

Computational Studies of Icing on Aircraft Wings

Cuong Tran and Vikram Garg

The University Of Texas At Austin
ASE 463Q
Fall 2006

Advisor: Dr. Graham Carey

January 2, 2007

Executive Summary

Cuong Tran and Vikram Garg
The University Of Texas at Austin
ASE 463Q
October 27, 2006

To Dr. Ronald Stearman:

The objective of this project was to study the effects of icing on the leading edges of wings through numerical simulation. The Finite Element Method was employed to solve the Navier Stokes equations that govern the physics of the flow problem. Unstructured triangular meshes were used to discretize the physical domain around the clean and iced airfoils. Numerical stabilization was added to the Finite Element formulation. The CFD code was assembled and debugged at the Computational Fluid Dynamics Lab in the Aerospace Engineering department. Graphs of lift and drag coefficients versus Reynolds number and angles of attack were prepared to compare the performance of the clean and iced airfoils.

The report gives a background on the icing problem as well as previous work done to understand and tackle it. The FEM formulation of the governing equations is derived along with a section on the numerical stabilization and its implementation. Results for steady flow at low and moderate Reynolds numbers are then presented. Unsteady flow results for moderate Reynolds numbers are also presented. We conclude with recommendations for future work that could be undertaken to further this project.

Vikram Garg
simulationist@gmail.com

Cuong Tran
durtyslanteye@mail.utexas.edu

Abstract

The objective of this project was to numerically analyze the effects of icing damage on the flow field over an airfoil. With the aid of software such as Matlab, TRIANGLE, and Libmesh, two-dimensional flow simulations were performed that compare the flow fields over a clean and an iced airfoil. A numerically stabilized, adaptive Finite Element method was used to find approximate solutions to the equations that govern the actual flow field. Parametric studies were conducted at various angles of attack and Reynolds numbers, with and without icing damage, in order to determine the impact of icing. Key aerodynamic characteristics such as the pressure distribution and the lift, drag, and moment coefficients were recovered by writing appropriate post-processing routines. Results at low Reynolds numbers indicated that the viscous forces dominated the flow-field and at low angles of attack, the viscous drag component was greater than the pressure drag component. Since the airfoil tested was designed for much higher Reynolds numbers, we concluded that a turbulence model shall be required to perform simulations at higher Reynolds numbers. The code that has already been developed and future additions like a turbulence model will provide the Aerospace Engineering department with a robust, reliable simulation package to study icing damage.

Acknowledgments

We could not have moved as far in this project without the help of the following members of the CFD lab group at the University of Texas at Austin:

Dr. Ronald Stearman: We thank Dr. Stearman for allowing us to undertake this non-conventional project and for his discussions on the fluid-structure interaction aspects of this problem.

Dr. Graham Carey: We wish to thank our adviser Dr. Carey for his guidance throughout our work.

Roy Stogner: Roy Stogner is the creator of the 'DiffSystem' framework within Libmesh which forms the underlying structure of our Finite Element Code. The code for our project could not have been written, debugged and validated as efficiently as it was without this framework. We also thank Roy for his informative discussions with us throughout the project.

We also wish to thank John Peterson and Derek Gasto, who are also Libmesh developers, for their help and guidance in using Libmesh.

Contents

1	Introduction	6
1.1	Project Goals	6
1.2	Background Information	6
1.2.1	Icing Damage	6
1.2.2	Motivation	7
1.2.3	Previous Work	8
1.3	Relevance to Engineering Design	8
1.4	Team Organization	10
2	Theory	11
2.1	Governing Equations	11
2.2	Finite Element Formulation	11
2.3	Numerical Stabilization	12
2.4	Expressions for Computing Aerodynamic Coefficients	13
3	Progress	15
3.1	Preliminary Work	15
3.1.1	Meshing	15
3.1.2	Code Validation	17
3.1.3	Validation of Domain Size	17
3.1.4	Grid Sensitivity Studies	18
3.2	Results	19
3.2.1	Steady Flow Results	19
3.2.2	Unsteady Flow Results	21
3.3	Cost Analysis	23
4	Conclusions	24
5	Future Work	25
5.1	Turbulence Modeling	25
5.2	Fluid-Structure Interaction code	25
6	Appendix	27

List of Figures

- Figure 1. Icing on leading edge of a wing
- Figure 2. Unstructured mesh around the ice horn
- Figure 3. Mesh after adaptive mesh refinement
- Figure 4. Backwards Facing Step Results
- Figure 5. Domain Sensitivity Studies at $Re = 1000$, $\alpha = 10$, clean airfoil
- Figure 6. Grid Sensitivity Studies at $Re = 1000$, various α , clean airfoil
- Figure 7. Airfoil performance dependence on Re number, $\alpha = 3$
- Figure 8. Recirculation region behind the ice horn, $\alpha = 3$, various Re
- Figure 9. Separation region on the top of the airfoil at $Re = 1000$, various α
- Figure 10. Airfoil performance dependence on α , Re number = 1000
- Figure 11. Airfoil performance dependence on time, Re number = 1000, $\alpha = 10$
- Figure 12. Separation region on the top of the airfoil at $Re = 1000$, $\alpha = 10$
- Figure 13. Viscous and Pressure drag components at $Re = 1000$

List of Tables

Table 1: Schedule Chart

1 Introduction

1.1 Project Goals

The objective of this project is to model and numerically analyze the effects of icing damage on the flow field over an airfoil. Using resources from the Computational Fluid Dynamics Laboratory at the University of Texas at Austin, we designed and implemented a Navier-Stokes flow solver that allowed for an extensive analysis of a two-dimensional airfoil's surrounding flow field that was dependent upon specific control parameters. The differences between an iced airfoil's flow field and a clean airfoil's flow field was then compared.

To implement the flow solver, we must first formulate and implement a Finite Element solution of the mathematical model. Two programs, TRIANGLE and Libmesh, successfully model an airfoil's flow field by creating a detailed, triangular mesh around the airfoil and solving the associated non-linear partial differential equations that govern the flow field. More details pertaining to the Finite Element formulation are available in section 1.3.2. After preparing a flow solver, simulations are then performed with different input parameters.

Data extraction from the simulations was used to compute aerodynamic characteristics such as velocity profiles, boundary layer thickness, and pressure distributions. The computation of performance parameters like the coefficient of lift and the coefficient of drag was also performed. The impact of icing on an airfoil's flow field was then validated by performing a parameter sensitivity analysis. Thus, a comparative analysis between an iced airfoil and a clean airfoil was performed.

It is hoped that the developed code will augment the experimental studies of wing icing in the Aerospace Engineering department. The user needs only to define the control parameters (Reynolds number, angle of attack) to perform simulations. The developed code will then yield various aerodynamic characteristics such as the lift and drag coefficients and allow for the determination of trends in the data.

1.2 Background Information

This section describes the accumulation of icing damage on wings and its effects on aircraft performance. It also elaborates on the motivation for the computational study of this problem and its relevance for design. Finally, we discuss previously completed experiments performed by other teams.

1.2.1 Icing Damage

Pilots can encounter icing damage during any season since icing occurs in any cloud where temperatures drop below freezing. The super-cooled water droplets found in these clouds rapidly freeze and accumulate on wings [2]. Initially, only a thin layer of water freezes to the airfoil's leading edge surface. However, additional layers of droplets

flash-freeze and accumulate atop one another. A protruding ice horn eventually develops and creates severe alterations to the geometry of the airfoil, as shown in Figure 1.



