Continuum Sensitivity Analysis for Shape Optimization in Incompressible Flow Problems

Aaron M. Turner

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Aerospace Engineering

Mayuresh Patil, Chair

Christopher Roy

Robert Canfield

May 8, 2017

Blacksburg, Virginia

Keywords: Sensitivity Analysis, Design Optimization

Continuum Sensitivity Analysis for Shape Optimization in Incompressible Flow Problems

Aaron M. Turner

(ABSTRACT)

An important part of an aerodynamic design process is optimizing designs to maximize quantities such as lift and the lift-to-drag ratio, in a process known as shape optimization. It is the goal of this thesis to develop and apply understanding of mixed finite element method and sensitivity analysis in a way that sets the foundation for shape optimization. The open-source Incompressible Flow Iterative Solution Software (IFISS) mixed finite element method toolbox for MATLAB developed by Silvester, Elman, and Ramage is used. Meshes are produced for a backward-facing step problem, using built-in tools from IFISS as well as the mesh generation software Gmsh, and grid convergence studies are performed for both sets of meshes along a sampled data line to ensure that the simulations converge asymptotically with increasing mesh resolution. As a preliminary study of sensitivity analysis, analytic sensitivities of velocity components along the backward-facing step data line to inflow velocity parameters are determined and verified using finite difference and complex step sensitivity values. The method is then applied to pressure drag calculated by integrating the pressure over the surface of a circular cylinder in a freestream flow, and verified and validated using published simulation data and experimental data. The sensitivity analysis study is extended to shape optimization, wherein the shape of a circular cylinder is altered and the sensitivities of the pressure drag coefficient to the changes in the cylinder shape are determined and verified.

Continuum Sensitivity Analysis for Shape Optimization in Incompressible Flow Problems

Aaron M. Turner

(GENERAL AUDIENCE ABSTRACT)

When looking at designing an aircraft, it is important to consider the forces air flow exerts on the wings. The primary forces of interest for aerodynamic analysis are lift, which generally acts upward perpendicular to the flow of air, and drag, which opposes the motion of the wing through the air. Optimization is the process of developing a design in such a way that a specific quantity, such as lift or drag, is either maximized or minimized. Many methods exist of predicting the behavior of air flow, and various methods of optimization exist which take already existing predictive software and progressively alter the design to try to meet the minimized or maximized objective. This thesis outlines a multi-step effort to modify an open source software such that it could be used for design optimization.

Acknowledgments

I have many people to thank for the progress I have made in my research, though I know I can't include everyone.

I would like to thank Dr. Patil for his steady guidance through our exploration of the aforementioned topics; without his patience, enthusiasm, and curiosity, this work would not have been possible. Through everything he has taught me, he has awakened my own curiosity and sent me down a path I foresee following for the rest of my life, and I am forever grateful.

I would like to thank Dr. Canfield and Dr. Roy, for helping to mold me and expand my engineering abilities and understanding in my time as both an undergraduate and a graduate student at Virginia Tech. Their continued insight has helped me to not lose sight of the elements central to graduate research and my continuously evolving role as a student.

I also thank my fiancée Chelsea for supporting me and reminding me of my abilities and how far I have come over the course of my research, even as I stumbled in the dark. She has been a force working to make me a better person in every way imaginable, and I am excited at the idea of the two of us soon spending the rest of our lives together. Though I have said it many times over the years, I will continue to say it every day: I love you.

I would like to thank my family, especially my parents, for having been so supportive of me since I first learned about math and science. I was a curious and stubborn child then and, both with and independent of their help, that fact has not changed even as I have grown. I hope I am, or will one day be, the kind of person they can be proud of. Thank you, and I love you all.

I am grateful for my fellow Trefoil Academy camp staff for the lessons they have taught me, including how to manage stress and time and how to be resilient. Adapt, Improvise, Overcome, Mischief Managed.

Aaron Turner

Contents

1 Introduction **1**

2 Theory **3**

2.1 IFISS 4

2.1.1 Systems of Equations 5

2.1.2 Boundary Conditions 6

2.1.3 Mixed Finite Elements and Mesh Generation 7

2.1.4 Stokes Iteration 11

2.1.5 Picard Iterations 12

2.1.6 Newton Iterations 14

2.2 Sensitivity Analysis Methods 15

2.2.1 Analytic Sensitivity Calculation 16

2.2.2 Finite Difference Sensitivity Calculation 17

2.2.3 Complex Step Sensitivity Calculation 18

3 Results **20**

3.1 Analytic Sensitivity to Value 20

3.2 Continuum Analytic Sensitivity to Shape Parameter 29

4 Conclusion and Future Work **44**

4.1 Conclusion 44

4.2 Future Work 45

Appendices **46**

A Backward-Facing Step Meshes **47**

B Cylinder Meshes **51**

5 Bibliography **57**

List of Figures

2.1 Representations of the $Q_1 - P_0$ (top left), $Q_1 - Q_1$ (top right), $Q_2 - P_1$ (bottom left), and $Q_2 - Q_1$ (bottom right) mixed finite elements; for flow variables, the blue circles indicate nodes for u- and v-components of velocity while red diamonds indicate nodes for pressure and the red 'x' represents nodes at which both pressure and its spatial derivatives are stored. 7

2.2 Domain for the backward-facing step cases, with the origin at the edge of the step; flow enters through the inlet (cyan), passes the data line (green), and exits the domain through the outlet (purple). 9

2.3 Domain for the circular cylinder cases, with the origin at the center of the cylinder; the fluid enters through the inlet (cyan), flows over the cylinder (orange) and past the data line (green), and exits the domain through the outlet (purple). 10

2.4 Representation of changing cylinder designs; orange indicates the circular cylinder design, purple indicates the "wide" cylinder design, and cyan indicates the "narrow" cylinder design. 10

2.5 Graphic of sensitivity analysis categories. 16

3.1 Uniform coarse backward-facing step mesh generated in IFISS for sensitivity analysis. 21

3.2 Non-uniform coarse backward-facing step mesh generated in Gmsh for sensitivity analysis. . . 21

3.3 Vector field of backward-facing step flow. 22

3.4 Vector field of backward-facing step flow wake. 22

3.5 Plot of u-velocity components at x=1.0 for coarse (pink), medium (green), and fine (blue) uniform meshes. 23

3.6 Plot of v-velocity components at x=1.0 for coarse (pink), medium (green), and fine (blue) uniform meshes. 23

3.7 Velocity profiles for uniform and non-uniform meshes for flow over a backward-facing step. . . 24

3.8 Velocity profiles for the uniform backward-facing step problem with increasing mesh refinement. 24

3.9 Mesh convergence study of the uniform backward-facing step problem with increasing mesh refinement. 25

3.10 Plot of u-velocity sensitivities for coarse (circles), medium (triangles), and fine (x's) uniform structured meshes at x=1; the section of the greatest sensitivity magnitudes is expanded in the bottom right. 26

3.11 Plot of v-velocity sensitivities for coarse (circles), medium (triangles), and fine (x's) uniform structured meshes at x=1; the section of the greatest sensitivity magnitudes is expanded in the top right. 27

3.12 Coarse circular cylinder mesh generated in Gmsh for shape optimization cases. 29

3.13 Expanded view of coarse circular cylinder mesh generated in Gmsh for shape optimization cases. 30

3.14 Coarse wide elliptical cylinder mesh generated in Gmsh for shape optimization cases. 30

3.15 Coarse narrow elliptical cylinder mesh generated in Gmsh for shape optimization cases. . . . 30

3.16 Circular cylinder coarse mesh vector field. 31

3.17 Expanded view of cylinder in coarse mesh vector field. 31

3.18 Vortical structures in the cylinder wake. 32

3.19 Flow wake far downstream of the cylinder. 32

3.20 Distribution of u-velocity downstream of the cylinder at x = 0.3. 33

3.21 Distribution of v-velocity downstream of the cylinder at x = 0.3. 33

3.22 Plot of decreasing pressure drag coefficient with increasing a. 35

3.23 Velocity along centerline ahead of circular cylinder. 39

3.24 Expanded view of circular cylinder centerline velocity profile, with focus on the interface of mesh sections. 39

3.25 Expanded view of circular cylinder centerline velocity profile, with focus on velocity close to the leading edge. 40

3.26 Plot of pressure drag coefficient sensitivity approximation versus step size. 42

A.1 Coarse uniform structured mesh produced in IFISS for the backward-facing step grid convergence test. 47

A.2 Medium uniform structured mesh produced in IFISS for the backward-facing step grid convergence test. 48

A.3 Fine uniform structured mesh produced in IFISS for the backward-facing step grid convergence test. 48

A.4 Coarse non-uniform structured mesh produced in IFISS for the backward-facing step grid convergence test. 49

A.5 Medium non-uniform structured mesh produced in IFISS for the backward-facing step grid convergence test. 49

A.6 Fine non-uniform structured mesh produced in IFISS for the backward-facing step grid convergence test. 50

B.1 Coarse circular cylinder mesh generated in Gmsh for shape optimization cases. 51

B.2 Medium circular cylinder mesh generated in Gmsh for shape optimization cases. 51

B.3 Fine circular cylinder mesh generated in Gmsh for shape optimization cases. 52

B.4 Ultra-fine circular cylinder mesh generated in Gmsh for shape optimization cases. 52

B.5 Ultra-fine+ circular cylinder mesh generated in Gmsh for shape optimization cases. 52

B.6 Coarse wide elliptical cylinder mesh generated in Gmsh for shape optimization cases. 53

B.7 Medium wide elliptical cylinder mesh generated in Gmsh for shape optimization cases. 53

B.8 Fine wide elliptical cylinder mesh generated in Gmsh for shape optimization cases. 53

B.9 Ultra-fine wide elliptical cylinder mesh generated in Gmsh for shape optimization cases. . . . 54

B.10 Ultra-fine+ wide elliptical cylinder mesh generated in Gmsh for shape optimization cases. . . 54

B.11 Coarse narrow elliptical cylinder mesh generated in Gmsh for shape optimization cases. . . . 54

B.12 Medium narrow elliptical cylinder mesh generated in Gmsh for shape optimization cases. . . . 55

B.13 Fine narrow elliptical cylinder mesh generated in Gmsh for shape optimization cases. 55

B.14 Ultra-fine narrow elliptical cylinder mesh generated in Gmsh for shape optimization cases. . . 55

B.15 Ultra-fine+ narrow elliptical cylinder mesh generated in Gmsh for shape optimization cases. . 56

List of Tables

3.1 Table of calculated u-velocity sensitivities for the complex-step problem. 28

3.2 Table of calculated v-velocity sensitivities for the complex step problem. 28

3.3 Table of pressure drag coefficient values for the wide, circular, and narrow cylinder design cases. 34

3.4 Table of pressure drag coefficient sensitivities; values calculated using analytic continuum sensitivity analysis are compared to those obtained using numeric finite differences. 38

3.5 Table of normal spatial gradients at the circular cylinder leading edge point. 41

3.6 Table of normal spatial gradients at the circular cylinder top point. 41

Chapter 1

Introduction

Numerical computation of solutions to flow equations, such as the Navier-Stokes equations, involve discretizing the domain and calculating the flow solution at specific locations. Discretization methods for determining flow solutions include finite difference methods (FDM), finite element methods (FEM), and finite volume methods (FVM). Though all three methods have their distinct advantages and disadvantages, here FEM is used for the ease of calculating both solution and spatial derivatives of the solution throughout the domain.

In the case of FEM, the domain is broken up into discrete elements, where each element is governed by the same differential equations as the overall domain. Additionally, a solution form is assumed for each flow variable in each element, ensuring continuity between elements for the flow variables and some of their derivatives.[23] The degree of the assumed solution typically determines the order of convergence of the solution. Discrete values are stored at the nodes of each element, and a combination of the nodal values and the element shape functions is used to determine flow variable values at all non-nodal locations. The finite element method itself has commonly been applied to structural problems since at least the late 1950's. In 1956, Turner, Clough, Martin, and Topp wrote a paper in which they discussed several methods for performing

structural analysis.[29] Methods mentioned in the paper included elementary theories of flexure and torsion, wide beam theory, the method of redundant forces, and an early form of FEM known as "direct stiffness calculation." The primary focus of the paper was to discuss and apply the direct stiffness calculation method to specific cases and show the method's functionality using several test cases.

Though early forms of FEM were developed for structural analysis and have continued to be applied to structures,[25] versions of the method have been used for other applications. Some recent applications for FEM include fluid mechanics, thermal analysis, electromagnetism, and plasma dynamics. IFISS is an example of the application of mixed finite elements to incompressible flow problems.[9, 26, 27] Developed as a package for MATLAB, IFISS allows users to learn and apply all parts of the mixed finite element process, from mesh generation to simulation performance and data post-processing.

Sensitivity analysis is a tool for both analysis and design which identifies the effect of a particular design variable on a given performance metric.[2] As an example, sensitivities of lift and drag to local wing cross-section dimensions can be identified to aid in the optimization of wing design. There are widespread applications for sensitivity analysis in the computational physics community, including turbulence modeling, system dynamics and control, and structural analysis. With regard to flow of a fluid, sensitivity analysis has been used to identify controlling parameters of fluid flow simulations.[5, 11, 12] It is known that the determination of shape sensitivities, the focus of this thesis, has previously been an application of sensitivity analysis.[28] However, this thesis extends sensitivity analysis for shape optimization to the preexisting incompressible flow mixed FEM software IFISS to gauge the performance of analytical sensitivity calculations.

Chapter 2

Theory

As previously mentioned, the goals of this thesis are to gain understanding about mixed finite element methods using the open-source IFISS software, and to perform flow analysis and sensitivity analysis for flow over a backward-facing step and flow over a cylinder; results were verified using alternate methods and validated with experimental data. This section explains some of the theory behind the operation of IFISS as it was used, as well as details of the sensitivity analysis methods employed here. The first subsection discusses IFISS: the systems of equations it was written to solve and the applied boundary conditions, the use of mixed finite elements and program- and user-developed meshes for computation, and the three possible types of iterations used in determining a flow solution. The second subsection examines each of the three methods of sensitivity analysis analyzed and used for the present research, providing advantages and disadvantages for each.

Though much of the mixed FEM discussion in this section is in connection with IFISS, it is noted that many of the statements made apply to mixed FEM programs using Taylor-Hood elements.[9] Rather than using the program simply as a standalone, this paper analyzes IFISS as a lens through which to view and understand

the functions of mixed FEM programs.

2.1 IFISS

In this thesis, version 3.4 of the 2015 Incompressible Flow & Interative Solver Software (IFISS) developed by Silvester, Elman, and Ramage was used for incompressible flow problems. IFISS is an open-source mixed finite element toolbox for MATLAB that uses preconditioned iterative solvers for either steady or unsteady problems to solve one of several sets of equations, including the Stokes and Navier-Stokes equations.[9,10] Test cases for the program include elementary flows also used to test commercial codes; among these test cases are: a square cylinder, a backward-facing step, a channel, and a lid driven cavity. With some alterations of the code and mesh information from an external source, IFISS can solve for flow properties in a wide variety of geometries.

IFISS was developed as a toolbox in MATLAB to allow users to perform mixed FEM fluid simulations. IFISS was specifically chosen for the present work because it is an open-source software that provides the user with an opportunity to learn about the mixed FEM process, from mesh generation to the performance of a simulation. Each part of IFISS functionality is discussed in this section. First, the systems of equations IFISS is programmed to solve are discussed, followed by the boundary conditions applied for individual problems. Next, use of mixed finite element meshes and generation of meshes used for simulations are detailed. Finally, detailed descriptions are provided for processes used by each set of iterations to determine flow solutions.

2.1.1 Systems of Equations

IFISS has been programmed to solve systems of equations in two-dimensional problems for specific steady and unsteady systems of equations. IFISS is able to solve steady systems including the Poisson equation, the convection-diffusion equation, the Stokes equations, and the Navier-Stokes equations. Unsteady systems available to solve with IFISS include the heat equation, the convection-diffusion equation, the Navier-Stokes equations, and the Boussinesq flow equations. The momentum equations for the aforementioned steady systems are shown in Equations 2.1-2.4; the continuity equation $\nabla \cdot \vec{u} = 0$ is also needed for both the Stokes and Navier-Stokes systems.

$$-\nabla^2 u = f \tag{2.1}$$

$$-\epsilon \nabla^2 \vec{u} + \vec{w} \cdot \nabla u = 0 \tag{2.2}$$

$$-\nabla^2 \vec{u} + \nabla p = \vec{0} \tag{2.3}$$

$$-\nu \nabla^2 \vec{u} + \vec{u} \cdot \nabla \vec{u} + \nabla p = \vec{f} \tag{2.4}$$

The focus of the present analyses is the determination of a solution to the Navier-Stokes equations. To this end, it is important to note that the pressure in the above version of the Navier-Stokes equations has been normalized by the density; this is possible because the flow is incompressible and therefore the density is constant.

2.1.2 Boundary Conditions

For a given problem, boundary conditions are required to solve the system of equations at all interior points. Essential Dirichlet and natural Neumann boundary conditions are used in IFISS, depending on the desired behavior of the flow. Inlet and freestream velocity distributions, as well as no-slip velocity conditions at a wall or an object surface, are specified directly as Dirichlet boundary conditions. An example inlet Dirichlet boundary condition for flow past a backward-facing step is given by Equation 2.5; in the equation, a controls the maximum inlet velocity magnitude.

$$\vec{u}(a, y) = (ay(1 - y))\hat{i} + 0\hat{j}, 0 < y < 1 \tag{2.5}$$

It should be emphasized that the Dirichlet boundary conditions are essential, and thus strongly enforced, to ensure an exact inflow velocity distribution and no flow through physical walls.

For outlet or wake boundaries, a natural Neumann velocity boundary condition is commonly used, given by Equation 2.6.

$$\frac{\partial \overrightarrow{u}}{\partial n} - p\overrightarrow{n} = 0 \tag{2.6}$$

To weakly enforce that the velocity gradient in the Neumann boundary condition is equal to the pressure, the pressure is added as a forcing term to the equations.

2.1.3 Mixed Finite Elements and Mesh Generation

IFISS presents several options for mixed finite elements, which are used for the velocity and pressure flow variables as well as the geometry. There are four possible combinations of finite elements for the flow variables: $Q_1 - P_0$, $Q_1 - Q_1$, $Q_2 - P_1$, and $Q_2 - Q_1$. These finite elements determine the extent to which variable values are allowed to change over each element in both the x- and y-directions; P_0 indicates that a quantity is constant over the element, P_1 indicates a linear variation over an element that is not continuous between elements at boundaries, Q_1 allows linear variation with inter-element continuity, and Q_2 means the variation can be quadratic. Graphical representations of each of the four elements are shown in Figure 2.1.

