

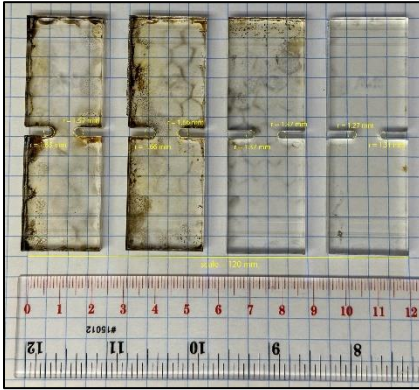


**MAE 503**  
**FINITE ELEMENTS IN ENGINEERING**

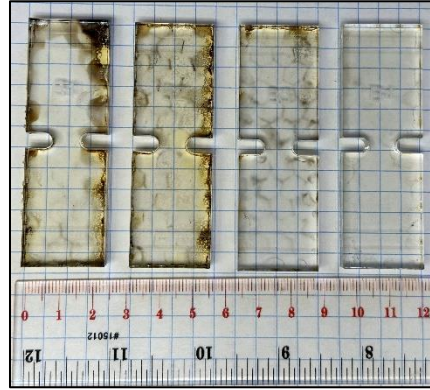
**Professor- Dr. Jay Oswald.**

Omkar Rajendra Gaikwad | ogaikwa1 | 1226843997  
Harshal Rajesh Tingre | htingre | 1226039050

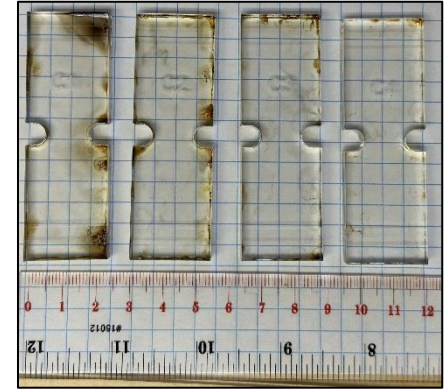
## Restating problem-



Case A (Test Specimen 1)



Case B (Test Specimen 2)



Case C (Test Specimen 3)

## Problem statement:

We have four parts for each case, and we can estimate their dimensions using the scale provided in the photos. We will compare our results with empirical formulas, ABAQUS solutions, and our own MATLAB FEM code for three different geometric configurations (Cases A, B, and C). We will also use Richardson extrapolation to estimate the rate of convergence of the maximum stress and the "exact" value of the maximum stress. We assume that the length of the plate is 75 mm (3x the width), and the nominal thickness is 1/4 inch (6.35 mm). The material properties we will use are Young's modulus of 3310 MPa, Poisson's ratio of 0.35-0.40, and tensile strength of 66 MPa.

## Methodology:

1. We Reviewed the problem statement and ensured that we understood the project's requirements, including the geometry of the notches, the material properties, and the desired outcomes.
2. We Define the problem in terms of a mathematical model. That includes the governing equations that describe the behavior of the material and the boundary conditions that apply to the problem.
3. We Choose an appropriate finite element method (FEM) software package such as ABAQUS for modeling the problem.
4. We Generated a 2D mesh of the geometry based on the type of elements you plan to use. Assign material properties and boundary conditions to the model.
5. We used ABAQUS software to solve the problem numerically. This will produce displacement and stress fields for the model.
6. Then we developed a MATLAB code for solving the problem using FEM. We Chose the appropriate type of elements based on the mesh and apply appropriate boundary conditions. Generate a table of results by varying the mesh resolution to show the convergence of the maximum stress.
7. We Plot the displacement and stress fields obtained from both the ABAQUS and MATLAB solutions.
8. Compared the stress concentration factor predictions obtained from the published empirical formula, the MATLAB code, and the ABAQUS simulation for all three cases.
9. Used the measured breaking loads to explain if the material fails at the same state of stress in all three cases.
10. Wrote a detailed report that summarizes the methodology, covered the results and the conclusions drawn from the analysis, and all the relevant plots and tables.

11. Reviewed and verified our results to ensure that they are accurate and consistent with the problem requirements.

### Strong form of the governing equation and Weak form of the governing equations:

The strong form of the governing equation is given by the following equations:

$$(\sigma * \nabla_s^T) + \vec{b} = 0$$

... (1-Equilibrium eq<sup>n</sup>)

$$\nabla \cdot \sigma + f = 0$$

$$\underline{\sigma} * \vec{n} = \vec{t} \text{ on } \Gamma_t$$

$$\vec{u} = \bar{u} \text{ on } \Gamma_u$$

The weak form of the governing equation is obtained by multiplying both sides of the strong form by a test function and integrating over the domain  $\Omega$ .

$$\int_{\Omega} (\sigma * \nabla_s w) d\Omega = \int_{\Omega} (w_i b_i) d\Omega + \int_{\Gamma_t} (w_i t_i) d\Gamma$$

Where,

$\sigma$  = Stress Tensor.

$F$  = Body force per volume.

$(\nabla \cdot)$  = Divergence Operator.

$\tau$  = tensor contraction.

$n$  = outward normal vector to the boundary  $\partial\Omega$

$d\Gamma$  = Integration over  $\partial\Omega$ .

To compute the finite element matrices needed for solving this problem, we need to discretize  $\Omega$  into a set of elements with nodes at their vertices.

The global stiffness matrix  $K$  and force vector  $F$  can then be assembled by summing the contributions from all elements.

For plane Stress Conditions:

$$\sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix}$$

$$\nabla_s = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}$$

### Boundary Conditions:

$\sigma_{xx} = 0$  ... (On the left and right side.)

$\sigma_{yy} = 0$  ... (On the bottom boundary.)

$\sigma_{yy} = U$  ... (On the top boundary. Where U = Applied load)

### Note-

The boundary conditions used in ABAQUS and MATLAB may differ, but they will both include prescribed displacements or forces on certain parts of the structure.

### Important Terms:

- Stiffness Matrice ( $K^e$ ) =

$$\int_{\Omega_e} [B^{eT}] * [D] * \underline{B}^e * d\Omega$$

- Force Matrice ( $f^e$ ) =

$$\int_{\Omega_e} N^e * \vec{b} * d\Omega$$

### Units-

We have used the following consistent units for specimen dimensions, stress, mass, force, and length while modeling the problem both in Abaqus and MATLAB.

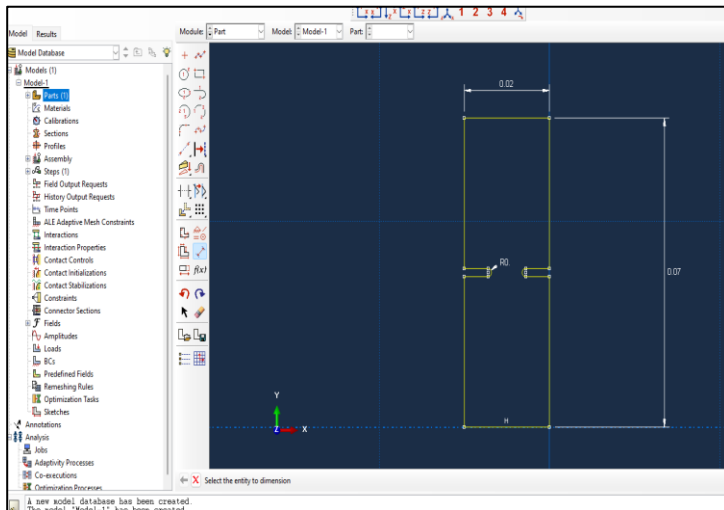
For the entire project, SI Units have been considered.

MASS	LENGTH/ RADIUS	TIME	FORCE	STRESS	ENERGY	Poisson's ratio of	YOUNG's (Mpa)
g	mm	ms	N	MPa	N-mm	0.35-0.40	3310

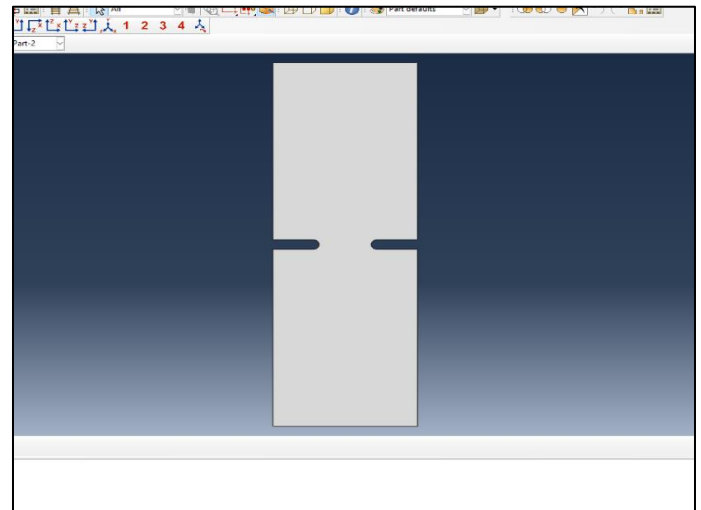
## MODELLING THE PROBLEM IN ABAQUS:

(We have maintained the same consistent units as above for modeling the problem in Abaqus.)

