Fluid Flow Simulations for Biomedical Applications

Samuel Scruggs

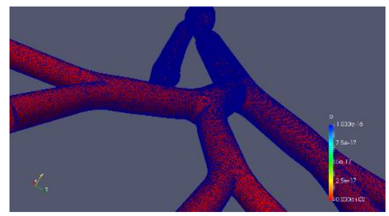
Mentor: Kwai Wong

August 5, 2016

1. Abstract

Using Computational Fluid Dynamics (CFD) the result of fluid flow through certain geometries can be calculated. The nature of the flow can be predicted with outside factors such as initial velocity of the flow, initial pressure of the fluid, and kinematic viscosity of the fluid to name a few. Taking all these factors into account, a detailed prediction of the fluid can be created. With the aid of this prediction, components of the flow can be altered in order to create the outcome that is wanted.

Programs have been written to help create these fluid predictions and simulate the outcomes in order to show the effects of the fluid through the different geometries. In this project, various programs were used in order to view an outcome of how particles flow through the human airways. As particles flow many different factors can speed them up, slow them down, gather them in one area and many more situations that can affect the lungs, trachea, and other parts of the airways in many different ways. With the result of these simulations, how these particles affect the airways in certain areas can be illustrated in order to study these effects.



2. Simulation Process

For the case of this project, simulating the fluid flow process involved many steps dealing with different programs to create each piece needed for an accurate result. Starting with the creation of the geometry, each program has a specific task that helps to describe what is going on and what the preceding program needs to do.

2.1 Geometry Creation

Using any preferred program, the creation of a simple shape or complex geometry is needed in order to have the actual object that the fluid will flow through or around. In the case of this project, the program AutoDesk Inventor was used in order to create a straight, 3-Dimensional pipe and an elbow pipe. Common geometries such as the two previously stated can be used as benchmark tests in order to assure that the process is being run correctly. The outcomes for these geometries is already known, and a comparison to the known outcome can aid in adjusting the user input conditions in order to attain the most realistic result.

2.2Meshing

In the meshing process, the geometry is broken down into many smaller geometries that can be analyzed individually in the solver. In the 2-Dimensional case, the main geometry is broken down into smaller triangles while the 3-Dimensional case breaks down into tetrahedral shapes. Each shape creates a cover that is wrapped around the main geometry and can be altered through user input.

Receiving the best outcome relies heavily on the meshing process. The detail of the mesh will determine how many points are put through the solver for that particular geometry. User input is critical in this step because a mesh with too few points can give an inaccurate result. In this case, more is better. As the number of points increases in the mesh (i.e. the geometry is broken down into smaller shapes) the solver has less data points it actually has to interpolate.

Compared to the mesh on the left, the right mesh has far more points. The final result produced will be much more accurate for the right mesh.

In the meshing process, the boundary conditions can be established for the geometry. For the case of the straight pipe, one side is deemed an inlet, the other an outlet, and the surrounding area is considered the boundary for the flow. This is described in the files that are written by the meshing program so that when the solving of the flow actually takes place, the program running the solver can execute the equations with the proper boundary conditions in place.

2.3 Solver

In the process of solving, the points on the geometry are used in a series of equations that solve the fluid flow at each particular point. In this case the incompressible Navier-Stokes equations were used to solve the flow.

$$\rho\left(\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}\right)=-\frac{\partial p}{\partial x}+\mu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right)+\rho g_x$$
$$\rho\left(\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z}\right)=-\frac{\partial p}{\partial y}+\mu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 v}{\partial z^2}\right)+\rho g_y$$
$$\rho\left(\frac{\partial w}{\partial t}+u\frac{\partial w}{\partial x}+v\frac{\partial w}{\partial y}+w\frac{\partial w}{\partial z}\right)=-\frac{\partial p}{\partial z}+\mu\left(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}+\frac{\partial^2 w}{\partial z^2}\right)+\rho g_z.$$

$$\textbf{Continuity:}\quad \frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0.$$

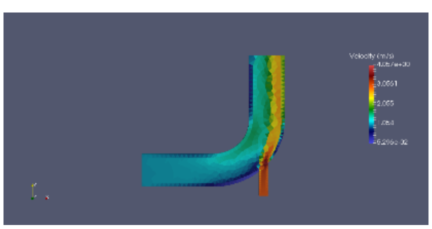
The main concept of these equations involves the conservation of mass, energy, and momentum. Whatever flows through the inlet of the geometry must maintain that same state as it exits. In the case of the 3-Dimensional pipe, the friction between the walls and the water slows the flow on the outside, therefore the speed of the water increases closer to the middle. This continuity holds true for all geometries and can be applied to the one used in this project.

