

2014

CFD Simulation of the Flow around NREL Phase VI Wind Turbine

Yang Song

University of Massachusetts Amherst

Follow this and additional works at: https://scholarworks.umass.edu/masters_theses_2

 Part of the [Mechanical Engineering Commons](#)

Recommended Citation

Song, Yang, "CFD Simulation of the Flow around NREL Phase VI Wind Turbine" (2014). *Masters Theses*. 47.
https://scholarworks.umass.edu/masters_theses_2/47

This Open Access Thesis is brought to you for free and open access by the Dissertations and Theses at ScholarWorks@UMass Amherst. It has been accepted for inclusion in Masters Theses by an authorized administrator of ScholarWorks@UMass Amherst. For more information, please contact scholarworks@library.umass.edu.

**CFD SIMULATION OF THE FLOW
AROUND NREL PHASE VI WIND TURBINE**

A Thesis Presented

by

YANG SONG

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

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

May 2014

Mechanical and Industrial Engineering

© Copyright by Yang Song 2014

All Rights Reserved

**CFD SIMULATION OF THE FLOW
AROUND NREL PHASE VI WIND TURBINE**

A Thesis Presented

by

YANG SONG

Approved as to style and content by:

J. Blair Perot, Chair

Matthew A. Lackner, Member

Ashwin Ramasubramaniam, Member

Donald Fisher, Department Head
Mechanical and Industrial Engineering

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis advisor, Prof. Blair Perot, for his continuous interest, support and guidance during this study. I am also thankful to Prof. Matthew Lackner and Prof. Ashwin Ramasubramaniam for their advice and help. I specially wish to thank Chris Zusi for the all the support and great suggestions he gave me during my research as well as during my life in the United States. I am thankful to Kyle G. Mooney who gave a lot of help on OpenFOAM. I would also like to thank Alexander Chan and Nathaniel Develder for all the useful discussions during this research. I thank all my friends for making my stay at University of Massachusetts, Amherst happy and meaningful.

Lastly, I would like to thank my parents and my fiancée, Pei Qiu, without their love, encouragement and support I wouldn't be able to finish my work.

ABSTRACT

CFD SIMULATION OF THE FLOW AROUND NREL PHASE VI WIND TURBINE

MAY 2014

YANG SONG, B.E., WUHAN INSTITUTE OF TECHNOLOGY

M.S.M.E, UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor J. Blair Perot

The simulation of the turbulent and potentially separating flow around a rotating, twisted, and tapered airfoil is a challenging task for CFD simulations. This thesis describes CFD simulations of the NREL Phase VI turbine that was experimentally characterized in the $24.4\text{m} \times 36.6\text{m}$ NREL/NASA Ames wind tunnel. All computations in this research are performed on the experimental base configuration of 0° yaw angle, 3° tip pitch angle, and a rotation rate of 72 rpm. The significance of specific mesh resolution regions to the accuracy of the CFD prediction is discussed. The ability of CFD to capture bulk quantities, such as the low speed shaft torque, and the detailed flow characteristics, such as the surface pressure distributions, are explored for different inlet wind speeds. Finally, the significant three-dimensionality of the boundary layer flow is demonstrated.

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS	iv
ABSTRACT	v
LIST OF TABLES	viii
LIST OF FIGURES	ix
CHAPTER	
1. INTRODUCTION.....	1
1.1 Motivation.....	1
1.2 Background	2
1.3 Objectives	4
2. MODELING FOR CFD SIMULATION	6
2.1 Pre-processing.....	6
2.1.1 Mesh Generation.....	6
2.1.2 Mesh Refinement	14
2.1.3 Domain Decomposition	16
2.2 Code Formulation	17
2.2.1 Generalized Grid Interface	18
2.2.3 Numerical Schemes and Solution Control	20
2.2.4 Boundary and Initial Conditions.....	20
2.2.5 Turbulence Model	21
2.3 Post-processing	22
3. RESULTS AND DISCUSSIONS.....	23
3.1 Pressure Distributions	23

3.2 Low-Speed Shaft Torque	33
3.3 3D Effects.....	35
3.4 Computational Costs.....	37
4. CONCLUSION	39
APPENDICES	
A. S809 AIRFOIL AND NREL PHASE VI WIND TURBINE DATA.....	41
B. IMPLEMENTATION OF GENERALIZED GRID INTERFACE (GGI) IN OpenFOAM	43
C. NUMERICAL SCHEMES AND SOLUTION CONTROL IN OpenFOAM...	46
D. RELATIVE VELOCITY FIELD AT VARIOUS BLADE SECTIONS FOR DIFFERENT INLET WIND SPEEDS.....	50
BIBLIOGRAPHY	59

LIST OF TABLES

Table	Page
2.1	Boundary conditions for velocity and pressure21
2.2	Boundary conditions for \tilde{v}22
3.1	Numerical performance of the simulation under different numbers of processors and different methods of decomposition.....38
A.1	S809 Airfoil Coordinates41
A.2	NREL Phase VI blade chord and twist distributions.....42

LIST OF FIGURES

Figure	Page
1.1	NREL Phase VI wind turbine in NASA Ames wind tunnel.....3
2.1	A typical 2D structured mesh consisting of rectangular elements.....7
2.2	A typical 2D unstructured mesh consisting of triangular elements.8
2.3	A sliced section of an unstructured mesh around a 3D cylinder.....9
2.4	Surface mesh from an unstructured mesh consisting of 1.6 million tetrahedrons and prisms over the NREL Phase VI blade geometry (small object in the middle). 10
2.5	Surface mesh of one side of the NREL Phase VI blade geometry..... 10
2.6	A sliced section of the mesh around the NREL Phase VI blade (before modified)..... 11
2.7	A sliced section of the mesh around the sharp trailing edge (before modified)..... 11
2.8	Modified trailing edge of the NREL Phase VI blade geometry. 13
2.9	A sliced section of the mesh around the NREL Phase VI blade (after modified)..... 13
2.10	A sliced section of the mesh around the modified trailing edge. 14
2.11	A sliced section of the mesh at 80% span of the NREL Phase VI blade with 4 million total mesh cells and a coarse resolution of the leading edge..... 15
2.12	A sliced section of the mesh at 80% span of the NREL Phase VI blade with 6.3 million cells total and almost twice the mesh resolution on the leading edge..... 15
2.13	A sliced section of the mesh at 80% span of the NREL Phase VI blade with 10 million cells total and sufficient leading edge resolution. 16
2.14	Comparison of pressure distributions with different mesh resolutions at 80% span for the 10m/s case..... 16
2.15	Surface mesh from an unstructured mesh over the NREL Phase VI blade geometry. Blade is the small strip object. Rotating mesh is the cylinder. Outer mesh is the wind tunnel. This is a coarse version of the mesh, not

	the final one.	19
2.16	Streamlines flow over wind turbine blade.	22
3.1	Comparison of the pressure distributions for the 5m/s case. (a) at 30% span (b) at 47% span (c) 80% span.	24
3.2	Comparison of the pressure distributions for the 10m/s case. (a) at 30% span (b) at 47% span (c) 80% span.	25
3.3	Comparison of the pressure distributions for the 21m/s case (a) at 30% span (b) at 47% span (c) 80% span.	26
3.4	Relative velocity field and its vector field at 30% of the blade for 10m/s of inlet velocity.	28
3.5	Relative velocity field and its vector field at 47% of the blade for 10m/s of inlet velocity.	29
3.6	Relative velocity field and its vector field at 80% of the blade for 10m/s of inlet velocity.	30
3.7	Relative velocity field and its vector field at 80% of the blade for 5m/s of inlet velocity.	31
3.8	Relative velocity field and its vector field at 80% of the blade for 21m/s of inlet velocity.	32
3.9	Comparison of low speed shaft torque for wind speeds of 5m/s, 7m/s, 10m/s, 13m/s, 15m/s, 18m/s and 21m/s.	33
3.10	Limiting streamlines on the suction side of the blade for the inlet velocity of 5m/s, 10m/s and 21m/s.	34
3.11	Two views of the streamlines on the suction side of the blade for 21 m/s case.	35
3.12	Iso-surface of the second invariant velocity gradient tensor at $Q = 0.3 \text{ 1/s}^2$, for the 10 m/s case. This identifies the trailing vortices at the tip and the root.	36
3.13	Iso-surface of X velocity at $U_X = 8.1 \text{ m/s}$, for the 10m/s case.	37
D.1	Relative velocity field and its vector field at 30% of the blade for 5m/s of inlet velocity.	50
D.2	Relative velocity field and its vector field at 47% of the blade for 5m/s of inlet velocity.	51

D.3	Relative velocity field and its vector field at 80% of the blade for 5m/s of inlet velocity.....	52
D.4	Relative velocity field and its vector field at 30% of the blade for 10m/s of inlet velocity.....	53
D.5	Relative velocity field and its vector field at 47% of the blade for 10m/s of inlet velocity.....	54
D.6	Relative velocity field and its vector field at 80% of the blade for 10m/s of inlet velocity.....	55
D.7	Relative velocity field and its vector field at 30% of the blade for 21m/s of inlet velocity.....	56
D.8	Relative velocity field and its vector field at 47% of the blade for 21m/s of inlet velocity.....	57
D.9	Relative velocity field and its vector field at 80% of the blade for 21m/s of inlet velocity.....	58

CHAPTER 1

INTRODUCTION

1.1 Motivation

Energy depletion is becoming one of the most significant concerns of society during the coming decades. The fossil fuels that are currently pushing human society to further prosperity were formed from the tissues of organisms that lived 100-500 million years ago. In order to keep up with the pace of modern civilization, fossil fuels, including coal, oil and natural gases, are being consumed at an incredibly fast rate. At this rate those valuable reserves will be depleted in the near future. Therefore, developing substitute energy sources is very important. Renewable energy sources, such as wind, solar, tides and geothermal energy have nearly an unlimited supply and can be converted into useful power.

Wind power is one of the most abundant energy sources on the earth. The first time people started to extract power from the wind can be traced back to the 1st century when the first wind power driven machine was invented. However, the application of wind power was very limited because of the structural complexity and high cost of wind turbines. The potential of wind energy as a power source was reinvigorated in the late 1960s because of the increasing concern for environment. As a consequence, public large scale modern wind turbines started emerging (Manwell

et al., 2009).

A lot of effort and resources are being spent by researchers today in order to efficiently extract the power from wind. Compared with setting up a wind tunnel and performing a full scale test, the cost of numerical simulation is much less. Therefore, Computational Fluid Dynamics (CFD) analysis is becoming more desirable. With the advance of computational science it is possible to perform large CFD simulations on powerful computers. However, it is still a big challenge for a CFD simulation to include all the complicated factors associated with the wind turbine aerodynamics. The current research is carried out to address these issues. This work aims to analyze the difficulties of CFD simulations and attempts to develop an effective CFD model that can simulate the flow past a rotating wind turbine.

1.2 Background

After taking advantage of airfoil analysis, rotor development, material technology, and stress analysis, wind power is becoming a suitable substitute for fossil fuels. Although the aerodynamic effects on the wind turbines are well known the detailed flow behavior remains mysterious. Researchers are now attempting to take on the challenge of simulating the unsteady behavior of the flow around the wind turbines. The unsteady behavior of such flow is presented in the NREL report by Robinson et al. (1999). Increased attention has been made on the CFD simulation of the flow behavior of the wind turbine in the last decade. In these approaches, the Reynolds

Averaged Navier-Stokes equations are solved numerically by the CFD code coupled with a turbulence model (Duque et al., 2003).

The NREL Phase VI test is a full scale Unsteady Aerodynamic experiment (UAE) on the double-bladed 10.058 m diameter NREL Phase VI Rotor based on S809 airfoil and performed in the 24.4 m \times 36.6 m NASA-Ames wind tunnel (Hand et al., 2001). Figure 1.1 below is the NREL Phase VI wind turbine in NASA Ames wind tunnel. The S809 airfoil coordinates and the blade chord and twist distributions are given in Appendix A. The test results obtained from this experiment have provided a data set for numerical simulation.



Figure 1.1 NREL Phase VI wind turbine in NASA Ames wind tunnel.

Researchers from different countries were motivated by the availability of the NREL experimental data and have performed CFD simulations using all kinds of

solvers under different conditions in order to further understanding of the detailed flow behavior of the wind turbine. Tangler (2002) tested multiple versions of a Blade Element Method code. Laino et al. (2002) performed a 2D simulation of the S809 airfoil using the AERODYN code and matches those results with the NREL data. Sorensen et al. (2002) applied an incompressible RANS code to predict several cases from the NREL and NASA wind tunnel tests. Duque et al. (2003) gave a comprehensive investigation of a RANS computation using the CAMRAD II and OVERFLOW-D2 codes performed on the double-blade NREL Phase VI rotor. Xu and Sankar (2000) performed a RANS computation using a 3D viscous flow model. Gonzalez and Munduate (2008) analyzed the aerodynamic properties of the blades, such as attached flow, separated flow, and stall, of parked and rotating configurations of the NREL Phase VI wind turbine by testing a 2D section of the blades. Similar results for the same configuration were also presented by Schmitz and Chattot (2006).

1.3 Objectives

The goal of this research is to analyze the difficulties of CFD simulations and try to build an effective CFD model that can be used to simulate the flow over a wind turbine blade. In order to achieve this goal, three sub-objectives are required. First of all successful grid generation is required. The performance of a CFD simulation relies on a fine mesh (with no high-skew elements). And this step requires a fair amount of time and efforts, especially in this specific project where the mesh generation on the

sharp trailing edge of the blade is really challenging.

After the grid is generated, accurate modeling for CFD simulation is required. Since there are two frames of reference, wind tunnel (stationary frame) and wind rotor (rotating frame), special techniques need to be applied to solve the Navier-Stokes equations. This work differs from those prior studies in its choice of turbulence model (Spalart-Almaras), and the decision not to use wall-function boundary conditions which algebraically model the boundary layer profile.

Once the CFD results are obtained, post-processing software is required to analyze and visualize the results. Comparisons between the CFD result and NREL Phase VI test result on different properties need to be done in order to validate the CFD model and to provide hints about modification. At the end, this project is expecting to present a good understanding of CFD modeling features and to develop a comprehensive functional CFD model that can be used for studying other features of wind turbine systems by other researchers in the future.

CHAPTER 2

MODELING FOR CFD SIMULATION

This chapter presents the methodology of the simulation in this research including: pre-processing, code formulation and post-processing. Pre-processing including mesh generation and converting the mesh file into a format that the CFD solver can understand, and decomposing the domain into several sub-domains to accelerate the computation of the next step. This part is usually the most time-consuming job in achieving the CFD simulation result. Code formulation is to set up the boundary and initial conditions, the turbulence model and other parameters and get numerical results from the simulation. Post-processing is using certain software to visualize the numerical results obtained from the CFD solver in a user friendly manner.

2.1 Pre-processing

Pre-processing is the first important step in obtaining CFD solution and also, as mentioned above, is the most tedious and time-consuming one. This task involves mesh generation and domain decomposition.

2.1.1 Mesh Generation

Meshing is the most important pre-processing step. It's very important to generate

meshes with high quality in order to obtain accurate CFD results. Before getting into meshing details of the current project we will discuss two types of widely used meshes: structured meshes and unstructured meshes.

A structured mesh can be defined as a type mesh that every interior cell has the same number of neighbor elements (Owen, 1998). Usually, the structured mesh generated by a grid generator is a collection of repeating quadrilaterals or hexahedrons. Since the elements in a structured mesh are regularly arranged, they can be simply identified by the index (i, j) in 2D and (i, j, k) in 3D. This will accelerate the computation process when running the simulation, but non-trivial boundaries are required (Owen, 1998). In order to get a detailed solution for certain areas of the domain, such as the boundary region, the meshes at those regions can be easily refined by stretching in certain direction to make denser grids (Hansen et al., 2005). The figure 2.1 shows a typical 2D structured mesh consisting of rectangular elements made by the mesh generating software, ICEM CFD (ANSYS, 2010).

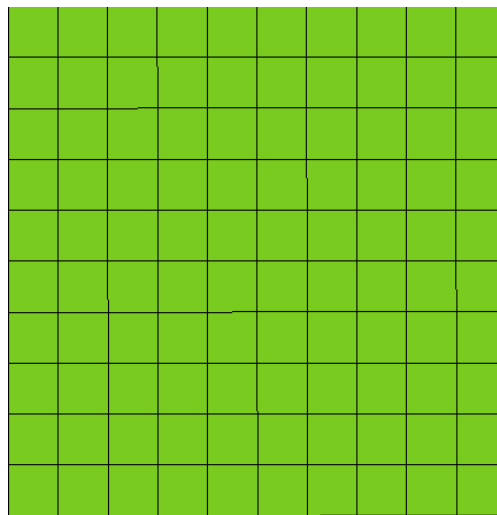


Figure 2.1 A typical 2D structured mesh consisting of rectangular elements.

Unstructured meshes, on the other hand, consist of irregularly arranged elements. Mesh elements are triangles and rectangles in 2D, and tetrahedrons, pyramids or prisms in 3D. Any combination of elements can meet at a single node so that the elements in an unstructured mesh cannot be identified by index (i, j) or (i, j, k) like a structured mesh. This means that a simulation with an unstructured mesh requires much more computational effort compared with a structured grid. Figure 2.2 shows a typical 2D unstructured mesh consisting of triangular elements.

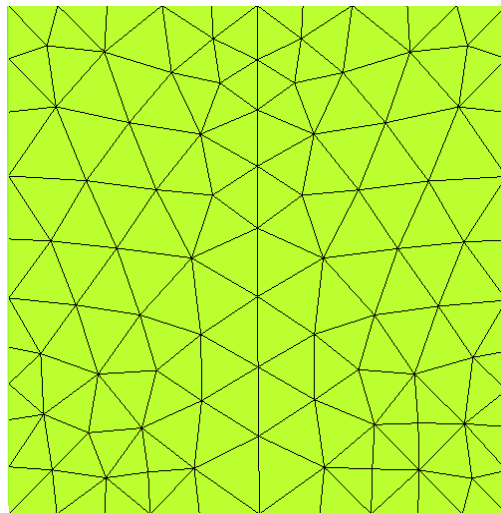


Figure 2.2 A typical 2D unstructured mesh consisting of triangular elements.

The advantage of unstructured meshes is that they can be applied to most geometries due to the flexibility of the shapes and sizes of the elements (ANSYS, 2010). Figure 2.3 shows a sliced section of an unstructured mesh around a 3D cylinder. The geometry is not complex but the unstructured mesh can be concentrated in the wake where it is needed most. Unstructured meshes are applied throughout this project so that the CFD simulation can be performed on the wind turbine blade which

has a complex geometry. Figure 2.4 shows the surface triangulation of an unstructured mesh consisting of 1.6 million tetrahedrons and prisms over the NREL Phase VI blade geometry. Figure 2.5 shows the surface mesh of one side of the NREL Phase VI blade geometry. Figure 2.6 shows a sliced section of the mesh, which is perpendicular to one the blade. However the mesh shown in figure 2.6 is not a desirable mesh for this project. The reason for that can be easily understood by a zoomed in picture of the geometry (Figure 2.7).

