有限元方法与应用课程作业

三角形单元程序设计与实例验证

学院：机械与运载工程学院

专业：机械工程

学号：S230200234

姓名：崔淑娟

班级：机械研 2303 班

2024 年 1 月

目 录

1、问题描述

一长宽为 150×100 mm 的平板左端宽边固定，上端与右端分别受到 41.3 N/mm 与 21 N/mm 均布载荷作用，中间直径为 30mm 的圆孔受到一个大小为 9400 N 的集中力作用，如图 1 所示。在进行仿真计算时尽可能将载荷平均分配到每个单元的节点上，首先使用 HyperMesh 有限元仿真软件和三角形单元 Matlab 程序分别计算节点位移 u，v 和节点应力 σ_x，σ_y，σ_{xy}，最后将 Matlab 程序计算结果与 HyperMesh 有限元仿真结果进行比较，验证 Matlab 程序的正确性。

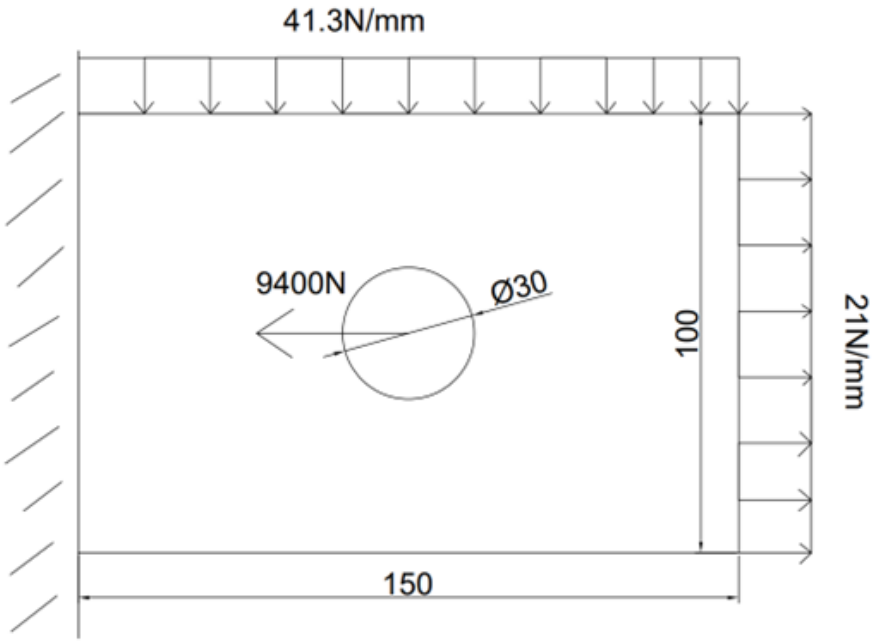

图 1 平面问题示意图

2、Hypermesh 有限元分析

2.1 建立有限元模型

该问题是一个典型的平面应力问题，可通过建立二维平面来进行分析。通过 HyperMesh 自带建模系统直接进行建模并划分网格，对于受载的位置需要对网格进行细化从而保证求解的精度，设计孔周网格大小为 2mm，其他网格大小为 5mm，共划分 1539 个三角形网格，843 个节点，如图 2 所示。

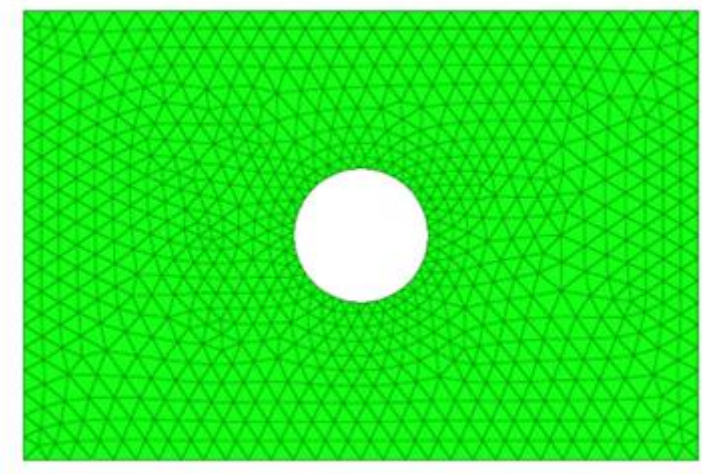

图 2 Hypermesh 网格划分示意图

网格模型建立好之后，对其进行参数设置：设置模型的弹性模量为 210000 Mpa，泊松比为 0.3，单元厚度为 2 mm，施加对对应的载荷及约束，建立好的有限元分析模型如图 3 所示。

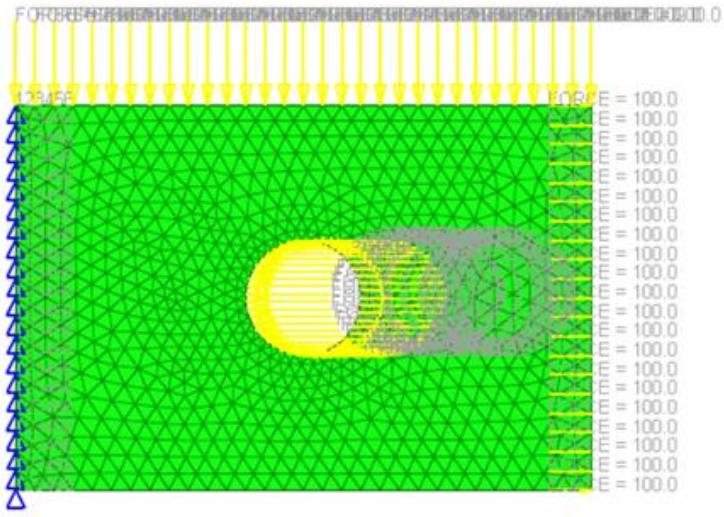

图 3 有限元分析模型

2.2 有限元结果分析

将设置好的有限元模型提交到 Optistruct 中进行静力学求解，计算完成后在 HyperView 窗口查看模型仿真分析结果，模型 X 方向与 Y 方向的位移云图分别对应图 4、图 5，且已在图中标志出该方向的最大位移和最小位移的节点位置；模型在单元积分点处的 X 方向应力云图、Y 方向应力云图、XY 方向剪切应力云图分别对应图 6、图 7、图 8，最大应力点与最小应力点也已在图中标出。

