

湖南大学

HUNAN UNIVERSITY

课程论文



论文题目	基于六面体单元网格的 凹型板材有限元分析
学生姓名	雷锐
学生学号	S230200246
学院名称	机械与运载工程学院
课程名称	有限元方法及应用
课程老师	王琥

2024 年 2 月 24 日

目 录

第 1 章 分析模型的建立	1
第 2 章 有限元编程分析	3
2.1 前处理过程	3
2.2 载荷及材料属性加载过程	3
2.3 刚度矩阵的建立	4
2.4 加载边界条件及求解	4
2.5 输出结果到 plt 文件中	5
第 3 章 结果输出及对比	6

第 1 章 分析模型的建立

利用 abaqus 进行建模并划分网格，并通过 inp 文件导出节点坐标存在 getcrood 子程序里面。

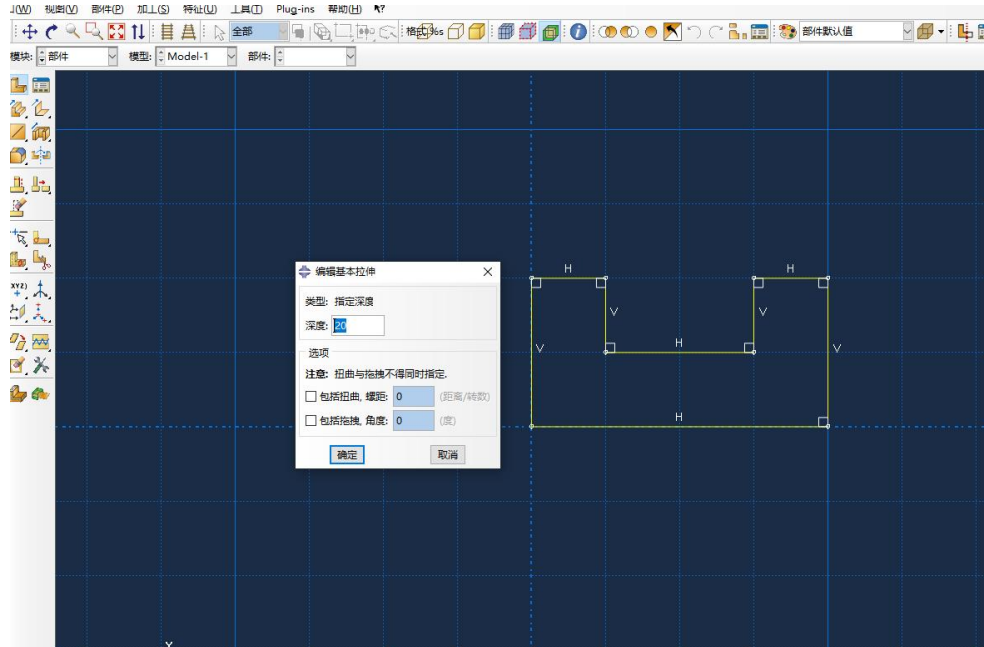


图 1.1 abaqus 建模

通过 abaqus 划分为六面体单元网格。

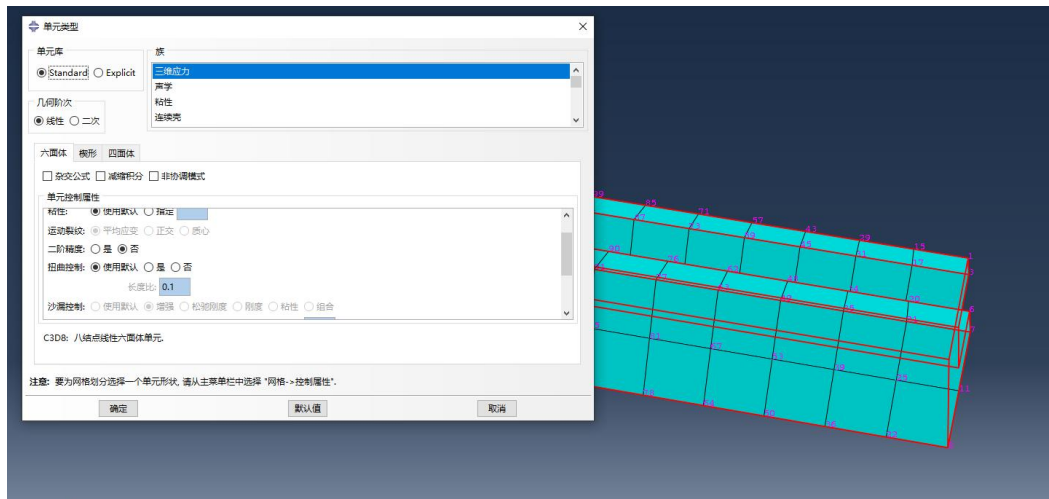


图 1.2 六面体单元网格划分

通过 abaqus 写入输入文件，生产节点坐标信息和节点编号信息。

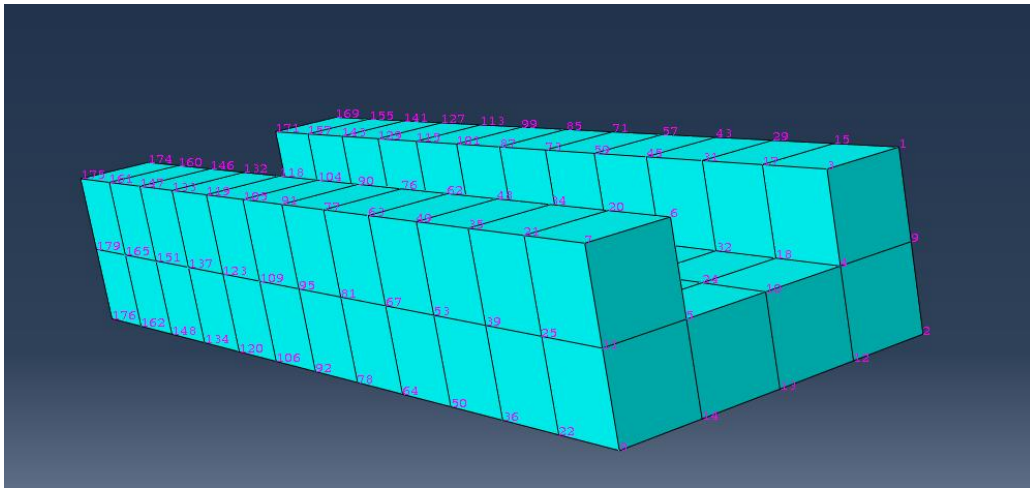


图 1.3 节点编号信息

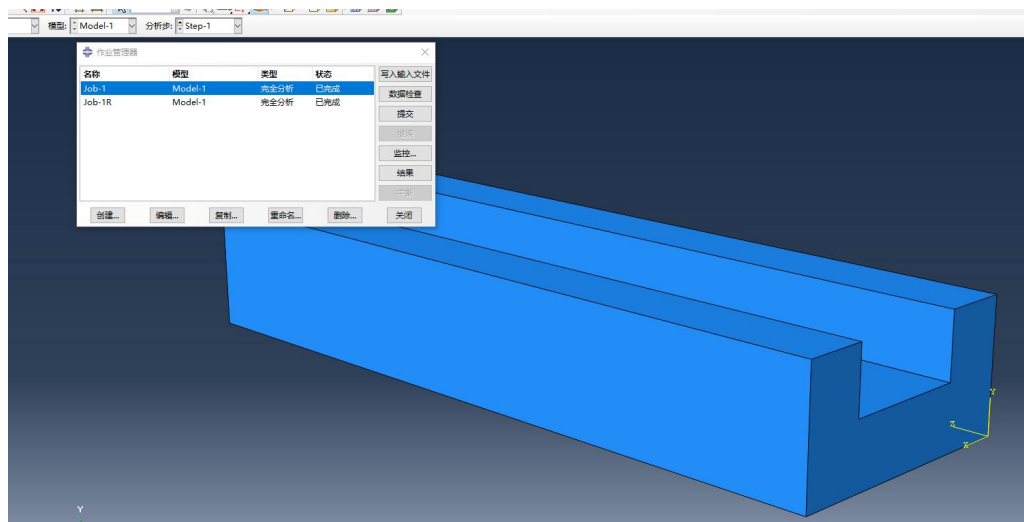


图 1.4 写入输入文件

第 2 章 有限元编程分析

2.1 前处理过程

此过程将程序所有变量初始化，将所有变量清零。以防上次计算结果影响本次程序的运行。

```
%% 前处理

%获取模型节点坐标值及坐标编号
[goords,nodes]=getcoord();
