Problem Set 4

Muthu Ram Kumar Avichi (1010188967)

Question 1

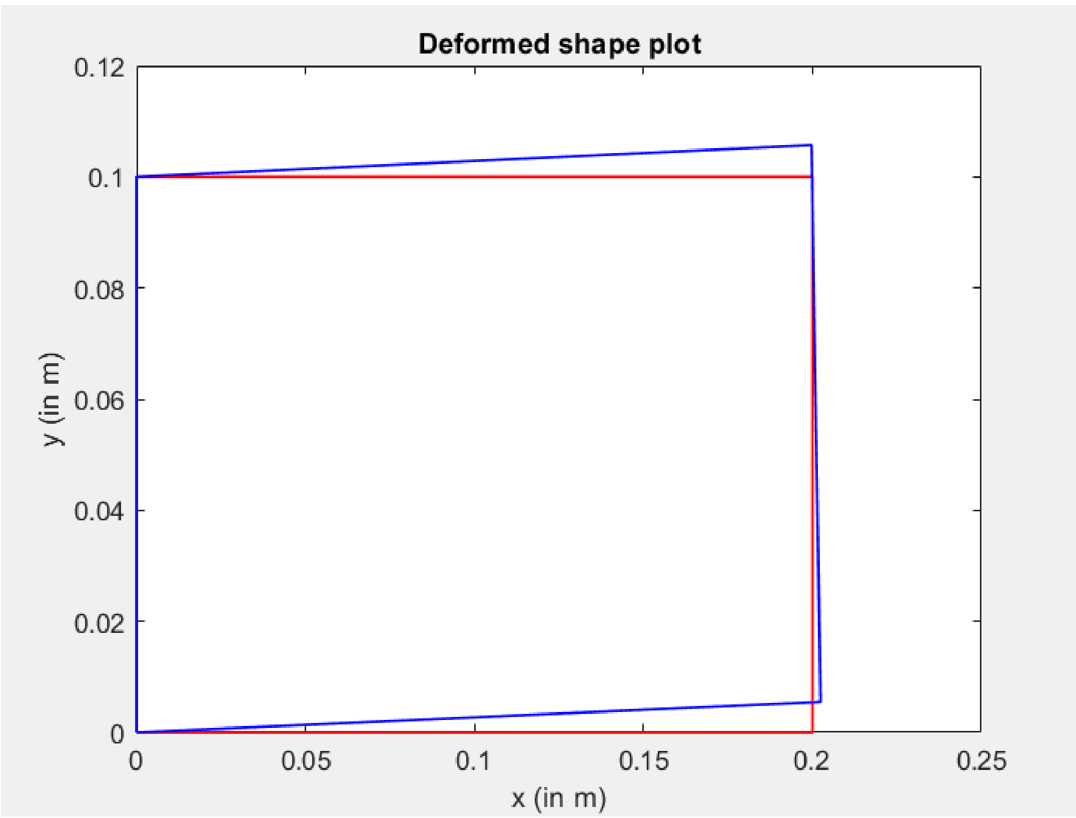
Matlab Code:

```
% ------------------------------------------------------------------------
% Matlab code for Problem Set 4 - Question 1
% Deformed shape, Von Mises stress.
% ------------------------------------------------------------------------
% clear data space
clear;
% close figure windows
close all;
% element geometry
x_e = [0 0.2 0.2 0]; y_e = [0 0 0.1 0.1]; % nodal locations (m)
area = (0.2*0.1); % area of element (m^2)
% initialize a few variables
K = zeros(8); % total stiffness matrix
K_star = zeros(8); % current term of stiffness matrix
f = zeros(8,1); % nodal force vector
d = zeros(8,1); % nodal displacement vector
t = 0.01; % Thickness
E = 120e9; % Young's modulus (Pa)
nu = 0.25; % Poisson's ratio
% matrix of elastic constants
D = (E/(1-nu^2)) * [1 nu 0; nu 1 0; 0 0 (1-nu)/2]; % (Pa)
% set up Gauss quadrature
xi = zeros(2);
eta = zeros(2);
xi(1) = - 1 / sqrt(3); xi(2) = 1 / sqrt(3); % Gauss points for x-direction
eta(1) = - 1 / sqrt(3); eta(2) = 1 / sqrt(3); % Gauss points for y-direction
w(1) = 1; w(2) = 1; % weights for Gauss quadrature
ax = 0; bx = 0.2;    % Limits for the x values
ay = 0; by = 0.1;    % Limits for the y values
J_det = ((bx - ax)/2)*((by - ay)/2); % determinant of Jacobian of
transformation to (xi, eta)
x = [gauss(ax, bx, xi(1)) gauss(ax, bx, xi(2))]; y = [gauss(ay, by, eta(1))
gauss(ay, by, eta(2))]; % physical locations of Gauss points (m)
for i = 1:2
   for j = 1:2
       % value of H at current Gauss point
       H = (1 / area) * [(y(j) - y_e(4)), 0, -(y(j) - y_e(4)), 0 (y(j) -
y_e(1)), 0 , -(y(j) - y_e(1)), 0;
       0, (x(i) - x_e(2)), 0, -(x(i) - x_e(1)), 0, (x(i) - x_e(1)), 0, -(x(i)
- x_e(2));
       (x(i) - x_e(2)), (y(j) - y_e(4)), -(x(i) - x_e(1)), -(y(j) - y_e(4)),
(x(i) - x_e(1)), (y(j) - y_e(1)),-(x(i) - x_e(2)), -(y(j) - y_e(1))];
       % contribution to stiffness matrix from current Gauss point
       K_star = w(i) * w(j) * J_det * H' * D * H;
```

```
        % current total stiffness matrix
        K = K + K_star;
    end
end
% multiply thickness to the K matrix
K = t.*K;
% assemble total applied force vector
% force is 10 kN at third node, dofs 5 and 6 at a 45 degree angle.
