

CFD with Open Source Software, Final Assignment

Python Scripting and M4-Scripting for Automation and Parameterization in OpenFOAM

Author :
Erwin Adi Hartono

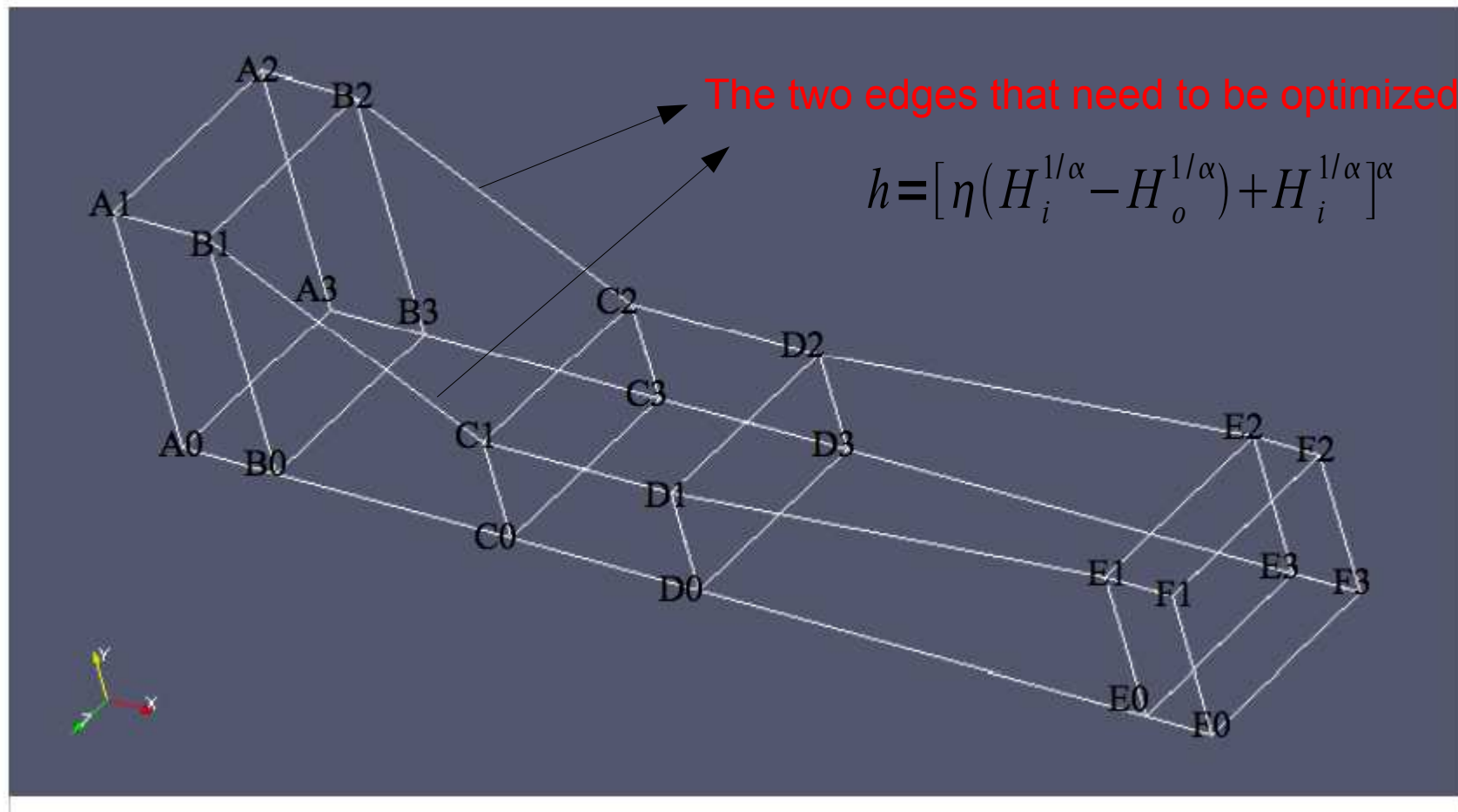
Content :

1. Background
2. The Geometry
3. Parameterization
4. Automatization
5. Result

Background

- In optimization, it is needed to have some set of variations of parameter.
- Each of variations need to be tested and analyzed.
- Do this one by one is a repetition work and time consuming.
- That's why we need automatization.

