

2010

Simulations of Flow Over Wind Turbines

Dnyanesh A. Digraskar
University of Massachusetts Amherst

Follow this and additional works at: <https://scholarworks.umass.edu/theses>



Part of the [Aerodynamics and Fluid Mechanics Commons](#), and the [Energy Systems Commons](#)

Digraskar, Dnyanesh A., "Simulations of Flow Over Wind Turbines" (2010). *Masters Theses 1911 - February 2014*. 417.
Retrieved from <https://scholarworks.umass.edu/theses/417>

This thesis is brought to you for free and open access by ScholarWorks@UMass Amherst. It has been accepted for inclusion in Masters Theses 1911 - February 2014 by an authorized administrator of ScholarWorks@UMass Amherst. For more information, please contact scholarworks@library.umass.edu.

SIMULATIONS OF FLOW OVER WIND TURBINES

A Thesis Presented

by

DNYANESH A. DIGRASKAR

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE

May 2010

Mechanical and Industrial Engineering

© Copyright by Dnyanesh A. Digraskar 2010

All Rights Reserved

SIMULATIONS OF FLOW OVER WIND TURBINES

A Thesis Presented

by

DNYANESH A. DIGRASKAR

Approved as to style and content by:

J. Blair Perot, Chair

Matthew A. Lackner, Member

Jon G. McGowan, Member

Donald L. Fisher, Department Head
Mechanical and Industrial Engineering

ACKNOWLEDGMENTS

I wish to thank my thesis advisor Dr. Blair Perot for all the guidance, advice and support that he has provided during this research. I am especially thankful for his valuable insights, and all his endurance in trying to make me a better researcher. I am also thankful to Dr. Jon McGowan and Dr. Matthew Lackner, for their advice and support during the last one year. I speacially thank them for their help regarding wind engineering concepts. I specially wish to thank Sandeep Menon, Shivasubramanian Gopalakrishnan, Michael B. Martell Jr., and Kshitij Neroorkar for their valuable discussions and for providing me help whenever needed. I would also like to thank Timothy McGuinness, Ali Khajeh Saeed, Nat Trask, Kyle Mooney and Raghuv eer Vajapeyayajula for all help and fun during work. I thank my friends and roommates for making my stay at University of Massachusetts, Amherst enjoyable and memorable.

Lastly, it is impossible to describe in words the love, encouragement and support that I received from my family.

ABSTRACT

SIMULATIONS OF FLOW OVER WIND TURBINES

MAY 2010

DNYANESH A. DIGRASKAR

B.E., UNIVERSITY OF MUMBAI, INDIA

M.S., UNIVERSITY OF MASSACHUSETTS, AMHERST

Directed by: Professor J. Blair Perot

One of the most abundant sources of renewable energy is wind. Today, a considerable amount of resources are being utilized for research on harnessing the wind energy efficiently. Out of all the factors responsible for efficient energy production, the aerodynamics of flow around the wind turbine blades play an important role. This work aims to undertake aerodynamic analysis of a Horizontal Axis Wind Turbine. A steady state, incompressible flow solver for multiple reference frames, MRFSimpleFoam is modified and used for performing simulations of flow over National Renewable Energy Laboratory Phase VI wind turbine rotor. The code is first tested on a locally modeled wind turbine blade and is then validated by using the actual NREL rotor. The flow behavior is studied and a comparison of results from the simulations and the experimental wind tunnel data is presented. The ability of Computational Fluid Dynamics (CFD) techniques in simulating wind flow over entire wind turbine assembly is also displayed by carrying out moving mesh simulations of a full wind turbine.

TABLE OF CONTENTS

	Page
ACKNOWLEDGMENTS	iv
ABSTRACT	v
LIST OF TABLES	ix
LIST OF FIGURES	x
 CHAPTER	
1. INTRODUCTION	1
1.1 Motivation	1
1.2 Background	3
1.3 Goals and Objectives	7
2. METHODOLOGY	10
2.1 Pre processing and Post processing	10
2.1.1 Mesh Generation	10
2.1.2 Domain Decomposition	14
2.1.3 Post processing	14
2.2 Code Formulation - MRFSimpleFoam	16
2.2.1 RANS modeling	17
2.2.2 Multiple Reference Frames	18
2.2.2.1 Working of Code on Zones	19
2.3 Code Formulation - TurbDyMFoam	20
2.3.1 The Generalized Grid Interface (GGI)	21
2.4 Boundary and Initial Conditions, and Turbulence Model	23

2.4.1	Boundary and Initial Conditions	23
2.4.2	Turbulence Model	25
3.	PRELIMINARY RESULTS	27
3.1	Flow Past a Rotating Tube	27
3.2	Flow Over a Flat Plate	30
3.3	Preliminary Code Validation for the Delft University (DU) Wind Turbine blade	35
3.3.1	Background	35
3.3.2	Results	37
4.	FINAL RESULTS	42
4.1	Pre processing	42
4.1.1	3D Modeling of the NREL Blade	42
4.1.2	Meshing	44
4.1.2.1	Meshes for MRFSimpleFoam Cases	44
4.1.2.2	Meshes for turbDyMFoam Cases	45
4.2	Results	48
4.2.1	CFD Simulations of the NREL Phase VI Experiment	49
4.2.2	Wind Flow Simulations over Full Wind Turbine	56
4.2.2.1	No Yaw Case	56
4.2.2.2	15 degree Yawed Rotor	60
4.2.3	Simulations with Shear in the Rotor Plane	61
5.	CONCLUSIONS AND FUTURE WORK	66
5.1	Summary and Conclusions	66
5.2	Future Work	67
 APPENDICES		
A.	THE ALGORITHM FOR SIMPLE LOOP	69
B.	MULTIPLE REFERENCE FRAMES: TOOLS AND UTILITIES	70
C.	IMPLEMENTATION OF GENERALIZED GRID INTERFACE (GGI) IN THE CFD SOLVER	71

D. S809 AIRFOIL DATA FOR THE DESIGN OF NREL PHASE VI EXPERIMENT WIND TURBINE BLADE.	73
E. CO-EFFICIENT OF PRESSURE PLOTS FOR DIFFERENT INLET WIND SPPEDES AT VARIOUS BLADE SECTIONS.	80
BIBLIOGRAPHY	84

LIST OF TABLES

Table	Page
2.1 List of some post-processing utilities used in the simulations.	15
2.2 Boundary conditions used and their operation.	24
2.3 Boundary conditions for $\tilde{\nu}$	26
D.1 Airfoil Co-ordinates.	74
D.2 Blade Chord and Twist Configuration.	75

LIST OF FIGURES

Figure	Page
1.1 NREL UAE wind turbine mounted in NASA Ames 80 × 120 foot wind tunnel [32]	4
1.2 Different blade tip geometries marked with streamlines, with leading edge of the blade on top [9].	6
2.1 A typical 2D structured grid consisting of rectangular elements.	11
2.2 A 3D unstructured grid around a finite cylinder consisting of tetrahedral elements.	12
2.3 Unstructured mesh consisting of 2 million tetrahedral elements around a wind turbine rotor.	13
2.4 Surface mesh showing triangular elements on a wind turbine blade.	13
2.5 Mesh around wind turbine rotor is decomposed into 24 smaller meshes using a METIS partitioning method. Colored by processor Id.	15
2.6 Co ordinate system representation of two reference frames.	18
2.7 Schematic representation of multiple reference frames. Fluid in the inertial frame and the wind turbine rotor, and domain in the rotating frame.	19
2.8 Representation of GGI patches in two non conformal meshes. GGI Inside for the rotor mesh and the GGI Outside for the stator mesh.	22
2.9 Schematic representation of periodic boundary conditions.	25
3.1 Case setup with a small tube aligned with an offset to the flow axis. Offset is selected so as to visualize the wake clearly.	28
3.2 Slice through the domain representing the grid resolution. Fine mesh around the cylinder is visible.	29

3.3	Various slices in the domain colored by velocity magnitude. Wake rotation is clearly visible.	29
3.4	Case setup with a flat plate at 4° angle of attack to the incoming flow.	30
3.5	Slice through the domain representing the grid resolution. A very fine mesh near the plate is seen.	32
3.6	Slice of stream wise vorticity at a distance $10 \times$ chord length from the plate.	32
3.7	Iso-surface plot of stream-wise vorticity.	33
3.8	Slice of streamwise vorticity at a distance $50 \times$ chord length from the plate.	34
3.9	Variation of streamwise vorticity with the mesh size at position 30c and 50c downstream.	34
3.10	Surface mesh on the wind turbine rotor.	36
3.11	Wedge shaped domain with periodic boundaries.	36
3.12	Power produced by the turbine blade for different values of wind speeds.	37
3.13	Wake behind the wind turbine rotor. It is colored by stream-wise velocity contours.	38
3.14	Slice across the domain representing the tip vortices.	39
3.15	Slice across the domain downstream of the rotor, very close to the outlet. It represents the vorticity wake.	40
3.16	Iso-surfaces of stream-wise vorticity representing vortices off the blade tips.	40
3.17	Side view of the iso-surface plot of stream-wise vorticity. It represents wake after 2 rotations.	41
4.1	3 dimensional model of the NREL Phase VI experiment wind turbine blade.	43

4.2	3 dimensional setup of periodic domain used for simulations over the NREL blade.	45
4.3	Slice across the 3D computational domain representing mesh around the blade.	46
4.4	3 dimensional setup of full wind turbine simulations case. Domain is modeled with wind turbine rotor, hub, tower, and ground.	47
4.5	Full wind turbine assembly meshed with tetrahedral elements.	47
4.6	Tetrahedral mesh around the 3D wind turbine model with the rotor yawed at an angle of 15 degrees.	48
4.7	Numerical setup along with boundary conditions for the NREL Phase VI simulations.	50
4.8	Comparison of power values between the CFD values and the NREL experimental values.	51
4.9	Variation of the rotor power with the variation of inlet eddy viscosity (changing turbulence parameters).	51
4.10	Co-efficient of pressure at 30%R blade location for 13 m/s inlet wind speed.	53
4.11	Co-efficient of pressure at 50%R blade location for 13 m/s inlet wind speed.	54
4.12	Co-efficient of pressure at 95%R blade location for 13 m/s inlet wind speed.	54
4.13	Comparison of power values obtained by MRFSimpleFoam and turbDymFoam simulations.	55
4.14	Position of wind turbine rotor at 2 different time steps indicating a correct implementation of the moving mesh algorithm in the solver.	58
4.15	Stream wise velocity in the computational domain. $U_{\infty} = 15m/s$	59
4.16	Iso surface plot of stream wise vorticity colored by velocity magnitude representing the wake off the rotor blades. $U_{\infty} = 15m/s$	59

4.17	Contours of streamwise velocity for 15° yaw. Effect of the rotor yaw is visible in the wake downstream. $U_\infty = 15m/s$.	60
4.18	Iso-surfaces of vorticity colored by stream-wise velocity for 15° yaw. $U_\infty = 15m/s$.	61
4.19	Iso-surfaces of Q colored by stream-wise velocity for 15° yaw. $U_\infty = 15m/s$.	62
4.20	Schematic representation of simulation of flow over wind turbine with a shear profile.	63
4.21	shearVelocity boundary condition applied to the inlet of the domain. Velocity contours in the domain. $U_\infty = 15m/s$.	64
D.1	S809 airfoil after plotting the x and y co-ordinates from table D.1.	76
E.1	Co-efficient of pressure at 30%R blade location for 15 m/s inlet wind speed.	80
E.2	Co-efficient of pressure at 50%r blade location for 15 m/s inlet wind speed.	81
E.3	Co-efficient of pressure at 95%r blade location for 15 m/s inlet wind speed.	81
E.4	Co-efficient of pressure at 30%R blade location for 20 m/s inlet wind speed.	81
E.5	Co-efficient of pressure at 50%R blade location for 20 m/s inlet wind speed.	82
E.6	Co-efficient of pressure at 95%R blade location for 20 m/s inlet wind speed.	82
E.7	Co-efficient of pressure at 30%R blade location for 25 m/s inlet wind speed.	82
E.8	Co-efficient of pressure at 50%R blade location for 25 m/s inlet wind speed.	83
E.9	Co-efficient of pressure at 95%R blade location for 25 m/s inlet wind speed.	83

CHAPTER 1

INTRODUCTION

1.1 Motivation

The rapidly depleting energy resources all over the world are a major cause of concern for mankind. These energy resources primarily the fossil fuels are also responsible for the fast global climate change. The Energy Information Administration estimates that about 86 percent of the world's total energy production results from the burning of fossil fuels [40]. Fossil fuels take millions of years to form, but in order to cater to the growing energy needs of the world, these reserves are being utilized at a much faster rate than they were created. Such is the rate of consumption of fossil fuels that the time seems near when these valuable sources of energy will be depleted. A closer look at some numbers will be helpful in understanding this problem. Estimates from the U.S. Dept. of Energy predict that the years of production left in the ground for oil is 43 years, gas 167 years, and coal 417 years [6]. This has led to the growing demand of renewable sources of energy production. Another major concern over the use of fossil fuels is the emission of tremendous amounts of harmful gases into the atmosphere. Burning of fossil fuels results a net increase of 10.65 billion tonnes of atmospheric carbon monoxide every year [41] which deteriorates the environmental balance. Therefore a need for clean and renewable sources of energy has been felt.

Large amounts of research and resources are being spent today in order to harness the energy from wind effectively. Wind energy as a source of energy is being favored widely as an alternative to fossil fuels, as it is plentiful, renewable, widely distributed, clean, and produces no greenhouse gas emissions. Wind turbines convert kinetic en-

ergy from the wind into mechanical energy which can be used to generate electricity. They are clean and inexpensive devices. Largest devices to date are capable of delivering upto 6MW power. However, the air flow environment in which wind turbines operate presents them with numerous challenges. Wind turbines experience large changes in angle of attack, atmospheric turbulence, wakes from the previous turbine in a wind farm, tower wakes, etc. Therefore, whenever a wind turbine is designed or a wind farm is being setup, all the above factors must be considered.

The need for accurate aerodynamic predictions associated with wind turbines is very important. Computational Fluid Dynamics (CFD) analysis of wind turbines can be performed at a cost less than that is required to set up a wind tunnel and perform full scale experiments. There has been a growing use of CFD codes over the last decade for wind turbine performance analysis. With advancements in science and technology, very powerful computers with high memory capacity are now available. It is now possible to simulate the average flow past a wind turbine using CFD codes and supercomputers. These codes employ a variety of turbulence models and thus make it possible to simulate atmospheric turbulence effects on wind turbines. But even then, it is still a challenge to use CFD codes to account for all the complicated flow phenomena associated with wind turbine aerodynamics. The aerodynamics of Horizontal Axis Wind Turbines (HAWT) are not straightforward. The airflow at the turbine hub is different than the airflow at the tip of the blades and further downstream the turbine. Thus, the aerodynamics of a wind turbine at the rotor surface exhibit a very different phenomenon which is rarely seen in other aerodynamic fields. Hence, motivated by all the above factors, this current research is carried out. This project intends to understand the unsteady aerodynamics associated with wind turbines by performing CFD simulations of the flow past a wind turbine.

1.2 Background

Advances in airfoil analysis, rotor development, power development, material technology and stress analysis have made wind turbines suitable alternatives to fossil fuels in terms of cost incurred per kilo-Watt hour (kW-hr). The aerodynamic effects on the wind turbines are mostly known but the details of flow behavior are still very poorly understood. There have been numerous studies carried out in this area over a long time. This section aims to highlight the previous work done in the field concerning this project. Unsteady behavior of flow associated with wind turbines is described in NREL report by Robinson et al. [29]. Accurately predicting the unsteady nature of flow around the wind turbines is a challenge. Aerodynamic predictions of wind turbines have been achieved in the past using a variety of methods. A relatively simple model, the Blade Element Method (BEM), is still quite commonly used for this purpose. These models are computationally undemanding and simple to implement [14]. On the other hand are the CFD models, which are more sophisticated and computationally much more expensive. CFD methods are based on numerically solving the Reynolds Averaged Navier-Stokes (RANS) equations with a accompanying turbulence model [7]. In the last decade or so, there has been growing research in the area of CFD applications to predict the flow behavior of wind turbines.

Full RANS simulations of a wind turbine rotor have been performed previously and researchers were successfully able to match the obtained CFD results to the experimental data. The NREL Phase VI test refer to a full scale Unsteady Aerodynamic Experiment on a two bladed wind turbine in the NASA Ames 80×120 foot wind tunnel [10]. The experimental setup of the NREL Phase VI rotor in NASA Ames wind tunnel is shown in the figure 1.1 below. Results obtained from this experiment were then considered as a benchmark for performing numerical studies.

Work done by Duque et al., Sorensen et al., Xu and Sankar has been cited frequently, when referring to CFD simulation of wind turbines. Duque et al. [7] per-



Figure 1.1. NREL UAE wind turbine mounted in NASA Ames 80×120 foot wind tunnel [32]

formed RANS computations on the two bladed NREL Phase VI rotor made up of S809 airfoil. In their work two commercial CFD codes, CAMRAD II and OVERFLOW-D were used for carrying out the simulations. CAMRAD II is a lifting line code modified for wind turbine modeling which works on 2D airfoil lift and drag data while OVERFLOW-D is a code which solves the RANS equations. They used the one equation Baldwin and Barth turbulence model in their code. It was observed in their work, that the RANS solutions matched the experimental results well. Aerodynamic performance characteristics like power generation, force coefficients, sectional pressure distribution were computed and matched with the benchmark values. Sorensen et al. [36] performed simulations of several cases from the NREL/NASA wind tunnel tests using an incompressible RANS code. Sectional force and pressure distributions

are calculated at varying wind speeds. After obtaining good agreement with the experimental results, they suggest that 3D CFD computations can be used to extract information about aerodynamic effects. Xu and Sankar [47] performed RANS simulations over wind turbines using a viscous flow model which simulated 3D unsteady viscous flow over the turbine rotor. They compared these CFD results with the NREL experimental Phase II (untwisted blade) and Phase III (twisted blade) results and hence validated their model. Gonzalez and Munduate [11] performed aerodynamic analysis of parked and rotating configurations of the NREL Phase VI blade. They studied properties like attached flow, separated flow, and stall by performing 2D sectional analysis of the blades. Specifically the leading edge and trailing edge separation were highlighted in their work. Similar results for the parked and rotating configuration of NREL Phase VI blades were also obtained by Schmitz and Chattot [31].

The availability of NREL experimental data also motivated many researchers to test the performance of commercially available CFD codes. Tangler [39] tested various versions of BEM based codes. Although, it was clear that BEM based codes lacked the accuracy of RANS solutions, reasonable quality of aerodynamic predictions was obtained. Laino [20] validated the AERODYN code by performing 2D analysis of the S809 airfoil and comparing the results with the NREL data.

CFD codes have also been used to study the rotor, tower wake shedding, and the role played by the blade tips in determining the performance of the machine. Wang Jian et al. [42] simulated a wind turbine blade with a tip vane using FLUENT and calculated the corresponding blade pressure distribution. With the help of these CFD solutions, they concluded that a blade with a tip vane has more pressure difference between the suction and the pressure side as compared to the usual blade. Ferrer and Munduate [9] worked specifically on wind turbine blade tips. They analyzed complex flow physics associated with the blade tips using a RANS CFD solver for

three different wind turbine blade tip geometries. The effects of varying wind speeds, and Reynold's number, for different angles of attack of the blade tips, were highlighted in their work. A considerable change in the pressure distributions depending on the blade tip geometries was noticed. Different blade tip geometries are shown in figure 1.2 below.

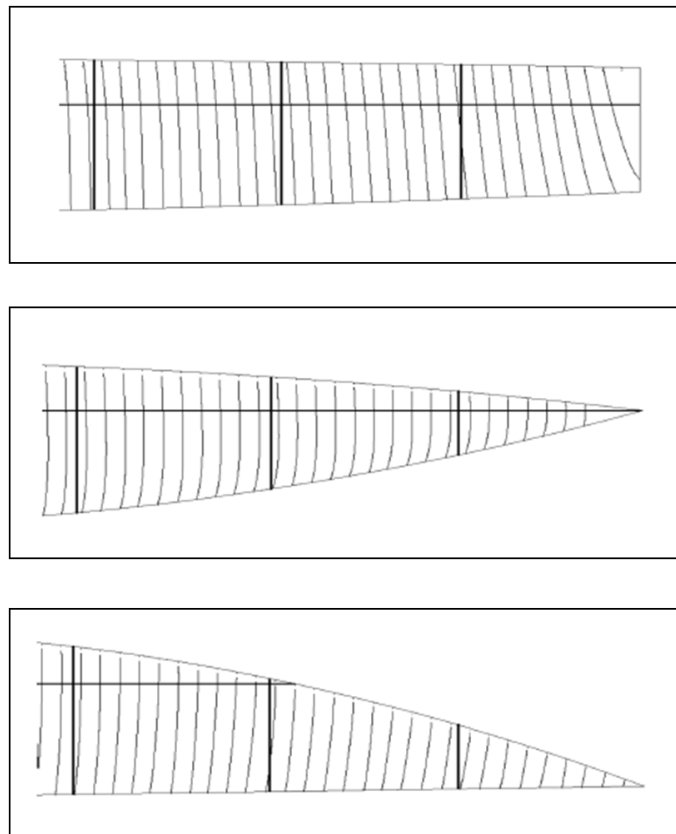


Figure 1.2. Different blade tip geometries marked with streamlines, with leading edge of the blade on top [9].

Upstream and lateral wake effects of wind turbines on the performance of nearby placed wind turbines were investigated by Neff et al. [25]. They made studies of interdependence of the wind turbines on the spacing between them when placed in the vicinity of each other. One of the other important factors responsible for unsteady effects on wind turbines is the interaction between the blades and the supporting tower. In the case of downwind configured wind turbines, disturbances can arise due

to the passing of the blades through the wake (or the shadow) of the supporting tower. The tower shadow effects have also been studied in the past using CFD. The region of tower shadow can be considered to be the region of velocity deficit in the wake of the tower. It leads to large changes in the velocities experienced by the blade. Wang et al. [43] developed a high resolution model for tower shadow effects on wind turbines. This model provides basic information on the blade aerodynamics due to the wake originating from the tower. The current work, is limited only to the upwind configuration of wind turbines in which the tower shadow effects do not play a role unsteady aerodynamics of wind turbine blades.

1.3 Goals and Objectives

The ultimate goal of this research is to understand the difficulties and complications associated with using CFD to predict the flow behavior over wind turbine blades. Computational prediction of the flow over wind turbines is a challenging numerical problem because of the complicated aerodynamics and large variation in length scales. Phenomena such as; the unsteady downstream wake, vortex shedding of the blade tips, flow separation, complicated blade geometry due to inherent twist and changes in angle of attack, and highly turbulent flow over blade sections makes CFD prediction challenging, as well as interesting. The objective is to gain an in depth understanding of the important factors related to computational fluid dynamic modeling of wind turbines.

In order to achieve this goal, certain specific objectives were determined. The first phase objective of the research is a CFD analysis of a Delft University (DU) wind turbine blade. The final objective of this research is to perform aerodynamic analysis of the NREL Phase VI rotor. The experimental values needed for the CFD comparison study will be obtained from the NREL Phase VI rotor wind tunnel test.

There are already some previous CFD results for this case. The aim is to match the results from this study to experiments and some of the previous CFD work.