Figure 1: Icing on leading edge of a wing

The addition of a protruding ice horn causes the airflow around the wing to become noticeably distorted, and several negative effects arise as a result. The negative effects include a loss in lift, an increase in drag, a lower stall angle, and a loss of stability. These serious effects of icing have already caused numerous crashes for planes without de-icing mechanisms resulting in an unnecessary, climbing death toll. Thorough research of icing damage would prove to be invaluable for design to prevent such fatalities and losses.

1.2.2 Motivation

Two methods are possible when performing flow field analyses: an experimental approach and a numerical approach. Each has its pros and cons and, in reality, both work best in conjunction with one another for verification purposes. This project employs the numerical approach for the following reasons.

On one hand, the experimental approach is limited in its choice of control parameters since certain scenarios are difficult to emulate in a physical laboratory. For example, extremely high Reynolds numbers are difficult to create in a laboratory setting due to equipment and cost limitations. On the other hand, the numerical approach has a much larger range for control parameters and thus results in the ability to perform a thorough optimization analysis.

The experimental approach is also time-consuming by its very nature. Each time an experiment is conducted, a person must physically change the input parameters such as ice horn position, ice horn shape, and free stream velocity. Once the experiment

is performed for a particular scenario, each control parameter must then be manually altered before the next experiment can begin. In comparison, the numerical approach allows for simple and robust data extraction. By merely changing a few variables in the code, control parameters can be easily and quickly modified. A simulation can then be performed that outputs any desired aerodynamic properties such as the lift, drag, and moment coefficients. The ability of such quick design modifications therefore allows for rapid data acquisition that facilitates optimization analysis.

1.2.3 Previous Work

The Federal Aviation Agency possesses a catalogue of different ice accretion shapes. The airfoil currently modeled in our project is a GLC-305 swept airfoil with a 944 glazed ice accretion. Past studies have run similar simulations with the aforementioned airfoil configurations; two particular studies conducted at Iowa State University and at the University of Illinois at Urbana-Champaign are of interest for our project.

The researchers at Iowa State University studied two-dimensional and three-dimensional simulations of clean and iced wings [6]. The wing configurations under investigation included a GLC-305 swept airfoil with a 944 glazed ice accretion - the same configuration that is the focus of our study. Simulations in this study tested the effects on the flow field by varying input parameters, such as angle of attack and ice shape dimensions.

The experiment at the University of Illinois at Urbana-Champaign conducted flow field measurements about an airfoil with leading edge ice shapes [3]. This study tested the flow fields for a GLC-305 swept airfoil configured with both glaze and rime ice shape accumulations. Unlike the Iowa group, the Illinois team ran physical experiments at the NASA Langley Research Center using the Low-Turbulence Pressure Tunnel. Experiments performed in this study varied input parameters such as the angle of attack, ice shape dimensions, and Mach number.

Both studies previously mentioned created several plots of aerodynamics characteristics that will be used for comparison with our own generated results. Our project will serve to verify other groups results or find plausible explanations for any discrepancies that may appear. Eventually, with additional time and commitment, our project will add to the growing field of icing damage analysis.

1.3 Relevance to Engineering Design

In *The Design of Everyday Things*, Dr. Donald A. Norman explains the psychology and behavior of people with a special emphasis on current technology and designs. Throughout the book, Dr. Norman emphasizes the importance of good design methods to ensure the desired outcome, whether the result is a toothbrush that can fit in toothbrush holders or a door that opens in an intuitively correct direction. Similarly, when determining how to negate the negative effects of icing damage on airplane wings, an engineering team must inevitably undertake a design process. As such, the project detailed in this report adheres to the four major principals of good design detailed in

Dr. Norman's book; the aforementioned principals are visibility, a conceptual model, mappings, and feedback [9].

Visibility is a principal that should be followed so that the user can tell the current state of any system. When running the simulations mentioned in this project, the use of a previously created linear solver was essential. The linear solver chosen to evaluate numerically the solution of partial differential equations was PETSc, the Portable Extensible Toolkit for Scientific computation. PETSc is integrated into the Libmesh Finite Element Library we used as the backbone of our code. With visibility in mind, input coding was written so that, during each successive iteration of the solver, a short output screen appeared that listed system values such as the current Reynolds number, angle of attack, number of solutions steps taken, and the time step. If an error occurred during the simulation, for example PETSC being unable to solve a linear system or the non-linear solver failing to converge, the output screen warned the user so that appropriate measures could be executed. Without visibility, the system would instead continue blindly and the user would have difficulty diagnosing errors. Considering the simulations in this experiment consumed between two to seven hours each, the output screen proved to be invaluable in the prevention of wasted computational time. Therefore, the current status of the system was displayed during each iteration of the linear solver, and visibility was incorporated.

The next key design principal outlined in Dr. Norman's book is the requirement that a conceptual model reflect the user's model. In the case of flow field simulation using computational methods, the simulation is the conceptual model, and the user is any airfoil designer. To satisfy this design principal, the simulation must accurately represent the flow field around an accurately captured airfoil and be designed in such a way that the end user can obtain any necessary information. As such, the project members took special care to scan in an actual two-dimensional image of a real airfoil, the GLC-305, and a standard, catalogued ice horn shape, a 944 glazed-ice accretion. When creating the airfoil geometry, we averaged out noisy data and wrote code to test and ensure the integrity of the airfoil shape. In addition to the airfoil geometry, the flow field was also validated by using various benchmark testing procedures. The aforementioned validation steps were essential when considering that the airfoil designer, the user, needs to acquire accurate data before deciding upon a specific airfoil design. Consequently, the conceptual model does, in fact, reflect the user's model.

Dr. Norman also writes about the importance of mappings which will relate what is required to how it is done. For the project, a schedule chart presented an outline of tasks that needed to be accomplished by the end of the semester. Each task was then subdivided into individual mappings that detailed how the task was to be accomplished, whether it was with the use of computer coding or by creating graphs to demonstrate the effects of varying input parameters. In the case of program coding, flowcharts were drawn that mapped the process by which a desired output would be attained. Clearly, the schedule chart and all of its subdirectories served as the map detailing what was required and how it was accomplished.

The final major principal of good design according to Dr. Norman is the use of feedback so that the user knows what has occurred. The most important feedback mechanism we employed was the use of adaptive mesh refinement. This process entailed using an existing solution to decide where the existing mesh should be enriched. It thus enhanced accuracy and minimized the required computational expense. During

the post-processing, Matlab code was written to extract performance characteristics such as the pressure profile, lift, drag, and moment coefficients. These coefficients were outputted in the form of text files which were then used in the creation of graphical representations of the data. Such graphs allowed for simple recognition of trends in the results. Therefore, several methods of feedback allowed the user to know the system's response to his design modifications.

The implementation of the four main principals of good design in this project resulted in a reliable and useful method for analyzing fluid flow over an airfoil. Without a doubt, Dr. Norman's The Design of Everyday Things proved to be a useful guide to consider during all steps of the design process. The advice presented in the book is not limited to only the specific application in this project, in fact, it applies to all facets of engineering design.

1.4 Team Organization

In order to achieve efficiently the project's goals, work was optimally distributed between the team members. Since Vikram Garg possessed prior experience with the Computational Fluid Dynamics laboratory, the programs developed by the CFD team, and the Finite Element method necessary for the project, he was selected as the team leader. Vikram was able to coordinate effectively the project's design path due to his thorough comprehension of the entire project.

Cuong Tran had previous experience working with multiple computer programming languages, and consequently, he was chosen to aid with the creation of any necessary external programming scripts. Cuong Tran conducted parametric studies to extract data from the simulations and created graphs presenting the results. Together, the team analyzed the results, discussed possible explanations, and formed conclusions from the trends in the data.

