Prediction of the Lift and Drag Coefficients of a Moving Airfoil Using Computational Fluid Dynamics Simulation

by

Fang Gao

Submitted in partial fulfilment of the requirements
for the degree of Master of Applied Science

at

Dalhousie University
Halifax, Nova Scotia
April 2014

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The purpose of this research is to numerically simulate, analyze, and visualize turbulent flow around rotating aerodynamic shaped 3-dimentional geometries using a custom-made software suite. The computational fluid dynamic program used for this research is called Numerical Wind Tunnel, NWT, which was developed by Dr. J. Militzer and his students over the last 15 years. In order to meet various simulation and prediction requirements of this research, the NWT was modified and improved by implementing many new features; in addition, many bugs have been fixed. Key features added to the NWT include improved boundary layer handling for Detached Eddy Simulation method, new implementations of Surrounding Cell Method and rewritten Lift and Drag Coefficients calculation algorithms, and new approaches to Mesh Refinement and Adaptation Criteria. The improved software is tested extensively by simulating turbulent flows around a rotating National Advisory Committee for Aeronautics (NACA) 0009 airfoil, and test results are compared with both experimental data and previous simulation data. The research was successful mainly because of the much-improved accuracy in predicting static lift and drag coefficients. Another achievement of this research is that the software also successfully predicted various events during an airfoil dynamic stall condition, which is a result of both accurate flow prediction and a NWT feature called Automatic Anisotropic Grid Adaptation.

LIST OF ABBREVIATIONS AND SYMBOLS USED

2D	Two-dimensional
3D	Three-dimensional
AOA	Angle of Attack
B-L	Baldwin-Lomax
CFD	Computational Fluid Dynamics
C_d	Drag coefficient
C_l	Lift coefficient
C_p	Pressure coefficient
C_{SGS}	SGS constant
δ_{ij}	Kronecker delta
Δ	Filter cutoff width of LES
Δt	Computational time step
DES	Detached Eddy Simulation
DNS	Direct Numerical Simulation
ϕ	Sum of a steady mean component Φ
ϕ'	Time varying fluctuating component of Φ
Φ	A steady mean flow property component
F_d	Drag force
F_l	Lift force
flop	FLoating-point Operations Per Second
IB	Immersed Boundary
k	Wavenumber
LES	Large Eddy Simulation
μ	Dynamic viscosity
μ_{SGS}	Dynamic SGS viscosity
Mflops	10^6 flops
NACA	National Advisory Committee for Aeronautics
NWT	Numerical Wind Tunnel

ρ	Density
RANS	Reynolds-averaged Navier-Stokes
Re	Ryenolds number
R_{ij}	LES Reynolds stresses
SGS	Sub-grid-scale stresses
Tflops	10^{12} flops
$\tau_{xx}, \tau_{yy}, \tau_{zz}$	Normal stress
$\tau_{xy}, \tau_{yz}, \tau_{xz}$	Shear stress
u', v', w'	Velocity fluctuation
$u+$	Ratio of local fluid velocity and friction velocity
x	Direction parallel to free stream
y	Direction parallel to airfoil span
$y+$	Dimensionless wall distance
z	Direction perpendicular to free stream and airfoil span

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr. Julio Militzer, for everything he taught me, and all the help and guidance he gave me. With his expertise in fluid mechanics and computational fluid dynamics, he pointed directions and guided me through every stage of my research. I also want to thank Cassio Faria and Dr. Blingen for providing experimental data and resources that are essential to this research. I would like to thank Dr. Rubens Campregher for providing assistances about the NWT when I needed them. I would like to further thank all my peers and friends for all of their support, especially Andrew Clarke, Alok Dhungana, Franco Barbi, Rob Doyle, and countless others. This work was made possible by the support of The Atlantic Computational Excellence Network (ACEnet) and Dalhousie University, for which both of these institutions have my deepest gratitude. Finally, I want to thank my parents and my wife, their constant support has made this possible.

CHAPTER 1 INTRODUCTION

Over the past decade, rapidly increasing computational power, thanks to parallel computing and inexpensive computers, has made computational fluid dynamics (CFD) a much more accessible tool for solving flow around complex geometries. Hence, many engineers have shifted their experimental research methods to CFD or a combination of experimental and computer simulation methods. One of the most studied fluid problems is the turbulent flow around aerodynamic shaped geometries such as an airfoil. A successful and useful airfoil simulation should be able to produce accurate lift, drag, and pressure coefficients of the immersed airfoil, and these data should also help engineers predict airfoil foil stall condition. Predicting these data is the goal of this research.

The CFD program used for this research is called Numerical Wind Tunnel (NWT), which is a complex CFD software package developed by Dr. J Militzer and his students. The first version of the NWT was developed 15 years ago and gradually improved by many different people over the years. The NWT was customized into many different versions to simulate many kinds of flows or flows around blunt objects, but it was never used to simulate flow around an airfoil, not to mention a moving airfoil. A NACA 0009 was chosen as the subject of the simulation. When it was first loaded into the NWT for testing, the NWT failed to produce acceptable results, but provided direction for modifications and improvements. After careful examination and analysis of shortcoming of the results, the NWT was modified and many new features were added. The following list is a short summary of key features added for this research:

1. The NWT utilizes Detached Eddy Simulation (DES) method to simulate turbulent flow, which is a hybrid method of both Large Eddy Simulation (LES) and Baldwin-Lomax (B-L) turbulent model. The DES implementation was modified to ensure a better and smoother transition between LES and B-L layers.

2. Surrounding Cell Method is a key feature of the NWT allowing it to track Cartesian grid locations surrounding the immersed boundary (IB). The feature was implemented to only work with cylinder geometries in the NWT, so a new version was implemented.
3. In order to accurately simulate flow around an airfoil, the NWT must be able to calculate lift and drag coefficients correctly, so a new algorithm capable of calculating lift and drag coefficients of a moving airfoil was created.
4. Mesh refinement and adaptation criteria methods were also implemented to control the mesh refinement near the IB, without significantly increasing total mesh size.

The thesis consists a total of five chapters. Chapter 2 current and past researches in the area of simulating turbulent flows around airfoils, and in particular moving airfoils. Chapter 3 discusses features implemented in the NWT, and focuses on new improvements and features mentioned in the previous list. Chapter 4 gives detailed comparisons between experimental and the NWT simulated airfoil data, it also highlights prediction differences before and after adding these new features and improvements.

CHAPTER 2 LITERATURE REVIEW

The focus of this chapter is to review the literature on turbulence modeling and the simulation and measurement of flows around airfoils. Section 2.1 presents the review of a wide range of turbulence modeling methods and their respective pros and cons. This guided the choice of the turbulence model used in the NWT implementation. Section 2.2 presents a review of a wide range of factors that affect airfoil lift and drag coefficients. In addition, consideration is given to static and dynamic stall predictions methods.

2.1 TURBULENCE MODELING

Turbulence can be characterized by considering the superposition of vortices of a wide range of sizes over the average flow. As a consequence there is an intense mixing and momentum transfer between the fluid particles that compose the flow. Historically, turbulent flow prediction relied heavily on experimental methods, but thanks to the rapid development of modern computing, Computational Fluid Dynamics (CFD) has become an important tool for turbulence study. Over the past several decades, many turbulence models were developed to improve CFD predictions and handle complex turbulent flows. In this section we discuss some of the more successful turbulence models.

2.1.1 Reynolds-averaged Navier-Stokes Models (RANS)

The following four types of turbulence models are based on Reynolds-averaged Navier-Stokes equations.

1. Algebraic (Zero-Equation) Models. This is based on Prandtl's mixing length hypothesis, was developed by Van Driest and improved by many others afterwards [3]. Baldwin and Lomax model is a widely used algebraic model that is suitable for high-speed flow simulation and it is known for giving very accurate results near wall [3]. This model is essential in this research, and it will be discussed in detail in the following chapters.

2. One-Equation Models. Just like its name suggested, this type of models solves the turbulent transport equation, namely the kinetic energy of the turbulent fluctuation k. However, it is inherently limited and has had very little success [3].
3. Two-Equation Models. Kolmogorov developed first of this kind in 1942 and called it the *k-ω* model. It solves a transport equation for the kinetic energy of turbulence plus another one for the transport of a parameter *ω,* known as "the rate of dissipation of energy per unit volume and time". Launder and Spalding's *k-ε* model, where *ε* is the rate of dissipation, was introduced later and became the most widely used turbulence model, and it's still being used in today's commercial CFD programs [3][32].
4. Second-Order Closure Models. These models calculate individual Reynolds stresses with differential equations that govern the evolution of the Reynolds stress tensor. Because these models require solving many equations, they are not widely used [3].

All four types of models above are referred to as RANS turbulence models. In these models, a flow property ϕ is defined as the sum of a steady mean flow property component Φ and a time varying fluctuating component ϕ'. Hence when the equation $\phi = \Phi + \phi'$ is substituted into the Navior-Stokes equations, the time varying fluctuating component will constitute an extra term in the time-averaged Navior-Stokes equation:

$$-\frac{\partial\left(\rho\overline{u'\phi'}\right)}{\partial x}-\frac{\partial\left(\rho\overline{v'\phi'}\right)}{\partial y}-\frac{\partial\left(\rho\overline{w'\phi'}\right)}{\partial z} \tag{2.1}$$

If the Navior-Stokes equation is decomposed into the momentum equations for x, y and z Cartesian directions, six Reynolds stresses result, namely three normal stresses

$$\tau_{xx} = -\rho\overline{u'^2} \qquad \tau_{yy} = -\rho\overline{v'^2} \qquad \tau_{zz} = -\rho\overline{w'^2} \tag{2.2}$$

and three shear stresses

$$\tau_{xy} = -\rho\overline{u'v'} \qquad \tau_{yz} = -\rho\overline{v'w'} \qquad \tau_{xz} = -\rho\overline{u'w'} \tag{2.3}$$

All these must be solved for the closure of the equations 2.1. Concluding the consideration of the classic RANS models we observe that they use different ways to calculate Reynolds stresses, and some are better than others, but they all have some

drawbacks. First of all, the RANS equations are obtained from time-average processes, so they do a better job simulating steady flows. Second, RANS models are highly case dependent, which means the models must be optimized and validated for specific types of turbulent flows, and cannot be used as a general-purpose turbulence model. And finally, turbulent flows consist of eddies of many different sizes, larger eddies behave very differently from smaller eddies. Since RANS models do not distinguish the size differences of eddies, RANS models try to describe eddies of all sizes with one single model, so the flow is not accurately simulated. A new approach is required to solve these issues.

2.1.2 Direct Numerical Simulation (DNS)

Ideally in a CFD calculation the grid should be as fine as possible. However, to consider the smaller size of eddies the grid must be so fine that it the solution is not practical, even with modern computer clusters. The reason to use a turbulence model to simulate a flow is to avoid the need for such fine grids. DNS computes the mean flow and all turbulent velocity fluctuations on a sufficiently fine spatial mesh with sufficiently small time steps, so that all turbulent eddies and all fluctuations are resolved. With this kind of accuracy, DNS has the potential to become the ultimate general-purpose simulation tool. However, as stated above, DNS requires too much computational power to be a practical method, at least at this time [1].

The driving force of the chaotic turbulent flow mixing are rotational flow structures called turbulent eddies. Eddies are always three-dimensional and come in a wide range of length scales. In order to capture all details in the flow and to describe processes at all length scales, the number of grid points needed is $Re^{9/4}$. Moin and Kim [33] estimated computing times of 100 hours to 300 years for turbulent flows at Reynolds numbers in the range of 10^4 to 10^6 based on high-performance computer speeds of 150 Mflops available at that time. Versteeg estimated in 2006 the supercomputer reached processor speed of the order 1-10 Tflops [2], which means it would take a 2006 supercomputer half

a month to simulate a flow of Re = 10^6. This kind of computation requirements limited the DNS computations to low Reynolds number flows, and it's still out of reach for most researchers.

2.1.3 Large Eddy Simulation (LES)

Fortunately, there is another way. Because turbulent eddies have a wide range of length, time and velocity scales, and the ratio between large and small length scales is proportional to $\mathrm{Re}^{-3/4}$ [3], their behaviour and characteristic are very different. Large eddies' shapes and sizes should depend more on the boundary conditions, driven by the mean flow energy, and behave highly anisotropically; whereas smaller eddies are affected more by rate of dissipation of turbulent energy and kinematic viscosity of the flow, and they are approximately isotropic. Instead of treating all eddies the same way, LES applies a spatial filter to the turbulent flow, and separates larger eddies from the smaller ones, and solves them separately.

In the LES method, a spatial filter function and a predefined cutoff width Δ are selected before the simulation. Essentially, the LES uses a filter function to divide eddies into two groups: all eddies larger than the cutoff width are solved with a time-dependent simulation; all smaller eddies are filtered out, and their interactions and effects on the flow are then modeled in the form of a new stress called sub-grid-scale stresses (SGS). The filter function is defined as follows:

$$\bar{\phi}(x,t) \equiv \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} G(x,x',\Delta)\phi(x',t)\,dx_1'\,dx_2'\,dx_3' \tag{2.4}$$

where $\bar{\phi}(x,t) =$ filtered function

$\phi(x,t) =$ original (unfiltered) function

$\Delta =$ filter cutoff width

and $G(x,x',\Delta)$ is a filter function

Note that in this equation, the overbar in $\bar{\phi}$ indicates spatial filtered instead of time-averaged variable. There are three different types of filters, but only the Top-hat filter is

suitable for finite volume implementations of LES, which is defined as follows:

$$\begin{aligned} G(x,x',\Delta) &= 1/\Delta^3 \quad when\ |x-x'| \le \Delta/2 \\ G(x,x',\Delta) &= 0 \qquad when\ |x-x'| > \Delta/2 \end{aligned} \tag{2.5}$$

After substituting the filtered flow variables into the Navier-Stokes equations, the filtered LES momentum equation is as follows:

$$\frac{\partial(\rho\overline{u}_i)}{\partial t} + div(\rho\overline{u}_j\overline{u}_i) = -\frac{\partial\overline{p}}{\partial x_i} + \mu div(grad(\overline{u}_i)) - div(\overline{\rho u_i u_j} - \rho\overline{u}_i\overline{u}_j) \tag{2.6}$$

$$\text{where } div(\overline{\rho u_i u_j} - \rho\overline{u}_i\overline{u}_j) = \frac{\partial\tau_{ij}}{\partial x_j}$$

and $\tau_{ij} = \overline{\rho u_i u_j} - \rho\overline{u}_i\overline{u}_j$ is the SGS stresses mentioned above. The SGS stresses consist of three types of stresses: Leonard stresses, cross-stresses, and LES Reynolds stresses, and among them, the most significant stresses are the LES Reynolds stresses, denoted as R_{ij}, which are due to interaction of unresolved SGS eddies, hence they must be modeled by SGS models.

Smagorinsky-Lilly's SGS model is one of the most popular and most widely used SGS models [3]. Smagorinsky proposed this model in 1963 base on the theory that R_{ij} are proportional to the local rate of strain of the resolved flow $\overline{S}_{ij}$, but later on, in 2000, Meinke and Krause [34] suggested all SGS stresses can be modeled as a single entity with a single SGS model, so the SGS stresses are written in the following form:

$$\tau_{ij} = -2\mu_{SGS}\overline{S}_{ij} + \frac{1}{3}\tau_{ij}\delta_{ij} \tag{2.7}$$

$$\text{where } \overline{S}_{ij} = \frac{1}{2}\left(\frac{\partial\overline{u}_i}{\partial x_j} + \frac{\partial\overline{u}_j}{\partial x_i}\right)$$

and δ_{ij} is called Kronecker delta, it's defined as $\delta_{ij} = 1$ *if* $i = j$ *and* $\delta_{ij} = 0$ *if* $i \neq j$. The Smagorinsky-Lilly SGS model essentially assumes that the dynamic SGS viscosity μ_{SGS} can be described in terms of one length scale Δ and one velocity scale $\sqrt{2\overline{S}_{ij}\overline{S}_{ij}}$. Thus,

the SGS viscosity is defined as:

$$\mu_{SGS} = \rho\left(C_{SGS}\Delta\right)^2 \sqrt{2\bar{S}_{ij}\bar{S}_{ij}} \tag{2.8}$$

where C_{SGS} is the SGS constant. Lilly originally proposed the constant to be between 0.17 and 0.21using analysis of the decay rates of isotropic turbulent eddies in the inertial subrange of the energy spectrum, but many scientists believed $C_{SGS} = 0.1$ is the most appropriate choice for internal flows [4]. The different choices for C_{SGS} suggested that the Smagorinsky-Lilly model requires case-by-case adjustment for the LES method. Another drawback of the Smagorinsky-Lilly model was that the direction of energy flow is exclusively from large resolved scale towards the sub-grid-scale, but Leslie and Quarini [9] showed that 30% of energy transferred in this direction is offset by energy flow from small to large length scale called backscatter energy. As a result, better and more universal approaches were developed.

Several models were developed to address the fixed SGS constant, including a mixed model proposed by Bardina *et al.* [10] and a dynamic SGS model proposed by Germano [11]. Both models were able to improve the results of the classic Smagorinsky SGS model. However, for this thesis, a different method called Multiscale model was used.

Multiscale models were introduced by Hughes *et al.* [7] [8] to tackle the LES problems from a different approach. As mentioned before, one major issue with the SGS models is that the omission of backscatter energy, which is caused by the definition of SGS viscosity assumption. Recall that the real SGS stresses τ_{ij} contain three types of stresses: Leonard stresses, cross-stresses, and LES Reynolds stresses R_{ij}, but in the Smagorinsky-Lilly SGS models, it is assumed that $R_{ij} = \tau_{ij}$, so when R_{ij} are modeled, effects of the other stresses and interactions are lost.

Hughes *et al.* proposed to spatially separate the flow of all eddies into three different scale ranges: large scales, small scales, and unresolved scales. Like the classic SGS models, both large and small scales are solved with time-dependent flow simulations, and

the unresolved scales are modeled. But, unlike the classic SGS models, large scale and small scale are both derived from the Navier-Stokes equations but separated with a spatial operator. This operator separates large and small scales at eddy wavenumber $\bar{k}$ and the cutoff wavenumber is k'. Hughes *et al.* proposed that if the small-scale wavenumber were defined properly, the unresolved scale would interact primarily with the small scale. Based on this theory, in their Multiscale models, the modeled unresolved scale is included into the small-scale equation, so that effects of all three previously mentioned stresses are accounted for [7]. The modified equation is written as:

$$\frac{\partial(\rho u_i)}{\partial t} + div(\rho u_j u_i) = -\frac{\partial p}{\partial x_i} + \mu div(grad(u_i)) - div(2\mu_t' S_{ij}') \tag{2.9}$$

where $\mu'_t = \rho(C_S'\Delta')^2 |S'|$

and $C_S'\Delta' = \left(\frac{2}{3\alpha}\right)^{3/4} \bar{k}^{-1} \left(\left(k'/\bar{k}\right)^{4/3} - 1\right)^{-3/4}$

where α is the Kolmogorov constant

In recent simulation works, the LES method showed promising results in many flows, even when solving highly complex flows, such as the gas turbine simulation conducted by Moin [12].

