# 有限元方法求解悬臂梁在均布载荷作用下的位移

**班级**：B1701091 **姓名**：张睿谦 **学号**：117010910026

## 问题描述

问题来自《结构有限元分析》一书。

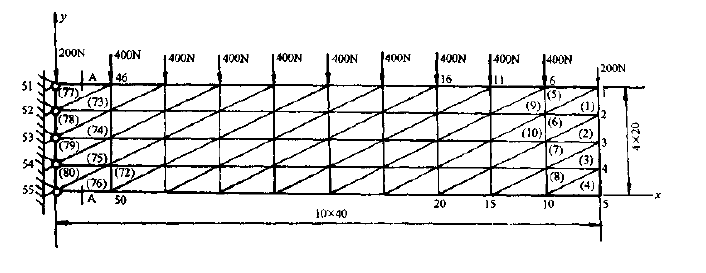


图1.离散化的悬臂梁示意图

如图所示的悬臂梁，受均布载荷的作用。其中悬臂梁的弹性模量，泊松比，厚度。求解悬臂梁的变形，并与理论解进行比较。

## 解决思路

根据有限元分析的一般步骤，可根据以下步骤解决此问题：

1. 将连续体离散化，对节点进行编号，获得节点坐标数据，节点载荷数据；
2. 求得各个单元的刚度矩阵，并将各个单元的刚度矩阵进行叠加，进而获得整体刚度矩阵；
3. 针对根部节点的约束类型对刚度矩阵和载荷矩阵进行处理；
4. 根据，利用列主元高斯消去法求得各节点位移；
5. 将节点位移与节点原坐标相加，绘制变形后图像；
6. 最后与理论计算结果进行比较。

## 算法推导

### 数据准备

首先输入悬臂梁的参数，杨氏模量、泊松比以及梁的厚度。

然后根据网格划分结果，分析单元节点编号矩阵、节点坐标矩阵与载荷矩阵特点。

如图所示划分网格，共有80个三角形三节点单元，55个节点，节点编号如图所示，从第一个节点开始，每五个节点之间y坐标相差20mm，k节点与k+6节点之间x坐标相差40mm。三角形单元内部有以下规律，当右上角节点编号为k时，逆时针方向另外两个节点的编号分别为k+6、k+1或者k+5、k+6。载荷分布如图所示，1和51节点y方向分量为-200N，6:5:46节点y方向分量为-400N。利用上述规律，可以很容易的建立单元节点编号矩阵、节点坐标矩阵与载荷矩阵。

### 三角形刚度矩阵与整体刚度矩阵的求解

刚度矩阵的求解为该算法的关键部分，以下为其推导过程。

在三角形单元中，有三个节点，每个节点分别有、两个方向上的自由度，共有六个自由度，用阵列表示为，



用位移法求单元刚度矩阵时要首先分析单元内部的位移，当单元很小时，可用插值的方式将单元内部位移用节点位移表示出来。

假设三角形单元内部位移、为、的线性函数，并且由于已知的节点位移有6个，所以、可假设为，





表示为矩阵形式为，



记，

将节点坐标带入上式



记为，

所以，，

令，称为形函数矩阵，由此，单元内部位移可由节点位移通过形函数矩阵插值得到。

进而可解得



其中，



为三角形单元的面积，

、、为矩阵的第一列代数余子式，、、和、、分别为第二列和第三列的代数余子式。

得到单元内部位移表达式后，可求得单元内部应变的表达式，



简记为，

其中为应变矩阵，



根据内部应变即可求得应力，



为应力矩阵





根据虚功原理，在任意给出的节点虚位移下，单元节点力做的功与内力做的虚功之和应该等于零，即



又因为，





所以，



为三角形三节点单元的刚度矩阵

根据上式积分可得出具有以下形式，



其中，



单元刚度矩阵可以从上述公式中求得，需要的数据除了杨氏模量、泊松比和梁的厚度外，只有节点的坐标。

利用上式可将划分的所有单元的刚度矩阵求得，将所有单元刚度矩阵叠加在一起，便得到了整体刚度矩阵。

### 约束条件处理

将节点载荷中的未知的约束反例全给为零，并用以大数M乘以中对应的对角线元素，这样求得其节点位移为一极小的值，即可满足约束条件。

根据题目要求，51到55号节点的方向为位移为0，所以将求得的刚度矩阵中的第101行到110行的对角线元素替换为一个大值。将载荷矩阵的第101到110个元素替换为0，即约束反力设为0。

### 列主元消去法

整体刚度矩阵求得后，利用即可求出个节点位移，这里我们求解线性方程组利用列主元消去法。

由高斯消去法可知，在消元过程中，可能会出现的情况，这时消元无法进行，即使主元素但很小时，用其作为除数，也会导致其他元素的数量级的严重增长和舍入误差的扩散，最后也使得计算解不可靠。列主元消去法可很好的解决这一问题。

