

同濟大學

TONGJI UNIVERSITY

《有限单元法》课程 第一次作业

作业名称 平面三角形单元有限元法程序

组 别 第七组

小组成员 2330739 公冶祥嵩 2330689 高子寒 2330596 李雨琪

2330954 李子轩 2330994 叶芳怡

指导教师 肖伟芳

日 期 2024 年 5 月 15 日

一、计算模型	1
二、计算原理	1
三、编程思路	2
四、计算公式	2
1、应变矩阵 B	2
2、弹性矩阵 D	2
3、单元刚度矩阵 K	3
五、MATLAB 程序代码	3
1、计算主程序 Main.m	3
2、单元刚度矩阵函数 TriangleElementStiffness.m	7
3、刚度矩阵集成函数 AssemTriangle.m	8
4、单元应力计算函数 TriangleElementStress.m	8
六、结果对比	9
1、ABAQUS 模型设置	9
2、结点位移对比	10
3、单元应力对比	11

一、计算模型

长 10m，高 2m，厚 10mm 的两端简支平面构件，受到竖直向下、大小为 20kN/m 的均布荷载作用，如图 1 所示。材料的弹性模量 $E=210\text{GPa}$ ，泊松比 $\nu=0.3$ 。

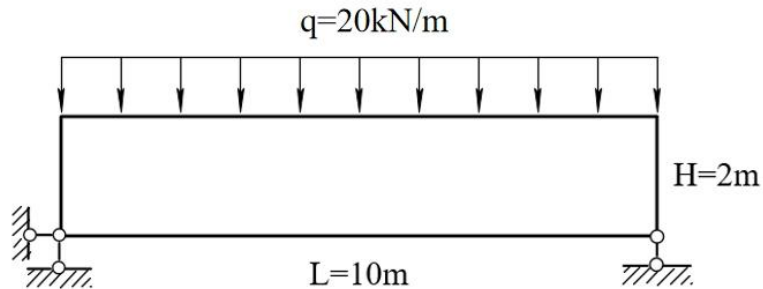


图 1 平面构件受力简图

二、计算原理

用矩阵位移法分析杆系结构时，把一个杆件作为一个单元，这些单元在结点上相互连接，以结点位移作为基本的未知量。由于结点的个数是有限的，所以结点平衡方程的个数也是有限的，从而可以借助线性代数（矩阵）方程组进行分析。

在一个连续介质中，互相连接的点是无限的，因而具有无限个自由度。但如果把这个连续介质离散成有限个单元的组合体，这些单元在有限个结点上相互连接，包含有限个自由度，从而可以用矩阵方法进行分析。

归纳总结得到二维连续介质的有限单元法分析的基本步骤如下：

- (1) 用虚拟的直线把原介质分割成有限个三角形单元，这些直线是单元的边界，几条直线的交点称为结点；
- (2) 假定各单元在结点上互相铰接，结点位移是基本的未知量；
- (3) 选择位移函数，用单元的三个结点的位移惟一表示单元内任一点的位移；
- (4) 通过位移函数，用结点位移惟一表示单元内任一点的应变（几何方程）；再利用广义虎克定律，用结点位移惟一表示单元内任一点的应力（物理方程）；
- (5) 利用能量原理，找到与单元内部应力状态等效的结点力；再利用单元应力与结点位移的关系，建立等效结点力与结点位移的关系（单元刚度矩阵）；
- (6) 将每一单元所承受的载荷，按静力等效原则移置到结点上；
- (7) 集成所有单元的刚度矩阵得到总刚度矩阵，利用位移边界条件和力边界条件建立静力平衡方程，得到一个线性方程组；通过求解该方程组，得到结点位移；进而得到单元应力。