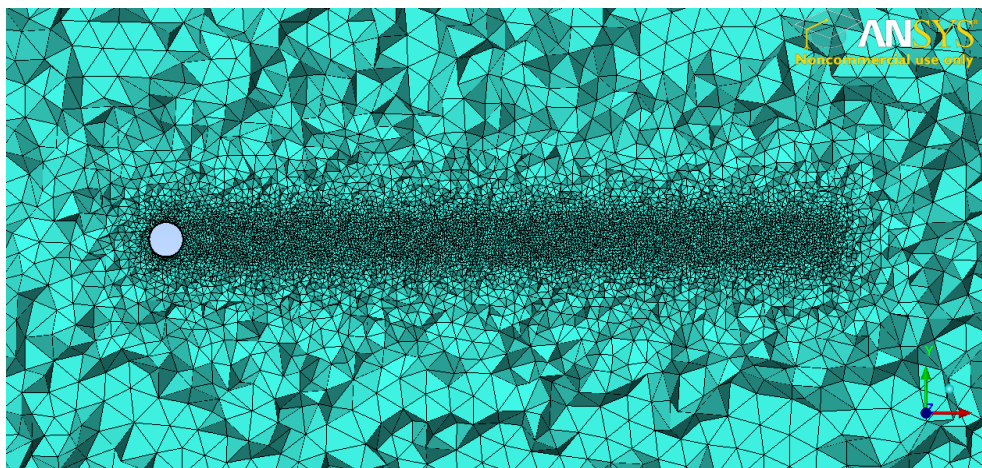


Figure 2.3 A sliced section of an unstructured mesh around a 3D cylinder.

The mesh around the sharp trailing edge of the blade consists of severely non-orthogonal cells which will make the CFD solution unstable and thus inaccurate. For this project we decided to slightly modify the blade geometry by making the sharp trailing edge blunt to get rid of the highly non-orthogonal elements.

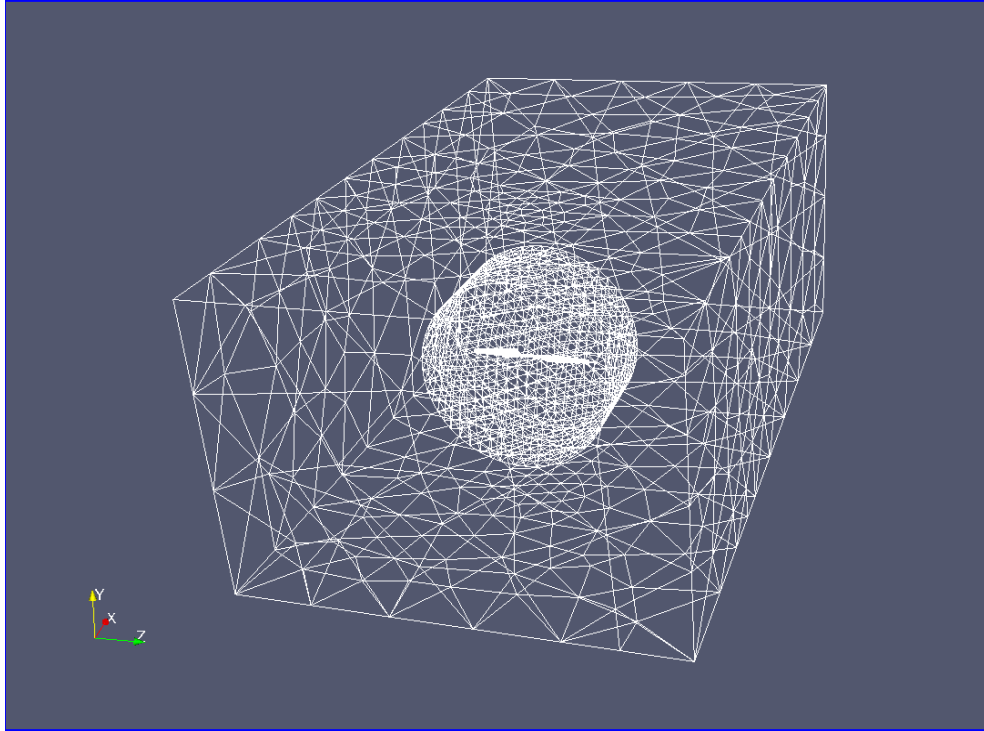


Figure 2.4 Surface mesh from an unstructured mesh consisting of 1.6 million tetrahedrons and prisms over the NREL Phase VI blade geometry (small object in the middle).

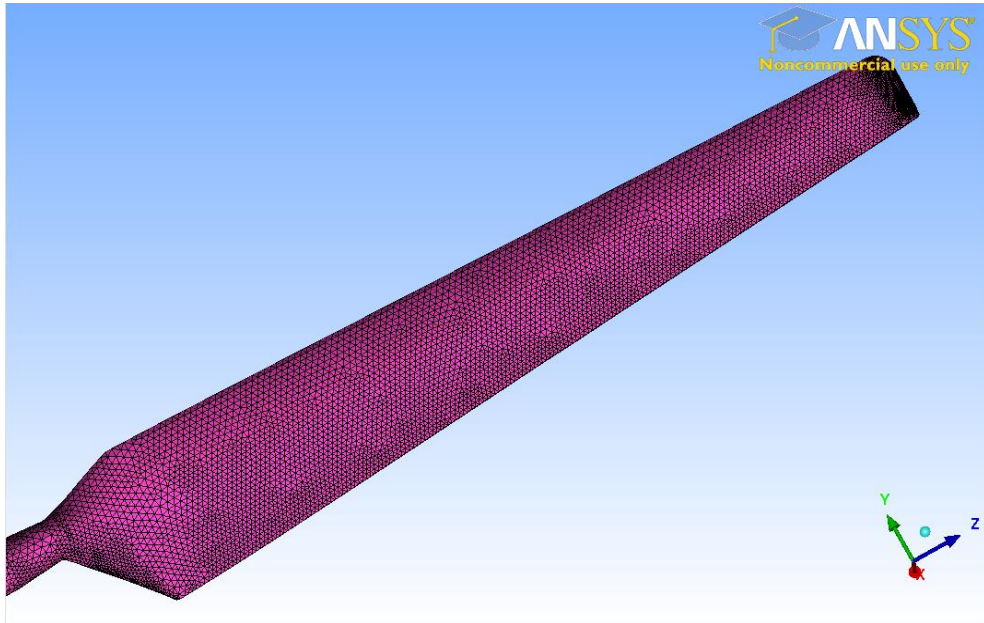


Figure 2.5 Surface mesh of one side of the NREL Phase VI blade geometry.

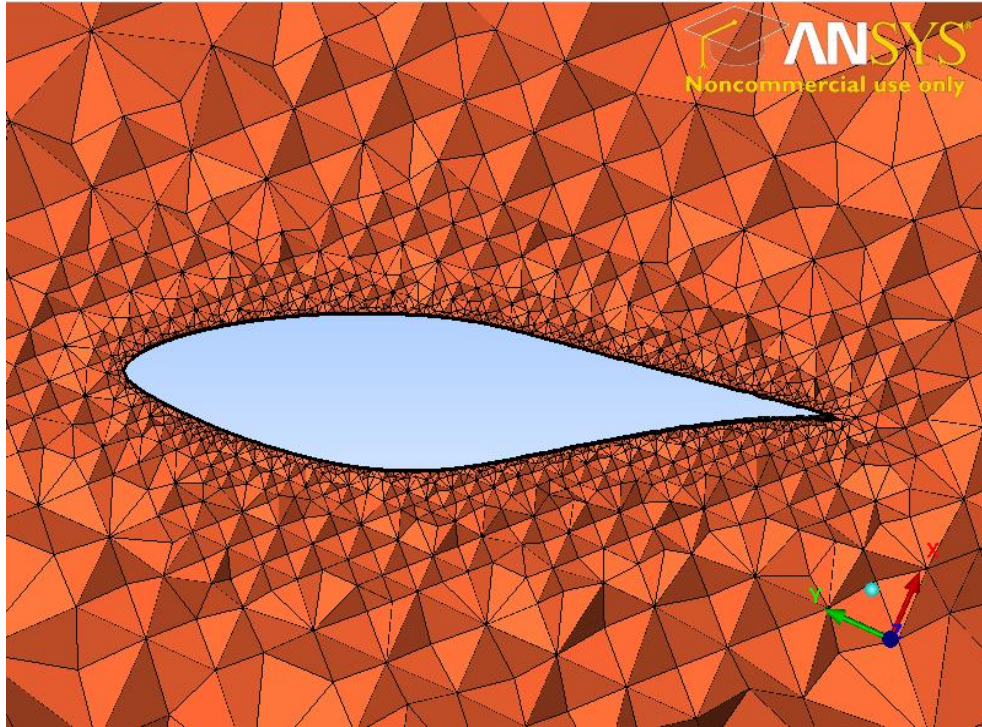


Figure 2.6 A sliced section of the mesh around the NREL Phase VI blade (before modified).

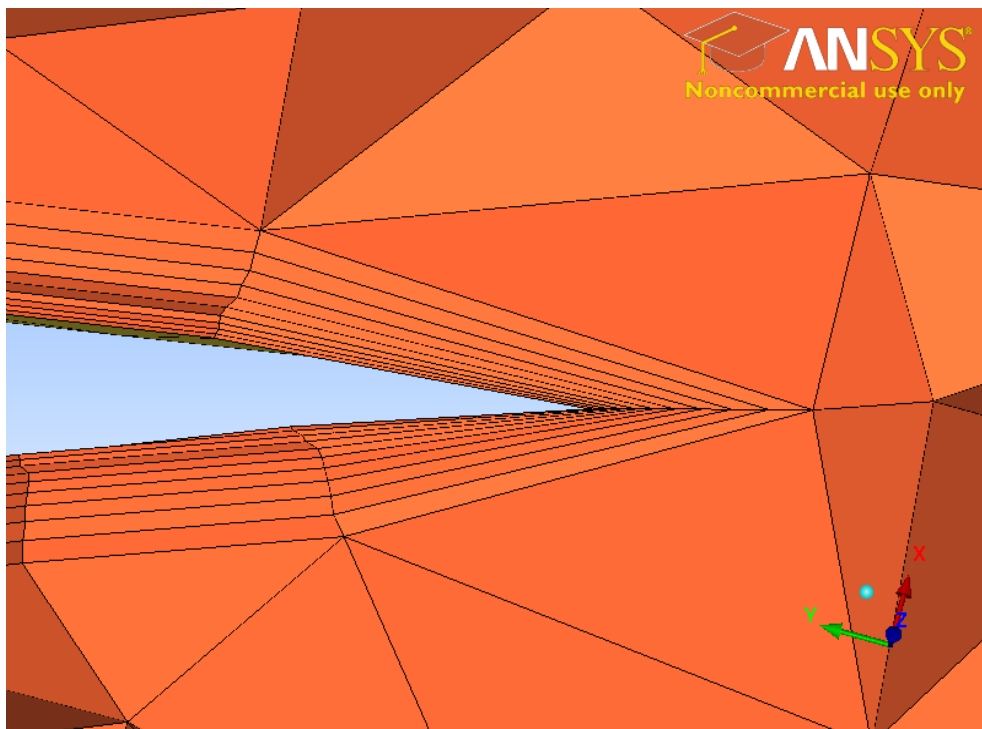


Figure 2.7 A sliced section of the mesh around the sharp trailing edge (before modified).

Figure 2.8 shows the modified trailing edge of the NREL Phase VI blade geometry. Figure 2.9 shows the sliced section of the mesh based on the modified geometry and Figure 2.10 shows a zoomed in picture of the mesh around the modified trailing edge. This modification reduces the chord length by roughly 2%.

In order to avoid using wall functions the mesh is highly refined in the wall normal direction in a thin layer next to the airfoil. In order to integrate the PDE without wall function boundary conditions the first grid point away from the wall needs to reside at a y^+ less than or equal to 5. High aspect ratio tetrahedra have very non-orthogonal faces, so next to the airfoil the mesh consists of very flat prisms. The prisms are aligned normal to the blade surface. In Figure 2.10, the rectangles near the airfoil are actually a slice through these prisms. To integrate the equations up to the wall, the first prism layer next to the wall needs to be 5×10^{-5} meters high. The thickness of each prism grows by 15% as the prisms move away from the wall. In many locations the prisms next to the airfoil are very thin and have an aspect ratio (height to width ratio) of over 400.

The grid generation software, ICEM CFD (ANSYS, 2010), has been used throughout this research to build meshes. The meshes made by ICEM CFD are converted into a format that OpenFOAM can recognize and then are inputted into the solver. In order to get stable and accurate CFD results, meshes with good quality are required. The work of grid generation requires a fair amount of time and effort.

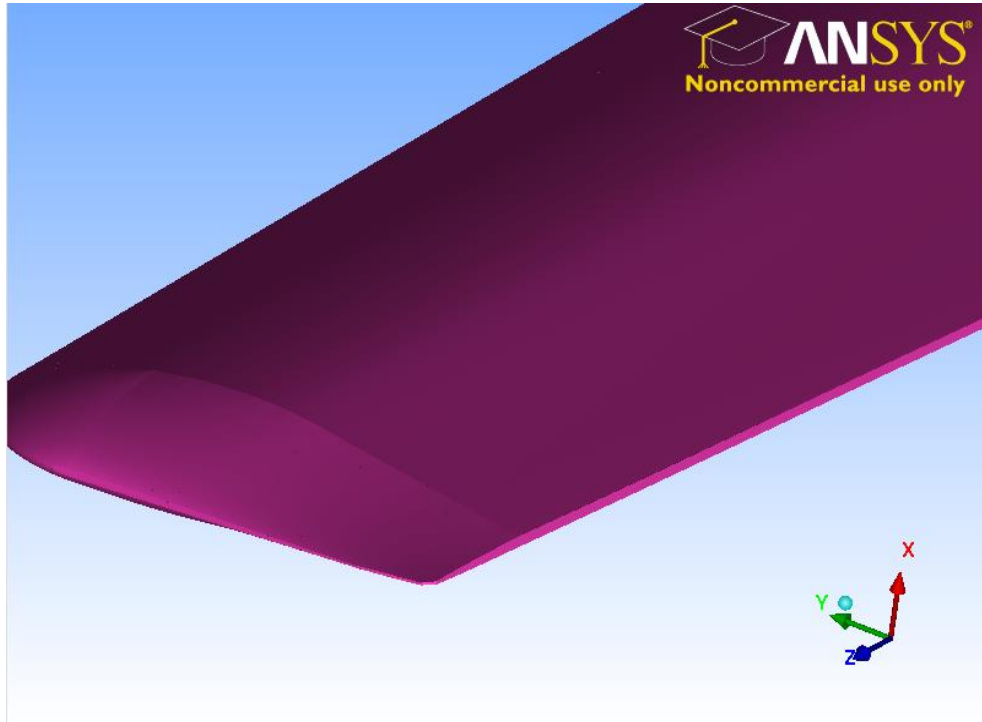


Figure 2.8 Modified trailing edge of the NREL Phase VI blade geometry.

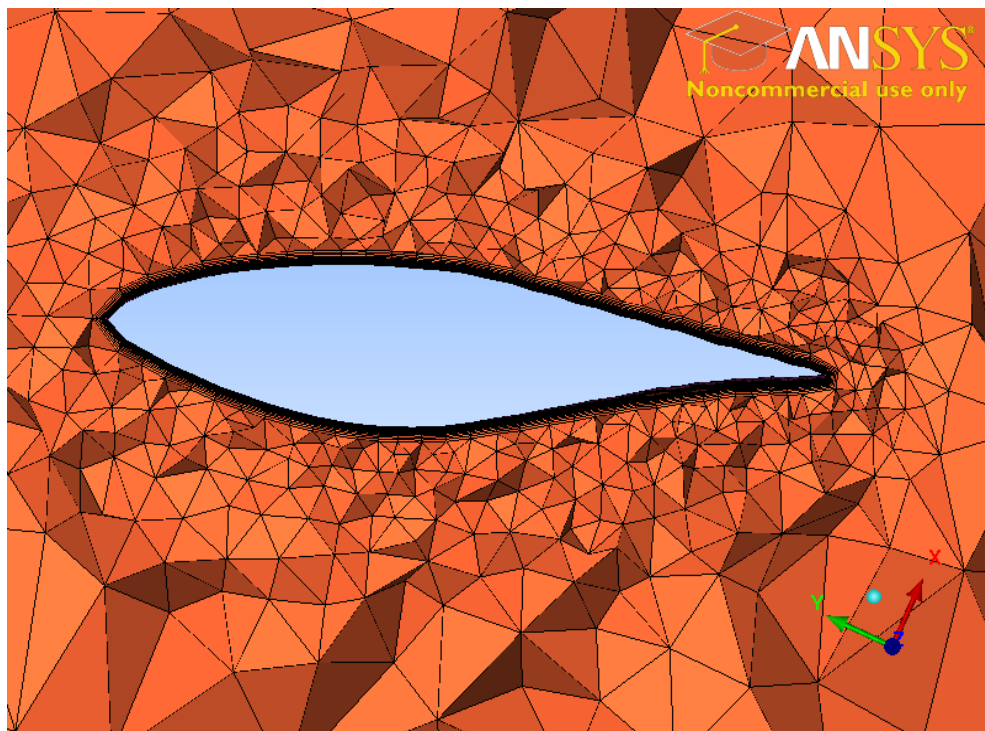


Figure 2.9 A sliced section of the mesh around the NREL Phase VI blade (after modified).

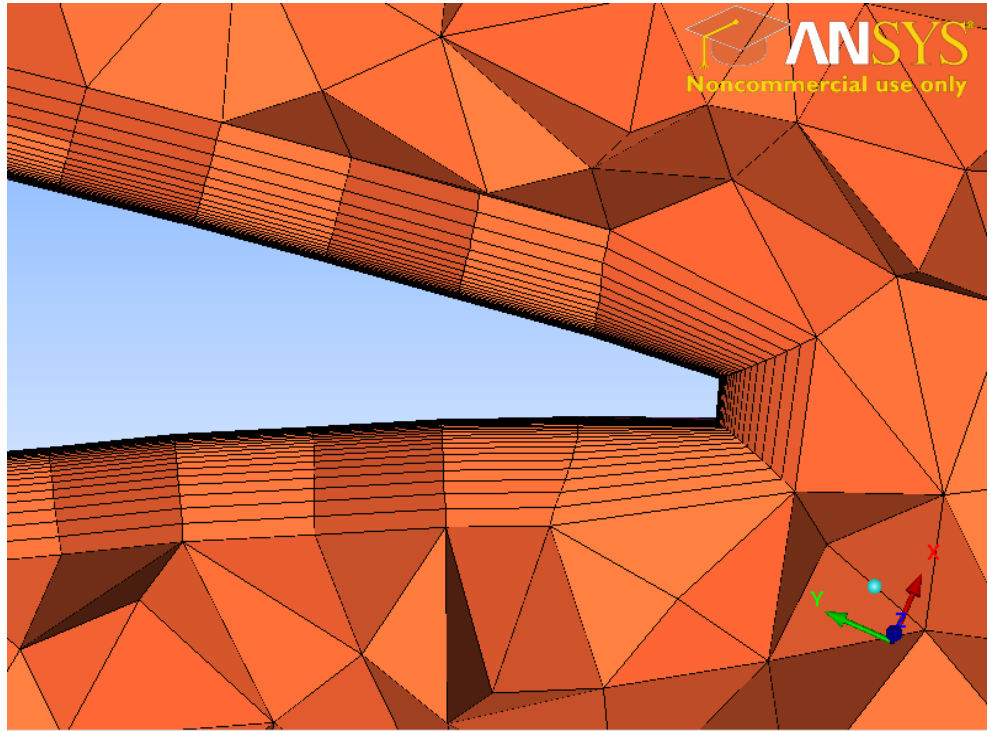


Figure 2.10 A sliced section of the mesh around the modified trailing edge.

