A RAPID DEVELOPMENT PROCESS FOR MARINE PROPELLERS THROUGH DESIGN, SIMULATION AND PROTOTYPING

XUEYIN WU

A Rapid Development Process for Marine Propellers Through Design, Simulation and Prototyping

by

© Xueyin Wu

A thesis submitted to the

School of Graduate Studies

in partial fulfilment of the

requirements for the degree of

Master of Engineering

Faculty of Engineering and Applied Science

Memorial University of Newfoundland

October 2010 Submitted

St. John's Newfoundland

Abstract

Flexibility and speed to market are the keys to successful product development. For marine propellers, these goals are achieved though iteration of design software and prototyping. In this thesis, an expanded propeller design code, "OpenPVL_SW", which was developed based on the Open propeller vortex lattice lifting line code (OpenPVL), was improved to include propeller geometry generation into SolidWorks, thrust simulation with CosmosFloWorks and strength assessment with CosmosWorks. The new code OpenPVL_SW is described in this thesis. In the OpenPVL_SW, a parametric design technique and a single propeller geometry generator are completed in MATLAB, and a propeller blade geometry file for SolidWorks is created after running the design program.

The purpose of this study is to extend the use of the original OpenPVL code not only for the propeller design, but also to achieve the thrust simulation and strength check using SolidWorks tool package, CosmosFloWorks and CosmosWorks. A propeller geometry, which is provided by Oceanic Consulting Corporation, is simulated to predict propeller thrust using CosmosFloWorks. A case study designed an AUV propeller and simulated the thrust with CosmosFloWorks to prove that the OpenPVL_SW code provides perfect propeller geometry and a reasonable simulation result of thrust.

Acknowledgements

First and foremost, appreciation and thanks are bestowed upon Prof. Andy Fisher for his guidance, support and advice. From the beginning stages to the submission of this thesis, he supported and kept me on the path towards completion, and I did learn a lot from him. I am greatly indebted.

I expresses thanks to Matthew Garvin, manager of design and fabrication in Oceanic Consulting Corporation, for providing propeller geometry, experimental results and guidance.

I would like to extend thanks to Prof. Mahmoud Haddara. I learned a lot from him about propeller design and testing.

I would like to acknowledge the support of the people in the rapid prototyping lab for their support during the propeller fabrication using Stratasys Fusion Deposition Modeling (FDM).

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In Newfoundland and Labrador, ocean technology is an important sector with many public and private firms involved in designing new or more efficient technologies. One of the main components of many of these developments is the marine propeller. Designing high efficiency and low cost propellers for ships, AUV's, ROV's and other marine applications is therefore a critical research topic. At the same time, the technologies of Computer-Aided Design (CAD) and Computer-Aided Manufacturing (CAM) have been rapidly advancing, making these tools much more accessible. More and more designs and fabrication processes are using computer technologies to achieve high accuracy and shortened cycle times for manufacture and testing. In this thesis, the focus is on a rapid development process for marine propeller design and prototype fabrications using these computer technologies.

Propellers transmit power by converting rotational motion into thrust. Traditionally, engineers needed to do a lot of analysis to determine the foil sections, pitch, blade angle and so on, in order to design and produce a suitable propeller. The process took engineers a long time to analyze, draw, fabricate and test. Nowadays, due to rapid advances in the development of computer technologies, most engineers are using computer programs to assist in the design of propellers, instead of using traditional methods. After designing a propeller in a computer, rapid prototyping technologies provide a very efficient way to fabricate a prototype propeller. The material that can be used in the rapid prototyping technologies is not perfect for the actual propeller, however, using rapid prototyping machines, prototype propellers can be produced quickly facilitating early testing. If the results show the propeller is not performing as desired, the propeller can be redesigned, reproduced and retested. Saving time is the main advantage of the rapid prototyping technologies.

The general processes of propeller design and fabrication is to first input the propeller parameters, generate a preliminary geometry and then to run a computer program to do analysis. If the analysis results show that the propeller is suitable then the propeller geometry outputs will be used for propeller fabrication and testing; if results show the propeller is not desired, engineers will change propeller parameters to redesign it. Figure 1 shows a flow chart of the proposed method of design and propeller fabrication that will be described in this thesis.

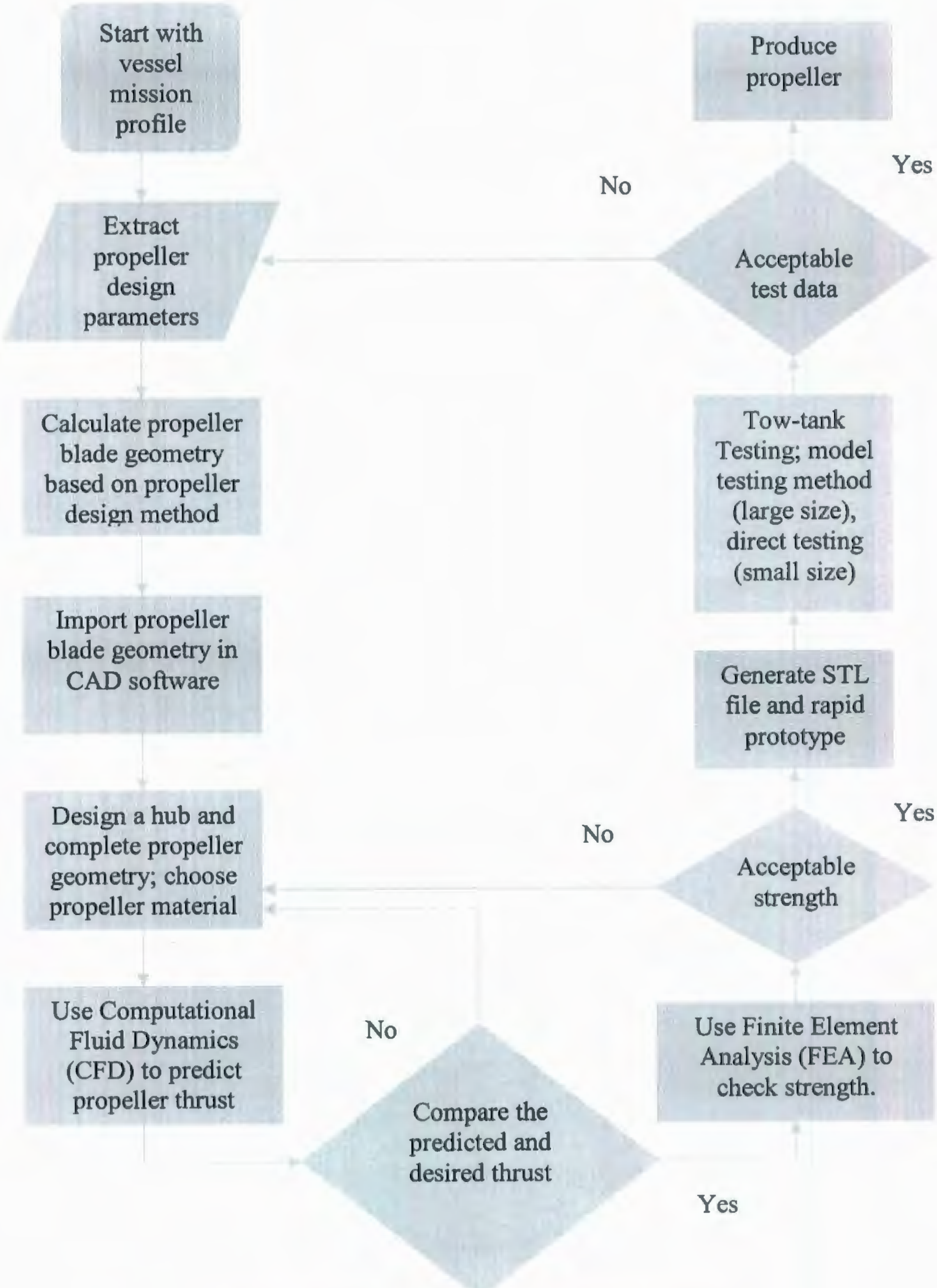

Figure 1: Flow Chart of Propeller Development

Simulation work is now commonly done before a propeller is fabricated. If an integrated software tool can include analysis, design, fabrication geometry and simulation, the design process will be much more convenient for developers.

In this thesis, the author is expanding an open source propeller design code to OpenPVL_SW code that is combined with SolidWork software, to analyze, design and fabricate propellers, and also to see how the propeller design can be simulated. A case study of an autonomous underwater vehicle (AUV) propeller completed by the OpenPVL_SW code will be presented in this thesis. In this case, the OpenPVL_SW code is used to create an AUV propeller and generate data for propeller geometry and simulation. Second case study used CosmosFloworks to simulate the thrust of the propeller where the geometry was provided by Oceanic Consulting Corporation. Compared with the real testing result of the thrust, this case study is used to prove that CosmosFloworks can provide a reasonable thrust prediction. After the thrust simulation, CosmosWorks is used to check the strength of the propeller deisgn using Finite Element Analysis (FEA). If the strength result is suitable, then the propeller geometry is ready to be fabricated; if the strength result is not desired, engineers can use higher strength material or redesign the propeller geometry to increase strength.

Chapter 2

Literature and Software Review

This chapter will review the literature concerning propeller design methods, recent propeller design codes, Computer-Aided Design (CAD) and Computer-Aided Manufacturing (CAM) for propellers and also provide a review of OpenPVL.

2.1 Review of Propeller Design Methods

The development of propeller design methods has evolved for a long time. When propellers were first used for propulsion, very few people knew how propellers operated and the optimal way to design them. Early designs were following the steps: trial, error and imagination. Momentum theory was applied to propellers in the late nineteenth century. It explained the resulting thrust of propellers, but it did not give a detailed reason for these results [1]. Design for blade strength was based on experience and then later on simple beam theory to determine the minimum root thickness and thickness distribution along the blade span. Nowadays, finite element analysis is commonly used to analyze propeller blades in details to ensure structural soundness.

Early theoretical analysis applied to propellers was based on momentum theory. In 1910, Betz was the first to formally formulate the circulation theory of aircraft wings for using with screw propellers [2]. The Vortex theory of propellers was developed with the basic assumption that certain geometric qualities of the flow in which a propeller operates must exist if the energy losses are to be minimized. In 1929, Goldstein developed this theory, showing that the flow past a vortex sheet could be calculated by relating the two theoretical cases of a propeller with a finite number of blades and a propeller with an infinite number of blades [3]. Goldstein formulated Betz's vortex theory for propellers for real cases of propellers with a finite number of blades. He showed that the velocity characteristics changed substantially by removing the assumption of the infinite blade number. His analysis was done for the case of optimum circulation distribution of a propeller with minimum energy loss. By calculating the ratio of the circulation with infinite and finite number of blades, the Goldstein coefficient, κ, was derived for periodic flow [3].

In 1947, Glauert published that the individual airfoil sections along the blade span could be included directly into a lifting line calculation to design a propeller owing to certain characteristics of lift and drag. He used momentum theory to determine the average induced velocity at the lifting line [4].

In 1952, Lerbs simplified the calculations by introducing the concept of induction factors [5]. The induction factors depend only on helix geometry and therefore can be calculated dependent of loading. To achieve this, Lerbs made two assumptions. One, that the radial induced velocity is assumed to be negligible in the cases of light to moderate loading; the other, that the calculated hydrodynamic pitch approximates the shape of the streamlines in the wake. These are assumed to depend only on the axial and tangential velocity components [5]. Lerbs extended Glauert's methods to calculate element contributions to thrust and torque along the blade span. The results were integrated over the propeller blades to give overall performance of the propellers. He effectively produced the first practical numerical propeller design method applied under various non-optimum conditions of loading [5]. Lerbs' method became known as the lifting line method for marine propellers, because the force produced by the circulation is assumed to act along radial lines in place of the blades, similar to the theory of airscrews [6].

In 1955, Eckhardt and Morgan developed an engineering approach to the lifting line method. They assumed an optimum distribution of circulation, and a reduced thrust, proportional to the Goldstein factor due to the number of blades, and proceeded to calculate induced velocities. This greatly reduced the amount of calculations. They also showed that their method produced only small errors under light and moderate loading conditions and therefore it is a very useful design tool [7].

In 1976, panel method was initially developed as lower-order method for incompressible and subsonic flows [8]. In 1985, Hess and Valarezo introduced the panel method for marine propellers that allowed them to vary the pitch values and distributions and take into account the inflow wake distribution and cavitation effects [9]. The panel method provides an elegant methodology to solve a class of flow past arbitrarily shaped bodies in both two and three dimensions. The basic idea is to discretize the body in terms of a singularity distribution on the body surface, to satisfy the necessary boundary conditions, and to find the resulting distribution of singularity on the surface, thus obtain fluid dynamic properties of the flow [10].

Nowadays, engineers are seeking ever faster ways to design propellers for customers' requirements and enhancing speed to market. The lifting line method and panel method are very useful and commonly used to design conventional propellers. The required input data for lifting line method and panel method is shown in table 1.

Table 1: Required Input Data for Lifting Line Method and Panel Method [11][12]

Input data for lifting line method		**Input data for panel method**
Number of blades Propeller speed Propeller diameter Required thrust Ship velocity Fluid density	equivalent	Number of blades Propeller speed Propeller diameter Required thrust Ship velocity Fluid density
Maximum iterations in wake alignment		Effective power
Hub vortex radius/hub radius		Ship advance coefficient
Shaft centerline depth		Wake fraction
Inflow variation		Thrust deduction fraction
Ideal angle of attack		Propeller advance coefficient
Hub diameter		Required resistance
		Propeller advance speed
		Required thrust coefficient
		Required torque
		Required torque coefficient
		Required propeller efficiency
		Delivered power (KW)

As can be seen in Table 1, the lifting line method does not require as much detail as the panel method, thus it can be obtained more easily and can still yield useful results [11]. By using the lifting line method, large advances in the design of marine propellers can be made without much more detailed numerical input required. The lifting line method is very useful for conventional propeller design and is used extensively by leading propeller manufacturers [6]. By using this method, the circulation distribution and hydrodynamic pitch can be calculated, and the required pitch distribution can be constant or varied. Blade cross-sectional characteristics are incorporated into the calculations by using detailed airfoil section data and lift-drag relationships. A strength analysis can also be included in the method by incorporating simple beam theory directly into the calculation to obtain the primary stresses [13] [14], which include bending moments due to thrust and torque. Equation 1 is the bending moment at the section due to the thrust on the blade. Equation 2 is the bending moment at the section due to the torque on the blade [15].

$$M_T = \int_{r_0}^{R} \frac{1}{Z} \frac{dT}{dr} (r - r_0) dr \qquad 1$$

$$M_Q = \int_{r_0}^{R} \frac{1}{rZ} \frac{dQ}{dr} (r - r_0) dr \qquad 2$$

T is thrust; Q is torque; R is the propeller radius; r_0 is hub radius; Z is the number of blades.

2.2 Review of Recent Propeller Design Software

Based on many years of development, propeller design methods have been enhanced. Due to the rapid development of computer technology, propeller designers are tending to use computer code more often to design propellers, and are designing much more complicated and multifunctional codes. In the early stages of propeller design software, the code was used for specific functions. For example, the WAOPTPROP code [16] was only used for propeller geometry, and the PPT2 code [17] was only used for propeller analysis. Later, computer code, such as the PVL code, was developed to design the propeller's geometry and simultaneously analyze the performance [17]. Subsequently, engineers wanted to combine computer code with CAD software to automatically finish the propeller's fabrication. For example, D'Epagnier created an OpenPVL code [11] not only for analysis and design, but also to create scripting for 3D printable files using the CAD software RHINO, which generates a .STL file that is used to build propellers by rapid prototyping machines.

In 1991, Hofmann wrote the WAOPTPROP code using VAX FORTRAN based on lifting line method [16]. The program calculated the induced velocities at the blade sections from a non-optimum circulation distribution, the required pitch distribution, and the thrust and torque coefficients for the design condition. Propeller design used these program results to determine the hydrodynamic pitch distribution by comparing the design point coefficient with the pitch distribution. WAOPTPROP can also provide information about final propeller geometry. PPT2 was another program written by Hofmann [16] for the analysis of propeller performance. Finally, WAOPTPROP combined with PPT2 can work as a fully integrated propeller design program. However, this design program has two main disadvantages: it needs two separate programs to achieve the design target, whereas, it could be more convenient to use one program. Another disadvantage is that the final propeller geometry has to be entered by hand into CAD software, which is troublesome when the designer has many priorities.

PVL code, was created by Professor J. Kerwin at MIT in 2001 using Fortran programming [17]. It was translated into MATLAB as an open source MPVL code released by Hsin-Lung in May 2007. MPVL code was based on lifting line method, and its optimization algorithm was based on Lerb's criteria [18]. MPVL code, working with the high-level technical computing language MATLAB, can be easily modified by users according to their specific needs; propeller designers are able to conduct both propeller analysis and single propeller design functions of the program. Compared with Hofmann's propeller design programs, MPVL code has the advantages of integrating design and analysis programming. However, there is no way to connect the propeller design program directly with CAD software, which means that even by using this code, users have to create the propeller geometry in the CAD software by hand.

In September 2007, D'Epagnier [11] modified the MPVL code to create a new program called OpenPVL. It operated using an evolved MPVL code while expanding upon MPVL's applications, and had been modified to create scripting for 3D printable files using a CAD interface to the commercial CAD software "RHINO". OpenPVL made up for the disadvantages of MPVL by connecting the program to CAD software. In OpenPVL, propeller blade geometry can be generated and imported into RHINO, and then saved. Once in RHINO, various CAD manipulations are possible, including the export of stereolithography (.STL) file, which can be used in a rapid prototyping (RP) machine to produce propeller prototypes.

2.3 Review of OpenPVL

OpenPVL is an open source for marine propeller design. There are two main components of OpenPVL. One is parametric analysis, which is used to combine all the propeller's parameters to analyze the efficiency and optimize the design. The other component is the propeller design function to generate files of propeller inputs, outputs, geometry and performance. The CAD software RHINO opens the geometry file and can from this one use standard CAD commands to create a propeller blade, design a hub and add other blades to complete the propeller.

2.3.1 Parametric Analysis

The number of blades, the propeller speed and the propeller diameter work as the three foundational parameters for propeller design. These three are combined with other additional parameters, including required thrust, ship speed, hub diameter, number of vortex panels over the radius, maximum number of iterations in wake alignment, ratio of hub vortex radius to hub radius, hub and tip unloading factor, swirl cancellation factor, water density and hub image flag, as input parameters to the analysis process.

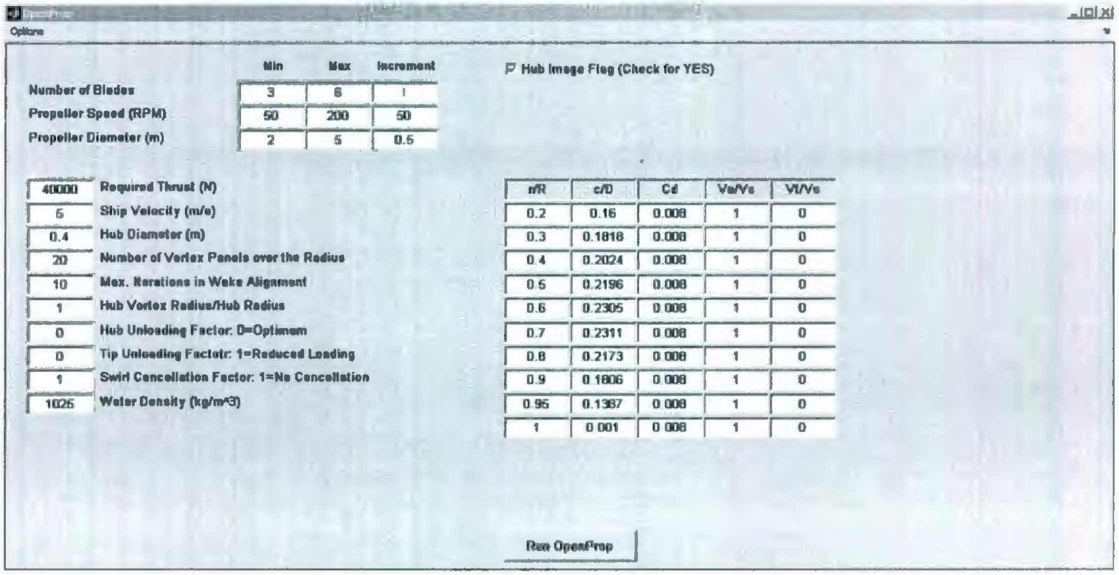

Figure 2: Parametric Analysis Matlab Interface

These input parameters are introduced in the sections below. Figure 2 shows the parametric analysis, which includes the user input fields required to run the analysis.

- Number of blades: The range of the number of blades is from two to six. Propellers normally have a number of blades within this range. More number of blades will increase thrust; however, it may cause cavitaion. Less number of blades will avoid cavitaion; however, it decreases thrust.

- Propeller speed: The unit of propeller speed is revolutions per minute (RPM). The restrictions are that the value must be positive, and the maximum value must be greater than or equal to the minimum value. The increment cannot be negative. High RPM will increase thrust; however it may cause cavitation. Low RPM will avoid cavitation; however, it decreases thrust. The range of RPM cannot exceed the rotational range by vessel engine.

- Propeller diameter: The unit of propeller diameter is meters (m). The value of propeller diameter must be positive, and the maximum value must be greater than or equal to the minimum value. The increment cannot be negative. The propeller diameter must be greater than the hub diameter. Large propeller diameter increases thrust, and small diameter will decrease thrust. The propeller diameter is limited by the geometry of vessels.

- Required thrust: The unit of required thrust is Newtons (N). The value must be greater than 0, and cannot be negative. The required thrust is derived based on the required ship speed and resistant force when a vessel is moving.

- Ship speed: The unit of ship speed is meter per second (m/s). The value must be greater than 0, and cannot be negative.

- Hub diameter: The unit of hub diameter is meters (m). The hub diameter has to be greater than 15% of the propeller diameter [18].

- Number of vortex panel over the radius: This input field represents how many vortex panels will be divided into the blade and thus affects the resolution of the propeller blade. The number is greater than 0, and is an integer. Usually, twenty panels can provide sufficient resolution [18].

- Maximum number of iterations in wake alignment: This input field determines how many iterations Open_PVL is allowed to align the wake. The number is greater than 0, and is an integer. Ten iterations are usually sufficient for the program to converge and align in the wake [18].

- Ratio of the hub vortex radius to the hub radius: The hub drag was computed as a function of the ratio of vortex core radius to hub radius, but the precise value of this ratio is not critical [17]. For convenience, this ratio was usually assumed to be one [18].

- Hub and tip unloading factors: These two factors are defined as the fractional amount that the difference between the optimum values of tan β_i and tan β are reduced. If hub unloading factor is 0, tan β_i - tan β at the hub is retained at its optimum value from Betz/Lerbs criterion. If hub unloading factor is 1, tan β_i - tan β at the hub is set to zero, and the values up to the mid span of the blade are blended parabolically to the optimum value. The same procedure applies to the tip [17].

- Swirl cancellation factor: The value of this factor is zero or one. The swirl cancellation factor is zero for contra-rotation propellers in which the tangential velocities from each blade were cancelled [17]. If there was no swirl cancellation, the value of this factor was one [17].

- Water density: The water density depends on the users' preference. The unit is kg/m^3. The default value is 1025 kg/m^3.

- Hub image flag: this input field is located on the upper right side. By checking this option, hub image is present, and the circulation has a finite value at the hub. Un-checking this option will have a zero circulation at the root of a propeller blade [18].

On the right side of the parametric analysis screen, there are five editable parameters. r/R is the ratio of the radial location to the total length of the blade radius to be set from 20% of propeller radius (r/R=0.2) to the blade tip (r/R=1). c/D is the non-dimensional chord distribution over the radius. Cd is the drag coefficient over the radius. Vs is to the advance velocity over the radius. Va/Vs is the ratio of the axial inflow velocity to the advance velocity over the radius. The value of Va/Vs is set to one to assume uniform inflow. Vt/Vs is the ratio of the tangential inflow velocity to the advance velocity over the radius. The value of Vt/Vs is set to zero to assume uniform inflow. Open_PVL automatically interpolates these input fields in accordance with the hub radius.

After the calculation process is completed, the program also creates the efficiency curves according to the blade number and propeller diameter, shown in Figure 3. Ideally, a good propeller has a large diameter, slow speed, low number of blades and high efficiency. However, the real propeller parameters are always restricted in size and speed. It is the purpose of the efficiency curves combined with different propeller diameters and speed to help designers to determine the optimum parameters for a propeller design. To clearly show the design approach for a propeller, Figure 3 is taken as an example. Due to the limitation of a ship body geometry, the propeller diameter is restricted to three meters. The ideal number of blades is small; however, less number of blades perhaps causes propeller vibration due to the increased thrust on each blade. For this reason, a four-blade design is adopted for this design. From the efficiency curves with three-meter diameter and four-blade, the propeller has the highest efficiency at 100 RPM. Finally, the propeller is chosen with three-meter diameter, four blades and 100 RPM. Now, the three major propeller parameters have been determined. The detailed blade design is ready to be conducted next.

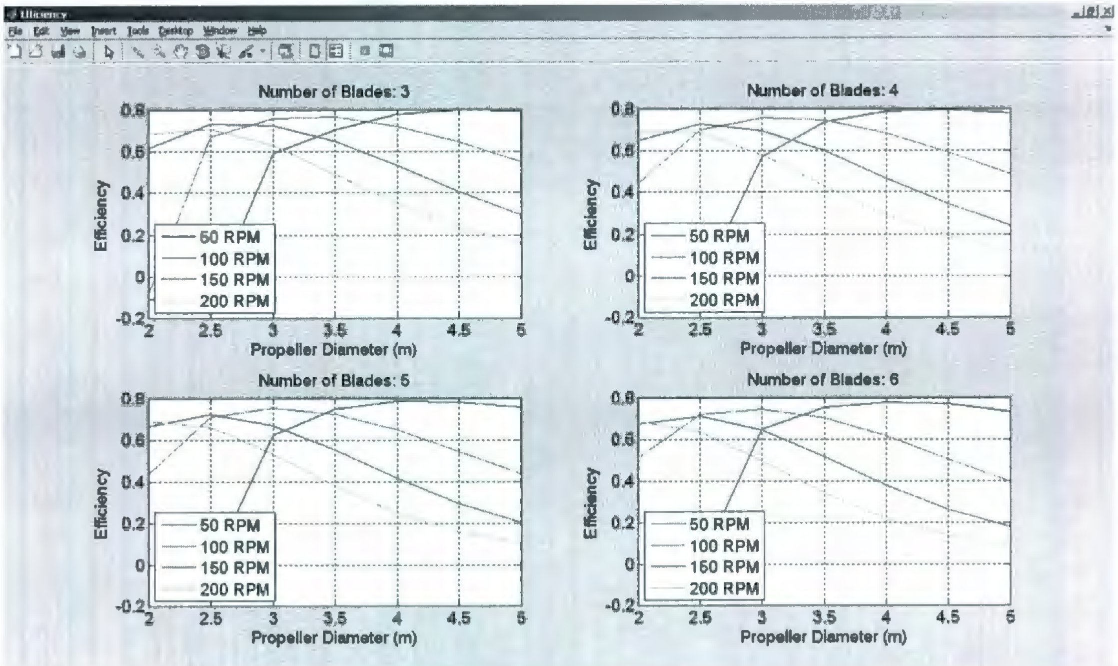

Figure 3: Efficiency Plot

2.3.2 Blade Design

After the parameters of a propeller with a viable efficiency curve have been established, the desired inputs are entered into the propeller design option, shown in Figure 4.

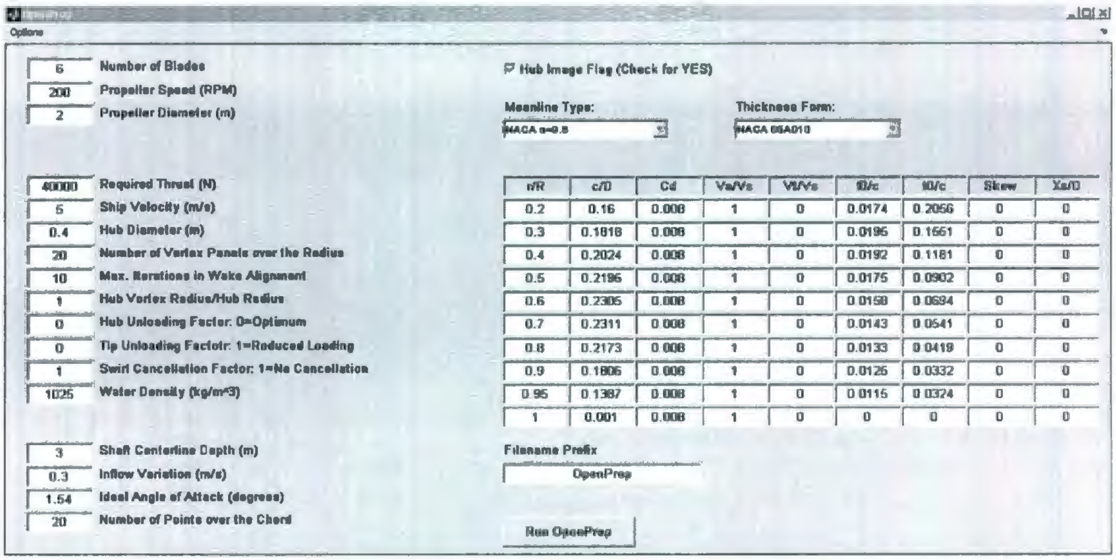

Figure 4: Propeller Design Matlab Interface

Several additional inputs are supported in the blade design function.

- Shaft centerline depth: The unit of this parameter is meters (m). It presents the depth when a propeller works.
- Inflow variation: It is required for the calculation of pitch angle variation, and the unit is m/s [17].
- Ideal angle of attack: This parameter is to calculate the pitch angle, and the unit is degree.
- The number of points over the chord: This parameter decides the resolution of a propeller blade. Usually, 20 points can provide sufficient resolution [18].

