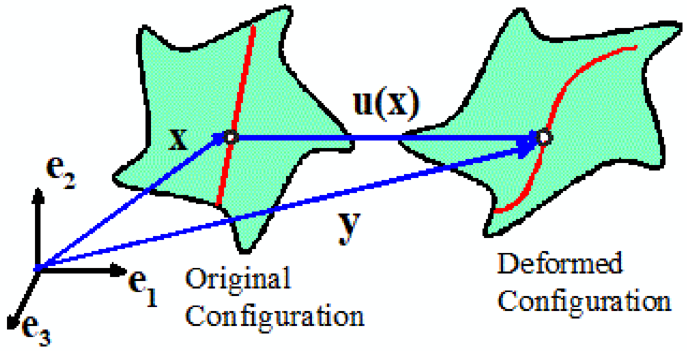
Review – Newmark time integration

Equation of motion $\mathbf{M}\ddot{\mathbf{u}} = -\mathbf{R}(\mathbf{u}) + \mathbf{F}(t)$



Integrate WRT time using time stepping:

(1) Initialize $\mathbf{u}^0 = \mathbf{u}(t=0) \qquad \mathbf{v}^0 = \frac{d\mathbf{u}}{dt}(t=0)$

$$\mathbf{a}^0 = -\mathbf{R}(\mathbf{u}^0) + \mathbf{F}(0)$$

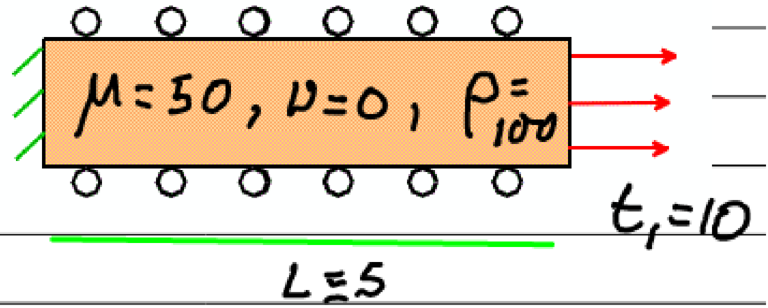
Newmark Loop

(1) Solve $\mathbf{M}\mathbf{a}^{n+1} = -\mathbf{R}\left(\mathbf{u}^n + \Delta t\mathbf{v}^n + \frac{\Delta t^2}{2}\left[(1-\beta_2)\mathbf{a}^n + \beta_2\mathbf{a}^{n+1}\right]\right) + \mathbf{F}(t_{n+1})$

(2) Update $\mathbf{v}^{n+1} = \mathbf{v}^n + \Delta t\left[(1-\beta_1)\mathbf{a}^n + \beta_1\mathbf{a}^{n+1}\right]$

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\mathbf{v}^n + \frac{\Delta t^2}{2}\left[(1-\beta_2)\mathbf{a}^n + \beta_2\mathbf{a}^{n+1}\right]$$

Example Problem



1-D Bar @ rest & stress free for $t < 0$
Constant $t_{11} = 10$ acts on $x = L$
for $t > 0$

$$c = \sqrt{\frac{2\mu(1-\nu)}{\rho(1-2\nu)}} \quad \text{(wave speed)}$$

Stress pulse propagates down bar @ speed c

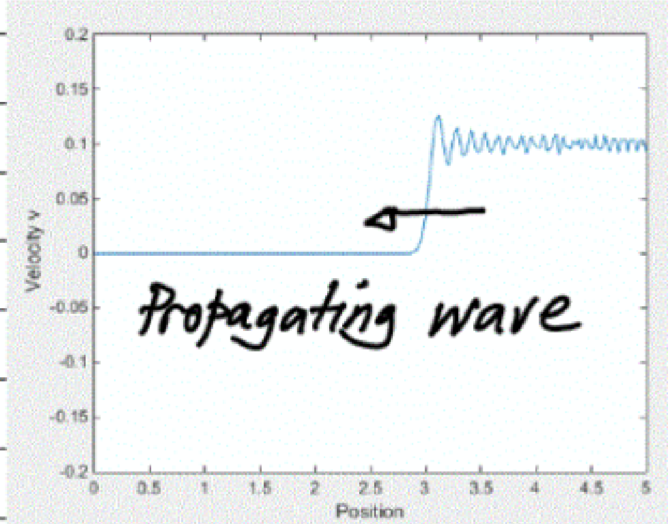
Behind stress pulse $\sigma_{11} = 10$; velocity = constant

