Albany Developer Meeting, October 2nd 2012, Albuquerque, NM

SAND2012-8952P

Constitutive Modeling – Implementation, Calibration, V&V

Qiushi Chen, Jake Ostien
Mechanics of Materials
Sandia National Laboratories, Livermore CA

Outline

Shear-dominated ductile failure

- Shear-modified Gurson damage model
    - Model formulation
    - Numerical implementation and verification
- Model Calibration and Validation
    - Experiments for 6061-T651 Aluminum
    - Optimization-based parameter calibration (use Dakota, with Kyle Karlson)

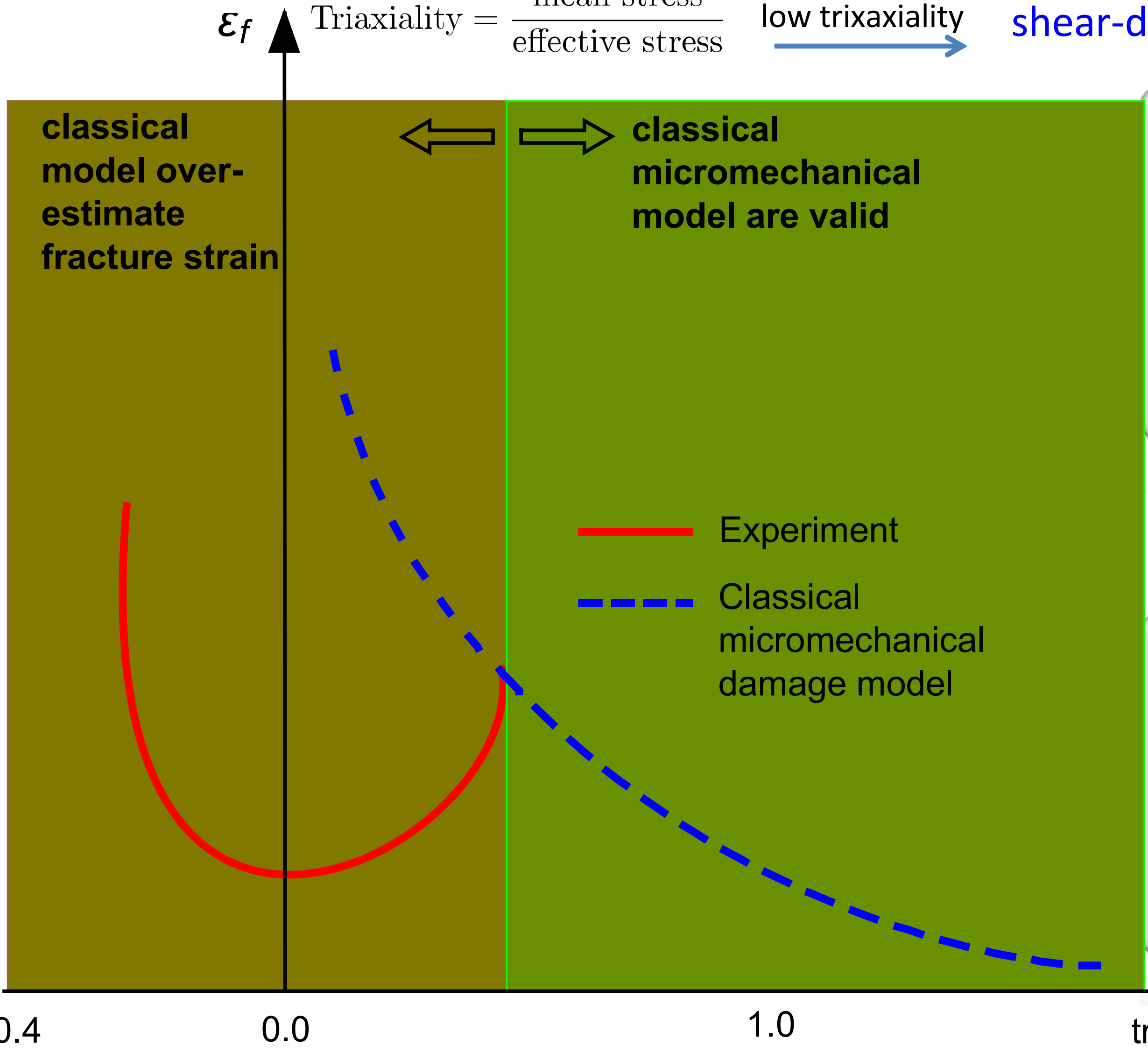
Cap-plasticity model for porous geomaterials

- Model formulation
- Numerical implementation and verification
- 3D penetration problem on Salem limestone (couple cap plasticity model with poro-mechanical problem)

Ductile Failure and Stress Triaxiality

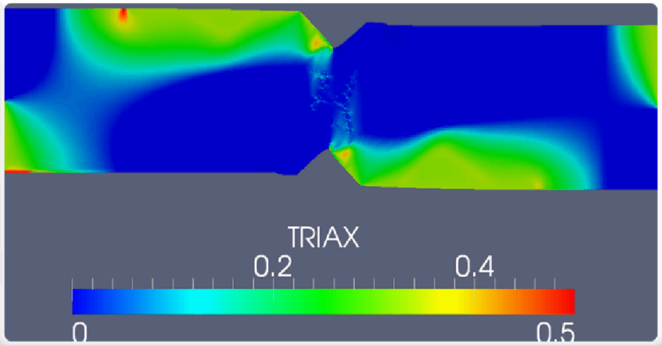
$$\text{Triaxiality} = \frac{\text{mean stress}}{\text{effective stress}}$$

low trixaxiality → shear-dominated state



Schematic representation of the strain to failure of a ductile metal

typical ductile failure of specimen in tensile test



Triaxiality contour in double-notched shear

Classical Gurson Damage Model

The macroscopic yield surface

$$\Phi(\boldsymbol{\tau}, Y, f) = \frac{3}{2}\frac{\boldsymbol{s}:\boldsymbol{s}}{Y^2} + \boxed{2q_1 f \cosh(\frac{3q_2 p}{2Y})} - q_3 f^2 - 1 = 0$$

pressure dependent

$p = \mathrm{tr}(\boldsymbol{\tau})/3$	mean stress
$s = \mathrm{dev}(\boldsymbol{\tau})$	deviatoric stress tensor
Y	current effective stress of the damage-free matrix material
f	void volume fraction ← damage parameter
q_1, q_2, q_3	model fitting parameters [Tvergaard 1990]

damage-free $f = 0$

$$\Phi = \frac{3}{2}\frac{\boldsymbol{s}:\boldsymbol{s}}{Y^2} - 1$$

OR $\Phi = \|\boldsymbol{s}\| - \sqrt{\frac{2}{3}}Y$

yield surface for J2 plasticity!

Hardening law for matrix material

- Saturation-type

$$Y = Y_0 + Y_\infty\left[1 - \exp(-\delta\varepsilon_q)\right] + K\varepsilon_q$$

- Power-law

$$Y = Y_0(1 + E\epsilon_q/Y_0)^N$$

- Hardening minus recovery model

$$Y = Y_0(\theta) + 2\mu(\theta)\epsilon_{ss}$$

Evolution of eqps

$$\dot{\varepsilon}_q Y(1-f) = \boldsymbol{\tau} : \left(\gamma\frac{\partial\Phi}{\partial\boldsymbol{\tau}}\right)$$

where

$$\frac{\partial\Phi}{\partial\boldsymbol{\tau}} = \boldsymbol{s} + \frac{1}{3}q_1 q_2 Y f \sinh(v)\mathbf{1}$$

Shear Modification

Classical damage evolution law

$$\dot{f} = \dot{f}_g + \dot{f}_{nu} \qquad \dot{f}_{nu} = A\dot{\varepsilon}_q \qquad A(\varepsilon_q) = \begin{cases} \frac{f_N}{s_N\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\varepsilon_q - \epsilon_N}{s_N}\right)^2\right], & p \geq 0 \\ 0, & p < 0 \end{cases}$$

original void growth law, physically based and derived by [Gurson 1977]

$$\dot{f}_g = (1-f)\,\mathrm{tr}\left(\gamma\frac{\partial\Phi}{\partial\boldsymbol{\tau}}\right)$$

← predicts no void growth under shear state

Recent experiment evidence suggests classical model over estimate the fracture strain

Shear-modified damage growth law

Introducing third-stress-invariant dependent [Nahshon-Hutchinson 2008]

$$\omega(\boldsymbol{\tau}) = 1 - \left(\frac{27J_3}{2\tau_e^3}\right)^2$$

where $\tau_e := \sqrt{3/2}\|\boldsymbol{s}\|$ effective stress

$J_3 := \det(\boldsymbol{s})$ third stress invariant

$$\dot{f}_g = (1-f)\,\mathrm{tr}\left(\gamma\frac{\partial\Phi}{\partial\boldsymbol{\tau}}\right) + k_\omega f\frac{\omega(\boldsymbol{\tau})}{\tau_e}\boldsymbol{s} : \left(\gamma\frac{\partial\Phi}{\partial\boldsymbol{\tau}}\right)$$

k_ω material constant that sets the magnitude of the damage growth rate in pure shear states

Hyperelastic Constitutive Relation

Strain energy function

$$\Psi = \Psi^{\mathrm{vol}}[J_e] + \Psi^{\mathrm{iso}}[\hat{\boldsymbol{b}}_e]$$

The volumetric and isochoric parts

$$\Psi^{\mathrm{vol}}[J_e] = \frac{1}{2}\kappa(\ln J_e)^2, \quad \Psi^{\mathrm{iso}}[\hat{\boldsymbol{b}}_e] = \frac{1}{4}\mu \ln \hat{\boldsymbol{b}}_e : \ln \hat{\boldsymbol{b}}_e$$

where

$$\hat{\boldsymbol{b}}_e := J_e^{-2/3}\boldsymbol{b}_e$$

$$J_e := \det \boldsymbol{F}_e$$

$$\boldsymbol{b}_e := \boldsymbol{F}_e\boldsymbol{F}_e^T$$

κ bulk modulus

μ shear modulus

Elastic constitutive law and the Kirchhoff stress

$$\boldsymbol{\tau} = \kappa \ln J_e \boldsymbol{g}^{-1} + \mu \ln \hat{\boldsymbol{b}}_e$$

The Kirchhoff pressure and deviatoric stress tensor

$$p = \frac{1}{3}\,\mathrm{tr}(\boldsymbol{\tau}) = \frac{1}{2}\kappa \ln \det \boldsymbol{b}_e$$

$$\boldsymbol{s} = \mathrm{dev}(\boldsymbol{\tau}) = \mu\,\mathrm{dev}\ln \boldsymbol{b}_e$$

Flow rule

from principal of maximum dissipation

$$-\frac{1}{2}L_v(\boldsymbol{b}_e)\cdot \boldsymbol{b}_e^{-1} = \gamma\frac{\partial \Phi}{\partial \boldsymbol{\tau}}$$

Numerical implementation

An implicit objective integration algorithm is implemented for integrating stress response over a finite time step. Backward Euler is applied to rate equations.

