National Taiwan University **Finite Element Method**
Department of Civil Engineering Instructors: C.-S. CHEN

Lab Assignment 7, 05/16/2019, 1800 -- 2000

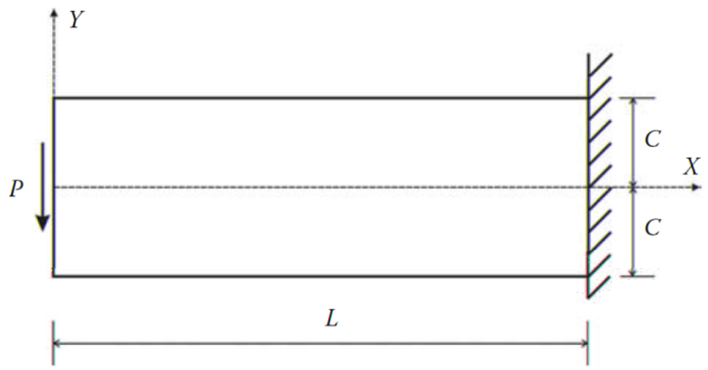
Due 2000

Lab Grading Policy: Attendance 20%, Score 80%, Bonus 20%

You are expected to complete the basic part during the Lab. In case you have difficulty in finishing the basic part on time, you should upload them before 2100 on Saturday and a penalty of 20% discount will be applied on your score. You are encouraged to complete the bonus part (no penalty applied). Basic and/or bonus parts should be submitted by **2100 on Saturday and no late submission is permitted**. We will in general post the reference solutions by **Sunday**.

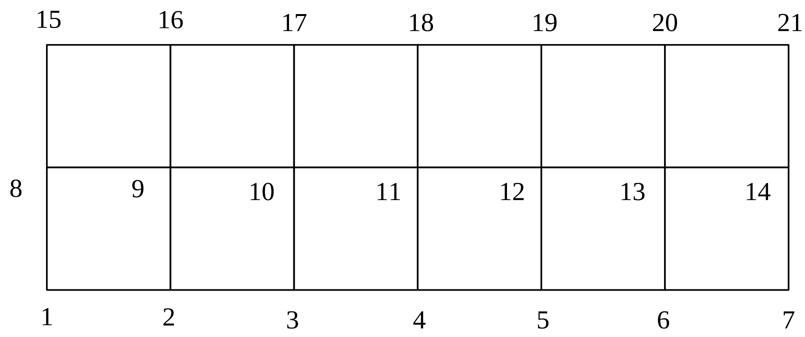
Download `RecSimple.zip` from the course website and unzip it. You will find a folder containing `problem2dTensileRec.m` file with seven functions, `drawingMesh.m` `formStiffnessRec.m` `guass2d.m` `outputDisplacements.m` `shapeFunctionQ4.m` `solution.m`.

1. **(40%)** Consider a cantilever subjected to a concentrated load shown below. Let us consider it as a plane-stress problem where C = 10 mm, L = 60 mm, t = 5 mm for the geometrical properties, a , ratio of 0.3 for the material properties, as well as a concentrated force P of 1000 N.

