Master Degree in Mechanical Engineering

High Fidelity CFD Simulation of Horizontal Axis Wind Turbines

By

Niccolò Nudda

******

Supervisor(s):

Prof. P.Asinari, Politecnico di Torino, Supervisor

Prof. Bletzinger, T.U.M., Supervisor

Prof. Wuechner, T.U.M., Co-Supervisor

M.Sc. S.Sharhok., T.U.M., Supervisor

Examination Committee:

Prof. A. Fasana,

Prof. M. Baratta,

Prof. D. Maffiodo,

Prof. R. Sesana,

Prof. A. Somà

Politecnico di Torino

2019

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Niccolò Nudda
2019

*

Abstract

Offshore wind turbines are one of the most promising renewable technologies. Even if it is estimated that they cover only 0.22% of the world energy production, they are the renewable technology that is growing more rapidly. Modern turbine blades are up to 180 m and their upscale makes the design procedure more and more complex. Thus, numerical simulations play a key role in the design process, in particular computational fluid dynamics and computational structural mechanics ones. In wind energy CFD and CSM are strongly coupled since aerodynamic loads cause considerable blade deformations that can significantly change the performance. As a consequence fluid-structure interaction analysis are fundamental for the design of modern offshore rotors.
Standard industrial methods, such as blade element momentum, are no more sufficient when dealing with large rotors, since heavy flow detachments and highly skewed flows are present. In regard to the aerodynamic load prediction, computational fluid dynamic simulations are preferred and their execution represents the main motivation of this thesis. The aim of the work is to carry out a methodology to perform high fidelity computational fluid dynamics simulations on the DTU 10MW reference wind turbine[11]. The methodology is needed by the *Lehrstuhl fuer Statik* (Chair of Structural Analysis) research group, from the Technical University of Munich, for executing fluid-structure interaction simulations by means of the coupling frame CFD-CSM called Enhanced Multi-Physics Interface Research Engine (EMPIRE) [48]. The resulting computational model is capable of obtaining accurate fluid dynamic results that are very similar to the ones found in literature [11][47][25]. Furthermore the computational costs with respect to the literature simulations are reduced, since we have much less number of elements due to the adoption of hybrid grids and wall functions. The thesis provides every instruction for generating the grid and analyses what are the proper boundary conditions, the optimum numerical solver algorithms,

the optimum numerical discretization schemes and the proper turbulence model.

Several studies were initially done on the NACA 0012 airfoil profile with the aim of familiarizing with the software *OpenFoam* and *Pointwise* and of shallowly forecasting what the behaviour of the turbine simulations would be by using similar settings.

The wind turbine grids were generated with hybrid technique employing structured mesh in the region closer to the blade and unstructured mesh in every other region of the domain. A method to avoid high non-orthogonality and high skewness is proposed. Four grids with different first extruded layer lengths were obtained.

The selection of the turbulence model was studied by comparing $k\omega SST$ and Spalart-Allmaras models performances. Both steady state and transient simulation were performed on the four grids and results were compared with the literature ones. $k\omega SST$ model gives better prediction of power and thrust and more accurate flow patterns. Spalart-Allmaras model gives results similar to $k\omega SST$ ones, is faster in convergence and easier to set-up.

The use of wall functions have been studied employing the four grids. The difference between direct solution and wall function solution could be studied since one grid is completely in the viscous layer, two are in the buffer zone and one is fully in the inertial layer. Spalart-Allmaras model is almost insensitive to wall functions, thus coarse grids can be correctly employed without any loss in accuracy. $k\omega SST$ model is more sensitive when we deal with higher wind speeds, but the differences are in the order of 1%.

The thesis was mainly carried out at the Statik Department, in the Technische Universitaet Muenchen under the supervision of Prof. Bletzinger and Prof. Asinari thanks to the ITALDESIGN-GIUGIARO project.

Contents

List of Figures

List of Tables

Chapter 1

Introduction to Off-Shore Wind Turbines and to DTU 10MW

1.1 Off-Shore Wind Turbines

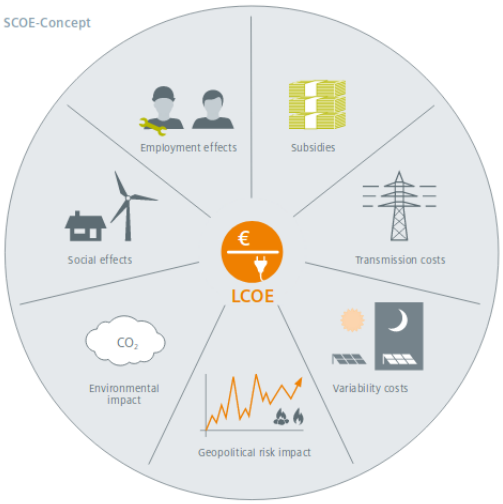
1.1.1 General Overview

Climate Agreement of Paris (COP21) subscribed by parties to the United Nations Framework Convention on Climate Change (UNFCCC) pointed out the importance to recognize that "*climate change represents an urgent and potentially irreversible threat to human societies and the planet and thus requires the widest possible cooperation by all countries, and their participation in an effective and appropriate international response, with a view to accelerating the reduction of global greenhouse gas emissions*" [55]. Being greenhouse gases mainly emitted by electric energy production [40], governments are incentivizing the use and research on renewable energies.

There are several ways to analyse the cost-effectiveness of an energy source, one of the most conventional is the *Levelized Cost of Energy (LCOE)*, a ratio between the total lifetime costs and the total electricity produced over the lifetime. Since the latter gives only a partial vision of the economy around an energy source SIEMENS invented a better parameter that takes into account the jobs created by the energy source, subsidies, transmission cost, variability

costs and geopolitical risk called "Society's Cost of Energy" (SCOE). This one would be a more appropriate parameter for allowing governments and investors to understand the benefits of every energy source.

(a) Levelized Cost of Energy[8]



(b) Society's Cost of Energy[Sie]

In order to underline the difference between the two indexes, SIEMENS makes a comparison between LCOE in 2013 and SCOE in 2025 from the most used energy sources.

Electricity Source	**LCOE (AC/MWh)**	**SCOE (AC/MWh)**
Nuclear	79	107
Coal	63	110
Gas	60	89
Photovoltaic	145	78
Onshore Wind	81	60
Offshore Wind	140	61

Table 1.1 SIEMENS Study for Various Energy Sources

It is evident that the most cost-effective energy resources are on-shore and off-shore wind turbines, even if off-shore cost is almost twice on-shore turbines cost. Indeed, wind turbines are one of the most promising renewable technologies and their use is in constant expansion since 1993 [36]. It is estimated

that the 5.6% of the global energy production has to be attributed to wind plants, increased nearly 11% with respect to 2017. While China and United States are the countries with the highest total wind power capacity, Denmark, Ireland, Sweden, Germany and Portugal are respectively the countries with the highest wind power capacity per capita. Germany is the 2018 top installer, its capacity has increased 33% relative to 2016 and accounts for 19% of its total net electricity in 2017. Although the majority of wind power is gathered with onshore plants (96%) offshore plants are the ones that are increasing more rapidly. In just one year (2017) total world offshore capacity has increased 30%. UK has the highest capacity with 6.8GW, followed by Germany (5.4 GW), China (2.8 GW), Denmark (1.3 GW) and the Netherlands (1.1 GW) [44].

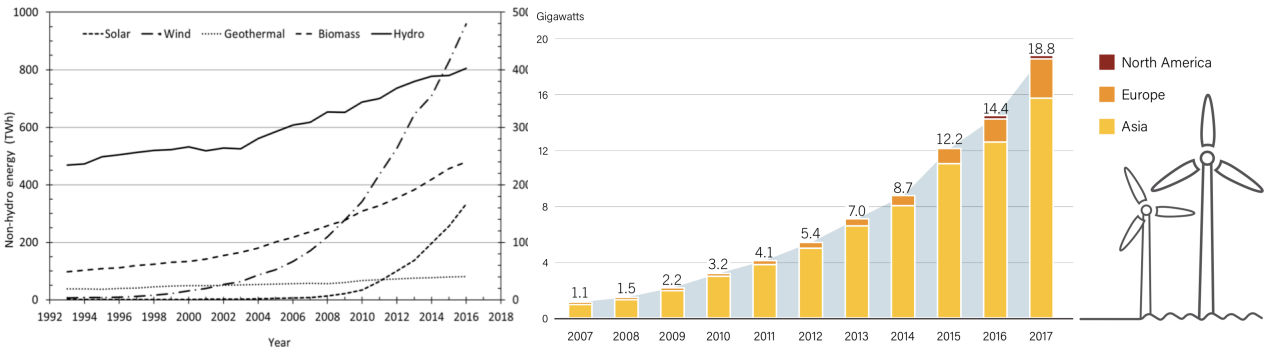


Fig. 1.2 (a) Renewable Energy Produced 1992-2018[36]; (b) Wind Power Offshore Global Capacity by Region,2007-2017[44]

Off-shore wind turbines have the advantage of a better wind source, due to the fact that off-shore wind is less turbulent, has as an average higher speed and is steadier, and of huge installation space although they are harder to install and to maintain[9].

Designers and researchers are now focusing in increasing turbine size in order to increase capacity factors, reduce installation costs and amortise total development cost for the same physical area. However the mass of the turbine increases with the cube of the rotor radius with linear upscaling making the design procedure more and more challenging [11]. The largest functional offshore wind turbines are three the Siemens Gamesa SG 8.0–167 DD, MHI Vestas V164–8.0

MW and Areva 3-bladed 8 MW. Industry and research are making huge efforts in order to upscale wind turbines up to 10-12MW.

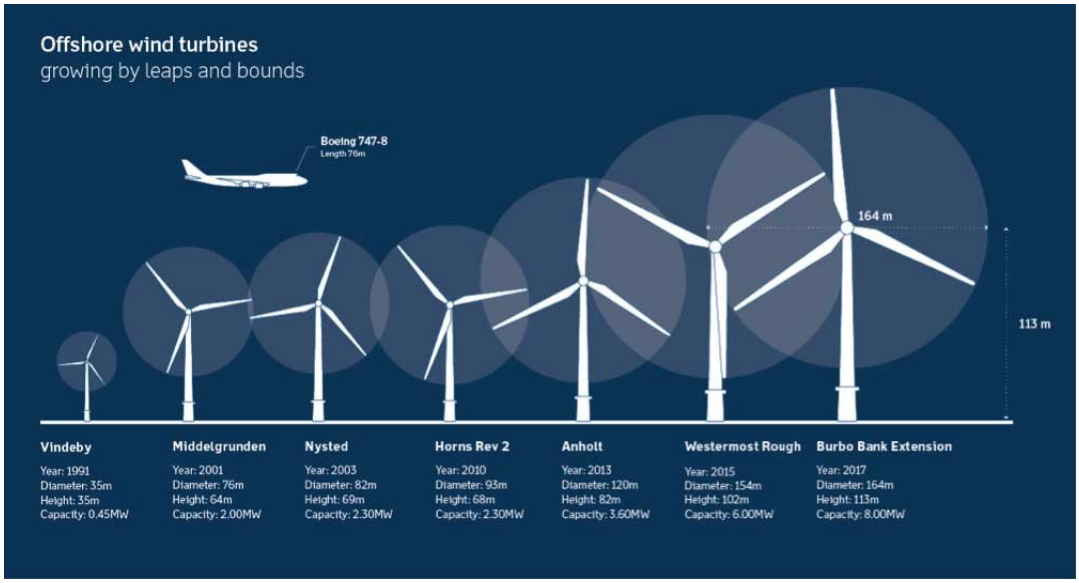


Fig. 1.3 Offshore Wind Turbine Size 1991-2017 [Ors]

At the moment the majority of installed plants are in water depths of < 40 m and < 50 Km from shore. Analysis demonstrated that deep offshore would open the markets potential in the Atlantic, Mediterranean and deep North Sea waters drastically decreasing LCOE.

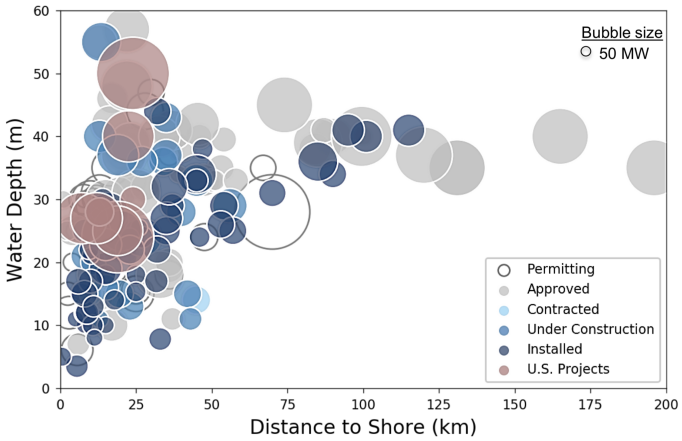


Fig. 1.4 Global Offshore Wind Projects as a Function of Water Depth and Distance to Shore[39]

Deep offshore design main challenges are foundations. Indeed turbine foundations have to bear both wind and hydrodynamic loads including axial force from the turbine support structure and cyclic loads from extreme sea states that vary in direction, amplitude and frequency.

1.1.2 Airfoil Aerodynamics Concepts

Wind turbines transform wind kinetic energy into rotor mechanical torque, that is, in turn, transformed into electrical energy by means of an alternator. Wind turbines can be divided into two macro-groups: horizontal axis wind turbines (HAWT) and vertical axis wind turbines (VAWT). The first ones are always preferred for large systems because they have higher power coefficients and a mechanical behaviour that is more stable with respect to VAWT [15].

The power that can be harvested by a wind turbine varies with the cube of the velocity $P = \rho A V^3$, where ρ is the air density, A is the swept area and V is the air velocity. According to the Betz limit only 59% of the aforementioned power can be theoretically harnessed. Modern turbines usually reach up to 0.5. In order to fully understand how a wind turbine works, it is necessary to understand the basic principles behind airfoil.

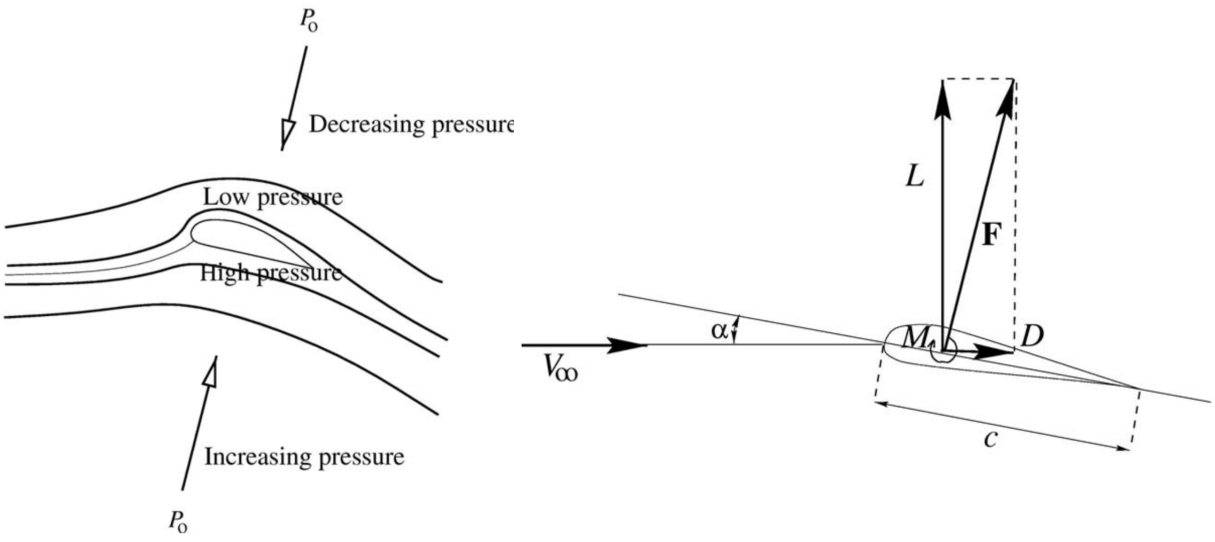


Fig. 1.5 (a) Definition of lift and drag[22]; (b)Explanation of the generation of lift[22]

Lift and Drag Coefficients

The generation of lift is the consequence of airfoil shape. From fluid dynamics we know that $\frac{\partial p}{\partial r} = \frac{\rho V^2}{r}$, a pressure gradient is needed to deform the streamlines, where r is the streamline curvature and V the velocity. Considering the pressure

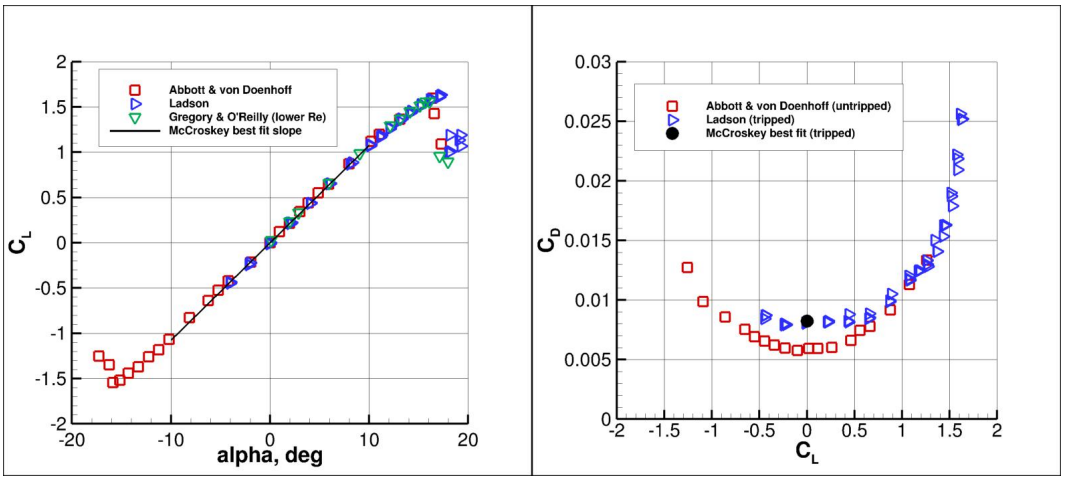
far from the airfoil as p_{inf} there must be a lower pressure on the upper side of the airfoil and a higher pressure on the lower side of the airfoil. The pressure gradient generates a force that can can be projected, as in the picture, in two components: lift(L) and drag(D). In order to have a dimensionless description of the airfoil characteristics the coefficients C_l and C_d are used.

The lift coefficient linearly depends on the angle of attack until it becomes non-linear and reaches the maximum value. The non-linear behaviour is due to the detachment of the fluid from the airfoil, called stall phenomenon. The drag coefficient has almost a parabolic dependence on the angle of attack, however the $C_d - C_l$ correlation is commonly preferred for defining the airfoil profile. The two coefficients are mathematically defined as:

$$C_l = \frac{L}{\frac{1}{2}\rho v_\infty^2 c} \tag{1.1}$$

$$C_d = \frac{D}{\frac{1}{2}\rho v_\infty^2 c} \tag{1.2}$$

where v_∞ is the far-field wind speed, ρ is the air density and c is the chord length. Below two curves obtained from experimental results are showed as example.



(a) $C_l - \alpha$ Curve - NACA 0012 [NLR] (b) $C_d - C_l$ Curve - NACA 0012 [NLR]

Pressure Coefficient

The pressure distribution along the airfoil is very important for load calculation and it is computed in terms of a dimensionless coefficient called pressure

coefficient, c_p. The latter is mathematically defined as:

$$c_p = \frac{p - p_\infty}{\frac{1}{2}\rho v_\infty^2} \tag{1.3}$$

If we are dealing with incompressible flows we can express the coefficient also as:

$$c_p = 1 - \left(\frac{v}{v_\infty}\right)^2 \tag{1.4}$$

This form shows better that at the stagnation point, where $v = 0$, the pressure coefficient is 1. Usually the pressure coefficient is depicted in a graph as function of chord percentage x/c as can be seen from the experimental results shown below.

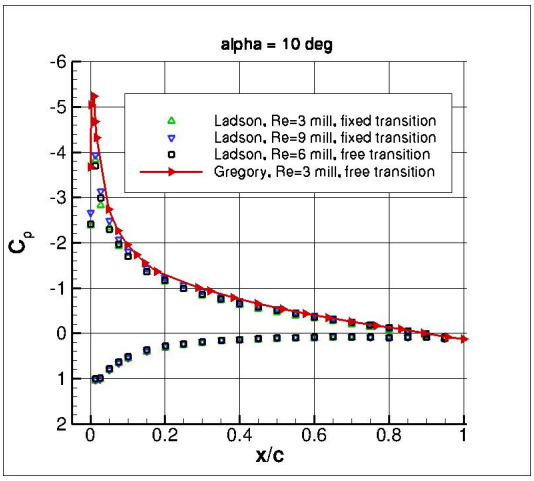


Fig. 1.7 $c_p - x/c$ Curve - NACA 0012 [NLR]

Airfoil Profiles

During 1980s and early 1990s NACA (National Advisory Committee for Aeronautics) airfoil have been extensively used for wind turbine applications. However computations and experiments showed efficiency lacks due to premature transitions (Summary of the Delft University Wind Turbine Dedicated Airfoil). Since a huge amount of experimental data have been acquired, NACA airfoils are still extensively used for numerical models validations and they are often used as baseline for optimizing aerodynamics performances (Computations of Active Flow Control Via Steady Blowing Over a NACA-0018 Airfoil: Implicit LES and RANS Validated Against Experimental Data). In 1932 NACA established standard codes for describing airfoil geometry. The 4 digit series was

the first and is the simplest standardization. Every digit is expressed as chord percentage. The first digit is the maximum camber, the second digit is the position in which the latter is located. The third and fourth digit describe the maximum thickness of the airfoil. We can clarify the code with an example considering the NACA 2215. In this case the maximum camber is $0.02 \cdot c$ located at $0.02 \cdot c$ with a maximum thickness of $0.15 \cdot c$, where c is the chord length. The image below shows some of the NACA 4-digit shapes [NLR].

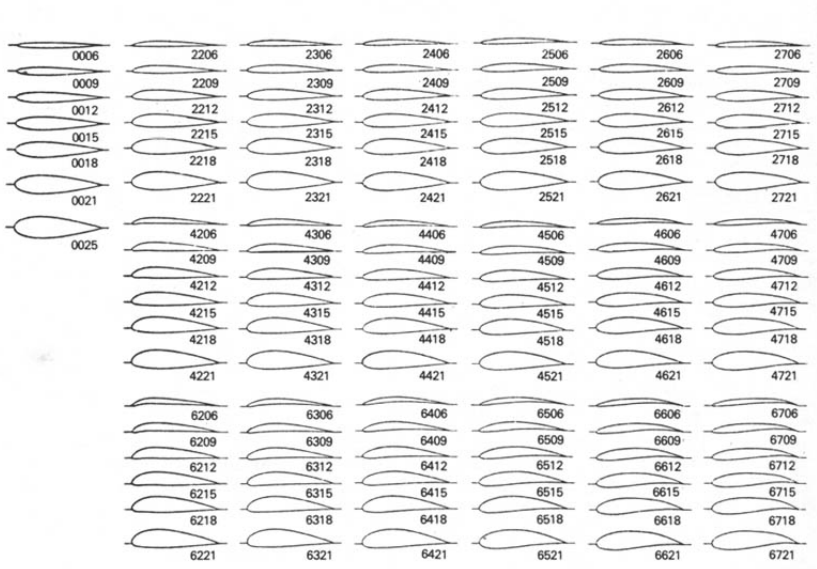


Fig. 1.8 NACA 4-digit Shapes [NLR]

In this thesis the NACA 0012 will be studied. Since the first two digits are zero, we will deal with a symmetrical airfoil shape.
In three dimensional wings the lift is reduced due to down-wash if compared to the one computed in a two dimensional profile. If we consider the rotation, in the separated flow, the momentum is negligible with respect to centrifugal forces which directs the flow span-wise. The Coriolis forces moves the flow to the trailing edge, increasing the lift force.

1.2 DTU 10MW Off-Shore Wind Turbine

In this thesis the aerodynamic characteristics of a DTU 10MW reference turbine will be studied. The DTU 10MW is an horizontal axis, variable pitch and variable speed wind turbine developed mainly by Technical University of Denmark and by Vestas. Its design is based on the enlargement of the NREL 5MW with the relative modifications that must be done for assuring stiffness and with the aim of maximizing power production. Starting from the NREL 5MW the following constraint were imposed in the design:

- rotor radius = 89.166m;
- minimum relative thickness = 24.1%, since high stiffness is needed for large rotors;
- twist curvature should not change sign, for simplifying the structure;
- maximum thrust at 11 m/s equal to 1500kN;
- maximum chord = 6m.

It is worth to consider that the largest off-shore wind turbine available now is the AD 8-180 produced by Adwen, that recently installed a prototype in Bremenhaven (Germany), and has rotor diameter of 180 m with a nominal power of 8MW.
The main data of the DTU 10MW are resumed in this table:

Parameter	**Data**
Rating	10 MW
Rotor orientation, configuration	Upwind, three blades
Control	Variable speed, collective pitch
Drivetrain	Medium speed, multiple stage gearbox
Rotor, hub diameter	178.3 m, 5.6 m
Cut-in, rated, cut-out wind speed	4 m/s, 11.4 m/s, 25 m/s
Cut-in, rated rotor speed	6 RPM, 9.6 RPM
Rated tip speed	90 m/s
Overhang, shaft tilt, pre-cone	7.07 m, 5 ı , 2.5 ı
Pre-bend	3m
Rotor mass	229 tons (each blade 41 tons)
Nacelle mass	446 tons
Tower mass	605 tons

Table 1.2 DTU 10MW Data

As far as the structural design is concerned, the blades are made of glass fibre reinforced with balsa wood, which is used as a sandwich core material. The airfoil are of the FFA-W3-xxx family, the root has a cylindrical shape while in the tip (r/R>99) the airfoil NACA0015 is employed. Two more airfoil were appositely created by DTU: one has a relative thickness of 48% and was obtained by scaling a 36% airfoil, the other has a relative thickness of 60% and was obtained interpolating the one with 48% and the cylinder part.

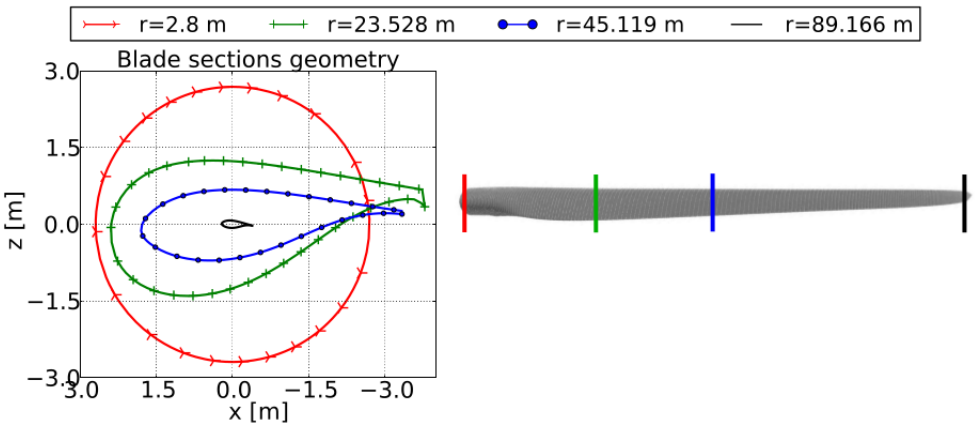
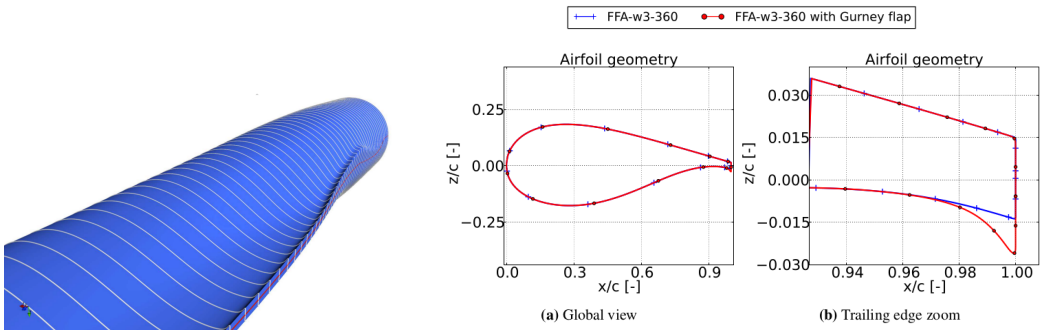


Fig. 1.9 Airfoil Profiles at different Blade Heights [25]

In order to improve the aerodynamic performance, wedge shaped Gurney flaps were added in the spanwise lenghts 36%, 48% and 60% and showed good results in staedy-state aerodynamic analysis. Gurney flaps were developed for automotive applications with the aim of improving down-force. They are flat plates on the order of 1-3% c, perpendicular to the chord, added in the trailing edge. They are very commonly used in subsonically systems to economically increase the lift to drag ratio. The Gurney flap should create two contrarotating vortices at the end of the trailing edge that shift the re-attachment of the suction side flow close to the trailing edge. Thus we have an improvement in the lift coefficient since we are increasing the attached flow on the wing. However Horcas et al. [25], by means of a aeroelestic simulation, demonstated that the one designed in this turbine caused a reduction of the power coefficient of 1.4% and an increase in thrust of 0.8%. The ineffectiveness of this device is caused by the important blade deflections parallel to the rotor axis that are in the order of the 44% of the blade tip-tower distance(18.26 m). The two pictures below clearly show what we expected to have and what simulations predict on the turbine blade[25][27].



(a) Global view **(b)** Trailing edge zoom

Fig. 1.10 DTU 10MW Gurney Flaps [25]

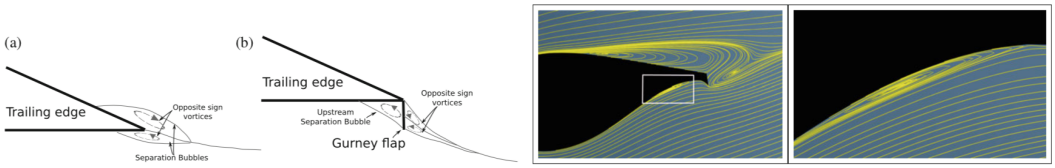


Fig. 1.11 Expected flow behaviour vs actual flow behaviour [25]

A pre-bend version of the blade was also developed using the deformation deriving from a wind speed of $5m/s$. The pre-bending is made to reduce the thrust load on the blade. It does not largely affects the rotor's natural frequencies but it reduces both thrust and mechanical power. In this thesis only the "straight" version of the blade will be used.

