




Universitetet
i Stavanger

FACULTY OF SCIENCE AND TECHNOLOGY

MASTER'S THESIS

Study programme/specialisation: Engineering Structures and Material/ Mechanical Systems	Spring semester, 2019 Confidential
Author: Eivind Innvær	 (signature of author)
Programme coordinator: Dimitrios Pavlou Supervisor(s): Knut Erik Teigen Giljarhus, Maximilian Leinenbach	
Title of master's thesis: Optimization of hydroturbine design using CFD	
Credits: 30	
Keywords: CFD, OpenFOAM, hydroturbine, blade optimization, torque	Number of pages: 78 +supplemental material/other: 2 Stavanger, 14th of June 2019 date/year

Abstract

In this thesis, research on the possibility of increasing the efficiency and torque generated by a turbine is done. The turbine is driven by a flow that is pumped through pipelines by utilizing the kinetic energy of the flow. A manual optimization study is done by running CFD simulations on different turbine geometries with different number of blades (5-7) and blade angles.

The CFD simulations were run with an open source software called OpenFOAM. The simulations use an automatic meshing tool called snappyHexMesh to generate necessary geometries. The traditional k- ϵ model was used for the turbulence modelling. A comparison of torque output from two different approaches (AMI and MRF) was evaluated. The relative difference of torque of these approaches was low. Therefore, the steady-state MRF-approach was used due to its much shorter simulation time.

In the attempt to validate the existing turbine design to experimental data, the torque outputs were unexpectedly low. Thus, it is concluded that the simulation set-up for the CFD analysis can be improved. However, the relative differences in this project is assumed to be the same. For an operational volume flow of 1313 l/min, larger stagger angles and inlet flow angles of the turbine blades reflected an increase of torque for all number of turbine blades evaluated. Lower number of turbine blades showed a higher efficiency, and lower static pressure drop. The best design in this project, show an increase of 21.56% efficiency and 27.3 % torque, compared to the pilot design.

Keywords: turbine, optimization, CFD, OpenFOAM, torque.

Acknowledgements

After 5 years, this project will bring my educational journey to an end. I would like to show my gratitude to my internal supervisor Knut Erik Giljarhus (UiS) and my external supervisor Maximilian Leinenbach (Fishbones). They have both helped me throughout the spring giving me inputs and feedback whenever I got stuck. I am forever grateful for the time, patience and support both supervisors have given me. Last, but not least, I would like to give a huge thanks to my family for the endless support during tough times.

Thank you!

Eivind Innvær, Stavanger, June 2019

Nomenclature

Acronyms

6DOF	6 degrees of freedom
AMI	Arbitrary Mesh Interface
CAD	Computer Aided Design
CFD	Computational Fluid Dynamics
DNS	Direct Numerical Simulation
GAMG	Generalized geometric-algebraic multi-grid
GUI	Graphic User Interface
HSE	Health, Safety and Environment
LES	Large Eddy Simulation
MB	Mega bytes
MRF	Multi Reference Frame
RANS	Reynolds-Average Navier-Stokes
RPM	Rounds per Minute
TSR	Tip speed ratio
UiS	University of Stavanger

Greek Symbols

α	Angle of attack
β_{in}	Flow inlet angle
β_{out}	Flow outlet angle
ϵ	Rate of dissipation of turbulence energy
η	Efficiency
μ	Dynamic viscosity
μ_t	Eddy viscosity
∇^2	Laplacian operator
ν	Kinematic viscosity
ω	Angular velocity
ρ	Density
ΣM	Sum of moments acting on turbine
σ	Turbine solidity
σ_{ij}	Viscous stress tensor
τ_w	Wall shear stress
θ	Stagger angle

Latin Symbols

u, \bar{v}, w	Mean velocities
\dot{Q}	Volume flow
f	Body force

$P_{available}$	Power available in flow
$P_{turbine}$	Power generated by turbine
y^+	Dimensionless distance from wall
c	Chord length
d	Diameter of turbine
g	Gravitational acceleration
H	Hydraulic head
I	Turbulence intensity
k	Turbulence kinetic energy
L	Length of where the flow is moving
l	Turbulence length scale
n	Number of turbine blades
p	Static pressure
r	Turbine radius
Re	Reynolds number
u',v',w'	Fluctuations
v	Fluid velocity
y	Distance from wall
z	Elevation from any point in space from a reference plane

Contents

Abstract	ii
Acknowledgements	iii
Nomenclature	iv
List of Figures	ix
List of Tables	xi
1 Introduction	1
1.1 Scope of project	2
1.2 Computational Fluid Dynamics	3
1.3 Methodology	4
1.4 Previous use of Fishbones Drilling	5
1.4.1 Turbine design	5
1.4.2 Installation and operation process	6
1.5 Thesis Outline	7
2 Theory	9
2.1 Governing Equations	11
2.2 OpenFOAM	14
2.2.1 snappyHexMesh	14
2.2.2 Turbulence Models	16
2.2.2.1 $k-\epsilon$ model	16
2.2.2.2 $k-\omega$ SST model	17
2.2.3 Turbulence boundaries	17
2.3 Near Wall Treatment	19
2.4 Turbine blade design	21
3 Methods	23
3.1 Geometry	24
3.2 Mesh Grid	26
3.2.1 Geometrical optimization	29
3.3 CFD Setup	30
3.3.1 Fluid conditions	30
3.3.2 Rotational Zone	30

3.3.3	Boundary conditions	31
3.3.4	Wall Functions	32
3.3.5	Solvers	33
3.3.6	Schemes	35
3.4	Post Processing	36
4	Case Validation	39
4.1	Mesh Quality	41
4.2	Mesh Convergence study	43
4.3	New vs old CAD-model	48
4.3.1	Boundary layer resolution	50
4.4	Flow fields	51
5	Results and Discussion	55
5.1	Designs	56
5.1.1	7 blades	56
5.1.2	6 blades	59
5.1.3	5 blades	62
5.1.4	Stagger and inlet flow angle effects	65
5.2	Evaluation of the best design	67
5.2.1	Geometry Comparison	68
5.2.2	Flow fields	70
5.2.3	Mesh quality of best design	71
6	Conclusion	73
6.1	Future work	74
	Bibliography	75
A	Appendix 1 - Folder Structure	I

List of Figures

1.1	Illustration of Fishbones technology [2]. Multiple subs attached to the main pipeline with three needles coming out creating laterals.	2
1.2	Schematic illustration of the optimization methodology.	4
1.3	1) String casing 2) Turbine and 3) Needle [2].	6
1.4	1) Stationary hollow tube and 2) Rotating drillbit [2].	6
2.1	Illustration of the blockMesh that cover all the walls and patches in the system.	15
2.2	Velocity distribution near a solid wall [8].	19
2.3	Velocity triangle of velocities acting on blade.	21
2.4	Turbine nomenclature.	21
2.5	Schematics of blade cross sections with velocity triangle, angle of attack α , stagger angle θ , flow inlet angle β_{in} and flow outlet angle β_{out}	22
3.1	Schematic axisymmetric representation of geometries/patches.	24
3.2	Comparison of original design and simplified turbine design.	25
3.3	Comparison of two different quality settings for .stl-files.	25
3.4	Bad mesh quality at the sharp edges and blade surfaces in the pilot turbine design.	26
3.5	Entire domain of the system. Flow coming in from the left hand side.	28
3.6	Refinement inside the rotating zone of the propeller.	28
3.7	Rotational Zone.	31
3.8	Residuals from MRF-simulation with mesh grid (64 64 304) for pressure and velocities.	37
3.9	Convergence of torque for mesh grid (64 64 304) for MRF-simulation.	37
4.1	Schematic illustration of actual hub/turbine positioning inside the casing from the experimental set-up.	39
4.2	Torque vs Volume flow.	40
4.3	The highly skew faces was very small and therefore visualized by spheres with define-able radius instead of faces. Mesh grid size is (40 40 190).	41
4.4	Only 6% deviation from simulation time of 0.04s and 0.1s of AMI-case.	43
4.5	Torque vs number of cells.	45
4.6	Mesh convergence of torque versus number of cells for the AMI-case.	47
4.7	Comparison of the AMI-case and MRF-case of the chosen mesh grid resolution for the pilot design.	47

4.8	Comparison between old (white) and new CAD-model (blue).	48
4.9	Comparison between old (white) and new CAD-model (blue).	49
4.10	Comparison of torque convergence of new and old CAD-model.	49
4.11	y^+ -values of the system from MRF-simulation.	50
4.12	y^+ -values of turbine from MRF-simulation old model.	51
4.13	Pressure field of the entire domain over 2000 iterations of the pilot design.	51
4.14	Pressure field of the cellZone over 2000 iterations of the pilot design.	52
4.15	Averaged velocity field over the turbine over 2000 iterations of the pilot design.	52
4.16	Averaged velocity field over the turbine with velocity vectors over 2000 iterations of the pilot design.	53
4.17	Averaged velocity field of the cellZone upstream (right), at blades (middle) and downstream (right) over 2000 iterations of the pilot design.	54
4.18	Averaged velocity field of the downstream far behind the turbine over 2000 iterations of the pilot design.	54
4.19	Averaged velocity field behind the turbine over 2000 iterations of the pilot design.	54
5.1	Torque-RPM table of 7 blades at volume flow of 1313 l/min.	57
5.2	Efficiency-RPM table of 7 blades at volume flow of 1313 l/min.	57
5.3	Thrust-RPM table of 7 blades at volume flow of 1313 l/min.	58
5.4	Torque-RPM table of 6 blades at volume flow of 1313 l/min.	60
5.5	Efficiency-RPM table of 6 blades at volume flow of 1313 l/min.	60
5.6	Thrust-RPM table of 6 blades at volume flow of 1313 l/min.	61
5.7	Torque-RPM table of 5 blades at volume flow of 1313 l/min.	63
5.8	Efficiency-RPM table of 6 blades at volume flow of 1313 l/min.	63
5.9	Thrust-RPM table of 5 blades at volume flow of 1313 l/min.	64
5.10	Stagger angle behavior for inlet flow angle (IFA) 43 and 46.	65
5.11	Inlet flow angle behavior for stagger angle (S) 75 and 77.	66
5.12	Comparison of the old design (white) and the best design (blue).	68
5.13	Cross-section view of the blade tip of old design (white) and the best design (blue).	69
5.14	Comparison of the old design and the best design.	69
5.15	Averaged velocity field of the best design, v10 , over 2000 iterations.	70
5.16	Pressure field of cellZone of the best design, v10 , over 2000 iterations.	71
5.17	The 2 skewed faces illustrated as spheres at the turbine blades for the best design.	71
A.1	Folder structure.	II

List of Tables

3.1	Pilot design parameters.* is set constant during this project.	29
3.2	Inlet flow velocities.	30
3.3	Boundary conditions used in this project for the AMI case.	32
3.4	Solvers used in this project for the AMI-case.	34
3.5	Solvers used in this project for the MRF-case.	34
3.6	Discretisation schemes used in this project.	35
4.1	Experimental data gathered in Fishbones' lab.	40
4.2	snappyHexMeshDict parameters changed compared to the tutorial case.	44
4.3	Simulation data from mesh convergence study of pilot design using simpleFoam - MRF.	44
4.4	Simulation data from mesh convergence study of pilot design using pimpleFoam - AMI.	45
4.5	Comparison between the old and new CAD-model for mesh grid (32 32 152) at 1927 RPM and 1313 l/min flow rate.	49
4.6	Average y^+ -values for the new and old CAD-model for mesh grid (32 32 152).	50
5.1	Design parameters for the study on 7 blades.	56
5.2	Design parameters for the study on 6 blades.	59
5.3	Design parameters for the study on 5 blades.	62
5.4	Parametric overview of the best designs.	67
5.5	Output overview of the best designs at 1927 RPM and 1313 l/min.	67

1

Introduction

In 1969, Norway became a part of the petroleum industry after the discovery of the offshore oilfield "Ekofisk". At that time, Norway experienced a lack of knowledge about the petroleum industry. Thus, through narrow cooperation with several international oil companies, Norway, as a nation, became the absolute first to produce oil from an offshore oilfield historically. This was roughly one and half years after the discovery (1971). Since then, Norway has become one of the leading oil nations worldwide. With wide experience and innovative solutions, the offshore business has flourished and the production and exploration are still ongoing, 50 years after the first discovery.

Many companies have been established for the purpose to digitalize, improve and make the entire process of each sector; downstream, midstream and upstream, more efficient. One of these companies is *Fishbones*.

Fishbones is a provider of a technology that has defined a new level of precision and efficiency in reservoir stimulation. This means that their technology can be utilized to increase production rates, reduce HSE exposure and accelerate the entire progress by avoiding "downhole" cementing, perforating, cleanouts etc. [1].

During well stimulation, one tries to alter the nearby formation near the main well. In many wells, it might be hard to stimulate the production with for example the traditional "fracturing"-process. In tight sandstone formations, it is very hard for liquid to penetrate through, thus the vertical flow is limited. This might limit the initial expectations of a well bore and can end in a non-profitable project. The solution to such a problem is to utilize Fishbones' Drilling Technology which can handle tight sandstone formations in the well bore [2].

For Fishbones Drilling, a sub is equipped with three needles connected to a turbine. At the tip of each needle there is a drillbit that is driven by the turbine. The turbines are driven by the flow pumped into the main bore. generating rotational torque for the drillbit as well as a thrust pushing the needles through the liner/casing string creating laterals in the formation. For this type of technology and business, it is very important to create satisfactory results for the clients and therefore one large objective for Fishbones is to deliver Dreamliner Subs that can guarantee fully extended needles. For a fully extended needle, the laterals' length can be up to 35ft / 10.8m [2].

1. Introduction

A very useful tool for turbine optimization is the use of Computational Fluid Dynamics (CFD). Instead of expensive and time consuming physical testing, CFD can be used to streamline the product development of the turbine to check whether a concept is good or bad. It can open an efficient and easy way to identify better designs.



Figure 1.1: Illustration of Fishbones technology [2]. Multiple subs attached to the main pipeline with three needles coming out creating laterals.

1.1 Scope of project

This project is done in cooperation with Fishbones AS. Fishbones has a Drilling System for stimulating oil productivity in oil wells. The objective of this project is to research the possibility of increasing the efficiency of the current turbine design for Fishbones Drilling. One are looking to increase the torque output from the turbine by having the same flow rate through the turbine. Increasing the torque might give a reduced RPM and that can create problems for the drillbit. Therefore, a compromise needs to be evaluated. An increased torque can open up a whole new market for Fishbones. If the torque is large enough, the drillbit might be able to drill through the casing and guarantee one hundred percent fully extended needles.

1.2 Computational Fluid Dynamics

Computational Fluid Dynamics, denoted CFD, is a way of analyzing fluid flows, chemical reactions and heat transfer by numerical calculations. In other words, it is an application that can be used in many fields. Instead of expensive and time consuming physical testing to detect potential flaws in an idea or design, one can utilize CFD and retrieve accurate solutions to a very wide range of problems. CFD can normally be divided into three parts: pre-processing, processing and post-processing.

In order to start the pre-processing process, one needs to have an initial plan or idea of what the analysis will involve. From there, one normally create a geometry using a CAD (Computer Aided Design) software. However, there are other ways to create geometries at this stage, but these will not be discussed. The geometry will then need a defined mesh quality depending on how complex the problem is and how accurate results that are required. Then, the fluid properties are specified. Finally, one have to specify boundary conditions, i.e. inlet velocity and outlet pressure.

During the solving process, a solver is to be selected. This solver should satisfy the physical nature of the problem. The solver will then, through numerical calculations, solve the defined problem from the pre-processing process. In this project, OpenFOAM is used for the CFD simulations. Throughout the years, the OpenFOAM team has developed and updated multiple solvers associated with rotating objects that are available in OpenFOAM [10]. *Mehdipour* did research on the use of AMI (Arbitrary Mesh Interface) and MRF (Multi Referenced Frame) in OpenFOAM. The purpose of that study was to compare the hydrodynamic performance of a propeller in both open-water and self-propulsion conditions. By comparing the simulation results from AMI and MRF in the open-water, the paper concluded that the results from both cases reflected a close agreement to the experimental data. However, for the self-propulsion simulations, the MRF approach requires multiple experiments in order to adopt the experimental values because the flow around the turbine is not instantaneous, but averaged. This was however not a problem for the AMI approach, as this is a time-dependent approach and propeller actually rotate. This enables the possibility to precisely simulate the flow around the propeller [6].

The post-processing is the last part. This involves analyzing the results from the solving process. OpenFOAM has a tool called *ParaView* for this. In *ParaView*, it is possible to visualize the numerical results. There are multiple built-in tools to create a better understanding of the results as well. Some tool examples are: streamline of the flow, contour plots, glyph plots etc. A common way of representing the results is by plotting the results vs time or number of iterations. It is also normal to plot residuals to verify the credibility of the solver. Another, is to evaluate the flow fields behave through the domain.

1.3 Methodology

Firstly, a study will be done by running CFD analysis on the current design with the goal to try to recreate already collected experimental data. The main focus is not to obtain the exact data, but create a decent case set-up that shows the same patterns and behavior as the experimental data. Thereafter, minor modifications to the blade design will be done. By varying number of blades and different angles, several concepts will be simulated and studied to identify their behavior at different RPMs. Torque, thrust and efficiency of the turbine will be evaluated for all concepts. OpenFOAM is a popular open-source CFD simulation software. Because of its flexibility and availability, this software was preferred by Fishbones as it is a cheap solution for a turbine optimization study in the future. Because of the complexity of rotating turbine simulations, it requires heavy computational power. Therefore, one will seek a case set-up that can complete a simulation in a medium amount of time and still obtain accurate data. In this project, one has access to server cluster at UiS. This allows for faster runs and parallel solving. All cases will be run with 20 cores. A comparison between MRF and AMI will be done in this project to evaluate the differences in simulation time and torque output.

The entire optimization cycle can be illustrated as shown in Figure 1.2.

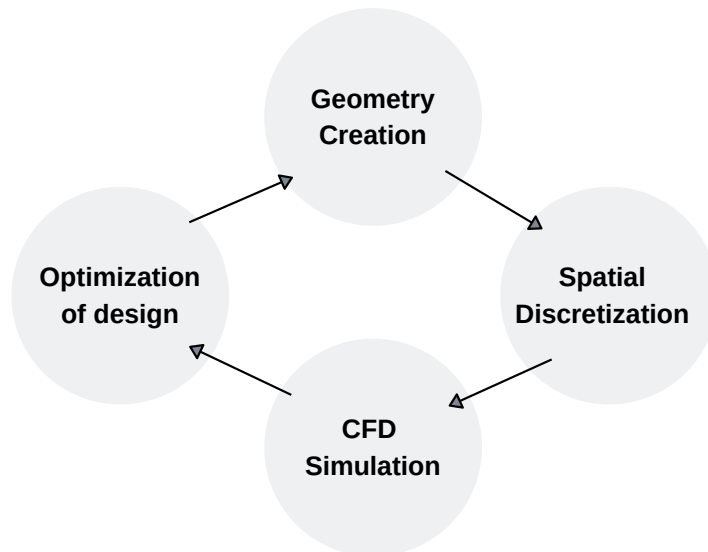


Figure 1.2: Schematic illustration of the optimization methodology.

Firstly, the wanted geometry is created, then by spatial discretization a mesh is created which is necessary for the CFD simulation in the next step. Next, one analyze the results from the simulation and do necessary optimization of the design.

1.4 Previous use of Fishbones Drilling

In 2015, Fishbones Drilling was installed in an offshore horizontal well located in the Norwegian Sea. During the creation of this well, the drilling team met a few challenges. One of challenges was due to the tight sandstone formation. Another challenge was that there was a depleted gas reservoir located rather close to the well. Therefore one could not use traditional hydraulic fracturing stimulation technology, often known as fracking. A stimulation technology without the risk of fracturing into the gas reservoir was therefore sought.

As mentioned earlier, sandstone has a low permeability and by using the traditional stimulation technology one would expect not only limited vertical flow, but also uncontrolled fracture in the formation. Initially the well was considered as non-economical to develop and the risk was too big [1].

In a report from Equinor, Petteresen addressed: "This has been a world class challenge. Very few offshore fields have been developed with such low permeability under normal pressure conditions" [3]. Thanks to Fishbones Drilling Technology one could not only see an economical project, but also an increased oil recovery and productivity from the reservoir reported two years after installation [7].

1.4.1 Turbine design

The current turbine design have had multiple successful runs down-hole. The turbine design was made by a former Fishbones employee, and there lacks documentation on the design process. Nevertheless, it is understood that the design development was done by a "try and fail"-approach. The development team did a lot of physical testing for multiple designs. They only looked at the same number of blades (7) and the twist between root and tip of the blades. This was a long and expensive process and the testing stopped as soon as a satisfying compromise between torque and RPM was accomplished. It was and still is, quite important to have enough RPM. If the RPM-value is not sufficient enough, the drillbit might experience problems when drilling. The turbine design was created in a CAD-software called SolidWorks and are provided by Fishbones.

1.4.2 Installation and operation process

As previously mentioned, each Dreamliner Sub contains three needles. Each drillbit is driven by a turbine, see Figures 1.3 and 1.4. The turbine is driven by the main flow through the bore pipe. There is no preparation needed on deck. The Dreamliner Subs are designed to be run in-hole and therefore easy to install. Dreamliner Anchors are also designed for easy installation. These anchors are added on deck in between the pipes to avoid temperature elongations of the subs. The string is then run into the open hole. As the flow increases the turbine and needles will reach the desired operational RPM. When the RPM is high enough, the needles are released into the formation, creating laterals. The turbines will continue rotating until the needles are fully extended. Multiple laterals have then easily been created, increasing the production rates.

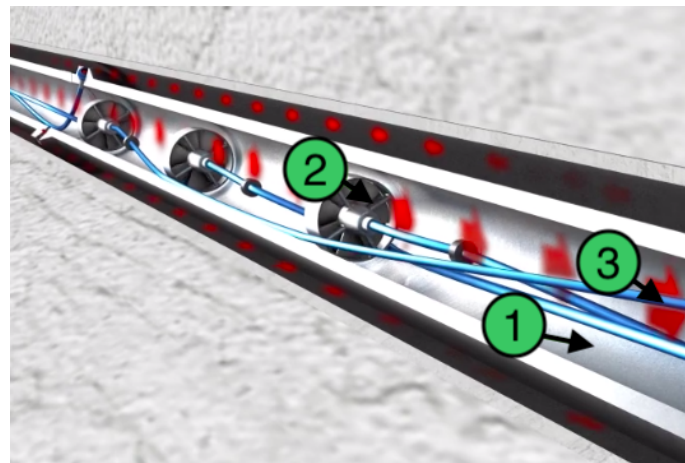


Figure 1.3: 1) String casing 2) Turbine and 3) Needle [2].

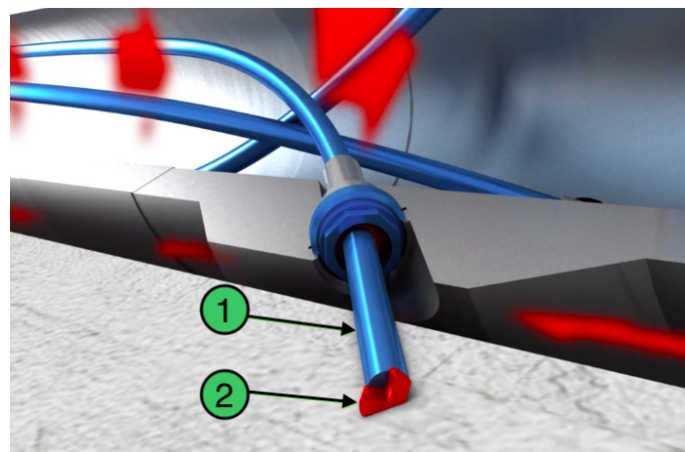


Figure 1.4: 1) Stationary hollow tube and 2) Rotating drillbit [2].

1.5 Thesis Outline

There are 6 chapters in this thesis. Firstly, an introduction to this project is described. Secondly, the relevant theory necessary is presented. In chapter 3, the methods used to meet the scope of the project is presented. Next, in chapter 4, a validation of the method are evaluated to verify the case set-up. In chapter 5, the results from the optimization of the turbine are presented and discussed. Lastly, to end this project, a conclusion and suggestions for future improvements are presented.

2

Theory

To get a better understanding about how to meet the scope of this project one need to dig into more details about the subject: *turbomachinery*. Turbomachinery is described as machines that transfer energy between a rotor and a fluid. In this project one are looking at a turbine that utilize the kinetic energy in the pumped water to make the turbine rotate. By having the same volume flow through the turbine, one are looking to optimize the turbine design to improve the torque output and the efficiency. To achieve this, there are several airfoil parameters that need to be evaluated. The complexity of turbine blade optimization is large, but in this project the chosen variable parameters for optimization are as follows:

- Flow inlet angle, β_{in}
- Stagger angle, θ
- Number of rotor blades, n
- Chord length, c

Because of the great variety and complexity of turbine designs it is very difficult to find a perfect solution. In most cases one need to create a compromised solution between airfoil parameters. The efficiency of a hydroturbine can be defined as the ratio of how much energy that is collected from the water flow and the available energy:

$$\eta = \frac{P_{turbine}}{P_{available}} \quad (2.1)$$

$P_{turbine}$ is the total power that the turbine generates and $P_{available}$ is the total power available in the flow.

