A 3D MULTIDISCIPLINARY AUTOMATED DESIGN OPTIMIZATION TOOLBOX FOR WIND TURBINE BLADES BASED ON NS SOLVER AND EXPERIMENTAL DATA

Sagi Sagimbayev, B.Eng Mechanical Engineering

Submitted in fulfilment of the requirements for the degree of Masters of Science
in Mechanical Engineering

School of Engineering
Department of Mechanical Engineering
Nazarbayev University

53 Kabanbay Batyr Avenue,
Astana, Kazakhstan, 010000

Supervisors: Michael Yong Zhao
Soo Lee Teh, Sai Fok

December 2017

Abstract

This thesis attempts to develop a framework to optimize wind turbine blades automatically by a multidisciplinary 3D modeling and simulation methods. The original NREL Phase VI wind turbine blade and its experimental measurements are used to validate the Computational Fluid Dynamics (CFD) model developed in ANSYS Fluent and based on the 3D Navier-Stokes (NS) solver with a realizable k-epsilon turbulence model, which is later used in the automation process. The automated design optimization process involves multiple modeling and simulation methods using Solidworks and ANSYS Mesher and ANSYS Fluent NS solver, which are integrated and controlled through Matlab by implementing the scripting capabilities of each software package. Then all scripts are integrated into one optimization cycle, with its optimization objective being the highest mean value of 3D Lift/Drag ratio (3DLDR) across the blade. A 3DLDR distribution across the blade can be calculated by the Inverse Blade Element Momentum (IBEM) Method based on experimental measurements. The optimization process is performed to find optimized twist angles across the blade using the Angle of Attack (AOA) with the highest 3DLDR as a reference, in order to

achieve the optimization objective. Therefore, the automatic optimization framework is based on 3D solid modeling and 3D aerodynamic simulation and guided by IBEM and experimental data. Thus the design tool is capable of exploiting the 3D stall delay of blades designed by the traditional 2D BEM method to enhance their performances. It is found that this automated framework can result in optimized blade geometries with improvement of performance parameters compared to the original ones.

Acknowledgements

Here I would like to thank my supervisors - professors Micheal Yong Zhao, Soo Lee The and Sai Fok for their advices and guidance. Without them this thesis will not be possible. They were always willing to help and provided support throughout the writing up of the thesis.

Also I would like to thank professor Minh Nguyen, who generously offered his work station to run ANSYS simulation.

Lastly, I have to express my gratitude to the Lee Jay Fingersh. He provided data from the NREL Phase VI experiment.

List of Abbreviations

CFD – Computational Fluid Dynamics

NS – Navier-Stokes

IBEM – Inverse Blade Element Momentum

AOA – Angle of Attack

BEM – Blade Element Momentum

CAD – Computer Aided Design

NREL – National Renewable Energy Laboratory

FEM – Finite Element Method

MRF - Moving Reference Frame

COM - Component Object Model

API - Application Programming Interface

CSV - Comma Separated Value

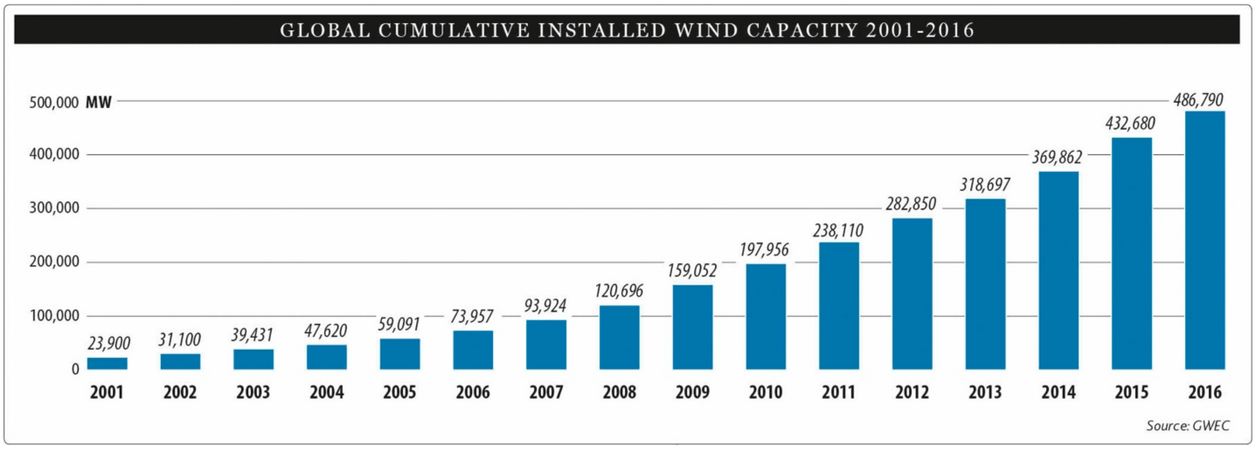
Table of Contents

Chapter 1 - Introduction

1.1 Background

The limited resources of the fossil fuels drive scientist and engineers around the globe to explore the new boundaries of renewable energy sources to avoid possible energy crisis and eliminate harmful impact of using nonrenewable energy products. Hence, switching to green energy will bring number of benefits to the environment as well as to the society. The most widespread energy resource on the Earth is wind and development of methods of extracting energy from it is the subject of crucial importance. The global demand for renewable energy is ever increasing, and this forces the development of new more efficient and powerful wind turbines. According to Figure 1.1 total installed capacity for wind energy increased from 2001 by 22 times and reached almost 500000 MW [1].

Figure 1.1: Total cumulative installed wind capacity



Throughout the existence of wind turbines, engineers have developed numerous techniques to design wind turbine blades. They range from purely empirical to fully computational ones. Each of them was developed based upon simplifications and assumptions that can only be applied in certain limited conditions. This leaves engineers with only handful of choices to pick from when designing blades for wind turbines. The most commonly used one, that is also used in many blade designing softwares, such as Q-blade, WT_perf etc.,[2] is the Blade Element Momentum (BEM) method. This method offers fast and easy way to design a blade, but lacks accuracy of predicting complex aerodynamic behaviors.

1.2 Problem statement

The investigation of the wind turbine performance using CFD is often complicated by the time consuming processes such as geometry modeling and meshing, which significantly limits ability to quickly test new designs and configurations. Some software packages have limited capabilities and forces users to rely on multiple programs at a time to evaluate desired parameters or performance. This work tries to create an environment that integrates capabilities of CAD and CFD packages together with aerodynamic performance prediction model in order to optimized an existing design and to significantly reduce time user spends on time consuming processes.

1.3 Motivation

The main motivation of this thesis is to make a contribution to the fast growing area of the renewable energy technologies and to automate the highly complex wind turbine blade design process using complex 3D aerodynamic and solid modeling and simulation methods.

Traditionally, wind turbine rotors and blades are designed by experience and highly simplified aerodynamic models, such as BEM, which assumes 2D inviscid flow and uses 2D airfoil data [3]. As a result, blades designed by the traditional BEM method, have stall delays because the 3D flow effects are not fully considered in their design methods. This also means that there is potential to exploit the 3D stall delays to improve the performances of the blades designed by the traditional methods by using more optimized geometries, because the 3D rotating flow around these wind turbine blades tend to delay the onset of flow separation which is beneficial to blade performance enhancement by increasing their AoA further. Therefore, it is proposed that the IBEM method [4] with experimental data are employed to guide the optimization of an existing blade (NREL IV blade). The Optimized design will be found through an automatic process and validated by a 3D NS solver with proper turbulence model to account for the highly complex flow physics.

Modern CFD packages have powerful capabilities but setting up a problem can take a lot of time and effort. This issue is especially prominent when a user tries to carry out repetitive operations of solid modeling, meshing and flow simulation. Additionally, if the problem includes the change of shape, it requires re-meshing of the new geometry in one run of the simulation. In the case of ANSYS, to perform shape alteration and meshing in the period of one simulation several modules, such as ANSYS Mechanical and ANSYS Fluent, are utilized simultaneously. The first module is responsible for the shape modification and creating mesh. The latter one executes CFD calculation. This approach is highly computationally inefficient and has limited control over 3D model shape adjustment. Furthermore, built in CAD packages offer very limited functionality compared to their standalone counterparts. This creates problems with transferring geometries from one system to another. Thus bridging the gap between a sophisticated CAD package and a sophisticated CFD solver is the subject of great interest. Moreover, in the case of ANSYS postprocessor, certain data quantities cannot be directly manipulated in a desired way, therefore limiting capabilities for data interpretation. This leaves the user with multiple separate programs at his hand, each having its own purpose. This creates an opportunity to create an integrated environment that combines advantages of all package that can be used in simulation process.

1.4 Objectives

1. **To build a parametric model of the NREL Phase VI wind turbine blade.** This will help to easily change the shape by varying a set of predefined parameters, allowing quick and easy creation of various configuration without building a model from the ground up every iteration. Parametric design facilitates the optimization process.
2. **To set up a CFD model for the blade with proper boundary conditions and computational mesh.** This will include choosing the right coordinate system for the rotating blade and the right boundary conditions, as well as generating

suitable meshes and conduct a mesh convergence study to choose the optimal mesh.

3. **To validate the blade CFD model with experimental results.** This process is very crucial because it will guarantee that produced results are correct and trustworthy and the model can be used for further calculations. The validated CFD model will be utilized in the optimization cycle.
4. **To create the automation process for optimization by coupling multiple sophisticated 3D design and simulation programs.** The developed framework will allow communication between Solidworks, ANSYS and Matlab and ensure data transfer among them.
5. **To use the Inverse Blade Element Momentum (IBEM) method to generate truly 3D lift and drag coefficients across the blade based on experimental data, which will then be used to guide the automatic design optimization to achieve its objective by exploiting the 3D stall delay of the NREL VI blade designed by the 2D BEM method.** This method will help to produce new blade geometries that have higher overall lift to drag ratio under 3D turbulent flow conditions.

1.5 Limitations

The work presented here was conducted based on several assumptions and simplifications in the CFD simulation part. This includes performing a steady-state calculation on the simplified fluid domain without accounting for fluid structure interaction. There are very limited instructions available about automation of ANSYS and Solidworks and all crucial bits of information were retrieved from internet forums and other similar works.

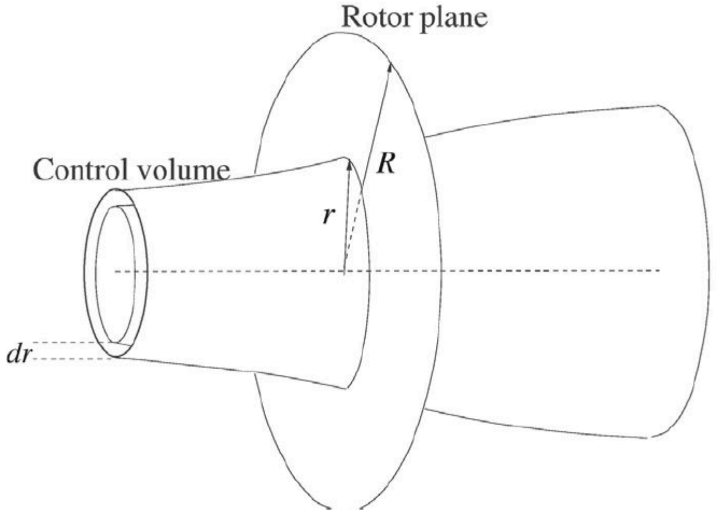
Chapter 2 – Literature Review

2.1 Wind Turbine Design Methods

2.1.1 Classical Blade Element Momentum theory

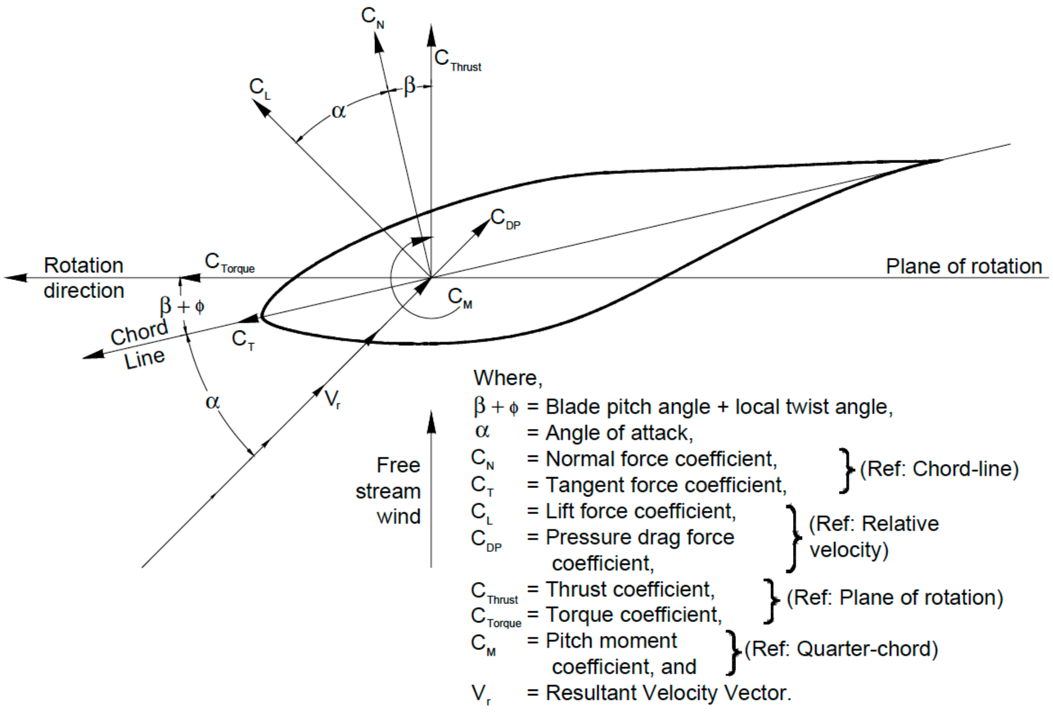
The main principles behind this theory was developed in 1935 by Glauert [5]. The main assumption incorporated in this technique is that fluid flow occurs in the individual sectional tubes which split blade surface into number of elements from which loadings are calculated. Figure 2.1 shows how this looks. As a result of this several constrains and assumptions are made: radial sections are not affecting each other and forces are constant for every element [6]. This approach proved itself to be simple, fast and to certain extent accurate. However, as it was mentioned earlier, it heavily relies on 2D sectional airfoil data and only existing experimental data can be used, which is a problem for completely new airfoil designs.

Figure 2.1: Control volume of the section used in BEM



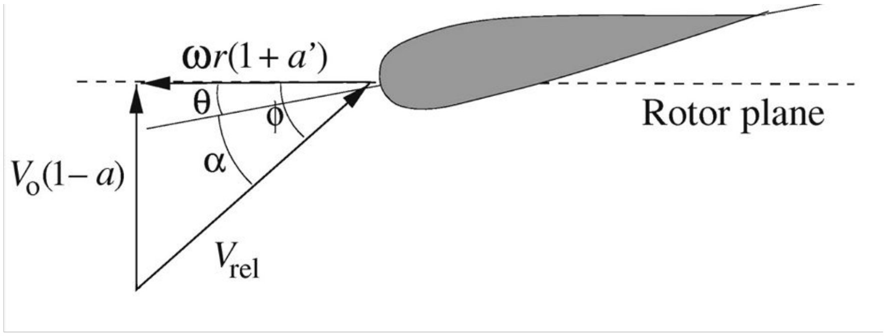
From its name, this theory combines two widely used models for calculating aerodynamics of the wind turbine rotors: blade element and momentum theory. This thesis will not elaborate on derivation of the equations included in these two theories, but rather show the governing relations and the designing process sequence. Figure 2.2 show widely accepted conventions for aerodynamic forces calculations [7]. This convention will be used throughout this work.

Figure 2.2: Schematic representation of the wind turbine rotor section and aerodynamic forces convention [5]



Standard BEM model can be represented by the set of equations that characterized by local parameters, such as distance from the center and local force coefficients, as well as global ones: upstream wind velocity, rotational speed of the blade, and number of blades. Figure 2.3 shows the relations between velocities acting on the blade cross section at certain span location [4]. Here, a and a’ are the axial and angular induction factors respectively

Figure 2.3: Direction of velocities at an annular element



From this, angle between relative velocity and plane of rotation can be determined as

$$\phi = atan\left[\frac{U_\infty(1-a)}{\Omega r(1+a')}\right] \quad (2.1)$$

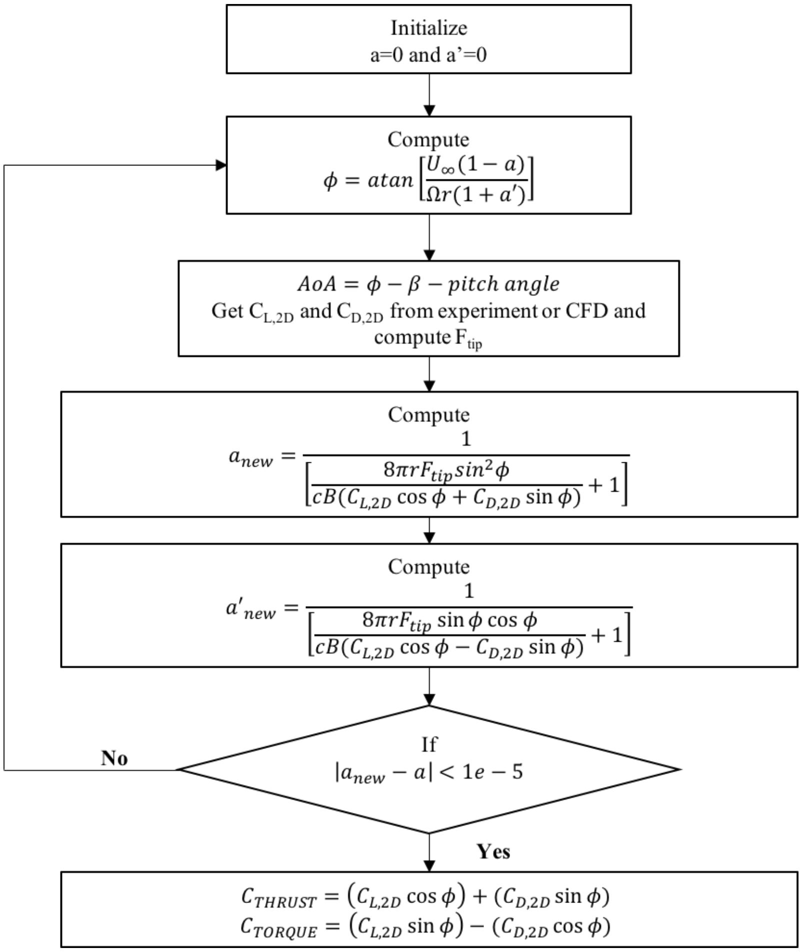
Then, angular and axial induction factors can be expressed in terms of lift and drag coefficients

$$a = \frac{1}{\left[\frac{8\pi r F_{tip} sin^2\phi}{cB(C_{L,2D}\cos\phi + C_{D,2D}\sin\phi)} + 1\right]} \quad (2.2)$$

$$a' = \frac{1}{\left[\frac{8\pi r F_{tip}\sin\phi\cos\phi}{cB\left(C_{L,2D}\cos\phi - C_{D,2D}\sin\phi\right)} + 1\right]} \quad (2.3)$$

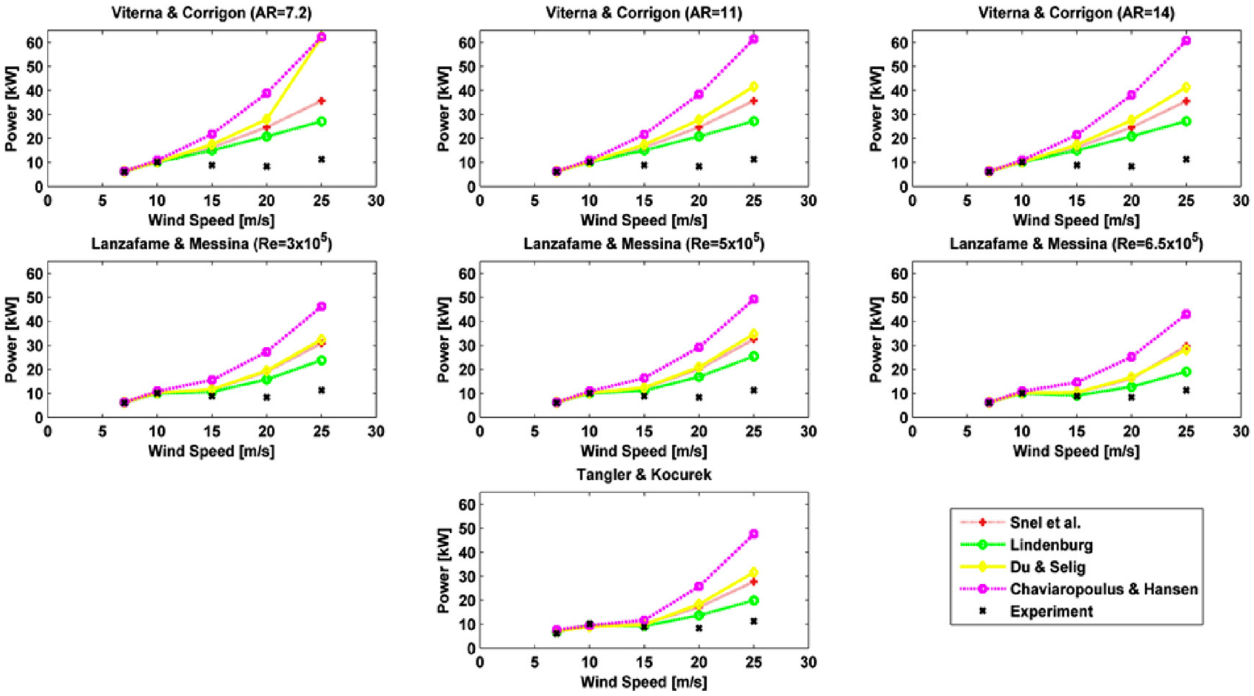
Lift and drag coefficients presented in the equations (2.2) and (2.3) are taken from 2D airfoil data [3]. These three equations are used in the BEM scheme to calculate forces acting on the annular element. Since these system of equations are non-linear it requires special iterative scheme to compute them. First of all, the blade must be divided into the number of sections on which calculations will be performed. Then the procedure depicted in the Figure 2.4 is carried out until the convergence is reached.

Figure 2.4: Classical BEM method scheme



The major drawback of this approach is the independence of the annular elements located along the blade. In other words, they do not account for any 3D flow effects occurring in the flow in the spanwise direction. Meaning that BEM method only assumes steady-state aerodynamics, where flow is considered inviscid. In the real life wind turbines operate under unsteady aerodynamic conditions which cause increased fluctuating loads. This makes model created by BEM unreliable [3]. One of this effects is called stall-delay and it is related to the unsteady flows in the spanwise direction caused by rotation of the wind turbine blade. This phenomenon is associated with increased lift in the regions where stall is expected, thus delaying the stall [2]. Another nuisance is the limited availability of the lift and drag coefficients for various angle of attacks and Reynolds numbers. This shortens the number of possible configurations that an engineer can work with. Many studies have attempted to come with correction factors and extrapolation techniques that can enhance BEM method to include 3D flow phenomena. Extrapolation method introduced by [8-10] used lift and drag coefficients at special angles of attacks obtained from experimental data to calculate coefficients for wider range of values of angle of attack. Other studies, such as [11-14], focused more on the including stall delay effect on the rotor surface and derived correction models to convert 2D lift and drag coefficients into 3D. In general, these models over-predicted power output of the wind turbine if compared to the experimental measurements, which is shown in the Figure 2.5 [2].

Figure 2.5: Power output obtained from different 3D correction models [2]

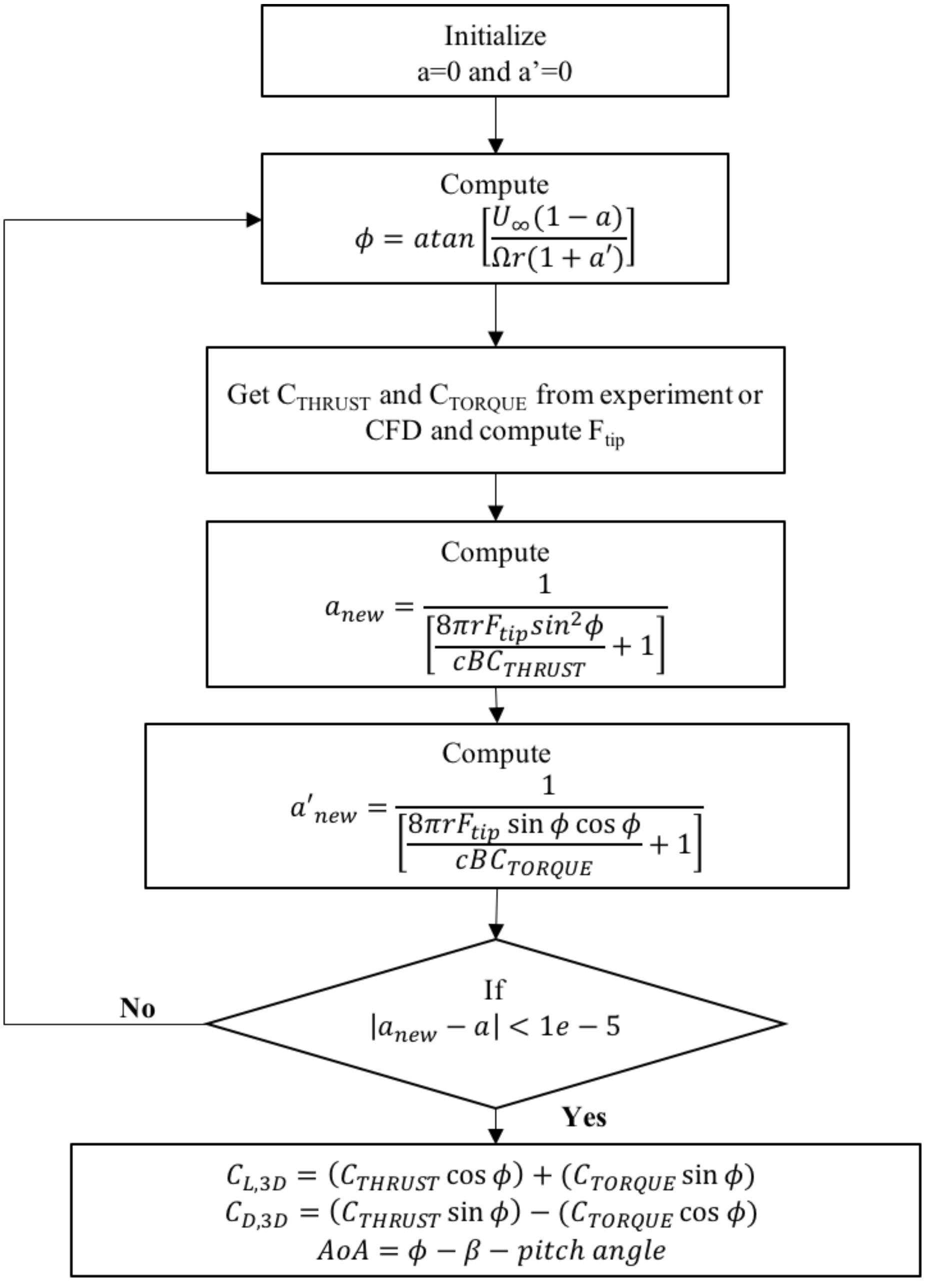


Despite previously mentioned drawbacks, BEM method is still widely used to design wind turbine blades, because it produces relatively accurate results at low cost. But the blade designed using this method will not account for load fluctuations and under/overestimated power outputs, which may lead to the damage to the wind turbine.

2.1.2 Inverse Blade Element Momentum (IBEM) Method

The approach used in implementing of this method is very similar to the standard BEM method. The main difference is that it uses force data obtained directly from experiment or simulation to evaluate lift, drag coefficients as well as angle of attack. This helps to include unsteady 3D flow effects, such as stall delay, in the calculation and as a result produce more accurate data. The sequence of actions used to calculate desired 3D lift and drag is the same as in the classical BEM. This scheme is presented in the Figure 2.6.

Figure 2.6: IBEM method scheme



In general sense, this method is the reversed engineered version of the standard BEM and is used to evaluate relative incoming velocity angles. Evaluating these angles is a challenging task both in the experiments and in the computer simulation. [4] performed an investigation of separate ways to evaluate flow angles and come to agreement that inverse BEM does a good job in estimating those angles.

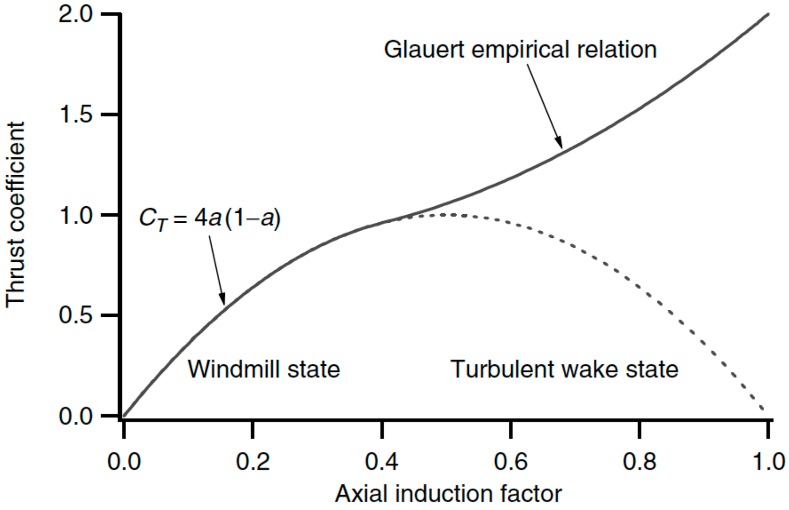
2.1.3 Prandtl /Glauert tip correction factor

Both inverse and traditional BEM method include blade tip correction factor inside axial and angular induction factor terms (Figures 2.4 and 2.6). It describes how flow occurring in the wake of a wind turbine is affecting aerodynamic parameters of the blade. The formulation of this correction is the following:

$$F_{tip} = \frac{2}{\pi}\cos^{-1}\left[\exp\left(-\frac{B(1-r/R)}{2r/R\sin\phi}\right)\right] \quad (2.4)$$

Equation 2.4 was first introduced by [5] and it is implanted in wide variety of BEM codes [15]. This work will also utilize this model in the implementation of the BEM code. Glauret model more accurately predicts the real behavior of the flow when axial induction factor exceeds 0.4, thus this correction will produce more accurate results [3]. Figure 2.7 shows the relation between thrust coefficient and axial induction factor.

Figure 2.7: Comparison of classical and Glauret models



2.2 Modern Wind Turbine Optimization Methods

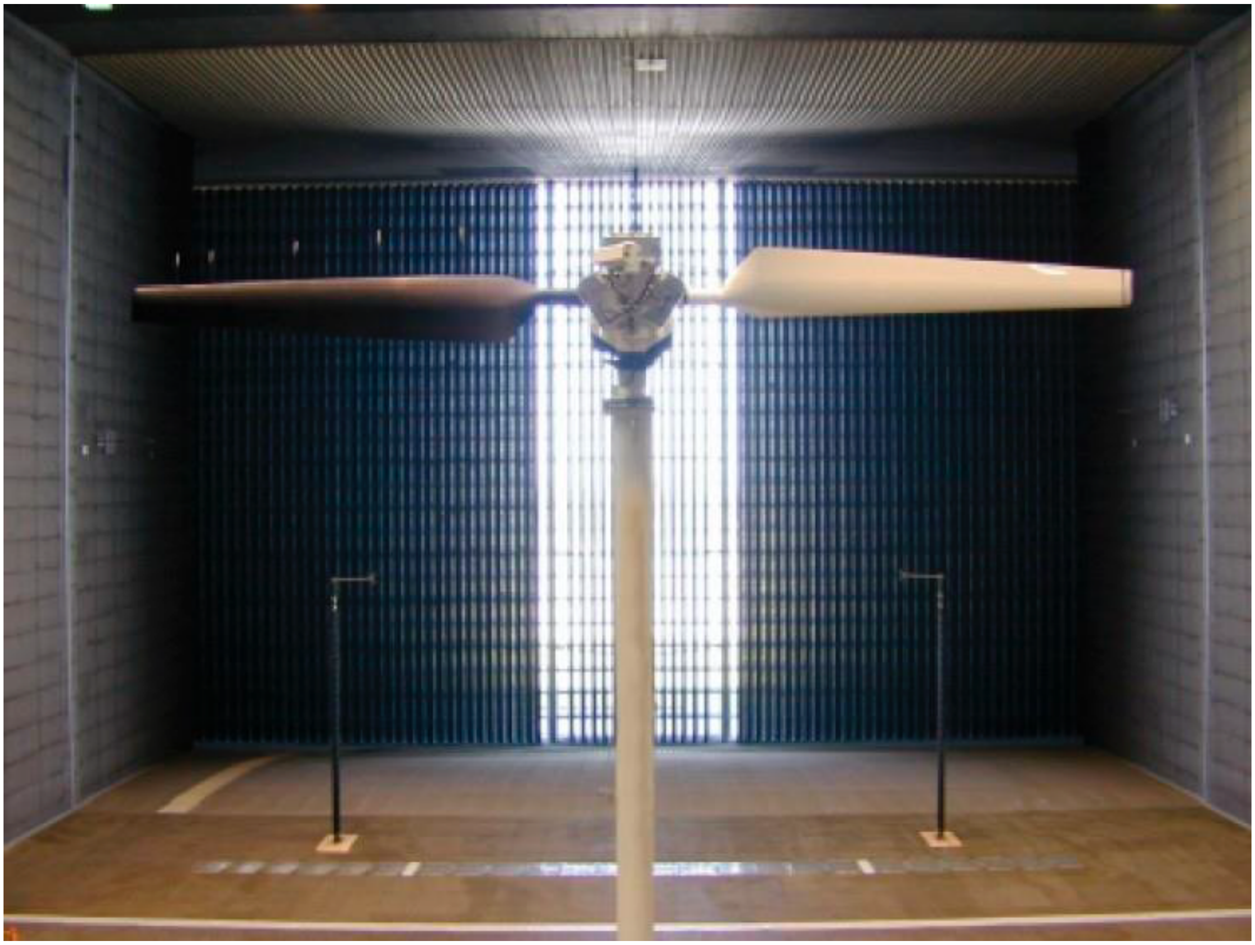
Design methods described in the previous sections are the tools for creating wind turbine blades from the ground up and in some times it can be redundant. Improving already existing designs is a better option because it can save a lot of time and effort and the improvements can be directly compared with the original design. There are wide variety of optimization studies performed on wind turbine blades. All of them are using different optimization methods and pursue different optimization goals. One study [16] used Artificial Neural Network to optimize NREL Phase VI wind turbine blade to produce higher power output. Neural networks use results obtained from CFD code in order to replicate its behavior to avoid time consuming simulations. Optimized blade showed 9% growth in power production. A multidisciplinary aero-elastic optimization of the wind turbine performed by [17] used special optimization toolbox in couple with BEM based CFD code to minimize mass and improve structural properties of the blade. Another study [18] optimized blade under fluid structure interaction conditions. Here, researchers used combination of software packages to perform structural and aerodynamics analysis and optimized blade wall thickness by using Sequential two-point diagonal quadratic approximate optimization. [19] demonstrated the optimization of the blade using twist angle, chord length and airfoil shape as variables. They used BEM based code together with Ant Colony algorithm to find optimal shape. Another use of the nature based optimization algorithm was presented by [20]. They coupled this algorithm together with BEM method, which evaluates aerodynamic performance, and the Finite Element Method (FEM) which calculates structural properties of the blade. The goal of optimization was to minimize tip deflection and mass of the blade. Matlab and ANSYS combination to solve optimization problem was utilized by [21]. As in [20] they used BEM and FEM to estimate the aerodynamic and structural performance. They used genetic algorithm for optimization to find blade design that satisfy the criteria of maximum energy production and minimum mass. It can be noted that all these studies are incorporating BEM method as a tool of evaluation of the aerodynamic performance. As it was mentioned in the previous section, the BEM cannot fully account for unsteady 3D aero dynamical effects.

