CFD Optimization tutorial

Author: Björn Bergqvist (creator of Discretizer- http://www.discretizer.org).

Aquiring software

For this tutorial we need:

1. OpenFOAM, CFD solver
2. Discretizer, parametric hex mesher
3. Opt4j with external solver plugin

OpenFOAM

Check instructions from http://www.opencfd.co.uk

Discretizer

Download http://www.discretizer.org/files/discretizer-32bit.tar.gz or http://www.discretizer.org/files/discretizer-64bit.tar.gz

Opt4j

Download from http://opt4j.sourceforge.net/

External solver plugin: http://www.discretizer.org/files/opt4j.external-2.1.jar (put in plugins folder)

Geometry creation

A parametric model is needed. In Discretizer a geometry is created (Figure 1). Selection of geometry for optimization is not important. This its the method used. 5 points in the geometry are going to get optimized positions.

Run script

Opt4j external solver interface gives arguments to a script or program and expects a value in return.Since discretizer comes with a ruby runtime it't going to be used. Shell script or Perl would also work fine. It is important that stderr also get redirected so opt4j don't take error messages as a result. The script:

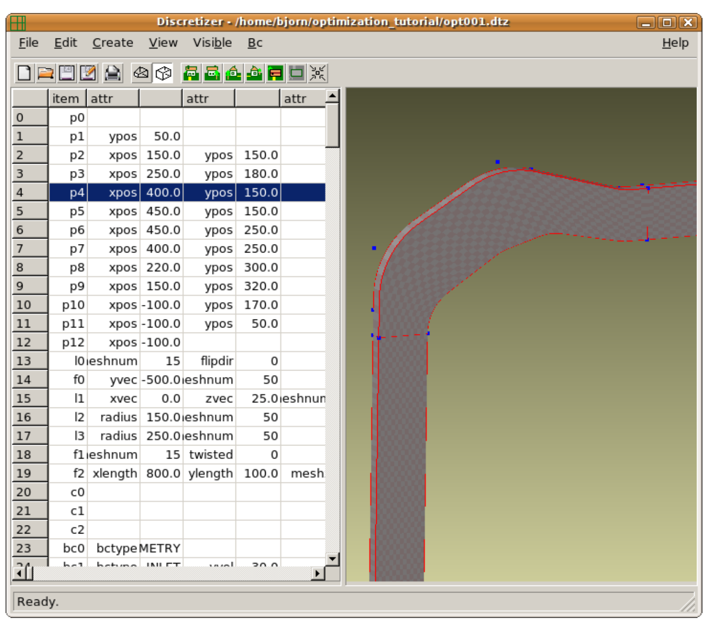

Figure 1: The geometry which is going to be optimized.

```
#!/home/bjorn/optimization_tutorial/discretizer/ruby/1.9.1-p243/bin/ruby
# input variables varies between 0 and 1
# line1
x1 = $*[0].to_f*100+100
y1 = $*[1].to_f*100+100
x2 = $*[2].to_f*100+200
y2 = $*[3].to_f*100+130
# line 2
x3 = $*[4].to_f*100+170
y3 = $*[5].to_f*100+250
x4 = $*[6].to_f*100
y4 = $*[7].to_f*100+270
x5 = $*[8].to_f*100-150
y5 = $*[9].to_f*100+120
begin
   system("./discretizer/discretizer64.sh opt001.dtz 2/xpos/#{x1} 2/ypos/#{y1}
3/xpos/#{x2} 3/ypos/#{y2} 8/xpos/#{x3} 8/ypos/#{y3} 9/xpos/#{x4} 9/ypos/#{y4}
10/xpos/#{x5} 10/ypos/#{y5} -exec &> log")
   system("patchAverage p INLET_c0_f5 -case opt001 -latestTime > press.txt 2>log")
   file = File.new("press.txt", "r")
   press = file.read
   file.close
   pressure = press.scan(/\]\s=\s\d*.\d*/).last.split("=").last

   begin
      file_in = File.open("results.csv", "r")
      prevres = file_in.read
      file_in.close
   rescue
      prevres = "press,x1,y1,x2,y2,x3,y3,x4,y4,x5,y5\n"
   end
   file_out = File.open("results.csv", "w")
   file_out.write(prevres)
   file_out.write("#{pressure},#{x1},#{y1},#{x2},#{y2},#{x3},#{y3},#{x4},#{y4},#{x5},#{
y5}\n")
   file_out.close

rescue
   pressure = 10000.0
end
puts pressure


system("rm -rf opt001")
```

