

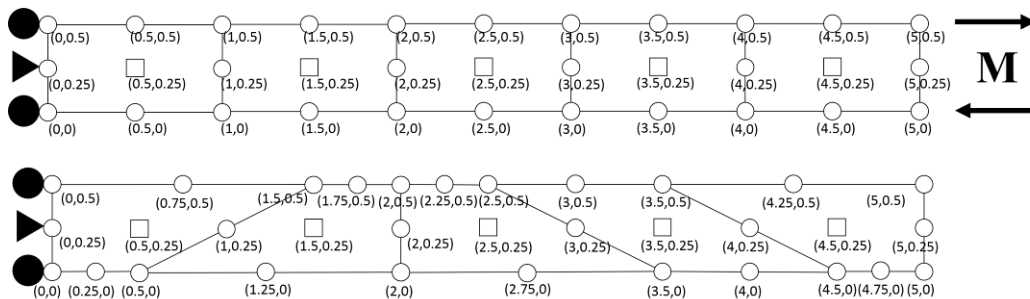
Lab Assignment 10, 06/06/2019, 1800 -- 2000

**Due 2000**

**Lab Grading Policy: Attendance 20%, Score 80%, Bonus 40%**

In case you have difficulty in finishing the exercises on time, you should upload them before **2100 on Saturday** and a penalty of 20% discount will be applied on your score. No late submission after 2100 on Saturday is permitted. We will in general post the reference solutions **by Sunday**.

1. (40%) Reconsider the example in Lab 9. The finite element mesh and element and nodal numbering are shown in the following figure. Use the properties given in Lab 9.



Extend your Q8 2x2 reduced integration implementation with a `stressRecovery` function for stress recovery. You should compute stresses at four Gauss points and extrapolate them to compute the nodal stresses. Finally, averaging these element nodal stresses in all elements at a common node to represent the stress at the node. Plot the element-wise discontinuous stress contour and smooth stress contour from average nodal stresses.

Below are sample outputs and contour plots:

**(please plot two figures, discontinuous contour and smooth stress contour)**

R-Q8R

Stresses at Gauss points

Element	gp	Sxx	Syy	Sxy
1	1	-8.3138e+03	7.5861e-10	-4.0338e-09
1	2	-8.3138e+03	-6.9035e-11	-4.1085e-09
1	3	8.3138e+03	1.6380e-10	-4.0268e-09
1	4	8.3138e+03	-8.0215e-10	-4.0968e-09
2	1	-8.3138e+03	-4.9921e-10	-3.9656e-09
2	2	-8.3138e+03	8.1619e-10	-4.3059e-09
2	3	8.3138e+03	7.5112e-11	-3.8401e-09
2	4	8.3138e+03	-3.9794e-11	-4.1330e-09
3	1	-8.3138e+03	7.6747e-10	-2.5363e-09
3	2	-8.3138e+03	-7.7751e-10	-2.7473e-09
3	3	8.3138e+03	-3.8548e-10	-2.6647e-09
3	4	8.3138e+03	1.0099e-10	-2.9723e-09
4	1	-8.3138e+03	-4.0264e-10	-3.5009e-09
4	2	-8.3138e+03	-8.9570e-10	-3.4718e-09
4	3	8.3138e+03	-6.1770e-10	-3.6441e-09
4	4	8.3138e+03	8.6830e-10	-3.3667e-09
5	1	-8.3138e+03	-9.6305e-10	-9.9276e-10
5	2	-8.3138e+03	5.7527e-10	-1.0242e-09
5	3	8.3138e+03	1.5582e-10	-1.0347e-09
5	4	8.3138e+03	-2.1010e-09	-9.5551e-10

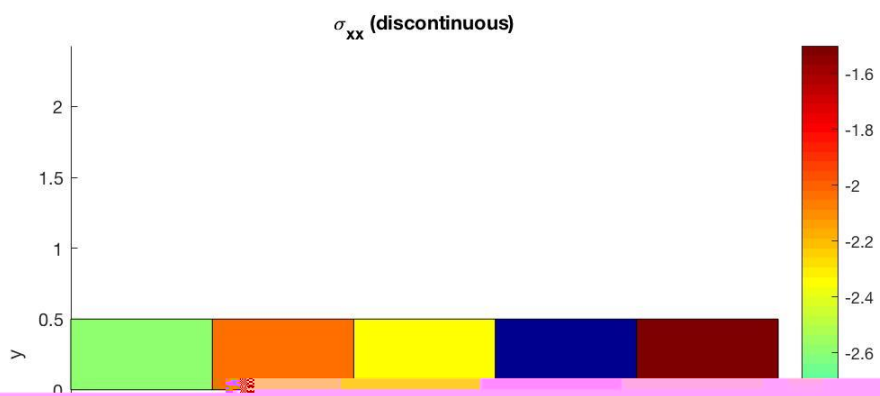
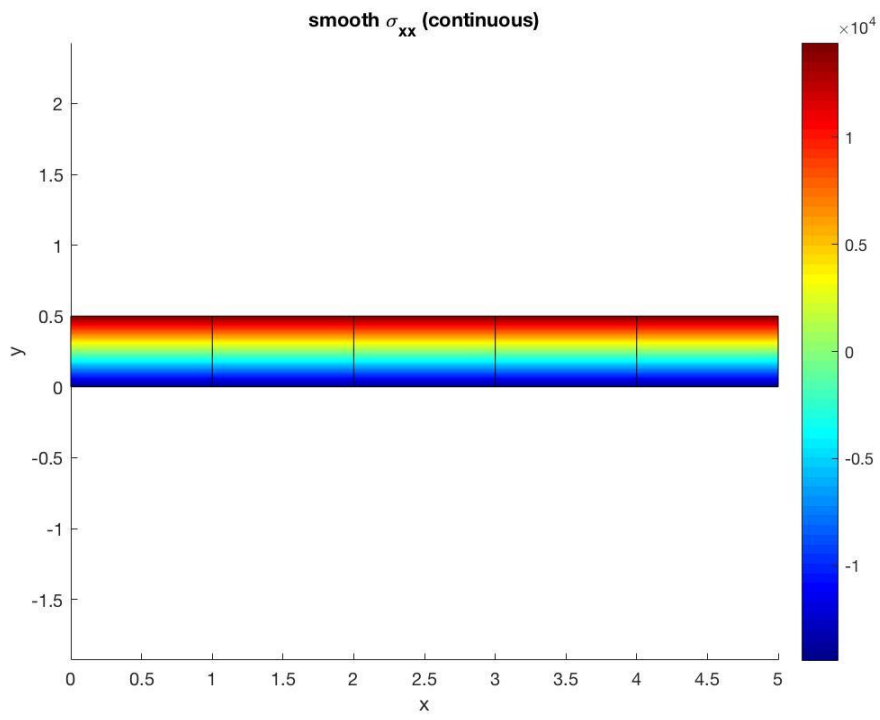
Element nodal Stresses