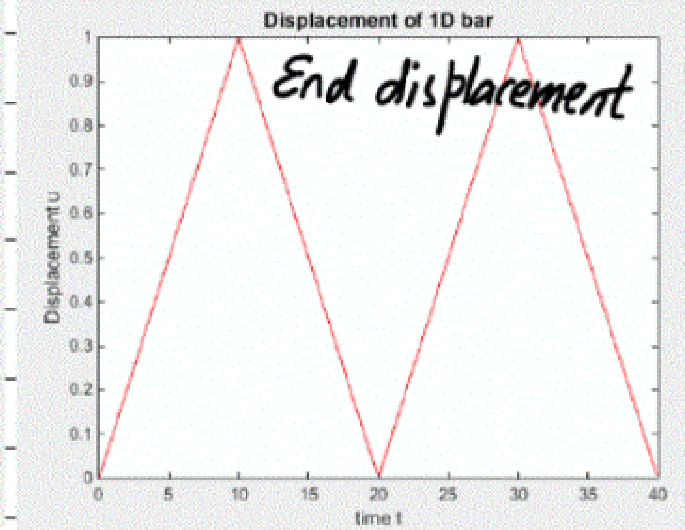
Reflections off both ends

Solve with 1D MATLAB

Solution with $\beta_2 = 0$ $\beta_1 = \frac{1}{2}$
$\Delta t = 0.004$

Lumped mass matrix

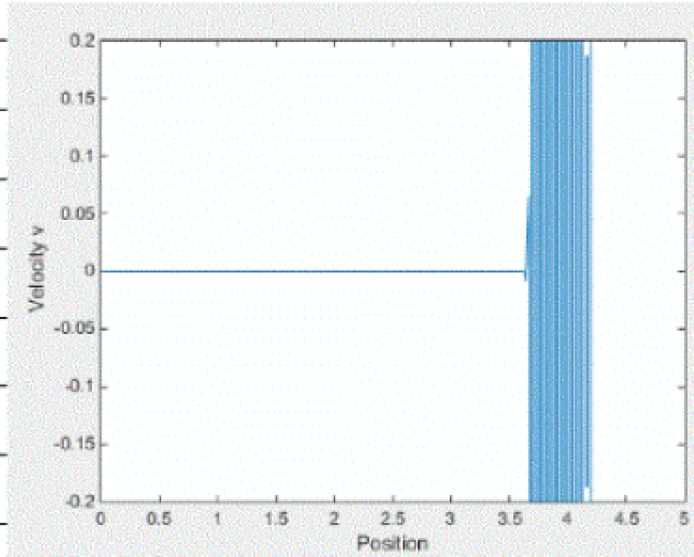


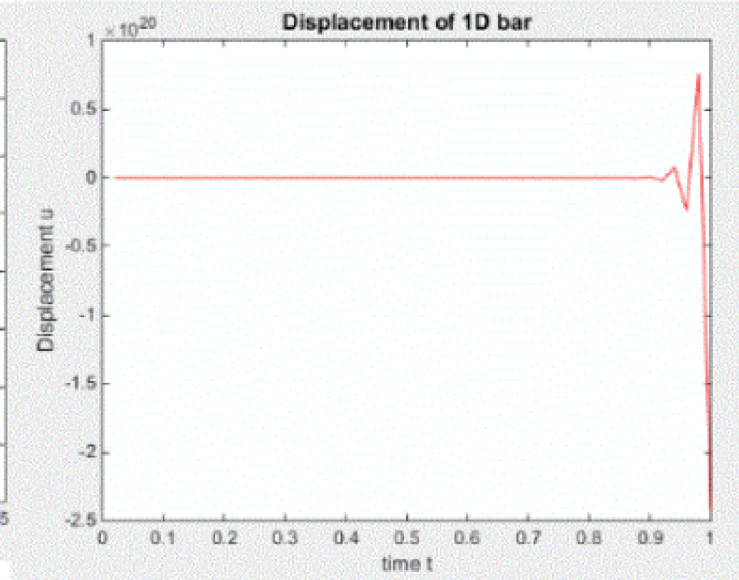


Solution with $\beta_2 = 0$
$\beta_1 = \frac{1}{2}$ $\Delta t = 0.02$

Numerical Instability

Lumped mass matrix

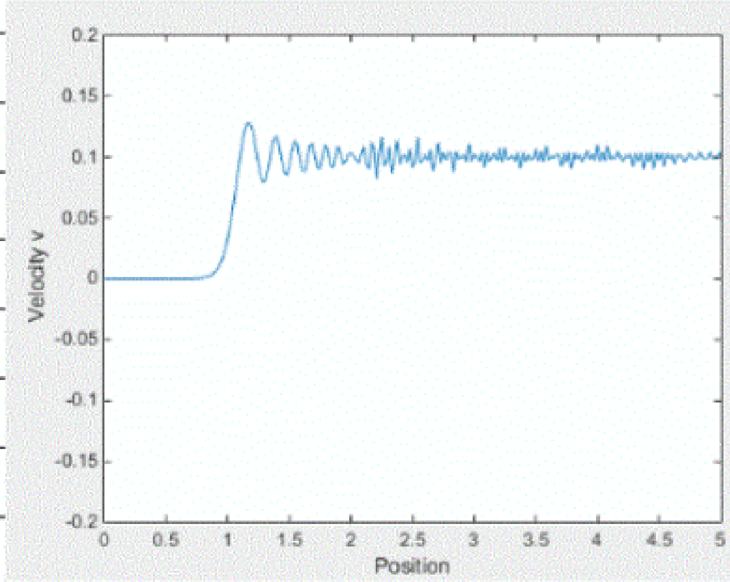


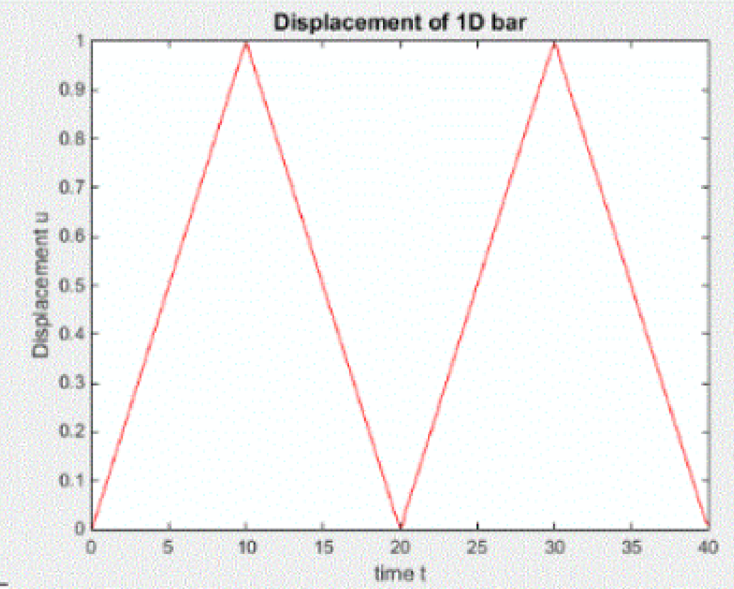


Solution with $\beta_1 = \beta_2 = \frac{1}{2}$

$\Delta t = 0.02$

Consistent mass matrix



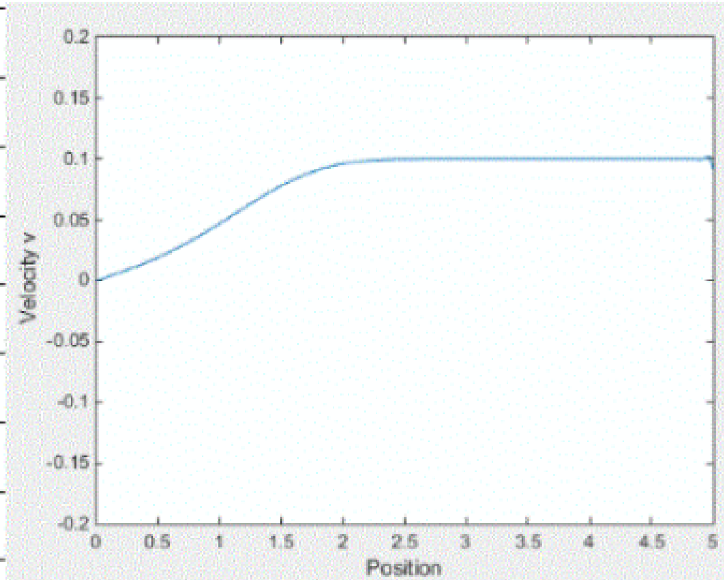


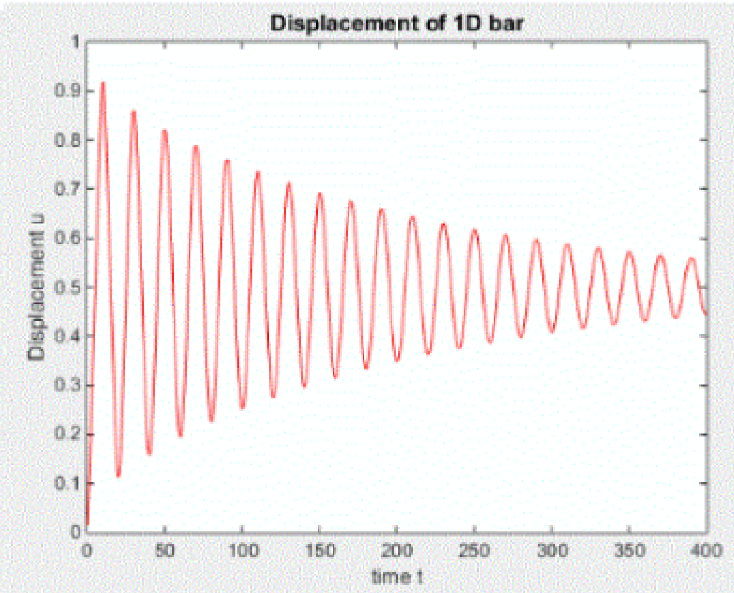
Solution with $\beta_2 = \beta_1 = 1$

$\Delta t = 0.2$

Consistent mass matrix

Stable but energy not conserved





Summary of tests

Explicit Dynamics $(\beta_2 = 0)$

(1) Conditionally stable – for stability $\beta_1 \geq 0.5$
$\Delta t < \Delta t_{crit}$

Lumped mass -v- consistent mass has negligible effect

