Mesh Generation

Input data

Surface mesh

If you have a surface mesh file with a very “bad” triangulation, do not worry too much because the surface is going to be remeshed. On the contrary, if the surface geometry is “bad” by itself (not smoothed, narrow bifurcations) this is not going to be fixed now. You have to fix it BEFORE starting generating the mesh.

Tool

VMTK

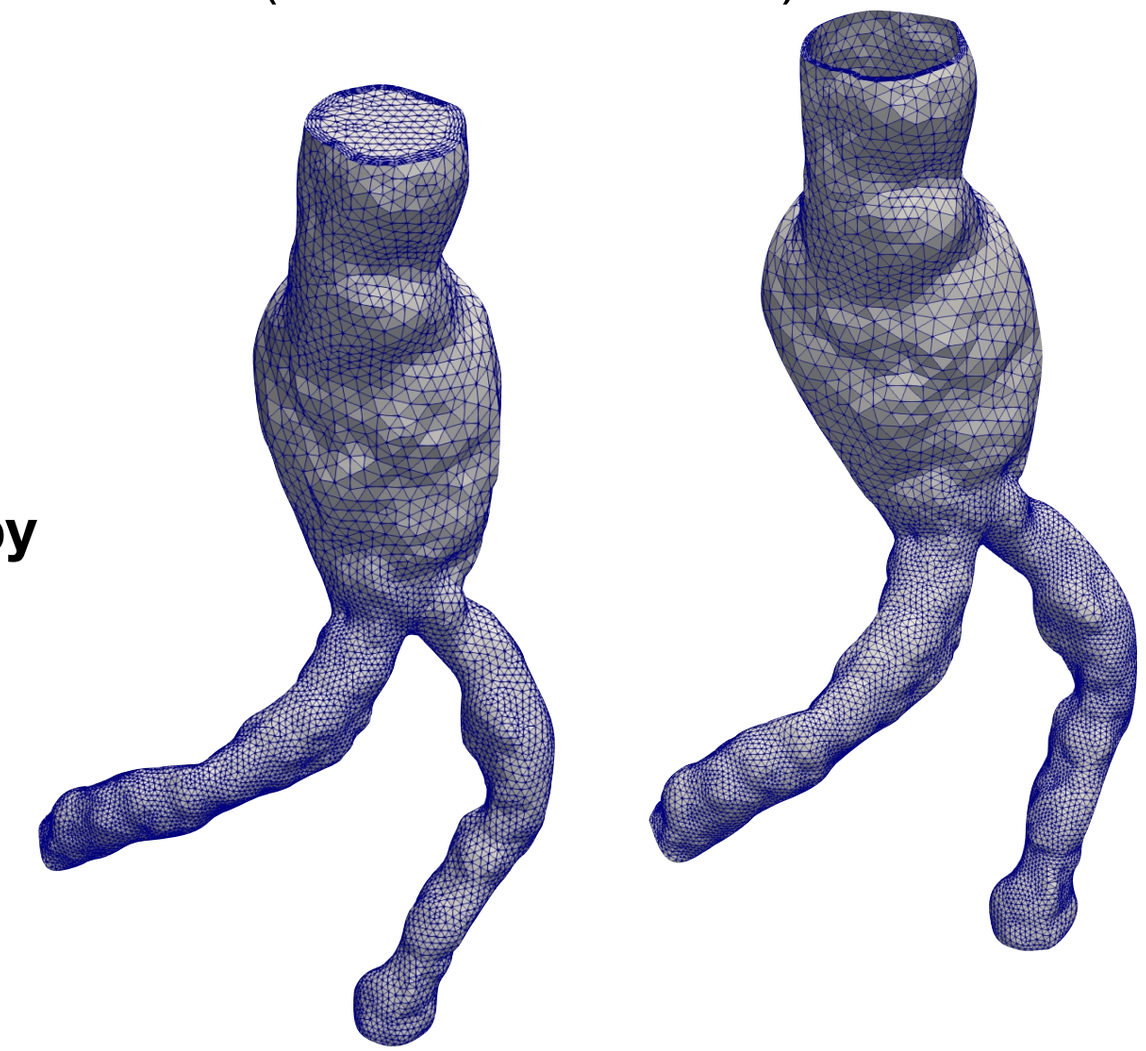
(www.vmtk.org)

fsimeshgenerator.py

Output data

Fluid and solid volume meshes

(.mesh LifeV format)



HOW TO CITE:

E. Faggiano, F. Formaggia, and L. Antiga. An open-source tool for patient- specific fluid-structure vessel mesh generation. In Modelling of Physiological Flows (MPF), 2013.

Pre-Mesh Elaboration

1. Surface smoothing (optional)

Smoothing the surface

Image segmentation can result in bumpy surfaces, especially if the image quality is not high and one didn't use any curvature term in level sets evolution. Since artifactual bumps in the surface can result in spurious flow features and affect wall shear stress distributions, one may want to increase surface smoothness prior to building the mesh.

`vmtksurfacesmoothing` is a wrapper around vtkWindowedSincPolyDataFilter, which implements one of the non-shrinking filters by Taubin.

By typing:

```
1. vmtksurfacesmoothing --help
```

we see that there are two main parameters controlling the amount of smoothing: passband, which is the cut-off spatial frequency of the low pass filter, and iterations, which is the number of smoothing passes. For more details, visit the vtkWindowedSincPolyDataFilter doxygen page.

For typical vessels, the following should be ok

```
1. vmtksurfacesmoothing -ifile foo.vtp -passband 0.1 -iterations 30 -ofile foo_sm.vtp
```

If you want more smoothing, try using a passband of 0.01. Be careful not to kill surface features by smoothing too much. Also, watch out the apex of bifurcations, since its curvature may decrease resulting in a shallower apex and affecting the simulated hemodynamics.

http://www.vmtk.org/tutorials/SurfaceForMeshing.html

Pre-Mesh Elaboration

2. OPEN Surface (COMPULSORY!)

Opening the surface

Under the assumption that you generated the surface using a deformable model, it's likely that your surface is closed at inlets and outlets, with a blobby appearance. We'll now proceed by opening the surface by clipping the blobby endcaps.