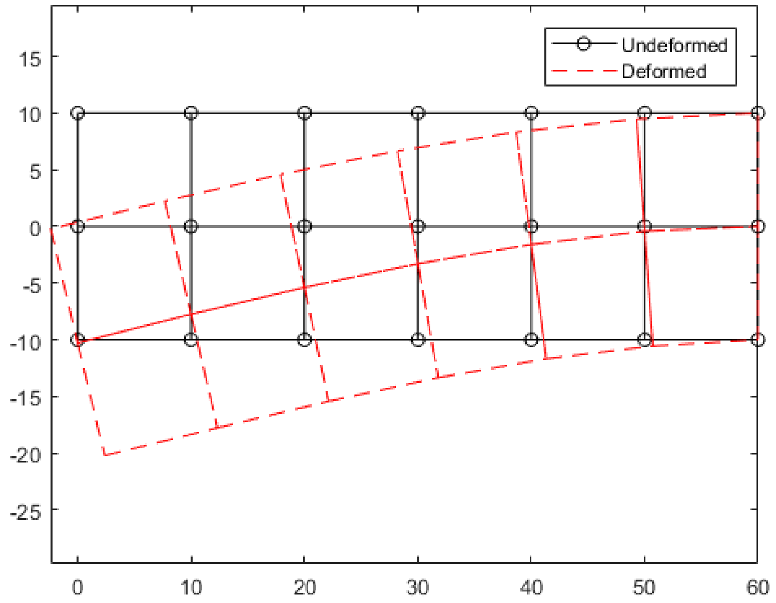


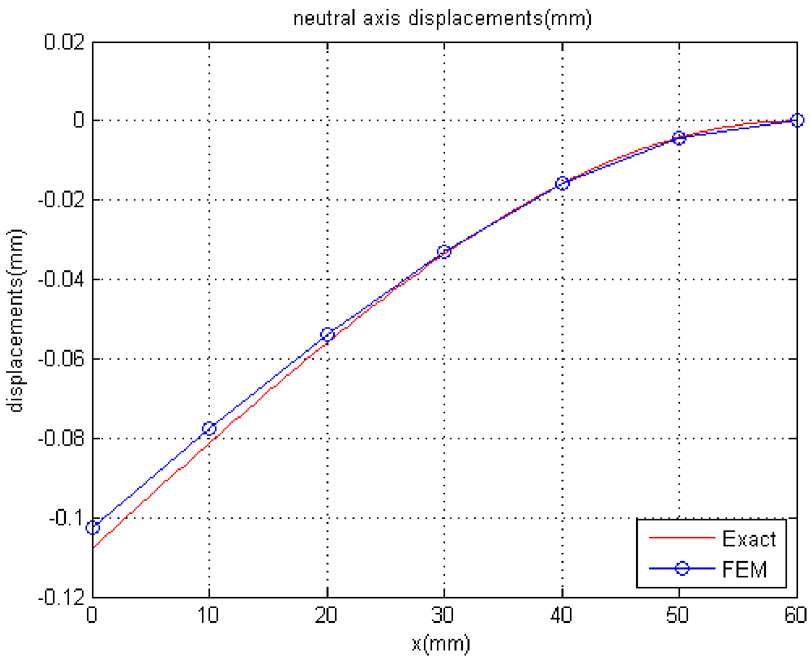
Use 12 rectangular elements to discretize the domain as shown in the figure below. The nodes numbered 7, 14, 21 represent the fixed end. The concentrated force P is applied at node 8. Plot the undeformed and deformed (magnification factor = 100) configurations and the vertical displacement of the nodes situated along the neutral axis of the cantilever and compare the results with analytical solution that gives the vertical displacement of any point in the domain.

$$u_y = \frac{\nu P x y^2}{2EI} + \frac{P x^3}{6EI} - \frac{P L^2 x}{2EI} + \frac{P L^3}{3EI}$$



Below are sample plots:






```
% A Thin Plate Subjected to Uniform Traction
% Rectangular Element Implementation
% 12 elements
% clear memory
clear all;
```

```
clc;
close all;

% magnification factor--------------------------------------------
M = 100;
%-----------------------------------------------------------------

% materials-------------------------------------------------------
E =2e5; poisson = 0.30; thickness = 5;
%-----------------------------------------------------------------

% matrix D for plane stress
D=E/(1-poisson^2)*[1 poisson 0;poisson 1 0;0 0 (1-poisson)/2];

% trivial preprocessing
% numberElements: number of elements------------------------------
numberElements = 12;
%-----------------------------------------------------------------

% numberNodes: number of nodes------------------------------------
numberNodes = 21;
%-----------------------------------------------------------------

% coordinates and connectivities----------------------------------
a=[1 2 9 8];
elementNodes = [a;a+1;a+2;a+3;a+4;a+5;a+7;a+8;a+9;a+10;a+11;a+12];
nodeCoordinates =  [[(0:10:60)';(0:10:60)';(0:10:60)'],[-
10*ones(7,1);0*ones(7,1);10*ones(7,1)]];
figure(1)
drawingMesh(nodeCoordinates,elementNodes,'Q4','k-o');
%-----------------------------------------------------------------

% GDof: global number of degrees of freedom
GDof = 2*numberNodes;

% boundary conditions---------------------------------------------
prescribedDof=[13 14 27 28 41 42]';
%-----------------------------------------------------------------

% force vector----------------------------------------------------
force=zeros(GDof,1);
force(16) = -1000;
%-----------------------------------------------------------------

% calculation of the system stiffness matrix
stiffness =
formStiffnessRec(GDof,numberElements,elementNodes,nodeCoordinates,D,thickness);

% solution
displacements = solution(GDof,prescribedDof,stiffness,force);

% output displacements
outputDisplacements(displacements, numberNodes, GDof);

% reform displacement vector to fit nodeCoordinates---------------
disp = vec2mat(displacements,2)*M;
figure(1)
hold on
drawingMesh(nodeCoordinates+disp,elementNodes,'Q4','r--');
hold on
```

```
p=plot(60,0,'k-o',60,0,'r--
');legend(p,'Undeformed','Deformed','Location','northeast')
%-------------------------------------------------------------------

% exact-------------------------------------------------------------
I = 1/12*5*20^3;
x=0:0.1:60;
u_exact = -1000/E/I*(poisson*x*0^2/2 + x.^3/6 - 60^2*x/2 + 60^3/3);
%-------------------------------------------------------------------

% plot exact--------------------------------------------------------
figure(2)
plot(x,u_exact,'r-',nodeCoordinates(8:14,1),displacements(8*2:2:14*2),'b-
o');grid on
legend('Exact','FEM','Location','southeast')
xlabel('x(mm)');ylabel('displacements(mm)');title('netural axis
displacements(mm)');axis([0 60 -0.12 0.02])
%-------------------------------------------------------------------
```

