Writing a UMAT or VUMAT

Overview

- ABAQUS/Standard and ABAQUS/Explicit have interfaces that allow the user to implement general constitutive equations.
  - In ABAQUS/Standard the user-defined material model is implemented in user subroutine **UMAT**.
  - In ABAQUS/Explicit the user-defined material model is implemented in user subroutine **VUMAT**.
- Use **UMAT** and **VUMAT** when none of the existing material models included in the ABAQUS material library accurately represents the behavior of the material to be modeled.

Overview

- These interfaces make it possible to define any (proprietary) constitutive model of arbitrary complexity.
- User-defined material models can be used with any ABAQUS structural element type.
- Multiple user materials can be implemented in a single **UMAT** or **VUMAT** routine and can be used together.

In this lecture the implementation of material models in **UMAT** or **VUMAT** will be discussed and illustrated with a number of examples.

Motivation

• Proper testing of advanced constitutive models to simulate experimental results often requires complex finite element models.

- Advanced structural elements
- Complex loading conditions
- Thermomechanical loading
- Contact and friction conditions
- Static and dynamic analysis

Motivation

- Special analysis problems occur if the constitutive model simulates material instabilities and localization phenomena.
  - Special solution techniques are required for quasi-static analysis.
  - Robust element formulations should be available.
  - Explicit dynamic solution algorithms with robust, vectorized contact algorithms are desired.
- In addition, robust features are required to present and visualize the results.
  - Contour and path plots of state variables.
  - X-Y plots.
  - Tabulated results.

Motivation

• The material model developer should be concerned only with the development of the material model and not the development and maintenance of the FE software.

- Developments unrelated to material modeling
- Porting problems with new systems
- Long-term program maintenance of user-developed code

Steps Required in Writing a **UMAT** or **VUMAT**

- Proper definition of the constitutive equation, which requires one of the following:
  - Explicit definition of stress (Cauchy stress for large-strain applications)
  - Definition of the stress rate only (in corotational framework)
- Furthermore, it is likely to require:
  - Definition of dependence on time, temperature, or field variables
  - Definition of internal state variables, either explicitly or in rate form

Steps Required in Writing a **UMAT** or **VUMAT**

- Transformation of the constitutive rate equation into an incremental equation using a suitable integration procedure:

    - Forward Euler (explicit integration)

    - Backward Euler (implicit integration)

    - Midpoint method

Steps Required in Writing a **UMAT** or **VUMAT**

This is the hard part! Forward Euler (explicit) integration methods are simple but have a stability limit, $|\Delta\varepsilon| < \Delta\varepsilon_{stab}$

where $\Delta\varepsilon_{stab}$ is usually less than the elastic strain magnitude.

- For explicit integration the time increment must be controlled.
- For implicit or midpoint integration the algorithm is more complicated and often requires local iteration. However, there is usually no stability limit.
- An incremental expression for the internal state variables must also be obtained.

Steps Required in Writing a **UMAT** or **VUMAT**

- Calculation of the (consistent) Jacobian (required for ABAQUS/Standard UMAT only).
  - For small-deformation problems (e.g., linear elasticity) or large-deformation problems with small volume changes (e.g., metal plasticity), the consistent Jacobian is

$$C = \frac{\partial \Delta \sigma}{\partial \Delta \varepsilon}$$

where $\Delta \sigma$ is the increment in (Cauchy) stress and $\Delta \varepsilon$ is the increment in strain. (In finite-strain problems, ε is an approximation to the logarithmic strain.)

Steps Required in Writing a **UMAT** or **VUMAT**

- This matrix may be nonsymmetric as a result of the constitutive equation or integration procedure.
- The Jacobian is often approximated such that a loss of quadratic convergence may occur.
- It is easily calculated for forward integration methods (usually the elasticity matrix).

Steps Required in Writing a **UMAT** or **VUMAT**

- Coding the UMAT or VUMAT:
  - Follow FORTRAN 77 or C conventions.
  - Make sure that the code can be vectorized (for VUMAT only, to be discussed later).
  - Make sure that all variables are defined and initialized properly.
  - Use ABAQUS utility routines as required.
  - Assign enough storage space for state variables with the *DEPVAR option.

Steps Required in Writing a **UMAT** or **VUMAT**

- Verifying the **UMAT** or **VUMAT** with a small (one element) input file.
  1. Run tests with all displacements prescribed to verify the integration algorithm for stresses and state variables.