f = 10000 * [0; 0; 0; 0; 0.70711; 0.70711; 0; 0]; % N
% partition matrix; dofs 1, 2, 7, 8 are zero
fixed = [1 2 7 8];
dofs = 1:1:8;
free = setdiff(dofs, fixed);
% solve matrix equation K d = f
d(free,:) = K(free, free) \ f(free,:);
disp(d);
% ------------------------------------------------------------------------
% Display the deformed shape
scale = 100;
d = scale.*d;
figure(1);
lx_e = [0 0.2 0.2 0 0]; ly_e = [0 0 0.1 0.1 0];
plot(lx_e, ly_e, 'r', 'LineWidth',1);
hold on;
ldx_e = [0+d(1) 0.2+d(3) 0.2+d(5) 0+d(7) 0+d(1)]; ldy_e = [0+d(2) 0+d(4)
0.1+d(6) 0.1+d(8) 0+d(2)];
plot(ldx_e, ldy_e, 'b', 'LineWidth',1);
xlabel('x (in m)');
ylabel('y (in m)');
title('Deformed shape plot');
hold off;
d = 0.01.*d;
% ------------------------------------------------------------------------
% Von Misses Stresses computation
% Create Meshgrid
n = 1000;    % Number of nodes in the grid
x = linspace(0, 0.2, n);
y = linspace(0, 0.1, n);
% Initialize von_mises arrays
von_mises = zeros(n,n);
for i = 1:n
    for j = 1:n
        % value of H at current Gauss point
        H = (1 / area) * [(y(j) - y_e(4)), 0, -(y(j) - y_e(4)), 0 (y(j) -
y_e(1)), 0 , -(y(j) - y_e(1)), 0;
        0, (x(i) - x_e(2)), 0, -(x(i) - x_e(1)), 0, (x(i) - x_e(1)), 0, -(x(i)
- x_e(2));
        (x(i) - x_e(2)), (y(j) - y_e(4)), -(x(i) - x_e(1)), -(y(j) - y_e(4)),
(x(i) - x_e(1)), (y(j) - y_e(1)),-(x(i) - x_e(2)), -(y(j) - y_e(1))];
        % Stress & Strain from lecture notes
        strain = H*d;
        stress = D*strain;
```

```
        von_mises(i, j) = sqrt(stress(1)^2 - (stress(1) * stress(2)) +
stress(2)^2 + (3 * stress(3)^2));
   end
end
% ------------------------------------------------------------------------
% Contours for von mises stress
figure(2);
clabel(contourf(x, y, von_mises));
colorbar;
hold on;
plot(lx_e, ly_e, 'r', 'LineWidth',1);
xlabel('x (in m)');
ylabel('y (in m)');
title('Von Mises Stress Contour for given element');
hold off;
% ------------------------------------------------------------------------
% Function definitions
% Function to compute the Gauss quadrature.
function result = gauss(a, b, c)
   % This function calculates the average of the square of a and b,
   % and the product of c and d.
   % Calculate the square of a and b, then find their average
   result = (a*(1 - c)/2) + (b*(1 + c)/2);
end
```