1. Open a new ABAQUS CAE model database and choose a standard or explicit model.
2. Draw a rectangle with dimensions of 37.5 mm length and 12.5 mm breadth.
3. Draw a circle with the center and perimeter points of (0,5.5), (0,6.5), and (0,9.5) for cases 1, 2, and 3, respectively.
4. Use the circle and rectangle to create an indent on the part.
5. Specify the mechanical properties of an elastic material, including Young's modulus and Poisson's ratio (3310 MPa and 0.375, respectively).
6. Create a solid, homogeneous section of the specified material with a plane stress or strain thickness of 6.35 mm and assign it to the entire part.
7. Seed the top and bottom portions of the rectangle towards the indent with sizes ranging from 0.76 to 3.8 mm.
8. Seed the right edge of the rectangle towards the indent with sizes ranging from 4 mm.
9. Seed the intact and indented parts of the left edge of the rectangle towards the indent with different sizes based on the case number.
10. Mesh the part using the mesh part option and create a new static, general step with nonlinear geometry off.
11. submit a new job for the model with a full analysis to obtain a solution.
12. After the analysis, plot the required data as graphs by selecting the desired plot contours for stress components at integration points and spatial displacement at nodes.
13. For Our project we took a section about Y-Y axis for better visualization.



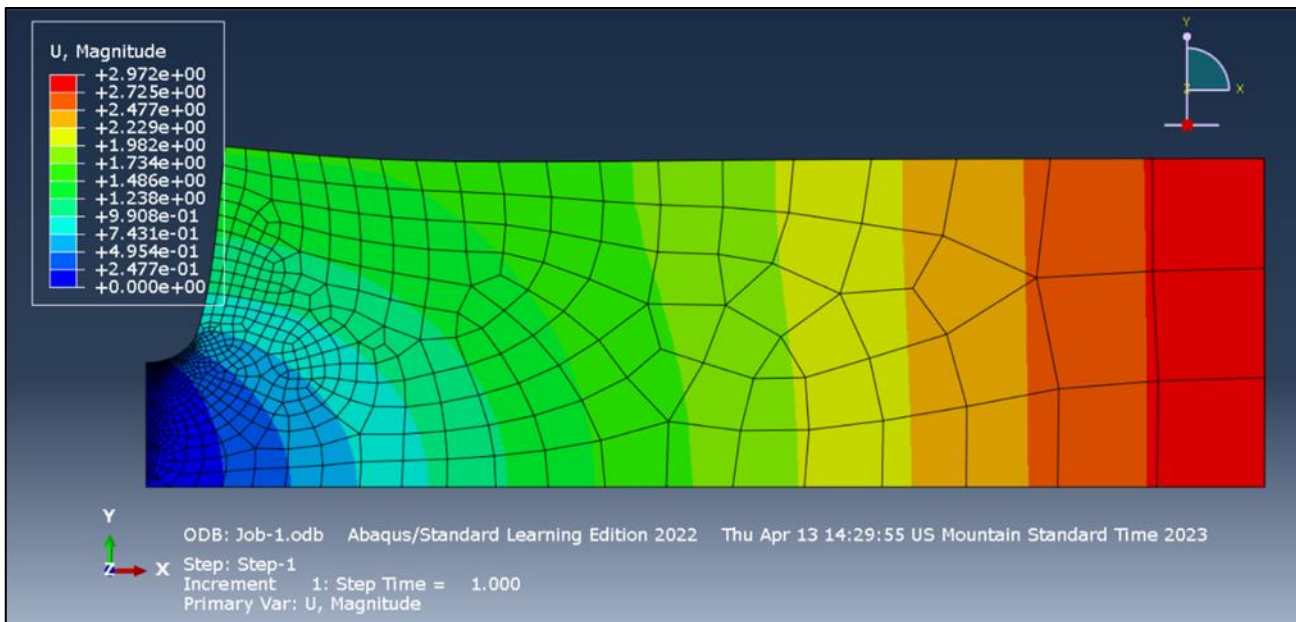
*Specimen with Actual Dimensions*



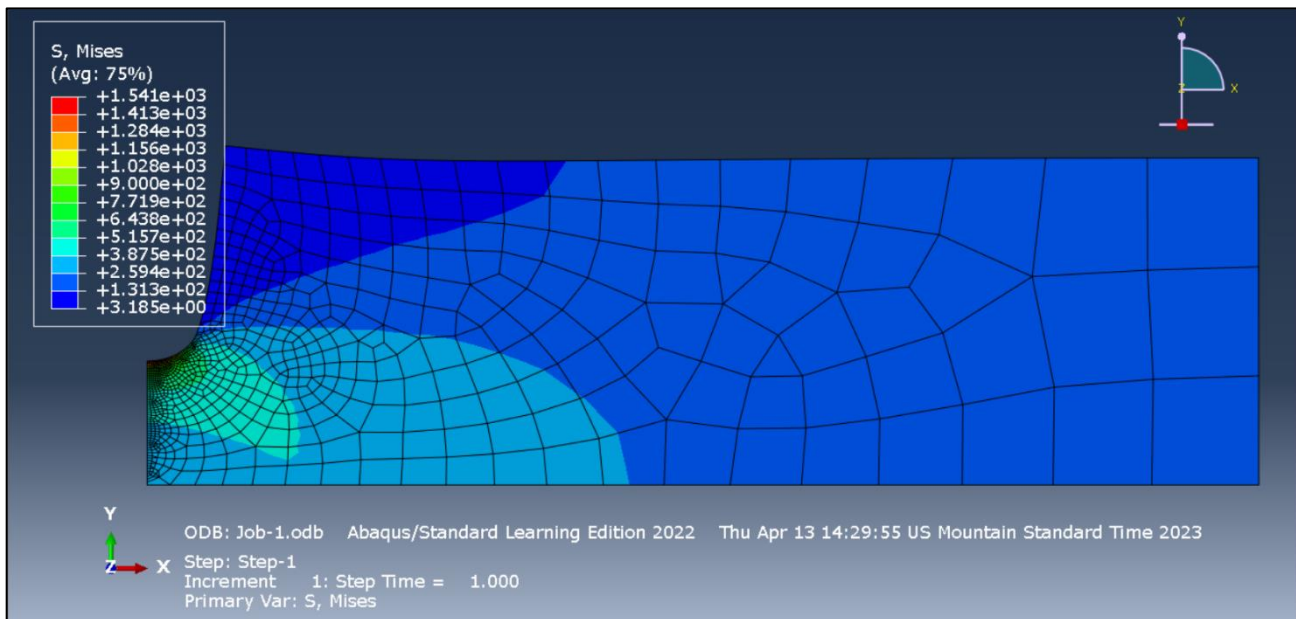
*CASE-A Specimen*

## Case-A (Section about Y-Y axis)

### Load distribution (Top-most position):



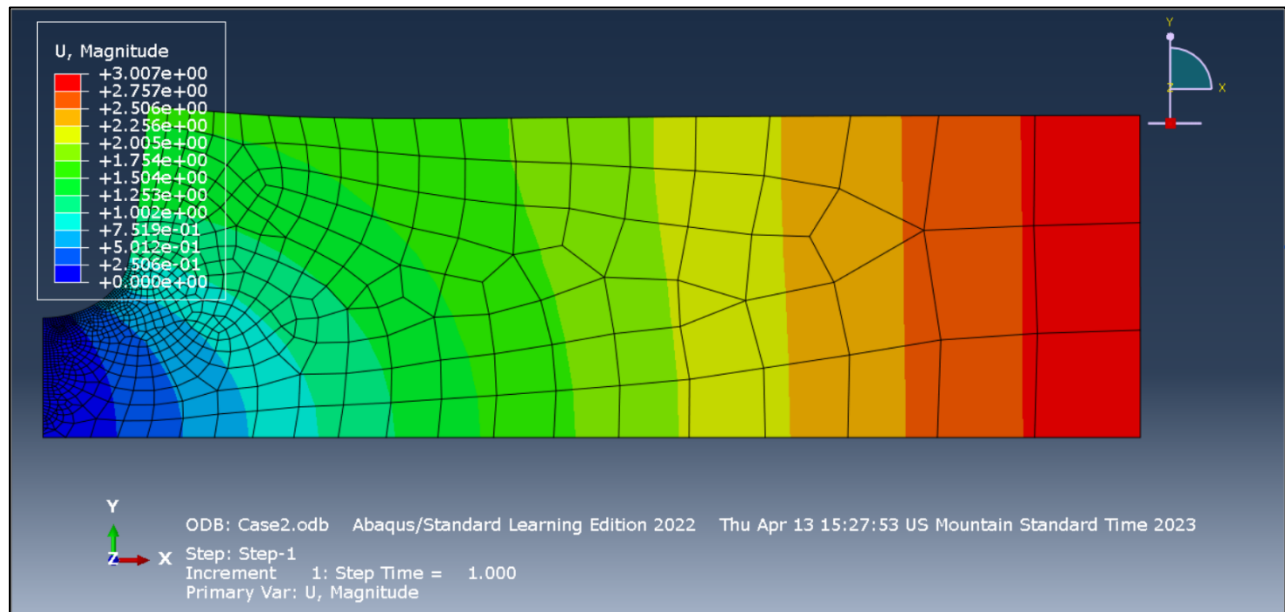
### Stress Field:



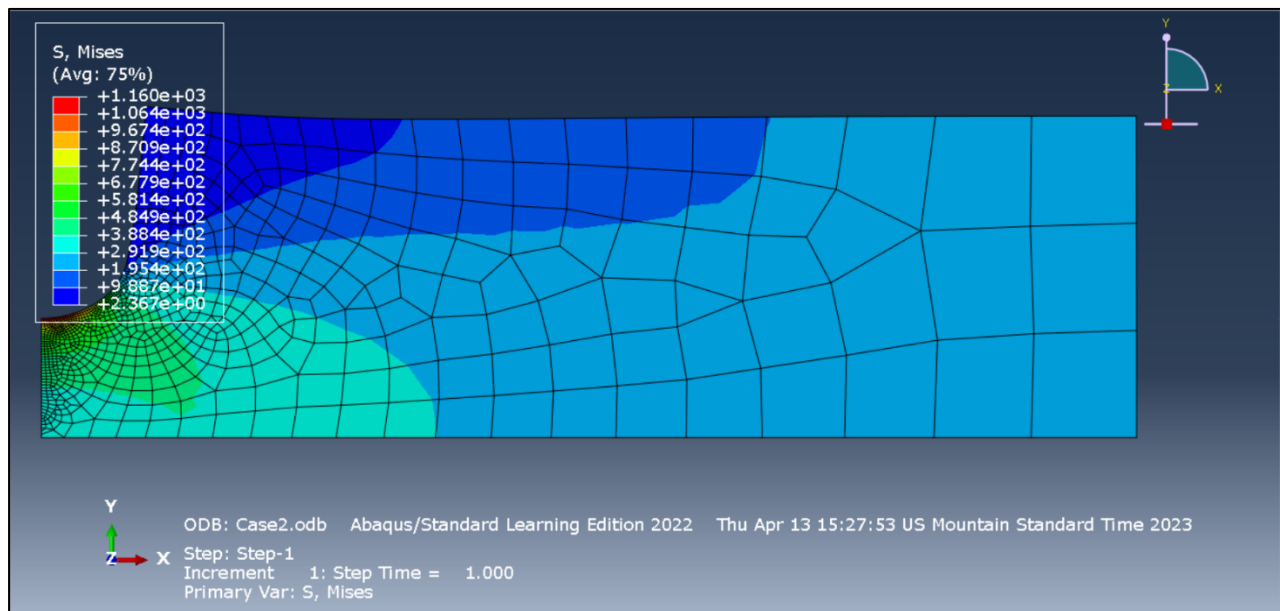


## Case-B (Section about Y-Y axis)

### Load distribution (Top-most position):

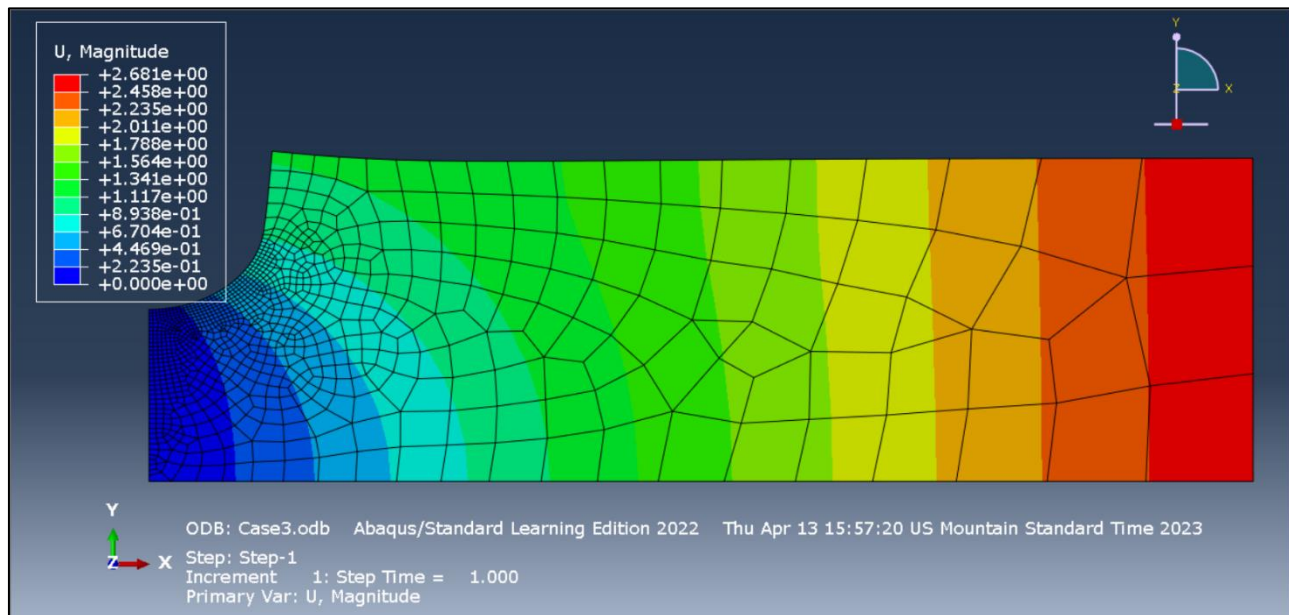


### Stress Field:

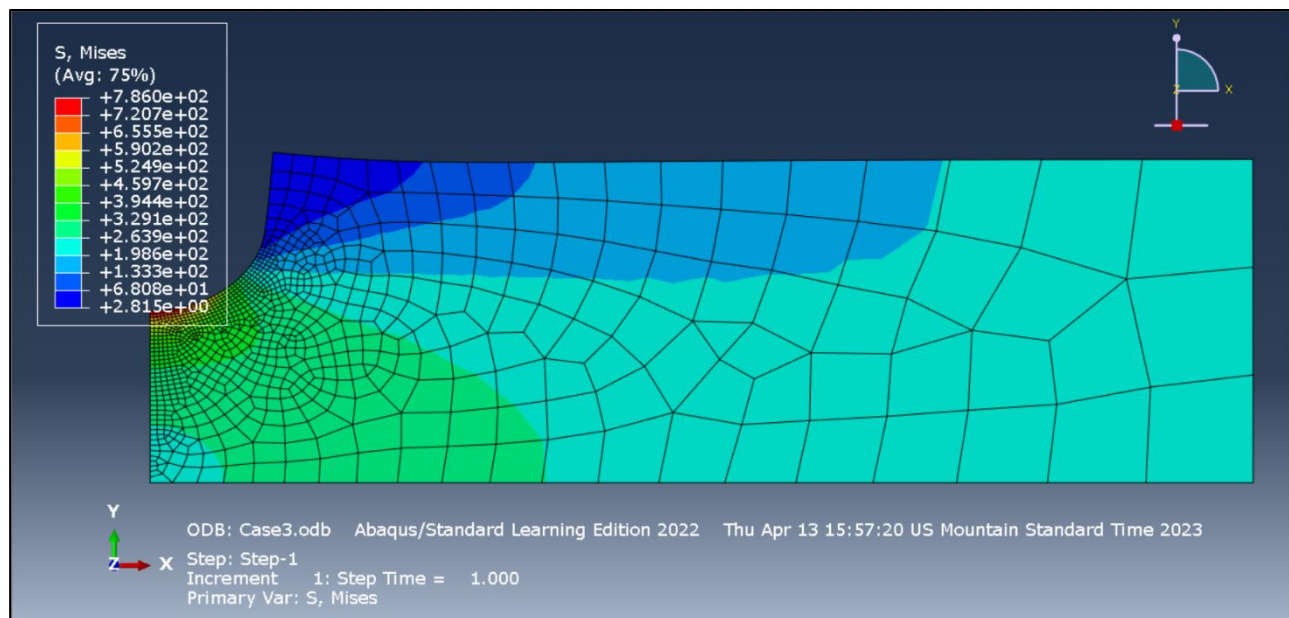


## Case-C (Section about Y-Y axis)

### Load distribution (Top-most position):



### Stress Field:





**MATLAB solution mesh refinement study, Plots of displacement ( $u_y$ ) and stress fields ( $\sigma_{yy}$ ), Rate of convergence, Table comparing the maximum stress ( $\sigma_{yy}$ ), where loading direction is parallel to the y-axis for each of the provided meshes.**

## CASE-A

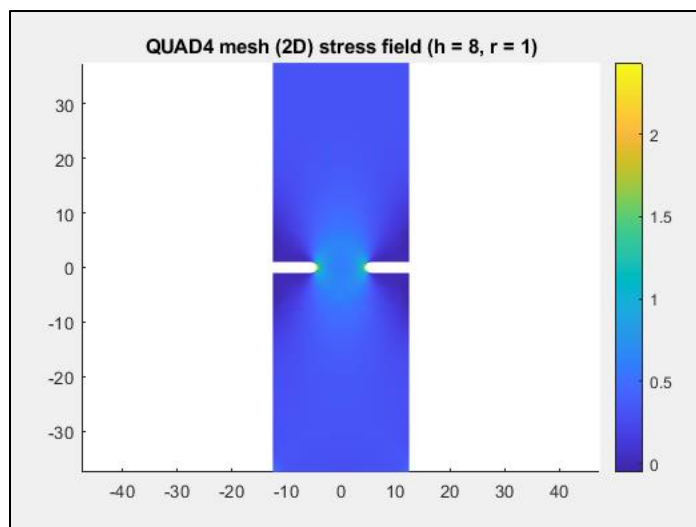
Program:

```

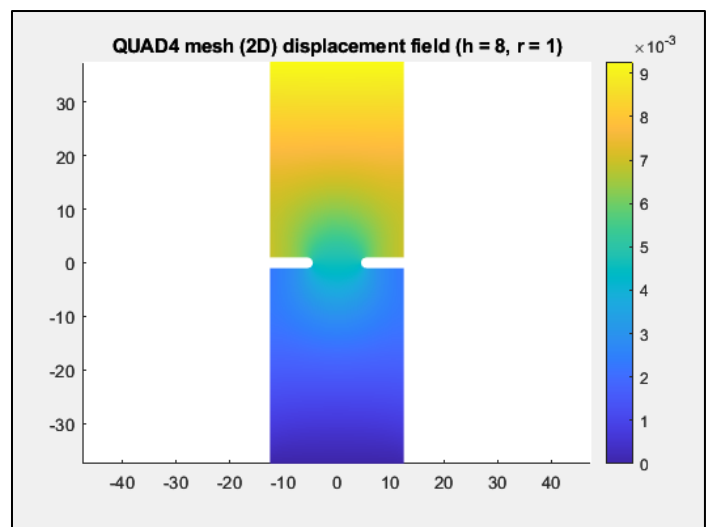
MainProjectCodeQUAD4Completed.m  OMI_RATE_CONVERGENCE.m  untitled.m  untitled *  +
1  %Program for Richardson extrapolation to estimate the rate of convergence of the maximum stress and the "exact" value of the maximum stress
2  %-----Program created by Omkar Gaikwad (ASU ID=1226843997) -----
3  % -----Harshal Tingre (ASU ID=1226039050)-----
4  %% CASE_A
5  % h=8 mm
6  % r=1 mm
7  %-----Case-A(Starts)-----
8  mesh1_r1 = abaqus_reader("C:\Users\omiga\OneDrive\Desktop\FEE_Project_2\quad4-meshes\u-notch2d-h8-r1-Q4-1.inp");
9  [sigyy1r1, max_stress1r1, maxKt1r1] = MainProjectCodeQUAD4(mesh1_r1, 8, 1);
10 mesh2_r1 = abaqus_reader("C:\Users\omiga\OneDrive\Desktop\FEE_Project_2\quad4-meshes\u-notch2d-h8-r1-Q4-2.inp");
11 [sigyy2r1, max_stress2r1, maxKt2r1] = MainProjectCodeQUAD4(mesh2_r1, 8, 1);
12 mesh3_r1 = abaqus_reader("C:\Users\omiga\OneDrive\Desktop\FEE_Project_2\quad4-meshes\u-notch2d-h8-r1-Q4-3.inp");
13 [sigyy3r1, max_stress3r1, maxKt3r1] = MainProjectCodeQUAD4(mesh3_r1, 8, 1);
14
15 % Finding the refinement ratio (R_Acase)
16 % Finding the order of convergence (P_Acase)
17
18 R_A_case = 2;
19 Omega_1=(abs(max_stress1r1 - max_stress2r1));
20 Omega_2=(abs(max_stress2r1 - max_stress3r1));
21
22 P_A_case = log(Omega_1/Omega_2)/log(R_A_case); % p is the order of convergence
23 disp('Different Meshes ratio= ')
24 disp(R_A_case)
25 disp('Order of Convergence= ')
26 disp(P_A_case)
27
28 % Finding the final and "exact" solution
29 Esol1 = max_stress3r1 + (max_stress3r1 - max_stress2r1)/(R_A_case^P_A_case - 1);
30 disp('the Final and "exact" solution ')
31 disp(Esol1)
32 %-----Case-A(Ends)-----
33 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Case-B%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
34

