National Taiwan University
Department of Civil Engineering

Finite Element Method
Instructors: C.-S. CHEN

Lab Assignment 6, 05/09/2019, 1800 -- 2000

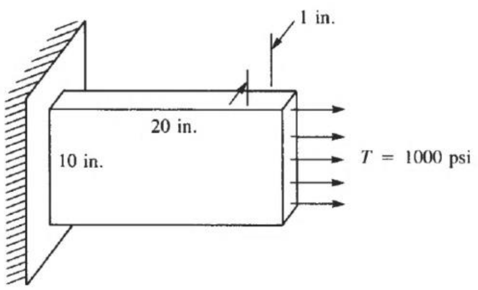
Due 2000

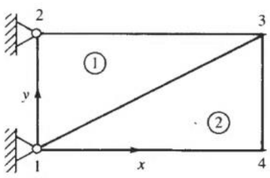
Lab Grading Policy: Attendance 20%, Score 80%, Bonus 20%

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 `T3Simple.zip` from the course website and unzip it. You will find a folder containing a `problem2dTensile.m` file with four functions, `formStiffness2D.m` `solution.m` `outputDisplacements.m`, `drawingMesh.m`.

1. **(40%)** Modify the MATLAB codes to calculate the element stresses and to draw the undeformed and deformed configurations. In drawing the deformed configuration, allow users to specify the magnification factor. Below is a sample run for the thin plate example subjected to the surface traction with two T3 elements ($E = 30Mpsi$ and $\nu = 0.3$):