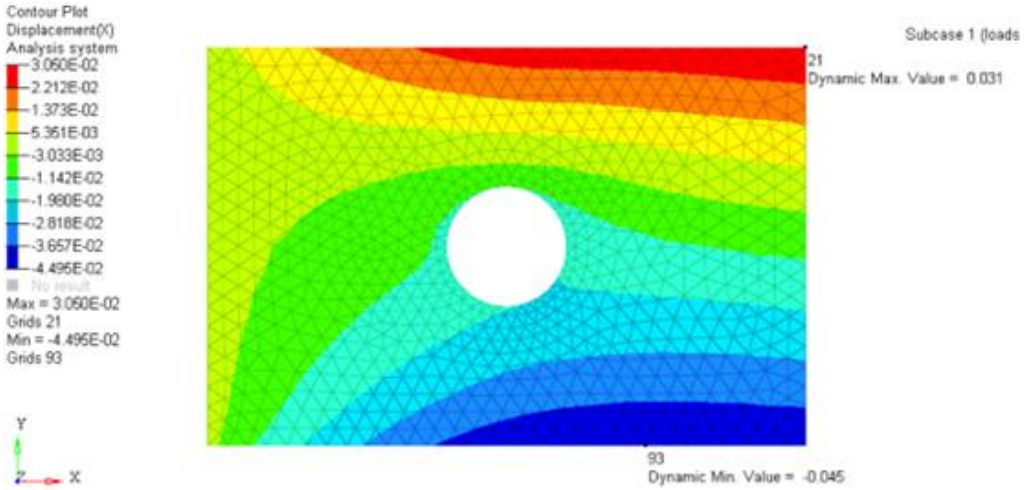

图 4 模型 X 方向位移云图

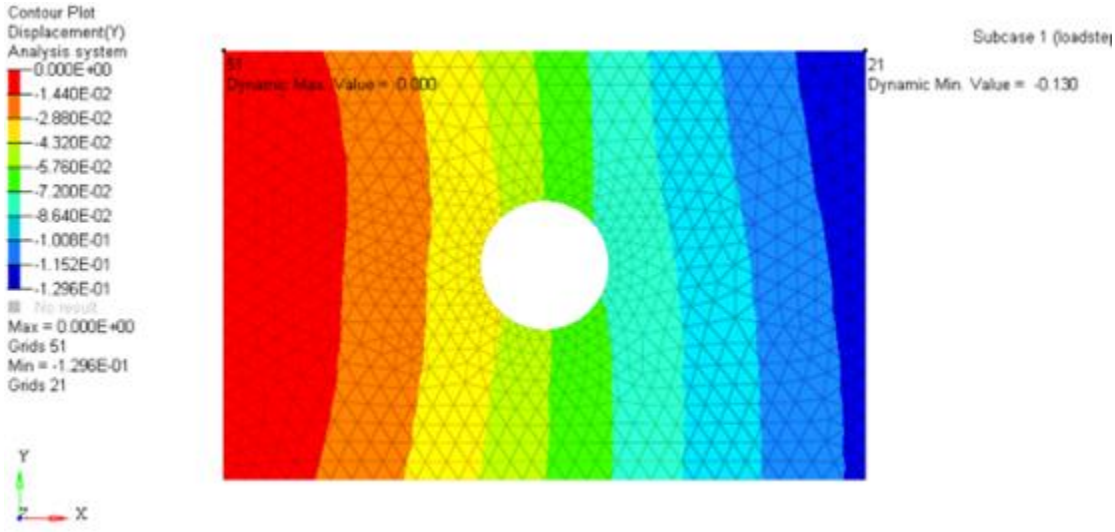

图 5 模型 Y 方向位移云图

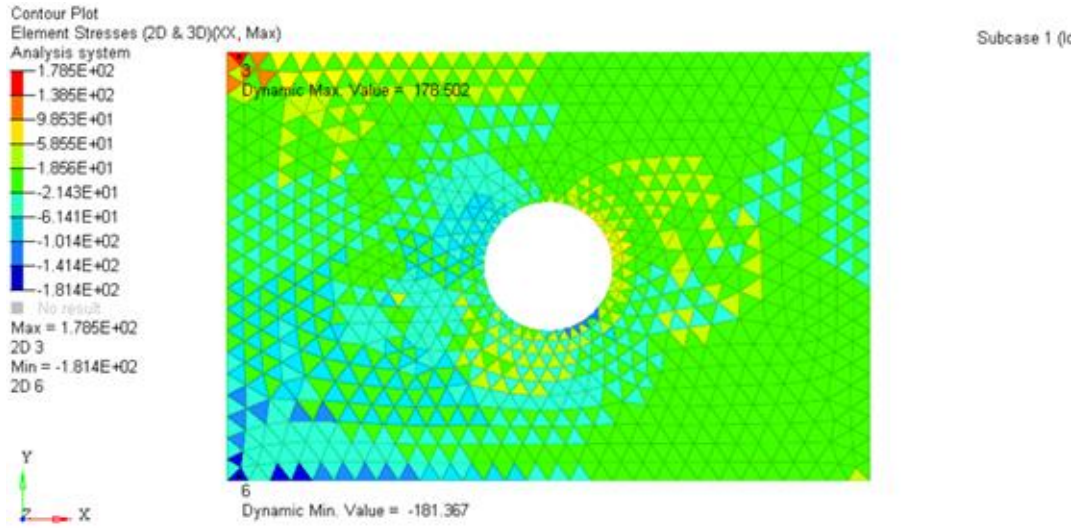

图 6 模型 X 方向应力云图

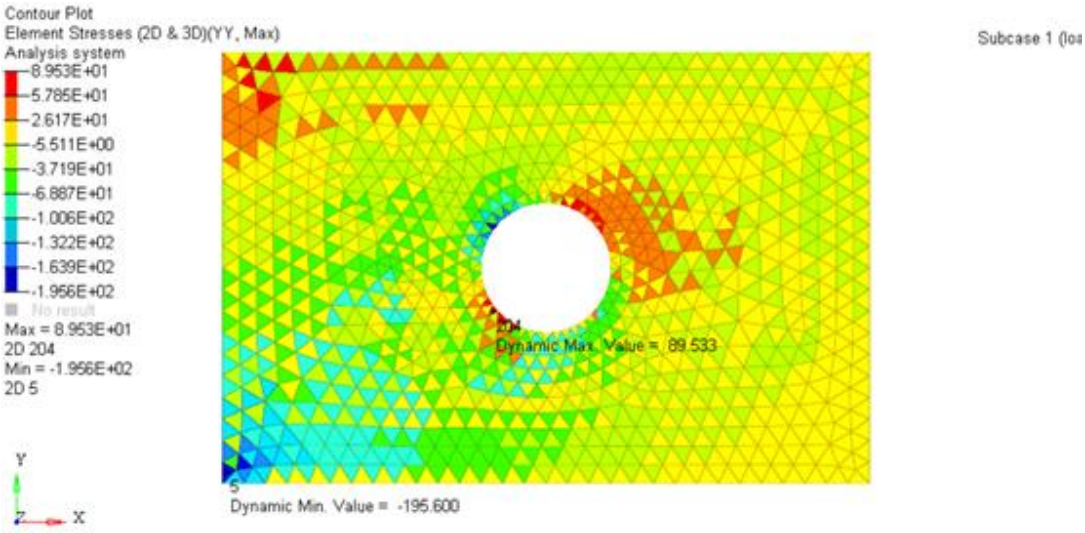

图 7 模型 Y 方向应力云图

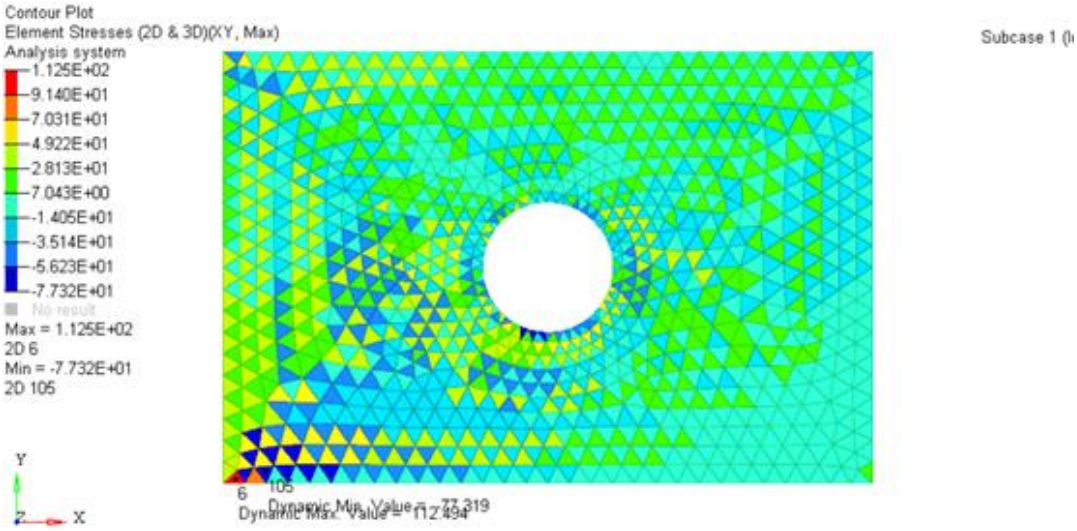

图 8 模型 XY 方向应力云图

3、Matlab 有限元程序编制

3.1 三角形单元有限元基础理论

有限元计算过程可以用过程框图来表达，如图 9 框图所示。

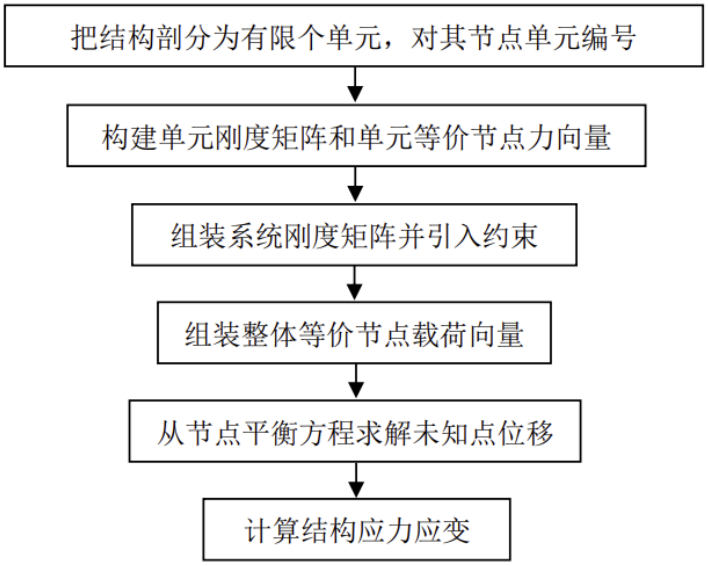

图 9 有限元计算过程框图

三角形任意一点的位移可表示为：

$$u(x,\ y) = \varphi_1 u_1 + \varphi_2 u_2 + \varphi_3 u_3$$

$$v\ (x,\ y\ ) = \varphi_1 v_1 + \varphi_2 v_2 + \varphi_3 v_3$$