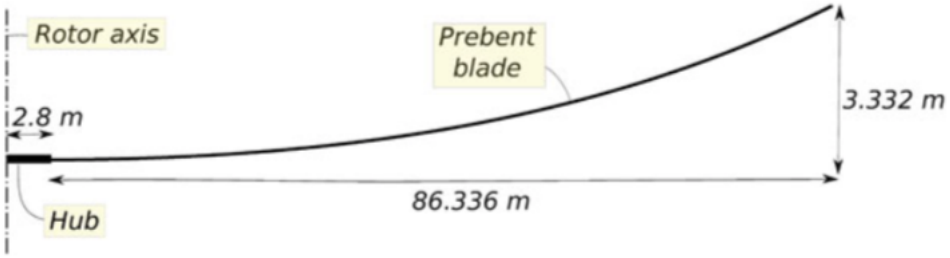


Fig. 1.12 Pre-bent blade [11]

1.3 Thesis Motivation and objectives

The larger dimensions make the numerical simulation an essential part of the design procedure since experiments can be done only on reduced scale models. Computational fluid dynamics and numerical structural analysis play a fundamental role in the design of these machines. The Blade Element Momentum (BEM) is extensively used in the design of onshore turbines but its use is limited in offshore rotors due the presence of highly skewed flows and significant detachments[25].
The design stages of a wind turbine can be resumed by this flow chart:

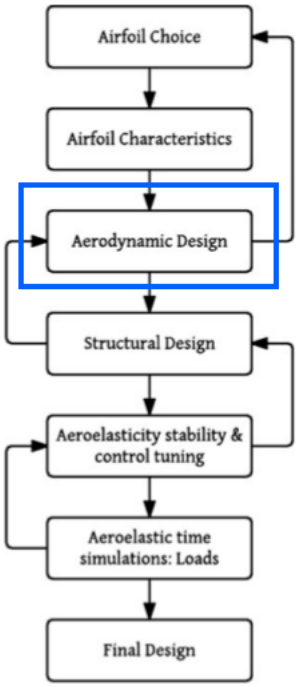


Fig. 1.13 Wind Turbine Design Algorithm [11]

The thesis is focused on the aerodynamic design stage and aims in finding out a methodology to realize high fidelity CFD simulations on the DTU 10MW reference wind turbine. The methodology is needed by the research group of the Statik Departement of the Technical University of Munich (TUM), department in which the thesis was done, in order to carry out fluid-structure interaction simulations by means of the coupling frame CFD-CSM called Enhanced Multi-Physics Interface Research Engine (EMPIRE) developed at Technical

University of Munich (TUM) [48]. The ways in which a mesh can be created with the software Pointwise will be analysed, studying the influence of the cell size on the results. Two turbulence models, Spalart-Allmaras and $k\omega SST$, will be used and the accuracy and sensitivity of each of them will be compared. Both steady state and transient simulations will be compared.
In the first part of the thesis several studies will be conducted on the NACA0012 airfoil profile, with the aim of familiarizing with the software Pointwise, OpenFoam and ParaView and with the purpose of understanding the settings that will be used in the turbine simulations. First a convergence study will be done using Nasa Langley Research grids, that provides a database of numerical and experimental results which will be used for validating our simulations. Then SIMPLE and SIMPLE-Consistent algorithm will be used, the latter, from the literature, seems to be faster and more robust. Four different turbulent models will be tested: Spalart-Allmaras, $k\omega SST$, $k\epsilon$ and $kk_l\omega$. From the literature, only Spalart-Allmaras and $k\omega SST$seems to be suitable for this kind of simulation in which high adverse pressure gradients are present. Also divergence discretization schemes will be compared since they play an important role for the convergence of the residuals and for the accuracy of the results. Since we expect to have a large number of elements in the turbine grid also the effects of wall functions with in-house generated grids will be studied. In the end the turbine results will be compared with the ones obtained by different research groups from the Denmark Technical University (DTU), University of Stuttgart (UoS) and with the Universitè de Mons (UMONS). The three groups made CFD simulations using different software and applying different strategies, we expect to obtain similar results with the methodology proposed in this thesis.

Chapter 2

Governing Equations

2.1 Navier-Stokes Equations

The mathematical model used for dynamically describe a fluid is based on the fundamental conservation laws of mass, linear momentum and energy. The continuity equation implies that, in absence of sources and sinks, the mass inside an infinitesimal volume is conserved.

$$\frac{\partial \rho}{\partial t} = -\frac{\partial \rho v_i}{\partial x_i} \tag{2.1}$$

The momentum equation is the equivalent of Newton's second dynamic principle applied to fluids. It was first formulated by Augustin-Louis Cauchy such as:

$$\frac{\partial \rho \boldsymbol{v}}{\partial t} + \frac{\partial \rho \boldsymbol{v} \cdot v_i}{\partial x_i} + \frac{\partial p}{\partial x_i} + \frac{\partial \tau_i}{\partial x_i} = \rho \boldsymbol{g} \tag{2.2}$$

where p is the static pressure, τ is viscous tensor and $\rho \boldsymbol{g}$ the gravity force. It is important to point out that the divergence term is not linear.
In order to obtain Navier-Stokes momentum equation we have to formulate the stress due to viscous forces as:

$$\tau = -\mu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) + \left(\frac{2}{3}\mu - \kappa \right) \left(\frac{\partial v_i}{\partial x_i} \right) I \tag{2.3}$$

where κ is the bulk viscosity and μ is the dynamic viscosity. Since we will deal with incompressible flows the last term is zero. Substituting this term in the

momentum equation we obtain a partial differential equation, the momentum Navier-Stokes equation:

$$\frac{\partial \rho \boldsymbol{v}}{\partial t} + \frac{\partial \rho \boldsymbol{v} v_i}{\partial x_i} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_i}\left[\mu\left(\frac{\partial v_i}{\partial x_i} + \frac{\partial v_j}{\partial x_i}\right)\right] + \rho \boldsymbol{g} \tag{2.4}$$

The two equations defined have analytical solutions just in only a few simple cases and they are really complicated to solve since: they are coupled via the velocity, the pressure appears in the momentum equation as source term and does not have a transport equation and, as aforementioned, they present a non-linearity term[37][49].

2.2 RANS Turbulence Models

When viscous forces are not sufficient to balance inertia forces we have a turbulent flow. The latter is characterized by a chaotic, time-dependent and highly diffusive behaviour. The first theory developed to describe this flow was made by Kolmogorov [30]. According to his theory the energy in turbulent flows is transmitted to the fluid in vortices structures that are comparable to the size of the moving body. The energy possessed by this vortices is then transmitted to smaller size vortices in a "energy cascade" manner. This process generates a chain in which the original vortex is broken-up into a smaller vortex, the smaller vortex is broken-up into another smaller vortex and so on, until the minimum size vortex is reached. After the smallest vortex the viscous forces are enough to balance the inertia forces. Due to the fact that the smallest vortex is really small, solving the entire domain in space and time would require the use of a very high number of cells. This causes a prohibitive computational cost that usually cannot be handled by industries. To tackle this problem *Reynolds Averaged Navier Stokes Equations* have been formulated. The latter can approximate the results and decrease the mesh size that, consequently, reduces the computational cost that becomes accessible. Every variable of each equation describing the flow is split into a time-averaged part and a fluctuating part:

$$\phi(x,t) = \overline{\phi}(x,t) + \phi'(x,t) \tag{2.5}$$

The term $\overline{\phi}$ is usually computed as:

$$\overline{\phi}(x) = \frac{1}{T}\int_{t}^{t+T} \phi(x,t)dt \tag{2.6}$$

In this way we introduced a new set of unknowns that are usually called *Reynold Stress Tensor* τ_R that can be solved using the Bousinnesq hypothesis.

$$\tau_R = -\rho\overline{v'v'} = -\rho \begin{bmatrix} \overline{u'u'} & \overline{u'v'} & \overline{u'w'} \\ \overline{u'v'} & \overline{v'v'} & \overline{v'w'} \\ \overline{u'w'} & \overline{v'w'} & \overline{w'w'} \end{bmatrix} \tag{2.7}$$

2.2.1 Boussinesq Hypothesis

The Boussinesq hypothesis assumes that the Reynold stress behaves like Newtonian fluids viscous stresses. Since we will deal with incompressible flows:

$$\tau_R = -\rho\overline{u'u'} = \mu_t \left(\frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_j}}{\partial x_i} \right) - \frac{2}{3}\rho k I \tag{2.8}$$

Now the problem turned into the computation of the turbulent viscosity that is differently calculated depending on the turbulence model. There is a large number of turbulence models based on this hypothesis, the four most known are the Spalart-Allmaras, the $k\epsilon$, the $k\omega$ and the $k\omega SST$. Between them only the Spalart-Allmaras and the $k\omega SST$ are suitable for adverse pressure gradients [13], the first is very robust for its simplicity and fast convergence while the second gives more accurate result, in particular in separations zones. They are also used in other CFD simulations of the DTU 10MW RWT found in literature [11] [25] [48], with which our results will be compared. Other papers [**?**] [50] show that the improved version of the model used in this thesis, such as $k\epsilon(RNG)$ and transtion γRe_θ, can reliably work for HAWT simulations. The use of the $kk_l\omega$ model has been proposed by the author's German supervisor, on the base of his experience.

2.2.2 Spalart-Allmaras Model

The Spalart-Allmaras model is a one-equation model in which the turbulent viscosity is computed by means of a transport equation. The latter computes the parameter $\tilde{\nu}$ that is related to the turbulent kinematic viscosity by the relation expressed in the equation (2.16). The model was created in 1992 and showed good results with airfoils simulations[53].

$$\begin{aligned}\frac{\partial \tilde{\nu}}{\partial t} + u_j \frac{\partial \tilde{\nu}}{\partial x_j} = c_{b1}(1 - f_{t2})\tilde{S}\tilde{\nu} - \left[c_{w1} f_w - \frac{c_{b1}}{\kappa^2} f_{t2}\right]\left(\frac{\tilde{\nu}}{d}\right)^2 + \\ + \frac{1}{\sigma}\left[\frac{\partial}{\partial x_j}\left((\nu + \tilde{\nu})\frac{\partial \tilde{\nu}}{\partial x_j}\right) + c_{b2}\frac{\partial \tilde{\nu}}{\partial x_i}\frac{\partial \tilde{\nu}}{\partial x_i}\right]\end{aligned} \tag{2.9}$$

2.2.3 $k\epsilon$ Model

The $k\epsilon$ model is a two equation model developed by Jones and Launder in 1972 [31]. It is one of the most common model used in computational fluid dynamics and has been validated many times.
The model aims to find the turbulent viscosity by means of two transport equations, one for the turbulent kinetic energy and one for the dissipation rate. The model assumes that the flow is fully turbulent and that the molecular viscosity is negligible, thus it is not suitable for shear flows but only for free shear flows.
The turbulent kinetic energy is defined as the kinetic energy produced by the fluctuating velocities, $\boldsymbol{v}' = \boldsymbol{v} - \overline{\boldsymbol{v}}$:

$$k = \frac{1}{2}\left((v_x')^2 + (v_y')^2 + (v_z')^2\right) \tag{2.10}$$

The turbulent kinetic energy transport equation is defined as:

$$\frac{\partial \rho k}{\partial t} + \frac{\partial \rho v_j k}{\partial x_j} - \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] = G_k + -\frac{2}{3}\rho\frac{\partial v_j}{\partial x_j} - \rho\epsilon + S_k \tag{2.11}$$

The term ϵ represent the turbulent energy dissipation per unit mass due to viscous stress and is defined as:

$$\epsilon = \frac{1}{2}\frac{\mu}{\rho}\overline{\left[\frac{\partial v_i'}{\partial x_j} + \frac{\partial v_j'}{\partial x_i}\right] : \left[\frac{\partial v_i'}{\partial x_j} + \frac{\partial v_j'}{\partial x_i}\right]} \tag{2.12}$$

Its transport equation is defined as:

$$\begin{aligned}\frac{\partial \rho\epsilon}{\partial t} + \frac{\partial \rho u_j \epsilon}{\partial x_j} - \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\epsilon}\right)\frac{\partial \epsilon}{\partial x_j}\right] = \frac{C_1 G_k \epsilon}{k} + \\ - \left(\frac{2}{3}C_1 + C_{3,RDT}\right)\rho\frac{\partial u_j}{\partial x_j}k - C_2\rho\frac{\epsilon^2}{k} + S_\epsilon\end{aligned} \tag{2.13}$$

Solving the two transport equations above we can calculate the turbulent viscosity as:

$$\mu_t = \rho C_\mu \frac{k^2}{\epsilon} \tag{2.14}$$

2.2.4 $k\omega$SST Model

The $k\omega$ model is a two equation model. It is similar to the $k\epsilon$ model but the ϵ term, the turbulent energy dissipation, has been turned into ω, the specific turbulence dissipation. The first complete model for this category was proposed by Kolmogorv [30] and was improved by Wilkox in 2006 [59].
The main advantage with respect to the $k\epsilon$ is that it can be integrated in the sub-layer without the need of damping functions. However the main drawback is that is that it is very sensitive to the free stream specified values.

Standard Wilkox $k\omega$ Model

The ω term is defined as:

$$\omega = \frac{\epsilon}{C_\mu k} \tag{2.15}$$

The turbulence kinetic energy equation is:

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho v_j k)}{\partial x_j} = P - \beta^*\rho\omega k + \frac{\partial}{\partial x_j}\left[\left(\mu + \sigma_k\frac{\rho k}{\omega}\right)\frac{\partial k}{\partial x_j}\right] \tag{2.16}$$

The turbulence specific dissipation rate transport equation is:

$$\begin{aligned}\frac{\partial(\rho\omega)}{\partial t}+\frac{\partial(\rho v_j\omega)}{\partial x_j}=\frac{\gamma}{k}P-\beta\rho\omega^2+\\ +\frac{\partial}{\partial x_j}\left[\left(\mu+\sigma_\omega\frac{\rho k}{\omega}\right)\frac{\partial\omega}{\partial x_j}\right]+\frac{\rho\sigma_{\omega d}}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial\omega}{\partial x_j}\end{aligned} \tag{2.17}$$

$k\omega$SST Model

The Shear Stress Transport (SST)formulation combines the advantages from the $k\epsilon$ and from the $k\omega$ models.
It was developed by Menter in 1993 [34] and is one of the most used turbulence models. It is identical to the standard $k\omega$ in the inner 50% boundary layer and it changes gradually to the standard $k\epsilon$ towards the boundary layer edge. For free shear layer is identical to the $k\epsilon$ model. In adverse pressure gradient boundary layers takes into account the principal turbulent shear stress that is computed with the Bradshaw's assumption, so it is proportional to the turbulent kinetic energy. So it has the advantages of not having any damping function in the sub layer and to behave well in the free stream.

The turbulence kinetic energy equation is now:

$$\frac{\partial(\rho k)}{\partial t}+\frac{\partial(\rho v_j k)}{\partial x_j}=P-\beta^*\rho\omega k+\frac{\partial}{\partial x_j}\left[(\mu+\sigma_k\mu_t)\frac{\partial k}{\partial x_j}\right] \tag{2.18}$$

The turbulence specific dissipation rate transport equation is modified as:

$$\begin{aligned}\frac{\partial(\rho\omega)}{\partial t}+\frac{\partial(\rho v_j\omega)}{\partial x_j}=\frac{\gamma}{\nu_t}P+\\ -\beta\rho\omega^2+\frac{\partial}{\partial x_j}\left[(\mu+\sigma_\omega\mu_t)\frac{\partial\omega}{\partial x_j}\right]+2(1-F_1)\frac{\rho\sigma_{\omega2}}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial\omega}{\partial x_j}\end{aligned} \tag{2.19}$$

2.2.5 $kk_l\omega$ Model

The $kk_l\omega$ model is a three equation model based on the $k\omega$ used for transitional flow simulations developed by K. Walters and D. Cokljat in 2008 [58]. It tries to mimic the physics of the phenomena by employing the third equation to predict

the magnitude of low-frequency velocity fluctuations in the pre-transitional boundary layer. The transport equations are illustrated here below:

Turbulent kinetic energy equation

$$\frac{\partial k_T}{\partial t}+\mathbf{v}\cdot\frac{\partial k_T}{\partial x_j}=P_{k_T}+R_{BP}+R_{NAT}-\omega k_T-D_T+\frac{\partial}{\partial x_j}\left[\nu+\left(\frac{\alpha_T}{\sigma_k}\right)\frac{\partial k_T}{\partial x_j}\right] \quad (2.20)$$

Laminar kinetic energy

$$\frac{\partial k_L}{\partial t}+v\cdot\frac{\partial k_L}{\partial x_j}=P_{K_L}-R_{BP}-R_{NAT}-D_L+\frac{\partial}{\partial x_j}\left[\nu\frac{\partial k_L}{\partial x_j}\right] \quad (2.21)$$

Scale-determining variable $\omega=\epsilon/k_T$ transport equation

$$\begin{aligned}\frac{\partial \omega}{\partial t}+v\cdot\frac{\partial \omega}{\partial x_j}=C_{\omega 1}\frac{\omega}{k_T}P_{k_T}+\left(\frac{C_{\omega R}}{f_W}-1\right)\frac{\omega}{k_T}(R_{BP}+R_{NAT})-C_{\omega 2}\omega^2\\ +C_{\omega 3}f_\omega\alpha_T f_W^2\frac{\sqrt{k_T}}{d^3}\left[\left(\nu+\frac{\alpha_T}{\sigma_\omega}\right)\frac{\partial \omega}{\partial x_j}\right]\end{aligned} \quad (2.22)$$

2.3 Wall Functions

Along solid boundaries a *no slip* boundary conditions is used:

$$v=v_{wall} \quad (2.23)$$

As a consequence near the boundaries there are always large gradients and, in addition, we assist to a reduction of turbulent kinetic energy that becomes comparable to the viscous stress. Due to these two issues we can adopt two solutions: reduce the elements size or use wall functions. The first way obviously increases the computational cost, the second one will be briefly described in this paragraph.

The wall boundary conditions are defined considering the y^+ value that is defined as:

$$y^+ = \frac{d_\perp u_\tau}{\nu} \tag{2.24}$$

where $d_\perp$ is the distance normal to the wall, ν is the kinematic viscosity, and u_τ is the friction velocity defined as:

$$u_\tau = \sqrt{\frac{|\tau_w|}{\rho}} \tag{2.25}$$

where τ_w is the wall shear stress.
From those two definitions we can define three sublayers:

- $0 < y^+ < 5$ viscous sublayer
- $5 < y^+ < 30$ buffer sublayer
- $30 < y^+ < 200$ inertial sublayer

Measurements and numerical results show that: in the viscous sublayer turbulent effects are negligible, in the buffer sublayer both viscous and turbulent effects are considered and in the inertial sublayer viscous effects are negligible. The transition from one sublayer to another happens around $y^+ = 11 - 12$. In order to understand if we are in the viscous or in the inertial sublayer we compute the y^+ value in the centroid nearest to the wall as:

$$y_C^+ = \frac{C_\mu^{1/4} {k_C}^{1/2}}{\nu} (d_\perp)_C \tag{2.26}$$

Viscous Sublayer

The flow is assumed to be laminar, thus the viscosity is the laminar one and the shear stress is computed following the laminar model. The main parameters in the viscous sublayer are computed as:

$$u^+ = \frac{|v - v_w|_{||}}{u_\tau} \tag{2.27}$$

$$k^+ = \frac{k}{u_\tau^2} \tag{2.28}$$

$$\epsilon^+ = \frac{\epsilon \nu}{u_\tau^4} \tag{2.29}$$

$$\omega^+ = \frac{\omega\nu}{u_\tau^2} \tag{2.30}$$

where $|v - v_w|$ is the relative velocity parallel to the wall.
For the first point near the wall the turbulent kinetic energy is set to zero while the production of turbulent kinetic energy is computed as:

$$P_k = \mu \frac{(|v_C - v_w|_{||})^2}{(d_\perp)_C^2} \tag{2.31}$$

Still in the first point for $k-\epsilon$ and $k-\omega$ model we can compute the quantities:

$$\epsilon_C = \frac{C_\mu \rho k_C^2}{\mu} \tag{2.32}$$

$$\omega_C = \frac{6\nu}{C_{\beta 1}(d_\perp)_C^2} \tag{2.33}$$

Inertial Sublayer

The inertial sublayer needs the assumption for the momentum profile: the flow is modelled as one dimensional Couette flow with zero pressure gradient. The quantities k, ϵ and ω are defined as:

$$u^+ = \frac{1}{\kappa} ln(d^+) + B \tag{2.34}$$

$$k^+ = \frac{1}{\sqrt{C_\mu}} = \frac{1}{\sqrt{\beta^*}} \tag{2.35}$$

$$\epsilon = \frac{\nu}{u_\tau \kappa d_\perp} \tag{2.36}$$

$$\omega^+ = \frac{\nu}{u_\perp \kappa d_\perp \sqrt{\beta^*}} \tag{2.37}$$

In this case the shear stress is computed using logarithmic wall functions as:

$$|\tau_w| = \tau_{lam} \frac{d^+}{u^+} \tag{2.38}$$

The production of turbulent kinetic energy is computed assuming that $\nabla k_C = 0$ and $\tau = \tau_C$, thus becoming:

$$P_k = |\tau_w| \frac{u_\tau}{\kappa (d_\perp)_C} \tag{2.39}$$

For which regards the $k\epsilon$ models, in every volume connected to the wall, the ϵ equation is not solved. The latter is imposed equal to the production of turbulent kinetic energy.

$$\rho \epsilon_C = P_k \Rightarrow \epsilon_C = \frac{C_\mu^{3/4} k_C^{3/2}}{\kappa (d_\perp)_C} \tag{2.40}$$

The same procedure is applied for $k\omega$, the ω equation is not solved while the dissipation of turbulence kinetic energy is imposed to be equal to its production rate.

$$\rho \epsilon_C = P_k \Rightarrow \omega_C = \frac{k_C^{1/2}}{\kappa C_\mu^{1/4} (d_\perp)_C} \tag{2.41}$$

Launder-Spalding Wall Function

The Launder-Spalding boundary condition is an improvement to the previously seen standard boundary condition. It is valid also in the case of non-equilibrium local conditions and improves the fact that, near the wall, the properties inside the volumes vary drastically so it is not so accurate to evaluate them at the cell centre. Using $\sqrt{k}$ as characteristic turbulent scale velocity the new production and dissipation terms are found:

$$\overline{P_k} = \frac{\rho C_\mu^{1/4} \sqrt{k_C}}{(d_\perp)_C (\frac{1}{\kappa} ln(d_C^*) + B)} [|v_C - v_w|_{||}]^2 \tag{2.42}$$

$$\overline{\epsilon_C} = \frac{C_\mu^{3/4} k_C^{3/2}}{(d_\perp)_C} \left(\frac{1}{\kappa} ln(d_C^*) + B \right) \tag{2.43}$$

nutUSpaldingWallFunction - OpenFoam

It is the wall function used for the kinematic turbulent viscosity that is already implemented in OpenFoam[ope]. It is based on the velocity and it uses Spalding law to give a continuous nut profile to the wall. The mathematical formulation is:

$$y^+ = u^+ + \frac{1}{E}\left[exp(\kappa u^+) - 1 - \kappa u^+ - 0.5(\kappa u^+)^2 - \frac{1}{6}(\kappa u^+)^3\right] \tag{2.44}$$

where E is the roughness parameter.

2.4 Multiple Reference Frames

The Multi Reference Frames method is used when we deal with moving and stationary objects in the same domain and we want to solve the problem as a steady-state problem[19].
The MRF divides the domain into sub-regions each possessing a certain relative velocity with respect to the inertial reference frame. In our case we will divide our domain in the stationary region and in the MRF region that comprehends the blade and a region of fine cells.
For which regard the stationary region the Navier-Stokes equations are expressed in terms of absolute velocity and the relative velocity is zero so they are in the same form seen in chapter 2. The equations for a rotating reference frame for an incompressible flow are now expressed. Indicating the inertial velocity as:

$$\boldsymbol{v}^I = \boldsymbol{v}^R + \boldsymbol{\Omega} \times \boldsymbol{r} \tag{2.45}$$

where $\boldsymbol{v}^R$ is the relative velocity, $\boldsymbol{\Omega}$ is the rotational velocity and $\boldsymbol{r}$ is the displacement vector. In the stationary reference frame the Navier-Stokes equation becomes:

$$\frac{\partial \rho v_i^I}{\partial x_i} = 0 \tag{2.46}$$

$$\frac{\partial \rho \boldsymbol{v}^I}{\partial t} + \frac{\partial \rho \boldsymbol{v}^I v_i^I}{\partial x_i} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_i}\left[\mu\left(\frac{\partial v_i^I}{\partial x_j} + \frac{\partial v_j^I}{\partial x_i}\right)\right] + \rho \boldsymbol{g} \tag{2.47}$$

since $\boldsymbol{v}^R = 0$ and $\boldsymbol{\Omega} = 0$. If we want to express the equations in the MRF zone we have to derive equation 9.1 with respect to time:

$$\frac{d\boldsymbol{v}^I}{dt} = \frac{d\boldsymbol{v}^R}{dt} + \frac{d\boldsymbol{\Omega}}{dt} \times \boldsymbol{r} + 2\boldsymbol{\Omega} \times \boldsymbol{v}^R + \boldsymbol{\Omega} \times \boldsymbol{\Omega} \times \boldsymbol{r} \tag{2.48}$$

Substituting into the equation 9.3 and developing the terms we can arrive to the momentum equation as a function of inertial velocity in the form:

$$\frac{\partial \rho \boldsymbol{v}^R}{\partial t} + \frac{d\boldsymbol{\Omega}}{dt} \times \boldsymbol{r} + \frac{\partial \rho \boldsymbol{v}^I v_i^R}{\partial x_i} + \boldsymbol{\Omega} \times \boldsymbol{v}^I = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_i}\left[\mu\left(\frac{\partial v_i^I}{\partial x_i} + \frac{\partial v_j^I}{\partial x_i}\right)\right] + \rho \boldsymbol{g} \tag{2.49}$$

while the continuity equation remains equal to the equation 9.2.
This method will be used in every steady state simulation of the wind turbine imposing a constant angular velocity in the MRF zone.

2.5 Dynamic Mesh

The problem can be solved also by means of dynamic grids. Instead of changing the reference frame we can impose a rotating motion directly on the grid. In order to do that we have to rewrite Navier-Stokes equations expressing the velocity as relative velocity between the absolute velocity of the fluid and the absolute velocity of the grid:

$$\begin{aligned}\frac{\partial \rho(\boldsymbol{v}-\boldsymbol{v_g})}{\partial t}+\frac{\partial \rho(\boldsymbol{v}-\boldsymbol{v}_g)(v-v_g)_i}{\partial x_i}=-\frac{\partial p}{\partial x_i}+\\ +\frac{\partial}{\partial x_i}\left[\mu\left(\frac{\partial (v-v_g)_j}{\partial x_i}+\frac{\partial (v-v_g)_i}{\partial x_j}\right)\right]+\rho\boldsymbol{g}\end{aligned} \tag{2.50}$$

We have to consider that also the volume is changing in function of time so we have to formulate the *space conservation law*:

$$\frac{\partial}{\partial t}\int_V \partial V-\oint_S \boldsymbol{n}\cdot\boldsymbol{v}_g\partial S=0 \tag{2.51}$$

In *OpenFoam* this method is embedded in the dynamic mesh version of the PIMPLE algorithm that is called *PimpleDyMFoam*[ope]. The latter will be used for every dynamic simulation of the wind turbine.

Chapter 3

Grids Generation

In this chapter the main characteristics of a finite volume grid are presented. Both the NASA meshes and the Pointwise in-house made meshes are described. The tetrahedral anisotropic extrusion and the three dimensional extrusion are presented.

3.1 Main Mesh Parameters

3.1.1 Structured Grids

Structured grids are the easiest kind of meshes and are used for simple geometries where low gradients are present. A mesh is considered structured when each element has the same number of neighbouring elements and can be defined by means of indexes, so it can be found incrementing or decrementing the indexes without knowing the exact position in the space. For 3D geometries it is common to use 3 indexes i, j, k oriented as the space axes x, y, z. Structured meshes do not need additional topological information because it is guaranteed by the use of indexes, thus reducing the memory required. Furthermore the resulting matrices have a fixed bandwidth[35].

3.1.2 Unstructured Grids

Unstructured terms allow to mesh with more flexibility in shape and in concentration of elements. They are more complex to create and to use since it is not possible to link each element with indexes. Each element is numbered sequentially so the topology has to be written explicitly, thus the memory usage increases with respect to the structured type. Each element can have a different number of neighbouring elements. The resulting matrices have not a fixed bandwidth and a renumbering algorithm, such as the Cuthil-McKee algorithm[32], is always needed for guaranteeing the efficiency of an iterative solver.

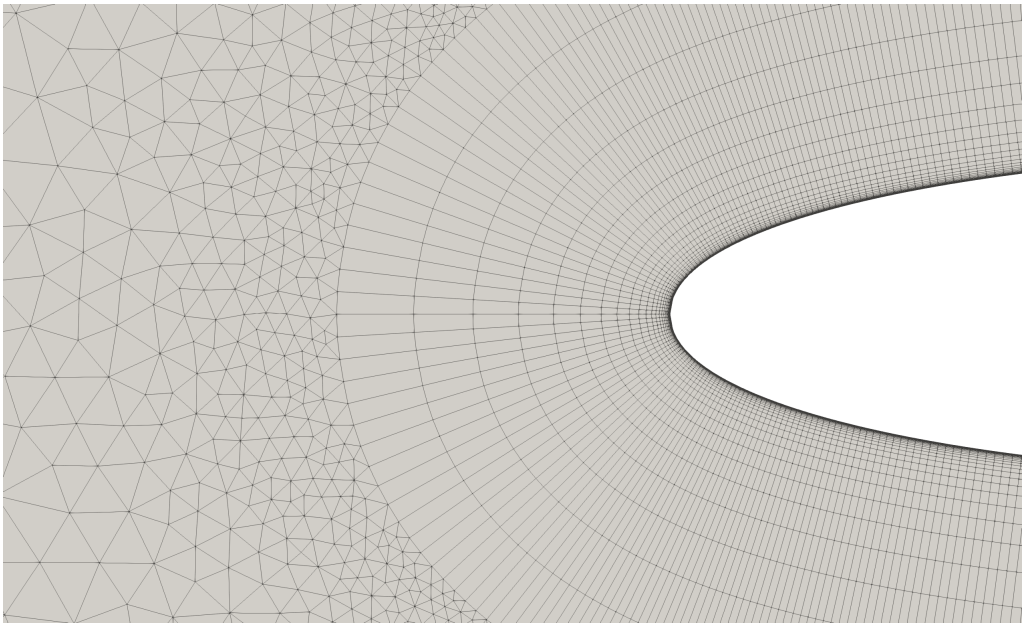


Fig. 3.1 Hybrid Grid - Structured Quad and Unstructured Tria

3.1.3 Non-Orthogonality

In every mesh that is not structured and orthogonal we can have a certain amount of non-orthogonality. The latter is defined by the angle formed by the segment joining two element centroids and the normal to the surface joining the two surfaces as in the picture. The lower the non-orthogonality the better is the mesh.

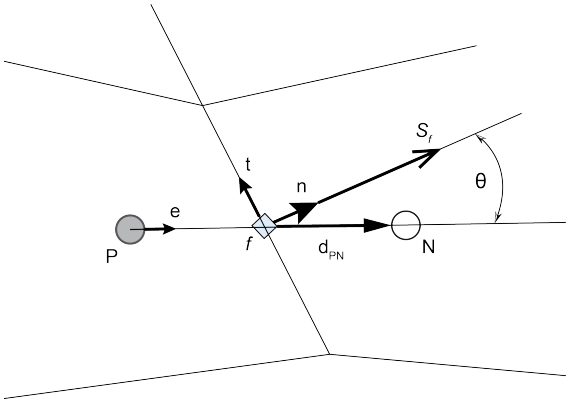


Fig. 3.2 Non-Orthogonal Elements

Non-orthogonality causes errors in the computation of surface gradients since $\boldsymbol{n}$ and $\overline{PN}$ are not collinear

$$(\boldsymbol{\nabla}\phi \cdot \boldsymbol{n})_f \neq \frac{\phi_P - \phi_N}{||\boldsymbol{r_P} - \boldsymbol{r_N}||} \tag{3.1}$$

For this reason the gradient must be divided in a orthogonal an non-orthogonal part:

$$(\boldsymbol{\nabla}\phi)_f \cdot \boldsymbol{S_f} = (\boldsymbol{\nabla}\phi)_f \cdot \boldsymbol{e} + (\boldsymbol{\nabla}\phi)_f \cdot \boldsymbol{t} \tag{3.2}$$

In order to solve this problem various techniques have been developed. The most efficacious one is the *Over Relaxed Approach* in which the term $(\boldsymbol{\nabla}\phi)_f \cdot \boldsymbol{t}$ is computed as:

$$(\boldsymbol{\nabla}\phi)_f \cdot \boldsymbol{t} = (\boldsymbol{\nabla}\phi)_f \cdot \left(\boldsymbol{n} - \frac{1}{cos\theta}\boldsymbol{e}\right) S_f \tag{3.3}$$

Since this term needs the value of $(\boldsymbol{\nabla}\phi)_f \cdot \boldsymbol{S_f}$ we must apply an iterative procedure. In *OpenFoam* the number of iterations is specified in the solvers parameters and is called *nNonOrthogonalCorrectos*[24].

