ME 309 Fluid Mechanics Honors

Computational Fluid Dynamics

Final Report

Instructor: Professor Ivan C. Christov

By: Patrick D. Tirtapraja

December 3, 2017

Abstract

The purpose of this experiment and report was to immerse myself in learning computational fluids dynamics (CFD) as part of the honors contract curriculum. The raw data of stream function values were given and this report analyzed its relationship with the flow's velocity as well as its streamline patterns. Several software, such as the Excel Spreadsheet and MATLAB were used to compute and plot the streamlines and velocity vectors given the stream functions values. The Navier–Stokes equation was derived with the additional assumption of irrotational flow on top of its initial assumption of the flow being incompressible and its viscosity remaining constant. The equation of stream function, ψ, was derived and the velocities in the x and y-direction were calculated. With the use of MATLAB, the streamlines, velocity vectors, and velocity contour lines were plotted. Different scenarios depicting various boundary conditions were used to analyze the response of the flow with such changes. It was observed that as the flow approaches the boundary conditions placed, in this scenario, the wall, its stream function decreases and increases again as it distanced itself from the wall. Despite the various assumptions used for CFD in this report, these assumptions led to relatively accurate estimates as to how flows behave in reality. This honors assignment sparked an interest within me to learn more about the theories and concepts of CFD and pushed me to utilize different software programs to accomplish such tasks.

Introduction

The main purpose of this report is to demonstrate the basic understanding of computational fluid dynamics (CFD) and how different programming software could help visualize the fluid patterns created with different boundary conditions.

The foundation of this experiment begins with assumptions that the flow is incompressible and viscosity remains constant. The Navier–Stokes equations and continuity equations were also used to solve for the velocity field as shown below,

$$\vec{V} = \hat{\imath}u + \hat{\jmath}v + \hat{k}w \text{ ... (Eq. 1)}$$

and pressure field p, given that there are sufficient initial and boundary conditions. The continuity equation, expressed in rectangular coordinates, is shown below:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \text{ ... (Eq. 2)}$$

The Navier–Stokes equation, also expressed in Cartesian coordinates is shown below:

$$\rho\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z}\right) = \rho g_x - \frac{\partial \rho}{\partial x} + \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) \text{ ... (Eq. 3a)}$$

$$\rho\left(\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z}\right) = \rho g_y - \frac{\partial \rho}{\partial y} + \mu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right) \text{... (Eq. 3b)}$$

$$\rho\left(\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z}\right) = \rho g_z - \frac{\partial \rho}{\partial z} + \mu\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right) \text{... (Eq. 3c)}$$

From the above equation, it can be seen that the in general, $u, v, w,$ and p depends on coordinates $x, y, z,$ and t.

There are several key points to be taken into consideration when analyzing the two equations above. First, both the mass conservation (Eq. 2) and momentum equations (Eq. 3) are coupled as all unknowns must be solved simultaneously in order for an answer to be obtainable. Second, since the Navier-Stokes equation has a convective acceleration term,

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z}$$

the equation dealt with is nonlinear. Lastly, due to the viscous terms in Equation 3,

$$\mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right)$$

the equation dealt with is also a second-order partial differential equation.

Although it is relatively simple to solve these equations when physical geometries and initial or boundary conditions are simple, most incompressible flows of interest have more complex geometries and would need numerical methods to analyze such problems. Computers are used to approximate solutions to the equations for a range of real-life engineering problems and this is where CFD comes in.

The usage of CFD covers a variety of applications. CFD software packages are used to study the flow field around vehicles as shown in Fig. 1 below. This way the flow through the catalytic converter can be visualized and analyzed in order to develop more effective ones.

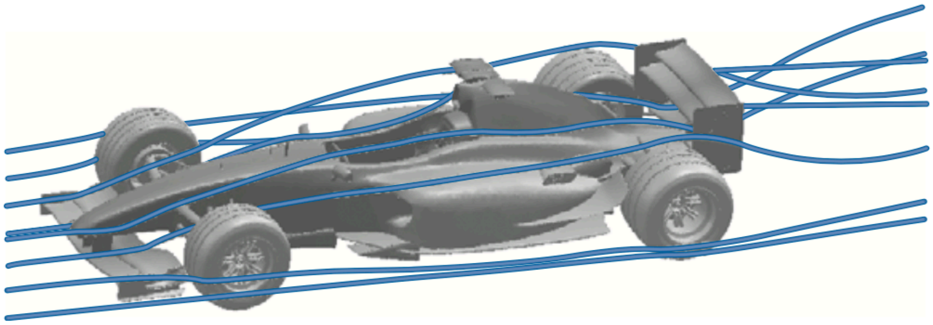


Figure 1: Pathlines around a Formula 1 Car as observed by a CFD Software

In this experiment, a basic application of the CFD was performed by using the Excel Spreadsheet to calculate the values and MATLAB to graphically plot the contours of the data, showing the streamlines and various velocity vectors. This basic application of a numerical method to fluid mechanics problem is when steady, incompressible, inviscid, and 2-D flow were assumed. Despite assuming a number of factors, these assumptions lead to reasonable estimates and flow predictions. The flows can be modeled with the following Laplace equation:

$$\frac{\partial^2\psi}{\partial x^2}+\frac{\partial^2\psi}{\partial y^2}=0 \text{ … (Eq. 4)}$$

where ψ is the stream function. By approximating each differential with Taylor series, the following numerical approximation is derived:

$$\frac{\psi_{i+1,j}+\psi_{i-1,j}}{h^2}+\frac{\psi_{i,j+1}+\psi_{i,j-1}}{h^2}-4\frac{\psi_{i,j}}{h^2}=0 \text{ … (Eq. 5)}$$

where h is the step size in either the i or j direction

$\psi_{i,j}$ is the value of the stream function at the ith and jth location of x and y direction respectively

By simplifying the (Eq. 5), the following equation is obtained:

$$\psi_{i,j}=\frac{1}{4}(\psi_{i,+1\ j}+\psi_{i-1,j}+\psi_{i,j+1}+\psi_{i,j-1} \text{ … (Eq. 6)}$$

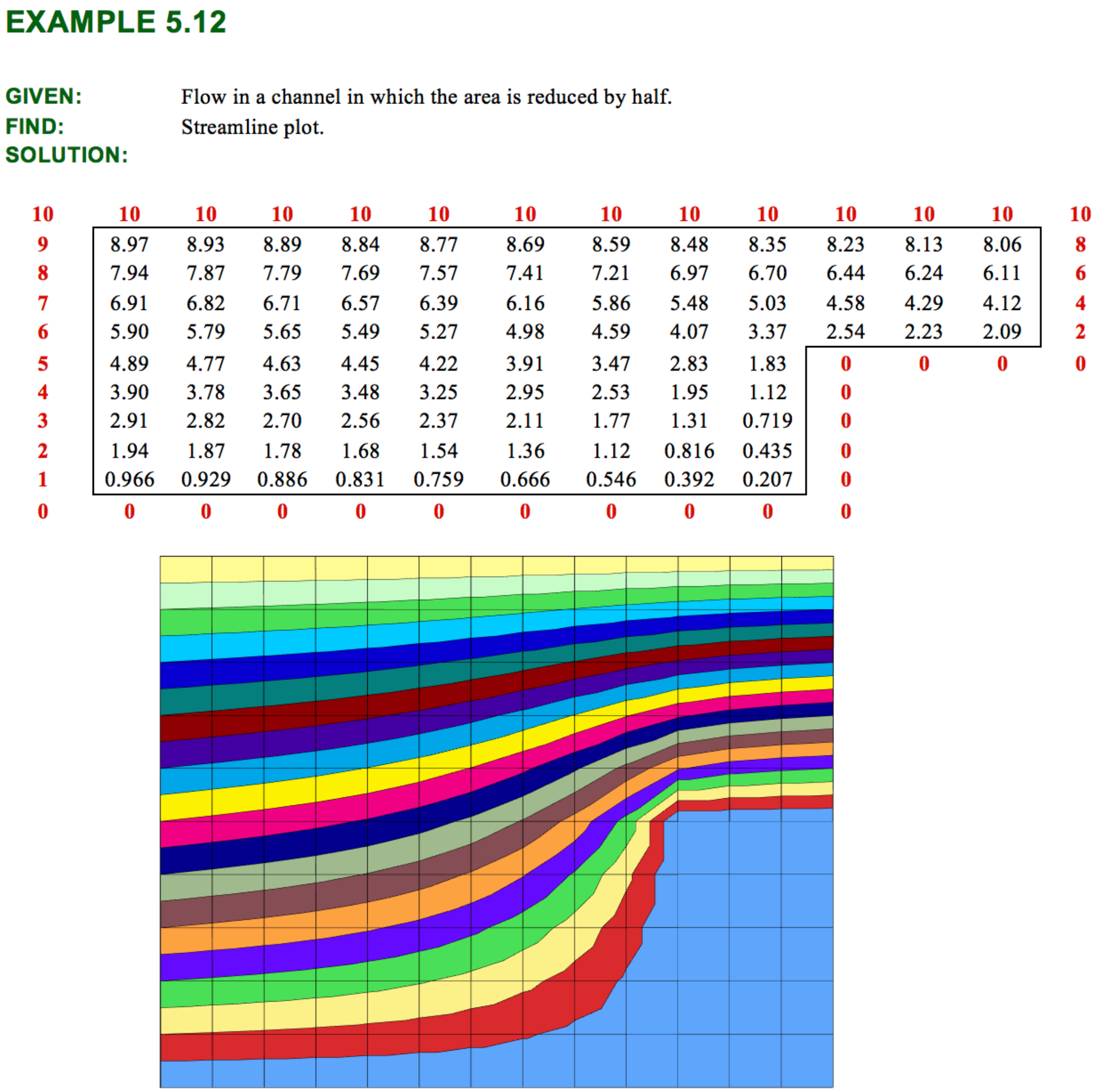
which shows that a stream function at a specific location is the average of its four neighboring coordinates.

