# Running OpenFoam cases in batch mode under the control of a python3 program

A project using OpenFoam simulation may involve a lot of different cases, or cases which take a long time to run, or – pity you – both. Managing the process is a disruptive task. The disruption can be reduced if the case directories are all prepared before-hand and then some other program used to run the cases one after another, as a batch. Because OpenFoam is a Linux-based application, it makes sense to write the control program in the python3 language. The python3 program I use is listed in Appendix "A". The following paragraphs describe, as an example, the data set which the listed program runs.

An OpenFoam case is a self-contained directory containing all of the files needed for one complete run of an OpenFoam solver. The files in this directory, which is called the "case directory", include descriptions of the geometry of the fluid, the nature of and initial conditions on the boundary surfaces, identification of the solver and turbulence models, identification of certain numerical routines to use, and so on. To run the case, one opens a terminal screen, changes to the case directory, and executes the solver.

Suppose a project involves 14 cases. The 14 case directories can be collected into a higher-level directory, as follows. The names shown here happen to be the ones referred to in the program listed in Appendix "A".

```
Project_Directory/
|
|---T_10cm_0deg_40mph_FaceOn_15000ft_6Across_kwSST/
|---T_10cm_15deg_40mph_FaceOn_15000ft_6Across_kwSST/
|---T_10cm_30deg_40mph_FaceOn_15000ft_6Across_kwSST/
|---T_10cm_45deg_40mph_FaceOn_15000ft_6Across_kwSST/
|---T_10cm_60deg_40mph_FaceOn_15000ft_6Across_kwSST/
|---T_10cm_75deg_40mph_FaceOn_15000ft_6Across_kwSST/
|---T_10cm_90deg_40mph_FaceOn_15000ft_6Across_kwSST/
|---T_12.5cm_0deg_40mph_FaceOn_15000ft_6Across_kwSST/
|---T_12.5cm_15deg_40mph_FaceOn_15000ft_6Across_kwSST/
|---T_12.5cm_30deg_40mph_FaceOn_15000ft_6Across_kwSST/
|---T_12.5cm_45deg_40mph_FaceOn_15000ft_6Across_kwSST/
|---T_12.5cm_60deg_40mph_FaceOn_15000ft_6Across_kwSST/
|---T_12.5cm_75deg_40mph_FaceOn_15000ft_6Across_kwSST/
|---T_12.5cm_90deg_40mph_FaceOn_15000ft_6Across_kwSST/
|---ofBatch.py
```

In addition to the 14 case directories, the project directory contains the python3 program itself. I use the name ofBatch for the program, where the prefix stands for <u>O</u>pen<u>F</u>oam. The extension .py identifies the file as a python3 executable.

Running the 14 cases as a batch is done by opening a new terminal screen, changing to the project directory and executing ofBatch.py. On the screen, the procedure looks like this:

```
jim@jim-CG8270:~$ cd Desktop
jim@jim-CG8270:~/Desktop$ cd Project_Directory
jim@jim-CG8270:~/Desktop/Project_Directory$ python3 ofBatch.py
```

On my computer, the project directory Project_Directory/ is located on the desktop. Your Linux prompt will, of course, be different.

Let's look at the details of ofBatch. As preliminary housekeeping, the program stores the name of the project directory. It stores the names of the case directories in a list called CaseNames[]. It makes a quick check to ensure that all of the case directories exist. This prevents a typing mistake from interrupting the batch.

The actual batch processing is done in the loop "`#Start of main loop`". The OpenFoam cases in this project all require parallel processing, so the main loop has five steps. They are:

Step #1: Change to the next case directory
Step #2: Decompose the case
Step #3: Run the case
Step #4: Reconstruct the case
Step #5: Change back to the project directory

The loop continues until all of the cases have been run.

Let me point out that OpenFoam runs the cases in exactly the same way as they would run had they been started manually. The decomposition and reconstruction of parallel cases happens in exactly the same way, too. Anything one can do when OpenFoam is started manually can also be done when OpenFoam is started programmatically. For example, one can change the settings in the /system/fvSolution file to change the convergence parameters "on-the-go", so to speak. One can also direct a copy of the output on the terminal screen to a file. In the program listed, the output on the terminal screen is copied to a file named ofLog.txt.

