POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Novel Techniques for Actuator Line Modelling of Floating Offshore Wind Turbines

Tesi di Laurea Magistrale in
Energy Engineering - Ingegneria Energetica

Author: **Agnese Firpo**

Student ID: 966273
Advisor: Giacomo Bruno Azzurro Persico
Co-advisors: Andrea Giuseppe Sanvito, Paolo Schito
Academic Year: 2021-22

Abstract

The main goal of this thesis work is the formulation and validation of three different versions of an in-house CFD solver. The code, consisting in a formulation of the Actuator Line Model (ALM) implemented into the OpenFOAM software, is suitable for floating offshore wind turbines applications. The new formulation proposes three new velocity sampling methods needed to provide an accurate estimate of the angle of attack and therefore of the blade loads. The aim is to sample, directly from the CFD flow field, the real velocity perceived by the blades excluding the bound circulation effect associated to lift production. The velocity is sampled close to the actuator line point, where the body-force is applied, but far enough not to be affected by oscillations and disturbances of a numerical nature. Before reaching the final configuration of the proposed methods, a detailed analysis is conducted using different sampling techniques to identify the main problems. From the resolution of the difficulties encountered, the optimal parameters of the sampling geometry, of the force distribution and the most adequate level of mesh refinement were defined. The three strategies are motivated by different physical reasons coming from a detailed flow field analysis. The validation is achieved through a comparison with experiments performed on a scaled-down wind turbine laboratory model, operating under steady wind and floating conditions, at the Wind Tunnel of the Politecnico di Milano.
Initially, each method is assessed in the bottom-fixed working condition and then is applied in presence of surge and pitching platform motions. The results of the three versions of the ALM code are found to be in excellent agreement with experimental data in fixed-bottom, surge and pitch conditions. The work proves that the new models enable to estimate reliable velocity with physical span-wise trends coherent with the real force distribution. The new methods appear to be a valid alternative to ALM simulation tools already tested in literature, thanks to their ease of implementation and the physical motivation on which they are based.

Keywords:: Floating Offshore Wind Turbines, Actuator Line Model, velocity sampling method, unsteady rotor aerodynamics

Abstract in lingua italiana

L'obiettivo principale di questa tesi è la formulazione e validazione di tre diverse versioni di un codice CFD interno. Il codice, che consiste in una formulazione del modello Linea Attuatrice (ALM) e che è implementato nel software OpenFoam, è adatto alle turbine eoliche offshore flottanti. La nuova formulazione propone tre nuovi metodi di campionamento della velocità, necessari per fornire una stima accurata dell'angolo di attacco e quindi dei carichi palari. Lo scopo è campionare, direttamente dal campo di moto CFD, la reale velocità percepita dalle pale escludendo l'effetto della circolazione associata alla generazione di portanza. La velocità viene campionata vicino al punto attuatore, dove è applicata la forza di volume, ma abbastanza distante per non risentire di oscillazioni o disturbi di natura numerica. Prima di raggiungere la configurazione finale dei metodi proposti, è stata realizzata un'analisi dettagliata su diverse tecniche di campionamento identificandone i principali problemi. Dalla risoluzione delle difficoltà incontrate, sono stati definiti i parametri ottimi della geometria di campionamento, della distribuzione della forza e il livello più adeguato di infittimento della mesh. Le tre strategie si basano su motivazioni fisiche derivanti da un'analisi dettagliata del campo di moto. I metodi sono stati validati confrontandoli con esperimenti realizzati su un modello scalato di turbina, operante in condizioni stabili e flottanti, presso la Galleria del Vento del Politecnico di Milano.
Inizialmente ogni metodo è stato validato nel caso di posizione fissa della turbina e poi applicato in presenza di moto di surge e pitch. I risultati ottenuti dalle tre versioni del codice ALM sono risultati in ottimo accordo con i dati sperimentali nel caso di base fissa e flottante. Questo lavoro dimostra che i nuovi modelli sono in grado di stimare una velocità affidabile, con andamenti palari fisici e coerenti con la reale distribuzione delle forze. I nuovi metodi si presentano come una valida alternativa agli strumenti basati sul ALM già testati in letteratura, grazie alla loro facilità di implementazione e alle ragioni fisiche su cui sono basati.

Parole chiave: Turbine Eoliche Offshore Flottanti, Modello a Linea Attuatrice, metodo di campionamento della velocità, Aerodinamica del rotore non stazionaria

Contents

1 | Introduction

1.1. The energy transformation

Reducing energy-related CO_2 emissions is at the heart of the actual energy transformation. If the world is to meet the avowed climate objectives, it must transition away from dirty fossil fuels toward cleaner, renewable types of energy. Climate change has become a great concern of this century. The Paris Agreement sets efforts to limit the global temperature rise to below 2°C and to limit warming to 1.5°C in the present century, compared to pre-industrial levels. To realise the climate targets of the Paris Agreement, a profound transformation in the global energy scenario is indispensable. In order to meet the goals of the Paris Agreement, energy-related CO_2 emissions should be reduced by around 3.5% per year from now until 2050 [15]. The global energy transformation can be reached with the combination of increasing low-cost renewable power technologies and the wider adoption of electric technologies for end-use applications in transport and heat. Energy sources like wind and solar will pave the path for this transformation. By 2050, onshore and offshore wind will supply more than one-third (35%) of the world electricity needs, taking over as the primary energy source [15] .

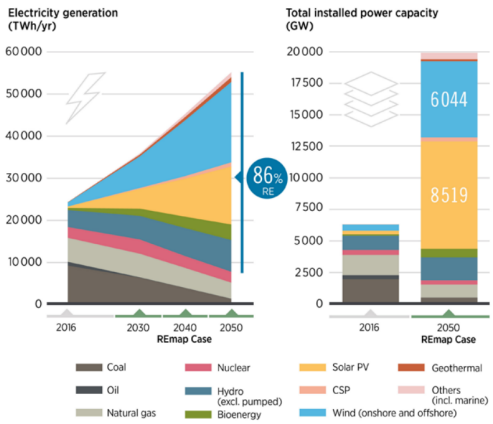


Figure 1.1: Future electricity generation and installed power capacity [15].

In 2050, wind power could help reduce CO_2 emissions by 6.3 Gt, 27% of the total emissions

reductions required to reach the Paris climate targets, as shown in Fig. 1.2.

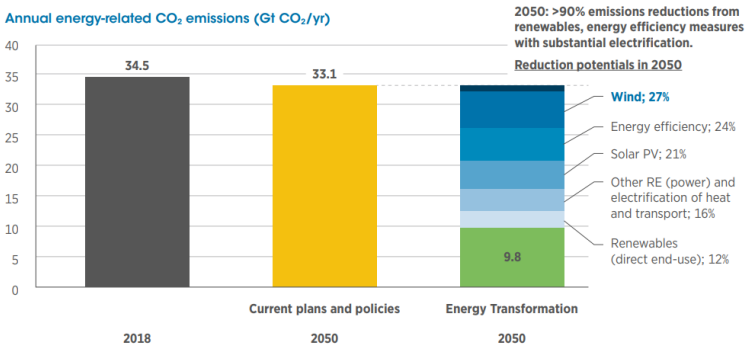


Figure 1.2: Annual energy-related CO_2 emissions [15].

1.1.1. Wind power in Europe

Only 17.4 GW of new wind installations were added in Europe in 2021 (14 GW onshore and 3.4 GW offshore), as delays in permits and the worldwide supply chain continue to prevent the development of new wind farms. Despite 2021 represents a record year for installations, they are still below the level needed by Europe to reach its 2030 energy and climate goals. In order to achieve the 40% renewable target, the EU-27 need to install 32 GW of new wind capacity each year [3]. The Realistic Expectations Scenario, proposed by Wind Europe [3], studies the best estimate of installed capacity in Europe and it demonstrates that a significant increase in wind power capacity is necessary over the course of the next five years. It is based on the supposition that governments uphold the promises they have previously made to enhance the approval processes for new wind farms, such as the new German policy that ensures more land for wind. Under this scenario, Europe will install 17.6 GW per year, for a total of 116 GW, which is well below of the 32 GW a year needed to reach the 40% renewable energy target. The peak for onshore installations, which will account for 76% of all new installations over the following five years, will be 18.3 GW in 2022. After that, they will gradually decline to 17.3 GW on average per year until 2026. Offshore installations will first breach the 4 GW limit in 2023, continuing to do so annually thereafter, and reaching 10 GW in 2026. Thanks to the UK's excellent Contracts for Difference (CfD) support program, almost 39% of European offshore installations will originate there.

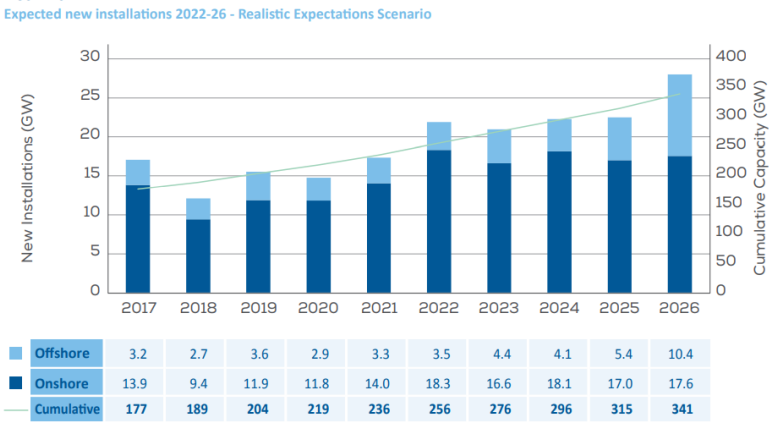


Figure 1.3: Expected new installations 2022-26 - Realistic Expectations Scenario [3]

According to the Realistic Expectations Scenario [3], between 2022 and 2026 Europe will install 27.9 GW of offshore wind capacity. Over the course of the next five years, offshore wind will account for around 24% of the market with an average annual output of 5.6 GW. The UK will host the majority of offshore installations (39%) but, as indicated in Fig. 1.4, eight more countries will also experience significant offshore development.

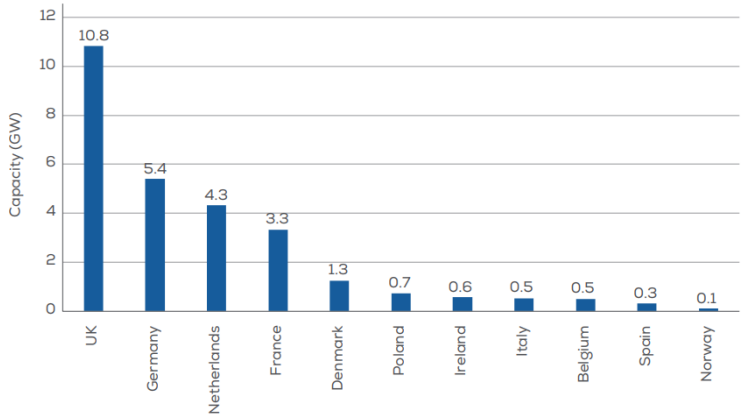


Figure 1.4: New offshore installations per country, 2022-26 - Realistic Expectations Scenario [3]

1.1.2. Floating offshore wind farms

The offshore wind is an important addition to the portfolio of low carbon technologies available to decarbonise the energy sector. This technology represents a competitive solution thanks to the numerous advantages with respect to the onshore one. An offshore wind farm is able to exploit higher wind average velocities and more stable winds. This is due to the presence, from a fluid-dynamic point of view, of a thinner boundary layer

on the sea surface, which leads to a faster wind shear development. Furthermore, offshore applications are advantaged by smaller seasonal effects increasing the equivalent hours of the plant. The other group of advantages is related to the lower environmental impact. Indeed, the big distance from the coast makes practically absent the visual impact. Additionally, the space available to install turbines is not physically limited allowing the construction of high power plants. For these reasons, the offshore market will grow significantly over the next three decades, with the total installed offshore wind capacity rising from just 23 GW in 2018 to 228 GW in 2030 and near 1000 GW in 2050 (Fig. 1.5). Offshore wind will represent nearly 17% of the total global installed wind capacity of 6044 GW in 2050 [15].

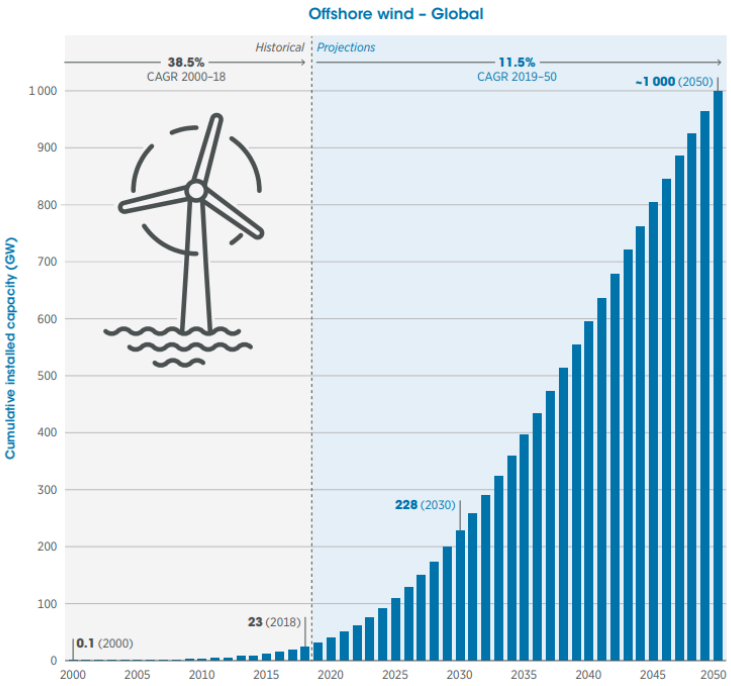


Figure 1.5: Global cumulative installed capacity from Offshore wind [15]

The great potential of these plants has promoted a strong improvement in technology over the years. This technological growth combined with the use of larger turbines has led offshore wind projects to reach capacity factor (CF) levels between 40% and 50% [11]. CF describes the average power output over the year relative to the maximum rated power capacity. These values enable wind energy to have very competitive Levelized Cost of Energy (LCOE), often already comparable to that of fossil fuel power plants. Given that the best offshore wind locations could provide more than the global electricity needs, offshore wind power has the potential to become one of the best renewable energy sources. Wind turbine offshore structures can be distinguished into two groups: bottom-fixed structures

and floating platforms. The former are less expensive installations located at a sea depth of up to 60 m, beyond which it would be too expensive an investment. Approximately 87,000 TWh per year is the global technical potential of the regions suitable for these kinds of structures [11]. The latter, conversely, can be installed also in areas very far from the coast, at a sea depth higher than 60 m and consequently at higher costs. Nevertheless, they allow to reach areas with a global technical potential of 330,000 TWh. The growing interest of industries will promote cost reductions and make this great potential more accessible. Europe, especially Northern Europe, is currently the leader in offshore wind. Since the Mediterranean Sea differs from the North Sea in a number of ways, particularly due to its steep bathymetric profile and generally deep waters, it is crucial to investigate various structural concepts that will promote floating technology.

The building of non-fixed offshore wind turbines faces a variety of difficulties: the issue of the floating platform stability, the power grid connection and the performance optimization. Where the first two issues can be addressed through the already developed technologies for the oil & gas field and through a consolidated knowledge on submarine power cables grids, the actual challenge is to gain an accurate prediction on floating wind turbine efficiency. There is a variety of floating foundations using different mooring technologies. In order to prevent displacements and consequent loss of efficiency, mooring systems hold floating platforms in fixed positions by applying forces to the seabed. Such mooring systems comprise of anchors and mooring lines made of different materials. Two main mooring systems can be distinguished [14]:

- **Taut-leg configuration**: they are mostly integrated with tension leg platforms (TLP). Taut-leg configuration consists of iron mooring lines made of synthetic fiber rope or wire. Taut-leg configurations have a smaller footprint and therefore a lower impact on the seabed because each mooring line is at an angle of 30-to-40 degrees. These systems show excellent stability and resistance to horizontal and vertical forces. However, because of the high tension provided by these configurations, they may suffer from metal fatigue. In addition, only special vessels can be used for this installation.

- **Catenary configurations**: they are usually used with spar-buoy and semi submersible foundations and generally require mooring lines made up of iron chains or wires. Compared to taut-leg systems, catenary configurations have a greater influence on the seabed and because of their larger footprint, they may interfere with fishing activities. This method provides a less stable but cheaper solution than the taut-leg configuration.

Currently, popular foundation types include spar-buoy foundation, tension leg platform, and semi-submersible foundation as reported in Fig. 1.6.

- **Spar-buoy**: The entire platform is stabilized installing spar buoys and ballast in the bottom half of the structure. Therefore, with a lower centre of gravity, dynamic stability mechanisms are not required. The structure is rather simple, therefore costs are limited. Spar-buoy foundations need to be set up at water depth no less than 100 metres. Installation and transportation are not easy, because of their big dimensions and weights. In this case the wind turbines installation must be carried out offshore, using lifting vessels and other maritime engineering equipment. The most critical aspect is represented by the bending moment insisting on the tower base, which is significantly higher than on other types of platforms. Spar-buoy floating foundation is one of the most mature technologies. An application of this technology can be found in Hywind Scotland, the world first MW-scale commercial wind farm.
- **Tension Leg platform, TLP**: Using mooring lines to create tension, this kind of foundations is the most stable. Due to their smaller sizes and lighter weights, these foundations are easier to manufact, assemble install and maintain. Assembling can be performed onshore before dragging them to the wind farm. However, to create tension using mooring lines, special vessels and demanding installation and maintenance procedures are required. Additionally, only some types of seabed are suitable for their anchors. Although TLP is one of the most popular offshore platform types, it is less mature than spar-buoy and semi-submersible turbines and it is still in the demonstration stage.
- **Semi-submersible**: Several buoys and spars are connected by semi-submersible foundations. The former give buoyancy, and the latter, together with mooring lines fixing foundations at a specific position, provide stability. Semi-submersible foundations are the most expensive as they are a complex and huge structure requiring many connectors. To ensure steadiness, dynamic stability system and ballast system are needed. Wind turbines can be assembled to their foundation before being dragged to the desired location. The process can be completed using standard port lifting equipment. Semi-submersible foundations are most suitable for waters measuring between 40 to thousands meters in depths. Semi-submersible is a fully developed technology already set in place. Nevertheless, lowering manufacturing costs poses an actual challenge.

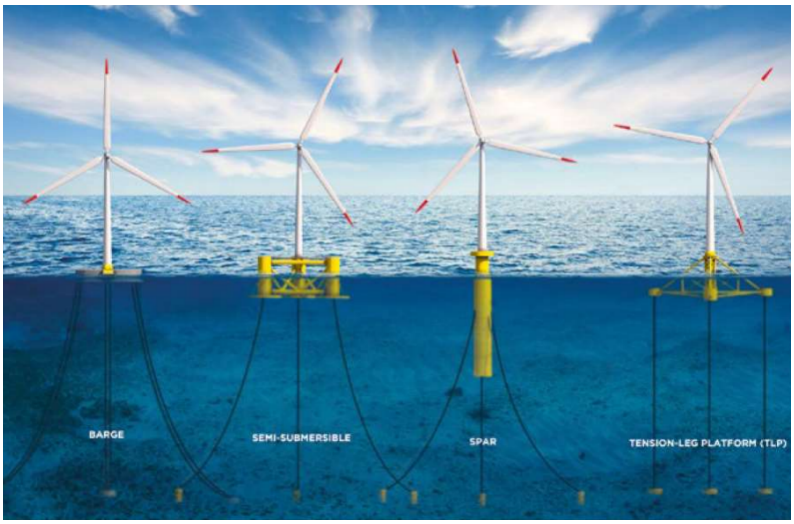


Figure 1.6: Floating platform configurations [1]

Focusing on the aerodynamic aspects, offshore machines experience six additional degrees of freedom (DOF) with respect to the fixed bottom solutions. These movements come from the sea waves, coupled with the incoming wind; they obviously have an impact on the blades unsteady aerodynamics performances and therefore on the power produced. The DOF can be distinguished in three translational (surge X, sway Y, and heave Z) and three rotational (roll RotX, pitch RotY, and yaw RotZ) as reported in Fig. 1.7.

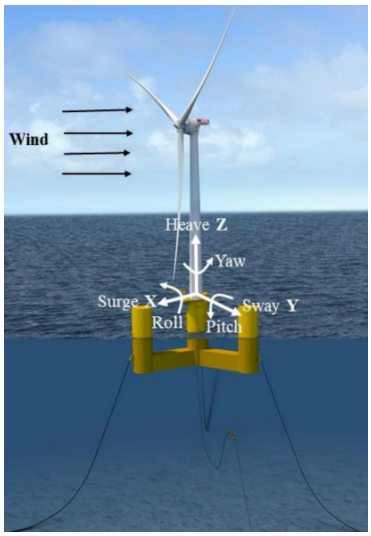


Figure 1.7: Six DOF for a floating platform [5]

Currently, a fast computation model for the unsteady aerodynamic study is one of the most important research objects. The most commonly used aerodynamic analysis techniques for wind turbines are the classical blade element momentum (BEM) theory and the fully

resolved computational fluid dynamics (CFD) method. As will be further explained in the following paragraphs, both methods have some limitations. The BEM theory introduces strong limits due to the severe simplifying assumptions while the complete CFD analysis has computational cost limits due to the small scales that must be resolved to model the boundary layer region. In this context, new approaches, with less computationally demand but good level of fidelity, are required. The need of a specific method for the study of a wind turbine aerodynamic in any possible working condition has led to the development of an innovative approach. In the early 2000s Sørensen and Shen in [35] introduced the basis for a new promising method involving CFD capabilities for accurately estimating the flow field: the Actuator Line (AL) model. AL methods consist in schematizing the wind turbine blades by means of actuator forces. These forces are then applied, in the corresponding mesh location, as source terms of the Navier-Stokes equations before solving the flow field. This approach leads, as a final result, to the replacement of the turbine with a force field rotating in space, providing an accurate resolution of the interaction between the turbine blades and wind. The main advantage of ALM is a strong reduction of the implementation effort associated to a fully resolved CFD simulation. Since the boundary layer resolution on the blade surfaces is not actually necessary, the amount of computer resources needed is significantly reduced. A complete and detailed description of this method is reported in the next sections.

1.2. OC6 phase III

The idea behind this thesis comes from the analysis of the results obtained in the context of the OC6 project. The acronym stands for Offshore Code Comparison Collaboration, Continuation, with Correlation and un-Certainty. It is a research project run under the International Energy Agency Wind Technology Collaboration Program (IEA WIND) Task 30, coordinated by NREL (National Renewable Energy Laboratory). The study aim is to move forward with the validation and comparison of academic in-house codes for the analysis of loads acting on a floating wind turbine under both steady and unsteady operating conditions. Validation is performed by comparing three tools: engineering-level models, such as the one presented in this work, high-fidelity numerical simulations and experimental data. The results have to work as guidelines for the development of future models and test campaigns. Phase III of the project, which has just been completed, was oriented to the pure aerodynamic load inspection. The participation by the Politecnico di Milano took place through the realization of the experimental campaign and the validation of the currently available in-house code.

By comparing the results provided by the original version of the code with those of the

other participants, some differences were identified. The main problem consists in an overestimation of the angle of attack which results in an overestimation of thrust and torque. From these considerations arose the desire to make some modifications to the code to improve the estimate of the forces and to make it more suitable for the analysis of turbines in unsteady conditions.

1.3. Thesis Outline and Scope

The present thesis is divided into the next ten chapters that address different topics:

2. Literature review: This chapter first presents the main techniques adopted for the numerical study of wind turbines. Among these, a more detailed description of the actuator line model is provided, focusing the attention on the different velocity sampling strategies currently available in literature. Finally, a presentation of the EVM model in its original version is reported.

3. Experimental set up definition: this chapter has the aim of clarifying the procedures with which the experimental campaigns, used to validate the numerical results, have been performed. A description of the turbine model and the load cases (LCs) definition is provided.

4. Governing equations, turbulence modelling, simulation initial and boundary conditions, numerical schemes and solvers adopted are described in details. After that the computational domain is described in all its characteristics.

5. Some more information about the simulation set-up are provided regarding the blades modelling, the force distribution and the velocity sampling strategy.

6. This chapter aims to retrace the main phases of the work carried out before reaching the definitive versions of the sampling methods. Some of the previous sampling attempts are analyzed to highlight the main problems and the consequent solutions that have been adopted.

7. The parameters definition of the final three sampling methods is here reported. Their main aspects and physical foundations are described.

8. Validation in fixed-bottom case. The results provided by the three strategies are analyzed and compared with the experiments and with other OC6 simulations. The integral loads, the blade aerodynamics and the comparison between the code outputs and the flow filed are described.

9. Validation in surge platform motion case. The results of the tested strategies are

validated focusing on the integral loads, blade aerodynamics and sampled velocity. Also in this case the validation is mainly performed using the available experimental and OC6 data.

10. Validation in pitch platform motion case. Once again the integral loads, blade aerodynamics and sampled velocity are assessed for all the new sampling methods.

11. Conclusions and further developments: once the objectives achieved have been defined, the strength and weak points of the implemented methods are described. Subsequently, possible improvements and developments of the strategies are proposed together with possible future analyzes which allow to confirm the validity of the proposed methods.

2 | Literature Review

2.1. Turbine numerical simulation

This chapter proposes an overview of HAWT simulation models, from the classical BEM method to the advanced CFD simulations, pointing the focus on body force methods like the Actuator Line one. The floating foundations of modern wind turbines drastically reduce the stiffness of the tower, so that significantly larger system motions can be expected. The environment in which these wind turbines operate is unsteady by nature and consequently the aerodynamic loads on the blades. Usually, unsteady aerodynamic effects can be divided into two groups [12]: unsteady profile aerodynamics and dynamic inflow effects. The former accounts for the dependence of sectional aerodynamic properties on the time-varying angle of attack. Time scales associated with these dynamic effects are proportional to c/Ωr, typically fractions of a second. Dynamic inflow effects account for the influence of time-varying shed vorticity on the induced velocity at the rotor plane with typical time scales determined by the ratio D/V_∞ in the order of 10s of seconds. Considering a floating wind turbine, the typical natural periods for platform motions are in the order of approximately 5 to 100 s. Therefore only the dynamic inflow effect has time scales of comparable order of magnitude as the platform motions. An essential point is to understand if currently available modelling methods are able to predict the loads and the machine performance associated to these conditions. [12]

2.1.1. Analytical methods

Blade Element Momentum method (BEM)

The first and most common strategy, used for HAWT load and performance calculations, is the Blade Element Momentum (BEM) method proposed by Glauert in 1935. This approach divides the flow into annular control volumes that span a large distance upstream and downstream. In each annulus the momentum balance and energy conservation are applied. The free wind velocity U_∞ is defined as the wind speed if there were no turbine present. The effective flow velocity at a specific position of the blade is composed by

this incoming flow field and the so-called induction velocity. The latter is the velocity change related to the pressure field created by the presence of the rotor. In this traditional method the basic assumptions are [34]:

- complete independence between different annulii, induction velocities used at a certain radial station depend only on the local aerodynamic forces at that same station;
- the radial independence principle is still used also for yawed flow conditions.
- two-dimensional flow over the blades, although corrected for 3D effects that are attributable to rotation.

These assumptions are not completely correct, but reduce the complexity of the problem by an order of magnitude making the method very fast to run on a computer. For this reason BEM has gained an enormous popularity allowing the model to be used in design iterations. Due to the limitations of representing all the various real flow situations (dynamic inflow, yaw misalignment, tip loss and heavily loaded rotors), different empirical corrections have been applied. This requires an increasing need to establish experiments and to develop more advanced models to evaluate the basic assumptions underlying the BEM model [35].
Moreover, since BEM approach estimates the induced velocities at the rotor plane thanks to the hypothesis of equilibrium between aerodynamic forces and induced flow field, the instantaneous momentum equilibrium will never be satisfied if the turbine is subject to dynamic motion such as surge or pitch. In case of surge, the dynamics of this process can be only modelled by means of engineering type dynamic inflow models. These engineering models are all dependent on modelling assumptions and their implementation inside the traditional BEM method may potentially introduce significant uncertainties. Furthermore, engineering models were derived and calibrated on the basis of experiment where loads were rapidly changed through pitch regulation on the blades. The source of dynamic loading on a rotor oscillating in surge is different with respect to what the dynamic wake models were originally intended for [12].
The need to overcome these limitations has led to more advanced inviscid models.

Vortex Wake Method

In this method the induced velocity field is computed starting from the shed vorticity in the wake. The vorticity can be distributed as vortex line elements or as discrete vortexes, with vortex distributions determined either as a prescribed wake or a free wake [35]. The former has fast computation of the results, but can give poor insight of the physical phenomenon, therefore it is rarely used. The latter may enforce a wake description closer to reality and provide all relevant information needed to understand the physics of the

wake. However, the free wake analysis is not so diffused for some different reasons: first of all it takes as initial assumption the flow to be inviscid and irrotational which is a huge limitation to study unsteady conditions; it can be very computing demanding and tends to diverge because of the singular velocity distribution of the vortexes employed; in addition, no vortex dissipation is considered making this method unreliable for describing far-wake phenomena. Therefore, although suitable for engineering design phase, its reliability has been proven only for quasi-steady conditions; in case of strong unsteady surge or pitch of the tower its application may become unreliable and it is still under investigation.

2.1.2. Actuator Methods

Another class of methods is the one that makes use of the computational fluid dynamic (CFD). As already mentioned, a fully resolved CFD analysis for the study of an HAWT is feasible but it requires an extremely high computational effort. Usually the complete CFD study is performed only for small time intervals with the target of merely validating simple models. In order to exploit anyway the high fidelity of a CFD analysis without solving all the flow features, some actuator-methods have been developed. They inherit the actuator disk theory from Betz and reproduce the turbine as volume forces distributed in different ways.

Actuator Disk Model

The actuator disk model represents any rotor by a distributed action over the flow without taking into account the complex, vortical structures that a real rotor generates in the flow. The axial force is applied on the disc surface depending on the wind speed related to the thrust coefficient curve of the aerogenerator. The power extraction and wind velocity deficit are applied to the streamlines that pass through the disc. The wind speed on the rotor U is lower than the undisturbed speed U_∞ applying the axial induction factor:

$$U = (1 - a)U_\infty \tag{2.1}$$

The thrust coefficient for the whole rotor C_T is defined as:

$$C_T = \frac{T}{0.5\rho U_\infty^2 A} \tag{2.2}$$

where T is the thrust force, ρ is the air density, U_∞ is the free stream velocity at the hub height and A is the rotor swept area. In this way it is possible to estimate the force that has to be applied and simulated within the CFD-RANS solution. The force is applied on the circular surface that represents the rotor on the computational grid (Fig. 2.1).

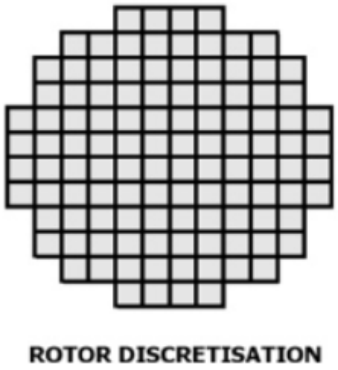


Figure 2.1: AD discretization over a Cartesian structured grid [9]

The force can be distributed in three different ways [9]:

- uniform distribution: it is the easiest method and gives results that can be directly compared with the analytical solution;
- parabolic or polynomial distribution: it helps to represent the blade-tip and hub losses;

Consequently, the real wind distribution and wake extension can be evaluated directly after the numerical CFD iterations. Thanks to its simplicity, it leads to a lower computational effort and, if used on a Cartesian grid, it can be easily integrated with an atmospheric boundary layer [9]. It is often a useful tool to reproduce wind deficits and wake losses and to model multiple wakes in large wind farm. The model can give a reliable description of the far wake and its interaction. There is also the possibility to implement the rotation of the disc to help the modelling of the wake swirl [37]. The drawback is that, especially if used with RANS (Reynolds Average Navier Stokes) solutions and the standard k–ϵ turbulence model, it can under predict the near wake effects, especially if turbulence model are not properly set. Another big limitation of the method is that the forces are distributed uniformly on the actuator disk, hence the influence of the blades is applied as an integrated quantity along the azimuthal direction. Due to its lack of reliability in describing the peculiar vortical structures of a single wind turbine, this model is used for generalized losses at a turbines array-level, not for an exhaustive wake simulation.

Actuator Line Model (ALM)

The actuator line model is a new approach that combines a three dimensional Navier-Stokes solver with a technique in which blades are reproduced through a series of actuator points, each of them consisting in a source force application. In this way, the presence of discrete blades allows the simulation of wake vortexes, aerodynamic effects and the

influence of the rotating blades on the flow field. The loading on each blade is represented by using tabulated airfoil data. The airfoil data and subsequent loading are determined iteratively by computing local angles of attack from the movement of the blades and the local flow field.

Thanks to its good trade-off between computational demand and results quality, the method applied in this work belongs to this family; therefore, a more detailed explanation on the necessary steps for the AL algorithm are deepened in section 2.2.

Actuator Surface Model (ASM)
This model uses the same principle of the AL method. The loading is both distributed along the span of the rotor blades and along the airfoil chord in each cross section [30]. Instead of applying the volume force on a single actuator point for each blade station, the force is distributed chord-wise. Forces are spread on a surface basing on the airfoil pressure distribution. Given the discontinuity generated by the presence of the surface, this method is able to more correctly predict the flow structure near the blade and in the tip vortex region. The main limit of this approach shows up when it is applied to small airfoil chords, since the solution tends to present singularities. As a drawback, the AS model requires an higher computational effort with respect to the previous models. This is the reason why this kind of approach still has quite limited applications.

2.2. Actuator Line Algorithm

The ALM was introduced by Sørensen at al. in [35]. Although the first version of the model was already a good starting point, in the following 20 years continuous improvements were applied, especially in the choice of the main parameters that underlie it. This thesis work aims to make a new improvement to further increase the accuracy of the method in estimating the machine performance. Before explaining the proposal modification, the basic features and the most valuable and recent developments of such method are exposed. As previously anticipated, the Actuator Line Model reproduces the presence of the turbine blades using distributed forces implemented as source term in the Navier-Stokes momentum equations. The rotating blade forces are estimated from the local inflow conditions, based on input tabulated 2D profile polar data. The model requires the use of a CFD-based approach to solve the corresponding flow field. Using a specific sampling strategy the relative velocity magnitude and angle are evaluated and the generated lift and drag are estimated by means of look up polar tables. The key point of the problem is the correct definition of the wind incidence angle and wind speed near the rotor that is responsible for the load on the blade. Different sampling approaches are now

available in literature and one of the most promising seems to be the Effective Velocity Method that is the one on which this work is focused, as deepened in 2.2.3. This method allows the detailed study of the wake and the tip vortexes dynamics and their influence on the induced velocities in the rotor plane. Since the actual geometry of the turbine blade is not numerically resolved, the overall grid density of the simulation, together with the corresponding computational cost, is notably reduced. This is one of the most significant advantage of this method.

In its most general formulation, the ALM algorithm can be synthetically described as follows. The local absolute velocity **U** is extracted directly from the resolved CFD flow field, according to selected velocity sampling strategy. Fig. 2.2 shows a cross-sectional element at radius r that defines the airfoil in the (tg,ax) plane.

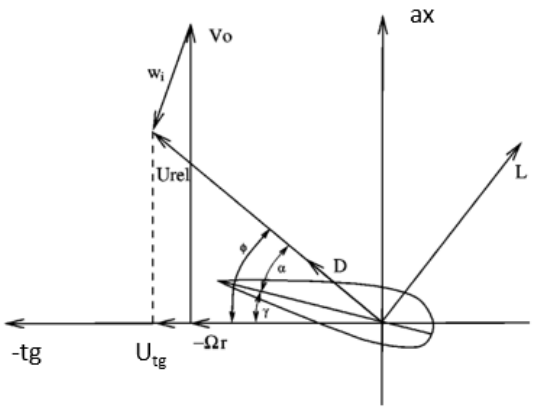


Figure 2.2: Cross-sectional airfoil element [35].