三、编程思路

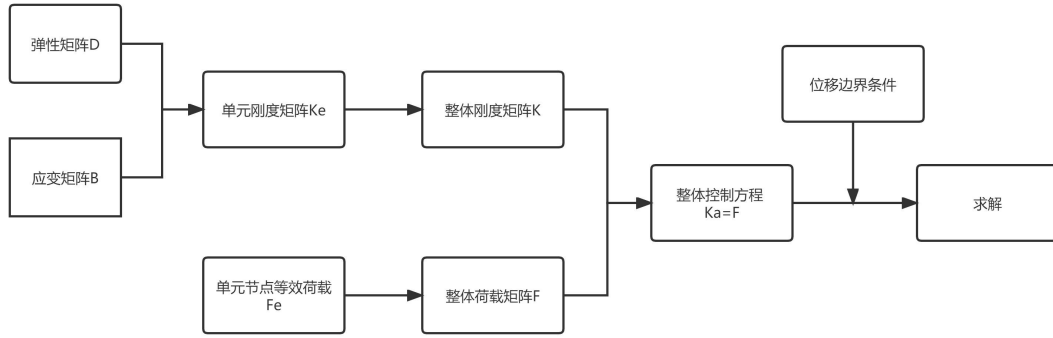


图 2 MATLAB 编程思路

- (1) 输入计算模型尺寸，设置材料参数。
- (2) 划分三角形网格，对单元、结点进行编号，其中结点编号以逆时针方向编码。
- (3) 利用弹性矩阵 D 和应变矩阵 B 计算单元刚度矩阵 K_e ，通过“对号入座”方式叠加形成整体刚度矩阵 K 。
- (4) 计算单元结点等效荷载 P_e ，通过“对号入座”方式叠加形成整体荷载列阵 P 。
- (5) 引入位移边界条件和力边界条件。
- (6) 求解平衡方程 $Ka = P$ ，得到结点位移及单元应力。

四、计算公式

1、应变矩阵 B

$$B_i = \frac{1}{2A} \begin{bmatrix} b_i & 0 \\ 0 & c_i \\ c_i & b_i \end{bmatrix} \quad (i, j, m)$$

$$A = \frac{1}{2}(a_i + a_j + a_m)$$

$$a_i = x_j y_m - x_m y_j$$

$$b_i = y_j - y_m \quad (i, j, m)$$

$$c_i = x_m - x_j$$

2、弹性矩阵 D

$$D = \frac{E_0}{(1-\nu_0^2)} \begin{bmatrix} 1 & \nu_0 & 0 \\ \nu_0 & 1 & 0 \\ 0 & 0 & \frac{1-\nu_0}{2} \end{bmatrix}$$

3、单元刚度矩阵 K

$$K_{rs} = B_r^T D B_s t A = \frac{E_0 t}{4(1-\nu_0^2) A} \begin{bmatrix} K_1 & K_3 \\ K_2 & K_4 \end{bmatrix} \quad (r, s = i, j, m)$$

$$K_1 = b_r b_s + \frac{1-\nu_0}{2} c_r c_s$$

$$K_2 = \nu_0 c_r b_s + \frac{1-\nu_0}{2} b_r c_s$$

$$K_3 = \nu_0 b_r c_s + \frac{1-\nu_0}{2} c_r b_s$$

$$K_4 = c_r c_s + \frac{1-\nu_0}{2} b_r b_s$$

五、MATLAB 程序代码

该程序适用于解决薄简支梁受均布荷载作用的问题，可以求解结点位移和单元应力。程序可以进行修改的变量有梁的长度、梁的高度、梁的厚度（此项应远小于梁另外两个维度的尺寸，以符合平面应力问题计算假定）、三角形网格尺寸、材料泊松比、材料弹性模量（假定材料为各向同性理想弹性材料）、均布荷载线集度。

当问题被明确定义后，程序将简支梁划分三角形单元网格，并对单元及结点进行编号（逆时针顺序）生成 **ele** 矩阵，计算结点坐标生成 **node** 矩阵。之后结合应变矩阵 **B** 和弹性矩阵 **D**，利用单元刚度矩阵函数 **TriangleElementStiffness.m** 计算各个单元的刚度矩阵，再利用刚度矩阵集成函数 **AssemTriangle.m** 组装单元刚度矩阵，形成总体刚度矩阵。引入位移边界条件和力边界条件后求解位移矩阵，再利用单元应力计算函数 **TriangleElementStress.m** 求解各个单元应力，最终通过绘图代码对结点位移和单元应力进行可视化输出。

