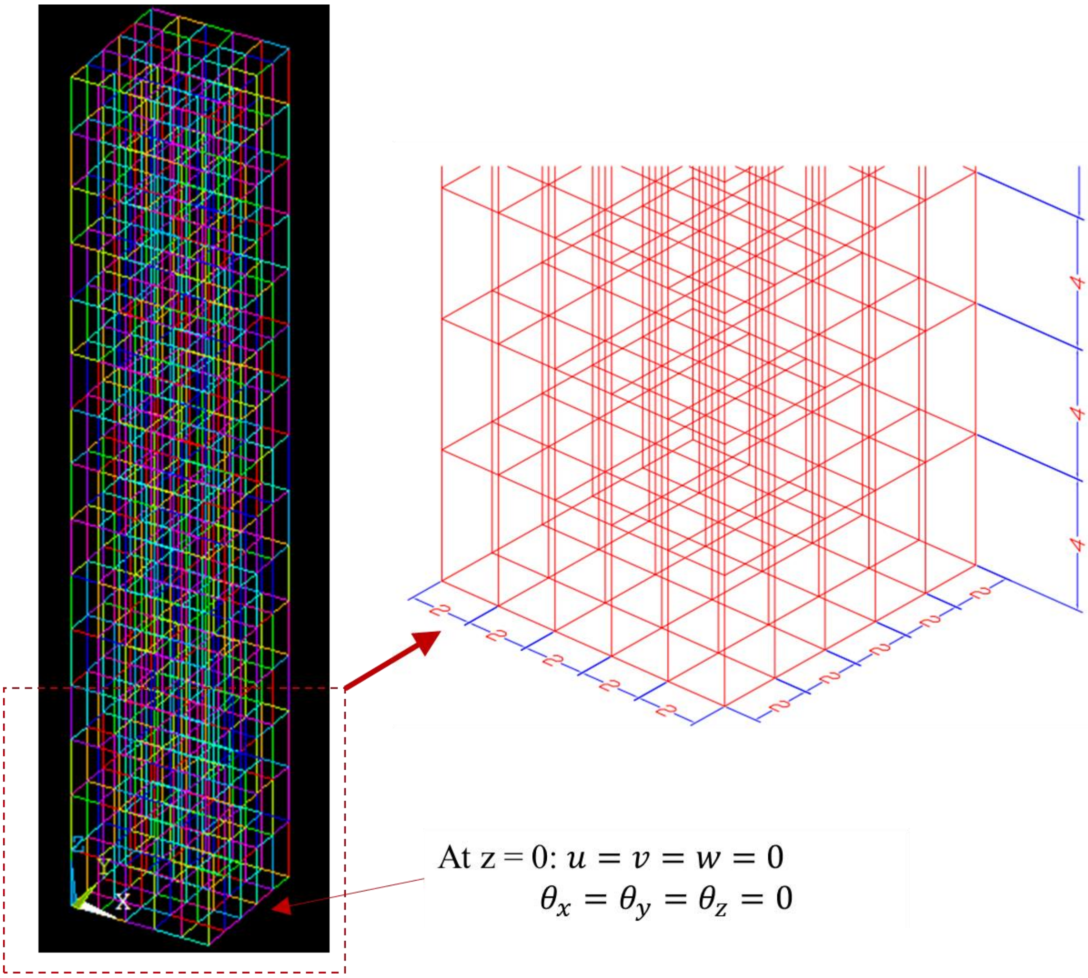
作业说明

加载：由于风荷载，在 x=0， z≥32 m 的垂直表面上，建筑物受到压力。

问题：求解梁的位移与旋转。

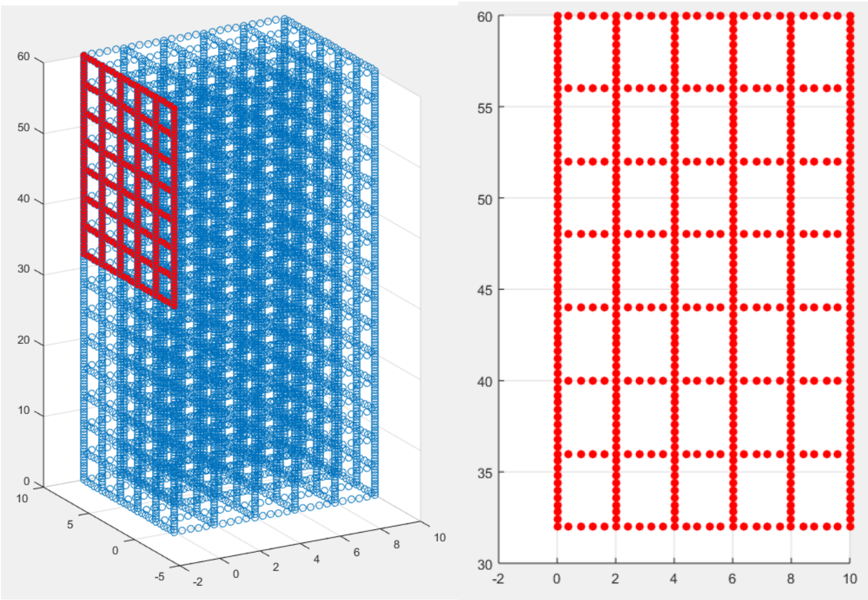


几何和材料特性：

强梁 = 所有垂直梁：$b = h = 0.4\,m$

法向梁 = 所有水平梁：$b = h = 0.2\,m$

材料：$E = 30$ GPa; $\nu = 0.2$



区域中心：　　　　$y = 1, 3, 5, 7, 9$

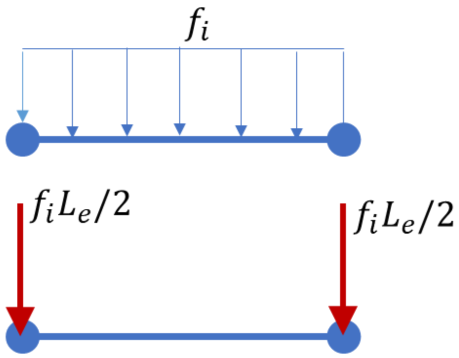
$z = 34, 38, 42, 46, 50, 54, 58$

每个矩形的面积：$A_i = 2 \times 4 = 8\,m^2$

将压力转换为力：$F_i = A_i\sigma = 8 \times 50 = 400\,N$

将 F_i 平均划分为矩形的周长：$f_i = \frac{F_i}{2+2+4+4} = \frac{400}{12}$ (N/m)

每个元件的负载：



主程序（main_Building.m）

过程：定义材料特性；从 ANSYS 导入网格；定义几何特性；定义单位向量；计算刚度矩阵；边界条件设置；加载条件设置；结果后处理。总计 9276 个节点，55656 个自由度。

