

**研究生期末大作业**

课程名称： 有限元方法及应用

研究生姓名： 曹健辉

学号： S230200204

任课教师： 王琥

目录

[一．问题重述 3](#_Toc159621694)

[1.1问题描述 3](#_Toc159621695)

[1.2理论分析 3](#_Toc159621696)

[二. 悬臂梁有限元分析的过程 3](#_Toc159621697)

[2.1实体类型模型分析 3](#_Toc159621698)

[2.1.1前处理 3](#_Toc159621699)

[2.1.2求解 4](#_Toc159621700)

[2.1.3后处理 4](#_Toc159621701)

[三．附录代码 6](#_Toc159621702)

[3.1代码解释 6](#_Toc159621703)

[3.2代码 6](#_Toc159621704)

[3.2.1实体模型代码 6](#_Toc159621705)

[四.致谢 11](#_Toc159621706)

悬臂梁abaqus有限元建模方法分析

# 一．问题重述

## 1.1问题描述

悬臂梁静力分析，某悬臂轻质梁，截面尺寸为200mmX400mm，长度为2500mm，弹性模量取30GPa，泊松比为0.2，悬臂梁顶部受0.5MPa均布载荷作用，端部受10kN的集中载荷，求结构整体位移云图、应力云图。。

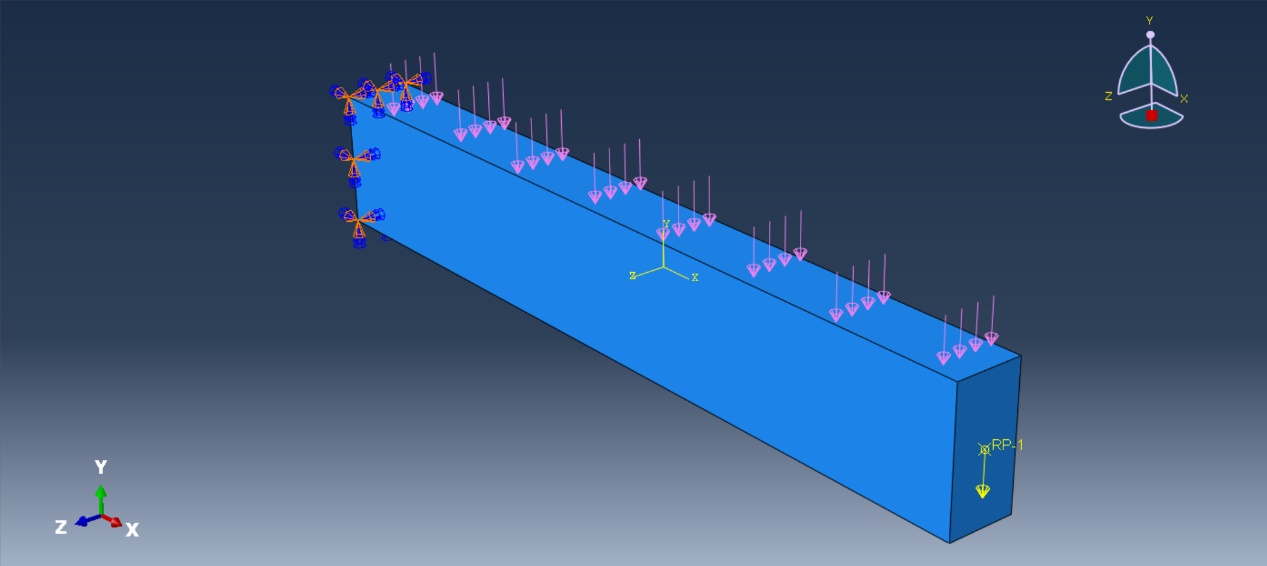


图1.悬臂梁的结构模型

## 1.2理论分析

于abaqus中建立悬臂梁模型，建立采取模型空间为三维，类型为可变形采用实体模型，计算最终结果，确定建模方式的可行性。

# 二. 悬臂梁有限元分析的过程

## 2.1实体类型模型分析

### 2.1.1前处理

几何模型的构建：采用模型空间为三维，类型为可变性，形状为实体的部件类型，得到一根长度为，边长为的梁。

材料参数的定义：杨氏模量为30GPa，泊松比为0.2。

网络系统的构建：采用C3D8R八结点线性六面体单元，10000个单元节点。

### 2.1.2求解

求解器的设定：设置一个静力学通用的分析步，设置保持默认。

边界条件的设定：模型左端设置为完全固定，右端首先在面中心设置一个参考点，然后将整个面耦合到参考点，最后再参考点施加一个延y方向负方向大小为100N的集中荷载，如图4所示。

