Project Number:

Modeling Fluid Flow Using Fluent

A Major Qualifying Project Report

Submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Bachelor of Science
in Mechanical Engineering

by
Jonathan Zoll

Date: 12/ 15/09

Approved:

Prof. Gretar Tryggvason, Major Advisor

Abstract

The study of fluids is vital for our understanding of the world. Traditionally this was done through studying fluid flow on models in something like a wind tunnel, but in the last century the field of computational fluid dynamics has come into being. One program that is capable of modeling fluid flow is Fluent. The aim of this project was to model a few scenarios using Fluent. The purpose of doing so was to see how accurate the program was at modeling fluid flow in order to see if computational fluid dynamics has advanced enough to do away with the traditional methods. After running simulations in both 2D and 3D I found that Fluent is not quite ready to replace the wind tunnel.

1.0 Introduction

The study of fluid has been around for millennium, dating back to ancient Greece, but their understanding did not go beyond what they needed to know to run aqueducts and other waterworks. Da Vinci further pursued the topic during the Renaissance observing waves and free jets. Even Newton studied fluids. The topic did not mature until people like Bernoulli and Euler investigated it and developed equations that were later named after them. The Euler equations were further modified by Claude Louis Marie Henry Navier and George Gabriel Stokes to create the Navier-Stokes equation. These men laid the groundwork that would be the foundation of computational fluid dynamics.

Computational fluid dynamics is a term used to describe a way of modeling fluids using algorithms and numerical methods. Currently they are solved utilizing computers but early methods were completed manually without the aid of a computer. Computational fluid dynamics are a powerful tool to model fluids, but even with the most state of the art supercomputers and technological advances they are only an approximation of what would occur in reality.

It is unclear exactly when computational fluid dynamics came into being. Lewis Fry Richardson attempted to predict the weather by creating a grid in physical space and using Bjerknes's "primitive differential equations". His method involved a stadium of 64,000 people each using a mechanical calculator to solve part of the flow equation. It ended in failure. In 1933, A. Thom was able to numerically compute flow past a cylinder. Another mechanical solution was made by M. Kawaguti which took 20 hours a week over 18 months. NASA's theoretical division also made contributions during the 1960s, but it wasn't until the 1980s when commercial methods for computational fluid dynamics became available.

2.0 Method

2.1 What is CFD?

CFD stands for computational fluid dynamics. It is a way of modeling complex fluid flow by breaking down geometry into cells that comprise a mesh. At each cell an algorithm is applied to compute the fluid flow for the individual cell. Depending on the nature of the flow either the Euler or Navier-Stokes equations can be used for the computation.

2.1 What is Fluent?

Explaining how to use FLUENT cannot be done without discussing GAMBIT first. GAMBIT *(Figure 1)* is an application that is distributed along with FLUENT. As of this writing, it is owned and distributed by ANSYS, Inc. GAMBIT is used as a tool to generate or import geometry so that it can be used as a basis for simulations run in FLUENT. It can either build a model or import existing geometries from various other CAD applications. With a geometry in place it generates a mesh for the surface and volume of the geometry allowing it to be used for computational fluid dynamics.

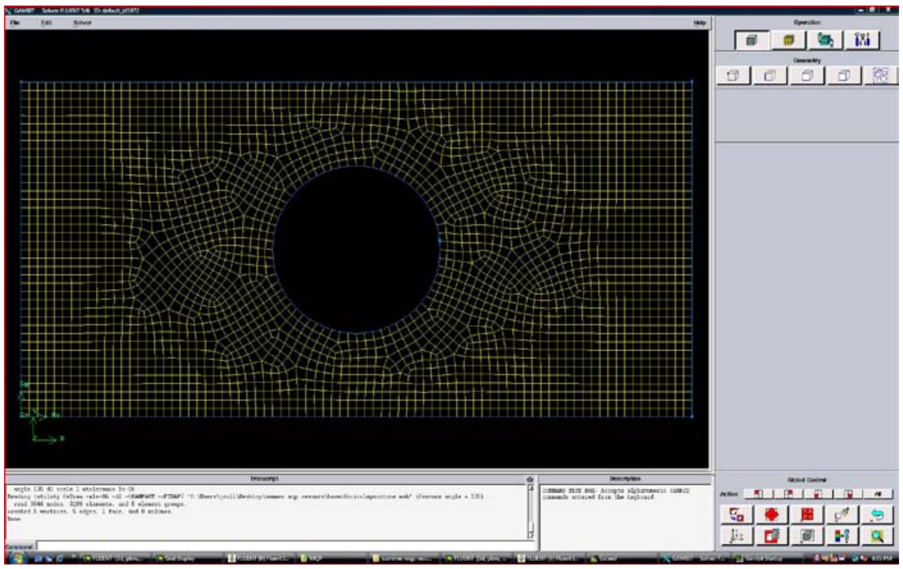
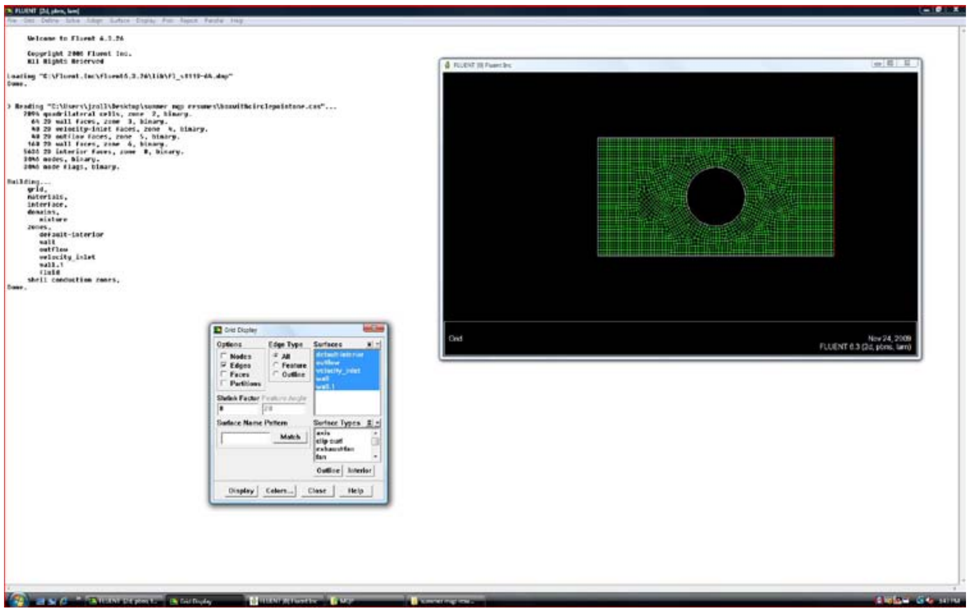


Figure 1: Gambit 2.4.6 General User Interface

FLUENT (*Figure 2)* is a "Flow Modeling Software" owned by and distributed by ANSYS, Inc. It is used to model fluid flow within a defined geometry using the principles of computational fluid dynamics. Unlike GAMBIT, which it is shipped with, it utilizes a multi window pane system for displaying various configuration menus and grids instead of a single window with several embedded sub-windows restricted within the space of the parent window. FLUENT is able to read geometries generated in GAMBIT and model fluid flow within them. It can model various scenarios using computational fluid dynamics, including compressible and incompressible flow, multiphase flow, combustion, and heat transfer.

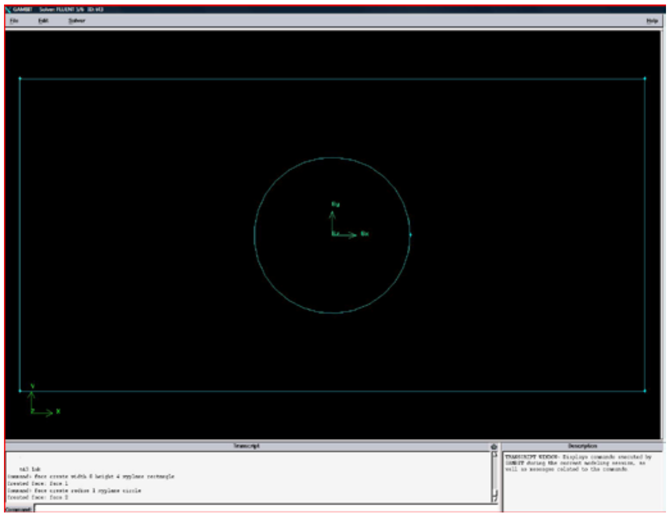


```
     Welcome to Fluent 6.3.26

     Copyright 2006 Fluent Inc.
     All Rights Reserved

Loading "C:\Fluent.Inc\fluent6.3.26\lib\fl_s1119-64.dmp"
Done.

> Reading "C:\Users\jzoll\Desktop\summer mgp resumes\boxwithcirclepointone.cas"...
    2096 quadrilateral cells, zone  2, binary.
     64 2D wall faces, zone  3, binary.
     48 2D velocity-inlet faces, zone  4, binary.
     48 2D outflow faces, zone  5, binary.
    160 2D wall faces, zone  6, binary.
   5456 2D interior faces, zone  8, binary.
   3096 nodes, binary.
   3096 node flags, binary.

Building...
     grid,
     materials,
     interface,
     domains,
        mixture
     zones,
        default-interior
        wall
        outflow
        velocity_inlet
        wall.1
        fluid
     shell conduction zones,
Done.
```

Figure 2: Fluent 6.3.26 General User Interface

2.3 Generating a Simple 2-dimensional Model with a Single Circle in the Center

In order to generate the model to be simulated, first one must open up GAMBIT. The geometry can either be imported from another source or built within the program. In this test the geometry was created within GAMBIT. I then selected solver and choose Fluent 5/6. This configures the program to generate a file that will be compatible with the version of FLUENT being used. Next I went to Geometry, select Face, and then select Create Face. Since the geometry is rather simple, composed of a circle within a rectangle, there was no need to create



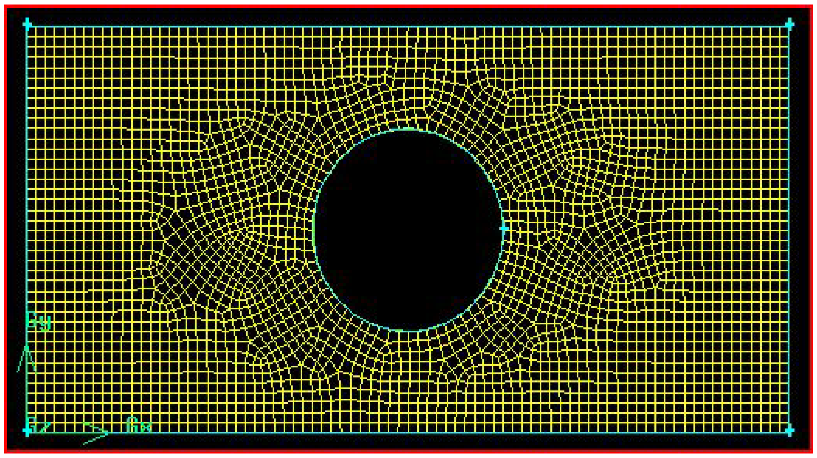
```
Command> face create width 8 height 4 xyplane rectangle
Created face: face.1
Command> face create radius 1 xyplane circle
Created face: face.2
```

TRANSCRIPT WINDOW: Displays commands executed by GAMBIT during the current modeling session, as well as messages related to the commands.

Fiaure 3: Rectanale and Circle Generated at Oriain

the vertexes individually. First, I created a rectangle with an x value of 8 and a y value of 4 and a circle with a radius of 1. Both objects will be created with their center being the origin *(Figure 3)*. In order to place the objects fully within the first quadrant I used the move command. Since they are halfway in both the positive x and y axis they must be moved 4 in the x direction and 2 in the y. Next the two geometries need to be consolidated into one. This was done by using the subtract command where one must select the rectangle and subtract the circle from it. After the operation is completed only one face appeared up in the menu.

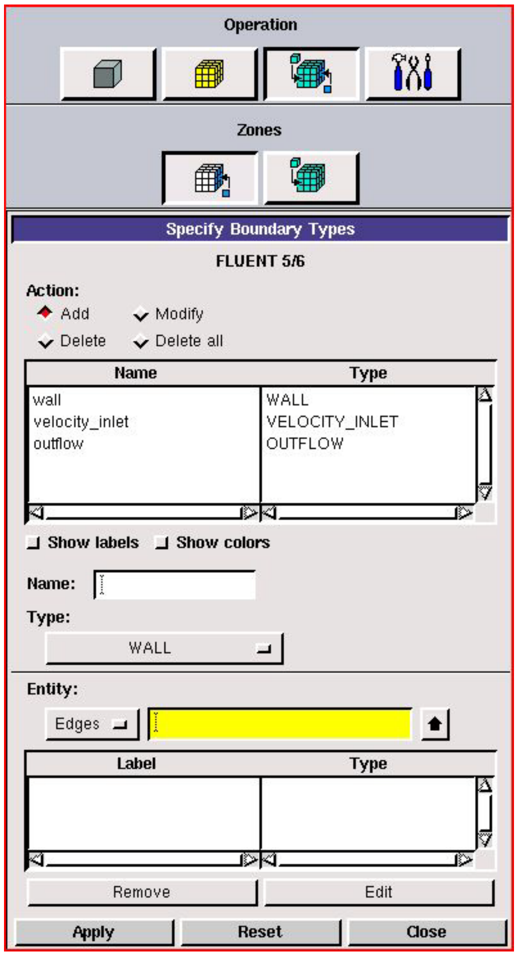
With the geometry created, it was now time to generate a mesh. I selected mesh, then face, and finally mesh faces. Leave the defaults except for the spacing. Spacing determines how far node points are away from each other and consequently how many are created. The spacing was done in the same units as the geometry used. At the default spacing of 1, a single node is created for every unit of 1. For a side that measures at 4 there would be 4 node points. I inputted the desired spacing to get the optimal resolution. For the



Fiaure 4: Mesh aenerated at 0.01 resolution

first preliminary tests spacing of 0.5, 0.25, and 0.1 were used, but later tests used 0.1, 0.05, and 0.01. After the mesh was exported I went back and replace the spacing for the mesh with a different one if the resolution isn't accurate enough. When the mesh was generated it looked like a grid that changes shape as it becomes closer to the circle *(Figure 4)*.

Next the boundary types needed to be defined. Not every wall of the geometry serves the same purpose, so it was important to determine how Fluent was going to interpret them. For example in this geometry water needs to enter from the left and exit through the right while going around the circle in the middle. On its own Fluent can't determine that is what the user wants, so at this point, inlets, outflows, and walls need to be defined. In order to do this, I selected Zones then Boundary Types *(Figure 5)*. From the drop down menu I selected the left edge. This can be determined by selecting an edge and clicking the arrow that points to the right which will move the label to the right options list. The edge will be highlighted on the geometry. When the correct edge was selected, I went to the Entity drop down menu and selected velocity_inlet. I called the edge velocity_inlet to label it for further reference. I repeated the steps for the right edge which was the outflow. The two remaining walls of the rectangle were designated and labeled as wall. Lastly the inner circle should be defined as a wall, but it was labeled circle so it could be accessed separately from the other walls.



Action: [✓ Add] Modify Delete Delete all

Name	Type
wall	WALL
velocity_inlet	VELOCITY_INLET
outflow	OUTFLOW

Name:

Type: WALL

Entity: Edges

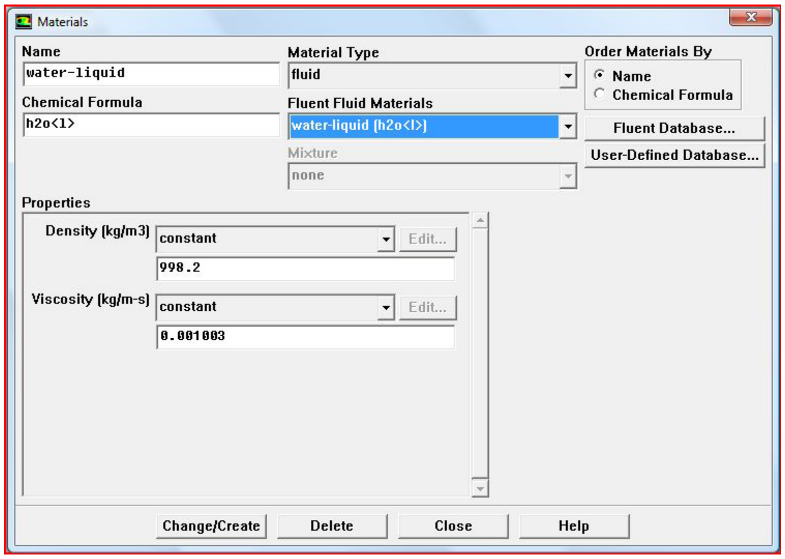
Label	Type

Figure 5: Boundary Types Menu

This was important when reading the forces acting on different elements of the geometry. Now that the mesh has been generated and the boundary types defined, I saved the file and exported as a mesh with the 2D option selected as the model is only utilizing the x and y dimensions.

The file was then opened in Fluent. It presented a list of options, 2d, 2ddp, 3d, and 3ddp. I selected 2d since the geometry generated in GAMBIT was 2-dimensional. Next I selected File then Read and then Case in order to import the file from GAMBIT, which ended with a .msh file extension. Before doing anything else in Fluent I checked that there were no errors in the geometry. This was done by selecting Grid then Check. Although it was not essential to do this step, doing so will prevent one from running a simulation on faulty geometry, which, considering the nature of how the program uses memory, may cause the program to lock up and the computer to run rather slow as it prints out a series of error messages. Please note that this did not catch all possible mistakes. In one test I accidentally labeled the inside circle as the wall where the fluid outflows. In this case it did not notify me of the mistake as the program will assume that was intended.

I preceded by selecting Display the Grid. A new configuration window asking for criteria to be determined opened up but the defaults were all that was needed so I just selected Display. This opened up a new window displaying the model created in GAMBIT. From this point the fluid


Materials

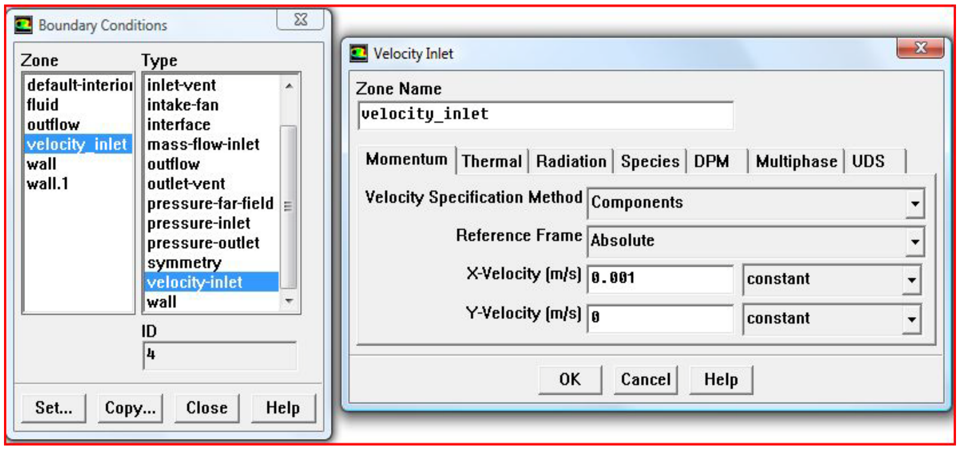
Name: water-liquid
Chemical Formula: h2o<l>
Material Type: fluid
Fluent Fluid Materials: water-liquid (h2o<l>)
Mixture: none
Order Materials By: [✓ Name] Chemical Formula

Properties
Density (kg/m3): constant
998.2
Viscosity (kg/m-s): constant
0.001003

Figure 6: Materials Window

needed to be defined. This is done by opening up the Materials window *(Figure 6)* which is located in the Define menu. By default Air is listed, but in this test water was used instead. I added water by clicking on the Database button. Another configuration window opened up, which listed various materials. I scrolled down to the bottom and selected Water. There were two entries for water so of the two I picked the one indicating liquid instead of vapor. After clicking Copy and water appeared in the main materials window. In some of the tests the viscosity of Water was changed from the default. For these tests the viscosity value was changed by typing in a new value and clicking Change/Create.

