Universitetet
i Stavanger

DET TEKNISK-NATURVITENSKAPELIGE FAKULTET

MASTEROPPGAVE

Studieprogram/spesialisering: **Structures and Materials** **Specialization in Renewable Energy**	Vårsemesteret, 2020 Åpen / ~~Konfidensiell~~
Forfatter: **Stian Hjorteland**	**Stian Hjorteland** (signatur forfatter)

Fagansvarlig: Knut Erik Teigen Giljarhus

Veileder(e): Knut Erik Teigen Giljarhus

Tittel på masteroppgaven: CFD analyse av luftstrømmer på store turbinblader under ekstreme forhold

Engelsk tittel: CFD-simulations of flow over large turbine blades under extreme load conditions

Studiepoeng: 30

Emneord: Offshore Wind Turbine, Extreme loading conditions, Computational Fluid Dynamics, CFD, Airfoil, Grid, Mesh, Turbulence, Navier-Stokes, Drag Coefficient, DU96-W-180, SNL 100-03, OpenFOAM, PyFR, first order solution, second order solution, higher order solutions.	Sidetall: 87 + vedlegg/annet: 5 Stavanger, 29.06.2020 dato/år

CFD-simulations of flow over large turbine blades under extreme load conditions

Master Thesis, Structures and Materials

Universitetet
i Stavanger

Written by
Stian Hjorteland

Figure 1: Offshore wind farm

Forord

Denne oppgaven tar for seg muligheten ved å analysere luftmotstand på et vindturbinblad med høy angrepsvinkel under ekstreme vindlaster ved hjelp av beregningsassistert fluiddynamikk. Oppgaven tar også for seg en analyse av luftmotstand på en airfoil. En airfoil kan sees på som et vilkårlig snitt av et vindturbinblad, og brukes her for å validere resultatene mot allerede eksisterende resultater, sammenligne ulike løsningsmetoder, samt gi en dypere forståelse av aerodynamikk under høye angrepsvinkler. Til slutt tar også oppgaven for seg problemstillingen rundt nøyaktigheten til andre ordens løsninger i beregningsassistert fluiddynamikk ved høy separasjon og presenterer en mulig løsning gjennom en introduksjon av høyere ordens løsninger.

Jeg ønsker å takke mine foreldre for den avgjørende støtten jeg har mottatt gjennom et 5-årig studieløp. Den har vært helt avgjørende for meg, og jeg er evig takknemlig. Videre ønsker jeg å takke min kjære samboer som alltid stiller opp når jeg trenger det mest. Jeg ønsker også å takke familie og venner for gode innspill og korrekturlesning på oppgaven. Til slutt ønsker jeg å rette en stor takk til min veileder Knut Erik Teigen Giljarhus for god veiledning gjennom hele oppgaven.

Abstract

The goal of this project is to investigate the opportunity of performing CFD-analysis of flow over a large offshore wind turbine at a high angle of attack during extreme loading conditions. At a 90° angle of attack, the drag force is at its highest, making this loading condition the most critical for the turbine blade. Little work has been published on wind turbines at a high angle of attack, making the project a relevant contributor to future research.

In simple terms, an airfoil can be considered as a cross section of a turbine blade, making it a good reference for how one can expect the aerodynamics to work and behave at high angle of attacks. Furthermore, fluid flow at a high angle of attack often results in a highly separated flow, which has been proven hard to predict with the help of first and second order solutions. Higher order solutions are introduced and investigated in the search of a solution to the expected inaccuracy.

Moreover, different CFD solvers were tested for the DU96-W-180 airfoil at 90° angle of attack at a Reynolds number of $0.7 \cdot 10^6$. The results was then compared to already existing data in order to validate the computational setup, as well as comparing the different solvers. The results indicated that the second order solutions from OpenFOAM had problems predicting the drag coefficient accurately, as they were a bit higher than the experimental result. However, the third order solutions from PyFR was a closer match, supporting the statement of higher order solutions being better at predicting the aerodynamics at highly separated flows. In addition, the simplified results from Xfoil also showed a good ability of predicting the drag coefficient.

A full-scaled offshore wind turbine blade called SNL100-03 was modelled from scratch in Qblade. Some difficulties was encountered for the mesh generation and are described in chapter 8. Nevertheless, two different meshes were generated and simulated for three different extreme loading conditions. The flow behind the blade was considered to be similar to the flow for the DU-96-W-180 airfoil. As for the drag forces, there was a clear difference between the Fine Mesh and the Course Mesh, where the Course mesh appeared to be much more unstable in its prediction. It was also concluded that a finer mesh with a lower y+ could be achieved by using Pointwise without increasing the number of cells to a large degree, as Pointwise and snappyHexMesh has different ways of representing the geometry, resulting in CFD having a potential for solving extreme loading conditions on a full sized wind turbine blade at a high angle of attack.

Table of content

Figure list

Table list

1. Introduction

Offshore wind turbines are getting larger and recent trends are indicating that this growth will continue in the years to come. As the offshore wind turbines increase in size, several design problems are encountered, where one of them are the consequences of extreme loading conditions on the blades. During extreme weather the wind turbine stops rotating, resulting in the wind hitting the blade at a high velocity. Even though wind turbines are usually installed with pitching systems to reduce the high forces that occur during such an extreme weather, situations like mechanical failure and extreme change in wind direction can still happen, which may result in the wind hitting the blade at a high angle of attack. From a designer's point of view, these extreme loadings are important in order to design the blade with dimensions and materials to withstand the expected loadings. At 90° angle of attack, the drag force is at its highest, making this angle critical for the blades ability to withstand the loadings. In order to estimate the drag force, one can perform experimental testing. To execute this with the aid of wind tunnels could turn out to be difficult due to the actual size of the blades, which could be more than 100m, as well as being very expensive. Scaling down the size of the blade also has its problems, indicating that CFD analysis of a turbine blade might be a reasonable option when determining the forces during extreme weather at a high angle of attack. CFD, short for computational fluid dynamics, is a tool for solving equations about fluid flow with the help of a computer.

Barely any research is published on CFD simulations at high angle of attacks in extreme weather. Therefore, in order to perform a valid CFD simulation, it is necessary to check the accuracy of the computational setup. For this project, simulations of a DU96-W-180 airfoil at 90° will be executed and compared to already existing experimental results, as well as simulated results. The simulations will consist of a simplified method, first and second order CFD simulations in OpenFOAM and higher order CFD simulations PyFR. A highly separated flow with a Reynolds number between 10^4 - 10^7 has been proven difficult to reconstruct accurately with first and second order solutions. Based on this, higher order solutions are introduced through PyFR in order to compare the accuracy, where a small introduction case for a 2D cylinder is presented. By using different solvers, a broader perspective and knowledge of the different results can be achieved. The fact that there are already existing research on the DU96-W180 airfoil at 90°, as well as being originally designed as an airfoil for wind turbines makes it a natural option for this project. Moreover, the investigation of different solvers is easier to perform on an airfoil compared to a full sized turbine blade.

Based on these findings, a computational setup for the SNL 100-03 turbine blade will be designed, meshed and simulated for extreme loadings according to the *"IEC 61400 - 1 Wind Turbines - Part 1: Design Requirements"*. Problems regarding design increases with the size of the turbine. Since the SNL 100-03 blade has a blade length of 100m it is considered a natural choice. Another advantage with this blade is that there are a detailed description of the different cross sections in the scientific report *"The SNL 100-03 Blade: Design Studies with Flatback Airfoils for the Sandia 100-meter Blade"* written by D. Todd Griffith and Phillip W. Richards in 2014. The final goal with the setup for the SNL 100-03 blade is to check if it is realistic to perform accurate CFD-simulations, as well as the necessary computational power for such a large blade.

As for the limitations of the project, only drag forces will be discussed for all the simulations as this is considered the dominating force at a 90° angle of attack. The flow pattern is also discussed for all the simulations.

1.1. Background and motivation for the thesis

Originally, I have a background from Civil engineering and structures. By starting on the following Master's Degree: *"Structures and Materials, Specialization in Renewable Energy"*, I was introduced to new and unknown subjects like CFD. It was a refreshing feeling to work on something completely different and I quickly realized that using this tool for solving a problem related to renewable energy could be interesting. Moreover, I truly believe that an energy transition to renewables will take place in the future, making this thesis an excellent opportunity to dig deep into the subject.

1.2. Overview of the Master thesis

This Master thesis starts with an introduction of the recent trends in the offshore wind turbine industry. Chapter 3 is a literature study, introducing important information and results for both the DU96-W-180 airfoil, as well as the SNL 100-03 blade. Moving on to chapter 4 and 5, which is a presentation of the most important theory regarding both wind turbines and CFD simulations. As for chapter 6, 7 and 8, the design, meshing and simulation work done by the author is described. Finishing off with the conclusion in chapter 9 and an appendix after the reference list.

2. Offshore wind turbines

2.1. A brief history review of wind energy

Wind has been a significant energy source for many centuries. Humans understood quickly that taking advantage of the wind could save them for a lot work and heavy lifting. Based on historical findings, the Persians developed windmills for grinding as early as 500 A.D (Third Planet Windpower, 2020).

In 1854, Daniel Halladay invented a windmill that was designed for pumping groundwater to the surface. The new idea behind Halladay's windmill design was the inclusion of a tail at the back of the windmill, giving it the ability to adjust itself based on the wind direction. With this inclusion, the windmill was always faced towards the wind direction without any human assistance. Charles Francis Brush invented the first wind turbine to produce electricity in 1888. The wind turbine was designed with a diameter of 17 meter and a total of 144 rotor blades, giving a power production close to 12 kW (Connecticut history, 2018; Third Planet Windpower, 2020).

As we move on to the 19th century the technology improved and it was not uncommon for people living at remote locations in America to use wind turbines to produce electricity. In the late 19th century, renewable energy starts to get more attention. As a consequence of the oil crises in 1973, the United States of America was forced to start investing in renewable energy since the prizes of fossil fuels kept rising. Europe followed, and within 2000, they were the world-leading region on wind turbine technology. Today the focus on renewable energy is greater than ever, and it is expected that the industry of wind energy will continue to grow in the years to come (Third Planet Windpower, 2020).

2.2. Recent trends on offshore wind turbines

WindEurope is an association promoting wind power through thorough research and analysis of the most important factors in the business. Every year WindEurope publishes a report consisting of important details in the market, the amount of installed wind turbines and recent trends in the market. In February 2020, WindEurope published the annual report for 2019 consisting of the following trends:

- The average rated capacity of installed turbines in 2019 was 7.8 MW, which is 1 MW more than in 2018.
- In one decade, the average size of constructing wind farm has almost doubled to a value of 621 MW.
- Both the average distance from shore (59 km) and the water depth (33 m) of newly installed wind turbines keeps increasing.
- The auctions results from 2019 varied between 40 - 50 €/MWh.
- The first GE Haliade-X 12 MW prototype was installed.

An increase by 1 MW in the average rated capacity of newly installed turbines from 2018 to 2019 can be considered an expected increase as the trends shown on figure 2 (Ramírez, Fraile, & Brindley, 2020, p. 17) indicates a linear increase since 2014. Moreover, as the technology improves, giving us the ability to go bigger, deeper and further away from shore and opening up new areas for installations, it is expected that this will continue to grow in the years to come (WindEurope, 2020; Ramírez, Fraile, & Brindley, 2020).

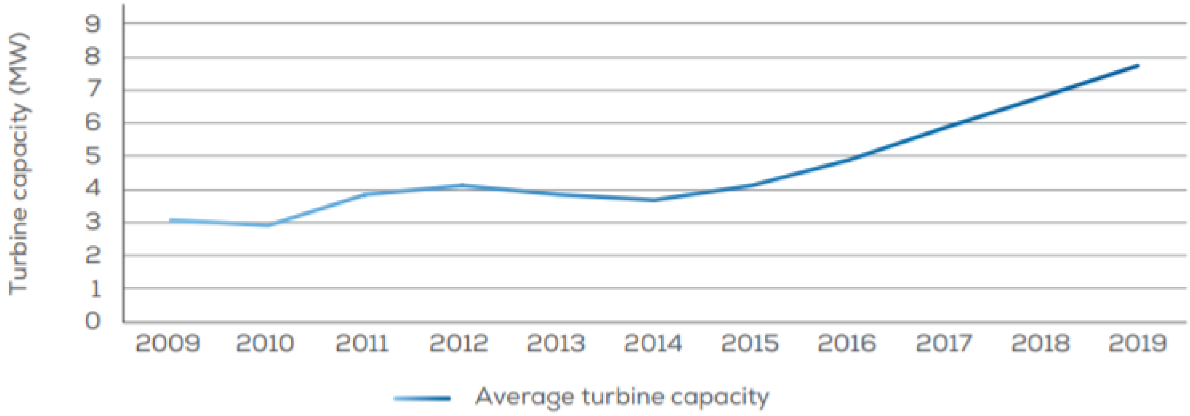


Figure 2: Average rated capacity of newly installed offshore wind turbines

Furthermore, the cost for investing in offshore wind turbines has drastically dropped the recent years. From a total of €4.41 million/MW in 2013, the prize dropped to €2.45 million/MW in 2018, resulting in the costs falling by 45% in only 5 years. This makes it easier for companies to invest in larger offshore wind turbines, making this an important factor when discussing the trends represented in figure 2 (Weston, 2019). Even though the average rated turbine capacity of newly installed turbines are expected to grow in the future, one might see that the graph in figure 2 flattens out over time due to strength and stiffness limitations in the structures and materials as the turbine blades gets larger.

2.3. Advantages by going offshore

Wind turbines are installed both onshore and offshore and there are advantages and disadvantages for both of them. When looking at available area for turbine installation, offshore has proven to be a better option compared to onshore. Onshore turbines are installed at a certain distance from cities and neighborhoods in order to avoid disturbance like sound, shadow flickering and visual impact. Taking this into account, the amount of available space onshore is reduced. On the other hand, going offshore will gain access to large areas with new installation opportunities. Furthermore, there is generally more wind and higher wind speed offshore due to less surface roughness, making offshore installations more attractive since there is a bigger probability for higher electricity production. Unlike onshore, where small roads and sharp turns could make the transportation difficult, transporting the tower, as well as the turbine blades with the help of ships is less complicated (Manwell, McGowan, & Rogers, 2009, pp. 461-465; Twidell & Gaudiosi, 2009, pp. 23-26). Table 1 (Manwell, McGowan, & Rogers, 2009, pp. 461-465; Twidell & Gaudiosi, 2009, pp. 23-26) summaries the most important pros and cons with offshore wind turbines:

Table 1: Pros and cons with offshore wind turbines

Advantages	**Disadvantages**
More wind and higher wind speed.	Maintenance is harder and more expensive.
More space available for turbine installation.	The installation offshore is more difficult and expensive.
Noise and shadow flickering is not considered a huge problem offshore.	The structural foundation is more expensive.
Easier to construct large turbines.	Offshore is a more corrosive environment.
Easier to transport the necessary parts by ships, compared to road transportation onshore.	The structure is exposed to a harsher environment in higher wind speeds, as well as wave loads.

2.4. The horizontal axis wind turbine

Many different types of wind turbines have been tried throughout history and even today new designs are under research. Nevertheless, the most popular wind turbine is the horizontal axis wind turbine (HAWT), which is shown in figure 1. Lift forces drive the rotational motion created in the HAWT. Even though the HAWT has proven to be the most successful design, other types like the vertical axis wind turbine (VAWT) is considered a possible future competitor (Manwell, McGowan, & Rogers, 2009, pp. 3-9).
For this particular project, only the HAWT will be discussed and considered.

3. Literature study

In order to be fully updated on relevant research, a detailed literature study will be presented throughout this chapter. A thorough literature study has several benefits. First of all, the opportunity to base the work on already existing research can increase the efficiency, as well as reducing the risk of making the same mistakes done by other researchers. Furthermore, being able to compare the results with other researchers' results is necessary in order to validate the work.

3.1. Some aspects of high angle-of-attack flow on airfoils for wind turbine application

"Some aspects of high angle-of-attack flow on airfoils for wind turbine application" is a scientific report written in January 2001 by W.A Timmer and R.P.J.O.M van Rooij. At this point in time, little research had been done on airfoil performance at high angle of attacks. The goal of this research was therefore to investigate the difference from 0° to 360° angle of attack between the DU96-W-180 airfoil, which is thin with a sharp trailing edge and the DU97-W-300 airfoil, which is thick with a blunt trailing edge. The experiment was performed with the help of a wind tunnel located at the Delft University. The test setup, which is illustrated in figure 3 (Timmer & van Rooij, 2001), was a 1.25x1.8m wind tunnel, where the airfoils were made in steel and had a chord length of 0.25m, while the span was 1.25m (Timmer & van Rooij, 2001).

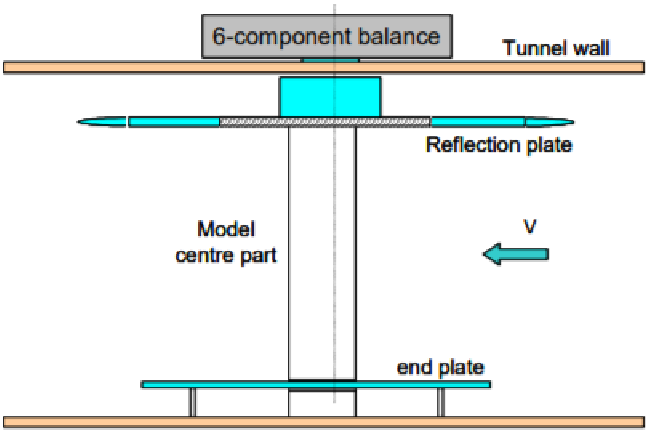


Figure 3: The test setup at Delft University

By using the test setup in figure 3 and varying the angle of attack from 0° to 360°, the performance of lift and drag for the two airfoils were obtained and is illustrated in figure 4 (Timmer & van Rooij, 2001). Also, note that the Reynolds number used for this particular experiment was $0.7 \cdot 10^6$. An important observation is how similar the drag coefficient is for the two airfoils, while the lift coefficients differs more. The highest drag coefficient came at 90° for the DU96-W-180 airfoil and had a value of 1.914 (Timmer & van Rooij, 2001).

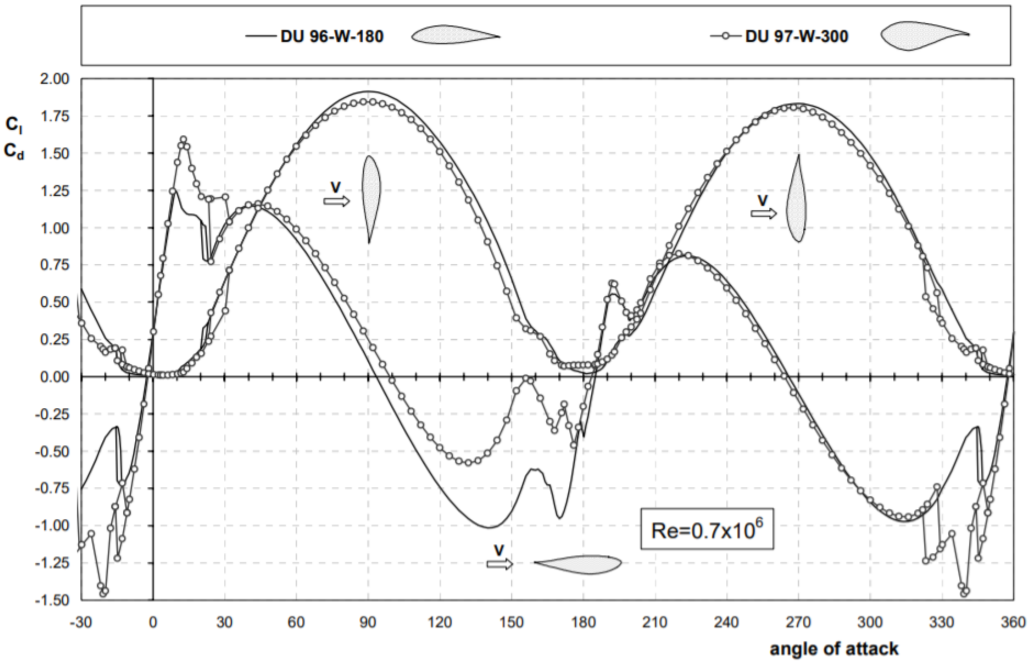


Figure 4: Lift and drag coefficients from 0° to 360° angle of attack

Furthermore, the critical lift and drag coefficients for both airfoils are seen in table 2 (Timmer & van Rooij, 2001).

Table 2: Critical results from the experiment

	DU96-W-180	DU97-W-300
C_d at 90°	1.914	1.845
C_d at 270°	1.832	1.806
C_l at 90°	0.106	0.25
C_l at 270°	-0.113	-0.116

Based on the findings, Timmer and van Rooij concluded that the biggest differences took place when the flow was separated from the leading edge, namely from 0° to 180°. Moreover, the differences in moment was considered to be very small. Finally, at approximately 156° the DU97-W-300 airfoil had close to zero lift (Timmer & van Rooij, 2001).

3.2. Vortex-induced vibrations of a DU96-W-180 airfoil at 90° angle of attack

“Vortex-induced vibrations of a DU96-W-180 airfoil at 90° angle of attack” is a scientific report from 2016, written by Niels Nørmark Sørensen and Witold Skrzypiński. The objective was to investigate the vibrations that occur on the DU96-W-180 airfoil at 90° angle of attack, in order to better evaluate the vibrations on a standstill wind turbine. Both the 2D RANS and the 3D DES CFD-simulations was done using an O-grid with a diameter of 30m. The turbulence model was a k-ω shear stress transport (SST) model. For the 2D simulation, the grid consisted of 32800 grid cells, using 256 cells parallel to the airfoil and 128 cells perpendicular to the airfoil. For the 3D simulation, the grid consisted of $12.6 \cdot 10^6$ cells, using 256 cells parallel to the airfoil, 384 cells perpendicular to the airfoil and 128 cells in the spanwise direction (Sørensen & Skrzypiński, 2014). The 3D grid is illustrated in figure 5 (Sørensen & Skrzypiński, 2014, p. 5).

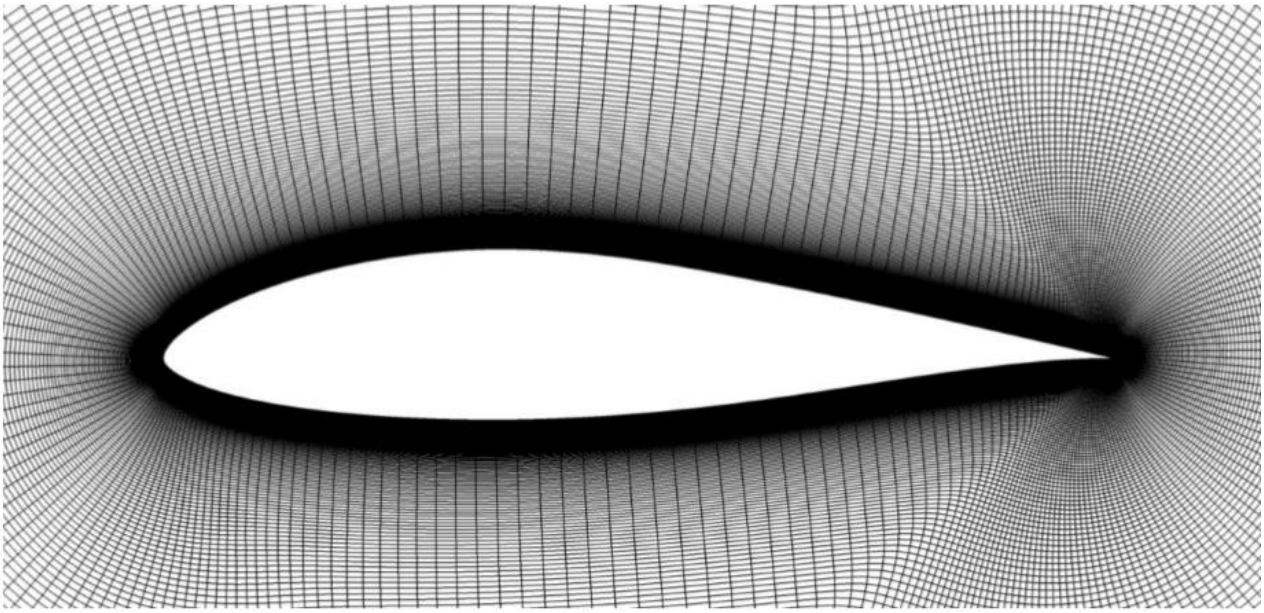


Figure 5: 3D grid of the DU96-W-180 airfoil

Furthermore, the drag coefficients for all the simulation can be seen in table 3 (Sørensen & Skrzypiński, 2014, p. 34), where 1C and 2C is one and two chord lengths in the spanwise direction respectively. Note that these values was obtained using a Reynolds number of $2 \cdot 10^6$, which is a bit higher then Timmer and van Rooij’s Reynolds number of $0.7 \cdot 10^6$ (Sørensen & Skrzypiński, 2014).

Table 3: Drag coefficients from all the simulations

Source	C_d
2D, RANS	3.1
3D, 1C, RANS	2.8
3D, 2C, RANS	2.8
3D, 1C, DES	2.3
3D, 2C, DES	2.1

Figure 6 (Sørensen & Skrzypiński, 2014, p. 16) is illustrating the vortices on the trailing and leading edge for the DU96-W-180 airfoil. Note that dark areas represents high magnitudes, while white areas represents low magnitudes of velocity (Sørensen & Skrzypiński, 2014).

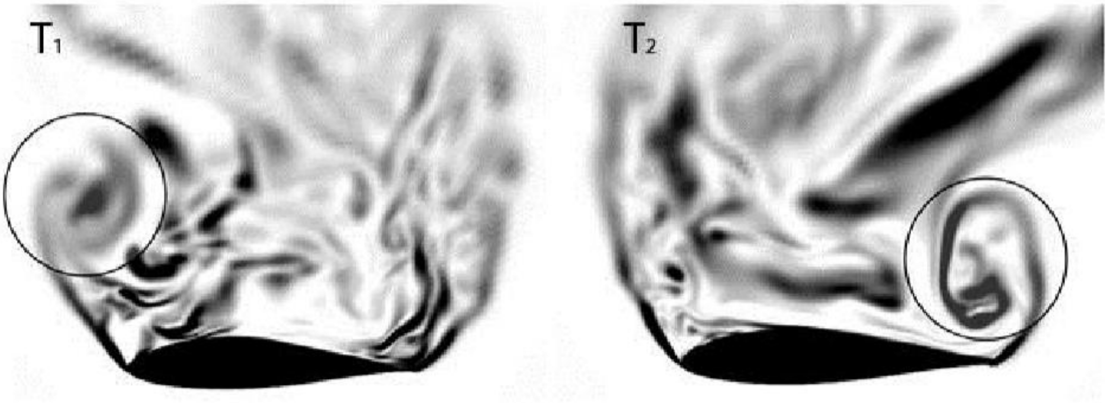


Figure 6: Vortices on the leading and trailing edges of the DU96-W-180 airfoil

3.3. CFD-simulations on a 90° DU96-W-180 airfoil

"CFD-simulations on a 90° DU96-W-180 airfoil" is a scientific report written by Stian Hjorteland in 2019 and was considered an introduction project for a future Master Thesis. The aim for this project was to perform CFD-simulation on a 90° DU96-W-180 airfoil and compare it with relevant experimental and simulated results. The grid was created based on the grid presented in the report written by Sørensen and Skrzypiński in 2014. Nevertheless, a grid refinement study was done in order to better evaluate how detailed the grid should be. By changing the number of cells as shown in table 4 (Hjorteland, 2019, p. 20), the following graph in figure 7 (Hjorteland, 2019, p. 20) was achieved (Hjorteland, 2019).

Table 4: Grid refinement study

Identity number	**Number of cells**	**Drag coefficient C_d**
2	1260	∞, Not converging.
4	2726	2.65
6	6726	1.47
8	17880	1.515
10	35700	1.54
12	48860	1.46

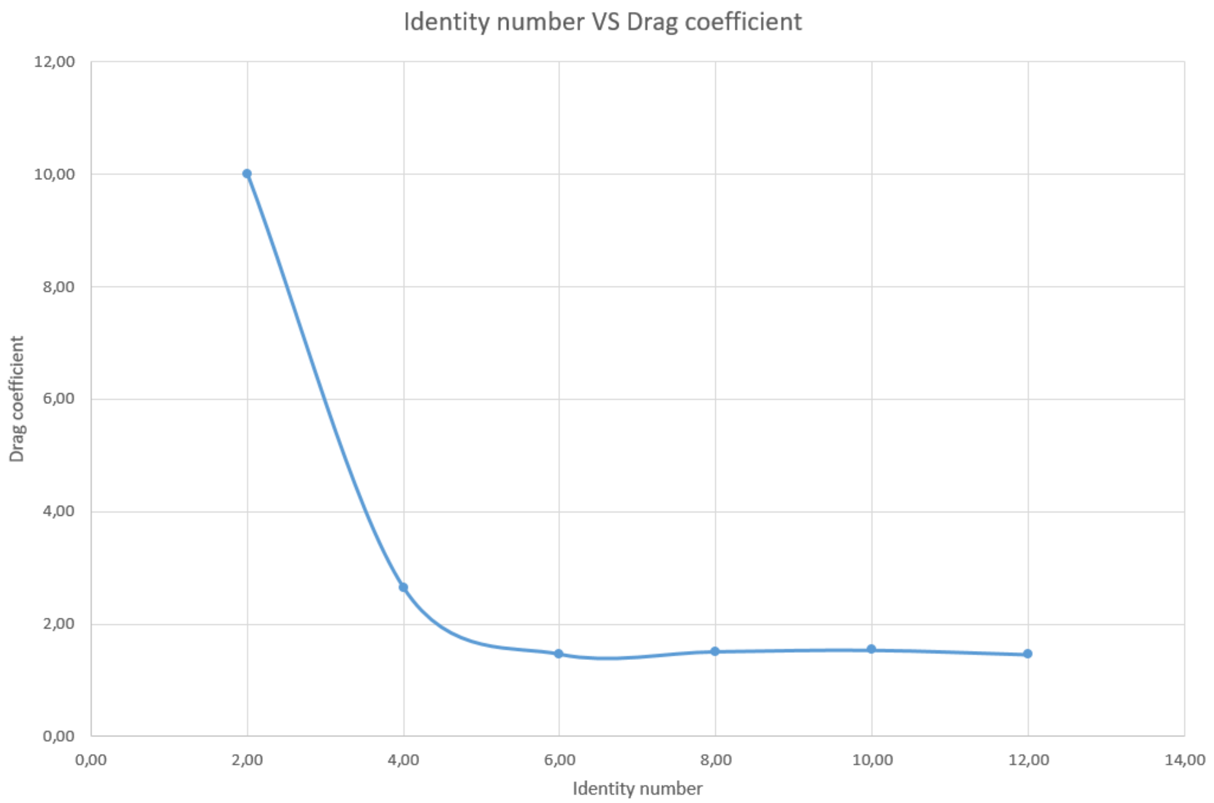


Figure 7: Grid refinement study

The report stated that a drag coefficient of 10 was selected in order to better illustrate the high non-converging value for the first grid in table 4. Moreover, by combining the results from the grid refinement study with the grid that was used by Sørensen and Skrzypiński, a grid with 35700 cells was selected as the 2D grid. For the 3D grid, the 2D grid was extruded in the span wise direction. The 3D grid was only extruded with five cells in the span wise direction, which is illustrated in figure 8 (Hjorteland, 2019, p. 21). The report stated that this was due the strength of the available computer, meaning that the simulation would not be completed before the deadline with a finer mesh.

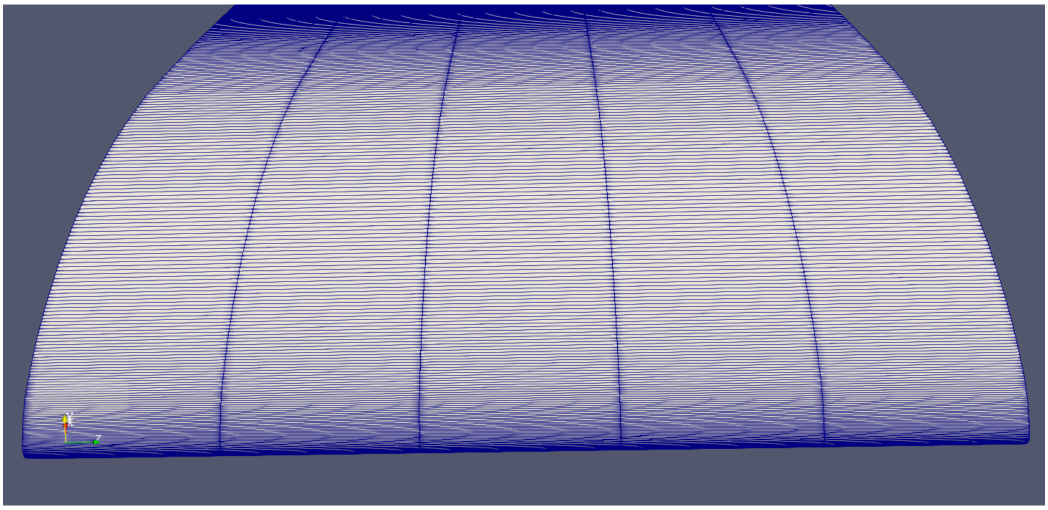


Figure 8: Illustrating the span wise mesh consisting of five cells

Nevertheless, since the purpose for this project was to get an introduction for a future master thesis, the 3D grid with only five cells in the span wise direction was kept in order to obtain valuable knowledge about meshing and simulations in 3D. The final results are presented in table 5 (Hjorteland, 2019, p. 28).