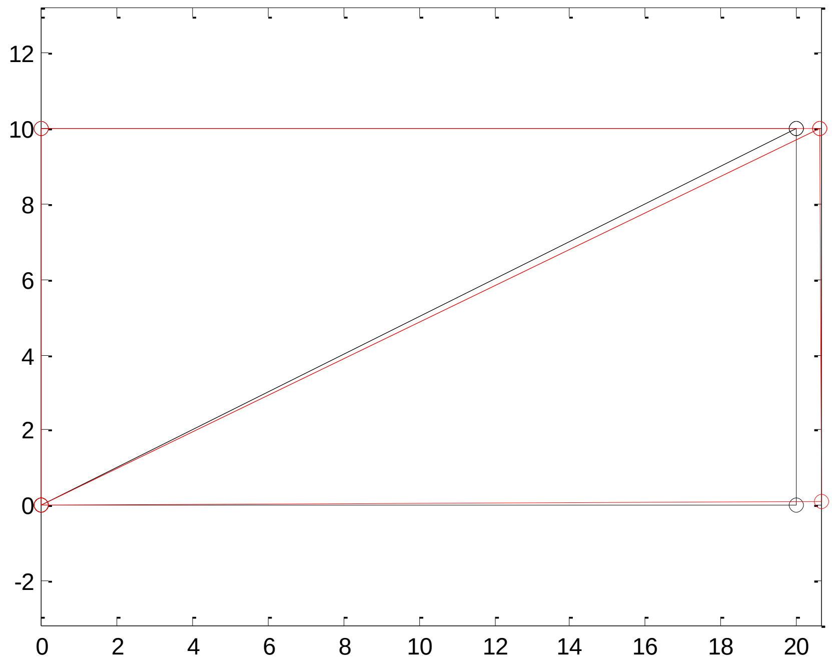




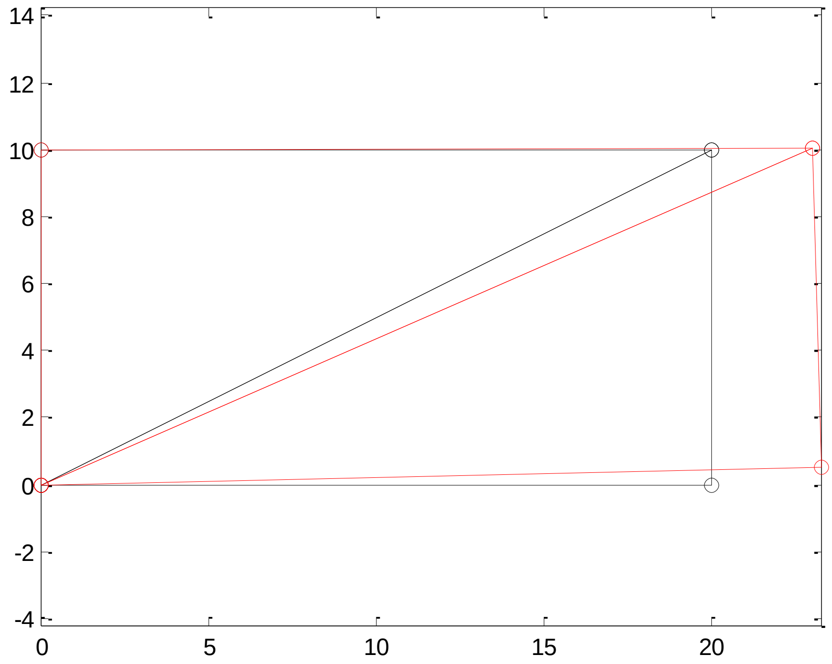

```
 Displacements
Node          UX           UY
   1   0.0000e+00   0.0000e+00
   2   0.0000e+00   0.0000e+00
   3   6.0958e-04   4.1633e-06
   4   6.6370e-04   1.0408e-04

Stress in element 1
Sigma_xx : 1004.803843
Sigma_yy : 301.441153
Sigma_xy : 2.401922


Stress in element 2
```

```
Sigma_xx : 995.196157
Sigma_yy : -1.200961
Sigma_xy : -2.401922
```

