SwiftComp™

Abaqus-SwiftComp GUI

Version 1.2.2 User's Manual

April 21, 2017

Bo Peng
Su Tian
Lingxuan Zhou
Wenbin Yu

Multiscale StructuralMechanics

PURDUE™ AERONAUTICS & ASTRONAUTICS

Table of Contents

1 GENERAL INFORMATION

Based on the recently invented Mechanics of Structure Genome (MSG), SwiftComp™ provides an efficient and accurate approach for modeling composite materials and structures. It can be used either independently as a tool for virtual testing of composites or as a plugin to power conventional FEA codes with high-fidelity multiscale modeling for composites. SwiftComp™ implements a true multiscale theory which assures the best models at a given level of efficiency to capture both anisotropy and heterogeneity of composites at the microscopic scale or any other scale of user's interest. SwiftComp™ enables engineers to model composites as a black aluminum, capturing details as needed and affordable. This saves orders of magnitude in computing time and resources without sacrificing accuracy, while enabling engineers to tackle complex problems effectively.

SwiftComp™ can be used as a standalone code or as a plugin for other commercial codes. To facilitate the use of SwiftComp™, a simple graphic user interface (GUI) with a toolbar integrated the functions of SwiftComp™ is developed in Abaqus. This manual focuses on explaining how to use Abaqus-SwiftComp GUI.

Chapter One introduces the installation of Abaqus-SwiftComp GUI, and provides a general guide for modeling and visualization in Abaqus-SwiftComp GUI. The guidance in Chapter One is brief, more details are included in the rest of chapters by examples.

Chapter Two describes how to quickly build common SGs and carry out homogenization and dehomogenization using SwiftComp™. Common SG can be built through a few steps, which will be a good start to get familiar with Abaqus-SwiftComp GUI.

Chapter Three introduces how to create user customized SGs through examples of general laminate (a common 1D SG), Square Pack Microstructure (a common 2D SG), 2D SGs for cross-sections with arbitrary shape, and 2D SGs for cross-section with airfoil shape. These two examples should give users a good preparation to build user-defined models according to different analysis needs.

1.1 Installation and Get Started

There are two ways to use this toolset.

Quick Start: To get started to use Abaqus-SwiftComp GUI on your local machine, you just need to simply unzip the distribution package into a folder of your own choice, then in this folder, click the short cut file *Abaqus-SwiftComp GUI* or type *abaqus cae –custom SwiftCompGUI.py* in the Abaqus Command to launch to Abaqus-SwiftComp GUI.

Installation: In this way, user can start the GUI and work on analysis from anywhere. Several steps are needed.

1) Download and unzip the package to any location and copy the full path of the directory;
2) Add the path to the *PYTHONPATH* of Abaqus. This constant is in the file *abaqus.aev*, which is usually located at *abaqus_home_directory*\SMA\site\. Insert the full path just before *.:$PYTHONPATH* at the end. Pay attention to the direction of the slash. Save;
3) Add a custom command to Abaqus. Open file *abaqus.app* in a text editor, which is usually located at *abaqus_home_directory*\SMA\site\. Append *swiftcomp cae –custom SwiftCompGUI* at the end of the command list. Save;

To start the GUI, type *abaqus swiftcomp* in the Abaqus Command.

Of course, you also need to have Abaqus and SwiftComp™ already installed on your computer. To install SwiftComp™ you should request the code from AnalySwift and follow the instruction inside the SwiftComp™ release package for installation.

After launched Abaqus-SwiftComp GUI, please set the Abaqus work directory (File->Set Work Directory) to be the work directory of your own choice. Please note that all the files generated will be stored in the work directory and the valid SwiftComp license requested from AnalySwift should also be stored in your work directory.

1.2 Abaqus-SwiftComp GUI General Guide

The only difference compared with the original Abaqus GUI is the added toolbar group as shown in Figure 1-1. The toolbar group composes of gadget buttons (1, 2), SG creation tool buttons (3, 4, 5, 6, 7, 8), SwiftComp™ analysis and visualization tool buttons (9, 10, 11, 12). Their functions can be simply described as follows:

1) **Work plane**: set sketch plane for 1D/2D customized SGs.
2) **New layups**: add new layups for 1D/2D SGs.
3) **1D SG**: create 1D SG, including common SGs and customized SGs.
4) **2D common SG**: create 2D common SGs.
5) **Assign layups**: create laminate for 2D cross-sections.
6) **Erase layups**: delete laminate for 2D cross-sections.
7) **Read file**: create 2D cross-section from input data file.
8) **3D common SG**: create 3D common SGs.
9) **Homogenization**: carry out homogenization analysis.
10) **Macro model**: import the homogenized properties.
11) **Dehomogenization**: carry out dehomogenization analysis.
12) **Visualization**: visualize the results with only SwiftComp analysis files.

Figure 1-1: The added toolbar group in Abaqus-SwiftComp GUI.

1.2.1 SG preparation

SG creation tool buttons (3-8) can help the user to set up the configurations of SGs.

Abaqus functions are highly integrated in building the SGs, such as materials, composite layup, section assignment, mesh etc. But some settings in the original Abaqus GUI are not used, and the meanings of some parameters are different, which needs special attentions from the user. The details will be explained in the following sections. For convenience, the term '**Abaqus GUI**' is referred in this manual when using the functions of the original Abaqus, otherwise '**Abaqus-SwiftComp GUI**' is explicitly specified. *Italic texts* represents the terms shown in dialog box of Abaqus GUI, and texts in ' ' represents terms shown in dialog box of Abaqus-SwiftComp GUI.

Geometry

Users can create common SGs using the 1D SG button (3), 2D SG button (4), or 3D SG button (8), which provides common microstructures of composites for 1D, 2D and 3D SGs respectively. Or they can build customized 2D SGs using button 5, 6, 7 or Abaqus functions.

According to the convention of SwiftComp™, for 1D SGs, the geometry should be aligned with Z direction in the global coordinate system; for 2D SGs, the geometry should be in the Y-Z plane in the global coordinate system. The work plane button help the user set up the work plane for customized 1D SGs and 2D SGs.

For 1D customized SG, the work plane button also creates a temporary line geometry in Z direction, and a set named 'Set_Layup'. Then the user can use *Create Composite Layup* tool button of Abaqus GUI to create composite layup, in which *Region* is assigned using the set 'Set_Layup'. The next step to create a 1D customized SG is to choose the 'Composite Layup' method in dialog box of the 1D SG button. After this step, the temporary line geometry will be deleted, and the actual 1D SG composed of consecutive line segments is created, of which each line segment represents a ply.

To create the geometry for a 2D customized SG, the next step is to use *Create Shell: Planar* button. Other procedures are the same as creating a 3D shell part in Abaqus GUI.

The procedure to create 3D customized SG geometry is the same as that to create a 3D part in Abaqus GUI.

Material

In the Material module, the *elastic* materials properties can be defined by types: *Isotropic, Engineering Constants, Orthotropic* and *Anisotropic*. If effective density of SG is required, *density* of the material should be specified; if thermoelastic analysis is conducted, *Expansion* and *Specific Heat* should be specified. If density, expansion and specific heat are required in the analysis but not specified, Abaqus-SwiftComp GUI will not ask for it, but use the default values, information will be shown in the message area. All the material properties can only be defined in the most basic way, advanced options such as *temperature-dependent data*, *discrete fields* cannot be used in the current version. Just like in Abaqus GUI, materials not used in SGs are allowed to exist.

Materials can also be imported from a file, whose format is XML. For more details, please go to Section 3.3.2.

Sections

In Abaqus-SwiftComp GUI, section types *Solid-Homogeneous*, *Shell-Homogeneous, Solid-Composite, Shell-Composite* are applied. Sections are defined in the same way as in Abaqus GUI. Material sections not assigned to SGs are allowed to exist. Note: For 2D SGs, the geometry should be in the Y-Z plane in the global coordinate system. Abaqus only allow shell feature in 3D space to be placed in the Y-Z plane, thus practically, user can only use *Shell-Homogeneous* sections.

Section Assignment

Material section assignment is defined in the same way as in Abaqus GUI. Section assignments are allowed to be depressed, and the depressed sections assignment will be ignored.

Composite Layup

Composite Layup of Conventional Shell of Abaqus GUI is used in creating composite layup for 1D SGs and 2D SGs. Plies are allowed to be depressed or be edited using the tools in the dialog box. However the current version of Abaqus-SwiftComp GUI sometimes will crash when the user revises an existing composite layup or changes the part name of a 1D SG part, so it is highly recommended to save your work before the revision.

For 1D SG, information of the *Index of the plies (*not the *Ply Name), Material, Thickness, Rotation Angle* and *Offset* setting of this dialog box will be read into the input file (*.sc) of SwiftComp™ when doing homogenization. *Region* should be the set 'Set_Layup' which contains the whole 1D SG geometry. The same as that in Abaqus, the laminate is stacked from index '1' to the maximum ply index. Things need to know also include:

1) The *layup orientation* can only be defined based on the *Part global* coordinate system.
2) Different from what defined in Abaqus, *Thickness* in the table is the actual thickness of each ply.
3) If Plate/Shell analysis is conducted, the first 4 option of *Offset* tab book can be used to shift the coordinate in the input file of SwiftComp™, which have the same definition as that of Abaqus. The shift will not show on the geometry of the SG. This will be further explained in Section 2.1.1 and Section 3.1.1.
4) Only one Composite layup is allowed to exist in a 1D SG part.

Layup file format

Composite Layup information can also be imported into Abaqus-SwiftComp GUI by choosing layup file (*.dat) with specific format. Currently this format is only valid for 1D SG. An example of the format is shown in Figure 1-2.

The file is composed of 3 parts.

In the first part (in the red frame) it is the layup control parameters.

The second part (in the blue frame) contains the information of each ply from bottom to top, which arranged in the sequence of: ply thickness (integer/float), ply angle (integer/float), material ID (integer).

The third part (in the green frame) provide the corresponding relationship between the material ID and the material name. The material with names shown in this part must be defined previously.

In details, in the first part, the first integer '4' is the number of layers written in the second part, which is not necessarily the actual total number of the laminate). The second integer '2' is the number of material types written in the third part. The character 's' in the third place means 'symmetric', therefore the current layup angle is [45/30/60/-45]s. This character can also be 'a' which means 'antisymmetric', if in the current example it is changed to 'a', the layup angle will be [45/30/60/-45/60/30/45]. The last float '0.005' is the offset ratio, which is explained in Section 2.1.1.

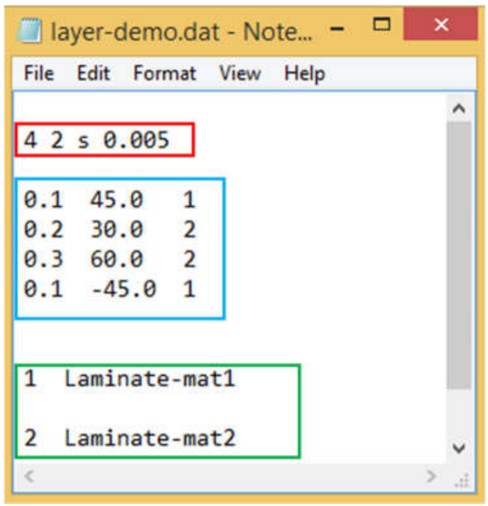


Figure 1-2: Layup file format example.

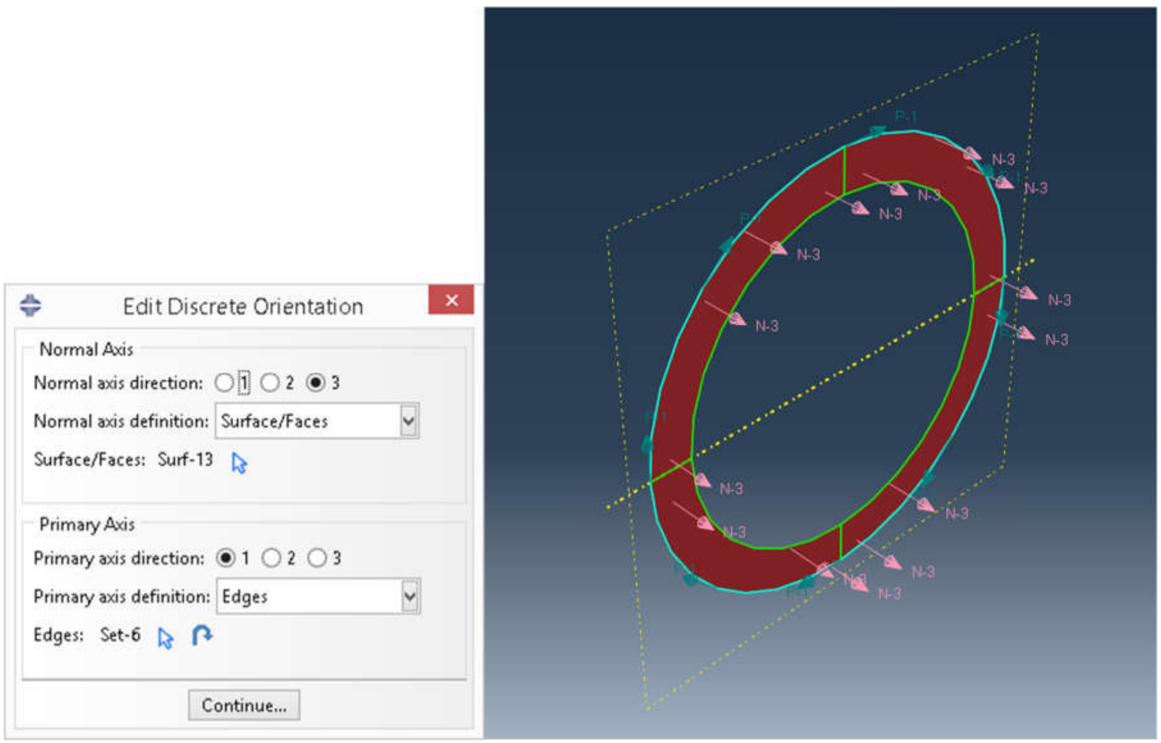
Other notes:

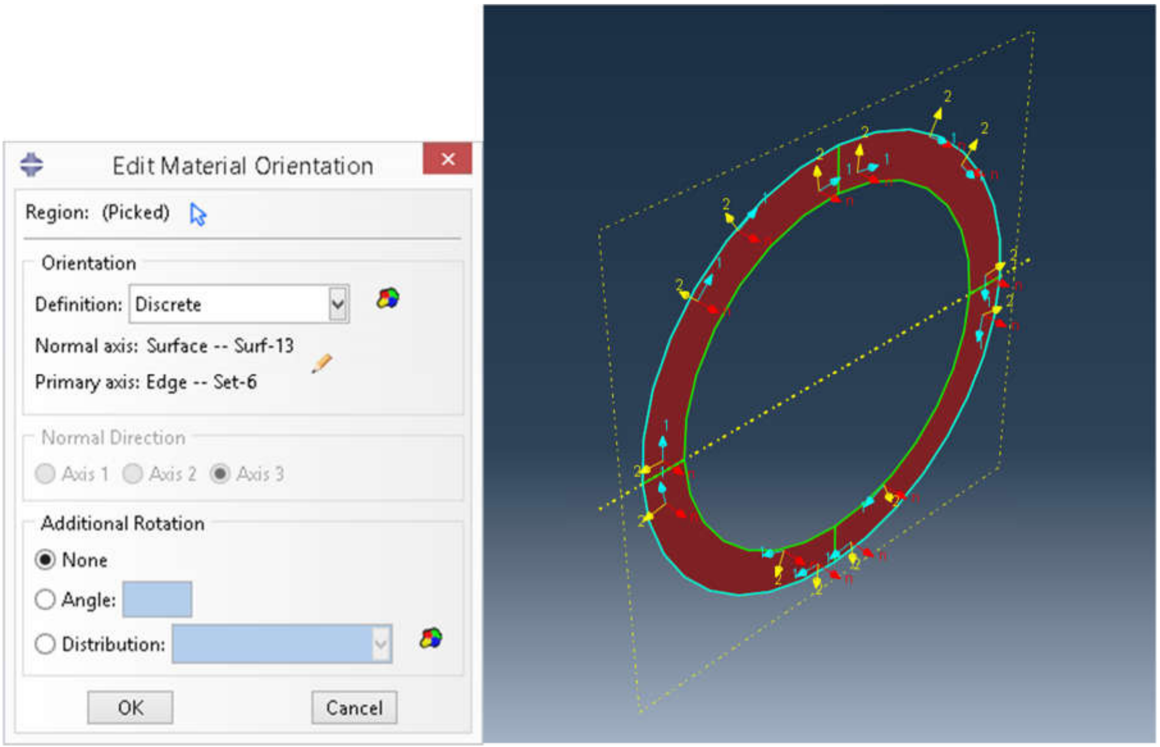
1) The character in the third place to denote 'symmetric/antisymmetric' can also be any string beginning with 's' or 'S' for symmetric laminate (for example 'sym'), or any string beginning with 'a' or 'A' for antisymmetric laminate (for example 'Antisym'). If the laminate is not symmetric or antisymmetric, any other string can be inputed in the third place of the first row.
2) If no offset is necessary, the float in the fourth place is not needed.

A layup can also be imported from a XML file. The 'New layups' function will read the material file and layup file and create one or more *Solid-Composite* sections. For more details, please go to Section 3.3.2.

Orientation Assignment

User can use Abaqus's own function to assign local element orientation. For 2D cross-section (shell feature in 3D space), there are several things user need to pay attention to. First, no matter what choices user makes for local axes numbering, the two axes shown at last will always be labeled 1 and 2. However, different choices may cause unexpected pointing directions of the two axes. Hence we need to set a rule here: following the convention of local coordinates used in SwiftComp, if a series of elemental coordinates changes along a curve C, the axis 1 in Abaqus will be the axis y2 in SwiftComp and tangent to C, and the axis 2 in Abaqus will be the axis y3 in SwiftComp and perpendicular to C. Second, it doesn't matter for axis 2 pointing inward or outward, as long as it keeps consistency with the global coordinate ***and*** fiber orientation defined by user, since the fiber orientation will change if the direction of axis 2 (y3) changes.





<u>Mesh</u>

Mesh must be generated on *Part* instead of on *Instance*. For 2D customized SGs, element types assigned in Abaqus GUI can be triangular (*TRI*) and quadrangular (*QUAD*), linear or quadratic such as S3, STRI3, S4, S4R, S8R, S8R5, etc. For 3D customize SGs, element types assigned in Abaqus GUI can be hexagonal (*Hex*) and tetrahedral (*Tet*), linear or quadratic such as C3D4, C3D10M, C3D8R, C3D8, C3D20R, C3D20 etc. However, only the nodal coordinates and element connectivities are read into SwiftComp^{TM} for analysis, therefore the results are the same whether C3D4 or C3D4R is used.

For 1D SGs, mesh will be done automatically by using the 1D SG button. The element type assigned in GUI is B31, but just as shown in the dialog box of the 1D SG button, the actual element type used in the SG can be two-noded, three-noded, four-noded or five-noded. Each edge in the geometry of 1D SG represents 1 ply and has only 1 element in SwiftComp^{TM}. Two-noded elements will be accurate for 3D solid, but five-noded elements are recommended for 2D Plate/Shell.

Special attention needs to be paid on generating periodic mesh. For periodic materials or heterogeneous materials with local periodicity, SwiftComp^{TM} requires a mesh with periodic nodes on the edges to rigorously satisfy the periodicity requirement. Theoretically speaking, a node on the boundary surface must have a corresponding node on the parallel boundary surface with the same coordinates except the coordinate component normal to the boundary surface. However, currently this can only be guaranteed by the user. However, for complex microstructures, if it is difficult to create corresponding nodes on the periodic edges, SwiftComp^{TM} provides a way to automatically provide the best approximate solution.

1.2.2 MSG analysis

SwiftComp™ analysis tool buttons include Homogenization button, Macro model button and Dehomogenization button.

Homogenization

Users can invoke SwiftComp™ to compute the effective properties for different structural models (beam, plate/shell or 3D solid). Please refer to SwiftComp™ manual for the meaning of the parameters. In the 'Dimensionally Reducible Structures' group box, only the 'Classical' in the specific model is supported in the current version. In the 'Options' group box, only 'Elastic' and 'Thermoelastic' analysis types are supported, the other options should be kept as default. Note 1D SG cannot be used for beam model.

The Abaqus-SwiftComp GUI can generate the SwiftComp input file (*.sc) from a CAE model or an Abaqus input file. The user can check the box before 'New SwiftComp file name' to specify the SwiftComp input file name, such as 'Layup2s', which will generate a file 'Layup2s.sc'. If the file name is not specified, the default file name will be in the form of:

```
partName_nSG#_macroModelDimension_elementType.sc
```

For 1D SG, the elementType is in the form of 'n#', where '#' is the number of nodes in each element of the SG. For 2D and 3D SG, the elementType is the element type name of the first element in the model, using the term in Abaqus.

The GUI can determine 'Omega' (the volume of SG in the macro model) by itself in simple cases as follows:

1) When the SG part has geometry information, and is a line, rectangle or cube.
2) When the SG part has no geometry information, and is a line, rectangle or cube (voids can be in the SG). In this case, when the SG has a huge a number of nodes, specifying 'Omega' by the user will speedup the preparation of the model needed for homogenization.

After the homogenization of SwiftComp™, the effective properties will pop up automatically. In some situations that error may occur, the user can check the command line window and it should report the error message of running SwiftComp™, which gives your indication what is wrong with the model.

Macro model: Import the homogenized properties

The effective properties obtained from homogenization can be used in macro model analysis. Using this button can help user import the effective properties into a newly created model with the name of the SwiftComp input file. In the current version of Abaqus-SwiftComp GUI, effective properties of 1D macroscopic model (beam) will not be imported into Abaqus, since no beam element in Abaqus can take a fully populated stiffness matrix. Thus user need to choose other methods to carry out the macroscopic model analysis for beams. More details are explained in Section 4.0.

Dehomogenization

To conduct dehomogenization analysis, users must provide the homogenization files (*.sc, *.sc.k, *.sc.opt) of the SG and the global behavior parameters. The homogenization files must be in the current work directory. There are two methods to provide the homogenizations files. The details are explained in Section 2.1.3.

User must specify the global behavior parameters. Please refer to SwiftComp™ manual for the meaning of each the global behavior parameter in this dialog box. With provided global behavior, users can invoke SwiftComp™ to compute the local fields including displacements, stresses and strains and other possible fields of interests, depending on the analysis. The information will then be used to create an odb file with the same name as the SwiftComp™ input file, so that the user can use the functions in the Visualization module of Abaqus GUI for postprocessing of the results. The post-processing results are automatically loaded, but the users need to change to the Visualization model by themselves. Contour plots are available for all local fields including three displacement components (U1, U2, U3) and its magnitude, six nodal strain components (EN11, EN22, EN33, 2EN23, 2EN13, 2EN12) and the derived quantities such as Mises strain, six nodal stress components (SN11, SN22, SN33, SN23, SN13, SN12), and other possible fields of interest, depending on the analysis and the derived quantities such as Mises stress.

In the current version, the odb file contains no information of geometry and material properties and geometry, although sections are kept. Under the *.odb trunk of the Output Database tree, sections can be found, and each section contains the elements with the same material properties. For 1D SGs, each section contains elements with the same material properties and the same layer angle.

1.2.3 Visualization button

Visualization

Users can use the Visualization tool button to create an *.odb file with the dehomogenization result files of SwiftComp™ (*.sc.u, *.sc.sn), and view the contour plots of the local fields including displacements, stresses and strains. Therefore it is not necessary to always save the *.cae file and *.odb file of the SG. Different material sections can be viewed, but the material properties cannot be reached in the *.odb file.

1.2.4 Limitations of the current version

As stated previously, the current version of the Abaqus-SwiftComp GUI has the following limitations:

1) Change part name and edit the composite layup of 1D SG may cause unexpected crash, therefore save your work frequently is highly recommended.
2) When a lot of work is done in the same session of Abaqus, sometimes in the command line window it shows: cannot find Notepad or SwiftComp, or the input line is too long. The user just need to close the current session of Abaqus and also close the command window, then restart the command window and the GUI.

2 CREATE COMMON MODEL

Abaqus-SwiftComp GUI provides a convenient way to create some common SG models. Engineers can easily create the geometry of these models, and invoke SwiftComp™ to perform homogenization and dehomogenization for different composites with common SGs.

Currently, Abaqus-SwiftComp GUI provides the following common SG models:

1) 1D SG: laminate.
2) 2D SG: square pack microstructure with or without interphase region, hexagonal pack microstructure with or without interphase region.
3) 3D SG: spherical inclusion microstructure with or without interphase region.

In this chapter, we will use some common SGs for different models (e.g. solid, plate/shell and beam) to illustrate how to use Abaqus-SwiftComp GUI to perform homogenization and dehomogenization.

2.1 Laminate (1D SG)

2.1.1 1D SG preparation

1D SG is simple compared with 2D/3D SG, which allows the GUI save most of the work for the user. This section will introduce 4 methods developed for generating 1D SG respectively. The four methods are: Fast Generate, Composite Layup, Composite Sections, Read from file. Fast Generate function only applies to laminates with constant layer thickness and single material property, while the other 3 methods can apply to general laminates.

Note users must pay attention to the general remarks illustrated in Section 1.2.1 Composite Layup.

The buttons for creating 1D SG are shown in the Figure 2-1.

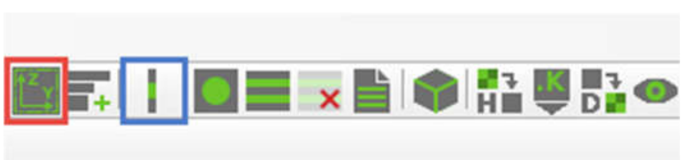


Figure 2-1: Work plane button (red box) and 1D SG button (blue box).