The blade relative velocity U_{rel} is obtained by combining the absolute one and the blade peripheral speed Ωr as reported in Eq. (2.3) :

$$\mathbf{U_{rel}} = (U_{tg} - \Omega r, U_{ax}) \tag{2.3}$$

The relative velocity is then used to evaluate the blade angle of attack α and chord-based Reynolds number Re, as in Eqs. (2.4) (2.5):

$$\alpha = \phi - \gamma \tag{2.4}$$

$$Re = \frac{\rho U_{rel} c}{\mu} \tag{2.5}$$

where $\phi = \arctan(\frac{U_{ax}}{(\Omega r - U_{tg})})$ is the angle between U_{rel} and the rotor plane, γ is the local pitch angle, c is the blade chord, ρ the free-stream flow density and μ the free-stream dynamic viscosity. Such information is used to compute the airfoil lift (C_l) and drag (C_d) coefficients via linear interpolation of the tabulated aerodynamic data given as input to the code.
The lift, the drag and the overall force can be computed as:

$$\begin{aligned} \mathbf{L} &= 0.5\rho U_{rel}^2 ctC_l(\alpha, Re)\mathbf{e_L} \\ \mathbf{D} &= 0.5\rho U_{rel}^2 ctC_d(\alpha, Re)\mathbf{e_D} \\ \boldsymbol{f_{2D}} &= \mathbf{L} + \mathbf{D} \end{aligned} \tag{2.6}$$

Where t is the length of line segment belonging to cell, e_L and e_D denote the versors in the directions of the lift and the drag, respectively. The direct insertion of this force into the mesh cell intersected by the actuator line would cause a singularity in the solution, with the corresponding instability, numerical oscillations issues and consequent numerical divergence. The applied aerodynamic blade forces need to be distributed smoothly on several mesh points in order to avoid singular behavior. This requires the application of a regularization kernel function considering that, in reality, the lift and drag forces are distributed according to the pressure field. The final force is formed by taking the convolution of the computed normal load, f_{2D}, and a regularization kernel, η_{ker} , as shown in Eq. (2.7):

$$\boldsymbol{f} = \boldsymbol{f_{2D}} \otimes \eta_{\mathbf{ker}} \tag{2.7}$$

This force is finally inserted in the Navier-Stokes equation to introduce the blade aerodynamics:

$$\frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{U} \cdot \boldsymbol{\nabla} \boldsymbol{U} = -\boldsymbol{\nabla} p + \nu \boldsymbol{\nabla}^2 \boldsymbol{U} + \boldsymbol{f} \tag{2.8}$$

The regularization kernel shape may vary depending on the study. Many sources are in good agreement in the choice of using a Gaussian distribution, anyway the literature studies are divided in those which prefer a 2D formulation whereas others a 3D one. Overall there are two key points to ensure a robust and reliable solver, the method for sampling the correct velocity magnitude and AoA together with a proper choice of the smearing function.

2.2.1. Volume force distribution

As mentioned in various earlier papers [35] [36] the aerodynamic forces need to be distributed smoothly on several mesh points in order to prevent singular behaviour. This is realized through the convolution integral in 2.7, which spreads the volume forces reducing the gradients. Most studies adhere to Srensen and Shen's original strategy [35], employing a Gaussian smearing function because it is simple to customise and only affected by the Gaussian shape parameter ϵ. In the literature, studies are divided between those that use a 2D distribution (2.9) and those that use a 3D distribution (2.10):

$$\eta(r) = \frac{1}{\pi\epsilon^2} exp(-\frac{d^2}{\epsilon^2}) \tag{2.9}$$

$$\eta(r) = \frac{1}{\pi^{\frac{3}{2}}\epsilon^3} exp(-\frac{d^2}{\epsilon^2}) \tag{2.10}$$

Where d is the distance from the force center point. There are advantages and disadvantages for both the two distributions. Indeed, although the fact that the overlapping of many Gaussian bells in a three-dimensional shape assures the absence of discontinuities along the radial direction, there are problems with limiting the smearing action at the tip leading to inaccurate load predictions. For this reason, when using 3D Gaussian functions, it is imperative to carefully design the tip-losses with corrective factors. In order to achieve a consistent force distribution along the blade, the aforementioned overlapping needs also be carefully constructed. All of these elements increase complexity that favour a 2D approach. Another benefit of using a 2D Gaussian distribution is that the end of the blade has a significant force distribution discontinuity, which creates vortical structures with a nature that is quite similar to the real tip vortexes. Consequently, an AL model accurately simulates the physical phenomenon of wake-induced drag. Nevertheless, this approach does not consider the loss in aerodynamic load in the tip region, due to the finite length of the blade. This lack is due to the fact that modeling each actuator point using 2D profile polars is not able to take into account the radial position. In conclusion, the 2D Gaussian approach is able to accurately consider tip vortexes induced drag but not to take into account the aerodynamic load loss towards the tip. This leads to an overestimation of the blade efficiency. To correct this problem, some specific tip-losses factors should be implemented. Traditional approaches, such as BEM, usually expect the use of tip-loss factors that include both the tip vortexes induced drag and the load reduction (Prandtl or Shen corrective factors). In the case of the AL model, only the latter effect should be implemented using the corrective factors otherwise the losses will be overestimated. The introduction of these parameters increases the complexity of the method therefore, in this work no corrective factor is applied since the application of a 2D regularization kernel is

sufficient to properly simulate the discontinuities at blade end.

In order to obtain reliable results, a correct radius of the body-force projection function (ϵ) and a proper grid spacing (Δ) along the actuator line must be selected. The optimal Gaussian radius is still a matter of debate. On one side, the radius should be as tiny as it can be to get a highly compact representation of the force distribution along the actuator line and so more accurately approximate a line force. Another viewpoint holds that the width should distribute the force throughout a space equivalent to the actual force; on a real blade, the force is distributed both radially and chordwise. Numerical stability, in general, controls the lower threshold for ϵ. According to Troldborg [36], ϵ/Δ=2 should be selected along the actuator line as a compromise between numerical stability and reliable turbine power prediction. When a constant Gaussian radius is applied along the blade (ϵ/Δ=const), the grid spacing Δ is commonly selected so that there are between 30 and 60 grid points along the actuator line [16]. The turbine power predicted by the ALM is, indeed, quite sensitive to the Gaussian radius and the mesh refinement, according to Martnez et al [20]. They discovered that as the grid is refined, the computed rotor power converges for a given inflow wind speed and a certain ϵ. Nevertheless, varying ϵ/Δ from 2 to 10.5, the predicted power undergoes to a significant increase of about 25%. It is also known that blade tip loads are over-predicted when the Gaussian radius is constant along the actuator line. Applying a tip-loss function would help solve this issue. However, choosing a constant ϵ does not guarantee that the forces are spread over a physically significant length scale. Shives and Crawforde [33] sustain that ϵ should be selected to disperse the applied loading over a spatial area that is consistent with the physical airfoil under study. The mesh should then be defined to properly resolve the resulting flow field and Gaussian force distribution. In particular, it is requested that Gaussian radius should be a quarter of the chord (ϵ = c/4), and the mesh spacing should satisfy a resolution of $\Delta = \epsilon/4$. These restrictions ask for a mesh refinement of 1/16 of the chord to resolve the rapidly varying velocity near the center of the loading distribution. This results in an excessively refined mesh discretization but it makes the actuator line method able to model more complex flows, such as turbine wakes, with greater assurance. Jha et al., resume this strategy in [16], noting how it is advantageous to have a regularisation kernel that varies with the radial position in an effort to approximate the blade shape. It was firstly demonstrated that choosing a grid-based Gaussian radius ϵ/Δ=const leads to an over prediction of blade tip loads which means that rotor power and thrust are not always accurate. The apparent problem of over predicting blade tip loads is not completely resolved by a chord-based Gaussian radius ϵ/c=1/4, but this solution leads to some improvement. Jha et al. [16] proposed that the Gaussian radius ϵ

should be chosen based on an elliptic blade planform of the same aspect ratio as the actual blade. By imposing a constant value of ϵ/c^* (where c^* represents the equivalent elliptic planform) along the blade, good comparisons were obtained against measured data and results computed by a BEM-based code. According to this article the $\epsilon/c=1/4$ constant force projection reflects the chord distribution and not the load distribution, therefore this method does not work as well as the $\epsilon/c^*=1/4$ constant projection.

2.2.2. Velocity sampling

As anticipated the crucial phase of the AL model is the correct evaluation of the angle of attack (AoA) from the CFD solved flow field. This operation is not trivial because it requires to compute a single incidence value from a three-dimensional velocity distribution. This operation is complicated by the presence of a bound circulation associated to lift production, which deforms the flow field in the plane normal to the blade axis (Fig. 2.3).

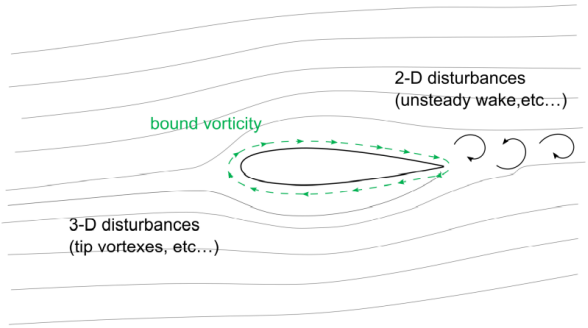


Figure 2.3: Schematic representation of bound circulation associated to lift production [22].

The circulation effect needs to be filtered out with specific techniques, without removing however other phenomena like those related to wake unsteadiness or inflow turbulence. Therefore, the sampling method should not detect the blade-local flow effects but should take into account the severe slowdown of the flow in the rotor axial induction zone, because that is the freestream to which the blades are exposed. Since the AoA is substantially a 2D concept, its evaluation is more challenging when three-dimensional analysis are performed. For HAWT the main issues are located at the root and tip sections of the blade where 3D flow structures are concentrated. The 3D effects produce a chordwise component of the bound circulation that is not usually evaluated by the different sampling strategies because they work only on the 2D airfoil plane. Another problem is related to the fact that the sampling phase is usually performed on a limited number of points near the airfoil, this makes the AoA very sensitive to the solved flow field, (e.g., numeric noise,

vortices, local errors, etc...) [22]. Some studies have proposed the application of ad hoc filtering strategies to reduce the oscillations of the solution. In particular, in the work of Shamsoddin and Porté-Agel [29] the AoA history from the previous rotor revolution is filtered with an eight-order polynomial before being applied as an input for the ALM algorithm.
Different methods have been developed over the years for the extraction of the angle of attack from the numerical flow field obtained from CFD simulation. Most of the researchers have proposed methods based on the use of one or several sampling points to collect information about the velocity field around the airfoil. In this section the main sampling methods available in literature are exposed starting from those using a single sampling point.

Shen methods

Shen proposed two different sampling methods that are both based on the evaluation of the velocity in one single point and corrected for bound vorticity:

- **Shen1**: The strategy was introduced in [32]. It is expected that the force distribution along the blade and the velocity field nearby have been determined using CFD calculations or experiments. In this technique the bound circulation around the airfoil section is represented by a point vortex which typically resides in the quarter chord point of the airfoil. Because of the error introduced by a point vortex instead of the true circulation distribution, the monitor point should not be too close to the airfoil. Therefore the sampling point is placed in the rotor plane in front of the airfoil section as reported in Fig. 2.4.

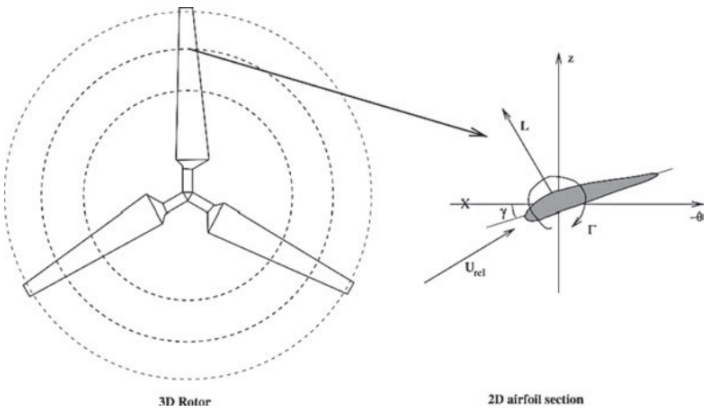


Figure 2.4: Transformation from 3D rotor into 2D airfoil sections. 'X' denotes the monitor point [31].

The AoA is determined by estimating first of all the lift force by the force projec-

tion along the incoming direction and the normal one; following that, the bound vortex is computed using the Kutta-Joukowski law; the induced velocity is obtained by the bound vortex using Biot-Savart law; the effective velocity is calculated at the sampling point by subtracting the induced velocity from the sampled one; the relative velocity is computed combining the absolute velocity with the peripheral one; eventually the AoA is calculated from the relative velocity [25]. This procedure is repeated until the convergence is achieved. In this method the bound vortex is considered as a line vortex along the blade, therefore the position of the sampling point is relevant. This method suffers from singularity issues when the sampling point is too close to the blade.

- **Shen2**: An alternative method was proposed in [31] to address the singularity problems of the prior method. In this study, instead of the concentrated bound vortex at the force center, a distributed bound circulation along the airfoil surface is used. With this modification, the monitor point can be located closer to the blade. It is demonstrated that for this method the results are not depending on the location of the sampling point [25]. As an advantage, the chordwise variation of aerodynamic forces, which is ignored when the vorticity is concentrated, can be taken into account. Additionally, this method is not iterative. Nevertheless, finding the separation point (SP), where the local circulation shifts sign, is challenging [31].

Sampling on the AL point

In the case of CFD simulations using the AL method another point sampling is usually adopted. For isotropic Gaussian actuator line force projection functions, the bound vortex cross section is circular. As a result, a lot of actuator-line users sample the velocity on the actuator point ([35], [20]) which is also the location of the bound vortex centre that is free of the influences of the blade-local flow. Martinez-Tossas et al. [21] indicate that sampling at the center of the bound vortex should give the correct freestream velocity. Nevertheless, in situations with severe drag, this velocity needs to be corrected.

In addition to these methods, more sophisticated techniques, such the 3-Points, the LineAverage, the Average Azimuthal method, the Herraèz method and the Effective Velocity Method (EVM), make use of numerous points that are conveniently placed around the turbine blade to filter out spurious aerodynamic disturbances from the flow field.

Three-points method

The 3-Points approach, which was first introduced by Rahimi et al. [24] to analyze the behavior of horizontal-axis turbines in yawed flow conditions, requires a total of six points,

three on the suction side and three on the pressure side of the airfoil. The key benefit is its considerable simplicity. Using this strategy, the calculation of the AoA (α) is fairly simple. By using three points, upwash and downwash effects, as well as the influence of bound circulation, are all removed. For each azimuthal position and also at the tip and root of the blade, this method is able to mimic the dynamic behavior of the local induced velocity and the AoA, which is crucial for the yawed flow. These points are situated at 25, 50 and 75% of chord length in the chord-wise direction as shown in Fig. 2.5. They are placed around 1*chord away in the axial direction from the airfoil, which means that their locations change over the blade span because dependent on the local chord. These locations are supposed to lessen the impact of flow separation and balance out the influence of the bound vortex on the resulting induced velocity.

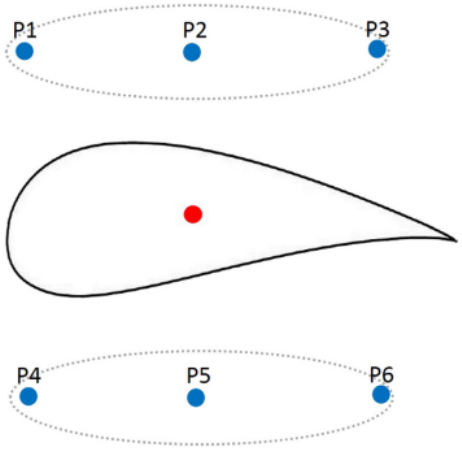


Figure 2.5: Schematic representation of the 3-Point method for the velocity sampling [25].

The extraction of the AoA occurs in two main steps: 1) an ad hoc interpolation function is used to average the velocity for each pair of points at upstream and downstream locations. As a result, the three velocities U_{1-4}, U_{2-5}, U_{3-6} are obtained; 2) the undisturbed velocity vector U is calculated as the arithmetic average of the three velocities coming from each station:

$$U_{averaged} = \frac{U_{1-4} + U_{2-5} + U_{3-6}}{3} \tag{2.11}$$

Line Average method

This method was developed by Jost et al. in [18]. In order to calculate the AoA, the Line average method averages the flow velocities along a symmetric, closed line surrounding the rotor blade. Theoretically, there are many different closed shapes that may be used, but Jost et al. in [18] showed that the circle shape produces the greatest results in terms of AoA sampling accuracy. The circle center is placed at the quarter chord position with

a radius r=1c as reported in Fig. 2.6.

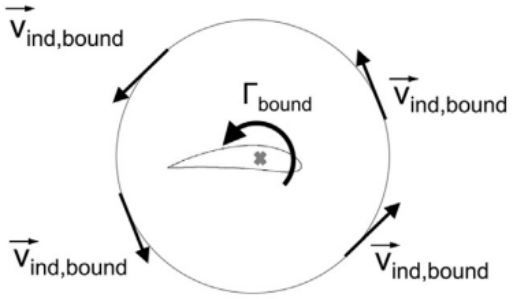


Figure 2.6: Line Average method [25].

According to the theory, the induced velocities at opposite places on the circle cancel each other out. In this way, the influence of bound circulation is eliminated and the local inflow velocity and the AoA are calculated by averaging the flow velocities along the circle.

Average Azimuthal method (AAT)

The inflow velocity is obtained as an annular average along the turbine swept area using several upstream and downstream locations [17]. For each radial location the inflow velocity is computed using two different circles: the upstream and the downstream annulus (Fig. 2.7).

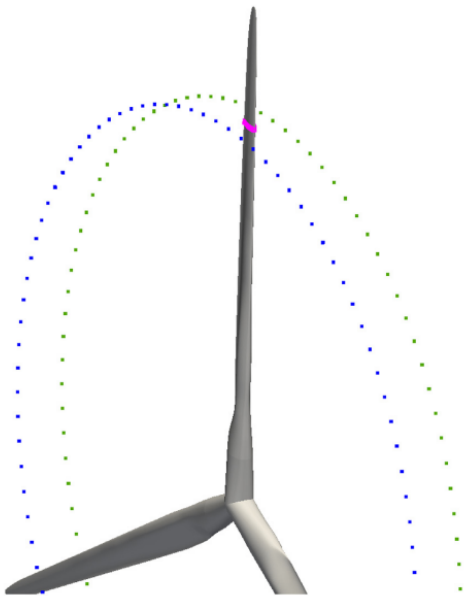


Figure 2.7: Average Azimuthal method [25].

The velocity sampled at each point for each annulus ring is averaged over the ring itself. Subsequently, the velocity at the rotor plane is computed by interpolating the upstream and downstream circle averaged velocities. The drawback is that this method can only accurately represent axial conditions and it can not account for the induced velocity dynamics in the case of yawed flows. Additionally, because the method relies on averaged data, a large number of points must be sampled. This aspect increases the computing cost with respect to alternative approaches. It is also mentioned that the monitor point locations and the interpolation procedure may affect the outcomes. Eventually, in the tip region, the annular averaged induced velocity provided by AAT is known to be different from the local induced velocity.

Herráez method

This method directly measures the velocities at a position in the rotor plane where the influence of each blade bound circulation is counterbalanced by the other blades [13]. In the case of axial and uniform inflow, this position corresponds to the bisectrix of the angle between two consecutive blades. For a 3-bladed wind turbine, the velocity can be sampled along the radial line located 60° ahead and behind an arbitrary blade, as shown in Fig. 2.8

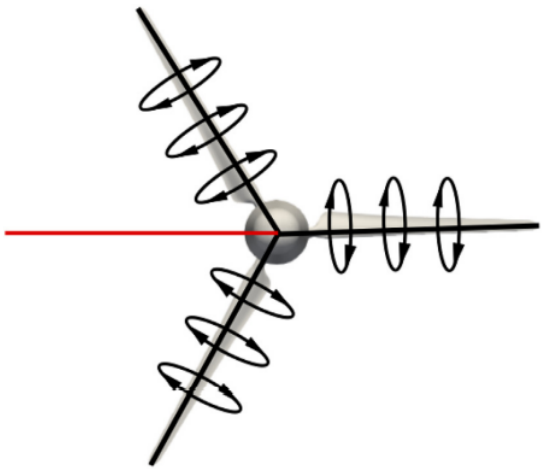


Figure 2.8: Location at which the undisturbed velocity field can be probed[25].

It is unnecessary to adjust for the effects of blade induction in the velocities retrieved from the rotor plane because they are only influenced by the wake induction. In this way, the axial wake induction is determined by the difference between free-stream velocity and the probed axial velocity. The tangential wake induction corresponds to the probed tangential velocity changed in sign. The primary benefit is its semplicity leading to a straightforward calculation of the AoA. Additionally, it is independent from user-defined input parameters

such engineering correction models and monitoring point location. Nevertheless, it is important to keep in mind that also the trailed and shed vorticity, as well as the bound circulation, contribute to the rotor flow non-uniformity. In the case of axial and uniform inflow and stationary turbine operation, no shed vorticity needs to be considered while the trailed vorticty may be non-negligible particularly in the tip region. Therefore, the fundamental drawback of the existing approach is that the trailed vorticity effects can not be adequately captured because the probing site of the velocity field is placed distant from the blade [13]. This approach can still be used in yawed or non-uniform inflow situations, but in this case it becomes significantly more difficult. The evaluation of the location at which the blade-induced velocities are counterbalanced requires calculating the local bound circulation along each blade beforehand.

2.2.3. Effective velocity method (EVM)

The aim of this thesis work is a modification of the sampling velocity method applied in the already available in-house code developed by Schito et al. [28]. Before presenting the new strategies, the previous EVM method, originally implemented in this code, is analyzed in details. One of the most challenging topic in wind turbines study is the correct reproduction of the wake. Nowadays, the wake produced by a wind turbine can be numerically calculated using modern CFD frameworks. In order to achieve this, the proper modeling of the interaction between the blade and the incoming flow is required. The key point of the problem is correctly defining the wind speed and incidence angle that cause the load on the blade. In this way, it is possible to obtain the proper estimation of the aerodynamic forces acting on the blades.
This model is called Effective Velocity Model since it seeks to evaluate from local inflow velocity fields the equivalent undisturbed velocity incident to the airfoil, used to appropriately interpolate the airfoil tabulated polar curve and define the forces acting on the blade section. The idea behind the Effective Velocity Model (EVM) is to make efficient the use of the vast amount of data that CFD provides about the flow around an airfoil. By numerically simulating the flow, it is possible to determine the velocity at any point around a wind turbine blade in order to obtain information about the incoming wind. The main premise of EVM development is that, evaluating AoA and effective velocity through the average on a segment, is more effective than sampling them on a single point. Therefore the suggestion presented by Schito et al. consists in sampling the velocity on a segment positioned perpendicular to the mean relative velocity direction and located at a certain distance upstream with respect to the actuator force (AF) point. The key challenge of EVM is defining the proper distribution of force around the AF point, the

position and the length of the sample segment and assessing the relationship between the sampled flow and the resultant aerodynamic forces. All this parameters are optimized in [28] comparing EVM results with an equivalent fully resolved case for a 2D simulation of a NACA0012 profile at various angles of attack. These simulations had the goal of setting up the EVM so that it could input lift and drag forces that were as similar as possible to those produced by the airfoil profile for every tested angles of attack.

Force distribution - Regularization Kernel factor

As mentioned in Par. 2.2.1, in order to prevent singular behaviour, the AF aerodynamic forces must be evenly distributed among several mesh points. In the original code, and also in the present work, a 2-dimensional normal distribution is applied:

$$\eta(r) = \frac{1}{\sqrt{2}\pi\epsilon^2} exp(-\frac{d^2}{2\epsilon^2}) \tag{2.12}$$

Where d is the distance between cell-centered grid points and the point where the actuator point lies.

This distribution is useful to enhance the solver flexibility and robustness by smoothing the solution. When choosing its radius, special attention must be taken to obtain the tightest distribution possible, which will more closely match the location where the blade forces are applied. In [28] the proper value of ϵ is chosen comparing the velocity magnitude and angle, inside the wake, between the AF simulation and the fully resolved simulation for three ϵ. The values are compared on two different lines placed 2 and 5 chord length behind NACA0012 profile and AF point (Fig. 2.9 and 2.10).

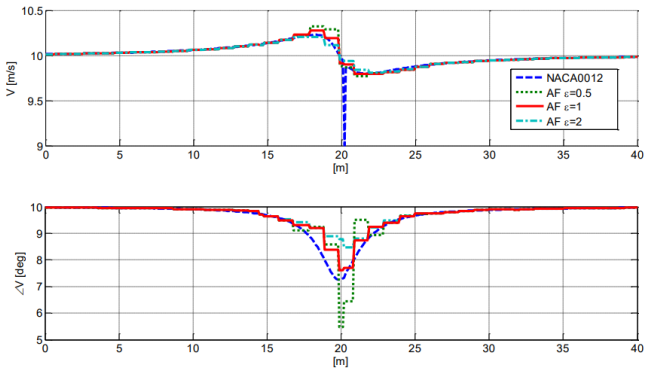


Figure 2.9: Wind velocity (V) and angle (∠V) sampled on a line 2 chords behind the AF point [28].

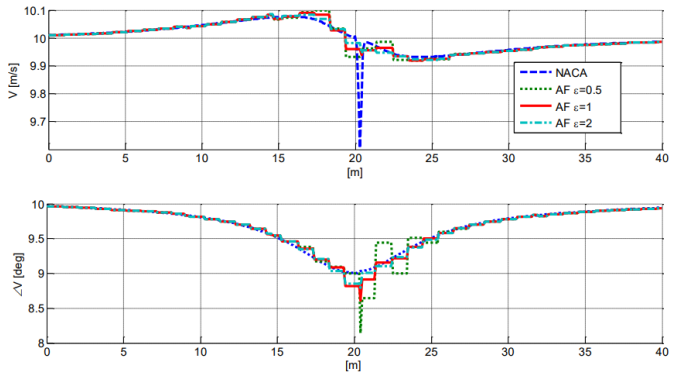


Figure 2.10: Wind velocity (V) and angle ($\angle$V) sampled on a line 5 chords behind the AF point [28].

For smaller ϵ and thinner spatial distribution of force the AF wake reproduces better the local shape of NACA0012 wake. Nevertheless, a spatial oscillation in the velocity and the angle is present for ϵ equal to 0.5. Therefore, ϵ=Δ is selected to prevent this numerical fluctuation (it can be noted that in the study of Schito and Zasso the mesh size Δ is equal to the chord length of the profile).

Step 1 - Position of the sampling line

The setting of EVM is based on the equivalence of the flow fields between the fully resolved simulation and the AF one. As mentioned, the sampled velocity is the average evaluated on a line perpendicular to the mean relative velocity direction and collocated at a certain distance upstream the Actuator Point. In order to define this distance, the analysis focuses on the velocity field in the region upstream of the profile. The best choice is obtain as a compromise between the need of sampling the velocity near the AF point, to evaluate the correct velocity felt by the real profile, and of limiting the relative error in the upstream flow field between the two simulations. The relative errors in magnitude $error_v(s,d)$ and phase $error_{\angle V}(s,d)$ between the velocities, computed on lines at different distance from AF point, have been obtained as:

$$error_v(s,d) = \left| \frac{V_{AF}(s) - V_{NACA}(s)}{V_{NACA}(s)} \right| \tag{2.13}$$

$$error_{\angle V}(s,d) = \left| \frac{\angle V_{AF}(s) - \angle V_{NACA}(s)}{\angle V_{NACA}(s)} \right| \tag{2.14}$$

The best condition is obtained for a distance d=1.5c being the relative error sufficiently limited both for the velocity magnitude and the phase as reported in Fig. 2.11

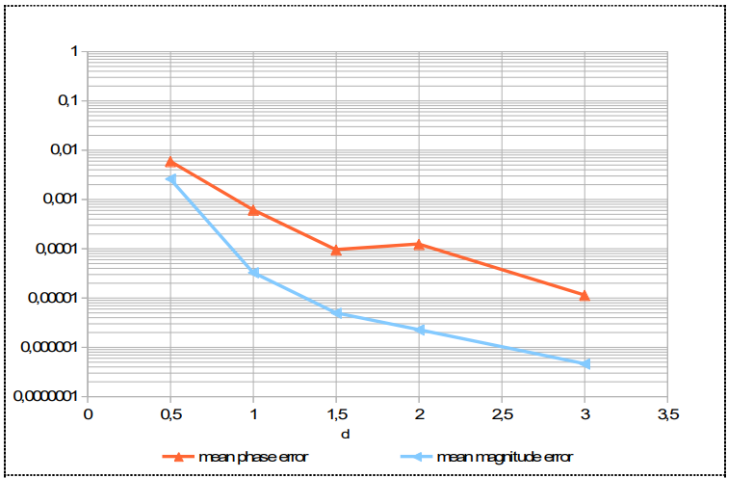


Figure 2.11: Comparison of the U (magnitude and angle) function of the coordinate s, observed on the line placed 1.5c in front of the AF point [28].

Step 2 - Length of the sampling line

Once the position of the sampling line is fixed, its length must be selected. For this purpose another flow comparison is performed between the two simulations. Using a line 1.5c upstream the AF point, the mean velocity angle is computed for each line length q. The absolute difference between the angle in the AF simulation and the one in the fully resolved simulation is reported in Fig. 2.12.

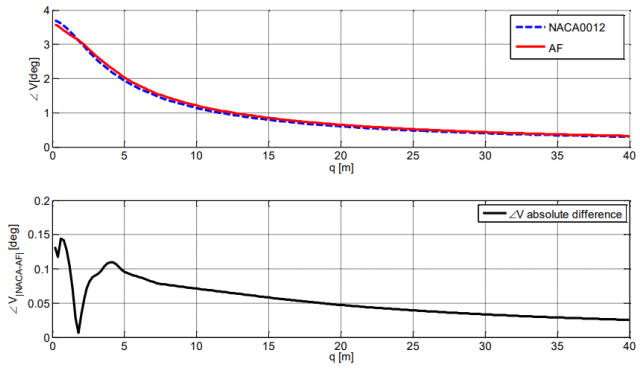


Figure 2.12: Difference in velocity angle between NACA0012 and AF data [28].

The difference between the curves reduces increasing the sampling line length because a long segment considers also information provided by cells distant from the airfoil location, where the flow is undisturbed. The best compromise is obtained for a line length equal to $l_{sampling}$=5Δ.

Step 3 - Attack angle correction

The last stage of the EVM construction aims to define the empirical formula between the angle sampled by the EVM and the undisturbed angle utilised to compute the aerodynamic forces produced by the airfoil. Therefore a correction must be applied to the angle α_{EVM} provided by the EVM method. The correction $\Delta\alpha$ is strictly related to the most important parameters that influence the aerodynamic force:

$$\Delta\alpha = \alpha_{EVM} - \alpha_{\infty} = f(C_l, C_d, c/\Delta) \tag{2.15}$$

Where α_{EVM} is the angle sampled by the EVM strategy, α_{∞} is the effective AoA, C_l and C_d are the lift and drag coefficients and c/Δ is the ratio chord over mesh size. To quantify the $\Delta\alpha$ correction, a large test campaign was performed in [28] from which it was deduced that:

- $\Delta\alpha$ is a linear function of C_l;
- C_d alters the slope of $\Delta\alpha(C_l)$ line;
- $\Delta\alpha$ is a linear function of c/Δ.

Therefore the final expression of the mentioned correction is:

$$\Delta\alpha = \frac{c}{\Delta}(1.2553 - 0.0552C_d)C_l \tag{2.16}$$

The EVM method does not require a correction for the absolute value of V, since the presence of profile induces only a deflection on the incoming flow. The main advantages of this method are the strong robustness, the possibility of using the angle correction for different profiles (because all the parameters are non dimensional) and the capability of well reproducing not only the blade loads but also the wake.

Nevertheless some problems have been identified and have motivated the realization of the present work. First of all, the empirical correction on the attack angle may not be adequate to estimate the correct loads in case of unsteady working conditions. Indeed, this formula is obtained from a test campaign carried out in specific steady conditions. In case of surge or pitch motion, the estimation of the angle of attack could be inaccurate therefore a more general sampling method only based on the CFD flow field is preferred. Moreover, some discrepancies from other participants results have been found during the third phase of the OC6 project [6]. In particular, a small overestimation of the attack angle results in too large thrust and torque in the steady and unsteady cases.

Another aspect of the original EVM that has been questioned concerns the characteristic quantity chosen to parameterize the distance and length of the sampling line. In the original code these quantities are defined as multiples of the profile chord for each section along the blade. Since the force is distributed on the mesh with a function of the cell size (Δ), it seems more coherent to also parameterize the velocity sampling as a function of the same cell size. Therefore, having chosen force distribution based on a numerical parameter, it is preferable to follow the same strategy also for the sampling to avoid having some quantities based on numerical parameters and some on physical ones. In doing so, a change in the mesh refinement produces a coherent scaling for the force and the velocity sampling keeping the method intact.
Finally, a consideration about the algorithm used for solving the problem has been made. In the original EVM code, the PISO strategy is adopted for both cases steady and unsteady. Considering that the AL solver advances explicitly in time, a PIMPLE algorithm may be more suitable for sampling the velocity coherently with the actual position of the AL point. More explanations on this aspect are provided in section 4.4.4.
All these reasons have justified the new proposed sampling methods and all the modifications made on the original code.

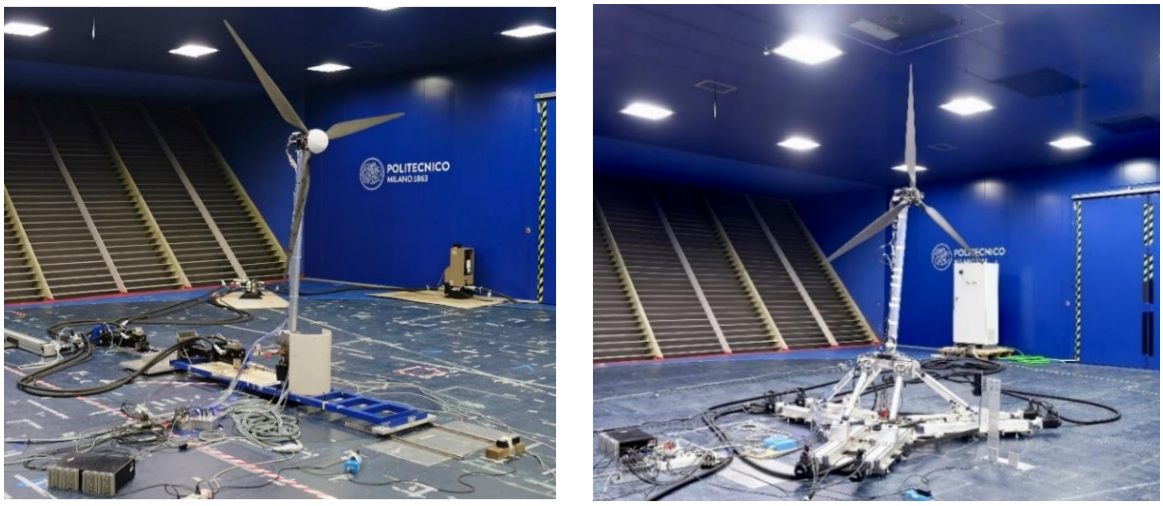
3 | Experiment Set Up and turbine model

The validation of the models proposed in this work is mainly achieved via experiments. Further confirmation of the numerical results is obtained analyzing the improvements as compared to the results of the original EVM model. Moreover, during this work, another validation was performed checking that results are consistent with those presented by other participants to OC6 project who make use of different numerical methods.