Magnification Factor = 1000



Magnification Factor = 5000

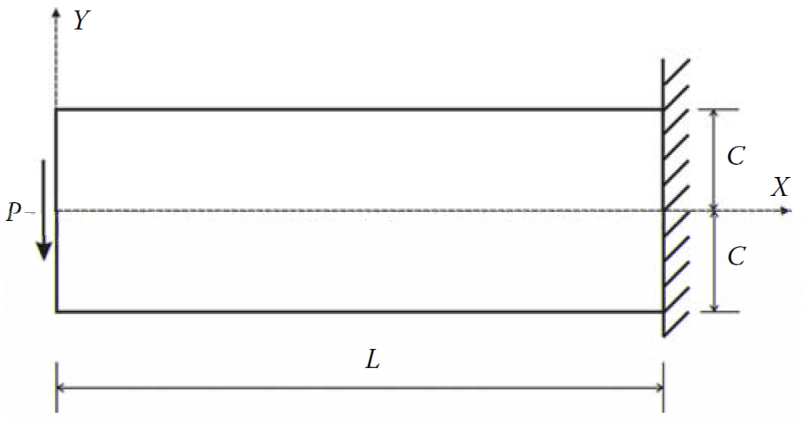


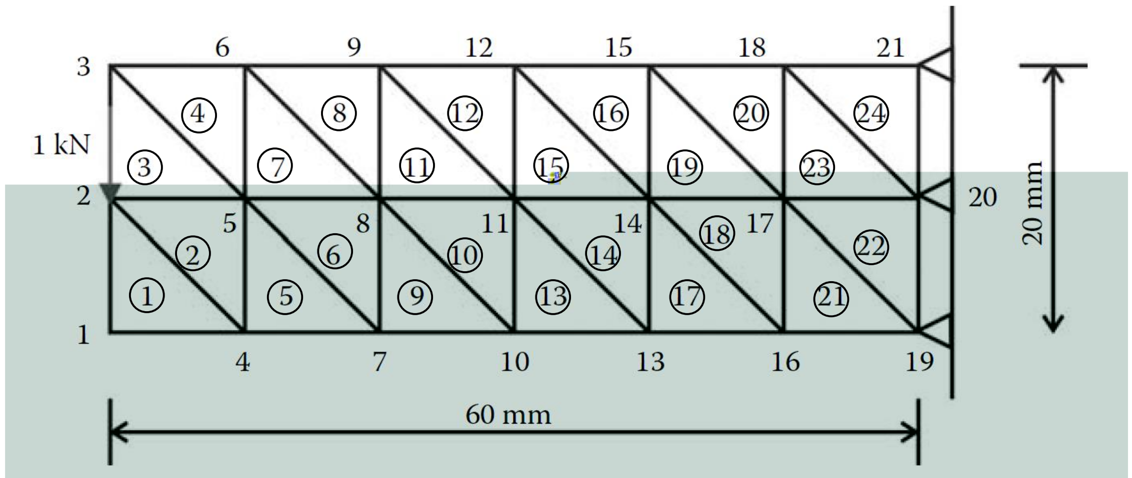

```
% A Thin Plate Subjected to Uniform Traction
% T3 Implementation
% 2 elements
```

```
% clear memory
clear all;
clc;
close all;
%-----------------------------------------------------------------------------
MagnificationFactor=[1000 5000];
%-----------------------------------------------------------------------------

% materials-------------------------------------------------------------------
E  = 30e6;    poisson = 0.30;  thickness = 1;
%-----------------------------------------------------------------------------

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

% trivial preprocessing-
% numberElements: number of elements-----------------------------------------
numberElements = 2;
%-----------------------------------------------------------------------------

% numberNodes: number of nodes-----------------------------------------------
numberNodes = 4;
%-----------------------------------------------------------------------------

% coordinates and connectivities---------------------------------------------
elementNodes=[1 3 2; 1 4 3];
nodeCoordinates=[0, 0; 0, 10; 20, 10; 20, 0];
%-----------------------------------------------------------------------------

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


% boundary conditions--------------------------------------------------------
prescribedDof = [1 2 3 4]';
%-----------------------------------------------------------------------------

% force vector---------------------------------------------------------------
force = zeros(GDof,1);
force(5)  = 5000;
force(7)  = 5000;
%-----------------------------------------------------------------------------

% calculation of the system stiffness matrix
stiffness = formStiffness2D(GDof,numberElements,...
   elementNodes,numberNodes,nodeCoordinates,D,thickness);

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