<u>Method 1: Fast Generate</u>

To create a 1D common SG quickly, Fast Generate function can be applied. Fast Generate function only applies to laminates with constant layer thickness and single material property.

Step 1: Create materials

The first step is to create a material in Abaqus GUI. The material 'material-1' used in this example has the properties as shown in Table 2-1.

Step 2: Create geometry and mesh of 1D SG

Then the user can click the 1D SG button in the blue frame shown in Figure 2-1. In the dialog box popped out in Figure 2-2, choose 'Fast Generate', and input the required information: layup, ply thickness, offset ratio, choose model and material which has been created, and element type. Layups can be specified following the tips provided in the dialog box. The meaning of the offset ratio is illustrated in Figure 2-3. The number after bracket means the repeating times and "s" means symmetry.

Table 2-1: Material properties.

Material	E_1 GPa	E_2 GPa	E_3 GPa	υ_{12}	υ_{13}	υ_{23}	G_{12} GPa	G_{13} GPa	G_{23} GPa
Material-1	117.0	8.54	8.54	0.278	0.278	0.5	3.9	3.9	2.83

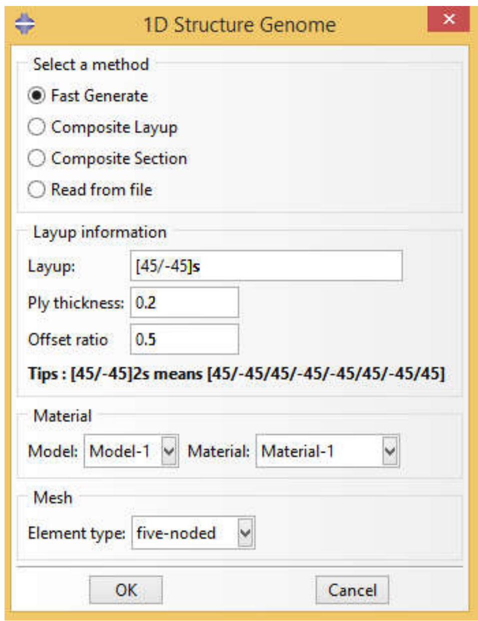
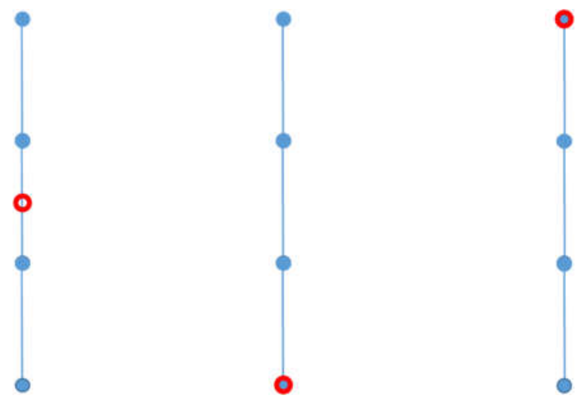


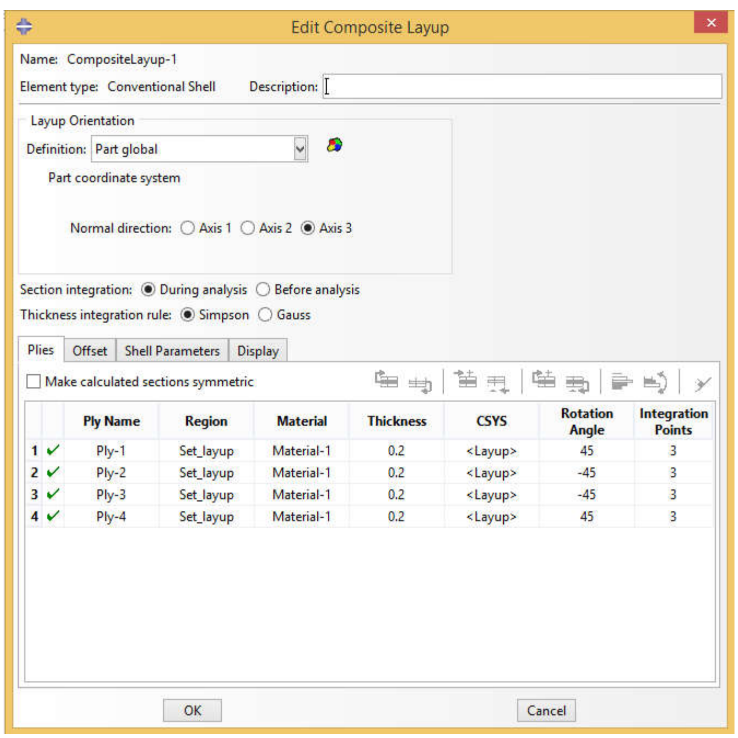
Figure 2-2: Create 1D SG dialog box.



(a) Offset = 0 (b) Offset = -0.5 (c) Offset = +0.5

Figure 2-3: The coordinate origin of the coordinate system in the generated SwiftComp input file (*.sc) by specifying Offset ratio, where the red hollow circle represents the coordinate origin.

Using the information provided in this dialog box, a 1D SG part named as 'Laminate' is created. The layup and offset ratio information are stored in the *Composite Layup* as shown in Figure 2-4. The offset ratio will not be shown in the geometry of the part 'Laminate', but only be saved in the *Offset* tab book in the dialog box and later be used in generating the input file of SwiftComp™. In other words, Abaqus *Composite Layup* manager can be used to double check whether the inputs for Fast Generate are correctly provided and interpreted.



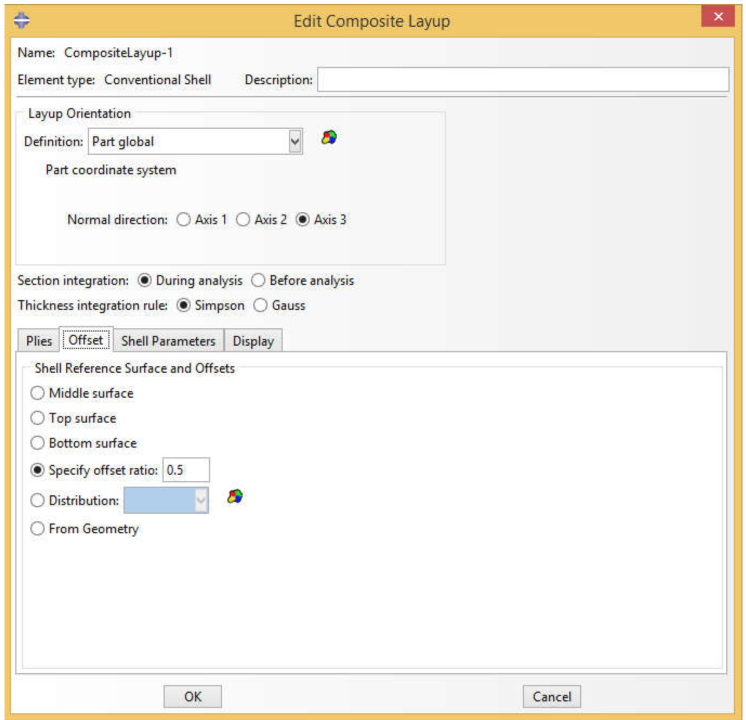


Figure 2-4: Composite Layup contains the layup and offset ratio information.

For laminates, the mesh has been generated right after defining laminate. As stated in Section 1.2.1-> Mesh, the element type chosen in Abaqus is B31 with 2 nodes in each element. In this example, we create a 1D SG containing 4 plies ([45/-45]s laminate), which require 4 elements in SwiftComp. Each five-noded element in SwiftComp is composed of 4 B31 element in Abaqus, therefore the 1D SG has 16 B31 elements in all as shown in Figure 2-5. Note it is not an approximation but a technique to trick Abaqus to generate 5-noded elements for SwiftComp, which is particularly needed if the macroscopic structural model is a plate/shell model.

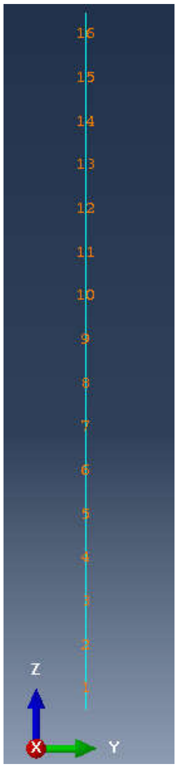


Figure 2-5: 16 B31 elements are generated for a 4 ply 1D SG with five-noded element type in SwiftComp.

Method 2: Composite Layup

This method takes advantage of the Composite Layup function of Abaqus GUI, which allows ply thickness and material properties to be different in each ply.

Step 1: Create materials

To create a customized 1D SG, the first step is to create the materials in Abaqus GUI. In this example, we use the material properties shown in Table 2-2.

Table 2-2: Material properties.

Material name	E_1 GPa	E_2 GPa	E_3 GPa	υ_{12}	υ_{13}	υ_{23}	G_{12} GPa	G_{13} GPa	G_{23} GPa
Laminate-mat1	250.0	50.0	50.0	0.25	0.25	0.25	2.0	2.0	5.0
Laminate-mat2	200.0	20.0	20.0	0.3	0.3	0.3	2.0	2.0	5.0

Step 2: Set work plane

As stated in section 1.2.1 Geometry, 1D SG must align with Z axis. This is achieved by the work plane button (in the red frame in Figure 2-1). Click the work plane button, the dialog box in Figure 2-6 pops out, then choose the model and give the part name for the 1D SG part, choose the SG dimension to be 1D. The work plane button will create a datum plane and a datum axis in the part.

The work plane button also creates a temporary line geometry in Z direction, and a set named 'Set_Layup' as shown in Figure 2-7.

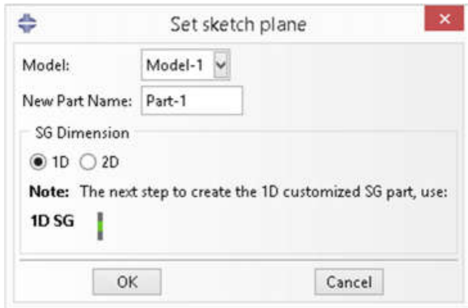


Figure 2-6: Work plane button dialog box.

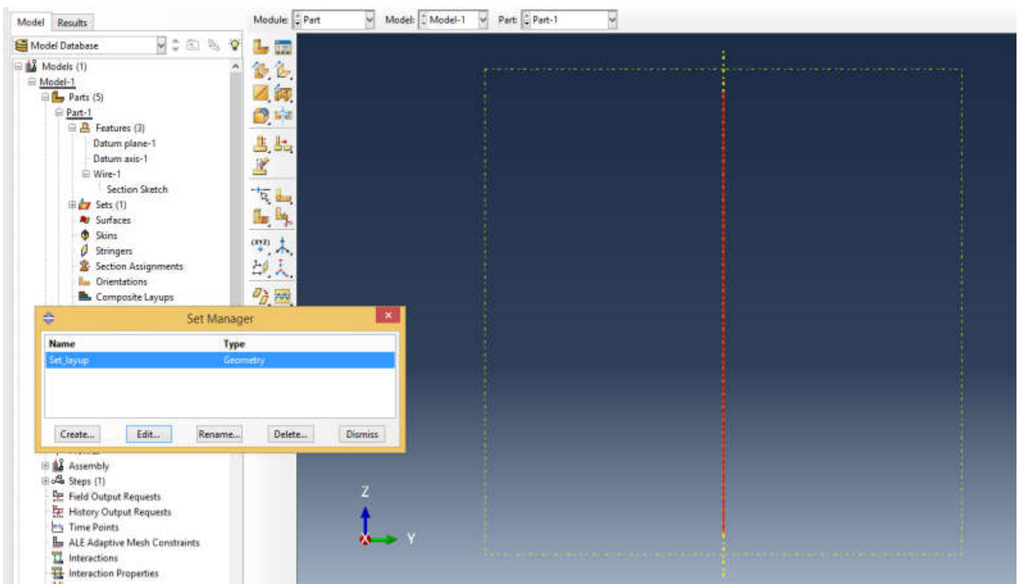


Figure 2-7: After using the work plane button.

Step 3: Create Composite Layup

Then the user can use *Create Composite Layup* tool in the Property module to create composite layup and assign *Region* using the set 'Set_Layup' as shown in Figure 2-8 and Figure 2-9. The first 4 options of the 'Offset' tab book can be used to specify offset of the coordinate origin in the *.sc file. In this example, the default offset ratio (0.0) is kept. The parameter 'Offset' has the same meaning of 'Offset ratio' as explained in Section 2.1.

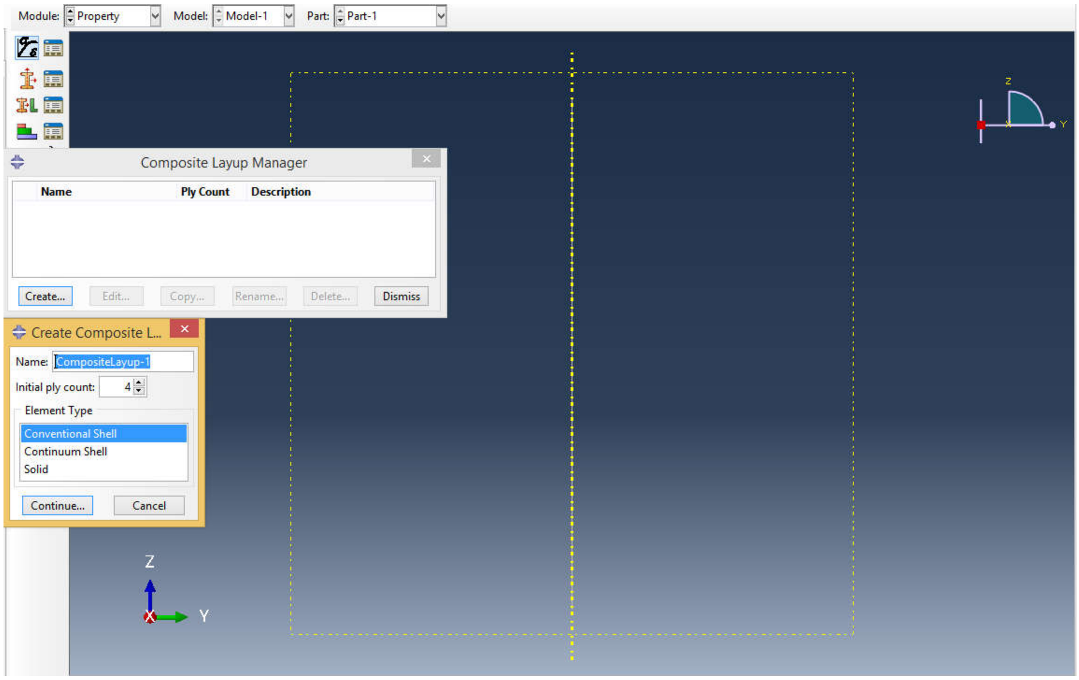
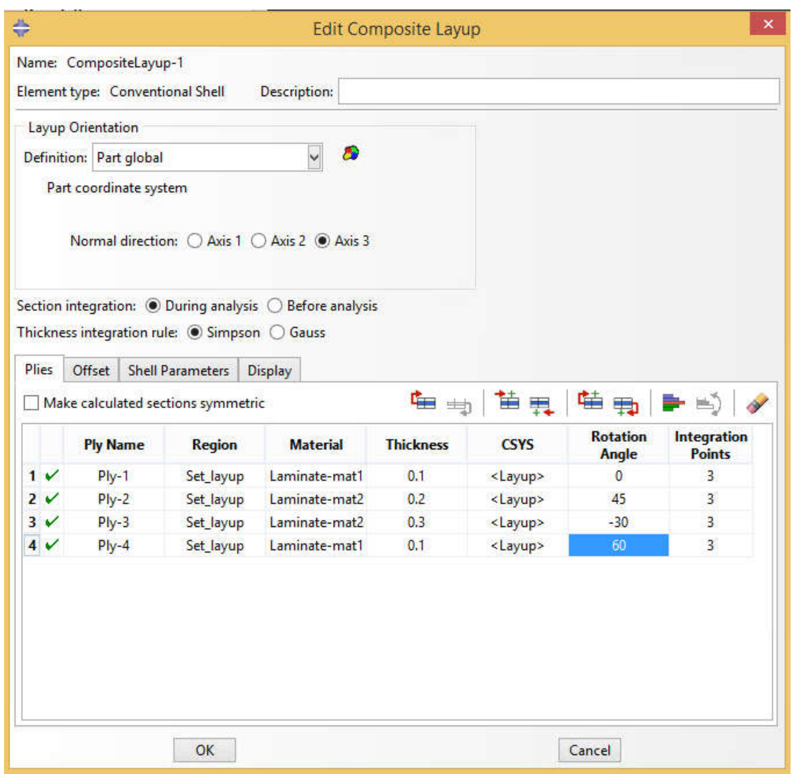


Figure 2-8: Create composite layup.



	Ply Name	Region	Material	Thickness	CSYS	Rotation Angle	Integration Points
1 ✔	Ply-1	Set_layup	Laminate-mat1	0.1	<Layup>	0	3
2 ✔	Ply-2	Set_layup	Laminate-mat2	0.2	<Layup>	45	3
3 ✔	Ply-3	Set_layup	Laminate-mat2	0.3	<Layup>	-30	3
4 ✔	Ply-4	Set_layup	Laminate-mat1	0.1	<Layup>	60	3

Figure 2-9: Configure the composite layup

Step 4: Create geometry and mesh of 1D SG

The next step to create a 1D customized SG is to choose the ‘Composite Layup’ method in dialog box of the 1D SG button (Figure 2-10). After this step, the temporary line geometry will be deleted, and the actual 1D SG composed of consecutive edges is created, of which each edge represents a ply.

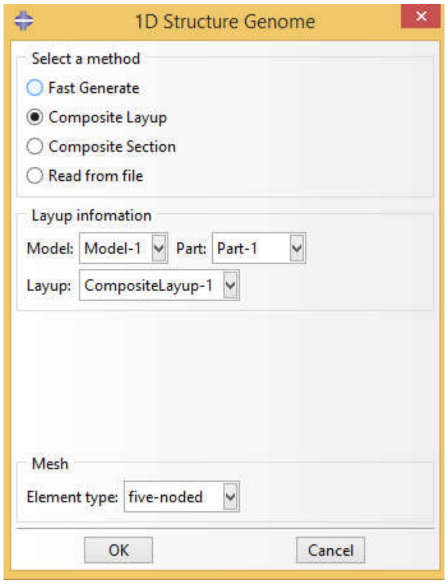


Figure 2-10: Use Composite Layup method to create general 1D SG.

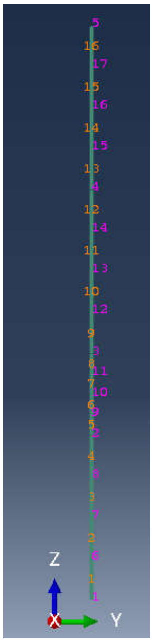


Figure 2-11: The 1D SG created (numbers in orange are element numbers).

After Step 4, the 1D SG has been built. The user can check the element number in the Mesh module as shown in Figure 2-11. Similar to the 1D SG generated by fast generated method, each edge represents an element in SG, which contains 4 B31 elements. Therefore there are 16 B31 elements in the model. The created part can be used for homogenization and dehomogenization analysis. Clearly the 'Fast Generate' method does the same thing as what does in 'Composite Layup' method for simple laminate cases.

Method 3: Composite sections

This method take advantage of the Sections-> *Composite sections* function of Abaqus GUI, which allows ply thickness and material properties to be different in each ply.

Step 1: Create materials and Composite sections

To use this method, first the user need to create mateirals and composite sections in the model, as shown in Figure 2-12. The *Element Relative Thickness* is the actual ply thickness. *Symmetric layers, Layup name* and *Ply Name* in the dialog box are ignored in creating 1D SG.

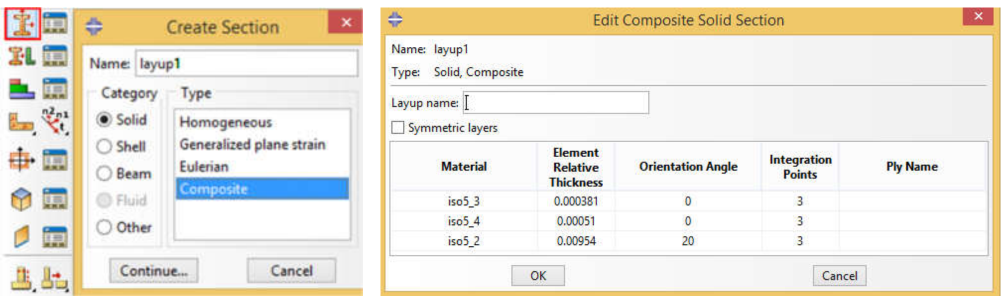


Figure 2-12: Create Composite Sections.

Users can use function 'New layups'  to create Solid-Composite sections, as explained in Section 3.3.

Step 2: : Create geometry and mesh of 1D SG

The next step to create a 1D customized SG is to choose the 'Composite Section' method in dialog box of the 1D SG button (Figure 2-13) and input the required parameters. The parameter 'Offset' has the same meaning of 'Offset ratio' as explained in Section 2.1.

Click OK, the 1D SG will be created. An Composite Layup containing the information from the Composite Section will also be created. In Homogenization, the information of the 1D SG will be read from the Composite Layup instead of the Composite Section. Therefore multiple Composite Sections are allowed to exist, but only 1 Composite Layup is allowed to exist in a SG.

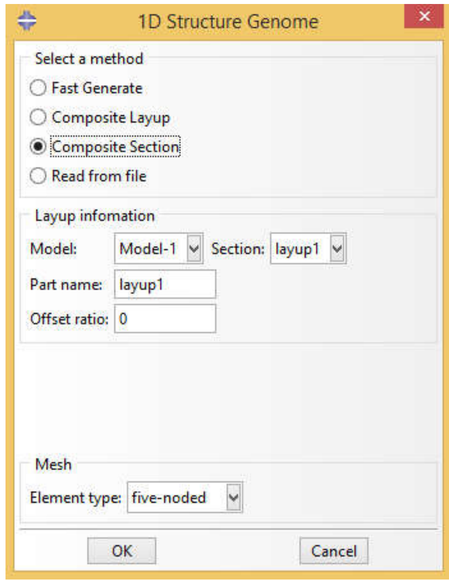


Figure 2-13: Use Composite Section method to create general 1D SG

<u>Method 4: Read from file</u>

This method also allows ply thickness and material properties to be different in each ply. The information except material properties will be read from a *.dat file.

In this example, we use the file showin in Figure 2-14.

Step 1: Create materials

To use this method, first the user need to create mateirals in the model. Here we use the material properties in Table 2-1.

Step 2: Prepare layup input file

In this example we prepare a file as shown in Figure 2-14. The format has explained in the Section 1.2.1 Layup file format.

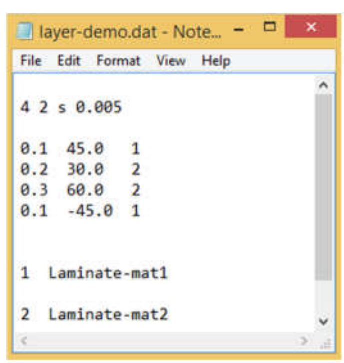


Figure 2-14: Layup input file (*.dat).

Step 3: Create geometry and mesh of 1D SG

Choose this file in the dialog box: 1D Structure Genome as shown in Figure 2-15.

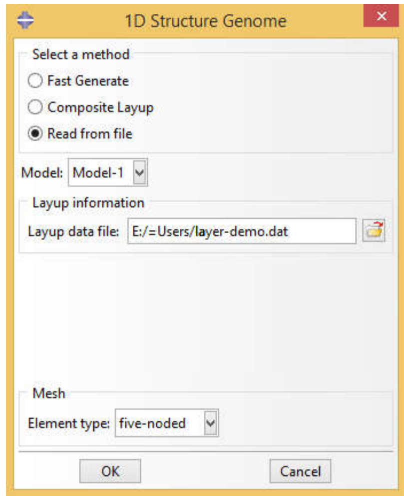


Figure 2-15: The 1D SG created (numbers in orange are element numbers).

Then the 1D SG will be created, of which the composite layup information can be checked in the Composite Layup Dialog box as shown in the Figure 2-16.

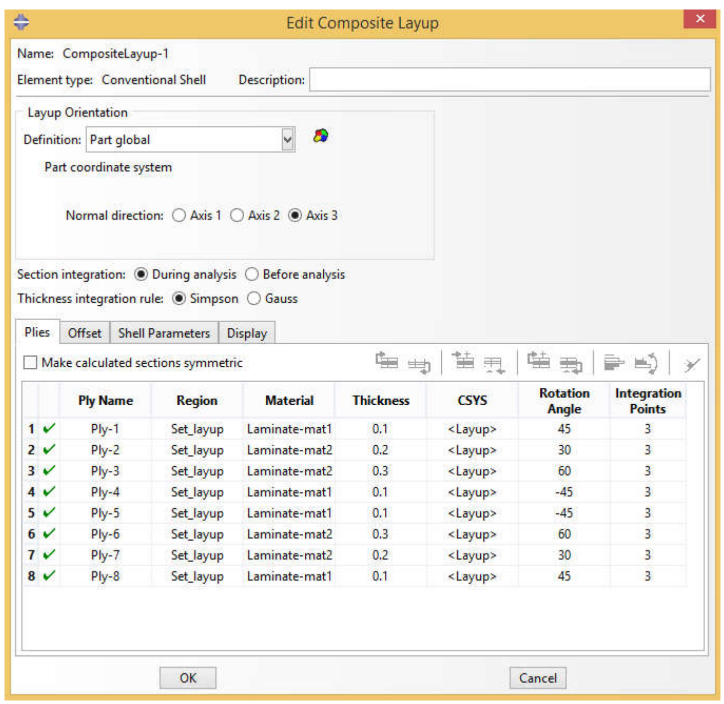


Figure 2-16: The composite layup of the1D SG created.

