دانشکده مهندسی مکانیک

درس مکانیک کامپوزیت پیشرفته

تمرین:تمرین دهم

نام و نام خانوادگی: صبا عباس‌زاده منتظری

شماره دانشجویی: ۴۰۳۷۴۳۳۷۶

نام استاد: دکتر محمود مهرداد شکریه

تاریخ: ۲۰ اردیبهشت ۱۴۰۴

Contents

1 Question 1

1. **Why is D matrix not a function of m (ply group) in unsymmetric cross-ply laminates?**

	ϵ_1^0	ϵ_2^0	ϵ_6^0	k_1	k_2	k_6
N_1	192	5.7		$-\frac{171}{m}$		
N_2	5.7	192			$\frac{171}{m}$	
N_6			14.3			
M_1	$-\frac{171}{m}$			64	1.9	
M_2		$\frac{171}{m}$		1.9	64	
M_6						4.7

D matrix is calculated by the formula below:

$$D_{ij} = \frac{h_0^3}{3} \sum_{t=1-\frac{n}{2}}^{\frac{n}{2}} Q_{ij}^{(t)} \left[t^3 - (t-1)^3 \right] \tag{1}$$

In this equation t_i is defined as:

$$t_{ij} = \frac{z_i}{h_0} \tag{2}$$

If a composite consists of 2 degrees, when we change the positions of n layers near to middle line qith the n layers near to middle line at the other side of the middle line, the D matrix remains the same.

For example consider the composite layup $[0_8/90_8]_T$. Changin 4 layers at top of middle line to 4 lines at the bottom of middle line leads to a layup $[0_4/90_4]_2T$ doesn't change the D matrix.

Since **with changing the n layers in respect to middle line, the t_i's remain the same magnitude but the oposite sign. Also the orders of top and bottom magnitudes change so the number of $t-(t-1)$ doesn't change.** This is also valid for cross-plies where $[45_8/-45_8]_T$ and $[45_4/-45_4]_2T$ D matrices are the same. In general, $[\theta_{m/n}/\beta_{m/n}]_{nT}$ D matrices are the same.

2 Question 2

Write a computer code to analyze the stress-strain of general laminated composites.

This code written works for the general laminated composites (unsymmetric and symmetric) under moments and force resultants. Details of this code are explained by comments in the code.

```
%%%Load the stiffness matrix Q_ij, Transformations
clc
clear all
load('stiffness')
load('Transformations')
syms teta
%%%%%Pre-allocation for an array that stores all plies' directions (angles)
all_plies = [];


%%Take the properties of the composite as inputs
%%%%e.g properties of the T300/5208
Ex = input('Enter Ex (GPa): ');
Ey = input('Enter Ey (GPa): ');
Vx = input('Enter Vx: ');
Es = input('Enter Es (GPa): ');

Properties = struct('Ex', Ex, ...
                    'Ey', Ey, ...
                    'Vx', Vx, ...
                    'Es', Es);
% Compute Vy dynamically and add it to the structure. It is computed by
% supposing symmetricity.
Properties.Vy = Properties.Vx * (Properties.Ey / Properties.Ex);
%%Calculation of on-axis stiffness for MaterialData
Q_on=subs(Q,fieldnames(Properties), struct2cell(Properties));



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%Calculation of invariants
U1=double(1/8*(3*Q_on(1,1)+3*Q_on(2,2)+2*Q_on(1,2)+4*Q_on(3,3)));
U2=double(0.5*(Q_on(1,1)-Q_on(2,2)));
U3=double(1/8*(Q_on(1,1)+Q_on(2,2)-2*Q_on(1,2)-4*Q_on(3,3)));
U4=double(1/8*(Q_on(1,1)+Q_on(2,2)+6*Q_on(1,2)-4*Q_on(3,3)));
U5=double(1/8*(Q_on(1,1)+Q_on(2,2)-2*Q_on(1,2)+4*Q_on(3,3)));


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Inputs from
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%user
sy = input('Is the composite symmetric? (yes/no): ', 's');
types=input('enter the number of all kinds of the plies:');
h = input(' Thickness of total layer (mm) = ' ) ;
C= input('c='); %Half thickness of the core used for simplification
n=0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Entring stress resultants
fprintf ('\n Enter Stress resultants (MN/m) :\n\n' );
N1 = input(' N1 = ' );
N2 = input(' N2 = ' );
N6 = input(' N6 = ' );
fprintf ('\n Enter Moment resultants (kN) :\n\n' );
M1 = input(' M1 = ' );
M2 = input(' M2 = ' );
M6 = input(' M6 = ' );
M = [M1 ; M2 ; M6] ;
%%%Make a matrix with 2 column, each row is dedicated to a specific ply(Its
    number and its orientation).
plies=zeros(types,2);
%%%Fill the matrix with the plies' details which are taken as inputs.
for i = 1:types
    % Determine the correct suffix
    if i == 1
        suffix = 'st';
    elseif i == 2
        suffix = 'nd';
    elseif i == 3
        suffix = 'rd';
```

```
    else
        suffix = 'th';
    end

    % Input number of plies
    plies(i,1) = input(sprintf('Enter the number of the %d%s kind of plies:
        ', i, suffix));

    % Input angle and convert to radians
    plies(i,2) = deg2rad(input(sprintf('Enter the angle of the %d%s kind of
        plies: ', i, suffix)));


    % Update total count
    n = n + plies(i,1);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%Make a list with angles of plies n times that are given from user.
%%This is essential for using it in loop
for i = 1:types
    reps = plies(i,1);           % Number of repetitions
    val = plies(i,2);            % Value to repeat
    all_plies = [all_plies; repmat(val, reps, 1)];
end
%%%%%double the plies in inverse order if the user says the composite is
%%%%%symmetrical
if strcmpi(sy, 'YES')
   all_plies = [all_plies; flipud(all_plies)]
end
%%%They become two times if the symmetry condition is applied
if strcmpi(sy, 'YES')
    n=2*n;
else
    n=n;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5

%%% Calculating the numbers essential
```

```
Nc = 2 * C ; % (Thickness of the core / Thickness of each layer )
h0 = h / (n + Nc) ; % (mm) (Thickness of each layer)
zc = C * h0;      %Thickness of core
I_star=(h)^3/12;
Zc_star=(2*zc)/h;
h_star = (1-(Zc_star)^3)*I_star ; % (mm)^3
z2 = h/2 ;  %%%% The magnitude of height above the layer
z1 = h/2 - h0 ;  %%%% The magnitude of height under the layer

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%Preallocation of V_i for A,B,D
V1_A=0;
V2_A=0;
V3_A=0;
V4_A=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
V1_B=0;
V2_B=0;
V3_B=0;
V4_B=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
V1_D=0;
V2_D=0;
V3_D=0;
V4_D=0;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%Calculating V_i for A,B,D for every orientation and adding them together
%%for total V_i calculation
for i = 1:length(all_plies)
    %%%%subtract core thickness if it exists
  if z2 == zc
    z2=-2*zc+z2
    z1=-2*zc+z1
  end
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  V1_A=V1_A+cos(2*all_plies(i))*(z2 - z1);
  V2_A=V2_A+cos(4*all_plies(i))*(z2 - z1);
```

```
    V3_A=V3_A+sin(2*all_plies(i))*(z2 - z1);
    V4_A=V4_A+sin(4*all_plies(i))*(z2- z1);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
    V1_B=V1_B+0.5*cos(2*all_plies(i))*(z2^2 - z1^2);
    V2_B=V2_B+0.5*cos(4*all_plies(i))*(z2^2 - z1^2);
    V3_B=V3_B+0.5*sin(2*all_plies(i))*(z2^2 - z1^2);
    V4_B=V4_B+0.5*sin(4*all_plies(i))*(z2^2 - z1^2);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    V1_D=V1_D+((1/(3))*cos(2*all_plies(i))*(z2^3 - z1^3));
    V2_D=V2_D+((1/(3))*cos(4*all_plies(i))*(z2^3 - z1^3));
    V3_D=V3_D+((1/(3))*sin(2*all_plies(i))*(z2^3 - z1^3));
    V4_D=V4_D+((1/(3))*sin(4*all_plies(i))*(z2^3 - z1^3));
      %%%%%%%%%%%%%%%%%%building an array for saving the heights at top and
      %%%%%%%%%%%%%%%%%%bottom of layer
    Z(2*i-1)=z2
    Z(2*i)=z1
      %%%%%%%%%%%%%%%%%%subtract a layer thickness
    z2=z2-h0;
    z1=z1-h0;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


    %%%%%%%%%%%%%%%%%%%Q_matrix
    theta=all_plies(i)
    %%Calculation of Positive strain transformation matrix for 45 degree
    TP=subs(T_strain_positive,[teta],theta);
    %%Calculation of Negative stress transformation matrix for 45 degree
    TN=subs(T_stress_negative,[teta],theta);
    % %Calculation of off-axis stiffness for Properties
    Q_off(:,:,i)=TN*Q_on*TP;

    %%%%%%%%%%%%%%%%%%%%%Transformation matrix
    Transformation_strain(:,:,i)=subs(T_strain_positive,[teta],theta);
    Transformation_stress(:,:,i)=subs(T_stress_positive,[teta],theta);

end
%%%%%% Add zero element if we have core
if zc~=0
   mid = ceil(length(Z)/2);
```

```
   Z = [Z(1:mid), 0, Z(mid+1:end)];
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Matrix_A=[U1,V1_A,V2_A;U1,-V1_A,V2_A;U4,0,-V2_A;U5,0,-V2_A;
    0 ,0.5*V3_A, V4_A;0 ,0.5*V3_A ,-V4_A];

Matrix_B=[U1,V1_B,V2_B;U1,-V1_B,V2_B;U4,0,-V2_B;U5,0,-V2_B;
    0 ,0.5*V3_B, V4_B;0 ,0.5*V3_B ,-V4_B];

Matrix_D=[U1, V1_D, V2_D;U1,-V1_D ,V2_D;U4, 0 ,-V2_D;
    U5, 0, -V2_D;0 ,0.5*V3_D, V4_D;0 ,0.5*V3_D ,-V4_D];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
U=[1;U2;U3];
U_A=[h;U2;U3];
U_B=[0;U2;U3];
U_D=[h_star;U2;U3];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
%%%Derivation of A matrix.
A=Matrix_A*U_A;
%%%Derivation of B matrix.
B=Matrix_B*U_B;
%%%Derivation of D matrix.
D=Matrix_D*U_D;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%Making a square matrix of A matrix.
A_square=[A(1),A(3),A(5);A(3),A(2),A(6);A(5),A(6),A(4)];

%%%Making a square matrix of DA matrix.
B_square=[B(1),B(3),B(5);B(3),B(2),B(6);B(5),B(6),B(4)];

%%%Making a square matrix of D matrix.
D_square=[D(1),D(3),D(5);D(3),D(2),D(6);D(5),D(6),D(4)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Stiffness=[A_square B_square
    B_square D_square];

```

```
%%%Inverse square S matrix to calculate d matrix.
C_matrix=inv(Stiffness)*1000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
alpha_matrix=C_matrix(1:3,1:3);
beta_matrix = C_matrix(1:3,4:6) ;
delta_matrix=C_matrix(4:6,4:6);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% General strain (epsilon_o) & curvature (k)
strain_vector = C_matrix * [N1 ; N2 ; N6 ; M1 ; M2 ; M6] ;
epsilon_o = strain_vector(1:3) / 10^3 ; % General strain
k = strain_vector(4:6) / 10^3 ; % General curvature (mm)^(-1)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf ('\n\n---------------------------------------\n' ) ;

Stiffness
fprintf('\n\nA_square (MN/m) = \n\n ') ;
fprintf('%.3f %.3f %.3f\n %.3f %.3f %.3f\n %.3f %.3f %.3f\n\n' ,A_square);
fprintf('B_square (kN) = \n\n ') ;
fprintf('%.3f %.3f %.3f\n %.3f %.3f %.3f\n %.3f %.3f %.3f\n\n' ,B_square);
fprintf('D_square (N.m) = \n\n ') ;
fprintf('%.3f %.3f %.3f\n %.3f %.3f %.3f\n %.3f %.3f %.3f\n\n' ,D_square);
fprintf ('---------------------------------------\n' ) ;

C_matrix
fprintf('\n\nalpha_square (GN/m)^(-1) = \n\n ' ) ;
fprintf('%.3f %.3f %.3f\n %.3f %.3f %.3f\n %.3f %.3f %.3f\n\n' , alpha_matrix);
fprintf('beta_square (MN)^(-1) = \n\n ' ) ;
fprintf('%.3f %.3f %.3f\n %.3f %.3f %.3f\n %.3f %.3f %.3f\n\n' ,beta_matrix);
fprintf('delta_square (kN.m)^(-1) = \n\n ' ) ;
fprintf('%.3f %.3f %.3f\n %.3f %.3f %.3f\n %.3f %.3f %.3f\n\n' , delta_matrix);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
R = 1 ;
```

```
for i = 1 : length(all_plies)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
format short e ;
epsilon_off_up(:,i) = epsilon_o + k * Z(2*i) ;
epsilon_off_down(:,i) = epsilon_o + k * Z(2*i-1) ;
epsilon_off_up(:,i) = round(epsilon_off_up(:,i),6) ;
epsilon_off_down(:,i) = round(epsilon_off_down(:,i),6) ;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
epsilon_on_up(:,i) = Transformation_strain(:,:,i) * epsilon_off_up(:,i) ;
epsilon_on_down(:,i) = Transformation_strain(:,:,i) * epsilon_off_down(:,i) ;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
format short g ;
sigma_off_up(:,i) = Q_off(:,:,i) * epsilon_off_up(:,i) * 1000 ; % (MPa)
sigma_off_down(:,i) = Q_off(:,:,i) * epsilon_off_down(:,i) * 1000 ; % (MPa)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sigma_on_up(:,i) = Transformation_stress(:,:,i) * sigma_off_up(:,i) ; % (MPa)
sigma_on_down(:,i) = Transformation_stress(:,:,i) * sigma_off_down(:,i) ; % (MPa)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf ('----------------------------------------' ) ;
fprintf ('\n\n ply %d  ' ,i) ;
fprintf ('\n\nFor the %d-radian ply : \n\n' ,deg2rad(all_plies(i))) ;
fprintf ('sigma_off_up (MPa) = \n\n %.3f\n %.3f\n %.3f\n\n' ,sigma_off_up(:,i)) ;
fprintf ('sigma_off_down (MPa) = \n\n %.3f\n %.3f\n %.3f\n\n' ,sigma_off_down(:,i));
fprintf ('sigma_on_up (MPa) = \n\n %.3f\n %.3f\n %.3f\n\n' ,sigma_on_up(:,i)) ;
fprintf ('sigma_on_down (MPa) = \n\n %.3f\n %.3f\n %.3f\n\n' ,sigma_on_down(:,i)) ;
fprintf ('epsilon_off_up = \n\n %.6f\n %.6f\n %.6f\n\n' ,epsilon_off_up(:,i)) ;
fprintf ('epsilon_off_down = \n\n %.6f\n %.6f\n %.6f\n\n' ,epsilon_off_down(:,i)) ;
fprintf ('epsilon_on_up = \n\n %.6f\n %.6f\n %.6f\n\n' ,epsilon_on_up(:,i)) ;
fprintf ('epsilon_on_down = \n\n %.6f\n %.6f\n %.6f\n\n' ,epsilon_on_down
```

```
    (:,i)) ;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sigma_off(:,2*i) = sigma_off_up(:,i) ;
sigma_off(:,2*i-1) = sigma_off_down(:,i) ;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sigma_on(:,2*i) = sigma_on_up(:,i) ;
sigma_on(:,2*i-1) = sigma_on_down(:,i) ;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
epsilon_off(:,2*i) = epsilon_off_up(:,i) ;
epsilon_off(:,2*i-1) = epsilon_off_down(:,i) ;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
epsilon_on(:,2*i) = epsilon_on_up(:,i) ;
epsilon_on(:,2*i-1) = epsilon_on_down(:,i) ;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Null = zeros (3,1) ;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%add a zero element in middle if we have core
if zc ~= 0
    mid = ceil(16/2);
    sigma_off = [sigma_off(:, 1:mid), Null, sigma_off(:, mid+1:end)];
    sigma_on = [sigma_on(:, 1:mid), Null, sigma_on(:, mid+1:end)];
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sigma_off_1 = round(double(sigma_off(1,:)), 2);
sigma_off_2 = round(double(sigma_off(2,:)), 2);
sigma_off_6 = round(double(sigma_off(3,:)), 2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sigma_on_x = round(double(sigma_on(1,:)), 2);
sigma_on_y = round(double(sigma_on(2,:)), 2);
sigma_on_s = round(double(sigma_on(3,:)), 2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%creating labels of sigma magnitudes of plots
Z = round(double(Z), 2);
label1 = cellstr(num2str([sigma_off_1(:) Z(:)], '%.2f , %.3f'));
label2 = cellstr(num2str([sigma_off_2(:) Z(:)], '%.2f , %.3f'));
```

```
label6 = cellstr(num2str([sigma_off_6(:) Z(:)], '%.2f , %.3f'));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
labelx = cellstr(num2str([sigma_on_x(:) Z(:)] , '%.2f , %.3f'));
labely = cellstr(num2str([sigma_on_y(:) Z(:)] , '%.2f , %.3f'));
labels = cellstr(num2str([sigma_on_s(:) Z(:)] , '%.2f , %.3f'));


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Off-axis stresses
F1 = figure('Name','sigma-1(off-axis)','NumberTitle','off');
plot(sigma_off_1,Z,'-o')
title('Off-axis Stress Distribution')
xlabel('\sigma1 (MPa)')
ylabel('Z (mm)')
grid on
text(sigma_off_1(:),Z,label1,'VerticalAlignment','bottom',...
'HorizontalAlignment','left')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
F2 = figure('Name','sigma-2(off-axis)','NumberTitle','off');
plot(sigma_off_2,Z,'-o')
title('Off-axis Stress Distribution')
xlabel('\sigma2 (MPa)')
ylabel('Z (mm)')
grid on
text(sigma_off_2(:),Z,label2,'VerticalAlignment','bottom',...
'HorizontalAlignment','left')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
F3 = figure('Name','sigma-6(off-axis)','NumberTitle','off');
plot(sigma_off_6,Z,'-o')
title('Off-axis Stress Distribution')
xlabel('\sigma6 (MPa)')
ylabel('Z (mm)')
grid on
text(sigma_off_6(:),Z,label6,'VerticalAlignment','bottom',...
'HorizontalAlignment','left')



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%On-axis stresses
F4 = figure('Name','sigma-x(on-axis)','NumberTitle','off');
```

```matlab
plot(sigma_on_x,Z,'-o')
title('On-axis Stress Distribution')
xlabel('\sigmax (MPa)')
ylabel('Z (mm)')
grid on
text(sigma_on_x(:),Z,labelx,'VerticalAlignment','bottom',...
'HorizontalAlignment','left')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
F5 = figure('Name','sigma-y(on-axis)','NumberTitle','off');
plot(sigma_on_y,Z,'-o')
title('On-axis Stress Distribution')
xlabel('\sigmay (MPa)')
ylabel('Z (mm)')
grid on
text(sigma_on_y(:),Z,labely,'VerticalAlignment','bottom',...
'HorizontalAlignment','left')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
F6 = figure('Name','sigma-s(on-axis)','NumberTitle','off');
plot(sigma_on_s,Z,'-o')
title('On-axis Stress Distribution')
xlabel('\sigmas (MPa)')
ylabel('Z (mm)')
grid on
text(sigma_on_s(:),Z,labels,'VerticalAlignment','bottom',...
'HorizontalAlignment','left')
```

