Simulation of Large, Horizontal Axis Wind Turbines.

Validation of Rigid Wind Turbine Simulations Using NREL Phase VI Experiment

-

Bachelor Thesis

Author: Roberto Palacios González (Matr.-Nr.: 350634)

Supervisor: A. Emre Öngüt, M.Sc.

Aachen, April 21, 2015

CHAIR FOR COMPUTATIONAL ANALYSIS OF TECHNICAL SYSTEMS

RWTH AACHEN UNIVERSITY

Contents

List of Figures

List of Tables

1. Introduction

1.1. Importance of Wind Energy

Reduction of fossil fuel consumption is one of the best arguments for supporting the use of wind power, therefore it is an alternative to reduce fossil fuel dependence. In addition, wind energy is not only a renewable source, which means it can be produced again and again since it is available in plenty, but also it does not pollute at all. Furthermore the cost of producing wind energy devices has come down steadily over the last few years.

All these facts mean that wind energy became an important energy source for Europe in the last years, and keeps growing. Today the installed wind power capacity in the European Union totaled about 117,300 MW, and the EU wind industry has had a compound annual growth rate of 10% between the years 2000 and 2013. According to the current capacity in one year the wind power installed would produce 257 TWh of electricity, enough to supply 8% of the EU's electricity consumption.

In the future, wind power is likely to continue to grow in the European Union. According to a European Environment Agency report, wind energy can play a major role in achieving the European renewable energy targets.

The European Wind Energy Association estimates that 230 GW of wind capacity will be installed in Europe by 2020. This would produce 14-17% of the EU's electricity, avoiding 333 million tones of CO2 per year and saving Europe 28 billion a year in avoided fuel costs.[1]

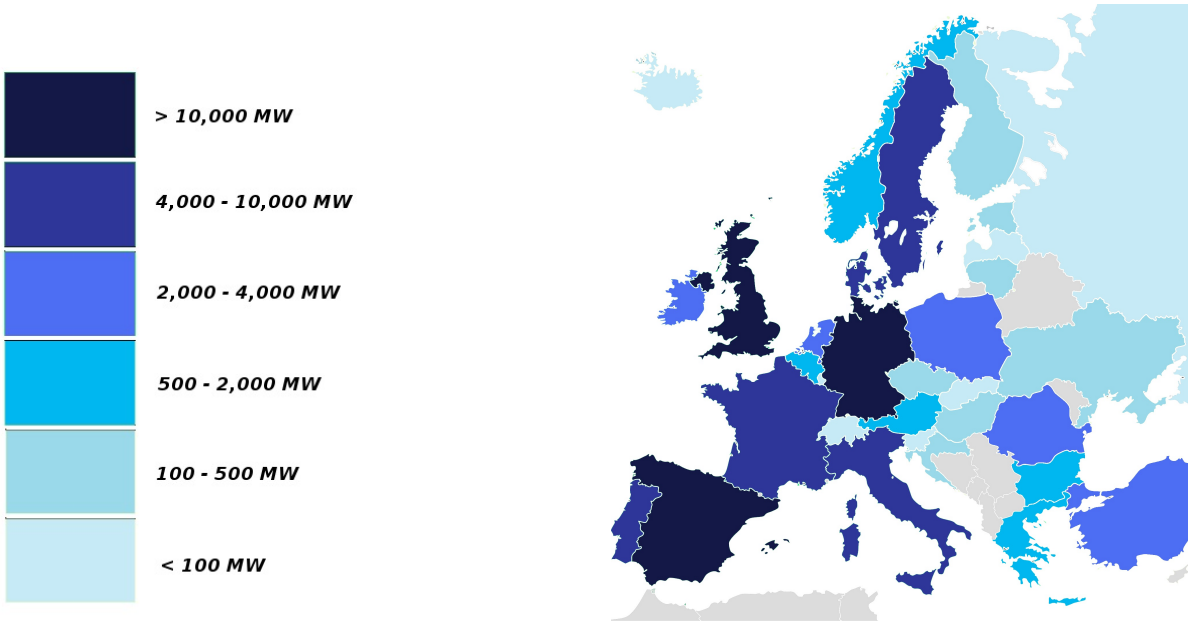


Figure 1.1: Wind power installed in Europe 2012.

1.2. Thesis Goals

Chair for Computational Analysis of Technical Systems (CATS) of RWTH Aachen University is investigating the effects of active control mechanisms for reducing the dynamic loads on the wind turbine drive train in order to increase the performance and reliability of large-scale wind turbines. This study is conducted within the context of the project "Aeroelastic Simulation of Large-Scale Wind Turbines", which is funded by the Federal Ministry of Environment, Nature Conservation and Nuclear Safety. CATS institute has already experience in fluid-structure interaction problems by using various fluid and structural solvers. In this project, the in-house finite element fluid solver XNS is used together with the computational aerolasticity software package 'Solid-Fluid Interaction' (SOFIA). This project is conducted in collaboration with the Institute for Machine Elements and Machine Design (IME) of the RWTH Aachen University.

The aims of this thesis are the generation of rigid wind turbine simulations according to NREL PHASE VI experiment, and compare the results. Therefore, the simulations can be validated if the results obtained are as expected. This project also has the goal to create a detailed systematic pattern of every step of the generation of the wind turbine simulation creation, which can be used for subsequent research as a guide.

2. Theory

2.1. Aerodynamics of Wind Turbines and Betz's Limit

The objective of this section is to understand the physical basics of energy conversion from wind turbines. The calculation of loads and the resolution of dynamic problems is only possible by a deep knowledge of the aerodynamics process in the rotor.

Wind turbines extract power from wind by removing momentum from the flow, for this reason it would be useful to know what is the maximum limit of energy removable, which is called Betz's limit. In order to calculate Betz's limit it is necessary to place the problem in an ideal situation, where all energy dissipating effects are removed. In this explanation all these points have been assumed:

1. It will be assumed that air is an ideal flow. It is not far from the reality since movement around a wind turbine happens at high Reynolds's numbers and low Mach numbers.
2. In order to simplify the study, it will be assumed that the incident wind is one-dimensional and with uniform pressure, density and velocity. Also the wind turbine will be distant enough from any obstacle for their influence to be negligible. The ground has some influence on the wind turbine, but for the current analysis it will be neglected.

3. Furthermore, it can be supposed that Mach number is zero, i.e., air have an infinite sound velocity, therefore is incompressible and doesn't undergo any temperature change.
4. Since the airflow is stable, all variables depend on space and not on the time when the turbine reaches its operational point.

If the purpose is to calculate the maximum energy, it should be considered that the flow transfer the same amount of energy from every point. Because of this the rotor is configured like a disc. Due to the fact that the axial velocity has to be equal and the mass should be conserved, the only way to extract energy is by a pressure difference. Thereby, this process can be defined by Bernoulli's equation in every current line where the total pressure is preserved.

$$p_t = p + \frac{1}{2} \cdot \rho \cdot v^2 \tag{2.1}$$

On the other hand, the amount of energy per mass unit (p/ρ) is not constant in the disc since an extraction is realized, where p is static pressure and $\frac{1}{2} \cdot \rho \cdot v^2$ dynamic pressure. Thereby, Bernoulli's equation can not be used through the disc.

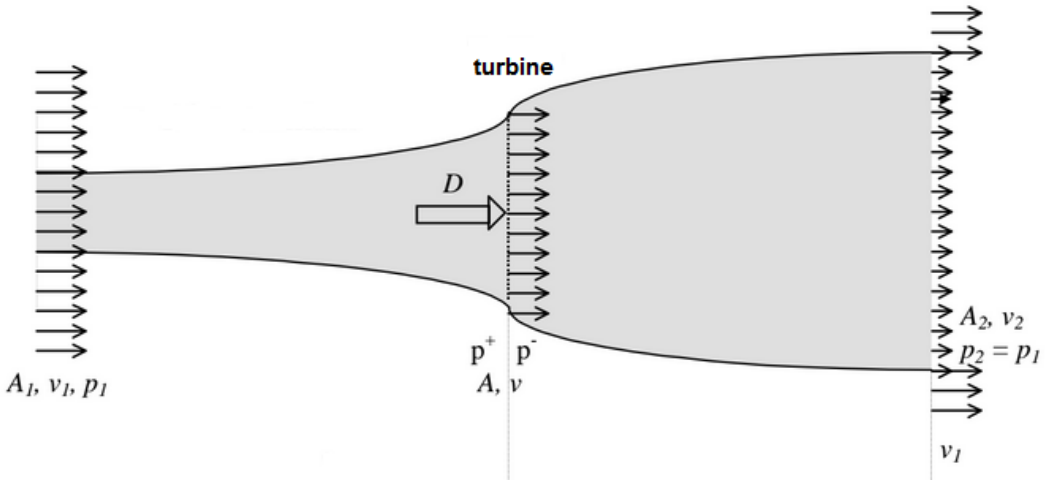


Figure 2.1: Betz's limit diagram.

Now conservation equations for the flow can be applied:

- **Continuity:** Mass flow has to be preserved among the flow.

$$\begin{aligned} \dot{m} &= \rho \cdot A_1 \cdot v_1 = \rho \cdot A_2 \cdot v_2 \\ v_1 \cdot A_1 &= v_2 \cdot A_2 \end{aligned} \tag{2.2}$$

- **Momentum:** The force of the disc over the flow with the current orientation -D, has to be equal to the difference of momentum in entry and exit flow.

$$D = -\dot{m}(v_2 - v_1) = \rho \cdot A \cdot v(v_1 - v_2) \tag{2.3}$$

- **Energy:** Agrees with the application of Bernoulli's equation between A_1 and A, and between A and A_2

$$p^+ + \frac{1}{2} \cdot \rho \cdot v^2 = p + \frac{1}{2} \cdot \rho \cdot v_1^2 \tag{2.4}$$

$$p^- + \frac{1}{2} \cdot \rho \cdot v^2 = p + \frac{1}{2} \cdot \rho \cdot v_2^2 \tag{2.5}$$

- **Equilibrium:** Since disc is stationary, the sum of forces about it must be zero.

$$D = (p^+ - p^-)A \tag{2.6}$$

Deducting from equation (2.4) and (2.5), the pressure difference through the disc is obtained:

$$p^+ - p^- = \frac{1}{2}\rho(v_1^2 - v_2^2) \tag{2.7}$$

And equalizing (2.3) and (2.6), the expression is obtained:

$$\rho \cdot A \cdot v(v_1 - v_2) = (p^+ - p^-)A \tag{2.8}$$

Applying (2.7) and simplifying:

$$\begin{aligned} \rho \cdot A \cdot v(v_1 - v_2) &= \frac{1}{2}\rho(v_1^2 - v_2^2)A \\ v &= \frac{v_1 - v_2}{2} \end{aligned} \tag{2.9}$$

As it has been said, the power of the turbine is extracted from the pressure differences, which is stated in the next equation.

