CAD Laboratory (CE4P001) — Assignment 2
Session: Autumn 2025

by

Himanshu bairagi

(22CE02010)

School of Infrastructure

**Indian Institute of Technology, Bhubaneswar**

Odisha

October 2025

Question 1: deformed shape of body

(a) Plotting the Scalar Field

- Body domain was defined: X1_min, X1_max = 0.1, 1.0,   X2_min, X2_max = 0.1, 1.0.

- I am Creating 10 characteristic lines on the body in both directions.

- To plot the original body shape X1_boundary and X2_boundary are as followed:
  X1_boundary = [X1_min, X1_max, X1_max, X1_min, X1_min]
  X2_boundary = [X2_min, X2_min, X2_max, X2_max, X2_min]

- Then 10 characteristics lines (horizontal & vertical) are created on same original configuration.

- Plotting of deformed shape boundary are as followed:

```julia
# Deformed boundary
x1_boundary_def = zeros(length(X1_boundary))
x2_boundary_def = zeros(length(X2_boundary))

for i in 1:length(X1_boundary)
    X1 = X1_boundary[i]
    X2 = X2_boundary[i]
    # x_i = X_i + u_i
    x1_boundary_def[i] = X1 + u1(X1, X2) #where u1(x1,y1)=0.2*log(1+x1+x2)
    x2_boundary_def[i] = X2 + u2(X1, X2) #where u2(x1,x2)=0.2*exp(x1)
end
```

- Then 10 characteristics lines (horizontal & vertical) are created on same deformed configuration.
- `Plots.jl` is used to generate plots.

Original and Deformed Configuration
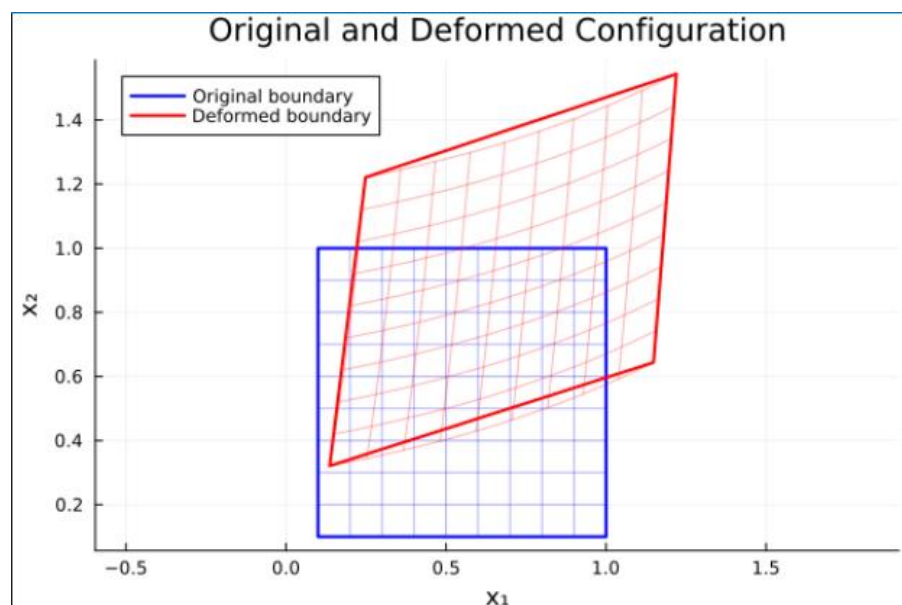
Question :2

- procedure is same as Q1.