Now I'm going to explain what is happening in the script. Lets start with

```
x1 = $*[0].to_f*100+100
```

x1 is assigned the first argument $*[0]. x1 are going to have values between 100 and 200 since input arguments from opt4j will be set to 0 to 1.

Then begin ->rescue->end is used since some simulations are going to fail, then a high value is passed to opt4j.

Discretizer gets some input arguments which will modify the geometry and run the simulation (simpleFoam). OpenFOAM has to be available. So this command: “discretizer opt001.dtz 2/xpos/80 -exec” will modify the xpos value at row 2 to value 80 before the geometry of the opt001.dtz file is created. The -exec flag will run discretizer without gui and also export mesh to directory opt001 and run a simpleFoam calculation.

Next thing is to get the pressure from the inlet boundary. patchAverage is then used which is an OpenFOAM command. The output from the patchAverage command is directed to a file which is parsed:

```
pressure = press.scan(/\]\s=\s\d*.\d*/).last.split("=").last
```
This gives the pressure.

Results from the simulation is put in results.csv which can be opened as a spreadsheet.

In the last row of the script the simulation directory is removed since if the simulation crash there a very high pressure will be returned to opt4j.

Run the optimization

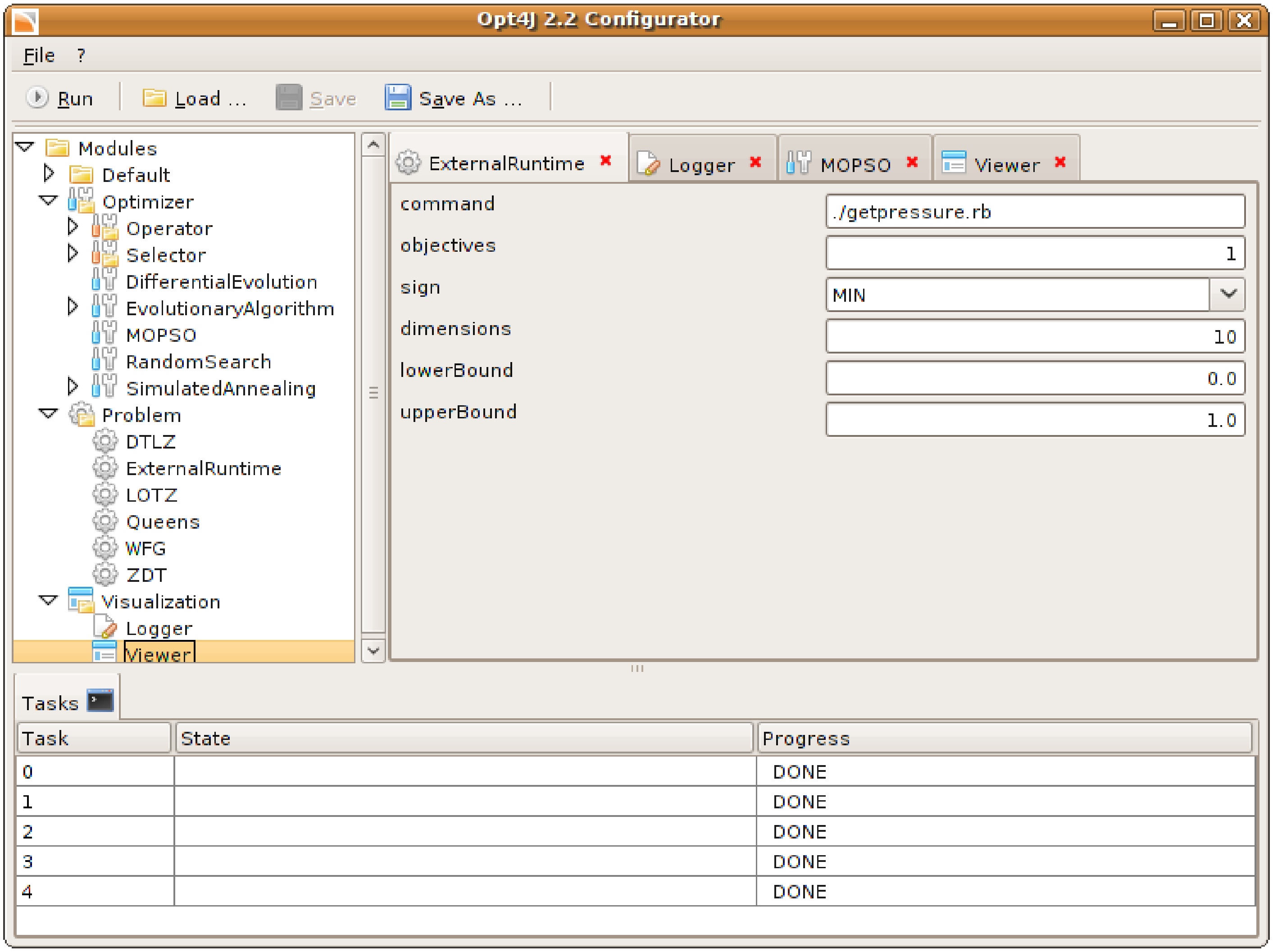

Figure 2: Configuration for this problem with 1 objective and 10 dimensions.

Opt4j is started with java -jar opt4j-2.2.jar. Select an optimization model. I used MOPSO. Also select (double click) logger and viewer.