2 Theory

In this section, we introduce the Navier-Stokes equations that govern the physics of our flow problem. The Finite Element formulation of the governing equations is then derived. Finally the numerical stabilization scheme is introduced and discussed.

2.1 Governing Equations

The incompressible, viscous NS equations for a Newtonian fluid describe fluid flow at subsonic speeds; they are expressed in non-dimensional form as:

$$\nabla \cdot \vec{U} = 0 \text{ on } \Omega \quad (2.1.1)$$

and

$$\vec{U}_t + (\vec{U} \cdot \nabla) \vec{U} + \nabla p - \frac{1}{Re} \Delta \vec{U} = 0 \text{ on } \Omega, \quad (2.1.2)$$

where Ω is the physical domain and $\partial\Omega$ is Ω 's boundary.

Equation (2.1.1) is simply a statement of mass conservation for an incompressible flow, whereas Eq. (2.1.2) represents the momentum balance across a differential control volume. The flow variables are the velocity $\vec{U} = (u, v)$ and pressure p ; together they constitute a complete, Eulerian description of the flow field. To complete the mathematical statement of the problem, the boundary conditions on $\partial\Omega$ have to be specified. $\partial\Omega$ is a union of the sets $\partial\Omega_i$ (inflow), $\partial\Omega_o$ (outflow) and $\partial\Omega_a$ (airfoil boundary). The boundary conditions are:

$$\vec{U} = (1, 0) \text{ on } \partial\Omega_i \times [0, T] \quad (2.1.3) \text{ and}$$

$$\vec{U} = (0, 0) \text{ on } \partial\Omega_a \times [0, T], \quad (2.1.4)$$

which are Dirichlet conditions for all time from 0 to T. The outflow boundary conditions will arise naturally from the Finite Element formulation, as we shall see later.

The non-dimensional parameter Re , Reynolds number, is defined as

$$Re = \frac{\rho U_\infty L}{\nu}, \quad (2.1.5)$$

where ρ and ν are the fluid density and viscosity. These variables can be given constants for this problem. U_∞ and L are characteristic flow speed and length and, for our problem, they are the airfoil's airspeed and chord length. The Reynolds number controls the flow field. Low Re numbers mean a more diffusive flow with fewer convective effects whereas moderate and high Re numbers imply a convection-dominated flow. Typically, at Re numbers on the order of 10^3 , the flowfield becomes turbulent, and the Navier Stokes equations need to be augmented with a turbulence model.

2.2 Finite Element Formulation

As a first step in discretizing and solving Eqs. (2.1.1) and (2.1.2) numerically, we prepare a weak form of these equations by first introducing test functions (\vec{V}, q) against which Eqs. (2.1.1) and (2.1.2) are integrated, as follows:

$$\int_{\Omega} (\nabla \cdot \vec{U}) q = 0 \quad (2.2.1) \text{ and}$$

$$\int_{\Omega} \vec{U}_t \cdot \vec{V} + \int_{\Omega} (\vec{U} \cdot \nabla) \vec{U} \cdot \vec{V} + \int_{\Omega} \nabla p \cdot \vec{V} - \int_{\Omega} \frac{1}{Re} \Delta \vec{U} \cdot \vec{V} = 0, \quad (2.2.2)$$

for all admissible (\vec{V}, q) .

We then define appropriate subspaces u_h , v_h , and p_h to approximate the flow variables. For a Galerkin Finite Element method, we choose the same subspaces for the test functions. The finite element formulation for Eq. (2.2.2) can then be derived by choosing the appropriate test functions $\vec{V} = (\phi_i, 0)$ and integrating by parts to recover the x and y components of Eq. (2.2.2), as shown in the expression

$$\int_{\Omega} u_t \phi_i - \int_{\Omega} (\vec{U} \cdot \nabla u) \phi_i + \int_{\Omega} p(\phi_i)_x - \frac{1}{Re} \int_{\Omega} \nabla u \cdot \nabla \phi_i = 0 . \quad (2.2.3)$$

A similar equation is used for the y component. The integration by parts gives the Neumann outflow boundary conditions, which are written as

$$\frac{1}{Re} \frac{\partial \vec{U}}{\partial \vec{n}} = p \vec{n} \text{ on } \partial \Omega_o \times [0, T] . \quad (2.2.4)$$

These boundary conditions are imposed naturally in the code.

Substituting the approximations u_h , v_h , and p_h by using the basis functions ϕ_j for the velocity components and ψ_j (one order lower than ϕ_j) for the pressure variables, we form the semi-discrete system of equations

$$M \frac{d\mathbf{U}}{dt} + c(\mathbf{U}) + B\mathbf{p} + \frac{1}{Re} A\mathbf{U} = 0 \quad (2.2.5) \text{ and}$$

$$B^T \mathbf{U} = \mathbf{0}, \quad (2.2.6)$$

where \mathbf{U} and \mathbf{p} are vectors of nodal values for u_h , v_h , and p_h ; M , A and B are the stiffness matrices; and $c(\mathbf{U})$ is the contribution to the system from the non-linear terms.

For the time discretization, we use the implicit backward-Euler method; details for this method can be found in any standard numerical methods text [5]. The resulting non-linear system of equations is solved using Newton's method; the corresponding analytic Jacobians are listed in the attached code.

2.3 Numerical Stabilization

Difficulties arise in the numerical solution of Eqs. (2.2.5) and (2.2.6) at high Reynolds numbers. The contribution from the symmetric diagonal dominant matrix A is reduced while the non-linear convective terms $c(\mathbf{U})$ dominate, which presents a substantial numerical challenge to most linear solvers. This problem can be resolved by adequate mesh refinement at increased computational cost. To avoid these higher costs, however, we implemented the Stream Line Upward Petrov Galerkin (SUPG) numerical stabilization in our formulation.

Stabilization entails adding a discontinuous stabilization term to our original test function (ϕ) to give a 'weighted' test function (w) expressed as follows:

$$\vec{w} = \vec{\phi} + \tau(\vec{U} \cdot \nabla) \vec{\phi}. \quad (2.3.1)$$

The matrix τ depends on the velocity and the mesh element geometry [11]. The new stabilized test functions weigh nodal contributions in the flow direction more than those in other directions and ensure numerical stability without further mesh refinement. The stabilized formulation is simply equation 2.2.3 replaced by the appropriate component of \vec{w} , which gives

$$\int_{\Omega} u_t w_i - \int_{\Omega} (\vec{U} \cdot \nabla u) \phi_i + \int_{\Omega} p(\phi_i)_x - \frac{1}{Re} \int_{\Omega} \nabla u \cdot \nabla(\phi_i) + \int_{\Omega} u_t \varphi_i - \int_{\Omega} (\vec{U} \cdot \nabla u) \varphi_i - \int_{\Omega} p_x(\varphi_i) + \frac{1}{Re} \int_{\Omega} \Delta u(\varphi_i) = 0, \quad (2.3.2)$$

where

$$\varphi_i = \tau_{xx}(u\phi_x + v\phi_y). \quad (2.3.3)$$

Once again, a similar equation can be derived for the y component. The equations are now more complex and the Jacobian entries are considerably more complicated; they are shown in the attached code.

It is commonly said that adding stabilization amounts to adding artificial stability to the momentum equation; however, previous studies have shown that stabilization has no major impact on modeling the actual physics of the problem [4]. It simply uses the arbitrariness of the test functions to give a numerical method better suited for convection dominated flow problems.