- According to question displacement   functions are   defined as:

  u_r(R, θ)= 0.4 * (R - 1)^2 * cos(3 * θ)

  u_ θ(R, θ)= 0.4 * (R - 1)^3

- To convert polar to cartesian coordinates

```
function polar_to_cartesian(R, θ)
    X1 = R * cos(θ)
    X2 = R * sin(θ)
    return X1, X2
end
```

- To compute deformed position in cartesian coordinates:

```
function compute_deformed_position(R, θ)
    # Original position in cartesian coordinates
    X1, X2 = polar_to_cartesian(R, θ)

    # Displacement components in polar coordinates
    ur = u_r(R, θ)
    uθ = u_θ(R, θ)

    # Base vectors in polar coordinates
    e_r_1 = cos(θ)  # e_r · e_1
    e_r_2 = sin(θ)  # e_r · e_2
    e_θ_1 = -sin(θ) # e_θ · e_1
    e_θ_2 = cos(θ)  # e_θ · e_2

    # Displacement in cartesian coordinates
    # u = u_r * e_r + u_θ * e_θ
    u1 = ur * e_r_1 + uθ * e_θ_1
    u2 = ur * e_r_2 + uθ * e_θ_2

    # Deformed position: x = X + u
    x1 = X1 + u1
    x2 = X2 + u2

    return x1, x2
end
```
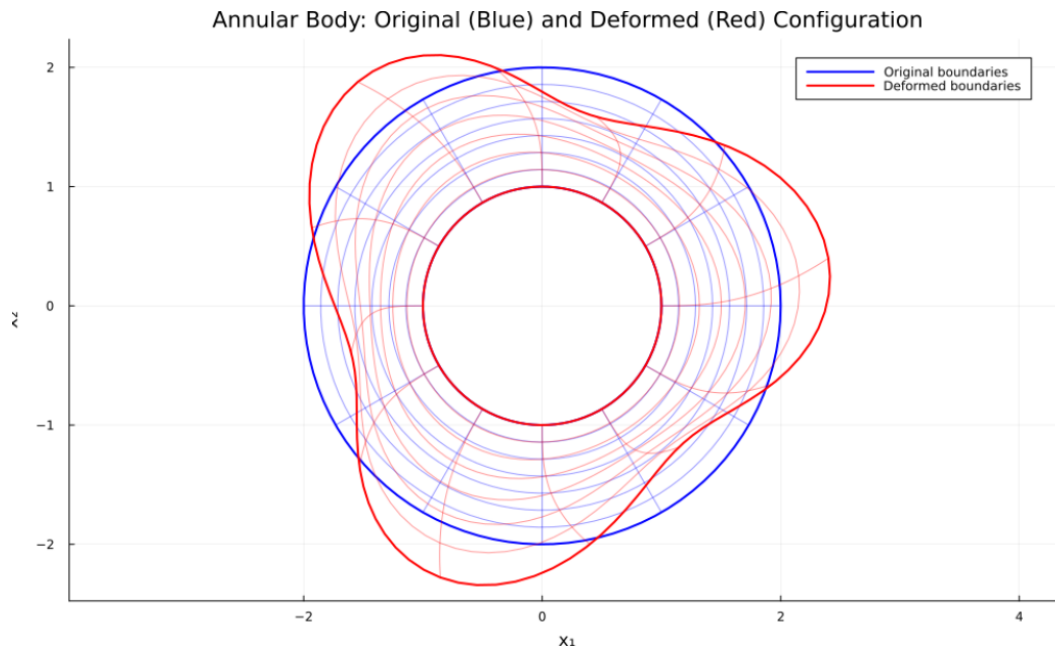
- Plotting procedure was same as Q1

Annular Body: Original (Blue) and Deformed (Red) Configuration

Question 3:

- The procedure was same as q1 and q2.

- According to question displacement functions are defined as:

  u_θ (x1,x2)= 0.2*log(1+x1+x2) and   u_r(x1,x2)= 0.2*exp(x1)

- To compute Compute polar angle θ from Cartesian coordinates:

```
function compute_theta(X1, X2)
    return atan(X2, X1)  # atan(y, x) gives correct quadrant
end
```

