AM5450 Assignment 7

Q1)

Plots from MATLAB Code:

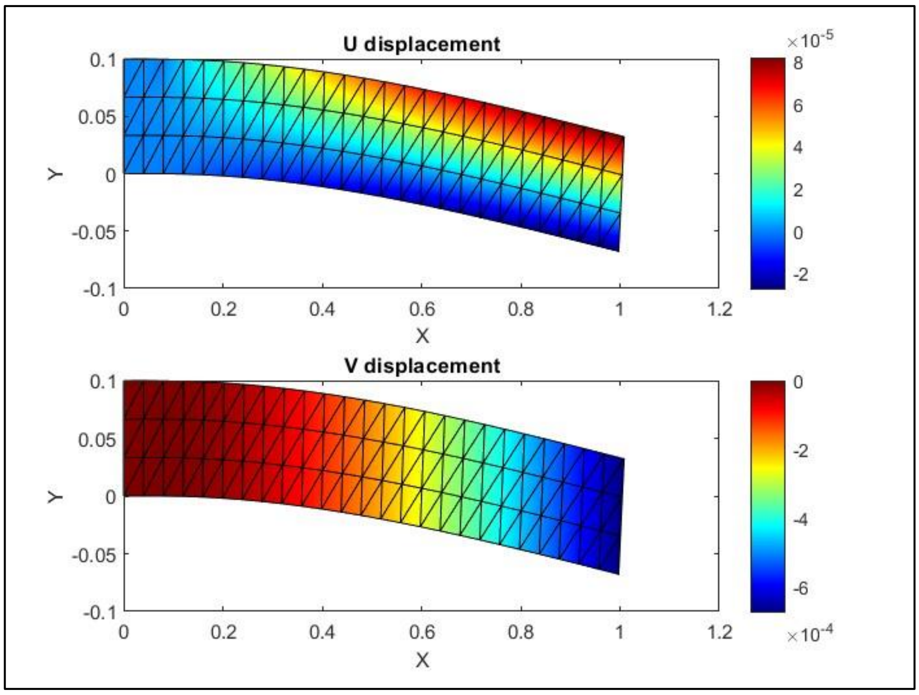


Fig 1: Contours of U and V displacements of the beam

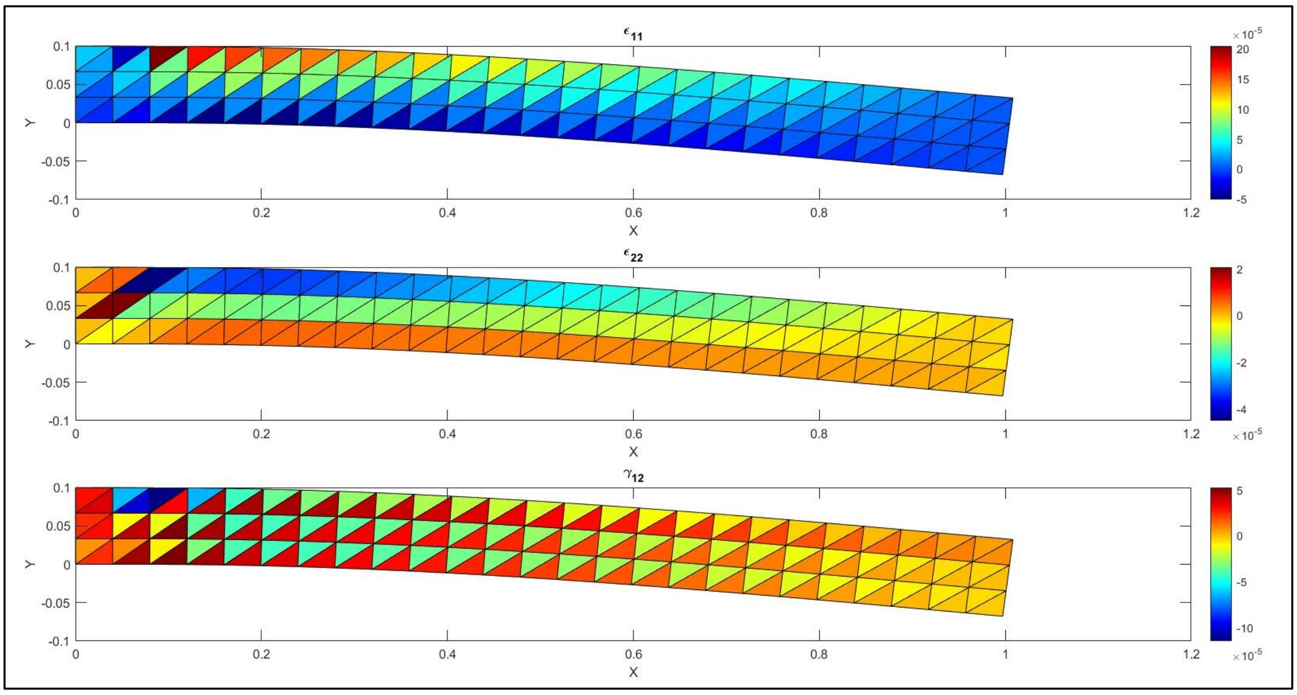


Fig 2: Strain components (ε_{11}, ε_{11}, Υ_{12}) for the deformed beam

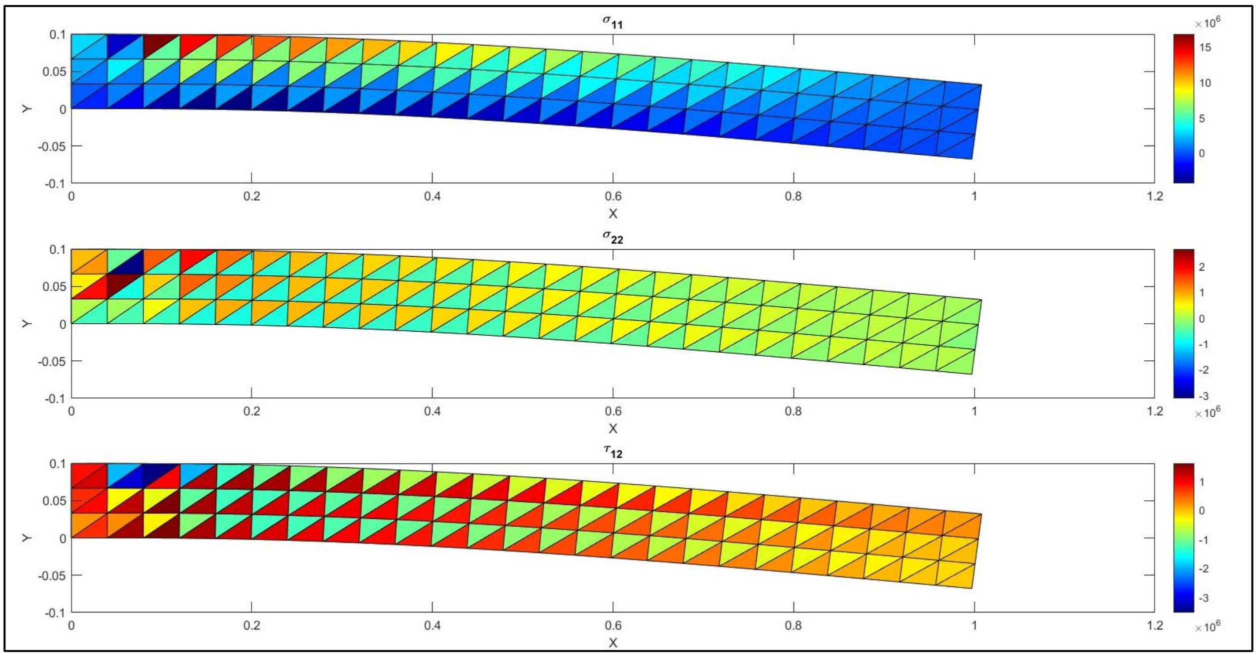


Fig 3: Stress components (σ_{11}, σ_{22}, τ_{12}) for the deformed beam

Plots from ABAQUS Simulation:

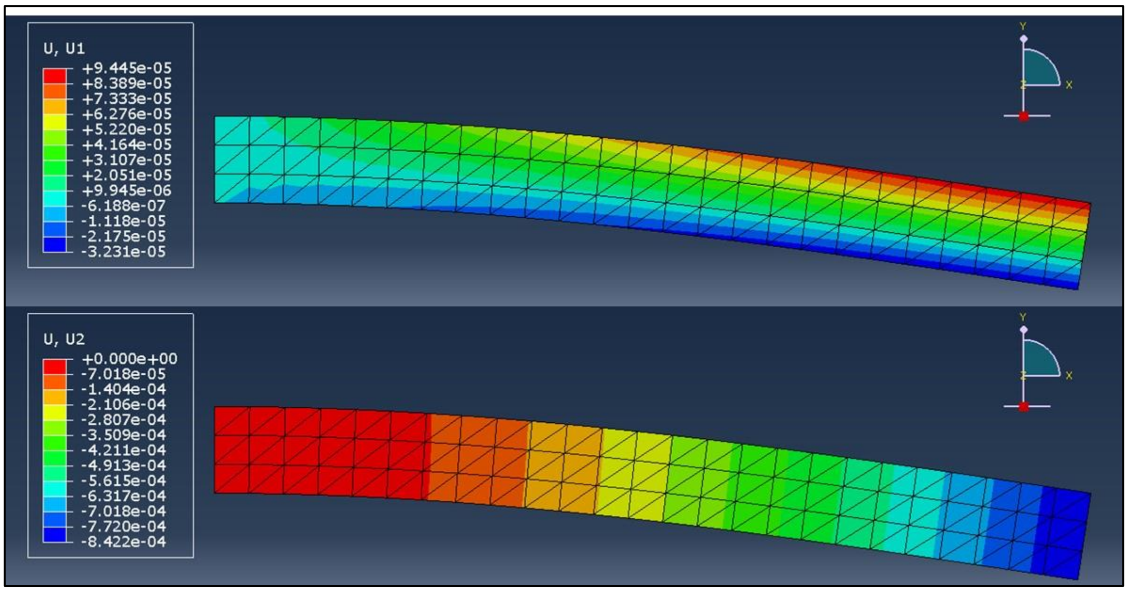


Fig 4: Contours of U and V displacements of the beam

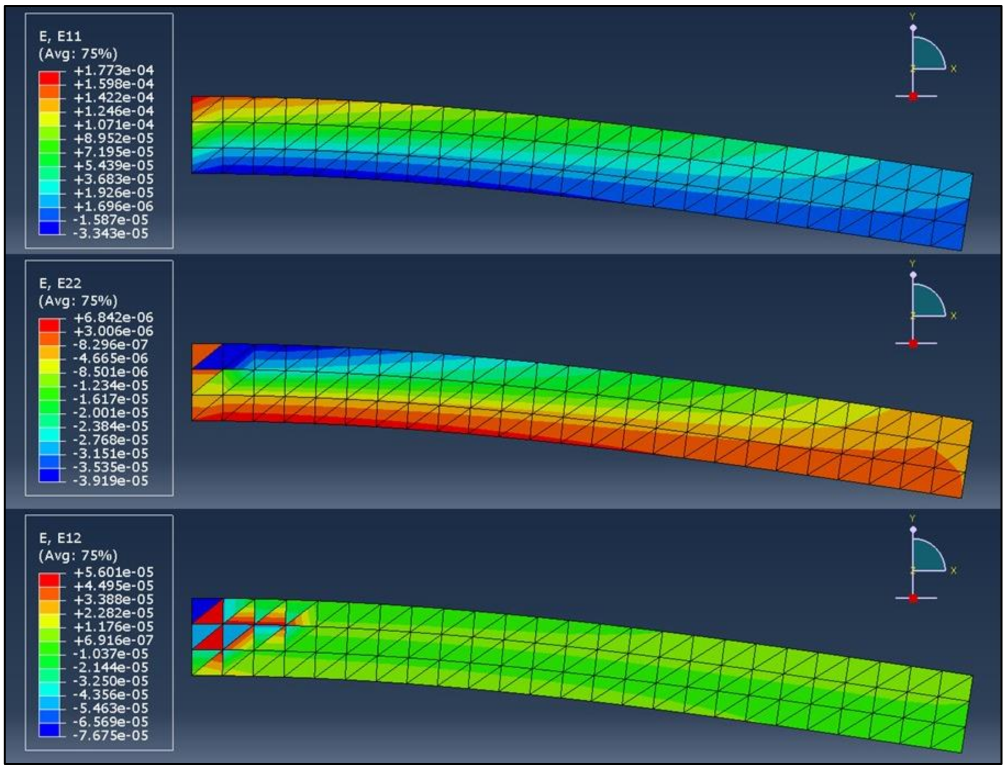


Fig 5: Strain components (E11, E22, E12) for the deformed beam

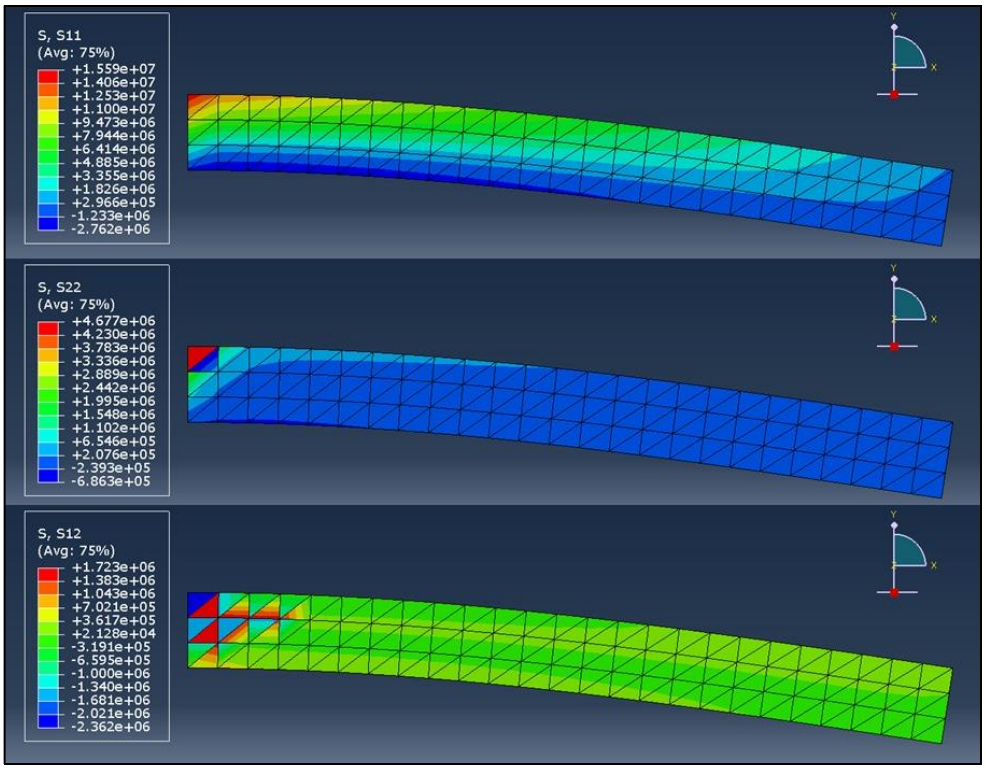


Fig 6: Stress components (S11, S22, S12) for the deformed beam

- The MATLAB code, excel sheet with mesh data from ABAQUS (Q1_Mesh) and tabulated comparison of displacement, strain and stress between the MATLAB code and ABAQUS (Q1_Comparison) are attached in the submission.
- The table below summarizes the percentage absolute error for all fields between the ABAQUS simulation and MATLAB Code which is calculated in the attached excel sheet.