2.2 AIRFOIL RESEARCHES

Flow around an airfoil is a very complex physical problem, as this type of flow involves transition and pressure-induced separation depending on Reynolds number, angle of attack, shape of airfoil, laminar separation, transition, turbulent reattachment and trailing edge separation, which are all considered highly unsteady and complex phenomena. Cebeci *et al.* [13] believes that lift generation by an aerodynamically shaped object is related to vortex generation. They argue that "lift can only be maintained if the airplane continues to generate trailing vortices, thus requiring a power plant to overcome the vortex drag needed to generate lift". Let's take a symmetric NACA 0009 airfoil as an example, when the airfoil is placed in a flow at an angle of attack of 0 degrees, both the

upper and the lower surface generate vorticity rotating in opposite directions, and they merge at the trailing edge to form a wake with net drag only. When the airfoil is suddenly rotated to a positive angle of attack, the lower surface generates counter clockwise vorticity predominates for a period of time, forming a distinct trailing edge vortex until a new equilibrium is established. The sharpness of the trailing edge is critical to prevent the viscous flow from going around the edge. It is also responsible for generating a vortex at the trailing edge, and a vorticity distribution in upper and lower surface boundary layers with a net positive total vorticity. Cebeci et al. explained the unsteady airfoil flow extensively in their book [13].

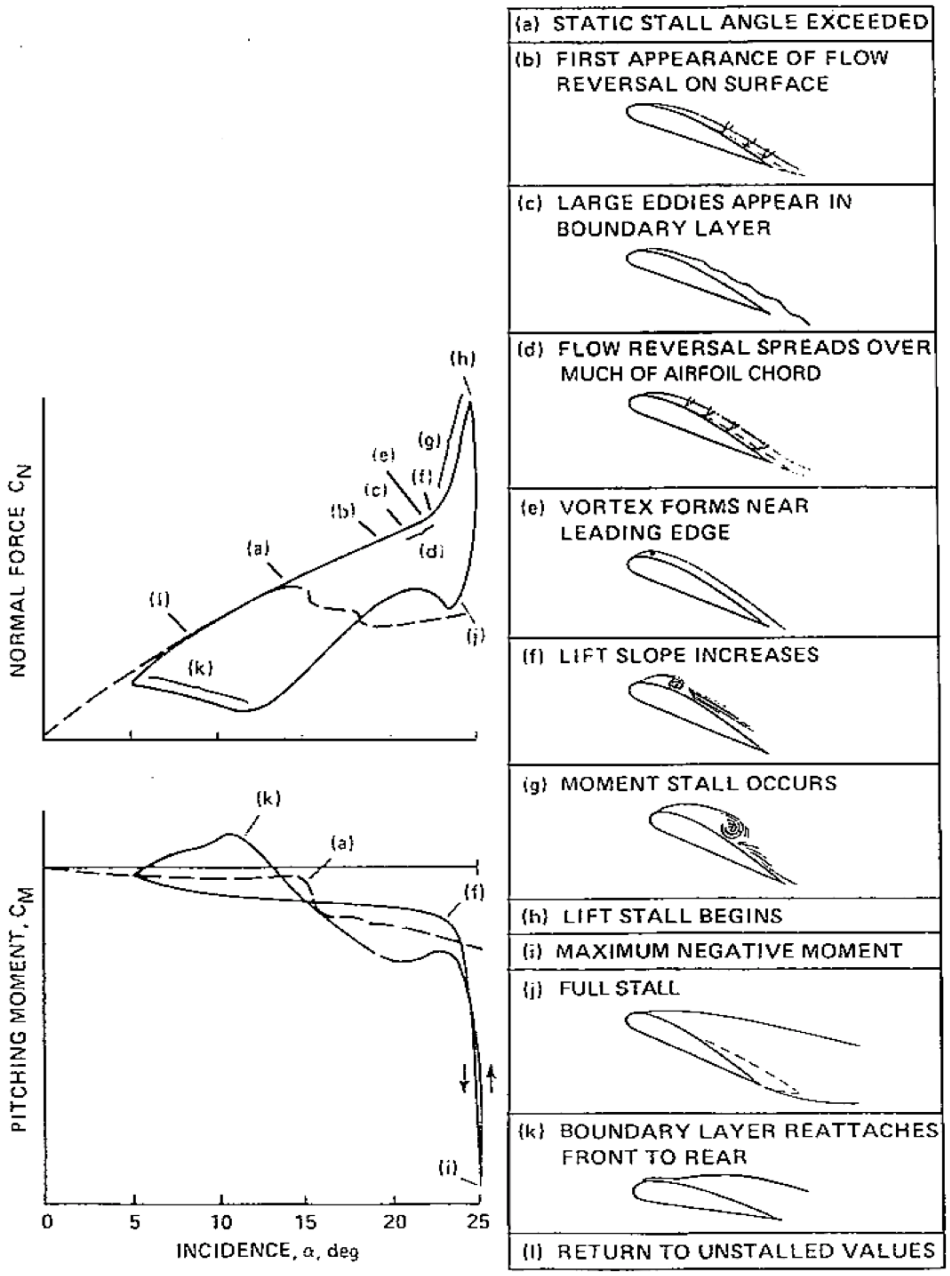

Figure 1 Dynamic stall events on a NACA 0012 airfoil at low free-stream Mach number [14]

This theory not only explained the relationship between lift and vorticity, but also linked static stall to dynamic stall, which is another complex topic related to flows around airfoils. They believe that static stall should be treated as a special case of dynamic stall, and the stall angle and maximum lift changes are due to sudden airfoil movement of the unsteady flow. A stall condition occurs when an aerodynamically shaped object increases

its angle of attack beyond a certain angle called critical angle of attack, and the lift of the object starts to decrease. The stall angle of attack depends heavily on the profile of the aerodynamic object and the flow Reynolds number. Static stall is measured when the airfoil is fixed at a certain angle of attack in the flow, but a dynamic stall condition is more complex to measure or predict. Figure 1 illustrates the event of dynamic stall in detail. When the angle of attack increases rapidly and exceeds the static stall angle, the pressure change on the upper surface takes time to establish, so the boundary layer separation is delayed. A vortex is formed near the leading edge of the airfoil and moves downstream along the upper surface. This vortex induces suction and causes lift increase. Depending on the airfoil shape and angle of attack, this dynamic lift increase can reach as much as 50% to 100% more than the maximum static lift [15]. The vortex downstream movement also causes a significant increase in nose-down pitching moment. As the vortex moves past the trailing edge, the lift decreases very rapidly. The stall creates a large separation from the leading edge to the tailing edge, and it's highly unsteady and three-dimensional, comparable to the vortex shedding behind bluff bodies. Despite the complexity of the airfoil flow and stall condition, it is important to study these phenomena, as they are crucial to aircraft designs. Dynamic stall prediction is also important in helicopter, turbo machinery and wind turbine designs.

Dynamic stall prediction methods are largely based on empirical or semi-empirical approaches, which were well documented in Ericsson and Redingin's paper [16]. Thanks to the advancement of CFD in recently years, complex airfoil flows with transition and pressure-induced separation can be solved using time dependent simulation methods such as LES. LES is chosen over classic RANS models because RANS models suffer from the inability to capture these transient flow patterns owing to their inherent damping of unsteady motion [17], whereas LES is capable of separating and solving flow patterns based on their eddy sizes, so it's more accurate in capturing details of time-dependent flows. However, one noticeable weakness of LES when solving flows near solid obstacles is that the method tends to solve the near wall flow directly. Given that energy containing eddies are so small, a very refined wall-normal grids are needed in boundary layers. The near wall grid refinement requirement is the same as a DNS requirement,

which is very expensive to achieve for high Reynolds numbers. This is why some researchers chose the DES method. DES is a hybrid simulation method of RANS and LES, and was first proposed by Spalart [17]. DES solves turbulent flow away from any wall using LES method, but solves flow near solid surface with a RANS model. This new approach smartly avoids expensive computations near the surface of the airfoil, but still generates accurate results. Another advantage of DES and LES over RANS airfoil simulation is the ability to simulate time dependent unsteady flow movement that occurs under dynamic stall conditions. When the DES method is applied to moving geometries and adaptive grids in the NWT, it is possible to simulate and visualize the complete event of a dynamic stall.

Many studies have been conducted regarding airfoils and the stall conditions using classic numerical methods, and some of them can give relatively accurate stall predictions, but most of these methods are based on decades of trial-and-error and fine-tuning of coefficients. Ekaterinaris and Platzer [15] did an extensive review of many classic numerical stall prediction methods. They found that while many classic models such as B-L models were able to predict dynamic stall, but these models had to be tested and validated every time for each new flow condition of interest, and some models were proven to be accurate for some flows but inaccurate for others. They also found that three-dimensional (3D) dynamic stall was the most difficult flow to simulate, and very few numerical investigations were performed. A 3D airfoil simulation conducted by Piziali [36] was only able to find agreement with 50% of the span location. In 2012, Douvi *et al.* tested three different turbulent models commonly used in today's commercial CFD programs in two-dimensional (2D) simulations, and they still found about 13% disagreement between the experimental and simulated lift coefficients [35]. Weber and Ducros [19] did several airfoil simulation comparisons between LES and RANS methods, and showed the advantages of LES. There are also a few investigations carried out with DES methods, such as the research conducted by Schmidt and Thiele [17]. The next Chapter presents a detailed discussion of the 3D DES simulation.

CHAPTER 3 RESEARCH METHODOLOGY

This chapter reviews the implementation details of many of the features of the NWT, which was developed over many years and by many different people. Section 3.1 gives an overview of the NWT's existing features. In order to predict flow around airfoils, new features and improvements were added for this research, Section 3.2 to 3.5 discuses the major changes made to the NWT.

3.1 THE NUMERICAL WIND TUNNEL

Given the complexities of the airfoil dynamic stall flow, a powerful simulation software is required for its simulation. The software must be able to simulate three-dimensional unsteady turbulent flows around a moving solid object. Many commercial CFD programs are capable of simulating these types of flows, but in order to generate accurate results for complex curved external boundary flows, most of these programs require a body-fitted grid system as an input of the simulation. However, if the geometry is constantly moving in the flow like in this research, it is a time consuming task to regenerate a different body-fitted grid system every time the geometry changes position. This problem is solved perfectly by the CFD program chosen for this research, the NWT. The NWT is a 3D CFD tool capable of simulating flows around multiple, complex, moving geometries using a Detached Eddy Simulation turbulence model. The key features of the NWT are summarized in the following list:

- Capable of simulating both laminar and turbulent flows.
- Uses DES method for turbulent flows.
- The Navier-Stokes equations are discretized using finite volume method.
- Unstructured Cartesian grid method with anisotropic adaptation for simulating time-dependent incompressible flows. The grid system uses a collocated, cell-centred arrangement for velocity and pressure.
- An IB method is used to handle boundaries around the geometry.

- Multi-processor parallelization and automatic load rebalancing for distributed processing, which greatly speeds up computation. This feature requires two additional open source libraries: The first one is a Message Passing Interface called MPICH, which is a standardized and portable message-passing software library routines designed for inter-processor communication and parallel computing (http://www.mpich.org/); The other piece of software is a Parallel Graph Partitioning and Fill-reducing Matrix Ordering system called ParMETIS, which is an MPI-based parallel library that implements a variety of algorithms for partitioning meshes, and for computing fill-reducing orderings of sparse matrices (http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview).
- It also uses a 3D unsteady implicit discretization of the equations, with Crank-Nicholson time advancement at a constant time step, which is a highly accurate method for unsteady flow.

Because of the anisotropic adaptation featured in the NWT, the setup time and steps of the flow simulations are greatly reduced. The following is a short summary of steps to complete a simulation using the NWT:

- Create the targeted geometry using Computer Aided Drafting (CAD) software such as AutoCAD, and generate the geometry in STL file format.
- Define the domain, boundary conditions and initial flow conditions (Reynolds number and inlet velocities) before the simulation. Define the initial position of the geometry in the simulation domain.
- Compile the program on a Linux multi-processor environment to fully utilize the parallelization and automatic load-rebalancing feature of the NWT. In this research, the program was deployed on 32 or 64 processors clusters provided by The Atlantic Computational Excellence Network (ACEnet), which is a consortium of Atlantic Canadian Universities providing researchers with high performance computing (HPC) resources (www.ace-net.ca).
- Run the program simulation for a number of time steps, the program saves and outputs current flow properties for the entire domain with a predefined frequency.

- Stop the program and move the geometry to a new position. Resume the program with previous flow data, simulation continues as if the geometry is continuously moving in the unsteady flow.
- The program calculates the fluid pressure value in each cell and velocities in all three dimensions. The data is available for the user at each time step.
- A new routine was created for the NWT to calculate and integrate lift and drag coefficients using the pressure data around the surface of the airfoil. This routine automatically determines tangential directions of all wall cells around aerodynamic shaped geometry of any size.
- Flow data is exported using either TecPlot or OpenDx file format, and can be easily imported into either software to generate visualizations. Both TecPlot (http://www.tecplot.com/) and OpenDx (http://www.opendx.org/) are popular visualization software packages used for engineering, scientific and analytical data processing.

3.2 DETACHED EDDY SIMULATION

As mentioned briefly in Chapter 2, choosing a suitable method of turbulent simulation is not an easy task. Some CFD software use custom tuned RANS models for some specific flows, but no general solutions for different types of flows can be obtained with one RANS model, and RANS's capability is really limited when used to predict flow around complex geometries and or unsteady flows. Some researchers opt for the LES method instead of RANS. Despite many advantages the LES method tends to over predict the turbulent viscosity in the near wall region of the flows. To improve the predictions it requires a DNS level of mesh refinement in the near wall regions. Fortunately DES is able to combine advantages of both RANS and LES, without significantly increasing the mesh density near the wall. The RANS model used in NWT is Baldwin-Lomax model, which is known for producing good results for the boundary layer, especially when properties such as displacement and velocity thickness are not known [3]. Camelli and Lohner [20] successfully combined Baldwin-Lomax (B-L) and LES methods in their

research to study wall-bounded turbulent flows with large separation regions. The NWT's first implementation used LES, and then improved by adding DES to handle the near wall regions based on Camelli and Lohner's [20] method. The method is referred to as hybrid modeling, and in recent years is quickly gaining popularity.

The idea behind the DES method is simple. Because the B-L model performs better near a solid wall and the LES performs much better anywhere else in the domain, the DES combines these two methods. The B-L model is used near the wall, while the LES is used elsewhere. As illustrated in Figure 2, the B-L model (represented by the dashed line) and the LES (represented by the dotted line) in a) are combined and become the hybrid DES model shown in b). The calculation of the turbulent viscosity begins at the wall, $y = 0$, using the B-L method and searches through all cells away from the wall and in the direction normal to the wall, until it reaches the distance y_{max}, at which point the calculation is switched to the LES.

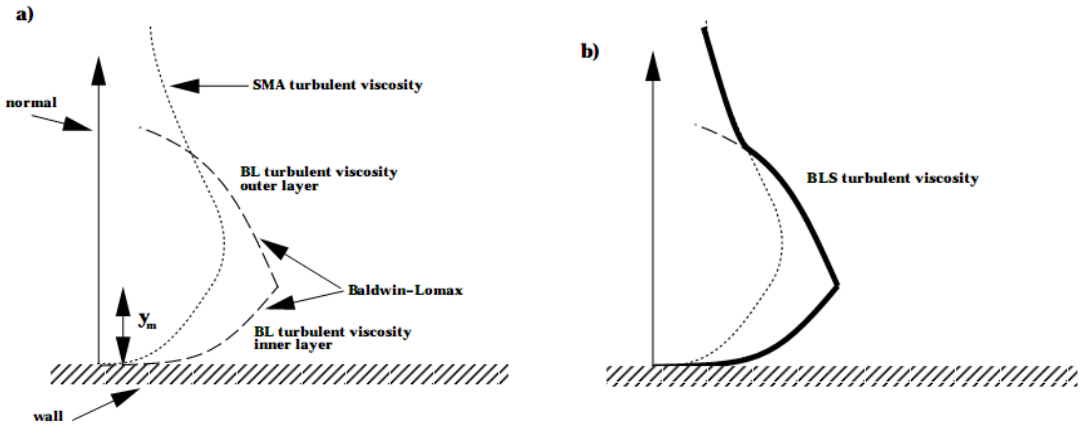

Figure 2 The B-L model, the LES Smagorinsky model and the combined profile of the hybrid DES model [20]

The algorithm is based on the Law of the Wall [21]. This theory assumes strongly damped flow velocity fluctuations near a smooth solid surface immersed in turbulent flow, so velocity fluctuations in all three dimensions at $y = 0$ are defined to be $u' = v' = w' = 0$. The law also states that there is a small layer of laminar flow near the wall within the turbulent boundary layer, which is called laminar sublayer. A suitable

form of Reynolds number in this region of flow is defined as $y+ = \frac{yu_*}{\nu}$, where u_* is referred to as the friction velocity and $y+$ is the dimensionless distance from the flow to the wall. It is clear that when the value y is small enough, or when this layer is close enough to the wall, the sublayer will behave as a laminar flow. A dimensionless value $y+$ is used in NWT to determine the wall distance of each cell in the domain, and the critical value of $y+$ is set at 7 [21][22] as confirmed in Figure 3. The DES used in NWT is summarized as follows:

- If $y+ \leq 7$, use B-L model
- If $y+ > 7$, use LES method

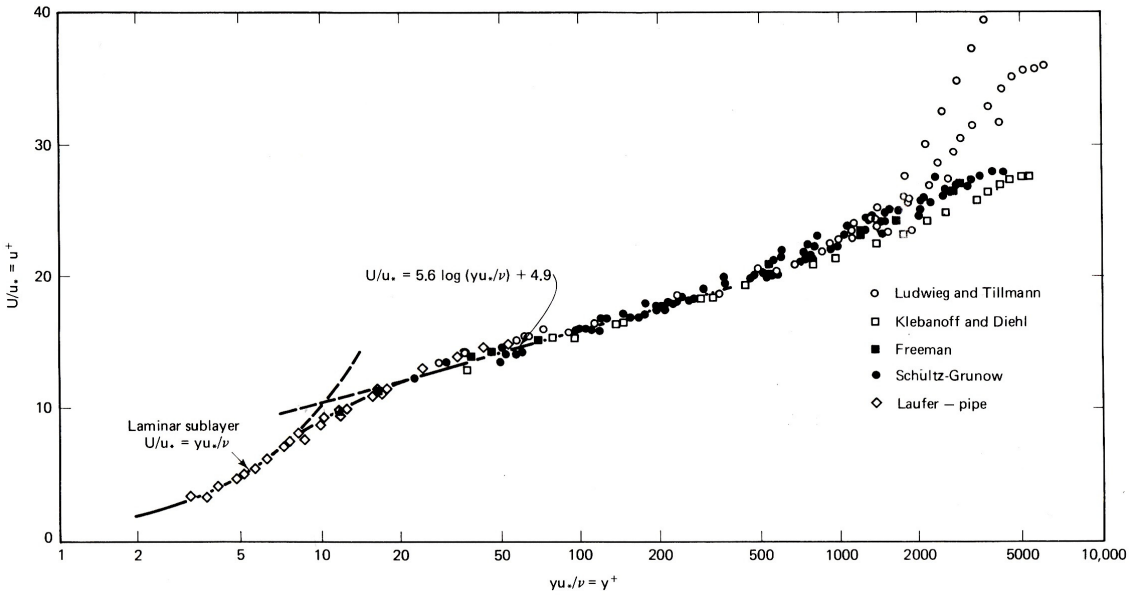

Figure 3 The Universal Law of Wall plot for turbulent boundary layers on smooth, solid surfaces [21]

Ludwig Prandtl developed the Law of the Wall in 1930s [21], which assumes that the dimensionless velocity u+ is function of $y+$, where $u+$ is a ratio of the local fluid velocity to the friction velocity. It is assumed the following:

$$\tau_w = \mu \frac{\partial U}{\partial y} \tag{3.1}$$

$$\text{where } U = \frac{y\tau_w}{\mu}$$

$$\text{and } \frac{U}{u_*} = \frac{yu_*}{\nu}$$

$$\text{thus } u+ = y+ \tag{3.2}$$

In these equations, U is the stream wise velocity, u_* is the friction velocity, and τ_w is the wall shear. The equation 3.2 is confirmed in Figure 3 in the region below $y+ \approx 5-7$, so the existence of a laminar sublayer is also confirmed. The $y+$ calculation algorithm was heavily modified for this research. In addition, $y+$ calculation also depends on an existing NWT feature called Surrounding Cell method, which was implemented to locate surrounding cells of any IB cells. The method was originally implemented for simple geometries such as a cylinder, and now it is rewritten for this research so that it works with more complex geometries such as an airfoil.