- To compute deformed position:

```
function compute_deformed_position(X1, X2)
    # Calculate polar angle at this point
    θ = compute_theta(X1, X2)

    # Displacement components in polar basis
    ur = u_r(X1, X2)
    uθ = u_θ(X1, X2)

    # Polar base vectors at this point
    # e_r = cos(θ)e_1 + sin(θ)e_2
    # e_θ = -sin(θ)e_1 + cos(θ)e_2
    e_r_1 = cos(θ)
    e_r_2 = sin(θ)
    e_θ_1 = -sin(θ)
    e_θ_2 = cos(θ)

    # Converting displacement to Cartesian components
    # u = u_r * e_r + u_θ * e_θ
    u1 = ur * e_r_1 + uθ * e_θ_1
    u2 = ur * e_r_2 + uθ * e_θ_2
```

```
# Deformed position: x = X + u
x1 = X1 + u1
x2 = X2 + u2

return x1, x2
end
```

- Plotting procedure was same as Q1

Original (Blue) and Deformed (Red) Configuration
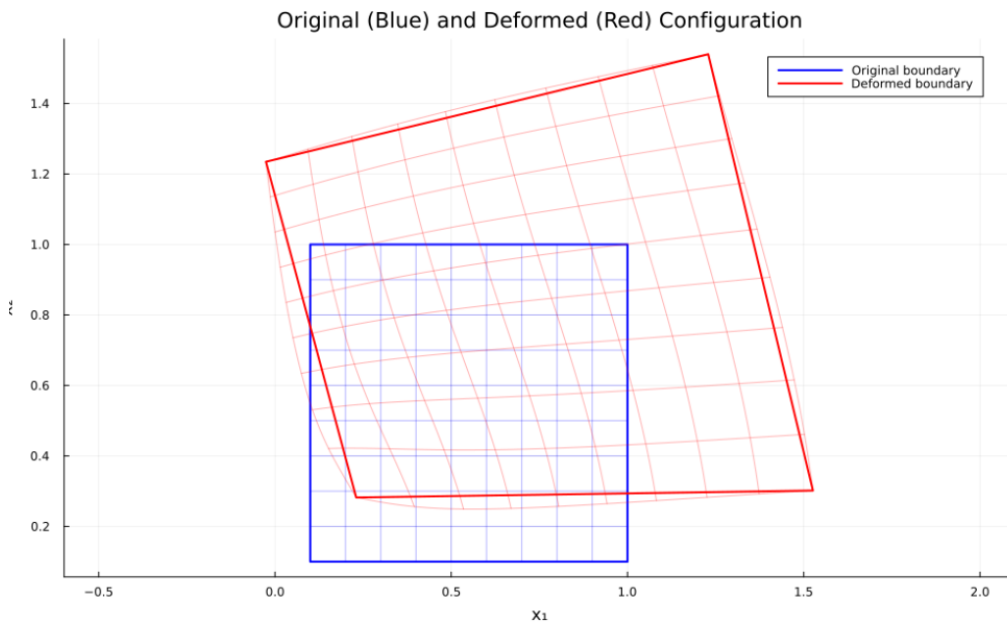
Question :4 2-D planner Plate Hole problem