Element	node	Sxx	Syy	Sxy
1	1	-1.4400e+04	1.8731e-09	-3.9640e-09
1	2	-1.4400e+04	-6.9749e-10	-4.1851e-09
1	3	1.4400e+04	8.4287e-10	-3.9519e-09
1	4	1.4400e+04	-1.9673e-09	-4.1649e-09
1	5	-1.4400e+04	5.8781e-10	-4.0745e-09
1	6	-2.5648e-10	7.2691e-11	-4.0685e-09
1	7	1.4400e+04	-5.6220e-10	-4.0584e-09
1	8	-2.5557e-10	-4.7081e-11	-4.0644e-09
2	1	-1.4400e+04	-1.3097e-09	-3.6950e-09
2	2	-1.4400e+04	1.7298e-09	-4.6858e-09
2	3	1.4400e+04	-3.1492e-10	-3.4776e-09
2	4	1.4400e+04	2.4714e-10	-4.3862e-09
2	5	-1.4400e+04	2.1004e-10	-4.1904e-09
2	6	-1.8554e-10	7.0742e-10	-4.0817e-09
2	7	1.4400e+04	-3.3890e-11	-3.9319e-09
2	8	-2.2555e-10	-5.3127e-10	-4.0406e-09
3	1	-1.4400e+04	1.7187e-09	-2.2300e-09
3	2	-1.4400e+04	-1.6283e-09	-2.9243e-09
3	3	1.4400e+04	-2.7823e-10	-2.4523e-09
3	4	1.4400e+04	-1.0671e-10	-3.3140e-09
3	5	-1.4400e+04	4.5201e-11	-2.5772e-09
3	6	-2.7467e-10	-9.5328e-10	-2.6883e-09
3	7	1.4400e+04	-1.9247e-10	-2.8831e-09
3	8	-1.9554e-10	8.0601e-10	-2.7720e-09
4	1	-1.4400e+04	-8.2039e-10	-3.6017e-09
4	2	-1.4400e+04	-1.0449e-09	-3.3570e-09
4	3	1.4400e+04	-1.1929e-09	-3.8497e-09
4	4	1.4400e+04	2.0104e-09	-3.1750e-09
4	5	-1.4400e+04	-9.3265e-10	-3.4794e-09
4	6	-5.1023e-10	-1.1189e-09	-3.6034e-09
4	7	1.4400e+04	4.0878e-10	-3.5124e-09
4	8	-2.3192e-10	5.9502e-10	-3.3884e-09

5	1	-1.4400e+04	-1.0133e-09	-1.0013e-09
5	2	-1.4400e+04	1.1956e-09	-1.0255e-09
5	3	1.4400e+04	9.2461e-10	-1.0739e-09
5	4	1.4400e+04	-3.4398e-09	-9.0650e-10
5	5	-1.4400e+04	9.1135e-11	-1.0134e-09
5	6	-1.9827e-10	1.0601e-09	-1.0497e-09
5	7	1.4400e+04	-1.2576e-09	-9.9019e-10
5	8	-1.0186e-10	-2.2266e-09	-9.5390e-10

Average nodal Stresses

node	Sxx	Syy	Sxy
1	-1.4400e+04	1.8731e-09	-3.9640e-09
2	-1.4400e+04	5.8781e-10	-4.0745e-09
3	-1.4400e+04	-1.0036e-09	-3.9400e-09
4	-1.4400e+04	2.1004e-10	-4.1904e-09
5	-1.4400e+04	1.7242e-09	-3.4579e-09
6	-1.4400e+04	4.5201e-11	-2.5772e-09
7	-1.4400e+04	-1.2244e-09	-3.2630e-09
8	-1.4400e+04	-9.3265e-10	-3.4794e-09
9	-1.4400e+04	-1.0291e-09	-2.1791e-09
10	-1.4400e+04	9.1135e-11	-1.0134e-09
11	-1.4400e+04	1.1956e-09	-1.0255e-09
12	-2.5557e-10	-4.7081e-11	-4.0644e-09
13	-2.4102e-10	-2.2929e-10	-4.0546e-09
14	-1.9054e-10	7.5671e-10	-3.4268e-09
15	-2.5329e-10	-1.7913e-10	-3.0383e-09
16	-3.0604e-10	-1.6727e-09	-2.2786e-09
17	-1.9827e-10	1.0601e-09	-1.0497e-09
18	1.4400e+04	-1.9673e-09	-4.1649e-09
19	1.4400e+04	-5.6220e-10	-4.0584e-09
20	1.4400e+04	5.4501e-10	-4.1691e-09
21	1.4400e+04	-3.3890e-11	-3.9319e-09
22	1.4400e+04	-2.1081e-10	-3.3958e-09
23	1.4400e+04	-1.9247e-10	-2.8831e-09
24	1.4400e+04	8.6611e-10	-2.8137e-09
25	1.4400e+04	4.0878e-10	-3.5124e-09
26	1.4400e+04	-2.3164e-09	-2.3781e-09
27	1.4400e+04	-1.2576e-09	-9.9019e-10
28	1.4400e+04	9.2461e-10	-1.0739e-09