In the program, two types of meanline, which is a line drawn midway between the upper and lower surface, are available: the NACA (National Advisory Committee for Aeronautics) a=0.8 and the parabolic meanline. There are three types of thickness forms: NACA 65 A010, elliptical, and parabolic. The thickness form of NACA 65 A010 is designed to obtain high lift coefficient and high speed. The NACA thickness form is combied with the meanline a=0.8 in this OpenPVL code to construct propeller foil sections. The meanline of a=0.8 indicates that the pressure distribution on 80% of the foil chord is uniform [19]. The thickness form of elliptical is designed to reduce drag and obtain a thin blade with necessary strength. The thickness form of parabolic combined with the parabolic mean line is used to reduce resistance force and is applied for high-speed applications. Nowadays, the commonly used foil sections are the NACA foil sections, which includes some series of models, such as 4-digit series, 5-digit series, 16-series, 6-series, 7-series [19]. Each series has its own advantages and disadvantages. Table 2 displays the advantages and disadvantages of each NACA series model.

Table 2: Advantages and Disadvantages of Each NACA Series [19]

Series	Advantages	Disadvantages
4-Digit Series	Good stall characteristics; Roughness has little effect	Low maximum lift coefficient; high drag
5-Digit Series	High maximum lift coefficient; roughness has little effect	Poor stall characteristics; high drag
16-Series	Avoid low pressure peaks; low drag at high speed	Low lift
6-Series	High maximum lift coefficient; low drag in the operating conditions; optimized for high speed	High drag outside of the operating conditions
7-Series	Low drag in the operating conditions	Low lift coefficient; high drag outside of the operating conditions

The NACA 65 A010 foil section is the only NACA model in the OpenPVL code; however, there is the potential for users to add more foil sections in the OpenPVL code in the thickness form and meanline sections in the original OpenPVL code.

The parameters that are on the right side of the blade design screen are introduced as below.

- f_0/c and t_0/c: f_0/c is the maximum camber distribution. t_0/c is the maximum thickness distribution. The maximum camber f_0 and the maximum thickness t_0 are shown in Figure 5. c is the length of the nose-tail line, which is the dashed line in the Figure 6.
- Skew: The unit of skew is degree. This parameter is applied to reduce the propeller-induced unsteady forces [17].
- Xs/D: This parameter is the non-dimensional rake. It is applied to reduce the propeller induced vibration [17].

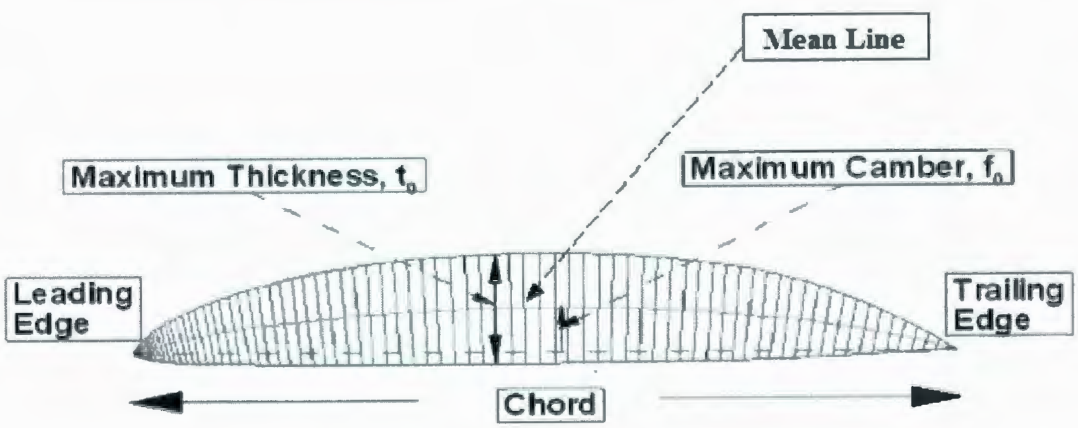

Figure 5. Foil Section Geometry [17]

After the calculation process, the graphical reports are created. In Figure 6, the upper left corner shows the non-dimensional circulation vs. the radial position, and the upper right shows the axial and tangential inflow velocities with the axial and tangential induced velocities vs. the radial position. The lower left corner shows the undisturbed flow angle and the hydrodynamic pitch angle vs. the radial position, and the lower right shows the chord distribution vs. the radial position.

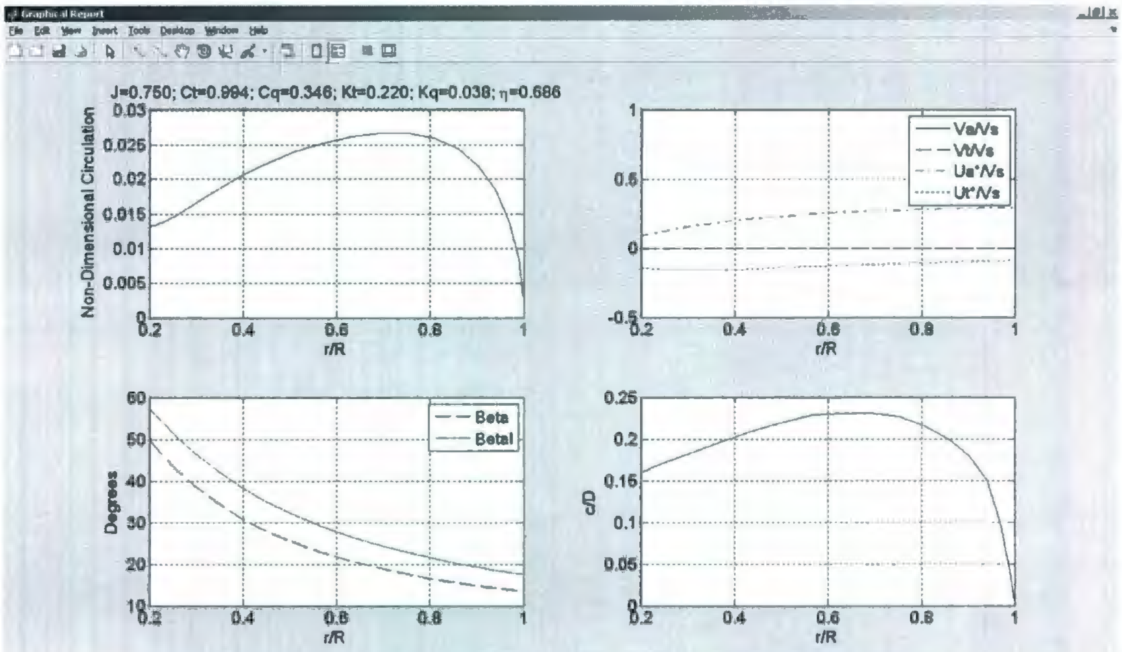

Figure 6: Graphical Report

Figure 7 presents five propeller blade profiles in a two dimensional view. This figure also displays the chord length, the pitch angle, the camber, and the thickness of the propeller foil section. Figure 8 shows a three-dimensional propeller image. It provides an instant graphical presentation of the propeller design showing the number of blades selected and a single cylindrical hub.

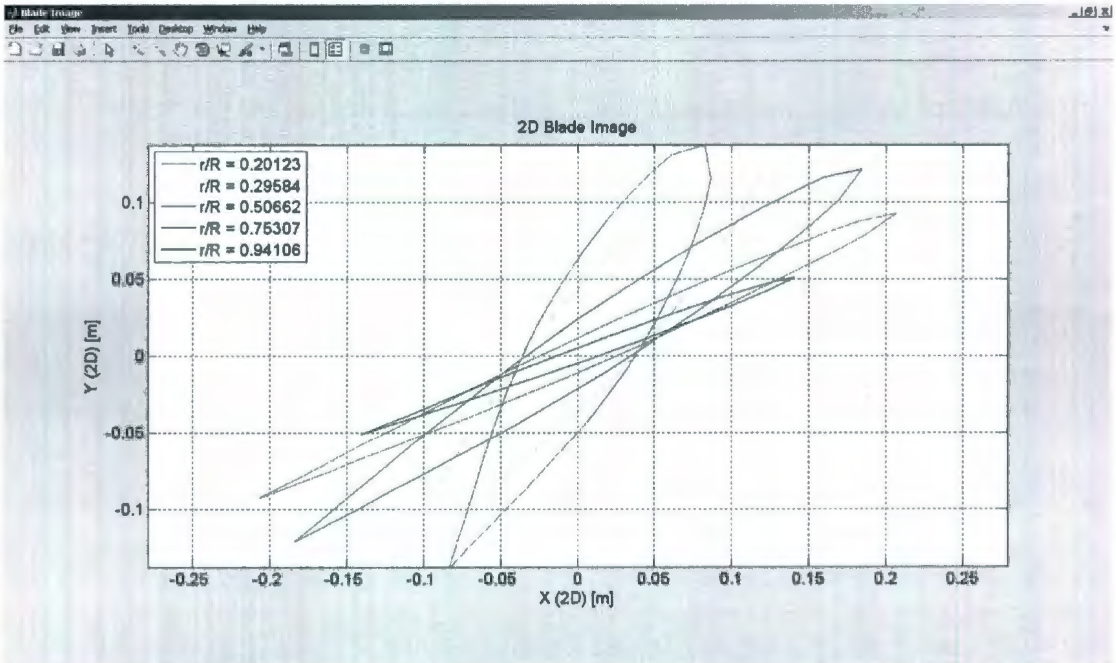

Figure 7: 2D Propeller Blade Profile

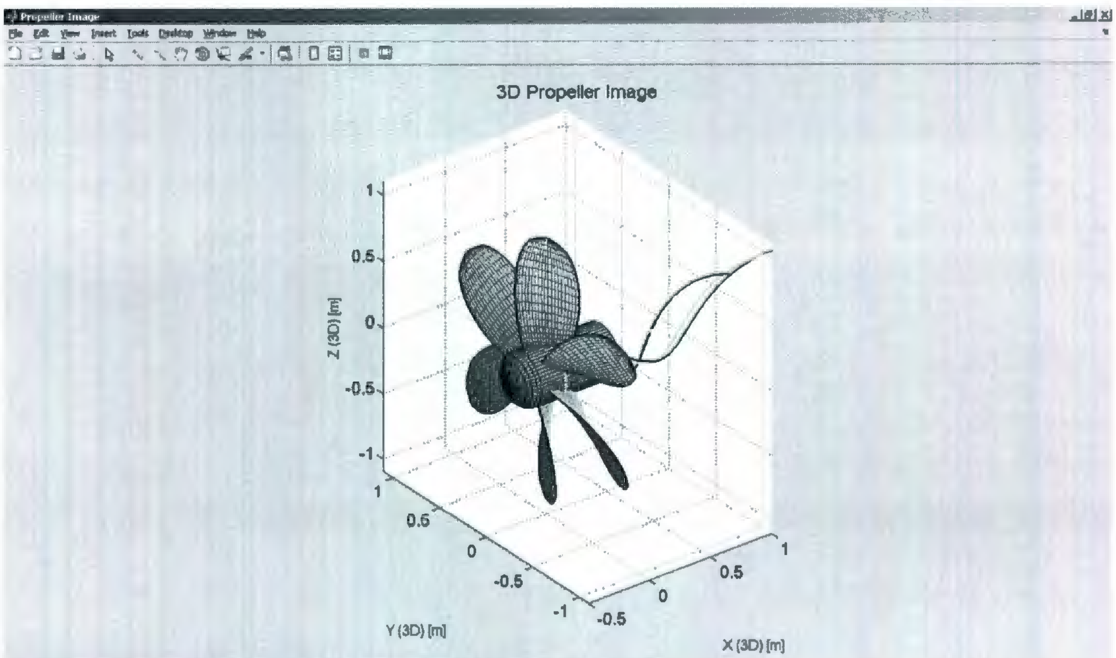

Figure 8: 3D Propeller Image

2.3.3 Propeller Geometry Development Through CAD

After the blade design is determined to be satisfactory, the program will create a text file (OpenPVL_CADblade.txt) to export the propeller blade geometry. The file provides a series of points that describe each foil section of the blade. This file was then used as an input for the CAD software RHINO, which generates the points for the foil sections. The propeller geometry is generated by creating closed splines for each sections and connecting the sections to generate a blade. The hub design is the following step, and then based on the designed number of blades to add all blades onto the hub [20]. After the propeller geometry is generated, a file can be created by RHINO to fabricate a prototype for testing by a rapid prototyping (RP) machine (.STL format). The RP part can then be tested. If the testing results are not satisfied, designers can go back to modify the propeller geometry in RHINO or regenerate a new design in OpenPVL, and then prototype and test again. These fabrication-testing-modification procedures are repeated until a satisfactory propeller is created.

In 2007, D'Epagnier designed an AUV propeller using OpenPVL code, and the AUV Propeller has the following characteristics [20]:

- The propeller has three blades.
- The propeller is operated on the vehicle at 120 RPM in order to reach a speed of 1.0 m/s.
- The diameter of the propeller is 0.6096 m.
- The required forward thrust of the propeller is 75 N.

- The diameter of the hub is 0.12192 m.
- Inflow wake velocity variation is –0.03 m/s.

After running the blade design function with the propeller parameters, the blade geometry was created in RHINO by importing the OpenPVL_CADblade.txt file and then manipulating the blade as described. Figure 9 shows the propeller in RHINO.

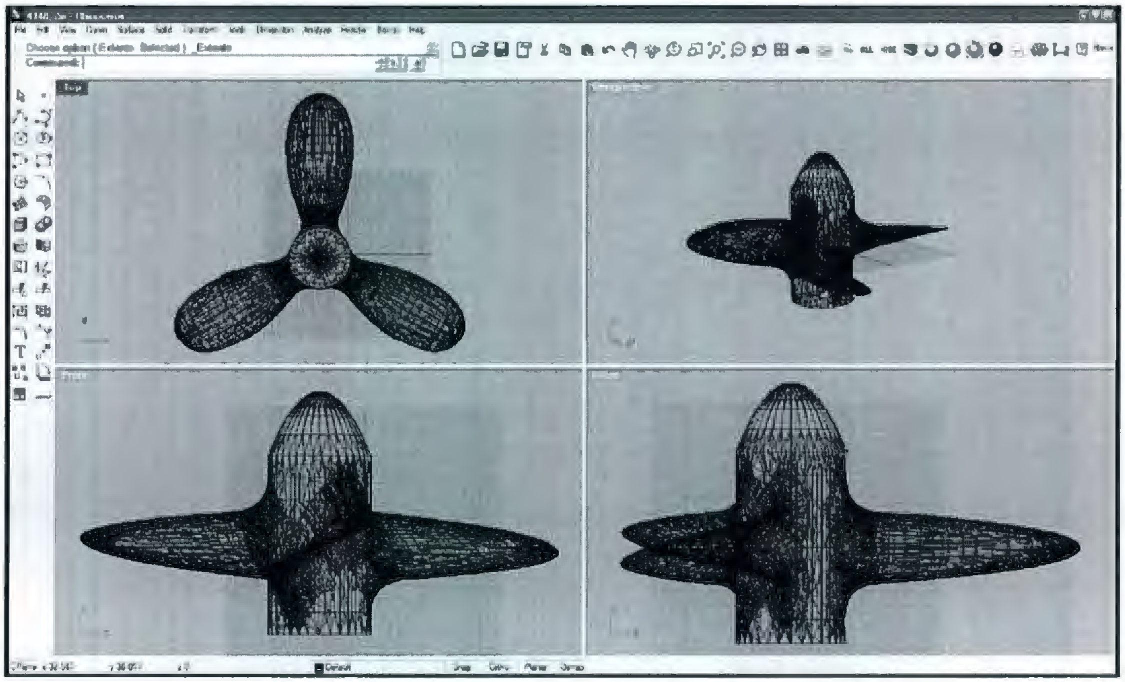

Figure 9. Propeller in Rhino [30]

D'Epagnier proved that the actual FDM-printed blade geometry was the same as the desired blade geometry that was determined in OpenPVL [20]. D'Epagnier used a milling machine with a dial indicator to measure the propeller blade with three different r/R values, 0.25, 0.70 and 0.80. Along each r/R value five evenly-spaced points were chosen for testing. In Figure 10, the green lines are the desired blade geometry with the three different r/R values of the AUV Propeller, and the triangular points are the tested points of the actual 3D printed blade. Due to the difficult in pinpointing the edge of the blade with a dial indicator and the effects of deflection, the measurements of the blade geometry at the leading and trailing edge are somewhat less in aggreement. However, the other points show good alignment with the desired results. This experiment was only applied for geometry validation that the produced RP test propeller was the desired propeller, which was designed in OpenPVL. The experiment didn't display the propeller's hydrodynamic parameters.

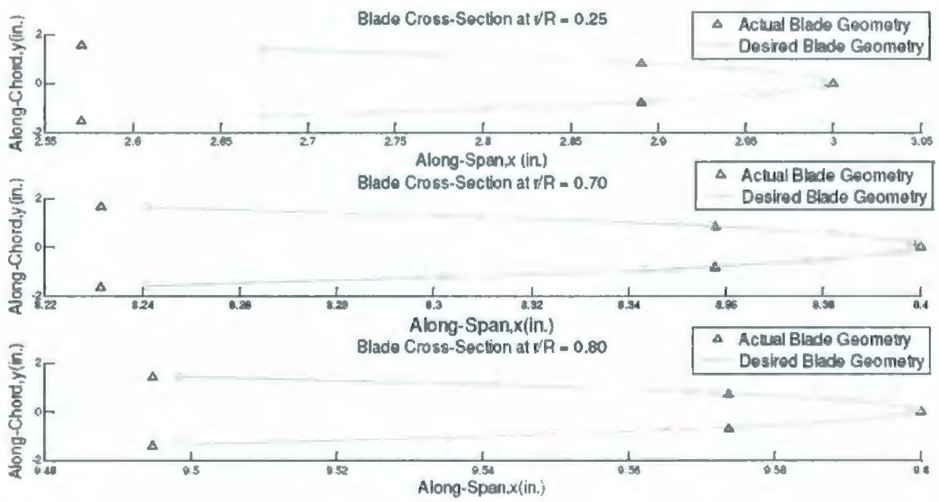

Figure 10: Propeller Geometry Validation [30]

2.4 Review of Computer-Aided Design (CAD)

Normally, propeller design codes are combined with Computer-Aided Design (CAD) technology to complete the design. Computer-Aided Design (CAD) is defined as, the use of computer technology for the design of objects. Started in the late 1980s, Computer-Aided Design programs were used to design curves and figures in two-dimensional (2D) space or curves, surface and solid in three-dimensional (3D) objects, thus beginning a trend for many companies to reduce cost in drafting departments. As a general rule, one CAD operator could replace at least three to five drafters who designs by hand. CAD could be used in many applications, such as the automotive, shipbuilding, aerospace industries, industrial and architectural design [21]. Due to fast paced development of personal computers, a large number of CAD software packages have been created and developed from 2D to 3D, from simple to complicated design and solid modeling, such as, AutoCAD, CADRA, MiniCAD, Euclid-IS, Pro/Engineer, SolidWorks, Solid Edge and so on [22].

SolidWorks, a mid-price CAD software package, has a large number of customers worldwide. In this thesis, SolidWorks is the CAD software used for propeller design and simulation. This CAD software is based on the parasolids solid modeler and utilizes a parametric feature-based approach to create models and assemblies for 3D mechanical design and solid modeling. Started in 1995, SolidWorks already has many applications and tools for mechanical design, such as drawing tools, design validation tools, product data management tools, design communication and collaboration tools and CAD productivity tools. SolidWorks has become a comprehensive mechanical CAE (Computer-Aided Engineering) software, to aid in engineering tasks [23]. CosmosFloWorks is the Computational Fluid Dynamics (CFD) application in SolidWorks and can be applied to fluid-flow simulation and thermal analysis. CosmosWorks is the Finite Element Analysis (FEA) application that can be used for stress analysis.

2.5 Review of Computational Fluid Dynamics (CFD) and Finite Element Analysis (FEA)

The technology of Computational Fluid Dynamics (CFD) as a design validation tool is brought into SolidWorks to use numerical methods and algorithms to solve and analyze problems related to fluid flows. CFD technology discretizes the spatial domain into small cells to form a volume mesh or grid, and then applies a suitable algorithm to solve the equations of motion. Treating a continuous fluid in a discretized format is the most fundamental consideration in CFD [24].

The Finite Element Method (FEM) and the Finite Volume Method (FVM) are the two main discretization methods widely used in the CFD field. The FEM uses standard techniques for finding solutions of partial differential equations (PDE) as well as integral equations. The solution approach is based on eliminating the differential equation completely or converting the PDE into an approximating system of ordinary differential equations, which are then numerically integrated using standard techniques such as Euler's method, Runge-Kutta and so on. FEM is mostly used in solving partial differential equations over complicated geometry, such as cars, to increase prediction accuracy in important areas [25].

Finite volume method (FVM) is a classical approach used most often in commercial and research codes. The governing equations are solved on discrete control volumes. FVM recasts the partial differential equation (PDE) of the Navier-Stokes equation in the conservative form and then discretizes the equations. This guarantees the conservation of fluxes through a particular control volume. Another advantage of the finite volume method (FVM) is that it is relatively easy to formulate for unstructured meshes [26]. FVM is often used in dynamic flow analysis, like marine propellers. In 2008, Pawel Dymarski [27] published a paper that presented a computer program SOLAGA to compute viscous flow around a ship propeller. The numerical model used for solving the system of main equations is based on FVM. The solution domain is subdivided into a finite number of control volumes, which are solved based on the integral from of the conservation equations. In the results, the calculated pressure distribution over the blades of the propeller is smooth, and the calculated propeller thrust and torque are in agreement with the experimental results. This paper showed that FVM is applicable to flow dynamic analysis for propellers. Presently, Finite Volume Method (FVM) is used in many computational fluid dynamics packages, such as, CosmosFloWorks, which is the SolidWorks integrated fluid simulation application.

SolidWorks also provides highly-advanced Finite Element Analysis (FEA) functions to designers and engineers. FEA is a numerical technique to solve engineering analysis problems for structural and field applications. The idea of FEA is to break a complicated structure into small elements and each element is based on physical law to calculate algebraic equations and solve engineering problems [28]. CosmosWorks is the application package for FEA analysis. CosmosFloWorks can calculate the surface pressure results of a propeller, and then transfer the results into CosmosWorks to do a stress analysis. CosmosWorks use the pressure results as an input of the stress analysis, and then to check the strength of the propeller design. The detailed steps of CFD and FEA for a propeller design are listed in Chapter 3.

2.6 Review of Propeller Fabrication

Following the design validation through CAE, a real prototype part is needed to be fabricated for testing. With the development of computer aided engineering (CAE) software, some software applications are available for propeller simulation, which can predict the results of a propeller test and be a convenient way for engineers to speed up the design of the final propeller. Because simulation does not reflect all the aspects of the real world accurately, simulation models are designed and used with the goal of approximating the testing process, but, can't completely replace the testing process. Physical prototype testing is used to increase to an acceptable level the confidence that the simulation results are correct for the real component [29]. For this reason, propellers need to be fabricated for testing after the simulation. There are generally three methods of propeller fabrication: Casting and Computed Numerically Control (CNC) machining technology are the traditional methods. In the late 1980s, rapid prototyping (RP) was introduced and by the late 90's it was used as a low cost and fast process for physical prototype fabrication.

2.6.1 Traditional Fabrication Processes

Casting is a manufacturing process which involves pouring liquid material into a mould designed with the desired shape, and then allowing to solidify. The solidified component known as a casting, is usually ejected or broken out of the mould and then put through a finishing process [30]. Figure 11 is the flow chart of a standard casting process.

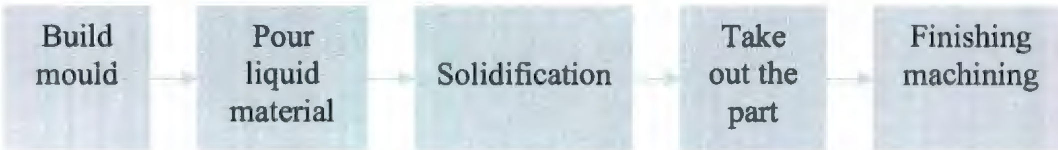

Figure 11: Flow Chart of Casting Processes

As a traditional manufacturing process, casting consists of several major steps from inception to completion of a product. These are: demand for a casting of specific shape and size; production of drawings, patterns or prototypes; the application of simple experienced-based rules to ensure good molten metal behavior. All these require some engineering to produce a good casting and will often take several attempts before a satisfactory result is obtained upon the development of a new product [31]. Due to these requirements of casting, cost and process time are major problems for building test prototypes. CNC machining technology is also commonly used to produce propellers. In the CNC process, CAD files provide the input for computer aided manufacturing programs, which are commonly used to extract computer file for a component and to extract the commands that are loaded into the CNC machines for production. CNC process can be significantly faster than casting but they are still fairly time consuming and expensive. For example, the time to produce a propeller with 200mm diameter usually takes more than two weeks, and the cost is around $2000, if the material is brass (Technical Service, Memorial University). Compared with CNC technology, casting will take a longer time and cost much more, because several processing technologies will be used for finishing machining. However, casting has two advantages: casting can save much more time than CNC for volume production; another is that compared with cutting off material in CNC, casting can save material. However, both of the casting and CNC technologies take a long processing time, which is inconvenient for a rapid propeller-testing requirement. This is a real problem when the actual performance has yet to be tested and the design is not entirely finalized. Rapid prototyping is a much faster and cheaper fabrication technology for rapid product development.

2.6.2 Rapid Fabrication Processes

In the late 1980s, the first technique for rapid prototyping became available, and was used to produce models and prototype parts. Rapid prototyping (RP), using additive techniques for the automatic construction of physical objects directly from three dimentional CAD data, can significantly reduce the time for the product development cycle and improve the final quality of the designed product [32]. A large number of prototyping technologies are available in the marketplace, such as, selective laser sintering (SLS), fused deposition modeling (FDM), stereolithography (SLA), Laminated object manufacturing (LOM), Solid Ground Curing (SGC) and 3D printing (3DP) [33]. Each prototyping technology uses specific materials and techniques to manufacture parts. Selecting the appropriate technology for RP fabrication depends on the desired accuracy and material requirements of the component being produced. Table 3 shows the material for each prototyping technology.

Table 3: Prototyping Technology Materials [34]

Prototyping technologies	Materials
Selective laser sintering (SLS)	Polycarbonate; Nylon; Glass filled nylon; Copper-impregnated nylon; Flexible rubber; Steel; Silica based sand; Zircon based sand; Investment casting wax
Fused deposition modeling (FDM)	Acrylonitrile Butadiene Styrene (ABS); Medical grade ABS; Methyl methacrylate ABS; Polycarbonate plastic; Investment casting wax;
Stereolithography (SLA)	Acrylin resin; Bi-colour acrylic resin; Epoxy resin; High temperature epoxy resin; Flexible epoxy resin
Laminated object manufacturing (LOM)	Adhesive backed paper; Adhensive backed polymer; Adhensive backed glass fibre
Solid Ground Curing (SGC)	Photoreactive resin
3D printing (3DP)	Water based liquid binder on cellulose starch powder formulation

Fused deposition modeling (FDM) [28] was developed by S. Scott Crump in 1990. The principle of FDM like all RP technologies is to lay down material in layers. An extrusion nozzle is heated to melt the material and can be moved in both horizontal and vertical directions by numerical controlled mechanism, directly controlled by a computer-aided manufacturing (CAM) software package. The main materials for FDM include Acrylonitrile Butadiene Styrene (ABS); Medical grade ABS; Methyl methacrylate ABS; Polycarbonate; Investment casting wax [34]. The process of the FDM machine is fast and relatively low cost. In 2007, Nadooshan [35] used FDM technology to construct a wind tunnel model with Polycarbonate plastic material, which is an actual impact-resistant industrial-grade thermoplastic and is structurally strong. Traditionally, wind tunnel models are made of metal and are very expensive. FDM was used as a way to reduce time and cost. Figure 12 displays the wind tunnel model constructed by FDM.

Figure 12: Wind Tunnel Model Constructed by FDM

This wind tunnel model was tested by engineers to compare with the real values of this wind tunnel model with metal material. The most purpose of this test in the wind tunnel is forces and moments. The results displayed that the accuracy of the data is lower than that of a metal model due to surface finish and dimensional tolerances, but the FDM model is quite accurate for a testing level. This FDM model cost about $650 and took 4 days to construct, while the metal model cost about $1300 and took a month to design and fabricate. The conclusion was that FDM technology is a timely and cost effective way of producing test parts.

In this wind tunnel model, the material needs to be strong enough to sustain the air force, which is on the nose cone and the edges of the wing tails. However, forces on a marine propeller are around all blades and nose cone and the fluid is much more dense. If a propeller needs to provide a large thrust for a vessel, this polycarbonate plastic material may not be strong enough. Selective Laser Sintering (SLS) is a better choice to produce a propeller where higher strength is required. SLS [34] is a rapid prototyping technology that uses a high power laser to fuse small particles of plastic, metal, ceramic or glass powers into a mass to build a desired 3-dimensional object. The laser selectively fuses material by scanning cross-sections, which will be used as the solid part of the object. The scanned parts are melted and become a solid. After each cross-section is scanned, the platen on which the object is built is moved down by one layer thickness, a new layer of material is applied on top of the solidified layer and the process is repeated until the object is completed. As shown in table 2, the SLS technology has a wide range of available materials. Using SLS technology, objects can be manufactured from high strength materials, such as uncoated or polymer-coated steel powders, which are unavailable for other technlogies. The layer thickness of SLS is 0.001-0.004 inch,and laser diameter is 0.004-0.02 inch [36]. With the small heating diameter, thin process layer and high strength of material, propellers can be produced by SLS with a good accuracy, surface finish and strong solid body. In this thesis, the availble prototyping machine was an FDM 2000 which was adequate for small testing propellers. A small tip can be used to make a smooth surface and the material of Polycarbonate Plastic (PC) can be used to make a strong propeller.

2.7 Review of Propeller Testing

After a propeller is fabricated, the physical testing procedure is used to test the propeller performance. Thrust is a very significant parameter for a propeller along with torque, so the testing device is set up to test the propeller thrust. Figure 13 is a sketch of the equipment setting for propeller testing.