$$P = (p^+ - p^-)A \cdot v = D \cdot v \tag{2.10}$$

Next, the last equation is developed (2.10) with (2.9) and (2.7).

$$\begin{aligned} P &= \frac{1}{2} \cdot \dot{m}(v_1^2 - v_2^2) \\ &= \frac{1}{2} \cdot \rho \cdot A \cdot v(v_1^2 - v_2^2) \\ &= \frac{1}{4} \cdot \rho \cdot A \cdot (v_1 - v_2)(v_1^2 - v_2^2) \\ &= \frac{1}{4} \cdot \rho \cdot A \cdot v_1^3 \cdot (1 - (\frac{v_2}{v_1})^2 + (\frac{v_2}{v_1}) - (\frac{v_2}{v_1})^3) \end{aligned} \tag{2.11}$$

If it is desired to obtain the power of the fluid P_{fluid} in (2.11) it has to be supposed that $\frac{v_2}{v_1}$ =1, and then would be it given as:

$$P_{fluid} = \frac{1}{2} \cdot \rho \cdot A \cdot v_1^3 \tag{2.12}$$

On the other hand, if the goal is to obtain the power which the turbine extracts, it can be renamed $\frac{1}{2} \cdot (1 - (\frac{v_2}{v_1})^2 + (\frac{v_2}{v_1}) - (\frac{v_2}{v_1})^3)$ by C_p, then the power obtainable from a cylinder of fluid with a cross sectional area A and velocity v_1 is :

$$P_{turbine} = C_p \cdot \frac{1}{2} \cdot \rho \cdot A \cdot v_1^3 \tag{2.13}$$

Where C_p means the power coefficient and represent the relation between the power of a fluid and the power that the turbine extract from this fluid ($C_p = P_{turbine}/P_{fluid}$). Thereby, if calculating the maximum C_p obtainable from a fluid, it would be given the maximum percentage of energy extractable from the fluid.

Furthermore, by differentiating $P_{turbine}$ with respect to $\frac{v_2}{v_1}$ for a given fluid speed v_1 and a given area A one finds the maximum or minimum value for P (Remember that C_p depends from v_1 and v_2). The result is that P reaches maximum value when $= \frac{v_2}{v_1} = \frac{1}{3}$. Substituting this value results in:

$$P_{turbine}^{max} = \frac{16}{27} \cdot \frac{1}{2} \cdot \rho \cdot A \cdot v_1^3. \tag{2.14}$$

In addition, the power coefficient has a maximum value of: $C_p^{max} = \frac{P_{turbine}^{max}}{P_{fluid}} = 16/27 = 0.593$ (or 59.3%);

To sum up, this means that a turbine can extract a maximum of 59.3% of the energy from the wind.[2]

2.2. Unsteady Aerodynamics Experiment Phase VI

The primary objective of the Unsteady Aerodynamics Experiment (UAE) [3] had been to provide information needed to quantify the full-scale, three-dimensional (3-D) aerodynamic behavior of horizontal-axis wind turbines (HAWT's). Since 1987, this experiment has been conducted by the National Renewable Energy Laboratory (NREL) at the National Wind Technology Center (NWTC) near Golden, Colorado.

During Phase VI experiments different combinations of flow speed, rotation speed, yaw, pitch and core angles and turbine orientations were used.

In this thesis simulations are performed particularly for "Ordinal Number S", which has the following properties.

- Orientation: Upwind
- Cone angle: 0.0 degrees

- Yaw angle: Locked
- Blade tip pitch: 3.0 degrees
- Rigid wind turbine
- Rotating turbine
- Rotation speed: 72 rpm

It has also been used in the article [4], which made a similar simulation of the *NREL Phase VI*, in order to compare our results with their Cp and torque data, and check that there aren't big differences. Additionally, other documents which address a similar issue as this thesis have been consulted, like [5], [6] and [7], although the results are not compared with them.

The purpose of our project is to validate simulation results with experimental values and create a model of the procedure which has been followed, in order to create a pattern if future research need an explanation of the process.

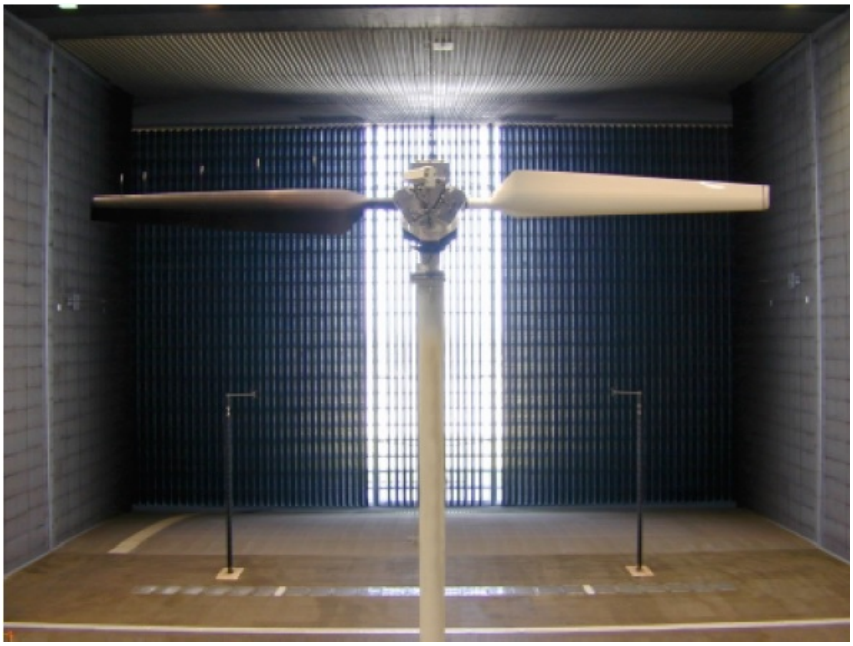


Figure 2.2: PHASE VI Turbine.

2.3. Finite Element Method

The finite element method (FEM) is a mathematical tool that finds approximate solutions to boundary value problems for partial differential equations. This method is commonly used in engineering and physical problems.

The FEM is intended for being used by computers and it allows solving differential equations associated with a physical problem in complicated geometries. FEM follows this procedure for solving problems:

- The continuous physical domain is divided into a finite number of parts, called elements, which behavior is specified by a finite number of parameters associated to certain characteristic points, called nodes. Nodes are joining points between different elements.
- The solution of the full system follows the rules of discrete problems. The full system is created by assembling of elements.
- The unknowns of the problem stop being mathematical functions and becomes the value of these functions in the nodes.
- The behavior in the inner part of every element is defined by the behavior of nodes by the functions of interpolation.[8]

The solver XNS uses a Newton-Raphson algorithm to realize the linearization, since equation systems in the simulations are non-linear. It also uses the Galerkin method in which the weight functions are equal to the interpolation functions.

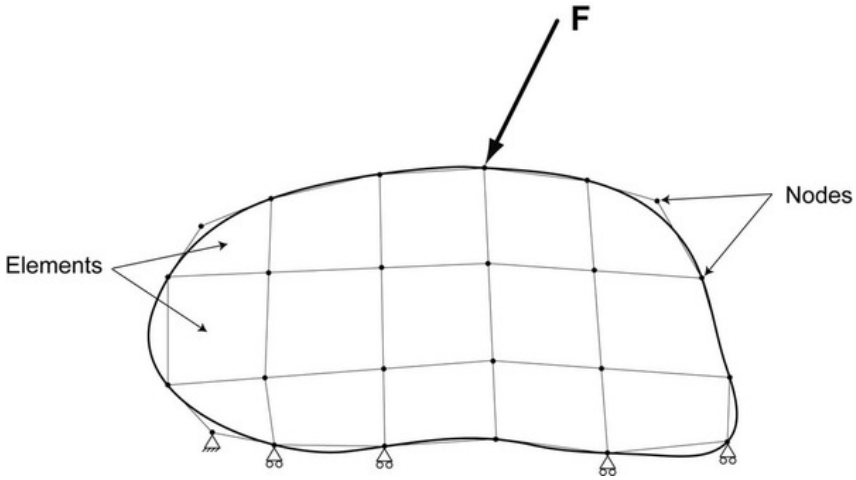


Figure 2.3: FEM diagram.

The governing equations, that is the incompressible Navier-Stokes equations, can be written in the following form:

$$\begin{aligned} \rho \cdot (\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - f) - \nabla \cdot \sigma = \mathbf{0} \; on \; \Omega_t, \forall t \in (0,T) \\ \nabla \cdot \mathbf{u} = \mathbf{0} \; on \; \Omega_t, \forall t \in (0,T) \end{aligned} \tag{2.15}$$

The space-time method is used for the discretization of the problem . In order to construct the finite element function spaces for the space-time method, the time interval is divided into subintervals $I_n = (t_n, t_{n+1})$, with t_n and t_{n+1} representing an ordered series of time $0 = t_0 < t_1 < \cdots < t_N = T$. Now, if $\Omega_n = \Omega_{t_n}$ and $\Gamma_n = \Gamma_{t_n}$, the space-time slab Q_n is defined as the domain enclosed by the surfaces Ω_n, Ω_{n+1} as well as the surface described by Γ_t as t transverses I_n, which shall be named P_n. (See Fig. 2.4). The following

finite element interpolation and weighting function spaces are defined for the velocity and the pressure, for each space-time slab:

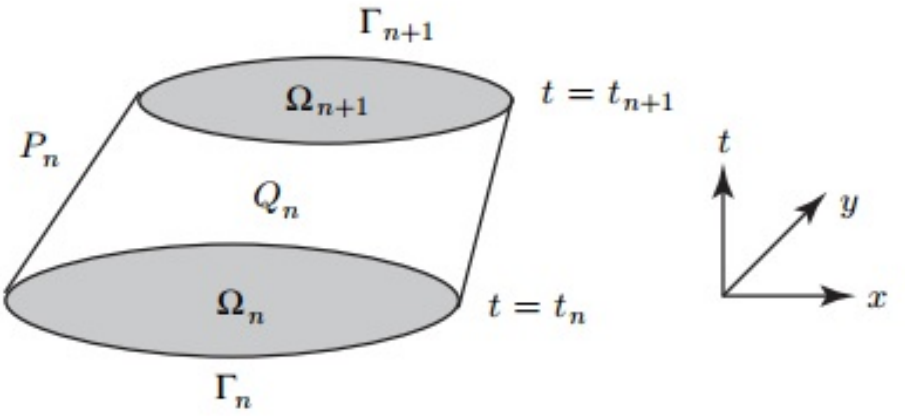


Figure 2.4: Sketch of a space-time slab.

$$
\begin{aligned}
(\mathbf{S_u}^h)_n &= \{\mathbf{u}^h | \mathbf{u}^h \in [H^{1h}(Q_n)]^{n_{sd}}, \mathbf{u}^h \doteq \mathbf{g}^h \ \ on \ \ (P_n)_g\} \\
(\mathbf{V_u}^h)_n &= \{\mathbf{w}^h | \mathbf{w}^h \in [H^{1h}(Q_n)]^{n_{sd}}, \mathbf{w}^h \doteq \mathbf{0} \ \ on \ \ (P_n)_g\} \\
(\mathbf{S_p}^h)_n &= (\mathbf{V_p}^h)_n = \{p^h | p^h \in H^{1h}(Q_n)\}
\end{aligned}
\tag{2.16}
$$

The interpolation functions used over the element domains are first-order polynomials in space as well as in time. They are continuous in space, but discontinuous in time.

