HUNAN UNIVERSITY

有限元大作业分析报告

姓　　　名	王钰彬
学 科 专 业	机械工程
学　　　号	S230200254
学　　　院	机械与运载工程学院
上 课 教 师	王琥

一. CST 单元介绍

三节点三角形单元有三个节点和三条边，单元内应力为常数，因此称为常应变三角形单元，即 Constant Stress Triangle Element, 简称 CST 单元。是有限元分析中使用的一种单元，用于在给定微分方程的精确解的二维域中提供近似解。当应用于平面应力和平面应变问题时，这意味着为应力场和应变场获得的近似解在整个单元域中是恒定的。

二. 理论部分

1. 位移场

常应变三角形单元(CST)有三个节点和三个直边，每个节点有两个平动自由度，如图 1，令单元节点位移$\boldsymbol{a} = [u_1\ v_1\ u_2\ v_1\ u_3\ v_3]^T$, 则单元中任一点的位移$[u\ v]$可表示为:

$$\begin{cases} N_1u_1 + N_2u_2 + N_3u_3 \\ N_1v_1 + N_2v_2 + N_3v_3 \end{cases}$$

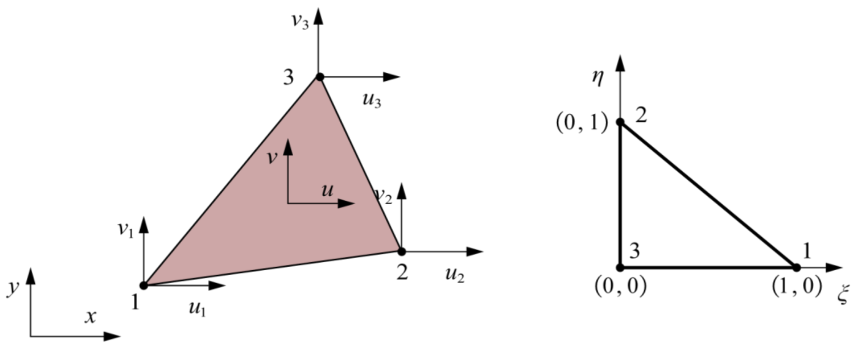


图 1 三节点三角形单元

用矩阵表示为:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{bmatrix} = \boldsymbol{Na}$$

将单元节点按图 1 所示进行编号，并用自然坐标表示，此时单元形函数N_i为

$$\begin{cases} N_1 = \xi \\ N_2 = \eta \\ N_3 = 1 - \xi - \eta \end{cases}$$

其中$\xi \in [0,1]$，$\eta \in [0,1]$，可以看出在单元节点i处，$N_i = 0$

基于单元节点坐标，使相同的形状函数 **N** 对单元内任意一点的几何坐标（x,y）进行插值(等参单元)可以得到单元内任一点的几何坐标表达式:

$$\begin{cases} x = N_1x_1 + N_2x_2 + N_3x_3 \\ y = N_1y_1 + N_2y_2 + N_3y_3 \end{cases}$$

可以求得 CST 单元的雅可比矩阵及其逆矩阵：

$$J = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{3} \frac{\partial N_i}{\partial \xi} x_i & \sum_{i=1}^{3} \frac{\partial N_i}{\partial \xi} y_i \\ \sum_{i=1}^{3} \frac{\partial N_i}{\partial \eta} x_i & \sum_{i=1}^{3} \frac{\partial N_i}{\partial \eta} y_i \end{bmatrix}$$

$$J^{-1} = \frac{1}{\det(J)} \begin{pmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{pmatrix}$$

2．几何方程与应变矩阵

根据几何方程，将单元内任一点的应变ε表示为单元节点位移a的函数，即：

$$\boldsymbol{\varepsilon} = \boldsymbol{Ba}$$

其中，$\boldsymbol{\varepsilon} = [\varepsilon_x \quad \varepsilon_y \quad \gamma_{xy}]^T$，$\boldsymbol{B}$称为应变矩阵，$\boldsymbol{a}$为此前定义的单元节点位移向量

