HOW TO WRITE ABAQUS UMAT SUBROUTINE

Two Workshops on an Isotropic Model and Tsai-Hill Failure Criterion

Street Address

City, ST ZIP Code

Phone

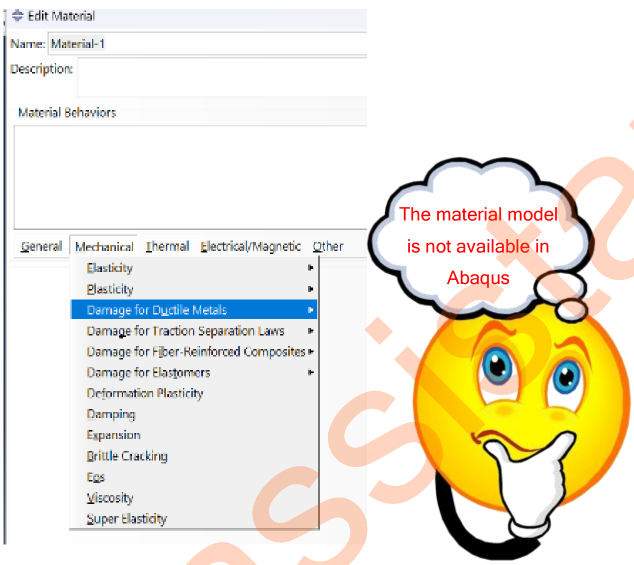
Email

Table of Contents

Writing UMAT Abaqus subroutine

1- What is UMAT and When Do We Need It?

In many cases, the material you are interested in cannot be defined using the models available in Abaqus library.



To address the problem, Abaqus allows you to define the material using the UMAT and VUMAT subroutines.

- ***The main difference between UMAT and VUMAT is that UMAT is used with the standard solver, while VUMAT is used with the explicit solver.***

In the following, we will discuss UMAT, explain the parameters it includes, and demonstrate how to write it. Afterward, we will hold a workshop to show how a UMAT code for a specific problem in Abaqus is written, solved, and its results can be viewed.

2- What does the UMAT do?

- ***UMAT is used to define materials that cannot be implemented in Abaqus, using its standard libraries.***

- ***You need to specify the stress-strain relationship within UMAT, which is done through the Jacobian matrix (D_{ijkl})***

$$\sigma \overset{?}{\propto} \varepsilon$$

- ***In addition, we need to calculate the stress value at each increment (σ_{ij}) inside the UMAT subroutine***

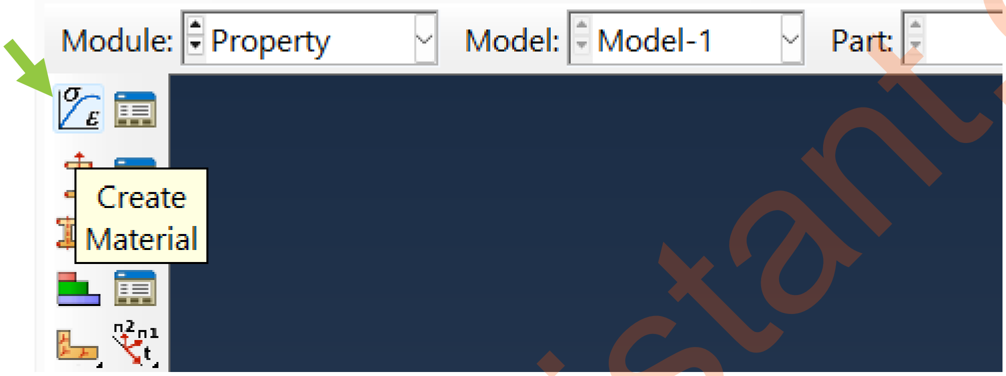
General Hooke's law $\rightarrow \sigma_{ij} = D_{ijkl}\varepsilon_{kl}$

- ***The subroutine sends these values to Abaqus.***

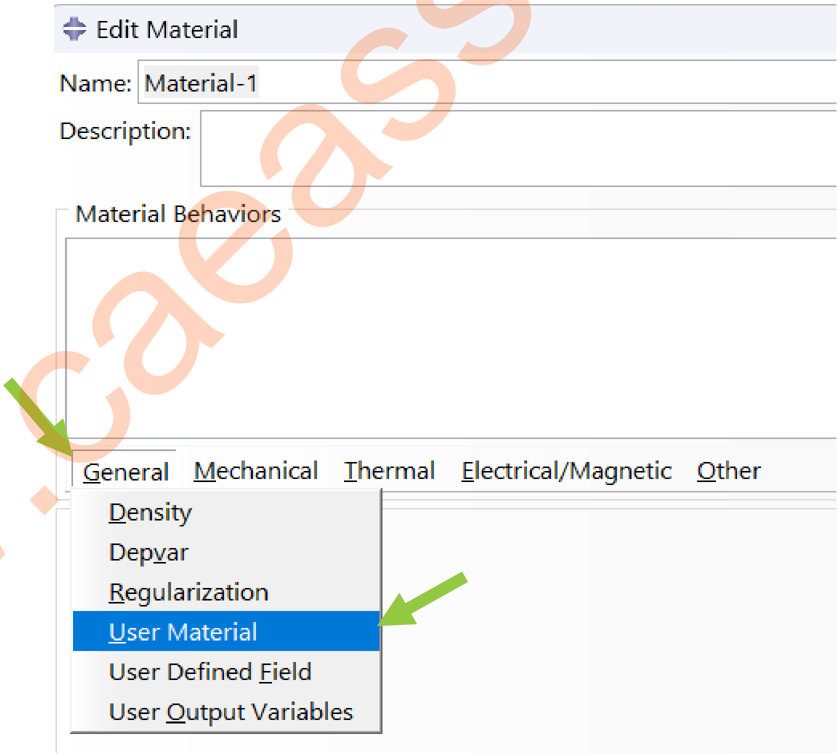
3- How to Implement UMAT in an Abaqus Material Model?

- *The first question that comes to mind is how to incorporate a UMAT subroutine into our Abaqus model*

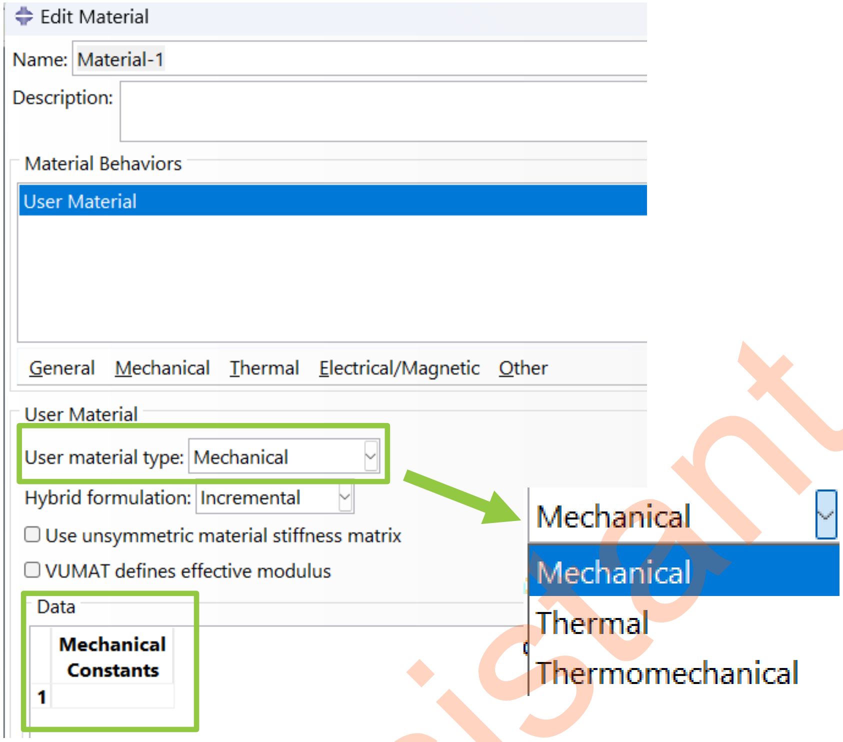
- *To do so, as shown in the image below, go to the Property module in Abaqus. Then, click on the Create Material icon.*



- *Next, as shown in the next figure, click on General and select User Material.*



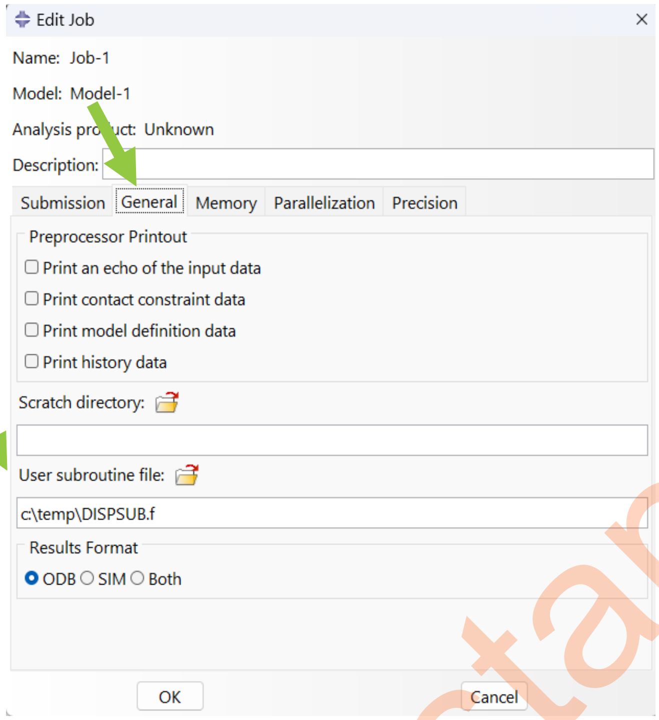
- *When you select this option, a table similar to the one shown in next figure will appear.*



- ***If your parameters in the subroutine are defined in a way that represents an unsymmetric material model, you should check the corresponding check box.***

As shown in figure, you can define a series of mechanical or thermal constants in this table. But what is the purpose of these constants?