3.3 SURROUNDING CELL METHOD (SCM)

This SCM was added to the NWT for simulating flow around immersed cylinders [37]. Two tasks must be accomplished by SCM, finding the normal vectors to the boundary surface and finding mesh cells that intersect these normal vectors. These tasks are not easy to implement because of the complexity of the geometry shape and the unstructured Cartesian grid generated by the NWT. One goal of this research is to create a simulation tool that can easily simulate flows around geometries of any shapes and forms, so the SCM must be able to find normal vectors dynamically along any solid surface. In addition, because the mesh is changing during every time step of the simulation due to anisotropic adaptation, the mesh cells that intersect with the normal vectors are constantly changing as well. As a result, a SCM algorithm must be executed every time the mesh changes. In order for the SCM to work with solid boundaries on a moving airfoil, the method was modified and improved for this research.

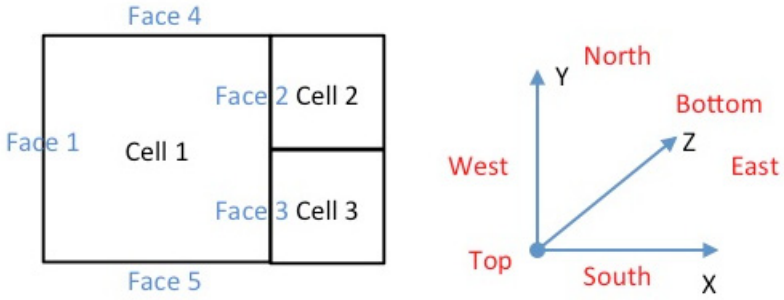

Figure 4 2D NWT mesh illustration and coordinates system

Before going into details of the SCM algorithm, it is necessary to explain the NWT Cartesian grid system design. Figure 4 shows a simplified 2D mesh with Cartesian grids and its coordinate system. In a 2D coordinate system, each cell has one center and four faces. In this example, Cell 1 has west face 1, east face 2 and 3, north face 4 and south face 5. In a 3D coordinate system, each cell has two additional faces along Z direction called top and bottom faces. Cell 2 and Cell 3 are both eastern neighbours of Cell 1. In the NWT grid data structure design, both Face 2 and Face 3 are linked to the eastern side of Cell 1, but Cell 2 and Cell 3 are only linked to Face 2 and Face 3 respectively.

Figure 5 illustrates several unstructured Cartesian mesh cells used in the NWT. Cells and cell faces in the NWT are stored in a list and sorted in three dimensions. It is easy to propagate along the list in sequence and access all cells and faces, but there is no apparent way of advancing from one cell to the next in a random spatial direction. For instance in Figure 5, if the Cell 3 is highlighted, it is possible to advance to Cell 4 or reverse to Cell 2 easily in the cell list, but there is no easy way to move from Cell 3 to Cell 1 or Cell 7, because the cell list keeps no info regarding the relative location of these three cells. In order to advance along a random spatial direction in the domain such as a normal vector, it is essential to be able to find surrounding cell information of any given cell, which is one of the features implemented in SCM.

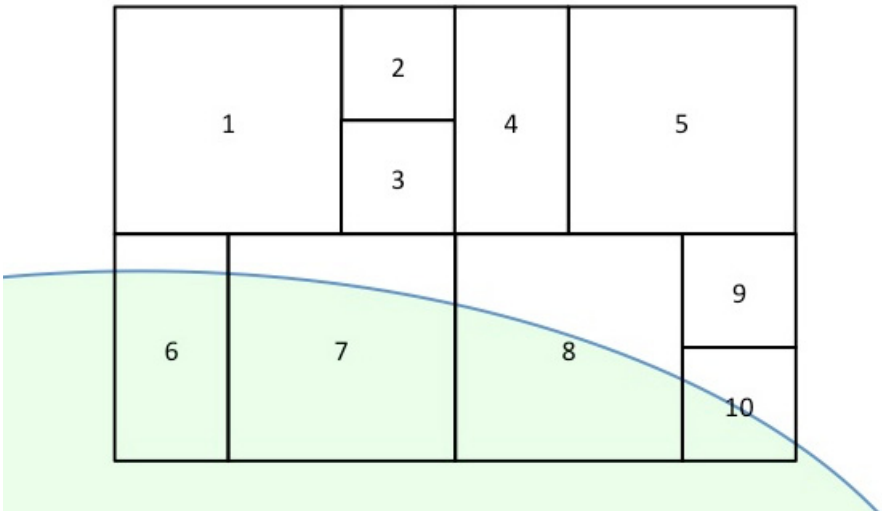

Figure 5 Unstructured Cartesian cells intersect with a solid wall

Now imagine the boundary wall of an airfoil intersecting Cell 6, 7, 8, and 10 as shown in Figure 5, all these cells now become IB cells, which are essential to the boundary layer flow simulation. It is also important to identify surrounding neighbouring cells of these IB cells, namely Cell 1, 3, 4, 5, and 9, because they are immediately adjacent to IB cells. These neighbouring cells are significantly affected by the wall boundary near them, and they are also important component of the interpolation method we will talk about next. The SCM method is capable of identifying the IB cells and their neighbouring cells.

One disadvantage of the Cartesian grid system is that for a curved boundary, the actual wall boundary does not coincide with IB grid. As a result, an IB method has to approximate the boundary face velocities using a certain interpolation method. The easiest and most common method to approximate a Cartesian grid IB is setting fluid velocities attached to IB faces inside the boundary to 0. For instance, in Figure 6, this method will set U_1 to 0. However, this method contradicts no-slip condition enforced on P (X_p, Y_P), because the approximated Cartesian grid IB boundary does not overlap with the surface of the curved solid wall [37], and the issue is worse when the mesh is coarse. Kim and Choi proposed a bilinear interpolation scheme to solve this problem [23]. Figure 6 illustrates the bilinear interpolation method. U_1 is the IB face velocity that needs to be calculated; U_2, U_3, and U_4 are the cell face velocities surrounding P.

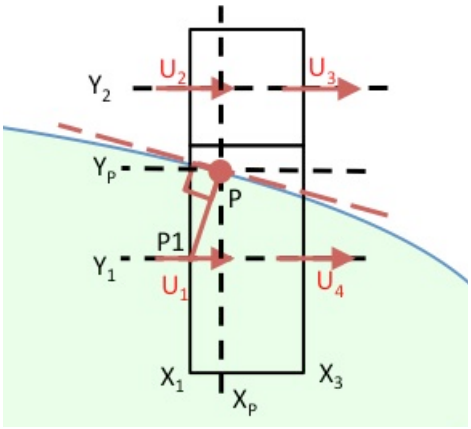

Figure 6 Second order bilinear interpolation

Based on bilinear interpolation scheme of [23], the formula to obtain U_1 is the following [37]:

$$U_1 = -\frac{\left[\alpha(1-\beta)U_2 + (1-\alpha)(1-\beta)U_3 + (1-\alpha)\beta U_4\right]}{\alpha\beta} \tag{3.3}$$

where $\alpha = \frac{X_3 - X_P}{X_3 - X_1}$ and $\beta = \frac{Y_2 - Y_P}{Y_2 - Y_1}$

In order to find the IB velocity U_1 of face F_1, the SCM is used to locate $U_{2\text{-}4}$ of $F_{2\text{-}4}$ and its steps are explained in details as follows:

- Locate F_1 and determine if it's a boundary face.
- If yes, find cells C_a and C_b that are linked to west and east sides of F_1, as shown in Step 1 of Figure 7.
- Find cells C_{bN} and C_{bS} that are linked to the north and south sides of C_b, as shown in Step 2 of Figure 7.
- Find faces F_2 and F_3 that are linked to the west and east sides of C_{bN}, as shown in Step 3 of Figure 7. Also find F_4 that's linked to the east side of C_b.
- If $F_{2\text{-}4}$ are outside of the IB, return these faces, otherwise just repeat the steps.

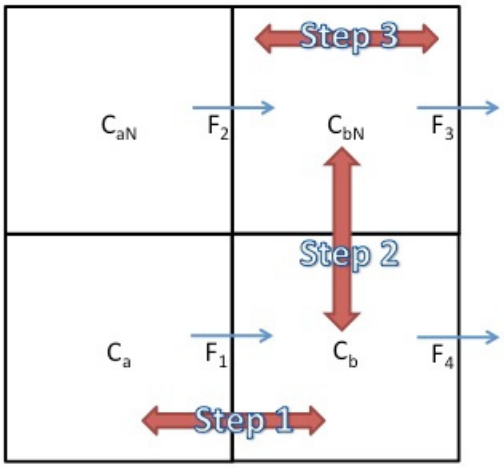

Figure 7 Surrounding Cell Method steps

In this particular illustration of SCM steps, the method goes from the southwest to the northeast direction, but in actual calculations, the algorithm travels in all directions, until it finds proper surrounding cell faces. Once the surrounding faces are located, the velocity and coordinates of these faces are stored for the IB face velocity calculation.

Then the method must also locate the point P as shown in Figure 6, which is defined as the intersection of the IB and the wall-normal line passing through P1. As mentioned before, the original algorithm was written for flow around a cylinder shaped geometry, so the location of P is relatively easy to find because the wall-normal line PP1 also passes through the center of the cylinder. However for a complex geometry such as an airfoil, there is no such relationship between the center of the geometry and the wall-normal line. Hence, a new method is created to find these wall-normal lines.

The NWT uses STL file format as input source file of the solid geometry, and one feature of this file format is that it automatically includes normal vectors for its solid walls. The NWT was planned to utilize this feature but later cancelled because it was hard to implement. Many three-dimensional model formats (including STL format) describe a solid wall using a number of interconnected raw unstructured triangulated surfaces, each triangle consists a normal vector and 3 vertices. Hence when a program reads out these

triangles and connects them into geometry, the location and orientation of the geometry is instantly known. However, the NWT also chose to use unstructured Cartesian Grid mesh system, and the grid cells do not match with the surface triangles, so it's hard to convert the geometry normal vectors into the wall-normal lines needed in this case.

A different approach was taken to solve this issue: to search through geometry surface triangles and determine the shortest distance between the certain point on surface and the face of U_1. As shown in Figure 8, Since the F_1 is located at the center of an IB cell face, it is always near one of the surfaces of the geometry. Among all distances between F_1 and any points on the surface of the geometry, the shortest distance is guaranteed to be the wall-normal distance F_1P. As shown in Figure 8, no matter how many lines are drawn from F_1 to the surface of the geometry, and no matter where P1 and P2 are defined on the surface of the geometry, $F_1P < F_1P1$ and $F_1P < F_1P2$ are always true statements.

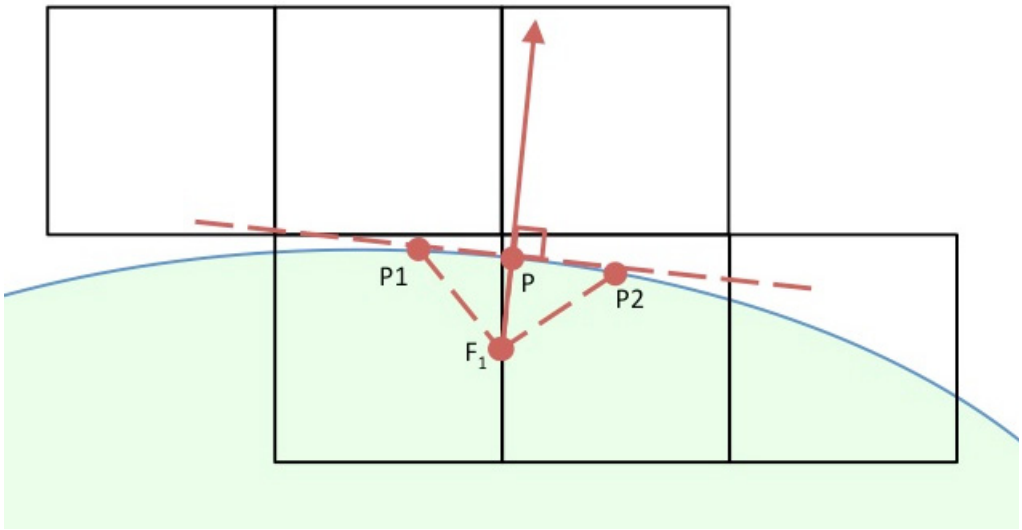

Figure 8 Wall-normal line illustrated in Cartesian grid system

The following steps describes the method to find the wall-normal line from an IB face:

- Start from IB face F_1 and keep a record of its coordinates.
- Loop through geometry surface triangles and their vertices.

- Calculate distance between F_1 and all triangle vertices. Record only the shortest distance and the coordinates of the vertex.
- Return the shortest distance line vertex P.

3.4 LIFT AND DRAG COEFFICIENTS CALCULATION

The next necessary step is to calculate lift and drag coefficients for the airfoil. When a flow moves around a solid geometry, the flow exerts a surface force on the object, which can be described with two stresses, namely viscous stress or induced wall shear stress and pressure induced by the normal stress. The integration of these stresses around the airfoil produces a resultant force. The component of the surface force perpendicular to the oncoming flow direction is called lift, and the component parallel to the oncoming flow direction is called drag.

Originally the NWT was only capable of calculating average lift and drag of a cylinder, but it cannot find lift and drag for any other shaped geometry such as an airfoil. The key difference between these two geometries was the method to obtain the tangential direction along the surfaces. As shown in the following equation:

$$F_D = \int_A dF_D = \int \left(-P\cos\theta + \tau_w \sin\theta\right) dA \tag{3.4}$$

$$F_L = \int_A dF_L = \int \left(-P\sin\theta + \tau_w \cos\theta\right) dA \tag{3.5}$$

lift and drag forces, F_D and F_L, can be calculated by integrating normal stress and shear stress. In the equation, P is the local pressure for each IB cell, and τ_w is the local shear stress for each IB cells. θ is the angle between the direction of the local pressure and oncoming flow direction. Figure 9 illustrates the angle θ for both the flow around a cylinder and the flow around an airfoil. Since the local pressure is always perpendicular to the local surface area on which it acts, θ has a different value for each local pressure, so each θ has to be calculated case-by-case based on their location on the surface of the geometry. These equations are true for both airfoils and cylinders, but the θ values are more difficult to find for an airfoil than a cylinder.

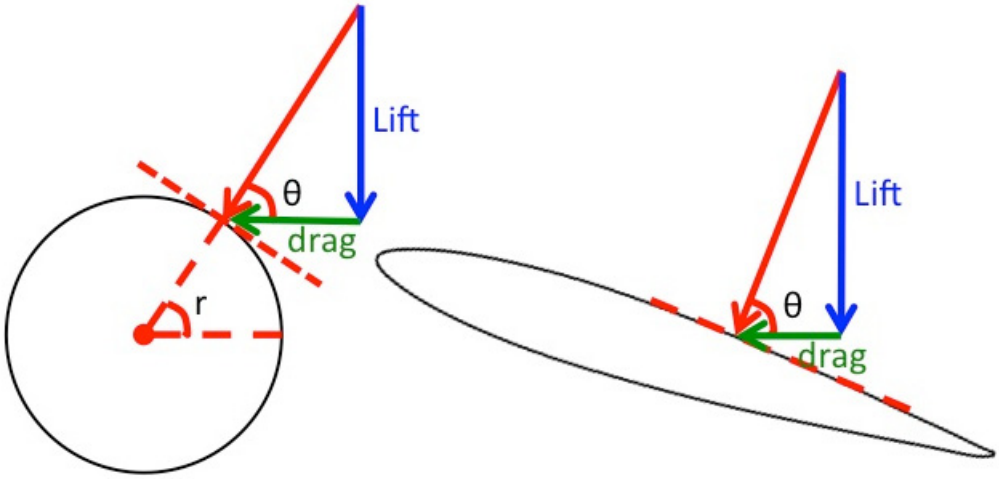

Figure 9 Comparison of the angle θ for an airfoil and a cylinder

As shown in the cylinder view of Figure 9, each local pressure is perpendicular to the local cylinder surface area on which it acts, and all local pressures point directly to the center of the cylinder, so the angle r is always equal to θ. However, for an airfoil, there is no such relationship between r and θ, so a different approach was taken.