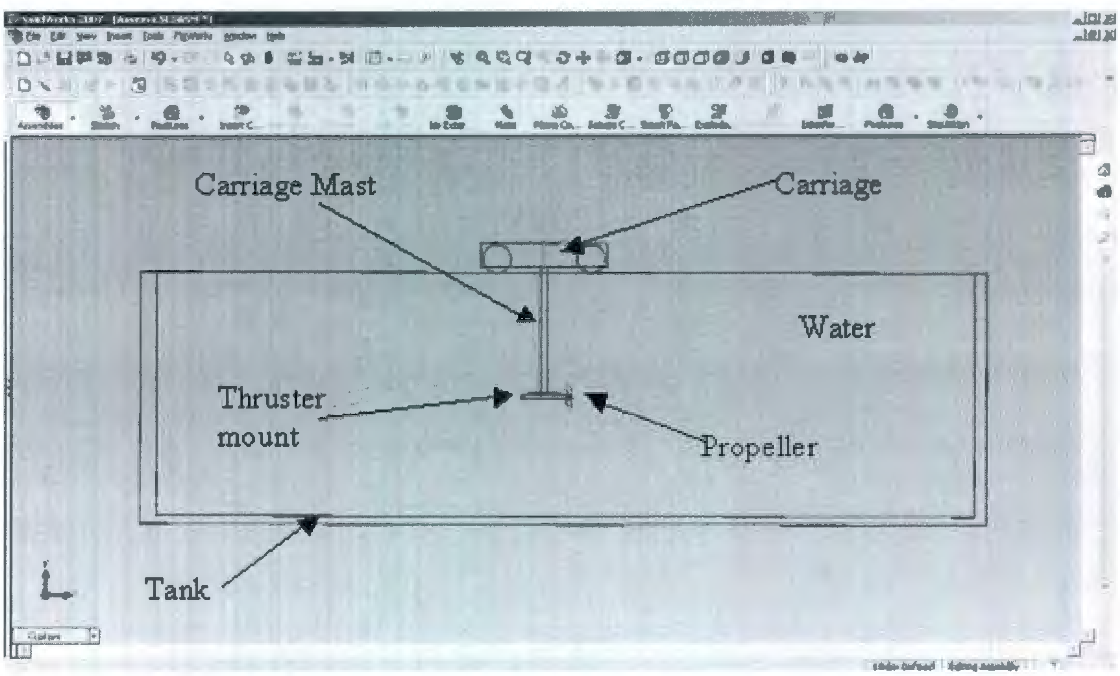

Figure 13: Sketch of a Propeller Testing Set-up

In the experiment for the propeller thrust test, a tow-tank is set up to provide the water domain. The tested propeller is mounted on a thruster mount. Figure 14 shows an example equipment set-ups of the thruster mount. The thruster mount is connected with the tow-tank carriage, which, working as the speed supply, is controlled to move with specific speeds to simulate ship motion. The equipment of a tow-tank carriage is shown in Figure 15. A load cell is attached in compression to the top of the thruster mount, and connects with a computer. In the computer, there is a program that is used to convert the load cell signals to calculate propeller thrust.

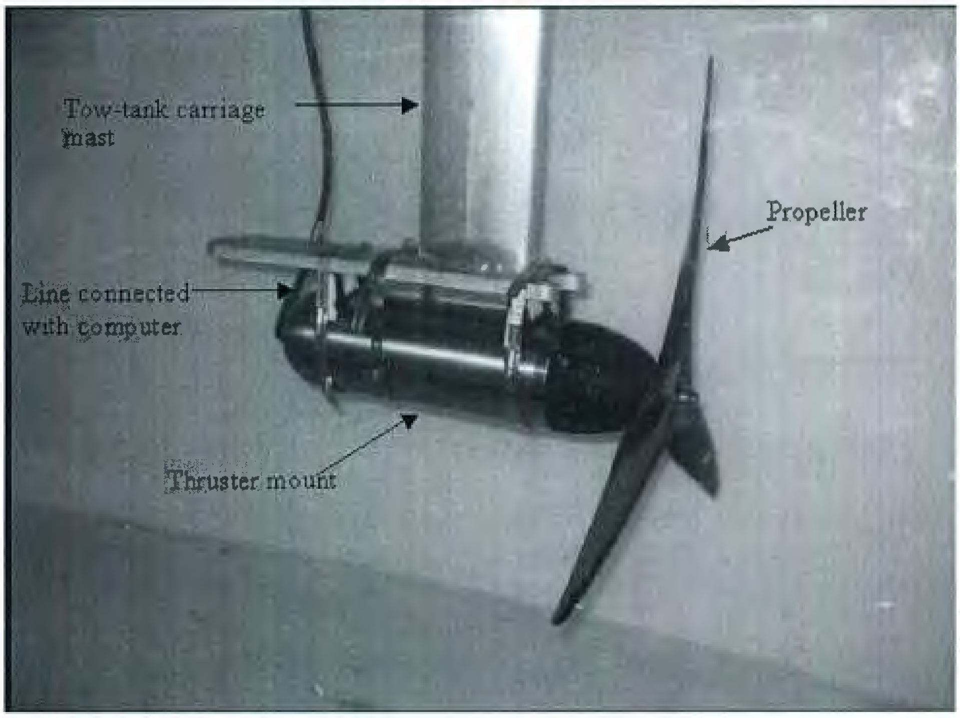

Figure 14: Thruster Mount

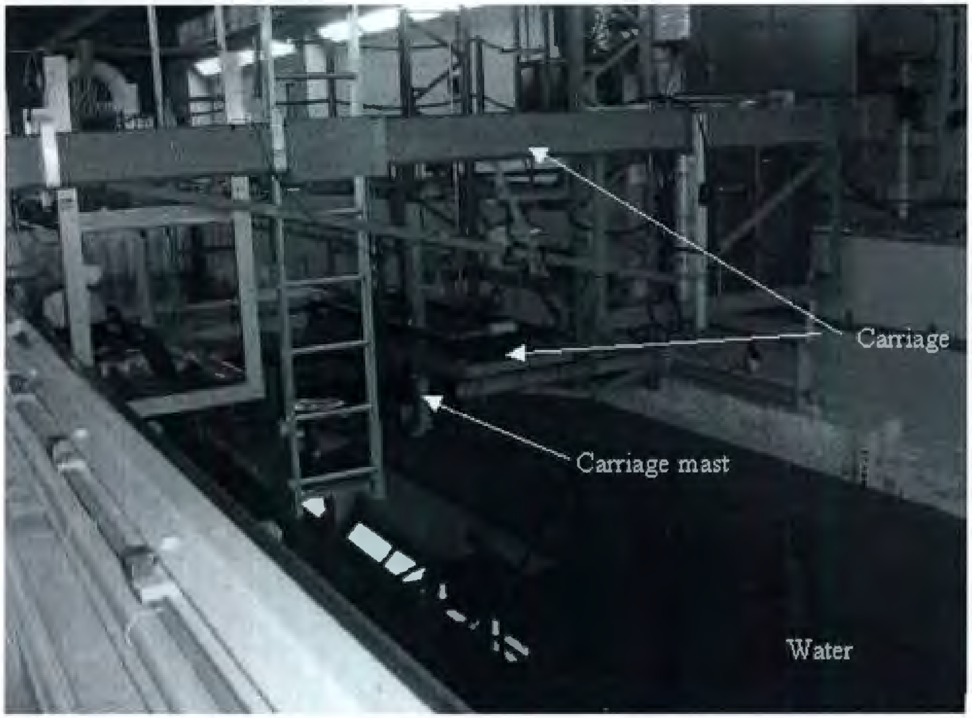

Figure 15: Tow-tank Carriage

Sometimes, the propeller size and thrust value are very large. Due to the limitation of cost, tank size and testing range of dynamometer, propellers are usually tested from a smaller size model first. The laws of similarity are used to provide the conditions, under which a model must operate, so that its performance will reflect the performance of the prototype [15]. The first condition is the geometrical similarity, which requires that the model's geometry is similar to the full-size propeller. Because of this, the ratio of every linear dimension of the full-size propeller is a constant ratio to the corresponding dimension of the model. If the diameter of a propeller is 5m, and the diameter of the model is 1m, the scale ratio $\lambda=5/1=5$, and this ratio should be held constant for all linear dimensions, such as the hub diameter, the chord lengths and blade thickness. The second condition is the kinematic similarity, which requires that the ratio of any velocity in the flow field of the full-size propeller to the corresponding velocity in the model is constant [15]. It reflects by equations as below.

$$\frac{V_{AS}}{n_S D_S} = \frac{V_{AM}}{n_M D_M} \text{ or } J_S = J_M \qquad [15]$$

S refers to the ship and M refers to the model. V_A is the speed of advance, and unit is meter per second (m/s). n is propeller revolution rate, and unit is revolution per minute (rpm). D is the propeller's diameter, and unit is meter (m). $J=V_A/nD$ is advance coefficient.

The third condition is the kinetic similarity, which requires that the ratio of the various forces acting on the full-size propeller is equal to the corresponding ratios in the model. This means the Froude number (Fn), Reynolds number (Rn) and Euler number (En) of the full-size propeller is equal to the model's.

$$F_n = \frac{V_A}{\sqrt{gD}}, \qquad R_n = \frac{V_A D}{\nu}, \qquad E_n = \frac{p}{\frac{1}{2}\rho V_A^{\ 2}} \qquad [15]$$

g is acceleration due to gravity, $g \approx 9.81 m/s^2$. $\nu=\mu/\rho$ is the kinematic viscosity of the fluid. ρ is mass density of water, $\rho=1000 kg/m^3$. p is the pressure associated with the propeller and the flow around it. μ is the coefficient of dynamic viscosity.

Furthermore, thrust coefficient (K_T) and torque coefficient (K_Q) of the full-size propeller is equal to the model's.

$$K_T = \frac{T}{\rho n^2 D^4}, \qquad K_Q = \frac{Q}{\rho n^2 D^5} \qquad [15]$$

T is thrust, and unit is Newton. Q is torque, and unit is Newton meter (Nm).

For given values of J, Fn, Rn, En, the values of K_T and K_Q are the same for the full-size propeller and it is geometrically similar model. All the above relations are used to calculate the thrust value for a propeller model test. If the tested value is close to the calculated value, it means the propeller has a desired performance as designed.

Chapter 3

Marine Propeller Design and Simulation in SolidWorks

Although the OpenPVL code has been developed extensively, there is still room for enhancement. Before making a prototype, simulation work is commonly done to predict the performance of the designed propeller. Based on simulation results, appropriate modifications will be applied, yet RHINO does not directly support simulation work. This simulation and analysis work has to be done by another software, which is not ideal. It is advisable to generate the propeller geometry in a CAD program, which can also directly support the simulation work. Solidworks, as an integrated CAD software package, is not only able to generate propeller geometry, but can also simulate the propeller fluid dynamics using the application CosmosFloWorks (CFD) and check the strength of the propeller design by using CosmosWorks (FEA). Updating the OpenPVL code and transferring the propeller geometry data into Solidworks became the continued issue. The author expands the application of Open_PVL code by creating OpenPVL_SW to generate propeller geometry for import into the SolidWorks software to initiate the simulation work.

In Chapter 3, the use of the OpenPVL_SW code for marine propeller design and the use of SolidWorks to generate propeller blade geometry and validate it through simulation will be discussed. In OpenPVL_SW, the parametric analysis function allows the user to combine several propeller parameters to optimize the propeller design, and then to use the optimized parameters to design a propeller blade. An output file OpenProp_Solidworks.txt from the propeller design function will be used as an input to generate the propeller blade structure in Solidworks. After a propeller blade is created, a hub will be designed. Based on the geometry of the blade and hub, other blades can be generated by the Circular Pattern function in SolidWorks. After a propeller is generated, CosmosFloworks and CosmosWorks will be utilized for propeller simulation.

3.1 Open_PVL and OpenPVL_SW codes

OpenPVL_SW is an extended version of Open_PVL to be able to create propeller geometry in SolidWorks, which can simulate the propeller working process using the integrated application CosmosFloWorks. The OpenPVL_SW code uses the same propeller analysis and design method as Open_PVL and is extended to generate a propeller blade geometry output file for SolidWorks. Compared with the Open_PVL code, the OpenPVL_SW code has two main advantages: The first, Open_PVL code generates a propeller blade geometry file for RHINO, which can not do the simulation work for a propeller. Thus, if engineers need to do a simulation to confirm the propeller design is good enough for a real testing, they have to transfer the propeller geometry into other simulation software. This is a waste of time during a large number of simulations and is inconvenient for engineers. The OpenPVL_SW code can generate a propeller blade geometry file for SolidWorks, which can use the integrated simulation application CosmosFloWorks to do a CFD simulation. This can save a lot of time and is much more convenient for engineers. Furthermore, the pressure output from the CFD simulation can be input into CosmosWorks that can be used to check the strength of the propeller design. The second, the propeller blade geometry file created by Open_PVL is the propeller foil sections points in RHINO. During each cycle at propeller design, engineers have to use several commands to create the propeller blade geometry and then design a full propeller. This is also inconvenient for engineers. The propeller geometry file created by OpenPVL_SW can automatically create a propeller blade geometry without any manual commands. This is much more convenient for engineers than Open_PVL.

As introduced in Chapter 2, the processes of propeller design by Open_PVL includes: parameter analysis, design of a propeller blade, generation of the propeller blade geometry in RHINO, hub design, adding other blades to complete the propeller, and fabrication of the prototype propeller. Because OpenPVL_SW code can create a propeller in SolidWorks, the CosmosFloWorks and CosmosWorks applications can be used for a propeller simulation to provide input into the design process prior to prototyping. Table 4 is the processes of propeller design and fabrication by Open_PVL and OpenPVL_SW.

Table 4: Processes of Propeller Design by Open_PVL and OpenPVL_SW

Processes of Propeller design by Open_PVL	Processes of Propeller design by OpenPVL_SW
Parameter analysis Selection of parameters Design of propeller blade Export blade section point data	
Generate a propeller blade geometry in RHINO	Generate a propeller blade geometry in SolidWorks
Design a hub Add other blades to complete the propeller	Design a hub Add other blades to complete the propeller
Rapid Prototyping	CFD by the SolidWorks integrated application CosmosFloWorks
	FEA by CosmosWorks to check the strength of the propeller design
	Adjust design as required
	Rapid Prototyping

Using OpenPVL_SW, a propeller design can be validated by the simulation work in CosmosFloWorks and CosmosWorks. The simulation work is quite convenient for propeller engineers to design and validate a propeller before fabrication.

3.2 Using Solidworks for Propeller Blade Geometry

A propeller can be designed by the parametric analysis and propeller design functions of OpenPVL. The OpenPVL_SW code was modified to create an output of OpenProp_Solidworks.txt file for generating a propeller blade in SolidWorks. After running the propeller design option, the file OpenProp_Solidworks.txt was created. Codes for generating a propeller blade is recorded in the .txt format file. The .txt file records the SolidWorks macro commands to generate a propeller blade. SolidWorks macro is a series of commands and actions that can be stored and run within SolidWorks to automatically draw geometry without manual working. Users can paste the codes in the window of Solidworks macro to generate a propeller blade. The file has three main functions: to generate foil section points for the propeller and connect each point to generate propeller foil sections; to connect each leading edge point of the foil section to create a reference line for the blade surface; to use SolidWorks "loft" function to generate the propeller blade based on the foil sections and the reference line. The flow chart shows how to generate a propeller blade in Figure 16.

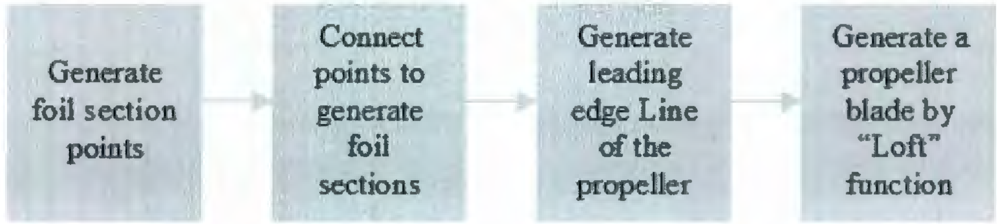

Figure 16: Flow Chart of Generating a Propeller Blade

3.2.1 Propeller Foil Sections

The foil section points are three dimensional values (x,y,z), therefore the SolidWorks function "3D sketch" is used to read the points information into SolidWorks. As referenced in Chung's paper [18], 40 points on a propeller foil section can provide sufficient resolution in geometry. OpenPVL_SW defines that each foil section has 40 points, which are in the order of 1-40. If users want to change the amount of foil section points, they can change the point amount in the OpenPVL_SW code, and confirm that the point amount is the same as the amount displayed in the propeller design function of Matlab graphical user interface (See in Chapter 2). The foil section point amount is defined by the commands of ReDim swSketchPt () in the OpenPVL_SW code. The number in the parenthesis mark is the number of foil section points. A Large number of points can provide a high resolution in propeller foil section geometry, however, too many points will waste more time on unnecessary calculations. Figure 17 shows the points on a propeller foil section.

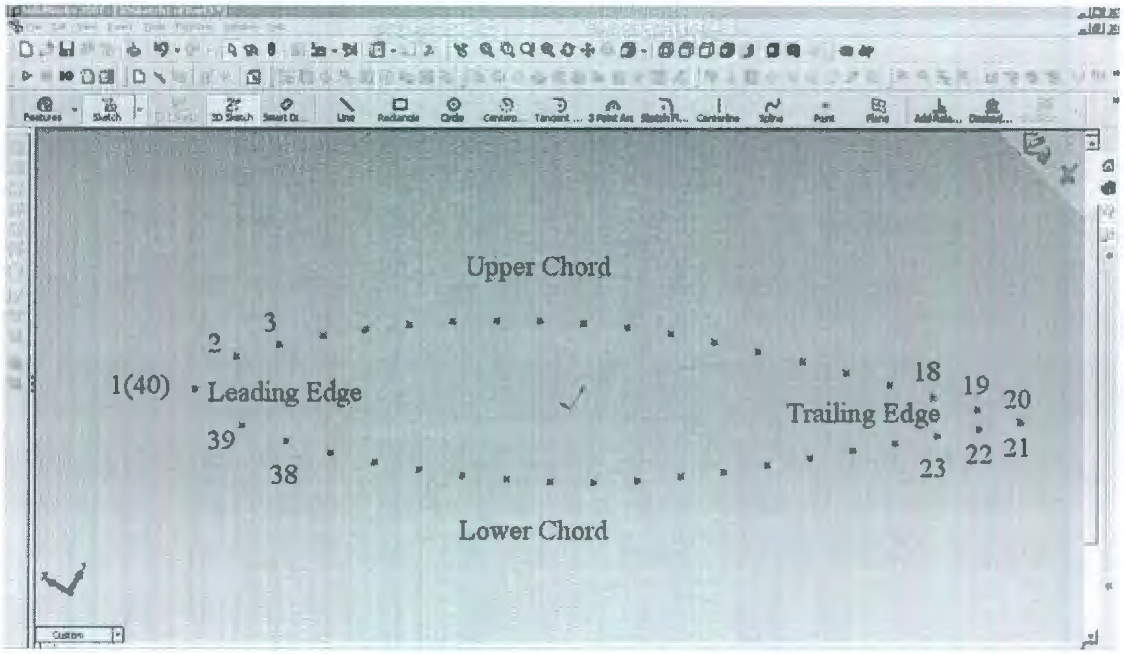

Figure 17: Foil Section Points

A curve passes through all the 40 points to generate a propeller foil section. The curve starts from point number 1, which is on the leading edge of a propeller blade. The curve will pass through all the upper chord points in an orderly way to point number 20, which is located on trailing edge of the propeller blade. After that, the curve will orderly pass through point number 21 to point number 40 (coincident with point number 1), which are on the lower chord of the propeller blade. As referenced in Karim's paper [37], B-spline is used as the curve to generate the propeller foil sections. The shape of the B-spline curve depends on the directions of tangent vector. B-spline can have as few as two points and can specify tangency at the end points. To achieve higher accuracy, a large number of points are needed for the approximation. In the OpenPVL_SW, 40 points are used in B-spline to provide a smooth and accurate curve for the propeller foil sections. A closed B-spline starts from the point number 1 and end at the point number 40, so the B-spline has the same tangency value at the leading edge point. Thus, a closed B-spline can provide a smooth curve at the leading edge point. In order to confirm that a propeller foil section is a closed curve, OpenPVL_SW defines that point number 1 and point number 40 have the same three dimensional values. The foil section is shown as Figure 18.

Figure 18: A Propeller Foil Section

OpenPVL_SW generates 21 foil sections to construct a propeller blade. These foil sections are defined in the order of 0-20 from blade root to tip. OpenPVL_SW generates the 21 foil sections in the order of 0-20. These 21 foil sections are shown in Figure 19.

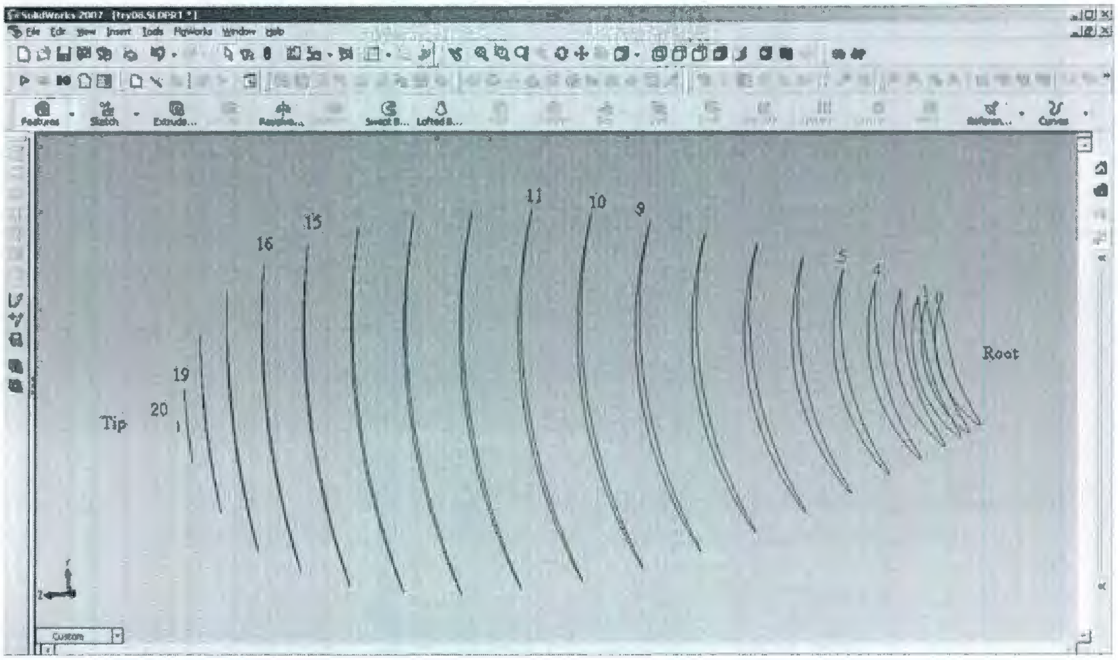

Figure 19: Overview of Propeller Foil Sections

Figure 20 is the side view of these 21 foil sections that shows the overall propeller geometry. These foil sections need to be linked by lofting a smooth surface to generate a propeller blade.

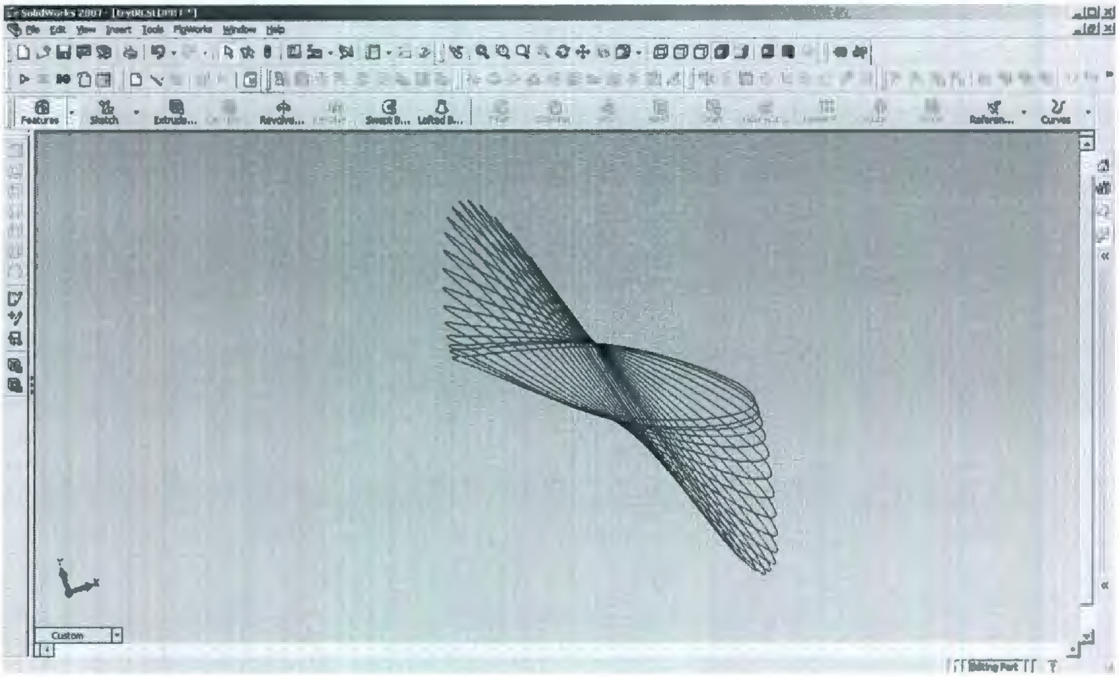

Figure 20: Sideview of Propeller Foil Sections

Propeller designers can add more types of foil sections in the thickness form and meanline sections of the original OpenPVL code. Every foil section is described by meanline and thickness form information. Meanlines are described by two sets of factors, the "foc" and the "dfdxN". The code of NACA a=0.8 mean line is shown as below. The factor "foc" is the ratio of camber to chord. The factor "dfdxN" is the slope of camber line. The information of these two factors can be found in NACA reports. If propeller designers want to use another mean line in the OpenPVL code, they can change the numbers that are in factors "foc" and "dfdxN" to reflect the characteristics of the new mean line.

Standard NACA a=0.8 meanline factors:

foc = [0 .287 .404 .616 1.077 1.841 2.483 3.043 3.985 4.748 5.367 5.863 6.248 6.528 6.709 6.79 6.77 6.644 6.405 6.037 5.514 4.771 3.683 2.435 1.163 0]./100

dfdxN = [.48535 .44925 .40359 .34104 .27718 .23868 .21050 .16892 .13734 .11101 .08775 .06634 .04601 .02613 .00620 -.01433 -.03611 -.06010 -.08790 -.12311 -.18412 -.23921 -.25583 -.24904 -.20385]

The NACA 65A010 thickness form is shown as below in OpenPVL code. The factor "toc" is the ratio of thickness to chord. In the original OpenPVL code, toc_65 is named as the ration of thickness to chord for the NACA 65A010. Users can rename the toc_65 for convince. For example, toc_63 is used to define the ration of thickness to chord for NACA 63A006. Propeller designers can change the thickness form factor to use another thickness form other than NACA 65A010, if desired.

NACA 65A010 thickness form

toc_65 = [0 .765 .928 1.183 1.623 2.182 2.65 3.04 3.658 4.127 4.483 4.742 4.912 4.995 4.983 4.863 4.632 4.304 3.899 3.432 2.912 2.352 1.771 1.188 .604 .021]./100

Figure 21 displays the foil section geometry using thickness form NACA 65A010 and a=0.8 meanline at the r/R=0.35.

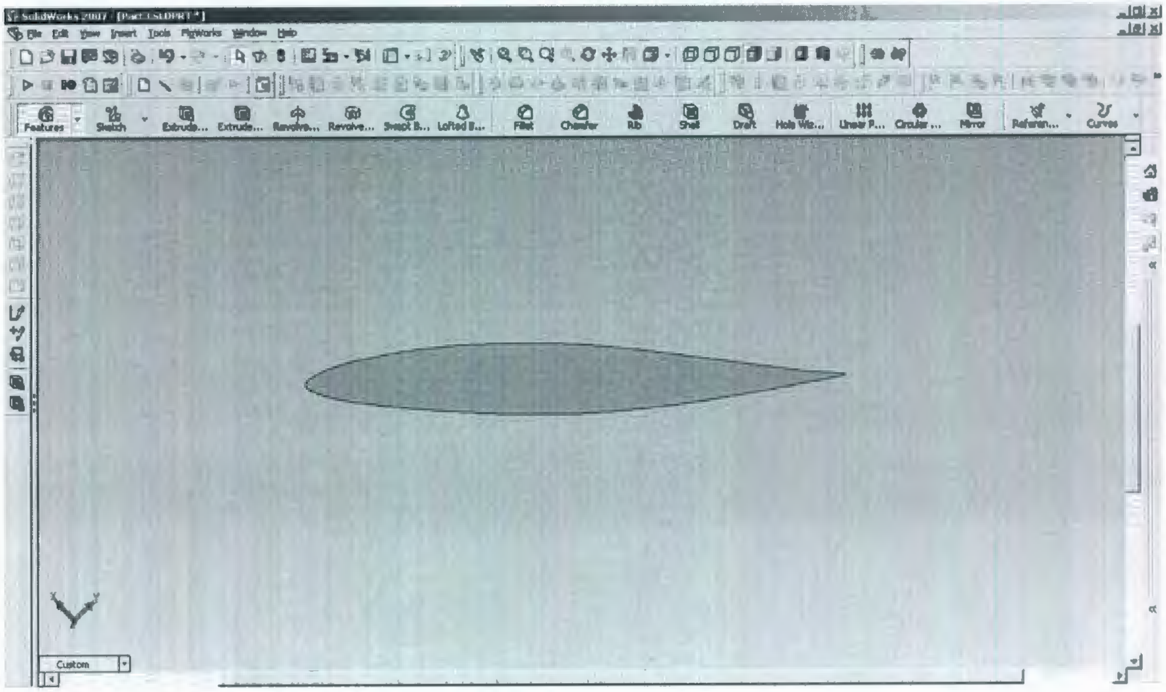

Figure 21: Foil Section using NACA 65A010 a=0.8 at the r/R=0.35

A foil section with a different thickness form (NACA 63A006) and meanline (a=0.8 modified) is provided as an example of how to change the geometry of foil sections. The information about the NACA 63A006 and mean line a=0.8 (modified) was found in NACA reports, and is shown in table 5.

Table 5: The information of NACA 63A006 and mean line a=0.8 (modified) [19]

NACA 63A006	toc_63 = [0 .495 .595 .754 1.045 1.447 1.747 1.989 2.362 2.631 2.820 2.942 2.996 2.985 2.914 2.788 2.613 2.396 2.143 1.859 1.556 1.248 .939 .630 .322 .013]./100
Mean Line a=0.8 (Modified)	foc = [0 .281 .396 .603 1.055 1.803 2.432 2.981 3.903 4.651 5.257 5.742 6.120 6.394 6.571 6.651 6.631 6.508 6.274 5.913 5.401 4.673 3.607 2.452 1.226 0]./100
	dfdxN = [.47539 .44004 .39531 .33404 .27149 .23378 .20618 .16546 .13452 .10873 .08595 .06498 .04507 .02559 .00607 -.01404 -.03537 -.05887 -.08610 -.12058 -.18034 -.23430 -.24521 -.24521 -.24521]

Figure 22 shows the foil section geometry using NACA 63A006 thickness form and a=0.8(modified) meanline at r/R=0.35.

