

一小时学会 abaqus 脚本

Learn Abaqus script in one hour

J.T.B Overvelde

简介

脚本是一个强大的工具，他将 abaqus 用户图形界面和 python 变成语言的功能结合起来。这个手册不等同于一个完整的 abaqus 脚本手册，它只是从实用的视角来做一个简介，试图为读者阐明一条简单，快速的脚本入门之路。如果你不曾使用过 abaqus 用户图形界面或者是有限元的话，你应该首先去体验一下。你不必非要对输入文件了如指掌。你应该可以在两小时或更少的时间内弄懂这个简介。

手册将包含以下内容：

- 利用用户图形界面生成简单模型
- 建立你的第一个模型数据脚本
- 利用用户图形界面生成输出文件
- 建立你的第一个输出文件脚本
- 例子：不同用途时对脚本的调整
- 持续使用脚本的建议

创建脚本最基本的原则有如下排序：

- 创建模型和保存模型
- 利用 abaqus 生成的文件创建脚本文件
- 创建输出文件
- 利用生成的脚本文件重做计算
- 调整脚本来生成不同的模型和输出文件

你可能知道 abaqus 用户图形界面在运行一个模拟的时候将生成一个输入文件，脚本文件也一样，将生成一个传递给求解器的输入文件。

利用用户图形界面生成简单模型

我所用的 abaqus 版本为 6.8-2，但其他版本也一样可以使用。

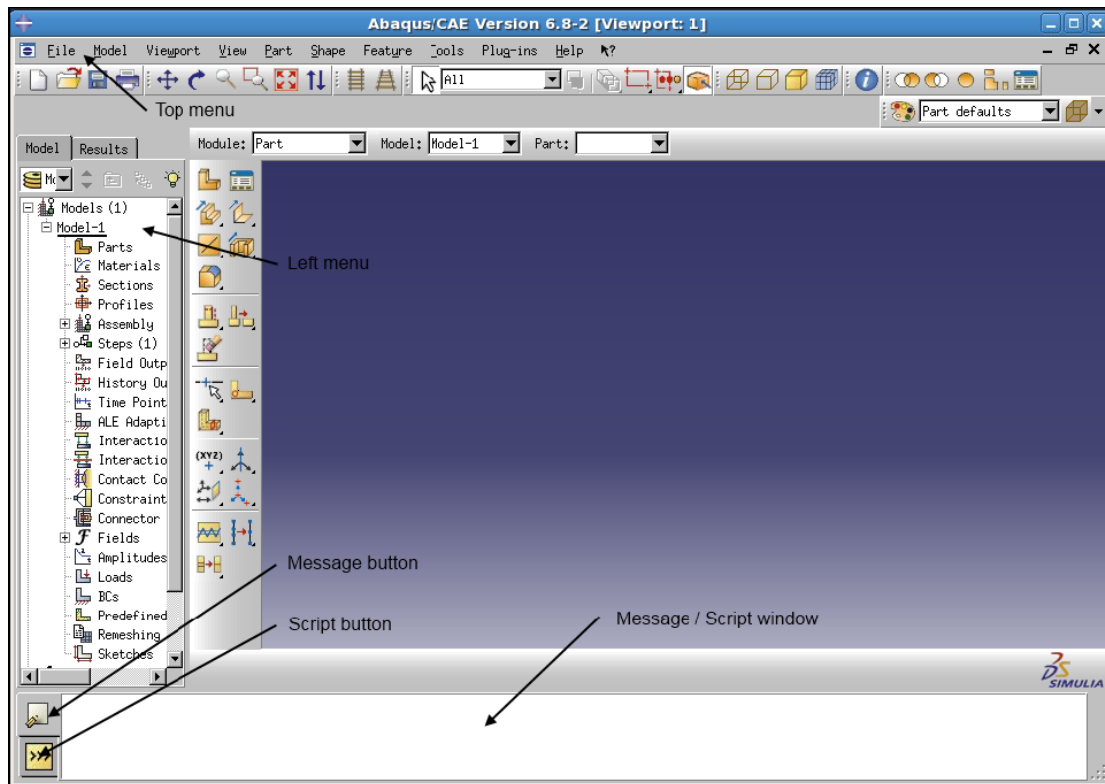


图 1 abaqusCAE 6.8-2 的用户图形界面

首先，启动 abaqus CAE。确保我们在同样的图一所示的用户界面。我添加了一些稍后会用到的菜单和按钮的名称。确保你在正确的工作目录下，我们要创建一些稍后要用到的文件，你要能找到他们。在我们开始建模之前我们点击图一中的“script button”，切换到脚本窗口，键入下面一行命令：

```
session.journalOptions.setValues(replayGeometry=COORDINATE,recoverGeometry=COORDINATE)
```

输入结束后，点击回车键。界面不会发生什么变化，但这行命令将使得之后创建 sets, surfaces, 选择 region 等等的 python 脚本更可读，关于这部分毋须多想。现在我们开始创建模型，步骤如下：

- 创建 part: 2D Planar, Deformable, Shell, Approximate size: 20
- 用点 (-5,-1) 和对角点 (5,1) 做矩形
- 创建 material: Linear elastic, $E=1e9$, $\nu=0.3$
- 创建 section: Solid, Homogenous, Use material just created
- 将截面 section 赋给实体 part
- 创建 set: 左边缘 (Tools-set)
- 创建 surface: 顶边缘 (Tools-surface)
- Mesh: 设置 mesh control 为 structured 和 quad
- Mesh: 设置单元类型为 Standard, quadratic 和 Plane Stress
- Mesh: 以 0.5 的 globe size 撒种子，然后 mesh part
- Assembly: 创建部件实体
- Step: 创建 General Static step, 并将 Nlgeom 设置为 on, 将 initial 和 maximum increment size 设为 0.1
- BC: 在创建的 set 上加 displacement, $U1=U2=UR3=0$