Implicit dynamics $\beta_1 > 0.5$ $\beta_2 \geq \beta_1$.

– Unconditionally stable
Need small Δt for accuracy

Energy conserved for $\beta_1 = \beta_2 = 0.5$

Explicit dynamics in ABAQUS

Acceleration update for explicit dynamics has form

$$[M]\,\underline{a}^{n+1} = \underline{F}(t+\Delta t) - \underline{R}[\underline{u}^n]$$

$$\underline{R} = \int_{R_0} \underbrace{[B]^T}_{\text{Usual } [B] \text{ or } [\bar{B}]} \underbrace{\underline{\sigma}(\Delta \xi)}_{\text{Cauchy stress}} J\, dV_0$$

Note that only $\underline{\sigma}$ is material dependent
- No need for a $[D]$ matrix

ABAQUS provides the "VUMAT" subroutine to calculate $\underline{\sigma}$

Input Variables:

State variables
Stress
Def grads
at time t_n

$\Delta \underline{\varepsilon}$ during Δt

Output variables

Stress (Cauchy)
State vars
@ $t+\Delta t$

```fortran
      subroutine vumat(nblock, ndir, nshr, nstatev, nfieldv, nprops,
     1  lanneal, stepTime, totalTime, dt, cmname, coordMp, charLength,
