ARL-TN-1023 ● June 2020

DEVCOM
ARMY RESEARCH
LABORATORY

Documentation of Distributed User Material (UMAT) and Vectorized User Material (VUMAT) Models

by C Gerlach, G Johnson, R Becker, J Lloyd, and A Tonge

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TN-1023 ● JUNE 2020

Documentation of Distributed User Material (UMAT) and Vectorized User Material (VUMAT) Models

C Gerlach, G Johnson
Southwest Research Institute

R Becker, J Lloyd, and A Tonge
Weapons and Materials Research Directorate, CCDC Army Research Laboratory

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*
June 2020

2. REPORT TYPE
Technical Note

3. DATES COVERED (From - To)
September 2017–June 2020

4. TITLE AND SUBTITLE
Documentation of Distributed User Material (UMAT) and Vectorized User Material (VUMAT) Models

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
C Gerlach, G Johnson, R Becker, J Lloyd, and A Tonge

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
CCDC Army Research Laboratory
ATTN: FCDD-RLW-B
Aberdeen Proving Ground, MD 21005

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TN-1023

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
ORCID IDs: Richard Becker, 0000-0003-0104-856X; Andrew Tonge, 0000-0002-7985-3027

14. ABSTRACT

This report presents an approach for documenting user material models in an Abaqus distributed user material (UMAT) or vectorized user material (VUMAT) format for transition to the US Department of Defense and potential use in application codes to support mission programs. This will facilitate rapid adoption of models developed through external collaborations.

15. SUBJECT TERMS
material model, technology transition, computational solid mechanics, collaboration, terminal ballistics

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Andrew Tonge
Unclassified	Unclassified	Unclassified	UU	26	**19b. TELEPHONE NUMBER (Include area code)** (410) 278-1069

Contents

List of Figures iv

List of Tables iv

1. Introduction 1

1.1 File of the Model Description (Called Either <NAME>.pdf or <NAME>.doc) 1

1.1.1 Summary 1

1.1.2 Description of the Model 1

1.1.3 Description of the Input Constants 1

1.1.4 Description of the State Variables 2

1.1.5 Description of the Additional State Variables 2

1.1.6 Discussion of Material Model Robustness, Range of Applicability, and Software Quality-Assurance Activities Associated with the Model 3

1.2 Text File Containing the Complete Abaqus User Material Input (<NAME>.mat) 4

1.3 Verification Problem (<NAME>_example.ppt or <NAME>_example.pdf) 5

1.4 Source Code for VUMAT or UMAT 5

Appendix A. User Instructions for Vectorized User Material (VUMAT) Input Data for Use in Southwest Research Institute (SwRI) Production Simulation Code 6

Appendix B. User Instructions for User Material (UMAT) Input Data for Use in the Host Code 12

List of Symbols, Abbreviations, and Acronyms 18

Distribution List 19

List of Figures

Fig. A-1 Description card for VUMAT input data 7
Fig. B-1 Description card for UMAT input data 13

List of Tables

Table 1 List of constants used in the model 2
Table 2 List of state variables used in the model 2
Table 3 List of additional arrays used in the model 3
Table 4 Description of material model maturity 4

1. Introduction

The purpose of this report is to provide instructions for distributable versions of stand-alone user material (UMAT) and vectorized user material (VUMAT) models to be linked into codes used by the US Department of Defense for materials and ballistic simulations. Instructions for users installing a material model through the VUMAT or UMAT interfaces are included in the code reference manuals. The distinguishing features for these instructions are that these stand-alone material model files and associated documentation will be distributed with future code versions by Southwest Research Institute, but the models will not be incorporated natively into the executables, and the documentation will not be included in the manuals. It is anticipated that most of the material models implemented in this manner will be complex, and considerable subject-matter expertise may be required to allow use of these models by nondevelopers. These models will be distributed with the production code versions within a stand-alone tar file. This stand-alone tar file will include the documentation, the sample input, a verification example, and the source code compatible with the code VUMAT or UMAT options. A maximum of five external models for VUMAT and five external models for UMAT can be used together in a single run. The tar file should contain the following four files (Sections 1.1–1.4).

1.1 File of the Model Description (Called Either <NAME>.pdf or <NAME>.doc)

1.1.1 Summary

This summary should provide the name of the model, the primary use of the model, a description of the units required for use of the model, a designation of VUMAT or UMAT, a published reference for the model (if available), and the name and e-mail address of the developer. Limited background information can also be included if desired. It should not exceed one page in length.

