Short documentation of subroutines (UMAT, UHARD) for gradient-enhanced damage modeling

Andreas Seupel (Andreas.Seupel@tu-freiberg.de)

January 8, 2018

Institute of Mechanics and Fluid Dynamics, TU Bergakademie Freiberg, Lampadiusstrasse 4, 09599 Freiberg, Germany

1 Content

We provide a minimal model (UHARD) and a full model (UMAT), which contain the formulation of gradient-enhanced ductile damage described within the paper. The implementation is done in Fortran 90 for Abaqus-version 6.14-2. As compiler, Fortran Intel(R) 64 version 11.0 is utilized. The following files are attached in the main directory:

- UHARD.f (Fortran subroutine for user-defined hardening)
- UMAT.f (Fortran subroutine for user-defined material without element deletion. This has been used for simulations until initiation of an incipient crack and the convergence study).
- UMAT_ELEMENT_EROSION.f (Fortran subroutine for user-defined material with element deletion. This has been used for the crack growth example.)
- SHEAR_BAND_SPECIMEN.inp (Abaqus input-file containing the 2D-shear band specimen example with $b_{\mathrm{e}}/L = 1/4$ for UHARD-usage)
- DOUBLE_NOTCH_TENSILE_TEST.inp (Abaqus input-file containing the 2D double-notched tensile test with $b_{\mathrm{e}}/L = 1/2$ for UMAT-usage)
- All input-files of the mesh-convergence study and global convergence study

For parallel processing on several CPUs we use a fully parallel MPI setting by adding the following lines to the Abaqus environment file (abaqus_v6.env, check the indentations after the for-loop):

```
import os, socket
os.environ['ABA_BATCH_OVERRIDE'] = '1'
os.environ['ABA_MPI_SKIP_BUNCH_NODES'] = "1"
cpus = globals().get('cpus', locals().get('cpus', 1))
memory=str(90/cpus) + '%'
mp_host_list = []
for i in range(cpus):
    mp_host_list.append([socket.gethostname(), 1])
del i
```

Otherwise, computing with one CPU is recommended. To start a simulation, the UHARD or UMAT-subroutine has to be provided. The following terminal command can be used to start the test example for UHARD on a local workstation with an Abaqus-installation:

abq6142 job=SHEAR_BAND_SPECIMEN.inp user=UHARD.f

The UMAT-example is similarly executed:

abq6142 job=DOUBLE_NOTCH_TENSILE_TEST.inp user=UMAT_ELEMENT_EROSION.f cpus=4

Here, several CPUS should be used. More information concerning the examples are given within the subsequent sections.

2 Application

Both implementations (UHARD, UMAT) of gradient-enhanced damage need the definition of a coupled temperature displacement load step in Abaqus. A steady-state heat conduction has to be applied. The experienced reader is encouraged to have a look into the provided input-files. A definition with automatic step-size control is given by:

```
*Step, name=Step-1, nlgeom=YES, inc=1000
*Coupled Temperature-displacement, creep=none, steady state
0.001, 1., 1e-12, 0.001
```

For the double-notched tensile test, a numerical damping is used. The step definition is therefore:

```
*Step, name=Step-1, nlgeom=YES, inc=8000
*Coupled Temperature-displacement, ..., stabilize, factor=0.0002,
allsdtol=0, continue=NO
0.001, 1., 1e-12, 0.001
```

2.1 Material definition of UHARD

In order to signal the usage of a user-defined hardening (UHARD), the material definition reads:

```
*Material, name=Material-1
*Conductivity
0.25,
*Density
7.85e-09,
*Depvar
6,
*Heat Generation
*Elastic
525., 0.3
*Plastic, hardening=USER, properties=4
1., 0.75, 0., 0.4
```

Six user-defined state variables are defined (*Depvar). Four properties of hardening/softening are specified in the *Plastic-modul. Default description of elasticity is utilized (*Elastic). The *Conductivity is related to the internal length L:

$$\text{*Conductivity} = L^2.$$

The *Heat Generation is needed to incorporate the HETVAL-subroutine. The properties and depvar definitions are summarized in Tab. 1.

Table 1: Properties and Depvar (STATEV in post-processing) of UHARD-subroutine.