2. **(40%)** Extend the MATLAB codes to compute the element nodal stresses and average nodal stresses for Problem 1. Average nodal stresses means averaging these element nodal stresses in all elements at a common node to represent the stress at the node. For example, the average nodal

from element 1,2,7 & 8) and then dividing them by 4. Below are sample outputs:

```
Element Nodal Stresses
Element  Node       Sxx             Syy             Sxy
    1       1    -1.6519e+01    -1.7574e+01     4.1970e+00
    1       2    -1.0279e+01     3.2269e+00     8.5226e+00
    1       3     2.0804e+00     6.9346e+00     1.5803e+01
    1       4    -4.1599e+00    -1.3866e+01     1.1477e+01
    2       1    -4.2459e+01    -6.4272e+00     1.9601e+00
    2       2    -4.2038e+01    -5.0241e+00     1.7549e+01
    2       3     2.5013e+00     8.3376e+00     1.8040e+01
    2       4     2.0804e+00     6.9346e+00     2.4512e+00
    3       1    -7.0978e+01    -1.3706e+01    -3.7581e+00
    3       2    -6.9436e+01    -8.5669e+00     2.1959e+01
    3       3     4.0430e+00     1.3477e+01     2.3758e+01
    3       4     2.5013e+00     8.3376e+00    -1.9595e+00
    4       1    -9.8764e+01    -1.7365e+01    -8.5658e+00
    4       2    -9.7780e+01    -1.4083e+01     2.7417e+01
    4       3     5.0278e+00     1.6759e+01     2.8566e+01
    4       4     4.0430e+00     1.3477e+01    -7.4169e+00
    5       1    -1.2814e+02    -2.3190e+01    -1.5073e+01
    5       2    -1.2510e+02    -1.3081e+01     3.1535e+01
    5       3     8.0603e+00     2.6868e+01     3.5073e+01
    5       4     5.0278e+00     1.6759e+01    -1.1535e+01
    6       1    -1.4922e+02    -2.0317e+01    -1.2823e+01
    6       2    -1.5728e+02    -4.7185e+01     4.2226e+01
    6       3    -1.3538e-13    -4.0614e-14     3.2823e+01
    6       4     8.0603e+00     2.6868e+01    -2.2226e+01
    7       1     4.1599e+00     1.3866e+01     1.1477e+01
```

```
     7       2    -2.0804e+00    -6.9346e+00    1.5803e+01
     7       3     1.0279e+01    -3.2269e+00    8.5226e+00
     7       4     1.6519e+01     1.7574e+01    4.1970e+00
     8       1    -2.0804e+00    -6.9346e+00    2.4512e+00
     8       2    -2.5013e+00    -8.3376e+00    1.8040e+01
     8       3     4.2038e+01     5.0241e+00    1.7549e+01
     8       4     4.2459e+01     6.4272e+00    1.9601e+00