2.1.2 Mesh Refinement

Normal resolution is required on the entire blade, but chordwise and spanwise resolutions are also required at the blade leading edge. At this location there are large pressure tangential pressure gradients that must be resolved for an accurate computation. Meshes with different leading edge resolutions were investigated. Figures 2.11, 2.12 and 2.13 shows sliced section of the mesh at 80% span of the blade, which contain a total of 4 million, 6.3 million and 10 million mesh cells respectively. Figure 2.14 shows the pressure distributions computed for these 3 different meshes. These results are for the inlet wind speed of 10m/s at 80% span. Only the largest mesh

is capable of predicting the pressure spike at the leading edge (and the resulting blade torque) reasonably accurately.

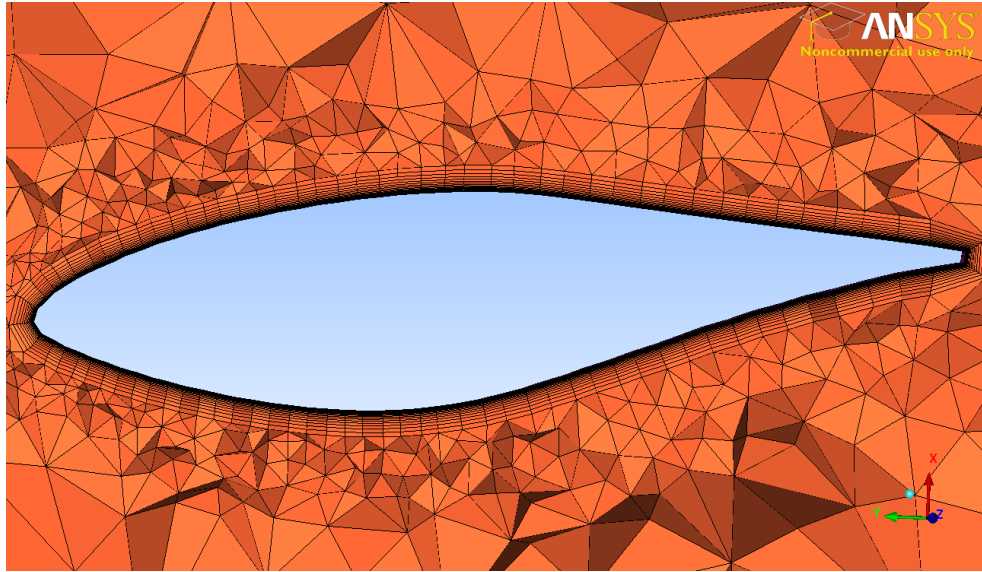


Figure 2.11 A sliced section of the mesh at 80% span of the NREL Phase VI blade with 4 million total mesh cells and a coarse resolution of the leading edge.

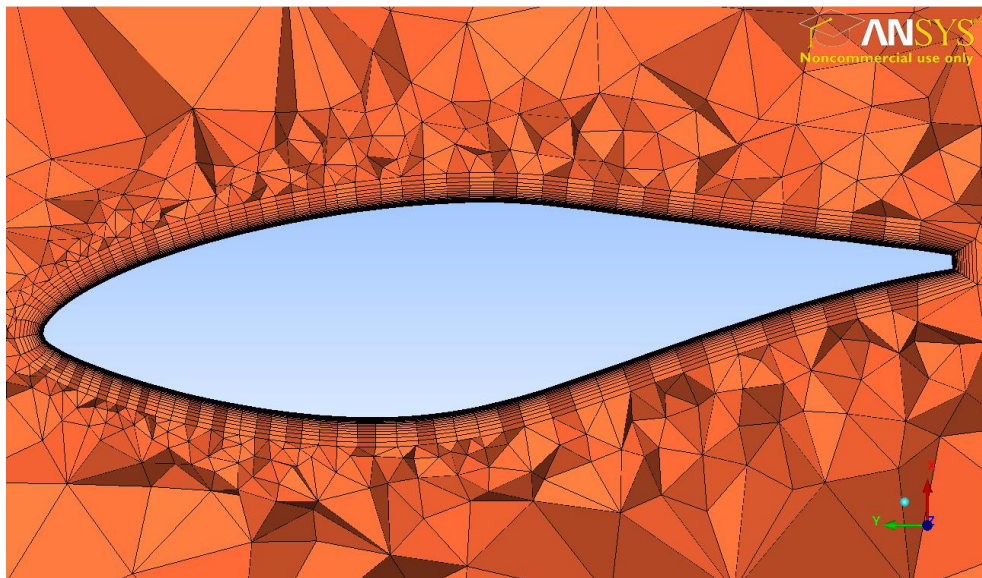


Figure 2.12 A sliced section of the mesh at 80% span of the NREL Phase VI blade with 6.3 million cells total and almost twice the mesh resolution on the leading edge.

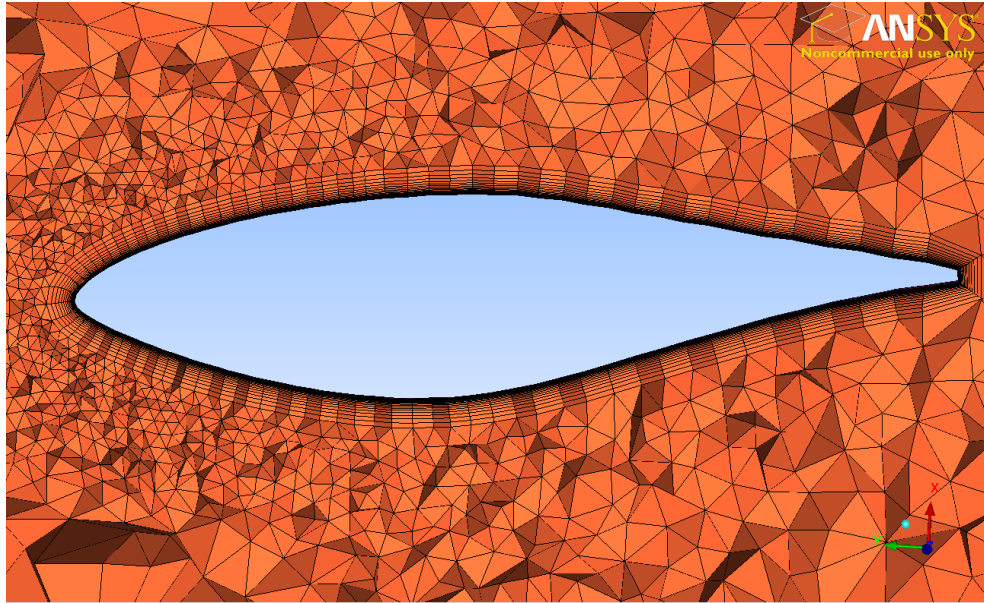


Figure 2.13 A sliced section of the mesh at 80% span of the NREL Phase VI blade with 10 million cells total and sufficient leading edge resolution.

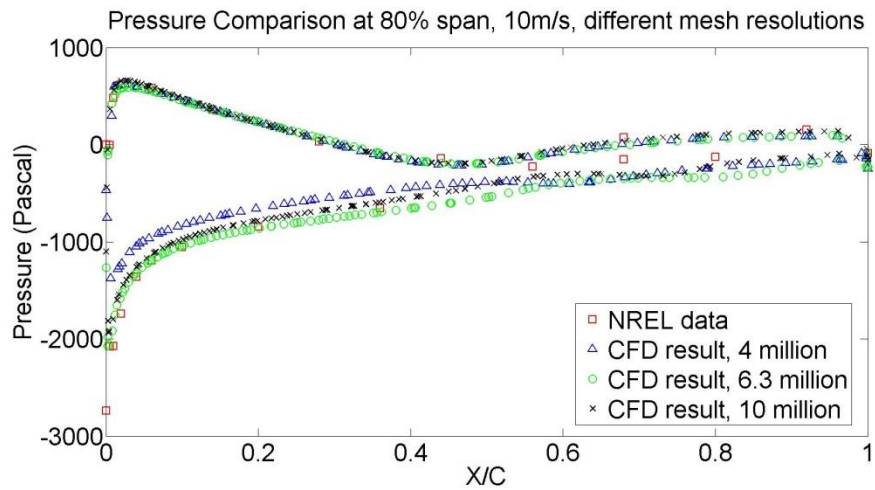


Figure 2.14 Comparison of pressure distributions with different mesh resolutions at 80% span for the 10m/s case.

2.1.3 Domain Decomposition

Mesh partitioning is one of the most important pre-processing steps. A CFD

simulation that is performed with a mesh with a large number of elements is usually computationally expensive and thus requires the mesh to be decomposed into several sub-domains and distributed to the same number of computer processors for performing parallel computation. The partition method used in this project is ‘Scotch’ decomposition. Scotch decomposition attempts to minimize the communication between different processors by minimizing the number of elements on processor boundaries and requires no specification of the geometry by the user. The Scotch method is more efficient than the ‘Simple’ method which equally chops the domain into several sub-domains based on geometry inputs specified by the user. The parallel computations in this research are performed on the University of Massachusetts Amherst in-house supercomputer cluster, Cyclops, which has 608 processing cores. In this research, calculations are distributed to 16, 32 or 64 processors using OpenFOAM. A comparison of the numerical speed by decomposing the domain with the Simple and Scotch methods was also done in order to prove the advantages of the Scotch method. The performance of the simulations on different numbers of processors using different decomposition methods is presented in Chapter 3.

2.2 Code Formulation

In this research, air flows in from the inlet of the tunnel, which is the inertial frame and goes through the rotating wind turbine rotor, which is the non-inertial frame. Thus, an effort must be made to apply the Navier-Stokes equations to multiple

frames of reference. Generalized Grid Interface (GGI) has been applied in this work to connect different frames of reference. The GGI interface uses a special interpolation algorithm that allows for general grid movement avoiding the complicated work of a topologically deforming mesh. In this research, the CFD solver that we use to solve the Reynold's Averaged Navier-Stokes equations is an incompressible transient turbulent flow solver. The Reynold's Averaged Navier-Stokes equations are given by equations 2.1 and 2.2. They represent the conservation of mass and momentum respectively (Wilcox, 2007).

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

$$\frac{D}{Dt} = -\frac{1}{\rho} \nabla p + (\nu + \nu_T) \cdot \nabla^2 U \quad (2.2)$$

Solving the incompressible Navier-Stokes equations requires great effort. The merged PISO-SIMPLE (PIMPLE) algorithm has been applied in solving equation 2.1 and 2.2 in this research. PISO stands for Pressure Implicit with Splitting the Operators algorithm while SIMPLE represents Semi-Implicit Method for Pressure-Linked Equation algorithm. For a detailed explanation of the PISO and SIMPLE algorithms, refer to (Jasak, 1996).

2.2.1 Generalized Grid Interface

In our simulation, we have two frames of reference: the wind tunnel stationary frame, and the rotor rotating frame. The Navier-Stokes equations in the stationary frame cannot be directly applied to the rotating frame. In order to solve this problem,

the Generalized Grid Interface (GGI) is applied to this simulation. The GGI interface uses a special interpolation algorithm to allow two different meshes to slide next to each other. Generating a grid for a GGI case requires the user to create two sub-grids representing the stationary and rotating domains and to define the boundary of the moving region with two patches. Figure 2.15 shows the GGI mesh created for the simulation in this work. The disk shaped mesh rotates inside the larger cuboid mesh. And the blade geometry rotates with the disk mesh. Based on the NREL Phase VI experiment data (Hand et al., 2001) the dimension of wind tunnel test section in this simulation is $24.4\text{m} \times 36.6\text{m}$. For more details about how to set up a GGI case, please refer to Schmitt (2009). The detailed settings of GGI in OpenFOAM are given in Appendix B.

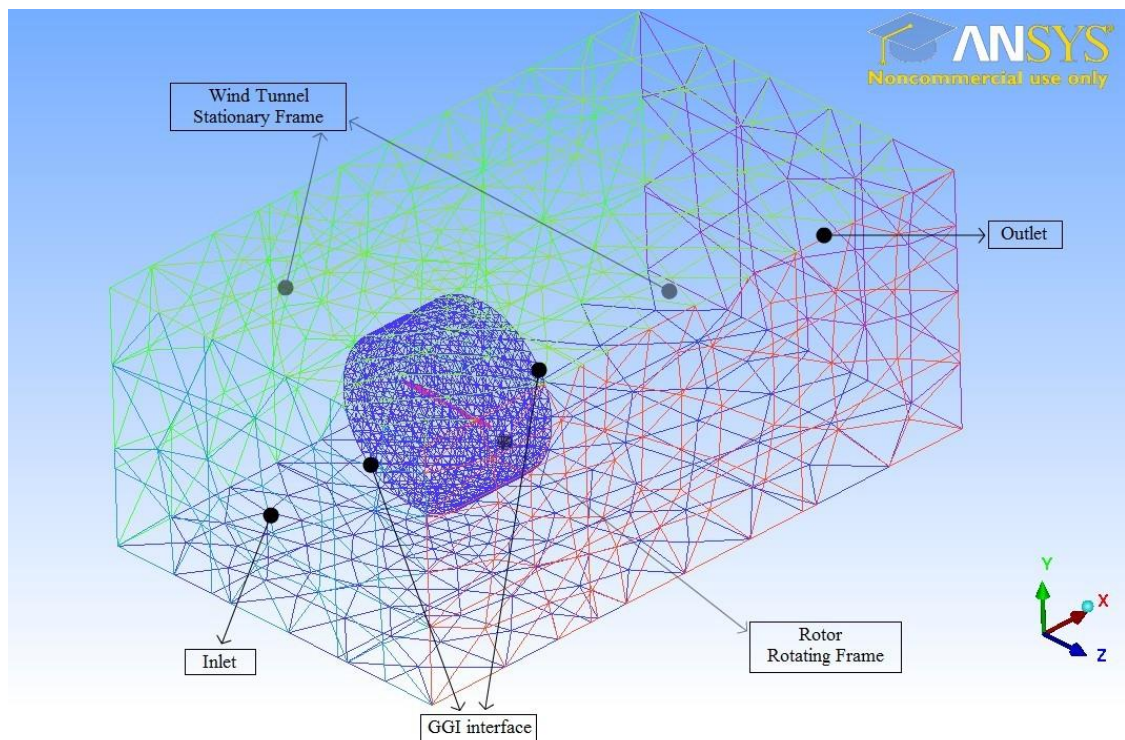


Figure 2.15 Surface mesh from an unstructured mesh over the NREL Phase VI blade geometry. Blade is the small strip object. Rotating mesh is the cylinder. Outer mesh is the wind tunnel. This is a coarse version of the mesh, not the final one.

2.2.3 Numerical Schemes and Solution Control

The numerical schemes and the solution parameters are controlled by the *fvSchemes* and *fvSolution* files in OpenFOAM. The Preconditioned Conjugate Gradient (PCG) and Preconditioned Bi Conjugate Gradient (PBiCG) were used to solve pressure and velocity terms respectively. The detailed settings of *fvSchemes* and *fvSolution* files are given in Appendix C. For more information about all the available options in those two files, please refer to Jasak (1996) and OpenFOAM (2011).

2.2.4 Boundary and Initial Conditions

For all the simulation presented in this work, the pressure is enforced as zero gradient at the inlet of the tunnel and zero value at the outlet, while the velocity is fixed at the inlet and has a zero gradient boundary condition at outlet. Slip conditions are used at the four side walls of the wind tunnel (so the thin boundary layers on the tunnel walls are not captured), and the no-slip boundary condition is applied on the blade surface. The OpenFOAM boundary condition settings for velocity and pressure are given in Table 2.1. For all the boundary conditions supported by OpenFOAM, refer to OpenFOAM (2011).

Table 2.1 Boundary conditions for velocity and pressure

Patch	BC for velocity	BC for pressure
Inlet	fixedValue	zeroGradient
Outlet	zeroGradient	fixedValue (0)
Side walls	slip	zeroGradient
S809 blades	movingWallVelocity	zeroGradient

2.2.5 Turbulence Model

The Spalart-Allmaras (SA) (Spalart and Allmaras, 1992) turbulence model was used to solve for the turbulent eddy-viscosity. The turbulent eddy-viscosity is given by the following equation:

$$\nu_t = \tilde{\nu} f_{\nu 1}, \quad f_{\nu 1} = \frac{X^3}{X^3 + C_{\nu 1}^3}, \quad X \equiv \tilde{\nu}/\nu \quad (2.3)$$

where ν is molecular viscosity and $\tilde{\nu}$ is a new variable given by the following equations:

$$\tilde{S} \equiv S + \frac{\tilde{\nu}}{k^2 d^2} f_{\nu 2}, \quad f_{\nu 2} = 1 - \frac{X}{1 + X f_{\nu 2}} \quad (2.4)$$

$$\frac{D\tilde{\nu}}{Dt} = c_{b1} \tilde{S} \tilde{\nu} + \frac{1}{\sigma} [\nabla \cdot ((\nu + \tilde{\nu}) \nabla \tilde{\nu}) + c_{b2} (\nabla \tilde{\nu})^2] - c_{\omega 1} f_{\omega} \left[\frac{\tilde{\nu}}{d} \right]^2 \quad (2.5)$$

In the SA turbulence model, $\tilde{\nu}$ is less expensive to compute than the turbulent kinetic energy k and dissipation rate ϵ . The SA turbulence model was developed at Boeing and is often favored in aerodynamic applications (Javaherchi T 2010). The SA turbulence model requires boundary condition on the variable $\tilde{\nu}$. In our simulation the boundary condition settings of $\tilde{\nu}$ on different patches are given in the following table.

Table 2.2 Boundary conditions for \tilde{v}

Patch	BC for \tilde{v}
Inlet	FixedValue (1.85e-4 m ² / s)
Outlet	zeroGradient
Side walls	zeroGradient
S809 blades	FixedValue (0)

2.3 Post-processing

Post-processing is the last step in obtaining the CFD solution. A post-processing software which is coupled to the CFD solver is required at this step. It usually involves visualization of the results of the CFD simulation. In our research, the open source post-processing software, ParaView (Squillacote, 2007), was used to analyze and visualize the CFD results. Figure 2.16 shows the streamlines of the flow over wind turbine blade.

