

**Eleventh South African Conference on Computational and Applied
Mechanics**

SACAM 2018

Vanderbijlpark, South Africa, 17-19 September 2018

Open source implementation of a user material subroutine for composite materials

Natasha Botha^{a,b}, Helen M. Inglis^b

^a*Advanced Mathematical Modelling, Modelling and Digital Science, Council of Scientific and Industrial Research (CSIR), Meiring Naude Avenue, Brummeria, Pretoria, 0084, South Africa*

^b*Department of Aeronautical and Mechanical Engineering, Faculty of Engineering, Built Environment and Information Technology, University of Pretoria, Lynnwood Road, Hatfield, Pretoria, 0028, South Africa*

email address : nbotha1@csir.co.za^a, helen.inglis@up.ac.za^b

Abstract

The modeling and simulation of composite materials requires the incorporation of anisotropic material models, as well as appropriate failure criteria or damage models.

This was normally done by developing in-house codes or writing additional programs to accompany existing finite element (FE) software. However, in recent years the ability to model and simulate the complex composite behaviour in commercial FE software packages has matured. That said, even commercial FE software has a limited range of failure criteria, damage models and degradation laws to choose from.

Using open source FE packages, such as Calculix, to analyse complex behaviour of composite materials requires considerably more user interventions. One of the only options is to define multiple materials to build the composite or use a composite shell element, but the software provides no method to initiate a progressive failure and damage analysis which is required to accurately simulate composite material behaviour.

From a research perspective, new contributions in this field are made by either developing new progressive damage models (with various modes of failure and damage) for composite materials or by implementing existing models through a user subroutine in a commercial or open source environment. A user subroutine allows for the customisation of the FE software by easily adding new material models, behaviour laws, boundary conditions, etc. without having to fully understand the complex structure and flow of the underlying FE code.

The purpose of this paper is to focus on the open source implementation of a composite material model by making use of the user subroutine to add material models (UMAT) not already available.

The process for implementing a UMAT in the open source software package Calculix is first established as an example for other users. The first step is therefore to develop a UMAT for a simple material model which is already available in the FE package. The implementation process is then verified and validated by comparing the UMAT to the built-in material definition.

Next, a composite material model is chosen from literature and implemented via a UMAT. The composite material model is verified through a simple single element test and validated with experimental test data.

In short, this paper provides a method for implementing a composite material model, through the use of a UMAT, to model the behaviour of composite structures in an open source FE software package.

Keywords: open source; user material subroutine (umat); composite; CalculiX; finite element analysis

1. Introduction

Composite materials commonly have a low weight to high stiffness ratio which has led to an increased uptake in various industrial applications, such as aerospace, medical and sport for example. As such there is a need to better understand the complex material behaviour and how it influences the overall structural behaviour, including the damage and failure mechanisms. Experimental studies provide a better understanding, but being able to characterise these observations mathematically allows for improved design, optimisation and development of composite structures.

Modelling of composite materials requires a mathematical description of the anisotropic or orthotropic material model to characterise the expected behaviour. Depending on the desired simulation, and the finite element (FE) software capabilities, this characterisation might include additional descriptions to simulate the material degradation and failure. In earlier years this was normally achieved by either developing in-house FE codes or writing additional programs, such as user material subroutines (UMAT), to accompany existing finite element (FE) software [1-2].

There is an increasing trend in the use of open source software, especially in an academic, and R&D environment, as it has no license fees and allows for custom modifications because of the easy access to source code [4]. However, these advantages are also one of the drawbacks as the software is often not as well developed or easy to use as their commercial counterparts [4]. Among all the open source FE solvers available, CalculiX and Code Aster are the most popular for general non-linear FEA as reported in a number of forums. Both solvers have similar capabilities. CalculiX is easier to use when starting out as it has a similar input file structure to Abaqus. The only drawback of Code Aster is that both the input file and documentation is in French.

The composite material modelling capability in CalculiX is limited to using either a built-in linear elastic material model defining the anisotropic or orthotropic material parameters and defining a composite shell element to define the different plies with their associated properties or implementing a user material subroutine (UMAT) with the desired composite material behaviour [5].

A UMAT allows the user to implement the desired material model (of any complexity) when it does not exist in the chosen FE program's material library [6]. This allows the user to focus on the material model development without having to be concerned with the development, maintenance or debugging of the FE software [6]. Abaqus is a popular commercial FE package and as such a lot of literature is available on how to create an Abaqus UMAT [6-7]. The Abaqus UMAT subroutine is popularly used in research [8-9] and there is a lot of information available regarding its development [3,6,7].

Although CalculiX can use the Abaqus UMAT structure there are no tutorials or documentation available on how to implement it. The purpose of this paper is to illustrate the process of developing and implementing an Abaqus UMAT in an open source FE software such as CalculiX. This is done by considering material models with increasing complexity, first a simple linear elastic isotropic material model, followed by a linear elastic orthotropic material model and an Abaqus UMAT composite material model obtained from literature.

2. Abaqus User Material Subroutine (UMAT)

UMATs allow the user to develop a subroutine which contains the constitutive equations which describe the desired material model [6-7]. Once implemented into the FE program, the routine will be called and executed incrementally at each material calculation point of the elements solving and updating the required stresses, strains and dependent state variables [7].

2.1. UMAT Development

An Abaqus UMAT subroutine is normally developed by following FORTRAN 77 and C conventions. Keeping this in mind, the steps required to develop a UMAT are [6]:

1. Define the constitutive equation by defining the stress (STRESS) (Cauchy stress for large strain applications) and the stress rate (in co-rotational framework).

2. Define dependencies on time (TIME, DTIME, PNEWDT), temperature (TEMP, DTEMP) or field variables (PREDEF, DPRED), if necessary.
3. Define internal state variables (STATEV) (explicitly or as a function of rate).
4. Transform the constitutive rate equation into an incremental equation (DROT) using a suitable integration method such as Forward Euler (explicit integration), Backward Euler (implicit integration) or Midpoint Method.
5. Calculate the consistent Jacobian matrix, also known as the stiffness matrix (DDSDDE).

An example of the typical structure for an Abaqus UMAT is shown as Algorithm 1. Note that normally in an Abaqus UMAT the command `INCLUDE 'ABA_PARAM.INC'` is used to call a file which defines the system precision that should be used. For CalculiX, this is simply replaced with the command `IMPLICIT REAL*8 (A-H, O-Z)` [5]. The deformation gradient (DFGRD0, DFGRD1), total strains (STRAN) and strain increments (DSTRAN) are provided as input into the subroutine which then outputs the material Jacobian matrix (DDSDDE) and updated stresses (STRESS) for the constitutive material model.