1、材料特性、几何特性

材料：$E = 30$ GPa; $\nu = 0.2$

编号: 1
梁截面子类型: 矩形
梁截面名称为:
梁截面数据汇总:

Area	= 0.16000
Iyy	= 0.21333E-02
Iyz	=-0.54210E-19
Izz	= 0.21333E-02
翘曲常数	= 0.39448E-06
扭转常数	= 0.36513E-02
质心 Y	=-0.64375E-17
质心 Z	= 0.13553E-17
剪切中心 Y	=-0.10522E-16
剪切中心 Z	= 0.19505E-16
剪切校正-yy	= 0.84211
剪切校正-yz	=-0.13313E-16
剪切校正-zz	= 0.84211

编号: 2
梁截面子类型: 矩形
梁截面名称为: :
梁截面数据汇总:

Area	= 0.40000E-01
Iyy	= 0.13333E-03
Iyz	=-0.33881E-20
Izz	= 0.13333E-03
翘曲常数	= 0.61637E-08
扭转常数	= 0.22821E-03
质心 Y	=-0.32187E-17
质心 Z	= 0.67763E-18
剪切中心 Y	=-0.52609E-17
剪切中心 Z	= 0.97525E-17
剪切校正-yy	= 0.84211
剪切校正-yz	=-0.13313E-16
剪切校正-zz	= 0.84211