By implementing this concept with a given Excel spreadsheet containing the stream function data as well as boundary conditions, a graph was plotted and analyze. By analyzing the streamlines and flow separation location, velocity and pressure values at a specific location can be predicted and calculated. The relationship between the different factors was then able to be distinguished

Methods and Materials

The materials used in this experiment were two software: Excel's Spreadsheet and MATLAB.

The following data was given in an Excel Spreadsheet as shown below:



EXAMPLE 5.12

GIVEN: Flow in a channel in which the area is reduced by half.
FIND: Streamline plot.
SOLUTION:

10	10	10	10	10	10	10	10	10	10	10	10	10	10
9	8.97	8.93	8.89	8.84	8.77	8.69	8.59	8.48	8.35	8.23	8.13	8.06	8
8	7.94	7.87	7.79	7.69	7.57	7.41	7.21	6.97	6.70	6.44	6.24	6.11	6
7	6.91	6.82	6.71	6.57	6.39	6.16	5.86	5.48	5.03	4.58	4.29	4.12	4
6	5.90	5.79	5.65	5.49	5.27	4.98	4.59	4.07	3.37	2.54	2.23	2.09	2
5	4.89	4.77	4.63	4.45	4.22	3.91	3.47	2.83	1.83	0	0	0	0
4	3.90	3.78	3.65	3.48	3.25	2.95	2.53	1.95	1.12	0			
3	2.91	2.82	2.70	2.56	2.37	2.11	1.77	1.31	0.719	0			
2	1.94	1.87	1.78	1.68	1.54	1.36	1.12	0.816	0.435	0			
1	0.966	0.929	0.886	0.831	0.759	0.666	0.546	0.392	0.207	0			
0	0	0	0	0	0	0	0	0	0	0			

Figure 1: Excel Raw Data

Initially, the experiment would start with replicating the data above in MATLAB. However, certain functions, such as Excel's ability to automatically iterate its tables' values are not available in MATLAB and implementing the necessary partial differentials would take longer time deviating from the purpose of the experiment itself.

As a trial and to get used to the features of Excel, the raw data was first remade onto a blank Excel Spreadsheet. The boundary conditions (0's and 10's) were then placed at the desired locations then each of the cells in the spreadsheet were equated to the average of its four neighboring cells. However, this will cause an error of circular calculation where one cell would be referring to a value that does not exist beforehand. This is where the Excel iterate feature comes in under *Tools/Options/Calculation*. The values were then iterated until the variations in values is zero or trivial. The surface plot was then plotted according to the attained data.

The data was then imported to MATLAB and the contour lines were plotted. The velocity vector was then calculated by first equation the velocity in the x-direction, denoted as V_x, and the velocity in the y-direction, denoted as V_y. Additional rows and columns were added if necessary for calculations of either velocities.

Given that the flow is incompressible, 2-D, and irrotational, the velocity components, V_x and V_y can be derived in terms of the stream function, ψ as shown below:

$$V_x = \frac{\partial \psi}{\partial y} \text{ … (Eq. 7a)}$$

$$V_y = -\frac{\partial \psi}{\partial x} \text{ … (Eq. 7b)}$$

where ∂x and ∂y were initialized as equal to 0.1.

The velocity at a specific location was then calculated by using the following formula:

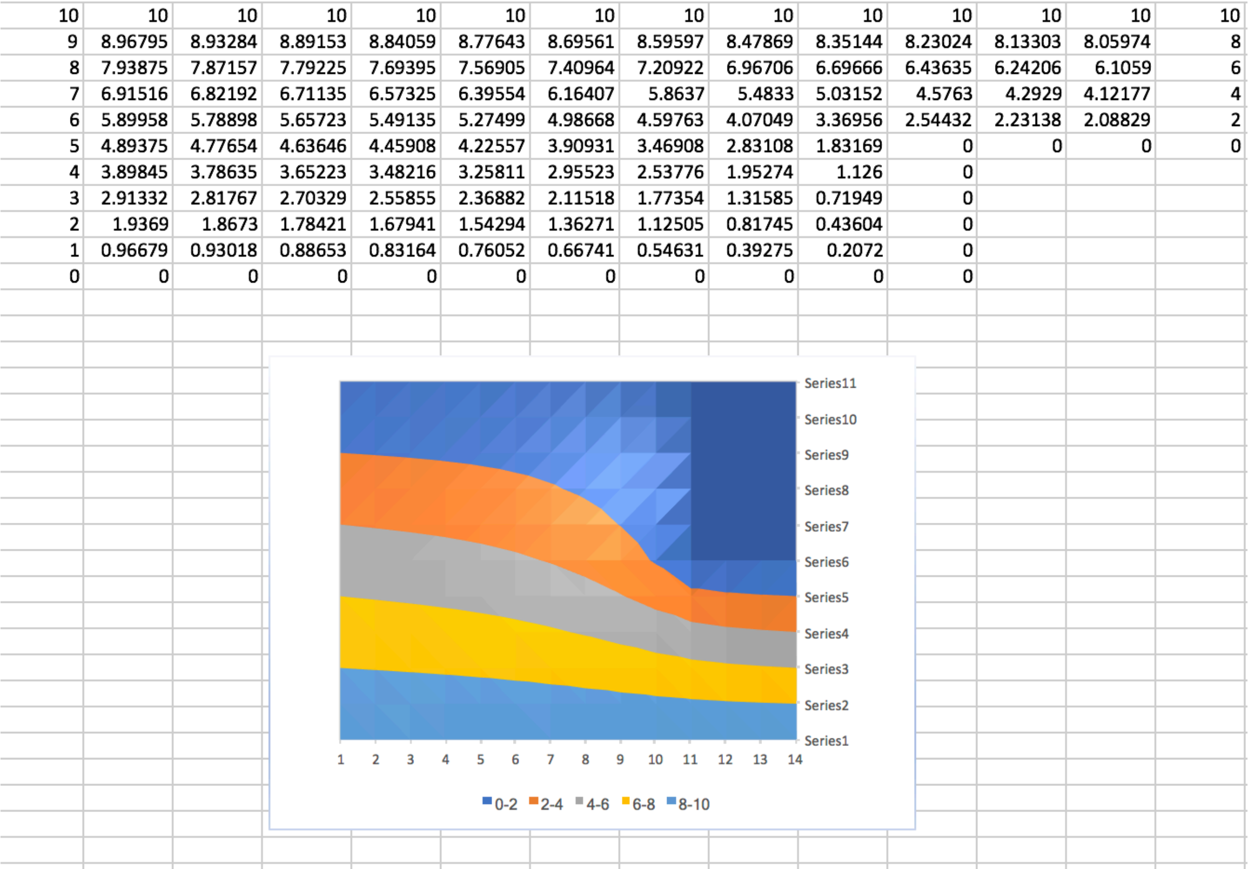
$$V_{x,y} = \sqrt{{V_x}^2 + {V_y}^2} \text{ … (Eq. 8)}$$

and for visualization purposes was then plotted by using the *quiver* function of MATLAB, representing its velocity directions.

Additional pathways and wall shapes and sizes were then analyzed by manipulating the boundary conditions.

Results

This section will present the results obtained throughout this experiment. The figure below shows the completion of the duplicated data of the excel sheet.



10	10	10	10	10	10	10	10	10	10	10	10	10	10
9	8.96795	8.93284	8.89153	8.84059	8.77643	8.69561	8.59597	8.47869	8.35144	8.23024	8.13303	8.05974	8
8	7.93875	7.87157	7.79225	7.69395	7.56905	7.40964	7.20922	6.96706	6.69666	6.43635	6.24206	6.1059	6
7	6.91516	6.82192	6.71135	6.57325	6.39554	6.16407	5.8637	5.4833	5.03152	4.5763	4.2929	4.12177	4
6	5.89958	5.78898	5.65723	5.49135	5.27499	4.98668	4.59763	4.07049	3.36956	2.54432	2.23138	2.08829	2
5	4.89375	4.77654	4.63646	4.45908	4.22557	3.90931	3.46908	2.83108	1.83169	0	0	0	0
4	3.89845	3.78635	3.65223	3.48216	3.25811	2.95523	2.53776	1.95274	1.126	0			
3	2.91332	2.81767	2.70329	2.55855	2.36882	2.11518	1.77354	1.31585	0.71949	0			
2	1.9369	1.8673	1.78421	1.67941	1.54294	1.36271	1.12505	0.81745	0.43604	0			
1	0.96679	0.93018	0.88653	0.83164	0.76052	0.66741	0.54631	0.39275	0.2072	0			
0	0	0	0	0	0	0	0	0	0	0			