- ***Parameters like the elastic modulus, which vary based on the type of composite, can be entered into this table as constants.***
- ***You can then access these values in your UMAT or VUMAT subroutine.***
- ***Notice that, you need to specify the path to the Fortran file containing the UMAT subroutine, in your Abaqus model, for its implementation (according to the next figure).***



3-1- Subroutine's interface

- ***The interface generally refers to the overall structure of a subroutine, in which our code needs to be written.***

- ***The interface consists of a heading that includes the default parameters required for the subroutine. After that, the dimensions of the arrays needed for the subroutine are defined.***

- ***Next is the user-defined coding section, and finally, the subroutine ends by returning the values to Abaqus.***

- ***The interface of the UMAT subroutine is shown in the figure below.***

```
SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
     1 RPL,DDSDDT,DRPLDE,DRPLDT,
     2 STRAN,DSTRAN,TIME,DTIME,TEMP,DTEMP,PREDEF,DPRED,CMNAME,
     3 NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,DROT,PNEWDT,
     4 CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,KSPT,JSTEP,KINC)
C
      INCLUDE 'ABA_PARAM.INC'
C
      CHARACTER*80 CMNAME
      DIMENSION STRESS(NTENS),STATEV(NSTATV),
     1 DDSDDE(NTENS,NTENS),DDSDDT(NTENS),DRPLDE(NTENS),
     2 STRAN(NTENS),DSTRAN(NTENS),TIME(2),PREDEF(1),DPRED(1),
     3 PROPS(NPROPS),COORDS(3),DROT(3,3),DFGRD0(3,3),DFGRD1(3,3),
     4 JSTEP(4)
```

UMAT Header

```
      user coding to define DDSDDE, STRESS, STATEV, SSE, SPD, SCD
      and, if necessary, RPL, DDSDDT, DRPLDE, DRPLDT, PNEWDT
```

User code

```
      RETURN
      END
```

Ending section

3-1- HEADER

You can simply copy the header from the Abaqus documentation and paste it into your file.

```
SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
     1 RPL,DDSDDT,DRPLDE,DRPLDT,
     2 STRAN,DSTRAN,TIME,DTIME,TEMP,DTEMP,PREDEF,DPRED,CMNAME,
     3 NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,DROT,PNEWDT,
     4 CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,KSPT,JSTEP,KINC)
C
      INCLUDE 'ABA_PARAM.INC'
C
      CHARACTER*80 CMNAME
      DIMENSION STRESS(NTENS),STATEV(NSTATV),
     1 DDSDDE(NTENS,NTENS),DDSDDT(NTENS),DRPLDE(NTENS),
     2 STRAN(NTENS),DSTRAN(NTENS),TIME(2),PREDEF(1),DPRED(1),
     3 PROPS(NPROPS),COORDS(3),DROT(3,3),DFGRD0(3,3),DFGRD1(3,3),
     4 JSTEP(4)
```

- ***Some of the header's parameters must be defined in the user code, while others are provided for informational purposes. We will discuss them in following.***

3-1-1- VARIABLES TO BE DEFINED

- ***A list of the variables that you must update in the code, is provided in the table below.***

Variable		Definition
DDSDDE	In all situations	Jacobian matrix
STRESS		Stress tensor at the end of increment
STATEV	In all situations (state variables are used)	Solution-dependent state variables
RPL	fully coupled thermal-stress analysis	Volumetric heat generation per unit time induced by mechanical working
DDSDDT		Variation of stress with respect to temperature
DRPLDE		Variation of Volumetric heat with respect to strain
DRPLDT		Variation of Volumetric heat with respect to temperature

According to the table, the first parameter to define in the UMAT subroutine is DDSDDE. DDSDDE (Jacobian matrix) describes the variation of the stress increment with respect to the strain increment.

$$DDSDDE = \frac{\partial \Delta\sigma}{\partial \Delta\varepsilon}$$

Here, Δσ represents the increment in (Cauchy) stress, while Δε stands for the strains increment.

- ***As shown in the next figure, DDSDDE has NTENS number of columns and NTENS number of rows.***

$$\text{DDSDDE} = \begin{pmatrix} 2\mu+\lambda & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & 2\mu+\lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & 2\mu+\lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{pmatrix}$$

(NTENS,NTENS)

NTENS= NDI+ NSHR

- ***In the figure, NDI represents the Number of direct stress components. NSHR is the Number of shear stress components***

$$\begin{matrix} (NDI) \\ \\ (NSHR) \end{matrix}\left\{\begin{matrix} \Delta\sigma_{11} \\ \Delta\sigma_{22} \\ \Delta\sigma_{33} \\ \Delta\sigma_{23} \\ \Delta\sigma_{31} \\ \Delta\sigma_{12} \end{matrix}\right\}$$

- ***Direct stress components refer to the components of stress that act along the principal axes.***

- ***Engineering shear stress components refer to the components of stress that act tangentially to a surface, causing shearing or distortion.***

- ***Accordingly, DDSDDE for a 3D problem consists of 36 components. But, under typical conditions, the number of independent components can be reduced to as few as 5, as shown in figure below.***

Stress & Strain

$$\begin{pmatrix}\sigma_{xx}\\ \sigma_{yy}\\ \sigma_{zz}\\ \sigma_{xy}\\ \sigma_{yz}\\ \sigma_{xz}\end{pmatrix}=\begin{pmatrix}D_{11} & D_{12} & D_{13} & 0 & 0 & 0\\ . & D_{11} & D_{13} & 0 & 0 & 0\\ . & . & D_{33} & 0 & 0 & 0\\ . & . & . & D_{44} & 0 & 0\\ . & . & . & . & D_{44} & 0\\ . & . & . & . & . & \dfrac{D_{11}-D_{12}}{2}\end{pmatrix}\begin{pmatrix}\varepsilon_{xx}\\ \varepsilon_{yy}\\ \varepsilon_{zz}\\ \gamma_{xy}\\ \gamma_{yz}\\ \gamma_{xz}\end{pmatrix}$$

	36	Component	
Symmetrical Matrix	21	Component	Tri-Clinic Anisotropic Material
+ Symmetry Plane	13	Unique Component	Mono-Clinic Material
+ Two Symmetry Planes	9	Unique Component	Orthotropic Material
Symmetric About an Axis That is Normal to a Plane of Isotropy	5	Unique Component	Transverse Isotropic

- ***The second parameter we want to discuss is STRESS. We can use it both as a variable that provides us with information and also update it.***

- ***STRESS initially stores the stress value at the start of the increment, and we can call it. We also need to update it to store the stress value at the end of the increment.***

STRESS

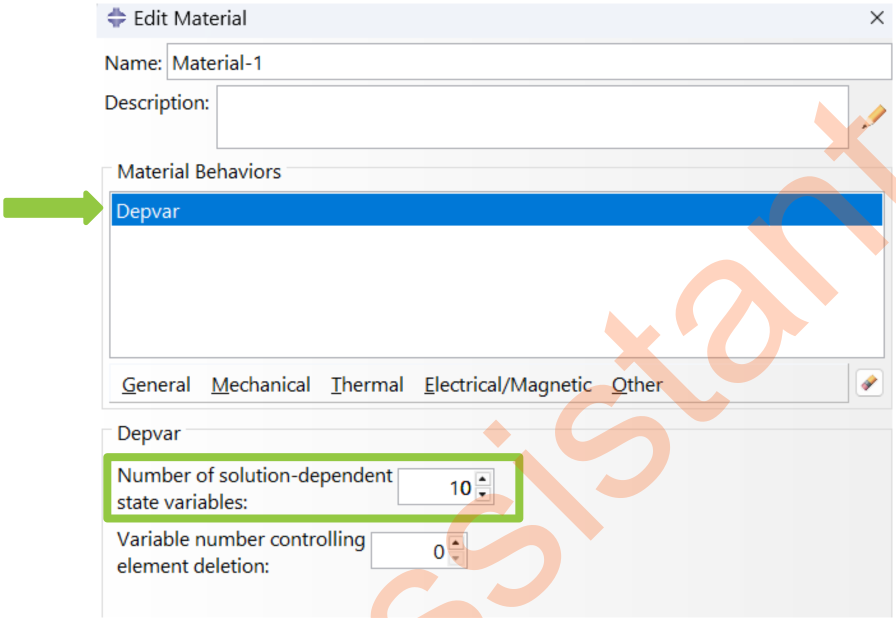
$$\begin{pmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{33}\\ \sigma_{23}\\ \sigma_{31}\\ \sigma_{12}\end{pmatrix}_{\text{(NTENS)}}$$

$$\sigma_i=\sigma_{i-1}+\Delta\sigma_i$$

As shown in the figure, the stress vector has NTENS components.

- ***The third item we want to discuss here is the state variables.***

- ***They are defined by the user and may or may not be required depending on the problem at hand, according to the figure below.***

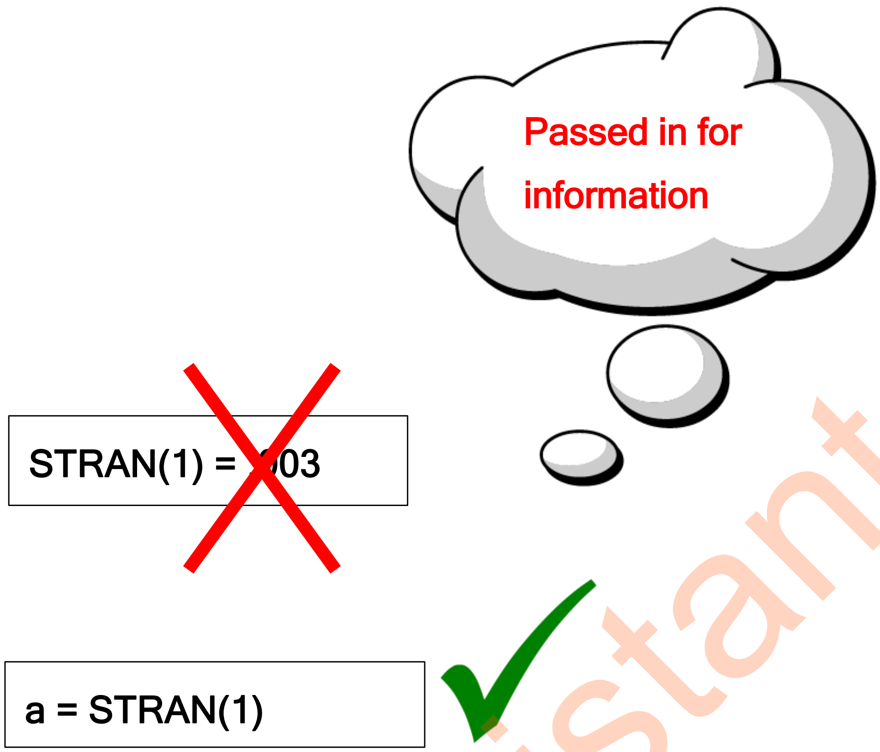


- ***If they are not needed, you can choose not to update them at all.***
- ***Other parameters, such as RPL and DDSDDT, are also presented in the table. But, it is not necessary to define them in all cases. Only for fully-coupled thermal-mechanical analysis, do we need to define appropriate values for them.***

3-1-2- VARIABLES PASSED IN FOR INFORMATION

- ***The material behavior at any given moment may depend on various parameters such as time.***

- ***Accordingly, there are certain variables within the UMAT that provide you with the necessary information.***



When a parameter inside the subroutine is provided only for informational purposes, you cannot assign a value to it, which leads to errors in your code.