1.1.2 Description of the Model

This section should include the essential formulation of the model. The equations provided should include all of the constants and state variables in the model. References may be included for extended descriptions.

1.1.3 Description of the Input Constants

This section should include a description of the constants used in the model, the associated units, and the value of the constant for the example material. The constants may or may not use the same nomenclature as used in the formulation description, but they should be equated to the constants in the formulation if a different nomenclature is used. For example, σ_{max} may be used in the formulation and SMAX may be used as an alternate synonym in the description of the input constants. Consistent units should be used, (English, International System of Units [SI], or centimeter–gram–second) and the unit of time must be consistent with the database models if the model is to be used along with native material models. All of the constants should be included in a table 1 similar to the following:

Table 1 List of constants used in the model

Constant		Units	Value for	Description
Formulation	Alternate	(SI)	example material	
σ_{max}	SMAX	Pa	1.0 E9	Max stress in Eq. 1
α	ALPHA	…	2.0	…
β	BETA	…	3.0	…
C1	C1	…	4.0	…

1.1.4 Description of the State Variables

A description of the history variables used by the UMAT/VUMAT should be listed in a Table 2, similar to the following. As with the input parameters, the variable names used in the formulation should be included in the table to provide the user with proper context. The alternate names would correspond to those defined in the state variable section of the input file.

Table 2 List of state variables used in the model

State variable		Units	Description
Formulation	Alternate	(SI)	
$\overline{\sigma}$	SBAR	Pa	Average flow strength in Eq. 3
γ_p	EBAR	…	Plastic shear strain in Eq. 6
E_c	EC	J/g	Cold energy in Eq. 4
SV12	D1	…	…
SV13	D2	…	…

1.1.5 Description of the Additional State Variables

The implementations request state variables from VUMAT and UMAT models that are outside of the guaranteed arguments passed from the routine. (Among the guaranteed arguments are current stress, strain, and strain increments.) These additional values aid in setting the time step and shock viscosity. They also map to the standard history variables used to enable element deletion. Similar to the constants, these state variables may have different names depending on whether they are encountered in the formulation or the code itself. These additional state variables should be included in a Table 3 similar to the following:

Table 3 List of additional arrays used in the model

State variable		Units	Description
Formulation	**Alternate**	**(SI)**	
c_s	CS	m/s	Sound velocity (required) in Eq. 1
ε_p	EBAR	…	Equivalent plastic strain in Eq. 2
D	DAM	…	Damage in Eq. 3
Q	Q	Pa	Artificial viscosity in Eq. 4 (sometimes required)

If the state variables need to be initialized to something other than zero, it should either be noted in the variable description or immediately after Table 3.

With the exception of equivalent plastic strain (EPS [EBAR]) and damage (DAM) (which are already internal variables), none of the additional state variables will be written to time-history files. They will always be included in the restart file, and they can be postprocessed via the exodus output, using output flag (PAT) with values from 405 to 408.

1.1.6 Discussion of Material Model Robustness, Range of Applicability, and Software Quality-Assurance Activities Associated with the Model

The goal of this section of the model description is to establish the expectations for the performance of the material model and document the level of testing that has been performed on the model. Every material model has a range of applicability and was likely designed to approach specific problems. By documenting the types of problems that a model *should* work for, the types of problems that a model *might* work for, and the types of problems that the model is *unlikel*y to work for, users are less likely to be frustrated by a model not working for their chosen application.

This section should also include an assessment of the model maturity. Table 4 provides some guidance for the maturity levels of different models. All models are not expected to have a maturity level of 3 to be useful, but it is important to have an assessment of the maturity of each model.

Table 4 Description of material model maturity

Maturity level	Tests passed (including lower tests)	Representative problems
3 – Production 100 M elements × 100 k time steps = 10^{13} evaluations	• Strain increment of 0.1 in a time step • Fully reversed loading in a time step • Recovers from erroneous state variables, temperatures, energies, and strain increments • Equation of state for relative densities from 0.5 to 2.0 and temperatures from –77 to 3,000 C°	• Cavity expansion • Penetration • Deep drawing and extrusion • Cold spray • Hypervelocity impact • Shape charge formation
2 – Robust	• Can do a strain increment of 0.02 in a time step for strain rates ranging from 10^{-3} to 10^5/s • Handles volumetric compression, shear, and uniaxial strain equally along different orientations • Box loading from fully loaded to unloaded • Known range of validity	• Rolling operations • Complex specimen geometries loaded to failure • Spall • Taylor cylinder impact • Low-speed impact without penetration
1 – Basic	• Converges over a moderate range of loading rates • Will not crash if object remains unloaded for several time steps • No hardwired material parameters • Satisfies frame objectivity	• Simple loading of multiple elements • Single-element problems
0 – Single use	• Physically reasonable and mathematically correct	…

