Implementation of a material model in the Abaqus

New features can be added to Abaqus by writing “User subroutines”.
Unfortunately, (Fortran) codes for the UMAT and for the VUMAT differ.

Uncommon features can be simulated by FE-method as follows:

- I Commercial code and its existing feature.
- II Commercial code where you add new features.
- III Free-ware code with existing feature of added feature.
- IV In-house code. (Be careful. Huge amount of work may be needed!!!)

Topic of this lecture. How to prepare a mechanical material model for UMAT.
ABAQUS/Standard; UMAT
ABAQUS/Explicit; VUMAT

ABAQUS User Subroutine Reference Manual kohta 1.1.31 (v. 6.6) starts:

WARNING: The use of this option generally requires considerable expertise. The user is 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.

Usually I have had two different FEM models for testing the material model:
Small model with 1 element.
Large model with 5-6 elements.

Let the material model have the form

$$\dot{\boldsymbol{\varepsilon}} = \dot{\boldsymbol{\varepsilon}}^{\mathrm{e}} + \dot{\boldsymbol{\varepsilon}}^{\mathrm{Th}} + \dot{\boldsymbol{\varepsilon}}^{\mathrm{v}} \quad \Longrightarrow \quad \boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^{\mathrm{e}} + \boldsymbol{\varepsilon}^{\mathrm{Th}} + \boldsymbol{\varepsilon}^{\mathrm{v}}. \tag{1}$$

In Equation (1) $\boldsymbol{\varepsilon}^{\mathrm{e}}$ elastic strain.
$\boldsymbol{\varepsilon}^{\mathrm{Th}}$ thermal strain. (2)
$\boldsymbol{\varepsilon}^{\mathrm{v}}$ viscous strain (creep strain).

Material models for strain tensors $\boldsymbol{\varepsilon}^{\mathrm{e}}$ and $\boldsymbol{\varepsilon}^{\mathrm{Th}}$ are modelled as follows:

$$\boldsymbol{\sigma} = \mathbf{C}:\boldsymbol{\varepsilon}^{\mathrm{e}} \qquad \text{and} \qquad \boldsymbol{\varepsilon}^{\mathrm{Th}} = \hat{\alpha}\,(T - T_{\mathrm{r}})\,\mathbf{1}, \tag{3}$$

where $(T - T_{\mathrm{r}})$ is temperature difference between current and reference state.

Viscous strain tensor $\boldsymbol{\varepsilon}^{\mathrm{v}}$ is assumed to be modelled as follows:

$$\dot{\boldsymbol{\varepsilon}}^{\mathrm{v}} = \overset{\circ}{\varepsilon}_{\mathrm{re}} \left[\frac{\left\langle J_{\mathrm{vM}}(\boldsymbol{\sigma} - \boldsymbol{\beta}^1) - \beta^2 - \sigma_{\mathrm{tr}} \right\rangle}{\sigma_{\mathrm{re}}} \right]^n \frac{\mathbf{s} - \mathbf{b}^1}{J_{\mathrm{vM}}(\boldsymbol{\sigma} - \boldsymbol{\beta}^1)}, \tag{4}$$

where the kinematic hardening $\boldsymbol{\beta}^1$ takes the following appearance:

$$\dot{\boldsymbol{\beta}}^1 = m^1 \dot{\boldsymbol{\varepsilon}}^{\mathrm{v}} - m^2 \left[\frac{J_{\mathrm{vM}}(\boldsymbol{\beta}^1)}{\sigma_{\mathrm{re}}} \right]^k \frac{\mathbf{b}^1}{J_{\mathrm{vM}}(\boldsymbol{\beta}^1)} \tag{5}$$

and the isotropic hardening β^2 is obtained from

$$\dot{\beta}^2 = n^1 J_{\mathrm{vM}}(\dot{\boldsymbol{\varepsilon}}^{\mathrm{v}}) - n^2 \left(\frac{\beta^2}{\sigma_{\mathrm{re}}} \right)^m. \tag{6}$$

Task for the UMAT subroutine

Task for the UMAT subroutine can be expressed for short as follows:
Values of the variables passed in to UMAT are given at time t.
Values of the variables are updated in UMAT for the time $t + \Delta t$.

