TKI 2016 XIV Konferencja Naukowo-Techniczna TECHNIKI KOMPUTEROWE W INŻYNIERII

Strain Rate Dependent User Defined Material Subroutine in LS-DYNA

Leszek Flis

Zakład Budowy Maszyn, Instytut Budowy i Eksploatacji Okrętów
Wydział Mechaniczno-Elektryczny Akademii Marynarki Wojennej w Gdyni
email: l.flis@amw.gdynia.pl

ABSTRACT: The user-defined subroutines in LS-DYNA allow the program to be customized for particular applications. The user-defined subroutines allow constitutive models but also equation of state, user defined elements and more [3] be added to the program. In the article it will demonstrate problem of the implementation own a user material subroutine for LS-Dyna and demonstrate it with an example utilizing the strain rate dependent user defined material subroutine model (UMAT). Detailed descriptions are given of the data required for the UMAT, the additional statements to be included, the variable that are available within the subroutine, and freedom the user has in defining complex material models. Examples of UMAT and flow charts were used to illustrate the points made in the article.

KEY WORDS: LS-Dyna, user defined subroutine, strain rate

1. Introduction

The Finite Elements Method today based in practice on proven commercial software applications like ANSYS and ALTAIR HyperWorks. In use, they are still ABAQUS, MSC NASTRAN and MARC and or till LS-DYNA. There are a lot free software like CalculiX, Elmer or OpenFOAM [1] and the newly established worth watching, such as IMPETUS [2]. All of them apart from LS-DYNA have one common disadvantages they are not programmable by the user but if they are it is not user friendly and not well documented. This means that the user of FEM programs can only use these options program, which imposed a programmer and in all of these programs do not differ significantly from each other.

The exception is the user-defined features in LS-DYNA which are powerful tools that allow users in academia or industry to verify research results in the context of general and complicated finite element applications. An overview of current user-defined interfaces in LS-DYNA is presented in [3,4]. Official document [5] describes technical bases of using UMAT and in [6] we can find the practical examples of implementation UMAT.

2. Material Model Theoretical Development

2.1. Isotropic von Mises Model

For the plane stress condition, the von Mises yield criterion simplifies to the following as a function of the stress tensor components:

$$\sigma_{11}^2 + \sigma_{22}^2 - \sigma_{11}^2\sigma_{22}^2 + 3\sigma_{23}^2 = \sigma_y^2 \qquad (1)$$

where σ_y is the flow stress which corresponds to data from uniaxial tensile tests.

2.2 Flow Stress

Flow stress represents the size of the yield function during deformation. An appropriate equation describing changes in the flow stress of the material depends on deformation conditions such as temperature, strain rate, etc.

In this example we will use the Power-Law model which is written as follows:

$$\bar{\sigma}^p(\bar{\varepsilon}^p) = K(\bar{\varepsilon}^p + \varepsilon_0)^n \qquad (2)$$

where K is the strength hardening coefficient and n strain-hardening exponent are material constants that are extracted from experimental tensile tests. $\bar{\varepsilon}^p$ is the is the effective plastic strain and ε_0 is a constant representing the elastic strain at yield which can be either supplied or calculated as follows:

$$\varepsilon_0 = \left(\frac{E}{K}\right)^{\left(\frac{1}{n-1}\right)} \qquad (3)$$

In most cases at low stress level, the stresses in elastoplastic material depend only on the state of strain; however, above a certain stress level, called the yield stress, non-recoverable plastic deformations are obtained.

In case of strain rate effect should be taken into consideration there is other Johnson-Cook model which has the following general relationship that defines the yield strength in terms of plastic strain, strain rate and temperature:

$$\sigma = \left(A + B\varepsilon^n\right)\left[1 + C\ln\left(\frac{\dot{\varepsilon}}{\dot{\varepsilon}_0}\right)\right]\left[1 - (T^*)^m\right] \qquad (1)$$

where: σ – von Mises flow stress,
A – yield stress,
B – effects of strain hardening,
ε – equivalent, effective plastic strain,
n – exponent strengthening,
C – strain rate constant,
$\dot{\varepsilon}$ – strain rate,
$\dot{\varepsilon}_0$– threshold strain rate,
$\dot{\varepsilon}^* = \frac{\dot{\varepsilon}}{\dot{\varepsilon}_0}$ – dimension less plastic strain rate,
T^* – homologous temperaturę,
m – temperature exponent.

All above and any sophisticated constitutive equations can be implemented in UMAT. The all required steps to will be presented in the article.