There are three minor objectives involved in this project which contributed towards the successful completion of the project. These objectives involved 3 dimensional (3D) computer representation (CAD modeling) of a wind turbine rotor according to a given airfoil distribution and given sectional chord and twist data. Objectives also involve the development of a CFD code capable of handling rotating geometries, and/or rotating reference frames, and the associated Coriolis terms. Implementation of appropriate boundary conditions, using an appropriate turbulence model, and running the code in parallel on supercomputers were also some of the objectives.

After the code development, the capability of the CFD code to handle rotating geometries effectively needed to be verified. The focus then shifted towards simulations and code validation. A ‘Delft University’ wind turbine blade was used to carry out the initial simulations and validations. Validation included calculating the pressure distribution and power production by the wind turbine at various speeds and matching them with the experimental NREL values.

Once the code was validated and preliminary results for the ‘Delft University’ wind turbine blade were obtained, the most important objective of the project could be considered. This involves modeling the NREL phase VI rotor blade and building the case such that it matches the experimental setup of the NASA Ames wind tunnel test. It also includes performing CFD simulations and generating results. In depth analysis needs to be done which requires calculations of overall pressure distribution, blade sectional pressure distribution, calculation of blade forces, analysis of flow separation, and rotor power production characteristics based on wind speeds.

After the validation of the NREL Phase VI rotor experiment is done, the focus will shift towards completion of three additional objectives which will not be a part of validation studies. Emphasis will be given on the completion of these objectives in the

very final stages of the project. CFD simulations of a full wind turbine comprising of the entire rotor-hub assembly, tower, and the ground will be carried out. This will give an interesting insight on wind flow behavior over the rotor in presence of the tower. Along with that effort will also be made to simulate the wind flow over a wind turbine with yawed rotor configuration. This will be helpful in visualizing the effects of the yaw angle on the tip vortices and the wake downstream the rotor. Finally, the inflow wind with a shear profile will be simulated over a full wind turbine. This will be one of the efforts to duplicate the real life wind flow situation over wind turbines using CFD. A fully functional CFD analysis package for wind turbines has been developed that can be used for analyzing future wind turbine systems.

CHAPTER 2

METHODOLOGY

2.1 Pre processing and Post processing

Pre-processing and post-processing are two important steps in achieving the CFD solution. Pre-processing involves grid generation and converting the grid into a format which can be understood by the CFD solver. It is often the most time consuming and tedious jobs in achieving the CFD solution. Post processing is the final step in a CFD analysis of a problem. It involves visualization of the results obtained from the solver. It is often done by a post-processing software which is coupled to the CFD solver. The following sections explain the steps involved in pre-processing and post-processing in different cases used in this study.

2.1.1 Mesh Generation

As mentioned above, pre-processing is the first important step towards obtaining the CFD solution. Grid generation is the most important pre-processing step. It is very important to generate accurate grids for the solver so as to obtain correct results. The accuracy of the CFD solution depends on the quality of the grid used to perform the calculations. The two types of grids that are mostly used are structured and unstructured grids.

A structured grid is a collection of regular repeating elements. Such grids are generally represented by quadrilateral elements in 2D and hexahedral elements in 3D. Since the elements are arranged in a regular repeating pattern, the connectivity information of the elements is stored implicitly. Each cell in the grid is directly addressed

by the index (i, j) in 2D or (i, j, k) in 3D. This helps in saving a lot of computational effort while performing calculations. In order to improve the overall accuracy of the CFD solution, these grids can be made finer by stretching in a particular direction so that there is a large number of closely spaced grid points in a region where large gradients need to be solved [1]. The figure 2.1 represents a typical structured grid consisting of quadrilateral elements. The downside of using such grids is their inability to resolve finely around complicated geometries, and over refinement away from the bodies.

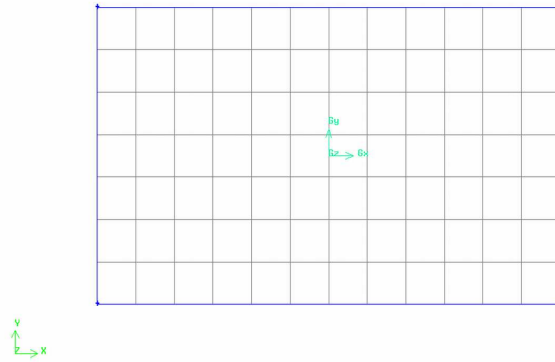


Figure 2.1. A typical 2D structured grid consisting of rectangular elements.

Unstructured meshes consists of arbitrarily shaped elements which do not have any regularity among them. Such type of grids are mostly used to perform finite volume and finite element calculations. The main advantage of using unstructured meshes is their ability to resolve finely around very complex geometric topologies. Such type of meshes are generally represented by triangles in 2D and tetrahedrons in 3D geometries. Since there is no repeating pattern of the elements, the connectivity information of the elements needs to be stored explicitly. This comes at some additional computational cost since extra memory is utilized in storing the mesh connectivity information. The advantage is that, algorithms exist to fill any space,

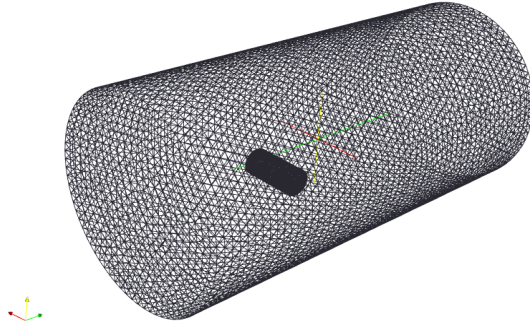


Figure 2.2. A 3D unstructured grid around a finite cylinder consisting of tetrahedral elements.

no matter how complex the shape is, with an unstructured mesh. With the latest developments in the meshing algorithms, various mesh generators are equipped with advanced meshing options which give the user considerable control in generating very accurate unstructured meshes [2]. Figure 2.2 shows a cylindrical domain with typical unstructured (tetrahedral) mesh around a 3D finite cylinder. Unstructured meshes are used throughout this research so that accurate CFD calculations can be performed over the wind turbine blades which have reasonably complex geometry. Figure 2.3 shows a 3D mesh around a wind turbine rotor while Figure 2.4 shows the surface mesh consisting of triangles on one of the wind turbine blades.

The mesh generation software, Gambit [2] has been used throughout this work to generate meshes. The meshes made in Gambit are converted into a format suitable for OpenFOAM applications and are then fed to the solver. In order to have a stable solution, care should be taken that the grid consists of least number of elements with high aspect ratios (skewed elements). Thus good mesh generation requires a fair amount of experience, and some careful interpretation by the user. In this work, generating good quality (less skewed elements) tetrahedral meshes around wind tur-

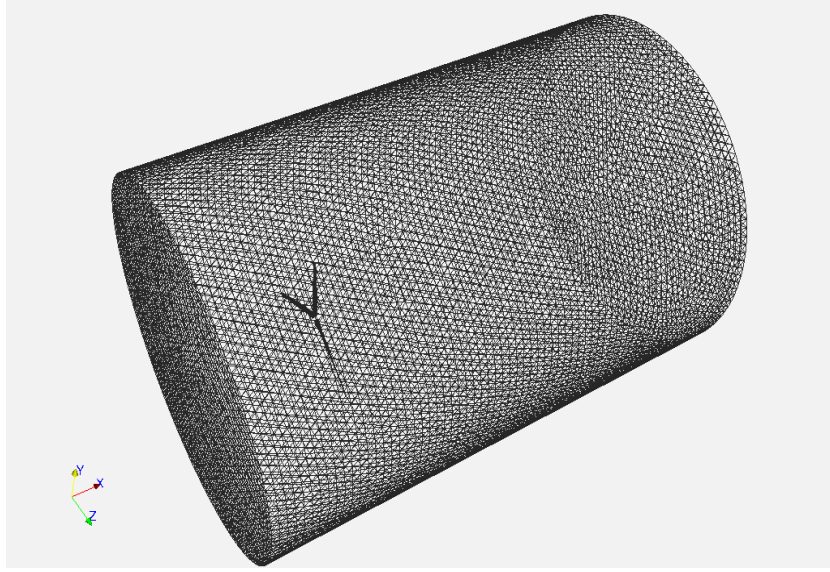


Figure 2.3. Unstructured mesh consisting of 2 million tetrahedral elements around a wind turbine rotor.

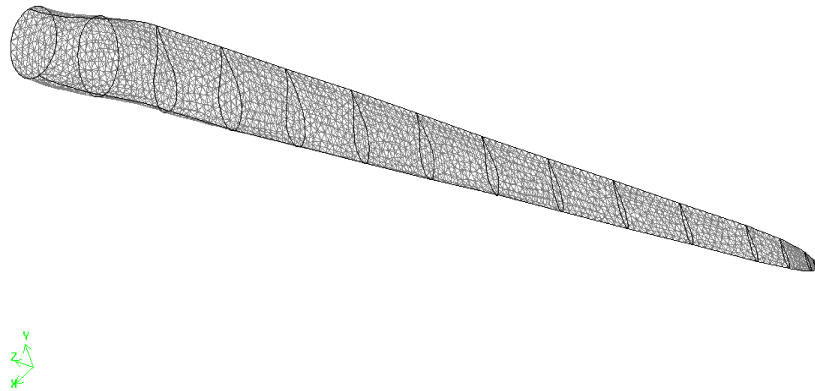


Figure 2.4. Surface mesh showing triangular elements on a wind turbine blade.

bine rotor, required spending a considerable amount of time and efforts (about 8 - 10 hours).

2.1.2 Domain Decomposition

Domain decomposition is also one of the major steps involved in pre-processing. For performing large calculations which are computationally expensive the underlying mesh is decomposed into various number of smaller meshes that are distributed to various processors for performing numerical calculations in parallel using many CPUs. The program used for partitioning the mesh is METIS. The figure 2.5 shows a typical representation of METIS partitioned mesh colored by the processor IDs. The University of Massachusetts in-house supercomputer cluster, Cyclops, was used for parallel computations. Cyclops is a 76 node, 608 core, parallel computing cluster. The number of partitions of the meshes, for parallel processing were made according to the complexity of the problem. For example, meshes used for flow simulations over wind turbine rotors consisted of about 2.5 million tetrahedral elements. These meshes were decomposed into 24 smaller partitions such that the case runs on 24 processors with about 100,000 cells on each processor. This specific number for partitioning was decided so that there was least waiting time on each processor during the parallel run.

2.1.3 Post processing

The post processing software, ParaView [19], was used in the current work for analyzing as well as visualizing the results obtained from the MRFSimpleFOAM solver. ParaView is accessed with the help of the OpenFOAM built in post processing module *paraFoam*. The output obtained from the solver is in the *foam* format and is imported in ParaView with the help of *paraFoam*. Although, ParaView has built-in capability of performing various post processing operations, some utilities were separately written to perform some specific post processing calculations. The table

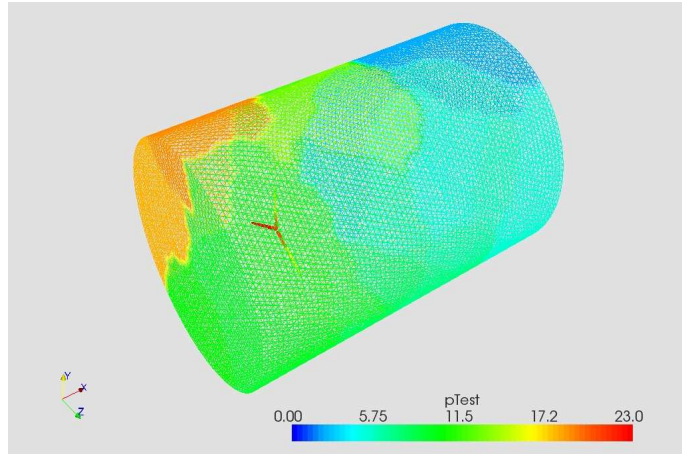


Figure 2.5. Mesh around wind turbine rotor is decomposed into 24 smaller meshes using a METIS partitioning method. Colored by processor Id.

2.1 gives the list of hard coded utilities that were used to perform further analysis on the CFD results. Along with these utilities many other built-in ParaView utilities were used to process the results. For a more detailed explanation on the use of ParaView, please refer to [21].

Utility	Operation
vorticity	Calculates the vorticity magnitude as well as components.
calculateTorque	Calculates torque and power at the specified patch.
patchIntegrate	Integrates a field (p, U, k, etc) over the specified patch.
Q	Calculates the second invariant of the velocity gradient tensor.

Table 2.1. List of some post-processing utilities used in the simulations.

When the code is run in parallel on multiple processors, post processing can be done in two ways. One approach would be to post process the data on each individual domain while the other approach is to reconstruct the complete domain and fields from different processors on to a single processor and analyze the whole

simulation completely at once using ParaView. The latter approach of post processing is employed in this work.

2.2 Code Formulation - MRFSimpleFoam

The Multiple Reference Frames(MRF) application has been implemented in OpenFOAM in this work. Flow of air over a rotating wind turbine rotor implies that the fluid flows in an inertial frame of reference while the rotor rotates in a non inertial reference frame. Thus in this work, an effort has been made of applying the Navier-Stokes equations in multiple frames of reference. In this approach, the fluid flow is solved in a rotating reference frame while some of the boundary conditions are specified in the stationary (or laboratory) reference frame. Implementing this principle in a CFD solver involves coupling of non-inertial forces to the existing fluid flow equations in the inertial frame. The Navier-Stokes equations are solved in a non-inertial frame that rotates with the blades. A detailed explanation of code implementation is given in the following sections. Modeling the flows that involve rotating as well as stationary zones can be done by various other methods that involve mesh motion [15] [27]. These methods make the mesh in the rotating patch move, thus requiring high computational effort. This particular model has been selected because of its ability to work in an equally efficient way even on coarser grids [5]. This method makes the mesh work instead of moving the blade through the mesh. In this method, the entire domain rotates while some of the boundaries are stationary. The reason for faster working and simplicity of this method as compared to the methods involving mesh motion is that it does not have to perform calculations of each time step at different positions of the mesh since the position of the mesh is always fixed. Let us now look at the backbone of the code and the way code encounters the MRF zones.

2.2.1 RANS modeling

The CFD solver used for the purpose of this work is a steady state turbulent flow solver. The Reynold's Averaged Navier-Stokes equations for the incompressible fluid flow are given by the following equations 2.1 and 2.2. They represent the conservation of mass and momentum respectively [46].

$$\nabla \cdot \vec{u} = 0 \quad (2.1)$$

$$\rho \frac{D}{Dt} = -\nabla p + (\mu + \mu_t) \cdot \nabla^2 u \quad (2.2)$$

The above two equations are also referred to as the continuity and momentum equations.

Finding the solution for the incompressible Navier-Stokes equations is not simple because there is no time advancement for pressure. One approach to solve the incompressible Navier-Stokes equations is to derive an explicit expression for pressure by taking the divergence of momentum equation (2.2). We get the following equation:

$$-\nabla \cdot (u \cdot \nabla u) = \nabla^2 \cdot \frac{p}{\rho} \quad (2.3)$$

One approach to solving 2.1 and 2.2 is the Semi Implicit Method for Pressure Linked Equations (SIMPLE) algorithm. In this method, an approximate velocity field is obtained by solving the momentum equation with an old pressure and then the pressure term is calculated using equation 2.3 and the latest velocity information. This algorithm is implemented in OpenFOAM and was used to solve the equations. For a detailed explanation of the implementation of the SIMPLE algorithm in OpenFOAM, refer to [16]. The algorithm for solving the Navier-Stokes equation with the SIMPLE loop is given in Appendix A.

2.2.2 Multiple Reference Frames

In our simulations the wind turbine rotor rotates about a particular axis at a certain angular speed set by the user. The Navier-Stokes equations for a fluid in the inertial (laboratory) reference frame cannot be directly applied to the reference frame spinning with the rotor. Thus, a co-relation between the two reference frames needs to be established and simultaneously changes must be made to the Navier-Stokes equations. Figure 2.6 shows a schematic representation of two reference frames. The inertial (or laboratory) frame is denoted by the upper case letters X, Y, Z while the non-inertial (or rotating) frame is denoted by the lower case letters x, y, z . The non-inertial axes x, y, z is placed at a distance $R(t)$ from the inertial frame and rotates with an angular velocity $\Omega(t)$ that are partially functions of time.

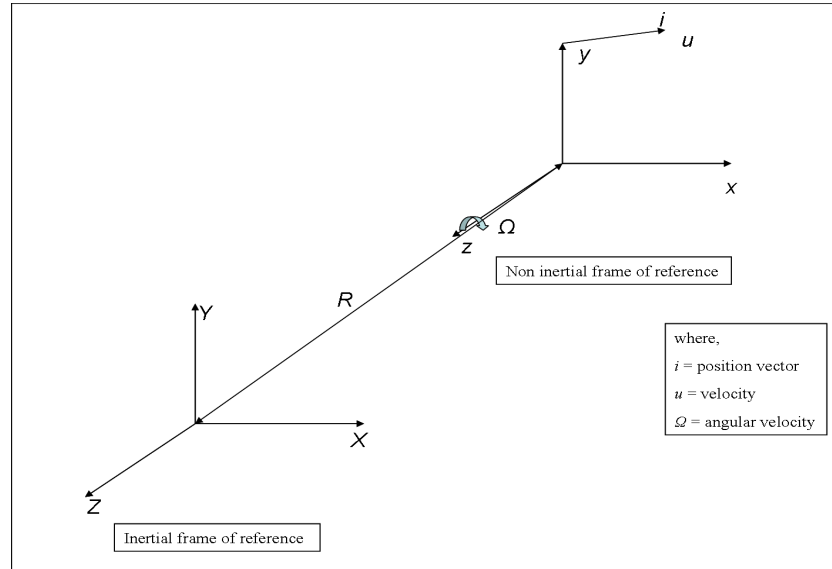


Figure 2.6. Co ordinate system representation of two reference frames.

The Navier-Stokes equation from the equations 2.1 and 2.2 are applicable only in the inertial frame. For a non-moving, steady rotating reference frame ($R = 0, \Omega = \text{constant}$), the above NS equations become:

$$\rho \frac{Du}{Dt} = -\nabla p + \mu \nabla^2 \vec{u} - \rho \cdot [2 \cdot \vec{\Omega} \times \vec{u} + \vec{\Omega} \times \vec{\Omega} \times \vec{r}] \quad (2.4)$$

where, \vec{r} = position vector of the rotating frame.

$2\vec{\Omega} \times \vec{u}$ = the Coriolis acceleration term due to rotation.

$\vec{\Omega} \times \vec{\Omega} \times \vec{r}$ = the centripetal force term.

The above equation 2.4 is the Navier-Stokes equations for non-inertial reference frame [44]. This is implemented in the MRF version of OpenFOAM to account for the rotation effect of the wind turbine rotor. Thus, this approach of treating the rotating geometry is quite simple since it only involves the addition of an extra Coriolis term (the centripetal force term $\vec{\Omega} \times \vec{\Omega} \times \vec{r}$ can be absorbed into the pressure term (∇p)).

2.2.2.1 Working of Code on Zones

In this section, the way in which separate zones are made and the code operates on them is shown. This is important because some of the planar boundary conditions are specified in non-rotating frame. The details about the rotating zones are specified in the OpenFOAM *MRFZones* file. The name of the rotating patch, the desired axis about which it rotates and the desired angular velocity are specified by the user. An example *MRFZones* file is given in the Appendix B.

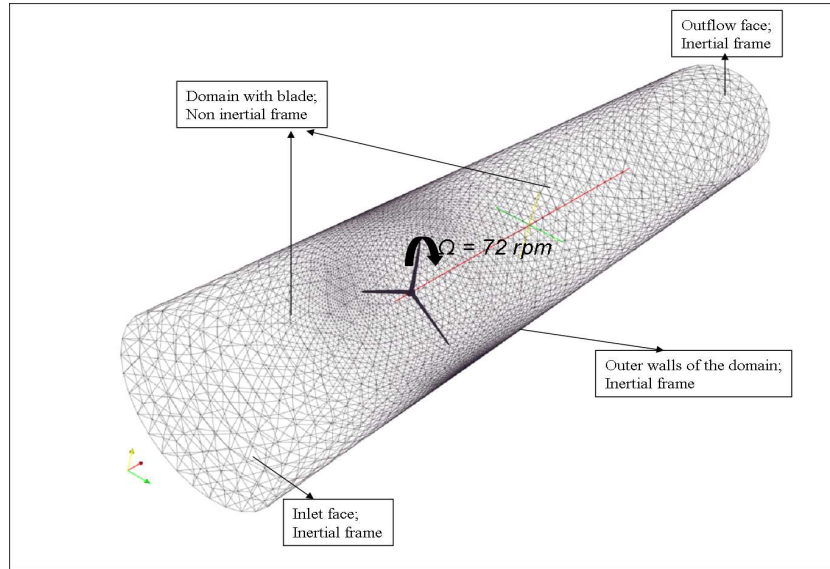


Figure 2.7. Schematic representation of multiple reference frames. Fluid in the inertial frame and the wind turbine rotor, and domain in the rotating frame.

Once the rotating and inertial zones are classified and defined in the code, the next task is to generate cells, faces, and points belonging to the rotating patch or patches. A pre-processing script named *makeMesh* was formulated using some already existing OpenFOAM mesh handling utilities. Now the code is totally aware of the cells and faces in the rotating zones and Coriolis terms are added to the equations on those patches. Figure 2.7 shows the representation of different reference frames in the working model.

2.3 Code Formulation - TurbDyMFoam

In the later phases of this project, simulations of wind flow are carried out over an entire wind turbine unit i.e. consisting of rotor, hub, and the tower. The MRFSimpleFoam solves for fluid flow in non-inertial reference frames. In the rotor simulations, the rotor along with the computational mesh was set to be in the rotating reference frame while the inlet and the outlet boundaries were in stationary reference frames. With the addition of tower and the ground plate, the domain becomes a combination of rotating as well as non rotating elements. In this situation, the MRF solver is unable to resolve the wake in the domain consisting of the stationary tower. Hence, there was a need for a rotating (moving mesh) solver that would rotate the just the mesh around the rotor.

TurbDyMFoam, a transient state, moving mesh solver for turbulent incompressible flows, was used for these simulations. This code solves the Reynold's Averaged Navier-Stokes equation for incompressible fluids. The moving and the static parts of the mesh are explicitly specified in the solver settings. This solver uses the Pressure Implicit with Split Operator (PISO) algorithm for solving the pressure-velocity coupling in the NS equations. The moving mesh operation of the code depends upon the *dynamicMeshDict* file which contains the information about the moving and the static patches. Also, depending upon the type of the interface specified, the code handles

the mesh motion. For a sliding interface type of boundary, the code performs mesh manipulation after each time step. For a GGI type of interface, the code performs interpolation between the rotating and the static patches of the mesh.

2.3.1 The Generalized Grid Interface (GGI)

In the case of CFD simulations involving rotating parts, a moving mesh approach is generally used. In this approach, the interface between the the two parts is called the sliding interface. At each time step, in order to account for the rotation, the neighboring cells of the sliding interface are moved.

The GGI is a coupling interface used for joining two non-conformal meshes in which patches and the mesh points on the either sides do not match with each other. For example, consider the case of flow of a liquid in a mixer vessel. The interface in between the rotor and the stator parts can be a GGI interface. Instead of re-meshing, GGI uses weighted interpolation methods to evaluate and transmit the flow variables across the coupled patches.