CST 单元的应变矩阵推导过程如下：

$$\varepsilon = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \\ \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} \end{bmatrix} = \frac{1}{\det(J)} \begin{bmatrix} J_{22} & -J_{12} & 0 & 0 \\ 0 & 0 & -J_{21} & J_{11} \\ -J_{21} & J_{11} & J_{22} & -J_{12} \end{bmatrix} \begin{bmatrix} \frac{\partial u}{\partial \xi} \\ \frac{\partial u}{\partial \eta} \\ \frac{\partial v}{\partial \xi} \\ \frac{\partial v}{\partial \eta} \end{bmatrix}$$

$$= \frac{1}{\det(J)} \begin{bmatrix} J_{22} & -J_{12} & 0 & 0 \\ 0 & 0 & -J_{21} & J_{11} \\ -J_{21} & J_{11} & J_{22} & -J_{12} \end{bmatrix} \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & 0 & \frac{\partial N_2}{\partial \xi} & 0 & \frac{\partial N_3}{\partial \xi} & 0 \\ \frac{\partial N_1}{\partial \eta} & 0 & \frac{\partial N_2}{\partial \eta} & 0 & \frac{\partial N_3}{\partial \eta} & 0 \\ 0 & \frac{\partial N_1}{\partial \xi} & 0 & \frac{\partial N_2}{\partial \xi} & 0 & \frac{\partial N_3}{\partial \xi} \\ 0 & \frac{\partial N_1}{\partial \eta} & 0 & \frac{\partial N_2}{\partial \eta} & 0 & \frac{\partial N_3}{\partial \eta} \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{bmatrix}$$

令$\boldsymbol{B} = \boldsymbol{AG}$, 则上式可简写为：

$$\boldsymbol{\varepsilon} = \boldsymbol{AGa}$$

其中，

$$\boldsymbol{a} = \frac{1}{\det(J)} \begin{bmatrix} J_{22} & -J_{12} & 0 & 0 \\ 0 & 0 & -J_{21} & J_{11} \\ -J_{21} & J_{11} & J_{22} & -J_{12} \end{bmatrix}$$

$$\boldsymbol{G}=\begin{bmatrix}\frac{\partial N_1}{\partial \xi} & 0 & \frac{\partial N_2}{\partial \xi} & 0 & \frac{\partial N_3}{\partial \xi} & 0\\ \frac{\partial N_1}{\partial \eta} & 0 & \frac{\partial N_2}{\partial \eta} & 0 & \frac{\partial N_3}{\partial \eta} & 0\\ 0 & \frac{\partial N_1}{\partial \xi} & 0 & \frac{\partial N_2}{\partial \xi} & 0 & \frac{\partial N_3}{\partial \xi}\\ 0 & \frac{\partial N_1}{\partial \eta} & 0 & \frac{\partial N_2}{\partial \eta} & 0 & \frac{\partial N_3}{\partial \eta}\end{bmatrix}$$

对于 CST 单元，矩阵$\boldsymbol{A}$和$\boldsymbol{G}$均为常数矩阵，则应变矩阵$\boldsymbol{B}$也为常数矩阵。

3．单元刚度矩阵

单元的应变能可表示为:

$u=\frac{1}{2}\int_V \ \sigma^T\varepsilon\, dV=\frac{1}{2}a^T[\int_V \ \boldsymbol{B^T DB}\, dV]a=\frac{1}{2}a^T[t\int_0^1\int_0^{1-\eta}\boldsymbol{B^T DB}\det(J)\, d\xi\, d\eta]a$

由最小势能原理可得单元的刚度矩阵k为:

$$k=t\int_0^1\int_0^{1-\eta}\boldsymbol{B^T DB}\det(J)\, d\xi\, d\eta=A_e\boldsymbol{B^T DB}t$$

其中t为单元厚度，$A_e=\frac{1}{2}\det(J)$,等于 CST 单元的面积。对于特定的 CST 单元，式中各项均为常数矩阵，因此可直接计算单元刚度矩阵，无需数值积分。