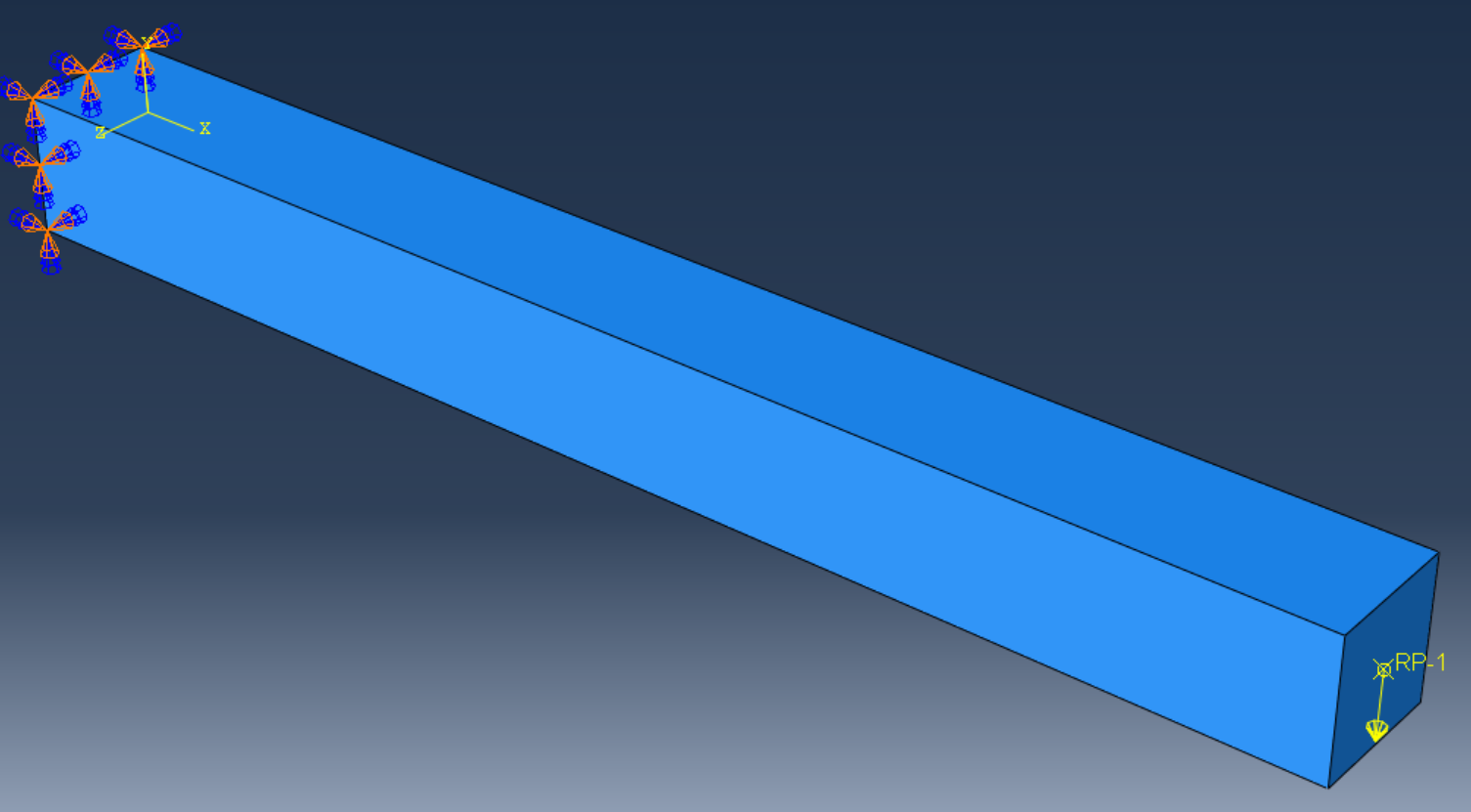