You have at least three options.

The first option is to use `vmtksurfaceclipper.` If you call

```
1.  vmtksurfaceclipper -ifile foo.vtp -ofile foo_cl.vtp
```

a rendering window will show up.

http://www.vmtk.org/tutorials/SurfaceForMeshing.html

Pre-Mesh Elaboration

2. OPEN Surface (COMPULSORY!)

Press `i` to start the interaction. A cube will appear (like in `vmtkimagevoiselector`). Position the cube in such a way that the portion of the surface you want to clip lies inside the cube (it would be easier with a cyberglove, uh?). Try to position one face of the cube perpendicular to the vessel. You don't need to be extremely precise, just do the best you can. It'll get easier with time.

Press the `space bar` to proceed with clipping.

Press `i` again if you want to clip another piece, or `q` if you want to quit.

The third option is to clip endcaps automatically. No 3D interaction involved. Endcap clipping can be performed using `vmtkendpointextractor`, which needs centerlines to be computed beforehand with `vmtkcenterlines`. Let's set up the corresponding pype

```
1.  vmtksurfacereader -ifile foo.vtp --pipe vmtkcenterlines --pipe vmtkendpointextractor --pip
    e vmtkbranchclipper --pipe vmtksurfaceconnectivity -cleanoutput 1 --pipe vmtksurfacewriter
    -ofile foo_ct.vtp
```

When prompted, select one source point on one inlet or outlet and press `q`. Then select as many target points as the rest of inlets/outlets and press `q`.

If you look at the output file, you'll see that the endcaps have been removed. If you want to clip a larger extent of the endcaps, just use the `vmtkendpointextractor` option `numberofendpointspheres` (the default is 2, increase it to 3 or 4 and see what happens).

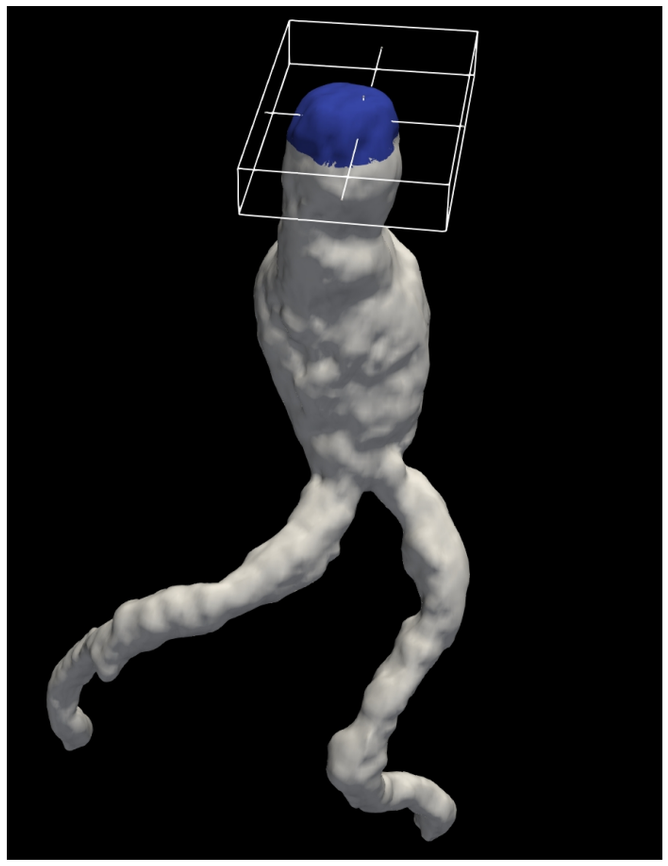
http://www.vmtk.org/tutorials/SurfaceForMeshing.html

Pre-Mesh Elaboration

2. OPEN Surface (COMPULSORY!)

Two tools:

- VMTK: vmtksurfaceclip
- PARAVIEW:
  - Open model_lumen.vtp in paraview
  - Select Clip surface
  - Select box
  - Apply
  - Extract surface
  - Triangulate
  - Save in .stl



NB: Paraview is based on VTK (same as VMTK!)

Pre-Mesh Elaboration

3. Add flow extensions (optional)

Adding flow extensions

Flow extensions are cylindrical extensions added to the inlets and outlets of a model. They are important for ensuring that the flow entering and leaving the computational domain is fully developed, so that fully developed boundary conditions aren't forcing the solution in the actual vessel.

Adding flow extensions is a typical problem in CFD modeling. Stitching cylindrical flow extension to an inlet or outlet of a realistic vessel is not a trivial task, and may result in worsening the reproducibility and adding operator-dependence to the modeling procedure. A fully automatic procedure can solve this often overlooked problem preserving reproducibility and speeding up the modeling phase considerably.

Once again, adding flow extensions relies on centerlines. This time centerlines can use open inlet and outlet profiles for the definition of seed and targets.

Let's see how this is done

```
1.  vmtksurfacereader -ifile foo.vtp --pipe vmtkcenterlines -seedselector openprofiles --pipe
    vmtkflowextensions -adaptivelength 1 -extensionratio 20 -normalestimationratio 1 -interacti
    ve 0 --pipe vmtksurfacewriter -ofile foo_ex.vtp
```

The `adaptivelength` argument of the `vmtkflowextensions` script is a boolean flag which enables computing the length of each flowextension proportional to the mean profile radius. The proportionality factor is set through `extensionratio`. The `normalestimationratio` argument controls how far into the centerline the algorithm looks for computing the orientation of the flow extension.

http://www.vmtk.org/tutorials/SurfaceForMeshing.html

Mesh Generation

Run FSIMESHGENERATOR (for fluid and solid mesh)

```
(vmtk) fsimeshgenerator -ifile lumen.stl -fluidname fluid.mesh -solidname solid.mesh
```

vmtk fsimeshgenerator --help
Executing fsimeshgenerator --help

Automatic piping LifeVFSIgeneratorOpt
Parsing options LifeVFSIgeneratorOpt

LifeVFSIgeneratorOpt : the code generates a fluid mesh and a structure mesh in LifeV format starting from an interface mesh. Multiple areas delineation is also supported.
Input arguments:

-ifile SurfaceInputFileName (str,1): filename for the default surface reader
-fluidname LifeVFluidMeshName (str,1); default=None: fluid mesh file name (with .mesh extension)
-solidname LifeVSolidMeshName (str,1); default=None: solid mesh file name (with .mesh extension)
-scale ScaleFactor (float,1) >= 0.0; default=1: scaling parameter to scale the mesh coordinates
(e.g. from mm to cm -scale 0.1)

Mesh Generation

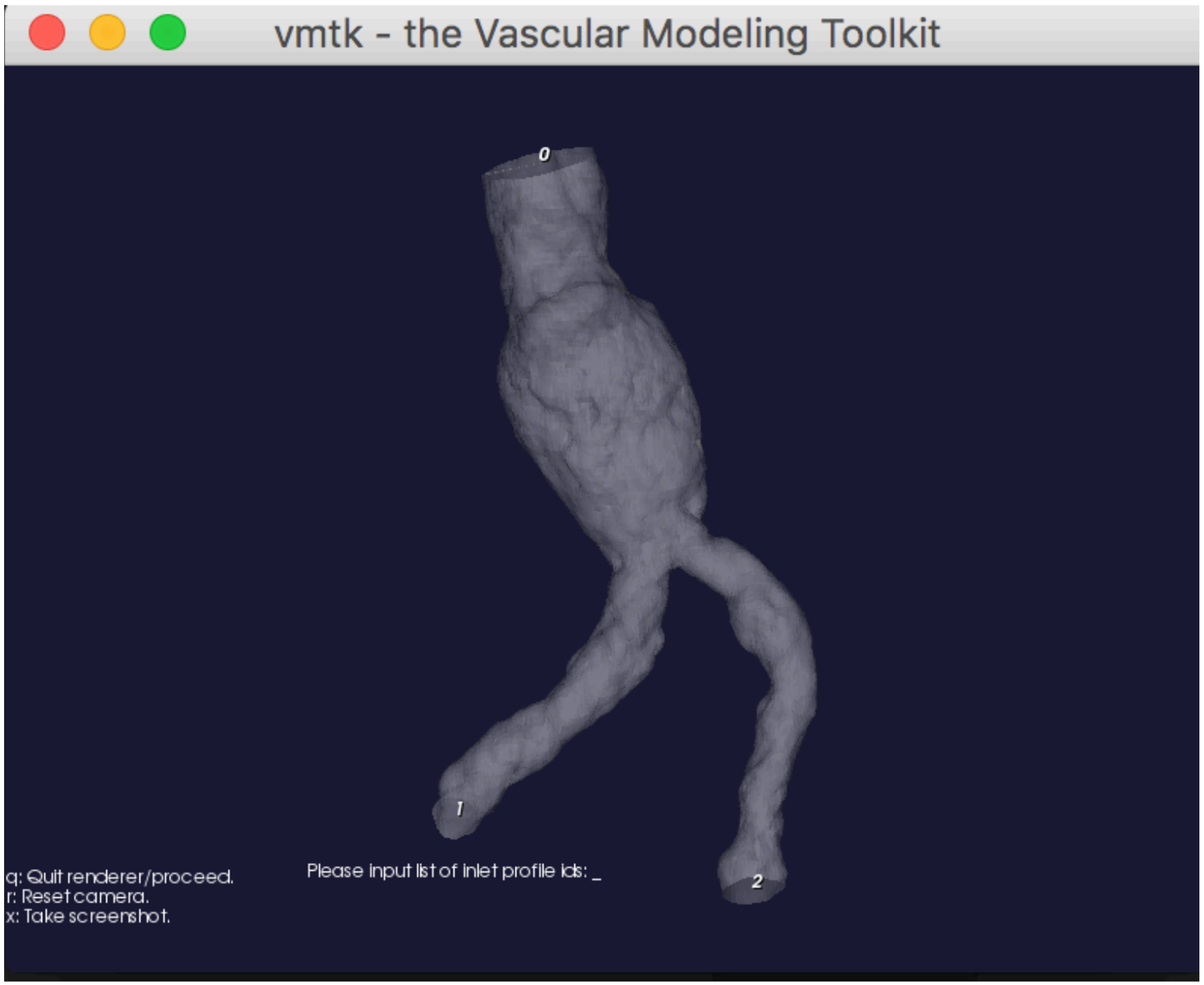
Run FSIMESHGENERATOR (for fluid and solid mesh)

```
(vmtk) fsimeshgenerator -ifile lumen.stl -fluidname fluid.mesh -solidname solid.mesh
```

FLUID MESH:

1.Centerline computation
Input labels
Output labels
NB: surface must be OPEN!

2.Surface remeshing
3.Changing volume factor
4.Adding boundary layer



Mesh Generation

Run FSIMESHGENERATOR (for fluid and solid mesh)

```
(vmtk) fsimeshgenerator -ifile lumen.stl -fluidname fluid.mesh -solidname solid.mesh
```

FLUID MESH:

1.Centerline computation

2.Surface remeshing

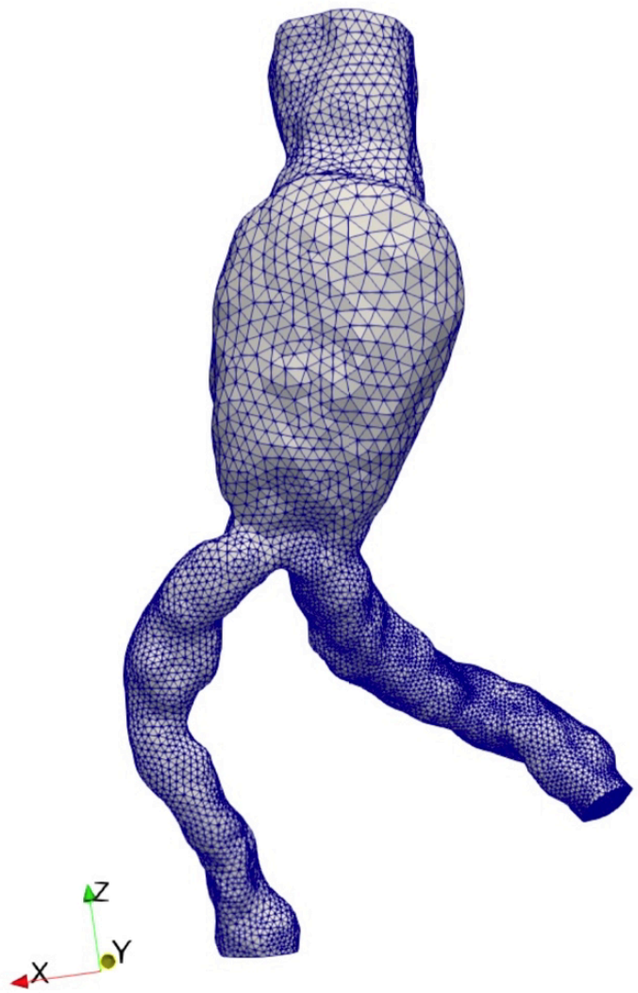
- Constant remesh (h)
- Radius dependent remesh (alpha and beta)

  NB: check units before meshing

  (the -scale option is used at the end, only in the writing of the results)

3.Changing volume factor

4.Adding boundary layer



Mesh Generation

Run FSIMESHGENERATOR (for fluid and solid mesh)

```
(vmtk) fsimeshgenerator -ifile lumen.stl -fluidname fluid.mesh -solidname solid.mesh
```

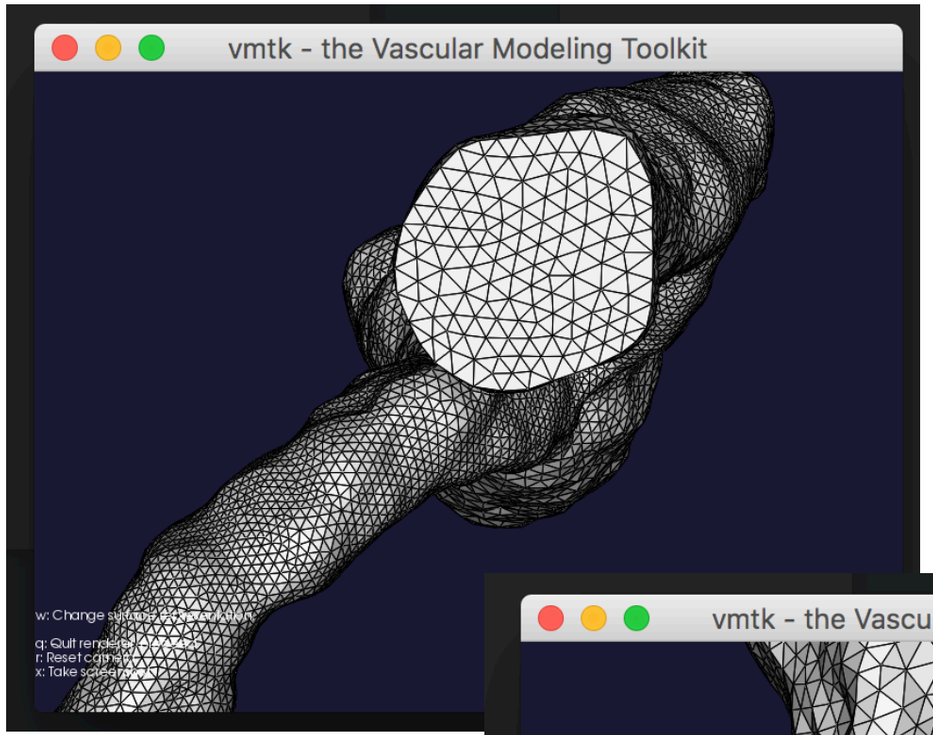
FLUID MESH:

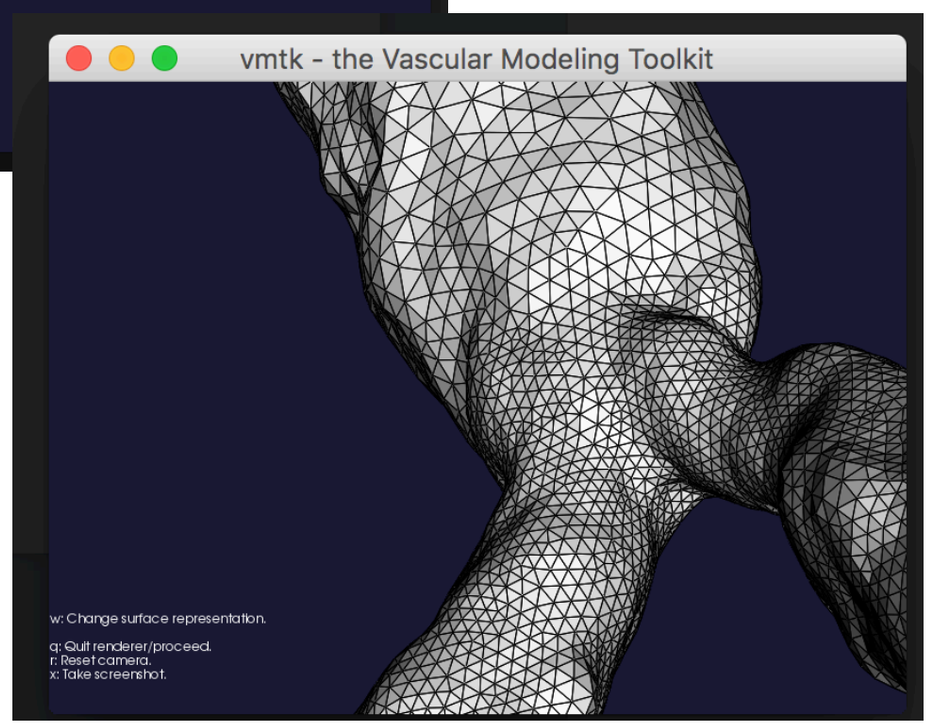
1. Centerline computation
2. Surface remeshing
3. **Accepting result/Changing volume factor**