2、从 ANSYS 导入网格

```matlab
id = ~isnan(NLIST(:,1));
NLIST = NLIST(id,1:4);

id = ~isnan(ELIST(:,1));
ELIST = ELIST(id,1:8);

%%%%% 重新命名节点坐标和节点
xyz = NLIST(:,2:4);
nP = size(xyz,1);

nE = size(ELIST,1);
eNodes = ELIST(:,7:8);
```

NLIST.lis

	A	B	C	D	E	F
	NLIST					
	数值	数值	数值	数值	数值	数值
1						
2	LIST	ALL	SELECTED	NODES.	DSYS=	0
3	SORT	TABLE	ON	NODE	NODE	NODE
4						
5	NODE	X	Y	Z		
6	1	2.00000000000	4.00000000000	16.0000000000		
7	2	2.00000000000	6.00000000000	16.0000000000		
8	3	2.00000000000	4.40000000000	16.0000000000		
9	4	2.00000000000	4.80000000000	16.0000000000		
10	5	2.00000000000	5.20000000000	16.0000000000		
11	6	2.00000000000	5.60000000000	16.0000000000		
12	7	0.164000000000E-13	4.00000000000	16.0000000000		
13	8	0.142000000000E-13	6.00000000000	16.0000000000		
14	9	0.159600000000E-13	4.40000000000	16.0000000000		
15	10	0.155200000000E-13	4.80000000000	16.0000000000		
16	11	0.150800000000E-13	5.20000000000	16.0000000000		
17	12	0.146400000000E-13	5.60000000000	16.0000000000		
18	13	0.620000000000E-14	10.0000000000	12.0000000000		
19	14	0.620000000000E-14	10.0000000000	16.0000000000		
20	15	0.620000000000E-14	10.0000000000	12.4000000000		
21	16	0.620000000000E-14	10.0000000000	12.8000000000		
22	17	0.620000000000E-14	10.0000000000	13.2000000000		
23	18	0.620000000000E-14	10.0000000000	13.6000000000		

NLIST.lis ELIST.lis

	A	B	C	D	E	F	G	H	I
	ELIST								
	数值	数值	数值	数值	数值	数值	数值	数值	数值
1									
2	LIST	ALL	SELECTED	ELEMENTS.	(LIST	NODES)			
3									
4	ELEM	MAT	TYP	REL	ESY	SEC	NODES		
5									
6	1	1	1	1	0	2	1	3	0
7	2	1	1	1	0	2	3	4	0
8	3	1	1	1	0	2	4	5	0
9	4	1	1	1	0	2	5	6	0
10	5	1	1	1	0	2	6	2	0
11	6	1	1	1	0	2	7	9	0
12	7	1	1	1	0	2	9	10	0
13	8	1	1	1	0	2	10	11	0
14	9	1	1	1	0	2	11	12	0
15	10	1	1	1	0	2	12	8	0
16	11	1	1	1	0	1	13	15	0
17	12	1	1	1	0	1	15	16	0
18	13	1	1	1	0	1	16	17	0
19	14	1	1	1	0	1	17	18	0
20	15	1	1	1	0	1	18	19	0
21	16	1	1	1	0	1	19	20	0
22	17	1	1	1	0	1	20	21	0
23	18	1	1	1	0	1	21	22	0