The Basic Geometry and Vertices



Parameterization



blockMeshDict.m4

We need to parameterize the `blockMeshDict`, so that it is easier to change a single parameter and will be affected into the whole geometry.

Open `blockMeshDict.m4`

Run using

```
m4 -P blockMeshDict.m4 >  
blockMeshDict
```

```
convertToMeters 1;  
  
vertices  
(  
    // front side  
    (xA 0 0) vlabel(A0)  
    (xB 0 0) vlabel(B0)  
    (xC 0 0) vlabel(C0)  
    (xD 0 0) vlabel(D0)  
    (xE 0 0) vlabel(E0)  
    (xF 0 0) vlabel(F0)  
    (xF fanOutletHeight 0) vlabel(F1)  
    (xE diffuserOutletHeight 0) vlabel(E1)  
    (xD testSectionOutletHeight 0) vlabel(D1)  
    (xC contractionOutletHeight 0) vlabel(C1)  
    (xB settlingChamberOutletHeight 0) vlabel(B1)  
    (xA settlingChamberInletHeight 0) vlabel(A1)  
  
    // back side  
    (xA 0 z) vlabel(A3)  
    (xB 0 z) vlabel(B3)  
    (xC 0 z) vlabel(C3)  
    (xD 0 z) vlabel(D3)  
    (xE 0 z) vlabel(E3)  
    (xF 0 z) vlabel(F3)  
    (xF fanOutletHeight z) vlabel(F2)  
    (xE diffuserOutletHeight z) vlabel(E2)  
    (xD testSectionOutletHeight z) vlabel(D2)  
    (xC contractionOutletHeight z) vlabel(C2)  
    (xB settlingChamberOutletHeight z) vlabel(B2)  
    (xA settlingChamberInletHeight z) vlabel(A2)  
  
);  
  
blocks  
(  
    // settlingChamber  
    hex (B3 A3 A2 B2 B0 A0 A1 B1) (10 40 1) simpleGrading (1 0.05 1)  
    // contraction  
    hex (C3 B3 B2 C2 C0 B0 B1 C1) (50 40 1) simpleGrading (1 0.05 1)  
    // testSection  
    hex (D3 C3 C2 D2 D0 C0 C1 D1) (50 40 1) simpleGrading (1 0.05 1)  
    // diffuser  
    hex (E3 D3 D2 E2 E0 D0 D1 E1) (60 40 1) simpleGrading (1 0.05 1)  
    // fan
```

Automatization

This automatization procedure is done by using python script.

To see more detail open `erwin.py`

To run the script:
In terminal window in `windTunnel` directory type `./erwin.py`

→ `erwin.py`

```
#!/usr/bin/python

import sys, os, shutil, math, commands, glob

from pylab import *

from PyFoam.Execution.UtilityRunner import UtilityRunner
from PyFoam.Execution.BasicRunner import BasicRunner
from PyFoam.RunDictionary.SolutionDirectory import SolutionDirectory
from PyFoam.RunDictionary.BlockMesh import BlockMesh

from os import path

from subprocess import Popen
from subprocess import call

orig=SolutionDirectory(path.expandvars("$FOAM_RUN/windTunnel/baseline"),archive=None,paraview=
lse)

alphas = [-0.25, 0.25, 1000]

for alpha in alphas:

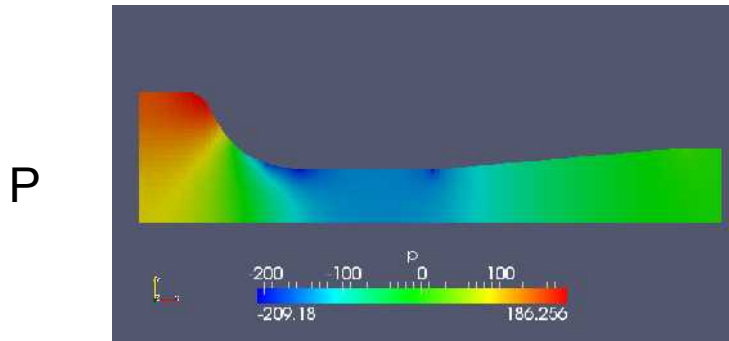
    print "alpha:",alpha
    # clone the original case
    case=orig.cloneCase("modified%f" % alpha).name

    print "m4 -F blockMeshDict.m4 > blockMeshDict"
    # cd into case directory
    os.chdir(case)

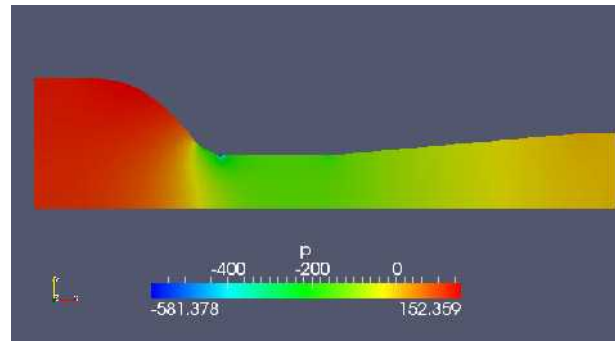
    # change the alpha value inside currentAlpha
    infilename='constant/polyMesh/currentAlpha'
    outfilename='constant/polyMesh/currentAlphaTemp'
    ifile = open( infilename, 'r')
    ofile = open(outfilename, 'w')
    lines = ifile.readlines()
    ofile.write(str(alpha))
    ifile.close()
    ofile.close()
    os.remove(infilename)
    os.rename(outfilename,infilename)
```