Algorithm 1: Standard Abaqus UMAT structure adapted for CalculiX (C indicates a comment with the comment syntax in blue)

```

C =====
C STANDARD ABAQUS UMAT SUBROUTINE STRUCTURE
C =====
SUBROUTINE UMAT (STRESS, STATEV, DDSDDE, SSE, SPD, SCD, RPL,
1 DDSDDT, DRPLDE, DRPLDT, STRAN, DSTRAN, TIME, DTIME, TEMP, DTEMP,
2 PREDEF, DPRED, CMNAME, NDI, NSHR, NTENS, NSTATV, PROPS, NPROPS,
3 COORDS, DROT, PNEWDT, CELENT, DFGRD0, DFGRD1, NOEL, NPT, LAYER,
4 KSPT, KSTEP, KINC)
C
C INCLUDE 'ABA_PARAM.INC'
IMPLICIT REAL*8 (A-H, O-Z)
C
CHARACTER*80 CMNAME
C
DIMENSION STRESS (NTENS), STATEV (NSTATV), DDSDDE (NTENS, NTENS),
1 DDSDDT (NTENS), DRPLDE (NTENS), STRAN (NTENS), DSTRAN (NTENS),
2 PREDEF (1), DPRED (1), PROPS (NPROPS), COORDS (3), DROT (3, 3),
3 DFGRD0 (3, 3), DFGRD1 (3, 3)
C
PARAMETER (ONE=1.D0, TWO=2.D0, THREE=3.D0)
C =====
C User Coding AA:
C user coding to define DDSDDE, STRESS, STATEV, SSE, SPD, SCD
C and, if necessary, RPL, DDSDDT, DRPLDE, DRPLDT, PNEWDT
C =====
RETURN
END

```

2.2. Example: Linear Elastic Isotropic

The development of an Abaqus UMAT, discussed in the previous section, is illustrated using a very simple linear elastic isotropic material model. The linear relationship between stress and strain is defined using Hooke's Law [10,11]:

$$\sigma_{ij} = C_{ijkl} \varepsilon_{kl} \quad (1)$$

The coefficients in the fourth order elasticity stiffness tensor are denoted by C_{ijkl} , the Cauchy stress is defined by σ_{ij} and the strain by ε_{kl} . In the case of isotropy the relationship is defined in terms of Lamé's parameters and the Kronecker delta such that the linear elastic constitutive equation becomes [10]:

$$\sigma_{ij} = \frac{E}{1+\nu} \varepsilon_{ij} + \frac{\nu E}{(1+\nu)(1-2\nu)} \varepsilon_{kk} \delta_{ij} = 2\mu \varepsilon_{ij} + \lambda \varepsilon_{kk} \delta_{ij}, \quad (2)$$

where:

$$\mu = \frac{E}{1+\nu}; \lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}$$

Summation on repeated indices is assumed and the Kronecker delta becomes 1 if $i = j$ and 0 if $i \neq j$. Applying symmetric conditions due to isotropy, Eq (2) reduces to six continuity equations written in matrix form [11]:

$$\begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \tau_{12} \\ \tau_{13} \\ \tau_{23} \end{pmatrix} = \begin{bmatrix} \lambda + \mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + \mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + \mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5\mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5\mu \end{bmatrix} \begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{pmatrix} \quad (3)$$

The fourth order elasticity stiffness tensor (DDSDDE) and consequent stress update (STRESS) based on Hooke's Law is programmed into the standard Abaqus UMAT template, shown in Algorithm 1, by replacing the user coding section. This implementation is shown in Algorithm 2 which follows the following steps:

1. The elastic modulus (EMOD) and Poisson's Ratio (ENU) are obtained from the input file through the PROPS command (Lines 10-12). The Lamé's Parameters (EMU, ELAM) are then calculated (Lines 13-17).
2. The stiffness matrix (DDSDDE) is defined using two for loops from an index value of 1 until NDI, which is the number of direct stress components. The off-diagonal and diagonal stiffness coefficients for the principal stresses and strains are first defined in the first for-loop block (Lines 21-27). In the second for-loop block the stiffness coefficients for the shear stresses and strains is defined with a for-loop from NDI+1 until NTENS, the total number of stress components, total direct stress and total shear stress (NDI+NSHR) (Lines 28-30).
3. Lastly, the stresses (STRESS) is updated through Hooke's Law, Eq (1), by multiplying the elasticity tensor (DDSDDE) with the incremental strain (DSTRAN) by means of a for loop block from 1 until NTENS (Lines 31-38).

Algorithm 2: User coding AA to define DDSDDE and STRESS for a linear elastic isotropic material (C indicates a comment with the comment syntax in blue)

```

1 C =====
2 C User Coding AA:
3 C user coding to define DDSDDE, STRESS, STATEV, SSE, SPD, SCD
4 C and, if necessary, RPL, DDSDDT, DRPLDE, DRPLDT, PNEWDT
5 C =====
6 C OBTAIN AND DEFINE MATERIAL PROPERTIES (TO BE DEFINED IN THE INPUT FILE)
7 C EMOD = ELASTIC MODULUS
8 C ENU = POISSON'S RATIO
9 C =====
10 C PROPERTIES FROM INPUT FILE
11     EMOD = PROPS(1)
12     ENU = PROPS(2)
13 C CALCULATE PROPERTIES REQUIRED
14 C EMU - LAME'S FIRST PARAMETER
15     EMU = EMOD / (ONE + ENU)
16 C ELAM - LAME'S SECOND PARAMETER
17     ELAM = (EMOD * ENU) / ((ONE + ENU) * (ONE - TWO * ENU))
18 C =====
19 C DEFINE THE ELASTIC STIFFNESS MATRIX
20 C =====
21 C Entries for the principal stresses
22     DO K1 = 1, NDI
23         DO K2 = 1, NDI
24             DDSDE(K2,K1) = ELAM
25         END DO
26     DDSDE(K1,K1) = ELAM + EMU
27     END DO
28     DO K1 = NDI + 1, NTENS
29         DDSDE(K1,K1) = EMU / TWO
30     END DO
31 C =====
32 C CALCULATE THE STRESS USING HOOKES LAW
33 C =====
34     DO K1 = 1, NTENS
35         DO K2 = 1, NTENS
36             STRESS(K2) = STRESS(K2) + DDSDE(K2,K1) * DSTRAN(K1)
37         END DO
38     END DO
39 C
40     RETURN
41     END

```

2.3. Implementation into CalculiX

The easiest way to implement an Abaqus UMAT subroutine into CalculiX is to modify the CalculiX source files and then recompile the executable. For this study CalculiX version 2.13 [5] is used on an Ubuntu 16.04 platform. The source files for CalculiX are downloadable from the website along with easy to follow instructions on how to install from source code.