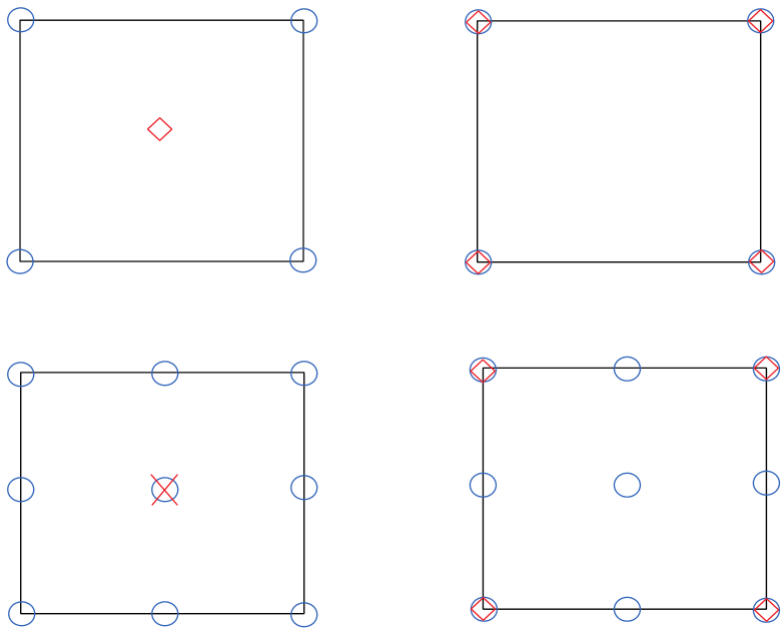

Figure 2.1: Representations of the $Q_1 - P_0$ (top left), $Q_1 - Q_1$ (top right), $Q_2 - P_1$ (bottom left), and $Q_2 - Q_1$ (bottom right) mixed finite elements; for flow variables, the blue circles indicate nodes for u- and v-components of velocity while red diamonds indicate nodes for pressure and the red 'x' represents nodes at which both pressure and its spatial derivatives are stored.

The mixed finite elements used for flow properties in this paper are Taylor-Hood $Q_2 - Q_1$ elements, which

allow for quadratic variation for the x- and y-components of velocity, and linear variation for pressure. This means the equations used for the flow quantities take the forms:

$$u = a_1 + b_1x + c_1y + d_1xy + e_1x^2 + f_1y^2 + g_1x^2y + h_1xy^2 + k_1x^2y^2$$

$$v = a_2 + b_2x + c_2y + d_2xy + e_2x^2 + f_2y^2 + g_2x^2y + h_2xy^2 + k_2x^2y^2$$

$$p = a_3 + b_3x + c_3y + d_3xy$$

As can be seen in the above equations and the earlier mixed finite element representations, second-order finite elements utilize nine nodes to accommodate the nine terms of a biquadratic equation; by contrast, a first-order element uses four nodes in conjunction with the four terms of a bilinear equation.

For geometry, all finite elements are bilinearly mapped from the master coordinate system into the physical space. Because all elements used in the flow have straight edges, curved walls and object surfaces in the flow are approximated using straight lines. This also means increasing mesh refinement results in a closer approximation of curved surfaces.

It should be noted that, in connection to the generation of the mesh, initial forms of the flow Jacobian and the right-hand-side vector for the system of equations must be calculated using Gaussian quadrature with the geometric data. This is best performed using an internal IFISS function with the generated mesh data as input.

An IFISS user can generate either structured or unstructured quadrilateral meshes with elements of predetermined orders, import the mesh data generated by the mesh generation software into IFISS, and solve large systems of equations using sparse matrices. The 2017 open-source extension T-IFISS developed by Silvester, Bespalov, and Liao allows for similar problem solving using triangular meshes.

Six meshes were developed for the backward-facing step problem analysis - three uniform meshes developed within IFISS, and three non-uniform meshes developed using Gmsh. The domain used for all six meshes is shown in Figure 2.2. A Dirichlet boundary condition is used at the inlet, described by the previously mentioned Equation 2.5. The walls forming the top and bottom boundaries of the domain, as well as the cylinder surface, have zero-velocity Dirichlet boundary conditions. A Neumann boundary condition is prescribed at the outlet, described by Equation 2.6.

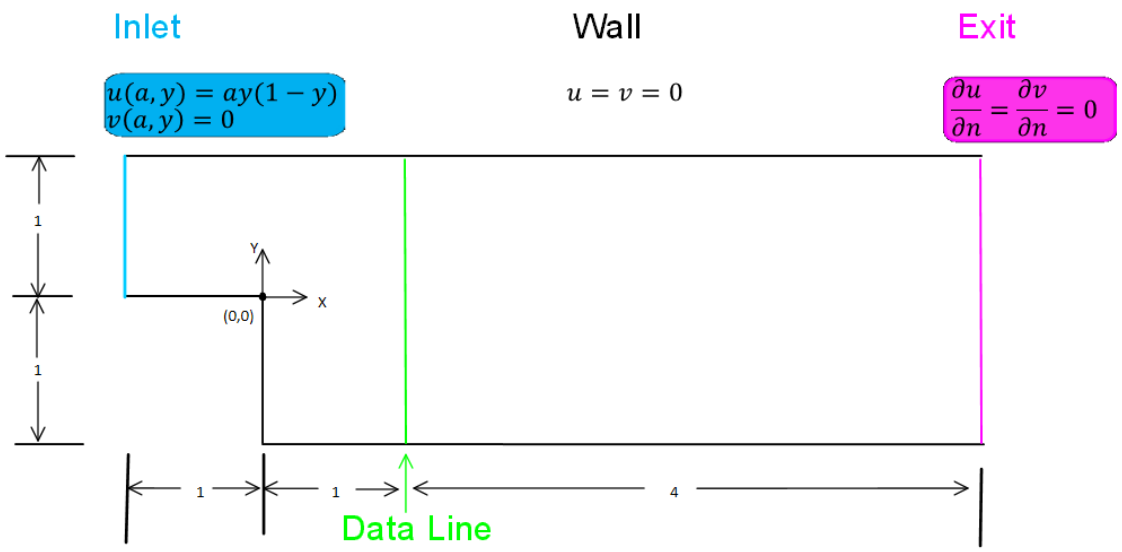

Figure 2.2: Domain for the backward-facing step cases, with the origin at the edge of the step; flow enters through the inlet (cyan), passes the data line (green), and exits the domain through the outlet (purple).

A number of meshes were generated of the cylinder problem for shape optimization: three sets of coarse, medium, fine, and ultra-fine meshes. The first set of meshes incorporated the same circular cylinder with a sectioned region immediately surrounding. The second and third set of four meshes included an elliptical cylinder obtained by altering the x-direction diameter a by 10%; the second set increased a by 10% to obtain a "wide" elliptical cylinder, whereas the third set decreased a by 10% to obtain a "narrow" cylinder. An "ultra-fine+" mesh was later added for each set to better observe result trends. All circular cylinder meshes were developed using the domain shown in Figure 2.3 can be found in Appendix B. Representations of the

three cylinder design types are shown in Figure 2.4

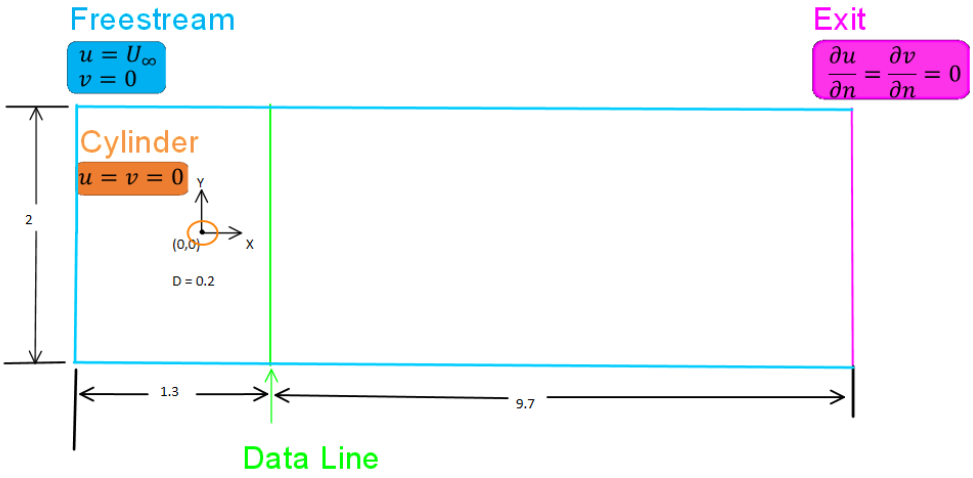

Figure 2.3: Domain for the circular cylinder cases, with the origin at the center of the cylinder; the fluid enters through the inlet (cyan), flows over the cylinder (orange) and past the data line (green), and exits the domain through the outlet (purple).

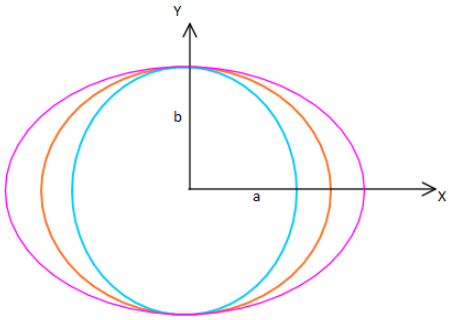

Figure 2.4: Representation of changing cylinder designs; orange indicates the circular cylinder design, purple indicates the "wide" cylinder design, and cyan indicates the "narrow" cylinder design.

The process used by IFISS to solve the aforementioned fluid flow problems consists of three distinct segments: a single Stokes iteration, a block of Picard iterations, and a block of Newton iterations. While the Stokes iteration solves a simple linearized system of equations, the Picard and Newton iteration blocks attempt improve the accuracy of the results by approximating the nonlinear terms. Each of these three segments

uses a distinct solution process, as discussed below.

2.1.4 Stokes Iteration

The program first solves a linearized version of the Stokes equations. Equation 2.7 shows a general form of the system, in which $[A]$ is the vector-Laplacian matrix and $[B]$ is the divergence matrix.

$$\begin{bmatrix} [A] & [B]^T \\ [B] & [0] \end{bmatrix} \begin{Bmatrix} \{\overrightarrow{U}\} \\ \{p\} \end{Bmatrix} = \begin{Bmatrix} \{f\} \\ \{g\} \end{Bmatrix} \tag{2.7}$$

Separating the velocity vector into its x- and y-components, and separating the A- and B-matrices similarly, the system can be expressed as Equation 2.8. Here $[A_m]$ represents a nonzero block of the larger matrix $[A]$, and the two $[A_m]$ blocks are equal. Additionally, $[B_x]$ and $[B_y]$ are coefficient matrices for derivatives in the x- and y-directions. As a result of separating $\{\overrightarrow{U}\}$ into x- and y-components of velocity, three distinct pieces can be observed: the x- and y- momentum equations, and the continuity equation.

$$\begin{bmatrix} [A_m] & [0] & [B_x]^T \\ [0] & [A_m] & [B_y]^T \\ [B_x] & [B_y] & [0] \end{bmatrix} \begin{Bmatrix} \{u\} \\ \{v\} \\ \{p\} \end{Bmatrix} = \begin{Bmatrix} \{f_x\} \\ \{f_y\} \\ \{g\} \end{Bmatrix} \tag{2.8}$$

To solve the Stokes equations, forms for the Jacobian matrix and right-hand-side vector which were previously formed using Gaussian quadrature are updated by imposing the boundary conditions. The state vector $\{\vec{x}\} = \begin{Bmatrix} \{u\} \\ \{v\} \\ \{p\} \end{Bmatrix}$ is determined by solving the linear system, shown in Equation 2.8.

As a result of the mixed FEM formulation, while both components of velocity have one value for every node in the domain, pressure has less than half as many values. Though the order of variables is consistent, the order of the nodes themselves for each variable is dependent on how the meshes were developed.

2.1.5 Picard Iterations

The determined Stokes solution and Navier-Stokes residual then become input for the first Picard iteration. Picard iterations linearize the equations in a way that the results of the previous iteration are included in the calculation of the solution for the current iteration. As an example, consider the equation $x^2 - 4x + 4 = 0$, where the initial guess x_0 equals 5 and the current guess is 1. For this example, the Picard iteration would solve for a new quantity x_i in terms of the previous quantity x_{i-1} using an altered form of the equation: $x_{i-1}\Delta x_i - 4\Delta x_i + 4 = R$, where R is the residual. The new value of x_i would be updated using the equation $\Delta x_i = x_i - x_{i-1} = -\frac{R}{J}$. The guesses would continue, with x_i serving as the current guess and x_{i-1} serving as the previous guess, until either the iteration counter reached the maximum number of iterations or the residual fell below some preset tolerance. In the example problem, the program would start to converge to the correct answer of 2 in 5 iterations.

The Navier-Stokes residual is determined using a modified form of the system. The most general form of the system is shown in Equation 2.9. Here, $[N(\vec{x}_{old})]$ is a convection matrix determined using Gauss integration

with the Stokes solution values and ν is the kinematic viscosity.

$$\begin{bmatrix} \nu[A] + [N(\vec{x}_{old})] & [B]^T \\ [B] & [0] \end{bmatrix} \{\vec{x}_{new}\} = \begin{Bmatrix} \{f\} \\ \{g\} \end{Bmatrix} \tag{2.9}$$

Separating velocity into its x- and y-components as before, an expanded version is found as shown in Equation 2.10. In the equation, $[N_m]$ is a non-zero block of $[N]$ determined using Gauss integration with the updated state variable values; as with $[A]$, the two non-zero diagonal blocks of $[N]$ are equal.

$$\overrightarrow{\Delta x} = \begin{bmatrix} \nu[A_m] + [N_m(\vec{x}_{old})] & [0] & [B_x]^T \\ [0] & \nu[A_m] + [N_m(\vec{x}_{old})] & [B_y]^T \\ [B_x] & [B_y] & [0] \end{bmatrix}^{-1} \vec{R} \tag{2.10}$$

With the boundary conditions applied to the system, the residual for the Navier-Stokes equations using the Stokes iteration solution is then found using Equation 2.11. The updated flow solution and the residual are then carried to the next iteration.

$$\vec{R} = \begin{bmatrix} \nu[A_m] + [N_m(\vec{x}_{old})] & [0] & [B_x]^T \\ [0] & \nu[A_m] + [N_m(\vec{x}_{old})] & [B_y]^T \\ [B_x] & [B_y] & [0] \end{bmatrix} \begin{Bmatrix} \{u\} \\ \{v\} \\ \{p\} \end{Bmatrix} - \begin{Bmatrix} \{f_x\} \\ \{f_y\} \\ \{g\} \end{Bmatrix} \tag{2.11}$$

Within the context of this thesis, the Picard iterations first update the solution using $\vec{x}_{new} = \vec{x}_{old} + \overrightarrow{\Delta x}$, where $\overrightarrow{\Delta x}$ is given by Equation 2.10. As detailed above, $[A_m]$ and $[N_m(\vec{x}_{old})]$ are the non-zero block diagonal matrices from $[A]$ and $[N(\vec{x}_{old})]$, respectively, and $[B_x]$ and $[B_y]$ contain x- and y-derivative coefficients. It should also be noted that $[N_m(\vec{x}_{old})]$ is updated at each iteration, using Gauss integration with the flow solution from the previous iteration. Once the state variable values have been updated using $\overrightarrow{\Delta x}$, the residual is found using Equation 2.11.

For the first Picard iteration in IFISS, the nonlinear coefficient matrix and residual vector based on the linear Stokes solution are used to calculate the update to the present flow solution. A new matrix N is calculated and is used to update the velocity sub-matrix in the nonlinear coefficient matrix. Boundary conditions are then re-imposed on both the matrix and the right-hand-side vector, a new residual is calculated, and the information from the current Picard iteration is used for the next Picard iteration until either the maximum number of allowable iterations is achieved or the residual for the equation falls below the tolerance.

2.1.6 Newton Iterations

Similarly to the use of the Stokes solution as input for the first Picard iteration, the solution from the last Picard iteration is then used as starting information for the first Newton iteration. Within the Newton iterative cycle, the change in state variable from one iteration to the next is determined and applied, similar to Picard iterations. However, the flow Jacobian is used to update the solution; the $[N_{xx}]$ and $[N_{yy}]$ matrices are introduced, as well as the off-diagonal $[N_{xy}]$ and $[N_{yx}]$ sub-matrices that incorporate linearized terms from the nonlinearities in the calculation. The process used to update the state vector at each Newton iteration is shown in Equation 2.12.

$$\overrightarrow{\Delta x} = \begin{bmatrix} \nu[A_m] + [N_m(\vec{x}_{old})] + [N_{xx}(\vec{x}_{old})] & [N_{xy}(\vec{x}_{old})] & [B_x]^T \\ [N_{yx}(\vec{x}_{old})] & \nu[A_m] + [N_m(\vec{x}_{old})] + [N_{yy}(\vec{x}_{old})] & [B_y]^T \\ [B_x] & [B_y] & [0] \end{bmatrix}^{-1} \{\vec{R}\} \qquad (2.12)$$

Once the state vector is updated using $\overrightarrow{\Delta x}$, the residual for the solution is determined as in Equation 2.11 from the Picard iterative cycle.

2.2 Sensitivity Analysis Methods

Sensitivity analysis is the process of determining the magnitude of effect of one or multiple design parameters on one or more performance metrics. An example of this would be finding sensitivities of a function f to three independent design variables x_1, x_2, and x_3 where f is given by $f(x_1, x_2, x_3) = x_1^2 + \sin(\frac{x_2}{2x_3})$. The sensitivities of f with respect to each of the design variables, in terms of the variables themselves, would be:

$$\frac{df}{dx_1} = 2x_1$$
$$\frac{df}{dx_2} = \frac{1}{2x_3}\cos(\frac{x_2}{2x_3})$$
$$\frac{df}{dx_3} = -\frac{x_2}{2x_3^2}\cos(\frac{x_2}{2x_3})$$

There are three large categories of sensitivity analysis: numeric, analytic, and automatic differentiation.[16] A sensitivity analysis taxonomy is provided in Figure 2.5, outlining the different categories of sensitivity analysis.

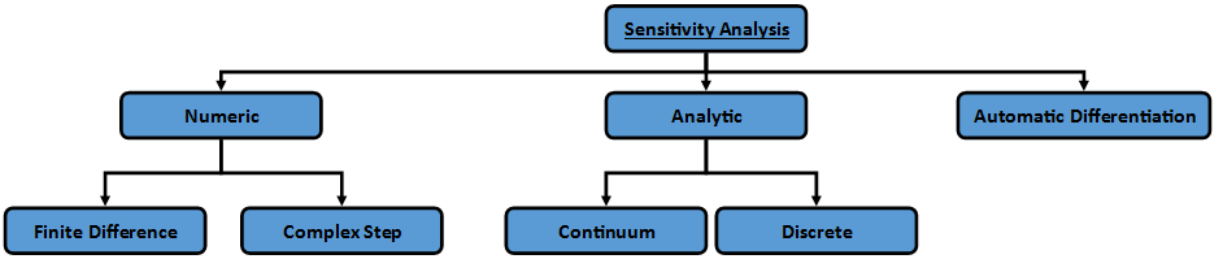

Figure 2.5: Graphic of sensitivity analysis categories.