- Load: 在创建的 surface 上施加 pressure, 设置 magnitude 为-1e-5
- Job: 创建一个名为 EXAMPLE 的 Job 并提交分析
- Save As: 命名为 EXAMPLE

你已经创建了模型, 请确保工作目录下包含了文件 ‘EXAMPLE.jnl’

建立你的第一个模型数据脚本

我们从创建脚本文件开始。打开 ‘EXAMPLE.jnl’ 文件, 并将该文件另存为 ‘EXAMPLE.py’。我将在文档的最后附上 ‘EXAMPLE.py’ 的内容。Python 文件看起来一团糟, 给他建立一个结构是一个好的习惯。让我带你来浏览一下代码, 告诉你脚本的每一行分属那一块。对这些脚本的透彻的理解还需要更多的经验和练习。但是, 试试看吧, 看你能不能识别出一些你在用户图形界面中的操作。

```
# -*- coding: mbcs -*-
```

这一行是注释行。所以不那么重要, 但要了解 python 的指令行都以#开头。

```
from job import *
from part import *
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from sketch import *
from visualization import *
from connectorBehavior import *
```

我们在 python 的环境中工作, 它不包含 abaqus 的所有函数块。这些行是将脚本中用到的这些 abaqus 模块引入进来。

```
mdb.models['Model-1'].ConstrainedSketch(name='__profile__', sheetSize=20.0)
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(-5.0, -1.0), point2=(
    5.0, -1.0))
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, -1.0))
mdb.models['Model-1'].sketches['__profile__'].HorizontalConstraint(
    addUndoState=False, entity=
    mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, -1.0),
    ))
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(5.0, -1.0), point2=(
    5.0, 1.0))
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((5.0, 0.0))
mdb.models['Model-1'].sketches['__profile__'].VerticalConstraint(addUndoState
```

=

```

False, entity=
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((5.0, 0.0),
))
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, -1.0))
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((5.0, 0.0))
mdb.models['Model-1'].sketches['__profile__'].PerpendicularConstraint(
addUndoState=False, entity1=
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, -1.0),
), entity2=mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((
5.0, 0.0), ))
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(5.0, 1.0), point2=(
-5.0, 1.0))
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, 1.0))
mdb.models['Model-1'].sketches['__profile__'].HorizontalConstraint(
addUndoState=False, entity=
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, 1.0),
))
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((5.0, 0.0))
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, 1.0))
mdb.models['Model-1'].sketches['__profile__'].PerpendicularConstraint(
addUndoState=False, entity1=
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((5.0, 0.0), )
, entity2=mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((
0.0, 1.0), ))
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(-5.0, 1.0), point2=(
-5.0, -1.0))
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((-5.0, 0.0))
mdb.models['Model-1'].sketches['__profile__'].VerticalConstraint(addUndoState
=
False, entity=
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((-5.0, 0.0),
))
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, 1.0))
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((-5.0, 0.0))
mdb.models['Model-1'].sketches['__profile__'].PerpendicularConstraint(
addUndoState=False, entity1=
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, 1.0), )
, entity2=mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((
-5.0, 0.0), ))
mdb.models['Model-1'].Part(dimensionality=TWO_D_PLANAR, name='Part-1',
type=
DEFORMABLE_BODY)
mdb.models['Model-1'].parts['Part-1'].BaseShell(sketch=
mdb.models['Model-1'].sketches['__profile__'])

```

```
del mdb.models['Model-1'].sketches['__profile__']
```

这些行是创建草图，并生成部件。

```
-----
mdb.models['Model-1'].Material(name='Material-1')
mdb.models['Model-1'].materials['Material-1'].Elastic(table=((1000000000.0,
    0.3), ))
mdb.models['Model-1'].HomogeneousSolidSection(material='Material-1',
name=
    'Section-1', thickness=None)
mdb.models['Model-1'].parts['Part-1'].SectionAssignment(offset=0.0,
    offsetField='', offsetType=MIDDLE_SURFACE, region=Region(
    faces=mdb.models['Model-1'].parts['Part-1'].faces.findAt((-1.666667,
    -0.333333, 0.0), (0.0, 0.0, 1.0)), ), sectionName='Section-1',
    thicknessAssignment=FROM_SECTION)
Material, section 和 section assignment 由上面几行指令完成。
-----
```

```
-----
mdb.models['Model-1'].parts['Part-1'].Set(edges=
    mdb.models['Model-1'].parts['Part-1'].edges.findAt((-5.0, 0.5, 0.0), ),
    name='Set-1')
mdb.models['Model-1'].parts['Part-1'].Surface(name='Surf-1', side1Edges=
    mdb.models['Model-1'].parts['Part-1'].edges.findAt((2.5, 1.0, 0.0), ))
这里我们创建了 set 和 surface。注意每行末尾的 findAt 指令。findAt 被用于
替代 getSequenceFromMask (abaqus 用的数字系统)，因为我们是在建立 abaqus
模型之前将指令输入脚本窗口。
-----
```

```
-----
mdb.models['Model-1'].parts['Part-1'].setMeshControls(elemShape=QUAD,
regions=
    mdb.models['Model-1'].parts['Part-1'].faces.findAt((-1.666667, -0.333333,
    0.0), ), technique=STRUCTURED)
mdb.models['Model-1'].parts['Part-1'].seedPart(deviationFactor=0.1, size=0.5)
mdb.models['Model-1'].parts['Part-1'].seedEdgeBySize(constraint=FINER,
    deviationFactor=0.1, edges=
    mdb.models['Model-1'].parts['Part-1'].edges.findAt((-2.5, -1.0, 0.0), ), (
    2.5, 1.0, 0.0), ), size=2.5)
mdb.models['Model-1'].parts['Part-1'].seedEdgeBySize(constraint=FINER,
    deviationFactor=0.1, edges=
    mdb.models['Model-1'].parts['Part-1'].edges.findAt((-2.5, -1.0, 0.0), ), (
    2.5, 1.0, 0.0), ), size=0.5)
mdb.models['Model-1'].parts['Part-1'].generateMesh()
mdb.models['Model-1'].parts['Part-1'].setElementType(elemTypes=(ElemType(
    elemCode=CPS8R, elemLibrary=STANDARD), ElemType(elemCode=CPS6M,
    elemLibrary=STANDARD)), regions=(
    mdb.models['Model-1'].parts['Part-1'].faces.findAt((-1.666667, -0.333333,
    0.0), ), ))
-----
```