3. Implementation of the UMAT

3.1. Fortran implementation of the UMAT

An example of the theoretical models (discussed in paragraph 2) and the stress integration (return mapping) procedure are implemented in Fortran. The file generated will be used in the next paragraph to compile a working copy of LS-Dyna and then used to run sample models. The UMAT Fortran file will be presented in this paragraph.

3.2. Implementation In LS-Dyna & Compilation

After development of the UMAT, the code must be incorporated into LS-Dyna so as to correctly compile a working LS-Dyna executable. LS-Dyna provides object files and multiple source routine where you can add the developed UMAT to them after which they can be compiled into a working executable. This executable can then be used to run the FEA simulations. In this paragraph it will explain how to successfully incorporate user code into LS-Dyna and then compile the overall code. Details about the implementation will be discussed in this paragraph.

```
   subroutine umat43 (cm,eps,sig,epsp,hsv,dt1,capa,etype,tt,
   1 temper,failel,crv)
c
c***********************************************************
c|  Livermore Software Technology Corporation  (LSTC)      |
c|  -------------------------------------------------------|
c|  Copyright 1987-2008 Livermore Software Tech. Corp      |
c|  All rights reserved                                    |
c***********************************************************
c
      include 'nlqparm'
      include 'iounits.inc'
      common/bk06/idmmy,iaddp,ifil,maxsiz,ncycle,time(2,30)
      dimension cm(*),eps(*),sig(*),hsv(*),crv(lq1,2,*)
      character*5 etype
      logical failel
c
      if (ncycle.eq.1) then
        call usermsg('mat43')
      endif
c
c***********************************************
 
ENTER YOUR CODE HERE
 
c***********************************************
c
      return
      end
```

Fig. 1. The view of the UMAT subroutine with the place for own code

4. UMAT Verification Examples

In this paragraph, it will be verified and compiled UMAT by running a couple of examples and compare the results against analytical (predicted) results. UMAT verification is an important step in any UMAT development in order to guarantee a high level of confidence and quality in the developed material model.

After successfully developing the user material subroutine and compiling it, it is needed to verify the accuracy of the implementation against known values. We will verify the code using two models: one element model and tensile test model.

In the one tensile model, a one square element with unity dimensions (1mmx1mm) will be stretched in the x-direction. The model is shown in Fig. 2.

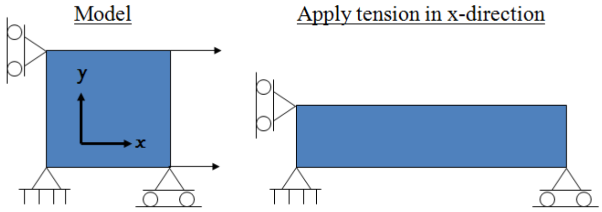

Fig. 2. Single shell unity element with extension applied in the x-direction used to verification of he UMAT

After running the model using the developed and compiled UMAT we would use the output files in the post-processor LS-PrePost (free pre-post processor supplied by LSTC) to view the results. Fig. 3 shows the true strain result of the model run using the developed UMAT. As seen, the final value of the true strain is equal to 1, which matches the theoretical value.

Fig. 3. True strain in the x-direction of the single element model. The final true strain is equal to 1 and the model matches theoretical value.

It should be noted that validation of the developed user material subroutine is extremely important and a required step. If we plot a history of the variable for the tensile model above of true width strain/true thickness strain it should equal 1 (which it is). For an anisotropic model the value should be different depending on the material used.

5. Conclusions

The best solution seems to be using the tested and supported FEM code including only own user procedures to the master program what is full free and possible with LS-DYANA what is described in the article.

Calculations were carried out at the Academic Computer Centre in Gdańsk.

References

[1] https://en.wikipedia.org/wiki/List_of_finite_element_software_packages
[2] http://www.impetus-afea.com/products/features/
[3] Borrvall T. and others, *A User_Defined Element Interface in LS-DYNA*, LS-DYNA Anwenderforum, Ulm 2006.
[4] Erhart T., *An Overview of User-Defined Interfaces in LS-DYNA*, LS_DYNA Forum, Bamberg 2010.
[5] Hallquist J. O., *LS-DYNA. Keyword User's Manual*, Vol I, LS_DYNA R8.0, LSTC, CA, USA 2015.
[6] Adou, M., Lapoujade V., *Examples manual for USER_LOADING option*, 4th European LS-DYNA Conference, Ulm, Germany 2003.