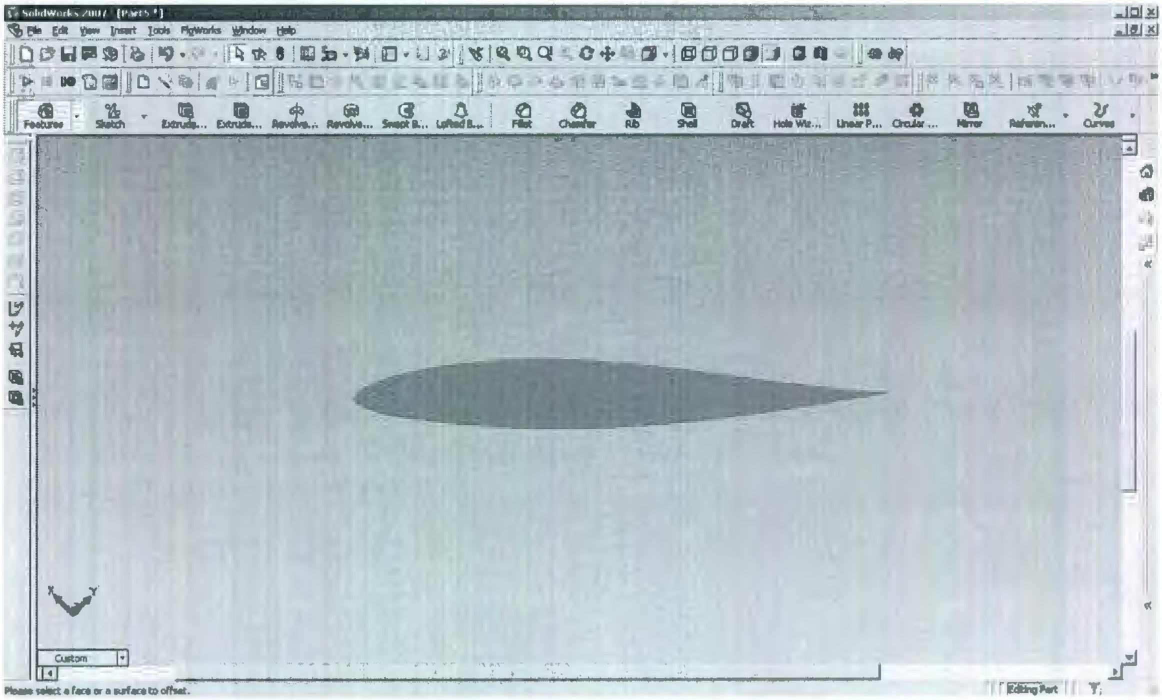

Figure 22: Foil Section Using NACA 63A006 a=0.8(modified) at the r/R=0.35

The method for incorporating a change the thickness form and meanline is to change the parameters of "foc", “dfdxN” and "toc" in the OpenPVL code as demonstrated.

3.2.2 Propeller Blade Surface Development

After the foil sections are created, a method is needed to connect them together to generate a derived propeller blade. In OpenPVL_SW, a leading edge line is created on the tip of the leading edge, which is shown as Figure 23. The leading edge line is created by connecting all the point number 1(40) of the 21 foil sections. All the foil sections follow the leading edge line to construct the propeller blade. The leading edge line is shown in Figure 23.

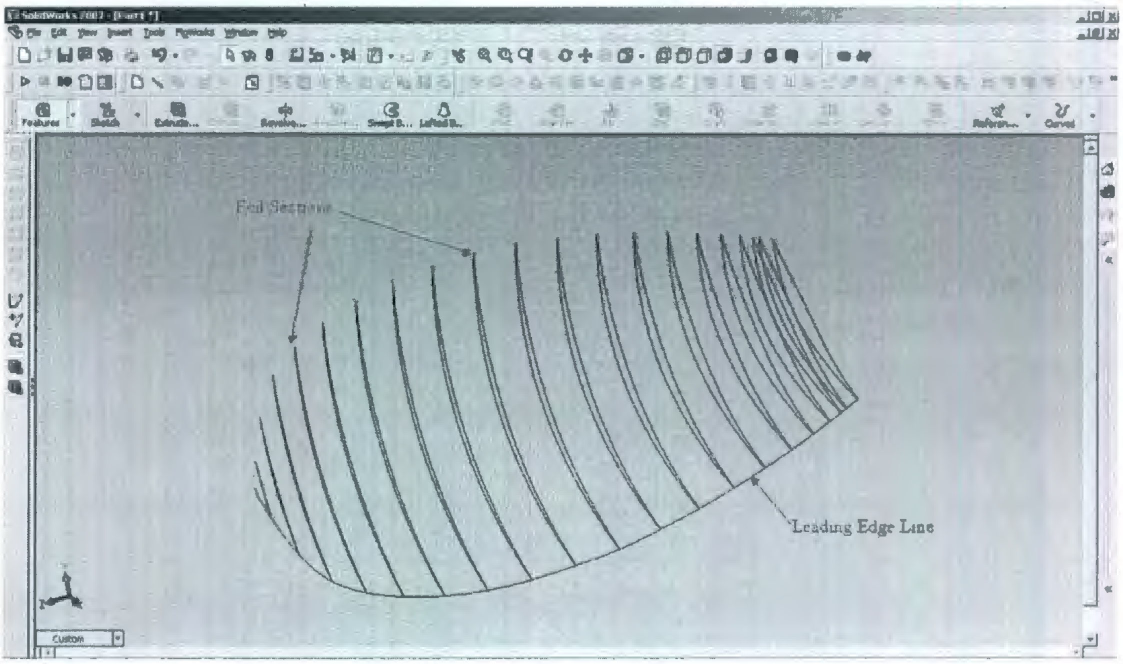

Figure 23: Propeller Foil Sections and the Leading Edge Line

After the foil sections and the leading edge line are created, the SolidWorks "Loft" function will be used to generate the propeller blade geometry. The loft function is used to create a feature by following transitions between profiles in SolidWorks. Usually, this function needs multiple sketches and a reference line to create the feature. Following the leading edge reference line, the loft function connects all foil sections to generate the propeller blade. OpenPVL_SW code defines that the foil sections are lofted in the order of 0-20, which is shown as Figure 19. The propeller blade is shown in Figure 24.

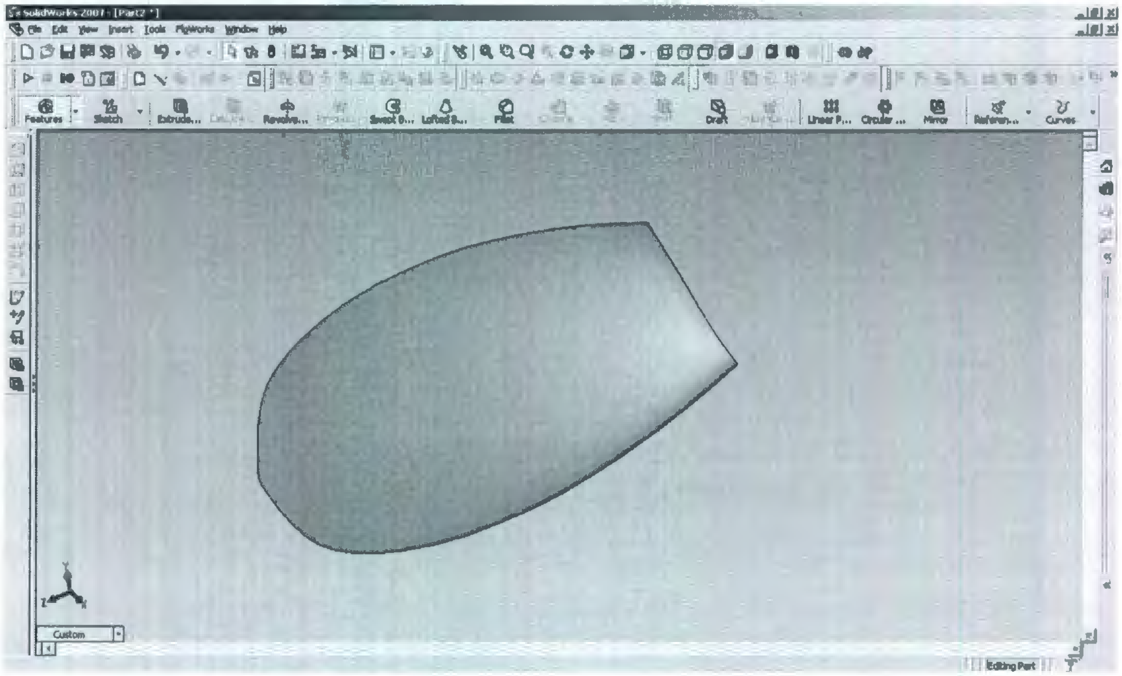

Figure 24: A Propeller Blade

After a propeller blade is created, a hub needs to be designed based on the hub parameter set out in the propeller design option of the OpenPVL_SW code. Usually, a nose cone is used on a hub to reduce the inflow effect when a propeller is rotating. Figure 25 shows a propeller blade, a hub with the diameter of 0.2m and a nose cone with 0.05m.

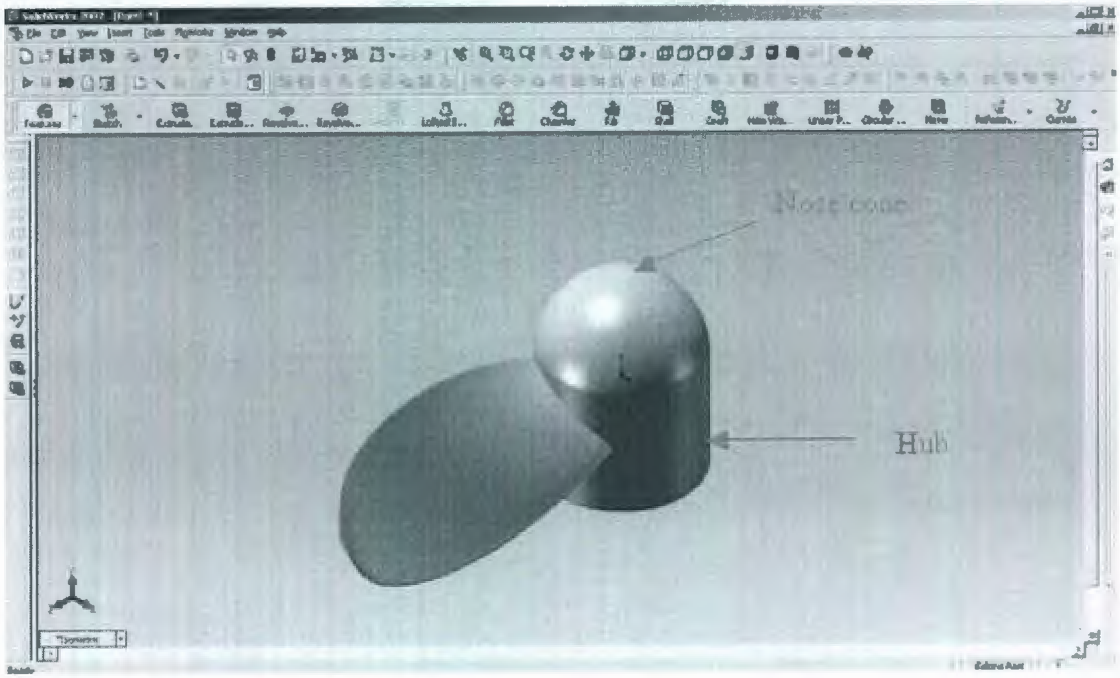

Figure 25: A Propeller Blade with a Hub and a Nose Cone

The following steps are used to generate other blades manually to complete the propeller geometry. The axis of the hub is set up as a reference line, and SolidWorks "Circular Pattern" function needs to be used to generate other blades based on the reference line. Figure 26 is an example of propeller geometry with three blades.

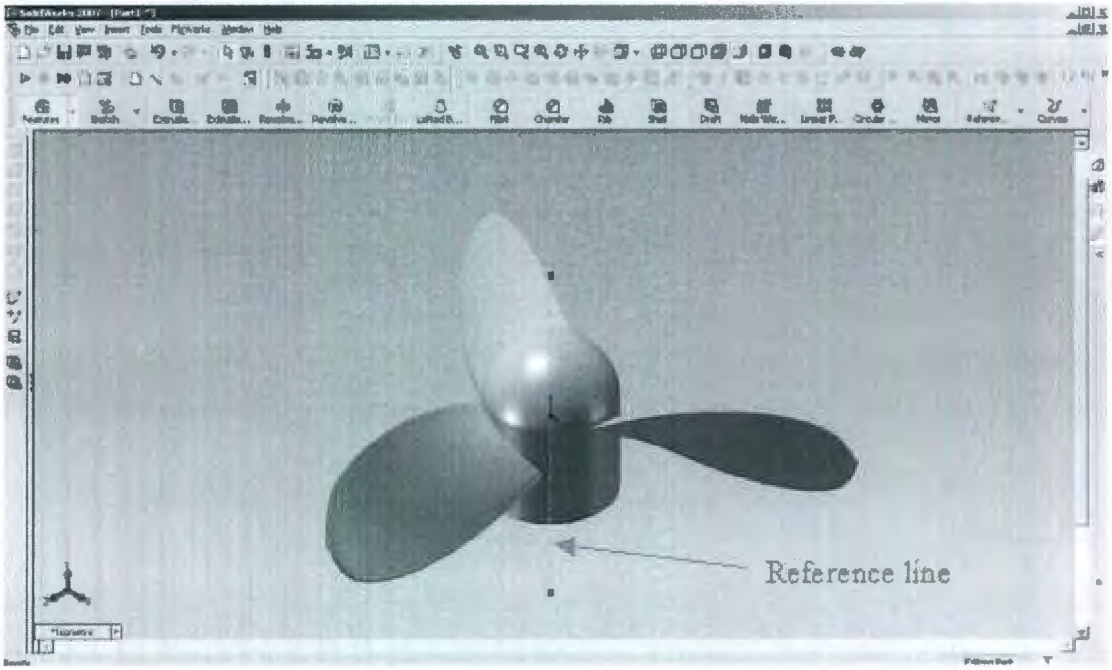

Figure 26: A Propeller with Three Blades

3.3 Simulations using CosmosFloWorks

Once the propeller geometry is fully described and satisfactory, CFD simulation work can begin using the SolidWorks application "CosmosFloWorks". CosmosFloWorks is an easy-to-use fluid-flow simulation and thermal analysis program that is fully embedded in Solidworks. CosmosFloWorks can be used to simulate thrust, which is one of the most important parameter in a propeller design. Based on real experimental settings that were introduced in chapter 2, a simulation structure is created in CosmosFloWorks. Because CosmosFloWorks doesn't allow to set up the velocity parameter for a solid part, it cannot simulate the motion of the carriage as described in chapter 2. Instead, the fluid velocity is controlled in order to simulate the carriage motion. For example, if the test carriage moves at 2 m/s forwards, CosmosFloworks can set up that the propeller is fixed with the fluid moving at 2 m/s backwards. The resolution of the automatic meshing is controlled through settings in CosmosFloWorks. The resolution level can be set to various levels with finer and finer granularity. The software does not permit detailed configuration of the mesh around particular geometric features. A higher level of resolution breaks the simulation domain into more small elements and can provide much more accurate results but it also requires significantly more computational effort. Determination of the exact flows, stresses and deflections in areas such as the trailing edge of the blades (where there are very fine geometric features) would benefit from an ability to adjust the mesh size locally. The simulation settings and assumptions are provided in the following lists:

- Analysis type is external for propeller simulation. (ie. Fuild flows outside of propeller)
- Water is used as the fluid domain.
- The only considered physical features are gravity and rotation, acceleration of gravity $g=9.81m/s^2$
- The temperature of water is 293.2 K.
- The air pressure is 101325 Pa.
- Pressure potential is considered.
- Assume no cavitation is in the simulation
- Assume adiabatic wall is used.

The simulation structure is shown in Figure 27. This simulation model provides the water domain, which serves the same function as the tow-tank in the real experiment.

Figure 27: Simulation Domain of Propeller

If users desire to model a specific test region and know the real testing tank size, they can set up the simulation domain size the same as the actual one. If users don't know the real testing domain size, they need to define the domain size based on several simulation results. When the propeller is rotating in the simulation domain, water is moving around the propeller. If the size of simulation domain is small, the simulation programs will be impacted by the flow effect at the edge of simulation domain. These impacts will cause calculation errors for the propeller thrust prediction. Larger domain size is better for simulation result accuracy, however; too large of a domain will increase the number of simulation calculations, which can waste a lot of time. The guideline for simulation domain design is to reduce the flow effect as low as possible at the domain edge. First, the depth and width of a simulation domain are set up based on the propeller working depth. If the propeller is designed to work at 3m below water, the depth and width of the simulation domain are set to 6m and 6m. The propeller is in the center of the domain. The second is the setting of domain length. In real propeller testing experience, 30m is a good starting setting for a propeller simulation. The effect of moving water at the edge of simulation domain can be monitored by water pressure. Figure 28 displays pressure results of a propeller simulation with domain size of 6m×6m×30m (width×depth×length). The diameter of the propeller is 0.6m and rotation speed is 120 RPM.

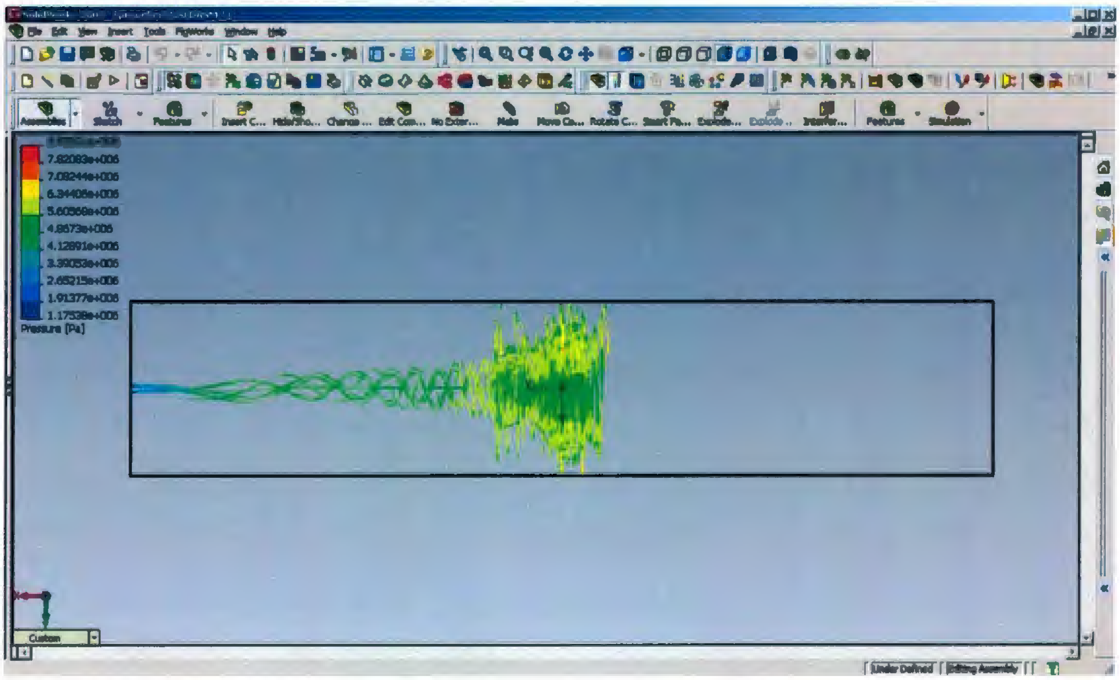

Figure 28: Pressure Results of a Propeller Simulation with Domain Size 6m×6m×30m (width×depth×length)

In figure 28, most of pressure effect is within the simulation domain. That means the size of simulation domain is appropriate for this propeller. If the same simulation is run with domain reduced to 4m×4m×20m (width×depth×length), the results are shown in Figure 29.

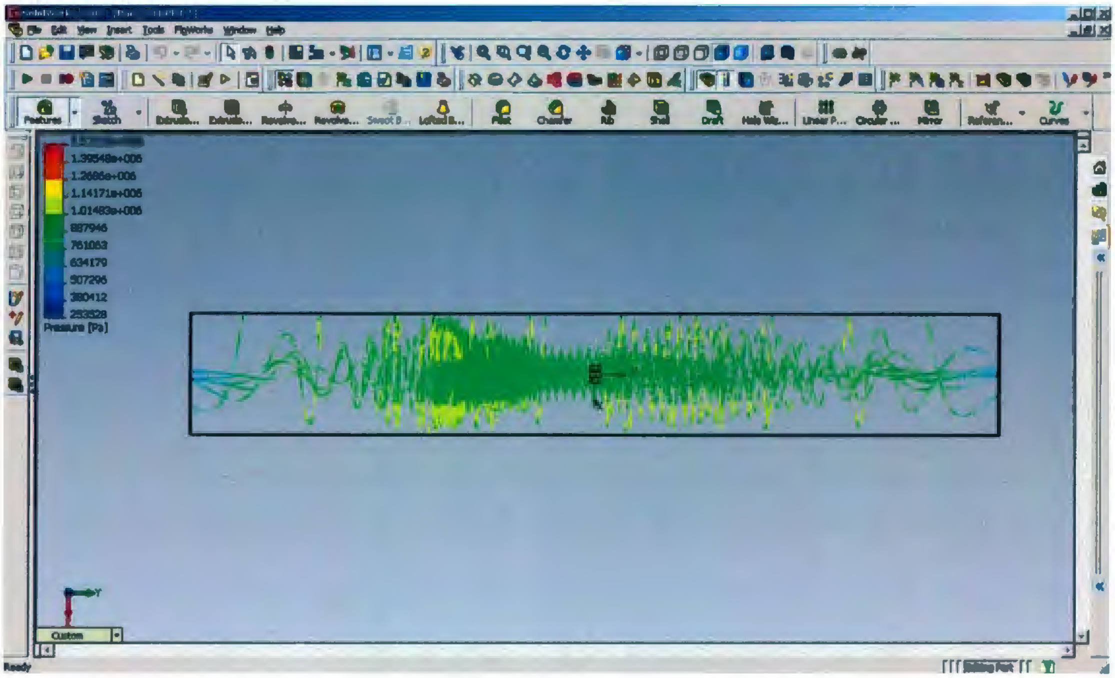

Figure 29: Pressure Results of a Propeller Simulation with Domain Size 4m×4m×20m (width×depth×length)

In this reduced simulation domain, there is much more pressure effect on the domain edge. That means this domain size is not big enough for this propeller simulation. These increased pressure effects will cause an error of propeller thrust prediction. Selecting a simulation domain with little water pressure at the edge is a good choice for a propeller simulation.

The simulation model provides water flow with a specific velocity to simulate ship motion, which is the same function as the tow-tank carriage. This model also provides propeller geometry. But, the model shown in figure 27 is not the same as a real experiment. The propeller geometry is enlarged in figure 30, so the difference is shown clearly.

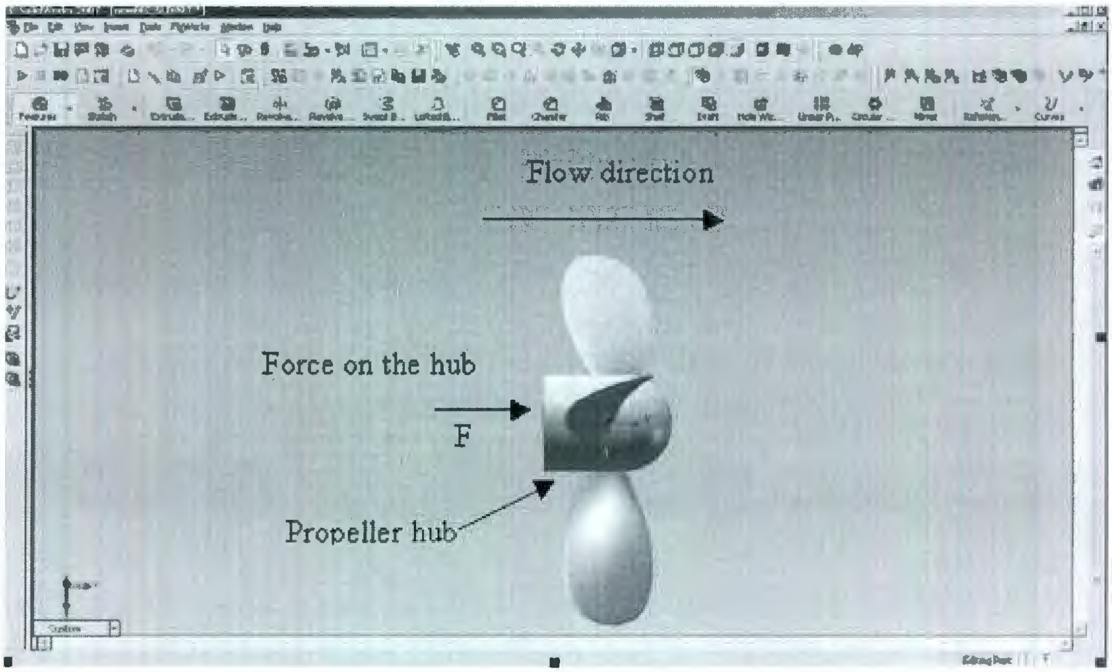

Figure 30: Side View of a propeller

A hub is in the center of a propeller. There are two sides of a hub: the dome side is to reduce the force from fluid; the flat side is to fix a propeller on a boat. As shown in Figure 14, the flat side of a hub is connected with a carriage and a thruster mount, so there is no fluid force on this side. From the simulation model as shown in Figure 30, there is force on the flat side of a hub when fluid is moving. This force will result in a discrepancy between the simulation and real experimental results. A method to eliminate this force will be introduced in the simulation model to get a more accurate result. Because all the simulation parts in CosmosFloWorks have to be in the simulation domain, a design of extending the flat hub side outside of the domain to eliminate the force on the flat side doesn't work. Finally, the propeller simulation model is separated into two steps. First, a simulation is run as the propeller structure in figure 30. Second, a simulation is run for the propeller without blades, which is shown in figure 31.

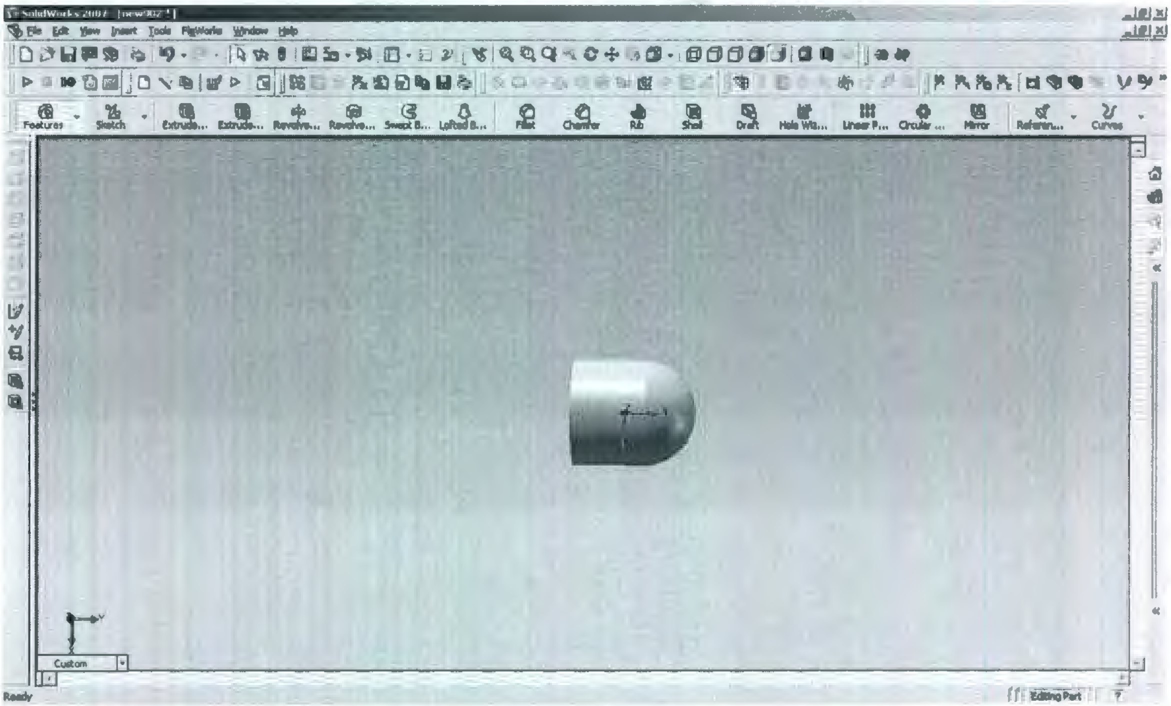

Figure 31: A Simulation Model Without Blades

The direction of thrust and force on the hub is opposite, so the force on the hub reduces the simulation result for the thrust. The final simulation value of propeller thrust is the simulated thrust value from CosmosFloWorks adds the force on the hub.

3.4 Simulations using CosmosWorks

CosmosWorks is a design analysis application, which is integrated in SolidWorks. CosmosWorks uses the Finite Element Analysis (FEA) to simulate the working conditions of designs and predict the behavior. With the fast solvers, CosmosWorks can not only quickly analyze the designs and search for the optimum solution for designers, but also shortens time to market by testing the designs on the computer instead of expensive and time-consuming field tests.

CosmosWorks can be used to check the strength of the propeller design. In CosmosFloWorks, the propeller is considered as a solid part without deformation; however, the propeller will be undergoing stress when it is rotating. FEA analysis is used to check that the propeller is strong enough when it is rotating. CosmosFloWorks can provide the resultant fluid pressure that is acting on the propeller blades. This pressure difference is the source of bending force, which results the stress that could cause a propeller to be destroyed. CosmosWorks can be used to analyze propeller's strength by FEA. CosmosWorks uses the fluid pressure as an input parameter to calculate the stress within the propeller and determine the Factor of Safety (FOS). The value of FOS is calculated by limit yield stress over each stress value. The acceptable FOS value for a propeller is not less than 1.5 [38]. Instead of 1.5, some designers use 2 to make a higher strength propeller. The FEA analysis can help engineers to choose a good material and hub design to make sure the propeller design is strong enough. The steps of FEA analysis are shown as below:

- Use CosmosFloWorks to calculate the surface pressure of propeller blades
- Transfer the CosmosFloWorks results to CosmosWorks in the Tools option
- Create a static study in CosmosWorks
- Load the CosmosFloWorks results file (.fld) into the static study
- Set the restraint for the propeller. The bottom of the hub is fixed.
- Select propeller material

- Set contact and centrifugal features for the propeller. Contact: bonded between the hub and propeller blades; centrifugal: it is used to simulate the rotation of the propeller, the unit is rad/s.
- Mesh and run the FEA
- Plot the Stress distribution and FOS results
- If FOS ≥ 1.5, the propeller is strong enough; if FOS <1.5, the propeller has a weak strength, designers can change the propeller geometry to increase the strength or use much more strong material for the propeller.