1、计算主程序 Main.m

```
clear;
clc;
%-----
%-----
%程序计算模型定义，包括构件几何参数，单元参数，材料参数，均布荷载大小。
%程序单位采用国际单位制基本单位，长度单位：m，力的单位：N，应力单位：Pa。
L = 10.0;%m 构件长度
H = 2.0;%m 构件高度
t=0.01; %m %单元厚度
ele_length = 0.1;%m 单元长度
ele_height = 0.1;%m 单元高度
E=2.1e11; %Pa 材料弹性模量
miu=0.3; %泊松比
load_q=20000; %N/m 均布荷载
node_load=load_q*ele_length; %等效结点荷载
%-----
```

```

-----

%-----
-----

%网格划分
ele_num_L = L/ele_length; %长度方向单元个数
ele_num_H = H/ele_height; %宽度方向单元个数
node_num_L = ele_num_L + 1; %长度方向节点个数
node_num_H = ele_num_H + 1; %宽度方向节点个数
node = []; %定义节点列表
ele = []; %定义单元列表
%生成节点列表 节点列表结构为 [节点编号 x坐标 y坐标 z坐标 (平面问题, 该项为0)]
for j = 1:node_num_H
    for i = 1 : node_num_L
        node_new = [i+(j-1)*node_num_L ele_length*(i-1) ele_height*(j-1)
0];
        node = cat (1,node,node_new);
    end
end
%生成单元列表 单元列表结构为[单元编号 第一结点编号 第二结点编号 第三结点编号] (左下角为起点, 逆时针顺序)
for j = 1:ele_num_H
    for i = 1 : ele_num_L
        ele_new_odd = [2*i-1+2*(j-1)*ele_num_L i+(j-1)*(ele_num_L+1)
i+1+(j-1)*(ele_num_L+1) i+ node_num_L+1+(j-1)*(ele_num_L+1)];
        ele_new_even = [2*i+2*(j-1)*ele_num_L i+(j-1)*(ele_num_L+1)
i+(j-1)*(ele_num_L+1)+ node_num_L+1 i+(j-1)*(ele_num_L+1)+
node_num_L];
        ele = cat (1,ele,ele_new_odd);
        ele = cat (1,ele,ele_new_even);
    end
end

%-----
-----

%生成单元刚度矩阵
%基础参数与单元定义
num_ele=size(ele,1);
n_ele=length(ele(:,1)); %单元数
dof=length(node(:,1))*2; %自由度, 梁单元的每个节点有 2 个自由度, 横向位移, 纵向位移
f=zeros(dof,1); %整体坐标系下结构整体外荷载矩阵
f_loc=zeros(6,1); %单元外荷载矩阵, 局部坐标系下

```

```

u=ones(dof,1);      %矩阵位移
K=zeros(dof);       %总体刚度矩阵
stress=zeros(n_ele,1); %单元应力矩阵
%生成总体刚度矩阵
for i=1:n_ele
    k_ele=TriangleElementStiffness(E,miu,t,node(ele(i,2:4),2:4));
    K=assemTriangle(K,k_ele,ele(i,2),ele(i,3),ele(i,4));
end

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

%定义边界条件
%力边界条件
for i=node_num_L*(node_num_H-1)+2: node_num_L*node_num_H-1
    f(2*i)=-node_load;%N 顶部
end

f(2*node_num_L*((node_num_H-1)+1)) = -node_load/2;%N 左右边缘两点
f(2*node_num_L*(node_num_H)) = -node_load/2;%N 左右边缘两点
% 位移边界条件;
u(1)=0;
u(2)=0;
%u(2*node_num_L-1)=0;
u(2*node_num_L)=0;

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

%求解
%求解未知自由度
index=[]; %未知自由度索
p=[]; %位置自由度对应的节点力矩阵matrix;
for i =1:dof
    if u(i)~=0
        index=[index,i];
        p=[p;f(i)];
    end
end
%求解位移矩阵
u(index)=K(index,index)\p;
%计算节点变形后位置（为画图明显，对位移进行 20 倍放大
x1=node(:,2)+20*u(1:2:node_num_L*node_num_H*2);
y1=node(:,3)+20*u(2:2:node_num_L*node_num_H*2);

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

