ARL-TN-1177 ● Oct 2023

User Guide to a Wrapper That Allows a Custom User-Defined Material Model (UMAT) Originally Developed for Abaqus/EPIC to be Used in LS-DYNA Simulations

by Stephen L Alexander, Richard Becker, and Tusit Weerasooriya

NOTICES

Disclaimers

ARL-TN-1177 ● Oct 2023

User Guide to a Wrapper That Allows a Custom User-Defined Material Model (UMAT) Originally Developed for Abaqus/EPIC to be Used in LS-DYNA Simulations

Stephen L Alexander
SURVICE Engineering Company

Richard Becker and Tusit Weerasooriya
DEVCOM Army Research Laboratory

REPORT DOCUMENTATION PAGE

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED START DATE	3. DATES COVERED END DATE
October 2023	Technical Note	7/01/2022	9/29/2023

4. TITLE AND SUBTITLE
User Guide to a Wrapper That Allows a Custom User-Defined Material Model (UMAT) Originally Developed for Abaqus/EPIC to be Used in LS-DYNA Simulations

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER	5e. TASK NUMBER	5f. WORK UNIT NUMBER

6. AUTHOR(S)
Stephen L Alexander, Richard Becker, and Tusit Weerasooriya

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
DEVCOM Army Research Laboratory ATTN: FCDD-LA-TB Aberdeen Proving Ground, MD 21005	ARL-TN-1177

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
DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES
ORCID IDs: Tusit Weerasooriya, 0000-0003-3299-2166; Richard Becker, 0000-0003-0104-856X

14. ABSTRACT
Commercial codes such as LS-DYNA and Abaqus have many different types of built-in material models to simulate different types of materials. They also provide an option for the user to implement custom user-defined material models (user-mats). This is a powerful option for researchers simulating advanced materials for which there are not yet any suitable established models. However, the different commercial codes have substantially different interfaces for implementing user-mats. These differences preclude the user-mats from being able to be interchanged between codes as-is. Here, a user guide is provided for a wrapper we developed in order to use, in LS-DYNA simulations, a user-mat that was originally developed for Abaqus. The present wrapper had been developed as part of our recently published computational framework for simulating ultra-high-molecular-weight polyethylene film-based composites. One of the material models we used was originally implemented as an Abaqus UMAT. We developed the present wrapper in order to use the Abaqus UMAT in our LS-DYNA simulations. This user guide includes stepwise directions for using the wrapper, clearly indicating the specific cases for which it can be used in its present form and what additions would be needed for more generalized cases.

15. SUBJECT TERMS
user-defined material model, UMAT, VUMAT, LS-DYNA, Abaqus, EPIC, FORTRAN, UHMWPE, software wrapper, Sciences of Extreme Materials, Terminal Effects

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	C. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UU	38

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Tusit Weerasooriya	(410) 652-9450

STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

1. Introduction

The commonly used finite element (FE) programs LS-DYNA, Abaqus, and EPIC allow researchers to implement their own custom material models. The model is implemented as a FORTRAN subroutine that interfaces with the FE solver. In a simplified description, the FE solver supplies the user subroutine with information at the current time increment. The user subroutine must then calculate essential quantities for the time step and pass these back to the FE solver. The specific quantities that are calculated vary between the different FE programs but include, at a minimum, the stress after the current time increment and the updates to any relevant state variables.

The details of the interface between the FE solver and the user subroutine differ between LS-DYNA and Abaqus, whereas Abaqus and EPIC follow similar conventions. Therefore, user subroutines that had originally been developed to be used in Abaqus/EPIC cannot be directly used as-is in LS-DYNA. Additional complications arise from there being two different types of user subroutines in Abaqus, the UMAT and VUMAT (vectorized UMAT). These two types also have different interfaces and formulations. In summary, there are (1) the custom material model in LS-DYNA, referred to here as the "LS-DYNA umat," and (2) the custom material models in Abaqus/EPIC, referred to as the Abaqus "UMAT" and "VUMAT."

We previously published a technical report describing a computational framework developed for FE simulation of ultra-high-molecular-weight polyethylene (UHMWPE) films based on crystal-plasticity deformation and subsequent shear delamination failure mechanisms (Alexander et al. 2023). That report compared FE simulations between a sub-mesoscale (SMS) model and an elemental macroscale (EMS) model. All simulations were run in LS-DYNA. However, the EMS model had originally been developed by Becker (2019) as an Abaqus/EPIC UMAT. Therefore, we developed a code, referred to as a "wrapper," to be able to run the EMS UMAT model in LS-DYNA.

The wrapper has utility to the broader research community by providing a framework for using custom material models in LS-DYNA that had originally been developed to be run as a UMAT in Abaqus/EPIC. The wrapper was initially developed only to be used for the EMS UMAT model. To run the EMS model (which was written for Abaqus) in LS-DYNA, the wrapper supplies the UMAT with the deformation gradient from the end of the previous time step and the deformation gradient at the end of the current time step. The latter was supplied by setting the IHYPER flag in LS-DYNA to 1, and the former was supplied by saving the deformation gradient at each time step into the state variables. In Abaqus, input and output to the UMATs are in the current configuration. Accordingly, the EMS UMAT was also developed for input variables in the current configuration. Using the EMS UMAT in LS-DYNA required the IORTHO flag to be set to 0 in order for LS-DYNA to also provide input variables in the current configuration.

The wrapper could be used for other material models written for Abaqus UMATs, since UMATs in Abaqus are in general required to input and output in the current configuration. Users wishing to apply the wrapper for a different UMAT other than the EMS model may need to explicitly assign additional parameters based on what is used within their specific Abaqus UMAT. Here, we provide details for using the wrapper to run UMATs in LS-DYNA in the current configuration. The code for the wrapper is provided in Appendix A.

2. Overview

The wrapper is a FORTRAN subroutine called `UMAT42V` and acts as a LS-DYNA umat (overall concept is summarized in Fig. 1). Appendix A provides the wrapper code. For each time step, LS-DYNA supplies variables to the wrapper at the current time step, including the stress and strain increments, state variables, and material properties. The first part of the wrapper converts these variables from the LS-DYNA convention to the Abaqus convention. The wrapper then calls the Abaqus UMAT, referred to here as `UMAT_ABAQUS`. The Abaqus UMAT calculates the stress and updates any history variables as it would when running normally in Abaqus/EPIC. The wrapper then converts the stress and history variables back to the LS-DYNA convention.

The wrapper has several write statements that write to terminal output. For example, in the first cycle, the wrapper writes out all the material properties that were specified in the LS-DYNA input file, enabling the user to confirm the setup.