The goal of this research is to lay the foundation of shape optimization, determining shape sensitivities by implementing continuum sensitivity analysis with the aforementioned IFISS code. This section outlines the sensitivity analysis methods employed in the process of the research. The first section provides general details of the analytic methods of sensitivity analysis, with particular emphasis on the use of continuum sensitivity analysis (CSA) for shape optimization. The second and third sections discuss the numeric finite difference and complex step methods, respectively.

2.2.1 Analytic Sensitivity Calculation

Analytic sensitivity analysis methods determine performance metric sensitivities to design variables using a combination of differentiation and discretization. The order of the two steps with regard to the performance metric equations determines in which category the sensitivity analysis method falls. Differentiation of the partial differential equations with respect to the design variable before discretizing the equations using FEM classifies the method as a continuum sensitivity analysis method, whereas the reverse order results in discrete sensitivity analysis.

CSA specifically is used for this research rather than discrete sensitivity analysis, as it allows for the calculation of sensitivities without the calculation of the change in mesh with changing cylinder geometry;

the determination of mesh sensitivity as used in discrete methods should be avoided because, as noted by Duvigneau and Pelletier[8] and Kulkarni,[16] the process of finding such sensitivities is computationally expensive. Duvigneau and Pelletier also note that it is not necessary to use the same algorithm to determine continuum sensitivities as was used to solve the governing differential equations. A variety of examples exist of the successful implementation of CSA, including applications to fluids,[11] structures,[15] and aeroelasticity.[18]

A local implementation of CSA was used for this particular research, and specific details regarding the implementation can be found in the Results section. Generally speaking, it is recognized that a change in body boundary results in both direct and indirect effects on the flow. To this end, the boundary conditions for the sensitivity analysis are in terms of $\frac{\partial u}{\partial a}$ and $\frac{\partial v}{\partial a}$ rather than u and v; this requires the calculation of spatial gradients of the primal (flow analysis) solution, as well as derivatives of the geometry with respect to the design variable. A new solution containing $\{\frac{\partial u}{\partial a}\}$, $\{\frac{\partial v}{\partial a}\}$, and $\{\frac{\partial p}{\partial a}\}$ is then determined for the flow using the full flow Jacobian from the final Newton iteration and the altered right-hand-side vector. The action following the determination of the local sensitivities is dependent on the performance metric, though aerodynamic applications likely require the integration over the body surface of normal stresses, shear stresses, or a combination of the two. Once all desired continuum shape sensitivities have been determined, the groundwork is laid for shape optimization.

2.2.2 Finite Difference Sensitivity Calculation

For the finite difference method of approximating sensitivities, a baseline case is first run which uses the initial design variable value α to calculate a value for the performance metric $Q(\alpha)$. Another case is then run in which α is perturbed by some small amount $\Delta\alpha$ and the new value for the performance metric $Q(\alpha+\Delta\alpha)$ is determined. The sensitivity of Q to α is found by:

$$\frac{dQ}{d\alpha} \approx \frac{Q(\alpha + \Delta\alpha) - Q(\alpha)}{\Delta\alpha} \tag{2.13}$$

The fact that the results from one additional nonlinear simulation is needed for this sensitivity calculation is both an advantage and a disadvantage. Because there is no program adaptation necessary, this is an effective black-box method that requires minimal effort by the user. However, this also means the calculation of the sensitivity requires twice as much time as a standard simulation; while simple fluid simulations like panel methods operate on the order of seconds and therefore are not affected seriously, more complex turbulence modeling simulations such as those used by Lee[17] which can take months or even years to complete suffer significantly. Additionally, finite difference methods for determining sensitivities suffer from the "step size problem;" too large of a step size results in an inaccurate secant approximation, while too small of a step size results in subtractive cancellation due to finite precision and highly erroneous results.

2.2.3 Complex Step Sensitivity Calculation

With the complex step method, the design variable α is perturbed by an imaginary amount $i\Delta\alpha$, resulting in a complex parameter in the analysis. The complex sensitivity is then approximated by taking the imaginary part of the resulting value of performance metric Q and dividing by the perturbation, as demonstrated in Equation 2.14.

$$\frac{dQ}{d\alpha} \approx \frac{\Im(Q(\alpha + i\Delta\alpha))}{\Delta\alpha} \tag{2.14}$$

As with the other two methods, there are distinct advantages and disadvantages to the use of the complex step sensitivity method. With the use of complex numbers rather than all-real perturbations, the complex step method does not suffer from the step size problem as does the finite difference method; as such, imaginary perturbations as small as 1×10^{-305} have been used with few if any repercussions with respect to accuracy.[19] However, the program must be able to accommodate complex numbers to calculate complex step sensitivities; this can prove challenging depending on how and in what programming language the program is written. In the case of IFISS, the calculation of complex step sensitivities was trivial because of the capabilities of MATLAB.

Chapter 3

Results

3.1 Analytic Sensitivity to Value

As a precursor to the primary analysis, backward-facing step simulation results using an IFISS-generated uniform rectangular mesh were compared to those using non-uniform quadrilateral meshes generated in Gmsh. The coarse uniform and non-uniform backward-facing step meshes, respectively, are shown in Figures 3.1 and 3.2. Medium and fine meshes for both uniform and non-uniform backward-facing step cases, produced using a grid refinement factor of 2, are found in Appendix A.

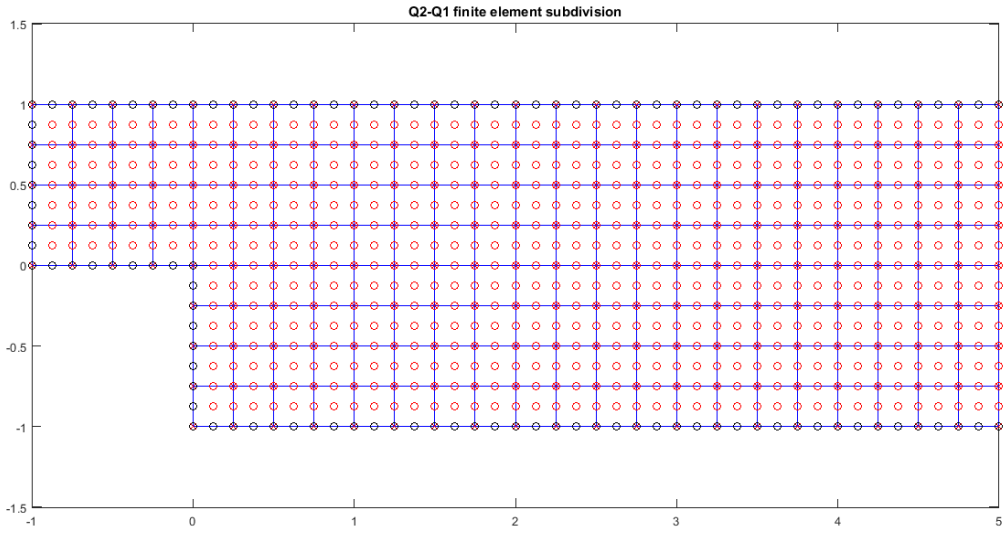

Figure 3.1: Uniform coarse backward-facing step mesh generated in IFISS for sensitivity analysis.

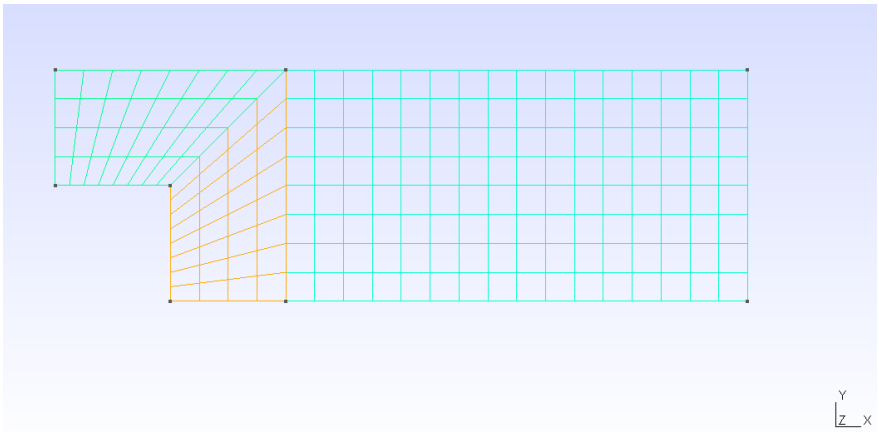

Figure 3.2: Non-uniform coarse backward-facing step mesh generated in Gmsh for sensitivity analysis.

As a grid convergence study, a flow with inlet Reynolds number value 50, parabolic velocity distribution at the inlet, zero Dirichlet boundary condition for both u- and v-components of velocity at the wall surfaces, and a Neumann boundary condition at the exit was simulated for the backward-facing step problem. Figure 3.3 shows the vector field within the domain produced by the fine non-uniform mesh, and Figure 3.4 shows a visualization of the wake immediately behind the step where a vortex can be seen. Data concerning u- and v-velocity was taken along the aforementioned data line for uniform and non-uniform meshes of varying

refinement. The u and v velocity results with the structured mesh are shown in Figures 3.5 and 3.6. As expected, the u-velocity values are completely positive at y-values above the surface of the step, and there is a vortex attached to the trailing edge of the step. Additionally, there is less than a 0.5% difference between velocity component values predicted using the coarse and fine meshes.

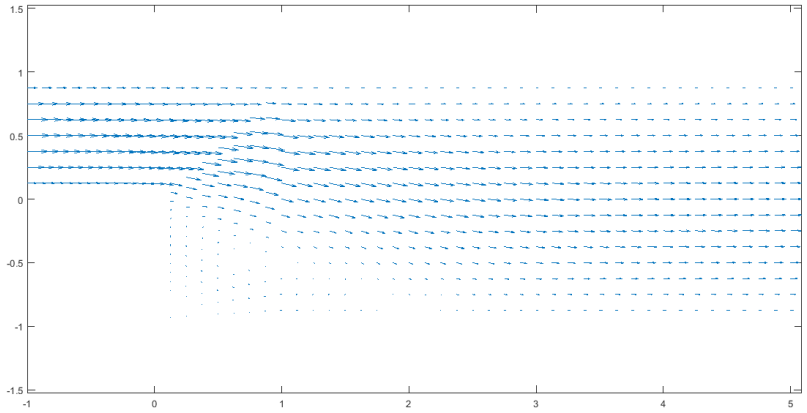

Figure 3.3: Vector field of backward-facing step flow.

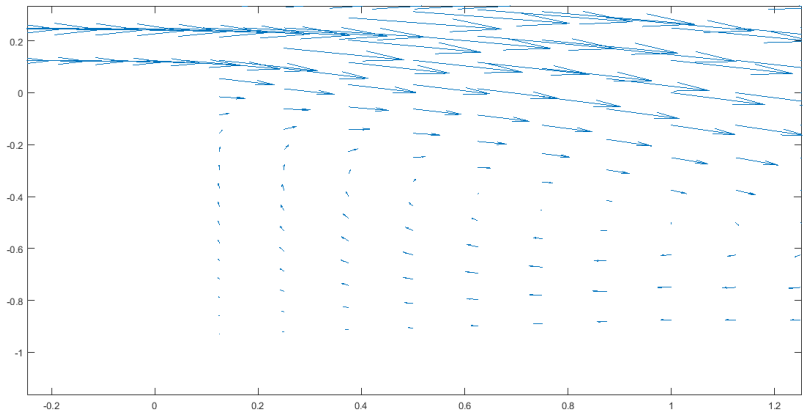

Figure 3.4: Vector field of backward-facing step flow wake.

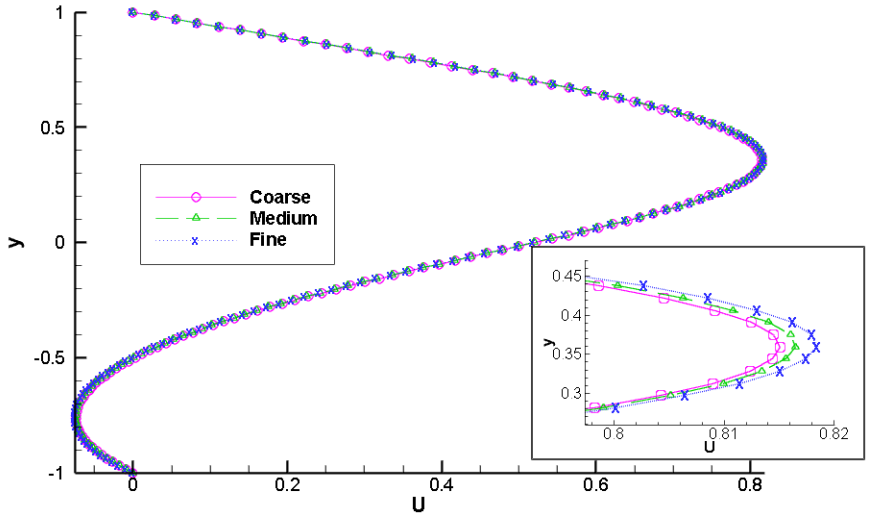

Figure 3.5: Plot of u-velocity components at x=1.0 for coarse (pink), medium (green), and fine (blue) uniform meshes.

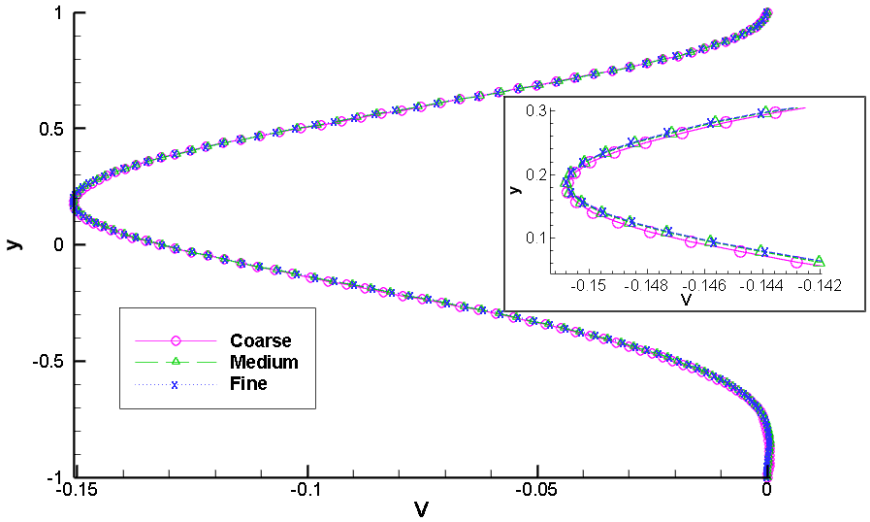

Figure 3.6: Plot of v-velocity components at x=1.0 for coarse (pink), medium (green), and fine (blue) uniform meshes.

Figure 3.7 shows a comparison of the uniform u-velocity results to those obtained using the non-uniform mesh in an expanded view of the region of the data line where the largest possible magnitude of u can be found. It can be seen in the figure that there is good agreement between the results, with less than 1%

difference in velocity prediction for any given level of mesh refinement.

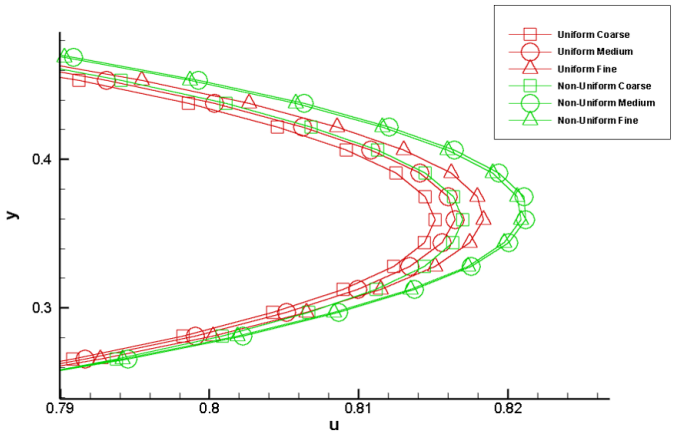

Figure 3.7: Velocity profiles for uniform and non-uniform meshes for flow over a backward-facing step.

To ensure that the uniform mesh results were converging with increasing mesh refinement, three additional meshes were generated and the velocity profiles were plotted and analyzed. The resulting velocity profile plot is shown in Figure 3.8.

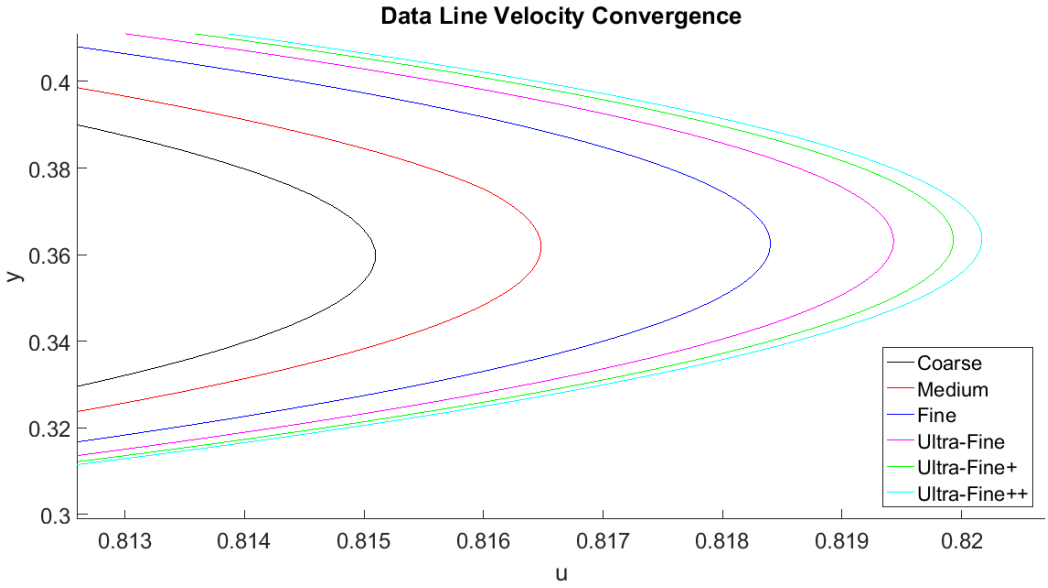

Figure 3.8: Velocity profiles for the uniform backward-facing step problem with increasing mesh refinement.