Next, a discussion of software testing and review activities should be provided. This is particularly important as models are updated and revised. As a model is revised, will answers change? Are checks made to document when a change to the model will (or may) change the results? Is the model being actively developed and can new users expect any type of support from the model developer? How should problems/issues be reported?

Finally, is there a path envisioned for model improvements? Is the developer accepting additions and modifications or will the user improvements have to be kept locally as the user's branch?

1.2 Text File Containing the Complete Abaqus User Material Input (<NAME>.mat)

This file should contain the same Abaqus–UMAT input used in the verification problem, and it should be formatted for Abaqus input. The user can then copy and paste this material definition into their own input file for use and modification. This input includes the user material parameters, the density, the number of parameters, the number of history variables, and the names of the history variables.

1.3 Verification Problem (<NAME>_example.ppt or <NAME>_example.pdf)

The definition and results (deformed geometry and/or state-variable contours) of a simple problem should be provided for verification that the model has been installed properly. It should be a single-material problem with simple boundary conditions and simple elements such that it can be run in a straightforward manner. If possible, the full input file to run the verification problem should be included (<NAME>.dat). Otherwise, the code development team will run the verification problem and include the associated input file with the code distribution. Additional assistance regarding the model must be provided by the developer.

1.4 Source Code for VUMAT or UMAT

The source code (Fortran or C) should be included. It must be slightly modified from that used when run within Abaqus. Details of the modifications are provided in the user instructions provided as Appendixes A and B for VUMAT and UMAT, respectively.

Note: Subroutines within the VUMAT and UMAT code must not have the same name as any subroutines in the host code. If multiple VUMAT/UMAT routines are used in a single run, their subroutine names must also be distinct from one another. Currently, no host-code subroutine names start with the letter “Y”.

Appendix A. User Instructions for Vectorized User Material (VUMAT) Input Data for Use in Southwest Research Institute (SwRI) Production Simulation Code

A.1 Description Card for VUMAT Input Data (3I5, 5X, I5, F5.0, 4I5, 3F10.0)

This card (Fig. A-1) signals the start of a section of Abaqus–VUMAT material input. This material-model option is provided to allow the use of complex material models developed by others. This Abaqus–VUMAT interface is commonly used by universities and US Department of Energy and Department of Defense laboratories. It accepts material-model subroutines written in the Abaqus–VUMAT format with some minor additions. A detailed description and verification examples are provided in SwRI Technical Report No. 18.22494/0005.

DESCRIPTION CARD FOR VUMAT INPUT DATA (3I5, 5X, I5, F5.0, 4I5, 3F10.0)

MATL	MTYPE	IQ	▨	ORTHO	EFAIL	CS#	EPS#	DAM#	Q#	CL	CQ	CH

MATERIAL ORIENTATION CARD FOR ORTHO = 1 OR 2 (3F10.0)

θ1	θ2	θ3	▨

VUMAT INPUT DATA – AS REQUIRED

Fig. A-1 Description card for VUMAT input data

The first addition for the VUMAT subroutine is to compute and pass the longitudinal sound velocity (CS) to EPIC in one of the VUMAT dependent-variables (for every element on every cycle). This is required as the host code uses the CS to compute an accurate and robust integration time increment. The CS is also used for the artificial viscosity when it is computed by the host code.

The second (optional) addition is to specify one of the VUMAT dependent-variables as an equivalent (plastic) strain (EPS#). When used, this VUMAT EPS is copied into the EBAR array in EPIC, where it is used to totally fail, erode, or convert elements. It is also available for postprocessing. If EPS# = 0, the host code will set EBAR = 0 (in the host code), and it will not be possible to erode or convert elements or to postprocess an equivalent strain.

The third (optional) addition is to specify one of the VUMAT dependent-variables as the damage (DAM#), which is a common variable for material models. When used, this VUMAT damage will be copied into the DAM array in the host code and available for postprocessing damage. If DAM# = 0 it will not be possible to postprocess damage.