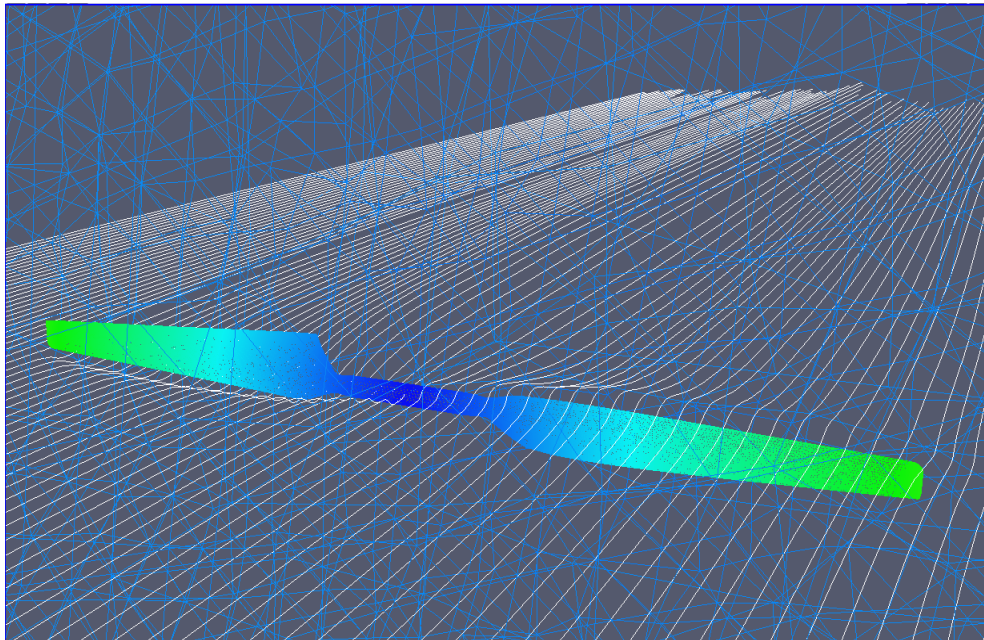


Figure 2.16 Streamlines flow over wind turbine blade.

CHAPTER 3

RESULTS AND DISCUSSIONS

In this chapter, the results obtained from the CFD simulation over the NREL Phase VI blade at a pitch angle of 3° and an angular velocity of 72 rpm will be presented. Our simulation has been tested at the inlet velocity of 5m/s, 7m/s, 10m/s, 13m/s, 15m/s, 18m/s and 21m/s. Comparison between results obtained from the CFD simulation and NREL Phase VI experiment is made in order to validate the CFD code.

3.1 Pressure Distributions

Comparisons of the NREL experimental data and the computed pressure distributions for 5m/s, 10m/s and 21m/s inlet wind speeds at three span sections, $r/R = 30\%$, 47% and 80% , are shown in Figure 3.1, 3.2 and 3.3, respectively.

For the case with a 5m/s inlet wind speed, good agreement is achieved for all three span locations, as shown in Figure 3.1. This is due to the fact that at this low inlet wind speed the blade functions as designed and there is no boundary layer separation.

For the higher inlet wind speed of 10m/s, good agreement is also found at the 80% span location, as shown in Figure 3.2. However, the 30% and 47% span locations are predicted less well. The difficulty lies on the top surface of the airfoil (lower curve) at the leading edge. It will be shown later that these cross sections are experiencing separation. The CFD simulation over predicts the pressure peak.

For the highest inlet wind speed of 21m/s, pressure distributions are again over

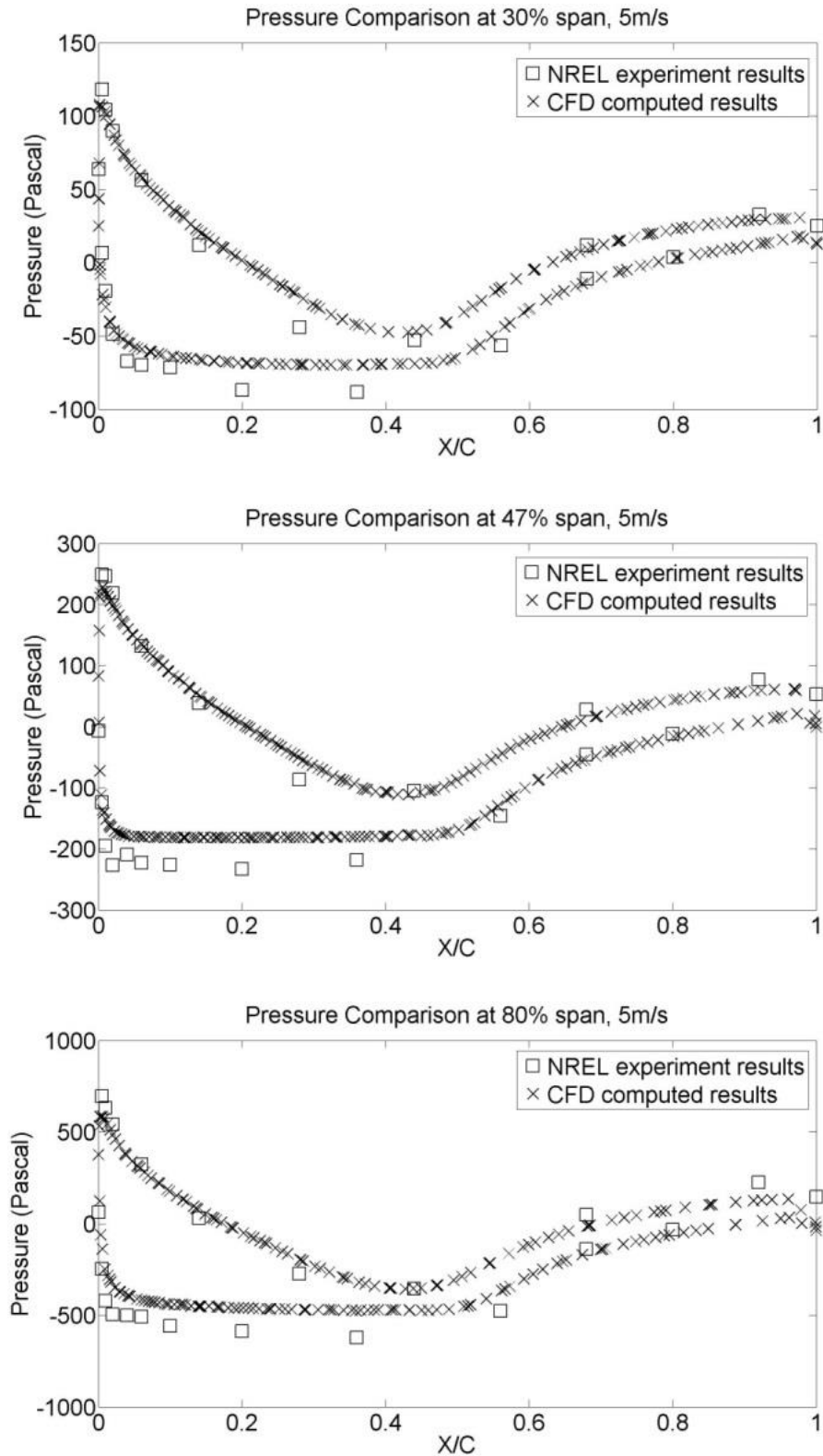


Figure 3.1 Comparison of the pressure distributions for the 5m/s case (a) at 30% span (b) at 47% span (c) 80% span.

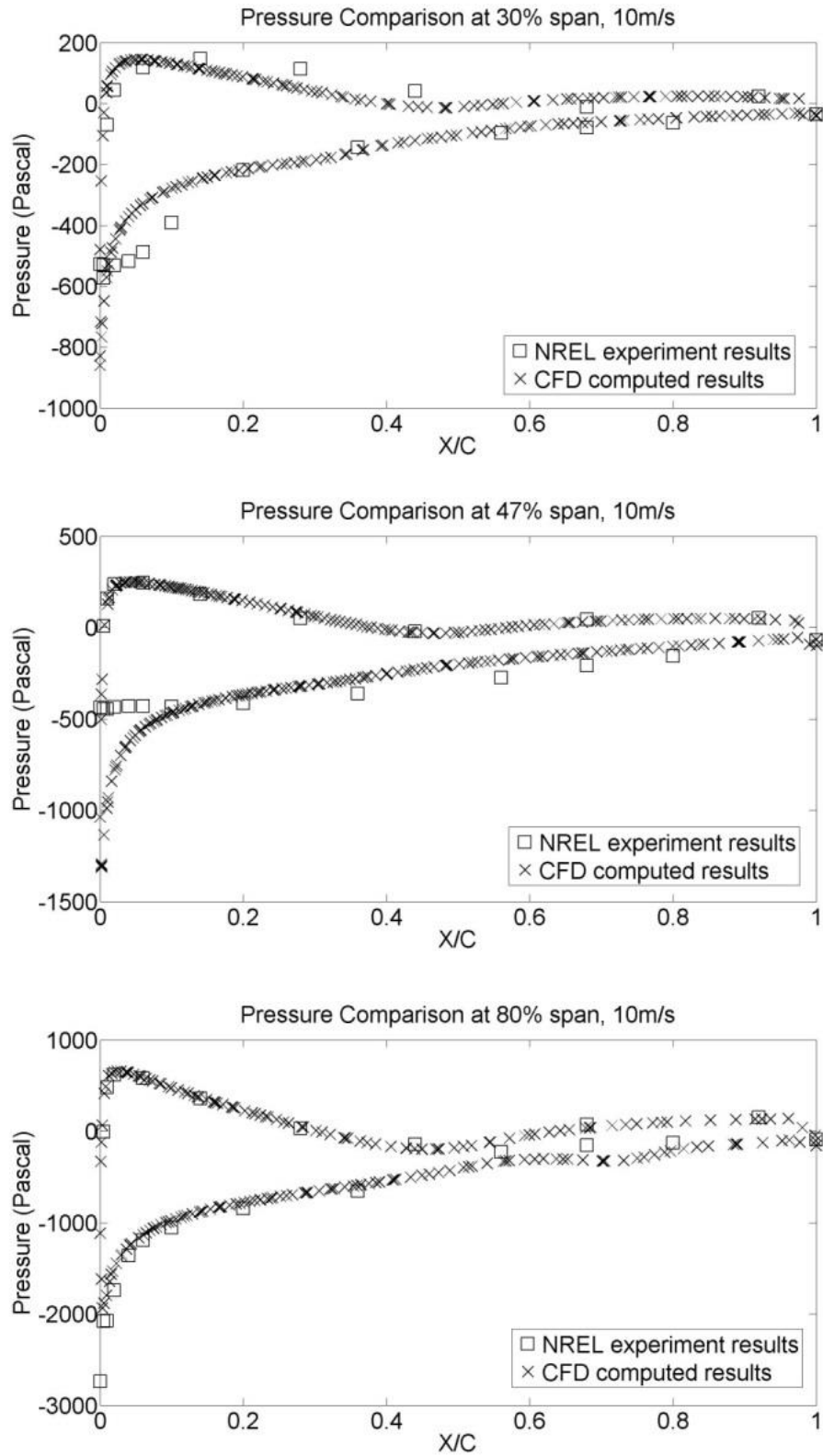


Figure 3.2 Comparison of the pressure distributions for the 10m/s case (a) at 30% span (b) at 47% span (c) 80% span.

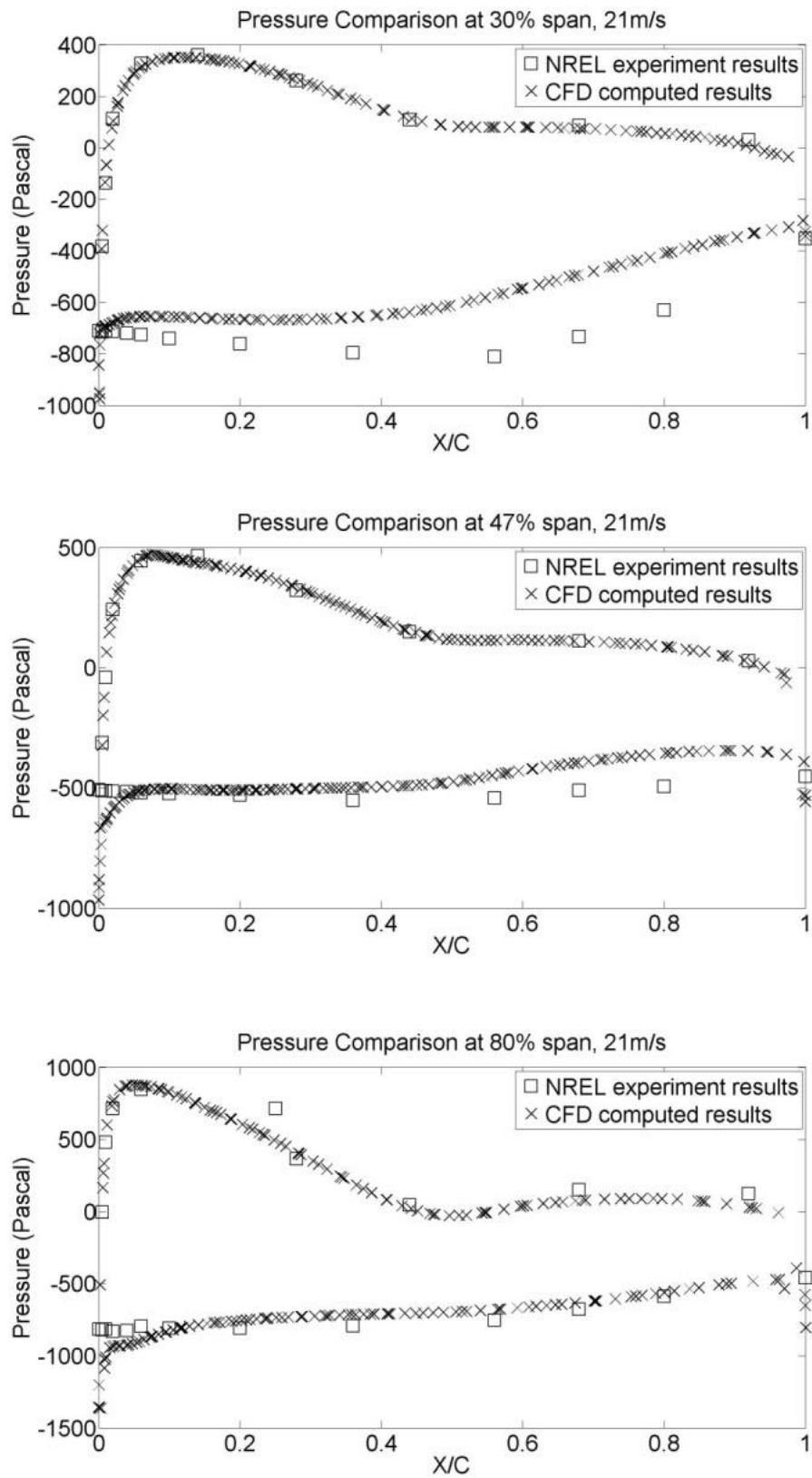


Figure 3.3 Comparison of the pressure distributions for the 21m/s case (a) at 30% span (b) at 47% span (c) 80% span.

predicted at the leading edge for all three span locations, as shown in Figure 3.3. For this case the incoming wind speed is so large that the entire blade is under complete stall conditions with separation occurring at the leading edge of the blades.

The stall effects can also be observed in the velocity field. Figure 3.4, 3.5 and 3.6 are the relative velocity field and its vector field of at 30%, 47% and 80% of the blade, respectively, for 10m/s of inlet velocity. At 30% and 47% span locations where the blade has large angle of attacks, the flow quickly leaves the blade surface on the suction side after passing the leading edge. While at 80% span, the flow separates much later. Figure 3.7 shows the relative velocity field and its vector field at 80% of the blade for 5m/s of inlet velocity, where the blade is operating under normal conditions. Figure 3.8 shows the relative velocity field and its vector field at 80% of the blade for 21m/s of inlet velocity, where the blade is operating under complete stall conditions. Additional velocity field plots are given in Appendix D. These figures are colored by the value of the streamwise velocity. The relative velocity (in the rotors frame of reference) at a certain point (x, y, z) is calculated by the following equation:

$$\mathbf{U}_{rel} = \mathbf{U} - \omega(z\mathbf{j} - y\mathbf{k}) \quad (3.1)$$

Where ω is the rotation rate, \mathbf{j} and \mathbf{k} are the unit vector in the y and z directions, respectively. This equation can be calculated by ParaView.

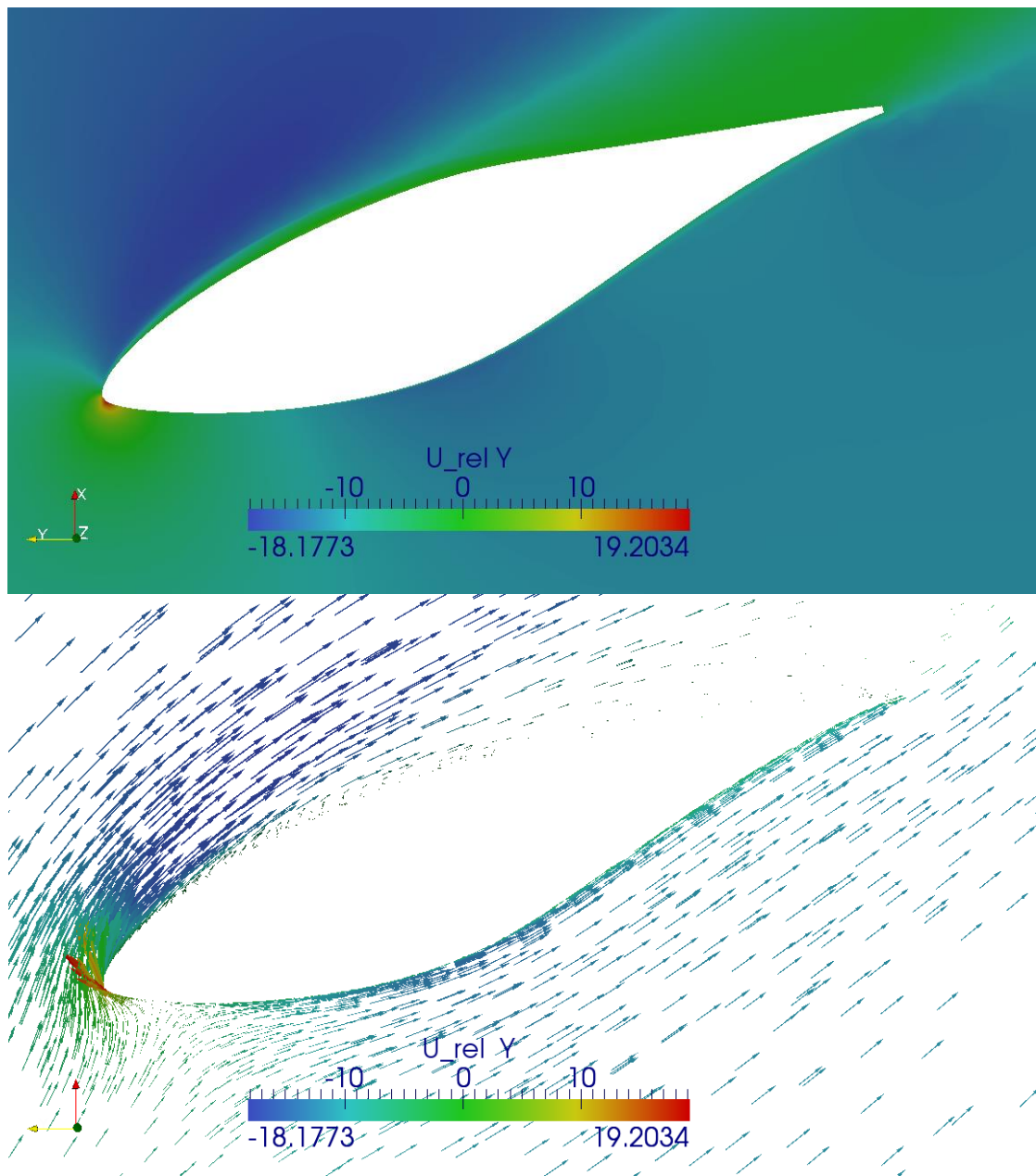


Figure 3.4 Relative velocity field and its vector field at 30% of the blade for 10m/s of inlet velocity.