2.3 NREL UAE Phase VI experiment

For the purpose of the validation of the various CFD codes, and numerical models concerning various unsteady aerodynamic effects the National Renewable Energy Laboratory (NREL) conducted wide range of experiments on the small wind turbine blade and recorded vast amount of data from the attached sensors. This includes testing wind turbine in upwind and downwind conditions, imposing different inlet wind speeds, pitch, yaw and teetering angles. Thus providing researches with ample variations of scenarios that they can validate and replicate.

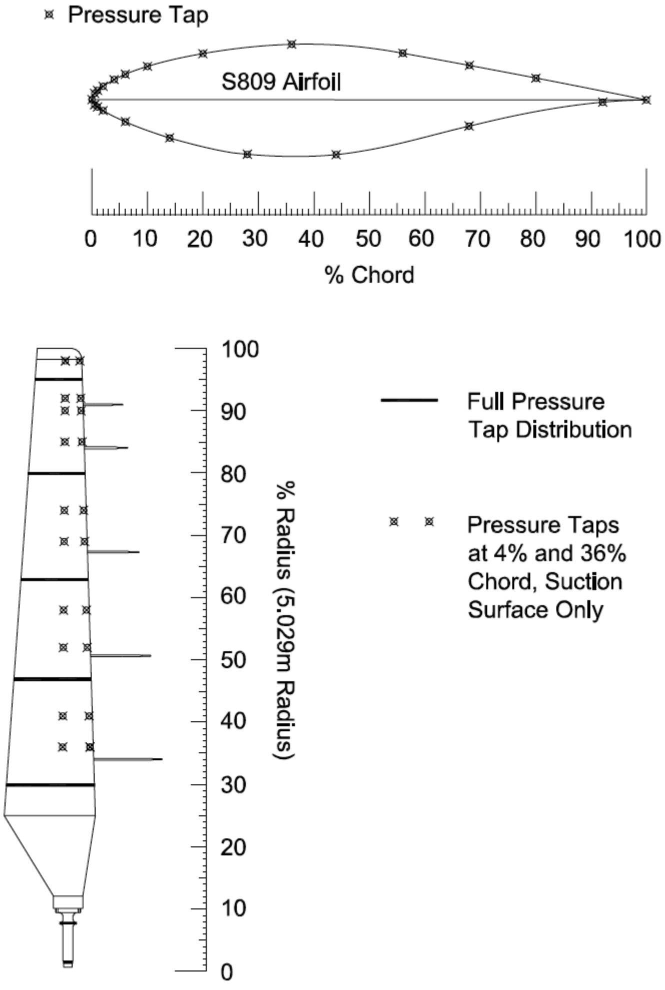
The experiment was conducted by NREL in the NASA Ames wind tunnel. The frontal dimensions of the tunnel are the following: 24.4 x 36.6m. The picture of the turbine inside the tunnel is shown in Figure 2.8 [7]. Wind turbine used in the experiment is the 10m diameter, 20kW rated two blade turbine.

Figure 2.8: NREL Phase VI wind turbine



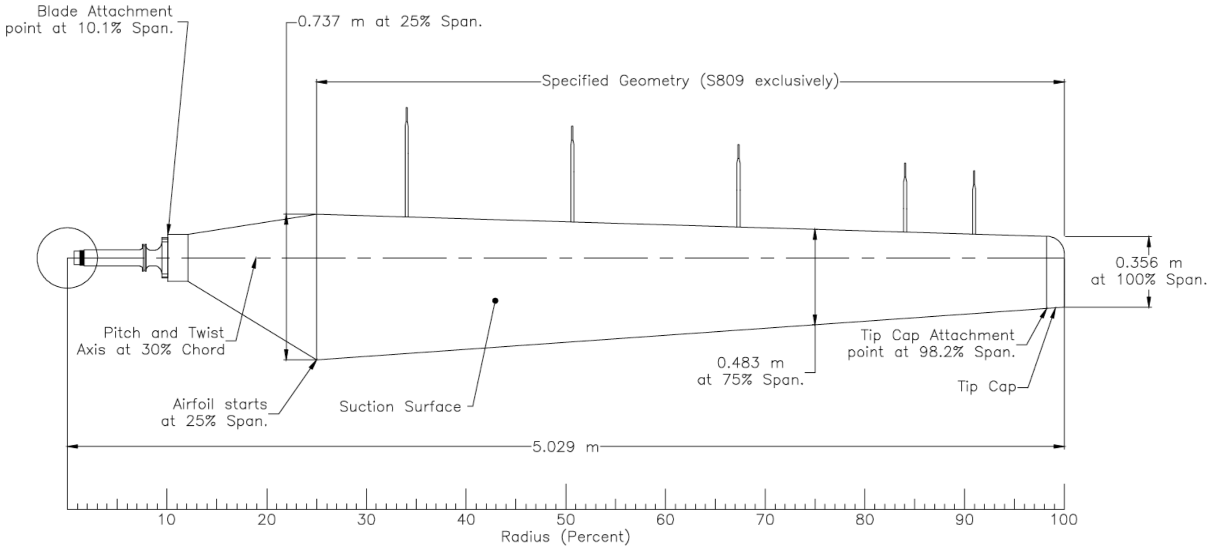
The blades used on this turbine are stall-regulated with variable twist angle in the spanwise direction and is based on NREL s809 airfoil profile. Blade surface is equipped with the pressure taps. They are located at the certain locations along the blade. There are five sections with full range of pressure taps across the chord length and ten locations with only two taps. There are also five rods attached to the leading edge of the blade. At the end of every rode there is 5-hole probe to measure flow conditions. The distribution is clearly shown on the Figure 2.9 [7].

Figure 2.9: Pressure tap locations



Pressure taps are used to measure dynamic pressure at the top and bottom surfaces. These measurements will be later used to validate the accuracy of the CFD simulation and estimate thrust and torque coefficients.

Figure 2.10: NREL Phase VI blade geometry



The blade consists of three major regions. The main part is build from s809 airfoil profile which is twisted along the radial direction. Then, there is a transition region which extends from cylindrical section to the airfoil profile. All these regions and exact location of the transition is depicted in the Figure 2.10 [7]. More detailed description is provided in the appendix.

NREL Phase VI data campaigns include measurements from 30 distinct sequences of tests that are investigating various flow conditions and wind turbine configurations. Each sequence is designated by the capital Latin letter. For the validation, presented in this work, sequence "S" was used. It is characterized by the 0 yaw angle, no tittering and absence of the 5-hole probes, to exclude their influence on the fluid flow. The rotational speed is 72RPM and incoming wind speed is ranged between 5 to 25 m/s. Direction of the wind is perpendicular to the rotation plane. Another defining characteristic of this sequence is the pitch angle, i.e. angle between chord located at the tip of the blade and plane of rotation [7].

Chapter 3 - Mathematical Formulations and Computational Simulation

3.1 Overview

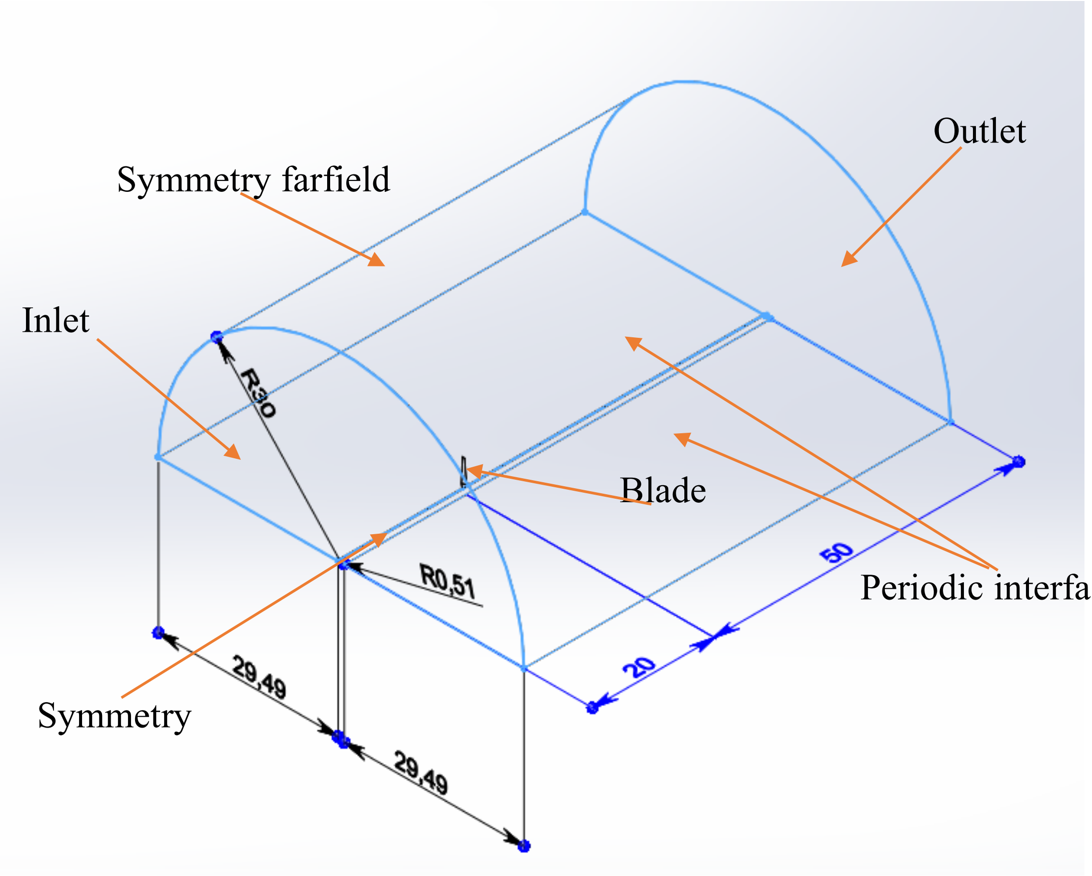
ANSYS Fluent was chosen as a tool for CFD validation and as a solver for the automation routine as it provides flexible and precise modeling capabilities when dealing with unsteady flow conditions as it offers wide range of turbulence models. Number of different studies attempted to corroborate NREL Phase VI experiment using different software packages such OpenFOAM [22], NUMECA [16] and ANSYS Fluent [23]. These studies used Reynolds-averaged Navier–Stokes (RANS) equations in combination with various turbulence models, which include k-omega, SST k-omega and k-epsilon. Additionally, these studies implemented both structured and unstructured meshes for simulation. They all come to close agreement with experimental measurements.

3.2 Computational Domain

For computational efficiency it was decided to exploit symmetrical and rotational nature of the wind turbine operation. ANSYS Fluent has a capability to solve this type of problems through the implementation of Moving Reference

Frame (MRF). This feature helps to simulate the interaction between rotating blade and incoming air flow. This allows user to use only one blade and the domain will be in the form of a half-cylinder with blade located inside of it. The radius of this half-cylinder was chosen to be 30m and the total length was set to 70m. The blade is located 20m from the inlet and 50m before outlet. A small cylindrical cut-off with the radius of 0.51m was used to replace the presence of the hub. All dimensions are presented in the Figure 3.1.

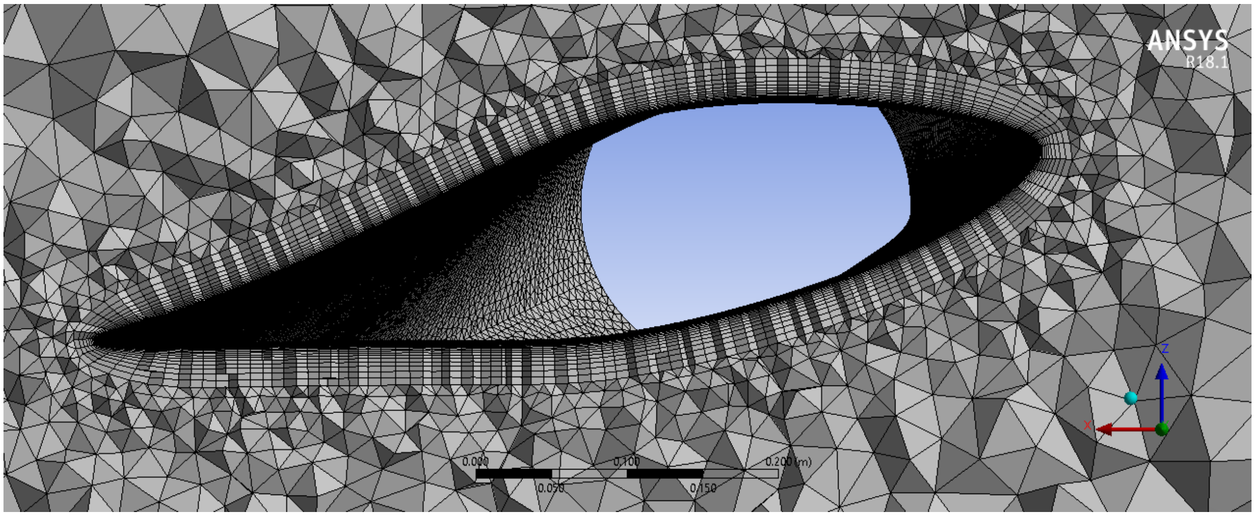
Figure 3.1: Fluid domain (units: meters)



3.3 Meshing

As it was noted previously, fluid dynamic studies of the NREL Phase VI wind turbine have used different meshing methods and software packages. It is generally accepted that structured mesh is more economical to use as it reduces total number of elements compared to an unstructured one. However, the major drawback of this type of mesh is that it requires great effort to construct, which is not the case for this study, as it practically impossible to automate. For that reason, unstructured mesh was used. Unstructured mesh also requires mesh refinement in the zones where complex fluid flow occurs. This include blade surface and, to certain extent, regions surrounding it. ANSYS Fluent built-in meshing tool can insert face sizing and inflation layers on top of the blade surface. Also, periodic boundary conditions require mesh matching at the regions where rotation will occur, this includes two bottom faces of the fluid domain shown in the figure 3.1. The resulting unstructured mesh is presented in the figure 3.2.

Figure 3.2: Mesh around wind turbine blade profile



When applying inflation layers, it is important to have sufficiently small first layer thickness in order to catch unstable flow effects along the surface. Thickness of 0.00001m was used to maintain Y+ value below 1, that will help to catch any boundary layer separation. Mesh refinement study will be presented in the results section.

3.4 Numerical Model Setup

3.4.1 Governing Equations

Since MRF approach is used to simulate wind turbine operation a special formulation of the governing equation must be handled. When simulating rotation 3D Navier-Stokes equations in the rotating frame of reference are formulated in the following way:

Conservation of mass

$$\frac{\partial\rho}{\partial t} + \nabla \cdot \rho\overrightarrow{v_r} = 0 \quad (3.1)$$

Conservation of momentum

$$\nabla(\rho\overrightarrow{v_r}\overrightarrow{v_r}) + \rho(2\vec{\omega}\times\overrightarrow{v_r} + \vec{\omega}\times\vec{\omega}\times\vec{r}) = -\nabla p + \nabla\overline{\tau_r} \quad (3.2)$$

Where

$\overrightarrow{v_r}$ - relative velocity

$\vec{\omega}$ - angular velocity

$\vec{\omega}\times\overrightarrow{v_r}$ - Corilois acceleration

$\vec{\omega}\times\vec{\omega}\times\vec{r}$ - Centripetal acceleration

$\overline{\tau_r}$ - viscous stress

This set of equations do not include energy equation term because there is no heat transfer and fluid properties will not change during the simulation and the flow is considered to be incompressible. This will require special solution method that will be appropriate for these set of conditions. The solver was set to be pressure based steady state with the coupled pressure-velocity scheme as it allows for robust solution. This solution scheme is iterative, meaning that all discretized equations are solved until certain level of convergence is reached. The accuracy for spatial discretization of pressure, momentum and turbulent kinetic energy was set to second order with second order upwind for the latter two. Pseudo-transient option with high order term relaxation was applied to reduce the computational time [24]. These settings simplify the calculation process and significantly cuts off computational time

Because simulating a rotating blade involves unsteady aerodynamic effects choosing proper turbulence model is the subject of great significance. After testing different turbulence models it was evaluated that the best results were obtained when employing realizable k-epsilon model with Menter-Lechner near-wall treatment as it is best suited for flows that involve rotation. Equation 3.3-3.5 represent this model [24].

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_i}(\rho k u_i) - \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] = G_k - \rho\varepsilon + S_{near-wall} \quad (3.3)$$

$$\frac{\partial}{\partial t}(\rho \varepsilon) + \frac{\partial}{\partial x_i}(\rho \varepsilon u_i) - \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] = C_{1\varepsilon}\frac{\varepsilon}{k}G_k - C_{2\varepsilon}\rho\frac{\varepsilon^2}{k} \quad (3.4)$$

$$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon} \quad (3.5)$$

where

G_k - turbulence kinetic energy due to the mean velocity gradients

k – turbulent kinetic energy

ε - turbulent dissipation

μ - air viscosity

ρ – air density

μ_t - eddy viscosity

u - velocity component

$S_{near-wall}$ - active only at low Reynolds number values

$C_{1\varepsilon} = 1.44, C_{2\varepsilon} = 1.92, C_\mu = 0.09, \sigma_k = 1, \sigma_\varepsilon = 1.3$ - constants

This model showed better compliance with the NREL test data as this turbulence model is best suited for flow with rotation present in it [24].

3.5 Boundary conditions

The computational domain described in the section 3.2, depicts the geometry and named selections, that represent boundaries of the domain. The origin of the global coordinate system is located at the hub of the wind turbine. The description of the boundary conditions is presented in the table 3.1.

Table 3.1: List of boundary conditions

Boundary conditions	
Inlet	
Type	Velocity inlet
Initial Gauge Pressure (Pa)	0
Velocity (m/s)	5, 10, 15
Direction	–Z
Turbulence Intensity (%)	5
Turbulent Viscosity Ratio	10
Outlet	
Type	Pressure Outlet
Gauge Pressure (Pa)	0
Backflow Turbulent Intensity (%)	5
Backflow Turbulent Viscosity Ratio	10
Blade surface	
Type	Wall
Shear condition	No Slip
Farfield	
Type	Symmetry
Periodic Interfaces	
Type	Rotational
Offset (deg)	180^o
Rotational Velocity (RPM)	72

Since the problem is simplified to the half-cylinder domain with the imposed rotation around Z-axis, two interfaces shown in the Figure 3.1 are used to represent rotation of the domain. This includes setting offset of 180^o, because wind turbine has only two blades. The rotational speed is set to steady 72RPM in the positive Z direction. The idea of using periodic interfaces is to imitate the rotation of the blade around the global Z axis. The direction of the incoming wind is in the negative Z direction, and inlet boundary condition is set to velocity inlet. The turbulence intensity of the flow was set to a medium value of 5%. Outlet is treated as pressure outlet with the default gauge pressure of 0Pa. The top surface of the domain or the farfield is treated as a symmetry boundary conditions. The small cylindrical cut out, which represents the hub of the wind turbine, is also declared as symmetry boundary condition. Similar practice was used in [22]. The blade surface is defined as a wall with no slip conditions. The interior of the domain is represented by the air, which has properties listed in the table 3.2.

Table 3.2: Air properties

Air properties	
Density (kg/m3)	1.225
Viscosity (kg/m-s)	$1.7894*10^{-5}$
Temperature (K)	288.16

Chapter 4 – NREL Phase VI blade modeling

4.1 Blade characteristics

NREL Phase VI wind turbine blade was modeled in Solidworks using information provided in the NREL report [7]. From the Figure 2.10 blade comprised of three main sections: root, transitional region and NREL s809 airfoil profile blade. Latter region starts at 1.257m from the hub and ends at 5.029m. The detailed information about this region is presented in the Table 4.1. Chord length decreases toward the end of the blade and local twist angle changes in the clockwise direction. Presented data have information about twist angle and chord length only at limited number of span stations which might not be enough for future calculations. For this reason, this values were used to create cubic spline that resembles twist and chord distribution.

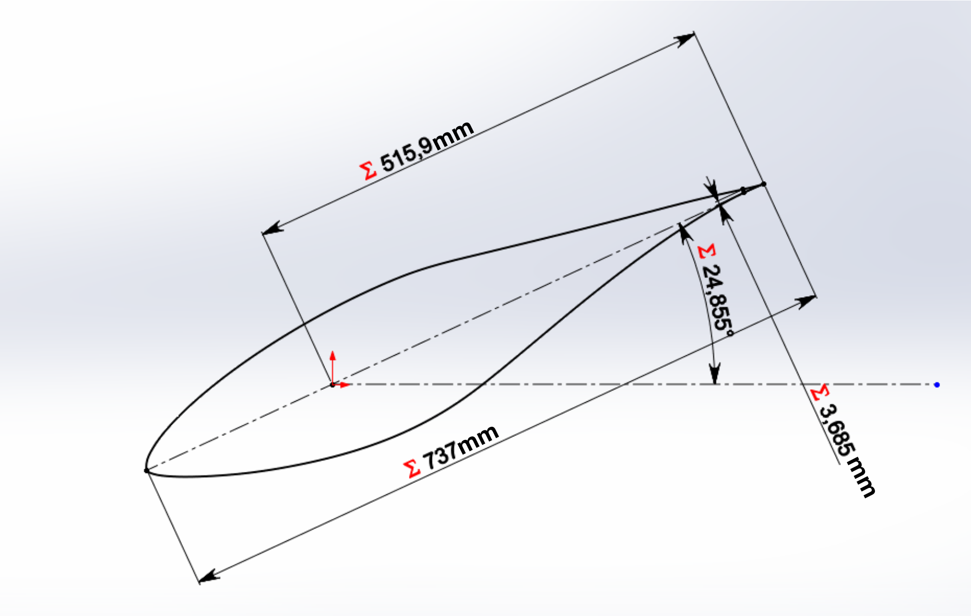
Table 4.1: Twist and chord distribution NREL s809 airfoil profile region

Radial distance (m)	r/R (5.029m)	Chord (m)	Twist (deg)	Thickness
1.257	0.25	0.737	20.04	0.154
1.343	0.267	0.728	18.074	20.95%
1.51	0.3	0.711	14.292	20.95%
1.648	0.328	0.697	11.909	20.95%
1.952	0.388	0.666	7.979	20.95%
2.257	0.449	0.636	5.308	20.95%
2.343	0.466	0.627	4.715	20.95%
2.562	0.509	0.605	3.425	20.95%
2.867	0.57	0.574	2.083	20.95%
3.172	0.631	0.543	1.15	20.95%
3.185	0.633	0.542	1.115	20.95%
3.476	0.691	0.512	0.494	20.95%
3.781	0.752	0.482	-0.015	20.95%
4.023	0.8	0.457	-0.381	20.95%
4.086	0.812	0.451	-0.475	20.95%
4.391	0.873	0.42	-0.92	20.95%
4.696	0.934	0.389	-1.352	20.95%
4.78	0.95	0.381	-1.469	20.95%
5	0.994	0.358	-1.775	20.95%
5.029	1	0.355	-1.815	20.95%

4.2 Blade profile

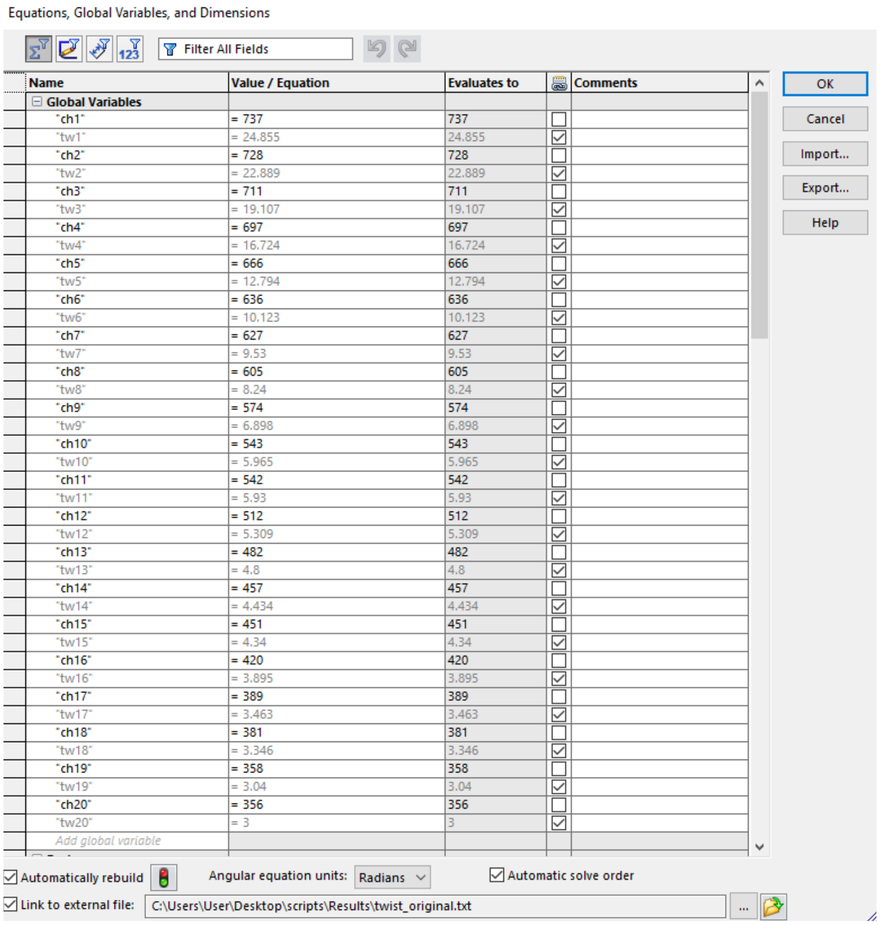
Modeling of the main part of the blade, which is NREL s809 profile region, was performed by creating individual sketches that replicates the shape of s809 airfoil and have the dimensions listed in the Table 4.1. As an example, the sketch of the profile of the first section is shown in the Figure 4.1. These sketches are located on the planes that are distant from the origin by the corresponding to the given span station magnitude. After all sketches are created they are joined together using loft function in the Solidworks. This creates smooth transition between every section. The transition region is also created using this feature, only in this case the first section of s809 portion is joined with the circular profile of the cylindrical region.

Figure 4.1: Sketch of the first section of s809 profile



Because every section has its own chord length and twist angle it is necessary to declare them as global variables. This will allow to vary them both inside and outside of the program, which is crucial in creating an automation routine for Solidworks. Once declared, every parameter is listed in the equation manager (Figure 4.2). Then they can be exported in separate file and current values will be linked to it. This means that Solidworks updates the variables if they were altered in the file. There are two main variables for each section: chord length and twist angle.

Figure 4.2: Equation manager



Equations, Global Variables, and Dimensions

Name	Value / Equation	Evaluates to		Comments
⊟ Global Variables				
"ch1"	= 737	737	☐	
"tw1"	= 24.855	24.855	☑	
"ch2"	= 728	728	☐	
"tw2"	= 22.889	22.889	☑	
"ch3"	= 711	711	☐	
"tw3"	= 19.107	19.107	☑	
"ch4"	= 697	697	☐	
"tw4"	= 16.724	16.724	☑	
"ch5"	= 666	666	☐	
"tw5"	= 12.794	12.794	☑	
"ch6"	= 636	636	☐	
"tw6"	= 10.123	10.123	☑	
"ch7"	= 627	627	☐	
"tw7"	= 9.53	9.53	☑	
"ch8"	= 605	605	☐	
"tw8"	= 8.24	8.24	☑	
"ch9"	= 574	574	☐	
"tw9"	= 6.898	6.898	☑	
"ch10"	= 543	543	☐	
"tw10"	= 5.965	5.965	☑	
"ch11"	= 542	542	☐	
"tw11"	= 5.93	5.93	☑	
"ch12"	= 512	512	☐	
"tw12"	= 5.309	5.309	☑	
"ch13"	= 482	482	☐	
"tw13"	= 4.8	4.8	☑	
"ch14"	= 457	457	☐	
"tw14"	= 4.434	4.434	☑	
"ch15"	= 451	451	☐	
"tw15"	= 4.34	4.34	☑	
"ch16"	= 420	420	☐	
"tw16"	= 3.895	3.895	☑	
"ch17"	= 389	389	☐	
"tw17"	= 3.463	3.463	☑	
"ch18"	= 381	381	☐	
"tw18"	= 3.346	3.346	☑	
"ch19"	= 358	358	☐	
"tw19"	= 3.04	3.04	☑	
"ch20"	= 356	356	☐	
"tw20"	= 3	3	☑	
Add global variable				

☑ Automatically rebuild

Angular equation units: Radians

☑ Automatic solve order

☑ Link to external file: C:\Users\User\Desktop\scripts\Results\twist_original.txt

The final design of the blade with all sections highlighted is displayed on the Figure 4.3. It can be noted that sections are not equally spaced between each other. It may cause some difficulties since span section that were chosen in the optimization do not coincide with the sections from which the model was build.

This problem is resolved by utilizing splines and linked file to reflect any possible changes dictated by the optimization tool.

Figure 4.3: NREL Phase VI blade with indicated span stations

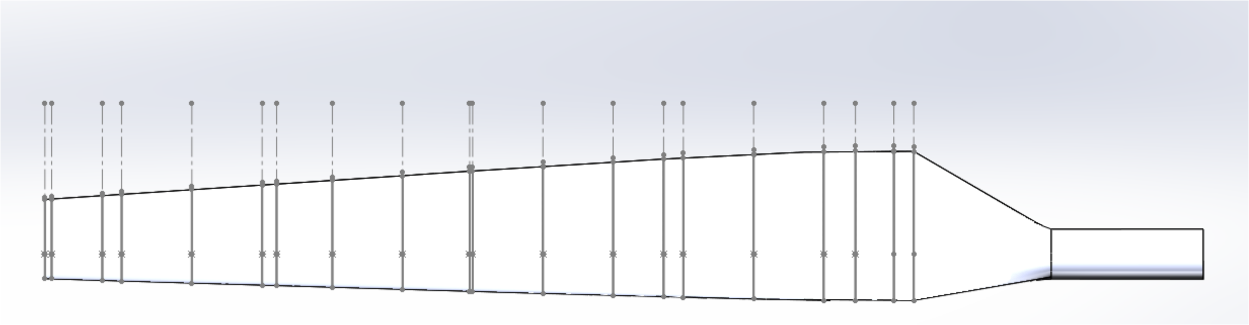
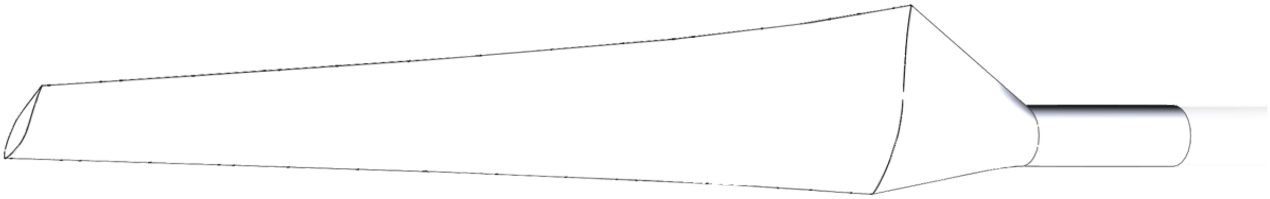
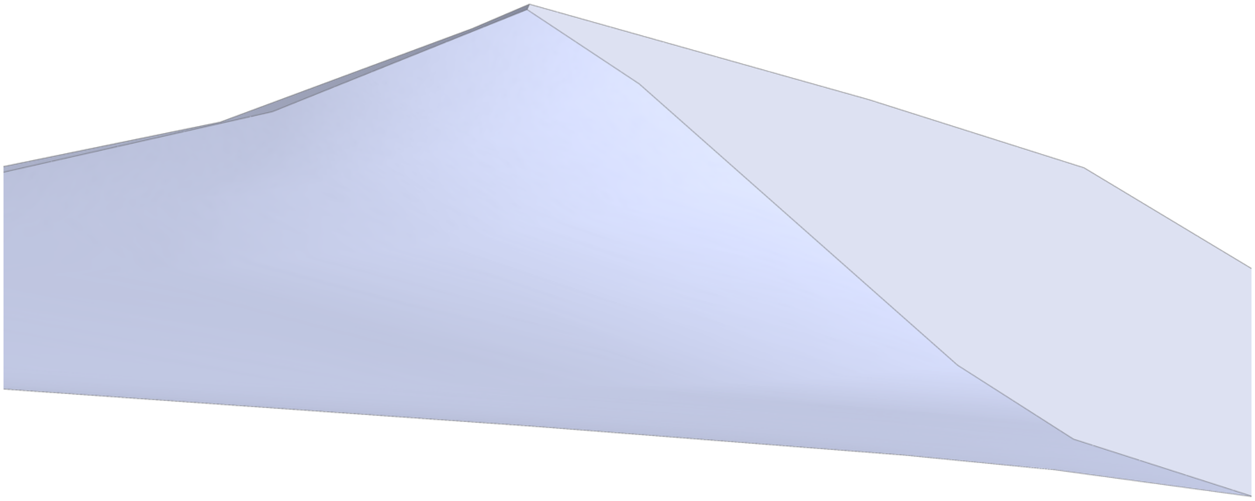


Figure 4.4: NREL Phase VI blade



The complete rendition of the blade design is presented in the figure 4.4. Blade was oriented in such manner to satisfy 3^{o} pitch angle, i.e. the top section has local twist measurement of 3^{o}. This forces all local twist angles to change their magnitudes in accordance. The original design of the blade has sharp edge, which can cause problems during calculations because it can cause singularities. Therefore, trailing edge was cut to have blunt end. The length of cut for individual section is equal to 0.5% of chord length. The appearance of this modification is demonstrated in the Figure 4.5.

Figure 4.5: Blunt trailing edge of the blade



Together with blade, the full fluid domain was also constructed in the Solidworks. This was done to avoid any tinkering with the blade model in the ANSYS Design Modeler. The solid part of the fluid domain is air, that is why blade was converted into an empty space. This was achieved by separating blade domain from fluid domain. In the end, the final 3D model is the half-cylinder fluid domain with the blade, characterized as void, contained inside of it.