     9       1    -2.5013e+00    -8.3376e+00   -1.9595e+00
     9       2    -4.0430e+00    -1.3477e+01    2.3758e+01
     9       3     6.9436e+01     8.5669e+00    2.1959e+01
     9       4     7.0978e+01     1.3706e+01   -3.7581e+00
    10       1    -4.0430e+00    -1.3477e+01   -7.4169e+00
    10       2    -5.0278e+00    -1.6759e+01    2.8566e+01
    10       3     9.7780e+01     1.4083e+01    2.7417e+01
    10       4     9.8764e+01     1.7365e+01   -8.5658e+00
    11       1    -5.0278e+00    -1.6759e+01   -1.1535e+01
    11       2    -8.0603e+00    -2.6868e+01    3.5073e+01
    11       3     1.2510e+02     1.3081e+01    3.1535e+01
    11       4     1.2814e+02     2.3190e+01   -1.5073e+01
    12       1    -8.0603e+00    -2.6868e+01   -2.2226e+01
    12       2    -1.3538e-13    -4.0614e-14    3.2823e+01
    12       3     1.5728e+02     4.7185e+01    4.2226e+01
    12       4     1.4922e+02     2.0317e+01   -1.2823e+01
```

```
  Average Nodal Stresses
  Node        Sxx            Syy            Sxy
    1     -1.6519e+01    -1.7574e+01    4.1970e+00
    2     -2.6369e+01    -1.6001e+00    5.2414e+00
    3     -5.6508e+01    -9.3650e+00    6.8953e+00
    4     -8.4100e+01    -1.2966e+01    6.6968e+00
    5     -1.1296e+02    -1.8636e+01    6.1721e+00
    6     -1.3716e+02    -1.6699e+01    9.3560e+00
    7     -1.5728e+02    -4.7185e+01    4.2226e+01
    8     -7.8604e-14    -3.7836e-13    1.1477e+01
    9     -6.3949e-14    -2.6068e-13    9.1271e+00
   10     -1.2301e-13    -1.2257e-13    8.0402e+00
   11     -1.1680e-13     8.0824e-14    8.1706e+00
   12     -1.2212e-13     3.1974e-14    8.5155e+00
   13     -1.3767e-13    -3.5527e-14    6.4232e+00
   14     -1.3538e-13    -4.0614e-14    3.2823e+01
   15      1.6519e+01     1.7574e+01    4.1970e+00
   16      2.6369e+01     1.6001e+00    5.2414e+00
   17      5.6508e+01     9.3650e+00    6.8953e+00
   18      8.4100e+01     1.2966e+01    6.6968e+00
   19      1.1296e+02     1.8636e+01    6.1721e+00
   20      1.3716e+02     1.6699e+01    9.3560e+00
   21      1.5728e+02     4.7185e+01    4.2226e+01
```

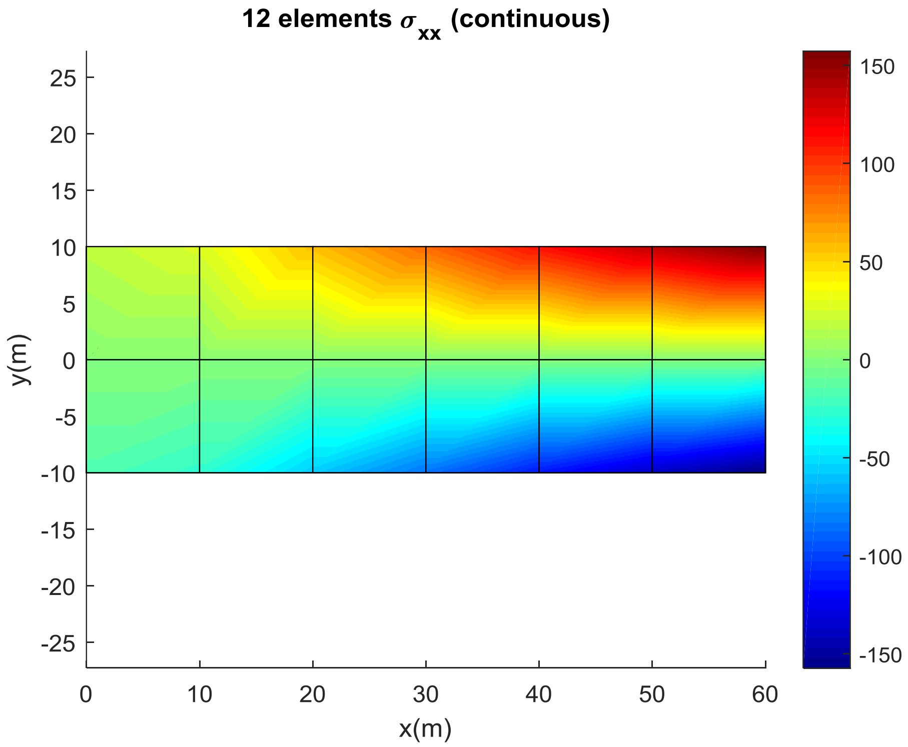
```
% A Thin Plate Subjected to Uniform Traction
% Rectangular Element Implementation
% 12 elements
% clear memory
clear all;
clc;
close all;

% materials-------------------------------------------------------
E =2e5; poisson = 0.30; thickness = 5;
```