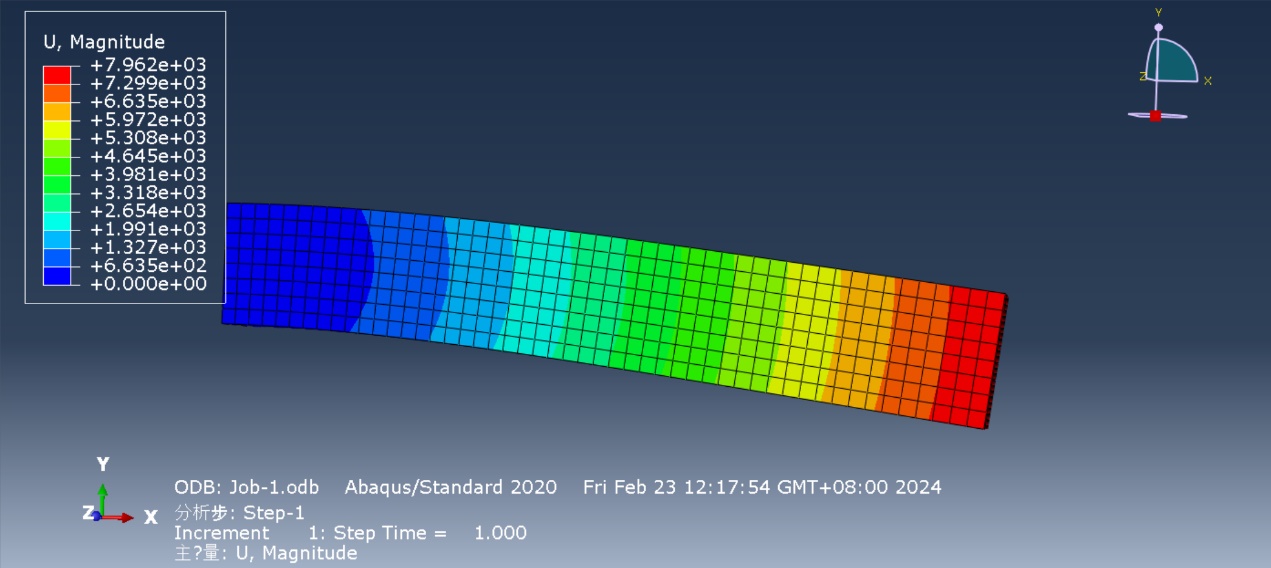