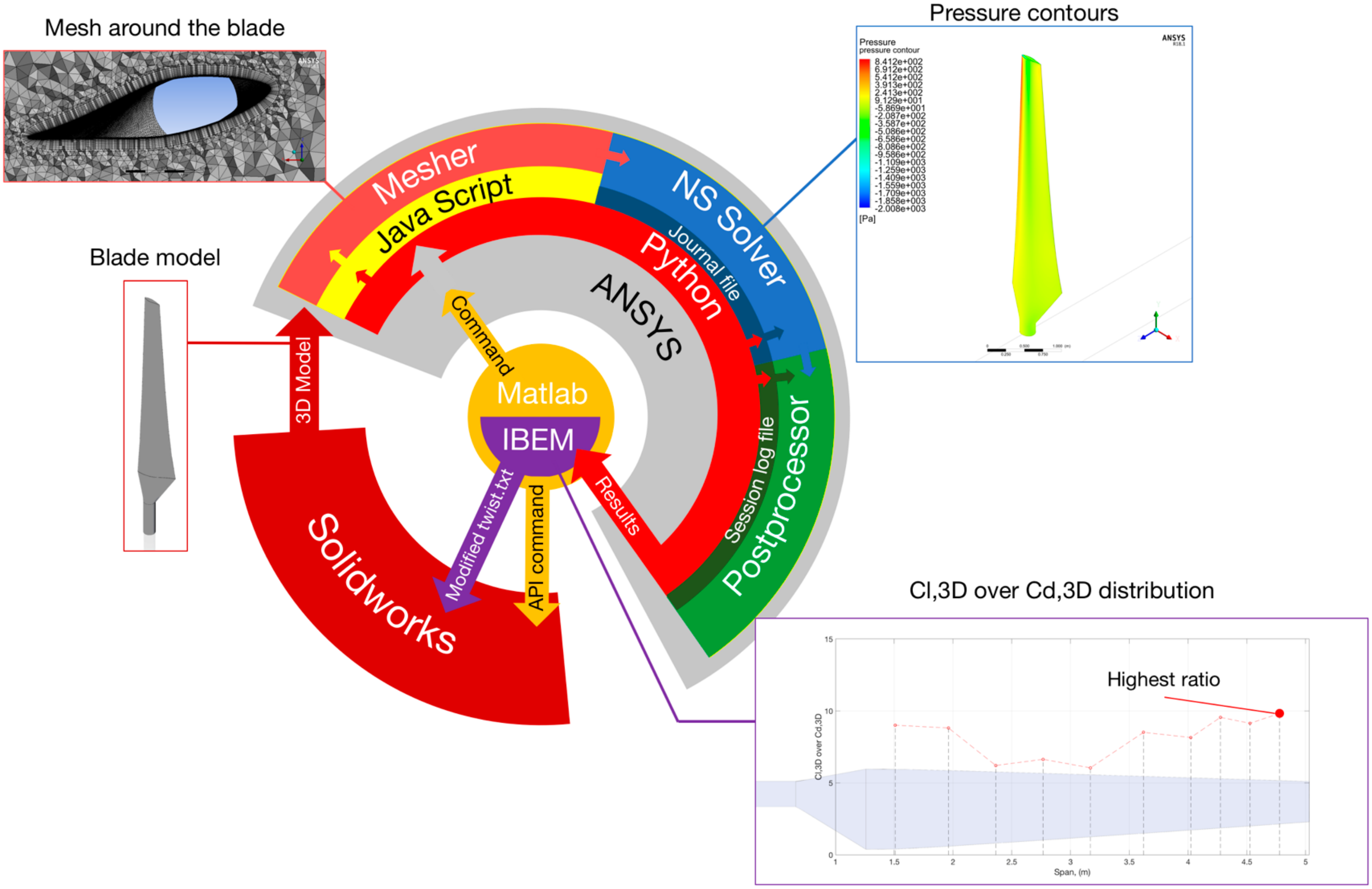
Chapter 5 – Automation process development for Design Optimization

The optimization process requires constant adjustments of the parameters and features of the model. These parameters variation are performed on the basis of the results obtained from the simulation and predefined conditions. All component in the optimization environment must be intertwined between each other. Thus constant communication between components is crucial. In the given case, system consists of three main parts: geometry builder, CFD solver and optimizer function. These components must constantly transfer and accept data from one another. This will require meticulous study of how each system operates on the subshell level, that is understanding how system works under graphical user interface (GUI). In many cases, each software package comes with the application programming interface (API), which allows users to access internal procedures of the given program directly. This provides an opportunity to create custom functions and operations tailored to the needs of the programmer. Using this approach will help to avoid repetitive manual labor of setting up and updating elements in the particular program each time adjustment is needed.

5.1 General framework

As it was mentioned earlier, three major parts will be used in the automation process. The first of them is the software responsible for generating 3D model and updating the blade geometry. It must contain all the necessary information upon which wind turbine blades will be generated and adjusted. The software of choice for this is the Solidworks as it has well documented API and allows to store model features in the external file, that can be modified by other programs in the environment. Additionally, it allows to fastly and efficiently build solid models, that later can be used in the manufacturing. All CFD calculations will be performed inside ANSYS as it contains mesher and solver inside one package. In addition, it has direct compatibility with the Solidworks. All results retrieved from the solver are then transferred to the core component of the system – Matlab. It does all data interpretation and optimization steps by running individual scripts. Every separate component is manipulated by means of its own scripting capabilities, but the whole framework is managed by Matlab. It handles calling external objects with the help of Windows Component Object Model (COM) interface and Windows command prompt. Solidworks is launched by creating a COM server with the Solidworks application process ID, which is “'SldWorks.Application'”. This action allows to manipulate and call all the necessary programming methods and interfaces associated with the opened application that are well described in the API manual [25]. Command prompt is used to interfere with ANSYS. It natively supports command prompt functionality, such as launching script file of interest, and closing the application when all calculations are done [26]. Moreover, these commands can be sent to Windows command prompt directly from Matlab. All together, this gives a possibility of calling all essential components in sequential fashion from the core. The layout of the automation process is presented in the Figure 5.1. The similar automation layout was implemented by the [27]. They used ANSYS and Solidworks to evaluate various designs of the robotic hand.

Figure 5.1: Schematic representation of the automation process



More detailed description of inner workings of each component will be presented in the following sections.

5.2 Solidworks

The first element of the process deals with generating and updating the 3D model of the blade according to changes imported from the optimizer. The wind turbine blade was generated using airfoil profiles at various spanwise locations, which have their own local twist angles (tw) and chord lengths (ch). These two parameters are the global variables, which are contained in the equation manager inside the Solidworks environment, and can be exported as a text file (see the appendices). Moreover, this file can be linked to the model geometry. Thus, by varying values of global variables in the linked file user will change the shape of the wind turbine blade. In our case, the variable of interest will be the magnitude of local twist angle (tw). It is important to note, that Solidworks do not support negative values for angles, therefore they must be converted to positive equivalent angles. Such conversion must be performed at the stage where new local twist angles are formed and written to the "twist.txt" file. Solidworks component, which is presented in the Figure 5.1, is initiated from the core component – Matlab. To perform this Matlab creates COM server. Then it declares two variables specific to Solidworks API. The first of them will include all methods on the application level and will be of the type "SldWorks". The second variable will contain all operations that can be performed on the model. These two variable include methods that are unique only to their native specifications. The methods related to the application are responsible for launching, opening a part and closing the program. Methods that are linked to a part file are dealing with any changes performed on the part itself, i.e. updating global variables from equation file, rebuilding geometry features and saving the document. All methods used in this work are listed in the Table 5.1.

Table 5.1: List of methods used specific to variables

Application variable methods	Model variable methods
OpenDoc	GetEquationManager
	GetCount
ExitApp	LinkToFile
	UpdateValuesFromExternalEquationFile
	EditRebuild3
	Save

- "OpenDoc" – opens documents stored at predetermined path
- "ExitApp" – closes current session and exits application
- "GetEquationManager" – calls equation manager object which contains all equations and global variables
- "GetCount" – counts the number of equations stored in the equation manager object
- "LinkToFile" – checks whether the specified equation or global variable is linked to an external file
- "UpdateValuesFromExternalEquationFile" – updates value of a given equation or global variable

- “EditRebuild3” – rebuilds part according to changes made
- “Save” – saves the document to the current folder, where it stored

Detailed code that invokes all methods described here is presented in the appendix.

5.3 ANSYS

ANSYS workbench environment is constructed from various blocks and containers that accomplishes a specific task. A typical arrangement of these containers is shown in the Figure 5.2. In general, CFD simulation is carried out in multiple steps. Firstly, the geometry of the blade and fluid domain must be constructed or imported. Secondly, mesh for a given geometry must be constructed. Thirdly, physics setup of the problem must be defined and solution initiated. The last sage is exporting generated results from ANSYS workbench via postprocessor. These components have their own unique way executing operations inside of them.

Figure 5.2: ANSYS Workbench components

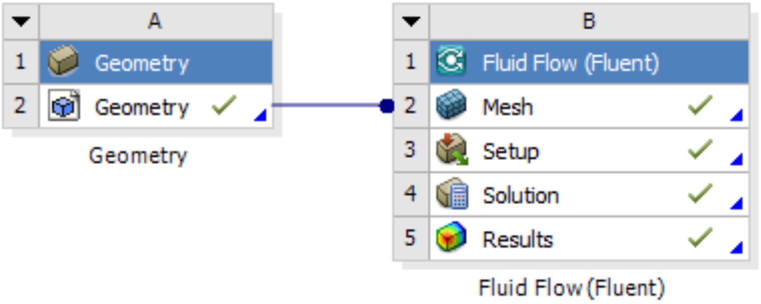


Table 5.2: Scripting capabilities of ANSYS workbench components

Workbench components	**Native Scripting Language**	**Supports Scripting with SendCommand**	**Supports Journaling with SendCommand**
Mechanical APDL	APDL	Yes	
Mechanical	JScript	Yes	
CFX	CCL	Yes	Yes
Fluent	Scheme	Yes	Yes
Aqwa	JScript	Yes	
Autodyn	Not Available		
CFD-Post	CCL	Yes	Yes
FE Modeler	JScript	Yes	
DesignModeler	JScript	Yes	
Meshing	JScript	Yes	
Polyflow	Not Available		
IcePak	Not Available		
ICEM CFD	TCL	Yes	No

Information presented in the Table 5.2 shows how interactions on the individual component level are performed. Highlighted parts are the ones that will be used in the automation process. Since these are the subcomponents of the workbench the main portion of scripting will be done here. Workbench natively supports python programming language and the principal communication with subcomponents will occur here. Figure 5.1 illustrates how ANSYS communicates with subcomponents through the python script. Individual scripts specific only to that component must be implemented in its native scripting language. As demonstrated in the Figure 5.1, meshing of the geometry is performed by Java Script. The script contains mesh specification such as element sizing, body sizing, inflation layers and mesh element sizes. In order to apply this features in Java Script language the specific methods for setting mesh refinement parameters must be extracted from the libraries stored in the program installation directory and can only be retrieved manually. However, this holds true only for mesher module, other modules can be set up using journaling features, which allows user to record sequence of actions and save it as a readable file, that can be used by workbench later. From the Figure 5.1, NS Solver and Postprocessor components support journaling capabilities. The recorded journals are launched from the main Python script. NS Solver component is responsible for physics setup and executing calculation. Postprocessing component exports pressure data at the specified spanwise locations as Comma Separated Value (CSV) files, which later are used by Matlab.

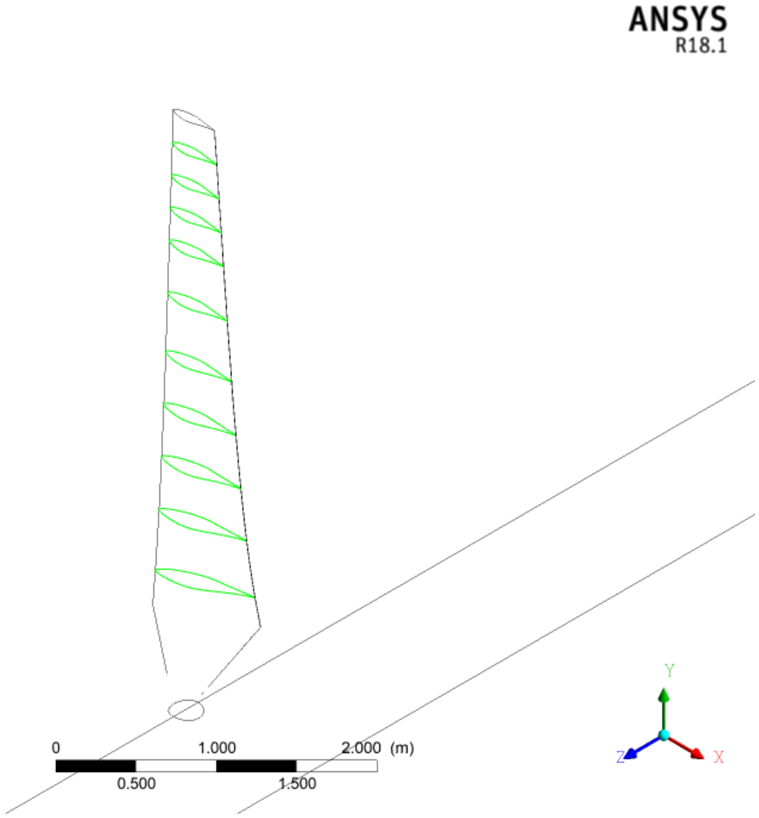
As it was stated earlier, meshing automation is executed by inserting mesh sizing options by using Java Script. In order to streamline this process named selection feature is used. Named selections designate various parts of the fluid domain which will be later used in setting mesh sizing as well as boundary conditions. The following named selections are utilized in the model:

- NS_inlet – inlet of the fluid domain
- NS_outlet – outlet of the fluid domain
- NS_symmetry_top – far field of the domain
- NS_symmetry – cylindrical representation of the hub
- NS_interface_1 – periodic symmetric interface
- NS_interface_2 – periodic symmetric interface
- NS_blade – blade surface

These selections are easily transferable from Solidworks to ANSYS environment with the help of the add-ins. Solidworks traces any changes in the shapes and keeps the named selections updated. This is especially useful for constantly changing geometric features of the model that later will be meshed. In the meshing process solver preferences must be set to CFD and various meshing options must be implemented on different elements of the domain. A face sizing option has to be used on the blade surface in order to increase accuracy of the calculations. Also, it needs inflation layers to be inserted. The periodic boundary conditions, described in the section 3.5, require mesh matching between interfaces in order to use MRF approach later in the simulation stage. For this reason, meshing script must include all of the aforementioned meshing options to the blade and domain. This is done by specifying named selection associated with it and choosing the value of the parameter, i.e. element size, that will produce desirable quality mesh.

After the mesh is created, Fluent solver settings are setup by executing journal file, that was recorded before by manually putting solution parameters in a sequential fashion. Here, turbulence model, boundary conditions and number of iterations are determined. The exact parameters that were used in this stage are described in the section 3.4. The journal also includes initialization of the solution and saving the project files. When simulation is completed, results are extracted by using Postprocessor module inside the workbench. This includes pressure data at desirable spanwise locations. In our case these span locations are at the 30, 39, 47, 63, 72, 80, 85, 90 and 95% of length of the blade (Figure 5.5). The pressure data is extracted by creating horizontal planes and using polyline function to create an intersection curve between blade surface and a given plane. This operation must be carried out for all ten locations. This polyline will hold pressure reading from CFD simulation and will be used later in the IBEM calculations. The measurements are exported as a CSV file. Postprocessor also has capabilities of recording the journals, that can be fed into mainframe script, but it is different from the journal files used for NS Solver module. In the end ANSYS automation routine will consist of one main python script called “ansys_process.wbjn”, that will trigger “meshgen.js” meshing routine script, then will invoke “setup specs.jou” journal, which contains all physics preferences and will finish by sending separate commands to the Postprocessor module.

Figure 5.5: Span sections used in the optimization process



5.4 Matlab

Matlab component of the system, shown in the Figure 5.1, is the core module of the whole automation framework. It controls in what sequence each module is launched and ensures appropriate data transfer between elements of the automation process. It also handles data interpretation, so that each subsystem receives information in understandable to it form. As an example, it was mentioned earlier in this work that Solidwork can handle only positive values of the twist angles, but which are not appropriate for IBEM code and must be adjusted accordingly. Therefore, parameters should be constantly converted back and forth between these subsystems. Results obtained from the CFD postprocessor are in the table form and contain only raw pressure data. In this case, Matlab is used to manipulate these reading and convert them to the force and pressure coefficients. This conversion is crucial for IBEM code, since it utilizes thrust and torque coefficients as its input parameters.

One of the main tasks of Matlab subsystem is to read and store values of the local twist angles inside of the “twist.txt” text file. The blade model is built from twenty cross sectional sketches each having its own twist angle (tw) and spanwise location. However, automation process employs only ten sections, that do not overlap the ones used in the construction of the model. This requires creating a cubic spline which stores all possible twist angle values and is based on the spanwise location of the sections from which the 3D model was build. Then created spline is used to interpolate the local twist angles at the span stations defined in the automation process. Thus any changes imposed on ten sections will be extended to twenty sections. In this way distribution of the twist angles is preserved even if fewer sections are employed. This gives a possibility to go back and forth between ten and twenty values of twist angles without compromising on accuracy. Conversion between number of variables is crucial for IBEM and shape builder, because the first one handles ten variables and latter – twenty.

The raw data obtained from ANSYS Fluent is the pressure readings along the cross section of the blade at every span station. Because IBEM code can only handle thrust and torque coefficients, these pressure coefficients values must be translated into suitable magnitudes. The equations that are used to convert values are listed below. Firstly, pressure readings are converted into the pressure coefficients. These readings are evenly spread across the polyline which represents NREL s809 airfoil. The pressure coefficients along lower and upper portions of the blade surface are summed up one by one until it makes a full circle. Calculation is performed both along x and y axis of the local coordinate system (eq. 5.1 and 5.2). This will produce normal and tangential force coefficients. Once these values are obtained torque and thrust values can be evaluated based on the local twist angle (eq. 5.3 and 5.4) [7].

$$C_N = \sum_{i=1}^{\#taps} \left(\frac{C_{p_i} + C_{p_{i+1}}}{2} \right) (x_{i+1} - x_i) \ (5.1)$$

$$C_T = \sum_{i=1}^{\#taps} \left(\frac{C_{p_i} + C_{p_{i+1}}}{2} \right) (y_{i+1} - y_i) \ (5.2)$$

$$C_{TORQUE} = (C_N \sin(tw) + C_T \cos(tw)) \ (5.3)$$

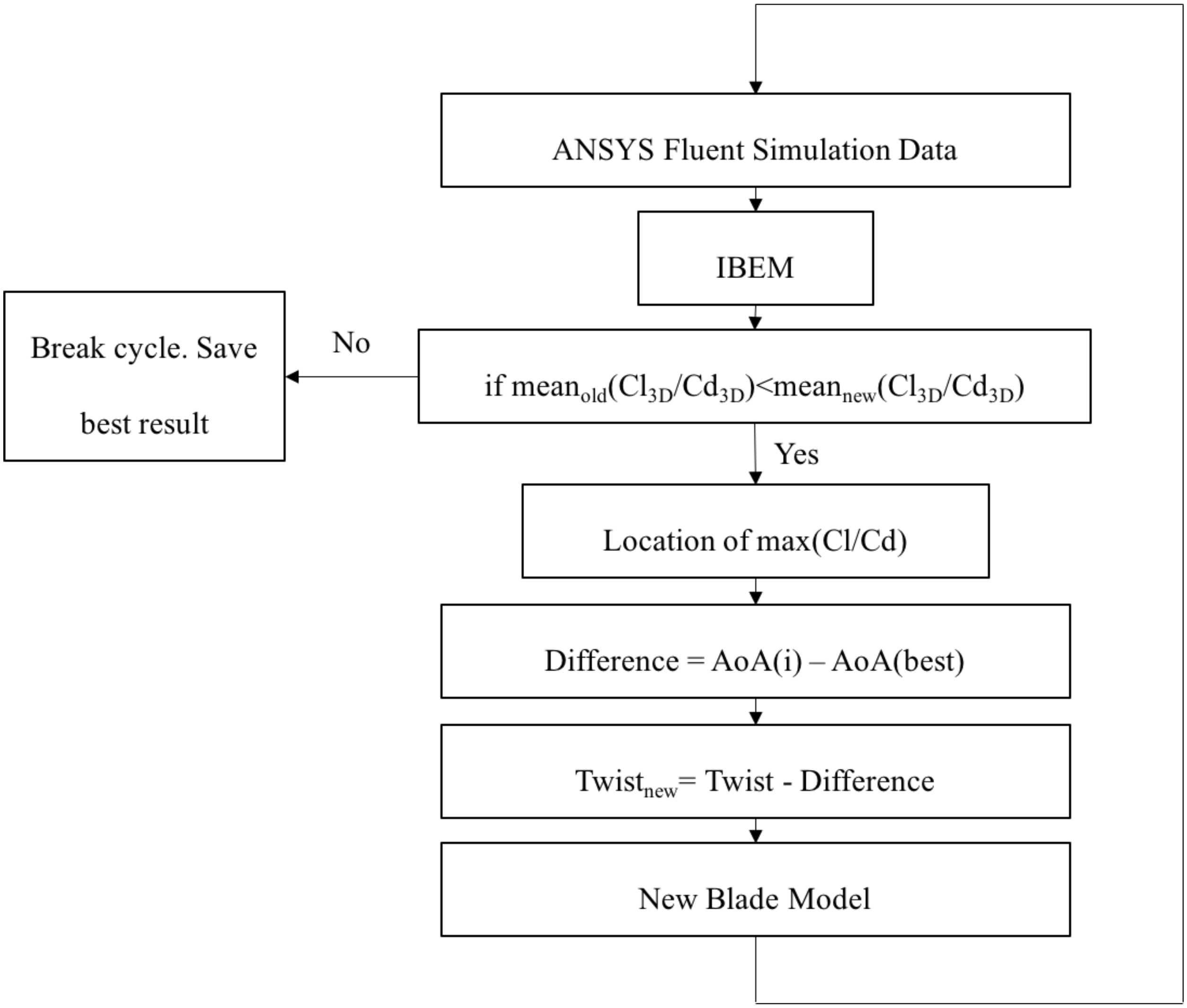
$$C_{THRUST} = (C_N \sin(tw) - C_T \cos(tw)) \quad (5.4)$$

This calculation is implemented into the “datareadcsv.m” script, that reads data from CSV type files and evaluates desirable coefficients.

5.5 3D lift/drag coefficients calculated by IBEM using pressure measurements

IBEM code implemented in Matlab, receives pressure readings from simulation and estimates angle of attack, 3D lift and drag coefficients. These estimations are then used as a foundation on which twist angles are altered. The main principle behind this optimization is the usage of optimal angle of attack at which ratio of lift and drag coefficients is the highest. Once the angle of attack distribution is obtained from the IBEM the location maximal lift to drag ratio is determined. This point will be the location of the optimal angles of attack. According to this angle all other angles are adjusted. The magnitude of adjustment is determined by the difference between the current and the best angle of attacks. Then it is subtracted from local twist angle value and new angle distribution is saved to the “twist.txt” equation file. The flow of this process is depicted in the Figure 5.6.

Figure 5.6: Optimization scheme using IBEM method



The decrease of the local twist angles in the blade geometry will increase AoA for the next iteration. Since optimization scheme considers 3D flow effects this will allow to exploit stall delay phenomena. Stall delays permits higher AoA, thus higher lift coefficients can be achieved.

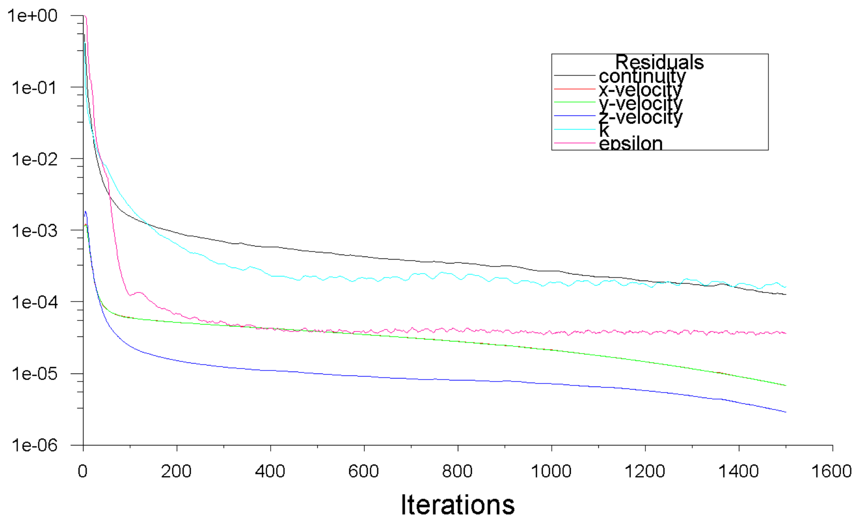
After all adjustments are done, a new model of the blade is formed for further evaluation. Then it again undergoes the same process in order to track any improvements. This process is aimed to maximize average lift over drag ratio, and it stops when there is no bettering of results. When process is finished it saves the best shape and the whole automation process stops.

Chapter 6 – Results and Discussion

6.1 ANSYS Fluent simulation results

After appropriate mathematical model is defined, the number of iteration must be established. For the case of this study, 1500 iterations were deemed to be sufficient enough to produce good results that are in accord with experimental values. Convergence of the residuals of the simulation is depicted in the Figure 6.1.

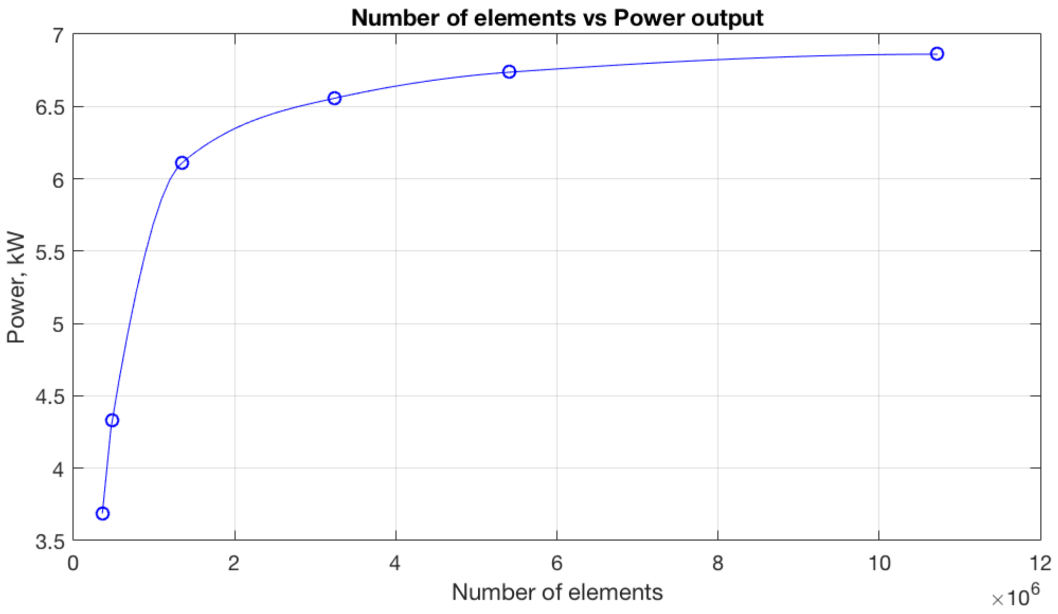
Figure 6.1: Residuals convergence graph



6.1.1 Mesh convergence study

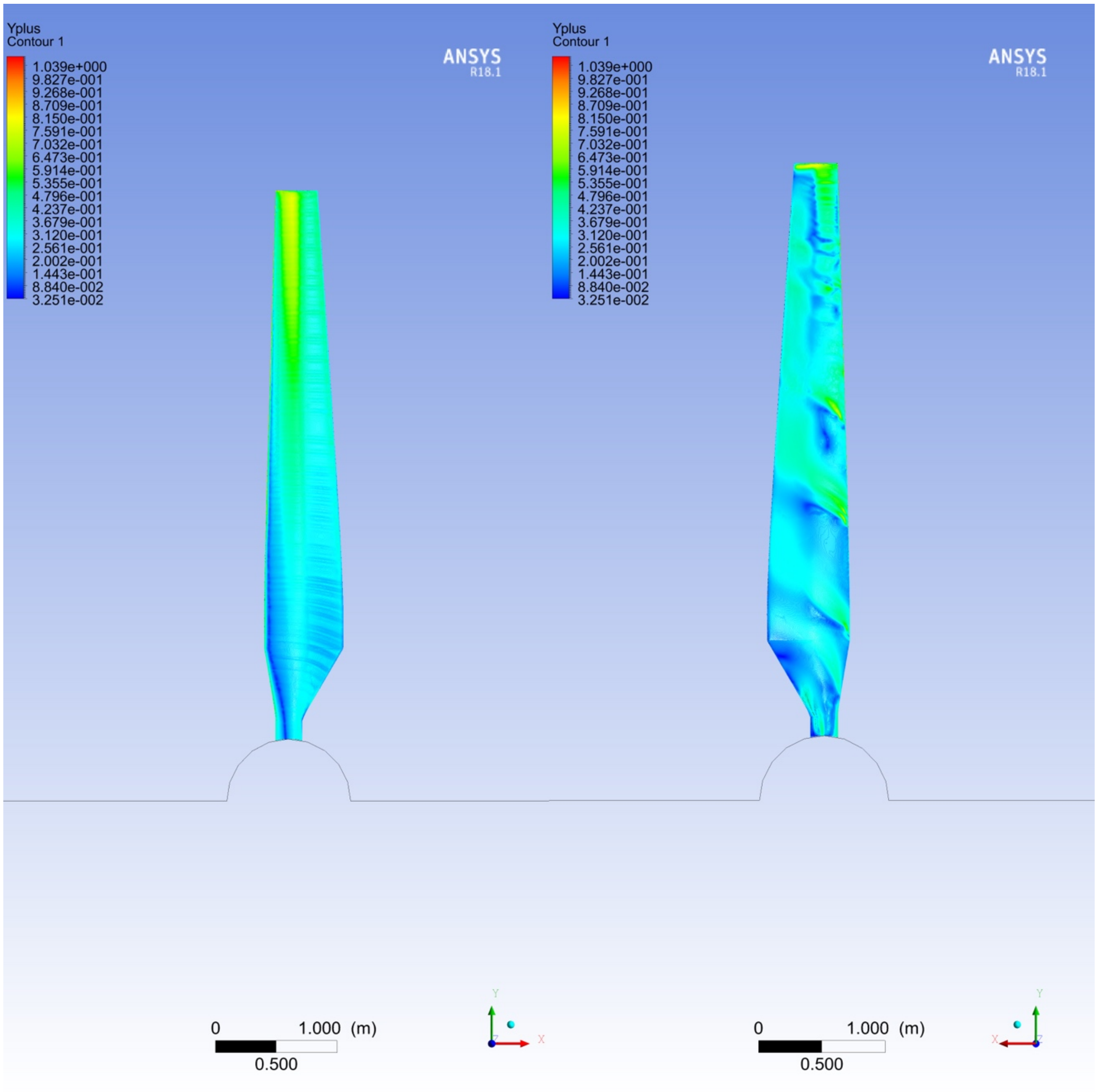
Prior to performing validation of the experimental results a mesh convergence study must be performed. This will guarantee the accuracy of the results and will help to choose optimal number of mesh elements to use in the final stages of the study. In order to do this, the simulation must be started with the default mesh parameters and the power output for this settings must be calculated. After this, some mesh refinement must be done to increase elements count. This procedure is repeated until change in the power output is minimal.

Figure 6.2: Mesh convergence study



The results of the refinement for the given case in demonstrated in the Figure 6.2. The change of the power output drastically diminishes after 6 million elements. Mesh evaluation is performed with 10 m/s wind speed and the power is calculated through extracting a torque on the blade about Z-axis. It was decided to use approximately 10 million elements for this study. Another important important aspect that will ensure the accuracy of the results is Y+ value across the surface of the blade. Final mesh, that was used in the validation process, produces the results shown in the Figure 6.3. Y+ is generally contained below 1, but maximum value is equal to 1.039. This ensures that turbulence model will produce reliable results.