4- Workshops

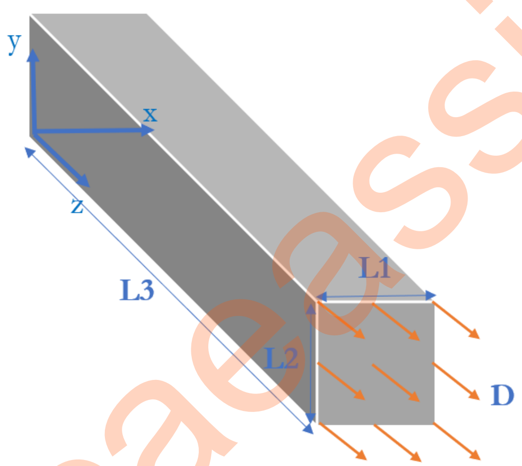
- ***In this section, we will conduct a series of workshops to teach you how to write UMAT code for various problems and use it effectively.***

Workshop 1: Isotropic Isothermal problem

- ***Isotropic refers to a property of materials or systems that exhibit the same characteristics or behavior in all directions.***

- ***Isothermal refers to a process or condition in which the temperature remains constant.***

- ***The problem characteristics are shown in the next figure.***



Property

Elastic:
- E = 210 GPa
- v = 0.3

Loads:
- D = 5 mm

Dimensions:
- L1 = 50 mm
- L2 = 50 mm
- L3= 600 mm

- ***According to the figure, the beam is fixed in xy plan.***

- ***The generalized Hook's low in Matrix form for such an Isotropic problem is shown in the following figure.***

$$\begin{pmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ \varepsilon_{33}\\ 2\varepsilon_{23}\\ 2\varepsilon_{31}\\ 2\varepsilon_{12}\end{pmatrix}=\begin{pmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ \varepsilon_{33}\\ \gamma_{23}\\ \gamma_{31}\\ \gamma_{12}\end{pmatrix}=\frac{1}{E}\begin{pmatrix}1 & -\nu & -\nu & 0 & 0 & 0\\ -\nu & 1 & -\nu & 0 & 0 & 0\\ -\nu & -\nu & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 2(1+\nu) & 0 & 0\\ 0 & 0 & 0 & 0 & 2(1+\nu) & 0\\ 0 & 0 & 0 & 0 & 0 & 2(1+\nu)\end{pmatrix}\begin{pmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{33}\\ \sigma_{23}\\ \sigma_{31}\\ \sigma_{12}\end{pmatrix}$$

- ***In which $\gamma_{ij=2\varepsilon_{ij}}$ is defined as the shear engineering strain.***

- ***Conversely, the stress in the material can be calculated from the strain using the following equation.***

$$\begin{pmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{33}\\ \sigma_{23}\\ \sigma_{31}\\ \sigma_{12}\end{pmatrix}=\frac{E}{(1+\nu)(1-2\nu)}\begin{pmatrix}1-\nu & \nu & \nu & 0 & 0 & 0\\ \nu & 1-\nu & \nu & 0 & 0 & 0\\ \nu & \nu & 1-\nu & 0 & 0 & 0\\ 0 & 0 & 0 & (1-2\nu)/2 & 0 & 0\\ 0 & 0 & 0 & 0 & (1-2\nu)/2 & 0\\ 0 & 0 & 0 & 0 & 0 & (1-2\nu)/2\end{pmatrix}\begin{pmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ \varepsilon_{33}\\ 2\varepsilon_{23}\\ 2\varepsilon_{31}\\ 2\varepsilon_{12}\end{pmatrix}$$

- ***The above equation can be rewritten in the following form using the Lamé constants.***

$$\begin{pmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{33}\\ \sigma_{23}\\ \sigma_{31}\\ \sigma_{12}\end{pmatrix}=\begin{pmatrix}2\mu+\lambda & \lambda & \lambda & 0 & 0 & 0\\ \lambda & 2\mu+\lambda & \lambda & 0 & 0 & 0\\ \lambda & \lambda & 2\mu+\lambda & 0 & 0 & 0\\ 0 & 0 & 0 & \mu & 0 & 0\\ 0 & 0 & 0 & 0 & \mu & 0\\ 0 & 0 & 0 & 0 & 0 & \mu\end{pmatrix}\begin{pmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ \varepsilon_{33}\\ 2\varepsilon_{23}\\ 2\varepsilon_{31}\\ 2\varepsilon_{12}\end{pmatrix}$$

In which λ is the Lamé constant, and μ is the shear modulus, both defined as follows.

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

- ***Finally, the stress increment in terms of the strain increment is expressed by the following equation.***

$$\Delta\sigma = DDSDDE \;\; \Delta\varepsilon^{e} = \begin{pmatrix} 2\mu+\lambda & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & 2\mu+\lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & 2\mu+\lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{pmatrix} \begin{pmatrix} \Delta\varepsilon_{11} \\ \Delta\varepsilon_{22} \\ \Delta\varepsilon_{33} \\ \Delta\gamma_{23} \\ \Delta\gamma_{31} \\ \Delta\gamma_{12} \end{pmatrix}$$

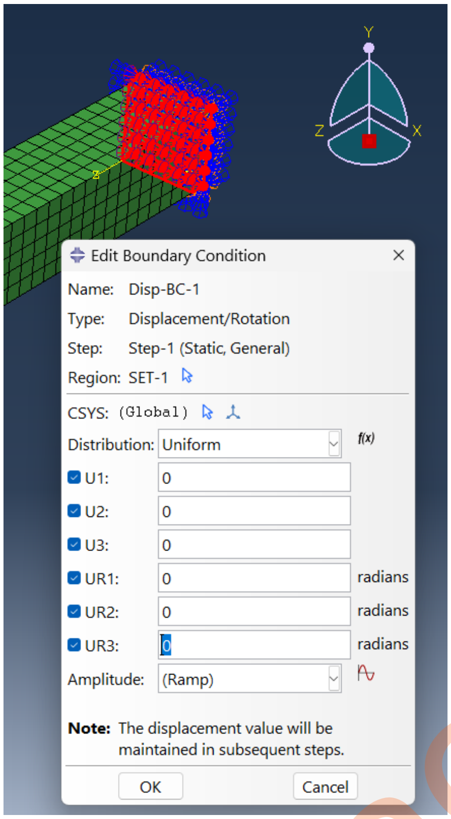
In which DDSDDE is the Jacobian matrix.

- ***Developing the UMAT code for this problem is the focus of this workshop.***

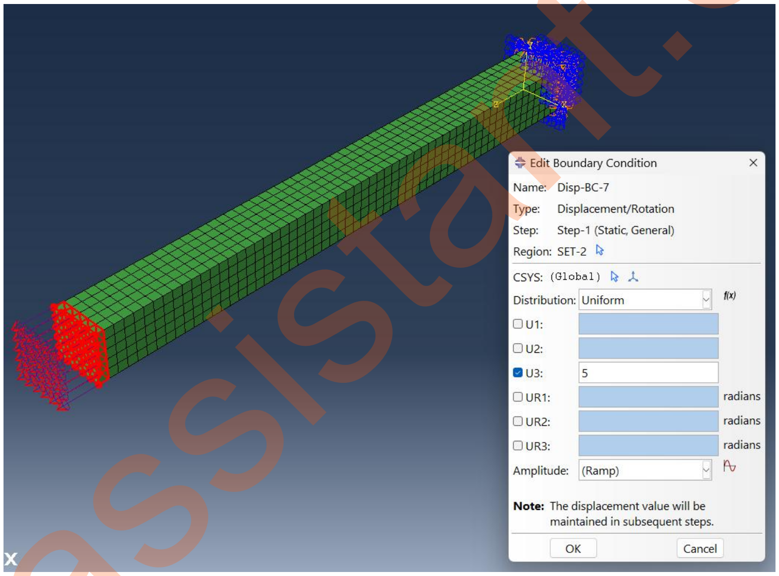
- ***Before we start writing the UMAT code, let's take a moment to review the modeling process in Abaqus together. This is illustrated in the next figure.***
  1. ***According to the figure, first, you need to create 3D deformable part.***
  2. ***Then you need to go to the property module and define the material properties***
  3. ***Then you need to define a section and assign it to the specimen.***
  4. ***Next, you must create an instance of the model in assembly module***
  5. ***Afterward, create a static general step***
  6. ***Now, you can apply boundary conditions, including the fixed end and the applied displacement***
  7. ***Then define the elements***
  8. ***Next, create a job in the Job module***
  9. ***Finally, add the fortran subroutine file address to edit job general menu.***

