Imperial College
London

Running Fluidity and Visualising the Results

Jon Hill[1]

1 - Dept of Earth Science and Engineering, Imperial College London

Outline

Running

Output

- Filetypes and tools
- The stat file
- Paraview
- Python

Parallel

- Running
- Post-running

Outline

Running

Output

- Filetypes and tools
- The stat file
- Paraview
- Python

Parallel

- Running
- Post-running

Running Fluidity

```
./fluidity
```

Running Fluidity

```
Revision: fluidity/4.1
Compile date: Nov 20 2012 10:16:12
OpenMP Support no
Adaptivity support yes
...
FEMDEM support no
Hyperlight support no


Usage: fluidity [options ...] [simulation-file]

Options:
 -h, --help
Help! Prints this message.
 -l, --log
Create log file for each process (useful for non-interactive testing).
 -v <level>, --verbose
Verbose output to stdout, default level 0
 -p, --profile
Print profiling data at end of run
This provides aggregated elapsed time for coarse-level computation
(Turned on automatically if verbosity is at level 2 or above)
 -V, --version
Version
```

Running Fluidity

```
./fluidity my.flml
./fluidity -l my.flml
./fluidity -l -v3 my.flml
```

Outline

Running

Output

- Filetypes and tools
- The stat file
- Paraview
- Python

Parallel

- Running
- Post-running

Filetypes

There are **two** main filetypes:

- .stat file
- Unstructured VTK file (.vtu or .pvtu)

You may also have log files:

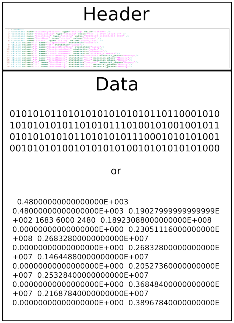
- fluidity.log.*
- fluidity.err.*

Tools

- Statplot
- Paraview
- Python
    - vtktools
    - fluidity.statparser

The stat file

- Bespoke data file type
- Various tools to read and process these data
- Either ASCII or binary



Header

```
<header>
<constant name="FluidityVersion" type="string" value="14689M" />
<constant name="CompileTime" type="string" value="Nov  1 2010 16:28:07" />
<constant name="StartTime" type="string" value="20101102 112633.018+0000" />
<constant name="HostName" type="string" value="Unknown" />
<constant name="format" type="string" value="plain_text" />
<field column="1" name="ElapsedTime" statistic="value"/>
<field column="2" name="dt" statistic="value"/>
<field column="3" name="ElapsedWallTime" statistic="value"/>
<field column="4" name="CoordinateMesh" statistic="nodes"/>
<field column="5" name="CoordinateMesh" statistic="elements"/>
<field column="6" name="CoordinateMesh" statistic="surface_elements"/>
<field column="7" name="TotalMemory" statistic="current" material_phase="Memory"/>
<field column="8" name="TotalMemory" statistic="min" material_phase="Memory"/>
<field column="9" name="TotalMemory" statistic="max" material_phase="Memory"/>
<field column="10" name="MeshMemory" statistic="current" material_phase="Memory"/>
<field column="11" name="MeshMemory" statistic="min" material_phase="Memory"/>
<field column="12" name="MeshMemory" statistic="max" material_phase="Memory"/>
```

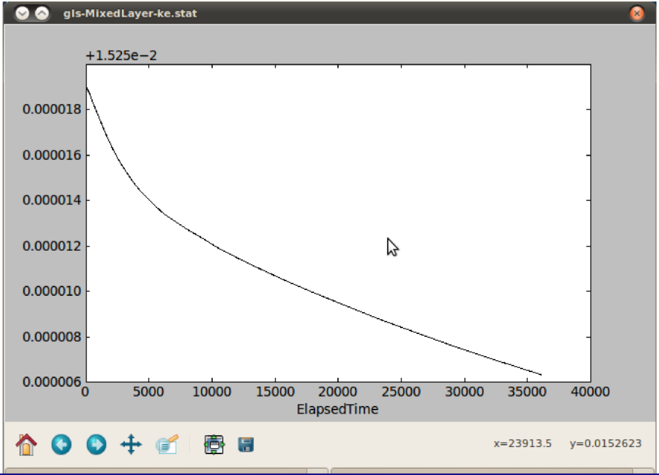
Data

```
0101010110101010101010101101100010 10
101010101011010101011101001010010 01011
0101010101011010101011100010101010 01
0010101010010101010100101010101 01000
```

