UMAT/VUMAT subroutines

User subroutine to define a material's mechanical behaviour

Norwegian University of Science and Technology

(V)UMAT subroutine

UMAT: User subroutine to define a material's mechanical behavior.

!

WARNING: The use of this subroutine generally requires considerable expertise. You are cautioned that the implementation of any realistic constitutive model requires extensive development and testing. Initial testing on a single-element model with prescribed traction loading is strongly recommended.

VUMAT: User subroutine to define material behavior.

(V)UMAT subroutine

From the ABAQUS documentation:

ABAQUS/standard

Overview

User subroutine **`UMAT`**:

- can be used to define the mechanical constitutive behavior of a material;
- will be called at all material calculation points of elements for which the material definition includes a user-defined material behavior;
- can be used with any procedure that includes mechanical behavior;
- can use solution-dependent state variables;
- must update the stresses and solution-dependent state variables to their values at the end of the increment for which it is called;
- must provide the material Jacobian matrix, $\partial\triangle\boldsymbol{\sigma}/\partial\triangle\boldsymbol{\varepsilon}$, for the mechanical constitutive model;
- can be used in conjunction with user subroutine **`USDFLD`** to redefine any field variables before they are passed in; and
- is described further in "User-defined mechanical material behavior," Section 26.7.1 of the Abaqus Analysis User's Guide.

ABAQUS/explicit

Overview

User subroutine **`VUMAT`**:

- is used to define the mechanical constitutive behavior of a material;
- will be called for blocks of material calculation points for which the material is defined in a user subroutine ("Material data definition," Section 21.1.2 of the Abaqus Analysis User's Guide);
- can use and update solution-dependent state variables;
- can use any field variables that are passed in; and
- can be used in an adiabatic analysis, provided you define both the inelastic heat fraction and the specific heat for the appropriate material definitions and you store the temperatures and integrate them as user-defined state variables.

(V)UMAT subroutine

H are the State Dependent Variables

Overall principle:

Hypo-elastic(-plastic) material model: $\varepsilon = \varepsilon_e + \varepsilon_p$

$\Delta\boldsymbol{\varepsilon}(t+\Delta t)$, $\boldsymbol{\sigma}(t)$, $H(t)$ → UMAT/VUMAT → $\boldsymbol{\sigma}(t+\Delta t)$, $H(t+\Delta t)$

ABAQUS/standard

$\boldsymbol{\sigma}(t+\Delta t), \boldsymbol{\sigma}(t)$ → STRESS

$H(t+\Delta t), H(t)$ → STATEV

$\Delta\boldsymbol{\varepsilon}(t+\Delta t)$ → DSTRAN

```
SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
+               RPL,DDSDDT,DRPLDE,DRPLDT,
+               STRAN,DSTRAN,TIMEA,DTIMEA,TEMP,DTEMP,PREDEF,DPRED,
+               CMNAME,NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,
+               DROT,PNEWDT,CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,
+               KSPT,KSTEP,KINC)
```

ABAQUS/explicit

```
 SUBROUTINE VUMAT(
+ NBLOCK, NDIR, NSHR, NSTATEV, NFIELDV, NPROPS, LANNEAL, STEPTIME,
+ TOTALTIME, DT, CMNAME, COORDMP, CHARLENGTH, PROPS, DENSITY,
+ STRAININC, RELSPININC, TEMPOLD, STRETCHOLD, DEFGRAdoLD, FIELdoLD,
+ STRESSOLD, STATEOLD, ENERINTERNOLD, ENERINELASOLD, TEMPNEW,
+ STRETCHNEW, DEFGRADNEW, FIELDNEW,
+ STRESSNEW, STATENEW, ENERINTERNNEW, ENERINELASNEW)
```

$\Delta\boldsymbol{\varepsilon}(t+\Delta t)$ → STRAININC

$\boldsymbol{\sigma}(t)$ → STRESSOLD

$H(t)$ → STATEOLD

$\boldsymbol{\sigma}(t+\Delta t)$ → STRESSNEW

$H(t+\Delta t)$ → STATENEW

(V)UMAT subroutine

H are the State Dependent Variables

Overall principle:

Hyper-elastic(-plastic) material model: $\mathbf{F} = \mathbf{F}_e \cdot \mathbf{F}_p$

$$\begin{matrix} \mathbf{F}(t+\Delta t) \\ H(t) \end{matrix} \longrightarrow \boxed{\text{UMAT/VUMAT}} \longrightarrow \begin{matrix} \boldsymbol{\sigma}(t+\Delta t) \\ H(t+\Delta t) \end{matrix}$$

Deformation gradient tensor $\mathbf{F}$

ABAQUS/standard

```
 SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
+               RPL,DDSDDT,DRPLDE,DRPLDT,
+               STRAN,DSTRAN,TIMEA,DTIMEA,TEMP,DTEMP,PREDEF,DPRED,
+               CMNAME,NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,
+               DROT,PNEWDT,CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,
+               KSPT,KSTEP,KINC)
```

$\mathbf{F}$

ABAQUS/explicit

```
 SUBROUTINE VUMAT(
+ NBLOCK, NDIR, NSHR, NSTATEV, NFIELDV, NPROPS, LANNEAL, STEPTIME,
+ TOTALTIME, DT, CMNAME, COORDMP, CHARLENGTH, PROPS, DENSITY,
+ STRAININC, RELSPININC, TEMPOLD, STRETCHOLD, DEFGRAdoLD, FIELdoLD,
+ STRESSOLD, STATEOLD, ENERINTERNOLD, ENERINELASOLD, TEMPNEW,
+ STRETCHNEW, DEFGRADNEW, FIELDNEW,
+ STRESSNEW, STATENEW, ENERINTERNNEW, ENERINELASNEW)
```

Stretch tensor $\mathbf{U}$

$\mathbf{F}$

(V)UMAT subroutine

Different energies can be updated and it can be useful to check energy balances.

Energies (elastic, plastic, creep)

ABAQUS/standard

```
 SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
+               RPL,DDSDDT,DRPLDE,DRPLDT,
+               STRAN,DSTRAN,TIMEA,DTIMEA,TEMP,DTEMP,PREDEF,DPRED,
+               CMNAME,NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,
+               DROT,PNEWDT,CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,
+               KSPT,KSTEP,KINC)
```

ABAQUS/explicit

```
 SUBROUTINE VUMAT(
+ NBLOCK, NDIR, NSHR, NSTATEV, NFIELDV, NPROPS, LANNEAL, STEPTIME,
+ TOTALTIME, DT, CMNAME, COORDMP, CHARLENGTH, PROPS, DENSITY,
+ STRAININC, RELSPININC, TEMPOLD, STRETCHOLD, DEFGRAdoLD, FIELdoLD,
+ STRESSOLD, STATEOLD, ENERINTERNOLD, ENERINELASOLD, TEMPNEW,
+ STRETCHNEW, DEFGRADNEW, FIELDNEW,
+ STRESSNEW, STATENEW, ENERINTERNNEW, ENERINELASNEW)
```

Internal energy

Dissipated plastic energy

(V)UMAT subroutine

! Some differences between UMAT and VUMAT

ABAQUS/standard		ABAQUS/explicit
$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{31} \\ \sigma_{23} \end{bmatrix} \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{12} \\ 2\varepsilon_{31} \\ 2\varepsilon_{23} \end{bmatrix}$$	Ordering tensor components	$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{23} \\ \sigma_{31} \end{bmatrix} \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \varepsilon_{12} \\ \varepsilon_{23} \\ \varepsilon_{31} \end{bmatrix}$$
Jaumann: $$\boldsymbol{\sigma}^{\nabla J} = \dot{\boldsymbol{\sigma}} - \mathbf{W} \cdot \boldsymbol{\sigma} + \boldsymbol{\sigma} \cdot \mathbf{W}$$	Stress rate (objectivity)	Green-Naghdi: $$\boldsymbol{\sigma}^{\nabla G} = \dot{\boldsymbol{\sigma}} - \boldsymbol{\Omega} \cdot \boldsymbol{\sigma} + \boldsymbol{\sigma} \cdot \boldsymbol{\Omega}$$

(V)UMAT subroutine

ABAQUS/standard

! Some differences between UMAT and VUMAT

The user must ensure the objectivity of all tensor stored in the state dependent variables

```
 SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
+               RPL,DDSDDT,DRPLDE,DRPLDT,
+               STRAN,DSTRAN,TIMEA,DTIMEA,TEMP,DTEMP,PREDEF,DPRED,
+               CMNAME,NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,
+               DROT,PNEWDT,CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,
+               KSPT,KSTEP,KINC)
```