   Tips: check the mesh at inlets/outlets (especially if you have vessels of different sizes and you selected a constant mesh grid), check at bifurcation points or points where the surface mesh was not regular, check the edges

   Volume element factor

4. Adding boundary layer





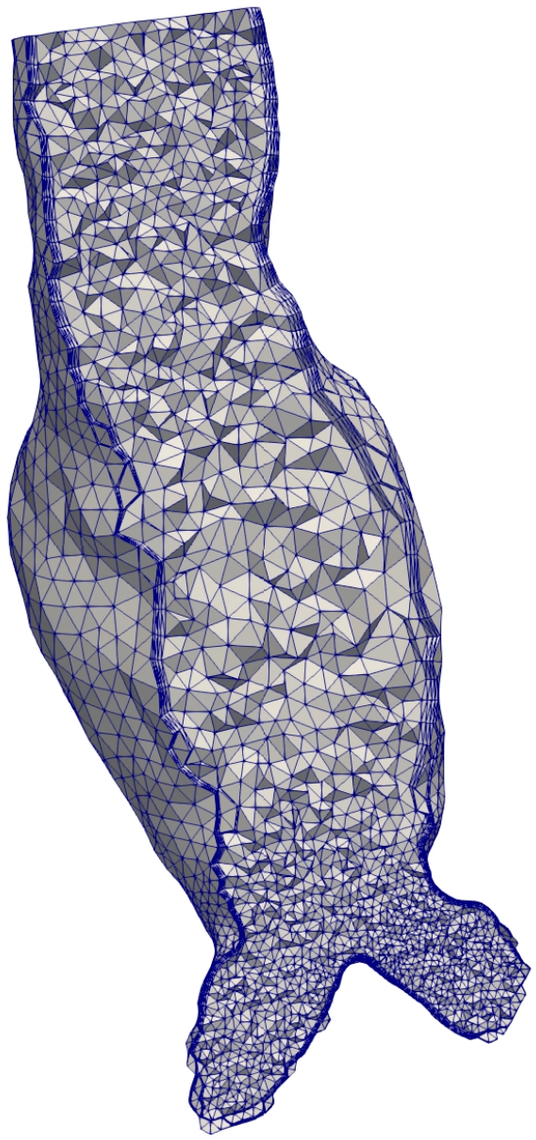
Mesh Generation

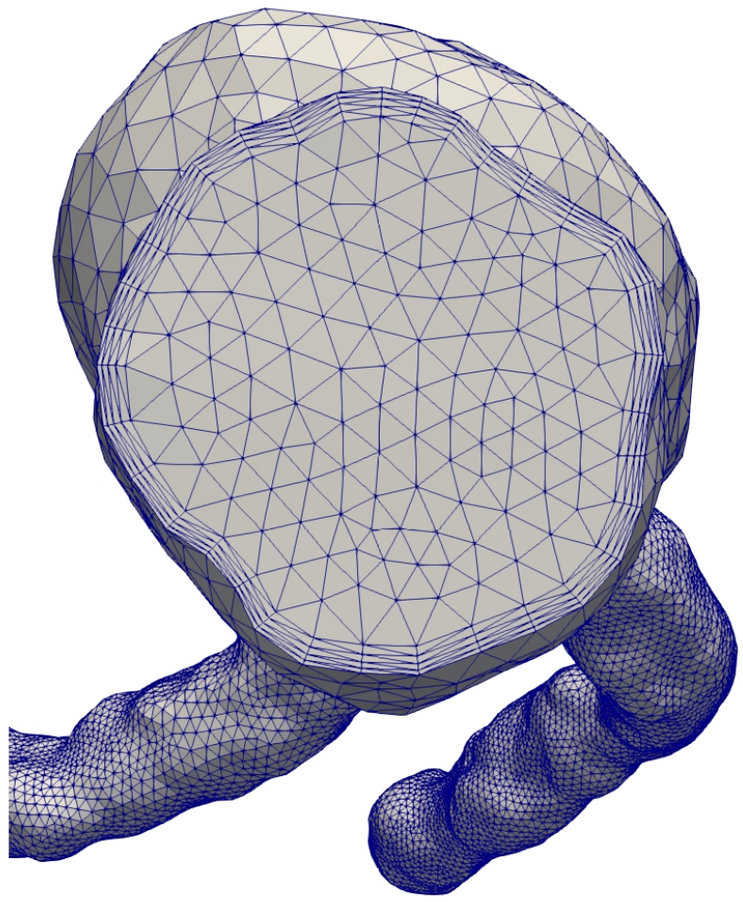
Run FSIMESHGENERATOR (for fluid and solid mesh)

```
(vmtk) fsimeshgenerator -ifile lumen.stl -fluidname fluid.mesh -solidname solid.mesh
```

FLUID MESH:

1.Centerline computation
2.Surface remeshing
3.Changing volume factor
4.Adding boundary layer
Number of layers
Ratio between layers
Boundary layer thickness:
constant (h)
radius dependent (alpha and beta)
NB: check messages on terminal!





Warning: boundary layer may worsen the conditioning number of your stiffness matrix!