3、方向向量(directVectors.m)

```
function [direcVec] = directVectors(nE,eNodes,xyz,refP)
%%%% 计算所有元素的方向向量
direcVec = zeros(nE,9);
for i = 1:nE
    node1 = eNodes(i,1);%%%% 开始节点
    node2 = eNodes(i,2);%%%% 终止节点
    dxyz = xyz(node2,:) - xyz(node1,:);
    [Le] = lenVect(dxyz);

    Vri = 1/Le*dxyz;%%%% 单位向量
    Vs0 = refP - xyz(node1,:);
    [le_Vs0] = lenVect(Vs0);
    Vs0 = 1/le_Vs0*Vs0;%%%% 单位向量

    Vti = cross(Vri,Vs0);
    [le_Vti] = lenVect(Vti);
    Vti = 1/le_Vti*Vti;%%%% 单位向量

    Vsi = cross(Vti,Vri);
    [le_Vsi] = lenVect(Vsi);
    Vsi = 1/le_Vsi*Vsi;%%%% 单位向量

    Vri = cross(Vsi,Vti);
    [le_Vri] = lenVect(Vri);
    Vri = 1/le_Vri*Vri;%%%% 单位向量


    direcVec(i,:) = [Vri,Vsi,Vti];
end
end
```

4、剪切和轴向变形(stiff_shear_axial.m)

```
function [keLoc] = stiff_shear_axial(invJ,nnode,A,E,detJ,ks,G,i)
%%%% 计算刚度矩阵
%%%% 用于剪切和轴向变形
%%%%% 初始化局部坐标系中的单元刚度矩阵
keLoc = zeros(12,12);

%%%%% xi = 0, weight = 2
%%%%% 用于剪切和轴向刚度矩阵
[shape,nDeriv] = shapeFunct_Beam(0);
gaussWt = 2;

Xderiv = nDeriv*invJ;
B = zeros(1,nnode); B(1:nnode) = Xderiv(:);
%%%%% 局部坐标系中的单元刚度矩阵
Ke_ax = A(i,1)*E(i,1)*(B'*B)*detJ*gaussWt;

B_s1 = [B, -shape'];
Ke_s1 = ks(i,1)*G(i,1)*A(i,1)*(B_s1'*B_s1)*detJ*gaussWt;

B_s2 = [B, shape'];
Ke_s2 = ks(i,1)*G(i,1)*A(i,1)*(B_s2'*B_s2)*detJ*gaussWt;

%%%%% 局部刚度矩阵
uDofL = [1,7];
keLoc(uDofL,uDofL) = keLoc(uDofL,uDofL) + Ke_ax;

s1DofL = [2,8, 6,12];
keLoc(s1DofL,s1DofL) = keLoc(s1DofL,s1DofL) + Ke_s1;

s2DofL = [3,9, 5,11];
keLoc(s2DofL,s2DofL) = keLoc(s2DofL,s2DofL) + Ke_s2;

end
```

5、弯曲和扭转(stiff_bend_tors.m)

```
function [keLoc2] = stiff_bend_tors(keLoc,gaussLoc,gaussWts,...
    detJ,invJ,nnode,kt,G,E,Iy,Iz,i)
%%%%% 计算弯曲和扭转刚度 矩阵
keLoc2 = keLoc;
%%%%%% gaussLocations = [-1/sqrt(3), 1/sqrt(3)], weight = [1, 1]
for ii = 1:length(gaussLoc)
    xi = gaussLoc(ii,1);
    [~,nDeriv] = shapeFunct_Beam(xi);
    gaussWt = gaussWts(ii,1);

    Xderiv = nDeriv*invJ;
    B = zeros(1,nnode); B(1:nnode) = Xderiv(:);

    %%%%% 局部坐标系中的单元刚度矩阵
    Ke_tx = kt(i,1)*G(i,1)*(B'*B)*detJ*gaussWt;
    Ke_by = E(i,1)*Iy(i,1)*(B'*B)*detJ*gaussWt;
    Ke_bz = E(i,1)*Iz(i,1)*(B'*B)*detJ*gaussWt;

    rxDofL = [4,10];
    ryDofL = [5,11];
    rzDofL = [6,12];

    keLoc2(rxDofL,rxDofL) = keLoc2(rxDofL,rxDofL) + Ke_tx;
    keLoc2(ryDofL,ryDofL) = keLoc2(ryDofL,ryDofL) + Ke_by;
    keLoc2(rzDofL,rzDofL) = keLoc2(rzDofL,rzDofL) + Ke_bz;
end
end
```