$$\varphi = \begin{bmatrix} \varphi_1 & 0 & \varphi_2 & 0 & \varphi_3 & 0 \\ 0 & \varphi_1 & 0 & \varphi & 0 & \varphi_3 \end{bmatrix}$$

$$\varphi_1 = \frac{1}{2A}(a_1 + b_1 x + c_1 y)$$

其中φ为三角形形函数，A 为三角形单元面积。

单元应力为:

$$\sigma = DBd^e$$

三角形单元刚度矩阵为：

$$K^e = \int_{\emptyset^e} B^T DBd\emptyset = B^T DBA^e$$

系统刚度矩阵为各单元矩阵集：

$$K = \sum_{e=1}^{N_{elem}} K^e$$

平衡方程为:

$$f = \sum_{e=1}^{N_{elem}} f^e$$

3.2 Matlab 程序设计说明

在使用 Hypermesh 软件划分网格时，可以显示每个单元的编号及每个单元对应节点的编号，如图 10 所示，将单元编号以及单元对应节点编号信息导出为 k 文件，使用记事本打开获取到各个节点所属的坐标值、单元的连通性、约束信息和载荷信息，将这些数据按顺序存入新的 txt 文本文件中，命名为 input.txt,如图 11 所示，作为后续 Matlab 编程的输入参数。