After defining the materials the boundary conditions needed to be defined. I opened the menu by clicking Define and then Boundary Conditions. Then, I selected fluid in the Zone list and then fluid in the Type list before pressing the Set... button on the bottom. In the drop down menu that says 'air', I selected it and changed it to water. This tells Fluent that it will use water as the fluid for the simulation. I then pressed okay and exited out of the subwindow. At this point I returned to the Materials window and deleted air from the list of available materials so that there won't be any confusion, but this was not necessary. I went back to the Boundary Conditions window *(Figure 7)* and selected the item velocity_inlet in both panes and pressed set... again. For



Zone: default-interior, fluid, outflow, velocity_inlet, wall, wall.1

Type: inlet-vent, intake-fan, interface, mass-flow-inlet, outflow, outlet-vent, pressure-far-field, pressure-inlet, pressure-outlet, symmetry, velocity-inlet, wall

ID: 4

Zone Name: velocity_inlet

Velocity Specification Method: Components

Reference Frame: Absolute

X-Velocity (m/s): 0.001 constant

Y-Velocity (m/s): 0 constant

Figure 7: Boundary Conditions and Velocity Inlet Windows

the Velocity Specification Method, I changed from the default option in the drop down menu to Components then changed the X-Velocity to 0.001 as that value was be used in this test. Then I pressed OK and exited out of the Boundary Conditions window.

At this point the solution needed to be initialized. To do this I went into the Solve menu, pressed Initialize and then Initialize... which opened up a window titled Solution Initialization. In the new window I clicked on the drop down menu and selected velocity_inlet as where it will compute from. For the X Velocity I inputted the same number used before which was 0.001 m/s. I then clicked Init and closed the window.

At this point all conditions were satisfied to run the simulation. From the Solve menu I clicked Monitors and then Residual. This window set the parameters of the simulation. For this test the default options were left alone. I check the radio button next to the Plot option then pressed OK. In order to run the simulation I clicked Solve then Iterate to open the Iterate window. For number of iterations I typed 1000 and then pressed Iterate. The second window that displayed the geometry was replaced with a plot with new points being added as time went. The number of iterations were also be tracked in the main window. Depending on the resolution running the solution varied in terms of length

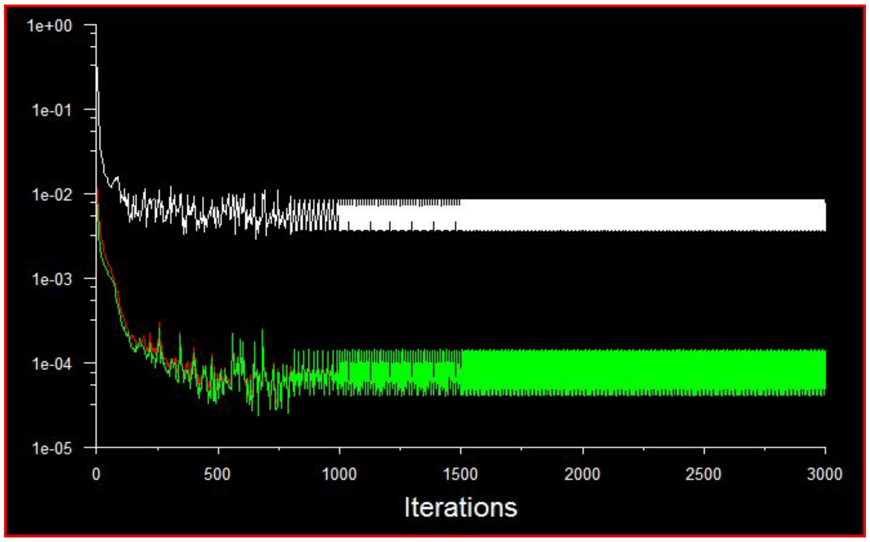


Figure 8: Sample of 3000 Iterations

(Figure 8). In a few circumstances the simulation may ended before it could finish all 1000

iterations. This meant the solution had converged and the main window indicated that convergence had been found. In some tests it stopped computing the solution before convergence was found because the computer ran out of memory to run the operation. In other tests the solution did not converge after 1000, which prompted me to go back and run further iterations to see if it converged with more. In the case that they still did not converge, I compared the earlier solution with the one generated after further iterations. After I compared the two, I determined whether or not they are close enough to pick a solution.

Since the simulation completed, it was necessary to interpret the results. I did this by clicking Display, then Contours to open up the Contours configuration window. This displayed the results of the simulation in contours over the geometry based on the defined parameters that were being measured. I checked the Filled radio button and then switch the options in the drop down menus to say Pressure. Clicking Display changed the second plot window into a contour graph overlaid on top of the geometry. I then checked whether or not the distribution of pressure forces makes sense using prior knowledge of fluid flow, using the color key on the right to determine what color means what value. Red represented a higher pressure while blue indicated low pressure. To see what the actual forces are on specific parts of the geometry, I clicked Report and then Forces. Under Wall Zone I selected the entries for Wall and Circle as those are the objects that were being measured in this test. The entry for Force Vector indicated the direction of the measurement, meaning value of 1 for X and 0 for Y measured forces in just the x direction. By switching the values and it measured in the y direction. Since the fluid flow was going horizontally there were minimal forces in the y direction. I checked the forces in the y direction to verify that was indeed the case. Pressing Print displayed the pressure, viscous, and total forces for each zone along with the corresponding coefficients.

2.4 Generating a Simple 3-dimensional Model with a Single Sphere in the Center

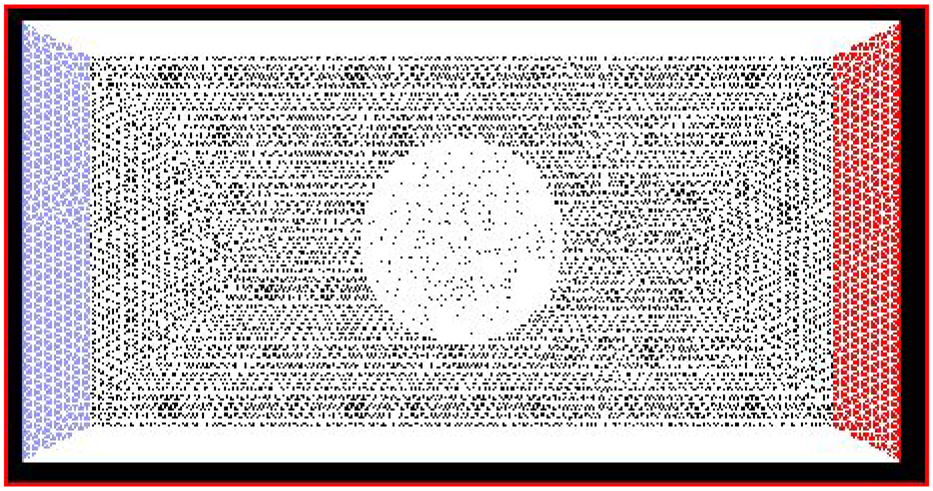


Figure 9: 3D Model with a Single Sphere

Since the previous test set out to test accuracy by running the simulation at multiple resolutions, it was only logical to extend the simulation to the third dimension. Starting out the steps were generally the same, but keeping the third dimension in mind. Instead of creating a rectangle, I went to the volume subsection and generated a rectangular prism instead, with the dimensions of 8x4x4. I substituted the circle with a sphere with a radius of 1 and moved it to the location of (4,2,2) , which should be the exact center of the geometry *(Figure 9)*. Like before, the sphere was subtracted from the rectangular prism. Meshes were generated at three resolutions once again under the volume subsection. Using knowledge gained from the previous test, this simulation was run at only the 0.1, 0.05, and 0.01 resolutions since the 0.5 and 0.25 resolutions were too low to accurately model the fluid flow. Defining the boundary types was similar to before except I needed to include the other sides of the rectangular prism as part of the wall. The only other detail I had to keep in mind for GAMBIT was to not check off the option to export in

2D. There were no further differences in Fluent except for selecting the 3D option when opening up the program for obvious reasons.

2.5 Flow Through a 2 Dimensional Pipe With Seven Circles

Fluids often face obstructions as they flow in a direction. The previous test demonstrated what would happen when a singular object blocked part of the fluid's path in a somewhat idealized scenario with even spacing on both sides. Even at different resolutions it was reasonable to say the flow would not noticeably change. But often in reality there are often more than one object obstructing a fluid's flow. This test was designed to model the flow around seven randomly placed circles and compare them with three different configurations of order.

The general methodology for measuring flow two dimensions with seven circles inserted at random was very similar to the test with a single circle. Namely, the big difference is the number of circles and the size of each individual circle. Back at previous test, I followed the same procedure until the moment I needed to generate the circle at a radius of 1. Instead of that circle I generated seven circles at the sizes 0.25, 0.50, 0.75, 1.00, 0.75, 0.50, 0.25 respectively *(Figure 10)*. The order of the circles in terms of size and creation was not particularly important, but chosen for identification purposes so that there would be no confusion between the same size circles being numbered so close together.

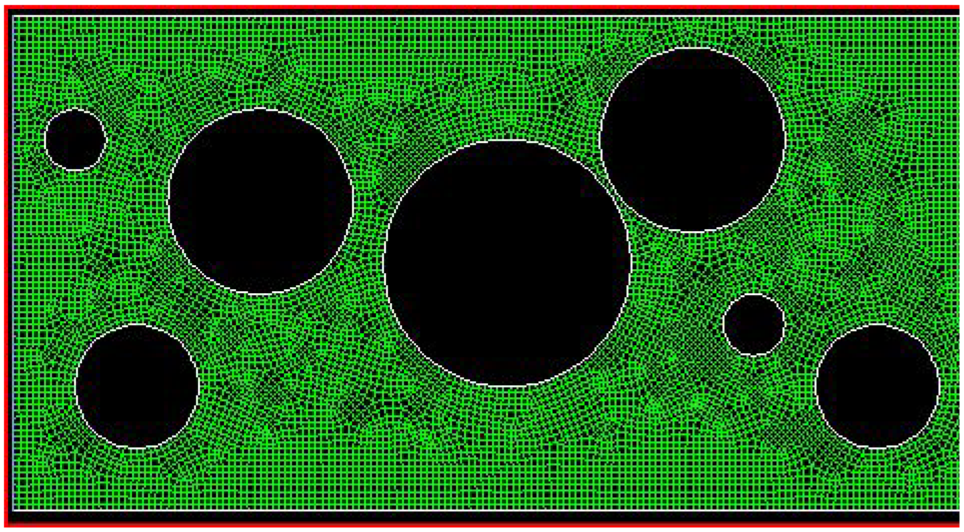


Figure 10: A 2 Dimensional Pipe wth Seven Circles

From here the circles were moved inside the rectangular boundary to seemingly random positions for each of the three variations. It should be noted that in the first attempt to run this test, the first two variations exported without any difficulty, but GAMBIT had difficulty dealing with the third. This was because one of the circles touched the wall and therefore shared a boundary. The solution to this problem was found by moving the circle that was creating difficulty away from the wall.

Again the same steps were taken as the first test up until the boundary types were defined. In the same order as their creation, I labeled the circles separately as numbers 1,2,3,4,5,6, and 7. After importing the geometry into Fluent, one thing that stood out was that it renamed the circles as zone-(number) with number being the same number I selected from before. This was not a big deal in this circumstance as they maintained the order that was desired. Further steps were identical to the previous test, using the same numbers for viscosity, density, and velocity.

2.6 Flow Through a 3 Dimensional Pipe With Three Spheres

As stated before, fluids do not simply move around a single object and continue unhindered. The same can be said for fluids simply moving in two dimensions. Fluid flow in two dimensions only exists in drawings and theoretical models. If one wanted to even hope of accurately modeling a fluid flow they need to make the shift to the third dimension. This test is similar to the previous 3d accuracy test the same way the previous test was similar to the 2d accuracy test. To clarify, in this test three spheres are placed at random throughout the rectangular prism boundary with three variations. Because the computer was unable to model the fluid flow in three dimensions at .01 and .05 resolutions before running out of memory, a resolution of 0.1 was chosen to be used for this test.

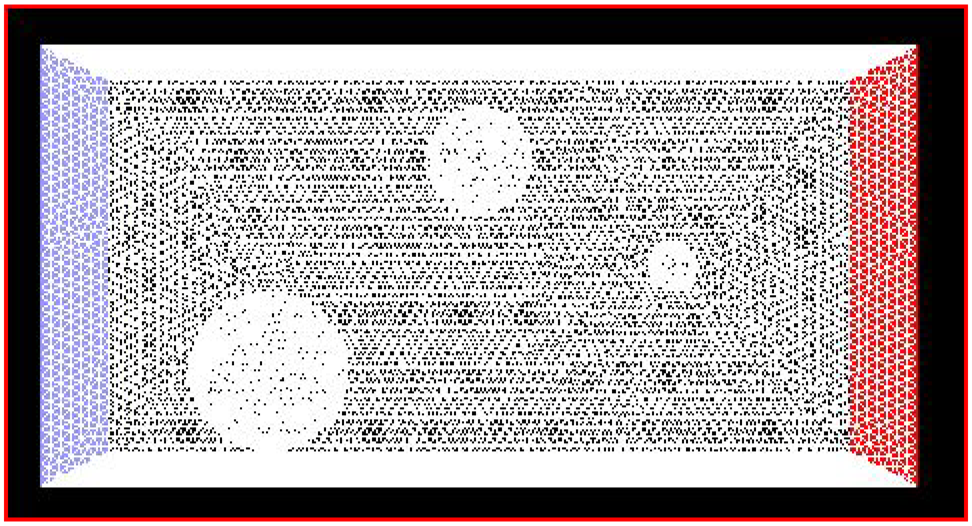


Figure 11: A 3 Dimensional Pipe with Three Spheres: First Variation

The same steps from the 3 dimensional test were repeated for this simulation up until the creation of the spheres. Instead of the sphere with a radius of 1, three spheres were generated with radii of 0.25, 0.50, and 0.75 *(Figure 11)*. These were moved to random locations in the x, y, and z axis for each variation. Like in the simulation with seven circles, each sphere was later on labeled 1, 2, and 3 for their respective order. Again, Fluent renamed the circles, but since they retained the numbers it was not an issue. From that point on, the rest of the steps were identical to the previous tests.

3.0 Results

3.1 Accuracy Testing in 2D

To determine whether or not the later tests were indeed accurate, a test of accuracy was developed. This involved a simple two dimensional flow within a cylinder with a circle placed in the middle. First it was run to see if any forces acted on the circle. Since a fluid flow entered from the velocity inlet in the x direction and by the circle on the way to the outflow, a force should be present on the circle in the x direction. While true for the x direction this should not the case for the y direction as there is no net pressure force acting in the y direction. While some force was registered in the test in the y direction in addition to the x direction, it was found to be sufficiently small to be considered an aberration.

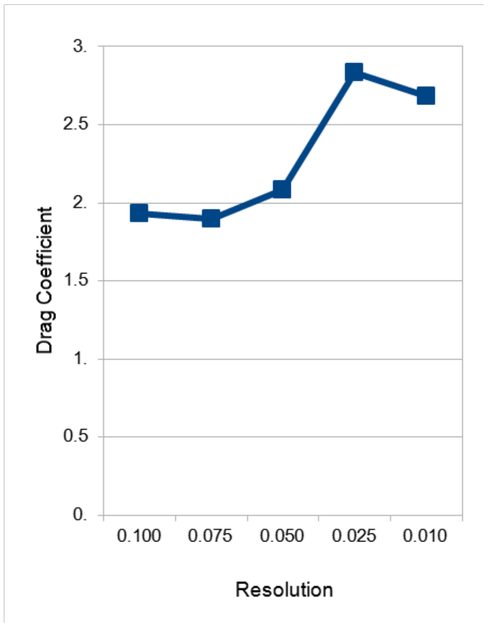


Figure 12: Graph of change in drag coefficient for 0.001003 viscosity

Another test of accuracy involved running the same simulation at different resolutions. This involved changing the spacing of the node points in GAMBIT for the creation of the mesh. The smaller the spacing between node points, the greater the resolution, which means greater accuracy of the model. Ideally one would want the highest resolution possible, but with a higher resolution it takes more processing power to compute the simulation. Running a simulation at a low resolution may take seconds while at a higher resolution can take hours for the same number of iterations. To find a good balance between speed and accuracy, this test was run at various resolutions. At lower resolutions there is a degree of noise and variation in what is computed, but as the resolution gets higher the variation between the calculated values become smaller. Once the variation between resolutions is diminished to a point it can be considered negligible, an ideal resolution is found.

The first test was performed at the .5, .25. and .1 resolutions. These tests acted as a preliminary because at the lower resolutions the geometry is still rather blocky. This was most noticeable in the center circle where the sides and edges are noticeable. At higher resolutions the corners and sides are less visible because of the higher degree of node points. These were computed with a velocity of 0.001 and a viscosity of 0.001003. This resulted in a Reynold's number of around 2000. The drag coefficient is also around 4 for the two lower resolutions. It is cut in half for the .1 resolution. It is also worth noting that the lower resolutions converged after less than 50 iterations.

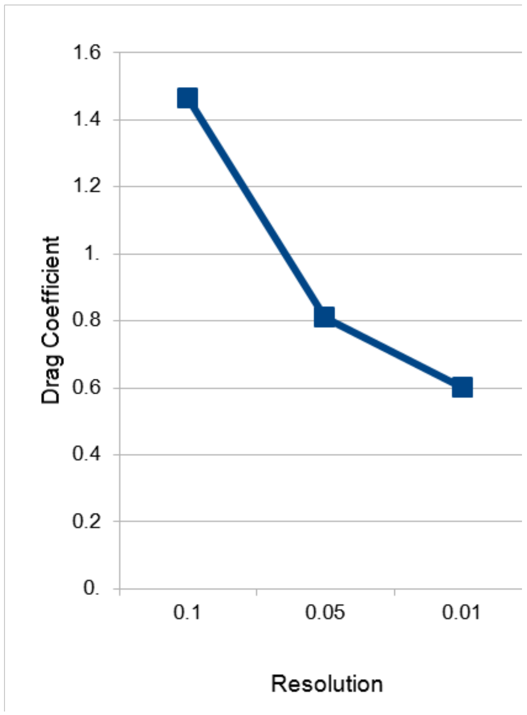


Figure 13: Graph of change in drag coefficient at low viscosity

The same test was then run at 0.05 and 0.01 resolutions. It was significantly more difficult to run FLUENT at these resolutions because of the number of calculations it needed to process. With the grid displayed, any window movement would cause the grid to refresh, locking all operations until it finished drawing the window. Since the trend from the last three tests indicated a downward slope as the resolution increased the drag coefficient decreased, it was expected the next two would follow this trend and settle with a drag coefficient of around 1 as that is the ideal drag coefficient of a cylinder. That did not happen. At 0.05 resolution the resolution jumped to a little above 2 while at 0.01 is went up to 2.68 *(Figure 12)*. While the first one could be considered noise if the actual drag coefficient was indeed close to 2, the second coefficient was too much of a jump upwards to be discounted by the imperfections of the program. Since the Reynold's number was rather large, it was determined to be a likely cause of the bump in drag coefficient.

Reynold's number is determined by velocity/(u), density/(ρ) , diameter/(L), and viscosity/ (μ) for flow through a pipe. It can be expressed as Re= $\rho\ u\ L\ /\ \mu$. To see what factor could cause this a single variable was changed. The viscosity was lowered to 0.00001 and raised 0.01. At the lower viscosity the Reynold's number went up to 199, 640 and at a higher viscosity it dropped to 199.64. Since at this point it could be determined that both the 0.5 and 0.25 resolutions were too low to accurately determine flow, they were not used in this simulation. Only the 0.1, 0.05, and 0.01 resolutions were used. With a lower viscosity the drag coefficient dropped to 1.46, 0.81, and 0.6 in order of lowest resolution to highest *(Figure 13)*. This seemed to correlate to the pattern shown at the lower resolutions in which the drag coefficient dropped as the resolution became larger. In addition the drop between the resolutions appeared to become smaller as if it were to converge at a single number if higher resolutions were used. Of course since only 3 resolutions were used in this example the trend cannot be considered

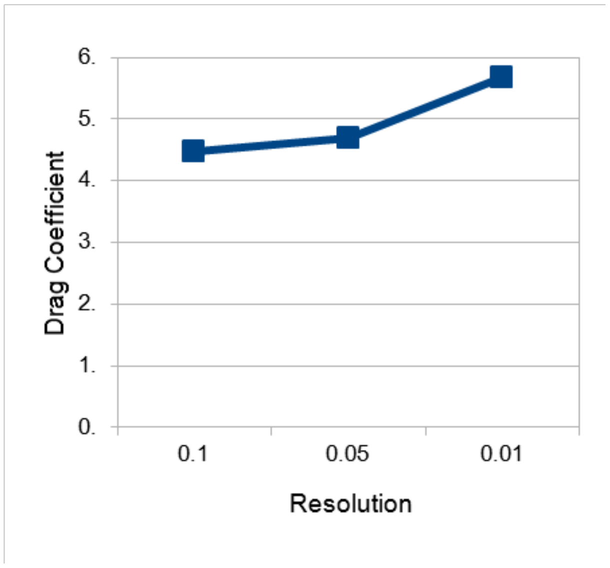


Figure 14: Graph of change in drag coefficient for high viscosity

significant due to small sample size. In addition the problem before was initially hypothesized to be due to an unusually high Reynold's Number. By lowering the viscosity of the fluid it only raised the Reynold's Number even further. In order to lower the Reynold's number according to the equation, the viscosity needed to be raised. Unfortunately raising the viscosity did not achieve the desired result either. While the Reynold's number appeared more manageable, the drag coefficient shot up to 4.48, 4.68, and 5.67 in order of lowest resolution to highest *(Figure 14)*. Again there appears to be an undesired upward slope with the last jump being greater than the difference between the first two. It should be noted that an upward slope is not 'bad' by definition but indicates a trend going away from the desired drag coefficient of 1. One should also recognize that the 0.01 resolution only took 26 iterations before it converged while the other two resolutions did not converge after 1000 iterations. This appeared to be peculiar as it would seem that a greater resolution would require more iterations in order to calculate a convergence point.

