A 2D FLOW SIMULATOR USING FINITE DIFFERENCE METHODS

DEPARTMENT OF MECHANICAL ENGINEERING

Author(s): jhedin

Checked by:

Ref: CC-NN-RR
Filename: flow_simulator.docx
Date: 18-Dec-2014
Total Pages: 12

Abstract

This report highlights elements from the implementation of a 2D flow simulator using finite difference techniques. This was done as a two-step process on a mesh of a system. First, the portions of the velocities that don't depend on the pressures were calculated, then, the pressures are optimized to minimize the change in momentum between adjacent cells. This is done for each time step. The simulation is visualized by drawing a contour plot of the stream function, aka, the stream lines of the flow.

The simulator was used to show the effects of varying parameters on the Karman Vortex Street problem, a flow obstructed by a cylinder. Under low Reynolds numbers, the flow proved to be inviscid, closing right away behind the obstacle, while under higher Reynolds numbers, the flow separates behind the obstacle, forming vortices. Under very high Reynolds numbers, the simulator proves insufficient to model the flow.

Table of Contents

1 Introduction 1
1.1 Resources Used 1

2 Discussion 1
2.1 Implementation 1
2.1.1 Boundaries 2
2.1.2 Obstacles 3
2.1.3 Rendering 3
2.2 Results 3
2.2.1 Laminar Flow 3
2.2.2 Flow Separation 4
2.2.3 Numerical Instability 6

3 Conclusion 8
3.1 Recommendations 9

4 References 9

Table of Figures

Figure 1 Viscous flow 4
Figure 2 Viscous flow with an obvious stagnation point 4
Figure 3 Starting of a turbulent flow 5
Figure 4 Development of vortices 5
Figure 5 Developed vortices 6
Figure 6 Vortices start developing in front of the obstacle 7
Figure 7Very large vortices develop in front of the obstacle 7
Figure 8 Vortices velocities tend towards infinity 8
Figure 9 All values are NAN, making the contours a grid 8

1 INTRODUCTION

This report details the implantation of a 2D flow simulator, using a finite difference approach to numerically integrate the Navier-Stokes equation. This simulator is a step towards creating a simulation for modelling the deformations and forces generated in a sail in wind, which proved to be too ambitious for an undergraduate project.

1.1 RESOURCES USED

A number resources were combined to create this simulator. First and foremost is prior experience in working with finite difference solvers, obtained in Mech 395: Heat and Mass Transfer, though finite differences were used to determine the temperature distribution in finned heat syncs, rather than for fluid flow. In this, the course textbook, *Fundamentals of Heat and Mass Transfer* was used to reference how the boundary conditions work in an analogous system.

A report from the Los Alamos National Laboratory was used heavily for reference equations and derivations of the method used. This included equations for the explicit and implicit steps of the iteration, which were modified to use alternative finite difference approximations. This also provided some reference screenshots, with comparisons to experimental data, allowing for verification of the developed flows. This also suggested implementations for some boundary conditions for system edges and obstacles to the flow.

A slideshow from the Indian Institute of Technology Bombay was used to check and correct the math and indices from the Los Alamos report.

2 DISCUSSION

2.1 IMPLEMENTATION

The implementation makes several key assumptions to simplify the 2D Navier-Stokes equation. These are:

- The flow is incompressible, eg, the density is fixed throughout the system
- The system is adiabatic, eg, there is no transfer of heat energy between the system and the surroundings.

Together, these assumptions require the pressures to be calculated implicitly, and lead to the simplified Navier-Stokes equations:

$$\frac{\partial v}{\partial t} + \frac{\partial uv}{\partial x} + \frac{\partial v^2}{\partial y} = -\frac{\partial P}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right)$$

and

$$\frac{\partial u}{\partial t} + \frac{\partial uv}{\partial y} + \frac{\partial u^2}{\partial x} = -\frac{\partial P}{\partial x} + \nu(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2})$$

This equation is solved by approximating each partial derivate as a finite difference. It is of note that to speed up the calculation, most of the equation doesn't depend on the pressure, and can be calculated separately.

The remaining pressure needs to be calculated implicitly, by optimizing the mass equation,

$$D = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$$

Where D should converge to 0 throughout the system. This is approximated again, by evaluating the partial derivatives as finite differences.