这几行创建的 mesh control, element type 和 mesh。笔者撒种子过程中有过修改, 全部都在脚本中有所体现。

```

mdb.models['Model-1'].rootAssembly.DatumCsysByDefault(CARTESIAN)
mdb.models['Model-1'].rootAssembly.Instance(dependent=ON,
name='Part-1-1',
    part=mdb.models['Model-1'].parts['Part-1'])

```

上面是 assembly 的过程。注意当用到 rootAssembly 时, 你不必须给出 assembly 的一个名称, 当然这只是因为这里只有一个装配件。

```

mdb.models['Model-1'].StaticStep(initialInc=0.1, maxInc=0.1, name='Step-1',
    nlgeom=ON, previous='Initial')
mdb.models['Model-1'].DisplacementBC(amplitude=UNSET,
createStepName='Step-1',
    distributionType=UNIFORM, fieldName="", fixed=OFF, localCsys=None,
name=
    'BC-1', region=
    mdb.models['Model-1'].rootAssembly.instances['Part-1-1'].sets['Set-1'],
u1=
    0.0, u2=0.0, ur3=0.0)
mdb.models['Model-1'].Pressure(amplitude=UNSET, createStepName='Step-1',
    distributionType=UNIFORM, field="", magnitude=1e-05, name='Load-1',
region=

```

```

mdb.models['Model-1'].rootAssembly.instances['Part-1-1'].surfaces['Surf-1'])

```

施加 step 载荷步, BC 边界条件, 和 load 载荷。

```

mdb.Job(atTime=None, contactPrint=OFF, description="", echoPrint=OFF,
    explicitPrecision=SINGLE, getMemoryFromAnalysis=True, historyPrint=OFF,
memory=50, memoryUnits=PERCENTAGE, model='Model-1',
modelPrint=OFF,
    multiprocessingMode=DEFAULT, name='EXAMPLE',
nodalOutputPrecision=SINGLE,
    numCpus=1, queue=None, scratch="", type=ANALYSIS, userSubroutine="",
waitHours=0, waitMinutes=0)

```

```

mdb.jobs['EXAMPLE'].submit(consistencyChecking=OFF)

```

这里创建了 job 并提交 job 进行分析。这些代码后面的所有行都是不重要的信息, 你可以删去他们。最终的脚本文件将附在附件中。

PS: 脚本的末尾

```

mdb.models['Model-1'].loads['Load-1'].setValues(magnitude=-1e-05)

```

表示的是笔者在之后对模型进行的修改。

在运行这个脚本之前, 将你工作目录下除去 'EXAMPLE.py' 的左右文件都删

掉。如果你的用户图形界面还打开着一个新的模型，请不要保存任何东西。现在，在左上角的菜单栏中点击 **File**，然后点击 **Run script...**，然后选择你的脚本文件。如果每一步都准确无误的完成了，那么你的模型应该没有什么问题。确保文件 ‘EXAMPLE.odb’ 存在于你的工作目录下。如果你没有在你的工作目录下把之前的文件删除，abaqus 将会覆盖他们。这不会出现任何问题，不过，这里的目的是为了让你看看 odb 文件被创建的过程。

利用用户图形界面生成输出文件

关闭后重新打开 abaqus CAE 来重新记录脚本。现在点击左上角菜单栏的 **File** 点击 ‘Open...’，在工作目录下选择并打开文件 ‘EXAMPLE.odb’。我们用创建模型脚本一样的方法创建输出文件的脚本。最重要的不同之处在于这个过程被记录在 ‘Abaqus.rpy’ 中而不是 ‘EXAMPLE.jnl’。让我们在变形情况下创建一个应力的图形，按如下步骤操作：

Plot 变形状态下的应力。在 **File** 中点击 **print**，**destination** 选择 **file**，保存.tiff 格式的图形。命名为 EXAMPLE。