4．单元载荷列阵及等效节点力

单元的载荷主要包括面力(分布在单元边上的面力)、体力(分布在单元体积上的力)及温度载荷等。以体积力f_v为例，给出其等效节点载荷f的计算方法。

单元的体力表示为$\boldsymbol{f_v}=\left[f_{vx},f_{vy}\right]^T$。单元体力$\boldsymbol{f_v}$在单元变形$\boldsymbol{u}$上做的外力功势能可表示为:

$$W=-\int_V \ u^T\boldsymbol{f_V}\, dV$$

由最小势能原理可得，CST 单元体的等效节点荷载列阵为:

$$\boldsymbol{f}=\int_V \ N^T\boldsymbol{f_V}\, dV=t[\int_0^1\int_0^{1-\eta}N^T\det(J)\, d\xi\, d\eta]\boldsymbol{f_V}$$

积分得到 CST 单元荷载列阵f为:

$$\boldsymbol{f}=\frac{tA_e}{3}\begin{bmatrix}f_V\\ f_V\\ f_V\end{bmatrix}$$

三．问题描述

算例内容如下：是一根xz平面内的悬臂梁，悬臂长度 2.0m，梁高 0.5m，梁

宽 0.2m。梁左端嵌固，右端受到$-z$方向的集中力 $1000kN$。材料弹性模量$E = 200000MPa$，材料泊松比为 0.3，如图 2

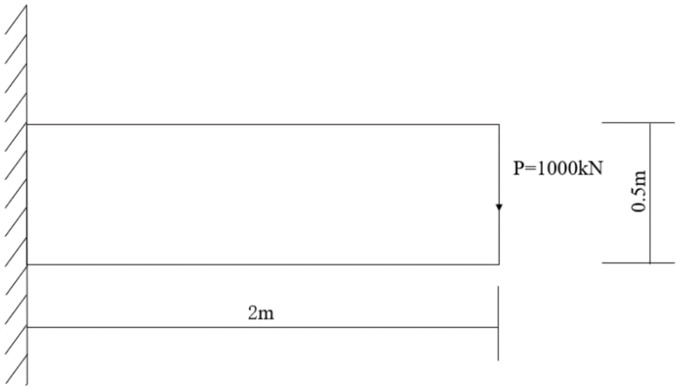


图 2 结构模型示意图

四．算例代码

代码如下：

```
close all
clear
clc
format shortEng
E = 2.000E+08;
v=0.3;
L=2;
H=0.5;
t=0.2;
scalefactor=0.2;
%该段代码定义了若干变量，包括材料的弹性模量 E，材料的泊松比 v，梁长 L、梁高 H、梁厚 t，并为变量赋值，另外定义了一个用于绘图的变量 scalefactor

DoE=0.125;
NEofH=round(4*H/DoE);
NEofL=round(4*L/DoE);
nodeCoord=zeros((NEofH+1)*(NEofL+1),2);
%规划网格划分，此处规划的网格是 16×64=1024 个

for i=1:NEofL+1
    for j=1:NEofH+1
        nodeCoord((i-1)*(NEofH+1)+j,1)=(L/NEofL)*(i-1);
        nodeCoord((i-1)*(NEofH+1)+j,2)=(H/NEofH)*(j-1);
    end
end
nodeCoord
```

```
EleNode=zeros(NEofL*NEofH*2,3);
for i=1:NEofL
    for j=1:NEofH
        EleNode((i-1)*NEofH*2+(2*j-1),1)=(NEofH+1)*(i-1)+1+(j-1);
        EleNode((i-1)*NEofH*2+(2*j-1),2)=(NEofH+1)*i+1+(j-1);
        EleNode((i-1)*NEofH*2+(2*j-1),3)=(NEofH+1)*(i-1)+2+(j-1);
        EleNode((i-1)*NEofH*2+(2*j),1)=(NEofH+1)*(i-1)+2+(j-1);
        EleNode((i-1)*NEofH*2+(2*j),2)=(NEofH+1)*i+1+(j-1);
        EleNode((i-1)*NEofH*2+(2*j),3)=(NEofH+1)*i+2+(j-1);
    end
end
EleNode
```

%本段代码对结构进行有限元网格划分，网格划分完成后，所有的有限元节点的坐标存储于矩阵 nodeCoord 中，所有单元节点编号存储于矩阵 E1eNode 中，本程序共有 1105 个节点，每个节点有 x、y 两个坐标，因此 nodeCoord 为 1105×2 的矩阵；本程序一共有 16×64×2=2048 个三角形单元，每个单元有三个节点，因此 E1eNode 为 2048×3 的矩阵，节点和单元编号顺序为从左到右，从下到上

```
numEle=size(EleNode,1);
numNode=size(nodeCoord,1);
```