Euler forward method is used for computation of values of functions at time $t + \Delta t$.

$$f(t + \Delta t) = f(t) + \dot{f}(t)\,\Delta t \tag{7}$$

or

$$f(t + \Delta t) = f(t) + \Delta f(t), \qquad \text{where} \qquad \Delta f(t) := \dot{f}(t)\,\Delta t. \tag{8}$$

Euler forward method is uneffective but easy to code.
It is a good practice first to use Euler forward method.
Then (if necessary) Euler backward can be applied.

Variables to be defined in UMAT

Values of variables are known in beginning of increment, i.e. at time t.
Variables passes in for information.

UMAT computes values of variables at end of increment, i.e. at time $t + \Delta t$.
Variables to be defined.

The model given on the previous page is investigated.

One space integration point is studied separately in the UMAT subroutine.
Usually there is no need to know which integration point is studied.
Hardening takes different values in different parts of the body.
This is taken into account by STATEV variables.

The following names (such as NTENS) are due to Abaqus UMAT.
Descriptions of variables are copied from the Abaqus manual. (not exactly)
Variables (such as $\Delta\boldsymbol{\varepsilon}$) are due to the author.

Variables passed in for information.
Values of these variables are available in UMAT.
Total strains at the beginning of the increment: STRAN(NTENS) = $\boldsymbol{\varepsilon}(t)$.
Number of stress or strain components: NTENS $(\sigma_{11},\sigma_{22},\sigma_{12})$ NTENS=3.
Strain increments during the time increment Δt: DSTRAN(NTENS) = $\Delta\boldsymbol{\varepsilon}$.
Stresses at the beginning of the increment: STRESS(NTENS) = $\boldsymbol{\sigma}(t)$.
Total time at the beginning of the increment: TIME(2) = t.
Time increment: DTIME = Δt.
Increment of the temperature: DTEMP = ΔT.

Values of the material parameters: PROPS(NPROPS).

For example: STRESS(1) = σ_{11}
STRESS(2) = σ_{22}
STRESS(3) = σ_{12}

STRESS(NTENS) This array is passed in as the stress tensor at the beginning of the increment and must be updated in this routine to be the stress tensor at the end of the increment.

Viscous strain increment tensor $\Delta\boldsymbol{\varepsilon}^{\mathrm{v}}$ is obtained from

$$\Delta\boldsymbol{\varepsilon}^{\mathrm{v}} = \dot{\boldsymbol{\varepsilon}}^{\mathrm{v}}(t)\,\Delta t\,. \tag{9}$$

Viscous strain rate tensor $\dot{\boldsymbol{\varepsilon}}^{\mathrm{v}}(t)$ present in Eq. (9) is obtained from Eq. (4), viz.

$$\dot{\boldsymbol{\varepsilon}}^{\mathrm{v}} = \overset{\circ}{\varepsilon}_{\mathrm{re}}\left[\frac{\left\langle J_{\mathrm{vM}}(\boldsymbol{\sigma}-\boldsymbol{\beta}^{1}) - \beta^{2} - \sigma_{\mathrm{tr}}\right\rangle}{\sigma_{\mathrm{re}}}\right]^{n}\frac{\mathbf{s}-\mathbf{b}^{1}}{J_{\mathrm{vM}}(\boldsymbol{\sigma}-\boldsymbol{\beta}^{1})}, \tag{4}$$

Based on Eq. $(3)_2$ the thermal strain increment tensor $\Delta\boldsymbol{\varepsilon}^{\mathrm{Th}}$ is

$$\Delta\boldsymbol{\varepsilon}^{\mathrm{Th}} = \hat{\alpha}\,\Delta T\,\mathbf{1}. \tag{10}$$

Note! ΔT present in Eq. (10) is the temperature increment during Δt.