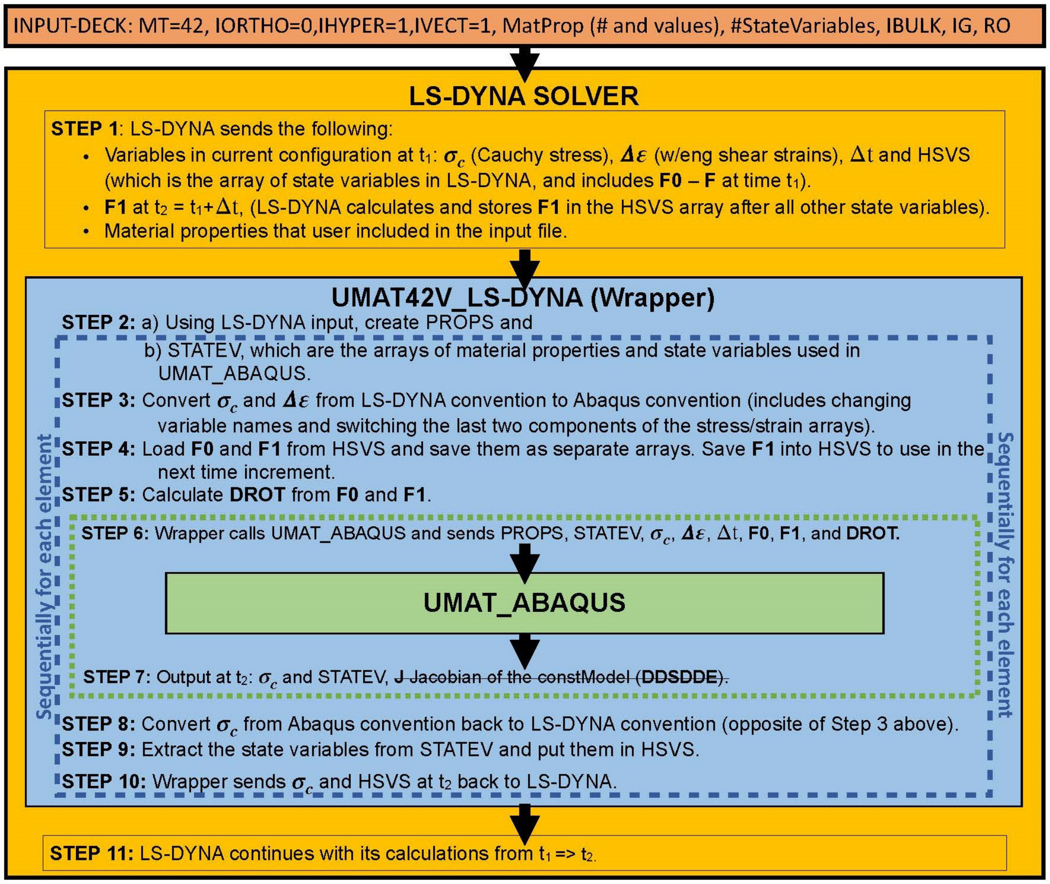

Fig. 1 Summary of wrapper concept. Only the key inputs and outputs at the various levels are included here to help with visualization.

3. Stepwise Directions to Use the Wrapper

3.1 Ensure Abaqus UMAT Meets Specifications of the Wrapper

Figure 2 shows the form that the wrapper expects for the header of `UMAT_ABAQUS`.

```
      SUBROUTINE UMAT_ABAQUS(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
     1 RPL,DDSDDT,DRPLDE,DRPLDT,
     1 STRAN,DSTRAN,TIME,DTIME,TEMP,DTEMP,PREDEF,DPRED,CMNAME,
     1 NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,DROT,PNEWDT,
     1 CELENT,F0,F,NOEL,NPT,LAYER,KSPT,KSTEP,KINC)
C
      IMPLICIT NONE
C     Argument variables
      REAL*8 CMNAME
      INTEGER KINC, KSPT, KSTEP, LAYER, NDI, NOEL, NPROPS, NPT, NSHR
      INTEGER NSTATV, NTENS
      REAL*8 CELENT, COORDS(3), DDSDDE(NTENS,NTENS), DDSDDT(NTENS)
      REAL*8 F0(3,3), F(3,3), DPRED(1), DROT(3,3)
      REAL*8 DRPLDE(NTENS), DRPLDT, DSTRAN(NTENS), DTEMP, DTIME, PNEWDT
      REAL*8 PREDEF(1), PROPS(NPROPS), RPL, SCD, SPD, SSE
      REAL*8 STATEV(NSTATV), STRAN(NTENS), STRESS(NTENS), TEMP, TIME(2)
```

Fig. 2 Form of the header for the Abaqus UMAT that the wrapper expects

Figure 2 shows that the wrapper expects the `UMAT_ABAQUS` header to include 37 total variables. Normally in Abaqus, all 37 variables could be used in the UMAT for calculations, as could be seen in standard Abaqus documentation. However, the present wrapper only assigns values to the specific variables that were necessary for using the EMS UMAT. The wrapper assigned values to 13 variables: `STRESS`, `STATEV`, `DSTRAN`, `DTIME`, `NDI`, `NSHR`, `NTENS`, `NSTATV`, `PROPS`, `NPROPS`, `DROT`, `F0`, and `F`. The remaining 24 variables are declared in `UMAT42V` but never explicitly assigned values before being sent to the `UMAT_ABAQUS`. Therefore, an Abaqus UMAT, using this wrapper as-is, should not use these variables for its calculations. Appendix B lists all 37 of the `UMAT_ABAQUS` argument variables with additional information regarding the 13 that were explicitly assigned values by the wrapper.

3.2 Compile LS-DYNA to Use the Wrapper

General information on compiling LS-DYNA in order to use a LS-DYNA umat has been previously documented by Hampton et al. (2022). Users should be aware that the present wrapper was developed with a LS-DYNA distribution that used double precision variables by default. Appendix C provides an example of how this can be confirmed.

In LS-DYNA, vectorized user material models are included in the source file "`dyn21umatv.f`". LS-DYNA organizes this source file as a series of subroutines. Each subroutine has a title in the format: `UMATxxV`, where `xx` is a number between 41 and 50. In the original source file from LS-DYNA, each subroutine is simply a placeholder or example. The present wrapper was written to take the place of the subroutine titled `UMAT42V`.

The following steps should be executed prior to compiling LS-DYNA in order to use the Abaqus UMAT by means of the wrapper:

1) Completely remove the default `UMAT42V` subroutine that LS-DYNA provides as a placeholder in the `dyn21umatv.f` source file.

2) The present wrapper with accompanying utility subroutines was saved in a file called "`wrapper.for`", which is provided in Appendix A. The user needs to add the `wrapper.for` file to the same directory as the LS-DYNA `dyn21umatv.f` source file.

3) Add the following statement to the `dyn21umatv.f` source file:

   ```
   include 'wrapper.for'
   ```

This statement must be added at a location that is outside the scope of any of the subroutines in the source file. For example, it could be added to the very top of the file. Or it could be added between the end of the `UMAT41V` subroutine and the start of the `UMAT43V` subroutine, taking the place of the `UMAT42V` subroutine that was deleted in Step 1.

4) The wrapper assumes that the Abaqus UMAT is a subroutine titled `UMAT_ABAQUS` (see Fig. 2). Furthermore, the wrapper assumes that the `UMAT_ABAQUS` subroutine, together with any additional custom subroutines that the Abaqus UMAT calls, is saved in a file called "`umat_abaqus.for`". The user needs to add the `umat_abaqus.for` file to the same directory as the LS-DYNA `dyn21umatv.f` source file.