Dixon et. Al. [11] states that the effects of turbulence on the performance of turbomachines is small. The viscous boundary layers are generally turbulent at turbine blades, thus the turbulence effects on the global flow field is low. On the other hand, efficiency is the factor that might be affected the most. In most cases where the fluid is water, these turbulence effects may be neglected when studying the efficiency due to the relative low kinematic viscosity. They also states that at high Reynold numbers that surface roughness at the blades are very important. Greater roughness tends to increase the friction losses and decrease the total efficiency of the turbine [11].

2. Theory

To understand what affects Equation 2.1, one need to look at how $P_{turbine}$ and $P_{available}$ can be expressed further. The total power that the turbine generates and the available power in the flow can be expressed as,

$$P_{turbine} = \omega \Sigma M \quad (2.2)$$

$$P_{available} = \dot{Q} \rho g (\Delta H) \quad (2.3)$$

where ω is the angular velocity, ΣM is the sum of all the moments acting on the turbine. \dot{Q} is the volumetric flow rate, ρ is the density of the fluid, g is the gravity and ΔH is the difference in total hydraulic head.

A couple of assumptions have been made in this project to simplify the theory. The first one is that there is no energy loss due to friction in the piping, and second is that the pipe is horizontal. This means that ΔH is 0. Zero head means that altitude of incoming water is the same before and after the passing the turbine (see Figure 1.3). Due to this, a deeper look into the total static pressure drop of the horizontal pipeline is necessary.

Bernoulli's principle for a steady incompressible fluid is valid for any particle within a flow at any point in the control volume and is expressed as:

$$\frac{1}{2}v^2 + gz + \frac{p}{\rho} = constant \quad (2.4)$$

where v is the instantaneous fluid particle velocity, g is gravity, z is the elevation of any point away from a reference plane, p and ρ are pressure and fluid density. The density from Equation 2.4 remains constant because the mass is conserved throughout the system due to the assumption about incompressibility. The only changes to the fluid before and after passing the turbine is the pressure and the velocity. The net energy of the Bernoulli's equation is the sum of kinetic energy, potential energy and pressure. By considering energy conservation, the specific hydraulic energy term can be introduced [11]:

$$\frac{1}{2}v^2 + gz + \frac{p}{\rho} = gH \quad (2.5)$$

To get rid of ΔH in Equation 2.3, one can use Equation 2.5 to find the total pressure drop over the turbine:

$$\begin{aligned} \Delta H &= 0 \\ \left(\frac{1}{2}\rho v_1^2 + \rho g z_1 + p_1\right) - \left(\frac{1}{2}\rho v_2^2 + \rho g z_2 + p_2\right) &= 0 \end{aligned} \quad (2.6)$$

The total pressure, p_0 , is the sum of the static and dynamic pressure and the gravitational head. The total pressure can be defined as:

$$p_0 = \frac{1}{2}\rho v^2 + \rho g z + p = \rho g H \quad (2.7)$$

Thus the equation for the power available in the flow and the efficiency can be derived (Δp is the pressure drop over the turbine):

$$P_{available} = \dot{Q}(\Delta p_0) \quad (2.8)$$

$$\eta = \frac{\omega \Sigma M}{\dot{Q} \Delta p} \quad (2.9)$$

2.1 Governing Equations

Because the entire project is based around turbines there are a few theoretical properties to understand. The flow behind a turbine is in general understood to be irregular and chaotic and is described as a turbulent flow. The definition of a turbulent flow is that the flow pattern of a fluid particle has big changes in both pressure and velocity. Whether a flow is turbulent or not can be described theoretically with the Reynolds number,

$$Re = \frac{\rho v L}{\mu} \quad (2.10)$$

Where v is fluid velocity, L is the length of where the flow is moving and μ is the dynamic viscosity of the fluid. Usually the transition from laminar (stable) to turbulent flow, in a pipeline, takes place somewhere between $Re \approx 2000$ and 10^5 [8]. At the end of the day, the transition area for a flow is a highly discussed subject. It all really depends on the assumptions that are made and what application that is used.

In fluid dynamics there has been developed multiple models to simulate turbulence, i.e. Reynolds-Average Navier-Stokes (RANS), Direct Numerical Simulation (DNS) and Large Eddy Simulation (LES). In OpenFOAM one have Reynold-Average Simulation (RAS), which again can be divided into compressible and incompressible models. In this project, one are assuming a Newtonian fluid and describes the turbulence by the traditional $k - \epsilon$ model.

The nature of the turbine in this project is considered to be incompressible, turbulent and both transient (AMI) and steady-state (MRF). For an incompressible fluid, the mass continuity equation remains the same for steady and unsteady flows and can be expressed by [28]:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (2.11)$$

Where u , v and w are the velocities in each corresponding dimension.

RAS is a type of RANS-modelling. They both predict the turbulent flow by the use of averaging the Navier-Stokes equations. The Navier-Stokes equations, which is a cornerstone in fluid dynamics, can be expressed in many ways. In this project one are considering both a steady-state and a unsteady case. A simplified expression of the Navier-Stokes equations for an incompressible fluid are shown below.

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} + f_i \quad (2.12)$$

where ∇p is the pressure field, \mathbf{u} is the velocity field, ρ is the density and f_i is the body force. The denotation i can be substituted by x, y and z depending on what directions in space that are considered. ∇^2 is the Laplacian operator and is expressed as:

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \quad (2.13)$$

Cebeci et. Al.[28] expresses the momentum equations for each corresponding dimension respectively:

$$\begin{aligned} \rho \frac{Du}{Dt} &= -\frac{\partial p}{\partial x} + \left(\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} \right) + \rho f_x \\ \rho \frac{Dv}{Dt} &= -\frac{\partial p}{\partial y} + \left(\frac{\partial \sigma_{yx}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z} \right) + \rho f_y \\ \rho \frac{Dw}{Dt} &= -\frac{\partial p}{\partial z} + \left(\frac{\partial \sigma_{zx}}{\partial x} + \frac{\partial \sigma_{zy}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} \right) + \rho f_z \end{aligned} \quad (2.14)$$

The viscous and normal stresses for an incompressible viscous fluid can be given by the viscous stress tensor:

$$\sigma_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (2.15)$$

Whenever turbulence is presence in RANS-modelling, the velocity of the flow is decomposed into the sum of mean velocity and the fluctations in order to obtain conservation equations:

$$\begin{aligned} u &= \bar{u} + u' \\ v &= \bar{v} + v' \\ w &= \bar{w} + w' \end{aligned} \quad (2.16)$$

The Navier-Stokes from Equation 2.12 can now be expressed in the following form:

$$\begin{aligned} \rho \frac{D\bar{u}}{Dt} &= -\frac{\partial p}{\partial x} + \nu \nabla^2 \bar{u} + \rho f_x - \rho \frac{\partial}{\partial x} (\bar{u}'^2) - \rho \frac{\partial}{\partial y} (\bar{u}'\bar{v}') - \rho \frac{\partial}{\partial z} (\bar{u}'\bar{w}') \\ \rho \frac{D\bar{v}}{Dt} &= -\frac{\partial p}{\partial y} + \nu \nabla^2 \bar{v} + \rho f_y - \rho \frac{\partial}{\partial x} (\bar{v}'\bar{u}') - \rho \frac{\partial}{\partial y} (\bar{v}'^2) - \rho \frac{\partial}{\partial z} (\bar{v}'\bar{w}') \\ \rho \frac{D\bar{w}}{Dt} &= -\frac{\partial p}{\partial z} + \nu \nabla^2 \bar{w} + \rho f_z - \rho \frac{\partial}{\partial x} (\bar{w}'\bar{u}') - \rho \frac{\partial}{\partial y} (\bar{w}'\bar{v}') - \rho \frac{\partial}{\partial z} (\bar{w}'^2) \end{aligned} \quad (2.17)$$

These equations are a general presentation of the RANS-modelling equations. In this project the MRF- and AMI-approach are evaluated. As the turbine in the MRF-approach, do not rotate, a few modifications to the momentum equations are necessary to account for this. Petit et. Al. [35] show that one extra term appears in the equations for the MRF-approach. The equations are modified with Coriolis and centrifugal terms. The stationary frame observes the velocity field of the fluid with the quantity *absolute velocity*. Inside the rotating frame, the velocity is observed with a *relative velocity*. The absolute velocity are expressed as:

$$u_{abs} = u_r + \omega \times r \quad (2.18)$$

The Navier-Stokes equations for each frame can be expressed in the follow equations [34]:

$$\begin{aligned}
\nabla \cdot (u_{abs} \times u_{abs}) &= -\nabla \frac{p}{\rho} + \nu \nabla \cdot \nabla (u_{abs}) \\
\nabla \cdot (u_r \times u_r) + 2\omega \times u_r + \omega \times \omega \times r &= -\nabla \frac{p}{\rho} + \nu \nabla \cdot \nabla (u_r) \\
\nabla \cdot (u_{abs} \times u_r) + \omega \times u_{abs} &= -\nabla \frac{p}{\rho} + \nu \nabla \cdot \nabla (u_{abs})
\end{aligned} \tag{2.19}$$

The equation on the top yields for the stationary frame for absolute velocity. The middle and the bottom equation yields for the rotating frame for relative and absolute velocity. The Coriolis force is $(2\omega \times u_r)$ and the centrifugal force is $(\omega \times (\omega \times r))$.

2.2 OpenFOAM

In OpenFOAM there is multiple solvers available that are associated with rotating objects. Two of these are described below. **MRF** (Multi Referenced Frame) is a steady-state approach. It is fairly simple and robust. The approach splits the domain into multiple referenced frames; one rotating reference frame for the turbine and stationary frames for the rest of the domain. The turbine do not physically rotate even though it is positioned inside the rotating frame. Therefore it does not accurately represent the true unsteady flow. However, MRF uses an approximation of how the flow behaves at an instant of rotation and generates robust solutions when the case it set up correctly. In this project, *simpleFoam* was the solver used for this approach.

AMI (Arbitrary Mesh Interface) is an unsteady transient approach. It is more computationally heavy as the turbine is to rotate inside the rotating zone/frame. Through *cyclicAMI*, which is a boundary condition, the stationary and rotational regions are coupled at patch boundaries. All values are updated and calculated at every time step. As the turbine is rotating the flow in the domain acts accordingly. In this project, *pimpleFoam* was the solver used for this approach.

The OpenFOAM team is constantly searching for more robust and smart solutions for their CFD software. The AMI/MRF study done by *Mehdipour* [6] was back in 2013. In the recent update (2018, July), version 6 of OpenFOAM, major improvements to already existing codes and case folders were worked out. The most interesting update related to this project was the rotating/sliding geometry update [9]. These updates will not be researched nor validated.

2.2.1 snappyHexMesh

In OpenFOAM there is a utility available for automatic meshing of 3D geometries. In this section, a brief introduction about the possibilities *snappyHexMesh* gives will be presented.

snappyHexMesh reads the triangulated surfaced geometries generated by the formats likes .stl- and .obj-files, and then creates a mesh that contains of hexahedra and split-hexahedra. In *snappyHexMeshDict*, one can simply select a mesh refinement level for both surfaces and regions that determines the final quality of each region in the system. *snappyHexMeshDict* has many other parameters to play with as well, for instance: *snapControls*, *addLayersControls* and *meshQualityControls*. It is worth mentioning there are many sub-parameters under each of these mentioned above that will at the end of the day affect the final mesh quality.

The process of creating a successful and good mesh with *snappyHexMesh* might feel time-consuming and difficult in many cases. The meshing process in OpenFOAM is not directly connected to a graphic user interface (GUI) like many other programs like Pointwise, Ansys etc. OpenFOAM uses ParaView for post-processing. ParaView reads the outputs from running *snappyHexMesh* in OpenFOAM, and has many tools that can create a good visualization of the mesh and the problem. These

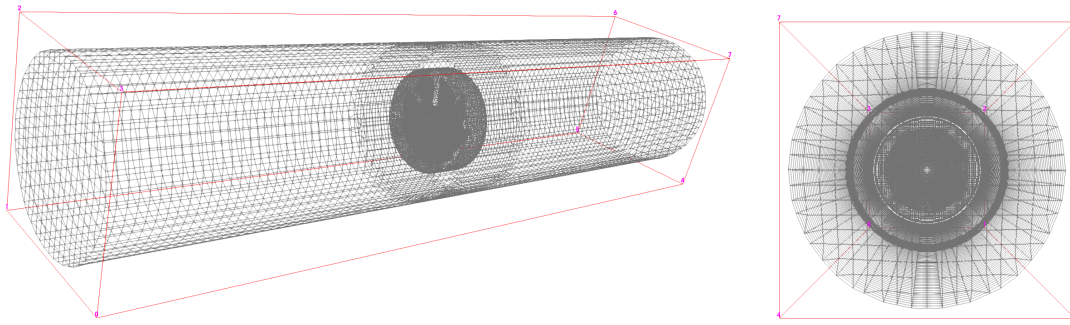


Figure 2.1: Illustration of the `blockMesh` that cover all the walls and patches in the system.

output files are a result of numerical calculations and a large hierarchy of C++ coding. Before `snappyHexMesh` can be used, one need to define a background mesh. This background mesh needs to fill the entire region within the external boundary. `blockMesh` is a simple tool that can be used for this process.

When the background hex mesh is created, `snappyHexMesh` moves on to the cell splitting at feature edges and surfaces inside the red border shown in Figure 2.1. The pre-defined quantities determines how good mesh quality the edges and surfaces gets. As mentioned above, there are many parameters available that affect the mesh quality. After the refinement and cell splitting is done, `snappyHexMesh` removes unwanted cells in the system. What cells to be removed, can be specified in the `snappyHexMeshDict` with `LocationInMesh`. Then, further cell splitting can be done if requested. This can be done to specified regions within the system. The next step is to move cell vertex points onto surface geometry to get rid of bad quality surfaces from the refinement and cell splitting process. The two last parts of the `snappyHexMeshDict` are `Mesh Layers` and `Mesh quality controls`. `Mesh Layers` can be turned on to add further cell splitting into the already split cells. This time, the thickness of each cell can be chosen and that can be very helpful and powerful for sharp and detailed geometries. `Mesh quality controls` is the quantities that tells when the numerical algorithms have a good enough solution to proceed. One example is the maximum non-orthogonality allowed in the entire mesh. Whenever `snappyHexMesh` is done, a simply check can be done to verify if the mesh is "OK". To check whether the Mesh quality controls are satisfied or not can be done by typing `checkMesh` in the terminal [17]. These controls check faces, cells and the orthogonality in the mesh, to see whether they satisfy the pre-defined parameters. It is difficult to avoid all errors by using an automated meshing tool. Therefore a validation of the mesh quality is done in Chapter 4, for the pilot design and the final best design from the optimization.

2.2.2 Turbulence Models

It is quite hard to fully describe the chaotic movement of particles in turbulence, but many approximate models have been developed through history. As mentioned previously, OpenFOAM comes with various turbulence models. Although there exists many models, it requires sufficient knowledge about the models to fully understand which one to choose for the specific problem one are trying to solve. The $k - \epsilon$ model is considered as a standard and good approach to start CFD analysis of a turbulence problem.

2.2.2.1 k- ϵ model

The most common numerical model for turbulent flows that uses Navier-Stokes equations is the k- ϵ model. This model focuses on mechanisms that affect the kinetic energy and the dissipation of the flow. The $k - \epsilon$ model is one of the simplest turbulence models and it is considered as a *two equation model*. This means that they include two extra equations to describe the turbulent flow. In the $k - \epsilon$ model, these equations are Equations 2.20 and 2.21 respectively. The reason for this models popularity is that it does not depend on the flow-input nor the geometry of the problem. This model is considered to be a robust and stable model for early problem analysis and are often set as default turbulence model by many industrial applications. Because of this, the k- ϵ model was used in this project.

Nevertheless, it is a few negative things about this model. As mentioned previously, there are a few constants that have been calibrated and almost set as standard due to their good fit for engineering problems. These adjustable constants are very accurate in cases where handling thin shear layer, recirculating and confined flows [14]. Thus, deviation from such behaviours may result in inaccurate and poor performance of the $k - \epsilon$ model.

In Equation 2.17, all terms have the time derivative included. This means that these Navier-Stokes equations are valid for the transient case (AMI). For the steady-state case (MRF), the left hand side of the equations becomes 0. As briefly discussed above, the $k - \epsilon$ model introduces two additional transport equations. For an unsteady flow they can be expressed as:

$$\frac{\partial(\rho k)}{\partial t} + \nabla \cdot (\rho k \mathbf{u}) = \nabla \cdot \left[\frac{\mu_t}{\sigma_k} \nabla k \right] + 2\mu_t \cdot S_{ij} \cdot S_{ij} - \rho \epsilon \quad (2.20)$$

$$\frac{\partial(\rho \epsilon)}{\partial t} + \nabla \cdot (\rho \epsilon \mathbf{u}) = \nabla \cdot \left[\frac{\mu_t}{\sigma_\epsilon} \nabla \epsilon \right] + C_{1\epsilon} \frac{\epsilon}{k} 2\mu_t S_{ij} \cdot S_{ij} - C_{2\epsilon} \rho \frac{\epsilon^2}{k} \quad (2.21)$$

Again, in order to achieve steady state, the time derivative need to be set equal to 0 in Equation 2.20 and 2.21. These equations are used to derive k , the turbulence kinetic energy, and ϵ , the rate of dissipation of the turbulence energy. The eddy viscosity in these equations is defined as,

$$\mu_t = \rho C_\mu \frac{k^2}{\epsilon} \quad (2.22)$$

In both Equation 2.20 and 2.21 there are multiple adjustable constants. Through decades of comprehensive data fitting for a great diversity of turbulent flows, the

values for these constants are provided as follow [8]:

$$C_\mu = 0.09, \sigma_k = 1.0, \sigma_\epsilon = 1.3, C_{\epsilon 1} = 1.44, C_{\epsilon 2} = 1.92.$$

The Prandtl numbers σ_k and σ_ϵ gives a relation between the eddy viscosity, k and ϵ . The constants $C_{\epsilon 1}$ and $C_{\epsilon 2}$ allow for the correct proportionality between Equation 2.20 and 2.21.

2.2.2.2 $k - \omega$ SST model

The $k - \omega$ model is another commonly used turbulence model. This is also considered as a *two equation model*. The ω part of this model describes the specific turbulence dissipation rate. It describes the scale of the turbulence. What makes the $k - \omega$ model good, is that the model is usable down to the wall through the viscous sub-layer. One can easily integrate the equations for the kinetic energy and specific dissipation directly and can avoid using the damping functions that the $k - \epsilon$ uses. Respectively, these equations are[16]:

$$\frac{Dk}{Dt} = \frac{\partial}{\partial x_k} \left[\left(\nu + \frac{k}{\omega \sigma_k} \right) \frac{\partial k}{\partial x_k} \right] + R_{ik} \frac{\partial u_i}{\partial x_k} - \beta^* k \omega \quad (2.23)$$

$$\frac{D\omega}{Dt} = \frac{\partial}{\partial x_k} \left[\left(\nu + \frac{k}{\omega \sigma_k} \right) \frac{\partial \omega}{\partial x_k} \right] + \alpha \frac{\omega}{k} R_{ik} \frac{\partial u_i}{\partial x_k} - \beta \omega^2 \quad (2.24)$$

How to calculate both k and ω from Equations 2.23 and 2.24 can be identified in *Turbulence Models and Their Application* [16].

The $k - \omega$ **Shear Stress Transport (SST) model** is another quite popular turbulence model. It combines the best features from both the $k - \epsilon$ and the $k - \omega$ model. It is said that this model often gives good behaviour in adverse pressure gradients and separating flow [15]. This describes the behaviour of a rotating turbine and in which case are assumed to be a good and quite suitable model in this project. By changing some of the constants and adding blending functions to Equations 2.23 and 2.24, one get a hybrid model. The blending function allows the use of the near wall treatment in the $k - \omega$ model and the $k - \epsilon$ models free stream advantage further away from the walls simultaneously. Again, because of the complexity of the theoretical approach of this model, one will not go further into detail in this project. See Cebeci [16] for details.

2.2.3 Turbulence boundaries

In CFD it is very important to have accurate initial and boundary conditions for the case to achieve as good results as possible. By deriving Equations 2.20 and 2.21 from Section 2.1, one can identify the equations for k and ϵ . In order to choose the most accurate initial conditions, one need to define several turbulent quantities. The first quantity is the turbulence intensity. The turbulence intensity is based on the Reynolds number of the stream [22]:

$$I = 0.16 Re^{(-1/8)} \quad (2.25)$$

2. Theory

By using the initial inflow conditions and the turbulence intensity the turbulence kinetic energy can be calculated by:

$$k = \frac{3}{2}(v \cdot I)^2 \quad (2.26)$$

When the turbulence kinetic energy is calculated, the dissipation of turbulence energy can easily be calculated, but first the turbulence length scale need to be defined. The turbulence length scale is a quantity that is related to the size of the large eddies that contain energy in the flow. In this project, the large eddies is restricted by the size of the pipe/casing. This is because the eddies cannot become larger than the pipe/casing it self. A suitable relationship between the turbulence length scale and pipe dimension is (L is the diameter of the pipe) [22]:

$$l = 0.07L \quad (2.27)$$

$$\epsilon = C_\mu \frac{k^{3/2}}{l} \quad (2.28)$$

2.3 Near Wall Treatment

In this project, the flow through the turbine is quite important. One are trying to extract as much energy from the flow as possible, and use it to generate mechanical energy. The turbine it self, is considered as a wall. This means fluid particles cannot go through it, but have to make their way around. Because of the presence of such a solid, the flow behavior becomes turbulent. How the flow behave close to this wall is therefore necessary to study. It will determine whether the case is good or not. The best way to achieve accurate wall behavior is to have small enough cells in the mesh grid. This is straightforward, but having many cells might be time consuming and computationally heavy. CFD researchers have therefore developed different wall functions to describe the turbulence in different problems.

The equation for Reynolds Number (Equation 2.10) can be presented in many forms, and one of them is a based on the distance, y , the flow is from the wall.

$$Re_y = \frac{vy}{\nu} \quad (2.29)$$

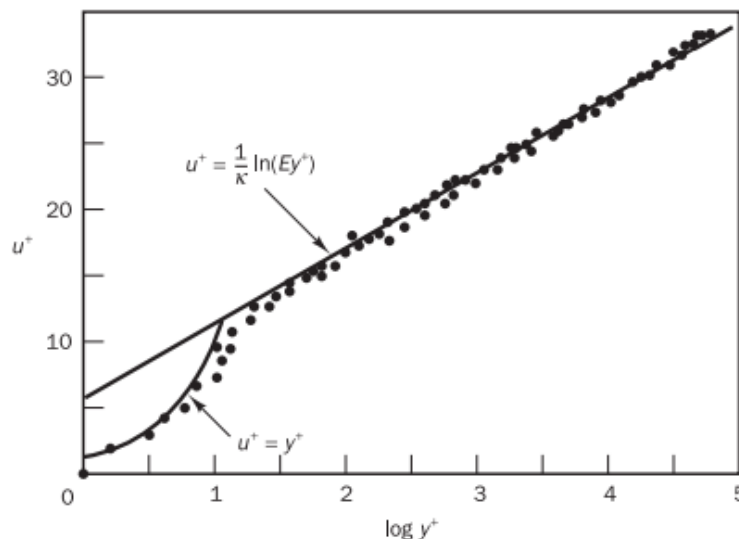


Figure 2.2: Velocity distribution near a solid wall [8].

From this equation one, can see that the Reynolds number will be larger further away from the wall, but when y approaches 0, the Reynolds number will become 1. And whenever y is 0, so will the Reynolds number. To understand this better one need to look at the turbulent boundary layer and introduce y^+ ; a dimensionless quantity derived from y . The turbulent boundary layer close to a wall is divided into two regions; the inner and the outer region.

The inner region is again divided into three layers; the linear sub-layer, buffer-layer and finally, the log-law layer. Closest to the turbines surface, where y^+ is the lowest, the viscous stresses is dominating. It is a very thin layer ($y^+ < 5$) and the shear stress is assumed to be constant and equal to the wall shear stress through the entire layer. Therefore, we have a linear relationship between u^+ and y^+ :

$$u^+ = y^+ \quad (2.30)$$

In the log-law layer, both turbulence and viscous effects, are present. With a few assumptions, one can derive a relationship between u^+ and y^+ in this region. In addition to this, a few numerical constants are introduced. Karman's constant ($\kappa \approx 0.4$) and the additive constant ($E \approx 9.8$):

$$u^+ = \frac{1}{\kappa} \ln(Ey^+) \quad (2.31)$$

The log-law layer is said to be effective in the y^+ -range between 30 and 500 [8]. One can see a close correlation to between experimental data and the equation introduced for linear sub-layer and the log-law layer, Equation 2.30 and 2.31, respectively [8]. The turbulent stresses dominate in this layer. Between the linear sub-layer and the log-law layer one have the buffer layer ($5 < y^+ < 30$). This is a transition layer where the viscous stresses and the turbulent stresses are assumed to be quite similar.