3 Question 3

Using CLT, consider a symmetric $[0_3/90_2/45_1/-45_3]_S$ and an unsymmetric $[0_3/90_2/45_1/-45_3]_{2T}$ laminated composites made of T300/5208, under N_i and M_i (Magnitudes to your choice). Calculate the on-axis strains and stresses of each layer.

Results of $[0_3/90_2/45_1/-45_3]_S$:

The entries:

Enter Ex (GPa): 181

Enter Ey (GPa): 10.3

Enter Vx: 0.28

Enter Es (GPa): 7.17

Is the composite symmetric? (yes/no): yes

enter the number of all kinds of the plies:4

Thickness of total layer (mm) = 2.25

c=0

Enter Stress resultants (MN/m):

N1 = 2

N2 = 1

N6 = 1

Enter Moment resultants (KN):

M1 = 2

M2 = 1

M6 = 1

Enter the number of the 1st kind of plies: 3

Enter the angle of the 1st kind of plies: 0

Enter the number of the 2nd kind of plies: 2

Enter the angle of the 2nd kind of plies: 90

Enter the number of the 3rd kind of plies: 1

Enter the angle of the 3rd kind of plies: 45

Enter the number of the 4th kind of plies: 3

Enter the angle of the 4th kind of plies: -45

Results: In this part, I showed the on-axis and off-axis stresses by stress distribution plots to over all the results in a compact form.

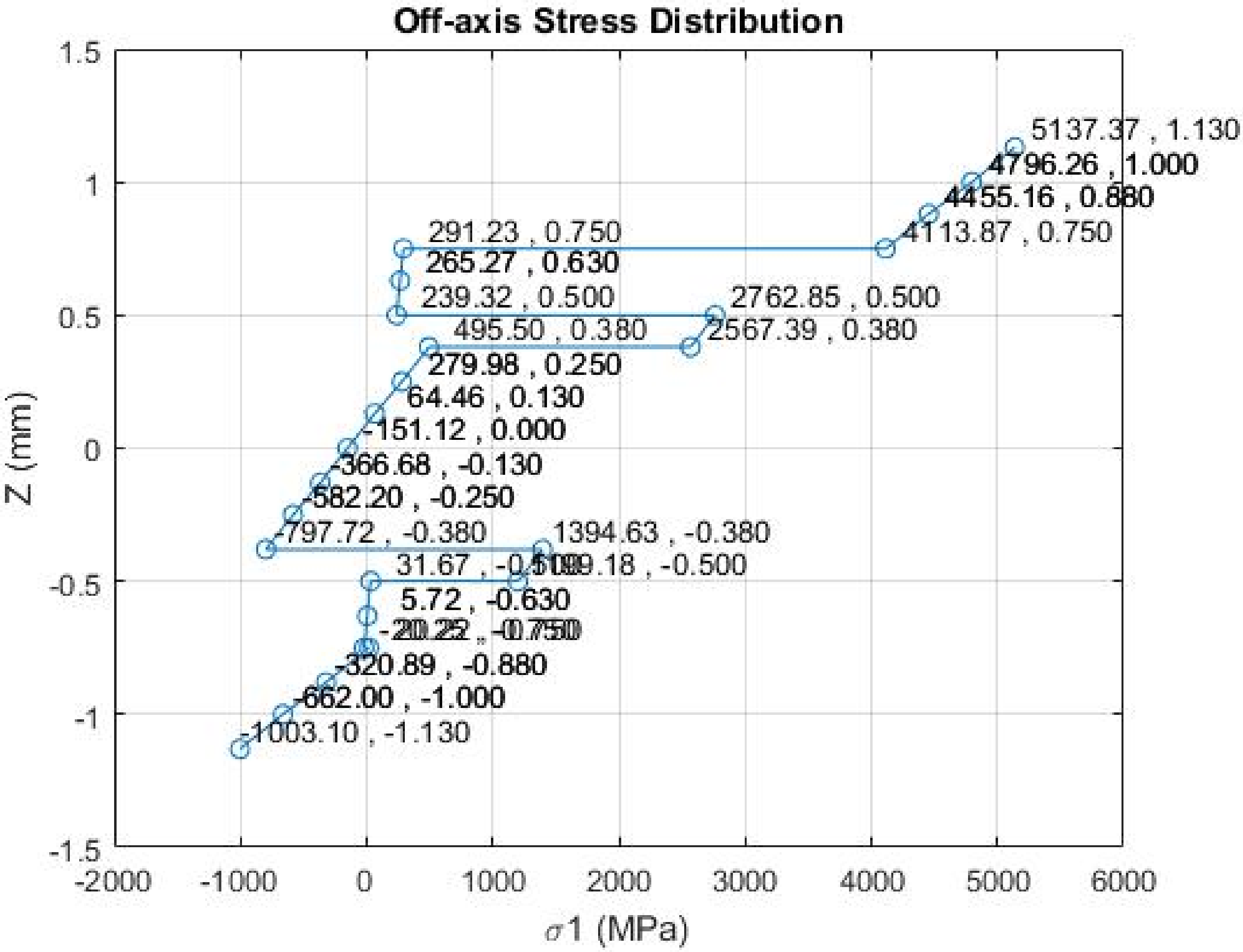


Figure 1: σ_1 off-axis stress distribution

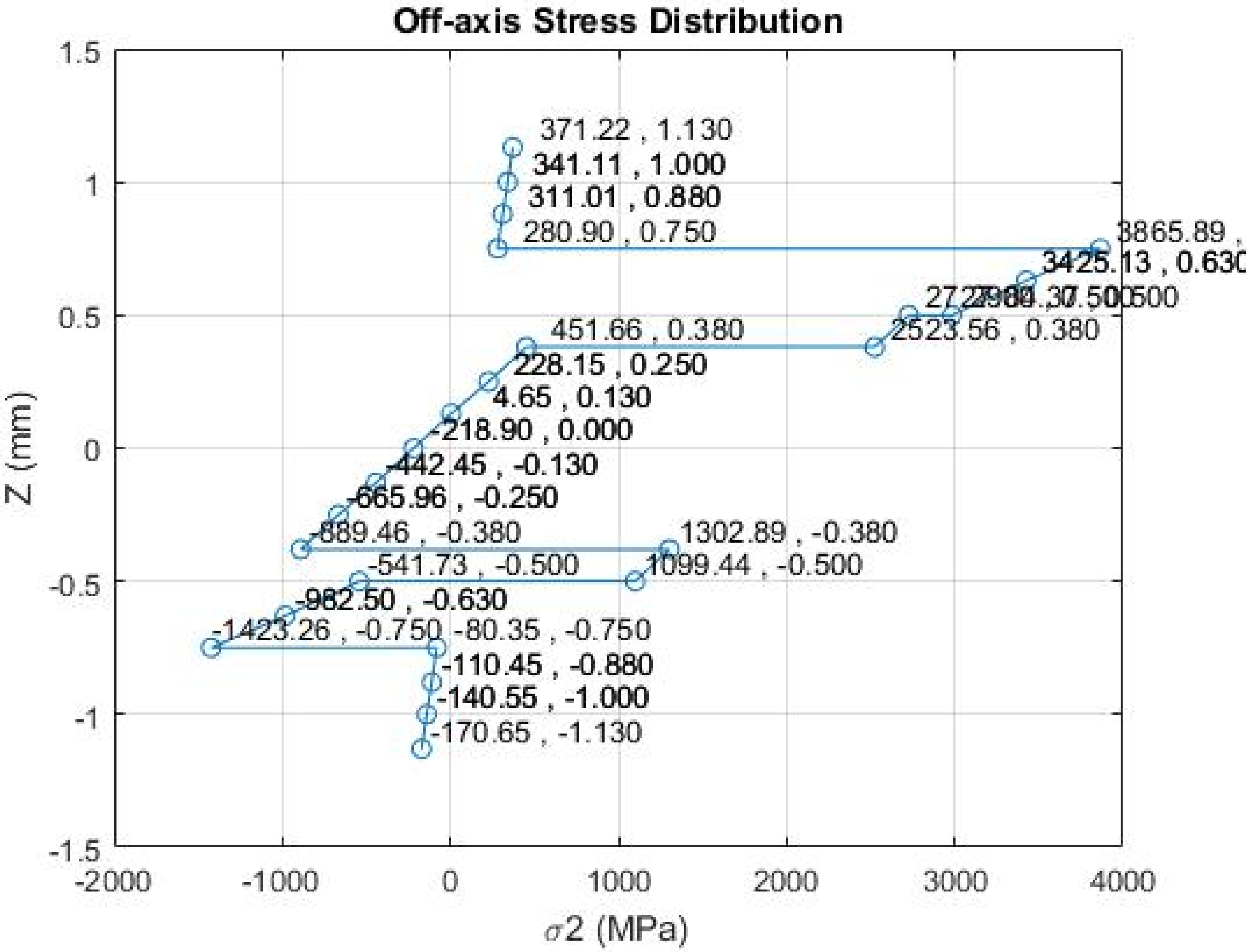


Figure 2: σ_2 off-axis stress distribution

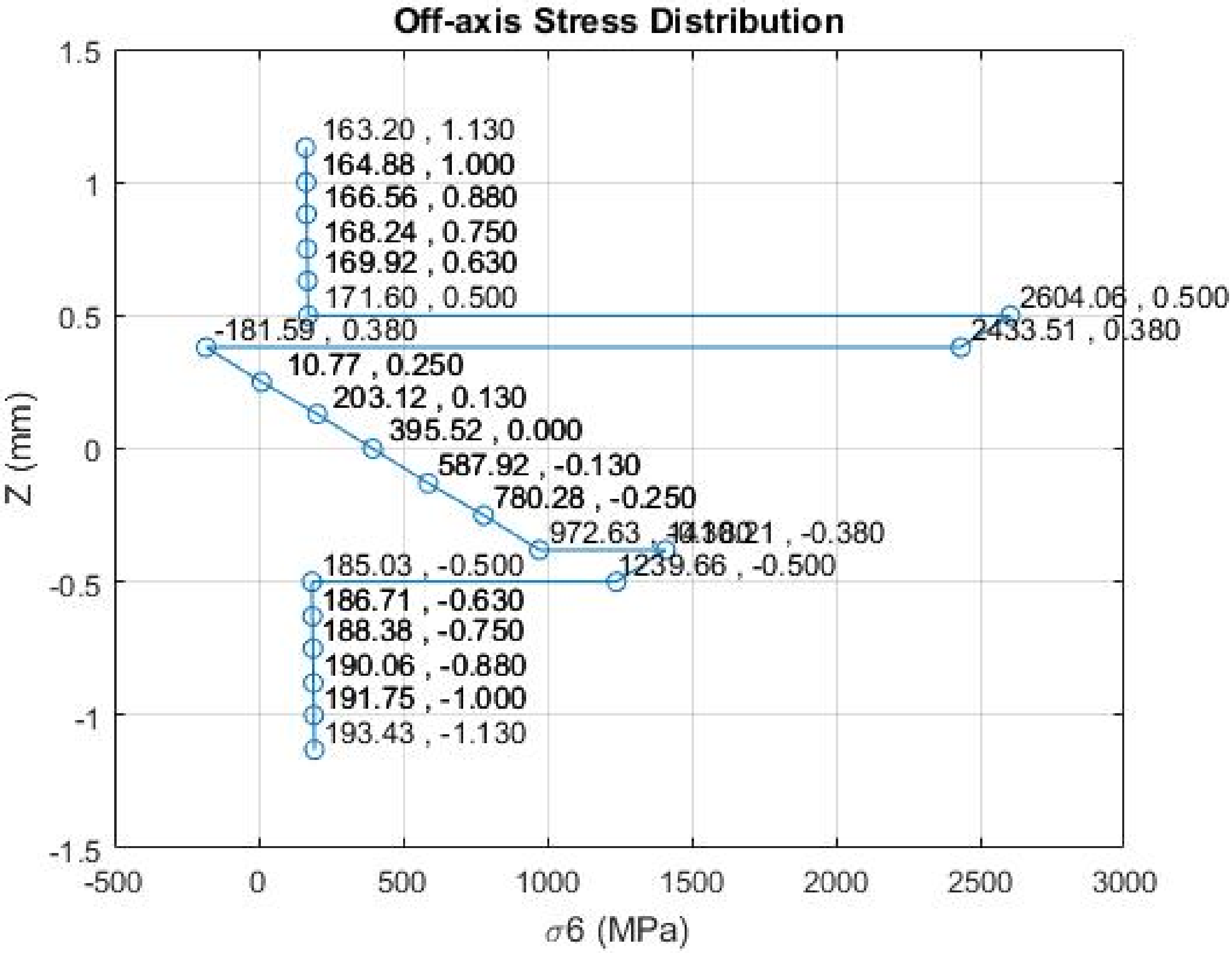


Figure 3: σ_3 off-axis stress distribution

$[90_4/0_4]_2$:

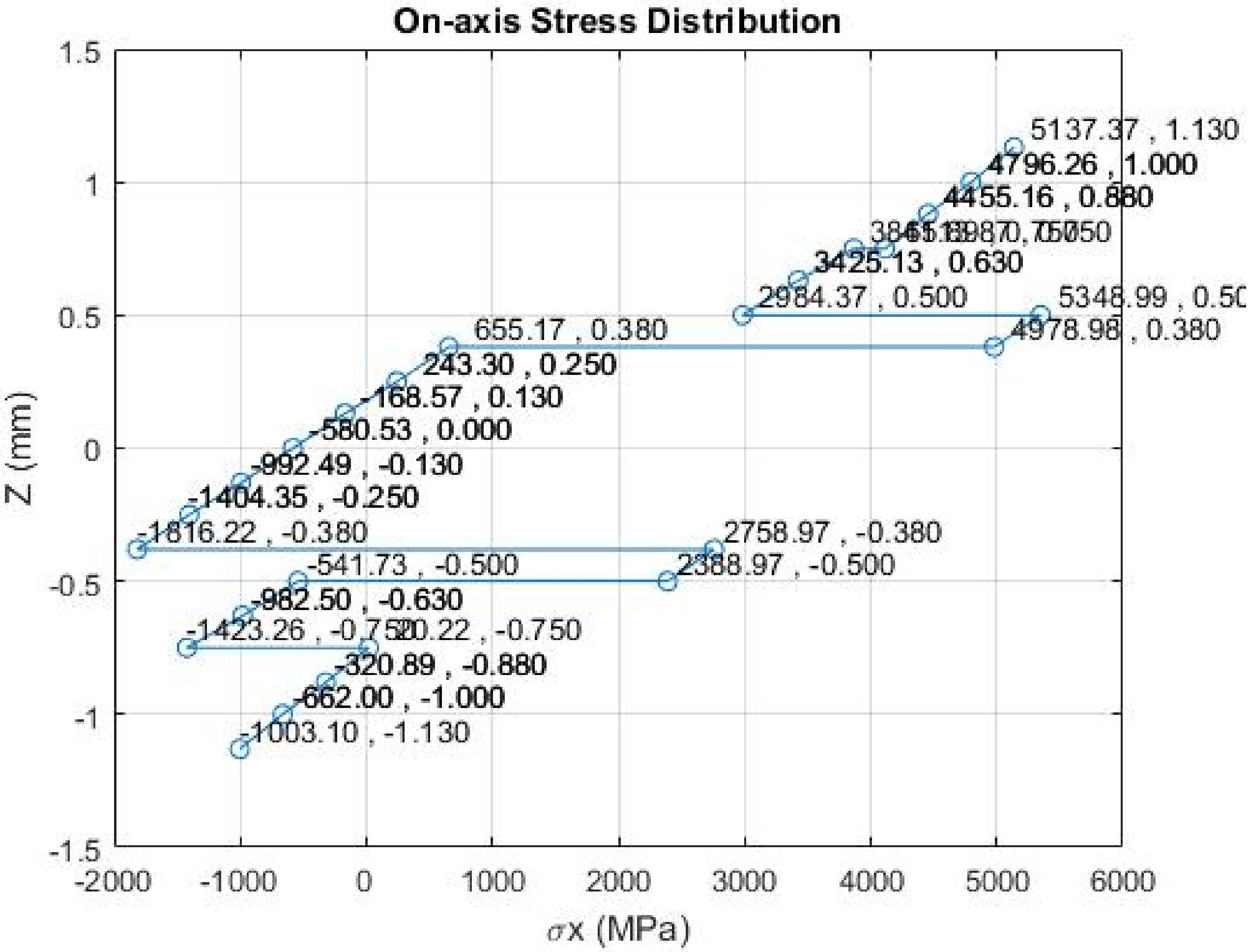


Figure 4: σ_x On-axis stress distribution

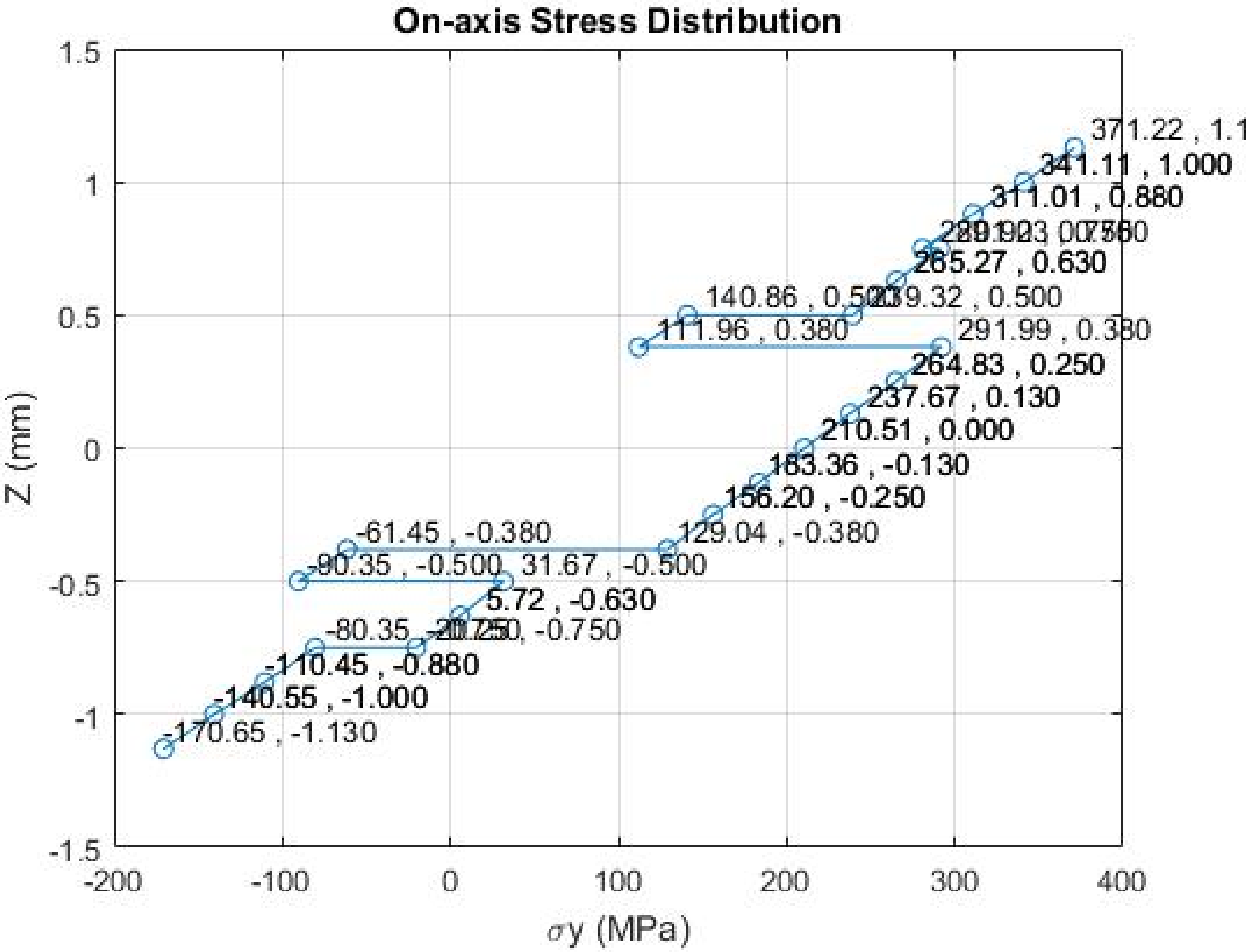


Figure 5: σ_y On-axis stress distribution

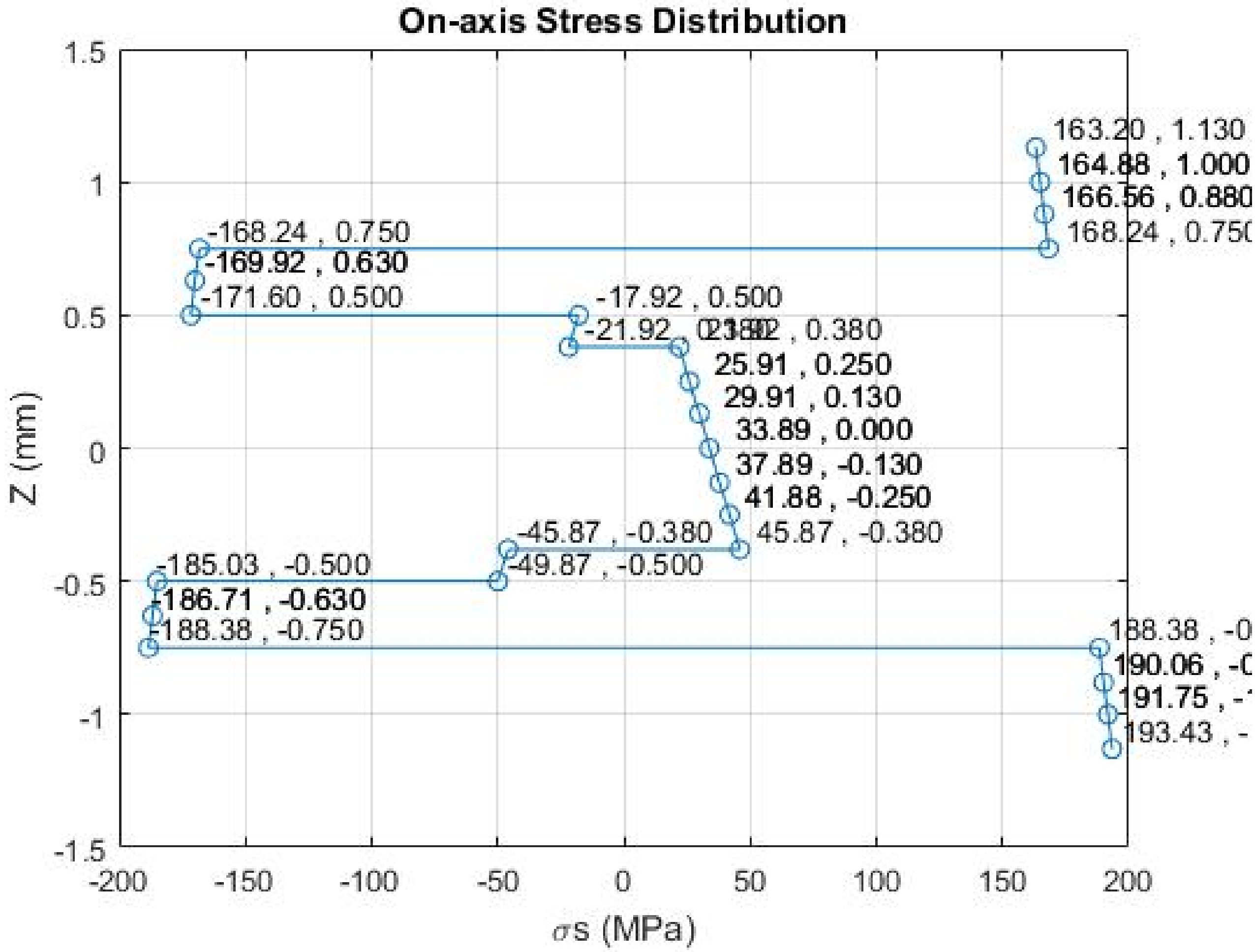


Figure 6: σ_s On-axis stress distribution

Results of $[0_3/90_2/45_1/-45_3]_{2T}$**:**

The entries:

Enter Ex (GPa): 181

Enter Ey (GPa): 10.3

Enter Vx: 0.28

Enter Es (GPa): 7.17

Is the composite symmetric? (yes/no): NO

enter the number of all kinds of the plies:8

Thickness of total layer (mm) = 2.25

c=0

Enter Stress resultants (MN/m) :

N1 = 2

N2 = 1

N6 = 1

Enter Moment resultants (KN) :

M1 = 2

M2 = 1

M6 = 1

Enter the number of the 1st kind of plies: 3

Enter the angle of the 1st kind of plies: 0

Enter the number of the 2nd kind of plies: 2

Enter the angle of the 2nd kind of plies: 90

Enter the number of the 3rd kind of plies: 1

Enter the angle of the 3rd kind of plies: 45

Enter the number of the 4th kind of plies: 3

Enter the angle of the 4th kind of plies: -45

Enter the number of the 5th kind of plies: 3

Enter the angle of the 5th kind of plies: 0

Enter the number of the 6th kind of plies: 2

Enter the angle of the 6th kind of plies: 90

Enter the number of the 7th kind of plies: 1

Enter the angle of the 7th kind of plies: 45

Enter the number of the 8th kind of plies: 3

Enter the angle of the 8th kind of plies: -45

Result:

In this part also, I showed the on-axis and off-axis stresses by stress distribution plots to over all the results in a compact form.

