

**研究生课程作业**

课程名称： 有限元方法及应用

研究生姓名： 熊一舸 学号： B230200139

作业成绩：

任课教师(签名)

交作业时间：2024年2月23日

剪贴变形下的空间梁的有限元分析

熊一舸

1 剪贴变形下的空间梁的有限元分析

1.1问题描述

如图1所示，某悬臂梁，有三个矩形开口，梁端承受一集中载荷F，大小为60000N。计算梁端A点的线位移，并和无开口梁的结果进行比较。悬臂梁材料的杨氏模量，泊松比。

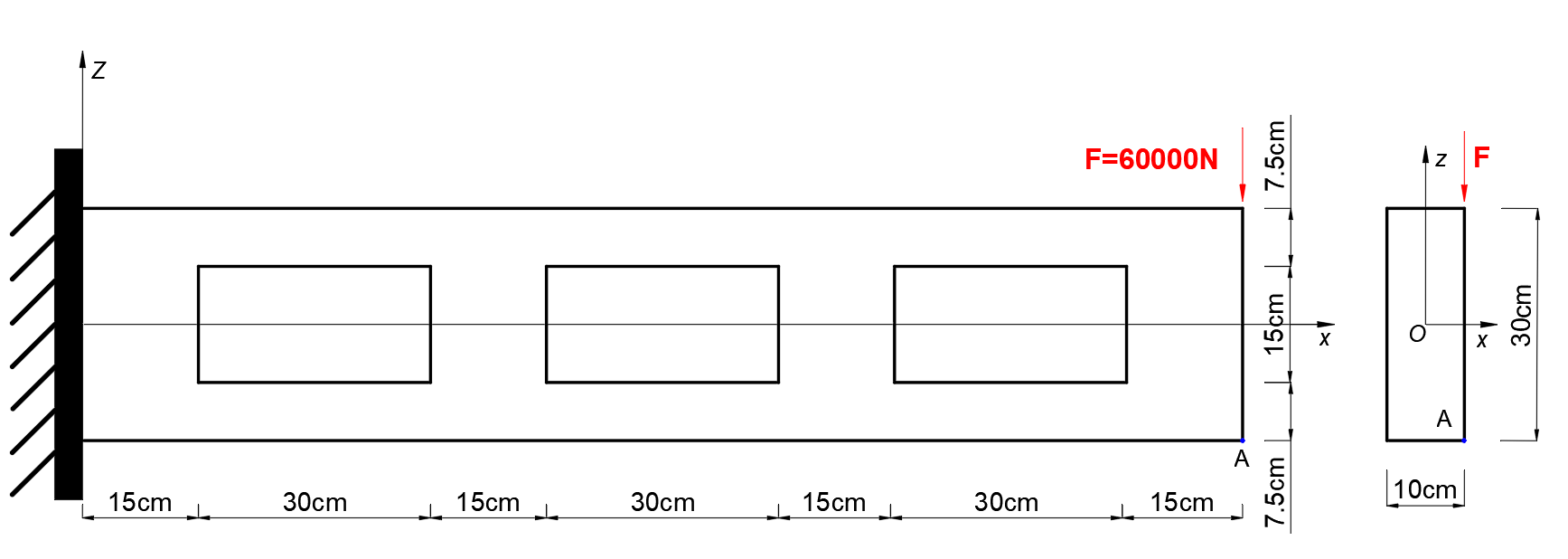


图1 矩形开口悬臂梁结构末端受集中力F作用示意图

1.2理论分析

考虑剪切变形空间梁的势能泛函和梁单元位移插值函数，并通过泛函变分原理推导得到梁单元的刚度方程

首先把空间梁单元分解成一个杆单元、一个轴单元，和xoy，xoz平面内的两个平面梁，由此可以把得到空间梁单元的能量泛函为：

首先给出如下的形函数（插值函数）:

,

对于考虑剪切变形的梁单元将挠度分解为：

则节点位移为：

考虑将采用Lagrange插值，即线性插值；采用与不考虑剪切变形梁单元的相同的Hermite插值，则有:

利用平衡方程可以得到最终平面梁单元的能量泛函为：

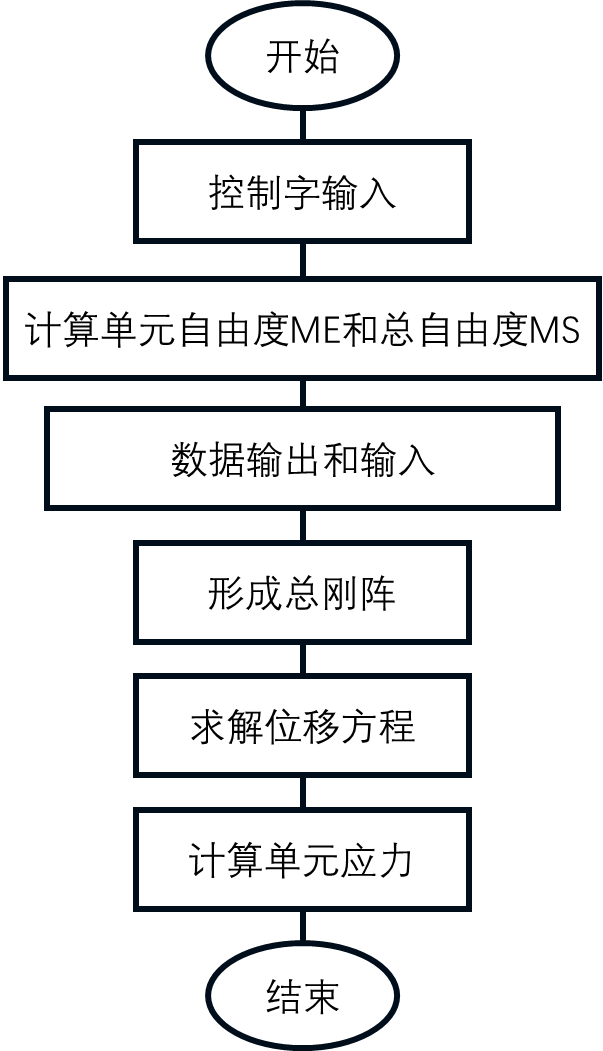
杆单元的能量泛函为：

同理可以得到轴的为：

最后进行组合可以得到总得能量泛函为

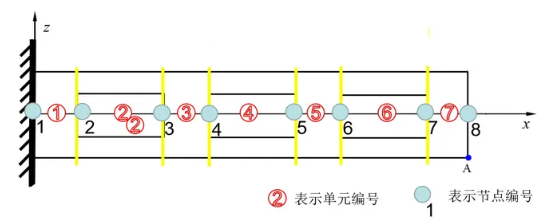
当取变积分为0时得到空间梁的刚度方程为

2. 有限元程序整体框架

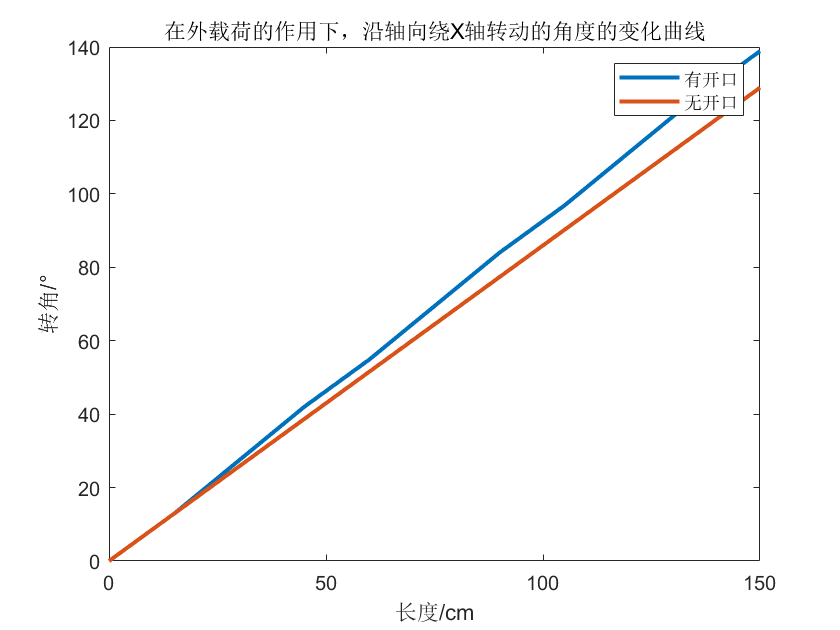
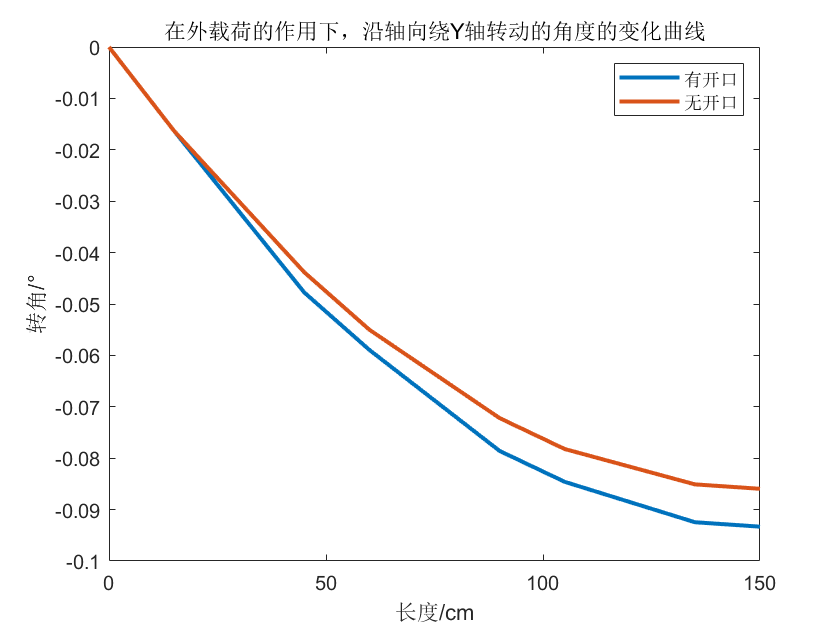
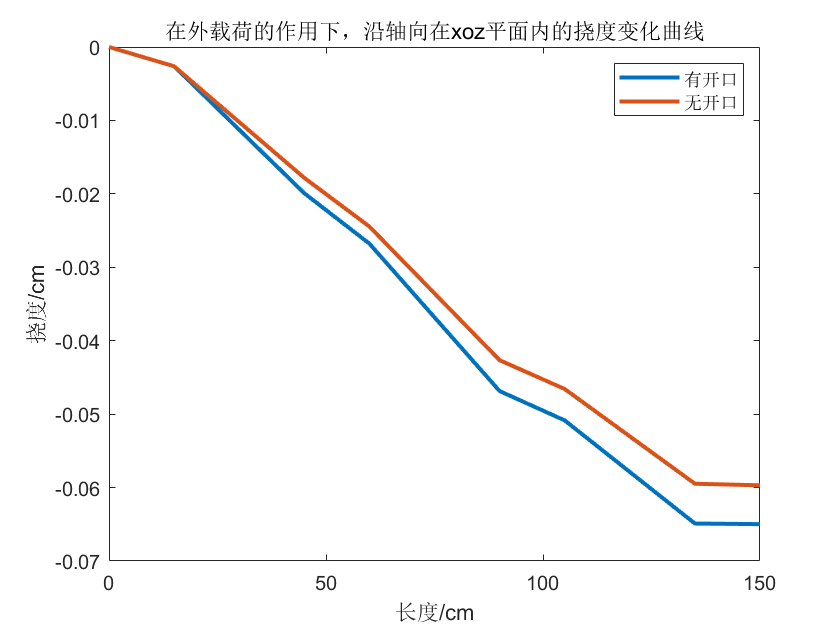


3. 离散化与编号

在本次作业中根据梁的截面变化将梁分成七个单元从左到右依次编号为①②③④…⑦,为了保证最后组集得到最终刚度矩阵的稀疏性，减小存储矩阵的半带宽，节点编号也是采用从左到右依次1,2,3,4，…8的编号方法，具体离散化和单元以及节点编号的示意图如下图所示：



4. 和无开口梁情况分析比较

为了直观体现有开口梁和无开口梁在外载荷作用下两者广义位移的变化，将两者的结果通过如下的对比图展示(由于没有给定材料密度，所以转动惯量无法计算，在此问题中，假设密度为1，绕y方向转动的角度没有实际意义，只用图来表示两者的变化趋势)