There are three core files that require modification in the CalculiX source folder, shown in Fig 1, to successfully implement an Abaqus UMAT for use:

1. Copy the developed UMAT into the CalculiX source folder, for this example the file is called *umat_linelas.f*.
2. Edit the *umat_abaqus.f* subroutine to call the developed subroutine by adding an if-statement which calls the *umat_linelas.f* subroutine.
3. Edit the *Makefile.inc* to include the developed umat by adding a statement *umat_linelas.f* / anywhere in the file, although the file is alphabetically organised for ease of reference.
4. Recompile CalculiX using the *make* command and perform the required actions as per the initial installation documentation to ensure all permissions are given to use the executable.

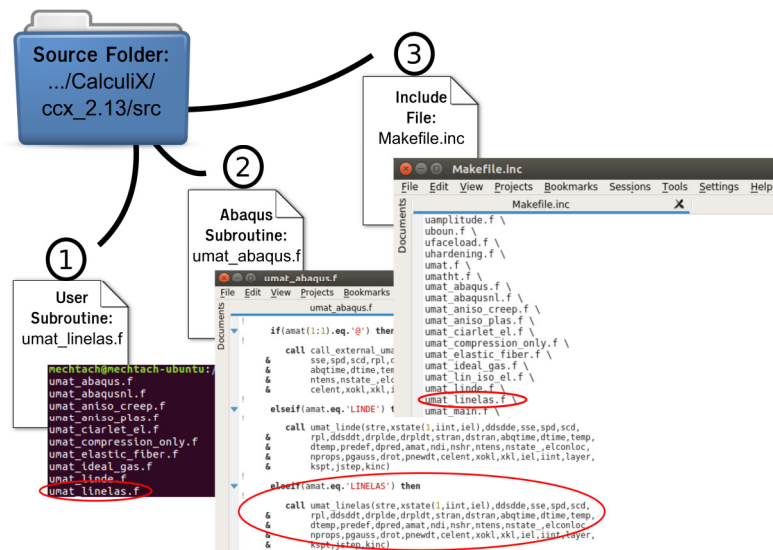


Fig. 1. Implementation of an Abaqus UMAT into CalculiX.

Once implemented into CalculiX, the developed Abaqus UMAT subroutine can be called in CalculiX through the *MATERIAL card with the necessary amendments as shown in Algorithm 3. ABAQUS should always precede the name given to the UMAT subroutine (see Line 6) to ensure the *umat_abaqus.f* file is called which transforms the Abaqus UMAT output (Lagrangian formulation) into the CalculiX format (logarithmic formulation) [5]. In addition, when internal state variables (STATEV) are defined in the UMAT, which is the case for more complex material models, an additional statement *DEPVAR should be included followed by the number of STATEV defined in the UMAT (Lines 9-11).

Algorithm 3: Material definition section in a CalculiX input file for the linear elastic isotropic material example (** indicates a comment with the comment syntax in gray)

```

1  ** =====
2  ** Material definition
3  ** Linear Elastic Isotropic
4  ** Elastic Modulus, Poisson's Ratio
5  ** =====
6  *MATERIAL, NAME=ABAQUSLINELAS
7  *USER MATERIAL, CONSTANTS=2
8  210.0E3, 0.33
9  ** Only add if state variables (STATEV) are defined in UMAT
10 ** *DEPVAR
11 ** 20
12 *SOLID SECTION, ELSET=Eall, MATERIAL=ABAQUSLINELAS
    
```

2.4. Verification: Patch Test

The last step in implementing any user defined material definition is to verify the continuity equation implementation. This is traditionally done by means of a patch test. For this paper a distorted five element patch test with a length of 40mm and height of 30mm is considered using an eight node quadratic plane stress element (CPS8), shown in Fig 2. A constant stress of 50 MPa is applied using the consistent nodal force approach to simulate each of the three considered cases, (1) uniaxial loading in the x direction, (2) uniaxial loading in the y direction and (3) shear loading in the x direction. For a quadratic element the consistent nodal force equates to one sixth of the applied stress at the ends and two thirds at the centre nodes.

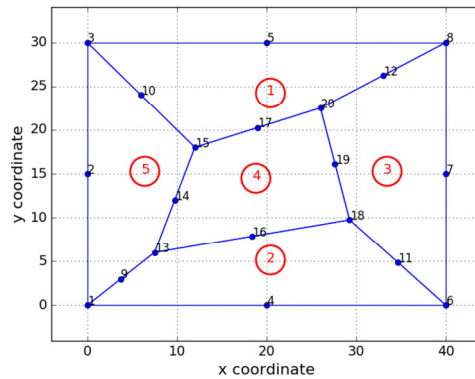


Fig. 2. Five element distorted patch test geometry used for verification of the Abaqus UMAT implementations.

For the linear elastic material an elastic modulus of 100GPa and a Poisson’s ratio of 0.4 are used. The results from the patch tests are shown in Fig 3 where the implemented UMAT is compared to the built-in linear elastic material model. For the linear elastic isotropic material model a strain of 0.05% is expected based on Hooke’s law with an elastic modulus of 100 GPa and a stress of 50 MPa. This is the expected result for the unidirectional cases as shown in Fig 3(a) and (b). In all three cases the built-in model and the developed Abaqus UMAT are equal which indicates that the isotropic continuity equation and Abaqus UMAT is implemented correctly.