The calculated D field is used to update the pressures, using the gradient descent

$$P' = P' + \frac{\partial P}{\partial D} * D$$

where

$$\frac{\partial D}{\partial p} = 2dt\left(\frac{1}{dx^2} + \frac{1}{dy^2}\right)$$

Using the new pressure field, a set of velocities is calculated using finite differences to evaluate $\frac{\partial P}{\partial x}$ and $\frac{\partial P}{\partial y}$, and combining those with the precalculated sections.

The D field is recalculated using new velocities, and iterated until max(D) is reasonably small.

The solver then continues to the next time step and recalculates the state.

2.1.1 BOUNDARIES

Due to the centered difference approach used to solve the partial derivatives, the boundary conditions are tricky and need to be set extra values outside the implicit and explicit steps. To do this, free-slip conditions were applied to the top and bottom edges, setting the tangent velocity to be the same as its calculated neighbor.

The left side's boundary is set to be a uniform speed, and the right hand side is done similar to the free slip condition, but the normal velocity is copied rather than the tangent one.

2.1.2 OBSTACLES

Obstacles are rasterized onto the velocity meshes at the end of each iteration of the optimizer. This sets the edges and solid body of the obstacle to 0, creating a no-slip boundary condition at the edge. A free-slip condition was also tested but resulted in solver instability for most cases.

2.1.3 RENDERING

The mesh data is rendered in two ways; a set of streamlines, calculated with conrec, and a colour map of the D values. A vector field of the stream function ψ is found as follows

$$\psi_{i,0} = 0$$

$$\psi_{i,j} = \psi_{i,j-1} + dy * u_{i,j}$$

The contours of ψ form streamlines of the system.

2.2 RESULTS

The flow simulator can be used to show a number of different flow conditions. AS an example, a cylinder was placed in the flow to simulate a Karman Vortex Street, in which case eddies are generated on alternating sides of the cylinder, for flows with larger Reynolds numbers. In each case, fluid enters at the left of the screen, and flows to the right, along the given streamlines.

2.2.1 LAMINAR FLOW

At low Reynolds numbers – low speed, high viscosity – viscous effects should dominate the flow, causing the flow to remain laminar, regardless of obstacles placed in the stream. This can be seen in Figure 1 and Figure 2, where the streams close quickly after the cylinder.

It is also of note that Figure 1 has a section with no visible stream lines. This is due to the stream lines being plotted on a linear scale, and thus the change in velocities too small to plot additional streamlines in between. Another point of interest is in Figure 2, where a streamline that (almost) passes through the stagnation point (point where the fluid first touches the cylinder) is visible. This is the center line, which goes almost straight along the middle axis of the cylinder. While it can't be rendered nicely, an ideal streamline of this nature would split at the stagnation point and rejoin at the end of the cylinder.