When it comes to determining the magnitude of the dimensionless quantity y^+ , it is closely related to the actual distance from the wall, wall shear stress, density and the kinematic viscosity of fluid.

$$y^+ = \frac{y}{\nu} \sqrt{\frac{\tau_w}{\rho}} \quad (2.32)$$

y is the distance from the wall, ν is kinematic viscosity, τ_w is the wall shear stress and ρ is the density of the fluid. The magnitude of both the kinematic viscosity and density, is based on what type of fluid as well as its temperature. The wall shear stress can be expressed as:

$$\tau_w = \mu \frac{\partial v}{\partial y} \quad (2.33)$$

Here the μ is the dynamic viscosity, v is the velocity and y is the distance from the wall.

2.4 Turbine blade design

The objective, as mentioned, is to evaluate the possibility to increase the torque generated by the available kinetic energy in the flow inside the casing. To achieve this, it is important to have an understanding of the theory that goes into blade design. A Kaplan turbine is a close approximation to this project as the curvature is quite similar. Kaplan turbines are positioned axial to the flow and are driven by the reaction of water flowing through it. The turbine start to rotate due to the lift forces that is generated by the pressure differential on the blades.

The reason for this can be described with so-called velocity triangles [18], but one will not dig too much into detail about this in this project. The main focus will be to study different inlet flow angles and stagger angles, number of blades and chord lengths. Fishbones stated that they wanted a turbine quite similar to the one that is already commercialized. This means that the diameter for the turbine will remain the same, as well as no optimized solution for the hub will be studied.

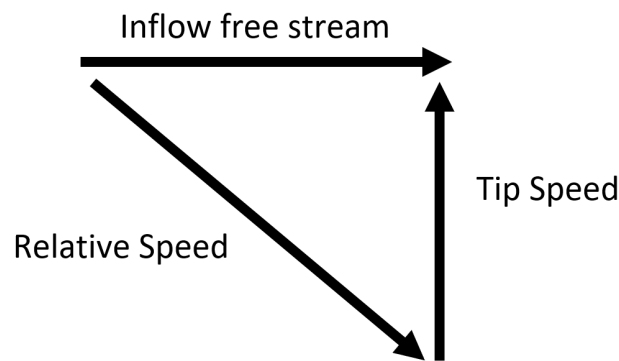


Figure 2.3: Velocity triangle of velocities acting on blade.

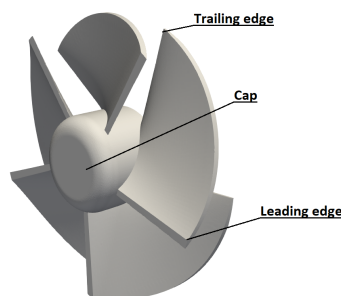


Figure 2.4: Turbine nomenclature.

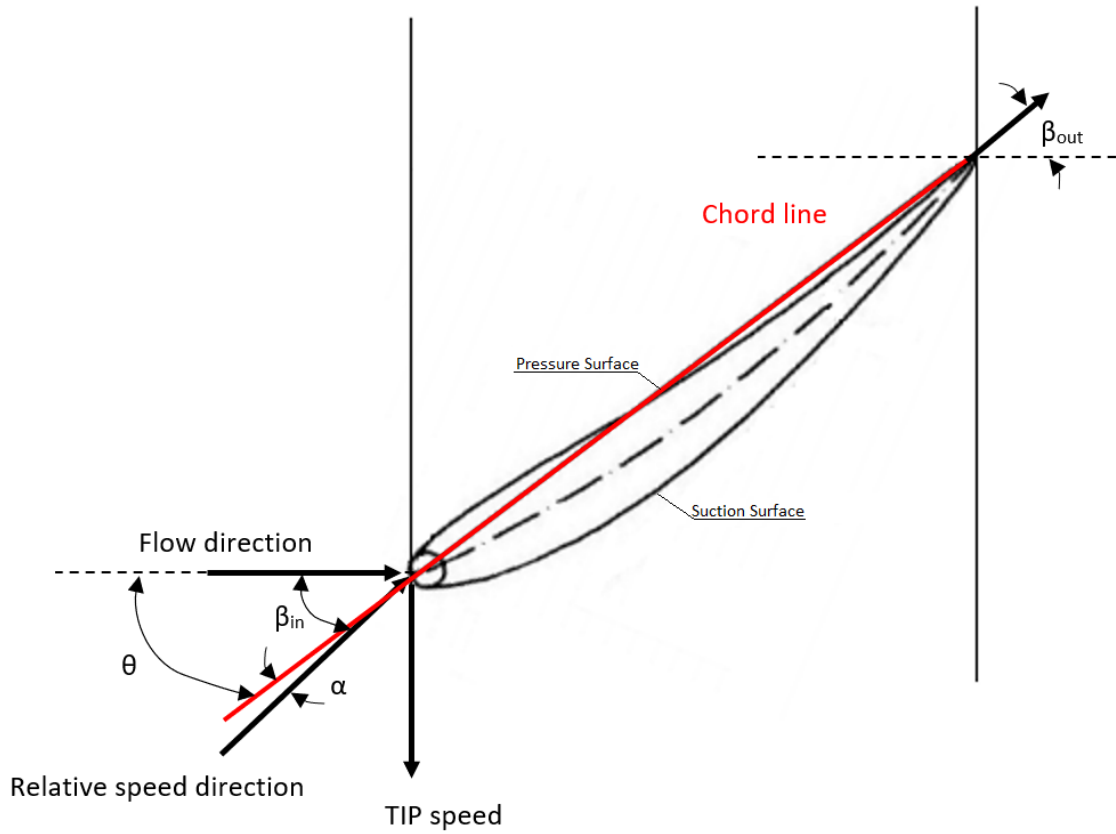


Figure 2.5: Schematics of blade cross sections with velocity triangle, angle of attack α , stagger angle θ , flow inlet angle β_{in} and flow outlet angle β_{out} .

Mukherji et. Al. [20] found that an increase in turbine solidity and number of blades, increase the power coefficient for any tip speed ratio (TSR). The tip speed ratio is ratio between angular velocity times the radius divided by the inflow velocity. Mukherji et. Al. also stated that the pressure drop increases with higher solidity. The turbine solidity can be defined by:

$$\sigma = \frac{nc}{2\pi r} \quad (2.34)$$

where n is number of blades, c is chord length and r is the turbine radius. By increasing the solidity, one can expect an increase in torque as well, yet a decrease in blade tip speed.

3

Methods

To obtain accurate data from CFD simulations of rotating objects is usually computationally expensive. It takes a lot of time, and therefore it is important to do a more narrow study on the parameters that affect the meshing and simulation time. The objective of this is to find a compromise between the meshing time and the mesh quality. First of all, the geometries that are necessary for this project are created in a CAD software. The existing model of the pilot turbine was created in SolidWorks 2017. All other turbine designs was also designed with SolidWorks 2017.

In the OpenFOAM 6, one have access to a tutorial case for a rotating propeller with *pimpleFoam* as solver. This case have a dynamic mesh and is used as a base for the rotating evaluations in this project. The same case was modified to a steady-state case with *simpleFoam* as solver. This was done to compare the simulation time and torque values for the different approaches.

The original geometry files in the tutorial case are replaced with the new .stl-files. In order for the simulation to work, all geometries are needed to be positioned carefully in the global space. From the schematic figure below, Figure 3.1, one can see that the system is axisymmetric and that it contains three cylinders, two stationary and one rotating. The turbine is positioned inside the rotating cylinder.

The original plan was to include the hub in the simulation, but after some discussion about its importance, it was excluded. This is because the main objective in this project is to find an optimized turbine design and not the hub design. The relative difference in turbine performance are assumed to be equal with or without the hub. In addition to this, the hubs function today is to have a stationary housing around the turbine. Without the hub, the turbine could not be placed in the casing and serve its purpose. The hub is not designed in a way to increase turbine performance. In this project it is nearly impossible to achieve exact results compared to the experimental data. The reasons for this will be discussed later on in Chapter 4.

The reason for this geometric set-up, is that during the meshing process, it is possible to choose the mesh quality in each region separately. For example in this case, the quality of the mesh incrementally increases as one get closer to the global origin (inside the turbine).

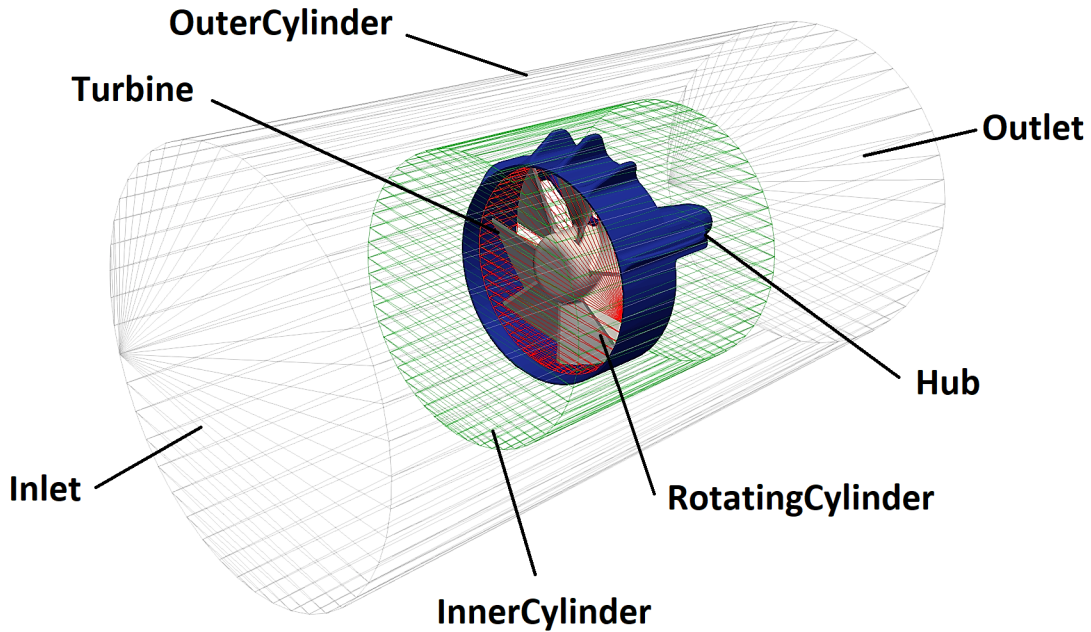


Figure 3.1: Schematic axisymmetric representation of geometries/patches.

3.1 Geometry

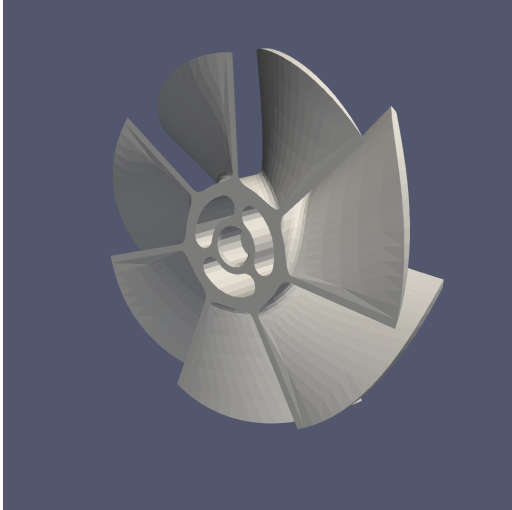
As mentioned earlier, the pilot design of the turbine is designed and provided by *Fishbones*. Yet, a few simplifications are made to the geometry (see Figure 3.2). These modifications are done to avoid unnecessary problems to occur during the meshing process.

Initially, all geometries are exported to .stl-files as ascii-format. By default the resolution of the .stl-files are set to "Fine" in SolidWorks 2017. A quick study on exportation settings was done to assure that good resolution of the .stl-files are achieved. In Figure 3.3, one can see how much smoother the blade surfaces are for the higher quality settings compared to the default settings. Note that these pictures are before any meshing of the turbine is done and are simply a -stl-file representation. The difference is due to the number of vertices and triangles each .stl-file contains. The file size in Figure 3.3a is 10.8 MB versus 2.5 MB in Figure 3.3b. A good quality mesh is essential for the crucial parts of the CFD simulation, i.e. the turbine, containing sharp edges.

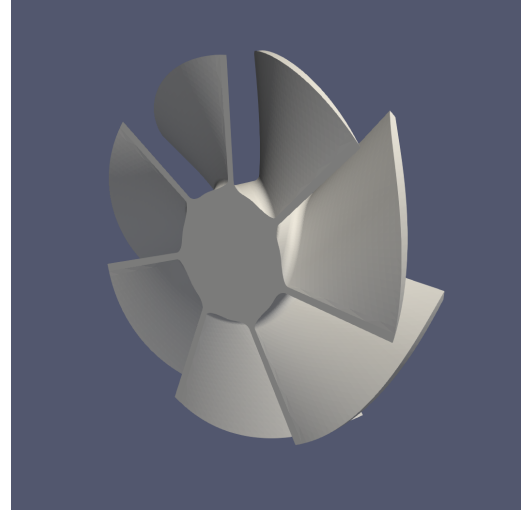
In Figure 3.2a, one can see some material is removed from the cap. The first simplification to the pilot design was to fill these holes, see Figure 3.2b.

Sharp edges at the turbine blades caused many problems in the early meshing phase in this project. Such sharp edges are often troublesome when using an automatic meshing tool like *snappyHexMesh*. Therefore, in order to avoid unnecessary trouble

for the meshing process, the most critical edges was smoothed by utilizing the *Fillet*-tool in SolidWorks 17. From Figure 3.4, one can see an example of very poor mesh quality at the edges and the blade surface.

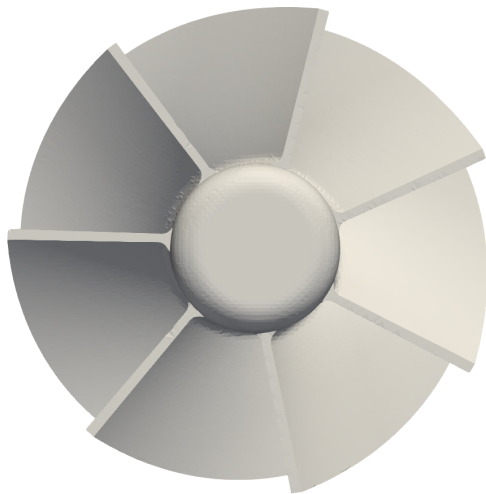


(a) Actual design of commercialized design (low quality).

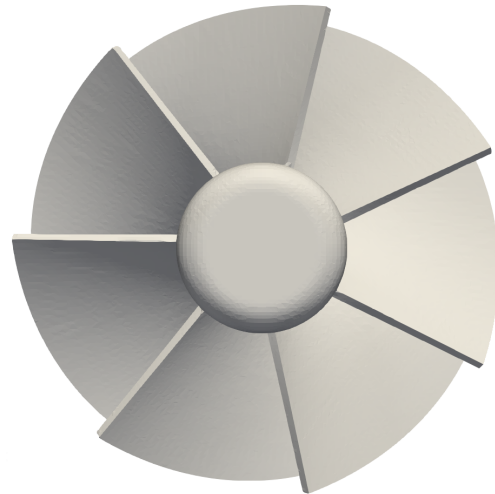


(b) Simplification to turbine design (higher quality).

Figure 3.2: Comparison of original design and simplified turbine design.



(a) Higher quality settings.



(b) Default settings.

Figure 3.3: Comparison of two different quality settings for .stl-files.

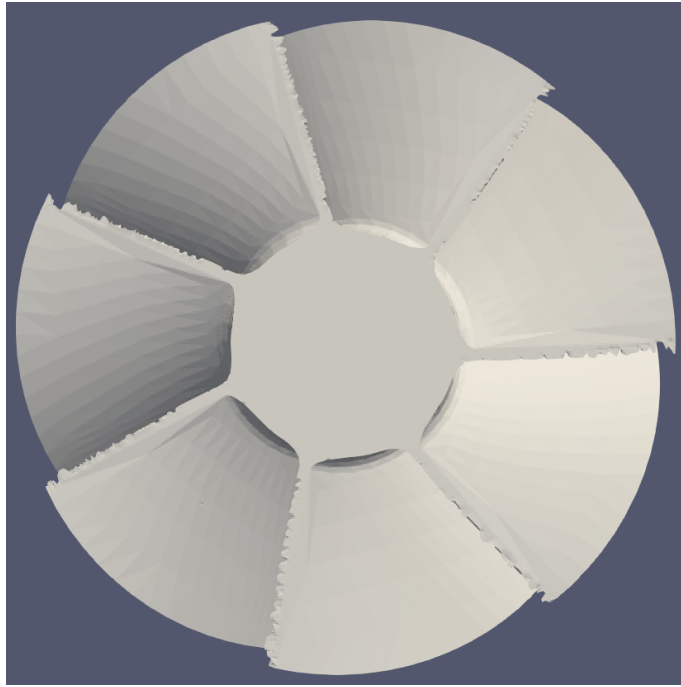


Figure 3.4: Bad mesh quality at the sharp edges and blade surfaces in the pilot turbine design.

3.2 Mesh Grid

The final mesh grid quality is determined and adjusted in the different regions in the system by the defined *-MeshDicts*. Initially, the global mesh grid quality is chosen in *blockMeshDict*. As briefly described earlier, *snappyHexMesh*, then refine edges and regions by further splitting the cells. An aspect ratio of 1, that is the ratio between the longest and shortest edge of a cell, was sought. This means that all sides of a cell is equally long. To achieve an aspect ratio of 1 in simple geometries is fairly easy, but might be rather difficult for cells near the turbine edges. The global mesh grid quality need to be chosen so that the domain is relatively equal in size. For example, in this project, the global domain size is x:80, y:80 and z:380, (80 80 380). To keep an aspect ratio of 1, the grid size can be chosen to be for example (40 40 190). This is often called uniform mesh. Nevertheless, OpenFOAM accepts higher aspect ratios to a certain level without causing critical errors to the mesh.

As seen in Figure 3.5, one can see how cells are the same at each end of the domain (quadratic). This is the coarsest areas in the domain and they have no additional refining from the *snappyHexMesh*. The number of cells incrementally increase moving closer to the turbine. The reason for this is that the flow will experience dramatic changes and become quite chaotic and turbulent when passing the turbine. A cross-section view of the turbine is shown in Figure 3.6 to illustrate how small the cells are inside the rotating zone. The inlet is positioned to the left hand side in Figure 3.5. The *innerCylinder* is extended behind the turbine. This is done to better keep the nature of the flow downstream after passing the turbine. More cells here will make sure of this. The entire domain is about $10d$, where d is

the diameter of the turbine.

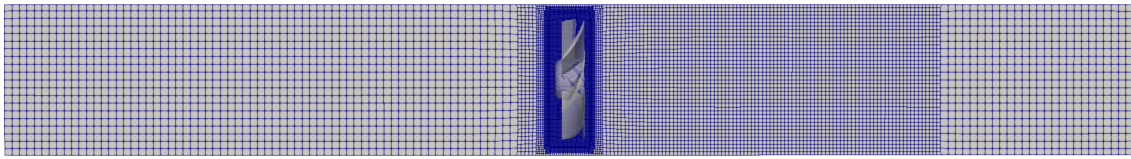


Figure 3.5: Entire domain of the system. Flow coming in from the left hand side.

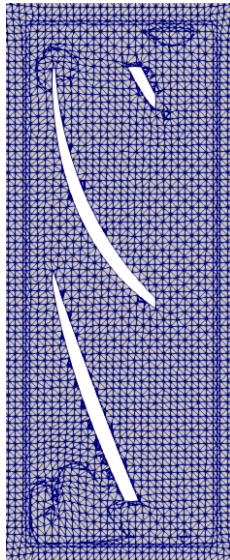


Figure 3.6: Refinement inside the rotating zone of the propeller.

The cell refinement done by *snappyHexMesh* inside the rotating zone is done in reference to the global mesh grid, defined in the *blockMeshDict*. Note that *snappyHexMesh* gives the opportunity to add layers to surfaces for detailed and complex geometries. This might not only help for the mesh quality, but improve the wall treatment on the turbine surface as well. However, this option was not used because results were considered acceptable.

In order to verify the case set-up, a mesh convergence study was done. In CFD, this is often done by keeping some parameters constant, while varying others, to see how each parameters effects the final results. Naturally, these parameters need to be relevant for the output and in this project. By keeping acceptable mesh quality parameters in the *snappyHexMeshDict* constant, multiple simulations with different global mesh grids were done. These parameters will be presented in Chapter 4. Comparing how the final torque deviated with different grid sizes was key to identifying how many global cells and faces that were necessary for a robust and converged solution.

3.2.1 Geometrical optimization

As mentioned earlier, there are some parameters that will be kept constant. Nevertheless, there are some parameters that can change and these are listed below:

- Inlet flow angle, β_{in}
- Stagger angle, θ
- Number of rotor blades, n
- Chord length, c

The existing CAD-file for the turbine was designed in a way that was difficult to change in a systematic manner. Simple changes to the curvature of the turbine blades created many errors and was nearly impossible to use for efficient optimization. Because of this, many hours were put into the creation of a much more detailed and easily modifiable CAD-model of the turbine. However, it was not possible to recreate the existing design 100% with the used method and therefore a quick research was done to check the actual deviation of the two models. It was decided to keep the root sketch parameters constant through the entire project. The main focus was therefore on the tip sketch and the parameters listed above. The pilot design have the following parameters shown in Table 3.1:

Table 3.1: Pilot design parameters.* is set constant during this project.

Root sketch*	
Inlet flow angle	34°
Outlet flow angle	43°
Stagger angle*	37°
Chord line length	28.17 mm
Tip sketch	
Inlet flow angle	46°
Outlet flow angle*	91°
Stagger angle	70°
Chord line length	65.79 mm
Solidity	1.3019

In order to extract as much energy as possible from the flow, a high solidity was sought. Designs with solidity under 1 were not evaluated as it was considered as non-efficient. While changing the stagger angle in the tip sketch, the chord line length changed accordingly. The stagger angle and chord length is directly connected to the turbines solidity in the CAD-model used. The number of blades evaluated in this project are 5-7 blades.

3.3 CFD Setup

3.3.1 Fluid conditions

The case set-up is copied from OpenFOAM 6 tutorials and therefore some modifications are needed to adapt to this projects requirements. The fluid is considered as incompressible water and therefore the density is changed to 1000 kg m^{-3} . The fluid is considered as a Newtonian Fluid and the kinematic viscosity, ν , is chosen to be $1 \times 10^{-6} \text{ m}^2\text{s}^{-1}$ according to standard tables [32].

These values are rounded off for simplified numbers. These quantities were used and kept constant throughout the entire project. The inlet velocities, shown in Table 3.2, were changed according to experimental data during the case set-up validation.

Table 3.2: Inlet flow velocities.

Experimental Volume Flow	Inlet velocity
613 l/min	1.275 m/s
704 l/min	1.464 m/s
801 l/min	1.666 m/s
908 l/min	1.889 m/s
1006 l/min	2.093 m/s
1109 l/min	2.307 m/s
1203 l/min	2.503 m/s
1313 l/min	2.731 m/s
1404 l/min	2.921 m/s

3.3.2 Rotational Zone

When doing simulations containing rotating geometry in OpenFOAM one need to define a rotational zone where the rotating geometry is to be positioned. All other geometries will remain stationary, see Figure 3.1. The turbine requires a very good mesh quality, otherwise there is a risk that some hexhedra cells might appear outside rotating cylinder causing errors in the dynamic calculations. The output results might also be very odd and deviate a lot from the expected values. This is what happened to Mehdipour [6].

Initially, the plan was to include the hub in the simulations, but because of limited time and discussion with the supervisors it was chosen to be excluded. In OpenFOAM one have to define rotational speed for the rotating zone as well as what axis it is to rotate about. This is defined in the *dynamicMeshDict* for the pimpleFoam case and in *MRFProperties* for the simpleFoam case. During the validation process, where recreation of experimental data was the objective. Both the inlet velocity and RPM were known parameters from measurements. At this stage a few problems arose. A simple change in the turbine blade design or any modification at all would affect the RPM values. In OpenFOAM, for AMI and MRF, the RPM value need to be defined for the case to run. When optimizing the turbine the RPM would turn into an

unknown parameter. However, it is possible to introduce a flow induced rotational case in OpenFOAM, but it is very complex and computationally. This method is a combination of AMI and 6DOF [21]. There are uncertainties about this method as well, for instance friction, but this was not researched further.

In order to create solutions to the different turbine concepts, multiple simulations with different RPM values were run. This was done to identify the torques behavior of each turbine design with different RPM. The operational volume flow for Fishbones Drilling is 1300-1400 l/min and therefore a constant volume flow was set. When the case set-up validation was done, all remaining simulations were run with 1313 l/min volume flow.

In theory, one can predict the RPM value in the RPM-Torque diagrams created, but in practice this value will most likely not be the correct value. Physical testing will therefore be necessary to validate the best design evaluated from the simulations.

