**Simulation Tutorial**

**Part 1: Basic Steps**

**1. Create a Geometry in Rhino.**

Use Rhino to create an geometry. Here we can use millimeter and large object (We will have to rescale it to centimeter later in order to conform with lifeV).  Be careful with the connections between solid shapes. A solid in Rhino is a surface completed sealed with no openings. Save the geometry in stl format and make sure to choose ASCII scheme when saving. For this write-up we simply name the geometry stent.stl

**2.      Create the Mesh of the Geometry in Netgen.**

Open netgen in terminal with command:

     $        netgen &

Use *mesh options* to adjust the finest of the mesh.

Use *mesh options >mesh size* to make the mesh more uniform if needed.

After the mesh is generated, we need to label all the boundaries with *edit boundary conditions*: first label everything 30, this will make give all walls 30; then label inflow 10 and outflow 20. Note here the wall has to have a greater number. Then save the mesh in vol format. It is a good idea to put all vol file in one directory. Here we call it stent.vol

**3.       Re-scale the Mesh for the Simulation.**

It is a good idea to create a directory to store all vol files.  For me, I created the following folder to put all vol files:

     /aut/proj/lifeV2/boyi/MeshFiles/stent.vol

Usually the best way is to make the geometry in the right scale in Rhino to avoid re-scaling later. However, if the Stl is already created in the wrong scale we can still re-scale it. In this folder *StlFiles*, there is executable file link named re-scale\_mesh. This is only a link to the executable file in Tiziano's folder, but we can use it to re-scale the stent.vol. Before we run reescale\_mesh, we need to set up perimeters in file data\_rescale which is in the same folder. Open data rescale in gedit and make changes in the following two lines:

          mesh\_file = stent.vol

          new\_mesh\_file = stent\_cm.mesh

 Note that the new mesh is in mesh format instead of the vol format. From the following line we can see that the executable would only work if we put the mesh under the same directory:

          mesh\_dir = ./

Of course we can edit this line to make the executable work in different directory. Now run rescale\_mesh in crunch (or kinbote):

          $ ./rescale\_mesh

Then we should see a file name stent\_cm.mesh in folder StlFiles.

**4.       Set Up a New folder before the Simulation to Store the Results.**

It is highly recommended that you created one folder for each simulations. For example, here we can create a folder using the following commands (or create it in the graphical interface):

          $        mkdir /aut/proj/lifeV2/boyi/simulations/2013\_3\_21StentTest

Enter the folder we just created:

          $        cd /aut/proj/lifeV2/boyi/simulations/2013\_3\_21StentTest

In this folder 2013\_3\_21StentTest we create a link to the executable in the under the testFolder folder:

          $        cp ../testFolder/applicationBloodFlow.exe .

(Note here the dot in the end means the destination is current directory.)

In fact the the executable in testFolder is a only a symbolic link to the one in Tiziano's folder. Thus we can also just create another link instead of copying it:

           $       ln -s ../testFolder/applicationBloodFlow.exe .

Then we need to copy the solverOptions.xml to the folder. SolverOptions is a file we don't normally change.

We also need to copy the data file from ApplicationBloodFlow to this folder:

          $        cp /aut/proj/lifeV2/boyi/simulations/testFolder/data .

 Open data in gedit and make the following changes:

 *inflowList       = '10'*

*wallList         = '30'*

*outflowList     = '20'*

*[fluid/space\_discretization]*

*mesh\_dir         = ../../MeshFiles/*

*mesh\_type        = .mesh*

*mesh\_file        = stent\_cm.mesh*

These changes will match the data file with the boundary labels in the mesh file stent\_cm.mesh and point it to the right location of the mesh. The data file can also be edited in VI directly and VI will be discussed in later tutorial. We can of course change other things such as end time and time step.

We also need to have a inflow file:

 $        cp /aut/proj/lifeV2/boyi/simulations/testFolder/inflow0.dat .

We can of course edit the inflow0.dat file in gedit or VI to customize the simulation. The numbers are just the flow rates in the first period. The numbers are equally spaced and then connected by a spline curve to form the inflow of the first period. We can also use patient specific inflow file, and there is one patient specific inflow file from a previous project with Dr. Samady and it is now in folder:

     /simulations/2013\_4\_2CurvyStented2/

If we use patient specific inflow file and its numbers are big, this usually means that the inflow is velocity but not flux. Thus we have to change the dataKind in the the data file:

            *dataKind = velocity*

*input\_file   = inflow0.dat*

We also want to adjust the initial time so that the first picture is saved right after the preload.

**5.       Run the Simulation**

Now we can run the executable file:

$ ./applicationBloodFlow.exe

We can also use multiple processors by the command:

$ mpirun -n 4 ./applicationBloodFlow.exe

**6.       Visualize the results in ParaView**

Connect to kinbote (used to be crunch) with display:

 ssh –X kinbote

If you are not a department server, use the following command:

          ssh -X username@kinbote.mathcs.emory.edu

Then run ParaView: (also works on lumens)

          $ /aut/proj/lifeV2/paraview/ParaView-3.14.1-Linux-64bit/bin/paraview &

We want to open case000, then click apply. ParaView will automatically load the rest of the cases. We can save snapshots and animations of the simulations with ParaView. More about ParaView will be discussed in later tutorial.