Equation $(1)_1$ is recalled. It is

$$\dot{\boldsymbol{\varepsilon}} = \dot{\boldsymbol{\varepsilon}}^{\mathrm{e}} + \dot{\boldsymbol{\varepsilon}}^{\mathrm{Th}} + \dot{\boldsymbol{\varepsilon}}^{\mathrm{v}}\,. \tag{$1)_1$}$$

Equation $(1)_1$ gives the elastic strain increment tensor $\Delta\boldsymbol{\varepsilon}^{\mathrm{e}}$, viz.

$$\dot{\boldsymbol{\varepsilon}}^{\mathrm{e}} = \dot{\boldsymbol{\varepsilon}} - (\dot{\boldsymbol{\varepsilon}}^{\mathrm{v}} + \dot{\boldsymbol{\varepsilon}}^{\mathrm{Th}}) \quad\Longrightarrow\quad \Delta\boldsymbol{\varepsilon}^{\mathrm{e}} = \Delta\boldsymbol{\varepsilon} - (\Delta\boldsymbol{\varepsilon}^{\mathrm{v}} + \Delta\boldsymbol{\varepsilon}^{\mathrm{Th}}). \tag{11}$$

Equation $(3)_1$ reads

$$\boldsymbol{\sigma} = \mathbf{C}:\boldsymbol{\varepsilon}^{\mathrm{e}} \quad\Longrightarrow\quad \Delta\boldsymbol{\sigma} = \mathbf{C}:\Delta\boldsymbol{\varepsilon}^{\mathrm{e}}\,. \tag{12}$$

Equations $(11)_2$ and $(12)_2$ yield

$$\Delta\boldsymbol{\sigma} = \mathbf{C}:\Delta\boldsymbol{\varepsilon}^{\mathrm{e}} = \mathbf{C}:\left[\Delta\boldsymbol{\varepsilon} - (\Delta\boldsymbol{\varepsilon}^{\mathrm{v}} + \Delta\boldsymbol{\varepsilon}^{\mathrm{Th}})\right]. \tag{12a}$$

Equation (8) is

$$f(t+\Delta t) = f(t) + \Delta f(t)\,, \qquad \text{where} \qquad \Delta f(t) := \dot{f}(t)\,\Delta t. \tag{8}$$

Equation $(8)_1$ gives

$$\boldsymbol{\sigma}(t+\Delta t) = \boldsymbol{\sigma}(t) + \Delta\boldsymbol{\sigma}\,. \tag{12b}$$

DDSDDE(NTENS, NTENS) The value of the Jacobian matrix of the constitutive model. According to ABAQUS USRM (v. 6.6, p. 1.1.31) the Jacobian matrix is $\partial\Delta\boldsymbol{\sigma}/\partial\Delta\boldsymbol{\varepsilon}$, for the mechanical constitutive model. It's value has to be determined at the end of the increment.

Equation $(1)_1$ is

$$\dot{\boldsymbol{\varepsilon}} = \dot{\boldsymbol{\varepsilon}}^{e} + \dot{\boldsymbol{\varepsilon}}^{Th} + \dot{\boldsymbol{\varepsilon}}^{v} \quad \Longrightarrow \quad \mathbf{C}:\Delta\boldsymbol{\varepsilon} = \mathbf{C}:\Delta\boldsymbol{\varepsilon}^{e} + \mathbf{C}:\Delta\boldsymbol{\varepsilon}^{Th} + \mathbf{C}:\Delta\boldsymbol{\varepsilon}^{v}. \quad (13)$$

Equation (12b) is

$$\Delta\boldsymbol{\sigma} = \mathbf{C}:\Delta\boldsymbol{\varepsilon}^{e}. \quad (12b)$$