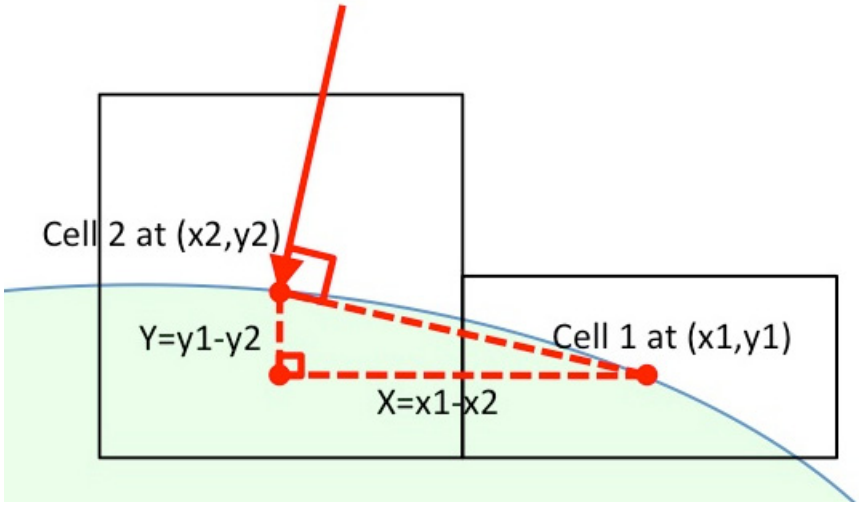

Figure 10 Enlarged IB cell view

Figure 10 illustrates an enlarged view of two surface cells. In order to find θ for the pressure, an approximated slope was determined using two neighbouring cells. In this

case, the surface slope of $Cell\ 2$ is:

$$slope = \frac{y1 - y2}{x1 - x2}$$
$$\text{and } \theta = \arctan\left(slope\right) - 90° \tag{3.6}$$

Another approximation of the slope of $Cell\ 2$ can be obtained from an average slope of both $Cell\ 1 - Cell\ 2$ and $Cell\ 2 - Cell\ 3$. Both types of approximations were programmed and their results were compared, but the overall lift and drag of the airfoil calculated in both methods were nearly identical. As a result, the first approach was used.

The final step of the calculation is to find lift and drag coefficients using the following formulas respectively:

$$C_D = \frac{2F_D}{\rho V^2 A} \text{ and } C_L = \frac{2F_L}{\rho V^2 A} \tag{3.7}$$

In order to apply these formulas and develop working functions in the NWT, large amount of programming is conducted and their steps are summarized below:

- Start process by identifying IB cells that intersect with the surface of the geometry. One complication here is that the NWT features multi processor parallelization, so boundary cells from different processors have to be located on each processer and them transferred to one main processor for combined calculation. The inter-processer transfer is accomplished using MPICH routines.
- Put all boundary cells into a list, and remove any duplicates.
- Sort each cell according to the angle it makes with respect to the center of the geometry and positive X, Y or Z coordinate vector direction based on the shape and orientation of the geometry.
- Consider all boundary cells in their sorted orders and find θ for each one of them.
- Apply lift and drag formulas on each cell.
- Integrate to obtain the total lift and drag coefficient of the geometry.

3.5 MESH REFINEMENT AND ADAPTATION CRITERIA

The mesh adaptation algorithm in the NWT works automatically during execution, but the user has the ability to control one of two adaptation criteria before each simulation. These two adaptation criteria are capable of controlling the refinement level of the entire flow domain, but the user has no control over local grid resolution. Local resolution control would be a useful feature in programs like the NWT that uses automatic grid adaptations, because it helps the user focus the computation in areas that are important, while keeping the total computation cost down to a minimum. As a result, a new mesh adaptation feature was added to the existing automatic adaptation criteria.

The adaptation criteria are based on the work of Ham *et al.* [29], the user can either set an error tolerance c or set a total number of cells for the domain. Both methods are essentially two different ways of controlling the same error tolerance value in order to change the Cartesian control volume sizes for each cell, so a desired total mesh size is achieved. However, only one of these criteria can be set for each simulation. The Cartesian control volume ΔV is defined as follows:

$$\iiint_{\Delta V} u_i \, dV = \left(u_i\right)_c \Delta x \Delta y \Delta z + \varepsilon_{u_i} \tag{3.8}$$

where ε_{u_i} is the error indicator in the control volume approximation and $\left(u_i\right)_c$ is the velocity at the enter of the volume. The error can be estimated by integrating the 2D Taylor series expansion of u about the centroid over the control volume, and the expression becomes:

$$\varepsilon_{u_i} = \frac{\Delta x^3 \Delta y \Delta z}{24} \frac{\partial^2 u_i}{\partial x^2} + \frac{\Delta x \Delta y^3 \Delta z}{24} \frac{\partial^2 u_i}{\partial y^2} + \frac{\Delta x \Delta y \Delta z^3}{24} \frac{\partial^2 u_i}{\partial z^2} \tag{3.9}$$

The optimal mesh is as the largest cell volume while the sum of all error indicator squares is less than the square of a specified tolerance c^2:

$$\sum_{i=1}^{3} \varepsilon_{u_i}^2 \le c^2 \tag{3.10}$$

If the estimation is substituted into the tolerance expression, the following equation is obtained:

$$\sum_{i=1}^{3} \frac{\Delta x^3 \Delta y \Delta z}{24} \frac{\partial^2 u_i}{\partial x^2} + \frac{\Delta x \Delta y^3 \Delta z}{24} \frac{\partial^2 u_i}{\partial y^2} + \frac{\Delta x \Delta y \Delta z^3}{24} \frac{\partial^2 u_i}{\partial z^2} \le c^2 \tag{3.11}$$

Ham *et al.* [29] proposed a simplified form to the criterion:

$$\frac{\Delta x^3 \Delta y \Delta z}{24} F_{xx} + \frac{\Delta x \Delta y^3 \Delta z}{24} F_{yy} + \frac{\Delta x \Delta y \Delta z^3}{24} F_{zz} \le c^2 \tag{3.12}$$

where $F_{xx} = \sqrt{\left(\frac{\partial^2 u}{\partial x^2}\right)^2 + \left(\frac{\partial^2 v}{\partial x^2}\right)^2 + \left(\frac{\partial^2 w}{\partial x^2}\right)^2}$

$$F_{yy} = \sqrt{\left(\frac{\partial^2 u}{\partial y^2}\right)^2 + \left(\frac{\partial^2 v}{\partial y^2}\right)^2 + \left(\frac{\partial^2 w}{\partial y^2}\right)^2}$$

$$F_{zz} = \sqrt{\left(\frac{\partial^2 u}{\partial z^2}\right)^2 + \left(\frac{\partial^2 v}{\partial z^2}\right)^2 + \left(\frac{\partial^2 w}{\partial z^2}\right)^2}$$

Hence a cell that satisfies the above formula will satisfy the error criterion so that the mesh will be close to optimal if cell sizes are maximized while satisfying this formula. Maximized cell volumes $\Delta x_{\max}$, $\Delta y_{\max}$, and $\Delta z_{\max}$ can be obtained using the method of Lagrange multipliers:

$$\Delta x_{\max} = \left(\frac{64 c^2 F_{yy} F_{zz}}{F_{xx}^4}\right)^{\frac{1}{10}} \tag{3.13}$$

$$\Delta y_{\max} = \left(\frac{64 c^2 F_{xx} F_{zz}}{F_{yy}^4}\right)^{\frac{1}{10}} \tag{3.14}$$

$$\Delta z_{\max} = \left(\frac{64 c^2 F_{xx} F_{yy}}{F_{zz}^4}\right)^{\frac{1}{10}} \tag{3.15}$$

One problem with these dimensions is that if the velocities are locally or globally linear in one or more of the coordinate directions, then $F_{x_i x_i}$ will be zero. So Ham *et al.* [29] also proposed the following dimension calculations in the case of one linear direction:

$$\Delta x_{\max} = \left(\frac{144 c^2 F_{yy}}{F_{xx}^3}\right)^{\frac{1}{8}} \tag{3.16}$$

$$\Delta y_{\max} = \left(\frac{144 c^2 F_{xx}}{F_{yy}^3} \right)^{\frac{1}{8}} \tag{3.17}$$

$$\Delta z_{\max} = \Delta z_{current} \tag{3.18}$$

and in the case of two linear directions:

$$\Delta x_{\max} = \left(\frac{24c}{F_{xx}} \right)^{\frac{1}{3}} \tag{3.19}$$

$$\Delta y_{\max} = \Delta y_{current} \tag{3.20}$$

$$\Delta z_{\max} = \Delta z_{current} \tag{3.21}$$

$\Delta y_{current}$ and $\Delta z_{current}$ are the current cell volumes of their respective dimensions. When the targeted dimensions of a cell in all three directions are obtained, the NWT compares its current size with the targeted size, if the current size is larger than the target, a refine operation will be registered, otherwise a coarsen operation will be called. As a result, if the user chooses to control the error tolerance value, dimensions of each cell can be calculated directly; on the other hand, if the user chooses to control the total number of cells, the dimensions of all cells can be calculated with the same formulas as well.

A brief review of the Cartesian grid dimension is need at this point to help understand how the adaptation criterion is applied to control the dimension of mesh cells. Each Cartesian cell is a three-dimensional cuboid, and its length, height and depth dimensions are parallel to the x, y, and z axes respectively. Since the NWT uses unstructured Cartesian mesh, each cell is represented using two sets of values: refinement level and index. A refinement level L_i represents the number of refinement needed for the given cell to reach its dimension from the global dimension in the direction of *i*-axis. Each refinement is defined as the operation that divides one of the current cell dimensions by 2. An index I_i represents the structured Cartesian index the cell would have if all cells in the direction of *i*-axis were refined to its level.

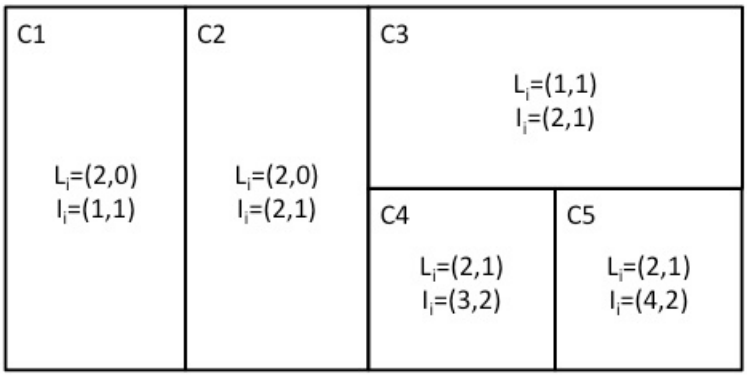

Figure 11 Cartesian grid index and level illustration cell

Figure 11 illustrates an example of cell levels and indices in a simplified 2D view. If these five cells occupy the entire simulation domain, each cell can be identified uniquely with only a L_i and an I_i values. For example, cell C2 has a refinement level of 2 and a index of 2 on the x-axis, which means the x dimension of cell C2 is defined to be ¼ of the global x dimension at $\Delta x = \frac{L_{global}}{2^{L_x}} = \frac{L_{global}}{2^2} = \frac{L_{global}}{4}$, and it's the second cell if all cells in the x direction were refined to the same level. By applying the adaptation algorithm mentioned above to cell C2, if $\Delta x > \Delta x_{\max}$, the level of C2 will be increased by 1, so it's new x dimension will be divided by 2, so the refined new $\Delta x = \frac{L_{global}}{4} \times \frac{1}{2} = \frac{L_{global}}{8}$.

However, as mentioned at the beginning of the section, the current implementation of the NWT controls the cell dimensions by manipulating error tolerance and/or total cell number, hence changes to the refinement level are always applied to the entire domain. For instance, if the error tolerance value is increased to twice of its previous value, the dimension of each direction for every cell in the domain will increase to four times as before. This creates a problem when the user only wants to refine a certain area of the flow, but not the entire domain. For instance, in the airfoil simulation, it is desirable to have a high-resolution mesh near the IB area of the domain, and keep the rest of the domain resolution as low as possible to keep the computation cost down. The new feature solved this issue by checking $y+$ values for each IB cells. Recall in section 3.2, in order

to successfully implement the DES method for the sublayer flow, the dimensionless distance $y+$ was calculated, and it is essential for $y+$ values of the IB cells to be less or equal 7. A new routine was added to the NWT so that every time an IB cell $y+$ values exceeds 7, the code sets a flag for that cell, so the cell will be refined in the next time step. The following steps briefly describe the algorithm:

- Use SCM to locate an internal IB cell.
- Calculate $y+$ for the cell.
- If $y+ > 7$, flag the cell for refinement.
- In adaptation routine, if the cell is flagged for refinement because of $y+$, check current refinement level. Compare it with user defined global minimum dimension level, if it's not going to exceed the minimum level after the refinement, return true.
- Check neighboring cells, if the cell dimension after refinement is more than twice as large or twice as small as its neighbor cell, flag its neighbors to recursively refine, and return true.
- Refine the cell if both previous conditions are met.
- Refine other cells based on global adaption criteria.

CHAPTER 4 RESULTS

This chapter presents the numerical results gathered from the NWT simulations of flow around NACA 0009 airfoil. Section 4.1 describes the setup and configurations used in the NWT for the simulation. Section 4.2 presents an extensive analysis and observation of the numerical results. It introduces details of experimental data for the airfoil, and then compares numerical results before and after various improvements made to the NWT. It also discusses both static and dynamic stall prediction results using the NWT. The validity of the numerical results is studied in last part of Section 4.2.

4.1 Simulation Setup and Configuration

Before a NWT simulation is started, the simulation domain has to be properly configured. This section uses a simulation of turbulent flow around NACA 0009 airfoil to illustrate all necessary setup processes for NWT to work properly. The flow described in this section is set to a Reynolds number of 127,000, and the airfoil is rotating around its center at a rate of 2 degrees per second.

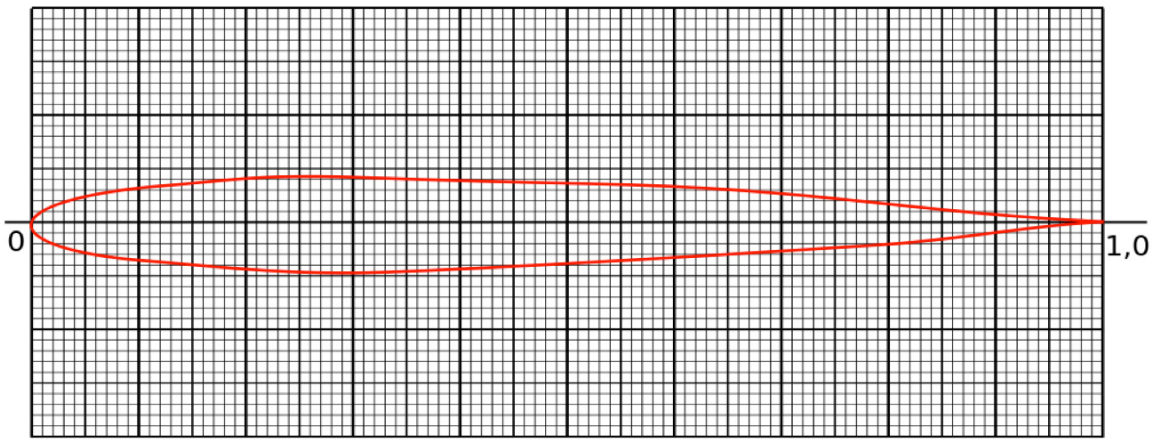

Figure 12 2D view of NACA 0009 airfoil

As shown in Figure 12, the geometry used for this simulation was a solid symmetric

NACA 0009 airfoil with a chord length of 1, and maximum thickness of 0.09. The 3D model of NACA 0009 airfoil is drawn with AutoCAD based on data from The UIUC Airfoil Data Site (http://www.ae.illinois.edu/m-selig/index.html) and exported in STL format before it is imported into the NWT. As illustrated in Figure 13, the domain of for the simulation is set to a rectangular cuboid shape of 5x2x1 in size. The center of the airfoil was located at the point (1.6, 1, 0.5). This arrangement ensures that the domain has enough space to show the flow in the wake of the airfoil, and enough space for the airfoil to rotate to a higher angle of attack.

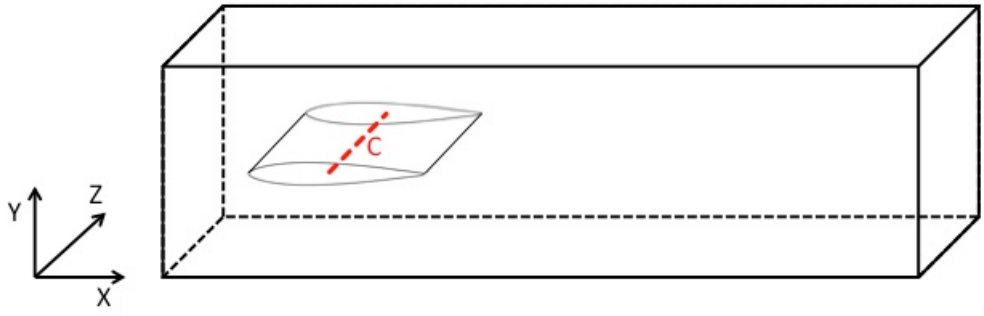

Figure 13 Simulation domain

The domain has 6 boundaries, West is the inlet; East is the outlet. The South, North, Top, and Bottom boundary are set to be periodic boundary condition. When the simulation starts, the flow moves in through the West boundary, goes around the airfoil, and then exits through the East boundary. The Reynolds number based on the chord length is set to 127000 for this study, so that the results can be easily compared with available experimental data. At the beginning of the simulations, the airfoil is horizontally placed in the domain just like it is shown in Figure 13. To test the airfoil at different angles of attack (AOA), the airfoil is rotated around the centerline C, which is parallel to the Z-axis.