Utility routine: `CALL ROTSIG(S,R,SPRIME,LSTR,NDI,NSHR)`

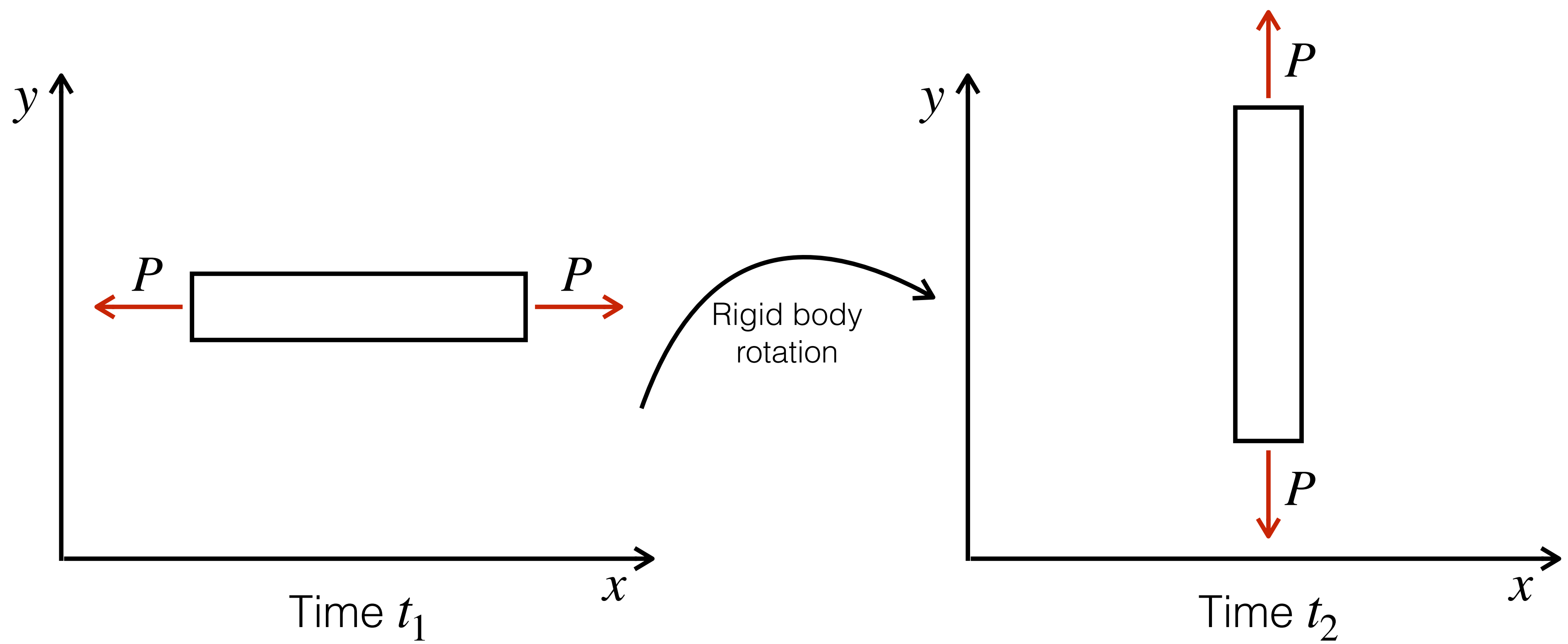
A simple workaround is to define a material orientation in the section card.

ABAQUS/explicit

The objectivity of all tensor stored in the state dependent variables is handled by ABAQUS

Interlude on stress objectivity

- Consider a tensile specimen (cross section A) subjected to a constant load P
- Initially (time t_1) the specimen is aligned with the x axis
- At time t_2 a rigid body rotation is applied such that the specimen is aligned with the y axis



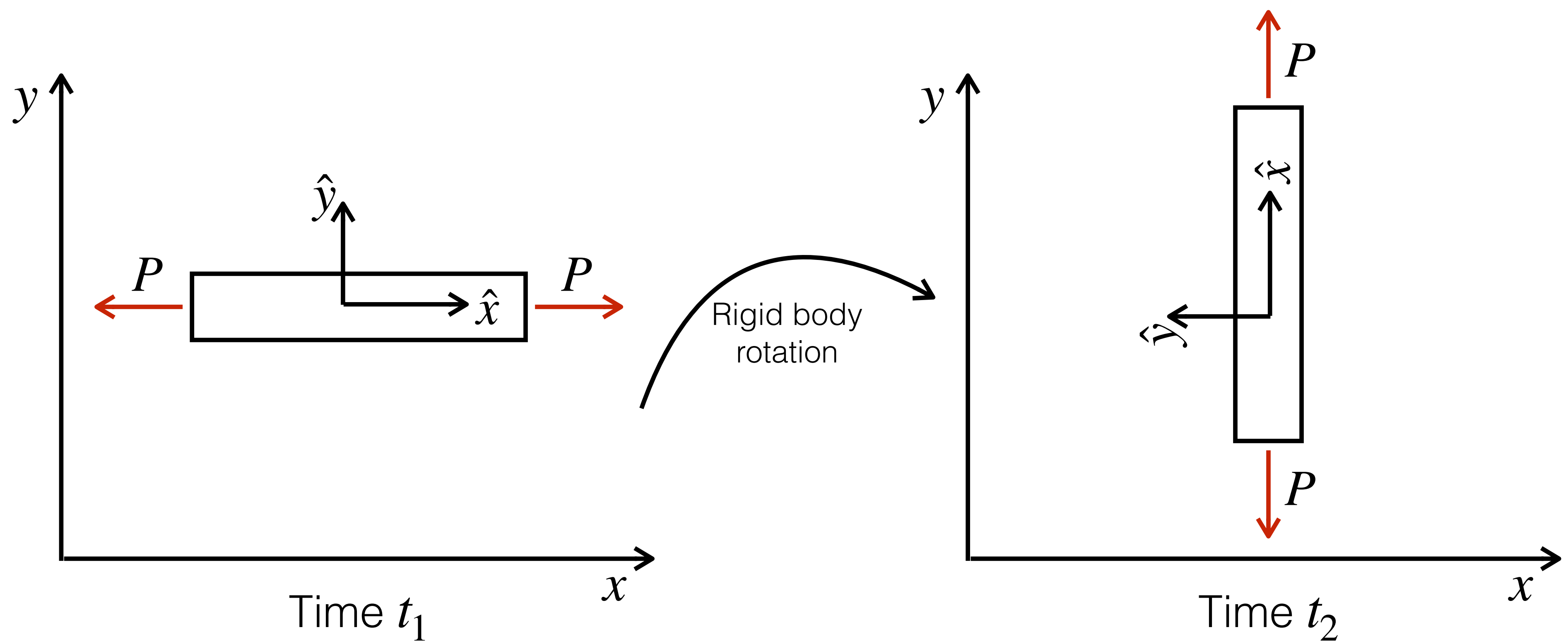
The resulting stress tensor $\boldsymbol{\sigma}$:

$$\boldsymbol{\sigma} = \begin{bmatrix} \frac{P}{A} & 0 \\ 0 & 0 \end{bmatrix} \qquad \boldsymbol{\sigma} = \begin{bmatrix} 0 & 0 \\ 0 & \frac{P}{A} \end{bmatrix}$$

Interlude on stress objectivity

Material axes follows the material:

Co-rotational approach: $\hat{\mathbf{D}} = \mathbf{R}^T\mathbf{D}\mathbf{R}$ $\quad\quad$ $\hat{\boldsymbol{\sigma}} = \mathbf{R}^T\boldsymbol{\sigma}\mathbf{R}$



The resulting stress tensor $\hat{\boldsymbol{\sigma}}$: $\quad \hat{\boldsymbol{\sigma}} = \begin{bmatrix} \frac{P}{A} & 0 \\ 0 & 0 \end{bmatrix}$

Polar decomposition: $\mathbf{F} = \mathbf{R}\mathbf{U}$

(V)UMAT subroutine

The implicit solver requires the tangent operator $\dfrac{\partial \Delta \boldsymbol{\sigma}}{\partial \Delta \boldsymbol{\varepsilon}}$

to form the stiffness matrix of the model.

ABAQUS/standard

```
 SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
+               RPL,DDSDDT,DRPLDE,DRPLDT,
+               STRAN,DSTRAN,TIMEA,DTIMEA,TEMP,DTEMP,PREDEF,DPRED,
+               CMNAME,NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,
+               DROT,PNEWDT,CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,
+               KSPT,KSTEP,KINC)
```

- The tangent operator (DDSDDE) is very important and can be the reason for poor convergence speed and crashes.
- The solver requires by default the consistent tangent operator which can be costly, and difficult to compute.
- A workaround is to supply the elasticity matrix into DDSDDE and use the quasi-newton solution technique from ABAQUS.

(V)UMAT subroutine

ABAQUS/explicit

At the first time step:

- A purely elastic computation is required to compute the stable time-step
- Special care must be taken for a visco-elastic, anisotropic elasticity, hyper elasticity etc…