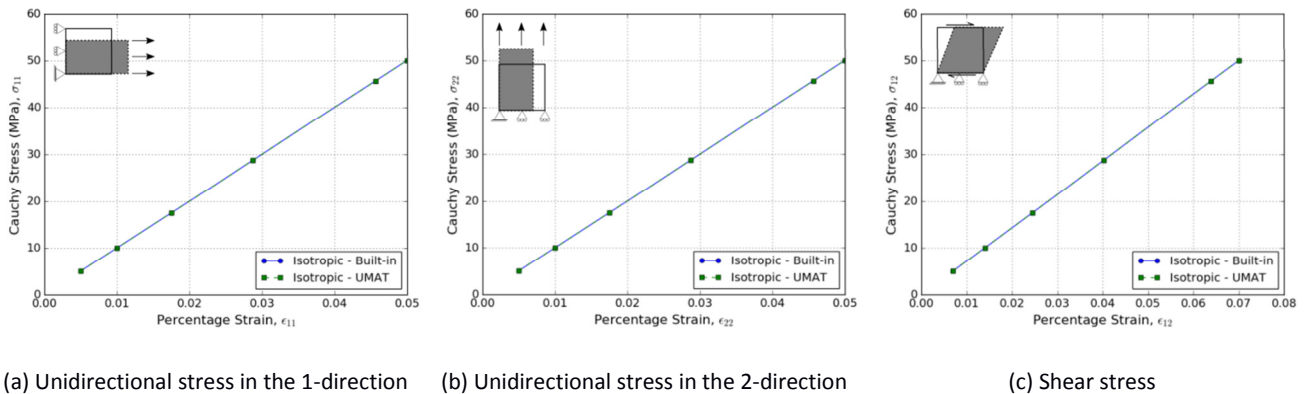


Fig. 3. Five element distorted patch test results comparing the built-in linear elastic isotropic material model with the developed Abaqus UMAT subroutine.

The patch test is said to be passed when a stress equal to the applied stress is obtained and all other stresses are close to zero for all three cases. This is the case and the patch test is therefore passed for the linear elastic isotropic UMAT implementation.

2.5. Benchmark Problem: Three Point Bending

A benchmark test case will also be considered in this paper as a second verification step. The benchmark problem chosen is actually one considered for composite materials and will be applied to the linear elastic material case as well to keep consistency through the paper. The benchmark problem is the NAFEMS R0031 laminated strip under three-point bending [12], illustrated in Fig 4, with a length of 50mm, a width of 10mm and a thickness of 1mm. A constant load of 10N/mm is applied along the centre line of the strip and simple supports are defined at points A and B. An S8R shell element is used with a mesh of 10 elements along the length, 2 elements along the width and 1 element through the thickness as reported in the benchmark problem [12]. A mesh refinement study is therefore not conducted. The material properties for the composite material are provided in Table 1. Results considered for comparison are the strip deflection in the z-direction at point C (u_3), maximum beam stress at point C (σ_{11}) and the interlaminar transverse shear stress at point D (σ_{13}). The average value across all the nodes in the y-direction for points C and D are reported in this study. For all simulations an S8R (eight node quadratic reduced integration formulation) shell element is used with the composite parameter to define the different composite layers.

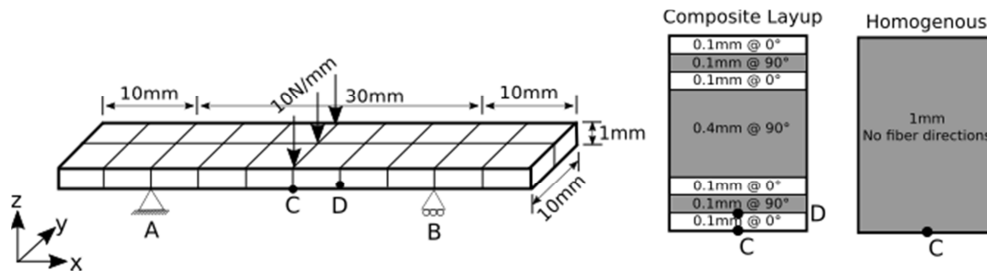


Fig. 4. Geometric and composite material descriptions for the NAFEMS R0031 laminated strip under three point bending benchmark problem [redrawn from 12]

Table 1. Material properties for the composite material used in the NAFEMS R0031 benchmark problem [12].

Property	E_{11}, ν_{12}, G_{12}	E_{22}, ν_{13}, G_{13}	E_{33}, ν_{23}, G_{23}
Elastic Modulus	100 GPa	5 GPa	5 GPa
Poisson's Ratio	0.4	0.3	0.3
Shear Modulus	3 GPa	2 GPa	2 GPa

For the linear elastic simulation the 1mm thickness of the laminate is assumed to be homogenous with an isotropic material with an elastic modulus of 100GPa and a Poisson's ratio of 0.4.

For the linear elastic material model considered we can determine the analytical deflection based on the strip geometry and applied conditions. From static mechanics the analytical deflection (u_3) and maximum beam stress (σ_{11}) are calculated from [13]:

$$u_3 = \frac{PL^3}{48EI}; \sigma_{11} = \frac{My}{I}; I = \frac{1}{12}bh^3 \quad (4)$$

The results from this simulation are shown in Table 2 where the values are averaged across all the nodes of interest. The linear elastic isotropic built-in and UMAT models are in good agreement with a 0.742% deviation from the analytical solution for the deflection and a 0.712% deviation from the analytical for the stress.

Table 2. Results from the three point bending strip for a linear elastic material.

Model	u_3 at C	σ_{11} at C
Analytical	0.675 mm	450.0 MPa
Built-in	0.670 mm (0.742%)	453.205 MPa (0.712%)
UMAT	0.670 mm (0.742%)	453.205 MPa (0.712%)

3. Composite Material Models

The focus of this study is to illustrate the process of implementing a composite material model. For this reason a composite material model was chosen from literature based on the availability and ease of access to an already developed Abaqus UMAT.

As a first step, the process outlined in the previous section is followed to develop an orthotropic material model, also used in the NAFEMS benchmark problem.