PROP No.	Symbol (as in paper)	Description
1	σ_0	initial yield stress
2	H	linear hardening modulus
3	ε_{i}	damage parameter, strain at damage initiation
4	ε_{c}	damage parameter, strain at failure
STATEV No.	Symbol (as in paper)	Description
1	r	source term (heat source)
2	κ	damage driving strain
3	D	damage variable
4	-	element deletion indicator (=0->deletion)
5	-	indicator for damage initiation (>0->initiation)
6	$\varepsilon_{\mathrm{eq}}$	equivalent plastic strain

2.2 Material definition of UMAT

In order to signal the usage of a user-defined material routine (UMAT), the material definition reads for example:

```
*Material, name=Material-1
*Conductivity
0.0625,
*Density
7.85e-09,
*Depvar,delete=12
12,
*Heat Generation
*User Material, constants=15, unsymm
525., 0.3, 1., 2., 1., 0., 0., 0.0
0.0, 0.0, 0., 0.0, 0.4, 0., 1.
```

Twelve user-defined state variables have to be defined (*Depvar). Depvar number 12 is chosen to act as element deletion signal. For visualization reasons, the field output STATUS has to be declared (see input-file). The *Conductivity-value is again related to the internal length L (see above). The *Heat Generation is needed to incorporate the HETVAL-subroutine. Fiveteen material parameters have to be defined and an unsymmetric material tangent (key word unsymm) has to be stated. As strain hardening law, a more general implementation compared to the paper is applied:

$$R = \sigma_0 + H\varepsilon_{\text{eq}} + R_\infty \left(1 - \exp\left(-a\varepsilon_{\text{eq}}\right)\right). \quad (1)$$

For $R_\infty = 0$ the linear hardening is realized. The properties and depvar definitions are summarized in Tab. 2.

2.3 Flowchart of the UMAT-subroutine

The UMAT-subroutine is divided into several sub-programs. The most important are:

- UMAT (main program): Organizes the elastic-plastic operator split, calculates updates of stress, internal variables, heat source and material tangent entries
- DERIVATIVES_NL/DERIVATIVES: Calculates residual functions and Jacobian-matrix for the non-local and local case
- PLASTICITY_MODUL_NL/PLASTICITY_MODUL: Solves the non-linear equation, returns updated stress and material tangent entries
- STRAIN_HARDENING: Definition of strain hardening rule and damage locus

2.4 Element deletion

The UMAT-subroutine UMAT_ELEMENT_EROSION.f can be used for crack growth simulations. Three ingredients are needed:

1. Input-file: a Depvar number has to be defined, which signals material failure by the key word: delete=12 (see input-file)
2. Input-file: to visualize element deletion the field output variable STATUS has to be requested

Table 2: Properties and Depvar (STATEV in post-processing) of UMAT-subroutine.

PROP No.	Symbol (as in paper and Eq. (1))	Description
1	E	Youngs-modulus
2	ν	Poissons-ratio
3	σ_0	initial yield stress
4	H	hardening modulus
5	N	free hardening exponent
6	R_∞	saturation stress
7	a	hardening exponent
8	C_1	Johnson-Cook-parameter
9	C_2	Johnson-Cook-parameter
10	C_3	Johnson-Cook-parameter
11	C_4	not used, set 0
12	-	not used, set 0
13	ε_c	damage parameter, strain at failure
14	DDS	indicator: stop at $D = D_\mathrm{c}$->set DDS=1
15	DAMAGE_CASE	=1->non-local case; =0->local case
STATEV No.	Symbol (as in paper)	Description
1	ε_eq	equivalent plastic strain
2	ε_eq	hardening variable
3	D	damage variable
4	-	indicator for damage initiation (>0->initiation)
5	$\varepsilon_\mathrm{i}^\mathrm{fix}$	strain at damage initiation
6	-	equivalent deviatoric strain
7	σ_eq	von Mises stress
8	κ	damage driving strain
9	h	stress triaxiality
10	r	source term (heat source)
11	$\frac{\mathrm{d}\,r}{\mathrm{d}\,\triangle\bar{\varepsilon}_\mathrm{d}}$	derivative of source term
12	-	element deletion indicator (=0->deletion)

3. Material properties (PROPS): the property No. 14 has to be set to 0. Setting the value to 1 stops the simulation when material failure is reached at an integration point

2.5 Error and warning messages of UMAT

Error messages are written to the .dat and .msg-files. The simulation is directly aborted if the number of PROPS or STATEV is wrong. The UMAT sends also a kill signal in the case of plane stress state. Problems concerning convergence are also written to the .dat-file.

2.6 Control parameters

For the crack growth simulations, some control settings of Abaqus are redefined. Firstly, the number of cutbacks of time increment is increased (20 allowed). Secondly, the increase in time increment while convergence is increased. Additionally, the convergence control for the displacement and the temperature degree of freedom are changed according to the paper. Please see input-file, keyword *Control.