or

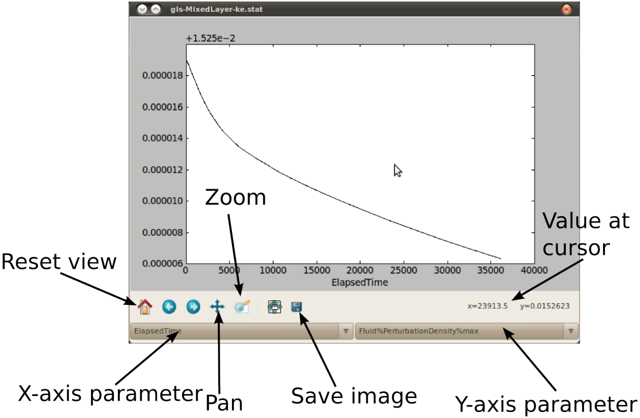
```
  0.48000000000000000E+003
 0.48000000000000000E+003  0.19027999999999999E
 +002 1683 6000 2480  0.18923088000000000E+008
 0.00000000000000000E+000  0.23051116000000000E
 +008  0.26832800000000000E+007
 0.00000000000000000E+000  0.26832800000000000E
 +007  0.14644880000000000E+007
 0.00000000000000000E+000  0.20527360000000000E
 +007  0.25328400000000000E+007
 0.00000000000000000E+000  0.36848400000000000E
 +007  0.21687840000000000E+007
 0.00000000000000000E+000  0.38967840000000000E
```

Statplot



gls-MixedLayer-ke.stat
+1.525e−2
0.000018
0.000016
0.000014
0.000012
0.000010
0.000008
0.000006
0
5000
10000
15000
20000
25000
30000
35000
40000
ElapsedTime
x=23913.5
y=0.0152623

Statplot



gls-MixedLayer-ke.stat

x=23913.5 y=0.0152623

ElapsedTime

Fluid%PerturbationDensity%max

Zoom

Value at cursor

Reset view

X-axis parameter

Pan

Save image

Y-axis parameter

Statplot keys

- s - scatter plot
- l - line plot
- r - refresh data
- R - refersh data, but keep current bounds
- x - switch x-axis from linear to log or vice versa
- y - switch y-axis from linear to log or vice versa
- q - quit (note: **no warnings!**)

Statplot example

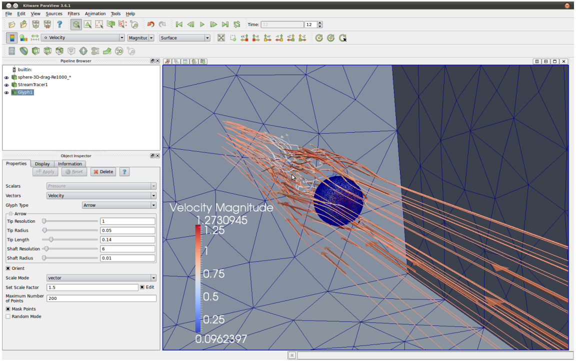
Open the stat file at from your advection problem
Things to try:

- Switch between scatter plot and line plot views
- Change the graph to show the number of elements through the run
- Plot velocity magnitude minimum against velocity magnitude maximum
- Zoom in and save a small part of the plot to file

Imperial College
London

Paraview

Open-source scientific visualisation software from KitView



Scalars: Pressure
Vectors: Velocity
Glyph Type: Arrow

Arrow
- Tip Resolution: 1
- Tip Radius: 0.05
- Tip Length: 0.14
- Shaft Resolution: 6
- Shaft Radius: 0.01