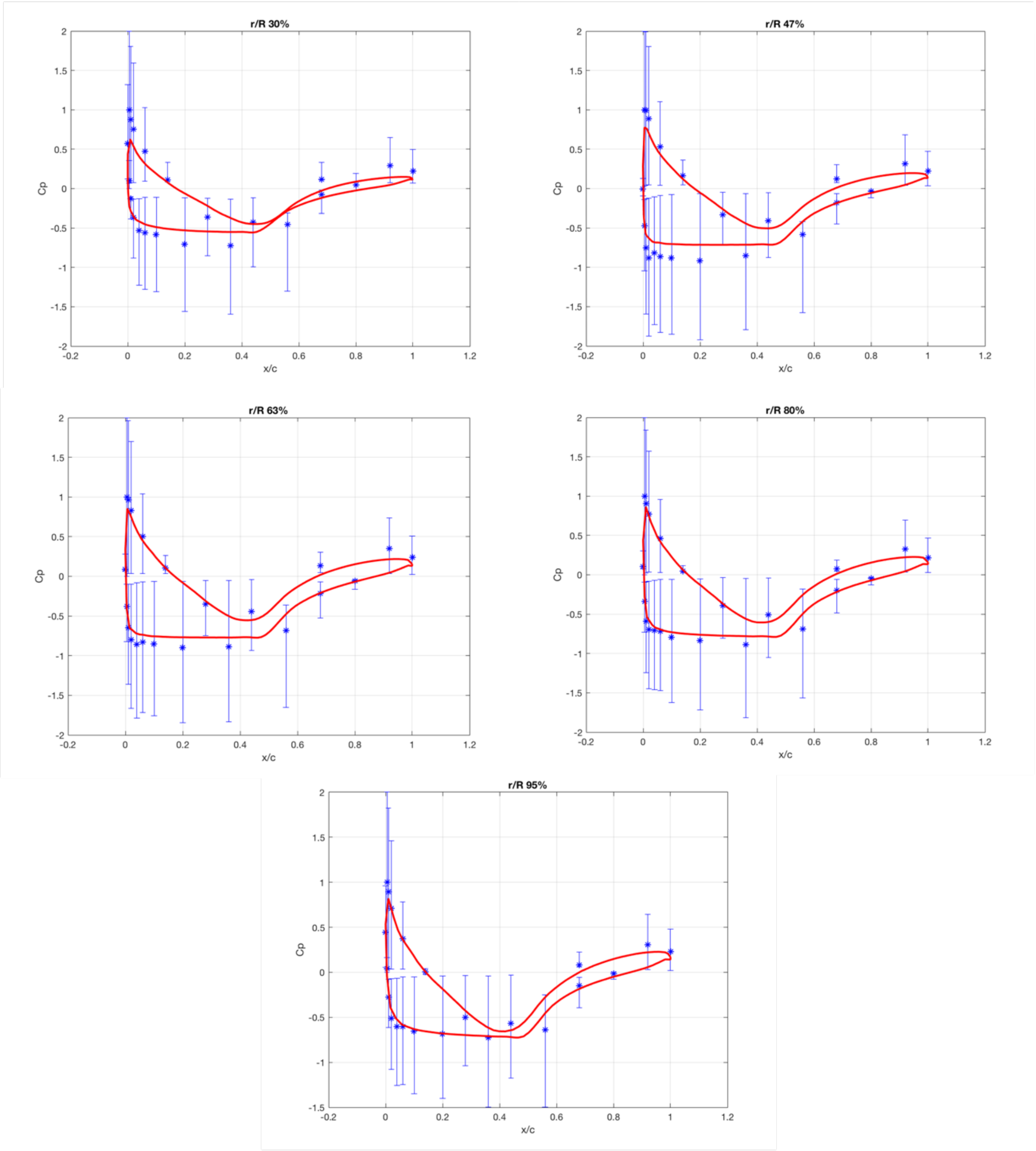
Figure 6.3: Y+ distribution across the surface of the blade. Left – pressure side; right – suction side



6.1.2 Pressure coefficients comparison

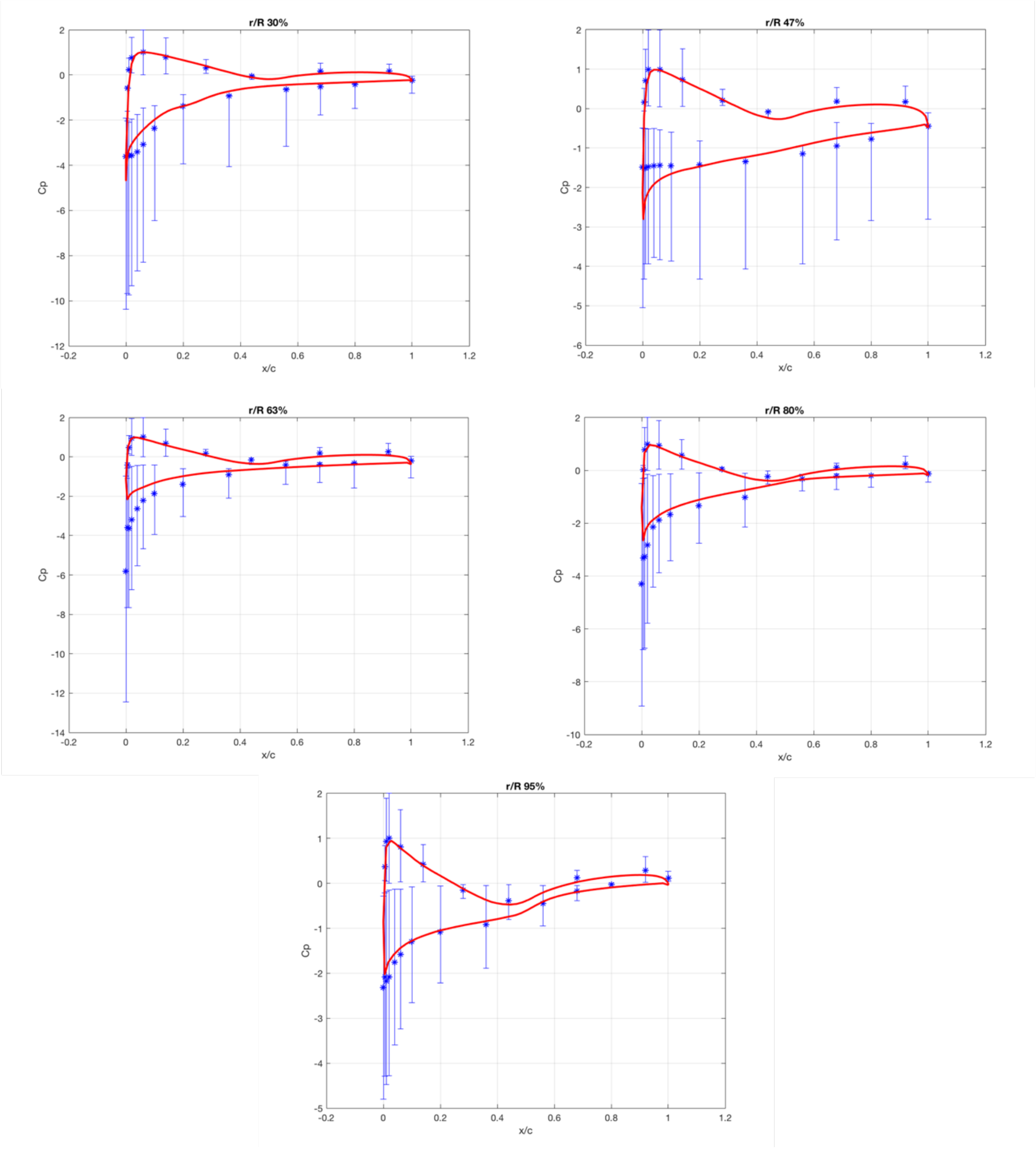
For the optimization procedure it is very important to have correct pressure reading along the blade surface, as it will be used as a basis of calculation in the inverse BEM method. Data from the NREL Phase VI experiment includes detailed pressure readings for five span locations: 30%, 47%, 63%, 80% and 95% of the 5.029m blade. Once the close match with the experimental data is reached, CFD results can be used for further calculations. The confirmation of results was performed at two wind speeds: 5 and 10 m/s and 72 RPM rotational speed. This is because 10 m/s speed was selected to be used in the optimization process as operating condition and CFD model was tailored for this value. Data extracted from NREL data campaigns include 15625 readings for every measurement. This implies that for one pressure tap located at particular span station there are 15625 values of pressure. To account for presented variations error bar plot was used. It has mean values highlighted by the "*" sign and the upper and lower thresholds display maximum and minimum values. The juxtaposition of the results is presented in the Figures 6.4 and 6.5. Red curve shows data obtained from Fluent and the blue stars with bars – NREL tests.

Figure 6.4: Pressure coefficients distribution for 5m/s



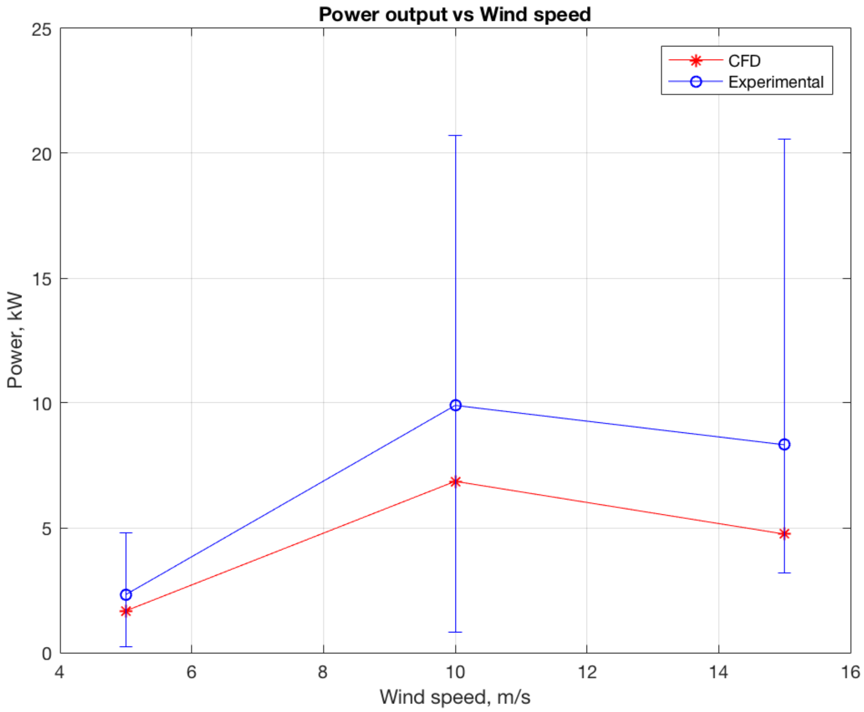
At the lower wind speed computational results come to close agreement with the real data. Both upper and lower surfaces demonstrate similar behavior even at the leading edge (Figure 6.4). However, things start to deviate a little more at the elevated speed (Figure 6.5). It is especially pronounced close to the leading edge on the suction side of the blade. Discrepancies are most notable at the 47 and 63% span stations. Despite this, the readings from simulations are located within a range of possible values measured during experiment.

Figure 6.5: Pressure coefficients distribution for 10 m/s



Another criterion that will show how well computed data relates to the real one is the power output graph. Wind turbine blade is tested under varying wind speeds and produced power is plotted against corresponding speed. Results of these actions are shown in the Figure 6.5. Here NREL results are shown with error bars, to show the variation of the recorded measurements. CFD results underestimate the mean value of the power output evaluated from real test, but it is still within error bar margins and repeats a pattern demonstrated. Wind turbine produces the greatest power and the 10 m/s wind speed. This speed will be used as an optimal design condition in the optimization cycle.

Figure 6.6: Power prediction comparison

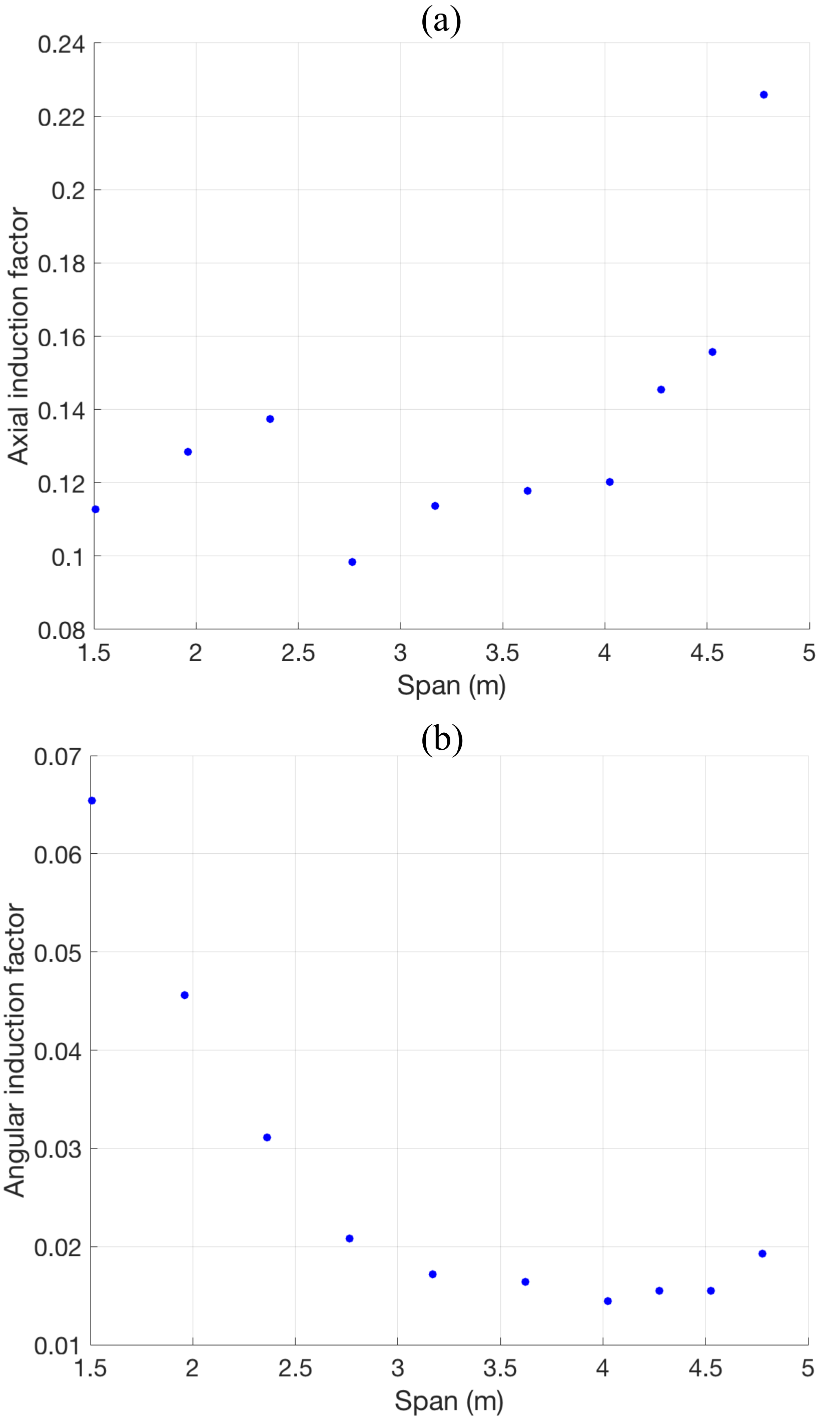


6.2 Optimization results

After running optimization process through the means of automation process described in the methodology section, the results obtained from the IBEM code are presented here. In total only two iterations were enough to determine better design that have the highest mean lift over drag ratio.

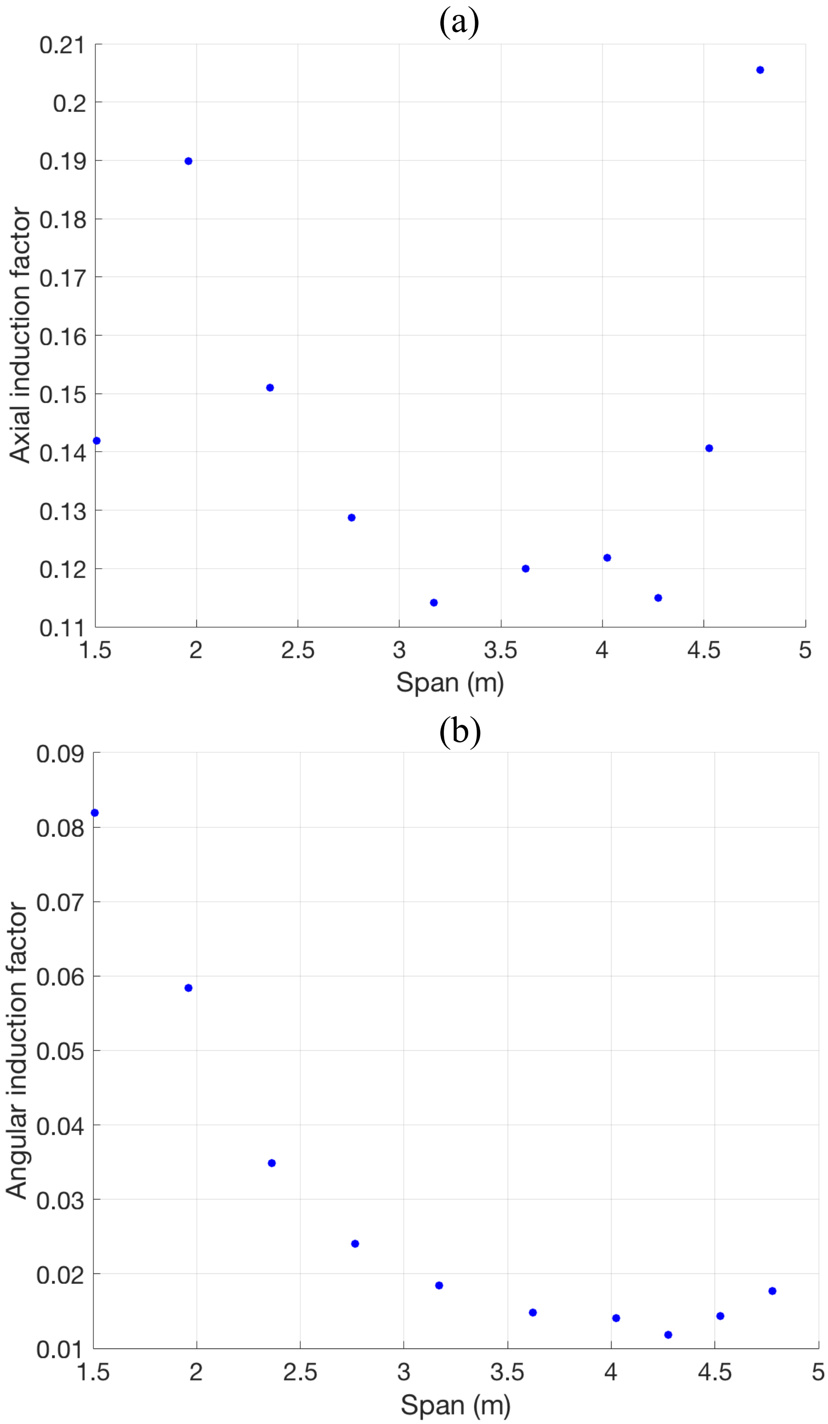
From the theory, axial induction factor should not exceed the value of 0.4 in order to produce more reliable and accurate results. For the original NREL Phase VI blade design both axial and angular factors do not exceed designated threshold. Figure 6.7 displays the distribution of these factors along the blade, where the maximum value for the axial induction factor slightly exceeds 0.22 and 0.06 for angular induction.

Figure 6.7: Axial (a) and angular (b) induction factors distribution of the unaltered design



Automatically modified design also does not show any values above appointed threshold, which is good for the estimation of angles of attack and lift over drag ratios. The distributions corresponding to the optimized design are shown in the Figure 6.8.

Figure 6.8: Axial (a) and angular (b) induction factors of the optimized design



Estimations of the ratio of 3D lift and drag coefficients across the blade of the original model is presented in the Figure 6.9. The highest ratio was located towards the end of the blade, at the last spanwise section. This point is indicated as the highest ratio in the Figure 6.9. This location of this point is the reference point from which difference between AoA will be calculated. Figure 6.10, which shows AoA distribution, indicates this reference point. The mean ratio across the blade is averaged to be 8.188. The optimized design must have a value higher than this.

Figure 6.9: Cl_{3D}/Cd_{3D} ratio distribution for original blade at 10 m/s

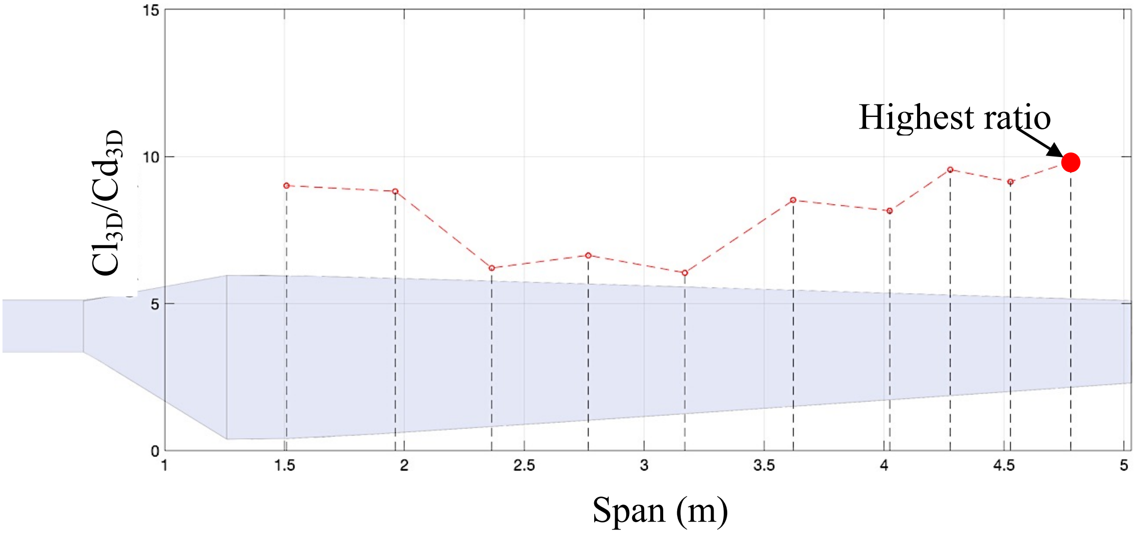
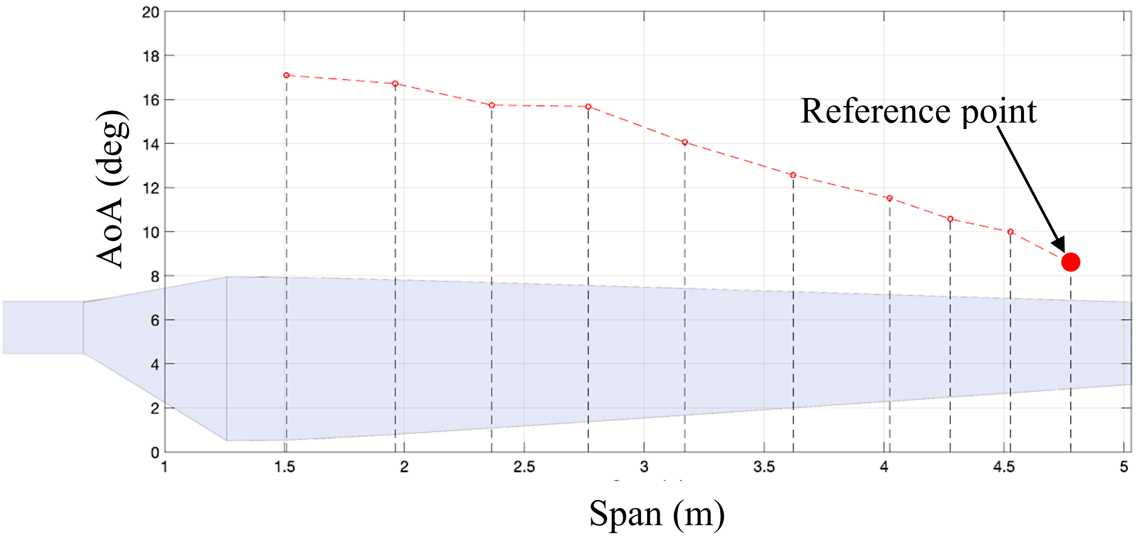


Figure 6.10: AoA distribution for original blade at 10 m/s



The results for the optimized design are depicted in the Figure 6.11 and Figure 6.12. Old results are superimposed on top of the new ones to highlight the differences. Optimization has raised mean lift to drag ratio and now it is equal to 8.8734. The comparison between old and new means is shown by dash-dot line in the Figure 6.11. The ratios of the optimized design remain higher than original blade until the middle of the blade. After this, ratio drops and continues this trend till the tip of the blade. Despite this, mean ratio of the new blade is higher, which is counted as the optimal design, according to the optimization goal.

Figure 6.11: Cl_{3D}/Cd_{3D} ratio distribution for optimized blade at 10 m/s

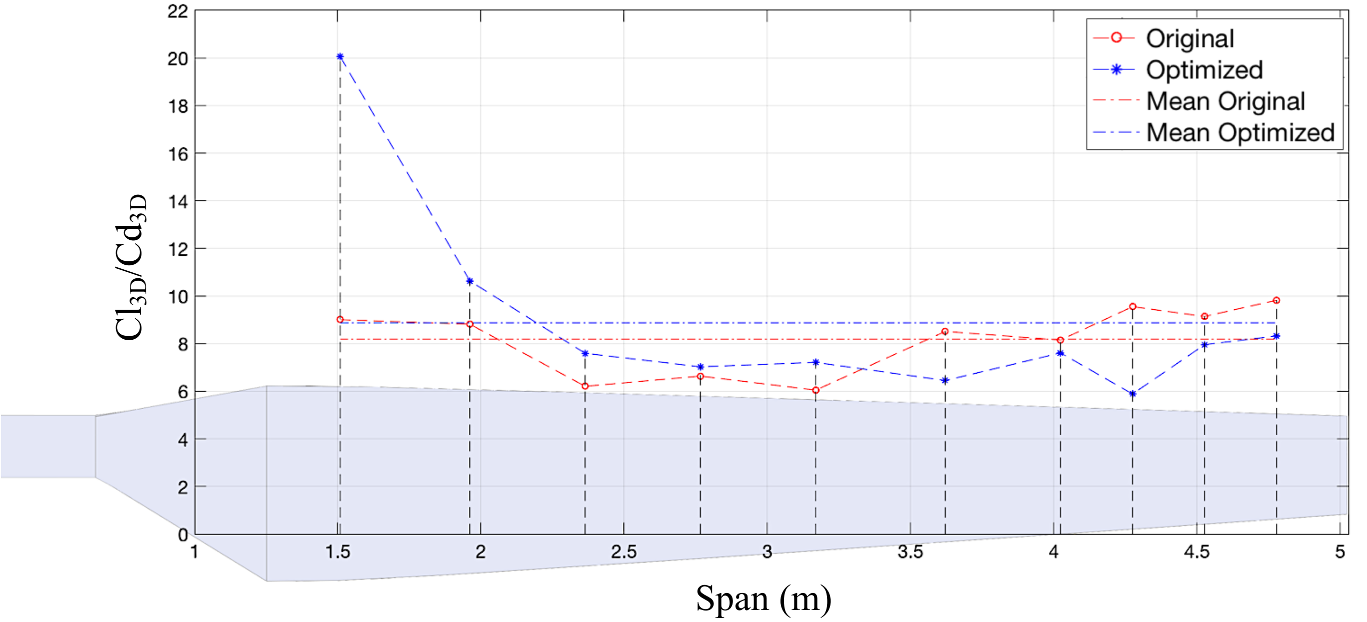
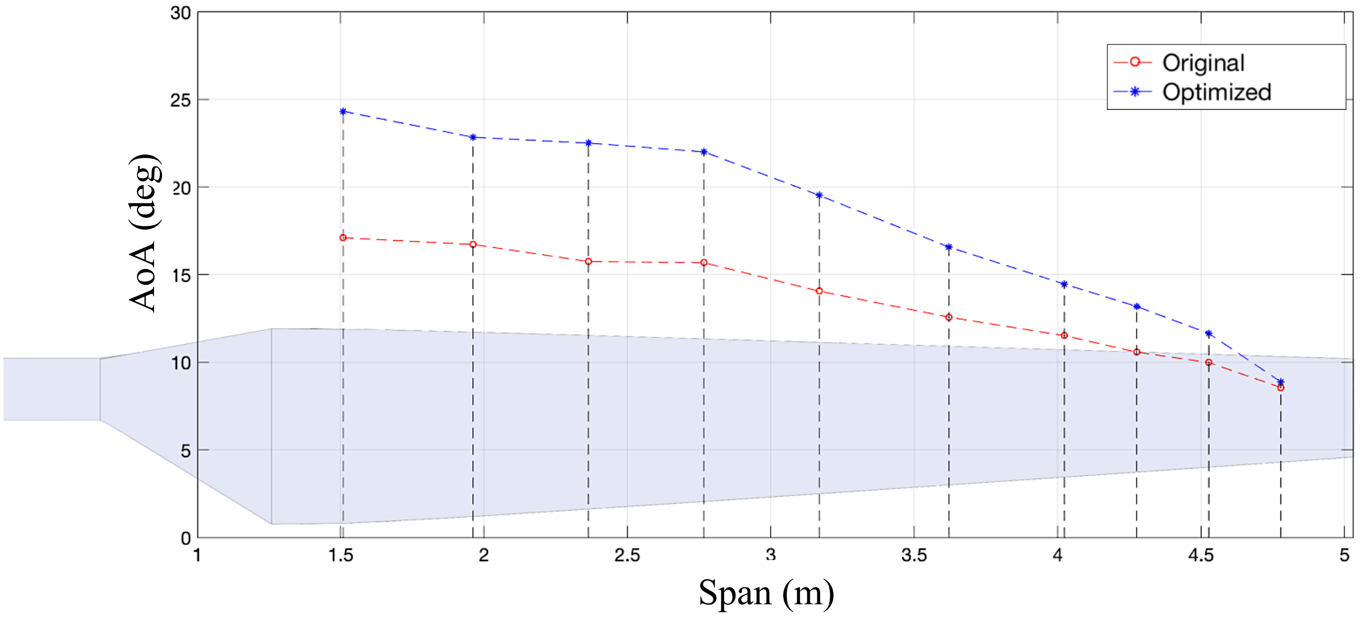
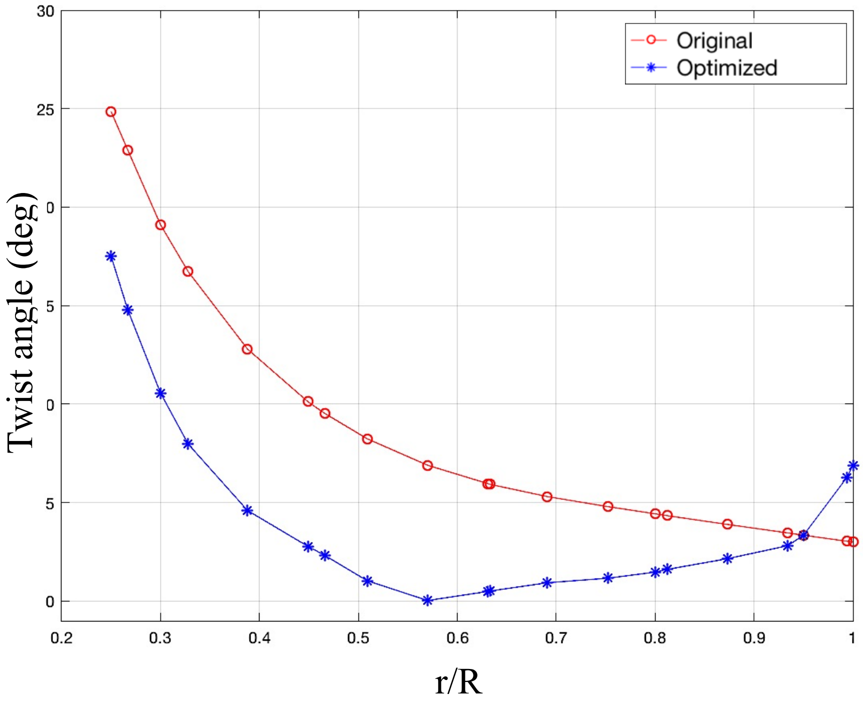


Figure 6.12: Cl_{3D}/Cd_{3D} ratio distribution for optimized blade at 10 m/s



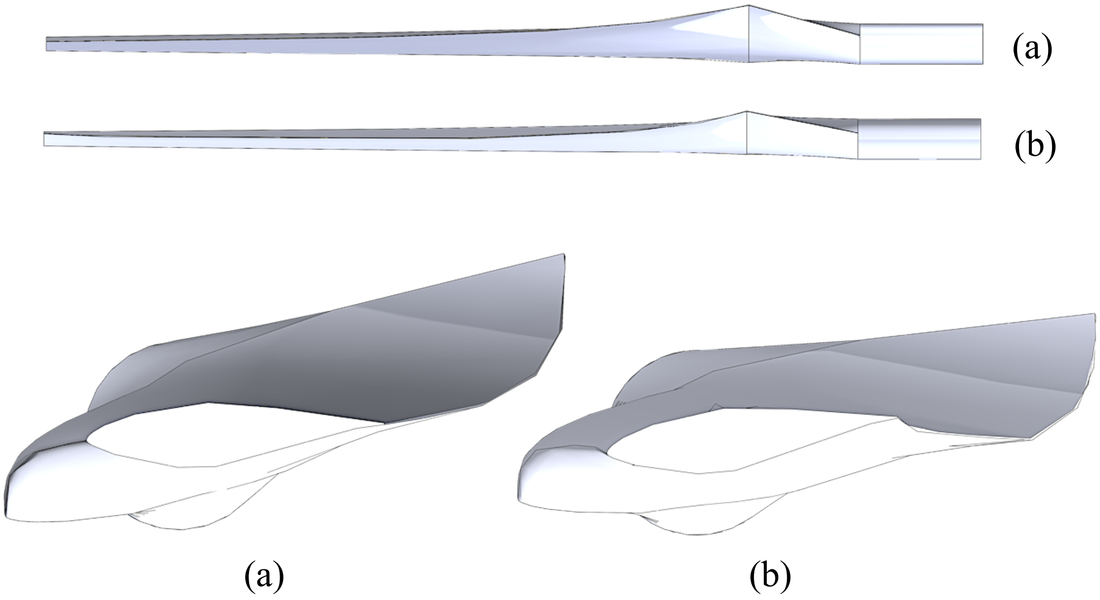
AoA for the optimized design has increased at every span station except the reference point, which is in accord with the proposed adjustment criteria. The difference between AoAs is presented in the Figure 6.12. The change of the AoA is more pronoun at the first half of the blade. After this, the difference between two versions diminishes. This is because the alteration of the twist angle in this region is smaller, as the maximum ratio location is presented here. This is evident is we look at the Figure 6.13, where the new and old twist angle distribution is juxtaposed. The red line shows the twist angle values along the span for the unchanged blade and the blue one depicts how angles are spread for the optimized blade.

Figure 6.13: Comparison of twist angle distribution



Using the optimized twist angles, Solidworks has built a 3D model of this wind turbine blade. The contrast between these two version is shown in the Figure 6.14. The visual difference between them is not very obvious. Trailing edge became more curved in the middle. This curvature is observed more clearly when looking at the top view, where the tip of the blade is inclined more towards the first section of the blade.

Figure 6.14: Side view of the (a) original and (b) optimized blade (top); top view of the (a) original and (b) optimized (bottom)



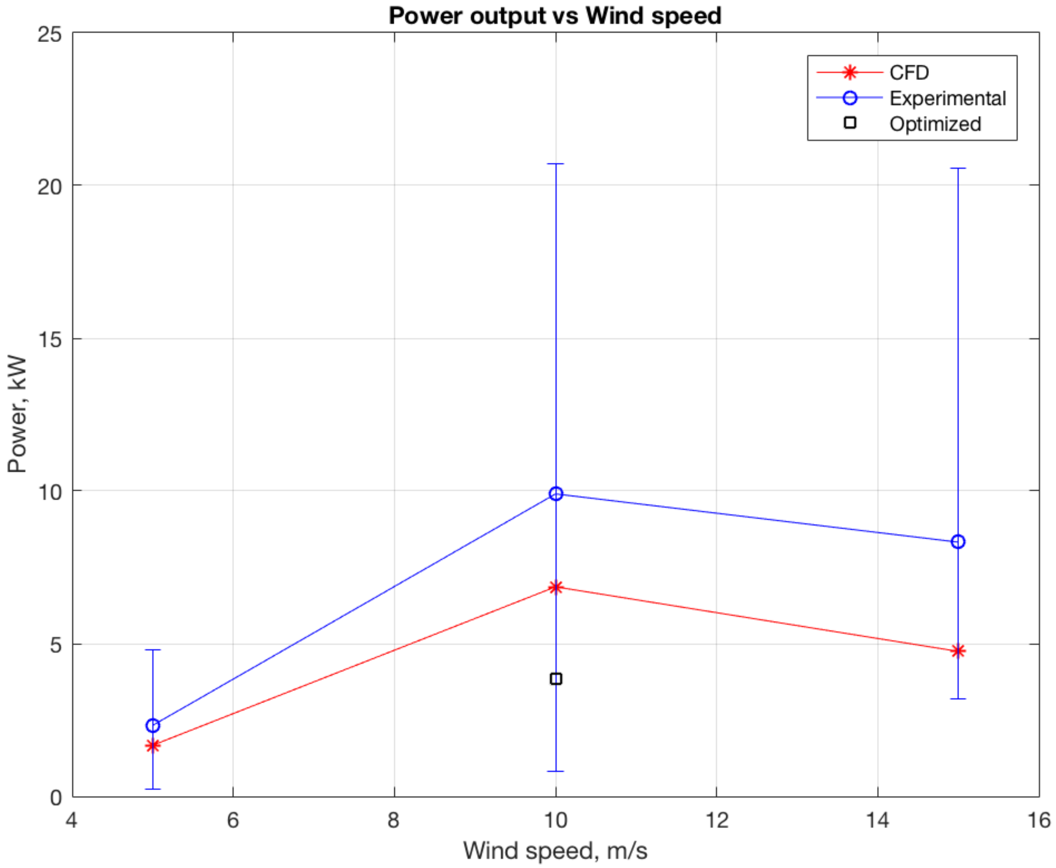
6.3 Discussion

The optimization process introduced in this work did improved the performance parameter, that is mean value of Cl_{3D} to Cd_{3D} ratio. This speaks in favor for the proposed optimization scheme. However, it only increased by small amount and at certain regions the ratio dropped. The possible reason behind this is that the amount by which twist angles are adjusted. Adjustment is calculated using values of ϕ from previous calculation, i.e. they are evaluated prior to the optimization. From the definition, ϕ is calculated from the system of non-linear equations. The input parameters of these equations depend on the simulation results. Since AoA is calculated from the ϕ, the only way to affect it is to change a local twist angle. Considering the complexity of calculating ϕ, this makes it difficult to correctly estimate the magnitude of the future AoA. Proposed scheme, that was aimed on increasing AoA, took an advantage of the stall delay effect by demonstrating increased lift coefficients in the first half of the blade, where AoA are higher. This leaves a room for improvement for the angle adjustment method.