```

Output (Fine Mesh):



CASE-A (Stress Field)



CASE-A (Displacement Field)

## CASE-B

Program:

```
%Program for Richardson extrapolation to estimate the rate of convergence of the maximum stress and the "exact" value of the maximum stress
%-----Program created by Omkar Gaikwad (ASU ID=1226843997) -----
%-----Harshal Tingre (ASU ID=1226039050)-----

%% CASE_B
% h=8 mm
% r=2 mm
%-----Case-B(Starts)-----
mesh1r2 = abaqus_reader("C:\Users\omiga\OneDrive\Desktop\FEE_Project_2\quad4-meshes\u-notch2d-h8-r2-Q4-1.inp");
[sigyy1r2, max_stress1_r2, maxKt1r2] = MainProjectCodeQUAD4(mesh1r2, 8, 2);
mesh2r2 = abaqus_reader("C:\Users\omiga\OneDrive\Desktop\FEE_Project_2\quad4-meshes\u-notch2d-h8-r2-Q4-2.inp");
[sigyy2r2, max_stress2_r2, maxKt2r2] = MainProjectCodeQUAD4(mesh2r2, 8, 2);
mesh3r2 = abaqus_reader("C:\Users\omiga\OneDrive\Desktop\FEE_Project_2\quad4-meshes\u-notch2d-h8-r2-Q4-3.inp");
[sigyy3r2, max_stress3_r2, maxKt3r2] = MainProjectCodeQUAD4(mesh3r2, 8, 2);

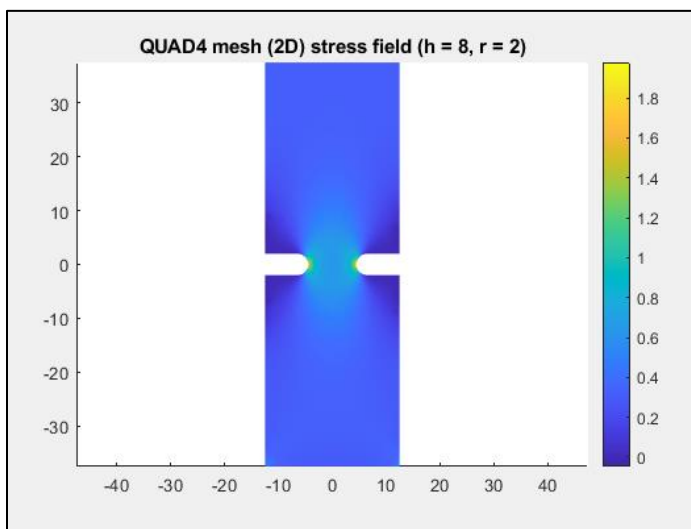
% Finding the refinement ratio (R_B_case)
% Finding the order of convergence (P_B_case)
R_B_case = 2; % r is the refinement ratio
Omega_1= (abs(max_stress1_r2 - max_stress2_r2));
Omega_2= (abs(max_stress2_r2 - max_stress3_r2));
P_B_case = log(Omega_1/Omega_2)/log(R_B_case); % p is the order of convergence

disp('Different Meshes ratio= ')
disp(R_B_case)

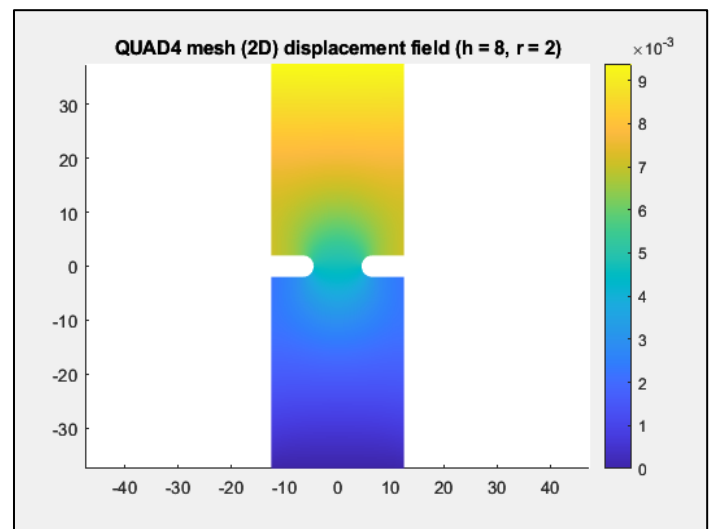
disp('Order of Convergence= ')
disp(P_B_case)

% Find "exact" solution
Esol2 = max_stress3_r2 + (max_stress3_r2 - max_stress2_r2)/(R_B_case^P_B_case - 1);
disp('the Final and "exact" solution ')
disp(Esol2)
%-----Case-B(Ends)-----
```

Output (Fine Mesh):



CASE-B (Stress Field)



CASE-B (Displacement Field)

## CASE-C

Program:

```
%Program for Richardson extrapolation to estimate the rate of convergence of the maximum stress and the "exact" value of the maximum stress
%-----Program created by Omkar Gaikwad (ASU ID=1226843997) -----
% -----Harshal Tingre (ASU ID=1226039050)-----
%% CASE_B
% h=6 mm
% r=3 mm
%-----Case-C(Starts)-----
mesh1 = abaqus_reader("C:\Users\omiga\OneDrive\Desktop\FEE_Project_2\quad4-meshes\u-notch2d-h6-r3-Q4-1.inp");
mesh2 = abaqus_reader("C:\Users\omiga\OneDrive\Desktop\FEE_Project_2\quad4-meshes\u-notch2d-h6-r3-Q4-2.inp");
mesh3 = abaqus_reader("C:\Users\omiga\OneDrive\Desktop\FEE_Project_2\quad4-meshes\u-notch2d-h6-r3-Q4-3.inp");

[sigyy1, max_stress_1, maxKt1] = MainProjectCodeQUAD4(mesh1, 6, 3);
[sigyy2, max_stress_2, maxKt2] = MainProjectCodeQUAD4(mesh2, 6, 3);
[sigyy3, max_stress_3, maxKt3] = MainProjectCodeQUAD4(mesh3, 6, 3);

% Finding the refinement ratio (R_B_case)
% Finding the order of convergence (P_B_case)
R_C_case = 2; % r is the refinement ratio
Omega_1=(abs(max_stress_1 - max_stress_2));
Omega_2=(abs(max_stress_2 - max_stress_3));

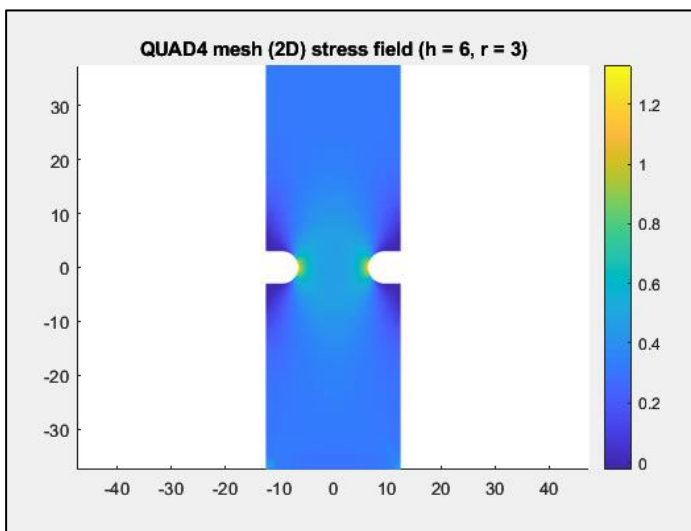
P_C_case = log(Omega_1/Omega_2)/log(R_C_case); % p is the order of convergence

disp('Different Meshes ratio= ')
disp(R_C_case)

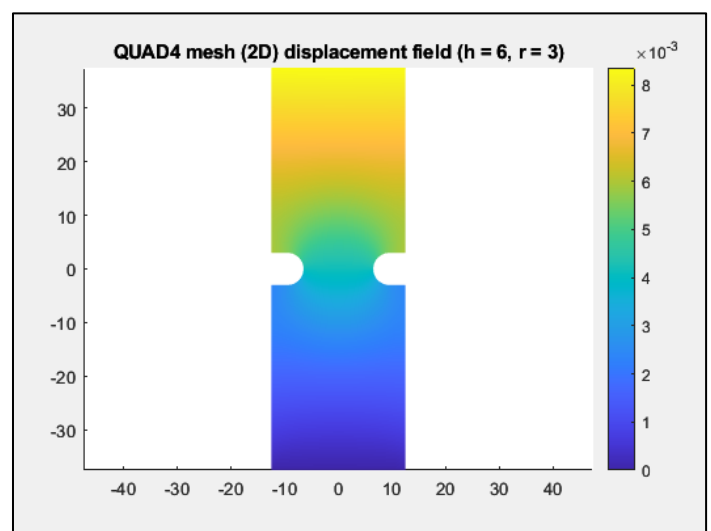
disp('Order of Convergence= ')
disp(P_C_case)

% Find "exact" solution
Esol = max_stress_3 + (max_stress_3 - max_stress_2)/(R_C_case^p - 1);
disp('The "exact" solution is: ')
disp(Esol)
%-----Case-C(Ends)-----
```