5.结论

通过利用有限元计算，得到端点处的广义位移如下图所示从左到右分别是：

其中长度单位是m，角度单位是弧度。从挠度角度来看，在外载荷和约束的相互作用下，无开口的梁在z方向的挠度为0.0596938cm，有开口梁挠度为0.0649792cm，开口梁在一定程度上破坏了梁的抗弯、抗扭性能，但从该数据来看，破坏的程度不是很大，但开口可以大大减轻梁的自身质量，所以在现代的工程中有也能看到具有开口的梁部件，特别是在大型工程中。

程序代码

global E mu G; % 将泊松比和杨氏模量声明为全局变量

E = 2.0e11; % 给杨氏模量赋予指定值

mu = 0.25; % 给泊松比赋予指定值

G = E/(2\*(1+mu)); % 有泊松比杨氏模量计算剪切模量

L=[15,30,15,30,15,30,15]\*10^(-2);

A=[30\*10,15\*10,30\*10,15\*10,30\*10,15\*10,30\*10]\*10^(-4);

Iy=[10\*(30^3),10\*(30^3)-10\*(15^3),10\*(30^3),10\*(30^3)-10\*(15^3),...

10\*(30^3),10\*(30^3)-10\*(15^3),10\*(30^3)]/12\*10^(-8);

Iz=[30\*(10^3),15\*(10^3),30\*(10^3),15\*(10^3),30\*(10^3),15\*(10^3),...

30\*(10^3)]/12\*10^(-8);

%%%假设材料单位面积密度为1

J=[10^3\*30+30^3\*10,10^3\*30+30^3\*10-(10\*3\*15+15^3\*10),10^3\*30+30^3\*10,...

10^3\*30+30^3\*10-(10\*3\*15+15^3\*10),10^3\*30+30^3\*10,10^3\*30+30^3\*10-(10\*3\*15+15^3\*10),...

10^3\*30+30^3\*10]/12\*10^(-12);

ky=[6/5,1.305102041,6/5,1.305102041,6/5,1.305102041,6/5];

kz=[1,1,1,1,1,1,1]\*6/5;

F\_load=zeros(1,6\*8);

% smys Fx Fy Fz Mx My Mz;

% F\_load(1:6)=[Fx,Fy,Fz,Mx,My,Mz];

F\_load(1:6)=[0,0,0,0,0,0];

F\_load(end-5:end)=[0,0,-60000,3000,0,0];

element\_array=[1,2,3,4,5,6,7];

K=zeros(48,48); % 初始化总的刚度矩阵

for n=1:1:length(element\_array)

k = Beam3D2Node\_Stiffness(A(n),J(n),[Iy(n),Iz(n)],...

12\*E\*[Iy(n)\*ky(n), Iz(n)\*kz(n)]/(G\*A(n)\*L(n)\*L(n)),L(n)); % 求每一个单元的刚度矩阵

K = Beam3D2Node\_Assembly\_Stiffness(K,k,n,n+1);

end

u\_carved=F\_load(7:48)/K(7:48,7:48);

display(u\_carved(end-5:end))

A0=A(1);

J0=J(1);

Iy0=Iy(1);

Iz0=Iz(1);

ky0=ky(1);

kz0=kz(1);

K1=zeros(48,48);

for n=1:1:length(element\_array)

k1 = Beam3D2Node\_Stiffness(A0,J0,[Iy0,Iz0],...

12\*E\*[Iy0\*ky0,Iz0\*kz0]/(G\*A0\*L(n)\*L(n)),L(n)); % 求每一个单元的刚度矩阵

K1 = Beam3D2Node\_Assembly\_Stiffness(K1,k1,n,n+1);

end

u=F\_load(7:48)/K1(7:48,7:48);

display(u(end-5:end));

plot([0,15,45,60,90,105,135,150],[0,u\_carved(3:6:42)]\*100,"LineWidth",2)

hold on

plot([0,15,45,60,90,105,135,150],[0,u(3:6:42)]\*100,"LineWidth",2)

title("在外载荷的作用下，沿轴向在xoz平面内的挠度变化曲线")

legend('有开口','无开口')

xlabel("长度/cm")