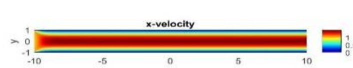
In the solver OpenFOAM the initial conditions of the simulation can be established in the velocity and pressure files. Typically titled U and p, within these files are the lists of conditions for each boundary of the geometry. The inlet of the geometry is set up with an initial velocity while the outlet usually has a certain pressure. The boundaries of the geometry that enclose the flow are given conditions as to how they interact with the flow. For example, the walls of the straight and elbow pipes are categorized as no slip boundaries.

2.4Post-Processor

Post-processing allows for the results to be viewed and analyzed after going through the solver process. The program used in this case, Paraview, displays results of velocity and pressure over the period of time the fluid flows through the geometry. Not only does it display the entire geometry at once, but it also has the ability to display slices of certain areas in the geometry itself. Along with these visuals, graphs can be made that show exact values throughout the shape.

The illustrations below show the velocity of water through both an elbow pipe and a straight pipe. With a legend to help show what the colors represent, the results are analyzed by viewing which colors are darker compared to lighter and checking these results with the initial and boundary conditions. In the example of the elbow tube, the inlet velocity through the horizontal, larger inlet is 1 m/s while the velocity through the small inlet is 3 m/s. The darker coloring goes with the smaller inlet to show that the velocity is faster.





3. File Formatting

Transferring a file from one program to the next requires proper formatting in order for that next program to properly read the file. When the geometry is created in a CAD program, it can be converted into an stl file. For the case of using the meshing program Gmsh, the best option is an stl file because the program is able to read them. Simple geometries such as a tube or an elbow pipe can be created in Gmsh, but when it comes to something more complicated, a CAD program is necessary. In Gmsh, using "add points", points for simple shapes can be plotted for lines to connect them. This helps in making squares, circles, and other shapes. Once these simple shapes are constructed, commands such as extrude surface can be used to transform 2-Dimensional shapes into 3-Dimensional. The surfaces of this 3-Dimensional shape can be highlighted and named in order to specify them as boundaries.

Once in Gmsh, a mesh file can be created. This is simply done by going under the mesh tab and clicking 3D mesh. The mesh can be refined in the tools tab by going to options, mesh, and adjusting the element size factor of the mesh. With this file comes the boundary names and conditions. The formatting of this file is vital for how the solver will eventually read it. An msh file is created by Gmsh and can be converted directly to OpenFOAM using the command gmshToFoam. This then takes the mesh file and converts it to Foam along with the naming scheme for the geometry. In this step, the system directory and constant directory must be present in order to execute the command. This command reads files from these folders that allow OpenFOAM to convert the file.

When the conversion to OpenFOAM is complete, the command icoFoam can be run in order to use the icoFoam solver. This is the solver used to solve the incompressible Navier-Stokes equations for the fluid flow. This solver utilizes the 0 directory for the initial conditions and the constant file for transport properties. Any simple misstep for a file that is converted will result in an error during solving. The velocity and pressure file within the 0 directory and the transport properties file must all match in order for OpenFOAM to properly execute solving the simulation.

The final step of the process is written by OpenFOAM and readable in Paraview. The file can be opened and the results viewed in the Paraview program in order to analyze how the fluid flows in a certain simulation. In order to access the program, the command paraview is used. This opens a window where you can load the simulation that was just run. As stated before, Paraview allows for many different viewpoints of the results, including graphs and slices of the geometry, so that the accuracy of the results can be assessed. In order to refine the results, the whole process can be redone with a finer mesh or more realistic boundary and initial conditions.