Figure 2: Remade Raw Excel Data

The following is the MATLAB code to import the data and plot the contour plot.

```
%%MATLAB Code to plot streamline contours

clear all

%streamline plot
psi = ...
[8.97   8.93    8.89    8.84    8.77    8.69    8.59    8.48    8.35    8.23    8.13    8.06;
7.94    7.87    7.79    7.69    7.57    7.41    7.21    6.97    6.70    6.44    6.24    6.11;
6.91    6.82    6.71    6.57    6.39    6.16    5.86    5.48    5.03    4.58    4.29    4.12;
5.90    5.79    5.65    5.49    5.27    4.98    4.59    4.07    3.37    2.54    2.23    2.09;
4.89    4.77    4.63    4.45    4.22    3.91    3.47    2.83    1.83    0       0       0;
3.90    3.78    3.65    3.48    3.25    2.95    2.53    1.95    1.12    0       0       0;
2.91    2.82    2.70    2.56    2.37    2.11    1.77    1.31    0.719   0       0       0;
1.94    1.87    1.78    1.68    1.54    1.36    1.12    0.816   0.435   0       0       0;
0.966   0.929   0.886   0.831   0.759   0.666   0.546   0.392   0.207   0       0       0];

%adding a row of zero at the bottom for boundary condition
psi = [psi; zeros(1,12)];
%additional column
add_col = [8.00; 6.0; 4.0; 2.0; 0; 0; 0; 0; 0; 0];
psi = [psi add_col];
```

```
%explain if add_col is absolute value instead 10 8 6

contour(flipud(psi), 'ShowText', 'on')
colorbar
```

which resulted into the following graph:

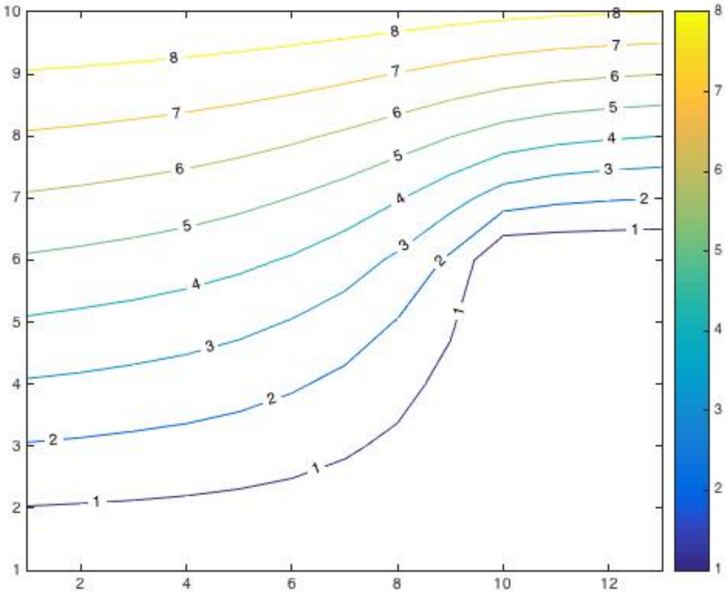


Figure 3: MATLAB plotted streamline

Based on the stream function values, a matrix for both V_x and V_y was calculated by implementing the following lines of code:

```
%assuming changes in x-dir = 0.1 units
dx = 0.1;
%assuming changes in y-dir = 0.1 units
dy = 0.1;

[m,n] = size(psi);

%Calculating Vx
for i = 1:m-1
   for j = 1:n-1
      Vx(i,j)= (psi(i, j+1) - psi(i, j)) / dy;
   end
end

%Calculating Vy
for i = 1:m-1
   for j = 1:n-1
      Vy(i,j)= -((psi(i+1, j) - psi(i, j))) / dx;
   end
end
```

To show the velocity vectors present in the set-up the *quiver* function was used as follows:

```
quiver(Vx, Vy)
```