IR-Q8R

Stresses at Gauss points

Element	gp	Sxx	Syy	Sxy
1	1	-8.3559e+03	7.5885e+00	-3.0251e+01
1	2	-8.3301e+03	5.5038e+01	3.3459e+01
1	3	8.3601e+03	-4.2973e+01	-1.4428e+01
1	4	8.3615e+03	2.3843e+01	4.3536e+01
2	1	-8.3557e+03	-4.0524e+01	-2.9785e+01
2	2	-8.2772e+03	-2.5747e+01	-1.9725e+01
2	3	8.3075e+03	1.9541e+02	9.1431e+01
2	4	8.2177e+03	-1.4925e+01	-5.7677e+01
3	1	-8.6145e+03	-9.9354e+01	1.9402e+02
3	2	-7.6650e+03	3.6905e+02	-4.4781e+02
3	3	7.8860e+03	-2.8019e+02	3.0764e+02
3	4	8.5640e+03	-1.8349e+02	-7.4478e+01
4	1	-8.0776e+03	-2.1776e+02	1.5870e+02
4	2	-9.0724e+03	-3.4406e+02	5.3369e+02
4	3	8.0995e+03	1.0083e+02	-4.7236e+01
4	4	8.5609e+03	3.3859e+02	-4.0036e+02
5	1	-7.5717e+03	4.2658e+02	-5.6181e+02
5	2	-8.4156e+03	-6.7916e+01	9.7943e+01
5	3	8.4206e+03	9.5563e+01	-1.0963e+02
5	4	7.9834e+03	-2.6109e+02	3.0079e+02

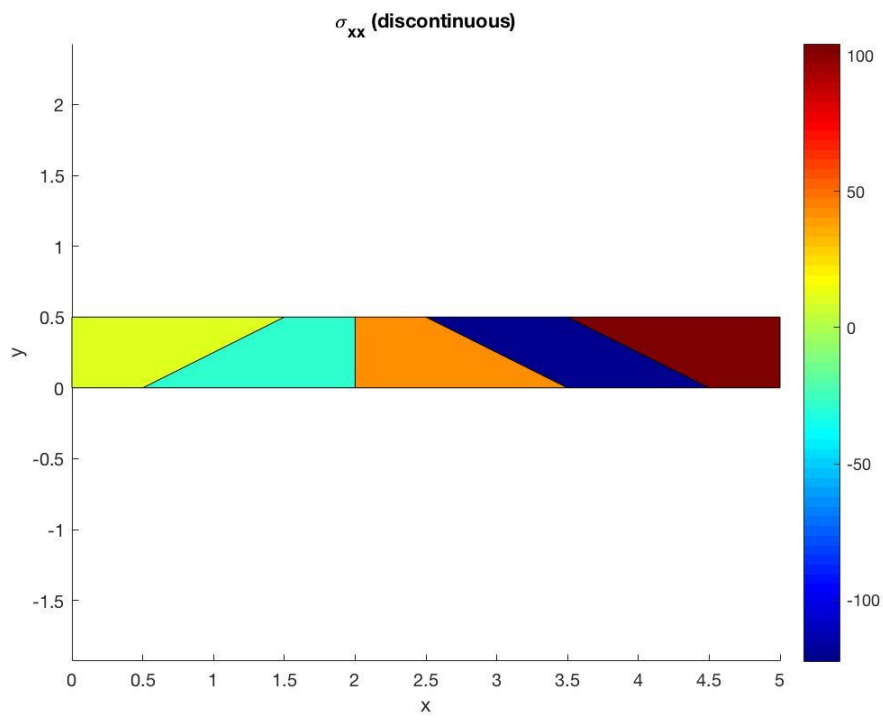
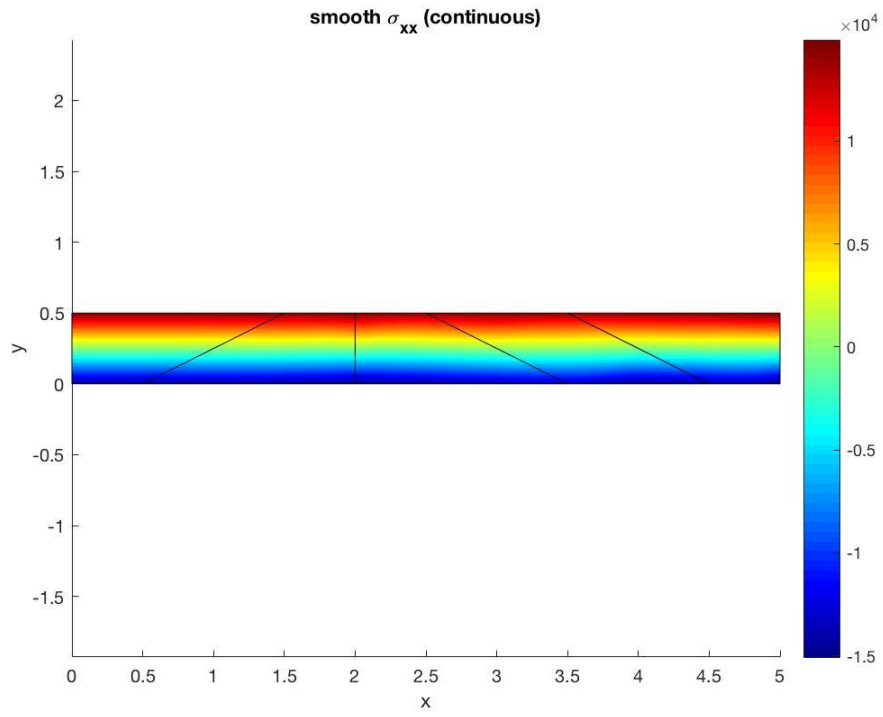
Element nodal Stresses