3.1. Linear Elastic Orthotropic

The linear elastic isotropic model from the previous section is expanded to an orthotropic model. Based on Hooke's Law as described in Eq (1) the constitutive relationship for an orthotropic material in matrix form is [11]:

$$\begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \tau_{12} \\ \tau_{13} \\ \tau_{23} \end{Bmatrix} = \begin{bmatrix} \frac{1-\nu_{23}\nu_{32}}{E_2E_3\Delta} & \frac{\nu_{21}+\nu_{23}\nu_{31}\nu_{31}+\nu_{21}\nu_{32}}{E_2E_3\Delta} & 0 & 0 & 0 \\ \frac{\nu_{21}+\nu_{23}\nu_{31}}{E_2E_3\Delta} & \frac{1-\nu_{13}\nu_{31}}{E_1E_3\Delta} & \frac{\nu_{32}+\nu_{12}\nu_{31}}{E_1E_3\Delta} & 0 & 0 & 0 \\ \frac{\nu_{31}+\nu_{21}\nu_{32}\nu_{32}+\nu_{12}\nu_{31}}{E_1E_3\Delta} & \frac{\nu_{32}+\nu_{12}\nu_{31}}{E_1E_3\Delta} & \frac{1-\nu_{12}\nu_{21}}{E_1E_2\Delta} & 0 & 0 & 0 \\ 0 & 0 & 0 & G_{12} & 0 & 0 \\ 0 & 0 & 0 & 0 & G_{13} & 0 \\ 0 & 0 & 0 & 0 & 0 & G_{23} \end{bmatrix} \begin{Bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{Bmatrix} \quad (5)$$

The determinant of the stiffness matrix (Δ) is [11]:

$$\Delta = \frac{1-\nu_{12}\nu_{21}-\nu_{23}\nu_{32}-\nu_{13}\nu_{31}-2\nu_{21}\nu_{32}\nu_{13}}{E_1E_2E_3} \quad (6)$$

It should also be noted that in an orthotropic material that $\nu_{ij} \neq \nu_{ji}$ therefore the relationships for the Poisson's ratios not defined are [11]:

$$\nu_{21} = \frac{E_2}{E_1}\nu_{12}; \nu_{31} = \frac{E_3}{E_1}\nu_{13}; \nu_{32} = \frac{E_3}{E_2}\nu_{23} \quad (7)$$

Following the methodology described in Sections 2.1 and 2.2 an Abaqus UMAT is developed for the orthotropic material model and is also shared in the Appendix.

The patch test, described in Section 2.4 is simulated to verify the implementation using the composite material properties provided in Table 1. In addition fibers are orientated in the 0° direction (along the x direction) by means of the *ORIENTATION card in the CalculiX input file. The results from the patch test are shown in Fig 5 where the response for the built-in model and the Abaqus UMAT model correlate well. In addition, the expected strain for the uniaxial case in the 1 direction is 0.05% and for the uniaxial case in the 2 direction is 1%. In both cases the built-in and Abaqus UMAT models obtained the expected results. The Abaqus UMAT passes the patch test indicating the implementation of the orthotropic material model continuity equations is verified. The results from the benchmark problem are provided in Table 4 and discussed in Section 3.3.

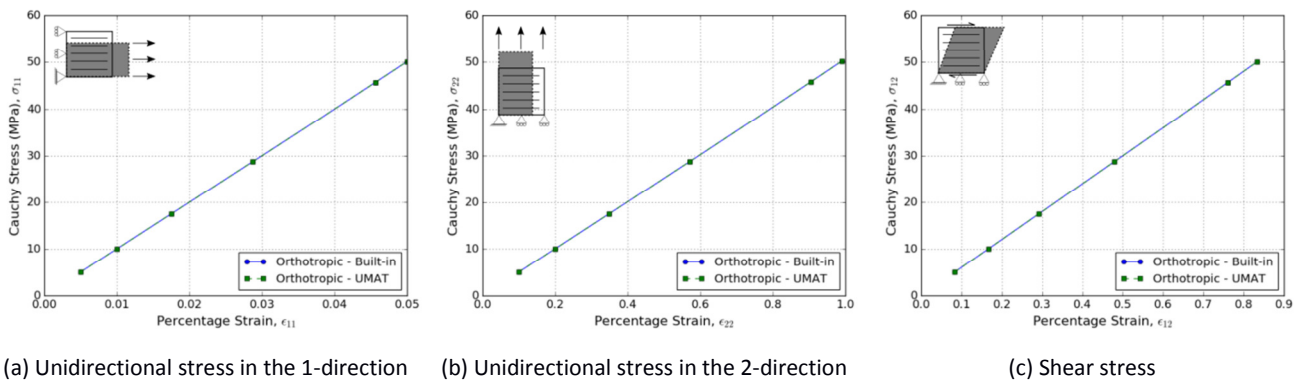


Fig. 5. Five element distorted patch test results comparing the built-in linear elastic orthotropic material model with the developed Abaqus UMAT subroutine.

3.2. Linde Material Model

The Linde composite material model Abaqus UMAT is available for download on the Abaqus Examples Manual [14]. The Linde material model was developed for modelling of fiber metal laminates which includes fiber-reinforced epoxy layers with the model able to predict the fiber/matrix failure and interlaminar delamination [8,14]. For the purposes of this paper though, the different failure mechanisms aren't evaluated.

The initial fiber/epoxy matrix is assumed to be transversely isotropic such that the relations in Eq (5) reduces to $C_{22} = C_{33}$, $C_{12} = C_{13}$, $C_{55} = C_{66}$, $C_{44} = \frac{C_{22}-C_{23}}{2}$, with $E_2 = E_3$, $\nu_{12} = \nu_{13}$, $\nu_{23} = \nu_{32}$, $G_{12} = G_{13}$ and $G_{23} = \frac{E_2}{2(1+\nu_{23})}$ [11]. Applying the progressive damage relationships reduces the fourth order elasticity stiffness tensor (\mathbf{C}) to [8,14]:

$$\mathbf{C} = \begin{bmatrix} (1-d_f)C_{11} & (1-d_f)(1-d_m)C_{12} & (1-d_f)C_{13} & 0 & 0 & 0 \\ (1-d_f)(1-d_m)C_{12} & (1-d_m)C_{22} & (1-d_m)C_{23} & 0 & 0 & 0 \\ (1-d_f)C_{13} & (1-d_m)C_{23} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & (1-d_f)(1-d_m)C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{bmatrix} \quad (8)$$

In the Abaqus UMAT subroutine, the stresses are evaluated according to Hooke's Law as presented in Eq (1). In addition viscous regularisation is used to improve convergence [14]. This material model requires 14 input variables which are stored in PROPS with the material properties provided in Tables 2 and 3. In addition, the two damage variables (d_f and d_m) are stored as internal state variables in STATEV. This allows them to be easily outputted into the results file by simply including the SDV parameter to the output line for the *EL PRINT or *EL FILE cards in the CalculiX input file. Also remember to include the *DEPVAR card (see Section 2.3) indicating the use of 10 internal state variables as defined in the UMAT subroutine. If you forget this card CalculiX will output a core dump error when trying to run the simulation.

Table 3. Additional material properties for the Linde material used in the patch test verification and NAFEMS R0031 benchmark problem.