The operation and implementation of GGI is simple. The patches on both sides of the interface are set as type GGI. One of them is the *GGI Master* while the other is the *GGI Shadow* patch. In order to clearly understand this concept, consider the example of simulation case in the current project. For the simulation of flow over full wind turbine, two non conformal meshes are used. The outer mesh consists of all the stationary parts (i.e. tower, plate, and the computational domain). The inner mesh consists of the rotary parts (i.e. the rotor and hub). The interface between these two meshes is the GGI interface. The outer face of the rotating mesh is the shadow patch of the inner face of the stationary mesh. The figure 2.8 shows the GGI patches in the domain.

As mentioned before, GGI patches interpolate flow values between them and thus account for the mesh rotation. In order to compute the flow values, a list of GGI

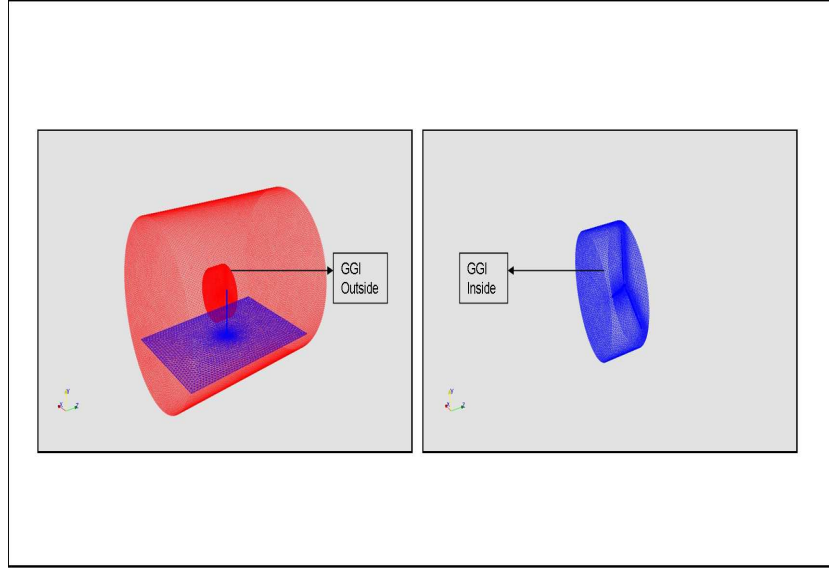


Figure 2.8. Representation of GGI patches in two non conformal meshes. GGI Inside for the rotor mesh and the GGI Outside for the stator mesh.

weighting factors need to be evaluated for both the master and the shadow patches. These weighting factors are basically the percentage of surface intersection between the two overlapping faces. The correct implementation of GGI in the turbDyMFoam depends on *boundary* and the *dynamicMeshDict* files in the solver. The *boundary* file describes the physics properties of the mesh patches while the *dynamicMeshDict* file describes the aspects of mesh rotation. For the boundary conditions of these patches, the definition of all the fields (U , p , k , ϵ , ν_t , ν_{tilda}), is simply GGI. The sample dictionary files are given in Appendix C.

The results obtained with GGI are compared with the results from a sliding interface solver in [4]. It is observed that the results with the non conformal meshes match well the results from the conformal version of the same test case. The GGI algorithm is fully parallelized, but all the faces of the interface should be present on one single processor. In the current project, for the mesh with 4 million cells, the code was run on 32 processors. It is observed that of all the steps, GGI weight evaluation was the

most time consuming part. Hence it is clear, that a further research in advancement of the GGI algorithm is needed to support large simulations.

2.4 Boundary and Initial Conditions, and Turbulence Model

2.4.1 Boundary and Initial Conditions

In order to complete the CFD problem that we are solving, it is necessary to specify boundary conditions for the domain patches. Patches of the domain on the boundaries are recognized by the solver as the boundary patches. The boundary conditions can be divided broadly into two types viz. Dirichlet and Neumann boundary conditions. If a dependent variable is described by a value on a patch, then such type of boundary condition is called as Dirichlet boundary condition. Similarly, if a gradient is specified at the boundary patches, then the boundary condition at those patches is called as Neumann boundary condition. Therefore, the boundary conditions used in the code are either of fixed value type or fixed gradient type.

There are several boundary conditions implemented in MRFSimpleFOAM and turbDyMFoam. The list of some of the useful boundary conditions used in this project is tabulated below in table 2.2. For the complete list of the boundary conditions supported by OpenFOAM, please refer [21].

The inlet patch of the computational domain is set to a fixed value of velocity. The major objective of this research was to establish a good match between the CFD results of the simulations and the experimental values for the flow over wind turbines. Many cases were simulated for different values of constant in-flow wind speeds. Thus, a change in the velocity boundary condition at the inlet ultimately results in the change in wind speeds all across the computational domain. Similarly, the outlet boundary of the domain was set to a constant pressure value. The pressure at the outlet was set to be atmospheric pressure. The objects in the computational domain (i.e. the flat plate, finite cylinder, and the wind turbine rotor), around which

Boundary Condition	Operation
fixedValue	Fixes the value of a particular patchfield to the value specified by the user.
zeroGradient	Normal gradient of the patchfield is 0.
slip	For a vector field, the normal component is set to 0 while tangential is set to zeroGradient. For a scalar field, it is same as zeroGradient.
cyclic (periodic)	Sets the periodic boundary condition across the patches.
ggi	interpolates flow variables between the moving and the static mesh patches.

Table 2.2. Boundary conditions used and their operation.

the flow was simulated, were set to be no-slip boundaries. The no-slip boundary conditions set the stream wise and the transverse velocities to zero. Periodic boundary conditions were also used during the simulations. These boundary conditions allow us to simulate $1/3^{(rd)}$ of the rotor and the domain. Therefore $1/3^{(rd)}$ of the computational effort is reduced by doing so. When periodic boundary conditions are applied to patches which are coupled to each other, it implies that the velocities going out of one patch enter into the domain through the other coupled patch. Figure 2.9 shows a schematic representation the periodic boundary conditions. Refer to [22] for a detailed explanation of implementation of boundary conditions in OpenFOAM. In these simulations, velocity in the entire domain was set to the inlet value as the initial condition. Similarly, for pressure and eddy-viscosity term $\tilde{\nu}$, initial conditions in the domain were set to the inlet value.

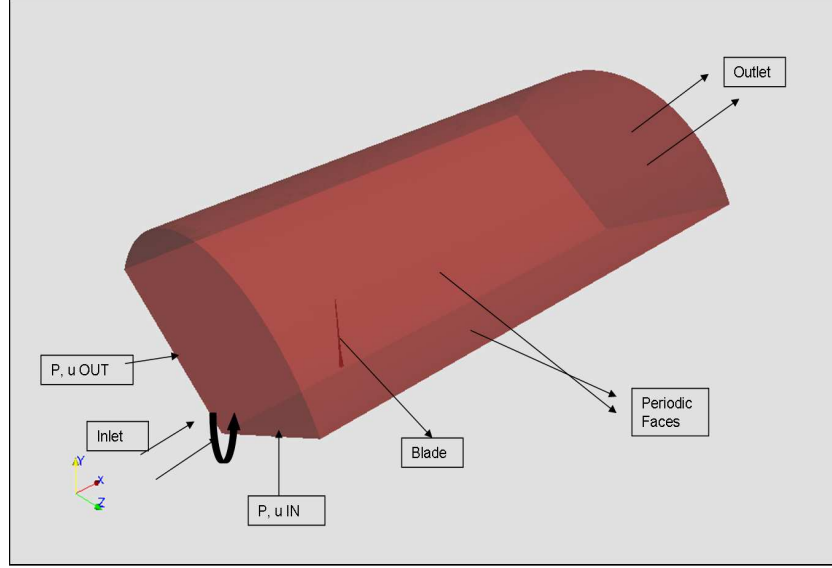


Figure 2.9. Schematic representation of periodic boundary conditions.

2.4.2 Turbulence Model

The Spalart-Allmaras (1994) [37] one-equation turbulence model was used in the code to solve for the turbulent eddy-viscosity. This model involves one differential equation for the eddy viscosity ν_T . The turbulent eddy viscosity is given by $\nu_T = \tilde{\nu} f_{v1}$ where $f_{v1} = \frac{X^3}{X^3 + C_{v1}^3}$ and $X = \frac{\tilde{\nu}}{\nu}$.

Here, ν is the molecular viscosity. $\tilde{\nu}$ follows the following transport equation

$$\frac{D\tilde{\nu}}{Dt} = c_{b1}(1 - f_{t2})\tilde{S}\tilde{\nu} + \frac{1}{6}[\nabla \cdot ((\nu + \tilde{\nu})\nabla\tilde{\nu}) + c_{b2}(\nabla\nu)^2] - [c_{w1}f_w - f_{t2}\frac{c_{b1}^2}{k}][\frac{\nu}{d}]^2 + f_{t1}\nabla^2(u^2) \quad (2.5)$$

Equation 2.5 is already implemented in OpenFOAM. For a detailed explanation of this turbulence model, the design of the above equation and its applications, refer to Spalart-Allmaras (1994) [37].

The Spalart-Allmaras turbulence model is a relatively simple one-equation model that works well for simple boundary layer turbulence. It solves for the modeled transport equation for the kinematic eddy viscosity. This model was designed specifically

for aerodynamic applications. The computational effort required in the use of one equation models is low because only one variable has to be solved for. Various comparisons of different turbulence models like Spalart Allmaras, standard $k - \epsilon$, $k - \omega$, and the SST model on the basis of performance and predictive capabilities have been made by many authors so far[3][8]. Also it has been shown that for the analysis of wake strength, turbulent transition and other aerodynamic properties on attached airfoils, the Spalart Allmaras turbulence model obtains reasonably accurate results [8]. It is also very convenient to use this model on unstructured meshes. The validation of this turbulence model has been done in detail and can be found in [37]. Therefore, because of its special application in the aerodynamics and turbo machinery, and because of its ability to save a lot of computational effort, this one-equation Spalart-Allmaras model was chosen for simulations.

The working of Spalart-Allmaras model depends on the boundary conditions set for $\tilde{\nu}$. In our simulations, following are the boundary conditions for $\tilde{\nu}$ on different patches:

Patch	BC
inlet	Fixed value (0.001)
outlet	zeroGradient
side walls	zeroGradient
rotor	zeroGradient
periodic face	Fixed value (0.001)

Table 2.3. Boundary conditions for $\tilde{\nu}$

CHAPTER 3

PRELIMINARY RESULTS

Before performing the CFD simulations over the actual NREL Phase VI wind turbine rotor, the CFD code is first validated for the case of flow over a rotating tube and the flow over a flat plate at an angle of attack. The rotating tube example validates the implementation of the CFD code in multiple frames of reference. The case of flow over a finite flat plate is chosen to study the behavior of flow past aerodynamic geometries. In this case the phenomena such as trailing edge vortices and the wake strength downstream the plate are studied.

3.1 Flow Past a Rotating Tube

MRFSimpleFoam was implemented for simulations over a rotating tube. Since the ultimate objective of this research work is to simulate a wind turbine rotor, a similar case was setup with a comparatively simple geometry. This is shown in figure 3.1. Figure 3.1 shows a schematic representation of the case setup.

This approach would help in testing the performance of the code in handling rotating geometries. A finite cylindrical tube is rotated at a constant angular velocity of 3.1416 rad/s or 30 rpm in the negative z direction. The rotation of the tube is implemented in the code by defining the zone around the tube to be the rotating zone and explicitly specifying the rotation axis, direction, and the magnitude. The rotational speed is selected such that the operating conditions of the setup match with those of a medium capacity wind turbine. The Reynold's number of the flow depends on the diameter of the rotating cylinder. The velocity of the incoming fluid

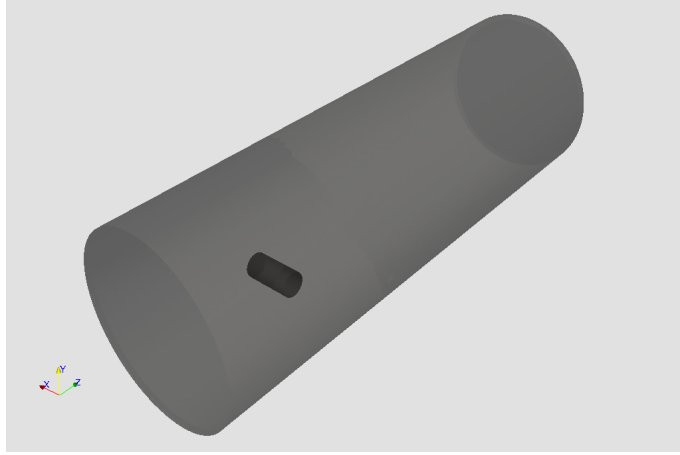


Figure 3.1. Case setup with a small tube aligned with an offset to the flow axis. Offset is selected so as to visualize the wake clearly.

is adjusted such that laminar $Re = 2 \times 10^5$ [7]. Fixed value boundary conditions are applied at the inlet as well as outlet. Inlet is fixed to a constant velocity of 2 m/s while the value of pressure is fixed at the outlet to the atmospheric pressure value. No-slip boundary condition is implemented on the walls of the rotating cylinder by setting the value of velocity on the walls equal to zero. This also ensures that there is no fluid going into the object. Initial conditions of the simulation are set to a uniform velocity of 2 m/s throughout the domain. It should be noted that the inlet, outlet, and side walls boundaries are in stationary frame (non-rotating). The code employs a steady state time scheme. The code uses the Spalart-Allmaras [37] turbulence model for simulating turbulent flow. In this turbulence model the eddy-viscosity variable $\tilde{\nu}$ is solved. $\tilde{\nu}$ is set to fixed value of 0.001 (m^2/s) as the initial condition. Inlet is fixed to the same value of 0.001 while all other boundaries i.e. tube, side walls, and outlet are set to zero gradient boundary condition. In order to clearly visualize the wake behind the rotating tube, a high resolution mesh with 1.2 million tetrahedral elements is used.

The figure 3.2 shows a slice across the meshed domain. The outputs are represented in the laboratory frame. It is quite clear that the grid was resolved finely

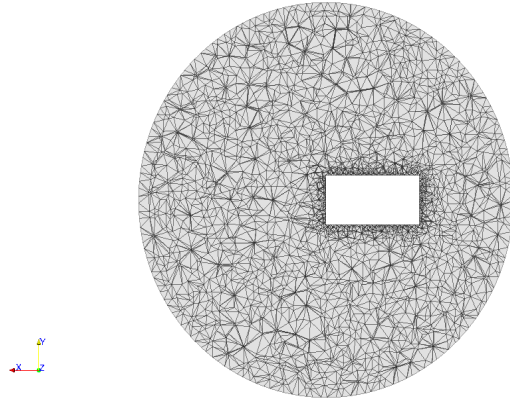


Figure 3.2. Slice through the domain representing the grid resolution. Fine mesh around the cylinder is visible.

around the cylinder in order to capture the boundary layer. Considering the fine grid, the initial time step was set to a low value. It is then adapted as the solution progresses so that the maximum Courant-Friedrichs-Lewy (CFL) number always remains less than 1. The maximum CFL number criterion was set to 0.4, so that the solution remains stable. The code is fully run in parallel on a 608 node in house supercomputing cluster, Cyclops.

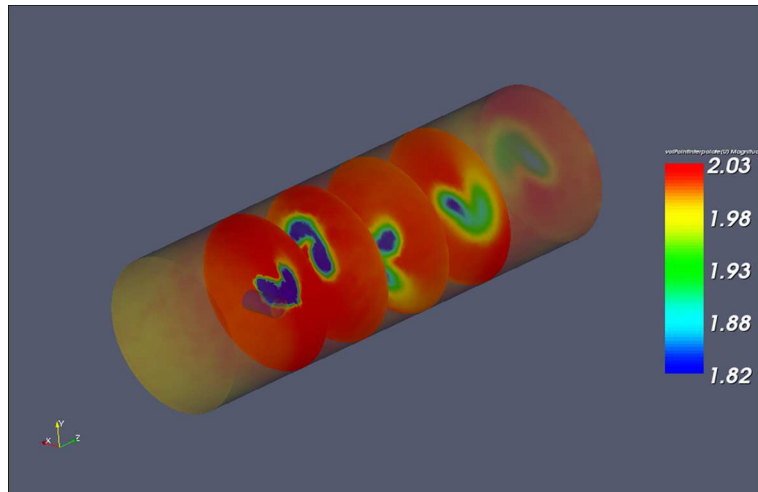


Figure 3.3. Various slices in the domain colored by velocity magnitude. Wake rotation is clearly visible.

Figure 3.3 shows slices at various sections downstream of the cylinder. A rotating wake behind the cylinder is clearly depicted by the contours of velocity magnitude at different positions downstream. This flow behavior indicates that the code can successfully handle rotating objects and thus validates the working of the MRF part in the code. Although, it is to be noted that a direct comparison between the experimental results and the CFD solution could not be done in this particular case.

3.2 Flow Over a Flat Plate

Upon the successful working of the code in rotating frame of reference, the next task was to study the aerodynamic behavior of the flow, specifically the vortex strength. The vortices coming off the tip of a wind turbine blade have a direct relation in determining the power output of the machine. Tip vortices are the tubes of circulating fluid that appear behind the object when it produces lift. The amount of vorticity generated by an object determines the lift of the object. Hence in order to analyze the vortex strength in the flow field a flat plate inclined at an angle to the incident flow was considered. Lift on an airfoil is generated by the pressure difference between the upper and the lower surface. Thus the use of an inclined flat plate is similar to the actual case of wind turbine blade.

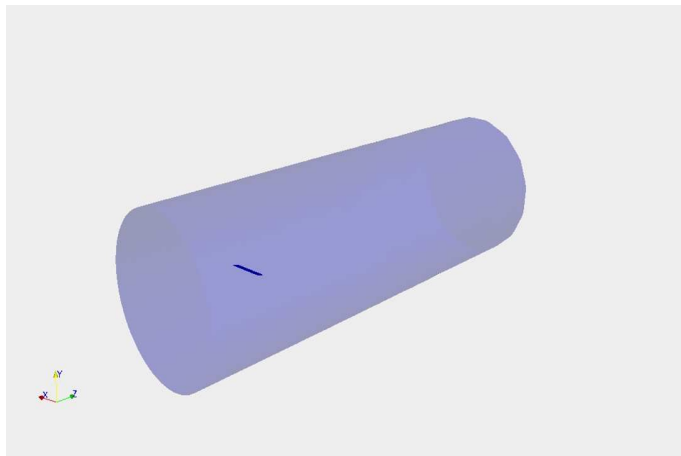


Figure 3.4. Case setup with a flat plate at 4° angle of attack to the incoming flow.

A thin flat plate is inclined at an angle of attack of 4° to the incoming flow. The aspect ratio of the plate is 1:7. Aspect ratio is defined as the ratio of chord to the wing span. The figure 3.4 shows the schematic representation of the case setup. The velocity of the inlet is set to 1 m/s and such that the Reynold's number based on the chord of the plate is close to 1.39×10^5 . The values of aspect ratio (1:7), angle of attack (4°), and the Reynold's number are selected so that the case setup matches to that of the experimental setup by Cayley [18] and Prandtl 1915 [26]. The boundary conditions used are fixed value of velocity and pressure at inlet and outlet respectively along with the surface of the plate being fixed to the velocity value of zero. This ensures that the walls of the plate act as no slip surfaces. The velocity of fluid at the inlet is fixed to the value of 1 m/s, while outflow boundary is set the constant pressure value equal to the atmospheric pressure. In this case, since the plate is stationary the rotation term in the code is turned off by specifying zero angular velocity. The Spalart-Allmaras [37] one-equation model of turbulence is used for solving the turbulent flow. The eddy-viscosity variable $\tilde{\nu}$ is set to fixed value of 0.001 (m^2/s) as the initial condition. $\tilde{\nu}$ at the inlet patch is fixed to the same value of 0.001 while all other boundaries i.e. plate, side walls, and outlet are set to zero gradient.

In this case four different meshes with increasing number of mesh elements were used. This was done primarily to study the effect of mesh resolution on the accuracy of the solution. The figure 3.5 above shows a cross section of the mesh with 2 million tetrahedral elements. The mesh is denser near the plate and gets coarser as it approaches towards the boundaries. This enables the simulation to capture the boundary layer near the plate. The figure 3.6 shows a cut at a distance $10 \times$ chord length ($10c$) from the plate. The trailing vortices behind the tip of the plate are clearly represented by two blobs in the figure. An iso-surface plot of the trailing vortices is also presented in figure 3.7.

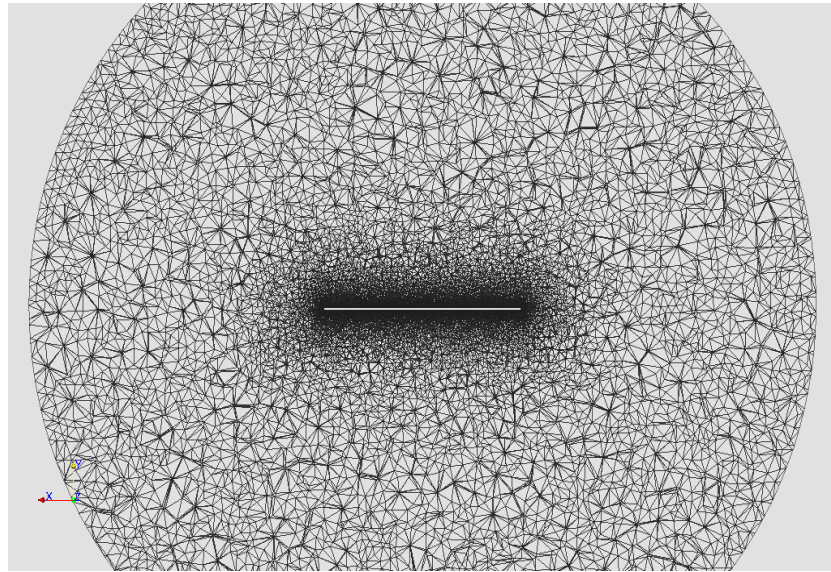


Figure 3.5. Slice through the domain representing the grid resolution. A very fine mesh near the plate is seen.

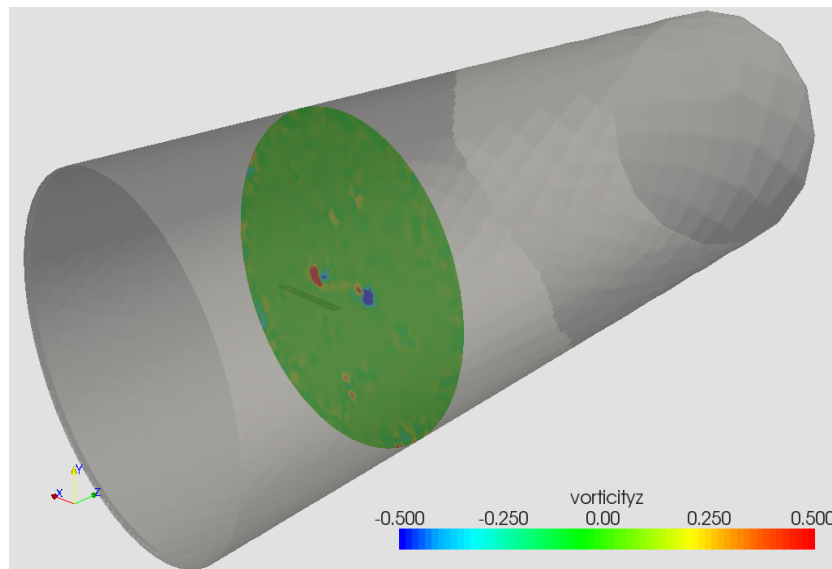


Figure 3.6. Slice of stream wise vorticity at a distance $10 \times$ chord length from the plate.