3.1. Wind tunnel configuration

Two different experimental campaigns are available for the validation phase, both were carried out at GVPM (Galleria del Vento Politecnico di Milano). The first dataset was developed in the UNAFLOW (UNsteady Aerodynamics for FLOating Wind) project, which was a collaboration between EU-IRPWIND and Politecnico di Milano (Polimi), ECN, University of Stuttgart, and the Danish Technical University (DTU). The second one was conducted in the frame of the OC6 research project.
In this thesis, the UNAFLOW campaign is described as experiment 1 and the OC6 campaign is indicated as experiment 2. Both steady and unsteady wind conditions are tested in experiment 1. Through the use of forced harmonic oscillations, the surge operating conditions were reproduced. Some of the conditions examined in experiment 1 were tested once again during the experiment 2. Additionally, the pitch condition is analyzed thanks to a proper platform motion. In Fig. 3.1 the tested turbine in the Politecnico di Milano wind tunnel is shown for the two campaigns.

The wind gallery facility is a closed-cycle one, with two rows of seven 2x2m independent fans, generating a total power of 1.4 MW. The so-called civil gallery is located in the upper part of the wind tunnel loop and is designed for large-scale tests at moderate speeds, while the high speed gallery is located in the lowest part and is used for studying high speed aerodynamics. The testing campaign of the OC6 project takes part in the upper part. The wind tunnel dimensions are 13.84 m wide x 3.84 m high x 35 m long. During the tests



(a) Experiment 1 [6] (b) Experiment 2 [6]

Figure 3.1: Testing Set-up for UNAFLOW and OC6.

the air density can be considered equal to 1.177 kg/m^3 and the turbulent intensity around 2%. At the inlet an empty configuration is set without obstacles, roughness elements or turbulence generators.

A schematic representation of the system during the two experimental campaigns is reported in Fig. 3.2 and the correspondent geometrical quantities are defined in Tab. 3.1.

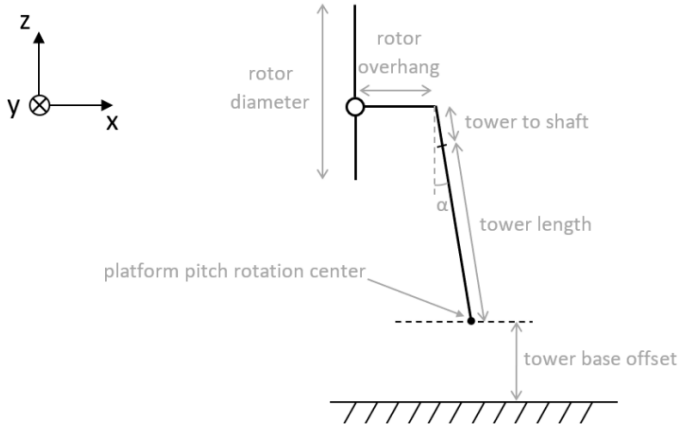


Figure 3.2: Experimental set-up for surge and pitch cases [6].

Parameter	Experiment 1	Experiment 2
Rotor diameter	2.38132 m	
Blade length	1.10166 m	
Hub diameter	0.178 m	
Tilt angle	5°	
Rotor overhang	0.09467 m	0.139 m
Tower to shaft distance	0.03667 m	0.064 m
Tower length	1.6057 m	1.400 m
Tower base offset	0.450 m	0.730 m

Table 3.1: DTU-10 MW RWT turbine parameters [6].

In Experiment 1, two hydraulic actuators at the base of the tower were used to generate the surge forced motion. In this campaign loads were measured with a 6 degree-of-freedom (DOF) load cell at the tower top location, 6-DOF load cell at the tower base, hot-wire probes to capture the wind speed along and across the wake, and particle image velocimetry (PIV) to analyse the blade tip vortex.
Instead of hydraulic actuators, experiment 2 used a 6-DOF robot to impose the base movement. The highest frequency tested was actually lower than the tower natural vibration mode: maximum oscillation performed was at 2 Hz, whereas the tower first fore-aft structural frequency was 6.5 Hz. This allowed to apply a linear model to the system and give permission to remove the inertia forces caused by the mass of the turbine. This leaves only the aerodynamic load as the output of the post-processing data.

3.2. Turbine model

The test turbine is a 1:75 scaled version of the DTU 10-MW RWT ([4]), designed by Polimi within the LIFES50+ project. The thrust (C_T) and power coefficients (C_P) of the full-scale design were intended to be preserved in the turbine scaling technique. The structural scaling was also oriented to preserve the stiffness and mass proportion of the real structure. The wind velocity was scaled by a factor of 3 (moving from a 12 m/s rated velocity to a 4 m/s one) leading to a 225-times reduction in the Reynolds number. In order to respect aerodynamic scaling, low-Reynolds profile were adopted (SD7032) and the chord and twist distributions were altered to achieve the correct thrust forces and also match the torque adequately (Fig. 3.3).

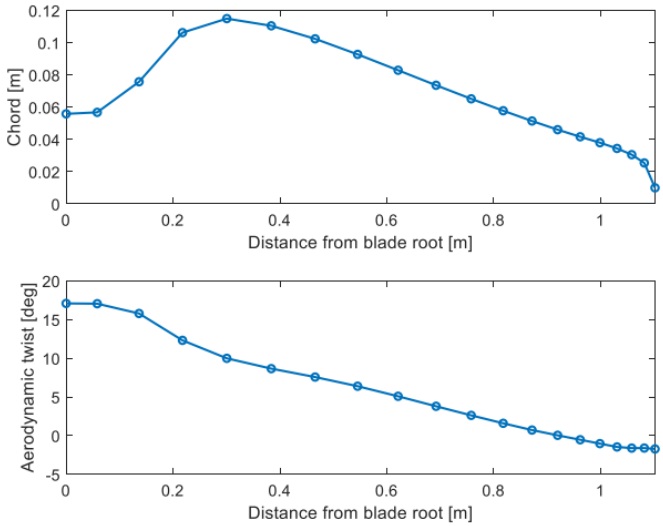


Figure 3.3: Chord and aerodynamic twist along the blade [27].

The airfoils used for the scaled turbine are a 10% thick version of the Selig Database SD7032 airfoil. To remove the challenge of scaling the aeroelastic behaviour, the rotor blades were designed to be rigid. A CAD model of the blade 3-D geometry is reported in Fig. 3.4.

Figure 3.4: 3-D geometry of the model blade [27].

Separate 2D testing was performed to characterize the airfoils and develop the polars for the profiles installed in correspondence of 20 selected radial stations of the turbine blade. C_l was derived from pressure probes values along the selected profile, and the C_d from a rear wake rake. For each of the 20 stations, the polar curves are provided for 7 different Reynolds numbers and various angles of attack up to the profile stall. Some post-processing procedures were applied:

- Extension to the whole $\pm 180°$ angle of attack was done by applying the Viterna method;
- Interpolation is used to generate the polars for the transition sections, when the airfoil progressively passes from the circular section to the SD7032;
- The polars of the single radial stations receive the 3D corrections.

3.3. Load cases definition

The major goal of the experiments was to apply specified motion to the platform in order to simulate unstable aerodynamic phenomena on a FOWT model. Naturally, motions in real-world situations are not predetermined; rather, they are a result of the forces brought on by wave and wind excitation. Amplitude is a complicated function of the shape and mass distribution of the floating platform. To get around problems with the hydro-aeroelastic coupling, which is not the main emphasis of the OC6 Phase III program, forcing movements have been imposed.

Since the OC6 target is the evaluation of loads acting on a floating wind turbine, both in fixed-bottom and unsteady working conditions, the tests performed can be subdivided into two classes: the fixed-bottom and unsteady load cases. In the fixed-bottom condition, three cases with different wind speeds (U_∞) are evaluated: two rated cases (Rated 1 and 2) and one above-rated case. The rated cases were run at a constant rpm (Ω), which corresponds to an optimal tip speed ratio (Λ) and with a zero pith angle (Θ_p). For the above-rated case, a pitch angle of 12.5 degrees was applied and a lower tip speed ratio was imposed to reduce the C_P. However, this work refers only to the Rated 2 case (LC1.1).
The unsteady conditions are performed as a set of surge and pitch platform motions with a combination of motion amplitudes and frequencies.
In the surge case (LC2.5) the movement identifies a mono-harmonic motion aligned with the wind direction. The difference in speed between the hub velocity, produced by the platform motion, and the free stream wind U_∞ is the apparent wind experienced by the rotor. The apparent wind varies throughout the motion, greatly altering the working condition. In the surge case the rotor is completely perpendicular to the wind, therefore no matter the radial or azimuthal position, these variations apply to all blades equally.
A different situation arises when pitching motion is imposed (LC3.5), which introduces variances spanwise for each individual blade as well as from one blade to another.

The tested load cases are summarized in Tab. 3.2. The complete set of information about experiments performed are reported in [27] while [19] provides information about the measurement chain and data filtering.

Case	Type	U_∞ [m/s]	Ω [rpm]	Λ [-]	Θ_p [°]	Amplitude	Frequency [Hz]
LC1.1	steady	4	240	7.1	0	-	-
LC2.5	surge	4	240	7.1	0	0.035 m	1.0
LC3.5	pitch	4	240	7.1	0	1.4°	1.0

Table 3.2: Load cases

To validate the new models some main quantities are taken into account. First of all the integral quantities for the thrust and torque are compared with the experimental ones. These values are obtained by means of an average over some rotor revolutions.
Also the blade aerodynamic response is analyzed considering the trends of local aerodynamic quantities along the blade. For these quantities, experimental measurements are not available but a comparison is performed using the results obtained by the other participants to OC6 project. These loads are computed by first averaging the single blade load over a rotor revolution and then by averaging across the three blades.
In addition, experimental tests provide measurements for the wake analysis because the AL model is a tool capable of reproducing the turbine wake. Nevertheless, the aim of this present work is to study the interaction between the blades and the wind focusing the attention in the region around the rotor where the forces and the velocity field are carefully analyzed, and therefore neglecting the wake region.

4 | Mathematical model and numerics

This chapter objective is to provide a brief review of the key concepts of CFD and its governing equations and turbulence models and to present a brief description of the open-source program OpenFOAM that was used to carry out the simulations for this thesis work.

4.1. Computational Fluid Dynamics and OpenFOAM

Computational fluid dynamics (CFD) is a particularly helpful computer-based tool for the analysis of systems involving fluid movement, heat transport, or other chemical phenomena. The most important benefits of CFD over competing technologies, such as full-scale measurements or Wind Tunnel experiments on scaled models, are:

- significant cost and lead time reduction;
- ability to examine systems where performing controlled tests is challenging or impossible;
- the capacity to research systems in risky environments beyond their typical performance limits;

Every CFD test case is built primarily out of three distinct phases:

- pre-processing phase: it is first necessary to define the computational domain, which entails designing the region of interest while accurately recreating the geometry of the system under study. The second procedure is the mesh generation; it divides the computational domain into numerous non-overlapping sub-domains. In this phase the user must establish the boundary and initial conditions for all the sides of the computational domain, as well as the fluid properties;
- solving phase: the numerical solution of the fluid dynamic equations is the subject of phase two. In order to solve the Navier-Stokes equations, the partial-differential

problem must be discretized over the cells. There are numerous strategies that can be used for this procedure, the finite element (FE), finite difference (FD), and finite volume (FD) strategies. Commercial codes, such as OpenFOAM, typically use the latter one. The fluid behaviour is described by the conservation laws. In the case of FV approach, these equations are expressed in an integral form.
The mathematical problem is solved in three steps: integration of the governing equations on the cells; discretization of the resulting equations into a system of algebraic equations; solution of the system with a proper iterative strategy.
Three properties of the computational mesh must be ensured in order to obtain the exact integral assessment from a numerical point of view: grid orthogonality, skewness, and regularity. Structured grids perfectly fulfil those requirements, but they are not optimal for conforming to complex geometries. On this aspect, the main advantage of the Actuator Line method is evident. Since it does not need to reproduce any surface, the cells can be simple hexaidral volumes completely regular;

- post-processing phase: nowadays, any CFD software comes with a variety of data visualisation options. These allow to display the computational domain, vector and surface plots of interesting quantities.

This thesis is developed using OpenFOAM (Open Field Operation and Manipulation), a widely used free and open-source CFD toolbox. It is used in most areas in engineering and scientific fields from complex flows involving chemical reactions, turbulence and heat transfer, to solid dynamics. OpenFOAM is a framework used to create application executables that make use of packaged functionality found in a group of about 100 C++ libraries. About 250 pre-built programs are available in OpenFOAM. They are divided in: solvers that are created to manage specific fluid mechanics problems and utilities that are created for data manipulation. By being open, OpenFOAM gives users the freedom to alter and increase its current functionality and extend the collection of solvers, utilities and libraries. It uses a very modular structure, where groups of functionality (such as numerical methods, meshing and physical models) are each compiled into a separate shared library. The functionality of the library is then easily linked to in executable applications. OpenFOAM fundamental technology consists of a modular set of effective C++ modules.

4.2. Navier Stokes Equations

The fundamental governing equations for fluid dynamics are the mass, momentum and energy conservation equations. In the case of an incompressible fluid, the density is assumed to be constant therefore it does not depend on the temperature. Starting from

this consideration, the two equations that must be solved are the continuity and the momentum equation. The original expression of the continuity equation is reported in Eq. 4.1. This equation becomes a simple condition on the velocity in the case of an incompressible fluid (Eq. 4.2).

$$\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{U}) = 0 \tag{4.1}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{U} = 0 \tag{4.2}$$

The Newton second law serves as the foundation for the formulation of the momentum equation, which takes into account all of the forces acting on an infinitesimal control volume (volume and surface forces):

$$\frac{\partial \rho \boldsymbol{U}}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{U}\boldsymbol{U}) - \boldsymbol{\nabla} \cdot \mu \boldsymbol{\nabla} \boldsymbol{U} = -\boldsymbol{\nabla}(p) \tag{4.3}$$

This equation is still set up to be written in the C++ language, which is the one used by OpenFOAM. Therefore, Eq. 4.3 can be easily implemented in the OpenFOAM code:

Algorithm 4.1 Momentum Equation implemented in OpenFoam

```
solve
(
fvm::ddt(rho,U)
+ fvm::ddt(phi,U)
- fvm::laplacian(mu,U)
==
- fvc::grad(p)
)
```

4.3. Turbulence modeling

In all the engineering applications the flow can be considered turbulent. The characteristics of laminar flow include a uniform, parallel, and well-ordered movement without any interruption between the layers, minimal momentum convection and constant pressure and velocity in time. On the other hand, the turbulent flow is characterised by irregular, chaotic and stochastic properties, high momentum convection, and fast variations in pressure and velocity in both space and time. The emergence of turbulence is caused by fluid-dynamic instabilities that, for sufficiently high Reynolds numbers, disrupt the nor-

mal laminar profiles and result in the development of unstable vortexes. The generated structures are then affected by additional instability processes, which quickly produce a chaotic flow motion that is, theoretically, difficult to predict. Therefore some modeling effort is necessary. The presence of numerous distinct scales, from the biggest structures of the mean flow down to the smallest vortexes that are dissipated by viscosity, further complicates the scenario.
Turbulence consists of unstable swirls (called eddy) of different size. Inside a turbulent flow there is a energy transfer that is described by the so called Kolmogorov cascade. This cascade starts with the biggest eddies that extract energy from the mean flow and move with the highest quantity of turbulent kinetic energy. Their Reynolds number is large therefore, the viscosity effects on this eddies are negligible. The unstable eddies break up transferring energy to smaller eddies. This energy cascade continues until the Reynolds number is so low that the molecular viscosity effect causes the kinetic energy to dissipate. Different techniques are distinguished depending on whether they simulate more or fewer scales inside this cascade:

- DNS: it simulates directly all the scales of turbulence. In this case the number of operations required is extremely high and consequently also the computational time;
- LES: it simulates the turbulent fluctuations above a certain length scale. Below this threshold the turbulence is modelled with semi empirical laws;
- RANS: it has the highest level of approximation. It solves time averaged equations obtained using statistic operators. It solves the mean field of a generic quantity and models the Reynold stresses deriving from the averaged Navier-Stokes equations;

4.3.1. Reynolds Averaged Navier-Stokes (RANS)

The aim is to solve the mean flow applying an average in time of the mass and momentum equations. Each flow quantity is assumed to be written as the summation of a mean and a fluctuating component:

$$\begin{aligned} \phi &= \overline{\phi(x)} + \phi' \\ \overline{\phi(x)} &= \frac{1}{\Delta T} \int_{t-\frac{\Delta T}{2}}^{t+\frac{\Delta T}{2}} \phi(x,t) dt \end{aligned} \tag{4.4}$$

Each quantity in the N-S equations is subjected to this decomposition, and then a temporal average is calculated. Since many terms simplify once the global average is taken, the mean mass balance equation is formally the same as the instantaneous one. Doing

the same for momentum equation, the linear terms are substituted by the corresponding mean quantities, while the non-linear advective term is represented by a mean component and a non linear combination of fluctuating components:

$$\frac{\partial \rho \overline{\boldsymbol{U}}}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \overline{\boldsymbol{U}} \otimes \overline{\boldsymbol{U}}) = \rho \boldsymbol{g} - \boldsymbol{\nabla}(\overline{p}) + \mu \boldsymbol{\nabla}^2 \overline{\boldsymbol{U}} + \frac{1}{3} \mu \boldsymbol{\nabla}(\boldsymbol{\nabla} \cdot \overline{\boldsymbol{U}}) - \boldsymbol{\nabla} \cdot (\rho \overline{\boldsymbol{v} \otimes \boldsymbol{v}}) \tag{4.5}$$

The fluctuating component, which is on the right side and expressed in divergence form, behaves as a stress applied to the viscous molecular component; it is known as the Reynolds stress tensor, it is symmetric, and it mixes velocity fluctuations. Physically, the driving force that accelerates diffusivity and mixing rate in turbulent flows is represented by the Reynolds stresses. The equations do not give a direct expression for the Reynolds stresses therefore, modelling is necessary to find a solution.
The symmetric Reynolds stress tensor can be written as the sum of an isotropic component and a deviatoric anisotropic component. The isotropic stress is -2/3ρk, it is introduced into the pressure terms (being often negligible with respect to that). In this case k is the turbulent kinetic energy. Boussinesq proposed a purely formal analogy with Newton's stress-strain-rate law to accomplish an effective modelling of the deviatoric component: $\bar{\bar{a}}$=2μ_T $\bar{\bar{D}}$. In this way, the deviatoric Reynolds stress is proportional to the mean strain rate through a turbulent or eddy viscosity μ_T.
By plugging Boussinesq model into RANS equations the system becomes:

$$\begin{cases} \boldsymbol{\nabla} \cdot \overline{\boldsymbol{U}} = 0 & \text{(4.6a)} \\ \dfrac{\partial \overline{\boldsymbol{U}}}{\partial t} + \boldsymbol{\nabla} \cdot (\overline{\boldsymbol{U}} \otimes \overline{\boldsymbol{U}}) = \boldsymbol{g} - \dfrac{1}{\rho} \boldsymbol{\nabla}(\overline{p}) + \dfrac{(\mu + \mu_T)}{\rho} \boldsymbol{\nabla}^2 \overline{\boldsymbol{U}} & \text{(4.6b)} \end{cases}$$

The system of equations is drastically simplified, reducing the non-linear fluctuating terms to diffusive terms. Nevertheless, to assess the eddy viscosity, a model is required.
Several models have been proposed in the past. The first model was introduced by Prandtl and Taylor. It allowed to assess the eddy viscosity through the mixing length concept in the case of simple shear flow:

$$\mu_T \propto l_T u_T \propto l_T^2 \frac{\partial \overline{\boldsymbol{U}}}{\partial y} \tag{4.7}$$

This represented a great improvement because it is more physically meaningful estimating the turbulent length scale (l_T) than the turbulent viscosity.
Successively, Smagorinsky extended the concept of mixing length to general three-dimensional flows, by expressing μ_T as a function of l_T and the strain-rate-tensor ($\bar{\bar{D}}$). Baldwin and Lomax proposed a similar expression, but using the mean rotation rate ($\bar{\bar{R}}$) instead of the mean strain rate. Both the two algebraic models are incomplete because l_T is to be

estimated depending on the flow.
Another class of approaches is the one-equation models. The most important was introduced by Prandtl removing the mixing length concept, and deducing the turbulent velocity scale directly from the turbulent kinetic energy:

$$\mu_T \propto l_T u_T \propto l_T \sqrt{k} \tag{4.8}$$

The turbulent kinetic energy is computed from a balance equation obtained by working out the RANS equations. Also in this case, the system can be closed by prescribing an appropriate value for l_T, for example using the Von Direst formulation.
More generic and complicated formulations, such as two-equations models, have been developed allowing for the consideration of a wider range of scenarios. These models derive the turbulent length and velocity scale from turbulent kinetic energy (k), turbulent kinetic energy dissipation rate (ϵ) or turbulent frequency (ω). The modelled balance equation is solved, as in one-equation models, to determine k. A second set of fully modelled transport equations is solved in order to assess the second turbulent quantity (ϵ or ω).
The k-ϵ model is the most famous and most applied approach in any commercial CFD codes. It is able to guarantee acceptable accuracy for simple flows. The k-ϵ model is singular at the wall, so it requires the implementation of wall functions. It is not recommended in case of reverse flows, extended walls, or strongly curved surfaces since it performs inadequately in the presence of negative pressure gradients.
The k-ω model is able to solve the flow up to the wall but it has worse convergence properties. It can better simulate the boundary layer in case of high pressure gradients and up to separation onset.
In the contest of HAWT, the best way to replicate wind tunnel tests with high precision is to combine the previously discussed eddy viscosity models. This is possible with the k-ωSST turbulence scheme, where SST stands for Shear Stress Transport. It combines advantages of the two main two-equations turbulence models. This model is able to behave like the k-ω formulation close to the walls and like the k-ϵ one far from the walls. Practical experience has shown that the SST is actually the best choice for industrial turbo-machinery flow calculation, and it is presently considered the standard choice for RANS calculation in this field of application. This model does not suffer from wall singularity and gives the opportunity to solve the boundary layer up to the wall if the mesh clustering is sufficient. Therefore, in this work it is necessary to adopt an Unstable Reynolds Averaged Navier-Stokes (URANS) formulation with SST turbulence model due to the unsteady nature of the FOWT physics.

4.4. OpenFOAM Settings

As previously mentioned, the AL code used in this work is implemented in the open source software OpenFOAM. In particular, the code is inserted in its 2.3.0 release version. It is a toolbox for solving partial differential equations on polyhedral grids using the FV method. Any case-scenario in OpenFOAM is fully characterised by the content of three main folders, namely the folders `0`, `system`, and `constant`, which define the appropriate numerical schemes and fluid properties. The next paragraphs describe the computational domain and the setup of these folders used for this thesis work.

4.4.1. Initial and boundary conditions

These conditions are set in the `0` folder. The initial condition represents the starting point for the time-marching unsteady simulation. For a RANS simulation adopting the k-ω turbulent model, the required fields are pressure p, velocity U, volume force *volumeForce*, turbulent kinetic energy k, specific dissipation rate of turbulent kinetic energy ω, turbulent kinematic viscosity ν_t and modified turbulent kinematic viscosity $\tilde{\nu}$. Those quantities need to be defined on the domain sides that are named the inlet, outlet, sky, ground and side patch. The boundary conditions for all the patches are presented in Tab. 4.1.

The quantities are subject to the following five main conditions:

- **fixedValue**: as a Dirichlet explicit condition it assigns a reference value to each cell face located on the corresponding patch;
- **zeroGradient**: it is a Neumann implicit boundary condition which links the field to the internal field value. In this case the gradient is imposed to be zero therefore it associates the value on the patch to the inner cell center one. The diagonal dominance of the corresponding linear system matrix is increased, accelerating and stabilising convergence.
- **calculated**: its value is computed by OpenFOAM via field assignment;
- **InletOutlet**: it is the typical condition in case of return flow. Compared to the zeroGradient solution, it is a less rigid restriction, enabling a better reproduction of the field at the domain outlet for partially stabilised flow;
- **Slip**: it does not apply any limitation on the normal viscous stress, contrarily to the noSlip one. It works by removing the variable normal component from the patch while leaving the tangential components intact.

Patches	U [m/s]	p $[m^2/s^2]$	k $[m^2/s^2]$	omega [1/s]	volForce $[m/s^2]$	nut $[m^2/s]$	nuTilda $[m^2/s]$
inlet							
type	fixVal	zeroGrad	TI 0.02	mixLen 0.1	fixVal	calc	fixVal
value	(4,0,0)	-	0.0096	1.7888	(0,0,0)	0	0
outlet							
type	inOut	fixVal	zeroGrad	zeroGrad	fixVal	calc	zeroGrad
value	(4,0,0)	0	-	-	(0,0,0)	0	-
sky							
type	slip	zeroGrad	zeroGrad	zeroGrad	fixVal	zeroGrad	slip
value	-	-	-	-	(0,0,0)	-	-
ground							
type	slip	zeroGrad	zeroGrad	zeroGrad	fixVal	zeroGrad	slip
value	-	-	-	-	(0,0,0)	-	-
sides							
type	slip	zeroGrad	slip	slip	fixVal	calc	slip
value	-	-	-	-	(0,0,0)	0	-

Table 4.1: Boundary and initial conditions.

4.4.2. Case main controls

Inside the `system/controlDict` dictionary the case principal specifications, including timing information, write format, and optional libraries, are defined.
Here the time step is chosen considering some conditions. The time step has to be small enough to resolve platform frequencies avoiding the leakage effect. Any CFD solver must also adhere to the rule that information is never sent from one time step to the next more than a cell distance away. This constraint is guaranteed limiting the non-dimensional Courant number (Eq. 4.9) below 1. Usually the Co limit is applied to the flow motion but in case of the AL code, the most stringent restriction is caused by the blade-tip velocity, which is far higher than the flow one. Therefore, the time step is selected in order to obtain a rotor tip that does not cross more than one cell per time step.

$$Co = \frac{U\Delta t}{\Delta x} \tag{4.9}$$

Considering these requirements, the time step value is set at $5\text{x}10^{-4}$ s corresponding to an azimuthal angle step of 0.72° and ensuring a $Co \leq 0.5$.
The end time of the simulations is chosen verifying that the convergence is reached not only on the blade-local solution but also on the whole flow field. The wake is carried downstream by the free stream therefore, a certain amount of time is needed in order to achieve the complete propagation and development of the wake. This is the reason why the number of periods simulated is always included between 5 and 20. In Tab. 4.2 the simulation time is specified for each operating condition.

Case	**Frequency [Hz]**	**Period duration [s]**	**endTime** [s]	**Number of periods**
LC1.1	-	-	5	-
LC2.5	1.0	1.0	10	10
LC3.5	1.0	1.0	10	10

Table 4.2: End time for each operating condition.

4.4.3. Numerical schemes

In the `system/fvSchemes` dictionary, instructions on which numerical method to use for the different terms inside the differential equations are contained:

- A second order method, which corresponds to a 0.9 blending of CrankNicolson, is used for the time derivatives;

- Regarding the gradient term, Gauss theorem is applied together with a linear interpolation to compute the volume-integral through face-integral calculation;
- For the divergence the Gauss QUICK scheme is used for the convective terms of velocity. It is a III order scheme, chosen to have high accuracy. For the other quantities a Gauss linear method is applied;
- Schemes for laplacians are Gauss linear corrected;
- The default interpolation scheme used is a linear one. For the velocity field the interpolation is performed with respect to cell-center values;
- Regarding the surface normal interpolation, a corrected calculation has been selected to limit numerical diffusion. The domain is discretized in purely cubic cells therefore no issues of non orthogonality should arise. Nevertheless, by exploiting `refineHexMesh` the margins of the refinement boxes are intrinsically non-orthogonal so the use of a corrected scheme is needed.

4.4.4. Numerical solvers

The `system/fvSolution` dictionary contains the numerical methods used to solve the linearized system derived from the differential equation discretization. For the pressure equation the GAMG (Geometric agglomerated Algebraic MultiGrid) solver combined with the smoother GaussSeidel is adopted. This method should guarantee a higher efficiency and usually is preferred for time saving reasons. For the other equations a PBiCG (Preconditioned Bi-Conjugate Gradient) with DILU (Diagonal-based Incomplete Lower Upper) preconditioner is necessary. The absolute tolerances are set to 1e-5 and 1e-6 for the final cycles while the relative ones are equal to 0.05 and 0 for the final cycles.

In this file also the controls for the solution algorithm are set. The AL solver is developed to advance over time explicitly. When the solver passes to the next time step the flow field immediately available is still the one of the previous instant. At this point the new velocity is sampled and consequently the new volume force is inserted in the momentum equation. Following this strategy the sampled velocity at time t^{n+1} is actually computed over the old flow field (the one of the time step t^n). The proposed new sampling methods, studied in this thesis work, are strictly dependent on the position of the AL point and on the surrounding flow field. Therefore, the sampled velocity at instant t^{n+1} must be consistent with the flow field evaluated at the same instant of time. To overcome this problem, given the explicit nature of the code, a combined PISO-PIMPLE algorithm needs to be applied. In particular the algorithm is set with `nOuterCorrectors`=2 and

nCorrectors=2 which means that for each time step two Simple cycles are performed and the pressure correction is solved twice per each loop. The presence of the two outer loops is necessary to solve the momentum equation a second time. In the second cycle the velocity is sampled on the updated flow field. Therefore, the two external cycles are used to obtain coherence between the sampled velocity and the flow field by solving the problem of the explicit nature of the code, while the two inner loops allow to reach convergence before moving to the next step. This kind of solution strategy becomes even more crucial in the unsteady simulations where a time-step convergence is required to capture the velocity fluctuations in line with platform motion.

4.4.5. Constant setting

Inside the constant folder all the terms that do not change during the entire simulation are defined: the mesh, the fluid features about turbulence and transport properties, airfoil polars and the actuator force parameters necessary for the AL implementation.
The fluid properties are defined in the transportProperties file. The kinematic viscosity is fixed equal to 1.5e-5 m^2/s.
In the turbulenceProperties file the RASmodel kOmegaSST is defined as motivated in section 4.3.1.
All the information about the airfoils polars are contained in the airfoilProperties subfolder.
The actuatorForceParam file describes the AL code settings. Here the geometrical parameters related to the velocity sampling are selected. Depending on the sampling criterion chosen, different parameters have to be indicated in this section (distance and length of the lines, radius of the circumference...). In the same file also the kind of volume force smearing function is chosen and the correspondent regularization Kernel factor is defined. For this work a 2D Gaussian force distribution (Eq. 2.9) is chosen with a constant smearing length scale $\epsilon/\Delta = 2$ along the blade. As already explained in section 2.2.1, this is the best compromise to reduce oscillations and limit the smoothing of the loading.
Finally, the polyMesh sub folder holds the information about the background mesh. More details are described in section 4.4.6.

4.4.6. CFD mesh

The discretized domain size is set following the physical dimensions of the wind tunnel used during the experiments. Nevertheless, some modifications must be applied:

- Due to the existence of the rotor and a contribution from the limited wake expansion, the wind flow experiences the blockage effect during the experimental tests. A boundary layer thickness must be deducted from the upper and lower sides in order to maintain the mass flow rate while without adding extra energy to the numerical calculation. This reduction corresponds to δ^*_{bottom}=0.125 m and $\delta^*_{ceiling}$=0.095 m. On the lateral walls the blockage effect is negligible therefore no modification is needed. Thus a slip condition is applied to the lateral, lower and upper walls;
- The domain must be extended along stream-wise dimension (x-axes) to guarantee the proper inlet velocity avoiding the presence of the rotor induction at the inlet and the complete pressure recovery in the downstream wake;

Hence the inlet is located 3D upstream from the rotor, the outlet is 14.5D downstream and the lateral sides are 5.8D far from the hub. The height of the domain is 3.62 m. These dimensions are chosen as a good compromise between the need of having the external patches far enough to not artificially influence the flow field and the need of limiting the computational cost. The meshing procedure starts with the realization of the background mesh in `blockMeshDict` where different blocks are defined (Fig. 4.1):

- Block E: this is the main block where the turbine is located. All the refinement regions lay in this domain portion to better describe the flow-blade interaction, rotor induction and wake development;
- Blocks A,B,C: they are necessary to move the inlet away not to affect the upstream flow field;
- Blocks D and F: they are placed to replicate the same width as the real wind tunnel and reduce the blockage effect.

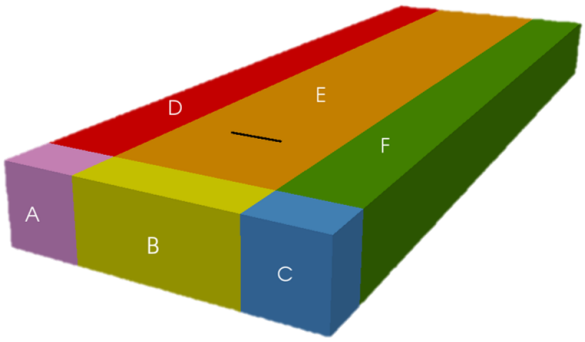
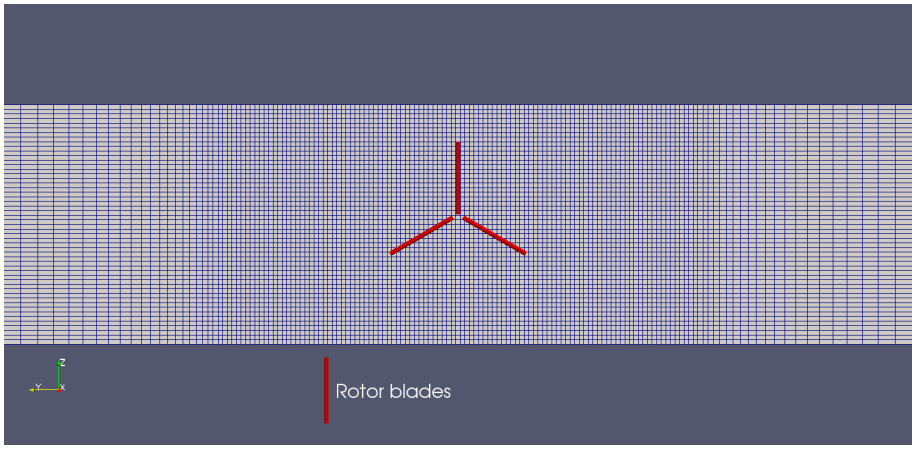
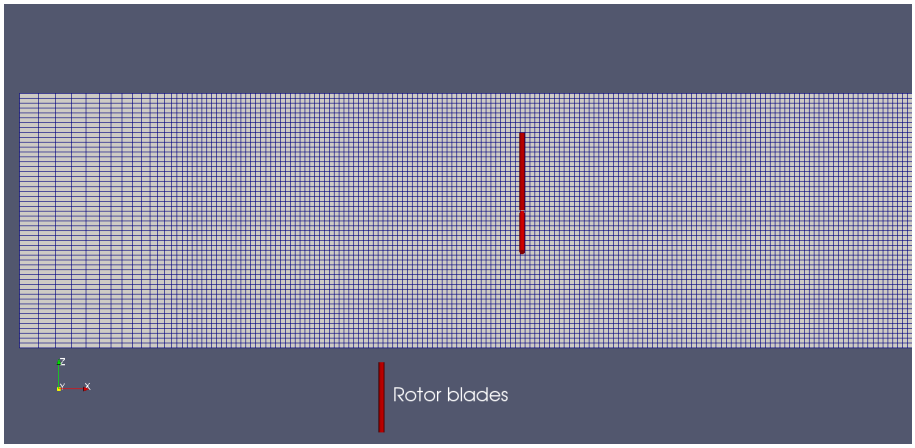


Figure 4.1: Mesh blocks layout.