你必须保存 odb 文件。确保 abaqus.rpy 存在于你的工作目录下。如果你找不到你可以试试在启动 abaqus CAE 的原始工作目录下找找看。

建立你的第一个输出文件脚本

将 ‘abaqus.rpy’ 保存为 ‘EXAMPLE_ODB.py’。这个脚本文件的内容在附件中给出。下面是一个简短的脚本文件。

```

-----
from abaqus import *
from abaqusConstants import *
session.Viewport(name='Viewport: 1', origin=(0.0, 0.0),
width=261.944432258606,
    height=178.915559887886)
session.viewports['Viewport: 1'].makeCurrent()
session.viewports['Viewport: 1'].maximize()
executeOnCaeStartup()
session.viewports['Viewport: 1'].partDisplay.geometryOptions.setValues(
    referenceRepresentation=ON)
o1 = session.openOdb(
    name='E:/zhouke/work/Abaqus/Abaqus-examples/learn abaqus script in
one hour/EXAMPLE.odb')
session.viewports['Viewport: 1'].setValues(displayedObject=o1)
-----
session.viewports['Viewport: 1'].odbDisplay.display.setValues(plotState=(
    CONTOURS_ON_DEF, ))
session.printToFile(fileName='EXAMPLE', format=TIFF, canvasObjects=(
    session.viewports['Viewport: 1'], ))

```

上面的语句打开了一些模块，创建了视图并打开了 odb 文件

这两句设置了变形情况下的应力表示，并且保存了图像。最终的输出脚本文件在附件中给出。

例子：不同用途时对脚本的调整

我将给出一个调整脚本的例子，我将调整一下内容：

- 改进于用固定的数值创建部件，我调整脚本文件开头的两处内容将使得形状可变。
- 我将让 `mdb` 和 `odb` 文件结合。

我将将文件附在附件中，并高亮修改部分。我将不会详细解释修改部分，弄清楚这部分也将作为你的一个训练。

这次，我将不会通过用户图形界面运行脚本，我将直接从终端运行它。

我们可以用两个不同的命令：

```
abaqus cae script=EXAMPLE_MDB.py
```

```
abaqus cae noGUI=EXAMPLE_MDB.py
```

第一行将打开 `abaqus cae` 并且你竟能够看到它做了什么。后一行将打开 `abaqus` 但并没有用户图形界面，你将只得到结果。

- 持续使用脚本的建议

到现在你应该对于工作方法比较熟悉了：剩下所有困难的工作都交给 `abaqus` 去做吧。我同样希望我没有骗你，你在一个小时之内理解到了要点。在手册的最后我想给出一些建议：

- 如果你刚开始在日常的模拟中使用脚本语言，你需要熟悉并习惯 `python` 语言并且联系写脚本，这是学习它最好的方式。我用了两个半月达到我现在的水平：写一个指导手册。
- 你可以在网上找到很多 `python` 的知识，`Google` 是一个有用的工具。但你可能找不到非常多关于脚本的东西，你可以试试看这个网站：<http://abaqusdoc.ucalgary.ca/v6.9/>。你可以找到一个完整的脚本命令的参考。
- 试着用 `MATLAB` 调用 `Abaqus CAE`。这会增加一个很大的数学工具箱。试试：

```
-unix([abaqus cae script=EXAMPLE_MDB.py])  
-system([abaqus cae script=EXAMPLE_MDB.py])
```


附件:

第一个模型 EXAMPLE.py

```
# -*- coding: mbcs -*-
```

```
from job import *
```

```
from part import *
```

```
from material import *
```

```
from section import *
```

```
from assembly import *
```

```
from step import *
```

```
from interaction import *
```

```
from load import *
```

```
from mesh import *
```

```
from sketch import *
```

```
from visualization import *
```

```
from connectorBehavior import *
```

```
mdb.models['Model-1'].ConstrainedSketch(name='__profile__', sheetSize=20.0)
```

```
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(-5.0, -1.0), point2=(5.0, -1.0))
```

```
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, -1.0))
```

```
mdb.models['Model-1'].sketches['__profile__'].HorizontalConstraint(
    addUndoState=False, entity=
    mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, -1.0),
    ))
```

```
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(5.0, -1.0), point2=(5.0, 1.0))
```

```
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((5.0, 0.0))
```

```
mdb.models['Model-1'].sketches['__profile__'].VerticalConstraint(addUndoState
```

```
=
    False, entity=
    mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((5.0, 0.0),
    ))
```

```
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, -1.0))
```

```
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((5.0, 0.0))
```

```
mdb.models['Model-1'].sketches['__profile__'].PerpendicularConstraint(
    addUndoState=False, entity1=
    mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, -1.0),
    ), entity2=mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((5.0, 0.0), ))
```

```
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(5.0, 1.0), point2=(-5.0, 1.0))
```

```
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, 1.0))
```

```
mdb.models['Model-1'].sketches['__profile__'].HorizontalConstraint(
    addUndoState=False, entity=
```

```

        mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, 1.0),
    ))
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((5.0, 0.0))
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, 1.0))
mdb.models['Model-1'].sketches['__profile__'].PerpendicularConstraint(
    addUndoState=False, entity1=
        mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((5.0, 0.0), )
    , entity2=mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((
        0.0, 1.0), ))
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(-5.0, 1.0), point2=(
    -5.0, -1.0))
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((-5.0, 0.0))
mdb.models['Model-1'].sketches['__profile__'].VerticalConstraint(addUndoState
=
    False, entity=
        mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((-5.0, 0.0),
    ))
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, 1.0))
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((-5.0, 0.0))
mdb.models['Model-1'].sketches['__profile__'].PerpendicularConstraint(
    addUndoState=False, entity1=
        mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, 1.0), )
    , entity2=mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((
        -5.0, 0.0), ))
mdb.models['Model-1'].Part(dimensionality=TWO_D_PLANAR, name='Part-1',
type=
    DEFORMABLE_BODY)
mdb.models['Model-1'].parts['Part-1'].BaseShell(sketch=
    mdb.models['Model-1'].sketches['__profile__'])
del mdb.models['Model-1'].sketches['__profile__']
mdb.models['Model-1'].Material(name='Material-1')
mdb.models['Model-1'].materials['Material-1'].Elastic(table=((100000000.0,
    0.3), ))
mdb.models['Model-1'].HomogeneousSolidSection(material='Material-1',
name=
    'Section-1', thickness=None)
mdb.models['Model-1'].parts['Part-1'].SectionAssignment(offset=0.0,
    offsetField='', offsetType=MIDDLE_SURFACE, region=Region(
    faces=mdb.models['Model-1'].parts['Part-1'].faces.findAt((( -1.666667,
    -0.333333, 0.0), (0.0, 0.0, 1.0)), ), ), sectionName='Section-1',
    thicknessAssignment=FROM_SECTION)
mdb.models['Model-1'].parts['Part-1'].Set(edges=
    mdb.models['Model-1'].parts['Part-1'].edges.findAt((( -5.0, 0.5, 0.0), ),
    name='Set-1')