Table 5: The results achieved in the project

	Hjorteland, C_d	Sørensen and Skrzypiński, C_d	Timmer and van Rooij, C_d
Experimental			1,914
RANS (2D)	1.54	3.1	
URANS (2D)	2.9		
URANS (3D)	2		
DES (3D)	2.4	2.3	

The report concluded with the stating that the flow over the airfoil had some clear similarities with flow presented by Sørensen and Skrzypiński. Moreover, the DES (3D) simulation had a converging C_d value close to Sørensen and Skrzypiński. Furthermore, the URANS (3D) converging C_d value was close to match the experimental value from Timmer and van Rooij. Nevertheless, the report concluded that there was a high risk of inaccuracy for all the 3D simulations due to a coarse mesh in the span wise direction and stating that a finer mesh is necessary for future work (Hjorteland, 2019).

3.4. The SNL 100-03 Blade: Design Studies with Flatback Airfoils for the Sandia 100-meter Blade

“The SNL 100-03 Blade: Design Studies with Flatback Airfoils for the Sandia 100-meter Blade” is a scientific report written by D. Todd Griffith and Phillip W. Richards in 2014. As wind turbine blades increases in size, there weight is also increased. The aim of the research was to investigate the changes in blade performance and weight when using flatback airfoils. The starting point for the research was the Sandia 100-meter blade design. By adjusting and improving the geometry, blades like SNL 100-01, SNL 100-02 and finally SNL 100-03 were created. Figure 9 (Griffith & Richards, 2014, p. 8) represents the blade mass (in tons) vs the rotor radius. From figure 14, one can see that the total weight of the SNL 100-03 turbine blade is 49.519 tons (Griffith & Richards, 2014).

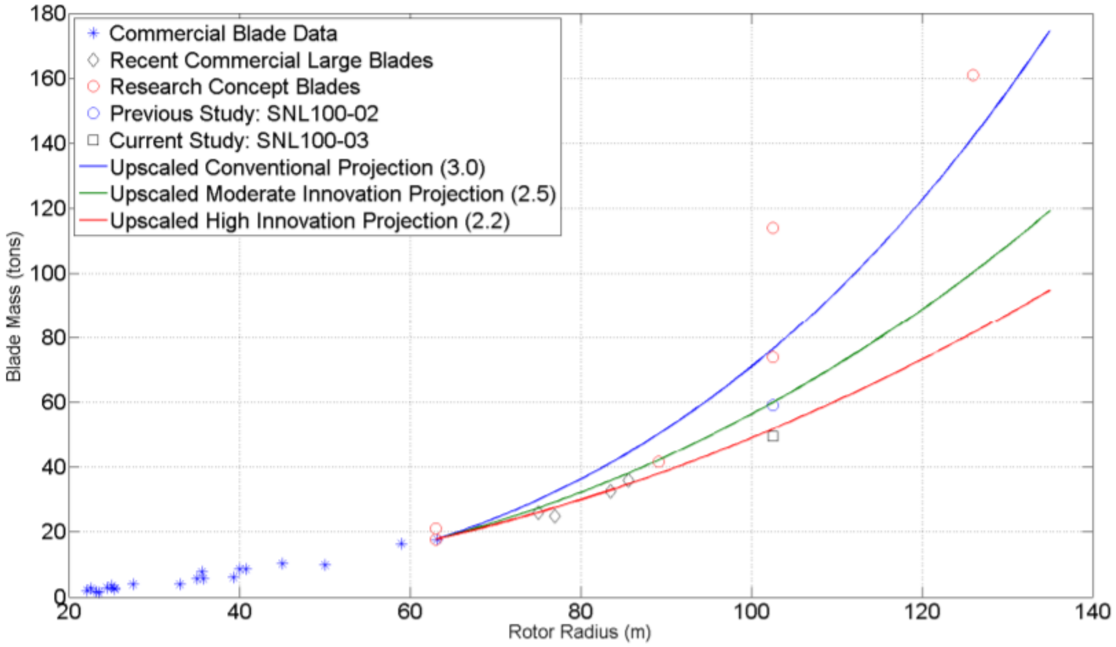


Figure 9: Showing blade mass vs rotor radius

Moving on, the details in geometry for the SNL 100-03 turbine blade can be found in table 6 (Griffith & Richards, 2014, p. 18).

Table 6: Geometry description of the SNL 100-03 blade

Station Number	**Blade Fraction**	**Chord (m)**	**Twist (deg)**	**Pitch Axis (Fraction)**	**Airfoil Description**
1	0.000	4.500	11.130	0.500	Cylinder
2	0.005	4.505	11.130	0.500	Cylinder
3	0.007	4.508	11.130	0.500	Transition (99.25%)
4	0.009	4.510	11.130	0.500	Transition (98.5%)
5	0.011	4.512	11.130	0.500	Transition (97.75%)
6	0.013	4.515	11.130	0.500	Ellipse (97%)
7	0.024	4.551	11.130	0.499	Ellipse (93.1%)
8	0.026	4.560	11.130	0.498	Interpolated
9	0.047	4.656	11.130	0.483	Interpolated
10	0.068	4.779	11.130	0.468	Interpolated
11	0.089	4.901	11.130	0.453	Interpolated
12	0.095	4.933	11.130	0.448	Interpolated
13	0.102	4.970	11.130	0.443	Interpolated
14	0.114	5.034	11.130	0.435	FB-6300-1800
15	0.146	5.155	11.130	0.410	FB-5487-1216
16	0.163	5.193	11.130	0.400	Interpolated

17	0.179	5.222	11.130	0.390	Interpolated
18	0.195	5.226	10.837	0.380	FB-4286-0802
19	0.222	5.213	10.186	0.378	Interpolated
20	0.249	5.181	9.572	0.377	FB-3423-0596
21	0.276	5.124	9.006	0.375	Interpolated
22	0.358	4.883	7.504	0.375	Interpolated
23	0.439	4.576	6.240	0.375	FB-2700-0230
24	0.520	4.225	5.132	0.375	Interpolated
25	0.602	3.825	4.147	0.375	Interpolated
26	0.667	3.472	3.444	0.375	NACA-64-618 (19%)
27	0.683	3.380	3.280	0.375	Interpolated
28	0.732	3.099	2.804	0.375	NACA-64-618
29	0.764	2.900	2.502	0.375	NACA-64-618
30	0.846	2.357	1.783	0.375	NACA-64-618
31	0.894	2.019	1.382	0.375	NACA-64-618
32	0.943	1.653	0.987	0.375	NACA-64-618
33	0.957	1.542	0.874	0.375	NACA-64-618
34	0.972	1.420	0.756	0.375	NACA-64-618
35	0.986	1.183	0.551	0.375	NACA-64-618
36	1.000	0.500	0.000	0.375	NACA-64-618

A more visual impression of the geometry for the SNL 100-03 blade is illustrated in figure 10 (Griffith & Richards, 2014, p. 19).

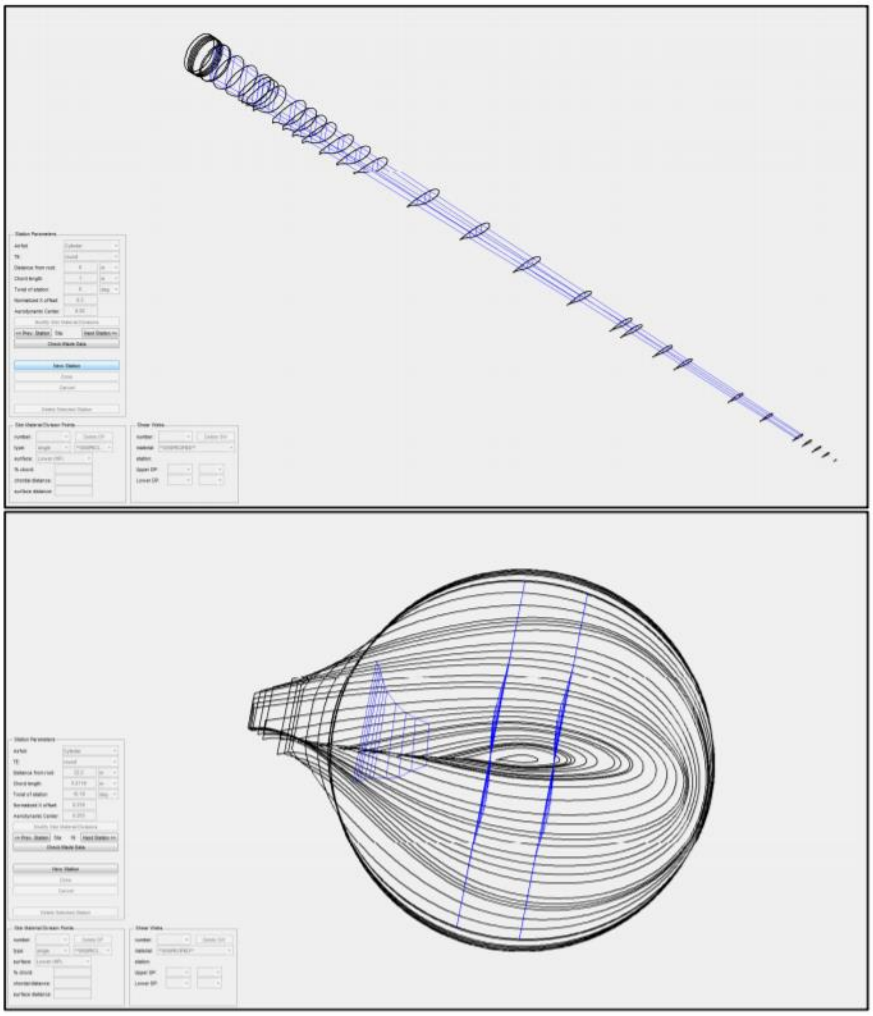


Figure 10: Illustrating the cross sections of the SNL 100-03 blade

The report concluded on successfully being able to further reducing the weight on the SNL 100-03 design. Furthermore, when reducing the weight the gravitational loading was also reduced, leading to an improvement in edge-wise fatigue resistance. In addition, some improvement on panel buckling was found when using airfoils that had a more slender planform. It is also stated that the aero elastic instability was above the operating range and that this was the case for all the designs. Recommendations for future work on aero elastic instability is also specified. Finally, it is mentioned that the SNL 100 series of turbine blades can be used as a reference for both performance studies, as well as cost studies (Griffith & Richards, The SNL 100-03 Blade: Design Studies with Flatback Airfoils for the Sandia 100-meter Blade, 2014).

4. Basic definitions and Wind Turbine theory

4.1. Basic definitions for fluid flows

The following list includes important definitions in fluid flows, which will be used frequently throughout the project. Every definition throughout chapter 4.1 are collected from the following reference: (Cengel & Cimbala, 2014, pp. 8-12).

- The No-Slip condition is a situation that arises when the fluid in motion sticks to the surface, resulting in zero velocity relative to the surface.

- Viscous flow is situations where the frictional effects are significant, typically close to a solid surface or between different fluid layers. All types of fluid flows have viscos effects to some extent. As we move away from solid surfaces, the viscous forces decreases and can in some cases be negligible compared to inertial or pressure forces.

- Inviscid flow is considered as the region where viscous flow is small enough to be neglected. These regions typically starts to appear when we move away from a solid surface. By neglecting the viscous term, calculations and analysis is greatly simplified without losing to much accuracy.

- External flow is a flow over a free surface where there is no confined space. This could typically be the flow over a plate or the flow over a wind turbine blade.

- Internal flow is flow inside a confined space. This could typically be water flow or oil flow inside a pipe.

- Incompressible flow is known to be a flow where the density remains more or less constant throughout the flow. This is a relevant approximation for many liquids.

- Compressible flow is known as a flow where the density could change due to a change in pressure. This typically the case for gases, which can experience a dramatic change in density when there is a change in pressure.

- Laminar flow is recognized by smooth and structured layers.

- Turbulent flow is characterized by an unstructured and chaotic flow with velocity fluctuations. It typically arises at high velocities.

- Transitional flow is simply a flow that is in the region between laminar and turbulent flow.

- Steady flow indicates no changes in properties like temperature and velocity per second.

- Unsteady flow is the opposite of steady flow. There is changes in properties like temperature and velocity per second.

4.2. Power output

As the wind flows over the turbine blades at a certain velocity, the kinetic energy in the wind will be converted into mechanical energy through a rotational motion. From there, the rotational motion will turn an internal shaft connected to a gearbox that speeds up the rotation. The gearbox is connected to a generator that uses the mechanical energy from the rotational motion to produce electricity (Awea, 2020).

Large wind turbines produce more energy and as described in chapter 2.2, investing in large wind turbines is getting cheaper, leading to an increase in both the amount of newly installed turbines, as well as turbine size.

When looking at the mathematical expression for power output from wind turbines, one can better see the need for the large sized blades. Equation (1) to (8) with all related theory is gathered from the following reference: (Jain, 2011, pp. 10-19). First of all, the amount of energy in the wind can be defined as kinetic energy:

$$E_{kin} = \frac{1}{2}mv^2 \tag{1}$$

Where m is defined as the wind mass, while v is defined as the wind velocity. Since the wind is constantly moving, an expression for the energy moving past the wind turbine at a certain point in time is:

$$\dot{E}_{kin} = \frac{1}{2}\dot{m}v^2 \tag{2}$$

Where $\dot{E}_{kin}$ is energy per second and $\dot{m}$ is mass per second. The amount of air moving past the wind turbine is treated as a cylinder where the disc will have the same radius as the length of the wind turbine blade, making mass per second the same as:

$$\dot{m} = \rho V \tag{3}$$

Where ρ is the air density and V is the volume of air moving past the wind turbine. Since volume is defined as area multiplied with a certain distance, the expression can be redefined as:

$$\dot{m} = \rho A v \tag{4}$$

Where A is the rotational area of the wind turbine and the wind velocity v is the length of the cylinder. Substituting equation (4) into equation (2), we obtain the following expression:

$$\dot{E}_{kin} = \frac{1}{2}\rho A v^3 \tag{5}$$

Energy per second is the same as power, while the area of the rotational disc is $A = \pi r^2$. Substituting the new expression for A into equation (5) results in the final expression for the available energy in wind:

$$P = \frac{1}{2}\rho\pi r^2 v^3 \tag{6}$$

Where P is the power available in the wind and r is the radius of the rotational area, which is the same as the length of the turbine blade. Even though equation (6) represents the amount of available power in the wind, it has been proven hard to extract all the energy. The power coefficient C_p, representing the maximum efficiency for a three bladed horizontal axis wind turbine was calculated by Albert Betz, and estimated to be 59.3%.

$$C_p = \frac{\text{Max power extracted}}{\text{Power available}} = \frac{16}{27} = 0.593 \tag{7}$$

However, the Betz limit is a theoretical value, which does not represent the efficiency of today's wind turbines. The actual efficiency of wind turbines is closer to a value between 25 - 45% (Windpower Engineering & Development, 2010), depending on factors like design and technology. Nevertheless, the Betz limit is an important factor as it keeps reminding us that wind turbines have the potential of becoming more power efficient in the future, as there is an upper limit to aim towards.
By multiplying equation (6) with any given power coefficient, the actual power output from the wind turbine is known to be:

$$P = \frac{1}{2} C_p \rho \pi r^2 v^3 \tag{8}$$

By taking a closer look at equation (8), one can see that a doubling of the blade length will increase the power output by a factor of four, making this one of the key reasons for wind turbines increasing in size. Furthermore, by doubling the wind speed, the power output will increase by a factor of eight. Unfortunately, an increase in wind speed does not always correspond to an increase in power output. The turbine itself has limitations of the amount of power it can generate as illustrated in figure 11 (Jain, 2011, p. 70). Figure 11 shows the difference in power output for pitch and passive stall regulated turbines. The graph is divided into three sections (Jain, 2011, pp. 68-71):

- Cut-in speed is the minimum wind speed needed for the turbine to start generating power.

- Rated speed is the maximum power output for pitch regulated turbines.

- Cut-out speed is the wind speed where the turbine stops producing power due to safety reasons as the mechanical parts inside the turbine is not able to handle the loads for wind speeds above this limit.

As the wind speed increases, the pitch-regulated turbine increases the pitch, making the angle of attack smaller. By adjusting the angle of attack after reaching the rated speed, the pitch-regulated turbine is able to maintain a stable power output until it reaches the cut-out speed. Unlike the pitch-regulated turbine, the passive stall-regulated turbine has a fixed pitch on the average wind speed at the location of interest. Since the pitch is not changed, the power output varies for all wind speeds, making it likely to have a more unstable power output. The pith-regulated turbine is a popular choice as it gives a stable and reliable power production for a larger wind speed interval (Jain, 2011, pp. 68-71).

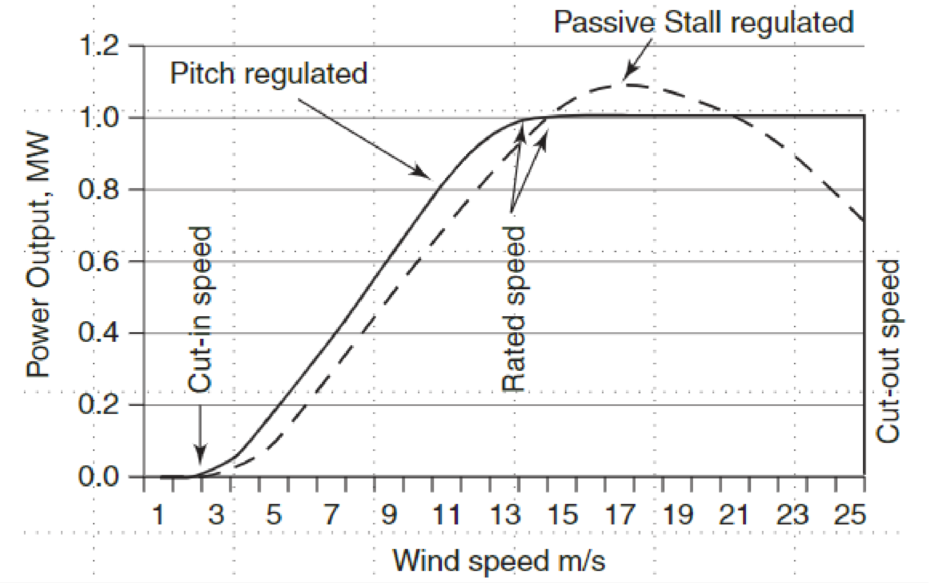


Figure 11: Power output for pitch and stall regulated wind turbines

4.3. Reynolds number

The Reynolds number is a non-dimensional parameter used for describing the characteristics of fluid flow. Typically, low Reynolds numbers results in laminar flow, while high Reynolds numbers results in a turbulent flow. The Reynolds number is defined as (Cengel & Cimbala, 2014, p. 11):

$$Re = \frac{UL}{\nu} = \frac{\rho UL}{\mu} \tag{9}$$

Where U is the velocity of the flow, L is the characteristic length, ν is the kinematic viscosity and µ is the fluid viscosity (Manwell, McGowan, & Rogers, 2009, p. 103). For an airfoil, the characteristic length is typically the chord length.
From figure 12 (Burton, Jenkins, Sharpe, & Bossanyi, 2011, p. 125), one can imagine the solid surface as the top surface of an airfoil. The flow starts out as laminar before reaching the airfoil. At some distance, the flow changes into a turbulent flow. Locating the point where the flow changes is done by introducing a new characteristic length l, which is the critical distance from the leading edge to the point where the turbulent flow starts.

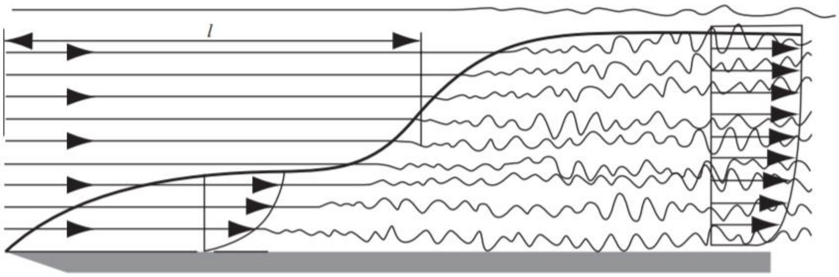


Figure 12: Laminar, transitional and turbulent boundary layers

Based on this, a new expression for the Reynolds number at the flow transition is (Burton, Jenkins, Sharpe, & Bossanyi, 2011, p. 125):

$$Re_{crit} = \frac{\rho U l}{\mu} \tag{10}$$

4.4. Aerodynamics of wind turbines

Wind turbine blades or designed to use the motion of air to produce lift, which will result in a rotational motion, developing mechanical power. The turbine blade can be treated as multiple airfoils with different designs in order to optimize the desired aerodynamic performance, the assumed airfoil properties, the maximum desired rotor power and strength considerations (Manwell, McGowan, & Rogers, 2009, pp. 101-104). An example of how the cross section of a wind turbine blade is changing as one move towards the tip can be seen in figure 13 (Sheibania & Akbari, 2015, p. 3776).

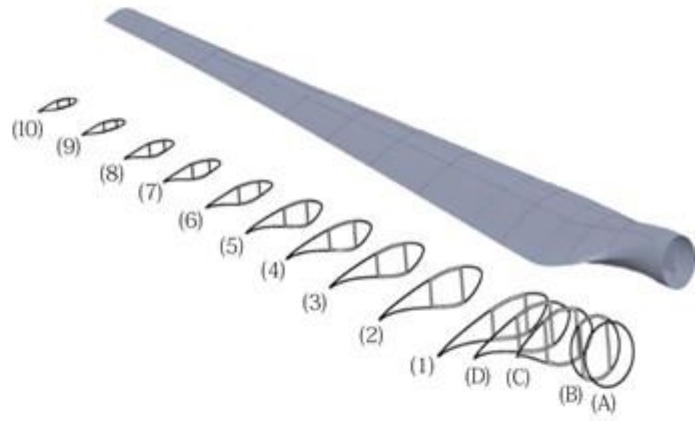


Figure 13: Different cross sections of a wind turbine blade

Based on this, a better understand of the aerodynamics of a wind turbine blade can be achieved by investigating a single airfoil. Some important names and definitions are illustrated in figure 14 (Patrol, 2006) and will be used frequently throughout the project.

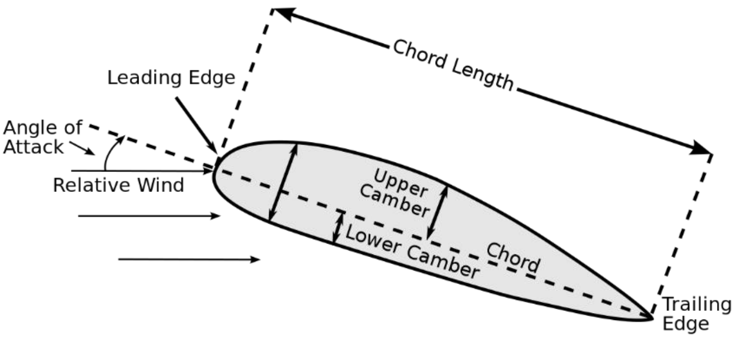


Figure 14: Airfoil nomenclature

According to Bernoulli’s principle, when assuming frictionless flow, the sum of static pressure and dynamic pressure are constant, leading to the following equation (Manwell, McGowan, & Rogers, 2009, p. 104):

$$p + \frac{1}{2}\rho U^2 = constant \tag{11}$$

Where p is defined as the static pressure and U can be considered as the velocity of the flow along the airfoil. The principle states that an increase in speed will lead to a decrease in pressure, while a decrease in speed will lead to an increase in pressure. When the flow reaches the leading edge of an airfoil, the flow separates into two different streams. The idea is that the flow, which was separated at the leading edge, will reach the trailing edge at the same time. As illustrated in figure 15 (Aviator, 2017), an unsymmetrical airfoil will have a longer surface length on the top half compared to the bottom half, making the flow on the top half travel faster and thereby creating the difference in pressure. This is a simplified, but helpful way of understanding how lift is generated, especially for unsymmetrical airfoils (Jain, 2011, p. 14; Manwell, McGowan, & Rogers, 2009, p. 104).

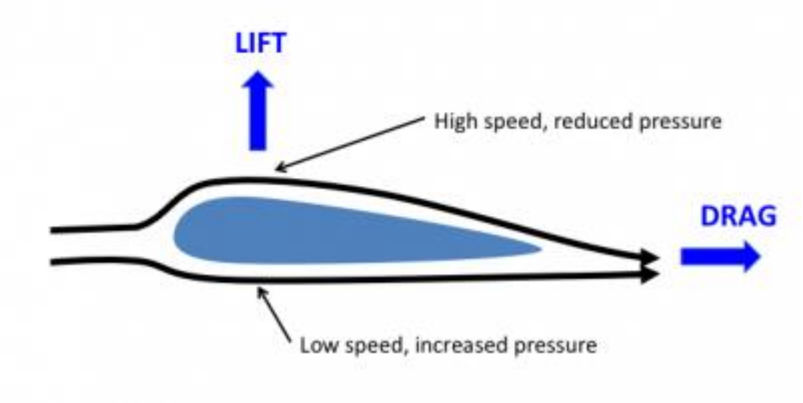


Figure 15: Flow over an airfoil

Nevertheless, a more accurate description of lift can be described by circulation, as symmetrical airfoils also produce lift. Circulation is defined as the summation of velocity around as closed path. Martin Wilhelm Kutta, who was an aerodynamicist from Germany, developed an idea saying that circulation is created on a sharp trailing edge. Based on his findings, a real fluid flow will leave the trailing edge tangentially and in order for this to happen, a clockwise circulation (assuming flow from left to right) must exist in order to move the trailing stagnation point to the trailing edge itself. When investigating figure 16 (Jain, 2011, p. 53), the inviscid flow to the left fails to have the stagnation point on the trailing edge. However, when adding circulation in the middle, the stagnation point is moved to the trailing edge. Another important detail is the upstream flow on the inviscid flow to the left, being parallel to the downstream flow. However, as circulation is included, the final result indicates that the upstream flow is no longer parallel to the downstream flow (Jain, 2011, pp. 51-54).

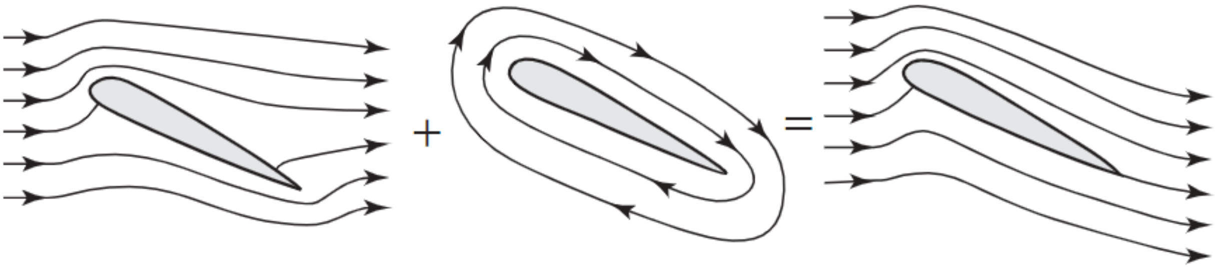


Figure 16: The flow over an airfoil consisting of inviscid flow and circulation

Equation (12) to (17), including the theoretical explanations, are all gathered from the following references: (Jain, 2011, pp. 51-54; Manwell, McGowan, & Rogers, 2009, p. 108). For circulation with a strength of Γ, the lift force L can be described by:

$$L = \rho v_0 \Gamma \tag{12}$$

Where v_0 is the free stream wind speed. By considering equation (12), if the circulation Γ is zero, then the lift force L must be zero. In other words, for lift to exist, circulation must also exist. By considering any type of body, the circulation is defined as:

$$\Gamma = \oint \vec{v} \cdot d\vec{s} \tag{13}$$

For simplicity, the closed path subjected to lift is considered to be a thin plate where the integral in equation (13) will sum up the dot product of the velocity around the closed path. The lift coefficients for a thin plate is proven to have values complementary to theoretical lift coefficients for symmetrical airfoils under ideal conditions, as they indeed have similar geometry. Solving equation (13) for the geometry of a thin plate results in the following expression for circulation:

$$\Gamma = \pi v_0 c \sin \alpha \tag{14}$$

Where c is the chord length of the plate and α is the angle of attack. Substituting equation (14) into equation (12), we obtain the following expression for lift per unit length:

$$L = \rho v_0 (\pi v_0 c \sin \alpha) \tag{15}$$

Since equation (15) represents lift per unit length of a plate, one can multiply the equation with the length l of the plate. Continuing by introducing the definition for the lift coefficient C_l:

$$C_l = \frac{L}{\frac{1}{2} \rho {v_0}^2 A} \tag{16}$$

Where A is the surface area, defined as $A = c \cdot l$. Combining the obtained information results in:

$$C_l = \frac{\rho {v_0}^2 \pi c l \sin \alpha}{\frac{1}{2} \rho {v_0}^2 c l} = 2\pi \sin \alpha \tag{17}$$

As mentioned above, the final expression for the lift coefficient of a thin plate represented in equation (17), has proven to have similar theoretical lift coefficients for symmetrical airfoils. This is especially the case for angles of attack between -15° to 15°. To further illustrate this, the graph in figure 17 (Manwell, McGowan, & Rogers, 2009, p. 108) illustrates the comparison between a thin plate and a symmetrical NACA 0012 airfoil.

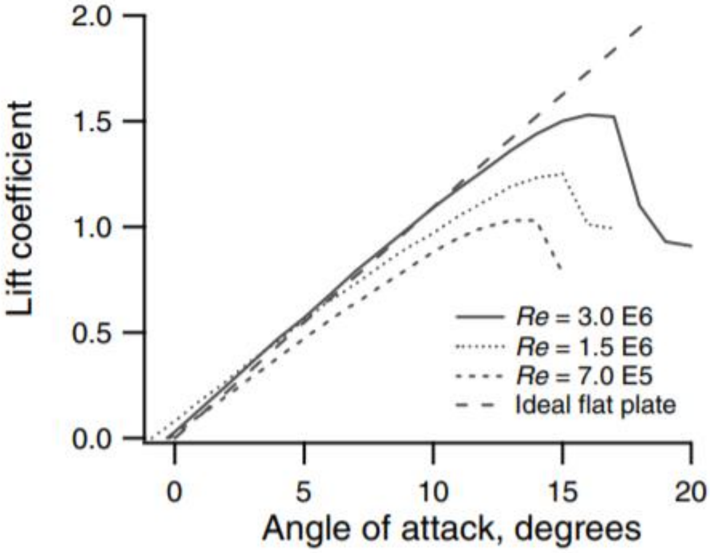


Figure 17: Comparison between a thin plate and a symmetrical airfoil for different Reynolds numbers

The drag force F_d is basically the force pushing and object in the direction of the flow. On figure 18 (Jain, 2011, p. 60), an illustration of flow perpendicular on a rectangular plate creating the drag force F_d, pushing the plate in the flow direction (Jain, 2011, pp. 59-60).

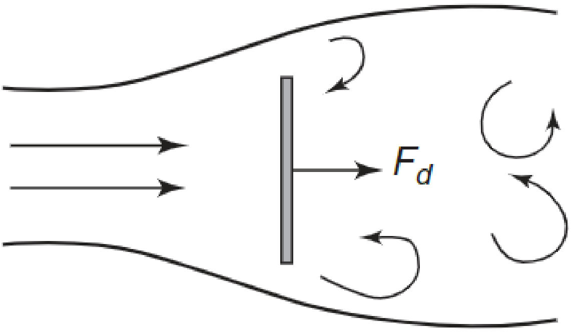


Figure 18: The drag force F_d pushing the flat plate in the flow direction

The drag coefficient C_d is a dimensionless constant and can be considered as the most common way of representing drag. It is expressed by (Jain, 2011, pp. 59-60):

$$C_d = \frac{F_d}{\frac{1}{2}\rho A v_0^2} \tag{18}$$

4.5. Extreme loading conditions

The International Electrotechnical Commission (IEC) are the creators of the *"IEC 61400 - 1 Wind Turbines - Part 1: Design Requirements"*. This international standard includes among other technical details, the different extreme wind speeds that are used for the different design classes. There is an IEC standard specifically for offshore wind turbines, which is called *"IEC 61400 – 3"*. However, this standard is not available for free, which is why this project is built on the *"IEC 61400 – 1"*. When discussing extreme loading conditions caused by high wind velocities on a wind turbine, there are three main cases (Twidell & Gaudiosi, 2009, p. 164; The International Electrotechnical Comission, 2005):

- The mean value of the extreme wind speed within an hour.
- The mean value of the extreme wind speed within an interval of 10 minutes.
- The value of an incident extreme wind speed within a short time interval.

As for this project, only the mean value of the extreme wind speed within an interval of 10 minutes is considered. IEC distinguish between four different design classes that are listed in table 7 (The International Electrotechnical Comission, 2005, p. 24).

Table 7: Extreme wind speed for different wind turbine classes

Wind Turbine Class	I	II	III	S
V_{ref} (m/s)	50	42.5	37.5	X
A I_{ref}(-)	0,16	0,16	0,16	X
B I_{ref}(-)	0,14	0,14	0,14	X
C I_{ref}(-)	0,12	0,12	0,12	X

Where V_{ref} is the average extreme wind speed within an interval of 10 minutes, I_{ref} is the expected turbulence intensity at hub height and A, B, and C are the category for higher, medium or lower turbulence characteristics. Note that wind turbine class S is devoted for offshore wind turbines. The values for this class should be obtained by investigating the desired offshore area (The International Electrotechnical Comission, 2005, p. 24).