Substitution of Eq. (12b) into Equation $(13)_2$ yields

$$\mathbf{C}:\Delta\boldsymbol{\varepsilon} = \Delta\boldsymbol{\sigma} + \mathbf{C}:\Delta\boldsymbol{\varepsilon}^{Th} + \mathbf{C}:\Delta\boldsymbol{\varepsilon}^{v} \quad \Longrightarrow \quad \Delta\boldsymbol{\sigma} = \mathbf{C}:\Delta\boldsymbol{\varepsilon} - \mathbf{C}:(\Delta\boldsymbol{\varepsilon}^{Th} + \Delta\boldsymbol{\varepsilon}^{v}). \quad (14)$$

Equation $(14)_2$ gives

$$\frac{\partial\Delta\boldsymbol{\sigma}}{\partial\Delta\boldsymbol{\varepsilon}} = \mathbf{C}:\frac{\partial\Delta\boldsymbol{\varepsilon}}{\partial\Delta\boldsymbol{\varepsilon}} = \mathbf{C}:\mathbf{I} = \mathbf{C}. \quad (15)$$

Result (15) can be written in the form

$$\frac{\partial\Delta\sigma_{ij}}{\partial\Delta\varepsilon_{st}} = C_{stij}. \quad (16)$$

If also plastic strain $\boldsymbol{\varepsilon}^{p}$ is modelled,

Tensor $\mathbf{C}$ in Eq. (16) has to be replaced by "elastic-plastic" tensor $\mathbf{C}^{ep}$.

Tensor $\mathbf{C}$ can be used when the implementation is tested.

STATEV (NSTATV) An array containing the (user defined) solution-dependent state variables. These are passed in as the values at the beginning of the increment and STATEV must be returned as the values at the end of the increment.

Internal forces $\boldsymbol{\beta}^1$ and β^2 are "user defined solution dependent state variables". STATEV variables.

Usually the STATEV variables are:

internal variables $(\boldsymbol{\alpha}^1, \alpha^2, ,\ldots)$

internal forces $(\boldsymbol{\beta}^1, \beta^2, \ldots)$

Kinematic and isotropic hardening parameters $\boldsymbol{\beta}^1$ and β^2 are updated as follows:

$$\boldsymbol{\beta}^1(t+\Delta t) = \boldsymbol{\beta}^1(t) + \dot{\boldsymbol{\beta}}^1(t)\,\Delta t \quad (18)$$

and

$$\beta^2(t+\Delta t) = \beta^2(t) + \dot{\beta}^2(t)\,\Delta t. \quad (19)$$

For example:

STATEV(1) = β^2

STATEV(2) = β^1_{11}

STATEV(3) = β^1_{12}

etc.

Implementation of 1D model in Abaqus UMAT subroutine

1D Material model given in Section 18.1 is studied (tension).
According to Eqs (18,18), (18.19), (18.20) and (19.21) the model reads

$$\varepsilon = \varepsilon^{e} + \varepsilon^{Th} + \varepsilon^{v}. \qquad (20)$$

where

$$\varepsilon^{e} = \frac{\sigma}{E} \quad \text{and} \quad \varepsilon^{Th} = \hat{\alpha}\,(T - T_r). \qquad (21)$$

and

$$\dot{\varepsilon}^{v} = \mathring{\varepsilon}_{re}\left(\frac{\sigma - \beta}{\sigma_{re}}\right)^{n} \quad \text{and} \quad \dot{\beta} = a\,\dot{\varepsilon}^{v} - b\left(\frac{\beta}{\sigma_{re}}\right)^{m}. \qquad (22)$$

Information on material parameters:

Position in *.inp file PROPS() vector	Fortran variable	Description
1	E	Young's modulus
2	Poisso	Poisson's ratio
3	Tr	Reference temperature
4	AlfaMa	Linear coeff. of thermal expansion $\hat{\alpha}$
5	EpsDRe	Reference strain rate $\dot{\varepsilon}_{re}$
6	SigRe	Reference stress σ_{re}
7		Creep exponent n
8	a	a
9	b	b
10		Exponent for internal force β (m)

For example! Above Table gives Tr = PROPS(3).