% 总单元数
nel=length(nodes(:,1));
%总节点数
nnode=length(goords(:,1))
%系统总自由度
sdof=3*length(goords(:,1));
%单元自由度
edof=24;
%初始化刚度、位移、载荷矩阵
KE=[edof,edof]; %单元刚度矩阵
K=sparse(sdof,sdof); %定义系统刚度矩阵
F=sparse(sdof,1); %定义力矩阵
disp=zeros(sdof,1); %定义位移矩阵 l=8; 每个元素的节点数
```

2.2 载荷及材料属性加载过程

本过程采用人机交互的形式，根据用户需求设置算例材料属性并设置施加载荷大小与数量。

```
%% 手动设立条件

%定义材料
E=input('请输入杨氏模量 E=\n');
P=input('请输入泊松比 P=\n');

%定义集中载荷数目
i=input('请输入需要施加的载荷数目=\n');

%循环读取载荷位置及大小，为该点自由度编号
for j=1:i
```

```

b=input('请输入需要施加的载荷编号=\n');
F(b,1)=input('请输入需要施加的载荷大小=\n');

end

```

2.3 刚度矩阵的建立

其中应变矩阵 **B** 矩阵以及本构矩阵 **D** 矩阵由子函数 **BDcalc** 计算完成。

```

%% 刚度矩阵的建立

for j=1:nel
    % 一次取一个单元的编号
    dex=[];
    dex=[nodes(j,2:9)] ;

    % 该单元节点坐标
    XYZ=[];
    for k=1:8