```

```
mdb.models['Model-1'].parts['Part-1'].Surface(name='Surf-1', side1Edges=
    mdb.models['Model-1'].parts['Part-1'].edges.findAt(((2.5, 1.0, 0.0), ))
mdb.models['Model-1'].parts['Part-1'].setMeshControls(elemShape=QUAD,
regions=
    mdb.models['Model-1'].parts['Part-1'].faces.findAt((-1.666667, -0.333333,
    0.0), ), technique=STRUCTURED)
mdb.models['Model-1'].parts['Part-1'].seedPart(deviationFactor=0.1, size=0.5)
mdb.models['Model-1'].parts['Part-1'].seedEdgeBySize(constraint=FINER,
    deviationFactor=0.1, edges=
    mdb.models['Model-1'].parts['Part-1'].edges.findAt((-2.5, -1.0, 0.0), ), (
    2.5, 1.0, 0.0), ), size=2.5)
mdb.models['Model-1'].parts['Part-1'].seedEdgeBySize(constraint=FINER,
    deviationFactor=0.1, edges=
    mdb.models['Model-1'].parts['Part-1'].edges.findAt((-2.5, -1.0, 0.0), ), (
    2.5, 1.0, 0.0), ), size=0.5)
mdb.models['Model-1'].parts['Part-1'].generateMesh()
mdb.models['Model-1'].parts['Part-1'].setElementType(elemTypes=(ElemType(
    elemCode=CPS8R, elemLibrary=STANDARD), ElemType(elemCode=CPS6M,
    elemLibrary=STANDARD)), regions=(
    mdb.models['Model-1'].parts['Part-1'].faces.findAt((-1.666667, -0.333333,
    0.0), ), ))
mdb.models['Model-1'].rootAssembly.DatumCsysByDefault(CARTESIAN)
mdb.models['Model-1'].rootAssembly.Instance(dependent=ON,
name='Part-1-1',
    part=mdb.models['Model-1'].parts['Part-1'])
mdb.models['Model-1'].StaticStep(initialInc=0.1, maxInc=0.1, name='Step-1',
    nlgeom=ON, previous='Initial')
mdb.models['Model-1'].DisplacementBC(amplitude=UNSET,
createStepName='Step-1',
    distributionType=UNIFORM, fieldName="", fixed=OFF, localCsys=None,
name=
    'BC-1', region=
    mdb.models['Model-1'].rootAssembly.instances['Part-1-1'].sets['Set-1'],
u1=
    0.0, u2=0.0, ur3=0.0)
mdb.models['Model-1'].Pressure(amplitude=UNSET, createStepName='Step-1',
    distributionType=UNIFORM, field="", magnitude=1e-05, name='Load-1',
region=

mdb.models['Model-1'].rootAssembly.instances['Part-1-1'].surfaces['Surf-1'])
mdb.Job(atTime=None, contactPrint=OFF, description="", echoPrint=OFF,
    explicitPrecision=SINGLE, getMemoryFromAnalysis=True, historyPrint=OFF,
    memory=50, memoryUnits=PERCENTAGE, model='Model-1',
modelPrint=OFF,
```

```

multiprocessingMode=DEFAULT,                                name='EXAMPLE',
nodalOutputPrecision=SINGLE,
numCpus=1, queue=None, scratch="", type=ANALYSIS, userSubroutine="",
waitHours=0, waitMinutes=0)
mdb.jobs['EXAMPLE'].submit(consistencyChecking=OFF)
mdb.jobs['EXAMPLE']._Message(STARTED, {'phase': BATCHPRE_PHASE,
    'clientHost': '20090629-1652', 'handle': 0, 'jobName': 'EXAMPLE'})
mdb.jobs['EXAMPLE']._Message(WARNING, {'phase': BATCHPRE_PHASE,
    'message': 'DEGREE OF FREEDOM 6 IS NOT ACTIVE IN THIS MODEL AND
CAN NOT BE RESTRAINED',
    'jobName': 'EXAMPLE'})
mdb.jobs['EXAMPLE']._Message(ODB_FILE, {'phase': BATCHPRE_PHASE,
    'file': 'E:\\zhouke\\work\\Abaqus\\EXAMPLE.odb', 'jobName': 'EXAMPLE'})
mdb.jobs['EXAMPLE']._Message(COMPLETED, {'phase': BATCHPRE_PHASE,
    'message': 'Analysis phase complete', 'jobName': 'EXAMPLE'})
mdb.jobs['EXAMPLE']._Message(STARTED, {'phase': STANDARD_PHASE,
    'clientHost': '20090629-1652', 'handle': 732, 'jobName': 'EXAMPLE'})
mdb.jobs['EXAMPLE']._Message(STEP, {'phase': STANDARD_PHASE, 'stepId': 1,
    'jobName': 'EXAMPLE'})
mdb.jobs['EXAMPLE']._Message(ODB_FRAME, {'phase': STANDARD_PHASE,
'step': 0,
    'frame': 0, 'jobName': 'EXAMPLE'})
mdb.jobs['EXAMPLE']._Message(STATUS, {'totalTime': 0.0, 'attempts': 0,
    'timeIncrement': 0.1, 'increment': 0, 'stepTime': 0.0, 'step': 1,
    'jobName': 'EXAMPLE', 'severe': 0, 'iterations': 0,
    'phase': STANDARD_PHASE, 'equilibrium': 0})
mdb.jobs['EXAMPLE']._Message(MEMORY_ESTIMATE, {'phase':
STANDARD_PHASE,
    'jobName': 'EXAMPLE', 'memory': 24.1799392700195})
mdb.jobs['EXAMPLE']._Message(ODB_FRAME, {'phase': STANDARD_PHASE,
'step': 0,
    'frame': 1, 'jobName': 'EXAMPLE'})
mdb.jobs['EXAMPLE']._Message(STATUS, {'totalTime': 0.1, 'attempts': 1,
    'timeIncrement': 0.1, 'increment': 1, 'stepTime': 0.1, 'step': 1,
    'jobName': 'EXAMPLE', 'severe': 0, 'iterations': 1,
    'phase': STANDARD_PHASE, 'equilibrium': 1})
mdb.jobs['EXAMPLE']._Message(ODB_FRAME, {'phase': STANDARD_PHASE,
'step': 0,
    'frame': 2, 'jobName': 'EXAMPLE'})
mdb.jobs['EXAMPLE']._Message(STATUS, {'totalTime': 0.2, 'attempts': 1,
    'timeIncrement': 0.1, 'increment': 2, 'stepTime': 0.2, 'step': 1,
    'jobName': 'EXAMPLE', 'severe': 0, 'iterations': 1,
    'phase': STANDARD_PHASE, 'equilibrium': 1})
mdb.jobs['EXAMPLE']._Message(ODB_FRAME, {'phase': STANDARD_PHASE,