Mesh Generation

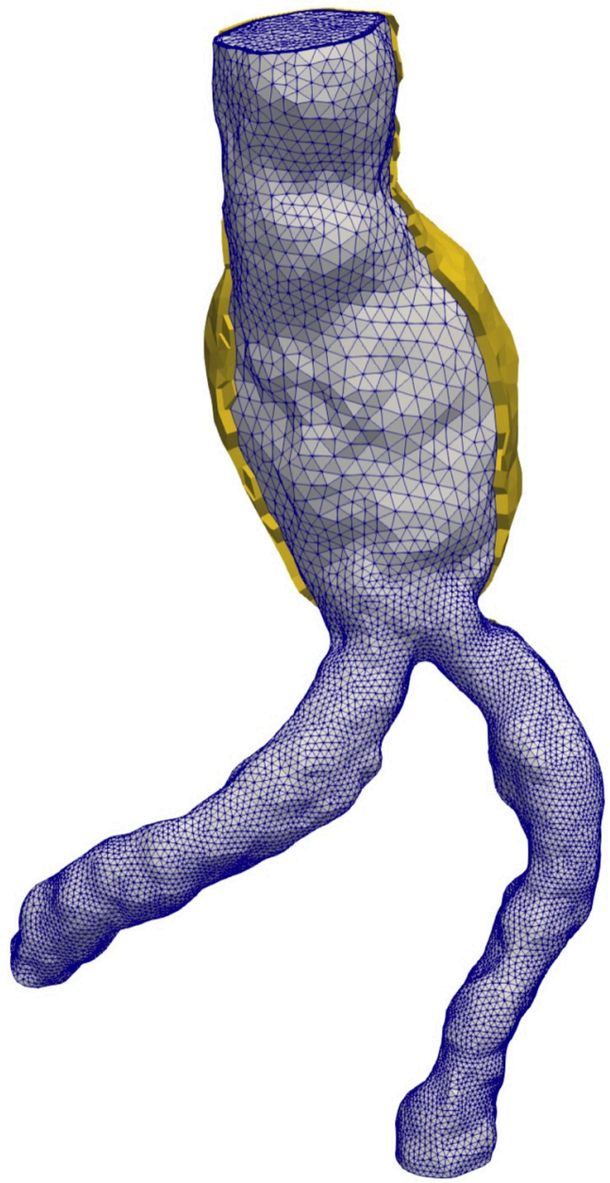
Run FSIMESHGENERATOR (for fluid and solid mesh)

```
(vmtk) fsimeshgenerator -ifile lumen.stl -fluidname fluid.mesh -solidname solid.mesh
```

SOLID MESH:

1.Number of layers
2.Type of Extrusion:
Constant
Radius dependent (linear)
NB: check units before meshing
(the -scale option is used at the end, only in the writing of the results)

3.Volume region differentiation
4.External interface region differentiation



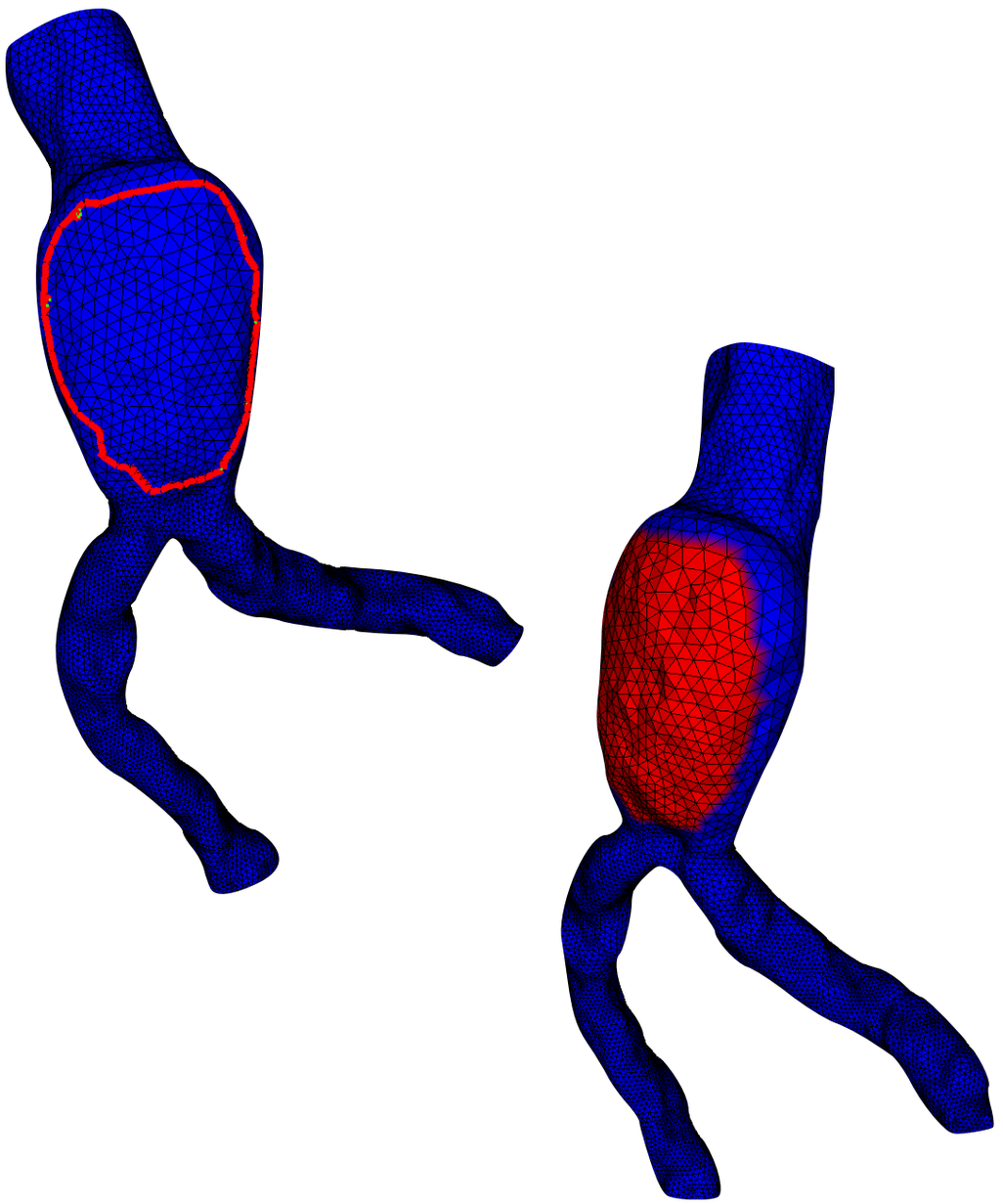
Mesh Generation

Run FSIMESHGENERATOR (for fluid and solid mesh)

```
(vmtk) fsimeshgenerator -ifile lumen.stl -fluidname fluid.mesh -solidname solid.mesh
```

SOLID MESH:

1. Number of layers
2. Type of Extrusion
3. **Volume region differentiation**
   - Manual region drawing
   - Region selection
   - Thickness changes
4. External interface region differentiation



