UTILIZATION OF COMSOL MULTIPHYSICS JAVA API FOR THE INTEGRATION OF COMPOSITE MATERIAL MODULE WITH A CUSTOMIZED USER INTERFACE

#### Authors:

Dr. Venkateswaran P Siemens Technology & Services Private Limited Bangalore 560100 Venkateswaran.p@siemens.com

Mr. Rishabh Malav Indian Institute of Technology, Madras

COMSOL CONFERENCE 2014 BANGALORE Excerpt from the Proceedings of the 2014 COMSOL Conference in Bangalore

## INTRODUCTION

### **Composite Materials**

- Composites are made up of individual materials referred to as constituent materials
- Two main categories of constituent materials: Matrix and Fiber
- Some Advantages: Light Weight, High Strength, Durability, Strength Related to Weight, Corrosion Resistance and Design Flexibility

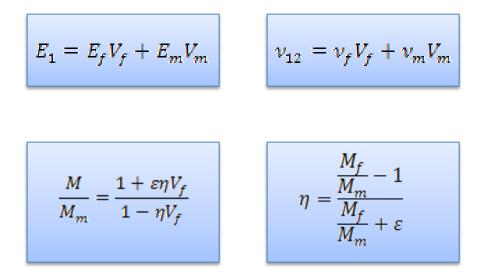
### Need to integrate with COMSOL

- COMSOL Multiphysics is capable of modeling and simulation of any physics-based system
- Adding a user-friendly interface to integrate Composite Material properties will further enhance its capabilities
- Use of Composite Materials in various industries such as aircraft, automotive, etc is rapidly getting acceptance
- So, in the process of modeling and simulation, many a times user wants to use Composite Materials properties in the analysis

- COMSOL API (Application Programming Interface) which is an interface based on Java, is used to develop custom application based on COMSOL
- An initial GUI is built based on a model of a simple 'Rectangular Block', having a point force acting on the edge
- Final GUI is built based on a model of 'Mechanical Part', having a combination of forces and moments acting on it
- Halpin-Tsai Model, a semi-empirical model chosen based on its accuracy and appropriateness, is used to develop algorithm to calculate Composite Material properties

## HALPIN-TSAI MODEL

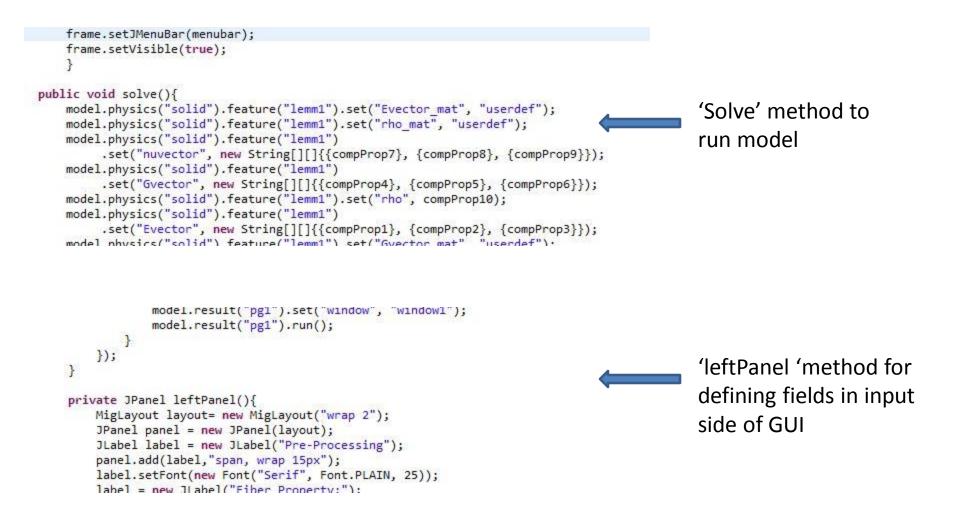
Halpin-Tsai equations are the handy forms of Hill's generalized self-consistent model results with engineering approximations to make them suitable for the designing of composite materials.