3.3 Set Up LS-DYNA Input (Keyword) File

The LS-DYNA simulation is set up through the input file, also referred to as the keyword file (k-file). After compilation as described in the previous section, elements can be assigned the `UMAT_ABAQUS` material model through the `*MAT_USER_DEFINED_MATERIAL_MODELS` keyword, abbreviated here as `*MAT_USER`. This section describes how to specify this keyword.

The keyword consists of at least three data cards (lines). The first two data cards contain different variables related to the settings for the material model. Then, the user lists the material properties in the subsequent data cards, using as many as needed to list all of the material properties. It is easier to set up the *MAT_USER keyword by first listing the material properties before specifying the settings in the first two data cards. Therefore, this user guide starts by describing the material property data cards before describing the first two data cards.

3.3.1 Specify Material Properties Starting from the Third Data Card

The user lists material properties to be read by LS-DYNA starting on the third data card. The material properties listed in these data cards include the following:

- First and second properties: estimates for the bulk and shear moduli for the custom user material model. LS-DYNA requires that estimates for the bulk, K, and shear, G, moduli of the custom user material be specified as material properties in the `*MAT_USER` keyword: "All user-defined material models require bulk and shear moduli for transmitting boundaries, contact interfaces, rigid body constraints, and time-step calculations" (ANSYS 2023a, Appendix A).

We recommend that these estimates be determined from the stiffness matrix of the material. Additional information regarding how to calculate the stiffness matrix for different classes of materials has been provided in Alexander and Weerasooriya (2021). We recommend the estimates be determined from the components of the stiffness matrix, C_{ij} as

$$\begin{aligned} K &\geq \max\,(C_{11}, C_{22}, C_{33}) \\ G &\geq \max(C_{44}, C_{55}, C_{66}). \end{aligned} \tag{1}$$

The maximums are used to provide conservative estimates for the time step calculation. The present wrapper assumes that K and G are listed as the first two material properties. The user can choose which comes first and second.

- Third property: the number of state variables used in the Abaqus UMAT. This number is an integer variable called `NSTATV`.
- Fourth property: the number of material properties used in the Abaqus UMAT. This number is an integer variable called `NPROPS`.
- Fifth and subsequent properties: values for the material properties used in the Abaqus UMAT.

Based on the above, the total number of material properties listed in the `*MAT_USER` keyword, referred to here as `NPROPS_DYNA`, is calculated as

$$\text{NPROPS\_DYNA} = \text{NPROPS} + 4. \tag{2}$$

The user needs to list all of these material properties in the `*MAT_USER` keyword using as many data cards as necessary. Each card holds eight material properties. If `NPROPS_DYNA` is not a multiple of 8, then empty positions on the last card are filled with 0’s.

3.3.2 Specify Variables in the First Two Cards of the Keyword

The first two data cards (lines) of this keyword contain different variables related to LS-DYNA settings for proper execution of the material model. The specific values required for each of these variables in order to use the wrapper are described below. In this list, only the values required by the wrapper for each variable are described. The user is referred to the *LS-DYNA Keyword User’s Manual*, Volume II (ANSYS 2023b) for additional information regarding each variable and its specific meaning.

- `MID`: a material identifier. The user must choose a value that is different from any other material used in the simulation.

- RO: must be set to the mass density of the material model in consistent units.
- MT: must be set to 42 to match the title of the wrapper subroutine, which was UMAT42V.
- LMC and LMCA: must be calculated by the user based on the number of material properties listed in the *MAT_USER keyword, calculated as NPROPS_DYNA in Eq. 2. Appendix D provides further details on how LMC and LMCA are determined by NPROPS_DYNA. In summary, if NPROPS_DYNA ≤ 48:

  $$\text{LMC} = \text{NPROPS\_DYNA and LMCA} = 0. \quad (1)$$

  If NPROPS_DYNA > 48:

  $$\text{LMC} = 48 \text{ and LMCA} = \text{NPROPS\_DYNA-48}. \quad (2)$$

- NHV: must be set to NSTATV + 9. NSTATV is the number of state variables used in the Abaqus UMAT. It was also listed as the third material property sent to LS-DYNA, as noted in Section 3.5.1. Appendix E provides additional information regarding the use of state variables by the wrapper. The information provided in Appendix E is particularly important if the user wants to visualize state variables during postprocessing.
- IORTHO: must be set to 0. This causes LS-DYNA to send the variables to the wrapper in the current global coordinate system rather than a local coordinate system. The wrapper needs the variables in the global coordinate system because UMATs in Abaqus are written for the global coordinate system.
- IBULK and IG: these indicate the location of the estimates for the bulk and shear modulus, K and G from Eq. 1, within the listing of material properties as described previously. The present wrapper assumes that K and G are listed in the first two slots for material properties. The user can choose which comes first and second.
- IVECT: must be set to 1. The present wrapper is written as a vectorized user material model rather than as a serial material model. The IVECT=1 flag indicates this to LS-DYNA.
- IFAIL: must be set to 0 because the present wrapper does not include capability for elements to be deleted based on a failure criterion.

- `ITHERM`: determines if LS-DYNA sends temperature information to the user material model. We have been using `ITHERM`=0. The wrapper does not explicitly handle any temperature variables.
- `IHYPER`: must be set to 1 in order for LS-DYNA to pass the deformation gradient at the end of the current timestep. Appendix E provides additional information.
- `IEOS`: is for equation of state. We have been using `IEOS`=0.

4. Verification Using Single Element Simulations

The implementation of the wrapper was verified by running single element simulations in LS-DYNA using the EMS UMAT material model by means of the wrapper. Analogous simulations were then run in ALE3D directly using the EMS UMAT material model. Numerical outputs from LS-DYNA and ALE3D were compared.

Figure 3 describes the boundary conditions for the single element simulations. All four nodes on the bottom surface were displacement-constrained in the x, y, and z directions. All four nodes on the top surface were displacement-constrained in the y and z directions.

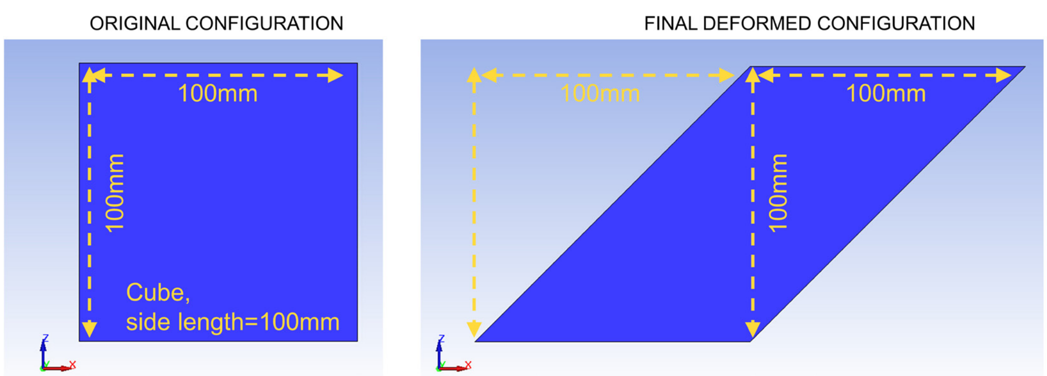

Fig. 3 Simple shear boundary conditions used for single element simulations in LS-DYNA and ALE3D to verify wrapper by comparing numerical outputs

Details regarding the EMS model have been previously provided by Becker (2019) and additionally in Alexander et al. (2023). The model includes a crystal plasticity framework consisting of a single slip system. The slip system was defined by two vectors: the normal to the slip plane and the slip direction. In the reference configuration, these two vectors were referred to as NORM0 and M0, respectively. The corresponding vectors in the deformed configuration were NORM and M1.