Power output estimation of the optimized blade for the design condition is presented in the Figure 6.13. The new design produces less power compared to the unaltered blade. This indicates that optimization should be multi-objective and maximizing power output must be include as optimization goal for future work.

Meaning that it should not only aim for the higher ratio but also for higher power production.

Figure 6.15: Power estimation of the optimized blade at the design condition



6.4 Conclusion and Future Work

The automation process demonstrated in this work showed that it can used to solve optimization problems. This was demonstrated by implementing IBEM method to optimize the model to produce higher overall lift to drag ratio. The main benefit of using this approach is that it considers 3D aerodynamics effects such as stall delay. The resulting blade showed an increased in mean ratio, although the power estimation at the design condition was lower compared to the old version of the blade. This opens possibilities to improve presented method by introducing multi-objective optimization and different optimization schemes such as genetic algorithm, neural networks, ant colony or particle swarm optimization. It can also include both single or multiple objectives, for instance increase lift to drag ratio and also try to enhance power output.

Current framework is only at the preliminary stage and later can be improved by creating graphical user interface and make codes more flexible. Also parameterize meshing procedure in order to allow for automatic mesh refinement process. The usage of the presented framework can be extended to other CFD or even to structural mechanic problems, as ANSYS allow to model great amount of physical processes.

References

[1]"GLOBAL STATISTICS | GWEC", Gwec.net, 2017. [Online]. Available: http://gwec.net/global-figures/graphs/. [Accessed: 30- Nov- 2017].

[2] I. Syed Ahmed Kabir and E. Ng, "Insight into stall delay and computation of 3D sectional aerofoil characteristics of NREL phase VI wind turbine using inverse BEM and improvement in BEM analysis accounting for stall delay effect", Energy, vol. 120, pp. 518-536, 2017.

[3] J. Manwell, J. McGowan and A. Rogers, Wind energy explained. Chichester: John Wiley & Sons, 2011.

[4] S. Guntur and N. Sørensen, "An evaluation of several methods of determining the local angle of attack on wind turbine blades", Journal of Physics: Conference Series, vol. 555, p. 012045, 2014.

[5] H. Glauert, "Airplane Propellers", Aerodynamic Theory, pp. 169-360, 1935.

[6] M. Hansen, Aerodynamics of wind turbines. London [etc.]: Routledge, 2015.

[7] M. Hand, D. Simms, L. Fingersh, D. Jager, S. Larwood and J. Cotrell, "Unsteady aerodynamics experiment Phase VI: wind tunnel test configurations and available data campaigns", NREL/TP-500-29955. Golden, CO: NREL; 2001.

[8] L. Viterna and R. Corrigan, "Fixed pitch rotor performance of large horizontal axis wind turbines." (1982).

[9] R. Lanzafame and M. Messina, "Fluid dynamics wind turbine design: Critical analysis, optimization and application of BEM theory", Renewable Energy, vol. 32, no. 14, pp. 2291-2305, 2007.

[10] J. Tangler and D. Kocurek, "Wind turbine post-stall airfoil performance characteristics guidelines for blade-element momentum methods." 43rd AIAA Aerospace Sciences Meeting and Exhibit. AIAA, 2005.

[11] H. Snel, et al. "Sectional prediction of 3D effects for stalled flow on rotating blades and comparison with measurements." Proc. European Community Wind Energy Conference, HS Stevens and Associates, LÃ1. Vol. 4. 1993.

[12] Z. Du and M. Selig, "A 3-D stall-delay model for horizontal axis wind turbine performance prediction", 1998 ASME Wind Energy Symposium, 1998.

[13] P. Chaviaropoulos and M. Hansen, "Investigating Three-Dimensional and Rotational Effects on Wind Turbine Blades by Means of a Quasi-3D Navier-Stokes Solver", Journal of Fluids Engineering, vol. 122, no. 2, p. 330, 2000.

[14] C. Lindenburg, "Investigation into rotor blade aerodynamics." Netherlands Society for Energy and the Environment, Paper ECN-C-03-025 (2003).

[15] J. Sørensen, "The Tip Correction", Research Topics in Wind Energy, pp. 123-150, 2015.

[16] M. Elfarra, N. Sezer-Uzol and I. Akmandor, "NREL VI rotor blade: numerical investigation and winglet design and optimization using CFD", Wind Energy, vol. 17, no. 4, pp. 605-626, 2013.

[17] C. Pavese, C. Tibaldi, F. Zahle and T. Kim, "Aeroelastic multidisciplinary design optimization of a swept wind turbine blade", *Wind Energy*, vol. 20, no. 12, pp. 1941-1953, 2017.

[18] D. Kim, O. Lim, E. Choi and Y. Noh, "Optimization of 5-MW wind turbine blade using fluid structure interaction analysis", *Journal of Mechanical Science and Technology*, vol. 31, no. 2, pp. 725-732, 2017.

[19] M. Tahani, T. Maeda, N. Babayan, S. Mehrnia, M. Shadmehri, Q. Li, R. Fahimi and M. Masdari, "Investigating the effect of geometrical parameters of an optimized wind turbine blade in turbulent flow", *Energy Conversion and Management*, vol. 153, pp. 71-82, 2017.

[20] S. Jelena, T. Zorana, B. Marija and P. Ognjen, "Rapid multidisciplinary, multi-objective optimization of composite horizontal-axis wind turbine blade", *2016 International Conference Multidisciplinary Engineering Design Optimization (MEDO)*, 2016.

[21] J. Zhu, X. Cai and R. Gu, "Multi-Objective Aerodynamic and Structural Optimization of Horizontal-Axis Wind Turbine Blades", Energies, vol. 10, no. 1, p. 101, 2017.

[22] Y. Song and J. Perot, "CFD Simulation of the NREL Phase VI Rotor", Wind Engineering, vol. 39, no. 3, pp. 299-309, 2015.

[23] R. Mahu and F. Popescu, "NREL PHASE VI ROTOR MODELING AND SIMULATION USING ANSYS FLUENT 12.1.", Mathematical Modeling in Civil Engineering, no. 12, pp. 185-194, 2011.

[24] ANSYS FLUENT 18.1 Documentation

[25] "2016 SOLIDWORKS API Help - Welcome", Help.solidworks.com, 2017. [Online]. Available: http://help.solidworks.com/2016/english/api/sldworksapiprogguide/welcome.htm. [Accessed: 30- Nov- 2017].

[26] ANSYS, "Workbench Scripting Guide", 2013. [Online]. Available: http://itrss.mst.edu/media/informationtechnology/itrss/documents/ansysdocs/Workbench%20Scripting%20Guide.pdf [Accessed: 30- Nov- 2017].

[27] B. Reddy and R. Brioso, "Automated and Generic Finite Element Analysis for Industrial Robot Design", MSc, Linköping University, 2017.

Appendices

Appendix A

Global variables contained in the “twist.txt” file

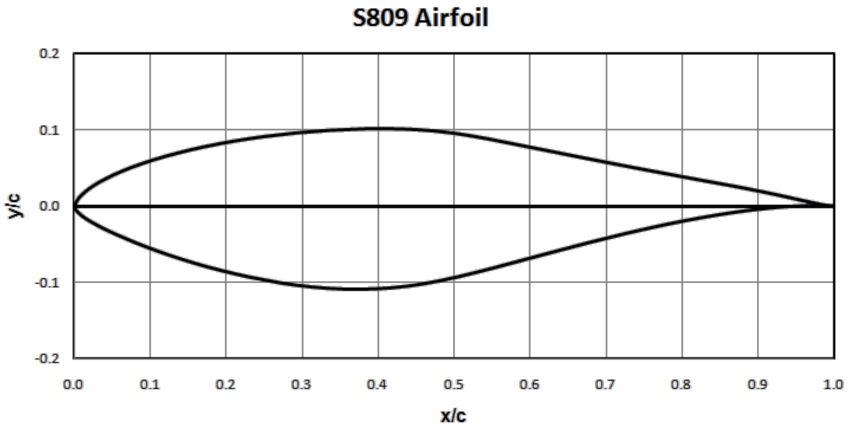
```
"tw1"=24.855
"tw2"=22.889
"tw3"=19.107
"tw4"=16.724
"tw5"=12.794
"tw6"=10.123
"tw7"=9.53
"tw8"=8.24
"tw9"=6.898
"tw10"=5.965
"tw11"=5.93
"tw12"=5.309
"tw13"=4.8
"tw14"=4.434
"tw15"=4.34
"tw16"=3.895
"tw17"=3.463
"tw18"=3.346
"tw19"=3.04
"tw20"=3
```

Appendix B

Table B1: Blade Chord and Twist distribution

Radial Distance r (m)	Span Station[1] (r/5.532 m)	Span Station[1] (r/5.029 m)	Chord Length (m)	Twist[2] (degrees)	Thickness (m)	Twist Axis (% chord)
0.0	0.0	0.0	Hub - center of rotation	Hub – center of rotation	Hub – center of rotation	Hub - center of rotation
0.508[3]	0.092	0.101	0.218 (root hub adapter)	0.0 (root hub adapter)	0.218	50 (root hub adapter)
0.660[4]	0.120	0.131	0.218	0.0	0.218	50
0.883[5]	0.160	0.176	0.183	0.0	0.183	50
1.008[5]	0.183	0.200	0.349	6.7	0.163	35.9
1.067[5]	0.193	0.212	0.441	9.9	0.154	33.5
1.133[5]	0.205	0.225	0.544	13.4	0.154	31.9
1.257[5]	0.227	0.250	0.737	20.040	0.154	30
1.343	0.243	0.267	0.728	18.074	20.95% chord	30
1.510	0.273	0.300	0.711	14.292	20.95% chord	30
1.648	0.298	0.328	0.697	11.909	20.95% chord	30
1.952	0.353	0.388	0.666	7.979	20.95% chord	30
2.257	0.408	0.449	0.636	5.308	20.95% chord	30
2.343	0.424	0.466	0.627	4.715	20.95% chord	30
2.562	0.463	0.509	0.605	3.425	20.95% chord	30
2.867	0.518	0.570	0.574	2.083	20.95% chord	30
3.172	0.573	0.631	0.543	1.150	20.95% chord	30
3.185	0.576	0.633	0.542	1.115	20.95% chord	30
3.476	0.628	0.691	0.512	0.494	20.95% chord	30
3.781	0.683	0.752	0.482	-0.015	20.95% chord	30
4.023	0.727	0.800	0.457	-0.381	20.95% chord	30
4.086	0.739	0.812	0.451	-0.475	20.95% chord	30
4.391	0.794	0.873	0.420	-0.920	20.95% chord	30
4.696	0.849	0.934	0.389	-1.352	20.95% chord	30
4.780	0.864	0.950	0.381	-1.469	20.95% chord	30
5.000	0.904	0.994	0.358	-1.775	20.95% chord	30
5.305	0.959	1.055	0.328	-2.191	20.95% chord	30
5.532	1.000	1.100	0.305	-2.500	20.95% chord	30

Table B2: NREL s809 airfoil coordinates



S809 Airfoil	
x/c	**y/c**
1	0
0.996203	0.000487
0.98519	0.002373
0.967844	0.00596
0.945073	0.011024
0.917488	0.017033
0.885293	0.023458
0.848455	0.03028
0.80747	0.037766
0.763042	0.045974
0.715952	0.054872
0.667064	0.064353
0.617331	0.074214
0.56783	0.084095
0.519832	0.093268
0.474243	0.099392
0.428461	0.10176
0.382612	0.10184
0.33726	0.10007
0.29297	0.096703
0.250247	0.091908

0.209576	0.085851
0.171409	0.078687
0.136174	0.07058
0.104263	0.061697
0.076035	0.052224
0.051823	0.042352
0.03191	0.032299
0.01659	0.02229
0.006026	0.012615
0.000658	0.003723
0.000204	0.001942
0	-0.00002
0.000213	-0.001794
0.001045	-0.003477
0.001208	-0.003724
0.002398	-0.005266
0.009313	-0.011499
0.02323	-0.020399
0.04232	-0.030269
0.065877	-0.040821
0.093426	-0.051923
0.124111	-0.063082
0.157653	-0.07373
0.193738	-0.083567
0.231914	-0.092442
0.271438	-0.099905
0.311968	-0.105281
0.35337	-0.108181
0.395329	-0.108011
0.438273	-0.104552
0.48192	-0.097347
0.527928	-0.086571
0.576211	-0.073979
0.626092	-0.060644
0.676744	-0.047441

0.727211	-0.0351
0.776432	-0.024204
0.823285	-0.015163
0.86663	-0.008204
0.905365	-0.003363
0.938474	-0.000487
0.965086	0.000743
0.984478	0.000775
0.996141	0.00029
1	0

Appendix C

Matlab Scripts

“objfunc0.m” – evaluates performance of the blade

```
function [mval,val,twn]=objfunc0(tw0)
    % This function evluates performance of the initial design

    sp=[0.25 0.267 0.3 0.328 0.388 0.449 0.466 0.509 0.57 0.631 0.633 0.691
0.752 0.8 0.812 0.873 0.934 0.95 0.994 1];
    spn=[0.3 0.39 0.47 0.55 0.63 0.72 0.8 0.85 0.9 0.95];



    twfunc2(tw0,sp); % Modifies twist.txt files in order to prepare SW model

    swApp = actxserver('SldWorks.Application'); % Launches Solidworks
application
    set(swApp, 'Visible', true);

    % Opens 3D model of the domain
    O1 =
invoke(swApp,'OpenDoc','C:\Users\User\Desktop\scripts\NRELPhaseVIpitch3degDom
ain.SLDPRT',1); % Opens previous version of the model
    SWeqn = invoke(O1, 'GetEquationMgr'); % creates an equation manager
objects
    % This for loop checks whether all equations are linked to the *.txt file
    ncount = invoke(SWeqn,'GetCount');
        for i=1:ncount
            EqnLnk = invoke(SWeqn,'LinkToFile');
            if EqnLnk==1
    %               path=get(SWeqn,'FilePath');
    %               disp(path)
```

```
                % Updates every global variable
                Up1 = invoke(SWeqn,'UpdateValuesFromExternalEquationFile');

            end
        end
    % Rebuilds the model accroding to the changes introduced by "twist.txt"
    Re2 = invoke(O1, 'EditRebuild3'); % Rebuilds the model
    invoke(O1, 'Save');               % Saves the model for further use

    %Re2 = invoke(O1,'ForceRebuild3','true');

    Ext = invoke(swApp, 'ExitApp');   % Exits the application

    % Sends command to start ANSYS and launch workbench script which performs
automation routine
    command = '"C:\Program Files\ANSYS
Inc\v181\Framework\bin\Win64\runwb2.exe" -R
"C:\Users\User\Desktop\scripts\ansys_process.wbjn" -I -X';
    status=system(command);

        if status==0
                % List of span stations that will be evaluated. Name of the
                % file represents at what percent of the total span this
                % cross section is located

                sec1='30.csv';
%%
                sec2='39.csv';
%Name of files containing exported pressure data from ANSYS
                sec3='47.csv';
%%
                sec4='55.csv';
                sec5='63.csv';
                sec6='72.csv';
                sec7='80.csv';
                sec8='85.csv';
                sec9='90.csv';
                sec10='95.csv';

                % Calculates force coefficients at the given span location

[Cnorm1,Ctang1,Ctq1,Cth1,Cp1,X1,ch1,R1,tw1]=datareadcsv(sec1);         %%

[Cnorm2,Ctang2,Ctq2,Cth2,Cp2,X2,ch2,R2,tw2]=datareadcsv(sec2);
%Extracting Thrust and Torque coefficients from data files

[Cnorm3,Ctang3,Ctq3,Cth3,Cp3,X3,ch3,R3,tw3]=datareadcsv(sec3);         %%

[Cnorm4,Ctang4,Ctq4,Cth4,Cp4,X4,ch4,R4,tw4]=datareadcsv(sec4);

[Cnorm5,Ctang5,Ctq5,Cth5,Cp5,X5,ch5,R5,tw5]=datareadcsv(sec5);

[Cnorm6,Ctang6,Ctq6,Cth6,Cp6,X6,ch6,R6,tw6]=datareadcsv(sec6);

[Cnorm7,Ctang7,Ctq7,Cth7,Cp7,X7,ch7,R7,tw7]=datareadcsv(sec7);
```

```
[Cnorm8,Ctang8,Ctq8,Cth8,Cp8,X8,ch8,R8,tw8]=datareadcsv(sec8);

[Cnorm9,Ctang9,Ctq9,Cth9,Cp9,X9,ch9,R9,tw9]=datareadcsv(sec9);

[Cnorm10,Ctang10,Ctq10,Cth10,Cp10,X10,ch10,R10,tw10]=datareadcsv(sec10);

                Cnorm=[Cnorm1, Cnorm2, Cnorm3, Cnorm4, Cnorm5, Cnorm6,
Cnorm7, Cnorm8, Cnorm9, Cnorm10];
                Ctang=[Ctang1 Ctang2 Ctang3 Ctang4 Ctang5 Ctang6 Ctang7
Ctang8 Ctang9 Ctang10];
                Cthr=[Cth1 Cth2 Cth3 Cth4 Cth5 Cth6 Cth7 Cth8 Cth9 Cth10];
                Ctor=[Ctq1 Ctq2 Ctq3 Ctq4 Ctq5 Ctq6 Ctq7 Ctq8 Ctq9 Ctq10];
                ch=[ch1 ch2 ch3 ch4 ch5 ch6 ch7 ch8 ch9 ch10];

                % Extracts the values of the local twist angles at the
                % given span stations
                twist=twgen(spn);
                R=[R1 R2 R3 R4 R5 R6 R7 R8 R9 R10];
                r=R./5.029;
                % Calculates lift, drag and angle of attack for all span
                % stations
                [Cl3d,Cd3d,AoA,phi]=invBEM(Cthr,Ctor,r,ch,twist);
        end

%     ratio = Cd3d./Cl3d;
%     Mratio = mean(ratio);
%     [minv,n] = min(ratio);
%     twspn = twfunc(h,n,twist);

    ratio = Cl3d./Cd3d;
    [val,loc]=max(ratio);
    mval=mean(ratio);
    disp(['original Cl,3D/Cd,3D  :' num2str(val)])

        for m=1:length(AoA)
        %     twn(m)=phi(i)-2*AoA(i)+AoA(loc);
            if AoA(m)>AoA(loc)
                da(m)=AoA(m)-AoA(loc);
                twn(m)=twist(m)-da(m);
            elseif AoA(m)<AoA(loc)
                da(m)=AoA(loc)-AoA(m);
                twn(m)=twist(m)+da(m);
            else
                twn(m)=twist(m);
            end
        end

    figure
    scatter(r,twn,'*')
    hold on
    scatter(r,twist,'o')

%     twn_sp=ntwist(twn,spn,sp);
%     figure
```

```
%       [twn_sp]=twtrans(twn,spn); % Updates twist.txt with the new values of
local twist angles

end
```

“invBEM.m” – code of the IBEM method

```
function [Cl3d,Cd3d,AoA,PHI]=invBEM(Cthr,Ctor,r,ch,twist)
    Nb=2;                                               % Number of wind
    er=1e-6;                                            % Error between
calculated values
    iter=1;

        W=72*pi/30;                                     % Rotational speed
        U=10;                                           % Inlet speed

%

%Declaring functions that will be used in invBEM method calculations

        f_phi=@(a,at,r) atan(((1-a)*U)/(W*(5.029*r)*(1+at)));
        f_F=@(phi,r) 2/pi*acos(exp(-Nb*(1-r)/(2*r*sin(phi))));
        f_an=@(F,phi,r,ch,Cthr)
1/(((8*pi*(5.029*r)*F*(sin(phi))^2)/(ch*Nb*Cthr))+1);
        f_atn=@(F,phi,r,ch,Ctor)
1/(((8*pi*(5.029*r)*F*(sin(phi)*cos(phi)))/(ch*Nb*Ctor))+1);

        n=length(Cthr);

% phi=f_phi(a,at,r)
% F=f_F(phi,r)
% an=f_an(F,phi,r)
% atn=f_atn(F,phi,r)

%     a = eps*ones(1,5);
%     at = eps*ones(1,5);
%     phi = zeros(1,5);

    a = zeros(1,n);
    at = zeros(1,n);

for i=1:n
%    disp(['  i   :' num2str(i)])
%    disp(['a1   :' num2str(a)])
%    disp(['at1   :' num2str(at)])
    while 1
%       disp(['iteration   :' num2str(iter)])
      iter=iter+1;
      phi(i)=f_phi(a(i),at(i),r(i));
      F(i)=f_F(phi(i),r(i));
      an(i)=f_an(F(i),phi(i),r(i),ch(i),Cthr(i));
      atn(i)=f_atn(F(i),phi(i),r(i),ch(i),Ctor(i));
%       disp(['anew   :' num2str(an(i))])
%       disp(['atnew   :' num2str(atn(i))])
```

```
          if abs(an(i)-a(i))<er && abs(atn(i)-at(i))<er
%               disp(['anew fin   :' num2str(an(i))])
%               disp(['atnew fin  :' num2str(atn(i))])
%               disp(['phi  :' num2str(phi(i))])
              iter=1;
              break

          end

      a(i)=an(i);
      at(i)=atn(i);
    end

    Cl3d(i)=Cthr(i)*cos(phi(i))+Ctor(i)*sin(phi(i));
    Cd3d(i)=Cthr(i)*sin(phi(i))-Ctor(i)*cos(phi(i));
    AoA(i)=radtodeg(phi(i))-(twist(i));
end


PHI=radtodeg(phi);

if min(Cd3d)<0
     Cd3d=Cd3d-2*(min(Cd3d));
end

plot(5.029*r,Cl3d,'b--')
        grid on
        xlabel('r (m)')
        ylabel('Cl,3D')
        subplot(4,1,2)
        plot(5.029*r,Cd3d,'b--')
        xlabel('r (m)')
        ylabel('Cd,3D')
        grid on
        subplot(4,1,3)
        plot(5.029*r,abs(Cl3d./Cd3d),'b--')
        hold on
        plot(5.029*r,mean(Cl3d./Cd3d)*ones(1,10),'r--')
        xlabel('r (m)')
        ylabel('Cl,3D/Cd,3D')

        grid on
        subplot(4,1,4)
        plot(5.029*r,AoA,'b--')
        xlabel('r (m)')
        ylabel('AoA')
        grid on

        figure
        scatter(r*5.029,a,'b','filled')
        xlabel('Span (m)')
        ylabel('Axial induction factor')
        grid on
```

```
        figure
        scatter(r*5.029,at,'b','filled')
        xlabel('Span (m)')
        ylabel('Angular induction factor')
        grid on

end
```

“datareadcsv.m” – converts ANSYS pressure data into force coefficients

```
function [Cnorm,Ctang,Ctq,Cth,Cp,X,chord,r,twist]=datareadcsv(secname)

% "datareadcsv" function handles pressure distribution readings across the
% curve length of the given span station. This readings are converted in
% the pressure coefficients forms and then various force coefficients are
% calculated.

M  = csvread(secname,5,0);                    % Reading data from ANSYS
TC = csvread('twistchord.csv',0,0);           % Reading twist and chord
distribution data
[~,name,~] = fileparts(secname);              % Stores the name of the CSV file
as the string type variable
span = str2double(name);                      % Converts string variable into
integer

% List of constants and variables that will be used in the calculation
% procedure
rho=1.225;                  % Air density
U=10;                       % Wind speed
omega=72*2*pi/60;           % Speed of rotation in rad/s
R=5.029;                    % Length of the blade
r=span*R/100;               % Location of the span station relative to the blade
length
prs = M(:,2);               % Pressure distribution at the given span station
sp=100*TC(:,1);             % Twist and chord values at the given span station

% Following cycle searches twist angle and the chord length at the given
% cross section according to the name of the section
for j=1:length(TC)
    if abs(sp(j)-span)<(1e-10)
        twist=TC(j,3);
%         disp(twist)
        chord=TC(j,2);
        break
    end
end

% This portion of code adjusts X coordinates of the original distribution
% and adjusts it according to the locl chord length
xx = M(:,1);
[~,locmax]=max(M(:,1));
[~,locmin]=min(M(:,1));
ch=xx/cos(deg2rad(twist));
```

```
chx=(ch-min(ch))/(max(ch)-min(ch));
% The following part estimates the direction in which pressure is
% distributed. This is necessary for forces coefficients calculations
diff=xx(2)-xx(1);

if diff>0

    prs_l=[prs(locmin:length(M))' prs(1:locmax)'];
    xx_l =[chx(locmin:length(M))' chx(1:locmax)'];

    prs_u=[prs(locmax:locmin)'];
    xx_u =[chx(locmax:locmin)'];

    X=[xx_u xx_l];
    % X=[fliplr(xx_u) fliplr(xx_l)];
    P=[prs_u prs_l];
    % P=[fliplr(prs_u) fliplr(prs_l)];

    Cp=P./(0.5*rho*(U^2+(omega*r).^2));
    [yyu,yyl]=yvalue(xx_u,xx_l);
    Y=[yyu yyl];
    elseif diff<0

        prs_l=[prs(locmax:length(M))' prs(1:locmin)'];
        xx_l =[chx(locmax:length(M))' chx(1:locmin)'];

        prs_u=[prs(locmin:locmax)'];
        xx_u =[chx(locmin:locmax)'];

        X=[fliplr(xx_u) fliplr(xx_l)];
%         X=[xx_u xx_l];
        P=[fliplr(prs_u) fliplr(prs_l)];
%         P=[prs_u prs_l];
        Cp=P./(0.5*rho*(U^2+(omega*r).^2));
        [yyu,yyl]=yvalue(xx_u,xx_l);
        Y=[fliplr(yyu) fliplr(yyl)];
%         Y=[yyu yyl];

end

% This cycle calculates normal and tangential force coefficients according
% to the method described in the NREL report
for i=1:length(M)
    n=i+1;
    if n==(length(M)+1)
        n=1;
        Cn(i)=abs(0.5*(Cp(i)+Cp(n))*(X(n)-X(i)));
        Ct(i)=abs(0.5*(Cp(i)+Cp(n))*(Y(n)-Y(i)));
    end

    Cn(i)=abs(0.5*(Cp(i)+Cp(n))*(X(n)-X(i)));
    Ct(i)=abs(0.5*(Cp(i)+Cp(n))*(Y(n)-Y(i)));
    Cnorm=sum(Cn);
    Ctang=sum(Ct);
end
```

```
% Calculation of thrust and torque forces coefficients
phi=deg2rad(twist);
Ctq=Cnorm*sin(phi)+Ctang*cos(phi);
Cth=Cnorm*cos(phi)-Ctang*sin(phi);

end
```

“twfunc2.m” – writes twist angles into “twist.txt”

```
function [twn_sp]=twfunc2(twn,spn)
% nv=length(twn);
%tw_n=15;
sp=[0.25 0.267 0.3 0.328 0.388 0.449 0.466 0.509 0.57 0.631 0.633 0.691 0.752
0.8 0.812 0.873 0.934 0.95 0.994 1];
% spn=[0.3 0.39 0.47 0.55 0.63 0.72 0.8 0.85 0.9 0.95];
fileID = fopen('C:\Users\User\Desktop\scripts\Results\twist.txt','r');
C=textscan(fileID,'%q %f','delimiter',sprintf('='));
fclose(fileID);
tw=C{2};
twn_sp=spline(spn,twn,sp);
% twn_sp_org=twn_sp;
fid=fopen('C:\Users\User\Desktop\scripts\Results\twist.txt','w');
ntw=1:20;
% hold on
% plot(sp,twn_sp,'s')
% grid on
for j=1:length(twn_sp)
    if twn_sp(j)<0
        twn_sp(j)=twn_sp(j)+(floor(abs(twn_sp(j))/360)+1)*360;
    elseif twn_sp(j)>360
        twn_sp(j)=twn_sp(j)-floor(twn_sp(j)/360)*360;
    end
end


% hold on
% plot(sp,twn_sp,'*')

% M=[ntw' twn_sp'];

for i=1:20
    fprintf(fid, '"tw%u"=%f\r\n',[ntw(i) twn_sp(i)]);
end

fclose(fid);

end
```

“twgen.m” – estimates local twist angles for the given span station

```
function twist=twgen(spn)
```

```
    sp=[0.25 0.267 0.3 0.328 0.388 0.449 0.466 0.509 0.57 0.631 0.633 0.691
0.752 0.8 0.812 0.873 0.934 0.95 0.994 1];
    fileID = fopen('C:\Users\User\Desktop\scripts\Results\twist.txt','r');
    C=textscan(fileID,'%q %f','delimiter',sprintf('='));
    fclose(fileID);
    tw=C{2};

    for i=1:length(tw)
        if (tw(i)-270)>0
            tw(i)=tw(i)-360;
        end
    end

    twist=spline(sp,tw,spn);
    disp(['Unaltered twist = ' num2str(twist)])


end
```

“structure.m” – main optimization script

```
%  This is the main script that launches all processes
% It manipulates Soliworks directly through invoking API commands
% ANSYS is operated by using command promt and workbench scripting
% Data extracted from ANSYS simulation is handeled by inverse BEM which
% outputs parameters on which optimization is based

clear
clc

% Firstly, the preocess is initiated by evaluating the standard design of
% the wind turbine blade. For that reason, orginal values of local twist
% angles are extracted from the linked property file
sp = [0.25 0.267 0.3 0.328 0.388 0.449 0.466 0.509 0.57 0.631 0.633 0.691
0.752 0.8 0.812 0.873 0.934 0.95 0.994 1];
fileID =
fopen('C:\Users\User\Desktop\scripts\Results\twist_original.txt','r');
C = textscan(fileID,'%q %f','delimiter',sprintf('='));
fclose(fileID);
tw0 = C{2}';
% The following line launches evaluation process based on intial parameters
% This function includes inverse BEM method as well as optmization routine
[mval0,val0,twn]=objfunc0(tw0);
RatioVal=[mval0 mval_n];
Twistangles=[twnsp;twists];

[~,lct]=max(RatioVal);
tw_best=Twistangles(lct,:);

[twn_sp]=twtrans(tw_best,sp)
while 1
```

```
    [mvaln,valn,twn_spn]=objfunc2;
    disp(['Mean CL/CD    :' num2str(mvaln)])
    disp(['Max  CL/CD    :' num2str(valn)])

        if  mvaln < mval
            break
        end
    iter=iter+1;
    disp(['iteration    :' num2str(iter)])
    mval=mvaln;


end
```

“objfunc2.m” – script that evaluates the performance of the modified designs

```
function [mval,val,twn]=objfunc2(tw0)

    sp=[0.25 0.267 0.3 0.328 0.388 0.449 0.466 0.509 0.57 0.631 0.633 0.691
0.752 0.8 0.812 0.873 0.934 0.95 0.994 1];
    spn=[0.3 0.39 0.47 0.55 0.63 0.72 0.8 0.85 0.9 0.95];

%     twfunc2(tw0,sp); % Modifies twist.txt files in order to prepare SW
model
     twfunc2(tw0,spn);


    swApp = actxserver('SldWorks.Application'); % Launches Solidworks
application
    set(swApp, 'Visible', true);


    O1 =
invoke(swApp,'OpenDoc','C:\Users\User\Desktop\scripts\NRELPhaseVIpitch3degDom
ain.SLDPRT',1); % Opens previous version of the model
    SWeqn = invoke(O1, 'GetEquationMgr'); % creates an equation manager
objects
    % This for loop checks whether all equations are linked to the *.txt file
    ncount = invoke(SWeqn,'GetCount');
        for i=1:ncount
            EqnLnk = invoke(SWeqn,'LinkToFile');
            if EqnLnk==1
    %             path=get(SWeqn,'FilePath');
    %             disp(path)
                Up1 = invoke(SWeqn,'UpdateValuesFromExternalEquationFile');

            end
        end

    Re2 = invoke(O1, 'EditRebuild3'); % Rebuilds the model
    invoke(O1, 'Save');               % Saves the model for further use
```

```
    %Re2 = invoke(O1,'ForceRebuild3','true');

    Ext = invoke(swApp, 'ExitApp');   % Exits the application

    command = '"C:\Program Files\ANSYS
Inc\v181\Framework\bin\Win64\runwb2.exe" -R
"C:\Users\User\Desktop\scripts\ansys_process.wbjn" -I -X';
    status=system(command);

        if status==0

                sec1='30.csv';
%%
                sec2='39.csv';
%Name of files containing exported pressure data from ANSYS
                sec3='47.csv';
%%
                sec4='55.csv';
                sec5='63.csv';
                sec6='72.csv';
                sec7='80.csv';
                sec8='85.csv';
                sec9='90.csv';
                sec10='95.csv';


[Cnorm1,Ctang1,Ctq1,Cth1,Cp1,X1,ch1,R1,tw1]=datareadcsv(sec1);         %%

[Cnorm2,Ctang2,Ctq2,Cth2,Cp2,X2,ch2,R2,tw2]=datareadcsv(sec2);
%Extracting Thrust and Torque coefficients from data files

[Cnorm3,Ctang3,Ctq3,Cth3,Cp3,X3,ch3,R3,tw3]=datareadcsv(sec3);         %%

[Cnorm4,Ctang4,Ctq4,Cth4,Cp4,X4,ch4,R4,tw4]=datareadcsv(sec4);

[Cnorm5,Ctang5,Ctq5,Cth5,Cp5,X5,ch5,R5,tw5]=datareadcsv(sec5);

[Cnorm6,Ctang6,Ctq6,Cth6,Cp6,X6,ch6,R6,tw6]=datareadcsv(sec6);

[Cnorm7,Ctang7,Ctq7,Cth7,Cp7,X7,ch7,R7,tw7]=datareadcsv(sec7);

[Cnorm8,Ctang8,Ctq8,Cth8,Cp8,X8,ch8,R8,tw8]=datareadcsv(sec8);

[Cnorm9,Ctang9,Ctq9,Cth9,Cp9,X9,ch9,R9,tw9]=datareadcsv(sec9);

[Cnorm10,Ctang10,Ctq10,Cth10,Cp10,X10,ch10,R10,tw10]=datareadcsv(sec10);

                Cnorm=[Cnorm1, Cnorm2, Cnorm3, Cnorm4, Cnorm5, Cnorm6,
Cnorm7, Cnorm8, Cnorm9, Cnorm10];
                Ctang=[Ctang1 Ctang2 Ctang3 Ctang4 Ctang5 Ctang6 Ctang7
Ctang8 Ctang9 Ctang10];
                Cthr=[Cth1 Cth2 Cth3 Cth4 Cth5 Cth6 Cth7 Cth8 Cth9 Cth10];
                Ctor=[Ctq1 Ctq2 Ctq3 Ctq4 Ctq5 Ctq6 Ctq7 Ctq8 Ctq9 Ctq10];
                ch=[ch1 ch2 ch3 ch4 ch5 ch6 ch7 ch8 ch9 ch10];
```

```
                twist=twgen(spn);
                 disp(['old twist  :' num2str(twist)])
                R=[R1 R2 R3 R4 R5 R6 R7 R8 R9 R10];
                r=R./5.029;

                [Cl3d,Cd3d,AoA,phi]=invBEM(Cthr,Ctor,r,ch,twist);
        end



for i=1:length(twist)
        if (twist(i)-270)>0
            twist(i)=twist(i)-360;
        end
    end


    ratio = Cl3d./Cd3d;
    [val,loc]=max(ratio);
    mval=mean(ratio);
    disp(['mean Cl,3D/Cd,3D  :' num2str(mval)])
    disp(['Angle of Attack  :' num2str(AoA)])
        for m=1:length(AoA)
        %     twn(m)=phi(i)-2*AoA(i)+AoA(loc);
            if AoA(m)>AoA(loc)
                da(m)=AoA(m)-AoA(loc);
                twn(m)=twist(m)-da(m);
            elseif AoA(m)<AoA(loc)
                da(m)=AoA(loc)-AoA(m);
                twn(m)=twist(m)+da(m);
            else
                twn(m)=twist(m);
            end


        end

    disp(['New twist  :' num2str(twn)])
    figure
    scatter(r,twn,'*')
    hold on
    scatter(r,twist,'o')

end
```