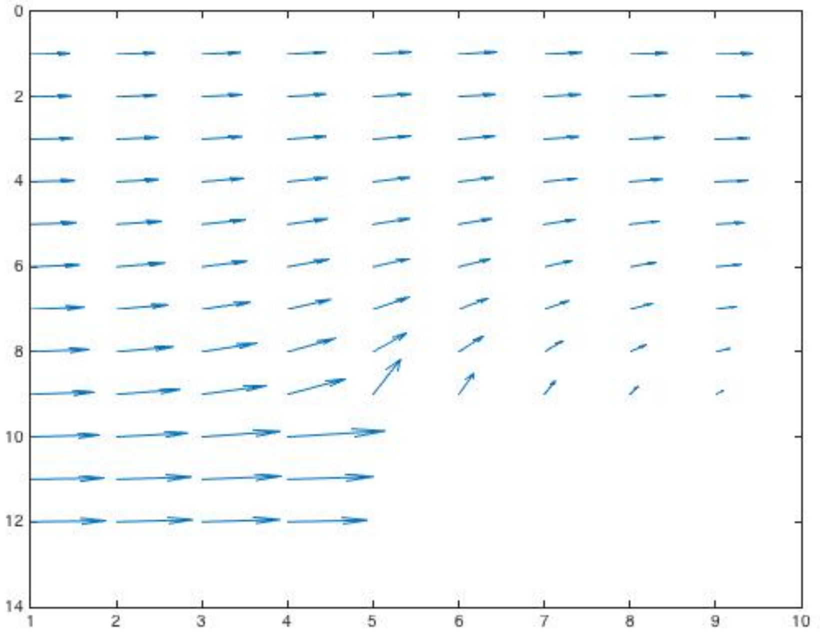


Figure 3: Velocity vector plot by plotting V_x and V_y

After the values of V_x and V_y were obtained, by using (Eq. 8), the velocity vector was calculated. The velocity was then plotted as a contour map to represent the speed of the system by using the following code:

```
%Calculating the veclocity at a given position
for i = 1:m-1
   for j = 1:n-1
      V(i,j) = sqrt(Vx(i,j).^2 + Vy(i,j).^2);
   end
end

contour(flipud(V), 'ShowText', 'on')
colorbar
```

The graph obtained was shown below:

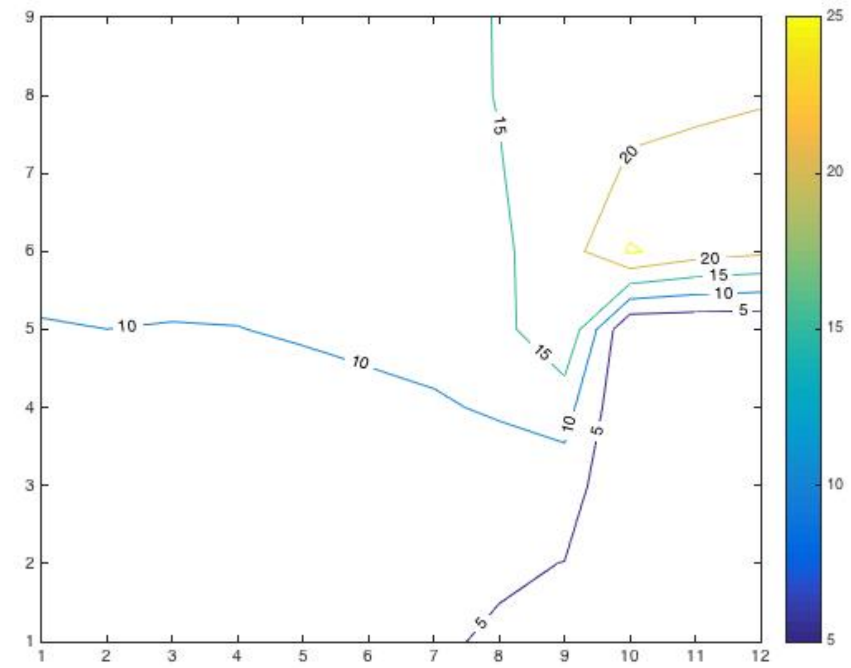
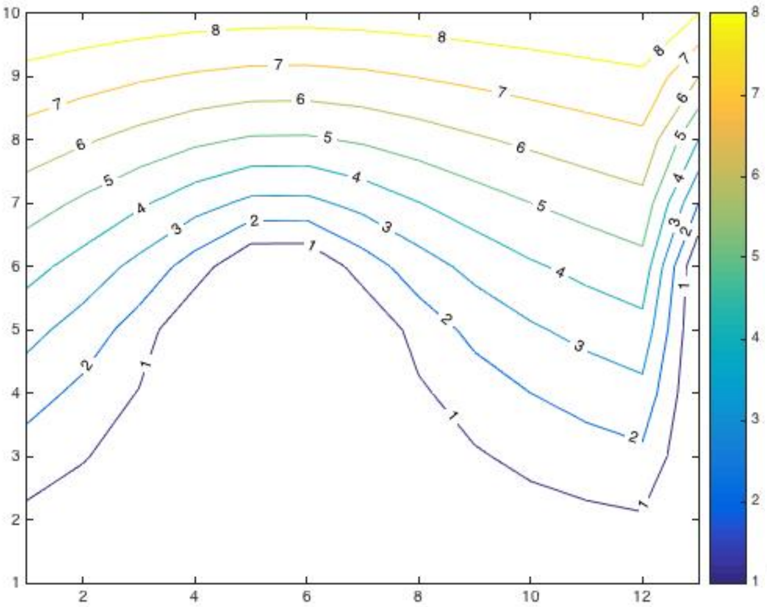


Figure 4: Velocity Contour Lines

With Figure 4 above, it is noticeable that the speeds decreased as it is closer to the wall (where it increases as the distance grew further away from the wall.

The following are some examples of different wall shapes used to analyze its streamlines and velocity contour lines.