     2  props, density, strainInc, relSpinInc,
     3  tempOld, stretchOld, defgradOld, fieldOld,
     4  stressOld, stateOld, enerInternOld, enerInelasOld,
     5  tempNew, stretchNew, defgradNew, fieldNew,
     6  stressNew, stateNew, enerInternNew, enerInelasNew )

       include 'vaba_param.inc'

      implicit double precision (a-h,o-z)

      dimension props(nprops), density(nblock), coordMp(nblock,*),
     1  charLength(nblock), strainInc(nblock,ndir+nshr),
     2  relSpinInc(nblock,nshr), tempOld(nblock),
     3  stretchOld(nblock,ndir+nshr),
     4  defgradOld(nblock,ndir+nshr+nshr),
     5  fieldOld(nblock,nfieldv), stressOld(nblock,ndir+nshr),
     6  stateOld(nblock,nstatev), enerInternOld(nblock),
     7  enerInelasOld(nblock), tempNew(nblock),
     8  stretchNew(nblock,ndir+nshr),
     9  defgradNew(nblock,ndir+nshr+nshr),
     1  fieldNew(nblock,nfieldv),
     2  stressNew(nblock,ndir+nshr), stateNew(nblock,nstatev),
     3  enerInternNew(nblock), enerInelasNew(nblock)
```