Output (Fine Mesh):



CASE -C (Stress Field)



CASE-C (Displacement Field)

**Table:****CASE-A**

The maximum stress in the y-y direction in MPa ( $\sigma_{yy}$ )	Different Meshes ratio	Order of Convergence	Final and "exact" solution
1. <u>Coarse Mesh</u> - For 1069 nodes and 990 elements $\sigma_{yy} = 2.4143$	2	2.760	2.6469
2. <u>Medium Mesh</u> - For 5854 nodes and 5677 elements $\sigma_{yy} = 2.6957$			
3. <u>Fine Mesh</u> - For 34374 nodes and 33988 elements $\sigma_{yy} = 2.6541$			

**CASE-B**

The maximum stress in the y-y direction in MPa ( $\sigma_{yy}$ )	Different Meshes ratio	Order of Convergence	Final and "exact" solution
1. <u>Coarse Mesh</u> - For 1238 nodes and 1153 elements $\sigma_{yy} = 1.9783$	2	0.2504	1.8608
2. <u>Medium Mesh</u> - For 6410 nodes and 6227 elements $\sigma_{yy} = 1.9596$			
3. <u>Fine Mesh</u> - For 36548 nodes and 36104 elements $\sigma_{yy} = 1.9438$			

**CASE-C**

The maximum stress in the y-y direction in MPa ( $\sigma_{yy}$ )	Different Meshes ratio	Order of Convergence	Final and "exact" solution
1. <u>Coarse Mesh</u> - For 1404 nodes and 1315 elements from $\sigma_{yy} = 1.3312$	2	0.4726	1.3023
2. <u>Medium Mesh</u> - 6201 nodes and 6008 elements $\sigma_{yy} = 1.3231$			
3. <u>Fine Mesh</u> - 38569 nodes and 38140 elements $\sigma_{yy} = 1.3173$			

## **Conclusion statement:**

### **Case A-**

- The maximum stress in the coarse mesh is much lower than the "exact" solution.
- Increasing the meshing brings the maximum stress closer to the "exact" solution.
- The maximum stress increases from a coarse to a medium mesh, but decreases from a medium to a fine mesh due to the small area of stress concentration, indicating that finer meshes are required for accurate results.

### **Case B-**

- In Case B, there is a steady, minor decrease in the maximum stress as the meshing increases, indicating that even a coarse mesh can produce accurate results compared to Case A.
- The mesh area where the maximum stresses are noted is much higher in Case B ( $h/r = 4$ ) compared to Case A.
- However, the "exact" solution of the maximum stress is slightly off compared to the maximum stresses observed in both cases, ranging from 1.95 to 1.97 compared to 1.86 for the "exact" solution.

### **Case C-**

- Case C exhibits a similar behavior to Case B regarding stress reduction with increased meshing.
- However, the maximum stress reduction in Case C is lower than in Case B due to larger U-notch areas.
- Compared to Case B, Case C has a better approximation of the "exact" solution for maximum stresses.

## **CONCLUSION ON BASIS OF PLOTS:**

- Maximum stresses are at the center of the U-notches in the stress field graphs.
- The maximum stresses decrease as the area of the notches increases.
- The stress is maximum at the smallest area because the pulling force is applied at the top of the section.
- The displacement fields are similar for  $h = 8, r = 1$  and  $h = 8, r = 2$ , with slightly larger maximum displacement for  $h = 8, r = 2$ .
- The maximum displacement for  $h = 6, r = 3$  is around 8 mm, and the variation in displacement is due to the differing stresses along the different cases.

Calculations and Plots of displacement ( $u_y$ ) and stress fields ( $\sigma_{yy}$ ) from your ABAQUS solution.

**CASE-A** ( $\sigma_{max} = 1541 \text{ mpa}$ )

$$\text{Stress concentration (Kt)} = \frac{\sigma_{max}}{\sigma_{normal}}$$

$$\sigma_{normal} = \frac{D \times t}{(D-2h) \times t} \times P$$

$$\begin{aligned} \therefore \sigma_{normal} &= \frac{D \times P}{(D-2h)} \\ &= \frac{25 \times 200}{25 - 2(8)} \end{aligned}$$

$$\therefore \sigma_{normal} = 555.56 \text{ MPa}$$

$$\therefore Kt = 2.773$$

**CASE-C** ( $\sigma_{max} = 786 \text{ mpa}$ )

$$\text{Stress concentration (Kt)} = \frac{\sigma_{max}}{\sigma_{normal}}$$

$$\sigma_{normal} = \frac{D \times t}{(D-2h) \times t} \times P$$

$$\begin{aligned} \therefore \sigma_{normal} &= \frac{D \times P}{(D-2h)} \quad \boxed{h=6} \\ &= \frac{25 \times 200}{25 - 2(6)} \end{aligned}$$

$$\therefore \sigma_{normal} = 384.61 \text{ MPa}$$

$$\therefore Kt = 2.043$$

**CASE-B**

$$Kt = \frac{\sigma_{max}}{\sigma_{normal}}$$

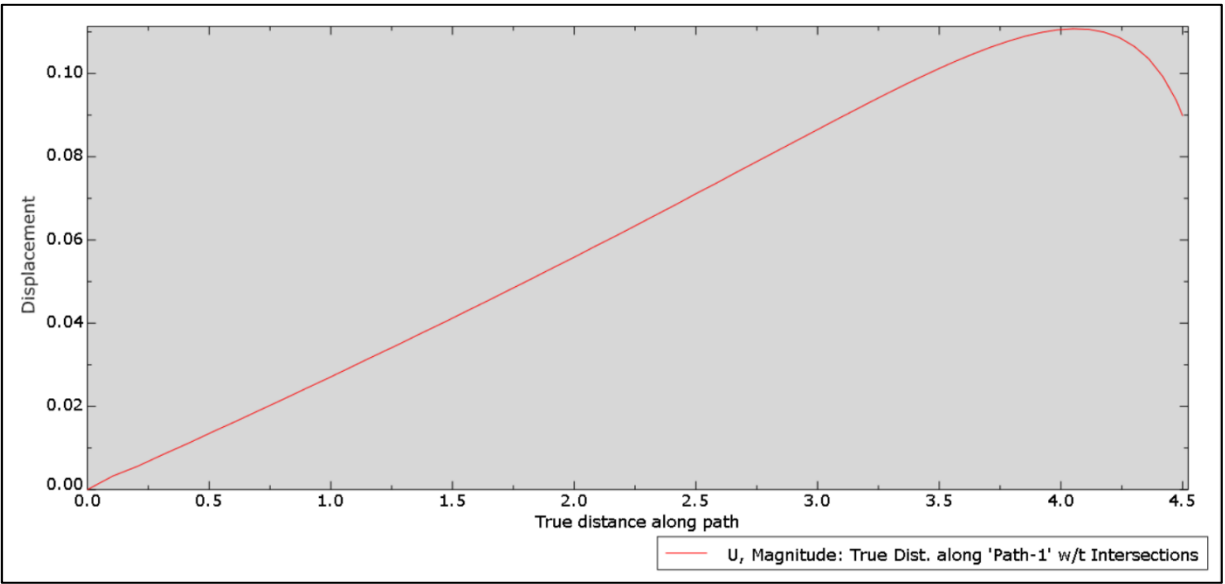
$\sigma_{normal}$  will be same as CASE-A. which is

