analysis of Dams*. ICOLD, PoLiMi, September 2019.

[YWKT23] Ching-Ching Yu, Andrew S. Whittaker, Benjamin D. Kosbab, and Payman Khalili Tehrani. Earthquake-induced impact of base-isolated buildings: theory, numerical modeling, and design solutions. *Earthquake Engineering & Structural Dynamics*, n/a(n/a), 2023.

[YWLZ19] Yu Yao, Rui Wang, Tianyun Liu, and Jian-Min Zhang. Seismic response of high concrete face rockfill dams subjected to non-uniform input motion. *Acta Geotechnica*, 14(1):83–100, Feb 2019.

[YWWM21] Jie Yuan, Jin-Ting Wang, Dong Wang, and Yong-Zheng Mao. Study on the dynamic interaction between an arch dam and its appurtenant structures. *Soil Dynamics and Earthquake Engineering*, 144:106679, 2021.

[YXQ15] Han Yang, Wenjie Xu, and Zhang Qibin. Macro- and meso-mechanism of strain localization in granular material. *Chinese Journal of Rock Mechanics and Engineering*, 34(8):1692–1701, 2015.

[YXSF16] Han Yang, Wen-Jie Xu, Qi-Cheng Sun, and Yuan Feng. Study on the meso-structure development in direct shear tests of a granular material. *Powder Technology*, 2016.

[YY20] Osman Yuksei and Cetin Yilmaz. Design of s broadband elastic metamaterial via topologically optimized inertial amplification mechanisms. In Manolis Papadrakakis, Michalis Fragiadakis, and Costas Papadimitriou, editors, *EURODYN 2020, XI International Conference on Structural Dynamics*, Athens, Greece, 23-26 November 2020 2020. https://doi.org/10.47964/1120.9337.19454.

[YZBW25] Zhongsheng Yang, Zixiang Zhang, Yu Bao, and Changxi Wang. Three-dimensional macroscopic model for simulating dynamic behavior of elastomeric bearing including rotational damping effects. *Journal of Structural Engineering*, 151(6):04025065, 2025.

[ZAM06] Zhenyu Zhu, Iftekhar Ahmad, and Amir Mirmiran. Fiber element modeling for seismic performance of bridge columns made of concrete-filled frp tubes. *Engineering structures*, 28(14):2023–2035, 2006.

[ZB14] Kaveh Zamani and Fabián A Bombardelli. Analytical solutions of nonlinear and variable-parameter transport equations for verification of numerical solvers. *Environmental Fluid Mechanics*, 14(4):711–742, 2014.

[ZB20] Adam Zsarnóczay and Jack W. Baker. Using model error in response history analysis to evaluate component calibration methods. *Earthquake Engineering & Structural Dynamics*, 49(2):175–193, 2020.

[ZBDN22] Zhibao Zheng, Michael Beer, Hongzhe Dai, and Udo Nackenhorst. A weak-intrusive stochastic finite element method for stochastic structural dynamics analysis. *Computer Methods in Applied Mechanics and Engineering*, 399:115360, 2022.

[Zbi91] H. M. Zbib. On the mechanics of large inelastic deformations: Noncoaxiality, axial effects in torsion and localization. *Acta Mechanica*, 87:179–196, 1991.

[ŽBI23] Alexandr Žák, Michal Beneš, and Tissa H. Illangasekare. Pore-scale model of freezing inception in a porous medium. *Computer Methods in Applied Mechanics and Engineering*, 414:116166, 2023.

[ZBLH97a] S. J. Zhou, D. M. Beazley, P. S. Lomdahl, and B. L. Holian. Large–scale molecular dynamics simulations of three–dimensional ductile failure. *Physical Review Letters*, 78(3):479–482, 1997.

[ZBLH97b] S.J. Zhou, D.M. Beazley, P.S. Lomdahl, and B.L. Holian. Large-scale molecular dynamics simulations of three-dimensional ductile fracture. *Phys. Rev. Lett.*, 78:479–482, 1997.