Note VUMAT & UMAT use different conventions
for many quantities

VUMAT conventions

(1) All quantities are provided in "blocks" of integration points

eg stressOld (k, I) gives I^{th} component of stress vector @ integration point k

User must update nblock data sets

(2) Storage for strain increment strainInc Order changed

$$\Delta \underline{\varepsilon} = [\Delta\varepsilon_{11} \quad \Delta\varepsilon_{22} \quad \Delta\varepsilon_{33} \quad \underbrace{\Delta\varepsilon_{12} \quad \Delta\varepsilon_{23} \quad \Delta\varepsilon_{13}}_{\text{No factor of 2}}]$$

(3) Stresses stored as

$$\underline{\sigma} = [\sigma_{11} \quad \sigma_{22} \quad \sigma_{33} \quad \sigma_{12} \quad \sigma_{23} \quad \sigma_{13}]$$

Example Problem: Vumat for viscoplasticity

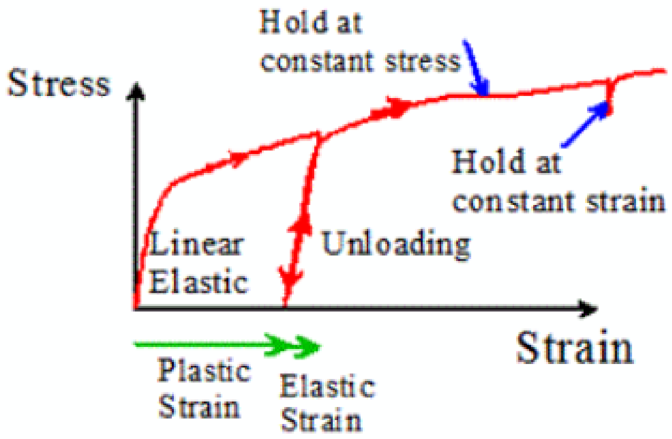
Small-strain rate dependent plasticity - same as used in statics

- Material model $\dot{\varepsilon}_{ij} = \dot{\varepsilon}_{ij}^{e} + \dot{\varepsilon}_{ij}^{p}$

$$S_{ij} = \sigma_{ij} - \frac{1}{3}\sigma_{kk}\delta_{ij} \qquad \sigma_e = \sqrt{\frac{3}{2}S_{ij}S_{ij}}$$

$$\dot{\varepsilon}_{ij}^{e} = \frac{(1+\nu)}{E}\dot{S}_{ij} + \frac{1-2\nu}{3E}\dot{\sigma}_{kk}\delta_{ij} \qquad \dot{\varepsilon}_{ij}^{p} = \dot{\varepsilon}_0\left(\frac{\sigma_e}{\sigma_0(\varepsilon_e)}\right)^m \frac{3}{2}\frac{S_{ij}}{\sigma_e}$$

$$\sigma_0(\varepsilon_e) = Y\left(1+\frac{\varepsilon_e}{\varepsilon_0}\right)^{1/n} \qquad \frac{d\varepsilon_e}{dt} = \sqrt{\frac{2}{3}\dot{\varepsilon}_{ij}\dot{\varepsilon}_{ij}} = \dot{\varepsilon}_0\left(\frac{\sigma_e}{\sigma_0}\right)^m$$



Implement the same implicit stress update that we used for statics

Viscoplastic stress update

(For a generic integration point)

Given $\{\Delta\varepsilon_{ij}, \sigma_{ij}^{n}, \varepsilon_e^{n}, \Delta t\}$ Compute $\{\sigma_{ij}^{n+1}, \varepsilon_e^{n+1}\}$

1. Compute $\Delta e_{ij} = \Delta\varepsilon_{ij} - \Delta\varepsilon_{kk}\delta_{ij}/3$ $S_{ij}^{n} = \sigma_{ij}^{n} - \sigma_{kk}^{n}\delta_{ij}/3$