☒ Orient

Scale Mode: vector

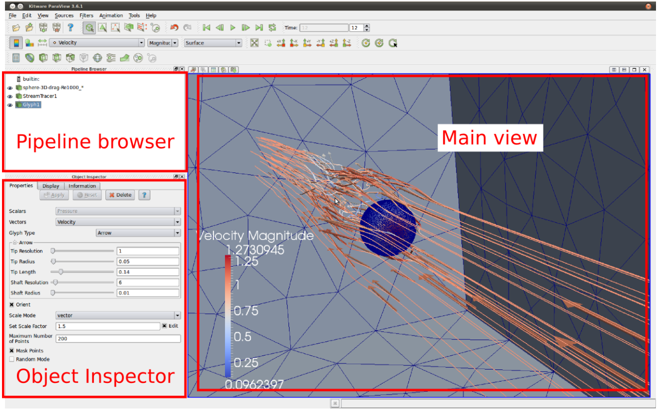
Set Scale Factor: 1.5 ☒ Edit

Maximum Number of Points: 200

☒ Mask Points

□ Random Mode

Paraview: main window

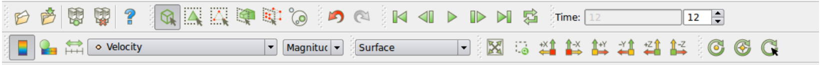


builtin:
sphere-3D-drag-Re1000_*
StreamTracer1
Glyph1

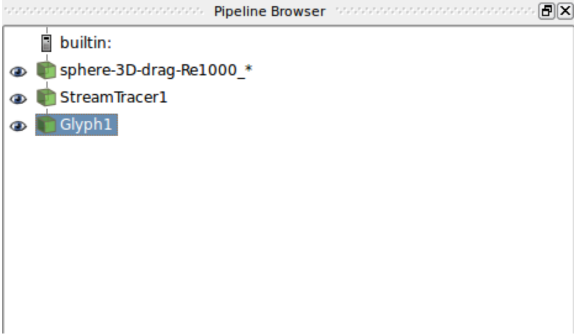
Scalars: Pressure
Vectors: Velocity
Glyph Type: Arrow
Arrow
Tip Resolution: 1
Tip Radius: 0.05
Tip Length: 0.14
Shaft Resolution: 6
Shaft Radius: 0.01
☒ Orient
Scale Mode: vector
Set Scale Factor: 1.5 ☒ Edit
Maximum Number of Points: 200
☒ Mask Points
☐ Random Mode

Imperial College
London

Paraview: main window



Paraview: main window

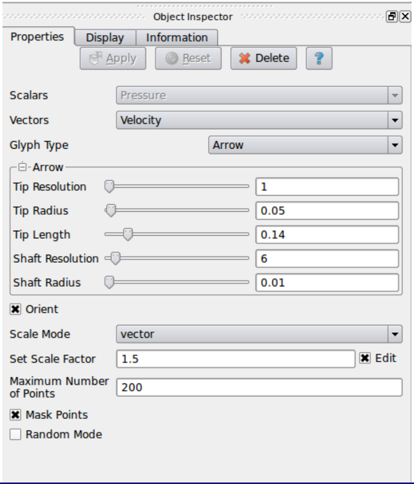


Pipeline Browser

- builtin:
- sphere-3D-drag-Re1000_*
- StreamTracer1
- Glyph1

Imperial College
London

Paraview: main window



Scalars: Pressure

Vectors: Velocity

Glyph Type: Arrow

Arrow

Tip Resolution	1
Tip Radius	0.05
Tip Length	0.14
Shaft Resolution	6
Shaft Radius	0.01