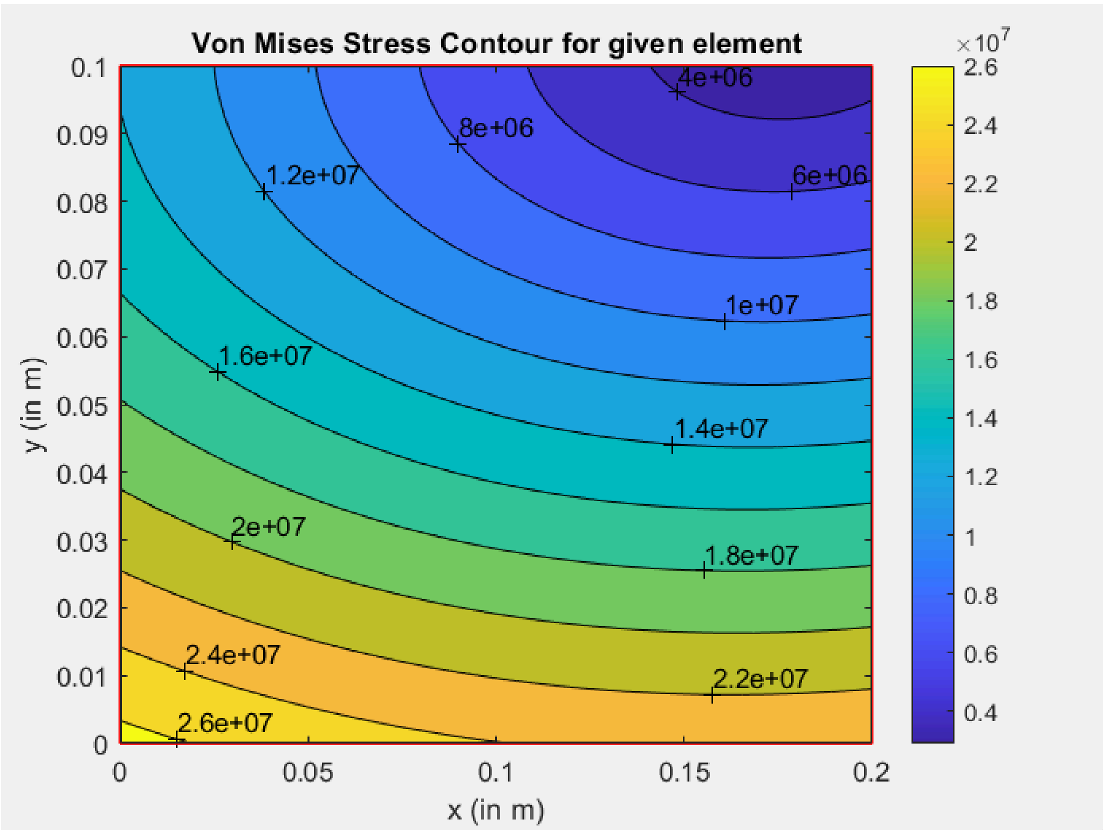
Output:



Values of displacement :

```
   1.0e-04 *

         0
         0
    0.2378
    0.5492
   -0.0274
    0.5704
         0
         0
```



Explanation:

The code blocks given in the lecture slides are followed systematically to implement the displacement calculations. Steps for the computation,

- Gauss points are computed from Gauss quadrature.
- Compute stiffness matrix from the equation, iterating over four gauss points.
- Multiply thickness to the stiffness matrix.

- Compute the Von Mises stress after computing strain and stress. Use resolved components in the equation: von_mises = sqrt(stress(1)^2 - (stress(1) * stress(2)) + stress(2)^2 + (3 * stress(3)^2));
- Plot contours of von_mises stress alongside meshgrid components.