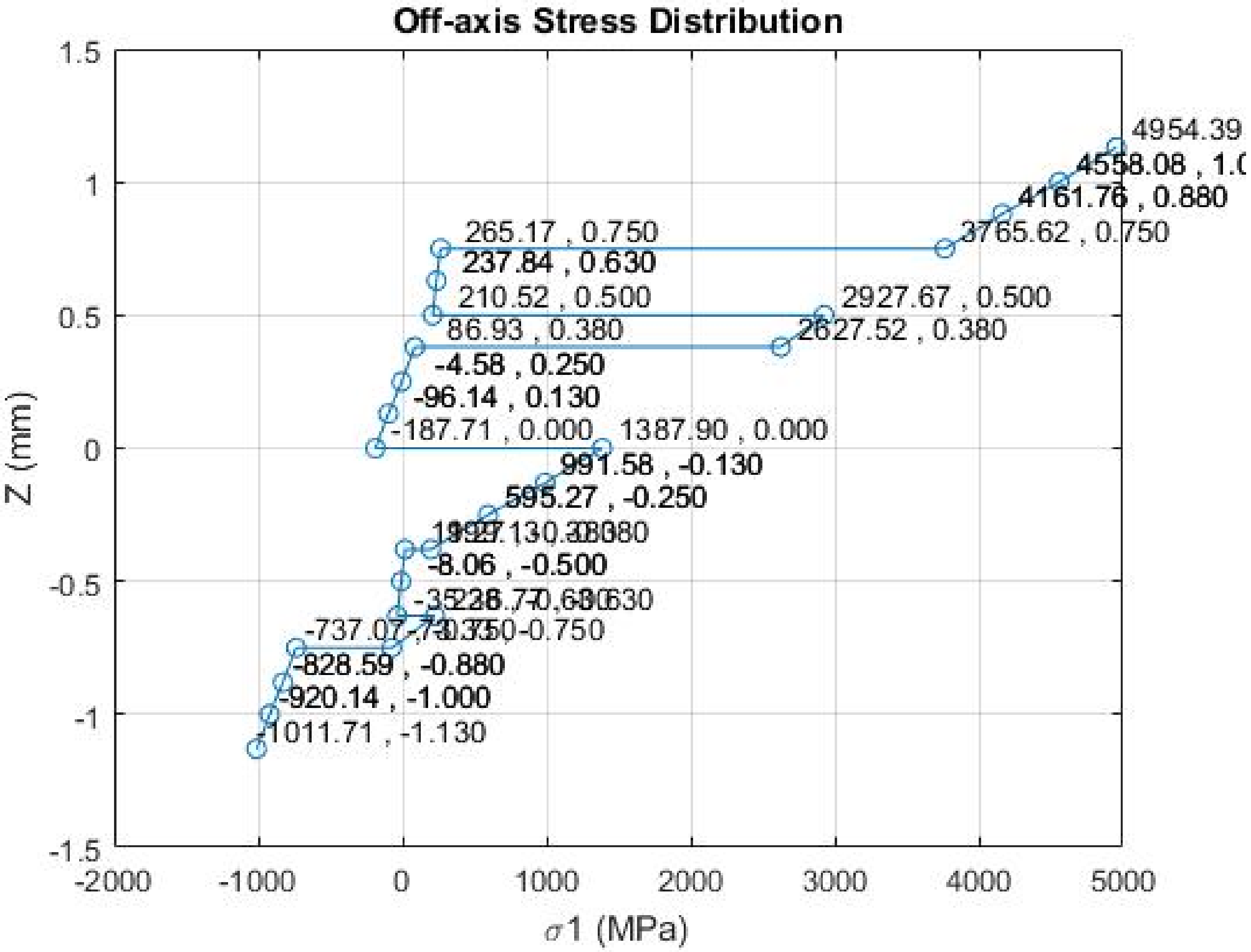


Figure 7: σ_1 off-axis stress distribution

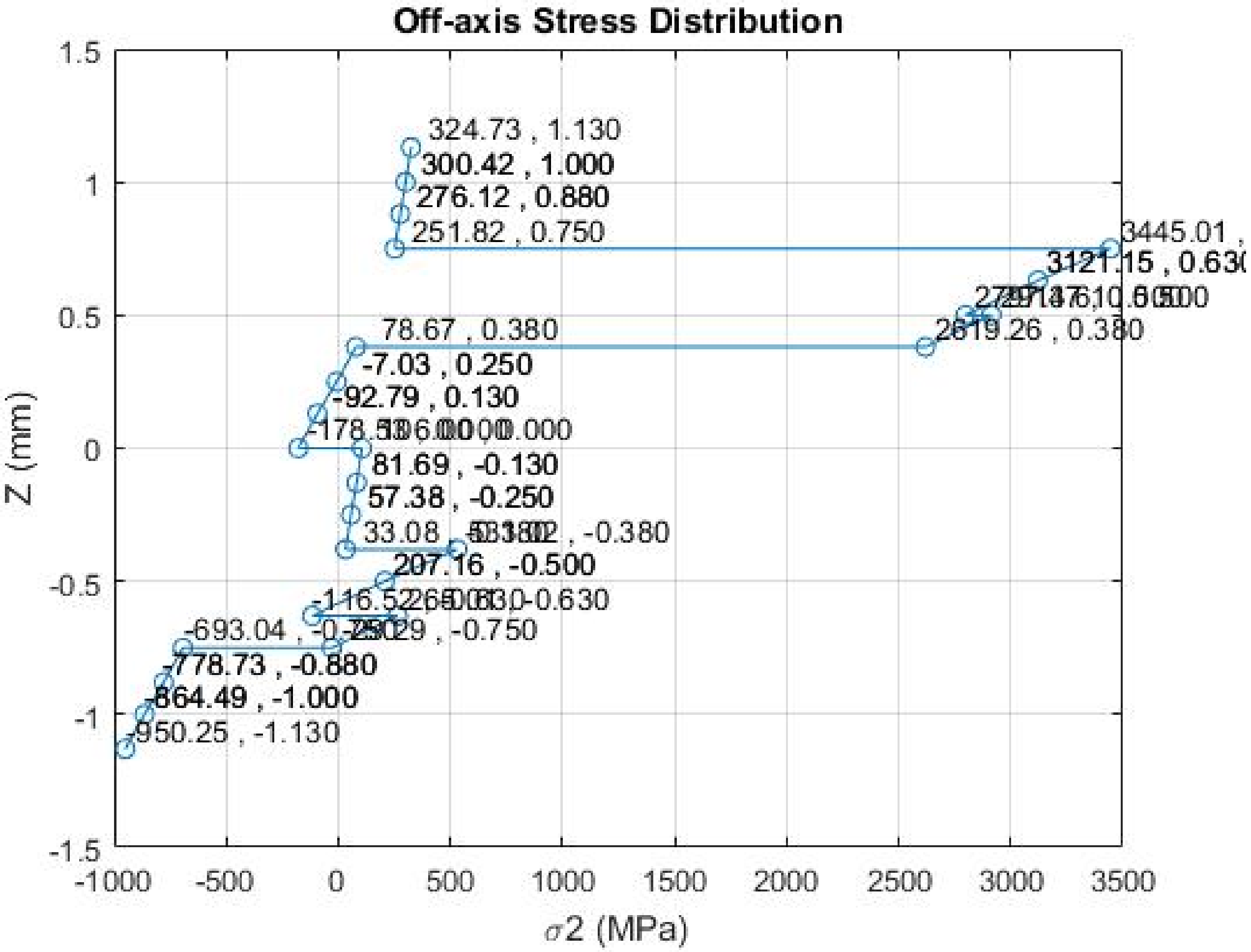


Figure 8: σ_2 off-axis stress distribution

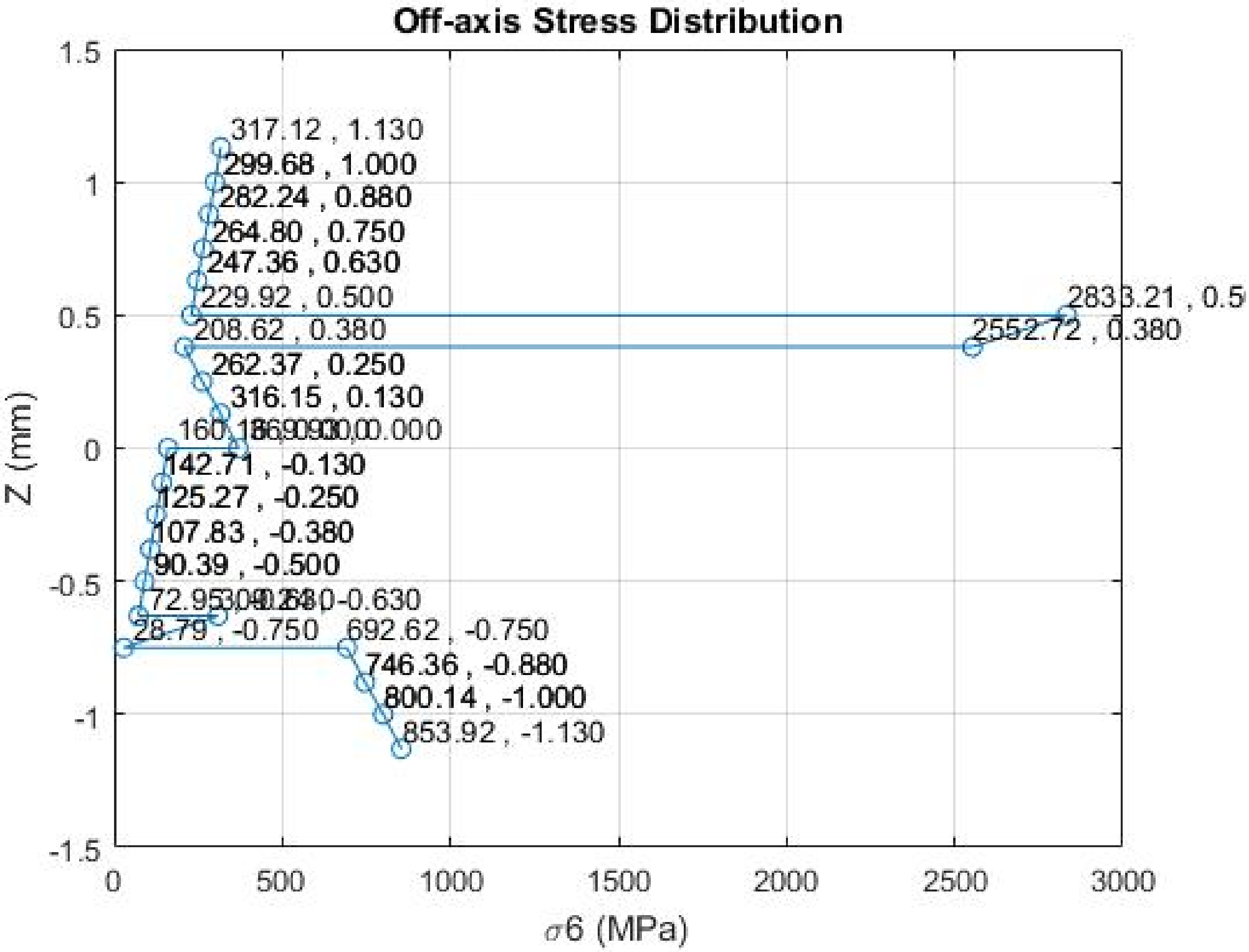


Figure 9: σ_3 off-axis stress distribution

$[90_4/0_4]_2$:

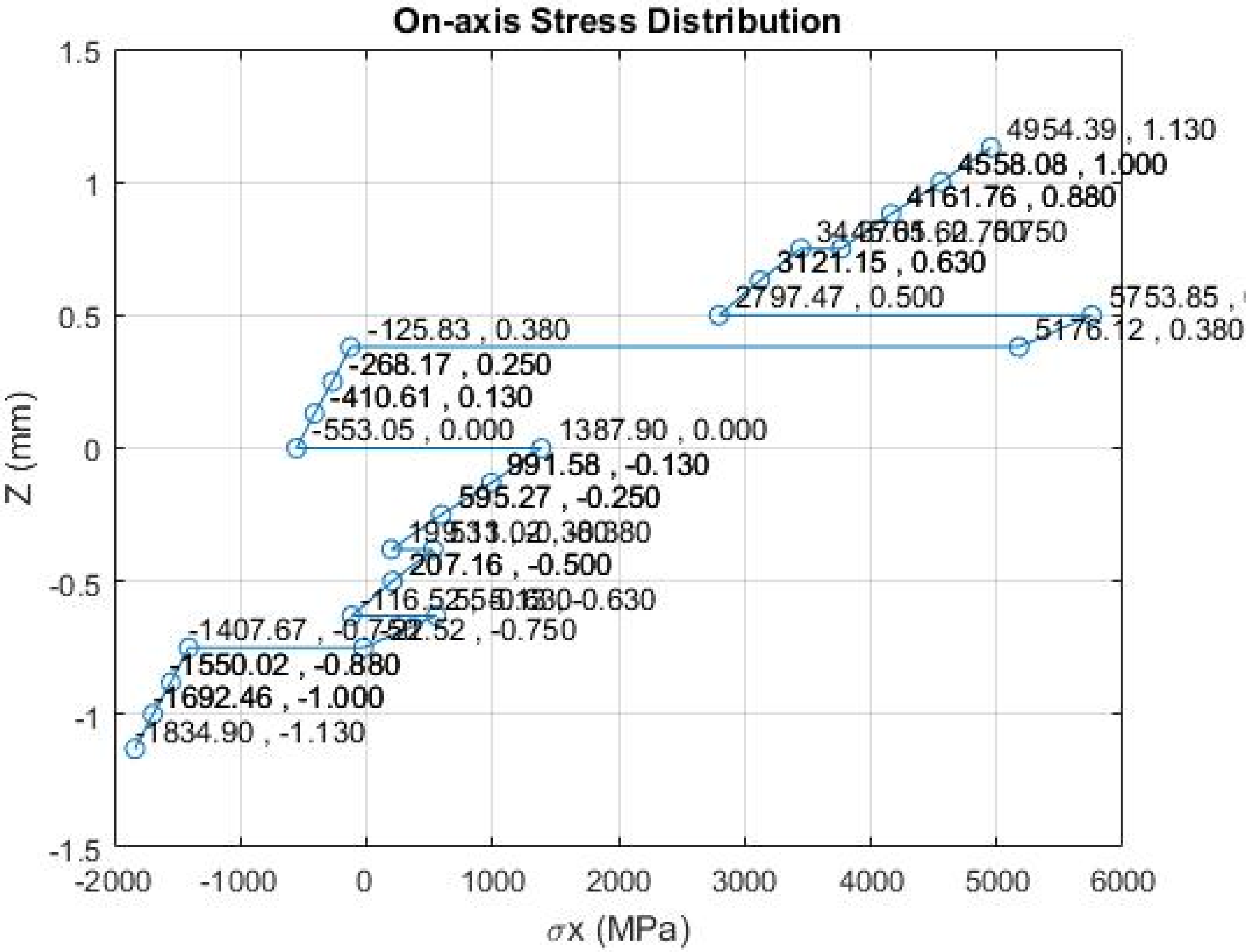


Figure 10: σ_x On-axis stress distribution

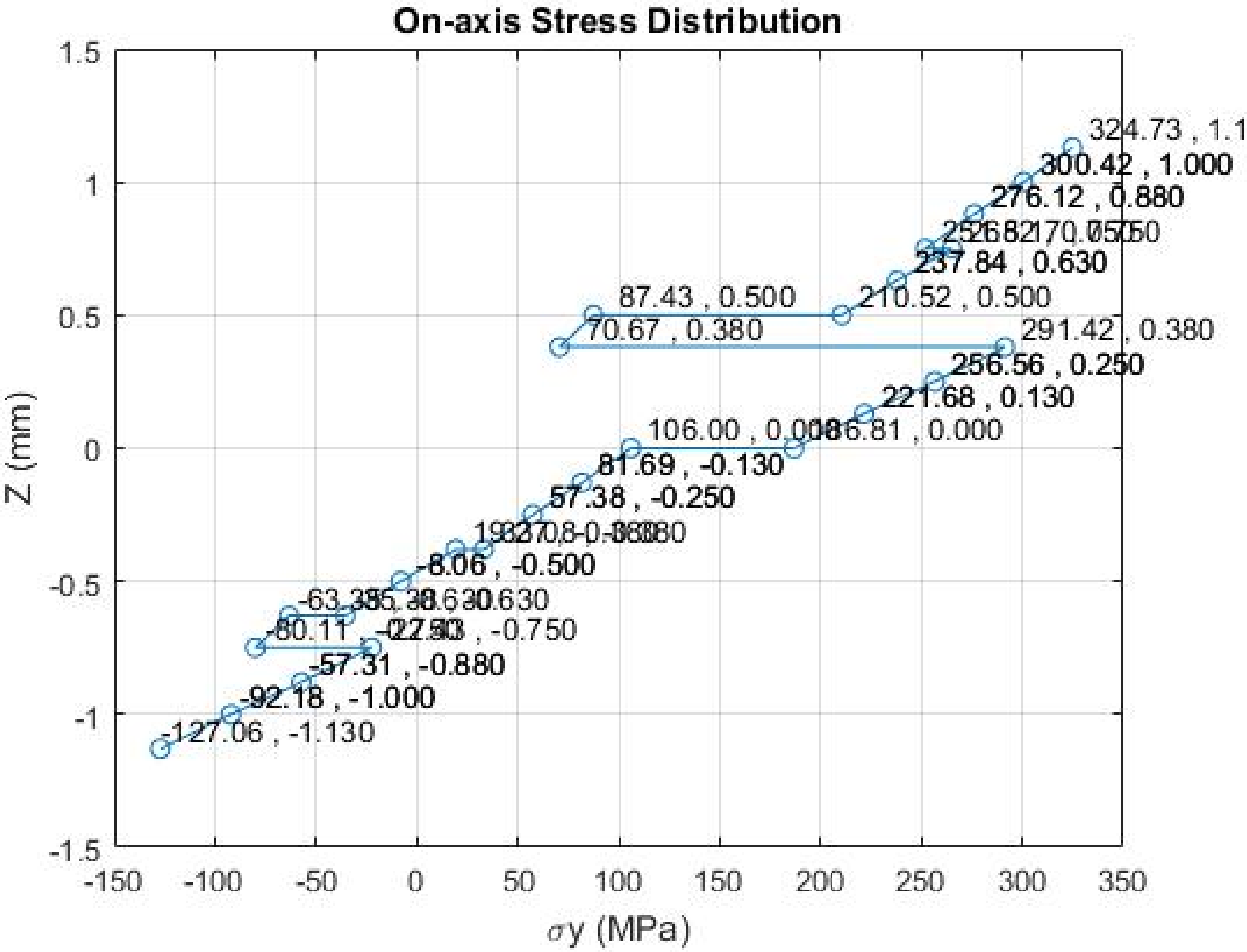


Figure 11: σ_y On-axis stress distribution

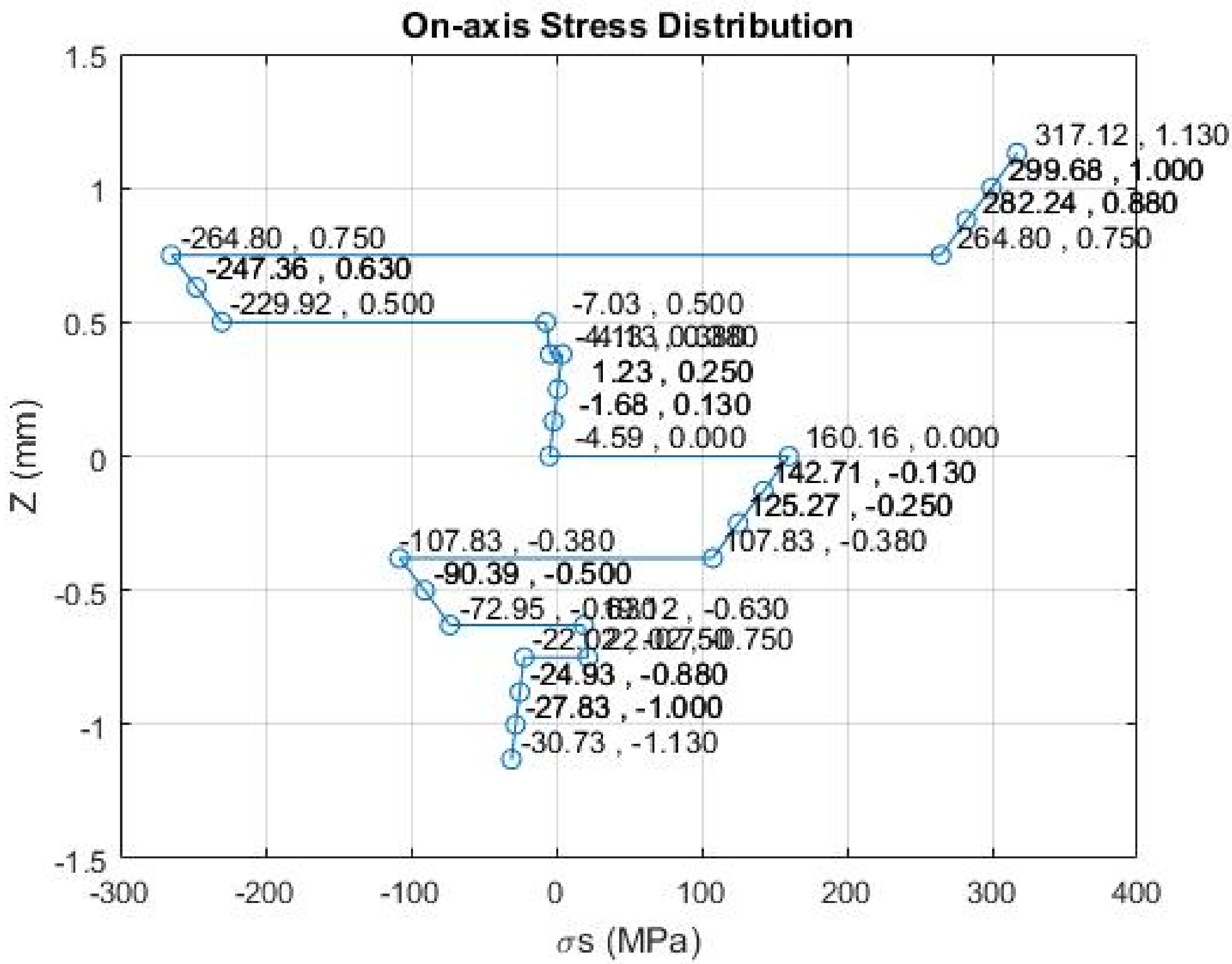


Figure 12: σ_s On-axis stress distribution

4 Question 4

5. **Use a finite element program and calculate the on-axis strains and stresses of each layer of question 3. Compare the results of CLT and FEM.**

In this Section we are going to explain the steps one by one.

Part:

We Sketched a 1 mm*1 mm square (The dimensions really don't matter).

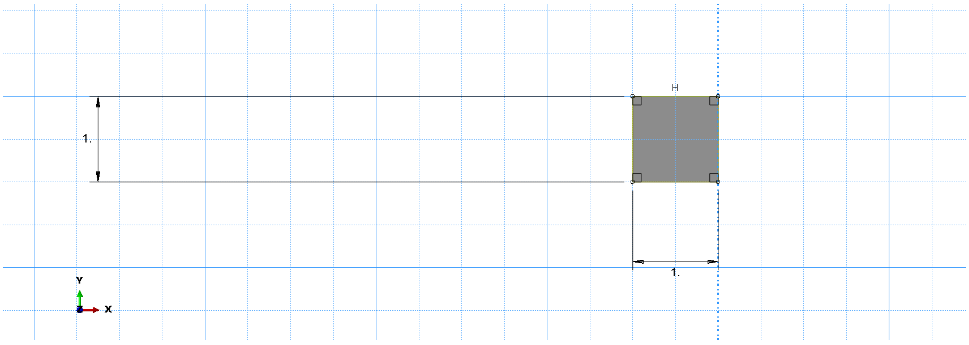


Figure 13: Sketching Part

We also built 3D deformable part as the core. Property:
In the material part, we define the properties of T300/5208 Material. We inserted these informations in the lamina part.

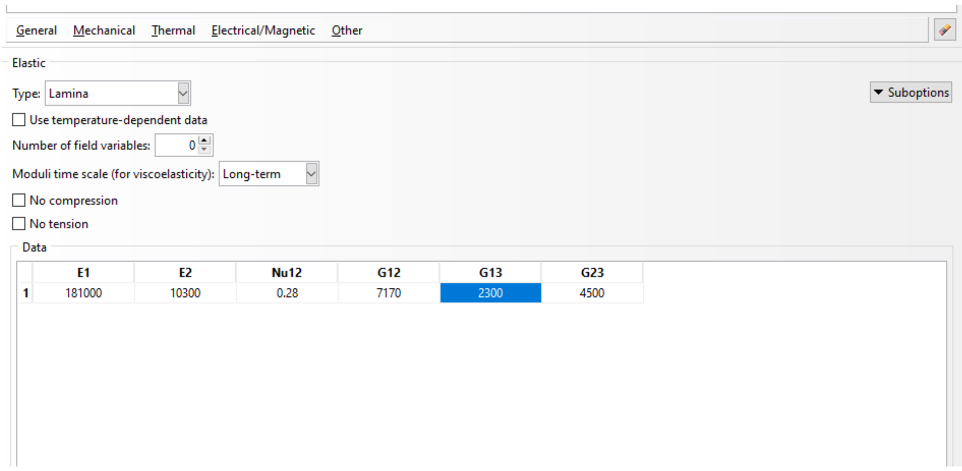
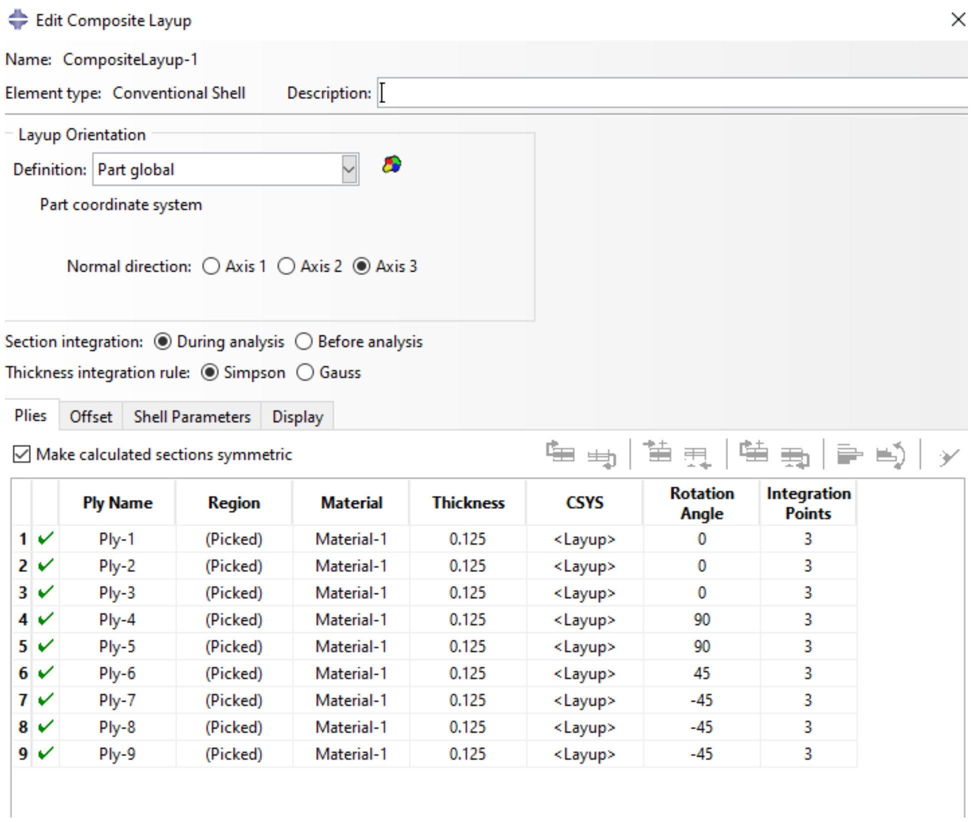


Figure 14: Inserting On-axis engineering constants

The layup is specified like the picture below:



	Ply Name	Region	Material	Thickness	CSYS	Rotation Angle	Integration Points
1	Ply-1	(Picked)	Material-1	0.125	<Layup>	0	3
2	Ply-2	(Picked)	Material-1	0.125	<Layup>	0	3
3	Ply-3	(Picked)	Material-1	0.125	<Layup>	0	3
4	Ply-4	(Picked)	Material-1	0.125	<Layup>	90	3
5	Ply-5	(Picked)	Material-1	0.125	<Layup>	90	3
6	Ply-6	(Picked)	Material-1	0.125	<Layup>	45	3
7	Ply-7	(Picked)	Material-1	0.125	<Layup>	-45	3
8	Ply-8	(Picked)	Material-1	0.125	<Layup>	-45	3
9	Ply-9	(Picked)	Material-1	0.125	<Layup>	-45	3

Figure 15: Stacking Sequence

The layup is illustrated in the figure below:

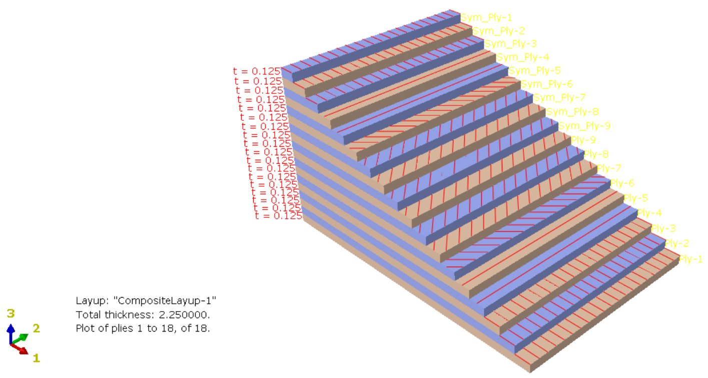


Figure 16: Stacking sequence

Step and Assembly:

Subsequently, we created a static-general step and added the part in the assembly.

Loading:

In the loading module, we applied moments and forces as It is mentioned in the table below. We avoid making boundary conditions to avoid stress concentrations.

Name	**Load Value**
Mx-1	-2000
Mx-2	-2000
Myy-1	-1000
Myy-2	-1000
N1	-2000
N1-2	-2000
N2	-1000
N2-2	-1000
N3-bottom	-1000
N3-left	1000
N3-right	1000
N3-top	-1000

Table 1: Load Values for Step-1

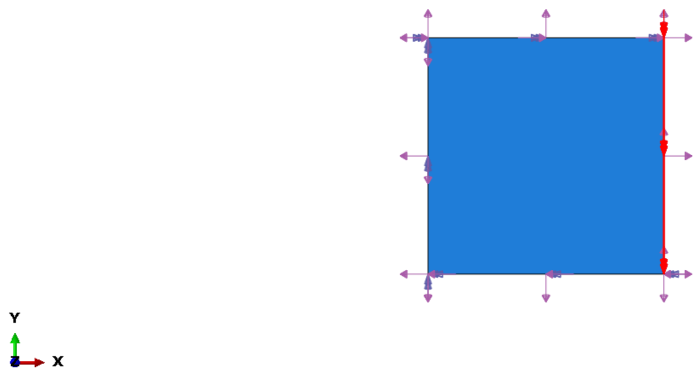


Figure 17: Loading directions

Mesh:

The element type was S4R and quad meshing in chosen for the process.The size of mesh doesn't really matter since the geometry is so simple and 4 number of meshes can give us a good result.