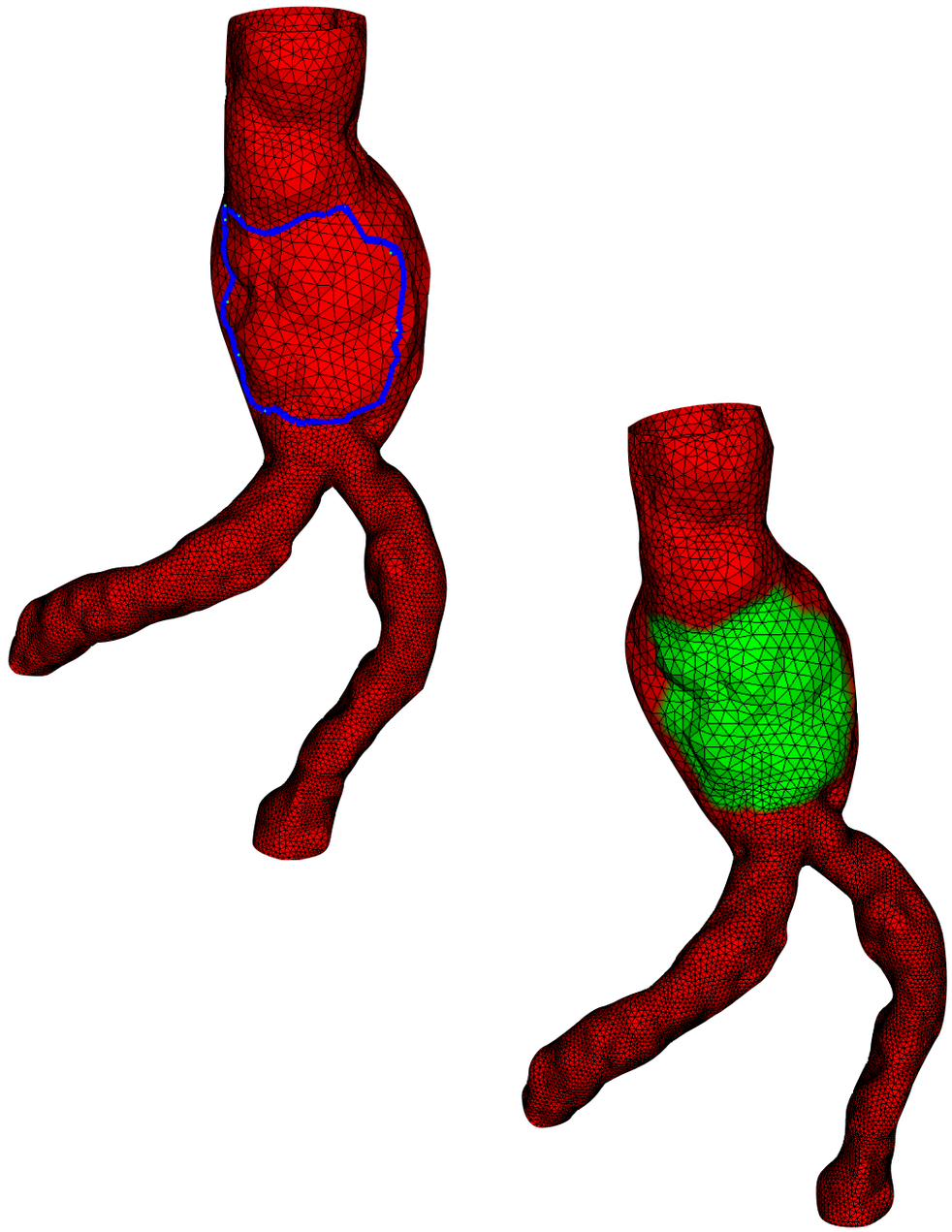
Mesh Generation

Run FSIMESHGENERATOR (for fluid and solid mesh)

```
(vmtk) fsimeshgenerator -ifile lumen.stl -fluidname fluid.mesh -solidname solid.mesh
```

SOLID MESH:

1. Number of layers
2. Type of Extrusion
3. Volume region differentiation
4. **External interface region differentiation**
   - Manual region drawing
   - Region selection
   - Thickness changes



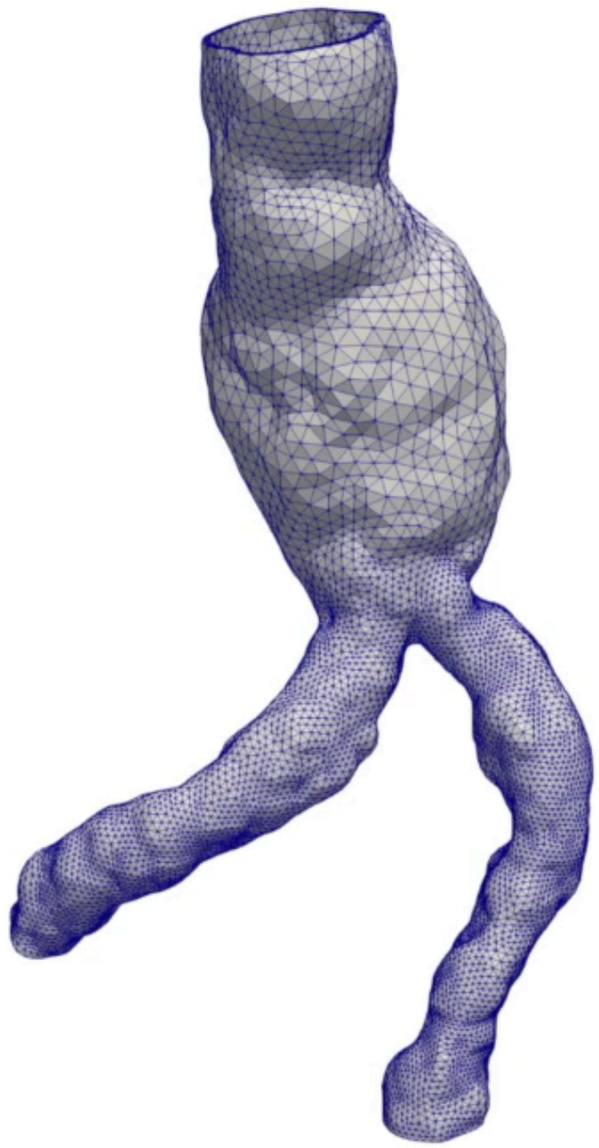
Mesh Generation

Run FSIMESHGENERATOR (for fluid and solid mesh)

```
(vmtk) fsimeshgenerator -ifile lumen.stl -fluidname fluid.mesh -solidname solid.mesh
```