图2.实体模型构建

### 2.1.3后处理

创建作业文件shiti，提交计算，得到图3。



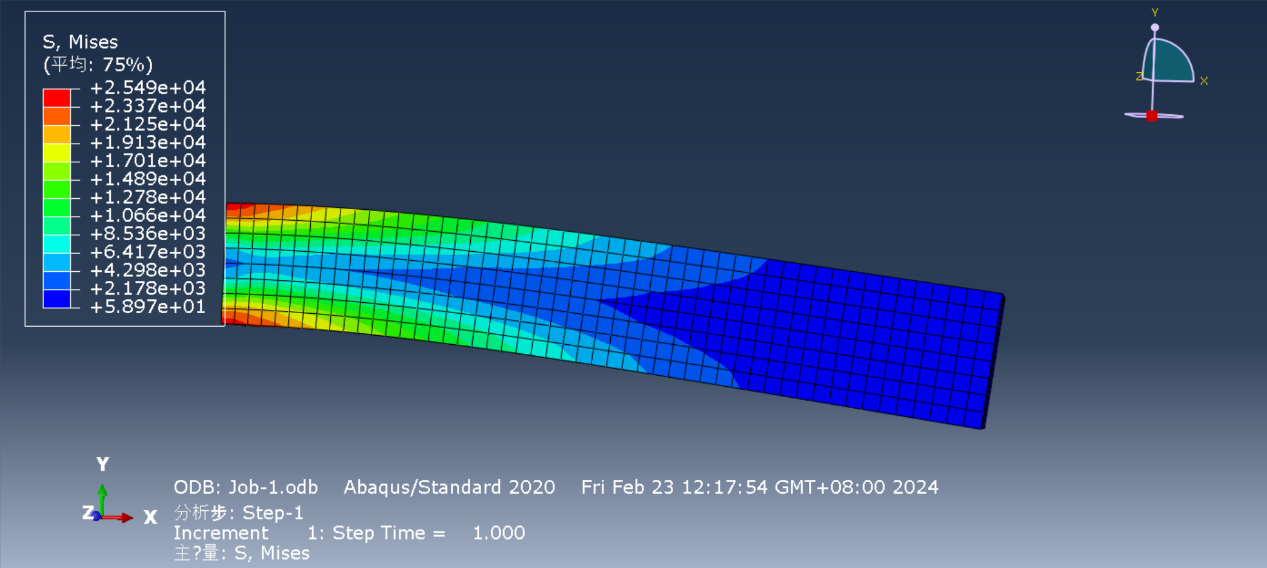


图3.实体模型后处理整体应力云图、位移云图

# 三．附录代码

## 3.1代码解释

代码是由abaqus软件附带的宏录制功能自动生成的abaqusMacros.py文件稍加更改后所得，将代码写入py文件后于abaqus中运行即可。

## 3.2代码

### 3.2.1实体模型代码

# -\*- coding: mbcs -\*-

# Do not delete the following import lines

from abaqus import \*

from abaqusConstants import \*

import \_\_main\_\_

import section

import regionToolset

import displayGroupMdbToolset as dgm

import part

import material

import assembly

import step

import interaction

import load

import mesh

import optimization

import job

import sketch

import visualization

import xyPlot

import displayGroupOdbToolset as dgo

import connectorBehavior

s = mdb.models['Model-1'].ConstrainedSketch(name='\_\_profile\_\_',

sheetSize=2500.0)

g, v, d, c = s.geometry, s.vertices, s.dimensions, s.constraints

s.setPrimaryObject(option=STANDALONE)

s.rectangle(point1=(-1250.0, -200.0), point2=(1250.0, 200.0))

session.viewports['Viewport: 1'].view.setValues(nearPlane=1908.76,

farPlane=2805.29, width=3628.61, height=1624.54, cameraPosition=(

424.962, 180.943, 2357.02), cameraTarget=(424.962, 180.943, 0))

p = mdb.models['Model-1'].Part(name='Part-1', dimensionality=THREE\_D,

type=DEFORMABLE\_BODY)

p = mdb.models['Model-1'].parts['Part-1']

p.BaseSolidExtrude(sketch=s, depth=200.0)

s.unsetPrimaryObject()

p = mdb.models['Model-1'].parts['Part-1']

session.viewports['Viewport: 1'].setValues(displayedObject=p)

del mdb.models['Model-1'].sketches['\_\_profile\_\_']

session.viewports['Viewport: 1'].partDisplay.setValues(sectionAssignments=ON,

engineeringFeatures=ON)

session.viewports['Viewport: 1'].partDisplay.geometryOptions.setValues(

referenceRepresentation=OFF)

mdb.models['Model-1'].Material(name='Material1')

mdb.models['Model-1'].materials['Material1'].Elastic(table=((30000.0, 0.2), ))

mdb.models['Model-1'].HomogeneousSolidSection(name='Section-1',

material='Material1', thickness=None)

p = mdb.models['Model-1'].parts['Part-1']

c = p.cells

cells = c.getSequenceFromMask(mask=('[#1 ]', ), )

region = regionToolset.Region(cells=cells)

p = mdb.models['Model-1'].parts['Part-1']

p.SectionAssignment(region=region, sectionName='Section-1', offset=0.0,

offsetType=MIDDLE\_SURFACE, offsetField='',

thicknessAssignment=FROM\_SECTION)