If something goes wrong during batch processing, and a case fails for any reason (such as a floating point exception), the python3 program will also terminate. Any previous cases which have already been completed will remain intact, but the failed case will need to be dealt with and any following cases will not have been started.

Running OpenFoam cases in batch mode only makes sense after the solver, supporting procedures, and other parameters of the simulation have been debugged and a number of "production runs", if I can describe them that way, need to executed.

Jim Hawley
June 2013

An email setting out errors and omissions would be appreciated.

## Appendix “A”

## Listing of python3 program ofBatch.py

```
# This is a Python program designed to run a batch of OpenFoam cases.

# Assumptions:
# 1. That this file is named "ofBatch.py";
# 2. That one run will be made for each case directory;
# 3. That the number of runs / case directories is 'NumCases' defined below;
# 4. That the case directories have the names listed below; and
# 5. That "ofBatch.py" is located in the same directory as the case directories.

# Invocation:
# 1. Open a terminal window;
# 2. Change directory to the directory containing "ofBatch.py";
# 3. Type the command "python3 ofBatch.py".

# File names:
NumCases = 14;   # Should equal the number of names exposed in the list.
CaseNames = []
CaseNames.append("T_10cm_0deg_40mph_FaceOn_15000ft_6Across_kwSST")   # Case directory #1
CaseNames.append("T_10cm_15deg_40mph_FaceOn_15000ft_6Across_kwSST")  # Case directory #2
CaseNames.append("T_10cm_30deg_40mph_FaceOn_15000ft_6Across_kwSST")  # Case directory #3
CaseNames.append("T_10cm_45deg_40mph_FaceOn_15000ft_6Across_kwSST")  # Case directory #4
CaseNames.append("T_10cm_60deg_40mph_FaceOn_15000ft_6Across_kwSST")  # Case directory #5
CaseNames.append("T_10cm_75deg_40mph_FaceOn_15000ft_6Across_kwSST")  # Case directory #6
CaseNames.append("T_10cm_90deg_40mph_FaceOn_15000ft_6Across_kwSST")  # Case directory #7
CaseNames.append("T_12.5cm_0deg_40mph_FaceOn_15000ft_6Across_kwSST")   # Case directory #8
CaseNames.append("T_12.5cm_15deg_40mph_FaceOn_15000ft_6Across_kwSST")  # Case directory #9
CaseNames.append("T_12.5cm_30deg_40mph_FaceOn_15000ft_6Across_kwSST")  # Case directory #10
CaseNames.append("T_12.5cm_45deg_40mph_FaceOn_15000ft_6Across_kwSST")  # Case directory #11
CaseNames.append("T_12.5cm_60deg_40mph_FaceOn_15000ft_6Across_kwSST")  # Case directory #12
CaseNames.append("T_12.5cm_75deg_40mph_FaceOn_15000ft_6Across_kwSST")  # Case directory #13
CaseNames.append("T_12.5cm_90deg_40mph_FaceOn_15000ft_6Across_kwSST")  # Case directory #14

# Imports
import os
import subprocess
import sys

# Store the name of the "ofBatch.py" directory in variable BatchDir.
BatchDir = os.getcwd()

# Ensure that all case directories exist.
fnames = os.listdir(BatchDir)
for I in range(0, NumCases):
   CaseDirExists = False
   for fname in fnames:
      if (os.path.isdir(fname)):
         if (fname == CaseNames[I]):
            CaseDirExists = True
   if (CaseDirExists == False):
      print("Error: The case directory " + CaseNames[I] + " was not found.")
      sys.exit()

# Start of main loop
for I in range(0, NumCases):

   # Step 1: Change to the next case directory.
   os.chdir("./" + CaseNames[I])

   # Step 2: decomposePar the case.
   # The cases here run in parallel on 8 processors.
   os.system("decomposePar")

   # Step #3: Run the OpenFoam case.
   # The output is logged to the file named "ofLog.txt".
   os.system("mpirun -np 8 simpleFoam -parallel | tee -a ofLog.txt")
```

```
# Step #4: reconstructPar the case.
os.system("reconstructPar")

# Step 5: Revert to the "ofBatch.py" directory.
os.chdir("../")
```