```

```

%应力求解及画图
stress=zeros(num_ele,3);
set(0,'defaultfigurecolor','w');
%mises 应力图
figure;
mises_stress=zeros(num_ele);
for i=1:n_ele

u1=[u(2*ele(i,2)-1);u(2*ele(i,2));u(2*ele(i,3)-1);u(2*ele(i,3));u(2*ele(i,4)-1);u(2*ele(i,4))];

stress(i,:)=TriangleElementStress(E,miu,node(ele(i,2:4),2:3),u1,1)';
%单元应力

mises_stress(i)=((stress(i,1)+stress(i,2))^2-3*(stress(i,1)*stress(i,2)-stress(i,3)^2))^0.5;%mises 应力
    patch(node(ele(i,2:4),2),node(ele(i,2:4),3),mises_stress(i));
end
colormap jet;
colorbar;
set(gca,'FontName','Times New Roman','FontSize',14);
set(get(colorbar,'title'),'string','\fontname{Times New Roman}\fontsize{14}Pa');
title('\fontname{宋体}\fontsize{14}单元应力\fontname{Times New Roman}\fontsize{14}(Mises)');
%X 方向主应力图
figure;
for i=1:n_ele
    patch(node(ele(i,2:4),2),node(ele(i,2:4),3),stress(i,1));
end
colormap jet;
colorbar;
set(gca,'FontName','Times New Roman','FontSize',14);
set(get(colorbar,'title'),'string','Pa');
title('\fontname{宋体}\fontsize{14}单元应力\fontname{Times New Roman}\fontsize{14}(\sigma_x)');
%Y 方向主应力图
figure;
for i=1:n_ele
    patch(node(ele(i,2:4),2),node(ele(i,2:4),3),stress(i,2));
end
colormap jet;
colorbar;
set(gca,'FontName','Times New Roman','FontSize',14);

```



```

set(get(colorbar,'title'),'string','Pa');
title('\fontname{宋体}\fontsize{14}单元应力\fontname{Times New Roman}\fontsize{14}(\sigma_y)');
%剪应力图
figure;
for i=1:n_ele
    patch(node(ele(i,2:4),2),node(ele(i,2:4),3),stress(i,3));
end
hold on;
colormap jet;
colorbar;
set(get(colorbar,'title'),'string','Pa');
set(gca,'FontName','Times New Roman','FontSize',14);
title('\fontname{宋体}\fontsize{14}单元应力\fontname{Times New Roman}\fontsize{14}(\tau_x)');
hold on;
%变形图
figure;
for i=1:n_ele

    patch(node(ele(i,2:4),2),node(ele(i,2:4),3),'w','FaceColor','none','LineStyle','-')
        hold on;

    patch(x1(ele(i,2:4)),y1(ele(i,2:4)),'w','FaceColor','none','EdgeColor','r');
end
set(gca,'FontName','Times New Roman','FontSize',14);
legend('\fontname{宋体}\fontsize{14}初始形态','\fontname{宋体}\fontsize{14}加载形态')

```

2、单元刚度矩阵函数 TriangleElementStiffness.m

```

function k_ele=TriangleElementStiffness(E,miu,t,node_ele)
% 三角单元刚度矩阵
x1=node_ele(1,1);
y1=node_ele(1,2);
x2=node_ele(2,1);
y2=node_ele(2,2);
x3=node_ele(3,1);
y3=node_ele(3,2);
% -----
A=(x1*(y2-y3)+x2*(y3-y1)+x3*(y1-y2))/2; %单元面积
a1=x2*y3-y2*x3;
a2=y1*x3-x1*y3;

```

```

a3=x1*y2-y1*x2;
b1=y2-y3;
b2=y3-y1;
b3=y1-y2;
c1=x3-x2;
c2=x1-x3;
c3=x2-x1;
D=E/(1-miu^2)*[1 miu 0;
               miu 1 0;
               0 0 (1-miu)/2];
B1 = [b1 0;0 c1;c1 b1];
B2 = [b2 0;0 c2;c2 b2];
B3 = [b3 0;0 c3;c3 b3];
B_single_element = [B1 B2 B3]./2./A;
k_ele = B_single_element'*D*B_single_element.*t.*A;