session.viewports['Viewport: 1'].partDisplay.setValues(sectionAssignments=OFF,

engineeringFeatures=OFF, mesh=ON)

session.viewports['Viewport: 1'].partDisplay.meshOptions.setValues(

meshTechnique=ON)

p = mdb.models['Model-1'].parts['Part-1']

p.seedPart(size=10.0, deviationFactor=0.1, minSizeFactor=0.1)

p = mdb.models['Model-1'].parts['Part-1']

p.seedPart(size=50.0, deviationFactor=0.1, minSizeFactor=0.1)

p = mdb.models['Model-1'].parts['Part-1']

p.generateMesh()

a = mdb.models['Model-1'].rootAssembly

session.viewports['Viewport: 1'].setValues(displayedObject=a)

session.viewports['Viewport: 1'].assemblyDisplay.setValues(

optimizationTasks=OFF, geometricRestrictions=OFF, stopConditions=OFF)

a = mdb.models['Model-1'].rootAssembly

a.DatumCsysByDefault(CARTESIAN)

p = mdb.models['Model-1'].parts['Part-1']

a.Instance(name='Part-1-1', part=p, dependent=ON)

session.viewports['Viewport: 1'].assemblyDisplay.setValues(

adaptiveMeshConstraints=ON)

mdb.models['Model-1'].StaticStep(name='Step-1', previous='Initial')

session.viewports['Viewport: 1'].assemblyDisplay.setValues(step='Step-1')

session.viewports['Viewport: 1'].assemblyDisplay.setValues(interactions=ON,

constraints=ON, connectors=ON, engineeringFeatures=ON,

adaptiveMeshConstraints=OFF)

a = mdb.models['Model-1'].rootAssembly

v11 = a.instances['Part-1-1'].vertices

a.DatumPointByMidPoint(point1=v11[6], point2=v11[5])

a = mdb.models['Model-1'].rootAssembly

d11 = a.datums

a.ReferencePoint(point=d11[4])

a = mdb.models['Model-1'].rootAssembly

r1 = a.referencePoints

refPoints1=(r1[5], )

region1=regionToolset.Region(referencePoints=refPoints1)

a = mdb.models['Model-1'].rootAssembly

s1 = a.instances['Part-1-1'].faces

side1Faces1 = s1.getSequenceFromMask(mask=('[#4 ]', ), )

region2=regionToolset.Region(side1Faces=side1Faces1)

mdb.models['Model-1'].Coupling(name='Constraint-1', controlPoint=region1,

surface=region2, influenceRadius=WHOLE\_SURFACE, couplingType=KINEMATIC,

localCsys=None, u1=ON, u2=ON, u3=ON, ur1=ON, ur2=ON, ur3=ON)

session.viewports['Viewport: 1'].assemblyDisplay.setValues(loads=ON, bcs=ON,

predefinedFields=ON, interactions=OFF, constraints=OFF,

engineeringFeatures=OFF)

session.viewports['Viewport: 1'].view.setValues(nearPlane=4814.48,

farPlane=6156.66, width=2900.06, height=1298.36, cameraPosition=(

-393.666, 3245.26, 4506.34), cameraUpVector=(0.162729, 0.558215,

-0.813581), cameraTarget=(109.45, -63.9089, 54.4588))

session.viewports['Viewport: 1'].view.setValues(nearPlane=4261.04,

farPlane=6627.6, width=2566.69, height=1149.11, cameraPosition=(

-2791.68, 1804.49, 4412.8), cameraUpVector=(0.228558, 0.768483,

-0.597658), cameraTarget=(146.283, -41.7787, 55.8956))

a = mdb.models['Model-1'].rootAssembly

f1 = a.instances['Part-1-1'].faces

faces1 = f1.getSequenceFromMask(mask=('[#1 ]', ), )

region = regionToolset.Region(faces=faces1)

mdb.models['Model-1'].EncastreBC(name='BC-1', createStepName='Step-1',

region=region, localCsys=None)