Field	**Absolute Mean % Error**
U displacement	25.08
V displacement	39.33
S11	7.15
S22	14.2
S12	11.7
E11	7.37
E22	10.7
E12	11.7

Table 1: Percentage Error between MATLAB and ABAQUS results

Q2)

- The MATLAB code and mesh data from ABAQUS (Q2_Mesh) is attached to the submission.
- We consider only one fourth of the cylinder and apply symmetry boundary conditions at the ends. Following are the plots for the displacement, stress and strain in the cylinder obtained from the MATLAB code.

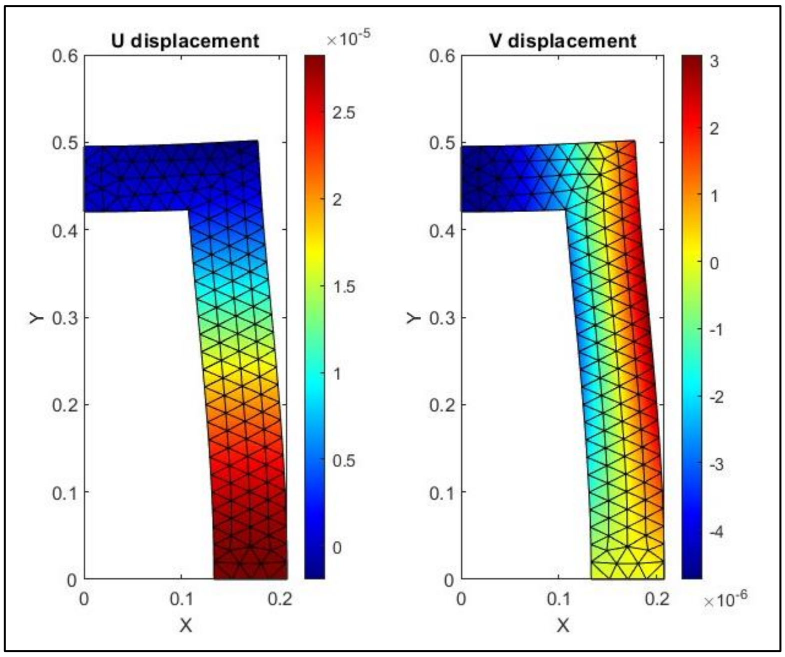


Fig 7: U and V displacements for the pressure vessel

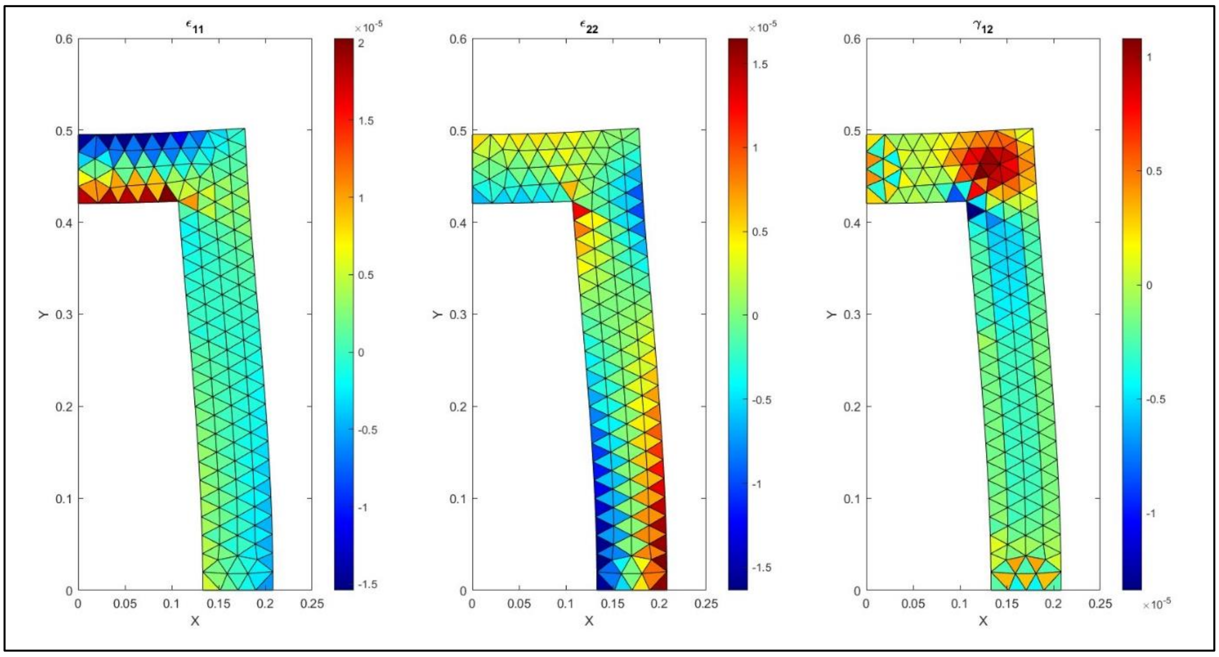


Fig 8: Strain components (ε_{11}, ε_{11}, Υ_{12}) for the deformed pressure vessel

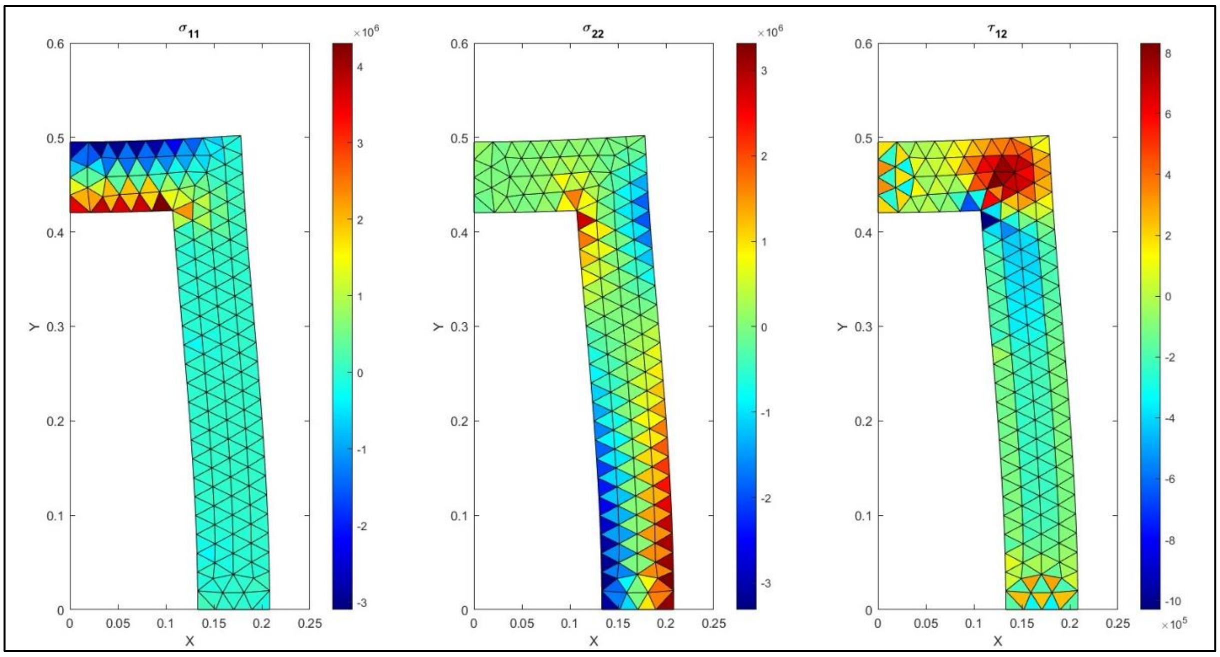
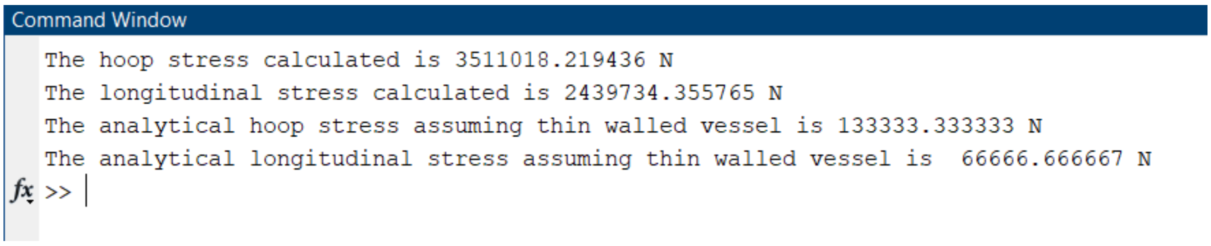