The fourth addition (which is sometimes required) is to specify one of the VUMAT dependent-variables as artificial viscosity (Q#). This is only required when the Q is computed within the VUMAT subroutine, and it is required because it is used to determine the integration time increment in the host code. If there is no Q or if the host code is selected to compute the Q, it is not necessary to specify this array and Q# = 0.

Several minor details. The argument CMNAME will be treated as a double-precision variable instead of a character string and will have 0.0 passed into it. There is currently no support for external field variables; NFIELDV will be zero, and FIELDOLD and FIELDNEW will be declared with dimension (NBELE,1) and set to zero; NDIR and NSHR will be set to 3, regardless of the dimension of the problem; LANNEAL, COORDMP, TEMPOLD, and TEMPNEW will always be set to zero. Instructions with details about compilation can be found in section 4.7.7 of the host-code manual.

State variable initialization is expected to be done inside the VUMAT routine, triggered by the fictitious time step at TIME = 0.0 that Abaqus uses to determine the initial time increment. (The host code mimics that fictitious time step.)

A characteristic of the Abaqus–VUMAT interface is that the stresses are computed in the local (material) coordinate system. This requires the stress and strain rate components to be rotated into the local coordinate system before every stress update and then rotated back into the system (global) coordinates such that the nodal forces can be computed from the stresses. If ORTHO > 0, the host code uses the Flanagan–Taylor algorithm to perform the rotations. For isotropic materials, when ORTHO = 0, the host code uses the Jaumann stress rate to update the stresses in the system coordinates (due to incremental rotations that occur during an integration cycle), and this approach is provided as an option. It does not require the transformations to and from the local coordinate system and is therefore more efficient. For anisotropic materials, the Jaumann stress rate cannot be used for finite rotations, so ORTHO should be greater than zero and the initial material orientation specified.

MATL = Material number (1–250) specified by user. Will supersede library material data with same material number.

MTYPE = Code to specify a VUMAT material. Must be in the range of MTYPE = 20–24. It is possible to include five different VUMAT material models in a single run. For MTYPE = 20, the VUMAT subroutine must be named VUMAT20; for MTYPE = 21, the VUMAT subroutine must be named VUMAT21; and so on.

Note: Subroutines within the VUMAT code must not have the same name as any of the host-code subroutines. If multiple VUMAT/UMAT routines are used in a single host-code run, their subroutine names must also be distinct from one another. Currently, no host-code subroutine names start with the letter ‘Y’.

IQ = 0 indicates artificial viscosity is not used within the VUMAT subroutine and is not requested to be used by the host code.

= 1 indicates that Q is computed within the VUMAT subroutine and is not computed by the host code. This option requires Q# to be specified.

= 2 will require the host code to compute Q. Only compressive viscosity will be computed and used.

= 3 will require the host code to compute Q. Both compressive and (linear only) tensile viscosities will be computed and used.

ORTHO = 0 will not rotate the material axes from the system axes and will not rotate the material axes due to rotations that occur during the computation. Uses the Jaumann stress rate to update the stresses.

= 1 will rotate the material axes through a 3-D rigid body rotation using the Euler angles in Fig. 4.50 in the host-code manual, and will include additional rotations that occur during the computation.

= 2 will rotate the material axes to an axisymmetric configuration about a line through the origin described by direction cosines. Here the z-axis is the centerline axis along the line out of the origin, the x-axis is in the radial direction, and the y-axis is in the hoop direction. Also will include additional rotations that occur during the computation.

EFAIL = Value of equivalent strain (carried in EPS#), or volumetric strain, which, if exceeded, will totally fail the element such that it produces no stresses or pressures. If EPS# = 0, the check will only be made for the volumetric strain. If EFAIL ≥ 999, the check for total failure will be omitted.

CS# = VUMAT dependent variable number that will contain the longitudinal sound speed. This is an additional variable required from VUMAT to compute the integration time increment in the host code.

EPS# = VUMAT dependent variable number that will contain an equivalent strain computed by VUMAT. This is usually an EPS (EBAR), but any strain measure can be carried in this array. When used, this VUMAT equivalent strain is copied into the EBAR array in the host code, where it is used to totally fail, erode, or convert elements. It is also available for postprocessing. If EPS# = 0, the host code will set EBAR = 0 (in the host code), and it will not be possible to erode or convert elements, or to postprocess an equivalent strain.