2.4 Expressions for Computing Aerodynamic Coefficients

Fluid motion past an object results in aerodynamic forces on the object due to the pressure distribution and the frictional viscous forces on its surface. These forces are generally normalized to give the aerodynamic coefficients for that object in a particular flowfield. We now derive expressions for the coefficients of lift (C_d), drag (C_d) and moment (C_m) that characterize the performance of an airfoil.

The net pressure force on the airfoil (A_p) due to a given pressure distribution ($P(S)$) is

$$A_P = \int_A P(S) d\vec{A}. \quad (2.4.1)$$

We next define L_p as the component of A_p in the positive y direction and compute

$$C_{lp} = \frac{L_p}{\frac{1}{2}\rho U_\infty^2 A_0} = \frac{\int_A P dA (\vec{n} \cdot \vec{j})}{A_0} = \int_0^S p(S) \frac{dS}{c} (\vec{n} \cdot \vec{j}) = \int_0^s p(s) (\vec{n} \cdot \vec{j}) ds, \quad (2.4.2)$$

where the upper and lower case letters represent dimensional and non-dimensionalized quantities, respectively. Note that \vec{n} is the outward unit normal here. We obtain $p(s)$ from numerical simulations and estimate C_{lp} as

$$C_{lp} \approx \sum_{i=1}^n p_i \Delta s_i (\vec{n} \cdot \vec{j})_i, \quad (2.4.3)$$

where the summation runs over all boundary elements on the airfoil.

Similarly, the approximation for D_p , the component of A_p in the negative x direction, yields

$$C_{dp} \approx \sum_{i=1}^n p_i \Delta s_i (\vec{n} \cdot \vec{i})_i . \quad (2.4.4)$$

The viscous force coefficients (Cl_v, Cd_v) can be computed using the viscous stress tensor τ_{ij} for a Newtonian fluid, as expressed below:

$$\tau_{ij} = \mu(u_{i,j} + u_{j,i}) . \quad (2.4.5)$$

μ is the fluid viscosity (dependent only on temperature), and $u_{i,j}$ is the partial of the i th component of the velocity in the j direction.

The viscous force in the i th direction is given by,

$$dF_{vi} = \tau_{ij} dA^j, \quad (2.4.6)$$

which in the x -direction becomes

$$dF_{v1} = \tau_{1j} dA^j = \tau_{11} dA^1 + \tau_{12} dA^2 = 2\mu(U_x(\vec{n} \cdot \vec{i}) + \frac{U_y+V_x}{2}(\vec{n} \cdot \vec{j}))cdS. \quad (2.4.7)$$

Note that \vec{n} is the inward unit normal here.

The viscous drag coefficient (C_{dv}) is defined in a similar manner to C_{dp} , as shown:

$$C_{dv} = \frac{\int dF_{v1}}{\frac{1}{2}\rho U_\infty^2 A_0} = \frac{2}{Re} \int_0^s (u_x(\vec{n} \cdot \vec{i}) + \frac{u_y+v_x}{2}(\vec{n} \cdot \vec{j}))ds . \quad (2.4.8)$$

This is computed approximately, in a manner similar to Eqs. (2.4.3) and (2.4.4).

The expression for the viscous lift coefficient (C_{lv}) is

$$C_{lv} = \frac{2}{Re} \int_0^s (\frac{u_y+v_x}{2}(\vec{n} \cdot \vec{i}) + v_y(\vec{n} \cdot \vec{j}))ds , \quad (2.4.9)$$

which is again approximated from simulation data. Note that the parameter, Re , scales the viscous coefficients and, at low Re , we anticipate having higher C_{dv} and C_{lv} .

Finally, the drag and lift on the object are defined relative to the direction of the incident flow. In other words, lift is the net force perpendicular to the flowfield, and drag is the net force parallel to it. Thus, we can say that have,

$$\begin{bmatrix} C_d \\ C_l \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} C_{dp} + C_{dv} \\ C_{lp} + C_{lv} \end{bmatrix} , \quad (2.4.10)$$

where α is the angle of attack. Note that we can easily compute the viscous and pressure components of the coefficients separately from the above expression.

3 Progress

This section first details the work accomplished prior to running the simulations, i.e., meshing, code validation, domain size validation, and mesh convergence studies. The results section then describes the progress we have made so far in studying the icing problem numerically. Table 1 displays a schedule chart that shows the project activities and their duration.

Table 1: Schedule Chart

Task Performed	Projected Completion Date
Familiarization with previous work and CFD lab	Completed
Scan 2-D images of iced and clean airfoils	Completed
Write code to extract X&Y coordinates from images	Completed
Create mesh for the Finite Element solution using TRIANGLE	Completed
Add numerical stabilization to code for high Re numbers	Completed
Add C++ routines to enable Libmesh to read meshes	Completed
Code validation using benchmark tests	Completed
Run low Re simulations and extract data	Completed
Prepare midterm report and presentation	Completed
Modify code to enable unsteady simulations with high Re	Completed
Domain/Mesh Sensitivity and Parametric Studies	Completed
Write routines to extract aerodynamic characteristics	Completed
Higher Reynolds number simulations	Completed
Prepare final report and presentation	Completed

3.1 Preliminary Work

3.1.1 Meshing

The first step in the numerical solution of Eqs. (3.1.1) and (3.1.2) was the discretization of the physical domain Ω . To accomplish this, we had to first adequately describe the GLC-305 airfoil geometry for both the clean and iced airfoils. The GLC-305 airfoil used in this experiment had a chord length of 18.72 inches and a thickness to chord ratio of 8.7 percent. Airfoil images were then acquired and scanned into a digital computer which stored the image in a pixel color and location matrix. We then wrote a Matlab script that extracted a subset of the pixel coordinates that described the airfoil boundaries. This data was subsequently smoothed to remove noise, and an array of boundary nodes was created that became the input for the triangulation program, TRIANGLE. The relevant Matlab code is attached in the appendix.

TRIANGLE accepted as input the nodal array and bounding box. It used a Delaunay triangulation algorithm [12] that ensured good quality triangles by maximizing the minimum angle of any interior angle of any triangle. Since TRIANGLE is integrated into Libmesh, we were able obtain an XDA (Libmesh's mesh format) file of the output which directly became the mesh input for the Navier-Stokes program that was later assembled. Figure 2 shows the mesh generated near the ice horn after one level of uniform refinement done in Libmesh. We see that the mesh around the ice horn is very

fine which enabled the accurate capturing of the effects of the ice horn on the flowfield around the airfoil.

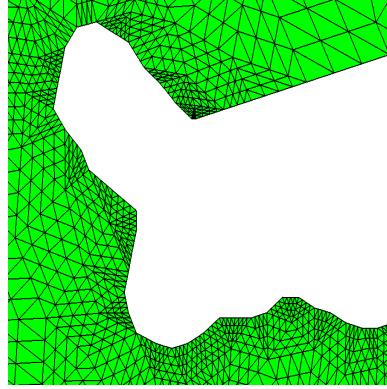


Figure 2: Unstructured mesh around ice horn

To capture the flowfield in finer detail and avoid heavy computational costs, we decided to use adaptive mesh refinement. Adaptive mesh refinement involves the computation of a solution to the PDE on a coarse mesh and using an “error indicator” to estimate the error on each element. We then flagged a subset of the elements where the error is above a certain threshold, and those elements were then refined to improve the accuracy of the solution. This ensured that the mesh was enriched only in areas where great mesh detail was needed to capture the physics. For our code, we used the flux jump error indicator [7], which allowed us to capture the viscous boundary layer and flow separation regions where there were large fluxes in the velocity. Figure 3 shows an adapted mesh after 7 adaptive steps.