For example, Figure 32 displays the surface pressure of a three-blade propeller with 120 RPM working in 5 meters depth of water. The material of this propeller is Titanium Alloy with 1.03×10^9 N/m^2 yield strength as an example. In the propeller manufacturing, there are several rules to select the material for a propeller [15]: the processes to fabricate a propeller are casting and machining, so the propeller material must be amenable to these processes; the propeller material should have a high strength and toughness, and the fatigue strength is important as well; resistance to corrosion in water is desirable quality in a propeller material; the propeller can be easily repaired if it is damaged; the cost of the propeller material is also a consideration. In this thesis, the material used in FEA only is an example without detailed considerations, and to display how the CosmosWorks works. In a real propeller deisgn, the selected material should select based on the rules.

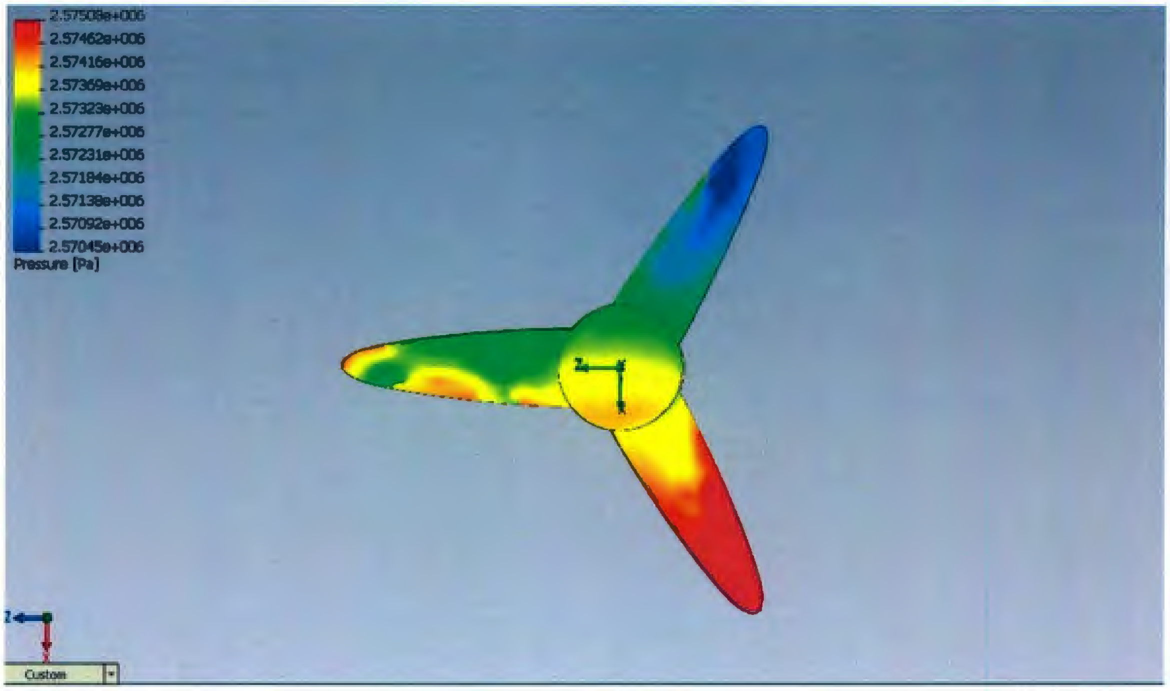

Figure 32: Surface Total Pressure Distribution of a 3-Blade Propeller

Figure 32 displays the total surface pressure, which includes static pressure and dynamic pressure. Static pressure is affected by water depth, and the equation is $p=\rho gh$. Propeller thrust is provided by the difference pressure between both sides of blades. Static pressure on the two sides of blades is the same, so it doesn't provide thrust. Dynamic pressure is the difference pressure on propeller blades to provide thrust. Figure 33 displays the dynamic pressure distribution of the propeller.

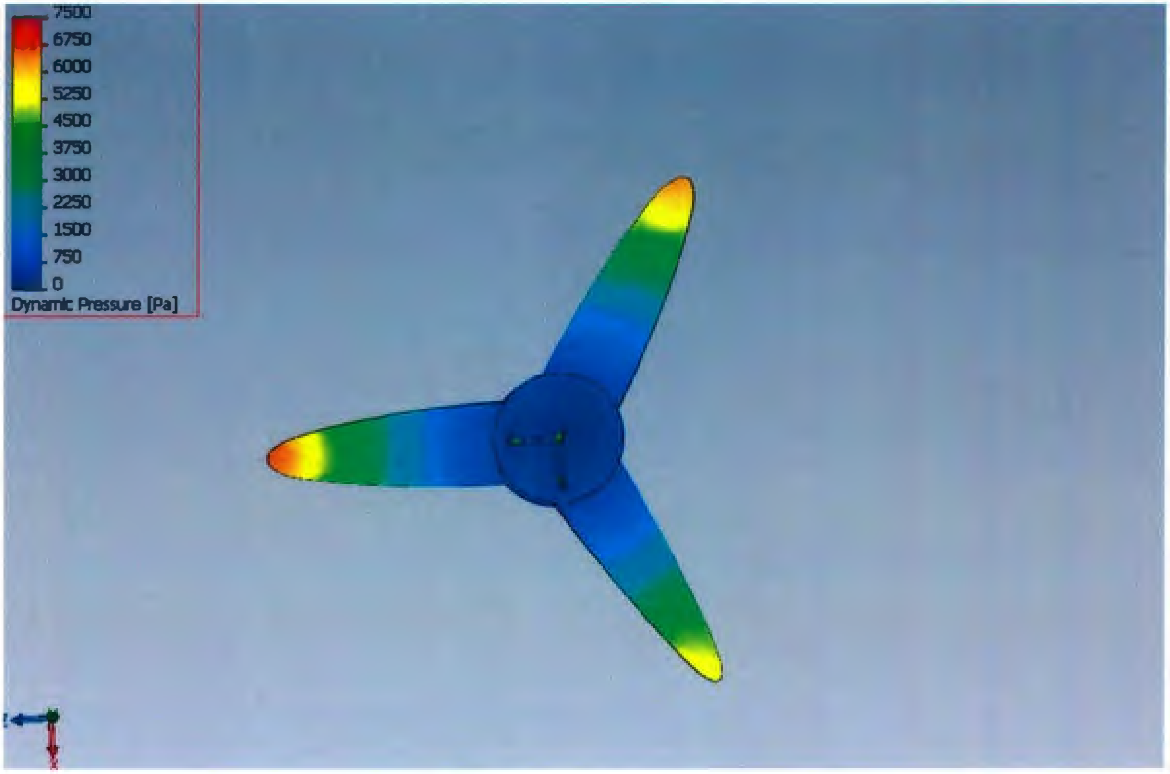

Figure 33: Dynamic Pressure Distribution of a 3-Blade Propeller

The total pressure distribution is the dynamic pressure (symmetrical distribution) combined with static pressure (changed by water depth). CosmosWorks use the surface pressure result as an FEA input parameter to analysis the propeller's strength. The pressure result needs to be loaded in the flow condition first, and then bonded and centrifugal force features are considered in this FEA. After finishing settings, run the FEA simulation and Figure 34 displays the stress results of the propeller.

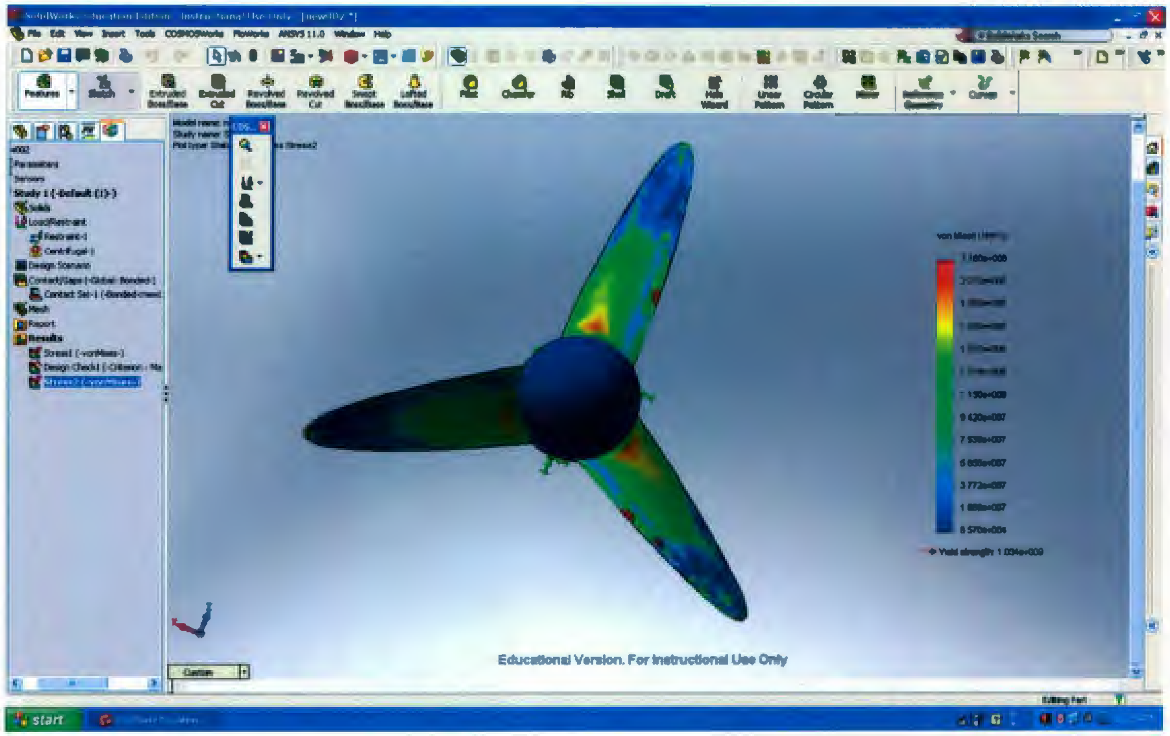

Figure34: Stress Distribution of the 3-Blade Propeller

In Figure 34, the areas near the hub has the higher stress, and these areas are needed to be seriously considered to confirm the propeller is strong enough that it will not be broken when it is rotating. The FOS graph can easily display the safety property. If FOS is not less than 1.5, it means the propeller is strong enough. Figure 35 shows the FOS of the propeller.

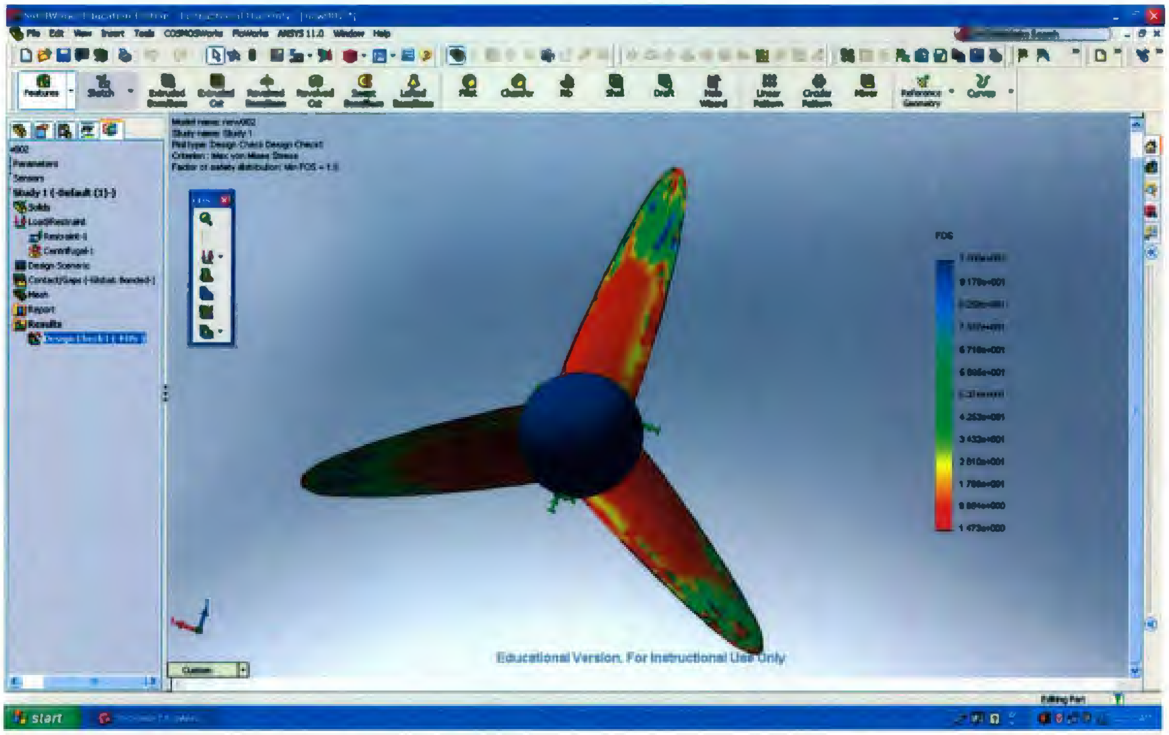

Figure 35: FOS of the 3-Blade Propeller

In Figure 35, the minimum FOS is 1.5 that displays at the left top corner of the FOS plot. That means all of the values of FOS are not less than 1.5, which means that this propeller with the Titanium Alloy material is strong enough. There are two main methods to change the strength of propeller design: One is to changing the material of propeller designs; another method is change the geometry of the propeller design. Studying from Figure 34, the connections of hub and propeller blades have the highest stress and can be broken with high probability. A good hub design can reduce the stress at the connections, such as adding much more material at the connections. Fillets can be added at the base of blades to increase the strength. Figure 36 dispalys a fillet at one root of a propeller blade. Figure 37 is the stress distribution that only adds fillets for this 3-blade propeller.

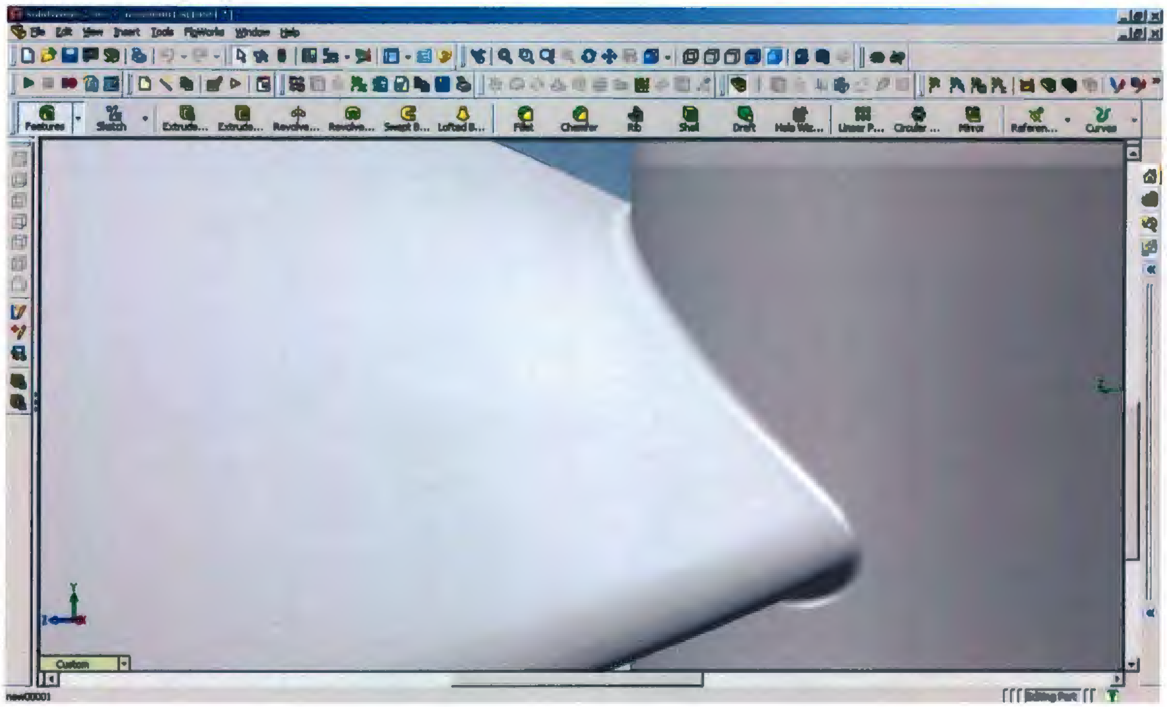

Figure 36: Fillets at the Root of a Propeller Blade

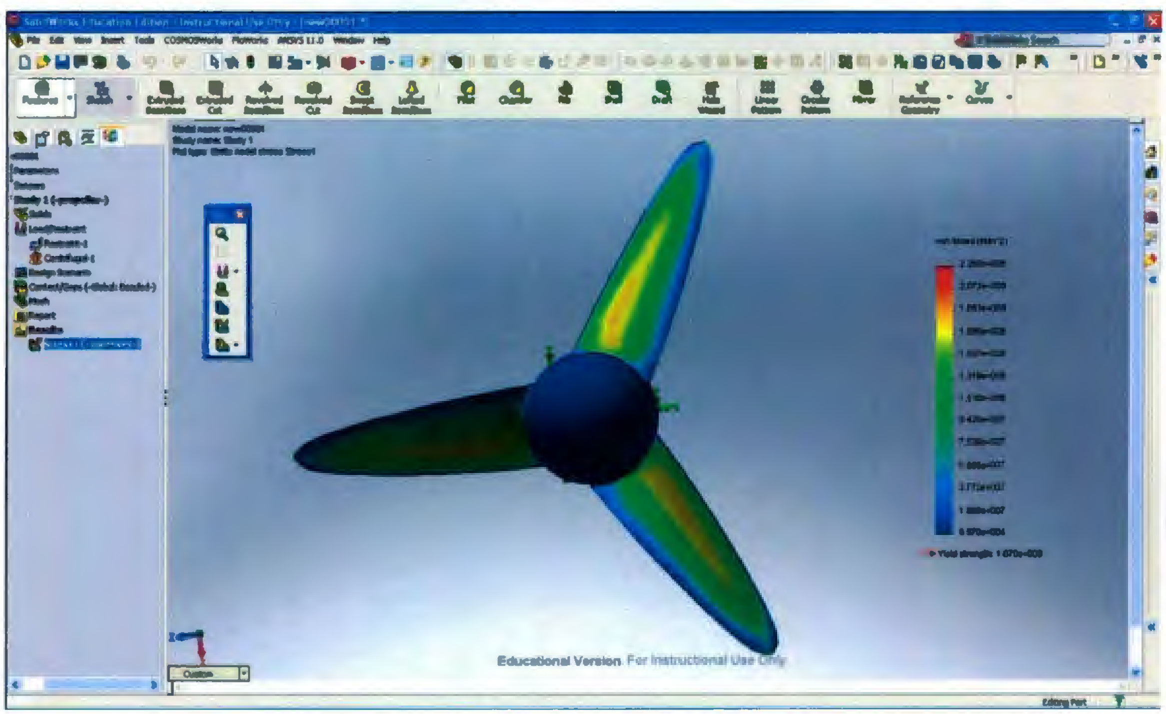

Figure 37: Stress Distribution with Fillets of the 3-Blade propeller

Figure 37 uses the same colore setting as the Figure 34. From Figure 37, the stress on the propelelr is decreased with adding fillets. This result can be easily displayed with FOS distribution, which is shown as Figure 38. From the left top of the FOS plot, the minimum FOS value with fillets is 5.1. The minimum FOS value without fillets is 1.5. That means adding fillets at base of blades can increase the strength of a propeller design.

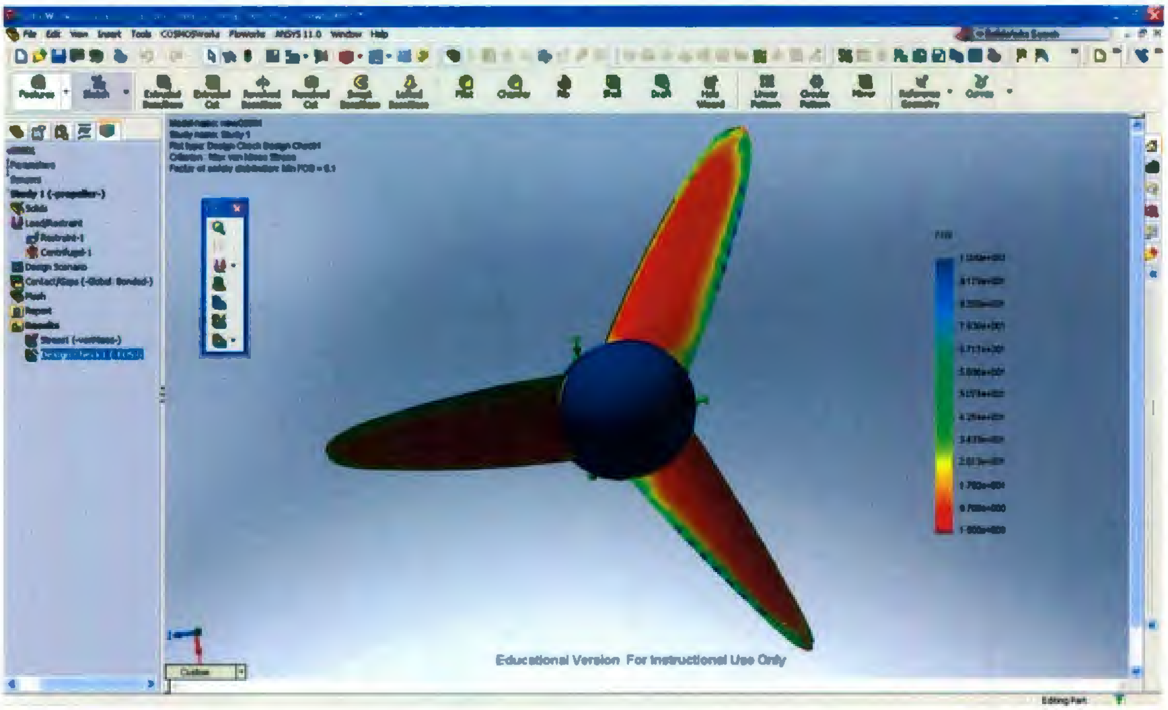

Figure 38: FOS Distribution with Fillets of the 3-Blade Propeller

Figure 39 is the flow chart of the propeller simulation process.

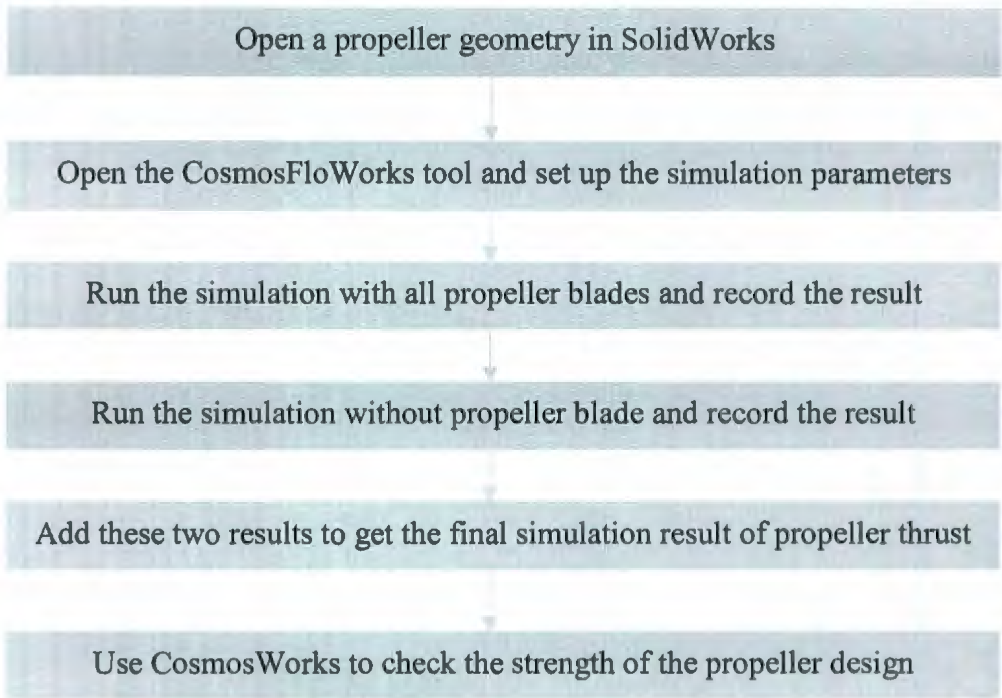

Figure 39: Flow Chart of a Propeller Simulation Process

In Chapter 3, the method of generating a propeller foil section, a leading edge line and a propeller blade is presented. Several blades are joined to a hub to create a full propeller model. Based on the real propeller settings, a simulation model is created by the simulation package CosmosFloWorks, which can simulate the rotation speed and the motion of a propeller, and then to predict the thrust of the propeller. CosmosFloWorks can also calculate the surface pressure of propeller blades, which can be used as the input for FEA analysis application "CosmosWorks" to check the strength of the propeller design.

Chapter 4

Case Study: Simulation of an AUV Propeller

As introduced in chapter 2, D'Epagnier designed an AUV propeller by OpenPVL code. In this chapter, the OpenPVL_SW code was run with the same parameters as in D'Epagnier's study, and generated a propeller geometry file for SolidWorks. Because OpenPVL and OpenPVL_SW have the same calculation methods of propeller design, the propeller geometry is the same. The main purpose of this case study is to verify that the OpenPVL_SW code is not only capable of generating a propeller geometry file for SolidWorks, but also the CosmosFloWorks can achieve reasonable simulation results for predicting propeller thrust. After running the propeller design function with the propeller parameters in OpenPVL_SW, the OpenProp_Solidworks.txt file was created to generate the propeller blade in SolidWorks. This file records the macro to automatically draw the propeller blade geometry in Solidworks. An abbreviated version of the macro is shown in Appendix B. The propeller blade geometry is shown in Figure 40.

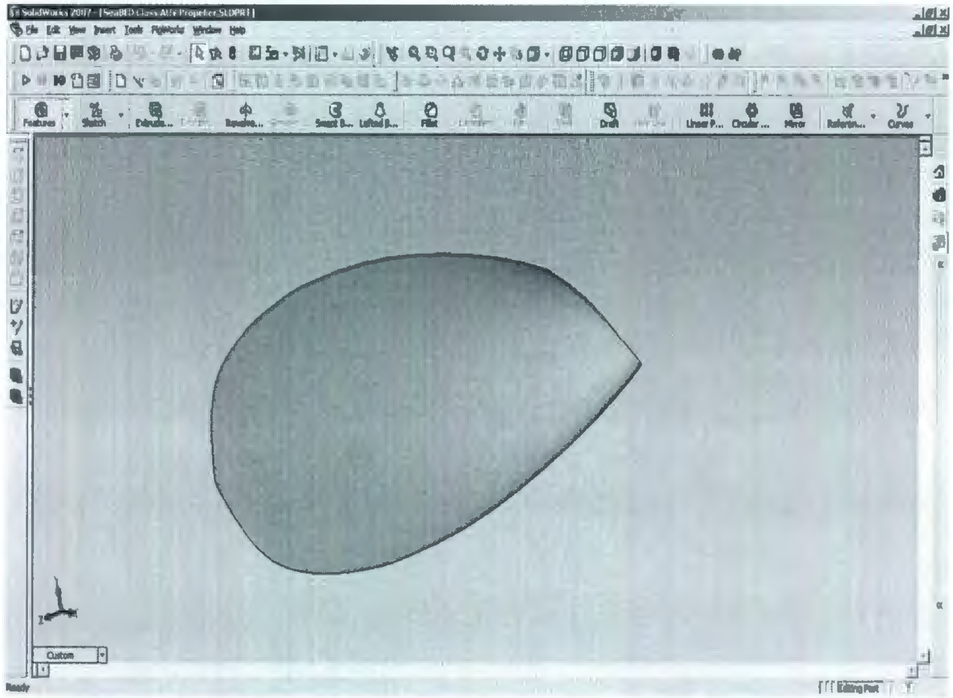

Figure 40: AUV Propeller Blade Geometry

After a propeller blade is created, a hub needs to be designe. Following the parameters of the AUV Propeller, the diameter of the hub is 0.12192 m. In this case, a simple nose cone with 0.06m height is provided to be the propeller hub. The nose cone is used to reduce the inflow force when the propeller will be worked on underwater vehicles. The AUV Propeller has three blades; therefore, another two blades will be added manually by the SolidWorks function “Circular Pattern”. The final propeller is shown in Figure 41.