Appendix D

Main python script “ansys_process.wbjn”

```
# encoding: utf-8
# Release 18.1
SetScriptVersion(Version="18.1.463")
template1 = GetTemplate(TemplateName="Geometry")
system1 = template1.CreateSystem()
geometry1 = system1.GetContainer(ComponentName="Geometry")
geometry1.SetFile(FilePath="C:/Users/User/Desktop/scripts/NRELPhaseVIpitch3degDomain.SLDPRT")
geometryProperties1 = geometry1.GetGeometryProperties()
geometryProperties1.GeometryImportNamedSelections = True
geometryProperties1.GeometryImportNamedSelectionsFilter = "NS"
template2 = GetTemplate(TemplateName="Fluid Flow")
system2 = template2.CreateSystem(
   Position="Right",
   RelativeTo=system1)
geometryComponent1 = system1.GetComponent(Name="Geometry")
meshComponent1 = system2.GetComponent(Name="Mesh")
meshContent1 = system2.GetContainer(ComponentName="Mesh")
geometryComponent1.TransferData(TargetComponent=meshComponent1)

meshContent1.Edit()

meshContent1.SendCommand(Command="WB.AppletList.Applet(\"DSApplet\").App.Script.doToolsRu
nMacro(\"C:/Users/User/Desktop/scripts/meshgen.js\")")
meshContent1.Exit()

meshComponent1.Update(AllDependencies=True)

Save(
   FilePath="C:/Users/User/Desktop/testing zone/testrun.wbpj",
   Overwrite=True)


system3 = GetSystem(Name="FFF")
setup1 = system3.GetContainer(ComponentName="Setup")
fluentLauncherSettings1 = setup1.GetFluentLauncherSettings()
fluentLauncherSettings1.Precision = "Double"
fluentLauncherSettings1.RunParallel = True
fluentLauncherSettings1.NumberOfProcessors = 12
setupComponent1 = system3.GetComponent(Name="Setup")
setupComponent1.Refresh()
fluentLauncherSettings1.SetEntityProperties(Properties=Set(EnvPath={}))
setup1.Edit()
setup1.SendCommand(Command='(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1"
(list ))')
setup1.SendCommand(Command="/file/read-journal \"C:\\Users\\User\\Desktop\\scripts\setup specs.jou\"
" )
```

```
Save(
    FilePath="C:/Users/User/Desktop/testing zone/testrun.wbpj",
    Overwrite=True)

system4 = GetSystem(Name="FFF")
results1 = system4.GetContainer(ComponentName="Results")
results1.Edit()


# Creates plane at 30% span location
results1.SendCommand(Command="> autolegend plot=/PLANE:span30, view=VIEW:View 1")
results1.SendCommand(Command="""PLANE:span30
Apply Instancing Transform = On
Apply Texture = Off
Blend Texture = On
Bound Radius = 0.5 [m]
Colour = 0.75, 0.75, 0.75
Colour Map = Default Colour Map
Colour Mode = Constant
Colour Scale = Linear
Colour Variable = Pressure
Colour Variable Boundary Values = Conservative
Culling Mode = No Culling
Direction 1 Bound = 1.0 [m]
Direction 1 Orientation = 0 [degree]
Direction 1 Points = 10
Direction 2 Bound = 1.0 [m]
Direction 2 Points = 10
Domain List = /DOMAIN GROUP:All Domains
Draw Faces = On
Draw Lines = Off
Instancing Transform = /DEFAULT INSTANCE TRANSFORM:Default Transform
Invert Plane Bound = Off
Lighting = On
Line Colour = 0, 0, 0
Line Colour Mode = Default
Line Width = 1
Max = 0.0
Min = 0.0
Normal = 1 , 0 , 0
Option = ZX Plane
Plane Bound = None
Plane Type = Slice
Point = 0 [m], 0 [m], 0 [m]
Point 1 = 0 [m], 0 [m], 0 [m]
Point 2 = 1 [m], 0 [m], 0 [m]
Point 3 = 0 [m], 1 [m], 0 [m]
Range = Global
Render Edge Angle = 0 [degree]
Specular Lighting = On
```

```
Surface Drawing = Smooth Shading
Texture Angle = 0
Texture Direction = 0 , 1 , 0
Texture File =
Texture Material = Metal
Texture Position = 0 , 0
Texture Scale = 1
Texture Type = Predefined
Tile Texture = Off
Transform Texture = Off
Transparency = 0.0
X = 0.0 [m]
Y = 0.3*5.029
Z = 0.0 [m]
  OBJECT VIEW TRANSFORM:
  Apply Reflection = Off
  Apply Rotation = Off
  Apply Scale = Off
  Apply Translation = Off
  Principal Axis = Z
  Reflection Plane Option = XY Plane
  Rotation Angle = 0.0 [degree]
  Rotation Axis From = 0 [m], 0 [m], 0 [m]
  Rotation Axis To = 0 [m], 0 [m], 0 [m]
  Rotation Axis Type = Principal Axis
  Scale Vector = 1 , 1 , 1
  Translation Vector = 0 [m], 0 [m], 0 [m]
  X = 0.0 [m]
  Y = 0.0 [m]
  Z = 0.0 [m]
  END
END""")
#
#
results1.SendCommand(Command="""# Sending visibility action from ViewUtilities
>show /PLANE:span30, view=/VIEW:View 1""")
results1.SendCommand(Command="""# Sending visibility action from ViewUtilities
>hide /PLANE:span1, view=/VIEW:View 1""")
results1.SendCommand(Command="> autolegend plot=/POLYLINE:pl30, view=VIEW:View 1")
results1.SendCommand(Command="""POLYLINE:pl30
Apply Instancing Transform = On
Boundary List = ns_blade
Colour = 0, 1, 0
Colour Map = Default Colour Map
Colour Mode = Constant
Colour Scale = Linear
Colour Variable = Pressure
Colour Variable Boundary Values = Conservative
Contour Level = 1
Domain List = /DOMAIN GROUP:All Domains
Input File =
```

```
Instancing Transform = /DEFAULT INSTANCE TRANSFORM:Default Transform
Line Width = 2
Location = /PLANE:span30
Max = 0.0
Min = 0.0
Option = Boundary Intersection
Range = Global
  OBJECT VIEW TRANSFORM:
  Apply Reflection = Off
  Apply Rotation = Off
  Apply Scale = Off
  Apply Translation = Off
  Principal Axis = Z
  Reflection Plane Option = XY Plane
  Rotation Angle = 0.0 [degree]
  Rotation Axis From = 0 [m], 0 [m], 0 [m]
  Rotation Axis To = 0 [m], 0 [m], 0 [m]
  Rotation Axis Type = Principal Axis
  Scale Vector = 1 , 1 , 1
  Translation Vector = 0 [m], 0 [m], 0 [m]
  X = 0.0 [m]
  Y = 0.0 [m]
  Z = 0.0 [m]
  END
END""")
results1.SendCommand(Command="""# Sending visibility action from ViewUtilities
>show /POLYLINE:pl30, view=/VIEW:View 1""")
#

# Creates plane at 39% span location
results1.SendCommand(Command="> autolegend plot=/PLANE:span39, view=VIEW:View 1")
results1.SendCommand(Command="""PLANE:span39
Apply Instancing Transform = On
Apply Texture = Off
Blend Texture = On
Bound Radius = 0.5 [m]
Colour = 0.75, 0.75, 0.75
Colour Map = Default Colour Map
Colour Mode = Constant
Colour Scale = Linear
Colour Variable = Pressure
Colour Variable Boundary Values = Conservative
Culling Mode = No Culling
Direction 1 Bound = 1.0 [m]
Direction 1 Orientation = 0 [degree]
Direction 1 Points = 10
Direction 2 Bound = 1.0 [m]
Direction 2 Points = 10
Domain List = /DOMAIN GROUP:All Domains
Draw Faces = On
Draw Lines = Off
```

```
Instancing Transform = /DEFAULT INSTANCE TRANSFORM:Default Transform
Invert Plane Bound = Off
Lighting = On
Line Colour = 0, 0, 0
Line Colour Mode = Default
Line Width = 1
Max = 0.0
Min = 0.0
Normal = 1 , 0 , 0
Option = ZX Plane
Plane Bound = None
Plane Type = Slice
Point = 0 [m], 0 [m], 0 [m]
Point 1 = 0 [m], 0 [m], 0 [m]
Point 2 = 1 [m], 0 [m], 0 [m]
Point 3 = 0 [m], 1 [m], 0 [m]
Range = Global
Render Edge Angle = 0 [degree]
Specular Lighting = On
Surface Drawing = Smooth Shading
Texture Angle = 0
Texture Direction = 0 , 1 , 0
Texture File =
Texture Material = Metal
Texture Position = 0 , 0
Texture Scale = 1
Texture Type = Predefined
Tile Texture = Off
Transform Texture = Off
Transparency = 0.0
X = 0.0 [m]
Y = 0.39*5.029
Z = 0.0 [m]
  OBJECT VIEW TRANSFORM:
  Apply Reflection = Off
  Apply Rotation = Off
  Apply Scale = Off
  Apply Translation = Off
  Principal Axis = Z
  Reflection Plane Option = XY Plane
  Rotation Angle = 0.0 [degree]
  Rotation Axis From = 0 [m], 0 [m], 0 [m]
  Rotation Axis To = 0 [m], 0 [m], 0 [m]
  Rotation Axis Type = Principal Axis
  Scale Vector = 1 , 1 , 1
  Translation Vector = 0 [m], 0 [m], 0 [m]
  X = 0.0 [m]
  Y = 0.0 [m]
  Z = 0.0 [m]
  END
END""")
```

```
results1.SendCommand(Command="""# Sending visibility action from ViewUtilities
>show /PLANE:span39, view=/VIEW:View 1""")
#
#
results1.SendCommand(Command="> autolegend plot=/POLYLINE:pl39, view=VIEW:View 1")
results1.SendCommand(Command="""POLYLINE:pl39
Apply Instancing Transform = On
Boundary List = ns_blade
Colour = 0, 1, 0
Colour Map = Default Colour Map
Colour Mode = Constant
Colour Scale = Linear
Colour Variable = Pressure
Colour Variable Boundary Values = Conservative
Contour Level = 1
Domain List = /DOMAIN GROUP:All Domains
Input File =
Instancing Transform = /DEFAULT INSTANCE TRANSFORM:Default Transform
Line Width = 2
Location = /PLANE:span39
Max = 0.0
Min = 0.0
Option = Boundary Intersection
Range = Global
  OBJECT VIEW TRANSFORM:
  Apply Reflection = Off
  Apply Rotation = Off
  Apply Scale = Off
  Apply Translation = Off
  Principal Axis = Z
  Reflection Plane Option = XY Plane
  Rotation Angle = 0.0 [degree]
  Rotation Axis From = 0 [m], 0 [m], 0 [m]
  Rotation Axis To = 0 [m], 0 [m], 0 [m]
  Rotation Axis Type = Principal Axis
  Scale Vector = 1 , 1 , 1
  Translation Vector = 0 [m], 0 [m], 0 [m]
  X = 0.0 [m]
  Y = 0.0 [m]
  Z = 0.0 [m]
  END
END""")
results1.SendCommand(Command="""# Sending visibility action from ViewUtilities
>show /POLYLINE:pl39, view=/VIEW:View 1""")
#

# Creates plane at 47% span location
results1.SendCommand(Command="> autolegend plot=/PLANE:span47, view=VIEW:View 1")
results1.SendCommand(Command="""PLANE:span47
Apply Instancing Transform = On
Apply Texture = Off
```

```
Blend Texture = On
Bound Radius = 0.5 [m]
Colour = 0.75, 0.75, 0.75
Colour Map = Default Colour Map
Colour Mode = Constant
Colour Scale = Linear
Colour Variable = Pressure
Colour Variable Boundary Values = Conservative
Culling Mode = No Culling
Direction 1 Bound = 1.0 [m]
Direction 1 Orientation = 0 [degree]
Direction 1 Points = 10
Direction 2 Bound = 1.0 [m]
Direction 2 Points = 10
Domain List = /DOMAIN GROUP:All Domains
Draw Faces = On
Draw Lines = Off
Instancing Transform = /DEFAULT INSTANCE TRANSFORM:Default Transform
Invert Plane Bound = Off
Lighting = On
Line Colour = 0, 0, 0
Line Colour Mode = Default
Line Width = 1
Max = 0.0
Min = 0.0
Normal = 1 , 0 , 0
Option = ZX Plane
Plane Bound = None
Plane Type = Slice
Point = 0 [m], 0 [m], 0 [m]
Point 1 = 0 [m], 0 [m], 0 [m]
Point 2 = 1 [m], 0 [m], 0 [m]
Point 3 = 0 [m], 1 [m], 0 [m]
Range = Global
Render Edge Angle = 0 [degree]
Specular Lighting = On
Surface Drawing = Smooth Shading
Texture Angle = 0
Texture Direction = 0 , 1 , 0
Texture File =
Texture Material = Metal
Texture Position = 0 , 0
Texture Scale = 1
Texture Type = Predefined
Tile Texture = Off
Transform Texture = Off
Transparency = 0.0
X = 0.0 [m]
Y = 0.47*5.029
Z = 0.0 [m]
  OBJECT VIEW TRANSFORM:
```

```
  Apply Reflection = Off
  Apply Rotation = Off
  Apply Scale = Off
  Apply Translation = Off
  Principal Axis = Z
  Reflection Plane Option = XY Plane
  Rotation Angle = 0.0 [degree]
  Rotation Axis From = 0 [m], 0 [m], 0 [m]
  Rotation Axis To = 0 [m], 0 [m], 0 [m]
  Rotation Axis Type = Principal Axis
  Scale Vector = 1 , 1 , 1
  Translation Vector = 0 [m], 0 [m], 0 [m]
  X = 0.0 [m]
  Y = 0.0 [m]
  Z = 0.0 [m]
  END
END""")
results1.SendCommand(Command="""# Sending visibility action from ViewUtilities
>show /PLANE:span47, view=/VIEW:View 1""")
#
#
results1.SendCommand(Command="> autolegend plot=/POLYLINE:pl47, view=VIEW:View 1")
results1.SendCommand(Command="""POLYLINE:pl47
Apply Instancing Transform = On
Boundary List = ns_blade
Colour = 0, 1, 0
Colour Map = Default Colour Map
Colour Mode = Constant
Colour Scale = Linear
Colour Variable = Pressure
Colour Variable Boundary Values = Conservative
Contour Level = 1
Domain List = /DOMAIN GROUP:All Domains
Input File =
Instancing Transform = /DEFAULT INSTANCE TRANSFORM:Default Transform
Line Width = 2
Location = /PLANE:span47
Max = 0.0
Min = 0.0
Option = Boundary Intersection
Range = Global
  OBJECT VIEW TRANSFORM:
  Apply Reflection = Off
  Apply Rotation = Off
  Apply Scale = Off
  Apply Translation = Off
  Principal Axis = Z
  Reflection Plane Option = XY Plane
  Rotation Angle = 0.0 [degree]
  Rotation Axis From = 0 [m], 0 [m], 0 [m]
  Rotation Axis To = 0 [m], 0 [m], 0 [m]
```

```
  Rotation Axis Type = Principal Axis
  Scale Vector = 1 , 1 , 1
  Translation Vector = 0 [m], 0 [m], 0 [m]
  X = 0.0 [m]
  Y = 0.0 [m]
  Z = 0.0 [m]
  END
END""")
results1.SendCommand(Command="""# Sending visibility action from ViewUtilities
>show /POLYLINE:pl47, view=/VIEW:View 1""")
#

# Creates plane at 55% span location
results1.SendCommand(Command="> autolegend plot=/PLANE:span55, view=VIEW:View 1")
results1.SendCommand(Command="""PLANE:span55
Apply Instancing Transform = On
Apply Texture = Off
Blend Texture = On
Bound Radius = 0.5 [m]
Colour = 0.75, 0.75, 0.75
Colour Map = Default Colour Map
Colour Mode = Constant
Colour Scale = Linear
Colour Variable = Pressure
Colour Variable Boundary Values = Conservative
Culling Mode = No Culling
Direction 1 Bound = 1.0 [m]
Direction 1 Orientation = 0 [degree]
Direction 1 Points = 10
Direction 2 Bound = 1.0 [m]
Direction 2 Points = 10
Domain List = /DOMAIN GROUP:All Domains
Draw Faces = On
Draw Lines = Off
Instancing Transform = /DEFAULT INSTANCE TRANSFORM:Default Transform
Invert Plane Bound = Off
Lighting = On
Line Colour = 0, 0, 0
Line Colour Mode = Default
Line Width = 1
Max = 0.0
Min = 0.0
Normal = 1 , 0 , 0
Option = ZX Plane
Plane Bound = None
Plane Type = Slice
Point = 0 [m], 0 [m], 0 [m]
Point 1 = 0 [m], 0 [m], 0 [m]
Point 2 = 1 [m], 0 [m], 0 [m]
Point 3 = 0 [m], 1 [m], 0 [m]
Range = Global
```

```
Render Edge Angle = 0 [degree]
Specular Lighting = On
Surface Drawing = Smooth Shading
Texture Angle = 0
Texture Direction = 0 , 1 , 0
Texture File =
Texture Material = Metal
Texture Position = 0 , 0
Texture Scale = 1
Texture Type = Predefined
Tile Texture = Off
Transform Texture = Off
Transparency = 0.0
X = 0.0 [m]
Y = 0.55*5.029
Z = 0.0 [m]
  OBJECT VIEW TRANSFORM:
  Apply Reflection = Off
  Apply Rotation = Off
  Apply Scale = Off
  Apply Translation = Off
  Principal Axis = Z
  Reflection Plane Option = XY Plane
  Rotation Angle = 0.0 [degree]
  Rotation Axis From = 0 [m], 0 [m], 0 [m]
  Rotation Axis To = 0 [m], 0 [m], 0 [m]
  Rotation Axis Type = Principal Axis
  Scale Vector = 1 , 1 , 1
  Translation Vector = 0 [m], 0 [m], 0 [m]
  X = 0.0 [m]
  Y = 0.0 [m]
  Z = 0.0 [m]
  END
END""")
results1.SendCommand(Command="""# Sending visibility action from ViewUtilities
>show /PLANE:span55, view=/VIEW:View 1""")
#
#
results1.SendCommand(Command="> autolegend plot=/POLYLINE:pl55, view=VIEW:View 1")
results1.SendCommand(Command="""POLYLINE:pl55
Apply Instancing Transform = On
Boundary List = ns_blade
Colour = 0, 1, 0
Colour Map = Default Colour Map
Colour Mode = Constant
Colour Scale = Linear
Colour Variable = Pressure
Colour Variable Boundary Values = Conservative
Contour Level = 1
Domain List = /DOMAIN GROUP:All Domains
Input File =
```

```
Instancing Transform = /DEFAULT INSTANCE TRANSFORM:Default Transform
Line Width = 2
Location = /PLANE:span55
Max = 0.0
Min = 0.0
Option = Boundary Intersection
Range = Global
  OBJECT VIEW TRANSFORM:
  Apply Reflection = Off
  Apply Rotation = Off
  Apply Scale = Off
  Apply Translation = Off
  Principal Axis = Z
  Reflection Plane Option = XY Plane
  Rotation Angle = 0.0 [degree]
  Rotation Axis From = 0 [m], 0 [m], 0 [m]
  Rotation Axis To = 0 [m], 0 [m], 0 [m]
  Rotation Axis Type = Principal Axis
  Scale Vector = 1 , 1 , 1
  Translation Vector = 0 [m], 0 [m], 0 [m]
  X = 0.0 [m]
  Y = 0.0 [m]
  Z = 0.0 [m]
  END
END""")
results1.SendCommand(Command="""# Sending visibility action from ViewUtilities
>show /POLYLINE:pl55, view=/VIEW:View 1""")
#

# Creates plane at 63% span location
results1.SendCommand(Command="> autolegend plot=/PLANE:span63, view=VIEW:View 1")
results1.SendCommand(Command="""PLANE:span63
Apply Instancing Transform = On
Apply Texture = Off
Blend Texture = On
Bound Radius = 0.5 [m]
Colour = 0.75, 0.75, 0.75
Colour Map = Default Colour Map
Colour Mode = Constant
Colour Scale = Linear
Colour Variable = Pressure
Colour Variable Boundary Values = Conservative
Culling Mode = No Culling
Direction 1 Bound = 1.0 [m]
Direction 1 Orientation = 0 [degree]
Direction 1 Points = 10
Direction 2 Bound = 1.0 [m]
Direction 2 Points = 10
Domain List = /DOMAIN GROUP:All Domains
Draw Faces = On
Draw Lines = Off
```

```
Instancing Transform = /DEFAULT INSTANCE TRANSFORM:Default Transform
Invert Plane Bound = Off
Lighting = On
Line Colour = 0, 0, 0
Line Colour Mode = Default
Line Width = 1
Max = 0.0
Min = 0.0
Normal = 1 , 0 , 0
Option = ZX Plane
Plane Bound = None
Plane Type = Slice
Point = 0 [m], 0 [m], 0 [m]
Point 1 = 0 [m], 0 [m], 0 [m]
Point 2 = 1 [m], 0 [m], 0 [m]
Point 3 = 0 [m], 1 [m], 0 [m]
Range = Global
Render Edge Angle = 0 [degree]
Specular Lighting = On
Surface Drawing = Smooth Shading
Texture Angle = 0
Texture Direction = 0 , 1 , 0
Texture File =
Texture Material = Metal
Texture Position = 0 , 0
Texture Scale = 1
Texture Type = Predefined
Tile Texture = Off
Transform Texture = Off
Transparency = 0.0
X = 0.0 [m]
Y = 0.63*5.029
Z = 0.0 [m]
  OBJECT VIEW TRANSFORM:
  Apply Reflection = Off
  Apply Rotation = Off
  Apply Scale = Off
  Apply Translation = Off
  Principal Axis = Z
  Reflection Plane Option = XY Plane
  Rotation Angle = 0.0 [degree]
  Rotation Axis From = 0 [m], 0 [m], 0 [m]
  Rotation Axis To = 0 [m], 0 [m], 0 [m]
  Rotation Axis Type = Principal Axis
  Scale Vector = 1 , 1 , 1
  Translation Vector = 0 [m], 0 [m], 0 [m]
  X = 0.0 [m]
  Y = 0.0 [m]
  Z = 0.0 [m]
  END
END""")
```

```
results1.SendCommand(Command="""# Sending visibility action from ViewUtilities
>show /PLANE:span63, view=/VIEW:View 1""")
#
#
results1.SendCommand(Command="> autolegend plot=/POLYLINE:pl63, view=VIEW:View 1")
results1.SendCommand(Command="""POLYLINE:pl63
Apply Instancing Transform = On
Boundary List = ns_blade
Colour = 0, 1, 0
Colour Map = Default Colour Map
Colour Mode = Constant
Colour Scale = Linear
Colour Variable = Pressure
Colour Variable Boundary Values = Conservative
Contour Level = 1
Domain List = /DOMAIN GROUP:All Domains
Input File =
Instancing Transform = /DEFAULT INSTANCE TRANSFORM:Default Transform
Line Width = 2
Location = /PLANE:span63
Max = 0.0
Min = 0.0
Option = Boundary Intersection
Range = Global
  OBJECT VIEW TRANSFORM:
  Apply Reflection = Off
  Apply Rotation = Off
  Apply Scale = Off
  Apply Translation = Off
  Principal Axis = Z
  Reflection Plane Option = XY Plane
  Rotation Angle = 0.0 [degree]
  Rotation Axis From = 0 [m], 0 [m], 0 [m]
  Rotation Axis To = 0 [m], 0 [m], 0 [m]
  Rotation Axis Type = Principal Axis
  Scale Vector = 1 , 1 , 1
  Translation Vector = 0 [m], 0 [m], 0 [m]
  X = 0.0 [m]
  Y = 0.0 [m]
  Z = 0.0 [m]
  END
END""")
results1.SendCommand(Command="""# Sending visibility action from ViewUtilities
>show /POLYLINE:pl63, view=/VIEW:View 1""")
#

# Creates plane at 72% span location
results1.SendCommand(Command="> autolegend plot=/PLANE:span72, view=VIEW:View 1")
results1.SendCommand(Command="""PLANE:span72
Apply Instancing Transform = On
Apply Texture = Off
```

```
Blend Texture = On
Bound Radius = 0.5 [m]
Colour = 0.75, 0.75, 0.75
Colour Map = Default Colour Map
Colour Mode = Constant
Colour Scale = Linear
Colour Variable = Pressure
Colour Variable Boundary Values = Conservative
Culling Mode = No Culling
Direction 1 Bound = 1.0 [m]
Direction 1 Orientation = 0 [degree]
Direction 1 Points = 10
Direction 2 Bound = 1.0 [m]
Direction 2 Points = 10
Domain List = /DOMAIN GROUP:All Domains
Draw Faces = On
Draw Lines = Off
Instancing Transform = /DEFAULT INSTANCE TRANSFORM:Default Transform
Invert Plane Bound = Off
Lighting = On
Line Colour = 0, 0, 0
Line Colour Mode = Default
Line Width = 1
Max = 0.0
Min = 0.0
Normal = 1 , 0 , 0
Option = ZX Plane
Plane Bound = None
Plane Type = Slice
Point = 0 [m], 0 [m], 0 [m]
Point 1 = 0 [m], 0 [m], 0 [m]
Point 2 = 1 [m], 0 [m], 0 [m]
Point 3 = 0 [m], 1 [m], 0 [m]
Range = Global
Render Edge Angle = 0 [degree]
Specular Lighting = On
Surface Drawing = Smooth Shading
Texture Angle = 0
Texture Direction = 0 , 1 , 0
Texture File =
Texture Material = Metal
Texture Position = 0 , 0
Texture Scale = 1
Texture Type = Predefined
Tile Texture = Off
Transform Texture = Off
Transparency = 0.0
X = 0.0 [m]
Y = 0.72*5.029
Z = 0.0 [m]
  OBJECT VIEW TRANSFORM:
```

```
  Apply Reflection = Off
  Apply Rotation = Off
  Apply Scale = Off
  Apply Translation = Off
  Principal Axis = Z
  Reflection Plane Option = XY Plane
  Rotation Angle = 0.0 [degree]
  Rotation Axis From = 0 [m], 0 [m], 0 [m]
  Rotation Axis To = 0 [m], 0 [m], 0 [m]
  Rotation Axis Type = Principal Axis
  Scale Vector = 1 , 1 , 1
  Translation Vector = 0 [m], 0 [m], 0 [m]
  X = 0.0 [m]
  Y = 0.0 [m]
  Z = 0.0 [m]
  END
END""")
results1.SendCommand(Command="""# Sending visibility action from ViewUtilities
>show /PLANE:span72, view=/VIEW:View 1""")
#
#
results1.SendCommand(Command="> autolegend plot=/POLYLINE:pl72, view=VIEW:View 1")
results1.SendCommand(Command="""POLYLINE:pl72
Apply Instancing Transform = On
Boundary List = ns_blade
Colour = 0, 1, 0
Colour Map = Default Colour Map
Colour Mode = Constant
Colour Scale = Linear
Colour Variable = Pressure
Colour Variable Boundary Values = Conservative
Contour Level = 1
Domain List = /DOMAIN GROUP:All Domains
Input File =
Instancing Transform = /DEFAULT INSTANCE TRANSFORM:Default Transform
Line Width = 2
Location = /PLANE:span72
Max = 0.0
Min = 0.0
Option = Boundary Intersection
Range = Global
  OBJECT VIEW TRANSFORM:
  Apply Reflection = Off
  Apply Rotation = Off
  Apply Scale = Off
  Apply Translation = Off
  Principal Axis = Z
  Reflection Plane Option = XY Plane
  Rotation Angle = 0.0 [degree]
  Rotation Axis From = 0 [m], 0 [m], 0 [m]
  Rotation Axis To = 0 [m], 0 [m], 0 [m]
```

```
  Rotation Axis Type = Principal Axis
  Scale Vector = 1 , 1 , 1
  Translation Vector = 0 [m], 0 [m], 0 [m]
  X = 0.0 [m]
  Y = 0.0 [m]
  Z = 0.0 [m]
  END
END""")
results1.SendCommand(Command="""# Sending visibility action from ViewUtilities
>show /POLYLINE:pl72, view=/VIEW:View 1""")
#

# Creates plane at 80% span location
results1.SendCommand(Command="> autolegend plot=/PLANE:span80, view=VIEW:View 1")
results1.SendCommand(Command="""PLANE:span80
Apply Instancing Transform = On
Apply Texture = Off
Blend Texture = On
Bound Radius = 0.5 [m]
Colour = 0.75, 0.75, 0.75
Colour Map = Default Colour Map
Colour Mode = Constant
Colour Scale = Linear
Colour Variable = Pressure
Colour Variable Boundary Values = Conservative
Culling Mode = No Culling
Direction 1 Bound = 1.0 [m]
Direction 1 Orientation = 0 [degree]
Direction 1 Points = 10
Direction 2 Bound = 1.0 [m]
Direction 2 Points = 10
Domain List = /DOMAIN GROUP:All Domains
Draw Faces = On
Draw Lines = Off
Instancing Transform = /DEFAULT INSTANCE TRANSFORM:Default Transform
Invert Plane Bound = Off
Lighting = On
Line Colour = 0, 0, 0
Line Colour Mode = Default
Line Width = 1
Max = 0.0
Min = 0.0
Normal = 1 , 0 , 0
Option = ZX Plane
Plane Bound = None
Plane Type = Slice
Point = 0 [m], 0 [m], 0 [m]
Point 1 = 0 [m], 0 [m], 0 [m]
Point 2 = 1 [m], 0 [m], 0 [m]
Point 3 = 0 [m], 1 [m], 0 [m]
Range = Global
```

```
Render Edge Angle = 0 [degree]
Specular Lighting = On
Surface Drawing = Smooth Shading
Texture Angle = 0
Texture Direction = 0 , 1 , 0
Texture File =
Texture Material = Metal
Texture Position = 0 , 0
Texture Scale = 1
Texture Type = Predefined
Tile Texture = Off
Transform Texture = Off
Transparency = 0.0
X = 0.0 [m]
Y = 0.80*5.029
Z = 0.0 [m]
  OBJECT VIEW TRANSFORM:
  Apply Reflection = Off
  Apply Rotation = Off
  Apply Scale = Off
  Apply Translation = Off
  Principal Axis = Z
  Reflection Plane Option = XY Plane
  Rotation Angle = 0.0 [degree]
  Rotation Axis From = 0 [m], 0 [m], 0 [m]
  Rotation Axis To = 0 [m], 0 [m], 0 [m]
  Rotation Axis Type = Principal Axis
  Scale Vector = 1 , 1 , 1
  Translation Vector = 0 [m], 0 [m], 0 [m]
  X = 0.0 [m]
  Y = 0.0 [m]
  Z = 0.0 [m]
  END
END""")
results1.SendCommand(Command="""# Sending visibility action from ViewUtilities
>show /PLANE:span80, view=/VIEW:View 1""")
#
#
results1.SendCommand(Command="> autolegend plot=/POLYLINE:pl80, view=VIEW:View 1")
results1.SendCommand(Command="""POLYLINE:pl80
Apply Instancing Transform = On
Boundary List = ns_blade
Colour = 0, 1, 0
Colour Map = Default Colour Map
Colour Mode = Constant
Colour Scale = Linear
Colour Variable = Pressure
Colour Variable Boundary Values = Conservative
Contour Level = 1
Domain List = /DOMAIN GROUP:All Domains
Input File =
```

```
Instancing Transform = /DEFAULT INSTANCE TRANSFORM:Default Transform
Line Width = 2
Location = /PLANE:span80
Max = 0.0
Min = 0.0
Option = Boundary Intersection
Range = Global
  OBJECT VIEW TRANSFORM:
  Apply Reflection = Off
  Apply Rotation = Off
  Apply Scale = Off
  Apply Translation = Off
  Principal Axis = Z
  Reflection Plane Option = XY Plane
  Rotation Angle = 0.0 [degree]
  Rotation Axis From = 0 [m], 0 [m], 0 [m]
  Rotation Axis To = 0 [m], 0 [m], 0 [m]
  Rotation Axis Type = Principal Axis
  Scale Vector = 1 , 1 , 1
  Translation Vector = 0 [m], 0 [m], 0 [m]
  X = 0.0 [m]
  Y = 0.0 [m]
  Z = 0.0 [m]
  END
END""")
results1.SendCommand(Command="""# Sending visibility action from ViewUtilities
>show /POLYLINE:pl80, view=/VIEW:View 1""")
#

# Creates plane at 85% span location
results1.SendCommand(Command="> autolegend plot=/PLANE:span85, view=VIEW:View 1")
results1.SendCommand(Command="""PLANE:span85
Apply Instancing Transform = On
Apply Texture = Off
Blend Texture = On
Bound Radius = 0.5 [m]
Colour = 0.75, 0.75, 0.75
Colour Map = Default Colour Map
Colour Mode = Constant
Colour Scale = Linear
Colour Variable = Pressure
Colour Variable Boundary Values = Conservative
Culling Mode = No Culling
Direction 1 Bound = 1.0 [m]
Direction 1 Orientation = 0 [degree]
Direction 1 Points = 10
Direction 2 Bound = 1.0 [m]
Direction 2 Points = 10
Domain List = /DOMAIN GROUP:All Domains
Draw Faces = On
Draw Lines = Off
```

```
Instancing Transform = /DEFAULT INSTANCE TRANSFORM:Default Transform
Invert Plane Bound = Off
Lighting = On
Line Colour = 0, 0, 0
Line Colour Mode = Default
Line Width = 1
Max = 0.0
Min = 0.0
Normal = 1 , 0 , 0
Option = ZX Plane
Plane Bound = None
Plane Type = Slice
Point = 0 [m], 0 [m], 0 [m]
Point 1 = 0 [m], 0 [m], 0 [m]
Point 2 = 1 [m], 0 [m], 0 [m]
Point 3 = 0 [m], 1 [m], 0 [m]
Range = Global
Render Edge Angle = 0 [degree]
Specular Lighting = On
Surface Drawing = Smooth Shading
Texture Angle = 0
Texture Direction = 0 , 1 , 0
Texture File =
Texture Material = Metal
Texture Position = 0 , 0
Texture Scale = 1
Texture Type = Predefined
Tile Texture = Off
Transform Texture = Off
Transparency = 0.0
X = 0.0 [m]
Y = 0.85*5.029
Z = 0.0 [m]
  OBJECT VIEW TRANSFORM:
  Apply Reflection = Off
  Apply Rotation = Off
  Apply Scale = Off
  Apply Translation = Off
  Principal Axis = Z
  Reflection Plane Option = XY Plane
  Rotation Angle = 0.0 [degree]
  Rotation Axis From = 0 [m], 0 [m], 0 [m]
  Rotation Axis To = 0 [m], 0 [m], 0 [m]
  Rotation Axis Type = Principal Axis
  Scale Vector = 1 , 1 , 1
  Translation Vector = 0 [m], 0 [m], 0 [m]
  X = 0.0 [m]
  Y = 0.0 [m]
  Z = 0.0 [m]
  END
END""")
```

```
results1.SendCommand(Command="""# Sending visibility action from ViewUtilities
>show /PLANE:span85, view=/VIEW:View 1""")
#
#
results1.SendCommand(Command="> autolegend plot=/POLYLINE:pl85, view=VIEW:View 1")
results1.SendCommand(Command="""POLYLINE:pl85
Apply Instancing Transform = On
Boundary List = ns_blade
Colour = 0, 1, 0
Colour Map = Default Colour Map
Colour Mode = Constant
Colour Scale = Linear
Colour Variable = Pressure
Colour Variable Boundary Values = Conservative
Contour Level = 1
Domain List = /DOMAIN GROUP:All Domains
Input File =
Instancing Transform = /DEFAULT INSTANCE TRANSFORM:Default Transform
Line Width = 2
Location = /PLANE:span85
Max = 0.0
Min = 0.0
Option = Boundary Intersection
Range = Global
  OBJECT VIEW TRANSFORM:
  Apply Reflection = Off
  Apply Rotation = Off
  Apply Scale = Off
  Apply Translation = Off
  Principal Axis = Z
  Reflection Plane Option = XY Plane
  Rotation Angle = 0.0 [degree]
  Rotation Axis From = 0 [m], 0 [m], 0 [m]
  Rotation Axis To = 0 [m], 0 [m], 0 [m]
  Rotation Axis Type = Principal Axis
  Scale Vector = 1 , 1 , 1
  Translation Vector = 0 [m], 0 [m], 0 [m]
  X = 0.0 [m]
  Y = 0.0 [m]
  Z = 0.0 [m]
  END
END""")
results1.SendCommand(Command="""# Sending visibility action from ViewUtilities
>show /POLYLINE:pl85, view=/VIEW:View 1""")
#

# Creates plane at 90% span location
results1.SendCommand(Command="> autolegend plot=/PLANE:span90, view=VIEW:View 1")
results1.SendCommand(Command="""PLANE:span90
Apply Instancing Transform = On
Apply Texture = Off
```

```
Blend Texture = On
Bound Radius = 0.5 [m]
Colour = 0.75, 0.75, 0.75
Colour Map = Default Colour Map
Colour Mode = Constant
Colour Scale = Linear
Colour Variable = Pressure
Colour Variable Boundary Values = Conservative
Culling Mode = No Culling
Direction 1 Bound = 1.0 [m]
Direction 1 Orientation = 0 [degree]
Direction 1 Points = 10
Direction 2 Bound = 1.0 [m]
Direction 2 Points = 10
Domain List = /DOMAIN GROUP:All Domains
Draw Faces = On
Draw Lines = Off
Instancing Transform = /DEFAULT INSTANCE TRANSFORM:Default Transform
Invert Plane Bound = Off
Lighting = On
Line Colour = 0, 0, 0
Line Colour Mode = Default
Line Width = 1
Max = 0.0
Min = 0.0
Normal = 1 , 0 , 0
Option = ZX Plane
Plane Bound = None
Plane Type = Slice
Point = 0 [m], 0 [m], 0 [m]
Point 1 = 0 [m], 0 [m], 0 [m]
Point 2 = 1 [m], 0 [m], 0 [m]
Point 3 = 0 [m], 1 [m], 0 [m]
Range = Global
Render Edge Angle = 0 [degree]
Specular Lighting = On
Surface Drawing = Smooth Shading
Texture Angle = 0
Texture Direction = 0 , 1 , 0
Texture File =
Texture Material = Metal
Texture Position = 0 , 0
Texture Scale = 1
Texture Type = Predefined
Tile Texture = Off
Transform Texture = Off
Transparency = 0.0
X = 0.0 [m]
Y = 0.90*5.029
Z = 0.0 [m]
  OBJECT VIEW TRANSFORM:
```

```
  Apply Reflection = Off
  Apply Rotation = Off
  Apply Scale = Off
  Apply Translation = Off
  Principal Axis = Z
  Reflection Plane Option = XY Plane
  Rotation Angle = 0.0 [degree]
  Rotation Axis From = 0 [m], 0 [m], 0 [m]
  Rotation Axis To = 0 [m], 0 [m], 0 [m]
  Rotation Axis Type = Principal Axis
  Scale Vector = 1 , 1 , 1
  Translation Vector = 0 [m], 0 [m], 0 [m]
  X = 0.0 [m]
  Y = 0.0 [m]
  Z = 0.0 [m]
  END
END""")
results1.SendCommand(Command="""# Sending visibility action from ViewUtilities
>show /PLANE:span90, view=/VIEW:View 1""")
#
#
results1.SendCommand(Command="> autolegend plot=/POLYLINE:pl90, view=VIEW:View 1")
results1.SendCommand(Command="""POLYLINE:pl90
Apply Instancing Transform = On
Boundary List = ns_blade
Colour = 0, 1, 0
Colour Map = Default Colour Map
Colour Mode = Constant
Colour Scale = Linear
Colour Variable = Pressure
Colour Variable Boundary Values = Conservative
Contour Level = 1
Domain List = /DOMAIN GROUP:All Domains
Input File =
Instancing Transform = /DEFAULT INSTANCE TRANSFORM:Default Transform
Line Width = 2
Location = /PLANE:span90
Max = 0.0
Min = 0.0
Option = Boundary Intersection
Range = Global
  OBJECT VIEW TRANSFORM:
  Apply Reflection = Off
  Apply Rotation = Off
  Apply Scale = Off
  Apply Translation = Off
  Principal Axis = Z
  Reflection Plane Option = XY Plane
  Rotation Angle = 0.0 [degree]
  Rotation Axis From = 0 [m], 0 [m], 0 [m]
  Rotation Axis To = 0 [m], 0 [m], 0 [m]
```

```
  Rotation Axis Type = Principal Axis
  Scale Vector = 1 , 1 , 1
  Translation Vector = 0 [m], 0 [m], 0 [m]
  X = 0.0 [m]
  Y = 0.0 [m]
  Z = 0.0 [m]
  END
END""")
results1.SendCommand(Command="""# Sending visibility action from ViewUtilities
>show /POLYLINE:pl90, view=/VIEW:View 1""")
#

# Creates plane at 95% span location
results1.SendCommand(Command="> autolegend plot=/PLANE:span95, view=VIEW:View 1")
results1.SendCommand(Command="""PLANE:span95
Apply Instancing Transform = On
Apply Texture = Off
Blend Texture = On
Bound Radius = 0.5 [m]
Colour = 0.75, 0.75, 0.75
Colour Map = Default Colour Map
Colour Mode = Constant
Colour Scale = Linear
Colour Variable = Pressure
Colour Variable Boundary Values = Conservative
Culling Mode = No Culling
Direction 1 Bound = 1.0 [m]
Direction 1 Orientation = 0 [degree]
Direction 1 Points = 10
Direction 2 Bound = 1.0 [m]
Direction 2 Points = 10
Domain List = /DOMAIN GROUP:All Domains
Draw Faces = On
Draw Lines = Off
Instancing Transform = /DEFAULT INSTANCE TRANSFORM:Default Transform
Invert Plane Bound = Off
Lighting = On
Line Colour = 0, 0, 0
Line Colour Mode = Default
Line Width = 1
Max = 0.0
Min = 0.0
Normal = 1 , 0 , 0
Option = ZX Plane
Plane Bound = None
Plane Type = Slice
Point = 0 [m], 0 [m], 0 [m]
Point 1 = 0 [m], 0 [m], 0 [m]
Point 2 = 1 [m], 0 [m], 0 [m]
Point 3 = 0 [m], 1 [m], 0 [m]
Range = Global
```

```
Render Edge Angle = 0 [degree]
Specular Lighting = On
Surface Drawing = Smooth Shading
Texture Angle = 0
Texture Direction = 0 , 1 , 0
Texture File =
Texture Material = Metal
Texture Position = 0 , 0
Texture Scale = 1
Texture Type = Predefined
Tile Texture = Off
Transform Texture = Off
Transparency = 0.0
X = 0.0 [m]
Y = 0.95*5.029
Z = 0.0 [m]
  OBJECT VIEW TRANSFORM:
  Apply Reflection = Off
  Apply Rotation = Off
  Apply Scale = Off
  Apply Translation = Off
  Principal Axis = Z
  Reflection Plane Option = XY Plane
  Rotation Angle = 0.0 [degree]
  Rotation Axis From = 0 [m], 0 [m], 0 [m]
  Rotation Axis To = 0 [m], 0 [m], 0 [m]
  Rotation Axis Type = Principal Axis
  Scale Vector = 1 , 1 , 1
  Translation Vector = 0 [m], 0 [m], 0 [m]
  X = 0.0 [m]
  Y = 0.0 [m]
  Z = 0.0 [m]
  END
END""")
results1.SendCommand(Command="""# Sending visibility action from ViewUtilities
>show /PLANE:span95, view=/VIEW:View 1""")
#
#
results1.SendCommand(Command="> autolegend plot=/POLYLINE:pl95, view=VIEW:View 1")
results1.SendCommand(Command="""POLYLINE:pl95
Apply Instancing Transform = On
Boundary List = ns_blade
Colour = 0, 1, 0
Colour Map = Default Colour Map
Colour Mode = Constant
Colour Scale = Linear
Colour Variable = Pressure
Colour Variable Boundary Values = Conservative
Contour Level = 1
Domain List = /DOMAIN GROUP:All Domains
Input File =
```

```
Instancing Transform = /DEFAULT INSTANCE TRANSFORM:Default Transform
Line Width = 2
Location = /PLANE:span95
Max = 0.0
Min = 0.0
Option = Boundary Intersection
Range = Global
  OBJECT VIEW TRANSFORM:
  Apply Reflection = Off
  Apply Rotation = Off
  Apply Scale = Off
  Apply Translation = Off
  Principal Axis = Z
  Reflection Plane Option = XY Plane
  Rotation Angle = 0.0 [degree]
  Rotation Axis From = 0 [m], 0 [m], 0 [m]
  Rotation Axis To = 0 [m], 0 [m], 0 [m]
  Rotation Axis Type = Principal Axis
  Scale Vector = 1 , 1 , 1
  Translation Vector = 0 [m], 0 [m], 0 [m]
  X = 0.0 [m]
  Y = 0.0 [m]
  Z = 0.0 [m]
  END
END""")
results1.SendCommand(Command="""# Sending visibility action from ViewUtilities
>show /POLYLINE:pl95, view=/VIEW:View 1""")
#

#Pressure data at the 30% span location
results1.SendCommand(Command="""CHART:pressure30
Chart Axes Font = Tahoma, 10, False, False, False, False
Chart Axes Titles Font = Tahoma, 10, True, False, False, False
Chart Grid Line Width = 1
Chart Horizontal Grid = On
Chart Legend = On
Chart Legend Font = Tahoma, 8, False, False, False, False
Chart Legend Inside = Outside Chart
Chart Legend Justification = Center
Chart Legend Position = Bottom
Chart Legend Width Height = 0.2 , 0.4
Chart Legend X Justification = Right
Chart Legend XY Position = 0.73 , 0.275
Chart Legend Y Justification = Center
Chart Line Width = 2
Chart Lines Order = Series 1,Chart Line 1
Chart Minor Grid = Off
Chart Minor Grid Line Width = 1
Chart Symbol Size = 4
Chart Title = Title
Chart Title Font = Tahoma, 12, True, False, False, False
```

```
Chart Title Visibility = On
Chart Type = XY
Chart Vertical Grid = On
Chart X Axis Automatic Number Formatting = On
Chart X Axis Label = X Axis <units>
Chart X Axis Number Format = %10.3e
Chart Y Axis Automatic Number Formatting = On
Chart Y Axis Label = Y Axis <units>
Chart Y Axis Number Format = %10.3e
Default Chart X Variable = X
Default Chart Y Variable = Pressure
Default Difference Line Calculation = From Points
Default Histogram Y Axis Weighting = None
Default Time Chart Variable = Pressure
Default Time Chart X Expression = Time
Default Time Variable Absolute Value = Off
Default Time Variable Boundary Values = Conservative
Default X Variable Absolute Value = Off
Default X Variable Boundary Values = Conservative