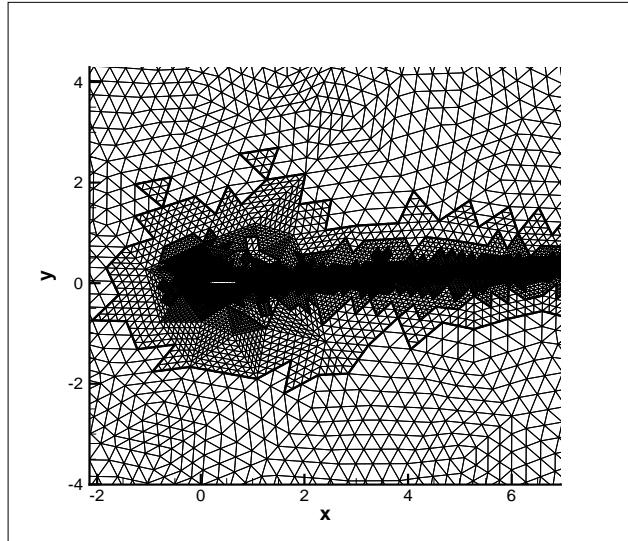


Figure 3: Mesh after adaptive mesh refinement

3.1.2 Code Validation

The entries for the stiffness matrices to be specified in the Libmesh program were derived in the theory section. Both the residuals (Eqs. (2.3.5) and (2.3.6)) and the corresponding Jacobians were coded in the program. The boundary conditions in Eqs. (2.1.3) and (2.1.4) were enforced using a penalty method whereas Eq. (2.2.4) was enforced naturally in the code. The analytic Jacobians were then verified by computing their finite-difference equivalents.

Jacobian contributions from the stabilization terms were computed analytically, and C++ routines were then written to implement the stabilization. Finally, the code was validated by solving a backwards facing step flow: computing the reattachment length of the separated flow at the base of the step and comparing with existing experimental and numerical data [1]. Figure 3 depicts the normalized reattachment length (x_{re}/S) where S is the step height versus Re_D . The plot illustrates the effect of stabilization; without it, we obtain an incorrect x_{re} at mesh spacing 0.125, whereas the stabilized code computes the correct x_{re} at the same mesh resolution.

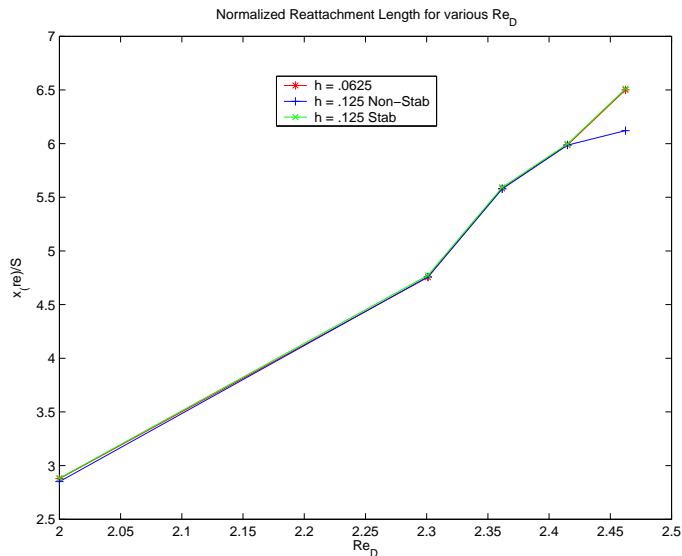
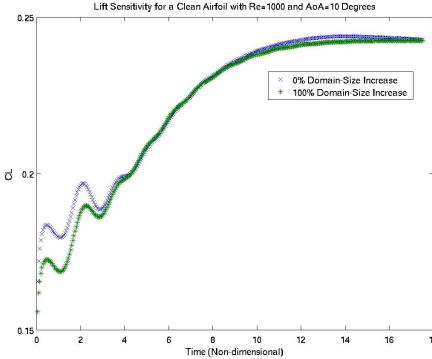


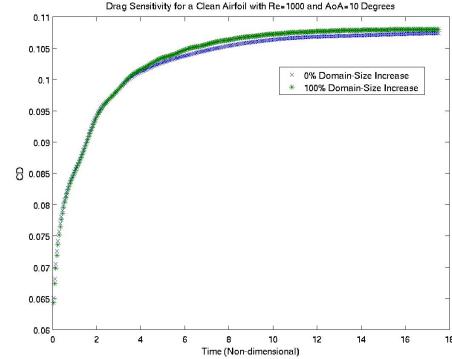
Figure 4: Backwards Facing Step Results

3.1.3 Validation of Domain Size

The natural boundary conditions in Eq. (2.2.4) necessitate the use of a sufficiently large domain size so as to allow any vortices shed from the airfoil to diffuse naturally before encountering the boundary. Therefore, we started with a mesh that was ten chord lengths long downstream of the airfoil and then enlarged it incrementally to check that the flowfield did not change with the domain size. Figure 5 shows the time-dependent lift and drag coefficients for the original domain size and a domain that was twice as large. It is clear that the original domain was adequate for the simulations.



(a) C_l v/s t

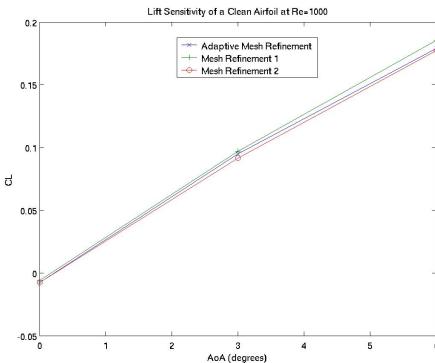


(b) C_d v/s t

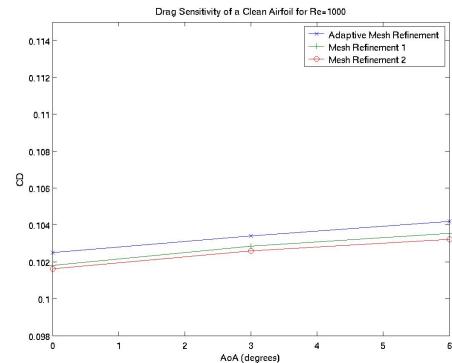
Figure 5: Domain Sensitivity Studies at $Re = 1000$, $\alpha = 10$, clean airfoil

3.1.4 Grid Sensitivity Studies

To ensure that our mesh resolution is adequate for capturing the flow field, we performed a grid sensitivity study. Simulations with 2 and 3 levels of uniform refinement were carried out to obtain a very fine grid solution. We computed the aerodynamic coefficients from these simulations and compared them to our adaptive solution to ensure that the results match up. Figure 6 shows the resultant C_l and C_d plots. Note that the adaptive grid had a finer grid in some regions than the uniformly refined grid, however the uniform grids had more global degrees of freedom.



(a) C_l v/s α



(b) C_d v/s α

Figure 6: Mesh Convergence Studies at $Re = 1000$, various α , clean airfoil

3.2 Results

3.2.1 Steady Flow Results

Figure 7 show the C_d and C_l for the clean and iced airfoil as a function of Reynolds number at an angle of attack of 3° . We see that at low Reynolds numbers the iced airfoil outperforms the clean airfoil. However, as Re increases the C_l for the clean airfoil increases substantially, due to the increasing effect of the pressure forces.