[ZC67] O.C. Zienkiewicz and Y.K. Cheung. *The finite element method in structural and continuum mechanics: numerical solution of problems in structural and continuum mechanics*. Number v. 1 in European civil engineering series. McGraw-Hill, 1967.

[ZC98] Eric T. Zechlin and Jianzhong Chai. Nonlinear dynamic analysis of large diameter pile foundations for the bay bridge. In Panos Dakoulas, Mishac Yegian, and Robert D. Holtz, editors, *Proceedings of a Specialty Conference: Geotechnical Earthwuake Engineering and Soil Dynamics III*, Geotechnical Special Publication No. 75, pages 1223–1234. ASCE, August 1998. 1998.

[ZCB80] O.C. Zienkiewicz, C. T. Chang, and P. Bettess. Drained, undrained, consolidating and dynamic behaviour assumptions in soils. *Géotechnique*, 30(4):385–395, 1980.

[ZCJ06] Li Zhao, Po Chen, and Thomas H Jordan. Strain green's tensors, reciprocity, and their applications to seismic source and structure studies. *Bulletin of the Seismological Society of America*, 96(5):1753–1763, 2006.

[ZCMF23] Ningning Zhang, Yuyan Chen, Alejandro Martinez, and Raul Fuentes. A bioinspired self-burrowing probe in shallow granular materials. *Journal of Geotechnical and Geoenvironmental Engineering*, 149(9):04023073, 2023.

[ZCP^{+}90] O.C. Zienkiewicz, A.H.C. Chan, M. Pastor, D. K. Paul, and T. Shiomi. Static and dynamic behaviour of soils: A rational approach to quantitative solutions. I. fully saturated problems. *Procedings of Royal Society London*, 429:285–309, 1990.

[ZCP^{+}99a] O. C. Zienkiewicz, A. H. C. Chan, M. Pastor, B. A. Schrefler, and T. Shiomi. *Computational Geomechanics with Special Reference to Earthquake Engineering*. John Wiley and Sons., 1999. ISBN 0-471-98285-7.

[ZCP^{+}99b] O.C. Zienkiewicz, A.H.C. Chan, M. Pastor, B.A. Schrefler, and T. Shiomi. *Computational Geomechanics– with Special Reference to Earthquake Engineering*. John Wiley and Sons Ltd., Baffins Lane, Chichester, West Sussex PO19 IUD, England, 1999.

[ZCP^{+}20] Rachele Zaccherini, Andrea Colombi, Antonio Palermo, Vasilis K. Dertimanis, Alessandro Marzani, Henrik R. Thomsen, Božidar Stojadinović, and Eleni N. Chatzi. Locally resonant metasurfaces for shear waves in granular media. *Phys. Rev. Applied*, 13:034055, Mar 2020.

[ZCY^{+}08] Yuyi Zhang, Joel P. Conte, Zhaohui Yang, Ahmed Elgamal, Jacobo Bielak, and Gabriel Acero. Two-dimensional nonlinear earthquake response analysis of a bridge-foundation-ground system. *Earthquake Spectra*, 24(2):343–386, 2008.

[ZD17] Zhibao Zheng and Hongzhe Dai. Simulation of multi-dimensional random fields by Karhunen-Loève expansion. *Computer Methods in Applied Mechanics and Engineering*, 324:221 – 247, 2017.

[ZDC^{+}] Wenyang Zhang, Yufeng Dong, Jorge G. F. Crempien, Pedro Arduino, and Ertugrul Taciroglu. Effects of soil nonlinearity on physics-based ground motion simulation and their implications on 1d site response analysis: An application to istanbul. *Earthquake Engineering & Structural Dynamics*, n/a(n/a).

[ZDC^{+}25] Wenyang Zhang, Yufeng Dong, Jorge G. F. Crempien, Pedro Arduino, and Ertugrul Taciroglu. Effects of soil nonlinearity on physics-based ground motion simulation and their implications on 1d site response analysis: An application to istanbul. *Earthquake Engineering & Structural Dynamics*, 54(9):2194–2211, 2025.