(1)
Create Part
Name: Part-2
Modeling Space
3D 2D Planar Axisymmetric
Type
Deformable
Discrete rigid
Analytical rigid
Eulerian
Options
None available
Base Feature
Shape
Solid
Shell
Wire
Point
Type
Extrusion
Revolution
Sweep
Approximate size: 200
Continue...
Cancel
Material
(2)
Module: Property
Create Material
Edit Material
Name: STEEL
Description:
Material Behaviors
User Material
General Mechanical Thermal Electrical/Magnetic Other
Density
Depvar
Regularization
User Material
User Defined Field
User Output Variables
Data
Mechanical Constants
1 210000
2 0.3
(3)
Assign Section
Create Section
Name: Section-2
Category
Solid
Shell
Beam
Other
Type
Homogeneous
Generalized plane strain
Eulerian
Composite
Continue...
Cancel
Section
Edit Section
Name: Section-2
Type: Solid, Homogeneous
Material: STEEL
Plane stress/strain thickness: 1
Assembly
(4)
Module: Assembly
Create Instance
Name:
Auto Specify:
Create instances from:
Parts Models
Parts
BEAM
Instance Type
A meshed part has been selected, so the instance type will be Dependent.
Auto-offset from other instances
OK
Apply
Cancel
Step
(5)
Module: Step
Model: UMATmodel
Step: Initial
Create Step
Edit Step
Name: Step-1
Type: Static, General
Basic Incrementation Other
Description:
Time period: 1
Nlgeom: Off On
(This setting controls the inclusion of nonlinear effects of large displacements and affects subsequent steps.)
Automatic stabilization: None
Include adiabatic heating effects
Boundary
(6)
Module: Load
Model: UMATmodel
Step: Step-1
Create Boundary Condition
Create Boundary Condition
Name: BC-8
Step: Step-1
Procedure: Static, General
Category
Mechanical
Electrical/Magnetic
Other
Types for Selected Step
Symmetry/Antisymmetry/Encastre
Displacement/Rotation
Velocity/Angular velocity
Connector displacement
Connector velocity
Element generation
(7)
Module: Mesh
Object: Assembly Part: BEAM
Mesh Part
Job definition
(8)
Create Job
Name: Job-1
Source: Model
Model-1
UMATmodel
Continue...
Cancel
Importing the subroutine
(9)
Edit Job
Name: Job-1
Model: UMATmodel
Analysis product: Abaqus/Standard
Description:
General Memory Parallelization Precision
Scratch directory:
User subroutine file:
C:\Users\poori\Desktop\umat.f
Results Format
ODB SIM Both
OK
Cancel

- ***Notice that, for the boundary conditions, you need to define two displacement-type boundary conditions at the start and end of the beam, with the details shown in the figure below.***



Fixed end



Applied displacement

Let's see how the UMAT code is written for such a problem.

- ***As shown in the next figure, we first copied the subroutine header from the Abaqus documentation and pasted it into the Fortran code.***

- ***Then, we defined several parameters in the real(8) format, including the modulus of elasticity, Poisson's ratio, a variation of the bulk modulus, a coefficient for the shear modulus and its value, as well as the values for one, two, and three.***

```
      SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
     1 RPL,DDSDDT,DRPLDE,DRPLDT,
     2 STRAN,DSTRAN,TIME,DTIME,TEMP,DTEMP,PREDEF,DPRED,CMNAME,
     3 NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,DROT,PNEWDT,
     4 CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,KSPT,JSTEP,KINC)
C
      INCLUDE 'ABA_PARAM.INC'
C
      CHARACTER*80 CMNAME
      DIMENSION STRESS(NTENS),STATEV(NSTATV),
     1 DDSDDE(NTENS,NTENS),DDSDDT(NTENS),DRPLDE(NTENS),
     2 STRAN(NTENS),DSTRAN(NTENS),TIME(2),PREDEF(1),DPRED(1),
     3 PROPS(NPROPS),COORDS(3),DROT(3,3),DFGRD0(3,3),DFGRD1(3,3),
     4 JSTEP(4)

      REAL*8 EMOD,ENU,EBULK3,EG2,EG,EG3,ELAM,ONE,TWO,THREE
      ONE=1.0
      TWO=2.0
      THREE=3.0
C
C UMAT FOR ISOTROPIC ELASTICITY
```

Header

Parameters

- ***In the next step, we need to ensure that this model is only applicable to 3D problems. Therefore, we first check this in our code.***

- ***If the number of direct stress components is not 3, the Abaqus file is incompatible with this code, and an error should be raised, causing the program to exit. This process is illustrated in the figure below.***

```
      IF (NDI.NE.3) THEN
      WRITE (7, *) 'THIS UMAT MAY ONLY BE USED FOR ELEMENTS
     1 WITH THREE DIRECT STRESS COMPONENTS'
      CALL XIT
      ENDIF
```

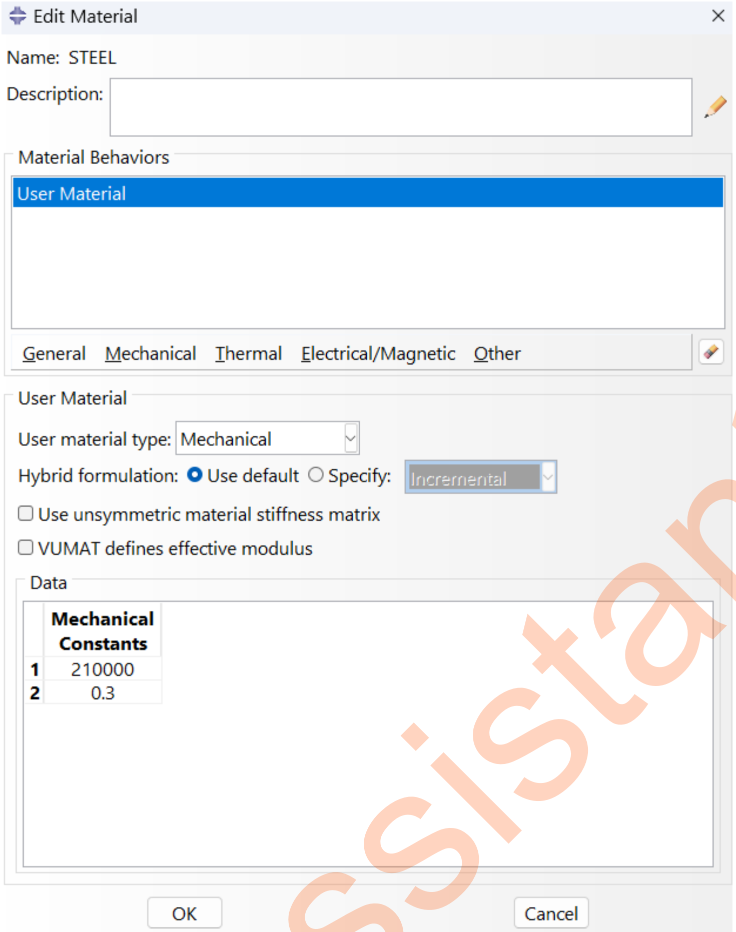
- ***Note that the command "write(7,*)" in the figure above means that Abaqus should print the warning in the message file.***

- ***The number 7 is used for printing in the message file. If you want it to be printed in a different file, you can change the number to your desired number according to the table below.***

Code	Unit Number	Description
ABAQUS/Standard	1	Internal database
	2	Solver file
	6	Printed output (`.dat`) file (You can write output to this file.)
	7	Message (`.msg`) file (You can write output to this file.)
	8	Results (`.fil`) file
	10	Internal database
	12	Restart (`.res`) file
	19–30	Internal databases (scratch files). Unit numbers 21 and 22 are always written to disk.
	73	Text file containing meshed beam cross-section properties (`.bsp`)

- ***In the next step, we define the physical constants for determining the stiffness matrix.***

- ***As shown below, we stored the values of the Young's modulus and Poisson's ratio as mechanical constants. Now, whenever we need these values, we can call them using props(1) and props(2), respectively.***



- ***In the next figure, EMOD is the Young's modulus, and ENU is the Poisson's ratio.***
- ***EBULK3 is the bulk modulus multiplied by 3.***
- ***EG2 is the shear modulus multiplied by 2.***
- ***EG is the shear modulus, and EG3 is the shear modulus multiplied by 3.***
- ***ELAM is the Lamé constant.***

```
C
C ELASTIC PROPERTIES
        EMOD=PROPS(1)
        ENU=PROPS(2)
        EBULK3=EMOD/(ONE-TWO*ENU)
        EG2=EMOD/(ONE+ENU)
        EG=EG2/TWO
        EG3=THREE*EG
        ELAM=(EBULK3-EG2)/THREE
        print*,'ELAM',ELAM
```

$$K=\frac{E}{3(1-2\nu)}$$

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

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

- ***Now it's time to calculate the Jacobian matrix.***

- ***To define the Jacobian matrix, we first write a loop to compute the components related to the direct stress, as shown in the figure below.***

- ***In the figure, the components on the diagonal are shown in blue, while the off-diagonal components are shown in orange.***

```
C
C ELASTIC STIFFNESS
C
        DO K1=1,NDI
        DO K2=1,NDI
        DDSDDE(K2,K1)=ELAM
        END DO
        DDSDDE(K1,K1)=EG2+ELAM
        END DO
        DO K1=NDI+1, NTENS
        DDSDDE(K1,K1)=EG
        END DO
```

$$\begin{pmatrix} 2\mu+\lambda & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & 2\mu+\lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & 2\mu+\lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{pmatrix}$$

After that, we defined another loop to calculate the stiffness components related to the shear stress, which are shown in red in the figure.