From the previous page:
Above Table gives Tr = PROPS(3).

If one wants to use the variable Tr, its value has to be given (as done above).
It take computing time.
If possible, use PROPS() variables.

In Fortran the variables which names start by characters I-N are usually integers.
Please, follow this custom.

Solution dependent state variables; STATEV(): DEPVAR = 1

Position in STATEV() vector	Description
1	Internal force β

UMAT updates STATEV() variables for every space integration point.
The values are available when same integr. point is studied next time in UMAT.
ABAQUS has an automatic bookkeeping on the integration points.

Variables generated by the programmer are lost during the exit from UMAT.
Values of the variables generated by the programmer depend on compiler.
SAVE command can be used for storing the programmers variables.

Programmers own variables in the UMAT (local variables)

Variable	Dim.	Type: R ≙ Real I ≙ Integer	Meaning
SigRe1		R	$1/\sigma_{re}$
SiBe		R	$\sigma - \beta$
EpsVD		R	$\dot{\varepsilon}^{v}$
DEpsV		R	$\Delta\varepsilon^{v}$
SeTerm		R	$b\,(\beta/\sigma_{re})^{m}$
DBeta		R	$\Delta\beta$
DSigma		R	$\Delta\sigma$

(1) First lines of the code are copied from the Abaqus manual (v. 6.6)

```
      SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
     1 RPL,DDSDDT,DRPLDE,DRPLDT,
     2 STRAN,DSTRAN,TIME,DTIME,TEMP,DTEMP,PREDEF,DPRED,
     * CMNAME,   ! This is in the manual in the previous line, but this is correct
     3 NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,DROT,
     * PNEWDT,
     4 CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,KSPT,KSTEP,KINC)
C
      INCLUDE 'ABA_PARAM_DP.INC' (check, if needed)
C
      CHARACTER*80 CMNAME
      DIMENSION STRESS(NTENS),STATEV(NSTATV),
     1 DDSDDE(NTENS,NTENS),DDSDDT(NTENS),DRPLDE(NTENS),
     2 STRAN(NTENS),DSTRAN(NTENS),TIME(2),PREDEF(1),DPRED(1),
     3 PROPS(NPROPS),COORDS(3),DROT(3,3),DFGRD0(3,3),DFGRD1(3,3)
C
C     DIMENSION ***  ! Here the dimensions of the user's variables
C
      CMNAME = 'Le Gac & Duval 1D Material Model' ! Name of model
```


(2) Changing the values for the material parameters

It takes computing time.

```
C
C***                              **********************************
C*** MATERIAL CONSTANTS ***********************************
C***                              **********************************
C*
C*** ABAQUS sends the values of the material constants into UMAT in
C*** PROPS()-array
C*
      E         = PROPS(1)
      Poisso    = PROPS(2)
      Tr        = PROPS(3)
      AlfaMa    = PROPS(4)
      EpsDRe    = PROPS(5)
      SigRe     = PROPS(6)          ! PROPS(7) and PROPS(10) are
      a         = PROPS(8)          ! missing, since they do not
      b         = PROPS(9)          ! have symbols
C*
C*** The above substitution takes time. Instead of the above use:
C*** PROPS()-vector [as with PROPS(7) and PROPS(10)]
```

Note! Lines starting with something (here C) in the 1st column are comments.
Charecter in the 6th column indicates continuation from previous line.
Coding such as E = , Poisso = etc. is started from the 7th column.
Line length is 80 columns
Thus, for real coding only 72 columns.
Command: I = 1 means that I is an integer and has exact value 1.
Command: S = 1.0 means the S is real variable and its value is ≈1.
Command: S = 1.0D0 means that S is a double precision variable.

Use this.