Though the results do not initially appear to be converging using only the coarsest three meshes, the distances between velocity values at a given y-value start decreasing with increasing mesh refinement after the fine mesh. A mesh convergence study was performed for the uniform meshes, and the results of the study are shown in Figure 3.9. The figure shows that the simulation is indeed converging with increasing mesh refinement after the medium mesh. The order of convergence is approximately 1.

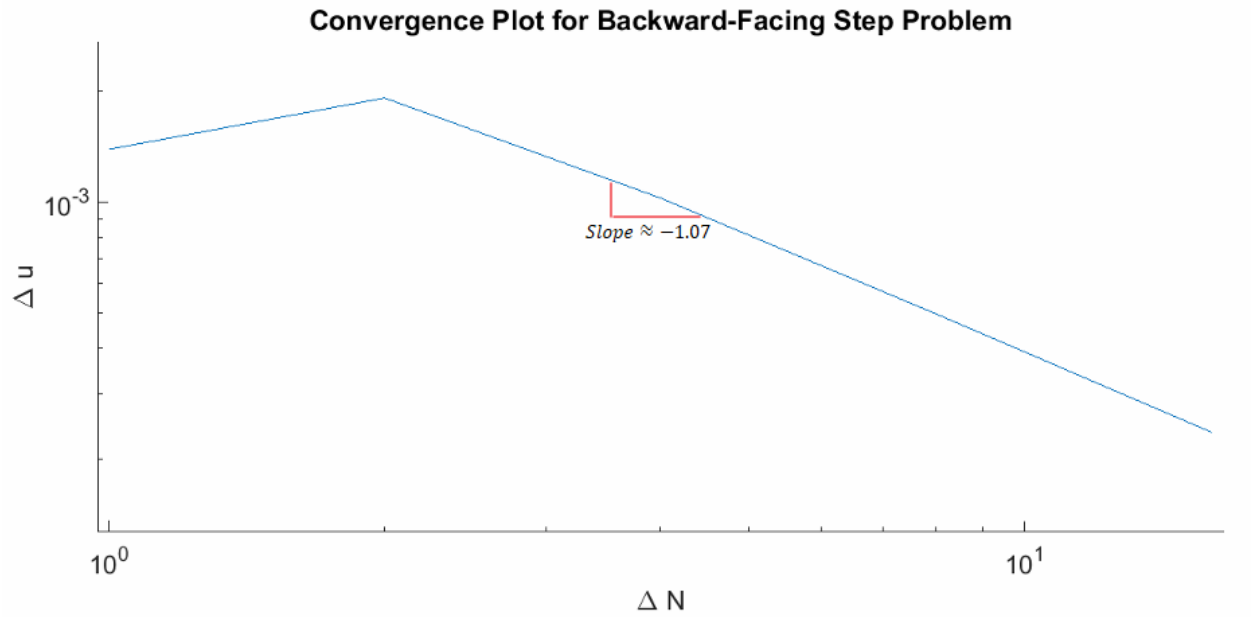

Figure 3.9: Mesh convergence study of the uniform backward-facing step problem with increasing mesh refinement.

In the backward-facing step cases, the sensitivities of both velocity components at points along the data line to the maximum inlet velocity magnitude were determined using three methods. In the first method, analytic sensitivities were calculated using a method similar to that outlined in the Analytic Sensitivity Calculation section; the boundary conditions were differentiated with respect to the parameter a, and a differentiated flow solution was obtained using the final Newton iteration Jacobian and the modified right-hand-side vector. For the second method, the flow solution was determined using a baseline maximum inlet velocity magnitude; another simulation was run using a perturbed maximum inlet velocity magnitude, and

a finite difference was used to calculate the overall sensitivity. The third method applied a small complex perturbation (1×10^{-20}) to the maximum inlet velocity magnitude, and the sensitivity was calculated using Equation 2.14. The results for the sensitivities of u- and v-velocity sensitivities are shown in Figures 3.10 and 3.11.

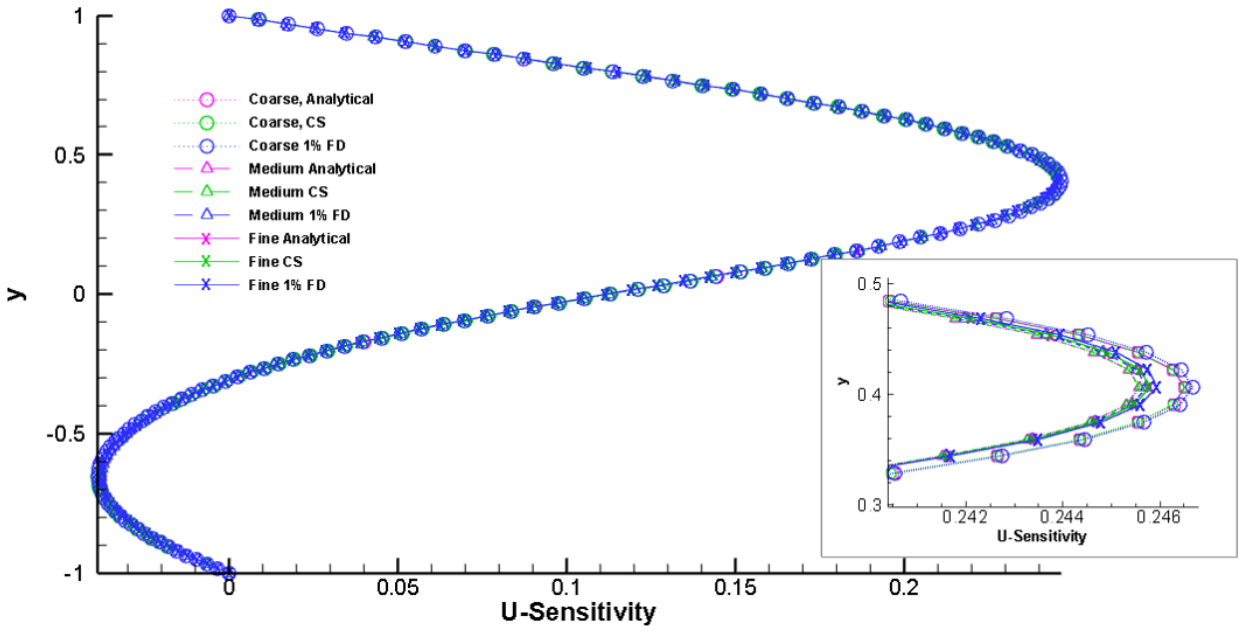

Figure 3.10: Plot of u-velocity sensitivities for coarse (circles), medium (triangles), and fine (x's) uniform structured meshes at x=1; the section of the greatest sensitivity magnitudes is expanded in the bottom right.

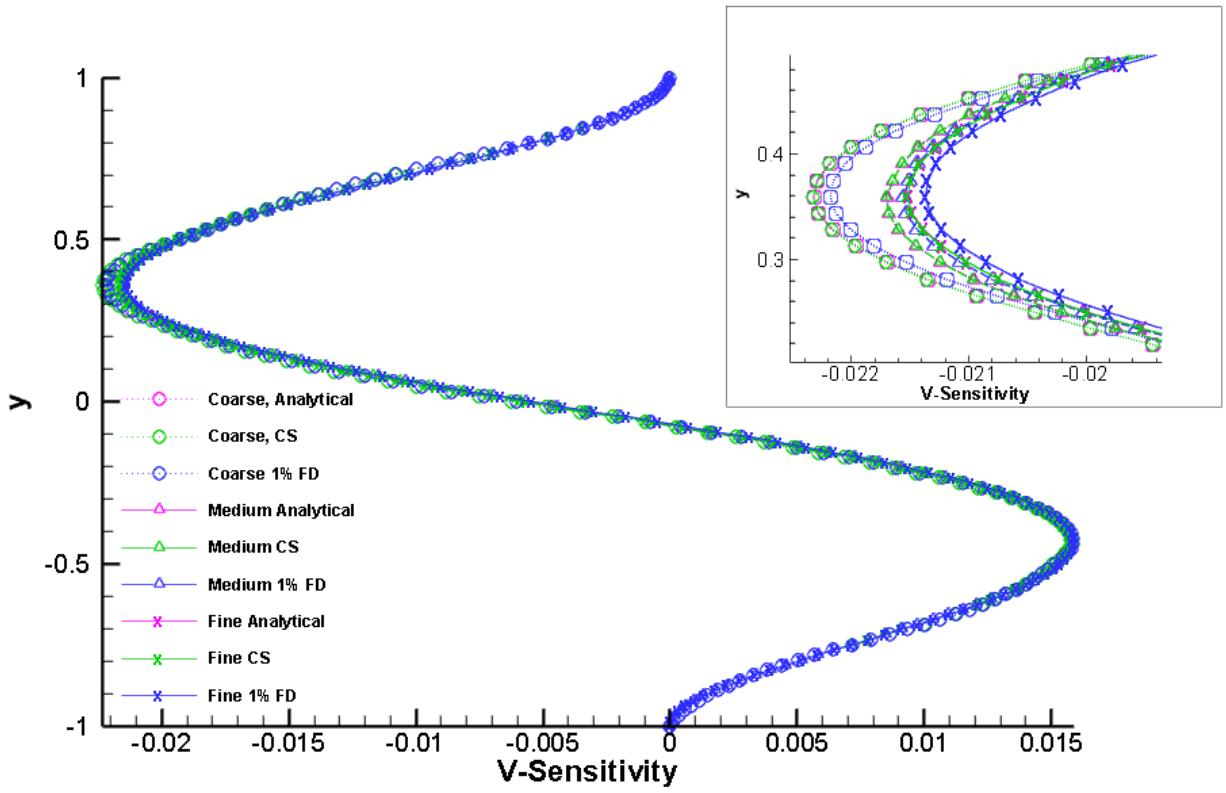

Figure 3.11: Plot of v-velocity sensitivities for coarse (circles), medium (triangles), and fine (x's) uniform structured meshes at x=1; the section of the greatest sensitivity magnitudes is expanded in the top right.

As seen in the figures, there is good agreement between all u-velocity sensitivities, with less than 1% difference between the results of the coarse mesh and those of the medium mesh. The v-velocity component sensitivities experience a larger gap, though the maximum difference between mesh results is less than 5%. An interesting but not critical detail to note in this case is that the u-velocity sensitivities experience oscillatory convergence with increasing mesh refinement; the coarse and medium mesh results bracket those of the fine mesh in the expanded area.

The determined sensitivity values at the points on the dataline with maximum u- and v-velocity sensitivities are shown in Tables 3.1 and 3.2; this corresponds to the points (1,0.4063) and (1,0.3594) for the u- and

v-velocity sensitivities, respectively.

Table 3.1: Table of calculated u-velocity sensitivities for the complex-step problem.

Method	Coarse Mesh	Medium Mesh	Fine Mesh
Analytic	0.246522903516081	0.245577303269798	0.245790373092267
Complex Step	0.246520966895026	0.245576634481022	0.245788999071464
Finite Difference, 1%	0.246676093005043	0.245722407830113	0.245929520302893
Finite Difference, 0.1%	0.246536645982986	0.245591385806238	0.245803205712269
Finite Difference, 0.01%	0.246522536478406	0.245578286670219	0.245790421289915

The analytic results for the u-velocity sensitivities match the complex step results up to 5 significant digits. By contrast, the quality of performance for the finite difference sensitivities is dependent on the step size; with step size reduced from 1% to 0.1%, the finite difference results approach those of the analytical and complex step methods.

Table 3.2: Table of calculated v-velocity sensitivities for the complex step problem.

Method	Coarse Mesh	Medium Mesh	Fine Mesh
Analytic	-0.0223202956158795	-0.0216963646965857	-0.0215173774492731
Complex Step	-0.0223274647645714	-0.0216995217267676	-0.0215186226680386
Finite Difference, 1%	-0.0221791472844259	-0.0215506394245932	-0.0213693943230787
Finite Difference, 0.1%	-0.0223126063550448	-0.0216846037675493	-0.0215036669923606
Finite Difference, 0.01%	-0.0223259786602528	-0.0216980419950863	-0.0215171267756251

The v-velocity sensitivity results are similar in performance to those seen with the u-velocity sensitivities, though the sensitivities of the v-velocity are an order of magnitude smaller. The sensitivity results converge to 5 digits after the decimal point but with only 4 significant digits.

3.2 Continuum Analytic Sensitivity to Shape Parameter

As mentioned earlier, three sets of meshes were developed to aid in shape sensitivity analysis of a cylinder in a two-dimensional flow at Reynolds number 60. For one set, the diameters in the x- and y-direction were equal resulting in a circular cylinder. A coarse mesh for the circular cylinder is shown in Figure 3.12, with a zoomed view shown in Figure 3.13. In the second mesh set, the diameter in the x-direction was increased by 10% resulting in a "wide" elliptical cross-section. The elliptical cylinder coarse mesh is shown in Figure 3.14. For the third set of meshes, the diameter of the cylinder in the x-direction was decreased by 10% from the circular cylinder diameter, resulting in a "narrow" elliptical cross section. A coarse mesh developed for the narrow elliptical cylinder case is shown in Figure 3.15. Finer meshes were developed for each cylinder case using a grid refinement factor of 2. All meshes used for the cylinder case are found in Appendix B. In all cylinder meshes, the line at which velocity data was taken was located at $x = 0.3$, three circular cylinder radii behind the center of the cylinder.

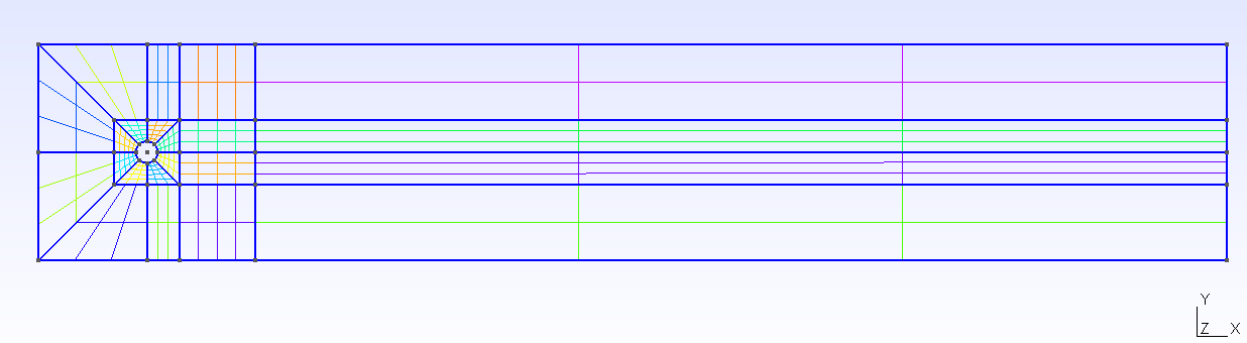

Figure 3.12: Coarse circular cylinder mesh generated in Gmsh for shape optimization cases.

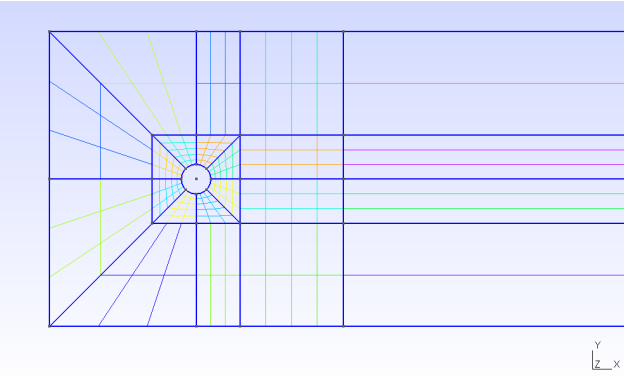

Figure 3.13: Expanded view of coarse circular cylinder mesh generated in Gmsh for shape optimization cases.

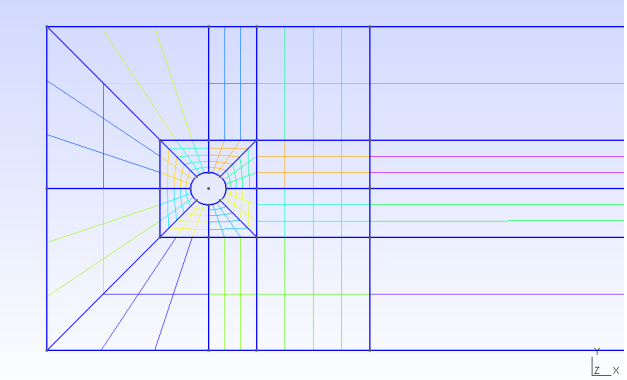

Figure 3.14: Coarse wide elliptical cylinder mesh generated in Gmsh for shape optimization cases.

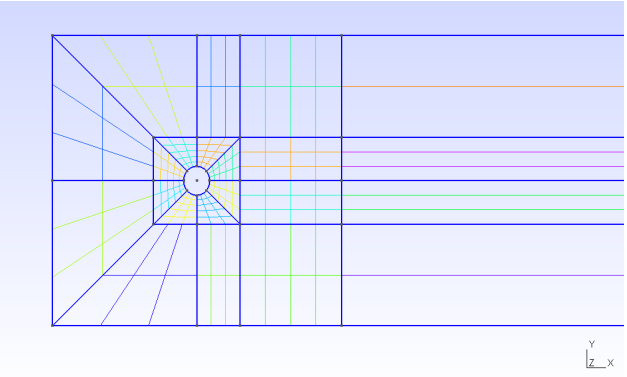

Figure 3.15: Coarse narrow elliptical cylinder mesh generated in Gmsh for shape optimization cases.

Solutions for the circular and elliptical cylinder uniform freestream flow cases were obtained using IFISS. As seen in Figures 3.16-3.19, the results of the simulation are as expected; the flow curves around the cylinder,

which has two attached counter-rotating vortical structures characteristic of low Reynolds number flow,[30] and gradually returns to a uniform velocity distribution.