$$\Delta t \approx l_e \sqrt{\frac{\rho}{E}}$$

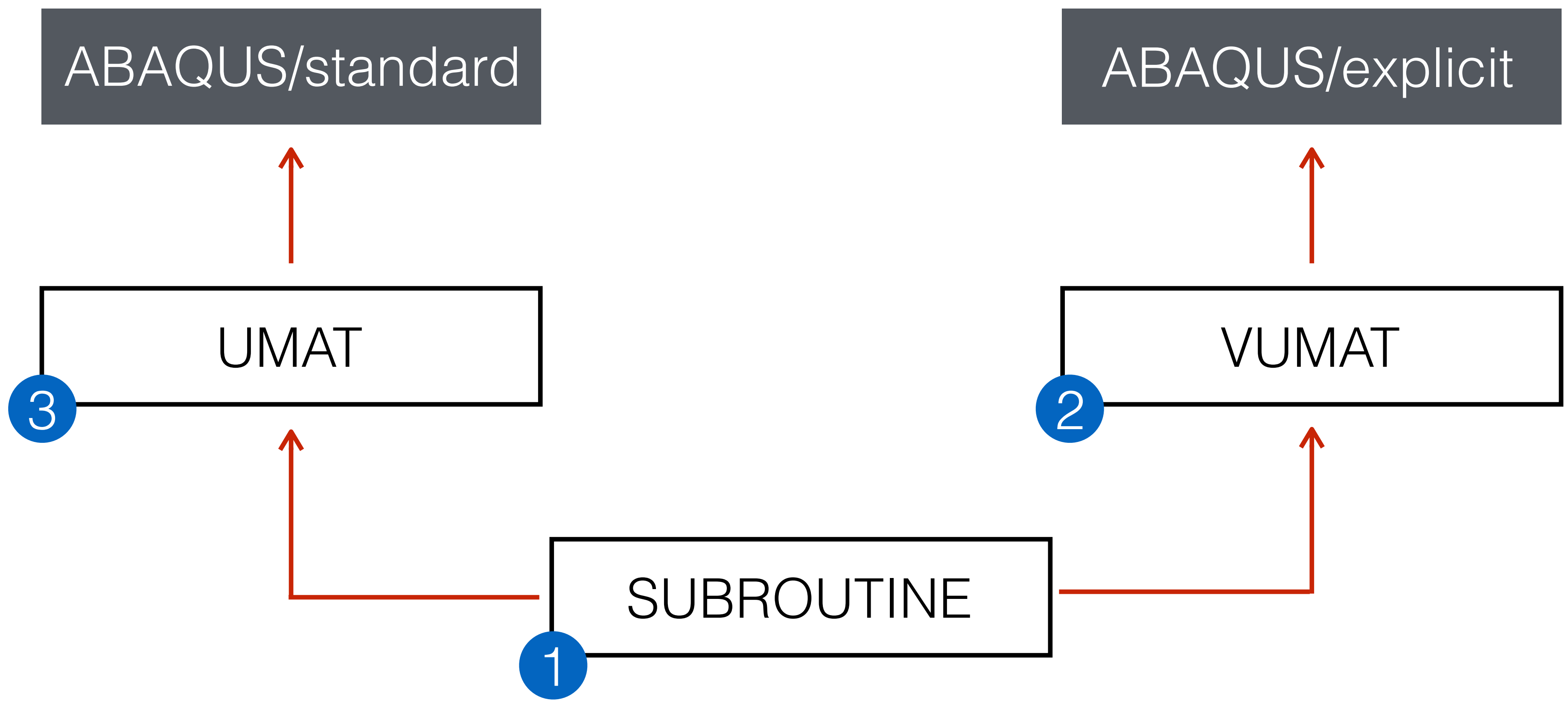
where:

- l_e is the element characteristic length
- ρ is the density of the material
- E is the Young's modulus

```
!------------------------------------------------------------------------
!     Initialization step (elastic)
!------------------------------------------------------------------------
      if((steptime.eq.totaltime).and.(steptime.eq.zero))then
        E0   = props(1)
        NU   = props(2)
        BULK = E0/(3.0*(1.0-2.0*NU))
        R2G  = E0/(1.0+NU)
        do i=1,NBLOCK
          if(NSHR.eq.3)then
            volstrain = STRAININC(i,1)+STRAININC(i,2)+STRAININC(i,3)
            STRESSNEW(i,1) = R2G*(STRAININC(i,1)-third*volstrain)
     +                     +BULK*volstrain
            STRESSNEW(i,2) = R2G*(STRAININC(i,2)-third*volstrain)
     +                     +BULK*volstrain
            STRESSNEW(i,3) = R2G*(STRAININC(i,3)-third*volstrain)
     +                     +BULK*volstrain
            STRESSNEW(i,4) = R2G*STRAININC(i,4)
            STRESSNEW(i,5) = R2G*STRAININC(i,5)
            STRESSNEW(i,6) = R2G*STRAININC(i,6)
          else
            volstrain = STRAININC(i,1)+STRAININC(i,2)+STRAININC(i,3)
            STRESSNEW(i,1) = R2G*(STRAININC(i,1)-third*volstrain)
     +                     +BULK*volstrain
            STRESSNEW(i,2) = R2G*(STRAININC(i,2)-third*volstrain)
     +                     +BULK*volstrain
            STRESSNEW(i,3) = R2G*(STRAININC(i,3)-third*volstrain)
     +                     +BULK*volstrain
            STRESSNEW(i,4) = R2G*STRAININC(i,4)
          endif
        enddo
!------------------------------------------------------------------------
```

(V)UMAT subroutine

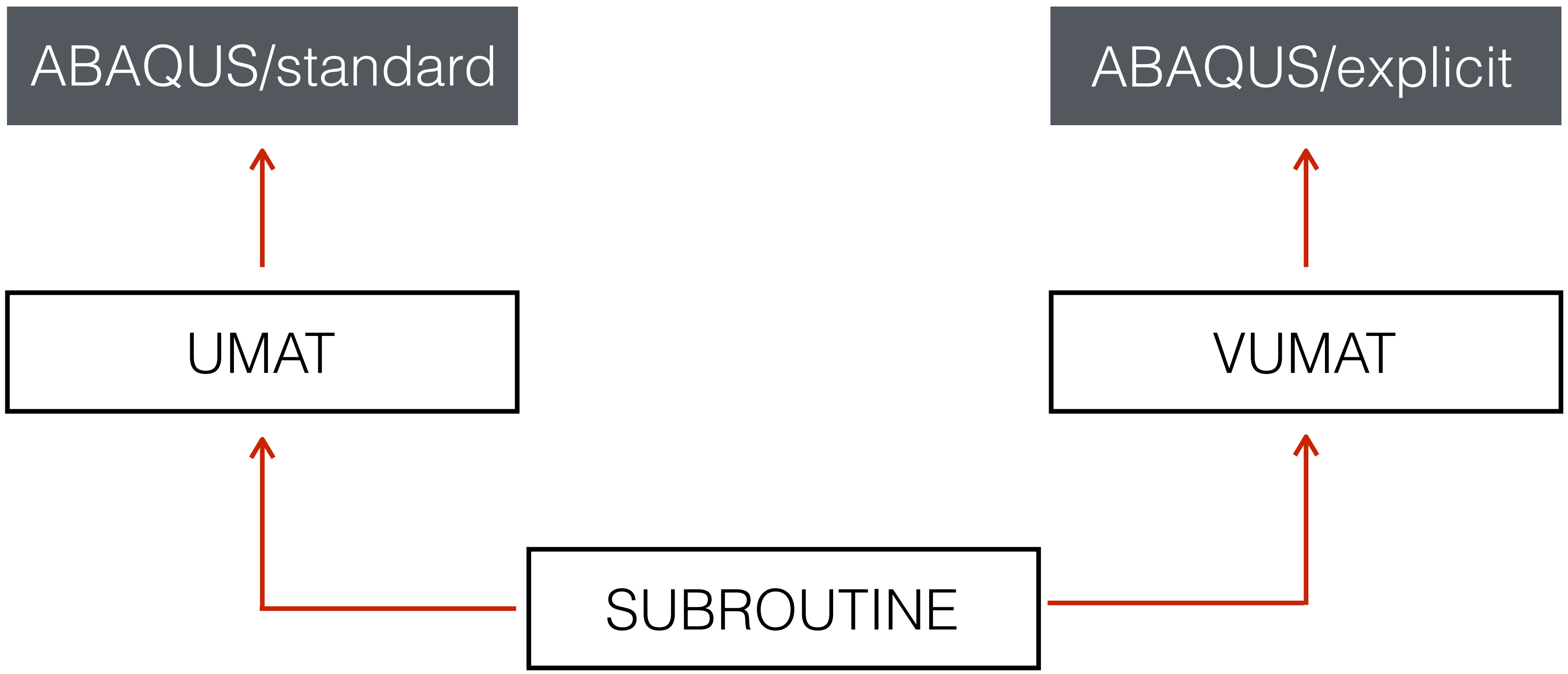
A simplified approach to UMAT/VUMAT coding:



1. Code and compile the material model in FORTRAN
2. Link the code to the VUMAT subroutine
3. Link the code to the UMAT subroutine

(V)UMAT subroutine

A simplified approach to UMAT/VUMAT coding:



- Easier to debug
- A model available in both ABAQUS/standard and explicit

- Less efficient code

(V)UMAT subroutine

ABAQUS/standard

- Include a subroutine with the code
- Send data to the subroutine

```fortran
      INCLUDE './UMAT_MODEL.f'
!-----------------------------------------------------------------------
      SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
     +                RPL,DDSDDT,DRPLDE,DRPLDT,
     +                STRAN,DSTRAN,TIMEA,DTIMEA,TEMP,DTEMP,PREDEF,DPRED,
     +                CMNAME,NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,
     +                DROT,PNEWDT,CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,
     +                KSPT,KSTEP,KINC)
      INCLUDE 'ABA_PARAM.INC'
!-----------------------------------------------------------------------
!-----Declaration ABAQUS variables
!-----------------------------------------------------------------------
      character*80 CMNAME
      DIMENSION STRESS(NTENS),STATEV(NSTATV),DDSDDE(NTENS,NTENS),
     +          DDSDDT(NTENS),DRPLDE(NTENS),STRAN(NTENS),DSTRAN(NTENS),
     +          TIMEA(2),PREDEF(1),DPRED(1),PROPS(NPROPS),COORDS(3),
     +          DROT(3,3),DFGRD0(3,3),DFGRD1(3,3)
!-----------------------------------------------------------------------
!     Call UMAT_MODEL
!-----------------------------------------------------------------------
      call UMAT_MODEL(DDSDDE,STRESS,STATEV,DSTRAN,PROPS,
     +                KINC,NTENS,NSTATV,NPROPS)
!-----------------------------------------------------------------------
!     End of subroutine
!-----------------------------------------------------------------------
      return
      end
```