session.viewports['Viewport: 1'].view.setValues(nearPlane=4078.95,

farPlane=6999.7, width=2457, height=1100.01, cameraPosition=(4566.42,

978.421, 3079.08), cameraUpVector=(-0.359605, 0.862, -0.357267),

cameraTarget=(-23.3453, -22.7351, 86.6422))

a = mdb.models['Model-1'].rootAssembly

r1 = a.referencePoints

refPoints1=(r1[5], )

region = regionToolset.Region(referencePoints=refPoints1)

mdb.models['Model-1'].ConcentratedForce(name='Load-1', createStepName='Step-1',

region=region, cf2=-10000.0, distributionType=UNIFORM, field='',

localCsys=None)

session.viewports['Viewport: 1'].view.setValues(nearPlane=4170.21,

farPlane=6900.48, width=2511.98, height=1124.62, cameraPosition=(

4007.9, 2220.5, 3205.87), cameraUpVector=(-0.531313, 0.727643,

-0.433869), cameraTarget=(-20.2701, -29.5741, 85.9441))

a = mdb.models['Model-1'].rootAssembly

s1 = a.instances['Part-1-1'].faces

side1Faces1 = s1.getSequenceFromMask(mask=('[#2 ]', ), )

region = regionToolset.Region(side1Faces=side1Faces1)

mdb.models['Model-1'].Pressure(name='Load-2', createStepName='Step-1',

region=region, distributionType=UNIFORM, field='', magnitude=250.0,

amplitude=UNSET)

session.viewports['Viewport: 1'].assemblyDisplay.setValues(loads=OFF, bcs=OFF,

predefinedFields=OFF, connectors=OFF)

mdb.Job(name='Job-1', model='Model-1', description='', type=ANALYSIS,

atTime=None, waitMinutes=0, waitHours=0, queue=None, memory=90,

memoryUnits=PERCENTAGE, getMemoryFromAnalysis=True,

explicitPrecision=SINGLE, nodalOutputPrecision=SINGLE, echoPrint=OFF,

modelPrint=OFF, contactPrint=OFF, historyPrint=OFF, userSubroutine='',

scratch='', resultsFormat=ODB, multiprocessingMode=DEFAULT, numCpus=1,

numGPUs=0)

mdb.jobs['Job-1'].submit(consistencyChecking=OFF)

session.mdbData.summary()

o3 = session.openOdb(name='D:/abaqus/exerciese/fem\_final\_work/cao/Job-1.odb')

session.viewports['Viewport: 1'].setValues(displayedObject=o3)

session.viewports['Viewport: 1'].makeCurrent()

session.viewports['Viewport: 1'].odbDisplay.display.setValues(plotState=(

CONTOURS\_ON\_DEF, ))

session.viewports['Viewport: 1'].view.setValues(nearPlane=4394.76,

farPlane=6817.96, width=2647.23, height=1185.17, cameraPosition=(

3173.9, 1243.98, 4524.53), cameraUpVector=(-0.300529, 0.835879,

-0.459336), cameraTarget=(107.715, -85.7877, 68.8434))

session.viewports['Viewport: 1'].view.setValues(nearPlane=4625.9,

farPlane=6537.49, width=2786.46, height=1247.5, cameraPosition=(

2348.99, 1.42957, 5168.76), cameraUpVector=(-0.143635, 0.936962,

-0.318545), cameraTarget=(102.34, -93.8843, 73.0413))

session.viewports['Viewport: 1'].odbDisplay.setPrimaryVariable(

variableLabel='U', outputPosition=NODAL, refinement=(INVARIANT,

'Magnitude'), )

# 四.致谢

研一的第一个学期已经圆满结束，我要由衷地感谢王琥老师在有限元方法课程中给予我的悉心教导。在这段时间里，不仅提升了运用Abaqus等软件的技巧，更重要的是，老师深入浅出地讲解了有限元方法的理论和应用，使我对这一领域有了更深入的认识和理解。通过课堂上的案例分析和实践操作，我不仅掌握了软件的使用技巧，还学到了如何将理论知识应用到实际工程问题中去解决。王老师耐心细致的讲解和悉心指导让我受益匪浅，老师的教诲将成为我在未来科研和工作中的宝贵财富。