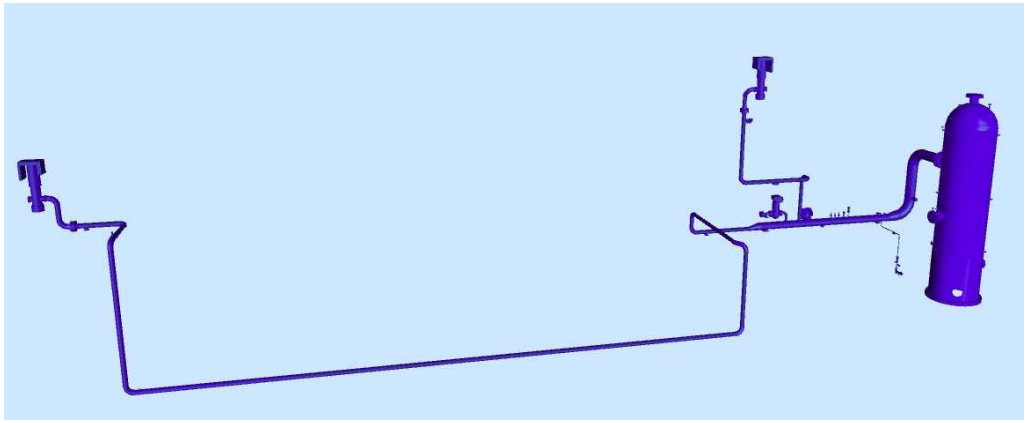
pipemesh: Preprocessing Pipe Network Generation for Use in Pipe Flow Analyses

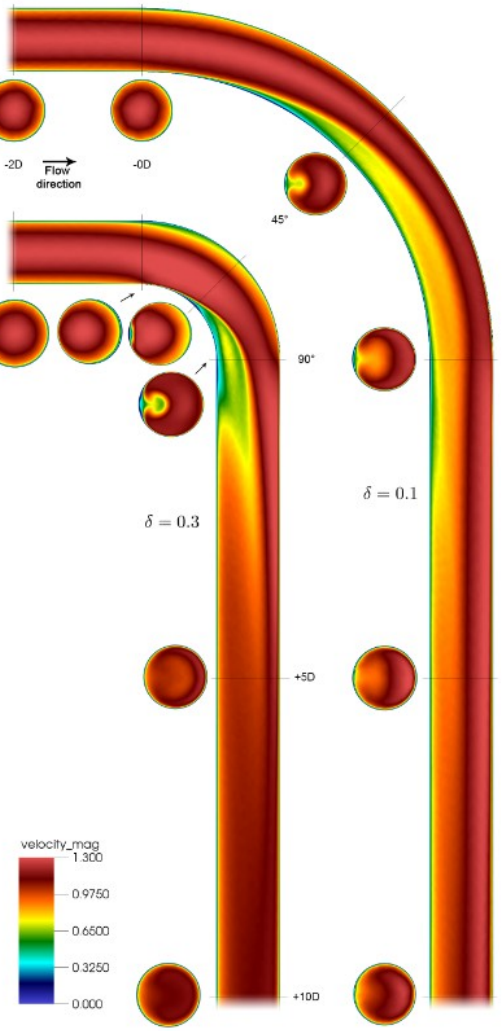
Duncan Hunter MSc ACSE
dh1515@ic.ac.uk
github.com/Duncan-Hunter/pipemesh

Motivation – Pipe flow

- Pipe flow analyses
  - Vary geometrical parameters
  - Industrial Designs



BP

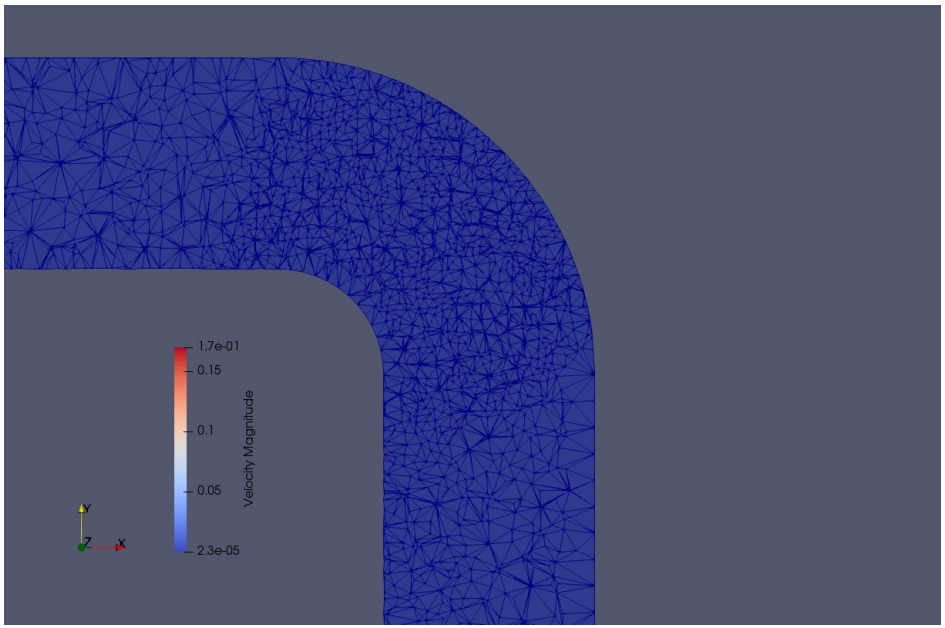


Hufnagel L (2016)

Duncan Hunter MSc ACSE
dh1515@ic.ac.uk github.com/Duncan-Hunter/pipemesh

IC-FERST

- Can simulate turbulent flow.
- Mesh Adaptivity (not often found in commercial software).



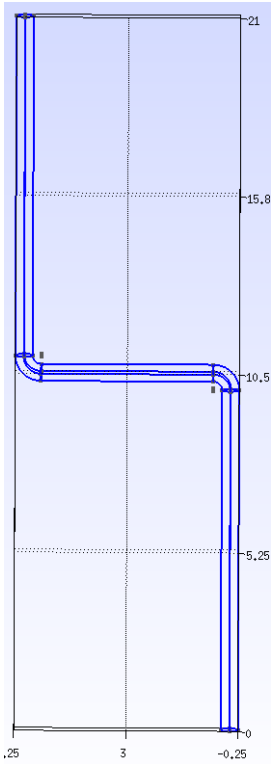
- Uses GMSH .msh files.
- Uses physical groups for boundary/initial conditions.

- http://multifluids.github.io/

Duncan Hunter MSc ACSE
dh1515@ic.ac.uk github.com/Duncan-Hunter/pipemesh

Motivation - GMSH

- Conventional method is barrier for use.