Element	node	Sxx	Syy	Sxy
1	1	-1.4488e+04	-3.1038e+01	-9.6880e+01
1	2	-1.4426e+04	1.2359e+02	9.0608e+01
1	3	1.4465e+04	-1.1861e+02	-6.9473e+01
1	4	1.4485e+04	6.9558e+01	1.0806e+02
1	5	-1.4457e+04	4.6275e+01	-3.1364e+00
1	6	1.9476e+01	2.4883e+00	1.0567e+01
1	7	1.4475e+04	-2.4527e+01	1.9294e+01
1	8	-1.6589e+00	1.9260e+01	5.5906e+00
2	1	-1.4449e+04	-2.9104e+01	-4.6301e+00
2	2	-1.4320e+04	-1.2748e+02	-7.5357e+01
2	3	1.4412e+04	3.7954e+02	2.0532e+02
2	4	1.4250e+04	-1.0874e+02	-1.4109e+02
2	5	-1.4385e+04	-7.8294e+01	-3.9993e+01
2	6	4.5948e+01	1.2603e+02	6.4983e+01
2	7	1.4331e+04	1.3540e+02	3.2115e+01
2	8	-9.9775e+01	-6.8922e+01	-7.2861e+01
3	1	-1.5468e+04	-3.1571e+02	6.6441e+02
3	2	-1.2791e+04	8.5384e+02	-1.0964e+03
3	3	1.3112e+04	-6.2893e+02	8.6120e+02
3	4	1.5318e+04	-1.0319e+02	-4.4980e+02
3	5	-1.4130e+04	2.6906e+02	-2.1601e+02
3	6	1.6025e+02	1.1246e+02	-1.1761e+02
3	7	1.4215e+04	-3.6606e+02	2.0570e+02
3	8	-7.4972e+01	-2.0945e+02	1.0730e+02
4	1	-1.3732e+04	-3.9011e+02	2.2315e+02
4	2	-1.5793e+04	-5.3819e+02	8.8652e+02
4	3	1.4287e+04	1.6170e+02	-1.3355e+02
4	4	1.4748e+04	6.4419e+02	-7.3131e+02
4	5	-1.4763e+04	-4.6415e+02	5.5483e+02
4	6	-7.5297e+02	-1.8824e+02	3.7648e+02
4	7	1.4518e+04	4.0295e+02	-4.3243e+02

4	8	5.0816e+02	1.2704e+02	-2.5408e+02
5	1	-1.2785e+04	9.7331e+02	-1.2624e+03
5	2	-1.5059e+04	-4.2278e+02	5.5878e+02
5	3	1.4915e+04	3.9998e+02	-4.7920e+02
5	4	1.3345e+04	-7.5738e+02	9.1012e+02
5	5	-1.3922e+04	2.7526e+02	-3.5181e+02
5	6	-7.1978e+01	-1.1402e+01	3.9790e+01
5	7	1.4130e+04	-1.7870e+02	2.1546e+02
5	8	2.8031e+02	1.0797e+02	-1.7614e+02

Average nodal Stresses

node	Sxx	Syy	Sxy
1	-1.4488e+04	-3.1038e+01	-9.6880e+01
2	-1.4457e+04	4.6275e+01	-3.1364e+00
3	-1.4438e+04	4.7242e+01	4.2989e+01
4	-1.4385e+04	-7.8294e+01	-3.9993e+01
5	-1.4894e+04	-2.2160e+02	2.9453e+02
6	-1.4130e+04	2.6906e+02	-2.1601e+02
7	-1.3262e+04	2.3187e+02	-4.3664e+02
8	-1.4763e+04	-4.6415e+02	5.5483e+02
9	-1.4289e+04	2.1756e+02	-1.8795e+02
10	-1.3922e+04	2.7526e+02	-3.5181e+02
11	-1.5059e+04	-4.2278e+02	5.5878e+02
12	-1.6589e+00	1.9260e+01	5.5906e+00
13	-4.0149e+01	-3.3217e+01	-3.1147e+01
14	-1.4512e+01	-4.1713e+01	8.6143e+01
15	3.3421e+02	1.1975e+02	-1.8585e+02
16	-2.3633e+02	-4.0138e+01	1.0017e+02
17	-7.1978e+01	-1.1402e+01	3.9790e+01
18	1.4485e+04	6.9558e+01	1.0806e+02
19	1.4475e+04	-2.4527e+01	1.9294e+01
20	1.4357e+04	-1.1368e+02	-1.0528e+02
21	1.4331e+04	1.3540e+02	3.2115e+01
22	1.4865e+04	1.3817e+02	-1.2224e+02
23	1.4215e+04	-3.6606e+02	2.0570e+02
24	1.3930e+04	7.6306e+00	6.4948e+01
25	1.4518e+04	4.0295e+02	-4.3243e+02
26	1.3816e+04	-2.9784e+02	3.8829e+02
27	1.4130e+04	-1.7870e+02	2.1546e+02
28	1.4915e+04	3.9998e+02	-4.7920e+02