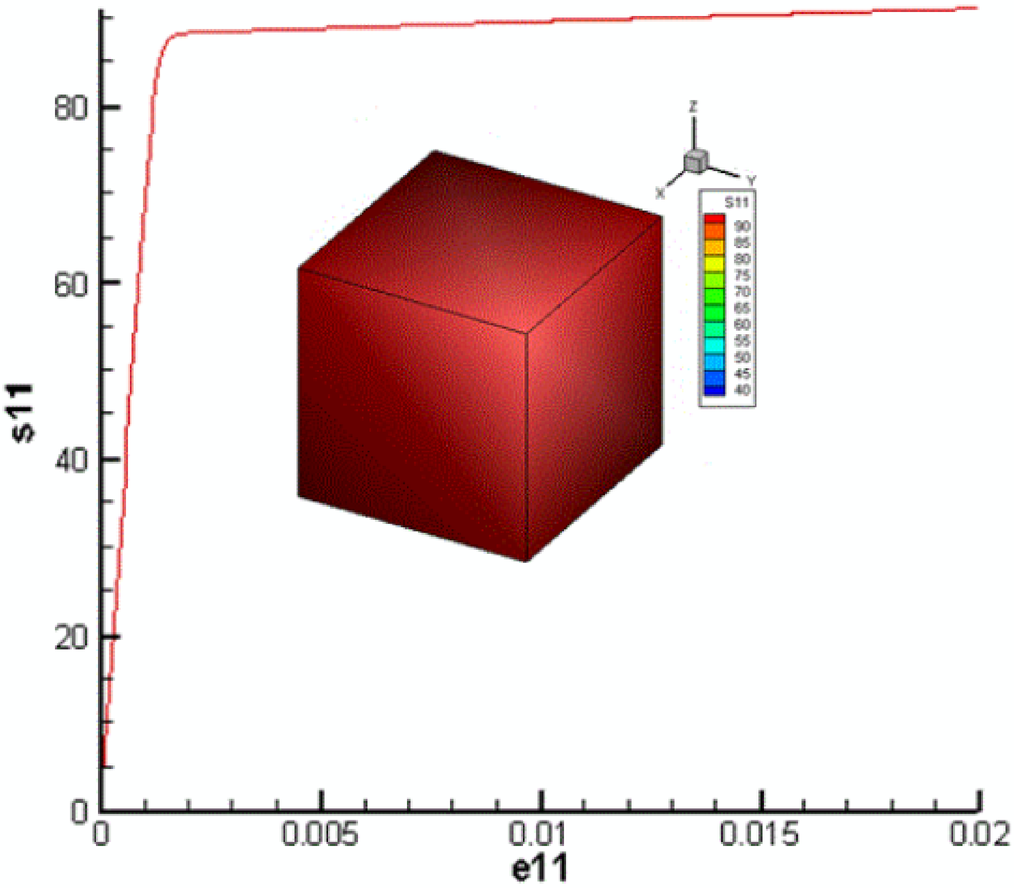
2. Compute $S_{ij}^{*} = S_{ij}^{n} + \dfrac{E}{(1+\nu)}\Delta e_{ij}$ $\sigma_e^{*} = \sqrt{3S_{ij}^{*}S_{ij}^{*}/2}$

3. Using Newton-Raphson iteration, solve for $\Delta\varepsilon_e$

$$\sigma_e^{*} - Y\left(1 + \frac{\varepsilon_e^{n} + \Delta\varepsilon_e}{\varepsilon_0}\right)^{1/n}\left(\frac{\Delta\varepsilon_e}{\Delta t\dot{\varepsilon}_0}\right)^{1/m} - \frac{3E}{2(1+\nu)}\Delta\varepsilon_e = 0$$

4. Calculate $\sigma_{ij}^{n+1} = \left(1 - \dfrac{3E\Delta\varepsilon_e}{2(1+\nu)\sigma_e^{*}}\right)S_{ij}^{*} + \left(\sigma_{kk}^{n} + \dfrac{E\Delta\varepsilon_{kk}}{(1-2\nu)}\right)\dfrac{\delta_{ij}}{3}$

Example simulation (see code online)




```
      MATERIAL, visplas_vumat
        STATE VARIABLES, 2          % Number of material state
variables (if the key is omitted the number of state vars
defaults to zero)
        PROPERTIES
         70.d03, 0.3d0             % E (MPa), nu;
         70.d0, 0.1d0, 5.d0        % Y (MPa), e0, n
         0.1d0, 10.d0               % edot0, m
        END PROPERTIES
      END MATERIAL

      DENSITY, 0.000005d0

      TIME STEP, 0.000001
      NUMBER OF STEPS, 20000
      STATE PRINT STEPS, 500
      USER PRINT STEPS, 50
```