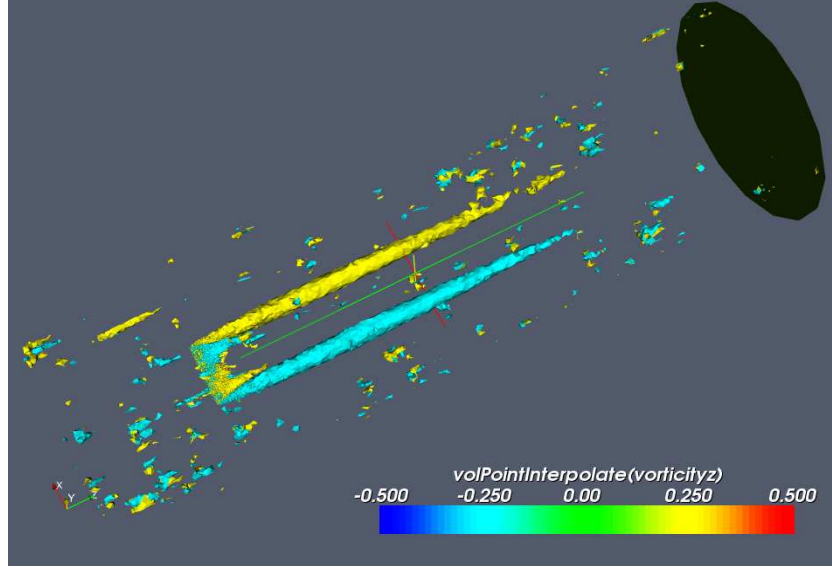


Figure 3.7. Iso-surface plot of stream-wise vorticity.

Further, a grid resolution study was done in order to determine the vortex strength downstream the plate. Since, one of the goals of this work is to analyze the wake downstream a full scale wind turbine, such a study was necessary to roughly estimate the mesh resolution that would be required. The same case was simulated using four different grids each with 0.7 million, 1.4 million, 2 million, and 3 million tetrahedral cells. The effect of vortices downstream the plate was studied. The figure 3.8 below shows a slice across the domain containing 3 million cells at a distance of $50c$ downstream the plate. The trailing vortices behind the plate look prominent at a distance of $50c$ from the plate.

The figure 3.9 shows the variation of z maximum vorticity at location $30c$ and $50c$ downstream the plate with the number of elements in the domain. This is represented by Δx on the X axis. The term Δx is the measure of the stream-wise length scale between the neighboring mesh elements. It is measured as:

$$\Delta x = \text{volDomain}/n\text{Cells}^{(\frac{1}{3})}$$

It is evident from the graph that the values begin to converge as the resolution of the mesh increases. For the vorticity values at $50c$ distance, it is clear that for low

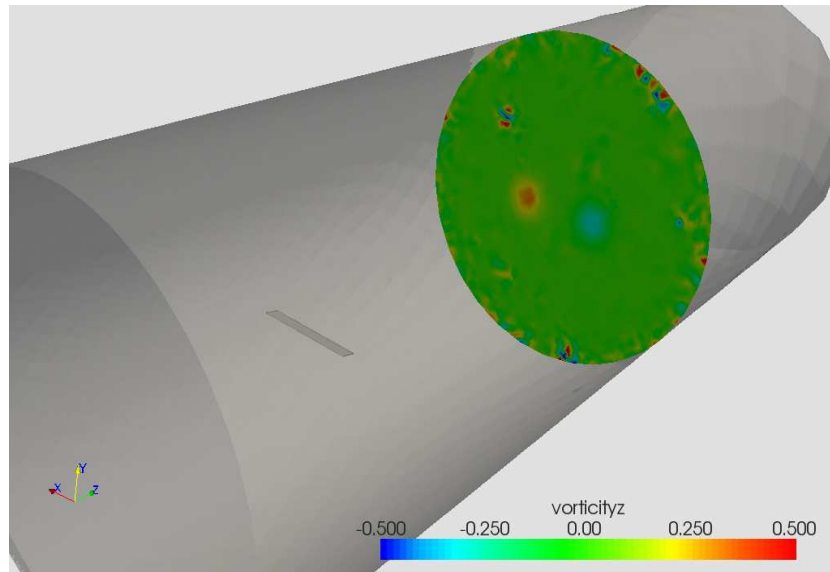


Figure 3.8. Slice of streamwise vorticity at a distance $50 \times$ chord length from the plate.

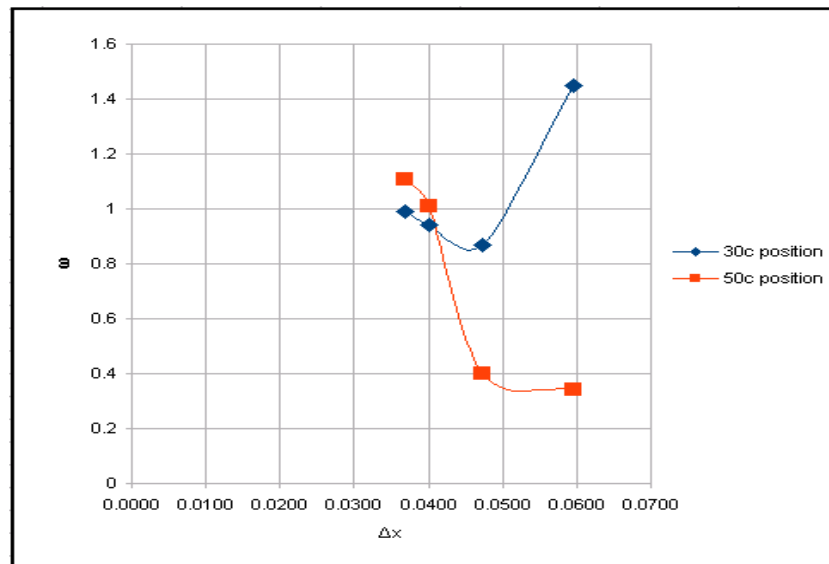


Figure 3.9. Variation of streamwise vorticity with the mesh size at position 30c and 50c downstream.

values of Δx , the solution appears to be converged. The two points, lie close to each other. Similarly, for the case of 30c, solution tends to converge for low Δx values.

3.3 Preliminary Code Validation for the Delft University (DU) Wind Turbine blade

3.3.1 Background

Upon the successful code validation for the rotating geometries as well as analyzing the vortex strength, simulations were performed over a medium scale wind turbine. Previous work by Jonkman [17] describes the design parameters for the wind turbine blade. The size and the power rating of the turbine is selected so that the system is cost effective as well as capable of producing considerable amount of energy [24]. Thus a 5MW wind turbine with 63m rotor radius was designed. Also, the aerodynamic properties of the blade show that eight different airfoil tables were used for the design. Different airfoil tables from the DU (Delft University) and the NACA (National Advisory Committee for Aeronautics) airfoil families were used in the blade design [17]. The DU blade has been used extensively for research purposes before [45], therefore this blade has been selected for the preliminary studies which include code validation for a wind turbine.

The surface of the blade is meshed with triangular elements. The figure 3.10 shows the surface mesh on the turbine rotor. The surface of the rotor consists of about 2 million triangular surface elements. The mesh on the tip of the blades is comparatively fine as compared to other parts of blade. This is because the wind turbine experiences vortex shedding from the tip. Therefore, having a fine mesh on the tips, will help capture boundary layer at the tips. The modeling of the wind tunnel is done so as to accommodate the wind turbine blade. Tetrahedral elements have been used to mesh the entire domain. The figure 3.11 shows the way in which the domain is meshed [33]. Since this blade is 63 m long, the channel (acting as the

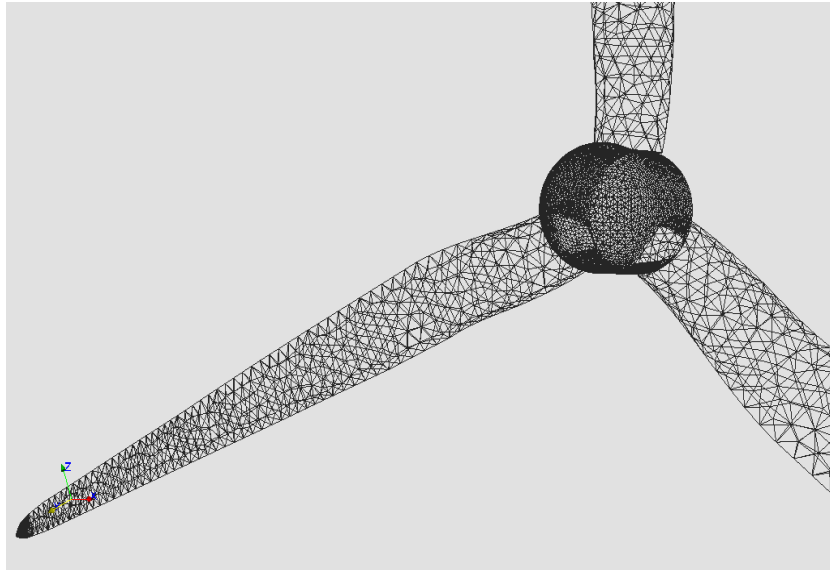


Figure 3.10. Surface mesh on the wind turbine rotor.

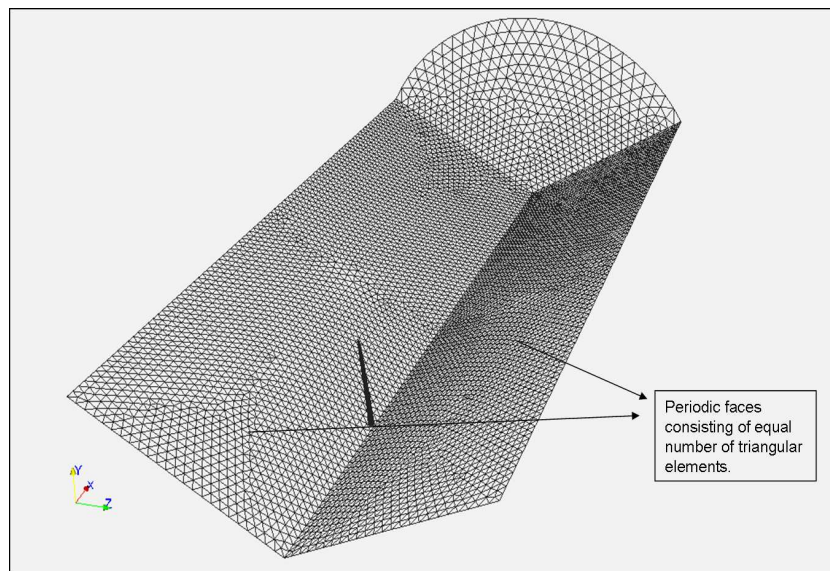


Figure 3.11. Wedge shaped domain with periodic boundaries.

wind tunnel) that encompasses the blade would be minimum of 70 m radius. This would make a very huge domain, and meshing it would involve a lot of tetrahedral cells consuming a lot of computer memory. Hence, to avoid this problem, a wedge shaped channel is meshed with 120 degree periodicity. Periodic boundary conditions are used for the outer walls. Periodic meshing is far less inexpensive than meshing the whole domain. The blade is rotated with the angular speed of 72 rpm, which is angular speed for the experimental setup. In this case the blade is in the MRF region and rotates with the angular speed of 72 rpm. The wind turbine rotor blade is pitched collectively at an angle of 4 degrees.

3.3.2 Results

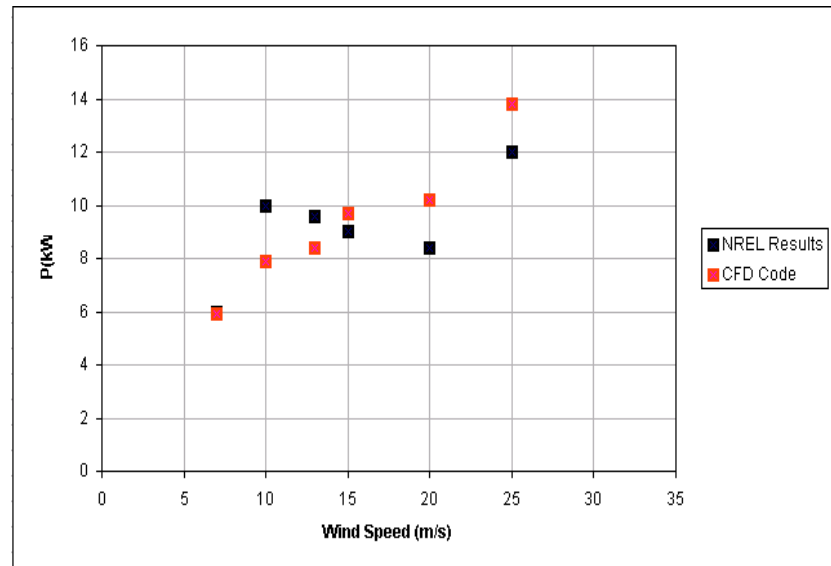


Figure 3.12. Power produced by the turbine blade for different values of wind speeds.

In this section, results obtained from the CFD solution of the simulation over the turbine blade are presented. In order to validate the results, a comparison is made with the experimental values for the NREL Phase VI rotor, which is considered to be a benchmark case in wind turbine design [33]. The blade in the NREL Phase VI

experiment uses the S809 airfoil, an airfoil from the NREL family of thick airfoils for HAWT applications [35] [28]. Thus the blade used for the current study is completely different in the airfoil distribution as compared to the blade used in the NREL Phase VI experiment. Therefore, the results do not exactly match with the experimental values, but a considerable proximity between the results is obtained.

Figure 3.12 shows the aerodynamic power produced by the wind turbine rotor and its comparison with the experimental test results. The graph shows the variation of the power produced with different values of wind speeds. Five different cases are simulated with the different values of the inlet wind speed and corresponding output power is calculated. The results obtained from the MRFSimpleFoam simulation match quite well with the power produced for low wind speed values. However, for higher wind speeds, where the blades operate in stalled conditions, a slight disagreement between the values is seen. This is because the DU blade is built with different airfoil than the S809 airfoil on the NREL blade. Though not perfectly, the CFD results do match with the experimental values reasonably well.

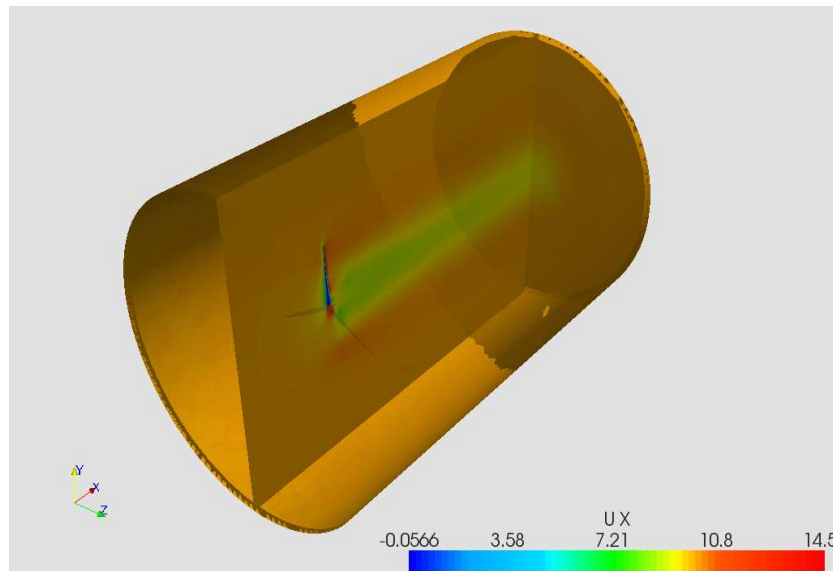


Figure 3.13. Wake behind the wind turbine rotor. It is colored by stream-wise velocity contours.

Results are also obtained from the simulations that represent the wake downstream of the wind turbine rotor. The figure 3.13 shows a cut plane colored by velocity magnitude across the domain. Wake behind the rotor is clearly visible. This is useful in analyzing the effect of downstream wakes on a closely situated wind turbine in a wind farm. Figure 3.14 shows a cut across the domain along the plane of the wind turbine rotor. A slice in the domain, very close to the outlet (about 25m away from the rotor) represents the streamwise vorticity contours in the figure. From the figure 3.15, it is quite clear that, there is still some portion of the wake towards the outlet.

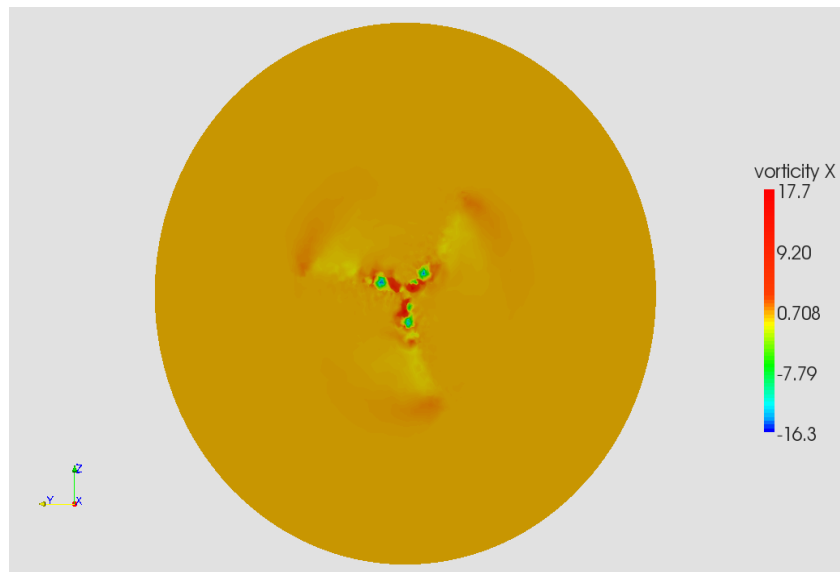


Figure 3.14. Slice across the domain representing the tip vortices.

Iso-surface plots of stream-wise vorticity are also plotted. From the figure 3.16, vorticity shedding from the tip is clearly visible. Figure shows the instantaneous vorticity iso-surface plots after the second rotor rotation. From the figures, we can see that the flow is attached and well defined vortices are shed from the tip of the blades. We also see that the wake is symmetric and vortices diffuse in the stream-wise direction in figure 3.17. However, this diffusion of vortices can also be attributed the grid resolution, which becomes coarser towards the outlet.

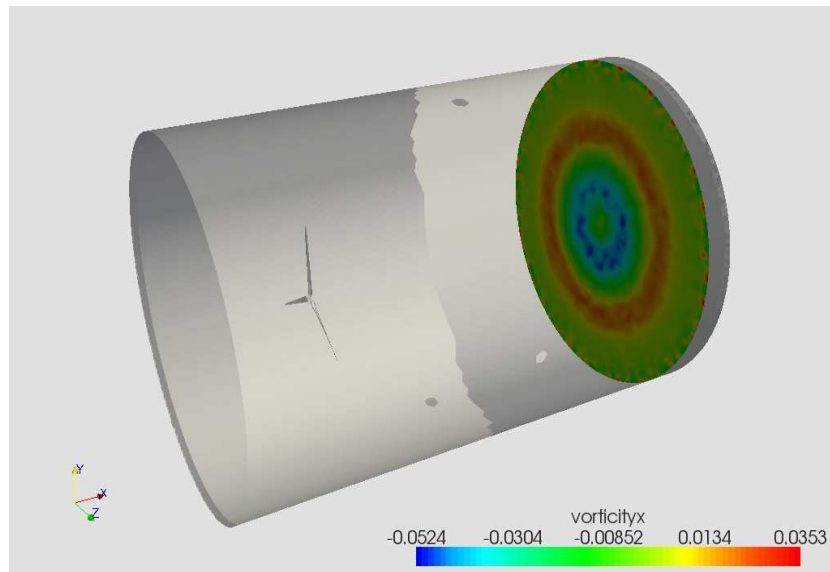


Figure 3.15. Slice across the domain downstream of the rotor, very close to the outlet. It represents the vorticity wake.

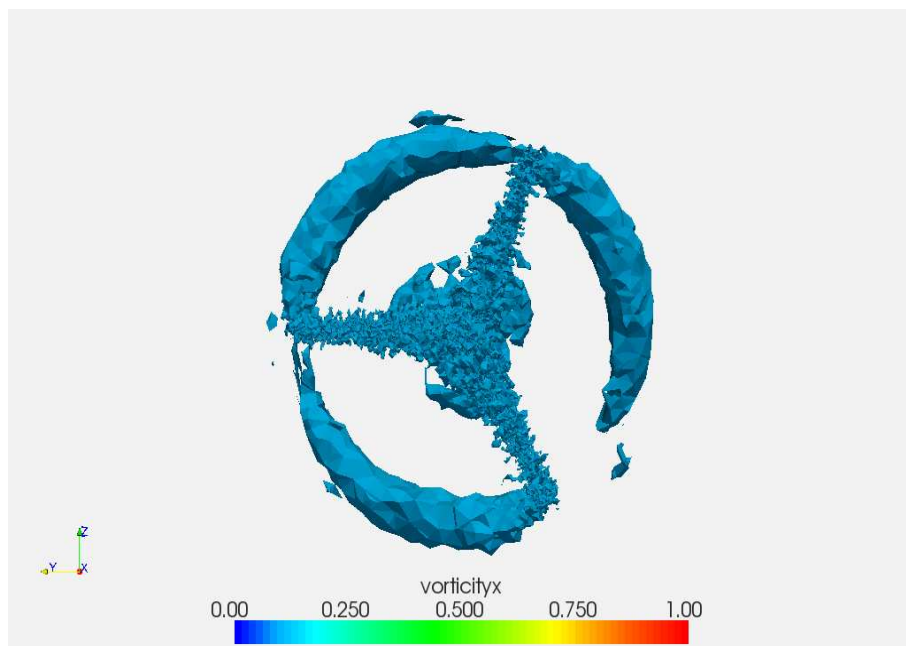


Figure 3.16. Iso-surfaces of stream-wise vorticity representing vortices off the blade tips.