- ***Now, it's time to calculate the stress. Using the formula below, we calculated the stress components within a for loop.***

```
C
C CALCULATE STRESS
C
      DO K1=1, NTENS
      DO K2=1, NTENS
      STRESS(K2)=STRESS(K2)+DDSDDE(K2,K1)*DSTRAN(K1)
      END DO
      END DO
C
     RETURN
     END
```

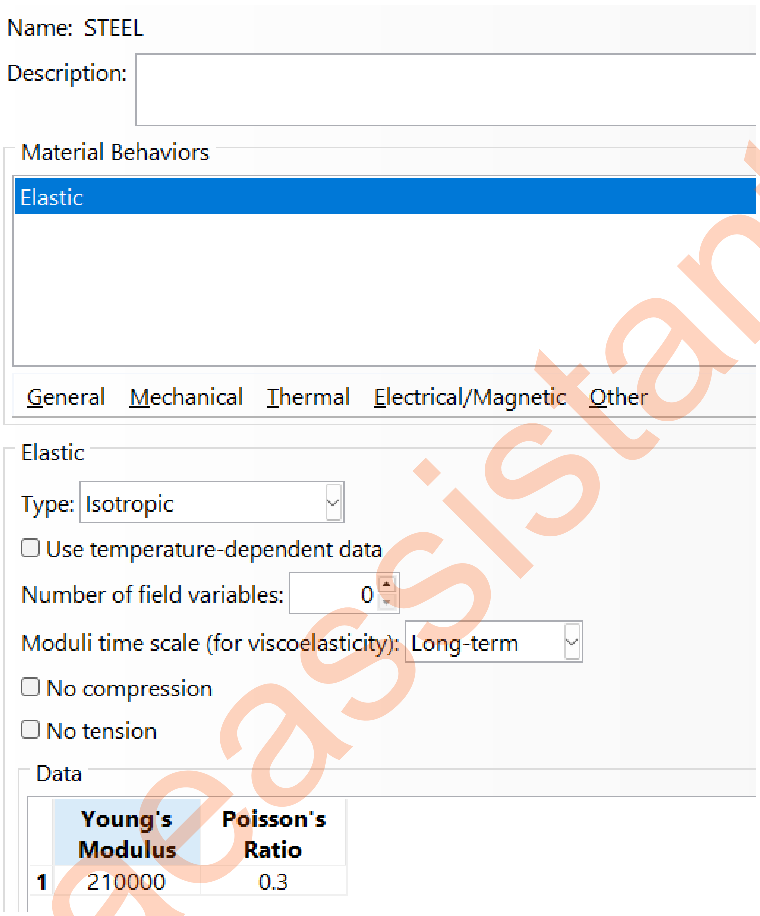
$$\sigma_i=\sum C_{ij}\varepsilon_j$$

- ***Finally, as shown in the figure below, the subroutine concludes and returns the values to Abaqus.***

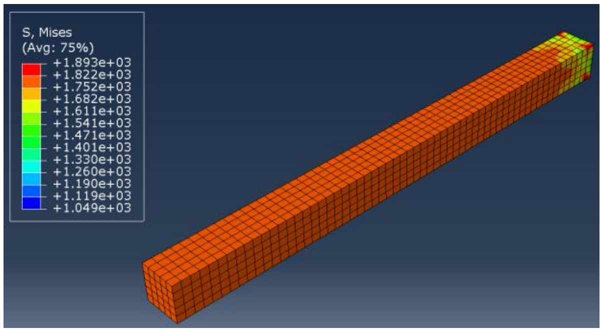
```
C
      RETURN
      END
```

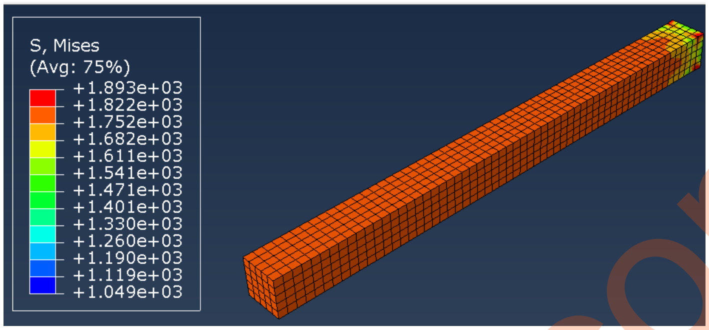
RESULTS

In this section, we solved the problem once using the UMAT subroutine and once using Abaqus's built-in material model, with the same properties. The process for defining the materials in Abaqus is illustrated in the figure below.



We plotted the Mises stress in the job output. As shown in the figure below, the pattern obtained using Abaqus's material model and the UMAT subroutine is identical, verifying the accuracy of the UMAT code.



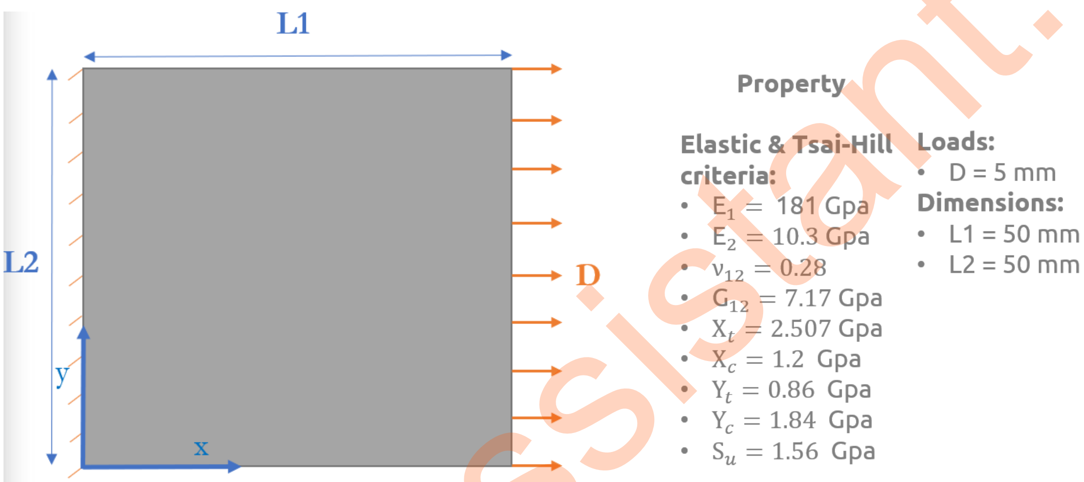


We have provided the Abaqus modeling files and Fortran code for this example at this link**. You can check it, while a step-by-step learning video is also available through the link for free.**

Workshop 2: Orthotropic isothermal elasticity for a plane stress problem | Implementation of the Tsai-Hill failure criteria

In our second workshop, we will use the UMAT subroutine for a 2D plane stress problem, where the Tsai-Hill criterion will also be introduced and analyzed.

- ***The problem's characteristics, including material properties, dimensions, and boundary conditions, are illustrated in the next below.***



- ***According to the figure, the value of the displacement of the lamina is equal to D and it is fixed in Y axis.***

Before we dive into coding and modeling the problem, let's first review the necessary formulas, and then we'll explain how to implement them in the subroutine.

- ***An orthotropic material is a type of material that has different properties in three mutually perpendicular directions.***

- ***For an orthotropic elasticity problem in the 2D plane stress state, the relationship between stress and strain is expressed as follows.***

$$\begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{pmatrix} = \begin{pmatrix} \frac{1}{E_1} & \frac{-\nu_{12}}{E_1} & 0 \\ \frac{-\nu_{12}}{E_1} & \frac{1}{E_2} & 0 \\ 0 & 0 & \frac{1}{G_{12}} \end{pmatrix} \begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \tau_{12} \end{pmatrix}$$

- ***Similarly, the inverse of the above relationship, which enables us to calculate stress based on strain, is written as follows.***

$$\begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \tau_{12} \end{pmatrix} = \begin{pmatrix} D_{11} & D_{12} & 0 \\ V_{21} * D_{11} & D_{22} & 0 \\ 0 & 0 & D_{66} \end{pmatrix} \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{pmatrix}$$

Where:

$$\nu_{21} = \frac{E_2}{E_1} \nu_{12}$$

- ***Emphasizing that the plane stress formulation is considered, the governing relationship between the strain increment and stress is expressed as follows.***

$$\Delta\sigma = DDSDDE \;\; \Delta\varepsilon^e = \begin{pmatrix} \frac{E_1}{1-\nu_{12}\nu_{21}} & \frac{\nu_{12}E_2}{1-\nu_{12}\nu_{21}} & 0 \\ \frac{\nu_{12}E_2}{1-\nu_{12}\nu_{21}} & \frac{E_2}{1-\nu_{12}\nu_{21}} & 0 \\ 0 & 0 & G_{12} \end{pmatrix} \Delta\varepsilon^e$$

In which DDSDDE is the Jacobian (Stiffness) matrix.

- ***The general relationship for the Tsai-Hill criterion is expressed as follows. In other words, if the value on the left side is greater than or equal to one, failure occurs.***