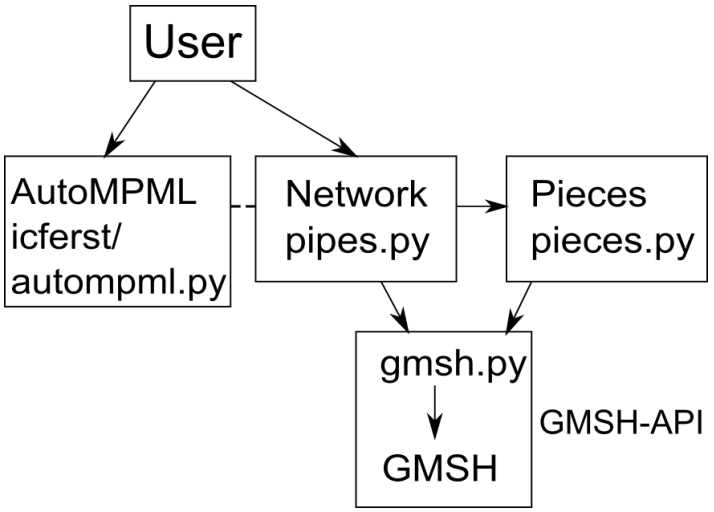



```
charLen = 0.2;
charLen2 = 0.1;
r=0.25;
ex1=10;
ex2=5;
ex3=10;
Point(1) = {0,0,0,charLen};
Point(2) = {r,0,0,charLen};
Point(3) = {0,r,0,charLen};
Point(4) = {-r,0,0,charLen};
Point(5) = {0,-r,0,charLen};
Circle(1) = {2,1,3};
Circle(2) = {3,1,4};
Circle(3) = {4,1,5};
Circle(4) = {5,1,2};
Line Loop(5) = {1,2,3,4};
Plane Surface(6) = {5};
extr1[] = Extrude {0,0,ex1} {
   Surface{6};
};
extr2[] = Extrude { {1,0,0}, // direction of rotation axis
                    {0,2*r,ex1}, // a point on the rotation axis
                    -Pi/2 } { // the rotation angle
   Surface{extr1[0]};
};
extr3[] = Extrude {0,ex2,0} {
  Surface{extr2[0]};
};
extr4[] = Extrude { {1,0,0}, // direction of rotation axis
                    {0,ex2+2*r,ex1+4*r}, // a point on the rotation axis
                    Pi/2 } { // the rotation angle
   Surface{extr3[0]};
};
Extrude {0,0,ex3} {
  Surface{extr4[0]};
}
Physical Surface(1) = {116};
Physical Surface(2) = {6};
Physical Surface(3) = {19, 15, 23, 27, 49, 45, 41, 37, 59, 63, 71, 67, 81, 85,
                       93, 89, 115, 111, 103, 107};
Physical Volume(4) = {5, 4, 3, 2, 1}; //4 and 2 are curves
Characteristic Length {13, 8, 6, 19, 21, 26, 57, 69, 64, 70, 82, 86} = charLen2;
```

Duncan Hunter MSc ACSE
dh1515@ic.ac.uk github.com/Duncan-Hunter/pipemesh

Software

- Object oriented.
  - Individual fittings, and entire pipe network represented as Classes.



- Used with executable scripts.
- Can be used on HPC:

  No GUI

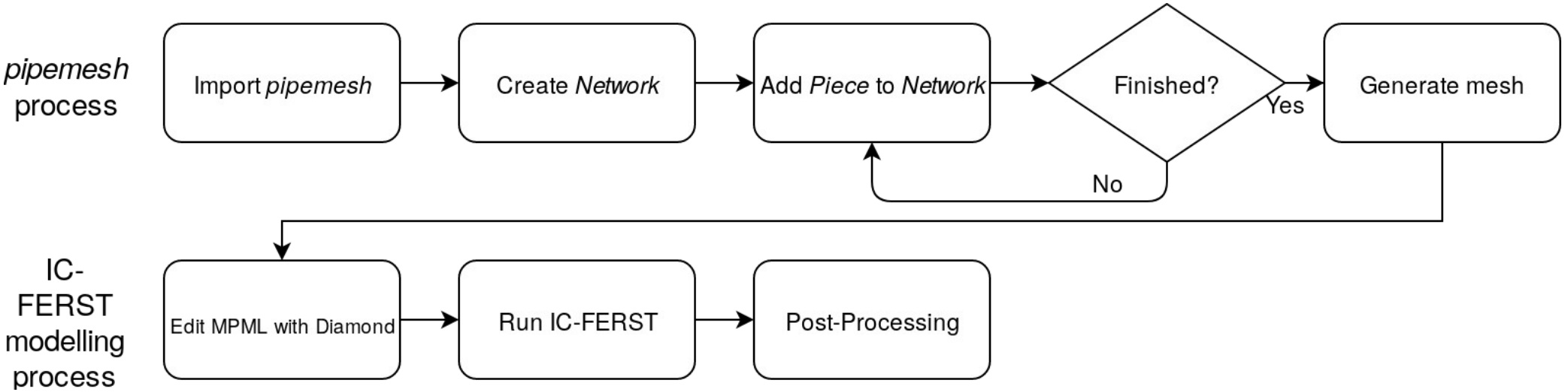
  No file transfer.
- Easy to use: User does not need to learn GMSH
- GMSH-API

  Python, C++, Julia files instead of .geo
- OpenCASCADE

  Volumes
  Boolean Operations

Duncan Hunter MSc ACSE
dh1515@ic.ac.uk
github.com/Duncan-Hunter/pipemesh

User process

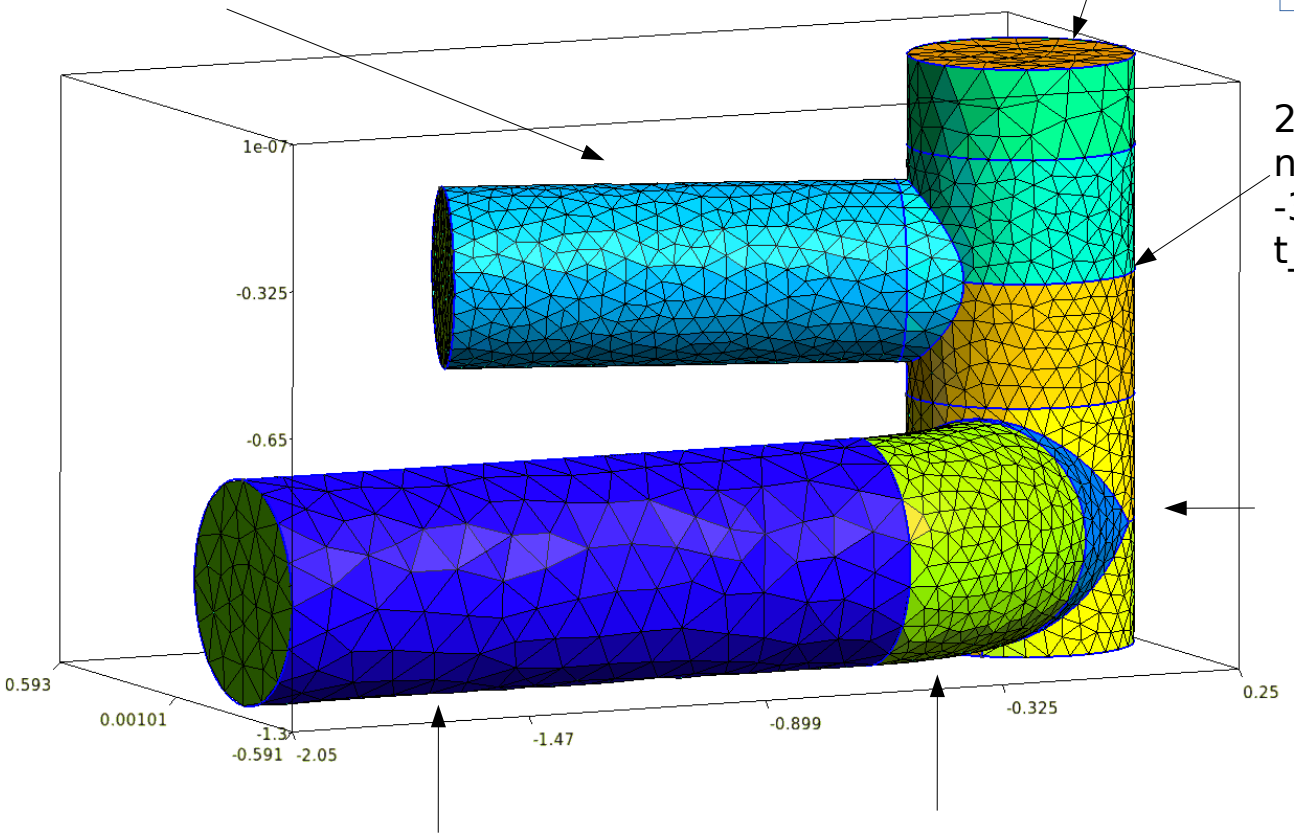


Duncan Hunter MSc ACSE
dh1515@ic.ac.uk
github.com/Duncan-Hunter/pipemesh

Network

1) network = Network(0.2, 0.25, [0, 0 , -1], 0.1)

network.add_cylinder(1, 0.05, out_number=2)