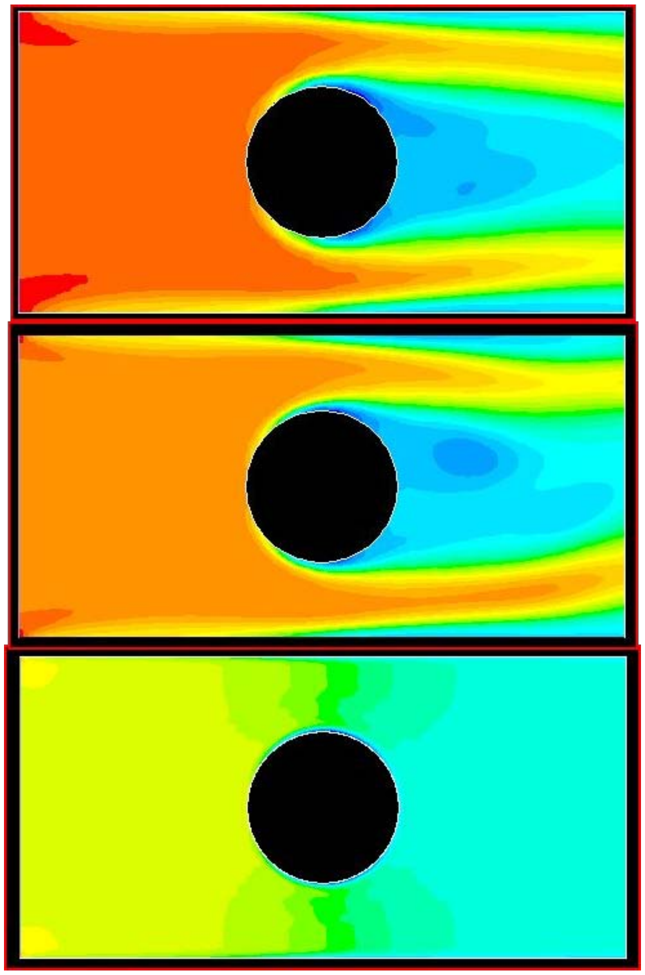


Figure 15: Variations in Pressure at .1, .05, and .01 resolutions at high viscosity

Comparing the same tests across resolutions based on contour graphs show a distinct similarity despite the resolutions. Generally it seemed that the overall shape remained the same for the three resolutions for both total pressure and velocity. This was most obvious when observing the 'tails' behind the circle, which were spots with low pressure and velocity *(Figure 15)*. Each viscosity had its own shape of tail. This was important to observe since it indicated that the shape of the contours were determined by the conditions of the test, such as viscosity, instead of the degree of resolution.

The major differences were that at the lower resolutions there seemed to be a more dramatic contrast between colors representing pressure ranges. At low resolutions dark red, which indicates high pressure or velocity, was much more present than when looking at the high resolutions. The general shape of the flow was also more asymmetric when slicing the geometry in half horizontally while it seems rather symmetric at the .01 resolution. There seemed to have been more of a blend of colors at the higher resolutions along with smoother shapes. This is most likely attributed to greater number of data points which allow for greater precision in calculation and certain roundness in shape. In most cases it appeared that the .05 resolution was a middle ground between the other two resolutions. One can observe the gradual change in colors and shapes from low to high resolution with the middle resolution acting as an intermediary step. Exceptions did exist. At the high resolution it was sometimes difficult to discern the original shape as the contrast between the colors had dulled so much. This was especially noticeable in high viscosity tests at 0.01 resolution.

3.2 Accuracy Testing in 3D

Overall the simulations that were run in 3D did not run as smoothly as the ones performed in 2 dimensions. Only one of the three prepared tests was completed, the simulations at 0.1 resolution. This was the lowest of the three resolutions, meaning the geometry required fewer nodes than the other two. GAMBIT was able to successfully export all three resolutions as a mesh so Fluent could read them. The process of building the mesh took a considerable amount of time compared to the 2D version from the last test. This was due to the fact that a third dimension was added, giving the mesh further complexity.

In Fluent everything proceeded smoothly until it was time to iterate. Iterations were able to be started but after a few passes the program alerted me that it was out of memory. This occurred in the 0.05 and 0.01 resolutions, both of which were held in files of considerable size, which seemed to verify that it was caused by a lack of random-access memory. The occurrence seemed rather puzzling, considering that the computer that the simulation ran on had considerable ram that would appear to be more than enough RAM. A possible reason behind this would be that the installed version of the program was 32 bit and was only able to see the max random-access limit of three gigabytes for 32 bit programs despite the fact that the system ran at 64 bits. Presumably this issue would go away if one were to install the 64 bit version of Fluent and ran the same simulation.

Because of the program's inability to properly simulate the simple model in three dimensions at resolutions higher than 0.1, only this example was recorded. For further tests I decided that it would not be a good idea to try and run the simulation of more complex geometries at such a high resolution. All following tests in three dimensions were run at the resolution of 0.1 as that was shown to work.

3.3 Flow through a 2 dimensional pipe with 7 circles

This test differed from the previous two in that not only that there were seven different size circles instead of one but that it focused on the variation of flow based on circle distribution than the effects of resolution against viscosity. The three contour graphs for each variation were quite similar. On the left there is a region of high pressure before the fluid reaches the circles. For the rest of the geometry the pressure is much lower in the yellow and blues except for the

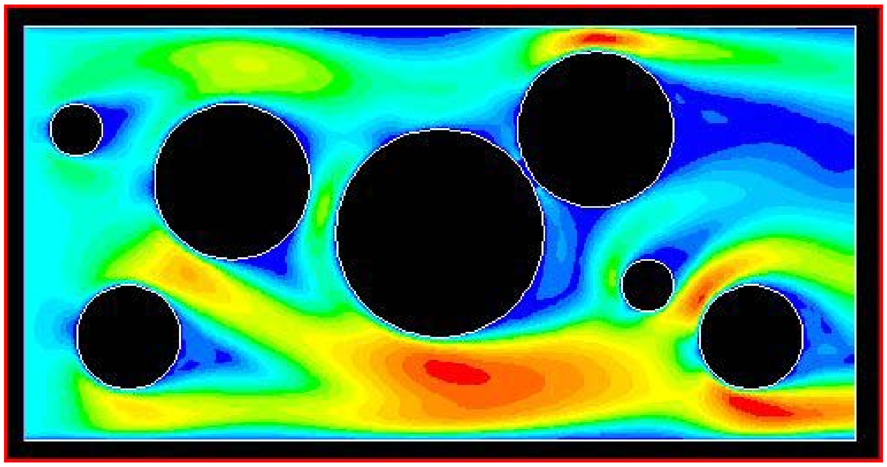


Figure 16: Contours through a 2 dimensional pipe with 7 circles

streams in between the circles where the fluid is able to travel the fastest. This is reflected in the velocity contours where the whole thing is turquoise except for the streams which can range from yellow to red. The tail areas in both are a deeper blue. This indicates a stagnation in terms of fluid flow behind the circles since in each variation there were multiple circles some of the circles are blocked from the flow of the fluid.

3.4 Flow through a 3 dimensional pipe with 3 Spheres

Since the test was designed to be rather similar to the previous one; it was no surprise that the contour graphs appeared similar as well. The pressure was highest on the sides of the spheres in the face of the water flow. Similarly behind each sphere it generated a tail of stagnant fluid flow. The changes in pressure and velocity were not as dramatic in this test due to the distribution of the circles in the three dimensions. Unlike the previous experiment where every circle's front face was at least partially blocks by the one placed left of it in the x dimension, the spheres had different z dimensions which allowed the fluid to flow mostly unobstructed as it hit each sphere.

4.0 Conclusion

Computational Fluid Dynamics for all its advances over the past few decades is still nothing but an approximation and these tests seemed to only reinforce the notion. As the resolution changes there drag coefficient and overall model changes. Even at a high .01 resolution the program didn't seem to have settled on a concrete value and one would have reason to believe that further tests at even higher resolutions would show a change in the model. Furthermore, due to limitations of computer hardware the higher levels of resolution cannot be computed without running out of memory. This is especially true considering the simulations run in this experiment can be considered relatively simple compared to modeling of real life applications and scenarios. The circumstances may be different if the tests were run in a completely 64 bit environment, but that was not available for use at the time of conducting the experiment.

If given more time and materials the next logical step would be to devise an experiment with conditions that can be replicated in both the program and in real life. This would be done using a wind tunnel or water tank where the fluid flow can be measured along with the forces. The same conditions would be created in Gambit and simulated in Fluent. In addition, this would require the 64 bit installation of Fluent on a computer with more than three gigabytes of random access memory. Only then would one be able to get a good grasp of how accurate the program is at running simulations.

Eventually it would seem that computational fluid dynamics would advance to the point that nobody would ever need to conduct actual simulations such as running a wind tunnel.

Unfortunately CFD is not yet at that point. For any real world applications it would still be best to at least run a wind tunnel concurrently with any computational fluid dynamics.

References

"ANSYS FLUENT Flow Modeling Software." *Welcome to ANSYS, Inc. - Corporate Homepage*. Web. 4 Sept. 2009. <http://www.ansys.com/products/fluid-dynamics/fluent/>.

"A Brief History of Computational Fluid Dynamics (CFD) from Fluent." *CFD Flow Modeling Software & Solutions from Fluent*. Web. 18 Oct. 2009. <http://www.fluent.com/about/cfdhistory.htm>.

"Reynolds Number." *Engineering ToolBox*. Web. 9 Nov. 2009. <http://www.engineeringtoolbox.com/reynolds-number-d_237.html>.

"Reynolds Number." *NASA - Title...* Web. 03 Dec. 2009. <http://www.grc.nasa.gov/WWW/BGH/reynolds.html>.

Appendix

A1 Accuracy Testing

A1.1 Low Resolution Test

A1.2 Accuracy Testing at Resolutions of .1, .05, and .01 at Low Viscosity

A1.3 Accuracy Testing at Resolutions of .1, .075 .05, .025 and .01 at Normal Viscosity

A1.4 Accuracy Testing at Resolutions of .1, .05, and .01 at High Viscosity

A2 Modeling Flow with Multiple Obstructions

A2.1 Modeling Fluid flow in 2D with Seven Circles at 0.01 Resolution

A2.2 Modeling Fluid flow in 3D with Three Spheres at 0.1 Resolution

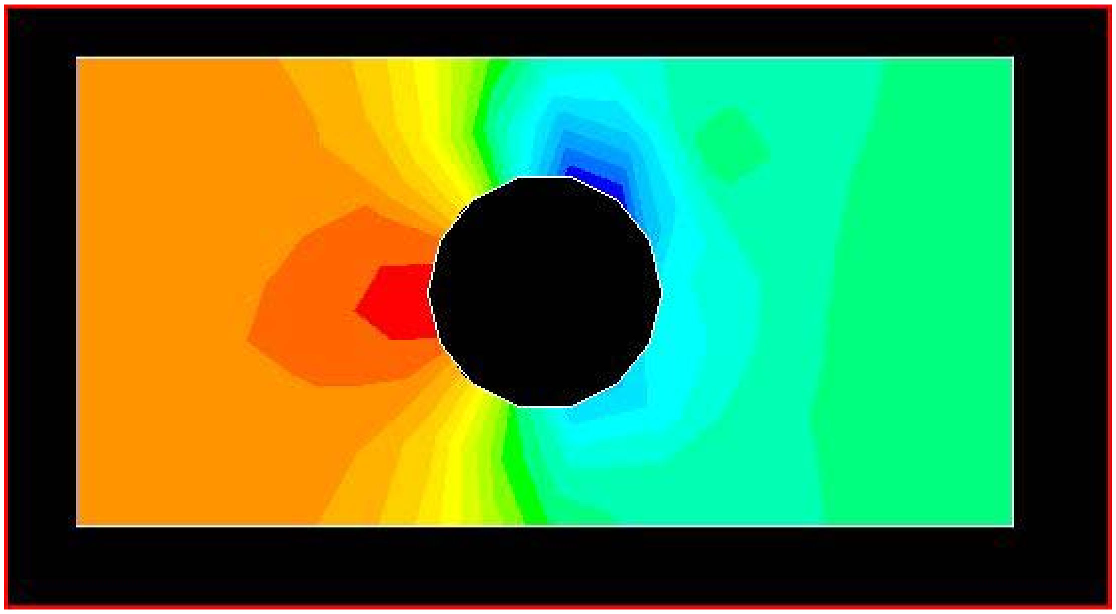
A1 Accuracy Testing

A1.1 Low Resolution Test

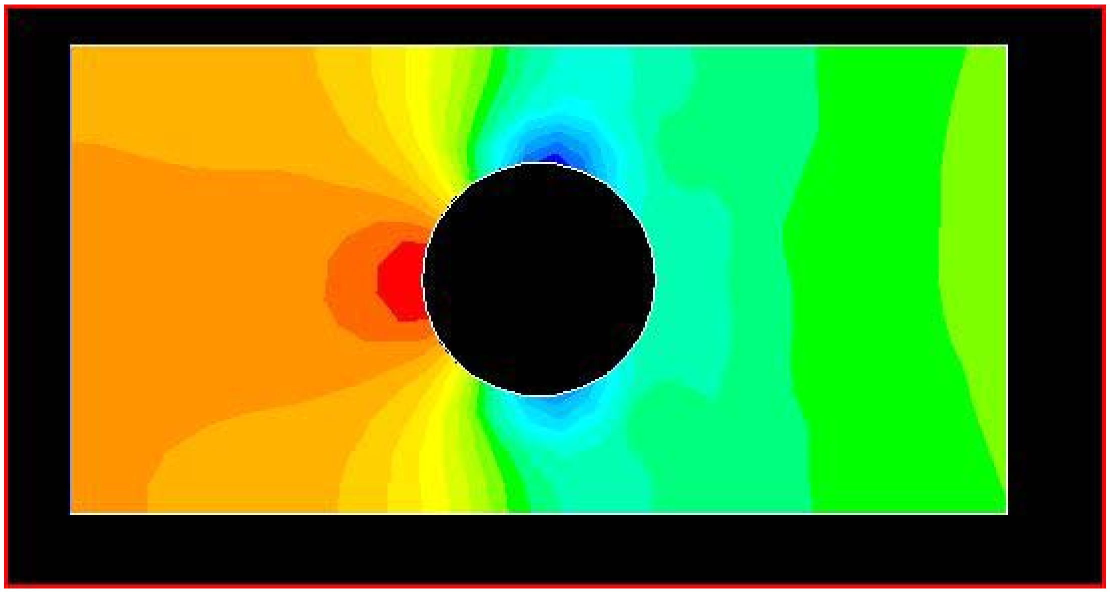
Accuracy Testing at Resolutions of .5, .25, and .1

Resolution	Density	Viscosity	Drag Force	Velocity	Drag Coef	Reynolds
0.5	998.2	0 . 00 103	0 . 00 385	0 . 00 1	3.86	1990.43
0.25	998.2	0 . 00 103	0 . 00 309	0 . 00 1	3.09557203	1990.43
0.1	998.2	0 . 00 103	0 . 00 193	0 . 00 1	1.93	1990.43

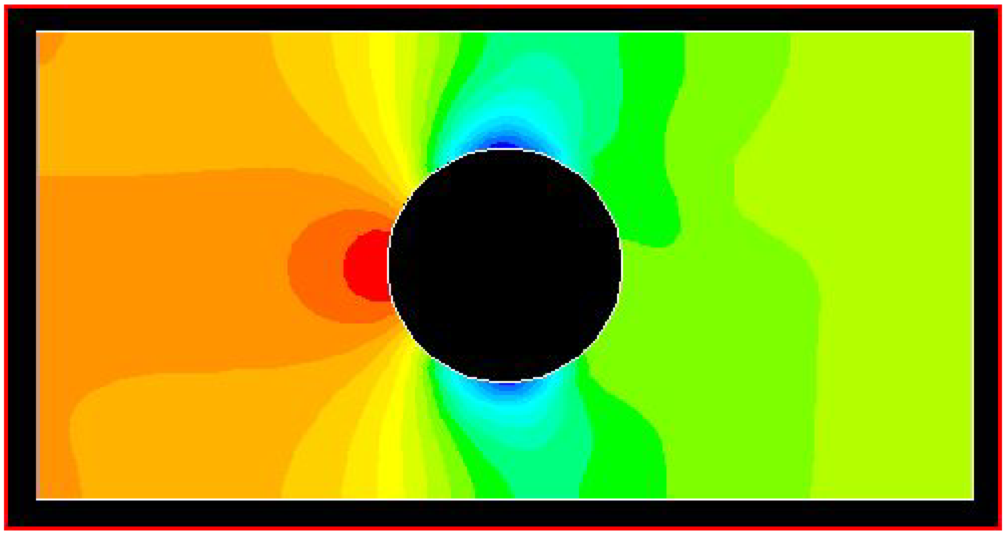
Pressure at .5 Resolution



Resolution



Pressure at .1 Resolution

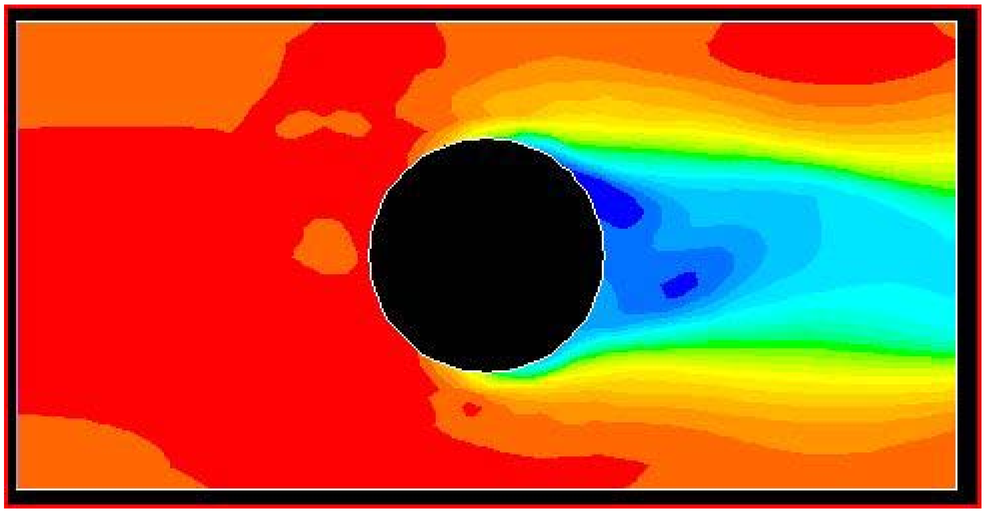


A1.2 Accuracy Testing at Resolutions of .1, .05, and .01 at Low Viscosity

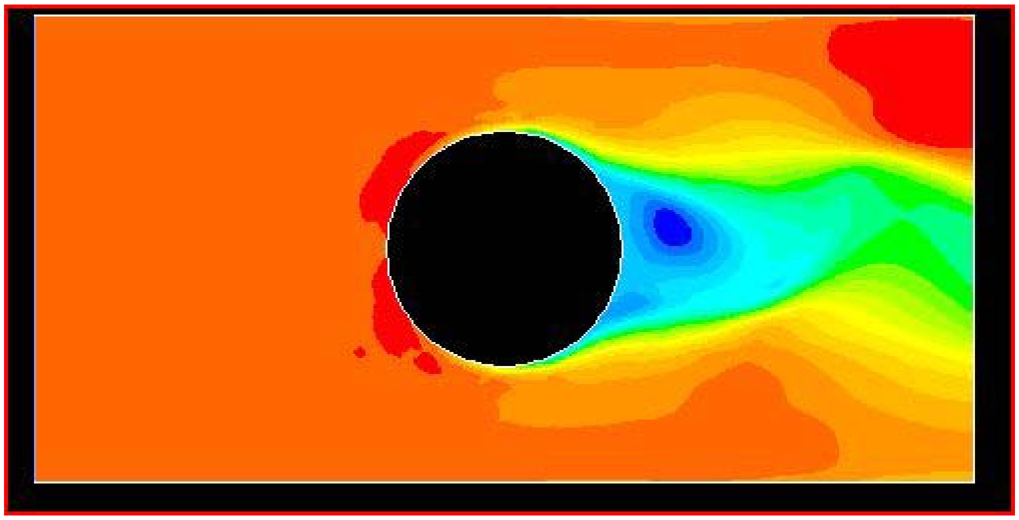
Accuracy Testing at Resolutions of .1, .05, and .01 at low viscosity