ylabel("挠度/cm")

figure(2)

plot([0,15,45,60,90,105,135,150],[0,u\_carved(5:6:42)]\*57.3,"LineWidth",2)

hold on

plot([0,15,45,60,90,105,135,150],[0,u(5:6:42)]\*57.3,"LineWidth",2)

title("在外载荷的作用下，沿轴向绕Y轴转动的角度的变化曲线")

legend('有开口','无开口')

xlabel("长度/cm")

ylabel("转角/°")

figure(3)

plot([0,15,45,60,90,105,135,150],[0,u\_carved(4:6:42)]\*57.3,"LineWidth",2)

hold on

plot([0,15,45,60,90,105,135,150],[0,u(4:6:42)]\*57.3,"LineWidth",2)

title("在外载荷的作用下，沿轴向绕X轴转动的角度的变化曲线")

legend('有开口','无开口')

xlabel("长度/cm")

ylabel("转角/°")

function k = Beam3D2Node\_Stiffness(A,J,I,b,L)

global E G % 声明全局变量

% 该函数计算单元的刚度矩阵

% 输入为梁的截面面积A，梁的转动惯量J，横截面惯性矩I(1\*2)[Ix,Iy]和梁单元的长度L

% 输出单元的刚度矩阵k(12x12)

% ----------------------------------------------------------

k=zeros(12\*12);

%%%空间梁轴向拉伸的刚度元素%%%%

k(1,1)= E\*A/L;

k(1,7)=-E\*A/L;

k(7,1)=-E\*A/L;

k(7,7)= E\*A/L;

%%%空间梁绕x轴的扭转的刚度元素%%%%

k(4,4)=G\*J/L;

k(4,10)=-G\*J/L;

k(10,4)=-G\*J/L;

k(10,10)=G\*J/L;

%%%%空间梁在xoy平面内的刚度元素%%%

k(2,2)=12\*E\*I(1)/(1+b(1))/(L^3);

k(2,8)=-12\*E\*I(1)/(1+b(1))/(L^3);

k(2,6)=6\*E\*I(1)\*L/(1+b(1))/(L^3);

k(2,12)=6\*E\*I(1)\*L/(1+b(1))/(L^3);

k(6,2)=6\*E\*I(1)\*L/(1+b(1))/(L^3);

k(6,8)=-6\*E\*I(1)\*L/(1+b(1))/(L^3);

k(6,6)=(4+b(1))\*E\*I(1)\*(L^2)/(1+b(1))/(L^3);

k(6,12)=(2-b(1))\*E\*I(1)\*(L^2)/(1+b(1))/(L^3);

k(8,2)=-12\*E\*I(1)/(1+b(1))/(L^3);

k(8,8)=12\*E\*I(1)/(1+b(1))/(L^3);

k(8,6)=-6\*E\*I(1)\*L/(1+b(1))/(L^3);

k(8,12)=-6\*E\*I(1)\*L/(1+b(1))/(L^3);

k(12,2)=6\*E\*I(1)\*L/(1+b(1))/(L^3);

k(12,8)=-6\*E\*I(1)\*L/(1+b(1))/(L^3);

k(12,6)=(2-b(1))\*E\*I(1)\*(L^2)/(1+b(1))/(L^3);

k(12,12)=(4+b(1))\*E\*I(1)\*(L^2)/(1+b(1))/(L^3);

%%%%空间梁在xoz平面内的刚度元素%%%

k(3,3)=12\*E\*I(2)/(1+b(2))/(L^3);

k(3,9)=-12\*E\*I(2)/(1+b(2))/(L^3);

k(3,5)=6\*E\*I(2)\*L/(1+b(2))/(L^3);

k(3,11)=6\*E\*I(2)\*L/(1+b(2))/(L^3);

k(5,3)=6\*E\*I(1)\*L/(1+b(1))/(L^3);

k(5,9)=-6\*E\*I(1)\*L/(1+b(1))/(L^3);

k(5,5)=(4+b(1))\*E\*I(1)\*(L^2)/(1+b(1))/(L^3);

k(5,11)=(2-b(1))\*E\*I(1)\*(L^2)/(1+b(1))/(L^3);

k(9,3)=-12\*E\*I(1)/(1+b(1))/(L^3);