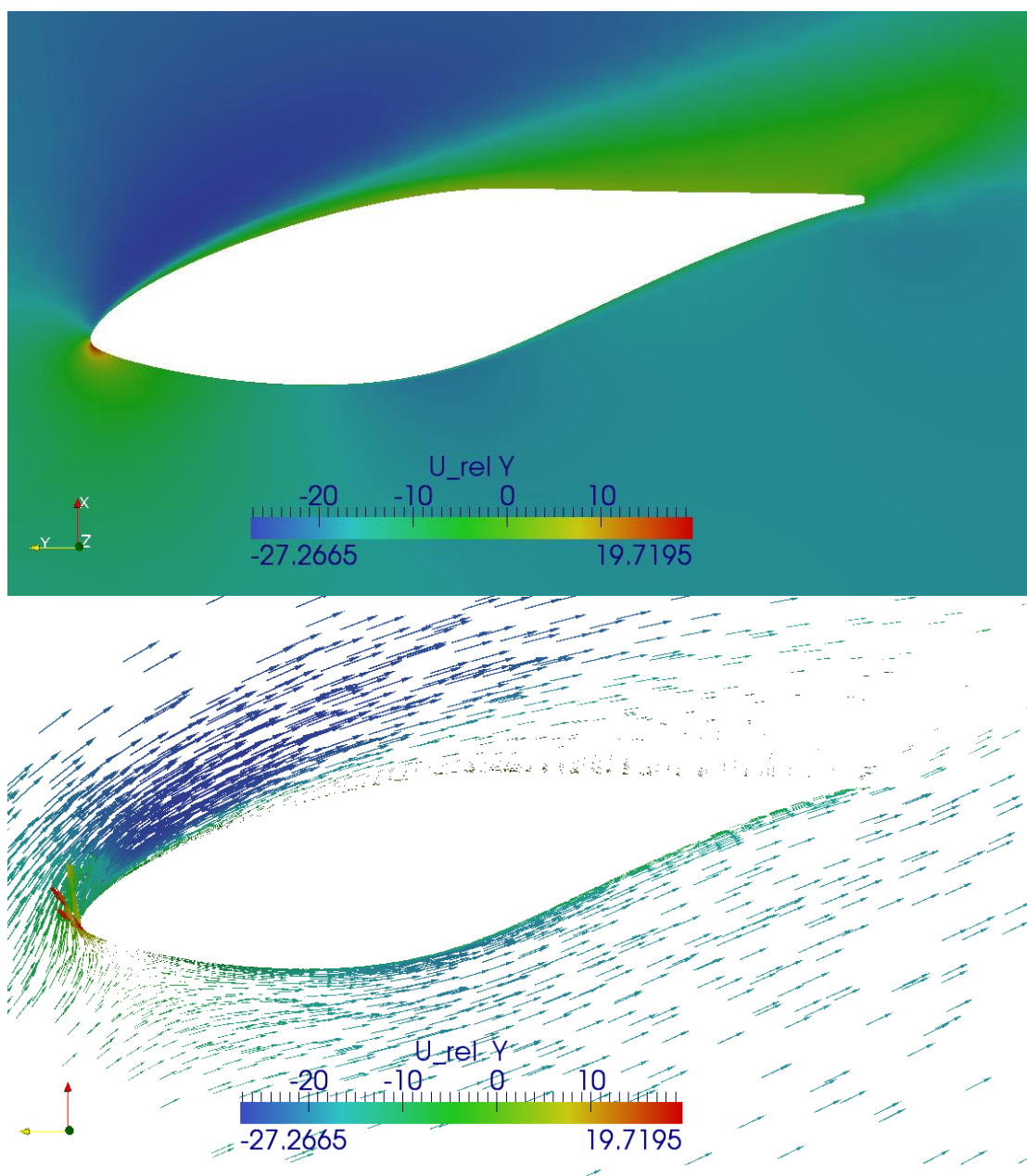


Figure 3.5 Relative velocity field and its vector field at 47% of the blade for 10m/s of inlet velocity.

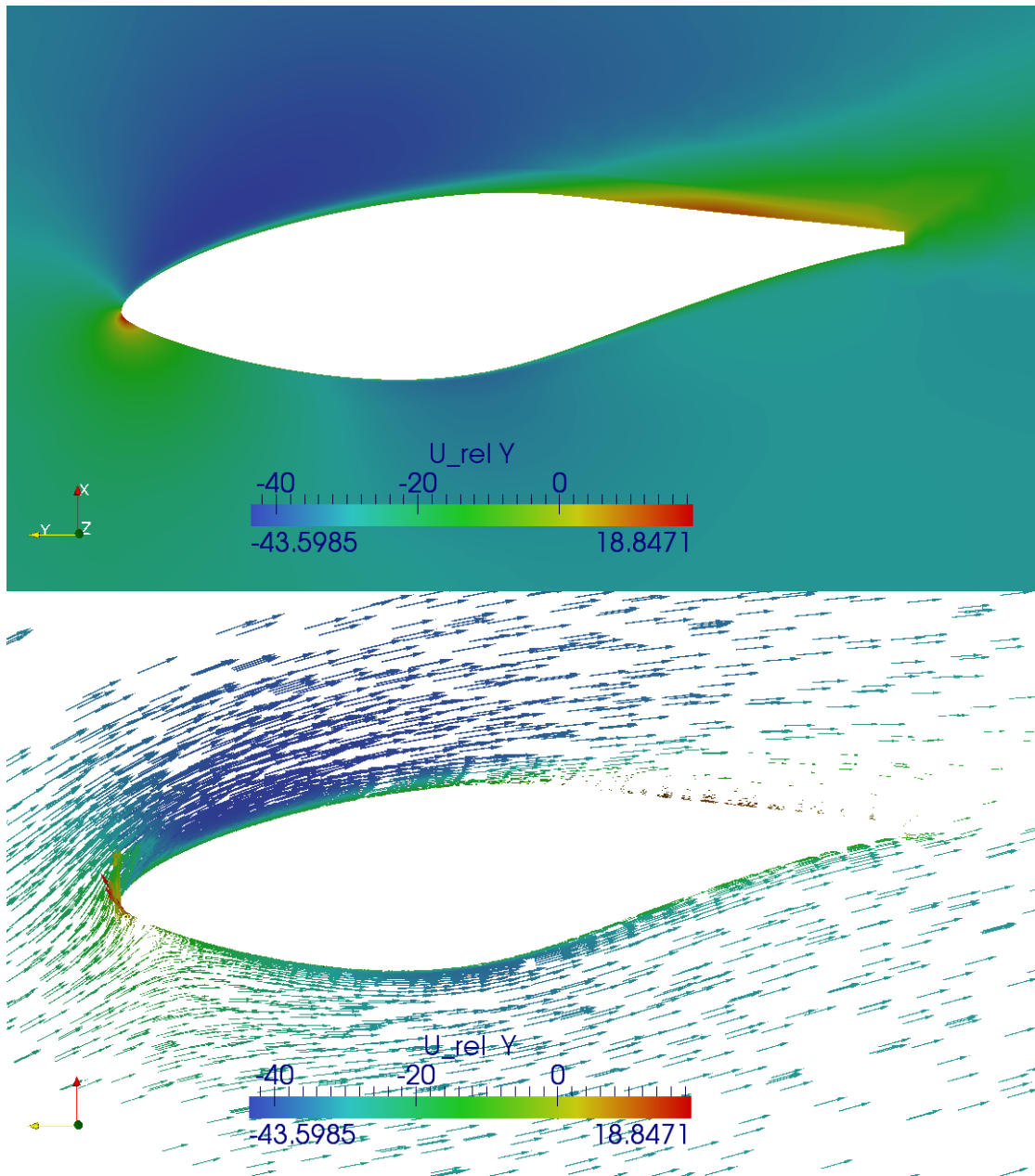


Figure 3.6 Relative velocity field and its vector field at 80% of the blade for 10m/s of inlet velocity.

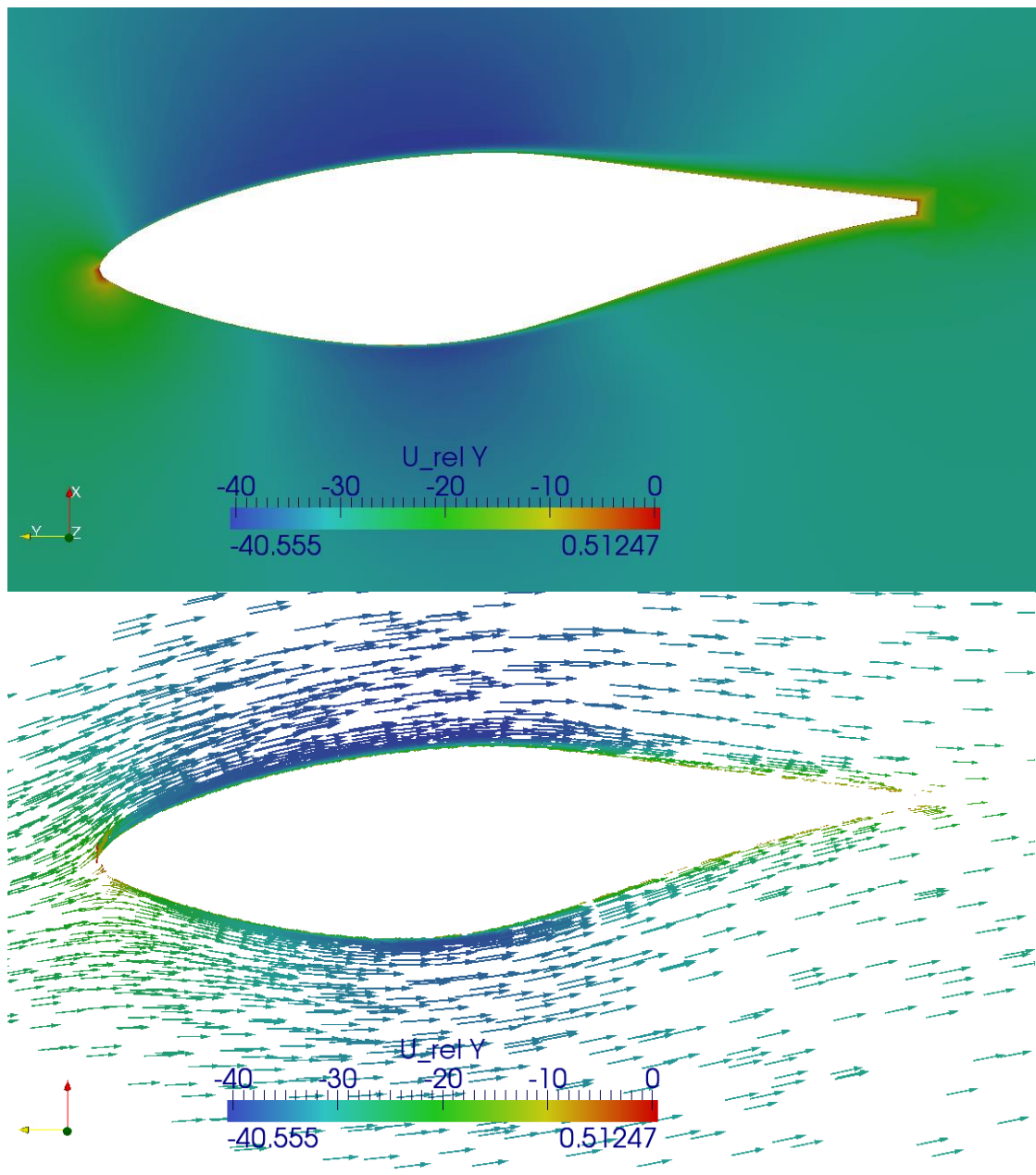


Figure 3.7 Relative velocity field and its vector field at 80% of the blade for 5m/s of inlet velocity.

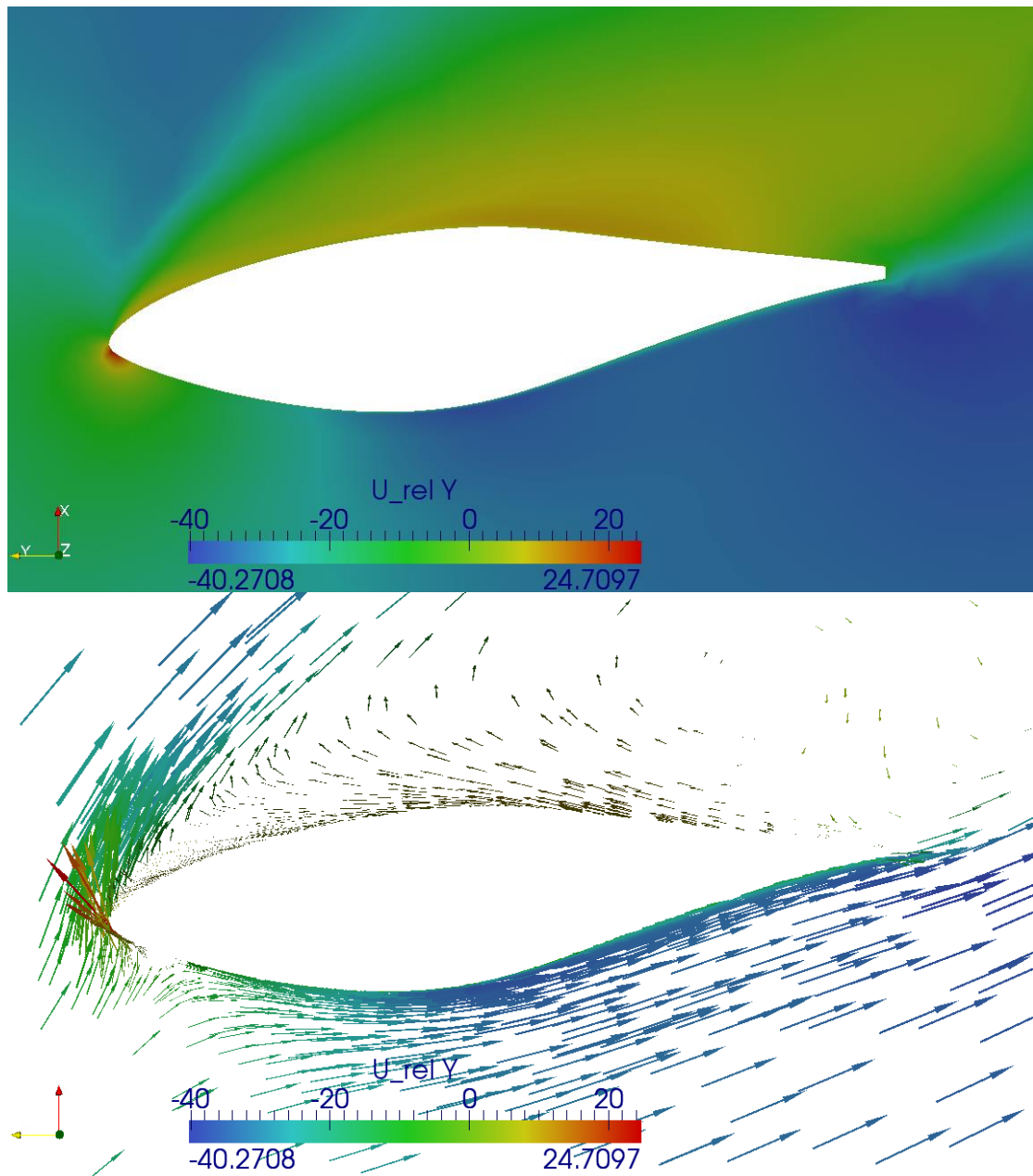


Figure 3.8 Relative velocity field and its vector field at 80% of the blade for 21m/s of inlet velocity.

3.2 Low-Speed Shaft Torque

In this section the low speed shaft torque (LSST) is computed for a series of simulations that all contain 10 million mesh cells but the inlet wind speed is varied from 5m/s to 21m/s. The results are shown in Figure 3.9, where it can be seen that the overall shape of the computed LSST curve is in general agreement with the experimental LSST curve. After 10 m/s the blades are almost entirely stalled. The CFD predictions however predict a stronger stall, and less torque, than found in the experiments. This is likely a result of the turbulence model. There are no turbulence models which are known to predict this type of strong stall well.

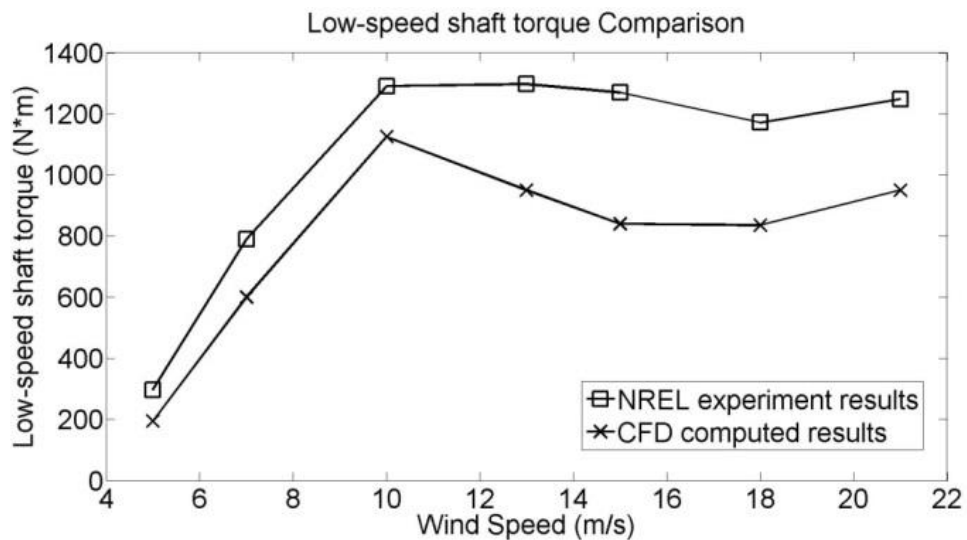


Figure 3.9 Comparison of low speed shaft torque for wind speeds of 5m/s, 7m/s, 10m/s, 13m/s, 15m/s, 18m/s and 21m/s.