Using Gmsh :-

- Started by entering the geometric parameters (L, H, thickness, hole radius).
- Placed the required corner points and the points describing the hole.
- Constructed the outer rectangle and the circular hole contours.
- Defined the plate's 2D area by subtracting the inner circle from the rectangle.
- Converted the 2D region into a 3D solid via extrusion.
- Added physical groups to organize boundaries and the main volume.
- Created the 3D finite element mesh.
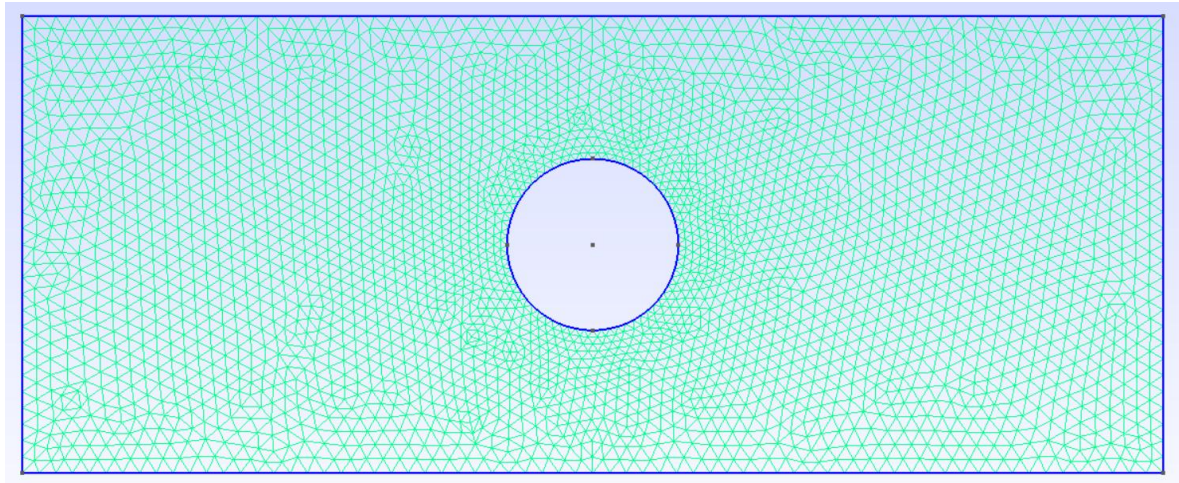
Governing PDEs:

1. $e_{ij} = \frac{1}{2}\left(u_{i,j} + u_{j,i}\right)$ 2. $\sigma_{ij} = \lambda\, e_{kk}\, \delta_{ij} + 2\mu\, e_{ij}$ 3. $\sigma_{ij,j} = 0$

The Combined Navier Equation (in displacement form) : $\mu\nabla^2 u + (\lambda + \mu)\nabla(\nabla \cdot u) + F = 0$
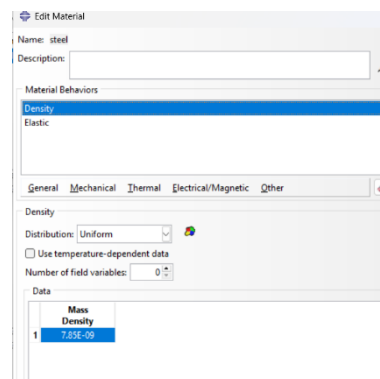
Using abaqus:

i. In part module a 2D shell part was created.
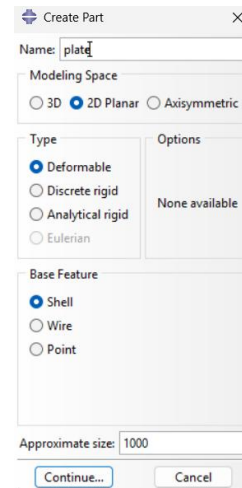
ii. Geometry was defined.
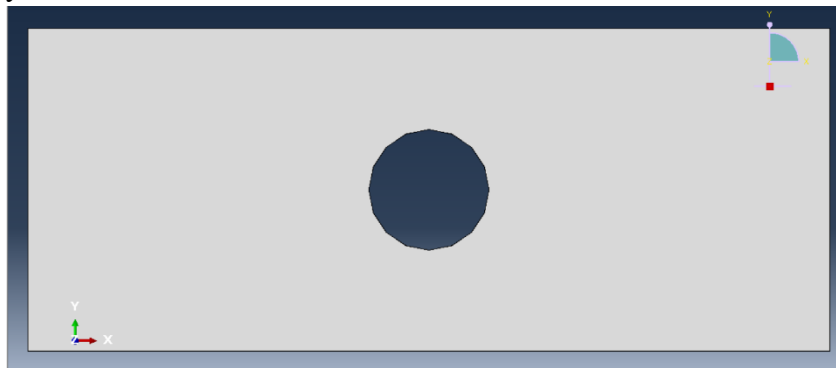
iii. In property module, properties mass density 7.85e-9 tonnes/mm^3, elastic modulus 210000 MPa and Poissons ratio 0.3 were defined. Then Section was created and 2mm thickness was defined.
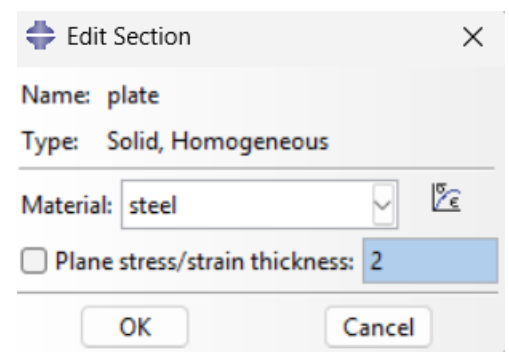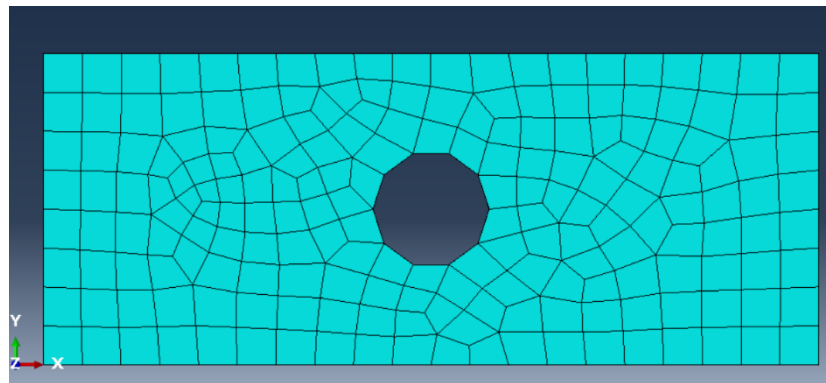