%该段代码定义了两个变量 numE1e 和 numNode，分别用于存储结构中的单元数量(即矩阵中的 E1eNMode 行数)和节点数量(即矩阵 nodeCoord 的行数)，因此 numE1e=2048, numNode=1105

```
restrainedDof=zeros(1,2*(NEofH+1));
for i=1:2*(NEofH+1)
    restrainedDof(i)=i;
end
restrainedDof
```

%本算例中，梁左端为嵌固，因此将最左一列节点(节点 1、2、3、4、5、6、7、8、9、10、11、12、13、14、15、16、17）的两个平动自由度进行约束，被约束的自由度编号存储在 restrainedDof 中，矩阵为 1×34 矩阵

```
xx=nodeCoord(:,1);
yy=nodeCoord(:,2);
```

%该段代码定义了两个列向量 xx 和 yy, 分别用于存储 nodeCoord 中所有结点的 x 坐标和所有节点的 y 坐标

```
numDOF=numNode*2;
```

%该段代码定义了一个变量 numDOF，用于存储自由度的总数。本算例共有 numNode=1105 个节点，每个节点有 2 个自由度，因此 numDOF 的值为 1105×2=2210

```
displacement=zeros(numDOF,1);
```

%该段代码定义了一个列向量 displacement, 用于存储整体位移。本例一共有 numDOF=2210 个自由度，令各节点初始位移为 0，故用 zeros(numDof,1)定义了一个 numDOF×1 的零向量，并赋给 displacement

```
force=zeros(numDOF,1);
```

%该段代码定义了一个一个 2210×1 的零向量 force, 用于存储整体节点力

```
P=-1000;
force(numDOF-NEofH)=P;
force
```

%该段代码中另有梁端集中弯矩、梁上均布载荷、梁自重载荷的施加，此处仅介绍两端集中载荷的施加。本算例在悬臂端1/2梁高处施加了-z方向的集中力1000KN，在结构整体自由度矩阵中查得相应的整体自由度编号为(numDOF-NEofH)，施加相应的载荷，其余元素的值仍为0

```
stiffness=zeros(numDOF);
D=(E/(1-v^2))*[1,v,0;v,1,0;0,0,(1-v)/2];
for i=1:numEle
    noindex=EleNode(i,:);
    xy=zeros(3,2);
    for j=1:3
        xy(j,1)=xx(noindex(j));
        xy(j,2)=yy(noindex(j));
    end
    J=CST_J(xy);
    Ae=1/2*det(J);
    A=1/det(J)*[J(2,2),-J(1,2),0,0;0,0,-J(2,1),J(1,1);-J(2,1),J(1,1),J(2,2),-J(1,2)];
    G=[1,0,0,0,-1,0;0,0,1,0,-1,0;0,1,0,0,0,-1;0,0,0,1,0,-1];
    B=A*G;
    eleK=t*Ae*B'*D*B;
    eleDof=[noindex(1)*2-1,noindex(1)*2,noindex(2)*2-1,noindex(2)*2,noindex
(3)*2-1,noindex(3)*2];
    stiffness(eleDof,eleDof)=stiffness(eleDof,eleDof)+eleK;
end
stiffness
```

%本算例一共有 numDOF 个自由度，因此该段代码定义了一个 numDOF×numDOF 的方阵 stiffness, 用于存储结构整体刚度矩阵。该段代码主要包括以下重要步骤：

1. 对所有单元进行遍历，求得各单元的矩阵刚度 eleK。

2.根据各单元自由度与整体自由度的编号对应关系，将各单元刚度矩阵组装成结构整体刚度矩阵

```
activeDof=setdiff([1:numDOF]',restrainedDof);
```

%该段代码定义了一个列向量 activeDof，用于存储处于激活状态的自由度，激活状态的自由度数量为 2210-34=2176 个

```
disp('Displacement')
displacement(activeDof)=stiffness(activeDof,activeDof)\force(activeDof);
```