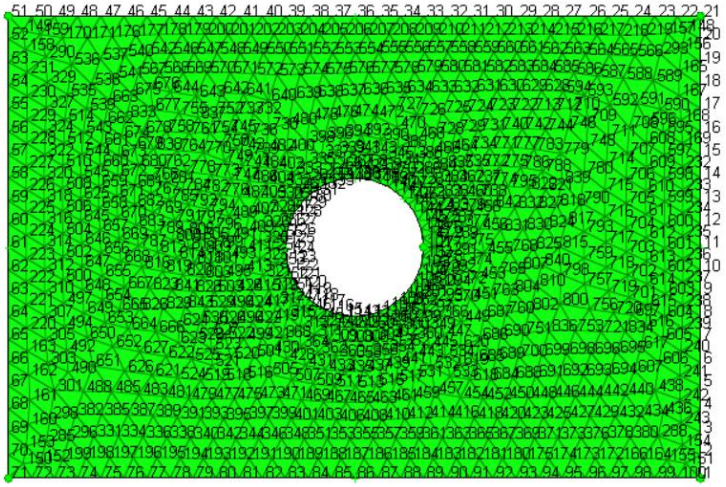

图 10 单元编号信息

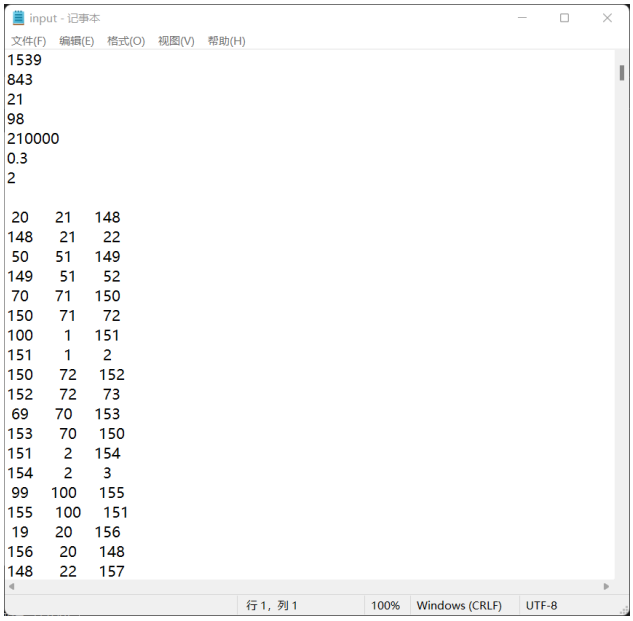

图 11 input.txt 文件内容

解决此次作业题目的 Matlab 程序如下所示（文件夹中含有 m 程序文件和需导入的数据 input.txt 文件），首先在程序前端定义好题目的具体参数如弹性模量、泊松比、厚度等，然后编制总体刚度矩阵、应变矩阵等程序对题目进行对应求解，得到有限元分析结果。

```
clear all;
first_time=cputime;
format short e %  设定输出类型
fprintf=fopen('input.txt ','rt'); %  打开输入数据文件，读入参数数据
nelement=fscanf(fprintf,'%d',1);%  单元个数
npiont=fscanf(fprintf,'%d',1);%  结点个数
nbccondit=fscanf(fprintf,'%d',1) %  受约束边界点数
nforce=fscanf(fprintf,'%d',1);%  受力节点个数
young=fscanf(fprintf,'%e',1);%  弹性模量
poission=fscanf(fprintf,'%f',1);%  泊松比
thickness=fscanf(fprintf,'%f',1);%  厚度
nodes=fscanf(fprintf,'%d',[3,nelement])';%  单元定义数组（单元结点号）
ncoordinates=fscanf(fprintf,'%f',[2,npiont])'; %  结点坐标数组
```

```
force=fscanf(fprintf,'%f',[3,nforce])'; % 结点力数组（受力结点编号 , x 方向 ,y 方向）
fc=fopen('constraint.txt ','rt');
constraint=fscanf(fc,'%d',[3,nbccondit])'; % 约束信息（约束点， x 约束， y 约束）%有约束为 1，无约束为 0
kk=zeros(2*npiont,2*npiont); % 生成特定大小总体刚度矩阵并置 0
for i=1:nelement
      D= [1 poission 0;
 poission 1 0;
 0 0 (1-poission)/2]*young/(1-poission^2) %生成弹性矩阵 D
A=det([1 ncoordinates(nodes(i,1),1) ncoordinates(nodes(i,1),2);
 1 ncoordinates(nodes(i,2),1) ncoordinates(nodes(i,2),2);
 1 ncoordinates(nodes(i,3),1) ncoordinates(nodes(i,3),2)])/2 %计算当前单元的面积A
for j=0:2
b(j+1)=ncoordinates(nodes(i,(rem((j+1),3))+1) ,2)-ncoordinates(nodes(i,(rem((j+2),3))+1),2);
c(j+1)=-ncoordinates(nodes(i,(rem((j+1),3))+1),1)+ncoordinates(nodes(i,(rem((j+2),3))+1),1);
end
 B=[b(1) 0 b(2) 0 b(3) 0;
 0 c(1) 0 c(2) 0 c(3);
 c(1) b(1) c(2) b(2) c(3) b(3)]/(2*A); %生成应变矩阵 B
B1( :,:,i)=B;
S=D*B;%求应力矩阵S
nk=B'*S*thickness*A; % 求解单元刚度矩阵
 a=nodes(i,:); % 临时向量,用来记录当前单元的节点编号
 for j=1:3
 for k=1:3
 kk((a(j)*2-1):a(j)*2,(a(k)*2-1):a(k)*2)=kk((a(j)*2-1):a(j)*2,(a(k)*2-1):a(k)*2)+nk(j*2-
1:j*2,k*2-1:k*2);
 % 根据节点编号对应关系将单元刚度分块叠加到总刚度矩阵中
```

```
  end
  end
  end
%**************************************************************************
%将约束信息加入总体刚度矩阵（对角元素改一法）
  for i=1:nbccondit
  if constraint(i,2)==1
  kk(:,(constraint(i,1)*2-1))=0; %  一列为零
  kk((constraint(i,1)*2-1),:)=0; %  一行为零
  kk((constraint(i,1)*2-1),(constraint(i,1)*2-1))=1; %  对角元素为  1
  end
if constraint(i,3)==1
  kk( :,constraint(i,1)*2)=0; %  一列为零
  kk(constraint(i,1)*2,:)=0; %  一行为零
  kk(constraint(i,1)*2 ,constraint(i,1)*2)=1; %  对角元素为  1
  end
  end
%**************************************************************************
%生成荷载向量
  loadvector(1:2*npiont)=0; %  总体荷载向量置零
  for i=1:nforce
  loadvector((force(i,1)*2-1):force(i,1)*2)=force(i,2:3);
  end
  %**************************************************************************
%求解内力
  displancement=kk\loadvector' %  计算节点位移向量
  edisplancement(1:6)=0; %  当前单元节点位移向量
  for i=1:nelement
```

```
 for j=1:3
 edisplancement(j*2-1:j*2)=displancement(nodes(i,j)*2-1:nodes(i,j)*2);
 % 取出当前单元的节点位移向量
 end
 i ;
 stress=D*B1(:, :, i)*edisplancement'; % 求内力
 stress1(i,:)=[stress'];
 stress_x(i)=stress1(i,1);
 stress_y(i)=stress1(i,2);
 stress_xy(i)=stress1(i,3);