```

```
'step': 0,
  'frame': 3, 'jobName': 'EXAMPLE'})
mdb.jobs['EXAMPLE']._Message(STATUS, {'totalTime': 0.3, 'attempts': 1,
  'timeIncrement': 0.1, 'increment': 3, 'stepTime': 0.3, 'step': 1,
  'jobName': 'EXAMPLE', 'severe': 0, 'iterations': 1,
  'phase': STANDARD_PHASE, 'equilibrium': 1})
mdb.jobs['EXAMPLE']._Message(ODB_FRAME, {'phase': STANDARD_PHASE,
'step': 0,
  'frame': 4, 'jobName': 'EXAMPLE'})
mdb.jobs['EXAMPLE']._Message(STATUS, {'totalTime': 0.4, 'attempts': 1,
  'timeIncrement': 0.1, 'increment': 4, 'stepTime': 0.4, 'step': 1,
  'jobName': 'EXAMPLE', 'severe': 0, 'iterations': 1,
  'phase': STANDARD_PHASE, 'equilibrium': 1})
mdb.jobs['EXAMPLE']._Message(ODB_FRAME, {'phase': STANDARD_PHASE,
'step': 0,
  'frame': 5, 'jobName': 'EXAMPLE'})
mdb.jobs['EXAMPLE']._Message(STATUS, {'totalTime': 0.5, 'attempts': 1,
  'timeIncrement': 0.1, 'increment': 5, 'stepTime': 0.5, 'step': 1,
  'jobName': 'EXAMPLE', 'severe': 0, 'iterations': 1,
  'phase': STANDARD_PHASE, 'equilibrium': 1})
mdb.jobs['EXAMPLE']._Message(ODB_FRAME, {'phase': STANDARD_PHASE,
'step': 0,
  'frame': 6, 'jobName': 'EXAMPLE'})
mdb.jobs['EXAMPLE']._Message(STATUS, {'totalTime': 0.6, 'attempts': 1,
  'timeIncrement': 0.1, 'increment': 6, 'stepTime': 0.6, 'step': 1,
  'jobName': 'EXAMPLE', 'severe': 0, 'iterations': 1,
  'phase': STANDARD_PHASE, 'equilibrium': 1})
mdb.jobs['EXAMPLE']._Message(ODB_FRAME, {'phase': STANDARD_PHASE,
'step': 0,
  'frame': 7, 'jobName': 'EXAMPLE'})
mdb.jobs['EXAMPLE']._Message(STATUS, {'totalTime': 0.7, 'attempts': 1,
  'timeIncrement': 0.1, 'increment': 7, 'stepTime': 0.7, 'step': 1,
  'jobName': 'EXAMPLE', 'severe': 0, 'iterations': 1,
  'phase': STANDARD_PHASE, 'equilibrium': 1})
mdb.jobs['EXAMPLE']._Message(ODB_FRAME, {'phase': STANDARD_PHASE,
'step': 0,
  'frame': 8, 'jobName': 'EXAMPLE'})
mdb.jobs['EXAMPLE']._Message(STATUS, {'totalTime': 0.8, 'attempts': 1,
  'timeIncrement': 0.1, 'increment': 8, 'stepTime': 0.8, 'step': 1,
  'jobName': 'EXAMPLE', 'severe': 0, 'iterations': 1,
  'phase': STANDARD_PHASE, 'equilibrium': 1})
mdb.jobs['EXAMPLE']._Message(ODB_FRAME, {'phase': STANDARD_PHASE,
'step': 0,
  'frame': 9, 'jobName': 'EXAMPLE'})
```

```
mdb.jobs['EXAMPLE']._Message(STATUS, {'totalTime': 0.9, 'attempts': 1,
    'timeIncrement': 0.1, 'increment': 9, 'stepTime': 0.9, 'step': 1,
    'jobName': 'EXAMPLE', 'severe': 0, 'iterations': 1,
    'phase': STANDARD_PHASE, 'equilibrium': 1})
mdb.jobs['EXAMPLE']._Message(ODB_FRAME, {'phase': STANDARD_PHASE,
'step': 0,
    'frame': 10, 'jobName': 'EXAMPLE'})