The stabilized space-time formulation of the incompressible Navier-Stokes equations can be expressed as follows: given $(\mathbf{u}^h)_n^-$ find $\mathbf{u}^h \in (\mathbf{S}_u^h)_n$ such that $\forall \mathbf{w}^h \in (\mathbf{V}_w^h)_n, \forall q^h \in (\mathbf{V}_p^h)_n$:

$$
\begin{aligned}
&\int_{Q_n} \mathbf{w}^h \cdot \rho(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h - \mathbf{f}) dQ + \int_{Q_n} \epsilon(\mathbf{w}^h) : \sigma(p^h, \mathbf{u}^h) dQ \\
&+ \int_{Q_n} q^h \nabla \cdot \mathbf{u}^h dQ + \int_{\Omega_n} (\mathbf{w}^h)_n^+ \cdot \rho((\mathbf{u}^h)_n^+ - (\mathbf{u}^h)_n^-) d\Omega \\
&+ \sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \tau_{MOM} \frac{1}{\rho} [\rho(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{w}^h) - \nabla \cdot \sigma(q^h, \mathbf{w}^h)] \\
&\cdot [\rho(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h - \mathbf{f}) - \nabla \cdot \sigma(p^h, \mathbf{u}^h)] dQ + \sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \tau_{CONT} \nabla \cdot \mathbf{w}^h \rho \nabla \cdot \mathbf{u}^h dQ \\
&= \int_{(P_n)_h} \mathbf{w}^h \cdot \mathbf{h}^h dP
\end{aligned}
\tag{2.17}
$$

The following notations are applied:

$$
\begin{aligned}
(\mathbf{u}^h)_n^{\pm} &= \lim_{\epsilon \to 0} \mathbf{u}(t_n \pm \epsilon) \\
\int_{Q_n} \dots dQ &= \int_{I_n} \int_{\Omega_t^h} \dots d\Omega dt \\
\int_{P_n} \dots dP &= \int_{I_n} \int_{\Gamma_t^h} \dots d\Gamma dt
\end{aligned}
\tag{2.18}
$$

The problem is solved sequentially for each space-time slab, starting with:

$$
(\mathbf{u}^h)_0^{+} = \mathbf{u}_0 \tag{2.19}
$$

The variational formulation is made following this procedure: The first three terms on the left hand side, together with the right hand side, form the standard Galerkin formulation, whereas the fourth term weakly enforces the continuity of the velocity across space-time slabs. The fifth and sixth term constitute the least-squares stabilization. These stabilization terms allow the use of equal-order interpolation functions for velocity and pressure, which would otherwise be unstable, and guard against spurious velocity oscillations in advection-dominated flows. [9]

2.4. Shear-Slip Mesh Update Method Explanation

Different strategies can be used to allow the simulation of meshes which have rotating and stationary parts; the Shear-Slip Mesh Update Method (SSMUM), is a commonly used method for this task. It takes advantage of the relaxed continuity requirements at the interface between space-time slabs (time steps). Stretched elements are produced in simulations where the mesh has to deal with large deformations. Due to the rotation of the rotor, this problem appears in the mesh which is being studied, and it necessitates remeshing. Remeshing can be computationally costly, if an automatic mesh generator needs to be called, and it requires a projection of the solution from the old mesh to the new one. These projections introduce additional inaccuracies to the computation, which, depending on the frequency and scope of remeshing, could be excessive.

The SSMUM uses rigid unstructured meshes attached to individual boundaries of the domain, and connect them together by introducing a layer of elements. The objective is to restrict remeshing to a small layer. In our case of study, the layer is a one-element thick disk enclosing the rotor, connecting two rigid meshes; one outer region with their outer channel boundaries, and one attached and rotating with the rotor. With this approach, the cost of remeshing becomes negligible, and consequently, such remeshing can be carried out frequently, even at every time step. [10] [11]

The update layer has three different parameters:

- The node distribution along the layer.

- The number of divisions in rotation direction.
- The thickness of the update layer.

The Shear-Slip Mesh Update Method is applicable for flow simulations involving large, but regular displacements of one or more boundaries of the computational domain.

3. Simulation

3.1. Design of Geometry Using Rhinoceros

3.1.1. Blade Design

First the defined points in the document [3] are copied in any text software and exported to Rhinoceros to determine the blade surface, in this case is the S809, with a length of 1 meter. In order to avoid the creation of elements with severely non-orthogonal faces; the trailing edge of the S809 surface has been cut out, as a result the blade length is reduced by 2% as recommended in [12].

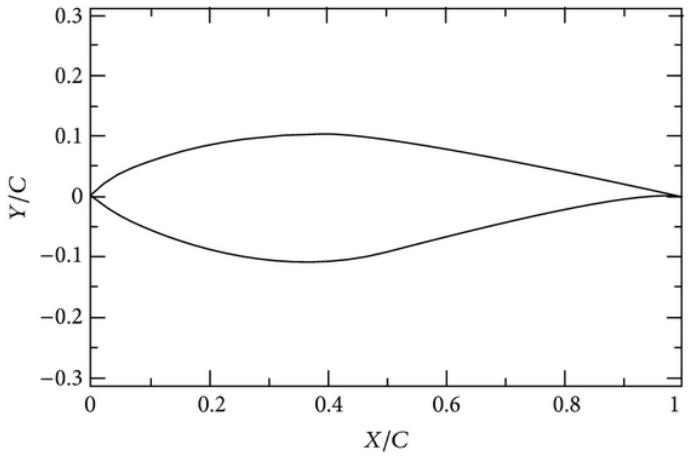


Figure 3.1: S809 airfoil profile.

The full table with all S809 surface point coordinates can be found in A.1. The full table with all the blade cross section geometry data can be found in A.2.

The next step is the creation of three curves through these predefined points: one for the top side, another for the bottom side, and a last one for the cut trailing edge surface. Then, these curves are copied with different scales and rotated according to the specifications given in reference document [3].

Afterwards, the rotor surfaces are created along the curves with the option *Loft* and *normal* style. Eventually, both, top and bottom of the blade are closed with two surfaces

with the option *Loft*. Since ordinal number S is used in this simulation 3 degrees of pitch angle is applied to the rotor blade.

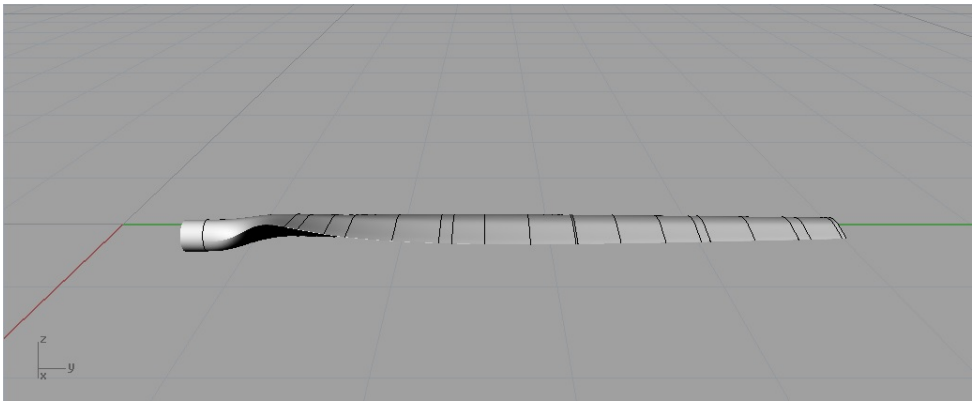


Figure 3.2: Blade.

3.1.2. Rotor and Tower Design

For the rotor a cylinder has been created, which is also defined on the document [3], and it is joined to the blade with the option *Blend Surface* and an adjustment to the curvature of the curves in the union is made.

Finally, there is a join part between both blades which have been created by our own criteria but according to the pictures of the experiment [3], since this part is not described in the original experiment; this way we can make a tighter comparison with this document.

For the tower the original height and the position regarding the rotor documented on [3] was respected, but the remaining data has been also created according to the pictures of the experiment by our own criteria.

All the documents were exported on *.igs* format.

3.2. Mesh Generation

3.2.1. Mesh Generation by Pointwise

The first step is to define tolerances in Pointwise properties menu, these are the ones used in this thesis:

- Model size: 1.0
- Node: $1.0 \cdot 10^{-6}$
- Connector: $1.0 \cdot 10^{-6}$
- Grid Point: $1.0 \cdot 10^{-7}$

Afterwards the design file is imported in *.stp* format, then, some unnecessary connectors are deleted in order to make the domain creation easier. 3 domains are needed for each blade, one per surface. For the creation of the largest connector, the one which traces the blade length, all the connectors generated by the option *Connectors on Database Entities* were joined. In addition, for this new connector 100 points were considered; for the connectors corresponding to the S809 surface were placed a different number of nodes, since as is shown in results the number of nodes in this region has a big influence on results. Finally, it has been considered that 80 points is an optimal number for these connectors. For the connectors belonging to the sharp surface there were only two points. All the domains on the blade were generated using structured elements, except the ones on the top and bottom surface.

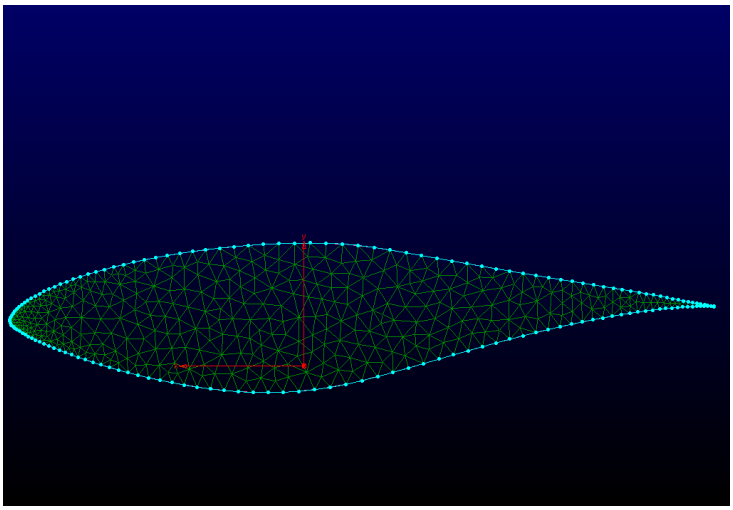


Figure 3.3: Blade Tip Distribution.