6、变换矩阵(transform_stiff.m)

```
function [keG,eDofs] = transform_stiff(direcVec,keLoc2,i,eNodei,nP)
%%%%%% 变换局部刚度矩阵: keLoc2
%%%%%% 到全局坐标系
%%%%%% 定义元素并正确分配元素刚度
%%%%%% 总刚度矩阵

H0 = zeros(3,3);
H = [direcVec(i,1:3);direcVec(i,4:6);direcVec(i,7:9)];
T = [H,  H0,H0,H0;
     H0,H,  H0,H0;
     H0,H0,H,  H0;
     H0,H0,H0,H];

keG = T'*keLoc2*T;

eDofs = [eNodei; eNodei+nP; eNodei+2*nP;...
    eNodei+3*nP; eNodei+4*nP; eNodei+5*nP];
eDofs = [eDofs(:,1);eDofs(:,2)];
end
```

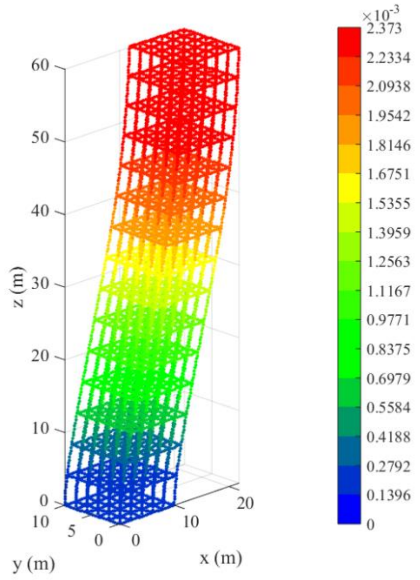
7、形函数和刚度矩阵

```
function [shape,nderiv] = shapeFunct_Beam(xi)
%%%% shapeN : 形函数 N1 和 N2
%%%% N_deriv: 推导 N1 and N2 w.r.t. xi
%%%% xi: 自然坐标 (-1 ... +1)

shape =1/2*[1-xi;1+xi];

nderiv = [-1;1]/2;

end
```

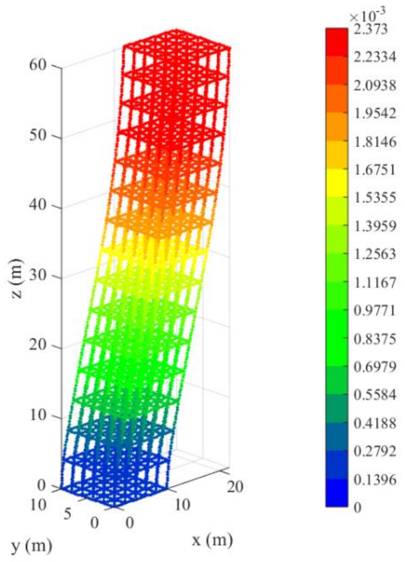
```
function [K] = funct_Stiff_Beam3D(nP,nE,eNodes,xyz,...
    A,E,ks,G,Iy,Iz,kt,direcVec)
%%%% 计算梁结构刚度矩阵
nDof = nP*6;
K = sparse(nDof,nDof);

gaussLoc = 1/sqrt(3)*[-1; 1];
gaussWts = [1; 1];

for i = 1:nE
    eNodei = eNodes(i,:); %%%%% eNodei = [node 1st, node 2nd]
    nnode = length(eNodei);%%%% 每个元素的节点数 = 2

    %%%% det(J) and J^-1
    [detJ, invJ] = detJ_invJ(xyz,eNodei);

    %%%% 简单计算剪切和轴向刚度矩阵
    %%%% 组装
    [keLoc] = stiff_shear_axial(invJ,nnode,A,E,detJ,ks,G,i);

    %%%% 完整计算弯曲和扭转刚度矩阵
    %%%% 组装
    [keLoc2] = stiff_bend_tors(keLoc,gaussLoc,gaussWts,...
        detJ,invJ,nnode,kt,G,E,Iy,Iz,i);
    %%%%% 局部刚度矩阵计算完成
    %%%%% 元素, 转移到全局坐标系

    %%%%%% 刚度的变换矩阵
    [keG,eDofs] = transform_stiff(direcVec,keLoc2,i,eNodei,nP);

    %%%%% 全局刚度矩阵
    K(eDofs,eDofs) = K(eDofs,eDofs) + keG;
end
```