以下是算法：

设 ，求解。

1. 计算与的秩，当秩相等，且等于的长度时，方程存在唯一解，其余情况无解或无穷多个解。
2. 对于
3. 按列选主元



1. 换行：
2. 消元计算

对于



对于





1. 回代求解



对于



### 绘图

由于节点变形较小，不易观察，可将节点位移放大20倍，建立变形后节点坐标矩阵，连接相邻节点，便可得到变形后图像，并与变形前进行比较。

## 程序框图

开始

离散化，并根据离散化结果建立单元节点编号矩阵、节点坐标矩阵、载荷矩阵

e=1

根据网格参数确定e单元的节点编号及坐标

输入参数，材料的杨氏模量、泊松比和梁的厚度

计算系数a、b、c及delta

将单元刚度矩阵计算结果放入临时整体刚度矩阵[K0]

临时刚度矩阵[K0]←0

整体刚度矩阵[K]←0

[K]=[K]+[K0]

e=e+1

e=80?

处理根部位移约束，修改刚度矩阵[K]和载荷矩阵[Q]

否

是

接下

求解

高斯消去法

将求得放大20倍

将每个节点坐标分别加上，建立新的节点坐标矩阵

绘制变形前与变形后的图像

结束

接上

图2.程序框图

## 结果与讨论

根据程序可得到悬臂梁变形前后的对照图，如下图所示



图3.悬臂梁变形前后对比图

根据材料力学知识，悬臂梁中性层方向位移公式为，



根据此公式可绘制悬臂梁中性层变形图，并与有限元结果相比较。



图4.悬臂梁中性面位移

与理论解对照发现，有限元的解与理论解趋势相同，但结果偏小，这与划分的网格数量有关，当网格数量增加时，有限元解会与理论解更加接近。并且理论解也是在平截面假设和小变形假设下求得的，所以也并非准确解。

## MATLAB程序

列主元高斯消去法程序

function [RA,RB,n,X]=liezhu(A,b)

B=[A b]; n=length(b); RA=rank(A);

RB=rank(B);zhica=RB-RA;

if zhica>0

disp('请注意：因为RA~=RB，所以此方程组无解.')

return

end

if RA==RB

if RA==n

disp('请注意：因为RA=RB=n，所以此方程组有唯一解.')

X=zeros(n,1); C=zeros(1,n+1);

for p= 1:n-1

[Y,j]=max(abs(B(p:n,p))); C=B(p,:);

B(p,:)= B(j+p-1,:); B(j+p-1,:)=C;

for k=p+1:n

m= B(k,p)/ B(p,p);

B(k,p:n+1)= B(k,p:n+1)-m\* B(p,p:n+1);

end

end

b=B(1:n,n+1);A=B(1:n,1:n); X(n)=b(n)/A(n,n);

for q=n-1:-1:1

X(q)=(b(q)-sum(A(q,q+1:n)\*X(q+1:n)))/A(q,q);

end

else

disp('请注意：因为RA=RB<n，所以此方程组有无穷多解.')

end

end

主程序

%数据准备

EE=2.1\*10^5; %弹性模量，N/mm2

uu=0.3; %泊松比

hh=10; %厚度，mm

cx=40; %节点x坐标基准

cy=20; %节点y坐标基准

cj=zeros(55,2); %创建节点坐标矩阵

m=0; %计数器

for i=55:-5:5 %给cj矩阵赋值

k=0;

for j=i:-1:i-4

cj(j,1)=m\*cx;

cj(j,2)=k\*cy;

k=k+1;

end

m=m+1;

end

m=0;k=0;i=0;j=0; %计数器归零

cd=zeros(80,3); %创建单元节点编号矩阵

m=1;

for i=1:8:80 %给cd矩阵赋值

k=m;

for j=i:i+3

cd(j,1)=k;

cd(j+4,1)=k;

cd(j,2)=k+6;

cd(j+4,2)=k+5;

cd(j,3)=k+1;

cd(j+4,3)=k+6;

k=k+1;

end

m=m+5;

end

m=0;k=0;i=0;j=0; %计数器归零

pp(110)=0; %载荷矩阵，N

pp(2)=-200; %载荷矩阵赋值

pp(102)=-200;

for i=12:10:92 %载荷矩阵中间各点赋值

pp(i)=-400;

end

i=0; %计数器归零

cysw=zeros(110,1); %约束位移值

cysj=[51 52 53 53 55]; %约束位移节点

%刚度矩阵计算

kk=zeros(110,110);