Additional Examples #1:
A boulder located at the center.



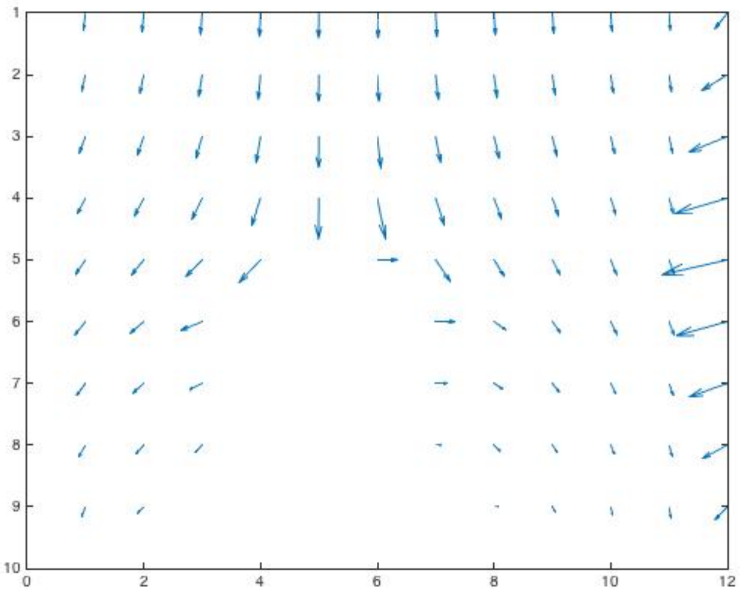


Figure 5a: Streamline with a boulder at the center

Figure 5b: Velocity vector

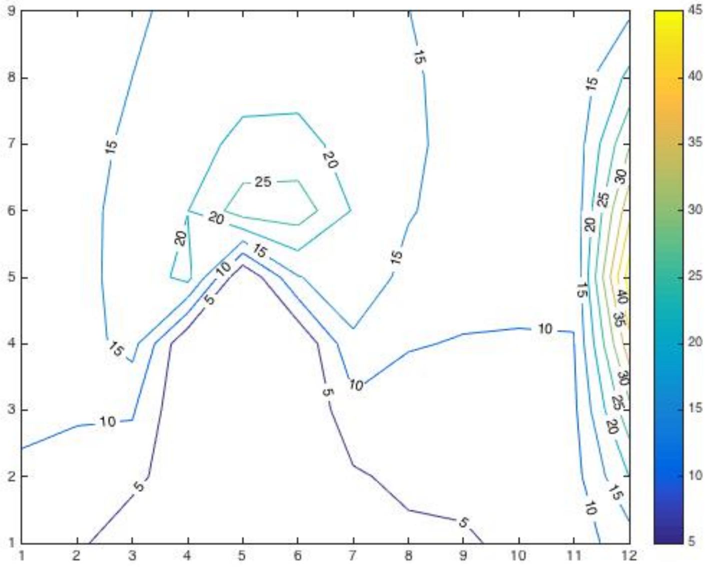


Figure 5c: Velocity Contour Lines

Figure 5 a, b, c shows the streamline, velocity vector, and velocity contour lines if there was a boulder located at the bottom center of the set-up as indicated by the white spaces in Figure 5a.

Additional Examples #2:
Two walls, one at the bottom left, the other one located at the top right.