Sol.

```
% clear memory
clear all; close all; clc;

%% initialize
% materials
E = 30000;    poisson = 0.30;  thickness = 1;

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

%% mesh Q8
for m=1:2
numberNodes = 28;
numberElements = 5;
switch m
    case 1
nodeCoordinates = [0,0; 0.5,0; 1,0;1.5,0 ;2,0;2.5,0;3,0;3.5,0;4,0;4.5,0;5,0;
                    0,0.25;1,0.25;2,0.25;3,0.25;4,0.25;5,0.25];

0,0.5;0.5,0.5;1,0.5;1.5,0.5;2,0.5;2.5,0.5;3,0.5;3.5,0.5;4,0.5;4.5,0.5;5,0.5];
    case 2
nodeCoordinates = [0,0; 0.25,0;
0.5,0;1.25,0 ;2,0;2.75,0;3.5,0;4,0;4.5,0;4.75,0;5,0;
                    0,0.25;1,0.25;2,0.25;3,0.25;4,0.25;5,0.25];

0,0.5;0.75,0.5;1.5,0.5;1.75,0.5;2,0.5;2.25,0.5;2.5,0.5;3,0.5;3.5,0.5;4.25,0.5;5,
0.5];
end
elementNodesB = [1,3,20,18,2,13,19,12];
elementNodes=elementNodesB;
for i=1:4
elementNodes=[elementNodes; elementNodesB+2*i];
end
elementNodes(:,[6,8])=[13 12;14 13;15 14;16 15;17 16];
% GDof: global number of degrees of freedom
GDof=2*numberNodes;

% boundary conditions
prescribedDof = [1,23,24,35];

% force vector
force=zeros(GDof,1);
force(21) = -1200;
force(55) = 1200;

% calculation of the system stiffness matrix
stiffness=formStiffness2DQ8(GDof,numberElements,...
    elementNodes,numberNodes,nodeCoordinates,D,thickness);

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

stressRecovery(nodeCoordinates,numberElements,elementNodes,D,displacements)
end
```

```

function
stressRecovery(nodeCoordinates,numberElements,elementNodes,D,displacements)
[a,~]=size(nodeCoordinates);
disContinuousStress=zeros(numberElements,3);
avgStress=zeros(a,3);
avgCount=zeros(a,1);

[gaussWeights,gaussLocations]=gauss2d('2x2');

%% gp stress
fprintf('Stresses at Gauss points\n');
fprintf('Element gp      Sxx      Syy      Sxy\n');
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
    elementStr=zeros(4,3);
    % cycle for Gauss point
    for q=1:size(gaussWeights,1)
        GaussPoint=gaussLocations(q,:);
        xi=GaussPoint(1);
        eta=GaussPoint(2);

% shape functions and derivatives
        [shapeFunction,naturalDerivatives]=shapeFunctionQ8(xi,eta);

% Jacobian matrix, inverse of Jacobian,
% derivatives w.r.t. x,y
        [Jacob,invJacobian,XYderivatives]=...
            Jacobian(nodeCoordinates(elementNodes(e,:),:),naturalDerivatives);

% 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,:);

% stiffness matrix
        stress=D*B*displacements(elementDof);
        elementStr(q,:)=stress;
        fprintf(' %2d %2d %11.4e %11.4e
%11.4e\n',e,q,stress(1),stress(2),stress(3));
    end
end

%% Nodal stress
fprintf('Element nodal Stresses\n');
fprintf('Element node      Sxx      Syy      Sxy\n');
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
end

```

```

end
elementStr=zeros(4,3);
Nstress=zeros(8,3);
% cycle for Gauss point
for q=1:size(gaussWeights,1)
    GaussPoint=gaussLocations(q,:);
    xi=GaussPoint(1);
    eta=GaussPoint(2);

% shape functions and derivatives
    [shapeFunction,naturalDerivatives]=shapeFunctionQ8(xi,eta);

% Jacobian matrix, inverse of Jacobian,
% derivatives w.r.t. x,y
    [Jacob,invJacobian,XYderivatives]=...
        Jacobian(nodeCoordinates(elementNodes(e,:),:),naturalDerivatives);

% 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,:);

% stiffness matrix
    stress=D*B*displacements(elementDof);
    elementStr(q,:)=stress;
end
nd_coor=[-1,-1;1,-1;1,1;-1,1;0,-1;1,0;0,1;-1,0]*sqrt(3);
for j=1:length(nd_coor)
    xi=nd_coor(j,1);
    eta=nd_coor(j,2);
    [shapeFunction,~]=shapeFunctionQ4(xi,eta);
    NodeS=shapeFunction*elementStr;
    Nstress(j,:)=NodeS;
    fprintf(' %2d %2d %11.4e %11.4e\n',e,j,NodeS(1),NodeS(2),NodeS(3));

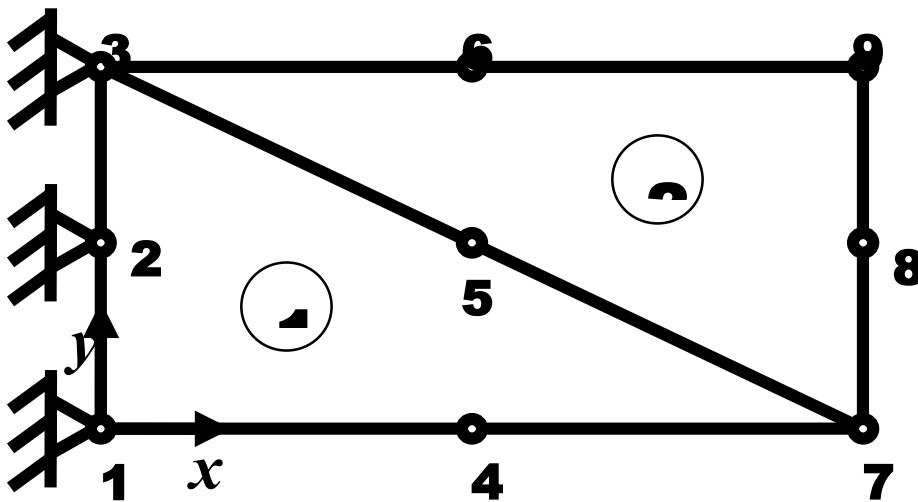
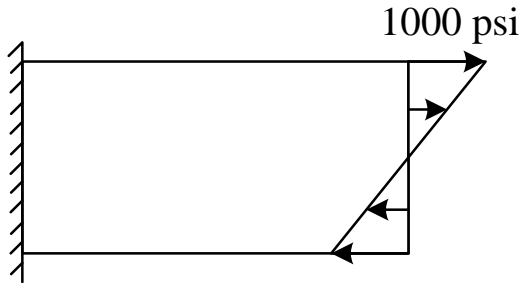
    avgStress(elementNodes(e,j),:)=avgStress(elementNodes(e,j),:)+NodeS;
    avgCount(elementNodes(e,j))=avgCount(elementNodes(e,j))+1;
end
xi=0; eta=0;
[shapeFunction,~]=shapeFunctionQ4(xi,eta);
NodeS=shapeFunction*elementStr;
disContinuousStress(e,:)=NodeS;
end
fprintf('Average nodal Stresses\n');
fprintf(' node Sxx Sy Sxy\n');
for k=1:length(nodeCoordinates)
    avgStress(k,:)=avgStress(k,+)/avgCount(k);
    fprintf(' %2d %11.4e %11.4e %11.4e\n',k,avgStress(k,:))
end
picElementN=[1 2 3 13 20 19 18 12;3 4 5 14 22 21 20 13;5 6 7 15 24 23 22 14; ...
    7 8 9 16 26 25 24 15;9 10 11 17 28 27 26 16];
figure();
patch('Faces',picElementN,'Vertices',nodeCoordinates,'FaceVertexCData',disContinuousStress(:,1),'FaceColor','flat');
colorbar;
colormap(jet)
axis equal;
xlabel('x');
ylabel('y');