4.5.1. Extreme Wind Speed model

Considering the steady extreme wind model, the V_{e50} extreme wind speed, having a recurrence period of 50 years can be estimated as follows (The International Electrotechnical Comission, 2005, p. 27):

$$V_{e50}(z) = 1.4V_{ref}\left(\frac{z}{z_{hub}}\right)^{0.11} \tag{19}$$

While the V_{e1} extreme wind speed, having a recurrence period of 1 year can be estimated by using the newly obtained expression for V_{e50} (The International Electrotechnical Comission, 2005, p. 28):

$$V_{e1}(z) = 0.8V_{e50}(z) \tag{20}$$

Furthermore, the turbulent extreme wind speed model for a recurrence period of 1 year, as well as 50 years can be determined as (The International Electrotechnical Comission, 2005, p. 28):

$$V_{50}(z) = V_{ref}\left(\frac{z}{z_{hub}}\right)^{0.11} \tag{21}$$

$$V_1(z) = 0.8V_{50}(z) \tag{22}$$

Where equation (19), (20), (21) and (22) are all a function of the height z, while V_{ref} is given in table 7 and z_{hub} is the hub height (The International Electrotechnical Comission, 2005, pp. 27-28).

4.5.2. Extreme Turbulence model

The extreme turbulence model builds on a normal wind profile model defined as (The International Electrotechnical Comission, 2005, p. 26):

$$V(z) = V_{hub}\left(\frac{z}{z_{hub}}\right)^{\alpha} \tag{23}$$

Where α usually is assumed to be 0.2 and V_{hub} is the wind speed at hub height. The turbulence model can then be defined as (The International Electrotechnical Comission, 2005, p. 29):

$$\sigma_1 = c\, I_{ref}\left(0.072\left(\frac{V_{ave}}{c}+3\right)\left(\frac{V_{hub}}{c}-4\right)+10\right);\ \ c = 2\,m/s \tag{24}$$

Where σ_1 is the turbulence standard deviation and V_{ave} is the annual average wind speed at hub height (The International Electrotechnical Comission, 2005, p. 29).

4.5.3. Extreme direction change

During extreme weather, most wind turbines will be on standstill. The turbine blades are usually pitched in order to adjust itself based on the wind direction in order to reduce specific loadings. A rapid change in direction can cause the extreme wind speed to hit the blade at high angle of attacks. The extreme direction change of the wind speed has a defined value of θ_e given as (The International Electrotechnical Comission, 2005, p. 29):

$$\theta_e = \pm\tan^{-1}\left(\frac{\sigma_1}{V_{hub}\left(1+0.1\left(\frac{D}{\Lambda_1}\right)\right)}\right) \tag{25}$$

Where the extreme transient $\theta(t)$ direction change is given as (The International Electrotechnical Comission, 2005, p. 29):

$$\theta(t) = 0° \qquad for \quad t < 0 \tag{26}$$

$$\theta(t) = \pm 0.5\theta_e \left(1 - \cos\frac{\pi t}{T}\right) \qquad for \;\; 0 \leq t \leq T \tag{27}$$

$$\theta(t) = \theta_e \qquad for \quad t > T \tag{28}$$

Where D is the rotor diameter, Λ_1 is treated as the turbulence scale parameter, t is time and T is the characteristic time for the gusts (The International Electrotechnical Comission, 2005, p. 29).

5. Governing equations and theory

5.1. Computational Fluid Dynamics

Calculations or experimental setups (sometimes a combination of both) is typically the way of solving engineering problems about fluid flow. If the structure is large, typically a 100m long turbine blade, the experimental setup can be both expensive and difficult. Moreover, solving the aerodynamics for the 100m turbine blade is extremely complex and is close to impossible without the aid of computers. This is where Computational Fluid Dynamics (CFD) have the potential to contribute in solving complex setups. CFD is defined as solving equations of fluid flow with the help of computers.

It is fair to say that the results from a CFD simulation is strongly dependent on the knowledge of the user. Being able to understand the outcome of a CFD simulation is key when deciding on whether the result is valid or not. The validation part can be done by comparing CFD simulations with experimental setups in order to validate the computational setup, making the CFD simulations more valuable (Cengel & Cimbala, 2014, p. 880). The rest of this chapter is devoted to presenting the most important and fundamental theory for CFD.

5.2. Navier Stokes equations

The continuity equation, also called the conservation of mass, for a compressible fluid can be described as (Cengel & Cimbala, 2014, pp. 444 - 445):

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot \left(\rho \vec{V}\right) = 0 \tag{29}$$

Where ρ is the density, t is the time, $\vec{V}$ is the velocity vector and $\vec{\nabla}$ is the divergence of a vector field, in this case $\vec{V}$. However, the continuity equation is often simplified in calculations by assuming incompressible flow. By assuming incompressible flow, there is no change in density at any point in time or space, making the time derivative in equation (29) zero, as well as taking the density outside of the divergence term. From there, the equation narrows down to (Cengel & Cimbala, 2014, pp. 444 - 445):

$$\vec{\nabla} \cdot \vec{V} = 0 \tag{30}$$

Expanding the vector in a three dimensional system, the final equation of the continuity equation in Cartesian coordinates when assuming incompressible flow is (Cengel & Cimbala, 2014, pp. 444 - 445):

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{31}$$

The Navier Stokes equation, which is considered a cornerstone in fluid mechanics by many researchers, is given in equation (32), when assuming incompressible flow (Cengel & Cimbala, 2014, pp. 466 - 468). Also, the gravity term is neglected.

$$\rho \frac{D\vec{V}}{Dt} = -\vec{\nabla} P + \mu \nabla^2 \vec{V} \tag{32}$$

Where $\frac{D}{Dt}$ represents the material derivative of the fluid, P is the pressure, μ is the dynamic viscosity and ∇^2 is the Laplacian operator, which is defined as (Cengel & Cimbala, 2014, pp. 466 - 468):

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \tag{33}$$

Expanding equation (32) in Cartesian coordinates results in the following three equations (Cengel & Cimbala, 2014, pp. 466 - 468):

$$\rho\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z}\right) = -\frac{\partial P}{\partial x} + \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) \tag{34}$$

$$\rho\left(\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z}\right) = -\frac{\partial P}{\partial y} + \mu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right) \tag{35}$$

$$\rho\left(\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z}\right) = -\frac{\partial P}{\partial z} + \mu\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right) \tag{36}$$

Where $u, v\ and\ w$ are velocity components. Equation (34), (35) and (36) have four unknowns. In order to obtain a solvable set of equations one has to include equation (31), the continuity equation (Cengel & Cimbala, 2014, pp. 466 - 468).

5.3. Turbulence modeling

Turbulence modeling is considered one of the key elements in CFD. As discussed in chapter 5.1, it is characterized by unstructured flow. Moreover, being treated as both time dependent and three-dimensional, turbulence is challenging to simulate correctly. Introducing turbulence models, some approximations and simplifications are involved in order to predict the turbulent flow. There are several different turbulence models with different strengths and weaknesses, including factors like simulation memory, accuracy, geometry and Reynolds number (Sadrehaghighi, 2020).

5.3.1. Reynolds Averaged Navier Stokes

The Reynolds Averaged Navier Stokes (RANS) is a model based on assuming that the turbulence can be broken down into time-averaged and fluctuating components. RANS is derived using the continuity and Navier Stokes equations described in chapter 6.2. The continuity equation is linear with respect to velocity, resulting in no changes. The Navier Stokes equation on the other hand does change. This is a non-linear equation and will be left with fluctuating components. A new term called Reynolds stresses $\overline{u_i' u_j'}$ is emerging to the Navier Stokes equation, resulting in the following equation for momentum (Sadrehaghighi, 2020):

$$\frac{\partial(\rho\bar{u}_i)}{\partial t}+\frac{\partial\left(\rho\bar{u}_i\bar{u}_j\right)}{\partial x_i} = -\frac{\partial\bar{p}}{\partial x_i}+\frac{\partial\left(\bar{\tau}_{ij}-\rho\overline{u_i'u_j'}\right)}{\partial x_j} \tag{37}$$

The partial time derivative in equation (37) can be removed for time averaging, which is integration over time. Moreover, $\bar{\tau}_{ij}$ is defined as the viscous stresses and can be rewritten as (Sadrehaghighi, 2020):

$$\bar{\tau}_{ij} = \mu\left(\frac{\partial\bar{u}_i}{\partial x_j}+\frac{\partial\bar{u}_j}{\partial x_i}\right) \tag{38}$$

The Reynolds stresses $\rho\overline{u_i'u_j'}$ in equation (38) must be considered as new unknowns. These unknowns are called the closure problem and can be treated as a system with more unknowns than available equations. Boussinesq suggested a possible solution to this when introducing the eddy viscosity v_t. In order to achieve mathematical closure, a relation between the mean flow properties and the Reynolds stresses must be achieved. The Reynolds stresses can be rearranged into (Giljarhus, 2019):

$$\rho\overline{u_i'u_j'} = -\mu_t\left[\frac{\partial\bar{u}_i}{\partial x_j}+\frac{\partial\bar{u}_j}{\partial x_i}\right]-\frac{2}{3}\delta_{ij}\rho k \tag{39}$$

Where the turbulent kinetic energy k is defined as (Sadrehaghighi, 2020):

$$k = \frac{1}{2}\left(\overline{u'^2}+\overline{v'^2}+\overline{w'^2}\right) \tag{40}$$

Where the bar on top of the variables symbolizes the mean value, u is defined as velocity, p is defined as pressure and δ_{ij} is the Kronecker delta (Sadrehaghighi, 2020).
For this particular project, the use of turbulence models will ensure the mathematical closure in the RANS equations. There are several turbulence models to choose from and they all have pros and cons. For this particular project, the Spalart Allmaras model was selected. This model is thoroughly described in chapter 5.3.3. In figure 19 (Sadrehaghighi, 2020, p. 27), an overview of different turbulence models are illustrated.

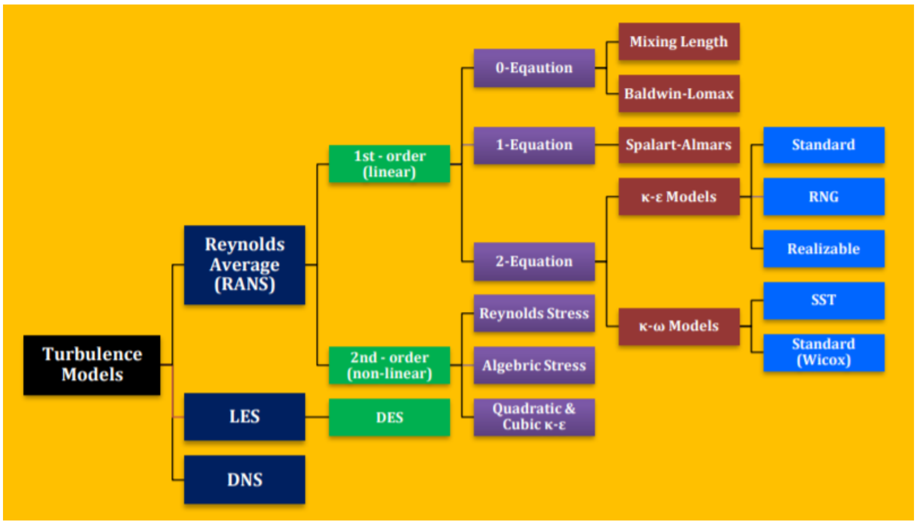


Figure 19: Overview of turbulence models

5.3.2. Unsteady Reynolds Average Navier Stokes

The most important theory regarding RANS is covered in chapter 5.3.1. Nevertheless, Unsteady Reynolds Average Navier Stokes, also called URANS represents a transient solution and differs from RANS in some areas. First of all, unlike RANS, the unsteady term in equation (37) is kept. In this case, the partial time derivative is not removed, making the unsteady flow a function of both time and space. As mentioned in chapter 6.3.1, the idea is to break down the turbulence into time averaged and fluctuating components, which are given in equation (41) (Sadrehaghighi, 2020).

$$\overline{U} = \frac{1}{\tau}\int_0^{\tau} U(\tau)d\tau \qquad , \qquad U = \overline{U}+u'' \tag{41}$$

$$\overline{U}_i = \overline{U}_i(x,y,z,\tau) \qquad , \qquad \overline{u_i''u_j''} = \overline{u_i''u_j''}(x,y,z,\tau) \tag{42}$$

Where $\overline{U}$ is the mean flow, U is the flow and u'' is the turbulent fluctuation. Based on this, the mean flow properties changes with time, unlike RANS where this is constant (Sadrehaghighi, 2020). Furthermore, the difference is clearly illustrated in figure 20 (Sadrehaghighi, 2020, p. 24).

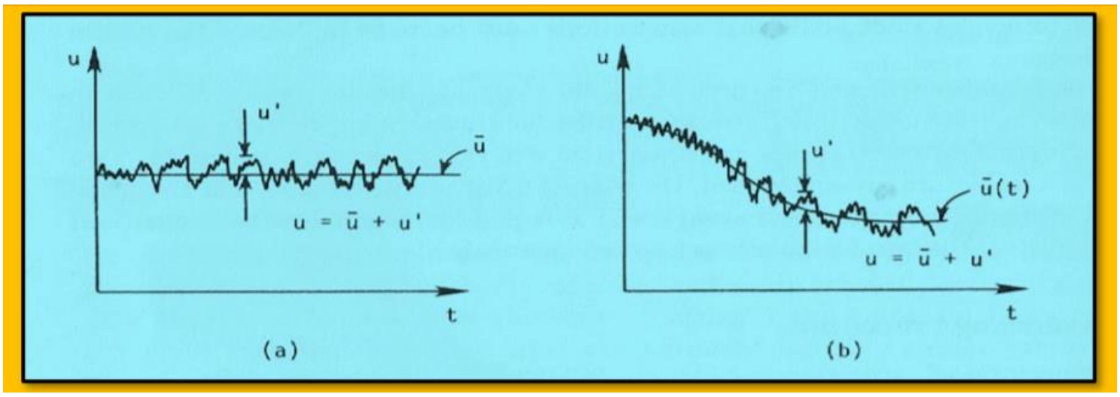


Figure 20: Illustration of steady flow (a) and unsteady flow (b)

5.3.3. Spalart-Allmaras

The Spalart-Allmaras turbulence model is known to be a reliable model when working with aerodynamic problems. In addition, as illustrated in figure 19, the Spalart-Allmaras is a one-equation turbulence model. This is a great advantage when working with large simulation files, as it is less memory intensive. This was the main argument when selecting a turbulence model, since this project is dealing with a large wind turbine blade that requires a large domain and a detailed mesh (Sadrehaghighi, 2020).
The equation for Spalart-Allmaras can be given as follows (NASA, 2019):

$$\frac{\partial \hat{v}}{\partial t} + u_j\frac{\partial \hat{v}}{\partial x_j} = c_{b1}(1-f_{t2})\hat{S}\hat{v} - \left[c_{w1}f_w - \frac{c_{b1}}{k^2}f_{t2}\right]\left(\frac{\hat{v}}{d}\right)^2 + \frac{1}{\sigma}\left[\frac{\partial}{\partial x_j}\left((v-\hat{v})\frac{\partial \hat{v}}{\partial x_j}\right) + c_{b2}\frac{\partial \hat{v}}{\partial x_i}\frac{\partial \hat{v}}{\partial x_i}\right] \tag{43}$$

Where the turbulence viscosity v_t is calculated from (CFD-Online, 2015):

$$v_t = \hat{v} f_{v1} \tag{44}$$

Mathematical relations for the two formulas above are given from equation (45) to (54) (NASA, 2019):

$$f_{v1} = \frac{\chi^3}{\chi^3 + {C_{v1}}^3} \tag{45}$$

$$\chi = \frac{\hat{v}}{v} \tag{46}$$

$$\hat{S} = \Omega + \frac{\hat{v}}{k^2 d^2} f_{v2} \tag{47}$$

$$\Omega = \sqrt{2 W_{ij} W_{ij}} \tag{48}$$

$$W_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right) \tag{49}$$

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}} \tag{50}$$

$$f_w = g \left[\frac{1 + {c_{w3}}^6}{g^6 + {c_{w3}}^6}\right]^{\frac{1}{6}} \tag{51}$$

$$g = r + c_{w2}(r^6 - r) \tag{52}$$

$$r = min\left[\frac{\hat{v}}{\hat{S} k^2 d^2}, 10\right] \tag{53}$$

$$f_{t2} = {c_{t3}}^{(c_{t4}\chi^2)} \tag{54}$$

Equation (43) to (54) includes some constants that are listed below (NASA, 2019):

$c_{b1} = 0.1335$
$\sigma = \frac{2}{3}$
$c_{b2} = 0.622$
$k = 0.41$
$c_{w2} = 0.3$
$c_{w3} = 2$
$c_{v1} = 7.1$
$c_{t3} = 1.2$
$c_{t4} = 0.5$
$c_{w1} = \frac{c_{b1}}{k^2} + \frac{1 + c_{b2}}{\sigma}$

Where $\hat{v}$ is the viscosity variable, d is defined as the distance from the closest surface, v is the molecular kinematic viscosity and μ is the molecular dynamic viscosity (NASA, 2019). The distance from the closest surface is not necessarily along a grid line or at a specific grid point. It is defined as the shortest distance from the point of interest to the solid wall. Two example are given in figure 21 (NASA, 2019).

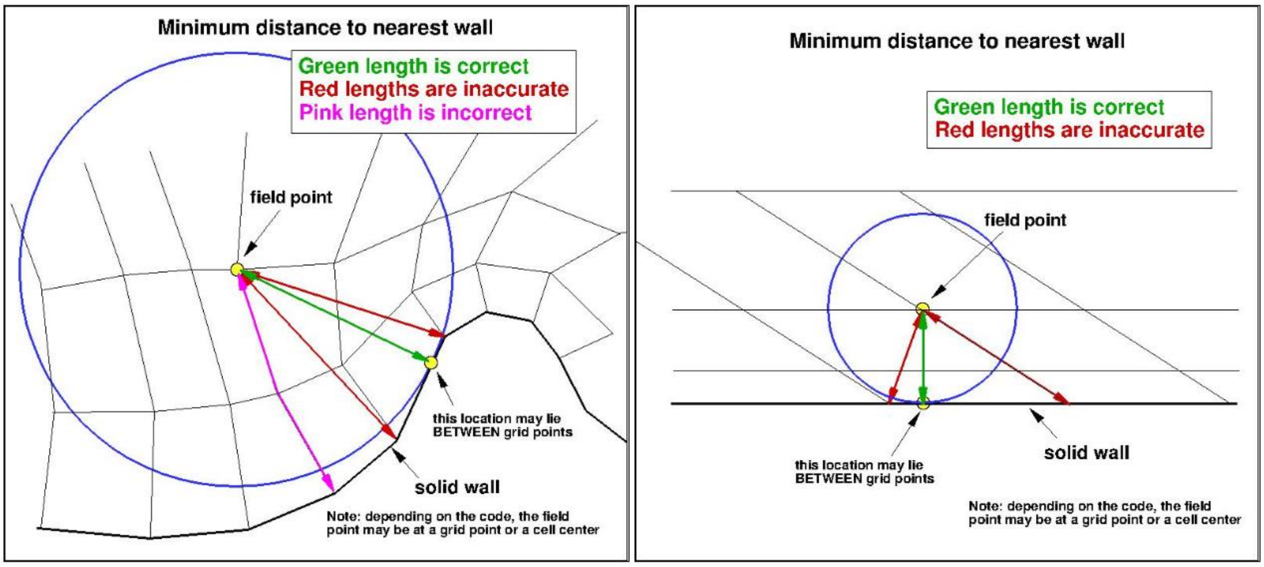

Minimum distance to nearest wall
Green length is correct
Red lengths are inaccurate
Pink length is incorrect
field point
this location may lie BETWEEN grid points
solid wall
Note: depending on the code, the field point may be at a grid point or a cell center
Minimum distance to nearest wall
Green length is correct
Red lengths are inaccurate
field point
solid wall
this location may lie BETWEEN grid points
Note: depending on the code, the field point may be at a grid point or a cell center

Figure 21: Illustration of how to determine the minimum distance d

5.3.4. Detached Eddy Simulation

Detached Eddy Simulation, also called DES, is a result of the work done by researchers trying to solve the challenges that are related to Large Eddy Simulations (LES), such as the computational cost. By combining RANS at the close wall regions and LES for the rest of the flow, the hybrid model DES is the final result. Originally, the DES model was created as an opportunity for replacing the distance function d in the Spalart-Allmaras turbulence model. A modified distance function $\tilde{d}$ was created and replaced the original distance function d (Caruelle & Ducros, 2003; CFD-online, 2008).

$$\tilde{d} = min[d, C_{DES}\Delta] \tag{55}$$

Where C_{DES} is a constant, while Δ is the largest dimension of the investigated grid cell. It is important to notice that equation (55) is using the minimum value of either d or $C_{DES}\Delta$, resulting in a model that is operating with RANS and the original Spalart-Allmaras when $\tilde{d} = d$. However, when $\Delta \ll d$ it is operating like a subgrid scale model. Moreover, as the viscous region thickens, the flow will also be treated like LES for $d > C_{DES}\Delta$. Finally, wakes and free shear layers is known to be away from a solid object, which means that these will also be treated as LES as long as the mesh is considered fine with a small Δ (Caruelle & Ducros, 2003; CFD-online, 2008).

5.4. Mesh

One of the key elements in order to have success in CFD is to achieve a good mesh. The definition of a good mesh varies for different case setups and the valuable knowledge can be considered as the ability to adapt and adjust the mesh generation for various case setups.

As a start, the mesh can be classified into four types, namely structured, unstructured, conformal and non-conformal mesh. An important difference between structured and unstructured mesh is the ability to mesh complex geometries. As an example, if a structured mesh were used for a motorbike, the sizes of the cells would have to be extremely small in order to obtain a smooth mesh. On the other hand, if unstructured mesh is used then the cells can adjust their shape based on the geometry making it easier to obtain a suitable mesh (Sarrate, 2018). An illustration of structured and unstructured mesh is given in figure 22 (Sarrate, 2018, p. 28).

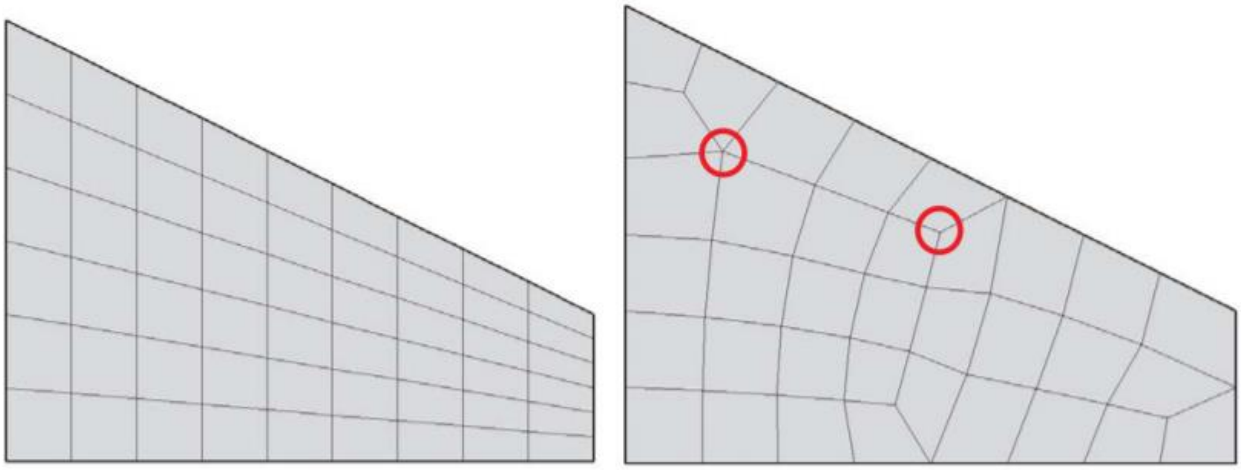


Figure 22: Structured and unstructured mesh

As for the conformal and non-conformal mesh, these can easily be separated by looking for hanging nodes. Both meshes on figure 22 can be treated as conformal, as they have no hanging nodes. However, on figure 23 (Sarrate, 2018, p. 29) there are three hanging nodes illustrated with a red circle, which is an illustration of a non-conformal mesh (Sarrate, 2018).

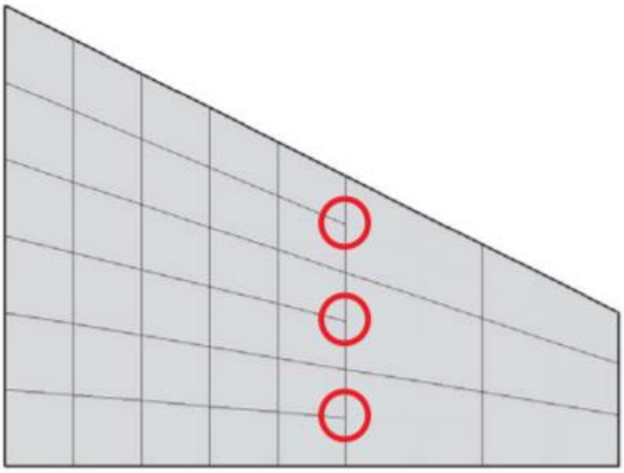


Figure 23: Non-conformal mesh

Moreover, the elements that make up the mesh can be classified as different polyhedral elements. Which elements to use totally depends on the geometry of the case. For complex geometries, a combination of more than one type of polyhedral elements is common in order to obtain a suitable mesh (Sarrate, 2018). An illustration of the most common polyhedral elements are given in figure

24 (Sarrate, 2018, pp. 31-32).

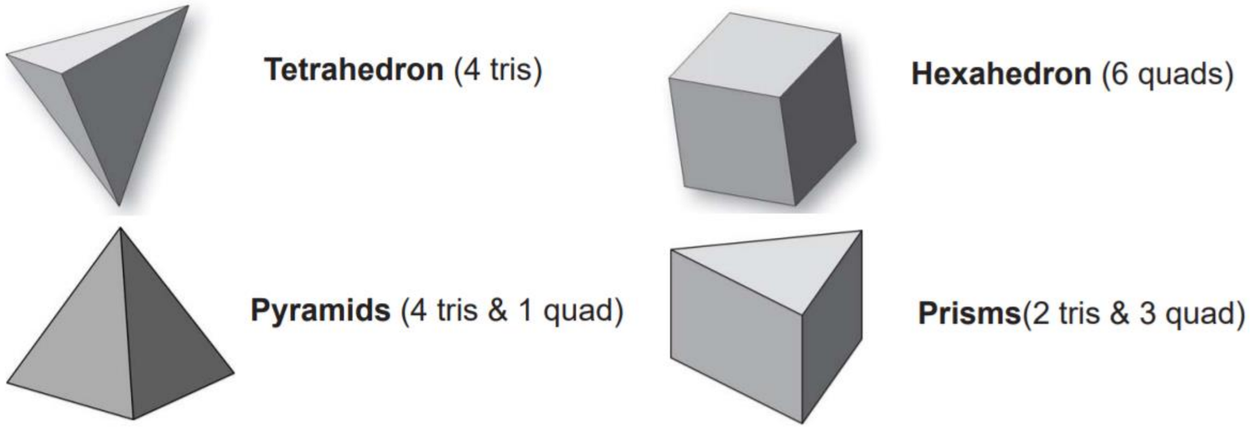


Figure 24: Different types of elements used in mesh generation

5.4.1. Near wall meshing

Ludwig Prandtl developed the concept behind the boundary layer, which is to divide the flow in two parts, an outer flow region and an inner flow region. The outer flow region typically represents the inviscid part. The inner flow region represents the viscous part, which for this theory is not neglected (Cengel & Cimbala, 2014). In figure 25 (Cengel & Cimbala, 2014, p. 566), the boundary layer is illustrated as a black line dividing the flow V in two parts.

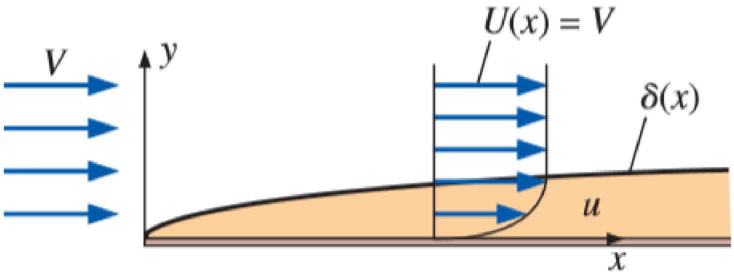


Figure 25: The boundary layer

Note that u and v are velocity components and that the flow V changes to $U(x)$ after reaching the solid surface. The reason is that the flow is changing as it is moving along the distance x on the solid surface. Finally, $\delta(x)$ represents the boundary layer thickness and can be treated as a function of the distance x as we move alone the solid surface.
The boundary layer is important to understand when working with the mesh generation. The quality of the mesh around a solid object can determine whether or not the user is able to fully resolve the flow along a solid object (Cengel & Cimbala, 2014).

In order to more accurately predict the turbulent flow behind an object, it is crucial to start with an accurate prediction of the near-wall flow. The non-dimensional parameter y^+ can be used to evaluate if the mesh is fine or coarse for any type of flow, especially in the near-wall region. y^+ can be calculated with the aid of the following formula (Cengel & Cimbala, 2014, p. 578):

$$y^+ = \frac{yu_*}{\nu} \tag{56}$$

Where the distance from the wall to the cell of interest is y, ν is the kinematic viscosity and finally u_* is defined as the friction velocity. Moreover, the friction velocity can be expressed as (Cengel & Cimbala, 2014, p. 576):

$$u_* = \sqrt{\frac{\tau_w}{\rho}} \tag{57}$$

Where τ_w is the shear stress and ρ is the density. On figure 26 (Xu Y. , 2016, p. 152), the different subdivisions for the near-wall region are illustrated with a coherent y^+ value.

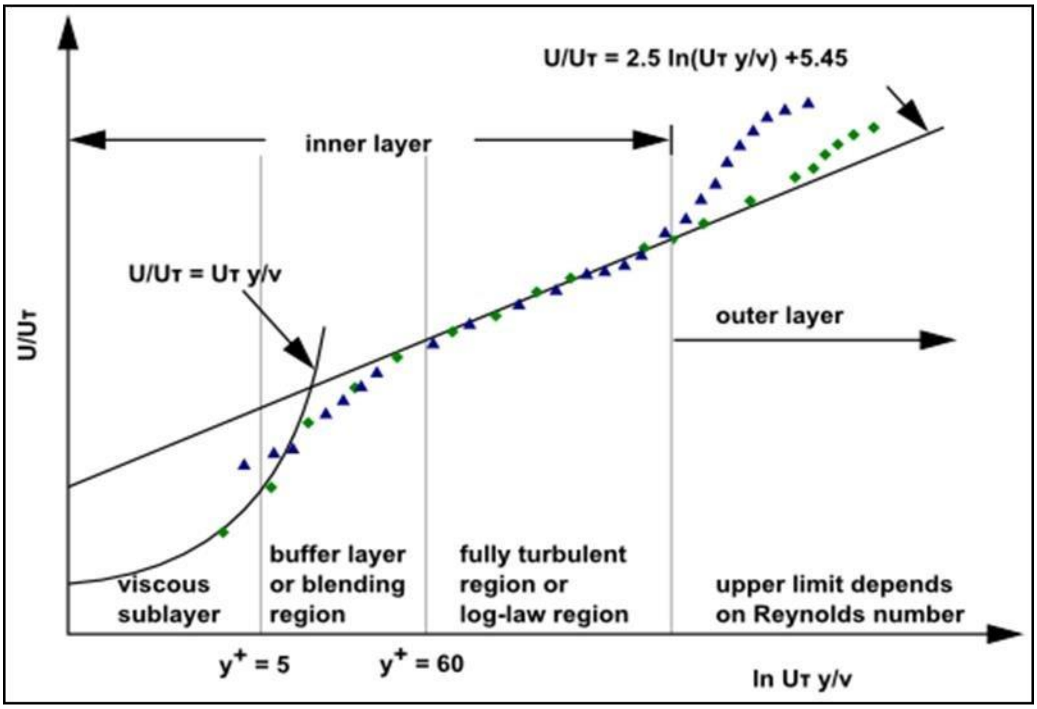


Figure 26: Subdivision of near-wall region

Furthermore, in figure 27 (Salim & Cheah, 2009, p. 2) a comparison of three different meshes are illustrated, as well as the coherent y^+ value. Pay attention to how dense the cells are together in Mesh 3, compared to Mesh 1, giving it a much lower y^+ value, which increases the ability of resolving the viscous sublayer near the surface (Salim & Cheah, 2009).