        XYZ(k,1)=goords(dex(1,k),2);
        XYZ(k,2)=goords(dex(1,k),3);
        XYZ(k,3)=goords(dex(1,k),4);
    end
    [KE,B,D]=Stiffnesske(E,P,XYZ); %计算应变矩阵

    index=[];
    for i=1:8
        index=[index 3*dex(1,i)-2 3*dex(1,i)-1 3*dex(1,i)];
    end

    K(index,index)=K(index,index)+KE;%并组装刚度矩阵
end

```

2.4 加载边界条件及求解

输入边界条件，修改刚度矩阵和载荷矩阵，求解出位移矩阵。

```

%% 边界条件及求解模块
%输入边界条件
alldofs = [1:sdof];
bcval=[169,170,171,172,173,174,175,176,177,178,179,180,181,182
];
c=length(bcval);
fixeddofs=[];

```

```

for i=1:c
    fixeddofs=[fixeddofs 3*bcval(i)-2,3*bcval(i)-1,3*bcval(i)];
end
n=length(fixeddofs);
% 根据置一法修改刚度矩阵和载荷矩阵
for i=1:n
    c= fixeddofs(i);
    for j=1:sdof
        K(c,j)=0;
    end
    K(c,c)=1;
    F(c)=0;
end

%求解位移矩阵
disp= K \ F;