```

```
title('\sigma_x_x (discontinuous)');

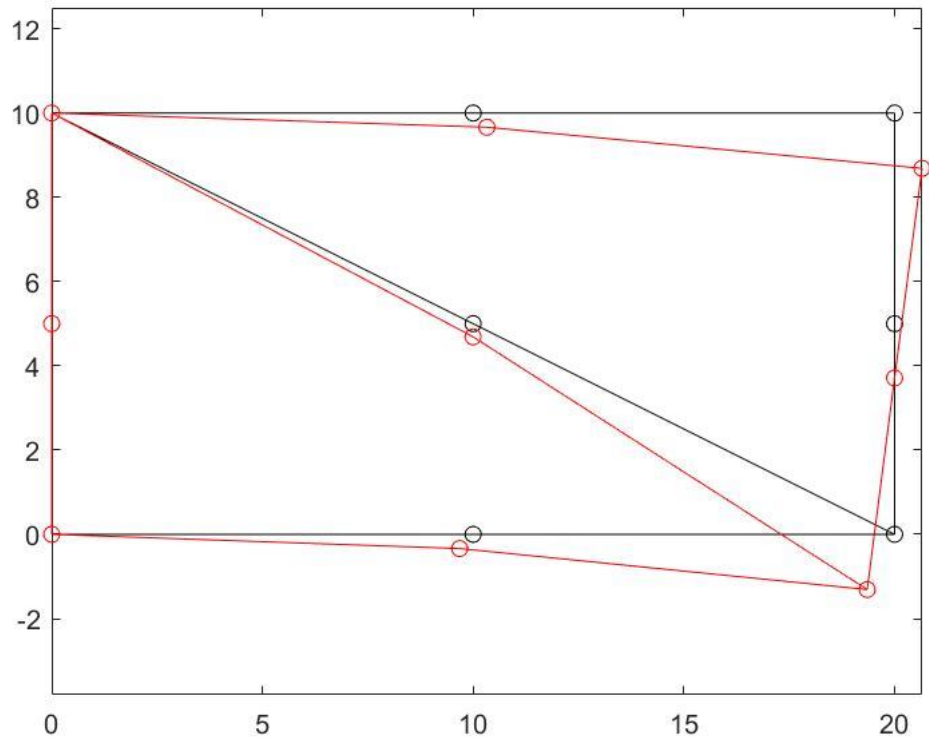
figure();
patch('Faces',picElementN,'Vertices',nodeCoordinates,'FaceVertexCData',avgStress
(:,1),'FaceColor','interp');
colorbar;
colormap(jet)
axis equal;
xlabel('x');
ylabel('y');
title('smooth \sigma_x_x (continuous)');
end
```

2. (40%) Consider a thin plate subjected to a bending moment. Discretize the model with 2 T6 elements shown below. Use the isoparametric formulation and three-point integration. Calculate the external force matrix by hands and specify their values at the corresponding nodes directly. Report the displacements and plot the undeformed and deformed configurations.



Below are the sample output and the sample plot:

Displacements		
Node	UX	UY
1	0.0000e+00	0.0000e+00
2	0.0000e+00	0.0000e+00
3	0.0000e+00	0.0000e+00
4	-3.2109e-04	-3.3718e-04
5	-2.3449e-06	-3.1243e-04
6	3.2553e-04	-3.3665e-04
7	-6.4951e-04	-1.3105e-03
8	-3.9366e-07	-1.2878e-03
9	6.5157e-04	-1.3135e-03



### Sol.

```

% A Thin Plate Subjected to Uniform Traction
% T6 Implementation
% 2 elements
% clear memory
clear all;
clc;
close all;

% 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 = 9;
%-----

% coordinates and connectivities-----
elementNodes=[1 7 3 4 5 2; 3 7 9 5 8 6];
nodeCoordinates=[[0 0 0]', [0 5 10]'; [10 10 10]', [0 5 10]'; [20 20 20]', [0 5
10]'];
drawingMesh(nodeCoordinates,elementNodes, 'T6', 'k-o'); hold on
%-----

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