4. Appendix (3D Tube)

I. Gmsh

a. In Gmsh create a new file to create the geometry and mesh

b. Click Geometry → Elementary Entity → Add → Point

   i. Add points to create a circle

c. Back Add → Circle Arc

   i. Choose the begging, middle, then endpoint to create an arc

   ii. Use this process to create four arcs for a circle

d. Click Add → Plane Surface to create a surface within the circle

e. Press the circle itself as the area of the surface, then press ‘e’ to add the surface and ‘q’ to finish the addition

f. Click Translate → Extrude Surface

   i. Highlight the surface and dictate what distance in the z direction the circle will extrude. Again press ‘e’ to execute and ‘q’ to quit

g. Click Physical Group → Add → Surface and select the individual surfaces to be categorized as certain boundaries (i.e. Inlet, Outlet)

h. Click Mesh → 3D to create a mesh for the geometry

i. The mesh can be refined by going to Tools → Options → Mesh → General and adjusting the Element Size Factor

j. Go to File → Save Mesh and this will save the mesh as a .msh file

II. OpenFOAM

k. Copy the newly created mesh file into a new directory

l. Within this new directory, the ‘constant’, system’, and ‘0’ folder should all be present and can be acquired either on the OpenFoam website or by copying these folders from the examples downloaded along with OpenFoam

    i. OpenFoam uses prewritten code from the systems file to format the Gmsh file in a way that is readable for OpenFoam

m. In the new directory, time the command ‘gmshToFoam (Filename).msh’ to create a mesh file that OpenFoam can read

    i. Doing this should create a ‘Polymesh’ folder in the ‘Constant’ directory
    ii. This folder contains all the boundaries and patches for the mesh

n. Within the ‘0’ directory, there are two files labeled ‘U’ and ‘p’ that contain initial conditions for velocity and pressure in the geometry. The names for the boundaries of the geometry in these files must match the names in the ‘boundaries’ file in the ‘Polymesh’ directory.

o. In the ‘U’ and ‘p’ files, conditions can be assigned to the boundaries based on their purpose (i.e. Inlet, Outlet, noSlip)

p. Under the ‘system’ directory, the file labelled ‘contolDict’ contains the adjustments for time the simulation runs and at what time step the data is recorded. Kinematic viscosity can be adjusted under the ‘transportSystems’ file in the ‘constant’ directory

q. After double checking to make sure all the boundary conditions and names are set properly, return to the newly created directory where the mesh file should be along with the ‘0’, ‘constant’, and ‘system’ directory. At this point, the command ‘icoFoam’ can be run

    i. icoFoam is the solver used for this simulation. It uses incompressible Navier-Stokes equations to solve the fluid flow

III. Paraview

r. After the solver has run its course, there should be new files within the directory that are numbered based on what time step was set for the collection of data

s. Enter the command ‘paraview’ to bring up the Paraview program to view the results of the solver

t. Click Open → search for the system folder and open the controlDict using the OpenFoam option

u. The result will be loaded into Paraview and then the ‘apply’ button is clicked to view the result

    i. Paraview has a wide variety of options to view results including individual slices of the geometry and graphs. Using various tutorials, Paraview is simple to navigate and can lead to the result that is needed

5. References

"ANSYS FLUENT Produced by FLUENT Inc. Version-12." N.p., n.d. Web. 4 Aug. 2016.

"Gmsh." *: A Three-dimensional Finite Element Mesh Generator with Built-in Pre- and Post-processing Facilities*. N.p., n.d. Web. 04 Aug. 2016.

"OpenFOAM | The OpenFOAM Foundation." *OpenFOAM*. N.p., 2016. Web. 04 Aug. 2016.

"Sandia National Laboratories: CUBIT Geometry and Mesh Generation Toolkit Technical Support." *Sandia National Laboratories: CUBIT Geometry and Mesh Generation Toolkit Technical Support*. N.p., n.d. Web. 04 Aug. 2016.

***SourceForge*. N.p., n.d. Web. 04 Aug. 2016.**

"Welcome to ParaView." *ParaView*. N.p., n.d. Web. 04 Aug. 2016.

6. **Acknowledgments**

This project is supported by the National Science Foundation, The Joint Institute for Computational Sciences, and The University of Tennessee.