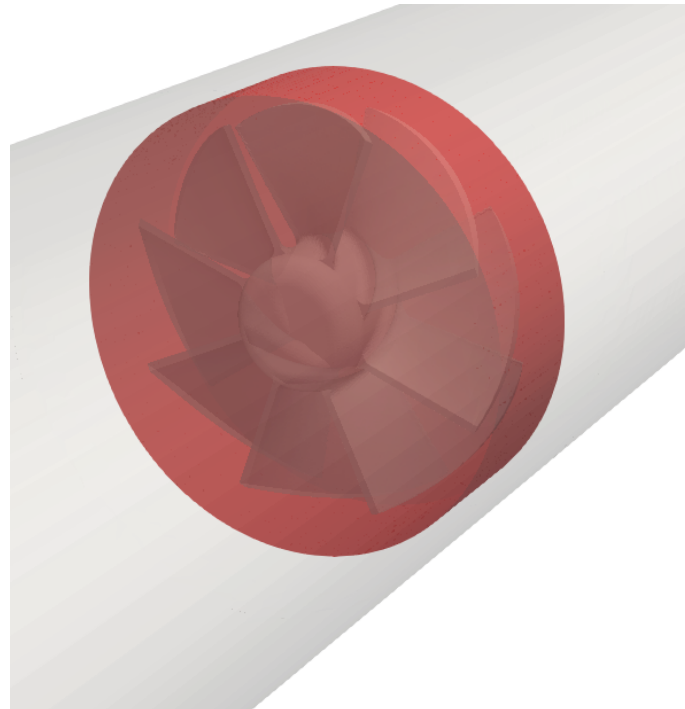


Figure 3.7: Rotational Zone.

3.3.3 Boundary conditions

Boundary conditions is something that is hard to avoid in CFD problems. Boundary conditions comes in many variants and it is quite important to understand their role in the simulation. Inlet and outlet conditions, physical boundaries, pressure conditions are a few examples. The inlet velocities from Section 3.3.1 is one example of how inlet velocity for the fluid is changed in this project. The boundary conditions in this project is listed below in Table 3.3.

Table 3.3: Boundary conditions used in this project for the AMI case.

Boundary conditions types					
	U	k	p	nut	epsilon
inlet	fixedValue	fixedValue	zeroGradient	calculated	fixedValue
outlet	inletOutlet	inletOutlet	fixedValue	calculated	inletOutlet
wall	noSlip	kqRWall-Function	zeroGradient	nutkWall-Function	epsilonWall-Function
turbine	movingWall-Velocity				

The inlet value for U is fixed and chosen according to Table 3.2. Throughout the entire domain the *internalField* is uniform. Uniform means that these values are constant for all cells in the mesh. At the outlet, the velocity should be equal to the inlet value. Based on the theory about turbulence boundaries the turbulence kinetic energy, k, the initial value is 0.012 and the *internalField* is uniform. The turbulence dissipation, ϵ is chosen to be 0.015 with a uniform *internalField*. The dynamic viscosity, nut, is uniform and calculated based on the behavior of k and ϵ . The pressure, p, is zeroGradient at inlet. This means that the pressure is developed through the domain and its gradient is equal to zero in the direction perpendicular to the patch. However, the pressure is a fixedValue at the outlet and is chosen to be uniformly 0.

All boundary conditions remains the same for the MRF case. The only difference is that for the turbine, the velocity boundary is changed to *noSlip*. This is because in MRF the turbine is treated as a wall and do not actually rotate.

3.3.4 Wall Functions

Mesh grading towards the wall is not necessary when using the standard $k-\epsilon$ model. The reason being that this model uses wall functions to model the flow behavior at the walls, rather than resolving it numerically [23]. The validation of these y^+ -values are presented in Chapter 4. In other words, during the meshing process there is no need for mesh layers because the surface of the turbine and the walls are modelled based on the theory from Section 2.3. As seen in Table 3.3, one can see what wall treatment each boundary condition use. The *noSlip*-condition for the velocity is an appropriate condition for walls. Because the velocity can be zero at the wall, there is no need for a pressure correction step here. Thus, the pressure is zeroGradient at the walls. The wall treatment for k, ϵ and nut, are mathematical models that describes the turbulent fluids behavior.

3.3.5 Solvers

OpenFOAM comes with many different solvers that are used to solve the discretized equations. As mentioned several times, the AMI-case is copied from the tutorial from OpenFOAM 6. For the MRF-case, the `fvSolution` is copied from the `mixerVessel2D`-case in tutorials. It is found in `tutorials/incompressible/simpleFoam/mixerVessel2D/system` in OpenFOAM.

The solvers in both cases are shown in the table below. No study were done for the different solvers available as the main focus was to set up a case that worked well enough.

GAMG (Generalized geometric-algebraic multi-grid) is the solver used to calculate the linear equations for pressure. In `fvSolution` one specifies what solvers, tolerances and relative tolerance etc. that each parameter (U, k, ϵ , p) should have, see A.1 for case structure. How long each iteration takes really depends on the matrix of equations that the solvers need to solve and the tolerances. The **tolerance** defines the required tolerance that the equations from the discretization needs to need to reach before a sufficiently accurate solution is acquired. The **tolerance** is set to be low for all parameters to ensure convergence, see Table 3.4 and 3.5. The **relTol** is the relative tolerance and is the ratio of current residual to the initial residual. One can also set a certain minimum or maximum iteration that the solver should reach before continuing. Despite of this, it was not used in this project.

For many of the available solvers in OpenFOAM there is a need to define what *smoother* the solver is going to use. The smoother is used to avoid getting caught up with errors, and the smoother simply smooths the iterations of the solver. The pressure, uses the GAMG solver, and therefore one need to define a smoother.

`DICGaussSeidel` and `GaussSeidel` are smoothers used in this project.

`DICGaussSeidel` smoother is used for the AMI-case because it generates a fast solution by generating a quick solution with a small number of cells, then it maps this solution onto the finer mesh. U, k and ϵ use a *smoothSolver* with `symGaussSeidel` and `GaussSeidel` as smoother. These two solvers use the same iteration-method, but `symGaussSeidel` is for symmetric matrices. A technique used to guarantee a diagonal equality of the matrix is to use

under-relaxation. It is used to improve stability of the solvers. In this project the **relaxationFactor** for velocity, turbulence dissipation and kinetic energy is $\alpha = 1$ in the AMI-case. This guarantee matrix diagonal equality [23]. This is necessary due to the smoother used for U, k and ϵ . However, it is not necessary for the MRF-case as the matrix can be asymmetric as well. The **relaxationFactor** for U, k and ϵ is 0.5 and 0.3 for pressure in the MRF-case.

The solver applications used in this project was `pimpleFoam` and `simpleFoam`.

The `pimpleFoam` solver uses an algorithm that combines the PISO and SIMPLE algorithms. All three algorithms are iterative procedures for coupling equations for momentum and mass conservation [23]. The PIMPLE algorithm allows to solve transient problems. Parameters like **nCorrectors**,

nOuterCorrectors, **nNonOrthogonalCorrectors** and **momentumPredictor** are used to affect how the looping of the algorithms are. In the AMI-case the **nCorrector** is set to 2. This means that the algorithms solves the pressure and momentum equation twice in each iteration step.

Table 3.4: Solvers used in this project for the AMI-case.

AMI solvers		
	U k epsilon	p
solver	smoothSolver	GAMG
tolerance	1e-6	1e-5
relTol	0.1	0.1
smoother	symGaussSeidel	DICGaussSeidel
nSweeps	1	1
relaxationFactor	1	-

Table 3.5: Solvers used in this project for the MRF-case.

MRF solvers		
	U k epsilon	p
solver	smoothSolver	GAMG
tolerance	1e-7	1e-8
relTol	0.1	0.1
smoother	GaussSeidel	GaussSeidel
nSweeps	2	-
relaxationFactor	0.5	0.3

nNonOrthogonalCorrectors is set to 1 and updates the explicit non-orthogonal correction. What this means will be explained further in the Schemes-section below. Finally, the last parameter used is **nOuterCorrectors**. It is set to 1 and it enables looping over all equations for each time step [23]. There is no correctors used in the MRF-case for the solver application.

3.3.6 Schemes

In *fvSchemes*, the methods for numerical discretization is selected. Again, there are multiple different schemes available, It is important to choose these schemes correctly or else it might cause numerical error in the simulation. The schemes used in this project for both cases are listed in Table 3.6 shown below:

Table 3.6: Discretisation schemes used in this project.

	AMI / MRF
Time Schemes	default <i>Euler</i> / default <i>steadyState</i>
Gradient Schemes	default <i>Gauss linear</i> <i>grad(p)</i> <i>Gauss linear</i> <i>grad(U)</i> <i>cellLimited Gauss linear 1</i>
Divergence Schemes	<i>div(phi, U)</i> <i>Gauss upwind</i> <i>div(phi, k)</i> <i>Gauss upwind</i> <i>div(phi, epsilon)</i> <i>Gauss upwind</i> <i>div((nuEff*dev2(T(grad(U)))))</i> <i>Gauss linear</i>
Laplacian Schemes	default <i>Gauss linear limited corrected 0.33</i>
Interpolation Schemes	default <i>linear</i>
snGrad Schemes	default <i>limited corrected 0.33</i>

It is very important that the discretisation scheme for time is chosen correctly. For the AMI-case the scheme, *Euler*, is chosen. This is because it is transient. If a steady-state scheme was used, with a dynamic rotating geometry, the solvers would not achieve to serve their purpose. For the MRF-case the scheme for time is *steadyState*.

The most used gradient scheme in OpenFOAM is *Gauss linear*. As you can see in Table 3.6 it is also the default scheme. It is a standard finite volume discretisation of Gaussian integration. It interpolates, linearly, the values from the cell centres to face centres through the domain. For the velocity gradient it uses a *cellLimited* scheme. The difference is that the cell values are in this case extrapolated to the faces by the use of the calculated gradient. The limiting coefficient, 1, guarantee boundedness. This scheme is used for the velocity gradient due to the risk of poor mesh quality of the turbine. It was important to have a case that could run even with a poor mesh quality.

The *Divergence Schemes* for U , k and ϵ is all Gauss upwind and the effective viscosity is Gauss linear. The *phi*, is the flux of velocity of the flow. These schemes are kept unchanged from the tutorial case as the first order stability and robustness of the simulation was considered good enough for this project.

The *Laplacian scheme* is the diffusion term in the momentum equations. OpenFOAM do not have as many options for this scheme compared to the divergence schemes. They only offer Gauss linear schemes. Nevertheless, there are several different variants available. The chosen scheme is *Gauss linear limited corrected 0.33*.

The *snGrad Schemes*, Surface normal gradient schemes, is closely related to the *Laplacian Schemes* because the *snGrad Schemes* are required to evaluate the *Laplacian Schemes*. To avoid orthogonality errors in the solvers a second-order accuracy, explicit non-orthogonal correction is done. Whenever the neighbouring cells in the mesh is non-orthogonal, an iterative correction process lead the angle between each cell towards 90° . The limited correction is chosen to be 0.33 because this avoids problems to occur. Without the limited correction one can experience problems due to too high corrections. The limited correction gives a great stability and also keeps a certainty of accuracy as well. Last, but not least, the default linear *Interpolation Scheme* is chosen. From the available tutorials in OpenFOAM, almost all cases use this scheme by default. Note that different schemes than the ones listed above were not evaluated in this project.

3.4 Post Processing

As stated in Section 2.2, MRF and AMI share the same principles, but are still different. The MRF-approach is a steady state method. In order to evaluate whether the solution has converged or not is based on number of iterations and not time, as it is for the AMI-approach. The residuals of quantities like pressure, velocity and torque are all evaluated to make sure the final results are stable. In Figure 3.9, the torque converges after approximately 600 iterations. And the residuals for pressure and velocities are considered acceptably low after 2000 iterations. As discussed later on in the Chapter 4, the AMI-approach was eventually dropped after identifying how small the deviation of torque outputs was compared to the MRF-approach. The AMI-approach was too computationally expensive to use for such a project, see Section 4.2.

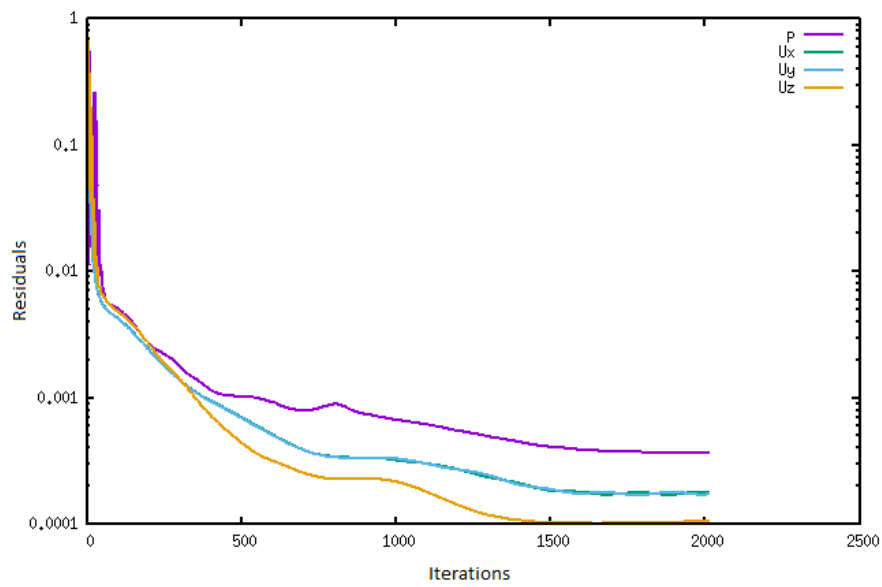


Figure 3.8: Residuals from MRF-simulation with mesh grid (64 64 304) for pressure and velocities.

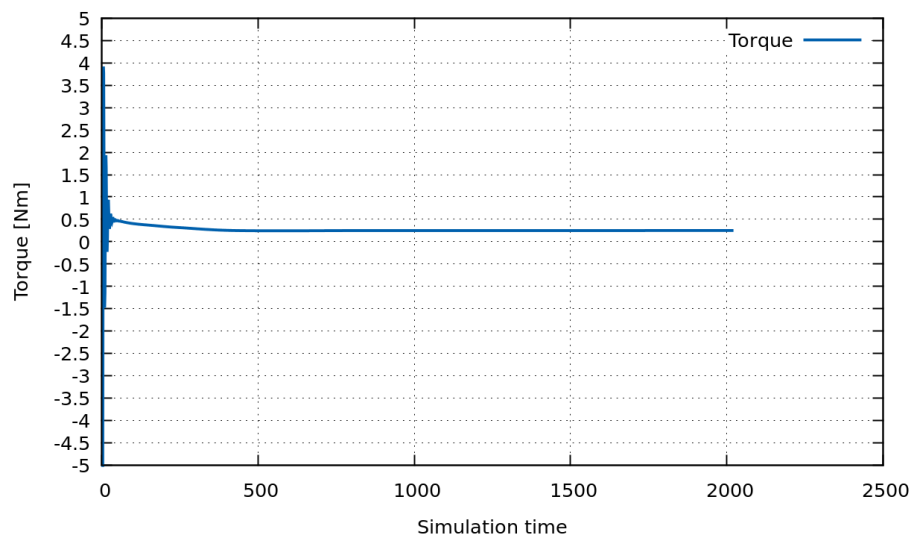


Figure 3.9: Convergence of torque for mesh grid (64 64 304) for MRF-simulation.

4

Case Validation

The main focus from the beginning of this project was to create a case set-up where different turbine designs could easily be changed and still run successfully without other changes. All simulations are an idealized set-up compared to the experimental set-up. Factors like the hub, hub/turbine angle in the casing and friction are all ignored. In addition to that, how this idealization affects the output results are not studied. The experimental data, gathered at Fishbones' office, is therefore to deviate from the simulated values. Frictional losses in the rotation is quite high due to the bent configuration the string have inside the hollow tube shown in Figure 1.4 and 4.1. In the simulation, the inflow fluid attack the turbine perpendicularly, but for the collected experimental data this is not the case. The hub/turbine lies in the casing with an angle. Thus, the inflow is not axisymmetric with the turbine center. In practice this will most likely create a more unstable moment towards the turbine blades and reduced efficiency due to higher static pressure drop. However, there are no test data to support this assumption.

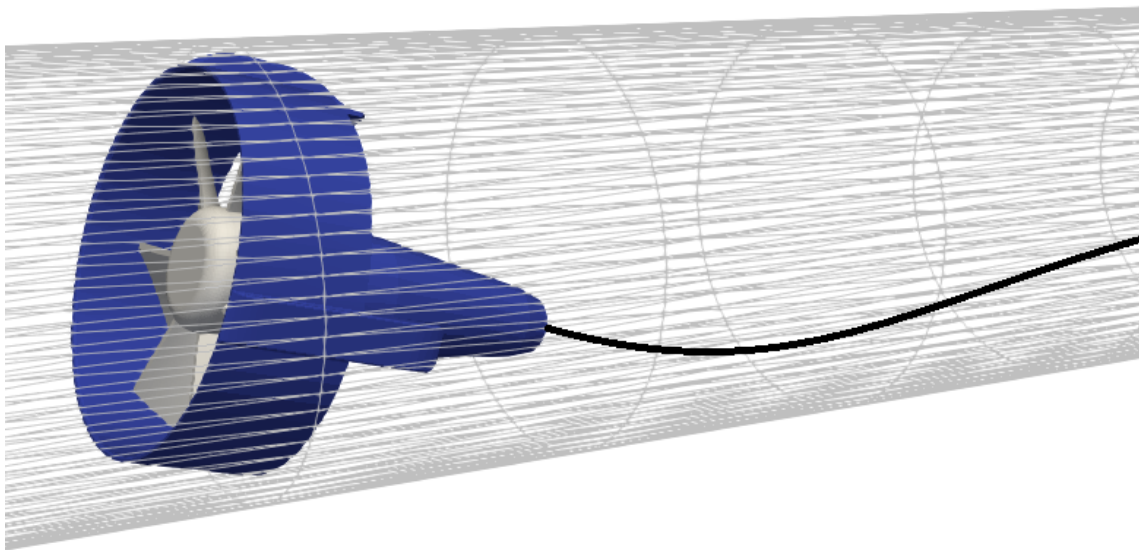


Figure 4.1: Schematic illustration of actual hub/turbine positioning inside the casing from the experimental set-up.

However, some unexpected problems was met in the late simulation process. During the validation of the experimental data shown in Table 4.1, the torque output from the simulations were discovered to be unexpectedly low, see Table 4.1 and Figure 4.5. The simulated values were over 500% off compared to the experimental values.

4. Case Validation

The simulated values were expected to be higher than the experimental data due to the idealization. Nevertheless, although these values were quite off and no solution was found to solve the issue, the data could still be used to see how different turbine designs behave under the same conditions in the same simulation set-up. The relative deviation of outputs to the commercialized pilot design, and experimental data, is therefore assumed to be the same for all optimized turbine designs. The pattern of Figure 4.2 will be evaluated and checked for the chosen final optimized design.

Table 4.1: Experimental data gathered in Fishbones' lab.

Experimental data		
Volume Flow	RPM of drillbit	Torque
613 l/min	775	0.22 Nm
704 l/min	922	0.36 Nm
801 l/min	1126	0.45 Nm
908 l/min	1334	0.64 Nm
1006 l/min	1497	0.73 Nm
1109 l/min	1687	0.98 Nm
1203 l/min	1827	1.09 Nm
1313 l/min	1927	1.22 Nm
1404 l/min	1937	1.35 Nm

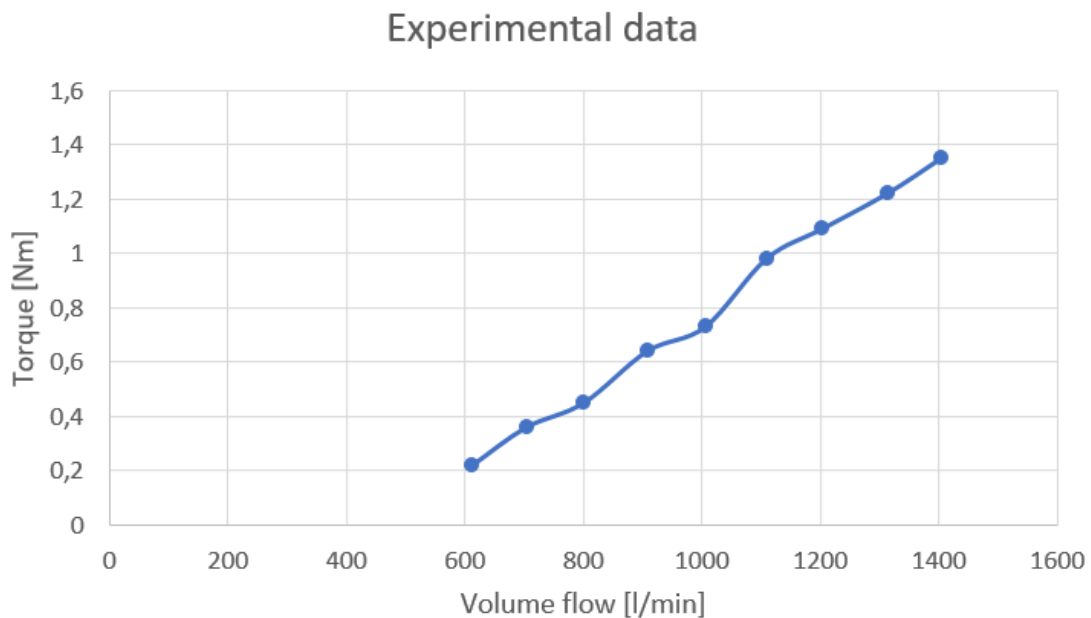


Figure 4.2: Torque vs Volume flow.

4.1 Mesh Quality

From a study on error analysis of finite volume methods in fluid dynamics [24] a few mesh-induced errors are described in depth. One of these is *Non-orthogonality*. This is also the first error defined in *snappyHexMeshDict*. Non-orthogonal cells are a significant contributor to error in OpenFOAM. The decision of whether a cell is non-orthogonal or not is defined by an angle in *snappyHexMeshDict*. A general thumb rule is that when the angle between a cell-center vector and face normal is larger than 70° one will experience a non-orthogonality. In the *snappyHexMeshDict* this angle is in most cases set to 65° to avoid the non-orthogonal cells [25].

The next challenge is the **Skewness**. During the mesh convergence study, the skewness was the only mesh error detected. The way that OpenFOAM calculates the fluxes between adjacent cells in the domain introduce this error to occur. It is a quantity that is evaluated by the distance between a face center and the interpolated face center divided by the distance between the adjacent cell centers. It basically means that the skewness is the difference between the shape of two equilateral cells with the same volume [26]. The number of skewed faces changed for the different mesh qualities studied during the mesh convergence of the pilot design. However, these numbers are not constant for all future design concepts in this project. The errors are expected to change for all concepts and mesh grid quality. The number of skew faces of the pilot design will be presented in next chapter for different mesh grid qualities.

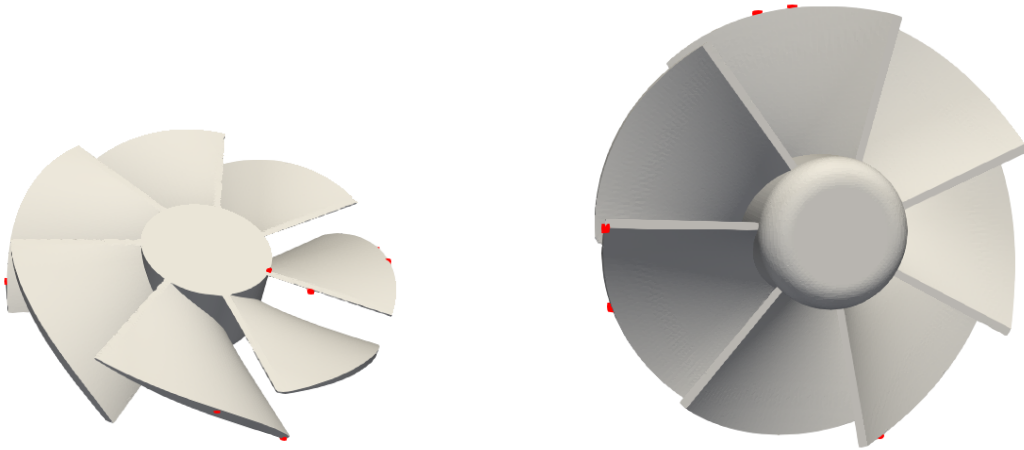


Figure 4.3: The highly skew faces was very small and therefore visualized by spheres with define-able radius instead of faces. Mesh grid size is (40 40 190).

As seen in Figure 4.3, the highly skewed faces are positioned at the blade edges. This is a typical area where the difference between two adjacent cells large because of the sharp edges.

Another error that may occur is the size of face area and cell volume. Whenever a face becomes very small it may cause problems in the meshing process. The **minVol** and **minArea** quality controls are both evaluated as "OK" for all design concepts in this project. In order to understand how small each face are in the domain, imagine the turbine with a diameter of 80.4mm. The maximum face area for the chosen mesh grid is $3.815e-5 m^2$ and the minimum face are is $1.55e-9 m^2$. These numbers are evaluated from the entire domain. The face areas at the turbine are expected to be much lower. In OpenFOAM, there are some quality controls that are more critical than others. It is possible to run a case successfully even though *checkMesh* says that the mesh is not OK for the following quality controls: *skewness*, *aspect ratio* and *minimum face area* [27]. As stated above, the only "not OK" message received in this project was the skewness. Thus, all cases could run successfully. Nevertheless, these errors might affect the total simulation time and if the errors are too vital, it may kill the solving process. This was not experienced in this project.