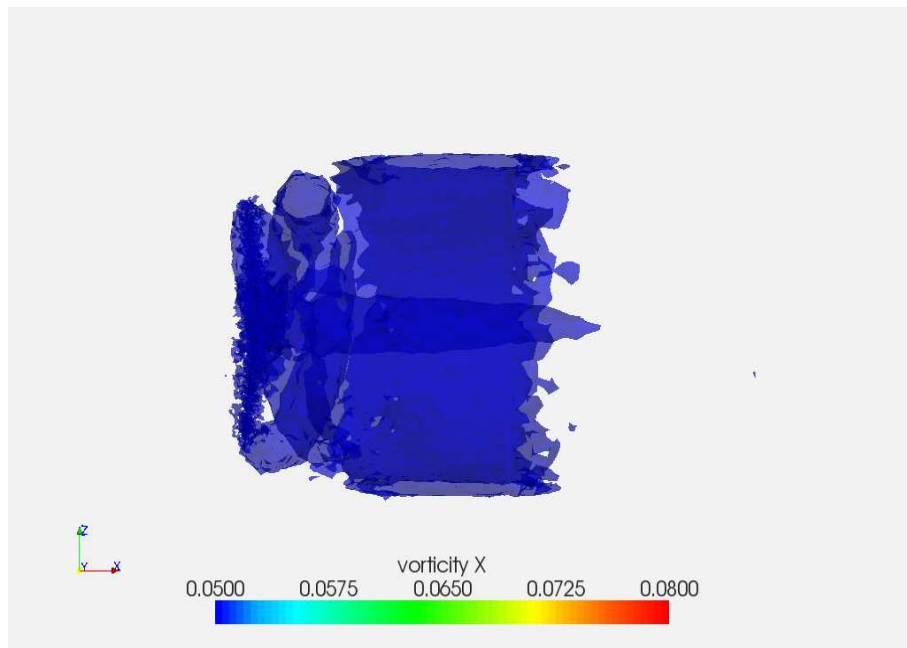


Figure 3.17. Side view of the iso-surface plot of stream-wise vorticity. It represents wake after 2 rotations.

CHAPTER 4

FINAL RESULTS

After the successful code validation and RANS simulations of wind flow over DU rotor, the focus of the project is shifted towards the completion of the final objective i.e. the numerical validation of the NREL Phase VI experiment. Simulations are performed at different wind speeds over the rotating blade in order to calculate power produced and measure the blade forces at all the different inlet wind speeds. A detailed account of the efforts taken to improve the mesh quality in the boundary layer surrounding the blade is also given. Finally, graphs showing the comparison of the CFD results with the experimental results are presented. This also validates the code and the methodology implemented. Wind flow simulations over the full wind turbine assembly give a representation of the actual wind flow behavior due to the presence of tower and rotor yaw.

4.1 Pre processing

4.1.1 3D Modeling of the NREL Blade

The shape and the geometry of wind turbine blades depends on the shape and the orientation of the underlying airfoil. A number of terms are used to describe an airfoil. The locus of points halfway between the upper and the lower surfaces of an airfoil is called the mean camber line. The most forward and the rearward points on this line are on the leading and the trailing edges respectively. The straight line connecting the leading and the trailing edges is called the chord of an airfoil [23]. Depending upon the chord of the airfoil and the angle by which the airfoil is twisted, the shape

of the blade changes. For example, generally, at the tip of the blades, chord decreases while the twist increases giving a small angle of attack and at the blade center, chord increases while the twist angle is almost zero degrees giving a large angle of attack.

The blades of the wind turbine rotor used in the NREL Phase VI experiment are made up of S809 airfoil. In order to build the 3D geometry of the blade, airfoil tables are followed. These tables give a detailed account of the chord and twist distributions of the airfoil along the length of the blade. During the modeling of the blade, the S809 airfoil data from the NREL technical report on the Unsteady Aerodynamics Experiment Phase VI was used [13] [34] [10]. The tables for the blade chord and twist distributions along with the airfoil profile co-ordinates used for modeling the blade in this project are included in Appendix D.

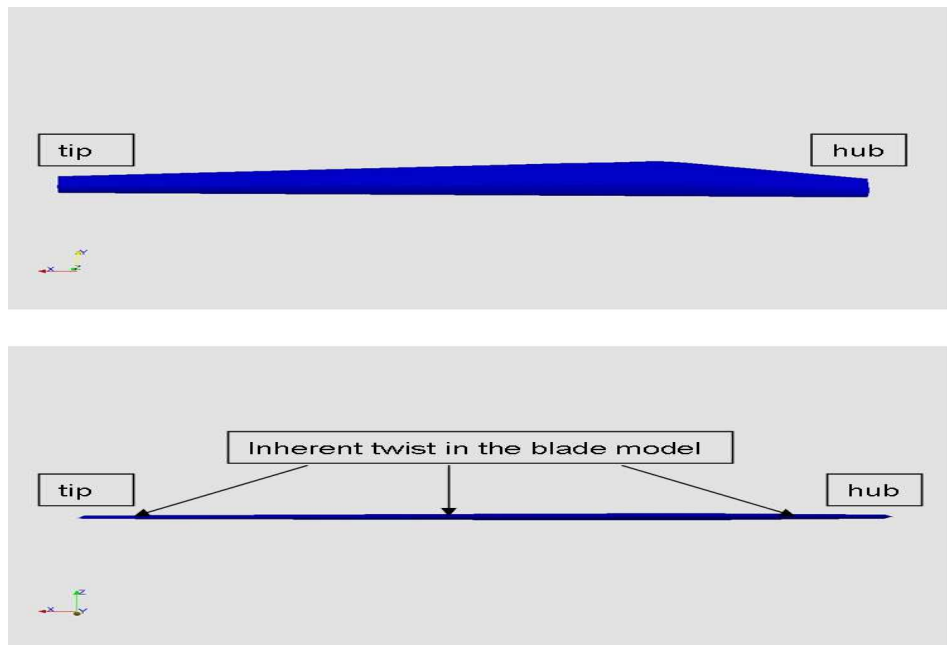


Figure 4.1. 3 dimensional model of the NREL Phase VI experiment wind turbine blade.

Thus, the above airfoil data can be very effectively used to model a wind turbine blade using any geometric solid modeling software. A MATLAB code was written to convert the above airfoil data into a blade surface mesh and later that surface was

exported to GAMBIT to build the three dimensional blade model. The MATLAB code for generating the blade surface mesh is also attached in Appendix D. The solid model of the NREL wind turbine blade modeled using the above mentioned airfoil data and used in this project for CFD simulations is shown in the figure 4.1.

4.1.2 Meshing

Three dimensional meshes for performing CFD simulations are also made using GAMBIT. Triangular mesh elements are used to mesh the blade surface while the computational domain is meshed using tetrahedral or prismatic elements. Tetrahedral elements are used because of the ease in making finer meshes on complicated geometries.

4.1.2.1 Meshes for MRFSimpleFoam Cases

The CFD code MRFSimpleFoam is used for the numerical validation of the NREL Phase VI experiment results. As mentioned previously, for working on the DU wind turbine blade, a periodic computational domain was used. Similarly, for performing CFD simulations over the NREL blade, $\frac{1}{3}^{rd}$ of a cylinder was modeled. A three dimensional model of the computational domain is shown in the figure 4.2.

In an effort to have an accurate solution, the domain is meshed with about 2 million tetrahedral cells. Flow of wind over the blade surface is highly aerodynamic. The upper surface of the blade experiences separation of the flow at certain Re. In this project, Re for all the cases is of the order of 10^5 . Therefore in an effort to accurately capture the separation of the flow over the blade surface, boundary layer meshing with a very large number of mesh points is done around the blade. This method, if supported by a proper turbulence model, will help in accurately capturing the flow separation, and in turn, calculating the exact blade forces. The figure 4.3 is a slice across the computational domain showing the mesh density near the blade surface. Boundary layer mesh surrounding the wind turbine blade is visible in the figure. All

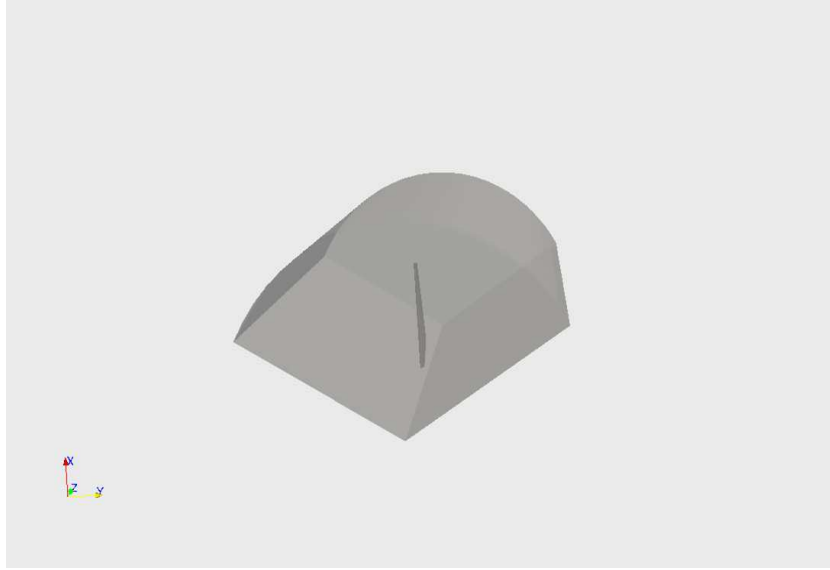


Figure 4.2. 3 dimensional setup of periodic domain used for simulations over the NREL blade.

the time in the process of mesh generation, utmost care is taken to avoid bad quality mesh elements in the computational domain. These meshes (periodic computational domain with boundary layer mesh near blade) are used for the calculation of power and blade forces.

4.1.2.2 Meshes for turbDyMFoam Cases

In the later part of the project, simulations of wind flow are performed over a full wind turbine. Since, this consists of rotor (*rotatingpart*), tower and ground (*stationarypart*), the MRFSimpleFoam solver, due to the lack of its ability to work in such environment, is not used. Instead, turbDyMFoam, a moving mesh solver in OpenFOAM, was used for these simulations. In order to explore the flow behavior over the wind turbines, a wind turbine rotor was modeled with a tower as well as a very flat plate at the bottom representing the ground. The dimensions of the tower are selected such that the height of the tower is about 1.2 times the rotor diameter, while the plate (representing the ground) extends throughout the domain at the base

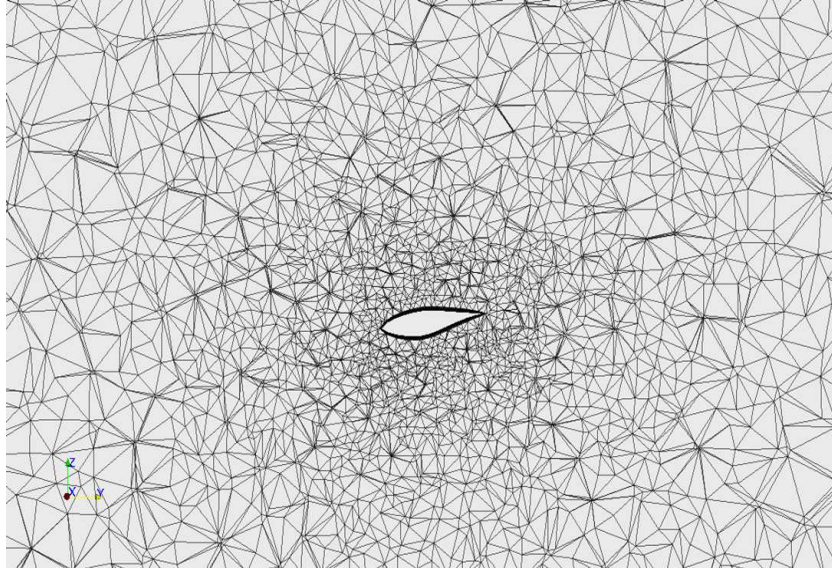


Figure 4.3. Slice across the 3D computational domain representing mesh around the blade.

of the tower. Figure 4.4 shows a schematic representation of the case setup with all the parts of the geometry.

Due to the addition of tower and ground parts, the complexity of the problem increases. It becomes further expensive to mesh the domain, considering the dimensions of the wind turbine components (i.e rotor, tower, and the plate used to represent the ground). Meshes used for these simulations consisted of about 4 million tetrahedral elements. The computational domain consists of 2 non-conformal meshes. The outer mesh consists of the stationary parts (tower, ground plate, and the entire computational flow field). The inner mesh consists of the rotating parts, rotor, and the hub. In order to have the downstream wake well resolved, the mesh in domain was made fine. The inner mesh rotates inside the outer mesh. Figure 4.5 shows the mesh of the computational domain containing the full turbine. A similar mesh is considered for performing simulations over a rotor yawed at 15 degrees to the inflow wind. The only change made in that case is the inner rotating mesh consisting of the rotor was rotated at an angle to the inflow wind direction. Again, it should be mentioned that

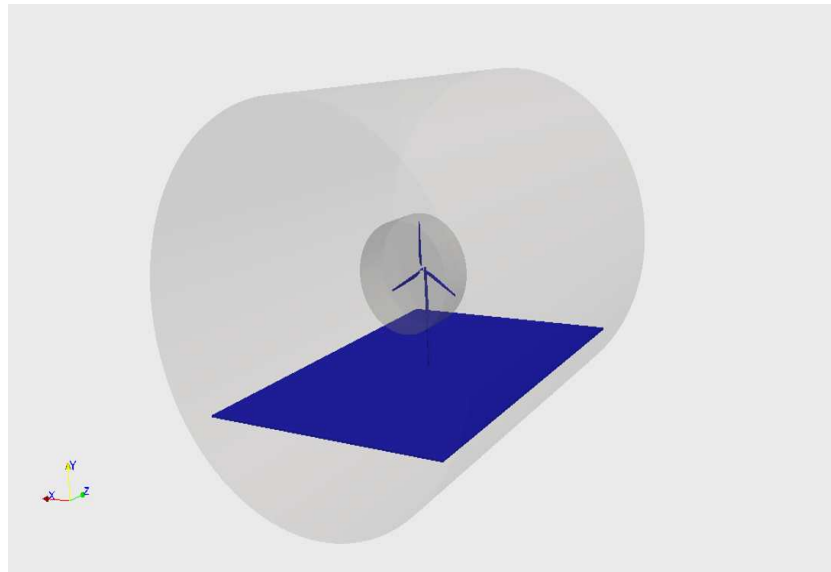


Figure 4.4. 3 dimensional setup of full wind turbine simulations case. Domain is modeled with wind turbine rotor, hub, tower, and ground.

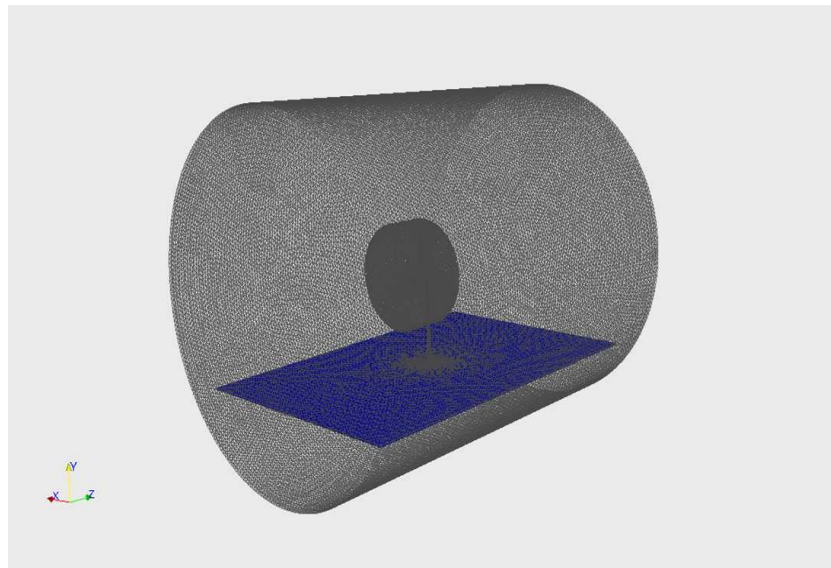


Figure 4.5. Full wind turbine assembly meshed with tetrahedral elements.

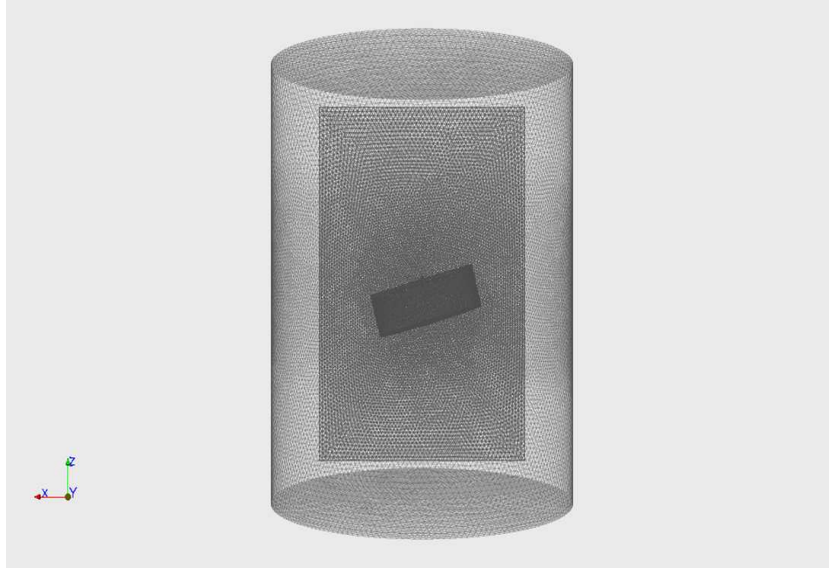


Figure 4.6. Tetrahedral mesh around the 3D wind turbine model with the rotor yawed at an angle of 15 degrees.

care is taken to assure that the mesh quality throughout the domain is good enough so as to have minimum skewed elements (with cell aspect ratio > 0.8). Meshes for performing the yawed rotor simulations also consisted of about 4 million tetrahedral cells. Figure 4.6 shows the mesh with the rotor with yaw angle of 15 degrees.

Working on these meshes required a lot of time and computational effort. This was because of the large dimensions of the domain and the wind turbine parts needed a lot of mesh points to mesh it entirely. Also, it was required to have a fine mesh in the domain downstream the rotor while a fairly coarse mesh over the rotor, hub, and the tower so as to save a considerable computational effort while running the code. It took about 3 days to build high quality meshes for each of these simulations.

4.2 Results

The results of the CFD simulations performed over the NREL blade as well as the full turbine geometry are discussed in the following sections. The steady state incompressible flow solver for multiple reference frames, MRFSimpleFoam was used

for carrying out simulations aimed towards validation of the NREL Phase VI experimental results. A detailed account of the comparison of the CFD results and the experimental results is given in the form of graphs and figures. On the other hand, a transient state solver, capable of moving the underlying mesh, turbDyMFoam was used in order to carry out the flow simulations over full wind turbines. These case studies are aimed towards understanding the aerodynamic effects of wind flow past a wind turbine.

4.2.1 CFD Simulations of the NREL Phase VI Experiment

The steady state solver MRFSimpleFoam is used to carry out simulations of flow over the NREL wind turbine blade. Geometric modeling of the blade is done using GAMBIT as explained in the previous sections. The goal of these simulations is to match the CFD results with the previously established NREL wind tunnel test results. Therefore, in accordance with the NREL experiment, a 5 m long blade is rotated at a constant rate of rotation of 72 rpm inside a computational domain. With this setup, simulations are run for 5 different inlet wind speeds viz. 7, 10, 13, 15, 20, and 25 m/s. Also, the blade is pitched all together by an angle of 4 degrees, which is also in accordance with the NREL experimental setup. Simulations are performed in a wedge shaped computational domain (120 degree periodicity) with periodic boundary conditions applied to the side faces of the domain. The numerical setup of the case is shown in the figure 4.7. During the simulations, the surface of the blade is maintained at 0 velocity by applying a no-slip, no-penetration boundary condition. Similarly the top wall of the domain is set as slip wall, while the inlet and outlet are set to velocity inlet and pressure outlet boundary conditions respectively. The solver employs the Spalart-Allmaras turbulence model to deal with the turbulent flow over the blades. A considerable amount of computational effort and time is saved by decomposing the

computational domain into 16 sub domains and running the code in parallel on 16 processors.

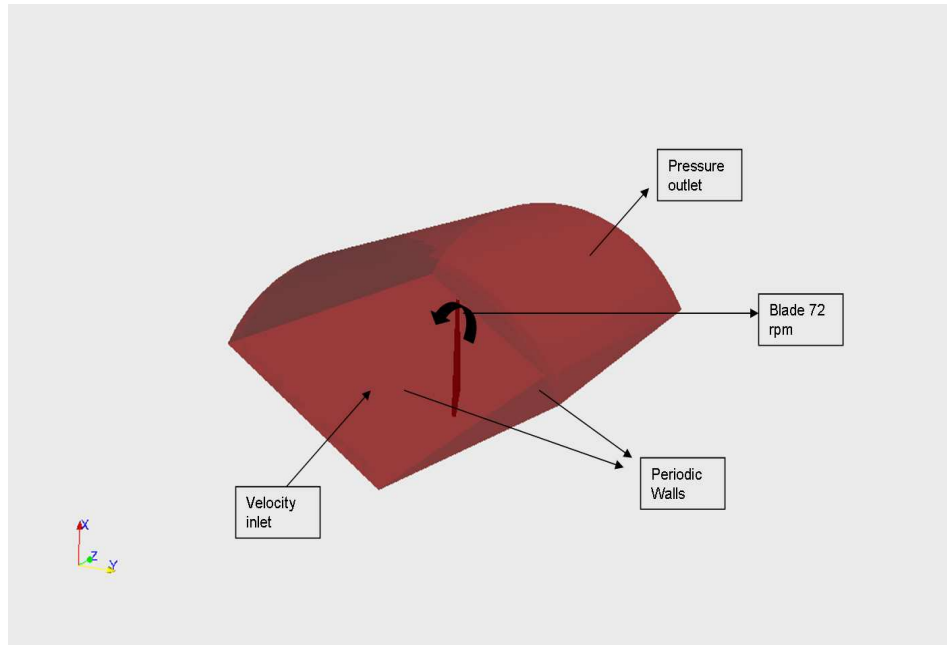


Figure 4.7. Numerical setup along with boundary conditions for the NREL Phase VI simulations.

Similar CFD studies on the NREL Phase VI experiment have been done by many researchers previously. The results from this project are compared to some of the results from [33], [7], [12]. While the methodologies, the CFD code, and its parameters used by these researchers may not be the same as that used in this work, the comparison of results gives a general idea of the ability of CFD codes to predict wind turbine aerodynamics. The first step towards the validation / comparison is taken by calculating the aerodynamic rotor power and comparing with the experimental values. Power is calculated by integrating pressure over the blade surface to find force, taking cross product of force vector with the radius vector to find torque and finally taking the dot product of torque with angular velocity. The figure 4.8 shows the values of power in kilo-Watts (kW) at different inlet wind speeds alongside the results from other sources.

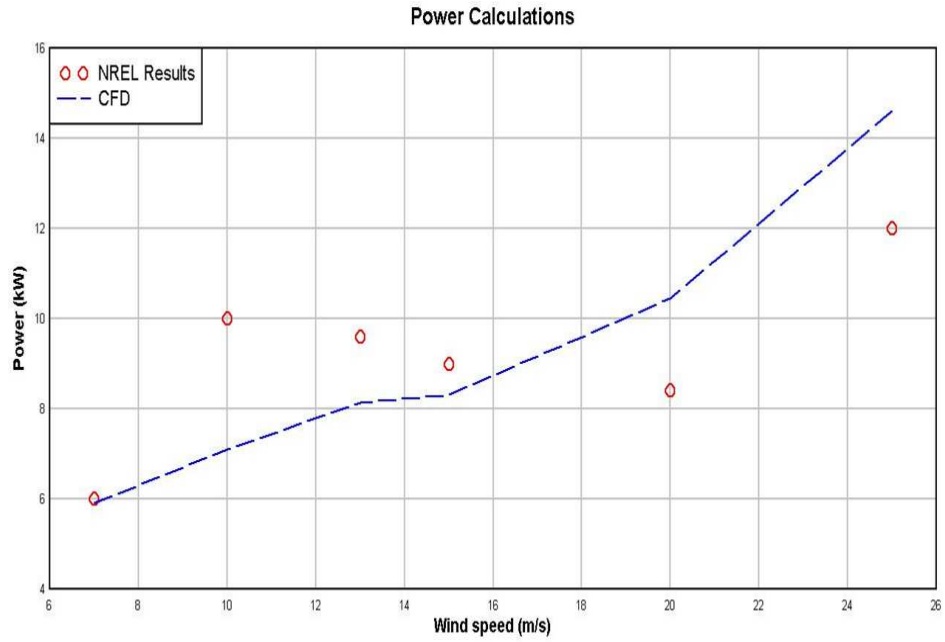


Figure 4.8. Comparison of power values between the CFD values and the NREL experimental values.