 sigma1(i)=0.5*(stress_x(i)+stress_y(i))+sqrt((stress_xy(i))^2+(0.5*(stress_x(i)-stress_y(i)))^2);
 sigma2(i)=0.5*(stress_x(i)+stress_y(i))-sqrt((stress_xy(i))^2+(0.5*(stress_x(i)-stress_y(i)))^2);
 stress_vonmises(i)=sqrt(0.5*((sigma1(i)-sigma2(i))^2+(sigma1(i)-0)^2+(0-sigma2(i))^2));
 dlmwrite('stress_vonmises.txt',stress_vonmises);
 dlmwrite('stress.txt',stress1);
 dlmwrite('displancement.txt',displancement);
 end
set(0,'defaultfigurecolor','w')
%画vonmiss应力云图
s1=max(stress_vonmises);%求出最大应力
s2=min(stress_vonmises);%求出最小应力
a=(s1-s2)/9;%将应力分成9份
stress_range=zeros(1,10);
stress_range(1)=s1;%stress_range(1)为最大应力
for i=2:10
    stress_range(i)=stress_range(i-1)-a;
end
range=stress_range;
```

```
figure(1);
color=jet(9);%将彩虹色分成9份
for i=1:size(nodes)
    ElementCoodinate=[ncoordinates(nodes(i,1),:)
                      ncoordinates(nodes(i,2),:)
                      ncoordinates(nodes(i,3),:)];
    x=ElementCoodinate(:,1);
    y=ElementCoodinate(:,2);
    s=stress_vonmises(i);
    %将应力大小与颜色对应
    if (range(1)>=s)&&(s>range(2))
        ColorSpec=color(9,:);
    elseif (range(2)>=s)&&(s>range(3))
        ColorSpec=color(8,:);
    elseif (range(3)>=s)&&(s>range(4))
        ColorSpec=color(7,:);
    elseif (range(4)>=s)&&(s>range(5))
        ColorSpec=color(6,:);
    elseif (range(5)>=s)&&(s>range(6))
        ColorSpec=color(5,:);
    elseif (range(6)>=s)&&(s>range(7))
        ColorSpec=color(4,:);
    elseif (range(7)>=s)&&(s>range(8))
        ColorSpec=color(3,:);
    elseif (range(8)>=s)&&(s>range(9))
        ColorSpec=color(2,:);
    else
        ColorSpec=color(1,:);
```

```
    end
    fill(x,y,ColorSpec);%画出单元应力对应的颜色
    hold on
end
%  画应力云图标签
range=sort(range);
colormap(color);
c=colorbar;
c.TickLabels=(range);
c.Ticks=[0,1/9,2/9,3/9,4/9,5/9,6/9,7/9,8/9,9/9];
axis equal;
title('vonmiss应力云图');

%画sigmay应力云图
s1=max(stress_y);%求出最大应力
s2=min(stress_y);%求出最小应力
a=(s1-s2)/9;%将应力分成9份
stress_range=zeros(1,10);
stress_range(1)=s1;%stress_range(1)为最大应力
for i=2:10
    stress_range(i)=stress_range(i-1)-a;
end
range=stress_range;
figure(2);
color=jet(9);%将彩虹色分成9份
for i=1:size(nodes)
    ElementCoodinate=[ncoordinates(nodes(i,1),:)
                      ncoordinates(nodes(i,2),:)
```

```
                                ncoordinates(nodes(i,3),:)];
    x=ElementCoodinate(:,1);
    y=ElementCoodinate(:,2);
    s=stress_y(i);
    %将应力大小与颜色对应
    if (range(1)>=s)&&(s>range(2))
        ColorSpec=color(9,:);
    elseif (range(2)>=s)&&(s>range(3))
        ColorSpec=color(8,:);
    elseif (range(3)>=s)&&(s>range(4))
        ColorSpec=color(7,:);
    elseif (range(4)>=s)&&(s>range(5))
        ColorSpec=color(6,:);
    elseif (range(5)>=s)&&(s>range(6))
        ColorSpec=color(5,:);
    elseif (range(6)>=s)&&(s>range(7))
        ColorSpec=color(4,:);
    elseif (range(7)>=s)&&(s>range(8))
        ColorSpec=color(3,:);
    elseif (range(8)>=s)&&(s>range(9))
        ColorSpec=color(2,:);
    else
        ColorSpec=color(1,:);
    end
    fill(x,y,ColorSpec);%画出单元应力对应的颜色