Default Y Variable Absolute Value = Off
Default Y Variable Boundary Values = Conservative
FFT Full Input Range = On
FFT Max = 0.0
FFT Min = 0.0
FFT Subtract Mean = Off
FFT Window Type = Hanning
FFT X Function = Frequency
FFT Y Function = Power Spectral Density
Histogram Automatic Divisions = Automatic
Histogram Divisions = -1.0,1.0
Histogram Divisions Count = 10
Histogram Y Axis Value = Count
Is FFT Chart = Off
Max X = 1.0
Max Y = 1.0
Min X = -1.0
Min Y = -1.0
Time Chart Keep Single Case = Off
Use Data For X Axis Labels = On
Use Data For Y Axis Labels = On
X Axis Automatic Range = On
X Axis Inverted = Off
X Axis Logarithmic Scaling = Off
Y Axis Automatic Range = On
Y Axis Inverted = Off
Y Axis Logarithmic Scaling = Off
  CHART SERIES:Series 1
  Chart Line Custom Data Selection = Off
  Chart Line Filename =
  Chart Series Type = Regular
  Chart X Variable = Chart Count
```

```
  Chart Y Variable = Density
  Histogram Y Axis Weighting = None
  Location = /POLYLINE:pl30
  Monitor Data Filename =
  Monitor Data Source = Case
  Monitor Data X Variable Absolute Value = Off
  Monitor Data Y Variable Absolute Value = Off
  Series Name = Series 1
  Time Chart Expression = Time
  Time Chart Type = Point
  Time Chart Variable = Density
  Time Chart X Expression = Time
  Time Variable Absolute Value = Off
  Time Variable Boundary Values = Conservative
  X Variable Absolute Value = Off
  X Variable Boundary Values = Conservative
  Y Variable Absolute Value = Off
  Y Variable Boundary Values = Conservative
    CHART LINE:Chart Line 1
    Auto Chart Line Colour = On
    Chart FFT Line Type = Bars
    Chart Line Colour = 1.0, 0.0, 0.0
    Chart Line Style = Automatic
    Chart Line Type = Lines
    Chart Line Visibility = On
    Chart Symbol Colour = 0.0, 1.0, 0.0
    Chart Symbol Style = None
    Fill Area = On
    Fill Area Options = Automatic
    Is Valid = True
    Line Name = Series 1
    Use Automatic Line Naming = On
    END
  END
  OBJECT REPORT OPTIONS:
      Report Caption =
  END
END""")
results1.SendCommand(Command="""EXPORT:
 Export File = C:/Users/User/Desktop/scripts/Results/30.csv
 Export Chart Name = pressure30
 Overwrite = On
END
>export chart""")
#

#Pressure data at the 39% span location
results1.SendCommand(Command="""CHART:pressure39
Chart Axes Font = Tahoma, 10, False, False, False, False
Chart Axes Titles Font = Tahoma, 10, True, False, False, False
Chart Grid Line Width = 1
```

```
Chart Horizontal Grid = On
Chart Legend = On
Chart Legend Font = Tahoma, 8, False, False, False, False
Chart Legend Inside = Outside Chart
Chart Legend Justification = Center
Chart Legend Position = Bottom
Chart Legend Width Height = 0.2 , 0.4
Chart Legend X Justification = Right
Chart Legend XY Position = 0.73 , 0.275
Chart Legend Y Justification = Center
Chart Line Width = 2
Chart Lines Order = Series 1,Chart Line 1
Chart Minor Grid = Off
Chart Minor Grid Line Width = 1
Chart Symbol Size = 4
Chart Title = Title
Chart Title Font = Tahoma, 12, True, False, False, False
Chart Title Visibility = On
Chart Type = XY
Chart Vertical Grid = On
Chart X Axis Automatic Number Formatting = On
Chart X Axis Label = X Axis <units>
Chart X Axis Number Format = %10.3e
Chart Y Axis Automatic Number Formatting = On
Chart Y Axis Label = Y Axis <units>
Chart Y Axis Number Format = %10.3e
Default Chart X Variable = X
Default Chart Y Variable = Pressure
Default Difference Line Calculation = From Points
Default Histogram Y Axis Weighting = None
Default Time Chart Variable = Pressure
Default Time Chart X Expression = Time
Default Time Variable Absolute Value = Off
Default Time Variable Boundary Values = Conservative
Default X Variable Absolute Value = Off
Default X Variable Boundary Values = Conservative
Default Y Variable Absolute Value = Off
Default Y Variable Boundary Values = Conservative
FFT Full Input Range = On
FFT Max = 0.0
FFT Min = 0.0
FFT Subtract Mean = Off
FFT Window Type = Hanning
FFT X Function = Frequency
FFT Y Function = Power Spectral Density
Histogram Automatic Divisions = Automatic
Histogram Divisions = -1.0,1.0
Histogram Divisions Count = 10
Histogram Y Axis Value = Count
Is FFT Chart = Off
Max X = 1.0
```

```
Max Y = 1.0
Min X = -1.0
Min Y = -1.0
Time Chart Keep Single Case = Off
Use Data For X Axis Labels = On
Use Data For Y Axis Labels = On
X Axis Automatic Range = On
X Axis Inverted = Off
X Axis Logarithmic Scaling = Off
Y Axis Automatic Range = On
Y Axis Inverted = Off
Y Axis Logarithmic Scaling = Off
  CHART SERIES:Series 1
  Chart Line Custom Data Selection = Off
  Chart Line Filename =
  Chart Series Type = Regular
  Chart X Variable = Chart Count
  Chart Y Variable = Density
  Histogram Y Axis Weighting = None
  Location = /POLYLINE:pl39
  Monitor Data Filename =
  Monitor Data Source = Case
  Monitor Data X Variable Absolute Value = Off
  Monitor Data Y Variable Absolute Value = Off
  Series Name = Series 1
  Time Chart Expression = Time
  Time Chart Type = Point
  Time Chart Variable = Density
  Time Chart X Expression = Time
  Time Variable Absolute Value = Off
  Time Variable Boundary Values = Conservative
  X Variable Absolute Value = Off
  X Variable Boundary Values = Conservative
  Y Variable Absolute Value = Off
  Y Variable Boundary Values = Conservative
    CHART LINE:Chart Line 1
    Auto Chart Line Colour = On
    Chart FFT Line Type = Bars
    Chart Line Colour = 1.0, 0.0, 0.0
    Chart Line Style = Automatic
    Chart Line Type = Lines
    Chart Line Visibility = On
    Chart Symbol Colour = 0.0, 1.0, 0.0
    Chart Symbol Style = None
    Fill Area = On
    Fill Area Options = Automatic
    Is Valid = True
    Line Name = Series 1
    Use Automatic Line Naming = On
    END
  END
```

```
  OBJECT REPORT OPTIONS:
    Report Caption =
  END
END""")
results1.SendCommand(Command="""EXPORT:
 Export File = C:/Users/User/Desktop/scripts/Results/39.csv
 Export Chart Name = pressure39
 Overwrite = On
END
>export chart""")
#

#Pressure data at the 47% span location
results1.SendCommand(Command="""CHART:pressure47
Chart Axes Font = Tahoma, 10, False, False, False, False
Chart Axes Titles Font = Tahoma, 10, True, False, False, False
Chart Grid Line Width = 1
Chart Horizontal Grid = On
Chart Legend = On
Chart Legend Font = Tahoma, 8, False, False, False, False
Chart Legend Inside = Outside Chart
Chart Legend Justification = Center
Chart Legend Position = Bottom
Chart Legend Width Height = 0.2 , 0.4
Chart Legend X Justification = Right
Chart Legend XY Position = 0.73 , 0.275
Chart Legend Y Justification = Center
Chart Line Width = 2
Chart Lines Order = Series 1,Chart Line 1
Chart Minor Grid = Off
Chart Minor Grid Line Width = 1
Chart Symbol Size = 4
Chart Title = Title
Chart Title Font = Tahoma, 12, True, False, False, False
Chart Title Visibility = On
Chart Type = XY
Chart Vertical Grid = On
Chart X Axis Automatic Number Formatting = On
Chart X Axis Label = X Axis <units>
Chart X Axis Number Format = %10.3e
Chart Y Axis Automatic Number Formatting = On
Chart Y Axis Label = Y Axis <units>
Chart Y Axis Number Format = %10.3e
Default Chart X Variable = X
Default Chart Y Variable = Pressure
Default Difference Line Calculation = From Points
Default Histogram Y Axis Weighting = None
Default Time Chart Variable = Pressure
Default Time Chart X Expression = Time
Default Time Variable Absolute Value = Off
Default Time Variable Boundary Values = Conservative
```

```
Default X Variable Absolute Value = Off
Default X Variable Boundary Values = Conservative
Default Y Variable Absolute Value = Off
Default Y Variable Boundary Values = Conservative
FFT Full Input Range = On
FFT Max = 0.0
FFT Min = 0.0
FFT Subtract Mean = Off
FFT Window Type = Hanning
FFT X Function = Frequency
FFT Y Function = Power Spectral Density
Histogram Automatic Divisions = Automatic
Histogram Divisions = -1.0,1.0
Histogram Divisions Count = 10
Histogram Y Axis Value = Count
Is FFT Chart = Off
Max X = 1.0
Max Y = 1.0
Min X = -1.0
Min Y = -1.0
Time Chart Keep Single Case = Off
Use Data For X Axis Labels = On
Use Data For Y Axis Labels = On
X Axis Automatic Range = On
X Axis Inverted = Off
X Axis Logarithmic Scaling = Off
Y Axis Automatic Range = On
Y Axis Inverted = Off
Y Axis Logarithmic Scaling = Off
  CHART SERIES:Series 1
  Chart Line Custom Data Selection = Off
  Chart Line Filename =
  Chart Series Type = Regular
  Chart X Variable = Chart Count
  Chart Y Variable = Density
  Histogram Y Axis Weighting = None
  Location = /POLYLINE:pl47
  Monitor Data Filename =
  Monitor Data Source = Case
  Monitor Data X Variable Absolute Value = Off
  Monitor Data Y Variable Absolute Value = Off
  Series Name = Series 1
  Time Chart Expression = Time
  Time Chart Type = Point
  Time Chart Variable = Density
  Time Chart X Expression = Time
  Time Variable Absolute Value = Off
  Time Variable Boundary Values = Conservative
  X Variable Absolute Value = Off
  X Variable Boundary Values = Conservative
  Y Variable Absolute Value = Off
```

```
  Y Variable Boundary Values = Conservative
    CHART LINE:Chart Line 1
    Auto Chart Line Colour = On
    Chart FFT Line Type = Bars
    Chart Line Colour = 1.0, 0.0, 0.0
    Chart Line Style = Automatic
    Chart Line Type = Lines
    Chart Line Visibility = On
    Chart Symbol Colour = 0.0, 1.0, 0.0
    Chart Symbol Style = None
    Fill Area = On
    Fill Area Options = Automatic
    Is Valid = True
    Line Name = Series 1
    Use Automatic Line Naming = On
    END
  END
  OBJECT REPORT OPTIONS:
      Report Caption =
  END
END""")
results1.SendCommand(Command="""EXPORT:
 Export File = C:/Users/User/Desktop/scripts/Results/47.csv
 Export Chart Name = pressure47
 Overwrite = On
END
>export chart""")
#

#Pressure data at the 55% span location
results1.SendCommand(Command="""CHART:pressure55
Chart Axes Font = Tahoma, 10, False, False, False, False
Chart Axes Titles Font = Tahoma, 10, True, False, False, False
Chart Grid Line Width = 1
Chart Horizontal Grid = On
Chart Legend = On
Chart Legend Font = Tahoma, 8, False, False, False, False
Chart Legend Inside = Outside Chart
Chart Legend Justification = Center
Chart Legend Position = Bottom
Chart Legend Width Height = 0.2 , 0.4
Chart Legend X Justification = Right
Chart Legend XY Position = 0.73 , 0.275
Chart Legend Y Justification = Center
Chart Line Width = 2
Chart Lines Order = Series 1,Chart Line 1
Chart Minor Grid = Off
Chart Minor Grid Line Width = 1
Chart Symbol Size = 4
Chart Title = Title
Chart Title Font = Tahoma, 12, True, False, False, False
```

```
Chart Title Visibility = On
Chart Type = XY
Chart Vertical Grid = On
Chart X Axis Automatic Number Formatting = On
Chart X Axis Label = X Axis <units>
Chart X Axis Number Format = %10.3e
Chart Y Axis Automatic Number Formatting = On
Chart Y Axis Label = Y Axis <units>
Chart Y Axis Number Format = %10.3e
Default Chart X Variable = X
Default Chart Y Variable = Pressure
Default Difference Line Calculation = From Points
Default Histogram Y Axis Weighting = None
Default Time Chart Variable = Pressure
Default Time Chart X Expression = Time
Default Time Variable Absolute Value = Off
Default Time Variable Boundary Values = Conservative
Default X Variable Absolute Value = Off
Default X Variable Boundary Values = Conservative
Default Y Variable Absolute Value = Off
Default Y Variable Boundary Values = Conservative
FFT Full Input Range = On
FFT Max = 0.0
FFT Min = 0.0
FFT Subtract Mean = Off
FFT Window Type = Hanning
FFT X Function = Frequency
FFT Y Function = Power Spectral Density
Histogram Automatic Divisions = Automatic
Histogram Divisions = -1.0,1.0
Histogram Divisions Count = 10
Histogram Y Axis Value = Count
Is FFT Chart = Off
Max X = 1.0
Max Y = 1.0
Min X = -1.0
Min Y = -1.0
Time Chart Keep Single Case = Off
Use Data For X Axis Labels = On
Use Data For Y Axis Labels = On
X Axis Automatic Range = On
X Axis Inverted = Off
X Axis Logarithmic Scaling = Off
Y Axis Automatic Range = On
Y Axis Inverted = Off
Y Axis Logarithmic Scaling = Off
  CHART SERIES:Series 1
  Chart Line Custom Data Selection = Off
  Chart Line Filename =
  Chart Series Type = Regular
  Chart X Variable = Chart Count
```

```
  Chart Y Variable = Density
  Histogram Y Axis Weighting = None
  Location = /POLYLINE:pl55
  Monitor Data Filename =
  Monitor Data Source = Case
  Monitor Data X Variable Absolute Value = Off
  Monitor Data Y Variable Absolute Value = Off
  Series Name = Series 1
  Time Chart Expression = Time
  Time Chart Type = Point
  Time Chart Variable = Density
  Time Chart X Expression = Time
  Time Variable Absolute Value = Off
  Time Variable Boundary Values = Conservative
  X Variable Absolute Value = Off
  X Variable Boundary Values = Conservative
  Y Variable Absolute Value = Off
  Y Variable Boundary Values = Conservative
   CHART LINE:Chart Line 1
   Auto Chart Line Colour = On
   Chart FFT Line Type = Bars
   Chart Line Colour = 1.0, 0.0, 0.0
   Chart Line Style = Automatic
   Chart Line Type = Lines
   Chart Line Visibility = On
   Chart Symbol Colour = 0.0, 1.0, 0.0
   Chart Symbol Style = None
   Fill Area = On
   Fill Area Options = Automatic
   Is Valid = True
   Line Name = Series 1
   Use Automatic Line Naming = On
   END
  END
  OBJECT REPORT OPTIONS:
    Report Caption =
  END
END""")
results1.SendCommand(Command="""EXPORT:
 Export File = C:/Users/User/Desktop/scripts/Results/55.csv
 Export Chart Name = pressure55
 Overwrite = On
END
>export chart""")
#

# Pressure data at the 63% span location
results1.SendCommand(Command="""CHART:pressure63
Chart Axes Font = Tahoma, 10, False, False, False, False
Chart Axes Titles Font = Tahoma, 10, True, False, False, False
Chart Grid Line Width = 1
```

```
Chart Horizontal Grid = On
Chart Legend = On
Chart Legend Font = Tahoma, 8, False, False, False, False
Chart Legend Inside = Outside Chart
Chart Legend Justification = Center
Chart Legend Position = Bottom
Chart Legend Width Height = 0.2 , 0.4
Chart Legend X Justification = Right
Chart Legend XY Position = 0.73 , 0.275
Chart Legend Y Justification = Center
Chart Line Width = 2
Chart Lines Order = Series 1,Chart Line 1
Chart Minor Grid = Off
Chart Minor Grid Line Width = 1
Chart Symbol Size = 4
Chart Title = Title
Chart Title Font = Tahoma, 12, True, False, False, False
Chart Title Visibility = On
Chart Type = XY
Chart Vertical Grid = On
Chart X Axis Automatic Number Formatting = On
Chart X Axis Label = X Axis <units>
Chart X Axis Number Format = %10.3e
Chart Y Axis Automatic Number Formatting = On
Chart Y Axis Label = Y Axis <units>
Chart Y Axis Number Format = %10.3e
Default Chart X Variable = X
Default Chart Y Variable = Pressure
Default Difference Line Calculation = From Points
Default Histogram Y Axis Weighting = None
Default Time Chart Variable = Pressure
Default Time Chart X Expression = Time
Default Time Variable Absolute Value = Off
Default Time Variable Boundary Values = Conservative
Default X Variable Absolute Value = Off
Default X Variable Boundary Values = Conservative
Default Y Variable Absolute Value = Off
Default Y Variable Boundary Values = Conservative
FFT Full Input Range = On
FFT Max = 0.0
FFT Min = 0.0
FFT Subtract Mean = Off
FFT Window Type = Hanning
FFT X Function = Frequency
FFT Y Function = Power Spectral Density
Histogram Automatic Divisions = Automatic
Histogram Divisions = -1.0,1.0
Histogram Divisions Count = 10
Histogram Y Axis Value = Count
Is FFT Chart = Off
Max X = 1.0
```

```
Max Y = 1.0
Min X = -1.0
Min Y = -1.0
Time Chart Keep Single Case = Off
Use Data For X Axis Labels = On
Use Data For Y Axis Labels = On
X Axis Automatic Range = On
X Axis Inverted = Off
X Axis Logarithmic Scaling = Off
Y Axis Automatic Range = On
Y Axis Inverted = Off
Y Axis Logarithmic Scaling = Off
  CHART SERIES:Series 1
  Chart Line Custom Data Selection = Off
  Chart Line Filename =
  Chart Series Type = Regular
  Chart X Variable = Chart Count
  Chart Y Variable = Density
  Histogram Y Axis Weighting = None
  Location = /POLYLINE:pl63
  Monitor Data Filename =
  Monitor Data Source = Case
  Monitor Data X Variable Absolute Value = Off
  Monitor Data Y Variable Absolute Value = Off
  Series Name = Series 1
  Time Chart Expression = Time
  Time Chart Type = Point
  Time Chart Variable = Density
  Time Chart X Expression = Time
  Time Variable Absolute Value = Off
  Time Variable Boundary Values = Conservative
  X Variable Absolute Value = Off
  X Variable Boundary Values = Conservative
  Y Variable Absolute Value = Off
  Y Variable Boundary Values = Conservative
    CHART LINE:Chart Line 1
    Auto Chart Line Colour = On
    Chart FFT Line Type = Bars
    Chart Line Colour = 1.0, 0.0, 0.0
    Chart Line Style = Automatic
    Chart Line Type = Lines
    Chart Line Visibility = On
    Chart Symbol Colour = 0.0, 1.0, 0.0
    Chart Symbol Style = None
    Fill Area = On
    Fill Area Options = Automatic
    Is Valid = True
    Line Name = Series 1
    Use Automatic Line Naming = On
    END
  END
```

```
  OBJECT REPORT OPTIONS:
      Report Caption =
  END
END""")
results1.SendCommand(Command="""EXPORT:
 Export File = C:/Users/User/Desktop/scripts/Results/63.csv
 Export Chart Name = pressure63
 Overwrite = On
END
>export chart""")
#

#Pressure data at the 72% span location
results1.SendCommand(Command="""CHART:pressure72
Chart Axes Font = Tahoma, 10, False, False, False, False
Chart Axes Titles Font = Tahoma, 10, True, False, False, False
Chart Grid Line Width = 1
Chart Horizontal Grid = On
Chart Legend = On
Chart Legend Font = Tahoma, 8, False, False, False, False
Chart Legend Inside = Outside Chart
Chart Legend Justification = Center
Chart Legend Position = Bottom
Chart Legend Width Height = 0.2 , 0.4
Chart Legend X Justification = Right
Chart Legend XY Position = 0.73 , 0.275
Chart Legend Y Justification = Center
Chart Line Width = 2
Chart Lines Order = Series 1,Chart Line 1
Chart Minor Grid = Off
Chart Minor Grid Line Width = 1
Chart Symbol Size = 4
Chart Title = Title
Chart Title Font = Tahoma, 12, True, False, False, False
Chart Title Visibility = On
Chart Type = XY
Chart Vertical Grid = On
Chart X Axis Automatic Number Formatting = On
Chart X Axis Label = X Axis <units>
Chart X Axis Number Format = %10.3e
Chart Y Axis Automatic Number Formatting = On
Chart Y Axis Label = Y Axis <units>
Chart Y Axis Number Format = %10.3e
Default Chart X Variable = X
Default Chart Y Variable = Pressure
Default Difference Line Calculation = From Points
Default Histogram Y Axis Weighting = None
Default Time Chart Variable = Pressure
Default Time Chart X Expression = Time
Default Time Variable Absolute Value = Off
Default Time Variable Boundary Values = Conservative
```

```
Default X Variable Absolute Value = Off
Default X Variable Boundary Values = Conservative
Default Y Variable Absolute Value = Off
Default Y Variable Boundary Values = Conservative
FFT Full Input Range = On
FFT Max = 0.0
FFT Min = 0.0
FFT Subtract Mean = Off
FFT Window Type = Hanning
FFT X Function = Frequency
FFT Y Function = Power Spectral Density
Histogram Automatic Divisions = Automatic
Histogram Divisions = -1.0,1.0
Histogram Divisions Count = 10
Histogram Y Axis Value = Count
Is FFT Chart = Off
Max X = 1.0
Max Y = 1.0
Min X = -1.0
Min Y = -1.0
Time Chart Keep Single Case = Off
Use Data For X Axis Labels = On
Use Data For Y Axis Labels = On
X Axis Automatic Range = On
X Axis Inverted = Off
X Axis Logarithmic Scaling = Off
Y Axis Automatic Range = On
Y Axis Inverted = Off
Y Axis Logarithmic Scaling = Off
  CHART SERIES:Series 1
  Chart Line Custom Data Selection = Off
  Chart Line Filename =
  Chart Series Type = Regular
  Chart X Variable = Chart Count
  Chart Y Variable = Density
  Histogram Y Axis Weighting = None
  Location = /POLYLINE:pl72
  Monitor Data Filename =
  Monitor Data Source = Case
  Monitor Data X Variable Absolute Value = Off
  Monitor Data Y Variable Absolute Value = Off
  Series Name = Series 1
  Time Chart Expression = Time
  Time Chart Type = Point
  Time Chart Variable = Density
  Time Chart X Expression = Time
  Time Variable Absolute Value = Off
  Time Variable Boundary Values = Conservative
  X Variable Absolute Value = Off
  X Variable Boundary Values = Conservative
  Y Variable Absolute Value = Off
```

```
  Y Variable Boundary Values = Conservative
    CHART LINE:Chart Line 1
    Auto Chart Line Colour = On
    Chart FFT Line Type = Bars
    Chart Line Colour = 1.0, 0.0, 0.0
    Chart Line Style = Automatic
    Chart Line Type = Lines
    Chart Line Visibility = On
    Chart Symbol Colour = 0.0, 1.0, 0.0
    Chart Symbol Style = None
    Fill Area = On
    Fill Area Options = Automatic
    Is Valid = True
    Line Name = Series 1
    Use Automatic Line Naming = On
    END
  END
  OBJECT REPORT OPTIONS:
      Report Caption =
  END
END""")
results1.SendCommand(Command="""EXPORT:
 Export File = C:/Users/User/Desktop/scripts/Results/72.csv
 Export Chart Name = pressure72
 Overwrite = On
END
>export chart""")
#

#Pressure data at the 80% span location
results1.SendCommand(Command="""CHART:pressure80
Chart Axes Font = Tahoma, 10, False, False, False, False
Chart Axes Titles Font = Tahoma, 10, True, False, False, False
Chart Grid Line Width = 1
Chart Horizontal Grid = On
Chart Legend = On
Chart Legend Font = Tahoma, 8, False, False, False, False
Chart Legend Inside = Outside Chart
Chart Legend Justification = Center
Chart Legend Position = Bottom
Chart Legend Width Height = 0.2 , 0.4
Chart Legend X Justification = Right
Chart Legend XY Position = 0.73 , 0.275
Chart Legend Y Justification = Center
Chart Line Width = 2
Chart Lines Order = Series 1,Chart Line 1
Chart Minor Grid = Off
Chart Minor Grid Line Width = 1
Chart Symbol Size = 4
Chart Title = Title
Chart Title Font = Tahoma, 12, True, False, False, False
```

```
Chart Title Visibility = On
Chart Type = XY
Chart Vertical Grid = On
Chart X Axis Automatic Number Formatting = On
Chart X Axis Label = X Axis <units>
Chart X Axis Number Format = %10.3e
Chart Y Axis Automatic Number Formatting = On
Chart Y Axis Label = Y Axis <units>
Chart Y Axis Number Format = %10.3e
Default Chart X Variable = X
Default Chart Y Variable = Pressure
Default Difference Line Calculation = From Points
Default Histogram Y Axis Weighting = None
Default Time Chart Variable = Pressure
Default Time Chart X Expression = Time
Default Time Variable Absolute Value = Off
Default Time Variable Boundary Values = Conservative
Default X Variable Absolute Value = Off
Default X Variable Boundary Values = Conservative
Default Y Variable Absolute Value = Off
Default Y Variable Boundary Values = Conservative
FFT Full Input Range = On
FFT Max = 0.0
FFT Min = 0.0
FFT Subtract Mean = Off
FFT Window Type = Hanning
FFT X Function = Frequency
FFT Y Function = Power Spectral Density
Histogram Automatic Divisions = Automatic
Histogram Divisions = -1.0,1.0
Histogram Divisions Count = 10
Histogram Y Axis Value = Count
Is FFT Chart = Off
Max X = 1.0
Max Y = 1.0
Min X = -1.0
Min Y = -1.0
Time Chart Keep Single Case = Off
Use Data For X Axis Labels = On
Use Data For Y Axis Labels = On
X Axis Automatic Range = On
X Axis Inverted = Off
X Axis Logarithmic Scaling = Off
Y Axis Automatic Range = On
Y Axis Inverted = Off
Y Axis Logarithmic Scaling = Off
  CHART SERIES:Series 1
  Chart Line Custom Data Selection = Off
  Chart Line Filename =
  Chart Series Type = Regular
  Chart X Variable = Chart Count
```

```
  Chart Y Variable = Density
  Histogram Y Axis Weighting = None
  Location = /POLYLINE:pl80
  Monitor Data Filename =
  Monitor Data Source = Case
  Monitor Data X Variable Absolute Value = Off
  Monitor Data Y Variable Absolute Value = Off
  Series Name = Series 1
  Time Chart Expression = Time
  Time Chart Type = Point
  Time Chart Variable = Density
  Time Chart X Expression = Time
  Time Variable Absolute Value = Off
  Time Variable Boundary Values = Conservative
  X Variable Absolute Value = Off
  X Variable Boundary Values = Conservative
  Y Variable Absolute Value = Off
  Y Variable Boundary Values = Conservative
    CHART LINE:Chart Line 1
    Auto Chart Line Colour = On
    Chart FFT Line Type = Bars
    Chart Line Colour = 1.0, 0.0, 0.0
    Chart Line Style = Automatic
    Chart Line Type = Lines
    Chart Line Visibility = On
    Chart Symbol Colour = 0.0, 1.0, 0.0
    Chart Symbol Style = None
    Fill Area = On
    Fill Area Options = Automatic
    Is Valid = True
    Line Name = Series 1
    Use Automatic Line Naming = On
    END
  END
  OBJECT REPORT OPTIONS:
      Report Caption =
  END
END""")
results1.SendCommand(Command="""EXPORT:
 Export File = C:/Users/User/Desktop/scripts/Results/80.csv
 Export Chart Name = pressure80
 Overwrite = On
END
>export chart""")
#

#Pressure data at the 85% span location
results1.SendCommand(Command="""CHART:pressure85
Chart Axes Font = Tahoma, 10, False, False, False, False
Chart Axes Titles Font = Tahoma, 10, True, False, False, False
Chart Grid Line Width = 1
```

```
Chart Horizontal Grid = On
Chart Legend = On
Chart Legend Font = Tahoma, 8, False, False, False, False
Chart Legend Inside = Outside Chart
Chart Legend Justification = Center
Chart Legend Position = Bottom
Chart Legend Width Height = 0.2 , 0.4
Chart Legend X Justification = Right
Chart Legend XY Position = 0.73 , 0.275
Chart Legend Y Justification = Center
Chart Line Width = 2
Chart Lines Order = Series 1,Chart Line 1
Chart Minor Grid = Off
Chart Minor Grid Line Width = 1
Chart Symbol Size = 4
Chart Title = Title
Chart Title Font = Tahoma, 12, True, False, False, False
Chart Title Visibility = On
Chart Type = XY
Chart Vertical Grid = On
Chart X Axis Automatic Number Formatting = On
Chart X Axis Label = X Axis <units>
Chart X Axis Number Format = %10.3e
Chart Y Axis Automatic Number Formatting = On
Chart Y Axis Label = Y Axis <units>
Chart Y Axis Number Format = %10.3e
Default Chart X Variable = X
Default Chart Y Variable = Pressure
Default Difference Line Calculation = From Points
Default Histogram Y Axis Weighting = None
Default Time Chart Variable = Pressure
Default Time Chart X Expression = Time
Default Time Variable Absolute Value = Off
Default Time Variable Boundary Values = Conservative
Default X Variable Absolute Value = Off
Default X Variable Boundary Values = Conservative
Default Y Variable Absolute Value = Off
Default Y Variable Boundary Values = Conservative
FFT Full Input Range = On
FFT Max = 0.0
FFT Min = 0.0
FFT Subtract Mean = Off
FFT Window Type = Hanning
FFT X Function = Frequency
FFT Y Function = Power Spectral Density
Histogram Automatic Divisions = Automatic
Histogram Divisions = -1.0,1.0
Histogram Divisions Count = 10
Histogram Y Axis Value = Count
Is FFT Chart = Off
Max X = 1.0
```

```
Max Y = 1.0
Min X = -1.0
Min Y = -1.0
Time Chart Keep Single Case = Off
Use Data For X Axis Labels = On
Use Data For Y Axis Labels = On
X Axis Automatic Range = On
X Axis Inverted = Off
X Axis Logarithmic Scaling = Off
Y Axis Automatic Range = On
Y Axis Inverted = Off
Y Axis Logarithmic Scaling = Off
  CHART SERIES:Series 1
  Chart Line Custom Data Selection = Off
  Chart Line Filename =
  Chart Series Type = Regular
  Chart X Variable = Chart Count
  Chart Y Variable = Density
  Histogram Y Axis Weighting = None
  Location = /POLYLINE:pl85
  Monitor Data Filename =
  Monitor Data Source = Case
  Monitor Data X Variable Absolute Value = Off
  Monitor Data Y Variable Absolute Value = Off
  Series Name = Series 1
  Time Chart Expression = Time
  Time Chart Type = Point
  Time Chart Variable = Density
  Time Chart X Expression = Time
  Time Variable Absolute Value = Off
  Time Variable Boundary Values = Conservative
  X Variable Absolute Value = Off
  X Variable Boundary Values = Conservative
  Y Variable Absolute Value = Off
  Y Variable Boundary Values = Conservative
    CHART LINE:Chart Line 1
    Auto Chart Line Colour = On
    Chart FFT Line Type = Bars
    Chart Line Colour = 1.0, 0.0, 0.0
    Chart Line Style = Automatic
    Chart Line Type = Lines
    Chart Line Visibility = On
    Chart Symbol Colour = 0.0, 1.0, 0.0
    Chart Symbol Style = None
    Fill Area = On
    Fill Area Options = Automatic
    Is Valid = True
    Line Name = Series 1
    Use Automatic Line Naming = On
    END
  END
```

```
  OBJECT REPORT OPTIONS:
    Report Caption =
  END
END""")
results1.SendCommand(Command="""EXPORT:
 Export File = C:/Users/User/Desktop/scripts/Results/85.csv
 Export Chart Name = pressure85
 Overwrite = On
END
>export chart""")
#

#Pressure data at the 90% span location
results1.SendCommand(Command="""CHART:pressure90
Chart Axes Font = Tahoma, 10, False, False, False, False
Chart Axes Titles Font = Tahoma, 10, True, False, False, False
Chart Grid Line Width = 1
Chart Horizontal Grid = On
Chart Legend = On
Chart Legend Font = Tahoma, 8, False, False, False, False
Chart Legend Inside = Outside Chart
Chart Legend Justification = Center
Chart Legend Position = Bottom
Chart Legend Width Height = 0.2 , 0.4
Chart Legend X Justification = Right
Chart Legend XY Position = 0.73 , 0.275
Chart Legend Y Justification = Center
Chart Line Width = 2
Chart Lines Order = Series 1,Chart Line 1
Chart Minor Grid = Off
Chart Minor Grid Line Width = 1
Chart Symbol Size = 4
Chart Title = Title
Chart Title Font = Tahoma, 12, True, False, False, False
Chart Title Visibility = On
Chart Type = XY
Chart Vertical Grid = On
Chart X Axis Automatic Number Formatting = On
Chart X Axis Label = X Axis <units>
Chart X Axis Number Format = %10.3e
Chart Y Axis Automatic Number Formatting = On
Chart Y Axis Label = Y Axis <units>
Chart Y Axis Number Format = %10.3e
Default Chart X Variable = X
Default Chart Y Variable = Pressure
Default Difference Line Calculation = From Points
Default Histogram Y Axis Weighting = None
Default Time Chart Variable = Pressure
Default Time Chart X Expression = Time
Default Time Variable Absolute Value = Off
Default Time Variable Boundary Values = Conservative
```

```
Default X Variable Absolute Value = Off
Default X Variable Boundary Values = Conservative
Default Y Variable Absolute Value = Off
Default Y Variable Boundary Values = Conservative
FFT Full Input Range = On
FFT Max = 0.0
FFT Min = 0.0
FFT Subtract Mean = Off
FFT Window Type = Hanning
FFT X Function = Frequency
FFT Y Function = Power Spectral Density
Histogram Automatic Divisions = Automatic
Histogram Divisions = -1.0,1.0
Histogram Divisions Count = 10
Histogram Y Axis Value = Count
Is FFT Chart = Off
Max X = 1.0
Max Y = 1.0
Min X = -1.0
Min Y = -1.0
Time Chart Keep Single Case = Off
Use Data For X Axis Labels = On
Use Data For Y Axis Labels = On
X Axis Automatic Range = On
X Axis Inverted = Off
X Axis Logarithmic Scaling = Off
Y Axis Automatic Range = On
Y Axis Inverted = Off
Y Axis Logarithmic Scaling = Off
  CHART SERIES:Series 1
  Chart Line Custom Data Selection = Off
  Chart Line Filename =
  Chart Series Type = Regular
  Chart X Variable = Chart Count
  Chart Y Variable = Density
  Histogram Y Axis Weighting = None
  Location = /POLYLINE:pl90
  Monitor Data Filename =
  Monitor Data Source = Case
  Monitor Data X Variable Absolute Value = Off
  Monitor Data Y Variable Absolute Value = Off
  Series Name = Series 1
  Time Chart Expression = Time
  Time Chart Type = Point
  Time Chart Variable = Density
  Time Chart X Expression = Time
  Time Variable Absolute Value = Off
  Time Variable Boundary Values = Conservative
  X Variable Absolute Value = Off
  X Variable Boundary Values = Conservative
  Y Variable Absolute Value = Off
```

```
  Y Variable Boundary Values = Conservative
    CHART LINE:Chart Line 1
    Auto Chart Line Colour = On
    Chart FFT Line Type = Bars
    Chart Line Colour = 1.0, 0.0, 0.0
    Chart Line Style = Automatic
    Chart Line Type = Lines
    Chart Line Visibility = On
    Chart Symbol Colour = 0.0, 1.0, 0.0
    Chart Symbol Style = None
    Fill Area = On
    Fill Area Options = Automatic
    Is Valid = True
    Line Name = Series 1
    Use Automatic Line Naming = On
    END
  END
  OBJECT REPORT OPTIONS:
      Report Caption =
  END
END""")
results1.SendCommand(Command="""EXPORT:
 Export File = C:/Users/User/Desktop/scripts/Results/90.csv
 Export Chart Name = pressure90
 Overwrite = On
END
>export chart""")
#

# Pressure data at the 95% pressure location
results1.SendCommand(Command="""CHART:pressure95
Chart Axes Font = Tahoma, 10, False, False, False, False
Chart Axes Titles Font = Tahoma, 10, True, False, False, False
Chart Grid Line Width = 1
Chart Horizontal Grid = On
Chart Legend = On
Chart Legend Font = Tahoma, 8, False, False, False, False
Chart Legend Inside = Outside Chart
Chart Legend Justification = Center
Chart Legend Position = Bottom
Chart Legend Width Height = 0.2 , 0.4
Chart Legend X Justification = Right
Chart Legend XY Position = 0.73 , 0.275
Chart Legend Y Justification = Center
Chart Line Width = 2
Chart Lines Order = Series 1,Chart Line 1
Chart Minor Grid = Off
Chart Minor Grid Line Width = 1
Chart Symbol Size = 4
Chart Title = Title
Chart Title Font = Tahoma, 12, True, False, False, False
```

```
Chart Title Visibility = On
Chart Type = XY
Chart Vertical Grid = On
Chart X Axis Automatic Number Formatting = On
Chart X Axis Label = X Axis <units>
Chart X Axis Number Format = %10.3e
Chart Y Axis Automatic Number Formatting = On
Chart Y Axis Label = Y Axis <units>
Chart Y Axis Number Format = %10.3e
Default Chart X Variable = X
Default Chart Y Variable = Pressure
Default Difference Line Calculation = From Points
Default Histogram Y Axis Weighting = None
Default Time Chart Variable = Pressure
Default Time Chart X Expression = Time
Default Time Variable Absolute Value = Off
Default Time Variable Boundary Values = Conservative
Default X Variable Absolute Value = Off
Default X Variable Boundary Values = Conservative
Default Y Variable Absolute Value = Off
Default Y Variable Boundary Values = Conservative
FFT Full Input Range = On
FFT Max = 0.0
FFT Min = 0.0
FFT Subtract Mean = Off
FFT Window Type = Hanning
FFT X Function = Frequency
FFT Y Function = Power Spectral Density
Histogram Automatic Divisions = Automatic
Histogram Divisions = -1.0,1.0
Histogram Divisions Count = 10
Histogram Y Axis Value = Count
Is FFT Chart = Off
Max X = 1.0
Max Y = 1.0
Min X = -1.0
Min Y = -1.0
Time Chart Keep Single Case = Off
Use Data For X Axis Labels = On
Use Data For Y Axis Labels = On
X Axis Automatic Range = On
X Axis Inverted = Off
X Axis Logarithmic Scaling = Off
Y Axis Automatic Range = On
Y Axis Inverted = Off
Y Axis Logarithmic Scaling = Off
  CHART SERIES:Series 1
  Chart Line Custom Data Selection = Off
  Chart Line Filename =
  Chart Series Type = Regular
  Chart X Variable = Chart Count
```

```
  Chart Y Variable = Density
  Histogram Y Axis Weighting = None
  Location = /POLYLINE:pl95
  Monitor Data Filename =
  Monitor Data Source = Case
  Monitor Data X Variable Absolute Value = Off
  Monitor Data Y Variable Absolute Value = Off
  Series Name = Series 1
  Time Chart Expression = Time
  Time Chart Type = Point
  Time Chart Variable = Density
  Time Chart X Expression = Time
  Time Variable Absolute Value = Off
  Time Variable Boundary Values = Conservative
  X Variable Absolute Value = Off
  X Variable Boundary Values = Conservative
  Y Variable Absolute Value = Off
  Y Variable Boundary Values = Conservative
    CHART LINE:Chart Line 1
    Auto Chart Line Colour = On
    Chart FFT Line Type = Bars
    Chart Line Colour = 1.0, 0.0, 0.0
    Chart Line Style = Automatic
    Chart Line Type = Lines
    Chart Line Visibility = On
    Chart Symbol Colour = 0.0, 1.0, 0.0
    Chart Symbol Style = None
    Fill Area = On
    Fill Area Options = Automatic
    Is Valid = True
    Line Name = Series 1
    Use Automatic Line Naming = On
    END
  END
  OBJECT REPORT OPTIONS:
      Report Caption =
  END
END""")
results1.SendCommand(Command="""EXPORT:
 Export File = C:/Users/User/Desktop/scripts/Results/95.csv
 Export Chart Name = pressure95
 Overwrite = On
END
>export chart""")
```