mdb.jobs['EXAMPLE']._Message(STATUS, {'totalTime': 1.0, 'attempts': 1,
    'timeIncrement': 0.1, 'increment': 10, 'stepTime': 1.0, 'step': 1,
    'jobName': 'EXAMPLE', 'severe': 0, 'iterations': 1,
    'phase': STANDARD_PHASE, 'equilibrium': 1})
mdb.jobs['EXAMPLE']._Message(END_STEP, {'phase': STANDARD_PHASE, 'stepId':
1,
    'jobName': 'EXAMPLE'})
mdb.jobs['EXAMPLE']._Message(COMPLETED, {'phase': STANDARD_PHASE,
    'message': 'Analysis phase complete', 'jobName': 'EXAMPLE'})
mdb.jobs['EXAMPLE']._Message(JOB_COMPLETED, {
    'time': 'Sun May 13 15:10:48 2012', 'jobName': 'EXAMPLE'})
# Save by mgemln on Sun May 13 15:12:14 2012
from job import *
from part import *
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from sketch import *
from visualization import *
from connectorBehavior import *
mdb.models['Model-1'].loads['Load-1'].setValues(magnitude=-1e-05)
# Save by mgemln on Sun May 13 15:20:46 2012
```

```
第一个输出文件: EXAMPLE_ODB.py
# -*- coding: mbcs -*-
#
# Abaqus/CAE Release 6.10-1 replay file
# Internal Version: 2010_04_29-14.17.36 102575
# Run by mgemln on Sun May 13 16:37:05 2012
# from driverUtils import executeOnCaeGraphicsStartup
# executeOnCaeGraphicsStartup()
#: Executing "onCaeGraphicsStartup()" in the site directory ...
from abaqus import *
from abaqusConstants import *
session.Viewport(name='Viewport: 1', origin=(0.0, 0.0),
width=261.944432258606,
height=178.915559887886)
session.viewports['Viewport: 1'].makeCurrent()
session.viewports['Viewport: 1'].maximize()
import sys
sys.path.append(r'C:\ABAQUS\6.10-1\abaqus_plugins')
sys.path.append(r'C:\ABAQUS\6.10-1\abaqus_plugins\abq_vcct')
import vcctKernelInit
import vcctInputFileWriter
import vcctInputFileReader
from caeModules import *
from driverUtils import executeOnCaeStartup
executeOnCaeStartup()
session.viewports['Viewport: 1'].partDisplay.geometryOptions.setValues(
referenceRepresentation=ON)
o1 = session.openOdb(
name='E:/zhouke/work/Abaqus/Abaqus-examples/learn abaqus script in
one hour/EXAMPLE.odb')
session.viewports['Viewport: 1'].setValues(displayedObject=o1)
#: Model: E:/zhouke/work/Abaqus/Abaqus-examples/learn abaqus script in one
hour/EXAMPLE.odb
#: Number of Assemblies: 1
#: Number of Assembly instances: 0
#: Number of Part instances: 1
#: Number of Meshes: 1
#: Number of Element Sets: 2
#: Number of Node Sets: 2
#: Number of Steps: 1
session.viewports['Viewport: 1'].odbDisplay.display.setValues(plotState=(
CONTOURS_ON_DEF, ))
session.printToFile(fileName='EXAMPLE', format=TIFF, canvasObjects=(
session.viewports['Viewport: 1'], ))
```