Resolution	Density	Viscosity	Drag Force	Velocity	Drag Coef	Reynolds
0.1	998.2	0 . 00 00 1	0 . 00 1459	0 . 00 1	1.46	199640
0.05	998.2	0 . 00 00 1	0 . 000 8080	0 . 00 1	0.81	199640
0.01	998.2	0 . 00 00 1	0. 000 5974	0 . 00 1	0.6	199640

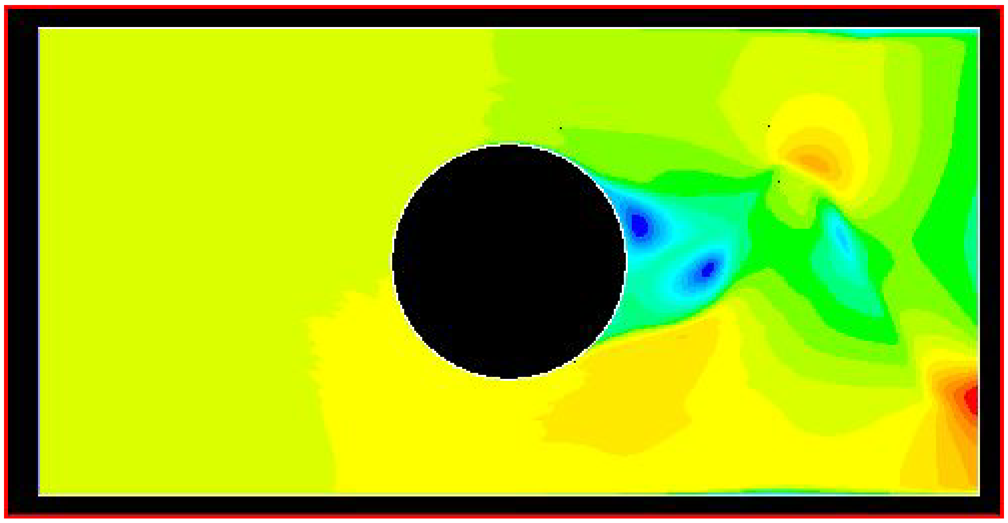
Pressure at .1 Resolution



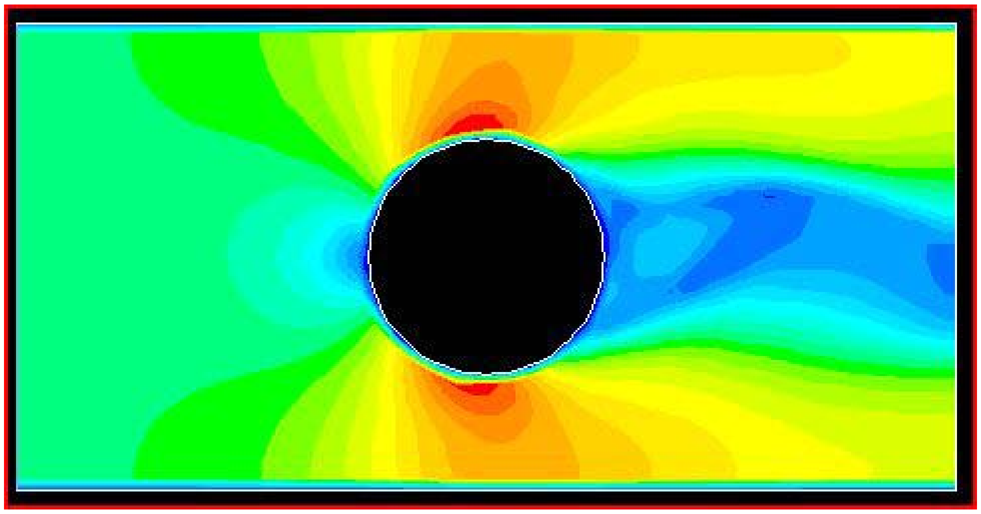
Pressure at .05 Resolution



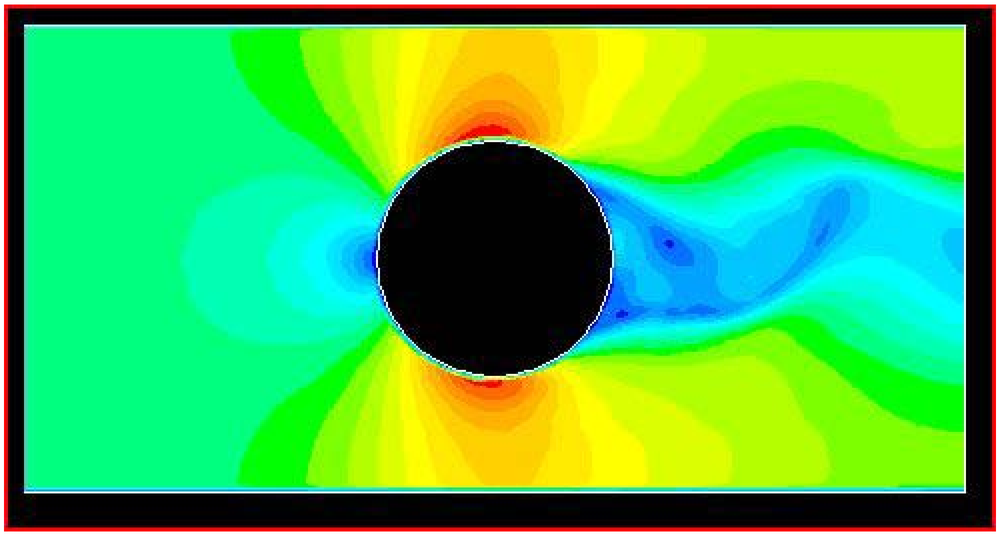
Pressure at .01 Resolution



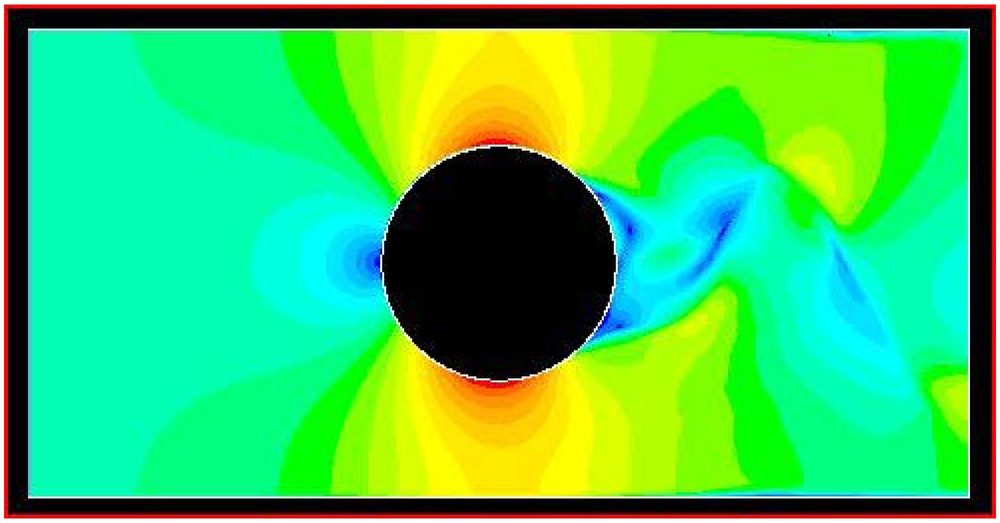
Velocity at .1 Resolution



Velocity at .05 Resolution



Velocity at .01 Resolution

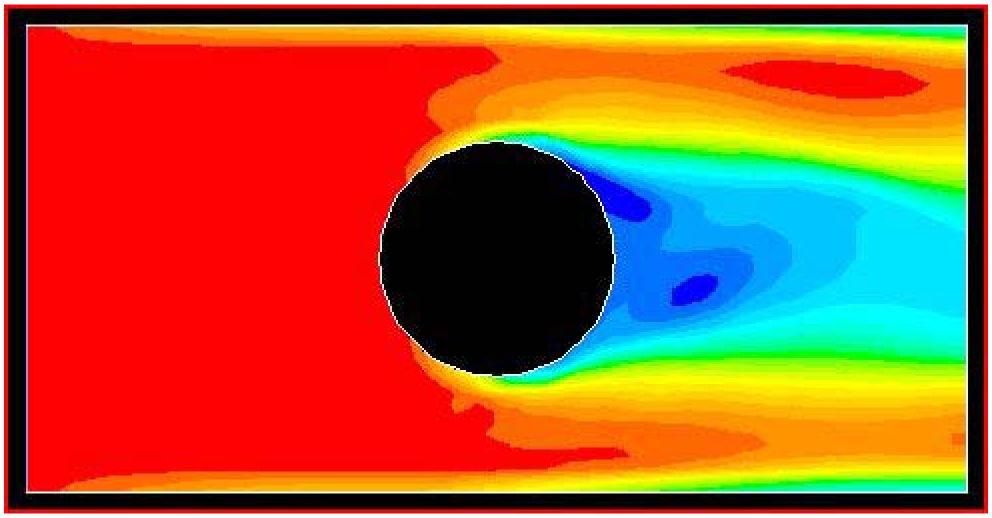


A1.3 Accuracy Testing at Resolutions of .1, .075 .05, .025 and .01 at Normal Viscosity

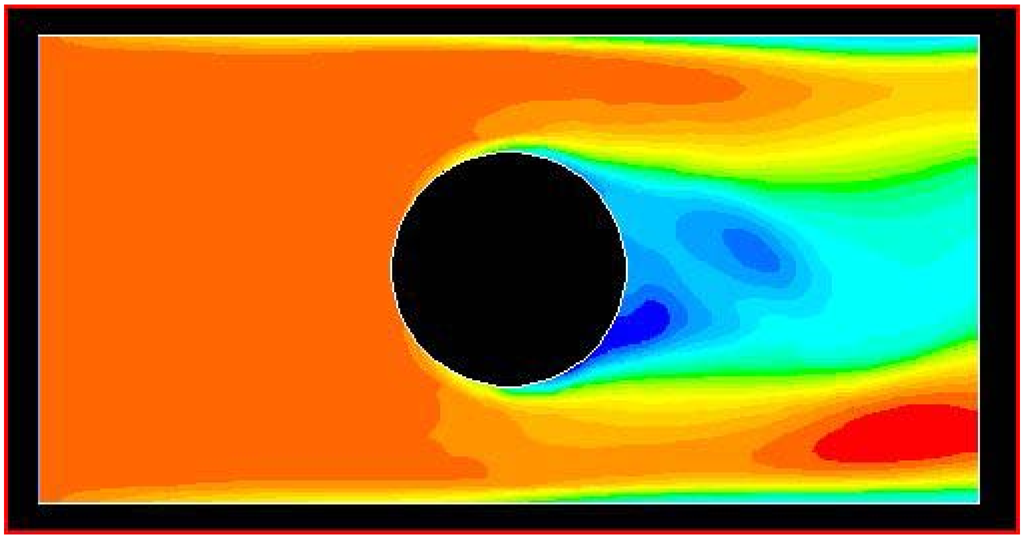
Accuracy Testing at Resolutions of .1, .05, and .01

Resolution	Density	Viscosity	Drag Force	Velocity	Drag Coefficient
0.1	998.2	0 . 00 103	0 . 00 193	0 . 00 1	1.93
0,75	998.2	0 . 00 103	0. 00 190	0. 00 1	1.9
0.05	998.2	0 . 00 103	0 . 00 208	0 . 00 1	2.08
0.03	998.2	0. 00 103	0. 00 283	0. 00 1	2.84
0.01	998.2	0 . 00 103	0 . 00 267	0 . 00 1	2.68

