

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

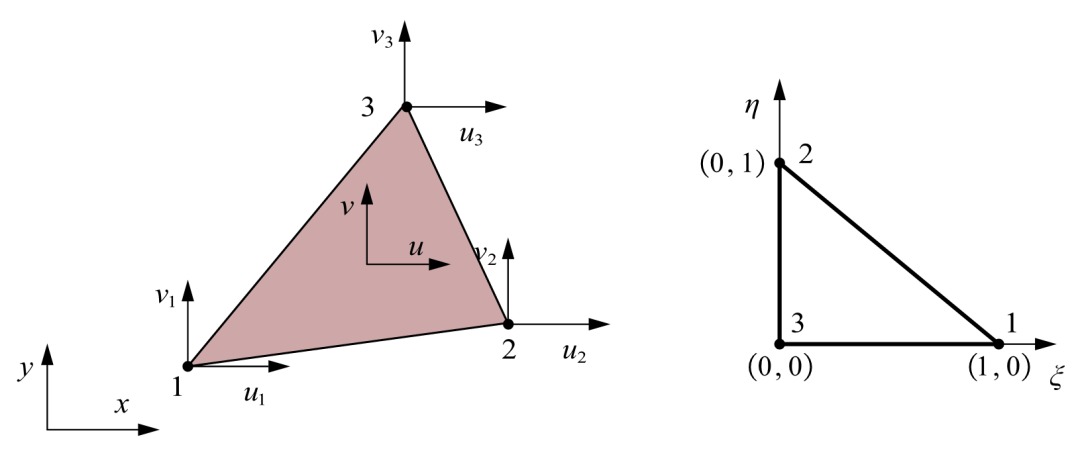
上 课 教 师 王琥

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,则单元中任一点的位移可表示为：

**图1 三节点三角形单元**

用矩阵表示为：

将单元节点按图1所示进行编号，并用自然坐标表示，此时单元形函数为

其中，，可以看出在单元节点处，

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：

可以求得CST单元的雅可比矩阵及其逆矩阵：

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表示为单元节点位移的函数，即：

其中，，称为应变矩阵，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：

令,则上式可简写为：

其中，

对于CST单元，矩阵和均为常数矩阵，则应变矩阵也为常数矩阵。

3．单元刚度矩阵

单元的应变能可表示为:

===

由最小势能原理可得单元的刚度矩阵为：

==

其中为单元厚度，,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

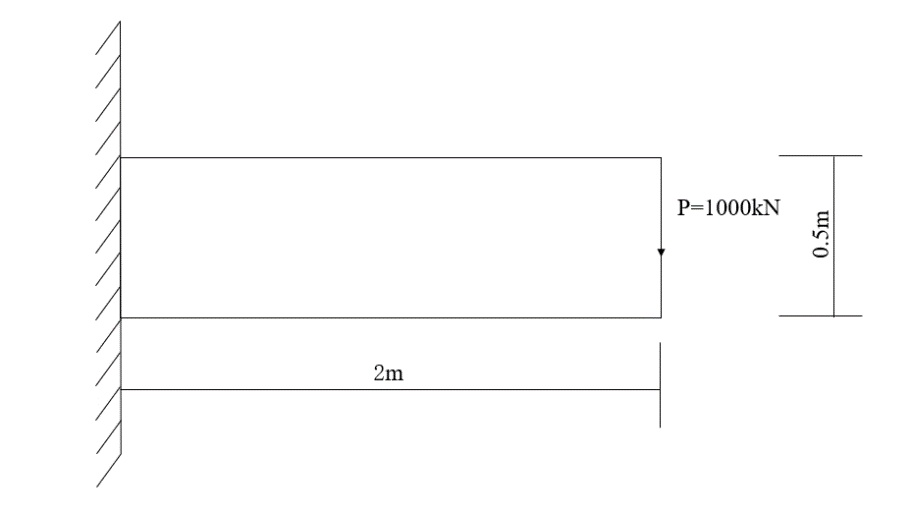
单元的载荷主要包括面力(分布在单元边上的面力)、体力(分布在单元体积上的力)及温度载荷等。以体积力为例，给出其等效节点载荷的计算方法。

单元的体力表示为。单元体力在单元变形上做的外力功势能可表示为：

由最小势能原理可得，CST单元体的等效节点荷载列阵为：

积分得到CST单元荷载列阵为：

三．问题描述

算例内容如下：是一根平面内的悬臂梁，悬臂长度2.0m，梁高0.5m，梁宽0.2m。梁左端嵌固，右端受到方向的集中力1000。材料弹性模量，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、粱高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、粱上均布载荷、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)

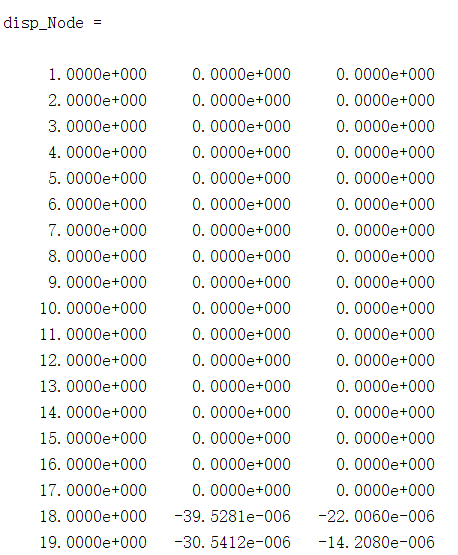
disp\_Node=zeros(numNode,3);