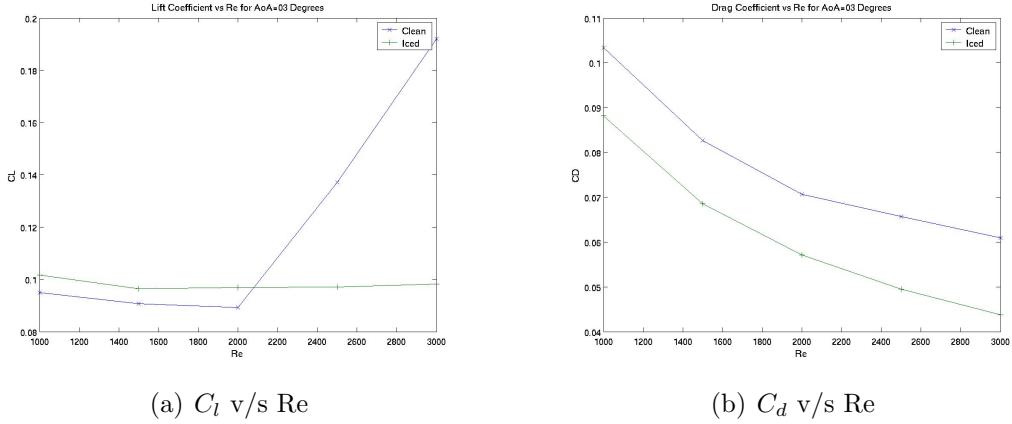


Figure 7: Airfoil performance dependence on Re number, $\alpha = 3^\circ$

Figure 8 shows the flow separation region aft of the ice horn at various Re numbers at an angle of attack of 3° . We see that the length of the separation region increases as we increase the Re number. It must be noted that the separation region decreases viscous drag. The flow moves in a direction opposite to the bulk fluid motion near the surface, thus the x-velocity gradients in Eq. (2.4.8) are negative over the separation region. However, separated flow also implies higher pressures and more pressure drag. Thus the lift of the airfoil decreases as the size of the separation region increases, as seen in Figure 7.

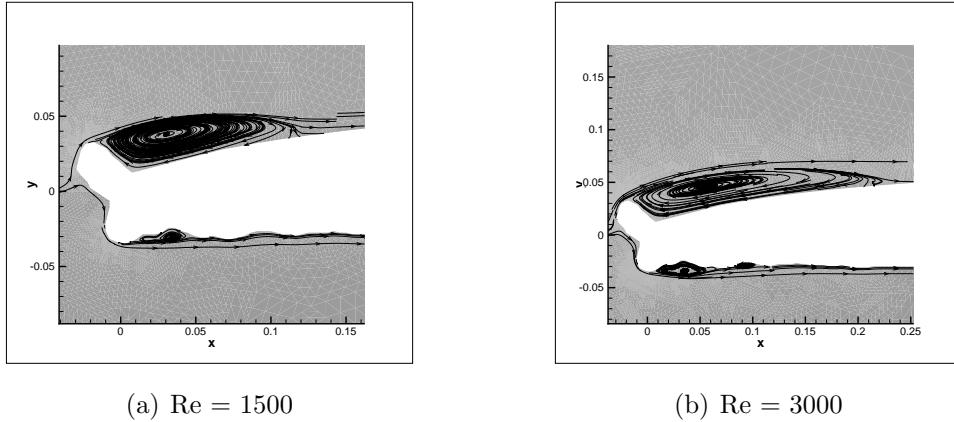


Figure 8: Recirculation region behind the ice horn, $\alpha = 3^\circ$

Figure 9 shows the increase in the length of the separated region behind the ice horn as we increase α . As before, a larger separation region implies less viscous drag but more pressure drag.

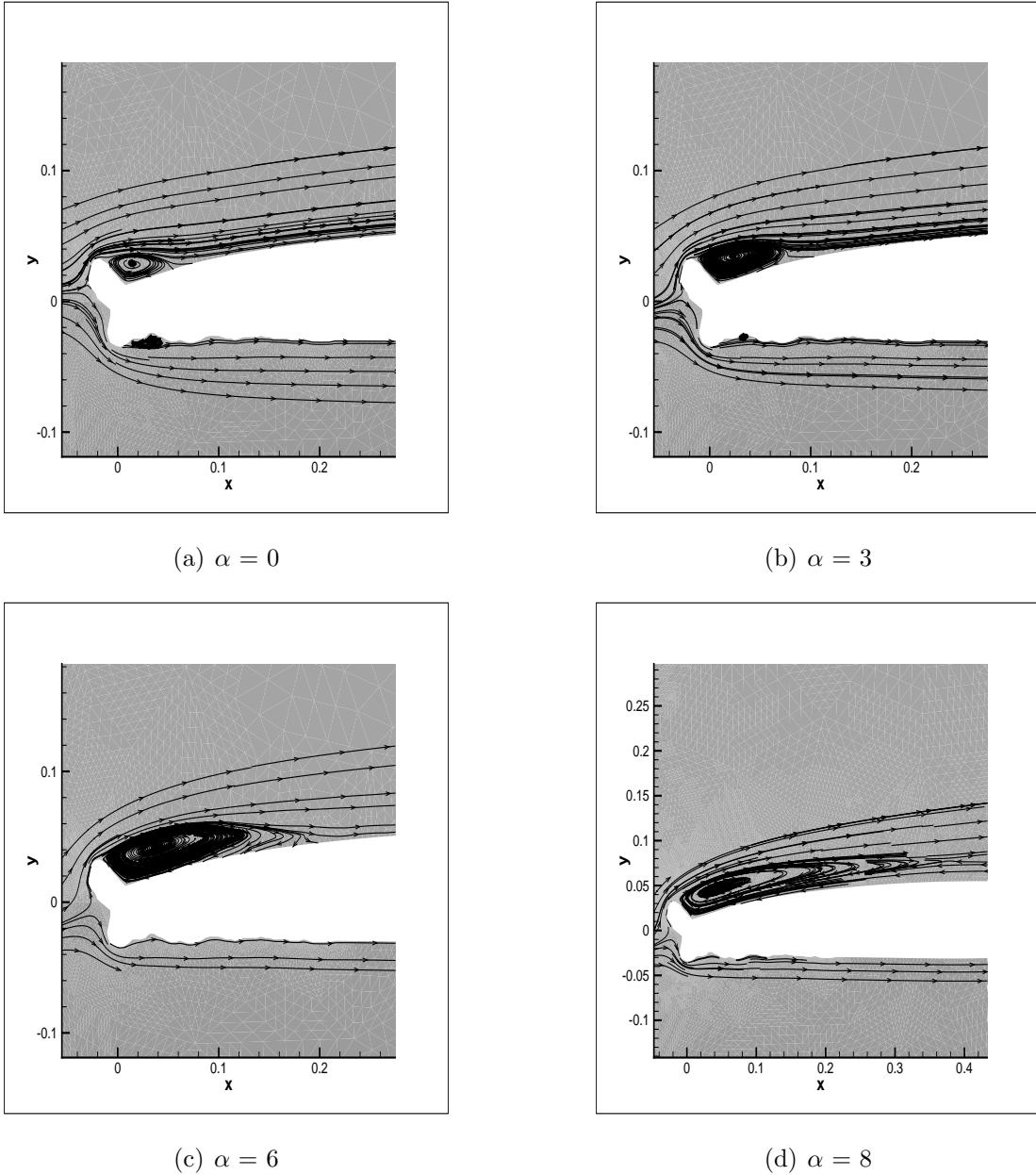


Figure 9: Separation region on the top of the airfoil at $Re = 1000$

Figure 10 shows C_d and C_l for the clean and iced airfoils at various angles of attack at the fixed Re number of 1000. The coefficients for the 10° angle of attack were mean values recovered from the observed non-stationary steady-state. The graph indicates that the flowfield possibly has a non-stationary steady state at $\alpha = 8^\circ$ just like the 10° case. However, a full unsteady simulation for the 8° case was not performed due to time constraints. We see that the clean airfoil performs better at high angles of attack,

but is outperformed by the iced airfoil at low angles of attack.