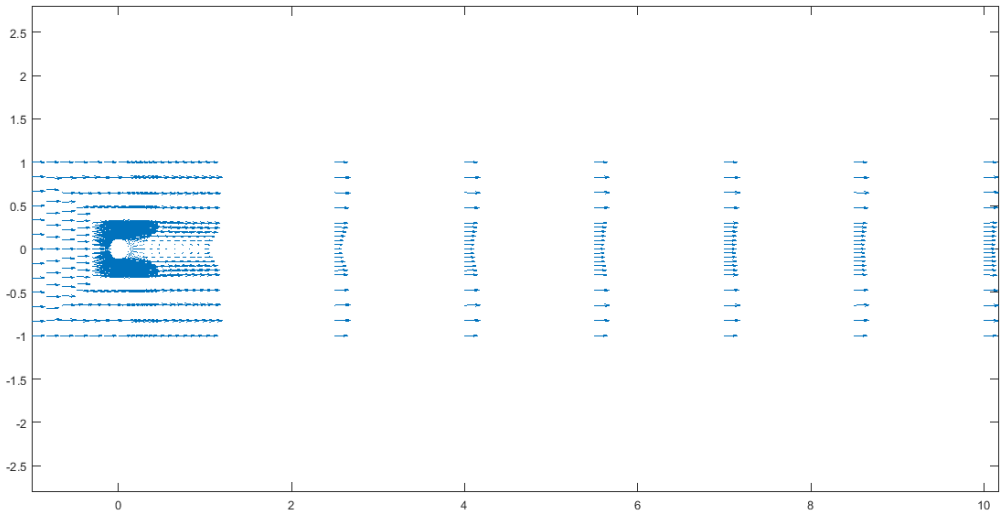

Figure 3.16: Circular cylinder coarse mesh vector field.

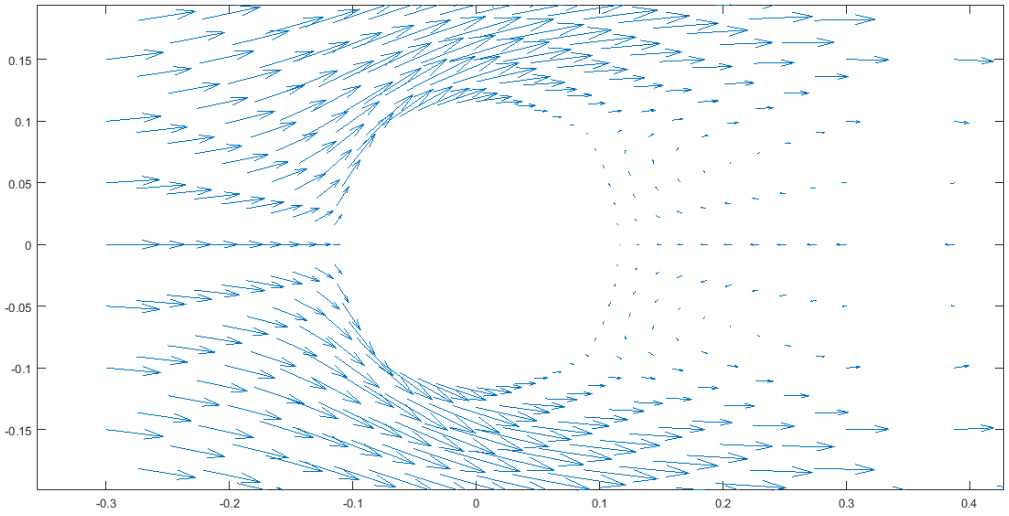

Figure 3.17: Expanded view of cylinder in coarse mesh vector field.

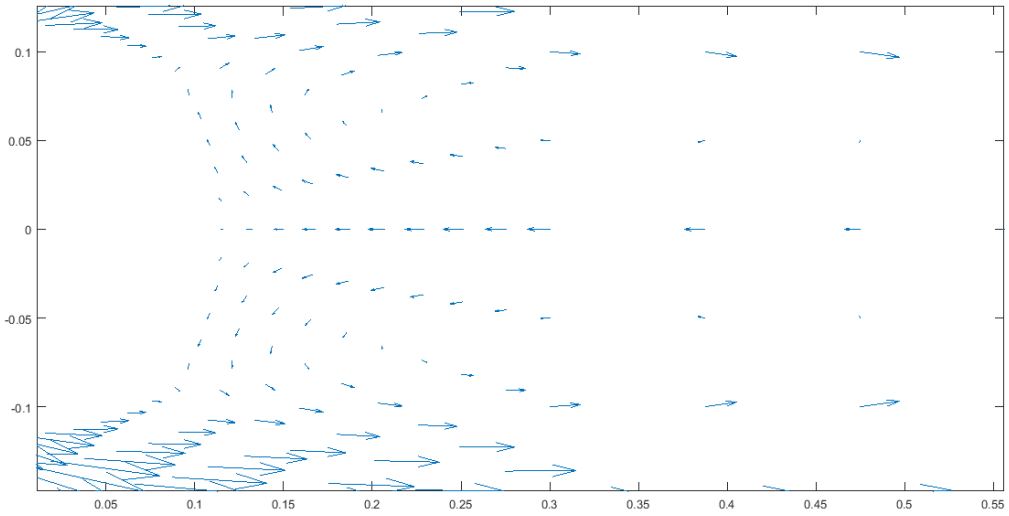

Figure 3.18: Vortical structures in the cylinder wake.

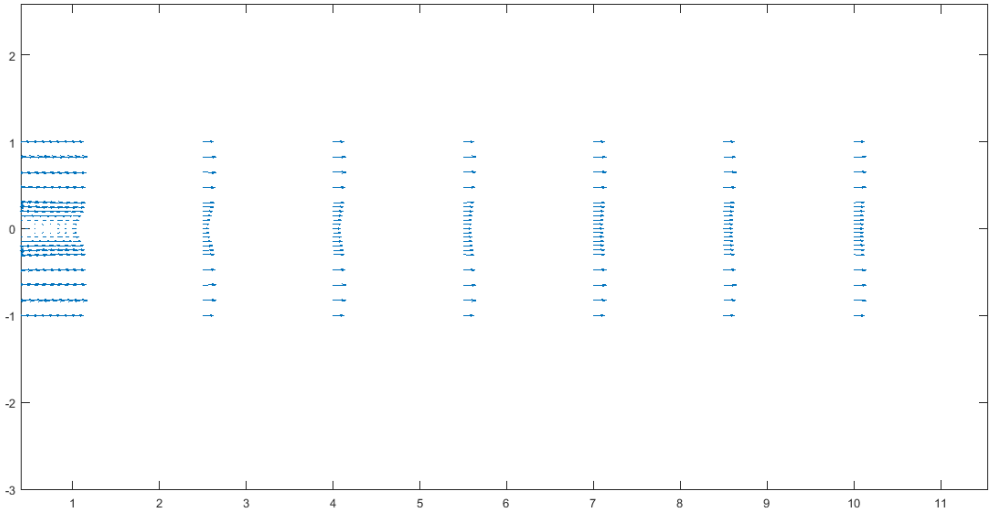

Figure 3.19: Flow wake far downstream of the cylinder.

Plots of u- and v-velocity along the dataline in the cylinder wake are shown in Figures 3.20 and 3.21, respectively. In these figures, the symmetry of the u-component of velocity and the antisymmetry of the v-component of velocity are clear.

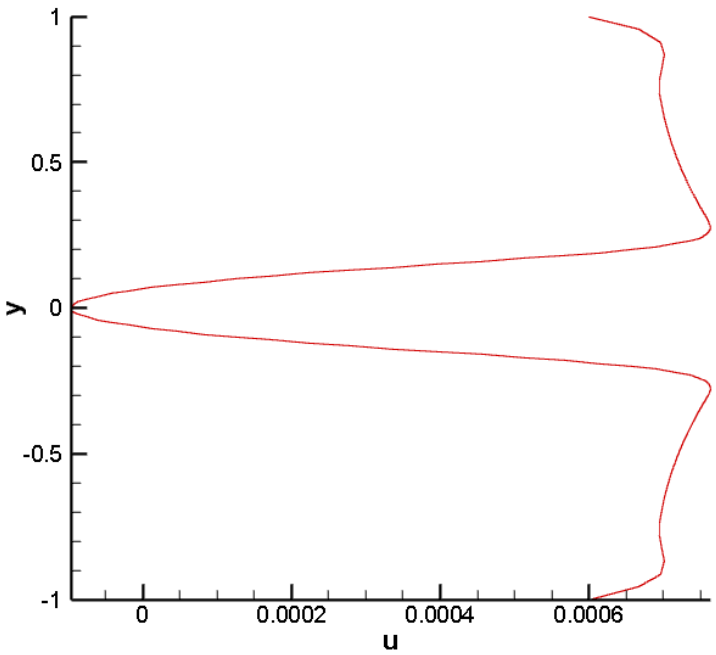

Figure 3.20: Distribution of u-velocity downstream of the cylinder at x = 0.3.

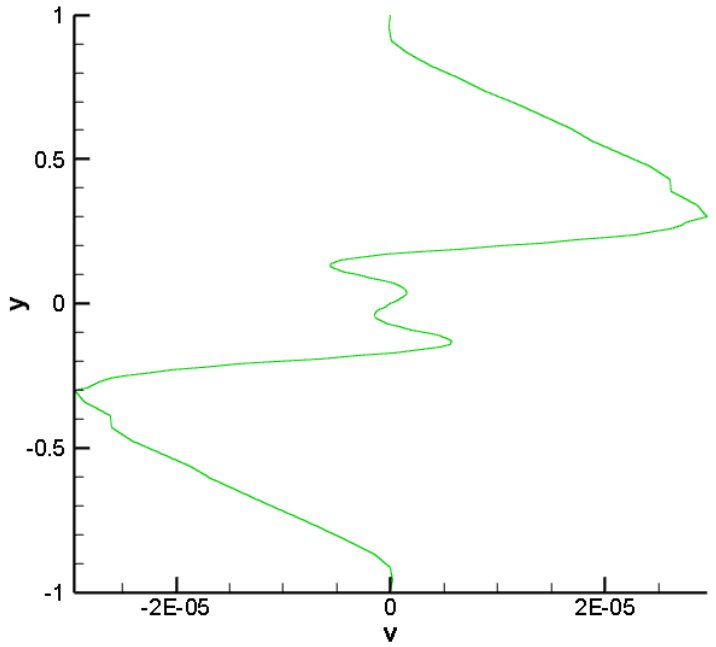

Figure 3.21: Distribution of v-velocity downstream of the cylinder at x = 0.3.

Values for pressure drag normalized by dynamic pressure and diameter were determined using Equations 3.1; in the last equation, the area seen by the flow upstream is taken to be a rectangle with height equal to the cylinder diameter and unit width. The resulting pressure drag coefficient values are shown in Table 3.2. As a preliminary verification and validation of results,[21] pressure drag coefficient values obtained using IFISS are compared to Fluent simulation values from Kim and Choudhury.[14] The value is also close to experimental values from Schlichting.[24]

$$\begin{aligned} \frac{\vec{F_p}}{\rho} &= \int_0^s \frac{p}{\rho}(-\hat{n})ds \\ \frac{F_{D_p}}{\rho} &= \frac{\vec{F_p}}{\rho} \cdot \hat{i} \\ C_{D_p} &= \frac{F_{D_p}}{\frac{1}{2}\rho u_{\text{inf}}^2 D(1)} \end{aligned} \tag{3.1}$$

Table 3.3: Table of pressure drag coefficient values for the wide, circular, and narrow cylinder design cases.

Mesh	C_{D_p}		
-	**Wide cylinder**	**Circular cylinder**	**Narrow cylinder**
Coarse	0.981176035504123	1.011044610084666	1.043848441343592
Medium	1.039944478649006	1.081169791522556	1.127018391350748
Fine	1.048739589051556	1.091937471558712	1.139844625646836
Ultra-fine	1.047936608993791	1.091490472518732	1.139779333955020
Ultra-fine+	1.046447358362873	1.090059050155277	1.138415852734465
Reference Data			
Simulation (Kim)[14]	-	1.097	-

As seen in the pressure drag coefficient table, the circular cylinder results agree well with the published simulation. Additionally, the pressure drag coefficient values consistently decrease with increasing a. A plot of the pressure drag coefficient values with varying a is shown in Figure 3.22. The gradual convergence of the pressure drag coefficient with mesh refinement resolution is evident in the figure, as increasing mesh resolution drives the curve toward the top-right of the plot. The pressure drag coefficient is also largely linear for the a interval studied here, which supports the use of finite differences in the calculation of pressure drag

coefficient sensitivities.

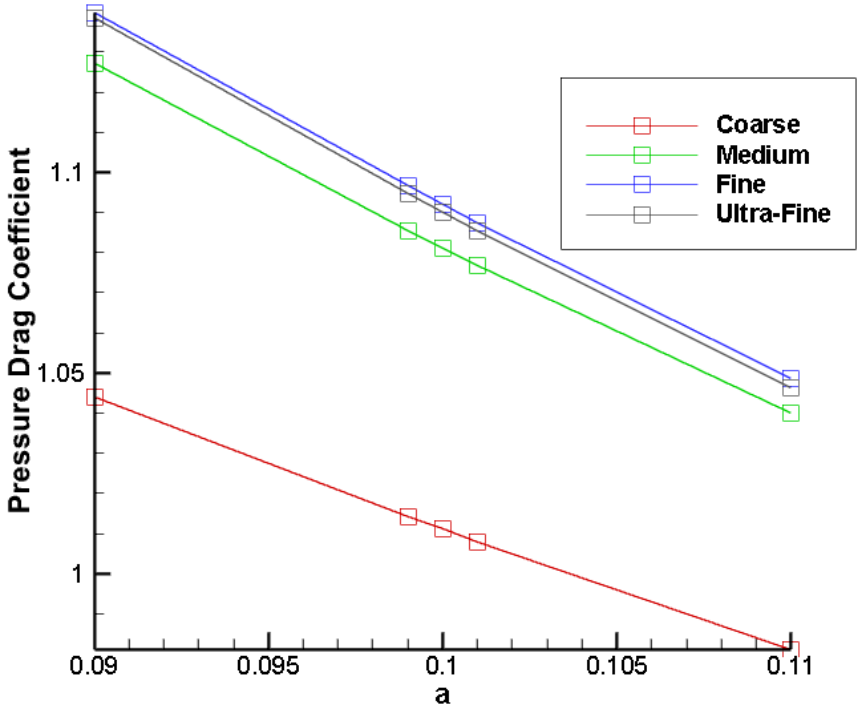

Figure 3.22: Plot of decreasing pressure drag coefficient with increasing a.

Continuum sensitivity results were then compared to those obtained using finite differences. The process of determining continuum sensitivities involved five major steps. In the first step, the sensitivities of the x- and y-coordinates of the cylinder surface to a change in a were calculated. The equation for the surface of an ellipse is shown in Equation 3.2. For this case, the cylinder surface was parameterized such that the x-coordinates surface shifted, but the y-coordinates of the surface themselves did not change with deformation of the surface. The surface derivatives were determined using Equations 3.2.

$$
\begin{aligned}
&(\frac{x}{a})^2 + (\frac{y}{b})^2 - 1 = 0 \\
&x = \pm\frac{a}{b}\sqrt{b^2 - y^2} \\
&\frac{dx}{da} = \left\{ \begin{array}{ll} \frac{1}{b}\sqrt{b^2 - y^2}, & x > 0 \\ -\frac{1}{b}\sqrt{b^2 - y^2}, & x < 0 \end{array} \right\}
\end{aligned}
\tag{3.2}
$$

The second step needed to calculate the pressure drag coefficient continuum sensitivities was to determine the spatial derivatives of the u- and v-velocities as well as the density-normalized pressure at each surface node. This was accomplished using Equations 3.3 from Ragab.[22] In the equations, ψ_i is the geometric parameterization shape function, $[J]_{element}$ is the element Jacobian matrix, $[xy]_{element}$ is the two-dimensional matrix containing the x- and y-coordinates of the element nodes, and $\{u\}$ contains the velocity values at each of the element nodes. Additionally, $[\tilde{B}]$ is evaluated at the point where the spatial derivatives are desired using the master coordinates ζ and η of the appropriate element. It should be noted that this process needed to be performed for density-normalized pressure separately from the u- and v-velocities due to the different element orders.

$$
\begin{aligned}
[\tilde{B}] &= \begin{bmatrix} \frac{\partial \psi_1}{\partial \zeta} & \frac{\partial \psi_2}{\partial \zeta} & \cdots \\ \frac{\partial \psi_1}{\partial \eta} & \frac{\partial \psi_2}{\partial \eta} & \cdots \end{bmatrix} \\
[J]_{element} &= [\tilde{B}][xy]_{element} \\
[B] &= [J]^{-1}_{element}[\tilde{B}] \\
\left\{ \begin{array}{l} \frac{\partial u}{\partial x} \\ \\ \frac{\partial u}{\partial y} \end{array} \right\} &= [B]\{u\}_{element}
\end{aligned}
\tag{3.3}
$$

Once the second step was completed, the Navier-Stokes equations could be solved again, where boundary conditions were in terms of the local sensitivity of velocity to the change in a. A Neumann boundary condition would still be used for the outlet, and the local velocity sensitivities to a would be zero at the inlet and top and bottom walls. It is known that the total sensitivity of velocity to changes in a at the surface of the cylinder be zero; as a result, the boundary condition used for the new simulation was given by Equation 3.4. As a reminder, the surface was parameterized such that the y-coordinates of the nodes are constant; as such, only derivatives involving x were needed.

$$\frac{\partial u}{\partial a} = -(\frac{\partial u}{\partial x}\frac{\partial x}{\partial a}) \tag{3.4}$$

Once the new simulation has been performed, the overall sensitivity of pressure to a is calculated, using Equation 3.5.

$$\frac{dp}{da} = \frac{\partial p}{\partial a} + \frac{\partial p}{\partial x}\frac{\partial x}{\partial a} \tag{3.5}$$

The final major step for calculating analytical pressure drag coefficient sensitivities was to determine the sensitivity of the force on the cylinder to a, isolate the density-normalized pressure drag force, and divide by the appropriate quantities. This part of the process was largely the same as the one used to find the pressure drag coefficient itself. However, both sides of the earlier force equation would be replaced with their sensitivities, resulting in Equation 3.6. The second integral in the equation is important, because it takes into account the change in normal vectors as well as surface area with the changing shape. The pressure drag coefficient sensitivity to a could then be found by isolating the x-component of $\frac{d\vec{F}}{da}$ and dividing by the product of the density-normalized dynamic pressure and the rectangular area seen by the flow.

$$\frac{d(\frac{\vec{F}}{\rho})}{da} = \frac{d}{da}\int_s (\frac{p}{\rho})(-\hat{n})ds = \int_s \frac{d(\frac{p}{\rho})}{da}(-\hat{n})ds + \int_s (\frac{p}{\rho})(-\frac{d(\hat{n}ds)}{da}) \tag{3.6}$$

A comparison of the analytically-determined pressure drag coefficient sensitivities to those calculated using second-order central finite differences are shown in Table 3.4; the pressure drag coefficient values for the wide and narrow elliptical cylinders are used to determine a second-order central finite difference using Equation 3.7.

$$\frac{DC_{D_p}}{Da} \approx \frac{C_{D_{p_{wide}}} - C_{D_{p_{narrow}}}}{a_{wide} - a_{narrow}} \tag{3.7}$$

Table 3.4: Table of pressure drag coefficient sensitivities; values calculated using analytic continuum sensitivity analysis are compared to those obtained using numeric finite differences.