The first set of single element simulations modeled the slip system with NORM0=[0,0,1] and M0=[1,0,0]. The stress in the 1–3 direction at the end of the simulation was compared. The stress from LS-DYNA deviated by 0.04% from the ALE3D computation.

The second set of simulations modeled the slip system rotated by 30° relative to the x-axis such that NORM0=[-sin(30),0,cos(30)] and M0=[cos(30), 0, sin(30)]. All six components of stress were compared. The stress from LS-DYNA deviated from the ALE3D computation by less than 0.02%. In addition, the vectors NORM and M1 were compared. The LS-DYNA computation deviated by less than 0.002% from ALE3D.

5. Limitations of Using Wrapper for Other Abaqus UMATs

The present wrapper is limited by the assumed form of the Abaqus UMAT header (Fig. 2). The latest versions of Abaqus have replaced the KSTEP integer variable with an array called JSTEP. The wrapper assumes the material model is isothermal and is written for 3-D simulations. Shells and UMATs coded for 2-D would require modification. In addition, the wrapper is limited by only assigning values to 13 of the 37 variables in the header. An Abaqus UMAT that uses any of the other 24 variables, which are not explicitly assigned in the wrapper, in its calculations will not work with the present form of the wrapper. The user will need to make additions to the present wrapper by explicitly assigning values to these variables.

6. Conclusion

The present wrapper can be directly used to run LS-DYNA simulations with the EMS material model for UHMWPE that was originally developed by Becker for use in Abaqus/EPIC. In addition, this user guide describes the steps to use the present wrapper to run LS-DYNA simulations with other UMATs that were originally developed for Abaqus or EPIC, provided that the UMAT header conform to Fig. 2 and that the UMAT performs all calculations using the 13 variables that are specifically assigned values by the wrapper in its present form. Finally, the present wrapper could be extended to more generalized Abaqus UMATs by adding additional variable assignments within the wrapper for any other variables needed by the Abaqus UMAT.

7. References

Alexander SL, Becker R, Weerasooriya T. An experimental-computational framework for UHMWPE films based on deformation and failure mechanisms: general approach and application to off-axis shear loading. DEVCOM Army Research Laboratory (US); 2023 Oct. Report No.: ARL-TR-9811.

Alexander SL, Weerasooriya T. Mechanical characterization of advanced ultra-high molecular weight polyethylene (UHMWPE) composites by ultrasonic method. DEVCOM Army Research Laboratory (US); 2021 Aug. Report No.: ARL-TR-9254. https://apps.dtic.mil/sti/pdfs/AD1146645.pdf.

ANSYS. LS-DYNA keyword user's manual. Version R14.0. ANSYS; 2023a July 31. (Vol. I).

ANSYS. LS-DYNA keyword user's manual. Version R14.0. ANSYS; 2023b Feb 24. (Material models; Vol. II).

Becker R. Investigation of slip kinematics in an ultra-high molecular weight polyethylene (UHMWPE) laminar composite model. DEVCOM Army Research Laboratory (US); 2019 Dec. Report No.: ARL-TN-0985. https://apps.dtic.mil/sti/pdfs/AD1087651.pdf.

Hampton CE, Alexander SL, Weerasooriya T. Conversion of a user-defined bone material model from Abaqus to LS-DYNA. DEVCOM Army Research Laboratory (US); 2022 Oct. Report No.: ARL-TR-9594. https://apps.dtic.mil/sti/trecms/pdf/AD1182198.pdf.

Appendix A. The wrapper.for file Containing Wrapper Code with Utility Subroutines