3.1.4 Skewness

Due to the fact that some variables have to be computed at the centre of the face connecting two elements, the skewness is an important parameter that characterize the mesh quality. The latter is defined as the distance between

the middle point and the point obtained from the intersection between the line connecting two centroids and the face. A correction is needed to keep the discretization method second order accurate and is achieved by means of a Taylor expansion:

$$\phi_f = \phi_{f'} + (\nabla\phi)_{f'} \cdot \mathbf{d}_{f'f} \tag{3.4}$$

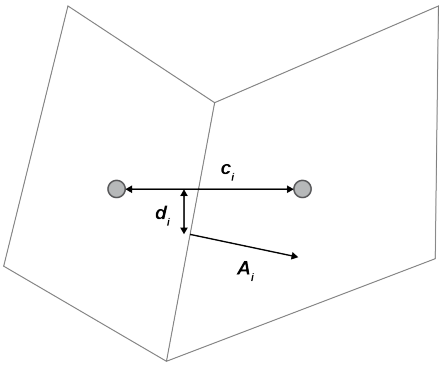


Fig. 3.3 Skewness

3.1.5 Aspect Ratio

In *OpenFoam* the aspect ratio is defined for each element of 2D grids as:

$$AR = \frac{max(a_x, a_y)}{min(a_x, a_y)} \tag{3.5}$$

and for 3D grids as:

$$AR = \frac{1}{6} \cdot \frac{|a_x| + |a_y| + |a_z|}{v^{\frac{2}{3}}} \tag{3.6}$$

For a good quality mesh the AR should be equal to 1. High AR values could cause interpolation errors.

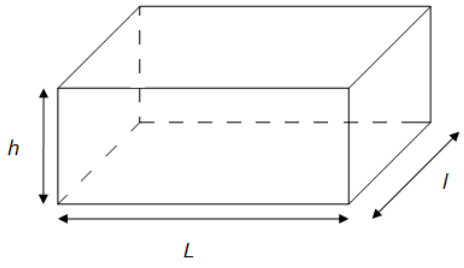


Fig. 3.4 Aspect Ratio

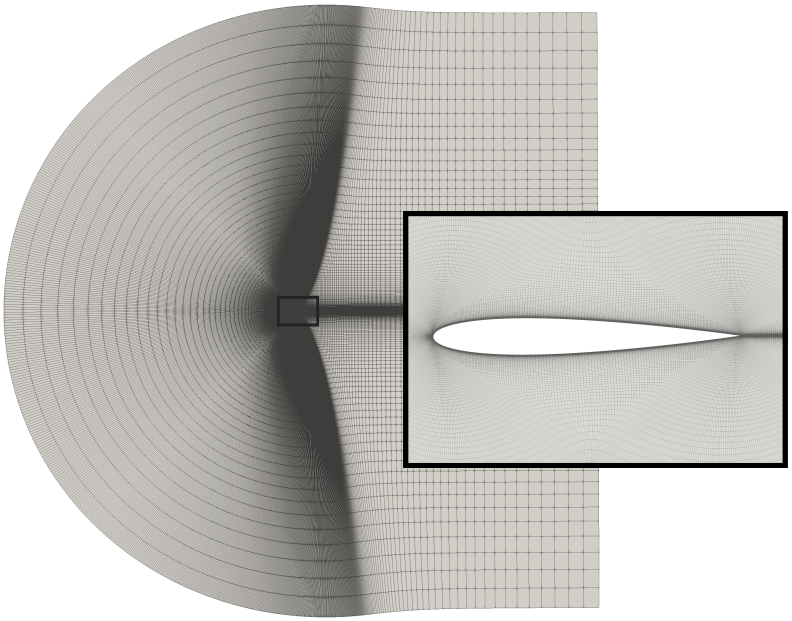
3.2 Structured C-grid

In order to numerically investigate airfoil aerodynamics two kind of grids are commonly used: the C-type grid and the O-type grid. Although they can give both reliable results, C-grids are generally preferred since they are easier to build and since it is less complicated to increase the number of elements behind the trailing edge, that is the region in which the grid is more sensitive[33].
The NASA Langley Research Center website provides five 2D C-grids for running the case[NLR]. Each grid was analysed with the OpenFoam function *checkMesh*. The table below shows the main characteristics of them. The mesh size is constituted by two numbers: the first one indicates the number of grid points in the airfoil while the second one the number of extrusion layers.

Mesh Size	**# Elements**	**Max Skewness**	**Average non-Orthogonality**	**Max non-Orthogonality**
113x63	3,584	0.5	15.0°	85.7°
225x65	14,336	0.5	10.2°	77.4°
449x129	57,344	0.3	5.0°	52.4°
897x257	229,376	0.2	1.6°	19.8°
1793x513	917,504	0.2	0.8°	56.9°

Table 3.1 NLRC Meshes

In every mesh the airfoil-upstream and airfoil-downstream distance is $500 \cdot c$, where c is the chord length. This distance is imposed in order to have the minimum influence on the result on the airfoil from the farfield and outlet boundary conditions. In order to assure a y^+ value between 0.1 and 0.2 the first layer extruded from the airfoil has a thickness of $4 \cdot 10^{-7}$.



(a) Nasa LRC C-Grid

3.3 3D Anisotropic Tetrahedral Extrusion

With the purpose of familiarizing with the *Pointwise* anisotropic tetrahedral extrusion tool, two c-grids were created. Furthermore we wanted to verify what is the difference between the solution that does not employ wall functions ($y^+ < 1$) and the one that employs wall functions without incurring int the buffer layer ($y^+ > 30$). For this reason the first layer, in the coarse version of the mesh, is $1e-5$ while in the fine mesh is $1e-7$.
In *Pointwise* the anisotropic tetrahedral extrusion, also called T-Rex, is an advancing layer algorithm that automatically generates resolved hybrid grids. T-rex starts from a surface triangular or quadrilateral elements and extrudes in the surface normal direction. It is possible to choose an initial height, the number of layer that we want to extrude and the growth rate. Vertices advances forming anisotropic right angle tetrahedra. As the vertices advances off from the surface, the resulting anisotropic tetrahedra are being checked against a set of quality criteria, some of which are defined by the user, and can stop locally if any of this criteria are violated but they are not going to prevent any

neighbouring elements from continuing to advance. The results are stacks of non-uniform layers anisotropic tetrahedra being generated which is advantageous for cases where surface cell sizes are going to vary and approaching fronts collide. At the end this stacks of tetrahedra can be combined to form triangular prisms, extruded from triangular surface mesh or even unstructured hexahedra mesh. If advanced from a quadrilateral or quad dominant surface mesh, they transform into isotropic tetrahedra in the farfield. The main parameters that can be selected are briefly described:

- **Layers**
  - **Max Layers**: maximum number of layers that has to be extruded before reaching isotropy
  - **Full Layers**: number of full layers that have to be extruded. Also indicates how many normals will protrude from sharp corners.
- **Boundary Conditions**
  Three kind of boundary conditions can be applied on the surfaces:
  - **Wall**: surfaces in which the layer are extruded with a given thickness Δs
  - **Match**: changes the grid spacing of a surface that is not undergoing extrusion but that is adjacent to a *t-rex* surface
  - **Adjacent**: it is similar to the wall but takes Δs automatically from adjacent surfaces.

- **Advanced Attributes**
  - **Isotropic Seed Layer**: specifies the number of layers of points that are created in the iso-portion of the volume mesh to improve resolution in the iso-region between two hexahedra regions.
  - **Collision Buffer**: specifies the minimum buffer to be maintained between encroaching fronts and it is a factor of the current cell height.
  - **Aniso-iso blend**: rate at which anisotropic elements on the surface are blended into isotropic elements on the interior of the volume mesh. Smaller volumes are going to preserve anisotropy of the front while larger values decrease the distance over which the decimation and transition occurs.

- **Smoothing**
  It is an iterative algorithm that solves the sharpness in transition between different size adjacent cells.

After the t-rex extrusion structured and unstructured grids always need a solver for improving the quality and avoiding the aforementioned problems. The main options of the *Pointwise* solver are briefly listed:

- **Unstructured Meshing Algorithms**: used both for solving and for generating a grid. *Pointwise* includes three algorithms that are used depending on the case needs.

  In every algorithm, we can choose to have triangles and triangles and quads. In our case only Delanuay algorithm has been used.

- **Max Edge Growth Rate**: specifies how much is the growth rate from structured to unstructured mesh as ratio of the elements length.

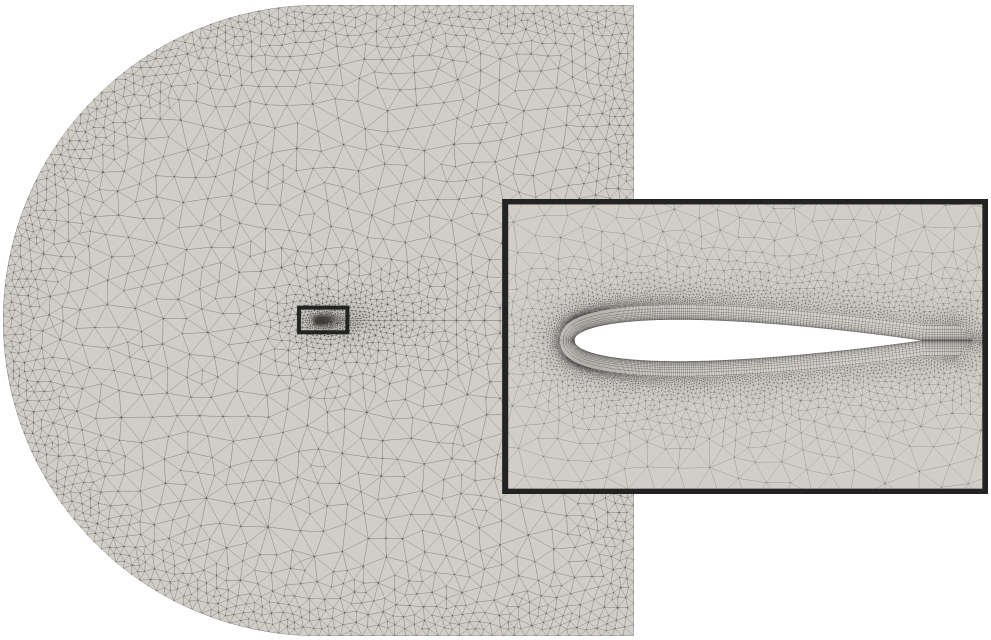
- **Pyramid Settings**
  Pyramid elements are used for the transition between hexahedral and tetrahedral elements. The following parameters can be chosen:
  - **Min. Height**
  - **Max. Height**
  - **Aspect Ratio**: we can decide the aspect ratio of the pyramids, that implies the flatness and sharpness and consequentially the max. included angle.
  - **Edge Length**
  - **Boundary Decay**: determines how far the dense edge clustering travels into the interior of the domain.

The main characteristics of the two grids are resumed in the following table:

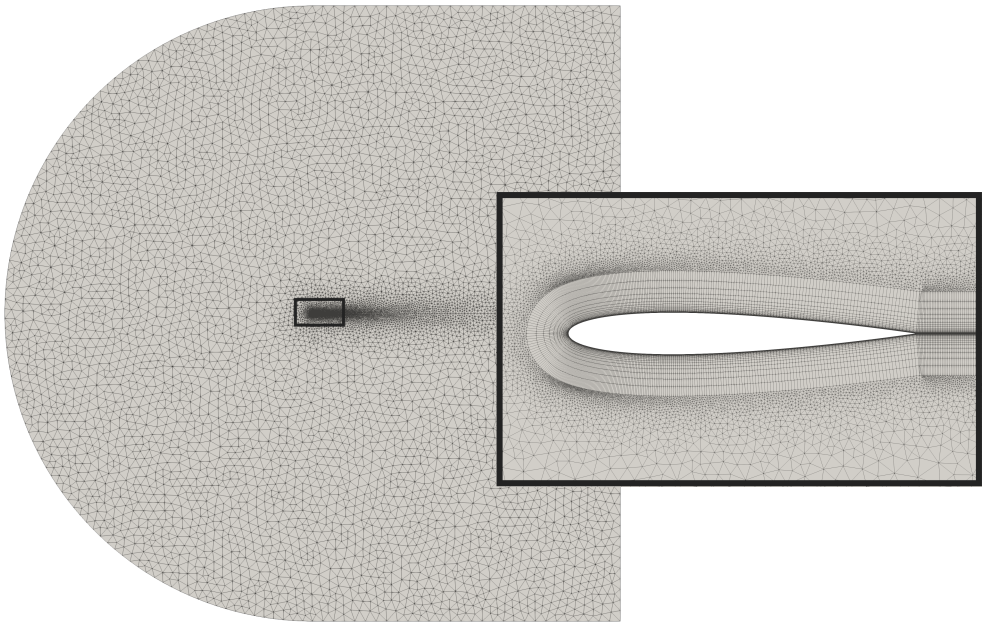
Mesh Size	**Elements**	**Max Skweness**	**Average non-Orthogonality**	**Max non-Orthogonality**
Pointwise Coarse	75.259	0.5	4.50°	37.93°
Pointwise Fine	77.987	0.8	3.33°	88.97°

Table 3.2 Pointwise Mesh

The two grids created are shown in the pictures below.



(a) Pointwise In-House **Coarse** C-Grid



(a) Pointwise In-House **Fine** C-Grid

3.4 Airfoil Extrusion for 3D Simulations

With the purpose of measuring the difference between 2D and 3D simulations a two dimensional mesh was extruded with the *OpenFoam* command *extrudeMesh*. The latter needs a file in the *system* folder called *extrudeMeshDict*. The extrusion was done linearly from the *front* patch and has 50 layers distributed in 10 meters. The expansion ratio was 1, the thickness of each layer is the same as shown in the picture.

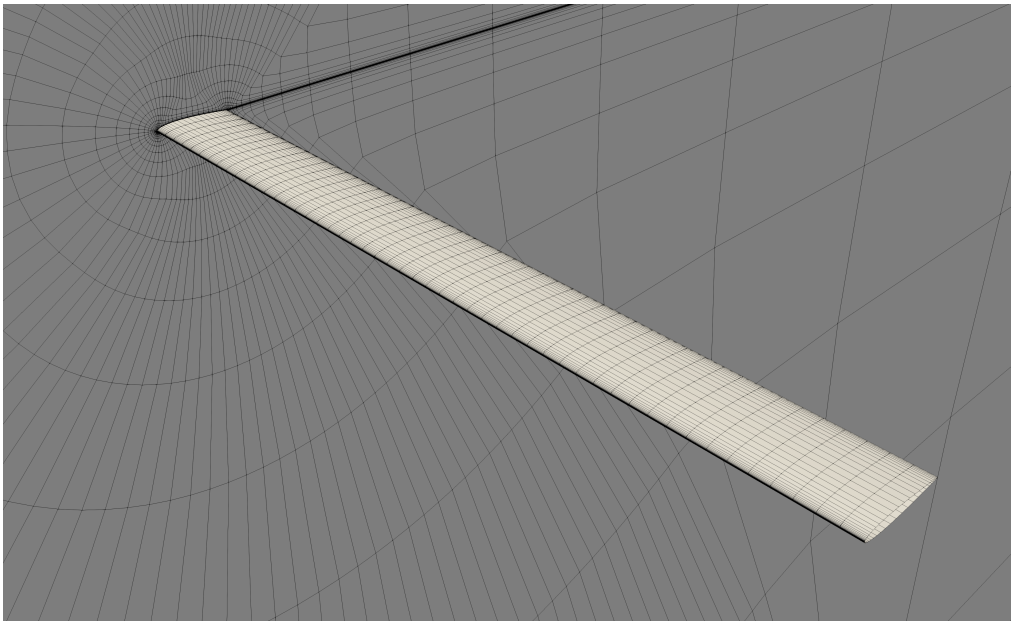


Fig. 3.8 3D Mesh Extrusion

3.5 DTU 10MW - In-House Pointwise Grids

With the purpose of finding the right size for having one mesh with $y^+ < 1$ and one with $y^+ > 30$ four grids were created with four different first element lengths. The CFD domain is constituted by one blade without any nacelle at the base that was removed in order to simplify the meshing procedure and to see the effects of the removal on the results. The grid is of the hybrid type and has been obtained with the aforementioned 3d anisotropic tetrahedral extrusion technique. The extrusion from the blade surface is structured and constituted by 60 hexahedra layers that become first pyramids and then unstructured tetrahedra which in turn become gradually isotropic tetrahedra. For the background mesh one third of a cylinder was token. This part has a finer mesh in the so called MRF zone, two coarser unstructured blocks in the front and in the back of the blade and the coarsest mesh block that covers every other block and extends radially from them.

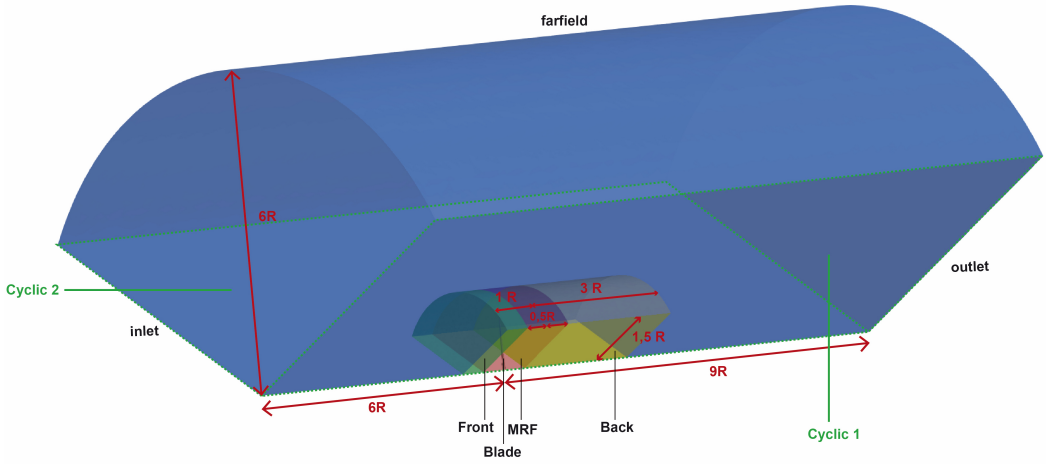


Fig. 3.9 DTU 10MW CFD Domain

The domain grids were fully created with the software *Pointwise* and they constitute one of the achievements of this thesis. More than one month was spent for creating the domain grid because it was not possible to use automatic Pointwise tools since we wanted the full control of every parameter and due to some problems caused by the blade shape. The main characteristics of the

grids were computed with the *OpenFoam* function *checkMesh* that are reported in the table below.

	First Layer Length	Elements	Max Non-Ort.	Average Non-Ort.	Non-Ort. >70°	Max Skewness	Max. Aspect Ratio
Pointwise	3.00E-005	6.8E+6	84.786	24.0	3402	2.9	29410.0
Pointwise	3.00E-004	6.0E+6	82.225	24.4	1522	3.1	3291.1
Pointwise	1.00E-003	6.0E+6	81.39	21.8	1317	2.9	995.7
Pointwise	3.00E-005	4.5E+6	85.314	18.9	6502	2.8	382.6
D.T.U.	2.00E-006	11.8E+6	/	/	/	/	/
U.o.S.	/	13E+6	/	/	/	/	/
U.MONS	3.00E-005	7.2E+6	/	/	/	/	/

Table 3.3 Mesh Characteristics Comparison

The characteristics can be compared with the one obtained by the Denmark Technical University, the University of Stuttgart and the University of Mons [11] [25] [48]. The three mesh compared are structured and include one third of the nacelle. The first two were created with *Pointwise* while U.MONS used $Autogrid5^{TM}$. Thanks to the hybrid technique the grids produced in this thesis have always a lower number of elements even if the dimensions are the same or similar. In fact the overall dimensions were taken from the paper published by the University of Stuttgart [47] that made a sensitivity analysis varying the grid dimension. Thus, we decided to take as reference length R, the blade radius, and to use $6R$ for the cylinder radius, $6R$ as distance from the blade to the front inlet and $9R$ as distance from the blade to the back outlet. Then, since we were dealing with an hybrid mesh, we decided to include three finer mesh blocks: MRF(red in the picture), that extends $0.5R$ in the front of the blade, $0.5R$ in the back of the blade, $1.5R$ radially and has an average node spacing of $1.5m$; *front*(green in the picture), the prolongs $0.5R$ from the MRF front and has an average node spacing of $4m$ and *back*(yellow in the picture) that extends $2.5R$ the back of MRF and has an average node spacing of $4m$. The remaining part of the background mesh has an average spacing of $20m$. The size of *front* and *back* blocks were taken from *Pointwise* tutorials while the spacing of the elements was chosen looking to the quality of the mesh, to the total number of elements and to horizontal cuts of the domain for seeing the pressure and velocity fields in order to avoid unexpected spikes.

The grid creation starts from the blade CAD file that can be freely downloaded from the DTU website. The surface mesh has 224 cells in the chord-wise direction and 100 cells in the span-wise direction. Every spacing was manually selected in order to assure low skewness and low non-orthogonality.

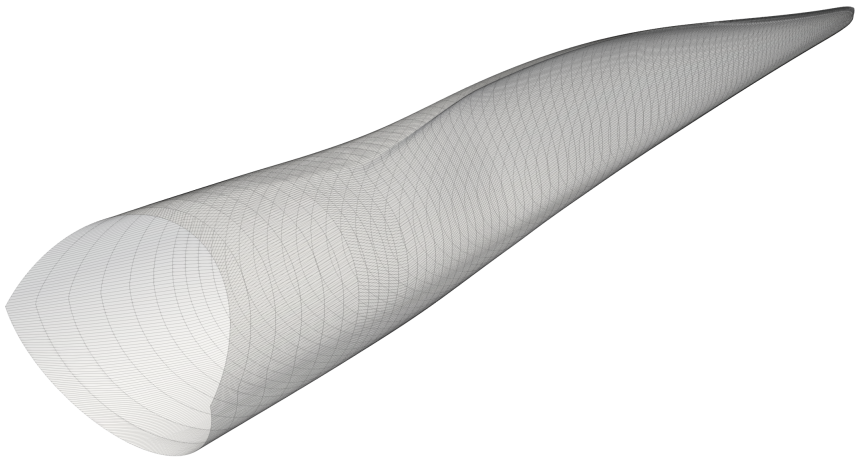


Fig. 3.10 Blade Surface Mesh

The surface mesh was then extruded for 60 structured hexaehedra layers starting from a precise first element length (see table 9.1) with an expansion ratio of 1.2. This procedure was one of the most critical parts because of the following issues:

- **Sharpness of the Gurney Flap**: this problem caused high non-orthogonality in the extruded elements. We tried to solve this problem by extruding one by one the cells in this part assigning the proper shape to the extrusion line. Although the non-orthogonality decreased we still had some elements with a non-orthogonality bigger than 89ř. Thus we decided to solve the surface with an orthogonal algorithm and with a very low

relaxation factor. This procedure rounded a little bit the sharp surfaces that became extrudable.

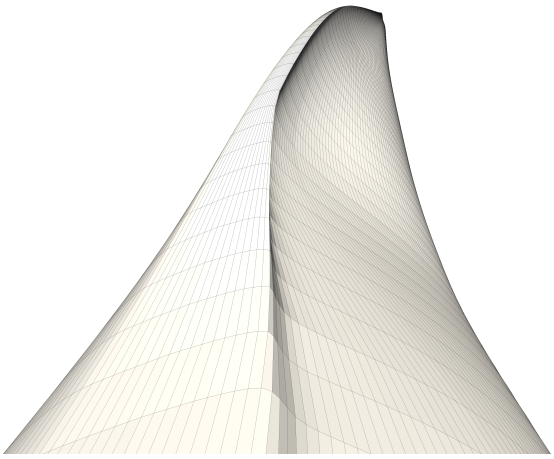


Fig. 3.11 Gurney Flap

- **Pointwise Non-Orthogonality Measurement**: the software is only capable of measuring the maximum included angle, the skewness and other parameters possessed by each element but does not measure the correlation between the neighbouring elements. Thus it is not possible to measure non-orthogonality and skewness as defined in chapter 5.

- **Automatic Extrusion**: *Pointwise* allows to extrude from a surface in the normal direction with a specific initial length for the first element and an expansion ratio. However it was not possible to extrude 60 layers by means of this tool because after a certain number of extruded layers it stopped since there were elements with negative Jacobian (inverted elements). The problem was solved by automatically extruding just the first 30 layers and then proceeding manually. Single lines were created from the resulting surface that were cut in the chord wise direction in six parts in order to create horizontal surface grids that were subsequently jointed by means of vertical connectors. Then horizontal surface grids

were created from two horizontal connectors and two vertical connectors both in the perpendicular and in the parallel direction. The resulting six volumes were initialized to form a structured grid.

- **Tip Problem**: as we can see from the figure 9.4 the elements extruded around the blade are very thin (in order to assure the wanted y^+) and the spacing between the nodes in the chord-wise direction is not as small as them. For this reason it is not possible to directly extrude in the vertical direction if we want to avoid significant differences between two neighbouring elements.

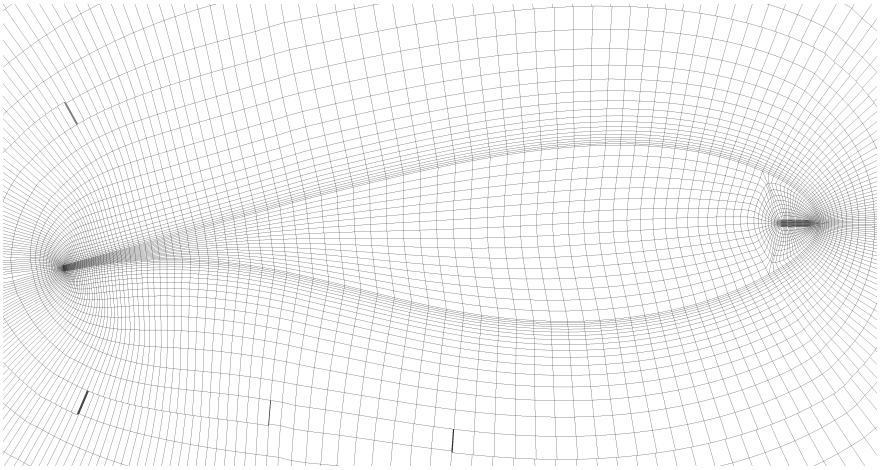


Fig. 3.12 Tip of the Extruded Mesh

For this reason the tip of the blade was copied, enlarged and pasted $30m$ above the original tip. Then, again using connectors, horizontal surface grids were created and then a volume mesh was initialized. Then the airfoil was horizontally extruded and the procedure previously described for the blade was repeated.

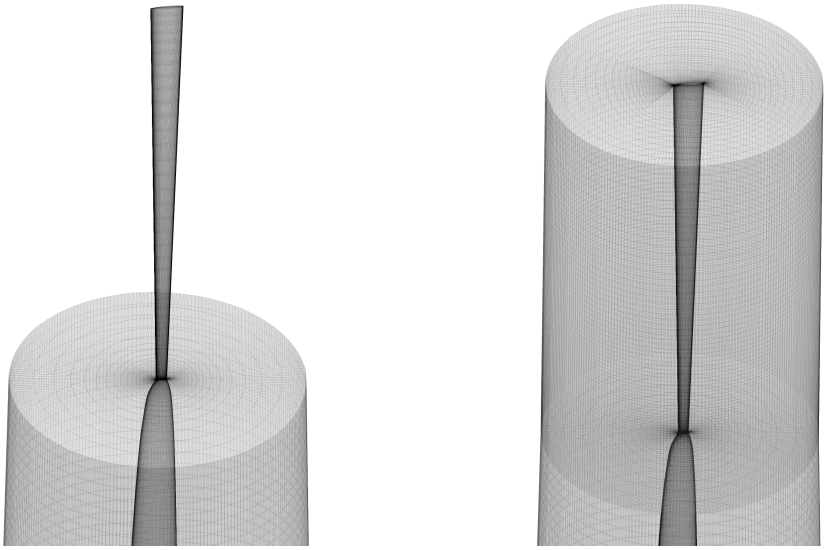


Fig. 3.13 Tip Extrusion

At the end of this procedure we obtained a structured mesh as in the picture below.

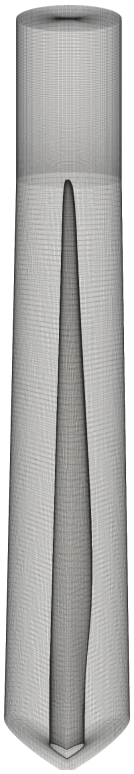


Fig. 3.14 Structured Blade Mesh Extrusion - 3e-5

The MRF zone was then created using connectors with an appropriate node spacing and projecting the resulting unstructured surface grids at the base in order to have an angle of 120ř between them.

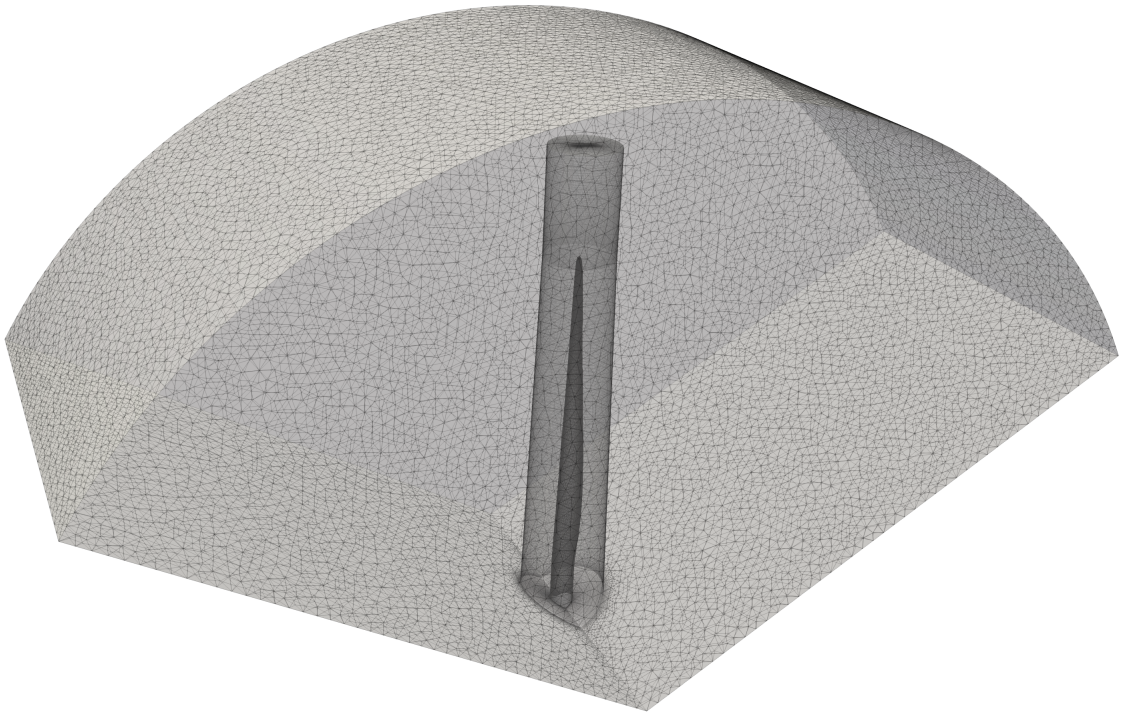


Fig. 3.15 MRF Zone

In order to create the unstructured volume mesh the unstructured surface grid of the MRF and the structured surface grid of the blade extruded has to be initialized by a *Pointwise* tool. This tool automatically creates pyramids onto the structured surface grids and fulfil the volume with tetrahedra respecting the spacing imposed in the surfaces. The selectable parameters of this tool are described in chapter 6. However this tool works only if the cells onto the structured surface have an acceptable aspect ratio. For this reason it was not possible extruding just 30 layers from the blade surface but it was mandatory to execute the aforementioned manual extrusion procedure. The two pictures show the differences between the two aspect ratios obtained.

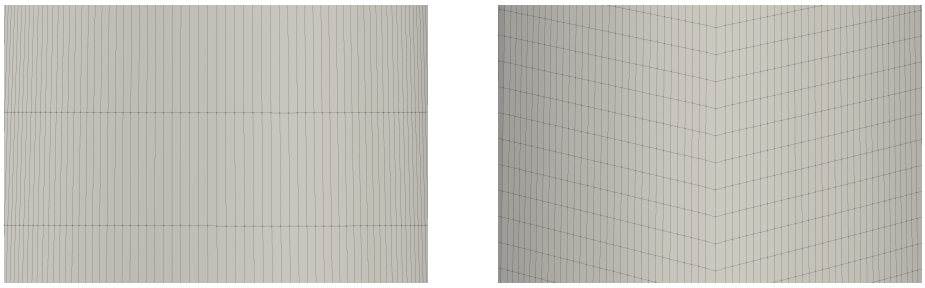


Fig. 3.16 Elements aspect ratio before and after the extrusion

This problem was present also in the tip surface that was solved with an orthogonal algorithm.