As reported in Fig. 4.2, the background mesh consists only of hexaidral cells thanks to the fact that the AL model does not require the reconstruction of the real blades shape. Therefore, a uniform and orthogonal mesh can be adopted. In order to low the cells size in the central region, where the turbine is located, a simple grading is applied making sure that the transition to the lateral walls is smooth.



(a) Background mesh slice on y-z rotor plane (front view)



(b) Background mesh slice on x-z plane (side view)

Figure 4.2: Different views of the background mesh.

Starting from the base mesh two refinement regions are introduced to better capture the flow field around the turbine, the vortex structures at the blade tip and the wake evolution. The refinement is needed also to set a proper number of actuator points along the blade. Since the implemented code defines the actuator point starting from the intersections found between the actuator line and the mesh, the cell size is a paramount parameter to fix the number of AL points. As already mentioned, good practice requires to have at least 30-60 points per blade [16]. Thus, two refinement cylindrical regions are introduced using recursive application of the `topoSet` and `refineHexMesh`. Their specifications

are reported in Tab. 4.3 and their representation is visible in Fig. 4.3.

Refinement	Type	p1 (x,y,z) [m]	p1 (x,y,z) [m]	Radius [m]
toposet.local.1	cylinder	(-2.4, 0, $TowerHt)	(34.5, 0, $TowerHt)	1.614 (1.36R)
toposet.local.2	cylinder	(-0.3, 0 , $TowerHt)	(0.3, 0, $TowerHt)	1.4 (1.18R)

Table 4.3: Refinement regions settings.

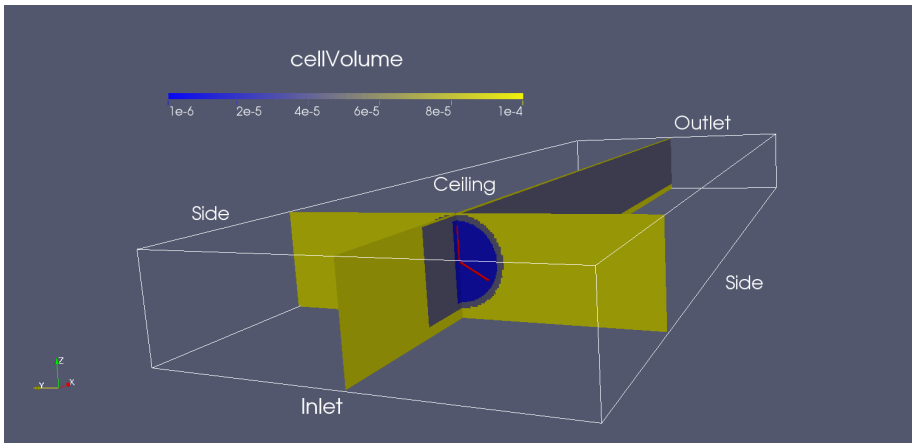
To accommodate the various experimental setups, the parameter `$TowerHt` is adjusted to 2.0634 m for unsteady cases and 1.961 m for fixed-bottom cases. The stream-wise extension of the refinement regions must be sufficiently big to overall contain the actuator lines in each possible position during the surge and pitch motion. It is important to avoid sudden span-wise changes of grid resolution to keep the solution stability. In this way, the final mesh presents a cell size in the E block Δ=0.017 m and an overall number of cells of 11.4 million. The number of actuator line points is included between 60-90 per blade line according to the azimuth position.

After the enforcement of the refinement regions, the `renumberMesh` application is performed to have a better conditioning of the problem. Eventually the mesh quality is analyzed running the utility checkMesh. The main parameters are reported in Tab. 4.4.

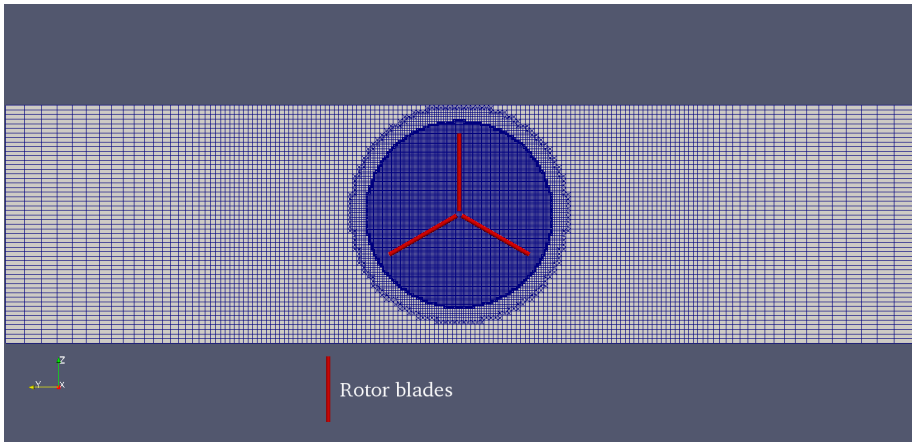
Parameter	Value
Hexahedra	11236336
Polyhedra	118742
Max aspect ratio	4.0109
Max non-othogonality	25.3292
Max skewness	0.3333

Table 4.4: Check mesh parameters.

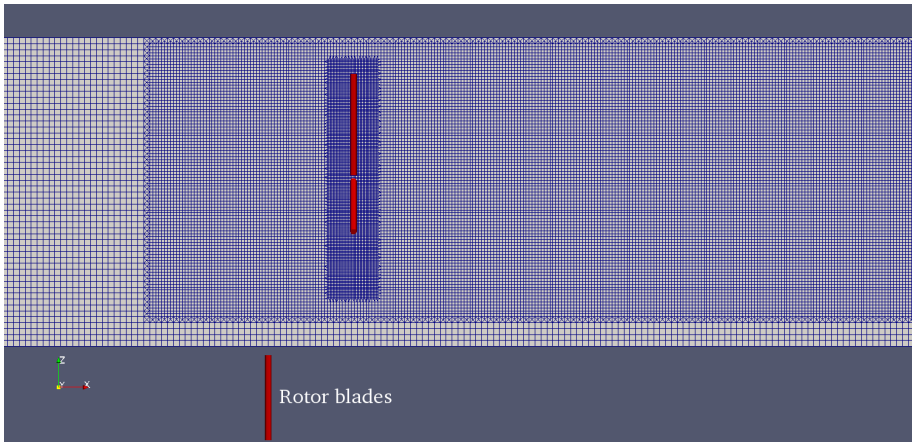
The two most important parameters are: the non- orthogonality which is, in this case, well below 70, therefore, no stability issues should be present; the maximum skewness is sufficiently small not to produce difficulties in reaching the convergence.
It can be inferred from that the mesh architecture has a good quality suitable for solving the given simulations.



(a) Numerical domain and mesh refinement levels



(b) Mesh slice on y-z rotor plane (front view)



(c) Mesh slice on x-z plane (side view)

Figure 4.3: Different views of the numerical domain and refinement regions.

A proper sensitivity analysis was not performed for some reasons. The final refinement mesh level, here proposed, is the result of some attempts done on different sampling strate-

gies. As explained in chapter 6, a higher refinement level generates an excessive number of AL points producing an over-fitting problem that results in nonphysical oscillations along the blade. At the same time, a coarser mesh would not be sufficient to properly reproduce the volume force distribution around the AL point. Furthermore, a proper grid independence study is not possible. Since in this work the smearing factor ϵ is a function of the cell size Δ, a change in the mesh refinement generates a change in force distribution and therefore making impossible an achievement of loads convergence. In [8] it is demonstrated that a grid independence could be reached only if $\frac{\epsilon}{\Delta} > 5$, being the force strongly distributed. This is not the case since in this work, a ratio $\frac{\epsilon}{\Delta} = 2$ is chosen following the reasons explained in chapter 5. Since a grid independence is not exactly possible, some studies have been carried out in literature deriving typical values of some parameters to obtain a correct estimate of the turbine performance. Therefore, in this thesis, the choices on the mesh refinement and on the forces distribution are made respecting the values suggested by literature. In particular, the smearing coefficient $\frac{\epsilon}{D} = 0.014$, equal to half of the ratio employed in [8] and corresponding to $\frac{\epsilon}{c} < 0.5$, is selected to ensure accuracy in power and blade loading prediction, as suggested in [7]. Therefore, the final mesh is selected as the result between a proper number of AL points, a proper accuracy in power prediction and an acceptable computational burden.

5 | Model set-up

In this thesis work only the three blades are simulated. To keep the model as simple as possible the nacelle and the tower are not modelled. In this situation the flow in the near-wake region (within 5D downstream) would be inadequately reproduced. However if the goal of the analysis is the study of the wake main features, its recovery, its helix structure or development in the far wake zone, modelling just the rotor may be sufficient. Additionally, during the OC6 experimental campaign a sensitivity analysis on the tower influence was performed demonstrating that the difference between the mean thrust computed with and without the presence of the tower is 0.05%. This result allows to consider equivalent the two solutions [27]. In order to improve the model, the nacelle could be simulated with an Actuator Disk and the tower with a fourth actuator line. In this way a complete characterization of the new two components must be realised in terms of C_l and C_d. These parameters are easily determined for the tower given its cylindrical shape. On the contrary, an extensive numerical campaign is required to properly model the nacelle geometry. To characterize the nacelle in different flow conditions, high fidelity simulations are necessary which turn out to be a very demanding procedure in terms of computational effort. In this work the only presence of the three blades has some benefits in terms of computational cost for two main reasons: the absence of additional obstacles, that further deviate the flow, leads to faster convergence; since the AL code necessitates knowing every object position and velocity at every time steps, fewer objects bring to less computational effort.

As already mentioned, the code used in this work is based on the so called Actuator Line Algorithm whose main steps are reported in Section 2.2. Among them, the two primary phases that need to be defined are:

- selection of a proper regularization kernel function for the force spreading;
- definition of a specific velocity sampling method to correctly estimate the attack angle and consequently the blade loads.

Regularization Kernel function

In this work, a 2D cylindrical force distribution is selected using a regularization Kernel

factor as reported in Eq. 5.1:

$$\eta(r) = \frac{1}{\pi\epsilon^2} exp(-\frac{d^2}{\epsilon^2}) \tag{5.1}$$

The smearing factor ϵ is defined as a function of the average cell dimension $\Delta = \sqrt[3]{V_{cell}}$. In particular, the value chosen is $\frac{\epsilon}{\Delta} = 2$ according to the sensitivity analysis proposed in [36]. The distribution of the force over the cells is a strategy needed to overcome a numerical problem, therefore a dependence of ϵ on the numerical resolution of the mesh is considered more accurate than a dependence on a physical parameter such as the chord. A symmetric Gaussian force distribution will insert the 68% of the force on a circular area of radius ϵ, the 95% on an area of radius 2ϵ and the 99.7% on an area of radius 4ϵ. In this case, for the mesh selected and described in Sec. 4.4.6 and for the chosen value $\frac{\epsilon}{\Delta} = 2$, the distribution over the cells around the AL point is the one reported in Fig. 5.1.

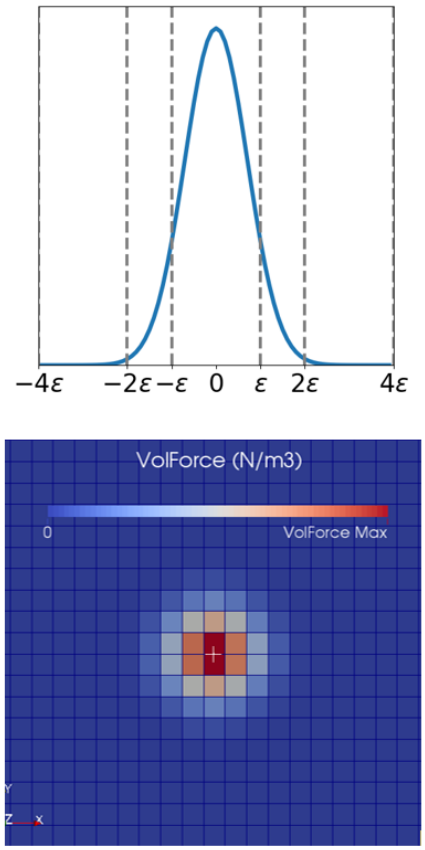


Figure 5.1: Gaussian 2D force distribution.

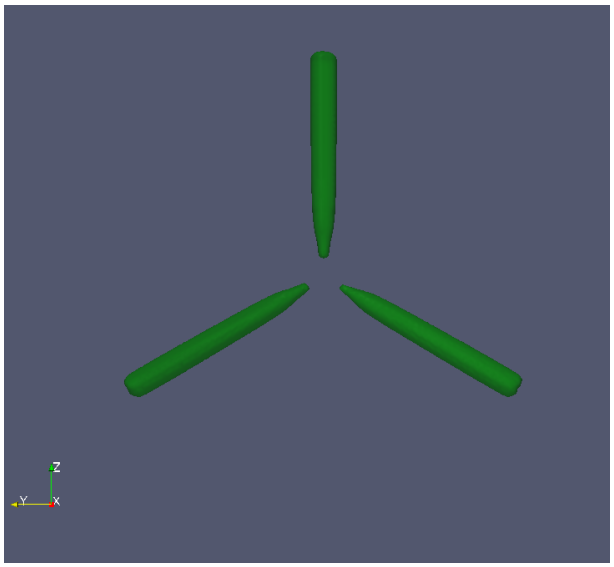


Figure 5.2: Isosurfaces of the volume forces along the blades.

The overall volume force distribution along the blade is shown in Fig. 5.2 where the iso-surfaces of the projection function are reported. Having selected a 2D smearing function it is possible to notice that at the tip the volume force distribution well reproduces the finite length of the blade. For this reason this model is able to properly simulate the detachment of the tip vortex and its helix evolution in the wake as shown in Fig. 5.3 where Q criterion iso-surfaces are shown in cyan (Q=10).

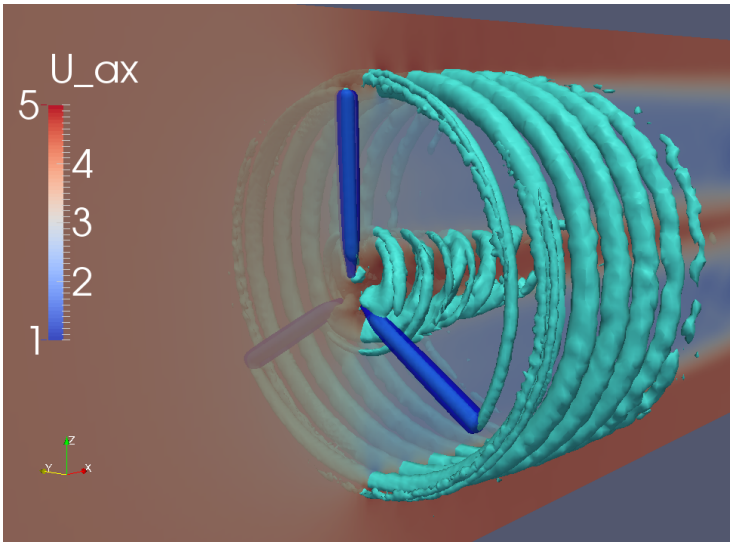


Figure 5.3: Tip vortex detachment: Q criterion iso-surfaces are shown in cyan. Iso-surfaces of the volume forces are depicted in blue.

However, the current model does not take the tip losses of the rotor into account. The

induced drag caused by the development of the tip vortex is inherently captured by solving the fluid domain, but the equalisation of the forces tending to zero is not imposed. The possible implementation of a corrective coefficient for tip losses can be realized in future analyses.

Velocity sampling strategy

The main goal of this thesis work is the validation of new velocity sampling methods capable of properly estimating the AoA and so the blade load in all the possible working conditions of a FOWT. From a purely theoretical point of view, the AoA can be computed in correspondence to the actuator line itself, as it is located where the influence of the bound vortex is cancelled out. Nevertheless, sampling the velocity in this point could result in a significant interpolation error, due to the high velocity gradients concentrated where the force is applied. Therefore, the idea is to sample the absolute velocity around the regularisation kernel region and then to compute the final value as an average. In this work, different sampling strategies in the fixed-bottom case were studied before finding the final ones.

The first implementation aims to keep one or more sampling lines in order to not deviate too much from the original EVM model. This is useful to demonstrate that even a simple and easy to implement sampling method can still provide valid results. Starting from this consideration, different strategies characterized by different positions of the lines were analysed. These attempts allow to collect information in order to define the best configuration capable of removing all the problems encountered. The sampling method is defined accurate when, for each AL point, the sampled velocity is equal to the velocity value of the flow field at the same point. In this way the sampled velocity can be considered free of the bound vortex effect, however avoiding the problems of interpolation and numerical stability that would be encountered by sampling directly at the actuator point. Some of the numerous attempts done in the steady case are described in detail in Sec. 6 to understand the difficulties encountered and the solutions applied before reaching the final version of the strategy.

If the first method was born to maintain a certain ease of implementation, the last two methods are instead based on more physical reasons. The second type of sampling is based on the so called Line Average method described in Sec. 2.2.2, while the last one follows the basic physical concept proposed in the Shen 1 scheme. All these approaches are fully described in the following sections.

6 | Tests on various sampling configurations

The first sampling strategy implemented consists of two sampling lines perpendicular to the freestream direction x, one upstream and one downstream. Lines are positioned, for each AL point, on the plane perpendicular to the actuator line. Their distance and length is parameterized according to the chord of the profile corresponding to that specific point. An example for ten stations with a distance of the sampling lines of 3c is reported in Fig.6.1

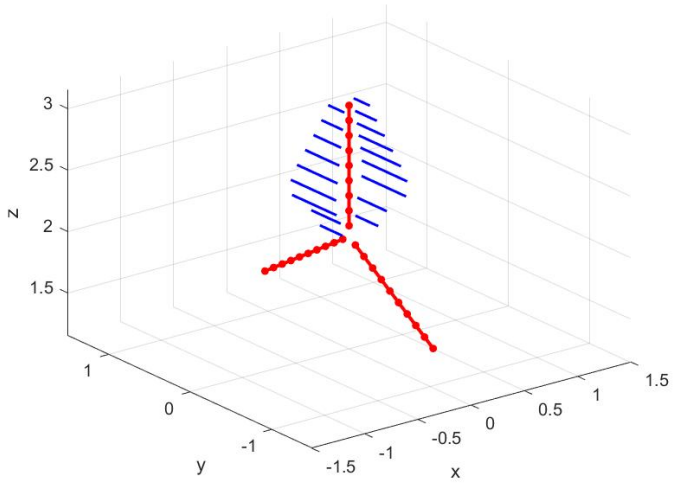
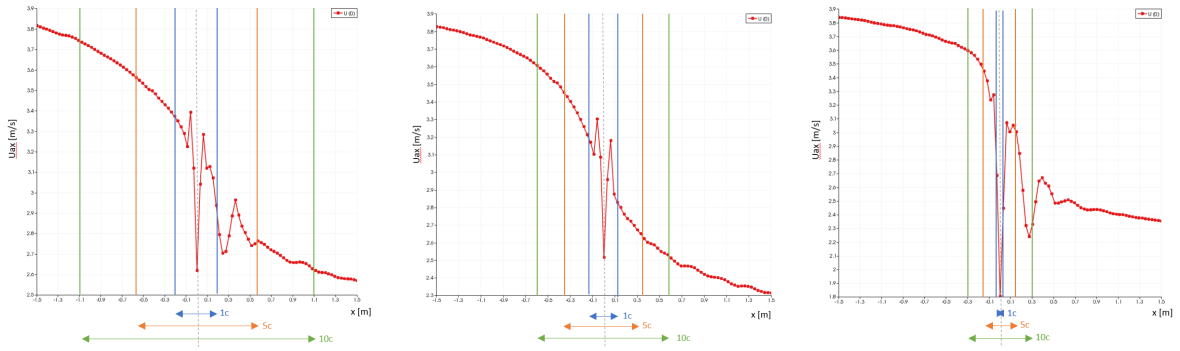


Figure 6.1: Sampling lines for the $LC11_A$ method.

The idea behind this method was born by observing the trend of the axial velocity along a line passing through the AL point and parallel to the flow direction x. It can be seen that, instead of a smooth decreasing trend, a strong oscillation is generated around the point due to the distribution of the force. Therefore, this irregularity has a purely numerical origin. To avoid this problem, the method samples the speed on two fairly distant lines and then averages it to reproduce the physical linear trend that the axial velocity should



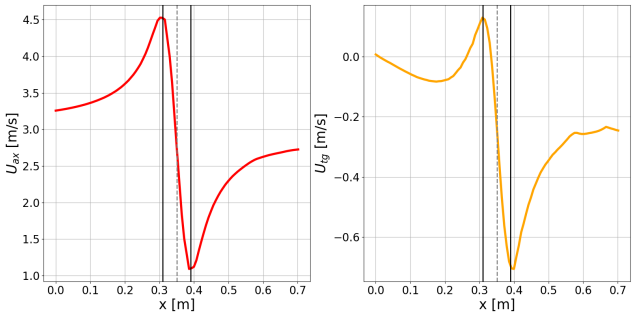
(a) $r/R = 36\%$ (b) $r/R = 71\%$ (c) $r/R = 96\%$

Figure 6.2: U_{ax} trend along the x-axes passing through the AL point. The vertical lines reproduce the sampling lines for three different distances 1c, 5c, 10c.

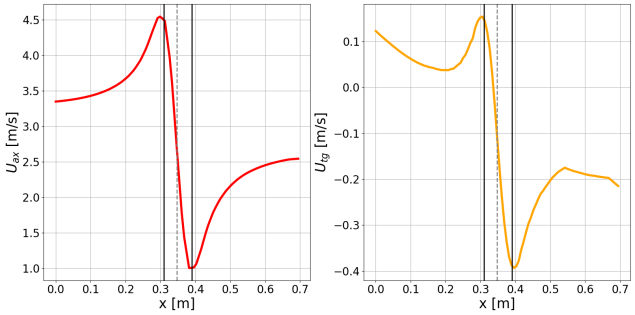
have. For this formulation, different distances of the lines were tested while the lines length was kept equal to 5c as in the original EVM model. In Fig. 6.2 the aforementioned U_{ax} trend is shown for three different radial positions: 36%, 71%, 96% of the rotor radius.

The figure shows that the most critical section is the one near the tip. In this region the oscillations extend more along the x-axis. Therefore, even a distance of 10c would not be enough to get out of the problem zone. This sampling can not be considered reliable in the tip region which, however, generates the greatest contribution in the exchange of forces. At the same time, a distance greater than 10c would lead the lines to lie on different stream lines and to lose the field information of the single AL point.

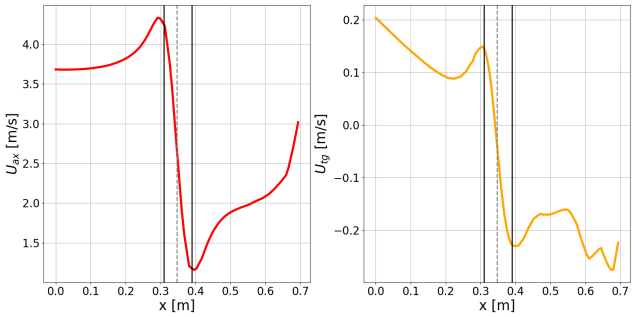
For this reason, this method was discarded in favor of others which sample no longer along the x-axis but along the direction of the relative velocity. Indeed, along this direction, the trends of U_{ax} and U_{tg} present, for each section, a linear stretch around the ALP with an amplitude equal to 2ϵ as shown in Fig. 6.3.



(a) $r/R = 36\%$



(b) $r/R = 71\%$



(c) $r/R = 96\%$

Figure 6.3: U_{ax} and U_{tg} trend along a line parallel to the relative velocity and passing through the AL point. The vertical black lines are located at a distance 2ϵ from the AL point (dashed line).

Therefore, it seems more convenient to sample along this direction as the trends are linear and uniform for each radial coordinate. Furthermore, since the amplitude of the linear portion is a function of ϵ and so of the cell size Δ, parameterize the sampling on this variable results more accurate. Hence, the geometry parameters of the methods proposed from here on will be expressed as a function of Δ. The new target was to sample very

close to the AL point in order to stay in the linear stretch. Starting from this concept two new strategies were tested:

- $LC11\_B$: the velocity is sampled on a single line parallel to the relative velocity and passing through the AL point. In Fig. 6.4 an example of the sampling along one blade is shown.

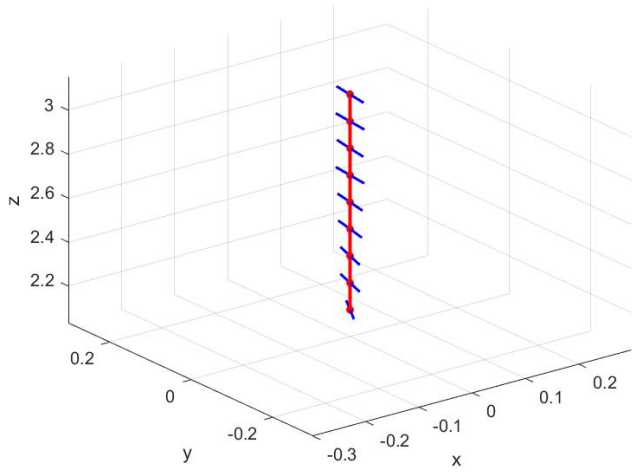


Figure 6.4: Sampling lines for the $LC11\_B$ method.

- $LC11\_C$: the velocity is sampled on two lines perpendicular to the relative velocity and then the average is computed. In Fig. 6.5 an example of the sampling along one blade is reported.

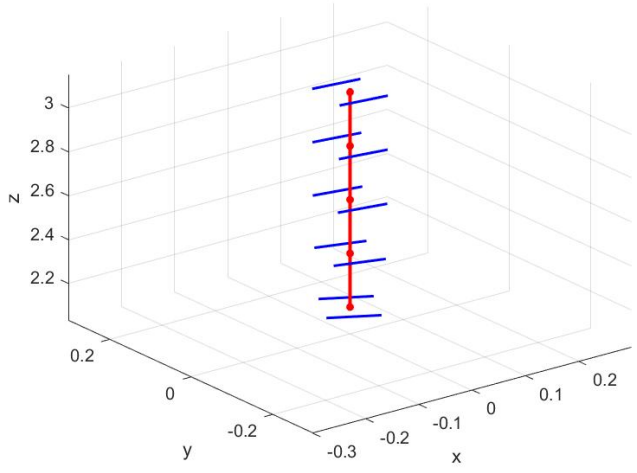


Figure 6.5: Sampling lines for the $LC11\_C$ method.

- $LC11\_D$: the velocity is sampled on two lines. Following the gradients of the two components U_{ax} and U_{tg}, the axial component is sampled on a line parallel to the relative velocity and passing through the AL point while the tangential component is sampled on a line perpendicular to the relative velocity and always passing through the AL point. In Fig. 6.6 an example of the sampling along one blade is reported.

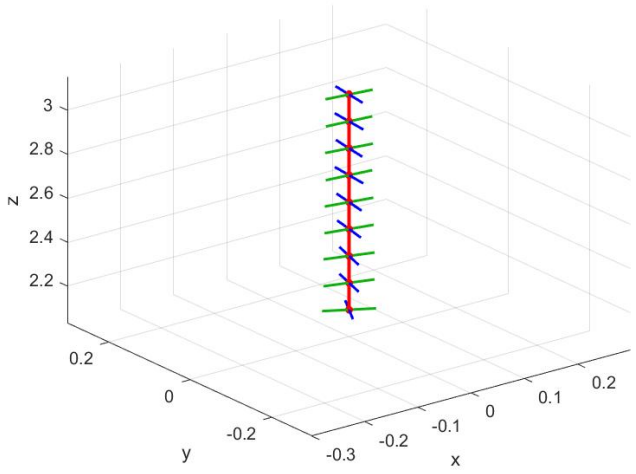


Figure 6.6: Sampling lines for the $LC11\_D$ method.

In order to verify the validity of these three methods, the integral quantities over time and the blade aerodynamics are investigated. In Fig. 6.7 the thrust and torque trends versus time are shown. The dashed line reports the experimental value.

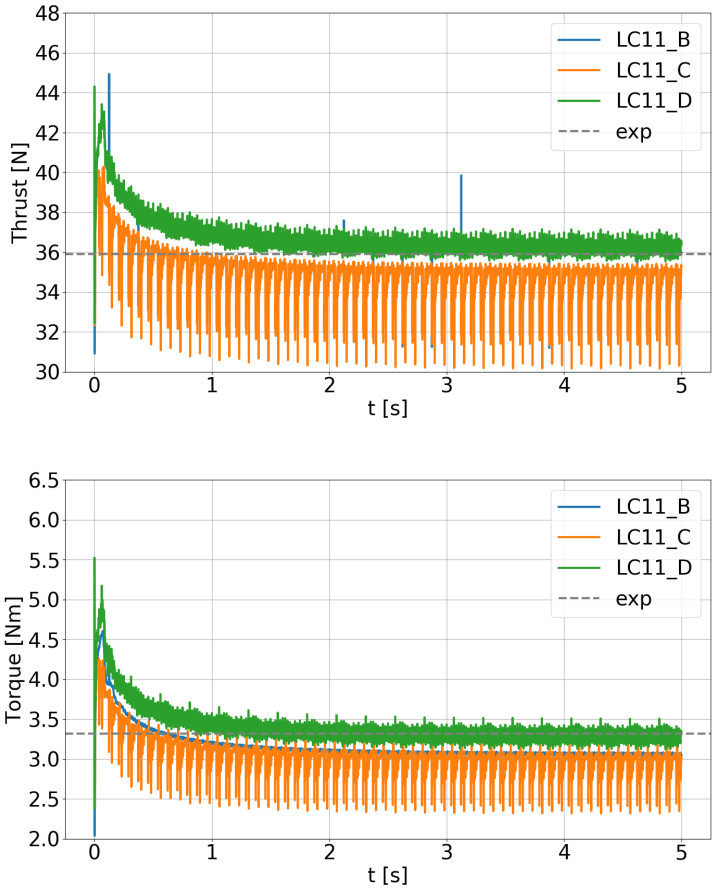


Figure 6.7: Thrust and torque convergence against simulated time for $LC1.1\_B$, $LC11\_C$ and $LC11\_D$

Fig. 6.8 shows the blade axial and tangential velocity components and the angle of attack for 60 stations along the blade. These quantities are averaged over the last revolution period for one blade but fully consistent plots were found for the other two blades.

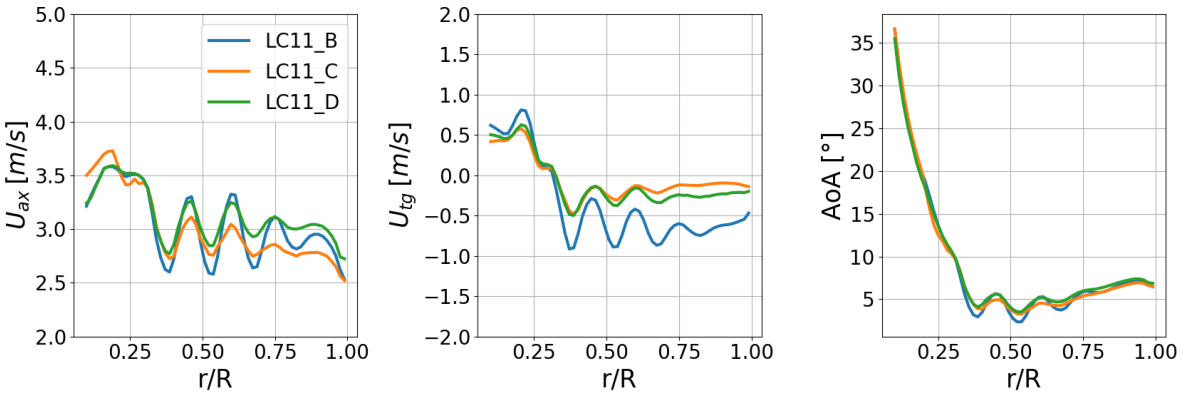


Figure 6.8: Sampled velocities and angles of attack along the blade span.

The physical analysis and considerations about these curves are performed in chapter 8 where the definitive sampling methods are described in detail. In this section, the goal is to identify the issues that led to the final configurations. For the $LC11\_B$, $LC11\_C$ and $LC11\_D$ simulations the main problems are:

- strong variations of the integral quantities during each revolution, especially for the $LC11\_C$ simulation.
- oscillations of the sampled speeds along the blade and consequently of the computed attack angle.

Both problems have no physical explanation. Regarding the integral quantities, an almost uniform trend of thrust and torque is expected, once convergence is reached. Since the flow approaching the rotor is uniform and the tower is not modeled, there is no reason to get big oscillations during a revolution. Similarly, a physical reason for the oscillations along the blade can not be provided. These irregular trends may be due to the fact that, with both methods, the sampling is carried out in an area with a very high speed gradient. The fact that a point of the sampling line can fall in a cell or in the immediately adjacent cell leads to sample a completely different speed thus generating those strong variations along the blade and then over time.

Following these considerations a new method was implemented. The $LC11\_E$ method samples U_{ax} on two lines perpendicular to the relative velocity (blue lines in Fig. 6.9) , one upstream and one downstream of the AL point. U_{tg} is sampled on two lines parallel to the relative velocity (green lines in Fig. 6.9), always one upstream and one downstream. The final components are then calculated as the average between the two lines.

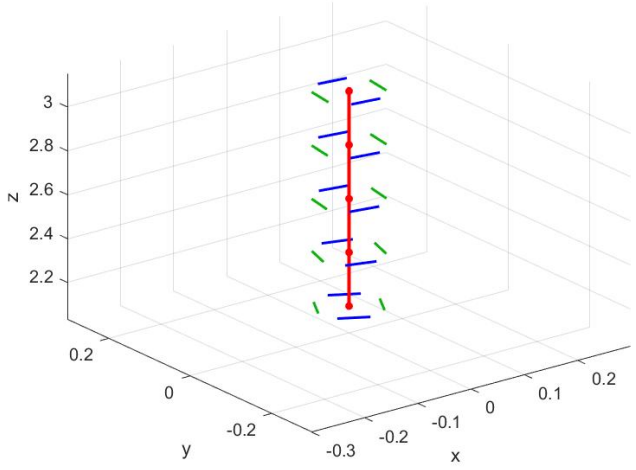


Figure 6.9: Sampling lines for the $LC11\_E$ method.

Before applying this new strategy, some considerations and some modifications were applied in the simulation set-up. First, all previous methods were tested using a force distribution factor ϵ=1Δ, following the value suggested by the original EVM model. During the flow field analysis it was decided to refine the force distribution thinking of reducing the previously mentioned oscillations. Therefore, the new $LC11\_E$ method was firstly applied to a new mesh characterized by a further level of refinement in the area around the rotor. The characteristic cell size of the new mesh is therefore half of the old one. To adequately reproduce the distribution of the force, ϵ=2Δ was set in order to have a total area of the force distribution equal to the previous one in absolute terms, but a better diffusion thanks to the smaller size cells (Fig. 6.10).

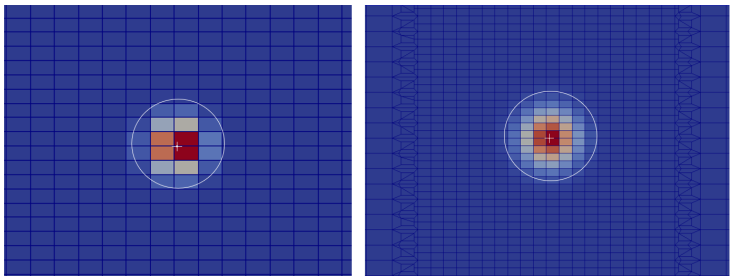


Figure 6.10: Force distribution of the coarse (left) and refined (right) mesh.