DAM# = VUMAT dependent-variable number that will contain material damage (for failure) computed by VUMAT. When used, this VUMAT damage will be copied into the damage (DAM) array in the host code and will be available for postprocessing damage. If DAM# = 0, it will not be possible to postprocess damage.

Q# = VUMAT dependent-variable number that will contain Q, required only if Q is computed within VUMAT. It is required from VUMAT because it is used to determine the integration time increment in the host code. If there is no Q or if the host code is selected to compute the Q, set Q# = 0.

CL = Linear artificial viscosity coefficient (CL ≈ 0.2). Used only when IQ = 2 or 3.

CQ = Quadratic Q coefficient (CQ ≈ 4.0). Used when IQ = 2 or 3. Also required when artificial viscosity is computed in VUMAT (IQ = 1).

CH = Hourglass Q coefficient for 2-D quad elements or 3-D brick elements (CH ≈ 0.3).

A.2 Material Orientation Card for ORTHO = 1 or 2 (3F10.0)

This card is required for ORTHO = 1 or 2. It defines the orientation of the material axes relative to the system coordinates.

θ1 = First Euler angle rotation about the initial z-axis, z_0 (degrees), for 3-D geometry and ORTHO = 1. Leave blank (θ1 = 0) for ORTHO = 0. Use θ1 as x direction cosine for z centerline axis out of origin for ORTHO = 2.

θ2 = Second Euler angle rotation about the previously rotated x-axis, x_1 (degrees), for 3-D geometry and ORTHO = 1. Leave blank (θ2 = 0) for ORTHO = 0. Use θ2 as y direction cosine for z centerline axis out of origin for ORTHO = 2.

θ3 = Third Euler angle rotation about the previously rotated z-axis, z_2 (degrees), for 3-D geometry and ORTHO = 1. Leave blank (θ3 = 0) for ORTHO = 0. Use θ3 as z direction cosine for z centerline axis out of origin for ORTHO = 2.

Subsequent to the one or two cards specifying a VUMAT material, the user is expected to paste in the Abaqus–UMAT definition for this material, exactly as it would appear in the Abaqus input file. The format is sketched out in the following:

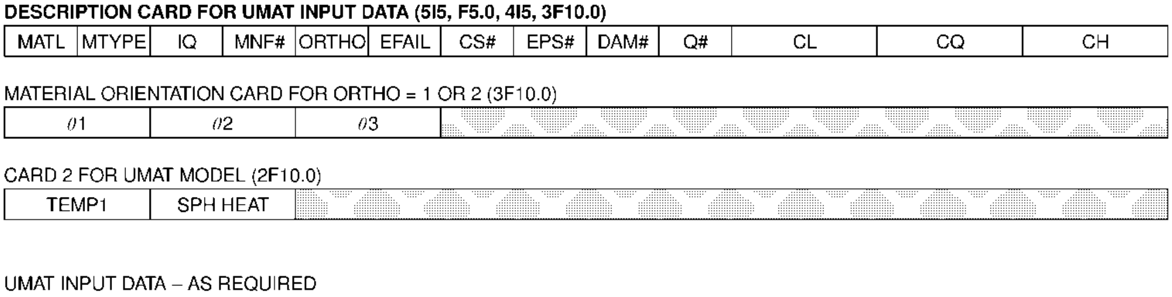
```
** Two asterisks indicates a comment, one indicates a keyword
* MATERIAL, NAME=<material_name>
* DENSITY
<real #>
* USER MATERIAL, CONSTANTS= <# of constants>
<real #>, <real #>, .... (8 reals per line until complete)
* DEPVAR, DELETE=<index of element deletion variable>
<# of state variables>
<index>, <name>,  "<description>"
(for as many lines as the # of state variables)
```

Note: If the DELETE state variable is defined and used, it will act in the same way as the EFAIL flag in the host code: it will delete elements without updating slide lines and should therefore not be used in conjunction with an eroding or converting slideline.

Appendix B. User Instructions for User Material (UMAT) Input Data for Use in the Host Code

B.1 Description Card for UMAT Input Data (5I5, F5.0, 4I5, 3F10.0)

This card (Fig. B-1) signals the start of a section of Abaqus–UMAT material input. This material-model option is provided to allow the host code to use complex material models developed by others. This Abaqus–UMAT interface is commonly used by universities and US Department of Energy and Department of Defense laboratories. A detailed description and verification examples are provided in Southwest Research Institute Technical Report No. 18.22494/005.