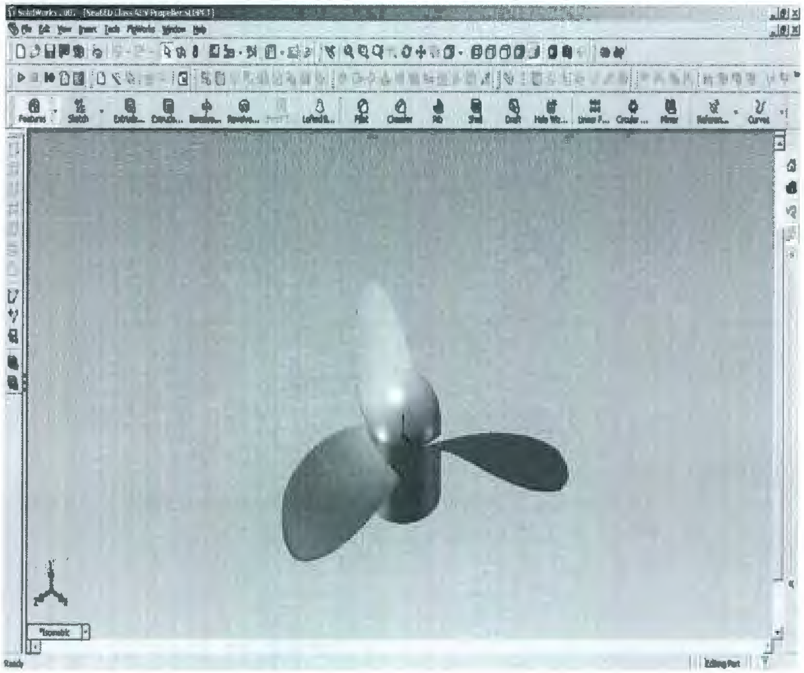

Figure 41: Final AUV Propeller

After the propeller geometry is created in SolidWorks, CosmosFloworks is used for the propeller simulation. In this simulation, the settings are as listed:

- Because the inflow wake velocity is too small compared with the ship velocity, the simulation ignores the inflow velocity.
- The propeller is set to immovable, and the fluid is uniform and moves at 1 m/s backwards to simulate the ship motion.
- The rotational speed of the propeller is 120 RPM.
- Gravity feature is considered, $g=9.81 m/s^2$.
- Water is used in this simulation.
- The simulation tank is $10m \times 10m \times 50m$ (width×depth×length)
- The temperature of water is 293.2 K.
- The air pressure is 101325 Pa.

- Assume the roughness is 0 micrometers.
- Pressure potential is considered.
- No cavitation is in the simulation
- Adiabatic wall is used.

The Figure 42 shows the simulated force result for the propeller with all three blades. The simulation results are based on several calculation cycles. At the beginning, the simulation has not enough calculation cycles, the results has large error, which shows as the plot before 10 iterations. After several iterations, the results are much accurate. The current value is the result after many cycles, so it well reflects the final simulation result. The current value as shown in Figure 42 is 78.8 N.

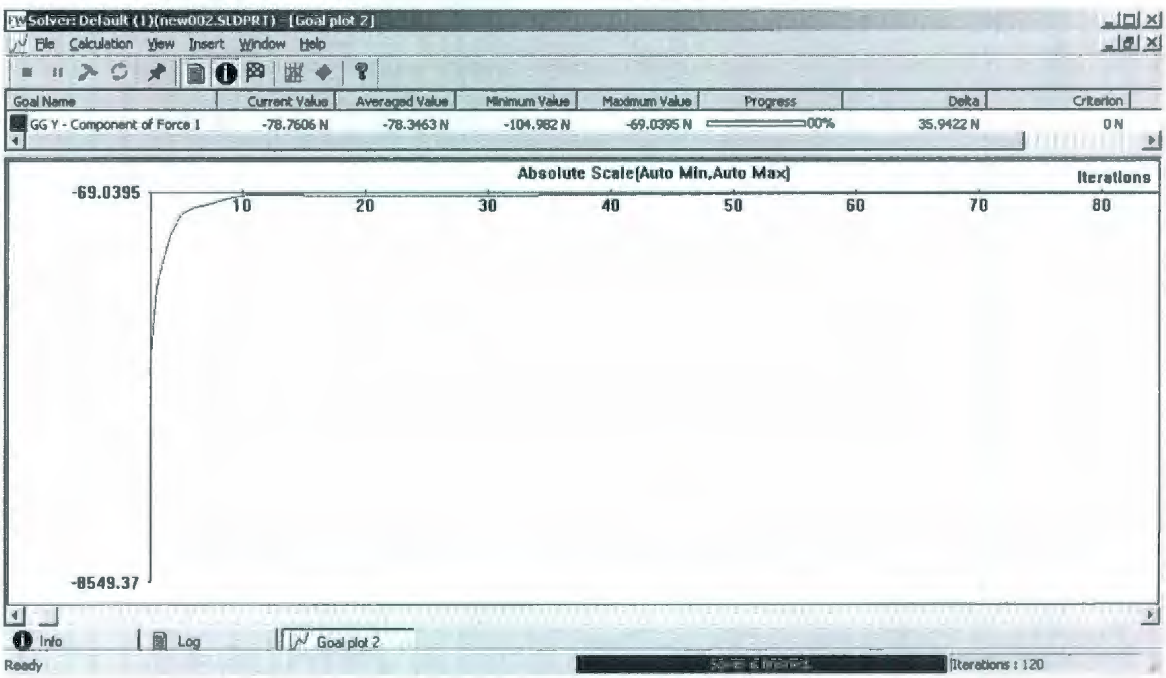

Figure 42: Predicted Result of Propeller Thrust With Blades

As introduced in chapter 3, another simulation, with no propeller blade, needs to be done to consider the force effect from the water inflow on the propeller. Figure 43 displays the simulation result with no propeller blade.

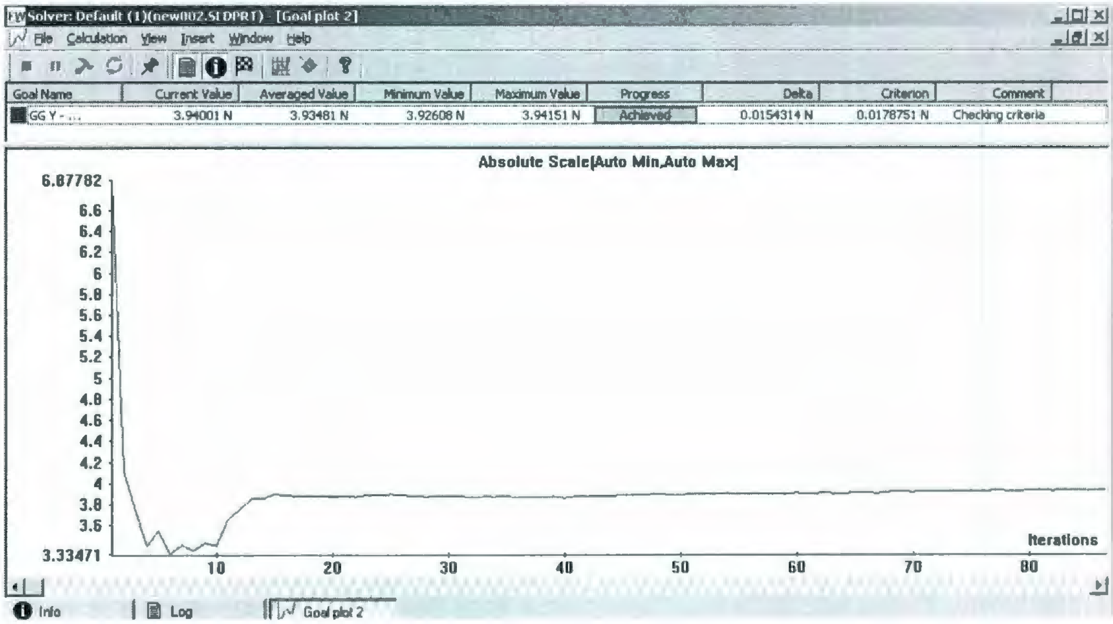

Figure 43: Predicted Result of Propeller Thrust Without Blade

The current value for this no blade simulation is 3.9N. Thus, the propeller thrust from this simulation is the sum of these two results, 82.7 N. The designed thrust of this propeller is 75N. There is only approximate 10% error between the designed and simulation result, however; the simulation result is quite accurate for a prediction level.

Figure 44 displays the relation between RPM and thrust of the propeller at the vessel speed 1m/s. The plot of RPM vs. thrust can be used for propeller design to avoid cavitation and sympathetic vibration. In Figure 44, the change of thrust trends to be small when the RPM is from large to low.

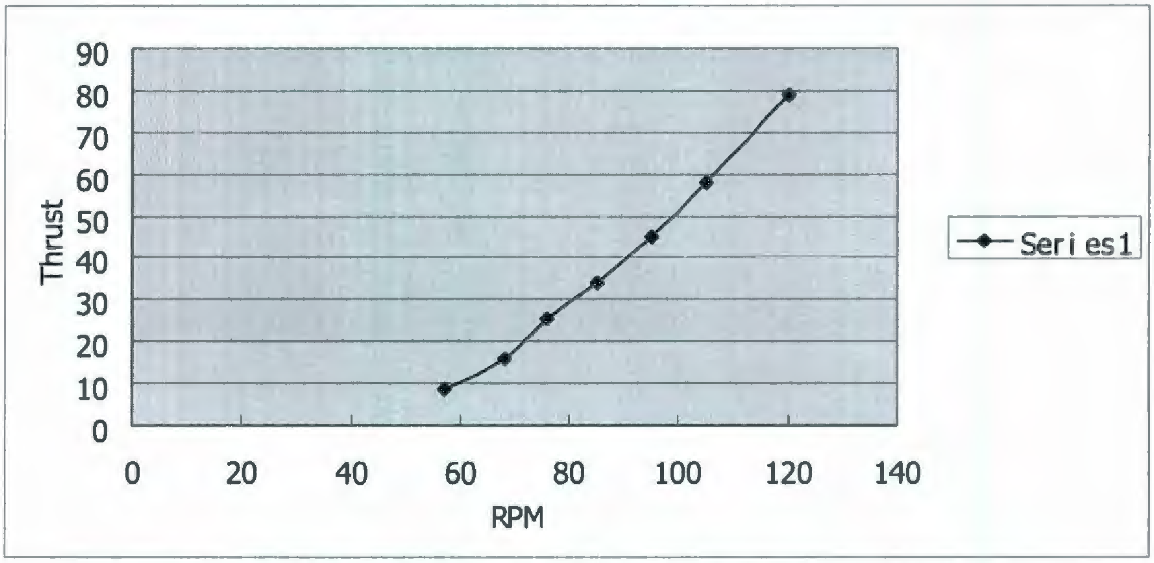

Figure 44: Relation of RPM and thrust at vessel speed 1m/s

In 2009, Dr. Vural finished several experiments to test several APC propellers and plot the graph of RPM and thrust for each propeller. The testings are finished by an electric motor, which is mounted on an RC engine that is mounted on a polycarbonate board. Load cells are mounted on the polycarbonate board to test thrust [39]. The graph of RPM and thrust from Dr. Vural's experiments is shown in figure 45.

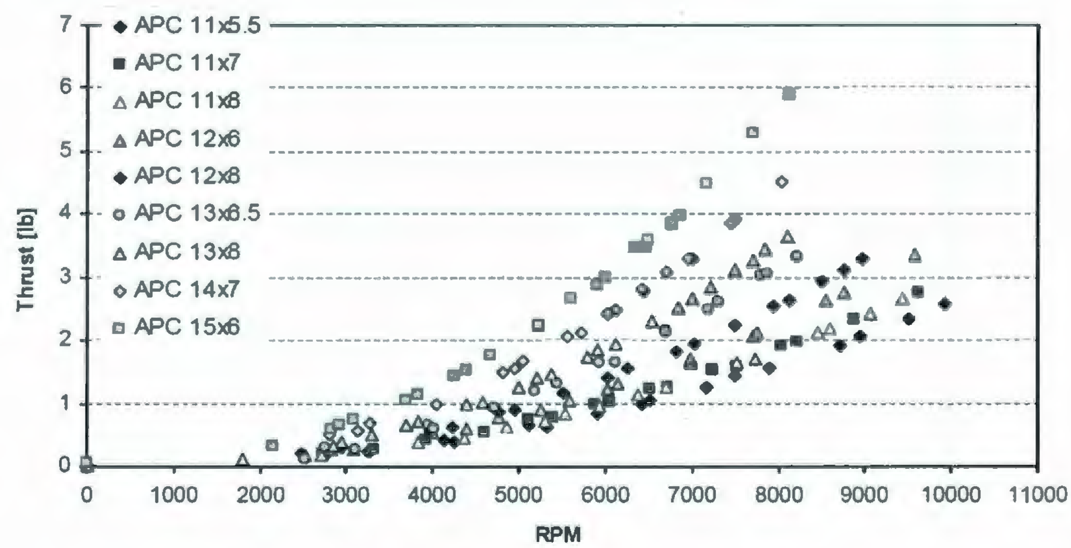

Figure 45: Plot of RPM and Thrust from Dr. Vural's Experiments [39]

The plots of RPM and thrust from our simulations and Dr. Vural's experiements have a similar trend, which prove that our simulation results for the relation of RPM and thrust are reasonable. The plot of RPM and thrust can calculate the RPM when the required thrust at certain vessel speed is known. RPM is a main factor of cavitation, which will destroy propeller and cause noise and vibration; low RPM is used to avoid cavitation. RPM is also a factor of sympathetic vibration between propeller and vessel. Certain RPM of propeller has its own frequency. If this frequency is equal to the natural frequency of vessel, sympathetic vibration, which can destroy propeller and vessel, will be created. If the thrust is known, this plot can provide the RPM information for designers to avoid cavitation and sympathetic vibration.

Another two propellers were developed using OpenPVL_SW with the designed thrust of 55N and 95N (other parameters are the same as the AUV propeller). Simulations were used for these two propellers to confirm CosmosFloWorks can provide a good predicted thrust result for propellers. Table 6 displays the thrust simulation results of the two propellers.

Table 6: Simulation Results of the Three Propellers

	Designed Thrust	Simulated Thrust	Error %	Simulation Domain Size (width×depth× length)
Propeller 1	55N	56.3N	2.38%	8m×8m×30m
Propeller 2	75N	82.7 N	10%	10m×10m×50m
Propeller 3	95N	99.5N	4.77%	15m×15m×70m

After the CFD simulation, CosmosWorks is used to check the strength of the propeller. Figure 46 displays the surface dynamic pressure on the AUV propeller blades with 75N-designed thrust. The dynamic pressure is the source to provide the thrust for a propeller design. Each blade provides the same force, that's why the distribution is symmetrical.

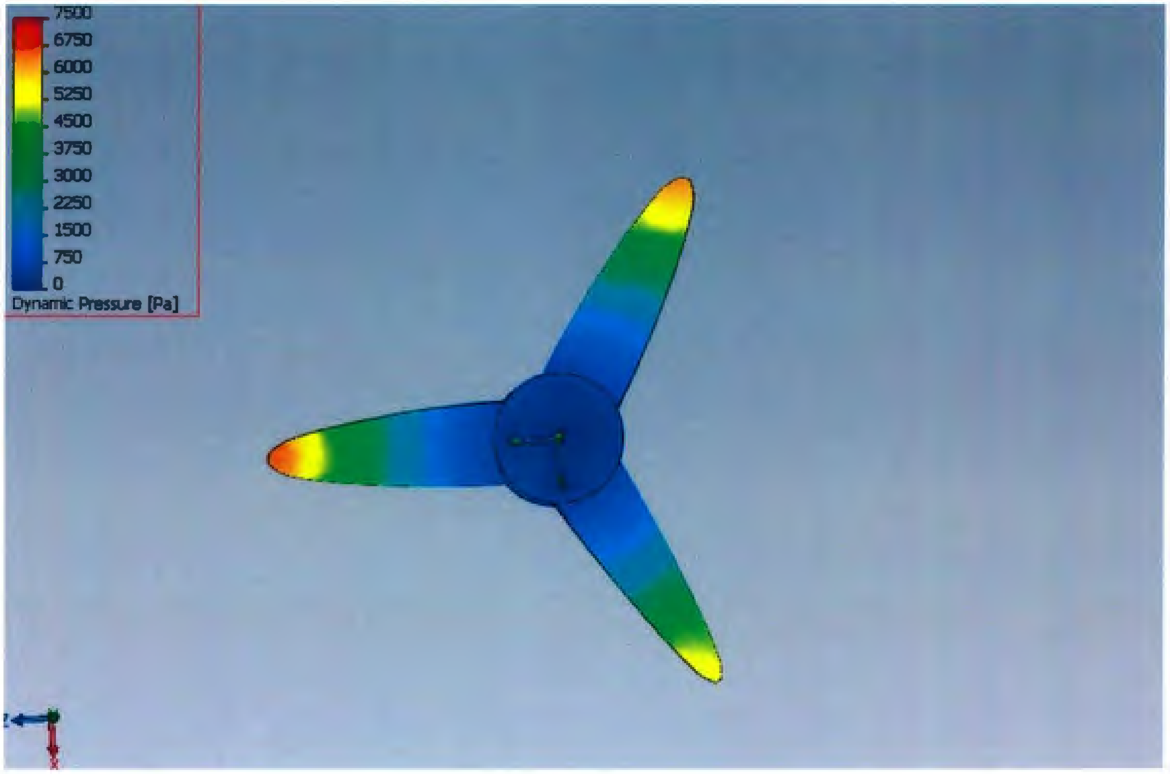

Figure 46: Surface Dynamic Pressure Distribution of the AUV Propeller

As stated in Chapter 3, CosmosWorks uses the surface pressure results as an input to check the strength of the propeller design. In this case, the Titanium Alloy was used as the propeller's material with yield strength 1.03×10^9 N/m^2. Other materials can also be used in the propeller design. The Titanium Alloy was only used as an example. The rule to choose a material is to make sure the propeller design has strong enough strength, and to be cheap as possible. Figure 47 displays the stress results of the propeller.

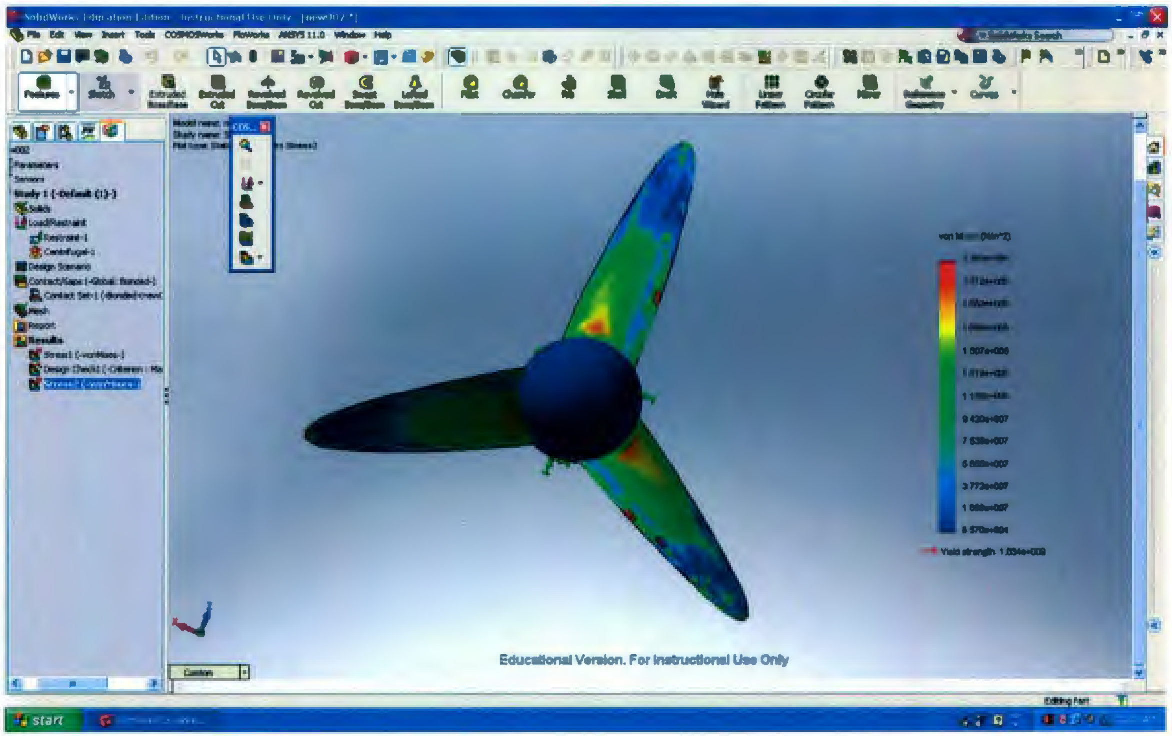

Figure 47: Stress Distribution of the AUV Propeller

In Figure 47, the areas near the hub have the higher stress, and these areas are needed to be seriously considered to make sure the propeller is strong enough. Stress on the propeller will ultimately cause deflections, which is also important when considering a propeller's acceptability. High deflection on a propeller will change the flow around the propeller and loads on it. Low deflection is desirable for a propeller design. CosmosWorks can output the displacements related to the stresses generated by the fluid/blade interaction. Deflections were briefly reviewed and seen to be minimal but the simulation values for stress and deflection should be studied in more detail in a further study. The analytical results should be validated through experimental work, which has been recommended in the conclusions of this thesis. Figure 48 displays the deflection of the AUV propeller.

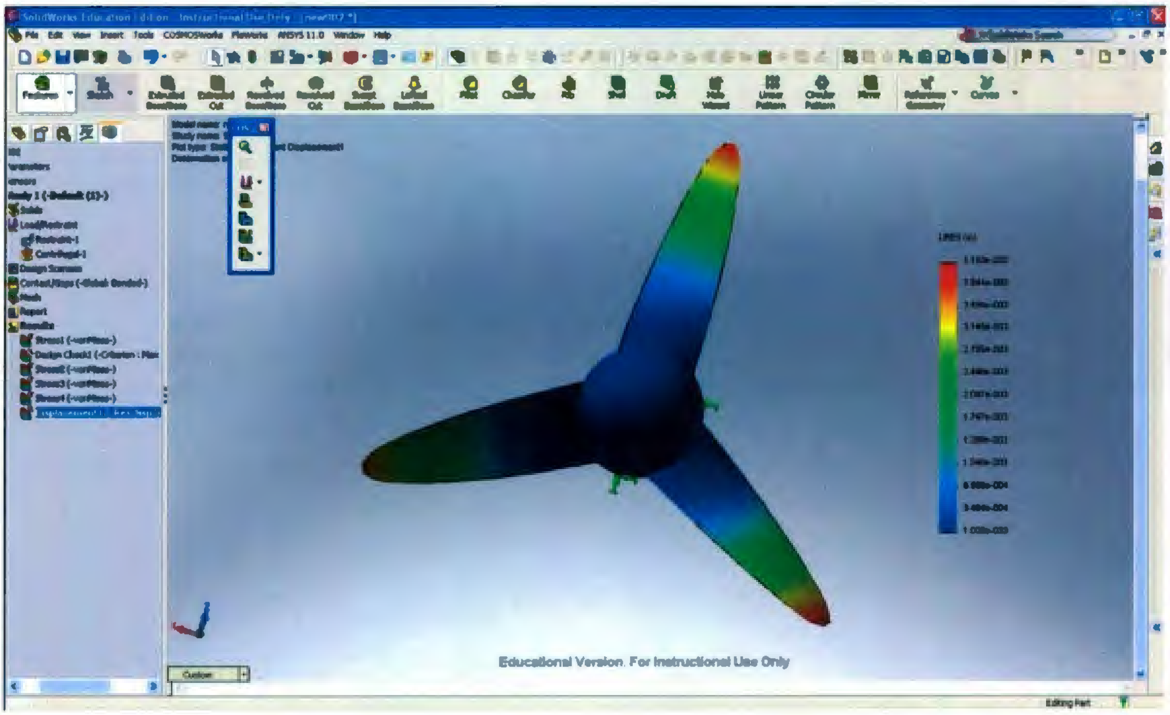

Figure 48: Deflection of the AUV Propeller

From figure 48, the highest deflection is at the top of the propeller blades, and the value is 0.0042 m. This deflection value is acceptable for a high propeller strength property.

Figure 49 shows the FOS of the propeller.

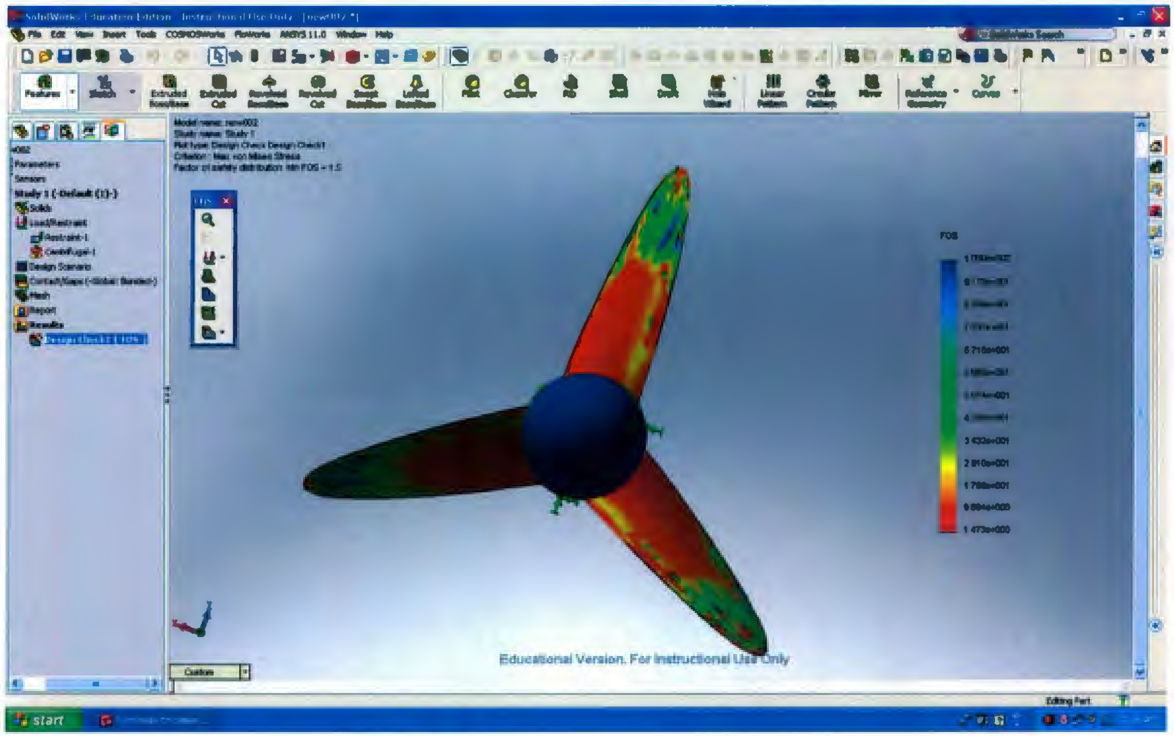

Figure 49: FOS of the AUV Propeller

In Figure 49, all of the values of FOS are bigger than 1.5, which means that this AUV propeller with the Titanium Alloy material is strong enough.

Chapter 4 outlines the way to design an AUV Propeller using OpenPVL_SW and SolidWorks. The exercise demonstrates that SolidWorks is not only able to support the design of a desired propeller, but also that CosmosFloworks can be used to predict the thrust result of a propeller and generate a pressure profile. CosmosWorks can then use the pressure profile to analysis propeller strength by FEA to ensure the propeller design is strong enough when it works. If the strength is weak, redesigning the propeller geometry or changing other higher strength material can be used to increase the propeller strength.

Chapter 5

Case Study: Simulation of a Marine Propeller Provided by Oceanic Consulting Corporation

Oceanic Consulting Corporation is a company, which provides marine related services about design evaluation and testing. In this case study, Oceanic Consulting Corporation provided a propeller geometry and tested thrust result for our simulation validation using CosmosFloWorks. In a propeller testing, the results may be affected by many factors. CosmosFloworks cannot simulate all of these factors. That is the main reason why there are some differences between the testing results and simulation results. The main purpose of this case study is to verify CosmosFloworks can provide a believable result of propeller thrust. A known propeller geometry was provided by Oceanic Consulting Corporation, as one of doing the simulation validation. The company had already determined the experimental values of the propeller thrust. The validation would come by seeing if the simulation value is close to the experimental one. This would mean that CosmosFloworks is a feasible tool for predicting propeller thrust.

The propeller has four blades, 0.15m diameter, rotates at 900 RPM and was tested in 0.5 m depth tank. The ship velocity is 1 m/s. The propeller geometry is provided by Oceanic Consulting Corporation with a SolidWorks file, which is shown in Figure 50. Figure 51 is the side view of the propeller geometry used in this simulation.

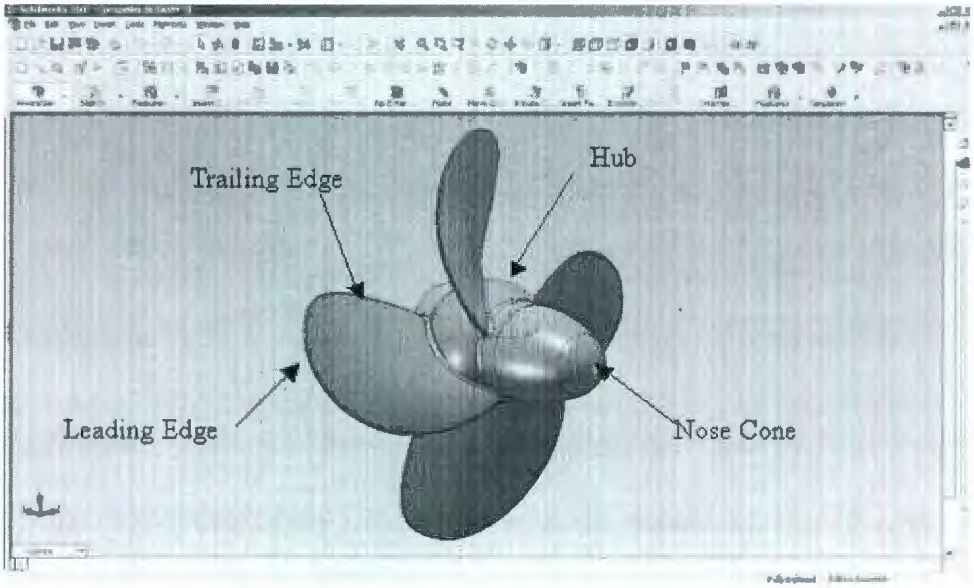

Figure 50: Propeller Geometry-4 Blades

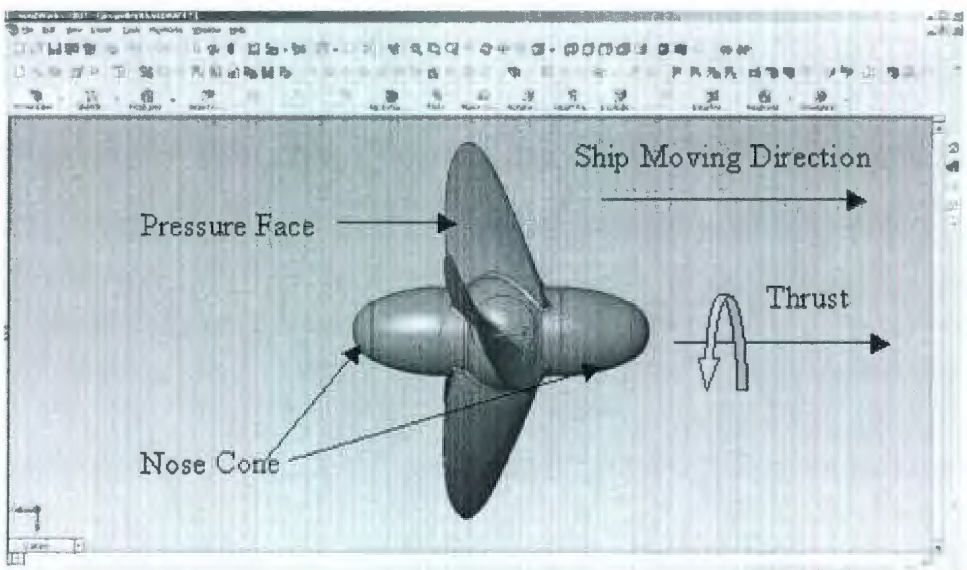

Figure 51: Sideview of the Propeller-4 Blades

In this simulation, two nose cones are added to the propeller in order to reduce the inflow force on the propeller. As introduced in chapter 3, the inflow force on the propeller is the main error between the simulation and experimental testing results. Thus, deceasing the inflow force is a way to increase the accuracy of the simulation results. In order to fast simulation, designers can use two nose cones simulation method to predict propeller thrust without using the no-blade method that is introduced in Chapter 3. It is only a method for rapid simulation, however; the simulation method as introduced in Chapter 3 is preferred. The rotation direction of the propeller is shown in Figure 51. The propeller is set to be immovable, and the fluid moves at 1 m/s in the direction shown in Figure 51 to simulate the ship motion. When the propeller is rotating, water will create force on pressure faces, and provide thrust for ship movement.