Figure 3.10 shows the limiting streamlines on the suction side of the blade for the inlet velocity of 5m/s, 10m/s and 21m/s. We can see that at 5m/s of inlet wind speed,

the blade is operating as designed and has no stall effects. This is believed to be one explanation for the good agreement for the low inlet wind speed cases. At 10m/s, although the blade is stalled near the root, the tip region is behaving fine where the blade has a lower angle of attack. The 21m/s case is completely stalled so it has a poor agreement.



Figure 3.10 Limiting streamlines on the suction side of the blade for the inlet velocity of 5m/s, 10m/s and 21m/s

3.3 3D Effects

Streamlines and the second invariant of the velocity gradient tensor are used to examine the three-dimensionality of the flow on the rotating blades. Figure 3.11 shows the streamlines on the suction side of the blade for the 21 m/s case. All the streamlines originate at the root. Because of the strong stall, fluid is moving down the blade towards the tip. It is also moving towards the leading edge. When it reaches the leading edge it is swept off the blade in the separation shear layer.

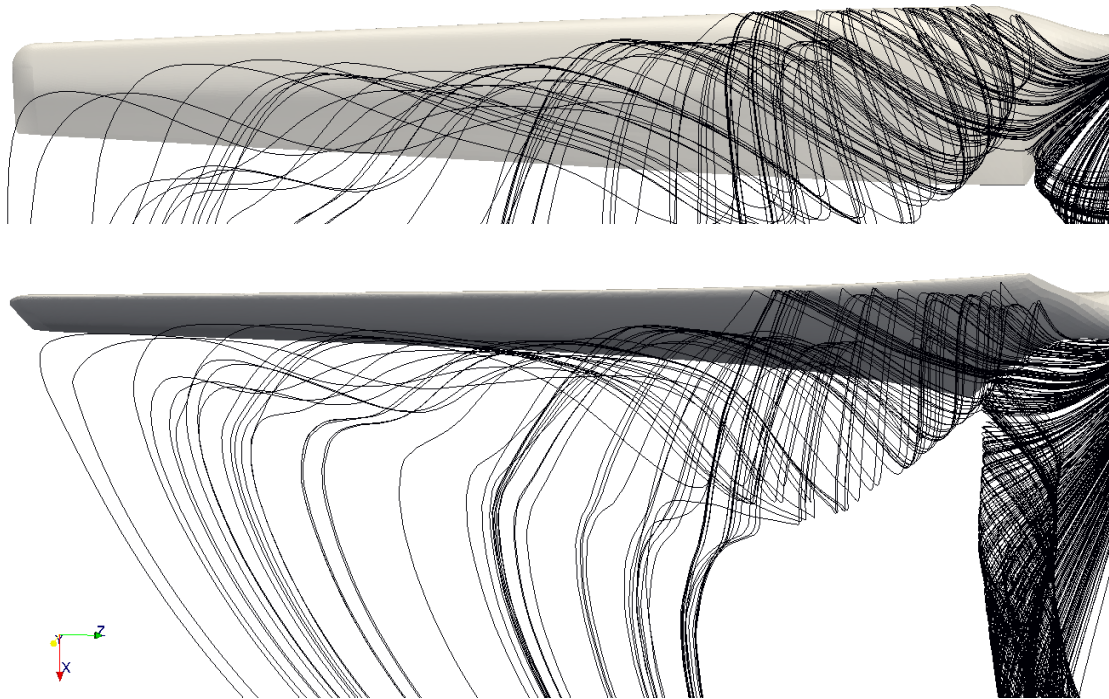


Figure 3.11 Two views of the streamlines on the suction side of the blade for 21 m/s case.

Figure 3.12 shows an iso-surface of the second invariant velocity gradient tensor at the value $Q = 0.3 \text{ 1/s}^2$ for the 10 m/s case. This invariant is a good indicator of vortices. In this case it clearly identifies the trailing tip vortices, and also an inner

pair of trailing vortices, that are even stronger, and that emanate from where the blades begin at the root. Figure 3.13 shows an iso-surface of the X velocity at $U_X = 8.1\text{m/s}$ also for the 10 m/s inflow case.



Figure 3.12 Iso-surface of the second invariant velocity gradient tensor at $Q = 0.3\text{ 1/s}^2$, for the 10 m/s case. This identifies the trailing vortices at the tip and the root.

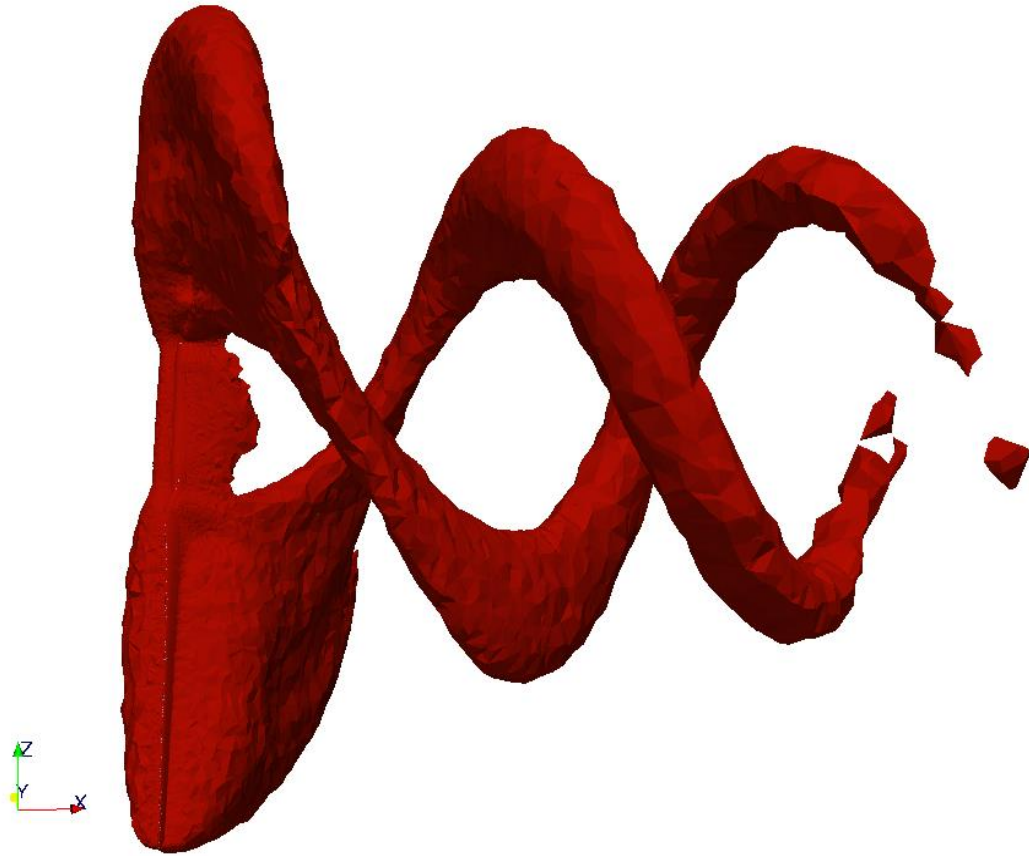


Figure 3.13 Iso-surface of X velocity at $U_X = 8.1$ m/s, for the 10m/s case.

3.4 Computational Costs

Tests of simulations under different numbers of parallel processors using different decomposing methods were also done. The results are presented in Table 3.1. This table shows average execution time and the clock time per timestep for simulations with different number of processors and different methods of decomposition. Execution time is the CPU time, while the clock time is execution time plus the time waiting on communication, which is the actual total time that each time step takes

(Younts et al., 2008). Those tests were performed on the supercomputer cluster, Cyclops with the same grid and the same settings including the numerical timestep. Based on the results we can conclude that the Scotch method is a more efficient decomposition method compared with the Simple method. And we can see that as the number of parallel computer processors increases, the clock time is not necessary decreasing. This is because the communication time will eventually increase with too many processors.

Table 3.1 Numerical performance of the simulation under different numbers of processors and different methods of decomposition.

Decomposition Method	Number of Processors	Execution Time (s)	Clock Time (s)
Simple	16	6	35
	32	4.3	21
	64	4.7	24
Scotch	16	10.5	29
	32	4.5	18
	64	4.1	18

CHAPTER 4

CONCLUSION

Under the situation of global energy depletion, it is important that renewable energy sources be investigated. This work aims to understand the difficulties of CFD simulation and attempts to develop a detailed CFD model that can be used to simulate the NREL Phase VI experiment, and thus make a small contribution to the renewable energy field. For more details of the geometry of NREL Phase VI blade, test configurations and available data sets, please refer to Hand et al., (2001).

A series of CFD simulations of the NREL Phase VI wind turbine with 0° yaw angle and 3° tip pitch angle at a rotation rate of 72 rpm were performed. Significant effort was made to refine the mesh at the leading edge and normal to the blade surface to accurately resolve the thin physical flow features in the velocity and pressure. Generally good agreement with the NREL experimental results was found for inlet wind speeds lower than 10m/s where the blades are not totally stalled. For inlet wind speeds higher than 10m/s, larger differences are observed between the simulations and experiments. It is likely these differences are due to the limitations of the turbulence model.

The importance of good mesh quality to a successful and accurate CFD prediction was analyzed by testing the CFD code performance with four different mesh resolutions. The shaft torque comparison shows that the computed CFD results are able to capture the basic trends of the NREL experimental results even though some

quantitative differences are observed. The three-dimensionality of the flow under separation conditions is shown to be very significant.

The results presented in Chapter 3 validate the working of the CFD solver PimpleDyMFoam and GGI interface. Other cases, such as wake analysis and tower modeling, might use the CFD framework developed in this work.

APPENDIX A

S809 AIRFOIL AND NREL PHASE VI WIND TURBINE DATA

Table A.1 S809 Airfoil Coordinates.

Upper Surface		Lower Surface	
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	1.00000	0.00000
1.00000	0.00000	0.00000	0.00000

Table A.2 NREL Phase VI blade chord and twist distributions.

Radial Distance r (m)	Span Station (r/5.532 m)	Span Station (r/5.029 m)	Chord Length (m)	Twist (degrees)	Thickness (m)	Twist Axis (% chord)
0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.508	0.092	0.101	0.218	0.000	0.218	50.000
0.660	0.120	0.131	0.218	0.000	0.218	50.000
0.883	0.160	0.176	0.183	0.000	0.183	50.000
1.008	0.183	0.200	0.349	6.700	0.163	35.900
1.067	0.193	0.212	0.441	9.900	0.154	33.500
1.133	0.205	0.225	0.544	13.400	0.154	31.900
1.257	0.227	0.250	0.737	20.040	0.154	30.000
1.343	0.243	0.267	0.728	18.074	20.95% chord	30.000
1.510	0.273	0.300	0.711	14.292	20.95% chord	30.000
1.648	0.298	0.328	0.697	11.909	20.95% chord	30.000
1.952	0.353	0.388	0.666	7.979	20.95% chord	30.000
2.257	0.408	0.449	0.636	5.308	20.95% chord	30.000
2.343	0.424	0.466	0.627	4.715	20.95% chord	30.000
2.562	0.463	0.509	0.605	3.425	20.95% chord	30.000
2.867	0.518	0.570	0.574	2.083	20.95% chord	30.000
3.172	0.573	0.631	0.543	1.150	20.95% chord	30.000
3.185	0.576	0.633	0.542	1.115	20.95% chord	30.000
3.476	0.628	0.691	0.512	0.494	20.95% chord	30.000
3.781	0.683	0.752	0.482	-0.015	20.95% chord	30.000
4.023	0.727	0.800	0.457	-0.381	20.95% chord	30.000
4.086	0.739	0.812	0.451	-0.475	20.95% chord	30.000
4.391	0.794	0.873	0.420	-0.920	20.95% chord	30.000
4.696	0.849	0.934	0.389	-1.352	20.95% chord	30.000
4.780	0.864	0.950	0.381	-1.469	20.95% chord	30.000
4.938	0.893	0.982	0.365	-1.689	20.95% chord	30.000
5.000	0.904	0.994	0.358	-1.775	20.95% chord	30.000
5.029	0.909	1.000	0.356	-1.815	20.95% chord	30.000
5.305	0.959	1.055	0.328	-2.191	20.95% chord	30.000
5.532	1.000	1.100	0.305	-2.500	20.95% chord	30.000

APPENDIX B

IMPLEMENTATION OF GENERALIZED GRID INTERFACE (GGI) IN OpenFOAM

For a detailed understanding of the GGI in OpenFOAM, *boundary* and *dynamicMeshDict* files are attached below. The physical properties of GGI patches are set in *boundary* while the dynamic mesh configurations are given in *dynamicMeshDict*.

boundary:

```
(
  GGIOUTSIDE //Shadow GGI Patch
  {
    type            ggi;
    nFaces          30914;
    startFace       29906040;
    shadowPatch     GGIINSIDE;
    zone            GGIOUTSIDE_zone;
    bridgeOverlap   true;
  }
  S809DOMAININLET
  {
    type            patch;
    nFaces          7368;
    startFace       29936954;
  }
  S809DOMAINOUTLET
  {
    type            patch;
    nFaces          7374;
    startFace       29944322;
  }
  S809DOMAINBACKFACE
  {
    type            wall;
    nFaces          12338;
```

```

        startFace      29951696;
    }
    S809DOMAINFRONTFACE
    {
        type            wall;
        nFaces          12362;
        startFace      29964034;
    }
    S809DOMAINTOP
    {
        type            wall;
        nFaces          17554;
        startFace      29976396;
    }
    S809DOMAINBOTTOM
    {
        type            wall;
        nFaces          17764;
        startFace      29993950;
    }
    S809BLADES
    {
        type            wall;
        nFaces          223808;
        startFace      30011714;
    }
    GGIINSIDE //Master GGI Patch
    {
        type            ggi;
        nFaces          30914;
        startFace      30235522;
        shadowPatch     GGIOUTSIDE;
        zone            GGIINSIDE_zone;
        bridgeOverlap   true;
    }
)

```

dynamicMeshDict:

```

dynamicFvMeshLib "libtopoChangerFvMesh.so"; // "topoChangerFvMesh.so";
dynamicFvMesh    mixerGgiFvMesh;
//dynamicFvMesh  dynamicMotionSolverFvMesh;

```



```

mixerGgiFvMeshCoeffs
{
    coordinateSystem
    {
        type            cylindrical;
        origin           (0 0 0);
        axis             (1 0 0);
        direction        (0 0 1);
    }

    rpm                 -72.0;    // Rotation speed, in rpm

    slider
    {
        moving (GGIINSIDE); //Moving Patch
        static (GGIOUTSIDE); //Static Patch
    }
}

```

APPENDIX C

NUMERICAL SCHEMES AND SOLUTION CONTROL IN OpenFOAM

The numerical schemes and the solution parameters are controlled by the *fvSchemes* and *fvSolution* files in OpenFOAM. The detailed settings are attached below.

fvSchemes:

```
ddtSchemes
{
    default          Euler;//steadyState;
}

gradSchemes
{
    default          Gauss linear;
    grad(p)          Gauss linear;
    grad(U)          Gauss linear;
}

divSchemes
{
    default          none;
    div(phi,U)       Gauss upwind;
    div(phi,k)       Gauss upwind;
    div(phi,epsilon) Gauss upwind;
    div(phi,R)       Gauss upwind;
    div(R)           Gauss linear;
    div(phi,nuTilda) Gauss upwind;
    div((nuEff*dev(grad(U).T()))) Gauss linear;
}

laplacianSchemes
{
    default          none;
    laplacian(nu,U)  Gauss linear corrected;
```

```

    laplacian(rAU,pcorr) Gauss linear corrected;
    laplacian(rAU,p) Gauss linear corrected;
    laplacian(nuEff,U) Gauss linear limited 0.5;
    laplacian((1|A(U)),p) Gauss linear limited 0.5;
    laplacian(DkEff,k) Gauss linear limited 0.5;
    laplacian(DepsilonEff,epsilon) Gauss linear limited 0.5;
    laplacian(DREff,R) Gauss linear limited 0.5;
    laplacian(DnuTildaEff,nuTilda) Gauss linear limited 0.5;
}

interpolationSchemes
{
    default          linear;
    interpolate(U)    linear;
}

snGradSchemes
{
    default          limited 0.5;
}

fluxRequired
{
    default          no;
    pcorr;
    p;
}

```

fvSolution:

```

solvers
{
    p
    {
        solver          PCG;
        preconditioner   DIC;
        tolerance        1e-07;
        relTol           0;
    }
    maxIter             50;
}

pFinal
{

```

```

        solver          PCG;
        preconditioner   DIC;
        tolerance        1e-07;
        relTol           0;
maxIter          50;
    }

    U
    {
        solver          PBiCG;
        preconditioner   DILU;
        tolerance        1e-07;
        relTol           0;
maxIter          10;
    }

    UFinal
    {
        solver          PBiCG;
        preconditioner   DILU;
        tolerance        1e-07;
        relTol           0;
maxIter          0;
    }

    k
    {
        solver          PBiCG;
        preconditioner   DILU;
        tolerance        1e-05;
        relTol           0;
    }

    epsilon
    {
        solver          PBiCG;
        preconditioner   DILU;
        tolerance        1e-05;
        relTol           0;
    }

    nuTilda
    {
        solver          PBiCG;

```

```

        preconditioner DILU;
        tolerance      1e-20;
        relTol         0;
    maxIter            5;
    }
}

PIMPLE
{
    nOuterCorrectors 2;
    nCorrectors       3;
    nNonOrthogonalCorrectors 1;
    pRefCell          0;
    pRefValue         0;
}

relaxationFactors
{
    p            0.6;
    U            0.7;
    nuTilda      0.7;
    k            1;
    epsilon      1;
}