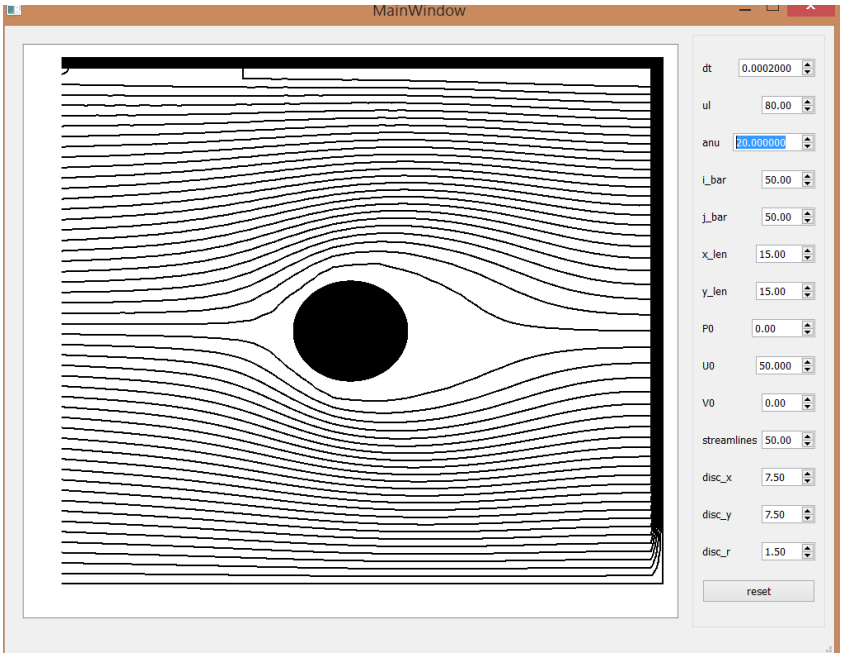

Figure 1 Viscous flow

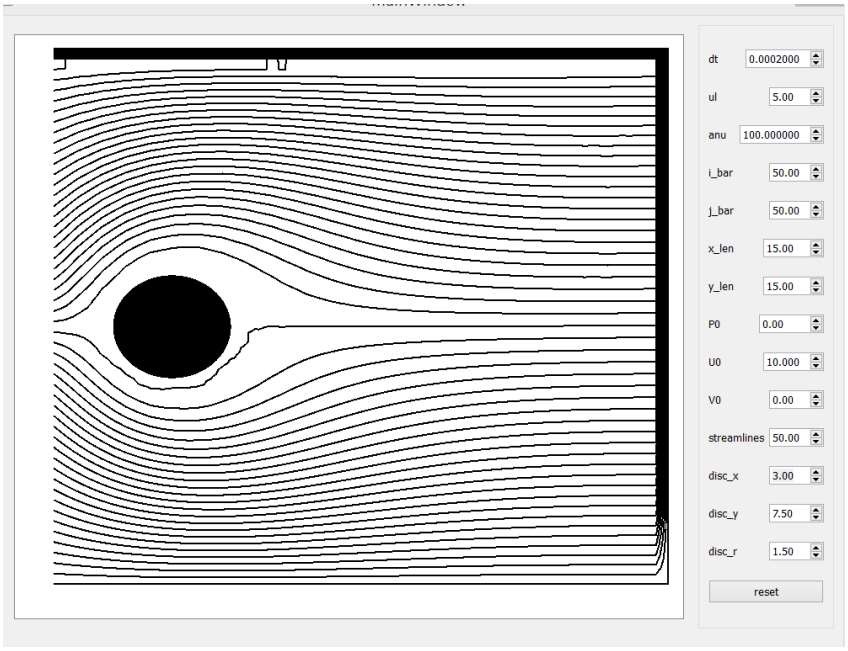

Figure 2 Viscous flow with an obvious stagnation point

2.2.2 Flow Separation

At higher Reynolds numbers – higher speeds, lower viscosities – the cylinder will start to shed vortices. Figure 3 shows the start of the flow, again, there's a stream line that

passes very close to the stagnation point, and thus, it jumps across the cylinder and continues after the obstacle.

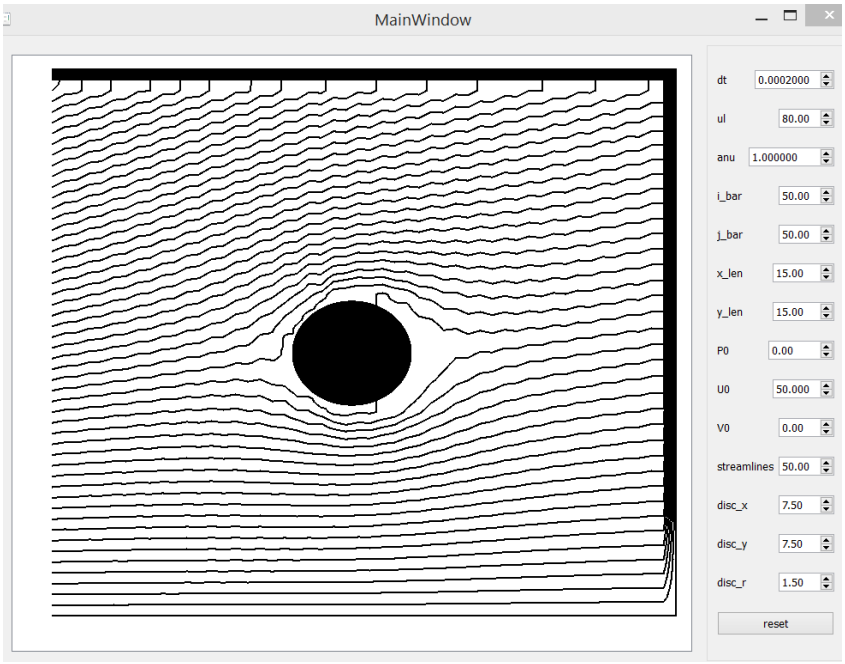

Figure 3 Starting of a turbulent flow

As more time continues, vortices start to form to the right of the obstacle, one on the top, spinning clockwise, and another on the bottom, rotating counter clockwise, as seen in Figure 4.