    hold on
end
%  画应力云图标签
```

```
range=sort(range);
colormap(color);
c=colorbar;
c.TickLabels=(range);
c.Ticks=[0,1/9,2/9,3/9,4/9,5/9,6/9,7/9,8/9,9/9];
axis equal;
title('sigmay应力云图');

%画sigmax应力云图
s1=max(stress_x);%求出最大应力
s2=min(stress_x);%求出最小应力
a=(s1-s2)/9;%将应力分成9份
stress_range=zeros(1,10);
stress_range(1)=s1;%stress_range(1)为最大应力
for i=2:10
    stress_range(i)=stress_range(i-1)-a;
end
range=stress_range;
figure(3);
color=jet(9);%将彩虹色分成9份
for i=1:size(nodes)
    ElementCoodinate=[ncoordinates(nodes(i,1),:)
                      ncoordinates(nodes(i,2),:)
                      ncoordinates(nodes(i,3),:)];
    x=ElementCoodinate(:,1);
    y=ElementCoodinate(:,2);
    s=stress_x(i);
    %将应力大小与颜色对应
```

```
    if (range(1)>=s)&&(s>range(2))
        ColorSpec=color(9,:);
    elseif (range(2)>=s)&&(s>range(3))
        ColorSpec=color(8,:);
    elseif (range(3)>=s)&&(s>range(4))
        ColorSpec=color(7,:);
    elseif (range(4)>=s)&&(s>range(5))
        ColorSpec=color(6,:);
    elseif (range(5)>=s)&&(s>range(6))
        ColorSpec=color(5,:);
    elseif (range(6)>=s)&&(s>range(7))
        ColorSpec=color(4,:);
    elseif (range(7)>=s)&&(s>range(8))
        ColorSpec=color(3,:);
    elseif (range(8)>=s)&&(s>range(9))
        ColorSpec=color(2,:);
    else
        ColorSpec=color(1,:);
    end
    fill(x,y,ColorSpec);%画出单元应力对应的颜色
    hold on
end
% 画应力云图标签
range=sort(range);
colormap(color);
c=colorbar;
c.TickLabels=(range);
c.Ticks=[0,1/9,2/9,3/9,4/9,5/9,6/9,7/9,8/9,9/9];
```

```
axis equal;
title('sigmax应力云图');

%画sigmaxy应力云图
s1=max(stress_xy);%求出最大应力
s2=min(stress_xy);%求出最小应力
a=(s1-s2)/9;%将应力分成9份
stress_range=zeros(1,10);
stress_range(1)=s1;%stress_range(1)为最大应力
for i=2:10
    stress_range(i)=stress_range(i-1)-a;
end
range=stress_range;
figure(4);
color=jet(9);%将彩虹色分成9份
for i=1:size(nodes)
    ElementCoodinate=[ncoordinates(nodes(i,1),:)
                      ncoordinates(nodes(i,2),:)
                      ncoordinates(nodes(i,3),:)];
    x=ElementCoodinate(:,1);
    y=ElementCoodinate(:,2);
    s=stress_xy(i);
    %将应力大小与颜色对应
    if (range(1)>=s)&&(s>range(2))
        ColorSpec=color(9,:);
    elseif (range(2)>=s)&&(s>range(3))
        ColorSpec=color(8,:);
    elseif (range(3)>=s)&&(s>range(4))
```

```
            ColorSpec=color(7,:);
        elseif (range(4)>=s)&&(s>range(5))
            ColorSpec=color(6,:);
        elseif (range(5)>=s)&&(s>range(6))
            ColorSpec=color(5,:);
        elseif (range(6)>=s)&&(s>range(7))
            ColorSpec=color(4,:);
        elseif (range(7)>=s)&&(s>range(8))
            ColorSpec=color(3,:);
        elseif (range(8)>=s)&&(s>range(9))
            ColorSpec=color(2,:);
        else
            ColorSpec=color(1,:);
        end
        fill(x,y,ColorSpec);%画出单元应力对应的颜色
        hold on
end
%  画应力云图标签
range=sort(range);
colormap(color);
c=colorbar;
c.TickLabels=(range);
c.Ticks=[0,1/9,2/9,3/9,4/9,5/9,6/9,7/9,8/9,9/9];
axis equal;
title('sigmaxy应力云图');

fclose(fc); %  关闭数据文件
fclose(fprintf); %  关闭数据文件
```