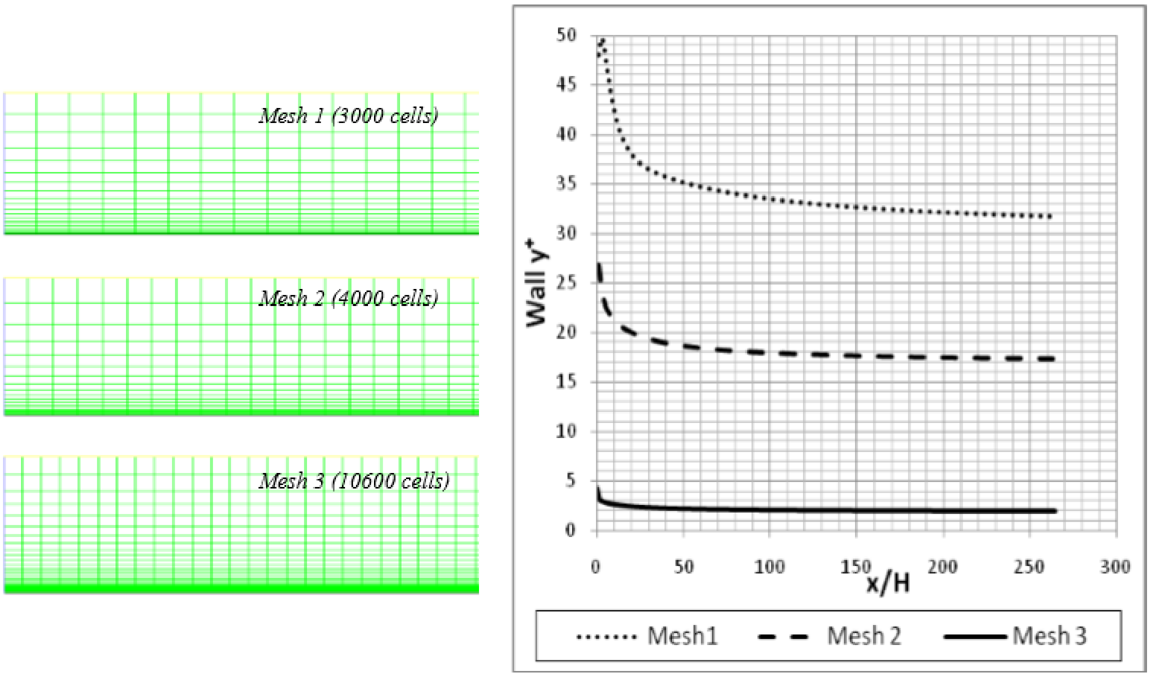


Figure 27: y^+comparison

5.5. Discretization process

As for differential equations, they represents some type of conservation of a physical problem, which can be anything from fluid dynamics to electromagnetics. It is fair to say that a differential equation consists of some quantity that is being treated as the dependent variable, where the equation itself describes how this quantity changes. Furthermore, the expression for the Generic Scalar Transport Equation can be written as (CFD-Online, 2006):

$$\underbrace{\frac{\partial}{\partial t}(\rho\phi)}_{Transient\ term} + \underbrace{\nabla(\rho\vec{u}\phi)}_{Convection\ term} = \underbrace{\nabla\left(\Gamma^{\phi}\nabla\phi\right)}_{Diffusion\ term} + \underbrace{S^{\phi}}_{Source\ term} \tag{58}$$

Where ρ is the density, $\vec{u}$ is the velocity vector and Γ is the diffusion coefficient. (CFD-Online, 2006). When applying these differential equations on a computational domain they are being discretized. The idea behind the discretization process is to turn the differential equations into a system of algebraic equations. These algebraic equations can then be solved as numerical solutions. The numerical solution is all about finding values for ϕ, which can be considered as the dependent variable. There will be separate values of ϕ for all the cells inside the domain, which makes it possible to describe and visualize the changes (Moukalled, Mangani, & Darwish, 2016). The complete discretization process is illustrated in figure 28 (Moukalled, Mangani, & Darwish, 2016, p. 86).

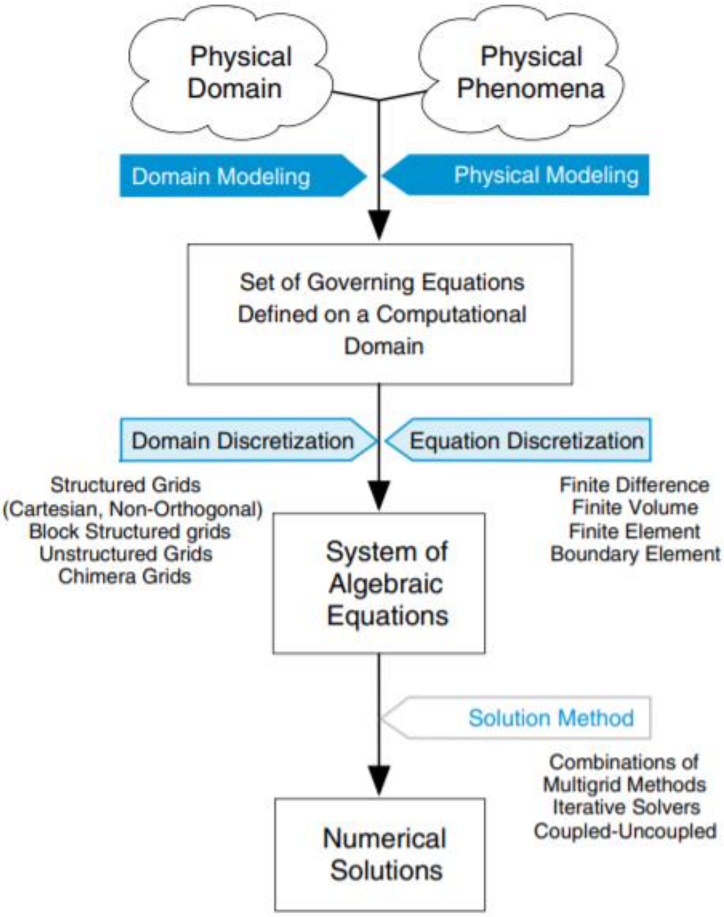


Figure 28: The complete discretization process

5.6. OpenFOAM

5.6.1. Finite Volume and case setup in OpenFOAM

Open Field Operation and Manipulation, known as OpenFoam, is a strong tool using the Finite Volume (FV) method for solving complex CFD-problems. The idea behind the FV method is to divide the computational domain into a finite number of control volumes. Looking at figure 29 (Tu, Ahmadi, & Inthavong, 2013, p. 175), the fluid flow properties is stored inside *P*. The transportation of these properties will be further explained in chapter 5.6.2 and 5.6.3. By imagining that a physical principal like conservation of mass is applied to the control volume, Tu, Ahmadi and Inthavong (2013, p. 174) states that: *"then the flux of a variable is equal to the net change in quantity of the variable inside the control volume"*. This is considered as one of the major principles of the FV method (Tu, Ahmadi, & Inthavong, 2013; Foroutan, 2014).

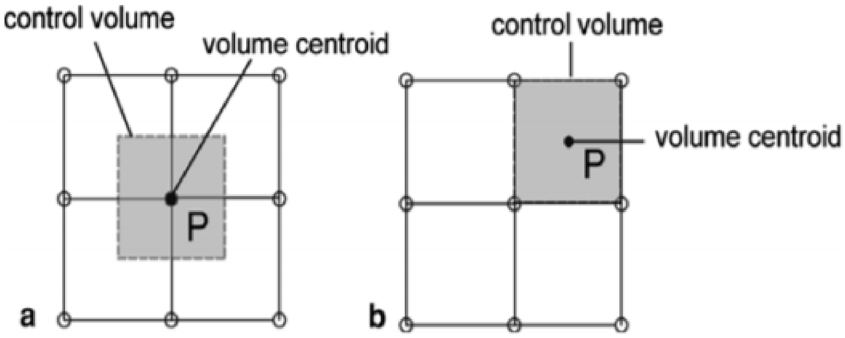


Figure 29: FV method for vertex centered (a) and cell centered (b)

Moving on to the case setup in OpenFOAM, where an example is illustrated in figure 30 (Foroutan, 2014, p. 41). Inside the case folder it is common to have three folders, namely *0*, *constant* and *system*. The *0* folder represents the initial conditions for case setup. This could be information about velocity, pressure, temperature etc. The *constant* folder consists of everything that is constant for the case setup, which includes both the turbulence model and the mesh of the case. Moreover, the system folder consist of details about how to solve the case. This includes details about the time step, solver settings and schemes on how the governing equations are discretized into algebraic equations ((Foroutan, 2014).

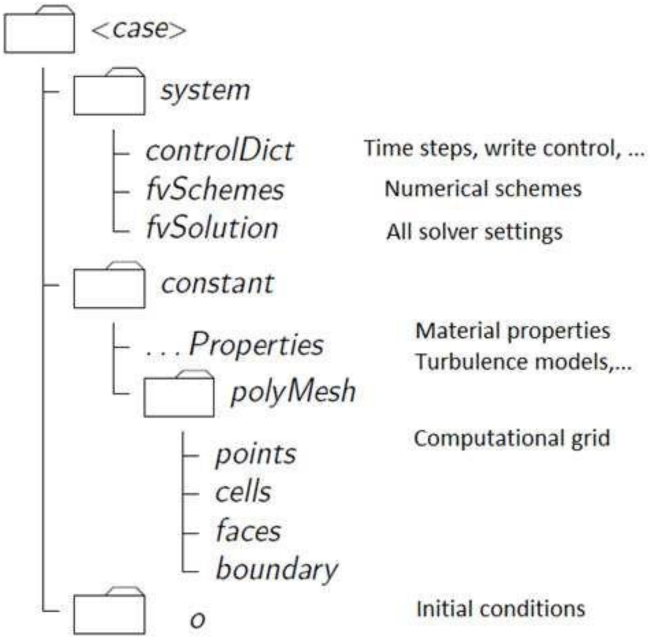


Figure 30: Illustration of the different folders in an OpenFOAM case

5.6.2. First order Upwind scheme

In CFD it is common to distinguish between first, second or higher order solutions. The difference between them in simple terms is how the dependent variable ϕ transports information from one cell center to the next face center. The Upwind scheme is a first order solution and the transportation of ϕ is illustrated in figure 31 (TU Wien, 2019, p. 3). The key detail in the first order solution is when the information stored in cell center P is transported to face center e, the value ϕ is constant for any positive flux through e (TU Wien, 2019). In simpler terms it can be illustrated as follows (TU Wien, 2019, p. 2):

$$\phi_P = \phi_e \qquad if\ \ F_e > 0 \tag{59}$$

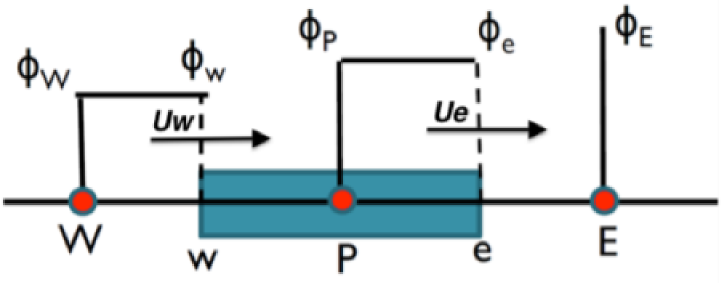


Figure 31: First order Upwind

5.6.3. Second order Upwind scheme

As a rule of thumb, second order solutions are more accurate compared to first order solutions. Unlike the first order Upwind scheme that is based on one neighbor at each side, the second order Upwind scheme is based on having two neighbors at each side and is illustrated in figure 32 (Chochua, 2002, p. 21). By having two neighbors, linear interpolation is performed, resulting in a linear line representing the transportation of the stored information in the cell center to the next face center (Chochua, 2002).

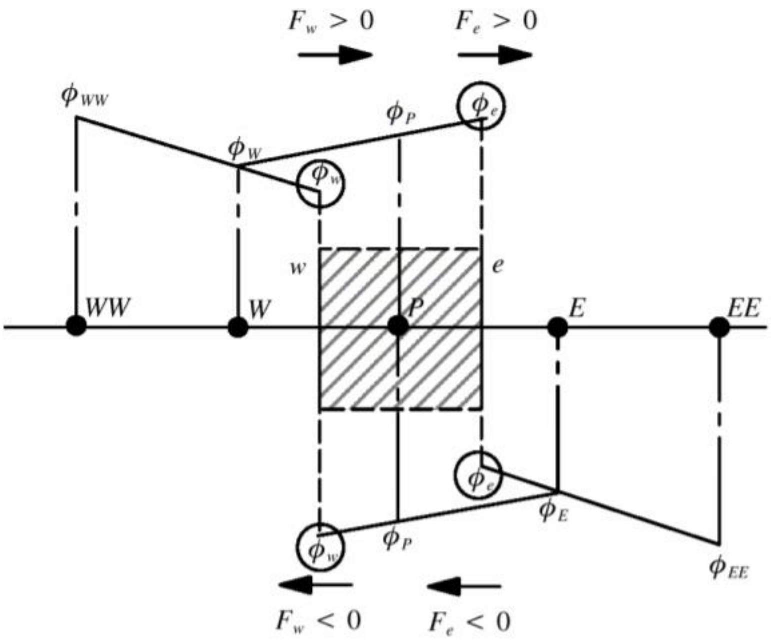


Figure 32: Second order Upwind scheme

5.7. PyFR

As previously described, OpenFOAM is generally based on the Finite Volume method (FV) and has the opportunity to choose between different first and second order solutions. PyFR on the other hand is a Python based framework using the Flux Reconstruction method (FR), which has the ability to solve higher order solutions, which per definition means third order or higher. A highly separated flow with a Reynolds number between 10^4 - 10^7 and a Mach number between 0.1 - 1 has been proven hard to reconstruct accurately with first and second order solutions. Due to this issue, there has been an increasing interest for higher order solutions (Witherden, Farrington, & Vincent, 2013). Moreover, as this project has its focus on flows at 90° angle of attack that result in a high flow separation, the investigation of PyFR could give valuable information and assist future research.

5.7.1. Governing equations

As mentioned, PyFR is based on Flux Reconstruction method where only the most important mathematical expressions will be explained in this chapter. For a detailed step-by-step solution and a detailed description of the variables, the following article can be useful: *"PyFR: An open source framework for solving advection-diffusion type problems on streaming on streaming architectures using the flux reconstruction approach"*, which was written in 2013 by F.D. Witherden, A.M. Farrington and P.E. Vincent. As a start, an arbitrary domain having an advection diffusion problem is considered (Witherden, Farrington, & Vincent, 2013, p. 3030):

$$\frac{\partial u_\alpha}{\partial t} + \nabla \cdot f_\alpha = 0 \tag{60}$$

Where u_α is some conserved quantity and f_α is the flux of the same quantity. Unlike the FV method, the FR method has multiple solutions within one cell, which is illustrated in figure 33 (Witherden, Farrington, & Vincent, 2013, p. 3031). The blue circle represents solution points, while the orange squares represents flux points. Since PyFR is operating with multiple solutions points per cell, the amount of cells needed to obtain a fine mesh is lower compared to OpenFOAM. (Witherden, Farrington, & Vincent, 2013).

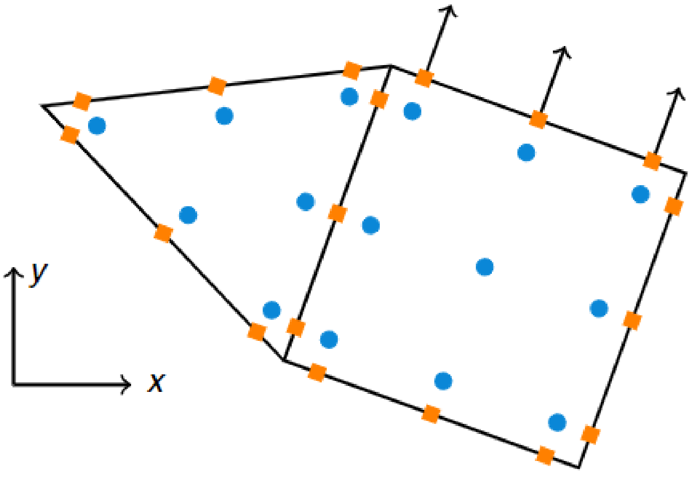


Figure 33: Solution and flux points for a two cells

Starting with a discontinues solution for the blue solutions points. Based on this, the first goal is to achieve a discontinues solution for the orange flux points. An illustration of such a discontinues solution is given in figure 34 (Witherden & Vincent, 2016, p. 15).

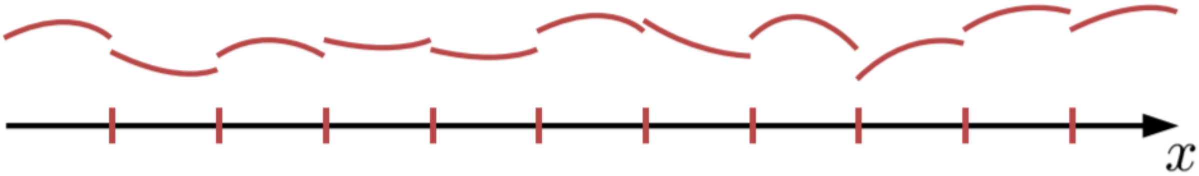


Figure 34: Example of a discontinues solution

Next step is to obtain an expression for the divergence of the continues flux, which can be given as (Witherden, Farrington, & Vincent, 2013, p. 3031):

$$\left(\bar{\nabla}\cdot\bar{f}\right)^{(u)}_{e\rho\pi\alpha} = \left[\bar{\nabla}\cdot g^{(f)}_{e\sigma}(\bar{x})\left\{\xi_\alpha \bar{f}^{(f\perp)}_{e\sigma\pi\alpha} - \bar{f}^{(f\perp)}_{e\sigma\pi\alpha}\right\} + \bar{f}^{(u)}_{e\nu\pi\alpha}\cdot\bar{\nabla} l^{(u)}_{e\nu}(\bar{X})\right]_{\bar{X}=\bar{X}^{(u)}_{e\rho}} \tag{61}$$

Where the governing system in equation (x) gets a semi-discretized form (Witherden, Farrington, & Vincent, 2013, p. 3031):

$$\frac{\partial u^{(u)}_{e\rho n\pi}}{\partial t} = -J^{-1(u)}_{e\rho n}\left(\bar{\nabla}\cdot\bar{f}\right)^{(u)}_{e\rho\pi\alpha} \tag{62}$$

Where $J^{-1(u)}_{e\rho n}$ is defined as (Witherden, Farrington, & Vincent, 2013, p. 3031):

$$J^{-1(u)}_{e\rho n} = \det J^{-1}_{en}\left(\bar{X}^{(u)}_{e\rho}\right) = \frac{1}{J^{(u)}_{e\rho\pi}} \tag{63}$$

The divergence of the continues flux is then evaluated at the solution points, which can be visualized in figure 35 (Witherden & Vincent, 2016, p. 28).

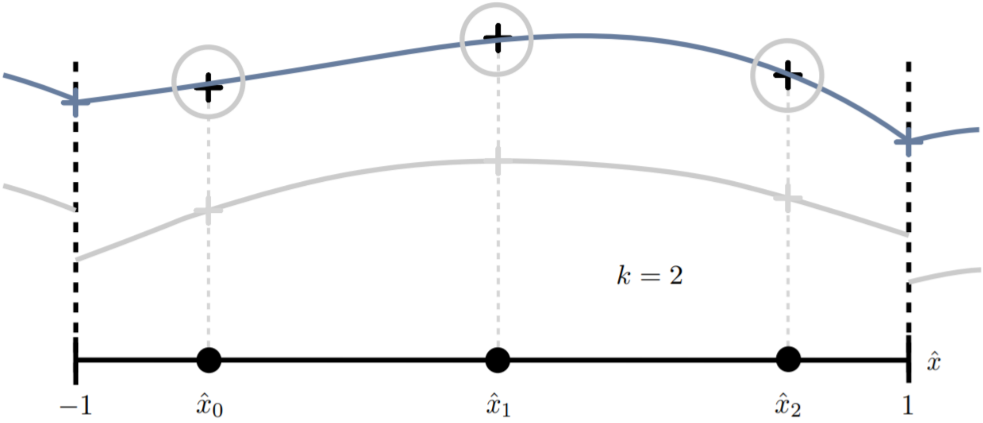


Figure 35: Evaluating the divergence of the continues flux at three arbitrary solution points

Currently, PyFR has the ability to run on two different governing systems, namely compressible Navier Stokes for viscous compressible flow and Euler for inviscid compressible flow. The compressible Navier Stokes will be considered here, where the flux now can be defined as (Witherden, Farrington, & Vincent, 2013, p. 3033):

$$f = f^{(inv)} - f^{(vis)} \tag{64}$$

Which states that flux is equal to the inviscid flux minus the viscous flux. The full mathematical solution for the inviscid and viscous flux can be further investigated in the reference mentioned in the beginning of chapter 5.7.1, namely; *"PyFR: An open source framework for solving advection-diffusion type problems on streaming on streaming architectures using the flux reconstruction approach"*, which was written in 2013 by F.D. Witherden, A.M. Farrington and P.E. Vincent (Witherden, Farrington, & Vincent, 2013).

5.8. Qblade and Xfoil

Qblade is an Open Source software that can be used for both design and simulation of wind turbines. By importing a set of different airfoils, the user has the opportunity to design a complete wind turbine blade, as well as performing simulations on the imported airfoils and the turbine blade by the help of Xfoil, which is integrated into the software. Xfoil is a handy tool for computing the flow around airfoil, especially for angles between -15° and 15° angle of attack. However, it has some limitations when computing stall. To deal with this matter, Xfoil has an algorithm called Montgomery. The idea behind this method is to solve the system as potential flow near 0° and 180°, while solving the rest of the angles as a thin plate. In order to connect these two different ways of solving the flow, a blending function is used, which results in a linear plot for small angle of attack and a curve for the higher angle of attacks (Qblade, 2020; Marten & Peukert, 2013).

6. Introducing PyFR for a 2D cylinder case

6.1. Pre-processing

In order to compare a second order solution with a higher order solution and investigate the differences in predicting flow and drag, a simple 2D cylinder was created using Blender. Blender is an Open Source software that has a wide range of functionalities, including 3D modelling, animation and simulation. As for CFD, Blender has the opportunity to create and export blockMesh dictionaries, which can be handy when designing an OpenFOAM case from scratch. Moreover, Blender works well for cleaning up, scaling, rotating and moving imported STL-files (Blender, 2020). The cylinder was exported as an STL-file and imported into Pointwise, which is a grid generation software primarily used to execute the preprocessing part of CFD analysis. It is fair to say that CFD analysis consists of three major parts, namely meshing, solution and post-processing. Pointwise is a user-friendly software, which makes the meshing easy and accurate. Furthermore, Pointwise is able to export and convert the mesh to different CFD-solvers likes OpenFOAM and PyFR (Pointwise, 2020).

The case setup was designed based on *"Investigation of transitional turbulence models to predict drag crisis for flows over spheres and cylinder"*, a master thesis written by Seyedeh Mona Nakhostin in 2019. Details about her case setup are given in table 8 (Nakhostin, 2019).

Table 8: Relevant information for the case setup

Reynolds number	3900	Dimensionless
y+	1	Dimensionless
Wall spacing mesh, Δs	0.004	m
Cells in radial direction	100	Dimensionless
Radius O-grid	25·Diameter	m

Nevertheless, Pointwise requires a number of cells parallel to the cylinder. In order to estimate the required grid size, a grid sensitivity study was done in openFOAM due to the expected simulation time being faster for RANS simulations in OpenFOAM compared to PyFR. The grid sensitivity study was done using a Reynolds number of 3900. The result is given in figure 36, indicating that a total number of 100 cells parallel to the cylinder should be sufficient.

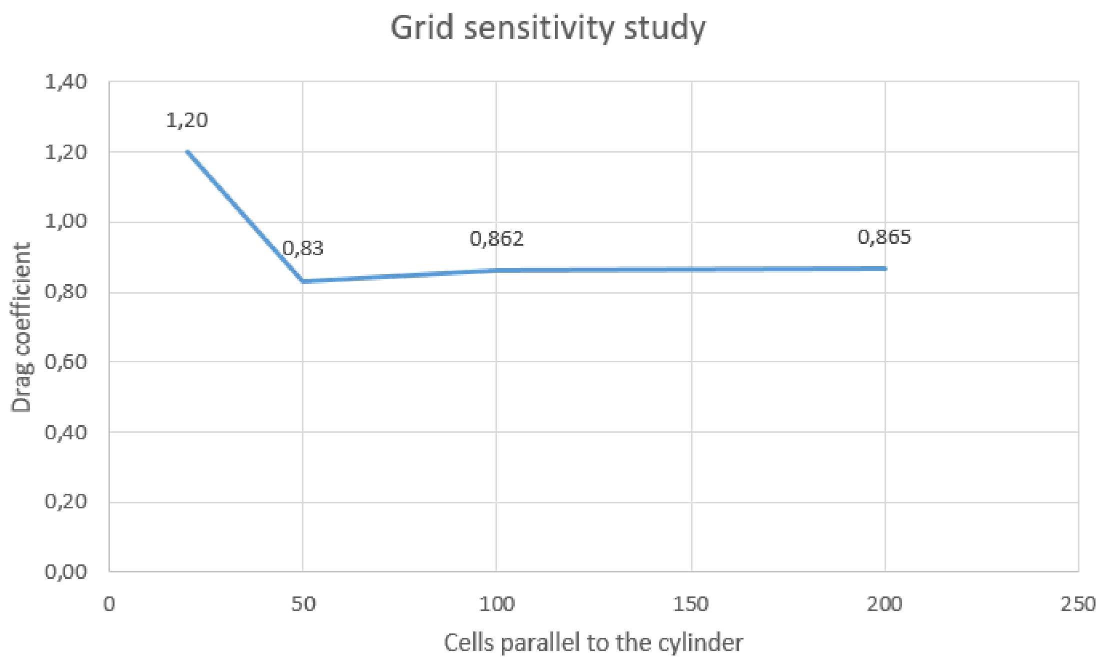


Figure 36: Grid sensitivity study for the 2D cylinder

The final grid for the cylinder is given in figure 37. As mentioned in chapter 5.7.1, PyFR has multiple solutions points per cell. When viewing the mesh in ParaView, which is an open source software that can be used for analyzing CFD-simulations, one can see that the cells in the mesh are subdivided in order to visualize every solution point. For a large and complex mesh, this could have a huge impact on the simulation time and the number of cells might need to be reduced before running the simulation with PyFR. However, for a small case like this no adjustments on the mesh was done, as the simulation time was expected to be low. In other words, the same original mesh was used for both OpenFOAM and PyFR. The differences is clearly illustrated when comparing figure 37 and 38. In addition, by having a closer look at figure 33, it is easier to understand how one cell is subdivided into multiple solution points.

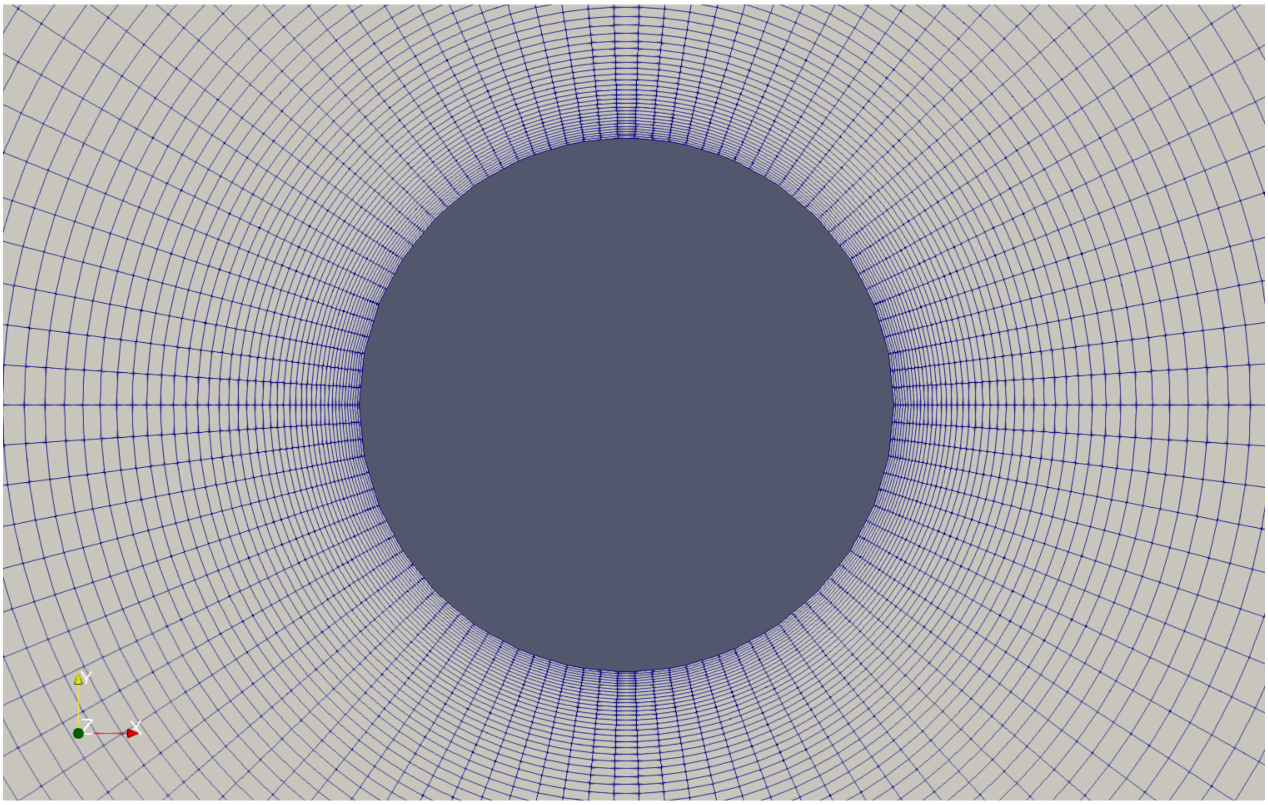


Figure 37: The generated mesh for the OpenFOAM simulation

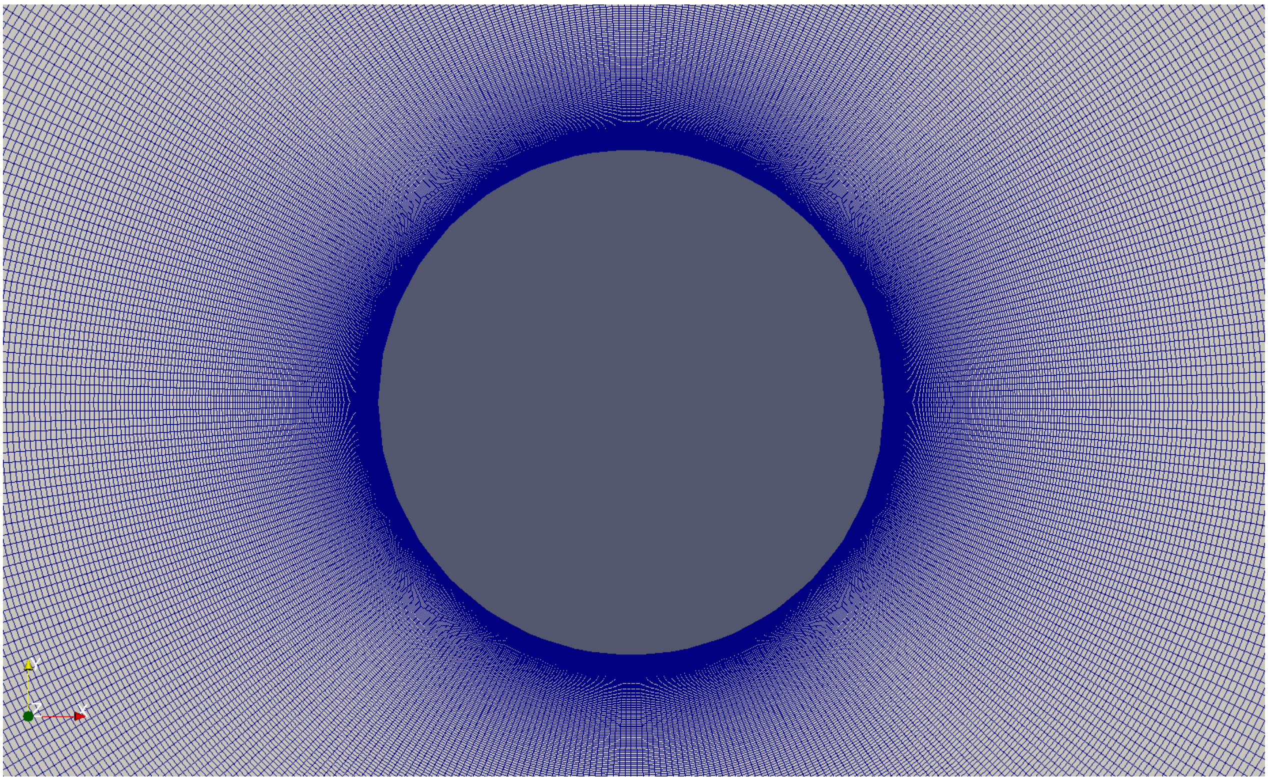


Figure 38: The generated mesh for the PyFR simulation

The case setup for PyFR works a bit different compared to OpenFOAM. There are only two files of importance. One *.pyfrm* file, which is the mesh file and *.ini* file where all the commands about how to solve the system are stored. For this 2D cylinder case, a combination of two tutorials were used in order to create the case setup. The Runge-Kutta rk45 Scheme was used for the following simulation. Other parameters and boundary conditions are listed in the following tables:

Table 9: Important parameters for the PyFR case

Reynolds number	3900	Dimensionless
Velocity, u	0.0585	m/s
Density, ρ	1.225	kg/m^3
Kinematic viscosity, ν	$1.470 \cdot 10^{-5}$	m^2/s
Dynamic viscosity, μ	$1.802 \cdot 10^{-5}$	$\mathrm{kg}/\mathrm{m} \cdot \mathrm{s}$
Gamma	1.4	Dimensionless
Prandtl number, Pr	0.71	Dimensionless
Diameter, L	1	m
Reference area, A_{ref}	1	m^2
y+	1	Dimensionless
Wall spacing mesh, Δs	0.004	m

Table 10: Boundary conditions for the PyFR case

Name	Boundary condition/Scheme
Airfoil	No-slip Adiabatic Wall
Front	Characteristic Riemann Invariant
Back	Slip Adiabatic Wall
Farfield	Slip Adiabatic Wall

6.2. Post-processing

The RANS 2D simulation started to converge after approximately 1000 iterations and reached a stable converging value of 0.862 after 2000 iterations. The flow around the cylinder is illustrated in figure 40.