Fig 9: Stress components (σ_{11}, σ_{22}, τ_{12}) for the deformed pressure vessel

- As we are using constant strain triangular elements (CST), the strain over each element is constant.
- We calculate the hoop stress as the maximum absolute value of the σ_{11} stress component in the domain and similarly the longitudinal stress is calculated as maximum absolute value of the σ_{22}.
- Using the theory of thin-walled cylindrical pressure vessels, we can calculate the hoop and longitudinal stresses as Pt/2d and Pt/4d respectively. Following is the code output which shows a comparison of both



```
The hoop stress calculated is 3511018.219436 N
The longitudinal stress calculated is 2439734.355765 N
The analytical hoop stress assuming thin walled vessel is 133333.333333 N
The analytical longitudinal stress assuming thin walled vessel is  66666.666667 N
>> 
```

The values do not match well with each other. There are two possible reasons for this

- Firstly, the mesh is coarse and a better result can be obtained for a finer mesh.
- Secondly, the d/t ratio for this vessel (5.33) is less than 20, so it does not satisfy the criteria for a thin-walled vessel (d/t > 20). Hence, the equation for a thick-walled pressure vessel should be used.

Hence, if we use a finer mesh by increasing the number of elements and calculate the analytical stresses using the formulation for a thick cylindrical pressure vessel, more accurate results can be obtained.

Q3)

- 3 different levels of mesh density are used - 6 elements, 54 elements and 216 elements.
- The MATLAB Code and excel sheets (Q3_Mesh) containing the mesh data are attached to the submission.
- Following are the plots of temperature distribution as we increase the number of elements and the grid convergence plot considering the temperature at (3,3).