2.1.2 Homogenization

We will use the model generated by fast generate to show the results of Homogenization and Dehomogenization.

Click the homogenization button in the red frame of Figure 2-17, a dialog box in Figure 2-18 will pop up.

In this example, if we choose 3D (solid) for the macro model dimension, then the default file name will be Laminate_nSG1_3D_n5.sc, where 3D stands for 3D solid model.

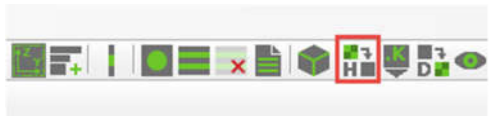


Figure 2-17: Homogenization button.

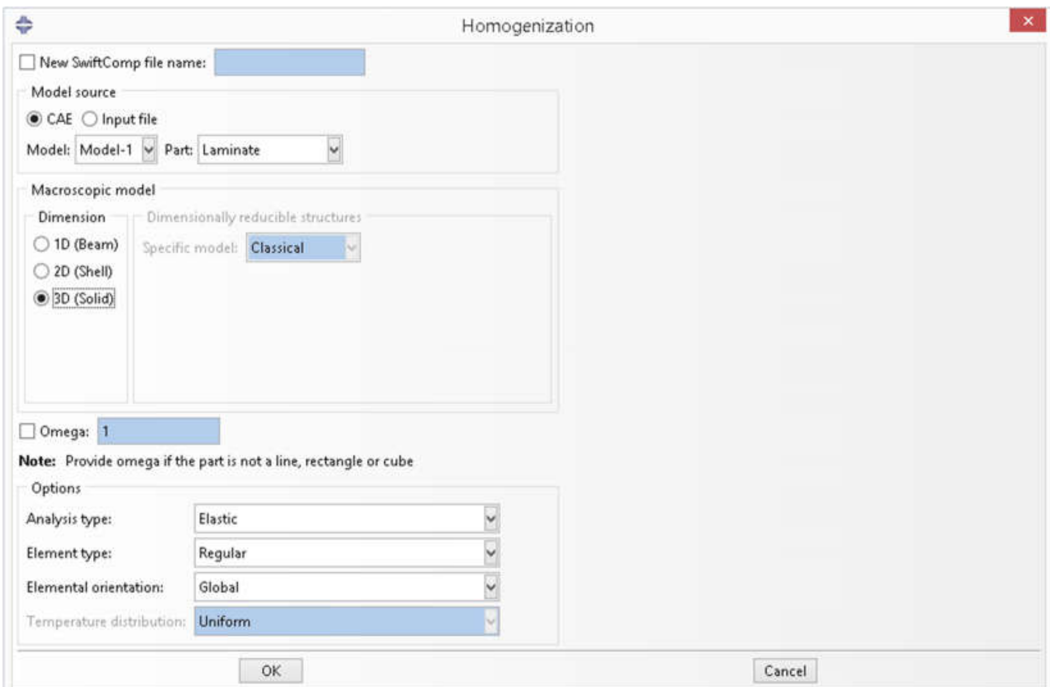


Figure 2-18: Homogenization dialog box for 3D solid model.

Click OK and wait for preparing the input file of SwiftComp™ and homogenization.

After the computation of SwiftComp™, the effective properties will pop up automatically (Figure 2-19).

```
The Effective Stiffness Matrix
-------------------------------------------
  3.9150122E+01  3.1350122E+01  5.3741674E+00  0.0000000E+00  0.0000000E+00  2.2105491E-15
  3.1350122E+01  3.9150122E+01  5.3741674E+00  0.0000000E+00  0.0000000E+00  3.8758836E-15
  5.3741674E+00  5.3741674E+00  1.1583814E+01  0.0000000E+00  0.0000000E+00 -6.4468002E-17
  0.0000000E+00  0.0000000E+00  0.0000000E+00  3.2799406E+00 -6.5926540E-17  0.0000000E+00
  0.0000000E+00  0.0000000E+00  0.0000000E+00 -6.5926540E-17  3.2799406E+00  0.0000000E+00
  2.2105491E-15  3.8758836E-15 -6.4468002E-17  0.0000000E+00  0.0000000E+00  3.0369255E+01

The Engineering Constants (Approximated as Orthotropic)
-------------------------------------------------------
 E1   =   1.3940283E+01
 E2   =   1.3940283E+01
 E3   =   1.0764479E+01
 G12  =   3.0369255E+01
 G13  =   3.2799406E+00
 G23  =   3.2799406E+00
 nu12=    7.8721572E-01
 nu13=    9.8718636E-02
 nu23=    9.8718636E-02

The Effective Compliance Matrix
-------------------------------------------
  7.1734557E-02 -5.6470571E-02 -7.0815376E-03  0.0000000E+00  0.0000000E+00  1.9705478E-18
 -5.6470571E-02  7.1734557E-02 -7.0815376E-03  0.0000000E+00  0.0000000E+00 -5.0597342E-18
 -7.0815376E-03 -7.0815376E-03  9.2898135E-02  0.0000000E+00  0.0000000E+00  1.6164459E-18
  0.0000000E+00  0.0000000E+00  0.0000000E+00  3.0488357E-01  6.1281352E-18  0.0000000E+00
  0.0000000E+00  0.0000000E+00  0.0000000E+00  6.1281352E-18  3.0488357E-01  0.0000000E+00
  1.9705478E-18 -5.0597342E-18  1.6164459E-18  0.0000000E+00  0.0000000E+00  3.2928038E-02


Effective Density =    1.0000000E-01
```

Figure 2-19: Effective properties of 3D solid model.

In the homogenization, the basic information of the SG model as shown in Figure 2-20 have been saved into sg model, which can be accessed by python script using the path 'mdb.customData.sgs[sg_name]', where 'sg_name' is the SwiftComp file name in homogenization.

```
---> Create sg model: Laminate_nSG1_3D_n5
     mdb.customData.sgs['Laminate_nSG1_3D_n5']
({'analysis': 0,
  'elem_flag': 0,
  'macro_model_dimension': '3D',
  'model_name': 'Model-1',
  'modelfromSource': 'fromCAE',
  'nSG': 1,
  'name': 'Laminate_nSG1_3D_n5',
  'part_name': 'Laminate',
  'swiftcomp_filename': 'Laminate_nSG1_3D_n5.sc',
  'temp_flag': 0,
  'trans_flag': 0,
  'volume': 0.8})
--------------------------------
```

Figure 2-20: SG model data created after homogenization.

If we choose plate/shell model (Figure 2-21), the default file name will be Laminate_nSG1_2D_n5.sc where 2D stands for a plate/shell model (2D). After homogenization, the ABD matrix of the homogenized plate/shell will be obtained (Figure 2-22). The offset ratio will influence the results of plate/shell model as it effectively sets the reference surface at different locations. Note 1D SG cannot be used for beam model.

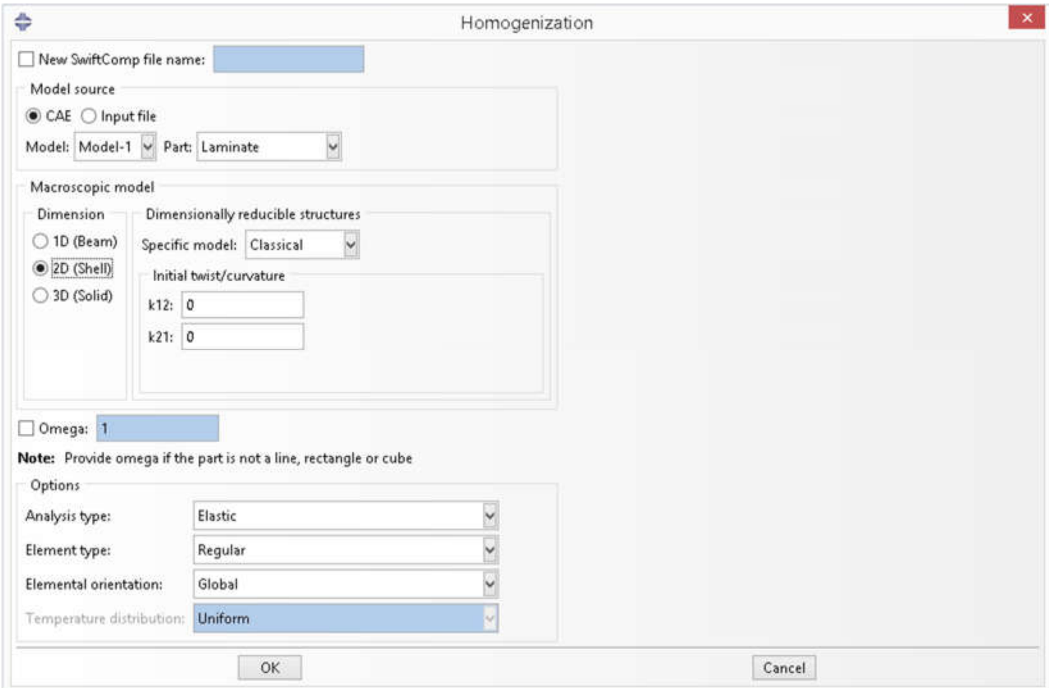


Figure 2-21: Homogenization dialog box-use 2D shell model.

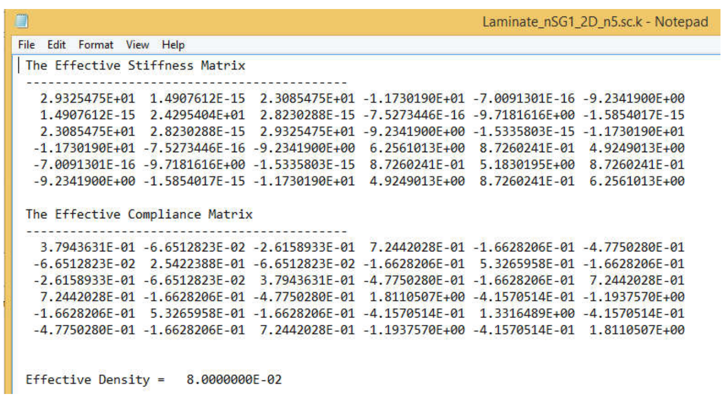


Figure 2-22: ABD matrix of classical plate/shell model.

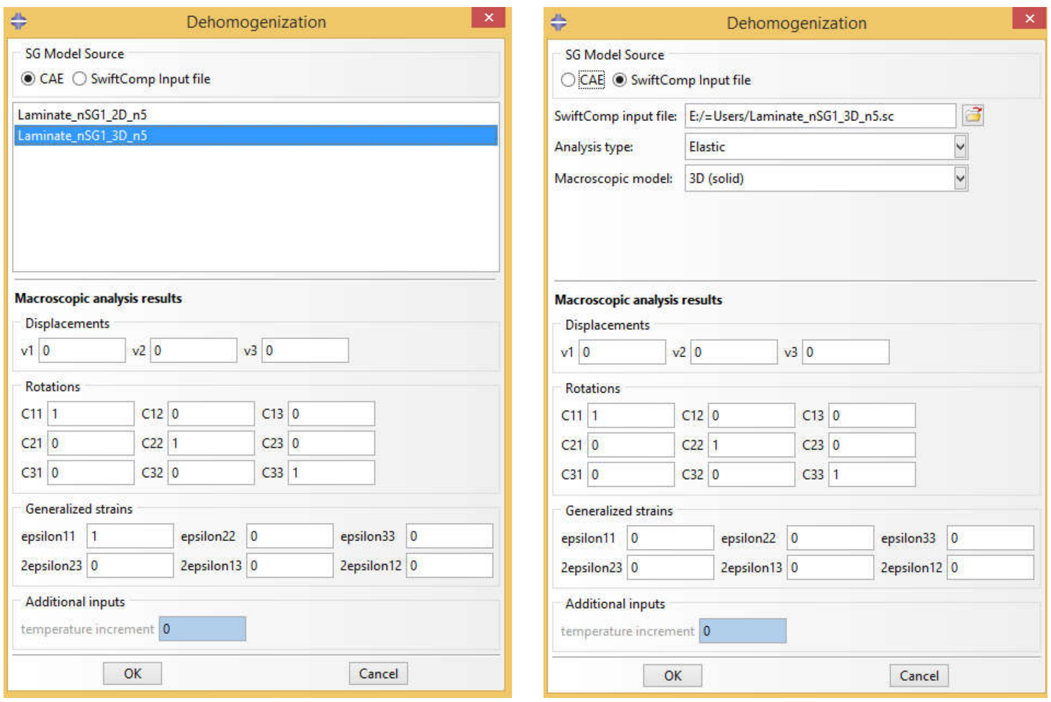
2.1.3 Dehomogenization

Click the dehomogenization button in the red frame of Figure 2-23, the dehomogenization dialog box in Figure 2-24 (a) will pop out.

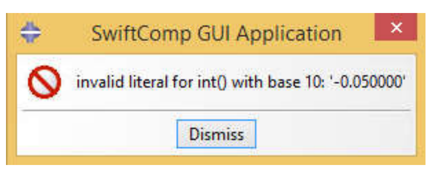
Figure 2-23: Dehomogenization button.

There are two methods provided to choose the SG we have created previously. The first method is to choose the sg_name shown in the list while the SG model source is ‘CAE’. The second method is to choose the SwiftComp input file 'Laminate_nSG1_3D_n5.sc' if the SG model source is 'SwiftComp input file'. Using this method, the Analysis type and the Macroscopic model type must be specified correctly, otherwise there will be an error message popped up (Figure 2-24 (b)). For both methods, the SwiftComp input file (.sc), the homogenization result files (.sc.k and .sc.opt) should be in the current work directory, which are needed in the dehomogenization. If you used CAE to generate the model, carried out the homogenization, the files are already stored in the work directory.

Specify the required inputs as shown in Figure 2-24. Please refer to SwiftComp™ manual for meaning of the global behavior parameters.



(a) Two methods to choose the homogenizaiton results files (.sc, .sc.k, .sc.opt)



(b) Error message if choose the wrong macroscopic model type

Figure 2-24: Dehomogenization parameters for 3D solid model.

Click OK and wait for SwiftComp™ to finish the computation. The post-processing results will be automatically loaded. However the user need to switch to the Visualization model to view all the field results components, including displacement (U1, U2, U3) and its magnitude, six nodal strain components (EN11, EN22, EN33, 2EN23, 2EN13, 2EN12) and the derived quantities such as Mises strain, six nodal stress components (SN11, SN22, SN33, SN23, SN13, SN12) and the derived quantities such as Mises stress. The nodal stress SN12 components is shown in Figure

2-25. In the odb tree, sections 'nlayer - 1', 'nlayer - 2' are created, with each section containing the plies with the same material properties and the same layer angle.

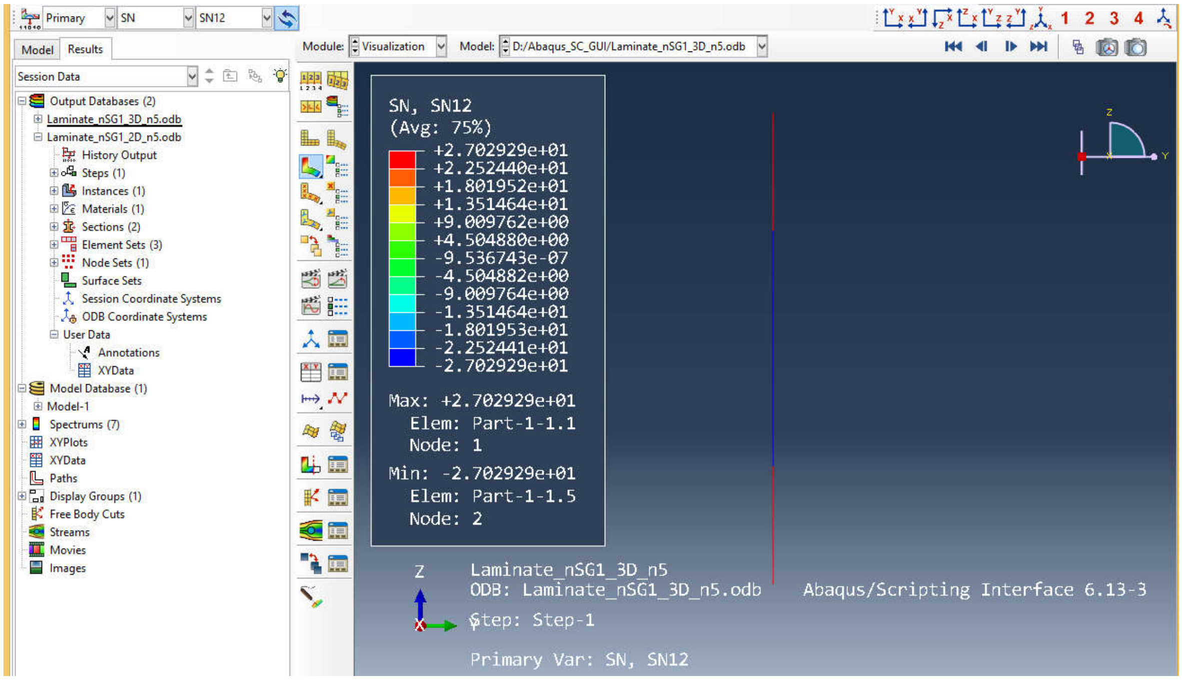


Figure 2-25: Dehomogenization results of 3D model.

If using shell model, choose the SwiftComp input file 'Laminate_nSG1_2D_n5.sc' and specify the required inputs as shown in Figure 2-26. Please refer to SwiftComp™ manual for meaning of the global behavior parameters.

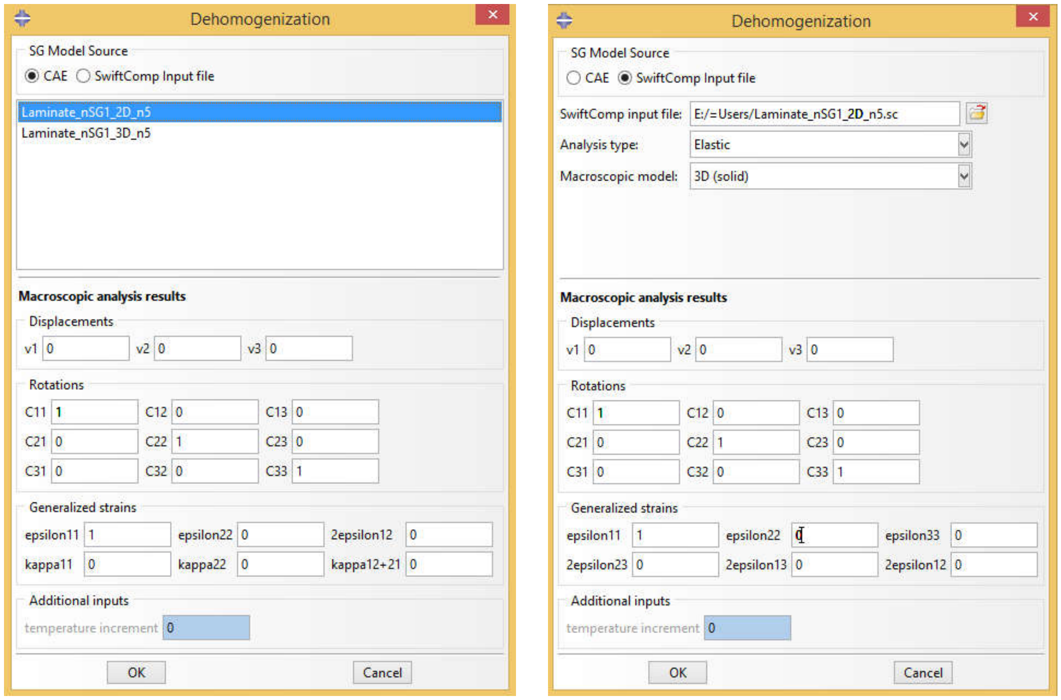


Figure 2-26: Dehomogenization parameters for shell model.

Contour plots are available for all local fields. In the odb tree, sections 'nlayer - 1', 'nlayer - 2' are created, with each section containing the plies with the same material properties and the same layer angle. The nodal stress SN12 component is shown in Figure 2-27.

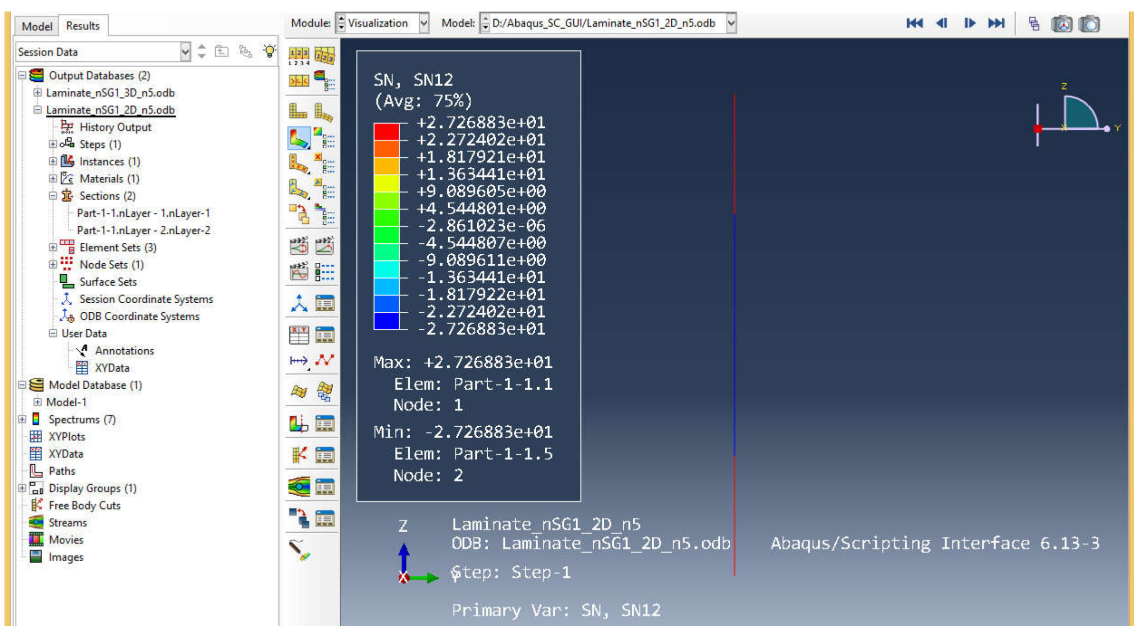


Figure 2-27: Dehomogenization results of shell model.

2.2 Square Pack Microstructure (2D SG)

2.2.1 2D SG preparation

In this example, assume that both fiber and matrix are isotropic materials (Material Fiber: Young's modulus E = 379.3 GPa, Poisson's ratio $\nu = 0.1$; Material Matrix: Young's modulus E = 68.3 GPa, Poisson's ratio $\nu = 0.3$). Create Materials in the Property module as shown in Figure 2-28.

Material	E	ν
Fiber	379.3 GPa	0.1
Matrix	68.3 GPa	0.3

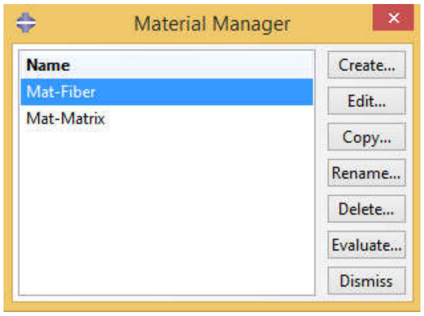


Figure 2-28: Material creation.

Then click the 2D SG button in the red frame in Figure 2-29 so that the dialog box in Figure 2-30 will pop out. The square pack microstructure SG has width 1.0. The user is allowed to choose the volume fraction or radius of fiber and interphase. If there is no interphase, the text field in the Interface group box is set to the default value 0.0.

Figure 2-29: Create 2D common SG button.

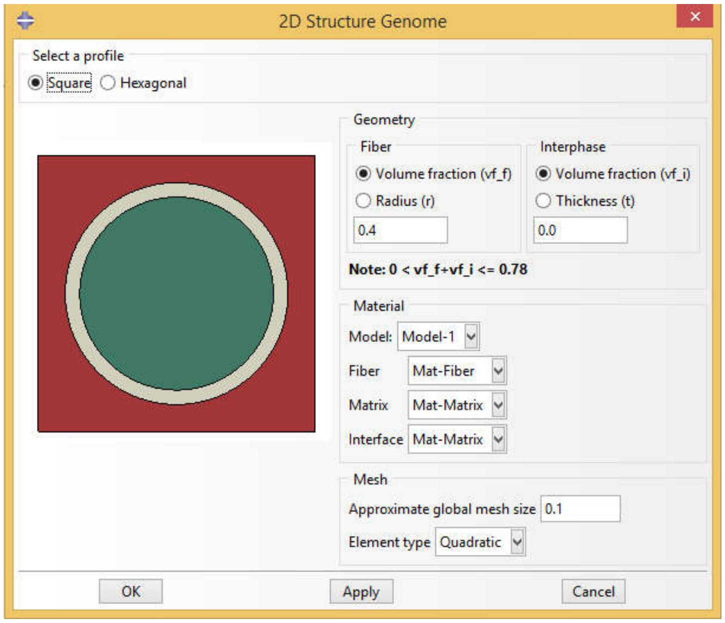


Figure 2-30: Create 2D SG dialog box.

After setting all the parameters as shown in Figure 2-30, click OK to create the SG (Figure 2-31).