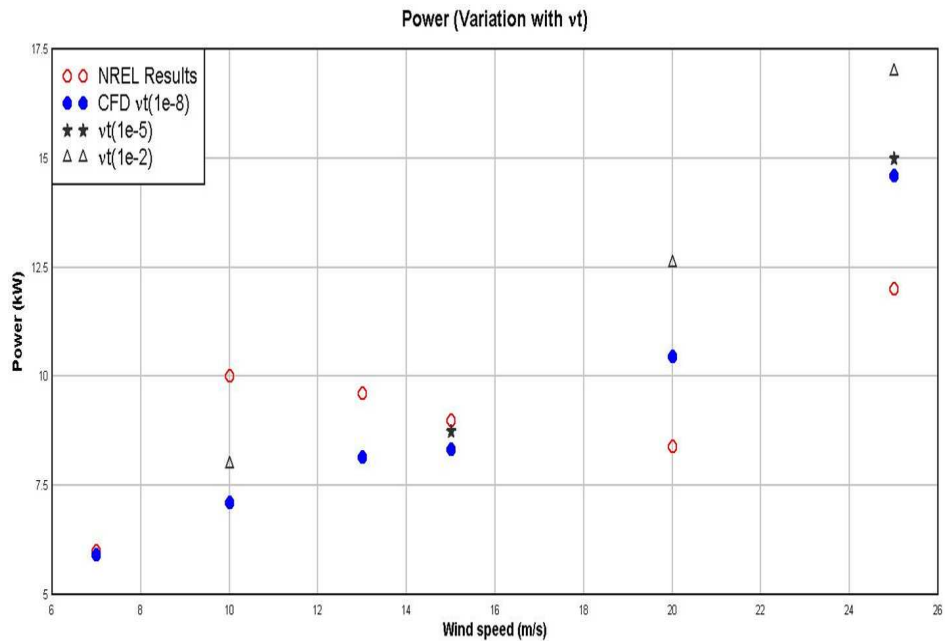


Figure 4.9. Variation of the rotor power with the variation of inlet eddy viscosity (changing turbulence parameters).

From the graph it is clear that the values obtained by the CFD code match to some extent with the experimental results. There is still a bit of discrepancy between the values from the current project and the experiment. This difference between the results is likely a result of the turbulence model used in the solver. Flow over the airfoil has many complicated factors such as separation, and the stall associated with it. The CFD code used for solving such problems should be capable of resolving all such aerodynamic effects. In this project, many efforts were made in order to achieve a closer match between the numerical and the experimental values. Boundary layer meshing was done in order to capture separation and predict stall. Similarly, a reasonable turbulence model for aerodynamic applications, the Spalart-Allmaras turbulence model, was used in the solver. However, this is a one-equation turbulence model and can only capture limited turbulence physics. In order to investigate the effects of the turbulence model on flow over the blades, the inlet turbulent viscosity (ν_t) was varied and power calculations were re-done for some cases. It is observed that for very high values of ν_t , higher power is achieved while for lower ν_t values, power values decrease. Therefore, it can be concluded that the turbulence model plays a very important role in governing the results obtained from the simulations. The figure 4.9 shows variation in the power values for different inlet wind speeds for different values of inlet ν_t .

After the calculation of power, further validation was done by calculating the pressure distribution over the surface of the blades. The coefficient of pressure is calculated over the top and the bottom surface of the blade. The coefficient of pressure is calculated using equation 4.1.

$$C_p = \frac{P - P_\infty}{0.5 * \rho(U_\infty^2 + (r^2)\Omega^2)} \quad (4.1)$$

P is the pressure measured along the blade section at different chord locations, U_∞ is the inlet velocity for the particular simulation, and Ω is the rotation rate i.e.

7.54 rad/s. Measurement of C_p is done at different sections along the blade. Thus, the numerical values of C_p are then compared with the experimental values which were obtained by calculating pressure over the blade surface with the use of pressure taps [13]. Measurement of C_p is done at three different radial sections along the blade viz. 30%R, 50%R, and 95%R representing the hub, mid section, and the tip of the blade (where $R = 5\text{m}$). The figures 4.10, 4.11, and 4.12 represent C_p calculated at these sections along the blade for 13 m/s case.

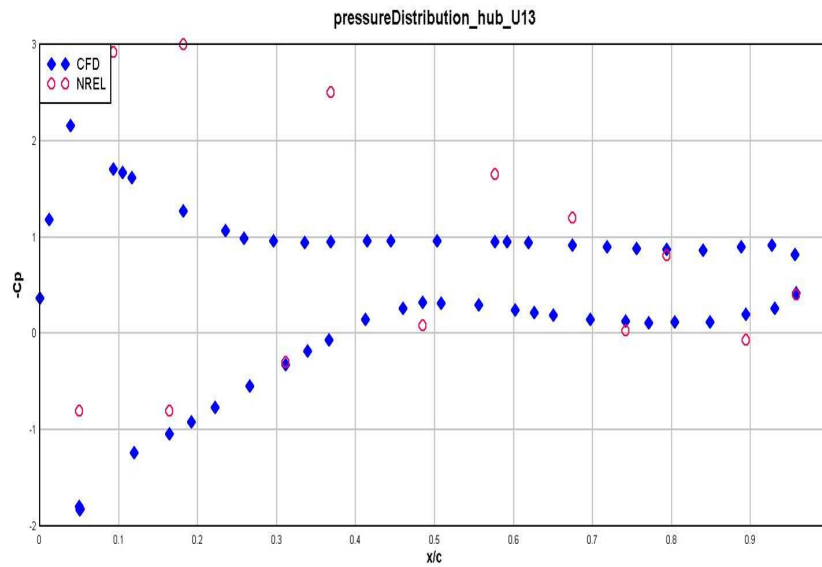


Figure 4.10. Co-efficient of pressure at 30%R blade location for 13 m/s inlet wind speed.

From the graphs it is observed that the pressure values obtained by CFD solver do not match with the experimental values. A substantial amount of variation is seen in the pressure values on the top surface of the blade while the pressure at the bottom blade surface matches quite well with the experimental results in all the cases. The top surface of the blade experiences a massive separation and the current turbulence model seem to be unable to resolve it correctly, therefore variation is observed in the pressure values comparison over the top surface. Additional C_p plots for all other inlet wind speeds are also attached in Appendix E. A thorough observation of these

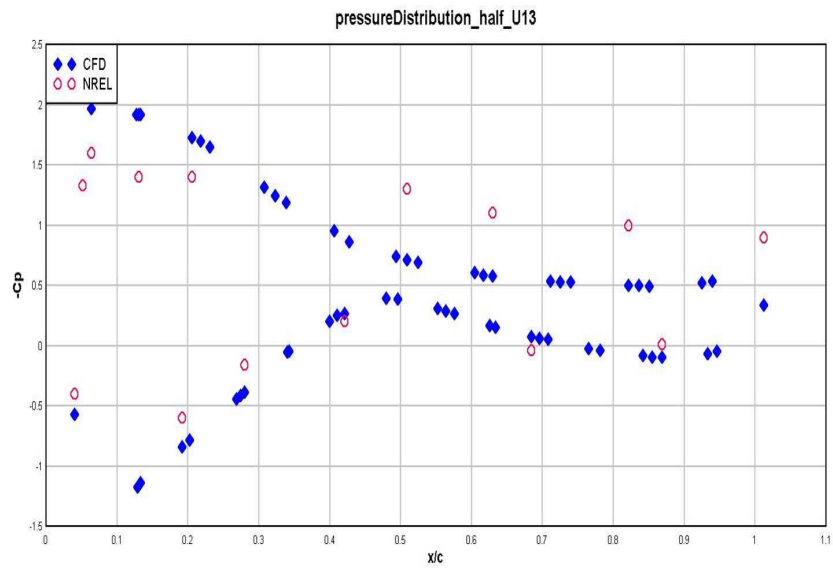


Figure 4.11. Co-efficient of pressure at 50%R blade location for 13 m/s inlet wind speed.

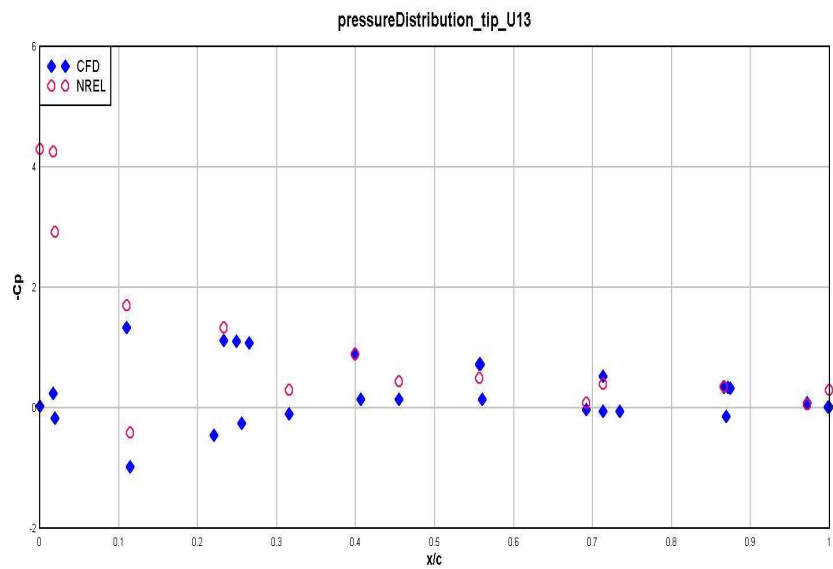


Figure 4.12. Co-efficient of pressure at 95%R blade location for 13 m/s inlet wind speed.

plots gives an explanation of the variations seen in the power calculations in figure 4.8. Of all the sections along the blade, the tip section experiences the maximum torque and thus plays a major role in determining the overall rotor power. If the C_p plots at the tip section are followed in detail, it is seen that, for high wind speed cases, the pressure values obtained from the CFD solver are constantly higher than the NREL values. Similarly, for the lower wind speed cases, it is seen that the CFD pressure values over the top surface of the blade are lower than the NREL values. Therefore, this gives an explanation for all the variations observed in the rotor power calculations.

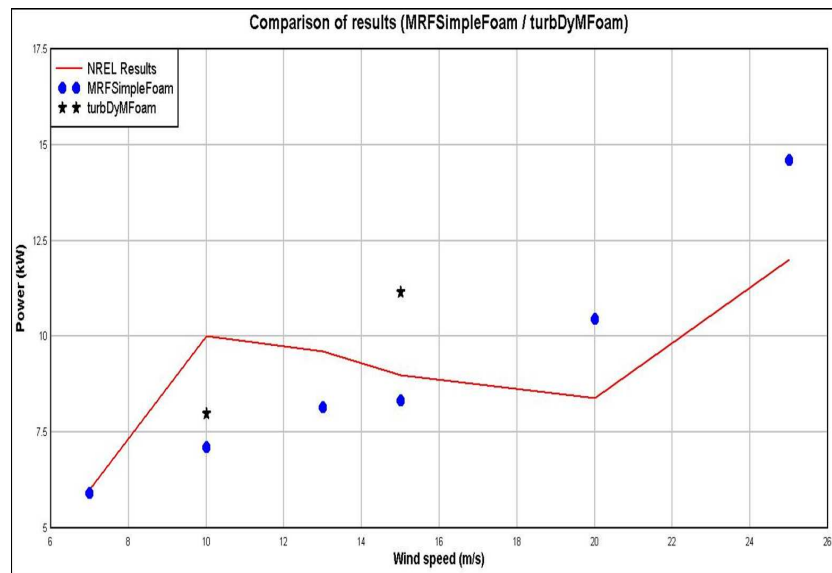


Figure 4.13. Comparison of power values obtained by MRFSimpleFoam and turbDymFoam simulations.

Thus, with the help of the graphs showing comparison between CFD and experimental values, validation of the current work is done. It is to be noted that the numerical results for the NREL Phase VI experiment were obtained using the steady state MRFSimpleFoam solver. In the later part of the project, a transient state solver, turbDymFoam, is also used for simulations involving wind flow over full turbine. This made us curious to calculate some of the above parameters using turbDymFoam and

validate the results obtained from the steady state solver. The figure 4.13 shows the comparison between the power values obtained by two different solvers. Due to the enormous mesh size involved in the simulations, all the cases could not be completed using turbDyMFoam. There is a considerable variation in the values obtained by both the solvers. Simulations with turbDyMFoam are performed with inlet ν_t equal to $1E^{-05}$. Power values would decrease further if the inlet ν_t is further reduced.

4.2.2 Wind Flow Simulations over Full Wind Turbine

After the validations of the NREL Phase VI test using the CFD solvers was done, the focus of the project shifted towards the completion of the last phase i.e. simulations of wind flow over a full wind turbine. The main goal of performing these simulations is to have some understanding of the flow physics past a wind turbine rotor in presence of a tower. The effects of the presence of tower on the downstream wake generated by the blades will be visualized in these simulations. These cases also test the implementation of turbDyMFoam, a moving mesh solver, and Generalized Grid Interface (GGI) for solving flow problems involving huge meshes and complex aerodynamics. Moreover, through these simulations an effort is made to look at the comprehensive picture of flow in different wind turbine operating conditions. Flow is simulated for two cases: *a*) when the rotor is aligned perpendicular to the inflow wind direction and *b*) when the rotor is at an angle of 15 degrees. The case setup replicates some of the actual operating conditions of an actual wind turbine.

4.2.2.1 No Yaw Case

The angle at which a wind turbine rotor is aligned to the incoming flow is called the Yaw angle. In the process of erecting wind turbines in a wind farm, first a detailed assessment of the wind flow over that site is done. Depending on the findings of the assessment, a mean wind flow speed and direction is established for that site. Depending on that prevailing wind direction all the turbines are aligned. Now some-

times, during the operation, there may be maximum wind from a certain different direction. Therefore, in order to extract the maximum energy out of the incoming wind, the orientation of the wind turbine rotors is changed so that they face the maximum incoming wind. This operation of changing the orientation by turning the rotor about the tower is called the Yaw operation and the angle to which the rotor is moved is the Yaw angle.

Simulations are performed over a full wind turbine setup with 0 degree rotor yaw. `turbDyMFoam`, a transient state solver for incompressible turbulent flows is used for performing simulations. This solver is capable of moving the underlying mesh with some angular speed specified by the user. In all the previous simulations, `MRFSimpleFoam` was used as the underlying CFD solver. Due to the inability of solver to resolve the flow field properly in presence of rotating as well as non rotating elements, there was a need for a moving mesh approach. Therefore, `turbDyMFoam` was used for simulations.

Wind flow was simulated for 2 inflow wind speeds, 10 m/s and 15 m/s. The domain is made up of two non conformal meshes. The outer mesh hosts the stationary elements while the inner mesh hosts the rotating parts of the geometry. Inlet and outlet of the domain are set to velocity and pressure boundary conditions respectively. The outer wall of the domain is set as a slip wall. Walls of the rotating mesh and the inner walls of the stationary mesh are set to Generalized Grid Interface (GGI) boundary conditions. Rotor and hub surfaces are set to 0 velocity (or wall boundary conditions). The rotating mesh or the rotor zone is rotated at an angular velocity of 72 rpm. The Spalart-Allmaras turbulence model is used for modeling the turbulent eddy viscosity in the flow field during the simulations. The mesh consists of 4 million tetrahedral elements. In these simulations, comparisons with NREL values were not made. Therefore, to limit the number of cells in the mesh, boundary layer meshing

was not done over the blade surfaces. In order to save considerable amount of time, the code was run in parallel on 32 processors on a supercomputer cluster.

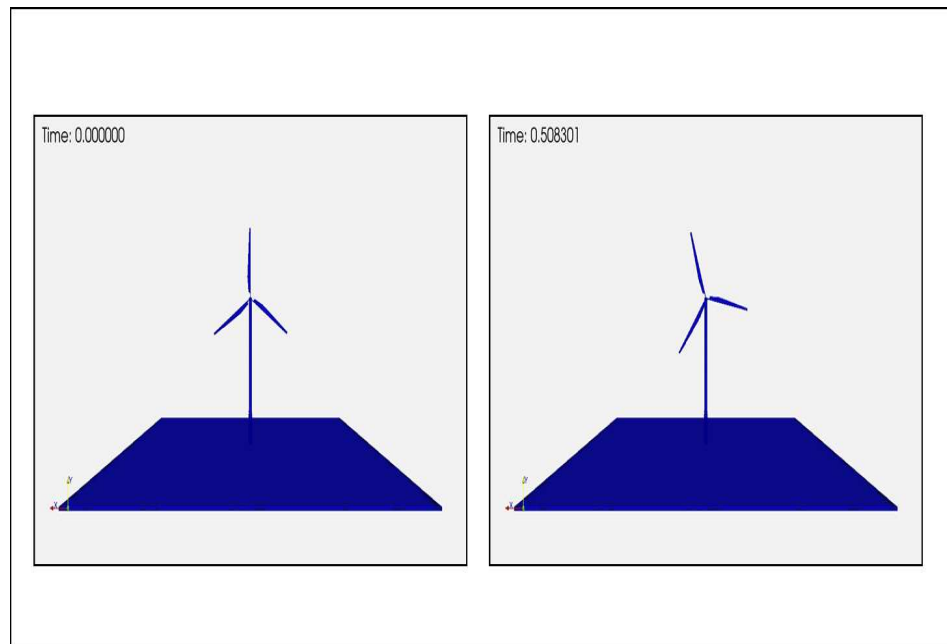


Figure 4.14. Position of wind turbine rotor at 2 different time steps indicating a correct implementation of the moving mesh algorithm in the solver.

Figure 4.14 shows the position of wind turbine rotor at different times. The turbine blades are rotated to 215 degrees in 0.508301 seconds. This indicates that the rotor is indeed rotating and thus validates the working of moving mesh part of the code. The flow field in the domain is also examined in order to study the wake downstream the wind turbine. Figure 4.15 shows a slice across the domain showing the streamwise velocity contours. The wake coming off the rotor and the turbine tower is visible clearly. It is evident that the presence of tower has a strong effect on the velocity field in the wake downstream and there is a considerable amount of reduction in wind speed in the wake. Streamwise vorticity is also plotted in order to visualize the vortices coming off the blade and in the domain downstream of the rotor. Figure 4.16 shows a instantaneous vorticity iso-surface plot. Vortices coming off the blade tips are clearly visible in the figures. Also, we can see that the wake

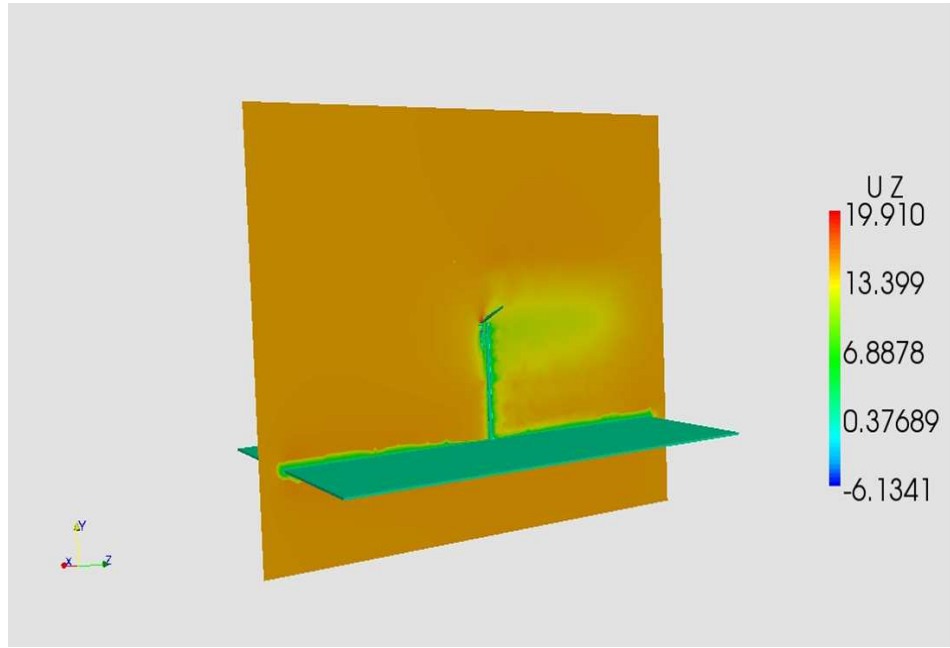


Figure 4.15. Stream wise velocity in the computational domain. $U_{\infty} = 15m/s$.

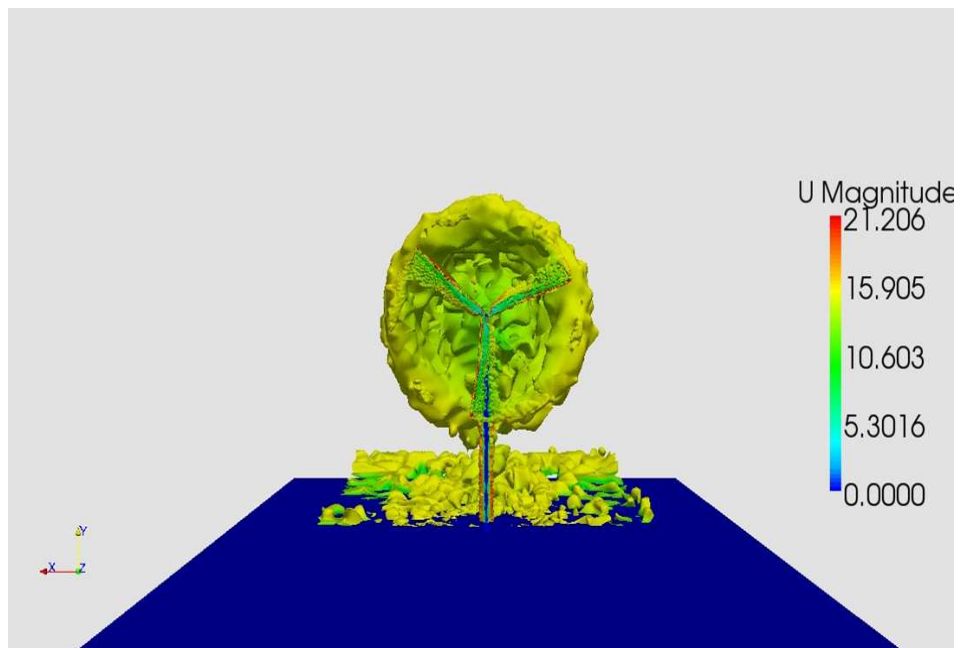


Figure 4.16. Iso surface plot of stream wise vorticity colored by velocity magnitude representing the wake off the rotor blades. $U_{\infty} = 15m/s$.

is symmetric about the rotor. As we move towards the outlet of the domain, the vortices start to diffuse.