SOLID MESH:

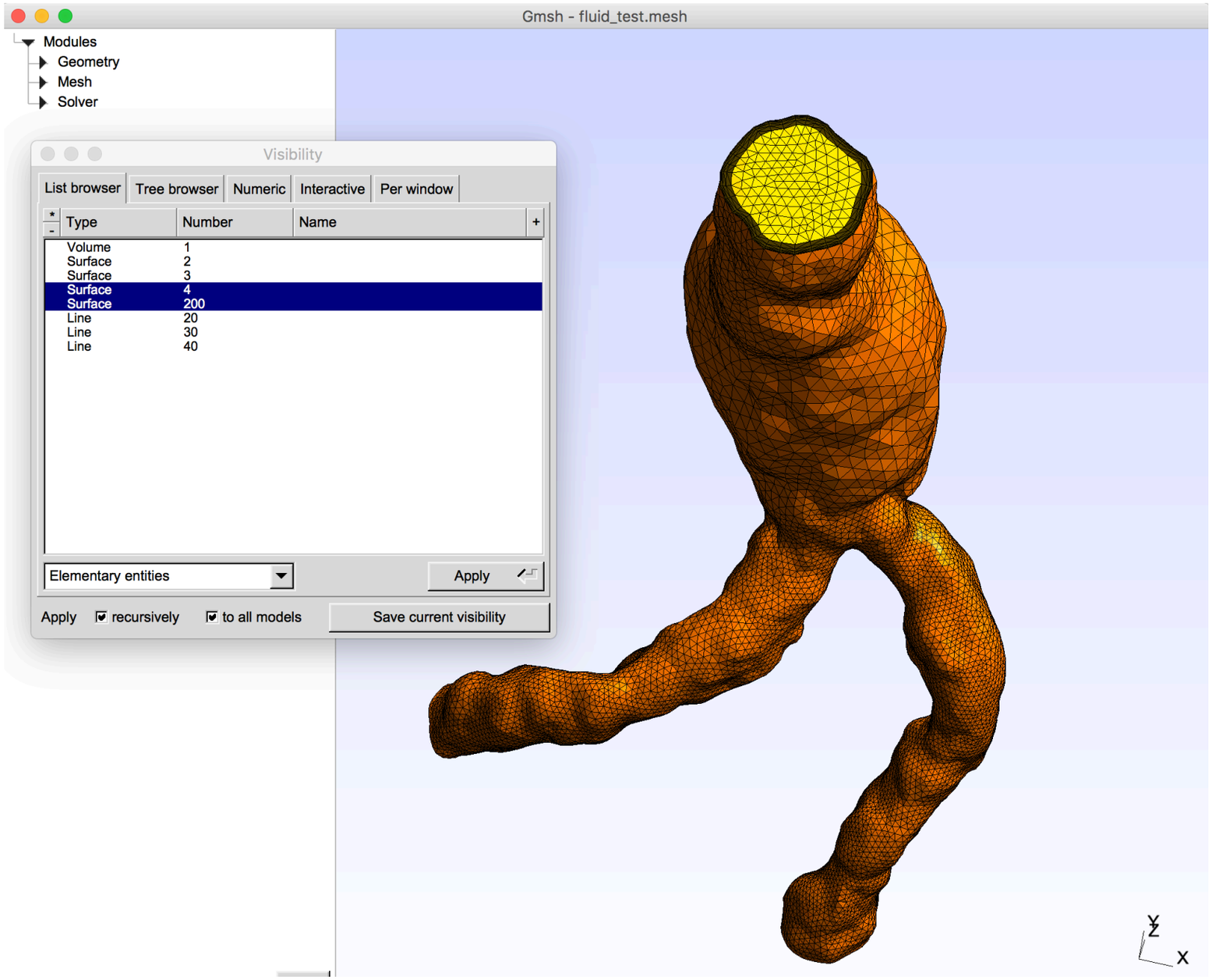
1. Number of layers
2. Type of Extrusion
3. Volume region differentiation
4. **External interface region differentiation**
   - Manual region drawing
   - Region selection
   - Thickness changes



Mesh Generation

Always check the boundary flags before your simulation!

Interface and External surface (solid) flags are fixed but the flags of at inlet/outlet surfaces may differ from one mesh generation to another (even when starting with the same surface).



Mesh Generation

Summary

What you can do with the *base* script

- Constant size/radius dependent size grid
- Scale units
- Add a boundary layer in the fluid mesh
- Constant size/radius dependent size boundary layer thickness
- Change volume element factor
- Constant/radius dependent extrusion for the solid mesh
- Define manually different flags on the solid external surface
- Define different volume flags in the solid mesh
- Define different thickness in different volumes of the solid mesh

Am I missing something?

Mesh Generation

Summary

What you cannot do with the *base* script

- Define manually different flags on the fluid-solid interface
  (Doable, look inside the code and repeat what it is done for the external solid surface on the interface!)

- Define different volume flags for the fluid
  (Doable but you have to define how to select the internal boundaries)

- Define different volume flags on different solid extrusion layers
  (probably doable with a bit of coding in vtk)

- Generate a mesh for a closed surface
  (I use Gmsh for closed surfaces)

- Control the value of the inlet/outlet boundary flags
  (but you can change them later in whatever you want using vtk or gmsh)

Any wishes?

Mesh Generation

Exercise

Play with fsimeshgenerator with the abdominal aortic aneurism with renal arteries:

Add flow extension at the inlet surface
Generate both fluid and solid mesh
Try constant mesh size and radius depend mesh size
Try to add a boundary layer

Pay attention to the renal bifurcations