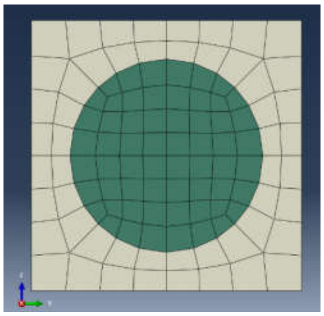


Figure 2-31: 2D SG.

2.2.2 Homogenization

Click the homogenization button in the red frame of Figure 2-32, a dialog box in Figure 2-33 will pop up. In this example, we choose the 3D solid macro model, then the default file name will be 'sqrP2_nSG2_3D_S8R.sc'. Click OK and wait for preparing the input file of SwiftComp™ and homogenization. After the computation of SwiftComp™, the effective properties will pop up automatically (Figure 2-34).

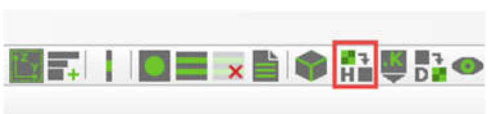


Figure 2-32: Homogenization button.

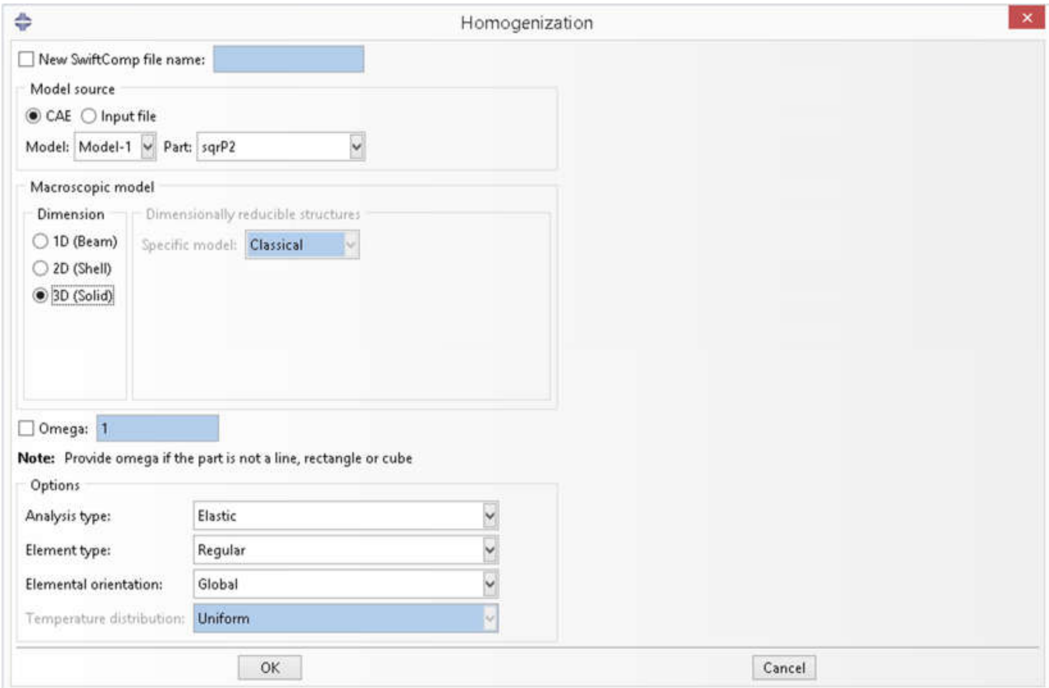
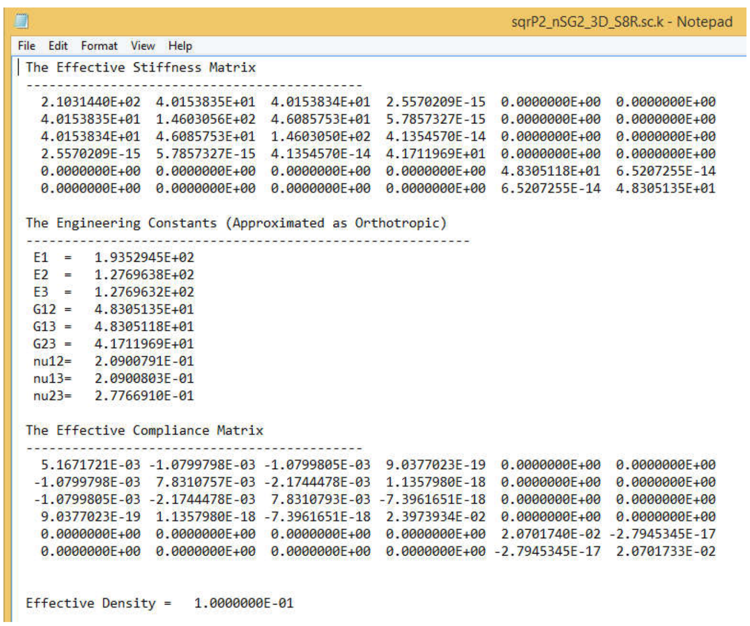


Figure 2-33: Homogenization dialog box-use 3D solid model.




```
The Effective Stiffness Matrix
-------------------------------------------
  2.1031440E+02  4.0153835E+01  4.0153834E+01  2.5570209E-15  0.0000000E+00  0.0000000E+00
  4.0153835E+01  1.4603056E+02  4.6085753E+01  5.7857327E-15  0.0000000E+00  0.0000000E+00
  4.0153834E+01  4.6085753E+01  1.4603050E+02  4.1354570E-14  0.0000000E+00  0.0000000E+00
  2.5570209E-15  5.7857327E-15  4.1354570E-14  4.1711969E+01  0.0000000E+00  0.0000000E+00
  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  4.8305118E+01  6.5207255E-14
  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  6.5207255E-14  4.8305135E+01

The Engineering Constants (Approximated as Orthotropic)
---------------------------------------------------------
 E1   =    1.9352945E+02
 E2   =    1.2769638E+02
 E3   =    1.2769632E+02
 G12  =    4.8305135E+01
 G13  =    4.8305118E+01
 G23  =    4.1711969E+01
 nu12=    2.0900791E-01
 nu13=    2.0900803E-01
 nu23=    2.7766910E-01

The Effective Compliance Matrix
-------------------------------------------
  5.1671721E-03 -1.0799798E-03 -1.0799805E-03  9.0377023E-19  0.0000000E+00  0.0000000E+00
 -1.0799798E-03  7.8310757E-03 -2.1744478E-03  1.1357980E-18  0.0000000E+00  0.0000000E+00
 -1.0799805E-03 -2.1744478E-03  7.8310793E-03 -7.3961651E-18  0.0000000E+00  0.0000000E+00
  9.0377023E-19  1.1357980E-18 -7.3961651E-18  2.3973934E-02  0.0000000E+00  0.0000000E+00
  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  2.0701740E-02 -2.7945345E-17
  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00 -2.7945345E-17  2.0701733E-02


Effective Density =   1.0000000E-01
```

Figure 2-34: Effective properties of 3D solid model.

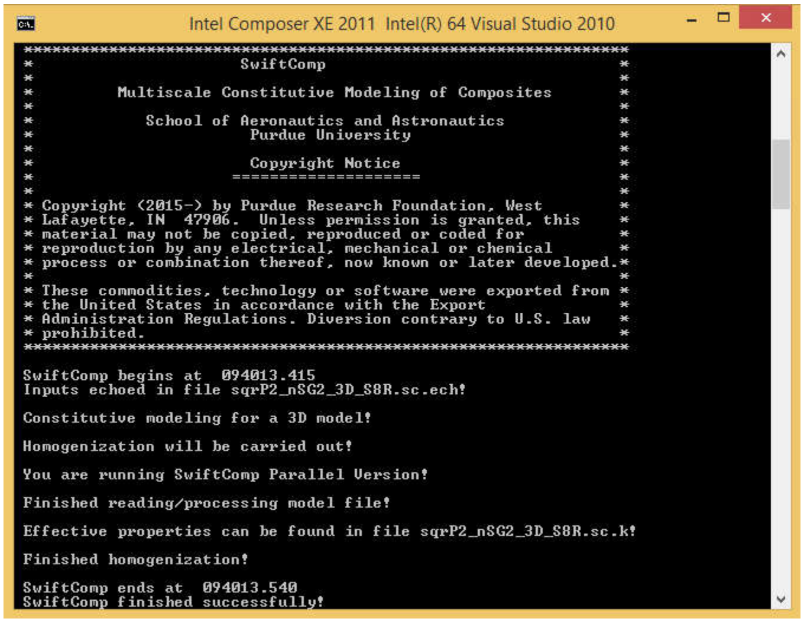


Figure 2-35: Command line window shows the progress and error messages.

2.2.3 Dehomogenization

Click the dehomogenization button in the red frame of Figure 2-36, the dehomogenization dialog box in Figure 2-37 will pop out. We can choose the SG from CAE or choose SwiftComp input file 'sqrP2_nSG2_3D_S8R.sc', and then specify the required inputs as shown in Figure 2-37.

Figure 2-36: Dehomogenization button.

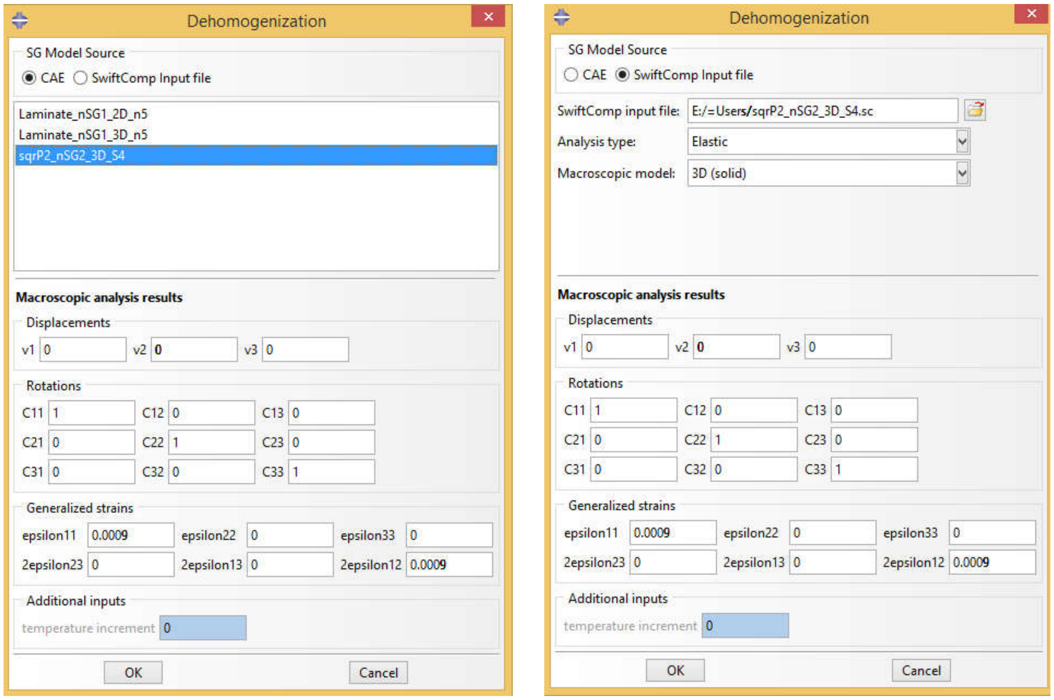


Figure 2-37: Dehomogenization parameters for 3D solid model.

Click OK and wait for SwiftComp™ to finish the computation. Switch to the Visualization model, the user is able to view all the field results components. In the odb tree, 2 sections are created, with each section containing the same material properties. The nodal stress SN11 component is shown in Figure 2-38.

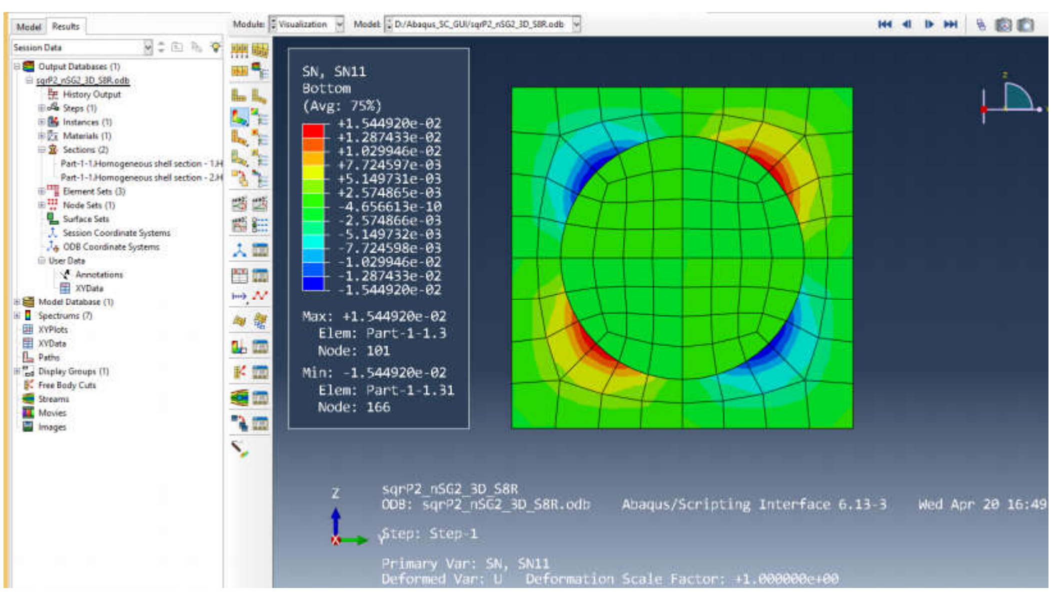


Figure 2-38: Dehomogenization results of 3D model.

2.3 Spherical Inclusion Microstructure with Interphase (3D SG)

2.3.1 3D SG preparation

In this example, assume that both fiber and matrix are isotropic materials (Material Fiber: Young's modulus E = 379.3 GPa, Poisson's ratio $\nu = 0.1$; Material Matrix: Young's modulus E = 68.3 GPa, Poisson's ratio $\nu = 0.3$; Material Interphase: E_1=117.0 GPa, E_2=E_3=8.54 GPa, $\nu_{12} = \nu_{13} = 0.278$, $\nu_{23} = 0.5$, G_{12}=G_{13}=3.9 GPa, G_{23}=2.83 GPa).

Create Materials in the Property module as shown in Figure 2-39.

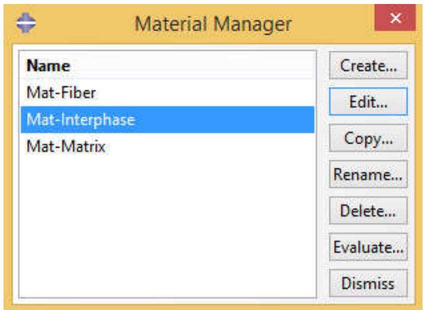


Figure 2-39: Material creation.

Then click the 3D common SG button in the red frame in Figure 2-40 so that the dialog box in the Figure 2-41 will pop out. This 3D SG has length 1.0 in every dimension. The user is allowed to choose the volume fraction or radius of inclusion and interphase. If there is no interphase, the text field in the Interface group box is set to be the default value 0.0.

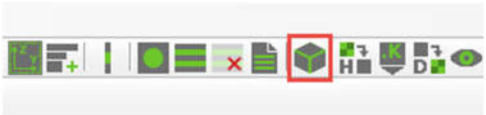


Figure 2-40: Create 3D common SG button.

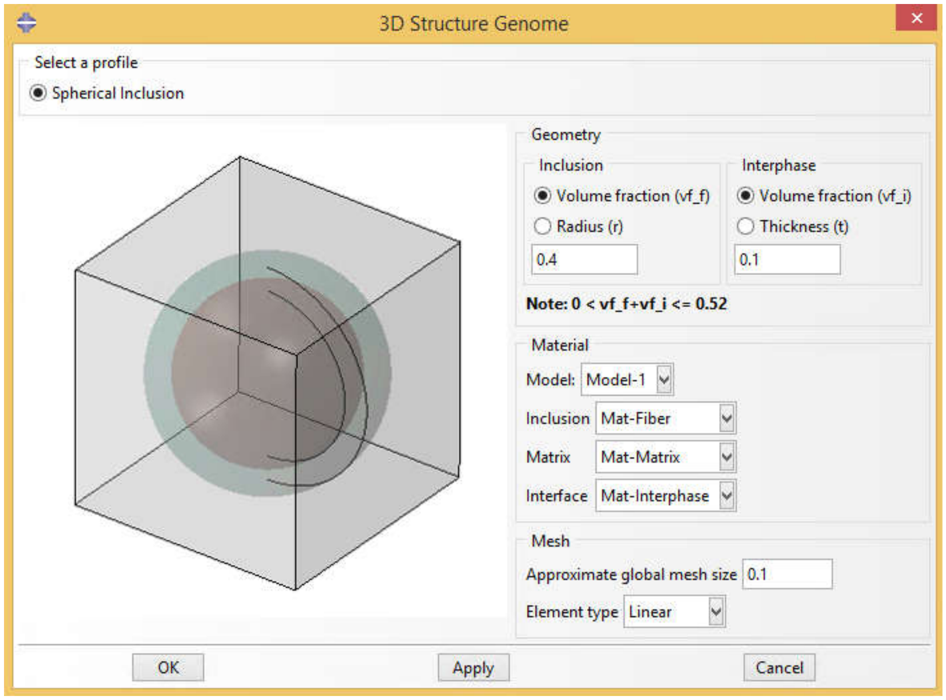


Figure 2-41: Create 3D SG dialog box.

After setting all the parameters as shown in Figure 2-41, click OK to create the SG (Figure 2-42).

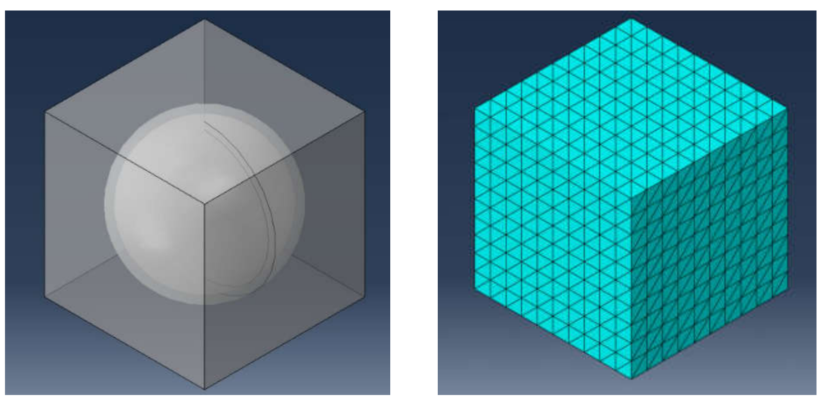


Figure 2-42: 3D SG.

2.3.2 *Homogenization*

Click the homogenization button, a dialog box in Figure 2-43 will pop up. In this example, we choose the 3D solid macro model, then the default file name will be 'inclusionP3_nSG3_3D_C3D4.sc'. Click OK and wait for preparing the input file of SwiftComp™ and homogenization. After the computation of SwiftComp™, the effective properties will pop up automatically (Figure 2-44).

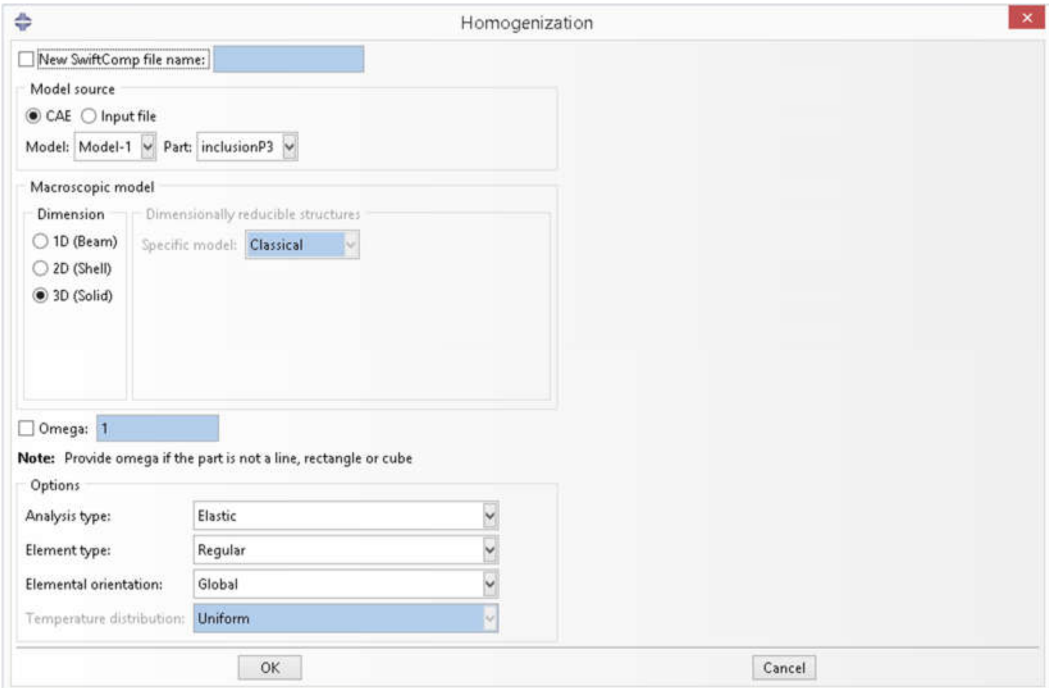


Figure 2-43: Homogenization dialog box-use 3D solid model.

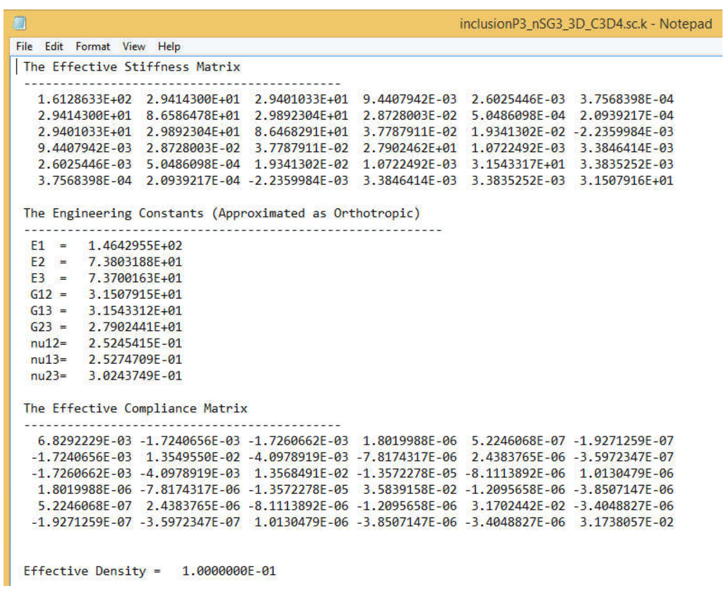


Figure 2-44: Effective properties of 3D solid model.

2.3.3 Dehomogenization

Click the dehomogenization button, the dehomogenization dialog box in Figure 2-45 will pop out. We can choose the SG from CAE or choose SwiftComp input file ‘inclusionP3_nSG3_3D_C3D4.sc’, and then specify the required inputs as shown in Figure 2-45.

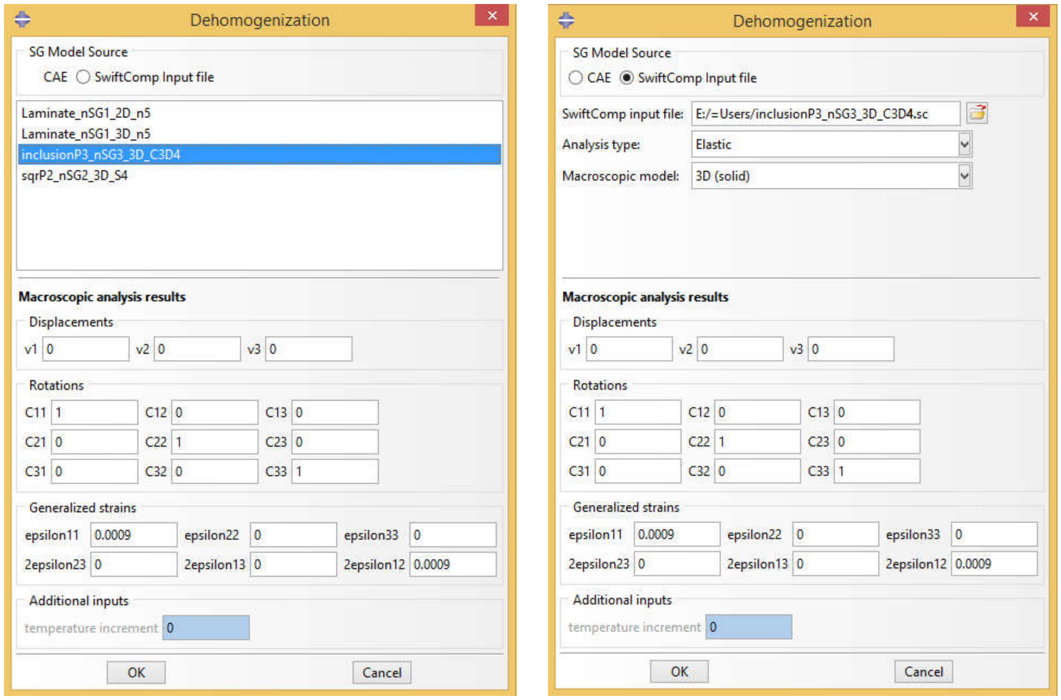


Figure 2-45: Dehomogenization parameters for 3D solid model.

Click OK and wait for SwiftComp™ to finish the computation. Switch to the Visualization model, the user is able to view all the field results. In the odb tree, 3 sections are created, with each section containing the same material properties.