The settings of this simulation are listed as below:

- The simulation tank is 1m×1m×10m (width×depth×length)
- The propeller is immovable, and the fluid moves at 1 m/s backwards.
- The propeller is operated at 900 rpm.
- The depth of propeller center in water is 0.5 m.
- The temperature of water is 293.2 K.
- The air pressure is 101325 Pa.
- Gravity feature is considered, $g=9.81m/s^2$
- The roughness of the propeller surface is 0.3 micrometer.
- Pressure potential is considered.

- Assume no cavitation is in the simulation
- Assume adiabatic wall is used.

Figure 52 displays the result of the propeller thrust with all blades. The force is 29.0 N.

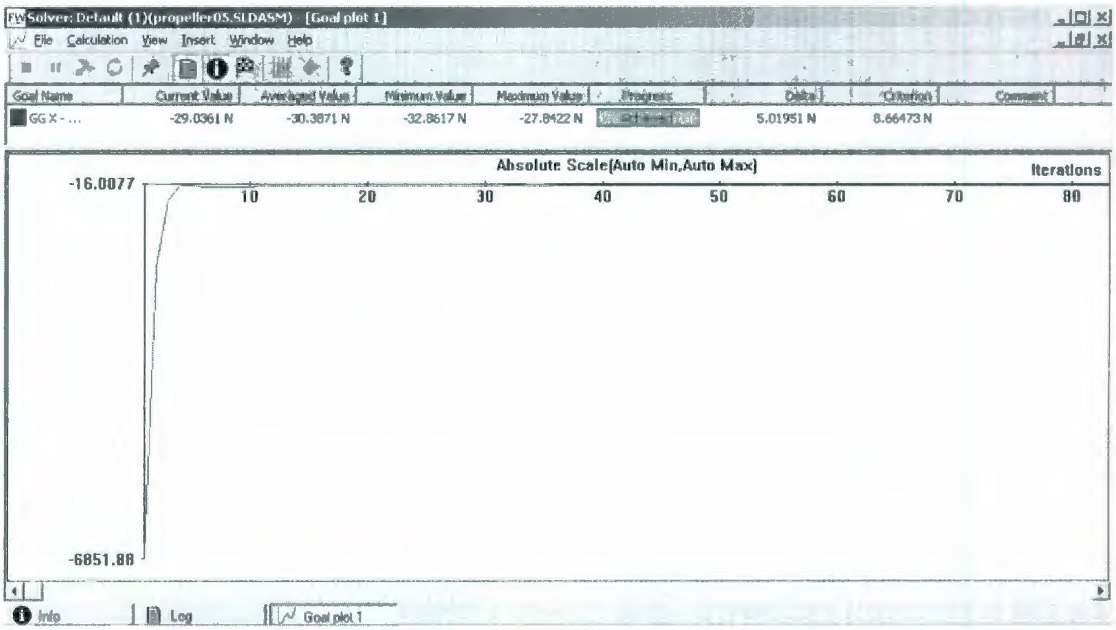

Figure 52: Predicted Propeller Thrust – 4 Blades

As introduced in chapter 3, another simulation needs to be done to consider the force effect of the water inflow on the propeller. Figure 53 displays the propeller geometry with no blades. When water is flowing, forces are created on the nose cone 1. Because the geometry of nose cone 1 is symmetrical, the summation force of these multidirectional forces is the opposite direction of the thrust of the propeller.

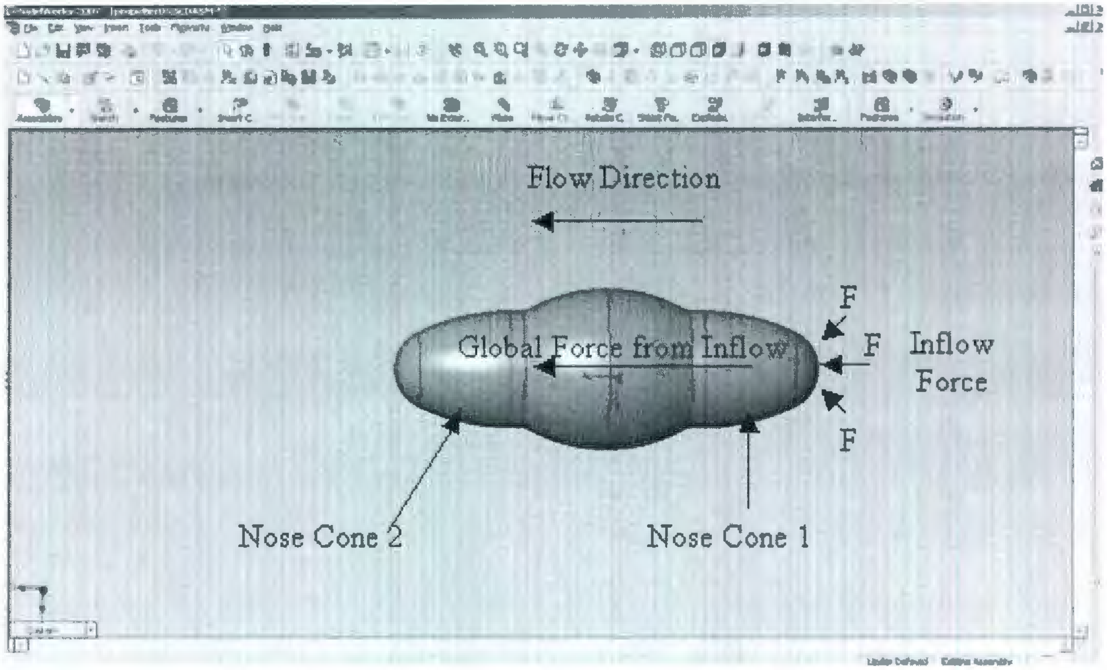

Figure 53: Propeller Geometry without blade

In this simulation without blades, all the parameter settings are the same as the simulation with blades. Figure 54 shows the simulation results without blade. The force is 0.2 N.

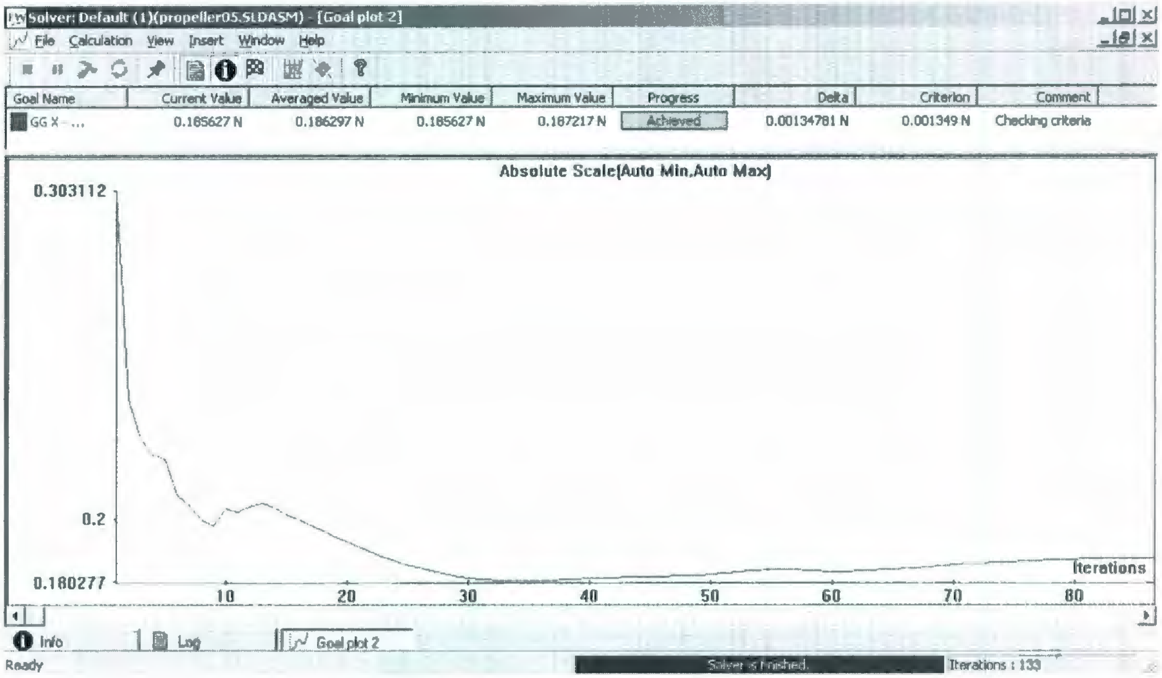

Figure 54: Predicted Propeller Thrust for the Propeller without Blade

The force from the simulation with all blades is 29.0 N, and the force from the simulation with no blade is 0.2 N. Thus, the propeller thrust from this simulation is the sum of these two results, 29.2 N. The real testing result of this propeller from Oceanic Consulting Corporation is 34.9N. There is approximate 16% error between the real and simulation results. The result has 16% error, however, it is quite accurate for a prediction level.

As introduced in Chapter 3, CosmosFloWorks can provide the surface pressure distribution to be as an input of FEA to check the strength of a propeller design. Figure 55 displays the dynamic pressure distribution of the propeller. The dynamic pressure is the source to provide the thrust for a propeller design. Each blade provides the same force, that's why the distribution is symmetrical.

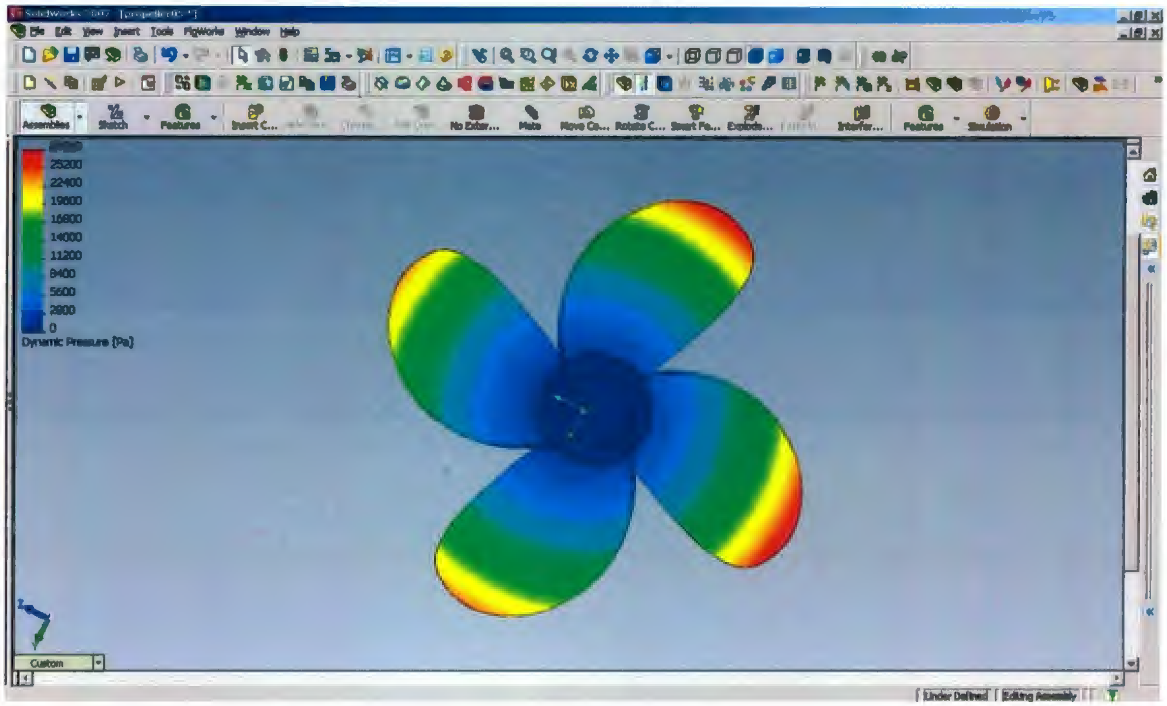

Figure 55: Surface Dynamic Pressure of the Propeller-4 Blades

Figure 56 displays the stress distribution results in the FEA analysis.

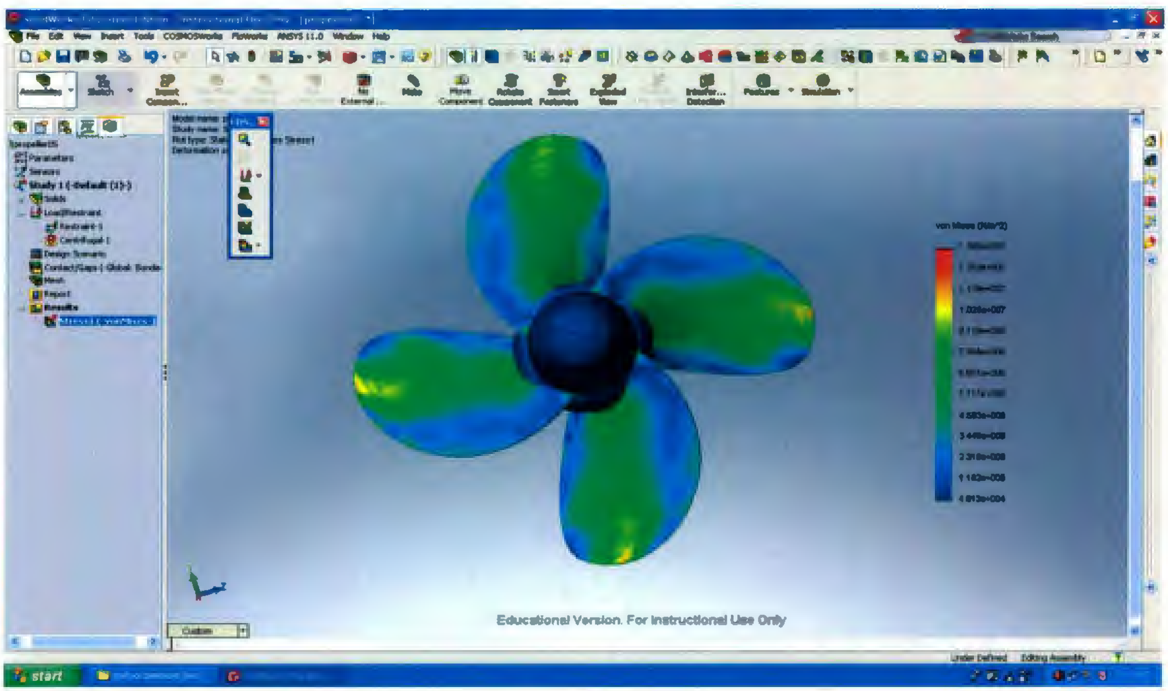

Figure 56: Stress Distribution of the Propeller-4 Blades

In the Figure 56, it can be seen that the higher stresses occur at the tips of the blades. In this propeller design, the hub has an elliptical geometry to reduce the flow force and much more material is applied between the hub and blades can provide higher strength potential. Figure 57 displays the deflection of the 4-blade propeller.

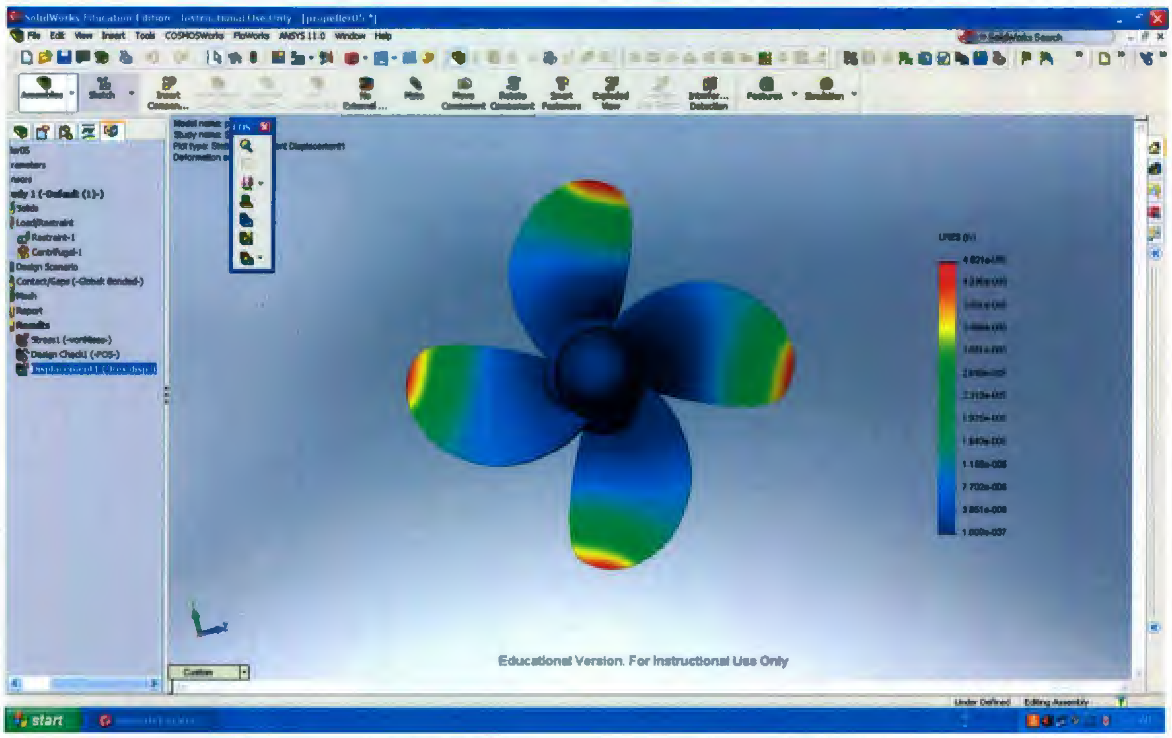

Figure 57: Deflection of the Propeller-4 Blades

In figure 57, the highest deflection is at the tip of each propeller blade, and the value is 0.000046 m. This deflection value is small enough that it is not a concern from a functional viewpoint.

FOS results are plotted to make sure this propeller design is strong enough. In this FEA analysis, Titanium Alloy was used as an example to be as the propeller's material. Figure 58 displays the FOS results of the propeller.

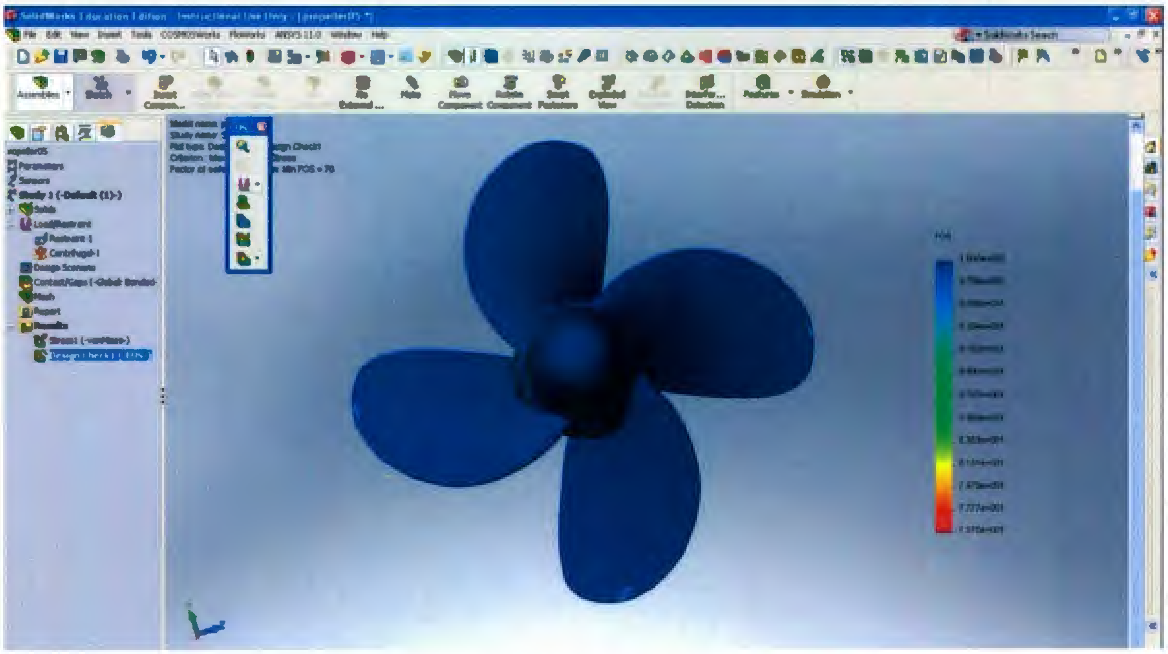

Figure 58: FOS of the Propeller-4 Blades

In this FOS results, all the values are bigger than 1.5, which means the propeller is strong enough with the Titanium Alloy material when it is rotating.

In this simulation study, the propeller geometry was provided by Oceanic Consulting Corporation. However, engineers often need to design based on a mission profile, and then to do a simulation. OpenPVL_SW can be used to design a propeller, create the propeller geometry in SolidWorks, simulate the propeller thrust in CosmosFloWorks, and check the strength of the propeller by CosmosWorks. Figure 59 is the flow chart of the procedures of a propeller design and simulation using OpenPVL_SW, SolidWorks, CosmosFloWorks and CosmosWorks.

Design a propeller blade using OpenPVL_SW:

- Parameter analysis
- Generate foil sections points using OpenPVL_SW
- Generate a propeller blade using the OpenPVL_Solidworks.txt file

Design a hub and add other blades to complete the propeller geometry in SolidWorks

CFD simulation:

- Open the CosmosFloWorks and set up the simulation parameters
- Run the simulation with all propeller blades and record the result
- Run the simulation without propeller blade and record the result
- Add these two results to get the final simulation result of propeller thrust

If the thrust is desired as designed, then go to the FEA for the strength analysis; if the thrust is not the desired as designed, then go back to redesign the propeller parameter or the hub.

FEA simulation:

- Open the CosmosWorks and create a static study
- Import the pressure result from CosmosFloWorks into the static study
- Select propeller material
- Set bonded and centrifugal features for the propeller
- Check the strength of the propeller by the stress distribution and FOS results

If the propeller has high strength, then be ready for fabrication to test; if the propeller has low strength, then go to redesign the hub or use higher strength property material for the propeller.

Figure 59: Flow Chart of the Procedures of a Propeller Design and Simulation using OpenPVL_SW, SolidWorks, CosmosFloWorks and CosmosWorks

Chapter 6

Propeller Fabrication by Rapid Prototyping

Several design and simulation iterations may be required before the designer is satisfied that the propeller geometry is appropriate. Once satisfied, the designer will need to fabricate a working prototype for testing. Rapid prototyping (RP) is the recommended approval to minimize the time and cost to arrive at a suitable prototype. As introduced in chapter 2, there are six rapid prototyping technologies are introduced. All of these technologies can be used to produce a propeller. The main differences between RP each technology are the materials to manufacture parts and the machine accuacy. Selection of a suitable RP technology for a propeller depends on the accuracy and the strength required for the specific working environment. Stratasys Fusion Deposition Modeling (FDM) 2000 is used in this thesis for propeller fabrication due to its accessibility at Memorial University. As stated the introduction in chapter 2, FDM technology is fast, at low cost and quite accurate for a testing level; however, the materials that can be used in FDM are limited with ABS, PC and wax. These materials have poor strength property, which limits the strength of FDM prototypes.

The Stratasys FDM 2000 RP process requires the input of an .STL file, which is a format used by Stereolithography software to generate information need to produce 3D models on Stereolithography machines. Specifications of the FDM 2000 used in this study are listed in Table 7, and the FDM 2000 is shown in Figure 60.

Table 7: Stratasys FDM 2000 Specifications [40]

Build Size	10 ×10×10 in
Achievable Accuracy	+/- 0.005 in
Modeling Materials	Acrylonitrile Butadiene Styrene (ABS); Medical grade ABS; Methyl methacrylate ABS; Polycarbonate plastic; Investment casting wax
Layer Width	Ranges from 0.010 to 0.100 in
Layer Thickness	Ranges from 0.002 to 0.030 in
Software	.STL files are processed using Quickslice Version 6.0

Figure 60: Stratasys FDM 2000

After a propeller solid model is created in SolidWorks, the propeller geometry can be exported as an .STL file. When an .STL file is input into Quickslice Version 6.0, a model structure is automatically created for the file, shown in Figure 61.

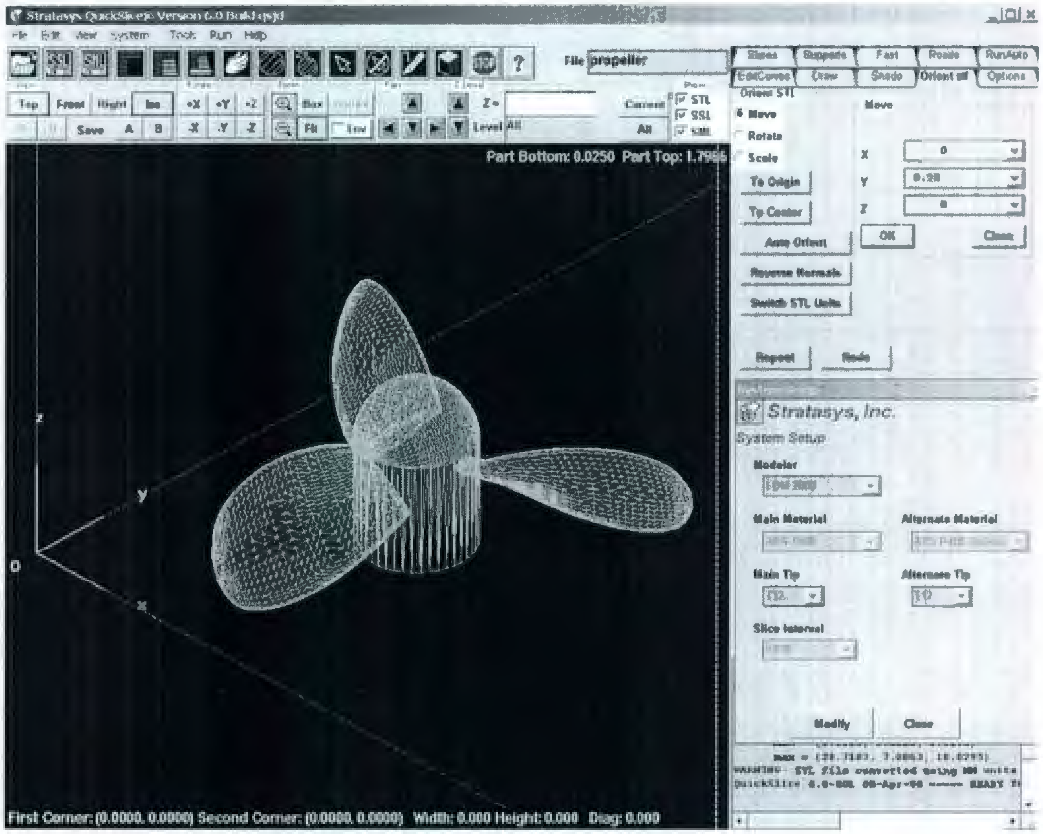

Figure 61: Screenshot of Propeller Blade Generated by Quickslice Software

The FDM printable file can generate both a total volume calculation and a completion time estimate before the production. The following lists are the steps for the fabrication using Stratasys FDM 2000.

- Start Quickslice and open the .STL file.
- Choose tips of material extrusion nozzle. Small tip can provide higher strength and smooth surface finish, however; it will take much more time and material. Because the blades of a propeller are thin and need to support high stress, small tip is preferred for propeller fabrication.
- Set up the orientation of a model in Quickslice. The two options of moving and rotating can be used for the orientation and make sure the model is in the

construction area, which is lined in the software. Because the FDM builds up models in layers along the z-axis, the orientation will affect how the final model comes out. The ideal orientation is stable, minimize support material, and place any intricate structures facing up so that support material cannot mar them. For propeller models, it is better to place the hub structure facing up, as shown in Figure 61.

- Create slices to build up models using Quickslice. The thickness of slice is from 0.002 in to 0.030 in. Low thickness can provide strong structure and good surface finish, however; it will take much more material and time.
- Create support that will allow parts to be built in it.
- Create toolpath for the fabrication. The toolpath of parallel straight lines are usually used in fabrication. The way of several curved parallel lines for the propeller blades' edges and straight lines for the middle area is suggested for propeller blades fabrication.
- Create base and then save as an .SML file, which is a solid model file that will be sent to the FDM machine
- Turn on the StrataSys FDM 2000 and warm up the machine. The required temperature for ABS is 270° (model material temperature) and 265° (support material temperature).
- Check FDM material loading to verify that the model and support materials are loaded correctly.
- Send the .SML file to Stratasys FDM 2000 to produce the part.