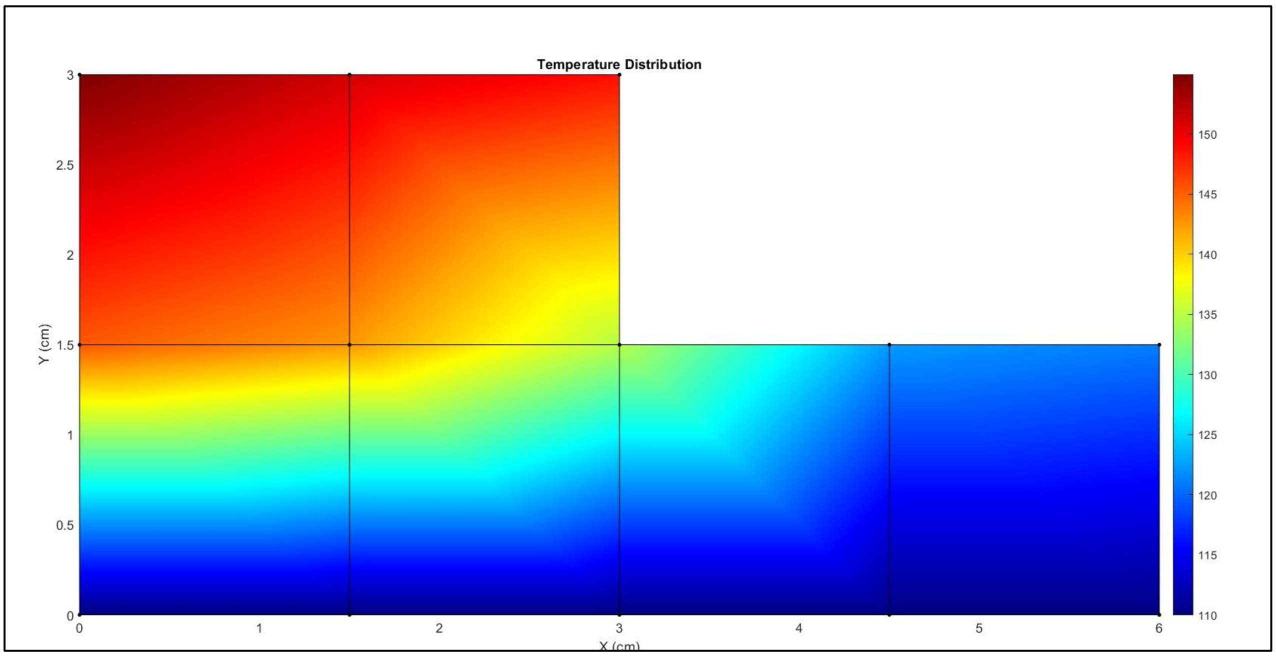


Fig 10: Temperature distribution in the domain with 6 elements

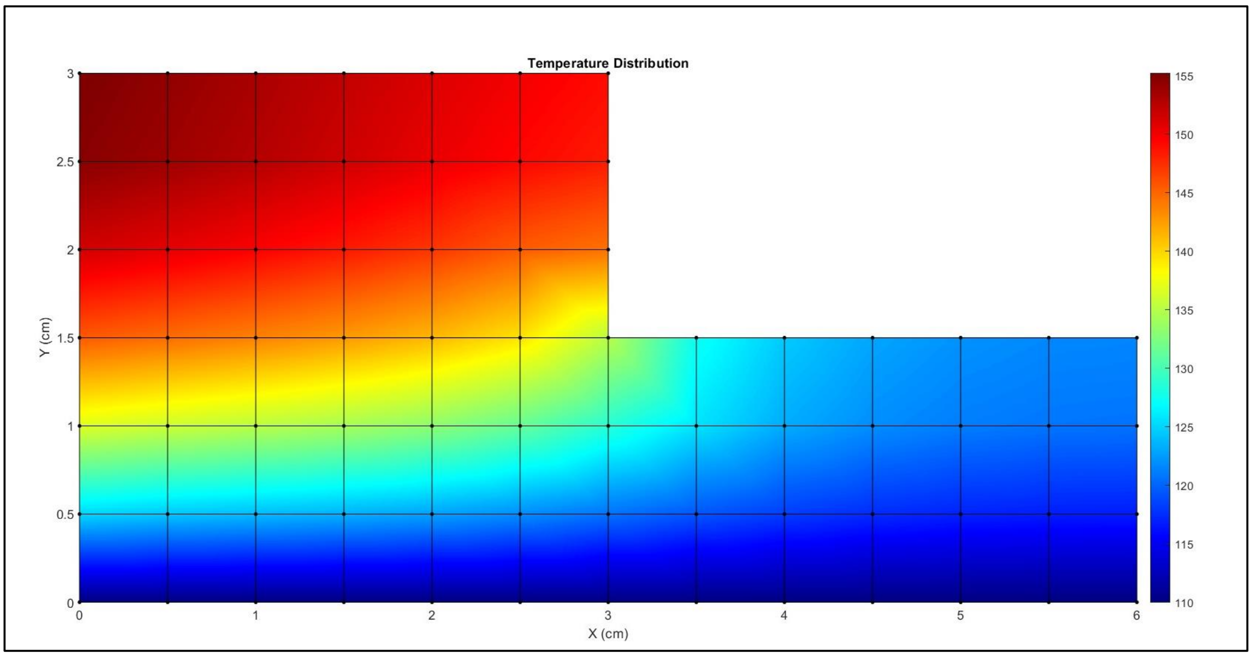


Fig 11: Temperature distribution in the domain with 54 elements

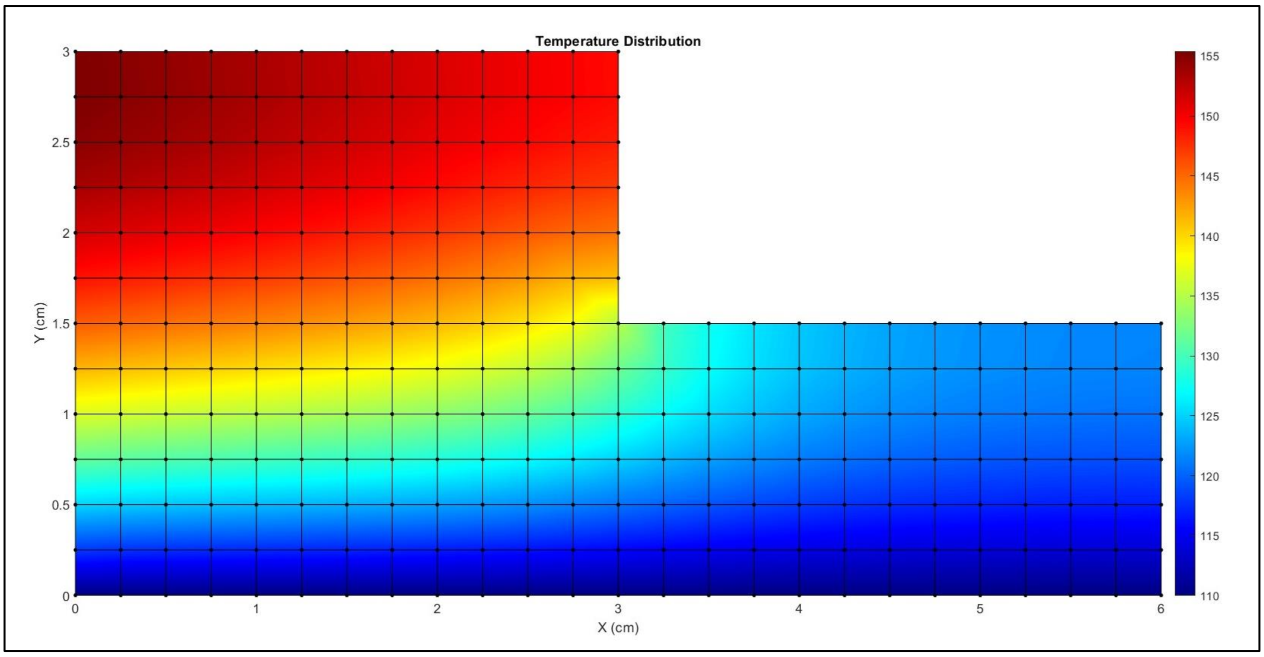


Fig 12: Temperature distribution in the domain with 216 elements

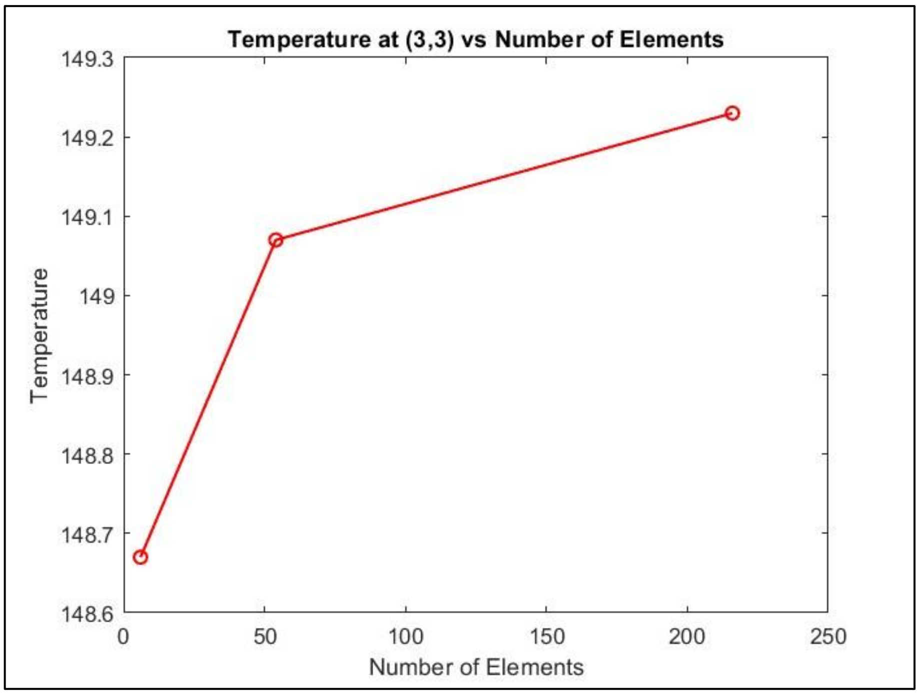


Fig 13: Temperature at (3,3) vs Number of elements

- From the above figure, we can see that as the number of elements is increased, the temperature at the fixed point (3,3) is converging to around 149.2 degrees.
- Using a few more mesh densities greater than 216 elements can show this mesh convergence in an even better way.

MATLAB Codes:

Q1)

```
clc; clear all; close all; format compact; format shortg;

%Constants
Fs = 500*10^3; %Surface Shear Force
Ft = 600; %Tip load
E = 80*10^9; %Young's Modulus
nu = 0.3; %Poisson Ratio
L = 1; w = 0.1; %Dimensions of domain

%Importing Mesh data
connectivity = readmatrix("Q1_Mesh.xlsx","Sheet",1);
coords = readmatrix("Q1_Mesh.xlsx","Sheet",2);
X = coords(:,2); Y = coords(:,3);

%Boundary nodes
EBC_nodes = [1, 27, 53, 79];
Ft_node = 26;
Fs_nodes = [79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95,
96, 97, 98, 99, 100, 101, 102, 103, 104];

N = size(connectivity,1); %Number of elements;
n = size(coords,1); %Number of nodes
A = L*w/N; %Area of each element

%Assembling global stiffness matrix
k_k_global = zeros(2*n,2*n);
for i = 1:N
    node1 = connectivity(i,2); node2 = connectivity(i,3); node3 =
connectivity(i,4);
    nodes = [node1,node2,node3];

    x = X(nodes);
    y = Y(nodes);

    k_k = stiffness_k(x,y,A,E,nu);
    for j = 1:numel(nodes)
        k_k_global([2*nodes(j),2*nodes(j)-1],2*nodes-1) =
k_k_global([2*nodes(j),2*nodes(j)-1],2*nodes-1) + k_k([2*j,2*j-1],[1,3,5]);
        k_k_global([2*nodes(j),2*nodes(j)-1],2*nodes) =
k_k_global([2*nodes(j),2*nodes(j)-1],2*nodes) + k_k([2*j,2*j-1],[2,4,6]);
    end
end

%Shear Force boundary
r_q_global = zeros(2*n,1);
for i = 1:numel(Fs_nodes)-1
    node1 = Fs_nodes(i); node2 = Fs_nodes(i+1);
    x1 = X(node1); x2 = X(node2);
    y1 = Y(node1); y2 = Y(node2);
    L = ((x1-x2)^2 + (y1-y2)^2)^0.5;

    r_q_global(2*node1-1) = r_q_global(2*node1-1) + (L/2)*Fs;
    r_q_global(2*node2-1) = r_q_global(2*node2-1) + (L/2)*Fs;
```

```
end

%Applying EBC by removing corresponding equations
k_r = k_k_global;
k_r(2*EBC_nodes,:) = [];
k_r(2*EBC_nodes-1,:) = [];
k_r(:,2*EBC_nodes) = [];
k_r(:,2*EBC_nodes-1) = [];

r_r = r_q_global;
%Adding point load to r vector
r_r(2*Ft_node) = r_r(2*Ft_node) + Ft;

r_r(2*EBC_nodes) = [];
r_r(2*EBC_nodes-1) = [];

%Solving for d vector
d_r = k_r\r_r;

d = zeros(2*n,1);
idx = 1:1:2*n;
idx([2*EBC_nodes,2*EBC_nodes-1]) = [];
d(idx) = d_r;

%For plotting we are multiplying the displacememtns by 100
%Otherwise the displacements are very small to be seen on the plot
X_new = X + 100*d(1:2:end);
Y_new = Y + 100*d(2:2:end);

%Plotting the displacement field
figure(1);
for i = 1:N
    node1 = connectivity(i,2); node2 = connectivity(i,3); node3 =
connectivity(i,4);
    nodes = [node1,node2,node3];

    x = X_new(nodes); y = Y_new(nodes);
    subplot(2,1,1), fill(x,y,d(2*nodes-1)); hold on;
    subplot(2,1,2), fill(x,y,d(2*nodes)); hold on;
end
subplot(2,1,1), xlabel("X");  ylabel("Y"); title("U displacement");
colormap("jet"); colorbar
subplot(2,1,2), xlabel("X");  ylabel("Y"); title("V displacement");
colormap("jet"); colorbar

%Strain Calculation
e11 = zeros(N,1); e22 = zeros(N,1); gamma12 = zeros(N,1);
for i = 1:N
    node1 = connectivity(i,2); node2 = connectivity(i,3); node3 =
connectivity(i,4);
    nodes = [node1,node2,node3];
    x = X(nodes);
    y = Y(nodes);

    dl = zeros(6,1);
    dl([1,3,5]) = d(2*nodes-1);
    dl([2,4,6]) = d(2*nodes);

    e = strain(x,y,dl,A);
    e11(i) = e(1); e22(i) = e(2); gamma12(i) = e(3);
end

%Plotting strain field
figure;
for i = 1:N
    node1 = connectivity(i,2); node2 = connectivity(i,3); node3 =
connectivity(i,4);
    nodes = [node1,node2,node3];

    x = X_new(nodes); y = Y_new(nodes);
    subplot(3,1,1), fill(x,y,e11(i)*ones(3,1)); hold on;
    subplot(3,1,2), fill(x,y,e22(i)*ones(3,1)); hold on;
    subplot(3,1,3), fill(x,y,gamma12(i)*ones(3,1)); hold on;
end
subplot(3,1,1), xlabel("X");  ylabel("Y"); title("\epsilon_{11}");
colormap("jet"); colorbar
subplot(3,1,2), xlabel("X");  ylabel("Y"); title("\epsilon_{22}");
colormap("jet"); colorbar
subplot(3,1,3), xlabel("X");  ylabel("Y"); title("\gamma_{12}"); colormap("jet");
colorbar

%Calculating and plotting stress field
sigma11 = zeros(N,1); sigma22 = zeros(N,1); tau12 = zeros(N,1);
figure;
for i = 1:N
    node1 = connectivity(i,2); node2 = connectivity(i,3); node3 =
connectivity(i,4);
    nodes = [node1,node2,node3];
    x = X_new(nodes); y = Y_new(nodes);

    C = (E/(1-nu^2))*[1, nu, 0;
                     nu, 1, 0;
                     0, 0 , (1-nu)/2];
    stress = C*[e11(i);e22(i);gamma12(i)];
    sigma11(i) = stress(1); sigma22(i) = stress(2); tau12(i) = stress(3);

    subplot(3,1,1), fill(x,y,sigma11(i)); hold on;
    subplot(3,1,2), fill(x,y,sigma22(i)); hold on;
    subplot(3,1,3), fill(x,y,tau12(i)); hold on;
end
subplot(3,1,1), xlabel("X");  ylabel("Y"); title("\sigma_{11}"); colormap("jet");
colorbar;
subplot(3,1,2), xlabel("X");  ylabel("Y"); title("\sigma_{22}"); colormap("jet");
colorbar
subplot(3,1,3), xlabel("X");  ylabel("Y"); title("\tau_{12}"); colormap("jet");
colorbar

%Function to calculate stiffness matrix k_k
function k_k = stiffness_k(x,y,A,E,nu)
    x1 = x(1); x2 = x(2); x3 = x(3);
    y1 = y(1); y2 = y(2); y3 = y(3);

    b1 = y2-y3; b2 = y3-y1; b3 = y1-y2;
    c1 = x3-x2; c2 = x1-x3; c3 = x2-x1;

    B = (1/(2*A))*[b1, 0, b2, 0, b3, 0;
                0, c1, 0, c2, 0, c3;
                c1, b1, c2, b2, c3, b3]';
    C = (E/(1-nu^2))*[1, nu, 0;
                      nu, 1, 0;
                      0, 0 , (1-nu)/2];

    k_k = A*(B*C*B');
end

function e = strain(x,y,d,A)
    x1 = x(1); x2 = x(2); x3 = x(3);
    y1 = y(1); y2 = y(2); y3 = y(3);

    b1 = y2-y3; b2 = y3-y1; b3 = y1-y2;
    c1 = x3-x2; c2 = x1-x3; c3 = x2-x1;

    B = (1/(2*A))*[b1, 0, b2, 0, b3, 0;
                0, c1, 0, c2, 0, c3;
                c1, b1, c2, b2, c3, b3]';

    e = B'*d;
end
```