Results

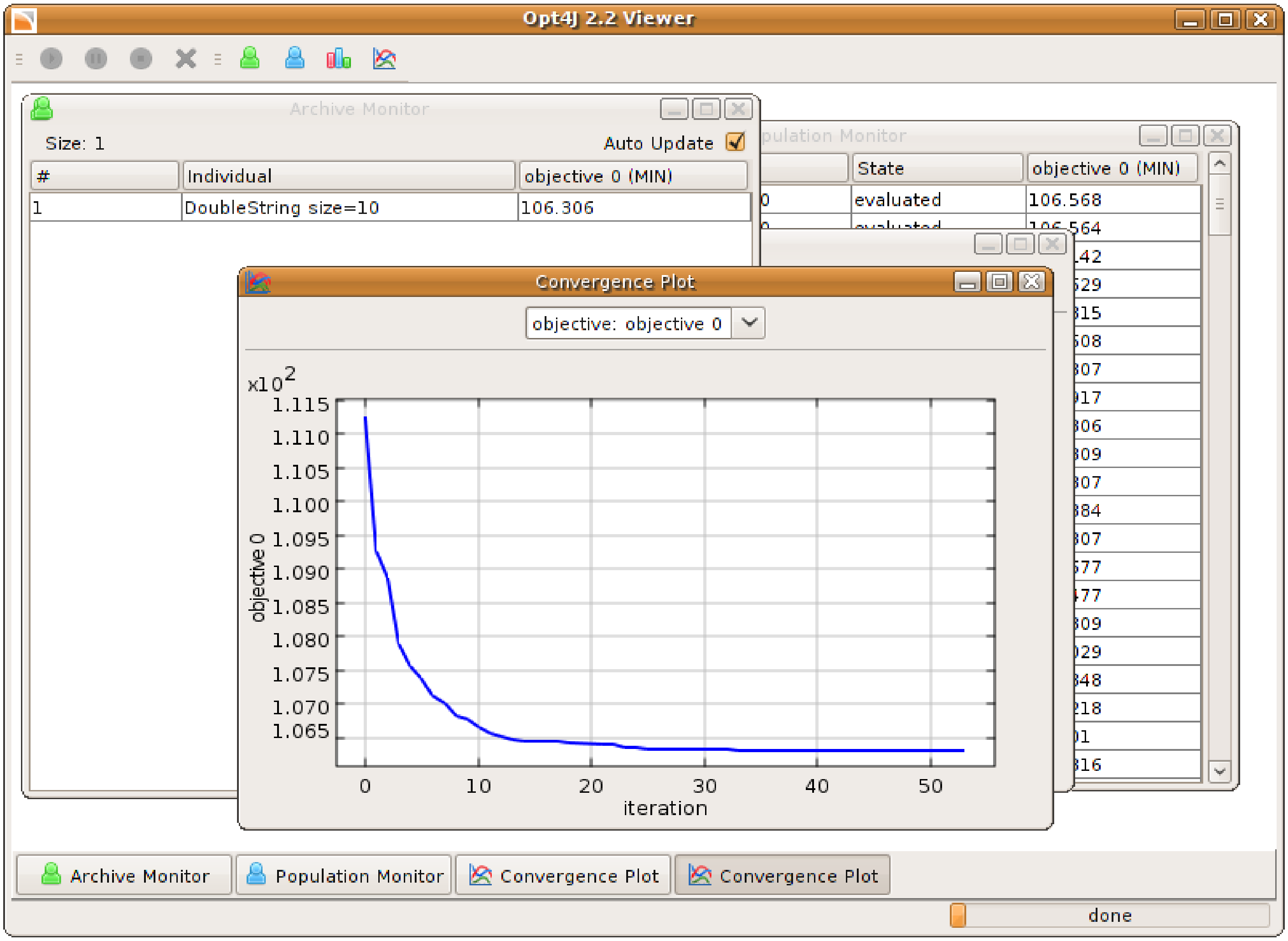

Figure 3: After 50 generation optimization was stopped. Best individual had a pressure drop of 106 Pa.

To compare with an ordinary bend a simulation was performed on a