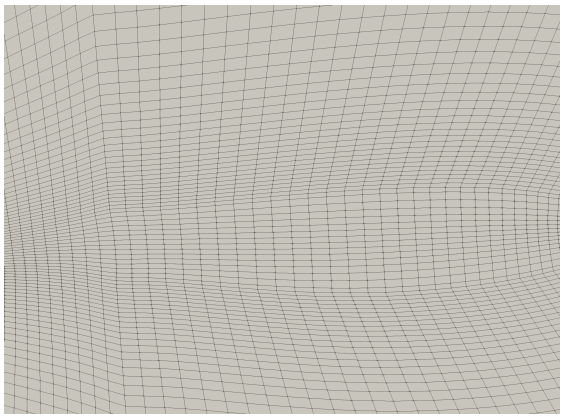


Fig. 3.17 Tip Extruded Surface Solved

At the end we obtained an hybrid MRF section that provides enough accuracy in the structured part and a significant reduction of elements in the unstructured part.

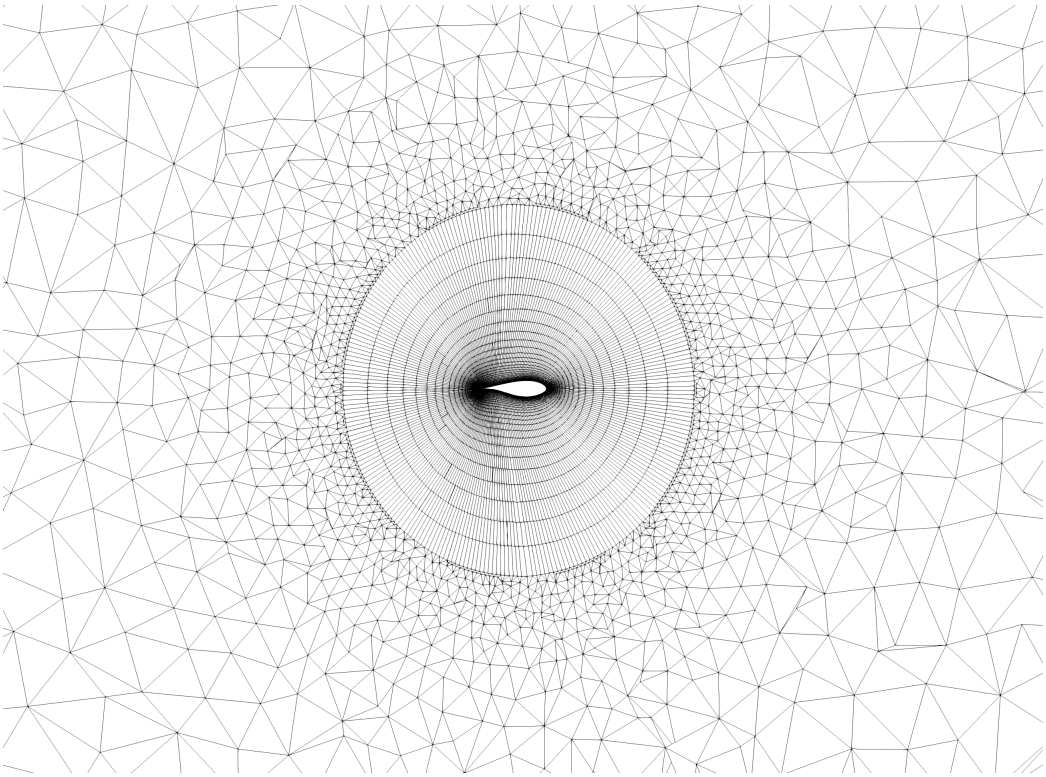


Fig. 3.18 Mesh Horizontal Cut - 3e-5

The remaining part of the background grid was done by means of connectors with the already mentioned node spacing and by projecting the base surfaces in order to form an angle of 120ř.

Chapter 4

NACA0012 Results

In this chapter the simulations done and the relative results obtained on the NACA0012 airfoil profile are illustrated.
Various numerical experiments were done in order to understand:

- the influence of the mesh size on the accuracy and on the convergence speed;
- the convergence speed difference between SIMPLE and SIMPLEC algorithm;
- the most appropriate turbulence models for the case;
- the influence on the discretization scheme of divergence terms;
- the difference between direct solution and wall function solution;
- the difference between 2-dimensional and 3-dimensional results;
- the difference between steady state and transient results.

The NASA Langley Research Center website provides the following physical parameters and boundary condition for the case validation:

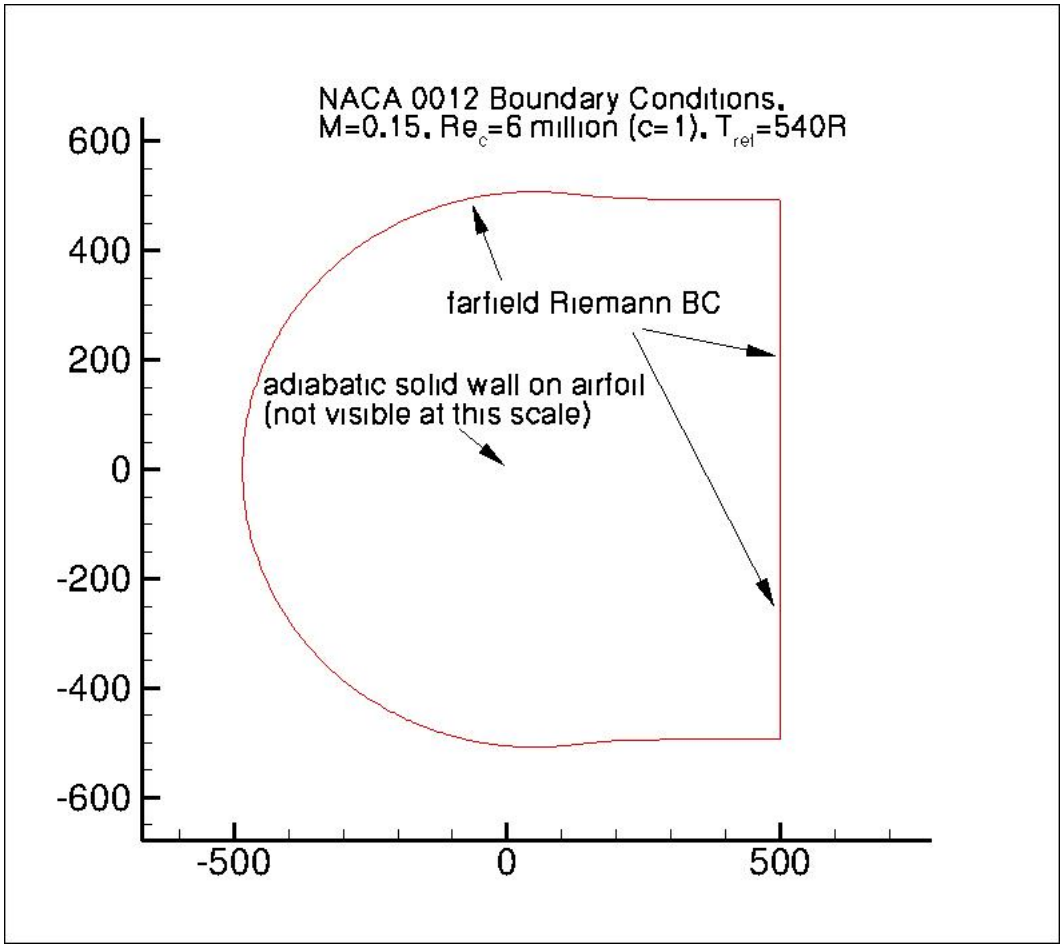


Fig. 4.1 Physical Parameters and Boundary Conditions

The results of each simulations will be compared with both numerical and experimental results. The numerical results are obtained by NASA LRC [NLR] from different codes and with different turbulence models. For their simulations the second finest mesh (897x257) has been used.

	Spalart-Allmaras					
	C_l			C_d		
Code	$\alpha = 0$	$\alpha = 10$	$\alpha = 15$	$\alpha = 0$	$\alpha = 10$	$\alpha = 15$
cfl3d	approx.0	1.0909	1.5461	0.00819	0.01231	0.02124
fun3d	approx.0	1.0983	1.5547	0.00812	0.01242	0.02159
nts	approx.0	1.0891	1.5461	0.00813	0.01243	0.02105
joe	approx.0	1.0918	1.549	0.00812	0.01245	0.02148
sumb	approx.0	1.0904	1.5446	0.00813	0.01231	0.02141
turns	approx.0	1.1000	1.5642	0.0083	0.01233	0.02140
ggns	approx.0	1.0941	1.5576	0.00817	0.0123	0.02073
overflow	approx.0	1.0990	1.5576	0.00838	0.01251	0.02149
Average	0	1.0942	1.5525	0.0082	0.0124	0.0213
	$k\omega SST$					
	C_l			C_d		
Code	$\alpha = 0$	$\alpha = 10$	$\alpha = 15$	$\alpha = 0$	$\alpha = 10$	$\alpha = 15$
cfl3d	approx.0	1.0909	1.5461	0.00819	0.01231	0.02124
fun3d	approx.0	1.0983	1.5547	0.00812	0.01242	0.02159
nts	approx.0	1.0891	1.5461	0.00813	0.01243	0.02105
overflow	approx.0	1.0990	1.5576	0.00838	0.01251	0.02149
Average	0	1.0943	1.5511	0.0082	0.0124	0.0213

Table 4.1 NASA LRC Numerical Results

Experimental Data – Ladson, NASA, 1988					
C_l			C_d		
$\alpha = 0$	$\alpha = 10$	$\alpha = 15$	$\alpha = 0$	$\alpha = 10$	$\alpha = 15$
-0.0126	1.0707	1.5129	0.00809	0.01201	0.019

Table 4.2 NASA Experimental Results

The simulations are executed with three different angles of attack. Instead of varying the mesh, the inlet velocity is projected creating an artificial angle of attack without moving the airfoil. Some of the simulations that will be described were run on multiple processors. OpenFoam is capable of parallelizing

a simulation and gives also the possibility to choose the decomposition algorithm, in our case the *scotch* algorithm has been used. When available, the cases were run on the cluster of the *Statik Department*.

4.1 Convergence Study on NASA Grids

The purpose of this study is to find which NLRC grid is better in terms of *computational cost-accuracy*. The CFD domain is the one described in chapter 3. For this case the Spalart-Allmaras turbulence model was used and the angle of attack was set to 10°. In order to see just the effect of the mesh size every discretization scheme and every solver is the same and have the same parameters. Every case was run for 30.000 iterations. In the table 4.1 the boundary conditions used for running this case are shown. We have to consider that the y^+ values are always lower than one so it is possible to run the cases without using any wall function.
We imposed:

$$\tilde{\nu} = 3 \cdot \nu \tag{4.1}$$

$$\nu_t = \tilde{\nu} \frac{\chi^3}{\chi^3 + c_{v_1}^3} \tag{4.2}$$

Spalart–Allmaras					
Variables		**Farfield**	**Outlet**	**Airfoil**	**Front-Back**
Pressure	type	zeroGradient	fixedValue	zeroGradient	empty
	value	/	Uniform 0	/	empty
Velocity	type	fixedValue	zeroGradient	zeroGradient	empty
	value	Uniform u_∞	/	/	empty
$\boldsymbol{\nu_t}$	type	fixedValue	zeroGradient	fixedValue	empty
	value	Uniform 3.07e-06	/	Uniform 0	empty
$\boldsymbol{\tilde{\nu}}$	type	fixedValue	zeroGradient	fixedValue	empty
	value	Uniform 0.0000438	/	Uniform 0	empty

Table 4.3 Spalart - Allmaras Boundary Conditions

In order to see which is the convergence speed we were monitoring the drag and lift coefficient. After a certain number of iterations, the coefficients converge at a certain value. For both the coefficients it is evident that the coarser the mesh the higher the convergence speed as we expected.
In regard to the lift coefficient, the three finest mesh converge approximately to the same result. Instead the drag coefficient is more sensitive to the cell size, thus the third finest mesh and second finest mesh converge to two different results while the finest mesh still has to converge, meaning that it needs more than 30.000 iterations.

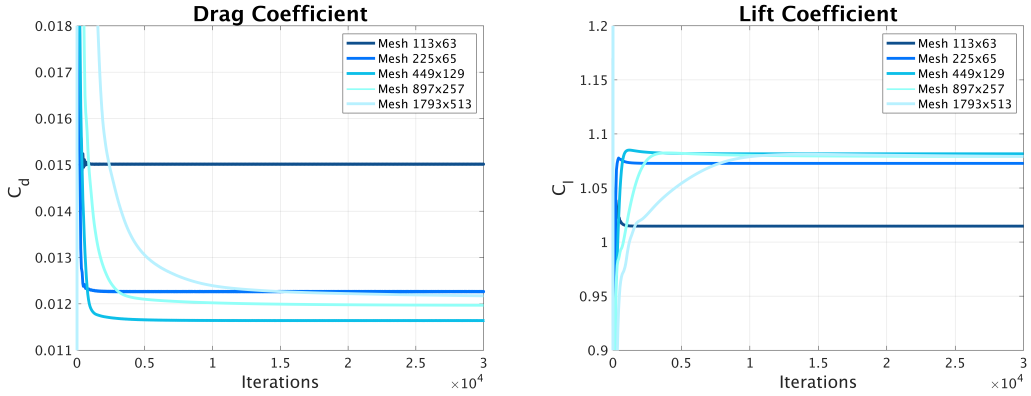


Fig. 4.2 Convergence Study

In the table 4.1 a comparison between the results obtained from the different meshes is made. The two coefficients are compared with the average numerical results obtained by NASA LRC. Notice that these simulations were not done in order to capture accurate results but just to see what is the influence of the mesh size. The errors in the drag coefficient are due to first order discretization schemes and low relaxation factors that slow down the convergence speed.

Convergence Study				
Mesh Size	C_l	**Num. Error**	C_d	**Num. Error**
113x63	1.0145	7.27%	0.00585	52.82%
225x65	1.0725	1.97%	0.00538	56.61%
449x129	1.0815	1.14%	0.00537	56.72%
897x257	1.0791	1.36%	0.00573	53.81%
1793x513	1.0792	1.38%	0.00590	52.40%

Table 4.4 Convergence Study

Also C_p curves are compared but, since they are almost overlapped one to the other except for the coarsest one, we cannot appreciate any difference in the accuracy of the results. The curves are compared with NASA experimental data.

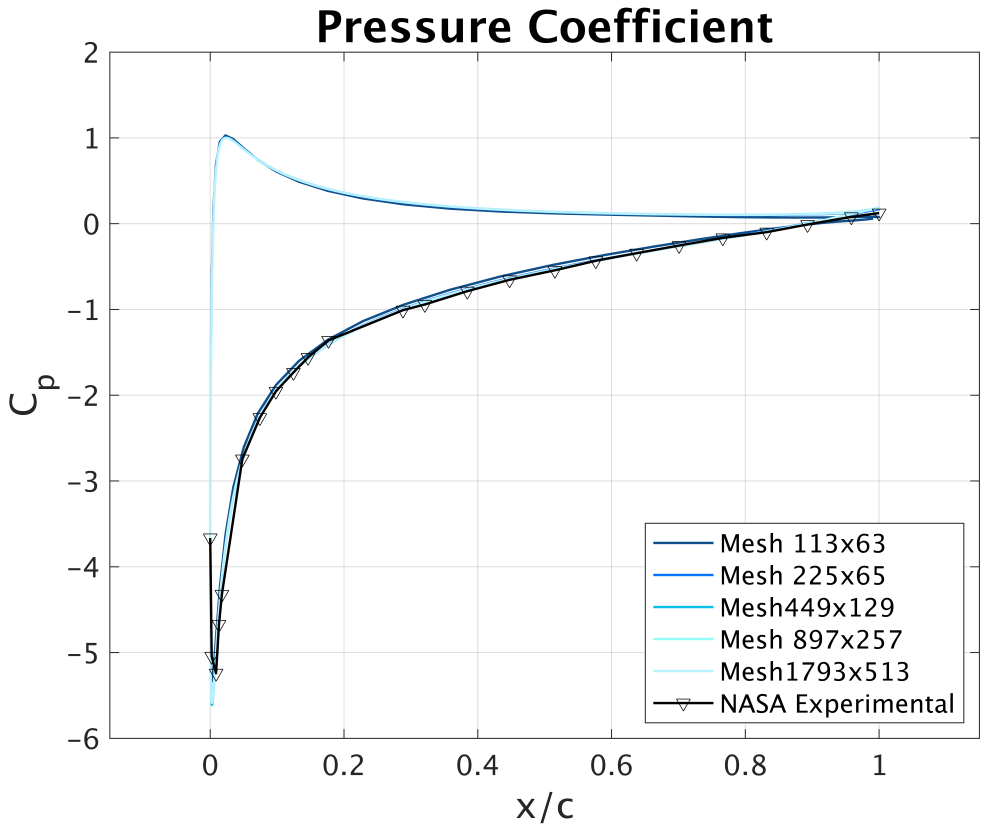


Fig. 4.3 C_p curve - Spalart-Allmaras - 10° angle of attack

As found also by NASA, the second finest mesh(897x257) seems to be the most appropriate one since it can converge in 30.000 iterations and seems to be as accurate as the finest one. Further studies done with this mesh will confirm that this grid can provide a very accurate prediction of the lift and drag coefficient.

4.2 SIMPLE-Consistent Algorithm Study

SIMPLEC algorithm [56] is an enhancement of the SIMPLE algorithm [42] obtained by introducing a consistent approximation on the pressure correction term, that does not need an under-relaxed approximation. In order to capture the difference in convergence rate the case has been run with Spalart-Allmaras Turbulence model, 10° angle of attack and 897x257 mesh size. Six simulations were run: three using SIMPLE algorithm and three using SIMPLE-C algorithm. For the six cases we used three different relaxation factors configurations, three for each algorithm.

Relaxation Factors			
Configuration	**Pressure**	**Velocity**	$\tilde{\nu}$
1	0.3	0.5	0.5
2	0.4	0.6	0.6
3	0.6	0.9	0.9

Table 4.5 Relaxation Factor Configurations

Here are shown the C_d-Iteration and C_l-Iteration curves.

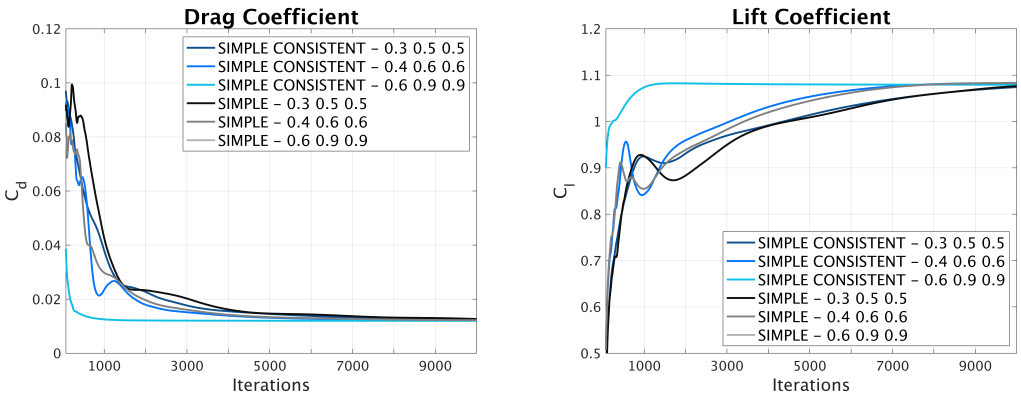


Fig. 4.4 SIMPLE-SIMPLEC Algorithm Convergence Study

As we expected from the theory, the higher the relaxation factors the faster will be the convergence rate. It is noteworthy that with equal relaxation factors SIMPLE-C converges always faster with respect to SIMPLE. In the configuration 3 the SIMPLE algorithm diverges immediately while the SIMPLE-C converges at the fastest rate.
For these reasons SIMPLEC will be always preferred to SIMPLE, and relaxations factors will be generally higher than the standard ones.

4.3 Turbulence Model Validation

As explained in chapter 2 every turbulence model is created for specific applications, for instance someone work better when inertial forces are dominant and some when viscous forces are dominant.
We wanted to understand what are the results from some of the most known turbulence models that are the ones described in chapter 2. Every result obtained is compared with both NASA numerical and NASA experimental results. As mentioned in the introduction to this chapter, NASA numerical results are very similar to the other and they are also close to experimental results. We expect to have the highest error for the highest angle of attack since we have more separation of the fluid vein.
We present here what are the boundary condition for each model.

$k\omega SST$					
Variables		**Farfield**	**Outlet**	**Airfoil**	**Front-Back**
Pressure	type	zeroGradient	fixedValue	zeroGradient	empty
	value	/	Uniform 0	/	empty
Velocity	type	fixedValue	zeroGradient	zeroGradient	empty
	value	Uniform u_∞	/	/	empty
ν_t	type	fixedValue	zeroGradient	fixedValue	empty
	value	1.31E-06	/	Uniform 0	empty
k	type	fixedValue	zeroGradient	fixedValue	empty
	value	6.91E-05	/	Uniform 0	empty
ω	type	fixedValue	zeroGradient	omegaWallFunction	empty
	value	Uniform 525.6	/	Uniform 525.6	empty

$k\epsilon$					
Variables		**Farfield**	**Outlet**	**Airfoil**	**Front-Back**
Pressure	type	zeroGradient	fixedValue	zeroGradient	empty
	value	/	Uniform 0	/	empty
Velocity	type	fixedValue	zeroGradient	zeroGradient	empty
	value	Uniform u_∞	/	/	empty
ν_t	type	fixedValue	zeroGradient	nutUWallFunction	empty
	value	Uniform 1.31e-07	/	Uniform 1.31e-07	empty
k	type	fixedValue	zeroGradient	kqRWallFunction	empty
	value	Uniform 7.67e-06	/	Uniform 7.67e-06	empty
ϵ	type	fixedValue	zeroGradient	epsilonWallFunction	empty
	value	Uniform 4.03e-05	/	Uniform 4.03e-05	empty

$kk_l\omega$					
Variables		**Farfield**	**Outlet**	**Airfoil**	**Front-Back**
Pressure	type	zeroGradient	fixedValue	zeroGradient	empty
	value	/	Uniform 0	/	empty
Velocity	type	fixedValue	zeroGradient	zeroGradient	empty
	value	Uniform u_∞	/	/	empty
ν_t	type	fixedValue	zeroGradient	fixedValue	empty
	value	Uniform 1.31e-06	/	Uniform 0	empty
k_t	type	fixedValue	zeroGradient	fixedValue	empty
	value	Uniform 7.67e-06	/	Uniform 1e-10	empty
k_l	type	fixedValue	zeroGradient	fixedValue	empty
	value	Uniform 1e-10	/	Uniform 1e-10	empty
ω	type	fixedValue	zeroGradient	omegaWallFunction	empty
	value	Uniform 525.6	/	Uniform 525.6	empty

Boundary conditions for $k\omega SST$ and for $k\epsilon$ were taken from FLUENT manual [19].

$$\nu_t = C_\mu \nu \tag{4.3}$$

$$k = k_\infty u_\infty^2 \tag{4.4}$$

$$\epsilon = \frac{\epsilon_\infty \rho u_\infty^4}{\mu} \tag{4.5}$$

$$\omega = \frac{\rho u_\infty^2 \omega_\infty}{\mu} \tag{4.6}$$

For $k\omega SST$ model:

$$\begin{aligned} k_\infty &= 9e-09 \\ \omega_\infty &= 1e-06 \end{aligned} \tag{4.7}$$

For $k\epsilon$ model:

$$\begin{aligned} k_\infty &= 1e-09 \\ \omega_\infty &= 1e-17 \end{aligned} \tag{4.8}$$

In regard to $kk_l\omega$ model we choose the airfoil chord length as turbulence length scale and we compute the boundary conditions with the following formulas:

$$k = \frac{3}{2}\,(U\,I)^2 \tag{4.9}$$

$$\omega = \frac{\sqrt{k}}{l} \tag{4.10}$$

After several attempts and an appropriate selection of solvers and discretization schemes the following results were obtained.

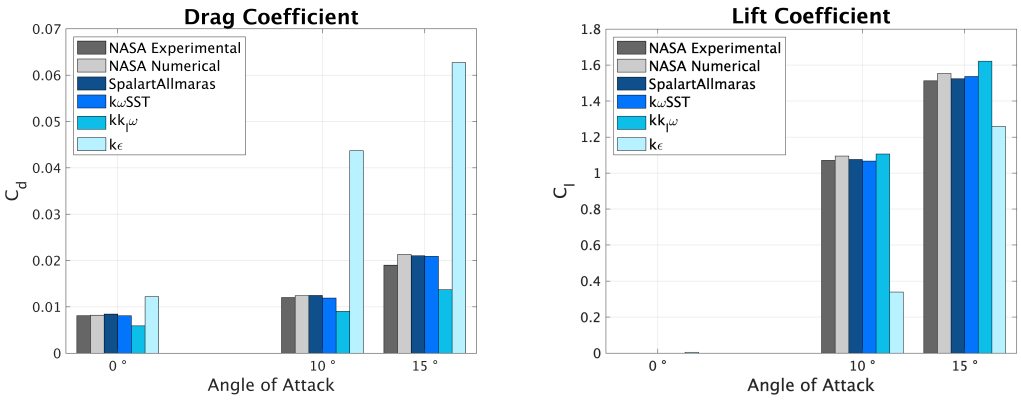


Fig. 4.5 Turbulence Model Validation

It is clear from the pictures that the only models that work properly are Spalart-Allmaras and $k\omega SST$, however we cannot find uniquely which one is the best.

Turbulence Model Comparison						
	C_l			C_d		
	$\alpha = 0$	$\alpha = 10$	$\alpha = 15$	$\alpha = 0$	$\alpha = 10$	$\alpha = 15$
Experimental Data Ladson, NASA, 1988	-0.0126	1.0707	1.5129	0.00809	0.01201	0.019
Average Numerical NASA Data	0	1.0940	1.5525	0.0082	0.0124	0.0213
OpenFoam Spalart-Allmaras	0	1.075	1.5244	0.00845	0.01246	0.02103
OpenFoam $k\omega SST$	0	1.0666	1.537	0.00808	0.0119	0.0209
OpenFoam $k\epsilon$	0.0046	0.3397	1.259	0.0122	0.0437	0.0627
OpenFoam $kk_l\omega$	0	1.1059	1.621	0.0059	0.00903	0.0137

Table 4.6 NASA and obtained results

Also by looking the results in digit form it is hard to find which model is better in terms of accuracy between the two. As expected, the errors are higher for the highest angle of attack.

Lift Coefficient Errors						
Turbulence Model	$\alpha = 0$		$\alpha = 10$		$\alpha = 15$	
	Exp.	**Num.**	**Exp.**	**Num.**	**Exp.**	**Num.**
OpenFoam Spalart-Allmaras	approx. 0%	approx. 0%	-0.40%	1.74%	-0.76%	1.81%
OpenFoam $k\omega SST$	approx. 0%	approx. 0%	0.38%	2.50%	-1.59%	1.00%
OpenFoam $k\epsilon$	approx. 0%	approx. 0%	68.27%	68.95%	16.78%	18.90%
OpenFoam $kk_l\omega$	approx. 0%	approx. 0%	-3.29%	-1.09%	-7.15%	-4.41%

Table 4.7 Lift Coefficient Errors

Drag Coefficient Errors						
Turbulence Model	$\alpha = 0$		$\alpha = 10$		$\alpha = 15$	
	Exp.	**Num.**	**Exp.**	**Num.**	**Exp.**	**Num.**
OpenFoam Spalart-Allmaras	-4.45%	-3.05%	-3.75%	-0.48%	-10.68%	1.27%
OpenFoam $k\omega SST$	0.12%	1.46%	0.92%	4.03%	-10.00%	1.88%
OpenFoam $k\epsilon$	-50.80%	-48.78%	-263.86%	-252.42%	-230.00%	-194.37%
OpenFoam $kk_l\omega$	27.07%	28.05%	24.81%	27.18%	27.89%	35.68%

Table 4.8 Drag Coefficient Errors

Also in this case we compared the C_p curves. For 0° angle of attack every model can successfully replicate the curve, while for higher angles of attack the $k\epsilon$ model clearly shows its problems. It is interesting to notice that from the curves we cannot realize the inaccuracies of $kk_l\omega$ model that also presents more precise values in the leading and in the trailing edge. However lift and drag coefficients are integrated quantities over the airfoil and so give a reliable idea on the accuracy of the model.

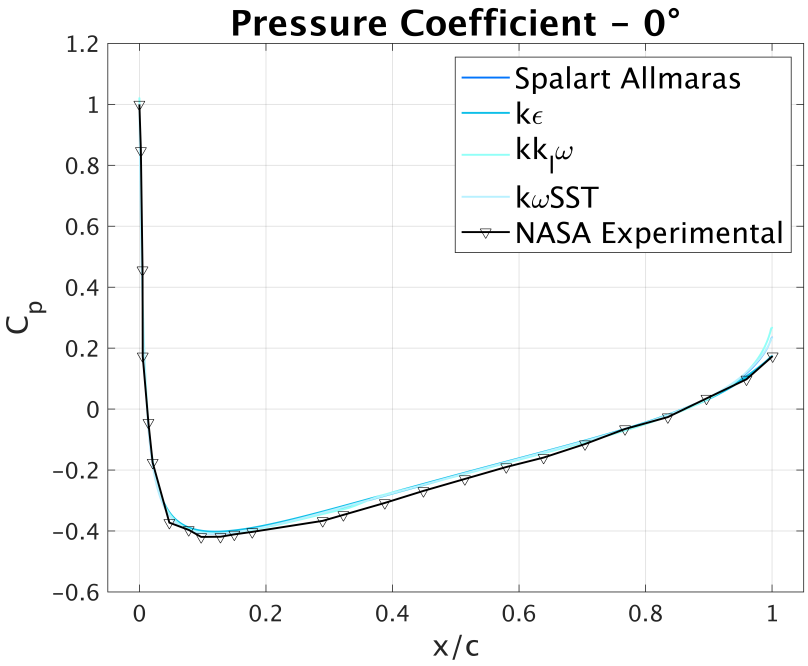


Fig. 4.6 C_p curves - Turbulence Model Comparison - 0° angle of attack

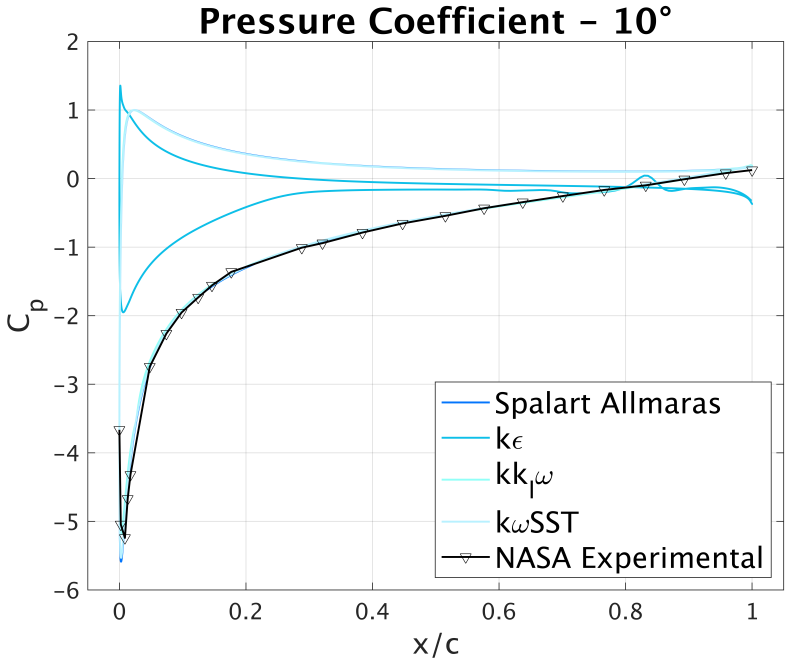


Fig. 4.7 C_p curves - Turbulence Model Comparison - 10° angle of attack

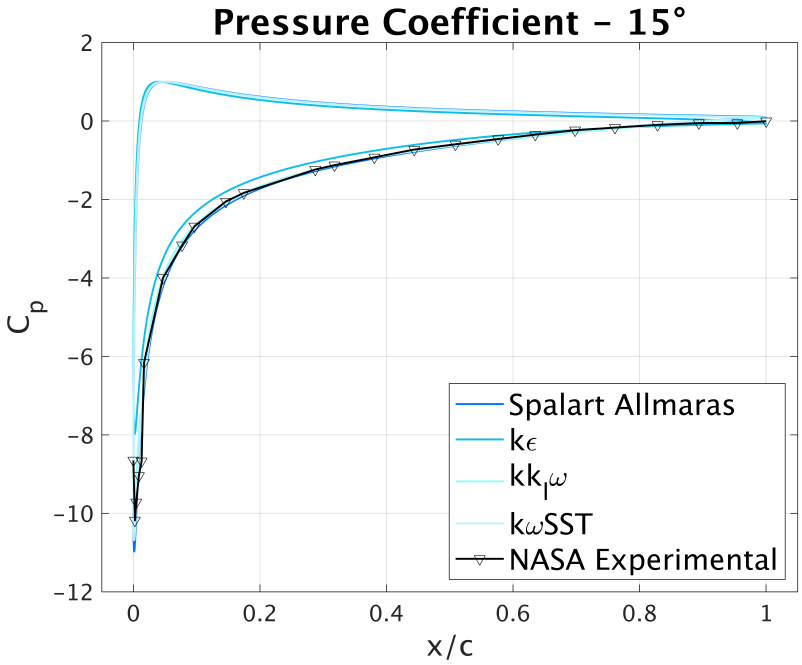


Fig. 4.8 C_p curves - Turbulence Model Comparison - 15° angle of attack

In order to understand which model converges faster we plotted once again the C_l - Iteration C_d - Iteration curves.
By means of the software *Paraview* we are capable of visualizing the results obtained. Velocity, pressure, ν_t and $\tilde{\nu}$ fields for Spalart-Allmaras model are shown for 0°, 10° and 15° angle of attack.