4.2 Mesh Convergence study

In order to find the best compromise between simulation time and output results, a broad mesh grid convergence study was done. By varying the global mesh size in *blockMeshDict* parameters like pressure, fluid velocity, torque and thrust were evaluated. The different parameters in *snappyHexMeshDict* was kept constant during this study. These values are shown in Table 4.2. All other parameters in the *snappyHexMeshDict* are unchanged and are equal to the tutorial case "*propeller*" located in

tutorials/incompressible/pimpleFoam/RAS in OpenFOAM 6.

To determine when the torque converged, *gnuplot* was utilized. Gnuplot is a terminal-line driven tool used to graph a chosen range of data. In steady-state cases, convergence is found by iterations. By running 2000 iterations with *simpleFoam*, converged and stable values for torque, pressure and velocity were found. In the unsteady transient case, the convergence is time dependent. Thus, another approach is needed to find the converged values. *endTime*, defined in the *controlDict* in the system folder, determines when a simulation ends. This needs to be large enough to make sure of convergence.

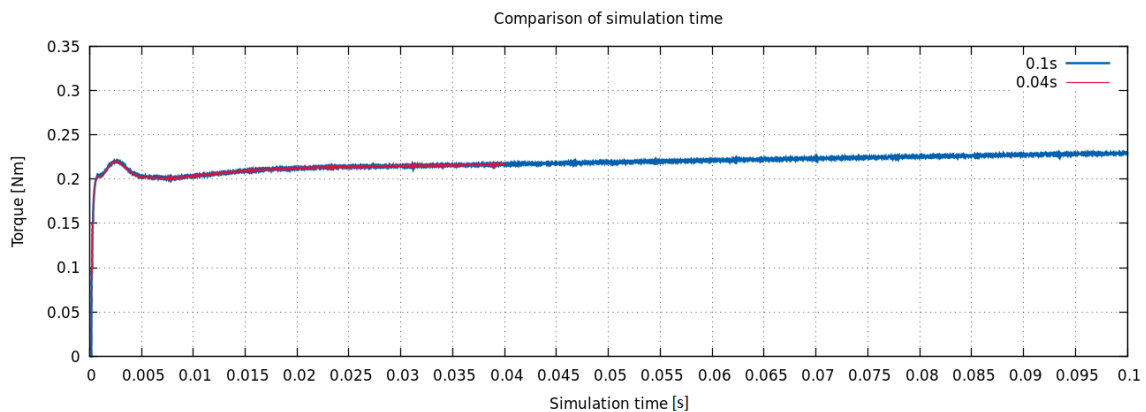


Figure 4.4: Only 6% deviation from simulation time of 0.04s and 0.1s of AMI-case.

From Figure 4.4, one can see how the torque changes with time for the transient case. The change from 0.04s to 0.1s is relatively small and is only 6%. The simulation time for a global mesh grid size of (32 32 152) is 9 hours for 0.04s and 21 hours for the 0.1s simulation. During the comparison study of AMI and MRF, the remaining AMI-simulations were therefore done only for 0.4s.

Table 4.2: snappyHexMeshDict parameters changed compared to the tutorial case.

Parameter	Input
MaxLocalCells	100 000
MaxGlobalCells	2 000 000
nCellsBetweenLevels	5
tolerance	2
nSolveIter	100
refinementSurfaces	
rotCylinder	(3 3)
outerCylinder	(0 0)
turbine	(3 3)
refinementRegions	
rotCylinder	(1E15 3)
outerCylinder	(1E15 1)

Table 4.3: Simulation data from mesh convergence study of pilot design using simpleFoam - MRF.

Mesh study simpleFoam - MRF				
Mesh grid	Total cells	Total faces	Mesh Error	Time
(10 10 48)	53 862	170 279	5 skewed	5 min
(16 16 76)	179 475	560 705	6 skewed	2 min
(20 20 95)	343 820	1 066 802	4 skewed	4 min
(32 32 152)	1 172 669	3 612 603	0 skewed	11 min
(40 40 190)	1 979 132	6 094 148	7 skewed	27 min
(50 50 238)	2 937 514	9 131 995	44 skewed	43 min
(60 60 285)	4 312 849	13 416 872	7 skewed	42 min
(64 64 304)	4 986 503	15 520 899	6 skewed	51 min
(80 80 380)	2 462 283	7 467 266	4 skewed	25 min

In Table 4.3, the output values are listed for the pilot design. The size of the mesh grid is chosen according to the size of domain size (80 80 380). It is quite interesting to see how the torque behaves when the mesh grid size is getting close to the domain size. As the mesh grid increase, so are the total cells and faces in the system. When the global mesh grid is equal to the domain size, the number of cells and torque values are reduced. In Figure 4.5, this is illustrated. As stated above in Table 4.2, the *MaxGlobalCells* is set constant at a value of 2 000 000. However, this value is an approximate value and deviations might occur. As the mesh grid increase to about 3 million cells, the torque output reaches a maximum. At this point, the mesh starts to fail and number of mesh errors starts to increase. This is what happened to the mesh grid (80 80 380). The mesh refinements in *snappyHexMeshDict* are not able to do their job when *MaxGlobalCells* are reached. This means that the mesh quality might become bad. However, the global mesh grid size is large and thus a decent mesh quality is accomplished without much refining anyway.

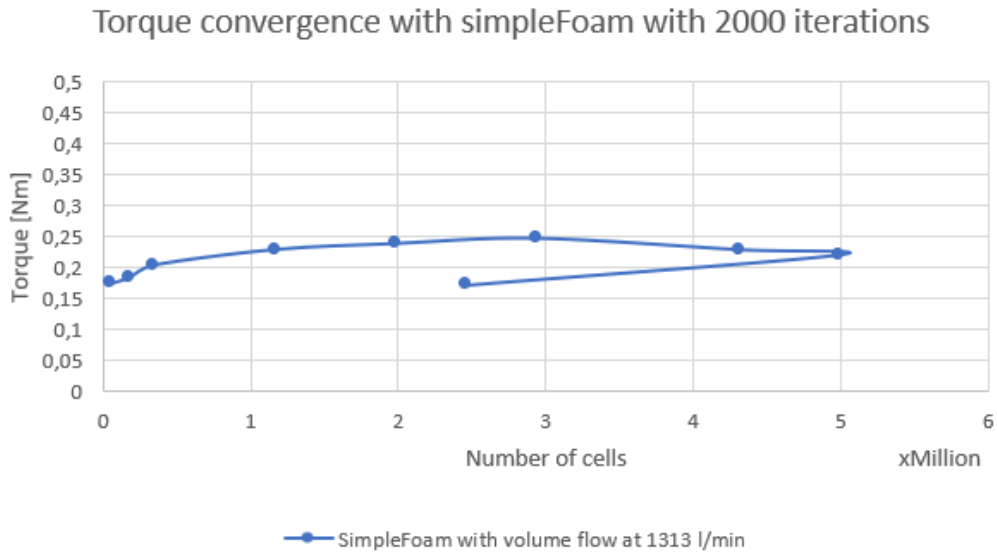


Figure 4.5: Torque vs number of cells.

Table 4.4: Simulation data from mesh convergence study of pilot design using pimpleFoam - AMI.

Mesh study pimpleFoam - AMI				
Mesh grid	Total cells	Total faces	Mesh Error	Time
(10 10 48)	53 862	170 279	5 skewed	6 min
(16 16 76)	179 475	560 705	6 skewed	53 min
(20 20 95)	343 820	1 066 802	4 skewed	3h 4min
(32 32 152)	1 172 669	3 612 603	0 skewed	9h
(40 40 190)	1 979 132	6 094 148	7 skewed	26h 46min
(50 50 238)	2 937 514	9 131 995	44 skewed	34h 12min
(60 60 285)	-	-	-	-
(64 64 304)	-	-	-	-
(80 80 380)	-	-	-	-

The chosen optimum mesh grid for the pilot commercialized design is chosen to be (32 32 152). This mesh grid gives the best compromise between a stable torque values, mesh quality and simulation time. During the entire project, all concepts were run with this mesh grid. In Table 4.4, a clear simulation time trend are shown. As the mesh grid increases the simulation time becomes very large. Have in mind that all these results is based on simulations run on a cluster with 20 cores. Thus, the computational power is fairly large. For the remaining results in Table 4.4, it was decided to skip running those cases due to lack of time. Nevertheless, the pattern is quite strong. When comparing the torque values from the AMI-case and the MRF-case the torque-outputs are relatively close to each other, see Figure 4.7. For the optimum mesh grid the difference in torque is only $\approx 6\%$ for the pilot design. From this evaluation, all new turbine designs are only run with MRF-simpleFoam because it is much faster and the deviations from the transient case are considered

4. Case Validation

acceptable.

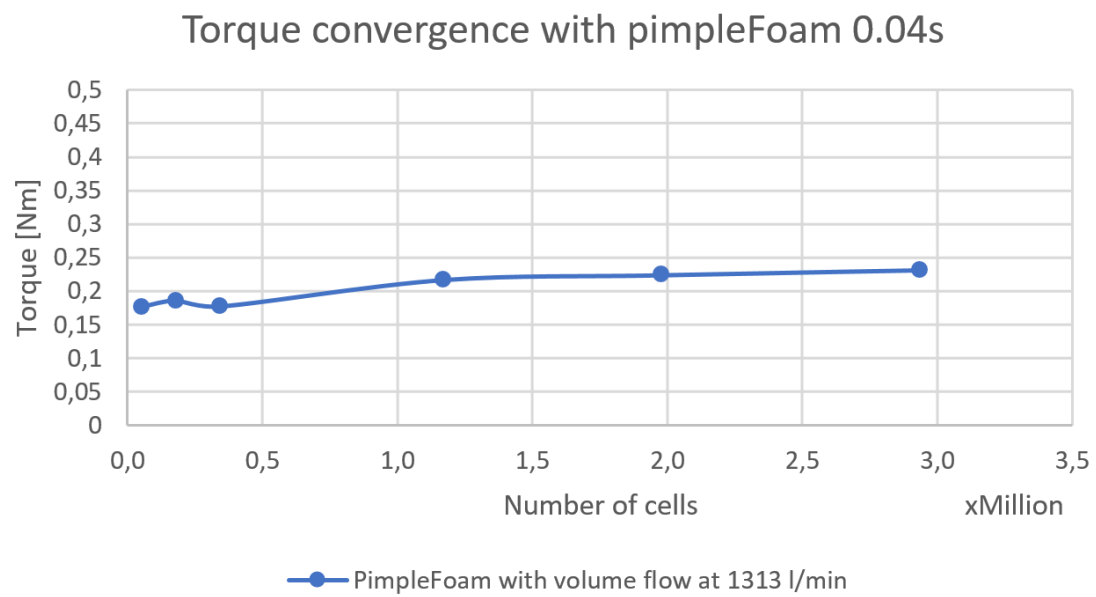


Figure 4.6: Mesh convergence of torque versus number of cells for the AMI-case.

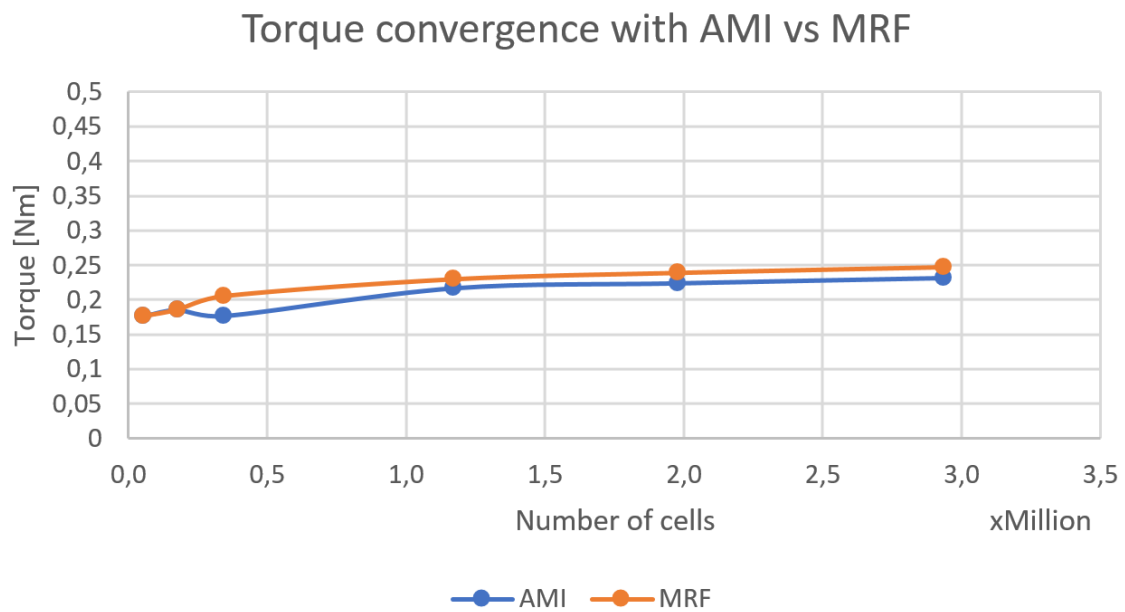


Figure 4.7: Comparison of the AMI-case and MRF-case of the chosen mesh grid resolution for the pilot design.

4.3 New vs old CAD-model

As mentioned in Section 3.2.1, it was not possible to recreate the CAD-model for the already commercialized design 100 % in a robust way. The CAD-model of the old design was created in a simple and easy way. In the old model, it was used a "Loft"-function between two enclosed sketches in two different planes to create the blades. In order to document what changes that is done to the design, one need to have a clear idea about which parameters that have been changed. This was not possible in the old CAD-model. No parametric dimensions were present in this model. The new CAD-model opens a much more efficient way to optimize and change parameters of the blade. A function called "Thicken" was used to create a solid of the pressure surface curve. Thus, an uniform thickness of the blade was created. Even though the pressure surface curve are almost identical, the old model have a different suction surface curve. As seen in Figure 4.9, the blue geometry is the new CAD-model and the white geometry is the old CAD-model. The curvature of the new model is defined by stagger angle, inlet flow angle and outlet flow angle. The outlet flow angle, β_{out} will remain constant in this project as stated in Section 3.2.1. In the figures and tables below, the largest deviations of the new and old model are illustrated.

As mentioned in the previous section, the number of mesh errors will change for all designs and mesh qualities. From now on, the new CAD-model will be referred to as the "pilot design". A mesh convergence comparison between the models are shown in Figure 4.10.

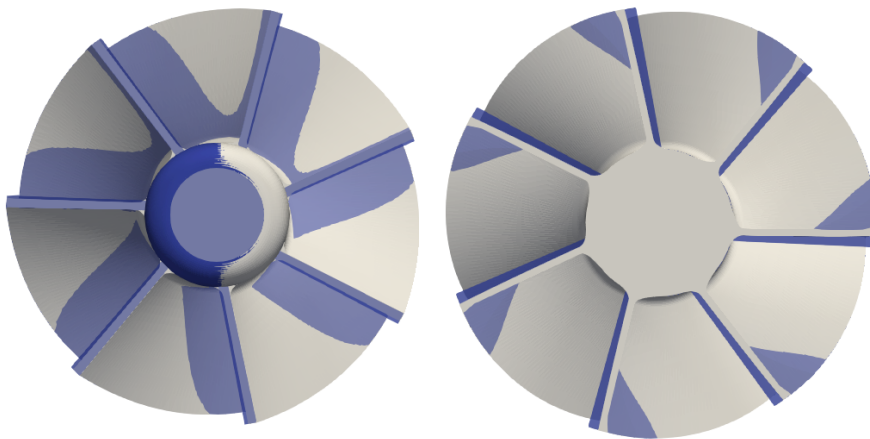


Figure 4.8: Comparison between old (white) and new CAD-model (blue).

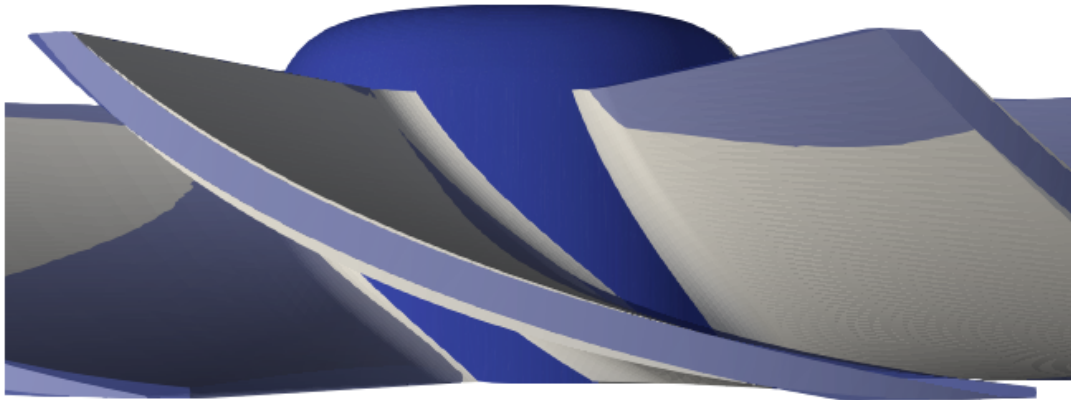


Figure 4.9: Comparison between old (white) and new CAD-model (blue).

Table 4.5: Comparison between the old and new CAD-model for mesh grid (32 32 152) at 1927 RPM and 1313 l/min flow rate.

	Torque [Nm]	Thrust [N]	Pressure drop [Pa]	Efficiency [%]
Old model	0.2457	61.06	5016	45.2
New model	0.2418	78.08	6017	37.1
Deviation	-0.0039	+17.02	+1001	-8.1
Deviation %	-1.61 %	+21.8%	+16.63%	-8.1%

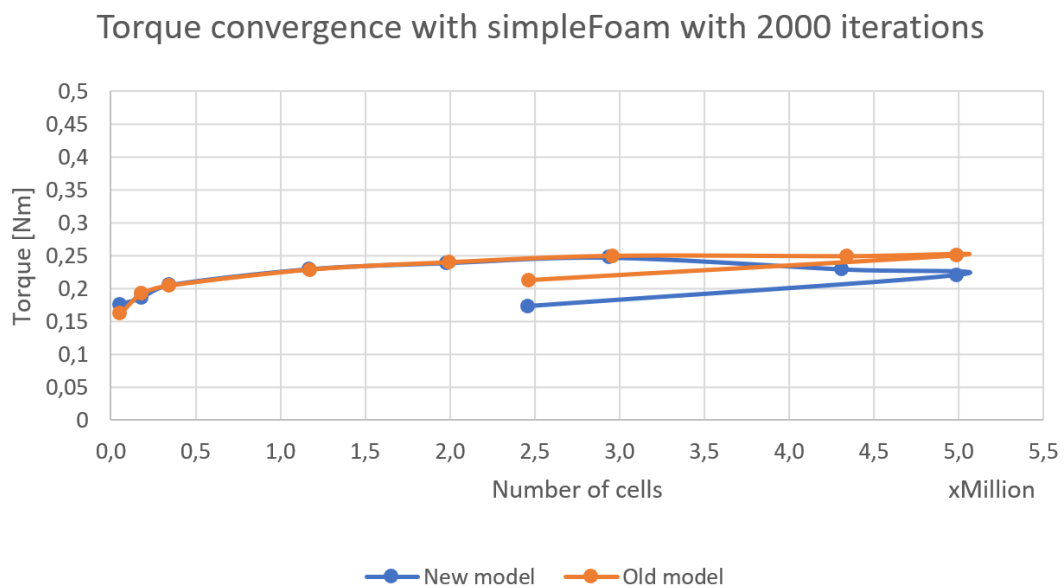


Figure 4.10: Comparison of torque convergence of new and old CAD-model.

4.3.1 Boundary layer resolution

In this project, a detailed boundary layer resolution study was not done. Nevertheless, the values of y^+ were evaluated for the chosen compromised mesh grid resolution of the old and new CAD-model. As discussed in Section 2.3, the log-law layer is said to be effective in the y^+ -range between 30 and 500 [8]. Thus, a check whether the average y^+ -values were within this range was done. By utilizing the $yPlus$ -function that is available in OpenFOAM, the minimum, maximum and average values for the walls (outerCylinder and turbine) were identified.

Table 4.6: Average y^+ -values for the new and old CAD-model for mesh grid (32 32 152).

y^+ -values for old model.			
Patch	min.	Average	max.
Turbine	12.6264	56.3329	978.334
outerCylinder	15.133	152.686	515.01
y^+ -values for new model.			
Patch	min.	Average	max.
Turbine	9.5072	57.1282	257.386
outerCylinder	15.1597	160.5	652.593

In Figures 4.11 and 4.12, one can clearly see the areas where the log-law layer is valid and not. The red areas indicate a high y^+ -value and all the other areas are acceptable. The red areas are the maximum values from Table 4.6 for the old model. These values are neglected as the average value was considered. From Table 4.6, one can see that the average y^+ -values are within the log-law layer for both the old and the new model.

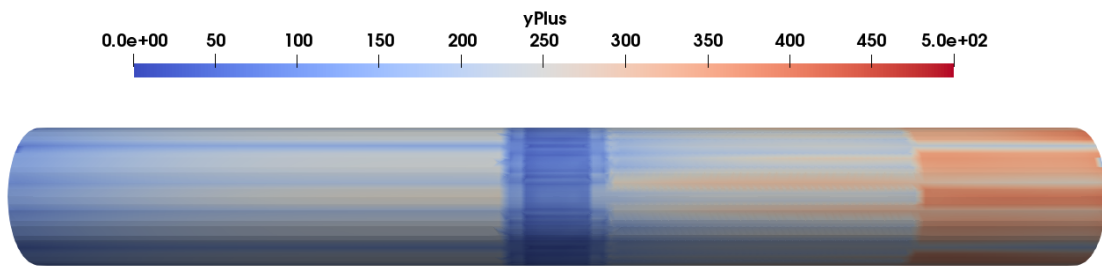


Figure 4.11: y^+ -values of the system from MRF-simulation.

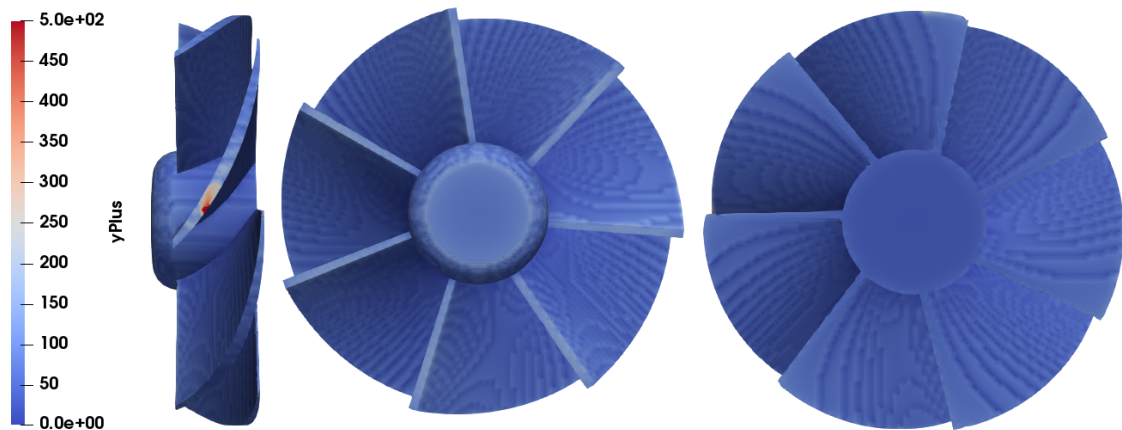


Figure 4.12: y^+ -values of turbine from MRF-simulation old model.

4.4 Flow fields

In order to evaluate the efficiency of the turbine, it is necessary to study the pressure fields in the system. The pressure drop is evaluated by creating a cellSet in front of the turbine and a cellSet behind it. Then, the pressure drop is evaluated by averaging the static pressure field between the chosen cellSet positions. These two cellSets together creates a cylindric cellZone. This cellZone is 20 mm long with the turbine positioned in the middle. At a volume flow of 1313 l/min, the pressure drop of the new CAD-model is measured to be 6017 Pa (see Table 4.5) for the optimum mesh grid.

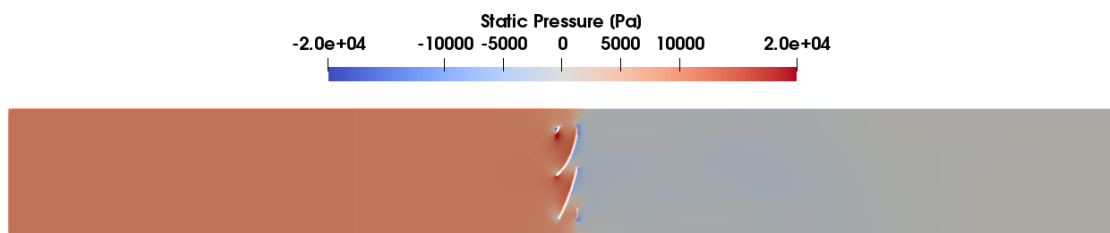


Figure 4.13: Pressure field of the entire domain over 2000 iterations of the pilot design.