4.2.2.2 15 degree Yawed Rotor

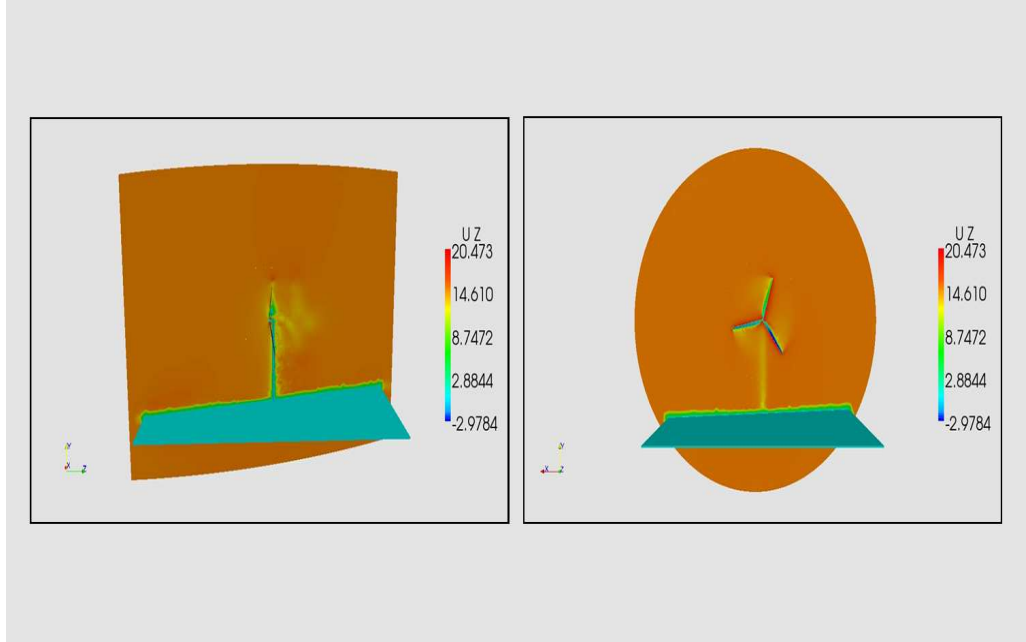


Figure 4.17. Contours of streamwise velocity for 15° yaw. Effect of the rotor yaw is visible in the wake downstream. $U_\infty = 15m/s$.

Similar simulations are also performed over a yawed rotor at an angle of 15 degrees. The initial setup of the case is similar to the 0 yaw case as seen in the previous section. The only difference is that, the wind turbine rotor is collectively rotated about the vertical axis by an angle of 15 degrees. This is done in order to study the effect of rotor yaw on the flow behavior around the wind turbine.

Velocity contours in the domain are plotted in the figure 4.17. Also, as expected the wake in this case is asymmetrical due to 15 degree yaw angle. The vorticity coming off the tip of the blades is shown in the figure 4.18. It is also observed that downstream of the tower, the vortices coming off the blade tips gets chopped at the tower location due to the presence of the tower. The vortices in the domain are further

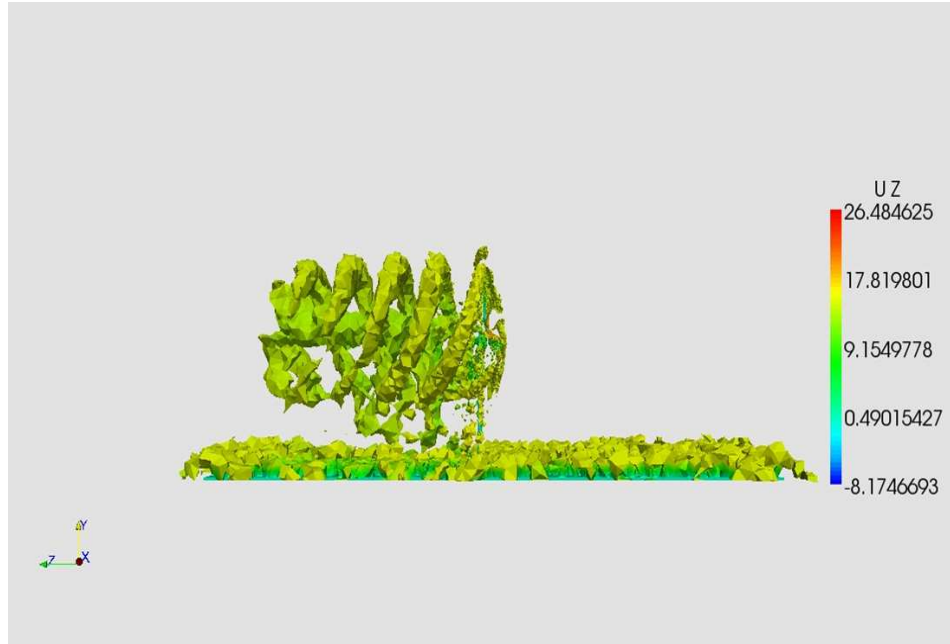


Figure 4.18. Iso-surfaces of vorticity colored by stream-wise velocity for 15° yaw. $U_\infty = 15m/s$.

refined by calculating Q , second invariant of velocity gradient. Q is calculated using the equation 4.2 [30].

$$Q = 1/2 * (\Omega^2 - S^2) \quad (4.2)$$

where,

$$\Omega = 1/2 * (\nabla U - \nabla U^T)$$

$$S = 1/2 * (\nabla U + \nabla U^T)$$

This helps plotting the vortices only in the direction of the flow. Thus only the dominant vortical flow structures are resolved in detail. The iso-surfaces of Q are shown in figure 4.19.

4.2.3 Simulations with Shear in the Rotor Plane

Wind turbines operate under an atmospheric boundary layer. Because of the no slip condition imposed on the ground, there is a gradient in the wind flow above the

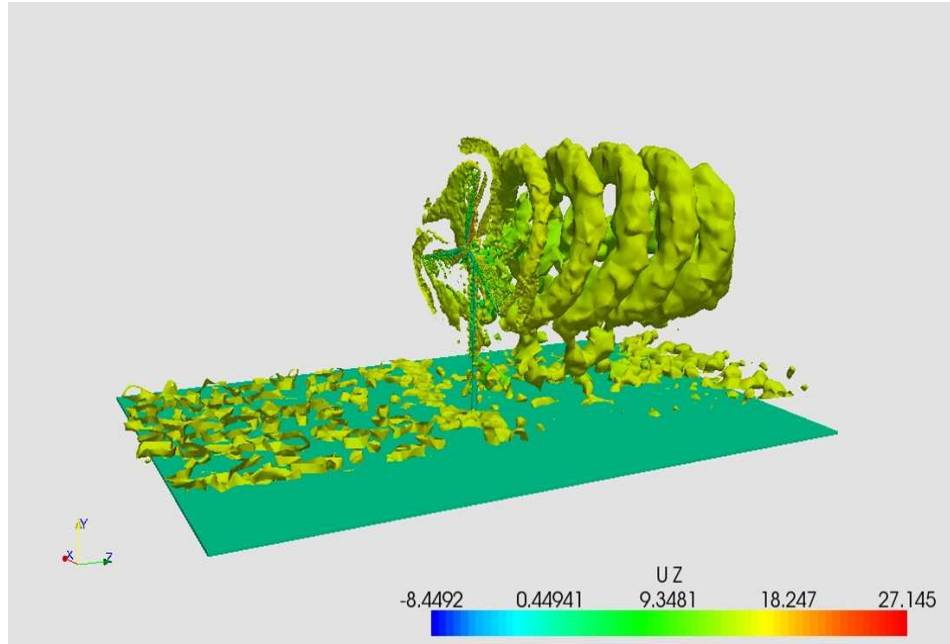


Figure 4.19. Iso-surfaces of Q colored by stream-wise velocity for 15° yaw. $U_\infty = 15\text{m/s}$.

earth's surface. As the height increases, the wind speed increases from zero to free stream value. The velocity of the wind in this layer generally follows a logarithmic profile. Therefore, due to this, the wind turbine rotor constantly experiences some wind shear during its operation. This becomes a major aspect in wind farm siting and suitability analysis for wind turbine placement. In all the previous simulations, the inlet wind speed was assumed to be perpendicular to the wind turbine rotor only i.e. $U = U(z)$ only. In this section an effort is made to simulate wind flow considering a shear profile in the rotor plane.

A schematic representation of the case to be simulated is shown in the figure 4.20. In this case, the computational domain is initialized with a free stream velocity value (e.g. 15 m/s). Inlet of the domain needs to be set with a velocity varying in the linear direction (shear profile) as well, such that $U = U(y, z)$. There is no such boundary condition already present in the OpenFOAM library. Therefore, a new boundary condition, called **shearVelocity** was written and implemented in the

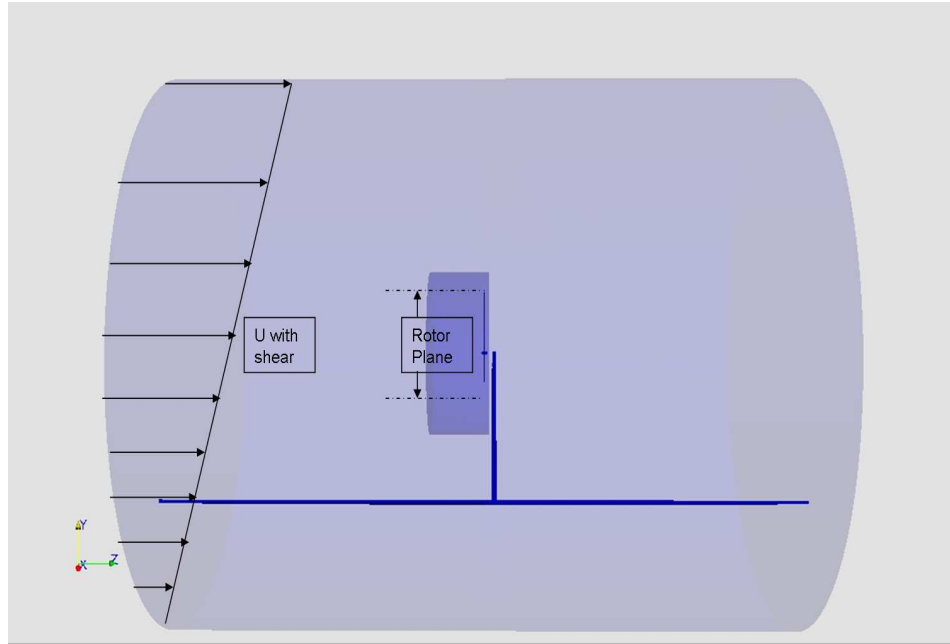


Figure 4.20. Schematic representation of simulation of flow over wind turbine with a shear profile.

solver. The inlet of the domain is set with the *shearVelocity* boundary condition, which is represented below.

```
inlet
{
type shearVelocity;
n (0 0 1); //Flow Direction Axis
y (0 1 0); //Vertical Axis in the domain
Value 15; //Velocity value
}
```

where,

vector n - the flow direction in the domain,

vector y - the vertical direction, and

scalar $Value$ - velocity (m/s) value in the computational domain.

With the use of these three values, the inlet velocity vector representing shear is calculated according to the equation 4.3.

$$U = n \star Value \star (((faceCtr - boxCtr) + boxMax) \times y) \quad (4.3)$$

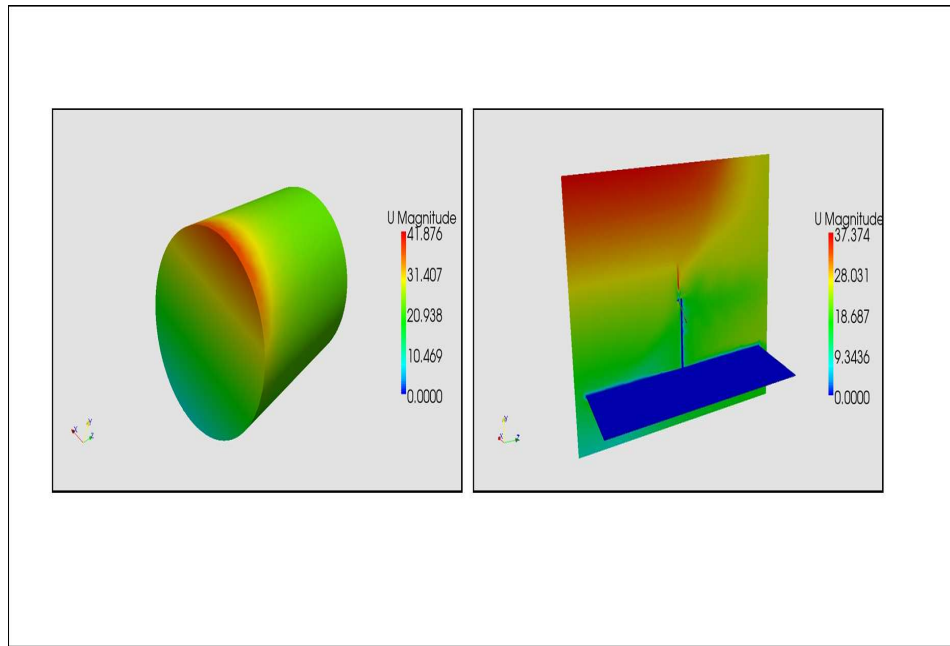


Figure 4.21. shearVelocity boundary condition applied to the inlet of the domain. Velocity contours in the domain. $U_{\infty} = 15m/s$.

The implementation of the boundary condition on the inlet of the domain is shown in the figure 4.21. From the figure it is clear that the velocity increases linearly from the ground to the top wall of the domain. Thus in this way the case is set such that the wind turbine rotor experiences some shear. Such analysis is helpful because this would help analyzing the shear exponent at various sections (i.e. bottom, hub, and top) of the wind turbine rotor. Simulations are then run with the same previous

settings, but with the new inlet boundary condition. Velocity profile in the domain is shown in figure 4.21. Shear exponent (α) in the rotor plane is calculated by the equation 4.4.

$$\frac{u_{y2}}{u_{y1}} = \frac{h_{y2}^\alpha}{h_{y1}^\alpha} \quad (4.4)$$

where,

u - velocity (m/s)

h - height above the ground (m)

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Summary and Conclusions

MRFSimpleFoam / DU rotor:

The results presented in the previous chapters validate the working of the CFD code MRFSimpleFoam. Also, the rotating cylinder case demonstrates the successful working of the MRF part of the code. The code can now be used to simulate the flow over rotating objects. Preliminary tests on the Delft University wind turbine blade were carried out. Power calculations were made and compared to the experimental wind tunnel results from the NREL Phase VI blade [33]. The results did not match exactly with each other but they set a trend and were in the comparison range. The reason for not getting the exact results was the basis of comparison of blades made up of all-together different airfoils. Before working on the actual blade from the NREL experiment, some more work needs to be completed on the Delft blade. Therefore, as expected, close but not totally accurate results were obtained from these simulations over DU blade.

NREL Phase VI Experiment:

This was the basis of performing CFD simulations and numerical validation of the NREL Phase VI experiment. The airfoil tables for the S809 airfoil were studied and the NREL Phase VI blade was modeled according to the specifications given in the NREL technical report. An extensive use of 3 dimensional modeling and meshing software GAMBIT was made throughout this project. RANS simulations with Spalart-Allmaras turbulence model were performed with different inlet wind speeds

over the modeled wind turbine blade. Power values were calculated and compared with the experimental results. The comparison fared well but the discrepancies in the results still lasted. These were resulted to the use of an appropriate turbulence model to predict the aerodynamics associated with the blade correctly. Co-efficient of power was also calculated at different blade locations and compared with the NREL results.

turbDyMFoam / Full turbine simulations:

Some simulations were also run for wind flow cases over full wind turbine with an aim to understand the flow phenomenon over rotor as well as tower. Rotor was yawed at 0 degrees and 15 degrees for the simulations. Wind speed and vorticity in the domain were visualized. These simulations validate the dynamic mesh application of the solver turbDyMFoam. Also, capability of GGI in performing huge CFD simulations with big meshes is highlighted in these results. It is proved that the GGI algorithm, if coupled with an appropriate solver and used on supercomputers, can be successfully used to simulate flow in any such applications involving big rotating geometries.

In the end this project gave a thorough understanding of the complex aerodynamics associated with the wind flow over wind turbines. A fine grasp over 3 dimensional modeling and meshing was achieved through this project. Also, the basics of object oriented programming and its implementation in OpenFOAM solvers for CFD applications were learned. A very important and useful concept of parallel programming using supercomputers was used throughout this project.

5.2 Future Work

The results of the NREL test simulations matched to some extent to the previously established values. Turbulence models play very important role in governing the solution in highly turbulent flow cases. The current turbulence model used in this

project is well validated for aerodynamic applications but still fell short in resolving the highly aerodynamic and turbulent flow over wind turbine blades. Separation over the blade surface could not be captured accurately and in turn the blade stall was not predicted correctly. Thus, there is a need of development of a highly accurate turbulence model incorporating these features so that such applications can be simulated precisely using CFD.

The turbDyMFoam solver along with the GGI boundary conditions is a perfect combination for accurately simulating huge moving mesh applications. But even with the presence of supercomputers for running the code in parallel, there are still some major issues regarding the performance of the solver in parallel. A further refined GGI algorithm can surely make the code go faster and can be efficiently implemented for huge turbo-machinery applications.

Others might use this CFD framework to carry out further wake analysis and tower modeling. Wake behind a wind turbine may considerably hamper the operation of another wind turbine situated in the wake. Hahm et al. [38] have studied the wind turbine wake characteristics for different wind speeds and established a close match between their CFD results and the experimental values. With the successful implementation of moving mesh CFD codes, simulations with a further broader perspective now seem possible. With the availability of high performance computers, a combined mesh consisting of 2 to 3 turbines can now be generated and used for simulations. This will help in visualizing the effects of one turbine onto the other one. Similar studies in areas of wind flow modeling and analysis could be expected by others in future using the current work.

APPENDIX A

THE ALGORITHM FOR SIMPLE LOOP.

The Semi Implicit Method for Pressure Linked Equations (SIMPLE) is a method which is used to solve the Navier Stokes equations iteratively. The algorithm used in incorporating SIMPLE loop in the CFD code is given below.

The SIMPLE Algorithm:

1. Set the boundary conditions.
2. Discretize and solve the momentum equation to calculate the velocity field.
3. Calculate the mass fluxes.
4. Solve the pressure equation and apply under relaxation.
5. Apply correction to the mass fluxes.
6. Apply correction to the velocity field.
7. Update the boundary conditions.
8. Repeat till convergence.

APPENDIX B

MULTIPLE REFERENCE FRAMES: TOOLS AND UTILITIES

MRFSimpleFoam is primarily used for problems dealing with rotating geometries. Operation on zones is mainly done by accessing the *MRFZones* file. The *MRFZones* file used in the MRFSimpleFoam is given below.

Sample *MRFZones* file:

```
(  
rotor //Name of the rotating zone  
{  
patches (rotor); //Name of the rotating patch  
origin [0 1 0 0 0 0 0] (0 0 0); //Position of the origin  
axis [0 0 0 0 0 0 0] (1 0 0); // Rotating about X axis  
omega [0 0 -1 0 0 0 0] -7.54; // Angular speed (rad/s)  
}  
)
```

APPENDIX C

IMPLEMENTATION OF GENERALIZED GRID INTERFACE (GGI) IN THE CFD SOLVER

For the detailed understanding of the GGI and the moving mesh part of the code, *boundary* and *dynamicFvMeshDict* files are attached below. The physical properties of the GGI patches are set in the boundary file while the information relevant to dynamic mesh part of the code is specified in the dynamicMeshDict file.

Boundary file:

```
ggiInside //Master GGI Patch
{
type ggi; //type GGI
nFaces 24140; //number of faces in the patch
startFace 3259791; //starting face number
shadowPatch ggiOutside; //shadow patch
zone ggiInside_zone; //name of the face zone
bridgeOverlap false; //true if the meshes overlap each other
}

ggiOutside //Shadow GGI Patch
{
type ggi;
nFaces 3908;
startFace 3283931;
shadowPatch ggiInside;
```

```
zone ggiOutside_zone;
bridgeOverlap false;
}
```

dynamicMeshDict file:

```
dynamicFvMeshLib "libtopoChangerFvMesh.so"; //Dynamic Mesh Library
dynamicFvMesh mixerGgiFvMesh; //Code
```

```
mixerGgiFvMeshCoeffs
{
    coordinateSystem
    {
        type cylindrical;
        origin (0 0 0);
        axis (0 0 1);
        direction (0 1 0);
    }
    rpm 72; //Angular velocity of rotation
    slider
    {
        moving (ggiInside); //Moving Patch
        static (ggiOutside); //Static Patch
    }
}
```

APPENDIX D

S809 AIRFOIL DATA FOR THE DESIGN OF NREL PHASE VI EXPERIMENT WIND TURBINE BLADE.

Modeling for the NREL Phase VI experiment blade is done using the airfoil data which contains information about the co-ordinates of the S809 airfoil. Also, information about the sectional twist along the blade and the thickness of the blade is required in order to perform a three dimensional modeling of the wind turbine blade. These details are given in tables below. The S809 airfoil co-ordinates are taken from table D.1. With the help of these co-ordinates airfoil can be modeled accurately. Now, once we know the airfoil shape and the accurate geometry, using the table D.2 we can model the geometry of the blade consisting of the above modeled S809 airfoils. During this process, S809 airfoils with various chord lengths(m) at different radial locations(r) along the blade and the corresponding angle of twist($degrees$) are modeled and assembled together so as to make an entire blade surface. From the table D.2 it is evident that the blade length is 5.02 m .

Co-ordinates of the S809 airfoil representing the x and y positions around the chord and the details required for the three dimensional blade modeling are given in the tables below. The airfoil obtained after plotting these values is shown in figure D.1.

The MATLAB code used for converting the above airfoil into a surface mesh of *.stl* type is attached below.

Code 1: Airfoil

```
%Airfoil Coordinates
```

UpperSurface		LowerSurface	
x/c	y/c	x/c	y/c
0.00037	0.00275	0.0014	-0.00498
0.00575	0.01166	0.00933	-0.01272
0.01626	0.02133	0.02321	-0.02162
0.03158	0.03136	0.04223	-0.03144
0.05147	0.04143	0.06579	-0.04199
0.07568	0.05132	0.09325	-0.05301
0.1039	0.06082	0.12397	-0.06408
0.1358	0.06972	0.15752	-0.07467
0.17103	0.07786	0.19362	-0.08447
0.2092	0.08505	0.23175	-0.09326
0.24987	0.09113	0.27129	-0.1006
0.29259	0.09594	0.31188	-0.10589
0.33689	0.09933	0.35328	-0.10866
0.38223	0.10109	0.39541	-0.10842
0.42809	0.10101	0.43832	-0.10484
0.47384	0.09843	0.48234	-0.09756
0.52005	0.09237	0.52837	-0.08697
0.56801	0.08356	0.57663	-0.07442
0.61747	0.07379	0.62649	-0.06112
0.66718	0.06403	0.6771	-0.04792
0.71606	0.05462	0.72752	-0.03558
0.76314	0.04578	0.77668	-0.02466
0.80756	0.03761	0.82348	-0.01559
0.84854	0.03017	0.86677	-0.00859
0.88537	0.02335	0.90545	-0.0037
0.91763	0.01694	0.93852	-0.00075
0.94523	0.01101	0.96509	0.00054
0.96799	0.006	0.98446	0.00065
0.98528	0.00245	0.99612	0.00024
0.99623	0.00054	1	0
1	0	0	0

Table D.1. Airfoil Co-ordinates.

