Optimization with CalculiX

Jeff Baylor – jbaylor@bConverged.com
Convergent Mechanical

Overview

This document summarizes methods to optimize with CalculiX. The focus is on the optimization process itself and not the generating the FEA model.

There are many optimization tools available. Among the open source solutions there are a few applications that focus entirely on optimization, including: Nomad and Dakoda. There are also numerical math environments akin to the commercial application Matlab with optimization capabilities. Most notable are: Octave, SciLab and SciPy. This document will focus on SciPy which is an extension of the Python scripting language to include many mathematical analysis tools.

Setting up SciPy

I run PySci using the Eclipse development environment (http://www.eclipse.org) with the PyDev plug-in (http://pydev.org). There are much simpler ways of using it, but I do a fair amount of Java development too, and Eclipse is my favorite development environment for Java. SciPy needs Python, Numpy and I also install matplotlib for plotting. You can find them at http://www.python.org and http://www.scipy.org

Another solution is to use Python(x,y) (http://www.pythonxy.com). While I have heard good things about it, I have not tried it yet. One of the conveniences seems to be that it is a bundle of all that you need to start using SciPy. It contains a build of Eclipse with PyDev as well as another IDE called Spyder.

There is a cookbook example of using optimization in SciPy at:
http://www.scipy.org/Cookbook/OptimizationDemo1

optimization.py

```
import os
import sys
from scipy import optimize
from numpy import *
from pylab import *

def objective(variables):
    var1,var2 = variables
    defFile = open(jobName, 'w')
    # write the model definition based on var1 and var2
    # I do one of three things here
    #   1) I write a *PARAMETER section of an INP file and then call the Abaqus to CCX translator
    #      in the solve.bat. One could also write an FBD file using this section and call the
    #      same translator before sending it to CGX in the solve.bat
    #   2) I write Gmsh GEO file with parameters then call Gmsh in the solve.bat file.  Then call
    #      the translator from Gmsh to CCX before calling CCX (also in the solve.bat file).
    #   3) For very simple geometries, I have also just written the mesh, or the fully defined GEO
    #      or FBD without parameters.
    #
    defFile.close()

    command = "solve.bat "+jobName
    print("executing: %s\n" % command)
    os.system(command)

    resFile = open(jobName+'.dat', 'r')
    # extract data from DAT file and determine objective function result (I am assuming it is in a
    # variable named result). I actually write the DAT extraction in C++ and call it the solve.bat,
    # but you can do it here just as easily.
    #
    # If more than one result is to be optimized, I usually minimize the sum of the squares. If
    # those values are not the same magnitude, or same importance, I weigh them as I combine the
    # squares.
    #
    resFile.close()

    optFile = open(jobName+'.opt', 'a')
    optFile.write('%f,%f,%f\n' % (var1,var2,result))
    optFile.close()

    return result

if __name__ == '__main__':
    #  initial parameters:
    v0 = [initVar1,initVar2]
    v = optimize.fmin(objective, v0, maxiter=100, maxfun=100)
```

solve.bat

```
@echo off
set CALCULIX_ROOT=C:\PROGRA~1\BCONVE~2
set TRANS_ROOT=C:\projects\translators
set PATH=%CALCULIX_ROOT%\CalculiX\bin;%CALCULIX_ROOT%\common\Python;%PATH%
set PATH=%CALCULIX_ROOT%\translation\occ;%CALCULIX_ROOT%\translation\bin;%PATH%
set JOB=%1%

del *.msh
del *.nam

rem ---  example of a problem when the full FBD file was written in the SciPy objective function
%CALCULIX_ROOT%\calculix\cgx\cgx -b %JOB%.fbd
rem ---  There are times when the element type needs to be changed
%TRANS_ROOT%\transCAEinput\Release\transCAEinput -ccx all.msh -etype S8 CAX8 all_cax8.msh
rem ---  Call CCX
%CALCULIX_ROOT%\calculix\ccx\ccx %JOB%
rem ---  If DAT file is parsed externally, call it here to write the results to a file: jobName.out
%TRANS_ROOT%\parseDat\Release\parseDat.exe %JOB%.dat %JOB%.out
```

Reference

SciPy Related References

Python - http://www.python.org
SciPy - http://www.scipy.org
Eclipse - http://www.eclipse.org
Eclipse PyDev plug-in http://pydev.org
Python(x,y) - http://www.pythonxy.com

Other References

CalculiX for Windows - http://bConverged.com/calculix
CalculiX for Linux/UNIX - http://calculix.de

Dakota - http://www.cs.sandia.gov/dakota/software.html
Nomad - http://www.gerad.ca/nomad/Project/Home.html
Octave - http://www.gnu.org/software/octave/
SciLab - http://www.scilab.org/