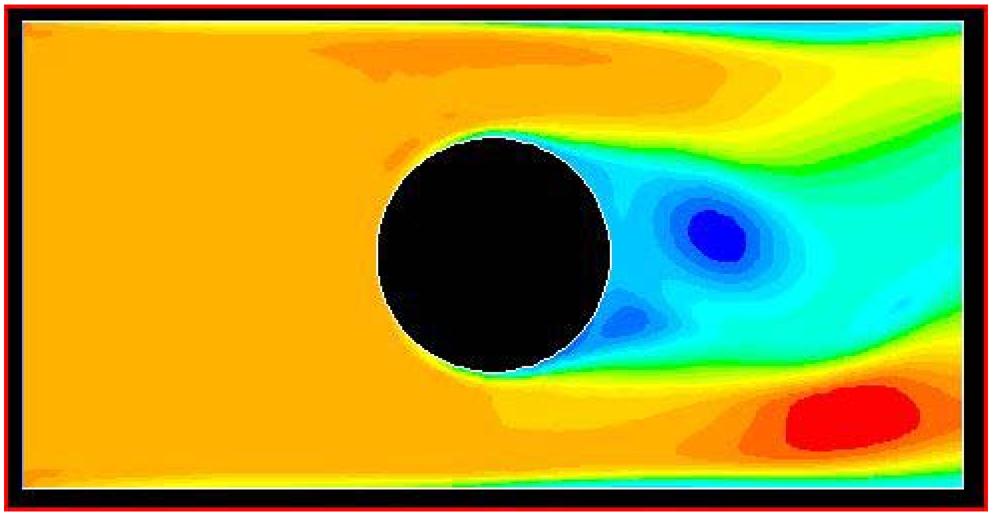
Pressure at .1 Resolution



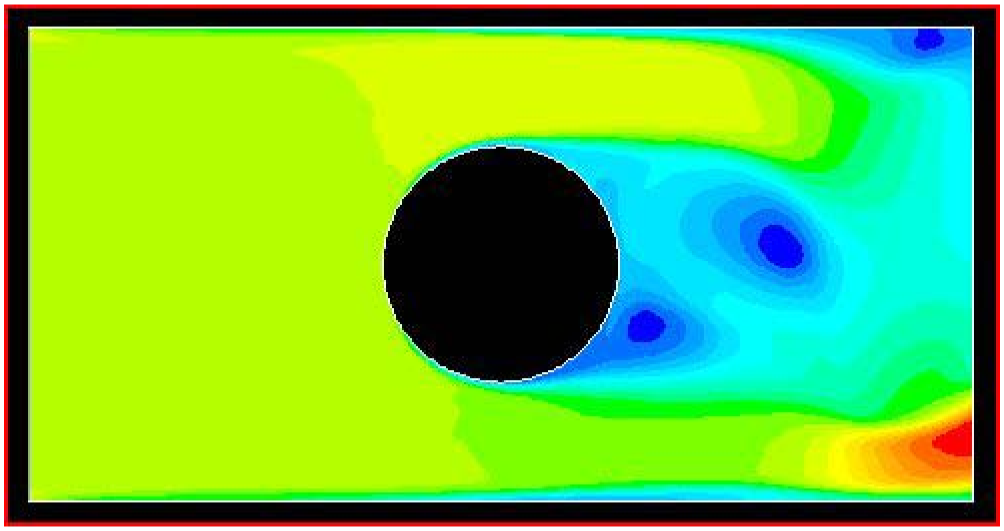
Pressure at .075 Resolution



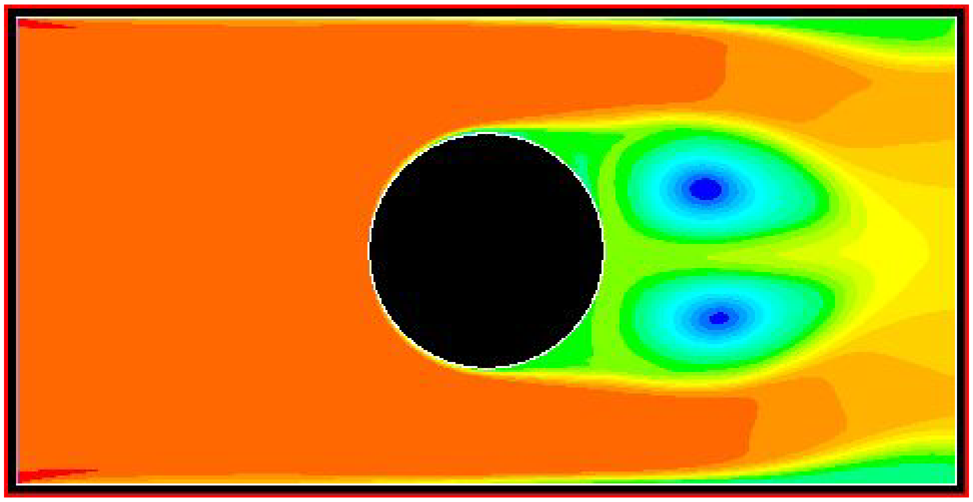
Pressure at .05 Resolution



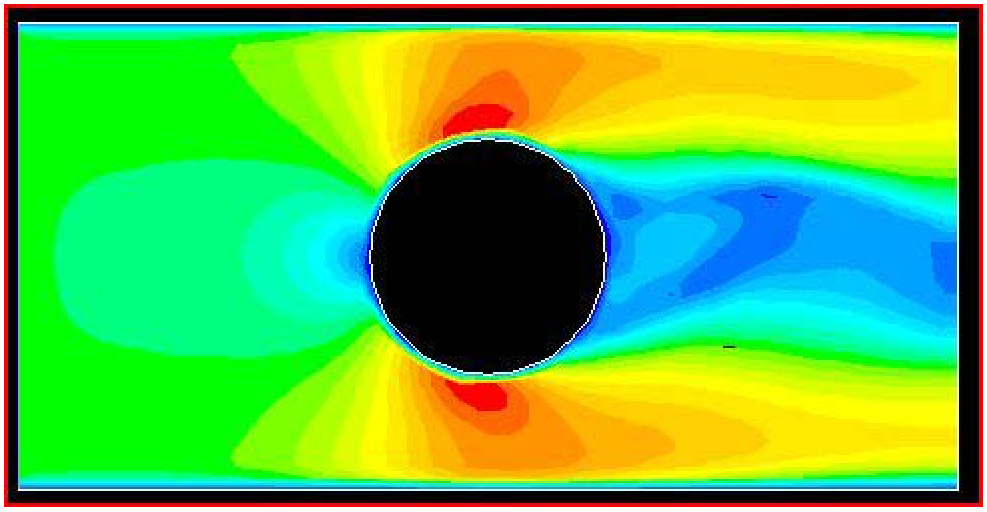
Pressure at .025 Resolution



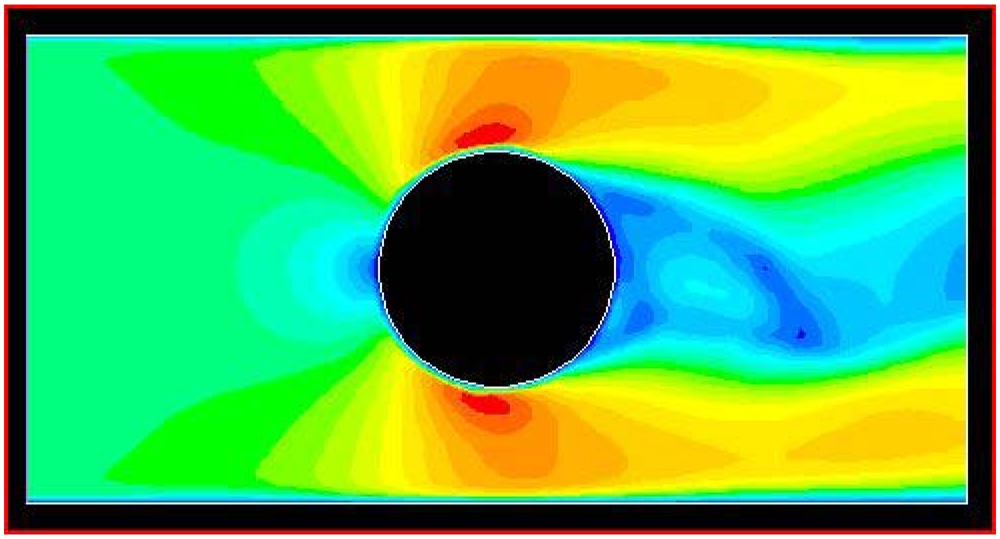
Pressure at .01 Resolution



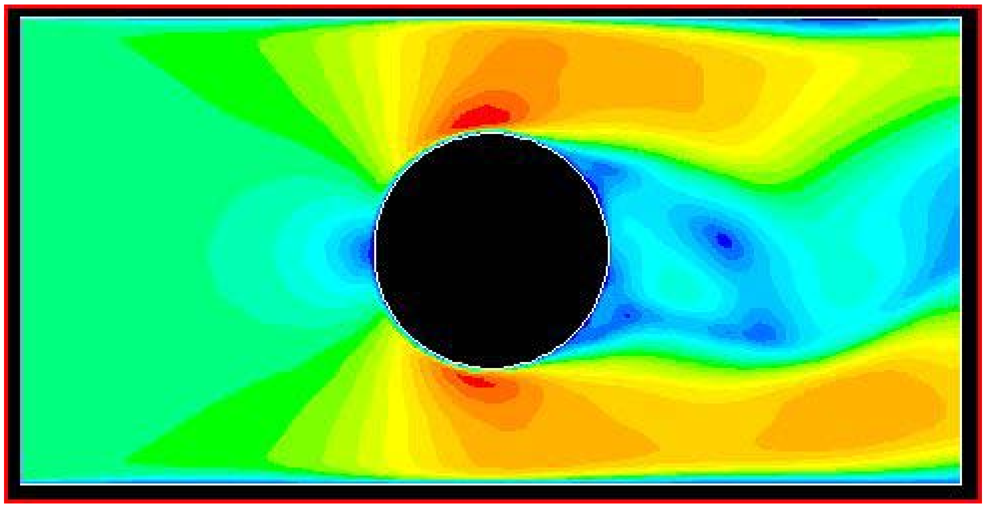
Velocity at .1 Resolution



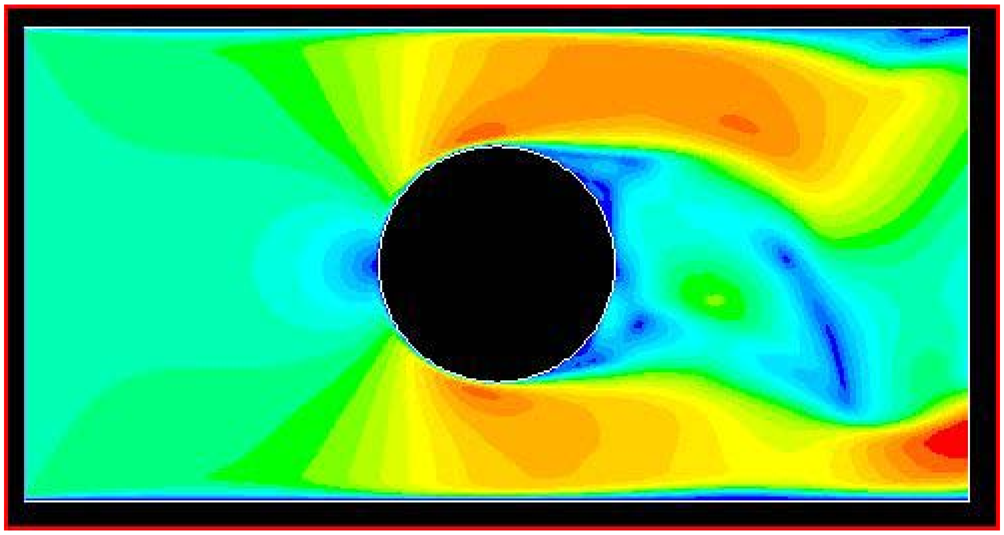
Velocity at .075 Resolution



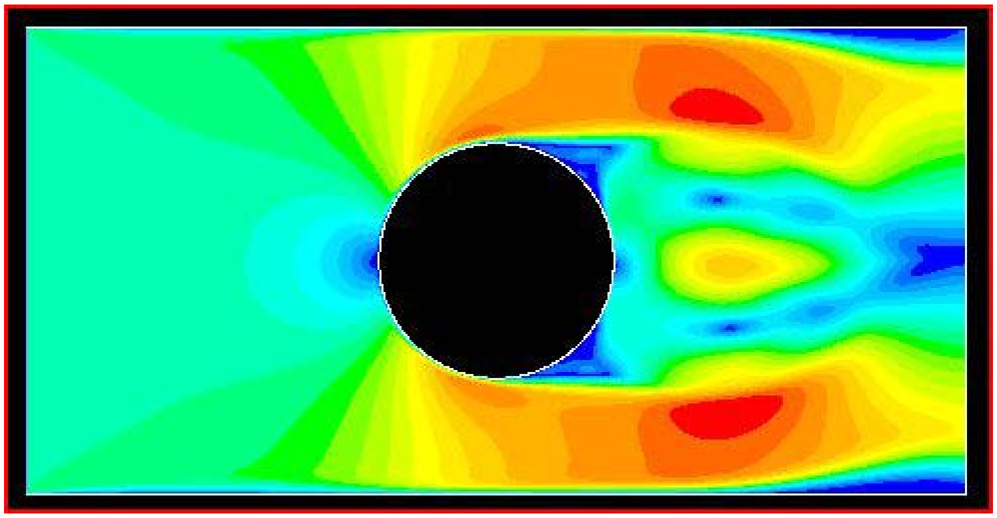
Velocity at .05 Resolution



Velocity at .025 Resolution



Velocity at .01 Resolution

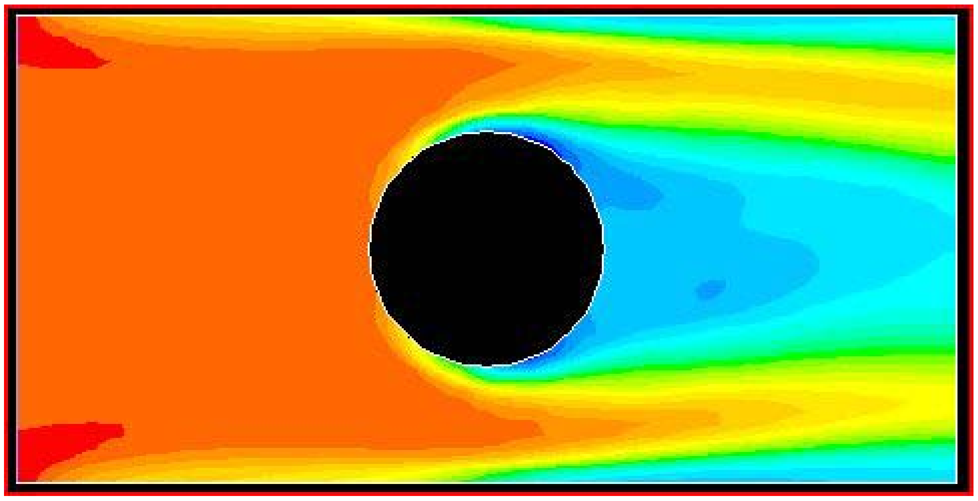


A1.4 Accuracy Testing at Resolutions of .1, .05, and .01 at High Viscosity

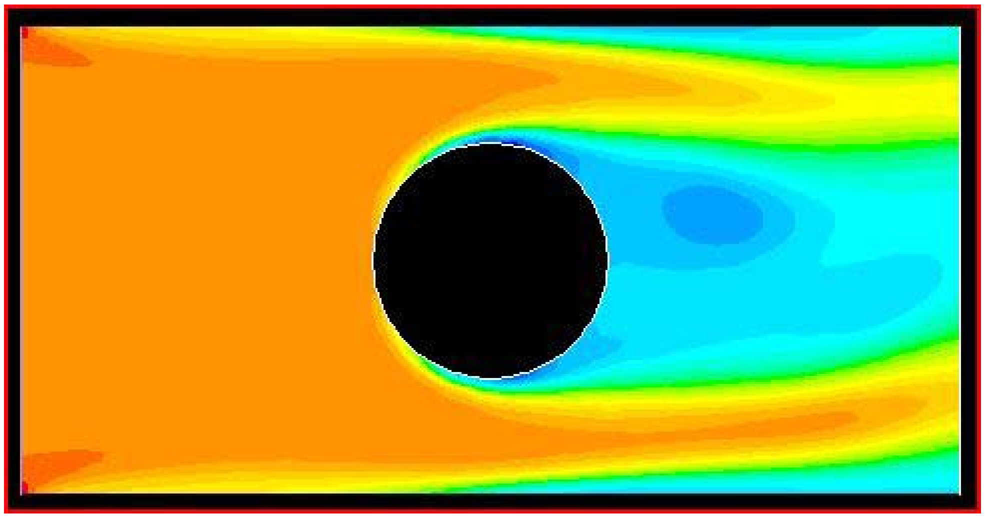
Accuracy Testing at Resolutions of .1, .05, and .01 at High Viscosity

Resolution	Density	Viscosity	Drag Force	Velocity	Drag Coef	Reynolds
0.1	998.2	0.01	0 . 00 447	0 . 00 1	4.48	199.64
0.05	998.2	0.01	0 . 00 468	0 . 00 1	4.68	199.64
0.01	998.2	0.01	0 . 00 566	0 . 00 1	567	199.64

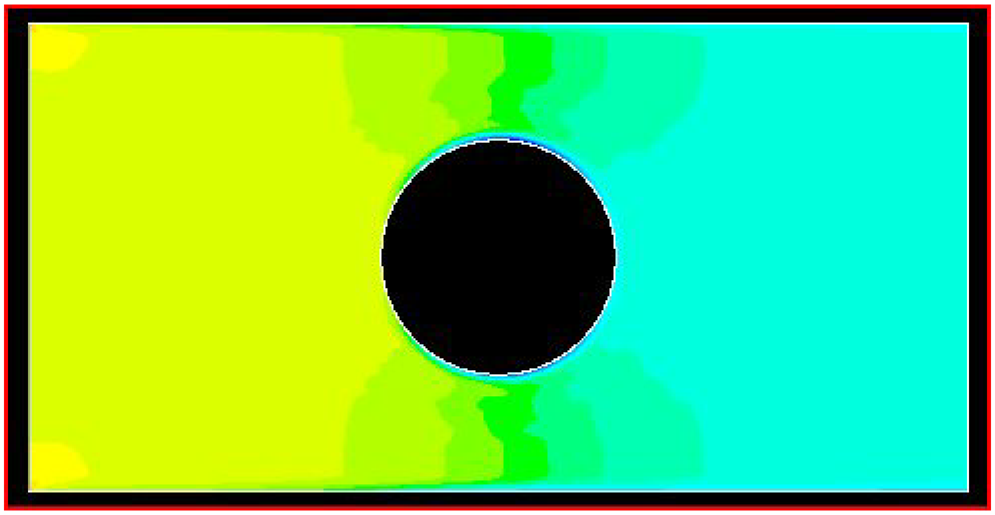
Pressure at .1 Resolution



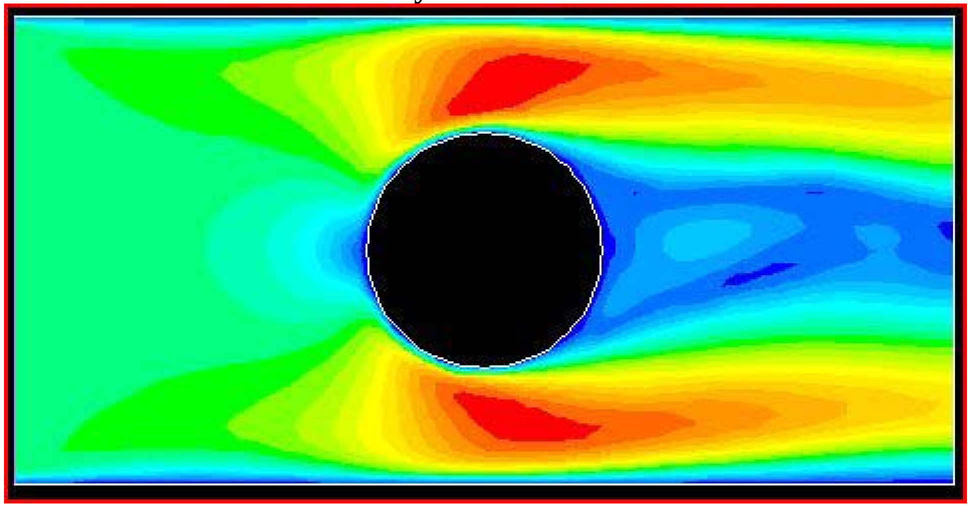
Pressure at .05 Resolution



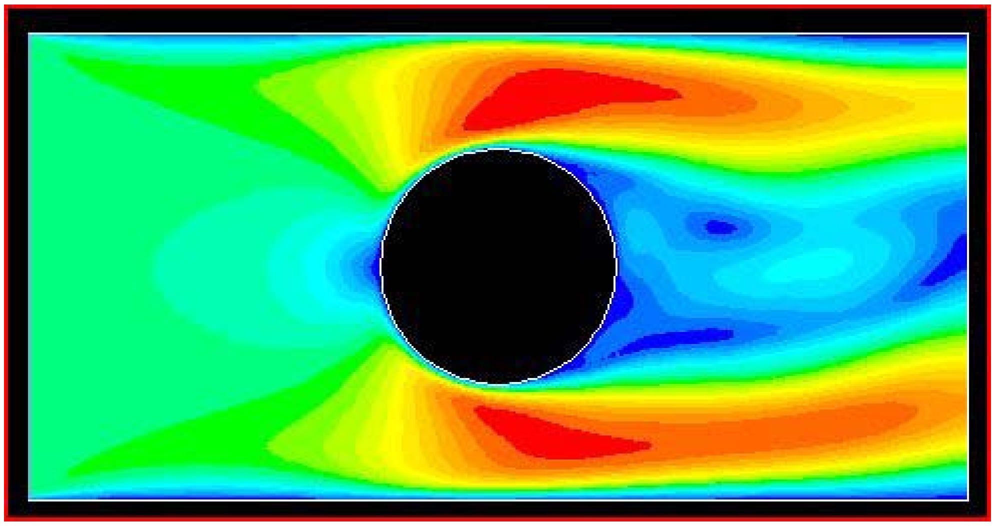
Pressure at .01 Resolution



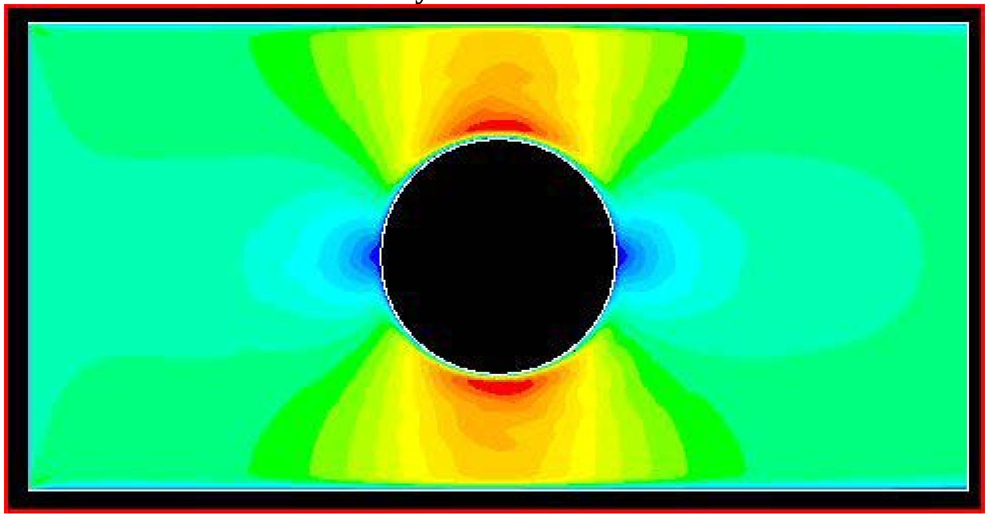
Velocity at .1 Resolution



Velocity at .05 Resolution



Velocity at .01 Resolution

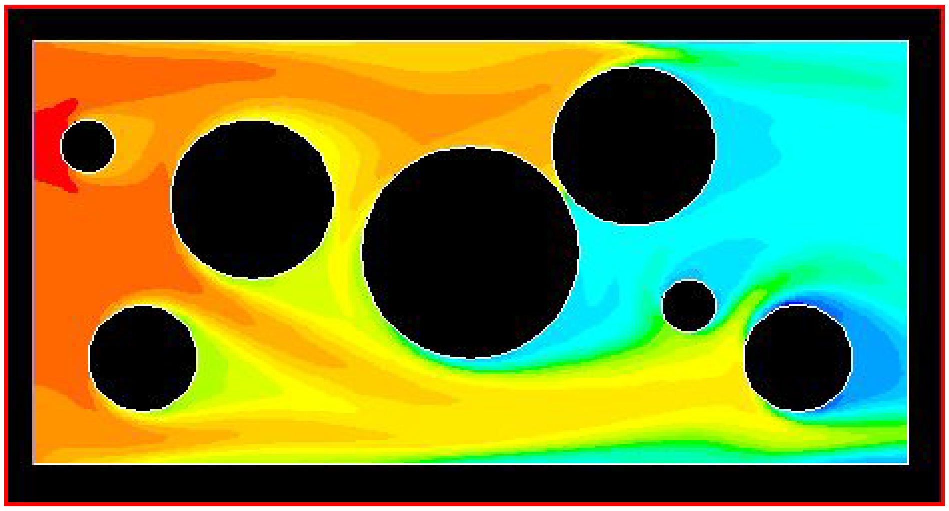


A2 Modeling Flow with Multiple Obstructions

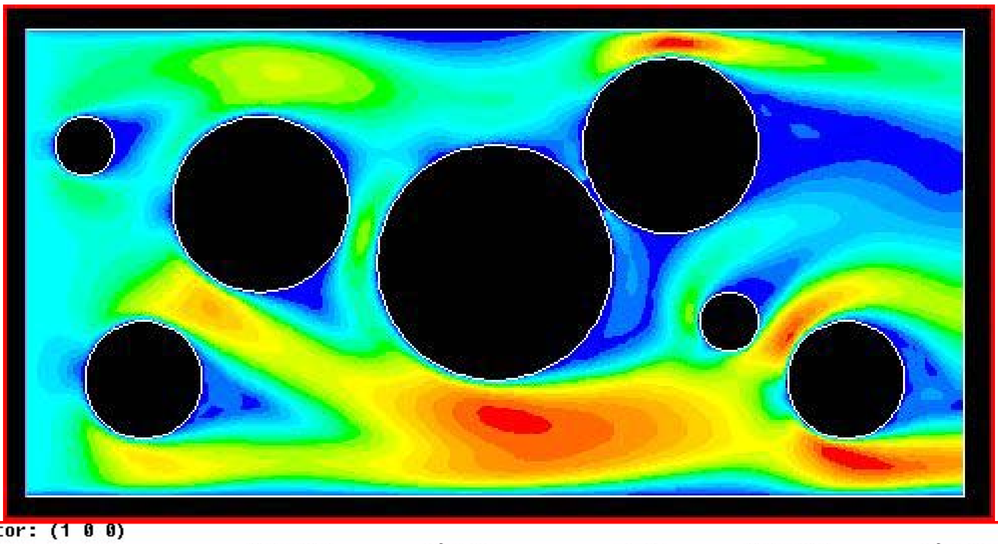
A2.1 Modeling Fluid flow in 2D with Seven Circles at 0.01 Resolution

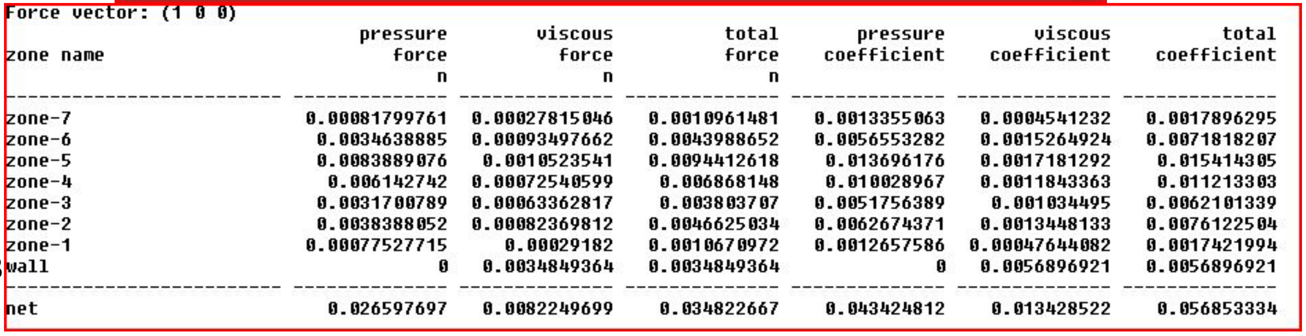
.05 multiple var 1	1	998.2	0.01	0.0011	0.5	0.001	4.27569625325586	199.64
	2	998.2	0.01	0.0047	1	0.001	9.34181526748147	199.64
	3	998.2	0.01	0.0038	1.5	0.001	5.08074534161491	199.64
	4	998.2	0.01	0.0069	2	0.001	6.8803846924464	199.64
	5	998.2	0.01	0.0094	1.5	0.001	12.6110465504575	199.64
	6	998.2	0.01	0.0044	1	0.001	8.81359487076738	199.64
	7	998.2	0.01	0.0011	0.5	0.001	4.39249889801643	199.64