For the remaining parts the domains are generated following the same procedure and keeping structured elements. Afterwards the grid is processed in order to obtain a smoother mesh around the surfaces. Then all structural domains are triangulated. Once one blade is completely defined, the complete rotor has to be created. For this task the blade is copied and turned 180 grades. Afterwards, the central join part is imported, and its domains are created. It has to be checked that connectors and domains are correctly connected each other, and do not exist duplicate connectors. For this task, the option *merge* on Pointwise *grid* menu shows all connectors topology.

For the blade, the option TRex has been used. This is an advanced feature that reduces boundary layer mesh generation time while producing high quality grids that give accurate CFD solutions. This feature extrudes regular layers of tetrahedra from boundaries.

Moreover, it is necessary to create two regions, one of which is rotating and called inner region, and the other one is called outer region. In the version without tower the inner region is created as a cylinder with 6.5 m radius and 6 m length, the rotor is situated

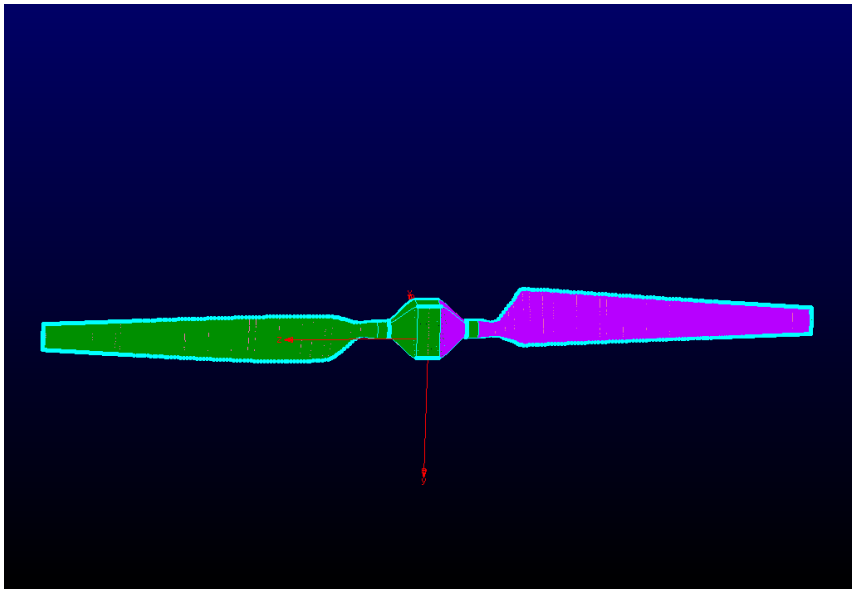
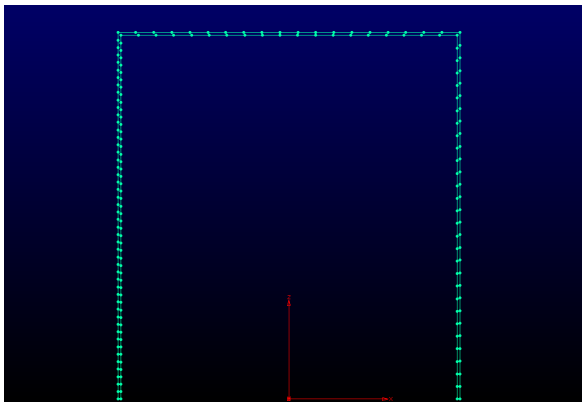


Figure 3.4: Complete Rotor.

such that there is 5 m space behind this region. The space between both block layers is always 0.05m.



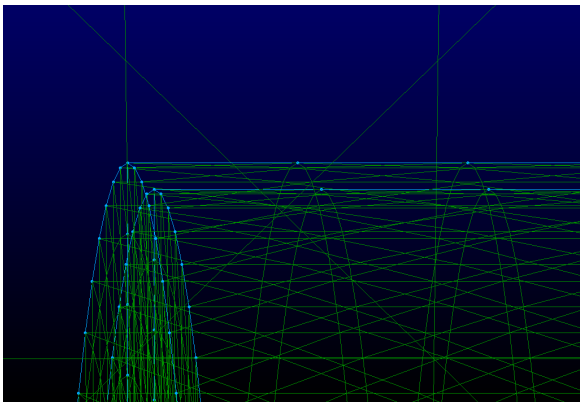


Figure 3.5: Inner and Outer Layers.

For the outer region a 40 m diameter cylinder is generated with 30 m of length. In the intersections between outer and inner regions two segments are created since the system used for solving the simulation is the SSMUM method, which has been explained in 2.4 subsection.

For the version with tower a nacelle was created also, which was designed on Pointwise as a cylinder of 0.65 m diameter, 2.75 m length, leaving a 0.3 m gap between nacelle and outer layer and 0.8 m gap between nacelle and the tower centre. The rotor of this version is imported from the other version, and the tower, which was created on Rhinoceros, is also imported. The domains are created following the same procedure

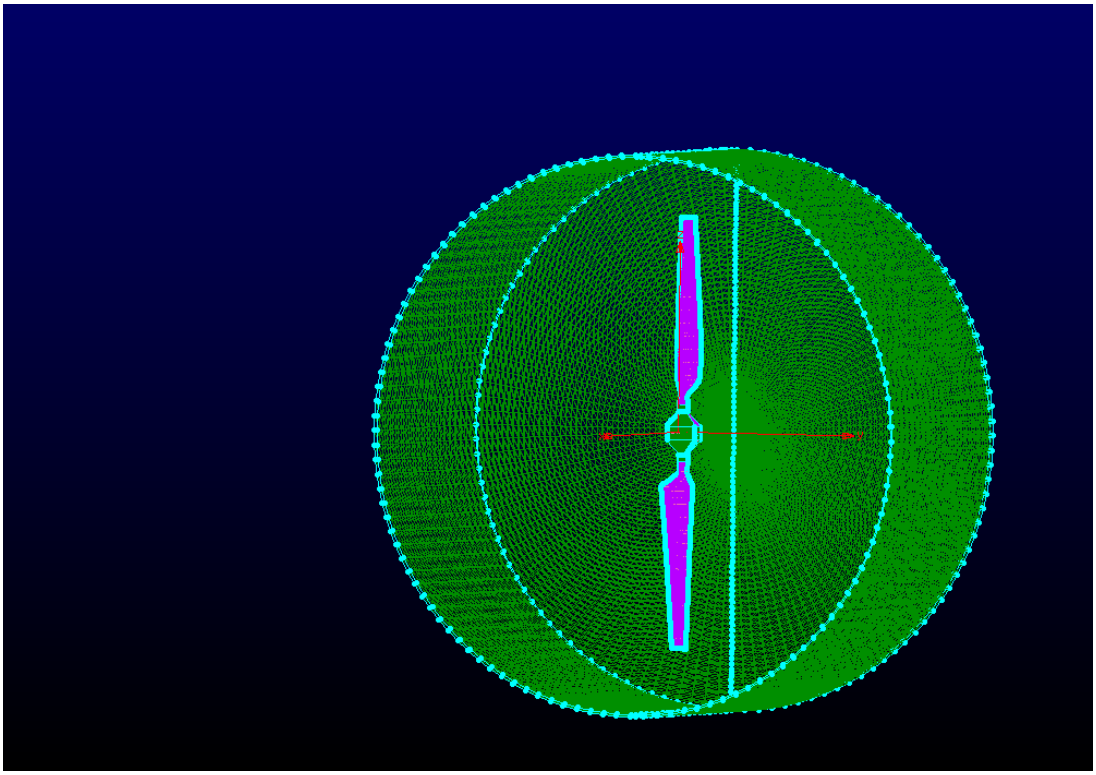


Figure 3.6: Inner region.

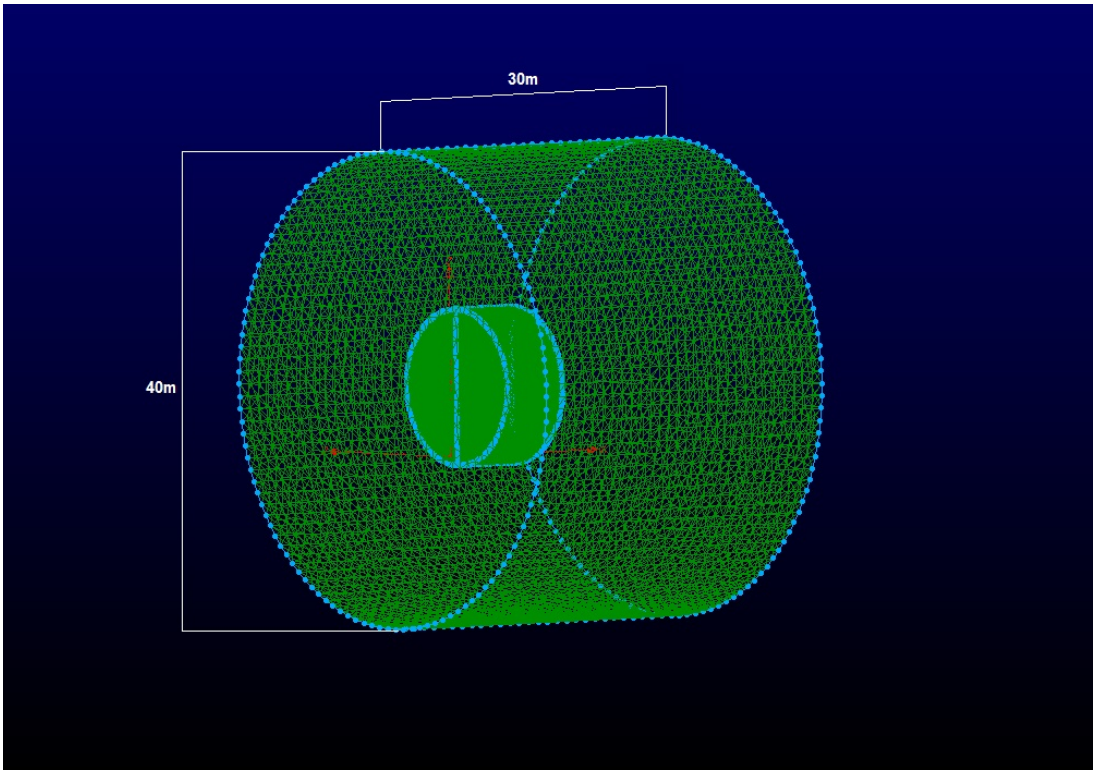


Figure 3.7: Outer region.

as for the rotor. In this case the inner region is a cylinder of 1.5 m length and 6.5 m radius. The outer region is a rectangular prism of 36 x 30 x 26.85 m.