Mesh	**Continuum**	**Finite Difference**	**% Difference**
Coarse	-2.922161485672502	-3.13362029197345	6.748067302301554
Medium	-2.214076013341719	-4.35369563508711	49.144905870356716
Fine	-3.491227216684973	-4.55525182976400	23.358195174341194
Ultra-Fine	-4.112366619799638	-4.59213624806145	10.447634877217457
Ultra-Fine+	-4.378985492912293	-4.598424718579596	4.772052150395655

As seen in the table, the continuum sensitivity results converge more slowly than the finite difference results; this slower convergence is believed to be as a result of inaccurate calculation of the spatial gradients. Based on the trend, it is likely that the good agreement of the coarse mesh is by chance.

To test the suspected spatial gradient inaccuracy concerns, a study was performed regarding velocity and

velocity gradients along the centerline ahead of the cylinder. A plot of the velocity profile along this line is shown in Figure 3.23, with expanded views in Figures 3.24 and 3.25.

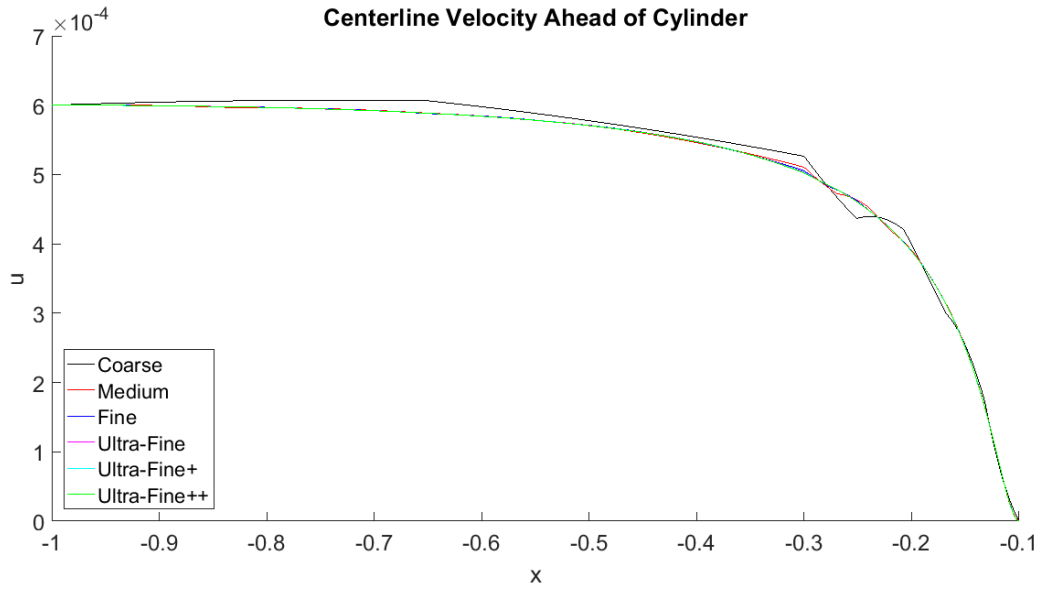

Figure 3.23: Velocity along centerline ahead of circular cylinder.

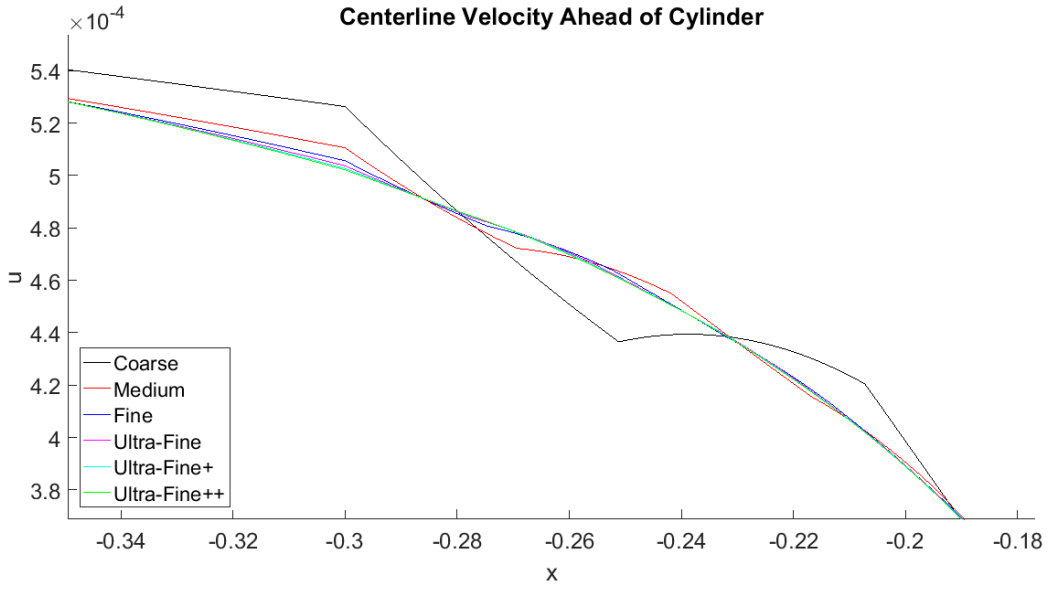

Figure 3.24: Expanded view of circular cylinder centerline velocity profile, with focus on the interface of mesh sections.

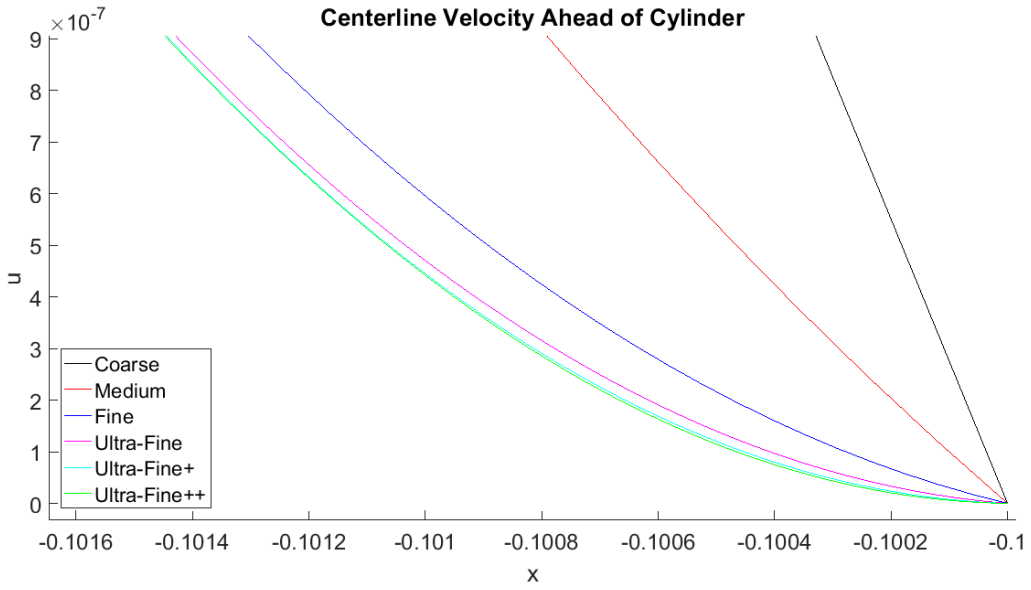

Figure 3.25: Expanded view of circular cylinder centerline velocity profile, with focus on velocity close to the leading edge.

As seen in the above figures, some error is introduced into the simulation due to the abrupt transition from large elements closer to the far-left freestream to small elements after x = -0.3. The velocity correspondingly jumps at this point and experiences sporadic behavior before smoothing out close to the cylinder surface. Increasing mesh refinement dramatically decreases the magnitude of the sporadic behavior but does not completely remove it. The coarser meshes result in much larger spatial velocity gradients at the cylinder surface, indicating that the precise behavior of velocity necessary for sensitivity calculations is much more reasonably captured by finer meshes.

Information was also collected at the surface of the cylinder regarding the gradients of pressure and normal components of velocity at the leading edge and top of the cylinder. The determined information for the two points are shown in Tables 3.5 and 3.6.

Table 3.5: Table of normal spatial gradients at the circular cylinder leading edge point.

	$\frac{\partial u}{\partial x}$	$\frac{\partial p}{\partial x}$
Coarse	-0.002723601838921	1.098098496346056e-06
Medium	-9.705824892770327e-04	1.152688658310785e-06
Fine	-2.683128118335103e-04	1.449269028807193e-06
Ultra-Fine	-8.932224555430733e-05	1.604507244526730e-06
Ultra-Fine+	-3.563455253552969e-05	1.685302606936736e-06
Ultra-Fine++	-1.701527657633562e-05	1.729242574522436e-06

Table 3.6: Table of normal spatial gradients at the circular cylinder top point.

	$\frac{\partial v}{\partial y}$	$\frac{\partial p}{\partial y}$
Coarse	0.001901023954449	1.195836611701426e-06
Medium	5.384641269781240e-04	9.677504141040243e-07
Fine	1.446220552359040e-04	1.033373124817503e-06
Ultra-Fine	4.883519892216531e-05	1.081143932245807e-06
Ultra-Fine+	1.992587121415009e-05	1.110152187797894e-06
Ultra-Fine++	9.811836359002323e-06	1.127066595677423e-06

As seen in the tables, the spatial velocity gradient magnitudes decrease significantly with increasing mesh refinement, while the pressure gradient magnitudes have a more or less first order convergence. More specifically, the velocity gradient values approach zero, meaning that while velocity itself is required to be zero at the surface, the gradient of the normal component of velocity in the normal direction is not automatically zero but rather converges to zero with increasing mesh refinement. Additionally, the pressure gradients behave curiously with the two coarsest meshes, suggesting that the precise pressure spatial gradient behavior is not captured well with those two meshes.

In addition to the determination of the pressure drag coefficient sensitivities and velocity and pressure behavior for sensitivity analysis, a preliminary step size study was performed using only the ultra-fine pressure drag coefficient data. As noted earlier, too small of a step size used to calculate finite sensitivities results in erroneous results due to subtractive cancellation, and too large of a step size results in secant error. Sensitivities were approximated second-order central finite differences with Δa values of $10\%a$, $3\%a$, $1\%a$, $0.3\%a$, and $0.1\%a$; the equation for the finite difference calculation is shown in Equation 3.7. The results of the step size study are shown in Figure 3.26.

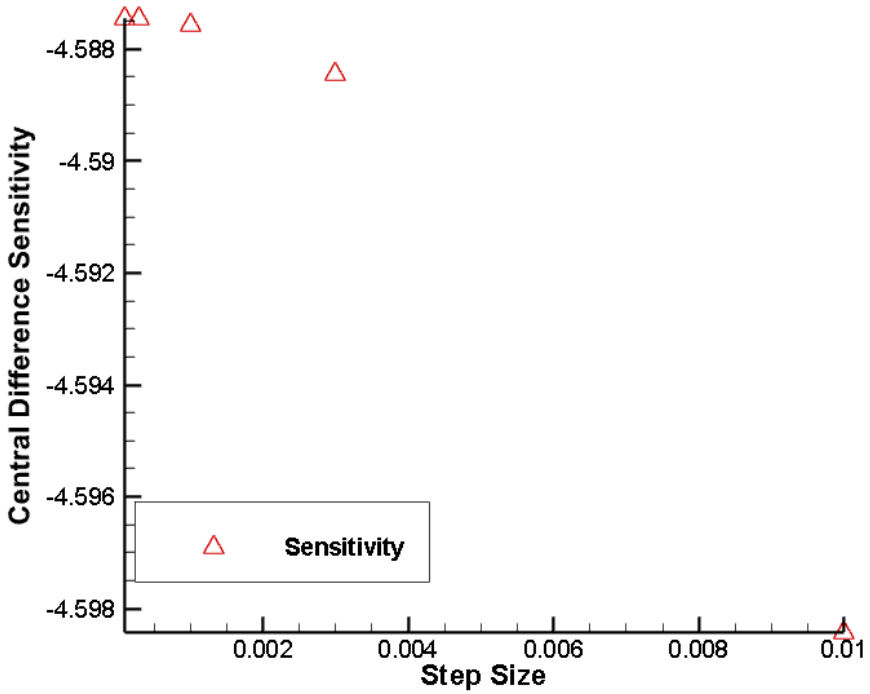

Figure 3.26: Plot of pressure drag coefficient sensitivity approximation versus step size.

All five step sizes shown in Figure 3.26 give similar values for the pressure drag coefficient sensitivity. There is less than a 1% difference between the results of $10\%a$ and $0.1\%a$, indicating both that a step size as small as $0.1\%a$ does not suffer significantly from subtractive cancellation and that a step size as large as 10% gives

an acceptable tangent approximation.

Chapter 4

Conclusion and Future Work

4.1 Conclusion

Simulations were performed with meshes of varying refinement for flow over a backward-facing step, and both velocity values and their sensitivities to the maximum inlet velocity magnitude were determined along the data sampling line and verified using finite difference and complex step methods. Pressure drag coefficient values on the surface of a circular cylinder were calculated using multiple meshes of differing refinements and compared to pressure drag coefficient values obtained for elliptical cylinders with marginally increased or decreased x-direction radius lengths; the pressure drag coefficient results for the circular cylinders were also verified using results from Fluent simulations and validated using experimental data. The sensitivities of pressure drag coefficient to the x-direction radius length were calculated analytically for the different refinements and compared to the second-order central finite difference results. The results indicate that continuum sensitivity analysis gives good results but convergence is slow.

4.2 Future Work

In future work, the author plans to further investigate the methods used to determine and understand the source of the differences in analytical and finite difference pressure drag coefficient sensitivity values. One topic the author intends to pursue is spatial gradient recovery, the purpose of which is to increase the accuracy of spatial gradients used in the overall sensitivity calculations.

Appendices

Appendix A

Backward-Facing Step Meshes

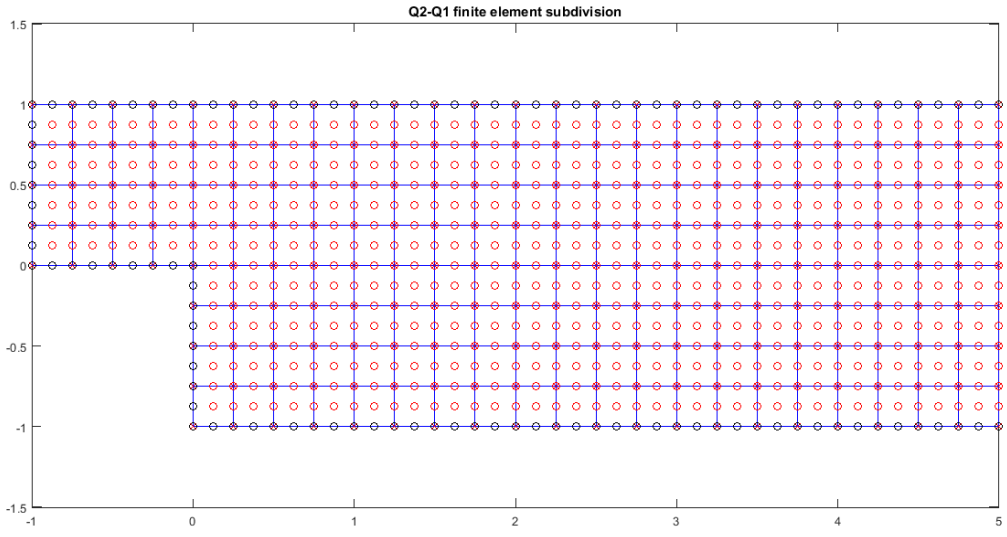

Figure A.1: Coarse uniform structured mesh produced in IFISS for the backward-facing step grid convergence test.

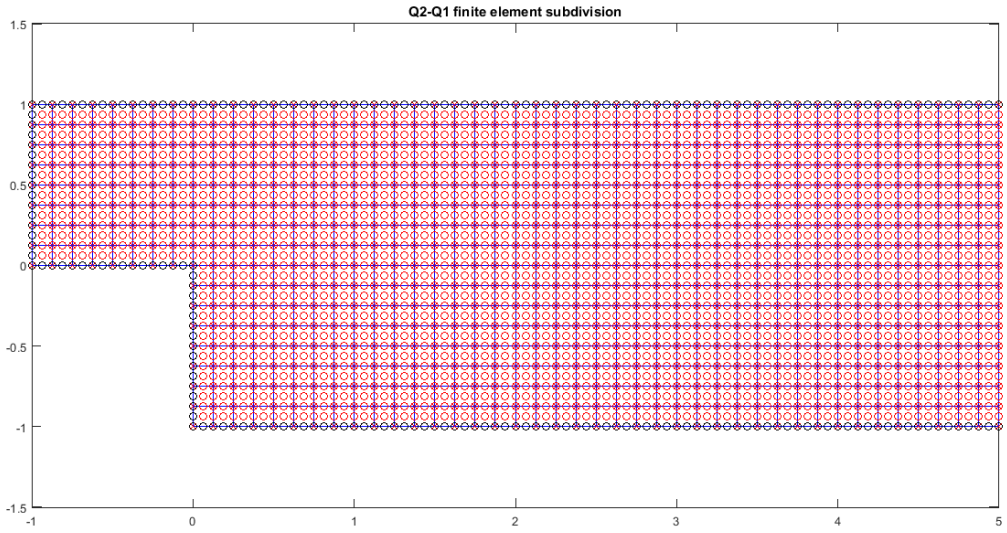

Figure A.2: Medium uniform structured mesh produced in IFISS for the backward-facing step grid convergence test.

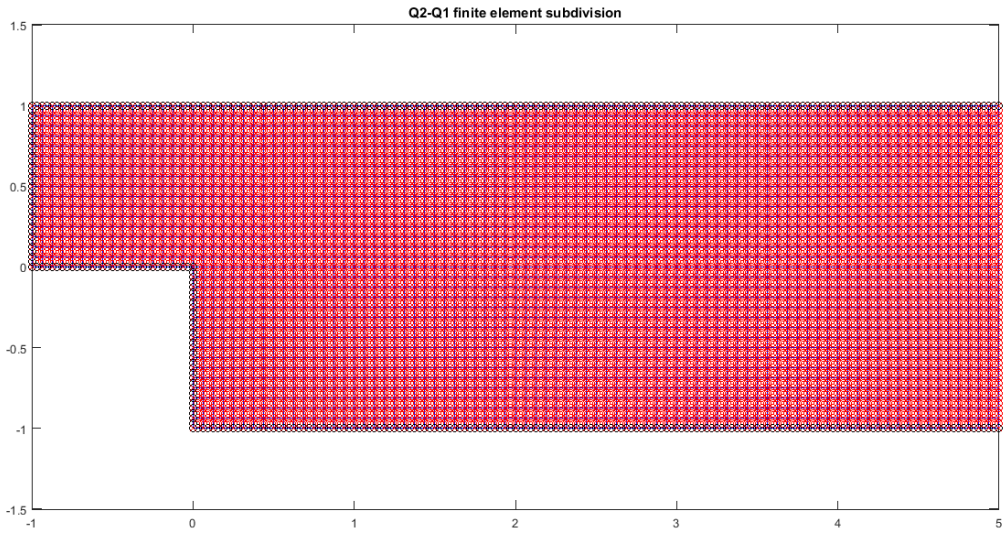

Figure A.3: Fine uniform structured mesh produced in IFISS for the backward-facing step grid convergence test.