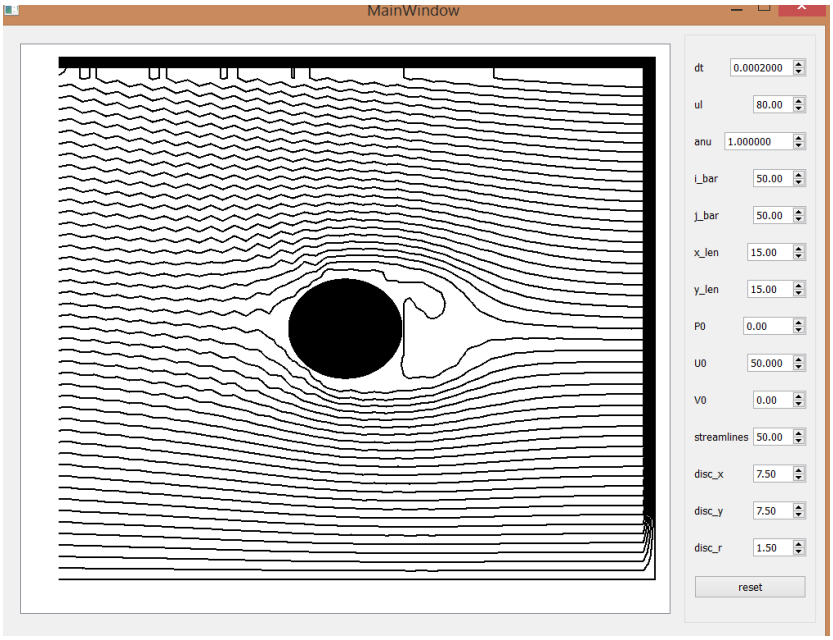

Figure 4 Development of vortices

Continuing in time, the bottom vortex grows large, as shown by the extra streamline inside it in Figure 5. After a longer time, the top vortex would grow instead, as the bottom one flows away.

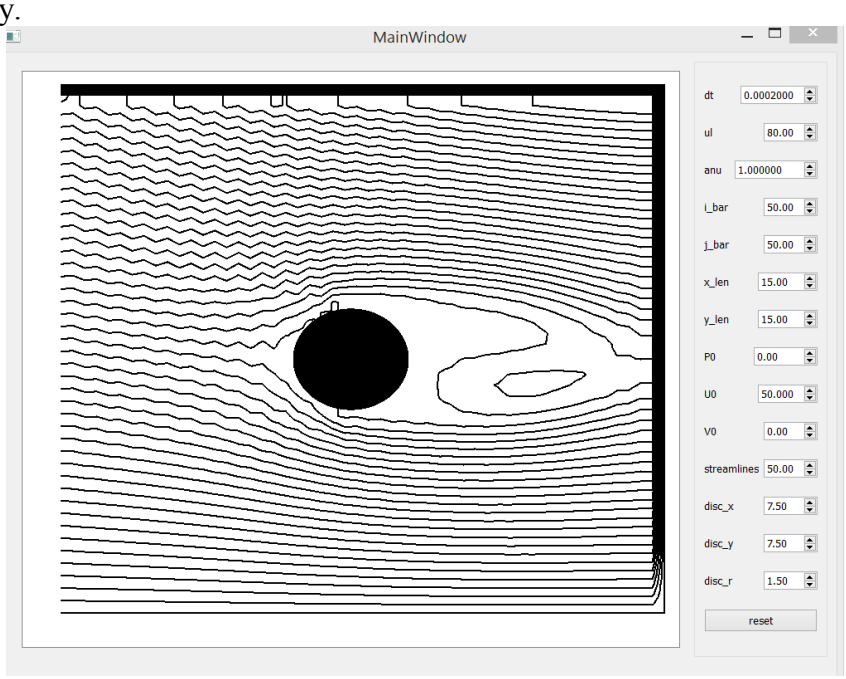

Figure 5 Developed vortices

In each of the above screenshots, the streamlines are much more jagged than the laminar flow ones. This is due to increased numerical instability as the Reynolds number increases.

2.2.3 Numerical Instability

When the Reynolds number gets very high, or the obstacle very large, or the time step too large, or the mesh cell sizes too large, the model loses its numerical stability. As a result of finite difference setup, the issues generally manifest as a shock at the start of the obstacle, which breaks the incompressibility assumption for the flow. Thus, as time goes on from Figure 6 to Figure 8, small vortices start to form at the front of the obstacle, then quickly explode towards infinity, at which point the next calculation results in a field of all NANs, resulting in the grid shown in Figure 9.