2)
network.add_t_junction([
-3,1,0], 0.05,
t_radius=0.2)

3)
network.add_t_junction([-1,-1,0], 0.05)

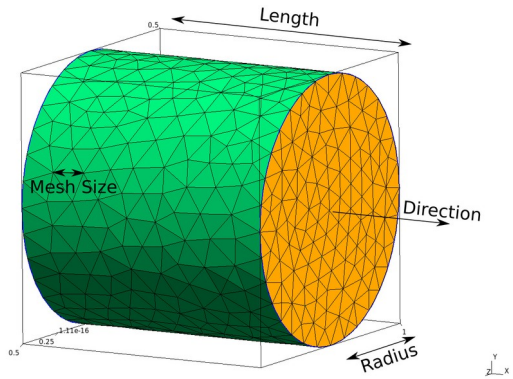
network.add_cylinder(1.5, 0.1, out_number=3)

network.add_curve([-1,0,0], 0.5, 0.05, out_number=3)

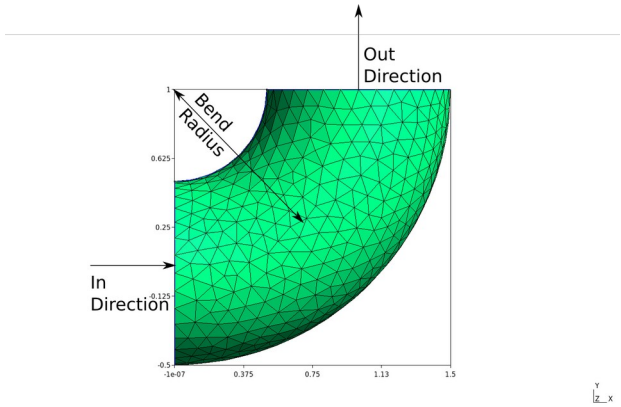
Duncan Hunter MSc ACSE
dh1515@ic.ac.uk github.com/Duncan-Hunter/pipemesh

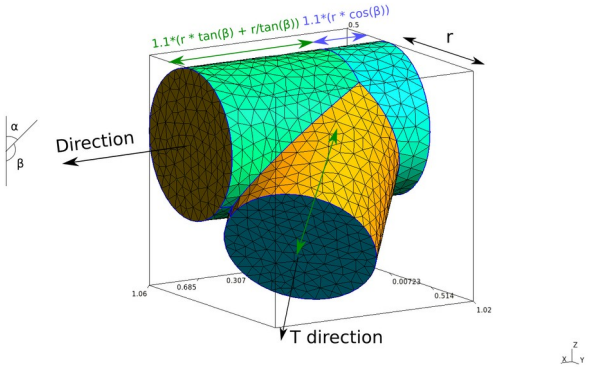
PipePiece

- Individual PipePiece fittings.
- Init -> Rotation -> Translation.

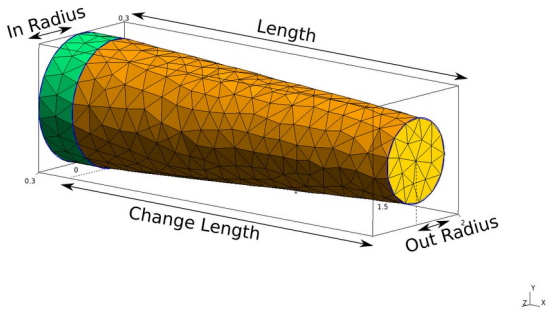


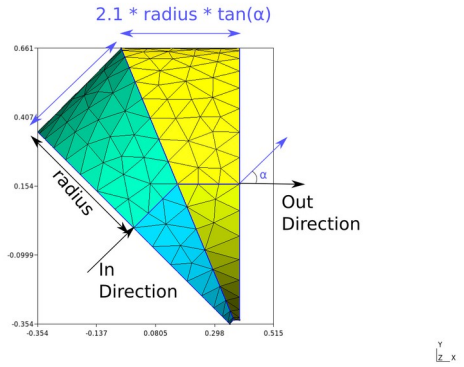
Arbitrary bend radii and directions.