%通过以上过程，已经求得了结构的整体刚度矩阵stiffness(eleDof,eleDof)、节点力向量force和节点位移向量displacement，该段代码采用“划行划列法”求取未约束自由度的节点位

移，通过用刚度矩阵stiffness(activeDof, activeDof)右除节点力向量force, 得到激活自由度的节点位移结果向量displacement(activeDof)

```
disp_Node=zeros(numNode,3);
for i=1:numNode
    disp_Node(i,:)=[i, displacement(i*2-1:i*2)'];
end
disp_Node
```

%该段代码按节点编号输出各节点的位移，其中第一列为节点编号，第二列，第三列依次是 x 向、y 向平动位移，如图 3：

```
disp_Node =

     1.0000e+000     0.0000e+000     0.0000e+000
     2.0000e+000     0.0000e+000     0.0000e+000
     3.0000e+000     0.0000e+000     0.0000e+000
     4.0000e+000     0.0000e+000     0.0000e+000
     5.0000e+000     0.0000e+000     0.0000e+000
     6.0000e+000     0.0000e+000     0.0000e+000
     7.0000e+000     0.0000e+000     0.0000e+000
     8.0000e+000     0.0000e+000     0.0000e+000
     9.0000e+000     0.0000e+000     0.0000e+000
    10.0000e+000     0.0000e+000     0.0000e+000
    11.0000e+000     0.0000e+000     0.0000e+000
    12.0000e+000     0.0000e+000     0.0000e+000
    13.0000e+000     0.0000e+000     0.0000e+000
    14.0000e+000     0.0000e+000     0.0000e+000
    15.0000e+000     0.0000e+000     0.0000e+000
    16.0000e+000     0.0000e+000     0.0000e+000
    17.0000e+000     0.0000e+000     0.0000e+000
    18.0000e+000   -39.5281e-006   -22.0060e-006
    19.0000e+000   -30.5412e-006   -14.2080e-006
```

图 3 按节点编号输出位移（部分数据） 单位：m

```
stress_Node=zeros(numEle*3,5);
for i=1:numEle
    noindex=EleNode(i,:);
    xy_node=zeros(3,2);
    d=zeros(6,1);
    for j=1:3
        xy_node(j,1)=xx(noindex(j));
        xy_node(j,2)=yy(noindex(j));
        d(2*j-1)=displacement((noindex(j)-1)*2+1);
        d(2*j)=displacement((noindex(j)-1)*2+2);
    end
    J=CST_J(xy_node);
    A=1/det(J)*[J(2,2),-J(1,2),0,0;0,0,-J(2,1),J(1,1);-J(2,1),J(1,1),J(2,2),-J(1,2)];
    G=[1,0,0,0,-1,0;0,0,1,0,-1,0;0,1,0,0,0,-1;0,0,0,1,0,-1];
    B=A*G;
    sigma=D*B*d;
    stress_Node((i-1)*3+1,:)=[i,noindex(1),sigma(1),sigma(2),sigma(3)];
    stress_Node((i-1)*3+2,:)=[i,noindex(2),sigma(1),sigma(2),sigma(3)];
```

```
	stress_Node((i-1)*3+3,:)=[i,noindex(3),sigma(1),sigma(2),sigma(3)];
end
disp('nodal stress')
stress_Node
```

%该段代码用于求解各节点的应力，并存储在 stress_Node 中。其中矩阵第 1 列用于存储单元编号，第 2 列用于存储单元的节点，从第 3 到第 5 列分别用于存储各节点的σ_x，σ_y和τ_y，输出结果如图 4，该段代码另外包含了各单元积分点应力的求解