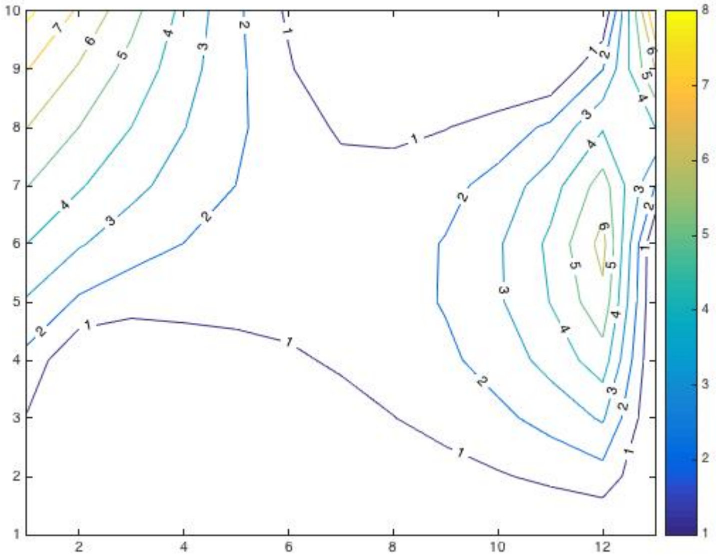


Figure 6a: Streamline with a boulder at the center

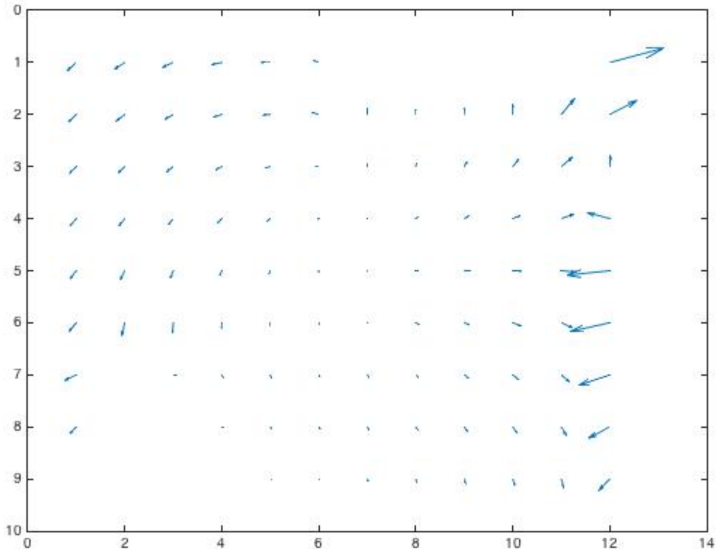


Figure 6b: Velocity vector

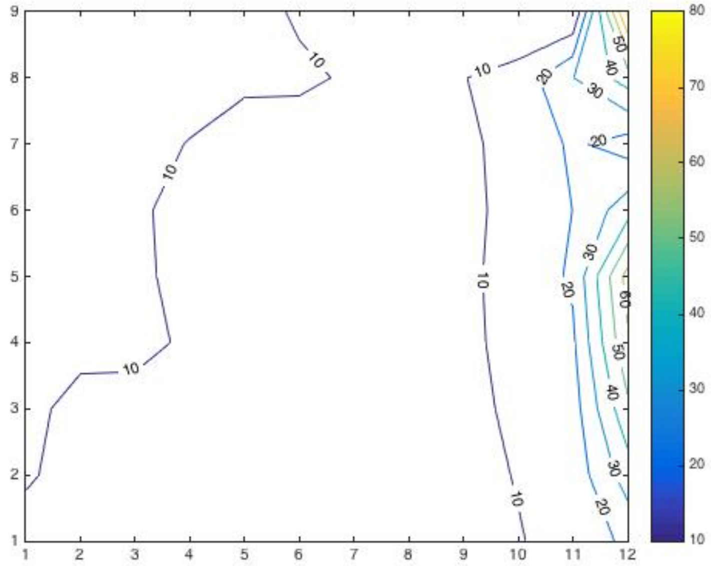


Figure 6c: Velocity Contour Lines

Similarly, Figure 6 a, b, c also shows the streamline, velocity vector, and velocity contour lines if the wall placements changed to the specified description above.

Discussion

The streamlines, velocity vectors, and velocity contour lines shown in Figures 5 and 6 were expected. It can be observed that as the flow travels closer to the wall, its velocity decreases. As the flow passed the obstruction, the further away the flow is to the object, the higher its velocity becomes until it reaches its free stream velocity.

By observing the individual set-ups contour lines, it can be seen that the velocity does not follow the streamlines' path. The circular shaped contour lines in Figure 4 and 5c indicates that as the flow came in contact with an object, some of the flow may flowed backwards.

However, there are several limitations to take note of in this experiment. The raw data Excel Spreadsheet containing the stream function values assumed that the velocity will increase with increments of two up till 10 right after the flow reaches a specific distance from the first point of contact with the wall. However, in real life, this may not be the case as the values of the stream functions may not increment with such discrete values.

Furthermore, the same idea could lead to uncertainties and errors when calculating the velocities in subsequent example scenarios #1 and #2. It was assumed that the stream functions would achieve its initial max flow rate eventually. However, such assumptions could be incorrect as the velocity could indeed slow down to a stop after it came in contact with the object itself. To minimize this problem, more detailed in real life analysis should be conducted to obtain more accurate results.

Conclusion

In conclusion, this experiment and report has been a great way to immersed myself in the basics of computation fluids dynamics. On top of the materials covered in the course, I managed to learn more about stream functions and its relationship with velocity, both magnitude and direction, alongside with its different interaction with boundary conditions. As the flow approaches the wall, the stream functions decreased until the flow distanced itself from the wall. Through this honors assignment, I was able to utilize the Excel Spreadsheet, MATLAB, and fluids concepts to learn about how a flow behaves through the use of CFD. I understood that CFD could be a relatively accurate estimate as to how flow will behave given that there are boundary conditions involved.

References

[1] Fox, R. W., McDonald, A. T., & Pritchard, P. J. (2012). Introduction to fluid mechanics (8th ed.). New Delhi, India: J. Wiley.