Q2)

```
clc; clear all; close all; format compact; format shortg;

%Constants
P = 50*10^3; %Pressure in the vessel
E = 200*10^9; %Young's Modulus
nu = 0.3; %Poisson Ratio

%Importing Mesh data
connectivity = readmatrix("Q2_Mesh.xlsx","Sheet",1);
coords = readmatrix("Q2_Mesh.xlsx","Sheet",2);
X = coords(:,2); Y = coords(:,3);

%Boundary nodes
Xsym_nodes = [4, 36, 35, 34, 5];
Ysym_nodes = [1, 7, 8, 9, 2];
P_vertical_nodes = [1, 13, 12, 11, 10, 3];
P_horizontal_nodes = [4, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27,
28, 29, 30, 31, 32, 33, 3];

N = size(connectivity,1); %Number of elements;
n = size(coords,1); %Number of nodes

%Assembling global stiffness matrix
k_k_global = zeros(2*n,2*n);
for i = 1:N
    node1 = connectivity(i,2); node2 = connectivity(i,3); node3 =
connectivity(i,4);
    nodes = [node1,node2,node3];

    x = X(nodes);
    y = Y(nodes);

    k_k = stiffness_k(x,y,E,nu);
    for j = 1:numel(nodes)
```

```
        k_k_global([2*nodes(j),2*nodes(j)-1],2*nodes-1) =
k_k_global([2*nodes(j),2*nodes(j)-1],2*nodes-1) + k_k([2*j,2*j-1],[1,3,5]);
        k_k_global([2*nodes(j),2*nodes(j)-1],2*nodes) =
k_k_global([2*nodes(j),2*nodes(j)-1],2*nodes) + k_k([2*j,2*j-1],[2,4,6]);
    end
end

%Horizontal Pressure Boundary
r_p_global = zeros(2*n,1);
for i = 1:numel(P_horizontal_nodes)-1
    node1 = P_horizontal_nodes(i); node2 = P_horizontal_nodes(i+1);
    x1 = X(node1); x2 = X(node2);
    y1 = Y(node1); y2 = Y(node2);
    L = ((x1-x2)^2 + (y1-y2)^2)^0.5;

    r_p_global(2*node1-1) = r_p_global(2*node1-1) + (L/2)*P;
    r_p_global(2*node2-1) = r_p_global(2*node2-1) + (L/2)*P;
end

%Vertical Pressure Boundary
for i = 1:numel(P_vertical_nodes)-1
    node1 = P_vertical_nodes(i); node2 = P_vertical_nodes(i+1);
    x1 = X(node1); x2 = X(node2);
    y1 = Y(node1); y2 = Y(node2);
    L = ((x1-x2)^2 + (y1-y2)^2)^0.5;

    r_p_global(2*node1) = r_p_global(2*node1) - (L/2);
    r_p_global(2*node2) = r_p_global(2*node2) - (L/2);
end

%Applying symmetry conditions by removing corresponding equations
k_r = k_k_global;
k_r([2*Ysym_nodes-1,2*Xsym_nodes],:) = [];
k_r(:,[2*Ysym_nodes-1,2*Xsym_nodes]) = [];

r_r = r_p_global;
r_r([2*Ysym_nodes-1,2*Xsym_nodes]) = [];

%Solving for d vector
d_r = k_r\r_r;

d = zeros(2*n,1);
idx = 1:1:2*n;
idx([2*Ysym_nodes-1,2*Xsym_nodes]) = [];
d(idx) = d_r;

%For plotting we are multiplying the displacememtns by 1000
%Otherwise the displacements are very small to be seen on the plot
X_new = X + 1000*d(1:2:end);
Y_new = Y + 1000*d(2:2:end);

%Plotting the displacement field
figure(1);
for i = 1:N
    node1 = connectivity(i,2); node2 = connectivity(i,3); node3 =
connectivity(i,4);
    nodes = [node1,node2,node3];

    x = X_new(nodes); y = Y_new(nodes);
    subplot(1,2,1), fill(x,y,d(2*nodes-1)); hold on;
    subplot(1,2,2), fill(x,y,d(2*nodes)); hold on;

end
subplot(1,2,1), xlabel("X");  ylabel("Y"); title("U displacement");
colormap("jet"); colorbar;
subplot(1,2,2), xlabel("X");  ylabel("Y"); title("V displacement");
colormap("jet"); colorbar;

%Strain Calculation
e11 = zeros(N,1); e22 = zeros(N,1); gamma12 = zeros(N,1);
for i = 1:N
    node1 = connectivity(i,2); node2 = connectivity(i,3); node3 =
connectivity(i,4);
    nodes = [node1,node2,node3];
    x = X(nodes);
    y = Y(nodes);

    dl = zeros(6,1);
    dl([1,3,5]) = d(2*nodes-1);
    dl([2,4,6]) = d(2*nodes);

    e = strain(x,y,dl);
    e11(i) = e(1); e22(i) = e(2); gamma12(i) = e(3);
end

%Plotting strain field
figure;
for i = 1:N
    node1 = connectivity(i,2); node2 = connectivity(i,3); node3 =
connectivity(i,4);
    nodes = [node1,node2,node3];

    x = X_new(nodes); y = Y_new(nodes);
    subplot(1,3,1), fill(x,y,e11(i)); hold on;
    subplot(1,3,2), fill(x,y,e22(i)); hold on;
    subplot(1,3,3), fill(x,y,gamma12(i)); hold on;
end
subplot(1,3,1), xlabel("X");  ylabel("Y"); title("\epsilon_{11}");
colormap("jet"); colorbar
subplot(1,3,2), xlabel("X");  ylabel("Y"); title("\epsilon_{22}");
colormap("jet"); colorbar
subplot(1,3,3), xlabel("X");  ylabel("Y"); title("\gamma_{12}"); colormap("jet");
colorbar

%Calculating and plotting stress field
sigma11 = zeros(N,1); sigma22 = zeros(N,1); tau12 = zeros(N,1);
figure;
for i = 1:N
    node1 = connectivity(i,2); node2 = connectivity(i,3); node3 =
connectivity(i,4);
    nodes = [node1,node2,node3];
    x = X_new(nodes); y = Y_new(nodes);

    C = (E/(1-nu^2))*[1, nu, 0;
                      nu, 1, 0;
                      0, 0 , (1-nu)/2];
    stress = C*[e11(i);e22(i);gamma12(i)];
    sigma11(i) = stress(1); sigma22(i) = stress(2); tau12(i) = stress(3);
    subplot(1,3,1), fill(x,y,sigma11(i)); hold on;
    subplot(1,3,2), fill(x,y,sigma22(i)); hold on;
    subplot(1,3,3), fill(x,y,tau12(i)); hold on;
end
subplot(1,3,1), xlabel("X");  ylabel("Y"); title("\sigma_{11}"); colormap("jet");
colorbar
subplot(1,3,2), xlabel("X");  ylabel("Y"); title("\sigma_{22}"); colormap("jet");
colorbar
subplot(1,3,3), xlabel("X");  ylabel("Y"); title("\tau_{12}"); colormap("jet");
colorbar

hoop = max(abs(sigma11));
long = max(abs(sigma22));

d = 0.4; t = 0.075;
hoop_thin = P*d/(2*t);
long_thin = P*d/(4*t);

fprintf("The hoop stress calculated is %f N\n",hoop)
fprintf("The longitudinal stress calculated is %f N\n", long);
fprintf("The analytical hoop stress assuming thin walled vessel is %f
N\n",hoop_thin)
fprintf("The analytical longitudinal stress assuming thin walled vessel is  %f
N\n", long_thin);

%Function to calculate stiffness matrix k_k
function k_k = stiffness_k(x,y,E,nu)
    x1 = x(1); x2 = x(2); x3 = x(3);
    y1 = y(1); y2 = y(2); y3 = y(3);

    f1 = x2*y3-x3*y2; f2 = x3*y1-x1*y3; f3 = x1*y2-x2*y1;
    A = 0.5*(f1+f2+f3);

    b1 = y2-y3; b2 = y3-y1; b3 = y1-y2;
    c1 = x3-x2; c2 = x1-x3; c3 = x2-x1;

    B = (1/(2*A))*[b1, 0, b2, 0, b3, 0;
                0, c1, 0, c2, 0, c3;
                c1, b1, c2, b2, c3, b3]';

    C = (E/(1-nu^2))*[1, nu, 0;
                     nu, 1, 0;
                     0, 0 , (1-nu)/2];

    k_k = A*(B*C*B');
end

function e = strain(x,y,d)
    x1 = x(1); x2 = x(2); x3 = x(3);
    y1 = y(1); y2 = y(2); y3 = y(3);

    f1 = x2*y3-x3*y2; f2 = x3*y1-x1*y3; f3 = x1*y2-x2*y1;
    A = 0.5*(f1+f2+f3);

    b1 = y2-y3; b2 = y3-y1; b3 = y1-y2;
    c1 = x3-x2; c2 = x1-x3; c3 = x2-x1;

    B = (1/(2*A))*[b1, 0, b2, 0, b3, 0;
```

```
                0, c1, 0, c2, 0, c3;
                c1, b1, c2, b2, c3, b3]';

    e = B'*d;
end
```