3.3Matlab 程序输出结果

Matlab 输出的所有节点位移与应力结果在附件中的“displancement.txt”和“stress.txt”文件。在位移结果“displancement.txt”中，从上往下按节点顺序依次是从1号节点 到843号节点的X向位移与Y向位移。在应力结果文件“stress.txt”中，第一列为 X 方向应力，第二列为 Y 方向应力，第三列为 XY 平面切应力。为了方便与 HyperMesh 分析的结果进行对比，在有限元程序中已编写相应的绘图程序段绘制了模型的 Vonmiss 应力云图、X 向应力云图、Y 向应力云图以及 XY 平面切应力云图，如图 12、13、14、15 所示。

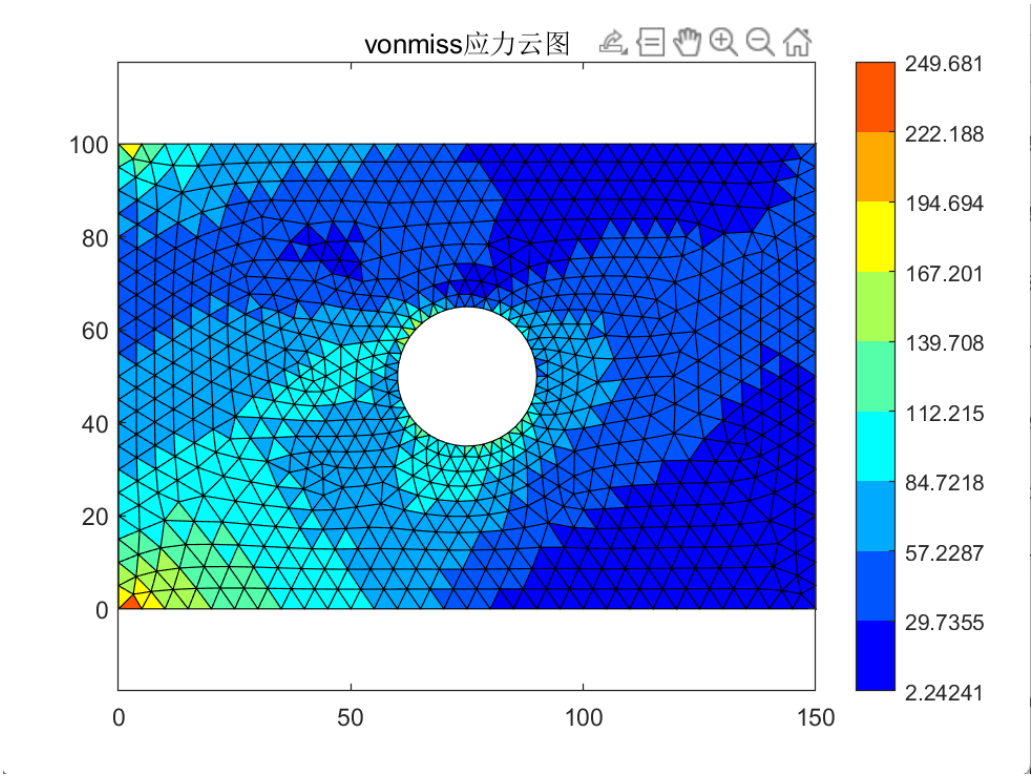

图 12 vonmiss stress 云图

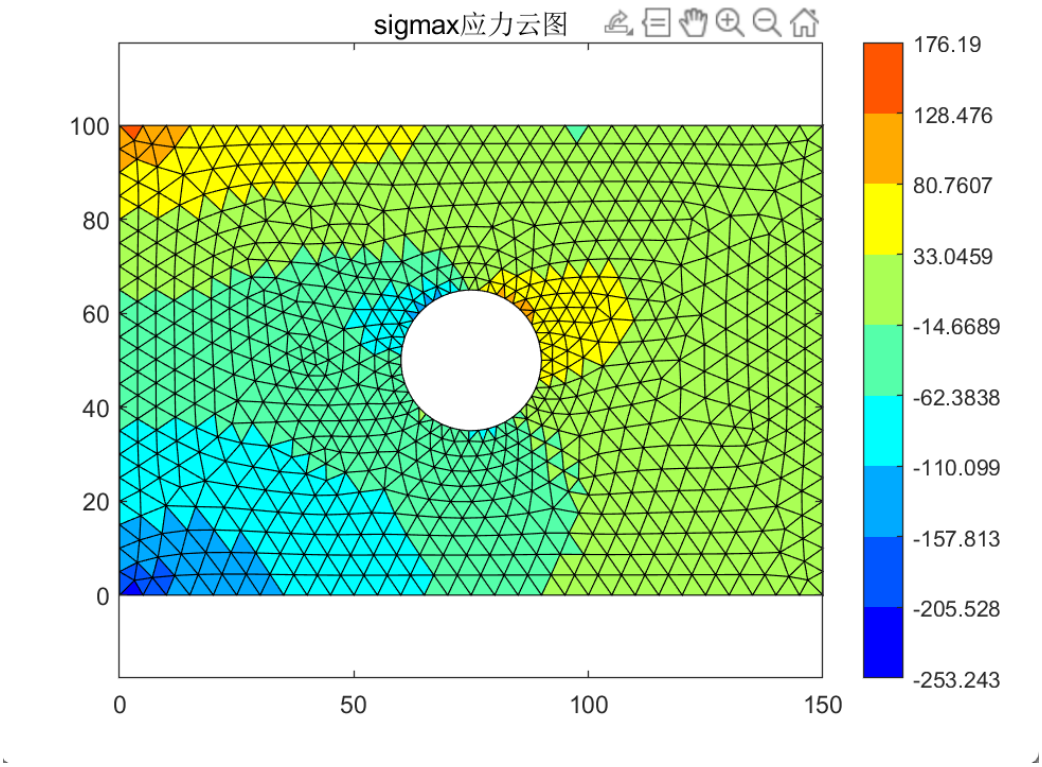

图 13 sigma_x stress 云图

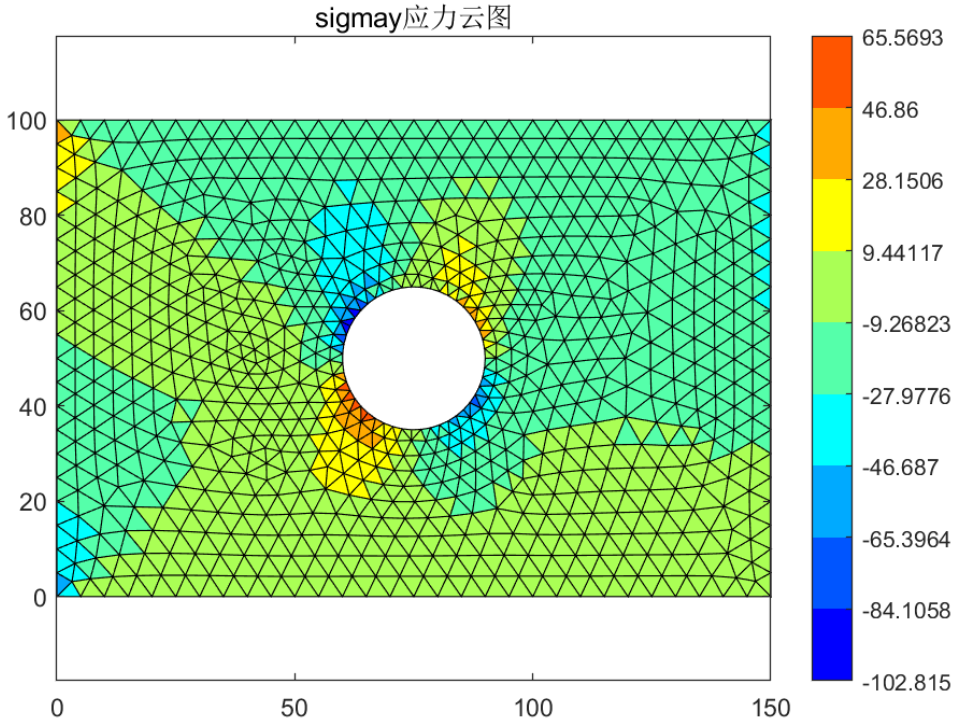

图 14 sigma_y stress 云图

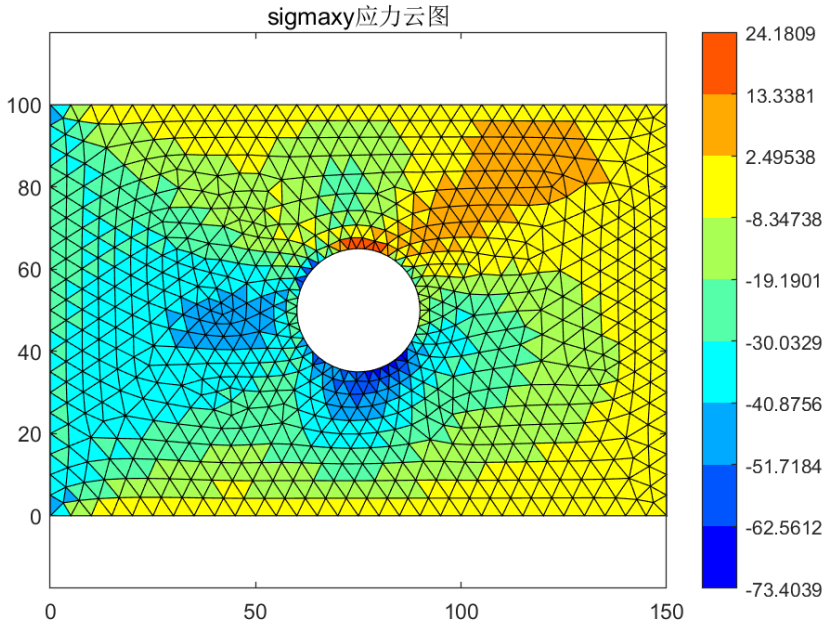

图 15 sigma_xy stress 云图

4、HyperMesh 与 Matlab 计算结果对比与分析