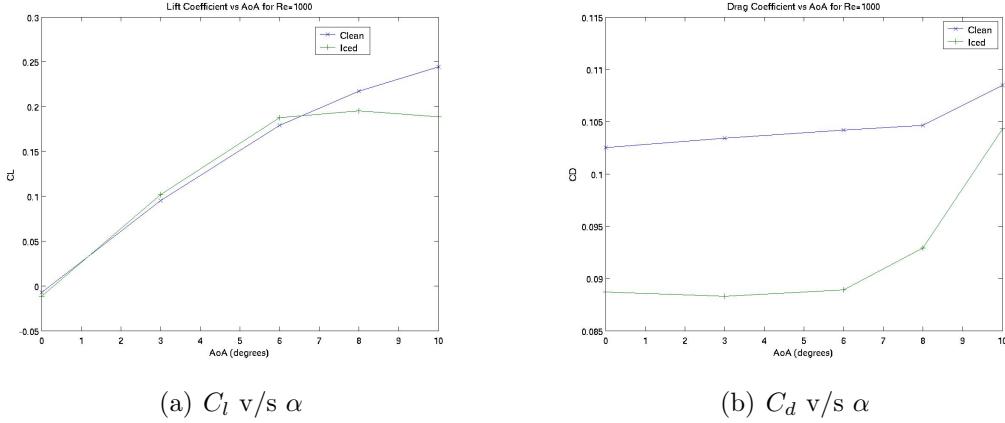


Figure 10: Airfoil performance dependence on α , Re number = 1000

3.2.2 Unsteady Flow Results

Unsteady simulations were only performed at Re of 1000. This corresponds to a velocity of 0.027 m/s at a fluid temperature of 273 K for the GLC-305 airfoil. The plots for time dependent lift and drag coefficients for low angles of attack indicated that the flow field reached a steady state. However at an angle of attack of 10, the plots indicated a non-stationary steady state as can be seen from Figure 11. This unsteadiness may be attributed to the fully separated flow seen on the top of the airfoil.

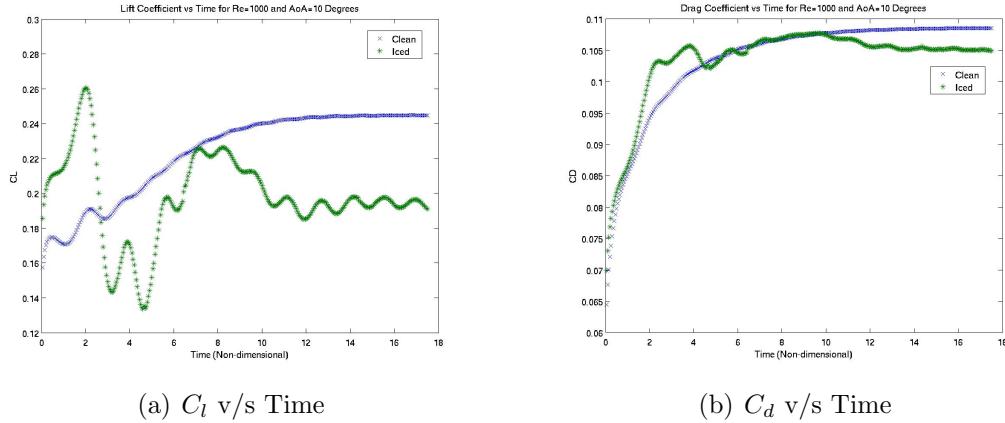
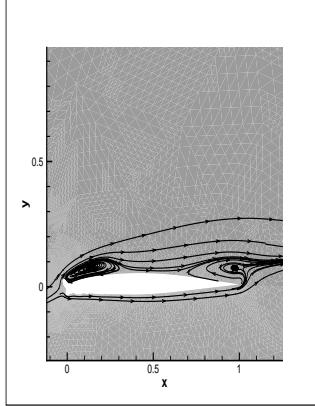
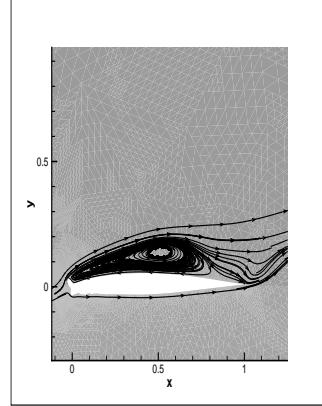


Figure 11: Airfoil performance dependence on time, Re number = 1000, $\alpha = 10$

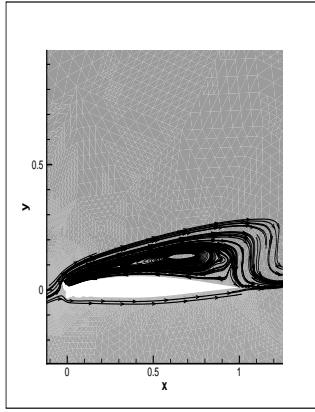
Figure 12 shows the growth of the separation region behind the ice horn. We see that this separation region grows with time and coalesces with the second separation bubble towards the trailing edge of the airfoil. We thus have a large separation region on top of the airfoil. Movies of the unsteady flow show that this separation region is not static, but exhibits periodic vortex shedding with time.



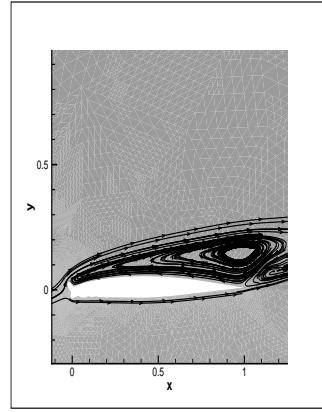
(a) Timestep 13 ($t = 0.65$)



(b) Timestep 31 ($t = 1.55$)



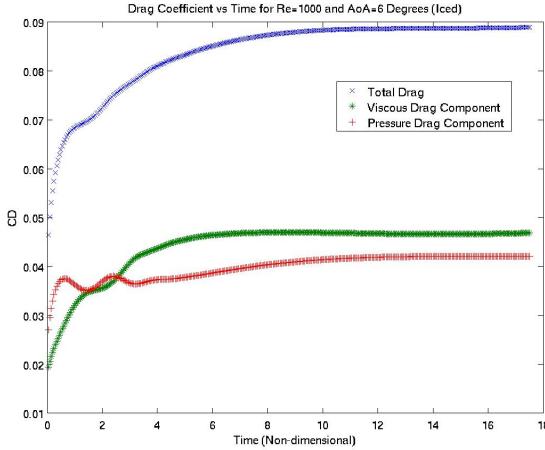
(c) Timestep 110 ($t = 5.5$)