iv. Created instance on instance module. Then step was defined on step module.

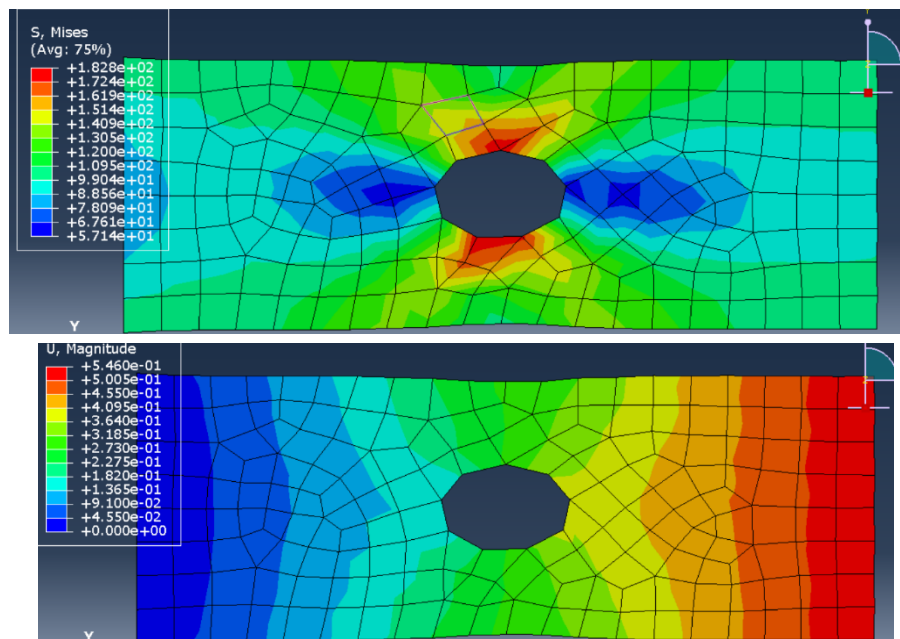v. A pressure load of 100e3 N/m was applied and fully constrained encastered boundary condition was set.