[ZDPB92a] Thomas Zimmermann, Yves Dubois-Pèlerin, and Patricia Bomme. Object oriented finite element programming: I. governing principles. *Computer Methods in Applied Mechanics and Engineering*, 98:291–303, 1992.

[ZDPB92b] Thomas Zimmermann, Yves Dubois-Pèlerin, and Patricia Bomme. Object oriented finite element programming: I. governing principles. *Computer Methods in Applied Mechanics and Engineering*, 98:291–303, 1992.

[ZDPB92c] Thomas Zimmermann, Yves Dubois-Pèlerin, and Patricia Bomme. *Object Oriented Finite Element Programming: Theory and Smalltalk V Implementation for FEM_Objects PC 01*. Elmepress International, P.O.Box 2 CH 1015 Lausanne 15, Switzerland, 1992.

[ZdPd^{+}06] Clara Zambelli, Claudio di Prisco, Anna d'Onofrio, Ciro Visone, and Filippo Santucci de Magistris. Dependency of the mechanical behaviour of granular soils on loading frequency: Experimental results and constitutive modelling. In *Soil Stress-Strain Behavior: Measurement, Modeling and Analysis Geotechnical Symposium in Roma*, March 16 & 17 2006.

[ZE96] Th. Zimmermann and D. Eyheramendy. Object–oriented finite elements I. principles of symbolic derivations and automatic programming. *Computer Methods in Applied Mechanics and Engineering*, 132:259–276, 1996.

[Zer09] Aspasia Zerva. *Spatial variation of seismic ground motions: modeling and engineering applications*. CRC Press, 2009.

[ZES04] M. Zeghal and U. El-Shamy. Micro mechanical modeling of site liquefaction and remediation through cementation. In *Geofrontiers 2005*, 2004.

[ZETS95] M. Zeghal, A-W Elgamal, H.T. Tang, and J.C. Stepp. Lotung downhole array. II: Evaluation of soil nonlinear properties. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 121(4):363–378, April 1995.

[ZFHL11] Zhong Zhi Fu and Si Hong Liu. Formulations of a hydromechanical interface element. *Acta Mechanica Sinica*, 27(5):697–705, 2011.

[ZFSP18] Aspasia Zerva, Mohammad Reza Falamarz-Sheikhabadi, and Masoud Khazaei Poul. Issues with the use of spatially variable seismic ground motions in engineering applications. In *Recent Advances in Earthquake Engineering in Europe: 16th European Conference on Earthquake Engineering-Thessaloniki 2018*, pages 225–252. Springer, 2018.

[ZGG21] Hao Zhang, Johann Guilleminot, and Luis J. Gomez. Stochastic modeling of geometrical uncertainties on complex domains, with application to additive manufacturing and brain interface geometries. *Computer Methods in Applied Mechanics and Engineering*, 385:114014, 2021.

[ZH82] Tony Faris Zahrah and William Joel Hall. Seismic energy absorption in simple structures. Technical report, University of Illinois Engineering Experiment Station, 1982.

[ZH84] Tony F Zahrah and William J Hall. Earthquake energy absorption in sdof structures. *Journal of Structural Engineering*, 110(8):1757–1772, 1984.

[ZHA94] Gordon W. Zeglinski, Ray S. Han, and Peter Aitchison. Object oriented matrix classes for use in a finite element code using C++. *International Journal for Numerical Methods in Engineering*, 37:3921–3937, 1994.

[Zha05] Lianyang Zhang. *Engineering properties of rocks*. Elsevier geo-engineering book series ;v. 4. Elsevier,, Amsterdam ; London, 2005.

[Zha09] Guang-Qing Zhang. Rock failure with weak planes by self-locking concept. *International Journal of Rock Mechanics & Mining Sciences*, 46:974–982, 2009.

[ZHL01] Y. Q. Zeng, J. Q. He, and Q. H. Liu. The application of the perfectly matched layer in numerical modeling of wave propagation in poroelastic media. *Geophysics*, 66(4):1258–1266, July-August 2001.