Local unknown vector (4x1)

$$\boldsymbol{X} = \{p, f, \varepsilon_q, \Delta\gamma\}$$

Local nonlinear system of equations (4x1)

$$R_1(\boldsymbol{X}) = \frac{1}{2}\boldsymbol{s} : \boldsymbol{s} - \frac{1}{2}\psi Y^2$$

$$R_2(\boldsymbol{X}) = p - p^{\mathrm{tr}} + q_1 q_2 \kappa \Delta\gamma Y f \sinh(\upsilon)$$

$$R_3(\boldsymbol{X}) = f - f_n - q_1 q_2 (1-f)\Delta\gamma Y f \sinh(\upsilon) - \sqrt{\frac{2}{3}}\Delta\gamma k_\omega f \omega(\boldsymbol{\tau})\|\boldsymbol{s}\| - A(\varepsilon_q - \varepsilon_{q(n)})$$

$$R_4(\boldsymbol{X}) = \varepsilon_q - \varepsilon_{q(n)} - \frac{\Delta\gamma}{Y(1-f)}\left(\boldsymbol{s} : \boldsymbol{s} + q_1 q_2 p Y f \sinh(\upsilon)\right)$$

Iterative solution procedure like the Newton's method requires consistent linearisation, therefore the computation of local Jacobian matrix (4x4) $\boldsymbol{J} = \partial \boldsymbol{R} / \partial \boldsymbol{X}$ ←

difficulty and time-consuming to derive analytical expression!

Remarks

Forward Automatic Differentiation (FAD) is used to obtain local Jacobian matrix.
LocalNonlinearSolver is used to solve the linearized equation, and upon convergence, compute the system sensitivity information.

Implicit integration algorithm for shear-modified Gurson damage model

```
GIVEN: $\varepsilon_{q(n)}, f_n, \boldsymbol{b}_{e(n)}$ and $\boldsymbol{F}$
FIND:    $\boldsymbol{\tau}, \varepsilon_q, f, \boldsymbol{b}_e(\boldsymbol{F}_p)$
STEP 1. Compute trial elastic left Cauchy-Green tensor $\boldsymbol{b}_e^{\mathrm{tr}}$
STEP 2. Compute trial Kirchhoff pressure and deviatoric tensor $p^{\mathrm{tr}}, \boldsymbol{s}^{\mathrm{tr}}$
STEP 3. Check yielding: $\Phi^{\mathrm{tr}}(p^{\mathrm{tr}}, \boldsymbol{s}^{\mathrm{tr}}, \varepsilon_{q(n)}, f_n) > 0$ ?
        No, set $p = p^{\mathrm{tr}}, \boldsymbol{s} = \boldsymbol{s}^{\mathrm{tr}}, \boldsymbol{b}_e = \boldsymbol{b}_e^{\mathrm{tr}}, \varepsilon_q = \varepsilon_{q(n)}, f = f_n$ and exit
STEP 4. Yes, local Newton loop
        4.1 Initialize $\boldsymbol{X}^k$ and iteration count $k = 0$
        4.2 Assemble residual $\boldsymbol{R}(\boldsymbol{X}^k)$
        4.3 Check convergence: $\| \boldsymbol{R} \| < tol$ ?
            Yes, converged and go to STEP 5
        4.4 No, compute local Jacobian matrix $\boldsymbol{J} = \partial \boldsymbol{R} / \partial \boldsymbol{X}$    FAD
        4.5 Solve system of equations $\boldsymbol{J} \cdot \delta \boldsymbol{X} = \boldsymbol{R}$ for $\delta \boldsymbol{X}$    LocalNonlinearSolver
        4.6 Update $\boldsymbol{X}^{k+1} = \boldsymbol{X}^k - \delta \boldsymbol{X}$, $k \rightarrow k + 1$ and go to 4.2
STEP 5. Update $\boldsymbol{\tau} = \boldsymbol{s} + p\boldsymbol{g}$, and $\varepsilon_q, f, \boldsymbol{F}_p^*$
```

*The plastic deformation gradient is updated upon local convergence

$$\boldsymbol{F}_p = \exp\left(\frac{\partial \Phi}{\partial \boldsymbol{\tau}}\right) \cdot \boldsymbol{F}_{b(n)}$$

This integration needs to be done for each global iteration within a global loading step.

example code from GursonFD_Def.hpp using FAD and NonLinearSolver

```
    // initialize local unknown vector
    X[0] = dgam;  X[1] = p;  X[2] = fvoid;  X[3] = eq;

    {// local Newton-Raphson loop

    // initialize DFadType local unknown vector Xfad
    // Note that since Xfad is a temporary variable that gets changed within local
iterations
    // when we initialize Xfad, we only pass in the values of X, NOT the system
sensitivity information
    for (std::size_t i = 0; i < 4; ++i) {
      Xval[i] = Sacado::ScalarValue<ScalarT>::eval(X[i]);
      Xfad[i] = DFadType(4, i, Xval[i]);
     }
    :
    :
    // local system of equations
    Rfad[0] = Phi;
    Rfad[1] = pfad - p
        + dgam * q1 * q2 * kappa * Ybar * fvoidfad * std::sinh(tmp);
    Rfad[2] = fvoidfad - fvoid - dfg - dfn;
    Rfad[3] = eqfad - eq - deq;

    // get ScalarT Residual
    for (int i = 0; i < 4; i++)
      R[i] = Rfad[i].val();

    // get local Jacobian
    for (int i = 0; i < 4; i++)
      for (int j = 0; j < 4; j++)
        dRdX[i + 4 * j] = Rfad[i].dx(j);
    :
    :
    // call LocalNonlinearSolver
    solver.solve(dRdX, X, R);
   } // end local Newton loop

    // compute sensitivity information w.r.t system parameters, and pack back to X
    solver.computeFadInfo(dRdX, X, R);
```

← “unpack” system sensitivity information

local Jacobian matrix (4x4)

$\boldsymbol{J} = \partial \boldsymbol{R} / \partial \boldsymbol{X}$

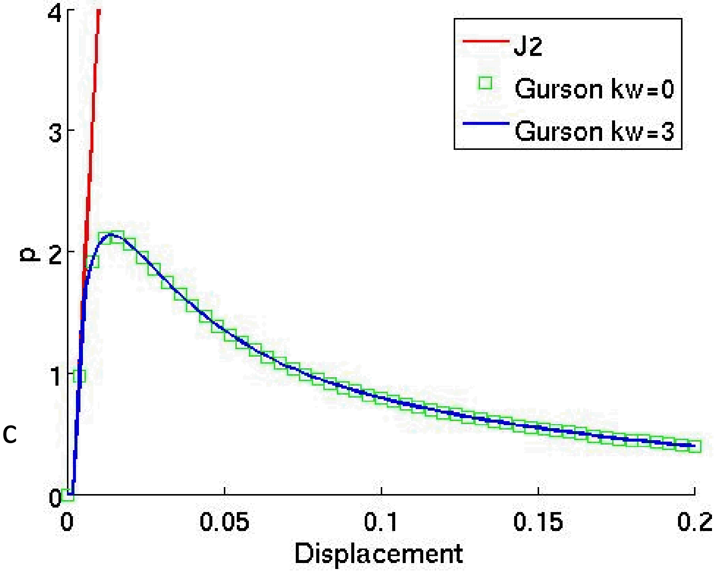
← LocalNonlinearSolver

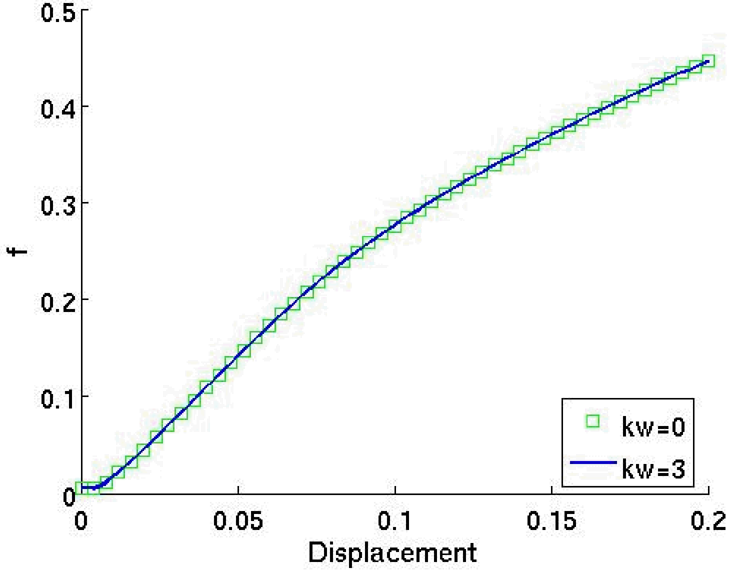
← compute system sensitivity

Verification: element tests

hydrostatic extensioin

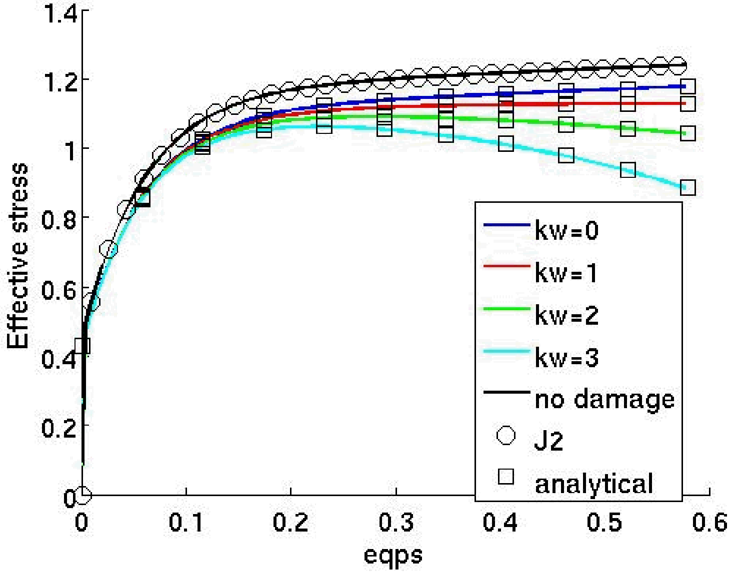
- J2 predicts no plasticity (pressure-independent).
- Gurson model response independent of shear term, because it's purely hydrostatic stress state.

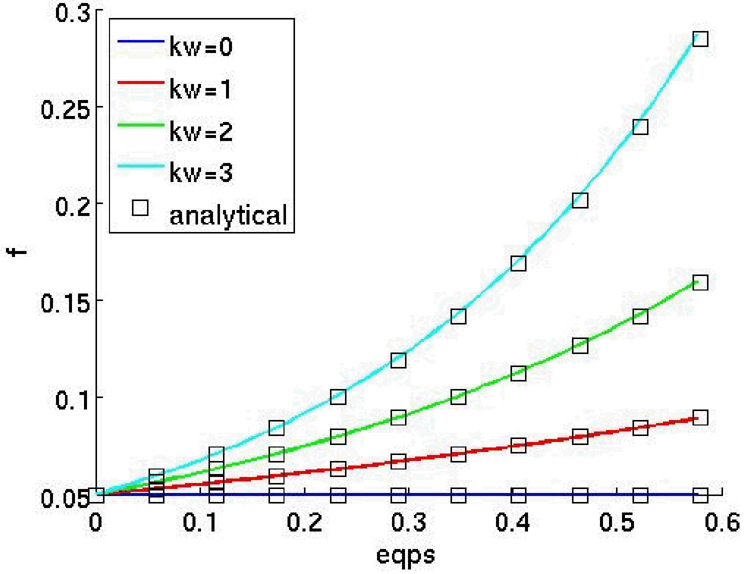




simple shear

- Analytical solution can be derived if void nucleation is neglected.
- Damage free Gurson recovers J2 model.
- Without shear term, the classical Gurson model predicts no damage growth.