Result

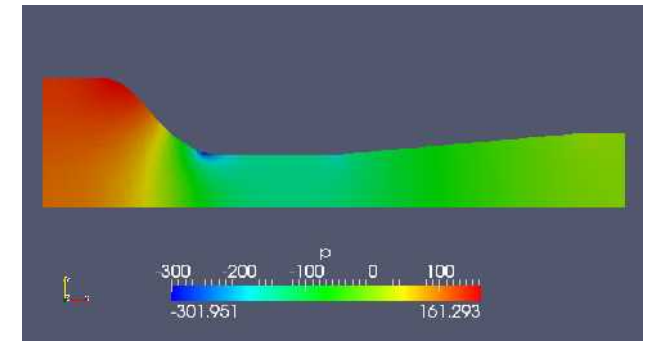
Alpha = -0.25



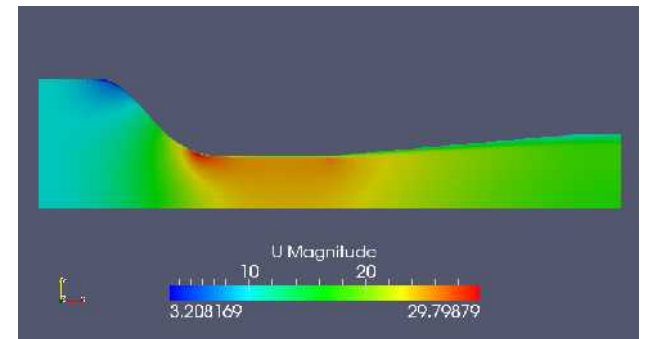
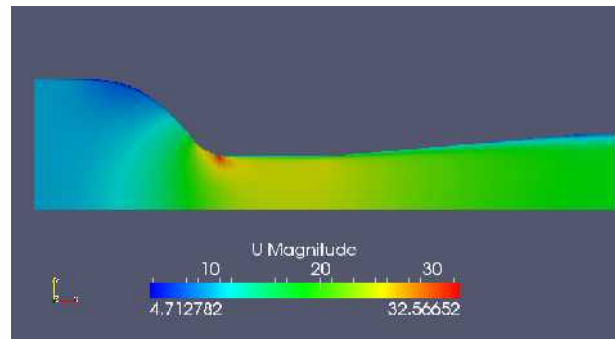
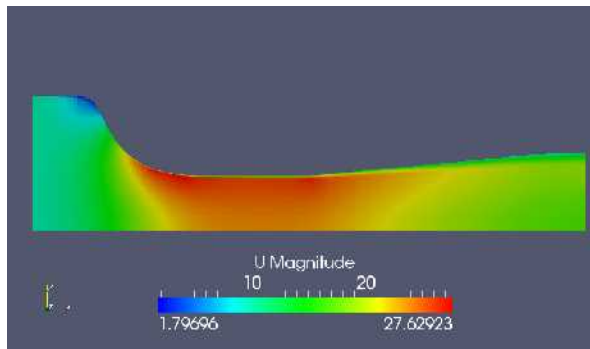
Alpha = 0.25



Alpha = 1000



U



THANK YOU FOR LISTENING

Questions ?
Comments ?