This appendix appears in its original form, without editorial change.

```
C    ASSUMES THAT UMAT_ABAQUS.FOR CONTAINS THE ABAQUS UMAT
C    AND ANY UTILITY SUBROUTINES THAT THE ABAQUS UMAT USES
      include 'umat_abaqus.for'
C
      subroutine umat42v(cm,d1,d2,d3,d4,d5,d6,sig1,sig2,
     . sig3,sig4,sig5,sig6,epsps,hsvs,lft,llt,dt1siz,capa,
     . etype,tt,temps,failels,nlqa,crv,nnpcrv,cma,qmat,elsizv,idelev,
     . reject)
C
      implicit none
C
      include 'nlqparm'
      include 'nhisparm.inc'
      include 'bk06.inc'
      include 'iounits.inc'
C
      REAL*8 cm(*),d1(*),d2(*),d3(*),d4(*),d5(*),d6(*),
     & sig1(*),sig2(*),sig3(*),sig4(*),sig5(*),sig6(*),
     & epsps(*),hsvs(nlqa,*),dt1siz(*),temps(*),crv(lq1,2,*),cma(*),
     & failels(*),qmat(nlq,3,3),elsizv(*)
      integer nnpcrv(*)
      integer*8 idelev(*)
      character*5 etype
      logical failels,reject
C
C     DECLARATIONS FOR ABAQUS VARIABLES
C
C     FIRST, PARAMETERS WHICH MUST BE CONFIRMED BY THE USER:
      INTEGER,PARAMETER ::  NTENS = 6,  NDI = 3, NSHR = 3
                !NTENS  = number of independent stress components
      INTEGER, DIMENSION(1) ::  elems_to_print  = (/ 1/) !information for these elements will be printed.
C
C     THESE NEXT VARIABLES DEAL WITH THE MATERIAL PROPERTIES AND STATE VARIABLES
C       Explanation:
C
C       NPROPS          :   total number of material properties used in the Abaqus UMAT
C       NSTATV          :   total number of state variables used in the Abaqus UMAT
C       totalNumProps   :   total number of material properties that user provides values for
C                           to the *MAT_USER keyword in LS-DYNA
C               It is assumed that the material properties that the user provides values for
C               in the *MAT_USER keyword have the following order:
C                   Positions 1 and 2 = estimates for bulk/shear moduli
C                   Position 3 = NSTATV
C                   Position 4 = NPROPS
C                   Starting at Position 5 user starts listing all the material properties used in the Abaqus UMAT.
C
      INTEGER, PARAMETER    ::  start_Abaqus_props = 5
                !index within CM array indicating start of material properties to send to Abaqus UMAT
                !assumes that remaining indices within CM array only contain material properties to send to Abaqus UMAT.
      integer numOtherProps !number of properties in CM array that are not properties to send to Abaqus UMAT
      integer totalNumProps !total number of properties listed in *MAT USER keyword
      INTEGER,PARAMETER :: CM_ARRAY_CUTOFF = 48
                !maximum value of material properties that can all be contained in the LS-DYNA array called CM.
                !if totalNumProps .LE. CM_ARRAY_CUTOFF, all the material properties are in the array CM.
                !otherwise, the remaining properties (totalNumProps - CM ARRAY CUTOFF) are in array CMA.
                !According to LS-DYNA Manual: CM_ARRAY_CUTOFF = 48 if IORTHO=0
```

```fortran
                !                              CM_ARRAY_CUTOFF = 40 if IORTHO=1
      INTEGER,PARAMETER :: DIMPROPS = 100, DIMSV = 100
C               These are only estimates used to dimension the PROPS and STATEV arrays below:
C                   DIMPROPS is an estimate for NPROPS, and DIMSV is an estimate for NSTATEV
C                   These estimates would need to be increased if NPROPS>100 or NSTATEV>100
C
C     OTHER ABAQUS VARIABLES
      INTEGER KINC, KSPT, KSTEP, LAYER, NOEL, NPT, NSTATV, NPROPS
      REAL*8 CMNAME
      REAL*8 CELENT, COORDS(3), DDSDDE(NTENS,NTENS), DDSDDT(NTENS)
      REAL*8 F0(3,3), F(3,3), DPRED(1), DROT(3,3)
      REAL*8 DRPLDE(NTENS), DRPLDT, DSTRAN(NTENS), DTEMP, DTIME, PNEWDT
      REAL*8 PREDEF(1), PROPS(DIMPROPS), RPL, SCD, SPD, SSE
      REAL*8 STATEV(DIMSV), STRAN(NTENS), STRESS(NTENS), TEMP, TIME(2)
C
C     DECLARATIONS FOR LOCAL VARIABLES
      integer km, i, j, iCount
      NSTATV    = NINT(cm(3))
      NPROPS    = NINT(cm(4))
      numOtherProps = start_Abaqus_props-1
      totalNumProps = NPROPS+numOtherProps
C
C     CHECK NSTATV AND NPROPS AGAINST DIMENSIONS OF PROPS AND STATEV
      if (NPROPS .GT. DIMPROPS) then
        write(iotty,*) 'ERROR: NPROPS LARGER THAN ',
     *                 'DEFAULT SIZE OF PROPS ARRAY'
        write(iotty,*) 'NPROPS: ', NPROPS
        write(iotty,*) 'DEFAULT SIZE: ', DIMPROPS
        write(iotty,*) 'INCREASE DIMPROPS ON LINE 56 IN WRAPPER.FOR'
        STOP
      endif
      if (NSTATV .GT. DIMSV) then
        write(iotty,*) 'ERROR: NSTATV LARGER THAN ',
     *                 'DEFAULT SIZE OF STATEV'
        write(iotty,*) 'NSTATV: ', NSTATV
        write(iotty,*) 'DEFAULT SIZE: ', DIMSV
        write(iotty,*) 'INCREASE DIMSV ON LINE 56 IN WRAPPER.FOR'
        STOP
      endif
C***************************************************************************
C
C
C
C     VARIABLES THAT ARE THE SAME FOR ALL ELEMENTS:
      TIME(1)    = tt          !total simulation time
      TIME(2)    = tt
      DTIME      = dt1siz(1)    !delta time (increment step size)
      if (totalNumProps .LE. CM_ARRAY_CUTOFF) then
        do i = start_Abaqus_props, totalNumProps
            PROPS(i-numOtherProps) = cm(i)
        enddo
      else
        do i = start_Abaqus_props, CM_ARRAY_CUTOFF
            PROPS(i-numOtherProps) = cm(i)
        enddo
        do i = CM_ARRAY_CUTOFF+1, totalNumProps
            PROPS(i-numOtherProps) = cma(i-CM_ARRAY_CUTOFF)
        enddo
      endif
C
C
C
```

```
C     CYCLE THROUGH THE ELEMENTS IN THE BLOCK
      ELEMENTCYCLE: do km = lft, llt
C
C     DEBUG PRINTING
      if (ncycle .EQ. 0) then
      if (ANY(elems_to_print == idelev(km))) then
          write(iotty,*) '*******************************************'
          write(iotty,*) 'CYCLE: ', ncycle
          write(iotty,*) 'CURRENT TIME: ', TIME(1)
          write(iotty,*) 'DTIME: ', DTIME
          write(iotty,*) 'PROPERTY CHECK: '
          do i = start_Abaqus_props,totalNumProps
            write(iotty,*) 'PROPS : ', i-numOtherProps ,
     *                                PROPS(i-numOtherProps)
            if (i .LE. CM_ARRAY_CUTOFF) then
                write(iotty,*) 'CM : ', i , cm(i)
            else
                write(iotty,*) 'CMA : ', i , cma(i-CM_ARRAY_CUTOFF)
            endif
          enddo
      endif
      endif
C
C     STRESS/STRAIN
C       Note that component ordering is different:
C           ABAQUS:   11,22,33,12,13,23
C           LS-DYNA:  11,22,33,12,23,31
      STRESS(1) = SIG1(km)
      STRESS(2) = SIG2(km)
      STRESS(3) = SIG3(km)
      STRESS(4) = SIG4(km)
      STRESS(5) = SIG6(km)    !ATTENTION TO CHANGED ORDER
      STRESS(6) = SIG5(km)    !ATTENTION TO CHANGED ORDER
C
      DSTRAN(1) = d1(km)
      DSTRAN(2) = d2(km)
      DSTRAN(3) = d3(km)
      DSTRAN(4) = d4(km)
      DSTRAN(5) = d6(km)
      DSTRAN(6) = d5(km)
C
C     STATE-DEPENDENT VARIABLES:
      do i = 1, NSTATV
        STATEV(i) = hsvs(km,i)
      enddo
C
C     LOAD DEFORMATION GRADIENT AT END OF CURRENT TIMESTEP:
      iCount = 1
      do i = 1,3
        do j = 1,3
            F(j,i) = hsvs(km,NSTATV+9+iCount)
            iCount = iCount+1
        enddo
      enddo
C
C     LOAD DEFORMATION GRADIENT AT END OF PREVIOUS TIMESTEP
C     AND
C     SAVE DEFORMATION GRADIENT AT END OF CURRENT TIMESTEP
      iCount = 1
      do i = 1,3
        do j = 1,3
            F0(j,i) = hsvs(km,NSTATV+iCount)
            hsvs(km,NSTATV+iCount) = F(j,i)
            iCount = iCount+1
        enddo
      enddo
```

```fortran
C FOR THE FIRST CYCLE:
C   OVER-RIDE F0 AND SET IT TO IDENTITY MATRIX
      if (ncycle .EQ. 0) CALL identityMat(F0)
C
C     CALCULATE ROTATION INCREMENT
      CALL CALC_DROT_FROM_DEFGRAD(F0,F,DROT)
C
C     CALL THE EXTERNAL UMAT (CONTAINED IN THE UMAT_ABAQUS.FOR FILE):
      CALL UMAT_ABAQUS( STRESS, STATEV, DDSDDE, SSE, SPD, SCD,
     & RPL, DDSDDT, DRPLDE, DRPLDT, STRAN, DSTRAN, TIME, DTIME,
     & TEMP, DTEMP, PREDEF, DPRED, CMNAME, NDI, NSHR, NTENS, NSTATV,
     & PROPS, NPROPS, COORDS, DROT, PNEWDT, CELENT, F0, F,
     & NOEL, NPT, LAYER, KSPT, KSTEP, KINC)
C
C     CONVERT OUT THE TWO VARIABLES WHICH MUST BE SENT BACK TO LS-DYNA:
C     (1) STRESS:
      SIG1(km) = STRESS(1)
      SIG2(km) = STRESS(2)
      SIG3(km) = STRESS(3)
      SIG4(km) = STRESS(4)
      SIG5(km) = STRESS(6)     !ATTENTION TO CHANGED ORDER
      SIG6(km) = STRESS(5)     !ATTENTION TO CHANGED ORDER
C     (2) STATE-DEPENDENT VARIABLES USED BY EXTERNAL UMAT:
      do i = 1, NSTATV
        hsvs(km,i) = STATEV(i)
      enddo
C
      END DO ELEMENTCYCLE
      end subroutine umat42v
C***********************************************************************
C
C     CALCULATE ROTATION INCREMENT (DROT) USING DEFORMATION GRADIENTS
C       (ALGORITHM FROM R BECKER SEP2022)
C
C***********************************************************************
      subroutine CALC_DROT_FROM_DEFGRAD(F0,F,DROT)
        implicit none
        !DECLARATIONS FOR ARGUMENTS:
        REAL*8, intent(in)     ::  F0(3,3),F(3,3)
        REAL*8, intent(out)    ::  DROT(3,3)
        !LOCAL DECLARATIONS:
        REAL*8  ::  defGrad_increment(3,3), defGrad_midstep(3,3)
        REAL*8  ::  defGrad_midstep_inv(3,3)
        REAL*8  ::  velocityGrad_midstep(3,3),spin_increment(3,3)
        REAL*8  ::  identity(3,3)
        REAL*8  ::  firstParenthesis(3,3), firstParenthesis_inv(3,3)
        REAL*8  ::  secondParenthesis(3,3)
        REAL*8  ::  testMat(3,3)
        INTEGER ::  i
        !
        !increment in deformation gradient
        defGrad_increment = F-F0
        !Mid-step deformation gradient
        defGrad_midstep = F0+0.5d0*defGrad_increment
        !
        !CALCULATE VELOCITY GRADIENT AT MID-STEP CONFIGURATION
        !   (1) Find inverse of mid-step deformation gradient
        CALL a33inverse(defGrad_midstep,defGrad_midstep_inv)
        !   (2) Velocity gradient (x delta time) at mid-step configuration:
        velocityGrad_midstep =
     *      MATMUL(defGrad_increment,defGrad_midstep_inv)
        !
        !SPIN TIMES DELTA T:
        spin_increment = 0.5d0*(velocityGrad_midstep-
     *                          TRANSPOSE(velocityGrad_midstep))
        !
```