Outline

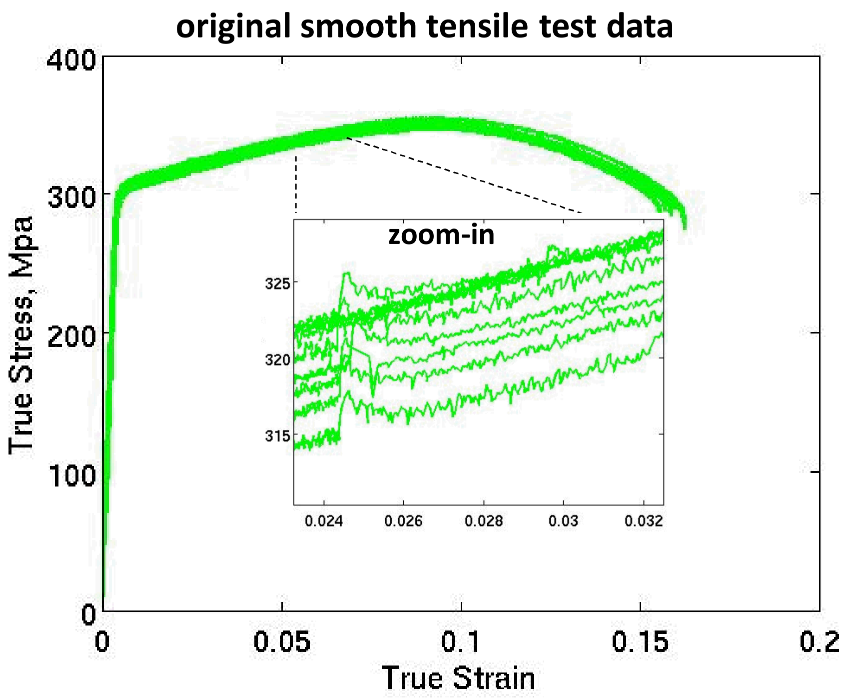
Shear-dominated ductile failure

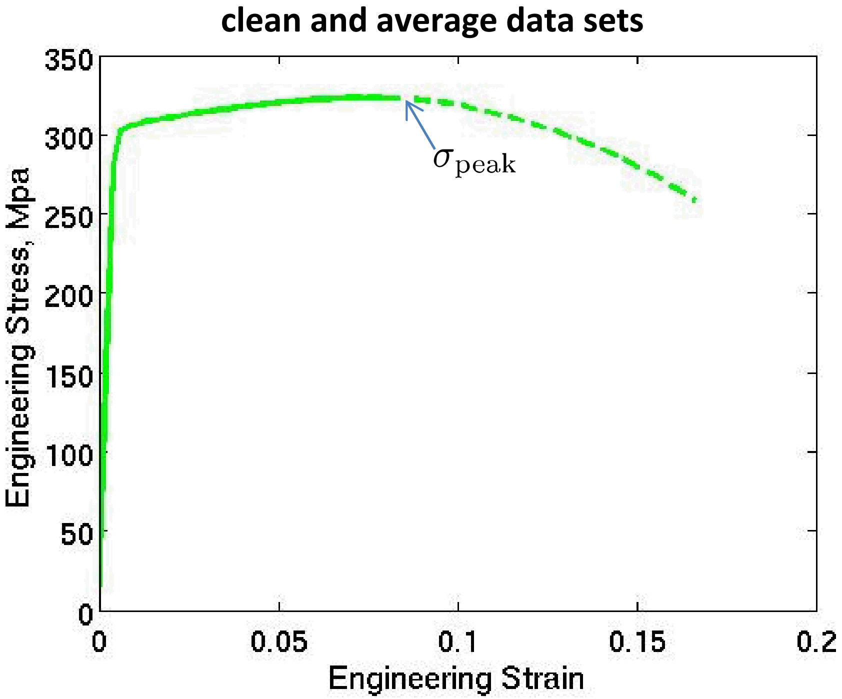
- Shear-modified Gurson damage model
  - Hyper-elastic formulation
  - Numerical implementation and verification
- Model Calibration and Validation
  - Experiments for 6061-T651 Aluminum
  - Optimization-based parameter calibration (Use ***Dakota***, with Kyle Karlson)

Cap-plasticity model for geomaterials

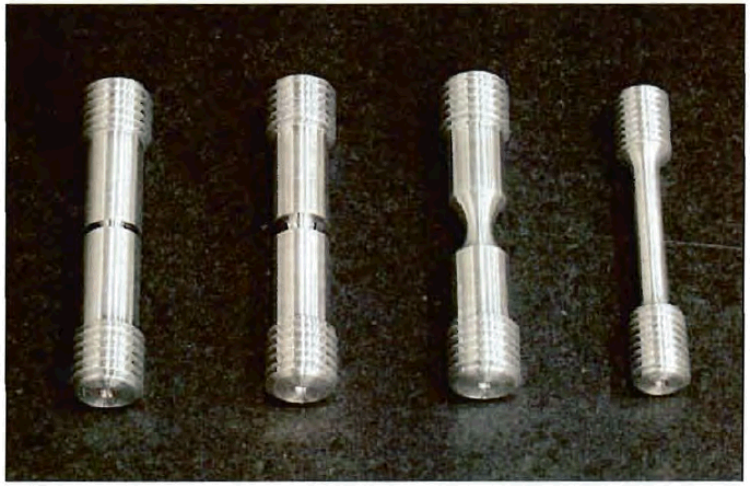
- Model formulation
- Numerical implementation and verification
- 3D penetration problem on Salem limestone (couple cap plasticity model with poro-mechanical problem)

Smooth Tensile Test for 6061-T651 Aluminum



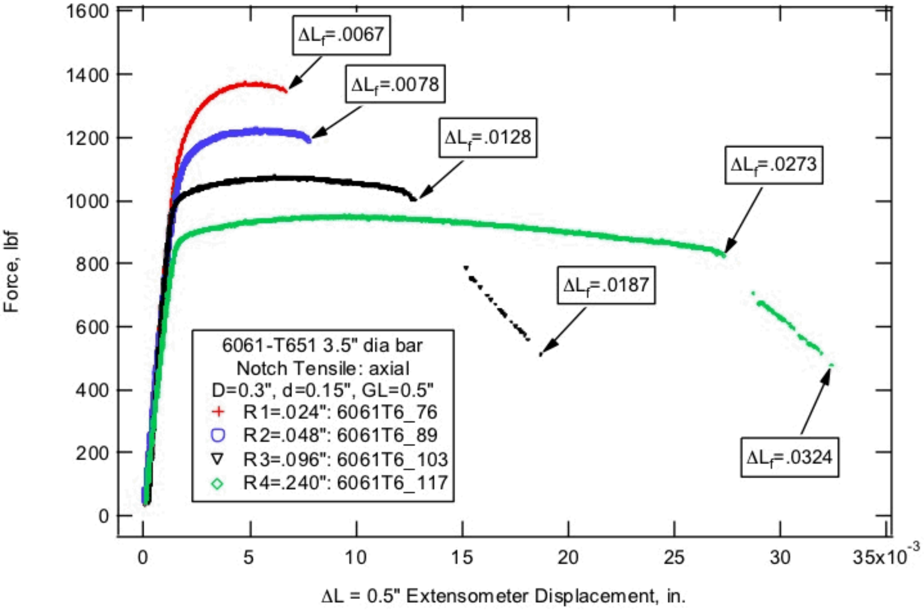


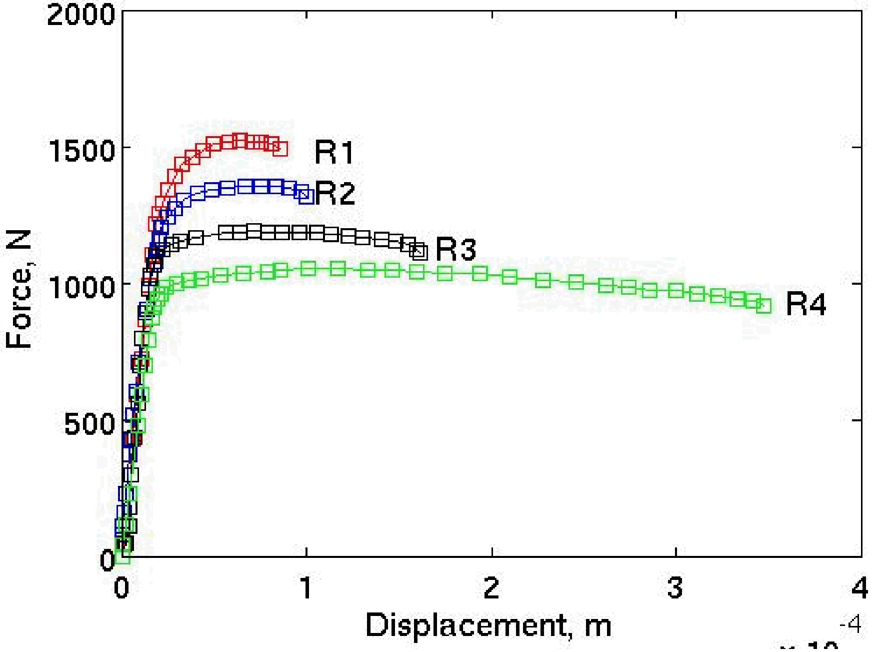
- 8 sets of experimental data
- Non-smooth and non-unique data points
- Calibrate model to fit engineering stress-strain curve up to peak load



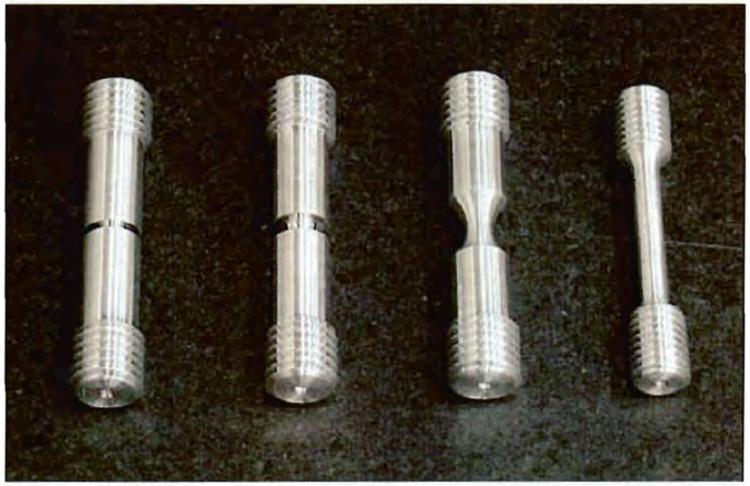
Tensile specimen: Notched and smooth

Notched Tensile Test for 6061-T651 Aluminum


original notched tensile test data


Digitalized data for notched tensile tests

- 4 different notch radius
- original figure is digitalized
- Notched tensile test used for calibration, also “validation”.