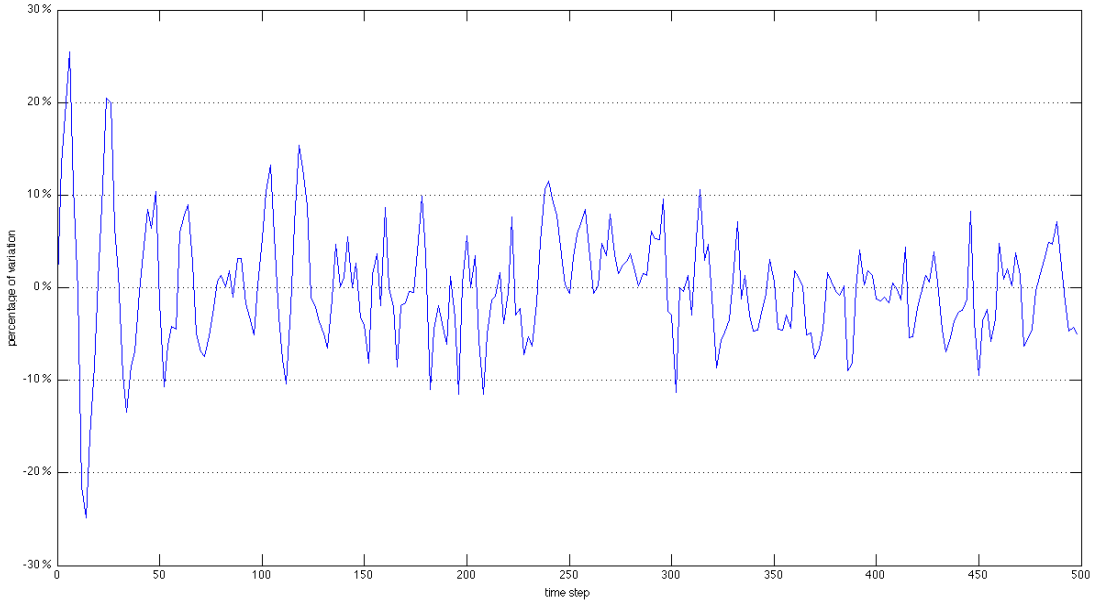

Figure 14 Percentage of lift fluctuation v.s. time steps

The airfoil remains horizontal for 1000 time steps or 1 second, and then starts to tilt around Z-axis at a pace of 500 time steps or 0.5 second per degree. Since the purpose of this research is to study lift and drag coefficients of an airfoil, these coefficients are selected for measuring the percentage of fluctuation. This number of steps at a given angle was chosen based on the fluctuation of the lift coefficient. As shown in Figure 14, the values of lift coefficients are taken from a simulation just after an AOA increase from 12 degrees to 13 degrees. The lift coefficients are normalized to percentage fluctuation around the mean lift coefficient value. The simulation lasted 500 time steps as shown by the X-axis, and it is easy to see that the fluctuations are decreasing as the time increases. At the beginning of the AOA change, the fluctuations reaches as high as +/-25%, but towards the end of 500 steps, the fluctuations deceases significantly to about +/-5%. In most cases, after about 200 time steps, the fluctuation has decreased to below +/-10%. It is desirable to prolong the simulation steps after each AOA increase for better accuracy, but the computation costs also increase. As a result, 200 time steps is selected to be the minimum time steps for each AOA change of 1 degree, and 500 time steps is selected as the cutoff point for each AOA change of 1 degree.

Every time the airfoil rotates, the geometry is reloaded into the domain at a new orientation, but all fluid properties are preserved from the previous time step. The NWT regenerates meshes to adapt to the new geometry location and shape using previous mesh information as a starting point. This rearrangement is accomplished by the NWT's automatic anisotropic Cartesian grids algorithm. However, this process would significantly change flow properties in many existing mesh cells, some cells moved from inside the geometry to the outside and vice versa. In this case, these inside cells used to contain no meaningful fluid properties before the transition, now require recalculation based on their new locations and new neighbouring cell flow properties. Some other cells are relocated the other way around, which also require adjustment. Because the geometry movement is relatively small at 1 degree at a time, the NWT is able to adapt the changes in the flow domain. A larger change of AOA each time might require more time steps to stabilize and may even cause divergence of the numerical solution.

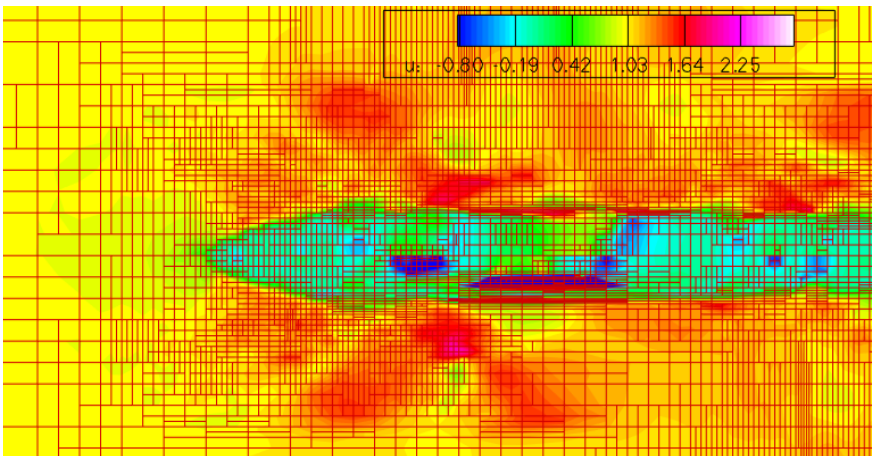

Figure 15 Airfoil NACA 0009 mesh and velocity 2D view

Figure 15 shows the mesh generated by NWT around the airfoil. The view shown in the figure is the center slice of the domain on the XY plan, viewed from the Z-axis direction. Cells in the domain are refined automatically based on combined criterion of velocity gradients, user-defined total cell number, and adaptation tolerance value. The figure clearly shows that finer meshes are generated near the boundary surface of the airfoil, where velocity gradients are large. The maximum number of cells for the simulation was set to 500,000, but the actual cell count was constantly changing because of NWT's

automatic anisotropic Cartesian grids adaptation. Refer to section 3.5 for a more detailed description of the adaptation criteria.

4.2 AIRFOIL RESULTS

This section presents the simulation results for the flow around a NACA 0009 airfoil and their comparison with experimental data obtained under similar test conditions. The simulated results are divided into three parts: first, the lift and drag coefficients obtained from original NWT code pressure distribution results; second, after applying improvements and mentioned in Chapter 3, more accurate lift and drag coefficients are obtained and analyzed; third, the results and visualizations were used to predict dynamic stall of the airfoil.

4.2.1 Experimental Data

A custom wind tunnel test using a rapid-prototyped NACA 0009 airfoil was selected as the experimental comparison baseline, for this research. Detailed steps and descriptions of this test are given in Bilgen's paper [24]. The NACA 0009 airfoil of choice has a maximum thickness of 11.3 mm, span of b = 133 mm and a chord of c = 127 mm. The airfoil is fabricated using a Stratasys Dimension SST rapid-prototyping machine available at the Mechanical Engineering Department of Virginia Tech using A P400 ABS type material. The wind tunnel used in this test is a low-speed, open circuit wind tunnel located at the Center for Intelligent Material Systems and Structures (CIMSS) in 381 Durham Hall, Blacksburg, VA. The wind tunnel setup and instrumentation were designed and fabricated specifically for the analysis of the airfoil. The airfoil is tested for lift and drag coefficients at an average flow speed of 15m/s and a chord Reynolds number of 127,000. Both experimental and numerical tests are intentionally set to the same Reynolds number, so that the results are comparable.

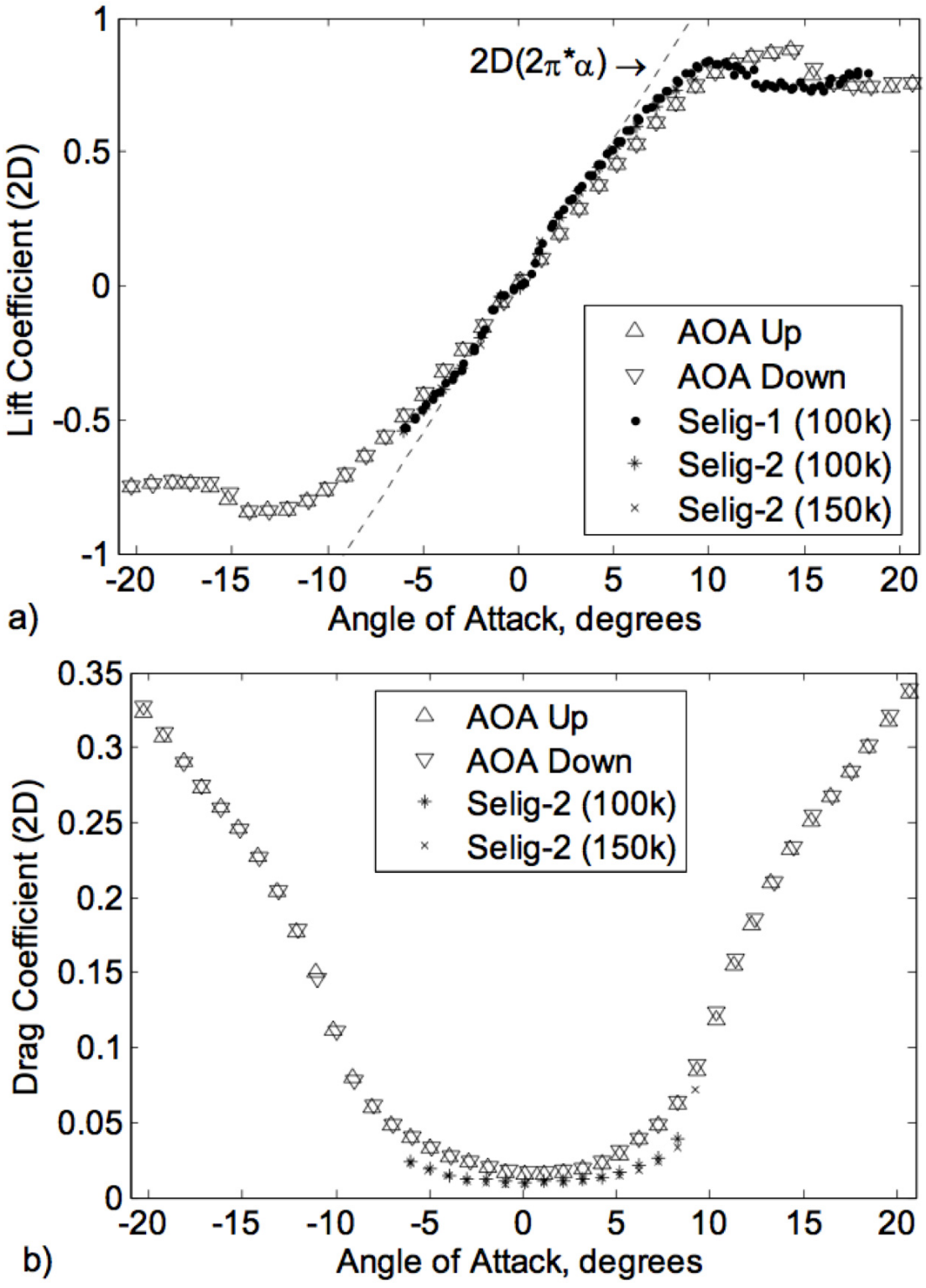

Figure 16 a) Lift and b) drag coefficients vs. angle of attack comparison for NACA 0009 airfoils [24]

In order to test the validity of the lift and drag coefficient data collected by Bilgen [24], the coefficients are compared with custom wind tunnel tests by Selig of University of Illinois (UIUC) in two publications: Selig et al. [26] (labeled as Selig-1), and Selig et al. [25] (labeled as Selig-2) as shown in Figure 16. Reynolds numbers in Selig's tests are 100,000 and 150,000 as presented in the figures, which are very close to the 127,000 Reynolds number used in this research. The measurements are recorded at a rate of 100 Hz for a period of 60 seconds in order to get an accurate mean.

After some careful comparisons using test data shown in Figure 16, it is concluded that coefficient results from both tests match up with each other very well. Lift coefficient curve slopes from both tests are nearly identical; while drag coefficients from Bilgen [24] is slightly higher than drag coefficients from Seliget al. [25][26], they do share similar slopes and trends Bilgen [24] gives a short analysis of possible causes of the differences in measurements, including wind tunnel wall effects, higher turbulence level in Selig's et al. [25] tests, and airfoil surface roughness differences. As a result, if the NWT returns similar numerical lift and drag coefficients, this research can be considered a success.

4.2.2 First Time Simulation Data

The NWT was developed and modified over many years and by many different people, it was first designed as a 2D single processor LES simulation tool, and later on improved by adding more features such as three-dimensional simulation, multi-processor support, and DES. The NWT was applied to simulate cavity flow, channel flow, and then flow around sphere and cylinder, and produced accurate prediction results [27] [28]. However, the NWT was never tested on complex aerodynamic shaped geometries such as an airfoil. This section presents the lift and drag coefficients data calculated using NWT with no airfoil optimization for the first time. The purpose of this section is to compare results before and after various modifications and improvements made to NWT for this research, so that the achievements of this research can be highlighted. Lift and drag coefficients obtained at this stage is shown in Figure 17 and 18 below.

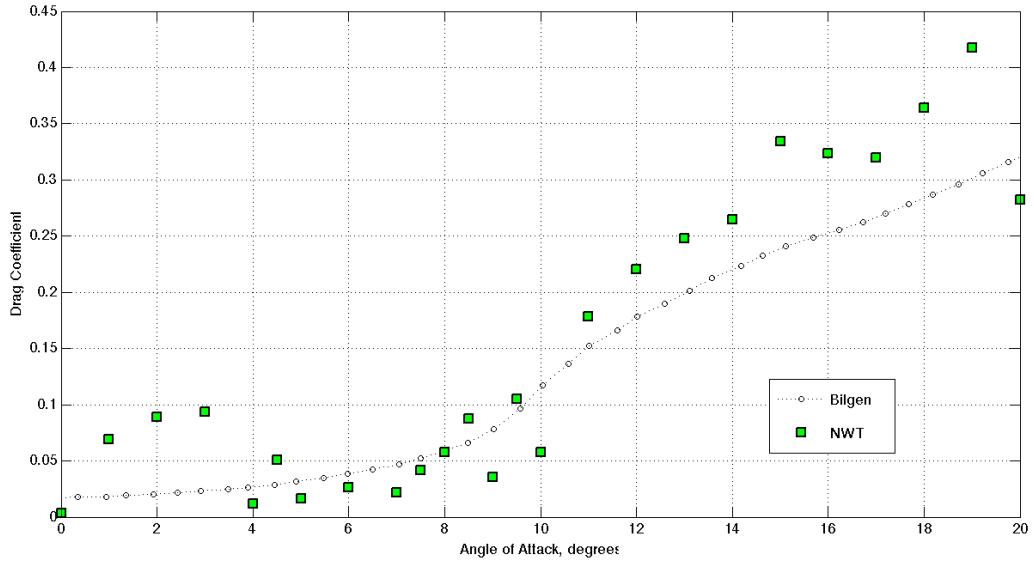

Figure 17 First test NWT simulated Drag Coefficients compared with Bilgen [24] experimental data

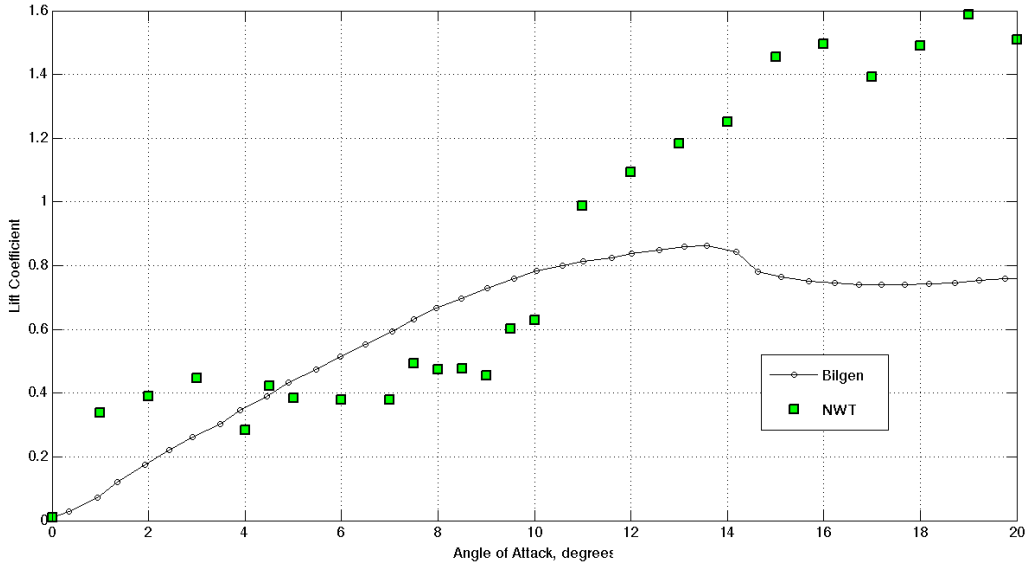

Figure 18 First test NWT simulated Lift Coefficients compared with Bilgen [24] experimental data

Figure 17 and 18 present results of the numerically simulated drag and lift coefficients of NACA 0009 airfoil immersed in turbulent flow while its AOA changes from 0 to 20

degrees (plotted with green dots), the numerical data is compared with experimental data of the custom made NACA 0009 airfoil from Bilgen's [24] research mentioned in 4.2.1 (plotted with black and white dots). In the drag coefficient graph, the numerical results matches the experimental results relatively well, and the simulated values mostly followed the pattern of changes of the experimental data, except for AOA from 1 to 3 degrees and above 14 degrees. Specifically, the simulated values in these ranges are too high. There are also fluctuations in the range between 4 and 14 degrees, but not as large as the rest of the data. In the lift coefficient graph, the coefficient at AOA of 0 degree matches the experimental value of 0 lift coefficient, which is correct considering the NACA 0009 airfoil is a symmetrical airfoil. But the rest of the curve is very different from the experimental data: the lift coefficient jumped from 0 to 0.35 in just 1 degree of AOA increase; between 1 and 10 degrees, the coefficients slowly increase from 0.35 to 0.61, where as the experimental lift coefficient increases smoothly from 0.02 to 0.8; in the range above AOA 11 degrees, as the airfoil approaches and exceeds the stall angle, the experimental lift coefficient reaches the highest point and starts to decrease near AOA of 13 degrees, but the numerical data failed to show the stall effect and continue to increase. The simulated lift coefficients above 11 degrees are also considerably larger than the experimental values. Some analysis was made to the data, and the following issues were identified:

Time steps per AOA increment requirement: One big issue with the numerical coefficients is the large fluctuation observed each time the AOA increases. As mentioned earlier in section 4.1, after each AOA increase during the simulation, the NWT detects a new position of the airfoil, and automatically invokes adaptive mesh methods and tries to generate new cells around the moved geometry while reusing old flow information from the previous time step. This process requires a number of time steps to finish and stabilize. One way of checking this stability is to calculate fluctuation percentage of lift and drag coefficients at each AOA. As concluded in section 4.1, at least 200 time steps are required for each degree of AOA increase, so lift and drag coefficients calculated during this 200 steps of adaptation period usually contain large fluctuations. When the data in Figure 18 and 19 were collected, only 200 times steps were executed during each degree

of AOA increase.