Q3)

```
clc; clear all; close all; format compact; format shortg;

%Constants
% L = 0.015; %For 6 elements
% L = 0.005; %For 54 elements
L = 0.0025; %For 216 elements

a = L/2; b = L/2; %a and b for the square element
k = 45; %Thermal Conductivity
q0 = 8000; %Heat flux of left boundary
Q = 5*10^6; %Constant heat generation in the domain
h = 55; %Convective heat transfer coefficient
T0 = 110; %Essential BC at bottom boundary
Tamb = 20; %Ambient temperature
alpha1 = -h; %alpha for  convective NBC
beta1 = h*Tamb; %beta for convective NBC
beta2 = q0; %beta for constant heat flux NBC

%Importing Mesh data from excel
% connectivity = readmatrix("Q3_Mesh1.xlsx","Sheet",1); %For 6 elements
% connectivity = readmatrix("Q3_Mesh2.xlsx"); %For 54 elements
connectivity = readmatrix("Q3_Mesh3.xlsx"); %For 216 elements

% coords = readmatrix("Q3_Mesh1.xlsx","Sheet",2);
% coords = readmatrix("Q3_Mesh2.xlsx","Sheet",2);
coords = readmatrix("Q3_Mesh3.xlsx","Sheet",2);
X = coords(:,2); Y = coords(:,3);

elements = connectivity(:,1);
N = numel(elements); %Number of elements
n = size(coords,1); %Number of nodes

%Boundary nodes
% % Using 6 elements
% left_NBC_nodes = [7, 11, 6]; %Constant heat flux q0
% right_NBC_nodes1 = [3, 4];  %Insulated
% top_NBC_nodes1 = [6, 10, 5]; %Convective
% top_NBC_nodes2 = [1, 9, 4]; %Convective
% right_NBC_nodes2 = [1, 5]; %Convective
% EBC_nodes = [7, 12, 2, 8, 3]; %Fixed temperature

% % Using 54 elements
% left_NBC_nodes = [7, 33, 32, 31, 30, 29, 6]; % Constant heat flux q0
% right_NBC_nodes1 = [3, 15, 16, 4];           % Insulated
% top_NBC_nodes1 = [6, 28, 27, 26, 25, 24, 5]; % Convective
% top_NBC_nodes2 = [1, 21, 20, 19, 18, 17, 4]; % Convective
% right_NBC_nodes2 = [1, 22, 23, 5];           % Convective
% EBC_nodes = [7, 34, 35, 36, 37, 38, 2, 10, 11, 12, 13, 14, 3]; % Fixed
temperature
```

```
%Using 216 elements
left_NBC_nodes = [7, 66, 65, 64, 63, 62, 61, 60, 59, 58, 57, 56, 6];  % Constant
heat flux q0
right_NBC_nodes1 = [3, 24, 25, 26, 27, 28, 4];                        % Insulated
top_NBC_nodes1 = [6, 55, 54, 53, 52, 51, 50, 49, 48, 47, 46, 45, 5];  % Convective
top_NBC_nodes2 = [1, 39, 38, 37, 36, 35, 34, 33, 32, 31, 30, 29, 4];  % Convective
right_NBC_nodes2 = [1, 40, 41, 42, 43, 44, 5];                        % Convective
EBC_nodes = [7, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 2, 13, 14, 15, 16, 17,
18, 19, 20, 21, 22, 23, 3]; % Fixed temperature

%Assembling global k_k matrix and r_q vector
k_k_global = zeros(n,n);
r_q_global = zeros(n,1);

for i = 1:N
    node1 = connectivity(i,2); node2 = connectivity(i,3); node3 =
connectivity(i,4); node4 = connectivity(i,5);
    nodes = [node1,node2,node3,node4];
    k_k = stiffness_k(k,k,a,b);

    for j = 1:4
    k_k_global(nodes(j),nodes) = k_k_global(nodes(j),nodes) + k_k(j,:);
    r_q_global(nodes(j)) =  r_q_global(nodes(j)) + a*b*Q;
    end
end

%Assembling k_alpha matrix and r_beta vector for NBCs
k_alpha_global = zeros(n,n);
r_beta_global = zeros(n,1);

%NBC on left boundary
%This method will work only when the boundary nodes are given sequentially,
%in the same order as they lie in the domain
for i = 1:numel(left_NBC_nodes)-1
    node1 = left_NBC_nodes(i);
    node2 = left_NBC_nodes(i+1);

    r_beta = [b*beta2; b*beta2];
    r_beta_global([node1,node2]) = r_beta_global([node1,node2]) + r_beta;
end

%NBC on top boundary part 1
for i = 1:numel(top_NBC_nodes1)-1
    node1 = top_NBC_nodes1(i);
    node2 = top_NBC_nodes1(i+1);

    k_alpha = [-2*b*alpha1/3, -b*alpha1/3;
               -b*alpha1/3, -2*b*alpha1/3];
    k_alpha_global([node1,node2],[node1,node2]) =
k_alpha_global([node1,node2],[node1,node2]) + k_alpha;

    r_beta = [a*beta1; a*beta1];
    r_beta_global([node1,node2]) = r_beta_global([node1,node2]) + r_beta;
end

%NBC for top boundary part 2
for i = 1:numel(top_NBC_nodes2)-1
    node1 = top_NBC_nodes2(i);
    node2 = top_NBC_nodes2(i+1);
    k_alpha = [-2*b*alpha1/3, -b*alpha1/3;
               -b*alpha1/3, -2*b*alpha1/3];

    k_alpha_global([node1,node2],[node1,node2]) =
k_alpha_global([node1,node2],[node1,node2]) + k_alpha;

    r_beta = [a*beta1; a*beta1];
    r_beta_global([node1,node2]) = r_beta_global([node1,node2]) + r_beta;
end

%NBC for convective right side boundary
for i = 1:numel(right_NBC_nodes2)-1
    node1 = right_NBC_nodes2(i);
    node2 = right_NBC_nodes2(i+1);

    k_alpha = [-2*b*alpha1/3, -b*alpha1/3;
                -b*alpha1/3, -2*b*alpha1/3];

    k_alpha_global([node1,node2],[node1,node2]) =
k_alpha_global([node1,node2],[node1,node2]) + k_alpha;

    r_beta = [b*beta1; b*beta1];
    r_beta_global([node1,node2]) = r_beta_global([node1,node2]) + r_beta;
end

%Adding all the terms
k_global = k_k_global + k_alpha_global;
r_global = r_q_global + r_beta_global;

k_r = k_global;
r_r = r_global;

%Removing equations (rows) corresponding to EBCs
k_r(EBC_nodes,:) = [];
%Removing the rows corresponding to the EBCs
r_r(EBC_nodes) = [];
%Adjusting the RHS using the EBCs
r_r = r_r - k_r(:,EBC_nodes)*T0*ones(numel(EBC_nodes),1);

%Removing the columns corresponding to the EBCs
k_r(:,EBC_nodes) = [];

%Solving for the temperature field
T = zeros(n,1);
T_r = k_r\r_r;

%Combining the calculated T values and known T values from EBCs
idx = 1:1:n;
idx(EBC_nodes) = [];
T(idx) = T_r;
T(EBC_nodes) = T0;

%Plotting Temperature distributions
figure; hold on;
for i = 1:N
    %Using Fill to color each element with the temperature
    node1 = connectivity(i,2); node2 = connectivity(i,3); node3 =
connectivity(i,4); node4 = connectivity(i,5);
```

```
    nodes = [node1,node2,node3,node4];
    fill(X(nodes),Y(nodes),T(nodes))
end
colormap("jet")
colorbar
scatter(X,Y,10,"filled",'ok')
xlabel("X (cm)"); ylabel("Y (cm)");
title("Temperature Distribution")
axis equal;
xlim([0,6]); ylim([0,3]);

%Plotting T at (3,3) with different number of elements
T_vec = [148.67,149.07,149.23];
N_vec = [6,54,216];

%Function to calculate k_k
function k_k = stiffness_k(kx,ky,a,b)
    k_k = [ky*a/(3*b) + kx*b/(3*a),   ky*a/(6*b) - kx*b/(3*a),    -ky*a/(6*b) -
kx*b/(6*a),    kx*b/(6*a) - ky*a/(3*b);
          ky*a/(6*b) - kx*b/(3*a),    ky*a/(3*b) + kx*b/(3*a),    kx*b/(6*a) -
ky*a/(3*b),    -ky*a/(6*b) - kx*b/(6*a);
         -ky*a/(6*b) - kx*b/(6*a),    kx*b/(6*a) - ky*a/(3*b),    ky*a/(3*b) +
kx*b/(3*a),    ky*a/(6*b) - kx*b/(3*a);
          kx*b/(6*a) - ky*a/(3*b),    -ky*a/(6*b) - kx*b/(6*a),    ky*a/(6*b) -
kx*b/(3*a),    ky*a/(3*b) + kx*b/(3*a)];
end
```