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

% draw deformed shape--------------------------------------------------------
for jj = 1:length(MagnificationFactor)
figure(jj)
drawingMesh(nodeCoordinates,elementNodes,'T3','k-o');hold on
deformedcoord = MagnificationFactor(jj)*vec2mat(displacements,2);
drawingMesh(nodeCoordinates+deformedcoord,elementNodes,'T3','r-o');
end
```

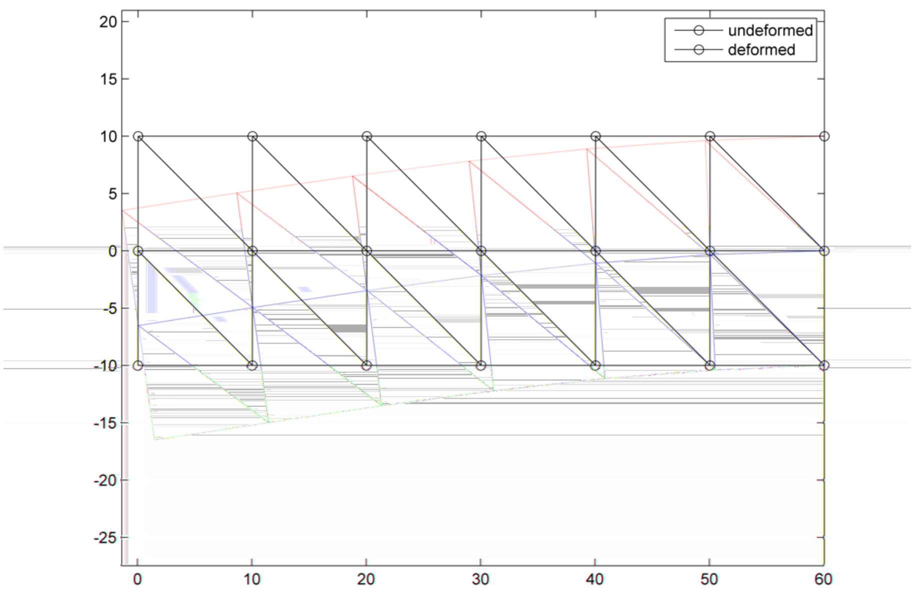
```
%-----------------------------------------------------------------------------

% output stress--------------------------------------------------------------
stress = findStress2D(GDof,numberElements,...
    elementNodes,numberNodes,nodeCoordinates,D,thickness,displacements);
for e=1:numberElements
    fprintf('\nStress in element %d\n',e);
    fprintf('Sigma_xx : %f\n',stress(1,e));
    fprintf('Sigma_yy : %f\n',stress(2,e));
    fprintf('Sigma_xy : %f\n',stress(3,e));
end
%-----------------------------------------------------------------------------


%..................................................................

function stress = findStress2D(GDof,numberElements,...
    elementNodes,numberNodes,nodeCoordinates,D,thickness,displacements)

% compute sstress for T3 elements

stress=zeros(3,numberElements);

for e=1:numberElements
  numNodePerElement = length(elementNodes(e,:));
  numEDOF = 2*numNodePerElement;
  elementDof=zeros(1,numEDOF);
  for i = 1:numNodePerElement
     elementDof(2*i-1)=2*elementNodes(e,i)-1;
     elementDof(2*i)=2*elementNodes(e,i);
  end

  %  B matrix
  x1 = nodeCoordinates(elementNodes(e,1),1);
  y1 = nodeCoordinates(elementNodes(e,1),2);
  x2 = nodeCoordinates(elementNodes(e,2),1);
  y2 = nodeCoordinates(elementNodes(e,2),2);
  x3 = nodeCoordinates(elementNodes(e,3),1);
  y3 = nodeCoordinates(elementNodes(e,3),2);
  A = 1/2*det([1 x1 y1; 1 x2 y2; 1 x3 y3]);
  B = 1/(2*A).*[y2-y3 0 y3-y1 0 y1-y2 0;
                     0 x3-x2 0 x1-x3 0 x2-x1;
                     x3-x2 y2-y3 x1-x3 y3-y1 x2-x1 y1-y2];

  stress(:,e) = D*B*displacements(elementDof);
end
```