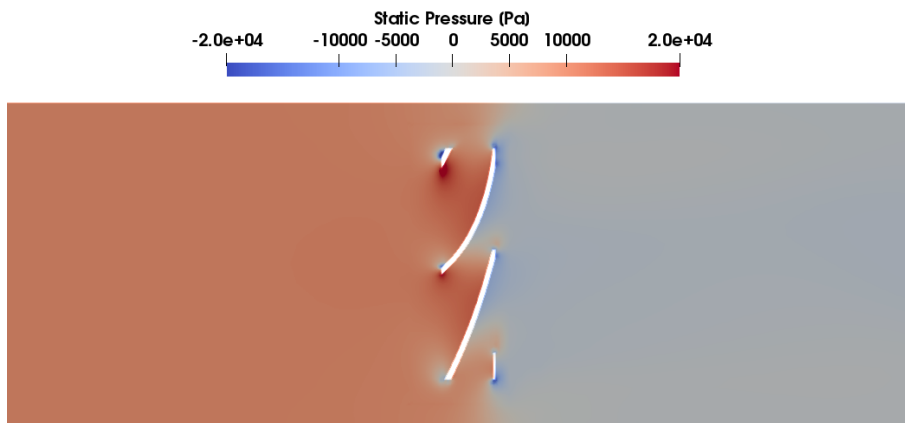


Figure 4.14: Pressure field of the cellZone over 2000 iterations of the pilot design.

When looking at the velocity field, it gives a quite accurate understanding about Bernoulli's equation. At the suction side of the blades, the velocity is much higher than on the pressure side. This can be observed in Figures 4.15 and 4.17. In Figure 4.16, it can be observed that some of the velocity vectors goes through the turbine blade. The figure is created by utilizing the *Slice*-tool in *ParaView*. The slice is done a few millimeter from the center of the turbine. Thus, particles that flow around the turbine can not be visualized in this plane. These vector arrows might be a ParaView visualization error. From the contour of the velocity field in Figure 4.16, one can see red zones at the suction side of the blades. However, there are minimal amount of velocity vectors present. The size of the velocity vectors are directly connected to the fluid velocity. Why there are no large velocity vectors behind the upper turbine blade in Figure 4.16 are quite strange, but again, it might be a ParaView visualization bug.

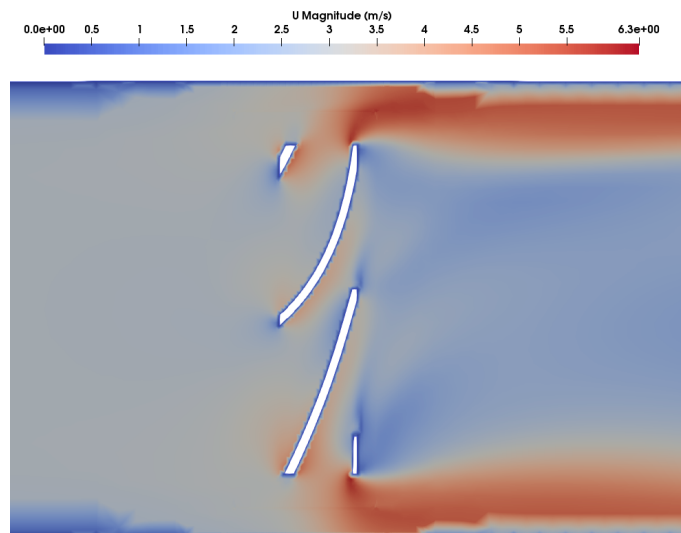


Figure 4.15: Averaged velocity field over the turbine over 2000 iterations of the pilot design.

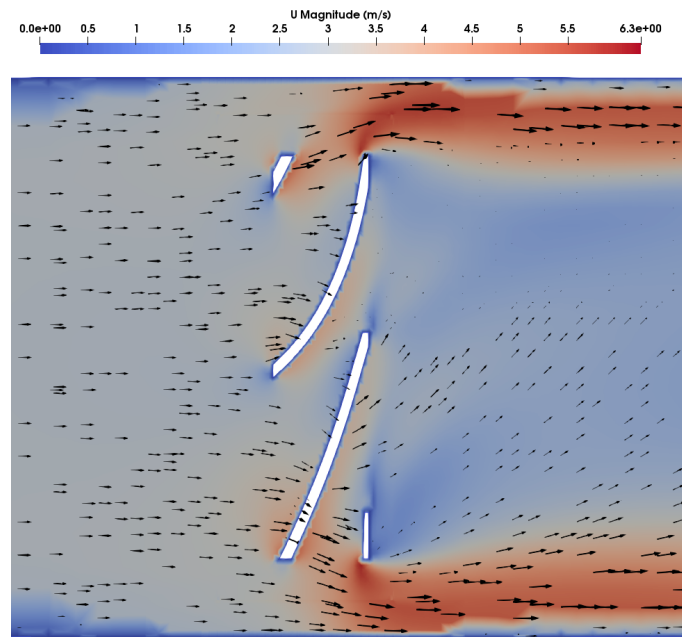


Figure 4.16: Averaged velocity field over the turbine with velocity vectors over 2000 iterations of the pilot design.

An other thing to observe in Figure 4.16, is that the water does not flow through the turbine as much as it can be expected. Due to the high RPM of the turbine it might force the water to flow around the turbine. At the blade tips of the cross-section, one can see that the water velocity accelerates a lot and that the average velocity is 6.3 m/s at the highest. In Figure 4.17 one can see the swirls created by the turbine in the downstream picture. This pattern is as expected. In Figure 4.18, one can see how the swirl quickly fades further behind the turbine and that the wake recovers. The velocity field starts to move towards a free stream condition just like in the upstream picture from Figure 4.17. Nevertheless, from the boundary conditions set, the outlet velocity should be equal to the inlet velocity. This is true, but as one can observe in Figure 4.19, the outlet at the right hand side is not entirely blue, but has some red areas. Thus, the swirl effect are still present. However, the velocity vectors seems to be rather co-linear with flow direction.

4. Case Validation

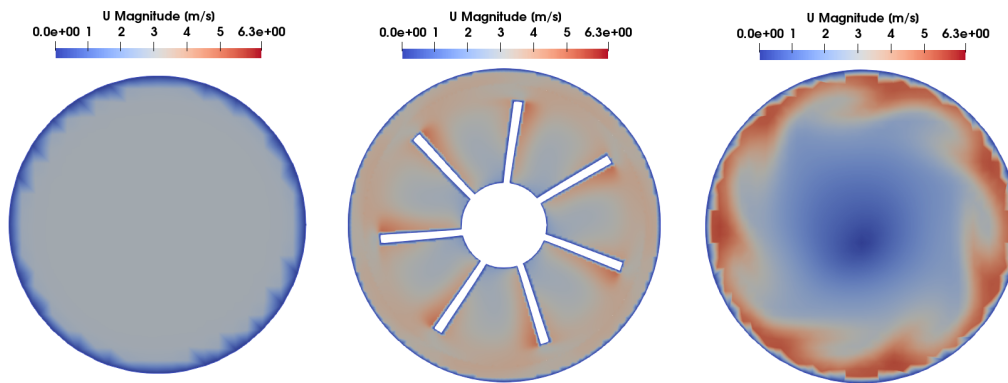


Figure 4.17: Averaged velocity field of the cellZone upstream (right), at blades (middle) and downstream (right) over 2000 iterations of the pilot design.

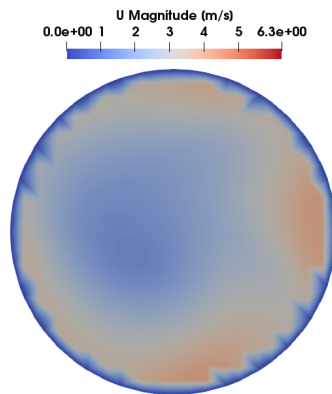


Figure 4.18: Averaged velocity field of the downstream far behind the turbine over 2000 iterations of the pilot design.

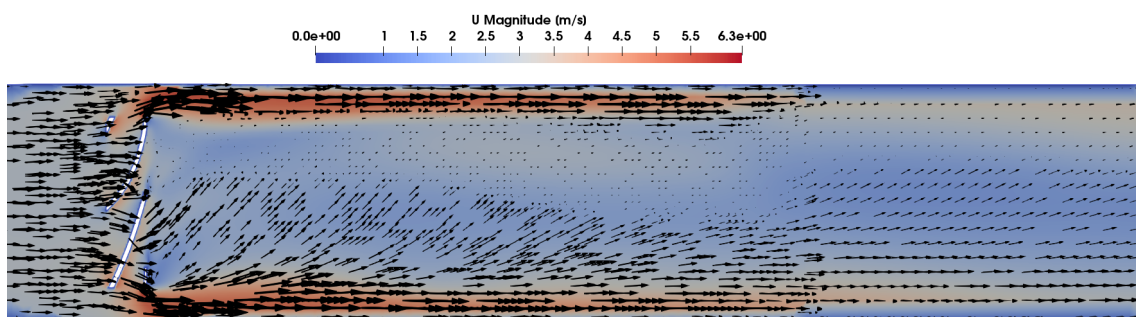


Figure 4.19: Averaged velocity field behind the turbine over 2000 iterations of the pilot design.

5

Results and Discussion

In this chapter the results of the different turbine design concepts are presented and discussed. Because the RPM for the optimized designs are unknown, multiple simulations were run with different RPMs to map the behaviour of each turbine design. Therefore, during this manual optimization process, Torque-RPM and Thrust-RPM graphs along with the turbines efficiency are evaluated. From Table 4.5, one can see that the deviation of thrust is quite high for the new model compared to the old model. The reason for this increase are most likely due to the larger flat areas behind the turbine seen in Figure 4.10. Also, the flat leading edges of the turbine blades are a little bit larger in area. Thus, the total flat surface area are larger. It is not beneficial for the technology that the turbine generates too high thrust force. Because of this, new designs with thrust deviations more than $\pm 15\%$ off compared to the old designs thrust values, are automatically rejected. Unfortunately, this means that the new CAD-model is rejected immediately.

Even though a high solidity gives higher torque and thrust values, it will most likely result in a higher pressure drop. A high efficiency is beneficial because in operation downhole, there are three turbines positioned in a row in each sub, motoring the three needles (see Figure 1.4). A high pressure drop will give even worse efficiency for the turbines behind the first one. Thus, a key factor in evaluating the best optimized design, was the efficiency.

Turbine blades with higher than 7 turbine blades are assumed to have a large pressure drop and low efficiency. However, it should result in a high torque output. Another aspect to discuss is the machining of the turbine it self. As the turbine diameter is to be constant, an increase of number of blades to 8 or 9 blades might give problems during machining. It might create lack of space for tools to operate. However, it is most likely possible, but the machining costs will increase. This is however not a problem for the designs studied in this project.

5.1 Designs

5.1.1 7 blades

The pilot design has 7 turbine blades. Small modifications to the design were done to evaluate behavior of torque, thrust and efficiency. In Table 5.1, all these parameters for the different designs are listed. Figures 5.1 and 5.2 show the expected torque behavior with different RPM. The large red dot in both graphs are the simulation results from the pilot design. These values are listed in Table 4.5. When studying the two figures and the table below, one can see that the torque is slightly increased for almost every design except **v5** and **v25**. The **v5** design is included to show that when the inlet flow angle becomes too small, the torque rapidly decreases of this turbine with 7 blades. The entire plot shifts down to the left hand side. However, even though almost all the designs gives an increase of torque output, the efficiency drops below the pilot design quite fast. This is due to their increase in solidity, which again gives a higher pressure drop. When looking at a compromise between efficiency and torque, only design **v26**, **v28** and **v39** are acceptable. All three designs give higher efficiency and torque values. Despite the fact that all these design are good, **v26** and **v39** are considered as the best designs for a 7 bladed turbine.

Table 5.1: Design parameters for the study on 7 blades.

	Stagger Angle [deg]	Inlet Angle [deg]	Chord line [mm]	Solidity
Pilot design	70	46	65.79	1.301
v5	70	30	65.79	1.301
v13	75	43	86.93	1.595
v24	75	35	86.93	1.595
v25	75	40	86.93	1.595
v26	75	46	86.93	1.595
v27	72	43	72.81	1.399
v28	77	43	100.02	1.776
v39	77	46	100.02	1.776
v40	77	40	100.02	1.776
v44	77	35	100.02	1.776
v45	72	46	72.81	1.399

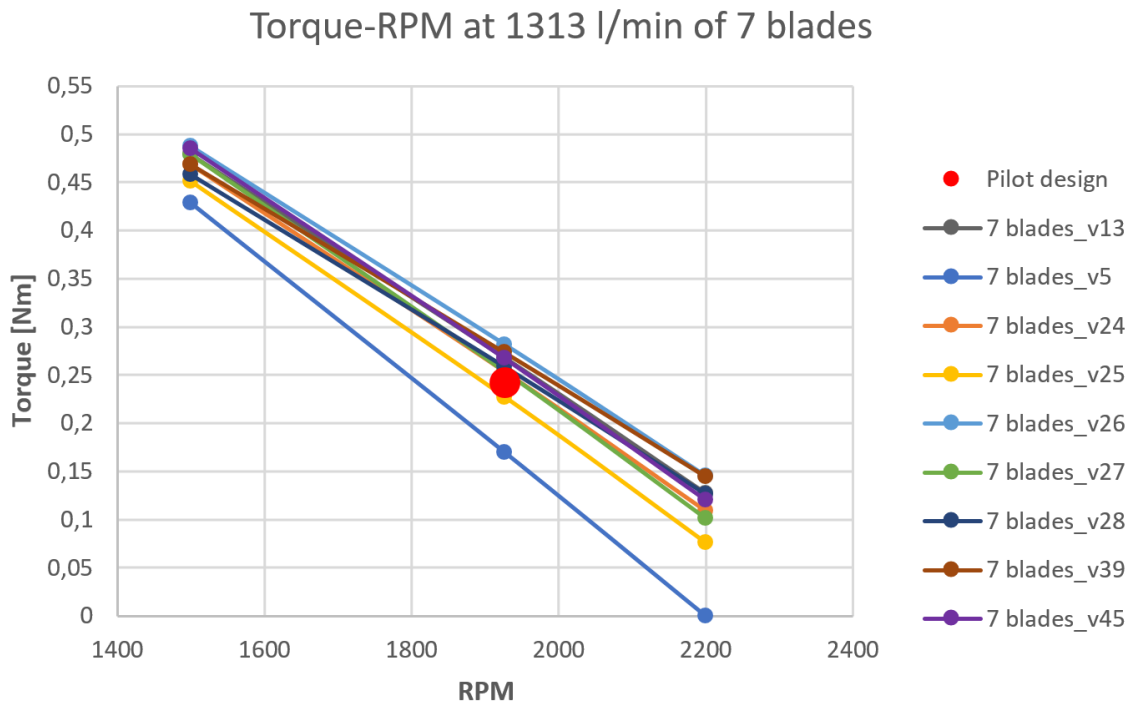


Figure 5.1: Torque-RPM table of 7 blades at volume flow of 1313 l/min.

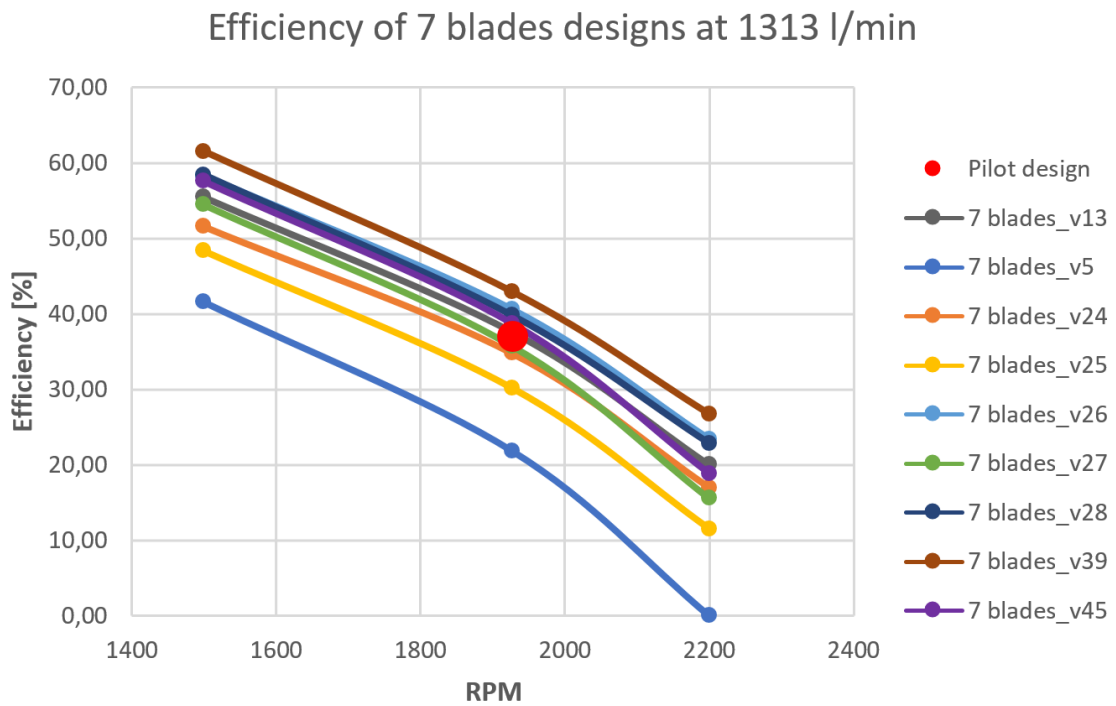


Figure 5.2: Efficiency-RPM table of 7 blades at volume flow of 1313 l/min.

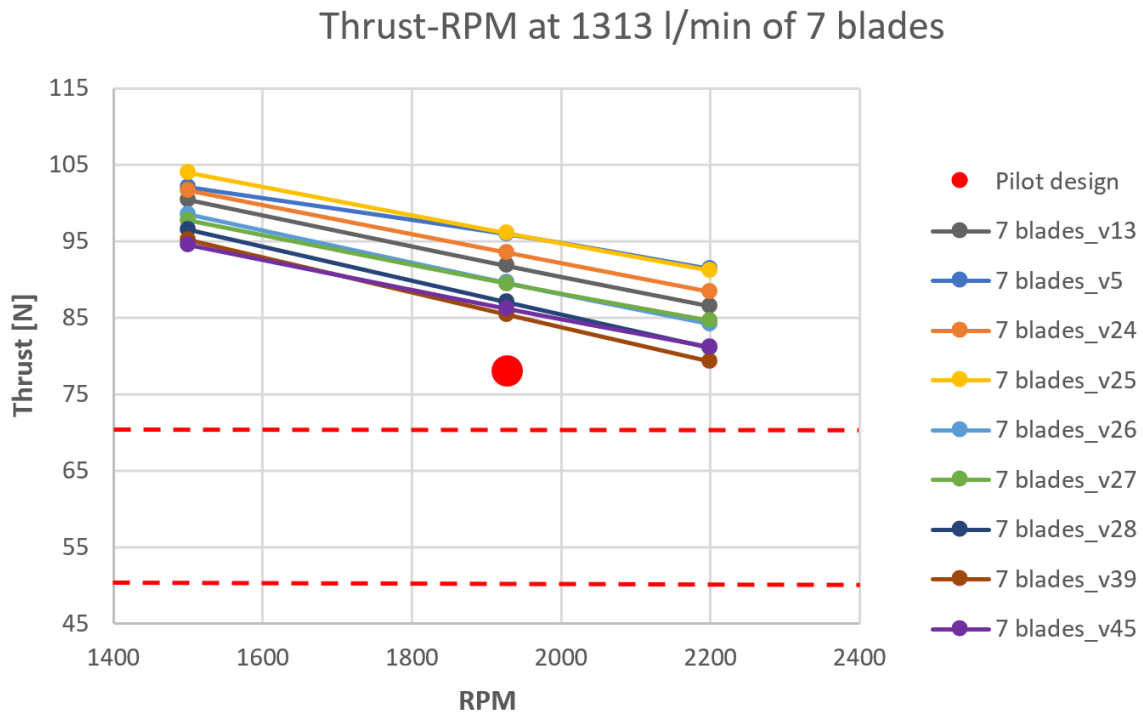


Figure 5.3: Thrust-RPM table of 7 blades at volume flow of 1313 l/min.

As discussed above, all designs with more than $\pm 15\%$ thrust compared to the thrust results from the old CAD-model are rejected. From Figure 5.3, one can see that all design concepts gives a higher thrust than wanted. Thus, all 7 bladed turbine concepts are rejected. Design **v28** and **v39** are considered acceptable for efficiency and torque, but not for the thrust. These thrust values are about 15% above the higher acceptable limit.

5.1.2 6 blades

It is quite interesting to see the results from the simulations of the 6 bladed turbines. Compared to the pilot design (new model), all designs are resulting in higher torque values and efficiency. The most remarkable design for the 6 bladed turbines is **v23** and is closely followed by **v16**. The solidity of **v23** and the pilot design is quite close, and therefore the RPM is expected to be almost identical. Due to the slightly higher solidity of **v23**, the turbine could operate at an even lower RPM which will result in a higher torque. Again, the limitations of thrust output eliminates the best design concepts for a 6 bladed turbine, see Figure 5.6. The only design to have an acceptable thrust value is **v15** and from Figure 5.4 one can see that the torque values are rather close to the pilot design. The solidity of **v15** is also lower than for the pilot design, thus a higher RPM and lower torque can be anticipated at a volume flow rate of 1313 l/min.

Table 5.2: Design parameters for the study on 6 blades.

	Stagger Angle [deg]	Inlet Angle [deg]	Chord line [mm]	Solidity
Pilot design	70	46	65.79	1.301
v15	70	46	65.79	1.116
v16	75	46	86.93	1.367
v17	75	35	86.93	1.367
v18	75	40	86.93	1.367
v19	75	43	86.93	1.367
v20	72	43	72.81	1.199
v21	77	43	100.02	1.523
v22	72	46	72.81	1.199
v23	77	46	100.02	1.523

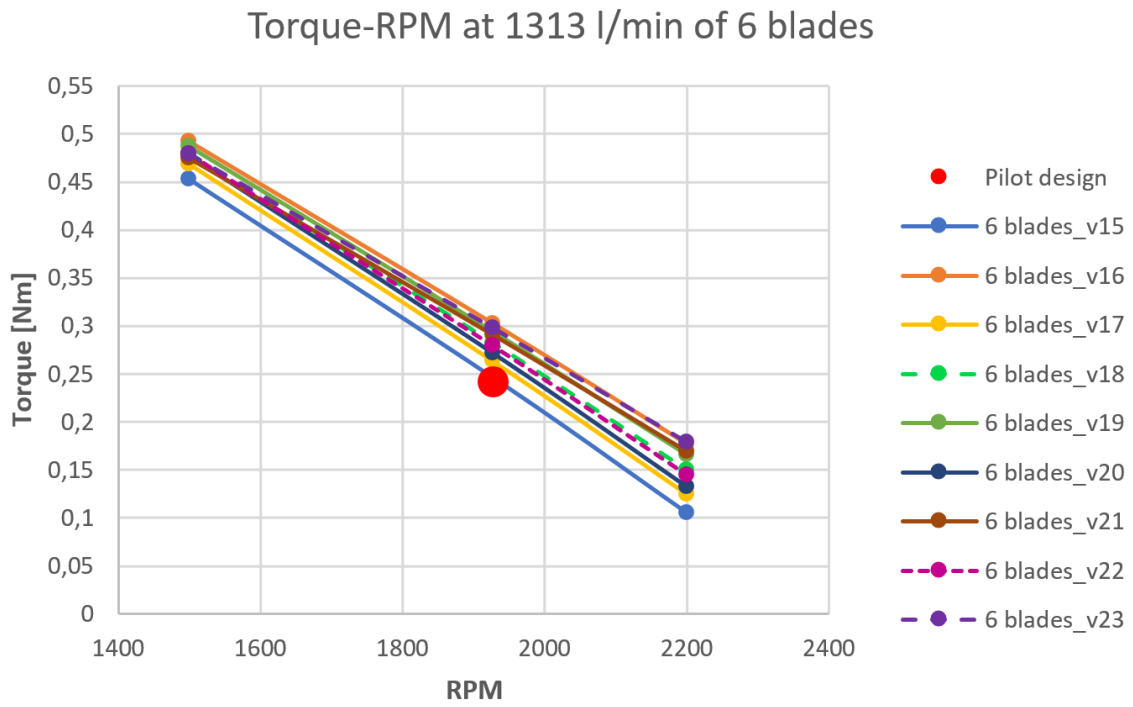


Figure 5.4: Torque-RPM table of 6 blades at volume flow of 1313 l/min.