IB resolution: The refinement of the mesh is essential to the accuracy of the numerical simulation, and it is especially the case for mesh grids near the IB region. As mentioned in section 3.2, in order to successfully simulate the viscous sublayer in the boundary using the BL turbulent model, the dimensionless distance between IB cells and the surface of the geometry must be within $y+ \leq 7$. An IB refinement study was conducted to analyze the refinement level of IB cells surrounding the airfoil at AOA of 0 degree, and the results showed that only 2% of all IB cells are located within $y+ \leq 7$. The results indicate that the refinement level at the IB is too coarse.

One possible cause for the coarse IB cells is the total refinement level defined for the entire domain is too small, or the predefined cell number is too small, so there are not enough cells to cover the entire domain without exceeding the desired total cell number. A study of grid independence was conducted to determine the cause, and the following results were obtained for the airfoil at AOA of 0 degree:

Table 1 Mesh independence study for first time NWT simulated airfoil NACA 0009 at AOA of 0 degree

Number of Cells	Drag Coefficient	Lift Coefficient	Adaptation Criterion
100,000	0.0020	0.0051	2.6E-05
250,000	0.0029	0.0089	5.0E-06
500,000	0.0031	0.0095	1.4E-06
1,000,000	0.0031	0.0096	2.9E-07

The data in Table 1 shows that for the defined domain size, the lift and drag coefficient is grid independent as the number of cells reaches 500,000, and even the number of cells increased to twice as many at 1,000,000, the results barely changed. Table 2 shows the percentage changes between each cell number increase:

Table 2 Mesh independence study for first time NWT simulated airfoil NACA 0009 percentage changes of coefficients

Range of Cells	Drag Coefficient	Lift Coefficient
100,000 – 250,000	31%	42%
250,000 – 500,000	9.6%	6.3%
500,000 – 1,000,000	0%	1%

The percentage changes confirmed that for this simulation domain increasing the number of cells or refining the grid beyond 500,000 cells produces little or no change in the predictions. So 500,000 become the targeted cell number used in the research. In this analysis, the $y+$ values are on average 5 times as large as the desired range. If the global refinement level is increased to this value, the computation cost is too high. These studies showed that a special refinement adjustment for the IB cells is required, so that the DES simulation can work properly.

Stall prediction: The numerical lift coefficients shown in the graph kept growing larger after 14 degrees of AOA. On the other hand, the experimental lift coefficient values stopped growing around 14 degrees and experienced a decrease afterwards, which means the stall AOA for a NACA 0009 airfoil, was about 13 degrees. At this point, the flow above the airfoil top surface should experience significant separation. However, at the time of this simulation, the NWT implementation of DES method failed to simulate the large separation that should appear on the upper surface of the airfoil even long after the experimental tested separation AOA of 14 degrees. Figure 19 shows the flow visualization of AOA at 18 degrees, the flow separation started to occur on the upper surface of the airfoil near the trailing edge of the airfoil, but the separation did not grow to the leading edge of the airfoil, hence no stall is predicted. As mentioned in section 2.2, the stall happens when the flow is completely separated from the upper surface of the airfoil as illustrated in Figure 1(g). As discussed in Chapter 2 and section 3.2, the DES method is capable of predicting with reasonable accuracy flows around airfoils, so the issue here seems to be a combination of implementation errors and lack of specific adjustments of the NWT for the prediction of airfoils or other aerodynamic shaped

geometries.

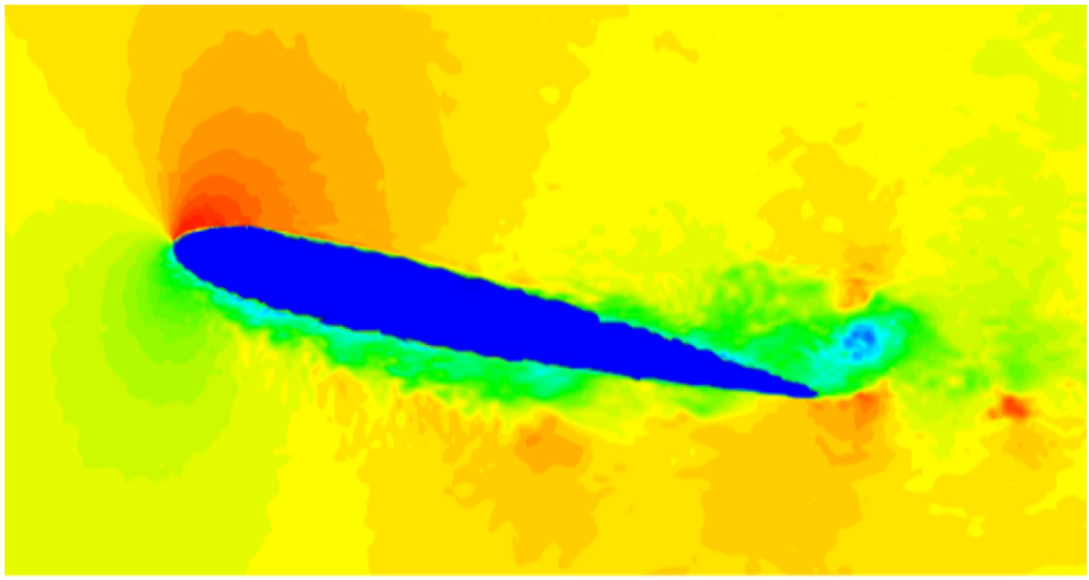

Figure 19 Velocity distribution of first NWT simulated NACA 0009 at AOA of 18 degrees in 2D view

After this first time simulation test, many changes and improvements, as mentioned in chapter 3, were implemented into the NWT in order to fix the issues stated above. Improved results are discussed in the following sections.

4.2.3 Improved Simulation Data and Discussion

In order to obtain more accurate airfoil simulation results using NWT, the NWT code was inspected and reviewed carefully. Then a number of improvements were made to the code, including new features mentioned in chapter 3, and many bug fixes. After implementing these improvements, more accurate lift and drag coefficient predictions were obtained. To test the validity of these improvements, a second simulation test was conducted using the same setting as in the first simulation test recorded in 4.2.2, and the new results are presented below.

Figure 20 shows a view of the mesh in the direction of Z-axis perpendicular to the XY-plane. Table 3 shows a grid-independence study using lift and drag coefficients obtained

from the second test, it's similar to the one shown in the previous section. The results indicated a similar conclusion as before: the grid is independent if the mesh size is above 500,000 cells with adaptation criterion less than 1.4E-06. The mesh shows that the grid in the wake of the airfoil and the grid surrounding the airfoil surface are much more refined than the rest. Since the mesh is automatically refined according to the grid independence study, the entire domain should be sufficiently refined.

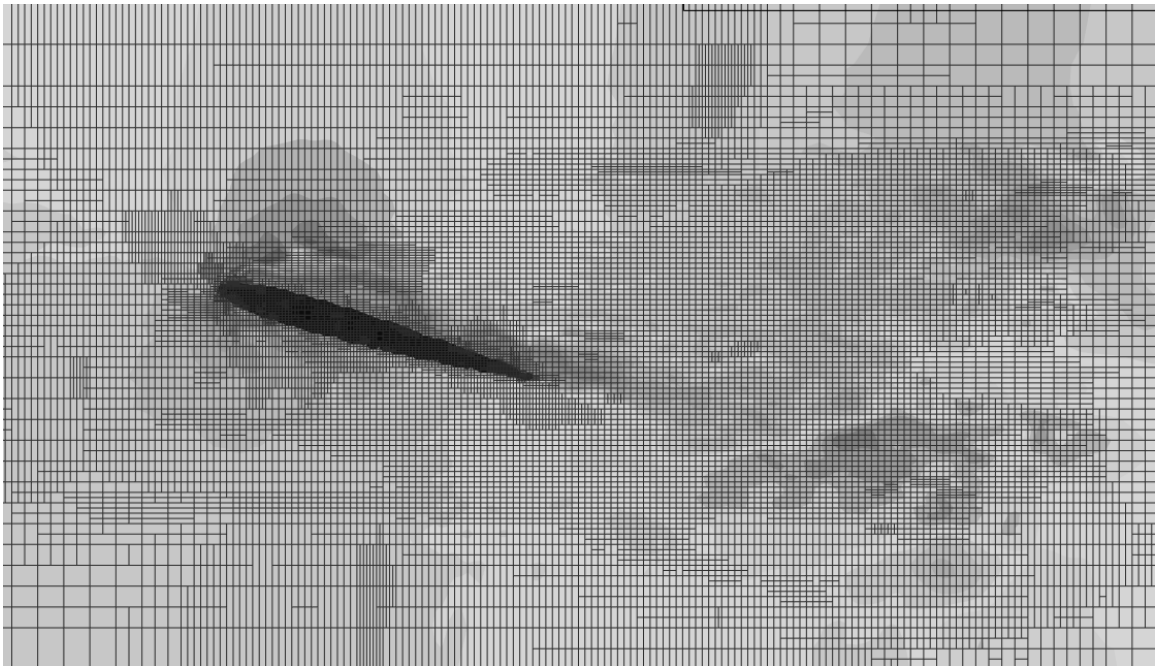

Figure 20 Mesh view of flow around NACA 0009 airfoil at Re = 127,000

Table 3 Mesh independence study for first time NWT simulated airfoil NACA 0009 at AOA of 16 degree

Number of Cells	Drag Coefficient	Lift Coefficient	Adaptation Criterion
100,000	0.283	1.213	2.3E-05
250,000	0.272	1.092	4.8E-06
500,000	0.265	1.003	1.3E-06
1,000,000	0.262	0.986	2.7E-07

As mentioned in the previous section, IB cell refinement level was determined to be too coarse, which causes dimensionless values $y+$ for IB cells larger than the required value of 7 so that the sublayer region failed to use B-L turbulent model. Another IB refinement

study was conducted to analyze the refinement level of IB cells surrounding the airfoil at AOA of 0 degree using the new test data, just like in the previous section, and the results showed that 100% of all IB cells are located within $y+ \leq 7$. Recall in the first test simulation in the previous section, the same percentage was calculated to be only 2%. The results indicate that the refinement level at the IB is much better, and that the viscous sublayer is always simulated with B-L turbulent model, thus producing much better results near the airfoil surface.

One issue mentioned in the previous section about the first NWT simulation was that the flow did not separate from the upper surface of the airfoil even when the AOA was much larger than the experimentally stall angle. The lack of flow separation was concluded to be the main cause for the failure in predicting accurately the lift coefficients. However, in the second simulation test, the separation was successfully reproduced at the same AOA of 18 degrees. As shown in Figure 21, a layer of low velocity flow from the leading edge to the trailing edge covering the entire upper surface of the airfoil. If the flow conditions are compared with the dynamic stall events in Figure 1, it is very similar to the full stall situation described in stage (j). The visualization shows clearly that the flow is separated from the flow and the airfoil is in stall condition. This improvement can directly be attributed to a better simulation of the flow near the surface of the airfoil, which was achieved by refining locally the grid and using a mixing layer type turbulence model in that region.

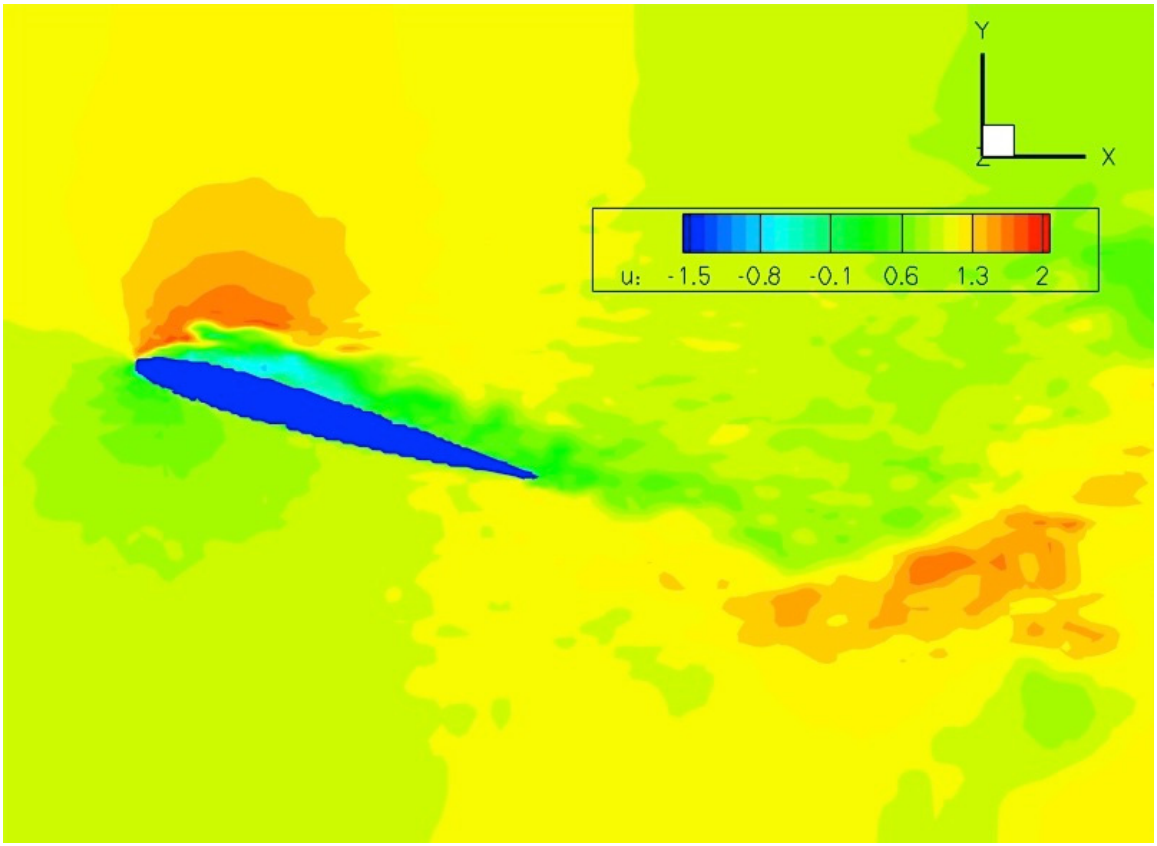

Figure 21 Velocity distribution of second NWT simulated NACA 0009 at AOA of 18 degrees in 2D view

Figure 22 shows another visualization of the airfoil at AOA of 17 degrees with velocity streamline enabled. This figure clearly shows flow separation on the upper surface of the airfoil, and the streamlines indicate vortices are forming near the upper surface.

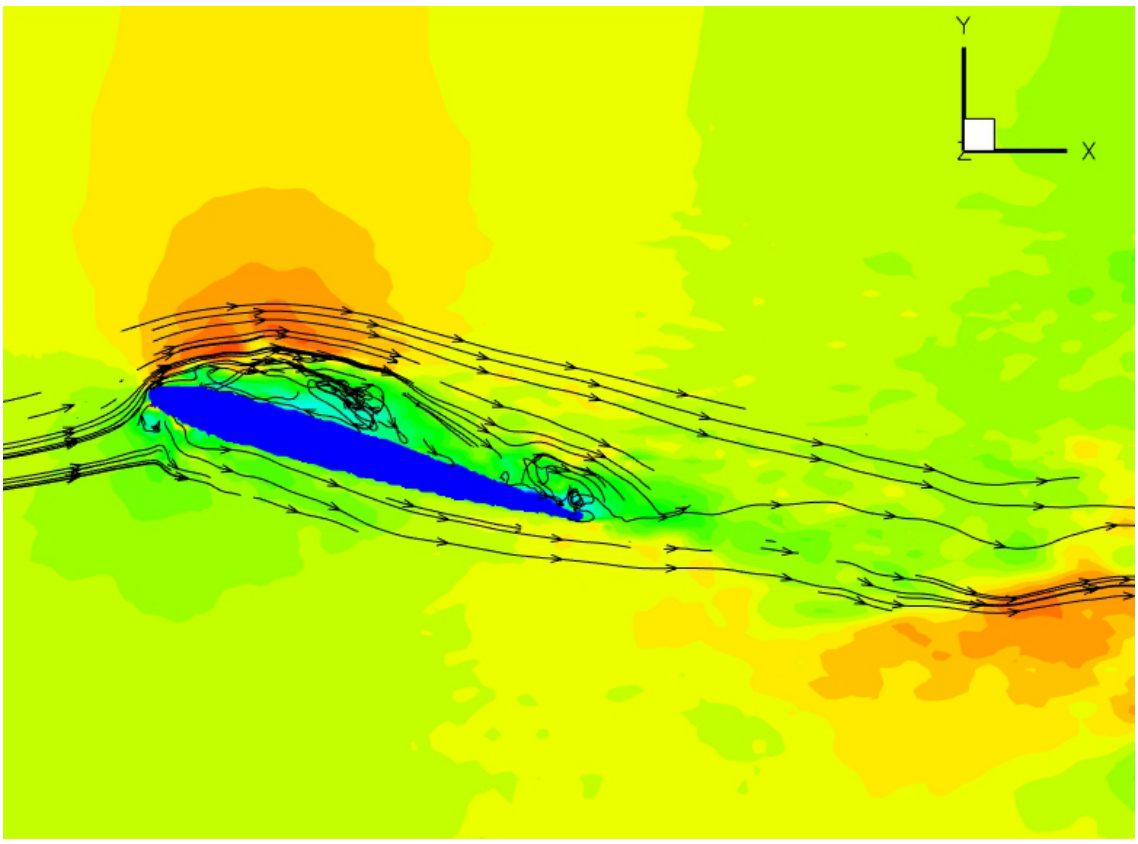

Figure 22 Velocity distribution with streamline of second NWT simulated NACA 0009 at AOA of 17 degrees in 2D view

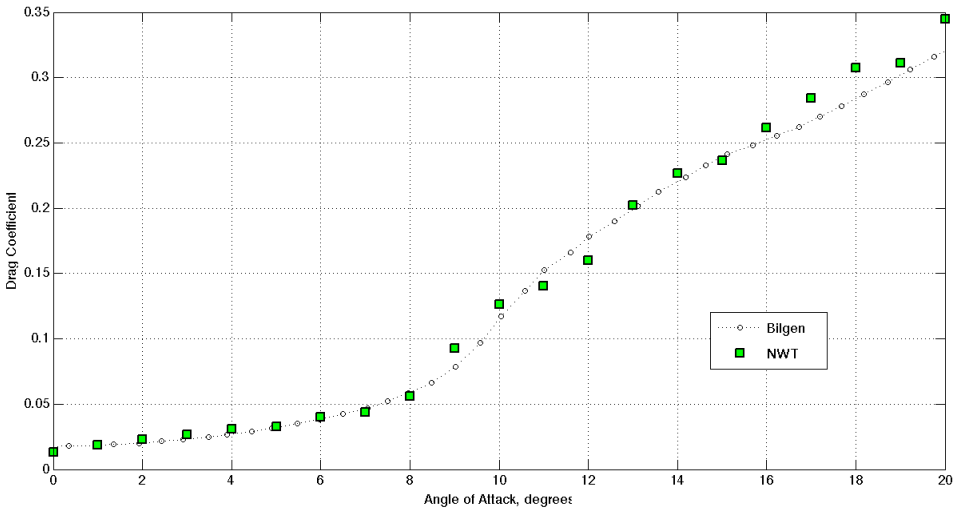

Figure 23 Second test NWT simulated Drag Coefficients compared with Bilgen [24] experimental data