Property	Without Damage	With Damage [8]
Failure Stress in 11 direction	$\sigma_{11}^{f,t} = 0$ GPa, $\sigma_{11}^{f,c} = 0$ GPa	$\sigma_{11}^{f,t} = 2.5$ GPa, $\sigma_{11}^{f,c} = 2$ GPa
	$\sigma_{22}^{f,t} = 0$ MPa, $\sigma_{22}^{f,c} = 0$ MPa	$\sigma_{22}^{f,t} = 50$ MPa, $\sigma_{22}^{f,c} = 150$ MPa
	$\tau_{12}^{f,s} = 0$ MPa	$\tau_{12}^{f,s} = 50$ MPa
Fracture Energy	$\gamma_f = 0$ J, $\gamma_m = 0$ J	$\gamma_f = 12.5$ J, $\gamma_m = 1$ J
Viscous Regularisation	$\eta = 0.0$	$\eta = 0.001$

The Abaqus UMAT for the Linde material model is implemented into CalculiX as discussed in Section 2 and the patch test simulation is used to verify its implementation with the results presented in Fig 6. As this material model allows for the progressive fiber and matrix damage the simulation was conducted with and without damage. In the case of no damage, the damage related input parameters were set to zero as shown in Table 3.

As the Linde UMAT was not developed in this study but was rather obtained from the Abaqus Examples Manual [14] it is assumed to have been fully verified and validated. The patch test in this case was used to verify the implementation into CalculiX and to test our understanding of the damage behaviour.

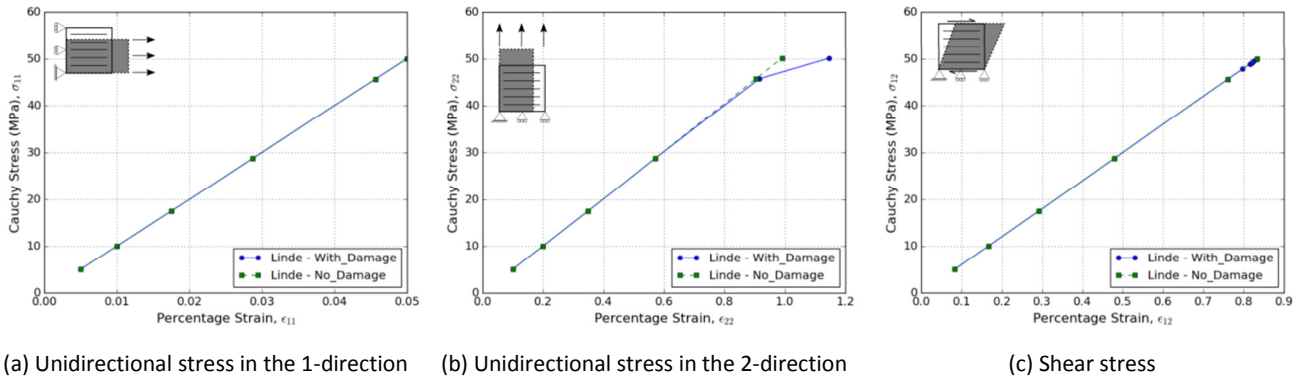


Fig. 6. Five element distorted patch test results comparing the Linde material model with and without damage.

The results for the uniaxial case in the 1 and 2 direction without damage are to be expected with a strain of 0.05% and 1%, respectively, based on Hooke’s Law and the transverse isotropic material assumption as discussed earlier. Once damage is allowed the fibers are not expected to break in the 1 direction as this is also the defined orientation of the fibers, but will rather increase the material stiffness. In the 2 direction it is noted that the material does experience matrix damage as is indicative by the non-linear response from approximately a 0.9% strain. This is expected as the uniform load is applied perpendicular to the fiber orientation which will likely result in the matrix being damaged due to fibers being pulled out.

The implemented Linde UMAT passes the patch test, which indicates that it has been implemented correctly. The results from the benchmark problem are provided in Table 4 and discussed in Section 3.3.

3.3. Benchmark Problem: Three Point Bending

The benchmark problem geometry and set up was discussed in Section 2.5 and shown in Fig 4. The NAFEMS benchmark problem expected results are provided in Table 4 along with the results obtained from the Abaqus simulation [12].

In the Abaqus simulation an orthotropic material model was used and as a first step the CalculiX built-in orthotropic material model and the developed Abaqus UMAT are implemented for comparison followed by the Linde material model (with and without damage). The results from these benchmark simulations are summarised in Table 4 with the contour plots provided in Fig 7.

The built-in and orthotropic UMAT material model implementations compare well with the NAFEMS and Abaqus models with a 0.755% deviation in the obtained displacement, a 0.827% deviation in the bending stress. The obtained interlaminar stress deviates with 51.22% from the NAFEMS results; however this is a deviation on a small baseline (i.e. only a 2.12MPa difference).

The Linde damage model was implemented with and without damage and both yielded the exact same results with the obtained displacement lower than expected (1.415%) and the bending stress slightly higher (0.929%). The interlaminar stress is also closer to that of the NAFEMS result with an error of 32.05%, again only differing by 1.314MPa. It is clear that at this lower applied load of 10N/mm damage is not yet initiated in the Linde model. From the patch test it was noted that damage was only initiated from a load of approximately 45 MPa.