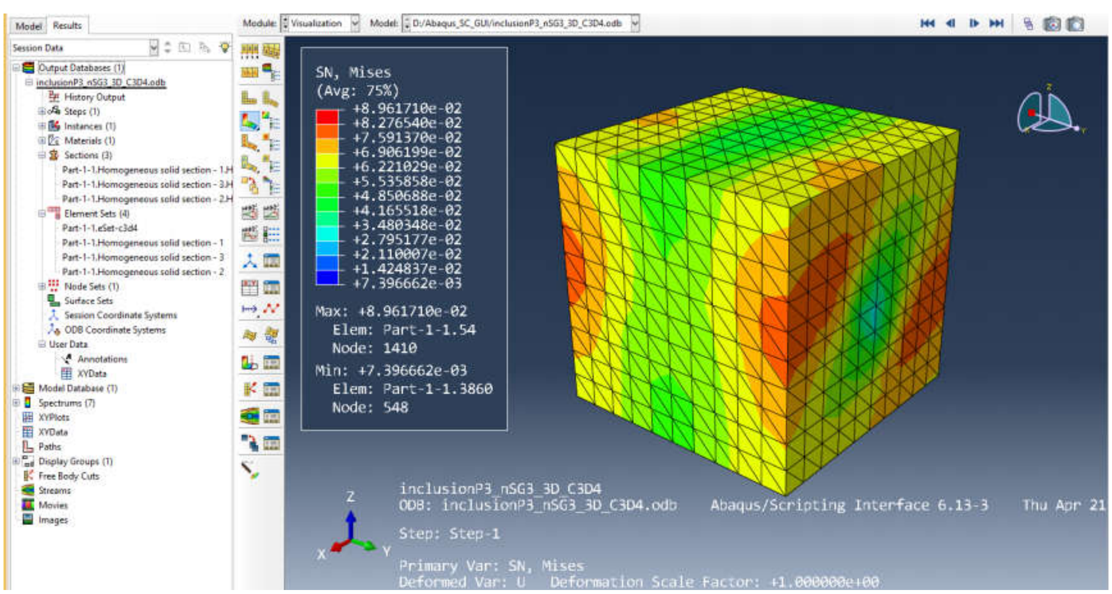


Figure 2-46: Dehomogenization results of 3D model.

2.4 Summary

Common SG function provides users a convenient way to quickly build the geometry of common SG, and easily specify material properties they want to test. Users can also test other capabilities of common SG function if they are interested.

3 CREATE USER-DEFINED MODEL

Although common SG function provides users a convenient way to build the geometry of the most common models, often users need to build their own models according to the microstructure they are analyzing. Without using the common SG function, this chapter will illustrate how to build 2D square pack microstructure SGs similar to the above common models. Example is not given for 3D SG since the process is exactly the same as creating a 3D part in Abaqus with information of elements, materials and section assignment. The only difference between user-defined model and common SG model is the generation of the SG. The homogenization and dehomogenization steps are the same. So we will only focus on the SG preparation in this chapter.

3.1 Square Pack Microstructure (2D SG)

3.1.1 Step 1: Create materials

To create a customized 2D SG, the first step is to create the materials in Abaqus GUI. For simplicity, this example uses the material properties in Section 2.3.

3.1.2 Step 2: Set work plane

As stated in Section 1.2.1 Geometry, 2D SG must in Y-Z plane. To set work plane to the Y-Z plane, click the work plane button in the red frame shown in Figure 3-1. In the dialog box popped out in Figure 3-2, choose the model and give the part name for the 2D SG, choose the SG dimension to be 2D. The work plane button will create a datum plane and a datum axis in the part.

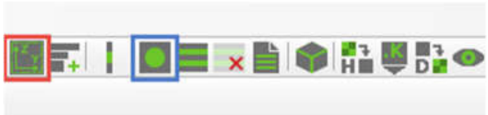


Figure 3-1: Work plane button (red box) and 2D SG button (blue box).

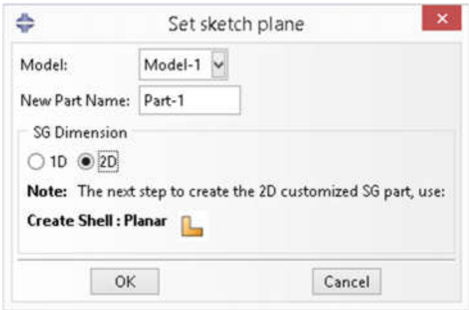


Figure 3-2: Work plane button dialog box.

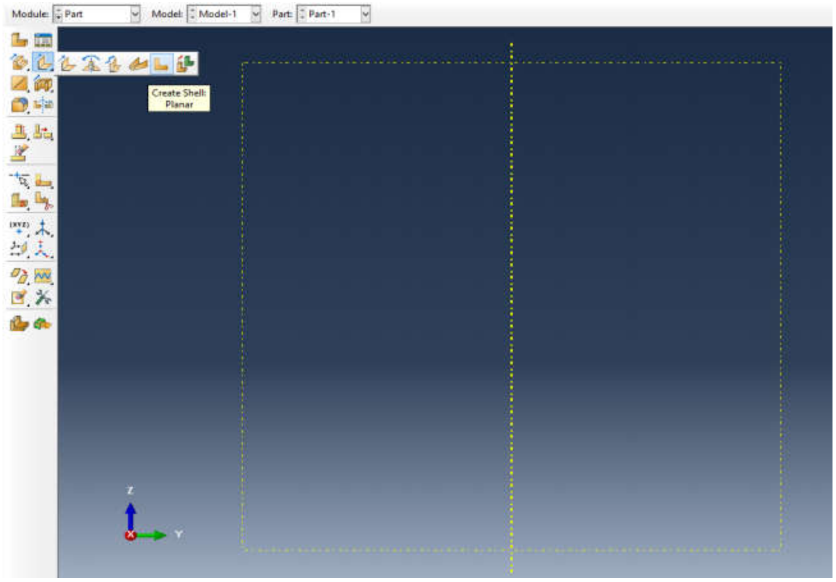
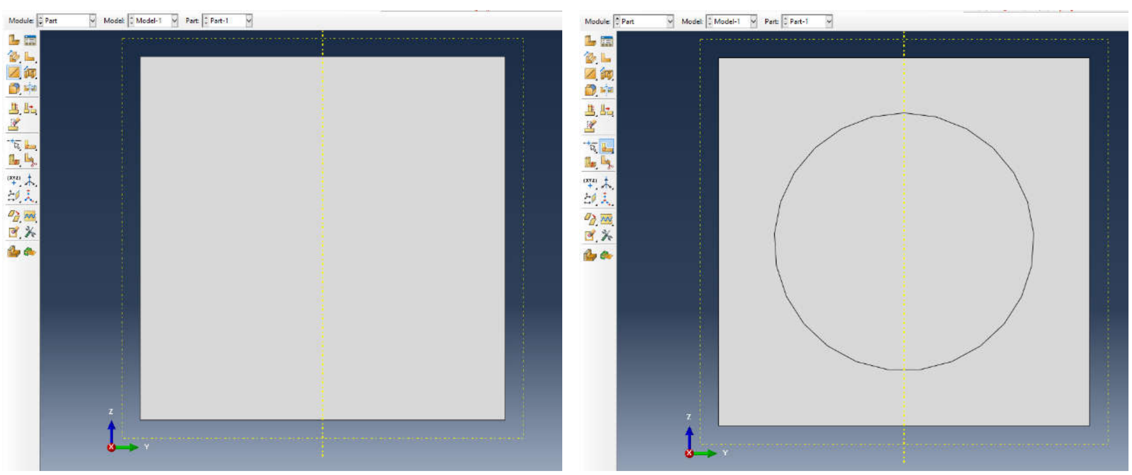


Figure 3-3: After use the work plane button.

3.1.3 Step 3: Create geometry of 2D customized SG

Follow the tips given in the work plane button dialog box, the next step is to create the 2D SG geometry using tool button 'Create Shell: Planar' in Part module. Follow the procedure, the user can create the geometry shown in Figure 3-4 (a). By Partition Face, the user can create the square pack microstructure as shown in Figure 3-4 (b).



(a) (b)

Figure 3-4: The square pack microstructure.

3.1.4 Step 4: Assign material sections, create mesh for the 2D customized SG

The user can then create material sections, assign material sections to the geometry and create mesh for the 2D customized SG, which are exactly the same as what one needs to do for a general Abaqus model. Please refer to Section 1.2 for the general remarks.

In this example, *Global seeds* are applied, and the mesh is generated using the *Mesh Controls setting*: *Quad-dominated Element Shape, Free-Technique, Advanced front–use mapped meshing where appropriate Algorithm*. The mesh shown in the Figure 3-5 is quite irregular but can be used for homogenization and dehomogenization analysis. However, if use *Medial axis Algorithm* while keep the other settings, although the mesh generated looks better (Figure 3-6), error will occur because of the mismatch of the periodic node pairs at the SG boundary edges. The error message can be checked in the command window as shown in Figure 3-7.

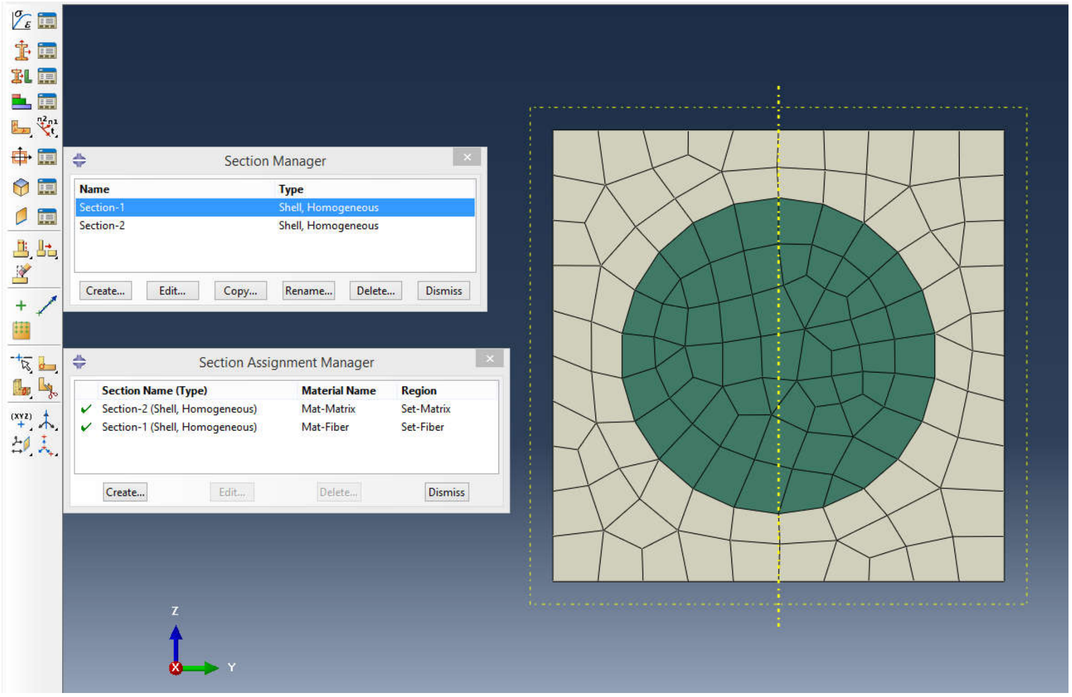


Figure 3-5: The square pack microstructure.

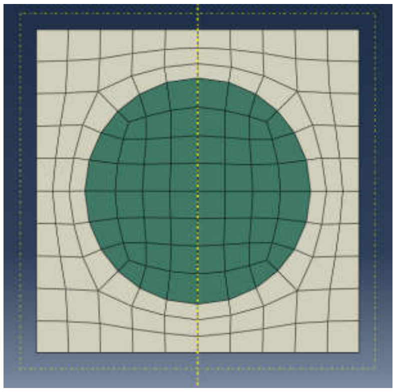


Figure 3-6: The square pack microstructure with mesh2- generated by 'Medial axis Algorithm'.

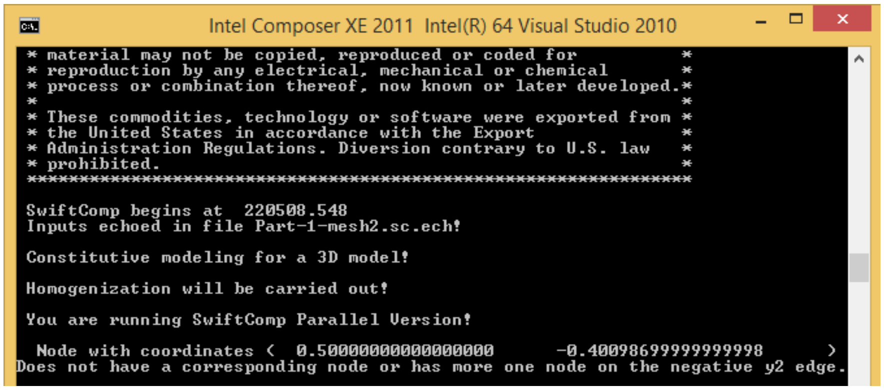


Figure 3-7: The error message when use mesh2.

Therefore the user should be skillful in generating good periodic mesh. Possible solutions can be:

1) Choose mesh techniques wisely and *do not allow the number of elements to change* (Figure 3-8).
2) For symmetrical SGs, the user can first create 1/4 of the geometry for 2D SG, or 1/8 SG for 3D SG, then use the tools in Assembly module (Patten, Rotate, Merge (mesh)) to create new part (Figure 3-9).
3) Using other software to generate mesh, for example Hypermesh, Ansys, which allow more flexible mesh techniques to create periodic mesh. Then the models created can be imported to Abaqus.

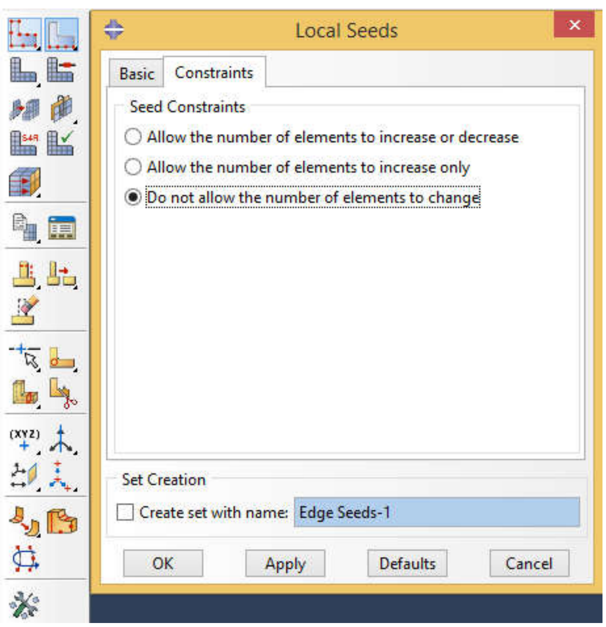


Figure 3-8: Seed by edge and constraint the element number

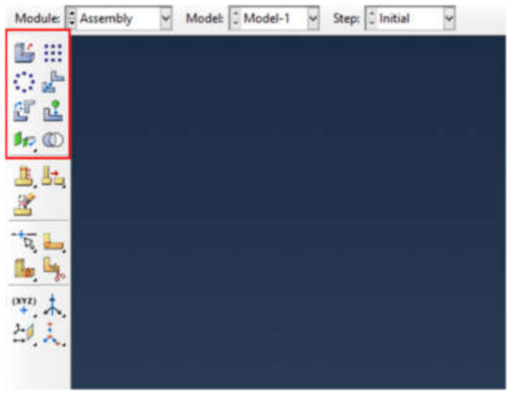


Figure 3-9: Tools in Assembly may help in generating periodic mesh

3.2 Arbitrary Shape Inclusions Microstructure (2D SG)

One more user-defined model is shown here, which is a rectangle SG with two arbitrary inclusions. Users can understand how to create complex shape in Abaqus-SwiftComp GUI, and also know the capability of SwiftComp™ to calculate such models.

3.2.1 Step 1: Create materials and sections

In this example, assume that both inclusions and matrix are isotropic materials (Material 1: E = 379.3 GPa, $\nu = 0.1$; Material 2: E = 279.3 GPa, $\nu = 0.1$; Material 3: E = 68.3 GPa, $\nu = 0.3$).

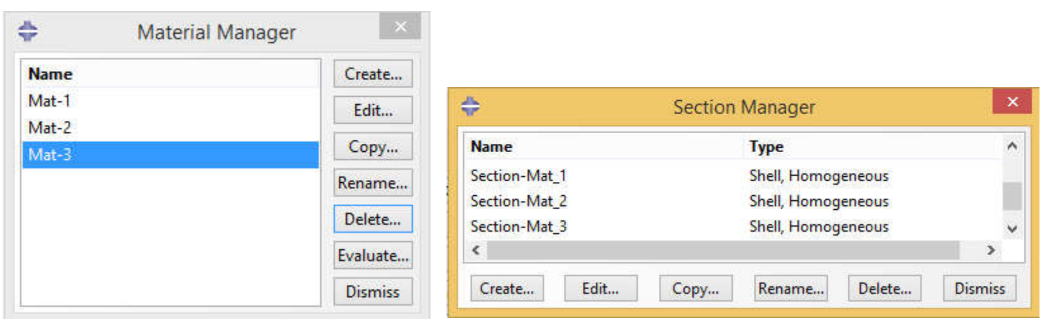


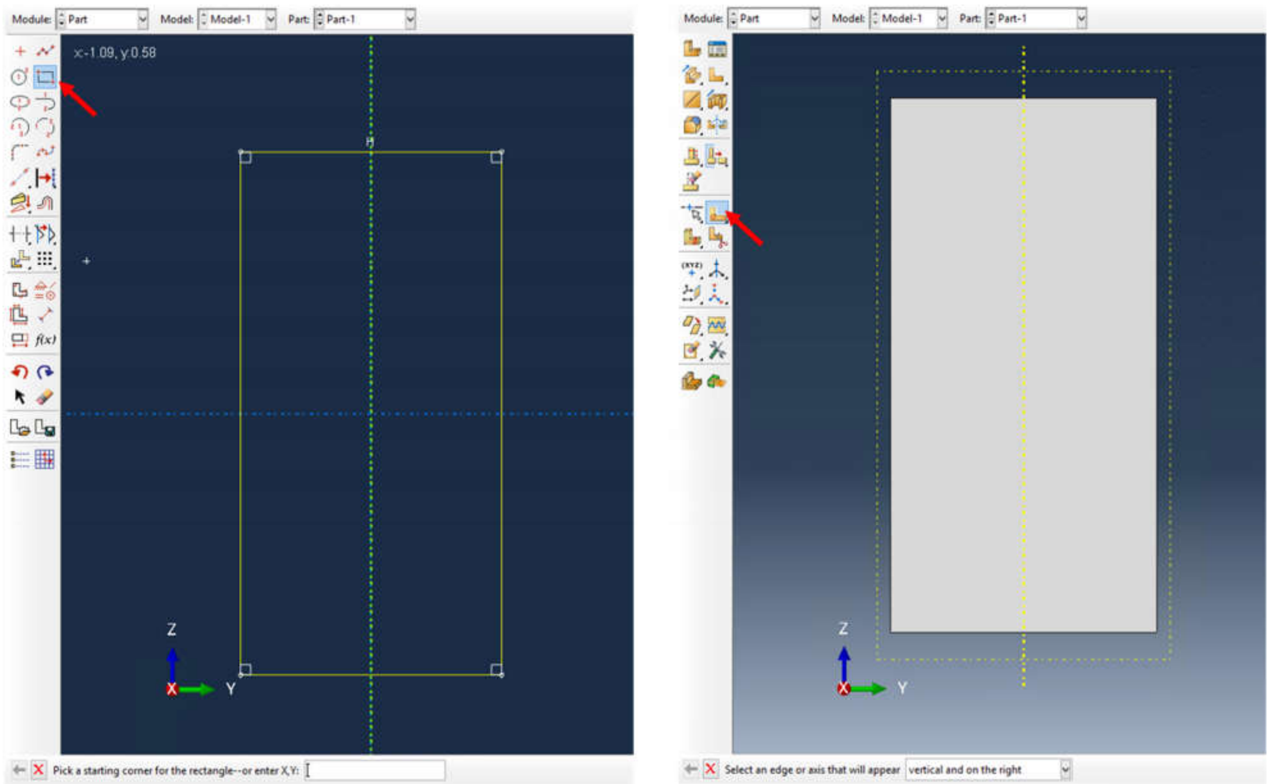
Figure 3-10: Create materials and sections.

3.2.2 Step 2: Set work plane

Following the procedure as shown in section 3.2 Step 2, the work plane of 2D SG can be created.

3.2.3 Step 3: Create geometry of 2D customized SG

First create the rectangular shell using points (-0.5, -1), (0.5, 1) as shown in Figure 3-11 (a).



(a) (b)

Figure 3-11: Create geometry of the SG.

Click the tool button ‘Partition face: Sketch’ (Figure 3-11 (b)), then create the isolated points (-0.2, 0.8, 0), (-0.3, 0.7, 0,), (-0.3, 0.4, 0), (-0.1, 0.6, 0), (0, 0.8, 0), (0, 0, 0), (-0.1, 0, 0), (-0.2, -0.3, 0), (0.1, -0.5, 0), (0.1, -0.3, 0) and (0.2, -0.1, 0) (see Figure 3-12 (a)). Create splines through the points as shown in Figure 3-12 (b).

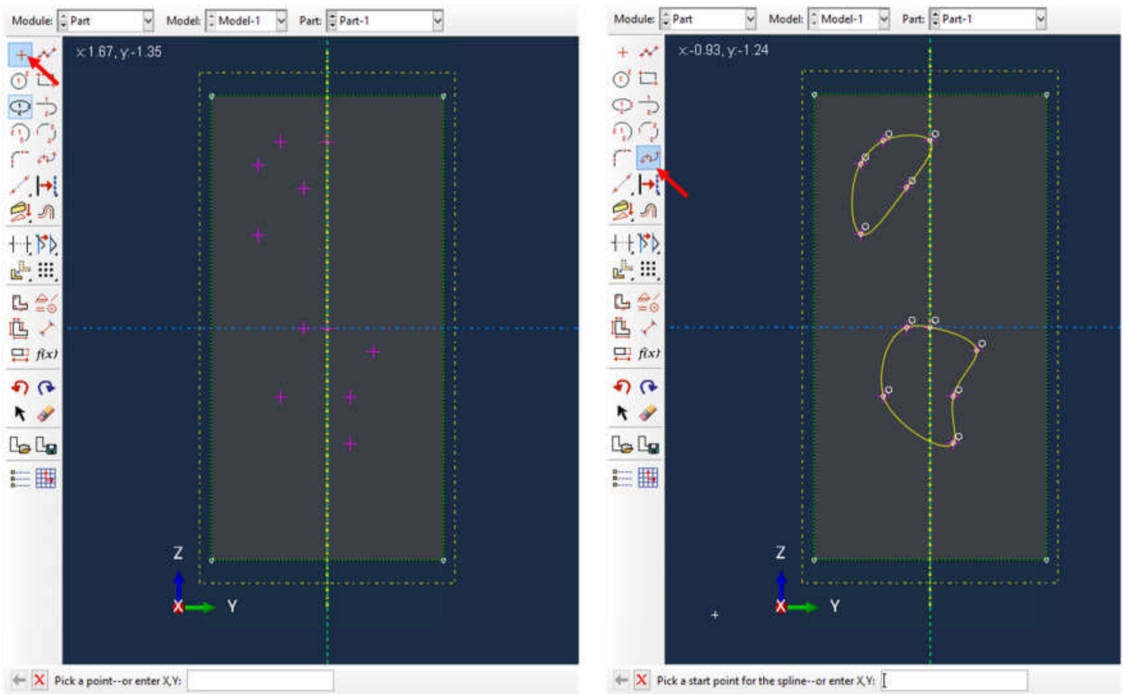


Figure 3-12

3.2.4 Step 4: Assign material sections, create mesh for the 2D customized SG

In the property module, the user can assign the material sections created in the first step to the geometry (Figure 3-13).

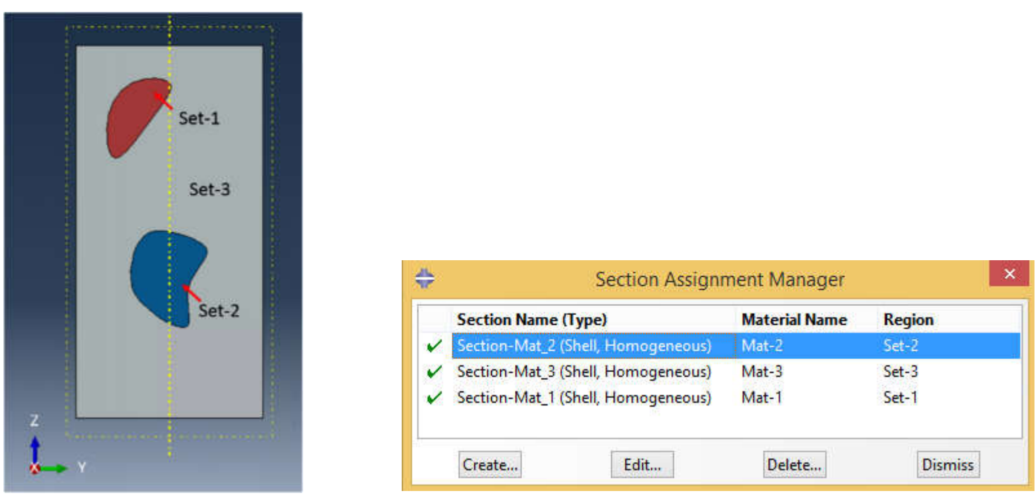


Figure 3-13: Assign material sections.