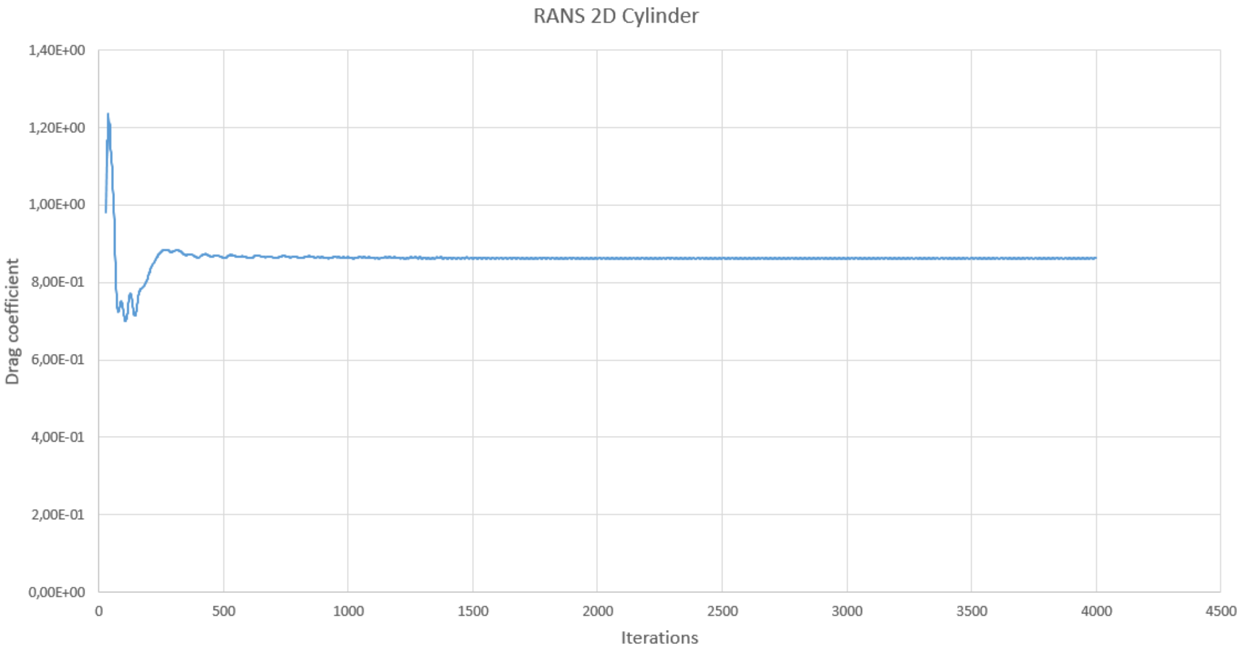


Figure 39: Drag coefficient vs Iterations, RANS 2D cylinder

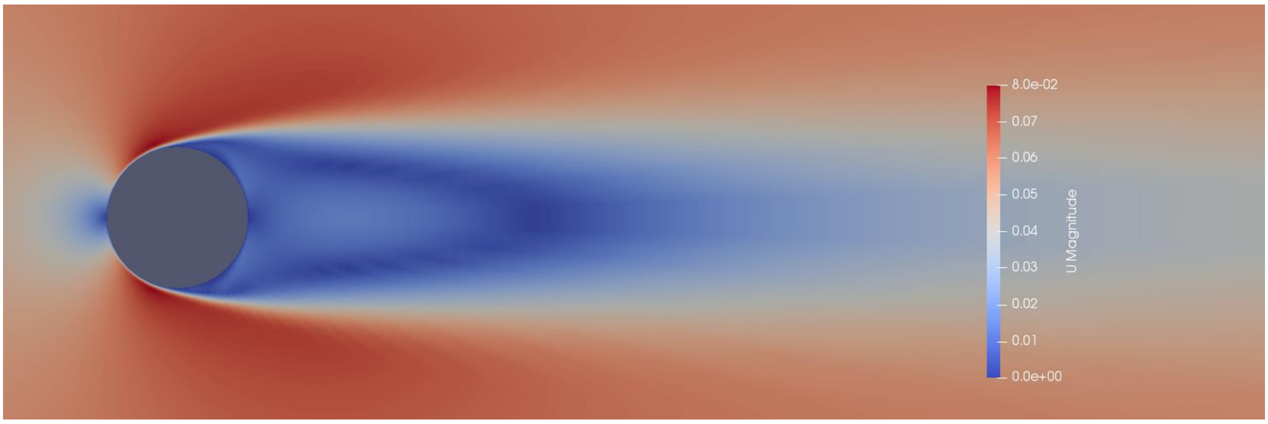


Figure 40: 2D cylinder, RANS second order solution

The drag coefficients results for the PyFR simulation is given in figure 41. Please note that the number of iterations on the x-axis is approximately 1000 times larger in reality and was reduced for simplicity when creating the plot. There are more or less no change in the drag coefficient between 46 and 71 on the x-axis. However, some small and long fluctuations are noticeable and an average drag coefficient of 0.97 is estimated.

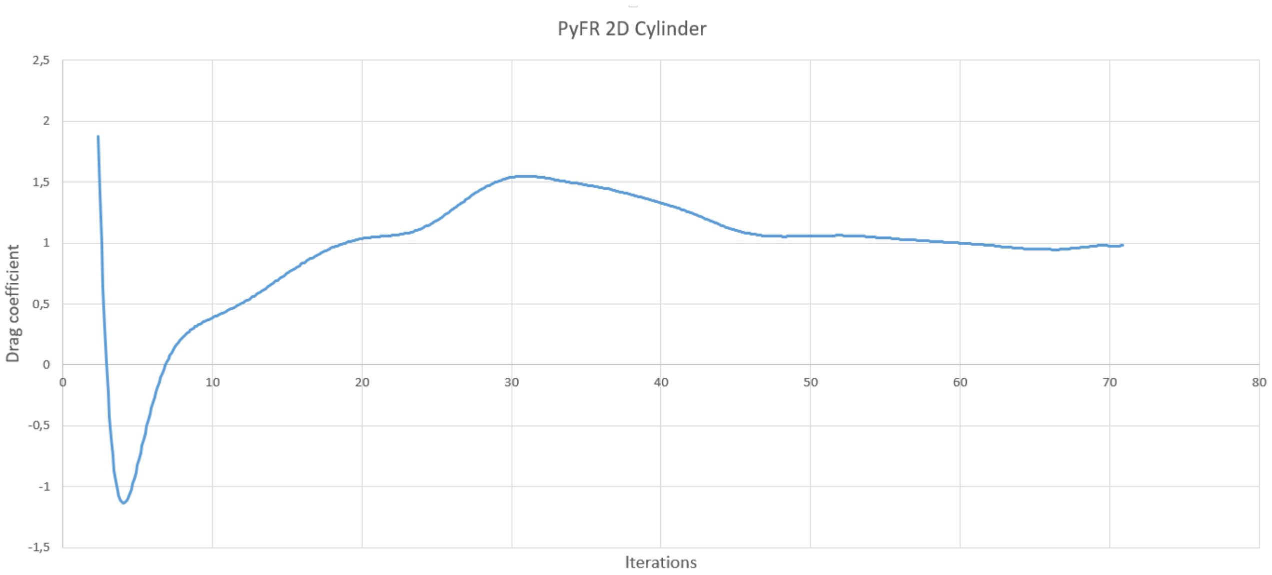


Figure 41: Drag coefficients vs Iterations, PyFR 2D Cylinder

The flow over the cylinder for the PyFR simulation is illustrated in figure 41. The simulation show a great ability to show the turbulent flow behind the cylinder, as some clear vortices are present. As this image shows the development of the turbulent flow, it is expected that these vortices will move away from the cylinder over time. The details in the flow are also of a high resolution, making it easier to see how the flow behaves and investigate areas with a high precision. The picture of the flow is taken after 70 000 iterations, which is approximately 70 on the x-axis in figure 41.

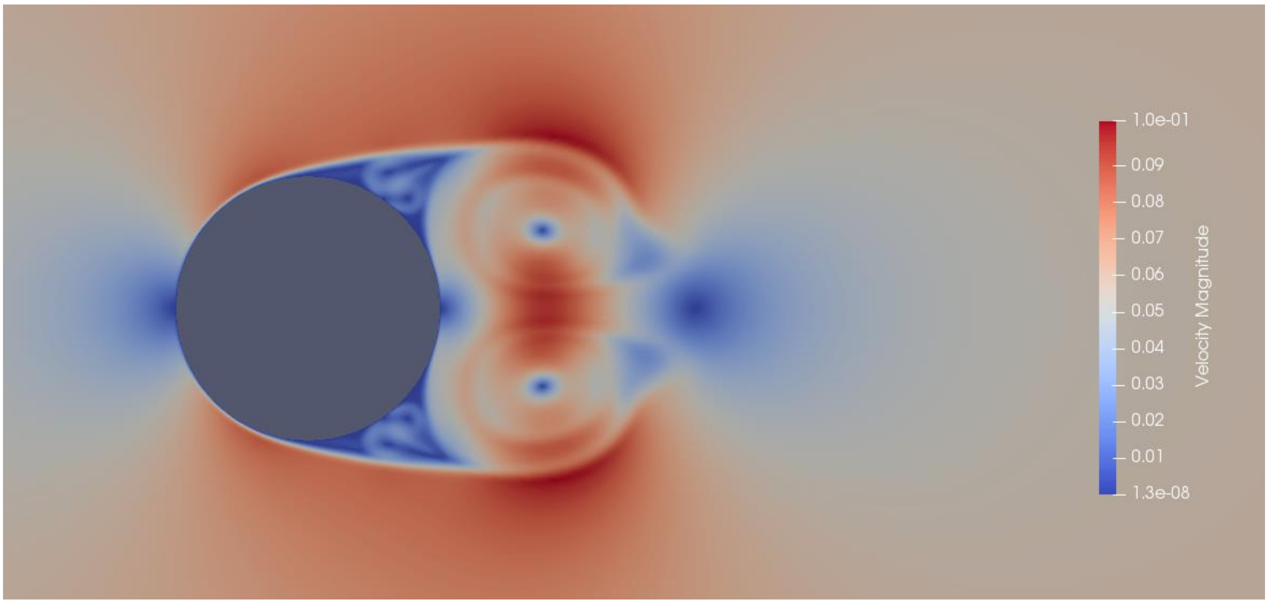


Figure 42: 2D cylinder, PyFR third order solution

In addition, some CFD simulations, as well as experimental results from other researchers are included in order to more accurate compare the simulations. A lot of previous research is done on cylinders at a Reynolds number of 3900, making it easier to get a clear picture of what to expect. All researchers in table 11 are given as a reference.

Table 11: Overview of results for a cylinder at Reynolds number = 3900

Experimental/Simulation	Source	Drag Coefficient
Experimental	(Lourenco & Shih, 1993)	0.99
Experimental	(Xu, Chen, & Lu, 2007)	0.98 ± 0.05
DES-SST	(Xu, Chen, & Lu, 2007)	1.076
DES-SA	(Xu, Chen, & Lu, 2007)	1.017
DES	(Nakhostin, 2019)	0.998
DNS	(Ma, Karamanos, & Karniadakis, 2000)	0.84
RANS	Hjorteland, 2020	0.862
PyFR	Hjorteland, 2020	0.97

6.3. Discussion

The PyFR simulation was a close match to the experimental results, as well as other second order solutions. Even though one can argue that the simulation could run for a longer period, it was stopped due to having more or less no changes in the drag coefficient for several days. Figure 42 shows that PyFR has a great potential for predicting and visualizing turbulent flow at a detailed level, making it an interesting contributor for further research on highly separated flows.

7. Validating the computational setup for the DU96-W-180 airfoil

7.1. Pre-processing

The DU96-W-180 airfoil was created by plotting a set of coordinates in AutoCAD, which is a computer aided design software from Autodesk, designed for making detailed 2D and 3D drawings. It is a common tool for both engineers and architects worldwide, as the software has multiple tools and options (Autodesk, 2020). From there, the airfoil was extruded in order to obtain a 3D shape. Importing the airfoil into Pointwise, a 255x128 and a 255x128x32 mesh was created for the 2D and 3D case. The domain is an O-grid with a radius of 30m for both cases. The reason for choosing these particular numbers were to obtain a similar mesh to the one used by Sørensen and Skrzypiński, presented in chapter 3.2.

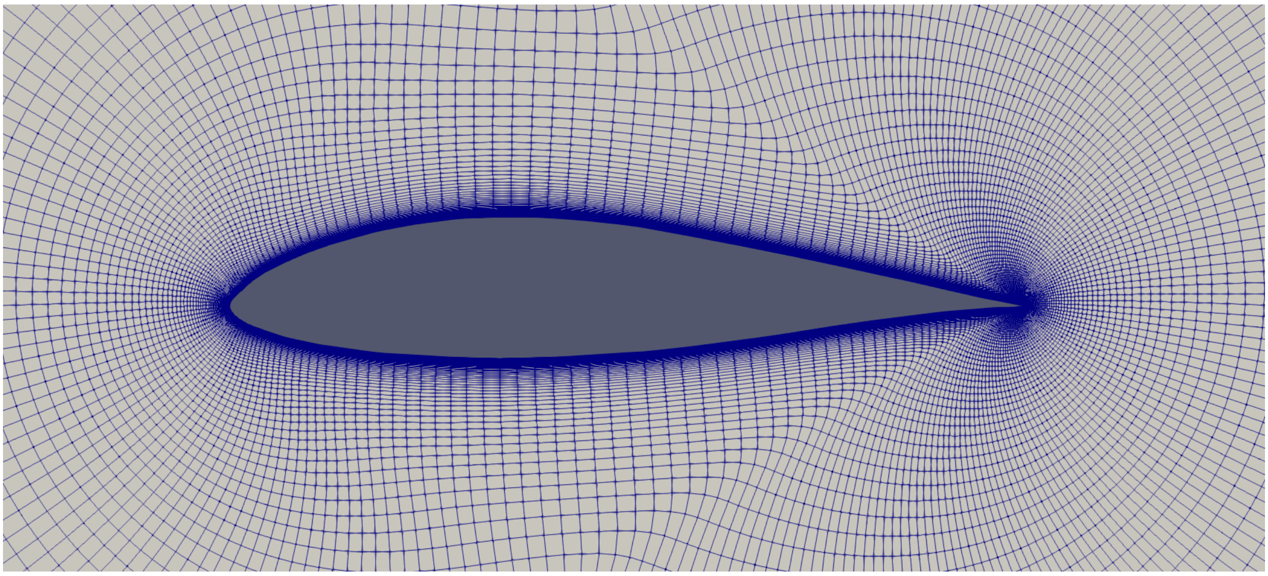


Figure 43: The mesh for the DU96-W-180 airfoil created in Pointwise

There are two differences worth discussing between the mesh created by Sørensen and Skrzypiński and the mesh presented here. The first difference is the number of cells parallel to the airfoil. There are 256 cells parallel to the airfoil in the mesh created by Sørensen and Skrzypiński, while there are 255 cells in the mesh presented here. When meshing in Pointwise, one can specify the number of cells in each direction and details on how the cells should be created. Nevertheless, Pointwise is built on a set of rules and tries to get as close as possible to the specified values, hence creating a 255x128 mesh. In addition, a 255x128 2D mesh consist of 32640 cells, which is within the converging range in the grid sensitivity study in figure 7. The second difference is the number of cells in the perpendicular and spanwise direction for the 3D mesh. Where Sørensen and Skrzypiński had a 256x384x128 mesh for the 3D case, a 255x128x32 mesh, having a total of 1044480 cells was designed for this project. The purpose for reducing cells in the perpendicular and spanwise direction is simply to reduce the simulation time and memory.
Furthermore, Pointwise has the ability to easy predefine a y+ value and calculate the delta spacing for the mesh generation. A y+ value of 0.5 was selected, giving a delta spacing of 0.00001638m for a Reynolds number of $0.7 \cdot 10^6$. An illustration of the spanwise mesh for the 3D airfoil is given in figure 44.

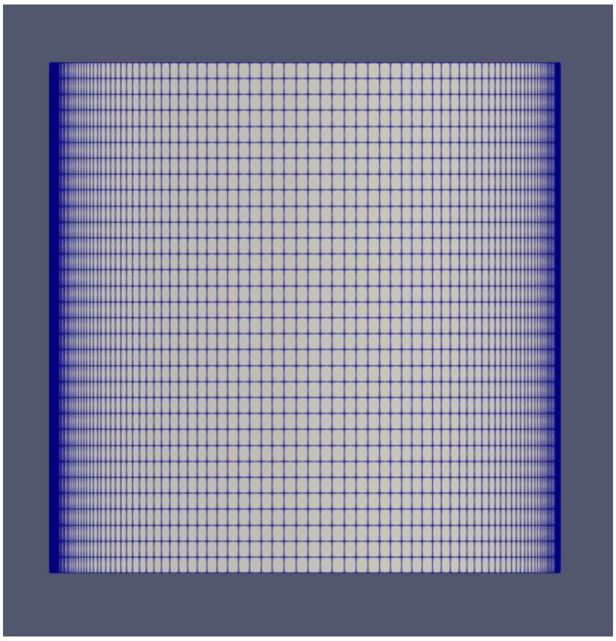


Figure 44: The mesh for the DU96-W-180 airfoil in the span wise direction

Before exporting the mesh, one has to select proper boundary conditions for the domain. The airfoil was defined as *wall*, the farfield was defined as *patch*, while the front and back was defined as *empty* for the 2D case and *cyclicAMI* for the 3D case. As for the initial conditions, the farfield has a *freestreamVelocity* of 10.3 m/s in the y-direction and a *freestreamPressure* value of zero. The airfoil is considered a *wall* where the *noslip* condition applies, while the pressure is set as *zeroGradient*. Other relevant conditions are found in table 12.

Table 12: List of constants for the case setup

Angle of attack	90°	Degrees
Reynolds number	$0.7 \cdot 10^6$	Dimensionless
Velocity, u	10.3	m/s
Density, ρ	1.225	kg/m^3
Kinematic viscosity, ν	$1.470 \cdot 10^{-5}$	m^2/s
Dynamic viscosity, μ	$1.802 \cdot 10^{-5}$	$kg/m \cdot s$
Chord length, L	1	m
Reference area, A_{ref}	1	m^2
y+	0.5	Dimensionless
Wall spacing mesh, Δs	0.00001638	m

The flow over an airfoil with a 90° angle of attack has a high flow separation and can be unpredictable. In order to more accurate see a pattern and understand the results, different solvers in OpenFOAM should be tested. It was decided to use *simpleFoam* (RANS), *pimpleFoam* (URANS) and *pisoFoam* (DES) as they represent different ways to solve the flow. Moreover, it was decided to test both *upwind* (first order) and *linearUpwind* (second order) for the *simpleFoam* (RANS) and *pimpleFoam* (URANS) simulations. In addition, simplified solutions from Xfoil and higher order solutions from PyFR can give valuable insight to the results and should also be considered.

7.2. Post-processing

As a start, first and second order solutions for RANS and URANS simulations are compared. There were two major reasons for this. First of all, as the project implements third order solutions through PyFR as a possible way of obtaining more accurate results, hence investigating the differences between first and second order solutions on an airfoil is considered relevant. Moreover, when working with the 3D simulations some difficulties regarding extreme divergence of the solutions was encountered. This was typically values for the drag coefficient going towards infinity. A way of investigating the problem could be to check if the same extreme divergence also occurs for the first order solution, where the following answer could give an indication of the problem.

In figure 45 the drag coefficient for RANS 2D first and second order solutions are plotted against the number of iterations. The drag coefficients starts to converge after approximately 700 iterations for both simulations. There is a small difference between the simulations, giving RANS 2D first order a converging drag coefficient of 1.74 and the RANS 2D second order a converging drag coefficient of 1.64.

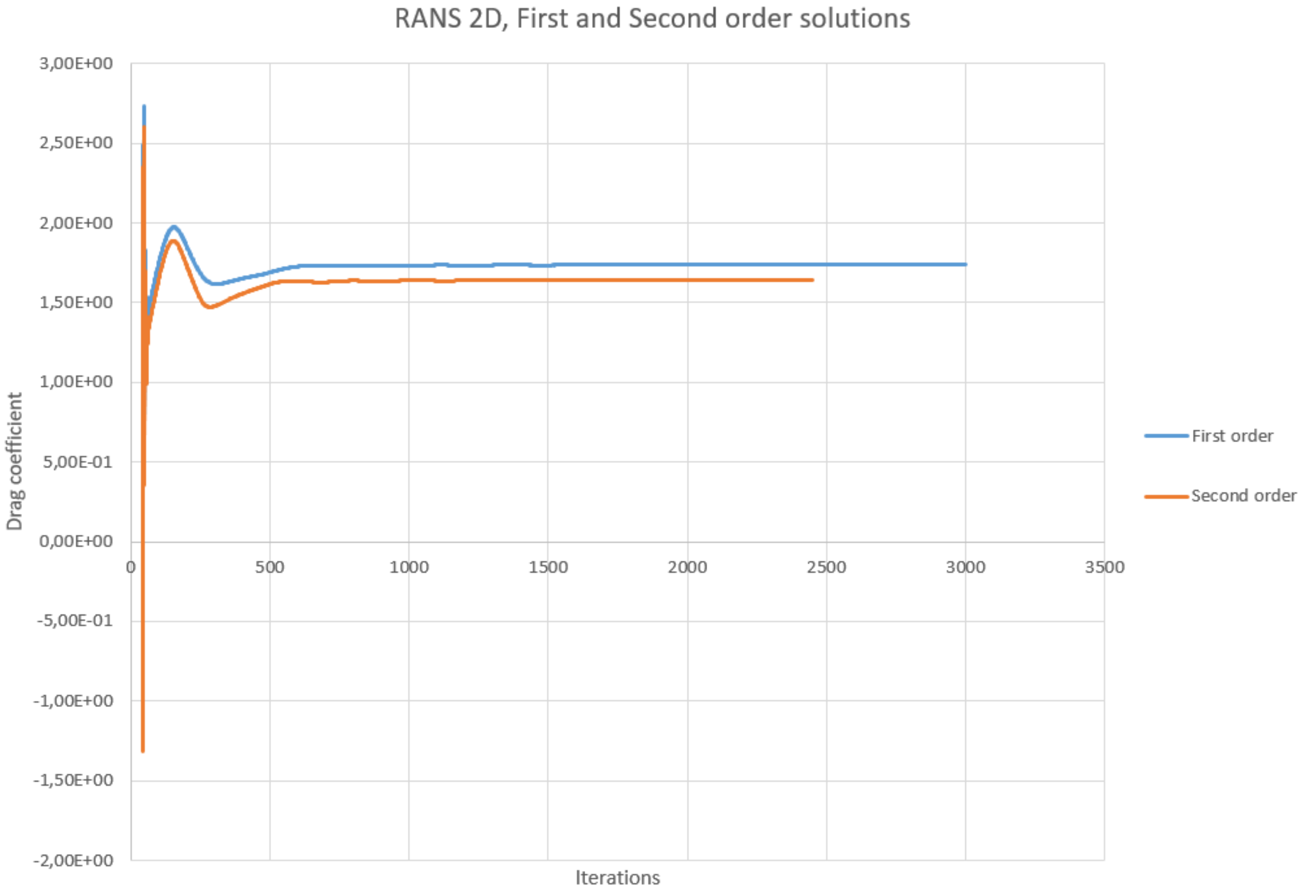


Figure 45: RANS 2D, First order VS Second order solution

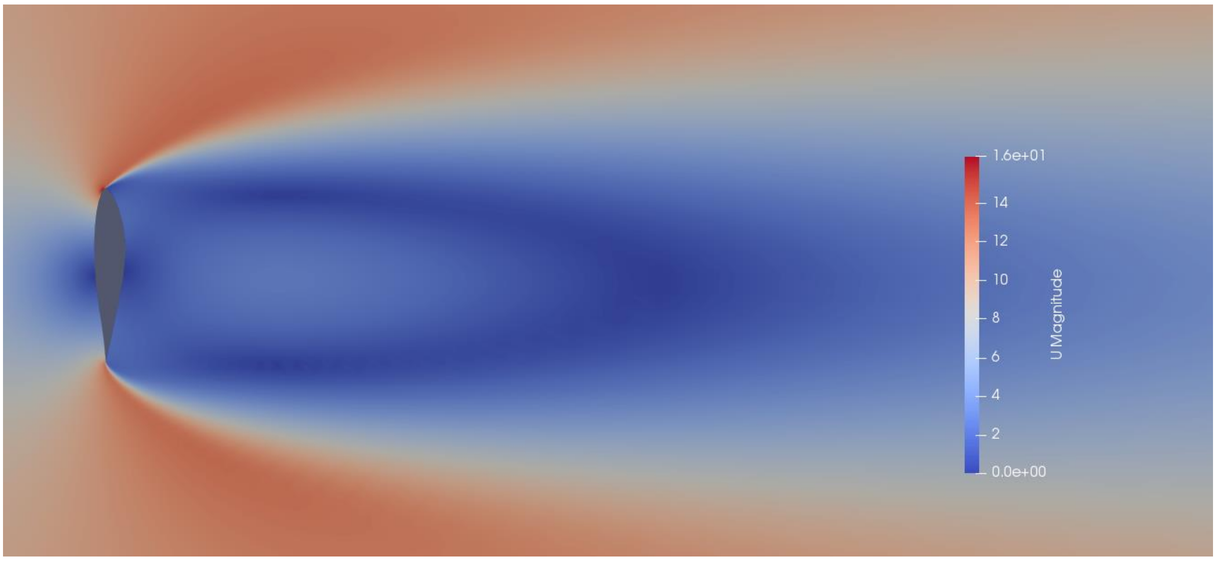


Figure 46: RANS 2D second order

Moving on to the RANS 3D simulations, where the drag coefficients are illustrated in figure 47. Unlike the RANS 2D simulations, the RANS 3D simulations do not converge towards a stable value but rather to a repetitive fluctuating pattern. This an indication that the flow over the airfoil does not stabilize in the same way as for the RANS 2D simulation and is clearly illustrated when comparing figure 46 and 48. Also, on figure 48 there is an indication of a vortex starting to appear on the leading edge. As for the drag, the first order solutions starts to repeat its fluctuating pattern after approximately 2100 iterations with a mean drag coefficient of 1.85, while the second order solutions starts to repeat its fluctuating pattern after approximately 1800 iterations having a mean drag coefficient of 2.25.

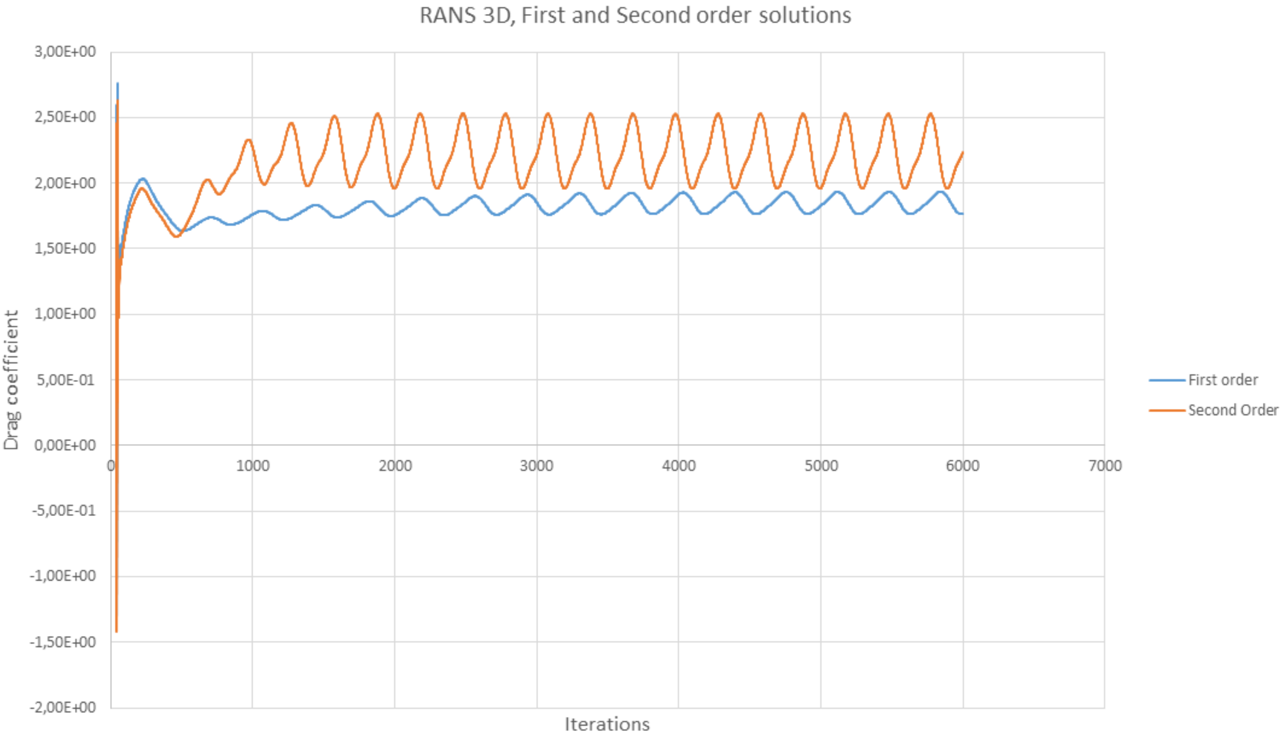


Figure 47: RANS 3D, First order VS Second order solution

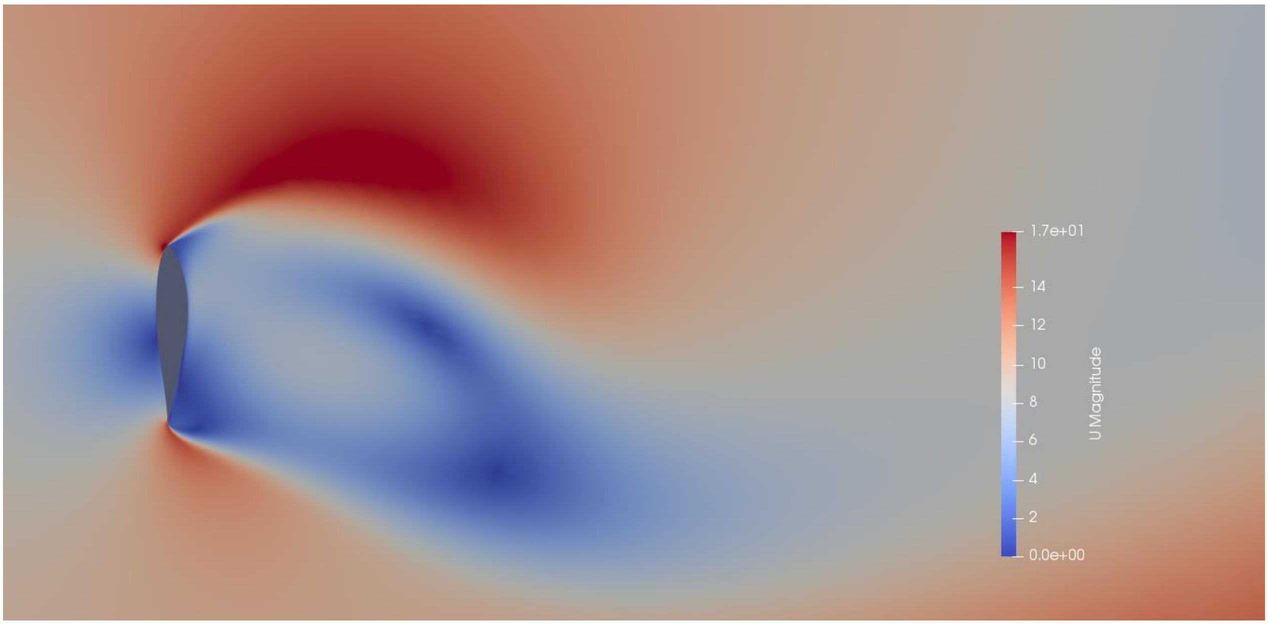


Figure 48: RANS 3D second order

Unlike RANS simulations that are steady state, the URANS simulation represents a transient solver of the flow. As described in chapter 5.3.2, this solution is dependent on both time and space, leading to a choice of a sufficient time step. By using *adjustTimeStep* and a *maxCo* (maximum courant number) of 0.9, openFOAM will automatically change the time step in such a way that it does not exceed the *maxCo*. The disadvantage with this method is that the computational time necessary for completing the simulation can be high as the time step tends to be very low. The first order solution needed approximately 245000 iterations to reach convergence, which is at approximately 3 on the x-axis in figure 49. Moreover, the second order solution started to reach repetitive fluctuations after 560000 iterations, which is at approximately 7 on the x-axis in figure 49. Calculating the mean drag coefficient for the second order solution after reaching repetitive fluctuations gave a value of 2.94, while the converging value of the first order solution was 1.65.