“meshgen.js” – Java Script file responsible for meshing of the model

```
var Mesh_Mod=DS.Tree.FirstActiveBranch.MeshControlGroup

DS.Script.SelectItems(""+Mesh_Mod.ID)
DS.Script.SelectItems("")
DS.Script.doGraphicsPickGeometry(1)
ListView.ActivateItem("Physics Preference")
ListView.ItemValue = "CFD"

//Mesh Sizing of the domain
DS.Script.SelectItems(""+Mesh_Mod.ID)
DS.Script.SelectItems("")
DS.Script.doInsertMeshSize(1)
DS.Script.doEditSelectAll(1)
ListView.ActivateItem("Scoping Method")
ListView.ItemValue="Geometry Selection"
ListView.ActivateItem("Geometry")
ListView.ItemValue="Apply"
ListView.ActivateItem("Size Function")
ListView.ItemValue = "Proximity and Curvature"
ListView.ActivateItem("Element Size")
ListView.ItemValue = "1,2"

//Mesh Sizing of the blade
DS.Script.SelectItems(""+Mesh_Mod.ID)
DS.Script.SelectItems("")
DS.Script.doInsertMeshSize(1)
DS.Script.doGraphicsSurfaceSelect(1)
ListView.ActivateItem("Scoping Method")
ListView.ItemValue = "Named Selection"
ListView.ActivateItem("Named Selection")
ListView.ItemValue = "NS_blade"
ListView.ActivateItem("Type")
ListView.ItemValue = "Element Size"
ListView.ActivateItem("Element Size")
ListView.ItemValue = "0,0065"
ListView.ActivateItem("Size Function")
ListView.ItemValue = "Curvature"


//Inflation layer around blade
DS.Script.SelectItems(""+Mesh_Mod.ID)
DS.Script.SelectItems("")
DS.Script.doInsertInflation(1)
DS.Script.doEditSelectAll(1)
ListView.ActivateItem("Scoping Method")
ListView.ItemValue="Geometry Selection"
ListView.ActivateItem("Geometry")
ListView.ItemValue="Apply"
```

```
ListView.ActivateItem("Boundary Scoping Method")
ListView.ItemValue="Named Selections"
ListView.ActivateItem("Boundary")
ListView.ItemValue="NS_blade"
ListView.ActivateItem("Inflation Option")
ListView.ItemValue="First Layer Thickness"
ListView.ActivateItem("First Layer Height")
ListView.ItemValue= "0,00001"
ListView.ActivateItem("Maximum Layers")
ListView.ItemValue= "40"
ListView.ActivateItem("Growth Rate")
ListView.ItemValue= "1,2"

//Match control for periodic interface
DS.Script.SelectItems(""+Mesh_Mod.ID)
DS.Script.SelectItems("")
DS.Script.doInsertMeshMatchControl(1)
DS.Script.doGraphicsSurfaceSelect(1)
ListView.ActivateItem("Scoping Method")
ListView.ItemValue="Named Selection"
ListView.ActivateItem("High Boundary")
ListView.ItemValue="NS_interface_1"
ListView.ActivateItem("Low Boundary")
ListView.ItemValue="NS_interface_2"
ListView.ActivateItem("Axis of Rotation")
ListView.ItemValue="Global Coordinate System"

//Update Mesh
DS.Script.doModelPreviewMeshFromToolbar(1)
```

“setup specs.jou” – journal file that setups mathematical model

```
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Setup|General"))
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Setup|General"))
(cx-gui-do cx-activate-item "NavigationPane*List_Tree1")
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Setup|General"))
(cx-gui-do cx-activate-item "General*Table1*ButtonBox1(Mesh)*PushButton3(Check)")
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Setup|Models|Viscous (Laminar)"))
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Setup|Models|Viscous (Laminar)"))
(cx-gui-do cx-activate-item "NavigationPane*List_Tree1")
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Setup|Models|Viscous (Laminar)"))
(cx-gui-do cx-set-toggle-button2 "Viscous Model*Table1*ToggleBox1(Model)*k-epsilon (2 eqn)" #t)
(cx-gui-do cx-activate-item "Viscous Model*Table1*ToggleBox1(Model)*k-epsilon (2 eqn)")
(cx-gui-do cx-set-toggle-button2 "Viscous Model*Table1*ToggleBox6(k-epsilon Model)*Realizable" #t)
(cx-gui-do cx-activate-item "Viscous Model*Table1*ToggleBox6(k-epsilon Model)*Realizable")
(cx-gui-do cx-set-toggle-button2 "Viscous Model*Table1*ToggleBox17(Near-Wall Treatment)*Menter-Lechner" #t)
```

```
(cx-gui-do cx-activate-item "Viscous Model*Table1*ToggleBox17(Near-Wall Treatment)*Menter-
Lechner")
(cx-gui-do cx-set-toggle-button2 "Viscous
Model*Table1*ToggleBox18(Options)*CheckButton4(Curvature Correction)" #t)
(cx-gui-do cx-activate-item "Viscous Model*Table1*ToggleBox18(Options)*CheckButton4(Curvature
Correction)")
(cx-gui-do cx-activate-item "Viscous Model*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Setup|Models|Viscous
(Realizable k-e, NWT Menter-Lechner)"))
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Setup|Cell Zone
Conditions|nrelphasevipitch3degdomain (fluid, id=2)"))
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Setup|Cell Zone
Conditions|nrelphasevipitch3degdomain (fluid, id=2)"))
(cx-gui-do cx-activate-item "NavigationPane*List_Tree1")
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Setup|Cell Zone
Conditions|nrelphasevipitch3degdomain (fluid, id=2)"))
(cx-gui-do cx-activate-item "Ribbon*Frame1*Frame4(User Defined)*Table1*Table3(Field
Functions)*PushButton2(Units)")
(cx-gui-do cx-set-list-selections "Set Units*List1(Quantities)" '( 2))
(cx-gui-do cx-activate-item "Set Units*List1(Quantities)")
(cx-gui-do cx-set-list-selections "Set Units*Frame3*List1(Units)" '( 2))
(cx-gui-do cx-activate-item "Set Units*Frame3*List1(Units)")
(cx-gui-do cx-activate-item "Set Units*PanelButtons*PushButton2(Cancel)")
(cx-gui-do cx-set-toggle-button2 "Fluid*Table3*CheckButton1(Frame Motion)" #t)
(cx-gui-do cx-activate-item "Fluid*Table3*CheckButton1(Frame Motion)")
(cx-gui-do cx-set-real-entry-list "Fluid*Table4*Frame1*Frame1(Reference
Frame)*Table1*Table1*Table4*Table1*Table1(Rotational Velocity)*Table1*RealEntry2(Speed)" '(
7.539826))
(cx-gui-do cx-activate-item "Fluid*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Setup|Boundary Conditions"))
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Setup|Boundary Conditions"))
(cx-gui-do cx-activate-item "NavigationPane*List_Tree1")
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Setup|Boundary Conditions"))
(cx-gui-do cx-set-list-selections "Boundary Conditions*Table1*List2(Zone)" '( 2))
(cx-gui-do cx-activate-item "Boundary Conditions*Table1*List2(Zone)")
(cx-gui-do cx-activate-item "Boundary
Conditions*Table1*Table3*Table4*ButtonBox1*PushButton1(Edit)")
(cx-gui-do cx-set-list-selections "Velocity
Inlet*Frame3*Frame1(Momentum)*Table1*DropDownList6(Velocity Specification Method)" '( 1))
(cx-gui-do cx-activate-item "Velocity
Inlet*Frame3*Frame1(Momentum)*Table1*DropDownList6(Velocity Specification Method)")
(cx-gui-do cx-set-real-entry-list "Velocity
Inlet*Frame3*Frame1(Momentum)*Table1*Table18*RealEntry2(Z-Velocity)" '( -10))
(cx-gui-do cx-activate-item "Velocity Inlet*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-set-list-selections "Boundary Conditions*Table1*List2(Zone)" '( 3))
(cx-gui-do cx-activate-item "Boundary Conditions*Table1*List2(Zone)")
(cx-gui-do cx-activate-item "Boundary
Conditions*Table1*Table3*Table4*ButtonBox1*PushButton1(Edit)")
(cx-gui-do cx-activate-item "Interface*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Setup|Mesh Interfaces"))
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Setup|Mesh Interfaces"))
```

```
(cx-gui-do cx-activate-item "NavigationPane*List_Tree1")
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Setup|Mesh Interfaces"))
(cx-gui-do cx-activate-item "Mesh Interfaces*Table1*Table2(Unassigned Interface
Zones)*Table2*Table2*PushButton2( Manual Create)")
(cx-gui-do cx-set-text-entry "Create/Edit Mesh Interfaces *Table1*TextEntry4(Mesh Interface)"
"periodic-interface")
(cx-gui-do cx-set-toggle-button2 "Create/Edit Mesh Interfaces *Table1*Frame7(Interface
Options)*CheckButton1(Periodic Boundary Condition)" #t)
(cx-gui-do cx-activate-item "Create/Edit Mesh Interfaces *Table1*Frame7(Interface
Options)*CheckButton1(Periodic Boundary Condition)")
(cx-gui-do cx-set-toggle-button2 "Create/Edit Mesh Interfaces *Table1*Frame7(Interface
Options)*CheckButton4(Matching)" #t)
(cx-gui-do cx-activate-item "Create/Edit Mesh Interfaces *Table1*Frame7(Interface
Options)*CheckButton4(Matching)")
(cx-gui-do cx-set-list-selections "Create/Edit Mesh Interfaces *Table1*List2" '( 0))
(cx-gui-do cx-activate-item "Create/Edit Mesh Interfaces *Table1*List2")
(cx-gui-do cx-set-list-selections "Create/Edit Mesh Interfaces *Table1*List3" '( 1))
(cx-gui-do cx-activate-item "Create/Edit Mesh Interfaces *Table1*List3")
(cx-gui-do cx-set-toggle-button2 "Create/Edit Mesh Interfaces *Table1*Frame8(Periodic Boundary
Condition)*Frame1*ButtonBox1(Type)*Rotational" #t)
(cx-gui-do cx-activate-item "Create/Edit Mesh Interfaces *Table1*Frame8(Periodic Boundary
Condition)*Frame1*ButtonBox1(Type)*Rotational")
(cx-gui-do cx-set-toggle-button2 "Create/Edit Mesh Interfaces *Table1*Frame8(Periodic Boundary
Condition)*Frame1*CheckButton3(Auto Compute Offset)" #f)
(cx-gui-do cx-activate-item "Create/Edit Mesh Interfaces *Table1*Frame8(Periodic Boundary
Condition)*Frame1*CheckButton3(Auto Compute Offset)")
(cx-gui-do cx-set-real-entry-list "Create/Edit Mesh Interfaces *Table1*Frame8(Periodic Boundary
Condition)*Frame1*Table2(Offset)*RealEntry4(Angle)" '( 3.141592))
(cx-gui-do cx-activate-item "Create/Edit Mesh Interfaces *PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item "Create/Edit Mesh Interfaces *PanelButtons*PushButton2(Cancel)")
(cx-gui-do cx-activate-item "Mesh Interfaces*PanelButtons*PushButton2(Cancel)")
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Solution|Methods"))
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Solution|Methods"))
(cx-gui-do cx-activate-item "NavigationPane*List_Tree1")
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Solution|Methods"))
(cx-gui-do cx-set-list-selections "Solution Methods*Table1*Table2(Pressure-Velocity
Coupling)*DropDownList2(Scheme)" '( 3))
(cx-gui-do cx-activate-item "Solution Methods*Table1*Table2(Pressure-Velocity
Coupling)*DropDownList2(Scheme)")
(cx-gui-do cx-set-list-selections "Solution Methods*Table1*Table3(Spatial
Discretization)*DropDownList4(Turbulent Kinetic Energy)" '( 1))
(cx-gui-do cx-activate-item "Solution Methods*Table1*Table3(Spatial
Discretization)*DropDownList4(Turbulent Kinetic Energy)")
(cx-gui-do cx-set-toggle-button2 "Solution Methods*Table1*CheckButton5(Pseudo Transient)" #t)
(cx-gui-do cx-activate-item "Solution Methods*Table1*CheckButton5(Pseudo Transient)")
(cx-gui-do cx-set-toggle-button2 "Solution Methods*Table1*Table7*CheckButton1(High Order Term
Relaxation)" #t)
(cx-gui-do cx-activate-item "Solution Methods*Table1*Table7*CheckButton1(High Order Term
Relaxation)")
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Solution|Monitors|Residual"))
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Solution|Monitors|Residual"))
```

```
(cx-gui-do cx-activate-item "NavigationPane*List_Tree1")
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Solution|Monitors|Residual"))
(cx-gui-do cx-set-real-entry-list "Residual
Monitors*Table1*Table2*Table1*Table1(Equations)*RealEntry11" '( 1e-06))
(cx-gui-do cx-set-real-entry-list "Residual
Monitors*Table1*Table2*Table1*Table1(Equations)*RealEntry17" '( 1e-06))
(cx-gui-do cx-set-real-entry-list "Residual
Monitors*Table1*Table2*Table1*Table1(Equations)*RealEntry23" '( 1e-06))
(cx-gui-do cx-set-real-entry-list "Residual
Monitors*Table1*Table2*Table1*Table1(Equations)*RealEntry29" '( 1e-06))
(cx-gui-do cx-set-real-entry-list "Residual
Monitors*Table1*Table2*Table1*Table1(Equations)*RealEntry35" '( 1e-06))
(cx-gui-do cx-set-real-entry-list "Residual
Monitors*Table1*Table2*Table1*Table1(Equations)*RealEntry41" '( 1e-06))
(cx-gui-do cx-activate-item "Residual Monitors*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Setup|General"))
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Setup|General"))
(cx-gui-do cx-activate-item "NavigationPane*List_Tree1")
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Setup|General"))
(cx-gui-do cx-activate-item "General*Table1*ButtonBox1(Mesh)*PushButton3(Check)")
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Solution|Initialization"))
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Solution|Initialization"))
(cx-gui-do cx-activate-item "NavigationPane*List_Tree1")
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Solution|Initialization"))
(cx-gui-do cx-set-toggle-button2 "Solution Initialization*Table1*ToggleBox3(Initialization
Methods)*Standard Initialization" #t)
(cx-gui-do cx-activate-item "Solution Initialization*Table1*ToggleBox3(Initialization
Methods)*Standard Initialization")
(cx-gui-do cx-set-list-selections "Solution Initialization*Table1*DropDownList1(Compute from)" '( 1))
(cx-gui-do cx-activate-item "Solution Initialization*Table1*DropDownList1(Compute from)")
(cx-gui-do cx-set-toggle-button2 "Solution Initialization*Table1*ToggleBox2(Reference
Frame)*Absolute" #t)
(cx-gui-do cx-activate-item "Solution Initialization*Table1*ToggleBox2(Reference Frame)*Absolute")
(cx-gui-do cx-activate-item "Solution Initialization*Table1*ButtonBox8*PushButton1(Initialize)")
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Solution|Run Calculation"))
(cx-gui-do cx-activate-item "NavigationPane*List_Tree1")
(cx-gui-do cx-set-list-tree-selections "NavigationPane*List_Tree1" (list "Solution|Run Calculation"))
(cx-gui-do cx-set-integer-entry "Run Calculation*Table1*IntegerEntry10(Number of Iterations)" 1500)
(cx-gui-do cx-activate-item "Run Calculation*Table1*IntegerEntry10(Number of Iterations)")
(cx-gui-do cx-activate-item "Run Calculation*Table1*PushButton22(Calculate)")
(cx-gui-do cx-activate-item "Information*OK")
(cx-gui-do cx-activate-item "MenuBar*FileMenu*Close Fluent")
```