Any joining direction, smaller joining radius

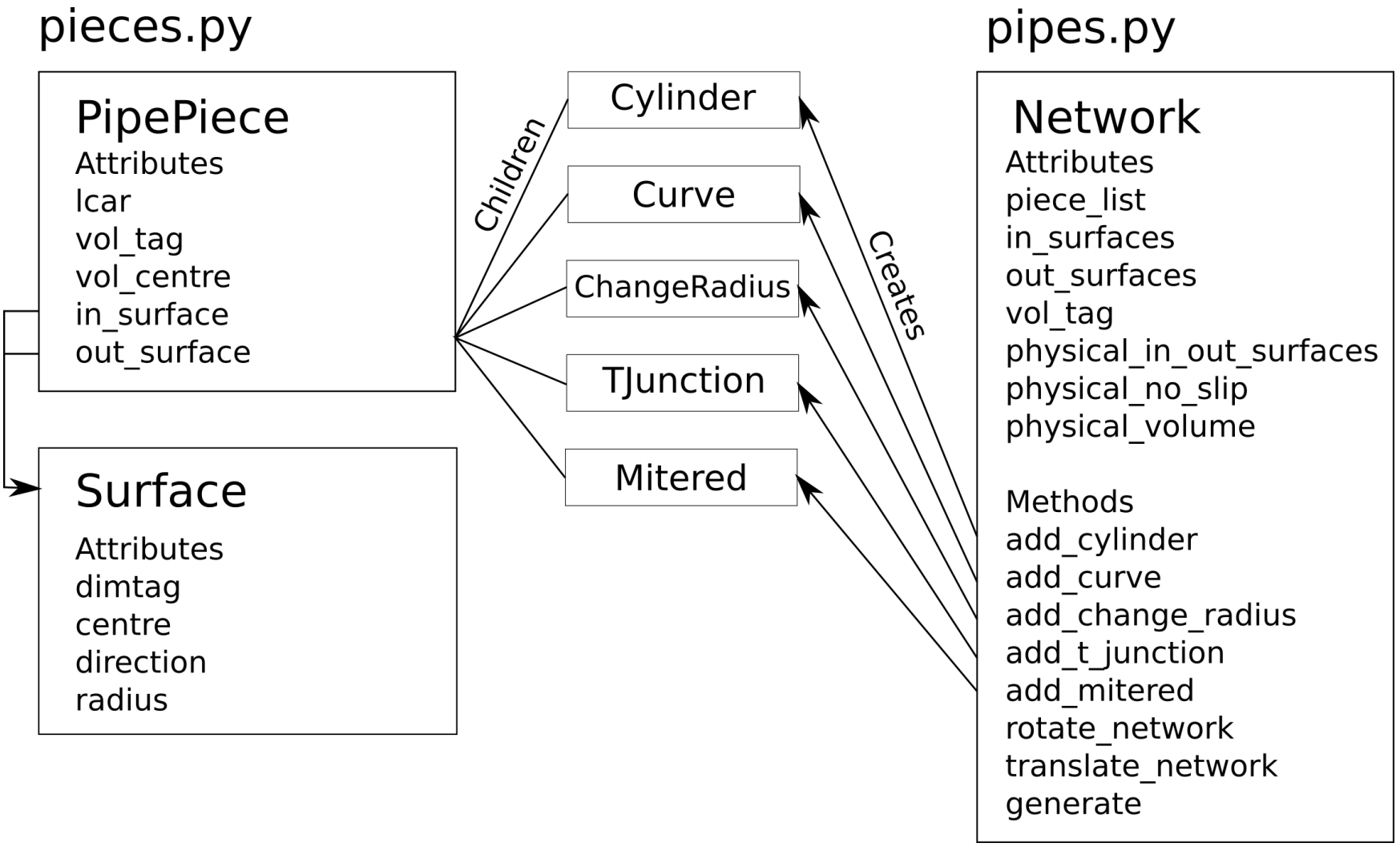




Arbitrary directions

Duncan Hunter MSc ACSE
dh1515@ic.ac.uk github.com/Duncan-Hunter/pipemesh

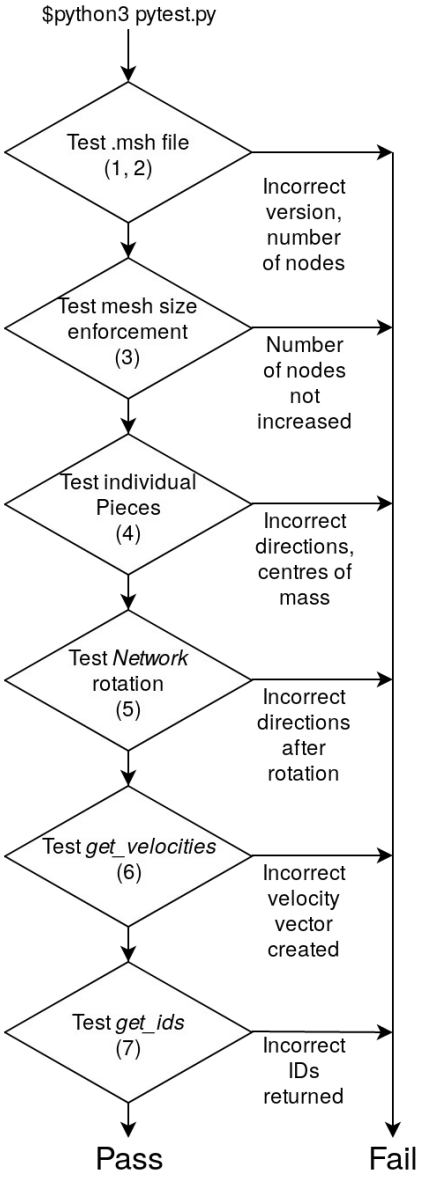
Network and PipePiece classes



Duncan Hunter MSc ACSE
dh1515@ic.ac.uk github.com/Duncan-Hunter/pipemesh

Testing

- Execute during development.
- Ensures new version of GMSH-API works.
- Ensures new functionalities don’t break code.
- IC-FERST test cases.



Duncan Hunter MSc ACSE
dh1515@ic.ac.uk github.com/Duncan-Hunter/pipemesh

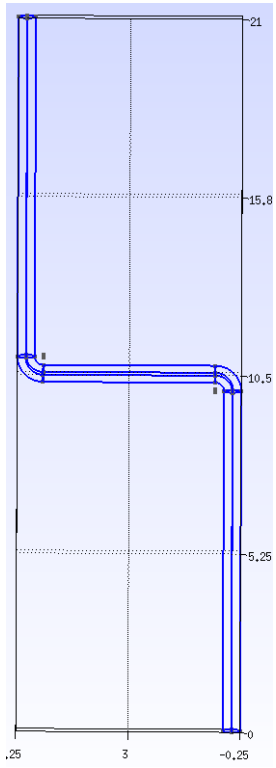
Results

- Commands saved per fitting added.
- Does not account for geometrical information and calculations.

	Cylinder	Curve	Mitered	ChangeRadius	TJunction
Creation	1	2	7	2	5
Rotate Inlet	1	1	1	1	1
Rotate Outlet	0	1	1	0	1
Translate	1	1	1	1	1
Total Commands	3	5	10	4	8

Duncan Hunter MSc ACSE
dh1515@ic.ac.uk github.com/Duncan-Hunter/pipemesh

Results




```
charLen = 0.2;
charLen2 = 0.1;
r=0.25;
ex1=10;
ex2=5;
ex3=10;
Point(1) = {0,0,0,charLen};
Point(2) = {r,0,0,charLen};
Point(3) = {0,r,0,charLen};
Point(4) = {-r,0,0,charLen};
Point(5) = {0,-r,0,charLen};
Circle(1) = {2,1,3};
Circle(2) = {3,1,4};
Circle(3) = {4,1,5};
Circle(4) = {5,1,2};
Line Loop(5) = {1,2,3,4};
Plane Surface(6) = {5};
extr1[] = Extrude {0,0,ex1} {
   Surface{6};
};
extr2[] = Extrude { {1,0,0}, // direction of rotation axis
                    {0,2*r,ex1}, // a point on the rotation axis
                    -Pi/2 } { // the rotation angle
   Surface{extr1[0]};
};
extr3[] = Extrude {0,ex2,0} {
  Surface{extr2[0]};
};
extr4[] = Extrude { {1,0,0}, // direction of rotation axis
                    {0,ex2+2*r,ex1+4*r}, // a point on the rotation axis
                    Pi/2 } { // the rotation angle
   Surface{extr3[0]};
};
Extrude {0,0,ex3} {
  Surface{extr4[0]};
}
Physical Surface(1) = {116};
Physical Surface(2) = {6};
Physical Surface(3) = {19, 15, 23, 27, 49, 45, 41, 37, 59, 63, 71, 67, 81, 85,
                       93, 89, 115, 111, 103, 107};
Physical Volume(4) = {5, 4, 3, 2, 1}; //4 and 2 are curves
Characteristic Length {13, 8, 6, 19, 21, 26, 57, 69, 64, 70, 82, 86} = charLen2;
```

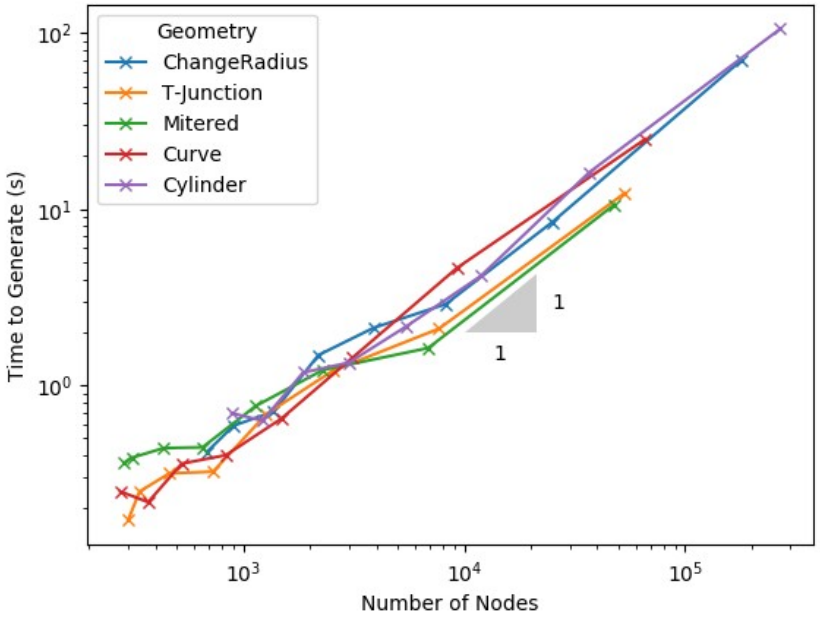
```
1  from pipemesh import pipes
2
3  network = pipes.Network(10, 0.25, [0, 0, -1], 0.1)
4  network.add_curve([0, -1, 0], 0.5, 0.1)
5  network.add_cylinder(5, 0.1)
6  network.add_curve([0, 0, -1], 0.5)
7  network.add_cylinder(10, 0.1)
8
9  network.generate(run_gui=True)
```