Pressure



Velocity

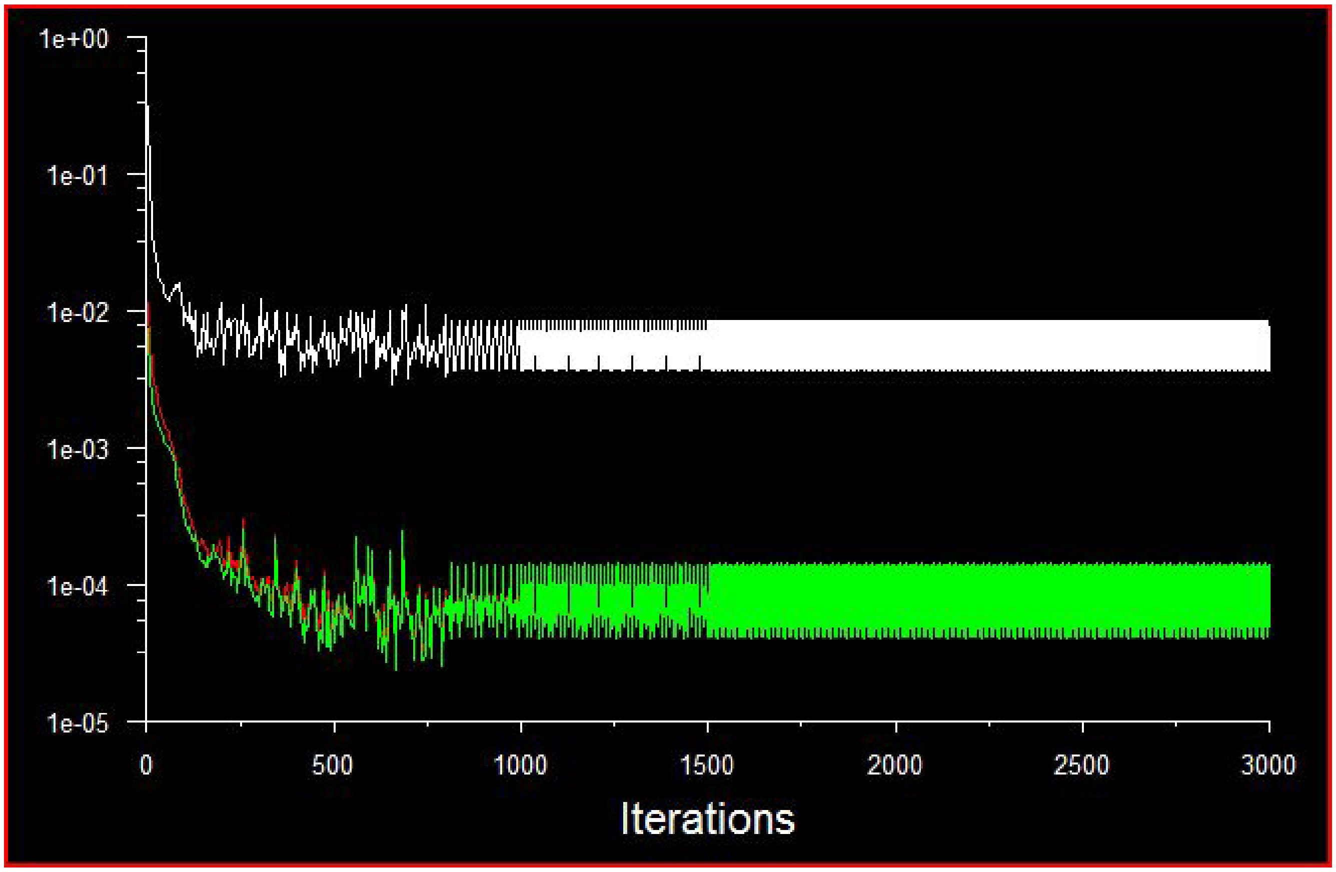




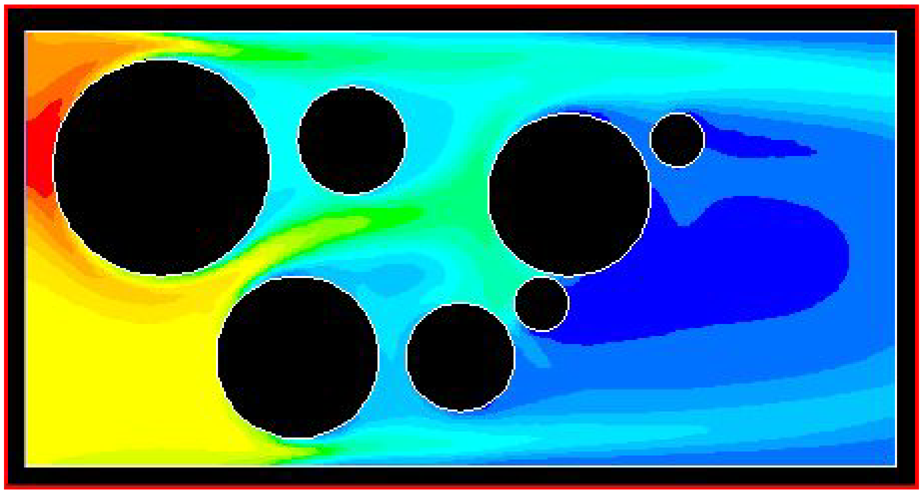
```
Force vector: (1 0 0)
                            pressure        viscous          total       pressure        viscous          total
zone name                      force          force          force    coefficient    coefficient    coefficient
                                   n              n              n
-------------------------- -------------- -------------- -------------- -------------- -------------- --------------
zone-7                 0.00081799761  0.00027815046   0.0010961481   0.0013355063   0.0004541232   0.0017896295
zone-6                  0.0034638885  0.00093497662   0.0043988652   0.0056553282   0.0015264924   0.0071818207
zone-5                  0.0083889076   0.0010523541   0.0094412618    0.013696176   0.0017181292    0.015414305
zone-4                   0.006142742  0.00072540599    0.006868148    0.010028967   0.0011843363    0.011213303
zone-3                  0.0031700789  0.00063362817    0.003803707   0.0051756389    0.001034495   0.0062101339
zone-2                  0.0038388052  0.00082369812   0.0046625034   0.0062674371   0.0013448133   0.0076122504
zone-1                 0.00077527715     0.00029182   0.0010670972   0.0012657586 0.00047644082   0.0017421994
wall                               0   0.0034849364   0.0034849364              0   0.0056896921   0.0056896921
-------------------------- -------------- -------------- -------------- -------------- -------------- --------------
net                      0.026597697   0.0082249699    0.034822667    0.043424812    0.013428522    0.056853334
```

.05 multiple var 2	1	998.2	0.01	0.0006	0.5	0.001	2.51792426367461	199.64
	2	998.2	0.01	0.0021	1	0.001	4.29553195752354	199.64
	3	998.2	0.01	0.0089	1.5	0.001	11.9378681626928	199.64
	4	998.2	0.01	0.0295	2	0.001	29.5031055900621	199.64
	5	998.2	0.01	0.0019	1.5	0.001	2.49115073799506	199.64
	6	998.2	0.01	0.0125	1	0.001	25.1033860949709	199.64
	7	998.2	0.01	0.0018	0.5	0.001	7.12474454017231	199.64

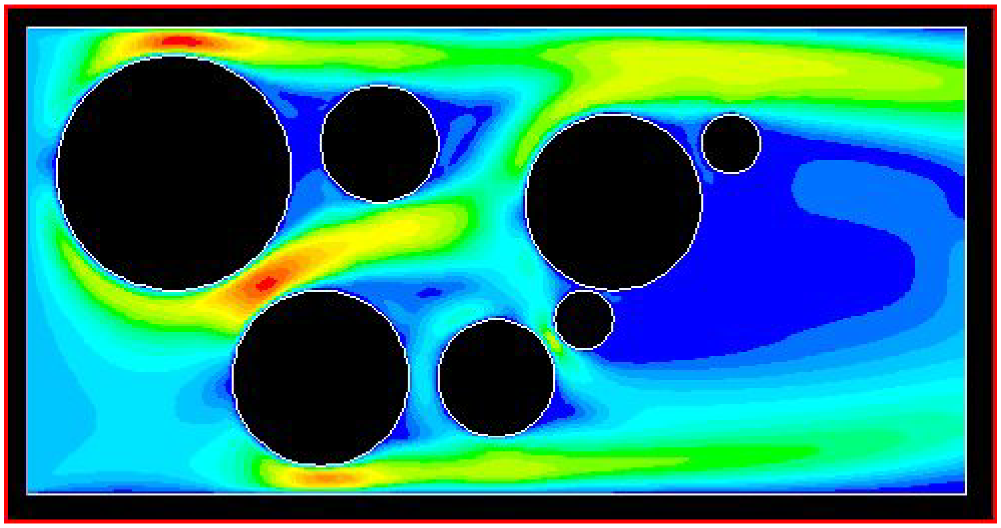
Iterations

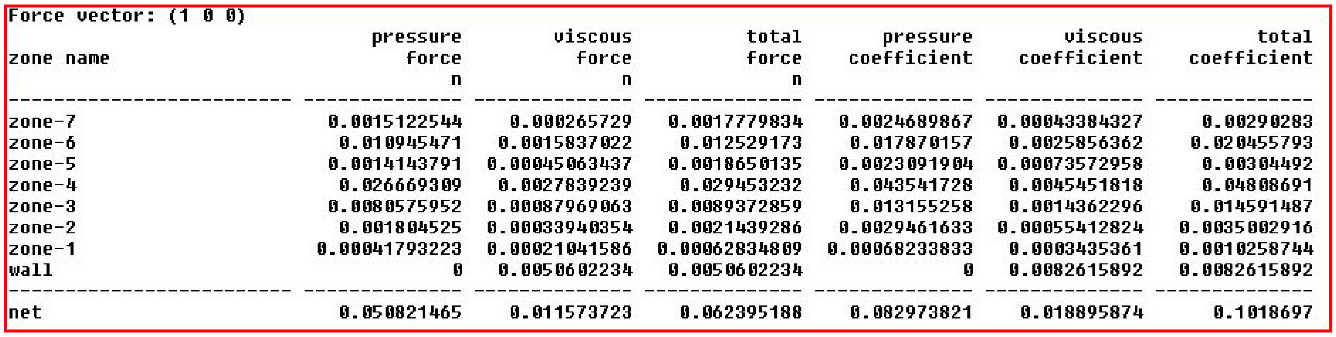


Pressure



Velocity

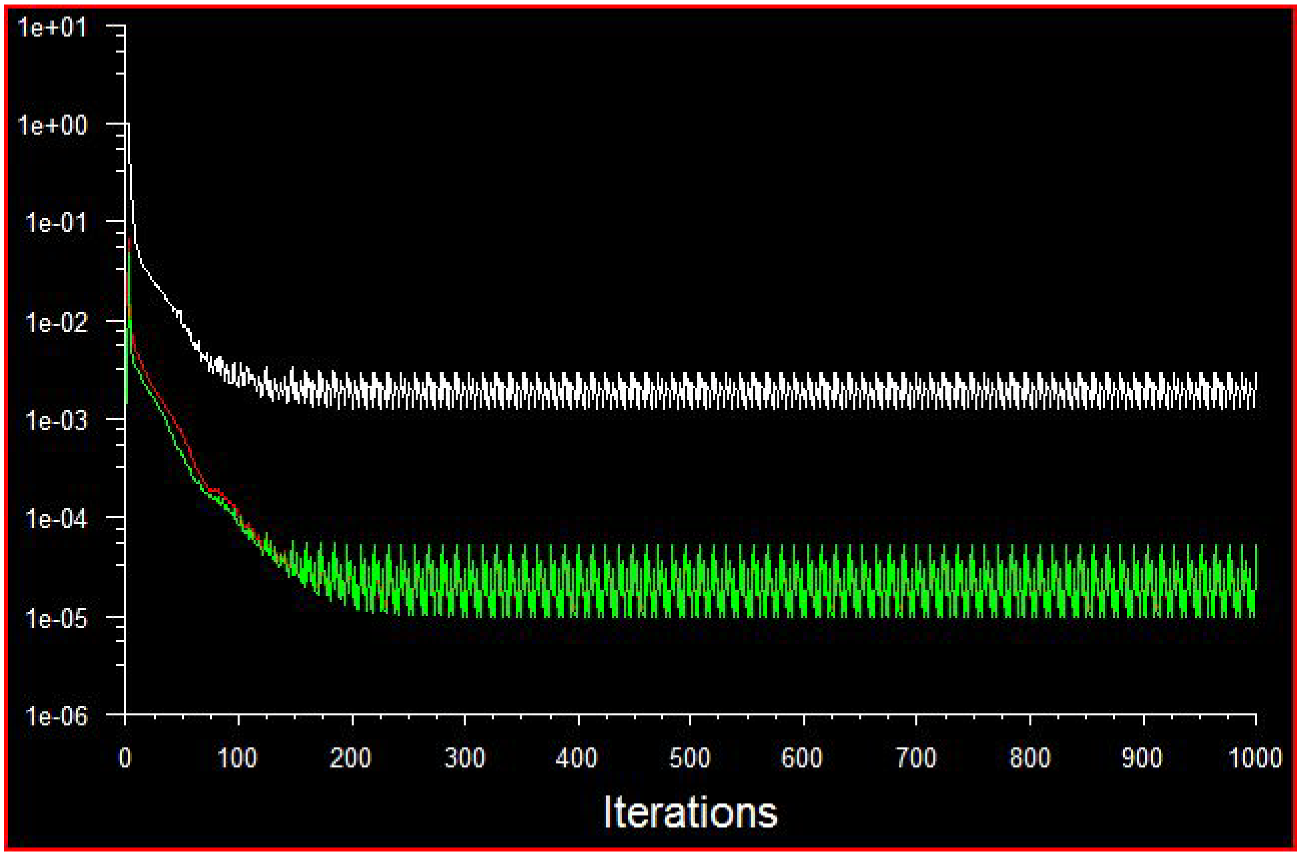




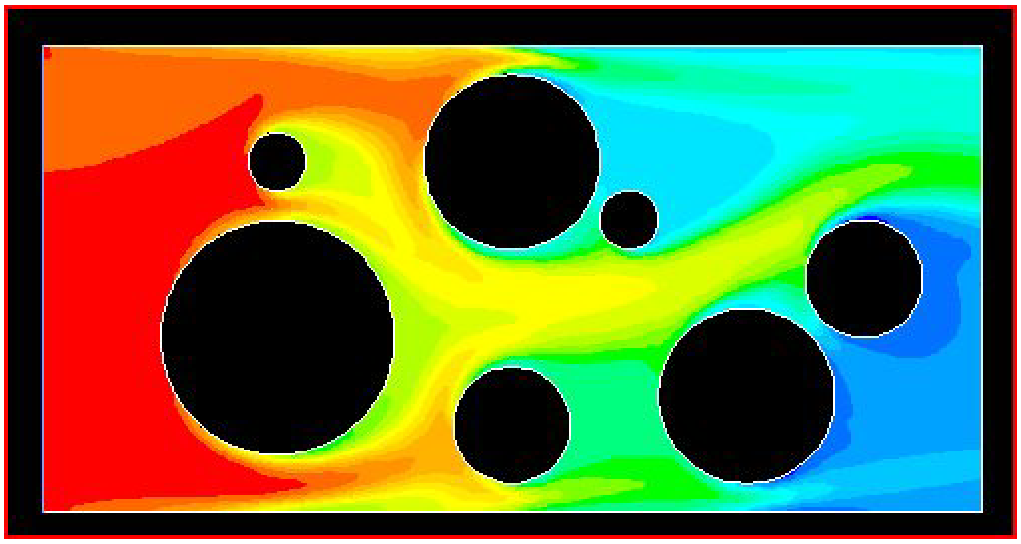
```
Force vector: (1 0 0)
                              pressure          viscous            total         pressure          viscous            total
zone name                        force            force            force      coefficient      coefficient      coefficient
                                     n                n                n
-------------------------  --------------  --------------  --------------  --------------  --------------  --------------
zone-7                    0.0015122544     0.000265729    0.0017779834    0.0024689867   0.00043384327      0.00290283
zone-6                     0.010945471    0.0015837022     0.012529173     0.017870157    0.0025856362     0.020455793
zone-5                    0.0014143791   0.00045063437    0.0018650135    0.0023091904   0.00073572958      0.00304492
zone-4                     0.026669309    0.0027839239     0.029453232     0.043541728    0.0045451818      0.04808691
zone-3                    0.0080575952   0.00087969063    0.0089372859     0.013155258    0.0014362296     0.014591487
zone-2                     0.001804525   0.00033940354    0.0021439286    0.0029461633   0.00055412824    0.0035002916
zone-1                   0.00041793223   0.00021041586   0.00062834809   0.00068233833    0.0003435361    0.0010258744
wall                                 0    0.0050602234    0.0050602234               0    0.0082615892    0.0082615892
-------------------------  --------------  --------------  --------------  --------------  --------------  --------------
net                        0.050821465     0.011573723     0.062395188     0.082973821     0.018895874       0.1018697
```

.05 multiple var 3	1	998.2	0.01	0.0014	0.5	0.001	5.44580244439992	199.64
	2	998.2	0.01	0.0036	1	0.001	7.27755960729313	199.64
	3	998.2	0.01	0.0079	1.5	0.001	10.536298670941	199.64
	4	998.2	0.01	0.0124	2	0.001	12.4223602484472	199.64
	5	998.2	0.01	0.0099	1.5	0.001	13.2371602217325	199.64
	6	998.2	0.01	0.0030	1	0.001	5.96193147665798	199.64
	7	998.2	0.01	0.0026	0.5	0.001	10.441194149469	199.64