(3) Compute $\Delta\varepsilon^{v}$ and $\Delta\beta$

Equations (22) are

$$\dot{\varepsilon}^{v} = \mathring{\dot{\varepsilon}}_{re}\left(\frac{\sigma-\beta}{\sigma_{re}}\right)^{n} \quad \text{and} \quad \dot{\beta} = a\dot{\varepsilon}^{v} - b\left(\frac{\beta}{\sigma_{re}}\right)^{m}. \tag{22}$$

Equations (22) give

$$\Delta\varepsilon^{v} = \dot{\varepsilon}^{v}\Delta t = \mathring{\dot{\varepsilon}}_{re}\left(\frac{\sigma-\beta}{\sigma_{re}}\right)^{n}\Delta t \tag{23}$$

and

$$\Delta\beta = \dot{\beta}\Delta t = \left[a\dot{\varepsilon}^{v} - b\left(\frac{\beta}{\sigma_{re}}\right)^{m}\right]\Delta t. \tag{24}$$

Following part of the UMAT computes the value for $\Delta\varepsilon^{v}$.

```
C*
C***                                  ******************************
C*** SOME PRELIMINARY WORK ******************************
C***                                  ******************************
C*
      SigRe1 = 1.0D0 / SigRe  ! Since division takes much more time than
C                             ! multiplication, it is done only once. *
C*
C***                                  ********************
C*** INCREMENT OF THE VISCOUS STRAIN ********************
C***                                  ********************
C*
      SiBe   = STRESS(1) - STATEV(1)                ! Sigma - Beta
      EpsVD  = EpsDRe * ( SiBe * SigRe1 ) ** PROPS(7) ! Viscous strain rate
      DEpsV  = EpsVD * DTIME                        ! Delta Viscous strain
```

* Here the division is carried out only twice, but in 3D code several times

Following part of UMAT computes the value for $\Delta\beta$.

```
C*
C***                                  ***************
C*** INCREMENT OF THE INTERNAL FORCE BETA ***************
C***                                  ***************
C*
      SeTerm = b * ( STATEV(1) * SigRe1 ) ** PROPS(10)  ! Second term in
C                                                       ! Delta Beta
      DBeta  = ( a * EpsVD - SeTerm ) * DTIME           ! Delta Beta
```

(4) Value for internal force β at the end of the increment

Equation (24) is

$$\Delta\beta = \dot{\beta}\Delta t = \left[a\dot{\varepsilon}^{v} - b\left(\frac{\beta}{\sigma_{re}}\right)^{m}\right]\Delta t. \tag{24}$$

Equation (24) gives

$$\beta(t+\Delta t) = \beta(t) + \Delta\beta = \beta(t) + \dot{\beta}\Delta t = \beta(t) + \left[a\dot{\varepsilon}^{v} - b\left(\frac{\beta}{\sigma_{re}}\right)^{m}\right]\Delta t. \tag{25}$$

Following part of UMAT computes the value for $\beta(t+\Delta t)$.

```
C*
C***                                  ******
C*** VALUE OF THE BETA AT THE END OF THE INCREMENT ******
C***                                  ******
C*
      STATEV(1) = STATEV(1) + DBeta
```

(5) Value of the stress σ at the end of the increment

Material Model (20) reads

$$\varepsilon = \varepsilon^{e} + \varepsilon^{Th} + \varepsilon^{v}. \qquad (20)$$

Model (20) gives [compare with Equation (11)]

$$\Delta\varepsilon^{e} = \Delta\varepsilon - (\Delta\varepsilon^{v} + \Delta\varepsilon^{Th}). \qquad (26)$$

Material Model (21) is

$$\varepsilon^{e} = \frac{\sigma}{E} \quad \text{and} \quad \varepsilon^{Th} = \hat{\alpha}\,(T - T_{r}). \qquad (21)$$

Model $(21)_1$ gives [Compare with Equation (12a)]

$$\Delta\sigma = E\,\Delta\varepsilon^{e} \quad \Longrightarrow \quad \sigma(t + \Delta t) = \sigma(t) + \Delta\sigma. \qquad (27)$$

Model $(21)_2$ gives

$$\Delta\varepsilon^{Th} = \hat{\alpha}\,\Delta T. \qquad (28)$$

In Eq. (28) the term ΔT is the temperature increment during the time increment Δt. In Eq. $(21)_2$ $(T - T_r)$ is difference between the present ant the reference value.
Reference temperature T_r can be something else than temperature at time $t = 0$.