```

```

% boundary conditions -----
prescribedDof = (1:6)';
%-----

% force vector -----
force = zeros(GDof,1);
syms xi y
[shape , ~] = shapefunc(-1,1,3);
b = -1000 + 200*y; % b(y) in right edge as 1D body force
y2 = nodeCoordinates(end,2);
y1 = nodeCoordinates(end-2,2);
J = (y2-y1)/2;
Y = (y1+y2)/2 + J*xi;
b = subs(b,Y);
shape = subs(shape,xi);
F = double(int(transpose(shape)*b*J,xi,-1,1));
force([13 15 17]) = F;
%-----

% calculation of the system stiffness matrix
stiffness = formStiffness2DT6(GDof,numberElements,...
    elementNodes,numberNodes,nodeCoordinates,D,thickness);

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

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

%-----
m = 1000;
deform = vec2mat(displacements,2)*m;
drawingMesh(nodeCoordinates+deform,elementNodes,'T6','r-o');
%-----

function [shape,naturalDerivatives]=shapeFunctionT6(xi2,xi3)

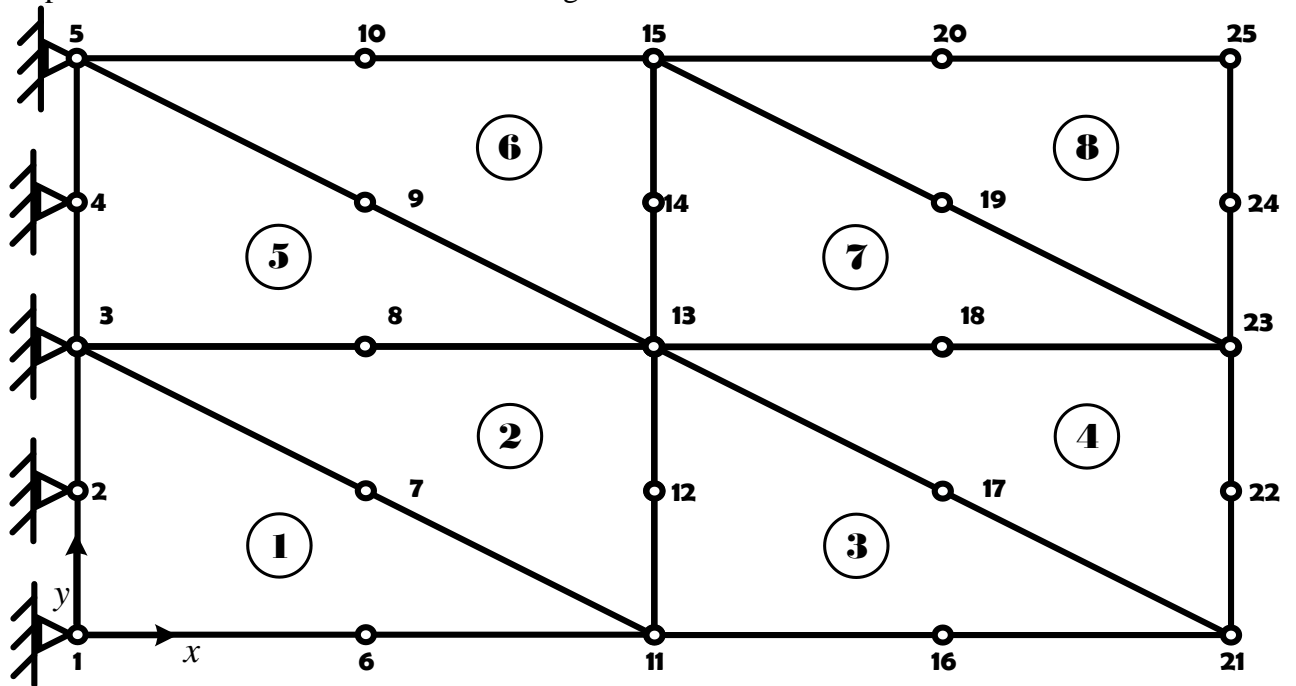
shape=[ (1-xi2-xi3)*(2*(1-xi2-xi3)-1), ...
        xi2*(2*xi2-1), ...
        xi3*(2*xi3-1), ...
        4*(1-xi2-xi3)*xi2, ...
        4*xi2*xi3, ...
        4*xi3*(1-xi2-xi3)];

naturalDerivatives=[4*xi2 + 4*xi3 - 3, ...
                   4*xi2 - 1, ...
                   0, ...
                   4 - 4*xi3 - 8*xi2, ...
                   4*xi3, ...
                   -4*xi3; ...
                   4*xi2 + 4*xi3 - 3, ...
                   0, ...
                   4*xi3 - 1, ...
                   -4*xi2, ...
                   4*xi2, ...
                   4 - 8*xi3 - 4*xi2];

end % end function shapeFunctionQ9

```

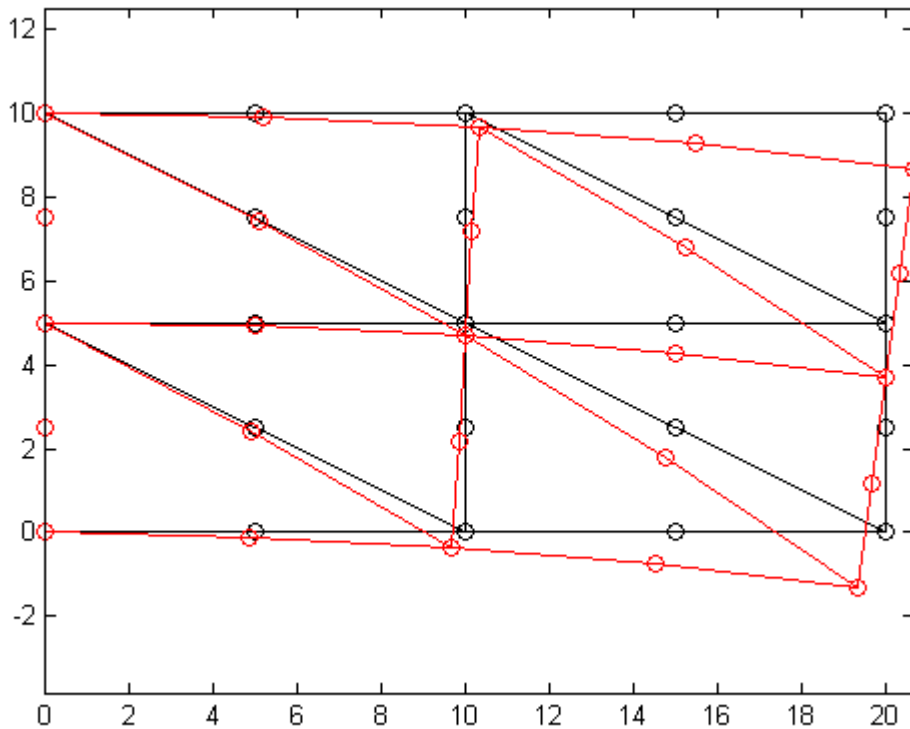
3. **Bonus (20%)**: Consider the same thin plate. Discretize the model with 8 elements shown below. Use the isoparametric formulation and three-point integration. Calculate the external force matrix by hands and specify their values at the corresponding nodes directly. Report the displacements and plot the undeformed and deformed configurations.