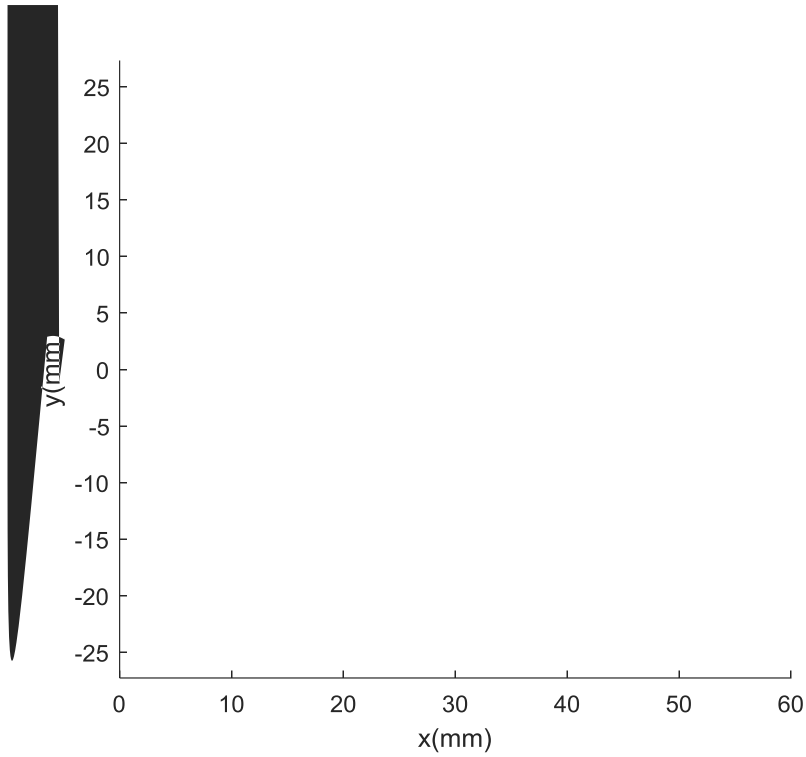
```
%-------------------------------------------------------------------

% matrix D for plane stress
D=E/(1-poisson^2)*[1 poisson 0;poisson 1 0;0 0 (1-poisson)/2];

% trivial preprocessing
% numberElements: number of elements-------------------------------
numberElements = 12;
%-------------------------------------------------------------------

% numberNodes: number of nodes-------------------------------------
numberNodes = 21;
%-------------------------------------------------------------------

% coordinates and connectivities-----------------------------------
a=[1 2 9 8];
elementNodes = [a;a+1;a+2;a+3;a+4;a+5;a+7;a+8;a+9;a+10;a+11;a+12];
nodeCoordinates =  [[(0:10:60)';(0:10:60)';(0:10:60)'],[-
10*ones(7,1);0*ones(7,1);10*ones(7,1)]];
%-------------------------------------------------------------------

% GDof: global number of degrees of freedom
GDof = 2*numberNodes;

% boundary conditions----------------------------------------------
prescribedDof=[13 14 27 28 41 42]';
%-------------------------------------------------------------------

% force vector-----------------------------------------------------
force=zeros(GDof,1);
force(16)  = -1000;
%-------------------------------------------------------------------

% calculation of the system stiffness matrix
stiffness =
formStiffnessRec(GDof,numberElements,elementNodes,nodeCoordinates,D,thickness);

% solution
displacements = solution(GDof,prescribedDof,stiffness,force);

% % output displacements
% outputDisplacements(displacements, numberNodes, GDof);

% stress------------------------------------------------------------
[stress,stress_avg]=formStressRec(GDof,numberElements,elementNodes,...
    nodeCoordinates,D,thickness,numberNodes,displacements);

fprintf('Element Nodal Stresses\n')
fprintf('Element   Node        Sxx         Syy         Sxy\n')
for e=1:numberElements
    for n = 1:4
        fprintf('%2d         %d      %11.4e  %11.4e  %11.4e\n',...
            e,n,stress((e-1)*4+n,1),stress((e-1)*4+n,2),stress((e-1)*4+n,3))
    end
end

fprintf('\nAverage Nodal Stresses\n')