Tensile specimen: Notched and smooth

Model Calibration

Parameters to calibrate in the damage model

Besides elastic constants, for shear-modified Gurson model, the following parameters need to be calibrated

- the matrix strain hardening parameters Y_0, N ← Smooth tensile test
- the macroscopic yield surface coefficients q_1, q_2, q_3 ← keep constant
- initial void volume fraction f_0 ← keep constant
- void nucleation law f_N, ϵ_N, s_N ← Notched tensile
- shear damage parameter k_ω ← Shear test
- mesh size D_0 ← mesh refinement study

Optimization-based model calibration

- Define objective function

$$f(p_1, ..., p_N) = \frac{1}{2}\sum_{i=1}^{n}[\underset{\text{response (simulation)}}{F_i(p_1, ..., p_N)} - \underset{\text{target (experiment)}}{\bar{F}_i}]^2$$

- Find the parameter set (p1,…,pN) that minimizes the objective function (maximize the agreement between simulation response and experiment)

Model calibration

Optimization-based model calibration

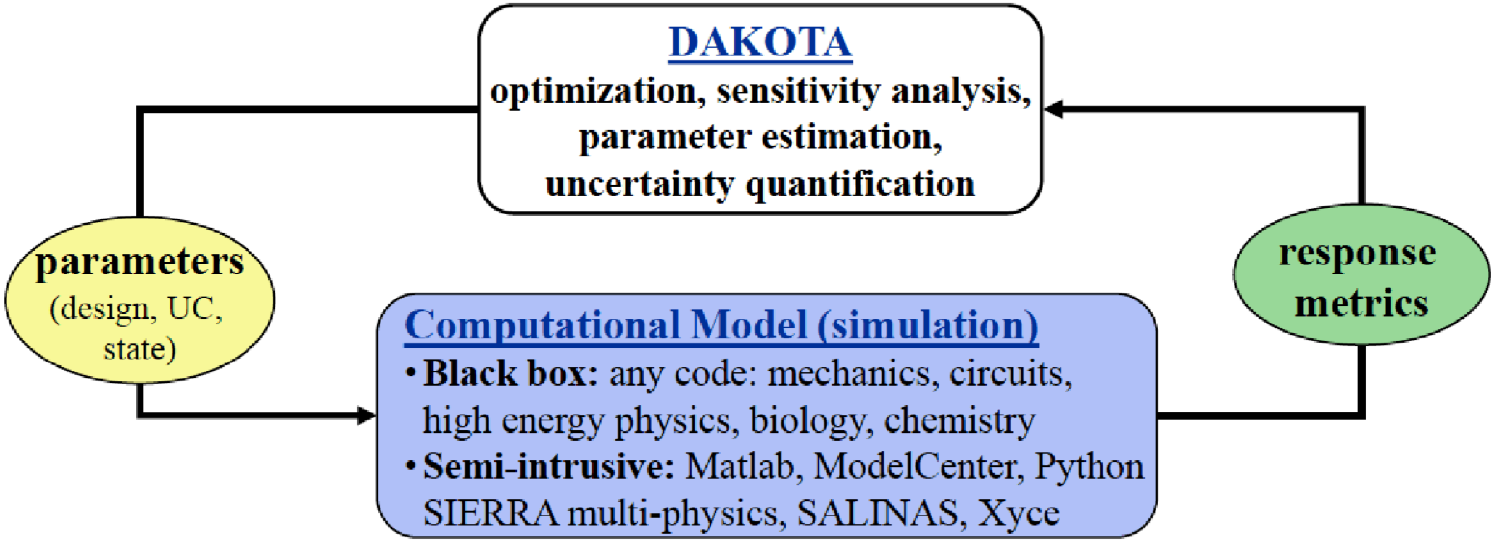
- Find the parameter set (p1,…,pN) that minimizes the objective function (maximize the agreement between simulation response and experiment)

$$f(p_1, ..., p_N) = \frac{1}{2}\sum_{i=1}^{n}[F_i(p_1, ..., p_N) - \bar{F}_i]^2$$

response (simulation)　　target (experiment)

- Solution of the optimization problem requires iterative process
- Requires computing derivative of objective function w.r.t. parameter sets (Use FAD in Albany?)

Dakota for automating the parameter variation process



DAKOTA
optimization, sensitivity analysis, parameter estimation, uncertainty quantification

parameters
(design, UC, state)

Computational Model (simulation)
- **Black box:** any code: mechanics, circuits, high energy physics, biology, chemistry
- **Semi-intrusive:** Matlab, ModelCenter, Python SIERRA multi-physics, SALINAS, Xyce

response metrics

Model calibration

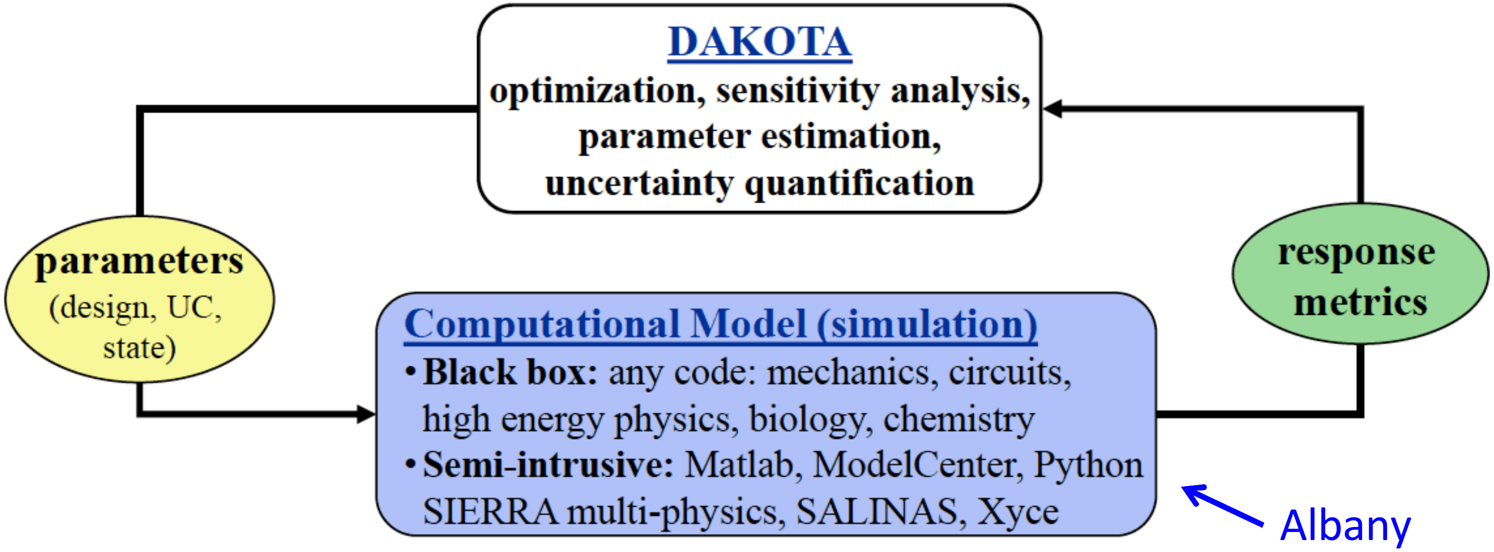
Optimization-based model calibration

- Find the parameter set (p1,…,pN) that minimizes the objective function (maximize the agreement between simulation response and experiment)

$$f(p_1,\dots,p_N)=\frac{1}{2}\sum_{i=1}^{n}[\underset{\substack{\uparrow\\ \text{response (simulation)}}}{F_i}(p_1,\dots,p_N)-\underset{\substack{\uparrow\\ \text{target (experiment)}}}{\bar{F}_i}]^2$$

- Solution of the optimization problem requires iterative process
- Requires computing derivative of objective function w.r.t. parameter sets

Dakota for automating the parameter variation process



DAKOTA
optimization, sensitivity analysis, parameter estimation, uncertainty quantification

parameters
(design, UC, state)

Computational Model (simulation)
- **Black box:** any code: mechanics, circuits, high energy physics, biology, chemistry
- **Semi-intrusive:** Matlab, ModelCenter, Python SIERRA multi-physics, SALINAS, Xyce

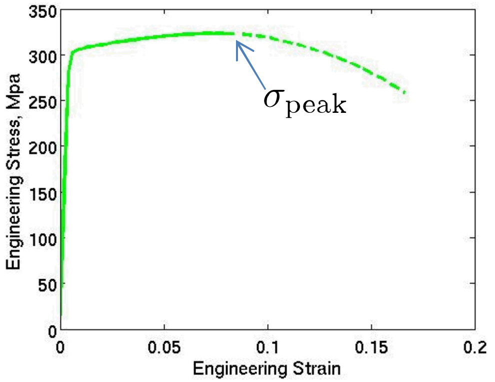
response metrics

Albany

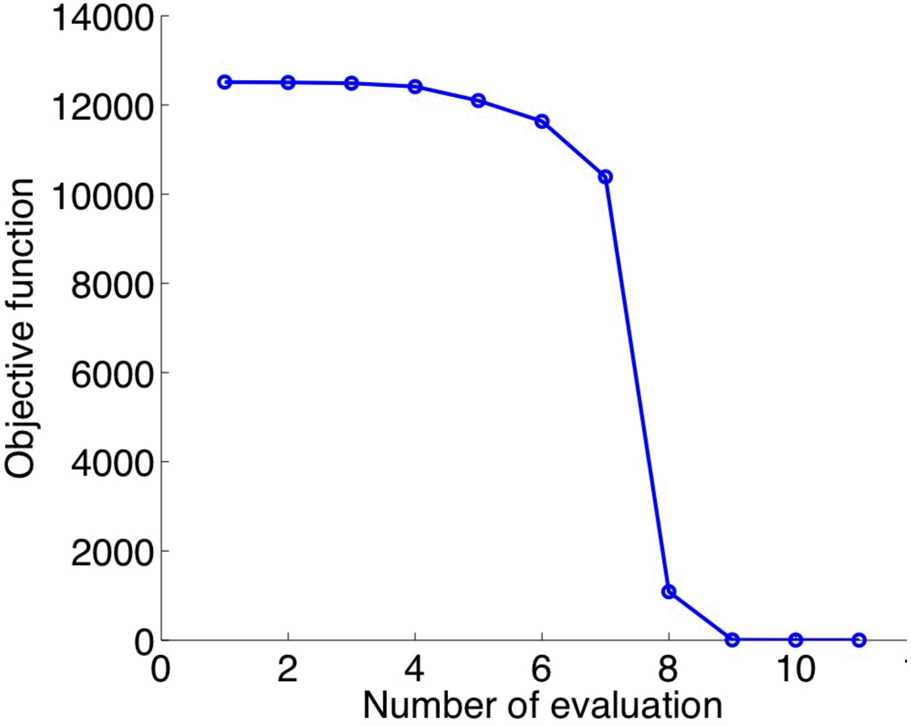
Calibration of the matrix strain hardening parameters

- Calibration up to peak load
- Dakota Least-square method
- Numerical gradient computed by finite difference

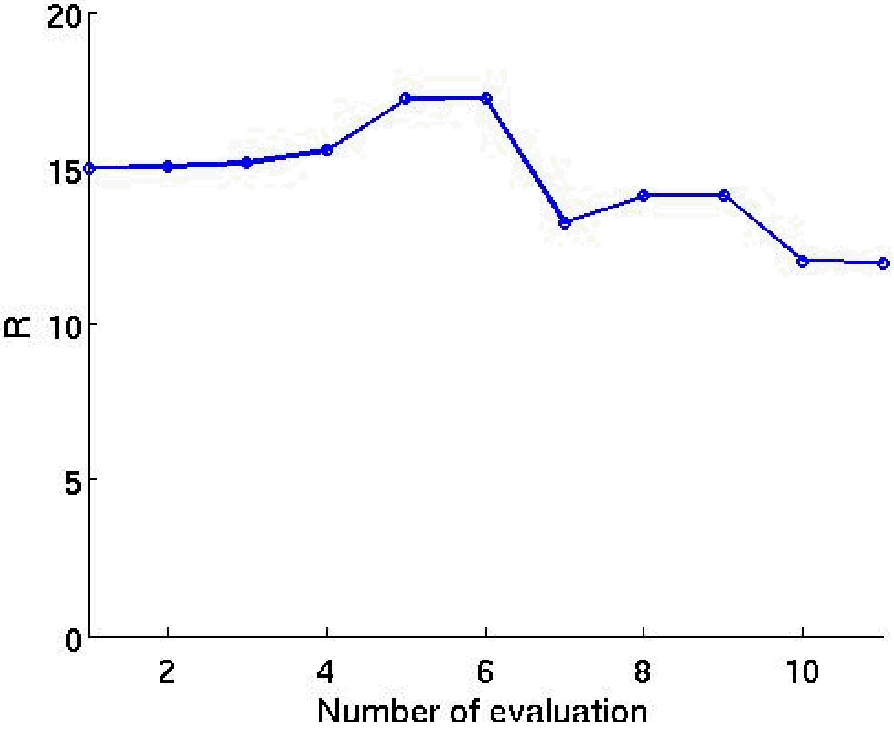
$$f(p_1, ..., p_N) = \frac{1}{2}\sum_{i=1}^{n}[F_i(p_1, ..., p_N) - \bar{F}_i]^2$$



