

UCD CompGeoMech Contributions to OpenSees: Deliverables

Prof. Yang, Dr. Cheng,
GSRs Jie, Sett, Taiebat, Preisig, Wu, Liu

Boris Jeremić,

University of California, Davis

OpenSees Annual Meeting, San Francisco, January 2007,

Outline

- 1 Material Models, Elements and Procedures
 - Elastic Material Models
 - Elastic–Plastic (Template3DEP) Continuum Models
 - Single Phase Solid Finite Elements
 - Multi Phase Solid Finite Elements, Coupled
 - Solution Control Computational Procedure
 - Domain Reduction Method: Seismic Loading Application
 - Other...
- 2 Deliverables: People, Implementation and Documents
 - Theory Manual
 - Implementation Manual
 - Examples Manual

Elastic Material Models

- Small deformation elasticity
 - linear isotropic
 - nonlinear isotropic
 - cross anisotropic
- Large deformation hyperelasticity
 - Neo–Hookean
 - Ogden
 - Logarithmic
 - Mooney–Rivlin
 - Simo–Pister

Elastic–Plastic Continuum Models: Small Deformations

- Yield surfaces:
 - von Mises
 - Drucker–Prager
 - Cam–Clay
 - Rounded Mohr–Coulomb
 - Parabolic Leon
- Plastic flow directions (plastic potential functions):
 - von Mises
 - Drucker–Prager
 - Cam–Clay
 - Rounded Mohr–Coulomb
 - Parabolic Leon
 - Dafalias Manzari

Elastic–Plastic Continuum Models: Small Deformations (continued)

- Evolution Laws (hardening and/or softening laws):
 - linear scalar,
 - nonlinear scalar (Cam–Clay type),
 - linear tensorial (kinematic hardening/softening: translational and/or rotational)
 - nonlinear tensorial (kinematic hardening/softening: translational and/or rotational)
 - Armstrong–Frederick hardening
 - bounding surface hardening/softening

Hyperelastic–Plastic Continuum Models: Large Deformations

- Yield surfaces
 - von Mises,
 - Drucker–Prager...
- Plastic flow directions (plastic potential functions):
 - Drucker–Prager,
 - von Mises,
- Evolution Laws:
 - linear and nonlinear scalar,
 - nonlinear scalar
 - linear and nonlinear (AF) tensorial (kinematic hardening/softening: translational and/or rotational)

Single Phase Formulations

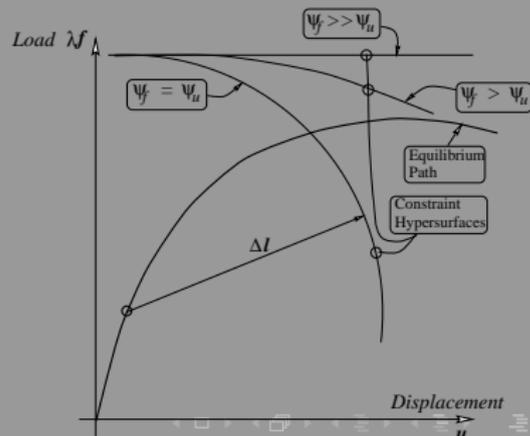
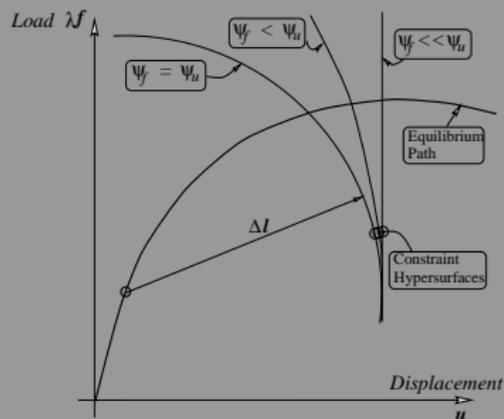
- Small deformation solid elements, bricks (8, 20, 21, 27, 8-20 variable node bricks)
- Stochastic Elastic–Plastic Finite Element Method
- Large deformation (total Lagrangian) solid elements, bricks (20 node brick)

Multi Phase Formulations

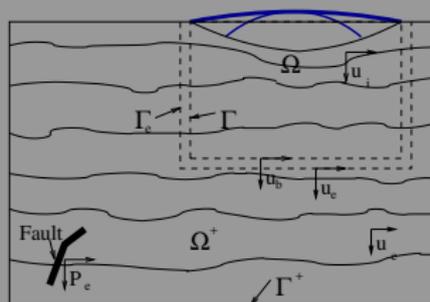
- Fully coupled, u – p – U elements (3D) for small deformations
- Fully coupled, u – p (3D) elements for small deformations
- Fully coupled u – p (3D) elements for large deformations

Hyperspherical Arc–length Solution Control

$$\begin{bmatrix} \mathbf{K}_t & -\mathbf{f}_{ext} \\ 2\frac{\psi_u^2}{u_{ref}^2}\Delta\mathbf{u}^T\mathbf{S} & 2\Delta\lambda\psi_f^2 \end{bmatrix} \begin{bmatrix} \delta\mathbf{u} \\ \delta\lambda \end{bmatrix} = - \begin{bmatrix} \mathbf{r}^{old} \\ a^{old} \end{bmatrix}$$



Domain Reduction Method (Bielak et al.)



$$\begin{Bmatrix} p_i^{eff} \\ p_b^{eff} \\ p_e^{eff} \end{Bmatrix} = \begin{Bmatrix} 0 \\ -M_{be}^{\Omega+} \ddot{u}_e^0 - K_{be}^{\Omega+} u_e^0 \\ M_{eb}^{\Omega+} \ddot{u}_b^0 + K_{eb}^{\Omega+} u_b^0 \end{Bmatrix}$$

Other Related (more or less) Work

- Plastic Domain Decomposition (PDD) method
- Stochastic Elastic–Plastic Finite Element Method
- Visualization for Earthquake Engineering Simulations (VEES) system driver (computational steering, pre– and post–processing)

Documentation: Literate Programming

- Implementation follows formulation
- Formulation follows mechanics and numerics
- Methodology goal: low Kolmogorov (program-size) complexity
- Verification, Validation and Prediction
- Open Source (GPL)

Theory Manual

- Detailed theoretical background
- 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*
- Support for literate programming (DEK)

Implementation Manual

- Application Programming Interface (API)
 - Public API is forever (asset, liability)
 - Realistically, API evolves
 - API design is tough (perfection is unachievable, but try anyway)
- Literate programming drives API

Examples Manual

- Basic validation examples, (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.*)
- Advanced, practical, real life predictive examples (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.)
- Command manual part of examples manual

Conclusions

- A rich variety of models, elements and procedures for have been
 - developed
 - implemented
 - documented
- Conclusions:
 - none really
 - work goes on after PEER ends