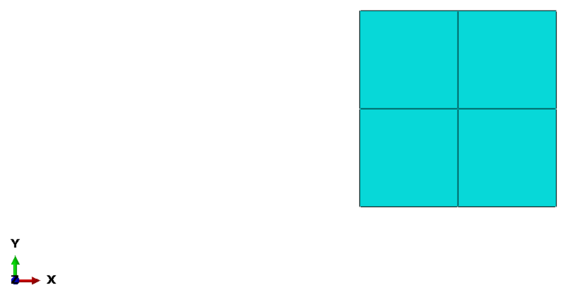


Figure 18: Meshed part

On-axis Results:

For e.g The on-axis S11 stress in the Ply-1 is illustrated below:

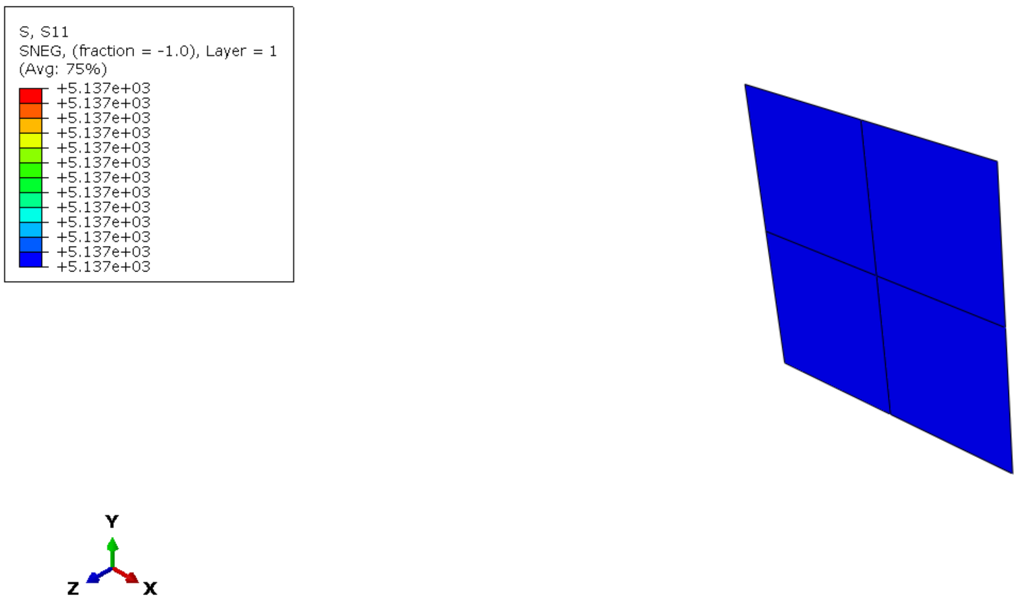


Figure 19: On-axis Stress in 1-direction for Ply1

This shows total compliance with Matlab code results. The results show that the magnitudes of stresses are very close to each other. We extracted the results by **.rpt** files. The distributions of stresses are illustrated in the figures below which are very close to the magnitudes of figures 4 and 5.

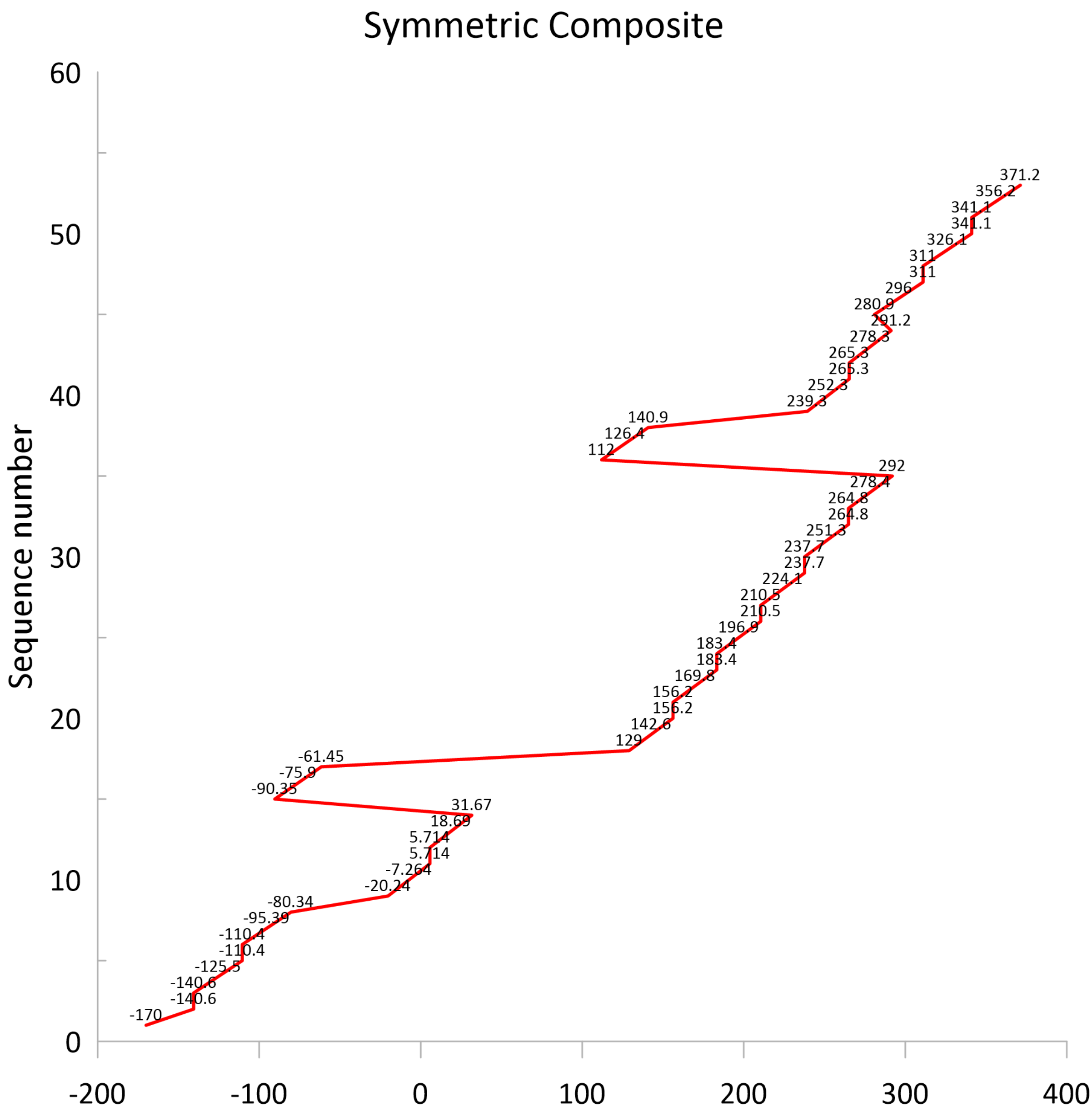


Figure 21: On-axis Stresses in 2-direction from abaqus results

Analysis of $[0_3/90_2/45_1/-45_3]_{2T}$**:**

The analysis is mostly like the previous one, but the layup changes.

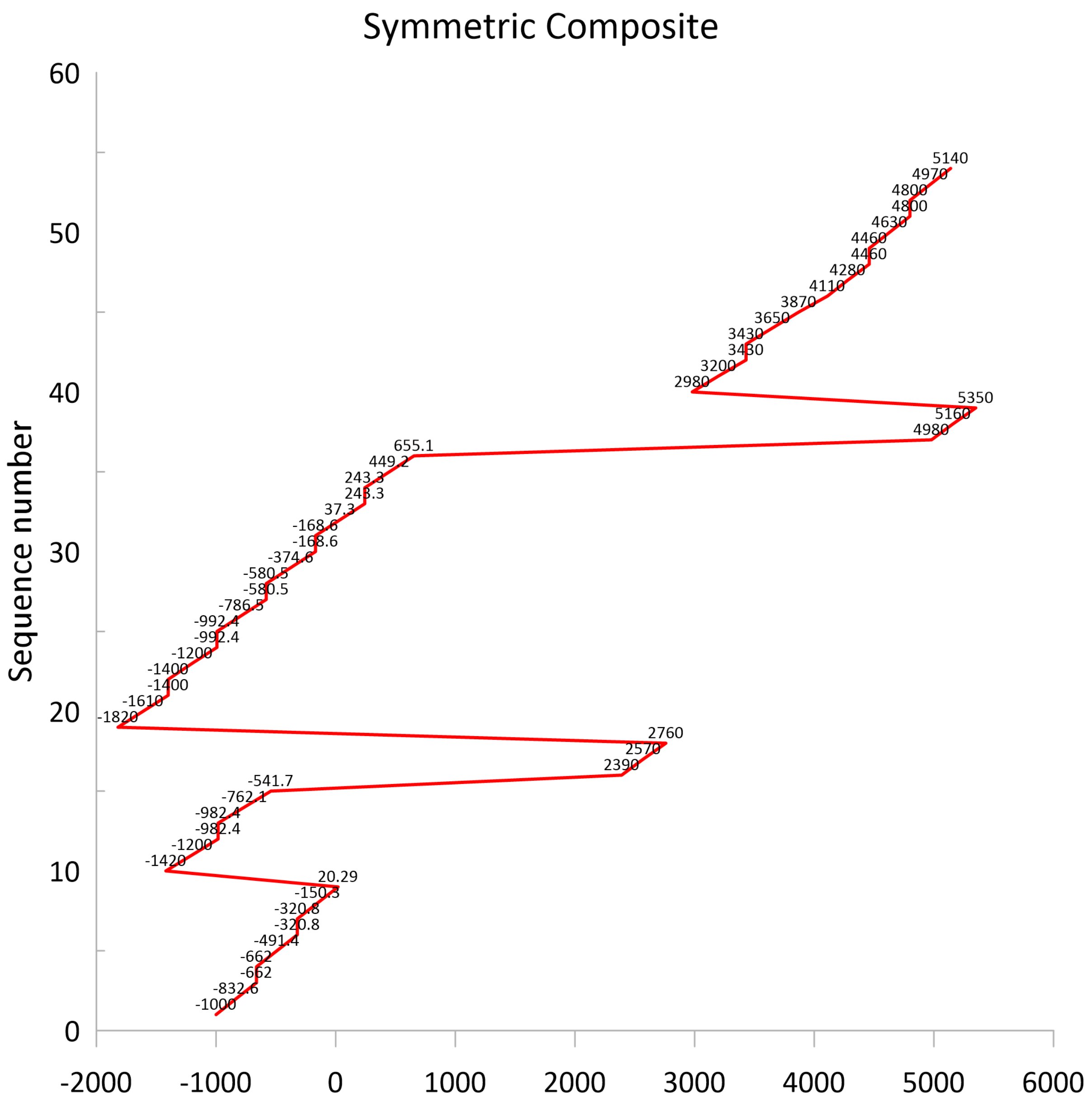


Figure 20: On-axis Stresses in 1-direction from abaqus results

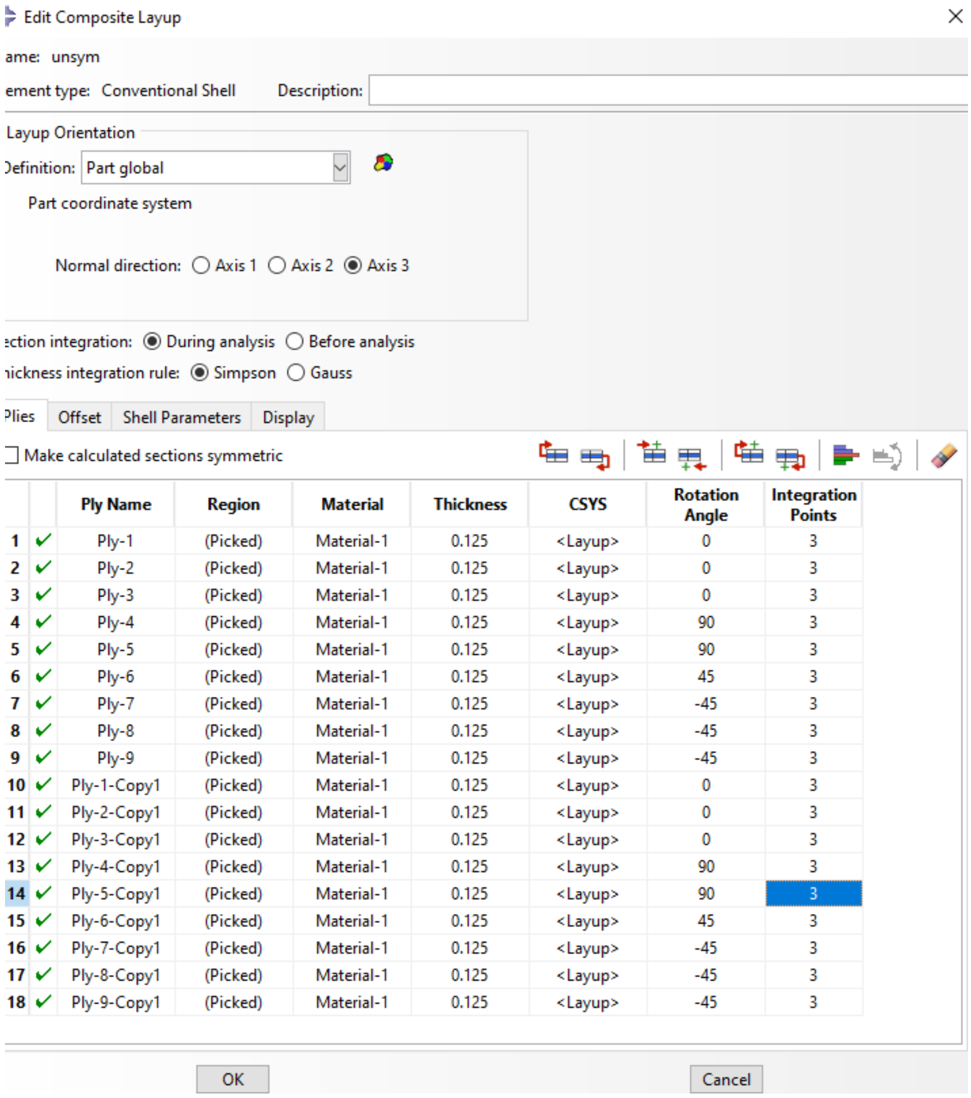


Figure 22: Layup of $[0_3/90_2/45_1/-45_3]_{2T}$

The results of this analysis is also very close to CLT analysis. The stress distributions are mentioned in the figures below and they are the same as the distributions in the figures 10 and 11.