     Suggested tests include:
     - Uniaxial
     - Uniaxial in oblique direction
     - Uniaxial with finite rotation
     - Finite shear
  2. Run similar tests with load prescribed to verify the accuracy of the Jacobian.

Steps Required in Writing a **UMAT** or **VUMAT**

3. Compare test results with analytical solutions or standard ABAQUS material models, if possible. If the above verification is successful, apply to more complicated problems.

UMAT Interface

- These input lines act as the interface to a UMAT in which isotropic hardening plasticity is defined.

***MATERIAL, NAME=ISOPLAS**
***USER MATERIAL, CONSTANTS=8, (UNSYMM)**
30.E6, 0.3, 30.E3, 0., 40.E3, 0.1, 50.E3, 0.5
***DEPVAR**
13
***INITIAL CONDITIONS, TYPE=SOLUTION**
Data line to specify initial solution-dependent variables
USER SUBROUTINES,(INPUT=file_name***)**

- The *USER MATERIAL option is used to input material constants for the **UMAT**. The unsymmetric equation solution technique will be used if the UNSYMM parameter is used.

UMAT Interface

- The UMAT subroutine header is shown below:

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

C
       INCLUDE 'ABA_PARAM.INC'
C
       CHARACTER*8 CMNAME
C
       DIMENSION STRESS(NTENS), STATEV(NSTATV), DDSDDE(NTENS, NTENS),
      1 DDSDDT(NTENS), DRPLDE(NTENS), STRAN(NTENS), DSTRAN(NTENS),
      2 PREDEF(1), DPRED(1), PROPS(NPROPS), COORDS(3), DROT(3, 3),
      3 DFGRD0(3, 3), DFGRD1(3, 3)
```

- The include statement sets the proper precision for floating point variables (REAL*8 on most machines).
- The material name, CMNAME, is an 8-byte character variable.

UMAT Interface

UMAT Variables

- The following quantities are available in **UMAT**:
  - Stress, strain, and **SDVs** at the start of the increment
  - Strain increment, rotation increment, and deformation gradient at the start and end of the increment
  - Total and incremental values of time, temperature, and user-defined field variables
  - Material constants, material point position, and a characteristic element length
  - Element, integration point, and composite layer number (for shells and layered solids)
  - Current step and increment numbers

UMAT Interface

UMAT Variables

- The following quantities must be defined:
  - Stress, SDVs, and material Jacobian
- The following variables may be defined:
  - Strain energy, plastic dissipation, and "creep" dissipation
  - Suggested new (reduced) time increment

Complete descriptions of all parameters are provided in the UMAT section in Chapter 24 of the ABAQUS/Standard User's Manual.

UMAT Interface

- The header is usually followed by dimensioning of local arrays. It is good practice to define constants via parameters and to include comments.