	1	2	3	4	5	6
1	1	1	-2.7800e+05	-8.3400e+04	-5.4169e+04	
2	1	18	-2.7800e+05	-8.3400e+04	-5.4169e+04	
3	1	2	-2.7800e+05	-8.3400e+04	-5.4169e+04	
4	2	2	-1.9834e+05	-9.5951e+03	-1.2852e+04	
5	2	18	-1.9834e+05	-9.5951e+03	-1.2852e+04	
6	2	19	-1.9834e+05	-9.5951e+03	-1.2852e+04	
7	3	2	-2.1480e+05	-6.4439e+04	-3.4973e+04	
8	3	19	-2.1480e+05	-6.4439e+04	-3.4973e+04	
9	3	3	-2.1480e+05	-6.4439e+04	-3.4973e+04	
10	4	3	-1.6340e+05	-1.8382e+04	-8.7364e+03	
11	4	19	-1.6340e+05	-1.8382e+04	-8.7364e+03	
12	4	20	-1.6340e+05	-1.8382e+04	-8.7364e+03	
13	5	3	-1.7350e+05	-5.2050e+04	-2.3189e+04	
14	5	20	-1.7350e+05	-5.2050e+04	-2.3189e+04	
15	5	4	-1.7350e+05	-5.2050e+04	-2.3189e+04	
16	6	4	-1.3341e+05	-2.0491e+04	-3.8981e+03	
17	6	20	-1.3341e+05	-2.0491e+04	-3.8981e+03	

图 4 节点应力输出（部分数据）

```
for i=1:size(restrainedDof,2)
	rownum=restrainedDof(i);
	reaction(i)=stiffness(rownum,:)*displacement-force(rownum);
end
disp('Reaction')
reaction'
```

%该段代码用于求取支座反力。本算例共有6个支座反力，通过将各自由度对应的整体刚度矩阵stiffness(rownum :)与整体位移向量displacement点乘得到, 支座反力结果输出如图5：

```
>> reaction'

ans =

   1.0380e+003
 429.9016e+000
   1.2311e+003
  90.2001e+000
   1.0160e+003
  61.0547e+000
 830.4362e+000
  29.6221e+000
 656.3756e+000
   5.6058e+000
 488.2844e+000
```

图5 支座反力输出结果（部分数据） 单位：kN

```
maxlen=-1;
```

```
for i=1:numEle
    noindex=EleNode(i,:);
    deltax=xx(noindex(2))-xx(noindex(1));
    deltay=yy(noindex(2))-yy(noindex(1));
    L=sqrt(deltax*deltax+deltay*deltay);
    if(L>maxlen)
        maxlen=L;
    end
end
maxabsDisp=0;
for i=1:numNode
    tempdispU=displacement(i*2-1);
    tempdispV=displacement(i*2);
    tempdisp=sqrt(tempdispU*tempdispU+tempdispV*tempdispV);
    if(tempdisp>maxabsDisp)
        maxabsDisp=tempdisp;
    end
end
factor=0.1;
if(maxabsDisp>1e-30)
    factor=scalefactor*maxlen/maxabsDisp;
end
figure
for i=1:numEle
    noindex1=EleNode(i,:);
    noindex=[noindex1,noindex1(1)];
    Line1=plot(xx([noindex]),yy([noindex]),'--','Color',[0.4,0.4,0.4],'LineWidth',2);
    hold on
end
xxDeformed=xx;
yyDeformed=yy;
for i=1:numNode
    xxDeformed(i)=xxDeformed(i)+factor*displacement(i*2-1);
    yyDeformed(i)=yyDeformed(i)+factor*displacement(i*2);
end
for i=1:numNode
    xxDeformed(i)=xxDeformed(i)+factor*displacement(i*2-1);
    yyDeformed(i)=yyDeformed(i)+factor*displacement(i*2);
end
for i=1:numEle
    noindex=EleNode(i,:);

coordx=[xxDeformed(noindex(1)),xxDeformed(noindex(2)),xxDeformed(noindex(3))
,xxDeformed(noindex(1))];
```