[ZHP94] O. C. Zienkiewicz, Maosung Huang, and M. Pastor. Numerical modelling of soil liquefaction and similar phenomena in earthquake engineering: State of the art. In Kandiah Arulanandan and Ronald F. Scott, editors, *Verification of Numerical Procedures for the Analysis of Soil Liquefaction Problems*, volume 2, pages 1401–1414, 1994.

[ZHY13] Yu-ye Zhang, Kent A Harries, and Wan-cheng Yuan. Experimental and numerical investigation of the seismic performance of hollow rectangular bridge piers constructed with and without steel fiber reinforced concrete. *Engineering Structures*, 48:255–265, 2013.

[Zie70] O.C. Zienkiewicz. *Analysis of Non Linear Problems in Rock Mechanics with Particular Reference to Jointed Rock Systems*. publisher not identified, 1970.

[Zie77a] H. Ziegler. *An Introduction to Thermomechanics*. North-Holland Publishing Company, Amsterdam, 1977.

[Zie77b] Olgierd Cecil Zienkiewicz. *The Finite Element Method*. McGraw - Hill Book Company, 3rd edition, 1977.

[Zie81] H Ziegler. Discussion of some objections to thermomechanical orthogonality. *Ingenieur-Archiv*, 50(3):149–164, 1981.

[Zie04] O. C. Zienkiewicz. The birth of the finite element method and of computational mechanics. *International Journal for Numerical Methods in Engineering*, 60(1):3–10, 2004.

[ZJS08a] H. Zhou, Y. Jia, and J.F. Shao. A unified elastic–plastic and viscoplastic damage model for quasi-brittle rocks. *International Journal of Rock Mechanics and Mining Sciences*, 45(8):1237–1251, December 2008.

[ZJS08b] H. Zhoua, Y. Jia, and J.F. Shao. A unified elastic-plastic and viscoplastic damage model for quasi-brittle rocks. *International Journal of Rock Mechanics & Mining Sciences*, 45:1237–1251, 2008.

[ZL98] W. K. Zietlow and J. F. Labuz. Measurement of the intrinsinic process zone in rock using acoustic emission. *International Journal of Rock Mechanics and Mining Sciences*, 35(3):291–299, 1998.

[ZL01] Ray Ruichong Zhang and Menglin Lou. Seismic wave motion modelling with layered 3D random heterogeneous media. *Probabilistic Engineering Mechanics*, 16(4):381–397, 2001.

[ZL03] FZ Zerfa and B Loret. Coupled dynamic elastic–plastic analysis of earth structures. *Soil dynamics and earthquake engineering*, 23(6):435–454, 2003.

[ZLG$^+$20] Wenyang Zhang, Keng-Wit Lim, S. Farid Ghahari, Pedro Arduino, and Ertugrul Taciroglu. On the implementation and validation of a three-dimensional pressure-dependent bounding surface plasticity model for soil nonlinear wave propagation and soil-structure interaction analyses. *International Journal for Numerical and Analytical Methods in Geomechanics*, n/a(n/a):1–29, 2020.

[ZLH22] Jiawen Zhang, Mingchao Li, and Shuai Han. Seismic analysis of gravity dam-layered foundation system subjected to earthquakes with arbitrary incident angles. *International Journal of Geomechanics*, 22(2):04021279, 2022.

[ZLLT05] H. Zheng, D. F. Liu, C. F. Lee, and L. G. Tham. Displacement-controlled method and its applications to material non-linearity. *International Journal for Numerical and Analytical Methods in Geomechanics*, 29(3):209–226, 2005.

[ZLWL99] J. Zhao, H.B. Li, M.B. Wu, and T.J. Li. Dynamic uniaxial compression tests on a granite. *International Journal of Rock Mechanics and Mining Sciences*, 36(2):273–277, February 1999.

[ZLX$^+$23] Enping Zhu, Tao Li, Jinbiao Xiong, Xiang Chai, Tengfei Zhang, and Xiaojing Liu. A super-real-time three-dimension computing method of digital twins in space nuclear power. *Computer Methods in Applied Mechanics and Engineering*, 417:116444, 2023.