Appendix B. Variables from Abaqus UMAT Heading That Are Used in Elemental Macroscale Material Model

Figure 2 in the report provided the form that the present wrapper requires for the Abaqus UMAT header. The figure shows that there were 37 total variables in the header.

The wrapper was developed for the specific use case of the elemental macroscale (EMS) material model developed by Becker.[1] This appendix provides information on which of the 37 variables are used within the EMS material model. Knowing which variables were used within the EMS material model can aid the user in understanding how to adapt the present wrapper to a more generalized use case.

All 37 variables are listed below. Variables listed in red font are specifically assigned values within the wrapper and can be used in calculations within the Abaqus UMAT. The single variable listed in blue font, DDSDDE, is calculated within the EMS material model for use in implicit codes. However, the present form of the wrapper does not use this variable in any way, and its value is never sent back to LS-DYNA. The remaining variables listed in black font are not used in the EMS material model and are never explicitly assigned values within the wrapper. Variables listed with parentheses are arrays, with dimensions as indicated by the parentheses. Variables without parentheses are scalars.

- STRESS(NTENS)
  - INTENT=IN/OUT
  - "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."[2]
  - Component ordering: 11,22,33,12,13,23
- STATEV(NSTATV)
  - INTENT=IN/OUT
  - State variables
- DDSDDE(NTENS,NTENS)
  - INTENT=OUT
- SSE

[1] Becker R. Investigation of slip kinematics in an ultra-high molecular weight polyethylene (UHMWPE) laminar composite model. DEVCOM Army Research Laboratory (US); 2019 Dec. Report No.: ARL-TN-0985. https://apps.dtic.mil/sti/pdfs/AD1087651.pdf

[2] Abaqus. SIMULIA user assistance 2023. Dassault Systèmes; 2023.

- NOT USED
- SPD
  - NOT USED
- SCD
  - NOT USED
- RPL
  - NOT USED
- DDSDDT(NTENS)
  - NOT USED
- DRPLDE(NTENS)
  - NOT USED
- DRPLDT
  - NOT USED
- STRAN(NTENS)
  - NOT USED
- DSTRAN(NTENS)
  - INTENT = IN
  - “array of strain increments”[2]
  - Component ordering: 11,22,33,12,13,23
- TIME(2)
  - NOT USED
- DTIME
  - INTENT = IN
  - Delta time
- TEMP
  - NOT USED
- PREDEF(1)

- NOT USED

- DPRED(1)
  - NOT USED
- CMNAME
  - NOT USED
- NDI [integer]
  - "Number of direct stress components."[2] Hard-coded in wrapper to be 3, for a full 3-D model.
- NSHR [integer]
  - "Number of engineering shear stress components."[2] Hard-coded in wrapper to be 3, for a full 3-D model.
- NTENS [integer]
  - INTENT = IN (used for dimensioning arrays)
  - "Size of the stress or strain component array (NDI+NSHR)."[2] The number of direct stress components plus the number of engineering shear stress components. This is currently hard-coded in the wrapper to be 6.
- NSTATV [integer]
  - INTENT = IN
  - "number of solution-dependent state variables"[2]
- PROPS(NPROPS)
  - INTENT = IN
  - "user-specified array of material constants"[2]
  - LS-DYNA equivalents are CM and CMA
- NPROPS [integer]
  - INTENT = IN
  - Number of material constants (dimension of PROPS)
- COORDS(3)
  - NOT USED

- DROT(3,3)
    - INTENT = IN
    - "Rotation increment matrix. This matrix represents the increment of rigid body rotation of the basis system in which the components of stress and strain are stored. It is provided so that vector- or tensor-valued state variables can be rotated appropriately [but] stress and strain components are already rotated by this amount before UMAT is called."[2]
- PNEWDT
    - NOT USED
- CELENT
    - NOT USED
- F0(3,3)
    - INTENT = IN
    - "Deformation gradient at the beginning of the increment."[2]
- F(3,3)
    - INTENT=IN
    - "Deformation gradient at the end of the increment."[2]
- NOEL, NPT, LAYER, KSPT, KSTEP, KINC [integers]
    - NOT USED