2. **(40%)** Consider a cantilever subjected to a concentrated load as 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 Young's modulus of 200,000 MPa and a Poisson's ratio of 0.3 for the material properties, as well as a concentrated force P of 1000 N. We will use 24 elements to discretize the domain as shown in the figure below. The nodes numbered 19, 20, and 21 represent the fixed end. The concentrated force P is applied at node 2.





(a) Plot the undeformed and deformed (magnification factor = 100) configurations. Below is a sample plot:



(b) Report the nodal displacements, reactions and element stresses. Below is a sample output:

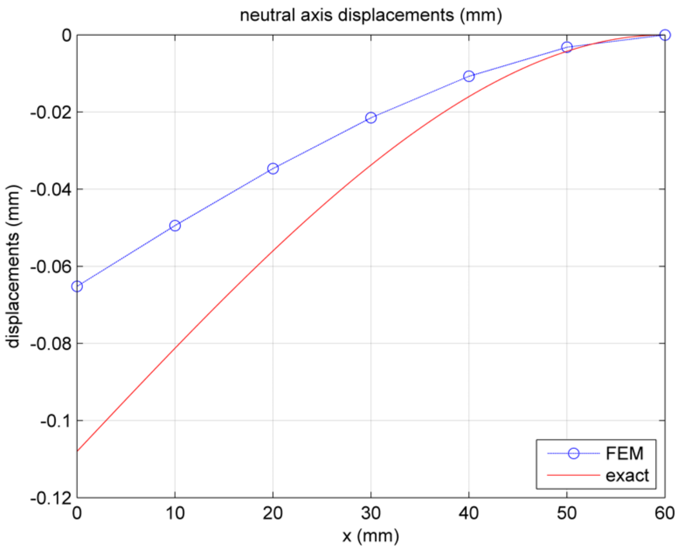
```
Displacements
Node         UX             UY
   1     1.4508e-02     -6.4933e-02
   2     3.2805e-04     -6.5208e-02
   3    -1.4239e-02     -6.4714e-02
   4     1.4233e-02     -4.9732e-02
   5     1.8295e-04     -4.9453e-02
   6    -1.3836e-02     -4.9409e-02
   7     1.2974e-02     -3.5050e-02
   8     1.3798e-04     -3.4663e-02
   9    -1.2672e-02     -3.4756e-02
  10     1.0922e-02     -2.1992e-02
  11     8.9523e-05     -2.1487e-02
  12    -1.0700e-02     -2.1696e-02
  13     8.0808e-03     -1.1348e-02
  14     2.5642e-05     -1.0726e-02
  15    -7.9099e-03     -1.1048e-02
  16     4.4638e-03     -3.8838e-03
  17    -6.6359e-05     -3.1907e-03
  18    -4.2651e-03     -3.6637e-03
  19     0.0000e+00      0.0000e+00
  20     0.0000e+00      0.0000e+00
  21     0.0000e+00      0.0000e+00
Reactions
node         reactions
37:          -2.9520e+03
38:          -1.3524e+02
39:          -9.6009e+01
40:          -2.7235e+02
41:           3.0480e+03
42:           1.4076e+03
Stress
Node         sigma_xx           sigma_yy            tau_xy
   1       -7.8546e+00        -7.8546e+00         7.8546e+00
   2       -1.3515e+00         5.1683e+00         1.3112e+01
   3        6.6118e-02         9.8937e+00         9.1400e+00
   4        9.1400e+00         3.6192e+00         9.8937e+00
   5       -2.5827e+01        -2.1744e+00         4.8607e+00
   6        1.5601e+00         8.1980e+00         1.5027e+01
   7       -6.9913e-01         6.6741e-01         5.9323e+00
   8        2.4966e+01         5.6374e+00         1.4180e+01
   9       -4.2552e+01        -5.0356e+00         1.6983e+00
  10        2.2662e+00         1.0785e+01         1.8024e+01
  11       -1.6757e+00        -2.3552e+00         2.8152e+00
  12        4.1961e+01         8.4119e+00         1.7462e+01
  13       -5.9121e+01        -7.6315e+00        -1.4550e+00
  14        2.6997e+00         1.3258e+01         2.0813e+01
  15       -2.7809e+00        -5.0108e+00        -2.2163e-01
  16        5.9202e+01         1.1322e+01         2.0864e+01
  17       -7.5391e+01        -1.0170e+01        -4.5429e+00
  18        2.5481e+00         1.4627e+01         2.3117e+01
  19       -4.1445e+00        -7.6816e+00        -3.0783e+00
  20        7.6988e+01         1.3636e+01         2.4504e+01
  21       -9.3536e+01        -1.4198e+01        -4.9720e+00
```

22	1.4584e+00	4.3753e-01	2.4544e+01
23	-1.6603e+00	-9.9582e+00	-7.7540e+00
24	9.3738e+01	2.8121e+01	2.8182e+01

(c) Plot 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:

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

Below is a sample plot:



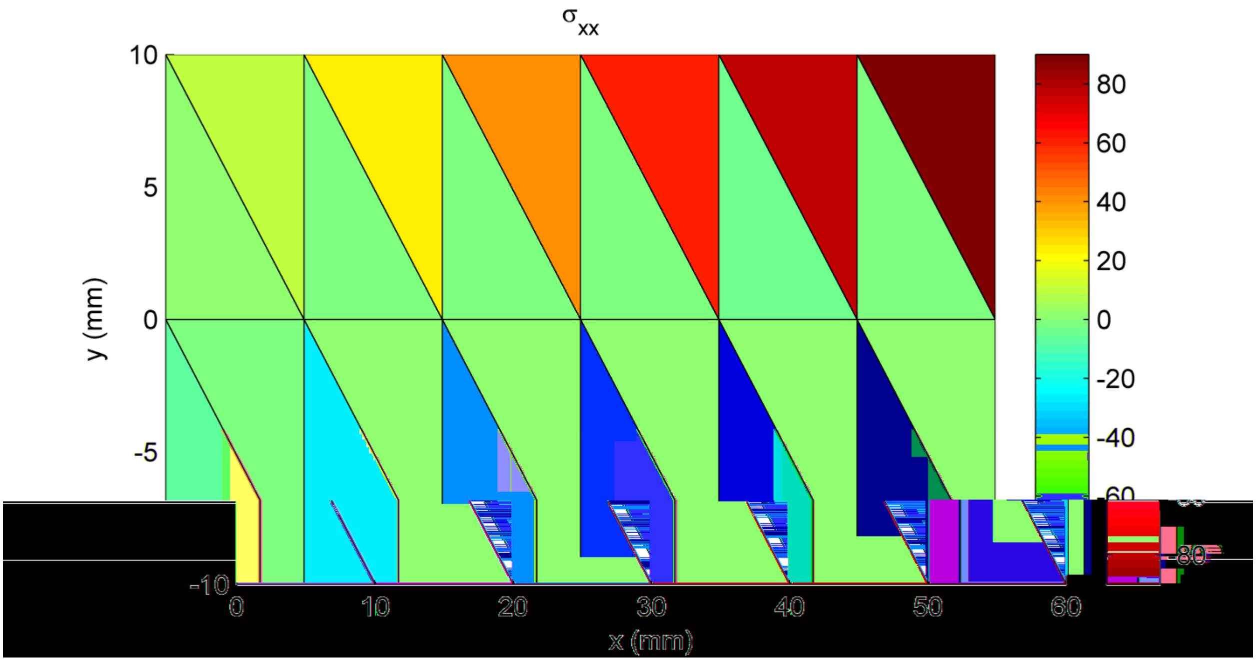

```
% A Thin Plate Subjected to Uniform Traction
% T3 Implementation
% 2 elements
% clear memory
clear all;
clc;
close all;

% MagnificationFactor---------------------------------------------------------
MagnificationFactor=100;
%---------------------------------------------------------------------------

% materials------------------------------------------------------------------
E  = 2*10^5;   poisson = 0.30;  thickness = 5; %N/mm^2
%---------------------------------------------------------------------------

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

% trivial preprocessing
% numberElements: number of elements---------------------------------------
numberElements = 24;
%---------------------------------------------------------------------------
```