where,
M = composite material modulus $E_{22}$ , $G_{12}$ or $v_{23}$
$M_f$ = composite material modulus $E_f$ , $G_f$ or $v_f$
$M_m$ = composite material modulus $E_m$ , $G_m$ , or $v_m$

```
import java.awt.BorderLayout;
 class RectModelDemo implements ActionListener {
     private JFrame frame;
     private JFrame frame1
     private Model model;
                                                                              Declaration of Class
     private JButton solveButton;
                                                                              and Parameters
     private JButton calculateButton;
     private NumberFormat percentFormat;
     private JTextField editV;
     private JTextField editE1;
     private lTextField editE2.
  static String[] compProperty = new String[10];
  public static void main(String[] args) {
      RectModelDemo demo = new RectModelDemo();
      demo.init();
      demo.start();
                                                                              Basic methods:
  public void init() {
     ModelUtil.initStandalone(true);
                                                                              Main, init, start
  public void start() {
      lookandfeel();
     frame= new JFrame("Model taken from COMSOL - GUI");
     frame.setDefaultCloseOperation(JFrame.EXIT ON CLOSE);
      frame.setSize(1200,900);
      GilPanel mainPanel = new GilPanel().
```

Excerpt from the Proceedings of the 2014 COMSOL Conference in Bangalore



```
model.result("pg4").set("window", "window1");
           model.result("pg4").run();
   });
   return panel;
                                                                                      'Solve Button' which
}
private JButton solveButton(){
                                                                                     calls 'solve' method
   solveButton = new JButton("Simulate");
   model= RectModel.run();
   solveButton.addActionListener(new ActionListener() {
   public void actionPerformed(ActionEvent e) {
       solve();
   });
   return solveButton;
}
private void sPlot(){
    model.result().dataset("cpl1")
       .set("genpoints", new String[][]{{xs1Coord, ys1Coord, zs1Coord}, {xs2Coord, ys2Co
      model.result().dataset("cpl1").set("planetype", "general");
                                                                                      'Calculate Button'
   model.result("pg6").set("window", "window1");
   model.result("pg6").run();
                                                                                      method calculates
}
                                                                                     composite properties
private JButton calculateButton(){
    calculateButton = new JButton("Calculate Composite Properties");
    calculateButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
       String fiberPer = editFC.getText();
        double fibPercent = Double.parseDouble(fiberPer);
        String matrixDar - aditMC gatTevt().
```

Iterator<Row> rowIterator = matrix.iterator();

int mat index=0;

while(rowIterator.hasNext()) {

Row row = row Tterator next():

```
public static void fiberProp(){
        try {
            FileInputStream fiberProperty = new FileInputStream(new File("D:\\Ris
            HSSFWorkbook workbook = new HSSFWorkbook(fiberProperty);
                                                                                         'fiberProp' method for
            for(int i=0; i<13; i++){</pre>
                HSSFSheet fiber = workbook.getSheetAt(i);
                                                                                         defining fiber properties
                Iterator<Row> rowIterator = fiber.iterator();
                int row index=0;
                while(rowIterator.hasNext()) {
                    Row row = rowIterator.next():
public static void matrixProp(){
   try {
        FileInputStream matrixProperty = new FileInputStream(new File("D:\\Rishabh
        HSSFWorkbook workbook = new HSSFWorkbook(matrixProperty);
                                                                                         'matrixProp' method for
        for(int i=0; i<13; i++){</pre>
           HSSFSheet matrix= workbook.getSheetAt(i);
```