DESCRIPTION CARD FOR UMAT INPUT DATA (5I5, F5.0, 4I5, 3F10.0)

MATL	MTYPE	IQ	MNF#	ORTHO	EFAIL	CS#	EPS#	DAM#	Q#	CL	CQ	CH

MATERIAL ORIENTATION CARD FOR ORTHO = 1 OR 2 (3F10.0)

θ1	θ2	θ3	▨

CARD 2 FOR UMAT MODEL (2F10.0)

TEMP1	SPH HEAT	▨

UMAT INPUT DATA – AS REQUIRED

Fig. B-1 Description card for UMAT input data

The primary difference between the Abaqus vectorized user material (VUMAT) and UMAT models is that UMAT models are designed to be used with implicit solvers, and the UMAT routines are called for each integration point individually, whereas the VUMAT models are designed to be explicitly integrated and are called for blocks of integration points. The host-code UMAT interface uses explicit integration in exactly the same way as it does for the VUMAT interface. The only differences between the host-code VUMAT and UMAT implementations are the arguments passed in and out of the routines.

The host code accepts material-model subroutines written in the Abaqus–UMAT format with some minor additions.

The first addition for the UMAT subroutine is to compute and pass the longitudinal sound velocity (CS) to the host code in one of the UMAT dependent-variables (for every element on every cycle). This is required because the host code uses the CS to compute an accurate and robust integration time increment. The CS is also used for the artificial viscosity (Q) when it is computed by the host code.

The second (optional) addition is to specify one of the UMAT dependent-variables as an equivalent plastic strain (EPS#). When used, this UMAT EPS is copied into the EPS (EBAR) array in the host code, where it is used to totally fail, erode, or convert elements. It is also available for postprocessing. If EPS# = 0, the host code will set EBAR = 0 (in the host code), and it will not be possible to erode or convert elements, or to postprocess an equivalent strain.

The third (optional) addition is to specify one of the UMAT dependent variables as the damage (DAM#), which is a common variable for material models. When used, this UMAT damage will be copied into the DAM array in the host code and available for postprocessing damage. If DAM# = 0, it will not be possible to postprocess damage.

The fourth addition (which is sometimes required) is to specify one of the UMAT dependent variables as Q (Q#). This is only required when the Q is computed within the UMAT subroutine, and it is required because it is used to determine the integration time increment in the host code. If there is no Q, or if the host code is selected to compute the artificial viscosity, then it is not necessary to specify this array and Q# = 0.

Several minor details: The argument CMNAME will be treated as a double-precision variable instead of a character string, and will have 0.0 passed into it; NDI and NSHR will be set to 3, and NTENS will be set to 6, regardless of the dimension of the problem; NOEL and NPT will be set to the element/particle number; RPL, DDSDDT, DRPLDE, DRPLDT, PREDEF, DPRED, COORDS, PNEWDT, LAYER, KSPT, JSTEP, and KINC are all set to 0, and any values given to them inside of UMAT will be ignored. Instructions with details about compilation can be found in section 4.7.8 of the host-code manual.

State variable initialization and initial material sound-speed computations are done inside the host-code routine PRE_UMAT_INIT, which must be appropriately modified by the user. The initial sound speed of the material must be defined and is necessary to compute the length of the first time step. The user may choose to initialize the state variables used by the UMAT routines, but PRE_UMAT_INIT will by default set them all to zero.

A characteristic of the Abaqus–UMAT interface is that the stresses are computed in the local (material) coordinate system. This requires the stress and strain rate components to be rotated into the local coordinate system before every stress update and then rotated back into the system (global) coordinates such that the nodal forces can be computed from the stresses. If ORTHO > 0, the host code uses the Flanagan–Taylor algorithm to perform the rotations. For isotropic materials, when ORTHO = 0, the host code uses the Jaumann stress rate to update the stresses in the system coordinates (due to incremental rotations that occur during an integration cycle), and this approach is provided as an option. It does not require the transformations to and from the local coordinate system and is therefore more efficient. For anisotropic materials, the Jaumann stress rate cannot be used for finite rotations, so ORTHO should be greater than zero and the initial material orientation specified.

MATL = Material number (1–250) specified by user. Will supersede library material data with same material number.