The original distribution involves a very low number of cells. A smoother trend over a greater number of cells is useful to better reproduce the bound vortex around the AL point and thus the sampling phase.
During this phase another important code change was applied. From the previous simulations, the sampled speed resulted always higher than the CFD velocity value of the flow field at the AL point. By analyzing the structure of the code in detail, it was observed that the code advances explicitly as explained in Sec. 4.4.4. Hence the decision to implement a PIMPLE solution strategy instead of the PISO originally used. From now on all simulations have been conducted with this new change. This method was tested, on the refined mesh, varying the length and the distance from the AL point of the sampling lines. In particular, three different cases were simulated and their characteristics are reported in Tab. 6.1.

	Distance from the ALP	Length
$LC11\_E\_mRef\_a$	4Δ	4Δ
$LC11\_E\_mRef\_b$	10Δ	4Δ
$LC11\_E\_mRef\_c$	10Δ	8Δ

Table 6.1: Geometrical parameters of the three tested cases for LC11_E method.

The results of these three attempts are shown in Fig. 6.11 and Fig. 6.12 as integral quantities and blade trends of U_{ax}, U_{tg} and AoA.

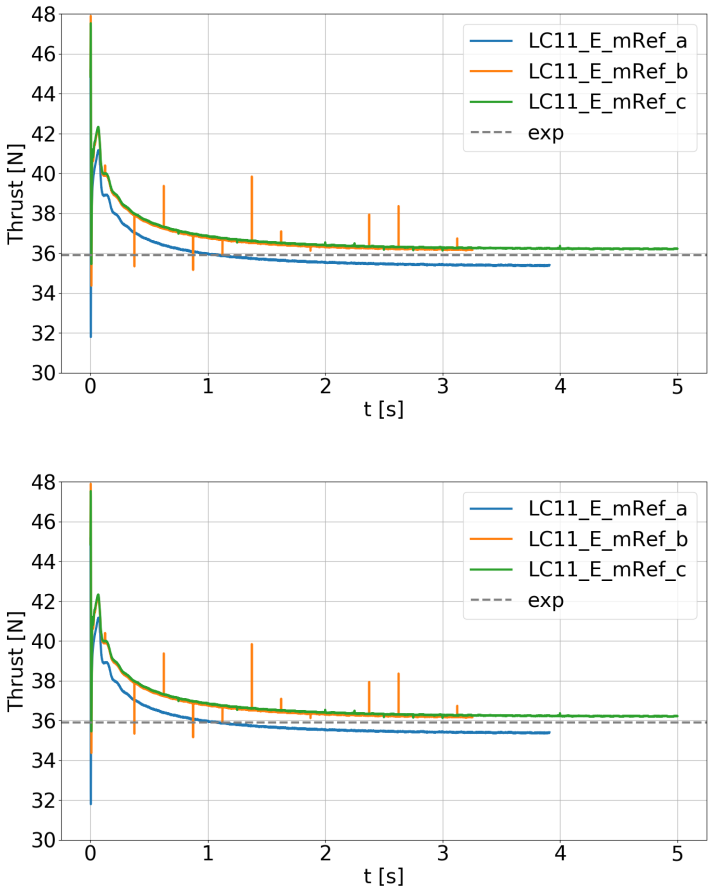


Figure 6.11: Thrust and torque convergence against simulated time.

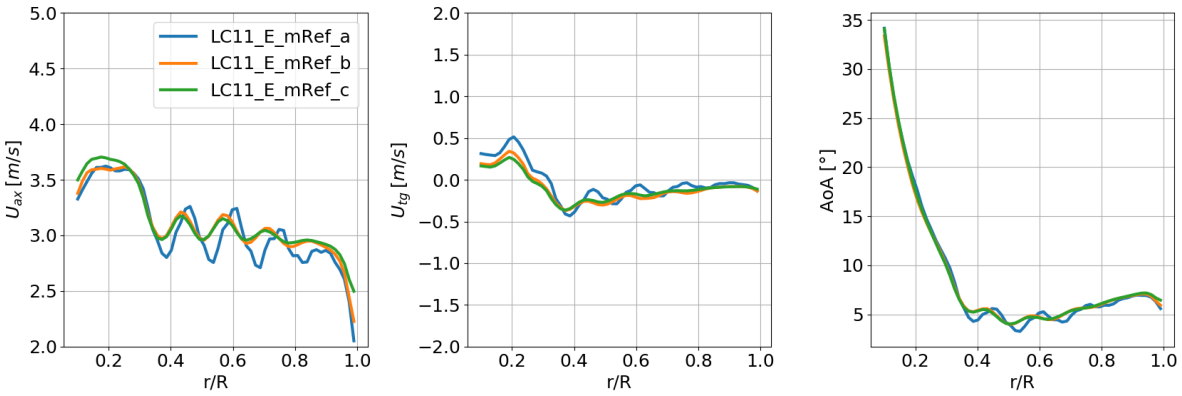


Figure 6.12: Sampled velocities and angles of attack at the blade stations along the blade span.

Analyzing these cases, the first important conclusion drawn is that the better force distribution can improve the sampling quality but is not responsible for the blade oscillations that are still present. Using the case $LC11\_E\_mRef\_a$ as a reference, it can be observed that the increase of the lines distance from the AL point is useful to reduce the oscillations being outside the region of maximum gradient. Nevertheless, the case $LC11\_E\_mRef\_b$ presents some peaks in the thrust and torque parameters that are not periodic with the revolution. They seem to show the presence of some numerical instabilities. On this side, a further improvement is obtained lengthening the sampling lines (case $LC11\_E\_mRef\_c$). This last strategy does not show peaks in the integral quantities and has limited oscillations in the blade trends.
At this point the problem is to understand what is actually the origin of the blade oscillations. Also in this case a further analysis on the code functioning was carried out. The code defines the AL points based on the intersections of the actuator line with the mesh. Therefore, the number of AL points depends on the mesh refinement and on the blade position. By increasing the level of refinement around the rotor, the number of AL points went up from 64-91 to 130-183. From literature, the suggested number of AL points is 30-60 ([10] and [16]). Exploiting this information, it was thought that the blade oscillations could be due to an over-fitting. The excessive refinement and therefore the too high number of AL points generates instabilities in the velocity sampling.
Therefore it is concluded that the oscillations are mainly due to two factors: sampling too close to the AL point and excessive number of AL points. From these considerations the parameters of the definitive sampling method $LC11\_E\_d$ based on lines were defined. This method is completely equivalent to the previous one as it samples on four lines but it is tested on the coarser mesh (described in Sec. 4.4.6) to avoid the over-fitting problem.

This also leads to distance the sampling lines since the distance and the length of the lines are expressed as a function of the characteristic cell size Δ. The force distribution parameter is however kept equal to 2 to have a smoother distribution. The lines distance and length are kept equal to the last case tested on the refined mesh (Tab. 6.2).

	Distance from the ALP	**Length**
$LC11\_E\_d$	10Δ	8Δ

Table 6.2: Geometrical parameters of the definitive LC11_E_d method.

The comparison between the $LC11\_E\_mRef\_c$ and $LC11\_E\_d$ cases is shown in Fig. 6.13 and 6.14.

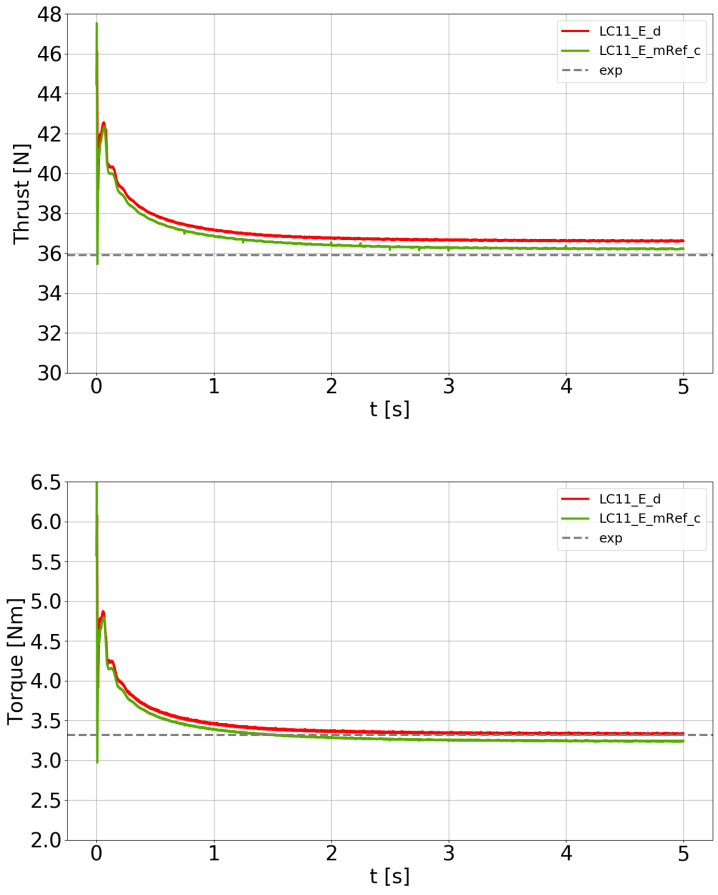


Figure 6.13: Thrust and torque convergence against simulated time.

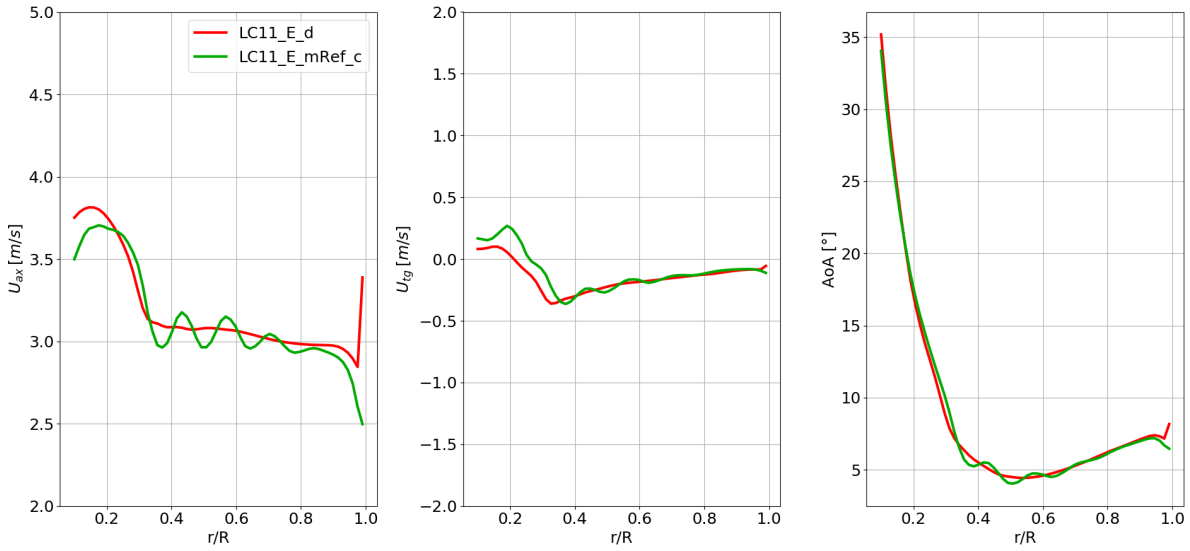


Figure 6.14: Sampled velocities and angles of attack at the blade stations along the blade span.

Fig. 6.14 shows that the simple reduction of the mesh refinement and the consequent increase of the sampling lines distance and length is able to completely remove the blade oscillations obtaining the average value of the previous oscillatory trend. The main differences are concentrated at the tip but the following analysis (chapter 8) will justify the growing trend of U_{ax} and confirm its correctness.
All the information and notions that have been obtained in this phase represent the basis for the definition of the final methods that are described in the following chapters.

7 | Definitive velocity sampling methods

In this chapter the final three sampling strategies are described in details. First of all a presentation of their physical reason and their geometric parameters is provided. Afterwards, their validation against experiments is presented in the fixed-bottom, surge and pitch cases. All the following simulations are carried out on the mesh described in Sec. 4.4.6 and using a smearing function parameter $\epsilon = 2\Delta$. Every AL (actuator-line) point is subject to the algorithm, which only considers components laying in the plane locally orthogonal to the blade axis. Starting from the average quantities of the velocity over the assigned sampling points the angle of attack is computed according to Eq. 7.1:

$$AoA = \arctan(\frac{\overline{U_{ax}}}{\Omega r + \overline{U_{tg}}}) - \gamma \tag{7.1}$$

where γ is the local pitch angle.

7.1. Lines sampling method

This method has been implemented with the aim of demonstrating that a very simple and easy to implement sampling criterion can still prove valid. Starting from the idea of the original EVM model, the line sampling is kept but the empirical correction on the angle of attack is eliminated. It is therefore necessary to sample a velocity already devoid of the bound vortex local effect, in such a way that the angle of attack computed is already the correct one. This method is based on the sampling of speeds on four lines:

- U_{ax} is sampled on an upstream and downstream line, both perpendicular to the relative velocity, and then averaged between the two.
- U_{tg} is sampled on an upstream and downstream line, both parallel to the relative velocity, and then averaged between the two.

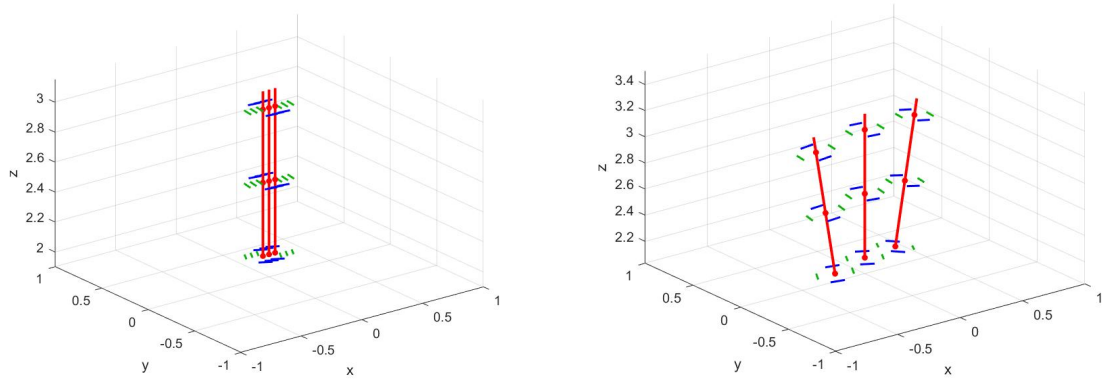
Thus, the method is fully equivalent to the one called $LC11\_E\_d$ in Sec. 6. The reasons

that led to this specific configuration have been analyzed in detail in the same section. The geometric parameters of the lines are shown in Tab. 7.1.

	Distance from the ALP	**Length**
_lines	10Δ	8Δ

Table 7.1: Geometrical parameters of the definitive lines method.

The lines are constructed on the plane perpendicular to the blade axis whatever the blade movement. In Fig. 7.1 the sampling lines are represented for three generic sections in case of surge and pitch motion only for the vertical blade. The correct position of the lines is obtained by first expressing their coordinates in the local reference system and then manipulating them with rigid body motion transformation matrix.



(a) Surge (b) Pitch

Figure 7.1: Sampling technique applied to different platform motions.

The green lines are parallel to the relative velocity and sample U_{tg} while the blue ones are perpendicular to the relative velocity and sample U_{ax}.
The code implementation of this method is completely described and reported in appendix A.

7.2. Circle sampling method

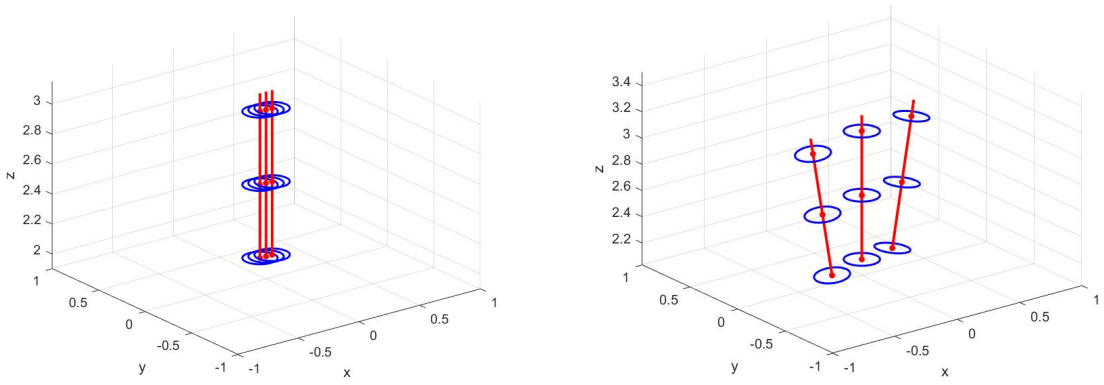
These last two methods, unlike the previous one, arise from a more physical motivation related to how the bound vortex acts on the flow field and therefore how its effect can be removed by velocity sampling. The method that samples on a circumference is based

on the same concept as the so-called "Line average" method described in Sec. 2.2.2. The induced velocities at opposite points of the circle simplify each other by eliminating the influence of bound circulation. Therefore, the velocity components are sampled at N points along the circle centered in the ALP and then averaged. The geometric parameters are listed in Tab. 7.2.

	Number of points along the circle	**Circle radius**
_circle	36	10Δ

Table 7.2: Geometrical parameters of the definitive circle method.

The circle radius is chosen equal to 10Δ to compare the results with the previous method. The number of points along the circle is set in order to obtain one point each 10°. This is a good compromise between high sampling accuracy and computational time. Fig. 7.2 shows the sampling circles for three generic sections in case of surge and pitch motion only for the vertical blade.



(a) Surge (b) Pitch

Figure 7.2: Sampling technique applied to different platform motions.

The code implementation of this method is completely described and reported in appendix A.

7.3. Vortex sampling method

This sampling method is based on some of the physical concepts that underlie the method called "Shen 1" in Sec. 2.2.2. The bound circulation around the airfoil section is represented by a concentrated point vortex centered in the ALP. In order to stay pretty close to

the original EVM model, the velocity is still sampled on a line upstream and perpendicular to the relative velocity. The sampled velocity will be influenced by the bound vortex effect so to remove it a correction must be applied. Instead of applying an empirical correction on the angle of attack, the sampled velocity is corrected by subtracting the induced one. To compute the circulation and therefore the induced speed, a circle centered on the ALP is once again defined. Therefore, the main steps of this sampling are:

1. On the upstream line N_{samp} velocities are sampled (according to the number of intersections found with the mesh);

2. The circle around the AL point is defined choosing a N_{circ} number of points. For each point a velocity vector is sampled.

3. Then the circulation is computed along the close circular path C, according to Eq. 7.3.

$$\Gamma = \oint_C \overline{Uds} \tag{7.2}$$

   In a discrete form, using quantities in the blade local frame of reference, it becomes:

$$\Gamma = \sum_{i=1}^{N_{circ}} U_{ax,i} * dax, i + \sum_{i=1}^{N_{circ}} U_{tg,i} * dtg, i \tag{7.3}$$

   Where *dax,i* and *dtg,i* are the gradients between the local coordinates of the circle points.

4. Using Biot-Savart the axial and tangential components of the induced velocity are computed. The velocity induced at the j-th point of the sampling line (ax_j,tg_j) by the point vortex centered at the AL point (ax_{ALP}, tg_{ALP}) is computed as follows:

$$r_j = \sqrt{(ax_j - ax_{ALP})^2 + (tg_j - tg_{ALP})^2} \tag{7.4}$$

$$U_{ind,ax,j} = \frac{\Gamma(tg_j - tg_{ALP})}{2\pi r_j^2} \tag{7.5}$$

$$U_{ind,tg,j} = -\frac{\Gamma(ax_j - ax_{ALP})}{2\pi r_j^2} \tag{7.6}$$

5. The induced velocity is subtracted to the sampled one for each line point. Thus, N_{samp} effective velocities are computed:

$$U_{eff,ax,j} = U_{samp,ax,j} - U_{ind,ax,j} \tag{7.7}$$

$$U_{eff,tg,j} = U_{samp,tg,j} - U_{ind,tg,j} \tag{7.8}$$

6. Finally, the average between all the line points is performed:

$$\begin{aligned} U_{eff,ax} &= \frac{\sum_{j=1}^{N_{samp}} U_{eff,ax,j}}{N_{samp}} \\ U_{eff,tg} &= \frac{\sum_{j=1}^{N_{samp}} U_{eff,tg,j}}{N_{samp}} \end{aligned} \tag{7.9}$$

The applied method is characterized by a 2D formulation, computing only the axial and tangential components of the induced velocity. Along the radial direction it is supposed to neglect the induction effect. Therefore the radial component of the velocity is simply calculated as the average between the sampled values on the line (Eq. 7.10).

$$U_{eff,rad} = \frac{\sum_{j=1}^{N_{samp}} U_{samp,rad,j}}{N_{samp}} \tag{7.10}$$

Actually this component is not used for computing the attack angle which is a purely 2D concept but the code requires to sample the three velocity components. The geometric parameters are reported in Tab. 7.3.

(a) Sampling line parameters

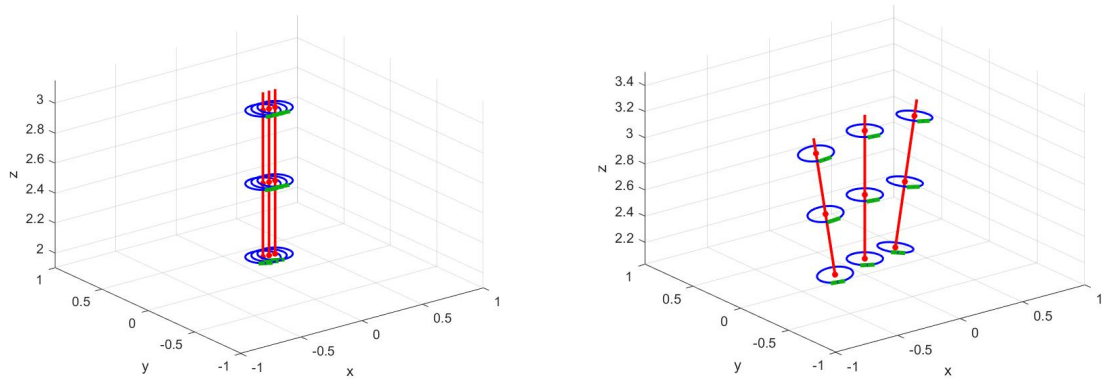
	Points number	**Length**	**Distance from ALP**
_vortex	Equal to intersections found	8Δ	10Δ

(b) Circle parameters to compute circulation

	Points number	**Radius**
_vortex	36	10Δ

Table 7.3: Geometrical parameters of the definitive vortex method.

The geometrical parameters of the line and circle are kept equal to the previous methods to make possible the results comparison. In Fig. 7.3 the sampling curves for three generic sections in case of surge and pitch motion only for the vertical blade are drawn.



(a) Surge (b) Pitch

Figure 7.3: Sampling technique applied to different platform motions.

The circles are depicted in blue while the upstream sampling line is green.
The code implementation of this method is completely described and reported in appendix A.

8 | Fixed-bottom case

The three alternative sampling techniques, previously described, are used to analyse the fixed-bottom HAWT in the load case LC1.1. To validate the sampling algorithms to be used with FOWT simulations, experimental evaluation is provided.
Before testing the new strategies a correction, on the original code, was applied on the force distribution along the blade. This is the reason why the blade trends of the LC11_lines are not exactly equal to the ones of LC11_E_d although the two methods are perfectly the same.

8.1. Integral loads

Overall rotor thrust, torque and power are computed from the code and here they are analysed. The blade thrust is calculated as in Eq. 8.1 by considering the lift and drag components perpendicular to the rotor plane. The blade torque is computed using the projections of lift and drag on the tangential direction multiplied by the radial coordinate, as shown in Eq. 8.2.

$$Thrust = \int_{rhub}^{r_{tip}} \overline{F_N}(r)dr = \int_{rhub}^{r_{tip}} [\overline{D}(r) \cdot sin(\phi(r)) + \overline{L}(r) \cdot cos(\phi(r))]dr \tag{8.1}$$

$$Torque = \int_{rhub}^{r_{tip}} \overline{F_T}(r) \cdot rdr = \int_{rhub}^{r_{tip}} [\overline{D}(r) \cdot sin(\phi(r)) + \overline{L}(r) \cdot cos(\phi(r))]dr \tag{8.2}$$

Where ϕ is the local inflow angle. Finally, the turbine thrust and torque are computed as the sum between the three blades. The power output is obtained as turbine torque multiplied by the rotational speed. In the AL code, these blade thrust and torque are computed as follows, starting from the lift and drag for each radial station:

```
scalar dalfaRad(dalfa*degToRad);
vector velRelDir(-cos(anglePhi-dalfaRad),sin(anglePhi-dalfaRad),0);
// vector (0,0,1) rotates lift perpendicular to
//relative wind velocity direction
lift = 0.5 * Cl * chord * VmagSect * VmagSect * velRelDir ^ vector(0,0,1);
drag = 0.5 * Cd * chord * VmagSect * VmagSect * velRelDir;
//- Blade point force in blade coordinates (tg,ax,rad)
vector forceBld(lift+drag);
thrust+=forceBld[1]*thick;
torque+=+forceBld[0]*thick*bladePtsStation[i];
```

Then, the sum between the three blades is performed.
Fig. 8.1 shows the thrust and torque trends versus the simulated time. As reported, 5 seconds of simulation (corresponding to 20 revolutions) are enough to reach a good convergence.

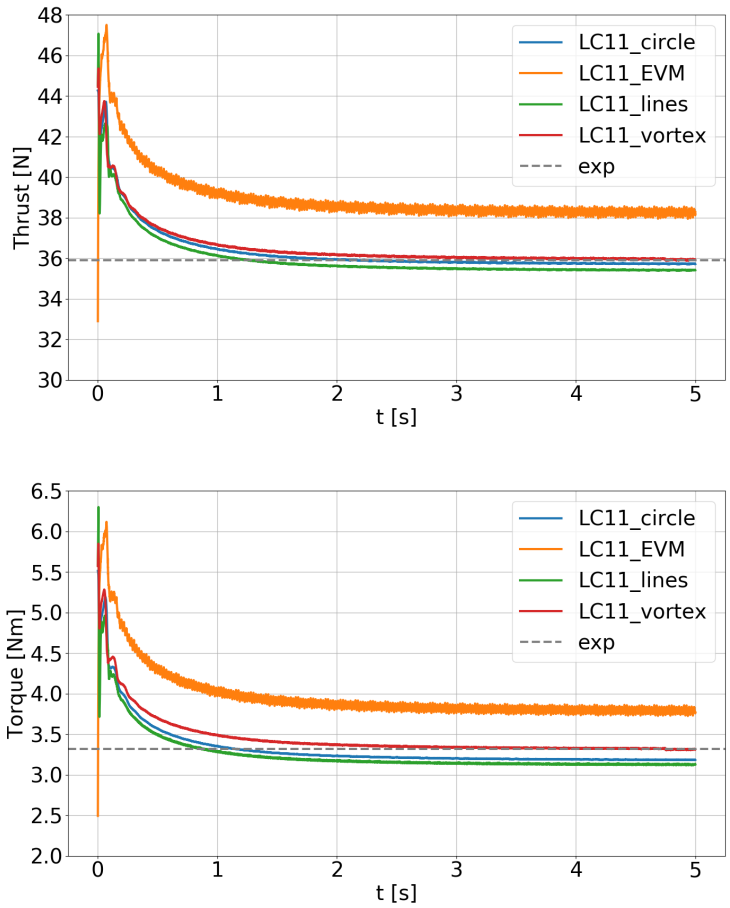
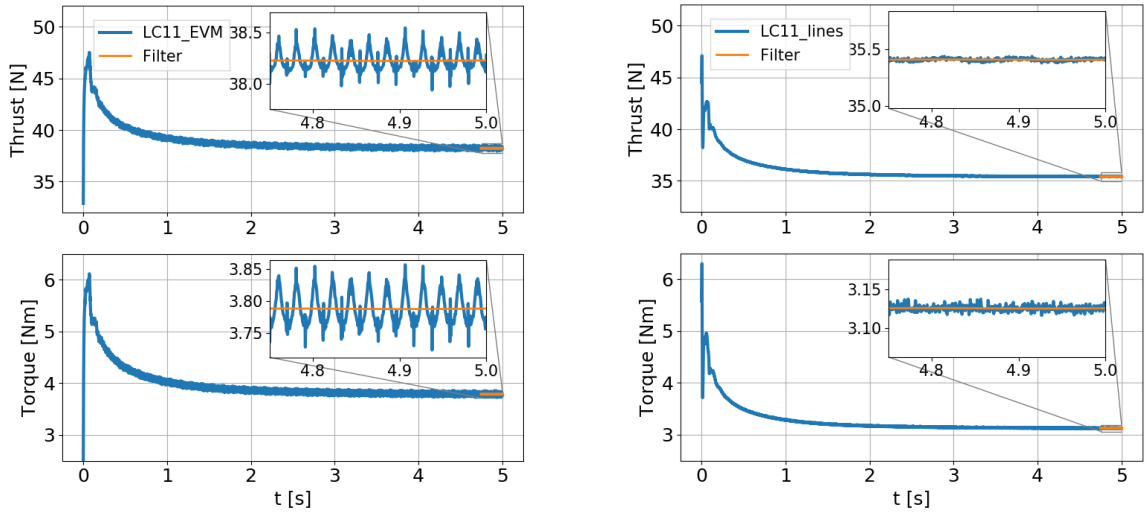


Figure 8.1: Thrust and torque convergence against simulated time for LC1.1 fixed-bottom case.

The simulation LC11_EVM consists in the fixed-bottom case performed with the original formulation of the AL model. The grey dashed line is the experimental mean value. From the figure it is possible to observed that the new methods present a more uniform trend in time with less oscillations. This is a positive aspect of the new formulations that seem to be more stable and capable of calculating more uniform integral quantities over a revolution. As previously mentioned, since the tower is not simulated and the flow is uniform there is no reason to have large thrust and torque variations during the single revolution. These oscillations are better shown in Fig. 8.2 where the LC11_EVM case is compared with the LC11_lines case.



(a) LC11_EVM

(b) LC11_lines

Figure 8.2: Thrust and Torque oscillations over time.

The zoomed area shows oscillations of the quantities obtained by the CFD-ALM code and the filtered signal (orange line), which has a cutoff frequency equal to the rotational frequency f = 4 Hz. The behaviour clearly demonstrates an unsteady fluctuation at the frequency associated with the passing of the blade, which is eliminated by the filter.
The final mean values are obtained by averaging over the last 5 revolution periods. They are compared with the experimental ones in Tab. 8.1. The experimental output contains significant noise by nature, so it has to be filtered with the same cut-off frequency especially to neglect the mass imbalance identified in the test model. Finally, the mean experimental value is the average extended to the whole time history of the collected signal.

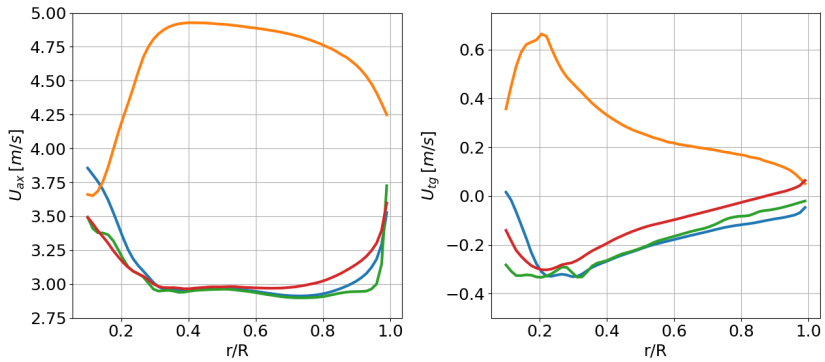
LC1.1	**Thrust [N]**	**Torque [Nm]**
EXP	35.91	3.32
_*EVM*	38.23	3.79
Error %	+6.45	+14.11
_*lines*	35.41	3.13
Error %	-1.39	-5.72
_*circle*	35.72	3.18
Error %	-0.52	-4.22
_*vortex*	35.93	3.31
Error %	+0.055	-0.3

Table 8.1: Rotor Thrust and Torque comparison with experimental data.

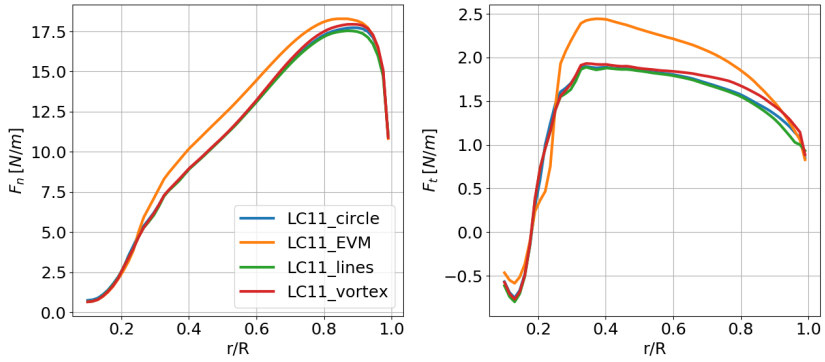
The original EVM presents the highest errors while all the other solutions can be considered in good agreement with experimental results. Better results could be obtained carrying out a more massive sensitivity analysis on the proposed sampling methods. It is possible to verify the outputs in case of variation of the sampling distance from the AL point. In any case, the obtained results are considered a good starting point being definitely in the uncertainty band of the experimental data.

8.2. Blade aerodynamics

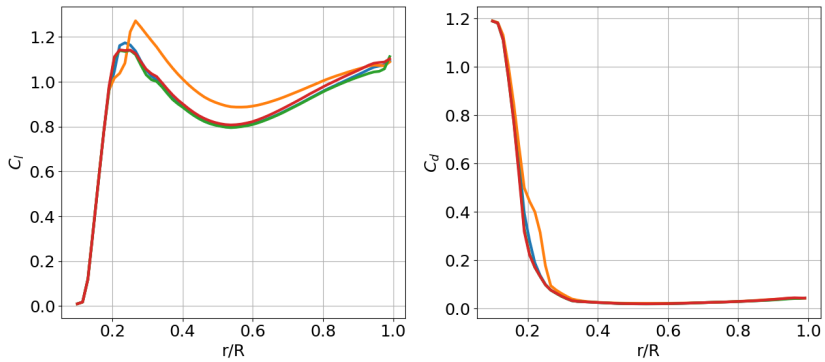
In this section the blade-distributed quantities are analyzed in order to assess if the numerical outcomes have a physical meaning. In particular, the focus is on the interconnection between the velocity sampled by the ALM and all the other quantities derived, such as angle of attack, blade forces and aerodynamic coefficients. In Fig. 8.3 the blade quantities for 60 stations along the blade span are shown. These curves are the averaged over the last five revolutions period for one blade, but fully consistent plots were found for the other two blades.