```
      DIMENSION EELAS(6), EPLAS(6), FLOW(6)
C
      PARAMETER(ZERO=0.D0, ONE=1.D0, TWO=2.D0, THREE=3.D0, SIX=6.D0,
     1          ENUMAX=.4999D0, NEWTON=10, TOLER=1.0D-6)
C
C -----------------------------------------------------------------
C    UMAT FOR ISOTROPIC ELASTICITY AND ISOTROPIC MISES PLASTICITY
C    CANNOT BE USED FOR PLANE STRESS
C -----------------------------------------------------------------
C    PROPS(1) - E
C    PROPS(2) - NU
C    PROPS(3..) - YIELD AND HARDENING DATA
C    CALLS UHARD FOR CURVE OF YIELD STRESS VS. PLASTIC STRAIN
C -----------------------------------------------------------------
```

- The PARAMETER assignments yield accurate floating point constant definitions on any platform.

UMAT Interface

UMAT Utilities

- Utility routines SINV, SPRINC, SPRIND, and ROTSIG can be called to assist in coding **UMAT**.
  - SINV will return the first and second invariants of a tensor.
  - SPRINC will return the principal values of a tensor.
  - SPRIND will return the principal values and directions of a tensor.
  - ROTSIG will rotate a tensor with an orientation matrix.
  - XIT will terminate an analysis and close all files associated with the analysis properly.
- For details regarding the arguments required in making these calls, refer to the **UMAT** section in Chapter 24 of the ABAQUS/Standard User's Manual and the examples in this lecture.

UMAT Interface

UMAT Conventions

- Stresses and strains are stored as vectors.
  - For plane stress elements: σ_{11} , σ_{22} , σ_{12} .
  - For (generalized) plane strain and axisymmetric elements: σ_{11} , σ_{22} , σ_{33} , σ_{12} .
  - For three-dimensional elements: σ_{11} , σ_{22} , σ_{33} , σ_{12} , σ_{13} , σ_{23} .
- The shear strain is stored as engineering shear strain,

$$\gamma_{12} = 2\varepsilon_{12}.$$

- The deformation gradient, F_{ij}, is always stored as a three-dimensional matrix.

UMAT Interface

UMAT Formulation Aspects

- For geometrically nonlinear analysis the strain increment and incremental rotation passed into the routine are based on the Hughes-Winget formulae.
  - Linearized strain and rotation increments are calculated in the mid-increment configuration.
  - Approximations are made, particularly if rotation increments are large: more accurate measures can be obtained from the deformation gradient if desired.
- The user must define the Cauchy stress: when this stress reappears during the next increment, it will have been rotated with the incremental rotation, DROT, passed into the subroutine.
  - The stress tensor can be rotated back using the utility routine ROTSIG if this is not desired.

UMAT Interface

UMAT Formulation Aspects

- If the ∗ORIENTATION option is used in conjunction with **UMAT**, stress and strain components will be in the local system (again, this basis system rotates with the material in finite-strain analysis).
- Tensor state variables must be rotated in the subroutine (use ROTSIG).
- If **UMAT** is used with reduced-integration elements or shear flexible shell or beam elements, the hourglass stiffness and the transverse shear stiffness must be specified with the

∗HOURGLASS STIFFNESS and ∗TRANSVERSE SHEAR STIFFNESS options, respectively.

UMAT Interface

Usage Hints

- At the start of a new increment, the strain increment is extrapolated from the previous increment.
  - This extrapolation, which may sometimes cause trouble, is turned off with *STEP, EXTRAPOLATION=NO.
- If the strain increment is too large, the variable PNEWDT can be used to suggest a reduced time increment.
  - The code will abandon the current time increment in favor of a time increment given by PNEWDT*DTIME.
- The characteristic element length can be used to define softening behavior based on fracture energy concepts.

Example 1: Isotropic Isothermal Elasticity

Governing Equations

- Isothermal elasticity equation (with Lamé's constants):

$$\sigma_{ij} = \lambda \delta_{ij} \varepsilon_{kk} + 2\mu \varepsilon_{ij}$$

or in a Jaumann (corotational) rate form:

$$\dot{\sigma}_{ij}^{J} = \lambda \delta_{ij} \dot{\varepsilon}_{kk} + 2\mu \dot{\varepsilon}_{ij}$$

- The Jaumann rate equation is integrated in a corotational framework:

$$\Delta \sigma^{J}{}_{ij} = \lambda \delta_{ij} \Delta \varepsilon_{kk} + 2\mu \Delta \varepsilon_{ij}$$

The appropriate coding is shown on the following pages.

Coding for Isotropic Isothermal Elasticity

```
C -----------------------------------------------------------
C        UMAT FOR ISOTROPIC ELASTICITY
C        CANNOT BE USED FOR PLANE STRESS
C -----------------------------------------------------------
C         PROPS(1) - E
C         PROPS(2) - NU
C -----------------------------------------------------------
C
      IF (NDI.NE.3) THEN
          WRITE (7, *) 'THIS UMAT MAY ONLY BE USED FOR ELEMENTS
     1 WITH THREE DIRECT STRESS COMPONENTS'
        CALL XIT
      ENDIF
C
C         ELASTIC PROPERTIES
            EMOD=PROPS(1)
            ENU=PROPS(2)
            EBULK3=EMOD/(ONE-TWO*ENU)
            EG2=EMOD/(ONE+ENU)
            EG=EG2/TWO
            EG3=THREE*EG
            ELAM=(EBULK3-EG2)/THREE
```

Coding for Isotropic Isothermal Elasticity