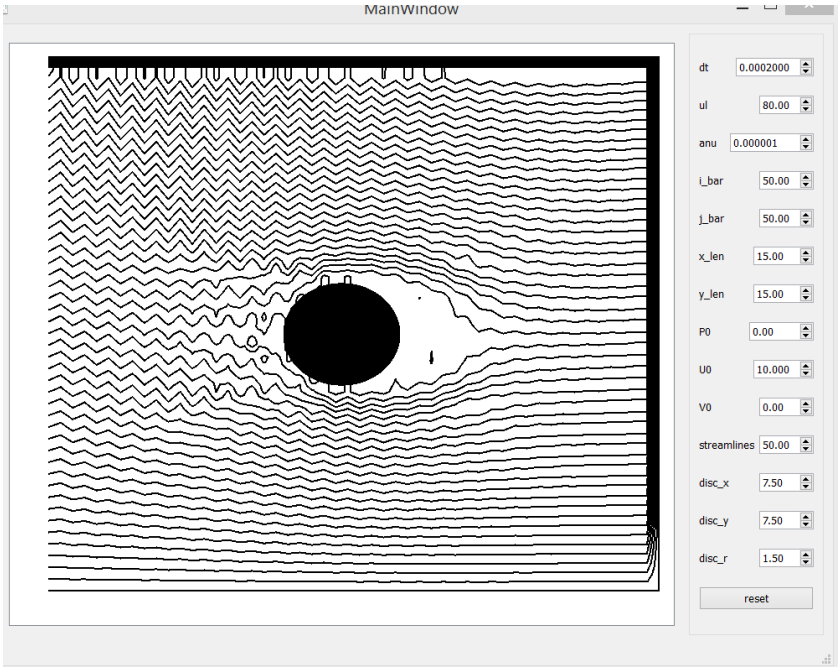

Figure 6 Vortices start developing in front of the obstacle

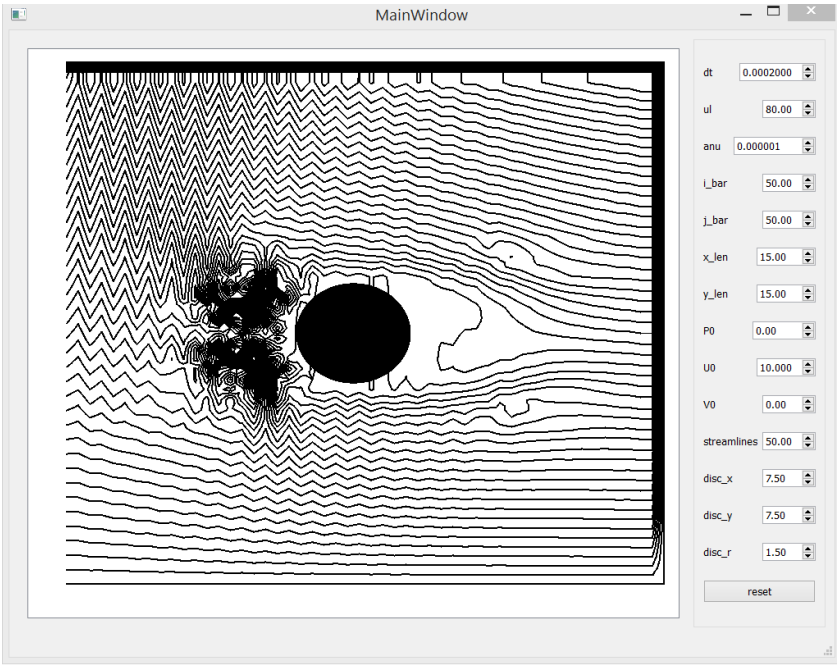

Figure 7Very large vortices develop in front of the obstacle

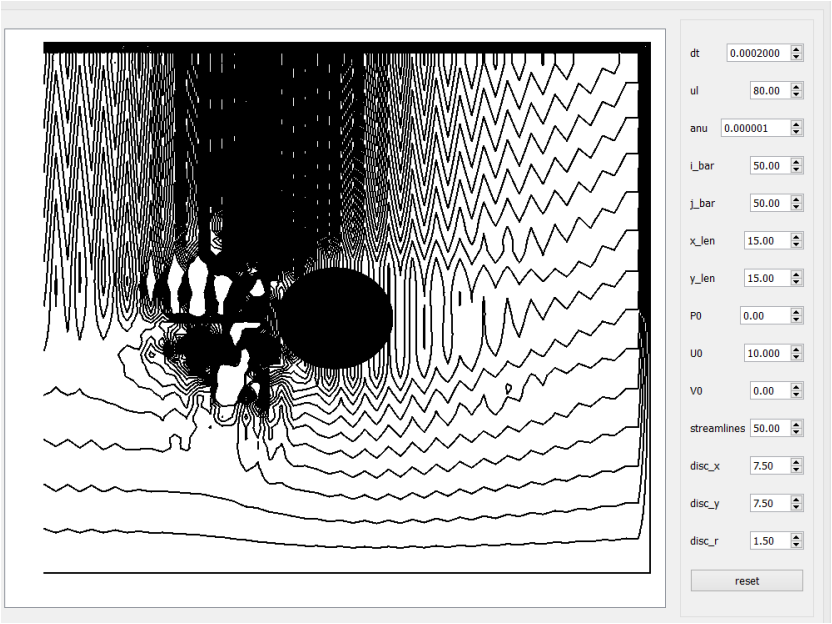

Figure 8 Vortices velocities tend towards infinity

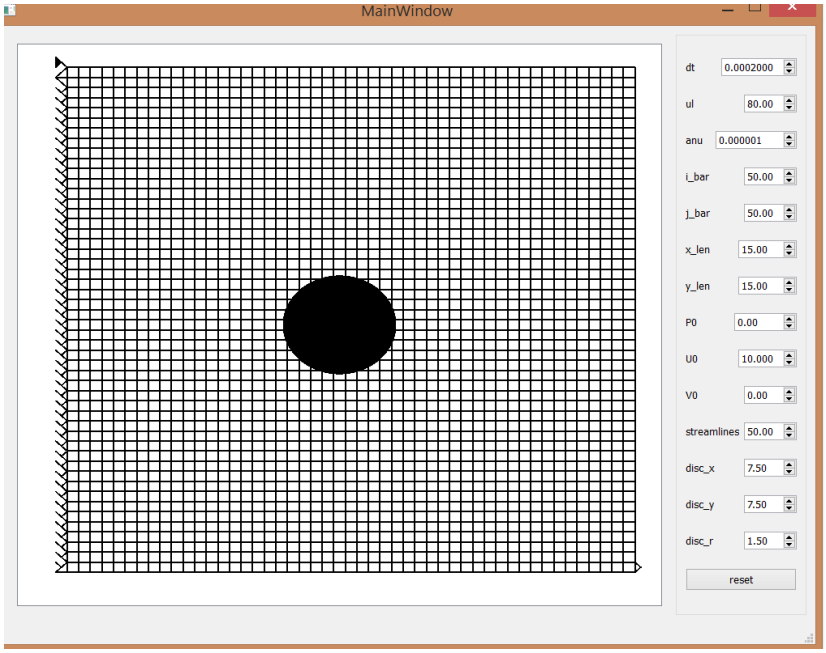

Figure 9 All values are NAN, making the contours a grid

3 CONCLUSION

A 2D flow simulator was implemented, and testing done on a Karman Vortex Street problem, showing the development of inviscid and turbulent flow regimes, along with situations where the model is insufficient to model the flow. Several algorithms were used to visualize the flow, along with knowledge and experience from previous courses.

3.1 RECOMMENDATIONS

The next advancement on this simulator is to add stresses and deflections to the obstacle using finite element analysis. This requires another step to be done before the optimization, calculating pressure differentials across the obstacle, and integrating a new shape for the obstacle. Then, the new shape is rasterized into the mesh.

Other advancements include adding support for mouse driven perturbations to the flow, allowing the user to create waves in the flow, and adding particle visualizations, so the pathlines can be seen, and the rotational nature of the vortices are easier to see.

4 REFERENCES

[1] *Fundamentals of Heat and Mass Transfer 7th Edition,* F Incropera (1981/01/01) URL: http://books.google.ca/books?id=vvyIoXEywMoC&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false

[2] *Introduction to Finite Difference Methods for Numerical Fluid Dynamics,* E. Scannapieco (2007/09/19) URL: http://zofia.sese.asu.edu/~evan/finite_diff2.pdf

[3] *Fundamentals of Finite Difference Methods (FDM),* S. Agarwal (2011/03/02) URL: http://www.leb.eei.uni-erlangen.de/winterakadèmie/2010/report/content/course02/pdf/0203.pdf