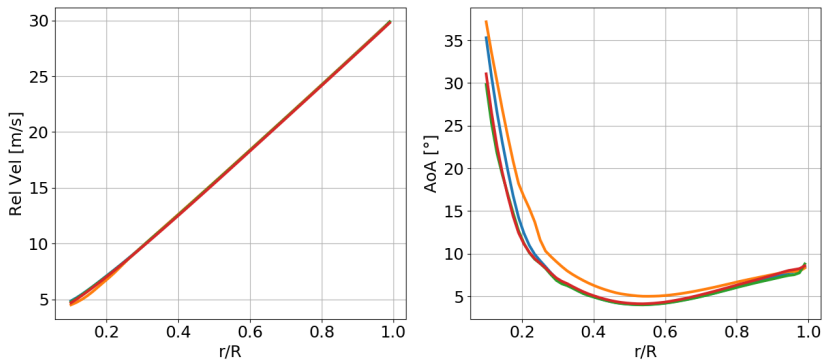
(a)



(b)



(c)



(d)

Figure 8.3: LC1.1 case - Spanwise distributed quantities.

All the trends depend on the sampled axial and tangential velocity components. Fig. 8.3(a) shows the velocities sampled by the different methods. The velocities obtained with the original EVM model are the ones obtained before applying the AoA correction, so they do not represent the physical induction upstream of the rotor as the velocity is sampled in the force distribution region where a U_{ax} peak is present. These speeds will therefore provide a strongly overestimated angle of attack. Nevertheless, the AoA is then corrected with the empirical formulation to finally obtain a trend in line with the other strategies. The new methods, instead, aim to already sample the correct velocity from which the AoA will be immediately determined, so the sampled speed must consider the induction effect upstream of the rotor. Therefore, axial speeds smaller than the free-stream velocity U_∞=4m/s correctly include this effect. The comparison between the new models demonstrates an overall consistency of the three methods. The main differences are concentrated at the root and at the tip regions:

- **Root region**: at the root, the *circle* method samples an higher axial velocity than the other two strategies. In this region, due to the high drag coefficient of the bluff cylindrical part, the fluid approaches the low-performance blade sections, which essentially produce passive torque. Between r/R=20% and r/R=30% the higher axial velocity results in slightly higher angles of attack with a consequent larger lift coefficient because the static stall angle is not reached. The blade aerodynamics close to the root is not adequately simulated because highly 3D flow and separation are not properly caught when boundary layer region is not resolved. The high values of C_l and C_d, between r/R=20% and r/R=30%, show that the ALM does not capture the typical separation/loss mechanism, in part because it lacks a root loss model that can minimise errors in those regions of complicated flow phenomena.
- **Tip region**: the blade tip region is of considerably greater significance, because there the strongest forces are transferred thanks to the high relative wind speeds and high angles of attack. The tangential component is not thoroughly studied because its weight is minimal towards the tip given the high peripheral speed. Therefore, the axial velocity component, that is crucial in determining the angle of attack, will be the subject of the analysis. *Circle* and *lines* sample the same U_{ax} from r/R=30% up to r/R=80%. After that the axial component starts to increase with different velocities. The *vortex*, instead, samples an increasing U_{ax} starting from r/R=60% with a smoother trend. All the three methods match one with each other at the very last stations. These small differences are simply related to the different sampling strategies that are adopted. The fact that the final AoA, forces and forces coefficients are, at the end, very well comparable demonstrates that the

methods can all be considered valid in any case. This increasing tip trend of U_{ax} can be explained considering that the volume force is increasing throughout the blade span until the severe drop in chord length close to the tip. Therefore, the force is lowered for the only influence of the chord because the tip loss correction is not activated, the regularisation kernel used in this study is strictly 2D and the force is applied on the same spreading area over the span (as it is parameterised with the cell dimension, which is constant in the rotor region). This reduction in force results in less deceleration of the flow and thus justifies the increase in U_{ax} at tip.

For this analysis, blade span quantities are not available by the two experimental campaigns, therefore the validation is performed through a comparison with some of the other OC6 participants. The institutes used for this comparison with the correspondent methods are:

- DTU (Technical University of Denmark): CFD with Vortex Filament
- NREL (National Renewable Energy Laboratory): BEM
- UNIFI (Università degli Studi di Firenze): Actuator Line Model.
- USTUTT (University of Stuttgart): Fully-resolved CFD.

Fig. 8.4 shows the results.

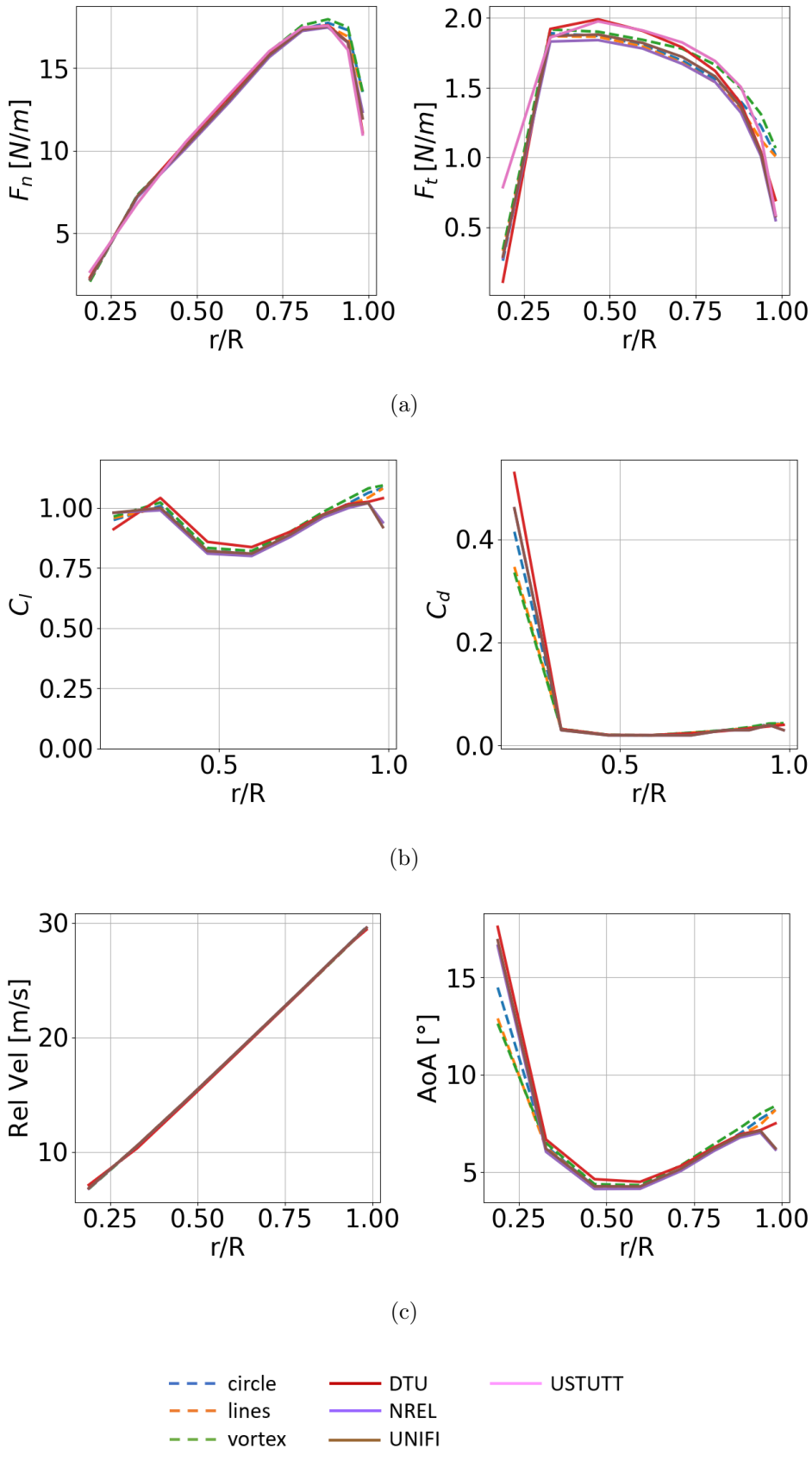
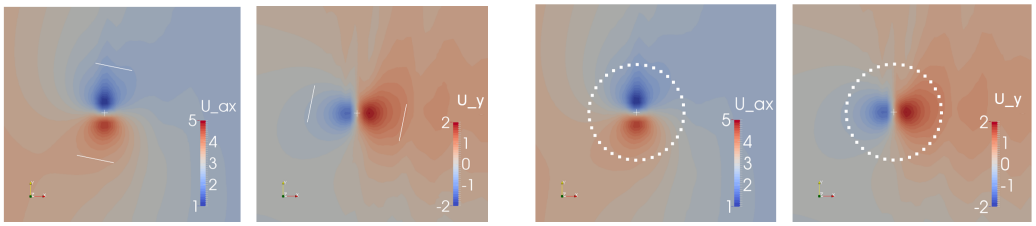


Figure 8.4: LC1.1 case - Spanwise distributed quantities. Comparison with OC6 results [26] and [2].

First of all, a good agreement with the other methods is visible for all the main quantities. As shown in Fig. 8.4(a) the main differences are located at the tip where the proposed three methods tend to over-estimate the normal and tangential force. Furthermore, a different trend is observed at the tip for the lift coefficient. For the actual methods, C_l increases while other participants obtain a reduction. These differences are reasonable considering that the tip losses are not implemented. If a loss coefficient were inserted, a more marked reduction of the forces at the tip and a corresponding reduction of the C_l should be obtained. Another observation can be provided for the attack angle which also has an increasing trend at the tip. The effect of tip loss insertion for the AoA is more complex to predict. Although a reduction of the forces at the tip and therefore an increase of the axial speed is expected, the consequent behavior of the angle of attack is not easily predictable being the flow three-dimensional. From experimental analyzes available in literature, a reduction of the angle of attack is expected, but for the specific case of the AL methods tested, conclusions will be reached only after having carried out this implementation.
The second region with the main differences is the hub. In particular, a different behaviour is visible for the attack angle. As already said, the novel sampling strategies are not able to correctly predict the axial velocity at the hub, being the flow highly 3D. Here the detachment of the flow and the trailing vorticity affect the sampling strategy and reduce the AL accuracy. Fortunately, this region has a very limited effect on the global performance of the turbine so it does not reduce the overall accuracy of the model. In any case, an improvement could be obtained by inserting a loss coefficient for this region as well.
Based on these considerations and on this comparison, the validity of the three new methods, in estimating blade aerodynamics, can be confirmed.

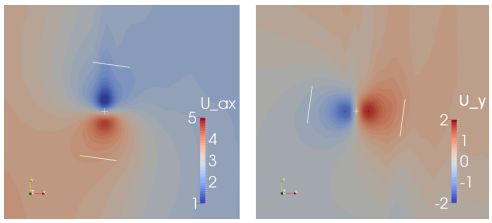
8.3. Flow field

In this section more details about the flow field features around the wind turbine rotor are provided. First of all, Fig. 8.5 and 8.6 show the U_{ax} and U_y components in the absolute reference frame for three blade sections at r/R = 40, 70, 95% with the correspondent sample method. It should be noted that the absolute velocity component U_y follows the absolute reference system of the fluid domain and therefore coincides with the inverse of U_{tg}. The approaches all provide consistent flow fields in accordance with the forces and angles of attack, as a further confirmation of the methodologies validity. At r/R=0.95 the tip vortex is observable despite it appears cut because the section plane is perpendicular to the blade axis.

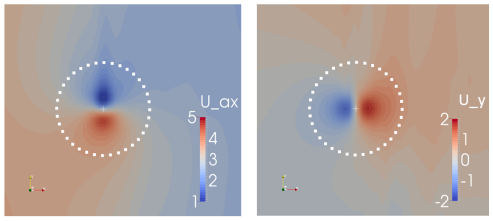


(a) LC1.1_lines: r/R=0.40

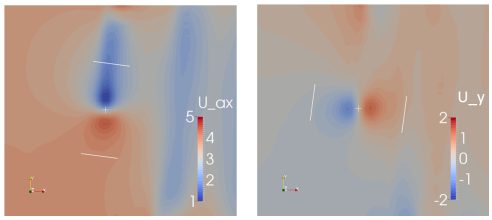
(b) LC1.1_circles: r/R=0.40



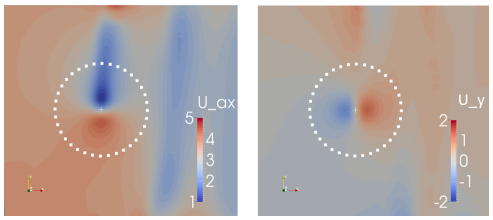
(c) LC1.1_lines: r/R=0.70



(d) LC1.1_circles: r/R=0.70

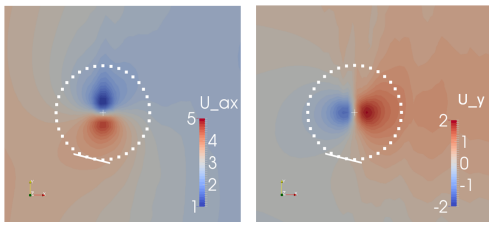


(e) LC1.1_lines: r/R=0.95

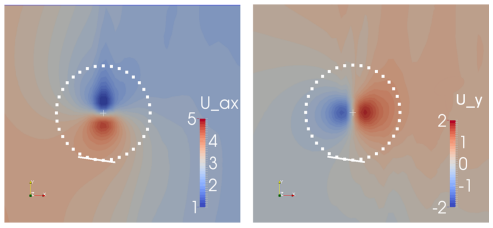


(f) LC1.1_circles: r/R=0.95

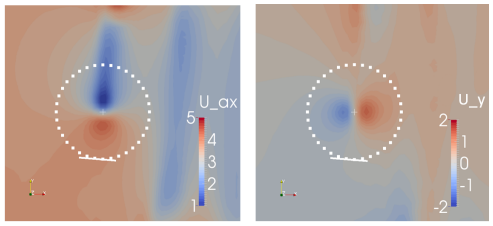
Figure 8.5: LC1.1_lines and LC1.1_circle: U_{ax} and U_y fields at blade sections 40, 70, 95% of the blade span.



(a) LC1.1_vortex: r/R=0.40



(b) LC1.1_vortex: r/R=0.70



(c) LC1.1_vortex: r/R=0.95

Figure 8.6: LC1.1_vortex: U_{ax} and U_y fields at blade sections 40, 70, 95% of the blade span.

As previously mentioned, the criterion chosen to validate the sampling method requires that the sampled velocity is as close as possible to the field velocity value at the AL point, where the induction effect is absent. To accomplish this verification, the velocities sampled by the code are compared with the interpolated fluid domain data extracted from the CFD-ALM post-processor. The first analysis is performed by comparing the axial velocity sampled along the actuator line with the interpolated velocity field along the same line. In Fig. 8.8 the sampled axial velocity is compared with the CFD field velocity obtained on three different lines:

- the central one coinciding exactly with the actuator line.
- one translated of +0.005 m on the plane of the rotor and parallel to the previous one.
- one translated of -0.005 m on the plane of the rotor and parallel to the previous one.

Before analyzing these curves, the lines layout is represented in Fig. 8.7. In this case the

vertical actuator line is considered.

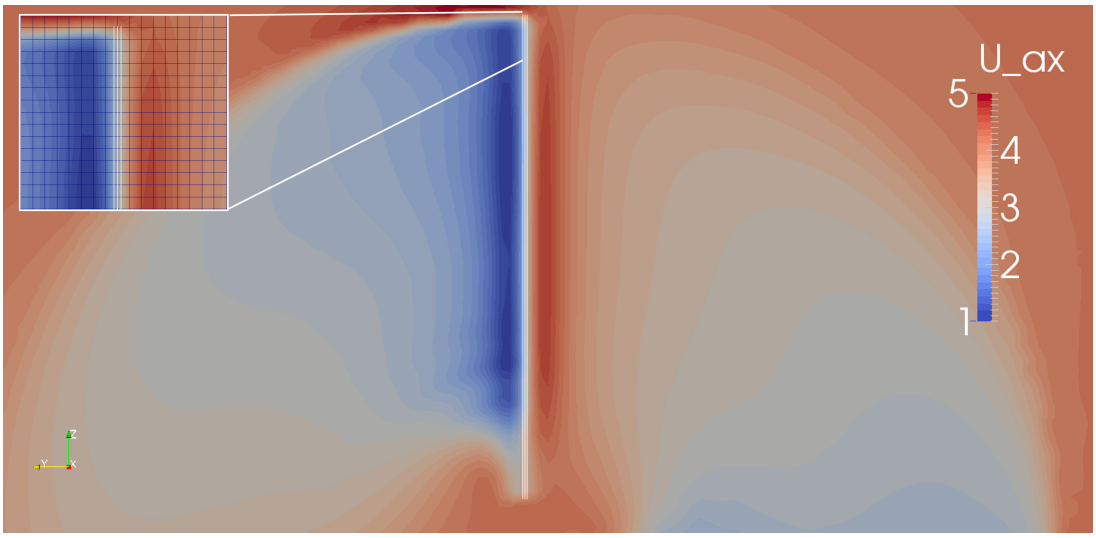
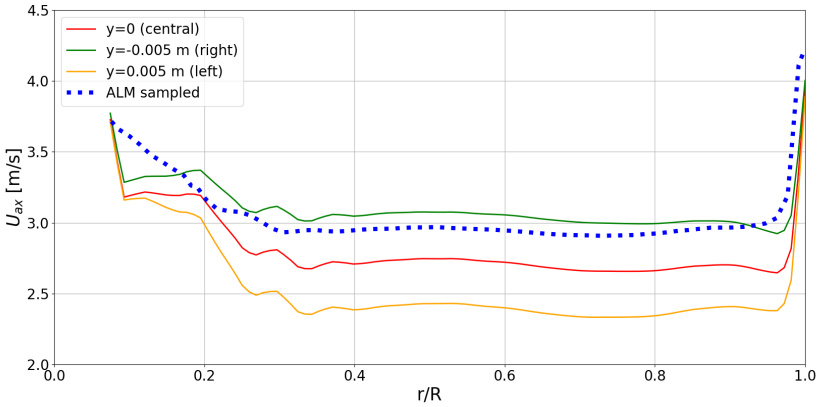
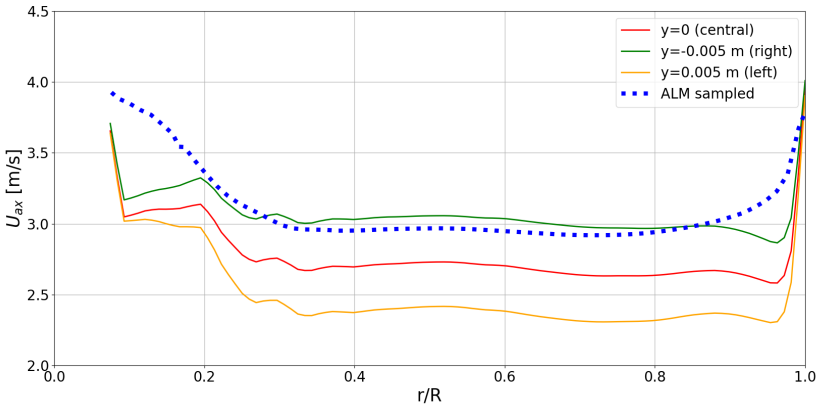


Figure 8.7: Disposition of three lines around the actuator line.

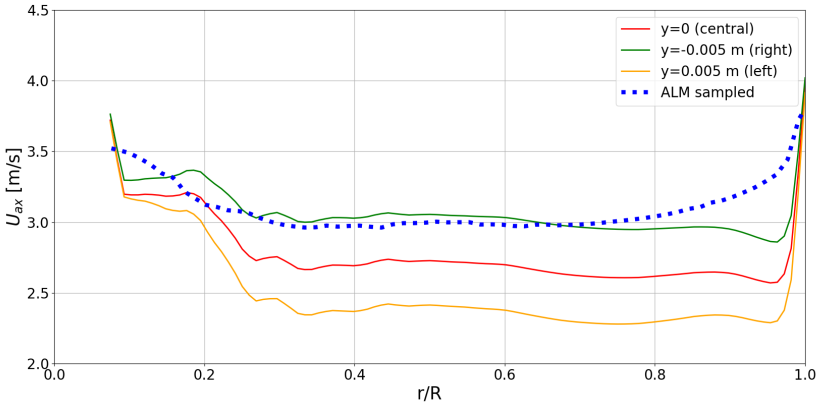
As shown in figure the lines are completely contained within a cell (being the characteristic cell dimension Δ=0.0174 m). This analysis is useful to understand how great is the velocity gradient around the AL points within a space that is lower than the cell size. The interpolated CFD velocity trend along each line is shown in Fig. 8.8.



(a) LC1.1_lines



(b) LC1.1_circle



(c) LC1.1_vortex

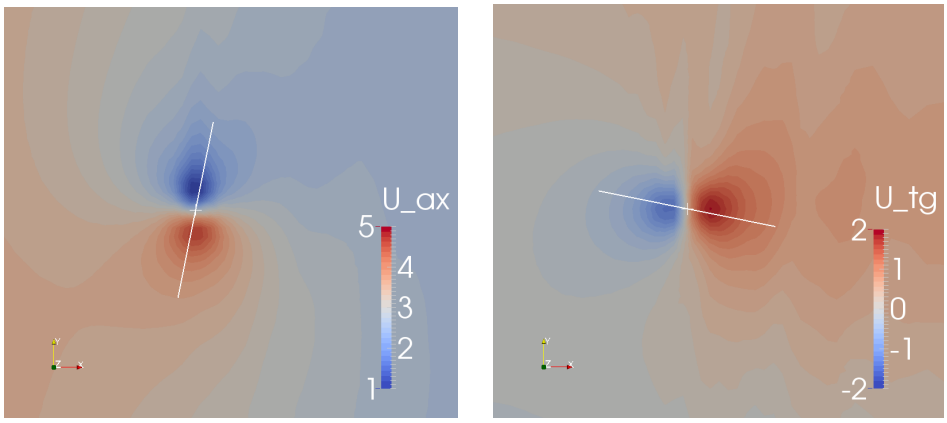
Figure 8.8: Interpolated CFD axial velocity along three lines, compared to the axial velocity sampled by the code.

This plot is intended to demonstrate how large the velocity change is around the actuator line. Therefore, it is not possible to imagine sampling the exact field velocity value (red line). The method is considered sufficiently accurate if it is able to sample a speed within this gradient. As reported, all methods sample in the range of interest except in the hub and tip regions. Leaving aside the sampling towards the hub where the presence of three-dimensional flows and the minimum contribution to the exchange of forces has already been discussed, the tip area is analyzed more in detail. *Lines* sampling seems to be the most faithful to the flow field values. However, the method with the *circles* also behaves as desired up to r/R=80% and then takes on a smoother trend of U_{ax} that seems to be more consistent with the reduction of forces towards the tip and therefore can be considered valid from a physical point of view. *Vortex* sampling shows a smooth increasing trend starting from r/R=70%. This method, unlike the others, is based on a different principle which requires the calculation of a physical quantity to correct the sampled speed. So a slightly different trend is expected. Probably a specific sensitivity analysis on the radius of the circumference for the circulation computation and on the distance of the sampling line would make bring the U_{ax} obtained with the vortex closer to that obtained with the other two strategies. In this work the goal is to demonstrate that different sampling techniques can be applied. Due to too long computational times, the improvement of these techniques with specific sensitivity analyzes is left to future works. The chosen methods are based on different strategies, so it is justified that they behave differently in the tip region where the flow takes on particular characteristics and sampling becomes very sensitive. Furthermore, there is no evidence of the fact that the best velocity trend, to obtain an excellent estimation of the blade loads, is exactly the one obtained in correspondence with the AL point. The mesh used is poorly refined and therefore, makes the resolution of the bound vortex inaccurate. For this reason, it can not be stated a priori that a local sampling method is necessarily the best. Indeed, observing the axial velocity curve on the actuator line (red curve) it can be seen that there is a little nonphysical trend towards the tip. The velocity reduction, before the final increase, does not properly track the force distribution. Additionally, the red curve is obtained through the interpolation process applied by Paraview. Since the mesh is poorly refined, errors may also be present in this passage.

In conclusion, the fact of having obtained sampled velocities close to the filed values but with more physical and coherent trends with the distribution of forces, in any case suggests the great validity of the tested methods.

Another way to validate the models is achieved by comparing the axial and tangential components of the sampled velocity with their field trends on the plane transversal to the blade. For the three previously mentioned sections (40,70,95%) the interpolated velocity

field is fetched along a line normal and parallel to U_{rel} direction along which axial and tangential velocities are respectively computed. Fig. 8.9 shows the lines, for a generic section, from which the trends represented in Fig. 8.10 and 8.11 are obtained. The U_{ax} is analyzed along the straight line parallel to the relative velocity while the U_{tg} on the perpendicular one.

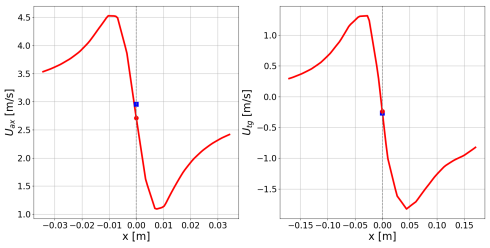


(a) Line $\parallel$ U_{rel} - where U_{ax} is analyzed (b) Line $\perp$ U_{rel} - where U_{tg} is analyzed

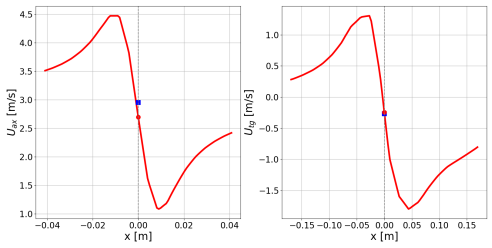
Figure 8.9: Lines parallel and perpendicular to U_{rel}.

As expected, for U_{ax} the values sampled by the codes are slightly higher than the interpolated ones, while they perfectly match in the case of U_{tg}. Despite the difference shown in this figure for U_{ax}, the previous analysis of Fig. 8.8 is useful to still consider the sampling methods valid.

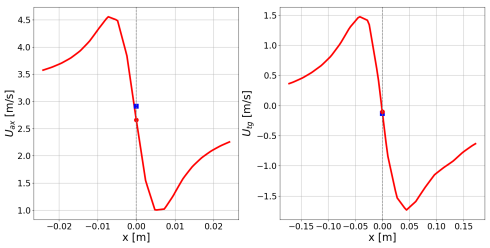
Once the validation process of the presented AL methods is satisfied for the fixed-bottom case, their application in case of surge or pitch platform motion is analyzed and described in the following sections.



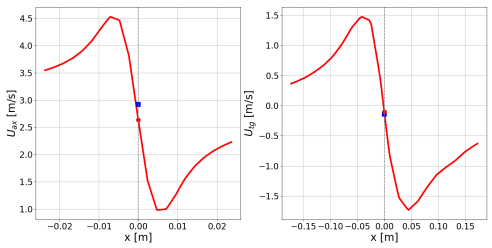
(a) LC1.1_lines: r/R=0.40



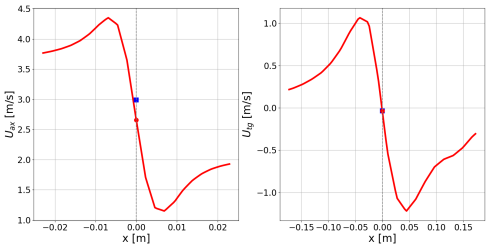
(b) LC1.1_circles: r/R=0.40



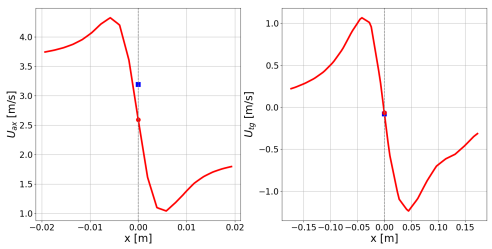
(c) LC1.1_lines: r/R=0.70



(d) LC1.1_circles: r/R=0.70



(e) LC1.1_lines: r/R=0.95



(f) LC1.1_circles: r/R=0.95

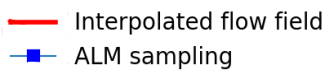
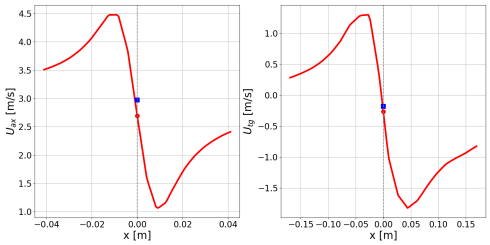
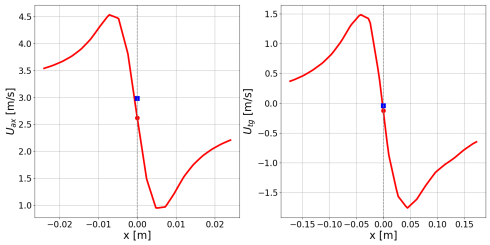


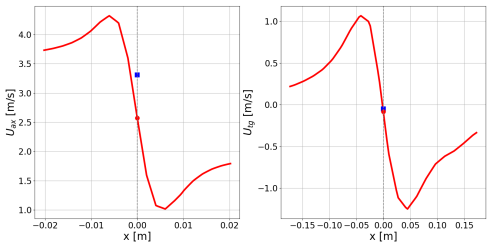
Figure 8.10: LC1.1_lines and LC1.1_circle: axial and tangential velocities sampled along a line passing by the ALP (red line) compared with velocity components given by the ALM solver (blue dots).



(a) LC1.1_vortex: r/R=0.40



(b) LC1.1_vortex: r/R=0.70



(c) LC1.1_vortex: r/R=0.95

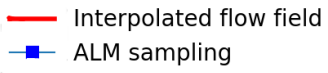


Figure 8.11: LC1.1_vortex: axial and tangential velocities sampled along a line passing by the ALP (red line) compared with velocity components given by the ALM solver (blue dots).

9 | Surge platform motion case

Surge hub displacement is implemented as a function of time (Eq. 9.1).

$$x_s(t) = A_s sin(2\pi f_s t + \Phi_s) \tag{9.1}$$

For this simulation, a phase shift Φ_s=180° is applied to first move the turbine forward, along the negative x coordinate. The hub velocity is computed as the displacement time derivative (Eq. 9.2):

$$\dot{x_s}(t) = -A_s 2\pi f_s cos(2\pi f_s t) \tag{9.2}$$

A direct relationship is applied for the apparent velocity definition over time due to the rigid motion imposed along the x-axis. The rotor is exactly perpendicular to the flow therefore each blade point is subjected to the same velocity:

$$U_{app}(t) = U_\infty - \dot{x_s}(t) \tag{9.3}$$

In the specific tested case, LC2.5, the frequency is f_s=1 Hz and the amplitude is A_s=0.035 m. A graphical plot of the hub position and apparent velocity trend is reported in Fig. 9.1. A maximum thrust and torque is expected in the first half a period when the rotor moves upward along negative x, since the observed relative velocity is maximised. The opposite occurs half a period later, when the rotor is travelling along the wind direction. At $\frac{1}{4}T_{surge}$ and $\frac{3}{4}T_{surge}$, when the translation velocity is null, fixed-bottom quantities are expected.

A 7° azimuthal phase shift is imposed, and the free stream velocity is set to 4 m/s as in the fixed-bottom example. Numerical simulations were run until periodic trends of thrust and torque were reached. It should be emphasised that the PIMPLE algorithm with two outer correctors was useful in achieving convergence in all key turbine parameters with barely perceptible numerical fluctuations during the platform motion phase. Nevertheless,

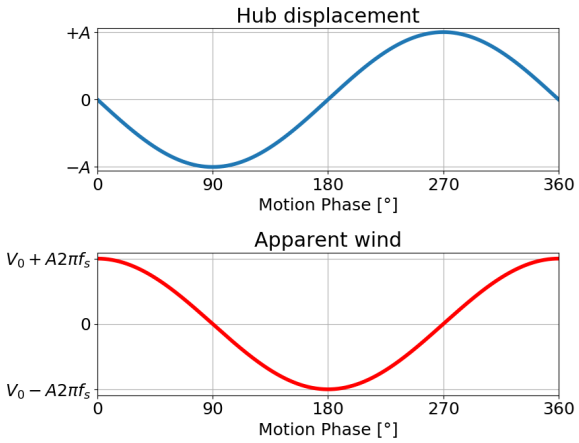


Figure 9.1: Hub displacement and apparent wind in the surge case.

a 1 Hz frequency filter was applied to neglect blade passing frequency. As proof of the sample velocity methodology validity in FOWT conditions, the two test campaigns of GVPM experiments at Politecnico di Milano are used as comparison.

Fig. 9.2 shows thrust and torque convergence in time for the three methods.

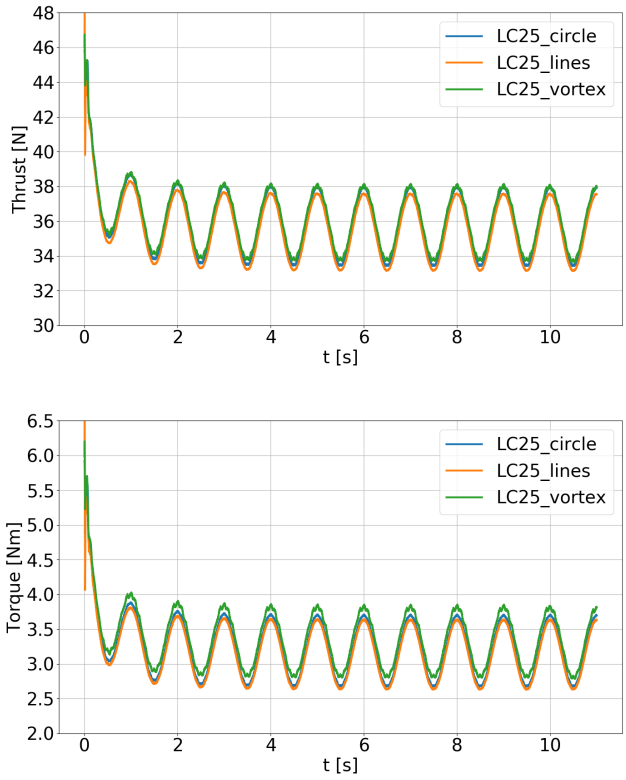


Figure 9.2: Thrust and torque convergence against simulated time for surge case.

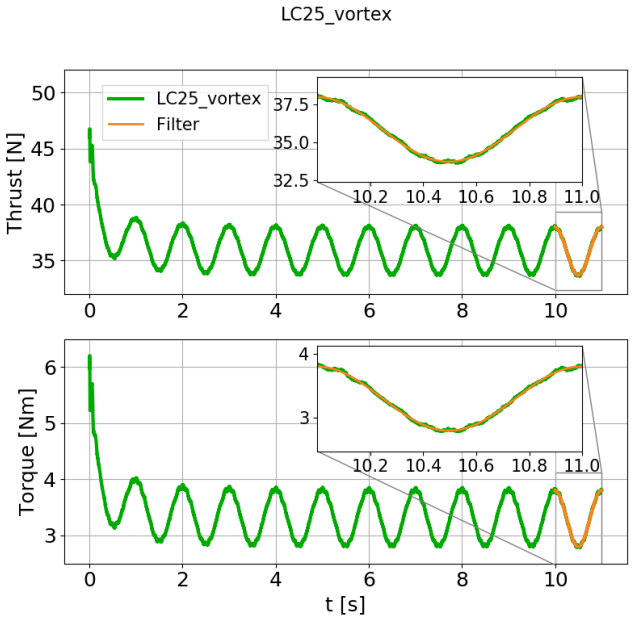


Figure 9.3: LC2.5_vortex: thrust and torque convergence and filtered data.

For each simulation the filter is applied on the last period. In Fig. 9.3 only the filter applied at the LC2.5_vortex case is reported as an example.

Fig. 9.4 shows thrust and torque evolution over one complete surge period T_{surge}= 1 s. The numerical results are compared with the two available experimental data and show how the expected trend is completely satisfied. The maximum and minimum thrust and torque is obtained when the correspondent maximum and minimum apparent velocity is achieved. At $\frac{1}{4}T_{surge}$ and $\frac{3}{4}T_{surge}$ when the rotor is at the two extremes with a null translation velocity, the fixed-bottom case quantities are reached (dotted lines). This indicates that the platform frequency and amplitude, here considered, do not change the mean operation of the turbine with respect to a fixed-bottom condition.