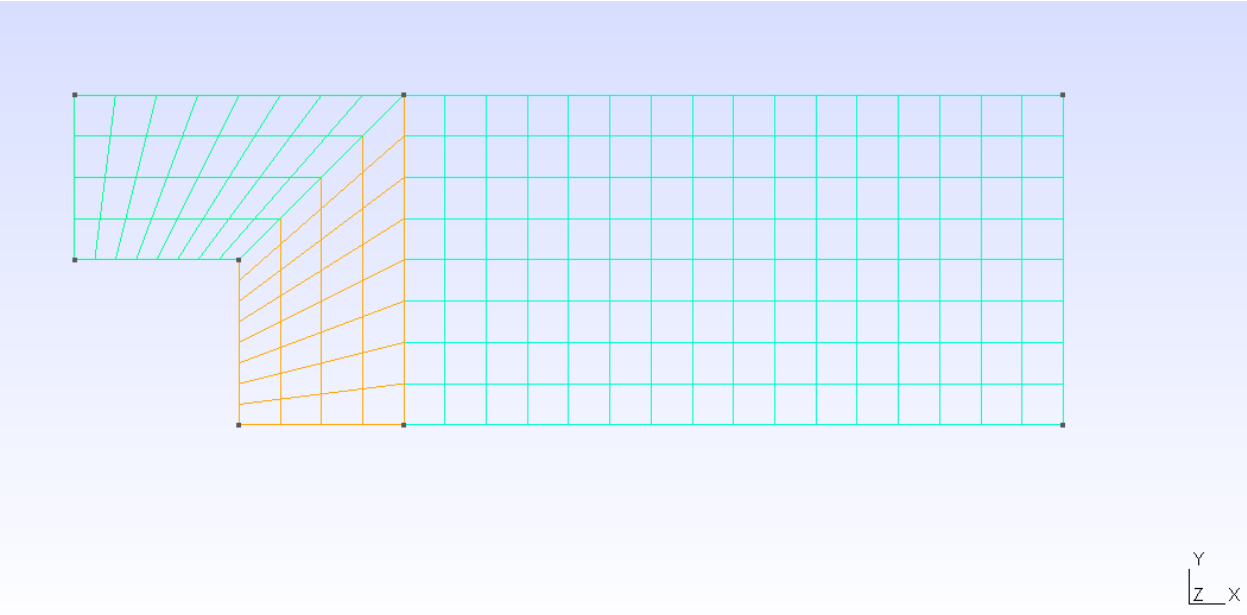

Figure A.4: Coarse non-uniform structured mesh produced in IFISS for the backward-facing step grid convergence test.

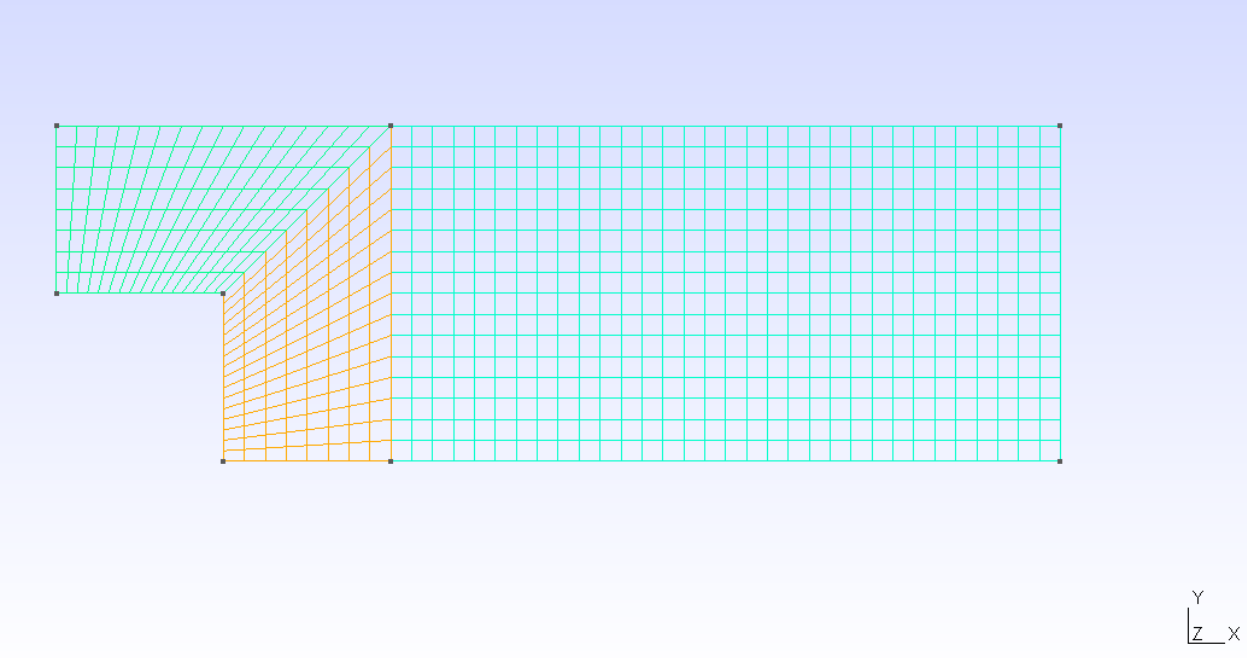

Figure A.5: Medium non-uniform structured mesh produced in IFISS for the backward-facing step grid convergence test.

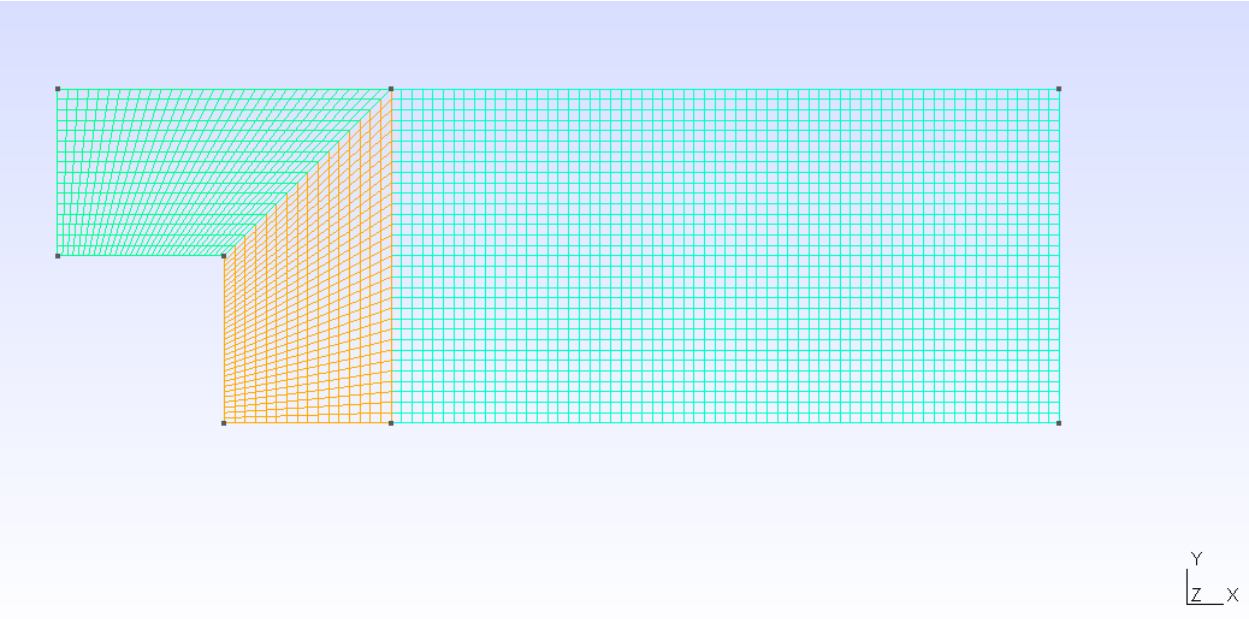

Figure A.6: Fine non-uniform structured mesh produced in IFISS for the backward-facing step grid convergence test.

Appendix B

Cylinder Meshes

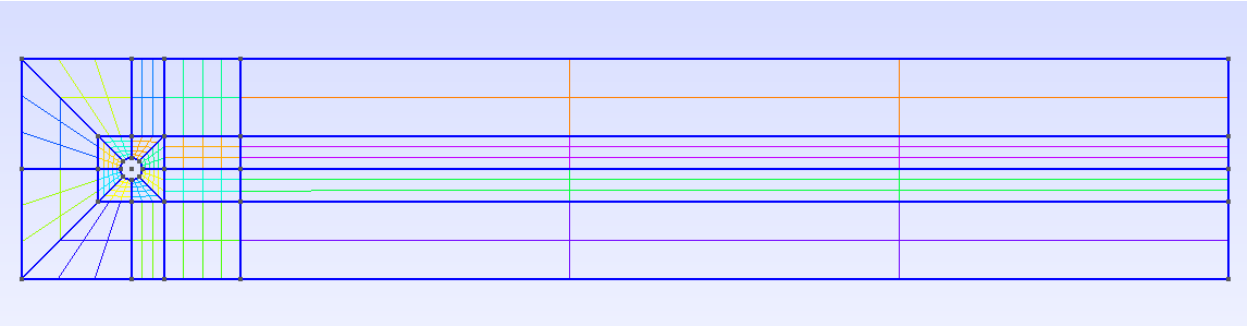

Figure B.1: Coarse circular cylinder mesh generated in Gmsh for shape optimization cases.

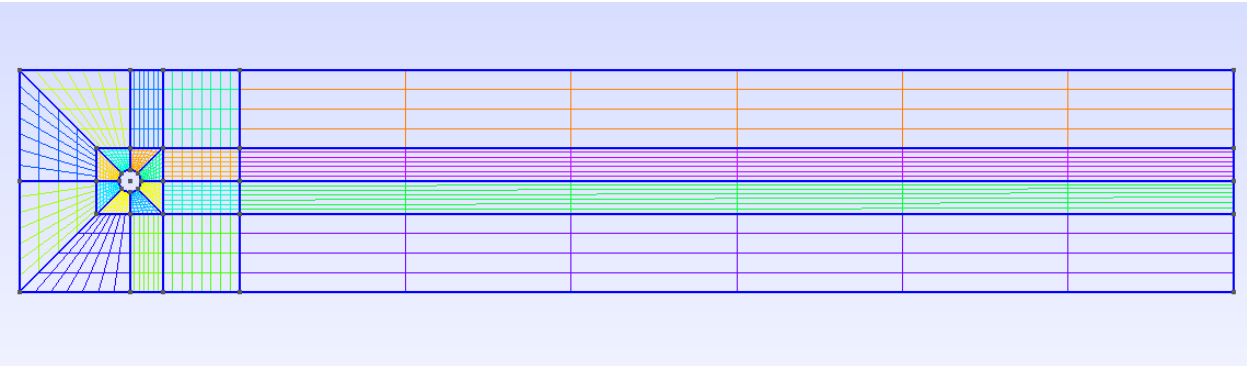

Figure B.2: Medium circular cylinder mesh generated in Gmsh for shape optimization cases.

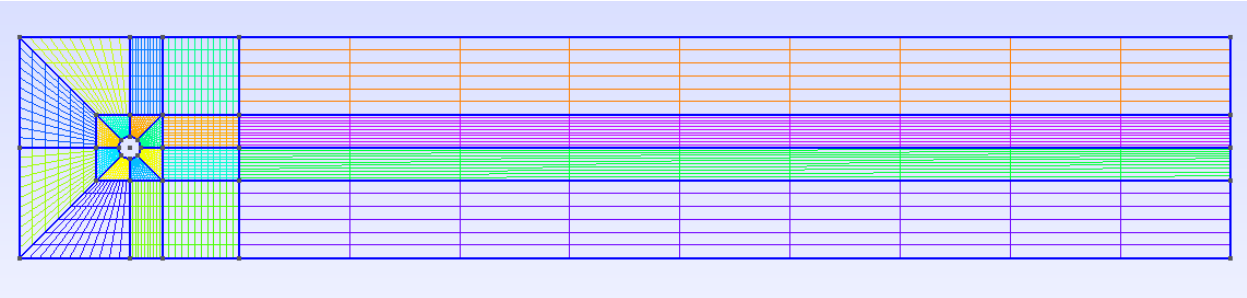

Figure B.3: Fine circular cylinder mesh generated in Gmsh for shape optimization cases.

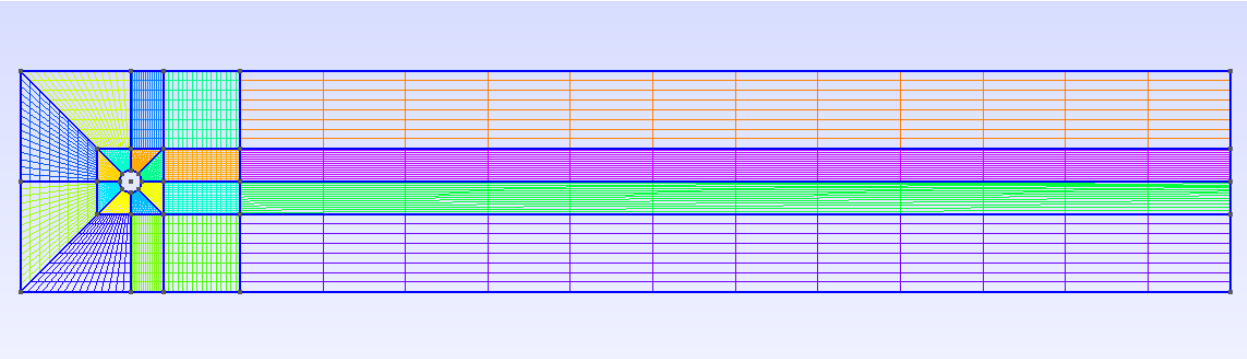

Figure B.4: Ultra-fine circular cylinder mesh generated in Gmsh for shape optimization cases.

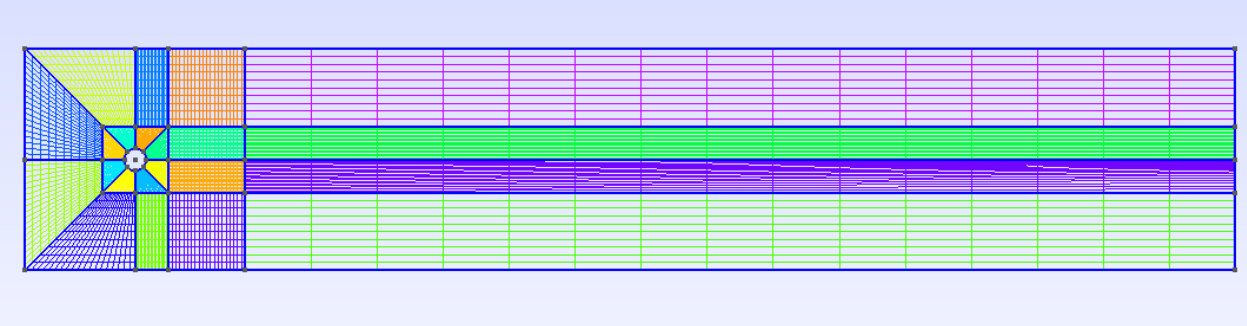

Figure B.5: Ultra-fine+ circular cylinder mesh generated in Gmsh for shape optimization cases.

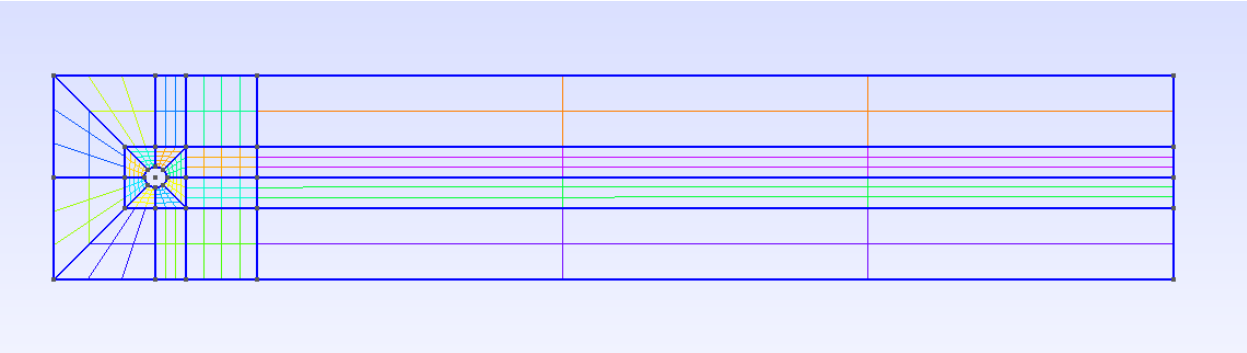

Figure B.6: Coarse wide elliptical cylinder mesh generated in Gmsh for shape optimization cases.

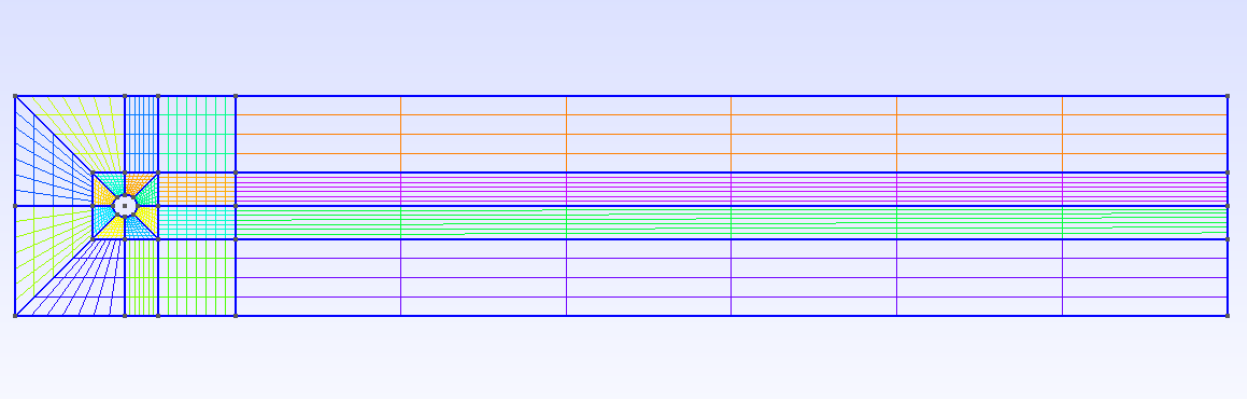

Figure B.7: Medium wide elliptical cylinder mesh generated in Gmsh for shape optimization cases.