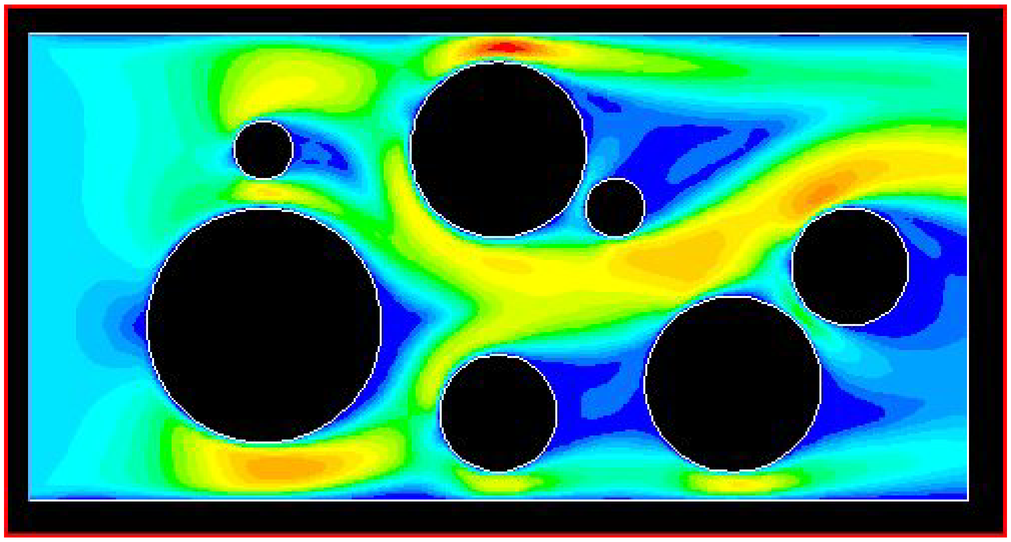
Iterations

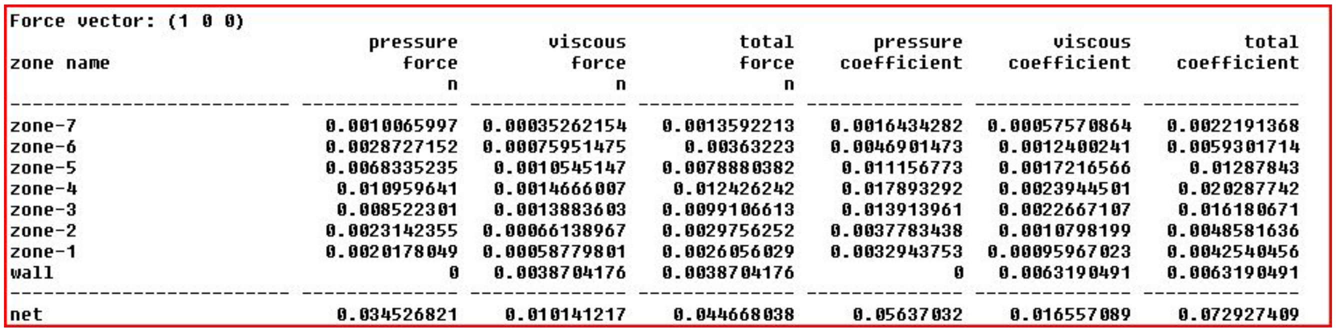


Pressure



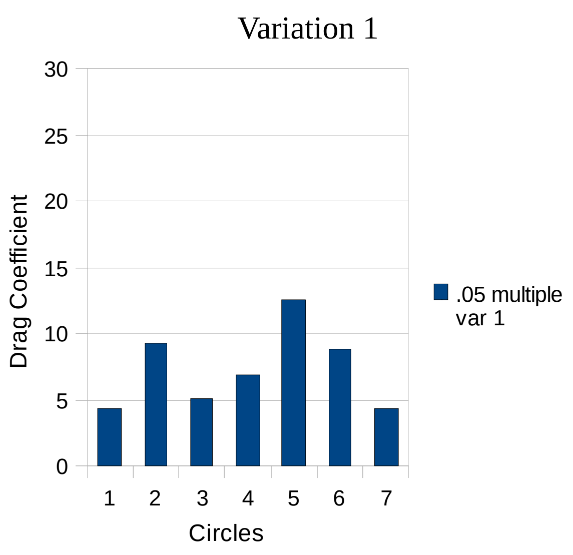
Velocity

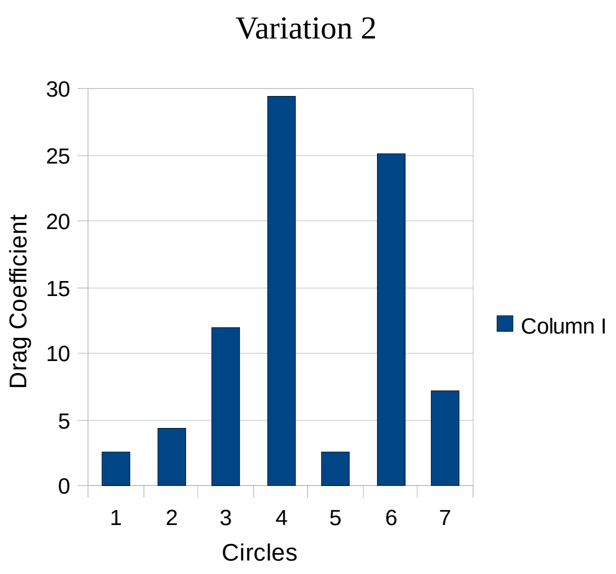


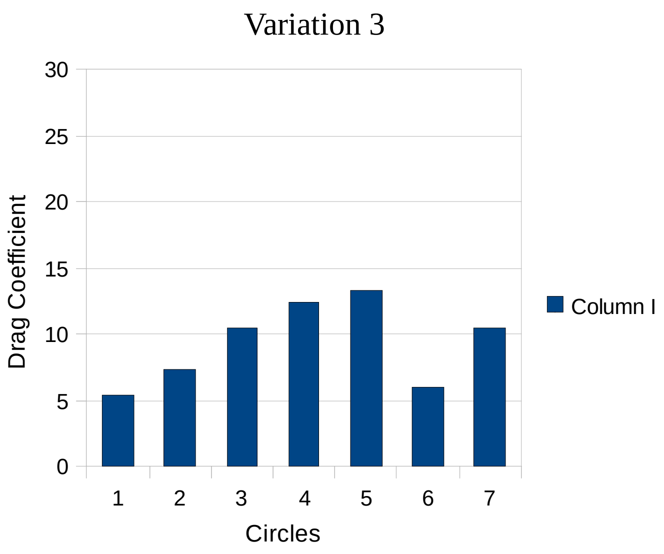


```
Force vector: (1 0 0)
                                 pressure         viscous           total        pressure         viscous           total
zone name                           force           force           force     coefficient     coefficient     coefficient
                                        n               n               n
------------------------- --------------- --------------- --------------- --------------- --------------- ---------------
zone-7                       0.0010065997   0.00035262154    0.0013592213    0.0016434282   0.00057570864    0.0022191368
zone-6                       0.0028727152   0.00075951475      0.00363223    0.0046901473    0.0012400241    0.0059301714
zone-5                       0.0068335235    0.0010545147    0.0078880382     0.011156773    0.0017216566      0.01287843
zone-4                        0.010959641    0.0014666007     0.012426242     0.017893292    0.0023944501     0.020287742
zone-3                        0.008522301    0.0013883603    0.0099106613     0.013913961    0.0022667107     0.016180671
zone-2                       0.0023142355   0.00066138967    0.0029756252    0.0037783438    0.0010798199    0.0048581636
zone-1                       0.0020178049   0.00058779801    0.0026056029    0.0032943753   0.00095967023    0.0042540456
wall                                    0    0.0038704176    0.0038704176               0    0.0063190491    0.0063190491
------------------------- --------------- --------------- --------------- --------------- --------------- ---------------
net                           0.034526821     0.010141217     0.044668038      0.05637032     0.016557089     0.072927409
```

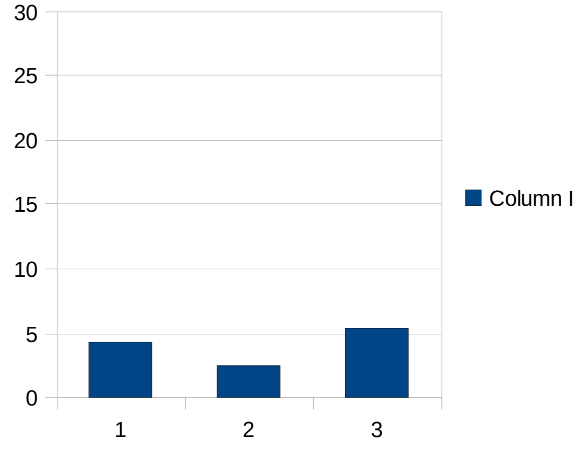
Comparison of the 3 Variations in 2D



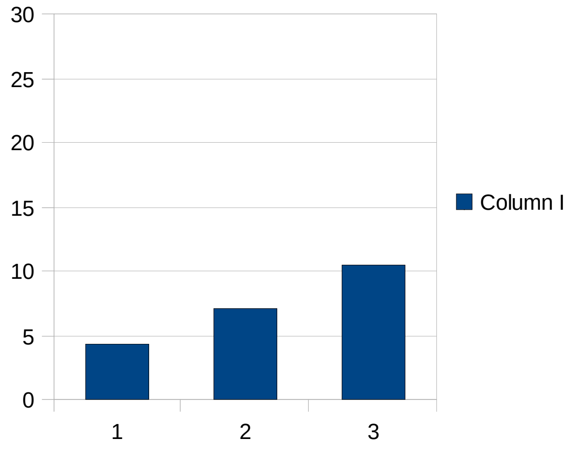




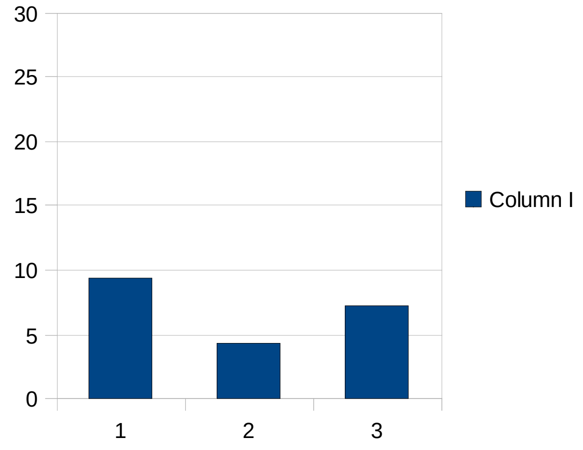
Circle 1



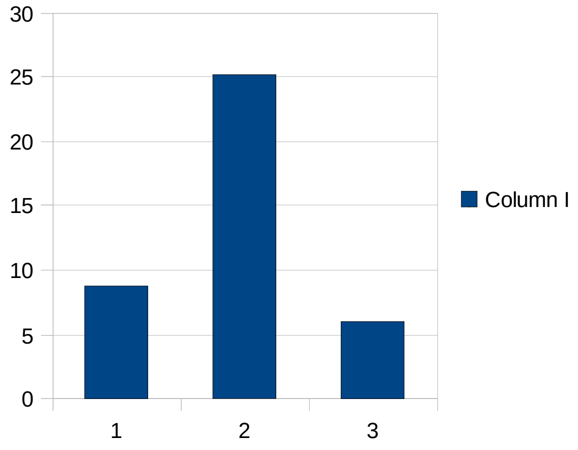
Circle 7

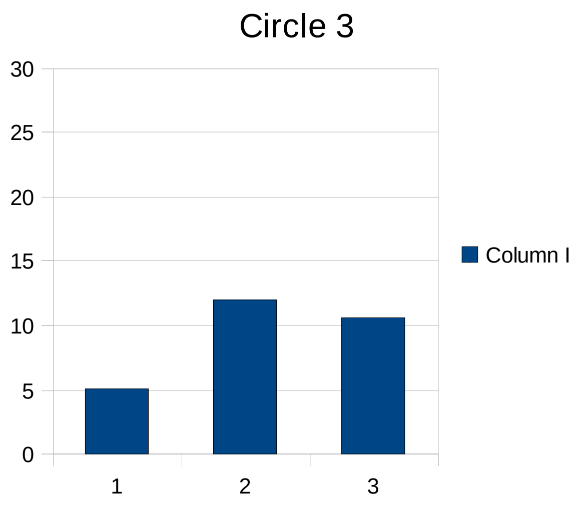


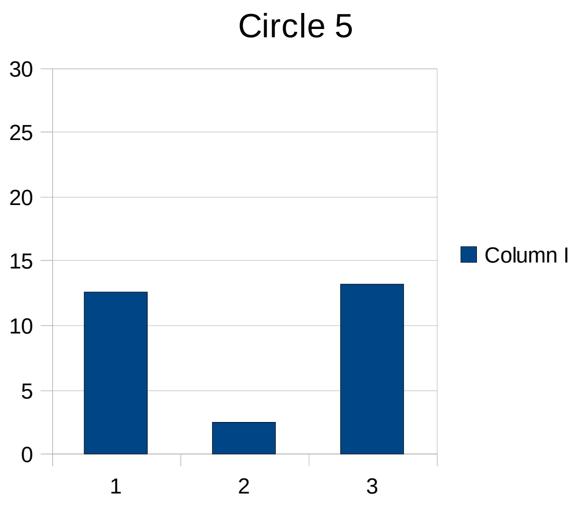
Circle 2

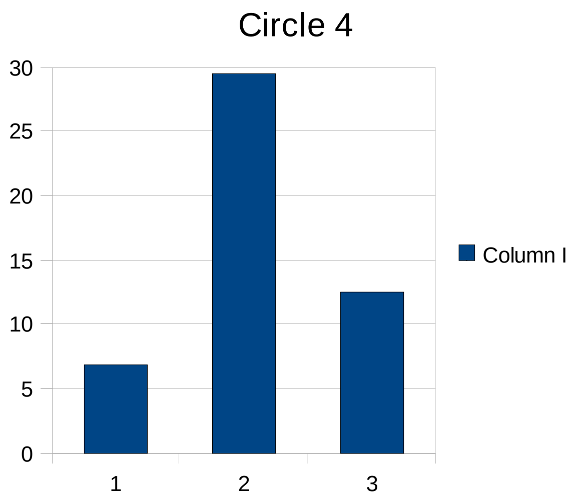


Circle 6







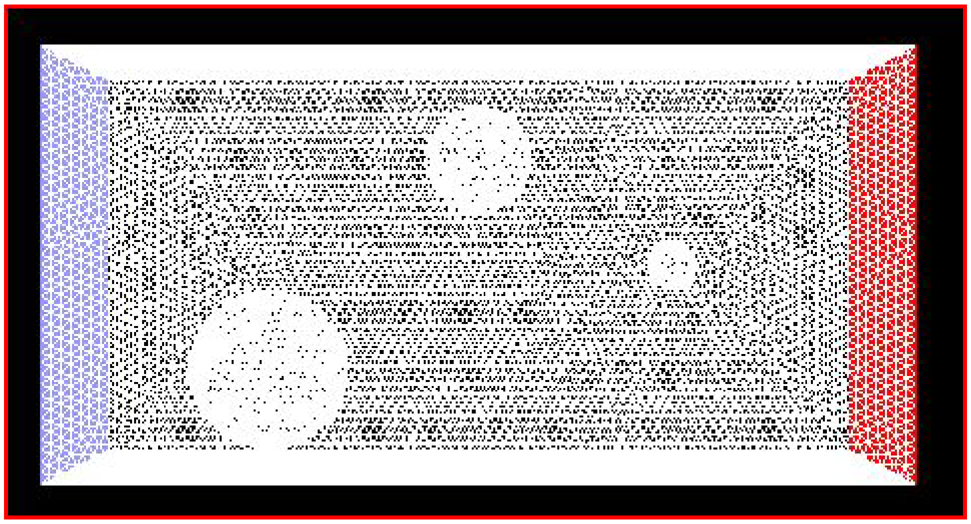


Circles	Void Frac.
1 and 7	0.01
2 and 6	0.02
3 and 5	0.06
and 4	0.1
Total	0.27

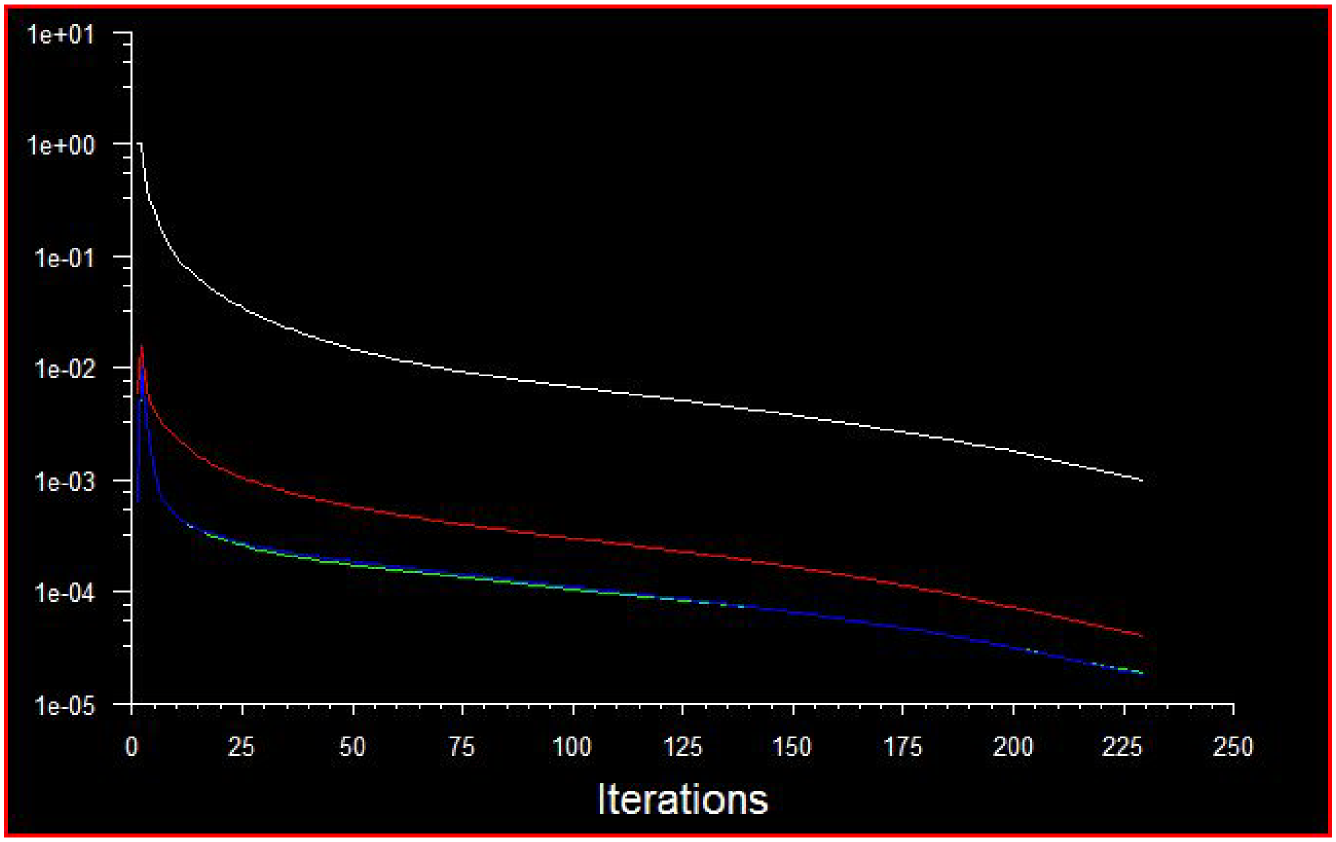
A2.2 Modeling Fluid flow in 3D with Three Spheres at 0.1 Resolution

3d multiple var 1	1	998.2	0.01	0.0004	0.2	0.001	3.79976925885019
	2	998.2	0.01	0.0009	0.79	0.001	2.31121932025478
	3	998.2	0.01	0.0022	1.77	0.001	2.48270231403779