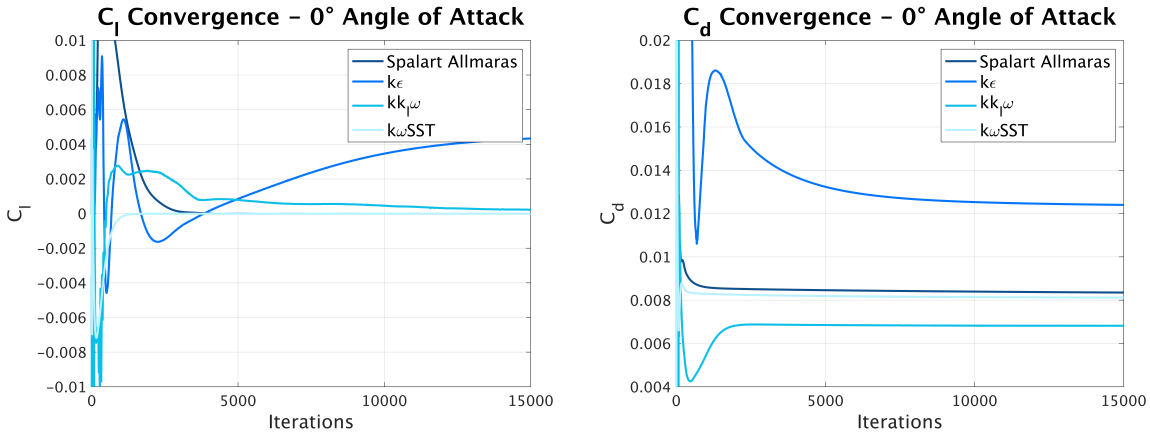


Fig. 4.9 Turbulence Model Validation

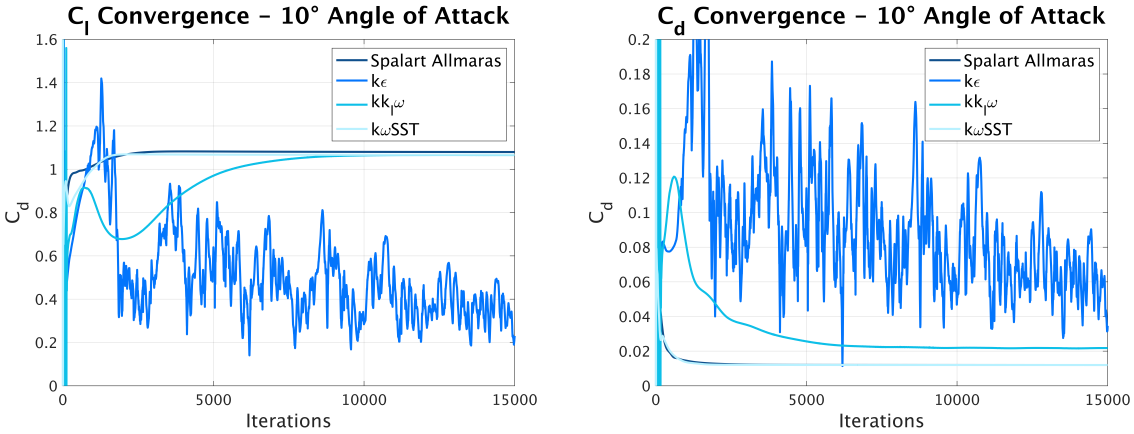


Fig. 4.10 Turbulence Model Validation

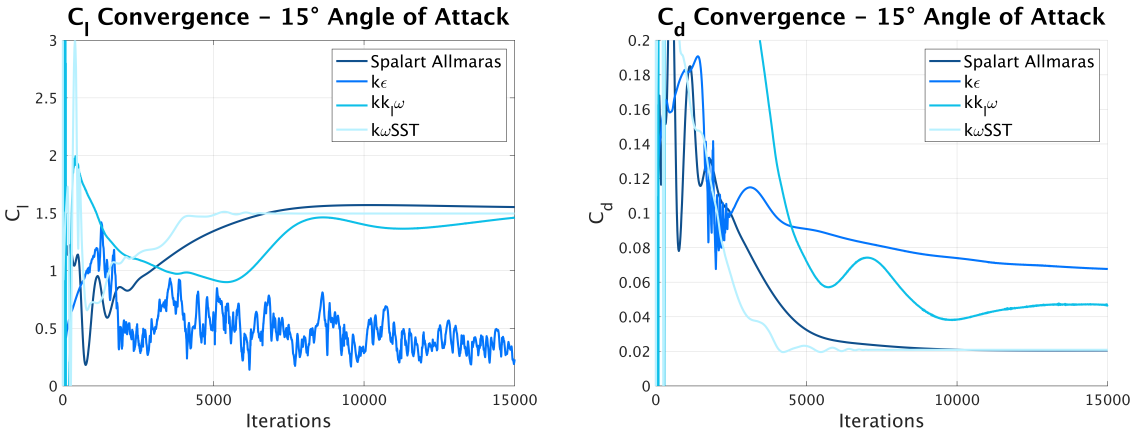


Fig. 4.11 Turbulence Model Validation

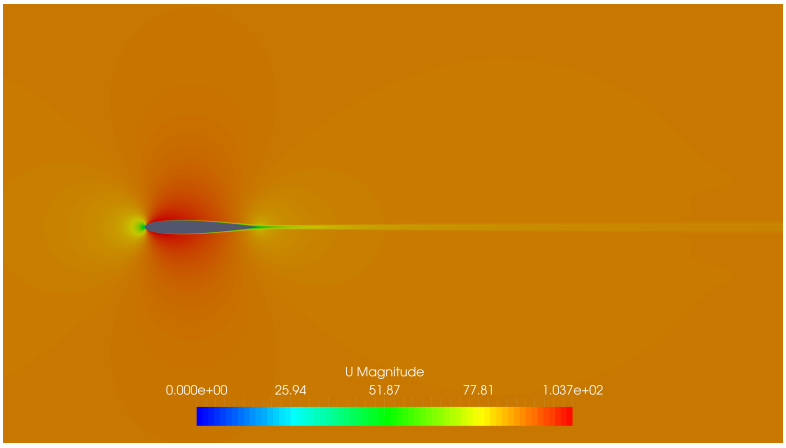


Fig. 4.12 Velocity profile - Spalart-Allmaras - 0° angle of attack

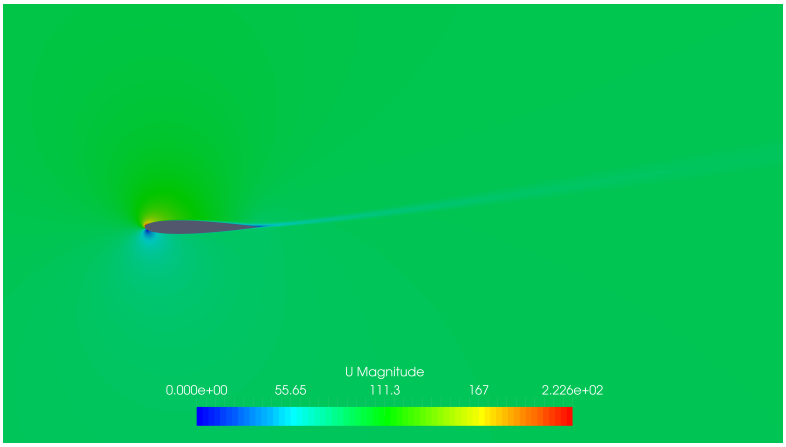


Fig. 4.13 Velocity profile - Spalart-Allmaras - 10° angle of attack

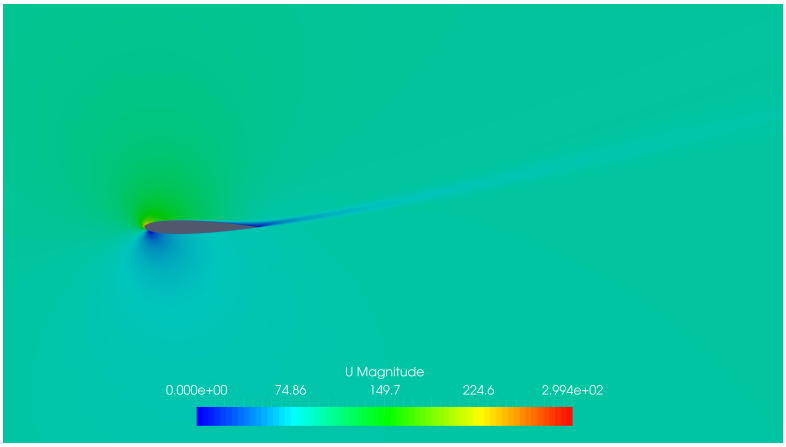


Fig. 4.14 Velocity profile - Spalart-Allmaras - 15° angle of attack

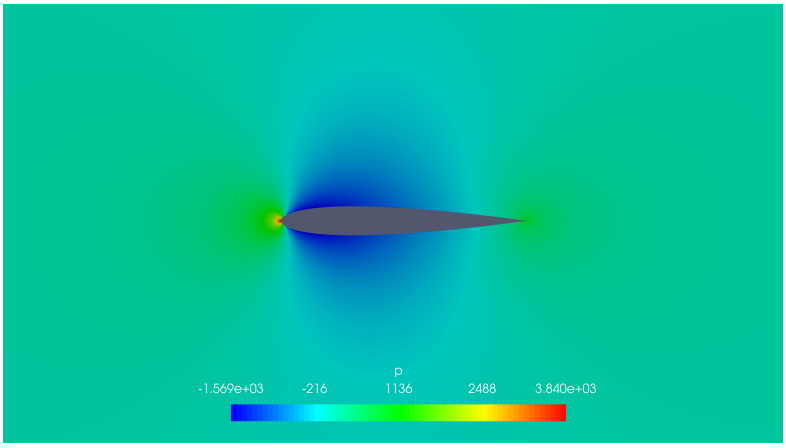


Fig. 4.15 Pressure profile - Spalart-Allmaras - 0° angle of attack

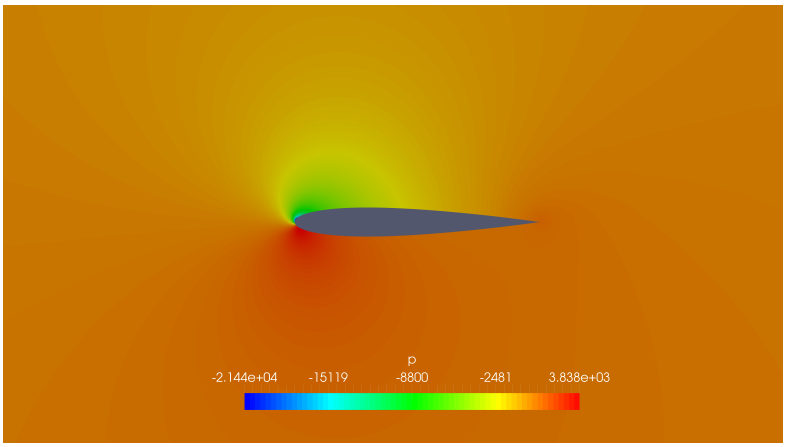


Fig. 4.16 Pressure profile - Spalart-Allmaras - 10° angle of attack

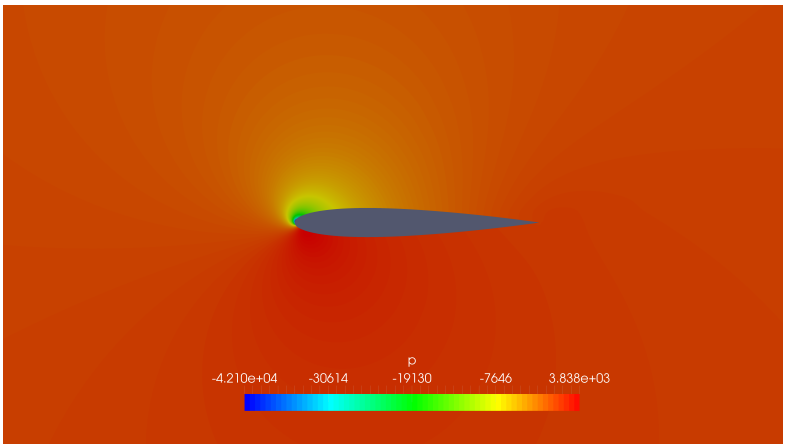


Fig. 4.17 Pressure profile - Spalart-Allmaras - 15° angle of attack

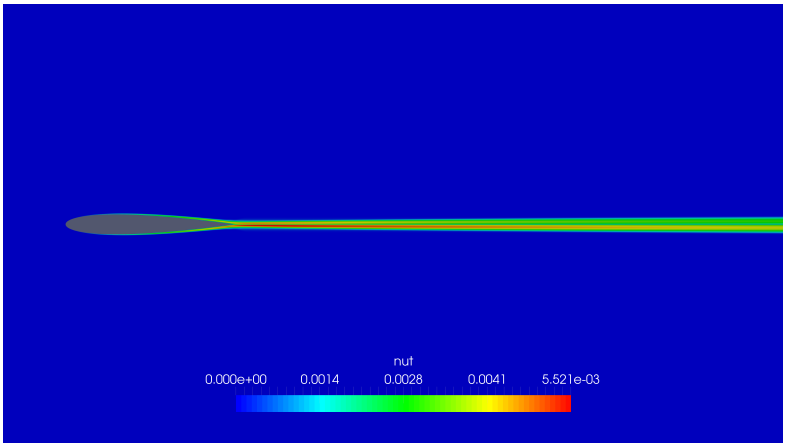


Fig. 4.18 ν_t profile - Spalart-Allmaras - 0° angle of attack

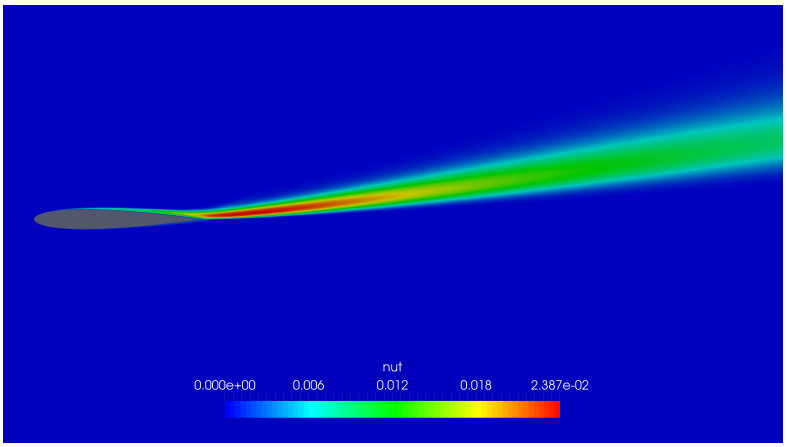


Fig. 4.19 ν_t profile - Spalart-Allmaras - 10° angle of attack

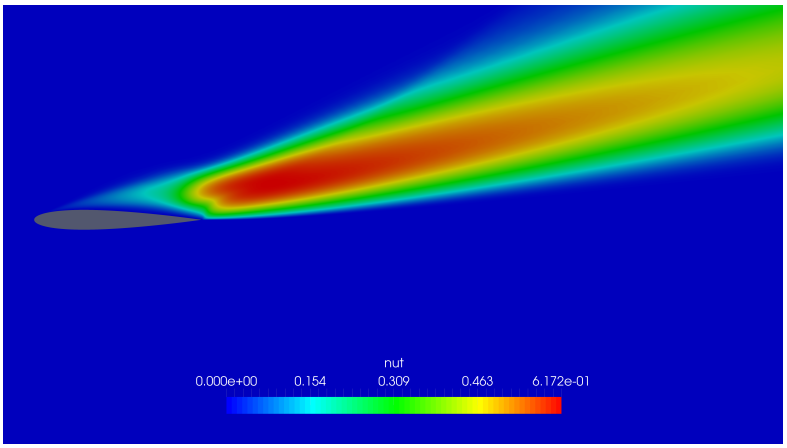


Fig. 4.20 ν_t profile - Spalart-Allmaras - 15° angle of attack

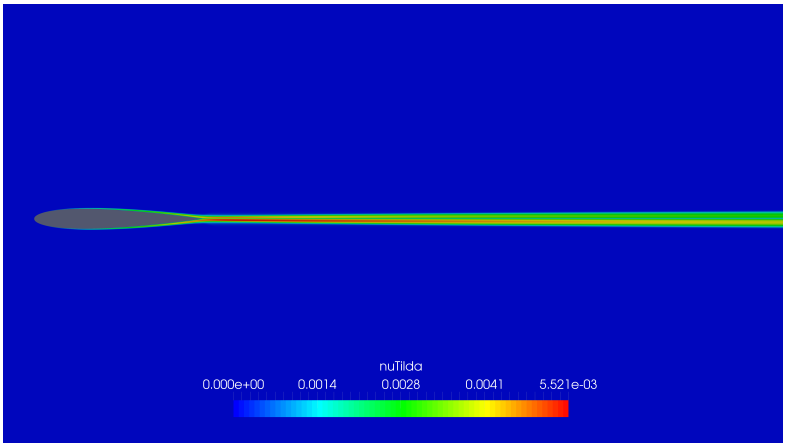


Fig. 4.21 $\tilde{\nu}$ profile - Spalart-Allmaras - 0° angle of attack

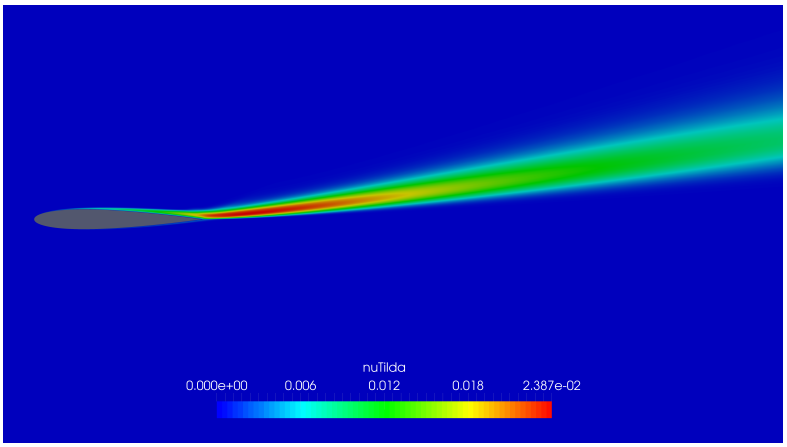


Fig. 4.22 $\tilde{\nu}$ profile - Spalart-Allmaras - 10° angle of attack

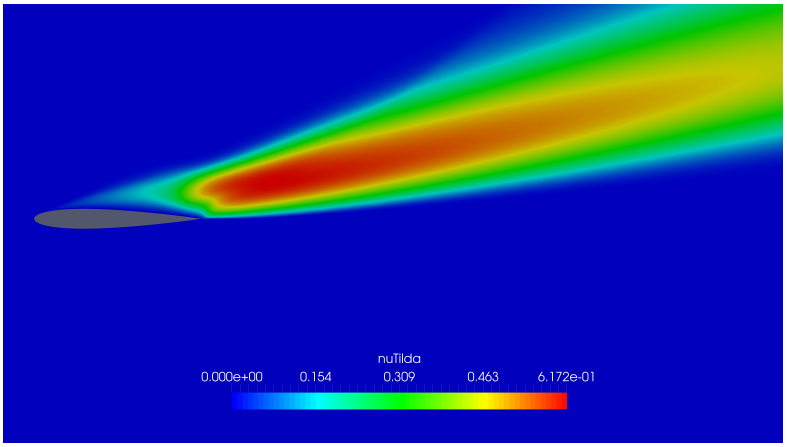


Fig. 4.23 $\tilde{\nu}$ profile - Spalart-Allmaras - 15° angle of attack

4.4 y^+ Values

y^+ values resulting from the simulation done with Spalart-Allmaras model for 0°, 10° and 15° angle of attack. Every simulation done with a NASA mesh, including the coarsest one, shows a y^+ value lower than one, thus there is no need of wall function.

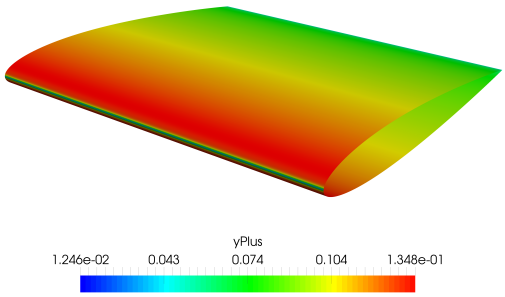


Fig. 4.24 y^+ profile - Spalart-Allmaras - 0° angle of attack

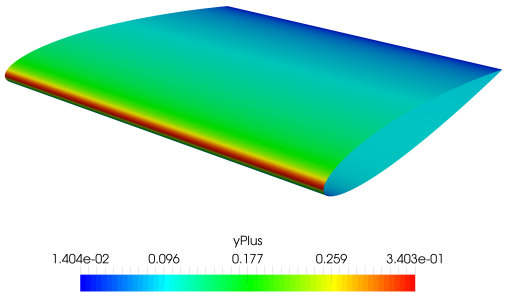


Fig. 4.25 y^+ profile - Spalart-Allmaras - 10° angle of attack

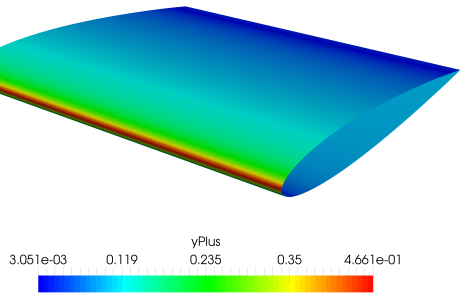


Fig. 4.26 y^+ profile - Spalart-Allmaras - 15° angle of attack

4.5 Comparison Between Two Divergence Discretization Schemes

The purpose of this experiment is to analyse what is the influence of the divergence discretization schemes on the the results.
While we were trying to find what are the best discretization schemes for the turbulence comparison experiment, we noticed that the most influencing ones are the divergence discretization schemes. In particular there is an evident difference in the result using first order and second order schemes for the velocity divergence terms.
This was recognizable while running the Spalart-Allmaras 10° a.o.a. case with the grids created with Pointwise. First we thought that the errors in the results were caused by the mesh but after several attempts with different Pointwise grids we saw that the results were always the same. Changing the divergence discretization scheme of the velocity solved the problem.
The results for both first order (upwind) and second order (linearUpwind) divergence schemes are showed.

0° Angle of Attack				
	Upwind	**Num. Error**	**linearUpwind**	**Num. Error**
C_d	0.0138	-68.45%	0.008970	-9.49%
C_l	-0.0003	approx. 0%	-0.000125	approx. 0%
10° Angle of Attack				
	Upwind	**Num. Error**	**linearUpwind**	**Num. Error**
C_d	0.0426	-244.03%	0.0140	-13.06%
C_l	0.9570	12.54%	1.0662	2.56%

Table 4.9 Discretization Schemes Comparison

In the table is clear the difference between the two methods. The results are not so accurate since we ran the simulations for just 10.000 iterations since the purpose of the experiment was just to see the difference in the results.

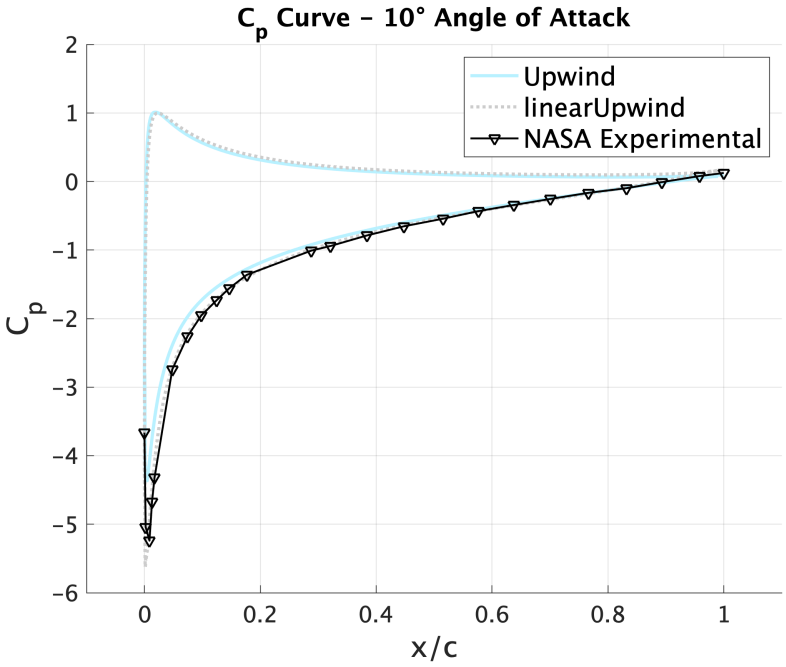


Fig. 4.27 C_p curves - Spalart-Allmaras - 10° angle of attack

We can see evident differences also in C_p curves.

4.6 Comparison Between Meshes with and without Wall Functions

With the purpose of capturing what are the dissimilarities in the results with and without wall functions two C-Grids were created as described in chapter 3. This was done in order to have an idea on what will happen with the wind turbine simulation. Reducing the first layer will reduce the total number of elements and the complexity related with the element extrusion from the surface.
The fine Pointwise mesh assures a $y^+ < 1$, while the coarse is above 30, thus the buffer layer is avoided.
The table 4.7.1 illustrates the average and maximum y^+ value.

Pointwise Mesh	**Angle of Attack**	$\boldsymbol{y^+}$ **Average**	$\boldsymbol{y^+}$ **Max**
Coarse	0°	136.332	171.866
	10°	133.093	346.598
Fine	0°	0.134	0.164
	10°	0.134	0.442

Table 4.10 y^+ - NACA0012 Pointwise Grids

Since we are just interested in the difference between direct solution and wall function one, we ran these simulations for only 10.000 iterations in order to save time. We can do this since the number of elements in the two grids is almost the same.

Pointwise Mesh	**Angle of Attack**	$\boldsymbol{C_l}$	**Difference**	$\boldsymbol{C_d}$	**Difference**
Coarse	0°	0	0.00%	0.00897	1.78%
Fine		0		0.00881	
Coarse	10°	1.0660	0.25%	0.0135	4.44%
Fine		1.0633		0.0129	

Table 4.11 Direct Solution and Wall Functions Results Comparison

Since results are similar to each other it will be worth to try what is the difference also in the wind turbine case. The use of wall function could strongly decrease computational costs.

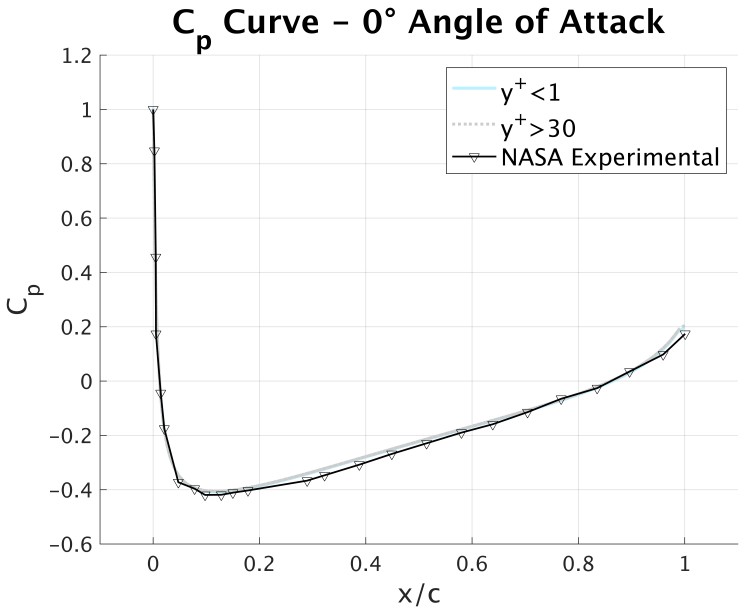


Fig. 4.28 C_p curve - Spalart-Allmaras - 0° angle of attack

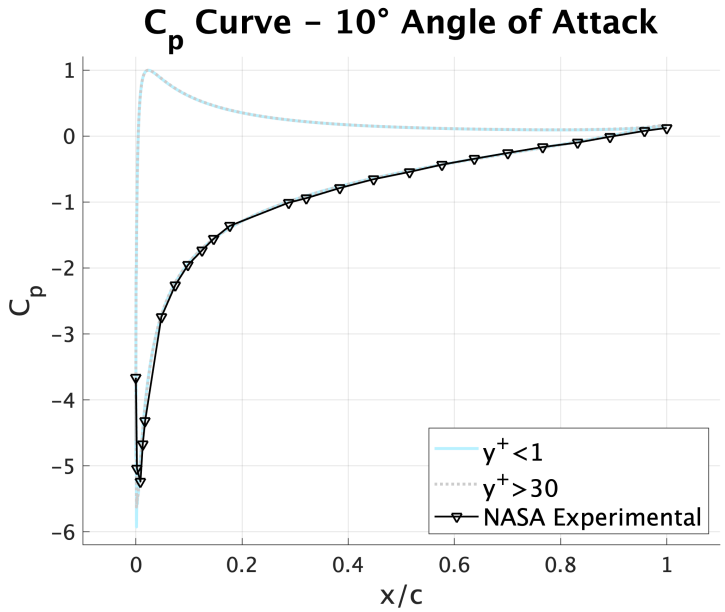


Fig. 4.29 C_p curves - Spalart-Allmaras - 10° angle of attack

4.7 Three Dimensional Simulation

Since turbulence is a three dimensional phenomena it is worth to simulate the same airfoil profile extruded as a wing. The extrusion procedure has been already described in chapter 3.
The c_p curve is a little bit less accurate with respect to the two dimensional case but is almost identical to the experimental one.