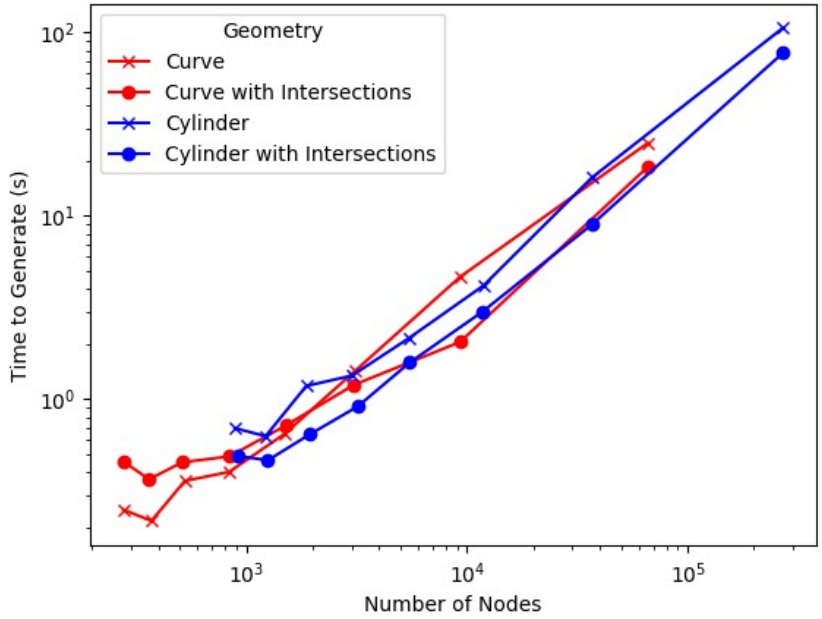
Duncan Hunter MSc ACSE
dh1515@ic.ac.uk github.com/Duncan-Hunter/pipemesh

Results

Dependent on number of nodes (especially at large no. of nodes).

Fusing multiple volumes together has no significant impact.

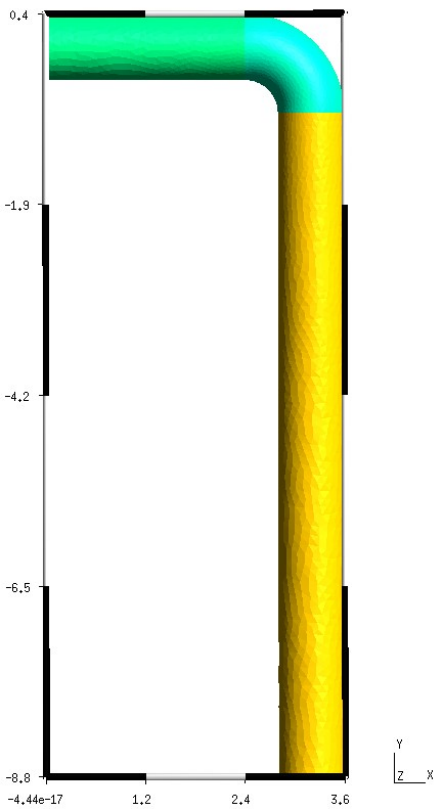


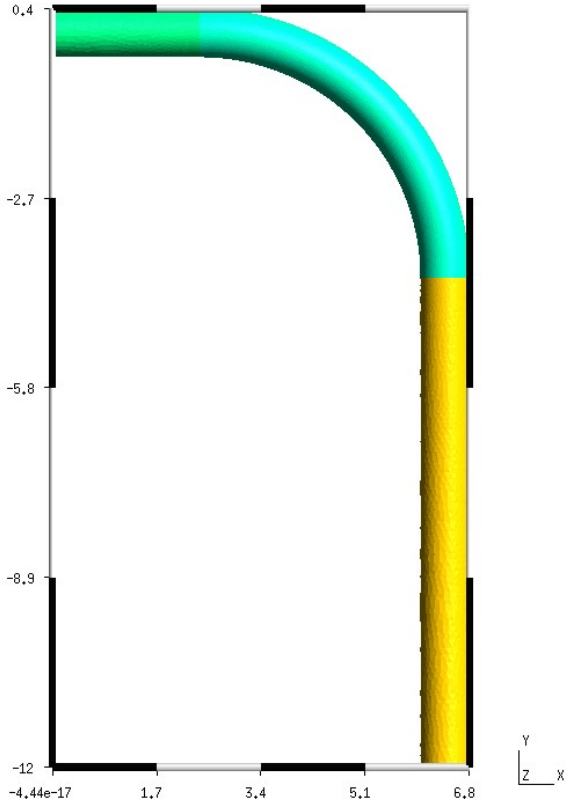


Duncan Hunter MSc ACSE
dh1515@ic.ac.uk
github.com/Duncan-Hunter/pipemesh

Results

Bend Ratio: 0.5 0.1

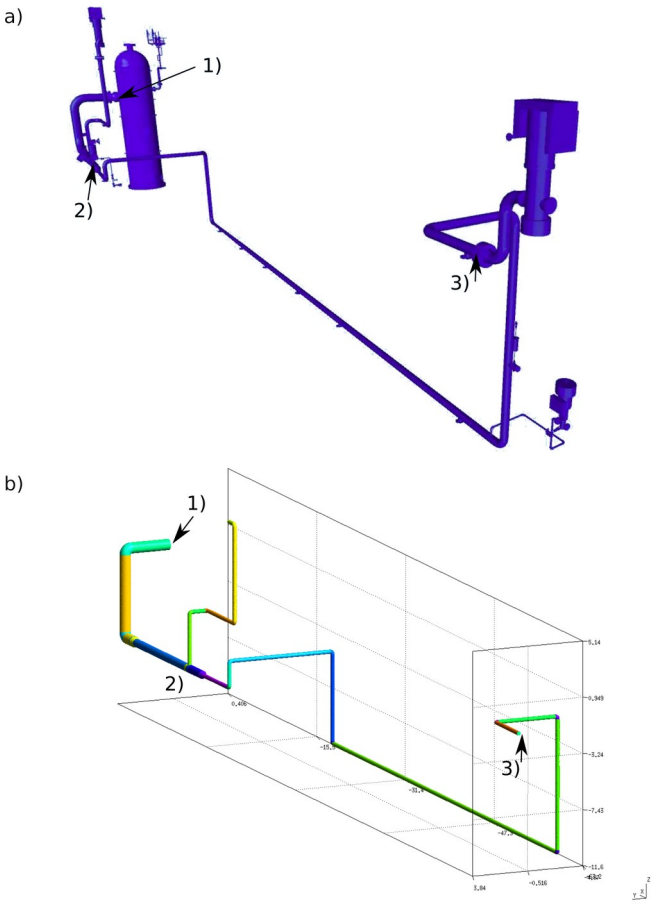





```python
from pipemesh import pipes
import os, sys

pipe_radius = 0.4
diameter = 2*pipe_radius

bend_ratios = [0.1, 0.2, 0.3, 0.4, 0.5]
bend_radii = [pipe_radius/i for i in bend_ratios]

for bend_radius in bend_radii:
    network = pipes.Network(3*diameter, pipe_radius, [1, 0, 0], 0.1)
    network.add_curve([0, -1, 0], bend_radius, 0.05)
    network.add_cylinder(10*diameter, 0.1)
    os.mkdir("radius_{}".format(bend_radius))
    os.mkdir("radius_{}/src".format(bend_radius))
    network.generate(filename="radius_{}/src/pipe".format(bend_radius),
                     write_info=True,
                     binary=True,
                     run_gui=False)
```