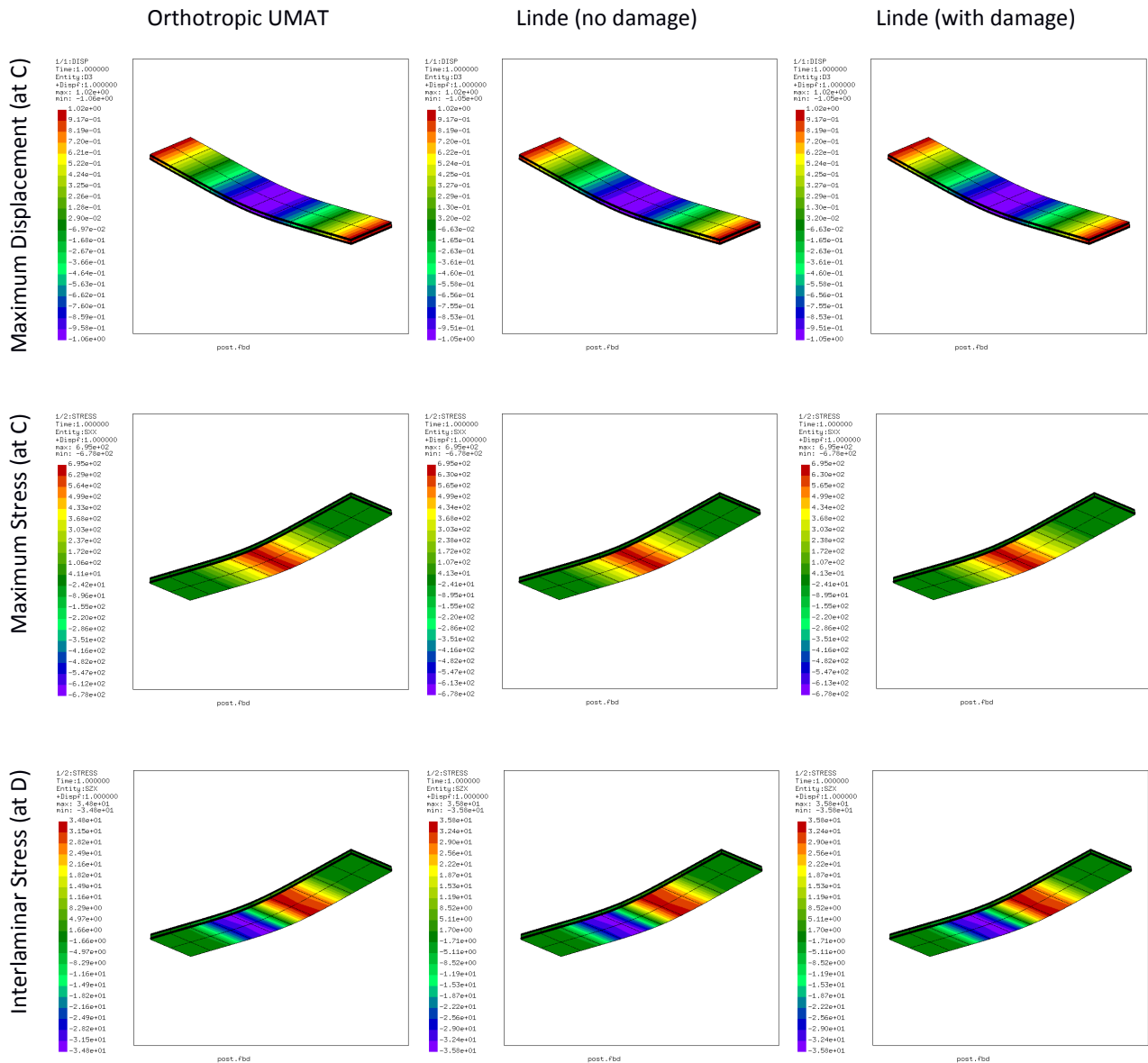


Fig. 7. Finite Element Analysis results for the maximum displacement at point C (row 1), maximum bending stress at point C (row 2) and the interlaminar stress at point D (row 3) for the orthotropic UMAT (column 1), Linde without damage UMAT (column 2) and Linde with damage UMAT (column 3). Note that displacements are exaggerated.

Table 4. Results from the three point bending strip for a linear elastic material.

Model	u_3 at C	σ_{11} at C	σ_{13} at D
NAFEMS	1.06 mm	684 MPa	4.1 MPa
Abaqus – Orthotropic	1.06 mm (0%)	681 MPa (0.439%)	4.08 MPa (0.488%)
CalculiX – Orthotropic (Built-in)	1.052 mm (0.755%)	689.659 MPa (0.827%)	6.2 MPa (51.22%)
CalculiX – Orthotropic (UMAT)	1.052 mm (0.755%)	689.659 MPa (0.827%)	6.2 MPa (51.22%)
CalculiX – Linde (No Damage)	1.045 mm (1.415%)	690.356 MPa (0.929%)	5.414 MPa (32.05%)
CalculiX – Linde (With Damage)	1.045 mm (1.415%)	690.356 MPa (0.929%)	5.414 MPa (32.05%)

4. Conclusion

This paper discussed the development and implementation of an Abaqus UMAT subroutine. The structure of an Abaqus UMAT and the process to interpret a set of continuity equations into the UMAT format was illustrated by means of a simple linear elastic isotropic material model. The implementation of an Abaqus UMAT into CalculiX by means of recompiling was then discussed along with how to call a UMAT in a CalculiX input file. A patch test was used to verify the UMAT development and implementation, followed by a second verification by means of a NAFEMS standard composite material benchmark problem.

After the UMAT development and implementation procedure was established, the linear elastic isotropic material model was expanded to an orthotropic material model and the UMAT is provided as an Appendix. Finally, a composite material model with an already developed Abaqus UMAT was obtained from literature [14]. The Linde composite material model included fiber and matrix damage. Both the orthotropic and Linde UMAT material models passed the patch test. The orthotropic material model results compared well with the NAFEMS benchmark simulation with the exception of the interlaminar stress which deviated by 51.22% (2.12MPa). On the other hand the Linde material model without damage also compared well with an interlaminar stress error of 32.05% (1.314MPa). It was noted though that at the lower applied load of 10N/mm that the Linde model with and without damage provided the same results in both cases. This indicated that damage was not yet initiated which was expected as the patch test indicated that damage was only initiated at approximately 45 MPa.

Future work will evaluate the progressive damage capability of composite material models in an open source environment.