[ZLZY24] Shixun Zhang, Feiyu Liu, Weixiang Zeng, and Mengjie Ying. Analyzing cyclic shear behavior at the sand-rough concrete interface: An experimental and dem study across varying displacement amplitudes. *International Journal for Numerical and Analytical Methods in Geomechanics*, n/a(n/a), 2024.

[ZM01] Jian Zhang and Nicos Makris. Rocking response of free-standing blocks under cycloidal pulses. *ASCE Journal of Engineering Mechanics*, 127(5):473–483, 2001.

[ZM02a] Jian Zhang and Nicos Makris. Kinematic response functions and dynamic stiffness of bridge embankments. *Earthquake Engineering and Structural Dynamics*, 31:1933–1966, 2002.

[ZM02b] Jian Zhang and Nicos Makris. Seismic response analysis of highway overcrossing including soil–structure interaction. *Earthquake Engineering and Structural Dynamics*, 31:1967–1991, 2002.

[ZML99a] Limin Zhang, Michael McVay, and Peter Lai. Numerical analysis of laterally loaded 3x3 to 7x3 pile groups in sands. *Journal of Geotechnical and Geoenvironmental Engineering*, 125(11):936–946, Nov. 1999.

[ZML99b] Limin Zhang, Michael C. McVay, and Peter W. Lai. Centrifuge modelling of laterally loaded single battered piles in sands. *Canadian Geotechnical Journal*, 36(6):1074–1084, 1999.

[ZMS18] Minjie Zhu, Frank McKenna, and Michael H. Scott. OpenSeesPy: Python library for the OpenSees finite element framework. *SoftwareX*, 7:6 – 11, 2018.

[ZN99] X. Zeng and B. Ni. Stress induced anisotroic G_{max} of sands and its measurments. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 125(9), 1999.

[ZN23] Zhibao Zheng and Udo Nackenhorst. A nonlinear stochastic finite element method for solving elastoplastic problems with uncertainties. *International Journal for Numerical Methods in Engineering*, 124(16):3411–3435, 2023.

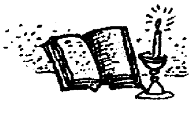
[ZNGD] Kuanshi Zhong, Javier G. Navarro, Sanjay Govindjee, and Gregory G. Deierlein. Surrogate modeling of structural seismic response using probabilistic learning on manifolds. *Earthquake Engineering & Structural Dynamics*, n/a(n/a).

[ZNM93] M. M. Zaman, Y. M. Najjar, and A. Muqtadir. Effects of cap thickness and pile inclination on the response of a pile group foundation by a three-dimensional nonlinear finite element analysis. *Computers and Geotechnics*, 15, 1993.

[Zoh23] T.I. Zohdi. Machine-learning a perfect bending soccer goal shot. *Computer Methods in Applied Mechanics and Engineering*, 415:116261, 2023.

[ZP71] O. C. Zienkiewicz and D. V. Phillips. An automatic mesh generation scheme for plane and curved surfaces by 'isoparametric' co–ordinates. *International Journal for Numerical Methods in Engineering*, 3:519–528, 1971.

[ZP02] Xiaoqing Zheng and Alex Pang. Volume deformation for tensor visualization. In *Processing of IEEE Visualization 02*, pages 379–386, Boston, 2002.



[ZP03] Xiaoqing Zheng and Alex Pang. Interaction of light and tensor fields. In G.-P. Bonnneau, S. Hahmann, and C. D. Hansen, editors, *Processing of Joint EUROGRAPHICS – IEEE TCVG Symposium on Visualization*, 2003.

[ZPC88] O.C. Zienkiewicz, D.K. Paul, and A.H. Chan. Unconditionally stable staggered solution procedure for soil-pore fluid interaction problems. *International Journal for Numerical Methods in Engineering*, 26:1039–1055, 1988.

[ZPHH0] D. Zhang, K. Phoon, H. Huang, and Q. Hu. Characterization of model uncertainty for cantilever deflections in undrained clay. *Journal of Geotechnical and Geoenvironmental Engineering*, 0(0):04014088, 0.