为了验证编写的程序是否能够有效计算，将结果与 ansys 软件的计算结果进行对比。

位移 u

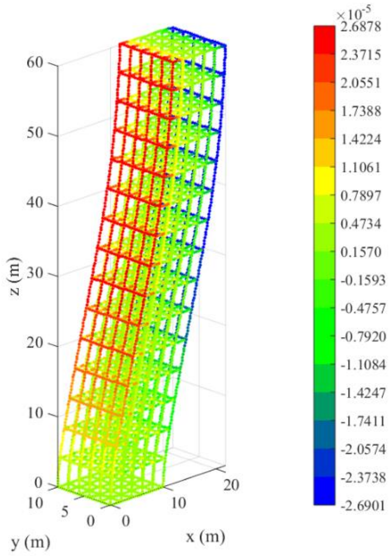


FEM 程序

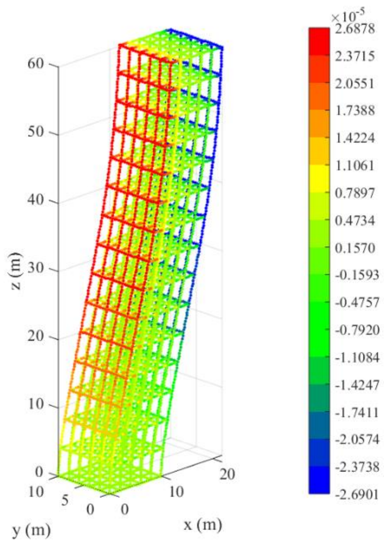


ANSYS

位移 w

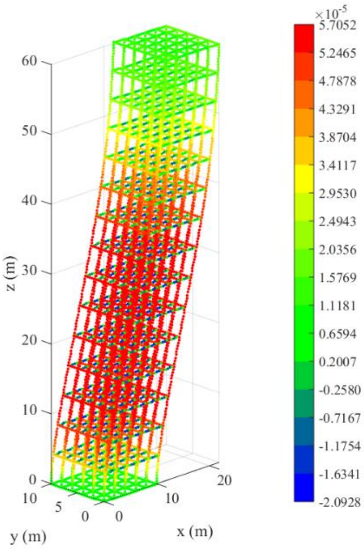


FEM 程序

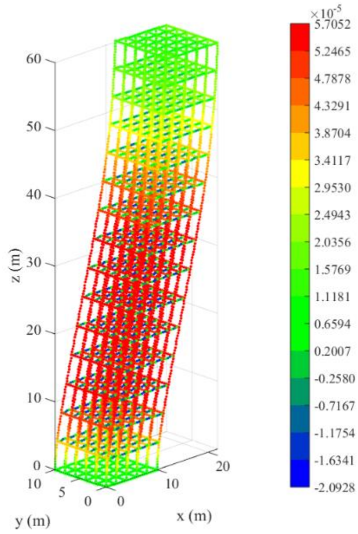


ANSYS

旋转 θ_y


×10^-5
5.7052
5.2465
4.7878
4.3291
3.8704
3.4117
2.9530
2.4943
2.0356
1.5769
1.1181
0.6594
0.2007
-0.2580
-0.7167
-1.1754
-1.6341
-2.0928
z (m)
x (m)
y (m)

FEM 程序


×10^-5
5.7052
5.2465
4.7878
4.3291
3.8704
3.4117
2.9530
2.4943
2.0356
1.5769
1.1181
0.6594
0.2007
-0.2580
-0.7167
-1.1754
-1.6341
-2.0928
z (m)
x (m)
y (m)

ANSYS