```
C
C          ELASTIC STIFFNESS
C
             DO K1=1, NDI
               DO K2=1, NDI
                    DDSDDE(K2, K1)=ELAM
               END DO
               DDSDDE(K1, K1)=EG2+ELAM
             END DO
           DO K1=NDI+1, NTENS
               DDSDDE(K1 ,K1)=EG
           END DO
C
C          CALCULATE STRESS
C
           DO K1=1, NTENS
              DO K2=1, NTENS
                   STRESS(K2)=STRESS(K2)+DDSDDE(K2, K1)*DSTRAN(K1)
              END DO
           END DO
C
           RETURN
           END
```

Example 1: Isotropic Isothermal Elasticity

Remarks

- This very simple **UMAT** yields exactly the same results as the ABAQUS *ELASTIC option.
  - This is true even for large-strain calculations: all necessary large-strain contributions are generated by ABAQUS.
- The routine can be used with and without the *ORIENTATION option.
- It is usually straightforward to write a single routine that handles (generalized) plane strain, axisymmetric, and three-dimensional geometries.
  - Generally, plane stress must be treated as a separate case because the stiffness coefficients are different.
- The routine is written in incremental form as a preparation for subsequent elastic-plastic examples.

Example 1: Isotropic Isothermal Elasticity

Remarks

- Even for linear analysis, **UMAT** is called twice for the first iteration of each increment: once for assembly and once for recovery.
  Subsequently, it is called once per iteration: assembly and recovery are combined.
- A check is performed on the number of direct stress components, and the analysis is terminated by calling the subroutine, XIT.
  - A message is written to the message file (unit=7).

VUMAT Interface

- These input lines act as the interface to a **VUMAT** in which isotropic hardening plasticity is defined.

```
*MATERIAL, NAME=KINPLAS
*USER MATERIAL, CONSTANTS=4
 30.E6, 0.3, 30.E3, 40.E3
*DEPVAR
 5
*INITIAL CONDITIONS, TYPE=SOLUTION
```

Data line to specify initial solution-dependent variables

VUMAT Interface

- The VUMAT subroutine header is shown below:

```
      SUBROUTINE VUMAT(
C Read only -
     1  NBLOCK, NDIR, NSHR, NSTATEV, NFIELDV, NPROPS, LANNEAL,
     2  STEPTIME, TOTALTIME, DT, CMNAME, COORDMP, CHARLENGTH,
     3  PROPS, DENSITY, STRAININC, RELSPININC,
     4  TEMPOLD, STRETCHOLD, DEFGRADOLD, FIELDOLD,
     5 STRESSOLD, STATEOLD, ENERINTERNOLD, ENERINELASOLD,
     6 TEMPNEW, STRETCHNEW, DEFGRADNEW, FIELDNEW,
C Write only -
     7 STRESSNEW, STATENEW, ENERINTERNNEW, ENERINELASNEW)
C
       INCLUDE 'VABA_PARAM.INC'
C
```

VUMAT Interface

- The VUMAT subroutine header is shown below:

```
      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),DEFGRADOLD(NBLOCK,NDIR+NSHR+NSHR),
     4   FIELDOLD(NBLOCK, NFIELDV), STRESSOLD(NBLOCK, NDIR+NSHR),
     5  STATEOLD(NBLOCK, NSTATEV), ENERINTERNOLD(NBLOCK),
     6  ENERINELASOLD(NBLOCK), TEMPNEW(NBLOCK),
     7  STRETCHNEW(NBLOCK, NDIR+NSHR),DEFGRADNEW(NBLOCK,NDIR+NSHR+NSHR),
     8  FIELDNEW(NBLOCK, NFIELDV), STRESSNEW(NBLOCK,NDIR+NSHR),
     9  STATENEW(NBLOCK, NSTATEV), ENERINTERNNEW(NBLOCK),
     1  ENERINELASNEW(NBLOCK)
C
      CHARACTER*8 CMNAME
```

VUMAT Interface

VUMAT Variables

- The following quantities are available in VUMAT, but they cannot be redefined:
  - Stress, stretch, and SDVs at the start of the increment
  - Relative rotation vector and deformation gradient at the start and end of an increment and strain increment
  - Total and incremental values of time, temperature, and user-defined field variables at the start and end of an increment
  - Material constants, density, material point position, and a characteristic element length
  - Internal and dissipated energies at the beginning of the increment
  - Number of material points to be processed in a call to the routine (NBLOCK)

VUMAT Interface

VUMAT Variables

- A flag indicating whether the routine is being called during an annealing process

- The following quantities must be defined:
  - Stress and SDVs at the end of an increment
- The following variables may be defined:
  - Internal and dissipated energies at the end of the increment

Many of these variables are equivalent or similar to those in UMAT.

Complete descriptions of all parameters are provided in the VUMAT section in Chapter 21 of the ABAQUS/Explicit User's Manual.

VUMAT Interface
Comparison of VUMAT and UMAT Interfaces

There are a number of significant differences between the **UMAT** and **VUMAT** interfaces.

- **VUMAT** uses a two-state architecture: the initial values are in the OLD arrays, the new values must be put in the NEW arrays.
- The **VUMAT** interface is written to take advantage of vector processing.
- The material Jacobian does not need to be defined.
- No information is provided about element numbers.
- The time increment cannot be redefined.
- Utility routines are not available because they would prevent vectorization.

VUMAT Interface
Comparison of VUMAT and UMAT Interfaces

- The header is usually followed by dimensioning of local arrays. It is good practice to define constants via parameters and to include comments.