```

2.5 输出结果到 plt 文件中

将最后的运算结果保存到plt文件中。

```

%% 后处理模块
%创建“456.plt”文件并将输出结果寄存再该文件中
fid_out=fopen('result .plt','w');
fprintf(fid_out,'TITLE="test case governed by poisson
equation"\n');
fprintf(fid_out,'VARIABLES="x" "y" "z" "u" "v" "w" \n');
fprintf(fid_out,'ZONE T="flow-field", N= %8d,E=%8d,ET=BRICK,
F=FEPOINT\n',nnode,nel);
for i=1:nnode
    fprintf(fid_out,'%16.6e%16.6e%16.6e%16.6e%16.6e%16.6e\n',
goords(i,2), goords(i,3),goords(i,4), disp(3*i-2,1)+0,
disp(3*i-1,1)+0,disp(3*i,1)+0);
end
for i=1:nel

fprintf(fid_out,'%8d%8d%8d%8d%8d%8d%8d%8d\n',nodes(i,2),nodes(
i,3),nodes(i,4),nodes(i,5),nodes(i,6),nodes(i,7),nodes(i,8),no
des(i,9));
end

```

第 3 章 结果输出及对比

对模型离散划分网格，使用 `tecplot` 对 `plt` 格式文件做后处理，得到仿真结果，并将该结果与 `abauqs` 仿真结果比较如下。

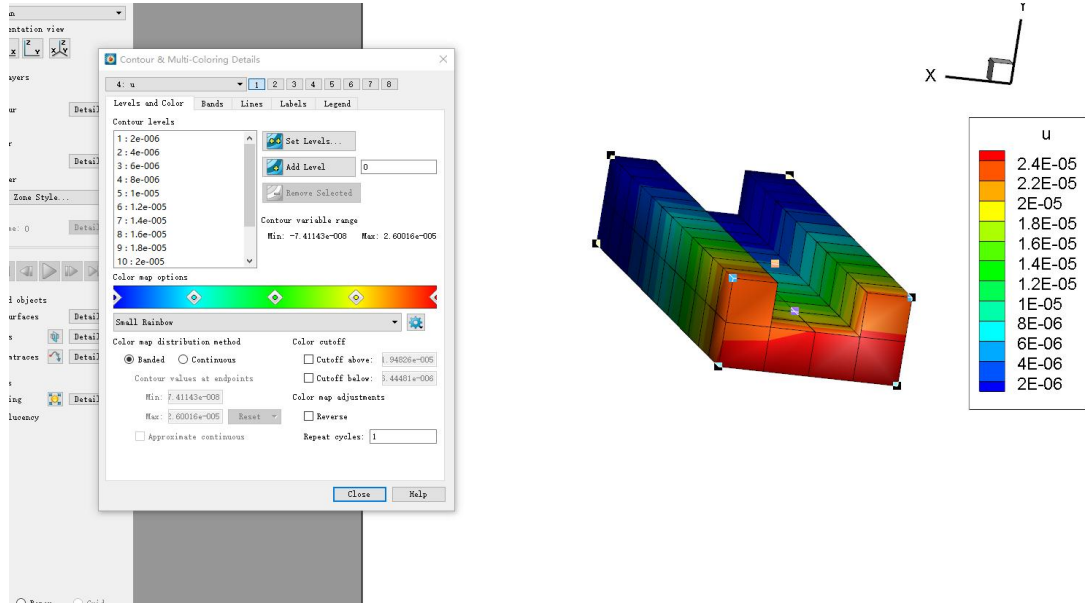


图 3.1 MATLAB 仿真结果（x 方向的位移）

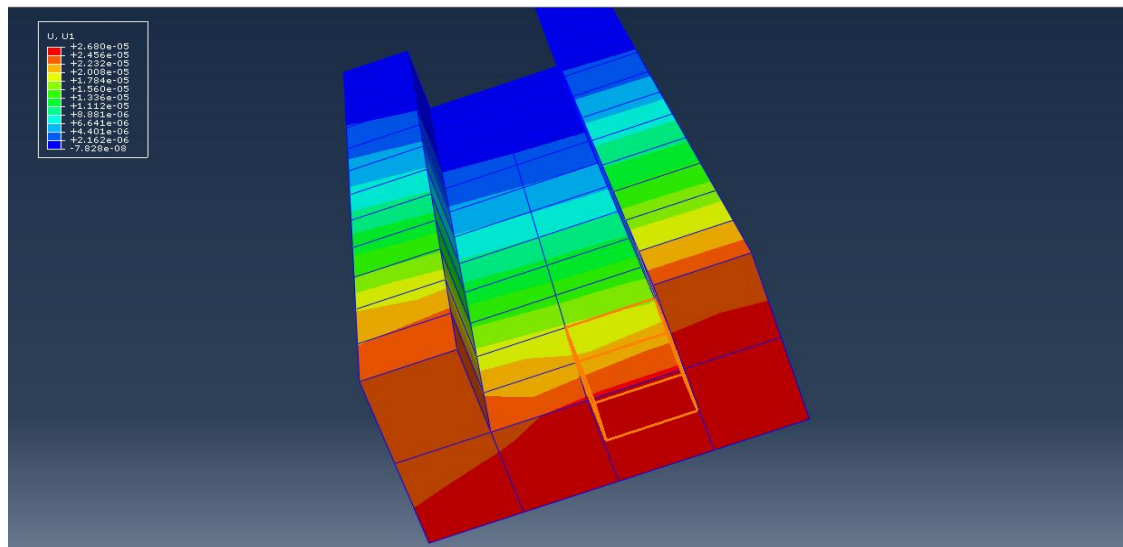


图 3.2 abaqus 仿真结果（x 方向的位移）

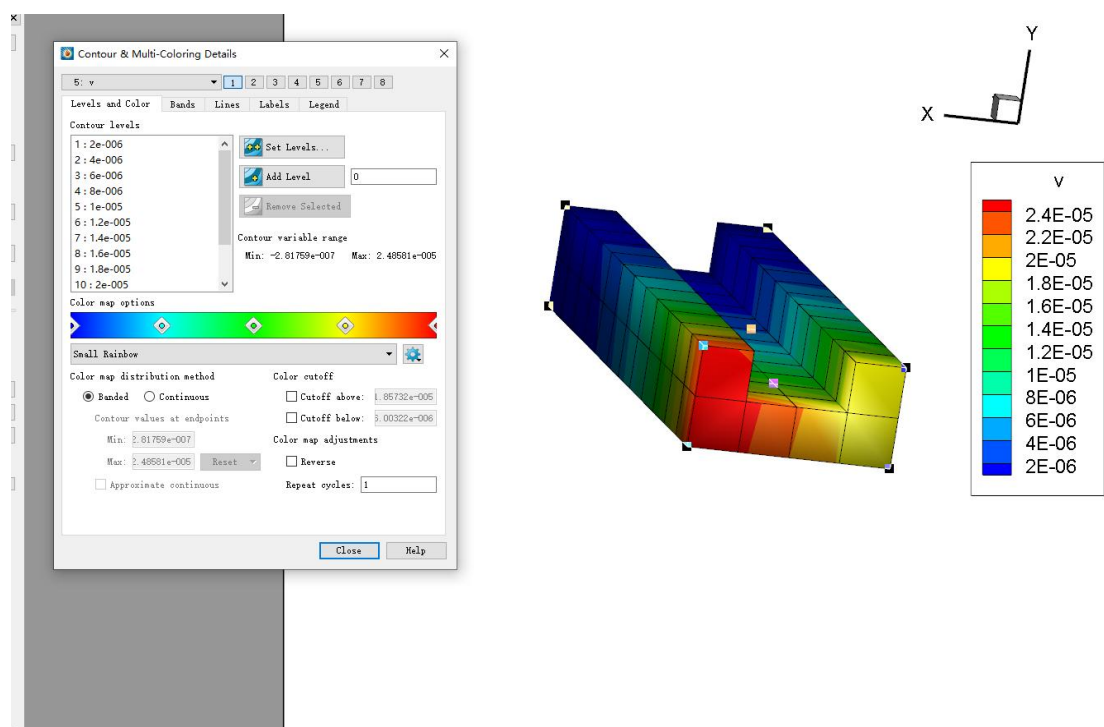