MTYPE = Code to specify a UMAT material. Must be in the range of MTYPE = 25–29. It is possible to include five different UMAT material models in a single run. For MTYPE = 25, the UMAT subroutine must be named UMAT25; for MTYPE = 26, the UMAT subroutine must be named UMAT26; and so on.

Note: Subroutines within the UMAT code must not have the same name as any host-code subroutines. If multiple VUMAT/UMAT routines are used in a single host-code run, their subroutine names must also be distinct from one another. Currently, no host-code subroutine names start with the letter ‘Y’.

25 = Code to specify a UMAT material.

IQ = 0 indicates Q is not used within the UMAT subroutine and is not requested to be used by the host code.

= 1 indicates that Q is computed within the UMAT subroutine and is not computed by the host code. This option requires Q# to be specified.

= 2 will require the host code to compute Q. Only compressive viscosity will be computed and used.

= 3 will require the host code to compute Q. Both compressive and (linear only) tensile viscosities will be computed and used.

MNF# = The UMAT dependent variable number that will have an integer flag passed into it based on whether the element shares a node with a failed element. The variable will be set to 1.0 for failed material, 2.0 for adjacent to failed material or on the surface of the problem, and 3.0 for completely interior material. Particles will always be set to 1.0.

ORTHO = 0 will not rotate the material axes from the system axes and will not rotate the material axes due to rotations that occur during the computation. Uses the Jaumann stress rate to update the stresses.

= 1 will rotate the material axes through a 3-D rigid body rotation using the Euler angles in Fig. 4.50 in the host-code manual. Also will include additional rotations that occur during the computation.

= 2 will rotate the material axes to an axisymmetric configuration about a line through the origin described by direction cosines. Here, the z-axis is the centerline axis along the line out of the origin; the x-axis is in the radial direction; and the y-axis is in the hoop direction. Also will include additional rotations that occur during the computation.

EFAIL = Value of equivalent strain (carried in EPS#), or volumetric strain, which, if exceeded, will totally fail the element such that it produces no stresses or pressures. If EPS# = 0, the check will only be made for the volumetric strain. If EFAIL ≥ 999, the check for total failure will be omitted.

CS# = UMAT dependent variable number that will contain the longitudinal CS. This is an additional variable that is required from UMAT to compute the integration time increment in the host code.

EPS# = UMAT dependent variable number that will contain an equivalent strain computed by UMAT. This is usually an EPS, but any strain measure can be carried in this array. When used, this UMAT equivalent strain is copied into the EBAR array in the host code, where it is used to totally fail, erode, or convert elements. It is also available for postprocessing. If EPS# = 0, the host code will set EBAR = 0 (in the host code), and it will not be possible to erode or convert elements or to postprocess an equivalent strain.

DAM# = UMAT dependent-variable number that will contain material damage (for failure) computed by UMAT. When used, this UMAT damage will be copied into the DAM array in the host code and will be available for postprocessing damage. If DAM# = 0, it will not be possible to postprocess damage.

Q# = UMAT dependent variable number that will contain Q, required only if Q is computed within UMAT. It is required from UMAT because it is used to determine the integration time increment in the host code. If there is no Q or if the host code is selected to compute the Q, set Q# = 0.

CL = Linear Q coefficient (CL ≈ 0.2). Used only when IQ = 2 or 3.

CQ = Quadratic Q coefficient (CQ ≈ 4.0). Used when IQ = 2 or 3. Also required when Q is computed in UMAT (IQ = 1).

CH = Hourglass Q coefficient for 2-D quad elements or 3-D brick elements (CH ≈ 0.3).

B.2 Material Orientation Card for ORTHO = 1 or 2 (3F10.0)

This card is required for ORTHO = 1 or 2. It defines the orientation of the material axes relative to the system coordinates.

θ1 = First Euler angle rotation about the initial z-axis, z_0 (degrees), for 3-D geometry and ORTHO = 1. Leave blank (θ1 = 0) for ORTHO = 0. Use θ1 as x direction cosine for z centerline axis out of origin for ORTHO = 2.

θ2 = Second Euler angle rotation about the previously rotated x-axis, x_1 (degrees), for 3-D geometry and ORTHO = 1. Leave blank (θ2 = 0) for ORTHO = 0. Use θ2 as y direction cosine for z centerline axis out of origin for ORTHO = 2.