[ZPVH22] Xiangcou Zheng, Federico Pisanò, Philip J. Vardon, and Michael A. Hicks. Fully implicit, stabilised, three-field material point method for dynamic coupled problems. *Engineering with Computers*, 38(6):5583–5602, 2022.

[ZR02] Lupei Zhu and Luis A Rivera. A note on the dynamic and static displacements from a point source in multilayered media. *Geophysical Journal International*, 148(3):619–627, 2002.

[ZRC$^+$20] Wenyang Zhang, Doriam Restrepo, Jorge G.F. Crempien, Bulent Erkmen, Ricardo Taborda, Asli Kurtulus, and Ertugrul Taciroglu. A computational workflow for rupture-to-structural-response simulation and its application to Istanbul. *Earthquake Engineering & Structural Dynamics*, n/a(n/a):20, 2020.

[ZRR96] M Zhou, G Ravichandran, and AJ Rosakis. Dynamically propagating shear bands in impact-loaded prenotched plates-ii. numerical simulations. *Journal of the Mechanics and Physics of Solids*, 44(6):1007–1032, 1996.

[ZS84] O. C. Zienkiewicz and T. Shiomi. Dynamic behaviour of saturated porous media; the generalized Biot formulation and its numerical solution. *International Journal for Numerical and Analytical Methods in Geomechanics*, 8:71–96, 1984.

[ZS95] Q. H. Zuo and H. L. Schreyer. A note on pure–longitudinal and pure–shear waves in cubic cristals. *Journal of Acoustic Society of America*, 98(1):580–583, July 1995.

[ZS99] N Zabaras and A Srikanth. Using Objects to Model Finite Deformation Plasticity 2 . A Lagrangian Analysis for Large Deformations of Microporous Materials. *Engineering with Computers*, pages 37–60, 1999.

[ZS14] Minjie Zhu and Michael H. Scott. Modeling fluide-structure interaction by the particle finite element method in opensees. *Computers & Structures*, 132(0):12 – 21, 2014.

[ZS19] Minjie Zhu and Michael H. Scott. Fluid-structure interaction and python-scripting capabilities in OpenSees. Technical Report 2019/06, Pacific Earthquake Engineering Research Center (PEER), August 2019.

[ZS21] Zeyu Zhao and Mrinal K. Sen. A gradient-based markov chain monte carlo method for full-waveform inversion and uncertainty analysis. *GEOPHYSICS*, 86(1):R15–R30, 2021.

[ZSA14] J Zamani, B Soltani, and M Aghaei. Analytical investigation of elastic thin-walled cylinder and truncated cone shell intersection under internal pressure. *Journal of pressure vessel technology*, 136(5):051201, 2014.

[ZSBT20] W. Zhang, M. Shokrabadi, Y. Bozorgnia, and E. Taciroglu. A methodology for fragility analysis of buried water pipes considering coupled horizontal and vertical ground motions. *Computers and Geotechnics*, 126:103709, 2020.

[ZSC20] Degao Zou, Yi Sui, and Kai Chen. Plastic damage analysis of pile foundation of nuclear power plants under beyond-design basis earthquake excitation. *Soil Dynamics and Earthquake Engineering*, 136:106179, 2020.

[ZSGD23] Farhad Zeighami, Leonardo Sandoval, Alberto Guadagnini, and Vittorio Di Federico. Uncertainty quantification and global sensitivity analysis of seismic metabarriers. *Engineering Structures*, 277:115415, 2023.

[ZSM95] Jorge G. Zornberg, Nicholas Sitar, and James K. Mitchell. Performance of geosynthetic reinforced slopes at failure: A centrifuge study. Geotechnical Engineering Report UCB/GT/95-01, University of California at Berkeley, April 1995.

[ZSM98a] Jorge G. Zornberg, Nicholas Sitar, and James K. Mitchell. Limit equilibrium as basis for design of geosynthetic reinforced slopes. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 124(8):684–698, 1998.

[ZSM98b] Jorge G. Zornberg, Nicholas Sitar, and James K. Mitchell. Performance of geosynthetic reinforced slopes at failure. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 124(8):670–683, 1998.