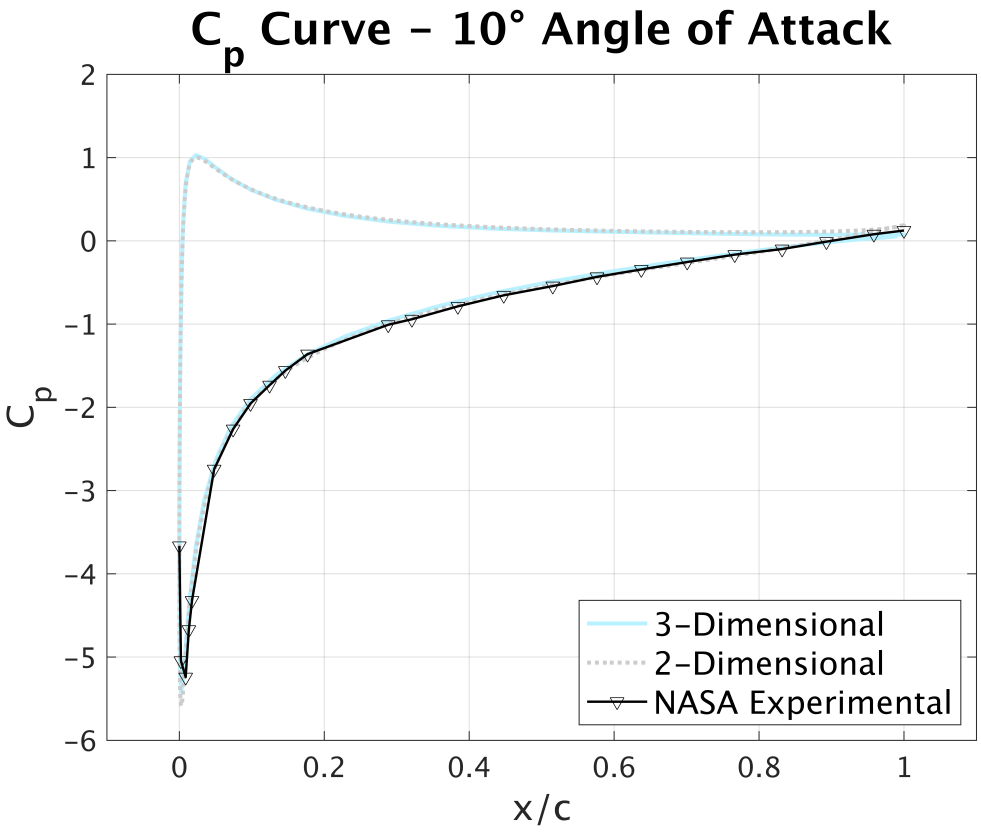


Fig. 4.30 C_p curves - Three Dimensional Simulation - Spalart-Allmaras - 10° angle of attack

Paraview images are reported for an additional comparison.

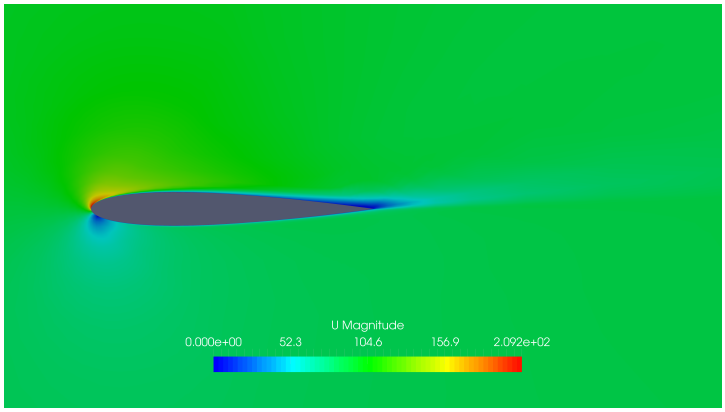


Fig. 4.31 Velocity profile - 3-D Simulation - Spalart-Allmaras - 10° a.o.a.

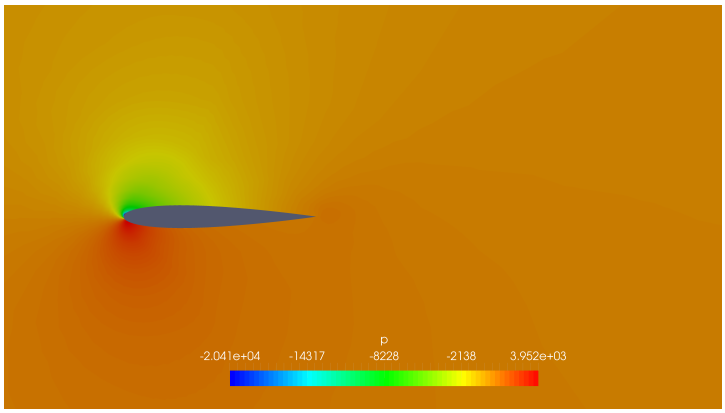


Fig. 4.32 Pressure profile - 3-D Simulation - Spalart-Allmaras - 10° a.o.a.

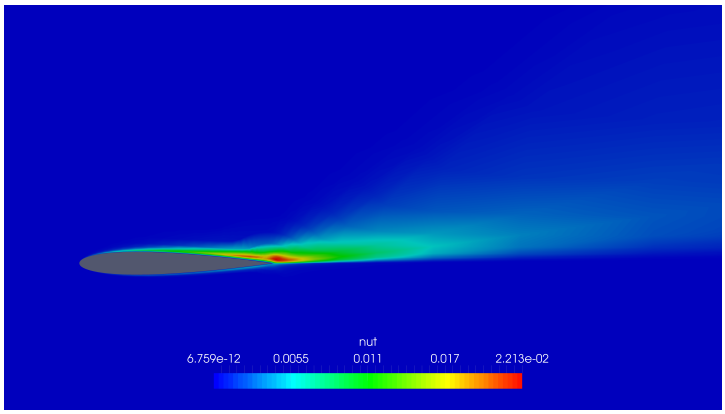


Fig. 4.33 ν_t profile - 3-D Simulation - Spalart-Allmaras - 10° a.o.a.

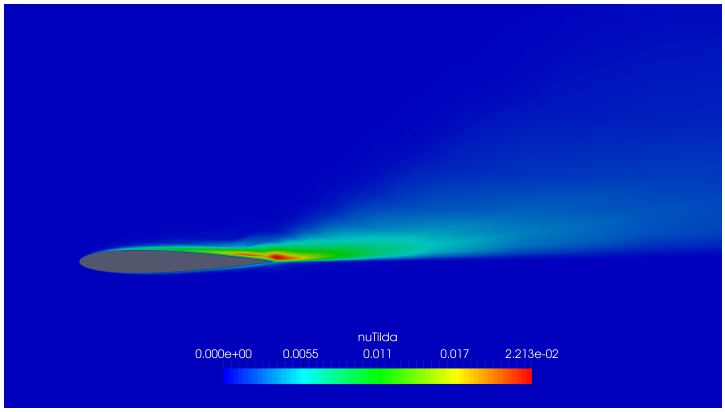


Fig. 4.34 $\tilde{\nu}$ profile - 3-D Simulation - Spalart-Allmaras - 10° a.o.a.

4.8 Transient Simulation

The last numerical experiment on the airfoil concerns with the comparison between steady state and transient simulation. The case run is the 10° a.o.a. with the second finest NASA LRC mesh (897x257). Only Spalart-Allmaras and $k\omega SST$ are used since they are the only two that work in steady state. Boundary conditions are maintained the same as in the steady state, *backward* scheme is used for time discretization and PIMPLE algorithm is used.

Transient Results

Turbulence	C_l			C_d		
Model	**Steady**	**Transient**	**Difference**	**Steady**	**Transient**	**Difference**
Spalart Allmaras	1.0750	1.0790	-0.37%	0.01246	0.0122	2.09%
$k\omega SST$	1.0666	1.0668	-0.02%	0.01190	0.0121	-1.68%

Table 4.12 Transient-Steady State Result Comparison

Results are very similar to steady-state ones and very similar to the experimental ones. Also two videos were extrapolated with paraview with the purpose of capturing what is the evolution of the flow until the convergence is reached. From the latter we can observe that $k\omega SST$ model can capture fluid detachment during the convergence phase while Splart-Allmaras does not.

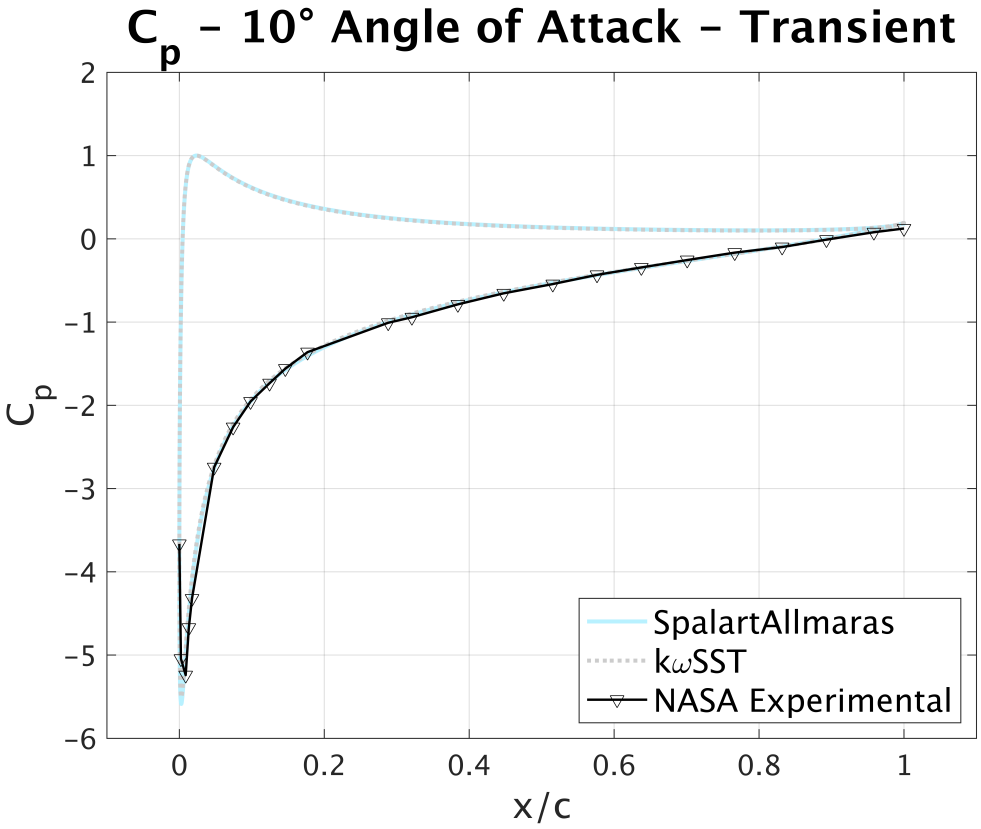


Fig. 4.35 C_p curves - Transient Simulation - 10° angle of attack

Chapter 5

DTU 10MW Results

Chapter five presents simulations executed on the DTU 10MW and the relative results. The simulations were executed using the four mesh presented in chapter 3. Both Spalart-Allmaras and $k\omega SST$ turbulence models were used. First, steady state simulations were executed and then they were used as initial values for transient simulations. Every simulations was run on the TUM Statik Department cluster, employing a maximum of 30 processors (when available) and using Scotch parallel decomposition algorithm. Both steady state and transient simulations were initially solved employing a bounded upwind scheme for the velocity divergence that was changed to a bounded linear upwind scheme when a stable solution was reached. Starting directly with the latter caused divergence of the residuals. Every other divergence term was discretized with a bounded upwind scheme. A linear scheme was employed for the gradient terms and laplacian terms. The latter were corrected for non-orthogonality with the already mentioned iterative procedure. A multi grid geometric solver was used for the pressure terms with a Gauss-Seidel smoother, while a preconditioned stabilized bi-conjugate gradient solver was used for every other variable. Simple consistent algorithm with high relaxation factors (0.7 for pressure, 0.9 for every other variable) were adopted for steady state simulations, while PIMPLE consistent algorithm was employed for transient simulation with a backward scheme for the temporal terms.
The simulations were run only for the speed ranges present in the table below that have zero pitch, this simplified the aforementioned meshing procedure.

Rotational Speed [RPM]	Wind speed [m/s]
6	7
6.426	8
7.229	9
8.032	10
8.836	11

Table 5.1 Velocities used for the simulations

Results are compared with Denmark Technical University [11], that has results both from CFD and BEM, with University of Stuttgart [47] and with Universtè de Mons [25].
In the table below the software and the turbulence model used by each university are showed.

University	Turbulence Model	Solver
Technical University of Denmark	$k\omega SST$	$EllipSys3D$
University of Stuttgart	$k\omega SST$	$FLOWer$
Universitè de Mons	$Spalart - Allmaras$	$FINE^{TM}/Turbo$

The two turbulence models used are $k\omega SST$ and Spalart-Allmaras, confirming that the conclusion made for the NACA0012 airfoil are consistent. Every simulation has been run with a density of $\rho = 1.225 kg/m^3$ and a dynamic viscosity of $\mu = 1.78406 \cdot 10^{-5} kg/(m \cdot s)$.

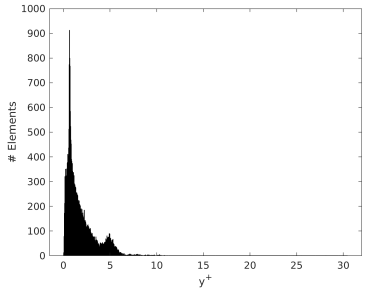
5.1 Spalart-Allmaras Steady State Results

In the finest mesh the y^+ values are, for a high percentage ($> 94\%$), lower than 1 so a direct solution without wall functions can be used. The latter fact is clarified by the images that show the y^+ distribution for the nominal wind speed (11 m/s) and by the table showing the percentage of elements in the viscous, buffer and inertial layers for each speed. This was done for every mesh in order to investigate the influence of y^+. For the remaining meshes a wall

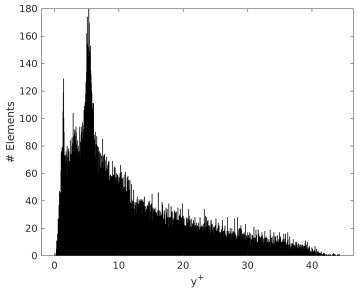
function has to be employed. The table below shows the boundary conditions used for each quantity in each patch of the mesh.

Spalart-Allmaras						
Variables		**Inlet**	**Outlet**	**Farfield**	**Blade**	**Cyclic1 Cyclic2**
p	Type	zeroGradient	fixedValue	zeroGradient	zeroGradient	cyclicAMI
	Value	/	Uniform 0	/	/	Uniform 0
U	Type	fixedValue	zeroGradient	zeroGradient	fixedValue	cyclicAMI
	Value	*inletSpeed*	/	/	Uniform 0	Uniform 0
nut	Type	fixedValue	zeroGradient	zeroGradient	nutUSpaldingWallFunction	cyclicAMI
	Value	Uniform 1.456e-5	/	/	Uniform 0	Uniform 0
nuTilda	Type	fixedValue	zeroGradient	zeroGradient	fixedValue	cyclicAMI
	Value	Uniform 4.37e-5	/	/	Uniform 0	Uniform 0

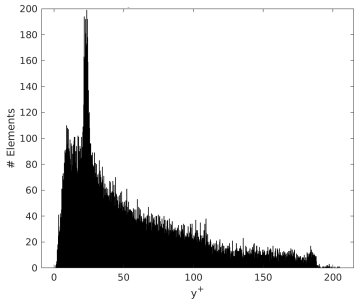
Table 5.2 Spalart-Allmaras Steady State Boundary Conditions



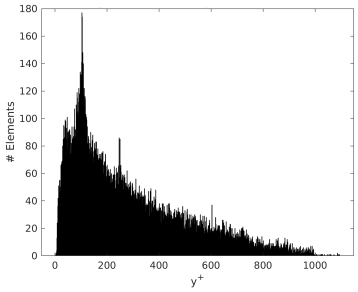
(a) Mesh 3e-5



(b) Mesh 3e-4



(c) Mesh 1e-3



(d) Mesh 5e-3

Fig. 5.1 y^+ Distribution

The pictures depict the y^+ element distribution computed for the nominal wind speed, for the other speeds only percentage values are given. It is evident that the finest mesh has the majority of elements in the viscous layer while coarsest mesh has the majority of elements in the inertial layer. The two meshes in between have a considerable percentage of elements in the buffer zone, thus it is interesting to monitor the effects caused by the latter.
The next two images make a visual comparison between the power obtained with every mesh and the results from the already mentioned universities.

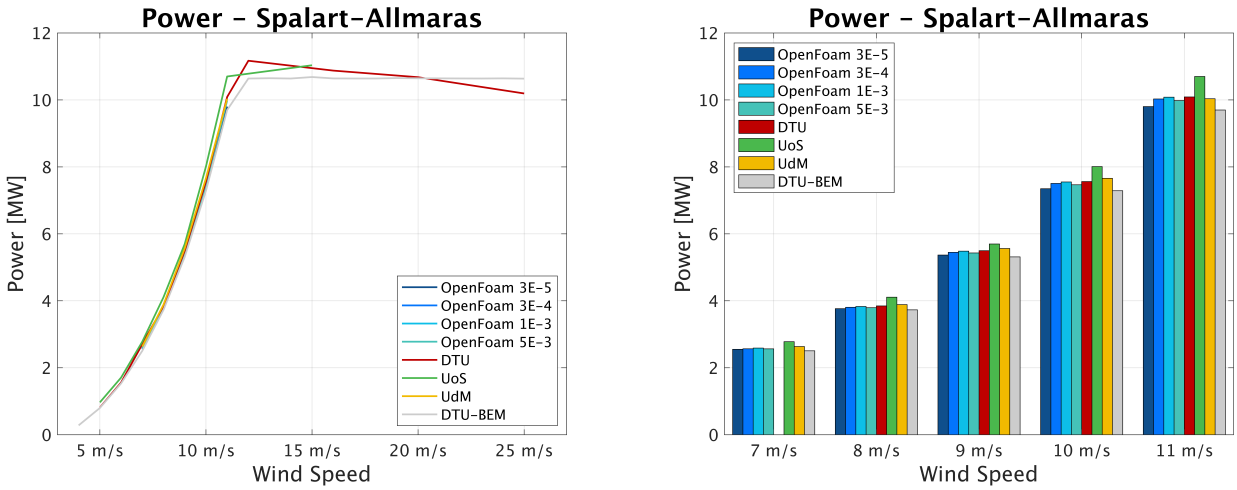


Fig. 5.2 Power Comparison - Spalart-Allmaras

Except from the results obtained from the University of Stuttgart and from the DTU-BEM, all the curves almost overlap with each other. It is interesting to notice that every university obtained similar results but still have noticeable differences. This can happen in computational fluid dynamics when we deal with sharp geometries such as this one.

Regarding the thrust, the results are not as similar as the power one. The errors with respect to DTU, that maybe considered as the most reliable one since they fully designed the turbine, are generally higher. Also the values obtained by the other universities are generally more different one from each other.
The two graphs below give an idea of the different values obtained:

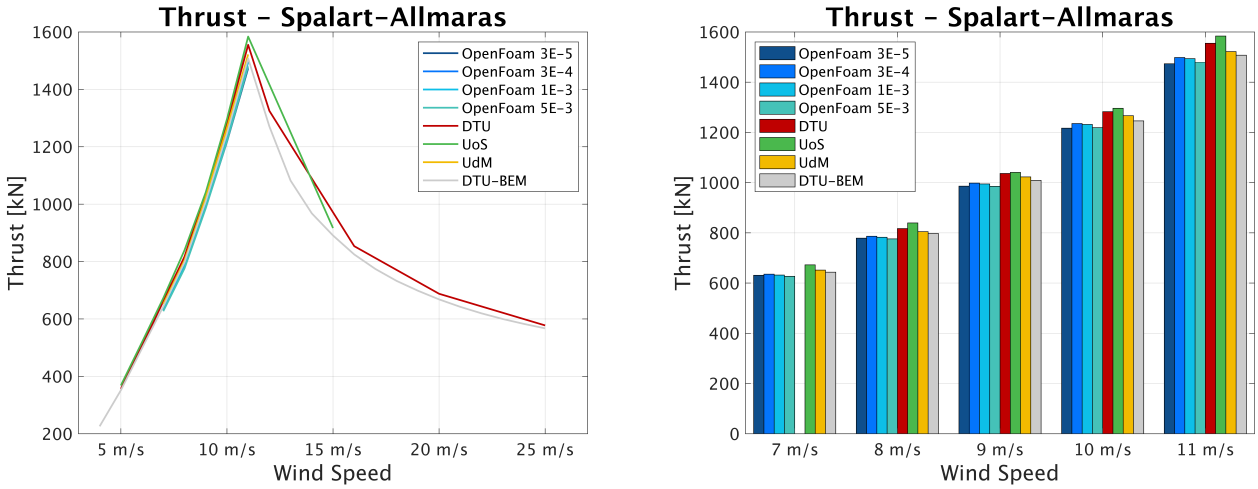


Fig. 5.3 Thrust Comparison - Spalart-Allmaras

In the following subsections three tables per mesh show the numerical results obtained for y^+, mechanical power and thrust. Every result is compared with the aforementioned universities and the difference in percentage is computed with the formula:

$$\frac{OpenFoam - ReferenceResult}{OpenFoam}\% \tag{5.1}$$

5.1.1 Mesh 3e-5

y^+				
Rotational Speed	**Wind speed [m/s]**	**Viscous**	**Buffer**	**Inertial**
6	7	99.81%	0.20%	0.00%
6.426	8	99.33%	0.68%	0.00%
7.229	9	99.12%	0.89%	0.00%
8.032	10	96.98%	3.03%	0.00%
8.836	11	94.13%	5.88%	0.01%

Table 5.3 y^+ - Spalart-Allmaras - Mesh 3e-5

Power						
Rotational Speed	**Wind speed [m/s]**	**Power[MW]**	**DTU**	**UoS**	**UdM**	**BEM**
6	7	2.552	/	8.20%	3.16%	-1.83%
6.426	8	3.766	2.14%	8.34%	3.09%	-0.94%
7.229	9	5.367	2.35%	5.78%	3.50%	-1.05%
8.032	10	7.348	2.82%	8.26%	4.08%	-0.84%
8.836	11	9.801	2.85%	8.40%	2.34%	-1.06%

Table 5.4 Power Results - Spalart-Allmaras - Mesh 3e-5

Thrust						
Rotational Speed	**Wind speed [m/s]**	**Thrust[kN]**	**DTU**	**UoS**	**UdM**	**BEM**
6	7	630.94	/	6.19%	3.19%	1.94%
6.426	8	779.01	4.65%	7.21%	3.18%	2.29%
7.229	9	986.03	4.88%	5.27%	3.60%	2.29%
8.032	10	1216.89	5.12%	6.10%	3.91%	2.32%
8.836	11	1473.38	5.25%	6.97%	3.18%	2.26%

Table 5.5 Thrust - Spalart-Allmaras - Mesh 3e-5

5.1.2 Mesh 3e-4

y^+				
Rotational Speed	**Wind speed [m/s]**	**Viscous**	**Buffer**	**Inertial**
6	7	25.64%	67.31%	7.06%
6.426	8	25.78%	67.25%	6.98%
7.229	9	22.44%	68.08%	9.50%
8.032	10	18.78%	68.57%	12.65%
8.836	11	16.17%	67.87%	15.97%

Table 5.6 y^+ - Spalart-Allmaras - Mesh 3e-4

Power						
Rotational Speed	**Wind speed [m/s]**	**Power[MW]**	**DTU**	**UoS**	**UdM**	**BEM**
6	7	2.57	/	7.64%	2.56%	-2.45%
6.426	8	3.81	1.05%	7.32%	2.01%	-2.07%
7.229	9	5.45	0.87%	4.35%	2.04%	-2.58%
8.032	10	7.51	0.72%	6.27%	2.00%	-3.02%
8.836	11	10.03	0.59%	6.27%	0.07%	-3.41%

Table 5.7 Power Results - Spalart-Allmaras - Mesh 3e-4

Thrust						
Rotational Speed	**Wind speed [m/s]**	**Thrust[kN]**	**DTU**	**UoS**	**UdM**	**BEM**
6	7	635.55	/	5.51%	2.48%	1.22%
6.426	8	786.55	3.73%	6.31%	2.24%	1.35%
7.229	9	998.18	3.71%	4.10%	2.42%	1.08%
8.032	10	1235.36	3.68%	4.68%	2.45%	0.84%
8.836	11	1498.22	3.65%	5.40%	1.55%	0.61%

Table 5.8 Thrust - Spalart-Allmaras - Mesh 3e-4

5.1.3 Mesh 1e-3

y^+				
Rotational Speed	**Wind speed [m/s]**	**Viscous**	**Buffer**	**Inertial**
6	7	3.33%	48.25%	48.25%
6.426	8	2.50%	46.93%	50.58%
7.229	9	1.75%	43.83%	54.43%
8.032	10	1.55%	40.81%	57.65%
8.836	11	1.21%	38.18%	60.61%

Table 5.9 y^+ - Spalart-Allmaras - Mesh 1e-3

Power						
Rotational Speed	**Wind speed [m/s]**	**Power[MW]**	**DTU**	**UoS**	**UdM**	**BEM**
6	7	2.59	/	6.89%	1.77%	-3.28%
6.426	8	3.83	0.44%	6.75%	1.41%	-2.69%
7.229	9	5.48	0.25%	3.76%	1.43%	-3.22%
8.032	10	7.55	0.14%	5.73%	1.43%	-3.62%
8.836	11	10.08	0.06%	5.77%	-0.46%	-3.96%

Table 5.10 Power Results - Spalart-Allmaras - Mesh 1e-3

Thrust						
Rotational Speed	**Wind speed [m/s]**	**Thrust[kN]**	**DTU**	**UoS**	**UdM**	**BEM**
6	7	631.87	/	6.06%	3.04%	1.79%
6.426	8	782.50	4.22%	6.80%	2.75%	1.86%
7.229	9	994.73	4.04%	4.43%	2.75%	1.42%
8.032	10	1231.59	3.98%	4.97%	2.75%	1.14%
8.836	11	1493.82	3.93%	5.68%	1.84%	0.90%

Table 5.11 Thrust - Spalart-Allmaras - Mesh 1e-3

5.1.4 Mesh 5e-3

y^+				
Rotational Speed	**Wind speed [m/s]**	**Viscous**	**Buffer**	**Inertial**
6	7	0.061%	6.37%	93.58%
6.426	8	0.003%	5.05%	94.95%
7.229	9	0.003%	4.68%	95.32%
8.032	10	0.013%	3.09%	96.91%
8.836	11	0.006%	3.24%	96.76%

Table 5.12 y^+ - Spalart-Allmaras - Mesh 5e-3

Power						
Rotational Speed	**Wind speed [m/s]**	**Power[MW]**	**DTU**	**UoS**	**UdM**	**BEM**
6	7	2.57	/	7.71%	2.63%	-2.37%
6.426	8	3.79	1.40%	7.65%	2.36%	-1.71%
7.229	9	5.43	1.22%	4.69%	2.38%	-2.22%
8.032	10	7.47	1.25%	6.77%	2.53%	-2.47%
8.836	11	9.98	1.08%	6.73%	0.56%	-2.90%

Table 5.13 Power Results - Spalart-Allmaras - Mesh 5e-3

Thrust						
Rotational Speed	**Wind speed [m/s]**	**Thrust[kN]**	**DTU**	**UoS**	**UdM**	**BEM**
6	7	626.94	/	6.79%	3.80%	2.56%
6.426	8	776.01	5.02%	7.57%	3.55%	2.67%
7.229	9	985.38	4.94%	5.33%	3.67%	2.35%
8.032	10	1218.99	4.96%	5.94%	3.74%	2.15%
8.836	11	1480.02	4.82%	6.55%	2.75%	1.82%

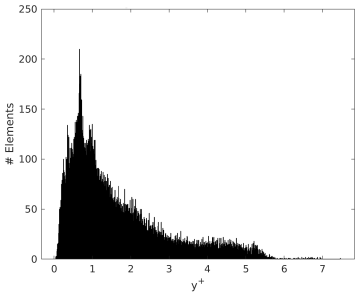
Table 5.14 Thrust - Spalart-Allmaras - Mesh 5e-3

5.2 $k\omega SST$ Steady State Results

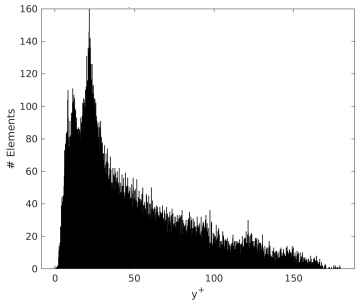
The same grids were then used for simulations in which $k\omega SST$ model was employed. We present here what are the boundary conditions that were imposed for this kind of turbulence model are.

kOmegaSST						
Variables		**Inlet**	**Outlet**	**Farfield**	**Blade**	**Cyclic1 Cyclic2**
p	Type	zeroGradient	fixedValue	zeroGradient	zeroGradient	cyclicAMI
	Value	/	Uniform 0	/	/	Uniform 0
U	Type	fixedValue	zeroGradient	zeroGradient	fixedValue	cyclicAMI
	Value	inletSpeed	/	/	Uniform 0	Uniform 0
nut	Type	fixedValue	zeroGradient	zeroGradient	nutUSpaldingWallFunction	cyclicAMI
	Value	1.4563	/	/	Uniform 0	Uniform 0
k	Type	fixedValue	zeroGradient	zeroGradient	kqRWallFunction	cyclicAMI
	Value	0.0074	/	/	Uniform 0	Uniform 0
Omega	Type	fixedValue	zeroGradient	zeroGradient	omegaWallFunction	cyclicAMI
	Value	1.5876	/	/	Uniform 0	Uniform 0

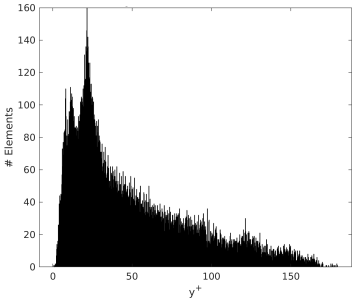
Table 5.15 $k\omega SST$ Steady State Boundary Conditions



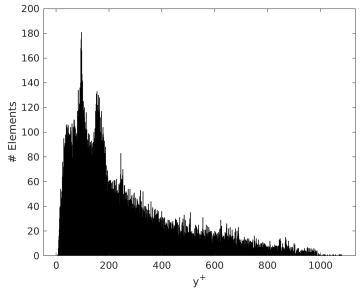
(a) Mesh 3e-5



(b) Mesh 3e-4



(c) Mesh 1e-3



(d) Mesh 5e-3

Fig. 5.4 y^+ Distribution

Also in this case we present the y^+ distribution for every mesh run at wind nominal speed. Once again the finest mesh presents y^+ values that are below one, thus a direct solution is sufficient.

As for the Spalart-Allmaras model the prediction of the power shows good results. In the graph below every line, with the exception of U.o.S. and D.T.U-BEM, is overlapped.

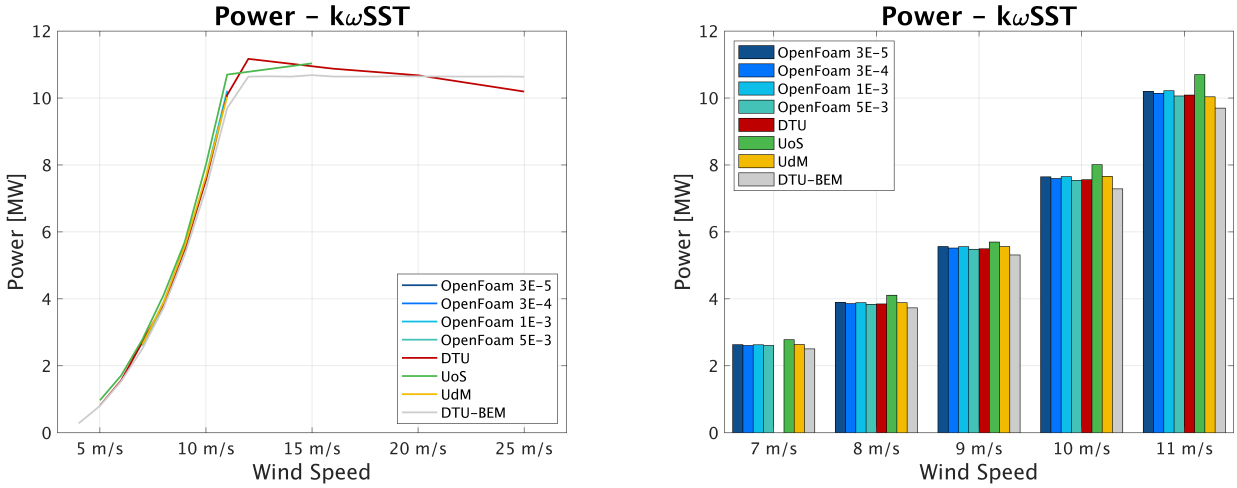


Fig. 5.5 Power Comparison - $k\omega SST$

The thrust results are still less accurate with respect to the power one, but seem to be still acceptable since we can assist to a noticeable difference in the results obtained by different universities. We can see for instance that the values are very similar from the ones obtained by U.d.M.