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

% coordinates and connectivities--------------------------------------------
a = [1 4 2;2 4 5;2 5 3;3 5 6];
b = [0,-10; 0 0; 0 10];
c = [10*ones(3,1),zeros(3,1)];
elementNodes=[a;a+3;a+6;a+9;a+12;a+15];
nodeCoordinates=[b;b+c;b+c*2;b+c*3;b+c*4;b+c*5;b+c*6];
%--------------------------------------------------------------------------------

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

% boundary conditions----------------------------------------------------------
prescribedDof = [37:42]';
%--------------------------------------------------------------------------------

% force vector---------------------------------------------------------------------
force = zeros(GDof,1);
force(4) = -1000;
%--------------------------------------------------------------------------------

% calculation of the system stiffness matrix
stiffness = formStiffness2D(GDof,numberElements,...
    elementNodes,numberNodes,nodeCoordinates,D,thickness);

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

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

% draw deformed shape--------------(a)--------------------------------------
figure(1)
p1 = drawingMesh(nodeCoordinates,elementNodes,'T3','k-o');hold on
deformedcoord = MagnificationFactor*[displacements(1:2:GDof),
displacements(2:2:GDof)];
p2 = drawingMesh(nodeCoordinates+deformedcoord,elementNodes,'T3','r-o');hold on
legend([p1,p2],'undeformed','deformed','Location','northeast');
%--------------------------------------------------------------------------------

% reaction-------------------------(b)--------------------------------------
outputReactions(displacements,stiffness,GDof,prescribedDof,force)
%--------------------------------------------------------------------------------

% output stress--------------------(b)--------------------------------------
stress = findStress2D(GDof,numberElements,...
    elementNodes,numberNodes,nodeCoordinates,D,thickness,displacements);
fprintf('Stress\n');
fprintf('Element    sigma_xx      sigma_yy        tau_xy  (N/mm^2)\n');
for e=1:numberElements
    fprintf('%2d     %11.4e    %11.4e
%11.4e\n',e,stress(e,1),stress(e,2),stress(e,3));
end
%--------------------------------------------------------------------------------

% exact----------------------------(c)------------------------------------
x = 0:0.1:60;
I = 1/12*5*20^3;
```

```
figure(2)
uy = -1000/E/I*(poisson*0 + x.^3/6 - 60^2*x/2 + 60^3/3);
plot(0:10:60,displacements(4:6:40),'b-o',x,uy,'r-');grid on
legend('FEM','exact','Location','southeast');title('neutral axis displacements
(mm)');
xlabel('x (mm)');ylabel('displacements (mm)')
%-----------------------------------------------------------------------------