```

3、刚度矩阵集成函数 AssemTriangle.m

```

function k_t=assemTriangle(k_t,k_ele,node1,node2,node3)
% 组装总刚
d(1:2)=2*node1-1:2*node1;
d(3:4)=2*node2-1:2*node2;
d(5:6)=2*node3-1:2*node3;
for ii=1:6
    for jj=1:6
        k_t(d(ii),d(jj))=k_t(d(ii),d(jj))+k_ele(ii,jj);
    end
end
end

```

4、单元应力计算函数 TriangleElementStress.m

```

function str=TriangleElementStress(E,miu,node_ele,u1,p)
% 三角单元应力矩阵
x1=node_ele(1,1);
y1=node_ele(1,2);
x2=node_ele(2,1);
y2=node_ele(2,2);
x3=node_ele(3,1);
y3=node_ele(3,2);
% -----
A=(x1*(y2-y3)+x2*(y3-y1)+x3*(y1-y2))/2; %为单元的面积
a1=x2*y3-y2*x3;
a2=y1*x3-x1*y3;
a3=x1*y2-y1*x2;
b1=y2-y3;
b2=y3-y1;
b3=y1-y2;

```

```

c1=x3-x2;
c2=x1-x3;
c3=x2-x1;
B=1/2/A*[b1 0 b2 0 b3 0;
          0 c1 0 c2 0 c3;
          c1 b1 c2 b2 c3 b3];
if p==1
    D=E/(1-miu^2)*[1 miu 0;
                   miu 1 0;
                   0 0 (1-miu)/2];
elseif p==2
    D=E/(1+miu)/(1-2*miu)*[1-miu miu 0;
                           miu 1-miu 0;
                           0 0 (1-2*miu)/2];
end
str=D*B*u1; %为单元应力的计算

```

六、结果对比

1、ABAQUS 模型设置

选取有限元软件 ABAQUS 与 MATLAB 计算程序进行对比。ABAQUS 模型中施加大小相同的均布荷载，边界条件采取左端固定支座，右端 Y 轴方向约束的形式，如图 3 所示。网格划分与 MATLAB 相同，单元大小 0.1m，X 方向 100 个单元，Y 方向 20 个单元，如图 4 所示。

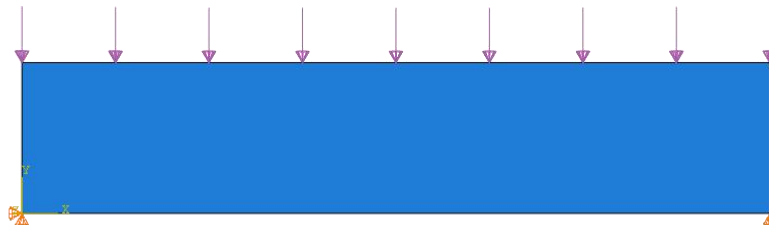


图 3 ABAQUS 模型的荷载布置与边界条件

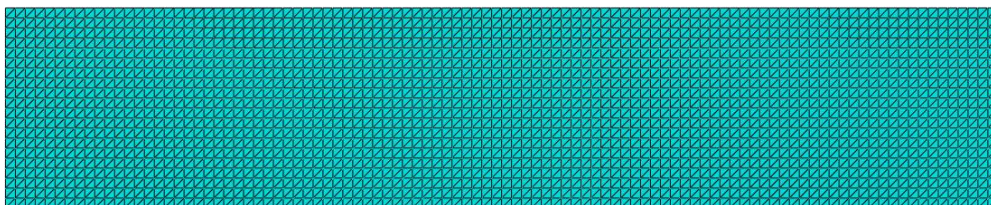
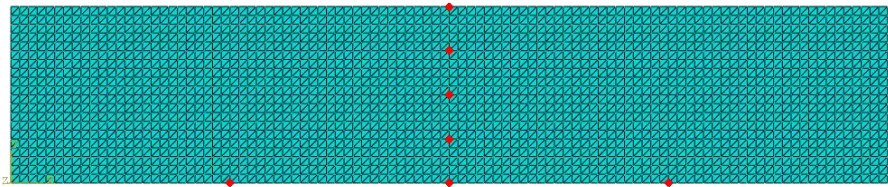