Distance r(m)	ChordLength(m)	Twist(degrees)	Thickness(m)
0	Hub-center	Hubcenter	Hubcenter
0.5083	0.218	0.0	0.218
0.6604	0.218	0	0.218
0.8835	0.183	0	0.183
1.0085	0.349	6.7	0.163
1.0675	0.441	9.9	0.154
1.1335	0.544	13.4	0.154
1.2575	0.737	20.04	0.154
1.343	0.728	18.074	20.95%chord
1.51	0.711	14.292	20.95%chord
1.648	0.697	11.909	20.95%chord
1.952	0.666	7.979	20.95%chord
2.257	0.636	5.308	20.95%chord
2.343	0.627	4.715	20.95%chord
2.562	0.605	3.425	20.95%chord
2.867	0.574	2.083	20.95%chord
3.172	0.543	1.15	20.95%chord
3.185	0.542	1.115	20.95%chord
3.476	0.512	0.494	20.95%chord
3.781	0.482	-0.015	20.95%chord
4.023	0.457	-0.381	20.95%chord
4.086	0.451	-0.475	20.95%chord
4.391	0.42	-0.92	20.95%chord
4.696	0.389	-1.352	20.95%chord
4.78	0.381	-1.469	20.95%chord
5	0.358	-1.775	20.95%chord
5.305	0.328	-2.191	20.95%chord
5.532	0.305	-2.5	20.95%chord

Table D.2. Blade Chord and Twist Configuration.

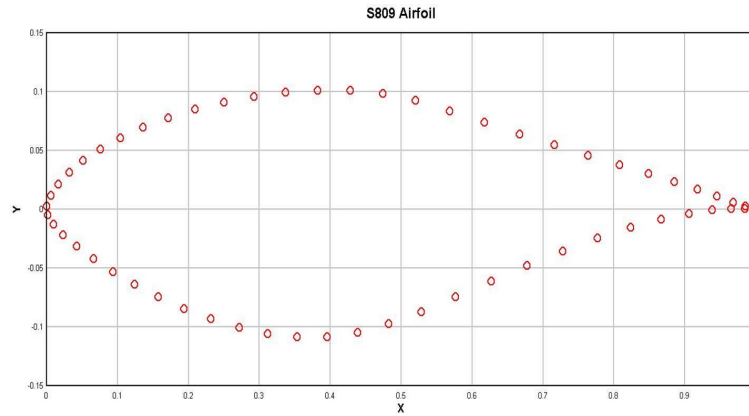


Figure D.1. S809 airfoil after plotting the x and y co-ordinates from table D.1.

```
function [Coordinates] = SectionSplinef(l)
format long

%read in airfoil data
CS = xlsread('airfoil.xls');
scale = xlsread('twist.xls','A2:D38');

%takes the length of blade desired and the given data for a different length, a
dr = scale;
dr(:,2) =scale(:,2)*l/max(scale(:,2));
dr(:,4) =scale(:,4)*l/max(scale(:,2));

%creates a 3 dimensional matrix that includes all coordinates at each cross
%section
for i=1:length(dr)
    Coor = dr(i,4)*CS;
    Coor(:,3) = dr(i,2);
    Coord(:,:,i) = Coor;
end
```

```

end

%takes the coordinates from above and alters it to include the twist angle
for j = 1:length(dr)
    for k = 1:length(CS)
        Coordinates(k,1,j) = Coord(k,1,j)*cosd(dr(j,3))-Coord(k,2,j)*sind(dr(j,3));
        Coordinates(k,2,j) = Coord(k,2,j)*cosd(dr(j,3))+Coord(k,1,j)*sind(dr(j,3));
        Coordinates(k,3,j) = Coord(k,3,j);
    end
end

end

fprintf

```

Code2: Blade Surface Mesh

```

% Meshing2

clc

%input the scaling factor into the Coordinate generating program
l = input('Input the total length of the blade, in meters, for scaling purposes');
Coordinates = SectionSplinef(l);

%start a new stl text file called airfoil
fid = fopen('airfoil2.stl','w');
fprintf(fid,'solid\n\t');

%creates a vector that goes around the airfoil cross section without any
%jumps or gaps.
Section = 1:31;
Section(32:61) = 61:-1:32;
Section

```

```

for j = 4:8      %36      %for all cross sections
    for i = 1:61      %for all points on the cross section
        k = i+1;
        if i == 61      %for the last point on the cross section
            k = 1;      %go back to the first cross section
        end
        A = Coordinates(Section(i),:,j);      %defines the coordinates of the four
        B = Coordinates(Section(k),:,j);
        C = Coordinates(Section(i),:,j+1);
        D = Coordinates(Section(k),:,j+1);
        a = B-A;      %defines the vectors from one point to the next
        b = C-A;
        normala = cross(a,b);      %defines the vector normal to the triangle
        fprintf(fid,'facet normal ');      %the following lines print the text
        fprintf(fid,'%f ',normala);
        fprintf(fid,'\n\t\t\touter loop\n');
        fprintf(fid,'\t\t\tvertex ');
        fprintf(fid,'%f ',A);
        fprintf(fid,'\n\t\t\tvertex ');
        fprintf(fid,'%f ',B);
        fprintf(fid,'\n\t\t\tvertex ');
        fprintf(fid,'%f ',C);
        fprintf(fid,'\n\t\t\tendloop\n\t\t\tendfacet\n\t');
        c = D-B;      %creates the second triangle and its normal
        d = C-D;
        normalb = cross(c,d);
    end
end

```

```

    fprintf(fid,'facet normal ');
    fprintf(fid,'%f ',normalb);
    fprintf(fid,'\n\t\touter loop\n');
    fprintf(fid,'\t\t\tvertex ');
    fprintf(fid,'%f ',B);
    fprintf(fid,'\n\t\t\tvertex ');
    fprintf(fid,'%f ',D);
    fprintf(fid,'\n\t\t\tvertex ');
    fprintf(fid,'%f ',C);
    fprintf(fid,'\n\t\tendloop\n\tendfacet\n');
end
end

fclose(fid);

fprintf('\nOpen the airfoil.stl file in GAMBIT to see the blade surface.')

```

APPENDIX E

CO-EFFICIENT OF PRESSURE PLOTS FOR DIFFERENT INLET WIND SPEEDS AT VARIOUS BLADE SECTIONS.

Co-efficient of pressure values for over the hub and tip sections of the blade for different inlet wind speeds are plotted below. A comparison is made between the CFD values and the NREL experimental values. In one case, comparison is also made with the numerical results from Duque et al. [7].

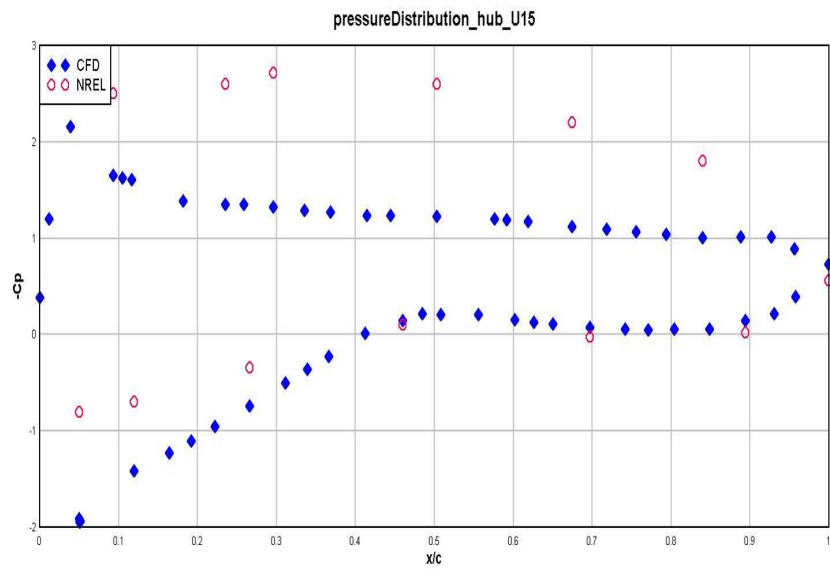


Figure E.1. Co-efficient of pressure at 30%R blade location for 15 m/s inlet wind speed.

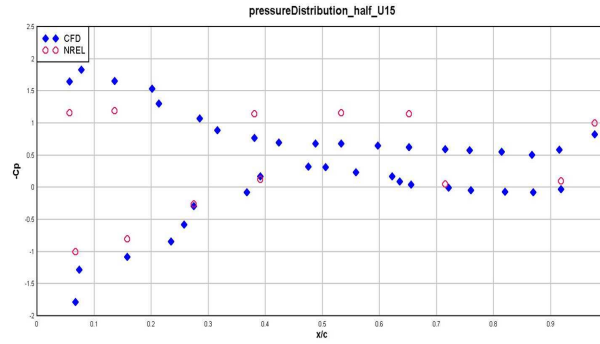


Figure E.2. Co-efficient of pressure at 50%r blade location for 15 m/s inlet wind speed.

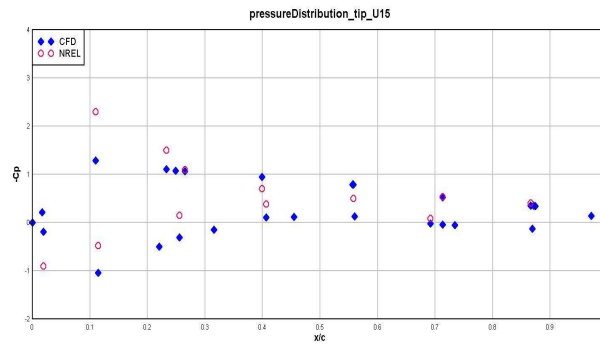


Figure E.3. Co-efficient of pressure at 95%r blade location for 15 m/s inlet wind speed.

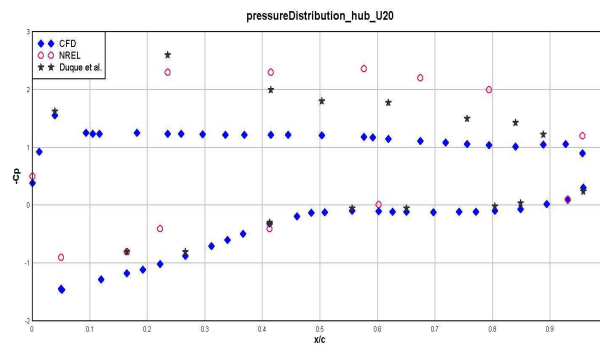


Figure E.4. Co-efficient of pressure at 30%R blade location for 20 m/s inlet wind speed.

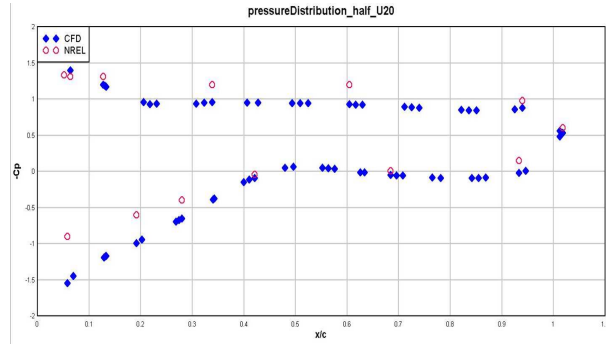


Figure E.5. Co-efficient of pressure at 50%R blade location for 20 m/s inlet wind speed.

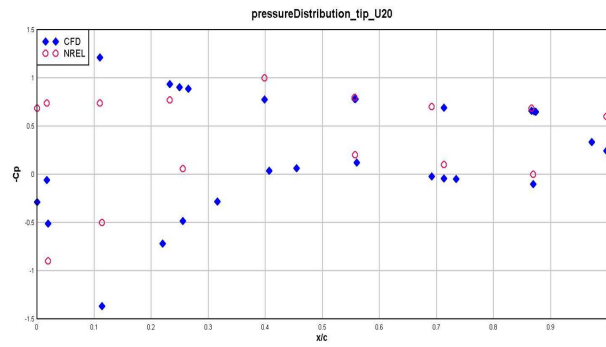


Figure E.6. Co-efficient of pressure at 95%R blade location for 20 m/s inlet wind speed.

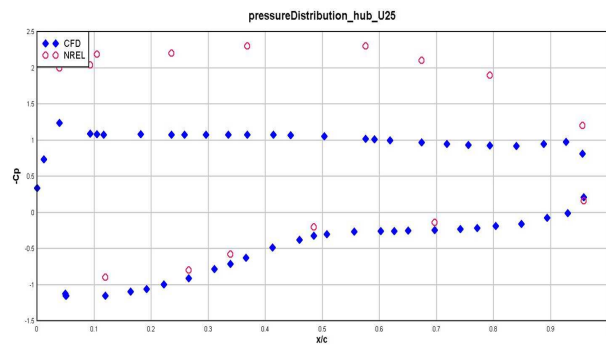


Figure E.7. Co-efficient of pressure at 30%R blade location for 25 m/s inlet wind speed.

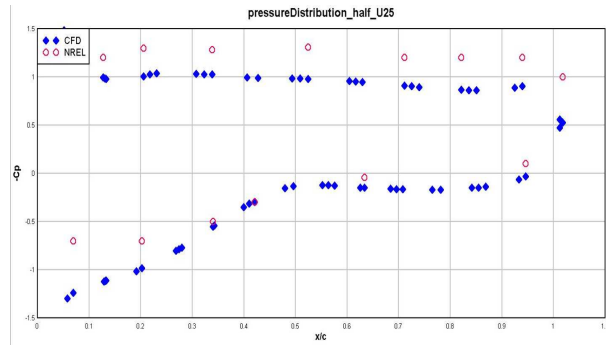


Figure E.8. Co-efficient of pressure at 50%R blade location for 25 m/s inlet wind speed.

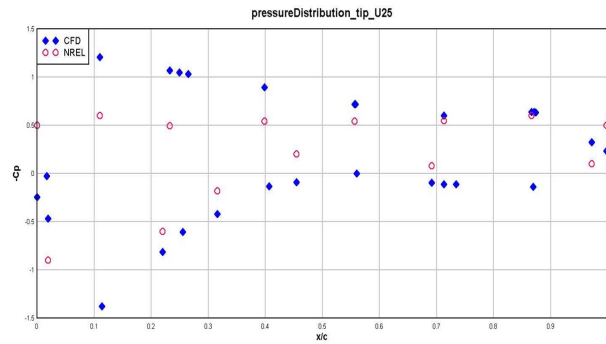


Figure E.9. Co-efficient of pressure at 95%R blade location for 25 m/s inlet wind speed.

BIBLIOGRAPHY

- [1] Anderson, J.D. *Computational Fluid Dynamics: The Basics with Applications*. McGraw-Hill, New York, NY, USA, 1995.
- [2] ANSYS. Gambit - Computational Fluid Dynamics Preprocessor from Fluent. <http://www.fluent.com/software/gambit/>, 2008.
- [3] Bardina, J. E., Huang, P. G., and Coakly, T. J. Turbulence Modeling Validation, Testing, and Development. Tech. rep., NASA Technical Memorandum, 1997.
- [4] Beaudoin, M., and Jasak, H. Development of a Generalized Grid Interface for Turbomachinery Simulations in OpenFOAM. *Open Source CFD International Conference 2* (2008).
- [5] Berg, Tobias, and Wikstrom, Anna. Fan Modeling for Front End Cooling with CFD. Master's thesis, Lulea University of Technology, 2007.
- [6] DOE, U.S. Fossil Energy. <http://www.fossil.energy.gov/index.html>, 2009.
- [7] Duque, Earl P. N., Burklund, Michael D., and Johnson, Wayne. Navier Stokes and Comprehensive Analysis Performance Predictions of the NREL Phase VI Experiment. *Journal of Solar Energy Engineering* 125 (2003), 457–467.
- [8] Fares, E., and Wolfgang, S. A General One Equation Turbulence Model for Free Shear and Wall Bounded Flows. *Flow, Turbulence, and Combustion* 73 (2004), 187–215.
- [9] Ferrer, E., and Munduate, X. Wind Turbine Blade Tip Comparison Using CFD. Tech. rep., 2007.
- [10] Fingersch, L. J., Simms, D., Hand, M., Jager, D., J.Cotrell, Robinson, M., Schreck, S., and Larwood, S. Wind Tunnel Testing of NREL's Unsteady Aerodynamics Experiment. *AIAA / ASME Wind Energy Symposium 0035* (2001).
- [11] Gonzalez, A., and Munduate, X. Three Dimensional and Rotational Aerodynamics on NREL Phase VI Wind Turbine Blade. *Journal of Solar Energy Engineering* 130 (2008), 1–7.
- [12] Gupta, A. Computational Fluid Dynamic Simulation of Wind Turbines. Master's thesis, The Pennsylvania State University, 2006.

- [13] Hand, M., Simms, D., Fingersch, L. J., Jager, D., Larwood, S., J. Cotrell, and Schreck, S. Unsteady Aerodynamics Experiment Phase VI: Wind Tunnel Test Configurations and Available Data Campaigns. Tech. rep., National Renewable Energy Laboratory (NREL), 2001.
- [14] Hansen, A. C., and Butterfield, C. P. Aerodynamics of Horizontal Axis Wind Turbines. *Annual Review of Fluid Mechanics* 25 (1993), 115–149.
- [15] Inc., Fluent. Fluent 5.5 Documentation. *User Guide 5.5* (2000).
- [16] Jasak, H. *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows*. PhD thesis, Imperial College of Science, Technology and Medicine, 1996.
- [17] Jonkman, J. M. Dynamics Modeling and Loads Analysis of an Offshore Floating Wind Turbine. Tech. rep., National Renewable Energy Laboratory (NREL), 2007.
- [18] Jr., John D. Anderson. *History of Aerodynamics and its Impact on Flying Machines*. Cambridge University Press, 1999.
- [19] Kitware. Paraview - Open Source Scientific Visualization. <http://www.paraview.org/>, 2009.
- [20] Laino, David J. Validation of AERODYN Subroutines using NREL Unsteady Aerodynamics Experiment. *AIAA 39* (2002), 179–189.
- [21] Ltd., OpenCFD. OpenFOAM: The Open Source CFD Toolox. *User Guide 1.5* (2008).
- [22] Ltd., OpenCFD. OpenFOAM: The Open Source CFD Toolox. *Programmer's Guide 1.5* (2008).
- [23] Manwell, J. F., McGowan, J. G., and Rogers, A. L. *Wind Energy Explained: Theory, Design, and Applications*. Wiley, 2004.
- [24] Musial, W., S. Butterfield, and Boone, A. Feasibility of Floating Platform Systems for Wind Turbines. *American Institute of Aeronautics and Astronautics(AIAA) AND American Society of Mechanical Engineers 42* (2004), 476–486.
- [25] Neff, D. E., Meroney, R. N., McCarthy, E., and Davis, E. Upstream and Lateral Wake Effects on Wind Turbine Performance. *U.S. Wind Engineering Conference 6* (1989).
- [26] Oertel, Herbert, Prandtl, Ludwig, Bohle, M., and Mayes, Katherine. *Prandtl's Essentials of Fluid Mechanics*. Springer, 2004.
- [27] Petit, Oliver. OpenFOAM: Different Ways to Treat Rotating Geometries. *User Guide 1.4.1* (2008).

- [28] Ramsay, R., Hoffman, M. J., and Gregorek, G. M. Effects of Grit Roughness and pitch oscillations on the S809 airfoil. Tech. rep., National Renewable Energy Laboratory (NREL), 1995.
- [29] Robinson, M.C., Hand, M. M., Simms, D. A., and Schreck, S. J. Horizontal Axis Wind Turbine Aerodynamics: Three Dimensional, Unsteady and Separated Flow Influences. Tech. rep., 1999.
- [30] Rutten, M., Alrutz, T., and Wendland, H. A Vortex Axis and Vortex Core Border Grid Adaptation Algorithm. *International Journal for Numerical Methods in Fluids* 58 (2008), 1379–1405.
- [31] Schmitz, Sven., and Chattot, Jean-Jacques. Characterization of Three Dimensional Effects for the Rotating and Parked NREL Phase VI Wind Turbine. *Journal of Solar Energy Engineering* 128 (2006), 445–454.
- [32] Schreck, S. J., and Robinson, M. C. Horizontal Axis Wind Turbine Blade Aerodynamics in Experiments and Modeling. *IEEE Transactions on Energy Conversions* 22, 1 (2007), 61–70.
- [33] Simms, D., Schreck, S., Hand, M., and Fingersch, L. J. NREL Unsteady Aerodynamics Experiment in the NASA-Ames Wind Tunnel: A Comparison of Predictions to Measurements. Tech. rep., National Renewable Energy Laboratory (NREL), 2001.
- [34] Somers, Dan M., and Tangler, J. Design and Experimental Results for the S809 Airfoil. *National Renewable Energy Laboratory (NREL)* (1997).
- [35] Sommers, D. M. Design and Experimental Results for the S809 Airfoil. Tech. rep., National Renewable Energy Laboratory (NREL), 1997.
- [36] Sorensen, N. N., Michelsen, J. A., and Schreck, S. Navier-Stokes Predictions of the NREL Phase VI rotor in the NASA Ames 80ft x 120ft Wind Tunnel. *Wind Energy* 5 (2002), 151–169.
- [37] Spalart, P. R., and Allmaras, S. R. A One Equation Turbulence Model for Aerodynamic Flows. *Recherche Aerospaciale* 1 (1994), 5–21.
- [38] Steffen, W., Sitzki, L., and Hahm, T. 3D - Simulation of a Turbulent Wake Behind a Wind Turbine. *Journal of Physics: Conference Series* 75 (2007), 012033.
- [39] Tangler, James L. The Nebulous Art of Using Wind Tunnel Airfoil Data for Predicting Rotor Performance. *AIAA* 40 (2002), 190–196.
- [40] U.S. Gov, Dept. of Energy. Energy Information Administration - Official Energy Statistics from the U.S. Government. <http://www.eia.doe.gov/>, 2009.

- [41] U.S. Gov, Dept. of Energy. U.S. Dept. of Energy on Green House Gases. <http://www.eia.doe.gov/bookshelf/brochures/greenhouse/Chapter1.htm>, 2009.
- [42] W., Wang J., B., Jia R., and Q., Wu K. Numerical Simulation on Effects of Pressure Distribution of Wind Turbine Blade with a Tip Vane. *Journal of Thermal Science 03* (2007), 203–207.
- [43] Wang, T., and Coton, Frank N. A High Resolution Tower Shadow Model for Downwind Wind Turbines. *Journal of Wind Engineering and Industrial Aerodynamics 89* (2001), 873–892.
- [44] Warsi, Z. U. A. *Fluid Dynamics: Theoretical and Computational Approaches*. CRC Press, 2000.
- [45] Wayman, E. Coupled Dynamics and Economic Analysis of Floating Wind Turbine Systems. Master’s thesis, Massachusetts Institute of Technology, 2006.
- [46] Wilcox, D. *Basic Fluid Mechanics*. DCW Industries, La Cañada, CA, USA, 2000.
- [47] Xu, G., and Sankar, Lakshmi N. Computational Study of Horizontal Axis Wind Turbines. *Journal of Solar Energy Engineering 122* (2000), 35–39.