Substitution of Eqs (26) and (28) into Equation $(27)_1$ gives

$$\Delta\sigma = E\,\Delta\varepsilon^{e} = E\left[\Delta\varepsilon - (\Delta\varepsilon^{v} + \Delta\varepsilon^{Th})\right] = E\left[\Delta\varepsilon - (\Delta\varepsilon^{v} + \hat{\alpha}\,\Delta T)\right]. \qquad (29)$$

Now Equation $(27)_2$ can be applied.
Following part of UMAT calculates the value for $\sigma(t + \Delta t)$.

```
C*
C***                                                          **
C*** VALUE OF THE STRESS TENSOR AT THE END OF THE INCR        **
C***                                                          **
C*
      DSigma = E * ( DSTRAN(1) - ( DEpsV + AlfaMa * DTEMP ) )
      STRESS(1) = STRESS(1) + DSigma
```


(6) Value of the Jacobian matrix $\partial\Delta\boldsymbol{\sigma}/\partial\Delta\boldsymbol{\varepsilon}$ at the end of the increment

Matrix $\partial\Delta\boldsymbol{\sigma}/\partial\Delta\boldsymbol{\varepsilon}$ gives an answer to the following question;

What would be the change for value of $\Delta\boldsymbol{\sigma}$, if the value of $\Delta\boldsymbol{\varepsilon}$ would deviate from that predicted by Abaqus.

Value for the matrix $\partial\Delta\boldsymbol{\sigma}/\partial\Delta\boldsymbol{\varepsilon}$ is computed at the end of the increment.

Result given in Equation (29) is

$$\Delta\sigma = E\left[\Delta\varepsilon - (\Delta\varepsilon^{v} + \Delta\varepsilon^{Th})\right]. \qquad (30)$$

Equation (30) gives the following derivative:

$$\frac{\partial\Delta\sigma}{\partial\Delta\varepsilon} = E. \qquad (31)$$

Following part of UMAT computes value for $\partial\Delta\sigma/\partial\Delta\varepsilon$ at the end of the increment.

```
C*
C***
C*** JACOBIAN MATRIX OF THE CONST. EQ. AT THE END OF THE INCR
C***
C*
      DDSDDE(1,1) = E
C*
      RETURN
      END
```

Differences between UMAT and VUMAT

(a) Components in the stress vector and strain vector differ.

UMAT:

$$\begin{Bmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ \gamma_{12}\end{Bmatrix} \text{ and } \begin{Bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{12}\end{Bmatrix} \quad \text{or} \quad \begin{Bmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ \varepsilon_{33}\\ \gamma_{12}\\ \gamma_{13}\\ \gamma_{23}\end{Bmatrix} \text{ and } \begin{Bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{33}\\ \sigma_{12}\\ \sigma_{13}\\ \sigma_{23}\end{Bmatrix}. \qquad (32)$$

VUMAT:

4 components

$$\begin{Bmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ \varepsilon_{33}\\ \varepsilon_{12}\end{Bmatrix} \text{ and } \begin{Bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{33}\\ \sigma_{12}\end{Bmatrix} \quad \text{or} \quad \begin{Bmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ \varepsilon_{33}\\ \varepsilon_{12}\\ \varepsilon_{23}\\ \varepsilon_{31}\end{Bmatrix} \text{ and } \begin{Bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{33}\\ \sigma_{12}\\ \sigma_{23}\\ \sigma_{31}\end{Bmatrix}. \qquad (32)$$

2D 3D

(b) Jacobian matrix $\partial\Delta\boldsymbol{\sigma}/\partial\Delta\boldsymbol{\varepsilon}$ is not needed in VUMAT.