(V)UMAT subroutine

ABAQUS/explicit

- Include a subroutine with the code
- Send data to the subroutine

- Vectorized to scalar
- Tensor ordering

```
      INCLUDE './UMAT_MODEL.f'
!-----------------------------------------------------------------------
      SUBROUTINE VUMAT(
     + NBLOCK, NDIR, NSHR, NSTATEV, NFIELDV, NPROPS, LANNEAL, STEPTIME,
     + TOTALTIME, DT, CMNAME, COORDMP, CHARLENGTH, PROPS, DENSITY,
     + STRAININC, RELSPININC, TEMPOLD, STRETCHOLD, DEFGRAdoLD, FIELdoLD,
     + STRESSOLD, STATEOLD, ENERINTERNOLD, ENERINELASOLD, TEMPNEW,
     + STRETCHNEW, DEFGRADNEW, FIELDNEW,
     + STRESSNEW, STATENEW, ENERINTERNNEW, ENERINELASNEW)
C
                        :
!-----------------------------------------------------------------------
c           GRAB HISTORY VARIABLES
!-----------------------------------------------------------------------
            do k=1,NSTATEV
               STATEV(k) = STATEOLD(i,k)
            enddo
!-----------------------------------------------------------------------
c           CALL UMAT_MODEL
!-----------------------------------------------------------------------
            call UMAT_MODEL(DDSDDE,STRESS,STATEV,DSTRAN,PROPS2,
     +                      KINC,NTENS,NSTATV,NPROPS)
!-----------------------------------------------------------------------
c           UNPACK STRESSES
!-----------------------------------------------------------------------
            STRESSNEW(i,1) = STRESS(1)
            STRESSNEW(i,2) = STRESS(2)
            STRESSNEW(i,3) = STRESS(3)
            STRESSNEW(i,4) = STRESS(4)
            if(NSHR.eq.3)then
               STRESSNEW(i,6) = STRESS(5)
               STRESSNEW(i,5) = STRESS(6)
            endif
!-----------------------------------------------------------------------
```

```
----------------------------------
GRAB STRAININC
----------------------------------
DSTRAN(1) = STRAININC(i,1)
DSTRAN(2) = STRAININC(i,2)
DSTRAN(3) = STRAININC(i,3)
DSTRAN(4) = 2.0*STRAININC(i,4)
if(NSHR.eq.3)then
   DSTRAN(5) = 2.0*STRAININC(i,6)
   DSTRAN(6) = 2.0*STRAININC(i,5)
endif
----------------------------------
```

(V)UMAT subroutine

The same material card input can be used between ABAQUS/Standard and ABAQUS/Explicit

ABAQUS/explicit

```
**------------------------------------------------------------
** MATERIALS
**------------------------------------------------------------
*material,name=EXAMPLE_VUMAT
*density
7.8e-9
*user material, CONSTANTS=24
**      E,       NU, BLANK,  BLANK, BLANK, BLANK,   BLANK, BLANK
 210000.0,      0.3,   0.0,    0.0,   0.0,   0.0,     0.0,   0.0
** SIGMA0,       T1,    Q1,     T2,    Q2,    T3,      Q3, BLANK
    250.0,  10000.0,  10.0, 1000.0, 100.0, 100.0,  1000.0,   0.0
**     WC,    DCRIT, BLANK,  BLANK, BLANK, BLANK,   BLANK, BLANK
    100.0,      1.0
*depvar, delete=3
3
1, P, "Equivalent plastic strain"
2, D, "Damage"
3,STATUS, "Status variable"
**------------------------------------------------------------
```

ABAQUS/standard

```
**------------------------------------------------------------
** MATERIALS
**------------------------------------------------------------
*material,name=EXAMPLE_UMAT
*user material, CONSTANTS=24
**      E,       NU, BLANK,  BLANK, BLANK, BLANK,   BLANK, BLANK
 210000.0,      0.3,   0.0,    0.0,   0.0,   0.0,     0.0,   0.0
** SIGMA0,       T1,    Q1,     T2,    Q2,    T3,      Q3, BLANK
    250.0,  10000.0,  10.0, 1000.0, 100.0, 100.0,  1000.0,   0.0
**     WC,    DCRIT, BLANK,  BLANK, BLANK, BLANK,   BLANK, BLANK
    100.0,      1.0
*depvar, delete=3
3
1, P, "Equivalent plastic strain"
2, D, "Damage"
3,STATUS, "Status variable"
**------------------------------------------------------------
```

(V)UMAT subroutine

ABAQUS/standard

Extra keywords for UMATs

Material orientation

```
*Orientation, name=Ori-1
**  n1x,    n1y,    n1z,    n2x,    n2y,    n2z
      1,      0,      0,      0,      1,      0
** axis, angle
      3,    0.0
*Solid Section, elset=ELEMENTS, material=EXAMPLE_UMAT, controls=EC-1, orientation=Ori-1
,

**-----------------------------------------------
** ELEMENT CONTROLS
**-----------------------------------------------
*Section Controls, name=EC-1, hourglass=ENHANCED
1., 1., 1.
**-----------------------------------------------
```

Section controls (Hourglass) after *assembly

```
**-------------------------------------
** STEP
**-------------------------------------
*Step, name=LOADING, nlgeom=YES, inc=10000
*Static
1.0e-4, 0.02,  1e-08,  2.0e-4
*Solution technique, type=QUASI-NEWTON
**-------------------------------------
```

Quasi-newton solver

(V)UMAT subroutine

In this example, we want to develop an elasto-plastic model with a ductile fracture model.

Hypo-elasticity:

$$\hat{\mathbf{D}} = \hat{\mathbf{D}}_e + \hat{\mathbf{D}}_p$$

Isotropic elastic:

$$\dot{\hat{\boldsymbol{\sigma}}} = \frac{E}{1+\nu}\hat{\mathbf{D}}_e' + \frac{E}{3(1-2\nu)}\mathrm{tr}\hat{\mathbf{D}}_e\mathbf{I}$$

Yield function:

$$f = \sigma_{eq} - \sigma_y$$

Associated plastic flow:

$$\hat{\mathbf{D}}_p = \dot{\lambda}\frac{\partial f}{\partial \hat{\boldsymbol{\sigma}}}$$

Equivalent stress:

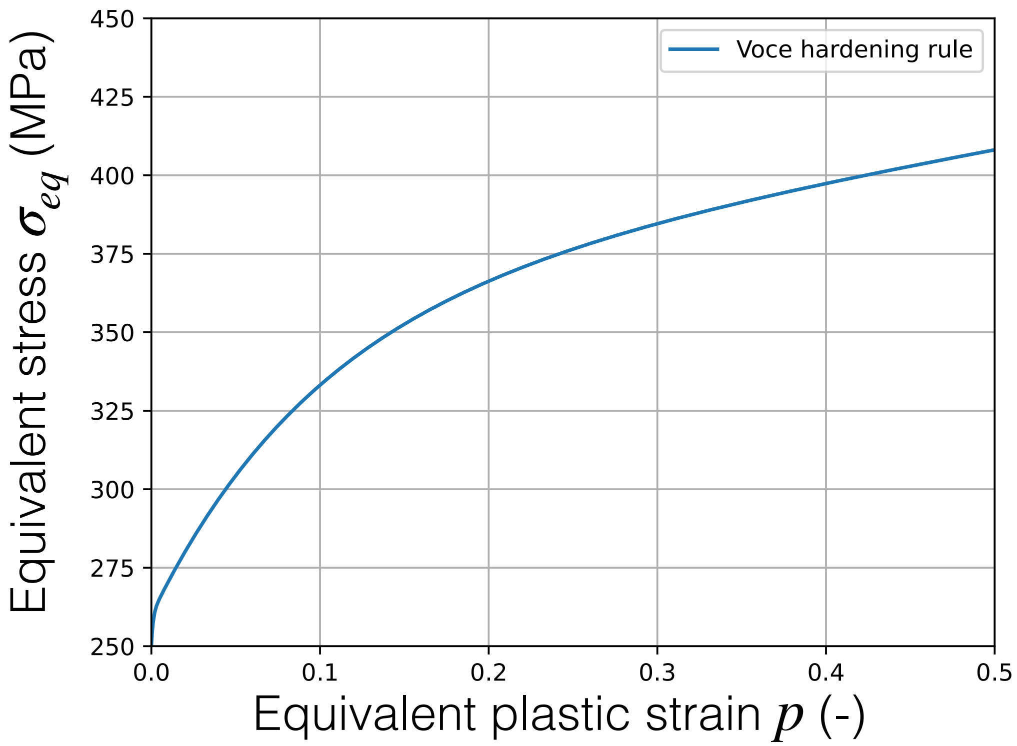
$$\sigma_{eq} = \sqrt{3J_2}$$

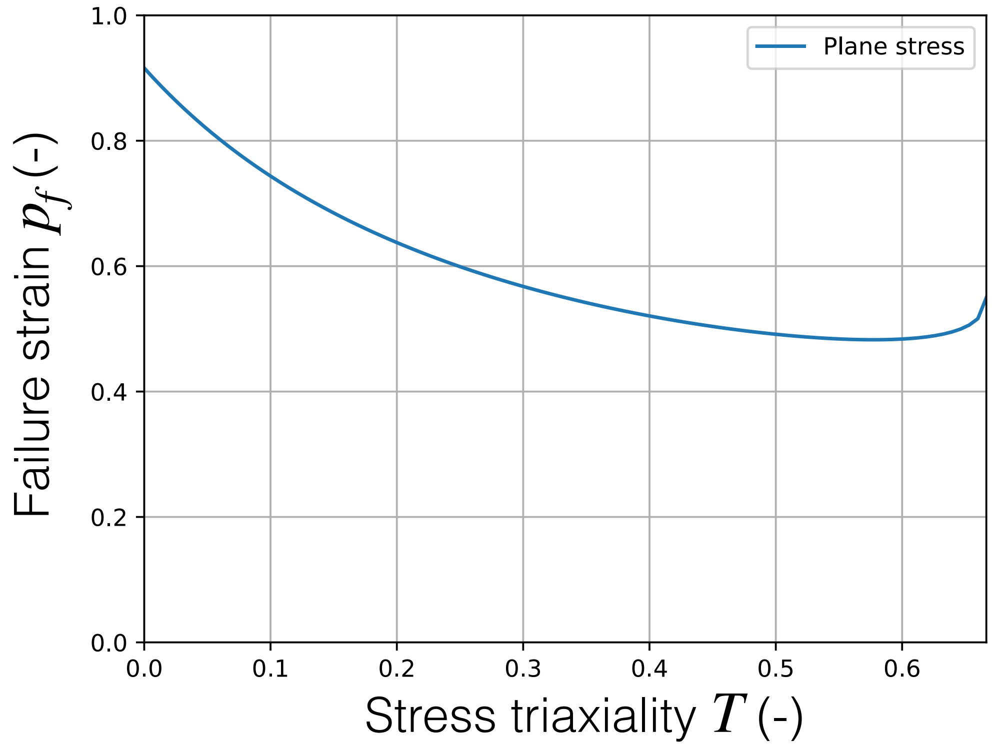
Isotropic hardening:

$$\sigma_y = \sigma_0 + \sum_{i=1}^{3} Q_i\left(1 - \exp\left(-\frac{\theta_i}{Q_i}p\right)\right)$$

Damage indicator model:

$$D = \int_0^{p_f} \frac{\langle\sigma_1\rangle}{W_c}\dot{p} \leq D_c$$





(V)UMAT subroutine

Computational plasticity:

1) Elastic prediction

$$\hat{\boldsymbol{\sigma}}^{TRIAL} = \hat{\boldsymbol{\sigma}}^{t} + \hat{\mathbf{C}}_e : \hat{\mathbf{D}}^{(t+\Delta t)}\Delta t$$

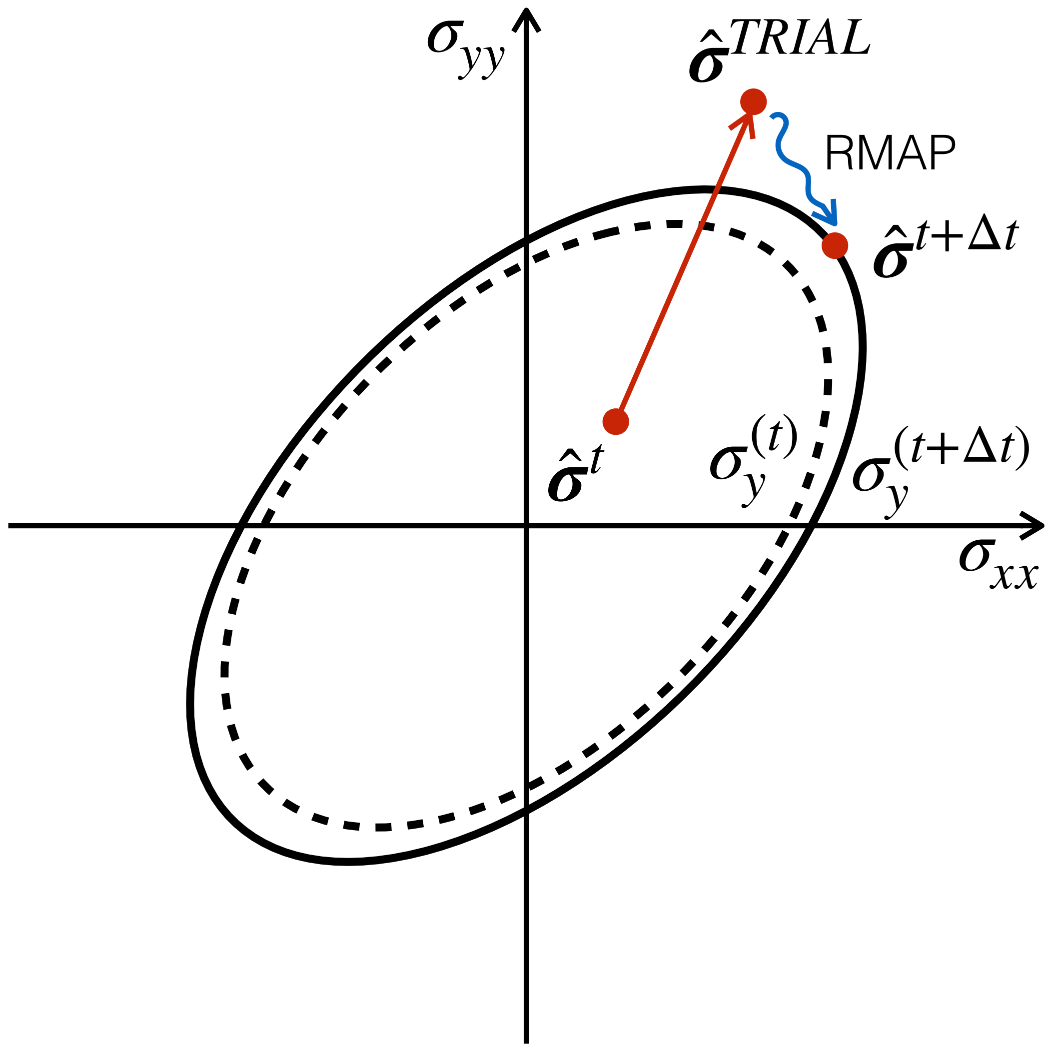
2) Compute yield function

$$f = \sigma_{eq}\left(\hat{\boldsymbol{\sigma}}^{TRIAL}\right) - \sigma_y^{(t)}$$

3) Check yield function

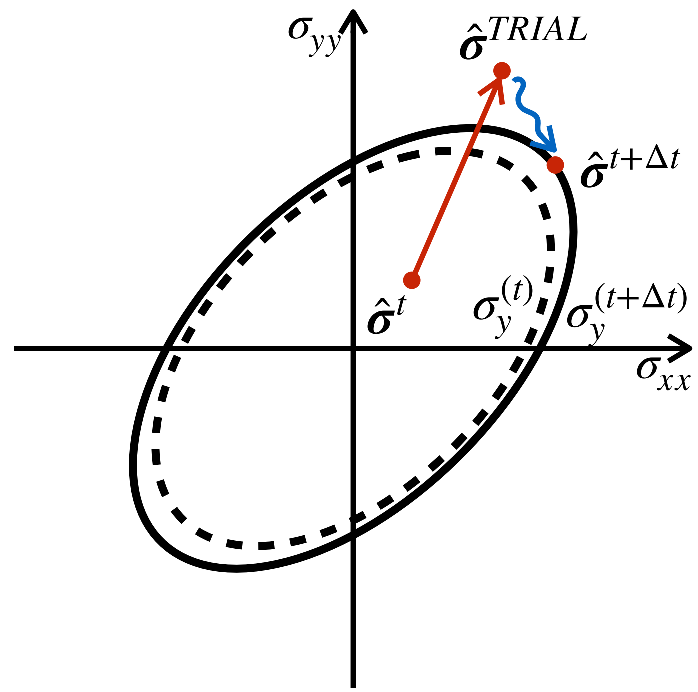
If $f < 0 \longrightarrow \hat{\boldsymbol{\sigma}}^{t+\Delta t} = \hat{\boldsymbol{\sigma}}^{TRIAL}$

If $f \geq 0 \longrightarrow$ Solve the non-linear system of equations (RMAP)

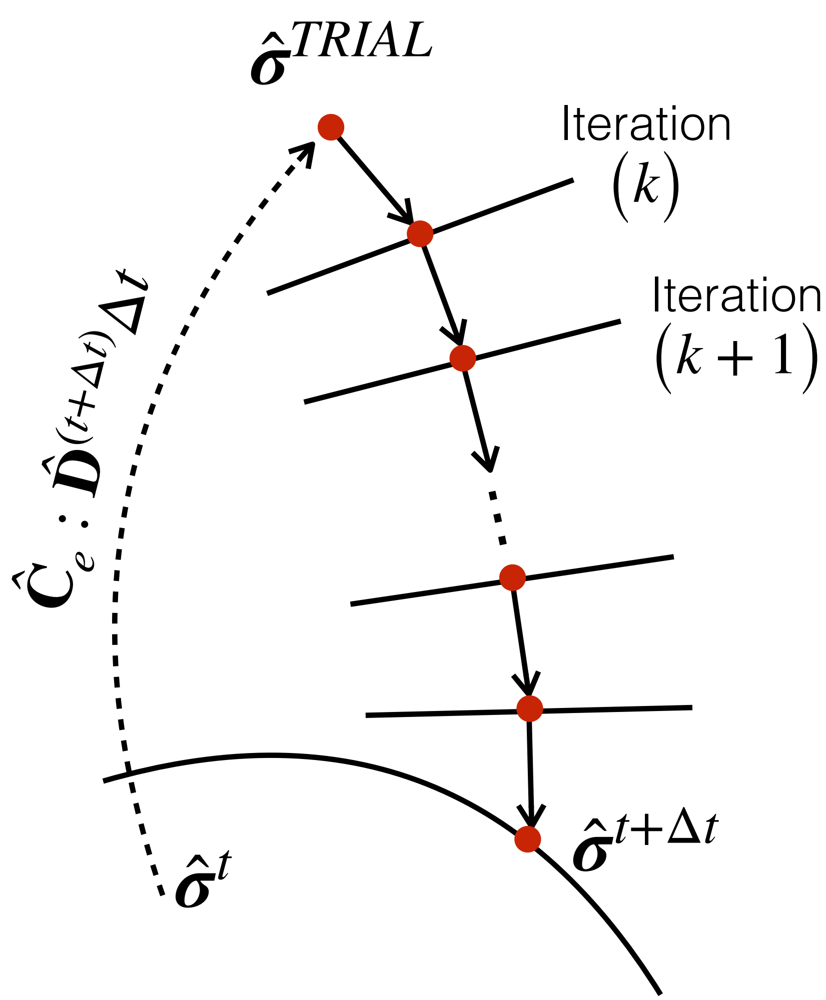


(V)UMAT subroutine

Return Map Algorithm:



Cutting-plane algorithm:



Linearisation of the yield surface: $f\left(\hat{\boldsymbol{\sigma}}\right)|_{k+1} = 0$

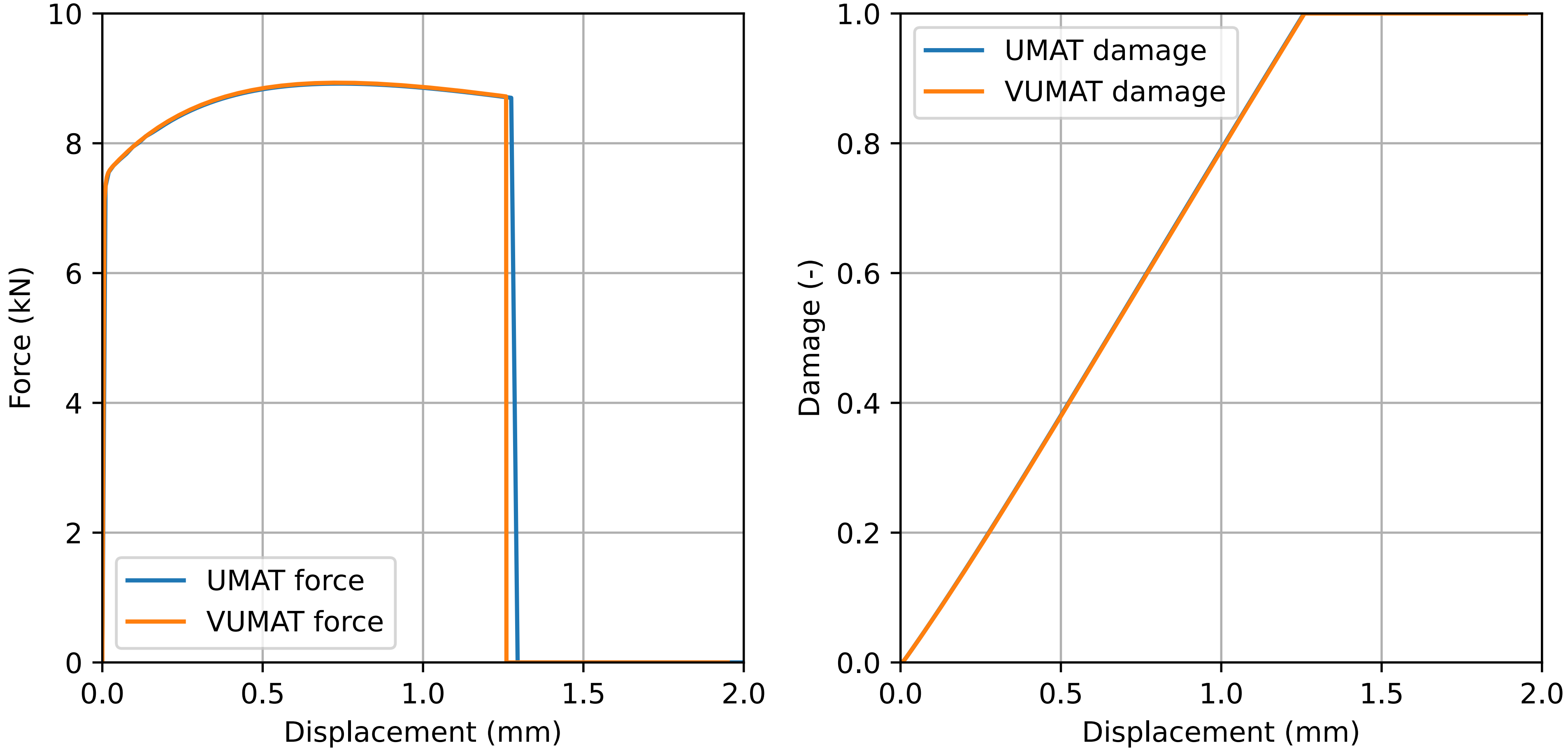
$$\delta\lambda^{(k+1)} = \frac{f\left(\hat{\boldsymbol{\sigma}}\right)\Big|_k}{\dfrac{\partial f}{\partial \hat{\boldsymbol{\sigma}}}\Big|_k : \mathbf{C}_e : \dfrac{\partial f}{\partial \hat{\boldsymbol{\sigma}}}\Big|_k - \dfrac{\partial f}{\partial \sigma_y}\Big|_k \dfrac{\partial \sigma_y}{\partial \lambda}\Big|_k}$$

$$\hat{\boldsymbol{\sigma}}^{(k+1)} = \hat{\boldsymbol{\sigma}}^{(k)} - \delta\lambda \hat{\mathbf{C}}_e : \frac{\partial f}{\partial \hat{\boldsymbol{\sigma}}}$$

$$\boldsymbol{\sigma}_y^{(k+1)} = \boldsymbol{\sigma}_y^{(k)} - \delta\lambda \frac{\partial f}{\partial \lambda}$$

(V)UMAT subroutine

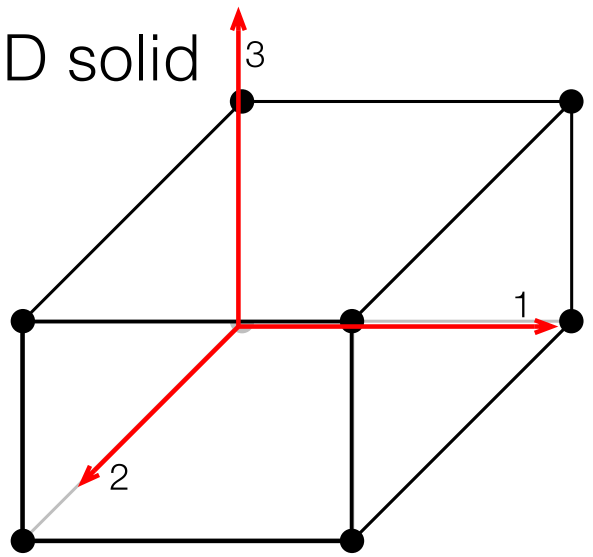
Results from the UMAT and VUMAT subroutines



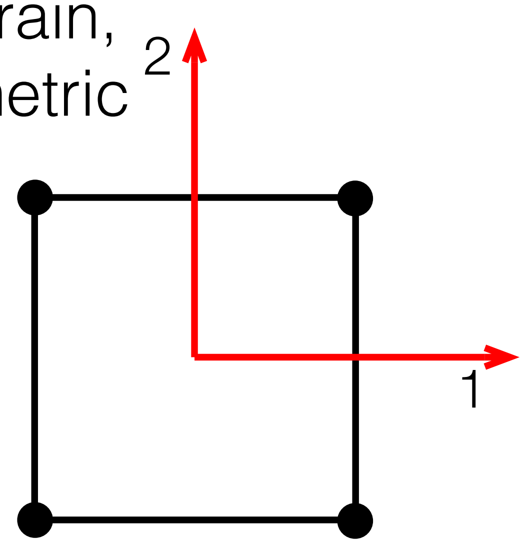
(V)UMAT subroutine

The UMAT and VUMAT subroutines should work for:

3D solid



2D plane strain,
2D axisymmetric



Stress tensor:

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{12} & \sigma_{22} & \sigma_{23} \\ \sigma_{23} & \sigma_{13} & \sigma_{33} \end{bmatrix}$$

Stress tensor:

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{bmatrix}$$

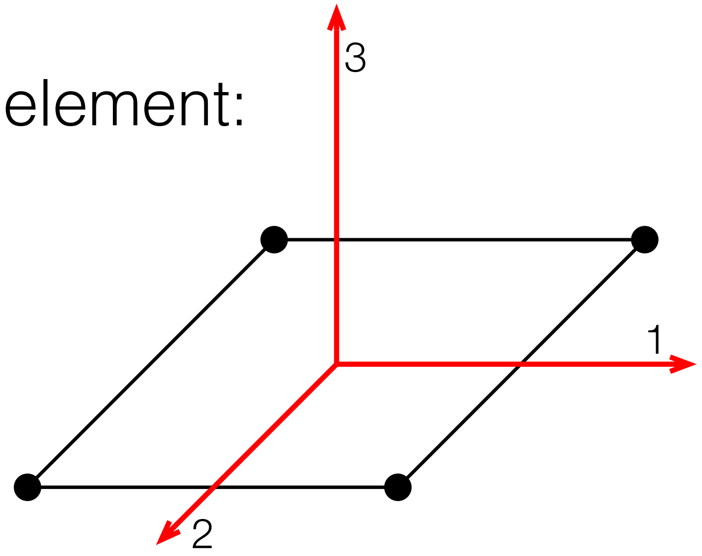
VUMAT

```
STRESS(1) = STRESSOLD(i,1)
STRESS(2) = STRESSOLD(i,2)
STRESS(3) = STRESSOLD(i,3)
STRESS(4) = STRESSOLD(i,4)
if(NSHR.eq.3)then
    STRESS(5) = STRESSOLD(i,6)
    STRESS(6) = STRESSOLD(i,5)
endif
```

```
DSTRAN(1) = STRAININC(i,1)
DSTRAN(2) = STRAININC(i,2)
DSTRAN(3) = STRAININC(i,3)
DSTRAN(4) = 2.0*STRAININC(i,4)
if(NSHR.eq.3)then
    DSTRAN(5) = 2.0*STRAININC(i,6)
    DSTRAN(6) = 2.0*STRAININC(i,5)
endif
```