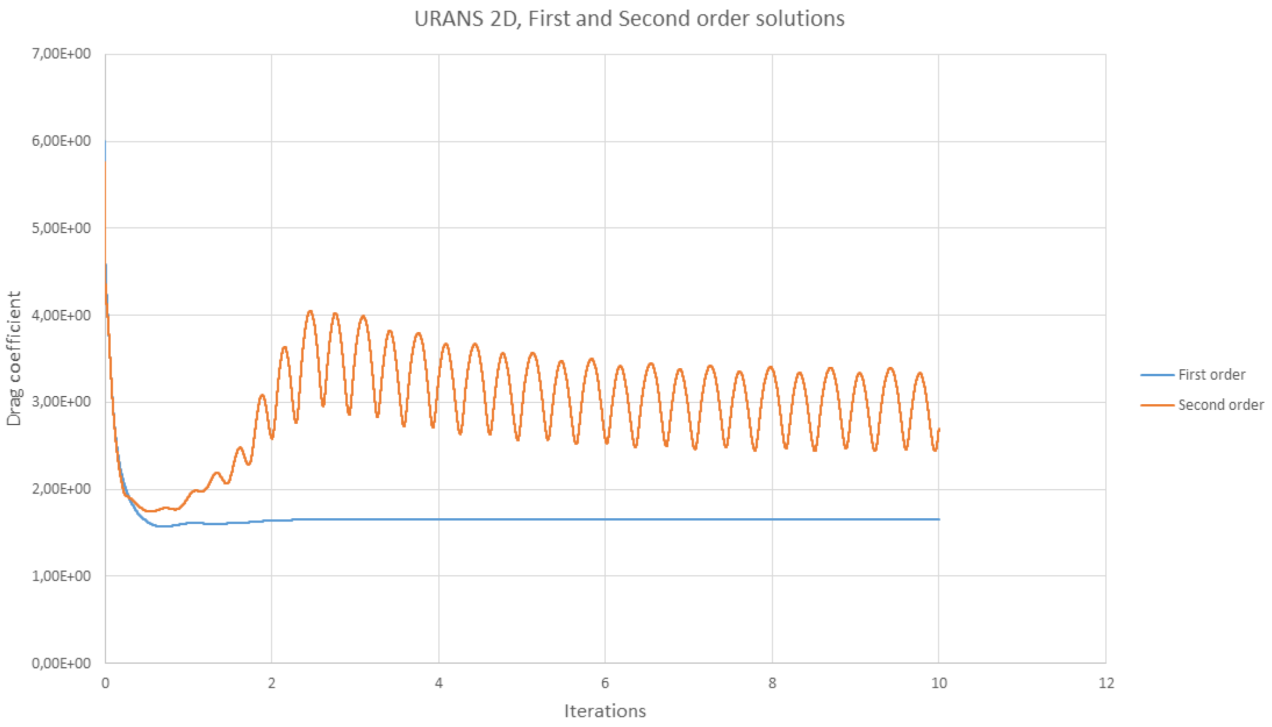


Figure 49: URANS 2D, First order VS Second order solution

Unlike RANS, the difference between the first and second order solution for the URANS 2D simulation is huge. The first order solution has a stable drag coefficient, hence indicating a stable flow pattern. However, the second order solution always has a changing drag coefficient, which is an indication of a more unstable flow pattern. This difference can clearly be visualized by comparing figure 50 and 51. Figure 51 has a turbulent flow with a clear vortex at the trailing edge. As discussed in chapter 5.6.3, the second order solutions tends to be more accurate, which should be the explanation on why the second order URANS solution has a greater ability to predict the turbulent flow.

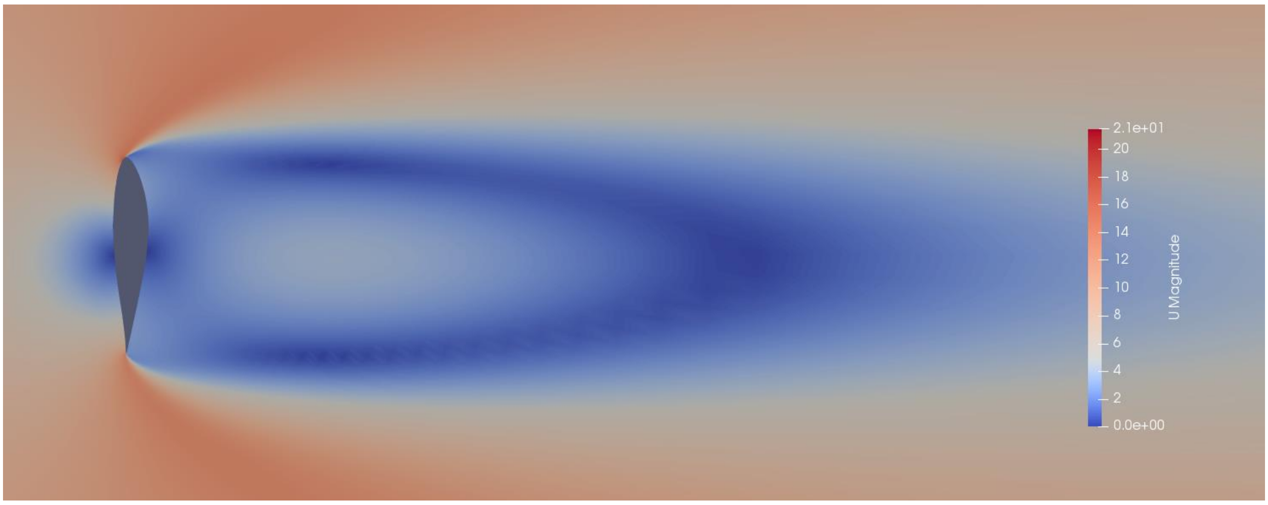


Figure 50: URANS 2D first order solution

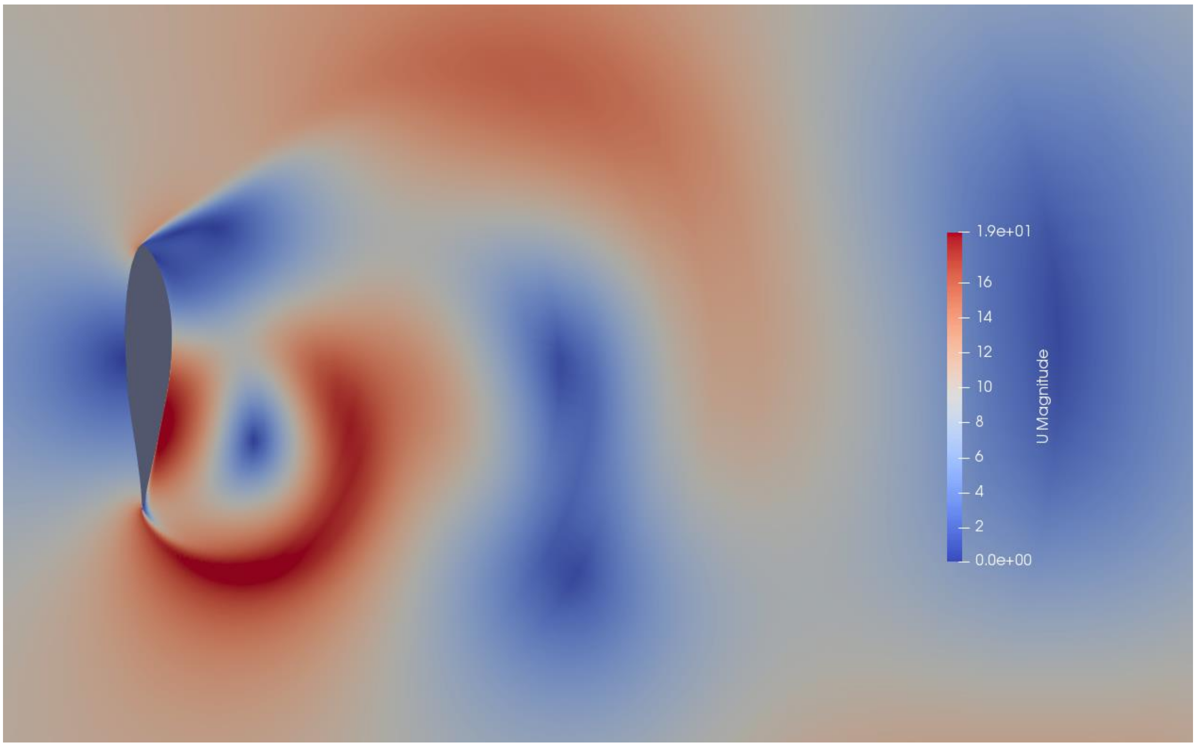


Figure 51: URANS 2D second order solution

When working with the URANS 3D simulation, some difficulties were encountered. By using the same method as for the 2D case, namely the *adjustTimeStep* and a *maxCo* number of 0.9, the simulation encountered some major problems with the time step being too low. Because of this, a fixed time step of 10^{-4} was selected in order for the simulation to finish up. In addition, both the first order and second order URANS 3D simulation was started from an already converged first and second order RANS simulation. This will avoid the URANS simulations to spend time on iterating down from high values in the beginning, hence making the simulation time shorter.

The second order solution started to have repetitive fluctuations after 29800 iterations, which is approximately 1403 on the x-axis. Moreover, it was stopped after reaching 50000 iterations since the fluctuations was clearly repeating. The mean drag coefficient was calculated to be 2.63. The simulation needed a little more than 6 weeks to reach this point, which illustrates the need for using a fixed time step, as well as starting the simulation from a converged RANS simulation. As for the first order solution, the drag coefficient seems to have the same stable pattern as for the 2D simulation, again indicating that the first order solution has problems predicting the turbulent flow. The drag coefficients are illustrated in figure 52, while the difference in flow pattern for the two solutions are illustrated in figure 53 and 54.

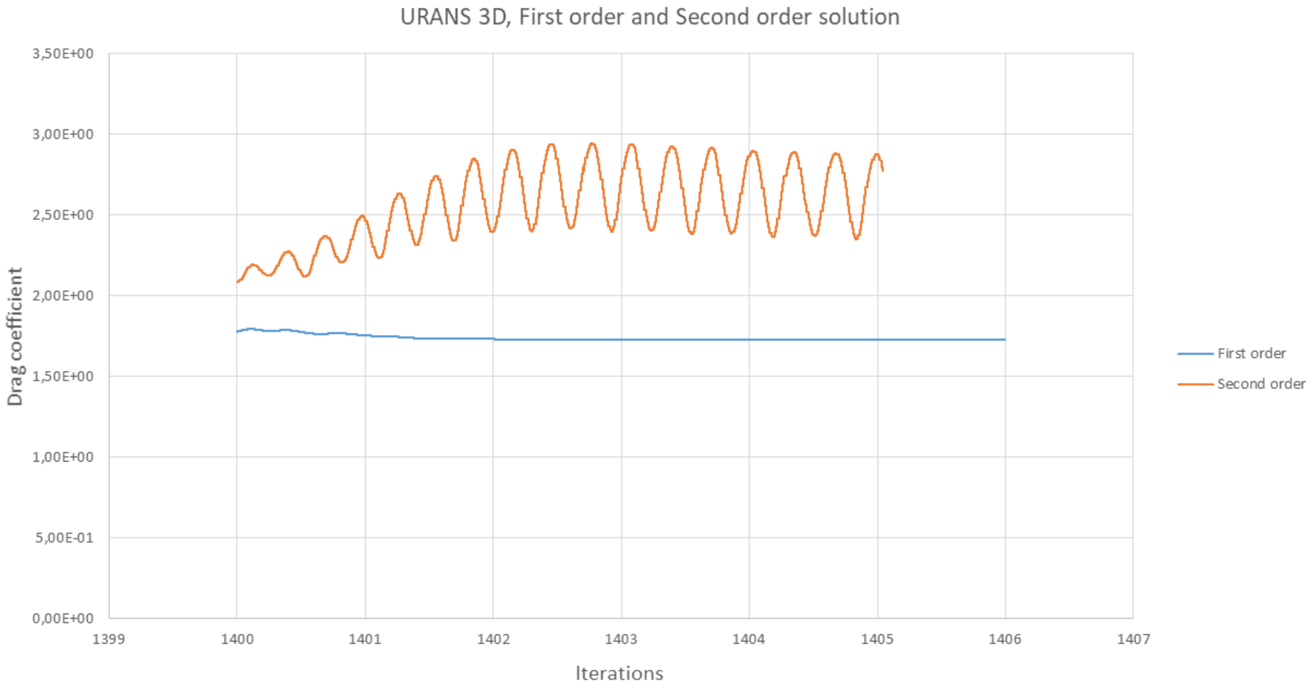


Figure 52: URANS 3D, First order VS Second order solution

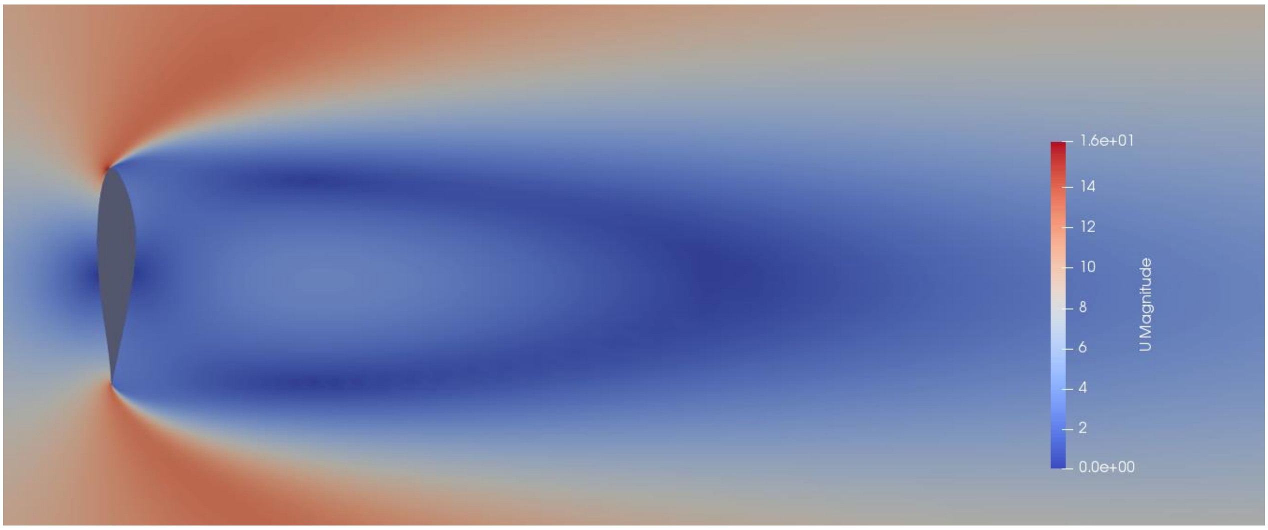


Figure 53: URANS 3D first order solution

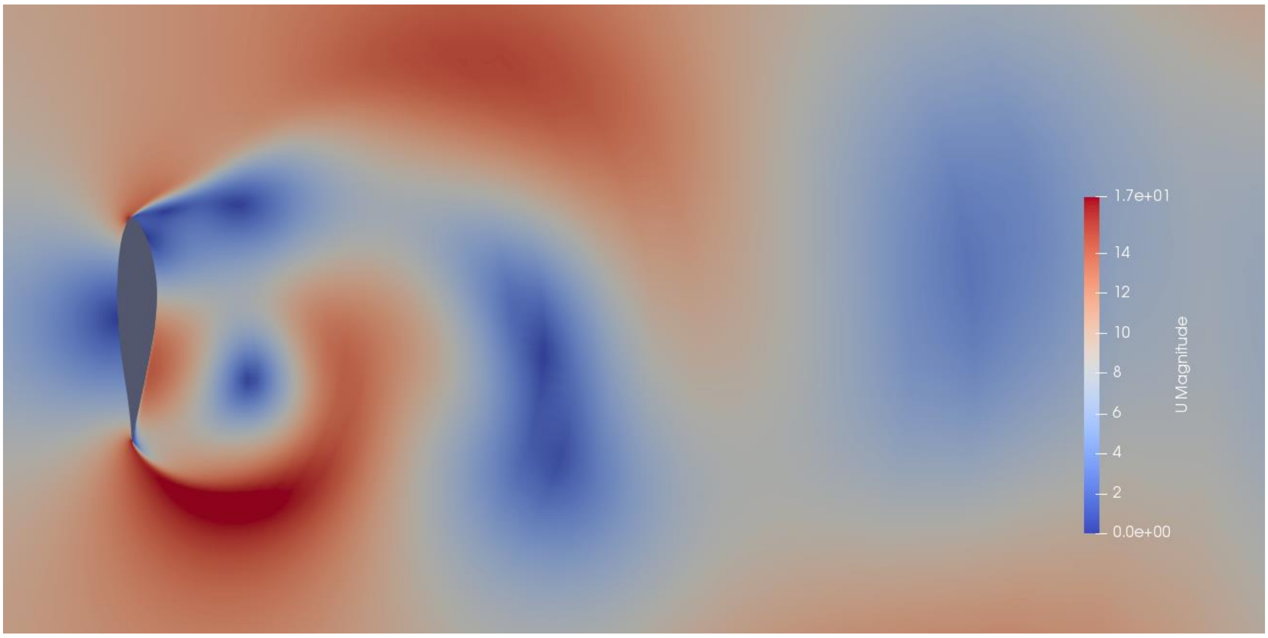


Figure 54: URANS 3D second order solution

In order to summarize the results this far, an overview is given in table 13. The first order solution for the transient simulations tends to have problems predicting the turbulent flow, hence supporting the statement on second order solutions being more accurate. Therefore, only second order solutions will be considered for the rest of the thesis.

Table 13: Overview of first and second order solutions of the drag coefficient C_d for the DU96-W-180 airfoil at 90°

Solver	First order	Second order
RANS 2D	1.74	1.64
RANS 3D	1.85	2.25
URANS 2D	1.65	2.94
URANS 3D	1.73	2.63

Just like URANS 3D simulation, the 3D DES simulation was also started from an already converged 3D RANS simulation, as it requires a high memory capacity. The same time step was selected as for the URANS 3D. The simulation started to reach repetitive fluctuations at 65000 iterations, which is approximately at 1406.5 on the x-axis, giving a mean drag coefficient of 2.6. The turbulent flow can be visualized in figure 56 and 57, having vortices at both the leading and trailing edge.

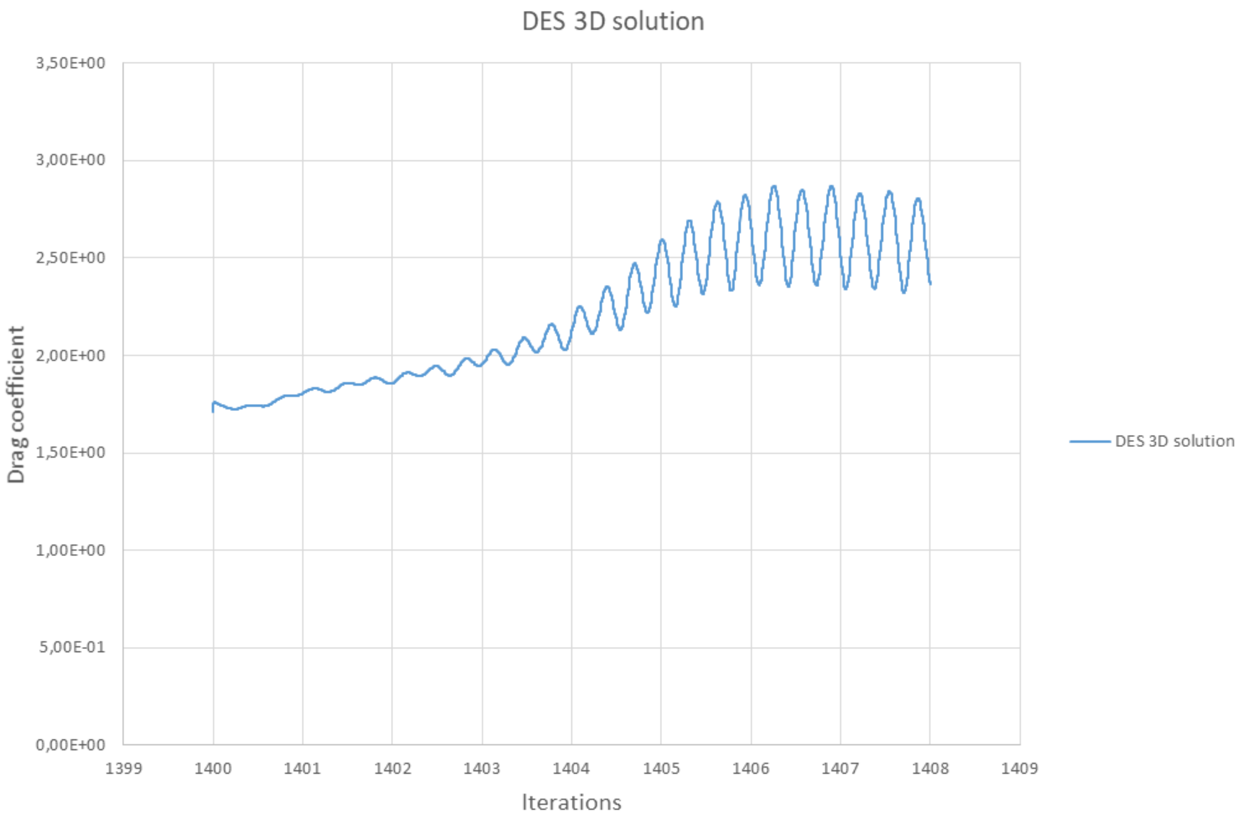


Figure 55: Drag coefficients for the 3D DES solution

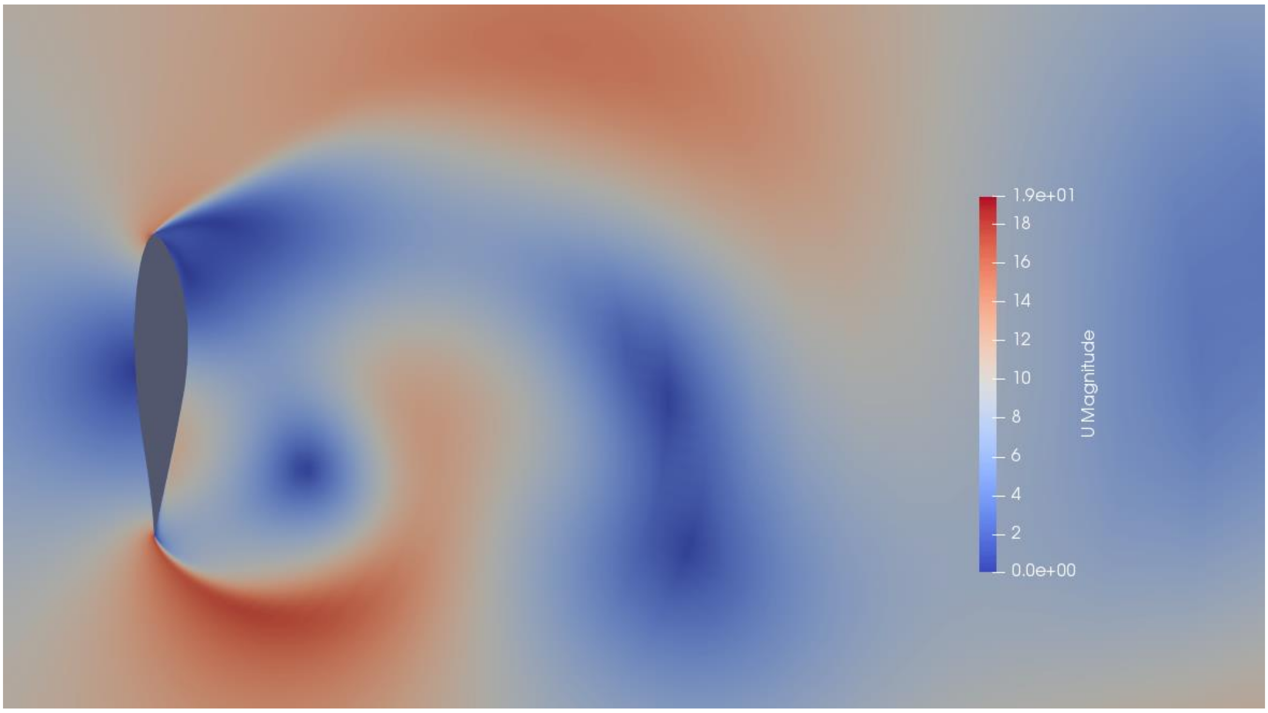


Figure 56: 3D DES solution

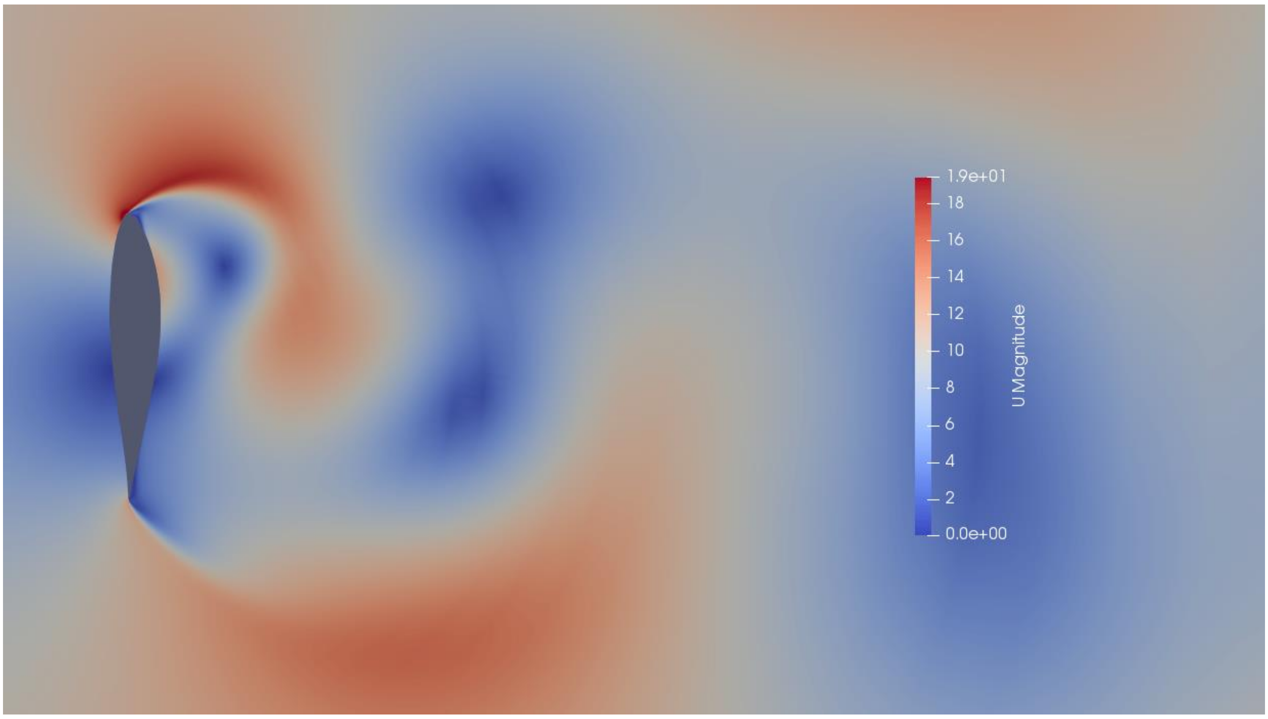


Figure 57: 3D DES solution

As descirbed in chapter 5.8, Xoil is a simplified way of estimating lift and drag for airfoils. Using Xfoil to estimate the drag coefficients from -180° to 180°, the highest drag coefficient appears at 90° degrees and is calculated to be 1.8. An overview is given in figure 58. Both the plot and maximum drag fits well with the results given in figure 4.

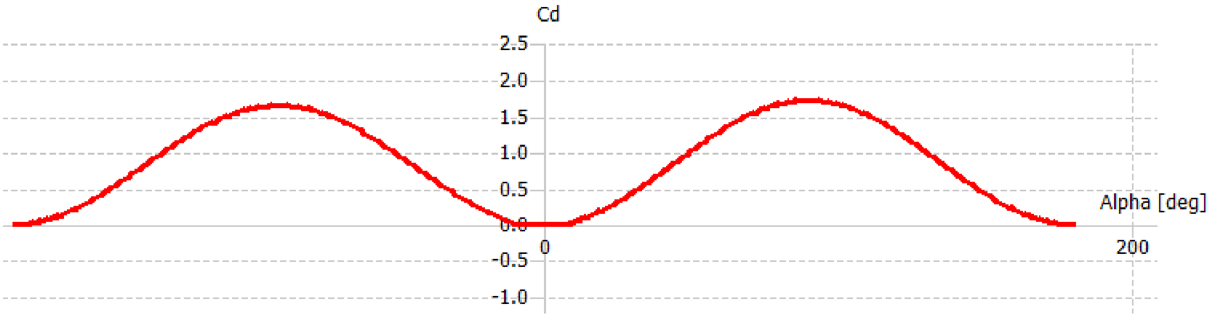


Figure 58: Drag coefficients in Xfoil for the DU96-W-180 airfoil from -180° to 180°

For the PyFR simulation, the foundation was knowledge and case setup obtained from chapter 6. Using the same setup and boundary conditions and changing only the Reynolds number, a simulation for the DU96-W-180 airfoil was performed. As discussed earlier, it is recommended reducing the number of cells for a large mesh before running PyFR due to multiple solutions points per cell. The final mesh in PyFR is given in figure 59.

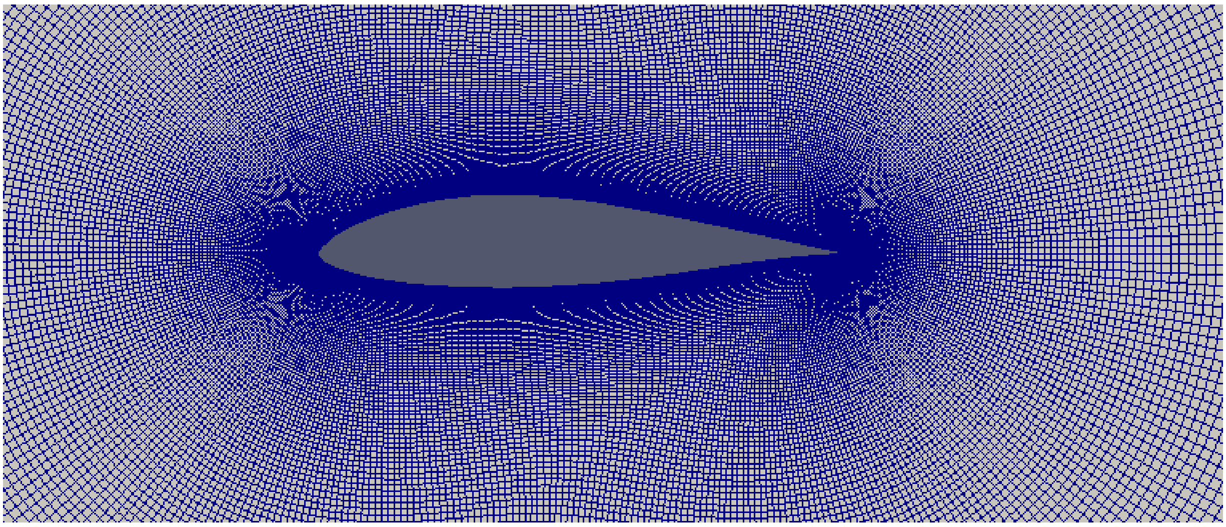


Figure 59: The mesh for the PyFR simulation

The drag coefficient for the PyFR simulation is visualized in figure 60. Please note that the x-axis representing iterations is a scaled number as it makes the visualization easier. The simulation was stopped after 306000 iterations, which is 0.58 on the x-axis. An average drag coefficient was estimated to be 2.1. The flow over the airfoil is illustrated in figure 61, where some clear vortices are developed. Having a closer look, one can detect small vortices developing at the leading and trailing edge, where they increase in size as they move away from the airfoil.

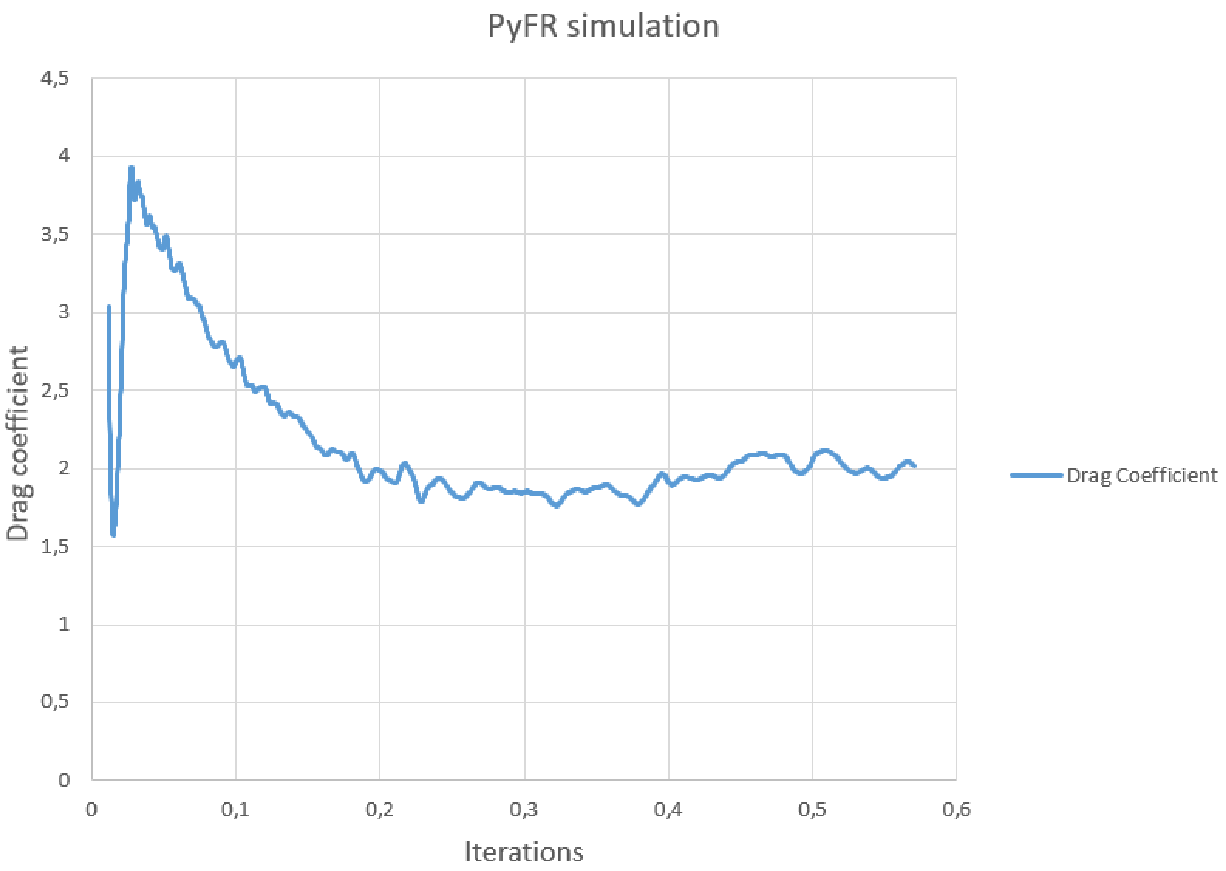


Figure 60: Drag coefficient for the PyFR simulation

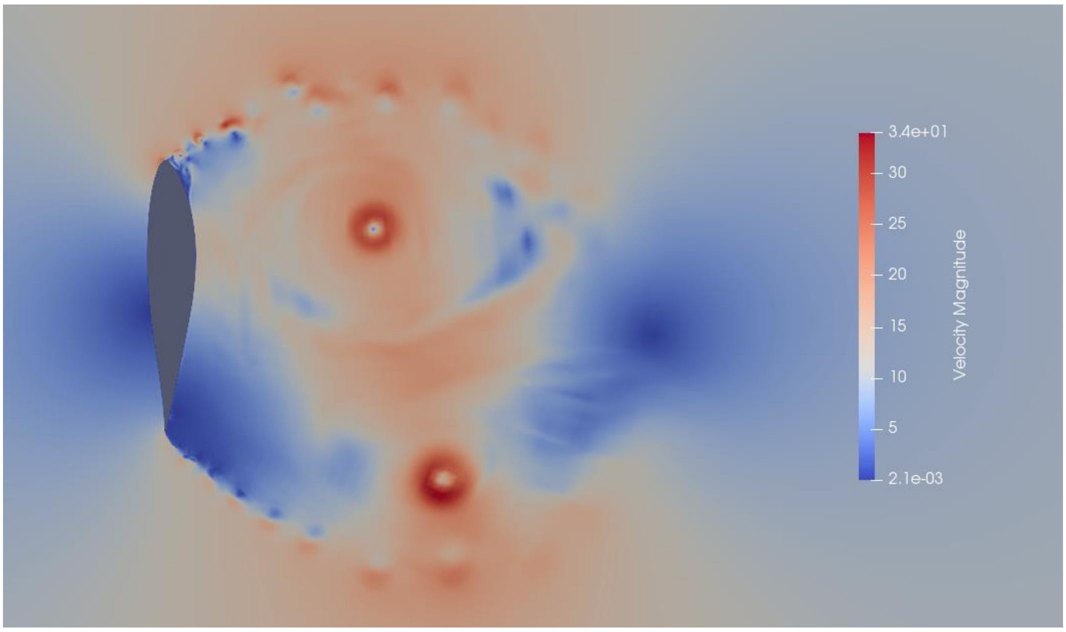


Figure 61: PyFR simulation

A complete overview of the results are given in table 14 for comparison.