bend with inner radius 150 mm. The pressure drop was 128 on this reference. The optimized geometry has a pressure drop of 106 Pa.

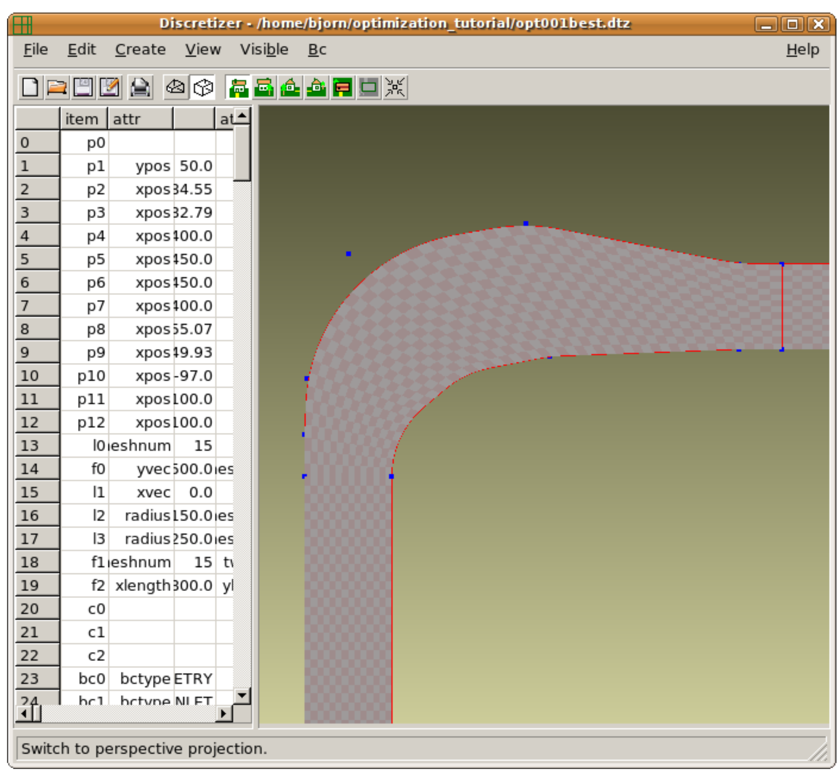

Figure 4: The shape of the optimized geometry.