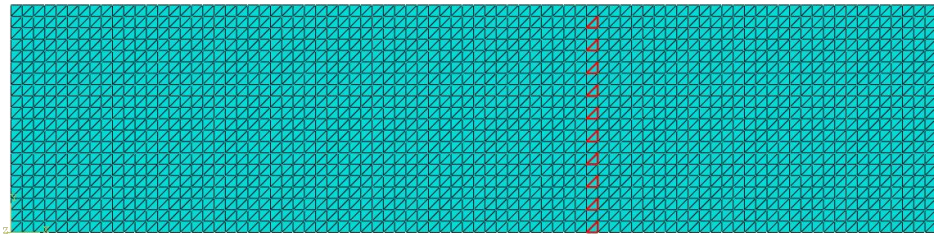
图 4 ABAQUS 模型网格划分

将 MATLAB 程序与有限元软件 ABAQUS 的结点位移、单元应力的计算结果进行对比，位移对比结点选取 26、51、76、556、1061、1566、2071 号结点，应力对比单元选取 103、

503、903、1303、1703、2103、2503、2903、3303、3703 号单元，如图 5 所示。



(a) 位移对比结点

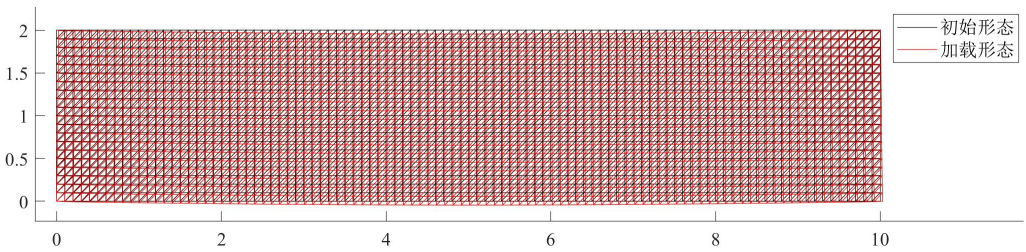


(b) 应力对比单元

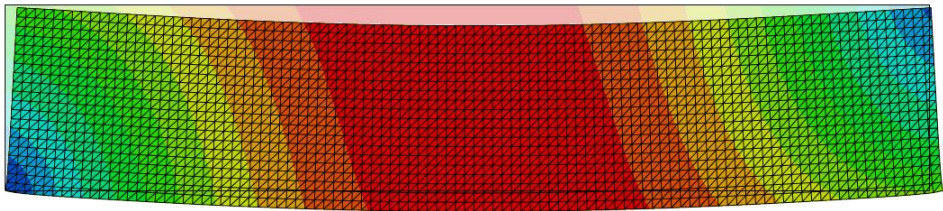
图 5 用于对比的结点位移与单元应力位置

2、结点位移对比

利用 MATLAB 程序与 ABAQUS 软件分别计算该简支梁的结点位移，结果如图 6 所示。MATLAB 计算结果图中黑色代表结构初始状态，红色代表结构受力后的位移状态，因结点位移数值较小，为绘图明显，将位移放大 20 倍。