A quantitative comparison with measurements is performed comparing the mean and amplitude values of thrust and torque. The amplitude is intend as the difference between the maximum and the minimum closest peak for the filtered signal. In Tab. 9.1 these values are compared computing also the percentage errors:

$$errMean = \frac{|mean(newALM) - mean(EXP)|}{mean(EXP)} \tag{9.4}$$

$$errAmpl = \frac{|ampl(newALM) - ampl(EXP)|}{mean(EXP)} \tag{9.5}$$

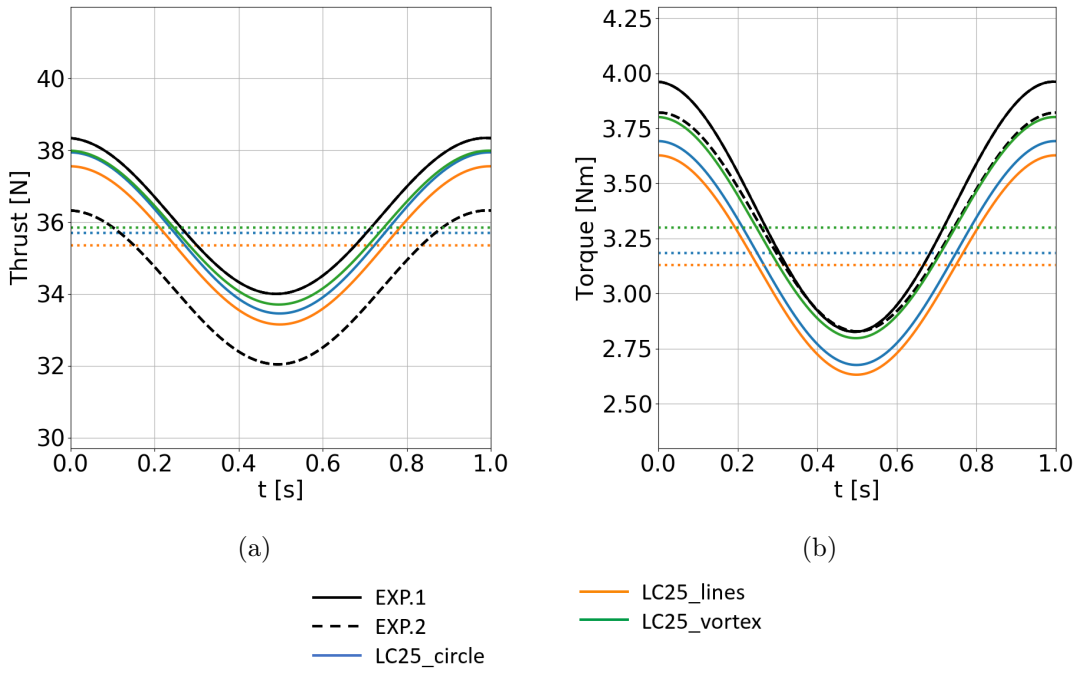


Figure 9.4: Filtered thrust and torque over one surge period.

	Thrust [N]		Torque [Nm]	
Case	**Mean**	**Ampl.**	**Mean**	**Ampl.**
EXP1	36.17	4.34	3.39	1.136
EXP2	34.18	4.28	3.32	0.994
_*lines*	35.35	4.40	3.129	0.996
Error % EXP1	-2.26	+0.17	-7.796	-4.13
Error % EXP2	+3.42	+0.35	-5.876	+0.06
_*circle*	35.7	4.48	3.18	1.017
Error % EXP1	-1.308	+0.39	-6.19	-3.51
Error % EXP2	+4.43	+0.59	-4.24	+0.69
_*vortex*	35.847	4.28	3.3	1.00
Error % EXP1	-0.897	-0.17	-2.79	-4.01
Error % EXP2	+4.865	-0.012	-0.764	+0.18

Table 9.1: LC2.5: Rotor thrust and torque comparison with experimental data.

A good agreement between experiments and numerical results is obtained for both thrust and torque. Regarding the thrust, all the simulations give results included between the two tests campaigns. Indeed, the percentage errors are within -2.5% and $+5\%$ and so

demonstrate the great validity of the new techniques in surge case. Also, the thrust absolute disparities are of the order of 2 N, which may be regarded as being extremely small and acceptable given the uncertainty in the load cell located on the turbine tower and the system inertia problems experienced in earlier tests.
As far as torque, instead, a strong difference in the errors with Experiment 1 and 2 is observed. In particular, the numerical simulations are in better agreement with Experiment 2 for both mean and amplitude values with error within -6% and $+0.7\%$. During the validation phase, these results were compared with those of the other participants in the OC6 project who used CFD methods. In particular, the institutes taken into consideration, with their methods, are:

- DTU (Technical University of Denmark): CFD with Vortex Filament
- NREL (National Renewable Energy Laboratory): BEM
- UNIFI (Università degli Studi di Firenze): Actuator Line Model.
- USTUTT (University of Stuttgart): Fully-resolved CFD.

It was observed that all the participants achieved curves more coherent with Experiment 2 test campaign as shown in Fig. 9.5 and Fig. 9.6.

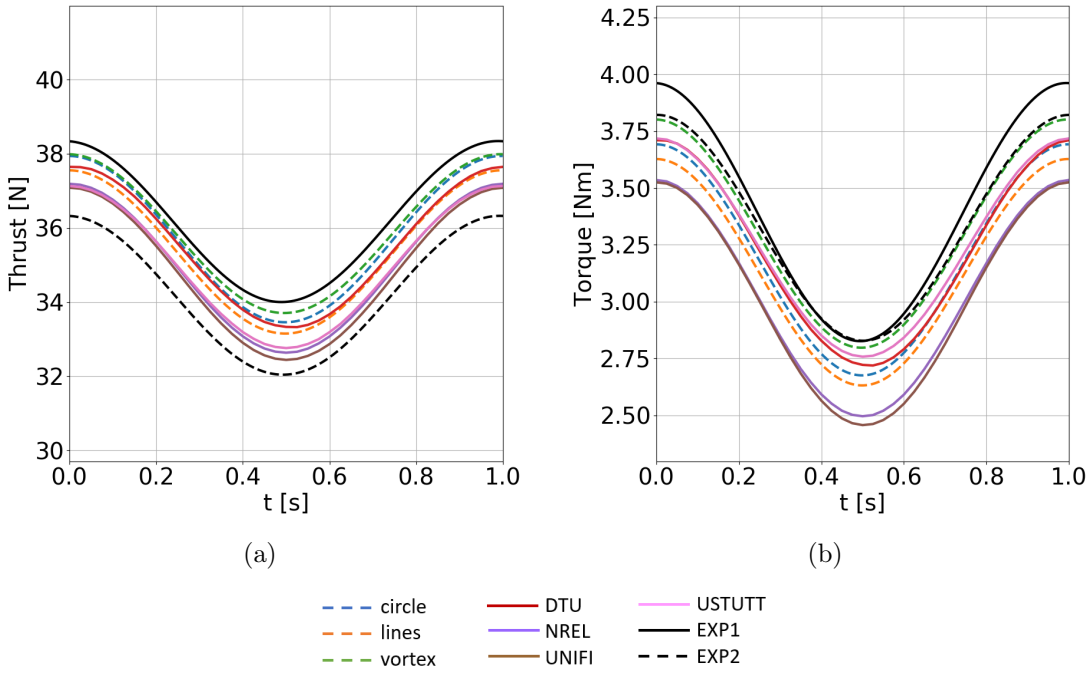
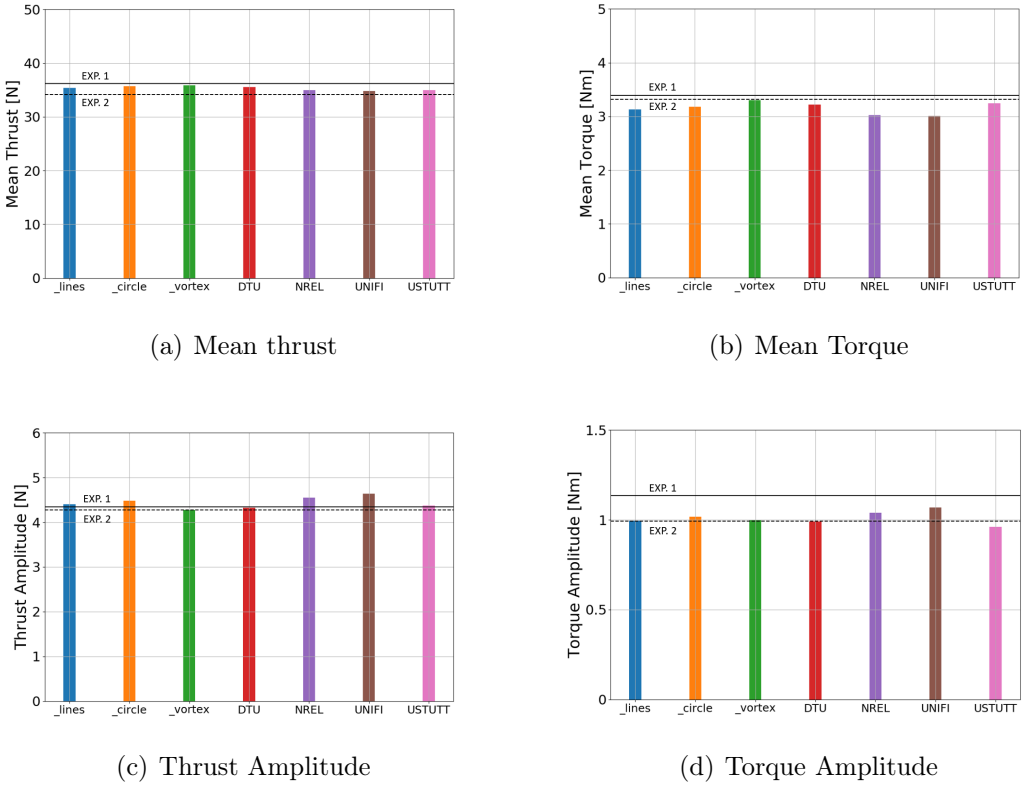


Figure 9.5: Thrust and torque over one surge period. Comparison with other participants to OC6 project [26] and [2].



(a) Mean thrust

(b) Mean Torque

(c) Thrust Amplitude

(d) Torque Amplitude

Figure 9.6: Mean and amplitude values. Comparison with other participants to OC6 project [26] and [2].

Especially for the torque, all the institutes have results more in line with Experiment 2. The error between the average mean torque of the three new strategies and the average mean torque of the OC6 participants is 2.5%. Instead, for the torque amplitude the error is 0.33%. Therefore, the new methods can be considered valid if compared with the others. The origin of this discrepancy has to be seek in a setting or measurement error of the experimental campaign 1 and not in a lack of the proposed methods. During the phase III of OC6 a problem in the Experiment 1 was identified in another surge case tested. In the set up of the experiment a controller on the rotational speed of the rotor was used. The probably too loose control logic allowed the rotor to undergo variations in the rotation speed during the surge motion, consequently generating variations in the torque amplitude. Considering that in the LC25 case the apparent speed is even higher, one can imagine that a similar problem arises in this case as well. For the above reasons, the difference with respect to Experiment 1 does not reduce the validity of the methods.

However, in order to more clearly appreciate how the inflow variation influences the oscillation of integral quantities, the discussion needs to shift to the span wise distribution

of velocities and angles of attack.

In Fig. 9.4 the physical oscillations of the integral quantities demonstrate the capability of the three methods to perceive the surge motion and so the apparent wind variation. Anyway, a deeper analysis is made by looking at the trends of some important quantities over a complete surge period for three different blade sections. To make the explanation less repetitive, this analysis is proposed only for the LC2.5_vortex case, being the one that showed greater consistency with the experimental data. For the other two cases considerations and results are quite similar.
First of all, in Fig. 9.7 the trends of the most important quantities over a surge period are shown for the 30%, 60% and 75% of the rotor radius. The main effects on blade performances are not related to the variation of relative speed, but mostly due to the variability of the angle of attack. The maximum axial translation velocity is, in this case, 0.2 m/s; it is equivalent to 5% of the free stream speed so it is not sufficient to act significantly on the relative speed perceived by the rotating blades, which is in the order of tens of m/s. On the contrary, the attack angle is particularly affected by the variation of the apparent wind and consequently generates strong oscillations also on C_l and partially on C_d.

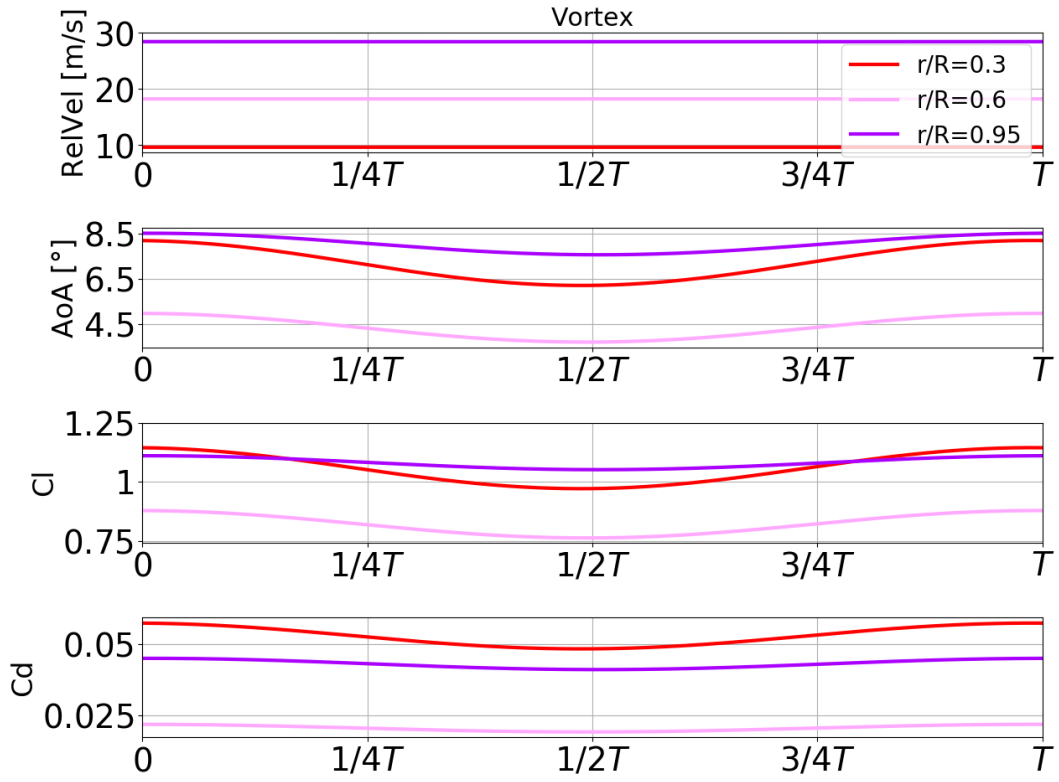


Figure 9.7: LC2.5_vortex: Aerodynamic quantities over a surge period.

Regarding the attack angle, at the root, the smaller peripheral speed makes the effect of the apparent wind variation more preponderant. Conversely, at the tip, the peripheral velocity is much grater than the axial one therefore the surge motion effect is not visible. At 30% of the span, the blade is not yet in stalled condition and the flow is still attached. At lower radial coordinates the blade works in stalled or low efficiency conditions, the angle of attack has stronger oscillations but less impact on blade performances. Since all the three sections are working in attached flow conditions, the same AoA oscillations recur in the C_l and C_d trends. These curves are coherent with what expected from a physical point of view so they further confirm the good working of the tested methods in surge conditions.
At this point another verification on the sampled velocity is performed. In order to obtain the real apparent velocity acting on the rotor, a motion composition between the flow field and the surge velocity must be applied. On the computational flow field, indeed, the surge motion is not visible because the translation rotor velocity is not considered when the cell center value is computed. The translation velocity is inserted before computing the AoA, when it is subtracted to the sampled one before computing the angle:

$$\begin{aligned} U_{ax,eff} &= U_{ax,samp} - U_{ax,trasl} \\ U_{tg,eff} &= U_{tg,samp} - U_{tg,trasl} \end{aligned} \tag{9.6}$$

$\overline{U_{trasl}}$ includes all the blade movements (rotation, surge and pitch) so it is fundamental to insert the surge effect in the attack angle and consequently in all the other aerodynamic quantities. In Fig. 9.8 the real velocity perceived by the rotor $U_{ax,eff}$ is shown for 4 time steps evenly spaced during the platform surge period, which is equivalent to 1 s. As expected, a maximum and a minimum value is reached at 0 and $\frac{1}{2}$ T, while the fixed-bottom condition is obtained at the two motion extremes. The amplitude between the maximum and minimum value with respect to the fixed-bottom one, for each radial coordinate, is about 0.2 m/s, which is exactly the one of the apparent wind. In this way, although the sampling strategy is not able to estimate the surge effect, the correction applied before the AoA computation is enough to include the apparent wind variation in all the aerodynamic parameters.

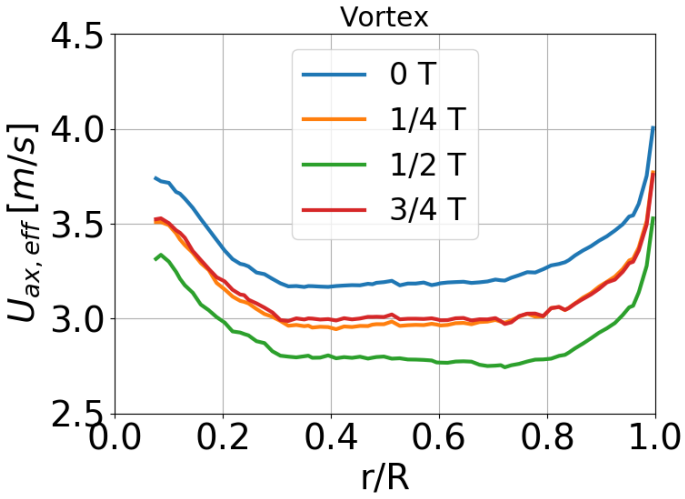


Figure 9.8: LC2.5_vortex: Real axial velocity during the surge period T.

10 | Pitch platform motion case

The pitch movement is applied with a sinusoidal motion law at the pitch angle:

$$\theta_p(t) = A_p \sin\left(2\pi f_p t + \Phi_p\right) \tag{10.1}$$

The pitch center is placed at the same height of the tower bottom which is close to the conditions encountered by FOWT exploiting barge of semi-submersed layouts. The tower bottom is not aligned with the hub because the tower is tilted forward by the same amount of the up-tilt nacelle angle, to obtain a perfectly vertical rotor in its starting configuration. The initial turbine position is shown in Fig. 10.1.

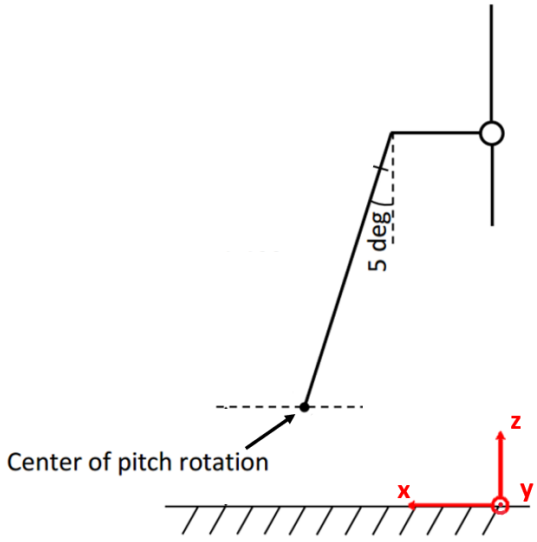


Figure 10.1: Turbine initial position.

The following set of equations, where the *cp* subscript denotes the coordinate of the pitch centre, can be used to describe the hub position over time:

$$\begin{cases} x_{hub}(t) = x_{hub,0} + (x_{hub} - x_{cp}) \cdot (cos\theta_{pitch} - 1) + (z_{hub} - z_{pitch}) \cdot \sin\theta_{pitch} \\ y_{hub} = y_{hub,0} \\ z_{hub}(t) = z_{hub,0} + (x_{hub} - x_{cp}) \cdot (-\sin\theta_{pitch}) + (z_{hub} - z_{pitch}) \cdot (cos\theta_{pitch} - 1) \end{cases} \tag{10.2}$$

In this case, LC3.5, the amplitude is set to $A_p = 1.4°$ and the frequency is $f_p = 1Hz$. The hub motion starts backward, along the positive x coordinates. In Fig. 10.2 the sinusoidal pitch angle, hub x displacement and apparent wind are shown.

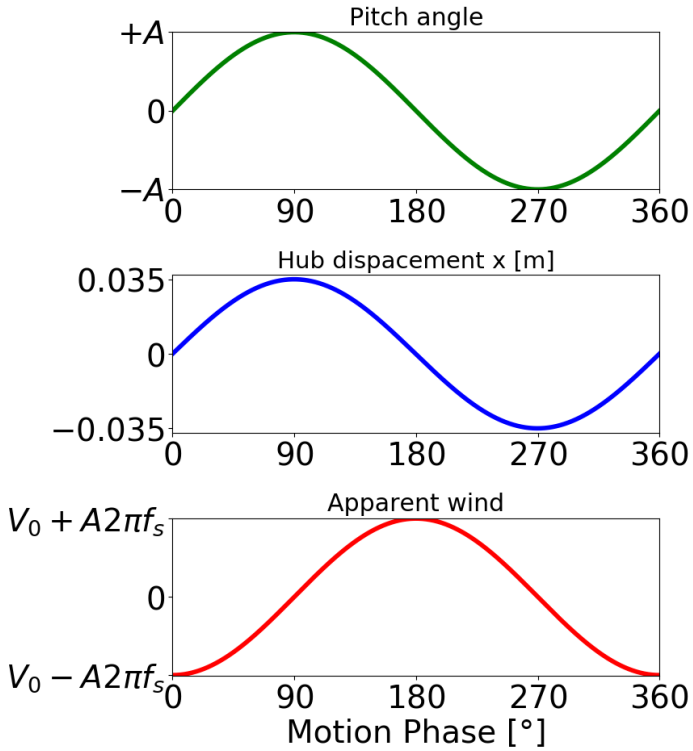


Figure 10.2: Pitch angle, hub displacement and apparent wind in the pitch case.

At 180° of the pitch phase the apparent axial velocity is the highest and, therefore, a peak of attack angle and consequently of loads is expected. As in the surge case, the simulations were run until periodic trends of thrust and torque were reached. The sinusoidal response of the ALM appears free of high-frequency oscillations thanks to the use of the PIMPLE algorithm that facilitates convergence, as shown in Fig. 10.3 where thrust and torque convergence in time for the three methods are reported. Nevertheless, a 1 Hz frequency filter was applied on the last period to neglect blade passing frequency. In Fig. 10.4 the filter enforced to the LC3.5_vortex is reported as an example.

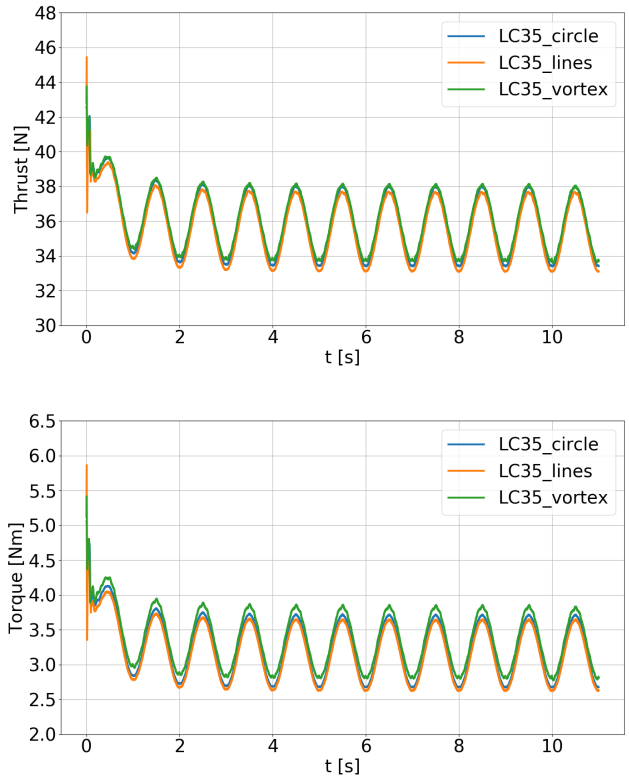


Figure 10.3: Thrust and torque convergence against simulated time for pitch case.

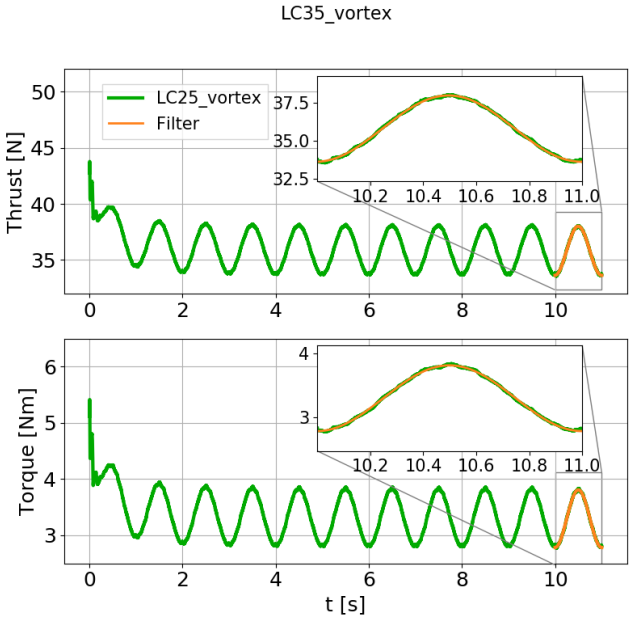


Figure 10.4: LC3.5_vortex: thrust and torque convergence and filtered data.

As proof of the sampling velocity methodology validity in FOWT conditions, the test campaign "Experiment 2" performed in the GVPM at Politecnico di Milano is used as comparison. This is the only experimental campaign in which the pitch motion was tested. Fig. 10.5 shows thrust and torque evolution over one complete pitch period T_{pitch}= 1 s. As expected, the maximum and minimum thrust and torque is obtained when the correspondent maximum and minimum apparent velocity is achieved. At $\frac{1}{4}T_{pitch}$ and $\frac{3}{4}T_{pitch}$, when the rotor is at the two extremes with a null translation velocity, the fixed-bottom case quantities are reached (dotted lines). This indicates that the platform frequency and amplitude here considered do not change the mean operation of the turbine with respect to a fixed-bottom condition.

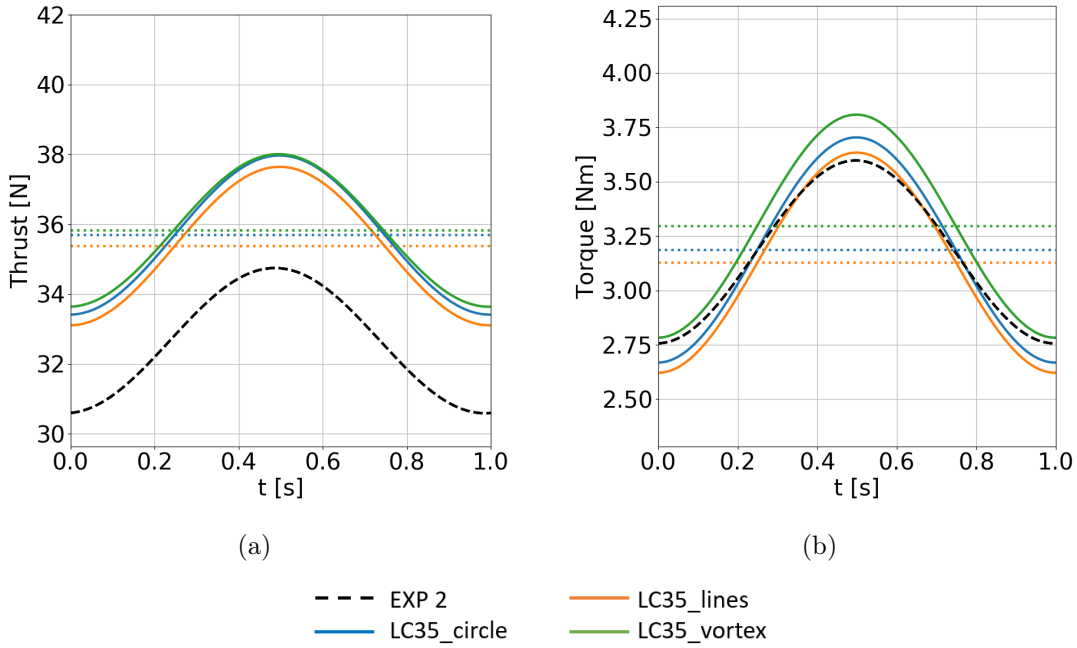


Figure 10.5: Filtered thrust and torque on one pitch period.

As performed in the previous chapter, a quantitative comparison with measurements is done comparing the mean and amplitude values of thrust and torque, also computing the correspondent errors. In Tab. 10.1 the comparison is exposed. The simulated pitch mean values for both thrust and torque are equal to the fixed-bottom ones as expected. Regarding the experiment, a great reduction of the mean values, with respect to the one of the fixed-bottom case, is observed, especially for the mean thrust. The greater errors are those computed for the mean thrust and for the torque amplitude. Anyway, also for the pitch case, during the methods validation phase, the results were compared with those obtained by the other CFD methods that participated in the OC6 project. The available simulations for the pitch case are:

- DTU (Technical University of Denmark): CFD with Vortex Filament
- NREL (National Renewable Energy Laboratory): BEM
- UNIFI (Università degli Studi di Firenze): Actuator Line Model.

	Thrust [N]		Torque [Nm]	
Case	**Mean**	**Ampl.**	**Mean**	**Ampl.**
EXP2	32.67	4.15	3.18	0.84
_*lines*	35.37	4.54	3.13	1.012
Error % EXP2	+8.29	+1.17	-1.56	+5.4
_*circle*	35.69	4.56	3.186	1.035
Error % EXP2	+9.25	+1.22	+0.28	+6.12
_*vortex*	35.82	4.37	3.296	1.03
Error % EXP2	+9.66	+0.65	+3.73	+5.81

Table 10.1: LC3.5: Rotor thrust and torque comparison with experimental data.

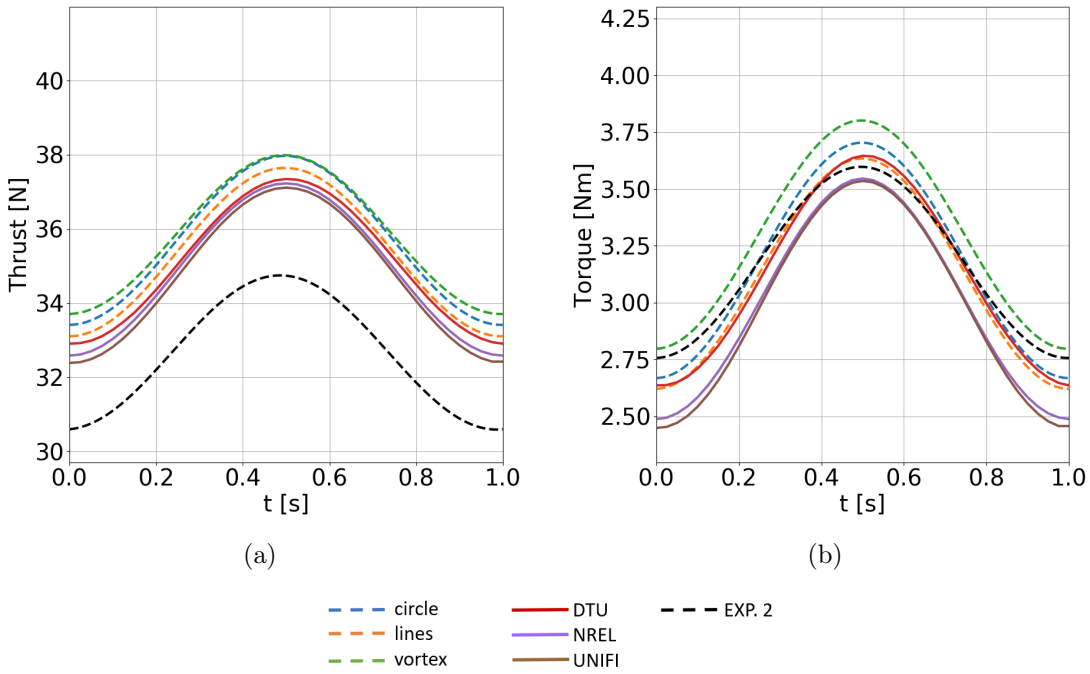
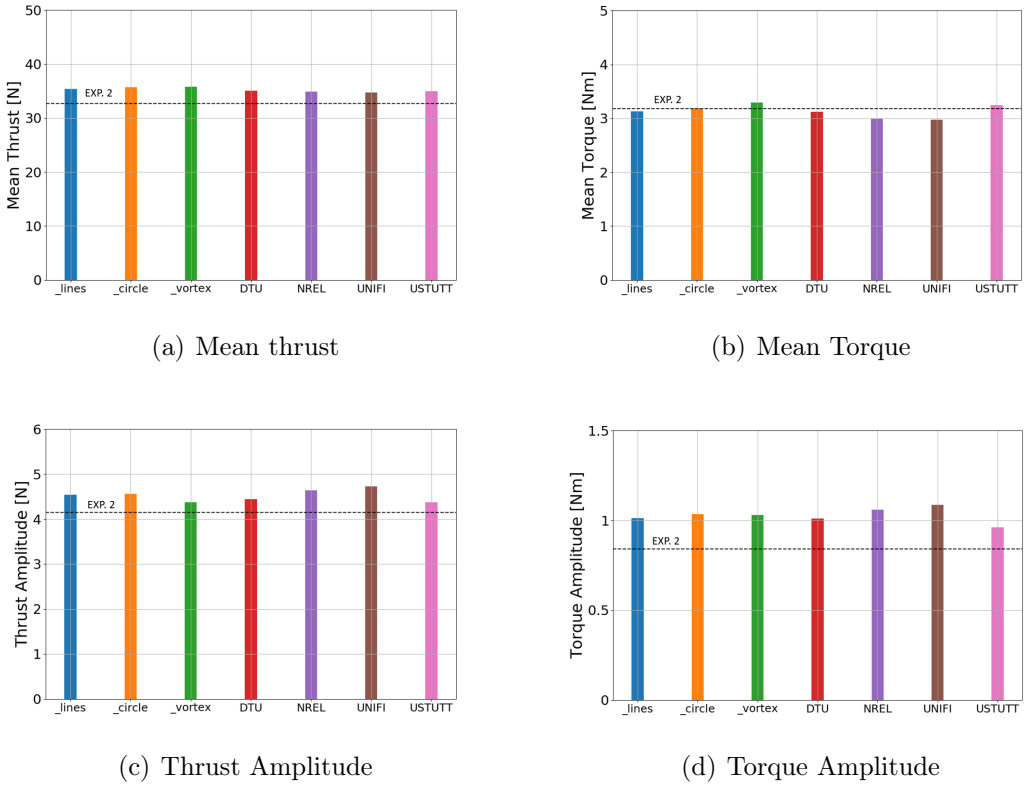


Figure 10.6: Filtered thrust and torque over one pitch period. Comparison with other participants to OC6 project [26] and [2].



(a) Mean thrust

(b) Mean Torque

(c) Thrust Amplitude

(d) Torque Amplitude

Figure 10.7: Mean and amplitude values. Comparison with other participants to OC6 project [26] and [2].