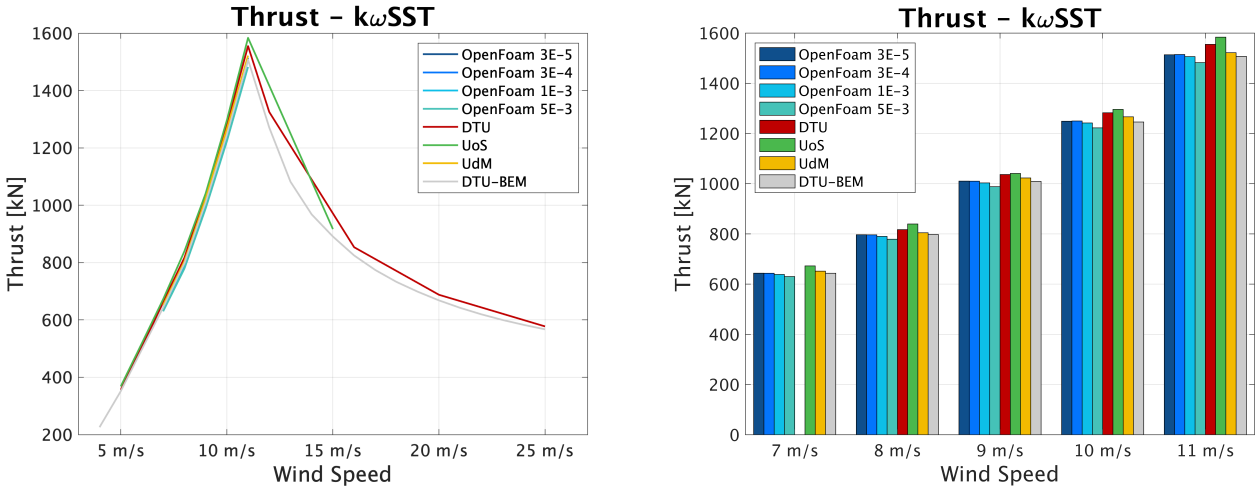


Fig. 5.6 Thrust Comparison - $k\omega SST$

In the next subsections we present three tables per mesh describing numerical results of y^+, mechanical power and thrust. The difference in percentage is still computed with the formula used for Spalart-Allmaras results.

5.2.1 Mesh 3e-5

y^+				
Rotational Speed	**Wind speed [m/s]**	**Viscous**	**Buffer**	**Inertial**
6	7	100.00%	0.00%	0.00%
6.426	8	99.92%	0.08%	0.003%
7.229	9	99.85%	0.15%	0.003%
8.032	10	99.26%	0.75%	0.003%
8.836	11	97.89%	2.11%	0.003%

Table 5.16 y^+ - $k\omega SST$ - Mesh 3e-5

Power						
Rotational Speed	**Wind speed [m/s]**	**Power[MW]**	**DTU**	**UoS**	**UdM**	**BEM**
6	7	2.63	/	5.33%	0.12%	-4.78%
6.426	8	3.90	-1.25%	5.17%	-0.26%	-4.25%
7.229	9	5.56	-1.22%	2.33%	-0.03%	-4.53%
8.032	10	7.65	-1.12%	4.54%	0.19%	-4.69%
8.836	11	10.20	-1.10%	4.68%	-1.63%	-4.91%

Table 5.17 Power Results - $k\omega SST$ - Mesh 3e-5

Thrust						
Rotational Speed	**Wind speed [m/s]**	**Thrust[kN]**	**DTU**	**UoS**	**UdM**	**BEM**
6	7	643.65	/	4.30%	1.24%	-0.04%
6.426	8	796.68	2.49%	5.11%	0.98%	0.08%
7.229	9	1010.22	2.54%	2.94%	1.24%	-0.11%
8.032	10	1248.87	2.63%	3.64%	1.38%	-0.25%
8.836	11	1513.41	2.67%	4.44%	0.55%	-0.40%

Table 5.18 Thrust - $k\omega SST$ - Mesh 3e-5

5.2.2 Mesh 3e-4

y^+				
Rotational Speed	**Wind speed [m/s]**	**Viscous**	**Buffer**	**Inertial**
6	7	28.08%	66.63%	5.29%
6.426	8	23.61%	68.24%	8.16%
7.229	9	19.64%	69.12%	11.24%
8.032	10	16.88%	69.01%	14.13%
8.836	11	14.78%	68.41%	16.83%

Table 5.19 y^+ - $k\omega SST$ - Mesh 3e-4

Power						
Rotational Speed	**Wind speed [m/s]**	**Power[MW]**	**DTU**	**UoS**	**UdM**	**BEM**
6	7	2.61	/	6.23%	1.07%	-4.01%
6.426	8	3.86	-0.29%	6.06%	0.68%	-3.45%
7.229	9	5.52	-0.38%	3.14%	0.80%	-3.88%
8.032	10	7.60	-0.47%	5.16%	0.83%	-4.25%
8.836	11	10.14	-0.51%	5.23%	-1.04%	-4.56%

Table 5.20 Power Results - $k\omega SST$ - Mesh 3e-4

Thrust						
Rotational Speed	**Wind speed [m/s]**	**Thrust[kN]**	**DTU**	**UoS**	**UdM**	**BEM**
6	7	643.44	/	4.33%	1.27%	-0.01%
6.426	8	796.23	2.54%	5.16%	1.04%	0.13%
7.229	9	1010.13	2.55%	2.95%	1.25%	-0.10%
8.032	10	1249.74	2.56%	3.57%	1.32%	-0.32%
8.836	11	1514.55	2.60%	4.37%	0.48%	-0.47%

Table 5.21 Thrust - $k\omega SST$ - Mesh 3e-4

5.2.3 Mesh 1e-3

y^+				
Rotational Speed	**Wind speed [m/s]**	**Viscous**	**Buffer**	**Inertial**
6	7	3.33%	48.25%	48.25%
6.426	8	2.50%	46.93%	50.58%
7.229	9	1.75%	43.83%	54.43%
8.032	10	1.55%	40.81%	57.65%
8.836	11	1.21%	38.18%	60.61%

Table 5.22 y^+ - $k\omega SST$ - Mesh 1e-3

Power						
Rotational Speed	**Wind speed [m/s]**	**Power[MW]**	**DTU**	**UoS**	**UdM**	**BEM**
6	7	2.63	/	5.42%	0.22%	-4.91%
6.426	8	3.89	-0.99%	5.41%	-0.01%	-4.17%
7.229	9	5.56	-1.12%	2.43%	0.07%	-4.64%
8.032	10	7.65	-1.21%	4.45%	0.09%	-5.03%
8.836	11	10.22	-1.28%	4.50%	-1.81%	-5.36%

Table 5.23 Power Results - $k\omega SST$ - Mesh 1e-3

Thrust						
Rotational Speed	**Wind speed [m/s]**	**Thrust[kN]**	**DTU**	**UoS**	**UdM**	**BEM**
6	7	638.31	/	5.10%	2.05%	0.79%
6.426	8	790.08	3.29%	5.89%	1.80%	0.91%
7.229	9	1003.17	3.22%	3.62%	1.93%	0.59%
8.032	10	1241.85	3.18%	4.18%	1.94%	0.32%
8.836	11	1506.3	3.13%	4.89%	1.02%	0.07%

Table 5.24 Thrust - $k\omega SST$ - Mesh 1e-3

5.2.4 Mesh 5e-3

y^+				
Rotational Speed	**Wind speed [m/s]**	**Viscous**	**Buffer**	**Inertial**
6	7	0.0385%	6.80%	93.17%
6.426	8	0.0128%	5.61%	94.39%
7.229	9	0.0096%	4.43%	95.57%
8.032	10	0.0032%	3.74%	96.26%
8.836	11	0.0032%	3.06%	96.95%

Table 5.25 y^+ - $k\omega SST$ - Mesh 5e-3

Power						
Rotational Speed	**Wind speed [m/s]**	**Power[MW]**	**DTU**	**UoS**	**UdM**	**BEM**
6	7	2.61	/	6.23%	1.07%	-4.02%
6.426	8	3.84	0.31%	6.63%	1.28%	-2.83%
7.229	9	5.48	0.28%	3.79%	1.46%	-3.18%
8.032	10	7.65	0.27%	5.85%	1.56%	-3.49%
8.836	11	10.22	0.26%	5.96%	-0.26%	-3.75%

Table 5.26 Power Results - $k\omega SST$ - Mesh 5e-3

Thrust						
Rotational Speed	**Wind speed [m/s]**	**Thrust[kN]**	**DTU**	**UoS**	**UdM**	**BEM**
6	7	630.03	/	6.33%	3.33%	2.08%
6.426	8	778.83	4.67%	7.23%	3.20%	2.32%
7.229	9	988.05	4.68%	5.07%	3.41%	2.09%
8.032	10	1222.44	4.69%	5.68%	3.47%	1.88%
8.836	11	1482.27	4.68%	6.41%	2.60%	1.67%

Table 5.27 Thrust - $k\omega SST$ - Mesh 5e-3

The results obtained shows that the $k\omega SST$ performs better than Spalart-Allamaras model. Using DTU results as reference, obtained with $k\omega SST$, the errors are always lower in thrust and in power except for the grid 1e-3 in which SA is more accurate only in power. The errors are lower even if we compare our results with the one gathered by the Universitè de Mons that employed Spalart-Allamaras model.
The buffer layer does not seem to be a problem for the steady state results, in fact the lowest differences in the results are obtained with grid 1e-3 that has, in average, 68% of elements in the buffer layer. As can be seen in the table 5.1 the grid 1e-3 has the lowest maximum non-orthogonality and the lowest average non-orthogonality. Thus we can infer that, in our simulations, non-orthogonality plays a more important role with respect to the buffer layer.

5.3 Turbulence Model Steady State Results Comparison

In order to compare the two turbulence models, graphs of power and thrust and the relative numerical values for the two turbulence model are shown. In this case we compute the difference by means of the formula:

$$\frac{k\omega SST - SpalartAllmaras}{k\omega SST} \cdot \% \tag{5.2}$$

Turbulence Model Comparison								
	Mesh 3e-5		Mesh 3e-4		Mesh 1e-3		Mesh 5e-3	
Wind Speed	Thrust	Power	Thrust	Power	Thrust	Power	Thrust	Power
7	1.97%	3.04%	1.23%	1.50%	1.01%	1.55%	0.49%	1.58%
8	2.22%	3.35%	1.22%	1.34%	0.96%	1.42%	0.36%	1.09%
9	2.39%	3.53%	1.18%	1.25%	0.84%	1.36%	0.27%	0.94%
10	2.56%	3.89%	1.15%	1.18%	0.83%	1.34%	0.28%	0.98%
11	2.64%	3.91%	1.08%	1.10%	0.83%	1.33%	0.15%	0.82%

Table 5.28 Turbulence Model Comparison

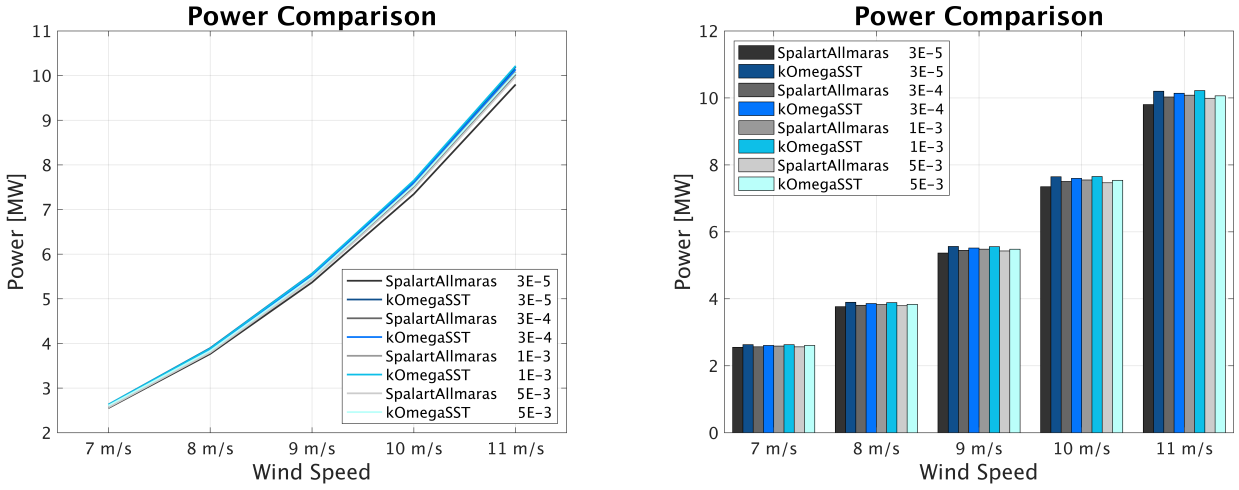


Fig. 5.7 Turbulence Model Comparison Comparison - Power

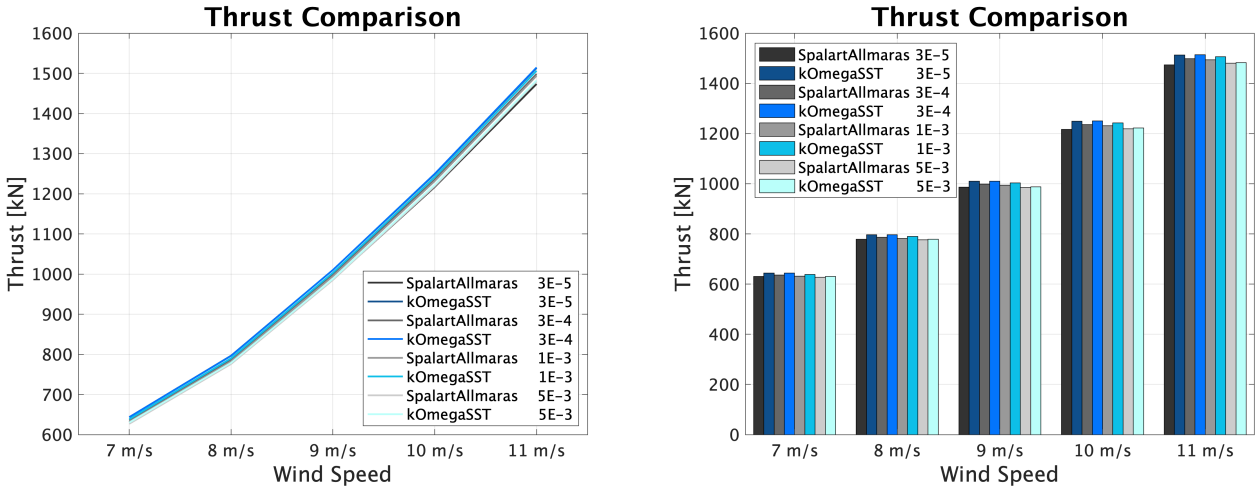


Fig. 5.8 Turbulence Model Comparison Comparison - Power

Mechanical power and thrust are always lower with Spalart-Allmaras turbulence model. The results are almost identical with the coarsest grid (5e-3), thus employing wall functions, while the highest difference is in the finest grid (3e-5). We can also visualize the pressure and velocity fields obtained with the two models for each mesh in three different sections: $25\%R$, $50\%R$ and $75\%R$, where R is the blade radius. On the left side we have the Spalart-Allmaras model while on the right side we have $k\omega SST$ model. Small differences can be identified.