θ3 = Third Euler angle rotation about the previously rotated z-axis, z_2 (degrees), for 3-D geometry and ORTHO = 1. Leave blank (θ3 = 0) for ORTHO = 0. Use θ3 as z direction cosine for z centerline axis out of origin for ORTHO = 2.

B.3 Card 2 for UMAT Model (2F10.0)

TEMP1 = Initial temperature of the material (degrees).

SPH HEAT = Specific heat (work/mass/degrees). If set to 0.0, the UMAT routine will always have TEMP1 passed in as the temperature argument. Otherwise, TEMP = TEMP1 + ES/(SPH HEAT*DENSITY), where ES is the internal energy of the element.

Subsequent to Card 2 for UMAT Model, the user is expected to paste in the Abaqus–UMAT definition for this material, exactly as it would appear in the Abaqus input file. The format is sketched out in the following:

```
** Two asterisks indicates a comment, one indicates a keyword
* MATERIAL, NAME=<material_name>
* DENSITY
<real #>
* USER MATERIAL, CONSTANTS= <# of constants>
<real #>, <real #>, .... (8 reals per line until complete)
* DEPVAR, DELETE=<index of element deletion variable>
<# of state variables>
<index>, <name>,  "<description>"
(for as many lines as the # of state variables)
```

Note: If the DELETE state variable is defined and used, it will act in the same way as the EFAIL flag in the host code: it will delete elements without updating slide lines and should therefore not be used in conjunction with an eroding or converting slideline.

List of Symbols, Abbreviations, and Acronyms

2-D	two-dimensional
3-D	three-dimensional
CS	sound velocity
DAM	damage
EPS	equivalent plastic strain
Q	artificial viscosity
SI	International System of Units
SwRI	Southwest Research Institute
UMAT	user material
VUMAT	vectorized user material

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

1 CCDC ARL
(PDF) FCDD RLD CL
TECH LIB

2 LOS ALAMOS NATL LAB
(PDF) B CLEMENTS
K BENNETT

2 SANDIA NATL LAB
(PDF) C HETTLER
J SERRANO

4 LAWRENCE LIVERMORE
(PDF) NATL LAB
N BARTON
D FAUX
E HERBOLD
J MCCARRICK

4 JOHNS HOPKINS UNIV
(PDF) L BRADY
T HUFNAGEL
V NAKANO
K T RAMESH

1 RUTGERS UNIV
(PDF) R HABER

1 UNIV OF DELAWARE
(PDF) J GILLESPIE

1 AFRL
(PDF) K J VANDEN

1 OSD
(PDF) K SPANGLER

4 NSWC IHEODTD
(PDF) A D GOLDBERG
S K DWIVEDI
B M BLAZEK
D ABELN

4 SOUTHWEST RSRCH INST
(PDF) T HOLMQUIST
G JOHNSON
C GERLACH
S BEISSEL

3 PEO SOLDEIR
(PDF) C BAKER
E BEAUDOIN
D OTTERSON

2 CCDC AC
(PDF) S S RECCHIA
D G PFAU

4 CCDC SC
(PDF) R DILALLA
JKIREJCZYK
C LEWIS
M ROTH

45 CCDC ARL
(PDF) FCDD RLC EM
J KNAP
FCDD RLR
D STEPP
FCDD RLV M
M HAILE
FCDD RLW
S SCHOENFELD
A RAWLETT
FCDD RLW B
R BECKER
J CAMPBELL
P GILLICH
C HOPPEL
B SCHUSTER
A TONGE
L VARGAS-GONZALEZ
FCDD RLW L
T SHEPPARD

FCDD RLW LB
G SUTHERLAND
FCDD RLW LH
J NEWILL
FCDD RLW M
E CHIN
FCDD RLW MA
E WETZEL
T BOGETTI
FCDD RLW MB
G GAZONAS
D HOPKINS
B LOVE
D OBRIEN
B POWERS
FCDD RLW ME
S COLEMAN
A DIGIOVANNI
J LASALVIA
J SWAB
FCDD RLW P
R FRANCART
FCDD RLW PA
S BILYK
FCDD RLW PB
J MCDONALD
S SATAPATHY
M SCHEIDLER
T WEERASOORIYA
FCDD RLW PC
D CASEM
J CAZAMIAS
J CLAYTON
M FERMEN-COKER
R LEAVY
J LLOYD
S SEGLETES
C WILLIAMS
FCDD RLW PD
R DONEY
C RANDOW
FCDD RLW PE
M LOVE
P SWOBODA