vi.   A pressure load of 100e3 N/m was applied and fully constrained encastered boundary
        condition was set.
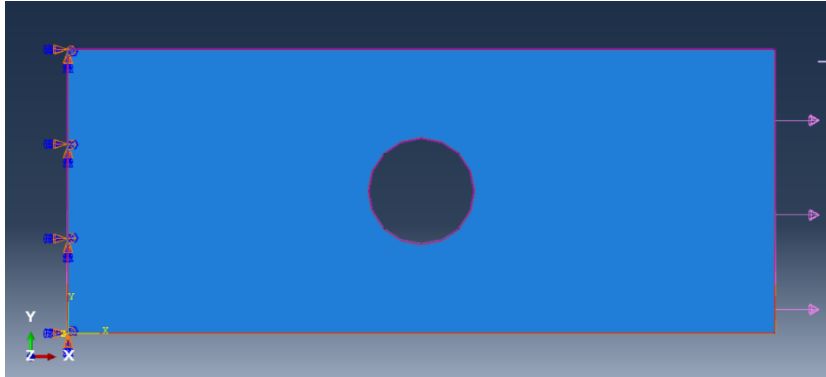

vii.   Mesh was generated.

viii.   Job was created and results for Von-Mises stress and displacement were visualized.

Q5: Cantilever beam with point load at free end

Using Gmsh:-

- Defined beam dimensions L,W,D
- Added rectangular cross-sectional points in Gmsh.
- Created lines and surface loop.
- Extruded surface to form cantilever volume.
- Governing PDEs was same as the Q4.
- Boundary conditions are: Fixed at left end, point load at other end.
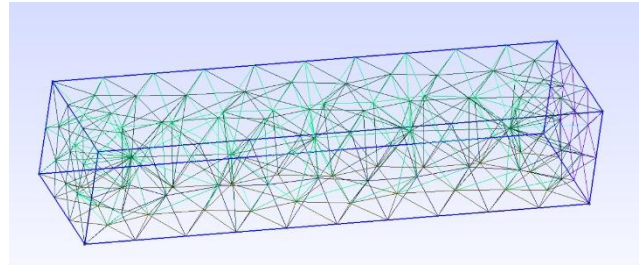

Using abaqus:

- Process was same But now in part module a 3d solid extrude was defined. Then geometry was created.
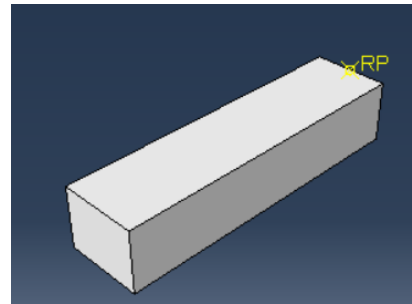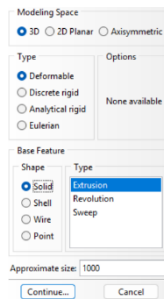
- Properties were defined according to given question then instance and step was created.
- In the load module A reference point was created to apply point load on that also coupling constraints were given to this reference point in the interaction module. Point load value 100N was defined at that reference point then encastered fix boundary condition was defined in the other end.
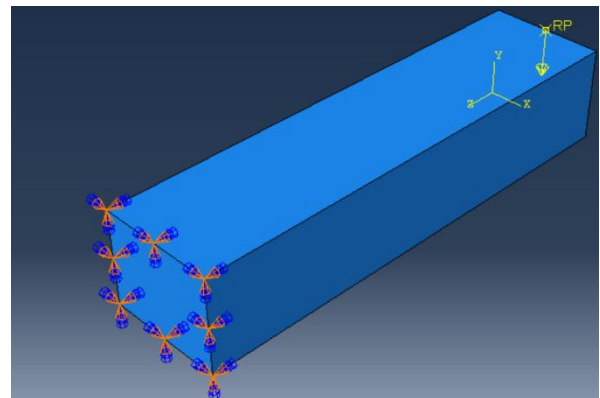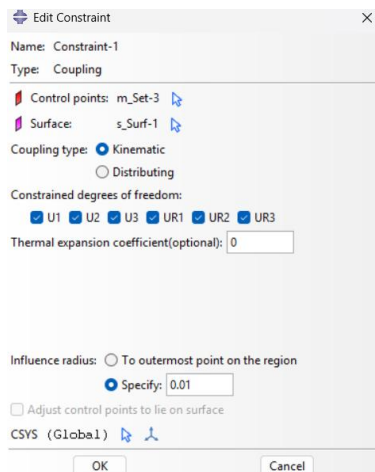
- Meshing was done.
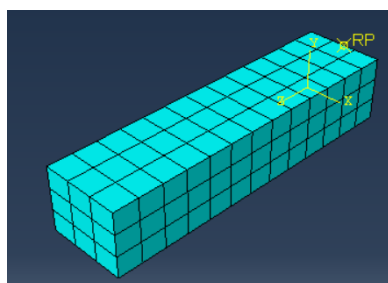
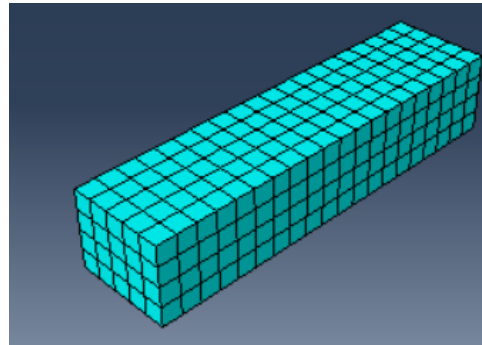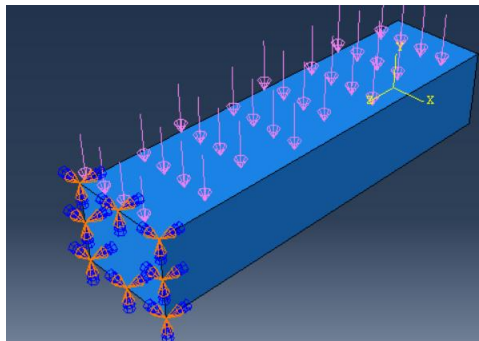- Results for von-mises stress and displacement were obtained.
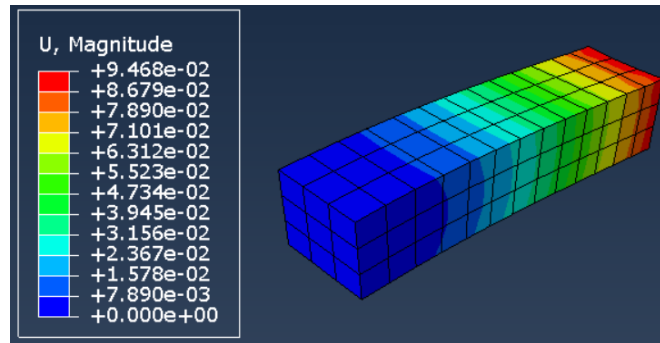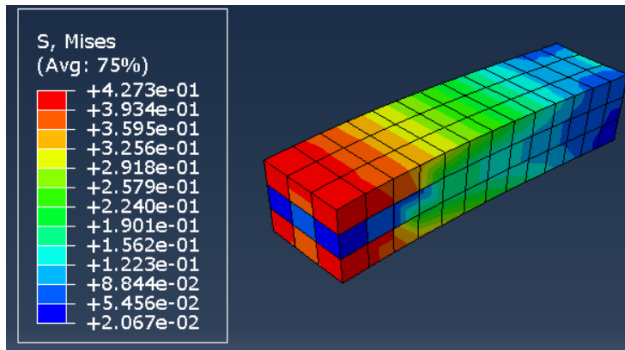
Q6: Cantilever beam with Uniform pressure at the Top.

Using Gmsh:

- The procedure for meshing was same as Q5.

- Only boundary conditions were different: fixed end + UDL on top face and other ends were traction free.

Using abaqus:

- Procedure was also same as Q5. Didn't need to define reference point and there constraints. Pressure were applied on top face and fix condition at Left end.

- Results were obtained for von-mises stress and displacement.