References

- [1] Borrelli R, Di Caprio F, Mercurio U, Romano F (2013). *Assessment of progressive failure analysis capabilities of commercial FE codes*. International Journal of Structural Integrity, **4**(3): 300-320.
- [2] Orifici AC, Herszberg I, Thomson RS (2008). *Review of methodologies for composite material modeling incorporating failure*. Composite Structures, **86**: 194-210.
- [3] Dassault Systemes (2014). *Abaqus 6.14 Analysis User's Guide, Materials Chapter 24: Progressive Damage and Failure*. Retrieved from: <http://abaqus.software.polimi.it/v6.14/books/usb/default.htm>. Accessed on: Nov 8, 2017.
- [4] Powell AC, Arroyave R (2008). *Open Source Software for Materials and Process Modeling*, JOM, May, 32-39.
- [5] Dhondt G (201). *Calculix crunchix user's manual version 2.13*. Retrieved from: <http://www.dhondt.de/ccx 2.13.pdf>. Accessed on Nov 9, 2017.
- [6] n.d. (2007). *Writing user subroutines with abaqus, lecture 6: Writing a umat or vumat*. Retrieved from: <http://imechanica.org/files/Writing2007>. Accessed on Nov 3, 2017.
- [7] Dassault Systemes (2014). *Abaqus 6.14 user subroutines reference guide, Chapter 1.1.41: Umat*. Retrieved from: <http://abaqus.software.polimi.it/v6.14/books/sub/default.htm>. Accessed on: Nov 8, 2017.
- [8] Linde P, Pleitner, J, de Boer H, Carmone C (2004). *Modelling and Simulation of Fibre Metal Laminates*. Abaqus Users' Conference, 421-439.
- [9] Ribeiro ML, Tita V, Vandepitte D (2012). *A new damage model for composite laminates*. Composite Structures, **94**: 635-642.
- [10] Reddy JN (2008). *An Introduction to Continuum Mechanics*, Cambridge University Press. USA
- [11] Herakovich CT (1998). *Mechanics of Fibrous Composites*. John Wiley & Sons, Inc. USA.
- [12] Dassault Systemes (2014). *Abaqus 6.14 Benchmarks Guide, Chapter 4.9.1: R0031(1) Laminated strip under three-point bending*. Retrieved from: <http://abaqus.software.polimi.it/v6.14/books/bmk/default.htm>. Accessed on: Apr 27, 2018.
- [13] Gere JM (2004). *Mechanics of Materials*, Sixth Edition, Boorks/Cole, Thomson Learning, Inc. USA.

[14] Dassault Systemes (2014). *Abaqus 6.14 Examples Guide, Chapter 1.4.6: Failure of blunt notched fiber metal laminates*. Retrieved from: <http://abaqus.software.polimi.it/v6.14/books/examples/default.htm>. Accessed on: Apr 17, 2018.

Appendix A. Linear Elastic Orthotropic Abaqus UMAT (umat_ortho.f)

```

C =====
C ORTHOTROPIC LINEAR ELASTIC UMAT EXAMPLE
C =====
      SUBROUTINE UMAT_ORTHO (STRESS, STATEV, DDSUDE, SSE, SPD, SCD, RPL,
1  DDSDDT, DRPLDE, DRPLDT, STRAN, DSTRAN, TIME, DTIME, TEMP, DTEMP,
2  PREDEF, DPRED, CMNAME, NDI, NSHR, NTENS, NSTATV, PROPS, NPROPS,
3  COORDS, DROT, PNEWDT, CELENT, DFGRD0, DFGRD1, NOEL, NPT, LAYER,
4  KSPT, KSTEP, KINC)
C
C   INCLUDE 'ABA_PARAM.INC'
      IMPLICIT REAL*8 (A-H, O-Z)
C
      CHARACTER*80 CMNAME
C
      DIMENSION STRESS (NTENS), STATEV (NSTATV), DDSUDE (NTENS, NTENS),
1  DDSDDT (NTENS), DRPLDE (NTENS), STRAN (NTENS), DSTRAN (NTENS),
2  PREDEF (1), DPRED (1), PROPS (NPROPS), COORDS (3), DROT (3, 3),
3  DFGRD0 (3, 3), DFGRD1 (3, 3)
C
      PARAMETER (ONE=1.D0, TWO=2.D0, THREE=3.D0)
C =====
C OBTAIN AND DEFINE MATERIAL PROPERTIES
C (TO BE DEFINED IN THE INPUT FILE)
C EMOD11 = ELASTIC MODULUS 11
C EMOD22 = ELASTIC MODULUS 22
C EMOD33 = ELASTIC MODULUS 33
C
C ENU12 = POISSON'S RATIO 12
C ENU13 = POISSON'S RATIO 13
C ENU23 = POISSON'S RATIO 23
C
C EG12 = SHEAR MODULUS 12
C EG13 = SHEAR MODULUS 13
C EG23 = SHEAR MODULUS 23
C =====
C PROPERTIES FROM INPUT FILE
      EMOD11 = PROPS (1)
      EMOD22 = PROPS (2)
      EMOD33 = PROPS (3)
      ENU12 = PROPS (4)
      ENU13 = PROPS (5)
      ENU23 = PROPS (6)
      EG12 = PROPS (7)
      EG13 = PROPS (8)
      EG23 = PROPS (9)
C
C DETERMINE ADDITIONAL PROPERTIES
      ENU21 = ENU12 * (EMOD22/EMOD11)
      ENU31 = ENU13 * (EMOD33/EMOD11)
      ENU32 = ENU23 * (EMOD33/EMOD22)
C =====
C DEFINE THE ELASTIC STIFFNESS MATRIX
C =====
C CALCULATE THE DETERMINATE OF THE STIFFNESS MATRIX
      DELTA = (ONE - (ENU12*ENU21) - (ENU23*ENU32) - (ENU13*ENU31) -
1  (TWO*ENU21*ENU32*ENU13)) / (EMOD11*EMOD22*EMOD33)
C
C Entries for the principal stresses
      DDSUDE (1, 1) = (ONE - (ENU23*ENU32)) / (EMOD22*EMOD33*DELTA)
      DDSUDE (2, 2) = (ONE - (ENU13*ENU31)) / (EMOD11*EMOD33*DELTA)
      DDSUDE (3, 3) = (ONE - (ENU12*ENU21)) / (EMOD11*EMOD22*DELTA)
C
      DDSUDE (1, 2) = (ENU21 + (ENU23*ENU31)) / (EMOD22*EMOD33*DELTA)
      DDSUDE (1, 3) = (ENU31 + (ENU21*ENU32)) / (EMOD22*EMOD33*DELTA)
      DDSUDE (2, 1) = (ENU21 + (ENU23*ENU31)) / (EMOD22*EMOD33*DELTA)
      DDSUDE (2, 3) = (ENU32 + (ENU12*ENU31)) / (EMOD11*EMOD33*DELTA)
      DDSUDE (3, 1) = (ENU31 + (ENU21*ENU32)) / (EMOD22*EMOD33*DELTA)

```

```
      DDSDE(3,2) = (ENU32 + (ENU12*ENU31))/(EMOD11*EMOD33*DELTA)
C
C Entries for the shear stresses
      DDSDE(4,4) = EG12
      DDSDE(5,5) = EG13
      DDSDE(6,6) = EG23
C =====
C CALCULATE THE STRESS USING HOOKES LAW
C =====
      DO K1 = 1, NTENS
        DO K2 = 1, NTENS
          STRESS(K2) = STRESS(K2) + DDSDE(K2,K1) * DSTRAN(K1)
        END DO
      END DO
C
      RETURN
      END
```