Table 4 Difference study between simulated and experimental drag coefficients

AOA	Experimental	First Test	PDD1	Second Test	PDD2
	Cde	Cd1	(Cd1-Cde)/Cde	Cd2	(Cd2-Cde)/Cde
0	0.0173	0.0031	-82.1%	0.013	-24.9%
1	0.194	0.0694	-64.2%	0.0184	-90.5%
2	0.0202	0.0888	339.6%	0.0227	12.4%
3	0.0235	0.0933	297.0%	0.0267	13.6%
4	0.027	0.0121	-55.2%	0.0309	14.4%
5	0.0325	0.0167	-48.6%	0.0327	0.6%
6	0.039	0.0267	-31.5%	0.0399	2.3%
7	0.0462	0.0216	-53.2%	0.0437	-5.4%
8	0.0588	0.0576	-2.0%	0.0557	-5.3%
9	0.0779	0.0353	-54.7%	0.0926	18.9%
10	0.1148	0.0576	-49.8%	0.1264	10.1%
11	0.1525	0.1787	17.2%	0.1406	-7.8%
12	0.1781	0.2205	23.8%	0.1598	-10.3%
13	0.1997	0.2477	24.0%	0.202	1.2%
14	0.2203	0.2646	20.1%	0.2266	2.9%
15	0.2392	0.3345	39.8%	0.2362	-1.3%
16	0.2525	0.3238	28.2%	0.2615	3.6%
17	0.2671	0.3194	19.6%	0.284	6.3%
18	0.2837	0.3637	28.2%	0.3077	8.5%
19	0.302	0.4178	38.3%	0.3109	2.9%
20	0.3202	0.2822	-11.9%	0.345	7.7%

Figure 23 and 24 shows the simulated lift and drag coefficients from the improved NWT simulations. In these figures, both the simulated lift and drag coefficients have shown significantly improvements in accuracy compared with the same experimental data from Bilgen [24] as in the previous section. The numerical drag coefficients are nearly identical to the Bilgen [24] tested drag coefficients. As shown in Table 4, a comparison study was conducted to show the percentage differences of drag coefficients (PDD2) between the simulated (Cd2) and experimental (Cde) data. In the study, the PDD2 is compared against percentage differences (PDD1) between the first time simulated (Cd1) and experimental (Cde) data. The results showed significant improvements in the second set of drag coefficients data Cd2, especially at AOA greater than 9 degrees. The average PDD2 is 11.9%, compared to the average PDD1 of 63.3%.

In general lift coefficients are much difficult to predict than drag coefficients, since they are much more closely related to the flow condition on the upper surface of the airfoil. The simulated lift coefficients for the second test are also significantly improved compared with the first test. A difference study similar to the drag coefficient difference study was conducted to show the percentage difference (PDL2) between experimental (Cle) and NWT data in the second test (Cl2) as well as the percentage difference (PDL1) between experimental (Cle) and NWT data in the first test (Cl1). The study results shown in Table 5 highlighted simulation improvements in two key areas. The first area is the numerical value accuracy improvements for AOA below 10 degrees. In this range, all lift coefficients match up with the experimental data very closely. The largest difference between numerical and experimental lift coefficients is 0.08 (10.5% difference), which occurred at AOA of 9 degrees. Secondly, the average PDL2 of lift coefficients in the AOA range of 0 to 20 degrees is now 17.1%, compared with the average PDL1 of 67.2%.

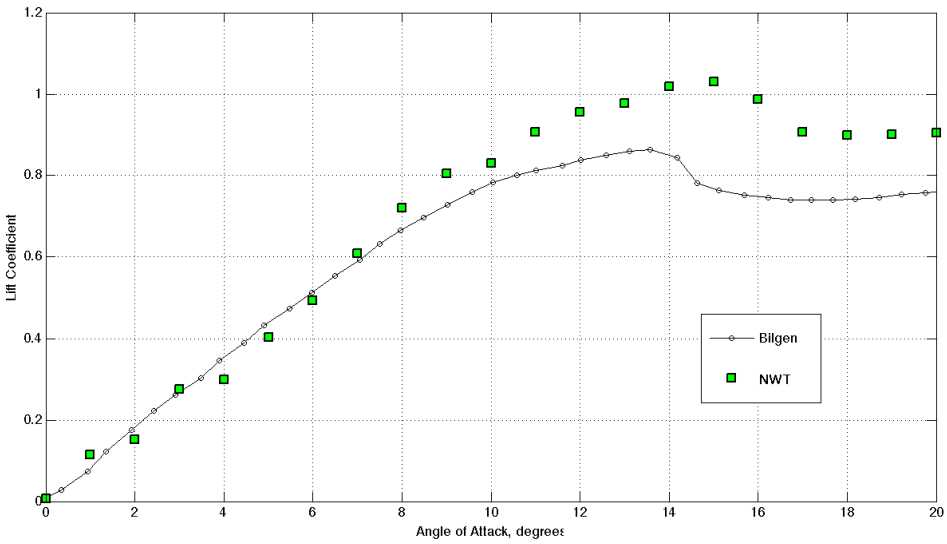

Figure 24 Second test NWT simulated Lift Coefficients compared with Bilgen [24] experimental data

Table 5 Difference study between simulated and experimental lift coefficients

AOA	Experimental	First Test	PDL1	Second Test	PDL2
	Cle	Cl1	(Cl1-Cle)/Cle	Cl2	(Cl2-Cle)/Cle
0	0.009	0.0095	5.6%	0.0059	-34.4%
1	0.081	0.3383	317.7%	0.1145	41.4%
2	0.181	0.3897	115.3%	0.1522	-15.9%
3	0.269	0.4474	66.3%	0.275	2.2%
4	0.358	0.2827	-21.0%	0.2984	-16.6%
5	0.44	0.3836	-12.8%	0.4036	-8.3%
6	0.519	0.3782	-27.1%	0.4927	-5.1%
7	0.591	0.3797	-35.8%	0.609	3.0%
8	0.668	0.4739	-29.1%	0.7209	7.9%
9	0.729	0.4556	-37.5%	0.8056	10.5%
10	0.781	0.63	-19.3%	0.83	6.3%
11	0.813	0.987	21.4%	0.9077	11.6%
12	0.838	1.0936	30.5%	0.9561	14.1%
13	0.859	1.1834	37.8%	0.978	13.9%
14	0.85	1.2504	47.1%	1.0187	19.8%
15	0.769	1.4558	89.3%	1.0313	34.1%
16	0.749	1.4945	99.5%	0.9863	31.7%
17	0.74	1.3913	88.0%	0.9076	22.6%
18	0.742	1.4893	100.7%	0.8999	21.3%
19	0.75	1.5872	111.6%	0.9005	20.1%
20	0.76	1.5102	98.7%	0.9053	19.1%

The discrepancy between the experimental and simulated lift coefficients was observed for AOA of 11 degrees and above. In the experimental data set, the lift coefficients stop to increase at AOA of 13.5 degrees, and then starts to decrease at AOA of 14 degrees. The maximum lift coefficient recorded in the data set is 0.87. However, in the simulated data set, airfoil stall did also occur during the simulation, it just occurred later than the experiment at AOA of 15 degrees. In the simulation, the lift coefficients start to decrease at AOA of 16 degree, with a maximum lift coefficient of 1.03. Despite the difference of maximum lift coefficient, both experimental and simulated lift coefficients share the same slope and profile pattern: the lift coefficients in both cases decrease sharply and stay nearly flat after the maximum lift coefficients are reached. In the experimental data, the lift coefficients after stall stay around 0.75, which marks decease of 12.7% from the

maximum lift coefficient; while in the simulated data, the lift coefficients after stall stay around 0.9, which marks 12.6% decease.

The difference between the experimental data and the simulated data can in part be attributed to the dynamic stall effect of the airfoil. During the NACA 0009 airfoil simulation, the airfoil is rotated upstroke at a constant pace of 1 degree per 500 time steps. Although the rotation is viewed as a discrete movement, the flow data in the simulation is continuous. As explained by Martinat *et al.* [31] in their paper, "If the angle of attack oscillates around a mean value of the order of the static stall angle, a large hysteresis cycle develops concerning the aerodynamic forces and moment coefficients". This effect leads to a dynamic stall, which largely exceeds the static one during the upstroke motion. The dynamic lift coefficients are higher than their static counterparts. However, the NWT is not able to predict the exact dynamic lift coefficients or the dynamic stall. Because dynamic stall is a very complex subject affected by many factors, including specific oscillation frequencies, which is not a feature provided by NWT. Further study of the dynamic effect on the lift coefficient requires the availability of experimental data in comparable condition as the simulation, which was also not available for this research. In order to prove the validity of this assumption, the airfoil is fixed at a post stall angle to obtain static lift coefficients, so the static coefficients are compared with coefficients in the moving case.

Figure 25 shows the average lift coefficients calculated for every 60 time steps over a total simulation length of 2200 time steps. The airfoil fixed at 16 degrees of AOA in the flow for the entire simulation. The graph clearly shows the lift coefficients start at a higher value of above 1 and slowly decrease to 0.93 and stabilized after 2000 time steps, which indicates that a stable static lift coefficient is reached near 2000 time steps. Shown in Figure 26 are the static lift coefficients of NACA 0009 AOA from 10 to 20 degrees compared with dynamic lift coefficients and experimental coefficients from Bilgen [24]. A difference study is conducted and the results are listed in Table 6. The stall angle occurred at AOA of 14 degree, which is a very accurate stall angle prediction, and it's only 0.5 degree larger than the experimental stall angle. The maximum lift coefficient is

calculated to be 1.00, compared to 1.03 dynamic maximum lift coefficient, the static prediction is 15% larger than the experimental value, which is a very reasonable prediction when using CFD to predict stall angle lift coefficients. In 2012, Douvi *et al.* [35] tested three different turbulence models commonly used in today's commercial CFD programs, and the difference between the experimental and all three predicted lift coefficients were 13% or more. Besides comparing predicted and experimental absolute value differences, it is also very important to notice that the NWT predicted numerical coefficients had the same behaviour as the experimental coefficients. Lift coefficients in both cases reached their maximum value near between AOA of 13 and 14 degrees, and both started to decrease rapidly about 13% after stall. As a result, the NWT capable of simulating airfoil stalls with reasonable accuracy.

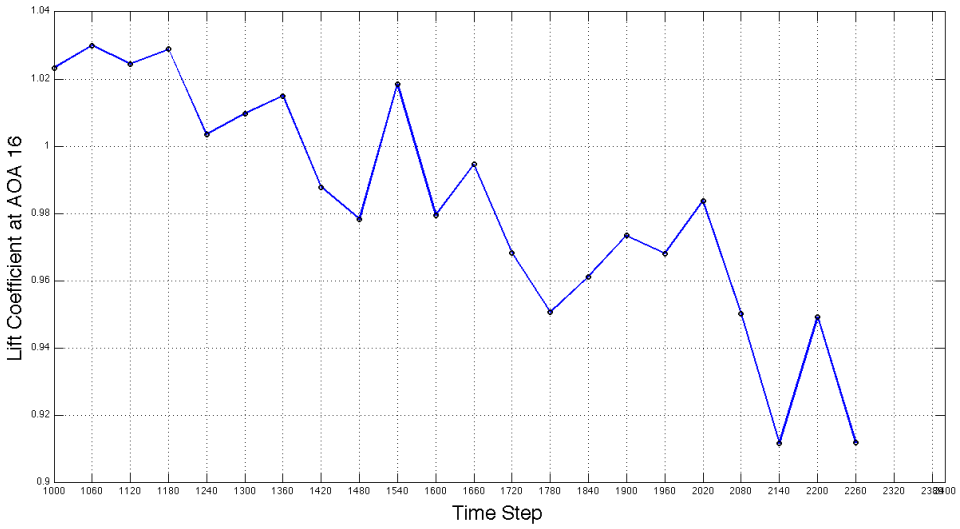

Figure 25 Lift coefficients of NACA 0009 fixed at 16 degrees of AOA

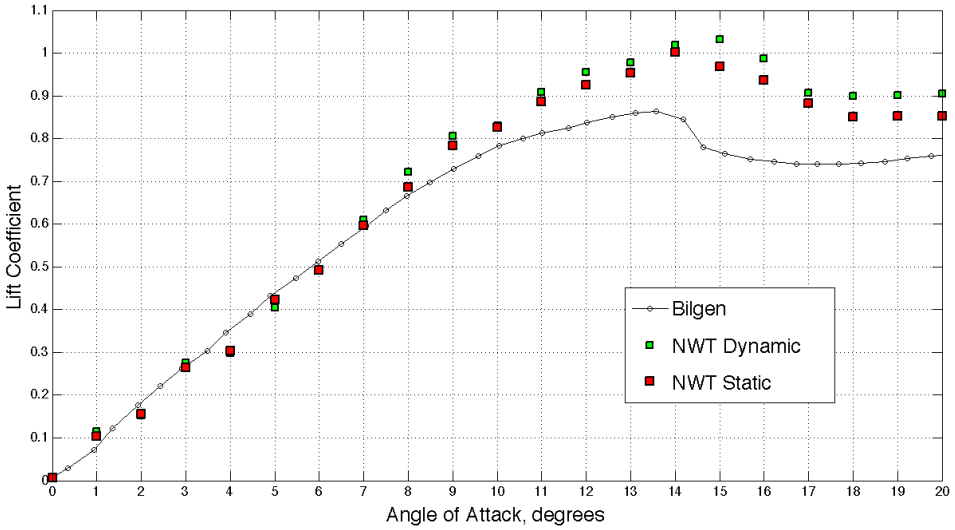

Figure 26 Static Lift coefficients of NACA 0009 from AOA of 10 to 20 degrees compared with dynamic and experimental lift coefficients

Table 6 Difference study between static and dynamic lift coefficients

AOA	Experimental	Dynamic	PDLD	Static	PDLS
	Cle	Cld	(Cld-Cle)/Cle	Cls	(Cls-Cle)/Cle
0	0.009	0.0059	-34.4%	0.0059	-34.4%
1	0.081	0.1145	41.4%	0.1045	29.0%
2	0.181	0.1522	-15.9%	0.1552	-14.3%
3	0.269	0.275	2.2%	0.2651	-1.4%
4	0.358	0.2984	-16.6%	0.3032	-15.3%
5	0.44	0.4036	-8.3%	0.4231	-3.8%
6	0.519	0.4927	-5.1%	0.4927	-5.1%
7	0.591	0.609	3.0%	0.5976	1.1%
8	0.668	0.7209	7.9%	0.6855	2.6%
9	0.729	0.8056	10.5%	0.7843	7.6%
10	0.781	0.83	6.3%	0.826	5.8%
11	0.813	0.9077	11.6%	0.8864	9.0%
12	0.838	0.9561	14.1%	0.9261	10.5%
13	0.859	0.978	13.9%	0.953	10.9%
14	0.85	1.0187	19.8%	1.0022	17.9%
15	0.769	1.0313	34.1%	0.9683	25.9%
16	0.749	0.9863	31.7%	0.9363	25.0%
17	0.74	0.9076	22.6%	0.8827	19.3%
18	0.742	0.8999	21.3%	0.8501	14.6%
19	0.75	0.9005	20.1%	0.8519	13.6%
20	0.76	0.9053	19.1%	0.8529	12.2%

The simulated static lift coefficients are, on average, 14% higher than the experimental lift coefficients at high AOA. One theory for the cause of this situation is that the flow separation in the simulation occurred much later and is smaller than for the flow in the experiment. The theory coincides with a Davison's [30] argument in his report. Davison claims the B-L model tends to over predict the Reynolds stress in the shear layer, and predicts the separation point too late, resulting in a separation zone that is too small, so the flow tends to stay attached to the upper surface a little longer. Since NWT uses the Baldwin-Lomax-Smagorinsky hybrid model, which combines Baldwin-Lomax mixing-length model for the boundary-layer region and the classic Smagorinsky LES model for flow outside the boundary layer as such it is expected to suffer from the same limitations as B-L model for the prediction of the point and length of the separation.

Another noticeable difference between the numerical and experimental data is the computation of flow reattachment post stall. As shown in Figure 26, experimental lift coefficients after AOA of 15 degrees drop back to the same lift coefficient as AOA of 9 degrees due to flow reattachment, which is due to flow reattachment on the upper surface after stall. But simulated lift coefficients do not show such characteristic, and the reattachment is not as significant. As Ekaterinaris and Platzer pointed out this issue in [15], current turbulence models show serious deficiencies when applied to flow reattachment process post dynamic stall.

4.2.4 Validation of the Numerical Simulation

In order to check the validity of the airfoil simulation, it is a common practice to calculate the pressure coefficients of the upper and lower surface. When the NACA 0009 airfoil is placed in the flow at AOA of 0 degree, the upper and lower surface pressure coefficients should be equal to each other because the airfoil is symmetrical. As shown in Figure 27, the pressure coefficients of upper and lower surface share the same profile. At 0 degree, the leading edge and trailing edge of the airfoil generates the largest pressure, and the surface near the thickest point of the airfoil presents the lowest pressure. When the leading edge of an airfoil is tilted upwards, the lower surface of the airfoil generates

much larger pressure than the upper surface as pressure coefficients at AOA of 18 degrees shown in Figure 28, so the pressure difference contributes to a positive lift force on the airfoil.