3.2.5 Step 5: Generate User-defined Model Mesh

In this example, Global seeds are applied, and the mesh is generated using the Mesh Controls setting: Quad-dominated Element Shape, Free-Technique, Advanced front–use mapped meshing where appropriate Algorithm. The mesh shown in the Figure 3-14 is quite irregular but can be used for homogenization analysis. However, it can be seen that there is one element (highlighted in the yellow ellipse is abnormal. If use this mesh to do the dehomogenization, there will be an error message in the command window (Figure 3-15). Such a mesh can be repaired as shown in Figure 3-16.

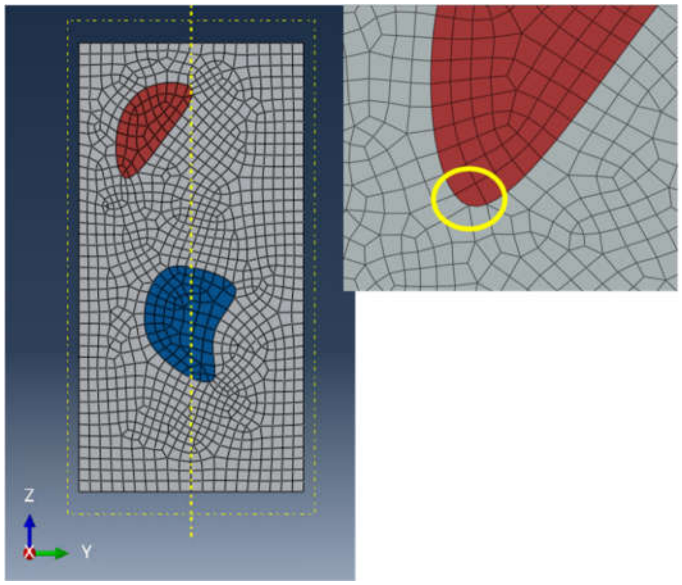


Figure 3-14: Mesh need to be repaired.

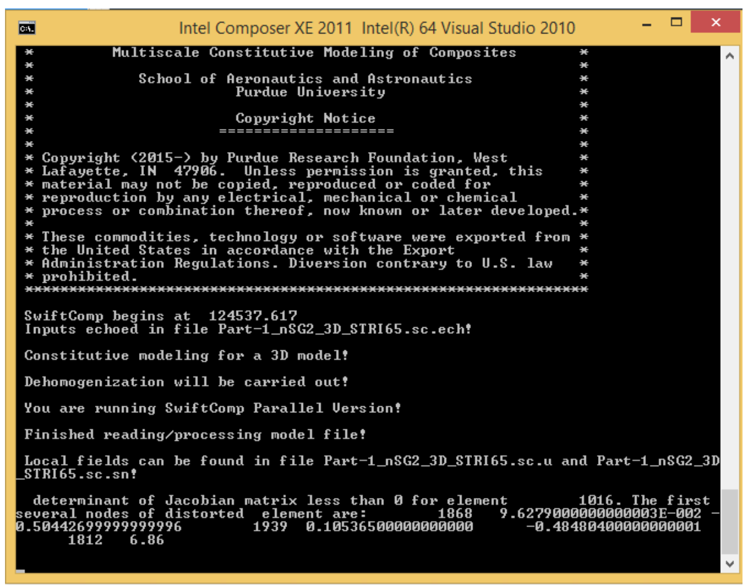


Figure 3-15: Error message in dehomogenization.

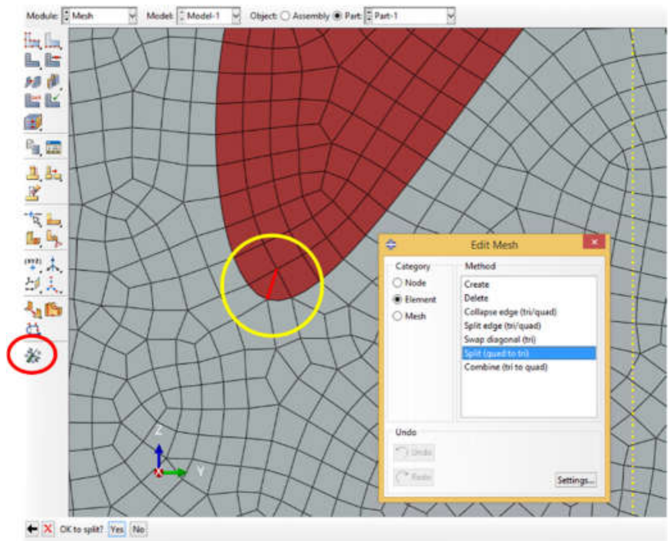


Figure 3-16: Repair the mesh.

3.2.6 *Step 6: Homogenization and dehomogenization*

Click the homogenization button, and fill the dialog box as shown in the Figure 3-17, the effective properties is obtained (Figure 3-18).

Click the dehomogenization button, and fill the dialog box as shown in the Figure 3-19, the local fields are obtained (Figure 3-20).

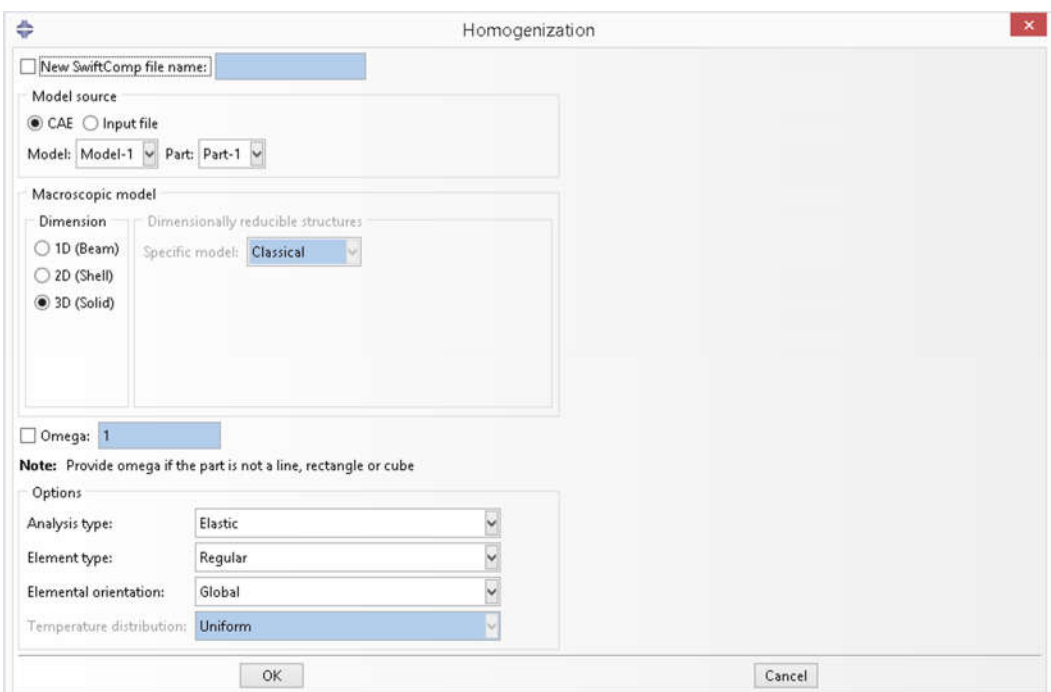
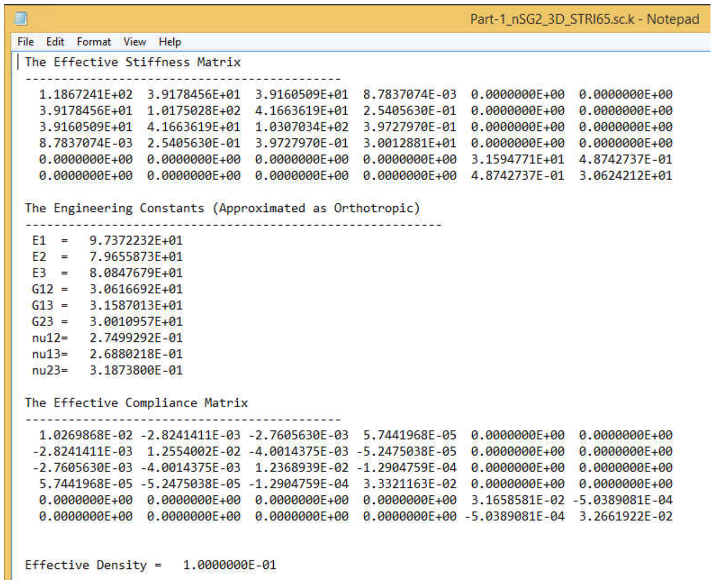


Figure 3-17: Homogenization.



```
The Effective Stiffness Matrix
-----------------------------------------------
  1.1867241E+02  3.9178456E+01  3.9160509E+01  8.7837074E-03  0.0000000E+00  0.0000000E+00
  3.9178456E+01  1.0175028E+02  4.1663619E+01  2.5405630E-01  0.0000000E+00  0.0000000E+00
  3.9160509E+01  4.1663619E+01  1.0307034E+02  3.9727970E-01  0.0000000E+00  0.0000000E+00
  8.7837074E-03  2.5405630E-01  3.9727970E-01  3.0012881E+01  0.0000000E+00  0.0000000E+00
  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  3.1594771E+01  4.8742737E-01
  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  4.8742737E-01  3.0624212E+01

The Engineering Constants (Approximated as Orthotropic)
-------------------------------------------------------------
 E1  =   9.7372232E+01
 E2  =   7.9655873E+01
 E3  =   8.0847679E+01
 G12 =   3.0616692E+01
 G13 =   3.1587013E+01
 G23 =   3.0010957E+01
 nu12=   2.7499292E-01
 nu13=   2.6880218E-01
 nu23=   3.1873800E-01

The Effective Compliance Matrix
-----------------------------------------------
  1.0269868E-02 -2.8241411E-03 -2.7605630E-03  5.7441968E-05  0.0000000E+00  0.0000000E+00
 -2.8241411E-03  1.2554002E-02 -4.0014375E-03 -5.2475038E-05  0.0000000E+00  0.0000000E+00
 -2.7605630E-03 -4.0014375E-03  1.2368939E-02 -1.2904759E-04  0.0000000E+00  0.0000000E+00
  5.7441968E-05 -5.2475038E-05 -1.2904759E-04  3.3321163E-02  0.0000000E+00  0.0000000E+00
  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  3.1658581E-02 -5.0389081E-04
  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00 -5.0389081E-04  3.2661922E-02


Effective Density =   1.0000000E-01
```

Figure 3-18: Effective properties.

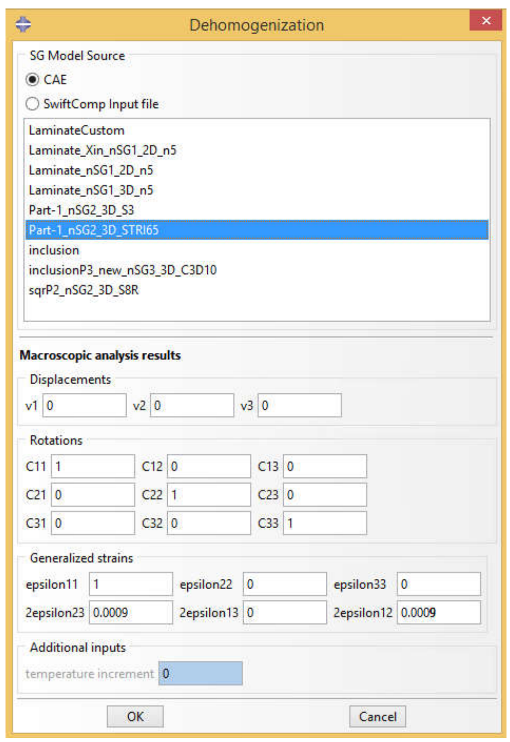


Figure 3-19: Dehomogenization.

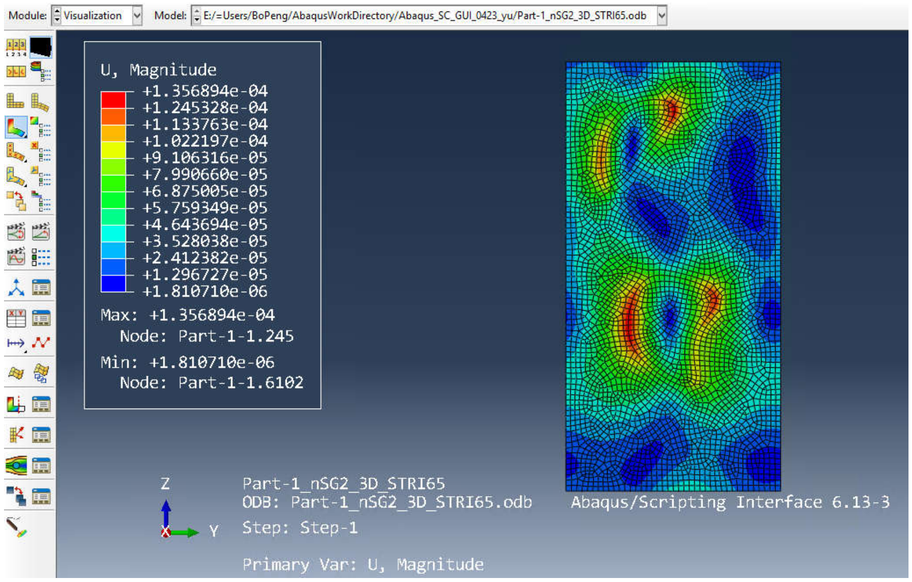


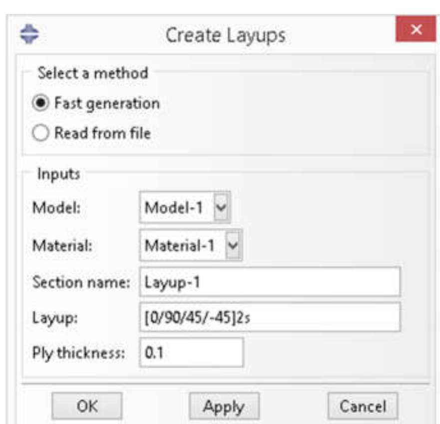
Figure 3-20: Dehomogenization result: magnitude of displacement.

3.3 Create Layups

Users can use function 'New layups'  to create *Solid-Composite* sections which will be used later. There are two ways to do this. One is through fast generation, the other one is reading from file.

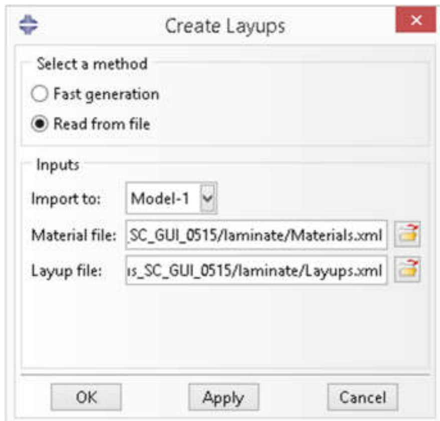
3.3.1 Fast generation

In this method, user can generate a layup through a rule, such as [0/90/45/-45]2s. User needs to first create a material. Then provide a new composite section name and the layup rule. The thickness at the last is for each ply. This method is only suitable for layups with a single material having different fiber orientations and all plies with the same thickness, which is the most common case in industry



3.3.2 Read from file

In this method, user needs to prepare a material file and a layup file. To create composite sections, select ‘Read from file’ in the dialog box, and choose the material file and the layup file, then click ‘OK’. Both the material file and layup file should use the XMLformat and details are described below. Users are encouraged to have some basic knowledge on XML files. A quick and simple tutorial can be found at http://www.tutorialspoint.com/xml/index.htm.



Material file

The root element is ‘<materials></materials>’ and each ‘<material></material>’ sub-element store one material. For the current version, we can only deal with materials with density and elastic properties.

```
<materials>
  <material type = "ENGINEERING CONSTANTS">
    <name>mat_1</name>
    <density>1.860000E+03</density>
    <e1>3.7000E+10</e1>
    <e2>9.0000E+09</e2>
    <e3>9.0000E+09</e3>
    <g12>4.0000E+09</g12>
    <g13>4.0000E+09</g13>
    <g23>4.0000E+09</g23>
    <nu12>0.28</nu12>
    <nu13>0.28</nu13>
    <nu23>0.28</nu23>
</material>
<material>
  ...
</material>
...
</materials>
```

Each material has a ‘type’ attribute, which has the same definition as Abaqus, ISOTROPIC, ENGINEERING CONSTANTS, ORTHOTROPIC or ANISOTROPIC. For each material, user needs to provide a unique ‘name’. The ‘density’ is optional. When omitted, it will use the default value 1.0. The components of elastic properties for each type are the same as those in Abaqus, and the arrangement of components will not be a problem. For ISOTROPIC, we have 2 components, 'e' and 'nu'. For ENGINEERING CONSTANTS, we have 9 components, 'e1', 'e2', 'e3', 'g12', 'g13', 'g23', 'nu12', 'nu13' and 'nu23'. For ORTHOTROPIC, we have 9 components, 'd1111', 'd1122', 'd2222', 'd1133', 'd2233', 'd3333', 'd1212', 'd1313' and 'd2323'. For ANISOTROPIC, we have 21 components, 'd1111', 'd1122', 'd2222', 'd1133', 'd2233', 'd3333', 'd1112', 'd2212', 'd3312', 'd1212', 'd1113', 'd2213', 'd3313', 'd1213', 'd1313', 'd1123', 'd2223', 'd3323', 'd1223', 'd1323' and 'd2323'.

Layup file

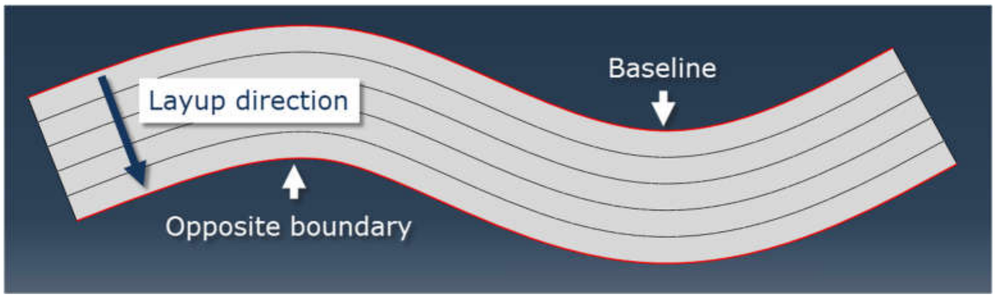
The root element is '<layups></layups>' and each '<layup></layup>' sub-element stores one layup.

```
<layups>
  <layup>
    <name>layup_1</name>
    <layer>
      <thickness>0.1</thickness>
      <material>mat_1</material>
      <fiber_orient>0</fiber_orient>
    </layer>
    <layer>
      ...
    </layer>
    ...
</layup>
...
</layups>
```

For each layup, user needs to provide a unique 'name'. Each layup has any number of layers, and for each layer, user needs to provide thickness, material name and fiber orientation.

3.4 Cross-Section with Arbitrary Shape

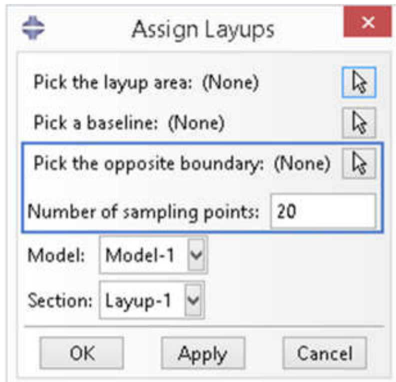
3.4.1 Laminate in the GUI



As shown in the figure, once the direction of layup is decided, the baseline is defined as the starting edge of the layup and the opposite boundary is the ending edge.

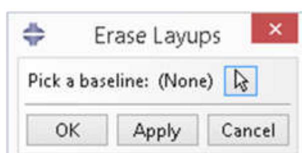
3.4.2 Dialog box

Clicking the icon button  will activate the 'Assign Layups' dialog box, shown as below.



A layup area is the whole segment face before partitioning into different layers and the baseline is shown as in the figure above. These two objects are required for all type of laminates. The next two inputs, the opposite boundary and number of sampling points, are required only when the baseline and opposite boundary are not straight lines. The sampling points are used to replicate those curved edges in Abaqus sketch module. The more sampling points, the more accurate the geometry will be. But large number of points will greatly reduce the speed of building the part. The last two dropdown lists are for users to choose composite layups.

Clicking the icon button  will activate the 'Erase Layups' dialog box, shown as below.



If a layup is assigned mistakenly, user can use this function to delete the wrong assignment. The baseline is the same as the one when assigning the layup. This function will help user to delete section assignments, sets and erase lines in the partition sketch, so that user can re-assign the area without worrying about duplicate or conflicting sets or section assignments.

3.4.3 Prerequisite

Before using the "Assign Layups" function, user needs to provide the geometry of the cross-section, materials and composite sections.

3.4.4 Example

<u>Step 1: Set work directory</u>

Menu > File > Set Work Directory.... Select the directory where you want to put all relative files, .cae, .inp, .sc, etc.

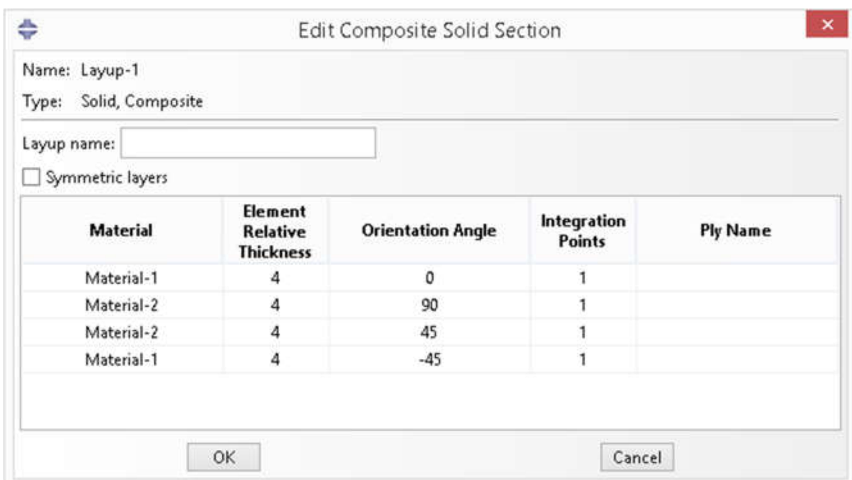
Step 2: Create materials

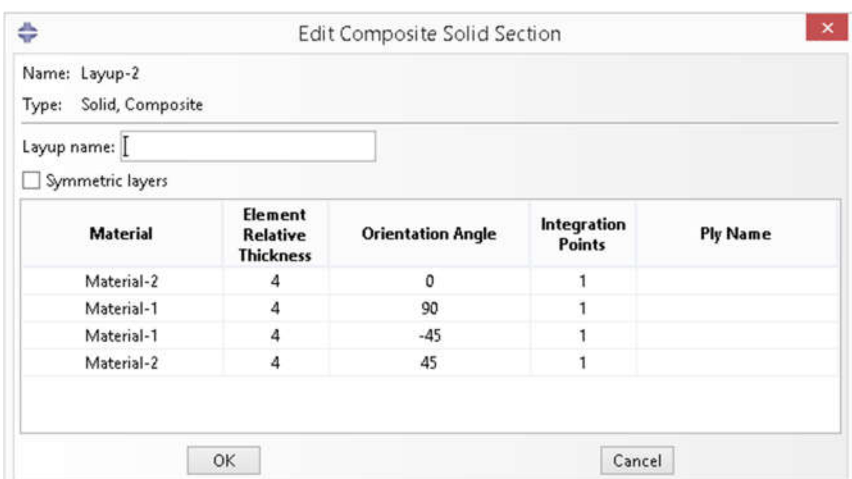
Module > Property > Create Material. Create two materials, Material-1 and Material-2, with properties you want.

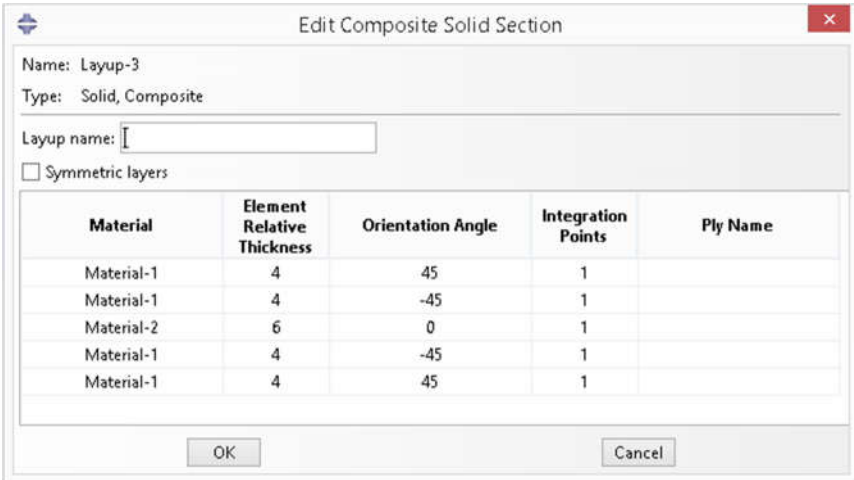
Step 3: Create composite sections

User can create layups using functions described in section 3.3, or using Abaqus's own functions by hand, which is described below.

Module > Property > Create Section. Create three sections, Layup-1, Layup-2 and Layup-3, shown as below. Either *Solid-Composite* or *Shell-Composite* is fine. Here we use *Solid-Composite* sections.