$$\sigma_{normal} = 555.56 \text{ mpa}$$

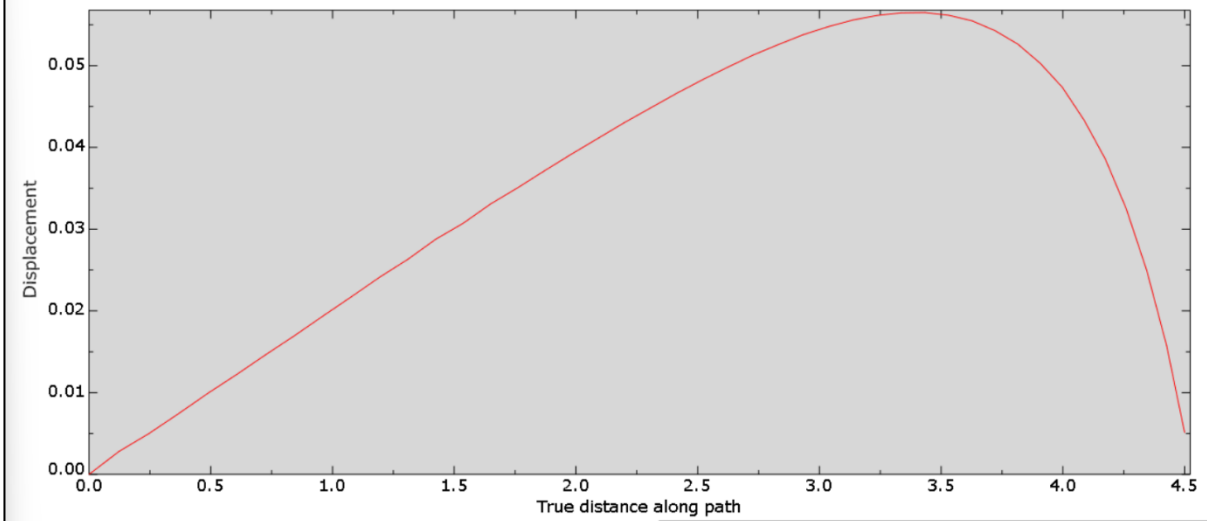
$$\therefore Kt = \frac{1160}{555.56}$$

$$\therefore Kt = 2.088$$

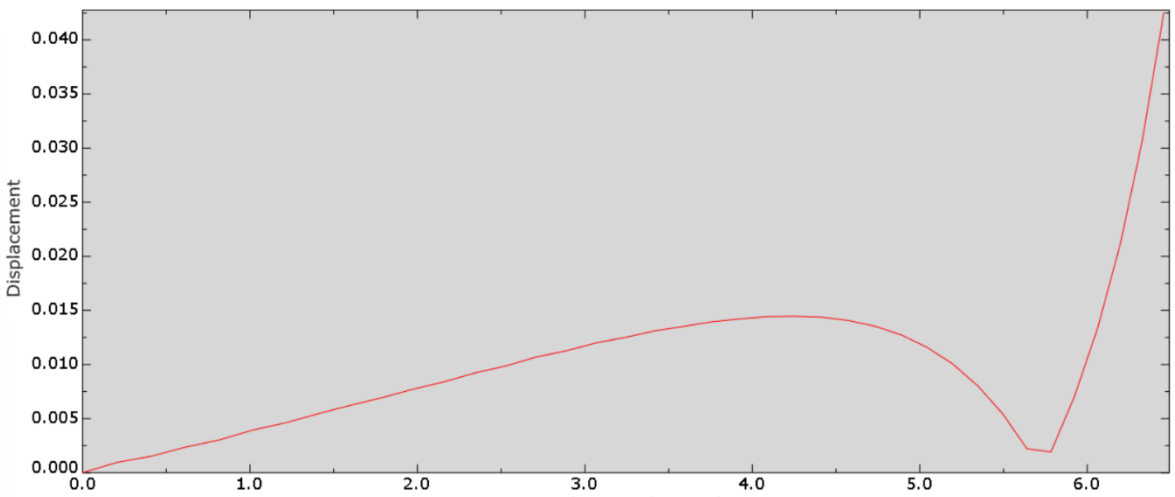




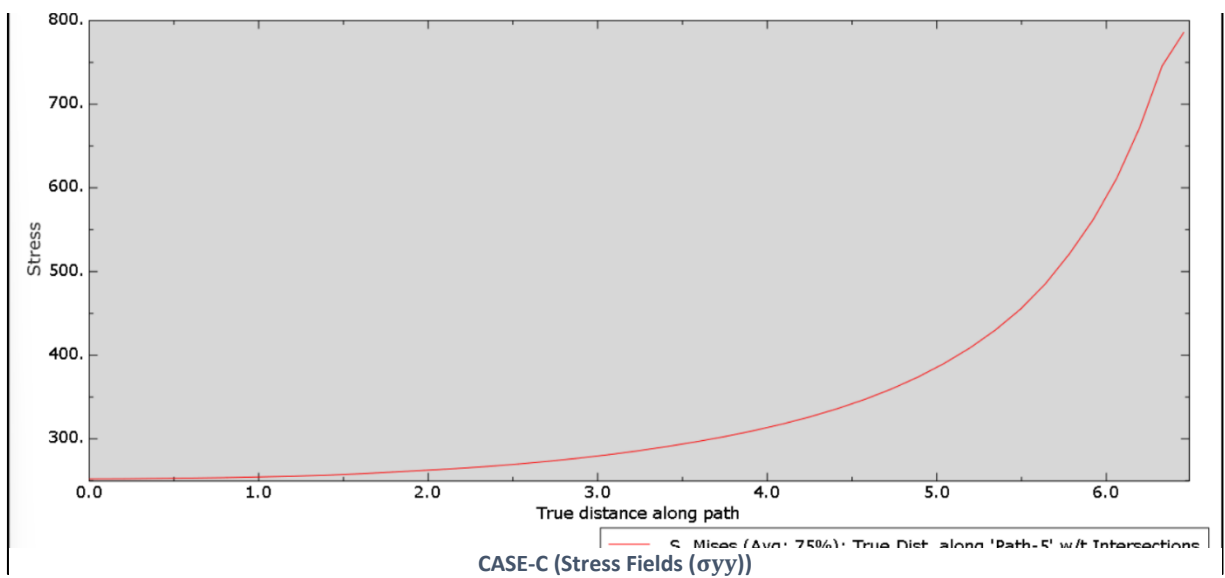
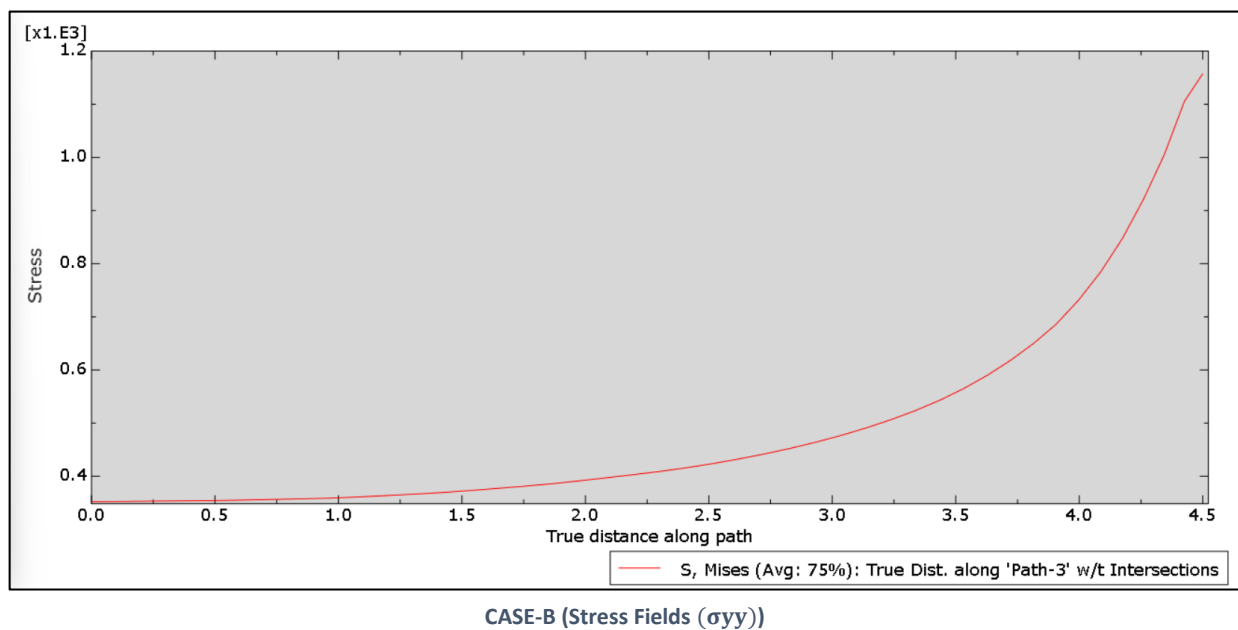
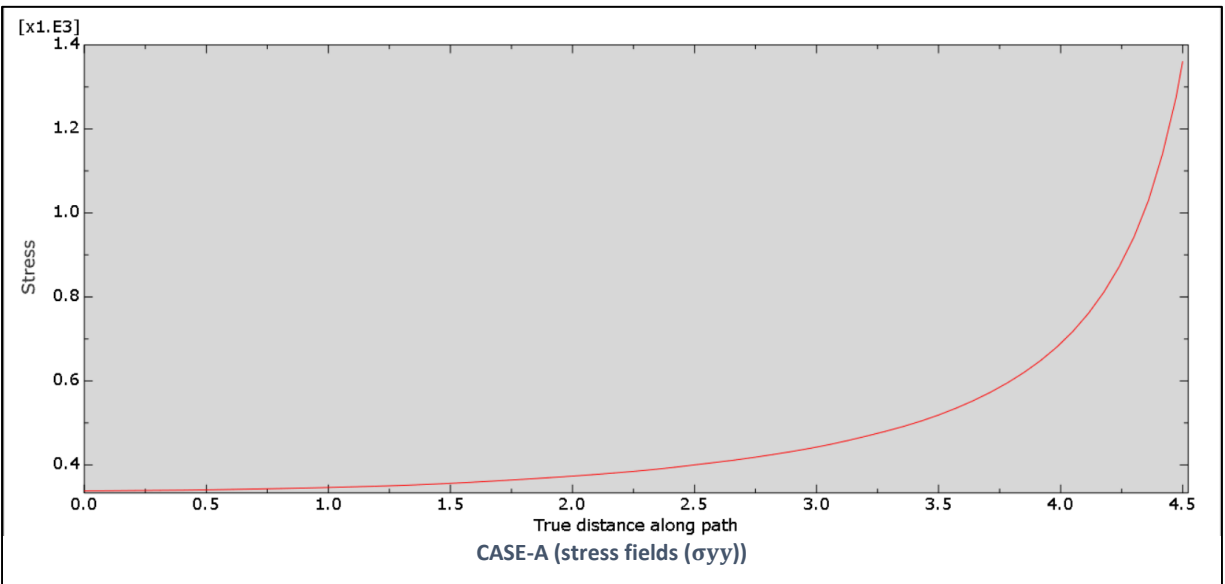
CASE-A (displacement ( $uy$ ))