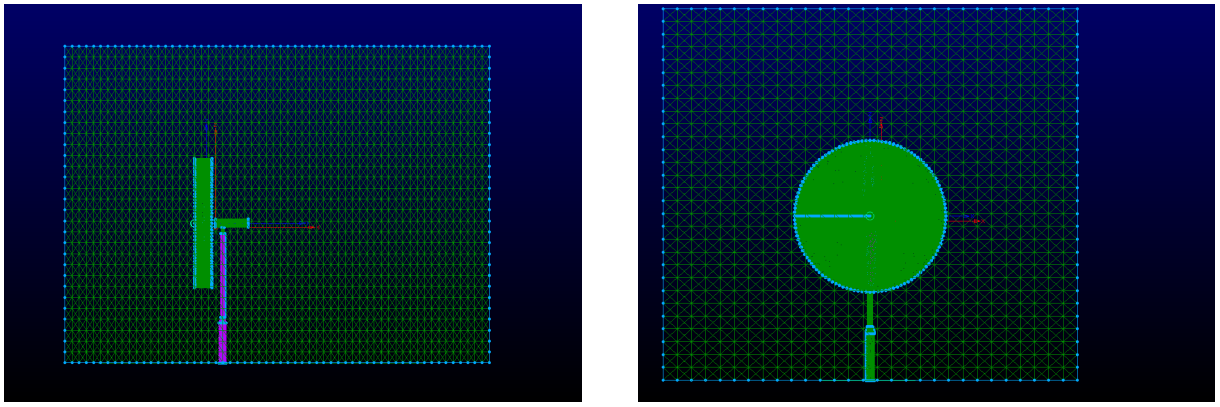


Figure 3.8: Tower Version.

Eventually, it is needed to create two blocks, one for the inner region and another for the outer region. Before exporting blocks the boundary conditions has to be fixed, where the following ordering was used.

- Blade domains were defined as *Wall* type.
- The domain where the flow enters was determined as *Inflow* type.
- The domain where the flow exits was determined as *Outflow* type.
- Lateral domains of the outer cylinder are determined as *Symmetry Plane* type
- Remaining domains are the limit of the SSMUM layer between inner and outer region, which were set in separated *Boundary conditions* but their type is *Unspecified*.

The most important part of setting boundary conditions is the name which is associated to every *Set*, which always have the format *BndCond #*. In addition, this name has to agree with their relative boundary condition in the RWTH Computer Cluster and Juqueen files for decomposing and solving the mesh. This procedure will be explained in more detail in further sections.

Last but not least, all the files for next steps were prepared in the following order:

1. The *Outer Block* is exported from Pointwise in format *.uns*, and named as **outer.uns**.
2. The *Inner Block* is exported from Pointwise mesh in format *.uns*, and named as **inner.uns**.
3. The *Inner Segment* is exported from a Pointwise mesh in format *.dat*. For this file first all the segments which mark out he area of SSMUM's layer have to be copied, inner and outer, in another Pointwise file, where the segments have to be joined, thereby must be only two segments, one for inner region and another for outer region. This file is named as **innersegment.dat**.

4. The *Outer Segment* is exported from a Pointwise mesh in format *.dat*, which is created equally as *Inner Segment*, and named as **outersegment.dat**.

3.2.2. Mesh Generation on the Cluster

Once all files are prepared they must be copied to Cluster. This task can be completed using command *scp*, where some tools will be imported for creating the mesh. First the modules MISC and VTK have to be imported, then it is necessary to check *settings.in* file in order to assure that the number of divisions use in the creation of the segments and the boundary conditions agree with this file. Then the file *create_ssmum_mixd_level1.sh* is executed. This step creates the SSMUM mesh and inner and outer regions in *mixd* format.

After this procedure the files *mergeblocks1.in* and *mergeblocks2.in* must be checked because they have to agree the numbering and the number of surfaces in these files with the ones in Pointwise's files. Then the file *create_ssmum_mixd_level2.sh* is executed. Now the mesh is completed. This step merges inner, outer mesh and SSMUM mesh.

Afterwards it is strongly recommended to check the final mesh with the tool *Paraview*, and if everything is correct, all the files have to be copied to Jülich Supercomputer, Juqueen, where it will be necessary to use some other tools for decomposing the mesh into partitions.

3.2.3. Mesh Partitioning on the Juqueen

Once we have all the files copied to Juquen, the next step is to decompose the mesh, so that, the mesh can be solved in parallel using more cores simultaneously, saving time. For this task it is needed to create files *dual* and *neim*. The file *neim* will determine the maximum number of elements touching a node, and in one node, this number have to agree with the *nne* value in the file *decompose.in*. Another important option in file *decompose.in* is the number of desired partitions *nparts*. Afterwards the file *run.juqueen* has to be set. The most important features of this file are explained below:

- **wall_clock_limit**: This option sets the maximum time the submitted job will be running in Juqueen. It is important since the higher the *wall_clock_limit* is, the more time Juqueen will delay its start. Hence, is essential to reach an efficient point between this limit and the time needed by the job to be solved, so that the needed time never exceeds the *wall_clock_limit*.
- **notification**: The user can choose between receiving a notification when the job starts and finish running on Juqueen or not, respectively *always* or *never* option.
- **notification_user**: Here the email where notifications will be sent is written.

- **bg_size**: In this section user can choose the number of cores to be used in the simulation.

3.3. Running Simulations on Juqueen

Finally, the mesh is prepared to be solved. First it is necessary to simulate steady conditions, and afterwards unsteady conditions. Due to the large size of the mesh, unsteady flow can not be solved in one simulation run. Therefore, it is solved partially every time it is sent to Juqueen. For this we will use the file *xns.in*, in which the following options have special importance:

- **restart on**: In this part directories for *data*, *mien* and *mxyz* files are given, which the simulation will take for solving, hence, they must be the ones of the last partial step that has been solved. In the beginning, for solving the mesh in steady conditions, these files are placed in *mesh* folder, but for starting the unsteady simulations, the directory of file *data* changes, this file is now placed on folder *steady* and have the name *data.out.steady*. In addition, to carry on the unsteady simulation, the file's directory must be changed, they are now in *unsteady* folder and have the name respectively *data.out*, *mien.out* and *mxyz.out*. Now the user have to replace these files in another folder, since if *run.juqueen* is run again, the new files will replace the older ones and these old files will be lost.
- **nts**: Number of time steps: This attribute determines the number of time steps that will be done in every partial simulation.
- **dt**: Time step size: This attribute sets the time that each time step will simulate. Therefore, the product between *nts* and *dt* is the time which the partial simulation is run.
- **material**: In this section the *density* and *viscosity* of the fluid must be determined in SI units.
- **OtherBCs**: This section will set the three velocity components of the flow in the inflow region. Therefore, it has to agree the boundary condition number of Pointwise's file with the one given in this section.
- **rngdrot**: This attribute can be found in *mutating on* section and determines the rotation velocity of the rotor.

In this thesis, every case has simulated 3 seconds, using approximately 12 hours for every partial unsteady simulation that correspond in average 0.22 seconds of the simulation(14 partial unsteady simulations per job), the average *number of time steps* used has been 1100 and the *time step size* 0.0002. Hence, every job takes about 168 hours (7 complete days) of computer solving; in real time this time increases due to either ev-

ery new partial simulation order is not sent immediately the last one have finished, or Juqueen doesn't start the job straightaway the new order is sent.

Also, it is necessary to take a look in the file *run.juqueen*, since it will probably be necessary to changed the limited time.

For the computation of the C_p coefficient it is necessary to run a final step using the option *ntsbrng 1*. This way vtk files can be obtained using the tool *mixd2vtk*. Hence, using software *Paraview* will be necessary. More information about Juqueen can be found on the webpage [13].

Other useful commands for solving step can be:

- **llq -u 'username' :** It is used to see the current state of jobs which have been sent to Juqueen. User name is the one a person use to connect to Juqueen
- **llcancel 'jobID' :** It is used to cancel a job which is now in Juqueen. JobID can be found by using llq -u 'username' command.

4. Results

4.1. Mesh Independence

Initially three simulations of the rotor has been generated, in order to determine how many elements in the S809 connector curves of the blade are necessary for obtaining good enough results. The variation of the number of these connector is essential due to they have got a high impact in the results. Therefore three different versions have been considered necessary to determine the optimal number of nodes, which allows enough precision in following simulations and also don't suppose a time solution problem.

Below is a table with the characteristics of these three versions.

	Nodes in S809 surface	TRex's Max Layers	TRex's Δ s	Total Cells	Total Nodes
Rotor Mesh 1	80	0	0	3.810.365	658.926
Rotor Mesh 2	120	15	0,0001	5.091.864	878.279
Rotor Mesh 3	160	15	0,0001	6.217.081	1.070.111

- **Note:** *Nodes in S809 Surface* make reference to the nodes that have been drawn for every connector among the blade to the S809 surface, i.e. for 80 nodes in S809 surface there are 40 nodes in the connectors in the flow direction face normal direction and another 40 nodes in their opposite face connector.

Figure 4.1 shows a comparison of the Torque between these three mesh and PHASE VI results, which are represented by a gray rectangle, for 5 m/s velocity.

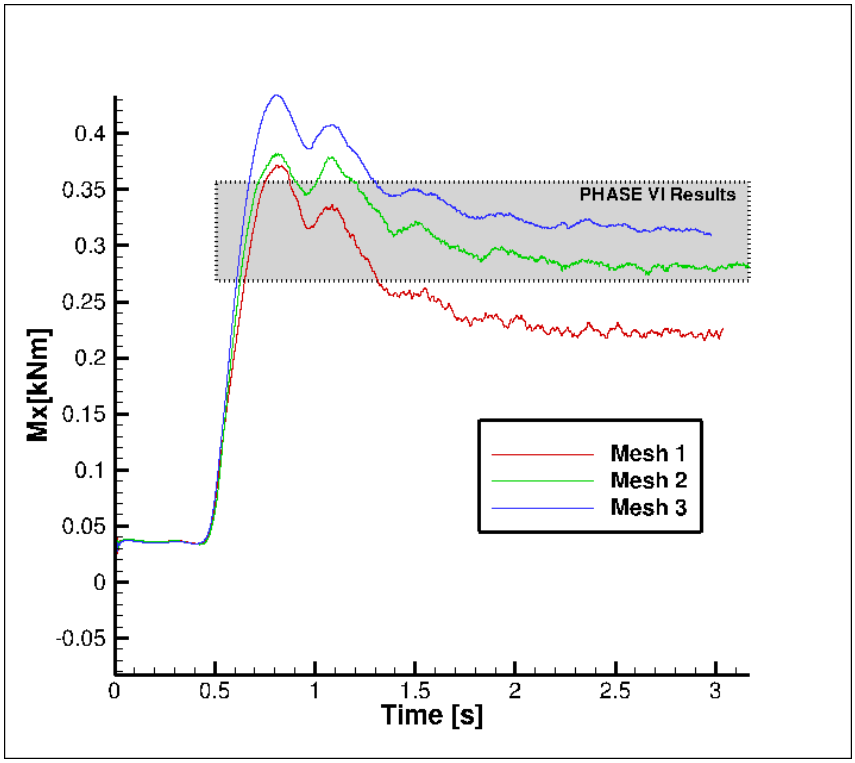


Figure 4.1: Torque Comparison.