Material options for Stratasys Fusion Deposition Modeling (FDM) 2000 include Acrylonitrile Butadiene Styrene (ABS); Medical grade ABS; Methyl methacrylate ABS; Polycarbonate plastic (PC); Investment casting wax. Wax is too weak to produce a propeller. ABS can be used to produce a propeller, and can be easily filed down to a smooth finish using sand paper, if necessary. PC is more rigid than ABS, but it is expensive. Medical grade ABS and Methyl methacrylate ABS are the ABS/PC blend material to reduce the cost if extreme strength is not necessary, and can achieve specific purposes. Methyl methacrylate ABS is often used as transparent glass substitute. Medical grade ABS can be used for surgical instruments, diagnostic devices and drug delivery systems [34]. The material will be selected based on the cost and rigidity requirements. ABS is economical but with low rigidity. PC is the most expensive with very high rigidity. Methyl methacrylate ABS and Medical grade ABS have moderate rigid and cheaper than PC. As introduced, PC is the best material to fabricate a propeller with higher strength. Figure 62 is an example of FDM propeller prototype with the material of ABS.

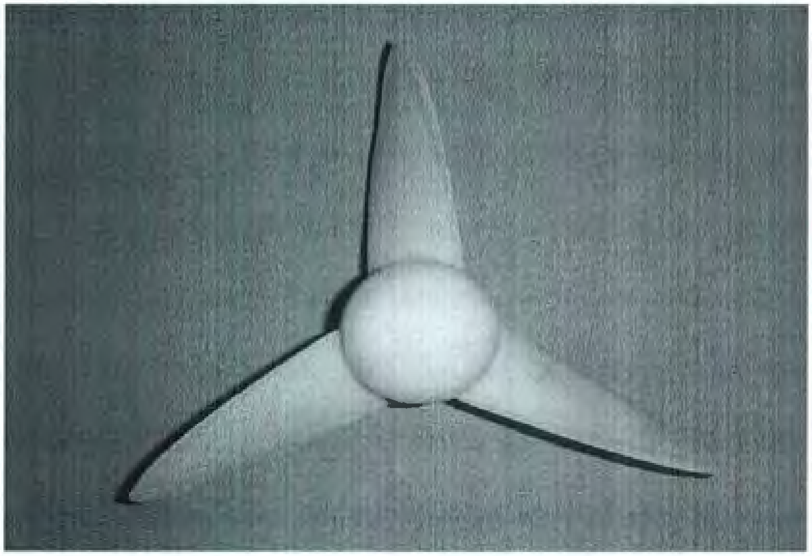

Figure 62: A Propeller Fabricated by FDM

Due to the accuracy limitation of the FDM machine, the edge and the surface of the propeller model are not very smooth. Glue can be used to cover the propeller to achieve a smooth propeller body. Epoxy is a good choice for a propeller surface coating, because epoxy can provide a thin wall around the surface and increase the strength of materials [41]. In order to avoid unequal coating thickness, it is better to coat one blade at a time and keep the blade flat when the glue is drying. When the epoxy is fully dried, sand paper can be used to make the blades much more smooth. As introduced before, PC is the strongest material to produce a strong propeller using FDM. PC coating with epoxy can provide higher strength than PC only. The detailed data for the strength of the PC coating with epoxy material will need some experiments in the future work. If the material is still not strong enough for some specific cases, the FDM propeller model can be used as a casting mould to produce a metal propeller.

In chapter 6, Stratasys Fusion Deposition Modeling (FDM) 2000 is introduced. The method of producing a propeller using Stratasys 2000 is also presented. The material that can be used in Stratasys 2000 is listed, and PC is the best material for a propeller fabrication using FDM. Coating with glue can provide smooth and higher strength property for a propeller model.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

The described rapid marine propeller design process was developed to facilitate a propeller design from simulation through to prototyping and testing processes. The OpenPVL_SW code was developed to generate the propeller geometry for automatic input into the CAD software, SolidWorks. SolidWorks includes the CFD simulation capability for propellers using the CosmosFloWorks package. This study has proven that CosmosFloworks can generate a reasonable prediction for propeller thrust. CosmosWorks can then be used to check the strength of a propeller design using FEA analysis.

The original OpenPVL code was able to generate the propeller design geometry for the CAD software, RHINO. This geometry is generated as the sets of points that are located on the foil sections of the propeller blade. Users then had to use several commands to plot out the geometry of the propeller in RHINO. The OpenPVL_SW code extended this capacity to generate the propeller geometry without any manual commands in

SolidWorks. This modification will clearly save users' time, especially when a number of design iterations need to be implemented. Solidworks allows users to design a specific propeller hub and create a 3D-printable file for rapid prototyping to produce a propeller for a model testing.

The CosmosFloworks package allows users to simulate the propeller working in a specific fluid environment through CFD. The simulation helps the designers to predict the thrust of the propeller as part of the design cycle. The CosmosWorks package will use the pressure results as an input of FEA to check the propeller strength.

7.2 Future Work

- Add propeller hub design function in the OpenPVL_SW code
- Coating with glue can increase the strength of ABS material. Future experiments can be conducted to find out the detailed data for the increased strength.
- Use SLS rapid prototyping technology to fabricate an ideal marine propeller and test its performance
- Test propeller's thrust and compared with the CFD result
- Test stress at the root of the propeller blades
- Fabricate a propeller with two methods. One is produced by CNC machining with cutting off material to left a solid part. Another one is produced by melting material layer to layer to fabricate the propeller part. Test the stress of the two propellers with different fabrication methods.
- Much more deflection analysis will be applied in the future work.

References

[1] Burtner, E., 1953, "A Relationship for Preliminary Propeller Diameter", Journal of the American Society of Naval Engineers Inc., Vol. 65, No. 3.

[2] Betz, A., 1910, "Schrauben Propeller mit Geringstem Energie Verlust", translated from German by T. Hofmann and N. Bose, "Screw Propellers with Minimum Energy Loss", 1990, Ocean Engineering Research Center, Memorial University of Newfoundland, Rep. No. OERC90-HYD-LR-02., pp. 20.

[3] Goldstein, S., 1929, "On The Vortex theory of Propellers", Proceedings of the Royal Society, Series A, Vol. 63, pp.440-465.

[4] Glauert,H., 1947, "The Elements of Airfoil and Airscrew Theory", Cambridge University.

[5] Lerbs, H.W., 1952, "Moderately Loaded Propellers with a Finite Number of Blades and an Arbitrary Distribution of Circulation", Society of Naval Architects and Marine Engineers, Vol.60, pp. 73-123.

[6] Kerwin, J.E., 2001, "Hydrofoils and Propellers", Massachusetts Institute of Technology.

[7] Eckhardt, M.K., Morgan W.B., 1955, "A propeller Design Method", Society of Naval Architects and marine Engineering, Vol. 63, pp.325-374.

[8] Erickson, L. L., 1990, "Panel Methods-An Introduction". NACA Technical Paper 2995, California

[9] Hess, J.L. and Valarezo, W.O., 1985. "Calculation of steady flow about propellers by means of surface panel method", Research and Technology Douglas Aircraft Company, Long Beach

[10] Ramachandran, P., Rajan, S. C. and Ramakrishna, M., 2003, "A Fast, Two-Dimensional Panel Method". Society for Industrial and Applied Mathematics

[11] D'Epagnier, P.K., Chung, H., Stanway, J.M., Kimball, W.R., 2007, "An Open Source Parametric Propeller Design Tool", Cambridge.

[12] Liu, P., Islam, M.F., Searle, S., 2006, "Design and Optimazation of An Ice Class propeller Under Shallow Water, Semitunnel Hull and Heavy Load Conditions", Proceeding of OMAE2006, 25th International Conference on Offshore Mechanics and Arctic Engineering, Hamburg, Germany.

[13] Muckle, W., 1941, "Stresses in Propeller Blades", The Shipbuilder and marine Engine Builder, Vol. 47, No. 388, pp.336-341.

[14] Morgan, W.B., 1954, "An Apimate Method of Obtaining Stresses in a Propeller Blade", David Taylor Model proxBasin, Report No. 919, Washington D.C., pp.18.

[15] Ghose, J. P., Gokarn, R. P., 2004, "Basic Ship propulsion", New Delhi.

[16] Hofmann, T., 1991, "Marine Propeller Design, Tests and Manufacture", Memorial University.

[17] Kerwin, J.E., 2001, "Hydrofoils and Propellers". Cambridge, MIT CopyTech.

[18] Chung, H., 2007, "An Enhanced Propeller Design Program Based on Propeller Vortex Lattice Lifting Line Theory", Master Thesis, Massachusetts Institute of Technology.

[19] Marzocca, P., 2009, "The NACA Airfoil Series". Clarkson University

[20] D'Epagnier, P.K., 2005, "A Computational Tool for the Rapid Design and Prototyping of Propellers for Underwater Vehicles". Master Thesis, Massachusetts Institute of Technology

[21] Farin, Gerald E., 1993, "Curves and surfaces for computer aided geometric design: a practical guide", Boston, Toronto

[22] "CAD softwares history", www. CADAZZ.com, Accessed 10 Jan2010.

[23] Lombard, M., 2009, "SolidWorks 2009 bible", Indiana,

[24] Anderson, J.D., 2009, "Computational fluid dynamics: an introduction". Berlin

[25] Chaskalovic, J., 2008, "Finite element methods for engineering sciences". New York

[26] Versteeg, H. K., 2007, "An introduction to computational fluid dynamics: the finite volume method". Toronto

[27] Dymarski, P., 2008, "Computations of the propeller open water characteristics using the SOLAGA computer program. Predictions of the cavitation phenomenon". Ship Design and Research Centre S.A.

[28] Nicholson, D. W., 2008, "Finite Element Analysis: Thermomechanics of Solids". Boca Raton

[29] Richard L. Van Horn, 1971, "Validation of Simulation Results". Carnegie Mellon University

[30] E. Paul DeGarmo, J. Temple Black, Ronald A. Kohser, 1988, "Materials and processes in manufacturing". New York

[31] Flinn, R.A., 1963, "Fundementals of Metal Casting", Addison Wesley Pub. Co. Inc.

[32] Ali Kamrani, Emad Abouel Nasr, 2006, "Rapid prototyping: theory and practice". New York

[33] D.T. Pham, S.S. Dimov, 2001, "Rapid Manufacturing: the technologies and applications of rapid prototyping and rapid tooling". New York

[34] Hague, R.J.M., Reeves, P.E., 2000, "Rapid Prototyping, Tooling and Manufacturing". United Kingdom

[35] Nadooshan, A.A., Daneshmand, S. and Aghanajafi, C., 2007, "Application of RP Technology with Polycarbonate Material for Wind Tunnel Model Fabrication". World Academy of Science

[36] "Metal Laser-Sintering."

http://www.eos.info/en/home.html. Accessed 28April2010.

[37] Karim, M. M., Suzuki, K. and Kai, H., 2004, "Optimal Design of Hydrofoil and Marine Propeller Using Micro-Genetic Algorithm". Journal of Naval Architecture and Marine Engineering

[38] Lee, S., 2008, "Ice Controllable pitch propeller-Strength Check based on IACS Polar Class Rule", Texas, USA

[39] Vural, M., 2009, "Experiment results of RPM and Thrust for Various APC Propellers". Experiment Report, Illinois Institute of Technology

[40] "StrataSys FDM 2000 Specifications."

http://maelabs.ucsd.edu/alex/rapid_prototyping.html. Accessed 10AUG2009.

[41] "Devcon Epoxy."

http://www.devcon.com/. Accessed 30April2010.

Appendix A: The Part of OpenPVL_SW MATLAB Source Code for Generating Propeller Geometry in SolidWorks

The original OpenPVL code is available at

http://web.mit.edu/openprop/www/Download.html. The idea of this OpenPVL_SW code is use Solidworks to instead of Rihno with the same calculation methods. Based on this, replace all of the "rihno" words with "Solidworks" in the original code, which includes:

- Make_rihno_flag=1 changes to Make_Slidworks_flag=1 in Matlab code line 727
- Geometry (…Make_rihno_flag…) changes to Geometry (…Make_Solidworks_flag…) in Matlab code line 730
- Function []=Geometry (…Make_rihno_flag…) changes to Function []=Geometry (…Make_Solidworks_flag…) in Matlab code line 1217

The part of Matlab code to generate a propeller blade in Solidworks is integrated in the Matlab code line 1514 to instead of Rihno code. The logic flowchart of this solidworks code displays as below:

Create all of the function commands at the beginning of the solidworks macro

Use the calculation results in the original OpenPVL and sketch the first foil section, which is located at the root of the blade. (get points first and then b-spline connect all of the points)

Sketch all the other foil sections in the order of from root to tip of the propeller blade.

Connect all the points, which are located in the leading edge line to be as a reference line for lofting future

Selecting all the foil sections and the reference line to loft them to generate a propeller blade

Hide all the points and curves for nice look

The Solidworks code is shown as below: Comments are following after the symbol %.

```
% ------------------------------------------------------ Make Solidworks files
if Make_Solidworks_flag
   % ------ Make _Solidworks.txt, with coordinates for the entire propeller
   filename_Solidworks = strcat(filename,'_Solidworks.txt'); %create the soildworks
macro file
   fid = fopen(filename_Solidworks,'w'); % open the file for writing
   fprintf(fid,'Sub main()\n'); %Macro starting command
```

Start macro and declare objects for the macro

```
   fprintf(fid,'Dim swApp              As SldWorks.SldWorks\n'); %declare swap is a
solidworks object
   fprintf(fid,'Dim swModel            As SldWorks.ModelDoc2\n'); % declare
swModel is a document object
   fprintf(fid,'Dim swSketchPt()       As SldWorks.SketchPoint\n'); % declare sw
SketchPt() is a sketch point object
   fprintf(fid,'Dim swGuidePt(21)      As SldWorks.SketchPoint\n'); % declare
swGuidePt(21) is a sketch point object
   fprintf(fid,'Dim swSketch(21)       As SldWorks.Feature\n'); % declare
swSketch(21) is a feature object
   fprintf(fid,'Dim swFeatCurve(21)    As SldWorks.Feature\n'); % declare
swFeatCurve(21) is a feature object
   fprintf(fid,'Dim swGuideCurve       As SldWorks.Feature\n'); % declare
wGuideCurve is a feature object
   fprintf(fid,'Dim z                  As Long\n'); % declare z is a 32-bit number
   fprintf(fid,'Dim bRet               As Boolean\n'); % declare bRet is a ture or false
value

   fprintf(fid,'Set swApp = CreateObject("SldWorks.Application")\n'); %open Solidworks
session
   fprintf(fid,'Set swModel = swApp.ActiveDoc\n'); %grab the current document
   fprintf(fid,'swModel.SetAddToDB True\n'); % set to ture allows data to be directly
added to Solidworks model

   fprintf(fid,'swModel.ClearSelection\n'); % set to clear anything currently selected
```

Start 3D sketch to generate propeller foil sections

```
      fprintf(fid,'''start 3D sketch\n');
      fprintf(fid,'swModel.Insert3DSketch\n'); % insert a new 3D sketch for the points
      fprintf(fid,'ReDim swSketchPt(40) \n'); % define 40 points in the sketch
```

Generate points on propeller foil sections and connect points to generate foil sections

```
for j=1:2*Np      % for each point  for a foil section
        fprintf(fid,'%s %g %s (%f, %f, %f)\n','Set swSketchPt(', j, ') =
swModel.CreatePoint2', X3D(1,j,1), 0.9*Y3D(1,j,1), 0.9*Z3D(1,j,1)); % print poits data

    end
    fprintf(fid,'Set swGuidePt(0) = swSketchPt(1) \n'); % set the first reference line point

    fprintf(fid,'''exit 3D sketch \n');
    fprintf(fid,'swModel.Insert3DSketch \n'); % insert the sketch in Feature object

    fprintf(fid,'Set swSketch(0) = swModel.FeatureByPositionReverse(0) \n'); % define
the reference line point in the feature object

    fprintf(fid,'swModel.ClearSelection \n'); % set to clear the current selections

    fprintf(fid,'For z = 22 To 39 \n');  % define the points 22 to 39
    fprintf(fid,'bRet = swSketchPt(z).Select2(True, 0) \n');
    fprintf(fid,'Next z \n');

    fprintf(fid,'For z = 1 To 20 \n'); % define the points 1 to 20
    fprintf(fid,'bRet = swSketchPt(z).Select2(True, 0) \n');
    fprintf(fid,'Next z \n');

    fprintf(fid,'swModel.Insert3DSplineCurve (True) \n'); % use spline to connect the
defined points

    fprintf(fid,'Set swFeatCurve(0) = swModel.FeatureByPositionReverse(0) \n'); %
define the first refernce line point position in Feature object

    fprintf(fid,'swModel.ClearSelection \n'); %clear anything currently selected

  for i = 1:Mp        % for each section along the span
        fprintf(fid,'''start 3D sketch\n');
        fprintf(fid,'swModel.Insert3DSketch\n');
        fprintf(fid,'ReDim swSketchPt(40) \n');

    for j = 1:2*Np   % for each point for a foil section, the each line has the same
function as introduced as before
        fprintf(fid,'%s %g %s (%f, %f, %f)\n','Set swSketchPt(', j, ') =
swModel.CreatePoint2', X3D(i,j,1), Y3D(i,j,1), Z3D(i,j,1));
```

```
    end
    fprintf(fid,'%s %g %s \n', 'Set swGuidePt(', i, ') = swSketchPt(1)');

    fprintf(fid,'''exit 3D sketch \n');
    fprintf(fid,'swModel.Insert3DSketch \n');

    fprintf(fid,'%s %g %s \n', 'Set swSketch(', i, ') =
swModel.FeatureByPositionReverse(0)');

    fprintf(fid,'swModel.ClearSelection \n');

    fprintf(fid,'For z = 22 To 39 \n');
    fprintf(fid,'bRet = swSketchPt(z).Select2(True, 0) \n');
    fprintf(fid,'Next z \n');

    fprintf(fid,'For z = 1 To 20 \n');
    fprintf(fid,'bRet = swSketchPt(z).Select2(True, 0) \n');
    fprintf(fid,'Next z \n');

    fprintf(fid,'swModel.Insert3DSplineCurve (True) \n');

    fprintf(fid,'%s %g %s \n', 'Set swFeatCurve(', i, ') =
swModel.FeatureByPositionReverse(0)');

    fprintf(fid,'swModel.ClearSelection \n');
  end
```

Generate leading edge line as the reference line for loft function

```
    fprintf(fid,'''create guide curve \n');  % create reference line

    fprintf(fid,'For z = 0 To 20 \n'); % define the point 0 to 20
    fprintf(fid,'bRet = swGuidePt(z).Select2(True, 0) \n');
    fprintf(fid,'Next z \n');

    fprintf(fid,'swModel.Insert3DSplineCurve (False) \n'); % connect the points from 0
to 20 to gebarate a reference line

    fprintf(fid,'Set swGuideCurve = swModel.FeatureByPositionReverse(0) \n'); %
define the reference line position in Feature object
```

Generate propeller blade by loft function

```
    fprintf(fid,'''create loft \n'); %create loft to generate a propeller blade
    fprintf(fid,'swModel.ClearSelection \n'); %clear anything currently selected
```

```
fprintf(fid,'''select profile curves \n'); % select all the foil sections first
fprintf(fid,'''mark = 1 \n');

fprintf(fid,'For z = 0 To 20 \n');% the order of selecting foi sections is from 0 to 20
fprintf(fid,'bRet = swFeatCurve(z).Select2(True, 1) \n');
fprintf(fid,'Next z \n');

fprintf(fid,'''select guide curves \n'); % select the reference line second
fprintf(fid,'''mark = 2 \n');
fprintf(fid,'bRet = swGuideCurve.Select2(True, 2) \n');

fprintf(fid,'swModel.InsertLoftRefSurface2 False, False, False, 1#, 0, 0 \n'); %loft to
generate a propeller blade
```

Hide the sketches for clean

```
fprintf(fid,'''hide profile curves \n') %hide the sketches for clean
fprintf(fid,'swModel.ClearSelection \n');
fprintf(fid,'bRet = swFeatCurve(0).Select2(True, 0) \n');
fprintf(fid,'bRet = swFeatCurve(1).Select2(True, 0) \n');
fprintf(fid,'bRet = swFeatCurve(2).Select2(True, 0) \n');
fprintf(fid,'bRet = swFeatCurve(3).Select2(True, 0) \n');
fprintf(fid,'bRet = swFeatCurve(4).Select2(True, 0) \n');
fprintf(fid,'bRet = swFeatCurve(5).Select2(True, 0) \n');
fprintf(fid,'bRet = swFeatCurve(6).Select2(True, 0) \n');
fprintf(fid,'bRet = swFeatCurve(7).Select2(True, 0) \n');
fprintf(fid,'bRet = swFeatCurve(8).Select2(True, 0) \n');
fprintf(fid,'bRet = swFeatCurve(9).Select2(True, 0) \n');
fprintf(fid,'bRet = swFeatCurve(10).Select2(True, 0) \n');
fprintf(fid,'bRet = swFeatCurve(11).Select2(True, 0) \n');
fprintf(fid,'bRet = swFeatCurve(12).Select2(True, 0) \n');
fprintf(fid,'bRet = swFeatCurve(13).Select2(True, 0) \n');
fprintf(fid,'bRet = swFeatCurve(14).Select2(True, 0) \n');
fprintf(fid,'bRet = swFeatCurve(15).Select2(True, 0) \n');
fprintf(fid,'bRet = swFeatCurve(16).Select2(True, 0) \n');
fprintf(fid,'bRet = swFeatCurve(17).Select2(True, 0) \n');
fprintf(fid,'bRet = swFeatCurve(18).Select2(True, 0) \n');
fprintf(fid,'bRet = swFeatCurve(19).Select2(True, 0) \n');
fprintf(fid,'bRet = swFeatCurve(20).Select2(True, 0) \n');
fprintf(fid,'bRet = swGuideCurve.Select2(True, 0) \n')
fprintf(fid,'swModel.BlankRefGeom \n')

fprintf(fid,'''hide the 3D sketches since they have a lot of points \n');
fprintf(fid,'''note that ModelDoc2::BlankSketch will only hide one sketch \n');
```

```
    fprintf(fid,'For z = 0 To 20 \n');
    fprintf(fid,'swModel.ClearSelection \n');
    fprintf(fid,'bRet = swSketch(z).Select2(True, 0) \n')
    fprintf(fid,'swModel.BlankSketch \n');
    fprintf(fid,'Next z \n');
    fprintf(fid,'swModel.SetAddToDB False \n');
    fprintf(fid,'End Sub \n');
    fclose(fid);

end                                    % (END IF Make_Solidworks_flag)
```

Appendix B: SolidWorks Macro for the AUV Propeller Geometry

```
Sub main()
Dim swApp              As SldWorks.SldWorks
Dim swModel            As SldWorks.ModelDoc2
Dim swSketchPt()       As SldWorks.SketchPoint
Dim swGuidePt(21)      As SldWorks.SketchPoint
Dim swSketch(21)       As SldWorks.Feature
Dim swFeatCurve(21)    As SldWorks.Feature
Dim swGuideCurve       As SldWorks.Feature
Dim z              As Long
Dim bRet           As Boolean
Set swApp = CreateObject("SldWorks.Application")
Set swModel = swApp.ActiveDoc
swModel.SetAddToDB True
swModel.ClearSelection
'start 3D sketch
swModel.Insert3Dsketch
ReDim swSketchPt(40)
Set swSketchPt( 1 ) = swModel.CreatePoint2 (0.041867, -0.022025, 0.050618)
Set swSketchPt( 2 ) = swModel.CreatePoint2 (0.040091, -0.016425, 0.052702)
Set swSketchPt( 3 ) = swModel.CreatePoint2 (0.036700, -0.012675, 0.053727)
Set swSketchPt( 4 ) = swModel.CreatePoint2 (0.033013, -0.009267, 0.054419)
Set swSketchPt( 5 ) = swModel.CreatePoint2 (0.029138, -0.006084, 0.054866)
Set swSketchPt( 6 ) = swModel.CreatePoint2 (0.025122, -0.003082, 0.055116)
Set swSketchPt( 7 ) = swModel.CreatePoint2 (0.020983, -0.000247, 0.055202)
Set swSketchPt( 8 ) = swModel.CreatePoint2 (0.016733, 0.002427, 0.055149)
Set swSketchPt( 9 ) = swModel.CreatePoint2 (0.012358, 0.004913, 0.054983)
Set swSketchPt( 10 ) = swModel.CreatePoint2 (0.007883, 0.007241, 0.054725)
Set swSketchPt( 11 ) = swModel.CreatePoint2 (0.003254, 0.009329, 0.054408)
Set swSketchPt( 12 ) = swModel.CreatePoint2 (-0.001498, 0.011218, 0.054050)
Set swSketchPt( 13 ) = swModel.CreatePoint2 (-0.006356, 0.012935, 0.053665)
Set swSketchPt( 14 ) = swModel.CreatePoint2 (-0.011298, 0.014511, 0.053261)
Set swSketchPt( 15 ) = swModel.CreatePoint2 (-0.016313, 0.015962, 0.052844)
Set swSketchPt( 16 ) = swModel.CreatePoint2 (-0.021401, 0.017307, 0.052419)
Set swSketchPt( 17 ) = swModel.CreatePoint2 (-0.026529, 0.018548, 0.051993)
Set swSketchPt( 18 ) = swModel.CreatePoint2 (-0.031643, 0.019731, 0.051556)
Set swSketchPt( 19 ) = swModel.CreatePoint2 (-0.036749, 0.020891, 0.051097)
Set swSketchPt( 20 ) = swModel.CreatePoint2 (-0.041847, 0.022055, 0.050605)
Set swSketchPt( 21 ) = swModel.CreatePoint2 (-0.041888, 0.021994, 0.050631)
Set swSketchPt( 22 ) = swModel.CreatePoint2 (-0.037971, 0.019019, 0.051823)
Set swSketchPt( 23 ) = swModel.CreatePoint2 (-0.034044, 0.015989, 0.052836)
Set swSketchPt( 24 ) = swModel.CreatePoint2 (-0.030132, 0.012909, 0.053672)
Set swSketchPt( 25 ) = swModel.CreatePoint2 (-0.026269, 0.009786, 0.054328)
Set swSketchPt( 26 ) = swModel.CreatePoint2 (-0.022426, 0.006638, 0.054802)
Set swSketchPt( 27 ) = swModel.CreatePoint2 (-0.018545, 0.003514, 0.055090)
Set swSketchPt( 28 ) = swModel.CreatePoint2 (-0.014618, 0.000456, 0.055200)
```

Declare all the objects as described as Appendix A

Start to create the points that on foil section number 1

```
Set swSketchPt( 29 ) = swModel.CreatePoint2 (-0.010628, -0.002503, 0.055145)
Set swSketchPt( 30 ) = swModel.CreatePoint2 (-0.006555, -0.005324, 0.054945)
Set swSketchPt( 31 ) = swModel.CreatePoint2 (-0.002373, -0.007967, 0.054624)
Set swSketchPt( 32 ) = swModel.CreatePoint2 (0.001943, -0.010395, 0.054215)
Set swSketchPt( 33 ) = swModel.CreatePoint2 (0.006341, -0.012684, 0.053725)
Set swSketchPt( 34 ) = swModel.CreatePoint2 (0.010845, -0.014799, 0.053182)
Set swSketchPt( 35 ) = swModel.CreatePoint2 (0.015441, -0.016765, 0.052595)
Set swSketchPt( 36 ) = swModel.CreatePoint2 (0.020137, -0.018573, 0.051984)
Set swSketchPt( 37 ) = swModel.CreatePoint2 (0.024951, -0.020205, 0.051371)
Set swSketchPt( 38 ) = swModel.CreatePoint2 (0.029920, -0.021619, 0.050793)
Set swSketchPt( 39 ) = swModel.CreatePoint2 (0.035141, -0.022703, 0.050318)
Set swSketchPt( 40 ) = swModel.CreatePoint2 (0.041867, -0.022025, 0.050618)
Set swGuidePt(0) = swSketchPt(1)
'exit 3D sketch
swModel.Insert3DSketch
Set swSketch(0) = swModel.FeatureByPositionReverse(0)
swModel.ClearSelection
For z = 22 To 39
bRet = swSketchPt(z).Select2(True, 0)
Next z
For z = 1 To 20
bRet = swSketchPt(z).Select2(True, 0)
Next z
swModel.Insert3DSplineCurve (True)
Set swFeatCurve(0) = swModel.FeatureByPositionReverse(0)
swModel.ClearSelection
```

Connect all points to generate the foil section and define the point position for the reference line

.

.

.

This step is repeated until get 21 foil sections

.

.

.

```
'create guide curve
For z = 0 To 20
bRet = swGuidePt(z).Select2(True, 0)
Next z
swModel.Insert3DSplineCurve (False)
Set swGuideCurve = swModel.FeatureByPositionReverse(0)
```

Create reference line for lofting. The detailed each step is described as Appendix A

```
'create loft
swModel.ClearSelection
'select profile curves
'mark = 1
For z = 0 To 20
bRet = swFeatCurve(z).Select2(True, 1)
Next z
'select guide curves
'mark = 2
bRet = swGuideCurve.Select2(True, 2)
swModel.InsertLoftRefSurface2 False, False, False, 1#, 0, 0
'hide profile curves
swModel.ClearSelection
bRet = swFeatCurve(0).Select2(True, 0)
bRet = swFeatCurve(1).Select2(True, 0)
bRet = swFeatCurve(2).Select2(True, 0)
bRet = swFeatCurve(3).Select2(True, 0)
bRet = swFeatCurve(4).Select2(True, 0)
bRet = swFeatCurve(5).Select2(True, 0)
bRet = swFeatCurve(6).Select2(True, 0)
bRet = swFeatCurve(7).Select2(True, 0)
bRet = swFeatCurve(8).Select2(True, 0)
bRet = swFeatCurve(9).Select2(True, 0)
bRet = swFeatCurve(10).Select2(True, 0)
bRet = swFeatCurve(11).Select2(True, 0)
bRet = swFeatCurve(12).Select2(True, 0)
bRet = swFeatCurve(13).Select2(True, 0)
bRet = swFeatCurve(14).Select2(True, 0)
bRet = swFeatCurve(15).Select2(True, 0)
bRet = swFeatCurve(16).Select2(True, 0)
bRet = swFeatCurve(17).Select2(True, 0)
bRet = swFeatCurve(18).Select2(True, 0)
bRet = swFeatCurve(19).Select2(True, 0)
bRet = swFeatCurve(20).Select2(True, 0)
bRet = swGuideCurve.Select2(True, 0)
swModel.BlankRefGeom
'hide the 3D sketches since they have a lot of points
'note that ModelDoc2::BlankSketch will only hide one sketch
For z = 0 To 20
swModel.ClearSelection
bRet = swSketch(z).Select2(True, 0)
swModel.BlankSketch
Next z
swModel.SetAddToDB False
End Sub
```

Hide all of the sketches for clean