```

APPENDIX D

RELATIVE VELOCITY FIELD AT VARIOUS BLADE SECTIONS FOR DIFFERENT INLET WIND SPEEDS

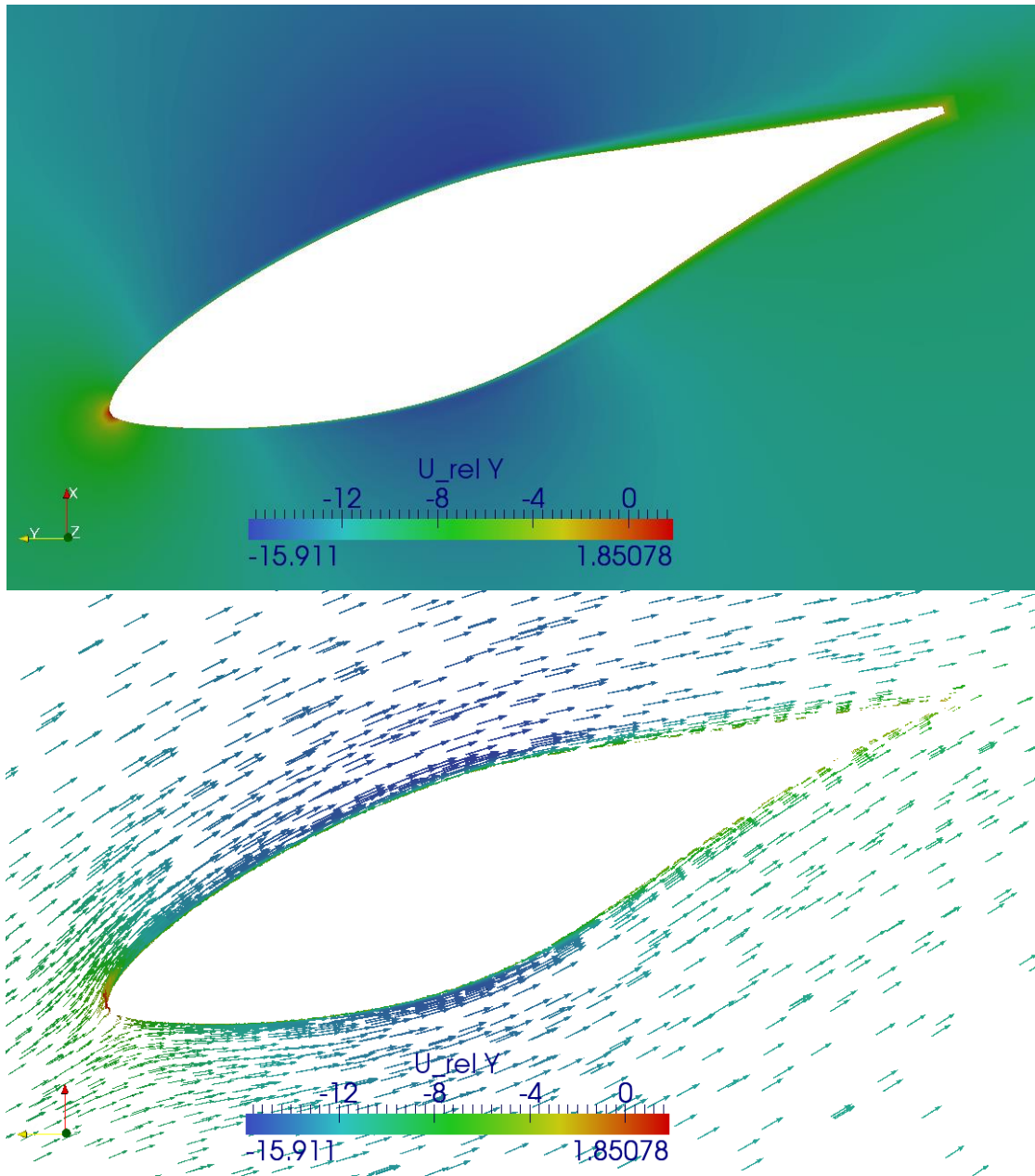


Figure D.1 Relative velocity field and its vector field at 30% of the blade for 5m/s of inlet velocity.

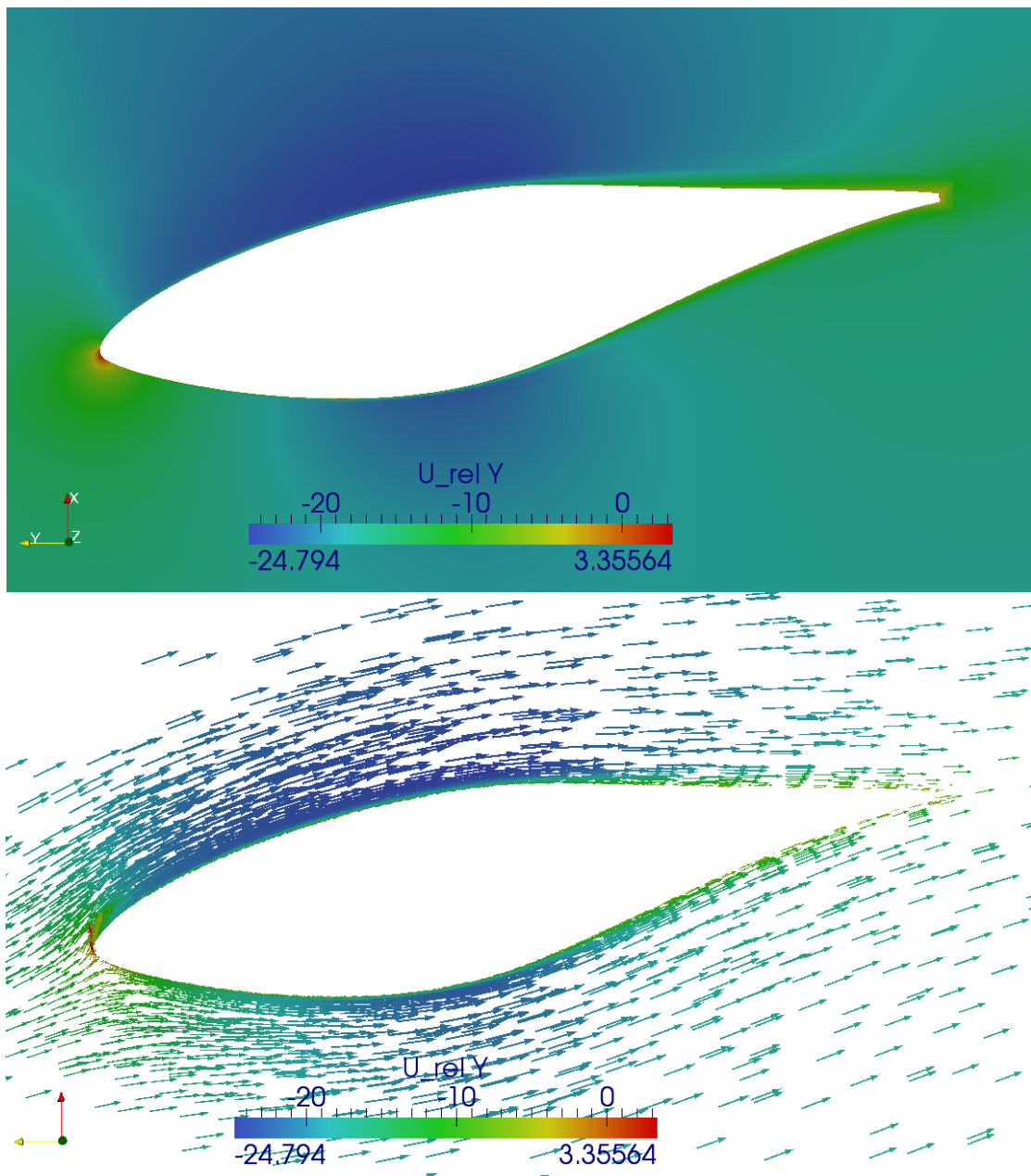


Figure D.2 Relative velocity field and its vector field at 47% of the blade for 5m/s of inlet velocity.

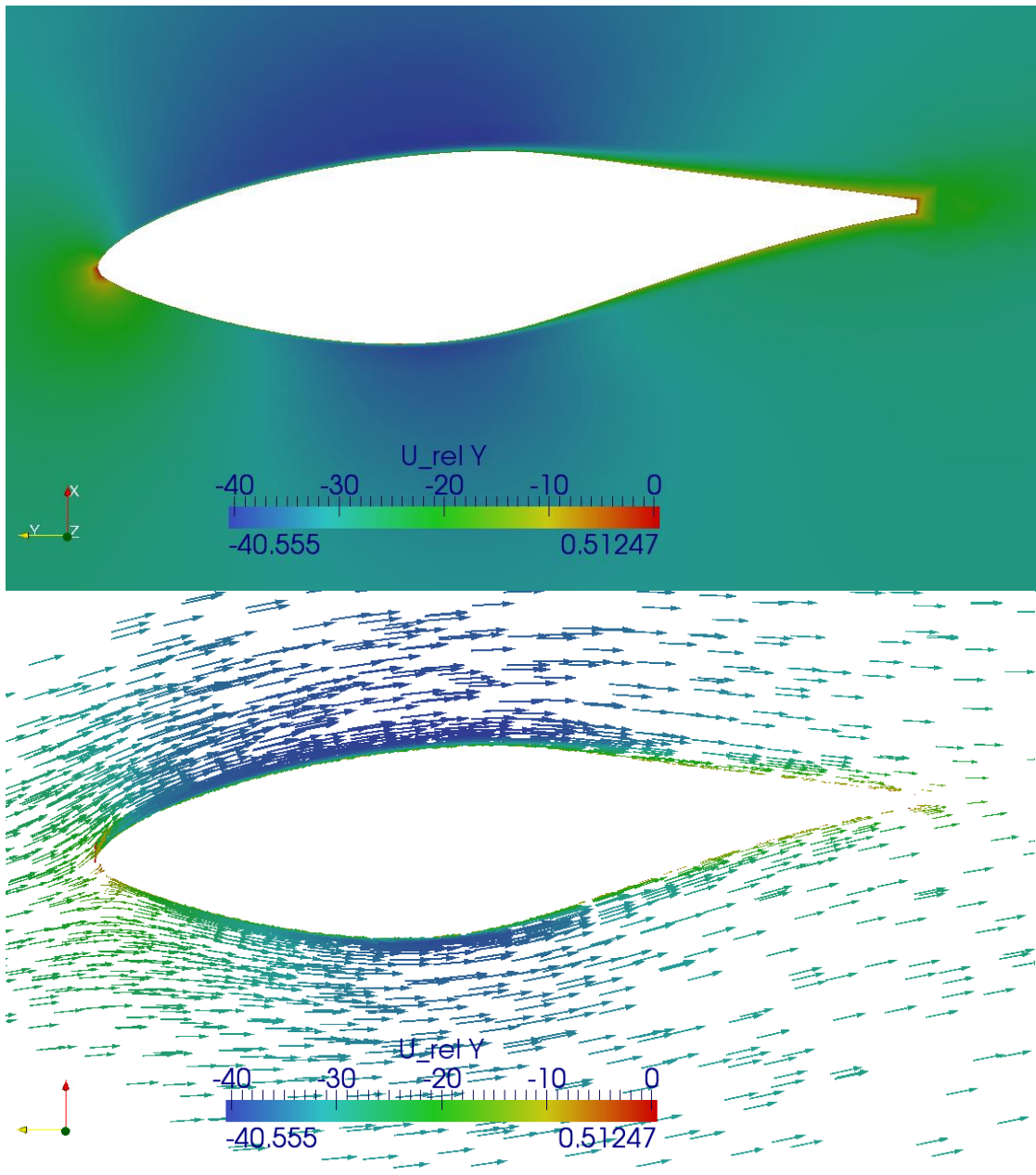


Figure D.3 Relative velocity field and its vector field at 80% of the blade for 5m/s of inlet velocity.

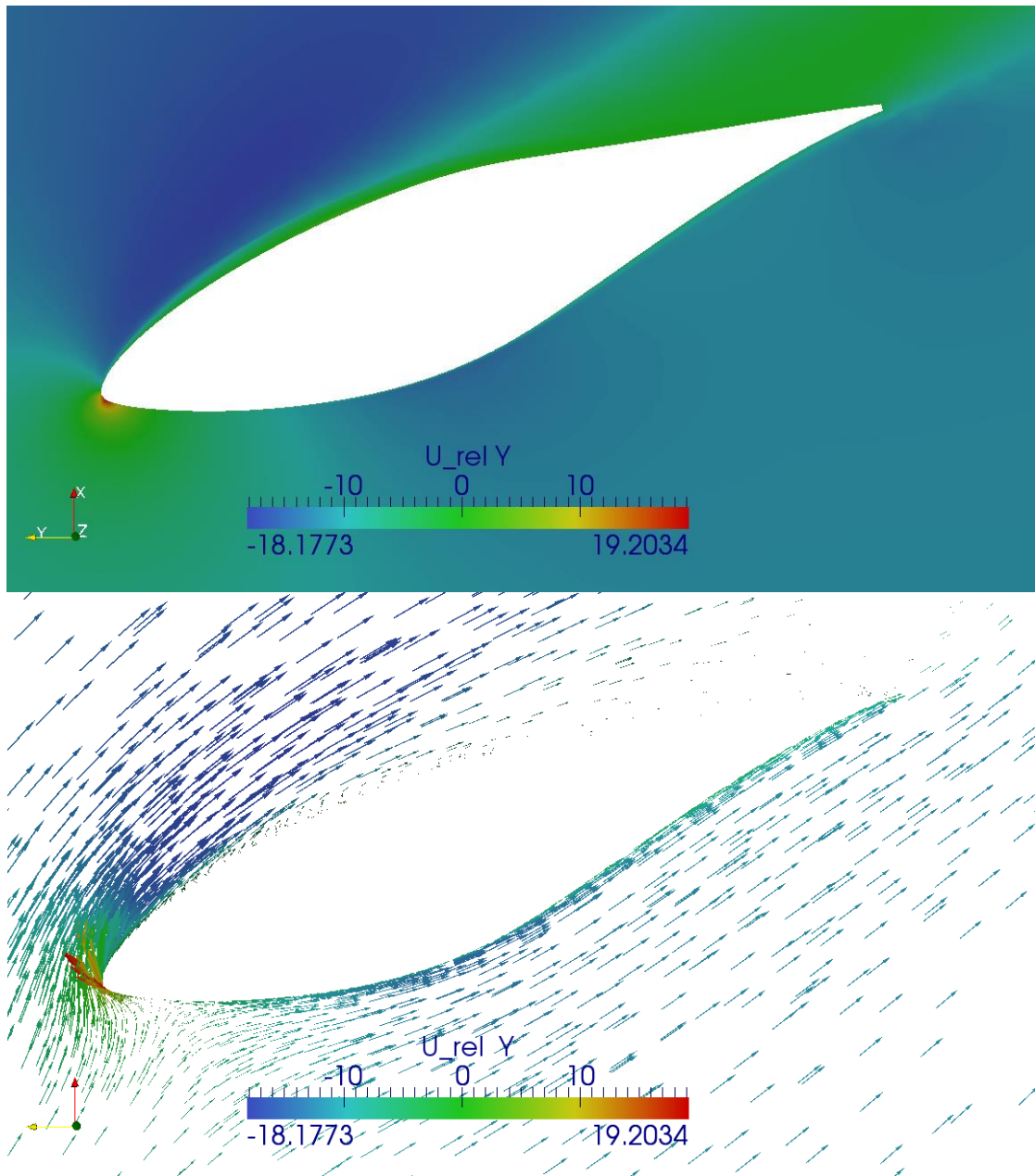


Figure D.4 Relative velocity field and its vector field at 30% of the blade for 10m/s of inlet velocity.

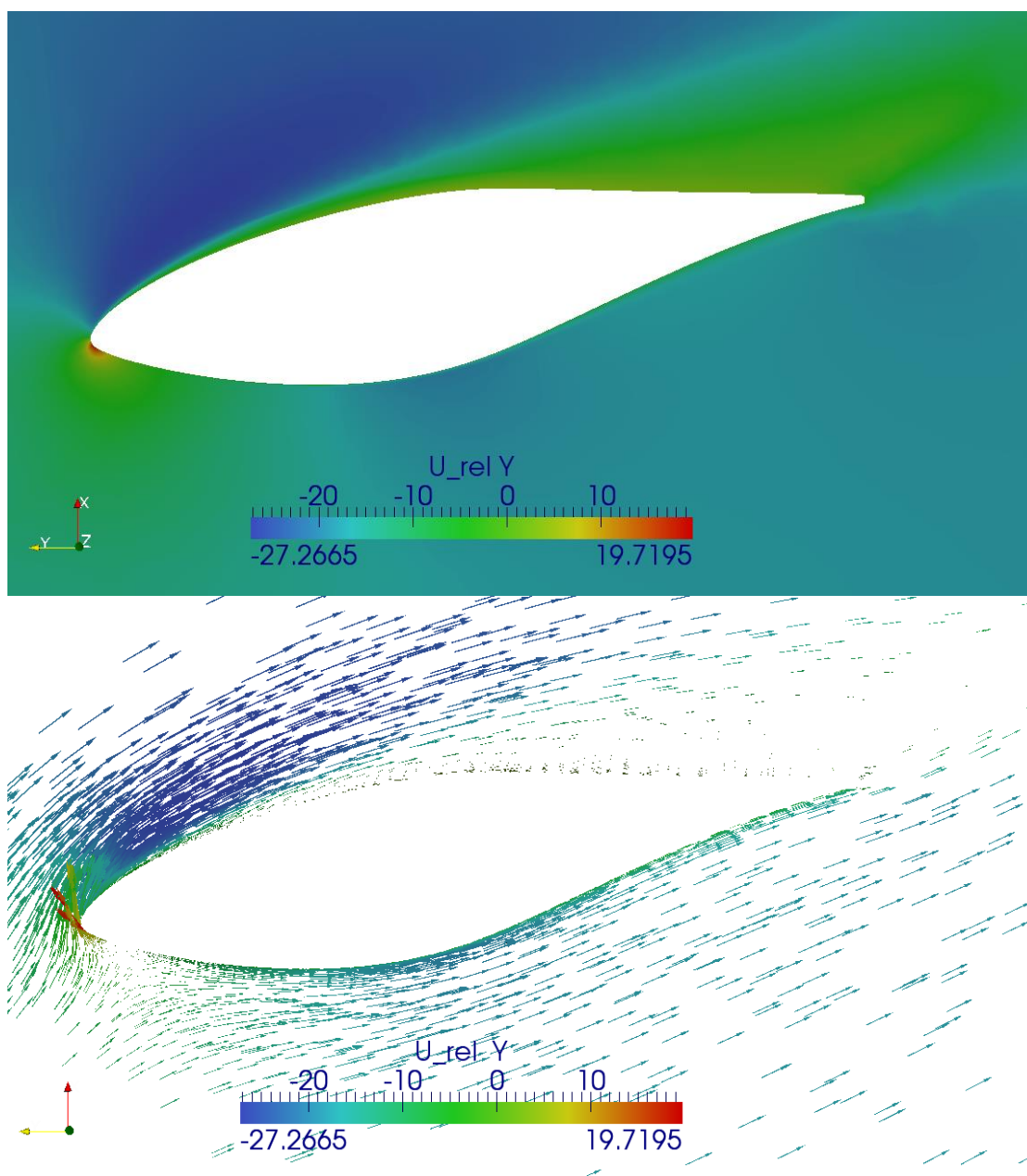


Figure D.5 Relative velocity field and its vector field at 47% of the blade for 10m/s of inlet velocity.