Below are the sample output and the sample plot:

Displacements		
Node	UX	UY
1	0.0000e+00	0.0000e+00
2	0.0000e+00	0.0000e+00
3	0.0000e+00	0.0000e+00
4	0.0000e+00	0.0000e+00
5	0.0000e+00	0.0000e+00
6	-1.5986e-04	-9.6298e-05
7	-8.0065e-05	-7.8171e-05
8	-9.0874e-07	-7.2163e-05
9	7.8939e-05	-7.7937e-05
10	1.6386e-04	-9.6280e-05
11	-3.2805e-04	-3.4093e-04
12	-1.6345e-04	-3.2233e-04
13	1.0228e-06	-3.1641e-04
14	1.6350e-04	-3.2341e-04
15	3.2665e-04	-3.4241e-04
16	-4.9396e-04	-7.5224e-04
17	-2.4672e-04	-7.3348e-04
18	4.2573e-07	-7.2713e-04
19	2.4707e-04	-7.3338e-04
20	4.9375e-04	-7.5216e-04
21	-6.6040e-04	-1.3294e-03
22	-3.3013e-04	-1.3107e-03
23	8.1776e-08	-1.3044e-03

24	3.3039e-04	-1.3105e-03
25	6.6067e-04	-1.3293e-03



Sol.

```

% A Thin Plate Subjected to Uniform Traction
% T6 Implementation
% 2 elements
% clear memory
clear all;
clc;
close all;

% 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 = 8;
%-----

% numberNodes: number of nodes-----
numberNodes = 25;
%-----

% coordinates and connectivities-----
a = [1 11 3 6 7 2;3 11 13 7 12 8];
elementNodes=[a;a+10;a+2;a+2+10];
a1 = linspace(0,10,5)';
nodeCoordinates=[0*ones(5,1),a1; 5*ones(5,1),a1; 10*ones(5,1),a1;
15*ones(5,1),a1; 20*ones(5,1),a1];

```

```

drawingMesh(nodeCoordinates,elementNodes,'T6','k-o'); hold on
%-----

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

% boundary conditions -----
prescribedDof = (1:10)';
%-----

% force vector -----
force = zeros(GDof,1);
syms xi y
[shape , ~] = shapefunc(-1,1,3);
for e = [4,8]
    b = -1000 + 200*y; % b(y) in right edge as 1D body force
    y2 = nodeCoordinates(elementNodes(e,3),2);
    y1 = nodeCoordinates(elementNodes(e,2),2);
    J = (y2-y1)/2;
    Y = (y1+y2)/2 + J*xi;
    b = subs(b,y,Y);
    shape = subs(shape,xi);
    F = double(int(transpose(shape)*b*J,xi,-1,1));
    force(elementNodes(e,[2,5,3])*2-1) = force(elementNodes(e,[2,5,3])*2-1) + F;
end
%-----

% calculation of the system stiffness matrix
stiffness = formStiffness2DT6(GDof,numberElements,...
    elementNodes,numberNodes,nodeCoordinates,D,thickness);

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

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

%-----
m = 1000;
deform = vec2mat(displacements,2)*m;
drawingMesh(nodeCoordinates+deform,elementNodes,'T6','r-o');
%-----

function stiffness=formStiffness2DT6(GDof,numberElements,...
    elementNodes,numberNodes,nodeCoordinates,D,thickness)

% compute stiffness matrix
% for plane stress T6 elements

stiffness=zeros(GDof);

% 3 by 3 quadrature-----
[gaussWeights,gaussLocations]=gauss2dTri('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

    % cycle for Gauss point
    for q=1:size(gaussWeights,1)
        GaussPoint=gaussLocations(q,:);
        %-----
        xi1=GaussPoint(1);
        xi2=GaussPoint(2);
        xi3=GaussPoint(3);
        %-----

    % shape functions and derivatives-----
        [shapeFunction,naturalDerivatives]=shapeFunctionT6(xi2,xi3);
    %-----

    % Jacobian matrix, inverse of Jacobian,
    % derivatives w.r.t. x,y
        [Jacob,invJacobian,XYderivatives]=...
            Jacobian(nodeCoordinates(elementNodes(e,:),:),naturalDerivatives);

    % 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,:);

    % stiffness matrix
        stiffness(elementDof,elementDof)=...
            stiffness(elementDof,elementDof)+...
            B'*D*thickness*B*gaussWeights(q)*det(Jacob);
    end
end

```

```

function [weights,locations]=gauss2dTri(option)
% Gauss quadrature in 2D in Triangle element
% option '1'
% option '3'
% option '4'
% option '6'
% locations: Gauss point locations
% weights: Gauss point weights

switch option
case '6'
    alpha1 = 0.8168475730;
    alpha2 = 0.1081030182;
    beta1 = 0.0915762135;
    beta2 = 0.4459484909;
    gamma3 = 0.1099517437/2;
    gamma4 = 0.2233815897/2;
    locations = ...
        [ alpha1 beta1 beta1;
          beta1 alpha1 beta1;
          beta1 beta1 alpha1;
          alpha2 beta2 beta2;
          beta2 alpha2 beta2;

```

```

        beta2 beta2 alpha2];
    weights = [ gamma3; gamma3; gamma3; gamma4; gamma4; gamma4;];

case '4'
    gamma1 = -27/96;
    gamma2 = 25/96;
    locations = ...
    [ 1/3 1/3 1/3;
      0.6 0.2 0.2;
      0.2 0.6 0.2;
      0.2 0.2 0.6];
    weights = [gamma1; gamma2; gamma2; gamma2];

case '3'
    locations=...
    [ 1/2 1/2 0;
      0 1/2 1/2;
      1/2 0 1/2];
    weights=[ 1/6;1/6;1/6];

case '1'
    locations=[1/3 1/3 1/3];
    weights=[1/2];

end

end % end function gauss2dT6

```