Table 14: Overview of the drag coefficient C_d for the DU96-W-180 airfoil at 90°

Solver	Hjorteland 2020	Hjorteland 2019	Sørensen and Skrzypiński 2016	Timmer and van Rooij 2001
Experimental				1.914
RANS 2D	1.64	1.54	3.1	
RANS 3D	2.25		2.8	
URANS 2D	2.94	2.9		
URANS 3D	2.63	2		
DES 3D	2.6	2.4	2.3	
Xfoil	1.8			
PyFR	2.1			

7.3. Discussion

After analyzing the results, there is a trend on the drag coefficients from the CFD simulations being higher than the experimental result. The same can be said about the simulation results obtained by Sørensen and Skrzypiński. This indicates that it is difficult to estimate the flow and forces with a high precision on an airfoil at a 90° angle of attack using second order solutions in OpenFOAM. Even though Xfoil has one of the best drag coefficient prediction when comparing with the experimental value, it does not provide as much information as the CFD simulation, especially on predicting the turbulent flow behind the airfoil. Moreover, as the drag coefficient is calculated based on a flat plate and not the airfoil itself, it should only be considered as an estimation and not an accurate result. Furthermore, the RANS 3D simulation is the best match with the experimental value out of the OpenFOAM simulations. Finally, the PyFR simulation, providing a higher order solution, has a good estimation of the drag coefficient compared to the experimental value. The drag coefficients that are plotted in figure 60 does not have a stable repeating pattern and it would have been more optimal to run the simulation for a longer time in order to be sure about convergence. However, this could be an indication that higher order solutions actually have a greater opportunity of solving aerodynamic problems involving a highly separated flow compared to second order solutions.

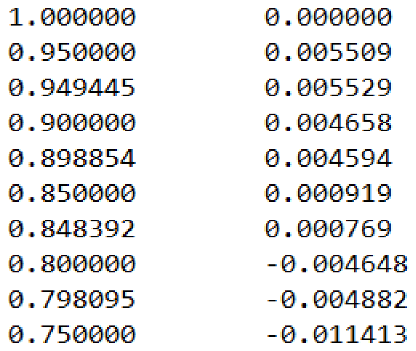
8. The SNL100-03 Turbine Blade

Sandia National Laboratories (SNL) are located in the United States of America and has several working areas where energy is one of them. SNLs Wind Energy Technologies Department has been creating and evaluating new designs for horizontal axis wind turbines. During the last decade, they have focused on creating a 13.2MW wind turbine with a blade length of 100m (Griffith & Ashwill, 2011). This research has resulted in the SNL 100 turbine series, where the SNL100-03 turbine blade has been selected for this project. The blade itself is described in chapter 4.4.

8.1. Pre-Processing

8.1.1. Creation of the SNL100-03 Turbine Blade

The first step in process of performing CFD-simulation on the SNL100-03 turbine blade is to create a model of the blade. In table 6, a recipe is given for creating a model of the turbine blade. By looking at the column called *"Airfoil description"* furthest to the right, the names of the different airfoils used in the turbine geometry are listed. The geometry for all the airfoils are available online and can be downloaded as a set of coordinates. As these coordinates are plotted, they will form the shape of a 2D-airfoil. An example for such a list of coordinates is given in figure 62. Note that this is not the complete list of points, but rather a brief example.



1.000000	0.000000
0.950000	0.005509
0.949445	0.005529
0.900000	0.004658
0.898854	0.004594
0.850000	0.000919
0.848392	0.000769
0.800000	-0.004648
0.798095	-0.004882
0.750000	-0.011413

Figure 62: Airfoil coordinates

Qblade has the ability to import these lists of coordinates and plot them in 2D. This process is fairly straightforward and the complete plot for all the airfoils forming the total geometry for the turbine blade is given in figure 63.

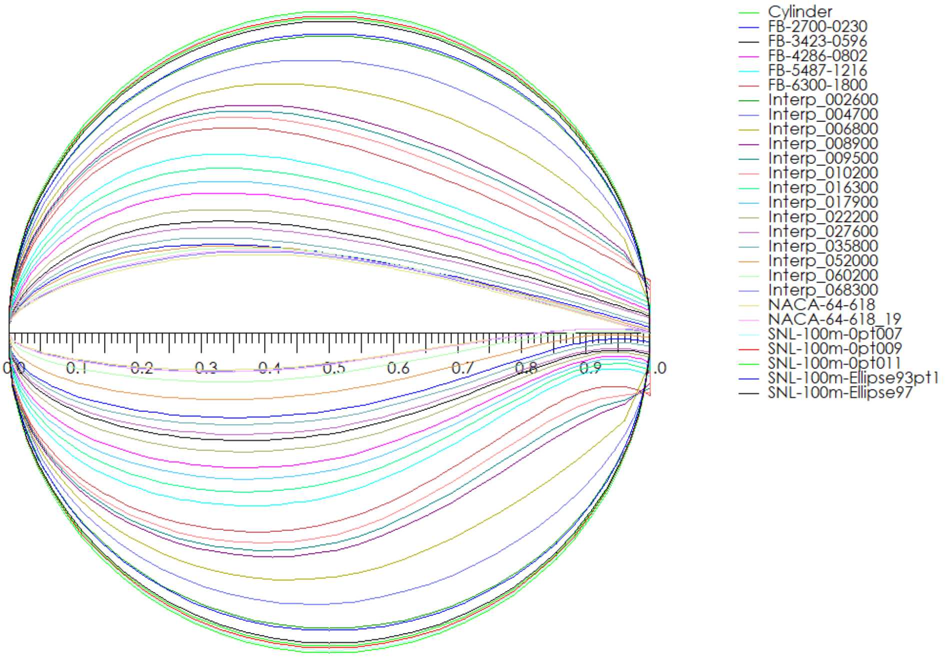
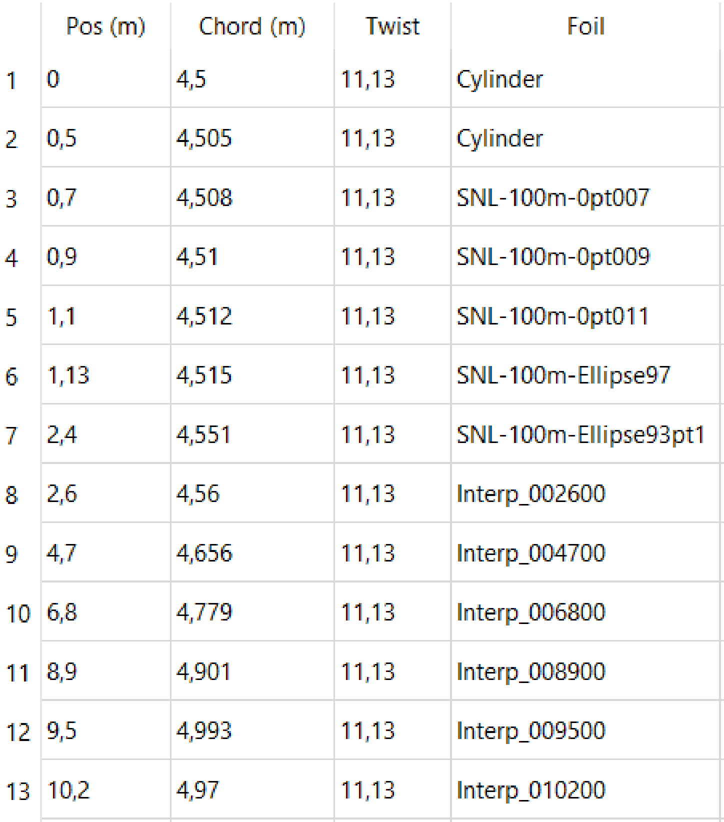


Figure 63: All the different cross-sections for the SNL100-03 turbine blade

Furthermore, the cross-sections can be adjusted individually according to the given information in table 6. Qblade has the ability to select a starting location in the *"Pos (m)"* column. From there, one can determine the next starting point for each cross-section all the way to the tip. Moreover, the *"Chord (m)"* and the *"Twist"* are specified for all the foils. An example of the first thirteen cross-sections are illustrated in figure 64.



	Pos (m)	Chord (m)	Twist	Foil
1	0	4,5	11,13	Cylinder
2	0,5	4,505	11,13	Cylinder
3	0,7	4,508	11,13	SNL-100m-0pt007
4	0,9	4,51	11,13	SNL-100m-0pt009
5	1,1	4,512	11,13	SNL-100m-0pt011
6	1,13	4,515	11,13	SNL-100m-Ellipse97
7	2,4	4,551	11,13	SNL-100m-Ellipse93pt1
8	2,6	4,56	11,13	Interp_002600
9	4,7	4,656	11,13	Interp_004700
10	6,8	4,779	11,13	Interp_006800
11	8,9	4,901	11,13	Interp_008900
12	9,5	4,993	11,13	Interp_009500
13	10,2	4,97	11,13	Interp_010200

Figure 64: The design process for the SNL100-03 turbine blade

The final results, which is a 3D model, can be investigated further to check for potential errors. From there, the turbine blade is exported as an STL-file. Note that the black lines over the turbine blade symbolizes a change in cross-section.

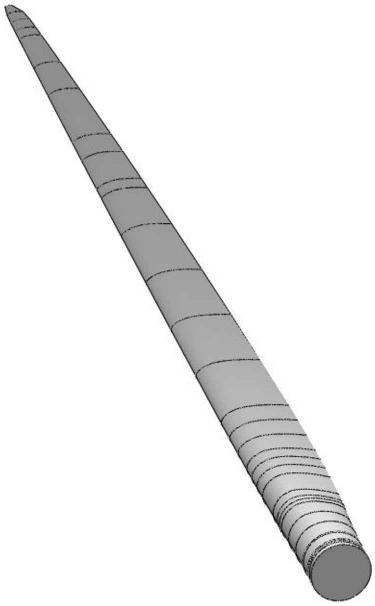


Figure 65: 3D view of the SNL100-03 turbine blade in Qblade

8.1.2. Mesh Generation

The mesh generation for the SNL 100-03 turbine blade was done differently compared to the DU96-W-180 airfoil as a direct consequence of the COVID-19 Virus. Meaning, since the University of Stavanger was closed for a certain period, Pointwise was unavailable. Nevertheless, a suitable substitute was selected to be a combination of the blockMesh tool in Blender and snappyHexMesh in OpenFOAM, both being free to use without a license.

As a start, the STL-file was imported into Blender. Since Qblade originally creates a horizontal axis wind turbine with three blades, it can be hard to determine the location for a single blade in a given coordinate system. In Blender, the turbine blade was relocated to the origin. Knowing the exact location of the 3D model will do the mesh generation easier and more efficient.

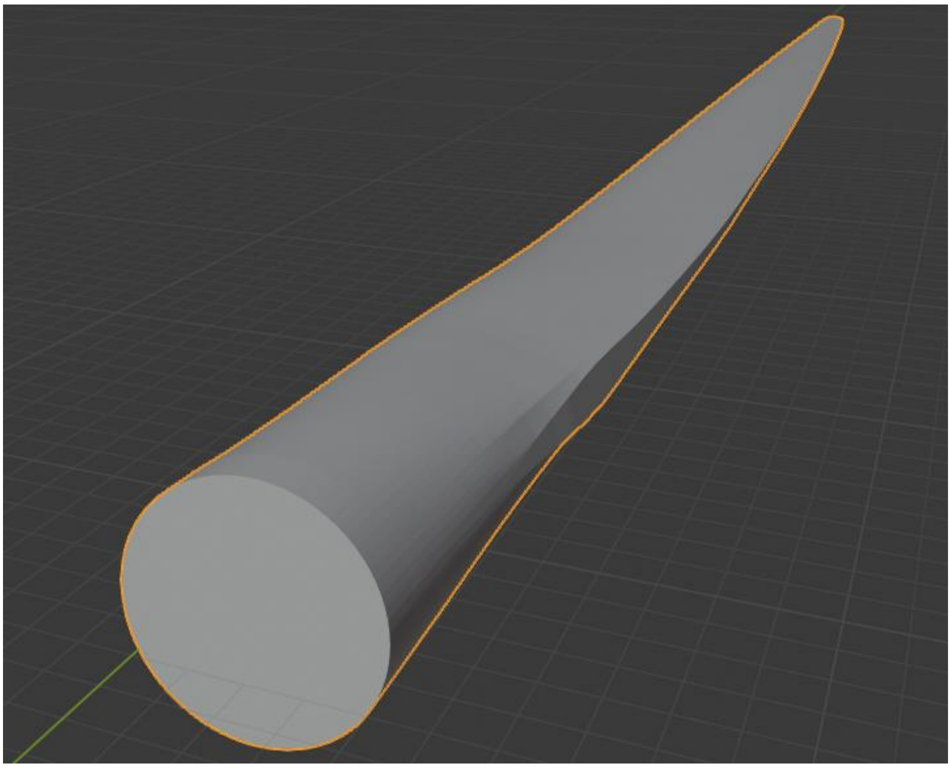


Figure 66: The turbine blade in Blender

A blockMesh domain was then created for the turbine blade, having the shape of a box with a length of 90m, a width of 30m and a height of 120m. The next step was then to use snappyHexMesh in order to mesh the turbine blade inside the already existing blockMesh domain. This part proved to be harder than expected since snappyHexMesh recreates the turbine blade when meshing, meaning that a high refinement level is crucial for the final turbine blade to not lose its shape and have a smooth geometry. Another problem that was encountered was the transition from large cells to smaller cells. However, this problem was solved using *nCellsBetweenLevels*, which makes this transition smoother by adding cells at areas where there is a sudden jump in cell size. In the end, two different meshes, hereby called Course Mesh and Fine Mesh, were tried. Information about the two meshes are given in the following tables:

Table 15: Course Mesh

Type of cells	Number of cells
Hexahedra	21254526
Prisms	171446
Tet Wedges	427
Polyhedra	1287790
Total number of cells	22714189

Table 16: Fine Mesh

Type of cells	Number of cells
Hexahedra	32907130
Prisms	153452
Tet Wedges	27
Polyhedra	1405115
Total number of cell	34465724

The difference between the Course Mesh and the Fine Mesh is the inclusion of layers and a refinement box. Where the Fine Mesh has both layers and a refinement box, the Course Mesh only has a high refinement level of 4-5 for the turbine blade. The purpose of the refinement box is to create smaller cells around a certain object, in this case the turbine blade. The layers are there to generate small cells close surface, resulting in a smaller value for y+. The Course Mesh had It is worth mentioning that an attempt of creating an even more detailed mesh compared to the Fine Mesh was done. However, this led to snappyHexMesh failing due to memory capacity several times. Therefore, it was decided to continue with the Course and Fine Mesh. A cross section of the meshes are shown in figure 67 and 68.

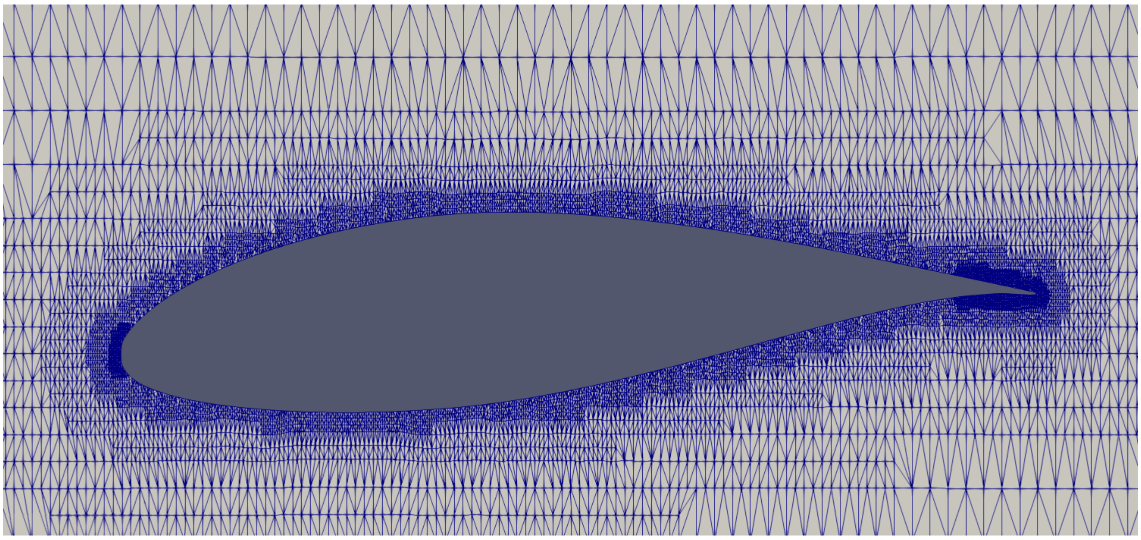


Figure 67: Course Mesh of a cross section at 60m

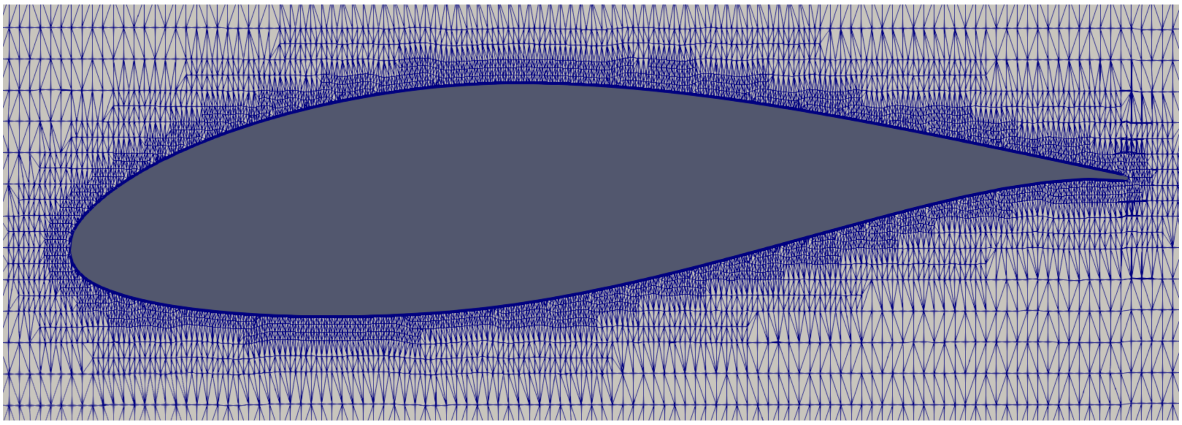


Figure 68: Fine Mesh of a cross section at 60m

A more close up picture of the layers on the Fine Mesh are illustrated in figure 69. Moreover, a picture of the trailing edge of the turbine blade is given in figure 70, illustrating the difficulties with generating such a mesh with snappyHexMesh as it is highly dependent on a high refinement level for recreating the model with a smooth surface.

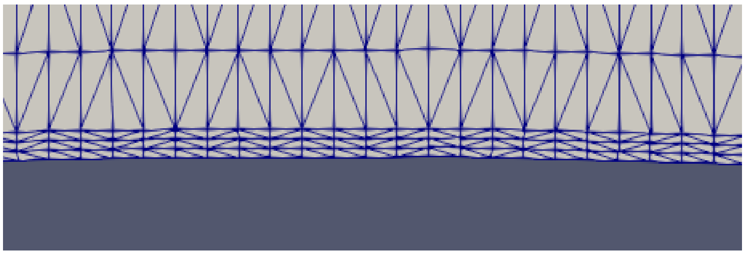


Figure 69: Layers on the Fine Mesh of a cross section at 60m

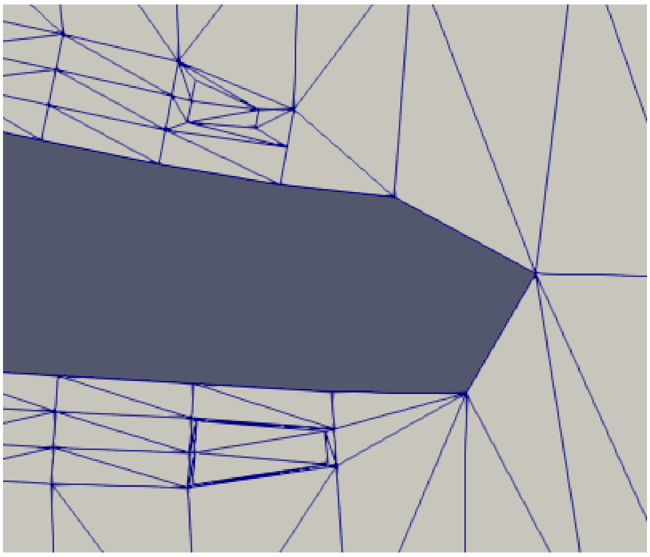


Figure 70: Trailing edge on the Fine Mesh of a cross section at 60m

Since the mesh for the full scaled turbine blade is very large, it will be difficult to run something else than a RANS simulation (*simpleFOAM*). In addition, having access to the UiS Cluster made it possible to run the simulation within an acceptable period of time. The simulation was performed with the following setup, where the Course Mesh and Fine Mesh were having an average y+ of 573 and 83 respectively for the simulation using 37.5 m/s:

Table 17: Important parameters for the SNL100-03 turbine blade simulation

Angle of attack	90°	Degrees
Velocity, u	37.5 , 42.5 and 50	$\mathrm{m/s}$
Density, ρ	1.225	$\mathrm{kg/m^3}$
Kinematic viscosity, ν	$1.470 \cdot 10^{-5}$	$\mathrm{m^2/s}$
Dynamic viscosity, μ	$1.802 \cdot 10^{-5}$	$\mathrm{kg/m \cdot s}$
Reference area, A_{ref}	442.3	$\mathrm{m^2}$

The case setup for the RANS 3D simulation presented in chapter 7 was used as a starting point for the case setup for the SNL 100-03 turbine blade. A new mesh required new boundary conditions, where the turbine blade was defined as *wall* where the *no-slip* condition applies, the inlet and outlet was defined as *patch*, while the sides where defined as *wall* where the *slip* condition applies.

8.2. Post-Processing

The following six simulations needed approximately 6 weeks on the UiS Cluster to reach the iteration numbers given on the following figures. The Fine and Course Mesh are plotted against each other in the following figures for the three extreme loading conditions given in chapter 4.5. In addition, the flow over the turbine blade is illustrated for two different cross sections for the loading conditions at 37.5m/s. The reason for only including pictures for the flow at 37.5m/s is that the flow pattern for the other two velocities are very similar and including them is considered unnecessary.