l=0;m=0;n=0; %计数器

for i=1:80 %遍历各单元计算单元刚度矩阵，直接赋值进入整体刚度矩阵

k0=zeros(110,110); %创建临时刚度矩阵

clear a b c;

l=cd(i,1);%赋值单元编号

m=cd(i,2);

n=cd(i,3);

delta=0.5\*(cj(l,1)\*cj(m,2)+cj(m,1)\*cj(n,2)+cj(n,1)\*cj(l,2))-0.5\*(cj(m,1)\*cj(l,2)+cj(n,1)\*cj(m,2)+cj(l,1)\*cj(n,2)); %计算参数

a(l)=(cj(m,1)\*cj(n,2)-cj(m,2)\*cj(n,1));

b(l)=-(cj(n,2)-cj(m,2));

c(l)=(cj(n,1)-cj(m,1));

a(m)=-(cj(l,1)\*cj(n,2)-cj(n,1)\*cj(l,2));

b(m)=(cj(n,2)-cj(l,2));

c(m)=-(cj(n,1)-cj(l,1));

a(n)=(cj(l,1)\*cj(m,2)-cj(m,1)\*cj(l,2));

b(n)=-(cj(m,2)-cj(l,2));

c(n)=(cj(m,1)-cj(l,1));

for j=1:3 %r,s=l,m,n

for k=j:3

if j==1

r=l;

else if j==2

r=m;

else r=n;

end

end

if k==1

s=l;

else if k==2

s=m;

else s=n;

end

end

k0(2\*r-1,2\*s-1)=(EE\*hh)\*(b(r)\*b(s)+((1-uu)\*c(r)\*c(s))/2)/(4\*(1-uu^2)\*delta);

k0(2\*s-1,2\*r-1)=k0(2\*r-1,2\*s-1);

k0(2\*r,2\*s-1)=(EE\*hh)\*(uu\*c(r)\*b(s)+((1-uu)\*b(r)\*c(s))/2)/(4\*(1-uu^2)\*delta);

k0(2\*s-1,2\*r)=k0(2\*r,2\*s-1);

k0(2\*r-1,2\*s)=(EE\*hh)\*(uu\*b(r)\*c(s)+((1-uu)\*c(r)\*b(s))/2)/(4\*(1-uu^2)\*delta);

k0(2\*s,2\*r-1)=k0(2\*r-1,2\*s);

k0(2\*r,2\*s)=(EE\*hh)\*(c(r)\*c(s)+((1-uu)\*b(r)\*b(s))/2)/(4\*(1-uu^2)\*delta);

k0(2\*s,2\*r)=k0(2\*r,2\*s);

end

end

kk=kk+k0; %将临时刚阵赋值给整体刚阵

end

%根部位移约束处理

for i=1:length(cysj)

pp(cysj(i)\*2)=0;

pp(cysj(i)\*2-1)=0;

end

for i=101:110

kk(i,i)=10e15;

end

%节点位移求解

[RA,RB,n,dd]=liezhu(kk,pp');

%绘图

size\_huitu=20; %绘图变形放大倍数控制：20倍

for i=1:55 %对节点位移值进行格式调整

dd\_huitu(i,1)=dd(2\*i-1);

dd\_huitu(i,2)=dd(2\*i);

end

cj\_huitu=cj+size\_huitu\*dd\_huitu; %将节点变形前坐标和变形位移叠加得新节点坐标

x=cj\_huitu(:,1); %节点变形后x坐标

y=cj\_huitu(:,2); %节点变形后y坐标

figure(1) %绘变形前后对比图

plot(x,y,'.b') %绘制变形后图形

for i=1:80 %绘制三角形

m1=x(cd(i,1));

m2=y(cd(i,1));

n1=x(cd(i,2));

n2=y(cd(i,2));

p1=x(cd(i,3));

p2=y(cd(i,3));

line([m1,n1],[m2,n2],'color','b')

line([n1,p1],[n2,p2],'color','b')

line([p1,m1],[p2,m2],'color','b')

end

hold on

plot(cj(:,1),cj(:,2),'.r') %绘制变形前图形

for i=1:80 %绘制三角形

m1=cj(cd(i,1),1);

m2=cj(cd(i,1),2);

n1=cj(cd(i,2),1);

n2=cj(cd(i,2),2);

p1=cj(cd(i,3),1);

p2=cj(cd(i,3),2);

line([m1,n1],[m2,n2],'color','r')

line([n1,p1],[n2,p2],'color','r')

line([p1,m1],[p2,m2],'color','r')

end

xlim([-1,500])

ylim([-100,100])

title('悬臂梁变形前后对比图（变形放大20倍）')

hold off

clear x y s r p2 p1 n2 n1 m2 m1 m n l

clear k0 i j k delta c RA RB%清除过程变量