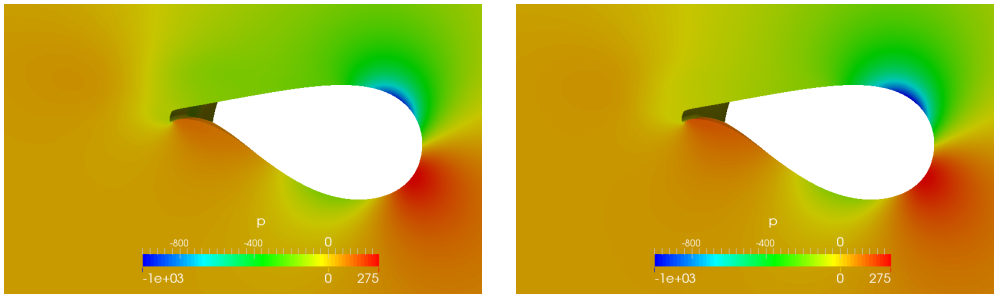


Fig. 5.9 Pressure Profile - 25%R - Mesh 3e-5

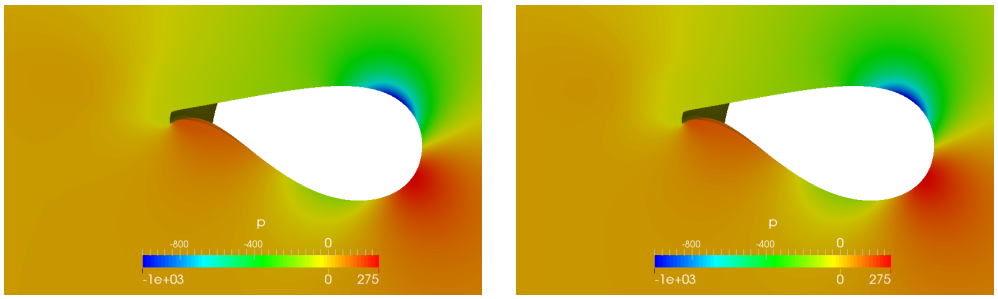


Fig. 5.10 Pressure Profile - 25%R - Mesh 3e-4

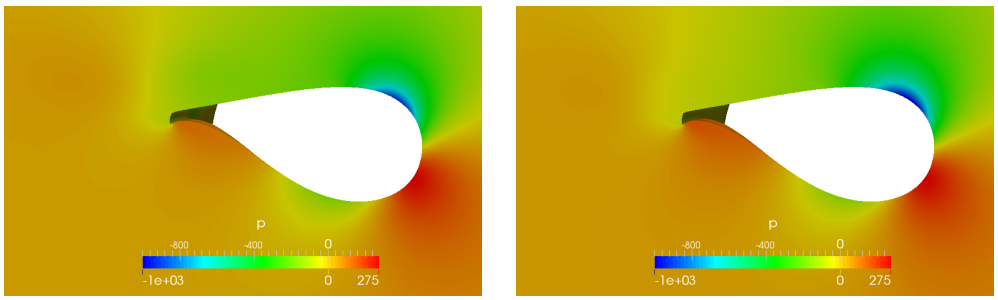


Fig. 5.11 Pressure Profile - 25%R - Mesh 1e-3

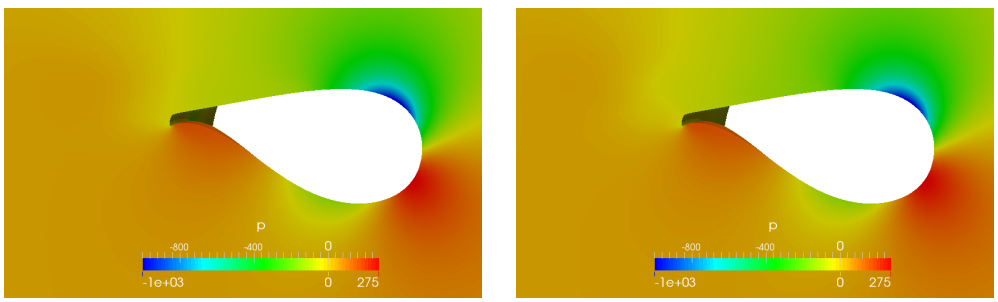


Fig. 5.12 Pressure Profile - 25%R - Mesh 5e-3

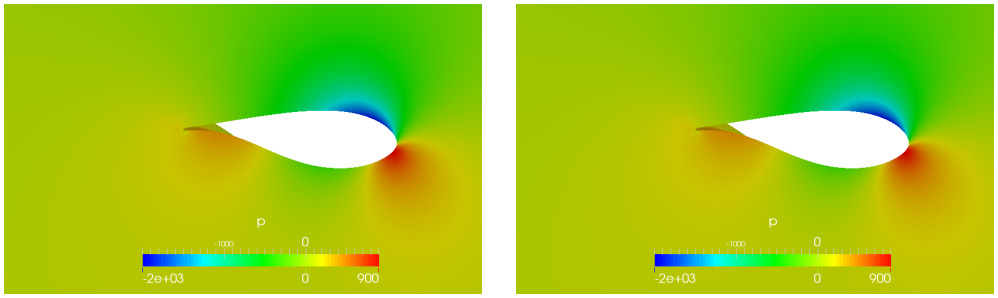


Fig. 5.13 Pressure Profile - 50%R - Mesh 3e-5

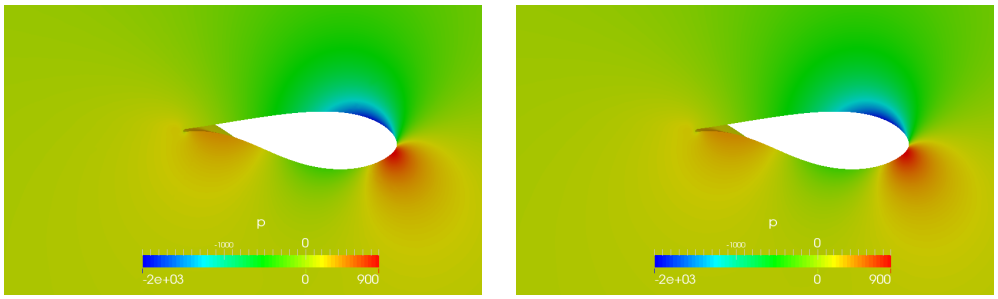


Fig. 5.14 Pressure Profile - 50%R - Mesh 3e-4

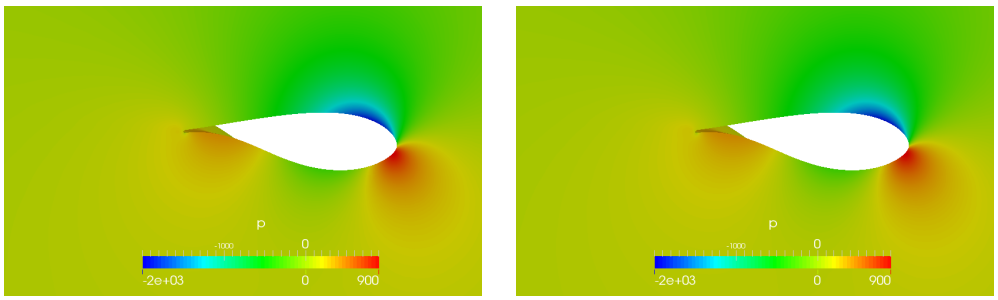


Fig. 5.15 Pressure Profile - 50%R - Mesh 1e-3

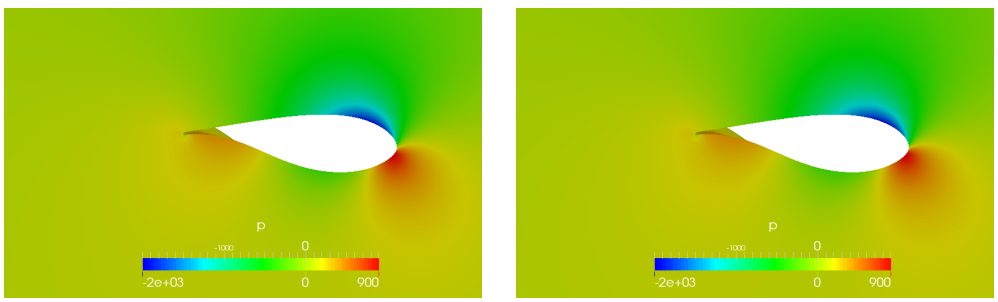


Fig. 5.16 Pressure Profile - 50%R - Mesh 5e-3

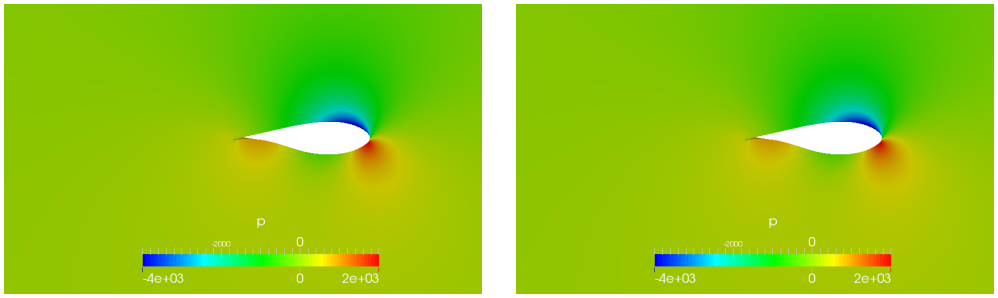


Fig. 5.17 Pressure Profile - 75%R - Mesh 3e-5

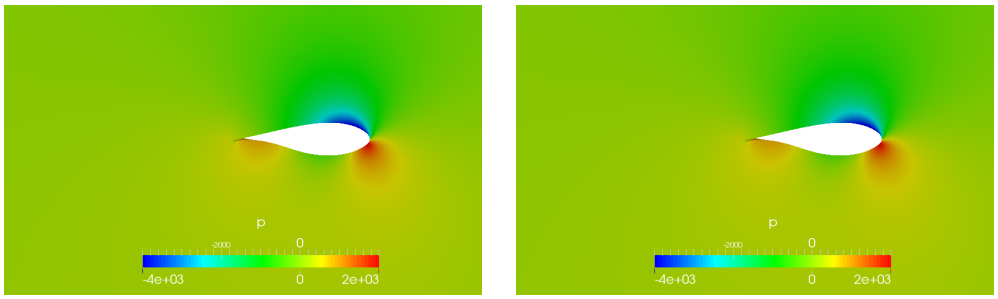


Fig. 5.18 Pressure Profile - 75%R - Mesh 3e-4

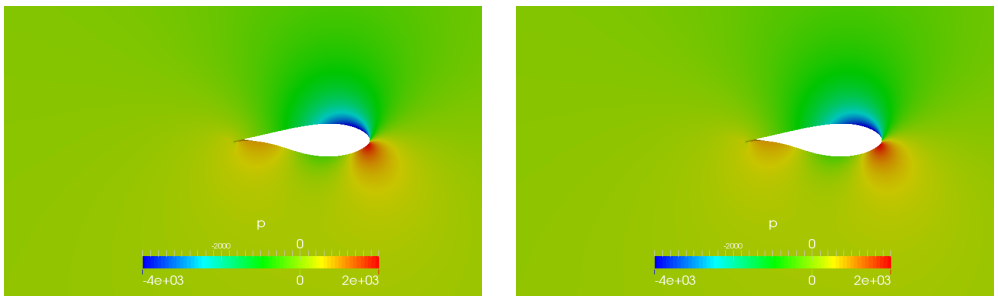


Fig. 5.19 Pressure Profile - 75%R - Mesh 1e-3

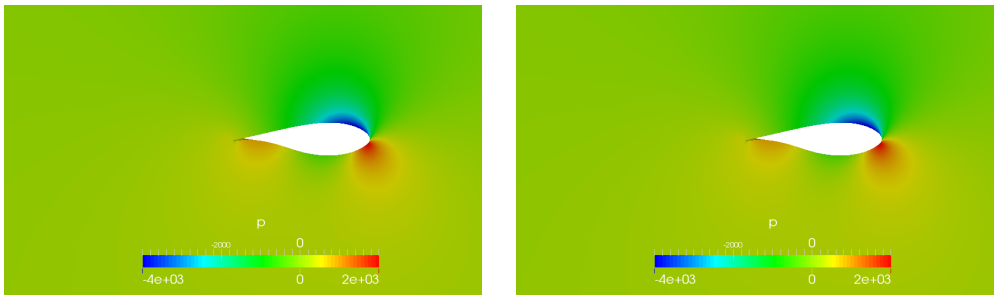


Fig. 5.20 Pressure Profile - 75%R - Mesh 5e-3

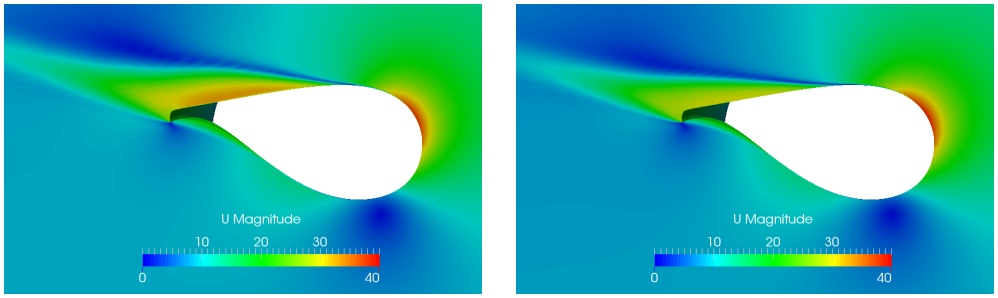


Fig. 5.21 Velocity Profile - 25%R - Mesh 3e-5

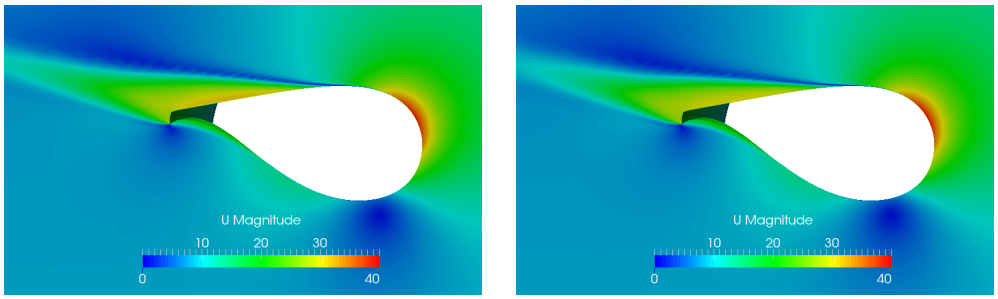


Fig. 5.22 Velocity Profile - 25%R - Mesh 3e-4

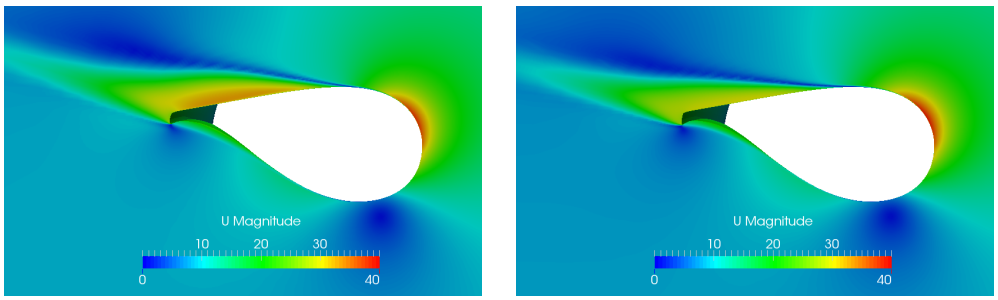


Fig. 5.23 Velocity Profile - 25%R - Mesh 1e-3

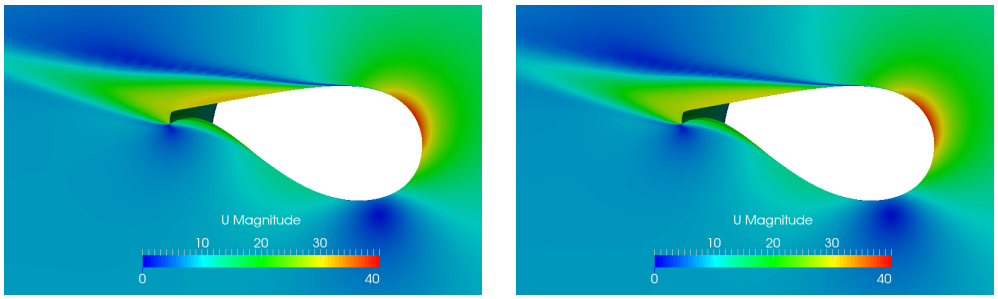


Fig. 5.24 Velocity Profile - 25%R - Mesh 5e-3

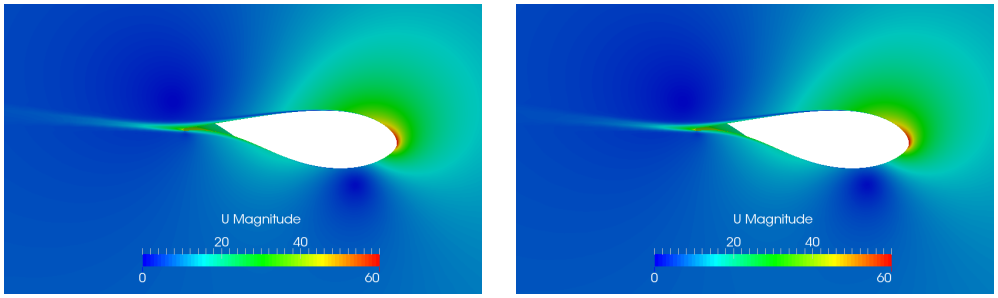


Fig. 5.25 Velocity Profile - 50%R - Mesh 3e-5

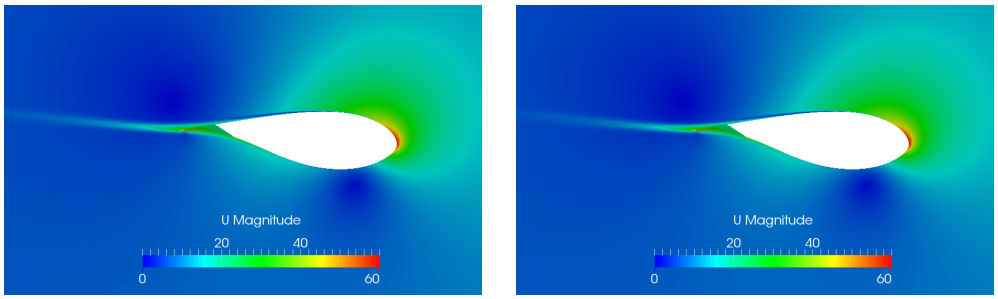


Fig. 5.26 Velocity Profile - 50%R - Mesh 3e-4

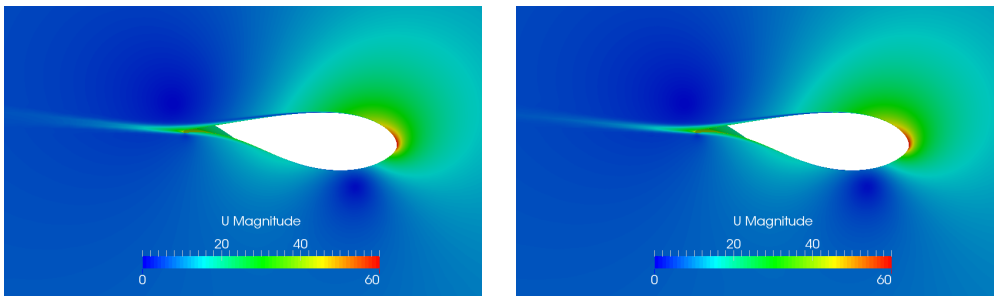


Fig. 5.27 Velocity Profile - 50%R - Mesh 1e-3

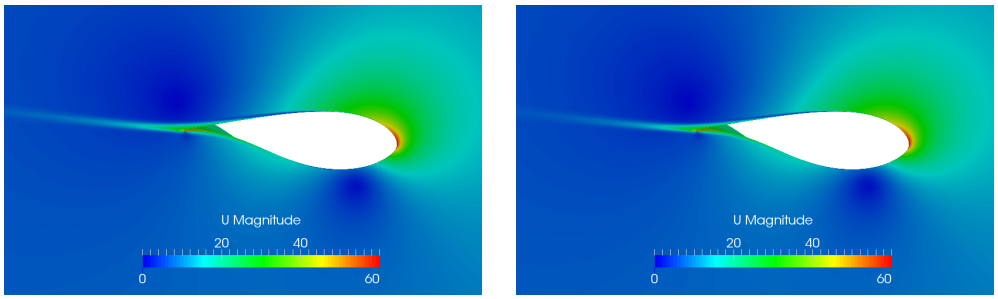


Fig. 5.28 Velocity Profile - 50%R - Mesh 5e-3

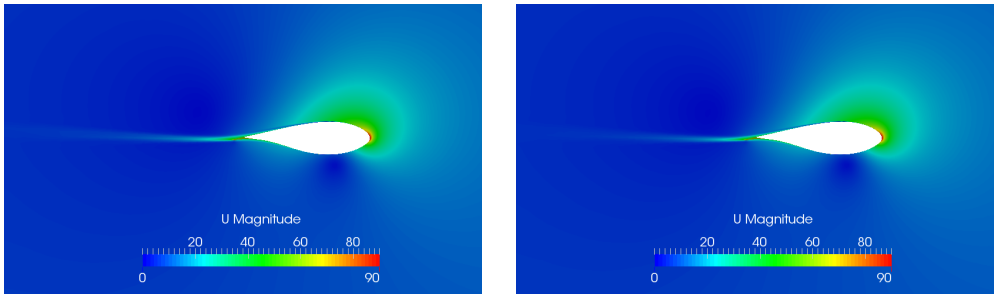


Fig. 5.29 Velocity Profile - 75%R - Mesh 3e-5

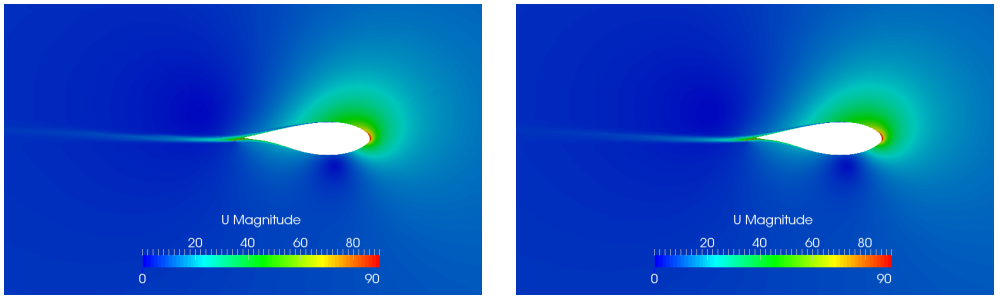


Fig. 5.30 Velocity Profile - 75%R - Mesh 3e-4

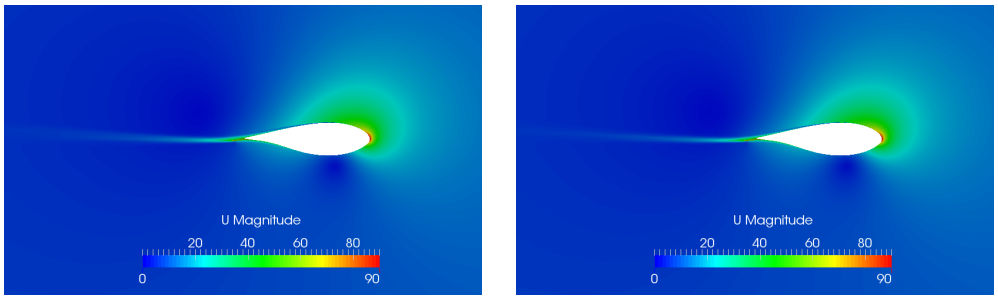


Fig. 5.31 Velocity Profile - 75%R - Mesh 1e-3

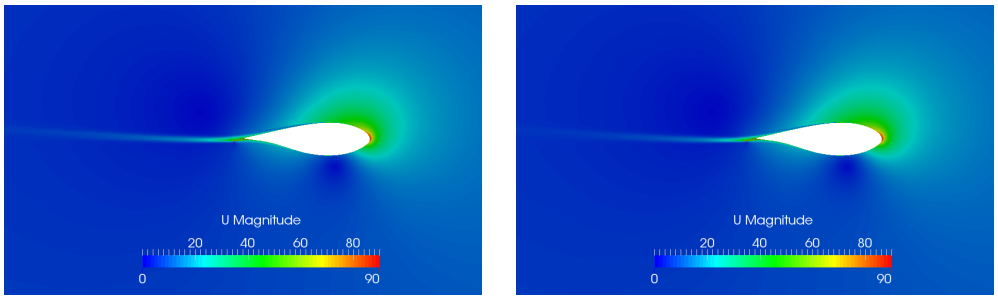


Fig. 5.32 Velocity Profile - 75%R - Mesh 5e-3

5.4 Wall Function Steady State Results Comparison

In order to understand the effectiveness of the wall functions, the results from the finest mesh ($y^+ < 1$) and the coarsest mesh ($y^+ > 30$) are compared.
In this case the difference is computed as:

$$\frac{Direct - WallFunctions}{Direct} \cdot \% \tag{5.3}$$

Wall Function Comparison				
	Spalart-Allmaras		***kωSST***	
Wind Speed	**Thrust**	**Power**	**Thrust**	**Power**
7	0.63%	-0.54%	2.12%	0.95%
8	0.39%	-0.75%	2.24%	1.54%
9	0.07%	-1.16%	2.19%	1.49%
10	-0.17%	-1.62%	2.12%	1.37%
11	-0.45%	-1.82%	2.06%	1.34%

Table 5.29 Wall Function Comparison

For the Spalart-Allmaras model differences are very small in thrust while the power computed employing wall functions is always higher with respect to the power computed directly. Taking as reference the DTU results, that did not employed wall functions, the errors are lower for the coarsest mesh.
The $k\omega SST$ model presents higher differences than the Spalart-Allmaras model. The direct solution gives always higher values both in power and thrust. In contrast to the Spalart-Allmaras results the differences are higher for thrust. Comparing the results with the DTU the error for the power is lower employing wall functions while the error for the thrust is lower with the finest mesh.

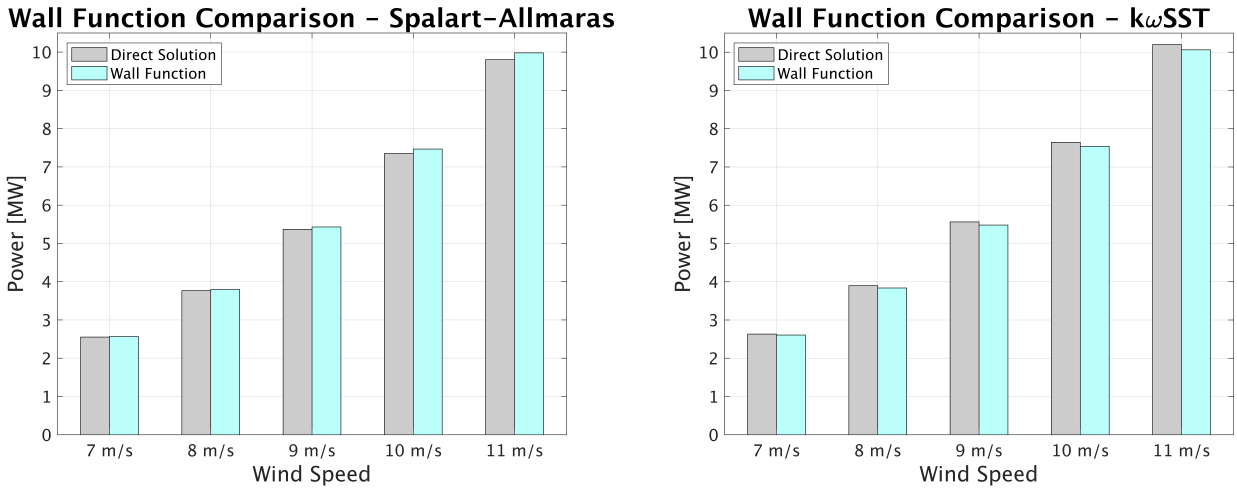


Fig. 5.33 Wall Function Comparison - Power

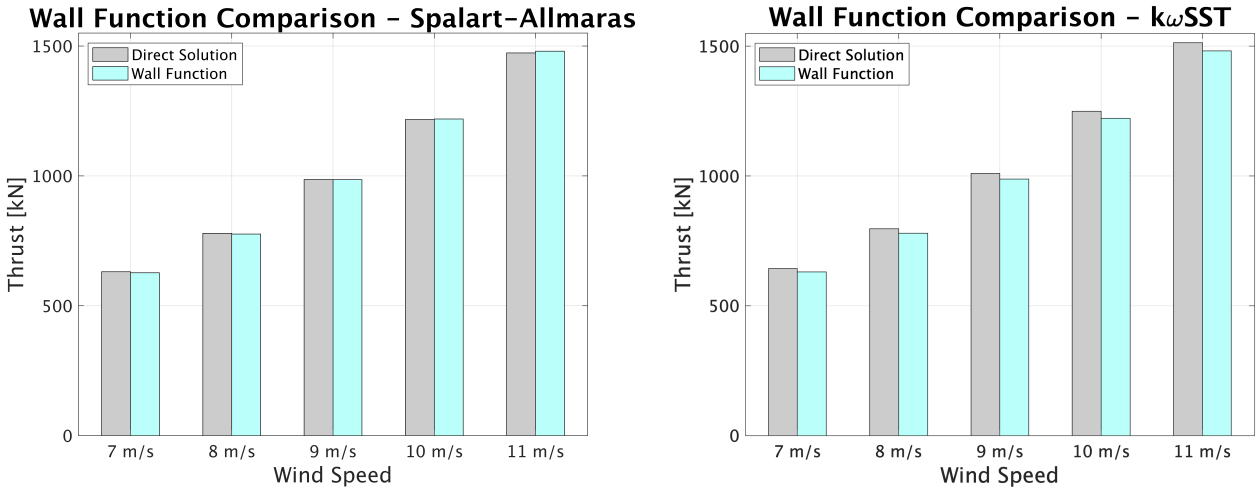


Fig. 5.34 Wall Function Comparison - Thrust

5.5 Transient Results

Starting from steady state results, dynamic simulations were run. Due to the high computational cost we decided to run these cases only at the nominal speed and only for four grids. The dynamic mesh technique described in chapter 2 has been used. The time step used corresponds to a blade rotation of 1° and each simulation was run for 12 rotations. We decided to run the coarsest mesh in order to see if also with transient simulations we can achieve good results as in the steady state. We also run cases with the mesh that has a high percentage of elements in the buffer layer (3-e4) for investigating its effects also in transient simulations. Both grids were run with Spalart-Allmaras and $k\omega SST$ model.

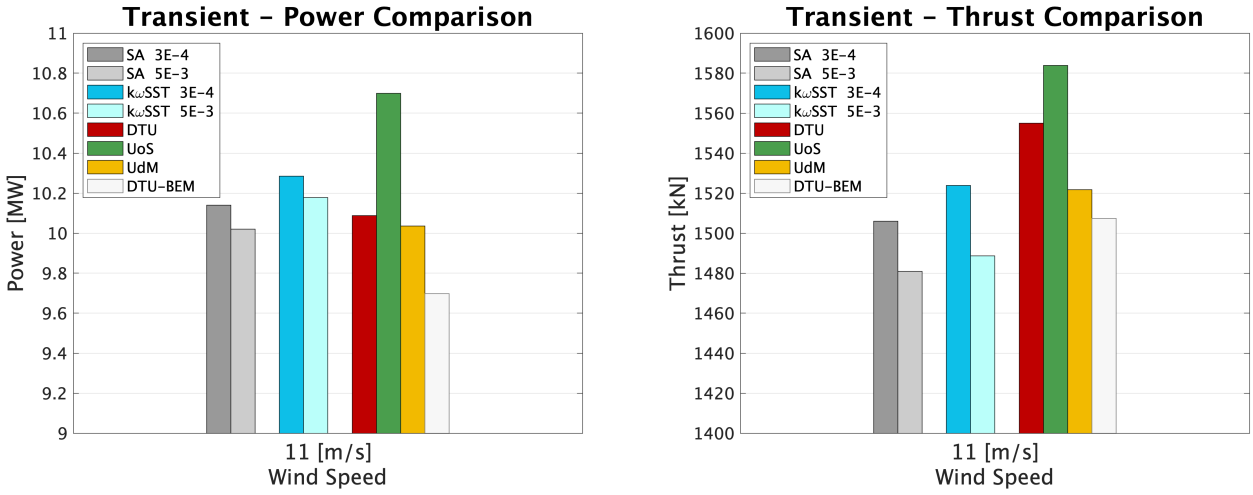


Fig. 5.35 Transient Result Comparison

Spalart Allmaras – Power					
First Element Length	**Power[MW]**	**DTU**	**UoS**	**UdM**	**BEM**
3.00E-004	10.14	-0.51%	5.23%	-1.04%	-4.56%
5.00E-003	10.02	0.68%	6.35%	0.16%	-3.32%
***kωSST* - Power**					
First Element Length	**Power[MW]**	**DTU**	**UoS**	**UdM**	**BEM**
3.00E-004	10.28	-1.95%	3.88%	-2.48%	-6.05%
5.00E-003	10.18	-0.88%	4.88%	-1.41%	-4.94%

Table 5.30 Transient Result Comparison - Power

Spalart Allmaras – Thrust					
First Element Length	**Thrust[kN]**	**DTU**	**UoS**	**UdM**	**BEM**
3.00E-004	1505.94	3.15%	4.92%	1.04%	0.10%
5.00E-003	1480.86	4.77%	6.50%	2.69%	1.76%
***kωSST* - Thrust**					
First Element Length	**Thrust[kN]**	**DTU**	**UoS**	**UdM**	**BEM**
3.00E-004	1523.94	2.00%	3.78%	-0.14%	-1.10%
5.00E-003	1488.60	4.27%	6.01%	-2.18%	1.25%

Table 5.31 Transient Result Comparison - Thrust

From the tables we can see that, for transient simulations, Spalart-Allmaras model is closer to DTU results in power while $k\omega SST$ model still behaves better for thrust prediction. Also in this case buffer layer does not produce significant differences in the results. The difference from the results obtained from the two grids is always lower than 1%.

Power Comparison			
First Element Length	**Spalart-Allmaras**	***kωSST***	**Difference**
3.00E-004	10.14	10.28	1.41%
5.00E-003	10.02	10.18	1.55%

Table 5.32 Transient Turbulence Model Thrust Comparison

Thrust Comparison			
First Element Length	**Spalart-Allmaras**	***kωSST***	**Difference**
3.00E-004	1505.94	1523.94	1.18%
5.00E-003	1480.86	1488.6	0.52%

Table 5.33 Transient Turbulence Model Thrust Comparison

The turbulence model seems to have the same impact on the results seen in steady state simulations, the differences in the results are always lower than 1.5%.

Chapter 6

Conclusions and Suggestions for Further Developments

6.1 Methodology

A methodology for executing high fidelity CFD simulations in the Statik Department at the Technische Unviersitaet Muenchen has been found. The methodology was needed by the department for executing fluid-structure interaction (FSI) simulations, a research field of the department, with the in-house software EMPIRE. The various steps of the methodology are illustrated here.

6.1.1 Grid Creation

The best solution for creating CFD grids was found to be the three dimensional tetrahedral extrusion implemented in the software Pointwise. Every step of the meshing creation procedure with the related problems has been illustrated in chapter 3. The hybrid meshing technique can significantly decrease the number of cells in the model with respect to a standard structured grid, thus decreasing the computational costs while assuring accuracy in the results. From a literature research we can see that the majority of the grids created for CFD wind turbine simulations is of the overset type [54] [17] [57] [45] [29][47], followed by hybrid mesh [18] [20] [21] and by structured mesh [11] [25] [12] [16]. The overset type would allow faster modifications of the grid, in particular it would

be very easy to change the pitch angle of the blade. Since now OpenFoam is capable of dealing with overset grids it would be worth carrying out a study on the accuracy of this method. Furthermore, the results show that the influence of the first element length, so of the boundary layer, is minimal. A parameter that seems to play a more important role is the non-orthogonality. Thus the author suggests to pay more attention to this parameter.

6.1.2 Boundary Conditions

Reducing the domain and using cyclic AMI boundary conditions is the fastest way if we are interested in study just the rotor performance, it is also found in literature to be a common solution. It could not be applied if we are interested in evaluating the global performance of the turbine. Rotor-tower interactions for optimizing performances is still a research field under study [26].
Turbulence boundary conditions quantities were computed using the cylinder diameter length as turbulence length as it was personally suggested to me by Prof. dr. ir. Joris Degroote from Ghent University.

6.1.3 Numerical Discretization Schemes

Numerical schemes plays a fundamental role in the convergence of the residuals and in the accuracy of results. Different attempts were made with different numerical discretization schemes in order to obtain the most accurate result. In the end for gradient, laplacian and interpolation schemes linear schemes were always adopted. The most influencing schemes were the divergence schemes, in particular velocity one, since it is the non-linear variable in NS equations. First order upwind was always used for turbulence variables while second order upwind was used for the velocity. The simulations, both steady state and transient, started with upwind divergence scheme. After the convergence was reached, the final results were used as initial conditions for a new simulation employing bounded second order upwind scheme (linearUpwind). For transient simulations *backward* scheme was always employed.

6.1.4 Solver Algorithm and settings

Simple consistent algorithm was used for every simulation, employing high relaxation factor, 0.7 for pressure field and 0.9 for every other variable. The number of iterations for correcting non-orthogonality was varied for depending on the grid, the range was always between 3 and 6. Since ewe were dealing with an high number of cells Geometric Algebraic Multi Grid (GAMG) [14] method was used for solving the pressure field. For every other variable Preconditioned Bi-Conjugate Gradient Stabilized [52][16] was found to be as the fastest in reaching residuals convergence. Absolute residuals tolerance was set to 1e-8 for each solver.

6.1.5 Turbulence Model and Boundary Conditions

From literature research we can state that for wind turbine simulation the $k\omega SST$ [11] [47] [20] [54] [45][12] model is the most used and thus one of the most reliable. Spalart-Allmaras model is preferred when FSI simulations are preferred [25] [26] [18] for its better convergence performances.
In our simulations the difference in turbulence models behaviour is higher when we have a $y^+ < 1$, while when we employ wall functions the error is, on average, lower than 1%. The latter consideration is valid for both steady state and transient simulations. $k\omega SST$ is more sensitive to wall function use while Spalart-Allmaras seems to be almost insensitive to this.
In the end $k\omega SST$ is suggested for concept studies on wind turbine aerodynamics in which we are interested in understanding how pressure and velocity are behaving, for instance in the study of Gurney flap employment. Spalart-Allmaras seems to be the best solution for concept structural design, so in FSI simulations, in which we prefer to have faster convergence and mid-level accuracy on load prediction.

6.2 Further Developments

Many studies to improve this methodology can be done. A sensitivity analysis can be done on the CFD domain in order to assure mesh independence and minimize the number of elements. Since OpenFoam is now capable of dealing with Overset mesh method, the grid creation and the relative results could be studied. Solvers and numerical schemes parameters can be optimized to reach a faster convergence. FSI simulation can be done in order to see what is the sensitivity of the turbulence model on the result. Since the nacelle was not considered in the simulations it would be worth adding it to see what is the influence on mechanical power and thrust computation.

|

Bibliography

[wik] The mrf development. http://openfoamwiki.net/index.php/See_the_MRF_development. Accessed: 2017-10-01.

[NLR] Nasa langley research center website. https://turbmodels.larc.nasa.gov/. Accessed: 2017-05-01.

[ope] Openfoam website. https://www.openfoam.com/. Accessed: 2017-04-01.

[Ors] Orsted website. https://orsted.com/en. Accessed: 2019-02-.

[Poi] Pointwise website. www.pointwise.com. Accessed: 2017-06-01.

[Sie] Scoe the true cost of offshore wind power. https://www.siemens.com/customer-magazine/en/home/energy/renewable-energy/scoe-the-true-cost-of-offshore-wind-power.html. Accessed: 2017-10-01.

[7] Ackermann, T. (2005). *Wind power in power systems*. John Wiley & Sons.

[8] AG, S. (2014). What is the real cost of offshore wind? Technical report, Siemens AG, Wind Power, Beim Strohhause 17-31, 20097 Hamburg, Germany.

[9] Agency, I. R. E. (2018). Renewable power generation costs in 2017.

[10] asbl/vzw, W. (2019). Wind energy in europe in 2018. Technical report, WindEurope asbl/vzw, Rue d'Arlon 80, B-1040 Brussels, Belgium.

[11] Bak, C., Zahle, F., Bitsche, R., Kim, T., Yde, A., Henriksen, L. C., Hansen, M. H., Blasques, J. P. A. A., Gaunaa, M., and Natarajan, A. (2013). The dtu 10-mw reference wind turbine. In *Danish Wind Power Research 2013*.

[12] Bangga, G., Guma, G., Lutz, T., and Krämer, E. (2018). Numerical simulations of a large offshore wind turbine exposed to turbulent inflow conditions. *Wind Engineering*, 42(2):88–96.

[13] Bardina, J. E., Huang, P. G., and Coakley, T. J. (1997). Turbulence modeling validation, testing, and development.

[14] Behrens, T. (2009). Openfoam's basic solvers for linear systems of equations. *Chalmers, Department of Applied Mechanics*, 18(02).

[15] Burton, T., Jenkins, N., Sharpe, D., and Bossanyi, E. (2011). *Wind energy handbook*. John Wiley & Sons.

[16] Cao, J., Zhu, W., Shen, W., Sørensen, J., and Wang, T. (2018). Development of a cfd-based wind turbine rotor optimization tool in considering wake effects. *Applied Sciences*, 8(7):1056.

[17] Douteau, L., Silva, L., Digonnet, H., Coupez, T., Le Touzé, D., and Gilloteaux, J.-C. (2019). Towards numerical simulation of offshore wind turbines using anisotropic mesh adaptation. In *Recent Advances in CFD for Wind and Tidal Offshore Turbines*, pages 95–104. Springer.

[18] Ehrich, S., Schwarz, C., Rahimi, H., Stoevesandt, B., and Peinke, J. (2018). Comparison of the blade element momentum theory with computational fluid dynamics for wind turbine simulations in turbulent inflow. *Applied Sciences*, 8(12):2513.

[19] Fluent, A. et al. (2015). Theory guide and user's guide. *Ansys Inc, USA*.

[20] Halawa, A. M., Yoshida, S., and Ismaiel, A. M. (2018). Fluid-structure interaction computations for wind turbine blade. In *Grand Renewable Energy proceedings Japan council for Renewable Energy*, page 179. Japan Council for Renewable Energy.

[21] Hand, B., Cashman, A., and Kelly, G. (2018). Aerodynamic analysis of a 5 mw stall-regulated offshore vertical axis wind turbine using computational fluid dynamics. In *Vietnam Symposium on Advances in Offshore Engineering*, pages 485–491. Springer.

[22] Hansen, M. O. (2015). *Aerodynamics of wind turbines*. Routledge.

[23] Hellsten, A. (1998). Some improvements in menter's k-omega sst turbulence model. In *29th AIAA, Fluid Dynamics Conference*, page 2554.

[24] Holzmann, T. (2016). Mathematics, numerics, derivations and openfoam®.

[25] Horcas, S. G. (2016). *CFD Methodology for Wind Turbines Fluid-Structure Interaction*. PhD thesis, Université de Mons (UMONS).

[26] Horcas, S. G., Debrabandere, F., Tartinville, B., Hirsch, C., and Coussement, G. (2016). Cfd study of dtu 10 mw rwt aeroelasticity and rotor-tower interactions. In *MARE-WINT*, pages 309–334. Springer, Cham.

[27] Jang, C. S., Ross, J. C., and Cummings, R. M. (1998). Numerical investigation of an airfoil with a gurney flap. *Aircraft Design*, 1(2):75–88.

[28] Jasak, H. (2009). Dynamic mesh handling in openfoam. In *47th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, page 341.

[29] Ke, S., Yu, W., Cao, J., and Wang, T. (2018). Aerodynamic force and comprehensive mechanical performance of a large wind turbine during a typhoon based on wrf/cfd nesting. *Applied Sciences*, 8(10):1982.

[30] Kolgomorov (1942). Equations of turbulent motion of an incompressible fluid. *zvestia Acad Sci USSR Phys 6(1 and 2):56–58*.

[31] Launder, B. and Spalding, D. (1974). The numerical computation of turbulent flow computer methods in applied mechanics and engineering.

[32] Liu, W.-H. and Sherman, A. H. (1976). Comparative analysis of the cuthill–mckee and the reverse cuthill–mckee ordering algorithms for sparse matrices. *SIAM Journal on Numerical Analysis*, 13(2):198–213.

[33] Lutton, M. J. (1989). Comparison of c-and o-grid generation methods using a naca 0012 airfoil. Technical report, AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING.

[34] Menter, F. R., Kuntz, M., and Langtry, R. (2003). Ten years of industrial experience with the sst turbulence model. *Turbulence, heat and mass transfer*, 4(1):625–632.

[35] Moraes, A., Lage, P., Cunha, G., and Silva, L. (2013). Analysis of the non-orthogonality correction of finite volume discretization on unstructured meshes.

[36] Moriarty, P. and Honnery, D. (2019). Global renewable energy resources and use in 2050. In *Managing Global Warming*, pages 221–235. Elsevier.

[37] Moukalled, F., Mangani, L., Darwish, M., et al. (2016). *The finite volume method in computational fluid dynamics*. Springer.

[38] Musial, W., Beiter, P., Schwabe, P., Tian, T., Stehly, T., Spitsen, P., Robertson, A., and Gevorgian, V. (2017). 2016 offshore wind technologies market report. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States).

[39] of Energy, U. D. (2018). 2017 offshore wind technologies market update. Technical report.

[40] on Climate Change, I. P. and Edenhofer, O. (2015). *Climate Change 2014: Mitigation of Climate Change*. Assessment report. Cambridge University Press.

[41] Ostachowicz, W., McGugan, M., Schröder-Hinrichs, J.-U., and Luczak, M. (2016). *MARE-WINT*. Springer.

[42] Patankar, S. V. and Spalding, D. B. (1983). A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. In *Numerical Prediction of Flow, Heat Transfer, Turbulence and Combustion*, pages 54–73. Elsevier.

[43] Quarteroni, A., Sacco, R., and Saleri, F. (2008). *Matematica numerica*. Springer Science & Business Media.

[44] REN21 (2018). Renewables 2018 global status report. Technical report, REN21.

[45] Rocchio, B., Deluca, S., Salvetti, M. V., and Zanforlin, S. (2018). Development of a bem-cfd tool for vertical axis wind turbines based on the actuator disk model. *Energy Procedia*, 148:1010–1017.

[46] Saffman, P. and Wilcox, D. (1974). Turbulence-model predictions for turbulent boundary layers. *AIAA journal*, 12(4):541–546.

[47] Sayed, M., Lutz, T., and Krämer, E. (2015). Aerodynamic investigation of flow over a multi-megawatt slender bladed horizontal-axis wind turbine. *Renew. Energ. Offshore*.

[48] Sayed, M., Lutz, T., Krämer, E., Shayegan, S., Ghantasala, A., Wüchner, R., and Bletzinger, K.-U. (2016). High fidelity cfd-csd aeroelastic analysis of slender bladed horizontal-axis wind turbine. In *Journal of Physics: Conference Series*, volume 753, page 042009. IOP Publishing.

[49] Sciacovelli, A., Borchiellini, R., and Verda, V. (2013). *Numerical design of thermal systems*. CLUT.

[50] Skye, H., Nellis, G., and Klein, S. (2006). Comparison of cfd analysis to empirical data in a commercial vortex tube. *International journal of refrigeration*, 29(1):71–80.

[51] Sleijpen, G. L. and Van der Vorst, H. A. (1995). *Hybrid bi-conjugate gradient methods for CFD problems*. Citeseer.

[52] Sleijpen, G. L., Van der Vorst, H. A., and Fokkema, D. R. (1994). Bicgstab (l) and other hybrid bi-cg methods. *Numerical Algorithms*, 7(1):75–109.

[53] Spalart, P. and Allmaras, S. (1992). A one-equation turbulence model for aerodynamic flows. In *30th aerospace sciences meeting and exhibit*, page 439.

[54] Tran, T. T. and Kim, D.-H. (2018). A cfd study of coupled aerodynamic-hydrodynamic loads on a semisubmersible floating offshore wind turbine. *Wind Energy*, 21(1):70–85.

[55] UN (2015). Adoption of the paris agreement. https://unfccc.int/resource/docs/2015/cop21/eng/l09r01.pdf.

[56] Van Doormaal, J. and Raithby, G. (1984). Enhancements of the simple method for predicting incompressible fluid flows. *Numerical heat transfer*, 7(2):147–163.

[57] Vishwakarma, P. and Patel, A. S. (2018). Aerodynamic design and study of wind turbine blade aerofoil.

[58] Walters, D. K. and Cokljat, D. (2008). A three-equation eddy-viscosity model for reynolds-averaged navier–stokes simulations of transitional flow. *Journal of fluids engineering*, 130(12):121401.

[59] Wilcox, D. C. (2008). Formulation of the kw turbulence model revisited. *AIAA journal*, 46(11):2823–2838.