(V)UMAT subroutine

Shell element:



$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13}^e \\ \sigma_{12} & \sigma_{22} & \sigma_{23}^e \\ \boxed{\sigma_{23}^e \quad \sigma_{13}^e} & & 0 \end{bmatrix}$$

Through-thickness shear stresses handled externally as elastic

```
**--------------------------------------------------------------
*material,name=EXAMPLE_VUMAT_SHELL
*density
7.8e-9
*user material, CONSTANTS=24
**      E,       NU, BLANK,  BLANK, BLANK, BLANK,  BLANK, BLANK
 210000.0,      0.3,   0.0,    0.0,   0.0,   0.0,    0.0,   0.0
** SIGMA0,       T1,    Q1,     T2,    Q2,    T3,     Q3, BLANK
    250.0, 10000.0,   10.0, 1000.0, 100.0, 100.0, 1000.0,   0.0
**     WC,    DCRIT, BLANK,  BLANK, BLANK, BLANK,  BLANK, BLANK
    250.0,      1.0
*Transverse Shear
6730.7,6730.7, 0
*depvar, delete=3
3
1, P, "Equivalent plastic strain"
2, D, "Damage"
3,STATUS, "Status variable"
**--------------------------------------------------------------
```

Transverse shear stiffnesses: K_{11}, K_{22}, K_{12}

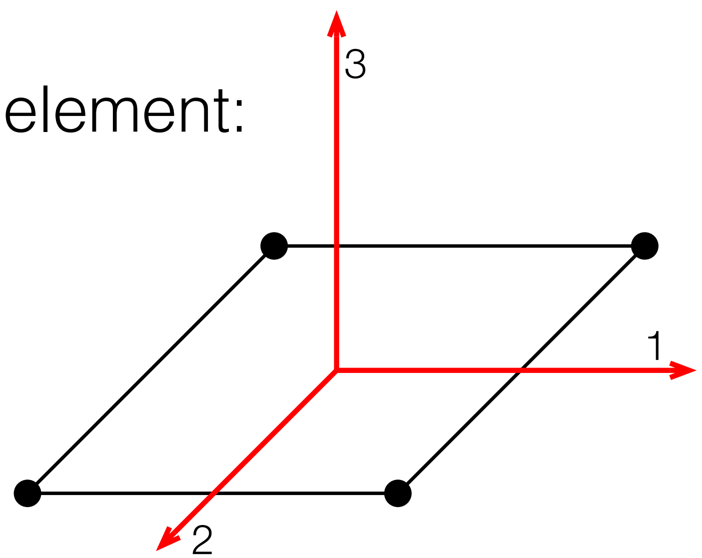
$$K_{11} = K_{22} = \frac{5}{6} G t_0$$

where G is the shear modulus and t_0 is the initial thickness of the shell element.

$$K_{12} = 0$$

(V)UMAT subroutine

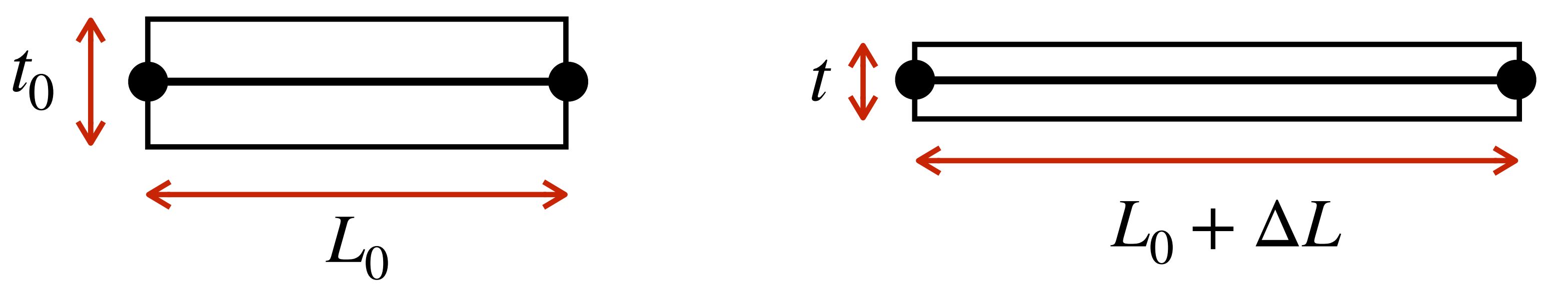
Shell element:



$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13}^{e} \\ \sigma_{12} & \sigma_{22} & \sigma_{23}^{e} \\ \sigma_{23}^{e} & \sigma_{13}^{e} & \boxed{0} \end{bmatrix}$$

Through-thickness stress $\boldsymbol{\sigma}_{33} = 0$

Thickness change:



$$t = t_0 \exp\left(\varepsilon_{33}\right)$$

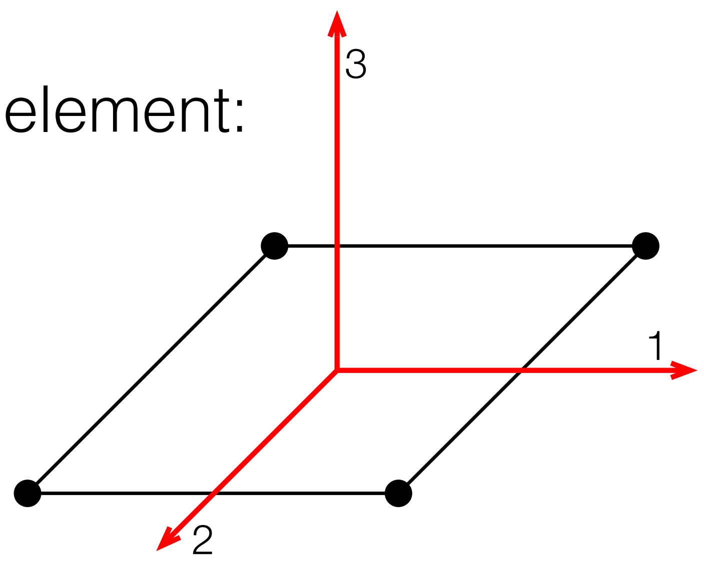
```
!-----------------------------------------
c            UNPACK STRESSES
!-----------------------------------------
          STRESSNEW(i,1) = STRESS(1)
          STRESSNEW(i,2) = STRESS(2)
          STRESSNEW(i,3) = 0.0
          STRESSNEW(i,4) = STRESS(4)
          STRAININC(i,3) = DSTRAN(3)
!-----------------------------------------
```

Plane stress hypothesis: $\sigma_{33} = 0$ → `STRESSNEW(i,3) = 0.0`

Thickness change: $\Delta\varepsilon_{33}$ → `STRAININC(i,3) = DSTRAN(3)`

(V)UMAT subroutine

Shell element:



$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13}^{e} \\ \sigma_{12} & \sigma_{22} & \sigma_{23}^{e} \\ \sigma_{23}^{e} & \sigma_{13}^{e} & 0 \end{bmatrix}$$

Through-thickness stress $\boldsymbol{\sigma}_{33} = 0$

Enforcing plane stress in a VUMAT:

1. Reformulate all constitutive equations according to plane stress hypothesis

2. Use 3D constitutive equations and iterate to enforce plane stress hypothesis

(V)UMAT subroutine

2 Use 3D constitutive equations and iterate to enforce plane stress hypothesis

Iteration 1

Assume elastic response

$$\Delta\varepsilon_{33} = -\frac{\nu}{1-\nu}\left(\Delta\varepsilon_{11} + \Delta\varepsilon_{22}\right)$$

Compute $\boldsymbol{\sigma}^{t+\Delta t}$, store $\Delta\varepsilon_{33}$ and σ_{33}

Iteration 2

Assume plastic incompressibility

$$\Delta\varepsilon_{33} = -\left(\Delta\varepsilon_{11} + \Delta\varepsilon_{22}\right)$$

Compute $\boldsymbol{\sigma}^{t+\Delta t}$, store $\Delta\varepsilon_{33}$ and σ_{33}

Iteration n

Use a secant update for $\Delta\varepsilon_{33}$

$$\Delta\varepsilon_{33}\,|_n = \Delta\varepsilon_{33}\,|_{n-1} - H_t\,.\,\sigma_{33}\,|_{n-1}$$

$$H_t = \frac{\Delta\varepsilon_{33}\,|_{n-1} - \Delta\varepsilon_{33}\,|_{n-2}}{\sigma_{33}\,|_{n-1} - \sigma_{33}\,|_{n-2}}$$

$$\Delta\boldsymbol{\varepsilon} = \begin{bmatrix} \Delta\varepsilon_{11} & \Delta\varepsilon_{22} & \Delta\varepsilon_{33} & \Delta\varepsilon_{12} & 0 & 0 \end{bmatrix}^T$$

3D SUBROUTINE

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{22} & \sigma_{33} & \sigma_{12} & 0 & 0 \end{bmatrix}^T$$

If $|\sigma_{33}| > \xi\left(|\sigma_{11}| + |\sigma_{22}| + |\sigma_{12}|\right)$

ξ is a numerical tolerance ($1e^{-5}$)