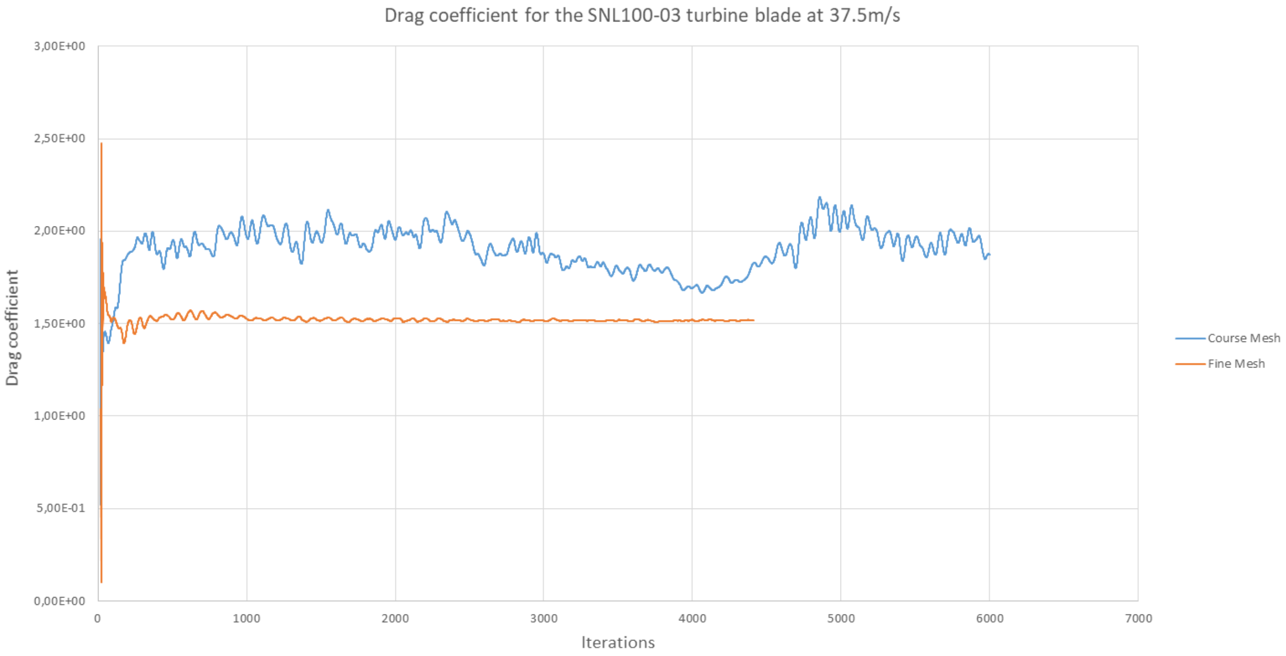


Figure 71: Drag coefficient for the SNL 100-03 turbine blade at 37.5m/s

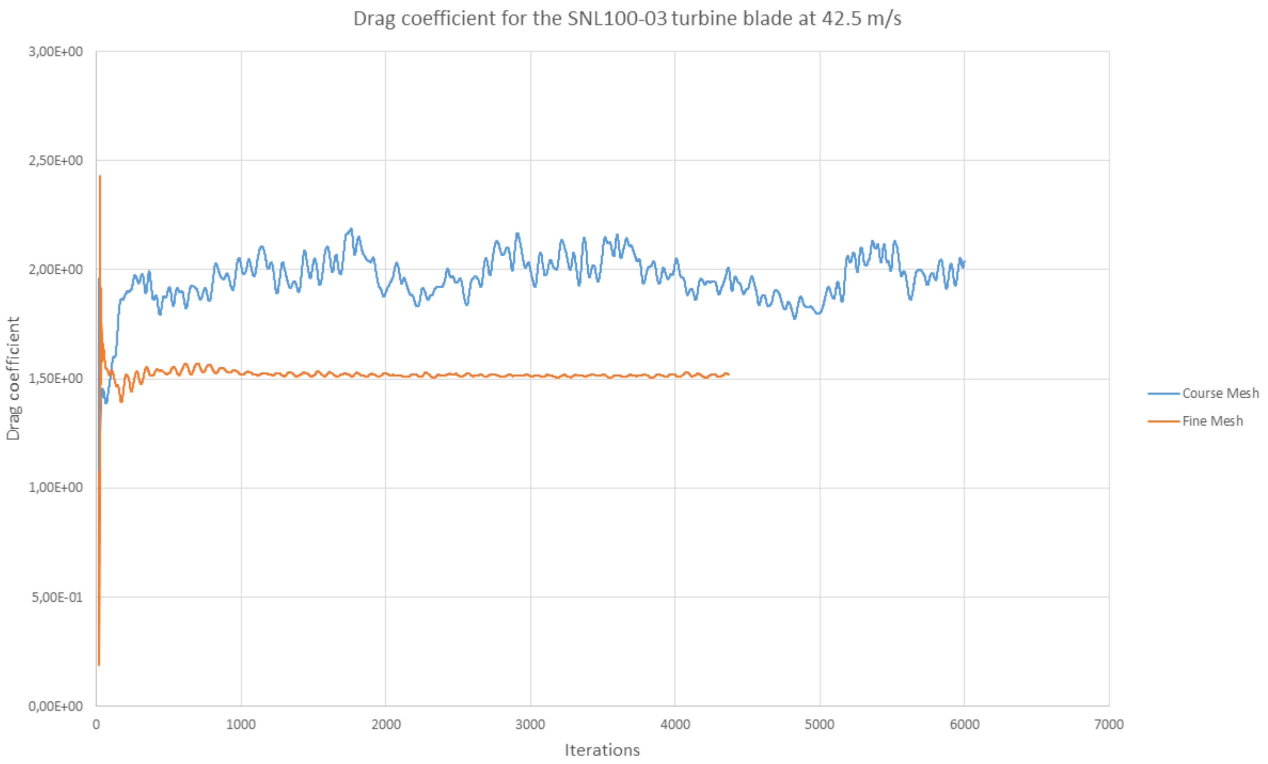


Figure 72: Drag coefficient for the SNL 100-03 turbine blade at 42.5m/s

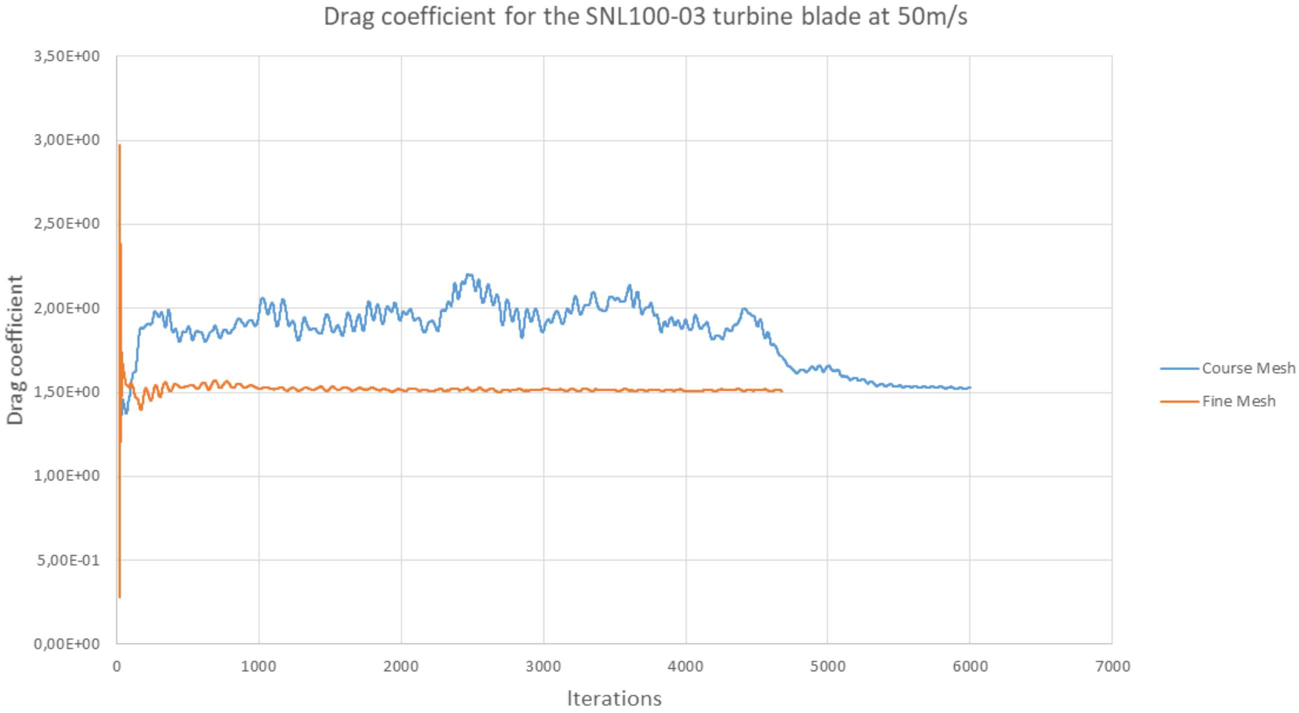


Figure 73: Drag coefficient for the SNL 100-03 turbine blade at 50m/s

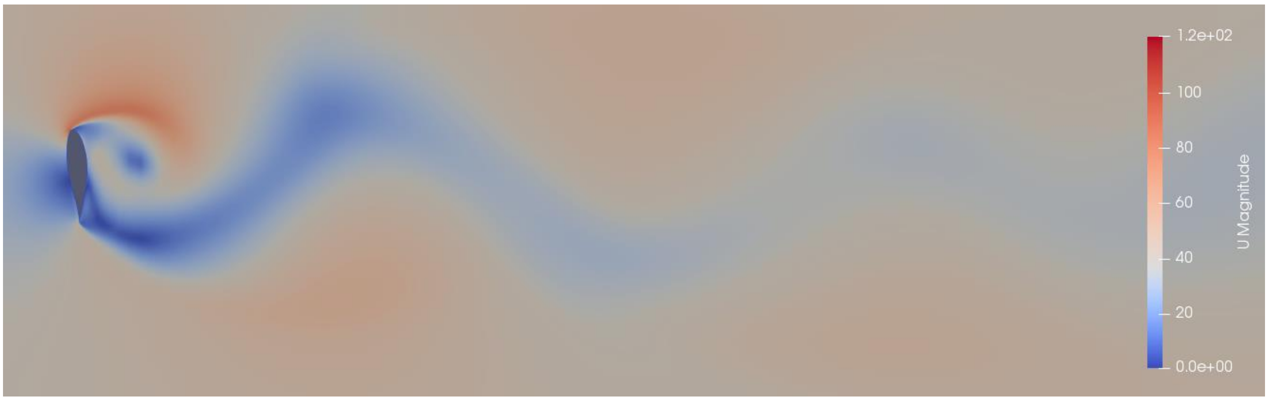


Figure 74: Flow over the turbine blade, Course Mesh, cross section at 60m

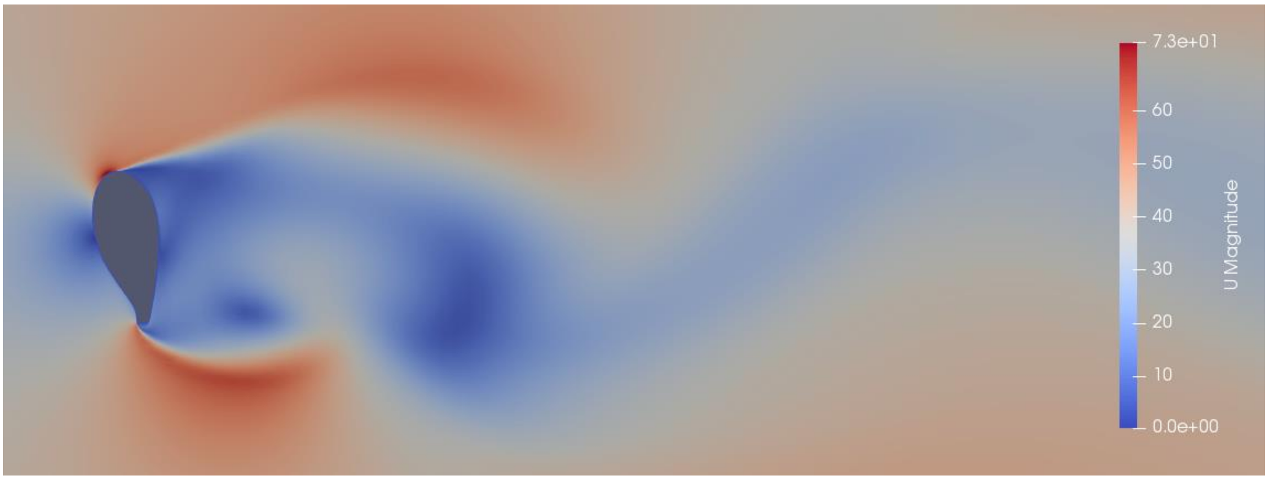


Figure 75: Flow over the turbine blade, Course Mesh, cross section at 20m

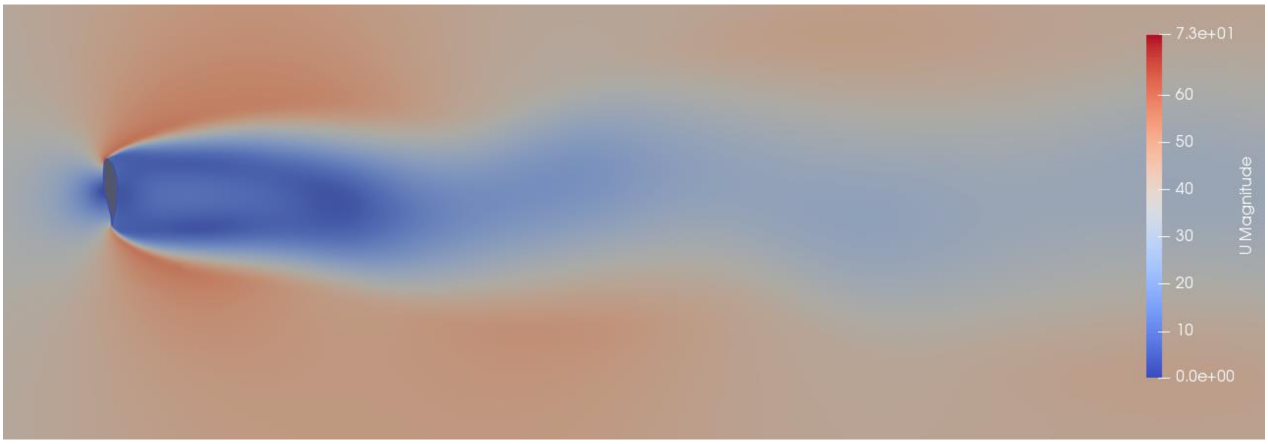


Figure 76: Flow over the turbine blade, Fine Mesh, cross section at 60m

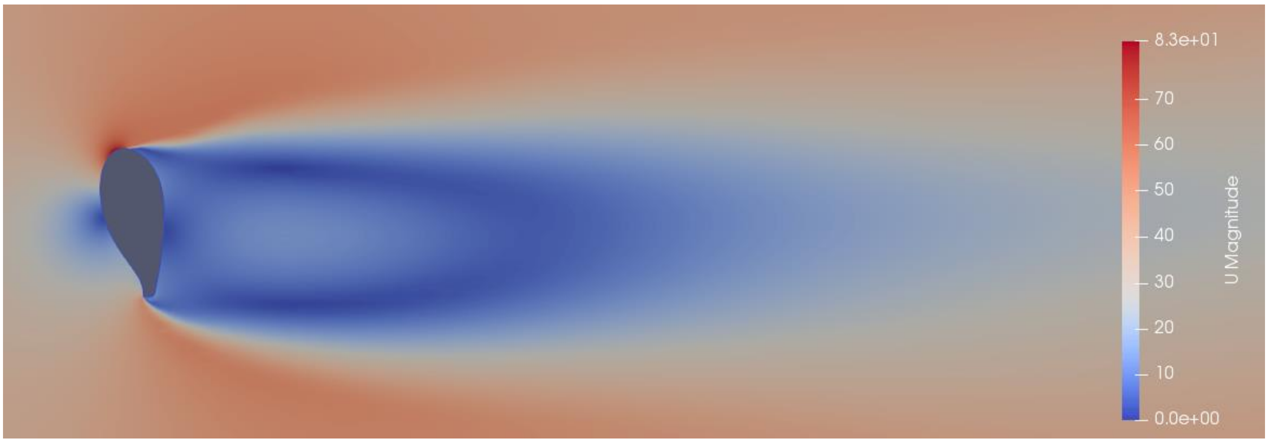


Figure 77: Flow over the turbine blade, Fine Mesh, cross section at 20m

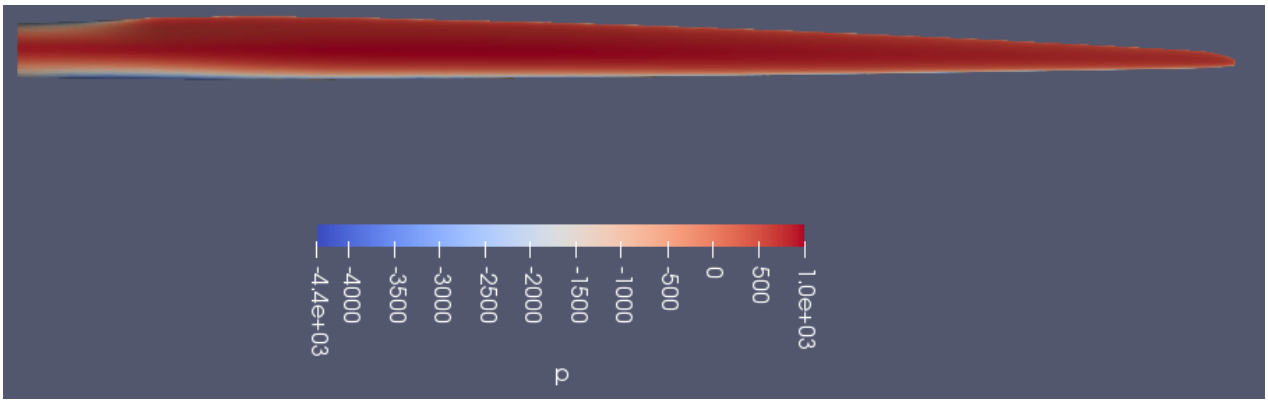


Figure 78: Pressure distribution for the turbine blade at the Course Mesh

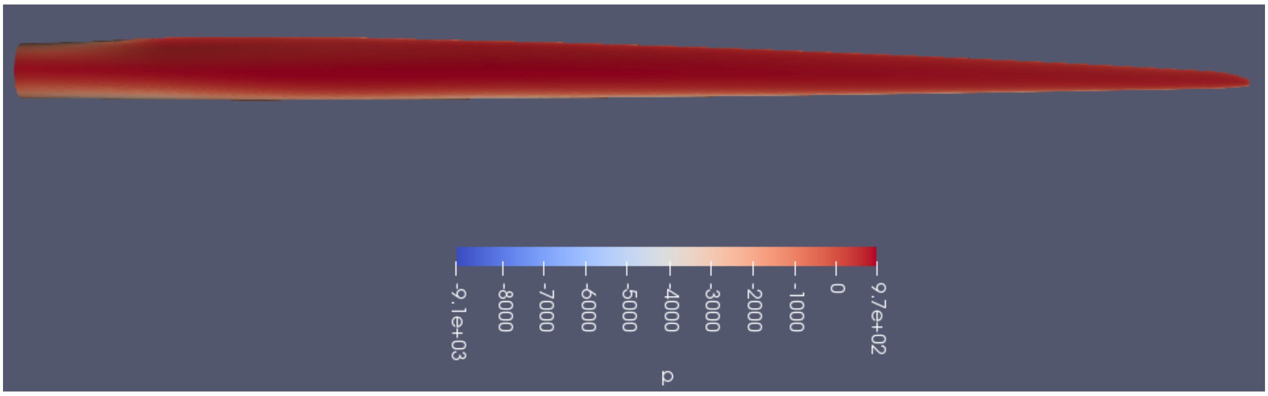


Figure 79: Pressure distribution for the turbine blade at the Fine Mesh

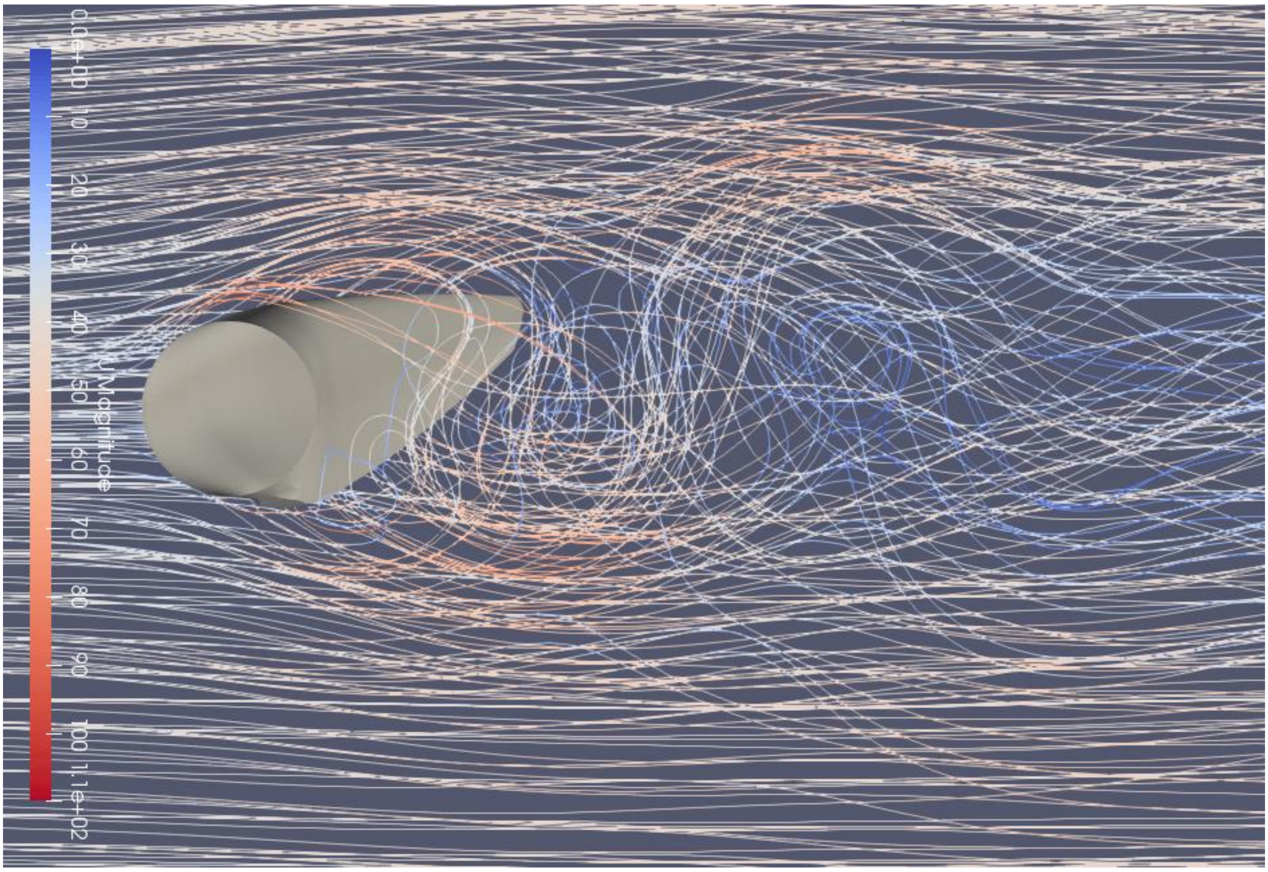


Figure 80: Streamlines for the turbine blade at the Course Mesh

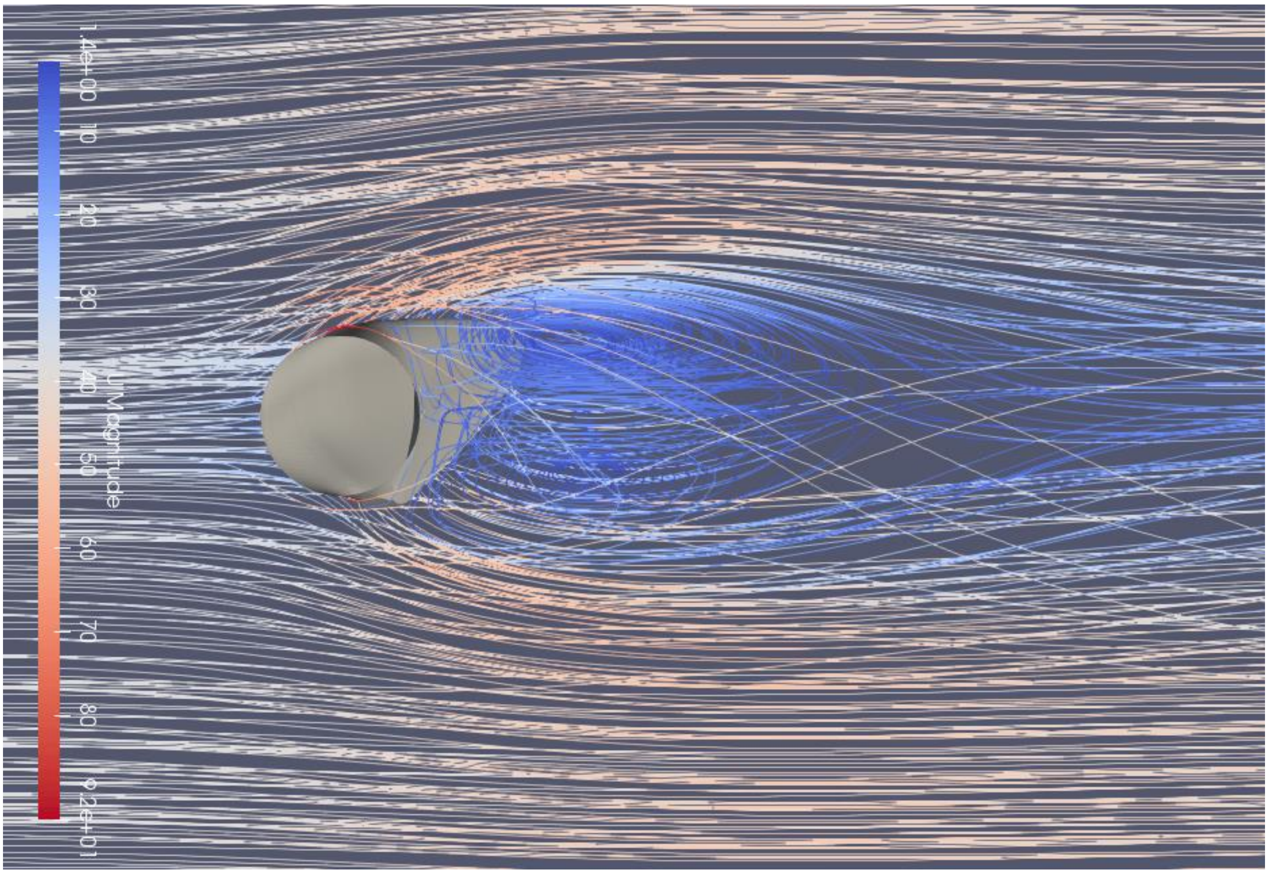


Figure 81: Streamlines for the turbine blade at the Fine Mesh

Table 18: Overview of the average drag coefficient

Velocity	Drag coefficient, Course Mesh	Drag coefficient, Fine Mesh
37.5	1.85	1.52
42.5	2.1	1.52
50	1.54	1.52

Table 19: Overview of the average drag force in Newton

Velocity	Drag force in Newton, Course Mesh	Drag force in Newton, Fine Mesh
37.5	704786.0 N	579067.5 N
42.5	1027590.8 N	743780.0 N
50	1042998.7 N	1029453.25 N

8.3. Discussion

First of all, it was expected that the Fine Mesh would provide more accurate results compared to the Course Mesh. Nevertheless, after the simulations were done it appears that the drag coefficient is the same for all three velocities in the Fine Mesh. This can either be due to the transition from layers to the next cell is not smooth enough resulting in problems estimating the forces correctly, or it could be possible that the increase in velocity increases the drag force to such a degree that the fraction representing the drag coefficient in equation 18 stays the same. In simpler terms, it would be as if one were multiplying both the numerator and denominator with the same number, which of course would result with the same fraction. The plot for the drag coefficients in the Course Mesh appears to be very unstable compared to the Fine Mesh, which has very small repeating fluctuations. Moreover, the flow pattern over the turbine blade for the cross sections at 20m and 60m are strongly comparable with the images obtained for the RANS simulations in chapter 7.2. Moreover, the pressure distribution and streamlines are indicating that the flow is more or less the same for all the 2D cross sections of the blade. However, the fact that there are no available results to compare the simulations with makes it harder to conclude if the results are correct. Without any external results, it would be necessary to create an even finer mesh with smoother transitions between cell sizes, and compare the changes with the already existing results for the Fine and Course Mesh.

On the other hand, the investigation of the ability to perform CFD analysis for a full sized wind turbine blade was achieved. First of all, snappyHexMesh has some major limitations for meshing a turbine blade of this size. Spending a huge amount of cells just to recreate the geometry of the blade leads to a high total number of cells and a long computational time. Even with a high refinement level, the inclusion of layers and a refinement box around the blade, the best y+ value was calculated to be 83 and cannot be considered a fine mesh. The importance of having a good mesh is also seen when looking at how big of an impact there is on the results in table 18 and 19 for different meshes. With Pointwise on the other hand, there is no need to spend a large amount of cells to recreate the geometry, meaning that one can achieve a mesh with the same amount of cells as the Fine Mesh, but with a lower y+ value. This can be considered as a decisive factor for the computational time needed to finish the simulation, as well as the accuracy of the result.

9. Conclusion

This thesis performed an investigation of different CFD solvers and their ability to predict the drag force, presented as the drag coefficient, and the turbulent flow for the DU96-W-180 airfoil at a 90° angle of attack. The results from this investigation indicated that the second order solutions obtained from OpenFOAM had some issues predicting an accurate drag coefficient compared to the experimental result. Interestingly, the results achieved from Xfoil was considered a closer match compared to the experimental result. However, as Xfoil uses values from a thin flat plate for calculations at high angle of attacks, this result should be treated as a simplified solution and is not necessarily valid for all airfoil geometries. Moreover, the higher order solution from PyFR was a close match to the experimental value for both the 2D cylinder introduction case and the DU96-W-180 airfoil. It also proved a great ability to predict the turbulent flow behind both the cylinder and airfoil at a detailed level. The computational time is considered as a challenge for some of the solvers especially the second order 3D URANS simulation, as well as the PyFR simulation.

The thesis also performed an investigation of the ability to perform CFD simulations on a full sized wind turbine blade during extreme loading conditions at a 90° angle of attack, in this case, the SNL 100-03 turbine blade. With the available meshing tool snappyHexMesh, this was proven to be difficult. Nevertheless, two different meshes were tested and simulated using the RANS 3D computational setup from the DU96-W-180 as a foundation for the new computational setup. The computational time needed for the finest mesh was approximately 6 weeks on the UiS Cluster, which has a strong capacity for running simulations. However, it is expected that Pointwise will be able to create a mesh with a lower value of y+ without increasing the total number of cells to a high degree, meaning that a more accurate results should be possible to obtain without increasing the computational time. In other words, there is a potential for CFD to solve extreme loading conditions on a full sized wind turbine blade at a high angle of attack.

For further work, it is suggested to continue investigating the ability of higher order solutions through PyFR to predict the drag coefficient and turbulent flow on aerodynamic problems at high angle of attack, with more accuracy. Moreover, there should be performed an attempt to create a mesh of the SNL 100-03 in Pointwise with a lower y+ value compared to the one in this thesis. In addition, the drag forces achieved through the simulations on the SNL 100-03 blade can, if possible, be compared to the expected forces that are used during the design process of such large wind turbine blades.

References

Autodesk. (2020). Retrieved from https://www.autodesk.co.uk/products/autocad/overview#

Aviator. (2017). Retrieved from https://aviation.stackexchange.com/questions/39340/why-the-dynamic-pressure-is-not-mentioned-in-the-explanation-of-lift-by-bernoull

Awea. (2020). Retrieved from https://www.awea.org/wind-101/basics-of-wind-energy

Blender. (2020). *blender.org*. Retrieved from https://www.blender.org/about/

Burton, T., Jenkins, N., Sharpe, D., & Bossanyi, E. (2011). Wind Energy Handbook.

Caruelle, B., & Ducros, F. (2003). *Detahced Eddy Simulations of Attached and Detached Boundary Layers.*

Cengel, Y. A., & Cimbala, J. M. (2014). Fluid Mechanics, Third Edition.

CFD-Online. (2006). Retrieved from https://www.cfd-online.com/Wiki/Generic_scalar_transport_equation

CFD-online. (2008). Retrieved from https://www.cfd-online.com/Wiki/Detached_eddy_simulation_(DES)

CFD-Online. (2015). Retrieved from https://www.cfd-online.com/Wiki/Spalart-Allmaras_model

Chochua, G. (2002). *Computations of gas annular damper seal flows [electronic resource].*

Connecticut history. (2018, August 29). Retrieved from https://connecticuthistory.org/halladays-revolutionary-windmill-today-in-history-august-29/

Foroutan, H. (2014). *Simulation, Analysis, and Mitigation of Vortex Rope Formation in the Draft Tube of Hydraulic Turbines.*

Giljarhus, K. E. (2019). *Turbulence and turbulence modelling.*

Griffith, D. T., & Ashwill, T. D. (2011). *The Sandia 100-Meter All-Glass Baseline Wind Turbine Blade: SNL100-00.*

Griffith, D. T., & Richards, P. W. (2014). *The SNL 100-03 Blade: Design Studies with Flatback Airfoils for the Sandia 100-meter Blade.*

Hjorteland, S. (2019). *CFD-simulations on a 90° DU96-W-180 airfoil.*

Jain, P. (2011). Wind Energy Engineering.

Lourenco, L. M., & Shih, C. (1993). *Characteristics of the plane turbulent near wake of a circular cylinder. A particle image velocimetry study.*

Ma, X., Karamanos, G. -S., & Karniadakis, G. E. (2000). *Dynamics and low-dimensionality of a turbulent near wake.*

Manwell, J. F., McGowan, J. G., & Rogers, A. L. (2009). Wind Energy Explained.

Marten, D., & Peukert, J. (2013). *QBlade Guidelines v0.6.*

Moukalled, F., Mangani, L., & Darwish, M. (2016). The Finite Volume Method in Computational Fluid Dynamics.

Nakhostin, S. M. (2019). *Investigation of transitional turbulence models to predict drag crisis for flows over spheres and cylinder.*

NASA. (2019). Retrieved from https://turbmodels.larc.nasa.gov/spalart.html

OpenFOAM. (2020). Hentet fra https://www.openfoam.com/

Patrol, C. A. (2006, June 4). *Wikipedia*. Hentet fra https://en.wikipedia.org/wiki/File:Airfoil.svg

Pointwise. (2020). Retrieved from https://www.pointwise.com/about/

PyFR. (2020). Hentet fra http://www.pyfr.org/index.php

Qblade. (2020). *q-blade.org*. Retrieved from http://www.q-blade.org/#welcome

Ramírez, L., Fraile, D., & Brindley, G. (2020). *Offshore Wind in Europe, Key trends and statistics 2019.* WindEurope.

Sadrehaghighi, I. (2020). *Turbulence Modeling - A Review.*

Salim, M. S., & Cheah, S. (2009). *Wall y+ Strategy for Dealing with Wall-bounded Turbulent Flows.*

Sarrate, J. (2018). A brief introduction to mesh generation.

Sheibania, M., & Akbari, A. A. (2015, November 15). Finite element modeling of a wind turbine blade.

Sørensen, N. N., & Skrzypiński, W. (2014). *Vortex-induced vibrations of a DU96-W-180 airfoil at 90° angle off attack.*

The International Electrotechnical Comission. (2005). *IEC 61400-1 Wind Turbines-Part 1: Design Requirements.*

Third Planet Windpower. (2020). Retrieved from http://www.thirdplanetwind.com/energy/history.aspx

Timmer, W. A., & van Rooij, R. P. (2001). *Some aspects of high angle-of-attack flow on airfoils for wind turbine application.*

TU Wien. (2019). *cfd.at.* Retrieved from https://www.cfd.at/tutorials?fbclid=IwAR2xu7WDvOfaPX1Zd2Fn1VnhTzxtDBlT5dCta7lACudhn-6CpCqtToUPEMQ

Tu, J., Ahmadi, G., & Inthavong, K. (2013). *Numerical Methods and Its Solution.*

Twidell, J., & Gaudiosi, G. (2009). Offshore Wind Power.

VirtualBox. (2020). Retrieved from https://www.virtualbox.org/

Weston, D. (2019, February 11). *Windpower Monthly*. Retrieved from https://www.windpowermonthly.com/article/1525362/europes-offshore-wind-costs-falling-steeply

WindEurope. (2020). Retrieved from https://windeurope.org/about-us/

Windpower Engineering & Development. (2010). *windpowerengineering.com*. Retrieved from https://www.windpowerengineering.com/calculate-wind-power-output/

Witherden, F., & Vincent, P. (2016). *PyFR: High-Order Accurate Cross-Platform Petascale Computational Fluid Dynamics with Python.*

Witherden, F., Farrington, A., & Vincent, P. (2013). *PyFR: An open source framework for solving advection-diffusion type problems on streaming architectures using the flux reconstruction approach.*

Xu, C., Chen, L., & Lu, X. (2007). *Large-Eddy and Detached-Eddy Simulations of the separated flow around a circular cylinder.*

Xu, Y. (2016). *Wind Shielding Analysis for Cold Regions Using Experimental and Numerical Techniques.*

Appendix

OpenFOAM is a well explored CFD-tool by researchers and there is a lot of information online on how to use it, as well as a great selection of different tutorials. Based on this, it is considered unnecessary to include the case setups from OpenFOAM. PyFR on the other hand is not explored to the same degree, and there are fewer tutorials and information online. Therefore, the case setup for both the cylinder and airfoil case are included in order to help further researchers exploring higher order solutions in CFD. Please note that command lines starting with a ; (semicolon) are treated as a comment and does not affect the simulation.

PyFR Cylinder:

```
[backend]
precision = double
rank-allocator = linear

[backend-openmp]
cc = gcc
cblas = /usr/lib/x86_64-linux-gnu/blas/libblas.so.3
cblas-type = parallel
;cblas = Enter path to local BLAS library for OpenMP backend

[constants]
gamma = 1.4
mu = 0.00001802
nu = 0.00001470
Pr = 0.71
M = 0.1
vc = 0.0585
rhoc = 1.225

[solver]
system = navier-stokes
order = 3
;anti-alias = flux, surf-flux

[solver-time-integrator]
formulation = std
scheme = rk45
controller = pi
tstart = 0.0
tend = 100.0
dt = 0.01
atol = 1e-6
rtol = 1e-6
; errest-norm = l2
; safety-fact = 0.8
; min-fact = 0.3
; max-fact = 2.5

[solver-interfaces]
riemann-solver = rusanov
;riemann-solver = roem
ldg-beta = 0.5
ldg-tau = 0.1

[solver-interfaces-quad]
flux-pts = gauss-legendre
;quad-deg = 11
quad-deg = 9
quad-pts = gauss-legendre

[solver-interfaces-line]
flux-pts = gauss-legendre
;quad-deg = 11
quad-deg = 9

[solver-elements-tri]
soln-pts = williams-shunn
quad-deg = 9

[solver-elements-quad]
soln-pts = gauss-legendre
;quad-deg = 11
quad-deg = 9
quad-pts = gauss-legendre

[solver-elements-hex]
soln-pts = gauss-legendre
;quad-deg = 11
quad-deg = 9
quad-pts = gauss-legendre

[soln-plugin-writer]
dt-out = 0.2
basedir = .
basename = Cylinder2D-{t:.4f}

[soln-plugin-fluidforce-cylinder]
nsteps = 100
file = cylinder-forces.csv
header = true


[soln-plugin-nancheck]
nsteps = 100

[soln-bcs-front]
type = slp-adia-wall
```

```
[soln-bcs-back]
type = slp-adia-wall

[soln-bcs-cylinder]
type = no-slp-adia-wall

[soln-bcs-farfield]
type = char-riem-inv
rho = rhoc
u = 0
v = vc
w = 0
p = rhoc*vc*vc/(M*M*gamma)


[soln-ics]
rho = rhoc
u = 0
v = vc
w = 0
p = rhoc*vc*vc/(M*M*gamma)
```

PyFR Airfoil:

```
[backend]
precision = double
rank-allocator = linear

[backend-openmp]
cc = gcc
cblas = /usr/lib/x86_64-linux-gnu/blas/libblas.so.3
cblas-type = parallel
;cblas = Enter path to local BLAS library for OpenMP backend

[constants]
gamma = 1.4
mu = 0.00001802
nu = 0.00001470
Pr = 0.71
M = 0.1
vc = 10.3
rhoc = 1.225

[solver]
```

```
system = navier-stokes
order = 3
;anti-alias = flux, surf-flux

[solver-time-integrator]
formulation = std
scheme = rk45
controller = pi
tstart = 0.0
tend = 60.0
dt = 0.01
atol = 1e-6
rtol = 1e-6
; errest-norm = l2
; safety-fact = 0.8
; min-fact = 0.3
; max-fact = 2.5

[solver-interfaces]
riemann-solver = rusanov
;riemann-solver = roem
ldg-beta = 0.5
ldg-tau = 0.1

[solver-interfaces-quad]
flux-pts = gauss-legendre
;quad-deg = 11
quad-deg = 9
quad-pts = gauss-legendre

[solver-interfaces-line]
flux-pts = gauss-legendre
;quad-deg = 11
quad-deg = 9

[solver-elements-tri]
soln-pts = williams-shunn
quad-deg = 9

[solver-elements-quad]
soln-pts = gauss-legendre
;quad-deg = 11
quad-deg = 9
quad-pts = gauss-legendre

[solver-elements-hex]
soln-pts = gauss-legendre
;quad-deg = 11
quad-deg = 9
quad-pts = gauss-legendre
```

```
[soln-plugin-writer]
dt-out = 0.01
basedir = .
basename = Airfoil-DU96-w-180-{t:.4f}

[soln-plugin-fluidforce-airfoil]
nsteps = 100
file = airfoil-forces.csv
header = true


[soln-plugin-nancheck]
nsteps = 100

[soln-bcs-front]
type = slp-adia-wall

[soln-bcs-back]
type = slp-adia-wall

[soln-bcs-airfoil]
type = no-slp-adia-wall

[soln-bcs-farfield]
type = char-riem-inv
rho = rhoc
u = 0
v = vc
w = 0
p = rhoc*vc*vc/(M*M*gamma)


[soln-ics]
rho = rhoc
u = 0
v = vc
w = 0
p = rhoc*vc*vc/(M*M*gamma)
```