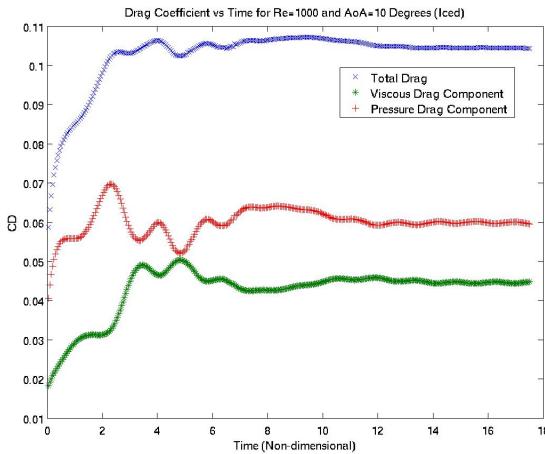
(d) Timestep 164 ($t = 8.2$)

Figure 12: Separation region on the top of the airfoil at $\text{Re} = 1000$, $\alpha = 10^\circ$

The fully separated flow leads to an increase in the pressure drag experienced by the iced airfoil. This can be seen from Figure 13 which shows a breakdown of the viscous and pressure components of the drag at angles of attack 6 and 10. We see that at $\alpha = 10^\circ$ the pressure drag is higher than the viscous drag which contrasts with the situation at $\alpha = 6$ where the viscous drag is higher. Thus the higher angle of attack leads to complete flow separation and more pressure drag, thus increasing the net C_d . It also results in a lower C_l . The periodic vortex shedding and changes in the size of the separation region lead to a time dependent lift and drag as shown in Figure 11.



(a) C_d v/s Time for $\alpha = 6$



(b) C_d v/s Time for $\alpha = 10$

Figure 13: Viscous and Pressure drag components at $Re = 1000$

3.3 Cost Analysis

The major cost factor involved in this project was the computational time. Greater mesh resolution amounted to better accuracy at the expense of higher computational costs. Also the natural farfield boundary conditions necessitated the use of large domains, resulting in higher costs.

The costs were minimized by using adaptivity to ensure high mesh resolution only in areas where it was absolutely essential, incremental domain size validation which ensured we used the minimum required domain size and parallel processing which reduced simulation time.

4 Conclusions

We have succeeded in developing and validating a stabilized adaptive finite element method based CFD code using the Diffsystem framework in the Libmesh Finite Element Library. We have conducted parametric studies at low and moderate Reynolds numbers to compare the performance of the clean and iced airfoils. We have also prepared plots and movies of our results to present our results.

It was found that the iced airfoil outperforms the clean airfoil at low angles of attack and Reynolds numbers. This seemingly non-intuitive result was explained by recovering the viscous and pressure components of the drag. It was found that at low angles of attack the viscous forces dominated the flowfield. However, at higher angles of attack the flow on the top of iced airfoil was fully separated, which resulted in higher pressure drag. Also the size of the separation region aft of the ice horn increased with Re number, again leading to higher pressures and a loss of lift for the iced airfoil.

We also researched the literature to find prior numerical and experimental work on flow over an airfoil at the Re numbers we used. We found that these flows have only recently been studied [8] due to their relevance for the design of micro-aircraft. Sun and Boyd [13] discuss how even flat plate's can outperform airfoils at low Reynolds numbers. Thus our results are perhaps not as unexpected as we initially thought. We have shown though that low Reynolds numbers flow are viscous dominated, especially at low angles of attack and viscous forces are important for the design of micro-aircraft.

5 Future Work

The ultimate objective of this project is to provide a framework for a comprehensive, robust and reliable fluid-structure code to augment the studies of icing in the Aerospace Engineering department. To achieve this end further goals have to be reached:

5.1 Turbulence Modeling

A turbulence model must be added to the existing code. This will enable simulations at Reynolds numbers on the order of 10^6 and bring us closer to simulating flows at Re numbers that these airfoils operate at. We have taken the first steps towards this, the framework for a two-equation turbulence model is already present in the code. A future team needs to first choose an appropriate turbulence model and then prepare a finite element formulation for it. The corresponding stiffness matrix entries can then be added directly to the code and changing a simple boolean variable in the code will enable turbulence modeling.

5.2 Fluid-Structure Interaction code

To simulate aeroelasticity and flutter one needs a structural dynamics code alongwith the time dependent lift and drag coefficients. We have demonstrated that time dependent coefficients can be accurately recovered from our fluids code. A future team can couple a structural dynamics code to the fluid mechanics code. A simple linear and torsional spring system may be used to model the motion of the airfoil as described in [10]. To avoid the complications of using a moving mesh, the Navier Stokes equations in a non-inertial frame of reference may be used,

$$\begin{aligned} \nabla \cdot \vec{u} &= 0 \\ \rho(\vec{u}_t + (\vec{u} \cdot \nabla)\vec{u}) &= -\nabla p - \rho(2\vec{\Omega} \times \vec{u} + \vec{\Omega} \times \vec{\omega} \times \vec{r}) + \mu \Delta \vec{u} \end{aligned} \quad (5.2.1)$$

where $\vec{\Omega}$ is the angular velocity of a co-ordinate frame fixed to the airfoil and \vec{r} is the radial vector from the origin of this frame to a given fluid element. The finite element formulation for this equation will only need to be done for the angular acceleration term since the rest of the equation is identical to equation 2.1.2 after non-dimensionalization.

The addition of a turbulence model and coupling with a structural dynamics code should give researchers in our department a valuable and effective tool for the study of icing and aeroelastic problems in general.

References

- [1] G. Biswas, M. Breuer, and D. Durst. "Backward-Facing Step flows for various expansion ratios at low and moderate Reynolds numbers". *J. Fluids Eng.*, 126:362–374, 2004.
- [2] M. Bragg, A. Broeren, and H. Addy L. Blumenthal. "Iced-Airfoil Aerodynamics". Report, 2004.
- [3] A. Broeren, H. Addy, and M. Bragg. "Flowfield measurements about an airfoil with leading-edge ice shapes". 2004.
- [4] A. Brooks and T.J.R. Hughes. "Stream Line Upward Petrov Galerkin formulations for Convection Dominated Flows with Particular Emphasis on The Incompressible Navier-Stokes Equations". *Computer Methods in Applied Mechanics and Engineering*, 32:199–259, 1982.
- [5] G.F. Carey and J.T. Oden. *Finite Elements: Fluid Mechanics*. Prentice-Hall, 1986.
- [6] X. Chi, B. Williams, and T.I-P Shih et al. "2-D and 3-D CFD Simulations Of Clean and Iced Wings". 2006.
- [7] D.W. Kelly et. al. "A Posteriori Error Analysis and Adaptive Processes in the Finite Element Method: Part 1 - Error Analysis". *International Journal For Numerical Methods in Engineering*, 19:1593–1619, 1983.
- [8] D. Mateescu and M. Abdo. "Aerodynamic Analysis of Airfoils at very low Reynolds Numbers". 2004.
- [9] Donald Norman. *The Design Of Everyday Things*. Basic Books Inc.: New York, 1988.
- [10] Augustin Petre and Holt Ashley. "Drag Effects on Wing Flutter". *Journal Of Aircraft*, 13:755–763, 1976.
- [11] F. Shakib, T.J.R. Hughes, and Z. Johan. "A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier-Stokes equations". *Computer Methods in Applied Mechanics and Engineering*, 89:141–219, 1991.
- [12] J. Shewchuck. "Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator". *Applied Computational Geometry: Towards Geometric Engineering*, 1148 of Lecture Notes in Computer Science:203–222, 1996.
- [13] Q. Sun and I. Boyd. "Flat-plate aerodynamics at very low Reynolds number". *J. Fluid. Mech.*, 502:199–206, 2004.

6 Appendix

Attached is the C++ and Matlab code written to complete this project.