Open Abaqus (**Since Matlab opens abaqus, you may need to do this after you create the Model in Matlab**)

Before starting creating the model we will enter a line in the Script window. Click the Script button to go to the Script Window. Enter the following line

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

Create Model – in Matlab:

Run main.m – Can .mat files be inputs for GridSpace?

%Created by J.T.B. Overvelde

%2012/05/03

%http://www.overvelde.com

clear, close all, clc;

%Variables

GridSpaceX=1;

GridSpaceY=1;

%mo='noGUI';

mo='script';

%Make python file with variables

delete('Var.py');

fid = fopen('Var.py', 'w');

fprintf(fid,'GridSpaceX = %0.12f\n',GridSpaceX);

fprintf(fid,'GridSpaceY = %0.12f\n',GridSpaceY);

fclose(fid);

%Make part(run Abaqus)

system(['abaqus cae ',mo,'=Main.py']); %Windows system?

Python Scripting – call it FILENAME.py (or whatever you want):

#Created by J.T.B. Overvelde

#2012/05/03

#http://www.overvelde.com

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 job import \*

from sketch import \*

from visualization import \*

from connectorBehavior import \*

from Numeric import \*

from abaqus import \*

from abaqusConstants import \*

import visualization

import os

import datetime

import shutil

from odbAccess import \*

#Load variables

execfile('Var.py')

### PART ###

#Create rentangle with variable size

mdb.models['Model-1'].ConstrainedSketch(name='\_\_profile\_\_', sheetSize=10.0)

mdb.models['Model-1'].sketches['\_\_profile\_\_'].rectangle(point1=(0.0, 0.0), point2=(GridSpaceX, GridSpaceY))

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\_\_']

### MATERIAL & SECTION ###

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') #####################change

### SET & SURFACE ###

mdb.models['Model-1'].parts['Part-1'].Set(edges=mdb.models['Model-1'].parts['Part-1'].edges.findAt(((-width/2.0, -height/4.0, 0.0), )),name='Set-1') ####################change

mdb.models['Model-1'].parts['Part-1'].Surface(name='Surf-1', side1Edges=mdb.models['Model-1'].parts['Part-1'].edges.findAt(((-width/4.0, height/2.0, 0.0), ))) ####################change

### MESH ###

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) ####################change

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), )), )) ####################change

mdb.models['Model-1'].parts['Part-1'].seedPart(deviationFactor=0.1, size=height/4) ####################change

mdb.models['Model-1'].parts['Part-1'].generateMesh()

### ASSEMBLY ###

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'].rootAssembly.regenerate()

### STEP, BC & LOAD ###

mdb.models['Model-1'].StaticStep(initialInc=0.1, maxInc=0.1, name='Step-1', 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=-100000.0, name='Load-1', region= mdb.models['Model-1'].rootAssembly.instances['Part-1-1'].surfaces['Surf-1'])

### JOB & CALCULATE ###

mdb.Job(contactPrint=OFF, description='', echoPrint=OFF, explicitPrecision=SINGLE, historyPrint=OFF, memory=90, memoryUnits=PERCENTAGE, model='Model-1', modelPrint=OFF, multiprocessingMode=DEFAULT, name='EXAMPLE',nodalOutputPrecision=SINGLE, numCpus=1, numDomains=1,parallelizationMethodExplicit=DOMAIN, scratch='', type=ANALYSIS, userSubroutine='')

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

mdb.jobs['EXAMPLE'].waitForCompletion() ####################change

### OUPUT ###

execfile('EXAMPLE\_ODB.py') ####################change

Before running this script, delete all files except `FILENAME.py' from your working directory.

If your GUI was still open start a new model and don't save anything. Now to run the script \_le go

to the Top menu click on 'File' and then on `Run script...' and select your script \_le. If everything

is done correctly your model should run without any problems. Make sure the file `EXAMPLE.odb'

exist in your working directory. If you wouldn't delete the \_les in your working directory Abaqus

would save over them. This is no problem, but for the purpose of this manual you want to see if

the files were created.

**Create your first script file for Output Database (odb) – 2 different examples**

Save the file `Abaqus.rpy' as `EXAMPLE ODB.py'. The content of this script file is given in the

attachment. Below is a short ordering of the script file.

**1) First EXAMPLE ODB.py'**

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

#

# Abaqus/CAE Version 6.8-2 replay file

# Internal Version: 2008\_07\_21-07.21.56 87172

# Run by overveld on Wed Nov 17 23:00:22 2010

#

# 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=268.952117919922, height=154.15299987793)

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

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

from caeModules import \*

from driverUtils import executeOnCaeStartup

executeOnCaeStartup()

o1 = session.openOdb(name='/home/overveld/EXAMPLE.odb')

session.viewports['Viewport: 1'].setValues(displayedObject=o1)

#: Model: /home/overveld/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'], ))

**2) Final EXAMPLE ODB.py'**

#open modulus, create viewport and open odb

from abaqus import \*

from abaqusConstants import \*

session.Viewport(name='Viewport: 1', origin=(0.0, 0.0), width=268.952117919922, height=154.15299987793)

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

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

from caeModules import \*

from driverUtils import executeOnCaeStartup

executeOnCaeStartup()

o1 = session.openOdb(name='/home/overveld/EXAMPLE.odb')

session.viewports['Viewport: 1'].setValues(displayedObject=o1)

### CREATE OUTPUT ###

session.viewports['Viewport: 1'].odbDisplay.display.setValues(plotState=(CONTOURS\_ON\_DEF, ))

session.printToFile(fileName='EXAMPLE', format=TIFF, canvasObjects=(session.viewports['Viewport: 1'], ))

**Using the GUI to create output**

Close and open Abaqus CAE to restart recording of the script. Now open the the file `EXAM-

PLE.odb' by clicking in the Top menu on `File' and then on `Open..'. Select your `EXAMPLE.odb'

from your working directory. We will use the same method as for creating the script file for the

model. The most important difference is that the actions you do are recorded in `Abaqus.rpy' and

not in `EXAMPLE.jnl'. Let us create a figure of the stress in the deformed state. Do the following

step:

Plot the stress in the deformed state. Now save the figure with a .tiff format by clicking on

the 'File' and then 'Print...' in the Top menu. The name is EXAMPLE.

You don't have to save the odb file. Make sure the file `Abaqus.rpy' exist in your working directory.

If you can't find it take a look into your initial working directory that is opened during start up of

Abaqus CAE.