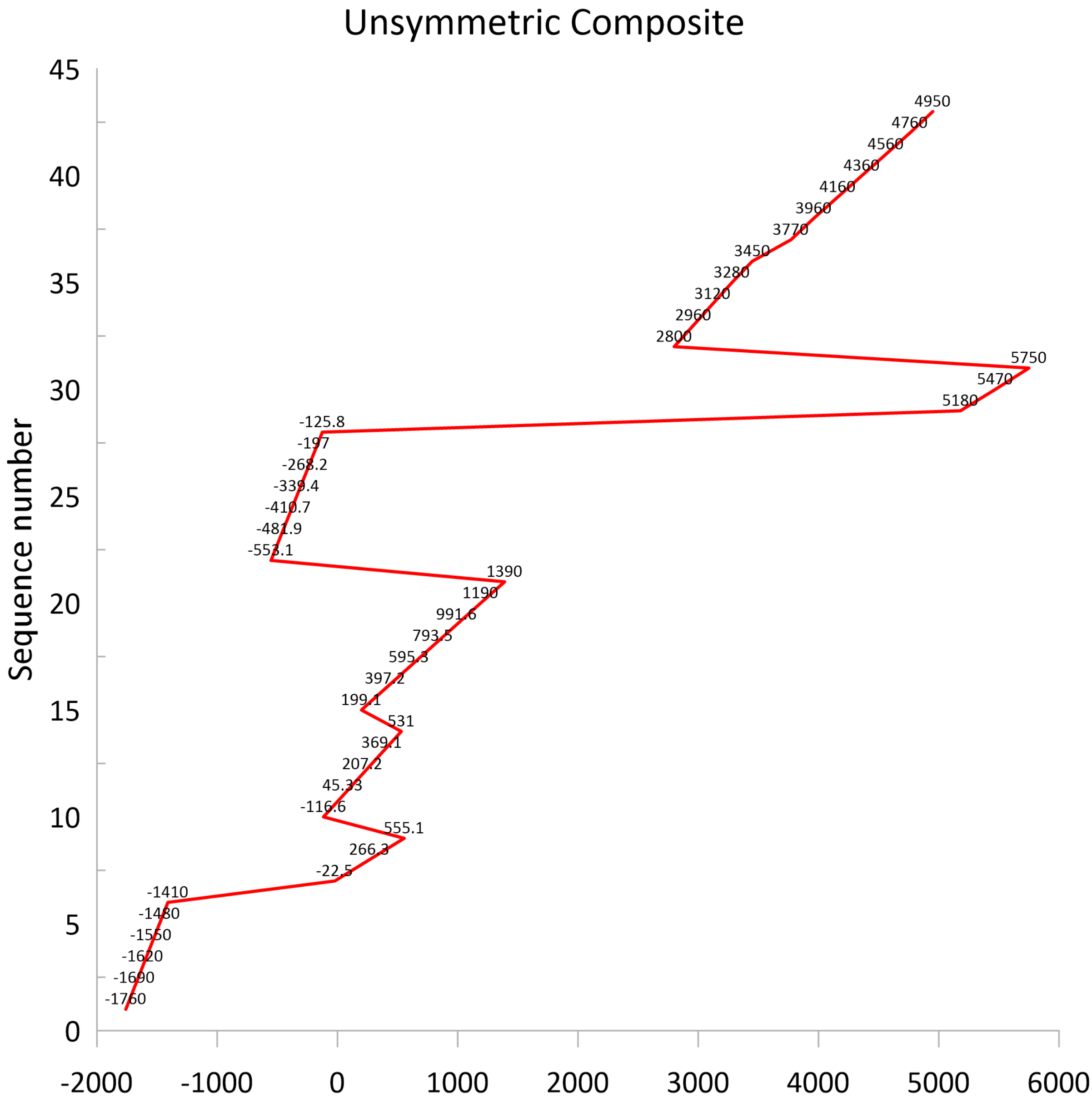


Figure 23: On-axis Stresses in 1-direction from abaqus results for unsymmetric composite

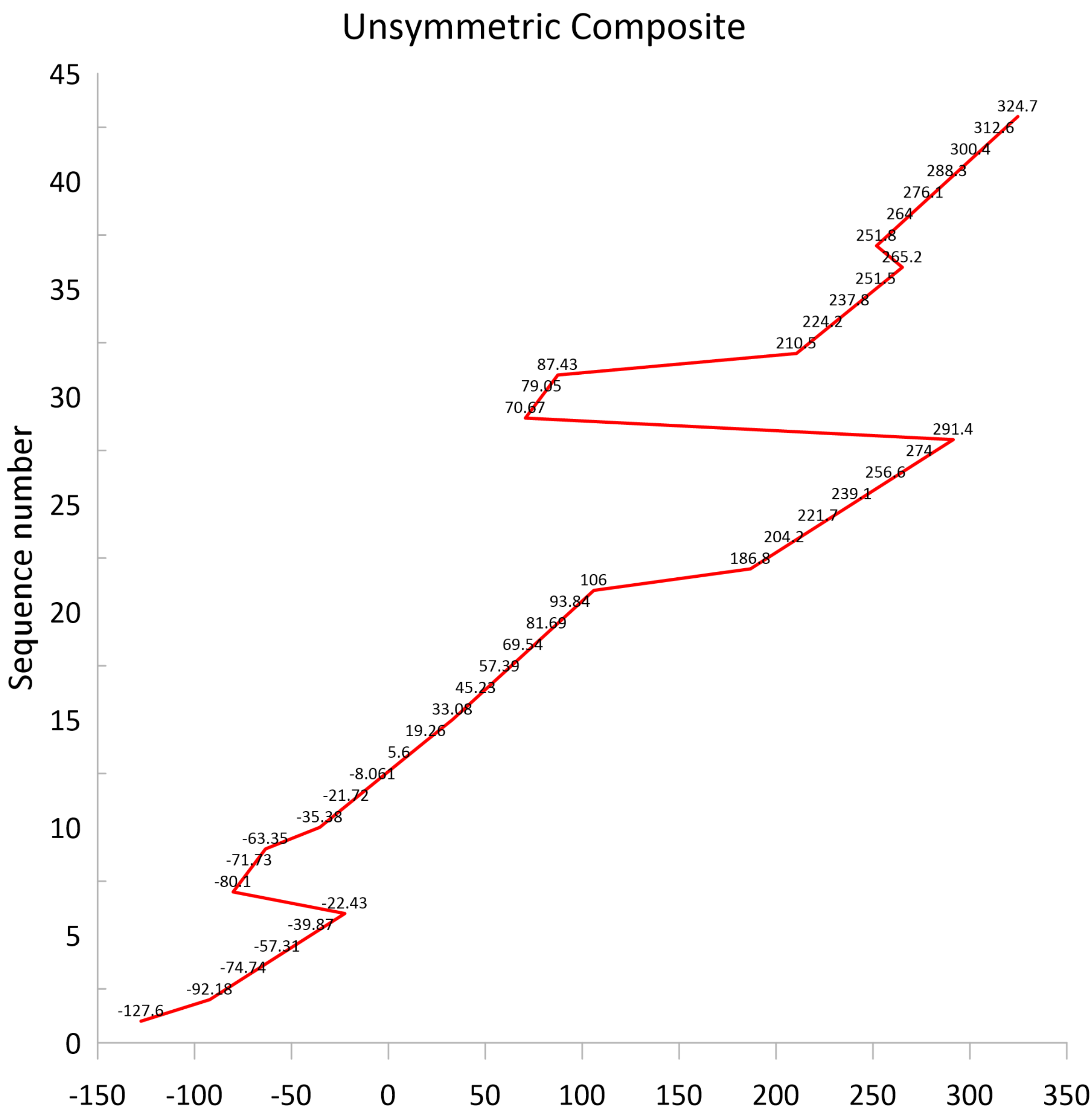


Figure 24: On-axis Stresses in 2-direction from abaqus results for unsymmetric composite

As It can be seen from the figures, CLT results are so close to the abaqus results.

5 Question 5

6. **Use a finite element program and calculate the on-axis strains and stresses of each layer of the following simply supported T-beam (L = 2 m), made of T300/5208, under a concentrated load (500 N) at the mid-span.**

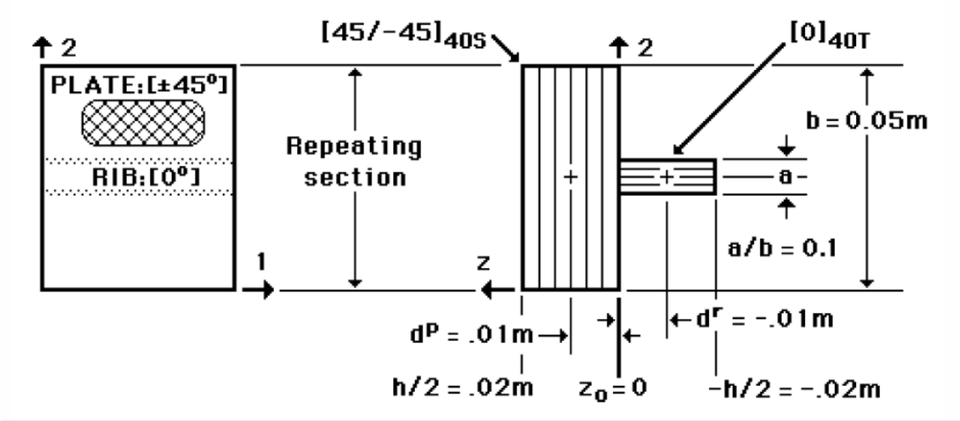
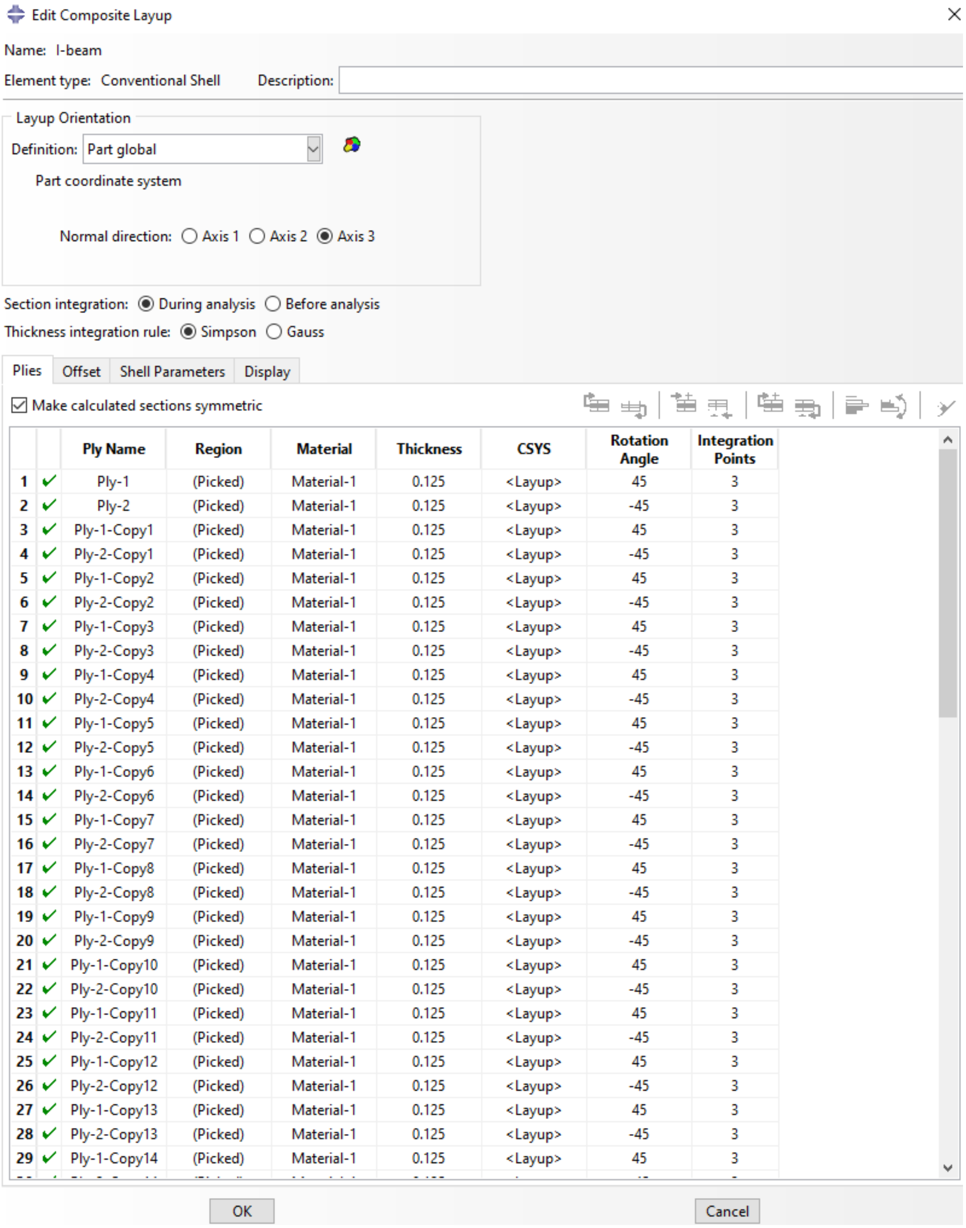


Figure 25: Schematic of question 5

Most part of the analysis is like the question before. The parts that are changed are mentioned in this section.
The dimensions of square has changed to 50 mm*50 mm and the layup has changed to the picture below. The sequence of 45/-45 has been repeated 40 times and then we made the composite symmetric.



		Ply Name	Region	Material	Thickness	CSYS	Rotation Angle	Integration Points
1	✔	Ply-1	(Picked)	Material-1	0.125	<Layup>	45	3
2	✔	Ply-2	(Picked)	Material-1	0.125	<Layup>	-45	3
3	✔	Ply-1-Copy1	(Picked)	Material-1	0.125	<Layup>	45	3
4	✔	Ply-2-Copy1	(Picked)	Material-1	0.125	<Layup>	-45	3
5	✔	Ply-1-Copy2	(Picked)	Material-1	0.125	<Layup>	45	3
6	✔	Ply-2-Copy2	(Picked)	Material-1	0.125	<Layup>	-45	3
7	✔	Ply-1-Copy3	(Picked)	Material-1	0.125	<Layup>	45	3
8	✔	Ply-2-Copy3	(Picked)	Material-1	0.125	<Layup>	-45	3
9	✔	Ply-1-Copy4	(Picked)	Material-1	0.125	<Layup>	45	3
10	✔	Ply-2-Copy4	(Picked)	Material-1	0.125	<Layup>	-45	3
11	✔	Ply-1-Copy5	(Picked)	Material-1	0.125	<Layup>	45	3
12	✔	Ply-2-Copy5	(Picked)	Material-1	0.125	<Layup>	-45	3
13	✔	Ply-1-Copy6	(Picked)	Material-1	0.125	<Layup>	45	3
14	✔	Ply-2-Copy6	(Picked)	Material-1	0.125	<Layup>	-45	3
15	✔	Ply-1-Copy7	(Picked)	Material-1	0.125	<Layup>	45	3
16	✔	Ply-2-Copy7	(Picked)	Material-1	0.125	<Layup>	-45	3
17	✔	Ply-1-Copy8	(Picked)	Material-1	0.125	<Layup>	45	3
18	✔	Ply-2-Copy8	(Picked)	Material-1	0.125	<Layup>	-45	3
19	✔	Ply-1-Copy9	(Picked)	Material-1	0.125	<Layup>	45	3
20	✔	Ply-2-Copy9	(Picked)	Material-1	0.125	<Layup>	-45	3
21	✔	Ply-1-Copy10	(Picked)	Material-1	0.125	<Layup>	45	3
22	✔	Ply-2-Copy10	(Picked)	Material-1	0.125	<Layup>	-45	3
23	✔	Ply-1-Copy11	(Picked)	Material-1	0.125	<Layup>	45	3
24	✔	Ply-2-Copy11	(Picked)	Material-1	0.125	<Layup>	-45	3
25	✔	Ply-1-Copy12	(Picked)	Material-1	0.125	<Layup>	45	3
26	✔	Ply-2-Copy12	(Picked)	Material-1	0.125	<Layup>	-45	3
27	✔	Ply-1-Copy13	(Picked)	Material-1	0.125	<Layup>	45	3
28	✔	Ply-2-Copy13	(Picked)	Material-1	0.125	<Layup>	-45	3
29	✔	Ply-1-Copy14	(Picked)	Material-1	0.125	<Layup>	45	3

Figure 26: Layup of question 5

In this part, we used a **trick** an defined offset ratio.In composite structures, especially in I-beams, the offset ratio can be used to define the location of the reference plane relative to the laminate's mid-plane. By adjusting the offset, you can effectively shift the neutral axis away from the composite sheet, influencing how the load is distributed.