objective function vs. iterations

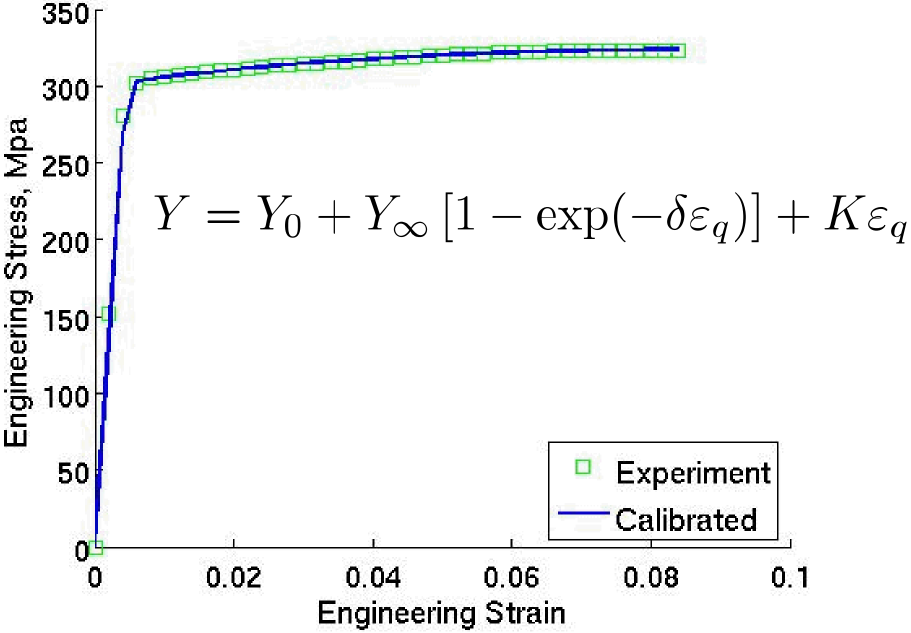


parameter R vs. iterations

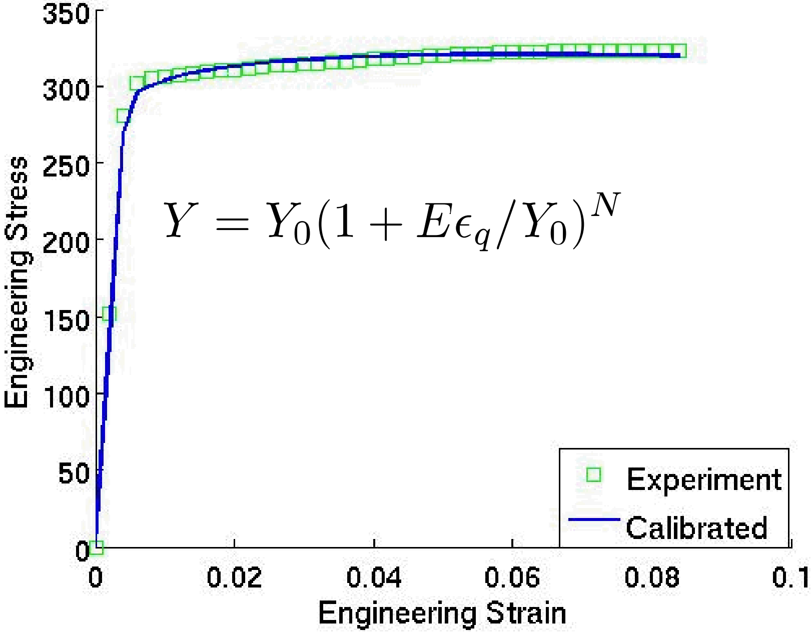


hardening minus recovery

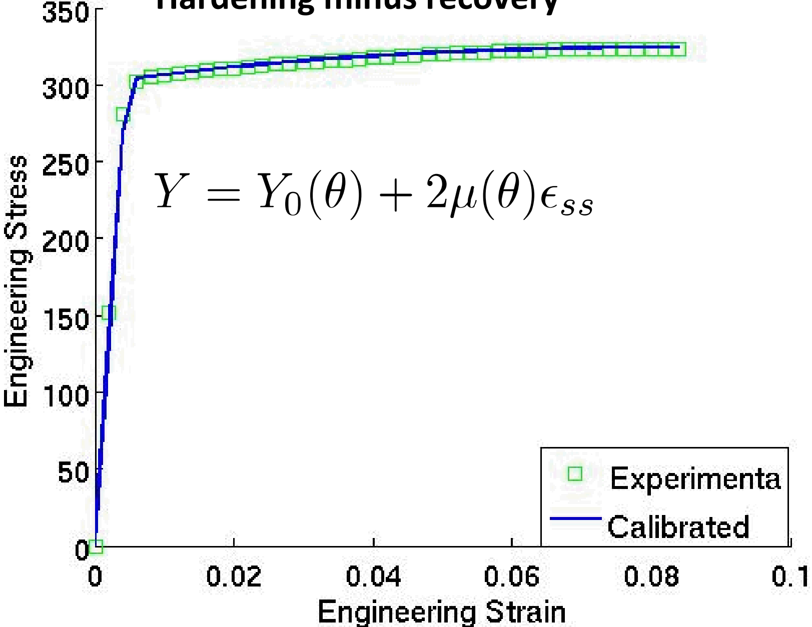
Calibration the matrix strain hardening parameters



saturation type hardening



power-law hardening

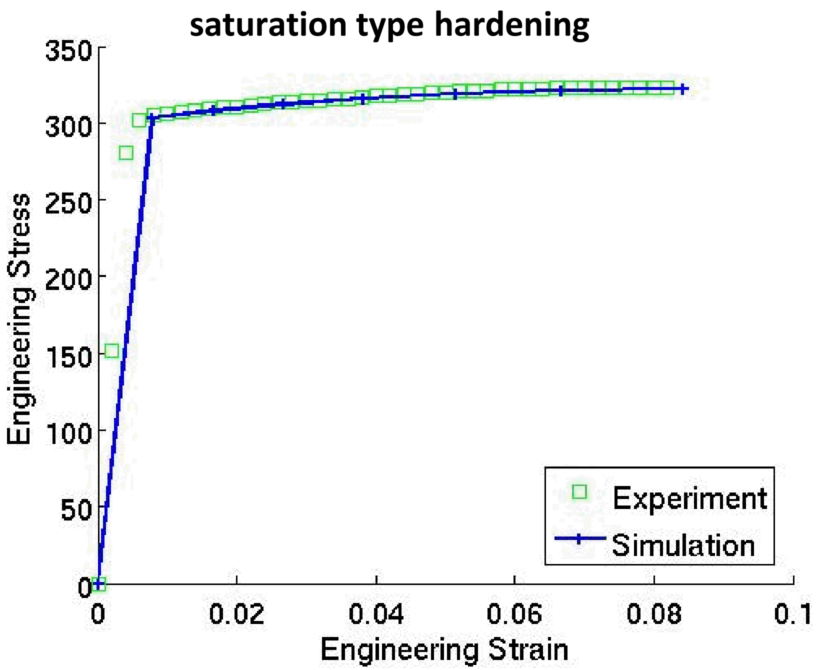


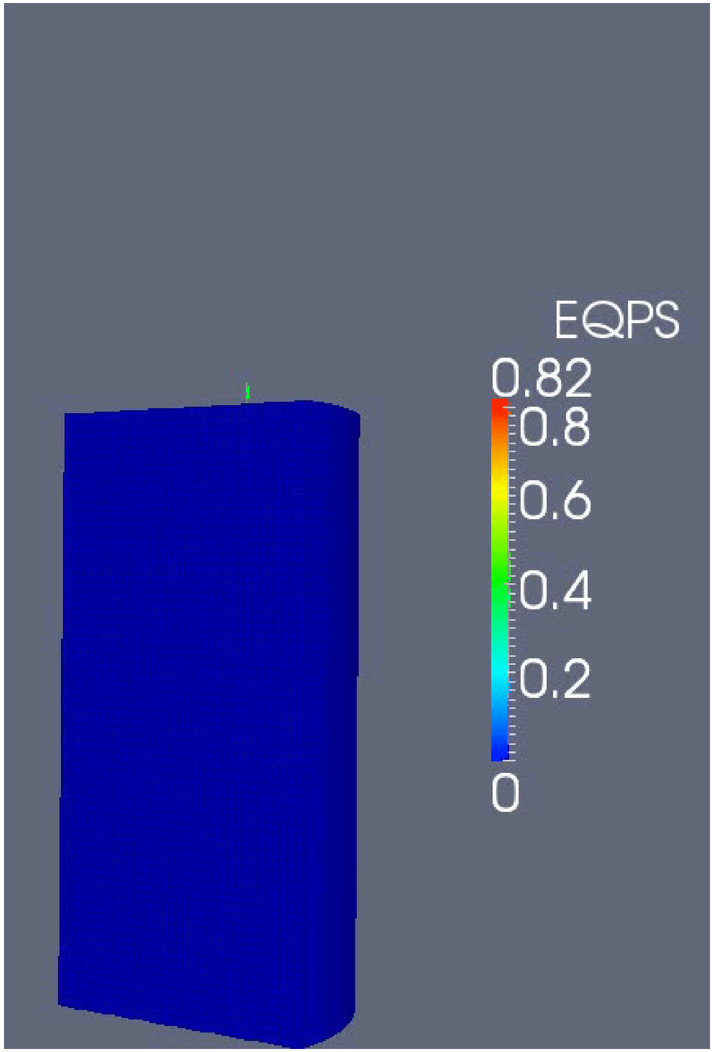
Hardening minus recovery

- Single element calibration
- different hardening laws are calibrated

Smooth tensile tests

- Verify the calibrated parameters from single element calibration
- Due to symmetry, only one-eighth of the sample is simulated
- Damage not included (does not affect pre-peak response)





material parameters

calibrated:

$Y_0 = 303.3$ MPa

$Y_\infty = 376.9$ MPa

$K = 30.4$ MPa

$\delta = 12.4$

constant:

$f_0 = 0$

$k_\omega = 0$

$f_N = 0$

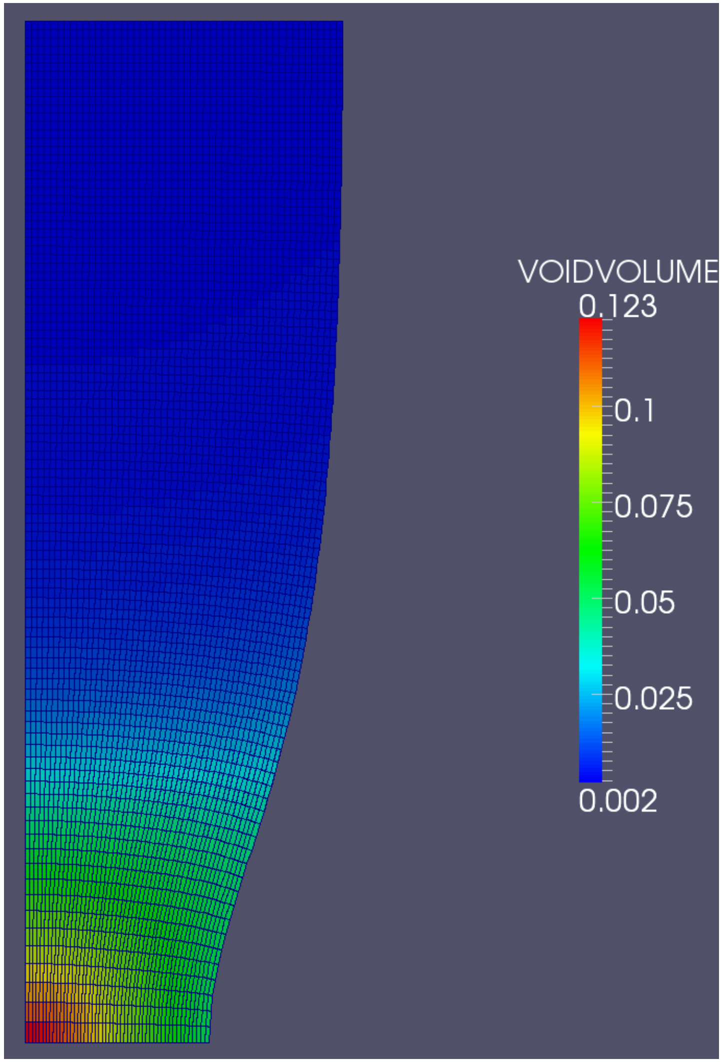
$q_1 = 1.5$

$q_2 = 1.0$

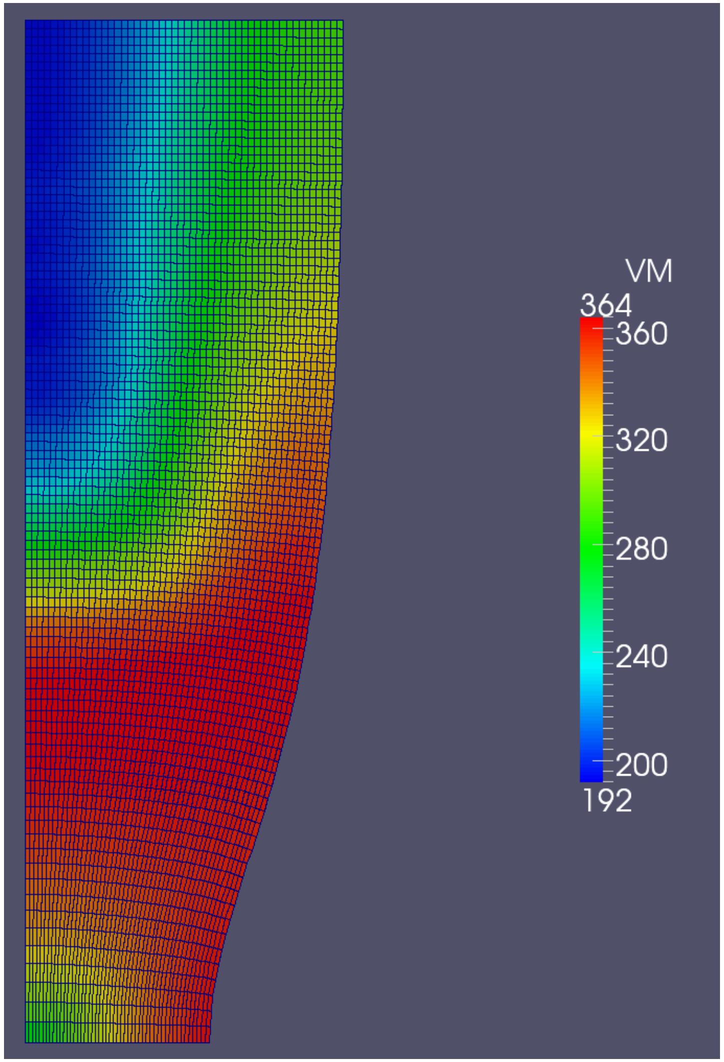
$q_3 = 2.25$

Smooth tensile tests

damage parameter

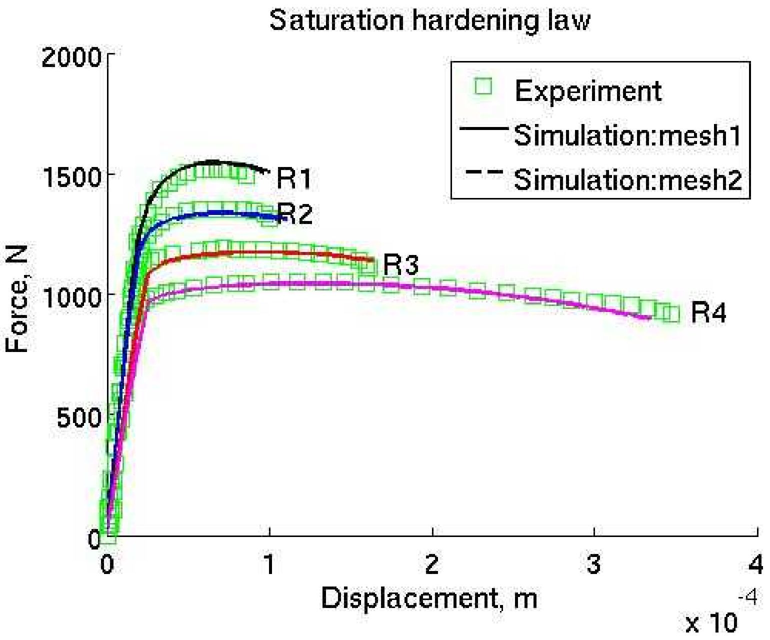


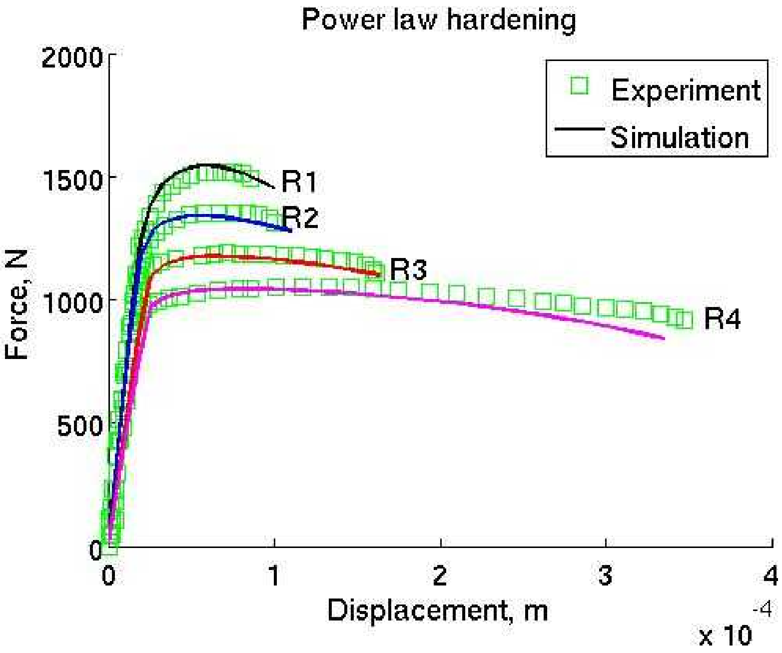
Von Mises stress



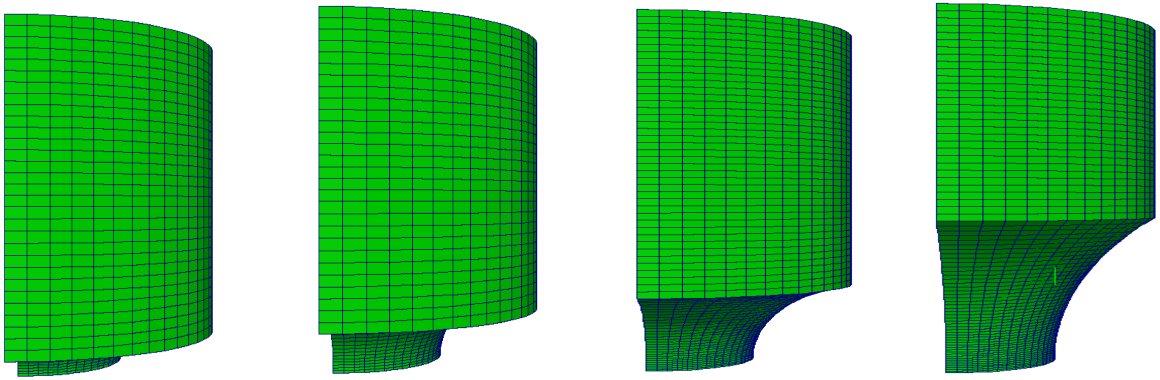
Notched Tensile Tests

- Introduce damage in the model (use constant from literature)
- Use calibrated hardening law from smooth tensile test (“validate”)
- Four different notch radius, one eighth of the sample modeled