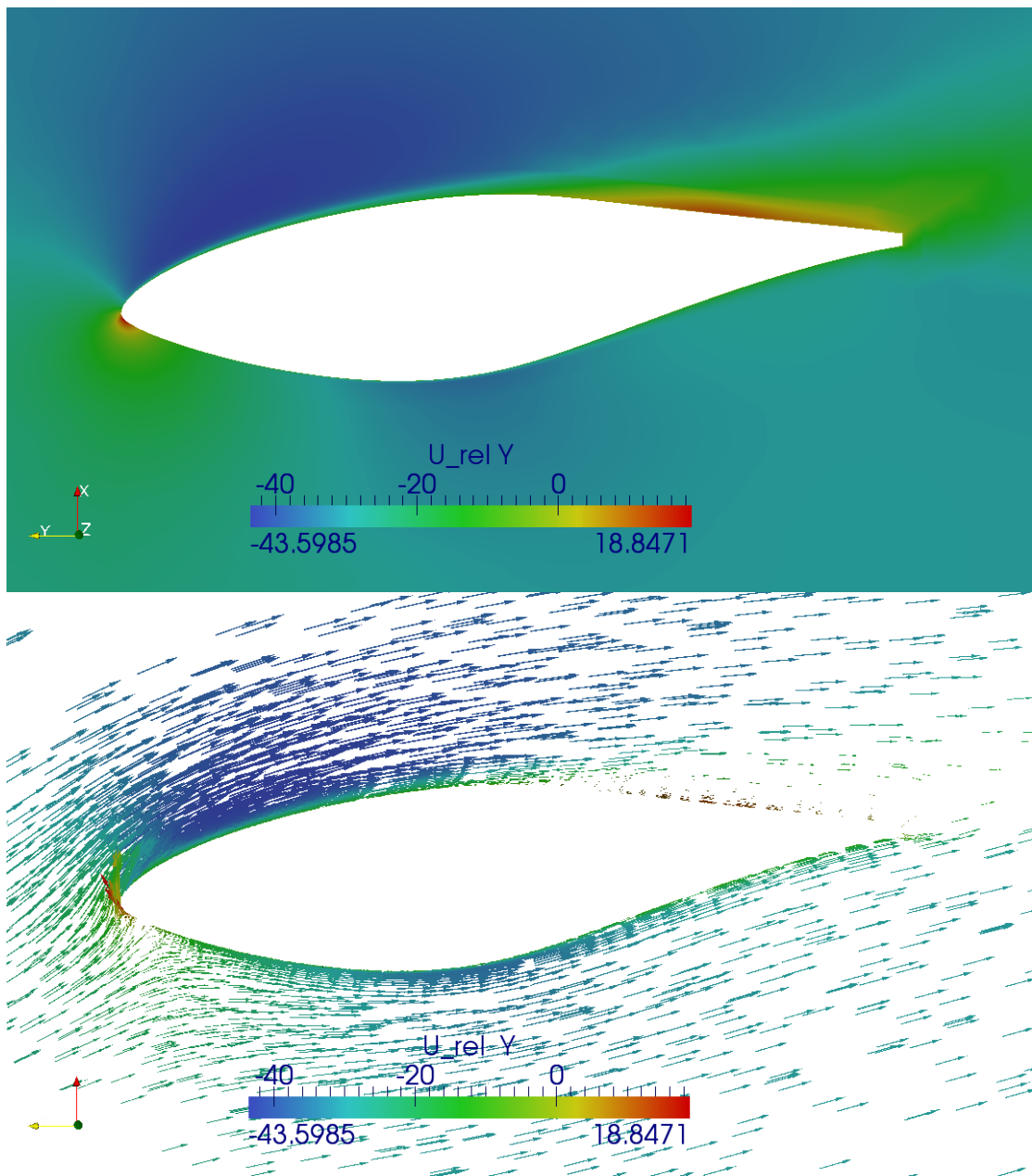


Figure D.6 Relative velocity field and its vector field at 80% of the blade for 10m/s of inlet velocity.

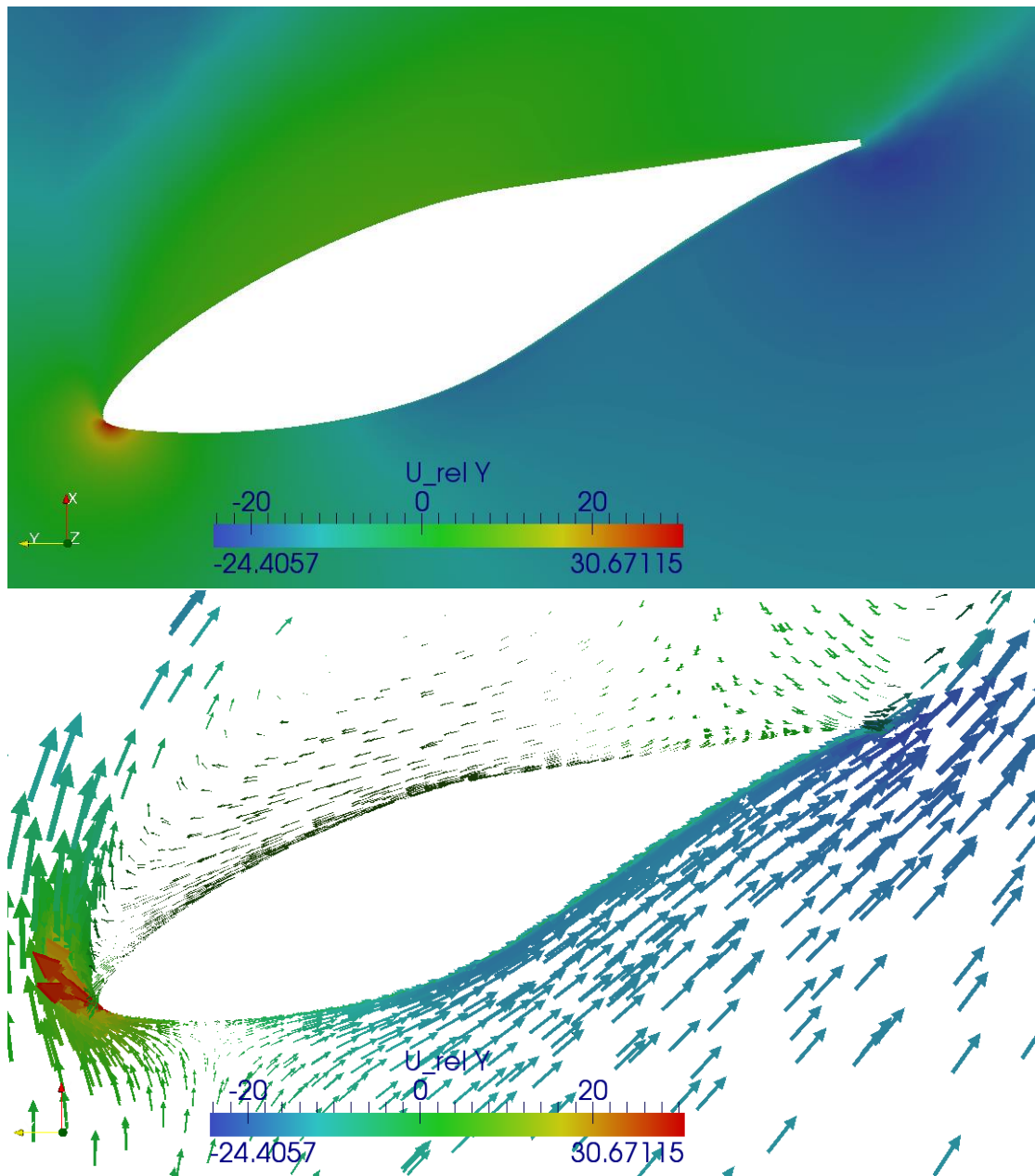


Figure D.7 Relative velocity field and its vector field at 30% of the blade for 21m/s of inlet velocity.

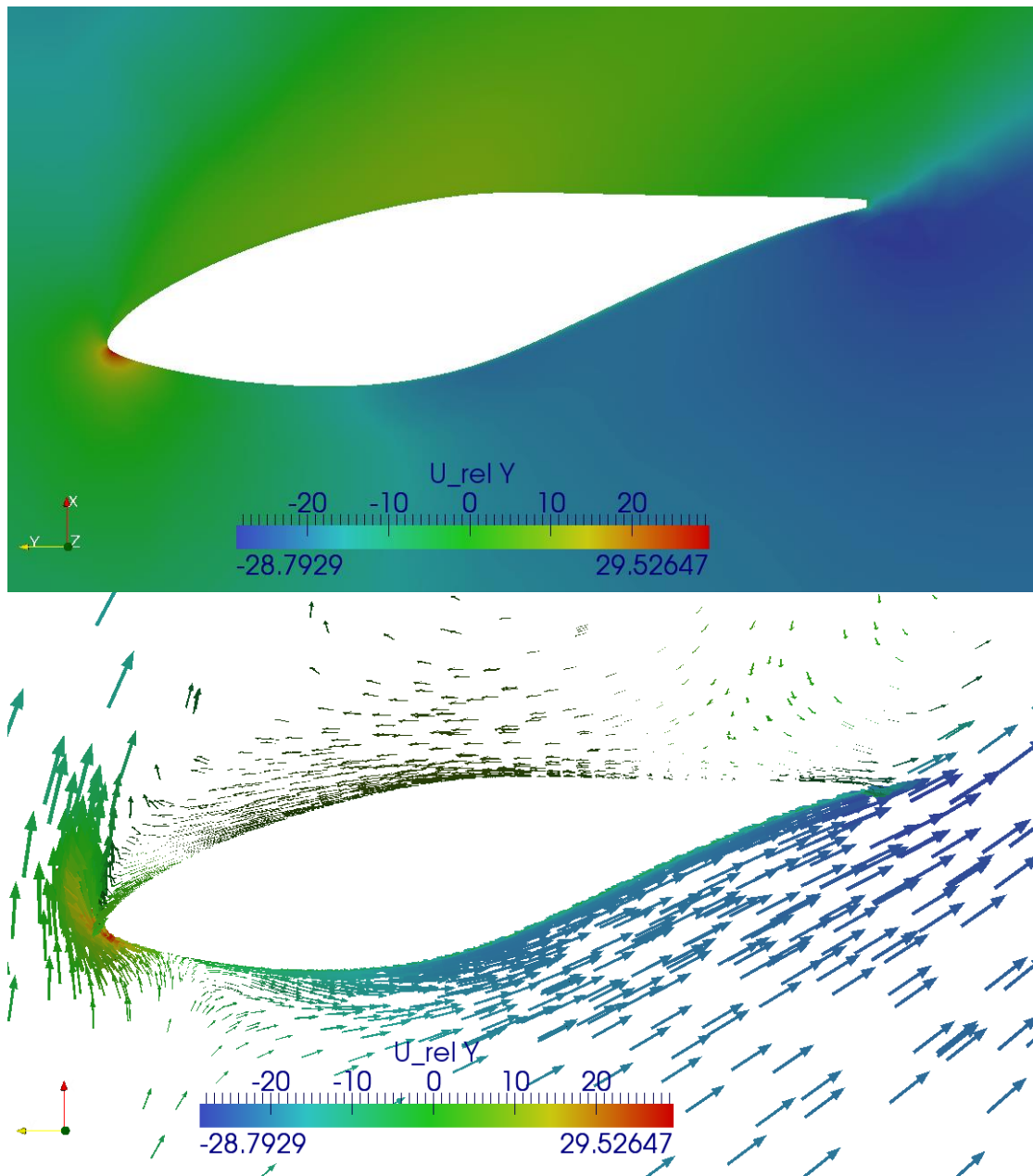


Figure D.8 Relative velocity field and its vector field at 47% of the blade for 21m/s of inlet velocity.

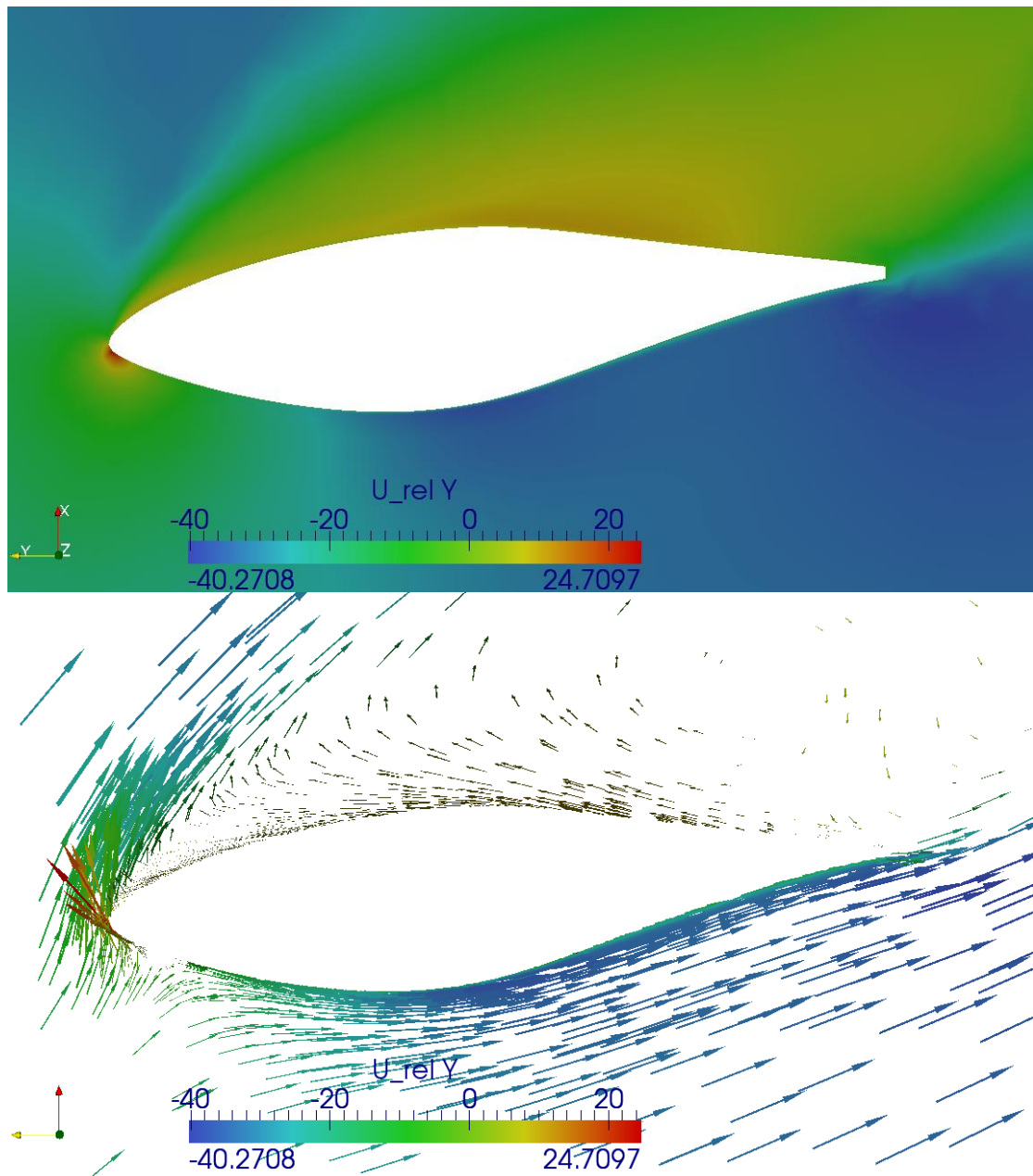


Figure D.9 Relative velocity field and its vector field at 80% of the blade for 21m/s of inlet velocity.

BIBLIOGRAPHY

- [1] ANSYS (2010). ICEM CFD - computational fluid dynamics preprocessor. <http://www.ansys.com/Products/Other+Products/ANSYS+ICEM+CFD>.
- [2] Duque EPN, Burklund MD (2003), Johnson W. Navier-Stokes and comprehensive analysis performance predictions of the NREL Phase VI experiment. *Journal of Solar Energy Engineering* 125, 457-467.
- [3] Gonzalez A, Munduate X (2008). Three dimensional and rotational aerodynamics on NREL Phase VI Wind Turbine Blade. *Journal of Solar Energy Engineering* 130, 1-7.
- [4] Hand MM, Simms DA, Fingersh LJ and *et al.* (2001). Unsteady Aerodynamics Experiment Phase VI: wind tunnel test configurations and available data campaigns. National Renewable Energy Laboratory (NREL).
- [5] Hansen GA, Douglass RW, Zardecki A (2005). Mesh Enhancement: Selected Elliptic Methods, Foundations and Applications. Imperial College Press.
- [6] Jasak H (1996). Error analysis and estimation for the finite volume method with applications to fluid flows. PhD thesis, Imperial College of Science, Technology and Medicine.
- [7] Javaherchi T (2010). Review of Spalart-Allmaras turbulence model and its modifications. University of Washington.
- [8] Laino DJ, Hansen AC, Minnema JE (2002). Validation of AERODYN Subroutines using NREL unsteady aerodynamics experiment data. *ASME 2002 Wind Energy Symposium*.
- [9] Manwell JF, McGowan JG, Rogers AL (2009). Wind Energy Explained: Theory, Design, and Application. WILEY.
- [10] OpenFOAM (2011). User Guide. <http://www.openfoam.org/docs/user/index.php>.
- [11] Owen SJ (1998). A Survey of Unstructured Mesh Generation Technology. Carnegie Mellon University and ANSYS Inc.
- [12] Robinson MC, Hand MM, Simms DA, Schreck SJ (1999). Horizontal axis wind turbine aerodynamics: Three dimensional, unsteady and separated flow influences. National Renewable Energy Laboratory (NREL).
- [13] Schmitt P (2009). OpenFOAM: Using GGI for rotating objects.
- [14] Schmitz S, Chattot JJ (2006). Characterization of three dimensional effects for the

- rotating and parked NREL Phase VI Wind Turbine. *Journal of Solar Energy Engineering* 128, 445-454.
- [15] Sorensen NN, Michelsen JA, Schreck S (2002). Navier-Stokes predictions of the NREL Phase VI rotor in the NASA Ames 80ft \times 120ft wind tunnel. *Wind Energy* 5, 151-169.
- [16] Spalart PR, Allmaras SR (1992). A one equation turbulence model for aerodynamic flows. AIAA 92-0439, AIAA 30th Aerospace Sciences Meeting and Exhibit.
- [17] Squillacote AH (2007). The ParaView guide : A Parallel visualization application. Kitware.
- [18] Tangler JL (2002). The nebulous art of using wind tunnel airfoil of using wind tunnel airfoil data for predicting rotor performance. National Renewable Energy Laboratory (NREL).
- [19] Wilcox DC (2007). Basic Fluid Mechanics. DCW Industries.
- [20] Xu G, Sankar LN (2000). Computational study of horizontal axis wind turbines. *Journal of Solar Energy Engineering* 122, 35-39.
- [21] Younts A, Howard A, Smith PM and *et al.* (2008). Bring disruptive technology to competition. Purdue University.