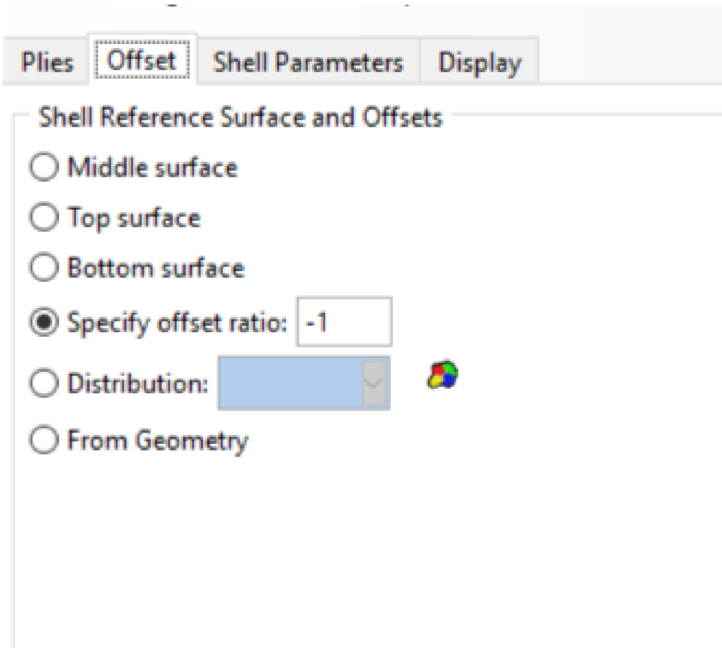


Figure 27: Offset trick

Afterwards, the loading changes to 500 N in 1-direction. The results extracted form .rpt file named I-beam. The S11 and S22 stress in the first ply areas shown in the figure below.

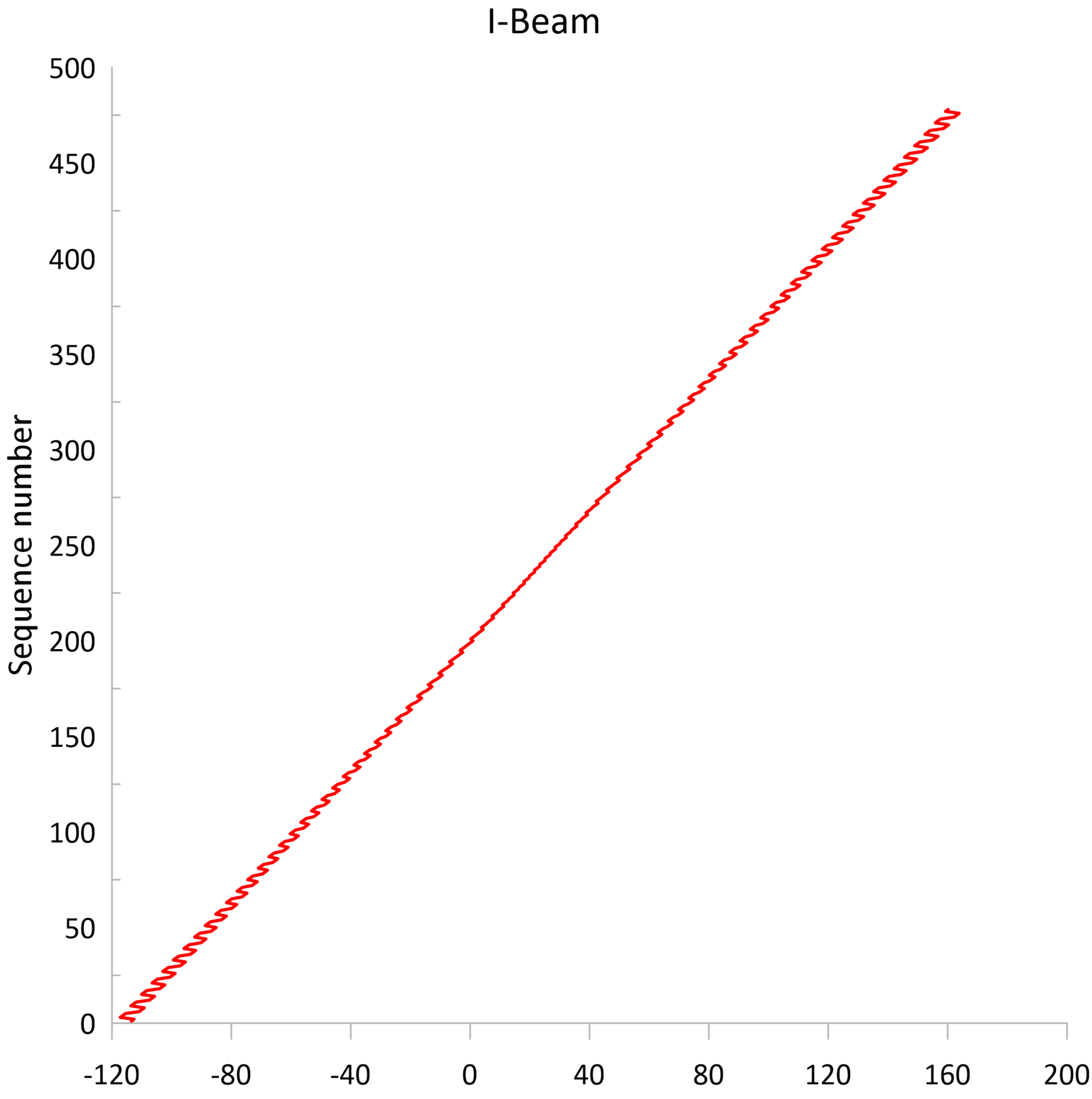


Figure 28: On-axis Stresses in 1-direction for I-beam

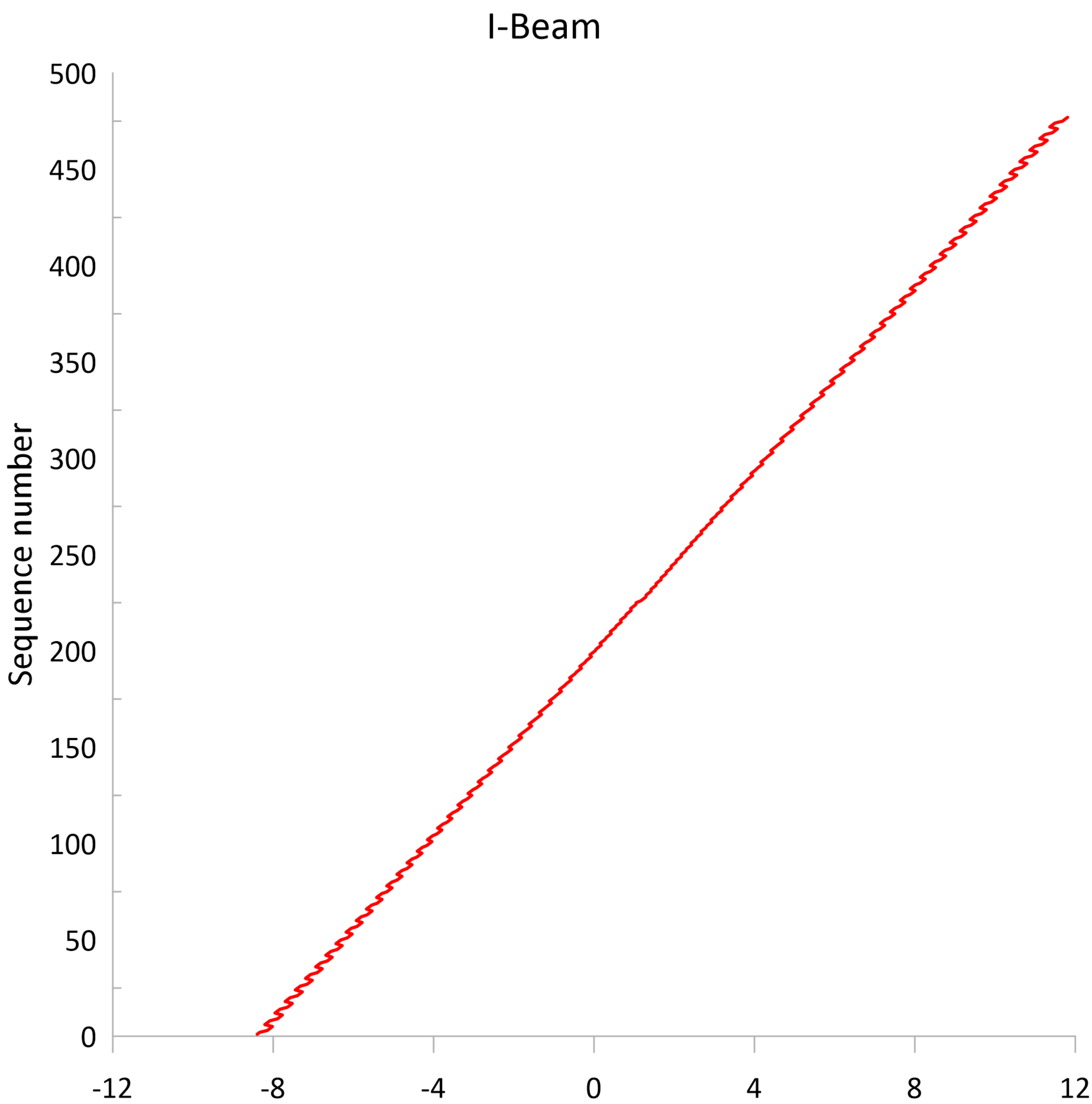


Figure 29: On-axis Stresses in 2-direction for I-beams

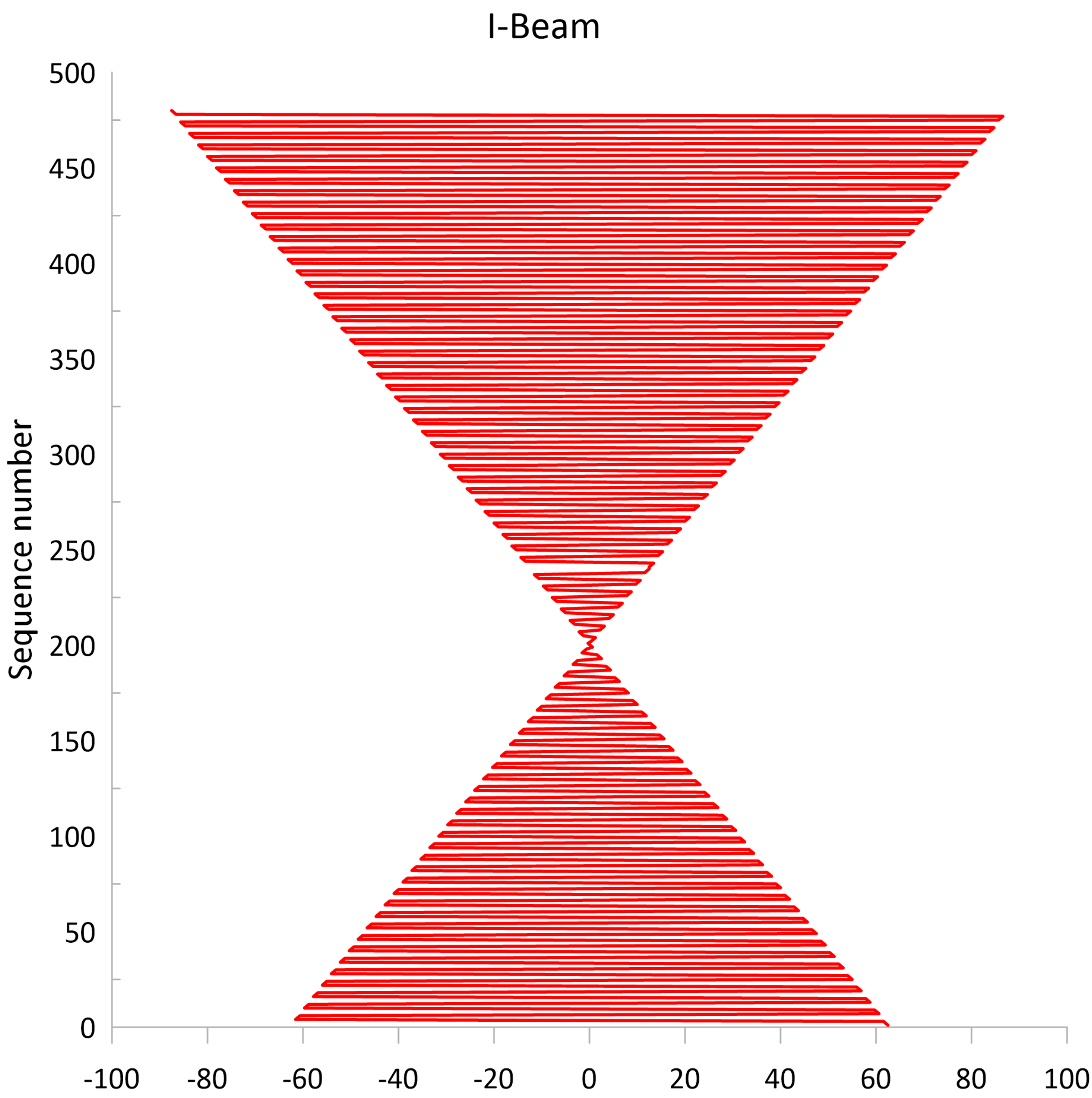


Figure 30: On-axis shear stresses for I-beams

This should be mentioned that we have 160 layers and 3 magnitudes for each layer (Top, middle and bottom). The sequence number 1 shows the end of the most bottom layer.

6 Question 6

6. **What is a balanced laminate? Give an example.** Balanced composites are fiber-reinforced laminate structures in which the fiber orientations are arranged such

that for every ply at an angle $+\theta°$, there is a corresponding ply at $-\theta°$. Also 90 and 0 degree orientations are allowed because negative of them is equal to themselves. This configuration ensures that the **in-plane mechanical properties** (such as stiffness and strength) are **orthotropic**, meaning they are direction-dependent but symmetric about two orthogonal axes. Balanced laminates provide consistent performance under in-plane loading and eliminate coupling between normal and shear deformations.

Examples:

$[0/+45_3/90/-45_3]_{2T}$

$[0/+45_3/90/-45_3]_s$

7 Question 7

6. **Is a balanced laminate also a symmetric laminate?** Balanced laminates are not necessarily symmetric. They can be asymmetric too, like the example below: $[0/+45_4/90/0/-45_4]_{2T}$