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：

**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)];

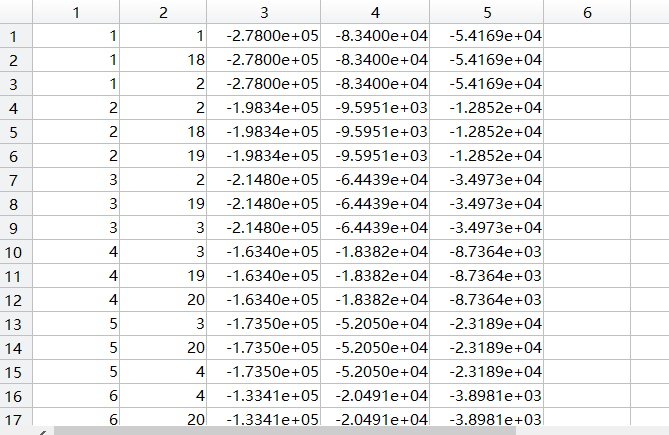
stress\_Node((i-1)\*3+2,:)=[i,noindex(2),sigma(1),sigma(2),sigma(3)];

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, 和,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

**图4 节点应力输出(部分数据)**

for i=1:size(restrainedDof,2)

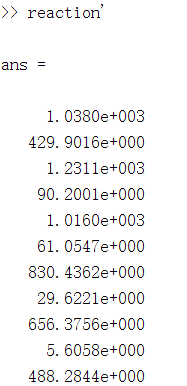
rownum=restrainedDof(i);

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：

**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');

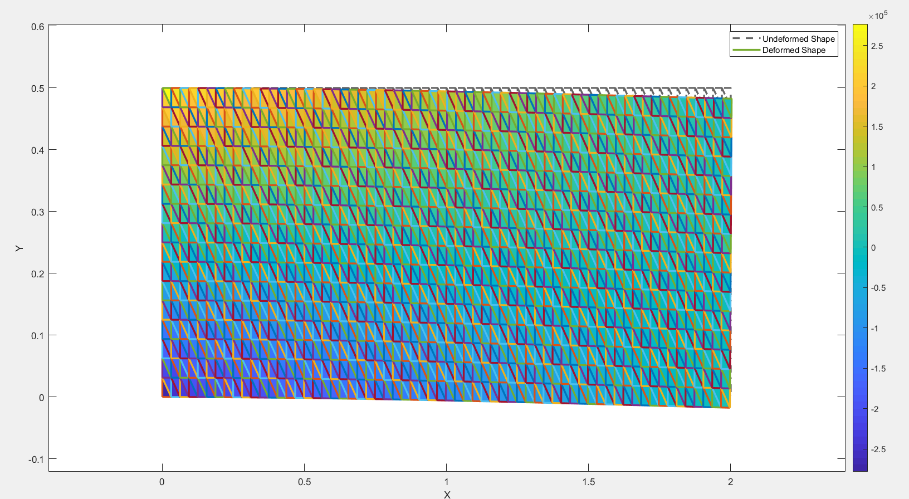
zlabel('Z');

legend([Line1,Line2],'Undeformed Shape','Deformed Shape')

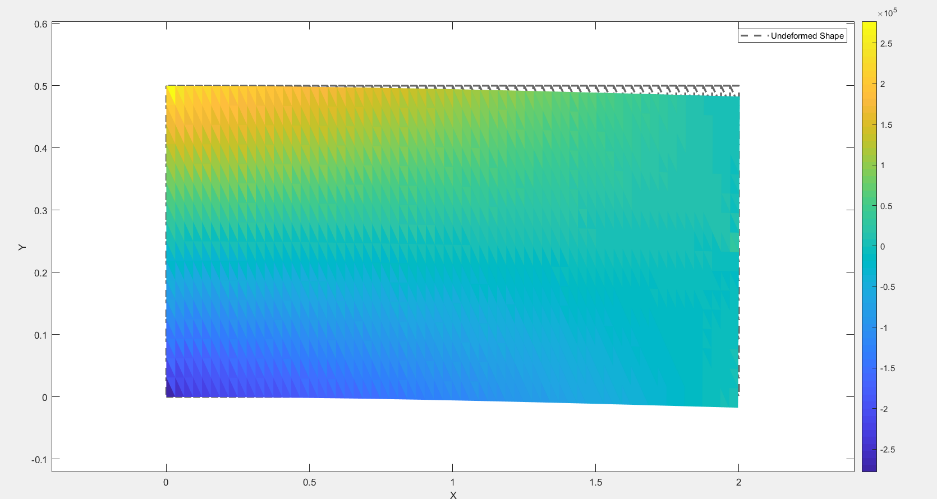
%该段代码用于绘制结构的变形图，主要包括以下内容:

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;绘制节点应力;绘制节点应力，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。