As can be seen, only meshes 2 and 3 are inside the range of the Phase VI Results data. For next simulations Mesh 3 has been chosen, since their results are closer to Phase VI average and their bigger number of cells make it more secure, although the computational work will be also increased. Therefore, it is assumed that solution is independent of mesh for mesh 3 for 5 m/s velocity.

4.2. Cp Distribution

For the computation of the Cp curves a few more steps are needed. First the simulation has to be run more time steps, in order to generate the *.rng* files which are used to create *.vtk* files with the *CATS* tool *mixd2vtk*.

Afterwards, the data are treated in *Paraview*, where the pressure data will be analyzed. It is important to emphasize these points:

- Due to turbulence model, it has been necessary to take the average value of each time step for creating the data.
- Since the rotor is rotating, it had to be aligned before the time averaging, this has been done with an in-house filter.

In figures 4.2, 4.3 and 4.4 results of C_p distribution are given for different locations (r/R=30%, 46.6%, 63.3%, 80% and 95%) and different flow speed (u= 5 m/s, 7 m/s and 10 m/s).

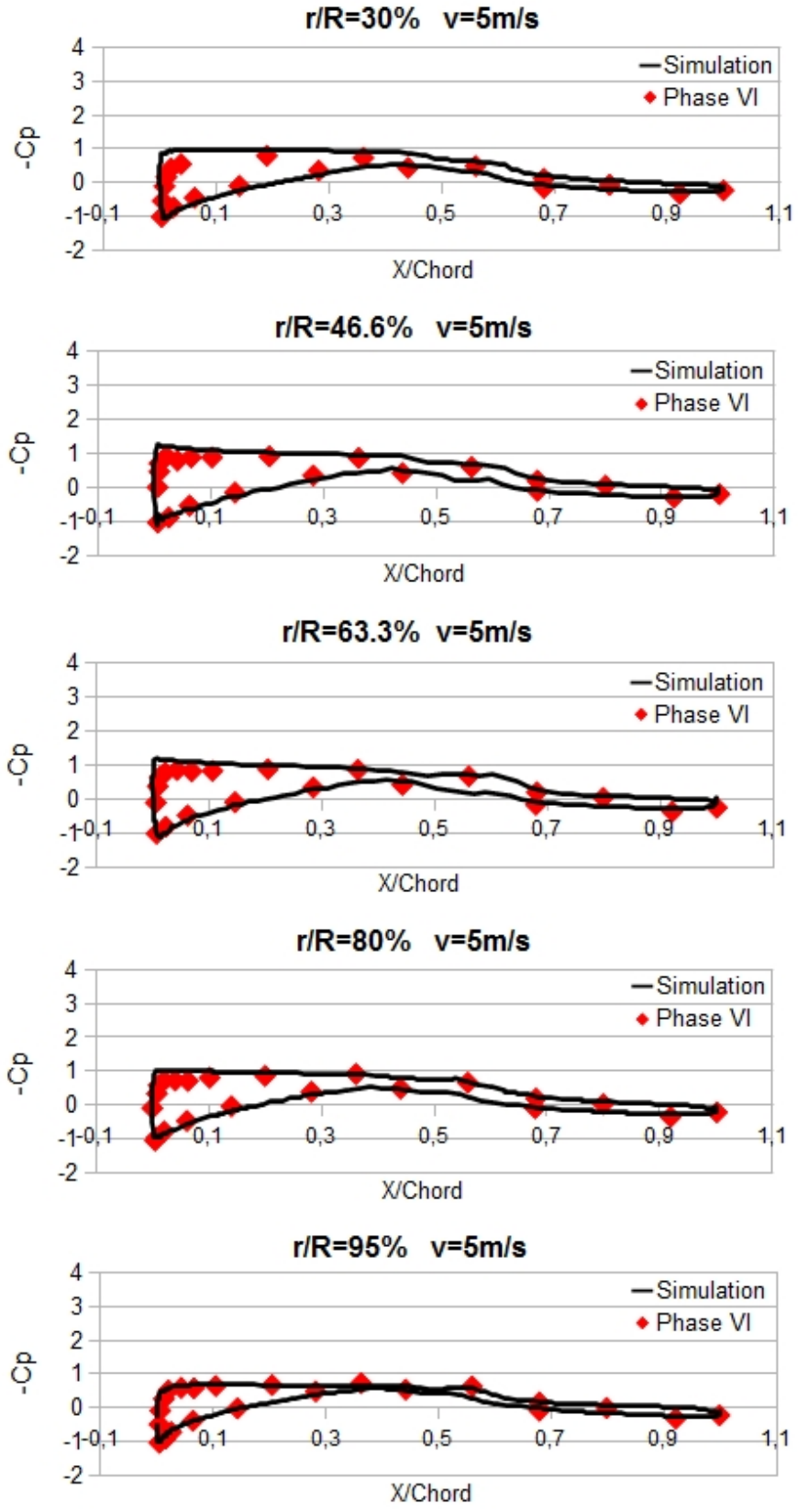


Figure 4.2: Cp Curves for 5m/s.

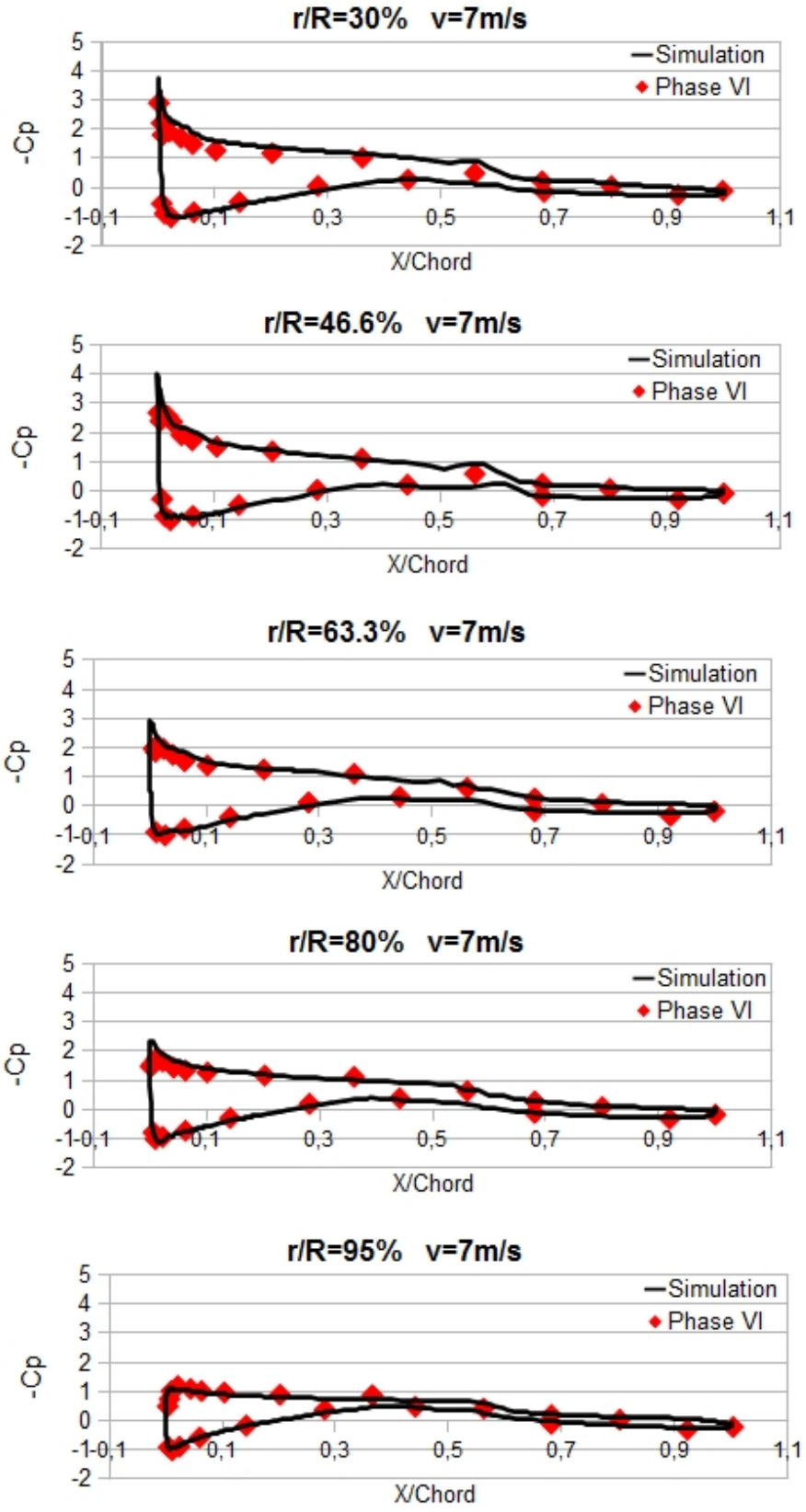


Figure 4.3: Cp Curves for 7m/s.