(a) MATLAB 变形



(b) ABAQUS 变形

图 6 结点位移结果对比

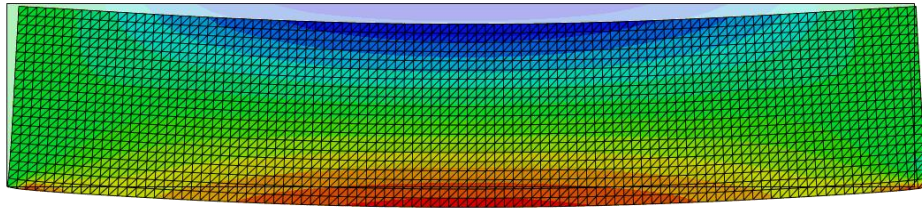
表 1 结点位移对比

位移 (mm)	ABAQUS		MATLAB		误差 (%)	
结点	X	Y	X	Y	X	Y
26	0.2987	-1.6720	0.3001	-1.6749	-0.4582	-0.1736
51	0.7089	-2.2415	0.7103	-2.2444	-0.1926	-0.1311
76	1.1191	-1.6803	1.1205	-1.6833	-0.1220	-0.1782
556	0.7097	-2.2617	0.7111	-2.2646	-0.1936	-0.1301
1061	0.7106	-2.2697	0.7119	-2.2727	-0.1945	-0.1295
1566	0.7114	-2.2664	0.7128	-2.2694	-0.1955	-0.1298
2071	0.7123	-2.2510	0.7137	-2.2539	-0.1964	-0.1307

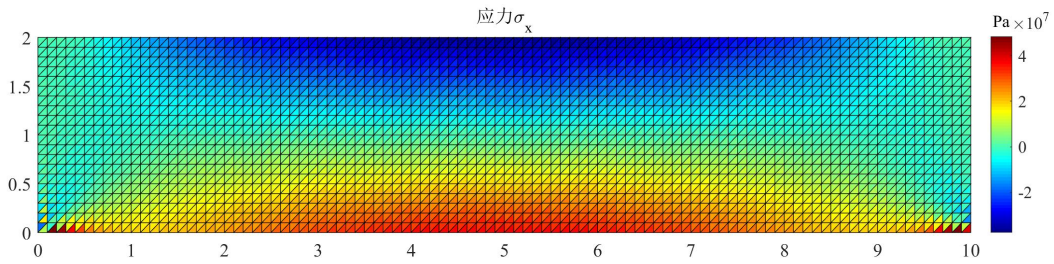
所选取结点的 X、Y 向位移以及误差的 MATLAB 程序与 ABAQUS 软件的计算结果如表 1 所示。由表可知，误差较小，在可以接受的范围之内，说明了 MATLAB 计算程序在结点位移计算方面的准确性。

3、单元应力对比

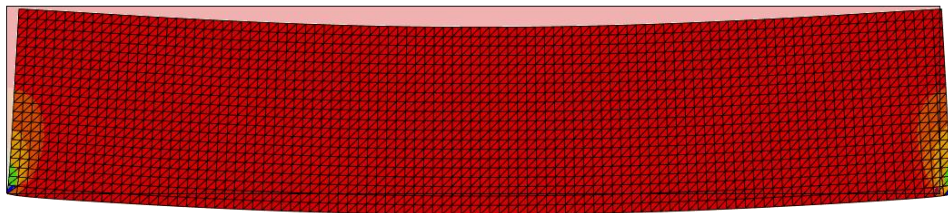
利用 MATLAB 程序及 ABAQUS 软件计算该简支梁施加均布荷载后各单元的受力结果，其应力云图如图 7 所示。