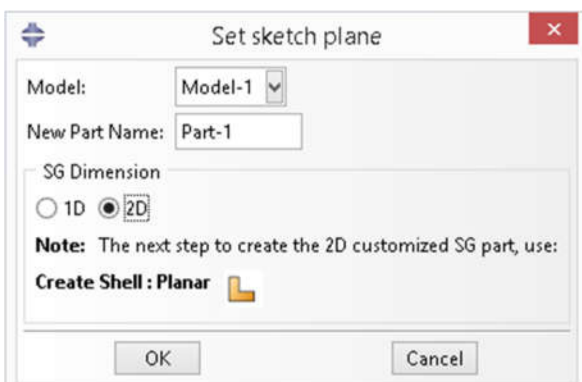


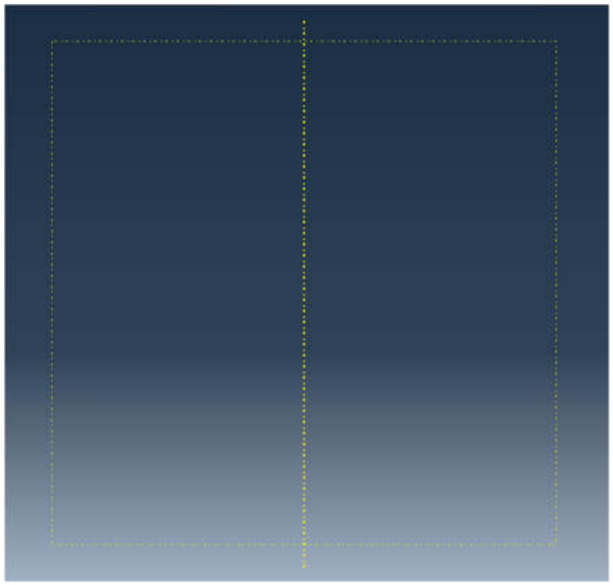
Note here that the thickness of each layer should be actual thickness instead of relative thickness.

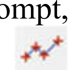
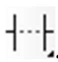
Step 4: Draw the general shape

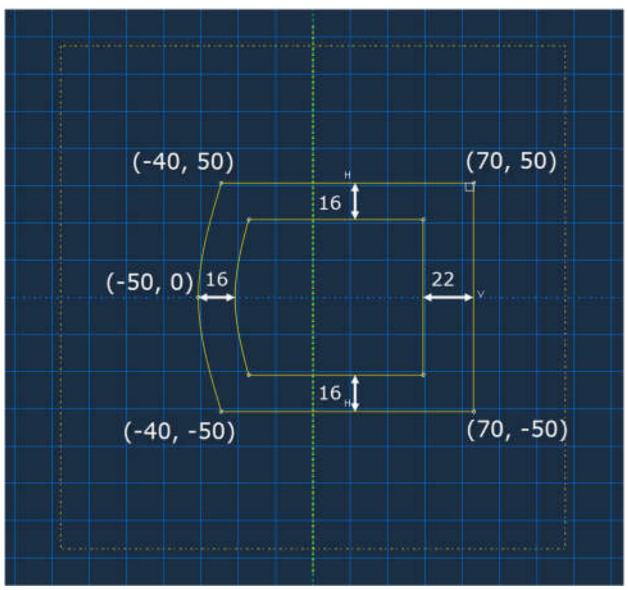
According to the coordinates convention in SwiftComp, the X-axis is along the beam reference line and the cross-section is in the Y-Z plane. Thus we need to first set our workplane to Y-Z plane.

In the SwiftComp Toolset, click the icon button , set the 'New Part Name' and select '2D' as the 'SG Dimension'. Click 'OK'.



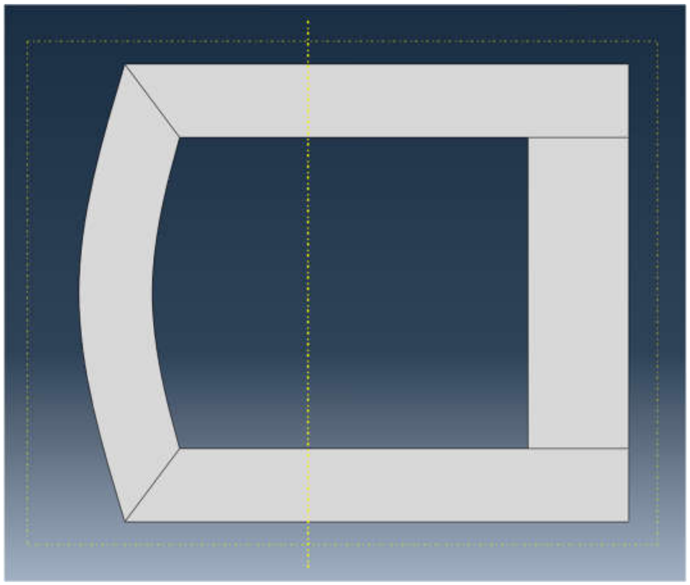


In the toolbox of the Part module, click ‘Create Shell: Planar’ , following the prompt, select the datum plane and the datum axis. Sketch the shape shown below, using commands , , , , with dimensions marked in the figure.



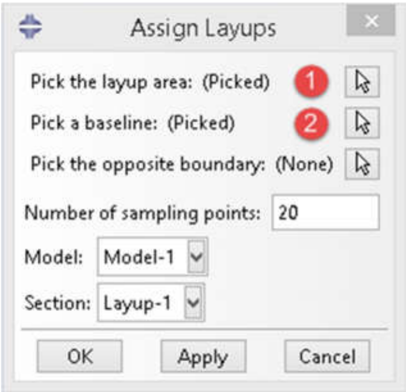
Click ‘Done’, then the part will be generated.

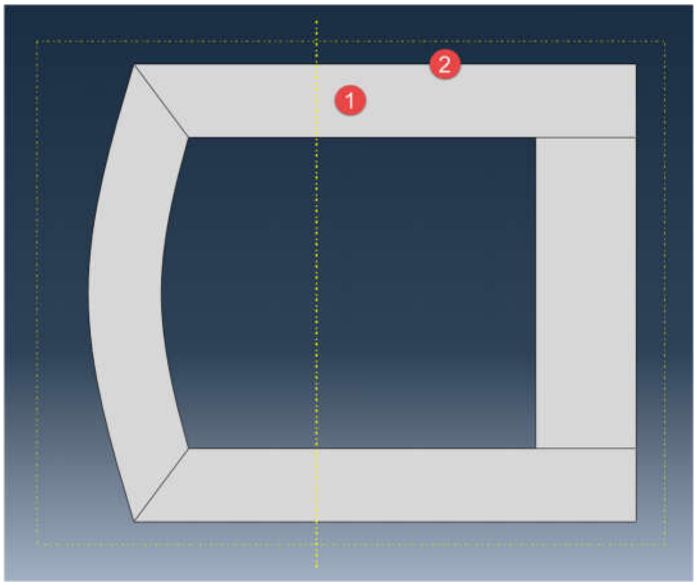
Next we need to divide this part into four segments. Click ‘Partition Face: Sketch’  and select a vertical edge on the right. Then sketch the partition and click ‘Done’.

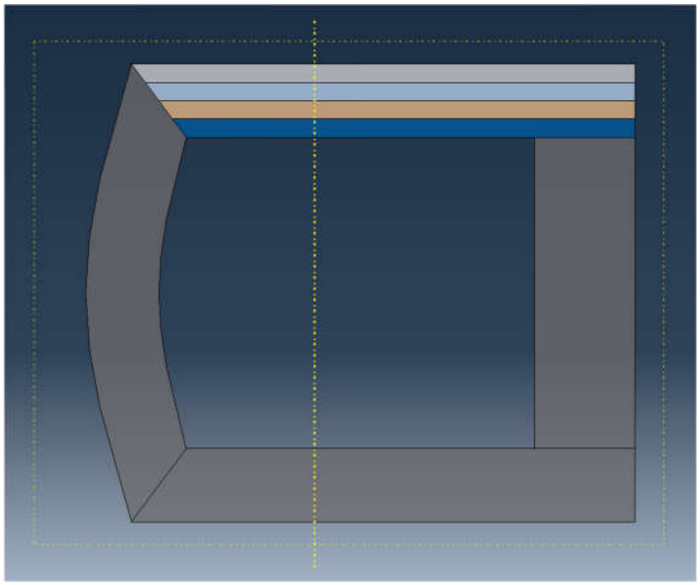


Step 5: Assign layups

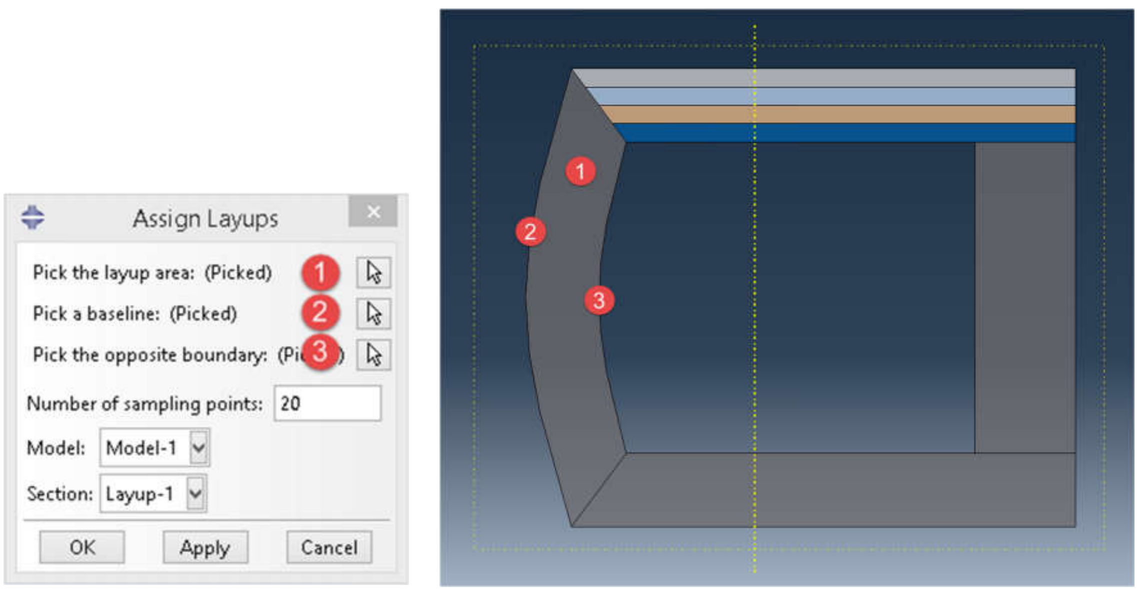
First we will assign the layup for the segment on the top. Click 'Assign Layups' in the SwiftComp toolset. Pick the area 1, pick the baseline 2 and select section 'Layup-1'. Click 'OK'.

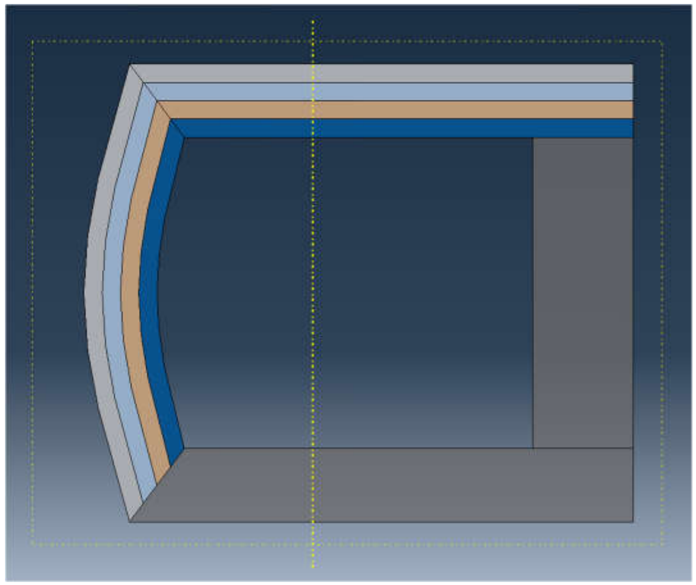




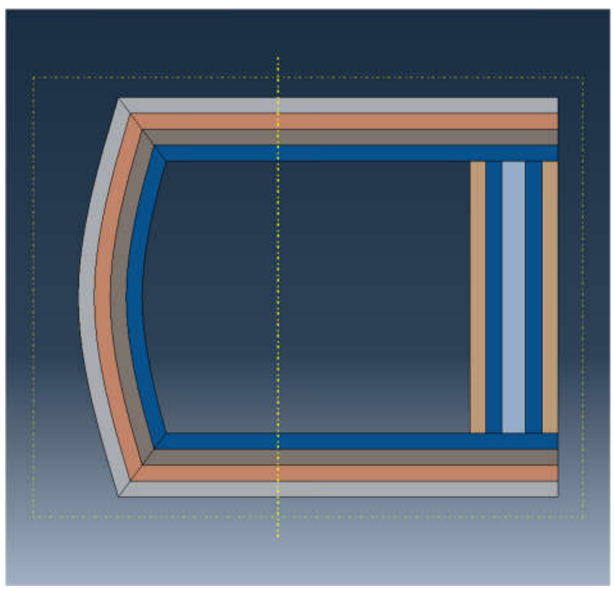


Next, assign the layup for the segment on the left. Pick the area 1, baseline 2, opposite boundary 3, give 20 sampling points and select section ‘Layup-1’. Click ‘OK’.



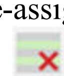


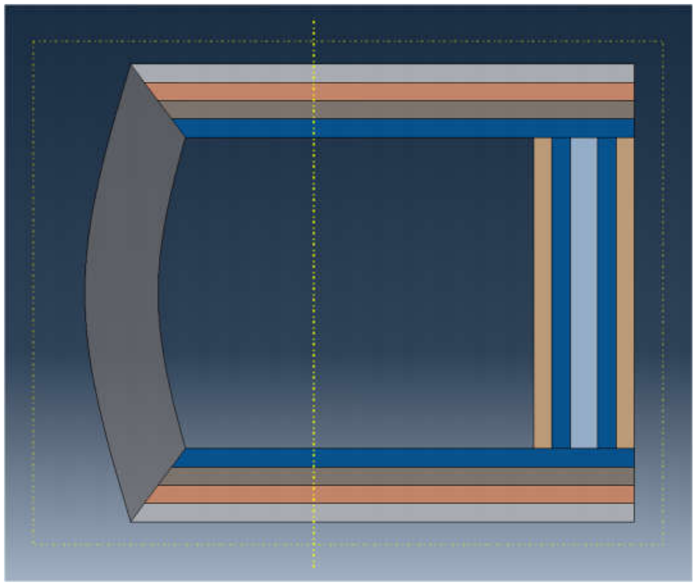
We can do the same for the rest two segments, except that we use ‘Layup-3’ for the right segment.



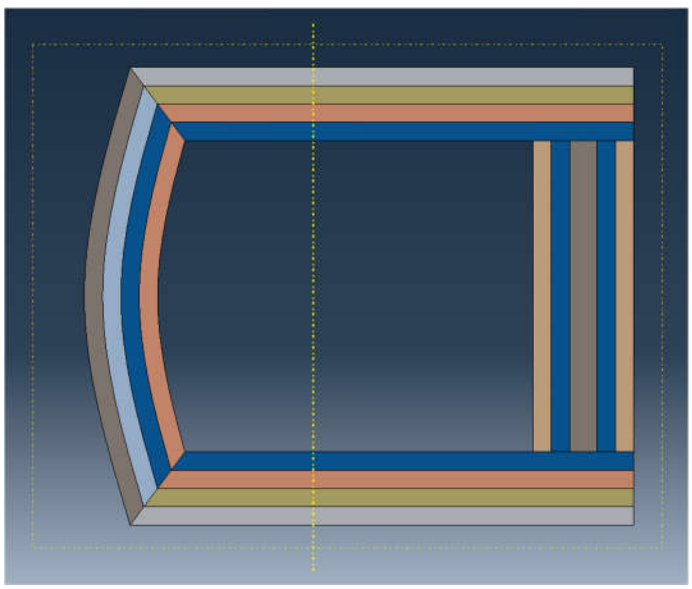
User may notice that the color for some layers changed after assigning the layup for the right segment. This is because we have some new sections created in ‘Layup-3’ and the order of the section list changed, so that the color map also changed. But no need to worry here, since the section assignment are still right, the only difference is the color.

<u>Step 6: Delete layups</u>

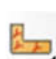
If users find that some layup is assigned mistakenly, it can be deleted and re-assigned. For instance, we want to re-assign the layup for the left segment. Click ‘Erase Layups’ . In the dialog box pick the same baseline used for assigning the layup. Click ‘OK’.

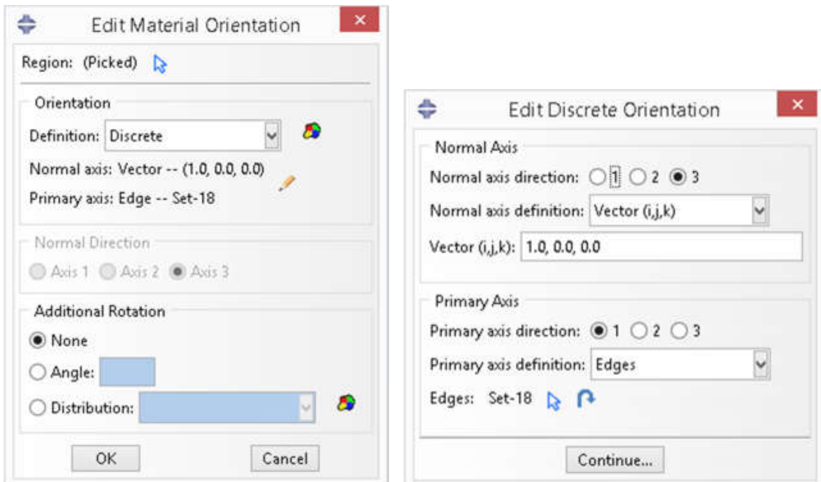


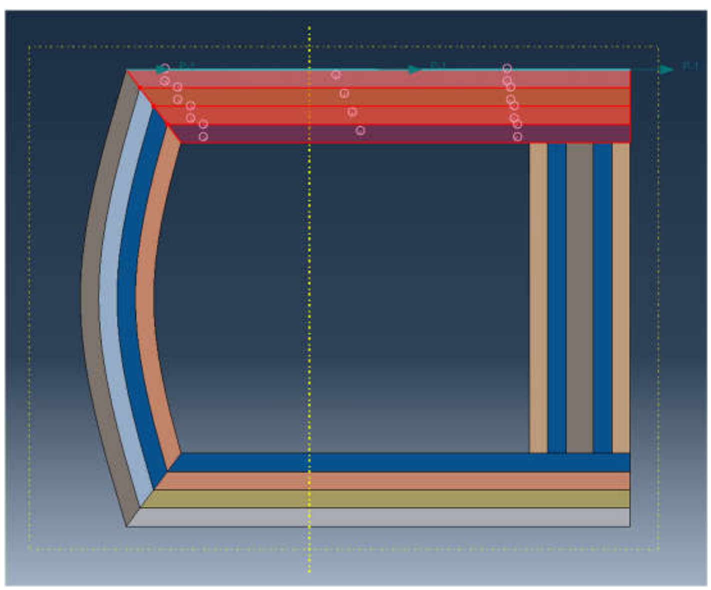
Once done, user can assign a new layup for the empty segment. Here we assign the 'Layup-2'.



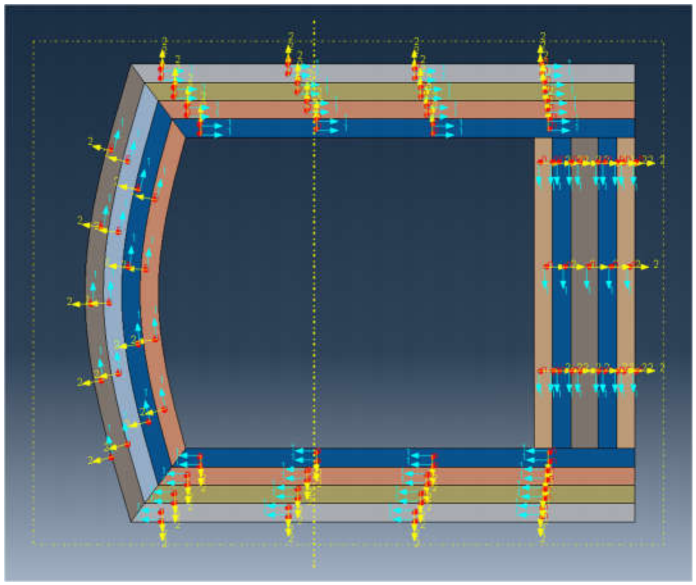
Step 7: Assign local coordinates

Module > Property > Assign Material Orientation . Note: for a shell part in Abaqus, we can only have axis 1 and 2 in the cross-section plane. Thus we will make axis 1 and 2 as the local y2 and y3 axis in SwiftComp convention respectively. Here we will keep axis 2 (y3) perpendicular to the outmost edges and pointing to the outside. User can also let axis 2 point inward, as long as the local orientation keeps consistency with the global coordinate ***and*** fiber orientations defined by users. Following the prompt, we first select the top segement to be assigned a local material orientation, and click 'Done'. Then in the prompt area, click 'Use Default Orientation or Other Method' button. In the 'Edit Material Orientation' dialog box, select 'Discrete' as the *Orientation Definition*. Then click , open the 'Edit Discrete Orientation' dialog box. In the *Normal axis* *definition* choose *Vector (i,j,k)*, and set the vector (1.0, 0.0, 0.0). For *Primary Axis*, choose 1 as the *Primary axis direction* and *Edges* as the *Primary axis definition*. Click to select the light blue edge shown in the figure below, then click 'Done'. Use *Flip Direction* to make the axis pointing to the right if necessary. Click 'Continue...' then 'OK' to finish this assignment. Same procedure for the rest three segments. Some steps may not be exactly the same. The only requirement is to make sure that the asix 2 pointing outwards.



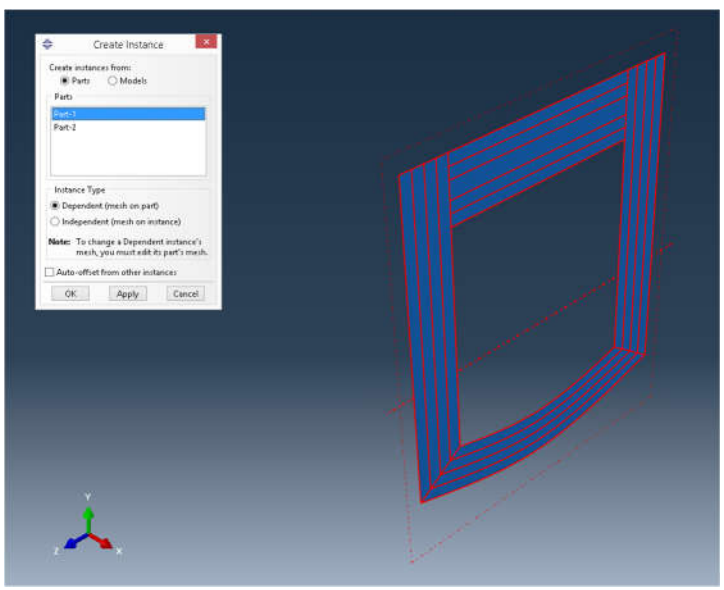


Once done, the local orientations should look like the same as the figure shown below.



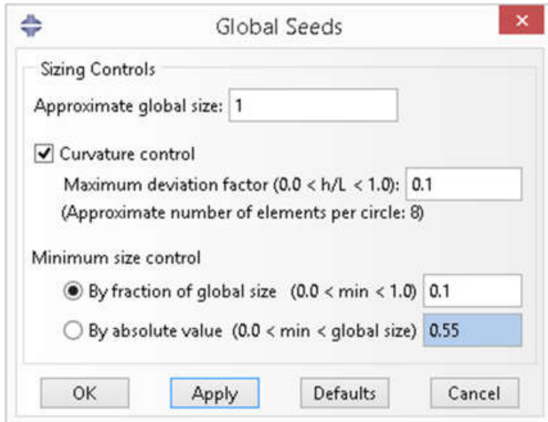
Step 8: Create assembly

Module > Assembly > Create Instance . In the 'Create Instance' dialog box, select 'Part-1', then click 'OK'.

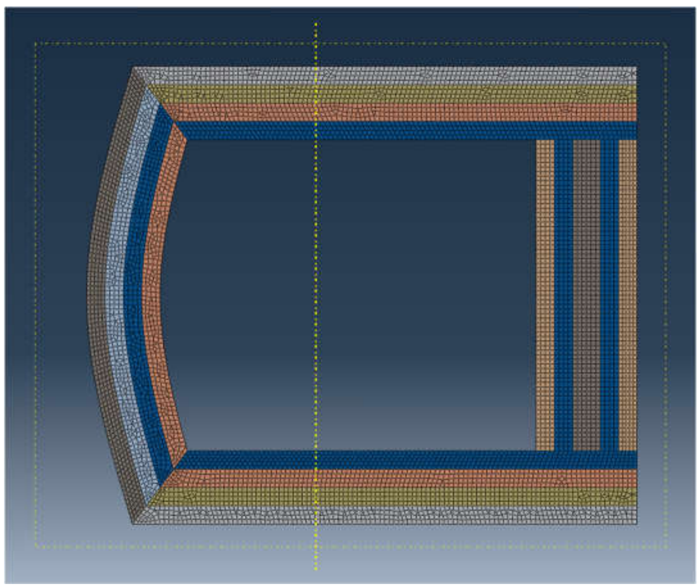


Step 9: Mesh

First choose 'Part' as the *Object*. Then click Module > Mesh > Seed Part . In the 'Global Seeds' dialog box, set *Approximate global size* to 1, then click 'OK', and click 'Done' in the prompt area.



Click 'Mesh Part' , then click 'Yes' in the prompt area.