Excerpt from the Proceedings of the 2014 COMSOL Conference in Bangalore

defining matrix properties

```
private JPanel rightPanel() {
    MigLayout layout = new MigLayout("wrap 6");
    final JPanel panell = new JPanel(layout);
    JLabel label = new JLabel("Post-Processing");
    panell.add(label,"span, wrap 15px");
    label.setFont(new Font("Serif", Font.PLAIN, 25));
    label = new JLabel("Contour Plot:");
    panell.add(label,"span, wrap 10px");
    label.setFont(new Font("Serif", Font.PLAIN, 17));
    String[] contourType = {"Select Contour Type", "Lines", "Filled"};
```

```
'rightPanel' method for
defining fields in
output side of GUI
```

```
private JButton linePlot(){
   JButton linePlot = new JButton("Line Plot");
   linePlot.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
           x1Coord = editX1.getText();
           y1Coord = editY1.getText();
           z1Coord = editZ1.getText();
           x2Coord = editX2.getText();
           y2Coord = editY2.getText();
           y2Coord = editY2.getText();
           z2Coord = editY2.
```



```
private void plot(){
           model.result().dataset("cln1")
           .set("genpoints", new String[][]{{x1Coord, y1Coord, z1Coord},
                   {x2Coord, y2Coord, z2Coord}});
           model.result("pg2").set("window", "window1");
                                                                                  'plot' method and
           model.result("pg2").run();
                                                                                   'surfacePlot' method for
                                                                                   plotting surface Plot
private JButton surfacePlot(){
   JButton surfacePlot = new JButton("Surface Plot");
   surfacePlot.addActionListener(new ActionListener() {
       public void actionPerformed(ActionEvent e) {
           xs1Coord = editXs1.getText();
           vs1Coord - editVs1 getText().
private JMenuBar menu() {
      JMenuBar menubar = new JMenuBar();
                                                                                   'menu' method for menu
     JMenu menu = new JMenu("File");
     menubar.add(menu);
                                                                                  bar at the top of GUI
     JMenuItem item = new JMenuItem("Open");
     menu.add(item);
     item = new JMenuItem("Exit");
     item.setActionCommand("exit");
      itom addActionListonan(this).
```

## **USER INTERFACE...**

### INPUT INTERFACE —

	Pre-Proce	essing			
				AS4 Graphite	-
	Fiber Property:	Kevlar 49 👻	 $\rightarrow$	Select Fiber Property	*
	% Fiber Content:	30		Boron Kevlar 49	Ξ
	Matrix Property:	Vinylester 🗸		AS4 Graphite E-Glass 21 x K43 Gevetex	
	% Matrix Content:	70		E-GLASS High Mod Graphite	
	Calculate Compo	osite Properties		HTS150	-
			$\rightarrow$	8551-7 Epoxy	-
$\rightarrow$	E11:	39615 MPa		Select Matrix Property	*
	E22:	4287.22 MPa		Polyester Vinylester	_
				3501-6 Epoxy	Ξ
	E33:	4287.22 MPa	1	5250-4 RTM	
	G12:	1494.97 MPa		5505 Epoxy	
	G23:	2143.61 MPa		8551-7 Epoxy BSL914C Epoxy	-
	G31:	1494.97 MPa			
	Nu12:	0.37			
	Nu23:	0			
	Nu31:	0			
	Density:	1.32 g/cm^3			
	Simulate	Visualize Meshing			
	D 11 C (1				

Excerpt from the Proceedings of the 2014 COMSOL Conference in Bangalore

# ... USER INTERFACE

**INTERFACE FOR POST-PROCESSING** 

OF RESULTS

Visualize Line Plot Back	
Surface Plot:	Contour Plo
Plot 👻	Select Contour 1
Enter First Point:	Plot
X: Y: Z:	
Enter Second Point:	
X: Y: Z:	
Enter Third Point:	
X: Y: Z:	
Visualize Surf Surface Plot Back	

Z:

Z:

Max/Min Line Plo	ot:
Select Plot 👻	
Plot	

Contour Plot:	
Select Contour Type	•
Plot	

Excerpt from the Proceedings of the 2014 COMSOL Conference in Bangalore

Line Graph:

Enter First Point:

Enter Second Point:

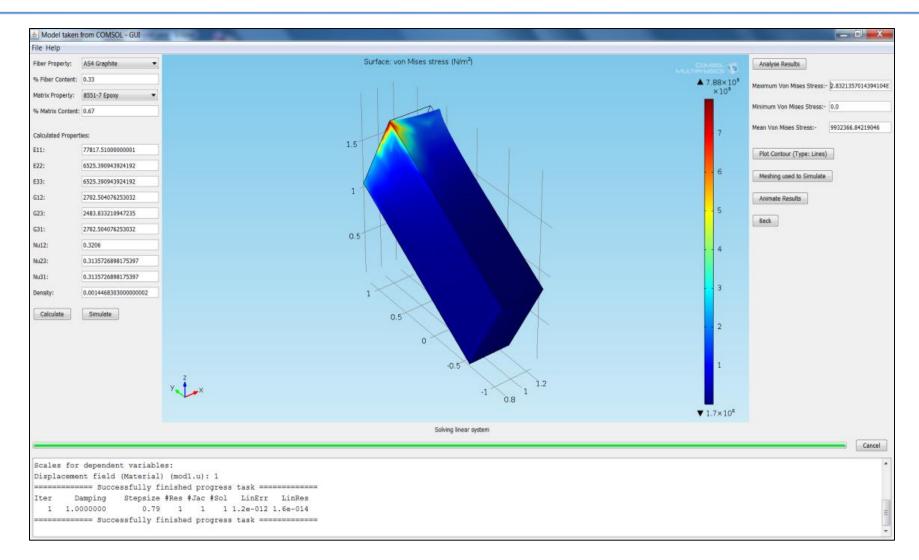
Y:

Y:

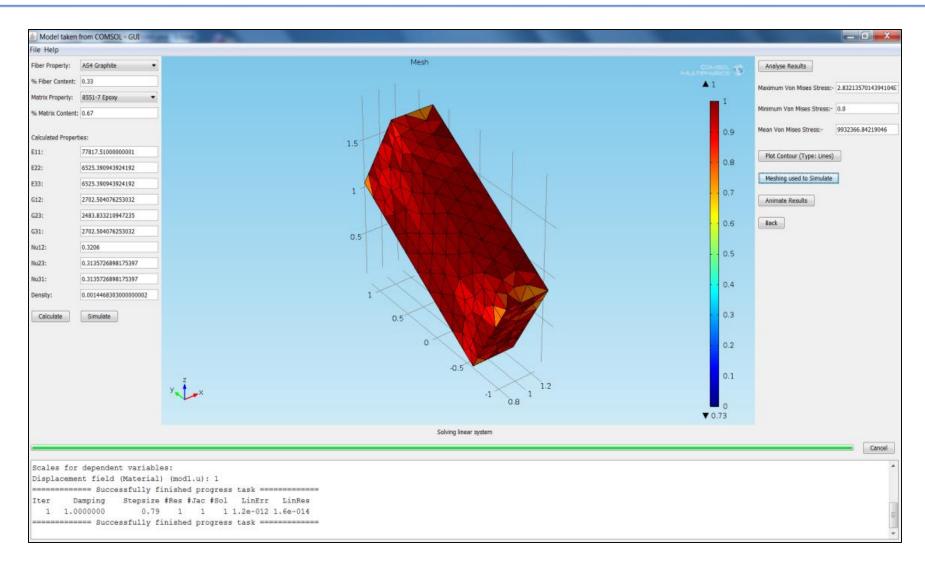
Plot

X:

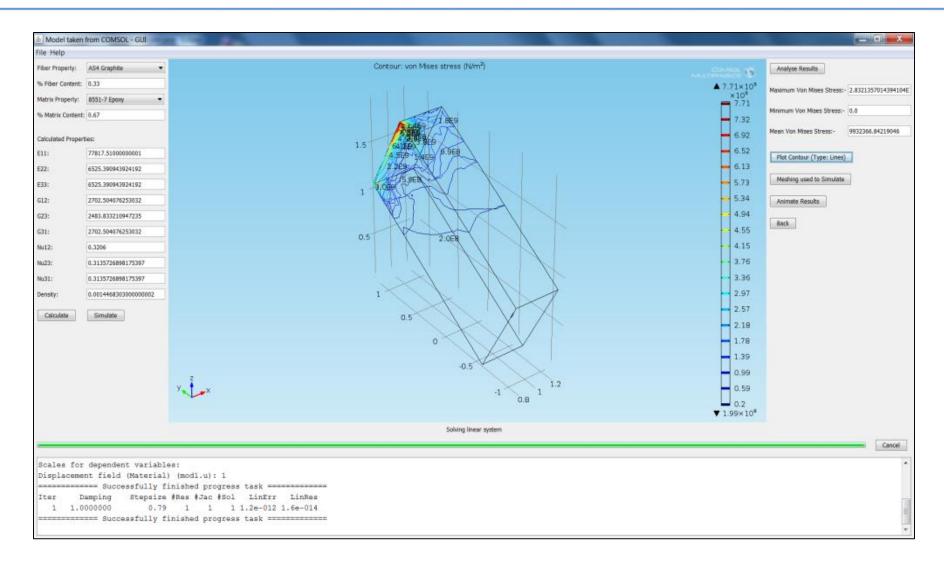
Х:

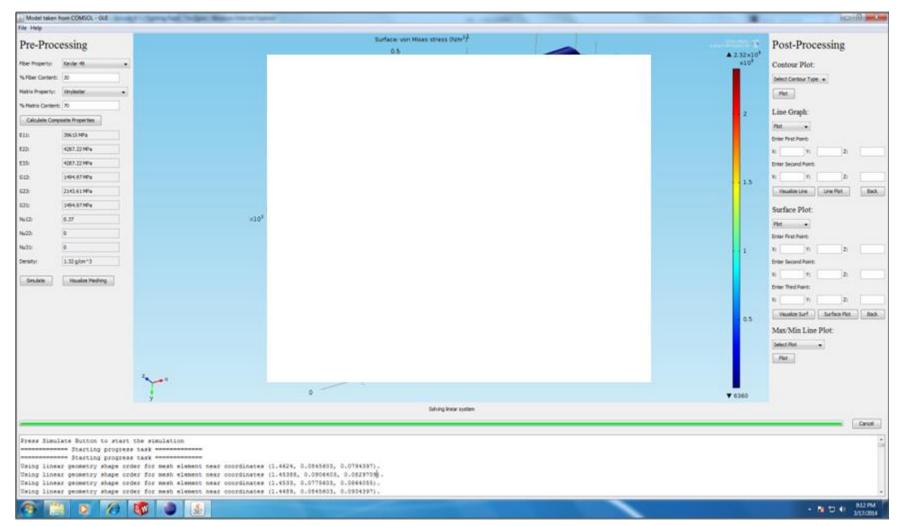


Excerpt fro



Excerpt from the Proversities Strate Strate





ExcerSIMULATION ONGAO MORE COMPLEX MODEL IN Bangalore

Model taken fr	Ion COMSOL - GUE	1 - Same had Sudjets Water	a laplest			- BS	a 🙃 🔜
Pre-Proce	accina				and the second second	Post-Processing	
Ple-Ploce	essing				▲ 2.27×10 <sup>9</sup>	Post-Processing	
be Property:	revier-to 🔹 🔹				*10 <sup>8</sup>	Contour Plot:	
% Fiber Content:	20					Lines +	
Nabia Property:	implexiter •				- 2.15	(.Het.)	
% Hatrix Cartient	2				2.03	and the second se	
Calculate Compo	oste Properties				1.92	Line Graph:	
111	2665 MPa				1.0	Pot •	
	4287.22 MPa					Enter Post Paint:	
	4287.2219%				1.69	x: v: 2:	
612	1404,97369				1.57	Enter Second Point	
	2143.61 MFa				1.45		
					1.34	Vaulae Une	. Seck
	1494,373499					Surface Plot:	
	8.37				1.22	PM	
w23					1.1	Enter First Paint:	
NuCO:	1				0.99	N: N: D	s - [
Density:	1.32 g/cm*3				0.87	Enter Second Points	
Smilele	Visualize Meshing				10220	10 m Z	F
the contraction of	Enconstruction for				0.76	Enter Third Paint:	
					0.64	10 10 Zz	-
					0.52	Vaualize Surf Surface Pla	et bek
					0.41	Max/Min Line Plot:	
					0.41	Select Plot +	
					0.29	PM	
					0.17		
		~			0.06		
		\$			▼ 5.81×10 <sup>3</sup>		
				Solving linear system			
-							Cantal
Fress Simula	ate Button to start	the simulation					
		s task					
		s tasy	ates (1.4624, 0.0845605, 0.079451	97).			
			ates (1.45388, 0.0906403, 0.08297 ates (1.4533, 0.0775603, 0.086405				
			ates (1.4419, 0.0241603, 0.09343)				
A 14	DA	🔞 🍙 🚯				- 10 0 0	913 PM
			Excerpt from the P				1/17/2014