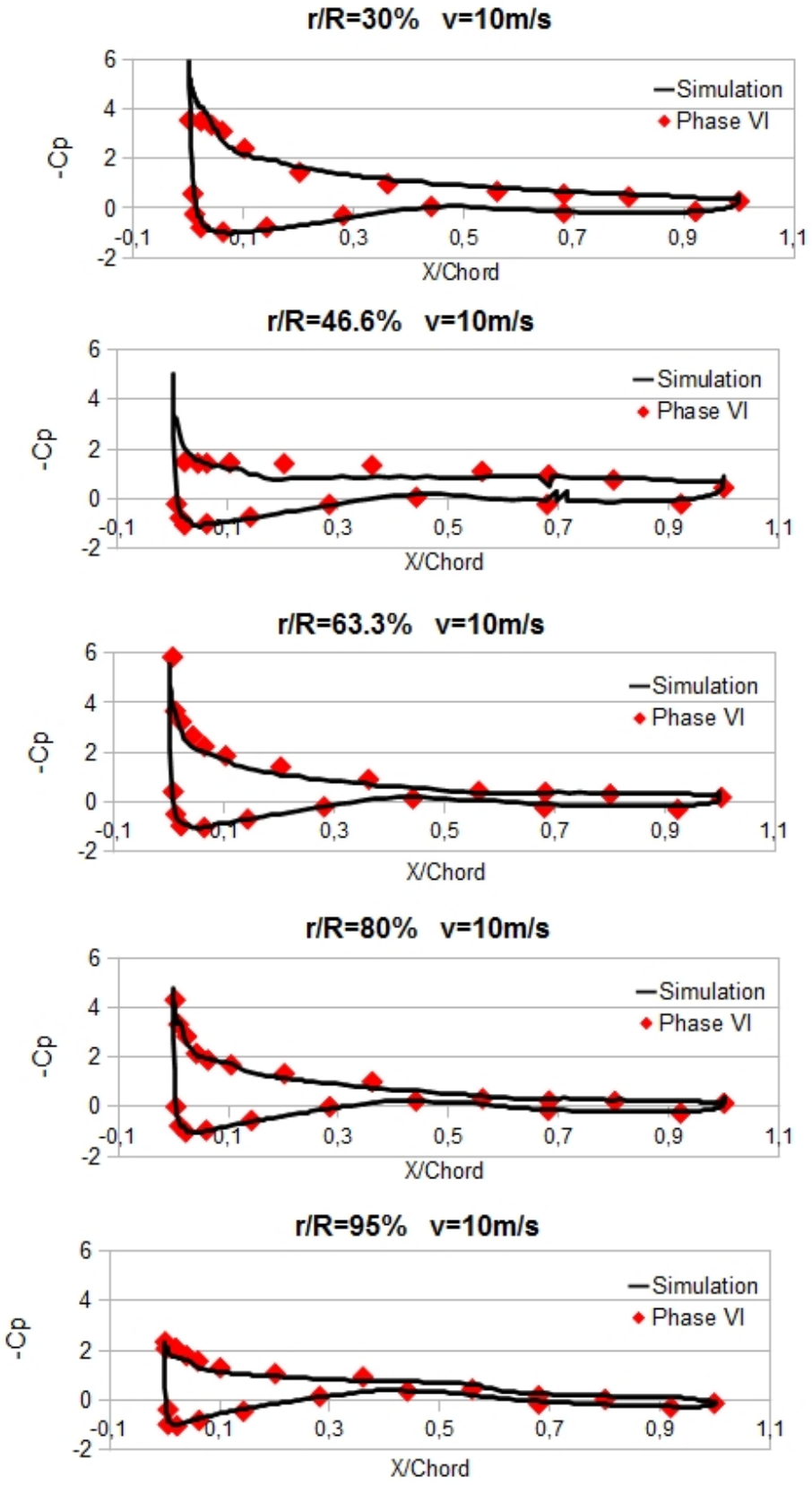


Figure 4.4: Cp Curves for 10m/s.

As it can be seen in C_p curves, results show a good agreement with the experimental data.

4.3. Torque Results

Moment data are exported directly from the simulation result data to any data processing software.

For analyzing moment data, the average of the last 1000 time steps of each simulation velocity has been taken, in order to create an accurate average. In the case of *PHASE VI* data, the showed value for the torque is an average between the maximum and the minimum accuracy limits of the experiment.

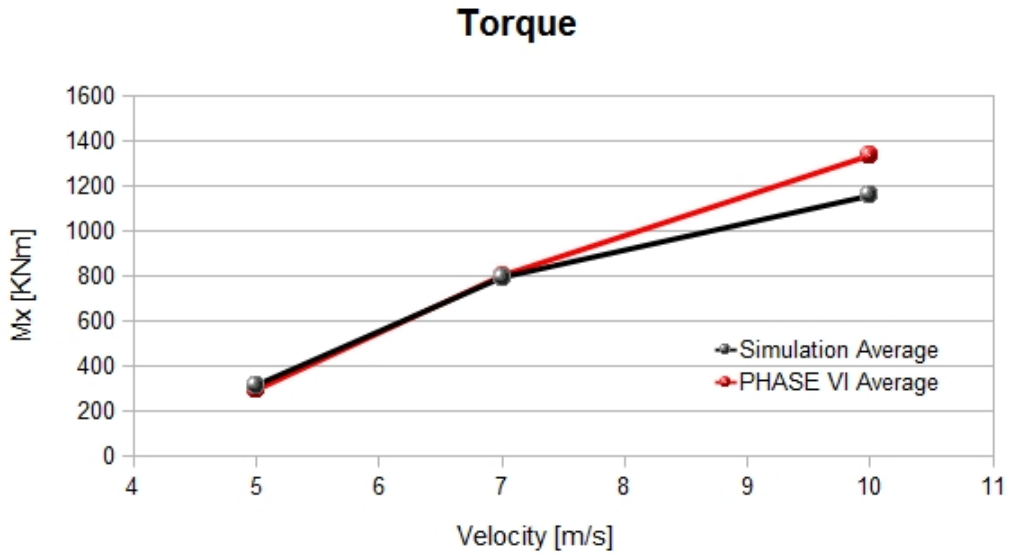


Figure 4.5: Moment Results.

As can be seen in the Figure 4.5, obtained values for 5 m/s and 7 m/s velocities are closer to the expected values. However, the obtained value for 10 m/s velocity is lower than expected. This could be due to the mesh is not fine enough. Probably, if the process which have been done for choosing the mesh in 5 m/s, would have been done for 10 m/s, it could be easily seen that the mesh were not fine enough and it would be optimized. But also for this procedure more time would have been invested, since the running process for 10m/s is slower than for 5 m/s.

4.4. Pressure Contours

Figures 4.6, 4.7 and 4.8 show the pressure contours at 80% spanwise location for 5, 7 and 10 m/s velocities, respectively; exact moments of the capture are 3.4 s, 3 s and 2.9 s, respectively.

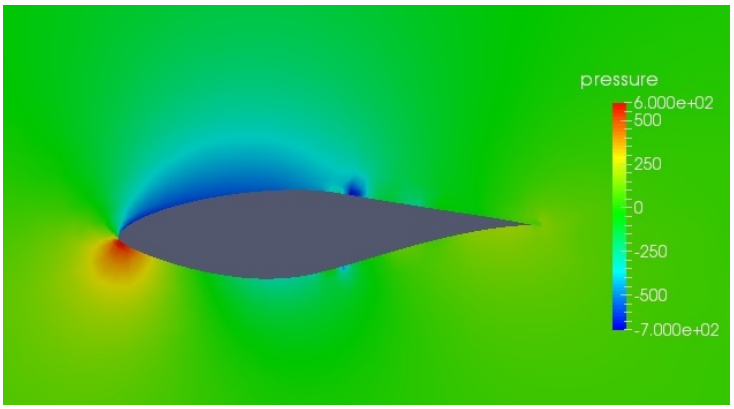


Figure 4.6: Pressure contour at 80% spanwise location for 5 m/s.

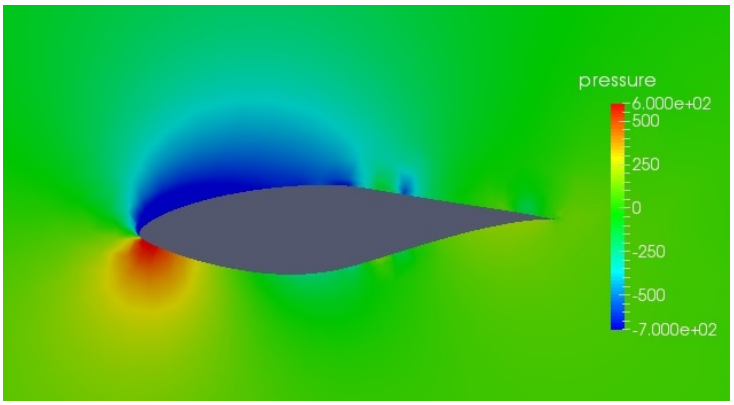


Figure 4.7: Pressure contour at 80% spanwise location for 7 m/s.

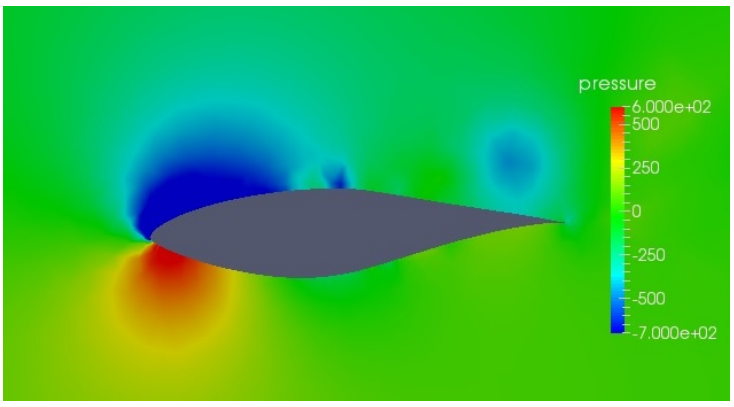


Figure 4.8: Pressure contour at 80% spanwise location for 10 m/s.

4.5. Velocity Contours

Figures 4.9, 4.10 and 4.11 show the velocity contours at 80% spanwise location for 5, 7 and 10 velocities respectively; exact moments of the capture are 3.4 s, 3 s and 2.9 s respectively.

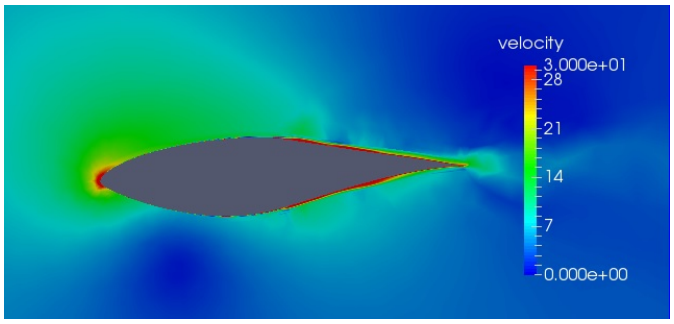


Figure 4.9: Velocity contour at 80% spanwise location for 5 m/s.

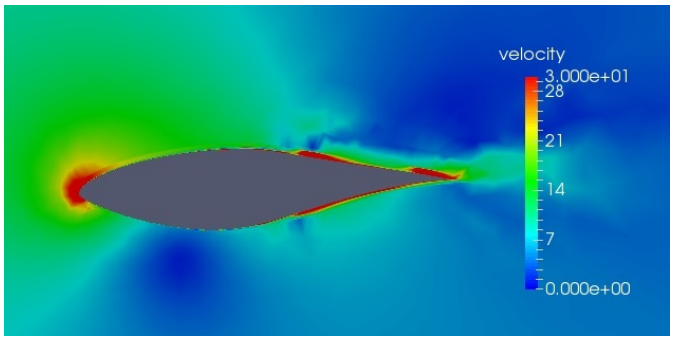


Figure 4.10: Velocity contour at 80% spanwise location for 7 m/s.