FE mesh for four different radius



damage parameters

$$f_0 = 0.002$$

$$k_\omega = 0$$

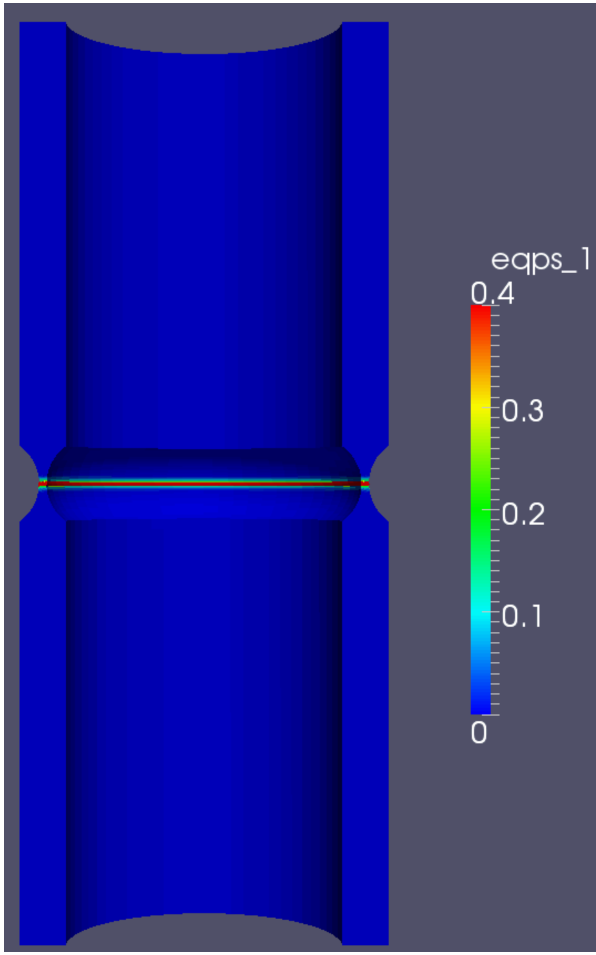
$$f_N = 0.04$$

Remarks and ongoing work

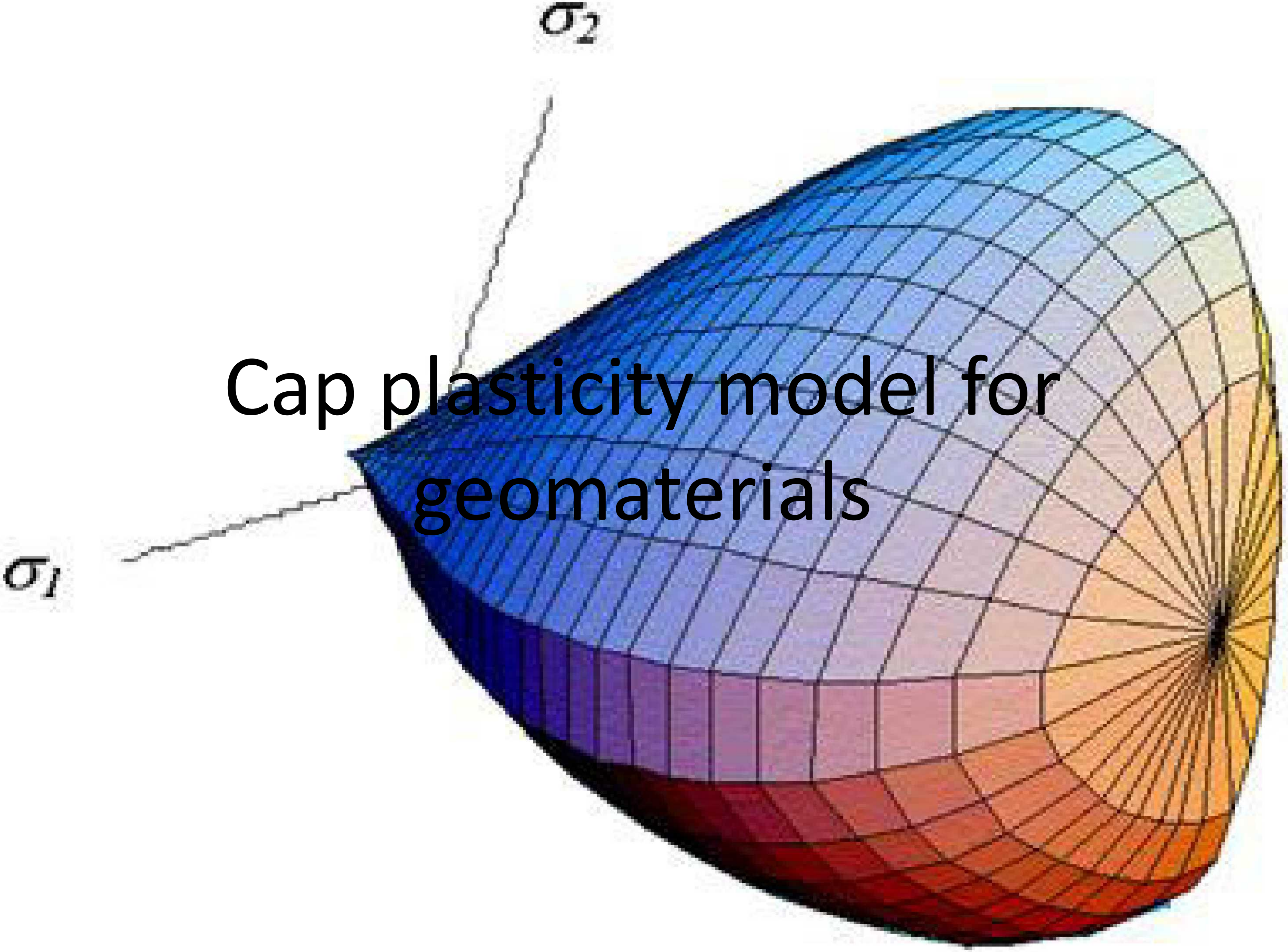
- Develop and implement a hyper-elastic formulation of the shear-modified Gurson model
- Use ***FAD*** and ***LocalNonlinearSolver*** assisting the implementation of the model
- Use ***Dakota*** to automate process of optimization-based parameter calibration
- Parameter Calibration:
  - Parameters for matrix hardening law calibrated using single element
  - Void nucleation parameter will require multi-element (specimen-scale computation) calibration using notched tensile test
  - The optimization process may not always converge and may not converge to local minimum

- Shear damage parameter calibration requires high fidelity test: Notched Tube

Torsion Test on Notched Tube

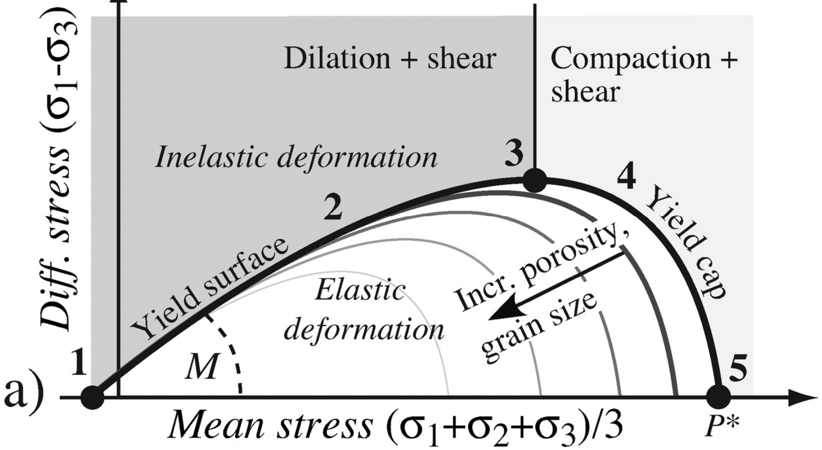


Cap plasticity model for geomaterials



Cap plasticity model

- For modeling complicated behavior of porous geomaterials, such as sandstone, limestone
- Capable of capturing shear localization in low porosity geomaterials, as well as compaction band in high porosity geomaterials
- Three invariant, isotropic and kinematic hardening, non-associative plasticity



Yield surface and plastic potential

$$f(I_1, J_2, J_3, \boldsymbol{\alpha}, \kappa) = (\Gamma(\beta))^2 J_2 - F_c(F_f - N)^2 = 0$$

$$g(I_1, J_2, J_3, \boldsymbol{\alpha}, \kappa) = (\Gamma(\beta))^2 J_2 - F_c^g(F_f^g - N)^2$$

where the exponential shear failure surface

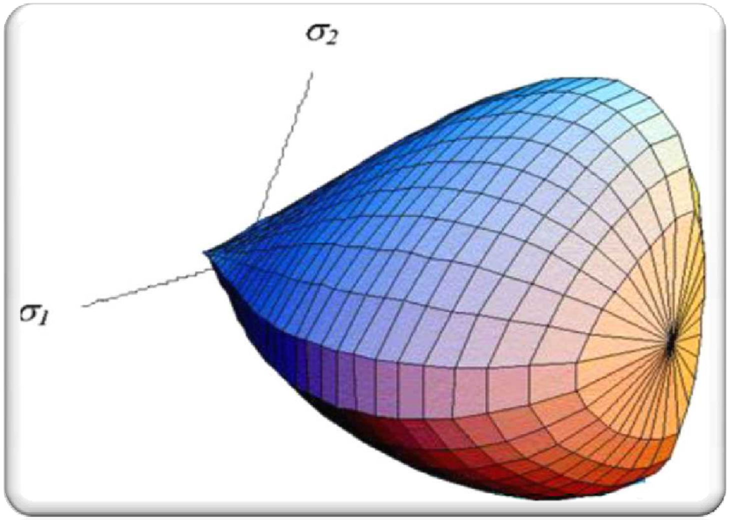
$$F_f(I_1) = A - C\exp(BI_1) - \theta I_1$$

$$F_f^g(I_1) = A - C\exp(LI_1) - \phi I_1$$

$\Gamma(\beta)$ is a function of the Lode angle, and takes into account differences in compression and extension

$$\beta = -\frac{1}{3}\sin^{-1}\left(\frac{3\sqrt{3}J_3}{2(J_2)^{3/2}}\right)$$

$$\Gamma(\beta) = \frac{1}{2}\left(1 + \sin 3\beta + \frac{1}{\psi}(1 - \sin 3\beta)\right)$$



Cap plasticity model

F_c provides an smooth elliptical cap to the yield function

$$F_c(I_1, \kappa) = 1 - H(\kappa - I_1) \left(\frac{I_1 - \kappa}{X(\kappa) - \kappa} \right)^2$$

where function X (the intersection of the cap surface with the mean stress axis in median plane)

$$X(\kappa) = \kappa - RF_f(\kappa)$$

Evolution laws for kinematic hardening

$$\dot{\boldsymbol{\alpha}} = \dot{\gamma} \boldsymbol{h}^\alpha(\boldsymbol{\alpha})$$

$$\boldsymbol{h}^\alpha = c^\alpha G^\alpha(\boldsymbol{\alpha}) \, \mathrm{dev}(\frac{\partial g}{\partial \boldsymbol{\sigma}})$$

where G is a function limits the growth of the back stress as it approaches the failure surface

$$G^\alpha(\boldsymbol{\alpha}) = 1 - \frac{\sqrt{J_2^\alpha}}{N}, \quad J_2^\alpha = \frac{1}{2} \boldsymbol{\alpha} : \boldsymbol{\alpha}$$

Evolution laws for isotropic hardening

$$\dot{\kappa} = \dot{\gamma} h^\kappa(\kappa)$$

$$h^\kappa = \frac{\mathrm{tr}(\partial g / \partial \boldsymbol{\sigma})}{\partial \epsilon_v^p / \partial \kappa}$$

where the following form of volumetric strain is used

$$\epsilon_v^p = W(\exp[D_1(X(\kappa) - X_0) - D_2(X(\kappa) - X_0)^2] - 1)$$

Numerical implementation

- Both implicit and explicit scheme has been used to integrate the cap plasticity model
- Explicit scheme:
  - an normal stress correction algorithm has been implemented to prevent stress drifting from yield surface.
  - no local system of equation needs to be solved.
  - smaller time steps are generally required
- Implicit scheme:
  - require iterative solution of local system of equation
  - larger time steps can be used

Local unknown vector (13x1)

$$\boldsymbol{X} = \{\boldsymbol{\sigma}, \boldsymbol{\alpha}, \kappa, \Delta\gamma\}$$

Local nonlinear system of equations (13x1)

Remarks

Forward Automatic Differentiation (FAD) is used to obtain local Jacobian matrix. **LocalNonlinearSolver** is used to solve the linearized equation, and upon convergence, compute the system sensitivity information.

example code from CapImplicit_Def.hpp using FAD and NonLinearSolver

```
// initialize local unknown vector
 X[0] = sigmaVal(0, 0); X[1] = sigmaVal(1, 1); X[2] = sigmaVal(2, 2);
 X[3] = sigmaVal(1, 2); X[4] = sigmaVal(0, 2); X[5] = sigmaVal(0, 1);
 X[6] = alphaVal(0, 0); X[7] = alphaVal(1, 1); X[8] = alphaVal(1, 2);
 X[9] = alphaVal(0, 2); X[10] = alphaVal(0, 1);
 X[11] = kappaVal; X[12] = dgammaVal;

{// local Newton-Raphson loop

for (int i = 0; i < 13; ++i) {
   XVal[i] = Sacado::ScalarValue<ScalarT>::eval(X [i]);
   Xfad[i] = DFadType(13, i, XVal[i]);
 }

 :
 :
 // local system of equations (13 x 1)
 :
 // get ScalarT Residual
 for (int i = 0; i < 13; i++)
   R[i] = Rfad[i].val();

 // get local Jacobian
 for (int i = 0; i < 13; i++)
   for (int j = 0; j < 13; j++)
     dRdX[i + 13 * j] = Rfad[i].dx(j);

 if (kappa_flag == true) {
   for (int j = 0; j < 13; j++)
     dRdX[11 + 13 * j] = 0.0;
   dRdX[11 + 13 * 11] = 1.0;
 }
 :
 :
 // call LocalNonlinearSolver
 solver.solve(dRdX, X, R);
} // end local Newton loop

 // compute sensitivity information w.r.t system parameters, and pack back to X
 solver.computeFadInfo(dRdX, X, R);
```