(c) Include commands

include ‘aba_param.inc’
include ‘vaba_param.inc’

Note! Include commands may change.

The above parts of the code are collected here.

```
      SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
     1 RPL,DDSDDT,DRPLDE,DRPLDT,
     2 STRAN,DSTRAN,TIME,DTIME,TEMP,DTEMP,PREDEF,DPRED,
     * CMNAME,   ! This is in the manual in the previous line, but this is correct
     3 NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,DROT,
     *PNEWDT,
     4 CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,KSPT,KSTEP,KINC)
C
      INCLUDE 'ABA_PARAM_DP.INC' (check, if needed)
C
      CHARACTER*80 CMNAME
      DIMENSION STRESS(NTENS),STATEV(NSTATV),
     1 DDSDDE(NTENS,NTENS),DDSDDT(NTENS),DRPLDE(NTENS),
     2 STRAN(NTENS),DSTRAN(NTENS),TIME(2),PREDEF(1),DPRED(1),
     3 PROPS(NPROPS),COORDS(3),DROT(3,3),DFGRD0(3,3),DFGRD1(3,3)
C
C     DIMENSION ***  ! Here the dimensions of the user's variables
C
      CMNAME = 'Le Gac & Duval 1D Material Model' ! Name of model
C
C***                              *********************************
C*** MATERIAL CONSTANTS *********************************
C***                              *********************************
C*
C*** ABAQUS sends the values of the material constants into UMAT in
C*** PROPS()-array
C*
      E          = PROPS(1)
      Poisso     = PROPS(2)
      Tr         = PROPS(3)
      AlfaMa     = PROPS(4)
      EpsDRe     = PROPS(5)
      SigRe      = PROPS(6)          ! PROPS(7) and PROPS(10) are
      a          = PROPS(8)          ! missing, since they do not
      b          = PROPS(9)          ! have symbols
```



```
C*
C*** The above substitution takes time. Instead of the above use:
C*** PROPS()-vector [as with PROPS(7) and PROPS(10)]
C*
C***                                 ******************************
C*** SOME PRELIMINARY WORK ******************************
C***                                 ******************************
C*
      SigRe1 = 1.0D0 / SigRe  ! Since division takes much more time than
C                             ! multiplication, it is done only once. *
C*
C***                                          *********************
C*** INCREMENT OF THE VISCOUS STRAIN *********************
C***                                          *********************
C*
      SiBe   = STRESS(1) - STATEV(1)                  ! Sigma - Beta
      EpsVD  = EpsDRe * ( SiBe * SigRe1 ) ** PROPS(7) ! Viscous strain r
      DEpsV  = EpsVD * DTIME                          ! Delta Viscous strain
C*
C***                                                  **************
C*** INCREMENT OF THE INTERNAL FORCE BETA **************
C***                                                  **************
C*
    SeTerm = b * ( STATEV(1) * SigRe1 ) ** PROPS(10)  ! Second term in
C                                                      ! Delta Beta
    DBeta  = ( a * EpsVD - SeTerm ) * DTIME           ! Delta Beta
C*
C***                                                          ******
C*** VALUE OF THE BETA AT THE END OF THE INCREMENT ******
C***                                                           ******
C*
    STATEV(1) = STATEV(1) + DBeta
C*
C***                                                            **
C*** VALUE OF THE STRESS TENSOR AT THE END OF THE INCR **
C***                                                             **
C*
```

```
      DSigma = E * ( DSTRAN(1) - ( DEpsV + AlfaMa * DTEMP )  )
      STRESS(1) = STRESS(1) + DSigma
C*
C***
C*** JACOBIAN MATRIX OF THE CONST. EQ. AT THE END OF THE INCR
C***
C*
      DDSDDE(1,1) = E
C*
      RETURN
      END
```