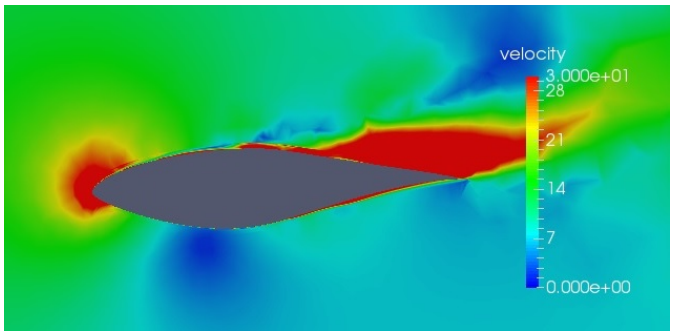


Figure 4.11: Velocity contour at 80% spanwise location for 10 m/s.

4.6. Tower Results

Figures 4.12 and 4.13 shown contours for pressure and velocity respectively for 7m/s velocity.

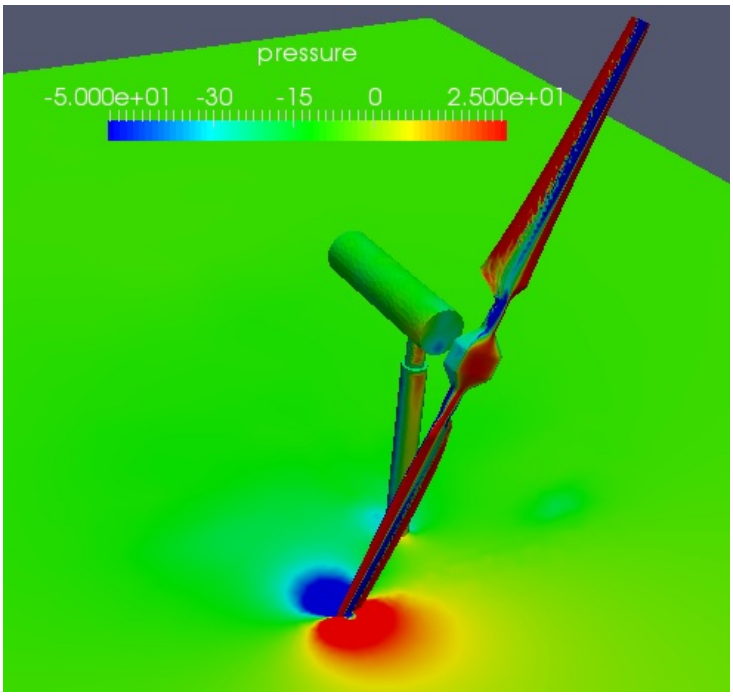


Figure 4.12: Velocity contour at 80% spanwise location for 7 m/s.

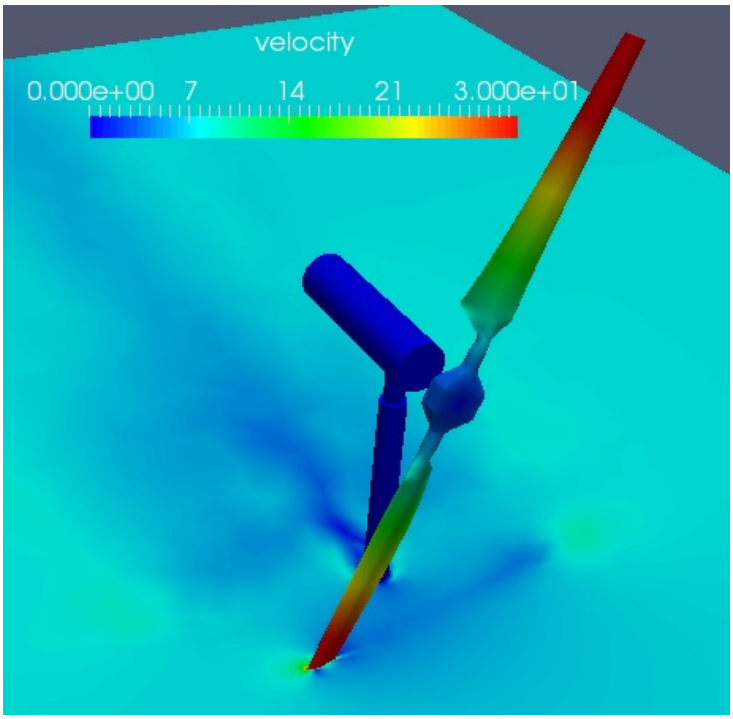


Figure 4.13: Velocity contour at 80% spanwise location for 7 m/s.

5. Summary

In this bachelor thesis the procedure for creating a wind turbine simulation has been explained in detail. The results show that the simulation is valid, although it is possible to do a further study and get improved results.

5.1. Suggestion for Further Investigations

As has been said in the torque results, the resulting moment for 10 m/s velocity is lower than should be. Therefore, it would be necessary to refine the mesh in order to achieve complete mesh independence; basically the same process that has been used for 5 m/s.

In addition, the result can be used to obtain other data, that have not been considered important for this thesis, like the root flap bending moment or other coefficients, such as torque coefficient. This kind of additional data is obtained in other studies such as [4]. More comparison with this and simulation results that are obtained by other researches could be done.

Moreover, the tower mesh used in this thesis have not been deeply used due to lack of time, but it is known that the tower has a meaningful impact on the results, since in all experiments it has been used. Therefore, it is recommendable to use this tower for further investigations. Although the tower increase the computational workload, results can be closer to reality.

A. Annex

A.1. Annex I: S809 surface point coordinates

Below is the table with all coordinate points from S809 surface, procured from [3].

Upper Surface		**Lower Surface**	
x/c	**y/c**	**x/c**	**y/c**
0.00037	0.00275	0.00140	-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
0.10390	0.06082	0.12397	-0.06408
0.13580	0.06972	0.15752	-0.07467
0.17103	0.07786	0.19362	-0.08447
0.20920	0.08505	0.23175	-0.09326
0.24987	0.09113	0.27129	-0.10060
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	0.67710	-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	-0.00370
0.91763	0.01694	0.93852	-0.00075
0.94523	0.01101	0.96509	0.00054
0.96799	0.00600	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.1: S809 Surface Points.

A.2. Annex 2: Blade Cross Section Geometry Data

Below is the table with all blade cross section geometry data, procured from [3].

Radial Distance (m)	Span Station (r/5.029)	Chord Lenght (m)	Twist (degrees)	Thickness (m)	Twist Axis (% chord)
0	0	Hub center of rotation	Hub center of rotation	Hub center of rotation	Hub center of rotation
0.508	0.101	0.218 *	0.0 *	0.218	50 *
0.66	0.131	0.218	0	0.218	50
0.883	0.176	0.183	0	0.183	50
1.008	0.2	0.349	6.7	0.163	35.9
1.067	0.212	0.441	9.9	0.154	33.5
1.133	0.225	0.544	13.4	0.154	31.9
1.257	0.25	0.737	20.04	0.154	30
1.343	0.267	0.728	18.074	20.95% chord	30
1.51	0.3	0.711	14.292	20.95% chord	30
1.648	0.328	0.697	11.909	20.95% chord	30
1.952	0.388	0.666	7.979	20.95% chord	30
2.257	0.449	0.636	5.308	20.95% chord	30
2.343	0.466	0.627	4.715	20.95% chord	30
2.562	0.509	0.605	3.425	20.95% chord	30
2.867	0.57	0.574	2.083	20.95% chord	30
3.172	0.631	0.543	1.15	20.95% chord	30
3.185	0.633	0.542	1.115	20.95% chord	30
3.476	0.691	0.512	0.494	20.95% chord	30
3.781	0.752	0.482	-0.015	20.95% chord	30
4.023	0.8	0.457	-0.381	20.95% chord	30
4.086	0.812	0.451	-0.475	20.95% chord	30
4.391	0.873	0.42	-0.92	20.95% chord	30
4.696	0.934	0.389	-1.352	20.95% chord	30
4.78	0.95	0.381	-1.469	20.95% chord	30
5	0.994	0.358	-1.775	20.95% chord	30
5.305	1.055	0.328	-2.191	20.95% chord	30
5.532	1.1	0.305	-2.5	20.95% chord	30

Table A.2: Blade Cross Section Geometry Data.

*Root hub adapter

References

[1] T. E. W. E. Association, "http://www.ewea.org/," visited: 5.02.2015.

[2] A. Lecuona Neumann, "Energía eólica. principios básicos y tecnología," *Universidad Carlos III de Madrid*, 2005.

[3] M. Hand, D. Simms, L. Fingersh, D. Jager, J. Cotrell, S. Schreck, and S. Larwood, "Unsteady aerodynamics experiment phase vi: Wind tunnel test configurations and available data campaigns," tech. rep., National Renewable Energy Lab., Golden, CO.(US), 2001.

[4] M.-C. Hsu, I. Akkerman, and Y. Bazilevs, "Finite element simulation of wind turbine aerodynamics: validation study using NREL phase vi experiment," *Wind Energy*, vol. 17, no. 3, pp. 461–481, 2014.

[5] P. Giguere and M. Selig, "Design of a tapered and twisted blade for the nrel combined experiment rotor," *NREL/SR*, vol. 500, p. 26173, 1999.

[6] N. N. Sørensen, J. Michelsen, and S. Schreck, "Navier–stokes predictions of the nrel phase vi rotor in the nasa ames 80 ft× 120 ft wind tunnel," *Wind Energy*, vol. 5, no. 2-3, pp. 151–169, 2002.

[7] M. M. Yelmule and E. A. VSJ, "Cfd predictions of nrel phase vi rotor experiments in nasa/ames wind tunnel," *International Journal of Renewable Energy Research (IJRER)*, vol. 3, no. 2, pp. 261–269, 2013.

[8] E. Frias Valero, *Aportaciones al estudio de las máquinas eléctricas de flujo axial mediante la aplicación del método de los elementos finitos*. Universitat Politècnica de Catalunya, 2004.

[9] S. Elgeti and M. Behr, "Block (2d space-time incompressible navier-stokes) derivation," *http://www.cats.rwth-aachen.de:8080/pdf/tn-blkins2dst-derivation.pdf*, visited: 29.01.2015.

[10] M. Behr and T. Tezduyar, "The shear-slip mesh update method," *Computer Methods in Applied Mechanics and Engineering*, vol. 174, no. 3, pp. 261–274, 1999.

[11] M. Behr and T. Tezduyar, "Shear-slip mesh update in 3d computation of complex flow problems with rotating mechanical components," *Computer Methods in Applied Mechanics and Engineering*, vol. 190, no. 24, pp. 3189–3200, 2001.

[12] Y. Song and J. B. Perot, "Cfd simulation of the nrel phase vi rotor," *arXiv preprint arXiv:1404.6183*, 2014.

[13] I. for Advanced Simulation, "Juqueen - http://www.fz-juelich.de/ias/jsc/en/expertise/supercomputers/juqueen/juqueen_node.html," visited: 10.01.2015.