为了验证所编写的 Matlab 程序的正确性，与之前 HyperMesh 软件计算的结果进行对比。在 Matlab 中，计算出来 X 方向的最大位移为：2.7881e-02，最小位移为：-4.4504e-02；Y 方向的最大位移为：0，最小位移为：-1.2725e-01。而在 HyperMesh 中计算出来 X 方向的最大位移为：3.050e-02，最小位移为：-4.495e-02；Y 方向的最大位移为：0，最小位移为：-1.296e-01，此结果由之前的位移云图也可以观察到。HyperMesh 与 Matlab 计算的结果对比如表 1 所示。从表 1 可以看出，在 HyperMesh 中计算出来的 X 方向最大位移与 Y 方向的最大位移都分别约等于与 Matlab 计算的 X 方向最大位移与 Y 方向的最大位移，两者误差很小。

表 1 位移对比表

	X 方向最大位移	Y 方向最大位移
Matlab 结果	2.7881e-02	0
HyperMesh 结果	3.050e-02	0
误差	8.59%	0

	X 方向最小位移	Y 方向最小位移
Matlab 结果	-4.4504e-02	-1.2725e-01
HyperMesh 结果	-4.495e-02	-1.296e-01
误差	0.99%	1.81%

此外，我们还可以从 Vonmiss 应力云图来看，更为直观地看出应力的分布状况基本一致，如图 16、17 所示，证明所编写程序是正确的。

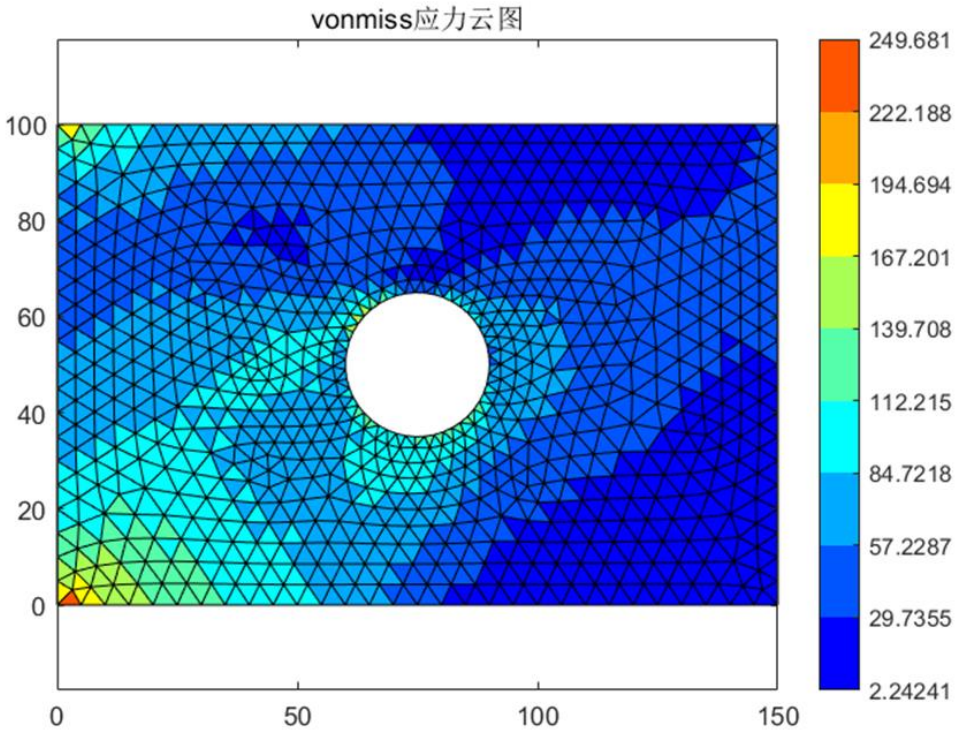

图 16 Matlab 程序运算后 Vonmiss 云图

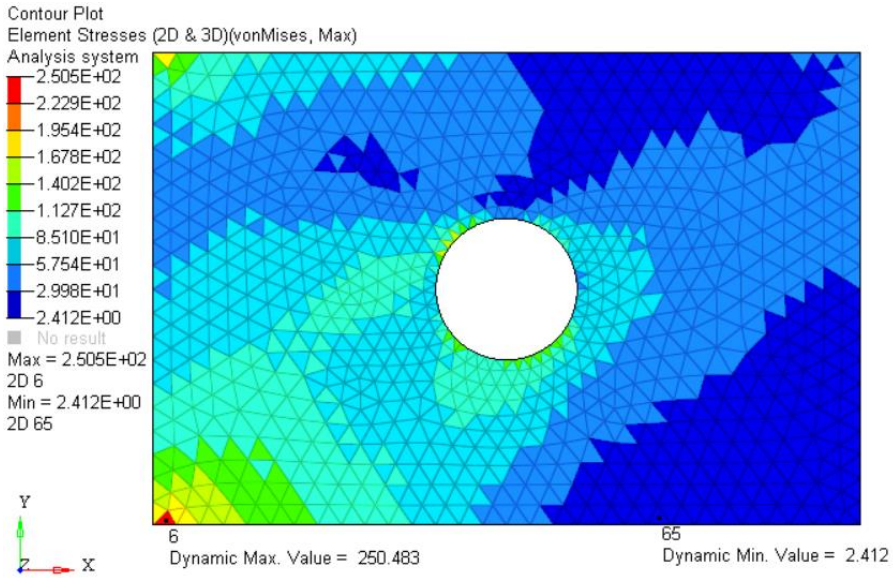

图 17 HyperMesh 分析的 Vonmiss 云图