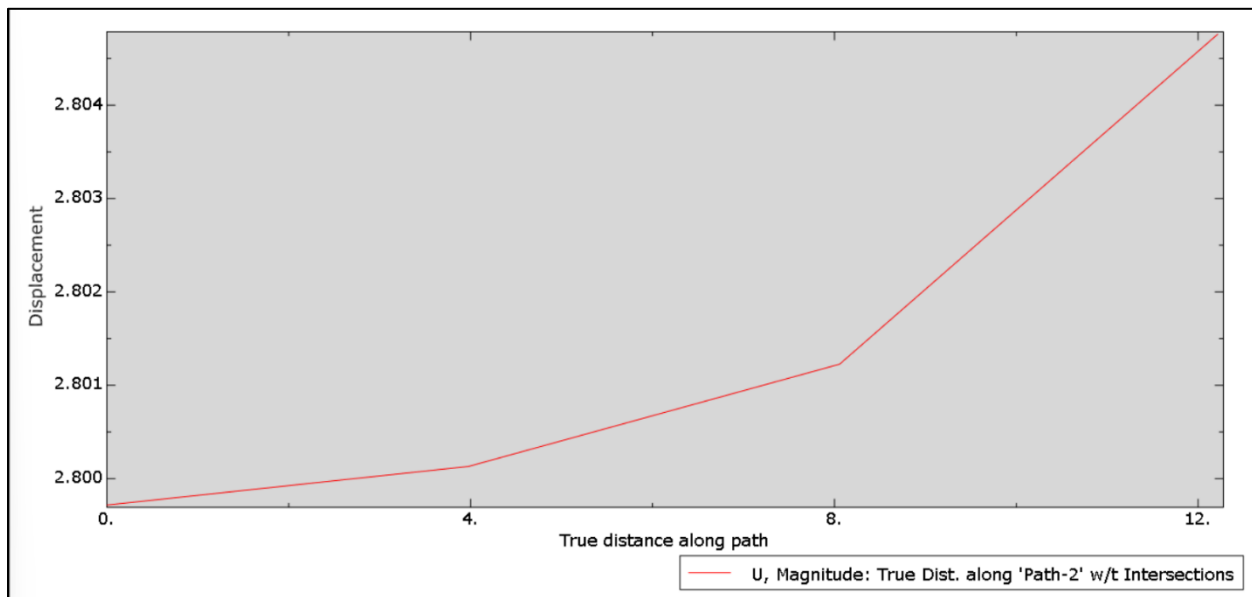


CASE-B (displacement ( $uy$ ))

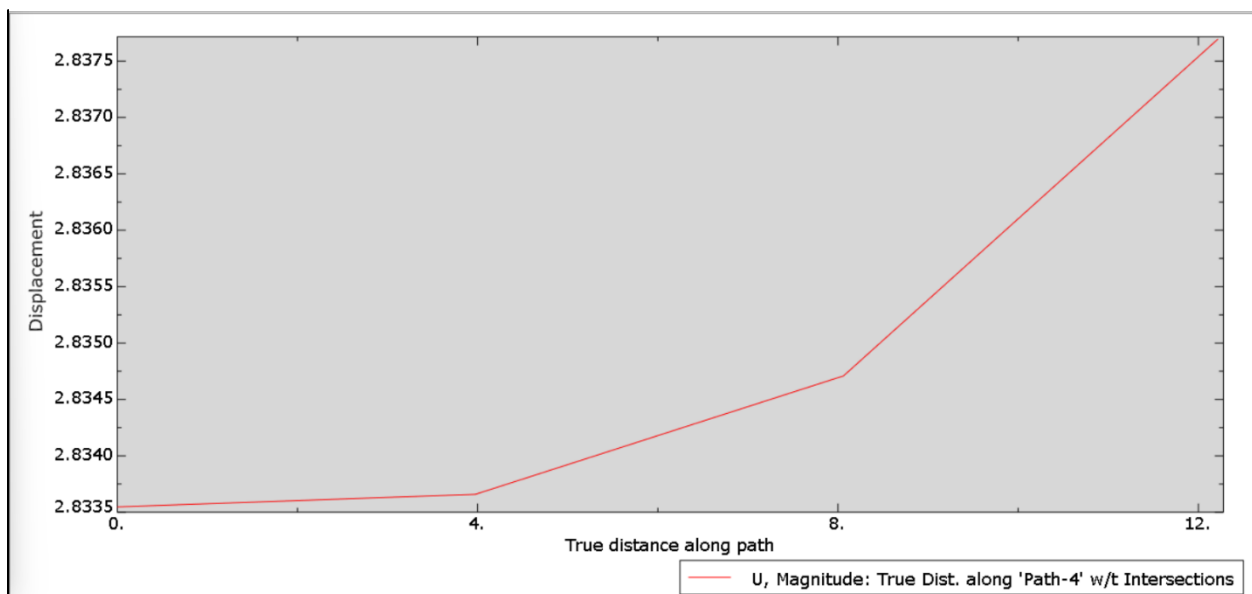


CASE-C (displacement ( $uy$ ))

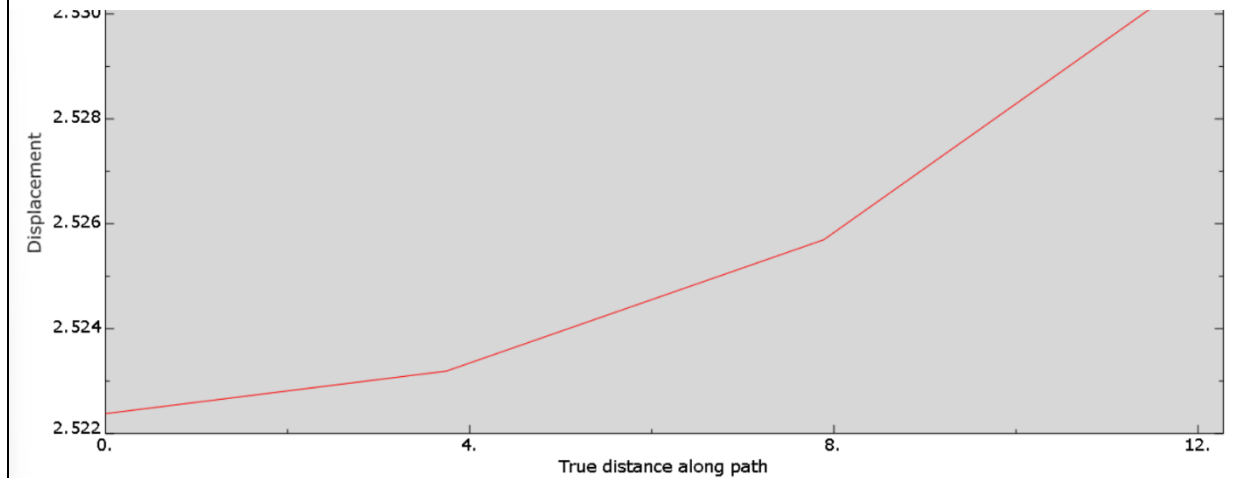




CASE-A (displacement ( $uy$ ))



CASE-B (displacement ( $uy$ ))



CASE-C (displacement ( $uy$ ))

Comparing the predictions for all three cases based on published stress concentration factor, the stress concentration factor computed using MATLAB code, and the stress concentration factor using ABAQUS code.

#### Case-A

Stress concentration factor $K_t$ (No unit)	Different Meshes ratio	$K_t$ (Roark's formula)	$K_t$ (ABAQUS)
1. <u>Coarse Mesh</u> - For 1069 nodes and 990 elements $\sigma_{yy} = 2.782$	2	2.80	2.78
2. <u>Medium Mesh</u> - For 5854 nodes and 5677 elements $\sigma_{yy} = 3.083$			
3. <u>Fine Mesh</u> - For 34374 nodes and 33988 elements $\sigma_{yy} = 3.033$			

#### Case-B

Stress concentration factor $K_t$ (No unit)	Different Meshes ratio	$K_t$ (Roark's formula)	$K_t$ (ABAQUS)
1. <u>Coarse Mesh</u> - For 1238 nodes and 1153 elements $\sigma_{yy} = 2.262$	2	2.14	2.09
2. <u>Medium Mesh</u> - For 6410 nodes and 6227 elements $\sigma_{yy} = 2.24$			
3. <u>Fine Mesh</u> - For 36548 nodes and 36104 elements $\sigma_{yy} = 2.220$			

#### Case-C

Stress concentration factor $K_t$ (No unit)	Different Meshes ratio	$K_t$ (Roark's formula)	$K_t$ (ABAQUS)
1. <u>Coarse Mesh</u> - For 1404 nodes and 1315 elements $\sigma_{yy} = 2.20$	2	2.17	2.04
2. <u>Medium Mesh</u> - For 6201 nodes and 6008 elements $\sigma_{yy} = 2.18$			
3. <u>Fine Mesh</u> - For 38569 nodes and 38140 elements $\sigma_{yy} = 2.17$			

### Conclusion statement:

1. Stress concentration factors obtained using FEM code and Roark's formula are generally higher than those obtained from ABAQUS due to limitations in the number of meshes ABAQUS can handle.
2. Coarse Mesh ' $Kt$ ' value, which is calculated with incomplete meshing, is lower than the values obtained from Roark's formulae.
3. Increasing meshing in Cases B and C results in decreased stress concentration factor.
4. Coarse Mesh ' $Kt$ ' value in Case A is much smaller than those of Medium and Fine Meshes due to incomplete meshing.
5. Finer meshing and appropriate calculation methods can improve the accuracy of stress concentration factor calculations.