```
      PARAMETER( ZERO = 0.D0, ONE = 1.D0, TWO = 2.D0, THREE = 3.D0,
     1   THIRD = 1.D0/3.D0, HALF = .5D0, TWO_THIRDS = 2.D0/3.D0,
     2   THREE_HALFS = 1.5D0 )
C  J2 Mises Plasticity with kinematic hardening for plane strain case.
C  The state variables are stored as:
C       STATE(*, 1) = back stress component 11
C       STATE(*, 2) = back stress component 22
C       STATE(*, 3) = back stress component 33
C       STATE(*, 4) = back stress component 12
C       STATE(*, 5) = equivalent plastic strain
```

- The PARAMETER assignments yield accurate floating point constant definitions on any platform.

VUMAT Interface
VUMAT Conventions

- Stresses and strains are stored as vectors.
  - For plane stress elements: σ_{11} , σ_{22} , σ_{12} .
  - For plane strain and axisymmetric elements: σ_{11} , σ_{22} , σ_{33} , σ_{12} .
  - For three-dimensional elements: σ_{11} , σ_{22} , σ_{33} , σ_{12} , σ_{23} , σ_{13} .

    For three-dimensional elements, this storage scheme is inconsistent with that for ABAQUS/Standard.
- The shear strain is stored as tensor shear strains:

$$\varepsilon_{12} = \frac{1}{2}\gamma_{12}$$

VUMAT Interface
VUMAT Conventions

- The deformation gradient is stored similar to the way in which symmetric tensors are stored.
  - For plane stress elements: F_{11}, F_{22}, F_{12}, F_{21}.
  - For plane strain and axisymmetric elements: F_{11}, F_{22}, F_{33}, F_{12}, F_{21}.
  - For three-dimensional elements: F_{11}, F_{22}, F_{33}, F_{12}, F_{23}, F_{31}, F_{21}, F_{32}, F_{13}.

VUMAT Interface
VUMAT Formulation Aspects
Vectorized Interface

- In VUMAT the data are passed in and out in large blocks (dimension NBLOCK). NBLOCK typically is equal to 64 or 128.
  - Each entry in an array of length NBLOCK corresponds to a single material point. All material points in the same block have the same material name and belong to the same element type.
- This structure allows vectorization of the routine.
  - A vectorized **VUMAT** should make sure that all operations are done in vector mode with NBLOCK the vector length.
- In vectorized code, branching inside loops should be avoided.
  - Element type-based branching should be outside the NBLOCK loop.

VUMAT Interface
Corotational Formulation

- The constitutive equation is formulated in a corotational framework, based on the Jaumann stress rate.
  - The strain increment is obtained with Hughes-Winget.
  - Other measures can be obtained from the deformation gradient.
- The user must define the Cauchy stress: this stress reappears during the next increment as the “old” stress.
- There is no need to rotate tensor state variables.

VUMAT Interface
Corotational Formulation

- The constitutive equation is formulated in a corotational framework, based on the Jaumann stress rate.
  - The strain increment is obtained with Hughes-Winget.
  - Other measures can be obtained from the deformation gradient.
- The user must define the Cauchy stress: this stress reappears during the next increment as the "old" stress.
- There is no need to rotate tensor state variables.