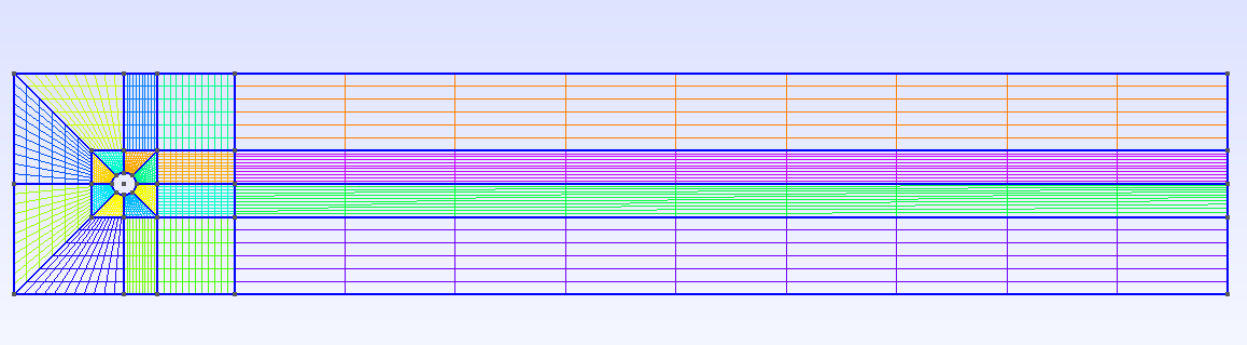

Figure B.8: Fine wide elliptical cylinder mesh generated in Gmsh for shape optimization cases.

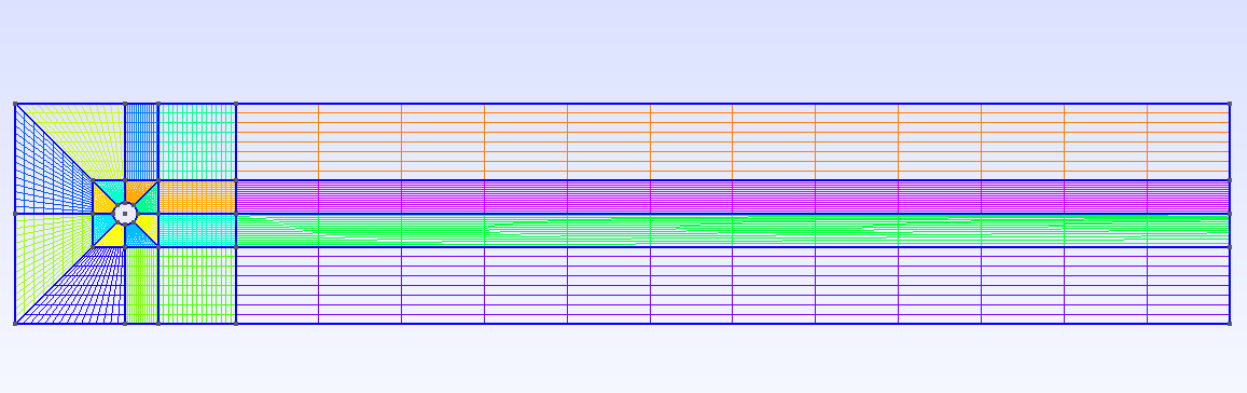

Figure B.9: Ultra-fine wide elliptical cylinder mesh generated in Gmsh for shape optimization cases.

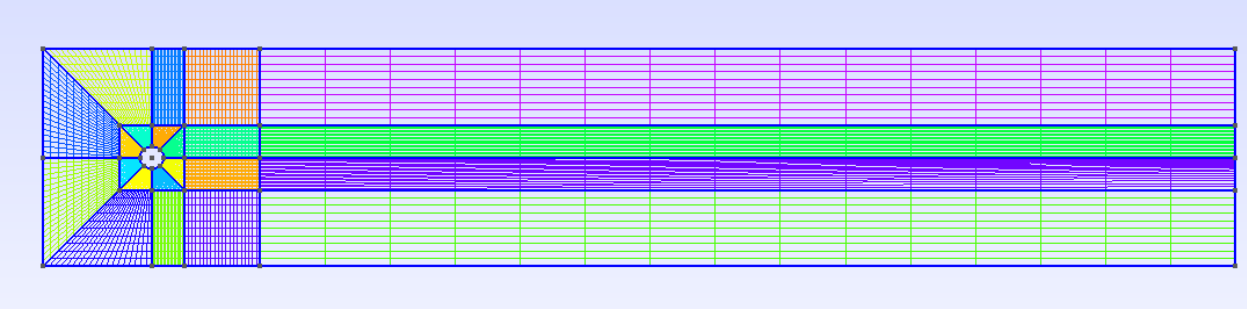

Figure B.10: Ultra-fine+ wide elliptical cylinder mesh generated in Gmsh for shape optimization cases.

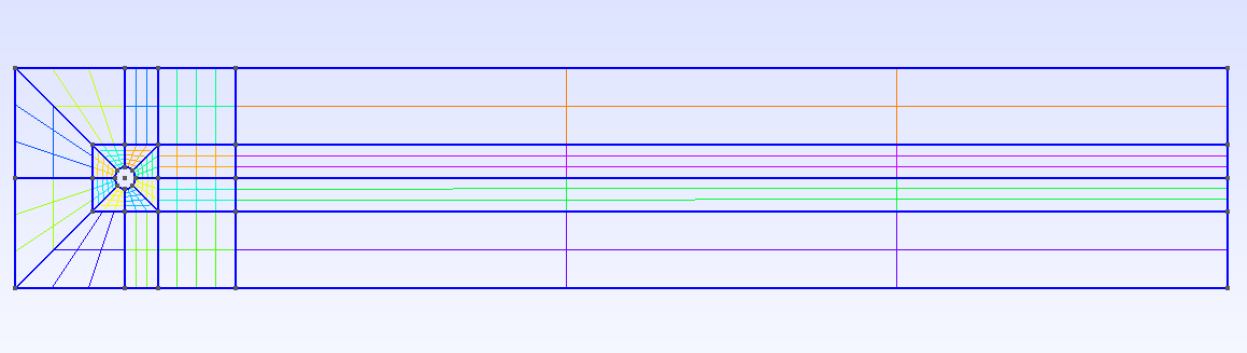

Figure B.11: Coarse narrow elliptical cylinder mesh generated in Gmsh for shape optimization cases.

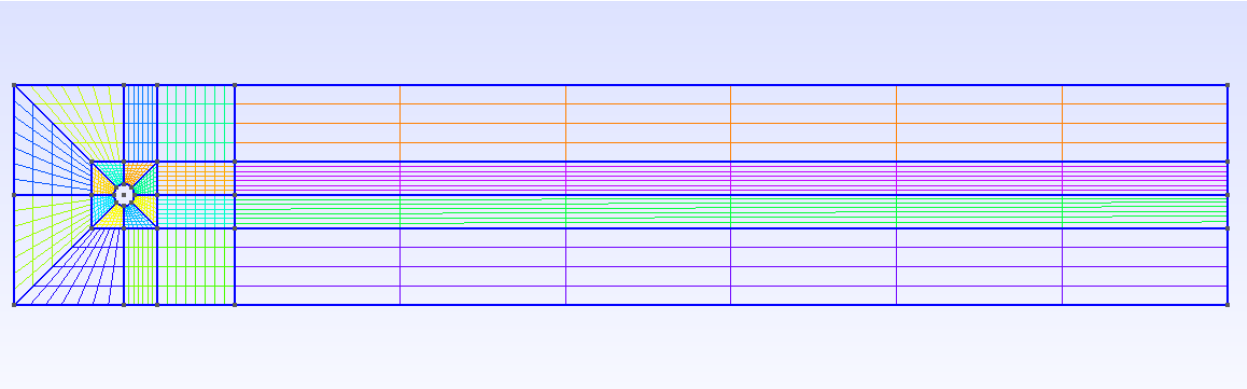

Figure B.12: Medium narrow elliptical cylinder mesh generated in Gmsh for shape optimization cases.

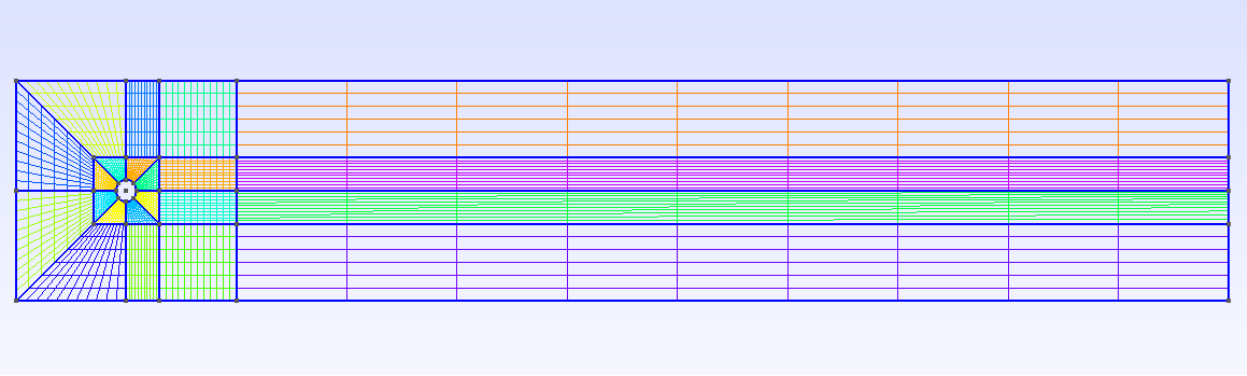

Figure B.13: Fine narrow elliptical cylinder mesh generated in Gmsh for shape optimization cases.

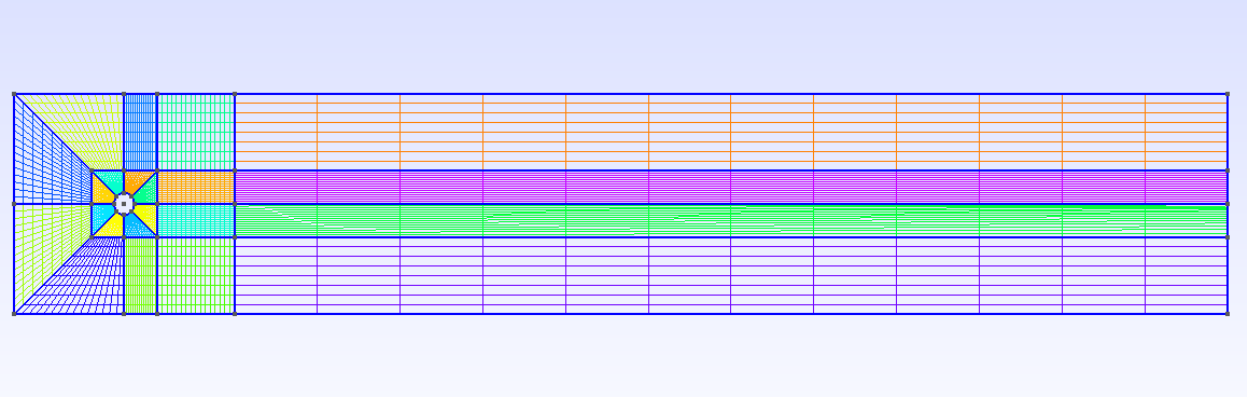

Figure B.14: Ultra-fine narrow elliptical cylinder mesh generated in Gmsh for shape optimization cases.

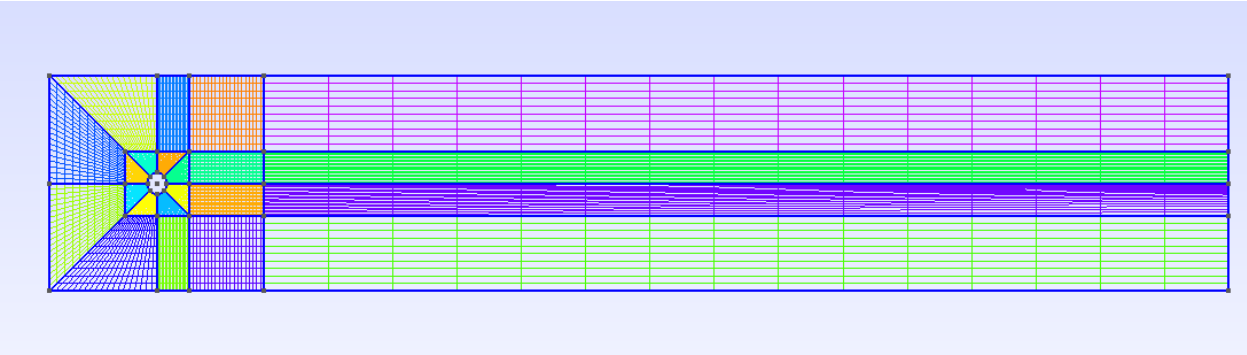

Figure B.15: Ultra-fine+ narrow elliptical cylinder mesh generated in Gmsh for shape optimization cases.

Bibliography

[1] Arbogast, Todd, et al. "A multiscale mortar mixed finite element method." *Multiscale Modeling & Simulation* 6.1 (2007): 319-346.

[2] Arora, Jasbir S., and Edward J. Haug. "Methods of design sensitivity analysis in structural optimization." AIAA journal 17.9 (1979): 970-974.

[3] Arora, Jasbir S., and Edward J. Haug. "Reply by Authors to GN Vanderplaats." AIAA Journal 18.11 (1980): 1407-1408.

[4] Bertin, John J. *Aerodynamics for Engineers*. Prentice Hall, Upper Saddle River, NJ, 2009.

[5] Blackwell, Ben F., et al. *Utilization of sensitivity coefficients to guide the design of a thermal battery*. No. SAND–98-1817C; CONF-981107–. Sandia National Labs., Albuquerque, NM (United States), 1998.

[6] Borggaard, J., D. Pelletier, and . Turgeon. "A study of optimal cooling strategies in thermal processes." 38th Aerospace Sciences Meeting and Exhibit. 2000.

[7] Chen, Yanping. "Superconvergence of Mixed Finite Element Methods for Optimal Control Problems. *Mathematics of Computation*, vol. 77, no. 263, 2008, pp. 12691291., www.jstor.org/stable/40234558.

[8] Duvigneau, R., and D. Pelletier. "On accurate boundary conditions for a shape sensitivity equation method." *International journal for numerical methods in fluids* 50.2 (2006): 147-164.

[9] Elman, Howard C., Alison Ramage, and David J. Silvester. "Algorithm 866: IFISS, a Matlab toolbox for modelling incompressible flow." ACM Transactions on Mathematical Software (TOMS) 33.2 (2007): 14.

[10] Elman, Howard C., David J. Silvester, and Andrew J. Wathen. *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*. Oxford University Press, Oxford, United Kingdom, 2014, doi:10.1093/acprof:oso/9780199678792.001.0001.

[11] Godfrey, Andrew, William Eppard, and Eugene Cliff. "Using sensitivity equations for chemically reacting flows." *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. 1998.

[12] Godfrey, Andrew, and Eugene Cliff. "Sensitivity equations for turbulent flows." *39th Aerospace Sciences Meeting and Exhibit*. 2001.

[13] Kim, Mi-Young, et al. "A multiscale mortar mixed finite element method for slightly compressible flows in porous media." Journal of the Korean Mathematical Society 44.5 (2007): 1103-1119.

[14] Kim, S-E., and D. Choudhury. "A Computational Study of Laminar Flows: Bechmark Calculations Using FLUENT." ASME-PUBLICATIONS-FED 160 (1993): 23-23.

[15] Komkov, Vadim, Kyung K. Choi, and Edward J. Haug. *Design sensitivity analysis of structural systems*. Vol. 177. Academic press, 1986.

[16] Kulkarni, Mandar D. Continuum Sensitivity Analysis using Boundary Velocity Formulation for Shape Derivatives. Diss. Virginia Tech, 2016.

[17] Lee, Myoungkyu, Nicholas Malaya, and Robert D. Moser. "Petascale direct numerical simulation of turbulent channel flow on up to 786k cores." *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. ACM, 2013.

[18] Liu, Shaobin, and Robert Canfield. "Continuum shape sensitivity for nonlinear transient aeroelastic gust response." 52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 19th AIAA/ASME/AHS Adaptive Structures Conference 13t. 2011.

[19] Martins, Joaquim R. *A coupled-adjoint method for high-fidelity aero-structural optimization*. STANFORD UNIV CA DEPT OF AERONAUTICS AND ASTRONAUTICS, 2002.

[20] Massey, B. S. *Mechanics of Fluids*. CRC Press, 2012.

[21] Oberkampf, William L., M. Sindir, and A. T. Conlisk. "Guide for the verification and validation of computational fluid dynamics simulations." *American Institute of Aeronautics and Astronautics*, Reston, VA (1998).

[22] Ragab, S.A. and H.E. Fayed. "Introduction to Finite Element Analysis for Engineers." CRC Press, Taylor & Francis Group LLC (2017).

[23] Reddy, J. N. *An Introduction to the Finite Element Method*. McGraw-Hill, New York, 2006.

[24] Schlichting, Hermann. *Boundary-Layer Theory*. McGraw-Hill, New York, 1968.

[25] Schröder, Andreas, Heiko Kleemann, and Heribert Blum. "Mixed finite element methods for two-body contact problems." Journal of Computational and Applied Mathematics 283 (2015): 58-70.

[26] Silvester, David, Howard Elman, and Alison Ramage. "Incompressible flow and iterative solver software (IFISS)." (2016).

[27] Silvester, David J., and Valeria Simoncini. "An optimal iterative solver for symmetric indefinite systems stemming from mixed approximation." *ACM Transactions on Mathematical Software (TOMS)* 37.4 (2011): 42.

[28] Turgeon, E., D. Pelletier, and J. Borggaard. "A General Continuous Sensitivity Equation Formulation for Complex Flows." , doi:10.2514/6.2000-4732.

[29] Turner, M. Jon. "Stiffness and deflection analysis of complex structures." *Journal of the Aeronautical Sciences* (2012).

[30] von Kármán, Theodore. *Aerodynamics*. Cornell University press: McGraw-Hill company, 1963.

[31] Wheeler, Mary F., and Ivan Yotov. "A multipoint flux mixed finite element method." *SIAM Journal on Numerical Analysis* 44.5 (2006): 2082-2106.