☒ Orient

Scale Mode: vector

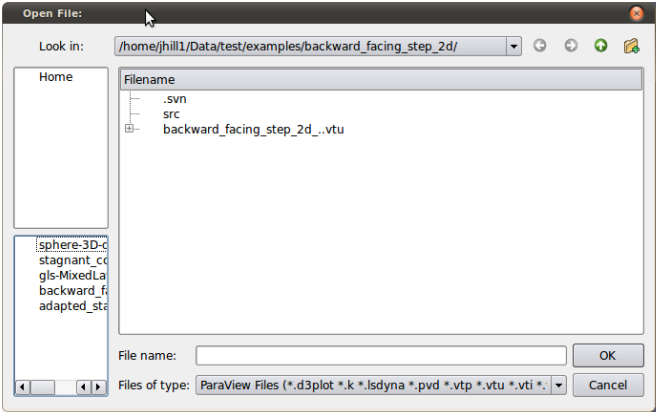
Set Scale Factor: 1.5 ☒ Edit

Maximum Number of Points: 200

☒ Mask Points

□ Random Mode

Paraview



Look in: /home/jhill1/Data/test/examples/backward_facing_step_2d/

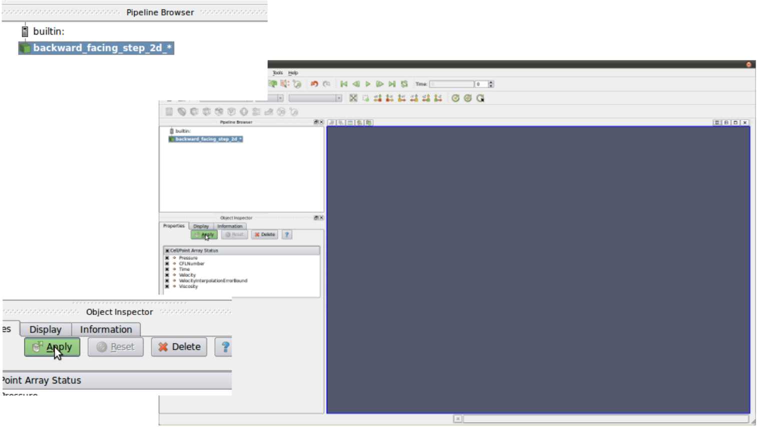
.svn
src
backward_facing_step_2d_..vtu

sphere-3D-c
stagnant_cc
gls-MixedLa
backward_fa
adapted_sta

File name:

Files of type: ParaView Files (*.d3plot *.k *.lsdyna *.pvd *.vtp *.vtu *.vti *.

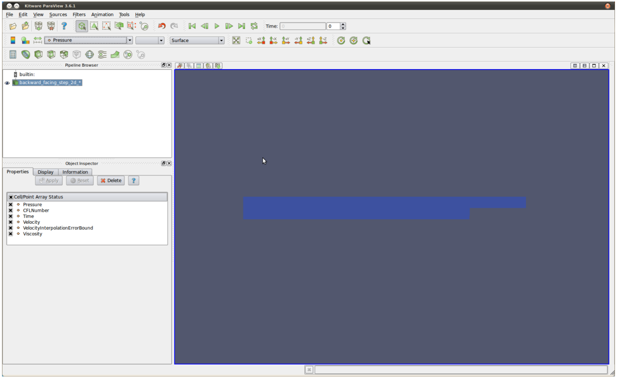
Paraview



Cell/Point Array Status

- ☒ Pressure
- ☒ CFLNumber
- ☒ Time
- ☒ Velocity
- ☒ VelocityInterpolationErrorBound
- ☒ Viscosity

Paraview



Kitware ParaView 3.6.1

Time: 0

○ Pressure

Surface

builtin:

backward_facing_step_2d_*

☒ Cell/Point Array Status

- ☒ ○ Pressure
- ☒ ○ CFLNumber
- ☒ ○ Time
- ☒ ○ Velocity
- ☒ ○ VelocityInterpolationErrorBound
- ☒ ○ Viscosity

Paraview

- Right click: Zoom-in and out
- Left-click: rotate
- Middle-button: move
- Zoom in and save a small part of the plot to file

Animations

1. File → Save Animation
2. Set up parameters
3. Click “Save Animation”
4. create folder and give filename

Animations

From PNGs produce movie via mencoder:

```
export opt=
"vbitrate=4705000:mbd=2:keyint=132:vqblur=1.0:cmp=2:subcmp=2:dia=2
mencoder -ovc lavc -lavcopts vcodec=msmpeg4v2:vpass=1:$opt
  -mf type=png:fps=10 -nosound -o /dev/null mf://*.png
mencoder -ovc lavc -lavcopts vcodec=msmpeg4v2:vpass=2:$opt
  -mf type=png:fps=10 -nosound -o output.avi mf://*.png
```

Script in fluidity/bin/encode

Python tools

- vtktools - read vtu files
- statparser - read stat files

Useful python modules

- numpy - numerical package, including arrays
- stats - linear regression, etc
- matplotlib - plotting 2- and 3-D

Python VTU

```
#!/usr/bin/env python
import vtktools
x0 = 0
y0 = 0

for file in filelist:
  num = int(file.split(".vtu")[0].split(’_’)[-1])
  u=vtktools.vtu(file)
  time = u.GetScalarField(’Time’)
  tt = time[0]
  den = u.GetScalarField(’Density’)
  p = u.GetLocations()
  xyz_data = []
  for i in range(0,len(den)):
    if (x0-0.1<p[i,0]<x0+0.1 and y0-0.1<p[i,1]<y0+0.1):
      xyz_data.append((p[i,0],p[i,1],-p[i,2],1024*den[i]))
```

Examples

```
#!/usr/bin/env python
from fluidity_tools import stat_parser

# load in statfile to get element info
stat=stat_parser( direc + '/' + stat_file )

elements = stat['CoordinateMesh']['elements']
nodes = stat['CoordinateMesh']['nodes']

maxVelocity = stat["Fluid"]["Velocity%magnitude"]["max"]
```