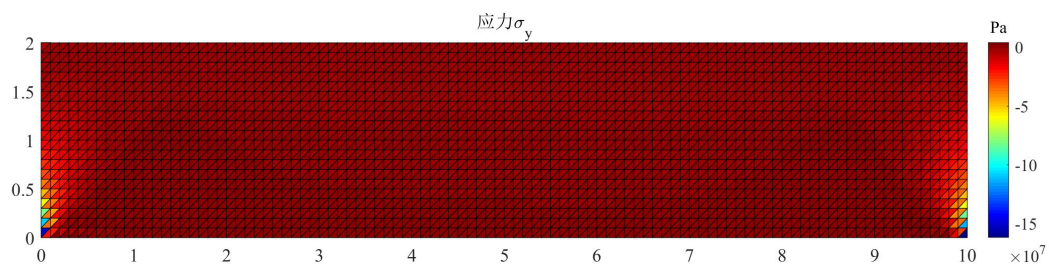
(a) ABAQUS X 向应力



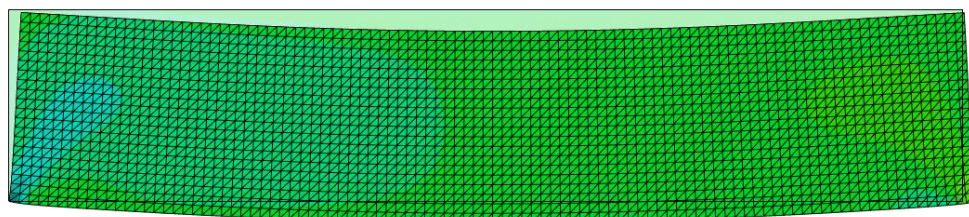
(b) MATLAB X 向应力



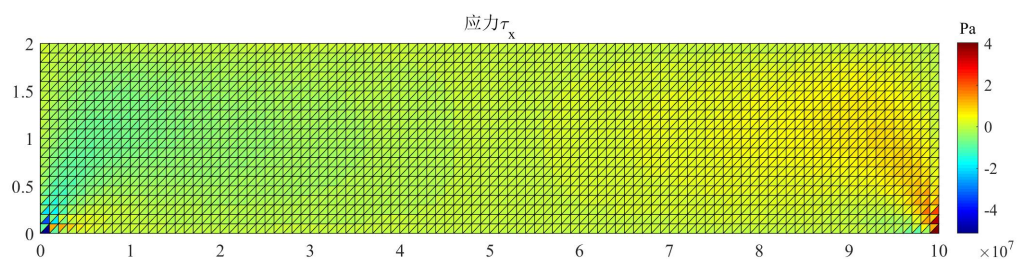
(c) ABAQUS Y 向应力



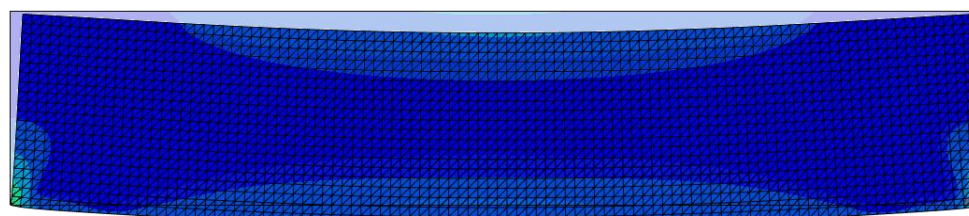
(d) MATLAB Y 向应力



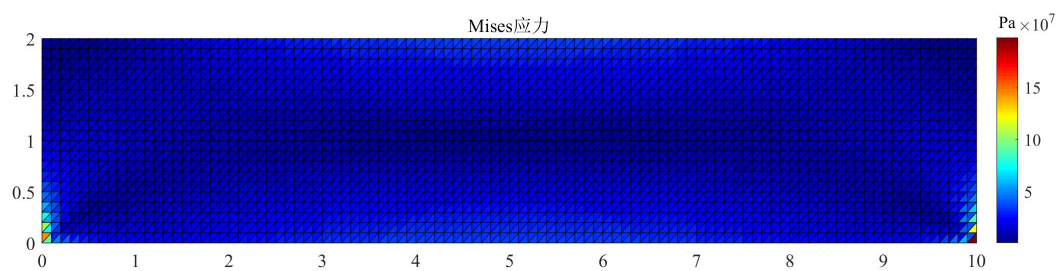
(e) ABAQUS 剪切应力



(f) MATLAB 剪切应力



(g) ABAQUS Mises 应力



(h) MATLAB Mises 应力

图 7 单元应力结果对比

表 2 单元应力对比

MISES 应力			
单元	ABAQUS	MATLAB	误差 (%)
103	37.4284	37.4283	0.0001
503	29.6726	29.6726	-0.0001
903	22.1544	22.1544	0.0002
1303	14.8165	14.8165	0.0001
1703	7.6126	7.6126	-0.0006
2103	0.9989	0.9989	-0.0034
2503	6.8565	6.8565	0.0005
2903	14.0615	14.0615	0.0000
3303	21.4078	21.4078	0.0002
3703	28.9377	28.9378	-0.0002

由图 7 可知，MATLAB 程序与 ABAQUS 软件计算得出的应力云图分布较为相似。所选取单元的 MISES 应力以及相对误差的 MATLAB 程序与 ABAQUS 软件的计算结果如表 2 所示，由表可知，误差极小，说明了 MATLAB 计算程序在单元应力计算方面的准确性。