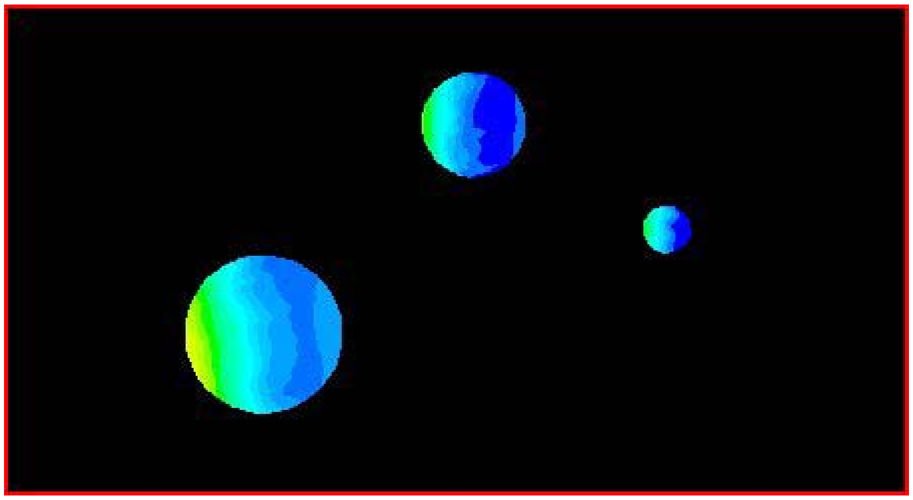
Model



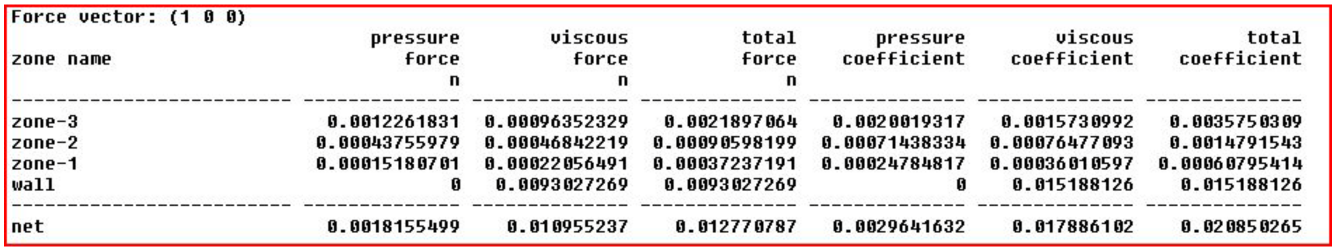
Iterations



Pressure



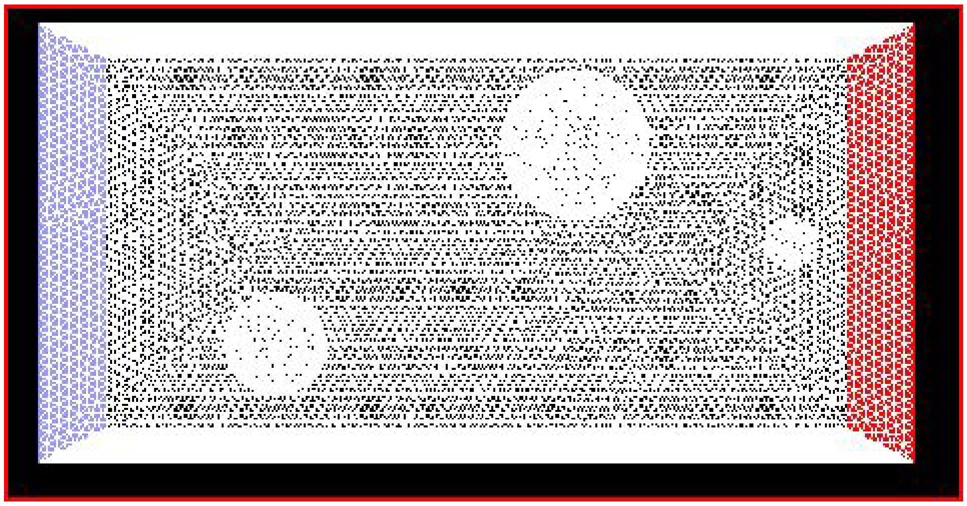
Forces



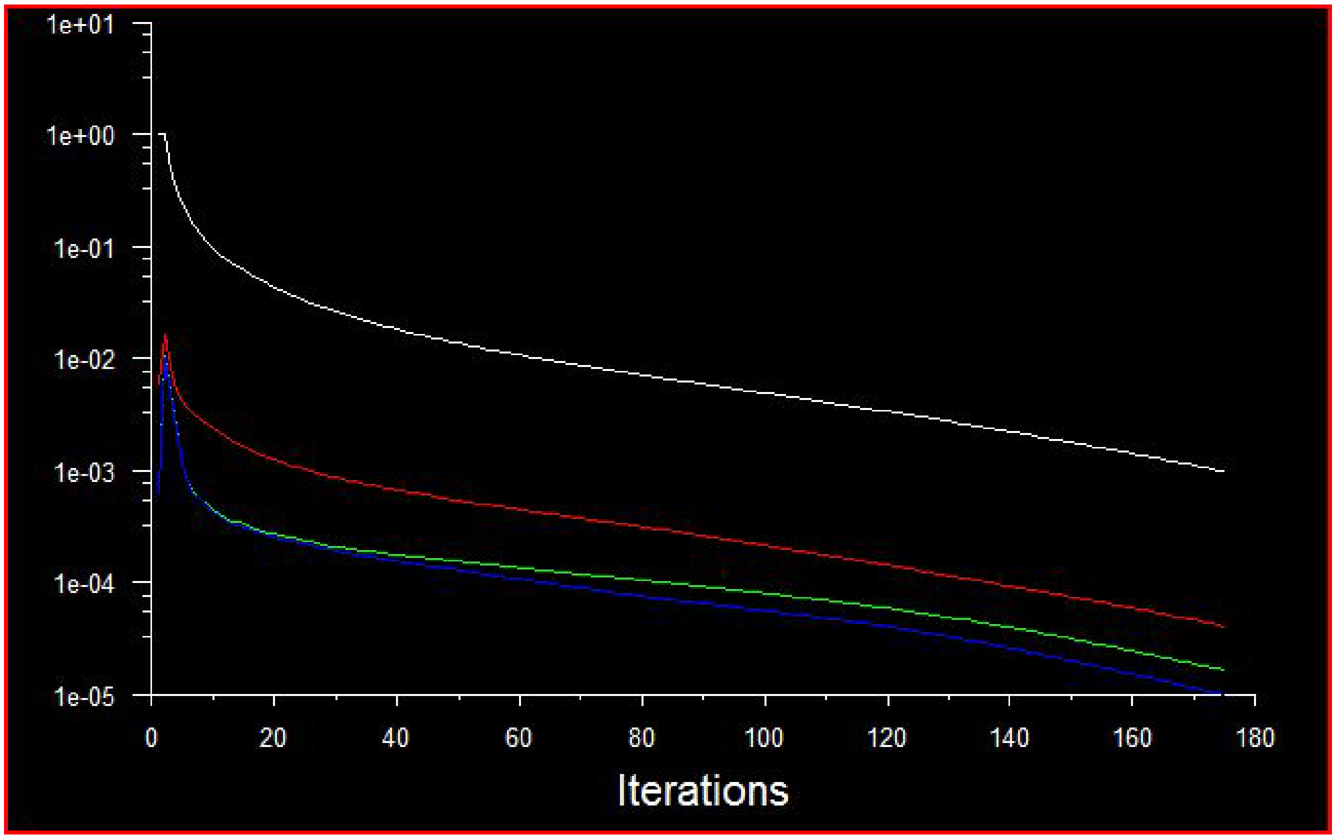
```
Force vector: (1 0 0)
                              pressure        viscous          total       pressure        viscous          total
zone name                        force          force          force    coefficient    coefficient    coefficient
                                     n              n              n
-------------------------  -------------- -------------- -------------- -------------- -------------- --------------
zone-3                       0.0012261831  0.00096352329   0.0021897064   0.0020019317   0.0015730992   0.0035750309
zone-2                      0.00043755979  0.00046842219  0.00090598199  0.00071438334  0.00076477093   0.0014791543
zone-1                      0.00015180701  0.00022056491  0.00037237191  0.00024784817  0.00036010597  0.00060795414
wall                                    0   0.0093027269   0.0093027269              0    0.015188126    0.015188126
-------------------------  -------------- -------------- -------------- -------------- -------------- --------------
net                          0.0018155499    0.010955237    0.012770787   0.0029641632    0.017886102    0.020850265
```

3d multiple var 2	1	998.2	0.01	0.0002	0.2	0.001	2.17351250674085
	2	998.2	0.01	0.0006	0.79	0.001	1.46431476393104
	3	998.2	0.01	0.0010	1.77	0.001	1.189365998733

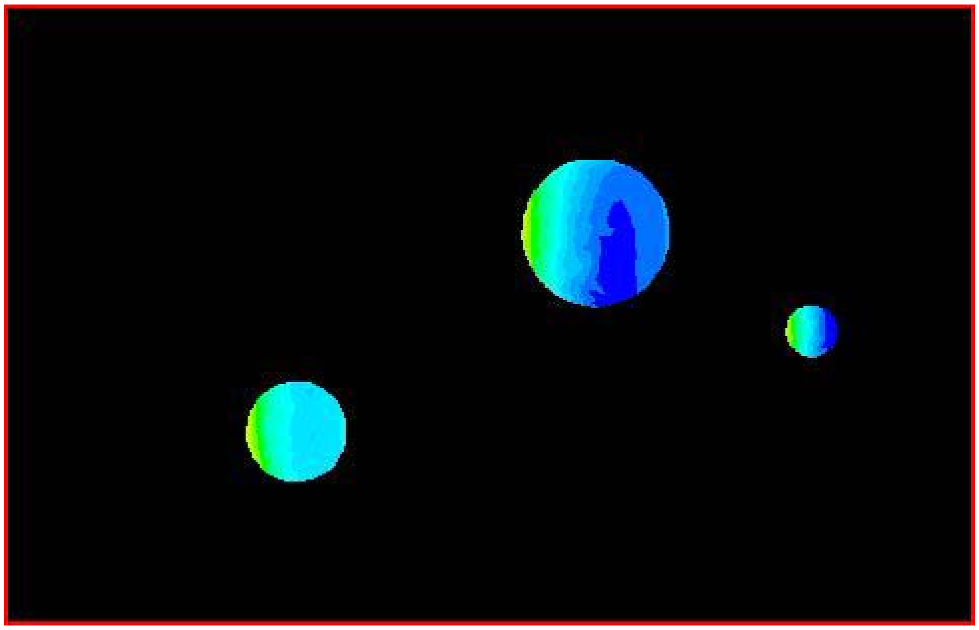
Model



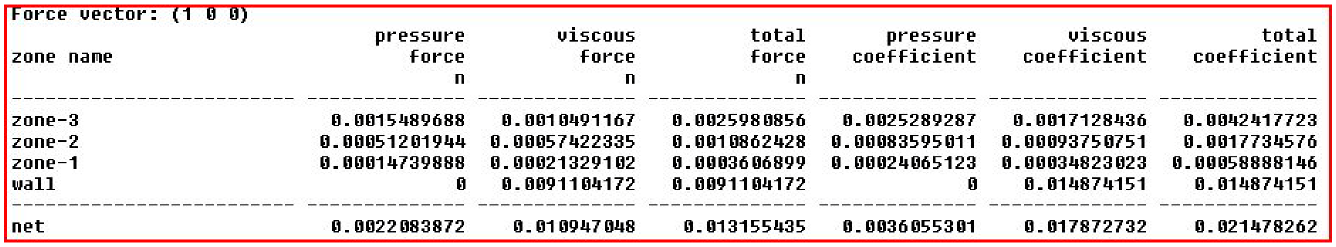
Iterations



Pressure



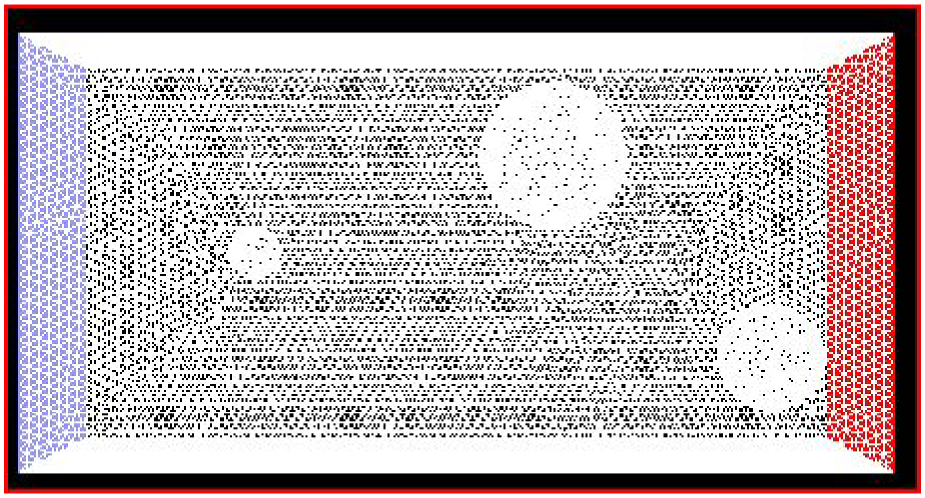
Forces



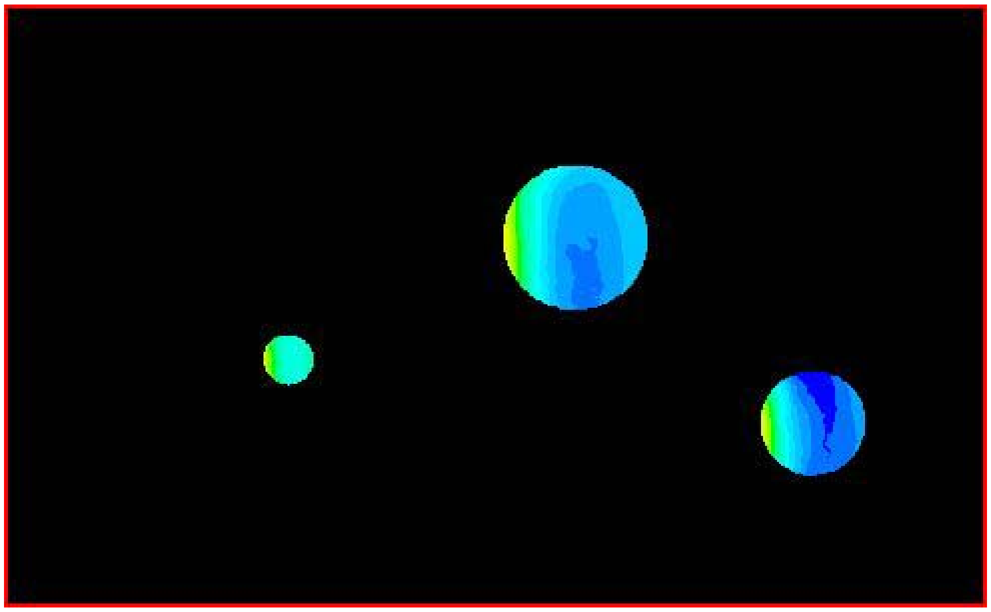
```
Force vector: (1 0 0)
                              pressure          viscous            total         pressure          viscous            total
zone name                        force            force            force      coefficient      coefficient      coefficient
                                     n                n                n
-------------------------  --------------  --------------  --------------  --------------  --------------  --------------
zone-3                    0.0015489688    0.0010491167    0.0025980856    0.0025289287    0.0017128436    0.0042417723
zone-2                  0.00051201944   0.00057422335    0.0010862428   0.00083595011   0.00093750751    0.0017734576
zone-1                  0.00014739888   0.00021329102    0.0003606899   0.00024065123   0.00034823023   0.00058888146
wall                                0    0.0091104172    0.0091104172               0     0.014874151     0.014874151
-------------------------  --------------  --------------  --------------  --------------  --------------  --------------
net                       0.0022083872     0.010947048     0.013155435    0.0036055301     0.017872732     0.021478262
```

3d multiple var 3	1	998.2	0.01	0.00024	0.2	0.001	2.47964102881703
	2	998.2	0.01	0.00111	0.79	0.001	2.82913775818733
	3	998.2	0.01	0.00220	1.77	0.001	2.49175589500679

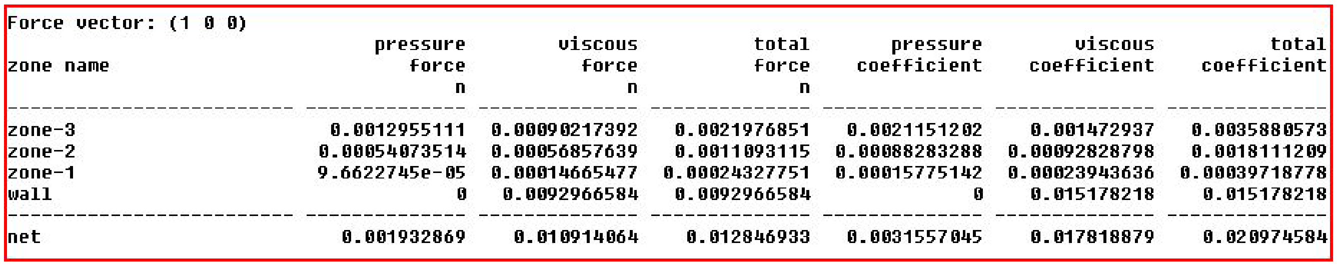
Model



Pressure

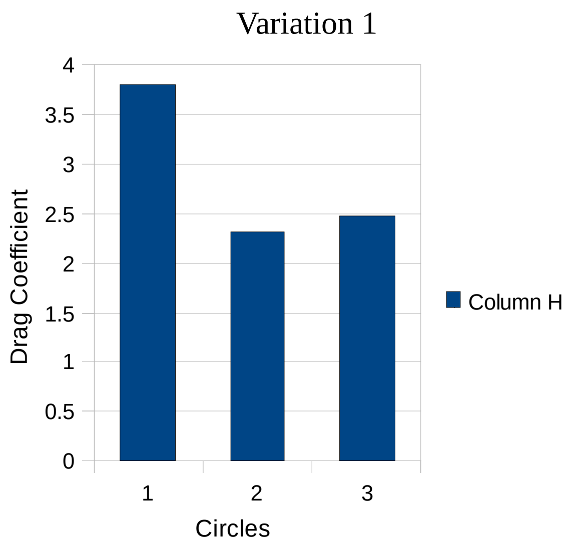


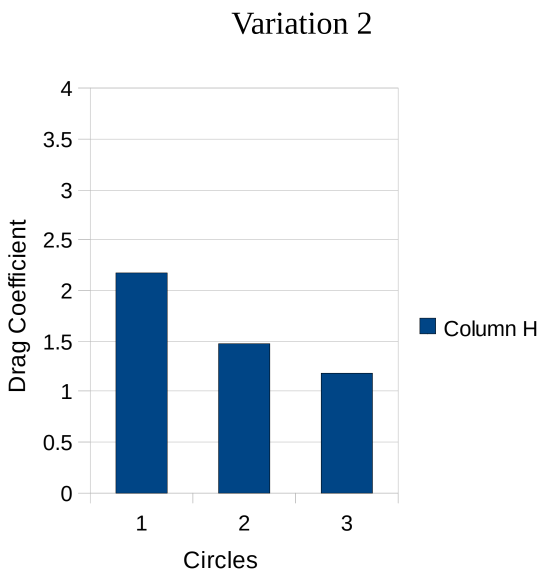
Forces

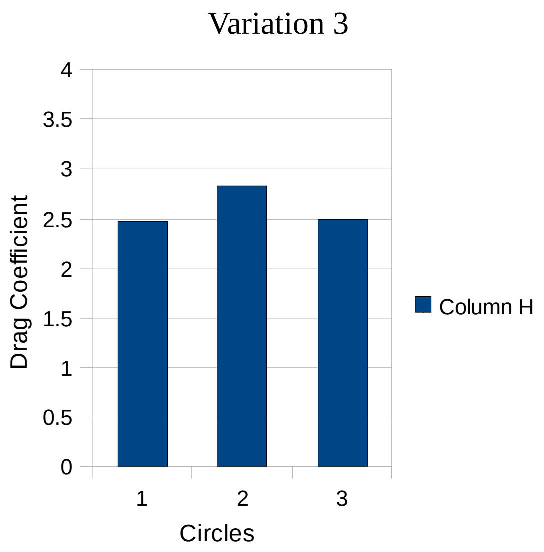


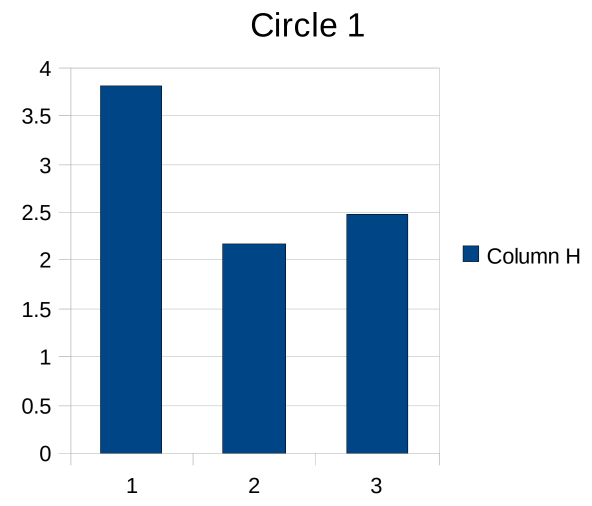
```
Force vector: (1 0 0)
                                pressure        viscous          total       pressure        viscous          total
zone name                          force          force          force    coefficient    coefficient    coefficient
                                       n              n              n
-------------------------- -------------- -------------- -------------- -------------- -------------- --------------
zone-3                      0.0012955111  0.00090217392   0.0021976851   0.0021151202    0.001472937   0.0035880573
zone-2                     0.00054073514  0.00056857639   0.0011093115  0.00088283288  0.00092828798   0.0018111209
zone-1                     9.6622745e-05  0.00014665477  0.00024327751  0.00015775142  0.00023943636  0.00039718778
wall                                   0   0.0092966584   0.0092966584              0    0.015178218    0.015178218
-------------------------- -------------- -------------- -------------- -------------- -------------- --------------
net                          0.001932869    0.010914064    0.012846933   0.0031557045    0.017818879    0.020974584
```

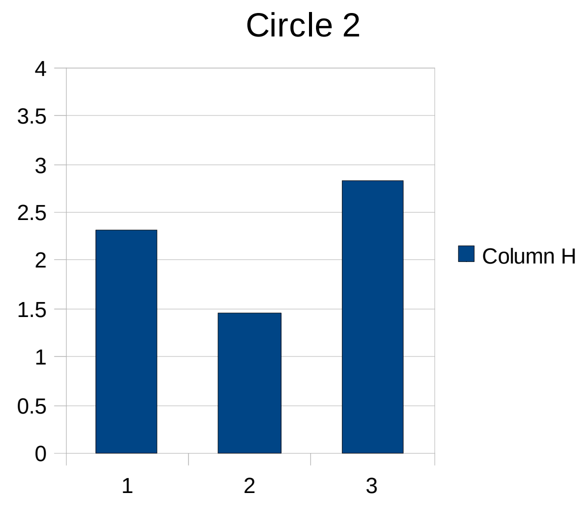
Comparison of the 3 Variations in 3D

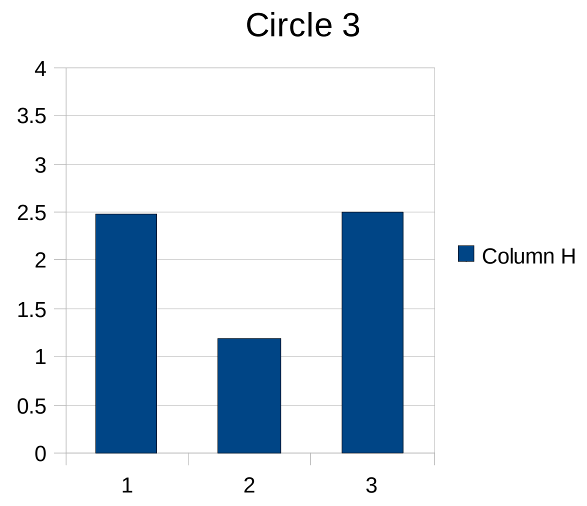












Circle		Void Frac
	1	0.0005113
	2	0.0040906
	3	0.0138058
total		0.0184078