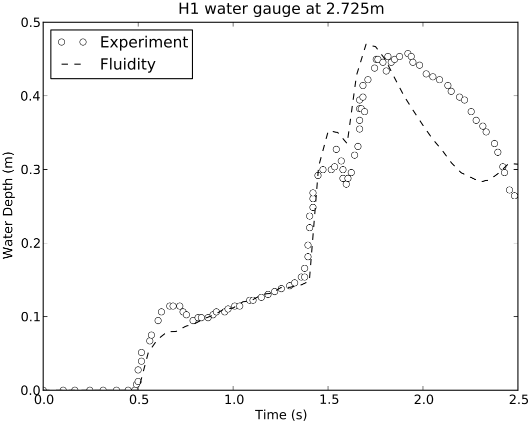
Examples

```
#!/usr/bin/env python
from pylab import *

figure(x)
title(warray[x]+"␣water␣gauge␣at␣"+str(xarray[x])+"m")
xlabel('Time␣(s)')
ylabel('Water␣Depth␣(m)')
experiment = numpy.load(warray[x]+".npy")
plot(experiment[:,0],
  experiment[:,1],marker='o',markerfacecolor='white',markersize=6,
  markeredgecolor='black',linestyle="None")

time = results[:,1]
plot(time, results[:,2+x],color='black',linestyle="dashed")
axis([0.0, 2.5, 0.0, 0.5])
legend(("Experiment", "Fluidity"), loc="upper␣left")
savefig("water_gauge_"+warray[x]+".pdf")
```

Examples


H1 water gauge at 2.725m
Experiment
Fluidity
Water Depth (m)
0.0
0.1
0.2
0.3
0.4
0.5
Time (s)
0.0
0.5
1.0
1.5
2.0
2.5

Outline

- Running
- Output
  - Filetypes and tools
  - The stat file
  - Paraview
  - Python
- Parallel
  - Running
  - Post-running

FLML

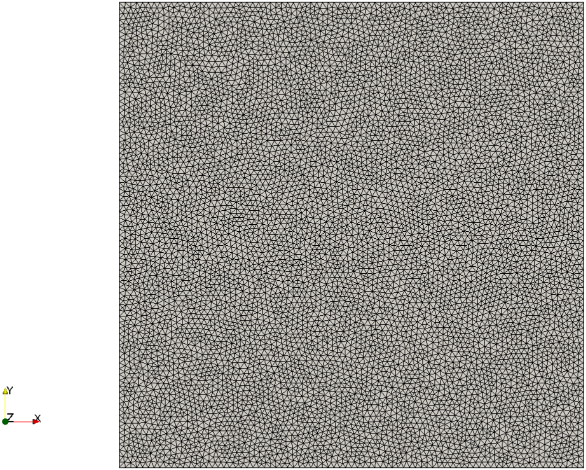
No changes required!

[Optional]

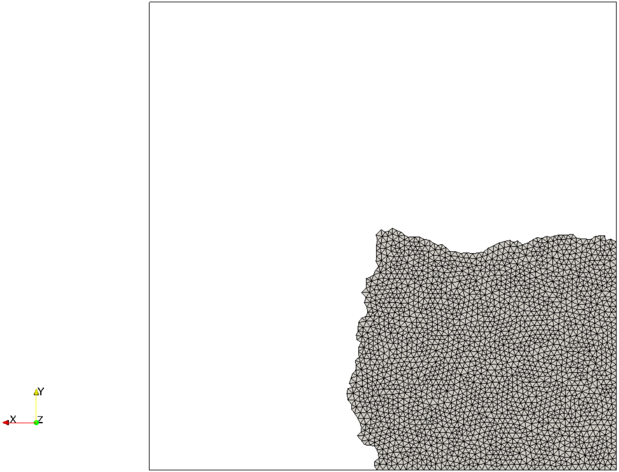
- Remove fields from stat file
- Remove some fields from VTU