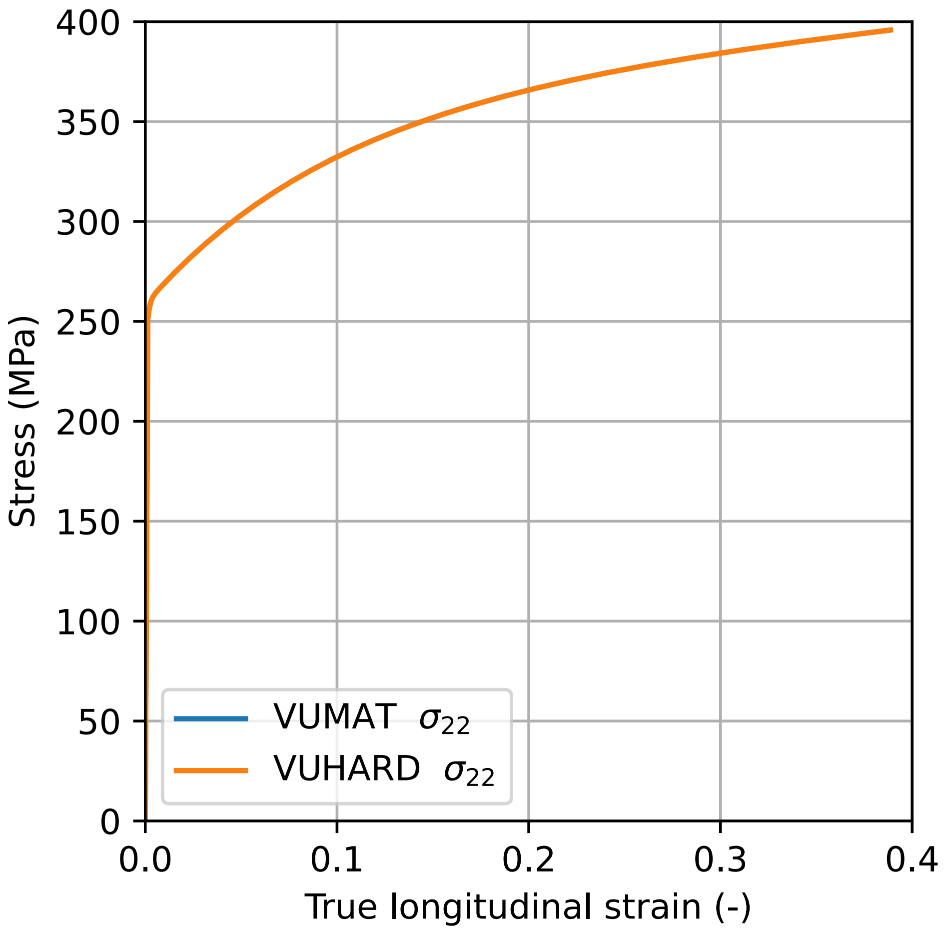
(V)UMAT subroutine

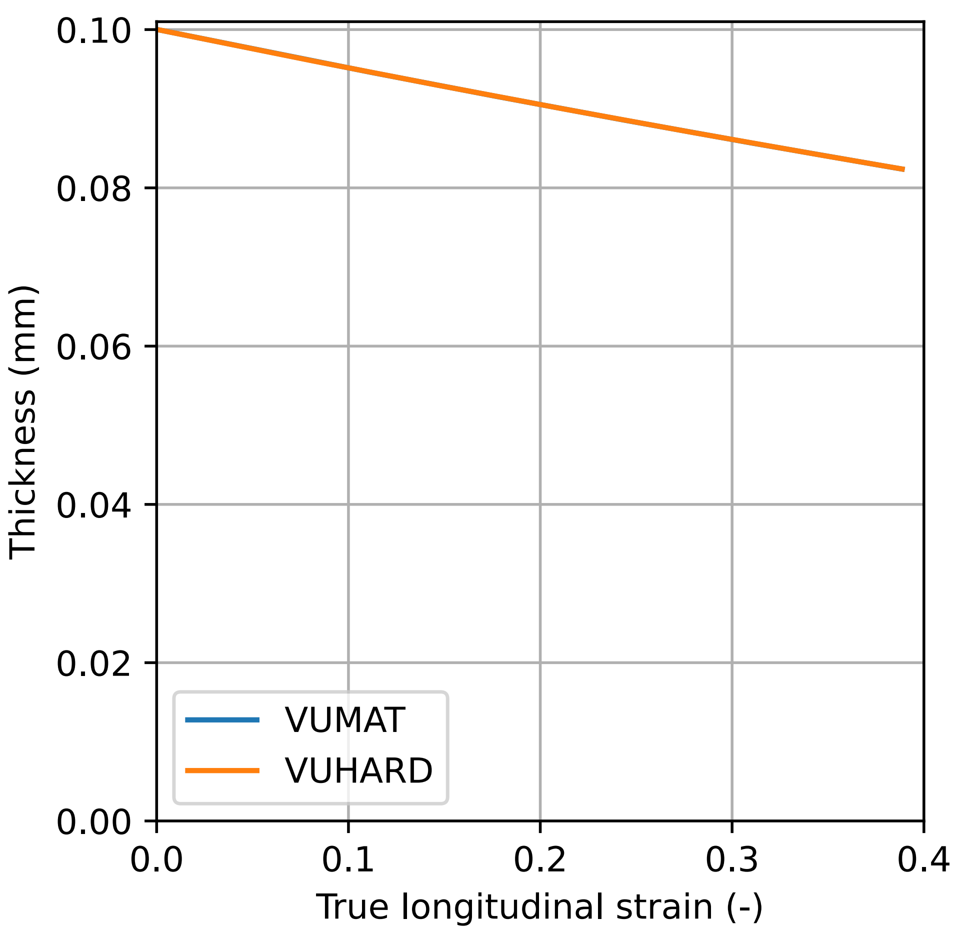
Enforcing plane stress in a VUMAT:

Method	1	2
	Reformulate all constitutive equations according to plane stress hypothesis	Use 3D constitutive equations and iterate to enforce plane stress hypothesis
+	Computationally efficient	No need to change constitutive equations
−	Constitutive equations only valid for plane stress	Computational cost

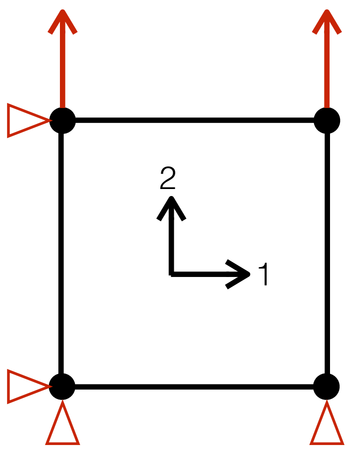
(V)UMAT subroutine

Using the VUMAT_SHELL.f subroutine and the VUHARD_V1.f subroutine:



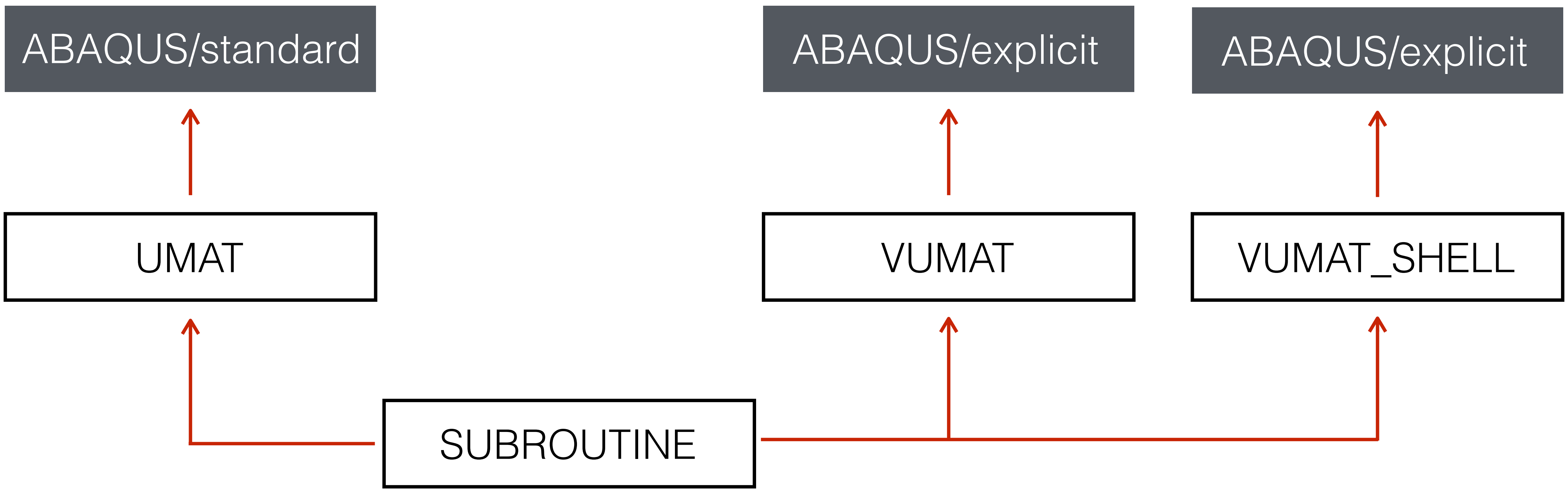


Single shell element in uniaxial tension



(V)UMAT subroutine

A simplified approach to UMAT/VUMAT coding:



+
- Easier to debug
- A model available in both ABAQUS/standard and explicit
- Both plane-stress and full 3D with one code

−
- Less efficient code