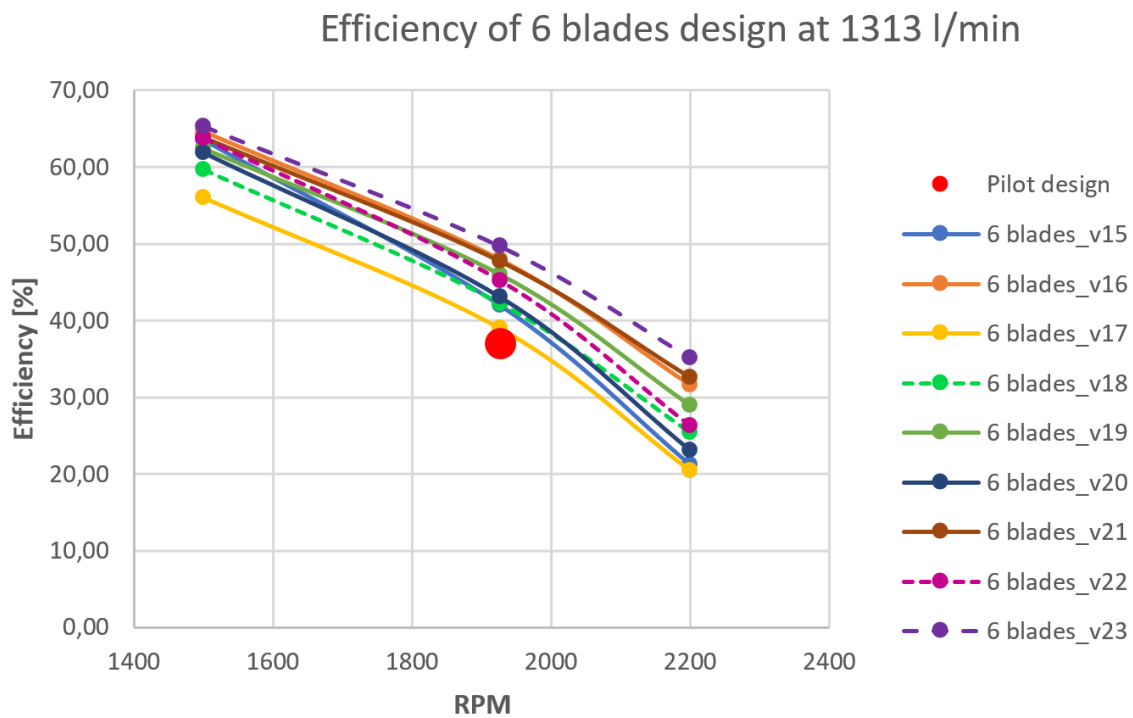


Figure 5.5: Efficiency-RPM table of 6 blades at volume flow of 1313 l/min.

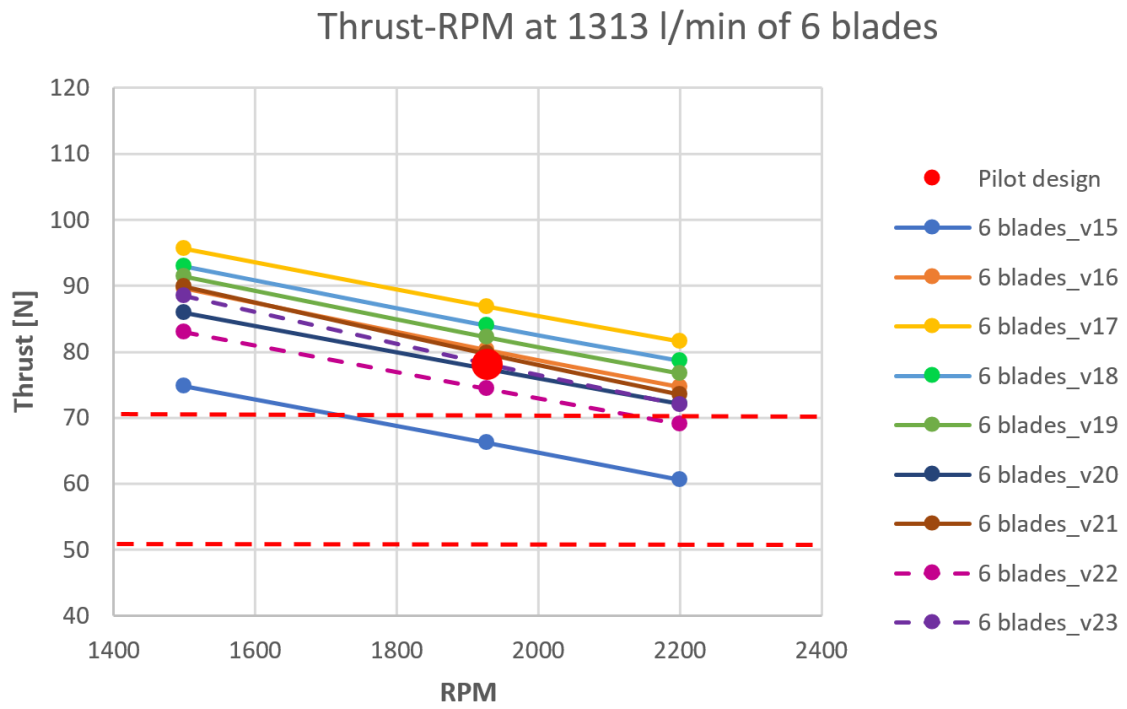


Figure 5.6: Thrust-RPM table of 6 blades at volume flow of 1313 l/min.

5.1.3 5 blades

For the 5 blades design concepts, all design concepts give higher torque and efficiency. The different designs are listed in Table 5.3 below. When looking at Figure 5.9, one can also see that most of the designs are within the acceptable thrust force range. Figure 5.7 show that for all RPM-values except 2200, that **v4** will be the superior design for achieving highest torque over a wide range of rotational speed.

Despite of this, **v4** does not give the best compromise between efficiency and torque. At 1927 RPM, the pilot design of the new CAD-model (7 blades) have an efficiency of 35.1%. **v10** has an efficiency of 58.67% and have a torque output equal to ≈ 0.31 Nm.

Table 5.3: Design parameters for the study on 5 blades.

	Stagger Angle [deg]	Inlet Angle [deg]	Chord line [mm]	Solidity
Pilot design	70	46	65.79	1.301
v4	75	43	86.93	1.139
v6	75	46	86.93	1.139
v8	75	35	86.93	1.139
v9	75	40	86.93	1.139
v10	77	46	100.02	1.269
v11	72	43	72.81	1.000
v12	77	43	100.02	1.269

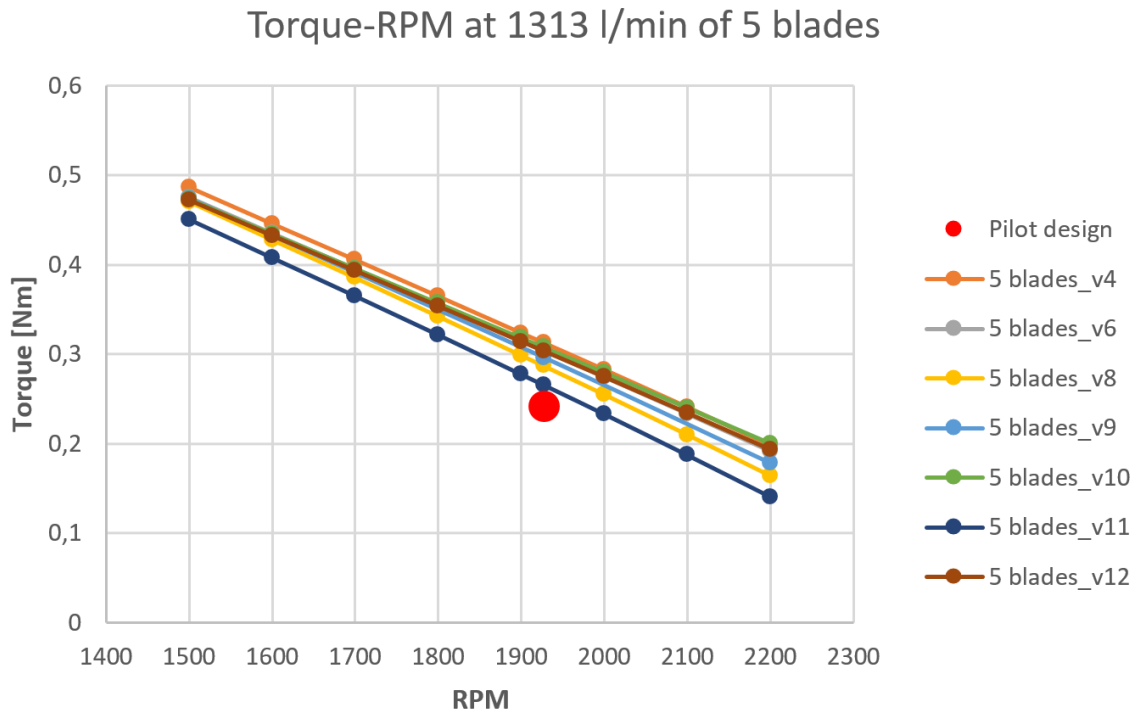


Figure 5.7: Torque-RPM table of 5 blades at volume flow of 1313 l/min.

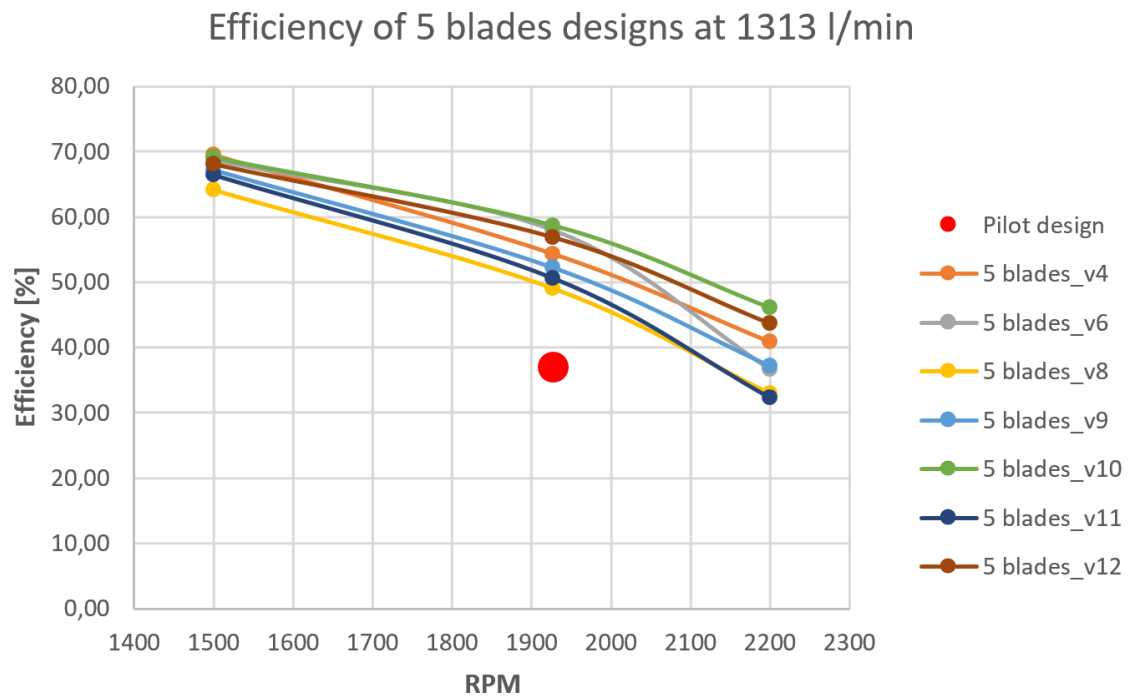


Figure 5.8: Efficiency-RPM table of 6 blades at volume flow of 1313 l/min.

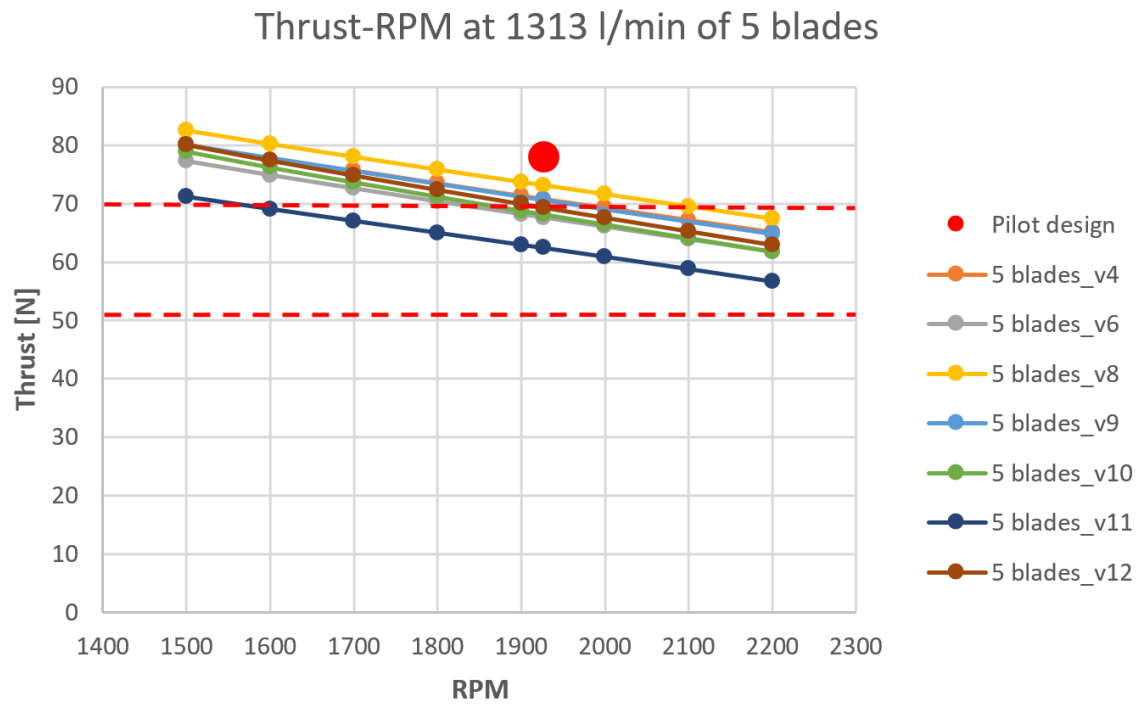


Figure 5.9: Thrust-RPM table of 5 blades at volume flow of 1313 l/min.

The reason why the efficiency curves in Figure 5.2, 5.5 and 5.8 are not all equal in shape because of the variety of stagger angles and inlet flow angles. What effects these parameters have will be presented and discussed in the next chapter.

5.1.4 Stagger and inlet flow angle effects

In order to identify what actually happens to the interesting quantities in this project when changing the design parameters Figures 5.11 and 5.10 are created. Based on the results presented in the sections above for the different number of turbine blades, a pattern was discovered. For the 7 bladed designs, **v28** and **v39** are chosen to be the best designs. And for the 6 bladed designs, **v16** and **v23** are the best designs. Last, but not least, for the 5 bladed designs, **v4** and **v10** are chosen. All these designs have some common parameters. The stagger angle is 75° or 77° and the inlet flow angle is either 43° or 46° . See the tables from the design concepts in the sections above.

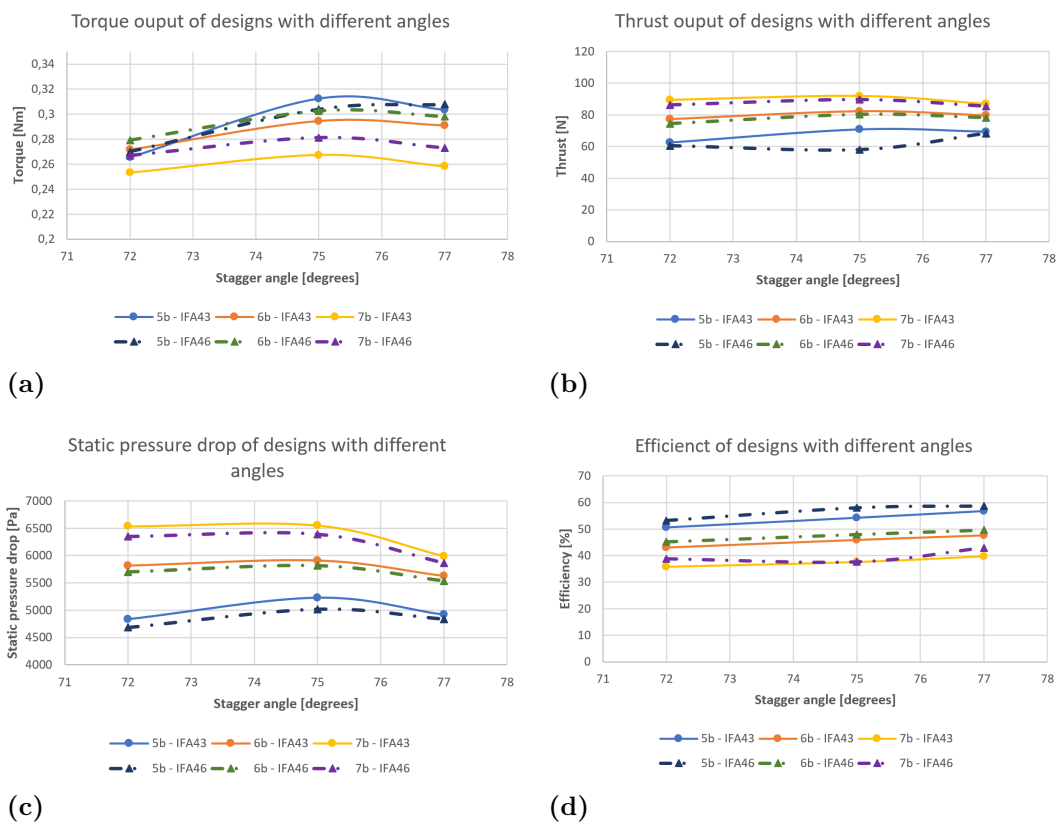


Figure 5.10: Stagger angle behavior for inlet flow angle (IFA) 43 and 46.

5. Results and Discussion

In Figure 5.10, shown above, designs with constant inlet flow angle of 43° and 46° are plotted against varying stagger angles. The dashed lines have an inlet flow angle (IFA) of 46° and the normal lines have an inlet flow angle of 43° . In Figure 5.10, one can see that designs with an inlet flow angle of 46° are dominating. These designs generally give higher torque output, a little bit lower thrust, and slightly higher efficiency. One can observe some interesting behavior of the 5 bladed design with 75° stagger angle and 43° inlet flow angle, design **v4**. Seen in Figure 5.10a, it is the only design that actually give a higher torque with an inlet flow angle of 43° . Otherwise, the dashed lines with an inlet flow angle of 46° is best.

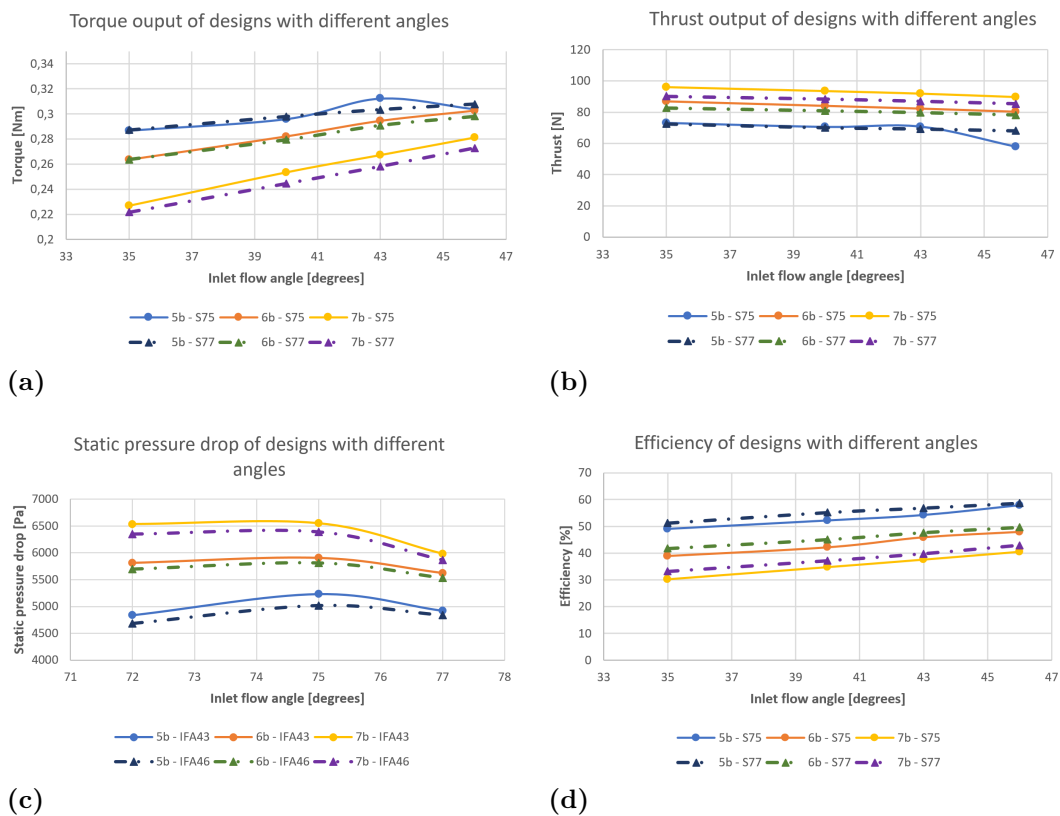


Figure 5.11: Inlet flow angle behavior for stagger angle (S) 75 and 77.

As mentioned in Section 3.2.1, the stagger angle is directly connected to the solidity of the turbines. Larger stagger angle, gives longer chord line and thus larger blade surfaces. When keeping the stagger angles constant and plotting the torque, thrust, static pressure drop and efficiency against inlet flow angle, the same trends as for Figures 5.10 are expected. In Figure 5.11, all the dashed lines are designs with a stagger angle equal to 77° , the normal lines are the designs with stagger angle 75° . From Figure 5.11a, one can clearly see that a low inlet flow angle is not beneficial in the pursue of achieving higher torque output. The static pressure drop is more or less the same for all the inlet flow angles evaluated. Once again, design **v4** (the blue normal lines), is the design that gives irregularities in the plots. All these results are gathered with a global mesh grid that is fairly low, (32 32 152). As concluded in the mesh convergence study, this was the optimum mesh grid. Even though the torque converges for chosen optimum mesh grid, it might be that the odd behaviors for design **v4** is due to bad mesh grid. But, it can also be as simple as that for 5 turbine blades, the torque starts to decline after peaking at an inlet flow angle of 43° . Due to limited time, this irregularity was not studied in depth.

5.2 Evaluation of the best design

As discussed in the sections above, there are mentioned two best designs for each of the number of blades studied. The parameters and results are listed in the tables below, Tables 5.4 and 5.5.

Table 5.4: Parametric overview of the best designs.

	Blades	Stagger Angle [deg]	Inlet Angle [deg]	Chord line [mm]	Solidity
v4	5	75	43	86.93	1.139
v10	5	77	46	100.02	1.269
v16	6	75	46	86.93	1.367
v23	6	77	46	100.02	1.523
v26	7	75	46	86.93	1.595
v39	7	77	46	100.02	1.776

Table 5.5: Output overview of the best designs at 1927 RPM and 1313 l/min.

	Blades	Torque [Nm]	Thrust [N]	ΔP [Pa]	Efficiency [%]
v4	5	0.3124	70.76	5232	54.29
v10	5	0.3078	68.16	4838	58.67
v16	6	0.3026	80.24	5816	47.97
v23	6	0.2981	78.20	5534	49.67
v26	7	0.2812	89.58	6392	40.57
v39	7	0.2729	85.40	5865	42.91

Assuming that all designs actually operate at 1927 RPM for a volume flow of 1313 l/min, the highest torque output observed from the design concepts is 0.3124 Nm. Compared to the pilot design, **v4** has an increase of 29.19% for torque and 17.23% efficiency points. However, the solidity of **v4** is lower than for the pilot design. Therefore, this turbine is likely to rotate even faster at a volume flow of 1313 l/min. This means that these numbers are expected to be a little bit lower in reality. In Table 5.5, one can see that **v10** is the design that gives the highest efficiency. With an efficiency of 4.38% points more than the second best design, it is considered to be the best compromised design. Although, **v4** gives the highest torque output, the graphs illustrated in Figure 5.11 and 5.10, show that there might be some errors in the simulation data for this design. These errors might be numerical trouble in the solvers or a mesh issue that was invisible to the "checkMesh" command.

5.2.1 Geometry Comparison

The figures below show a comparison of the old design and the best design in this project. Figures 5.14a and b, are the front and back views of the pilot design, and Figures 5.14c and d, are the front and back views of the best design. From these figures one can observe that the number of blades for the best design is 5 blades compared to the 7 blades in the pilot design. In Figure 5.13, a section view of the blade tip of both designs are illustrated. The chord length of the best design are clearly longer than the old design. Note that these cross-section views might have some deviations from the real geometry. Both designs have the same cap shape and size, and outer diameter as these are set constant during the project. Another thing that these designs have in common, is the root sketch and outlet flow angle at the tip sketch, see Table 3.1.