Imperial College
London

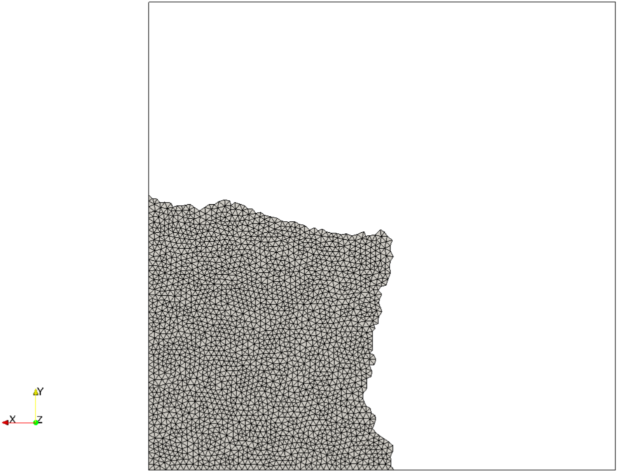
Decompose mesh



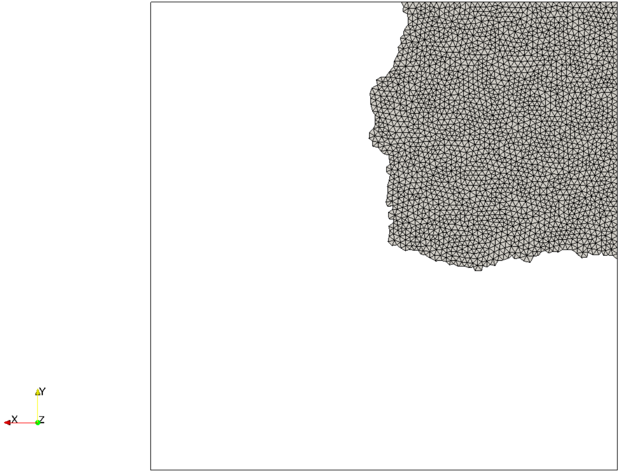
Decompose mesh



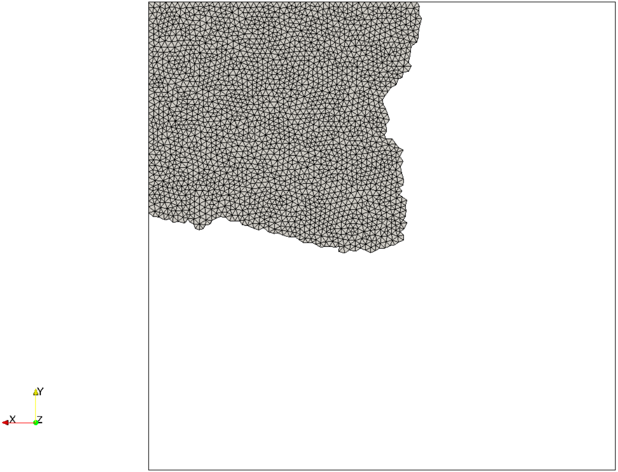
Decompose mesh



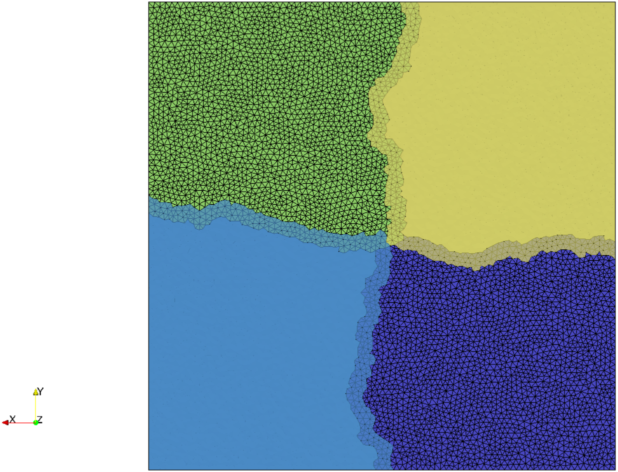
Decompose mesh



Decompose mesh



Decompose mesh



flredecomp

```
mpiexec -n 8 flredecomp -i 2 -o 8 InputFLML OutputFLML
```

Will decompose mesh from 2 to 8.

```
mpiexec -n 8 flredecomp -i 8 -o 2 InputFLML OutputFLML
```

Will decompose mesh from 8 to 2.
Both need running on 8 processors
Note: Do not add .flml to files, e.g. InputFLML not InputFLML.flml

Note: Change of FLML filename to run

Local systems

```
mpiexec -n 8 ../../bin/fluidity
my_flredecomped.flml
```

Visualisation

No different from serial - except .pvtu files, not .vtu

Log files (if used) will be one per processor.