Use the measured breaking loads to explain if the material fails at the same state of stress in all three cases:

Given Measured breaking loads (in lb.)

Four specimens were tested in each set.

CASE-->	A	B	C
1	219	423	521
2	254	299	434
3	173	315	405
4	148	246	439

Converting loads (lb.to N):

CASE-->	A $K_t$ (Roark's ) =2.80	B $K_t$ (Roark's ) =2.14	C $K_t$ (Roark's ) =2.17
1.	976.14 N	1881.6 N	2317.5 N
2.	1129.85 N	1330.01 N	1930.5 N
3.	769.54 N	1401.19 N	1801.5N
4.	658.3 N	1094.3 N	1952.7 N

Calculation of Nominal stress ( $\sigma_{nom}$ ):

$$\sigma_{nom} = \frac{Forces(F)}{\text{Cross – sectional area of the notch (A)}}$$

$$A = (25 - 2h) * 6.35$$

Where,

$h = 8 \text{ mm}$  (Case-A and Case- B)

$h = 6 \text{ mm}$  (Case C)

By calculating for each case, we get following table of exact values of Nominal stresses ( $\sigma_{nom}$ ) in MPa-



CASE-->	A $K_t$ (Roark's) =2.80	B $K_t$ (Roark's) =2.14	C $K_t$ (Roark's) =2.17
1.	17.04 MPa	32.92 MPa	28.07 MPa
2.	19.78 MPa	23.27 MPa	23.39 MPa
3.	13.46 MPa	24.52 MPa	21.82 MPa
4.	11.52MPa	19.15MPa	23.65 MPa

Maximum stress is a product of  $K_t$  (Roark's) and Nominal Stress:

$$\sigma_{max} = K_t * \sigma_{nom}$$

CASE-->	A	B	C
1	47.78 MPa	70.37 MPa	60.93 MPa
2	55.41 MPa	49.74 MPa	50.75 MPa
3	37.74 MPa	52.40 MPa	47.36 MPa
4	32.29 MPa	40.92 MPa	51.34 MPa
$\sigma_{max}$ (AVG.)	43.31 MPa	53.36 MPa	52.59 MPa

## MAIN CODE -

### 1-part

```
%-----Main Program of Quad_4 Complete code-----
%-----Program created by Omkar Gaikwad (ASU ID=1226843997) -----
% -----Harshal Tingre (ASU ID=1226039050)-----
%-----CODE STARTS-----
mesh = abaqus_reader("C:\Users\omiga\OneDrive\Desktop\FEE_Project_2\quad4-meshes\u-notch2d-h8-r1-Q4-1.inp");
%For other two CASES just change the path with respect to meshe file path
E = 3310; % Young's Modulus (MPa)
nu = 0.375; % Poisson's Ratio
OmegA=(E/(1-nu^2));
D = OmegA* [1 nu 0;
            nu 1 0;
            0 0 (1 - nu)/2];
M=(25*6.35);
Load_applied = 500; % Force applied to the rectangular bar in Newtons
Traction_applied = Load_applied/M; % Pressure on the rectangular bar in MPa
h = 8; %in mm
Thickness_D = 25; %in mm
ORA_t = 6.35; %in mm
nOMINAL_STREss = Load_applied/((Thickness_D - 2*h)*ORA_t);

nn = length(mesh.x); % Number of elements (QUAD4)
nne = 4; % Number of nodes per element (QUAD4)
W1=-sqrt(1/3);
W2= sqrt(1/3);
Quad_pts = [W1,W1,1; W2,W1,1; W2,W2,1; W1,W2,1];
Quad_Table = Quad_pts';
s1=2*nn;
s2=8*nn;
Ke = spalloc(s1,s1,s2); % Finding the stiffness matrix

for c = mesh.conn
    xe = mesh.x(:,c);
    dx = xe(:,2) - xe(:,1);
    sctr = [2*c - 1; 2*c];
    Ke = zeros(2*length(c));
    z0 = zeros(1, length(xe));
    for q = Quad_Table
        [N, dNdP] = shape4(q);
        J = xe * dNdP;
        dNdx = dNdP/J;
        Thickness_D = [dNdx(:,1)', z0; z0, dNdx(:,2)']; dNdx(:,2)', dNdx(:,1)'];
        Ke = Ke + Thickness_D' * D * Thickness_D * q(3) * det(J);
    end
    Ke(sctr,sctr) = Ke(sctr,sctr) + Ke;
end
```

### 2-Part

```

s_top = find(mesh.x(2,:) == max(mesh.x(2,:)));
[~,ii] = sort(mesh.x(1,s_top));
s_top = s_top(ii);
edge_conn = [s_top(1:1:end-1); s_top(2:1:end)];
%Stress = A\y;
%P.vertices = mesh.x';
%P.faces = mesh.conn';
%P.facevertexcdata = Stress(:,2);
quad_table_lin = [-0.577400000000000, 0.577400000000000; 1, 1];
% Force matrix
f = zeros(2*nn,1);
for c = edge_conn
    for q = quad_table_lin
        [N, dNdP] = shape2linear(q(1));
        xe = mesh.x(:,c);
        J = xe * dNdP;
        f(2*c) = f(2*c) + N * AppliedTraction * norm(J) * q(end);
    end
end
% Giving Boundary conditions
fixed_dof_pin = find((mesh.x(2,:) == min(mesh.x(2,:))))';
fixed_dof = [2*fixed_dof_pin; 2*fixed_dof_pin-1];
Ke(fixed_dof,:) = 0;
Ke(fixed_dof,fixed_dof) = eye(length(fixed_dof));
f(fixed_dof) = 0;
d = Ke\f;
A = spalloc(nn,nn,9*nn);
y = zeros(nn,3);
for c = mesh.conn
    xe = mesh.x(:,c);
    sctr = [2*c - 1; 2*c];
    de = d(sctr);
    Ae = zeros(length(c));
    z0 = zeros(1, length(xe));
    for q = Quad_Table
        [N, dNdP] = shape4(q);
        J = xe * dNdP;
        dNdx = dNdP/J;
        Thickness_D = [dNdx(:,1)', z0; z0, dNdx(:,2)'; dNdx(:,2)', dNdx(:,1)'];
        Stress = D * Thickness_D * de;
        Ae = Ae + N * N' * det(J) * q(end);
        y(c,:) = y(c,:) + N * Stress' * det(J) * q(end);
    end
    A(c,c) = A(c,c) + Ae;
end

```

### 3-Part

```
%finding Maximum stress in MPa
Stress = A\y;
P.vertices = mesh.x';
P.faces = mesh.conn';
P.facecolor = 'interp';
P.facevertexcdata = Stress(:,2);           % Maximum stress to be found
%plotting of graphs(Output)
figure(1); clf;
patch(P,'LineStyle','none')
axis equal;
colorbar;
axis([-15, 15, -40, 40]);
title('QUAD4 mesh (2D) stress field (h = 8, r = 1)')
disp('The maximum stress in the y-y direction in MPa is: ')
disp(max(P.facevertexcdata))
Kt = max(P.facevertexcdata)/NOMINAL_STRESS;
disp('The stress concentration factor of the cross-section (No unit) is:')
disp(max(Kt))
dy = d(2:2:end, 1);
P.vertices = mesh.x';
P.faces = mesh.conn';
P.facecolor = 'interp';
P.facevertexcdata = dy;
figure(2)
patch(P,'LineStyle','none')
hold on;
axis equal;
colorbar;
axis([-15, 15, -40, 40]);
title('QUAD4 mesh (2D) displacement field (h = 8, r = 1)')
%reference from assignments
function [N, dNdp] = shape4(p)
    psi = p(1);
    eta = p(2);
    N1 = 0.25*(1-psi)*(1-eta);
    N2 = 0.25*(1+psi)*(1-eta);
    N3 = 0.25*(1+psi)*(1+eta);
    N4 = 0.25*(1-psi)*(1+eta);
    N = [N1; N2; N3; N4];
    dNdpsi1 = 0.25*(eta - 1);
    dNdpsi2 = 0.25*(1 - eta);
    dNdpsi3 = 0.25*(1 + eta);
    dNdpsi4 = 0.25*(-1 - eta);
    dNdpsi = [dNdpsi1; dNdpsi2; dNdpsi3; dNdpsi4];
    dNdeta1 = 0.25*(psi - 1);
    dNdeta2 = 0.25*(- psi - 1);
    dNdeta3 = 0.25*(psi + 1);
    dNdeta4 = 0.25*(1 - psi);
    dNdeta = [dNdeta1; dNdeta2; dNdeta3; dNdeta4];
    dNdp = [dNdpsi dNdeta];
end

function [N, dNdp] = shape2linear(p)
    N1 = (1-p)/2;
    N2 = (1+p)/2;
    N = [N1; N2];
    dNdp = [-1/2; 1/2];
end
%-----CODE ENDS-----
```