**CONTOUR PLOT** 

Pre-Processing			and the second s	Post-Processing	
-					
be hisperty: Kevlar 40				Contour Plot:	
Willer Content: 30				riled •	
latix Property: Viryleater	nel l			.PM.:	
- Matrix Content: 20				Line Graph:	
Calculate Corposite Properties				Haes Stress	
10 31615 MPa				Enter First Pant.	
4387.32 HPs				8 0 10 4 Zi 3	1
131 K087.22.94Pa				Enter Second Paints	
121 1404.9714Pe				11: 2 11: 3 Z 4	
23: 2143.61.MPa				Visualize Line Plot	Sec.
32 1404.97MPa				Surface Plot:	
0.37				Put	
429 0				Enter Fest Paints	
udu 0				n n	
enaty: 1.32 glow*3				Enter Second Paint:	
				n n	
Servizte Vaualize Heating				Enter Trad Paint;	
				n n a	
				Visialize Surf Surface Plut	Bak
				Max Min Line Plot:	
				Total Displacement, +	
				Plat	
	in a second s				
		Salving Insur system			
					anot
Fress Simulate Button to	tart the simulation				
	gress task				
wing linear geometry sha	e order for mesh element near coordinates il				
	e order for mesh element near coordinates () e order for mesh element near coordinates ()				
	e order for mesh element near coordinates []				

Excerpt from the Proceed IN DIGRAPH COMSOL Conference in Bangalore

Des Desserts		Marchine Concession Marcon March	The second se	Deat Descention	
Pre-Processin	ig			Post-Processing	
ter Property: Keyler	**			Contour Plot:	
Her Contents 30				ried .	
abla Property: Jenyles	·			PM	
Mathix Content: 20				Line Graph:	
Calculate Corposite Pro	perties			Mass Stress .*	
10 29615	We .			Enter Fest Parts	
21 4287.2	21496			N. 8 W. 4 2	15
4287.2	2MPa			Enter Second Point:	
2 1941	7MPa			n (2 n (2 2	4
2140.4	110%			Viscalize Line Cire Plot	-Back
0 2494.1	259a			Surface Plot:	
0.37				Pat	
45 0				Onter First Paris	
db				x n 2	
enalty: 1.32 g	09/13			Enter Second Point:	
Smith No.	don Medining			10 Vi 21	
and the second second	Internet of the second s			Enter Third Point:	
				nn	
				Visualize Surf Surface Plot	. Back
				Max/Min Line Plot:	
				For Nees Stress	
				- PM.	
	1				
		Salang bear system			
		Sector Brand Street			Cancel
					0.000.000.00
	utton to start the simulation arting progress task				
51	arting progress task				
	metry shape order for mesh element hear ood metry shape order for mesh element near ood				
	netry shape order for mesh element near coo				

Excerpt from the Promote And Antime Conference in Bangalore

## LIMITATIONS AND CHALLENGES

- Integrating failure criteria's for composite materials is not possible in an user interface
- Selecting points in the 3-D model is not possible by using mouse events on graphics panel
- Meshing of an complex model is difficult to perform in an user interface

## CONCLUSION

- Composite Material properties were successfully integrated with COMSOL Multiphysics in a user-friendly interface
- GUI created is compatible to be used for any model for the purpose of performing simulation with Composite Material properties
- GUI can also be used to perform some of the important Post-Processing which can be cumbersome to do in COMSOL Multiphysics GUI for an inexperienced user

## **THANKS FOR YOUR ATTENTION**

Excerpt from the Proceedings of the 2014 COMSOL Conference in Bangalore