Duncan Hunter MSc ACSE
dh1515@ic.ac.uk
github.com/Duncan-Hunter/pipemesh

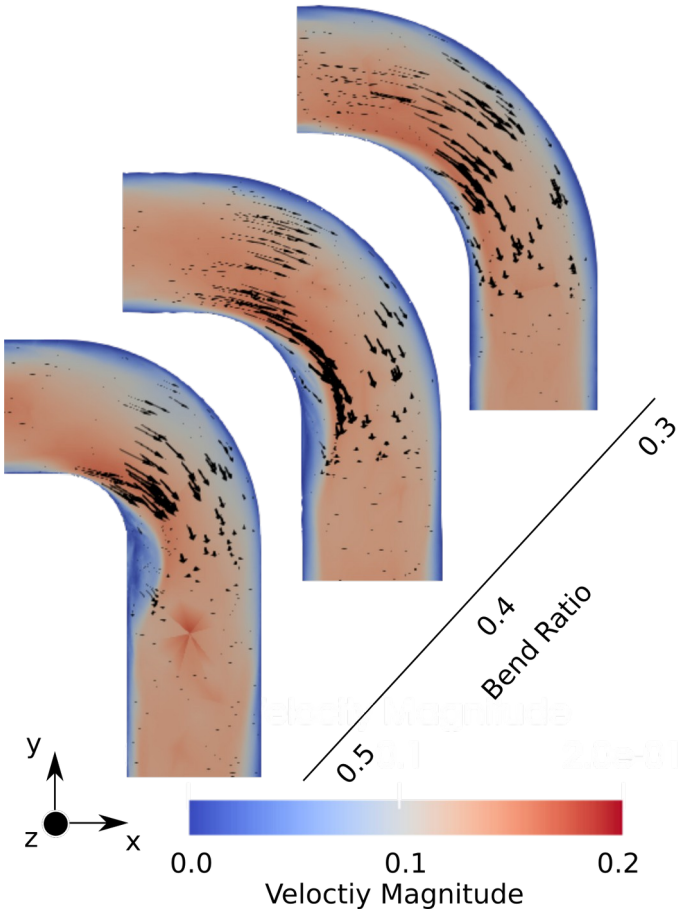
Results

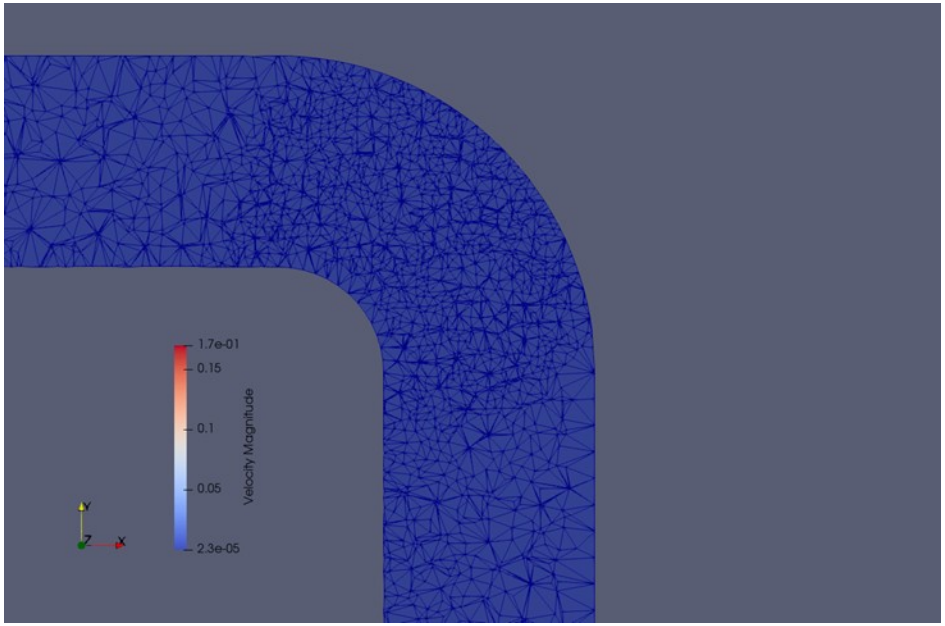



```
1  from pipemesh import pipes
2
3  network = pipes.Network(2.981, 0.4064, [0, 1, 0], 0.1)
4  network.add_curve([0, 0, -1], 0.45, 0.1)
5  network.add_cylinder(5.722, 0.1)
6  network.add_curve([-1, 0, 0], 0.45, 0.1)
7  network.add_cylinder(1.119, 0.1)
8  network.add_change_radius(0.5, 0.3048, 0.4, 0.1)
9  network.add_cylinder(9.12, 0.1)
10 network.add_t_junction([0, 0, 1], 0.1, t_radius=0.1524)
11  # Inlet A
12 network.add_cylinder(3.673, 0.08, out_number=2)
13 network.add_curve([0, -1, 0], 0.2, 0.08, out_number=2)
14 network.add_cylinder(1, 0.08, out_number=2)
15 network.add_curve([-1, 0, 0], 0.2, 0.08, out_number=2)
16 network.add_cylinder(4.5, 0.08, out_number=2)
17 network.add_curve([0, 0, 1], 0.2, 0.08, out_number=2)
18 network.add_cylinder(7, 0.08, out_number=2)
19 network.add_curve([1, 0, 0], 0.2, 0.08, out_number=2)
20 network.add_cylinder(0.8, 0.08, out_number=2)
21 # Inlet B
22 network.add_cylinder(1.9, 0.1)
23 network.add_change_radius(0.5, 0.1524, 0.4, 0.08)
24 network.add_cylinder(4.191, 0.08)
25 network.add_curve([0, 0, 1], 0.2, 0.08)
26 network.add_cylinder(1.6, 0.08)
27 network.add_curve([0, -1, 0], 0.2, 0.08)
28 network.add_cylinder(7.748, 0.08)
29 network.add_curve([0, 0, -1], 0.2, 0.08)
30 network.add_cylinder(6.446, 0.08)
31 network.add_curve([-1, 0, 0], 0.2, 0.08)
32 network.add_cylinder(39.916, 0.08)
33 network.add_curve([0, 0, 1], 0.2, 0.08)
34 network.add_cylinder(9.762, 0.08)
35 network.add_curve([0, 1, 0], 0.2, 0.08)
36 network.add_cylinder(4.498, 0.08)
37 network.add_curve([-1, 0, 0], 0.2, 0.08)
38 network.add_cylinder(3.984, 0.08)
39
40 network.generate(filename="pipe", binary=True, write_info=True)
```