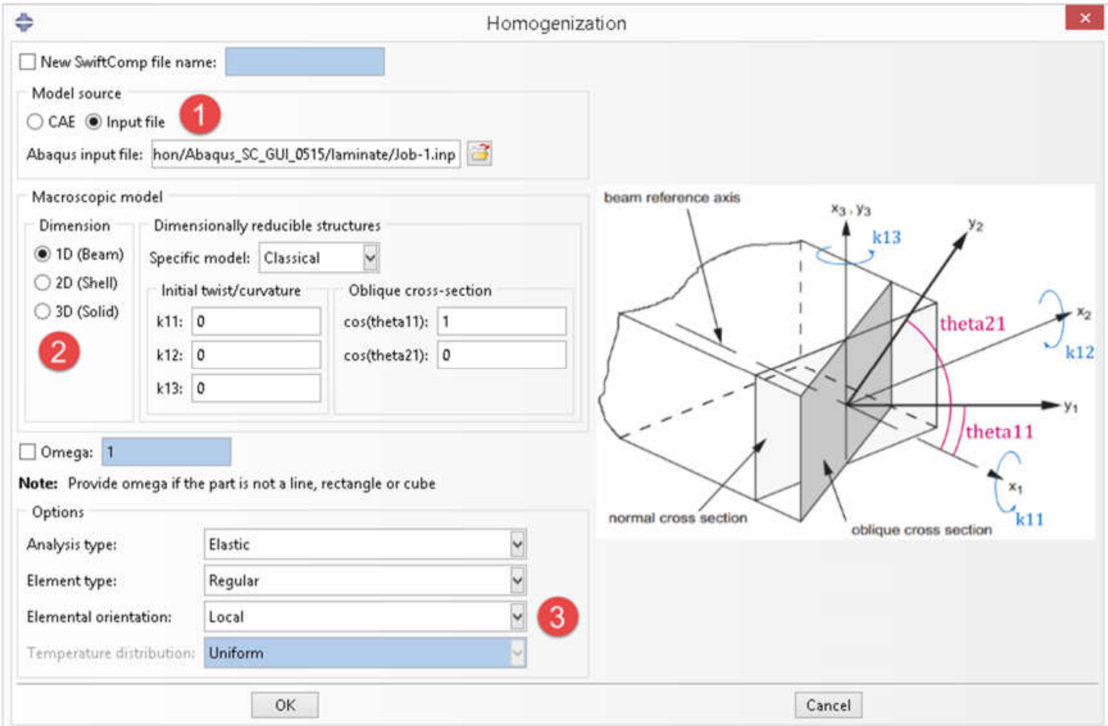


<u>Step 10: Write Abaqus input file</u>

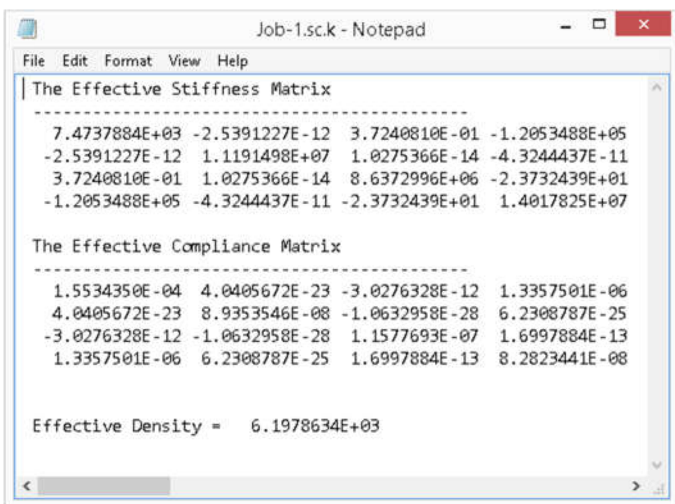
Module > Job > Create Job . In the 'Create Job' dialog box, set *Name* to 'Job-1' and click 'Continue...', and click 'OK' in the 'Edit Job' dialog box. In the model tree, right click 'Job-1', click *Write Input*, then an Abaqus input file 'Job-1.inp' is created.

<u>Step 11: Homogenization</u>

Click 'Homogenization' button in the SwiftComp toolset. In the dialog box, change the 'Model source' to 'Input file'. For 'Abaqus input file', select the abaqus input we just created, 'Job-1.inp'. Then click 'OK'. Note: since we assigned discrete local coordinates to the cross-section, those information for each element can only be found in the Abaqus input file. Thus user can only choose 'Input file' as the 'Model source' to do the homogenization.



After a certain time, depending on the model, SwiftComp will finish computing the cross-sectional properties and show the results in the notepad. If everything is fine, close the notepad and the process will end.



If error messages pop out, or an empty notepad file appears, please refer to the command line window for more information.

3.5 Cross-Section with Airfoil Shape

3.5.1 General

Cross-sections with airfoil shape usually contain several parts, like surfaces, webs and fillings, and the surface part has several segments and each segment has dozens or hundreds of layers, which makes it very difficult to build the model by hand as what we have shown in the previous section. The SwiftComp GUI will help user to draw the cross-section and prepare the SwiftComp input file automatically from airfoil data. Thus the main topic of this section is the preparation of data files. Once those files are ready, the whole process will be done by one click. A sample set of files are also given in the folder airfoil which the users can use as template to adapt for their own cross-sections.

3.5.2 Input data files

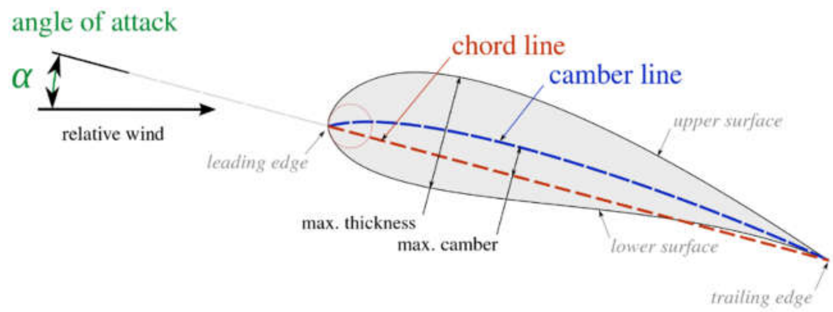
User needs to provide four files: control file, shape file, material file and layup file, which are all XML files. All files should be placed in the same folder. Now we will describe each file in details.

Control file

This is the main input that will be used in the SwiftComp GUI.

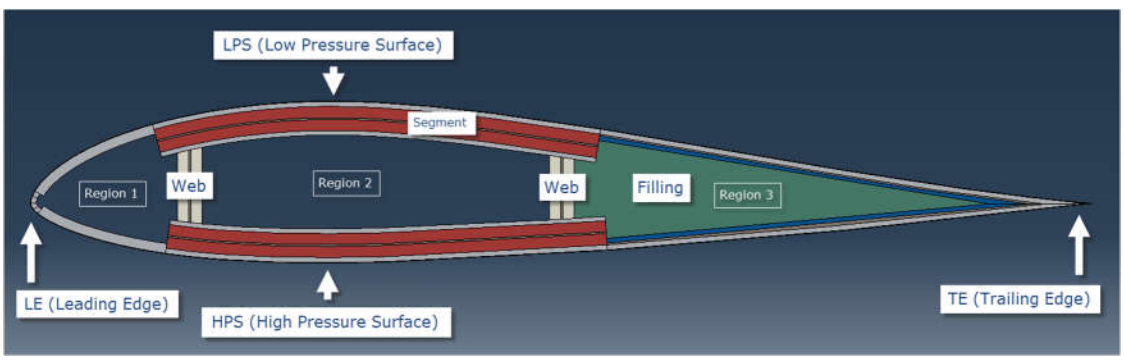
```
<project type="airfoil">
  <shapes>shape_filename</shapes>
  <materials>material_filename</materials>
  <layups>layup_filename</layups>
  <pitch_axis_yz>0.5  0.0</pitch_axis_yz>
  <twisted_angle>0.0</twisted_angle>
  <chord_length>1.9</chord_length>
  <flip>No</flip>
  <mesh_size>0.01</mesh_size>
  <element_type>linear</element_type>
</project>
```

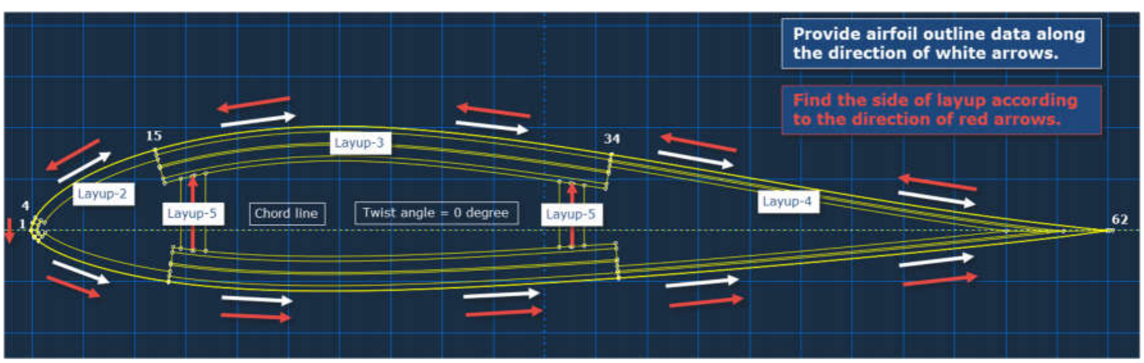
‘<project></project>’ is the root element, which has an attribute ‘type=“airfoil”’. Since the ‘Read File’ function mainly deals with cross-sections with airfoil shape, this arribute can be omitted for the current version. There are seven sub-elements below ‘<project></project>’. The first three sub-elements indicate the names of the other three data files, with file extensions “.xml” being appended or omitted. The next three lines tell the general shape parameter for the airfoil. Element ‘<pitch_axis_yz></pitch_axis_yz>’ indicates the normalized location of the pitch axis in the Y-Z plane, which is the cross-section plane. The two numbers are separated by spaces. Element ‘<twisted_angle></twisted_angle>’ contains the angle in degree between the actual chord line and the horizontal line. In the figure below, the ‘angle of attack’ is what we mean by ‘twisted angle’ here. Element ‘<chord_length></chord_length>’ stores the actual length of the chord line. The last element ‘<flip></flip>’ is used to indicate whether user want to switch the orientation of the airfoil. By default, the leading edge is on the left. See the figure below for a sketch of a typical cross-section of airfoil shape. User also need to provide a global mesh size and element type, which can be linear or quadratic.



Shape file

This file stores the shape data of the surfaces, webs and fillings, including the geometry information like coordinates of points and the segment division information like the dividing point number and the layup id that will be assigned to this segment. The structure is shown below.





```
<assembly type = "airfoil">
  <part structure = "surface">
    <baseline type = "spline" format = "lednicer">
      <lps>
      00
      0.00035937    0.0029595
      0.00162747    0.00696192
      ...           ...
      9.97E-01      1.33E-04
      1.00E+00      0.00E+00
      </lps>
      <hps>
      00
      0.00062671    -0.00294872
      0.0030035     -0.00629665
      ...           ...
      0.9970425     -0.00036119
      10
      </hps>
    </baseline>
    <layup side = "left">
      <lps>
      1  3 layup_1
      ...
      </lps>
      <hps>
      1  3 layup_1
      ...
      </hps>
    </layup>
  </part>
  <part structure = "web">
    <baseline>
      <y_z_angle>0.15  0.0  90.0</y_z_angle>
      ...
    </baseline>
    <layup side = "both">
    layup_2
    ...
    </layup>
  </part>
  <part structure = "filling">
<region material = "mat_form">3</region>
...
  </part>
</assembly>
```

There is a root element ‘<assembly></assembly>’, whose attribute ‘type’ can be omitted for the current version. The child element is ‘<part></part>’ and the attribute ‘structure’ for each part tells what the part is, surface, web or filling. For laminate-like part, surface and web, there are two sub-elements, ‘<baseline></baseline>’ and ‘<layup></layup>’.

The ‘baseline’ element of the ‘surface’ part contains two sub-elements, ‘<lps></lps>’ and ‘<hps></hps>’, which store the raw data of the airfoil low pressure surface and high pressure surface, respectively. Two attributes, ‘type’ and ‘format’ may be used in the future version and can be omitted here. The arrangement of data for both ‘lps’ and ‘hps’ is from (0, 0) to (1, 0), which is from the leading edge to the trailing edge. The ‘baseline’ element for the ‘web’ part contains any number of sub-elements ‘<y_z_angle></y_z_angle>’, each of whom stands for a web. Here a web is represented by a straight line. The first two numbers are the normalized position of the baseline on the chord line and the last number is the angle in degree measured from the chord line segment near the leading edge to the upper part of the web baseline.

The ‘layup’ element has an attribute ‘side’, which can have the value ‘left’, ‘right’ or ‘both’. The ‘side’ is defined according to the direction of the surface baseline, which starts from the trailing edge, goes along the low pressure surface baseline to the leading edge and back to the trailing edge along high pressure surface baseline. So basically, if the ‘side’ is ‘left’, then the plies are laid towards the inside of the airfoil. Thus if the surface baseline is the outmost profile and the leading edge is on the left, then the ‘side’ should be ‘left’. The value ‘both’ means doing the layup on both sides of the baseline, but will not double the layup. Or in other words, this means that the baseline is at the middle of the layup. For the current version, this case can only handle symmetric layup. Otherwise, user needs to translate the baseline to one side and change some values in the file accordingly. The ‘layup’ element for the ‘surface’ part contains two sub-elements ‘<lps></lps>’ and ‘<hps></hps>’. In each of them, there are several lines of data and each line represents a segment and has three values. The first two numbers are the starting and ending points for the segment and the last string is the layup name that will be defined in the layup file. The first number of the first line must be 1 and the second number of the last line has to be the total number of points for low pressure surface or high pressure surface. The ‘layup’ element for ‘web’ part is much simpler. Each line only contains the layup name for the web.

For the ‘filling’ part, such as foam, has only one type of sub-element, ‘<region></region>’. The number stored between the two tags is region id, which is defined as follows. If there are two webs, then the inner space of the airfoil is divided into three regions. Starting from the leading edge, those regions are labeled as ‘1’, ‘2’ and ‘3’. The attribute ‘material’ stores the material name, which is defined in the material file. Here we make the assumption that the materials used for the fillings are homogeneous, could not generally anisotropic, with material properties given in the global coordinates.

Material file

Please see section 3.3.2

Layup file

Please see section 3.3.2

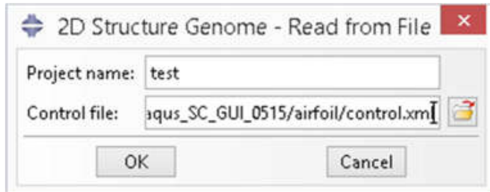
3.5.3 Create cross-section model in Abaqus from input files

Step 1: Set working directory

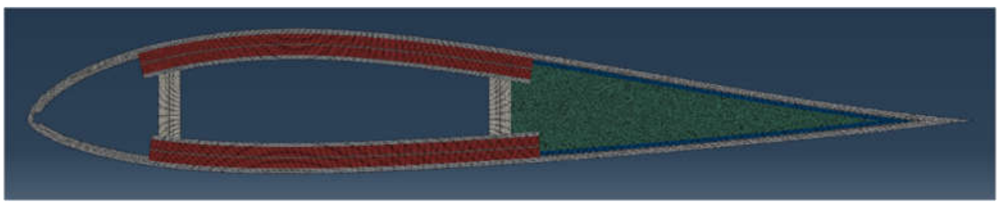
The working directory should contain those data files.

Step 2: Read input files

Click 'Read File' , set a project name and select the control file. Click 'OK'.



Once done, user should see a meshed airfoil cross-section with different colors representing different layer types, as shown in the figure below.



Step 3: Make some changes (optional)

At this stage, user can change material properties or meshes. Following these modifications, user need to re-write the Abaqus input file from the job.

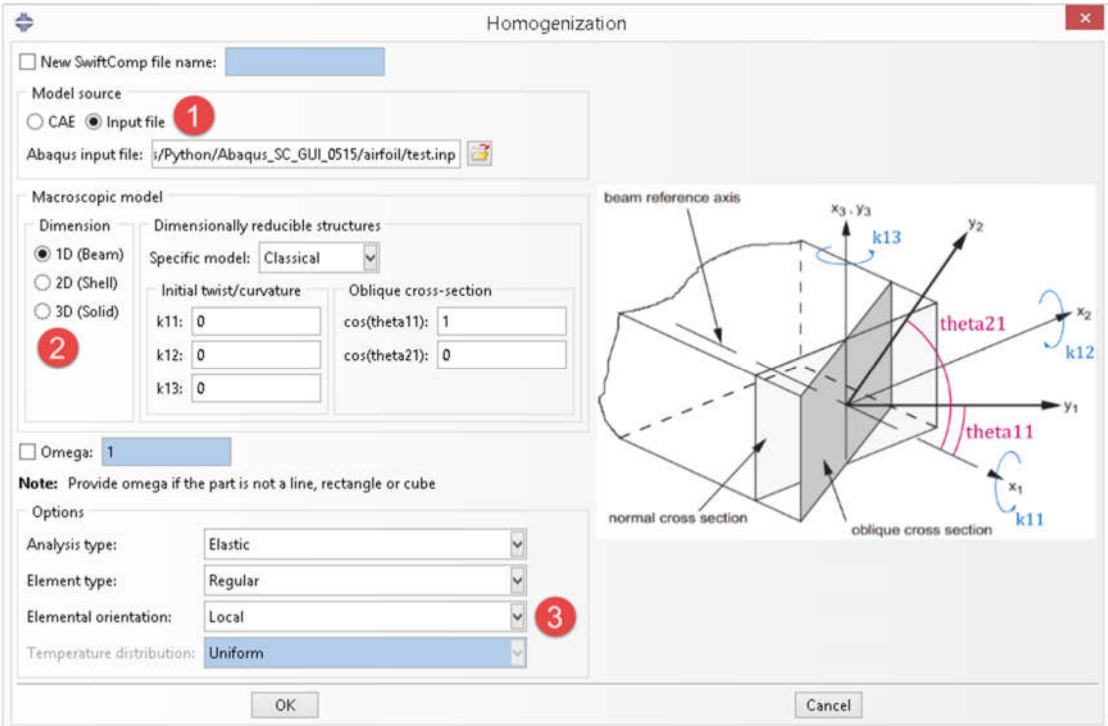
User can also do the creation in command line without entering Abaqus/CAE. First, user needs to enter the directory where the script 'drawCS.py' is located. Then type:

```
abaqus cae noGui=drawCS.py -- project_name control_file
```

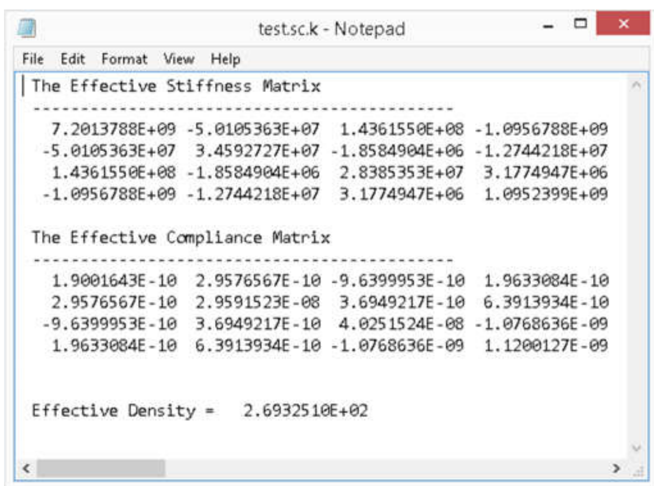
where 'control_file' is the complete file name including path and extension. Press 'Enter'. Once done, all those related files can be found in the same place where the control file is.

3.5.4 Homogenization

Click 'Homogenization' button  in the SwiftComp toolset. In the dialog box, change the three items shown as in the figure below. For 'Abaqus input file', select the abaqus input we just created, '*project_name*.inp'. Then click 'OK'.



After a certain time, depending on the model, SwiftComp will finish computing the cross-sectional properties and show the results in the notepad. If everything is fine, close the notepad and the process will end.




```
The Effective Stiffness Matrix
-----------------------------------------------
  7.2013788E+09 -5.0105363E+07  1.4361550E+08 -1.0956788E+09
 -5.0105363E+07  3.4592727E+07 -1.8584904E+06 -1.2744218E+07
  1.4361550E+08 -1.8584904E+06  2.8385353E+07  3.1774947E+06
 -1.0956788E+09 -1.2744218E+07  3.1774947E+06  1.0952399E+09

The Effective Compliance Matrix
-----------------------------------------------
  1.9001643E-10  2.9576567E-10 -9.6399953E-10  1.9633084E-10
  2.9576567E-10  2.9591523E-08  3.6949217E-10  6.3913934E-10
 -9.6399953E-10  3.6949217E-10  4.0251524E-08 -1.0768636E-09
  1.9633084E-10  6.3913934E-10 -1.0768636E-09  1.1200127E-09


Effective Density =    2.6932510E+02
```

If error messages pop out, or an empty notepad appears, please refer to the command line window for more information.

4 IMPORT HOMOGENIZED PROPERTIES TO MACRO MODEL ANALYSIS

The homogenized properties obtained from homogenization can be imported into a new model with the name of the sg model. Click the button as shown in the Figure 4-1, a window shown in Figure 4-2 will pop out. There are 2 methods to import the homogenized properties. If the sg model is existed, the user can choose the sg model name (sg_name); if there is only SwiftComp analysis file (.sc.k) available, the user can choose the file and also specify the Analysis type and Macroscopic model dimension. Click OK and the homogenized properties will be imported.

The examples of Section 2.1 Laminate (1D SG) are used to show how to importe homogenized properties in this Section.

If the macro model is plate/shell, general plate stiffness will be imported to sections in the Property module (Figure 4-2, Figure 4-3).

If the macro model is 3D solid, homogenized anisotropic material properties will be imported into materials in the Property module. If the homogenized material properties is orthotropic (when expressed in the global coordinate system) and can be expressed using Engineering Constants, a material defined using Engneering Constants will also be defined in materials (Figure 4-4).

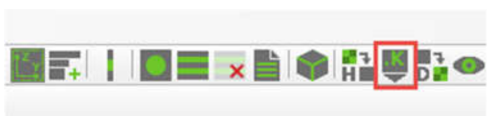
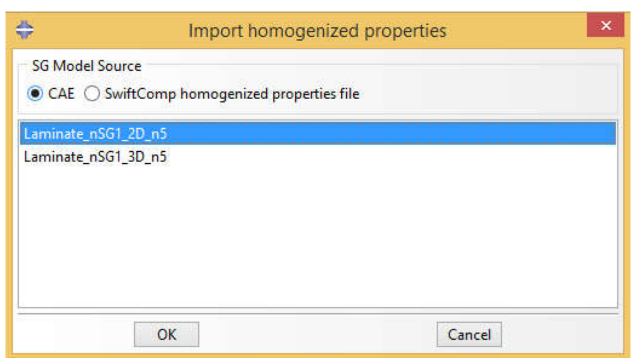
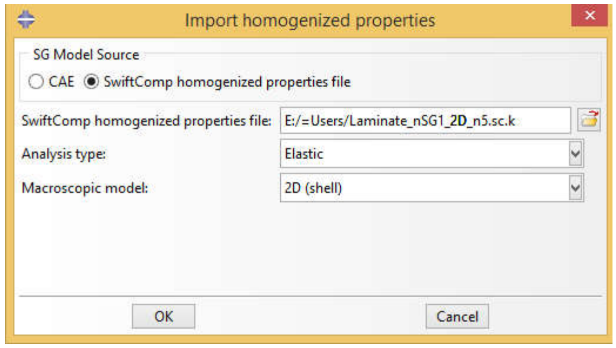


Figure 4-1: Import the homogenize properties in to Abaqus CAE.



(a) Import from sg model information



(b) Import from SwiftComp homogenization result file (*.sc.k)

Figure 4-2: Import the homogenize general stiffness section properties of shell model into Abaqus.

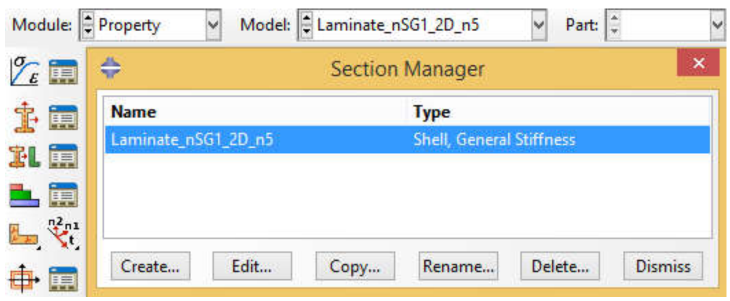


Figure 4-3: Imported homogenize general stiffness section properties of shell model.

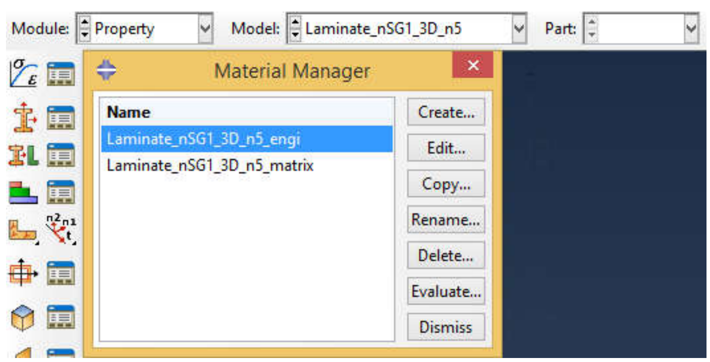


Figure 4-4: Imported homogenize material properties of solid model.