Appendix C. Confirming Double Precision in LS-DYNA

The present wrapper was compiled using a double precision distribution of LS-DYNA. Variables within subroutines were automatically double precision if they were not explicitly declared as a different type. Figure C-1 provides example code that could be added to `UMAT42V` to confirm double precision. Figure C-2 shows output from the write statements when using a double precision distribution of LS-DYNA.

```
C
C      CONFIRMATION OF DOUBLE PRECISION
C      TEST_A is never declared
       REAL*8 TEST_B
       REAL*4 TEST_C
       TEST_A = 0.123456789012345678901234567890d0
       TEST_B = 0.123456789012345678901234567890d0
       TEST_C = 0.123456789012345678901234567890d0
       write(iotty,*) 'kind(TEST_A) = ', kind(test_A)
       write(iotty,10) TEST_A
       write(iotty,*) 'kind(TEST_B) = ', kind(test_B)
       write(iotty,10) TEST_B
       write(iotty,*) 'kind(TEST_C) = ', kind(test_C)
       write(iotty,10) TEST_C
    10 FORMAT(f31.29)
```

Fig. C-1 Example of code to add to the wrapper to confirm that the LS-DYNA distribution is running in double precision. Expected outputs from the write statements are provided in Fig. C-2.

```
 kind(TEST_A) = 8
0.12345678901234567736988623210
 kind(TEST_B) = 8
0.12345678901234567736988623210
 kind(TEST_C) = 4
0.12345679104328155517578125000
```

Fig. C-2 Expected output from the write statements shown in Fig. C-1 when using a double-precision distribution of LS-DYNA. Highlighted portions show errors relative to the variable assignments in Fig. C-1.

Appendix D. Determining LMC and LMCA: Length of the Arrays of Material Properties in LS-DYNA

LS-DYNA stores material properties in two different arrays called CM and CMA. The *MAT_USER_DEFINED_MATERIAL_MODELS keyword requires the user to specify the LMC and LMCA, which are the lengths of CM and CMA. This appendix provides additional background information to supplement the directions provided in Section 3.5.2 for calculating LMC and LMCA.

The total number of material properties that are specified in the k-file was calculated in Section 3.5.1 as NPROPS_DYNA. LS-DYNA has a threshold value, referred to here as X. If NPROPS_DYNA ≤ X, all material properties are stored in an array called CM. If NPROPS_DYNA > X, the first X number of material properties are stored in CM, and the remaining properties are stored in an array called CMA.

The value of X depends on the value of the IORTHO variable in the *MAT_USER keyword (reference volume II of the LS-DYNA manual).[1] The present wrapper requires that IORTHO be set to 0. Therefore, the threshold is X=48.

[1] ANSYS. LS-DYNA keyword user's manual. Version R14.0. ANSYS; 2023b Feb 24. (Material models; Vol. II).

Appendix E. The Use of State Variables by the Wrapper, Including How Deformation Gradients Are Stored

The *MAT_USER_DEFINED_MATERIAL_MODELS keyword requires the user to specify the number of state variables (also referred to as history variables) required for the material model, NHV. Section 3.5.2 notes that NHV is calculated as

$$\text{NHV}=\text{NSTATV} + 9, \tag{E-1}$$

where NSTATV is the number of state variables used in the Abaqus UMAT. The reasoning for Eq. E-1 is due to the storage of the deformation gradient. This appendix provides additional information about how state variables are used by the wrapper, specifically how the deformation gradients are stored.

The wrapper provides the deformation gradient from the end of the previous timestep, F0, and the deformation gradient at the end of the current time step, F, to the Abaqus UMAT. To our knowledge, LS-DYNA does not provide F0. Therefore, the wrapper stores the nine components of F0 at the end of the state variable array.

LS-DYNA can provide F, but the user must specifically request this by setting the IHYPER variable in the keyword to IHYPER=1. Then, LS-DYNA appends the nine components of F to the end of the array of state variables, after the NHV state variables.

In summary, the state variable array that LS-DYNA supplies to the wrapper and that the wrapper sends back to LS-DYNA has the following structure:

- First are the state variables that are specifically used within the Abaqus UMAT. The number of these variables is NSTATV.
- Next are the nine components of F0 (NSTATEV+1 to NSTATEV+9).
- Finally are the nine components of F (NSTATEV+10 to NSTATEV+18).

Knowledge of this ordering is important if the user wants to visualize the state dependent variables during postprocessing.

Table E-1 lists the history variables that used by Alexander et al.[1] to run in LS-DYNA the EMS UMAT that had been developed earlier by Becker.[2] The EMS UMAT required NPROPS=24 state variables. Therefore, NHV was 33 by Eq. E-1. Finally, since IHYPER=1, the total number of state variables was 42.

[1] Alexander SL, Becker R, Weerasooriya T. An experimental-computational framework for UHMWPE films based on deformation and failure mechanisms: general approach and application to off-axis shear loading. DEVCOM Army Research Laboratory (US); 2023 Oct. Report No.: ARL-TR-9811.

[2] Becker R. Investigation of slip kinematics in an ultra-high molecular weight polyethylene (UHMWPE) laminar composite model. DEVCOM Army Research Laboratory (US); 2019 Dec. Report No.: ARL-TN-0985. https://apps.dtic.mil/sti/pdfs/AD1087651.pdf.

Table E-1 List of the history variables in LS-DYNA array for using EMS UMAT[2]

Number (position in array)	Variable name	Description
The first 24 state variables are specific to EMS UMAT.[2] Wrapper sends only these first 24 to the EMS UMAT.		
1	`SIGM(1)`	Matrix Cauchy stress: xx
2	`SIGM(2)`	Matrix Cauchy stress: yy
3	`SIGM(3)`	Matrix Cauchy stress: zz
4	`SIGM(4)`	Matrix Cauchy stress: xy
5	`SIGM(5)`	Matrix Cauchy stress: zx
6	`SIGM(6)`	Matrix Cauchy stress: yz
7	`SSTRAN`	Integrated interlaminar shear strain
8	`EPSP`	Effective plastic strain in the matrix
9	`KEEP`	Calculated on exit from EMS UMAT
10	`SS`	Sound speed to set the time step
11	`ERODSTR`	Erosion strain
12	`PLACE`	
13	`NORM(1)`	Normal to fiber plane, current configuration
14	`NORM(2)`	Normal to fiber plane, current configuration
15	`NORM(3)`	Normal to fiber plane, current configuration
16	`M1(1,1)`	Direction of 1st fiber, current configuration
17	`M1(2,1)`	Direction of 1st fiber, current configuration
18	`M1(3,1)`	Direction of 1st fiber, current configuration
19	`STRCH0(1)`	Max. stretch for fiber #1
20	`STRCH0(2)`	Max. stretch for fiber #1
21	`STRCH(1)`	Current fiber stretch for fiber #1
22	`STRCH(2)`	Current fiber stretch for fiber #1
23	`FFLAG(1)`	Integrated damage for fiber #1
24	`FFLAG(2)`	Integrated damage for fiber #1
Remaining state variables are used to hold F0 and F. Wrapper extracts F0 and F as separate arrays.		
25	`F0(1,1)`	Deformation gradient at end of previous time increment
26	`F0(2,1)`	Deformation gradient at end of previous time increment
27	`F0(3,1)`	Deformation gradient at end of previous time increment
28	`F0(1,2)`	Deformation gradient at end of previous time increment
29	`F0(2,2)`	Deformation gradient at end of previous time increment
30	`F0(3,2)`	Deformation gradient at end of previous time increment
31	`F0(1,3)`	Deformation gradient at end of previous time increment
32	`F0(2,3)`	Deformation gradient at end of previous time increment
33	`F0(3,3)`	Deformation gradient at end of previous time increment
34	`F(1,1)`	Deformation gradient at end of current time increment
35	`F(2,1)`	Deformation gradient at end of current time increment
36	`F(3,1)`	Deformation gradient at end of current time increment
37	`F(1,2)`	Deformation gradient at end of current time increment
38	`F(2,2)`	Deformation gradient at end of current time increment
39	`F(3,2)`	Deformation gradient at end of current time increment
40	`F(1,3)`	Deformation gradient at end of current time increment
41	`F(2,3)`	Deformation gradient at end of current time increment
42	`F(3,3)`	Deformation gradient at end of current time increment