Duncan Hunter MSc ACSE
dh1515@ic.ac.uk
github.com/Duncan-Hunter/pipemesh

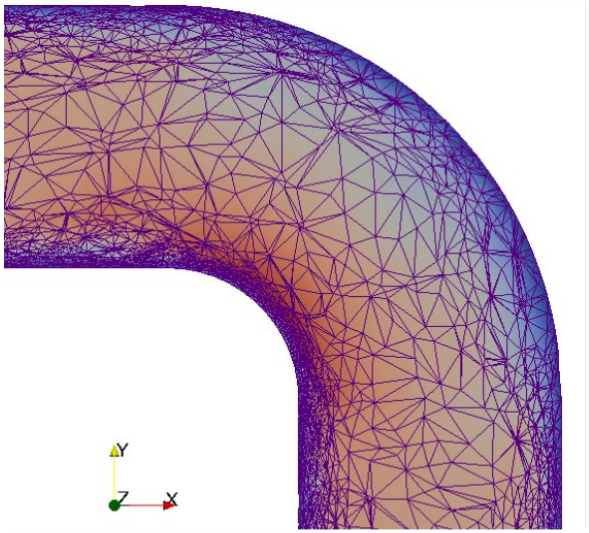
Results

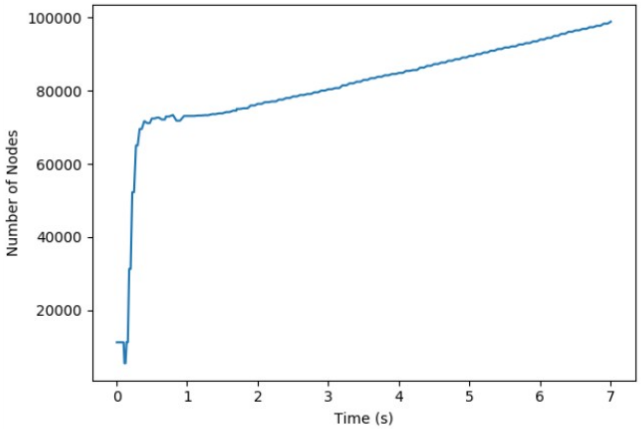


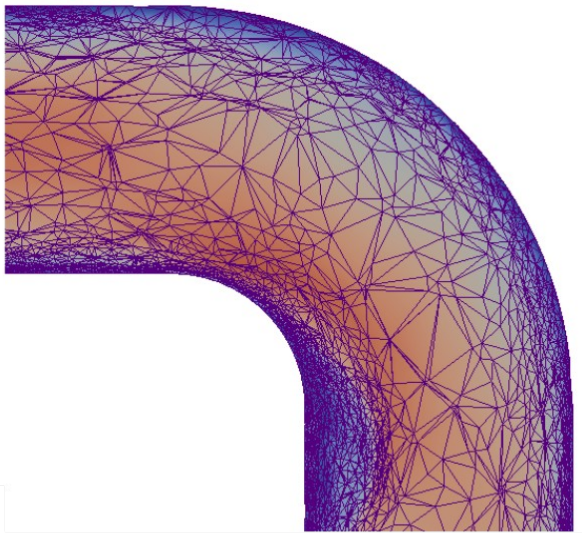


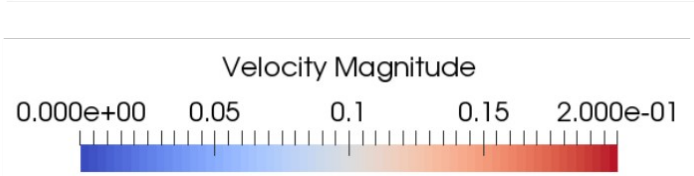
Duncan Hunter MSc ACSE
dh1515@ic.ac.uk
github.com/Duncan-Hunter/pipemesh