← “unpack” system sensitivity information

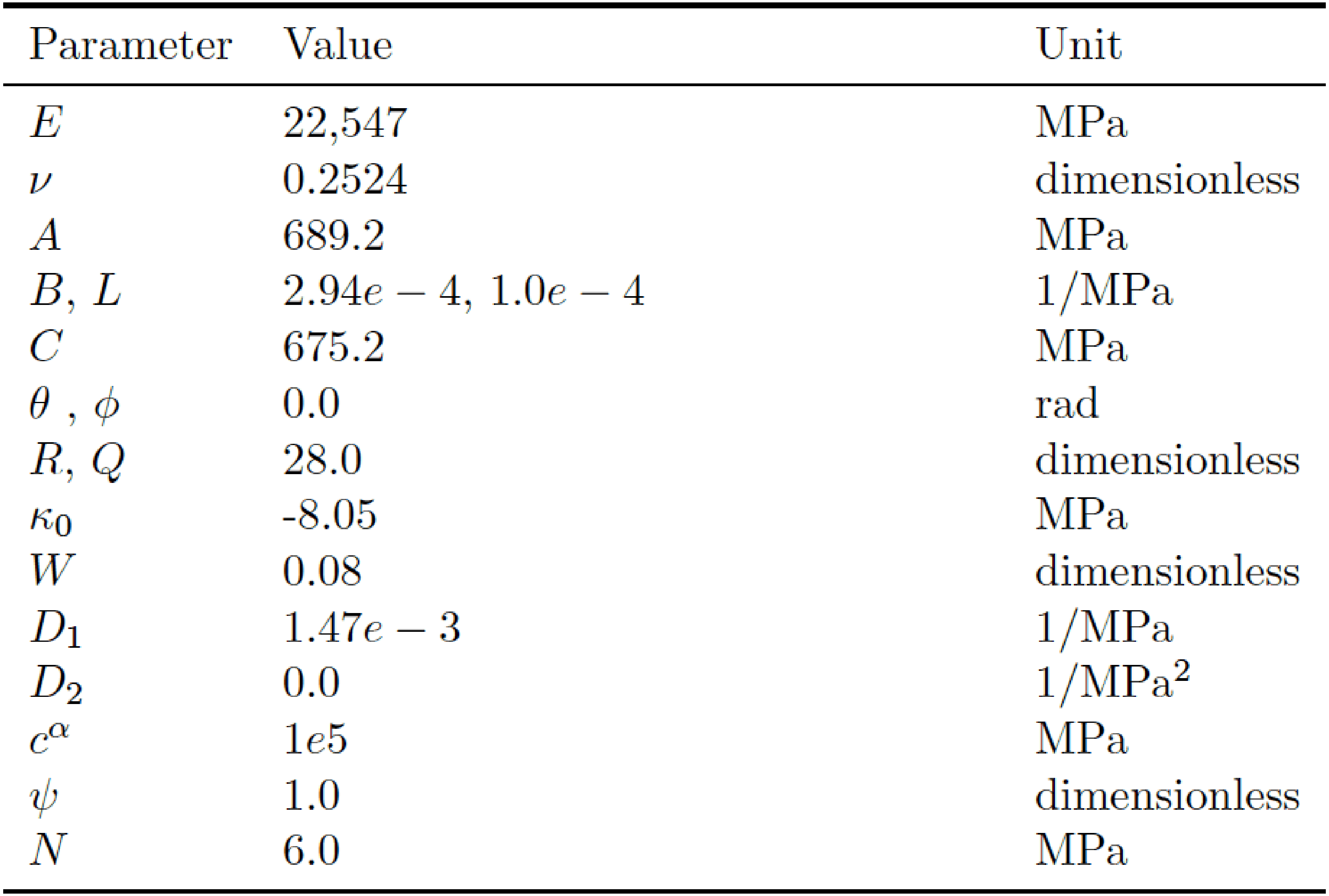
} local Jacobian matrix (13x13)

$\boldsymbol{J} = \partial \boldsymbol{R}/\partial \boldsymbol{X}$

← LocalNonlinearSolver

← compute system sensitivity

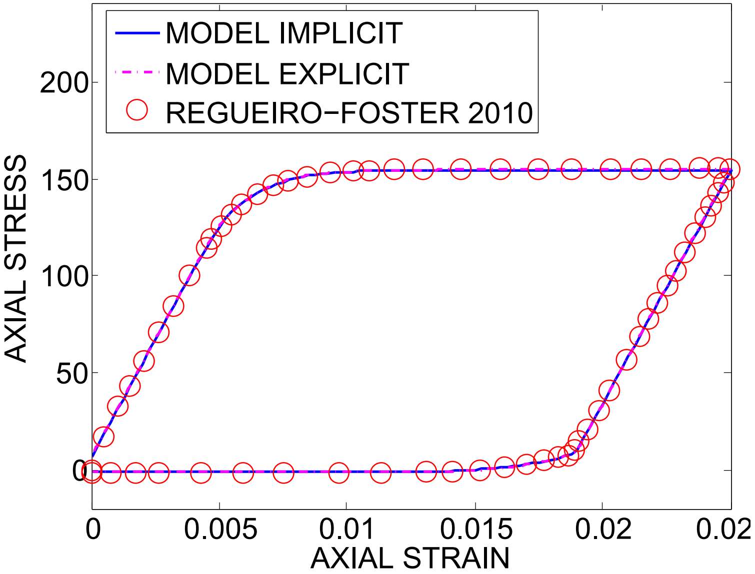
Material Parameters for Salem Sandstone



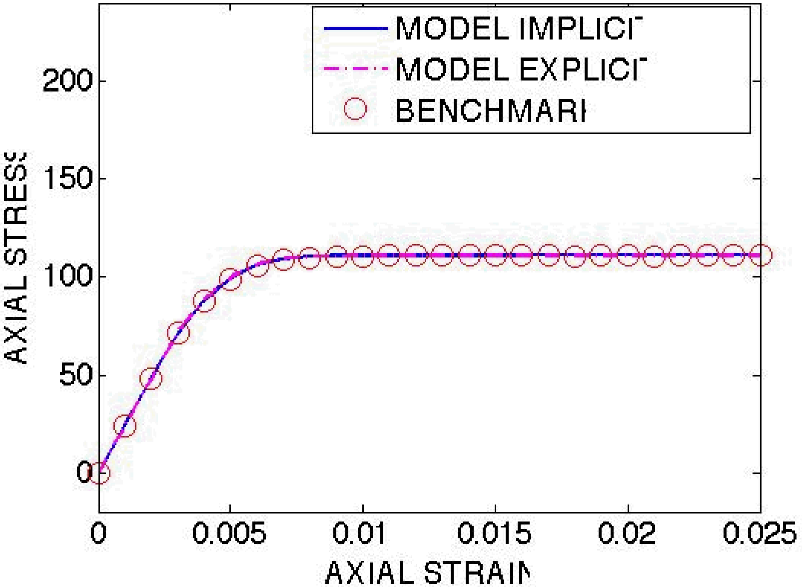
Parameter	Value	Unit
E	22,547	MPa
ν	0.2524	dimensionless
A	689.2	MPa
B, L	$2.94e-4$, $1.0e-4$	1/MPa
C	675.2	MPa
θ , ϕ	0.0	rad
R, Q	28.0	dimensionless
κ_0	-8.05	MPa
W	0.08	dimensionless
D_1	$1.47e-3$	1/MPa
D_2	0.0	$1/\mathrm{MPa}^2$
c^α	$1e5$	MPa
ψ	1.0	dimensionless
N	6.0	MPa

Verification: element tests

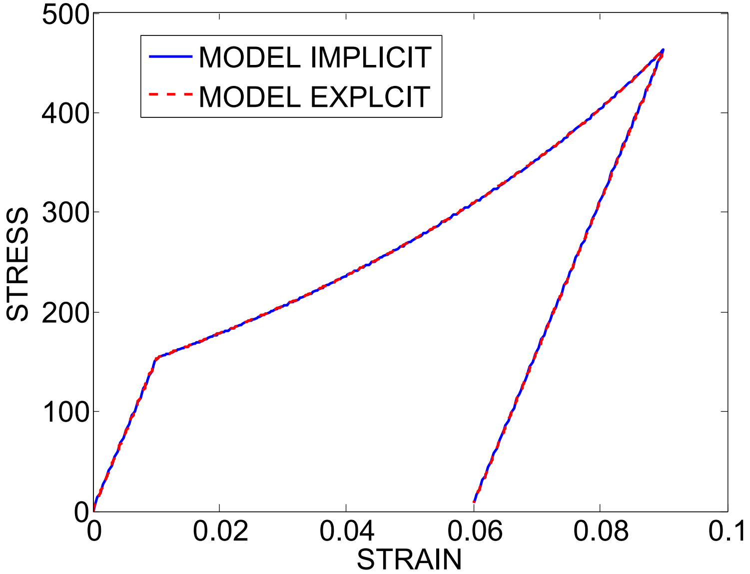
plane strain compression



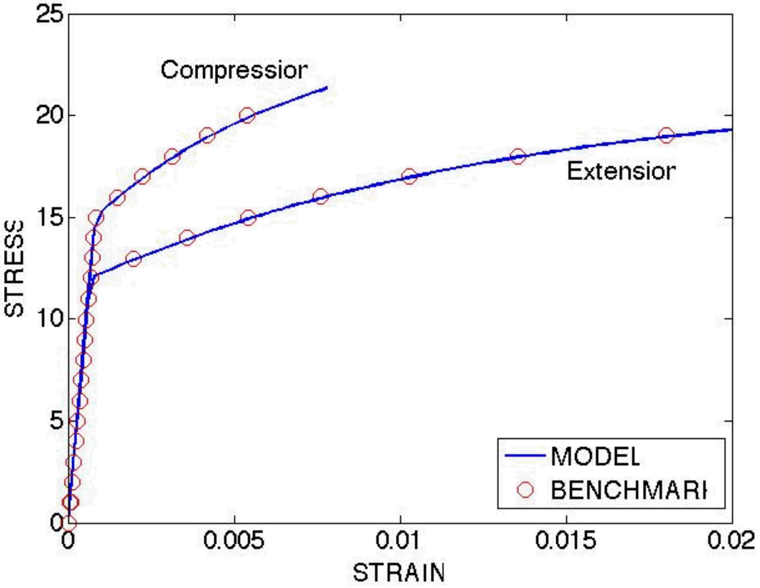
plane stress



volumetric extension



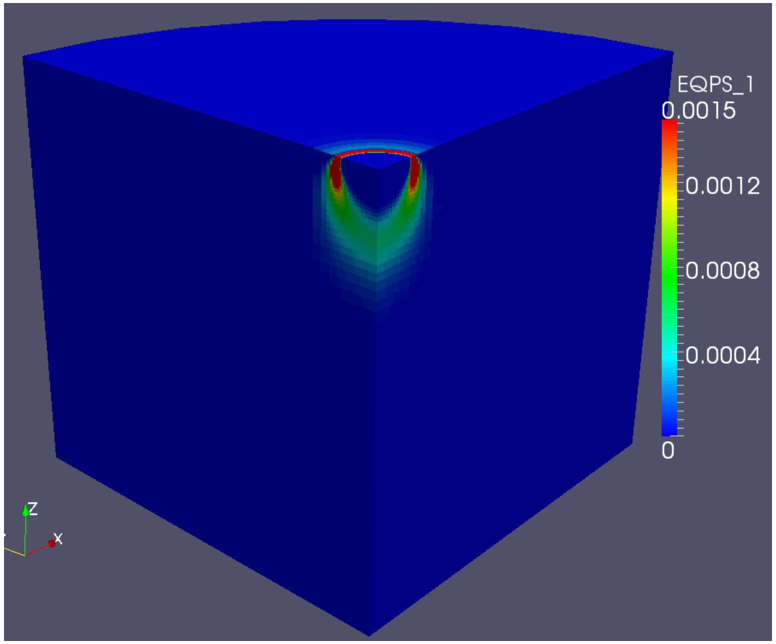
Triaxial compression and extension



3D Penetration Problem on Salem Limestone

Couple cap plasticity model with poro-mechanical problem in LCM

eqps contour



pore-pressure evolution