fprintf('Node       Sxx         Syy         Sxy\n')
for i=1:numberNodes
    fprintf('%2d    %11.4e  %11.4e  %11.4e\n',...
```

```
      i,stress_avg(i,1),stress_avg(i,2),stress_avg(i,3))
end
%--------------------------------------------------------------------


function
[stress,stress_avg]=formStressRec(GDof,numberElements,elementNodes,nodeCoordinat
es,D,thickness,numberNodes,displacements)
% compute stiffness matrix
% for plane stress rectangular elements-----------------------------
   stress = zeros(numberElements*4,3);
   stress_avg = zeros(numberNodes,3);
   fraction = zeros(numberNodes,3);
%--------------------------------------------------------------------

for e=1:numberElements
 numEDOF = 8;
 elementDof=zeros(1,numEDOF);
 for i = 1:4
 elementDof(2*i-1)=2*elementNodes(e,i)-1;
 elementDof(2*i)=2*elementNodes(e,i);
 end
 %
 % THIS IS A HACK: we assume node 1 and node 2 align with x-axis and node 2 and
node3 align with y-axis
 %
 a = 0.5*abs(nodeCoordinates(elementNodes(e,2),1) -
nodeCoordinates(elementNodes(e,1),1));
 b = 0.5*abs(nodeCoordinates(elementNodes(e,3),2) -
nodeCoordinates(elementNodes(e,2),2));

 % cycle for nodal point-------------------------------------------
 Locations = [-1 -1;1 -1;1 1;-1 1]; % ¥ª¤U¡B¥k¤U¡B¥k¤W¡B¥ª¤W
 %-------------------------------------------------------------------


 for q=1:size(Locations,1)
 GaussPoint=Locations(q,:);
 xi=GaussPoint(1);
 eta=GaussPoint(2);

% shape functions and derivatives
 [~,naturalDerivatives]=shapeFunctionQ4(xi,eta);
 XYderivatives(:,1) = 1/a * naturalDerivatives(1,:);
 XYderivatives(:,2) = 1/b * naturalDerivatives(2,:);

% B matrix
 B=zeros(3,numEDOF);
 B(1,1:2:numEDOF) = XYderivatives(:,1)';
 B(2,2:2:numEDOF) = XYderivatives(:,2)';
 B(3,1:2:numEDOF) = XYderivatives(:,2)';
 B(3,2:2:numEDOF) = XYderivatives(:,1)';

% stress-nodal
 stress((e-1)*4+q,:) = (D*B*displacements(elementDof))';
 end
% stress-avg-total
 stress_avg(elementNodes(e,:),:) = stress_avg(elementNodes(e,:),:) + stress(4*e-
3:4*e,:);
 fraction(elementNodes(e,:),:) = fraction(elementNodes(e,:),:)+1;
end
% stress-avg
```

```
stress_avg=stress_avg./fraction;
```