第一个模型文件调整

```

# -*- coding: mbcs -*-
from job import *
from part import *
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from sketch import *
from visualization import *
from connectorBehavior import *

h=0.2
w=0.1
mdb.models['Model-1'].ConstrainedSketch(name='__profile__', sheetSize=20.0)
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(-h, -w), point2=
(h, -w))
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, -w))
mdb.models['Model-1'].sketches['__profile__'].HorizontalConstraint(
    addUndoState=False, entity=
    mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, -w),
))
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(h, -w), point2=(
h, w))
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((h, 0.0))
mdb.models['Model-1'].sketches['__profile__'].VerticalConstraint(addUndoState
=
False, entity=
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((h, 0.0),
))
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, -w))
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((h, 0.0))
mdb.models['Model-1'].sketches['__profile__'].PerpendicularConstraint(
    addUndoState=False, entity1=
    mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, -w),
), entity2=mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((
h, 0.0), ))
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(h, w), point2=(
-h, w))
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, w))
mdb.models['Model-1'].sketches['__profile__'].HorizontalConstraint(

```



```

        addUndoState=False, entity=
        mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, w),
        ))
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((h, 0.0))
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, w))
mdb.models['Model-1'].sketches['__profile__'].PerpendicularConstraint(
    addUndoState=False, entity1=
        mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((h, 0.0), )
    , entity2=mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((
    0.0, w), ))
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(-h, w), point2=(
    -h, -w))
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((-h, 0.0))
mdb.models['Model-1'].sketches['__profile__'].VerticalConstraint(addUndoState
=
    False, entity=
        mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((-h, 0.0),
        ))
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, w))
mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((-h, 0.0))
mdb.models['Model-1'].sketches['__profile__'].PerpendicularConstraint(
    addUndoState=False, entity1=
        mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((0.0, w), )
    , entity2=mdb.models['Model-1'].sketches['__profile__'].geometry.findAt((
    -h, 0.0), ))
mdb.models['Model-1'].Part(dimensionality=TWO_D_PLANAR, name='Part-1',
type=
    DEFORMABLE_BODY)
mdb.models['Model-1'].parts['Part-1'].BaseShell(sketch=
    mdb.models['Model-1'].sketches['__profile__'])
del mdb.models['Model-1'].sketches['__profile__']
mdb.models['Model-1'].Material(name='Material-1')
mdb.models['Model-1'].materials['Material-1'].Elastic(table=((1000000000.0,
    0.3), ))
mdb.models['Model-1'].HomogeneousSolidSection(material='Material-1',
name=
    'Section-1', thickness=None)
mdb.models['Model-1'].parts['Part-1'].SectionAssignment(offset=0.0,
    offsetField='', offsetType=MIDDLE_SURFACE, region=Region(
    faces=mdb.models['Model-1'].parts['Part-1'].faces.findAt(((0.0,
    0.0, 0.0), (0.0, 0.0, 1.0)), ), sectionName='Section-1',
    thicknessAssignment=FROM_SECTION)
mdb.models['Model-1'].parts['Part-1'].Set(edges=
    mdb.models['Model-1'].parts['Part-1'].edges.findAt(((h, w/2, 0.0), ),

```

```

    name='Set-1')
mdb.models['Model-1'].parts['Part-1'].Surface(name='Surf-1', side1Edges=
    mdb.models['Model-1'].parts['Part-1'].edges.findAt(((h/2, w, 0.0), )))
mdb.models['Model-1'].parts['Part-1'].setMeshControls(elemShape=QUAD,
regions=
    mdb.models['Model-1'].parts['Part-1'].faces.findAt(((0.0, 0.0,
0.0), )), technique=STRUCTURED)
mdb.models['Model-1'].parts['Part-1'].seedPart(deviationFactor=0.1, size=h/4)
mdb.models['Model-1'].parts['Part-1'].generateMesh()
mdb.models['Model-1'].parts['Part-1'].setElementType(elemTypes=(ElemType(
    elemCode=CPS8R, elemLibrary=STANDARD), ElemType(elemCode=CPS6M,
    elemLibrary=STANDARD)), regions=(
    mdb.models['Model-1'].parts['Part-1'].faces.findAt(((0.0, 0.0,
0.0), )), ))
mdb.models['Model-1'].rootAssembly.DatumCsysByDefault(CARTESIAN)
mdb.models['Model-1'].rootAssembly.Instance(dependent=ON,
name='Part-1-1',
    part=mdb.models['Model-1'].parts['Part-1'])
mdb.models['Model-1'].StaticStep(initialInc=0.1, maxInc=0.1, name='Step-1',
    nlgeom=ON, previous='Initial')
mdb.models['Model-1'].DisplacementBC(amplitude=UNSET,
createStepName='Step-1',
    distributionType=UNIFORM, fieldName='', fixed=OFF, localCsys=None,
name=
    'BC-1', region=
    mdb.models['Model-1'].rootAssembly.instances['Part-1-1'].sets['Set-1'],
u1=
    0.0, u2=0.0, ur3=0.0)
mdb.models['Model-1'].Pressure(amplitude=UNSET, createStepName='Step-1',
    distributionType=UNIFORM, field='', magnitude=-1e-05, name='Load-1',
region=

mdb.models['Model-1'].rootAssembly.instances['Part-1-1'].surfaces['Surf-1'])
mdb.Job(atTime=None, contactPrint=OFF, description="", echoPrint=OFF,
    explicitPrecision=SINGLE, getMemoryFromAnalysis=True, historyPrint=OFF,
    memory=50, memoryUnits=PERCENTAGE, model='Model-1',
modelPrint=OFF,
    multiprocessingMode=DEFAULT, name='EXAMPLE',
nodalOutputPrecision=SINGLE,
    numCpus=1, queue=None, scratch="", type=ANALYSIS, userSubroutine="",
    waitHours=0, waitMinutes=0)
mdb.jobs['EXAMPLE'].submit(consistencyChecking=OFF)
mdb.jobs['EXAMPLE'].waitForCompletion()
####out put###

```

```
execfile('EXAMPLE_ODB.py')
```

米盖尔莫林纳