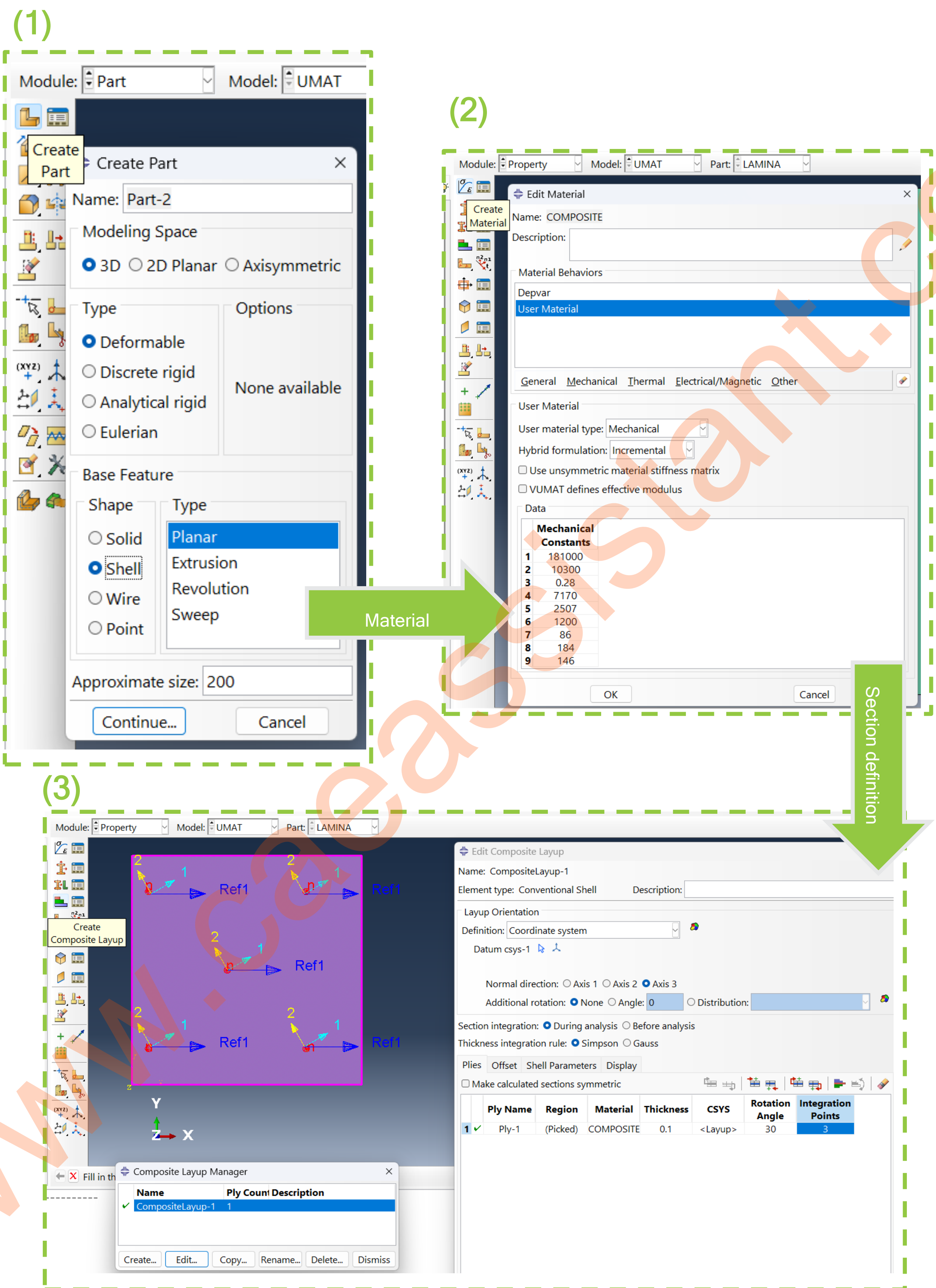
$$F = \frac{\sigma_{11}^2}{X^2} - \frac{\sigma_{11}\sigma_{22}}{X^2} + \frac{\sigma_{22}^2}{Y^2} + \frac{\sigma_{12}^2}{S^2} < 1.0$$

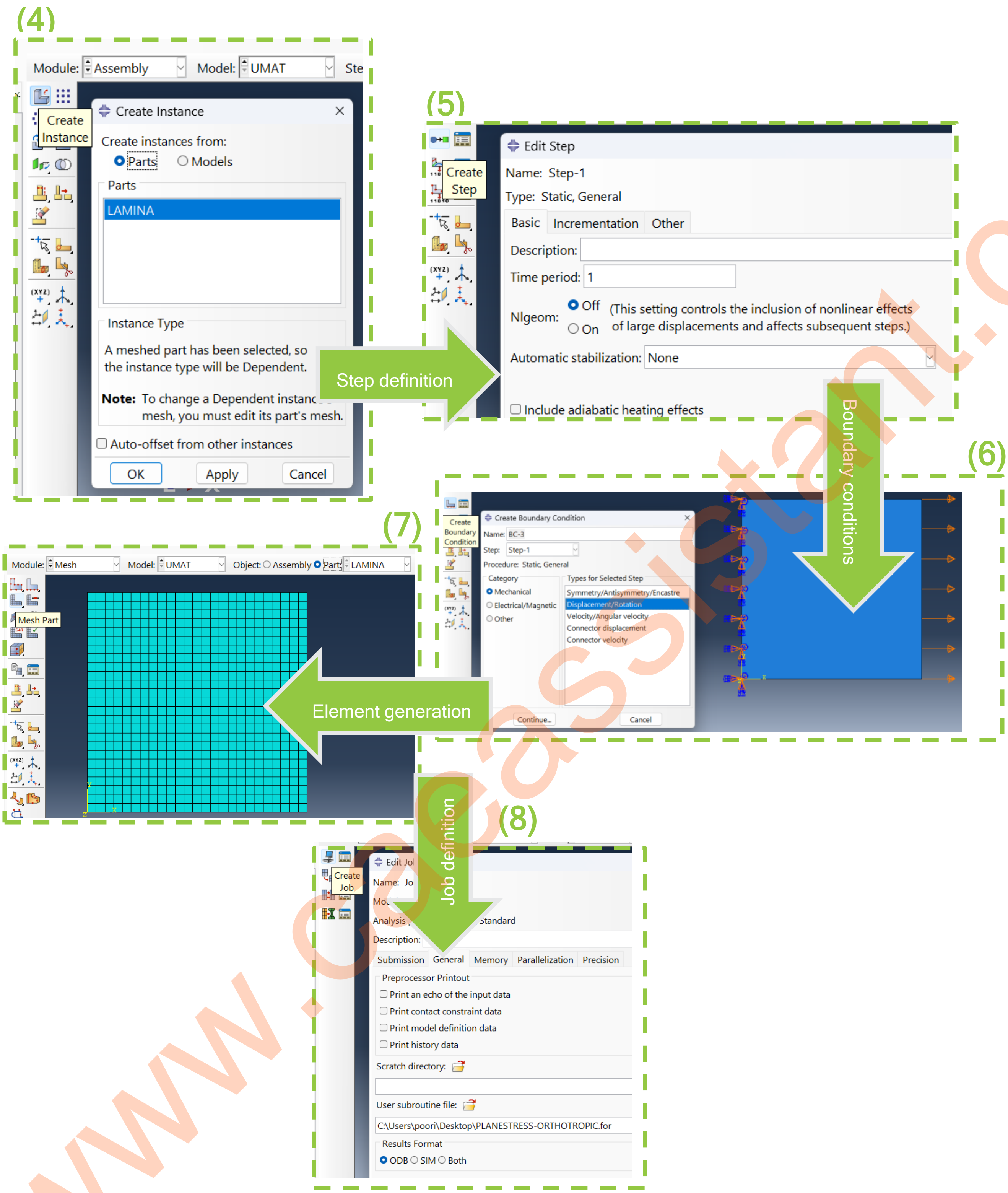
Where:

$$\text{if } \sigma_{11} > 0, X = X_t; otherwise, X = X_c$$

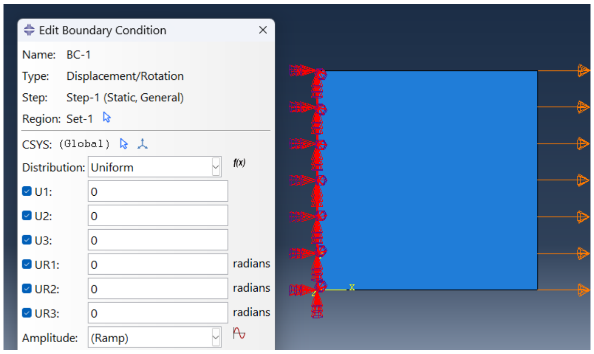
$$\text{if } \sigma_{22} > 0, Y = Y_t; otherwise, Y = Y_c$$

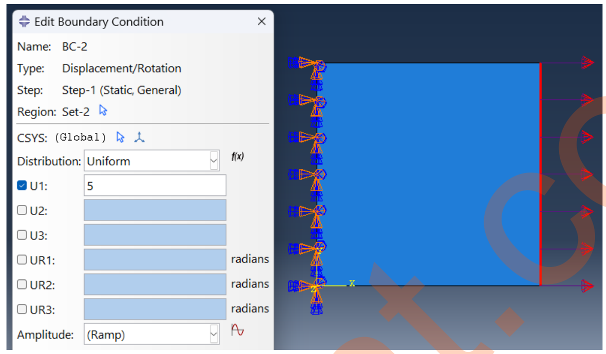
- ***X is the allowable strength of the ply in the longitudinal direction.***
- ***Y is the allowable strength of the ply in the transverse direction.***
- ***S is the in-plane shear allowable strength of the ply between the longitudinal and transverse directions.***
- ***Before we start writing the UMAT code, let's take a moment to review the modeling process in Abaqus. This is illustrated in the next figures.***
  1- ***According to the figure, you need to create 3D deformable shell part.***
  2- ***Then you need to go to the property module and define the material properties***
  3- ***Then you need to define a section and assign it to the specimen.***
  4- ***Next, you must create an instance of the model in assembly module***
  5- ***Afterward, create a static general step***
  6- ***Now, you can apply boundary conditions***
  7- ***Then define the elements***
  8- ***Next, create a job in the Job module***
  9- ***Finally, add the fortran subroutine file address to edit job general menu.***


(1)
Module: Part
Model: UMAT
Create Part
Create Part
Name: Part-2
Modeling Space
3D
2D Planar
Axisymmetric
Type
Deformable
Discrete rigid
Analytical rigid
Eulerian
Options
None available
Base Feature
Shape
Solid
Shell
Wire
Point
Type
Planar
Extrusion
Revolution
Sweep
Approximate size: 200
Continue...
Cancel
Material
(2)
Module: Property
Model: UMAT
Part: LAMINA
Create Material
Edit Material
Name: COMPOSITE
Description:
Material Behaviors
Depvar
User Material
General
Mechanical
Thermal
Electrical/Magnetic
Other
User Material
User material type: Mechanical
Hybrid formulation: Incremental
Use unsymmetric material stiffness matrix
VUMAT defines effective modulus
Data
Mechanical Constants
1 181000
2 10300
3 0.28
4 7170
5 2507
6 1200
7 86
8 184
9 146
OK
Cancel
Section definition
(3)
Module: Property
Model: UMAT
Part: LAMINA
Create Composite Layup
Ref1
Y
X
Composite Layup Manager
Name
Ply Count
Description
CompositeLayup-1
1
Create...
Edit...
Copy...
Rename...
Delete...
Dismiss
Edit Composite Layup
Name: CompositeLayup-1
Element type: Conventional Shell
Description:
Layup Orientation
Definition: Coordinate system
Datum csys-1
Normal direction: Axis 1 Axis 2 Axis 3
Additional rotation: None Angle: 0 Distribution:
Section integration: During analysis Before analysis
Thickness integration rule: Simpson Gauss
Plies
Offset
Shell Parameters
Display
Make calculated sections symmetric
Ply Name
Region
Material
Thickness
CSYS
Rotation Angle
Integration Points
1
Ply-1
(Picked)
COMPOSITE
0.1
<Layup>
30
3