图 3.3 MATLAB 仿真结果（y 方向的位移）

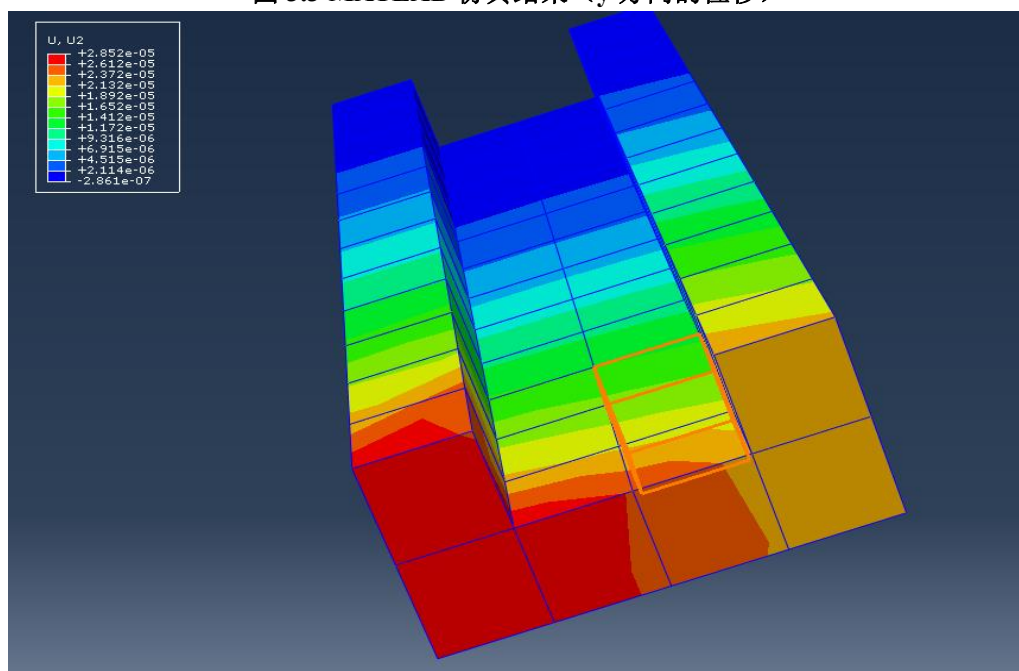


图 3.4 abaqus 仿真结果（y 方向的位移）

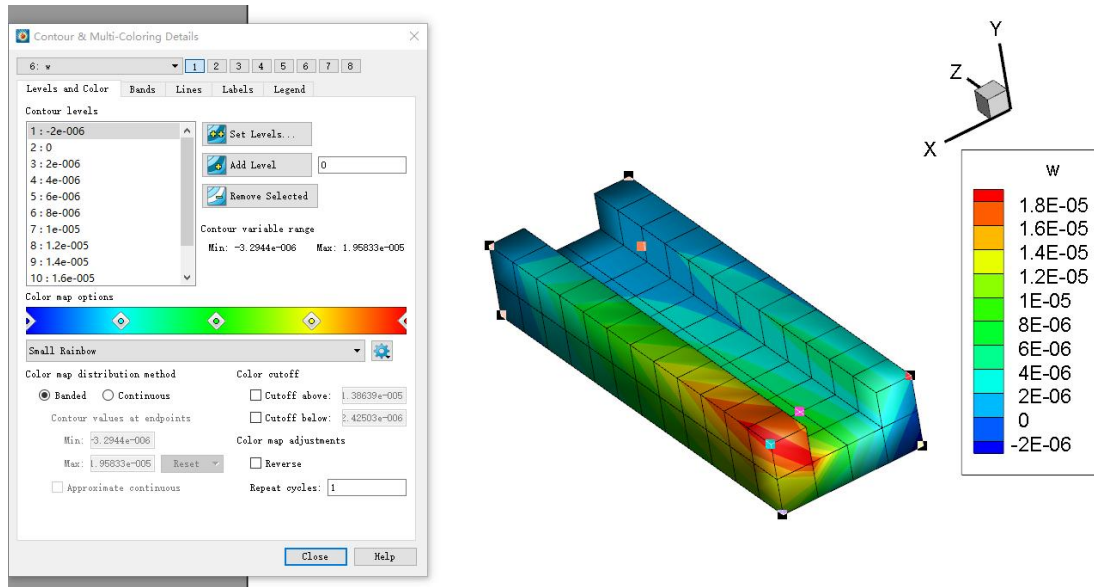


图 3.5 MATLAB 仿真结果（z 方向的位移）

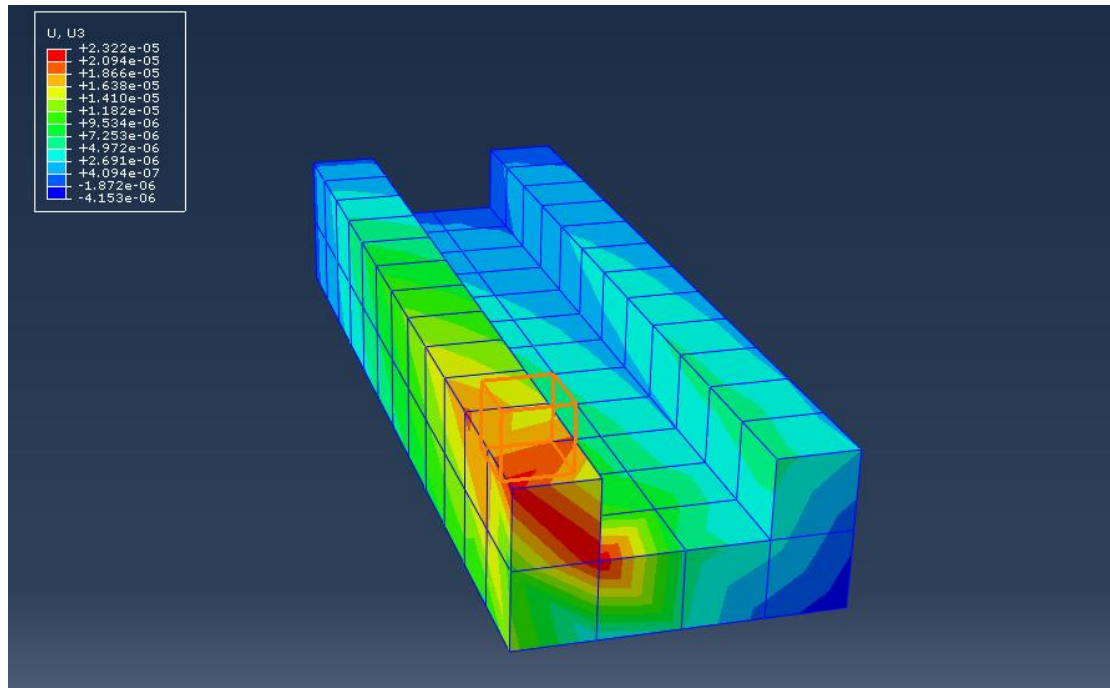


图 3.6 abaqus 仿真结果（z 方向的位移）

通过上述对比可知，MATLAB 编写的程序与 abaqus 仿真结果相近，但仍存在一定差距，说明此 MATLAB 程序在该实体的仿真上具有一定的准确性。