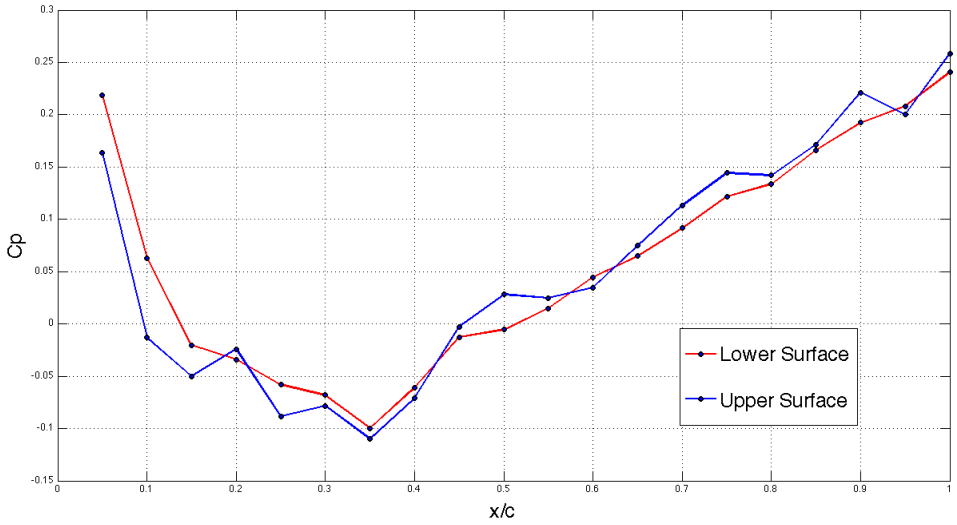

Figure 27 Pressure coefficients of upper and lower surface at AOA 0 degrees

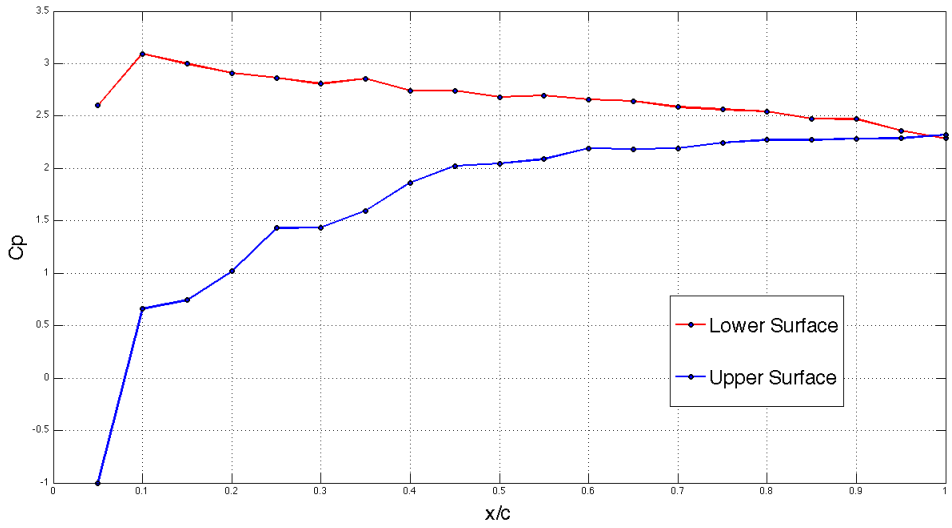

Figure 28 Pressure coefficients of upper and lower surface at AOA 18 degrees

To further validate the accuracy of the simulated flow distribution, the rest of the section is going to study the three-dimensional characteristics of the flow. Figure 29 is an iso-

surface visualization of the airfoil at AOA of 17 degrees. Although the graph looks a little chaotic at first glance, the graph clearly shows that the flow follows a similar span-wise pattern, which indicates that the flow across the span of the airfoil moves downstream at a uniform pace and frequency.

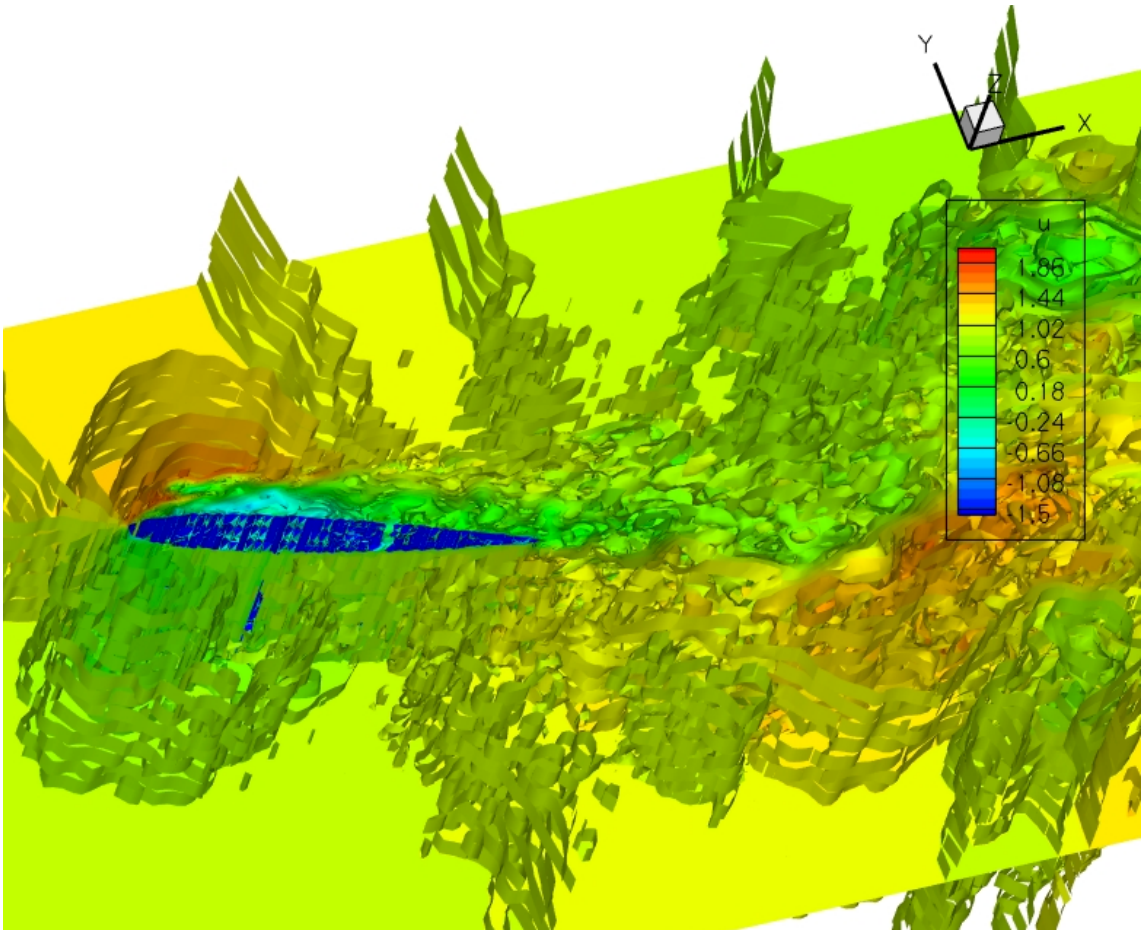

Figure 29 ISO-Surface -2.0 to +2.0 view of span-wise velocity

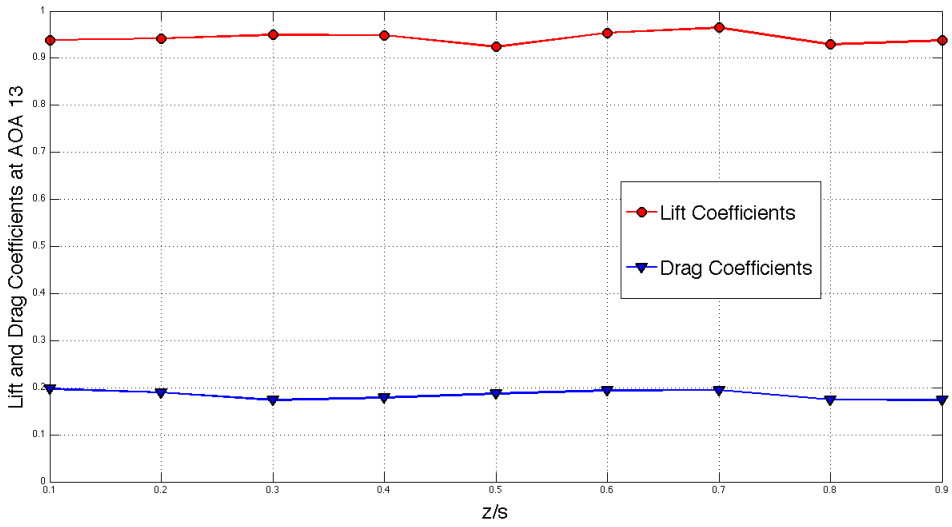

Figure 30 Span-wise lift and drag coefficients at AOA of 13 degrees

In order to verify the span-wise correlation of the three-dimensional flow, the airfoil is divided into nine equal width mini-airfoils, then, lift and drag coefficients were calculated separately for each divided piece. This test is able to directly compare the span-wise flow correlation. Figure 30 shows lift and drag coefficients across the airfoil span at AOA of 13 degrees. Both lift and drag coefficients fluctuate very little in the graph. Their average values are calculated to be 0.98 and 0.2 respectively, which equal to lift and drag coefficients plotted in Figure 23 and 24.

CHAPTER 5 CONCLUSION

This chapter concludes the thesis by giving a brief summary of achievements and several recommendations for future work.

5.1 CONCLUSIONS

The goal of the research is develop a software package based on NWT that's capable of simulating turbulent flows around aerodynamic shaped geometries such as an airfoil. The research started by conducting the first airfoil simulation using a NACA 0009 airfoil, but the first simulation was unable to produce reasonable lift and drag coefficients of the airfoil. The unsuccessful first test results were analyzed. Many improvements and new features were made to the NWT. Ultimately, the research achieved its goal by successfully predicting stall conditions using numerical simulated lift and drag coefficients. The achievements are summarised below:

- **DES method implementation:** The NWT was implemented using DES turbulent method. Since DES is a combination of LES and B-L model, the transition point between two models is vital to the accuracy of the DES method. The transition point is defined by a dimensionless value $y+$. Modifications were made to the $y+$ calculation method to ensure a better and smoother transition between LES and B-L layers.
- **Surrounding Cell Method:** SCM is a key feature of NWT to track Cartesian grid locations surrounding the IB. The feature was implemented to only work with cylinder geometries in NWT, so the existing routines was modified to support aerodynamic shaped geometries including airfoils.
- **Lift and drag coefficients prediction:** In order to accurately simulate flow around an airfoil, the NWT must be able to calculate lift and drag coefficients correctly, hence a new algorithm capable of calculating lift and drag coefficients is created.

- **Mesh refinement and adaptation criteria methods:** The IB cells generated by NWT's automatic mesh adaptation method were not refined enough for the requirements of DES sublayer. New methods were developed to specifically solve this issue by custom controlling mesh refinement near the IB without significantly increase total mesh size.
- **Stall prediction:** The NWT simulated lift and drag coefficients showed impressive agreement with experimental lift and drag coefficients data. Static stall of the NACA 0009 was successfully predicted using the lift and drag coefficients simulated by the NWT. In addition, dynamic stall condition is also observed from the moving airfoil simulation.

5.2 RECOMMENDATIONS

The results produced by NWT were mostly satisfactory and they proved the viability of simulating flow around aerodynamic shaped geometries and calculating lift and drag coefficients. This work also provided a foundation for future research in airfoil stall prediction methods and directions for turbulence modeling improvements.

- **Dynamic stall study:** The NWT in current form is still lacking features such as oscillation frequencies settings to accurately predict 3D dynamic stall effect of airfoils. The most difficult type of dynamic stall study is the simulation of 3D airfoil tip region. This type of flow involves intensive flow mixing, separation, and flow transition between turbulent and laminar flows even at relatively high Reynolds numbers. Better transitional and turbulent models are required to predict this type of flow.
- **DES model improvements:** The DES model used in the NWT is a hybrid model of B-L and LES methods. Although the method is proven to be able to accurately simulate turbulent flows around airfoil, it is still inadequate to accurately simulate flow separation and reattachment after dynamic stall condition. The B-L model could be the cause of this issue. The NWT implementation could be improved by exploring other viable models.

- **Continuously moving geometry algorithm:** The current NWT implementation is able to simulate moving geometries using a discrete approach, which is discussed in detail in section 3.1. Because the same geometry is always initially loaded into the program, it should be easy enough to implement a new routine that automatically change the geometry location at a predefined frequency. This algorithm should be able to improve the transition time between each geometry change such as AOA increase, and makes the geometry transition smoother.

BIBLIOGRAPHY

[1] Lesieur, M. (2007). Turbulence in fluids. Dordrecht: Springer.

[2] Versteeg, H. K., & Malalasekera, W. (2007). An introduction to computational fluid dynamics: The finite volume method, Second edition. Harlow, Essex, England: New York.

[3] Wilcox, D. C. (1993). Turbulence modeling for CFD. La Cãnada, CA: DCW Industries, Inc.

[4] Lumley, J. L. (January 01, 1998). Review of Turbulence in Fluids - Third Revised and Enlarged Edition. Aiaa Journal, 36, 9, 1768.

[5] Lilly, D.K. (1966). On the application of the eddy viscosity concept in the Inertial sub-range of turbulence. NCAR Manuscript 123, Boulder, Colorado.

[6] Lilly, D.K. (1967). The Representation of Small-Scale Turbulence in Numerical Simulation Experiments. In Proceedings of the IBM Scientific Computing Symposium on Environmental Sciences Yorktown Heights, New York.

[7] Hughes, T. J. R., Mazzei, L., & Jansen, K. E. (January 01, 2000). Large Eddy Simulation and the variational multiscale method. Computing and Visualization in Science, 3, 47-60.

[8] Hughes, T. J. R., Mazzei, L., Oberai, A. A., & Wray, A. A. (January 01, 2001). The multiscale formulation of large eddy simulation: Decay of homogeneous isotropic turbulence. Physics of Fluids, 13, 2, 505.

[9] Leslie, D. C. and Quarini, G. L. (1979). The Application of Turbulence Theory to the Formulation of Subgrid Modelling Procedures, F. Fluid Mech., Vol. 91, pp. 65-91.

[10] Bardina, J., Ferziger, J. H. and Reynolds, W. C. (1980). Improved Subgrid-scale Models for Large-eddy Simulation, AIAA Paper 80-1357.

[11] Germano, M(1986). A proposal for a Redefinition of the Turbulent Stresses in the Filtered Navier-Stokes Equations, Phys. Fluids, Vol. 29, pp. 2323-2324.

[12] Moin, P. (2002). Advances in Large Eddy Simulation Methodology for Complex Flows, Int. F. Heat Fluid Flow, Vol. 23, pp. 710-720.

[13] Cebeci, T. (2005). Analysis of low-speed unsteady airfoil flows. Long Beach, Calif: Horizons Pub.

[14] CARR, L. A. W. R. E. N. C. E. W. (January 01, 1988). Progress in analysis and prediction of dynamic stall. Journal of Aircraft, 25, 1, 6-17.

[15] Ekaterinaris, J. A., & Platzer, M. F. (January 01, 1997). Computational prediction of airfoil dynamic stall. Progress in Aerospace Sciences, 33, 759-846.

[16] Ericsson, L. E., & Reding, J. P. (January 01, 1987). Fluid dynamics of unsteady separated flow. Part II. Lifting surfaces. Progress in Aerospace Sciences, 24, 4, 249-356.

[17] Schmidt, S., & Thiele, F. (January 01, 2003). Detached Eddy Simulation of Flow around A-Airfoil. Flow, Turbulence and Combustion, 71, 1-4.

[18] Spalart, P. R. (August 1997). Comments on the feasibility of LES for wing and on a hybrid RANS/LES approach. 1st ASOSR CONFERENCE on DNS/LES. Arlington, TX.

[19] Weber, C., & Ducros, F. (January 01, 2000). Large Eddy and Reynolds-Averaged Navier-Stokes Simulations of Turbulent Flow Over an Airfoil. International Journal of Computational Fluid Dynamics, 13, 327-356.

[20] Camelli, F. E., & Lohner, R. (January 2002). Combining the Baldwin Lomax and Smagorinsky turbulence models to calculate flows with separation regions, AIAA Paper 2002-0426, Reno, NV.

[21] Schetz, J. A. (2010). Boundary layer analysis. Reston, Va: American Institute of Aeronautics and Aeronautics.

[22] Kundu, P. K., & Cohen, I. M. (2008). Fluid mechanics. Amsterdam: Academic Press.

[23] Kim, D., & Choi, H. (January 01, 2000). A Second-Order Time-Accurate Finite Volume Method for Unsteady Incompressible Flow on Hybrid Unstructured Grids. Journal of Computational Physics, 162, 2, 411-428.

[24] Bilgen, O. (2010). Aerodynamic and electromechanical design, modeling and implementation of piezocomposite airfoils. Blacksburg, Va: University Libraries, Virginia Polytechnic Institute and State University.

[25] Selig, M. S., Donovan, J. F., & Fraser, D. B. (1989). Airfoils at low speeds. Virginia Beach, Va., USA (1504 Horseshoe Cir., Virginia Beach 23451: H.A. Stokely.

[26] Selig, M. S., Guglielmo, J.J., Broeren, A.P., and Giguere, P. (1995). Summary of low speed airfoil data. Virginia Beach, Va: SoarTech Publications.

[27] Bell, T., & Dalhousie University. (2003). The numerical wind tunnel: A three-dimensional computational fluid dynamics tool. Halifax, N.S: Dalhousie University.

[28] Hicken, J., & Dalhousie University. (2004). Turbulence modelling on an unstructured, time-adaptive mesh. Halifax, N.S: Dalhousie University.

[29] Ham, F., Lien, F., & Strong, A. (January 01, 2002). A Cartesian Grid Method with Transient Anisotropic Adaptation. Journal of Computational Physics, 179, 2, 469-494.

[30] Davision, L. (1991). Predicting Stall of a Two-dimensional Airfoil Using an Algebraic Reynolds Stress Model, Cerfacs France Report TR/RF/91/52, 1991.

[31] Martinat, G., Braza, M., Harran, G., & Hoarau, Y. (November 01, 2008). Turbulence modelling of the flow past a pitching NACA0012 airfoil at 10^5 and 10^6 Reynolds numbers. Journal of Fluids and Structures, 24, 8, 1294-1303.

[32] Tucker, P. G., & Liu, Y. (October 01, 2007). Turbulence modeling for flows around convex features giving rapid eddy distortion. International Journal of Heat and Fluid Flow, 28, 5, 1073-1091.

[33] Moin, P., & Kim, J. (January 01, 1997). Tackling Turbulence with Supercomputers. Scientific American, 276, 1, 62-68.

[34] Meinke, M., & Krause, E. (January 01, 1998). Application of LES to Jets and Internal Turbulent Flows. Courses and Lectures International Centre for Mechanical Sciences, 395, 155-208.

[35] Douvi, C. E. G. G. (March 01, 2012). Evaluation of the turbulence models for the simulation of the flow over a National Advisory Committee for Aeronautics (NACA) 0012 airfoil. Journal of Mechanical Engineering Research, 4, 3.)

[36] Piziali, R. A. (1993) An experimental investigation of 2D and 3D oscillating wing aerodynamics for a range of angle of attack including stall. NASA Technical Memorandum 4632.

[37] Murrin, C. D. (May 29, 2007). A three-dimensional simulation of Vortex Induced Vibrations (VIV) on marine risers at high Reynolds number using computational fluid dynamics. St. John's, Newfoundland: Memorial University of Newfoundland.