(4)
Module: Assembly
Model: UMAT
Create Instance
Create instances from:
Parts
Models
LAMINA
Instance Type
A meshed part has been selected, so the instance type will be Dependent.
Note: To change a Dependent instance's mesh, you must edit its part's mesh.
Auto-offset from other instances
OK
Apply
Cancel
Step definition
(5)
Create Step
Edit Step
Name: Step-1
Type: Static, General
Basic
Incrementation
Other
Description:
Time period: 1
Nlgeom:
Off
On
(This setting controls the inclusion of nonlinear effects of large displacements and affects subsequent steps.)
Automatic stabilization: None
Include adiabatic heating effects
Boundary conditions
(6)
Create Boundary Condition
Name: BC-3
Step: Step-1
Procedure: Static, General
Category
Mechanical
Electrical/Magnetic
Other
Types for Selected Step
Symmetry/Antisymmetry/Encastre
Displacement/Rotation
Velocity/Angular velocity
Connector displacement
Connector velocity
Continue...
Cancel
(7)
Module: Mesh
Model: UMAT
Object: Assembly
Part: LAMINA
Mesh Part
Element generation
Job definition
(8)
Create Job
Submission
General
Memory
Parallelization
Precision
Preprocessor Printout
Print an echo of the input data
Print contact constraint data
Print model definition data
Print history data
Scratch directory:
User subroutine file:
C:\Users\poori\Desktop\PLANESTRESS-ORTHOTROPIC.for
Results Format
ODB
SIM
Both

Notice that, for the boundary conditions, you need to define two displacement-type boundary conditions at the left and right sides, with the details shown in the figure below.





Let's take a look at how the UMAT code is written for this type of problem.

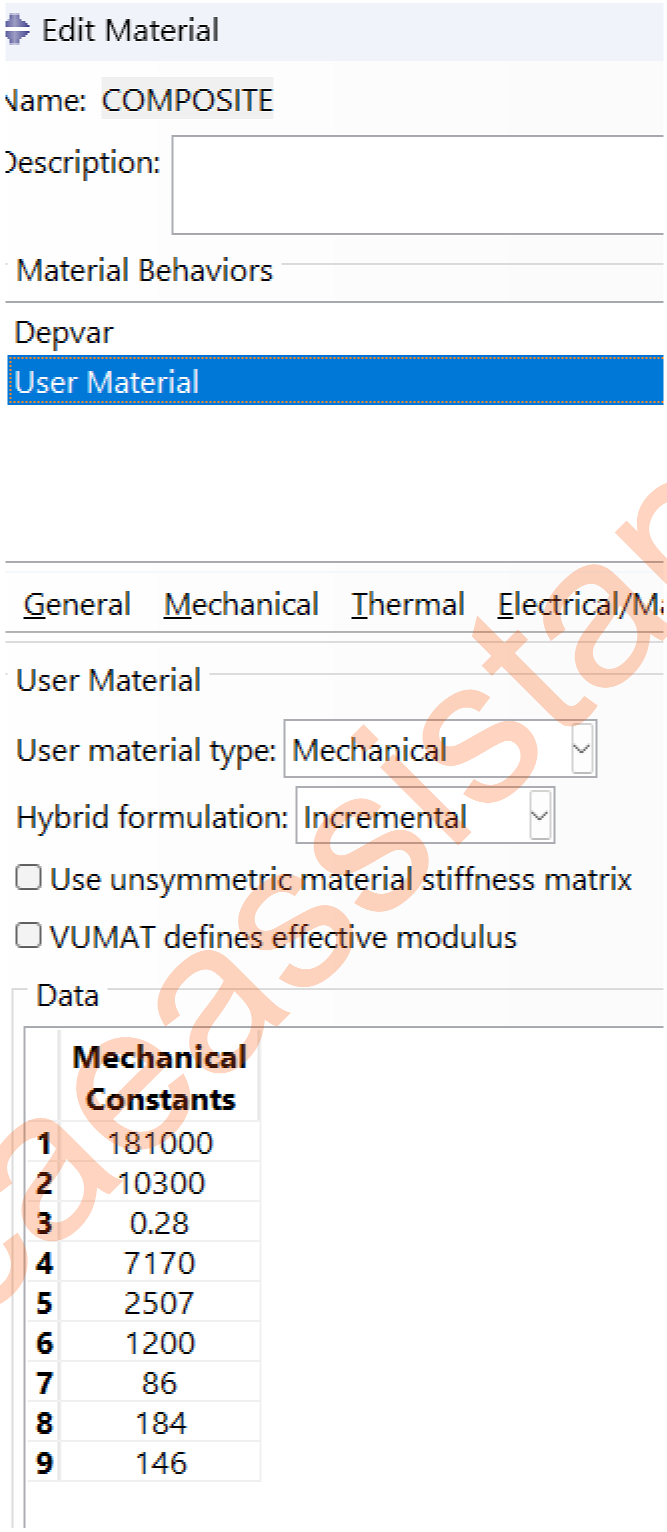
- ***Again, we began by copying the subroutine header from the Abaqus documentation and pasting it into the Fortran code.***

- ***After that, we defined several parameters using the real(8) format, as shown in the next figure.***

```fortran
      SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
     1 RPL,DDSDDT,DRPLDE,DRPLDT,
     2 STRAN,DSTRAN,TIME,DTIME,TEMP,DTEMP,PREDEF,DPRED,CMNAME,
     3 NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,DROT,PNEWDT,
     4 CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,KSPT,JSTEP,KINC)
C
      INCLUDE 'ABA_PARAM.INC'
C
      CHARACTER*80 CMNAME
      DIMENSION STRESS(NTENS),STATEV(NSTATV),
     1 DDSDDE(NTENS,NTENS),DDSDDT(NTENS),DRPLDE(NTENS),
     2 STRAN(NTENS),DSTRAN(NTENS),TIME(2),PREDEF(1),DPRED(1),
     3 PROPS(NPROPS),COORDS(3),DROT(3,3),DFGRD0(3,3),DFGRD1(3,3),
     4 JSTEP(4)
      REAL*8 E1,E2,V12,V21,G12,D11,D22,D66,CFULL(3,3),Y,X,Z,R,F11,F22,
     1F12,F66,F1,F2,F6,XT,XC,YT,YC,SU12
```

Now, we want to determine the values of the variables.

- ***Notice that, we entered some necessary values for the analysis related to our parameters as mechanical constants in Abaqus, as shown in the figure below.***



- ***In the figure, the first component is the Poisson's ratio (12), followed by the shear modulus (12) with its value provided.***

- ***Then, we entered the tensile and compressive strengths along direction 1, as well as the tensile and compressive strengths along direction 2.***

- ***Next, we defined the shear strength.***

- ***Now, as shown in the figure below, we called these values in the code in the same order and format as props, and stored them in the corresponding variables.***

```
E1=PROPS(1)
E2=PROPS(2)
V12=PROPS(3)
G12=PROPS(4)
XT=PROPS(5)
XC=PROPS(6)
YT=PROPS(7)
YC=PROPS(8)
SU12=PROPS(9)
PRINT*,' E2', E2
V21=(E2/E1)*V12
R=1.0/(1.0-(V12*V21))
D11=E1*R
D22=E2*R
D66=G12
D12=E2*V12*R
```

After that, we defined the stiffness coefficients in the code, taking into account the following formula.

$$\begin{pmatrix} D_{11} & D_{12} & 0 \\ V_{21} * D_{11} & D_{22} & 0 \\ 0 & 0 & D_{66} \end{pmatrix} = \begin{pmatrix} \dfrac{E_1}{1-\nu_{12}\nu_{21}} & \dfrac{\nu_{12}E_2}{1-\nu_{12}\nu_{21}} & 0 \\ \dfrac{\nu_{12}E_2}{1-\nu_{12}\nu_{21}} & \dfrac{E_2}{1-\nu_{12}\nu_{21}} & 0 \\ 0 & 0 & G_{12} \end{pmatrix}$$

- ***Now, according to the next figure, we define all the components of the Jacobian or stiffness matrix in the subroutine, using the formula above.***

```
DO K1=1,3
  DO K2=1,3
      DDSDDE(K2,K1)=0.0
  END DO
END DO
DDSDDE(1,1)=D11
DDSDDE(1,2)=D12
DDSDDE(2,2)=D22
DDSDDE(1,3)=0.0
DDSDDE(2,3)=0.0
DDSDDE(3,3)=D66
DDSDDE(2,1)=D12
DDSDDE(3,1)=0.0
DDSDDE(3,2)=0.0
```

- ***Since the problem is two-dimensional, the stiffness matrix is 3x3. In other words, in this problem, NDI is 2 and NSHR is 1, which means NTENS is 3.***

- ***In the next step, following the formula written in the figure below and the corresponding code, we calculated the stress components.***

```
C============================================================
C                            CALCULATE STRESS
C============================================================
         DO I=1,3
          Z=0.0
           DO J=1,3
           Z=Z+DDSDDE(I,J)*DSTRAN(J)
           END DO
          STRESS(I)=STRESS(I)+Z
         END DO
```

$$\sigma_i=\sum C_{ij}\varepsilon_j$$

- ***Now, we determine whether a failure occurs according to the Tsai-Hill criterion. To do so, we used the state variable. Let's see how the failure criterion is checked in the code and stored in the state variable.***

```
IF (STRESS(1).GE. 0.0)THEN
    X=XT
ELSE
    X=XC
END IF
IF (STRESS(2).GE. 0.0)THEN
    Y=YT
ELSE
    Y=YC
END IF
```