3. **(Bonus, 20%; you should finish Problems 1 and 2 first)** Draw discontinuous and continuous σ_{xx} stress contours of Problem 1. When drawing continuous stress contours, using the averaging nodal stresses. When drawing discontinuous stress contours, use the stresses at the element centroid.






```
% A Thin Plate Subjected to Uniform Traction
% Rectangular Element Implementation
% 12 elements
% clear memory
clear all;
clc;
close all;

% materials-------------------------------------------------------------
E =2e5; poisson = 0.30; thickness = 5;
%---------------------------------------------------------------------

% matrix D for plane stress
D=E/(1-poisson^2)*[1 poisson 0;poisson 1 0;0 0 (1-poisson)/2];

% trivial preprocessing
% numberElements: number of elements------------------------------
numberElements = 12;
%---------------------------------------------------------------------

% numberNodes: number of nodes------------------------------------
numberNodes = 21;
%---------------------------------------------------------------------

% coordinates and connectivities----------------------------------
a=[1 2 9 8];
elementNodes = [a;a+1;a+2;a+3;a+4;a+5;a+7;a+8;a+9;a+10;a+11;a+12];
nodeCoordinates =  [[(0:10:60)';(0:10:60)';(0:10:60)'],[-
10*ones(7,1);0*ones(7,1);10*ones(7,1)]];
%---------------------------------------------------------------------

% GDof: global number of degrees of freedom
GDof = 2*numberNodes;

% boundary conditions---------------------------------------------
prescribedDof=[13 14 27 28 41 42]';
```

```
%------------------------------------------------------------------

% force vector-----------------------------------------------------
force=zeros(GDof,1);
force(16) = -1000;
%------------------------------------------------------------------

% calculation of the system stiffness matrix
stiffness =
formStiffnessRec(GDof,numberElements,elementNodes,nodeCoordinates,D,thickness);

% solution
displacements = solution(GDof,prescribedDof,stiffness,force);

% stress-----------------------------------------------------------
[~,stress_avg]=formStressRec(GDof,numberElements,elementNodes,...
    nodeCoordinates,D,thickness,numberNodes,displacements);
stress_cent=formStressRec_central(GDof,numberElements,elementNodes,...
    nodeCoordinates,D,thickness,displacements);

% plot-nodal stress
figure(1)
patch('Faces',elementNodes,'Vertices',nodeCoordinates,'FaceVertexCData',stress_a
vg(:,1),'FaceColor','interp');
title('12 elements \sigma_x_x (continous)');ylabel('y(mm)');xlabel('x(mm)');
colormap(jet)
colorbar
axis equal

% plot-centroid stress
figure(2)
patch('Faces',elementNodes,'Vertices',nodeCoordinates,'FaceVertexCData',stress_c
ent(:,1),'FaceColor','flat');
title('12 elements \sigma_x_x (discontinous)');ylabel('y(mm)');xlabel('x(mm)');
colormap(jet)
colorbar
axis equal
%------------------------------------------------------------------


function
stress_cent=formStressRec_central(GDof,numberElements,elementNodes,nodeCoordinat
es,D,thickness,displacements)
% compute stiffness matrix
% for plane stress rectangular elements
stress_cent=zeros(numberElements,3);
% 2 by 2 quadrature
[gaussWeights,gaussLocations]=gauss2d('1x1');
for e=1:numberElements
 numEDOF = 8;
 elementDof=zeros(1,numEDOF);
 for i = 1:4
 elementDof(2*i-1)=2*elementNodes(e,i)-1;
 elementDof(2*i)=2*elementNodes(e,i);
 end
 %
 % THIS IS A HACK: we assume node 1 and node 2 align with x-axis and node 2 and
node3 align with y-axis
 %
 a = 0.5*abs(nodeCoordinates(elementNodes(e,2),1) -
nodeCoordinates(elementNodes(e,1),1));
```

```
 b = 0.5*abs(nodeCoordinates(elementNodes(e,3),2) -
nodeCoordinates(elementNodes(e,2),2));
 % cycle for Gauss point
 for q=1:size(gaussWeights,1)
 GaussPoint=gaussLocations(q,:);
 xi=GaussPoint(1);
 eta=GaussPoint(2);

% shape functions and derivatives
 [~,naturalDerivatives]=shapeFunctionQ4(xi,eta);
 XYderivatives(:,1) = 1/a * naturalDerivatives(1,:);
 XYderivatives(:,2) = 1/b * naturalDerivatives(2,:);
% B matrix
 B=zeros(3,numEDOF);
 B(1,1:2:numEDOF) = XYderivatives(:,1)';
 B(2,2:2:numEDOF) = XYderivatives(:,2)';
 B(3,1:2:numEDOF) = XYderivatives(:,2)';
 B(3,2:2:numEDOF) = XYderivatives(:,1)';
 end
 stress_cent(e,:) = (D*B*displacements(elementDof))';
end
```