[ZST19] W. Zhang, E. Esmaeilzadeh Seylabi, and E. Taciroglu. An ABAQUS toolbox for soil-structure interaction analysis. *Computers and Geotechnics*, 114:103143, 2019.

[ZT91a] Olgierd Cecil Zienkiewicz and Robert L. Taylor. *The Finite Element Method*, volume 2. McGraw - Hill Book Company, Fourth edition, 1991.

[ZT91b] Olgierd Cecil Zienkiewicz and Robert L. Taylor. *The Finite Element Method*, volume 1. McGraw - Hill Book Company, fourth edition, 1991.

[ZT95] J. Zheng and T. Takeda. Effects of soil–structure interaction on seismic response of pc cable–stayed bridge. *Soil Dynamics and Earthquake Engineering*, 14:427–437, 1995.

[ZT97] O.C. Zienkiewicz and R.L. Taylor. The finite element patch test revisited a computer test for convergence, validation and error estimates. *Computer Methods in Applied Mechanics and Engineering*, 149(1):223 – 254, 1997. Containing papers presented at the Symposium on Advances in Computational Mechanics.

[ZT21] Wenyang Zhang and Ertugrul Taciroglu. 3D time-domain nonlinear analysis of soil-structure systems subjected to obliquely incident sv waves in layered soil media. *Earthquake Engineering & Structural Dynamics*, 50(8):2156–2173, 2021.

[ZTA14] Irmelda Zentner, Paola Traversa, and Frederic Allain. Time histories for seismic analysis of structures - pros and cons of available methods. In *Second European Conference on Earthquake Engineering and Seismology*, 2014.

[ZTZ67] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals*. Elsevier, sixth, 2005 edition, 1967.

[ZVOB14] Rita E. Zapata-Vázquez, Anthony O'Hagan, and Leonardo Soares Bastos. Eliciting expert judgements about a set of proportions. *Journal of Applied Statistics*, 41:1919–1933, 2014.

[ZW87] Hans Ziegler and Christoph Wehrli. The derivation of constitutive relations from the free energy and the dissipation function. *Advances in Applied Mechanics*, 25:183–238, 1987.

[ZW12] Jian-Min Zhang and Gang Wang. Large post-liquefaction deformation of sand, part i: physical mechanism, constitutive description and numerical algorithm. *Acta Geotechnica*, 7(2):69–113, 2012.

[ZWD$^+$18] Mi Zhao, Lihua Wu, Xiuli Du, Zilan Zhong, Chengshun Xu, and Liang Li. Stable high-order absorbing boundary condition based on new continued fraction for scalar wave propagation in unbounded multilayer media. *Computer Methods in Applied Mechanics and Engineering*, 334:111–137, 2018.

[ZWJ13] Wangcheng Zhang, Jianfeng Wang, , and Mingjing Jiang. DEM-aided discovery of the relationship between energy dissipation and shear band formation considering the effects of particle rolling resistance. *ASCE Journal of Geotechnical and Geoenvironmental Engineering*, 139(9):1512–1527, September 2013.

[ZWL05] Hongwu Zhang, Hui Wang, and Guozhen Liu. Quadrilateral isoparametric finite elements for plane elastic Cosserat bodies. *Acta Mechanica Sinica*, 21(4):388–394, 2005.

[ZYL94] W. Q. Zhu, J. S. Yu, and Y. K. Lin. On improved stochastic averaging procedure. *Probabilistic Engineering Mechanics*, 9(3):203–211, 1994.

[ZZ02] Aspasia Zerva and Vassilios Zervas. Spatial variation of seismic ground motions: An overview. *ASME Applied Mechanics Revue*, 55(3):271–297, May 2002.

[ZZC03] H. Zhang, X. Zhang, and J.S. Chen. A new algorithm for numerical solution of dynamic elastic-plastic hardening and softening problems. *Computers and Structures*, 81:1739–1749, 2003.

[ZZZ] Chao Zeng, Chunfeng Zhao, and Farhad Zeighami. Seismic surface wave attenuation by resonant metasurfaces on stratified soil. *Earthquake Engineering & Structural Dynamics*, n/a(n/a).