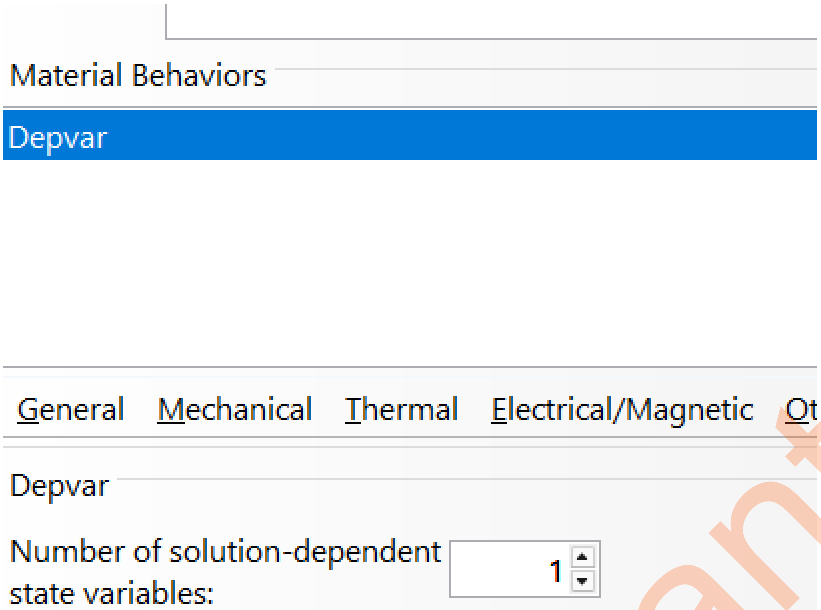
- ***As shown in the figure above, we first need to check whether the stress in each direction is tensile or compressive, and then, determine the material's resistance in that direction.***

- ***This process is carried out in the figure you see.***

- ***In the next step, we need to check whether the material yields according to the Tsai-Hill criterion.***

- ***We rewrite the associated formula and store its value in the first state variable.***

- ***To do so, the number of state variables in the Abaqus model file should be set to 1, as in the next figure.***



As shown in the figure below, the relevant formula has been defined in the code and stored in the first state variable. After that, the subroutine concludes.

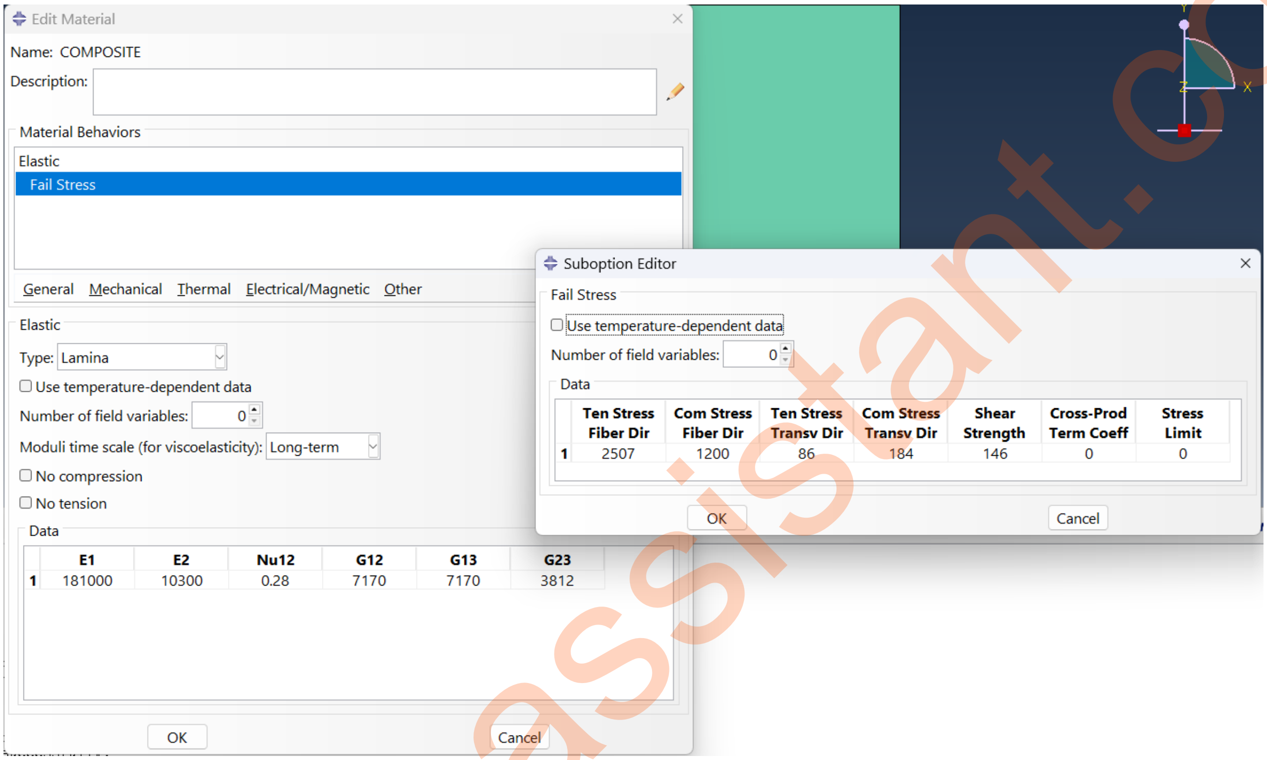
```
      F11=1.0/(X**2.0)
      F1=0.0
      F12=-1.0/(2.0*(X**2.0))
      F22=1.0/(Y**2.0)
      F2=0.0
      F66=1.0/(SU12**2.0)
      F6=0.0
C     FAILURE SURFACE
      F=F11*(STRESS(1))**2+F22*((STRESS(2))**2)+F66*((STRESS(3))**2
     1)+2.0*F12*STRESS(1)*STRESS(2)+F1*STRESS(1)+F2*STRESS(2)+F6*
     2STRESS(3)
      STATEV(1)=SQRT(F)

     RETURN
     END
```

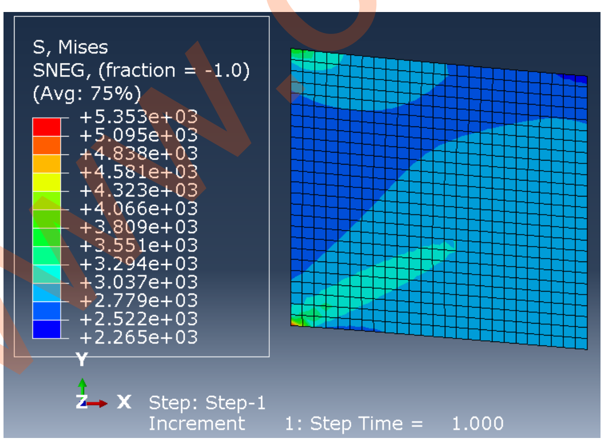
$$F = \frac{\sigma_{11}^2}{X^2} - \frac{\sigma_{11}\sigma_{22}}{X^2} + \frac{\sigma_{22}^2}{Y^2} + \frac{\sigma_{12}^2}{S^2} < 1.0$$

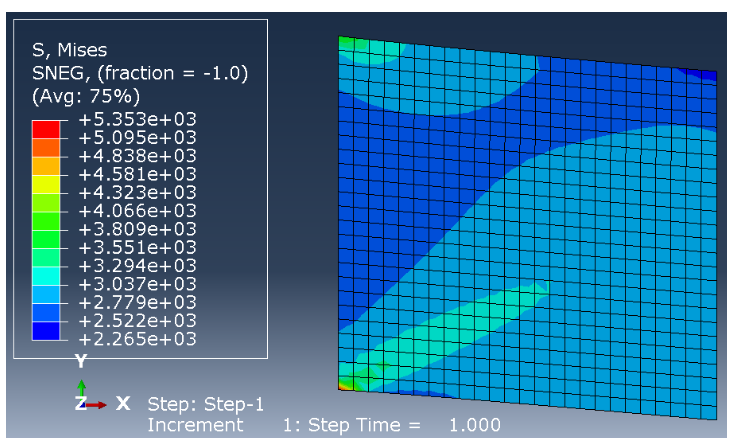
RESULTS

In this section, we solved the problem once using the UMAT subroutine and once using Abaqus's built-in material model, with the same properties. The process for defining the materials in Abaqus is illustrated in the figure below.

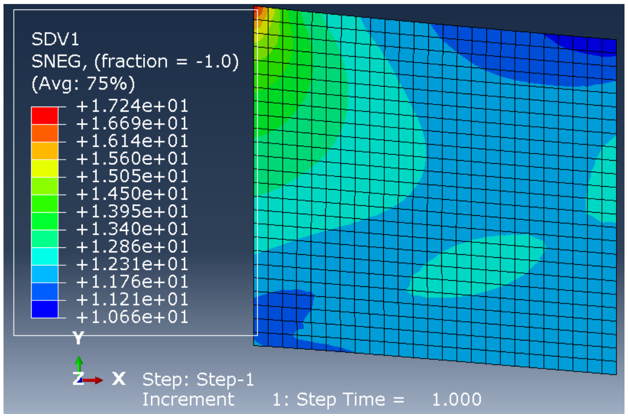


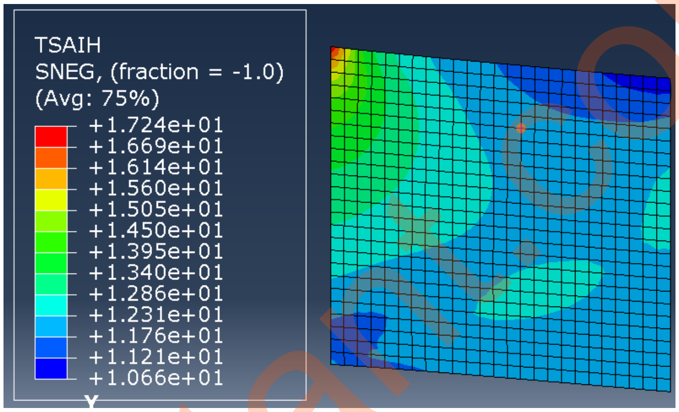
We plotted the Mises stress in the job output. As shown in the figure below, the pattern obtained using Abaqus's material model and the UMAT subroutine is identical, verifying the accuracy of the UMAT code.





After that, we plotted the Tsai-Hill criterion using Abaqus's built-in model and the value of statev(1) from the UMAT code. Once again, as shown, the results are identical.





- ***We have provided the Abaqus modeling files and Fortran code for this example at this link. You can check it, while a step-by-step learning video is also available through the link for free.***