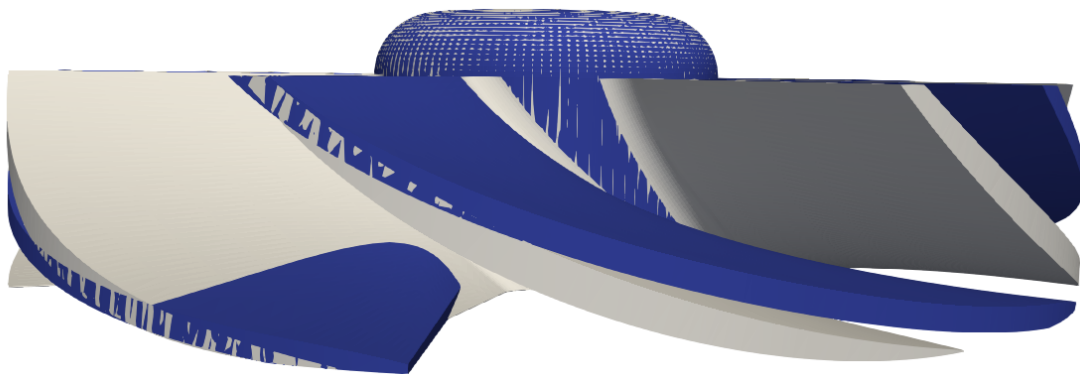


Figure 5.12: Comparison of the old design (white) and the best design (blue).

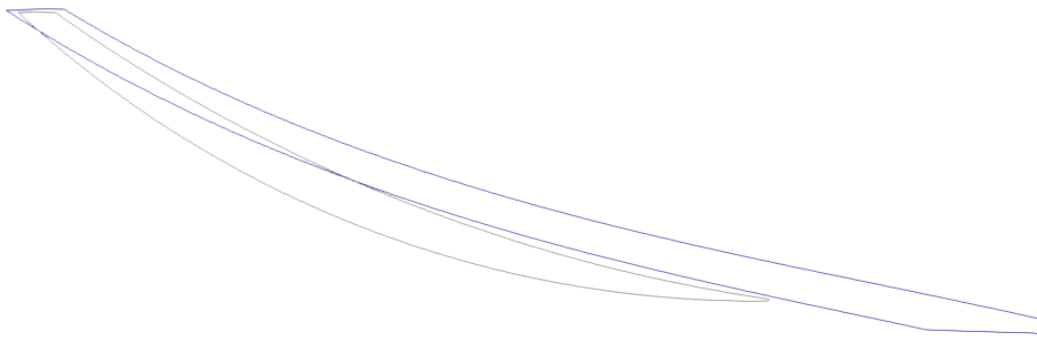
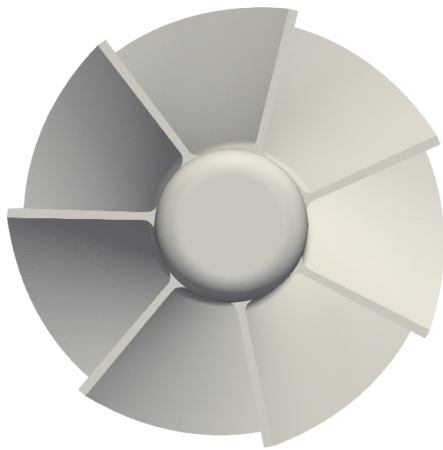
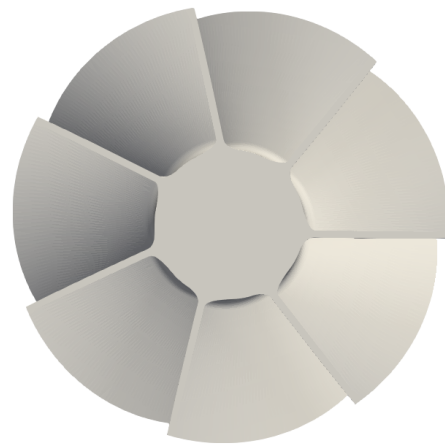


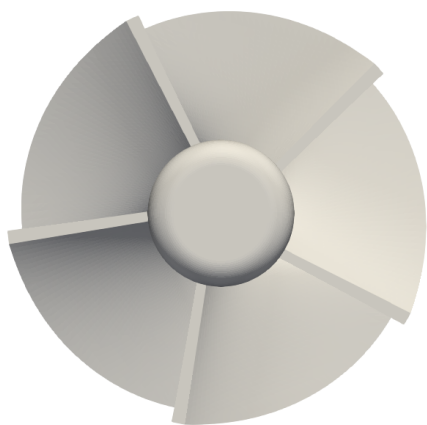
Figure 5.13: Cross-section view of the blade tip of old design (white) and the best design (blue).



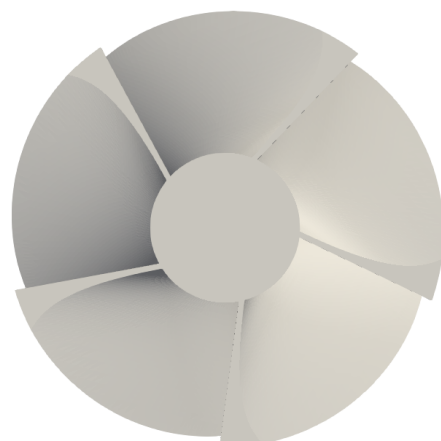
(a) Front view of pilot design.



(b) Back view of pilot design.



(c) Front view of best design.



(d) Back view of best design.

Figure 5.14: Comparison of the old design and the best design.

5.2.2 Flow fields

A comparison between the flow fields from Section 4.4, and the flow fields of the final design made in this section.

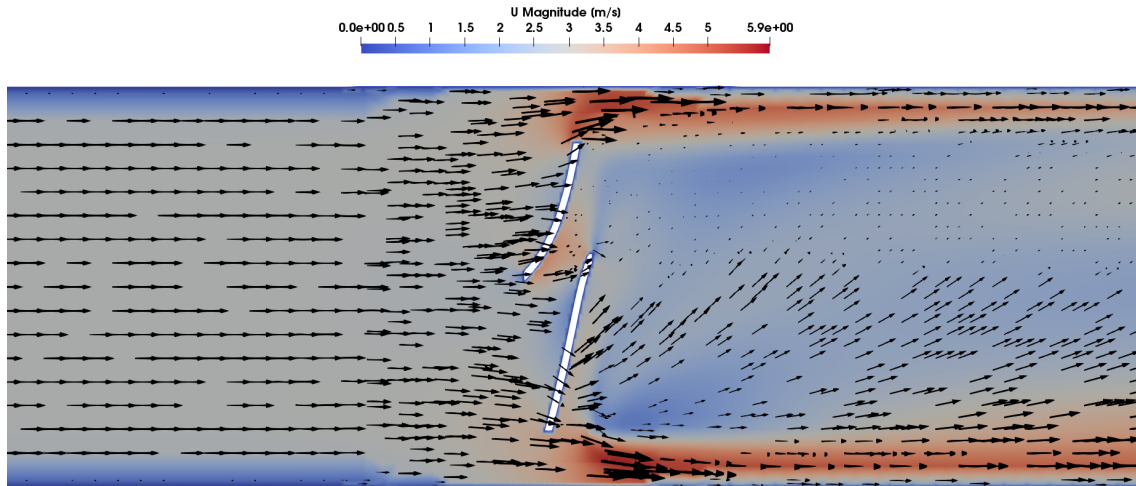


Figure 5.15: Averaged velocity field of the best design, **v10**, over 2000 iterations.

When comparing the two averaged velocity fields in Figures 5.15 and 4.16, the flow fields are quite the same. However, the highest mean velocity is 0.4 m/s higher than for the pilot design compared to the best design. This means that the kinetic velocity of the water reaching the second turbine in the well bore are lower. However, this difference is not much at all, and the deviation is negligible. Still, there are some directional velocity vectors going through the blades. For the pressure field, the pilot design and the best design, are quite similar as well, see Figures 4.14 and 5.16. The pressure drop in the pilot design was measured to be 6017 Pa. This turbine have 7 blades and a solidity of 1.301. The best design have 5 turbine blades and a solidity of 1.269. Thus, a lower static pressure drop is expected. This is also the case. The pressure drop is reduced with 1179 Pa.

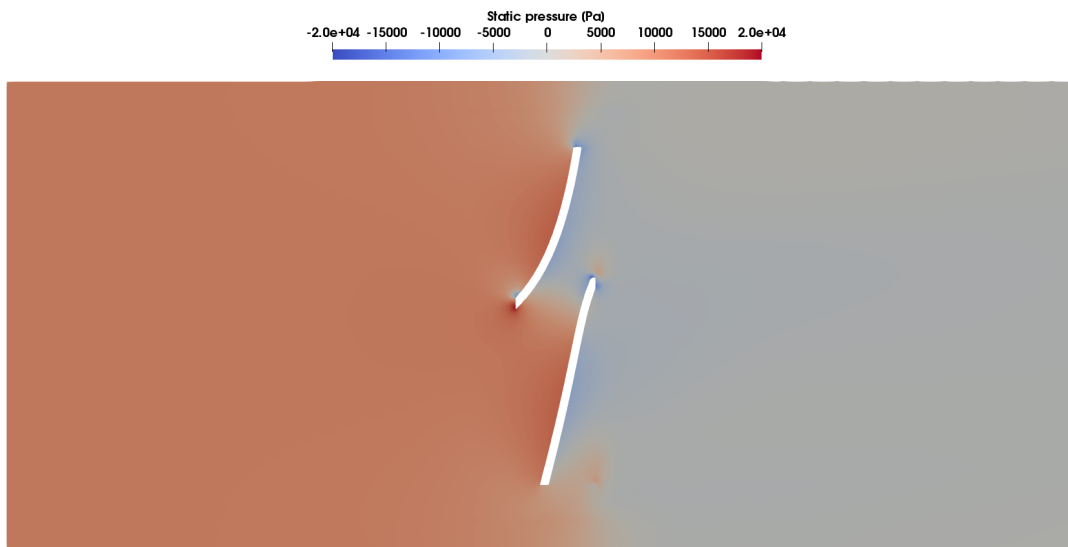


Figure 5.16: Pressure field of cellZone of the best design, **v10**, over 2000 iterations.

5.2.3 Mesh quality of best design

As mentioned previously, the mesh quality errors differ for all designs in this project. Even though the pilot design with a global mesh grid of (32 32 152) had zero mesh quality errors, the best design had 2 skewed faces. Both these highly skewed faces are at the sharp edges of the turbine blade, but these are considered not to be critical for the simulation outputs as Engys [27] stated.

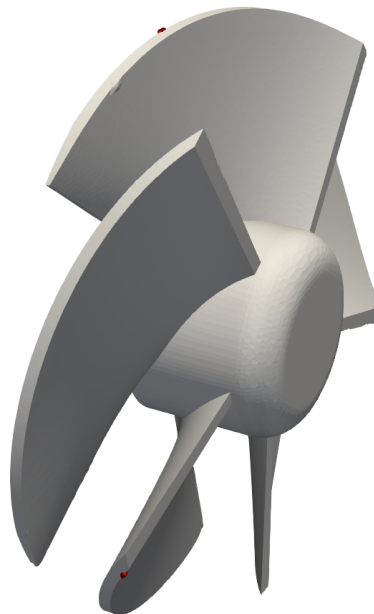


Figure 5.17: The 2 skewed faces illustrated as spheres at the turbine blades for the best design.

6

Conclusion

The objective of this project is to optimize the design of a turbine that utilizes the kinetic energy in a flow to generate torque. The optimization process is done manually by changing different angles of the turbine in a well-defined parametric CAD-model. The CFD simulations are run with an open source software called OpenFOAM. The simulation case is set up in a way that the different turbine concepts can easily be changed without other manipulations to the set-up. A total of 45 different turbine design concepts are simulated for different rotational speeds to map the torque behaviour of the turbines for the operational volume flow. The best design from the simulations is design concept **v10**. This design has an efficiency increase of 21.57% and a 27.3% increase in torque compared to the pilot design. It is highly recommended to produce this turbine and do physical testing to validate these results.

Because the simulated results deviates quite a lot compared to the experimental data, the simulation set-up is considered to have room for improvement. However, the set-up should reflect the relative deviations in a decent matter. OpenFOAM is a suitable software for this project as there are many concepts that need to be run and evaluated. OpenFOAM allows for easy evaluation of the simulated data and required post processing. The dynamic mesh-approach is in many cases considered to be a more accurate approach for rotating geometries. The non-rotating turbine-approach shows a close approximation and similar behavior of the results from the convergence study. It was therefore concluded that the rotating -approach is too time-consuming for a project like this, where evaluation of many different design concepts is needed.

6.1 Future work

In order to increase the efficiency and torque output of the turbine in this project there are several things to look into. The choices made in this project were done to limit the project. A possible solution for future study is to include stators to the system. It might be difficult to implement to this technology, but it can open for more flow to move directly onto the turbine blades by creating a pre-swirl [31].

As mentioned, the main task for the hub in this project is to have a stationary housing around the turbine. However, there are major possibilities for improvement of the current design. Introducing a wing geometry to the hub will allow the volume flow to accelerate more towards the turbine blades. This will open for a better utilization of the available energy in the flow. When it comes to improving the turbine geometry it self, experimenting with a more cone-shaped cap might increase the fluid velocity towards the turbine. Another interesting study might be to evaluate one more stagger and inlet flow angles to see if the trend shown in Figures 5.10 and 5.11 is continuous or not.

For the simulation set-up in this project, many meshing parameters and schemes were kept constant compared to the propeller tutorial in OpenFOAM. Thus, a study using different appropriate schemes and solvers might increase the possibility of achieving closer results compared to the experimental data. As the RPM of the turbine is flow driven, this is an unknown quantity for optimized turbine. The six degree of freedom simulation Holzmann [21] developed might be interesting to implement to see the results.

There are also other softwares that can be utilized for optimization. CAESES, for one, opens for an efficient way to find an optimized design [33].

Bibliography

- [1] Fishbones (2019). *Fishbones Stimulation Technology*. Accessed 21.01.2019 from: <http://fishbones.as/technology-in-brief>
- [2] Fishbones (2019). *Fishbones Stimulation Technology*. Accessed 23.01.2019 from: <http://fishbones.as/drilling>
- [3] Equinor (2015). *Smørbukk South Extension in operation - producing from "tight" reservoir*. Accessed 21.02.2019 from: <https://www.equinor.com/en/news/archive/2015/09/04/04SepSmoerbukk.html>
- [4] Stroud, I. (2006). *Boundary Representation Modelling Techniques*. Springer-Verlag London Limited, UK.
- [5] Greenshields, C. (2013). *OpenFOAM 2.2.0: snappyHexMesh*. Accessed 04.03.2019 from: <https://openfoam.org/release/2-2-0/snappyhexmesh-features-layers-baffles/>
- [6] Mehdipour R. (2013). *Simulating propeller and Propeller-Hull Interaction in OpenFOAM*. MSc Thesis, Centre for Naval Architecture Royal Institute of Technology, Stockholm, Sweden.
- [7] Equinor (2018). *Åsgard Fishbones Dreamliner experience Final evaluation 2018*. Unpublished manuscript.
- [8] Versteeg H. K. and Malalasekera W. (2007). *An Introduction to Computational Fluid Dynamics The Finite Volume Method, 2nd edition*. Pearson Education, Harlow, UK.
- [9] Greenshields, C. (2018, July). *OpenFOAM 6 Released*. Accessed 18.03.2019 from: <https://openfoam.org/release/6/>
- [10] OpenFOAM (2019) *ESI-OpenCFD*. Accessed 10.06.2019 from: <https://www.openfoam.com/about/>
- [11] Dixon, S. and Hall, C. (2014). *Fluid Mechanics and Thermodynamics - Turbomachinery, 7th edition*. The Boulevard, Langford Lane, Kidlington, Oxford, OX5 1GB, UK, 225 Wyman Street, Waltham, MA 02451, USA: Butterworth-Heinemann is an imprint of Elsevier.
- [12] Chanson, H. (2004). *Hydraulics of Open Channel Flow, 2nd Edition*. Elsevier Butterworth-Heinemann, Linacre House, Jordan Hill, Oxford OX2 8DP, 200 Wheeler Raad, Burlington, MA 01803, USA.
- [13] Oertel, H., Prandtl, L., Böhle, M., Mayes, K., (2004). *Prandtl's Essentials of Fluid Mechanics, 2nd Edition*. Springer-Verlag New York, Inc., 175 Fifth Avenue, New York, NY 10010, USA.
- [14] Tu, J., Yeoh, G-H., Liu, C., (2013). *Computational Fluid Dynamics A Practical Approach, 2nd Edition*. The Boulevard, Langford Lane, Kidlington,

- Oxford, OX5 1GB, UK, 225 Wyman Street, Waltham, MA 02451, USA: Butterworth-Heinemann is an imprint of Elsevier.
- [15] CFD Online (2011, February). *SST k-omega model*. Accessed 06.04.2019 from: https://www.cfd-online.com/Wiki/SST_k-omega_model
- [16] Cebeci, T. (2003). *Turbulence Models and Their Application*. Horizons Pub., Long Beach, California and Springer, Heidelberg, Germany.
- [17] Greenshields, C. (2018, July). *5.4 Mesh generation with the snappyHexMesh utility*. Accessed 10.04.2019 from: <https://cfd.direct/openfoam/user-guide/v6-snappyhexmesh/>
- [18] Abeykoon, C., Hantsch, T. (2017). *Design and Analysis of a Kaplan Turbine Runner Wheel*. Accessed 27.04.2019 from: https://www.research.manchester.ac.uk/portal/files/56597486/Design_and_Analysis_of_a_Kaplan_Turbine_Runner_Wheel.pdf
- [19] Muratoglu, A., Yuce, M. I. (2015). *Performance Analysis of Hydrokinetic Turbine Blade Sections*. Accessed 27.04.2019 from: <https://are.avestia.com/2015/PDF/001.pdf>
- [20] Mukherji, S.S.et Al. (2011), *Numerical investigation and evaluation of optimum hydrodynamic performance of a horizontal axis hydrokinetic turbine*. Accessed 06.05.2019 from: <https://www.lehigh.edu/~inenr/research/PDF/Hydrokinetic-AIP-2011.pdf>
- [21] Holzmann, T. (2019). *Vertical Axis Wind Turbine*. Accessed 08.05.2019 from: <https://holzmann-cfd.de/advanced-material/openfoam-tutorials/item/5-vertical-axial-wind-turbine-vawt>
- [22] ANSYS Inc (2009). *ANSYS FLUENT User's Guide - Using Flow Boundary Conditions*. Accessed 11.05.2019 from: <http://www.afs.enea.it/project/neptunius/docs/fluent/html/ug/node238.htm>
- [23] Greenshields, C.J. (2018). *OpenFOAM User Guide version 6*, pp. U-44, U-129-131.
- [24] Jasak, H. (1996). *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows*. Accessed 15.05.2019 from: <http://powerlab.fsb.hr/ped/kturbo/OpenFOAM/docs/HrvojeJasakPhD.pdf>
- [25] Rhoads, J. (2014). *Effects of grid quality on solution accuracy*. Accessed 15.05.2019 from: <https://afinemesh.files.wordpress.com/2014/07/ofw20141.pdf>
- [26] ANSYS Inc (2009). *ANSYS FLUENT User's Guide - Mesh Quality*. Accessed 19.05.2019 from: <http://www.afs.enea.it/project/neptunius/docs/fluent/html/ug/node167.htm>
- [27] Engys (2012). *A Comprehensive Your of snappyHexMesh* Accessed 27.05.2019 from: <https://openfoamwiki.net/images/f/f0/Final-AndrewJacksonSlides0FW7.pdf>
- [28] Cebeci, T., Shao, J. P., Kafyeke, F., Laurendeau, E. (2005). *Computational Fluid Dynamics for Engineers*. Horizons Pub., Long Beach, California and Springer, Heidelberg, Germany.
- [29] Nan, D., Shiegemitsu, T. and Zhao, S. (2018). *Investigation and Analysis of Attack Angle and Rear Flow Condition of Contra-Rotating*

-
- Small Hydro-Turbine*. Accessed 29.05.2019 from: <https://www.mdpi.com/1996-1073/11/7/1806/pdf>
- [30] Byeon, S., Kim, Y. (2013). *Influence of Blade Number on the Flow Characteristics in the Vertical Axis Propeller Hydro Turbine*. Accessed 05.06.2019 from: https://www.researchgate.net/publication/264132347_Influence_of_Blade_Number_on_the_Flow_Characteristics_in_the_Vertical_Axis_Propeller_Hydro_Turbine/download
- [31] Wärtsilä (2017). *Wärtsilä introduces an innovative pre-swirl stator to improve fuel efficiency*. Accessed 12.06.2019 from: <https://www.wartsila.com/media/news/07-11-2017-wartsila-introduces-an-innovative-pre-swirl-stator-to-improve-fuel->
- [32] Cengel, Y. A., Cimbala, J. M., Turner, R. H. (2012). *Fundamentals of Thermal-Fluid Sciences*. McGraw-Hill Education, 2 Penn Plaza, New York, NY 10121, USA.
- [33] CAESES (2019). *CAESES*. Accessed 13.06.2019 from: <https://www.caeses.com/products/caeses/>
- [34] OpenFOAM wiki (2009). *See the MRF development*. Accessed 13.06.2019 from: https://openfoamwiki.net/index.php/See_the_MRF_development
- [35] Petit, O., Page, M., Beaudoin, M., Nilsson, H. (2009). *The ERCOFTAC centrifugal pump OpenFOAM case-study*. Accessed 13.06.2019 from: https://www.researchgate.net/publication/239824858_The_ERCOFTAC_centrifugal_pump_OpenFOAM_case-study

A

Appendix 1 - Folder Structure

On the next page, Figure A.1, is a simplified folder structure of the case illustrated. In the *system*-folder there are more files like: *forces*, *decomposeParDict*, *residuals*, *surfaces* and *surfaceFeaturesDict*. The *decomposeParDict* let you run the case in parallel. The *forces* gives the force outputs, and *residuals* gives the wanted residuals data. All geometries (.stl-files) are positioned in the *triSurface*-folder under the *constant*-folder.

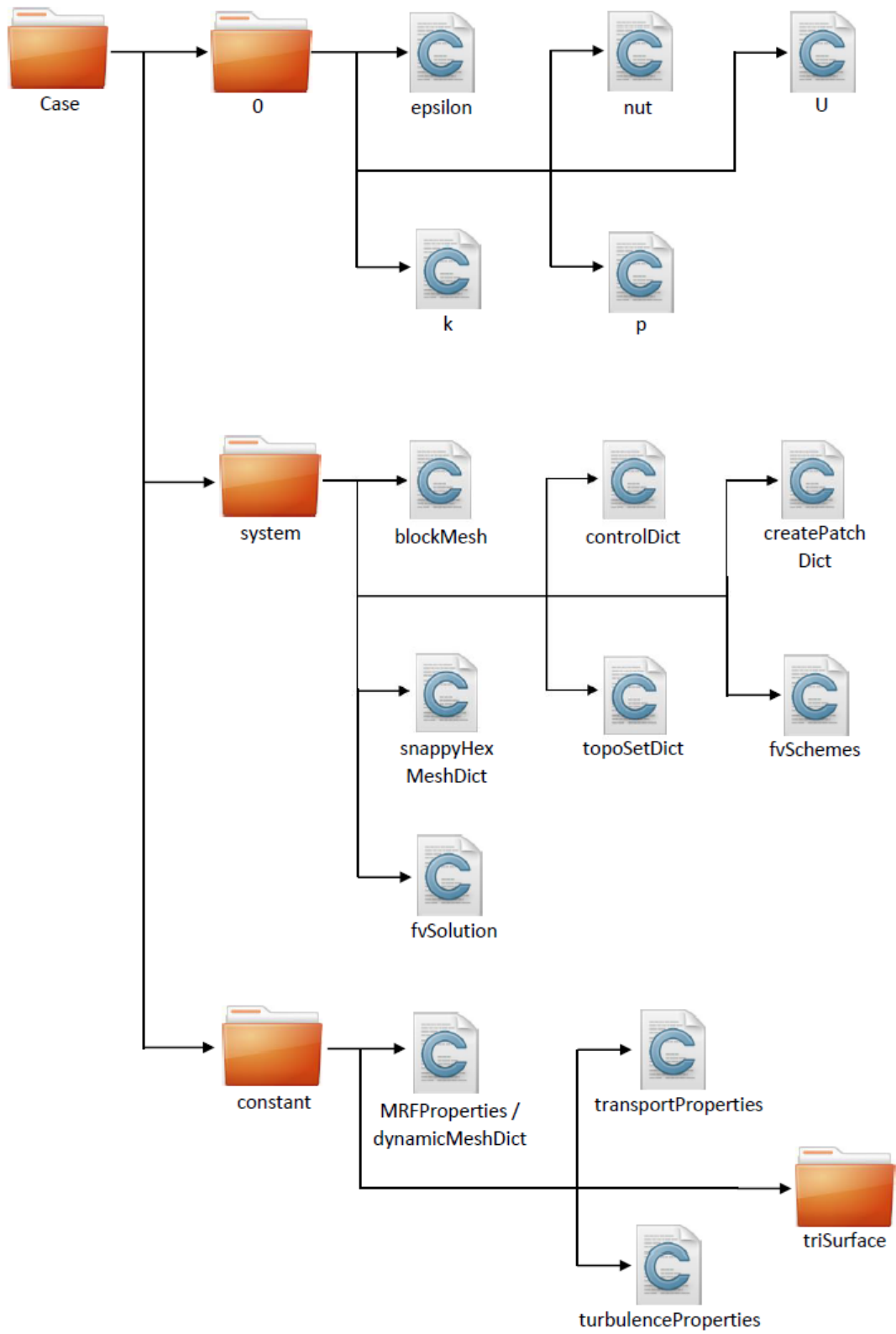


Figure A.1: Folder structure.