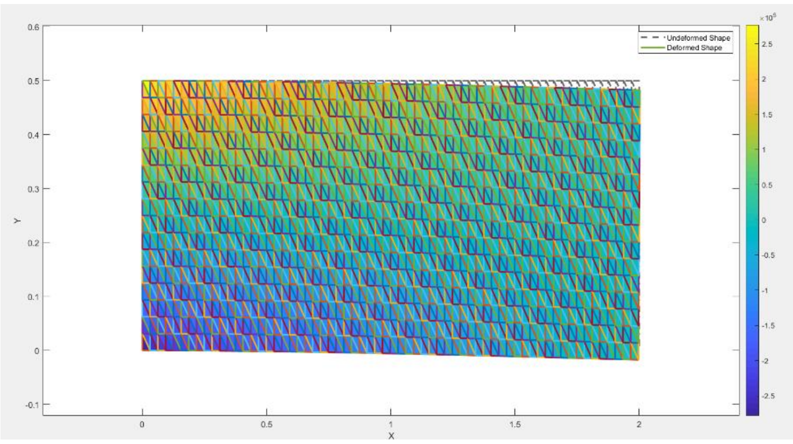
```
coordy=[yyDeformed(noindex(1)),yyDeformed(noindex(2)),yyDeformed(noindex(3))
,yyDeformed(noindex(1))];
        stressX=[stress_Node((i-1)*3+1,3),stress_Node((i-1)*3+2,3),stress_Node((i-
1)*3+3,3),stress_Node((i-1)*3+3,4)];
        fill(coordx,coordy,stressX);
          shading interp;
        colorbar;
end
  for i=1:numEle
          noindex1=EleNode(i,:);
          noindex=[noindex1,noindex1(1)];
          Line2=plot(xxDeformed([noindex]),yyDeformed([noindex]),'-','LineWidth',2);
          hold on
  end
minx=-0.2*(max(xxDeformed)-min(xxDeformed))+min(xxDeformed);
maxx=0.2*(max(xxDeformed)-min(xxDeformed))+max(xxDeformed);
miny=-0.2*(max(yyDeformed)-min(yyDeformed))+min(yyDeformed);
maxy=0.2*(max(yyDeformed)-min(yyDeformed))+max(yyDeformed);
axis([minx,maxx,miny,maxy]);
xlabel('X');
ylabel('Y');
zlabel('Z');
legend([Line1,Line2],'Undeformed Shape','Deformed Shape')
```

%该段代码用于绘制结构的变形图，主要包括以下内容:

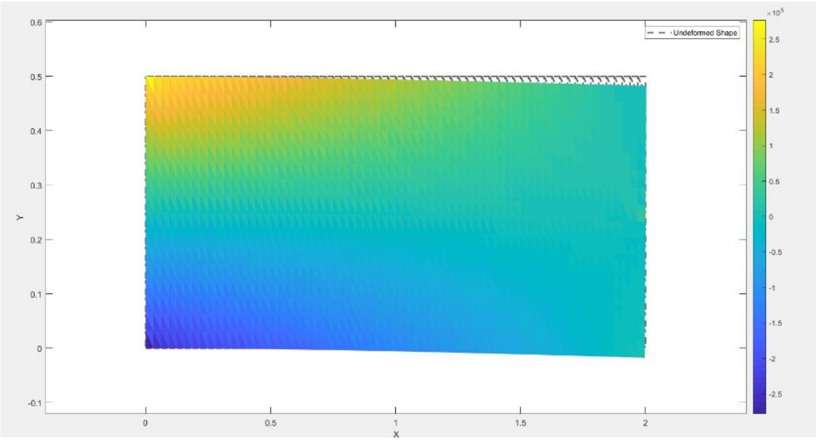
1.绘图参数设置:该段代码用于设置各项绘图参数，主要包括:a,获得各个单元的最大长度的maxlen;b,获得节点位移最大值 maxabsDisp; c,基于上述条件中求得的单元最大长度和节点位移的最大值，计算用于图形绘制的缩放系数 factor。

2.绘制变形图;该段代码用于绘制结构变形图，主要包括以下内容:a,遍历所有单元，根据单元与节点的关系，连点成线，绘制变形之前的结构;b,根据结构变形后的的节点坐标，绘制变形后的结构形状;c,在各节点处绘制相应的节点编号;d,在各单元处绘制相应的单元编号;e,通过设置 x、y 轴显示范围，使得图形以合适的比例进行显示。

3.绘制节点应力:该段代码用于在变形图的基础上绘制节点应力，主要包括以下内容:
绘制节点应力σ_x;绘制节点应力σ_y;绘制节点应力τ_y，绘制的应力图如下：共 2048 个单元



为使图像美观便于查看，去除线条，如下图：



由图可知，左上角区域受到了较大的拉应力，而左下角受到了较大的压应力。