List of Symbols, Abbreviations, and Acronyms

2-D/3-D	two-dimensional/three-dimensional
EMS	elemental macroscale
FE	finite element
SMS	sub-mesoscale
UHMWPE	ultra-high-molecular-weight polyethylene
UMAT	user-defined material model

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

1 DEVCOM ARL
(PDF) FCDD RLD CL
TECH LIB

13 DEVCOM SOLDIER CNTR
(PDF) M G CARBONI
D COLANTO
R DILLALLA
B FASEL
C HEWITT
J KIREJCZYK
R KOLLAR
D KUBIAK
C LINNINGTON
D OTTERSON
J PARKER
W SHAW
A STRZEPEK

4 PROG EXECUTIVE OFC
(PDF) SOLDIER
A DEGROOT
J HOPPING
P LOOMIS
N NGUYEN

1 PROG EXECUTIVE OFC
(PDF) IEWS
A FOURNIER

1 MTRL SCIENCES DIV
(PDF) LAWRENCE BERKELY
NATL LAB
R RITCHIE

3 SOUTHWEST RSRCH INST
(PDF) C ANDERSON JR
S CHOCRON
D NICOLELLA

2 NIST
(PDF) A FORSTER
M VANLANDINGHAM

5 MRDC DOD BLAST INJURY
(PDF) RSRCH PROGRAM COOR OFC
R GUPTA
A LEWIS
T PIEHLER
R SHOGE
R SPENCER

1 INST FOR DEFNS ANLYS
(PDF) Y MACHERET
YMACHERE@IDA.ORG

1 USMC
(PDF) C STEPHENS

4 US ARMY AEROMEDICAL
(PDF) RSRCH LAB
F BROZOSKI
V CHANCEY
B MCENTIRE
T ROOKS

1 IRREGULAR WARFARE TECH
(PDF) SUPPORT DIR
L TANYAG

1 DEVCOM GROUND VEHICLE
(PDF) SYSTEMS CENTER
R SCHERER

1 DEVCOM
(PDF) AMSRD PE
D RUSIN

2 DEVCOM CHEMICAL
(PDF) BIOLOGICAL CENTER
M HORSMON
N VINCELLI

1 OSD DOT&E
(PDF) J IVANCIK

5 US NAVAL RESEARCH
(PDF) LABORATORY
A BAGCHI
A ILIOPOULOS
J MICHOPOULOS
K TEFERRA
X TAN

3 DAC
(PDF) FCDD DAG S
K LOFTIS
FCDD DAS LBW
G DIETRICH
FCDD DAS LBE
J GURGANUS

2 DEVCOM ATLANTIC
(PDF) S COLEMAN
H PIETSCH

3 (PDF) USSOCOM
M CLARK
D GUILLEMIN
N TSANTINIS

142 (PDF) DEVCOM ARL
FCDD RLA A
S KARNA
P FRANASZCZUK
J NEWILL
A RAWLETT
S SCHOENFELD
FCDD RLA B
A WEST
M TSCHOPP
FCDD RLA CB
R BECKER
J CAMPBELL
P GILLICH
C HOPPEL
J LASALVIA
A TONGE
FCDD RLA FF
W HAIRSTON
FCDD RLA HC
A DAGRO
A EIDSMORE
T THOMAS
FCDD RLA M
B CHEESEMAN
K CHO
FCDD RLA MA
K BERNETICH
T BOGETTI
S BOYD
J CAIN
D KNORR
M NEBLETT
E SANDOZ-ROSADO
J SANDS
J STANISZEWSKI
M YEAGER
FCDD RLA MB
G GAZONAS
D GRAY
D MAGAGNOSC
P MOY
D O'BRIEN
J SIETINS
T WALTER
FCDD RLA MC
D CRAWFORD
R JENSEN
J SNYDER
FCDD RLA MD
A BUJANDA
J LA SCALA
E WETZEL
FCDD RLA ME
P PATEL
J SWAB
L VARGAS-GONZALEZ
FCDD RLA MF
K DARLING
D FIELD
A GIRI
S GRENDAHL
C HAINES
K LIMMER
H MURDOCH
FCDD RLA MG
J LENHART
R MROZEK
J ORLICKI
T SIRK
I YEH
FCDD RLA T
M FERMEN-COKER
R FRANCART
T HOLDREN
R YEAGER
FCDD RLA TA
S BILYK
M GRAHAM
S TURNAGE
W UHLIG
C WILLIAMS
FCDD RLA TB
S ALEXANDER
T BAUMER
D CASEM
J CLAYTON
B FAGAN
A GOERTZ
A GUNNARSSON
C HAMPTON
M KLEINBERGER
D KRAYTERMAN
E MATHEIS
J MCDONALD
P MCKEE
C MEREDITH
T PLAISTED
K RAFAELS
S SATAPATHY
L SHANNAHAN
M TEGTMEYER
C WEAVER
T WEERASOORIYA
S WOZNIAK
T ZHANG
FCDD RLA TD

R DONEY
R GUPTA
M KEELE
D KLEPONIS
B KRZEWINSKI
K MASSER
F MURPHY
C RANDOW
S SCHRAML
K STOFFEL
M ZELLNER
FCDD RLA TE
M BURKINS
D GALLARDY
W GOOCH
E KLIER
J LLOYD
M LOVE
P SWOBODA
FCDD RLA TF
J ANGEL
W BRUCHEY
J CAZAMIAS
R COATES
T EHLERS
P JANNOTTI
E KENNEDY
R LEAVY
J LEE
L MAGNESS
D MALLICK
C MEYER
J RUNYEON
FCDD RLA TG
C CUMMINS
D FOX
N GNIAZDOWSKI
S HUG
S KUKUCK
C PECORA
FCDD RLA V
S SILTON
FCDD RLA VA
R EMERSON
A GHOSHAL
FCDD RLA VB
A HALL
FCDD RLA W
T V SHEPPARD
FCDD RLA WA
J BRENNAN
FCDD RLA WC
M MINNICINO
FCDD RLB
J ZABINSKI
FCDD RLR A
J CIEZAK-JENKINS
D STEPP
FCDD RLR EM
A BROWN
C VARANASI
FCDD RLR ET
B ASHFORD
D COLE
D FORD
FCDD RLR EW
R ANTHENIEN

1 (PDF) SANDIA NATL LABS
B SANBORN

1 (PDF) LANGLEY RESEARCH CENTER
NASA
J CLINE