k(9,9)=12\*E\*I(1)/(1+b(1))/(L^3);

k(9,5)=-6\*E\*I(1)\*L/(1+b(1))/(L^3);

k(9,11)=-6\*E\*I(1)\*L/(1+b(1))/(L^3);

k(11,3)=6\*E\*I(1)\*L/(1+b(1))/(L^3);

k(11,9)=-6\*E\*I(1)\*L/(1+b(1))/(L^3);

k(11,5)=(2-b(1))\*E\*I(1)\*(L^2)/(1+b(1))/(L^3);

k(11,11)=(4+b(1))\*E\*I(1)\*(L^2)/(1+b(1))/(L^3);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

end

function K = Beam3D2Node\_Assembly\_Stiffness(K\_temp,k,i,j)

% 该函数用于单元刚度矩阵的组装

% 输入单元的刚度矩阵k(12x12)；和单元的节点编号i,j，K\_temp为过渡刚度矩阵

% 输出整体刚度矩阵K

% ----------------------------------------------------------

DOF(1)=6\*i-5;

DOF(2)=6\*i-4;

DOF(3)=6\*i-3;

DOF(4)=6\*i-2;

DOF(5)=6\*i-1;

DOF(6)=6\*i;

DOF(7)=6\*j-5;

DOF(8)=6\*j-4;

DOF(9)=6\*j-3;

DOF(10)=6\*j-2;

DOF(11)=6\*j-1;

DOF(12)=6\*j;

for n1=1:12

for n2=1:12

K\_temp(DOF(n1),DOF(n2))= K\_temp(DOF(n1),DOF(n2))+k(n1,n2);

end

end

K=K\_temp;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

end

function f\_equi = Beam3D2Node\_Elem\_Equivload(L,pz,py,px,Mx)

% 该函数用于计算单元等效节点载荷;

% 输入单元中间的外压力载荷函数p;

% 输入梁单元的长度L和所测点距单元左节点的距离x

% 输出该单元等效节点载荷F\_equiload

% ----------------------------------------------------------

syms x ;

e=x/L;

% 此行可以定义随x变化压力载荷函数，此处用了常压力-10N

%p=-250000;

N1=1 - 3\*e\*e + 2\*e\*e\*e;

N2=L\*(e - 2\*e\*e + e\*e\*e);

N3=3\*e\*e - 2\*e\*e\*e;

N4=L\*(e\*e\*e - e\*e);

N5=1-e;

N6=e;

s1=px\*N5;

s2=py\*(N1+N5)/2;

s3=pz\*(N1+N5)/2;

s4=Mx\*N5;

s5=py\*N2;

s6=pz\*N2;

s7=px\*N6;

s8=py\*(N3+N6)/2;

s9=pz\*(N3+N6)/2;

s10=Mx\*N6;

s11=py\*N4;

s12=pz\*N4;

F1=int(s1,[0,L]);

F2=int(s2,[0,L]);

F3=int(s3,[0,L]);

F4=int(s4,[0,L]);

F5=int(s5,[0,L]);

F6=int(s6,[0,L]);

F7=int(s7,[0,L]);

F8=int(s8,[0,L]);

F9=int(s9,[0,L]);

F10=int(s10,[0,L]);

F11=int(s11,[0,L]);

F12=int(s12,[0,L]);

f\_equi=[F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,F11,F12];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

end

function F\_equi = Beam3D2Node\_Assembly\_Equi(F\_equi,f\_equi,i,j)

% 该函数用于单元等效节点载荷的组装

% 输入单元的等效节点载荷f\_equi(12x1)；和单元的节点编号i,j

% 输出整体等效节点载荷F\_equi

% ----------------------------------------------------------

DOF(1)=6\*i-5;

DOF(2)=6\*i-4;

DOF(3)=6\*i-3;

DOF(4)=6\*i-2;

DOF(5)=6\*i-1;

DOF(6)=6\*i;

DOF(7)=6\*j-5;

DOF(8)=6\*j-4;

DOF(9)=6\*j-3;

DOF(10)=6\*j-2;

DOF(11)=6\*j-1;

DOF(12)=6\*j;

for n1=1:12

F\_equi(DOF(n1))= F\_equi(DOF(n1))+f\_equi(n1);

end

%

end