%...................................................................

function stress = findStress2D(GDof,numberElements,...
    elementNodes,numberNodes,nodeCoordinates,D,thickness,displacements)

% compute stress for T3 elements

stress=zeros(numberElements,3);

for e=1:numberElements
  numNodePerElement = length(elementNodes(e,:));
  numEDOF = 2*numNodePerElement;
  elementDof=zeros(1,numEDOF);
  for i = 1:numNodePerElement
      elementDof(2*i-1)=2*elementNodes(e,i)-1;
      elementDof(2*i)=2*elementNodes(e,i);
  end

  %  B matrix
  x1 = nodeCoordinates(elementNodes(e,1),1);
  y1 = nodeCoordinates(elementNodes(e,1),2);
  x2 = nodeCoordinates(elementNodes(e,2),1);
  y2 = nodeCoordinates(elementNodes(e,2),2);
  x3 = nodeCoordinates(elementNodes(e,3),1);
  y3 = nodeCoordinates(elementNodes(e,3),2);
  A = 1/2*det([1 x1 y1; 1 x2 y2; 1 x3 y3]);
  B = 1/(2*A).*[y2-y3 0 y3-y1 0 y1-y2 0;
                    0 x3-x2 0 x1-x3 0 x2-x1;
                    x3-x2 y2-y3 x1-x3 y3-y1 x2-x1 y1-y2];

  stress(e,:) = (D*B*displacements(elementDof))';
end


%...................................................................
function p=drawingMesh(nodeCoordinates, elementNodes, type, format)
switch type
    case 'T3'
        seg1 = [1,2,3,1];
    case 'Q4'
        seg1 = [1,2,3,4,1];
    otherwise
        disp('Type is not supported yet')
end

for e = 1:length(elementNodes(:,1))
    p =
plot(nodeCoordinates(elementNodes(e,seg1),1),nodeCoordinates(elementNodes(e,seg1
),2),format)
    hold on
end

axis equal
```

```
hold off
```

3. **(Bonus, 20%) You need to finish Problem 2.** Plot a contour of the longitudinal stress σ_{xx} from Problem 2 using the MATLAB patch function. Below is a sample plot:




```
% A Thin Plate Subjected to Uniform Traction
% T3 Implementation
% 2 elements
% clear memory
clear all;
clc;
close all;

% MagnificationFactor----------------------------------------------------------
MagnificationFactor=100;
%----------------------------------------------------------------------------

% materials------------------------------------------------------------------
E  = 2*10^5;   poisson = 0.30;  thickness = 5; %N/mm^2
%----------------------------------------------------------------------------

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

% trivial preprocessing
% numberElements: number of elements---------------------------------------
numberElements = 24;
%----------------------------------------------------------------------------

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

% coordinates and connectivities------------------------------------------
```

```matlab
a = [1 4 2;2 4 5;2 5 3;3 5 6];
b = [0,-10; 0 0; 0 10];
c = [10*ones(3,1),zeros(3,1)];
elementNodes=[a;a+3;a+6;a+9;a+12;a+15];
nodeCoordinates=[b;b+c;b+c*2;b+c*3;b+c*4;b+c*5;b+c*6];
%--------------------------------------------------------------------------

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

% boundary conditions----------------------------------------------------
prescribedDof = [37:42]';
%--------------------------------------------------------------------------

% force vector-----------------------------------------------------------
force = zeros(GDof,1);
force(4) = -1000;
%--------------------------------------------------------------------------

% calculation of the system stiffness matrix
stiffness = formStiffness2D(GDof,numberElements,...
    elementNodes,numberNodes,nodeCoordinates,D,thickness);

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

% output stress----------------------------------------------------------
stress = findStress2D(GDof,numberElements,...
    elementNodes,numberNodes,nodeCoordinates,D,thickness,displacements);
%--------------------------------------------------------------------------

% draw stress_xx contour-------------------------------------------------
figure(1)
patch('Faces',elementNodes,'Vertices',nodeCoordinates,'FaceVertexCData',stress(:
,1),'FaceColor','flat');
title('\sigma_x_x');ylabel('y(mm)');xlabel('x(mm)');
colormap(jet)
colorbar
hold off
```