Results



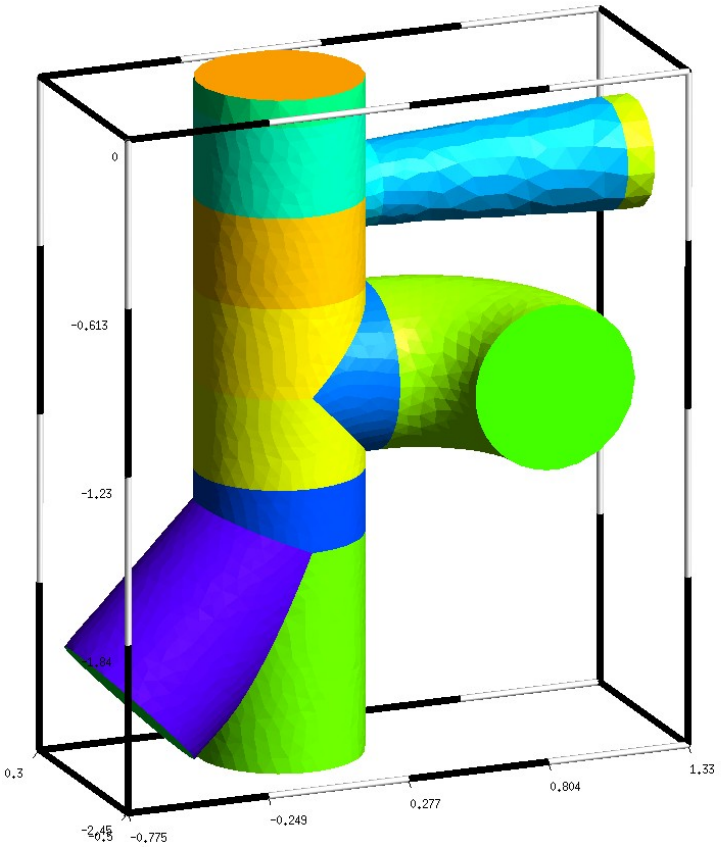


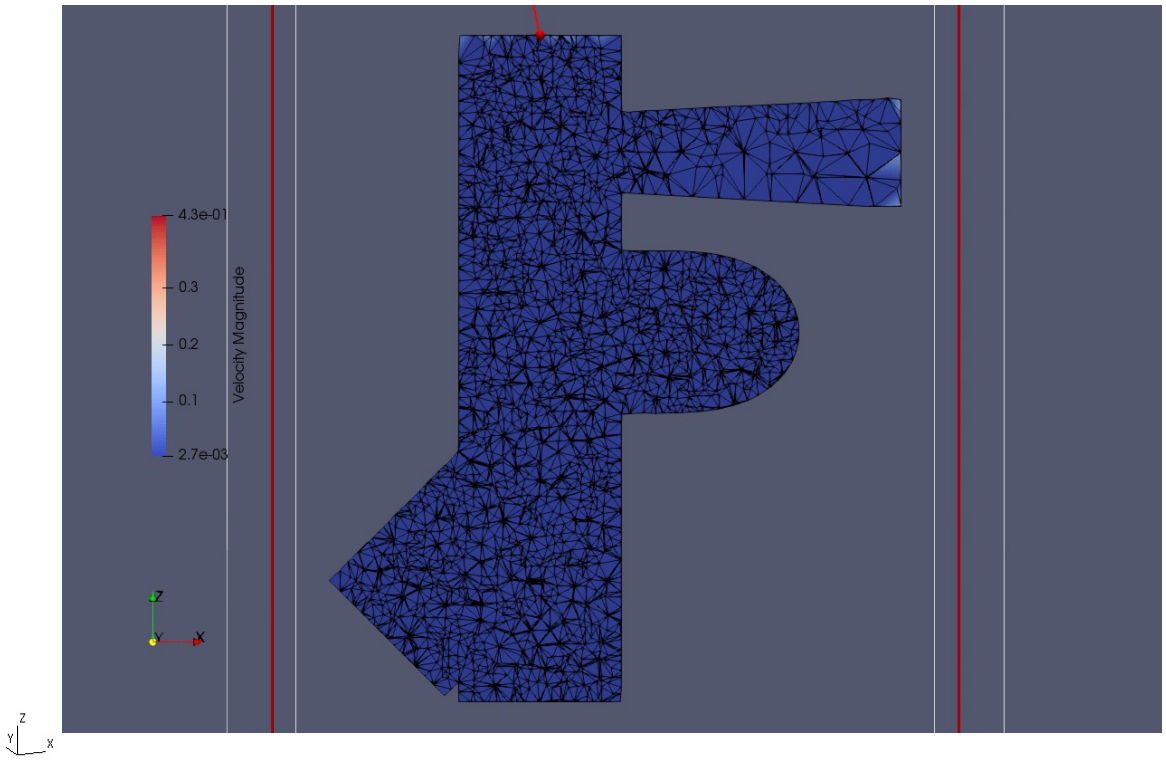




Duncan Hunter MSc ACSE
dh1515@ic.ac.uk
github.com/Duncan-Hunter/pipemesh

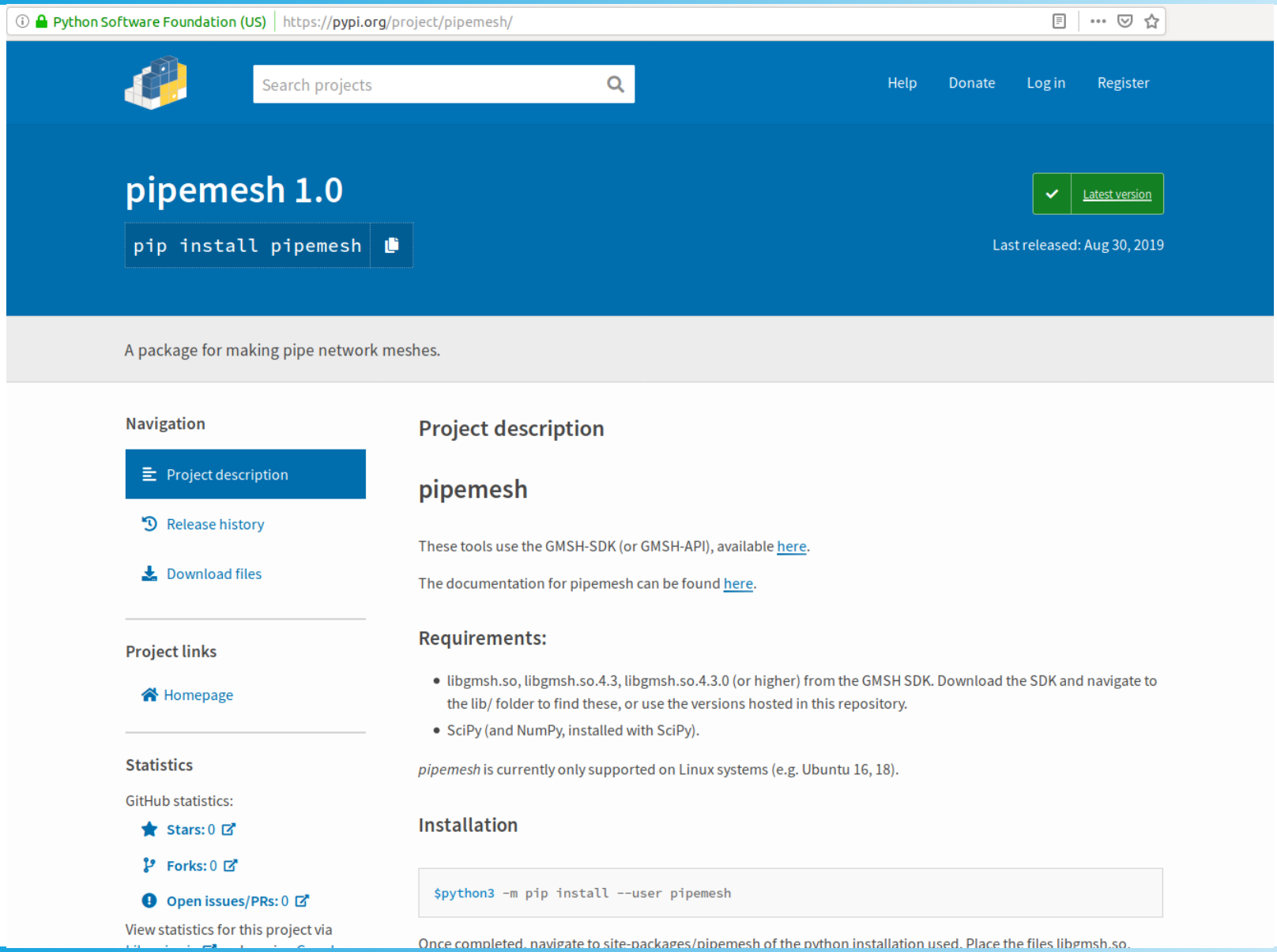
Results





Duncan Hunter MSc ACSE
dh1515@ic.ac.uk
github.com/Duncan-Hunter/pipemesh

Questions?



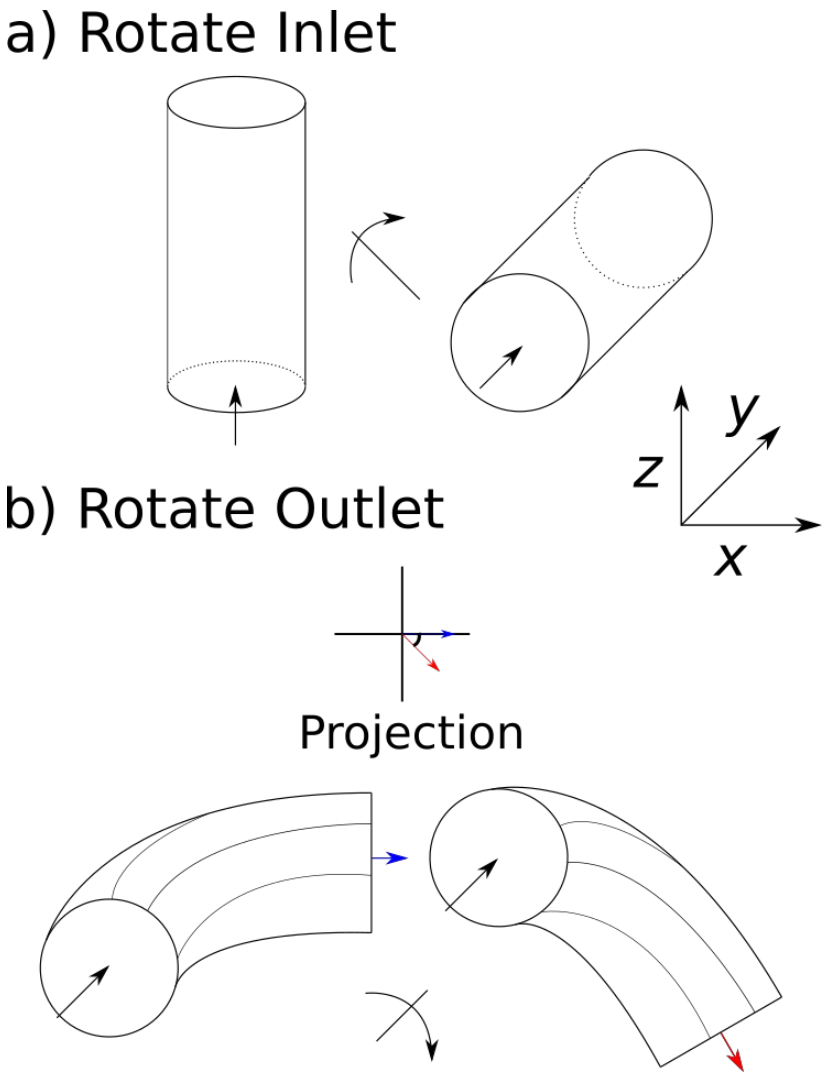
Duncan Hunter MSc ACSE
dh1515@ic.ac.uk
github.com/Duncan-Hunter/pipemesh

a) All Pieces.

Created up.
Rotated to user given direction.

b) Bends and Junctions.

Directions projected.
Angle calculated.
Rotated about inlet direction.



Duncan Hunter MSc ACSE
dh1515@ic.ac.uk github.com/Duncan-Hunter/pipemesh