Fig. 10.6 and Fig. 10.7 show the comparison with these institutes. As reported, all the computational models tend to overestimate the mean thrust and the torque amplitude with respect to the Experiment 2. In general, good consistency with the data of the other participants is reached. The error between the average mean thrust of the new methods and the average mean thrust of the OC6 participants is 2.16%. Instead, the error between the average torque amplitudes is 0.83%.

Taking this observation into account, errors of the order of 9% on the mean thrust and of 6% on the torque amplitude, with respect to Experiment 2, are considered acceptable. As regards the mean value of the torque, the new simulations show an excellent consistency with the values of the fixed-bottom case and present errors of less than 4% if compared to the experimental data.

In conclusion, the new proposed methods present a good agreement with the experimental data and when results tend to deviate from these, however, excellent coherence with the OC6 outputs is demonstrated. This allows to confirm the validity of the methods even in the case of pitch motion.

As realized in the surge case, the span wise distribution of velocities and angles of attack is analyzed to better appreciate the effect of the inflow variation on the oscillations of integral quantities. The physical oscillations of the integral quantities shown in Fig. 10.5 demonstrate the capability of the three methods to perceive the apparent wind variation during the pitch motion.
The trends of some important quantities over a complete pitch period for three different blade sections is represented in Fig. 10.8. As in the previous chapter, only the LC3.5_vortex case is considered, being the one used for the surge case and being the most innovative one. In any case, considerations and results for the other two cases are practically the same.

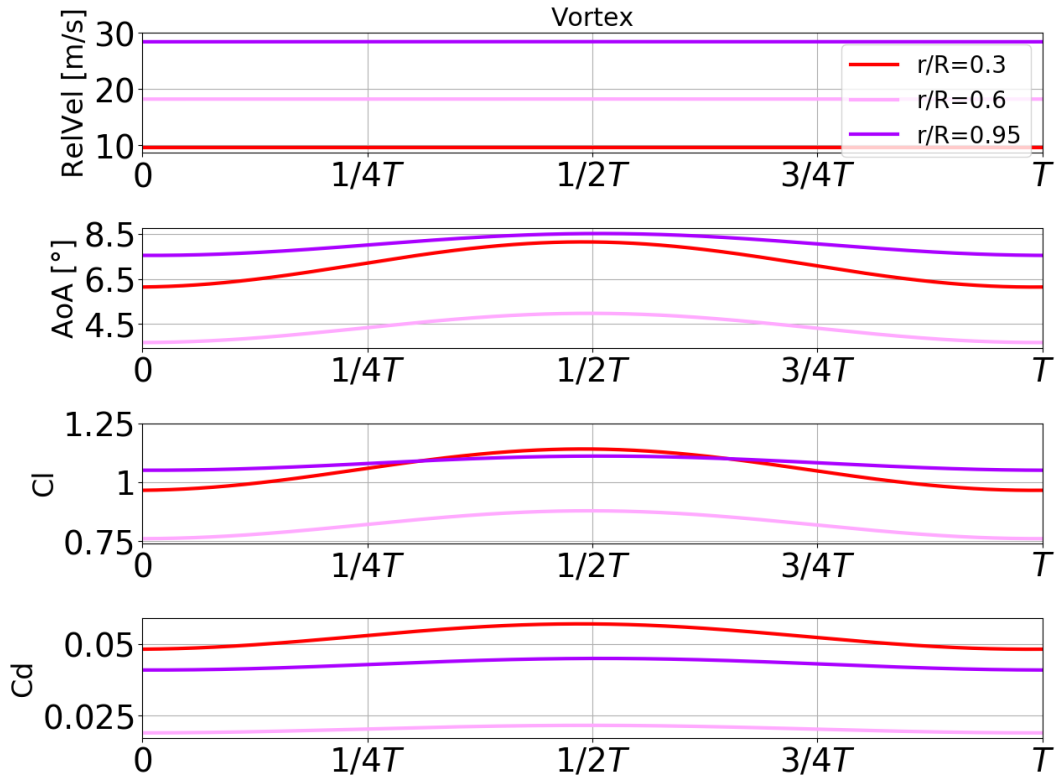
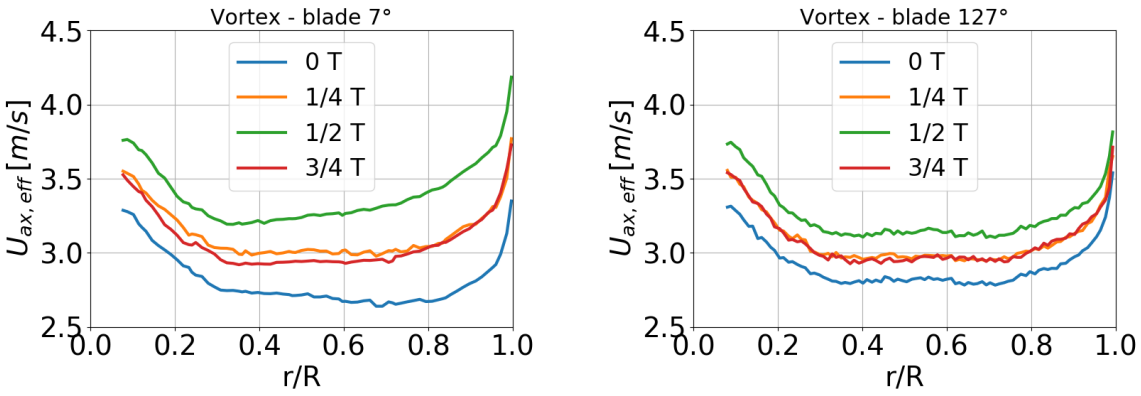


Figure 10.8: LC3.5_vortex: Aerodynamic quantities over a pitch period.

The maximum axial translation velocity is, in this case, 0.32 m/s; despite it is bigger than the surge one, it is still equivalent to 8% of the free stream speed so it is not sufficient to act significantly on the relative speed perceived by the rotating blades. The angles of attack suffer from the axial apparent wind variation. At the root, the smaller peripheral velocity makes the effect of the apparent wind more preponderant. At the tip the oscillations amplitude reduces because the peripheral velocity is much grater than the axial one. Since all the three sections are working in attached flow conditions, the same

AoA oscillations recur in the C_l and C_d trends with a progressive amplitude reduction towards the tip. These curves are coherent with what expected from a physical point of view so they work as a further confirmation of the great validity of the tested methods in pitch conditions.
As done in the surge case, a motion composition between the flow field and the pitch velocity has to be applied in order to compute the real apparent velocity acting on the blades. Indeed, due to how the code is implemented, the apparent speed is not applied to the cell center so in the flow field the effect of the apparent wind variation is not visible. Therefore, it becomes necessary to insert this velocity into the angle of attack calculation. In this case, $U_{ax,trasl}$ keeps into account the translation velocity given by the pitch for each single blade point. In Fig. 10.9 the real velocity perceived by one rotor blade ($U_{ax,eff}$) is shown for 4 time steps evenly spaced during the platform pitch period, which is equivalent to 1 s. A complete rotor revolution is performed by the single blade in 0.25 s. Therefore the four time steps, shown in figure, correspond to the same position of the blade. For instance, if the blade is vertical (θ=0°) at t= 0 T_{pitch} the blade is again in the vertical position also at t= $\frac{1}{4}T_{pitch}$, t= $\frac{1}{2}T_{pitch}$ and t= $\frac{3}{4}T_{pitch}$. In Fig. 10.9a) the axial velocity distribution is shown for the almost vertical blade (θ=7°) while Fig. 10.9b) is referred to the blade shifted by 120° (θ=127°).



(a) Blade at 7°

(b) Blade at 127°

Figure 10.9: LC3.5_vortex: Real axial velocity during the pitch period T.

This comparison is useful to highlight that, being the pitch motion a sort of rotation around the pitch center, the effect of the apparent wind variation depends on the blade position. The vertical blade has a greater effect towards the tip where the motion of the blade points has a bigger amplitude and so a higher apparent velocity. Indeed, the axial velocity variation between the four time steps is more marked in the tip region. The blade at θ=127° is overall closer to the pitch center therefore, the axial velocity variation is less

large. Furthermore, being positioned more horizontally with respect to the pitch center, it also sees less difference in amplitude between hub and tip. This effect is not directly visible in the attack angle amplitude and so, in all the other aerodynamic quantities, because for them also the contribution given by the peripheral velocity must be taken into account. For these quantities, as shown in Fig. 10.8 the amplitude always reduces towards the tip.

All these results are considered adequate to demonstrate the validity of the new sampling methods in the case of pitch motion. Pitch motion promotes more complicated flow phenomena in the flow field around the blade, therefore, in order to accurately estimate the axial velocities, which determines the angles of attack for each blade station, a robust velocity sample is even more crucial. The velocity sampling approach plays a crucial role for ensuring the reliability of actuator line codes since these medium fidelity solvers are highly dependent on flow data around the rotor. In this context, the proposed and tested methods can represent an important tool for improving the Actuator Line models.

11 | Conclusions and future developments

The main objective of this work was a formulation and validation of three different versions of a new Actuator Line Model for floating offshore wind turbines applications. In particular, three new sampling strategies of the velocity perceived by the rotor have been implemented in an already available in-house code. The modification made allows a more accurate estimation of the angle of attack and therefore of all the blade loads and makes the model more flexible and better applicable to unsteady motion conditions. The proposed methods have been assessed against two wind-tunnel experimental campaigns for a fixed-bottom, surging, and pitching turbine model.
Theoretically, velocities at the AL point are free of self-induction, nevertheless, a local sampling strategy becomes impracticable since the flow field in vicinity of the actuator line is characterized by severe gradients which make the correct evaluation of velocities unfeasible. Therefore, all the methods proposed in this work are based on velocity sampling at some distance from the AL point. At the basis of these methods there are different motivations, given the different nature of the strategies themselves:

- Sampling on lines: for this approach the objective is to demonstrate that a simple and easy to implement sampling technique can provide valid results.
- Sampling on circle: this strategy is based on the physical motivation according to which the induced velocities at opposite points of the circle simplify each other by eliminating the influence of bound circulation.
- Vortex sampling: in this case the sampled velocity, which is influenced by the bound vortex effect, is later corrected by subtracting the induced velocity. To compute the induced velocity, the bound circulation around the airfoil section is represented by a concentrated point vortex centered in the AL point.

Although there are slight changes in the order of measurement repeatability, the results of the three versions of the ALM code are found to be in excellent agreement with experimental rotor thrust and torque. When more consistent errors, with respect to the

experimental data, were identified, the results were compared with those of the other participants in the OC6 project. The other CFD methods results showed, both in the case of surge and pitch, a good consistency and coherence with what was here obtained. The work proves that the new models enable to estimate reliable velocity with physical span-wise trends coherent with the force distribution. The final simulation set-up avoids non-physical oscillations and captures the main rotor parameters also in surge and pitch platform movements. Although all the methods present valid results, the circle and the vortex ones are considered the most interesting as they are based on physical considerations. Moreover, between the two, the vortex strategy seems to be the most promising being more innovative than what is currently available in literature.

Possible future developments of the work primarily concern further improvements and validations of the proposed methods:

- a more specific sensitivity analysis on the geometrical parameters of the sampling methods is needed. In particular for the vortex method, a variation of the distance of the sampling line or of the circumference radius could modify the evaluation of the bound vortex effect and bring the U_{ax} span-wise trend closer to the one obtained with the other two approaches.
- simulations of more extreme surge and pitch cases must be tested to verify the good validity even in the case of more complex aerodynamic structures.
- implementation of a tip losses model to better simulate the tip vortex effect on the blade loads. A proper formulation for ALM codes, such as the one recently proposed by [23], is needed.

All these considerations aim to increase the accuracy of the AL model in the estimate of the loads but especially in the reproduction of the wake and tip effects. Indeed, the ALM represents a tool of great potential for studying the wakes interaction in wind farms. Therefore, it becomes necessary to verify the capacity of this model to accurately reproduce all the predominant aspects of the wake. In this perspective, other analyzes and implementations may be useful:

- wake analysis: to make the wake reproduction even more detailed, this analysis can be achieved by combining the ALM with large eddy simulation (LES) capable of capturing more specific features.
- tower and nacelle implementation: useful to assess the degree of accuracy when reproducing the near wake field. This should be carried out by tailoring the actuator line and actuator disk models to represent the tower and nacelle, respectively.

- development of a new code structure able to manage multi-turbine configurations through the AL representation. This will be useful to begin the study of wind turbine arrays starting from two turbines in tandem and then moving to more complex configurations.

Once these improvements have been implemented and the ability of the ALM to accurately reproduce the wake and the interaction between different turbines has been verified, the method can become suitable to help the development of new control strategies aimed at maximizing the wind farm power. It will be a fundamental tool for developing new algorithms or new artificial intelligence techniques for control systems.

Bibliography

[1] Wind europe, floating offshore wind energy, a policy blueprint for europe.

[2] doi: 10.21947/1922900. Atmosphere to Electrons (A2e). oc6/oc6.phase3. Maintained by A2e Data Archive and Portal for U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy. Accessed: 16 03 2023.

[3] Wind europe, wind energy in europe. 2022.

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

[5] S. Baschetty and S. Ozcelik. Review on dynamics of offshore floating wind turbine platforms. *Energies*, 2021.

[6] R. Bergua, A. Robertson, J. Jonkman, E. Branlard, A. Fontanella, M. Belloli, P. Schito, A. Zasso, G. Persico, A. Sanvito, et al. Oc6 project phase iii: validation of the aerodynamic loading on a wind turbine rotor undergoing large motion caused by a floating support structure. *Wind Energy Science Discussions*, pages 1–33, 2022.

[7] M. L. Blaylock, P. Sakievich, B. C. Houchens, A. Hsieh, D. C. Maniaci, L. A. Martinez-Tossas, and M. Churchfield. Comparison of actuator line model and a filtered lifting line correction implemented in nalu-wind large eddy simulations of the atmospheric boundary layer. Technical report, Sandia National Lab.(SNL-CA), Livermore, CA (United States); Sandia National ..., 2021.

[8] M. L. Blaylock, L. Martinez-Tossas, P. Sakievich, B. C. Houchens, L. Cheung, K. Brown, A. Hsieh, D. C. Maniaci, and M. J. Churchfield. Validation of actuator line and actuator disk models with filtered lifting line corrections implemented in nalu-wind large eddy simulations of the atmospheric boundary layer. In *AIAA SCITECH 2022 Forum*, page 1921, 2022.

[9] F. Castellani and A. Vignaroli. An application of the actuator disc model for wind turbine wakes calculations. *Applied Energy*, 101:432–440, 2013.

[10] M. J. Churchfield, S. J. Schreck, L. A. Martinez, C. Meneveau, and P. R. Spalart. An advanced actuator line method for wind energy applications and beyond. In *35th Wind Energy Symposium*, page 1998, 2017.

[11] L. Cottura, R. Caradonna; A. Ghigo; R. Novo; G. Bracco; G. Mattiazzo. Dynamic modeling of an offshore floating wind turbine for application in the mediterranean. *Energies*, 2021.

[12] J. De Vaal, M. Hansen, and T. Moan. Effect of wind turbine surge motion on rotor thrust and induced velocity. *Wind Energy*, 17(1):105–121, 2014.

[13] I. Herráez, E. Daniele, and J. G. Schepers. Extraction of the wake induction and angle of attack on rotating wind turbine blades from piv and cfd results. *Wind Energy Science*, 3(1):1–9, 2018.

[14] InfoLink. Floating wind turbine ii: Floating structure technology and market trends, 2021. URL `https://www.infolink-group.com/energy-article/Floating-wind-turbine-II-Floating-structure-technology-and-market-trends`.

[15] IRENA. Future of wind. 2019.

[16] P. K. Jha, M. J. Churchfield, P. J. Moriarty, and S. Schmitz. Guidelines for volume force distributions within actuator line modeling of wind turbines on large-eddy simulation-type grids. *Journal of Solar Energy Engineering*, 136(3), 2014.

[17] J. Johansen and N. N. Sørensen. Aerofoil characteristics from 3d cfd rotor computations. *Wind Energy*, 7(4):283–294, 2004.

[18] E. Jost, L. Klein, H. Leipprand, T. Lutz, and E. Krämer. Extracting the angle of attack on rotor blades from cfd simulations. *Wind Energy*, 21(10):807–822, 2018.

[19] S. Mancini, K. Boorsma, M. Caboni, M. Cormier, T. Lutz, P. Schito, and A. Zasso. Characterization of the unsteady aerodynamic response of a floating offshore wind turbine to surge motion. *Wind Energy Science*, 5(4):1713–1730, 2020.

[20] L. A. Martínez-Tossas, M. J. Churchfield, and S. Leonardi. Large eddy simulations of the flow past wind turbines: actuator line and disk modeling. *Wind Energy*, 18 (6):1047–1060, 2015.

[21] L. A. Martínez-Tossas, M. J. Churchfield, and C. Meneveau. Optimal smoothing

length scale for actuator line models of wind turbine blades based on gaussian body force distribution. *Wind Energy*, 20(6):1083–1096, 2017.

[22] P. F. Melani. *Power Augmentation of Darrieus-type Turbines by Means of Novel Solutions and Multi-Fidelity Simulations*. PhD thesis, Università degli Studi di Firenze, 2022.

[23] A. R. Meyer Forsting, G. R. Pirrung, and N. Ramos-García. A vortex-based tip/smearing correction for the actuator line. *Wind Energy Science*, 4(2):369–383, 2019.

[24] H. Rahimi, M. Hartvelt, J. Peinke, and J. Schepers. Investigation of the current yaw engineering models for simulation of wind turbines in bem and comparison with cfd and experiment. In *Journal of Physics: Conference Series*, volume 753, page 022016. IOP Publishing, 2016.

[25] H. Rahimi et al. Evaluation of different methods for determining the angle of attack on wind turbine blades with cfd results under axial inflow conditions. *Renewable Energy*, 125:866–876, 2018.

[26] A. Robertson. Oc6 phase iii: Validation of wind turbine aerodynamic loading during surge/pitch motion. doi: 10.21947/1922900. URL `https://www.osti.gov/biblio/1922900`.

[27] A. Robertson, R. Bergua, A. Fontanella, and J. Jonkman. Oc6 phase iii definition document. 2022.

[28] P. Schito and A. Zasso. Actuator forces in cfd: Rans and les modeling in openfoam. In *Journal of Physics: Conference Series*, volume 524, page 012160. IOP Publishing, 2014.

[29] S. Shamsoddin and F. Porté-Agel. A large-eddy simulation study of vertical axis wind turbine wakes in the atmospheric boundary layer. *Energies*, 9(5):366, 2016.

[30] W. Z. Shen, J. N. Sørensen, and J. Zhang. Actuator surface model for wind turbine flow computations. In *Proceedings of European Wind Energy Conference and Exhibition*, volume 7, 2007.

[31] W. Z. Shen, M. O. Hansen, and J. N. Sørensen. Determination of the angle of attack on rotor blades. *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, 12(1):91–98, 2009.

[32] W. Z. Shen, M.O.L. Hansen; J.N. Sorensen. Determination of angle of attack (aoa)

for rotating blades. *In Proceedings of the Euro Mech Colloquium*, pages p. 37–45, 2006.

[33] M. Shives and C. Crawford. Mesh and load distribution requirements for actuator line cfd simulations. *Wind Energy*, 16(8):1183–1196, 2013.

[34] H. Snel. Review of aerodynamics for wind turbines. *Wind Energy*, 6:203–211, 2003.

[35] J. N. Sorensen and W. Z. Shen. Numerical modeling of wind turbine wakes. *Journal of Fluids Engineering*, 124:393–399, 2002.

[36] N. Troldborg. *Actuator line modeling of wind turbine wakes.* PhD thesis, Technical University of Denmark, 2009.

[37] Y.-T. Wu and F. Porté-Agel. Large-eddy simulation of wind-turbine wakes: evaluation of turbine parametrisations. *Boundary-layer meteorology*, 138(3):345–366, 2011.

A | Velocity sampling methods code implementation

The lines of code written to implement the different sampling strategies are here reported. Depending on the method implemented, the sampling lines or circles are first constructed in local coordinates (tg,ax,rad) on a reference plane where the AL point is imagined to be located at the origin and the radial coordinate is equal to zero. Subsequently, thanks to two tensors already implemented to define the position of the AL points, the rotation and translation of the sampling line is carried out in order to position it on the plane perpendicular to the blade and passing through the AL point of interest. The two tensors, passed as input to the sampling function, consider the displacement due to both the surge and the pitch motion. At this point the velocity on the points of the sampling line is obtained by applying an interpolation function from the cell-centered velocity value of the cell containing the point. For the line and circle sampling method, the procedure is completed by averaging the different velocity components. For the vortex method first the circulation is computed, then the induced velocity is calculated and combined with the sampled one to finally obtain the effective one.

```
IMPLEMENTATION OF LINES METHOD
vector actuatorForce::calcRefVelocityEVM_modified(point &iEffForcePoint,
                                                  vector &iEVMplaneNormal,
                                                  tensor &AFtensCart2Pitch,
                                                  tensor &AFtensCart2Bld,
                                                  vector &iEVMupStreamDir,
                                                  scalar &iUnitLength,
                                                  bool& oLocalLine,
                                                  bool parallelCalc)
{

    //The axial component of V is sampled on two lines
    //perpendicular to the relative velocity
    vector asser(0,0,1);
    //trasformation of iEVMupStreamDir from the absolute to the local frame of reference
    vector iEVMupStreamDirLoc=AFtensCart2Bld&iEVMupStreamDir
    iEVMupStreamDirLoc[2]=0;
    vector EVMsampDir=asser^iEVMupStreamDirLoc; //perpendicular to iEVMupStreamDirLoc
    EVMsampDir/=mag(EVMsampDir);

    //mid point of the Vx sampling line upstream
    point M0_up=-dist*iUnitLength*iEVMupStreamDirLoc;
    //mid point of the Vx sampling line downstream
    point M0_down=+dist*iUnitLength*iEVMupStreamDirLoc;
    //extreme points of Vx sampling lines upstream/downstream
    point A0=M0_up-(length/2)*iUnitLength*EVMsampDir;
    point B0=M0_up+(length/2)*iUnitLength*EVMsampDir;
    point C0=M0_down-(length/2)*iUnitLength*EVMsampDir;
    point D0=M0_down+(length/2)*iUnitLength*EVMsampDir;

    //mid point of the Vy sampling line upstream
    point m0_up=-dist*iUnitLength*EVMsampDir;
    //mid point of the Vy sampling line downstream
    point m0_down=dist*iUnitLength*EVMsampDir;
    //extreme points of Vy sampling lines upstream/downstream
    point a0=m0_up-(length/2)*iUnitLength*iEVMupStreamDirLoc;
    point b0=m0_up+(length/2)*iUnitLength*iEVMupStreamDirLoc;
    point c0=m0_down-(length/2)*iUnitLength*iEVMupStreamDirLoc;
    point d0=m0_down+(length/2)*iUnitLength*iEVMupStreamDirLoc;

    point Extremes0[8]={A0,B0,C0,D0,a0,b0,c0,d0};
    point Extremes_ruotati[8];

    //rotation and traslation of points
    for (int i=0; i<8; i++)
    {
        Extremes_ruotati[i]=(inv(AFtensCart2Pitch)&inv(AFtensCart2Bld)&
        Extremes0[i])+iEffForcePoint;
    }
```

```
//First line for Ux sampling
point pStart1_Ux=Extremes_ruotati[0];
point pEnd1_Ux=Extremes_ruotati[1];
vector V(0.0,0.0,0.0);
vector V_Ux_media(0.0,0.0,0.0);
vector V1_Ux(0.0,0.0,0.0);
scalar totLength1_Ux(0);
bool oLocalLine1_Ux=false;

// Search intersections between the sampling line and the mesh.
if(mesh_.C().size()>0)//if processor has no cell near turbine site
//do not search for intersections
{
    intrsctData.clearList();
    intrsctData.findCellIntersctPts(pStart1_Ux,pEnd1_Ux);
    label oldMeshLbl;
    forAll(intrsctData.midPts(),i)
    {
        oldMeshLbl=MeshTurbine_.cellMap()[intrsctData.intersctCell()[i]];
        V1_Ux+=velInterp(intrsctData.midPts()[i],oldMeshLbl)*intrsctData.lenLine()[i];
        totLength1_Ux +=intrsctData.lenLine()[i];
    }
}
// Calculation of the first axial velocity provided by the upstream line,
//if some intersections have been found.
// Local velocity and corresponding portion of the EVM sampling line are collected
//for each midPoint (point between two intersections).
//At the end, the aritmetic mean velocity is returned.
if(parallelCalc)
{
    // collect and sum V and totLength of all processes
    reduce(totLength1_Ux, sumOp<scalar>());
    reduce(V1_Ux, sumOp<vector>());
    V1_Ux /= totLength1_Ux;
}
else
{
    if(totLength1_Ux<0.95*length*iUnitLength)// if the sampled line is shorter
    //than 0.95*length then probably the line belongs to two processors
    {
        oLocalLine1_Ux=false;
    }
    else
    {
        oLocalLine1_Ux=true;
        V1_Ux /= totLength1_Ux;
    }

}
```

```
//Second line for Ux sampling
point pStart2_Ux= Extremes_ruotati[2];
point pEnd2_Ux=Extremes_ruotati[3];
vector V2_Ux(0.0,0.0,0.0);
scalar totLength2_Ux(0);
bool oLocalLine2_Ux=false;

// Search intersections between the sampling line and the mesh.
if(mesh_.C().size()>0)
{
    intrsctData.clearList();
    intrsctData.findCellIntersctPts(pStart2_Ux,pEnd2_Ux);
    Label oldMeshLbl;
    forAll(intrsctData.midPts(),i)
    {
        oldMeshLbl=MeshTurbine_.cellMap()[intrsctData.intersctCell()[i]];
        V2_Ux+=velInterp(intrsctData.midPts()[i],oldMeshLbl)*intrsctData.lenLine()[i];
        totLength2_Ux +=intrsctData.lenLine()[i];
    }
}

if(parallelCalc)
{
    // collect and sum V and totLength of all processes
    reduce(totLength2_Ux, sumOp<scalar>());
    reduce(V2_Ux, sumOp<vector>());
    V2_Ux /= totLength2_Ux;
}
else
{
    if(totLength2_Ux<0.95*length*iUnitLength)
    {
        oLocalLine2_Ux=false;

    }
    else
    {
        oLocalLine2_Ux=true;
        V2_Ux /= totLength2_Ux;
    }

}

//The tangential component of V is sampled on two lines parallel to the relative
//velocity but on the plane of the airfoil (perpendicular to iEVMplaneNormal).
 //First line for Uy sampling
point pStart1_Uy=Extremes_ruotati[4];
point pEnd1_Uy=Extremes_ruotati[5];
```

```
vector V_Uy_media(0.0,0.0,0.0);
vector V1_Uy (0,0,0);
scalar totLength1_Uy(0);
bool oLocalLine1_Uy=false;

// Search intersections between the sampling line and the mesh.
if(mesh_.C().size()>0)
{
    intrsctData.clearList();
    intrsctData.findCellIntersctPts(pStart1_Uy,pEnd1_Uy);
    label oldMeshLbl;
    forAll(intrsctData.midPts(),i)
    {
        oldMeshLbl=MeshTurbine_.cellMap()[intrsctData.intersctCell()[i]];

        V1_Uy+=velInterp(intrsctData.midPts()[i],oldMeshLbl)*
        intrsctData.lenLine()[i];

        totLength1_Uy +=intrsctData.lenLine()[i];
    }
}

if(parallelCalc)
{
    // collect and sum V and totLength of all processes
    reduce(totLength1_Uy, sumOp<scalar>());
    reduce(V1_Uy, sumOp<vector>());
    V1_Uy /= totLength1_Uy;
}
else
{
    if(totLength1_Uy<0.95*length*iUnitLength)// if the sampled line is shorter
    //than 0.95*length then probably the line belongs to two processors
    {
        oLocalLine1_Uy=false;
    }
    else
    {
        oLocalLine1_Uy=true;
        V1_Uy /= totLength1_Uy;
    }
}

//Second line for Uy sampling
point pStart2_Uy=Extremes_ruotati[6];
point pEnd2_Uy=Extremes_ruotati[7];
vector V2_Uy (0,0,0);
scalar totLength2_Uy(0);
bool oLocalLine2_Uy=false;
```

```
// Search intersections between the sampling line and the mesh.
if(mesh_.C().size()>0)
{
    intrsctData.clearList();
    intrsctData.findCellIntersctPts(pStart2_Uy,pEnd2_Uy);
    label oldMeshLbl;
    forAll(intrsctData.midPts(),i)
    {
        oldMeshLbl=MeshTurbine_.cellMap()[intrsctData.intersctCell()[i]];
        V2_Uy+=velInterp(intrsctData.midPts()[i],oldMeshLbl)*intrsctData.lenLine()[i];
        totLength2_Uy +=intrsctData.lenLine()[i];
    }
}
if(parallelCalc)
{
    // collect and sum V and totLength of all processes
    reduce(totLength2_Uy, sumOp<scalar>());
    reduce(V2_Uy, sumOp<vector>());
    V2_Uy /= totLength2_Uy;
}
else
{
    if(totLength2_Uy<0.95*length*iUnitLength)// if the sampled line is shorter
    //than 0.95*length then probably the line belongs to two processors
    {
        oLocalLine2_Uy=false;
    }
    else
    {
        oLocalLine2_Uy=true;
        V2_Uy /= totLength2_Uy;
    }

}

if (oLocalLine1_Ux==false || oLocalLine2_Ux==false || oLocalLine1_Uy==false
|| oLocalLine2_Uy==false)
{
    oLocalLine=false;
}
else
{
    oLocalLine=true;
}

//Change of the reference frame from cartesian (x,y,z) to blade (t,a,r)
vector V1_Ux_Bld (0,0,0);
vector V2_Ux_Bld (0,0,0);
vector V1_Uy_Bld (0,0,0);
vector V2_Uy_Bld (0,0,0);
```

```
    V1_Ux_Bld=AFtensCart2Bld&V1_Ux;
    V2_Ux_Bld=AFtensCart2Bld&V2_Ux;
    V1_Uy_Bld=AFtensCart2Bld&V1_Uy;
    V2_Uy_Bld=AFtensCart2Bld&V2_Uy;

    //Average between V1_Ux_Bld and V2_Ux_Bld
    vector V_Ux_media_Bld (0,0,0);
    V_Ux_media_Bld=(V1_Ux_Bld+V2_Ux_Bld)/2;

    //Average between V1_Uy_Bld and V2_Uy_Bld
    vector V_Uy_media_Bld (0,0,0);
    V_Uy_media_Bld=(V1_Uy_Bld+V2_Uy_Bld)/2;

    vector V_Bld (0,0,0);
    V_Bld[0]=V_Uy_media_Bld[0];
    V_Bld[1]=V_Ux_media_Bld[1];
    V_Bld[2]=V_Ux_media_Bld[2];

    //Change of the reference frame from blade to cartesian
    V=inv(AFtensCart2Bld)&V_Bld;
    return(V);

}
```

```
IMPLEMENTATION OF CIRCLE METHOD
vector actuatorForce::calcRefVelocityEVM_modified(point &iEffForcePoint,
                                                  vector &iEVMplaneNormal,
                                                  tensor &AFtensCart2Pitch,
                                                  tensor &AFtensCart2Bld,
                                                  vector &iEVMupStreamDir,
                                                  scalar &iUnitLength,
                                                  bool& oLocalLine,
                                                  bool parallelCalc)
{
    // circumference on the local plane with r=0
         int N=36;
         scalar r=radius*iUnitLength;

    //vector of circumference angles phi
    float phi[N];
    float delta_phi=(2*Foam::constant::mathematical::pi)/N;
    phi[0]=0;
    for (int i=1 ; i<N; ++i)
    {
        phi[i]=phi[i-1]+delta_phi;
    }

    point puntiCrf[N];
    for (int i=0; i<N; i++)
    {
        puntiCrf[i].x()=r*sin(phi[i]);
        puntiCrf[i].y()=r*cos(phi[i]);
        puntiCrf[i].z()=0;
    }

    // circumference on the plane perpendicular to the blade
    //points obtained in cartesian coordinates (x,y,z)
    point puntiCrfRuotati[N];
    for (int i=0; i<N; i++)
    {
        puntiCrfRuotati[i]=(inv(AFtensCart2Pitch)&inv(AFtensCart2Bld)
        &puntiCrf[i])+iEffForcePoint;
    }

    vector V(0,0,0);
    oLocalLine=false;
    int count=0;
    if(mesh_.C().size()>0)//if processor has no cell near turbine site
    //do not search for intersections
    {
        for (int i=0; i<N; i++)
        {
            label Label=mesh_.findCell(puntiCrfRuotati[i]);
```

```
            if (Label>=1)
            {
                label oldMeshLbl=MeshTurbine_.cellMap()[Label];
                V+=velInterp(puntiCrfRuotati[i],oldMeshLbl);
                count+=1; //it counts how many circle points
                //are contained in the single processor
            }
        }
    }

    if(parallelCalc)
    {
        // collect and sum V of all processes
        reduce(V, sumOp<vector>());
        V /= N;
    }
    else
    {
        //if(totLength_up<0.95*length*iUnitLength)// if the sampled line is shorter
        //than 0.95*length then probably the line belongs to two processors
        if(count<N) //if the sampled points are lower than N
        //then probably the line belongs to two processors
        {
            oLocalLine=false;
        }
        else
        {
            oLocalLine=true;
                V /= N;
        }
    }
    return (V);
}
```

```
IMPLEMENTATION OF VORTEX METHOD
vector actuatorForce::calcRefVelocityEVM_modified(point &iEffForcePoint,
                                                  vector &iEVMplaneNormal,
                                                  tensor &AFtensCart2Pitch,
                                                  tensor &AFtensCart2Bld,
                                                  vector &iEVMupStreamDir,
                                                  scalar &iUnitLength,
                                                  bool& oLocalLine,
                                                  bool parallelCalc)
{
    //Monitoring line
    //upstream line on the local plane with r=0
    //Vsamp is sampled on an upstream line perpendicular to the relative velocity
    vector asser(0,0,1);
    //trasformation of iEVMupStreamDir from the absolute to the local frame of reference
        vector
    iEVMupStreamDirLoc=AFtensCart2Bld&iEVMupStreamDir;
    iEVMupStreamDirLoc[2]=0;
    vector EVMsampDir=asser^iEVMupStreamDirLoc; //direction perpendicular to
    //iEVMupStreamDirLoc
    EVMsampDir/=mag(EVMsampDir);

    //mid point of the sampling line upstream
    point M0_up=-dist*iUnitLength*iEVMupStreamDirLoc;
    //extreme points of Vx sampling lines upstream/downstream
    point A0=M0_up-(length/2)*iUnitLength*EVMsampDir;
    point B0=M0_up+(length/2)*iUnitLength*EVMsampDir;

    point LinePoints_Bld[3]={A0,M0_up,B0};
    point LinePoints_cart[3];

    //tranformation from local to cartesian coordinates
    for (int i=0; i<3; i++)
    {
        LinePoints_cart[i]=(inv(AFtensCart2Pitch)&inv(AFtensCart2Bld)&
        LinePoints_Bld[i])+iEffForcePoint;

    }
    point pStart=LinePoints_cart[0];
    point pEnd=LinePoints_cart[2];

    vector Veff(0,0,0);
    vector Vsamp(0,0,0);
    vector Vsamp_Bld(0,0,0);
    oLocalLine=false;
    scalar totLength(0);
    bool oLocalLine_samp=false;
    int count_samp(0);
```

```
// Search intersections between the sampling line and the mesh.
if(mesh_.C().size()>0)//if processor has no cell near turbine site
//do not search for intersections
{
    intrsctData.clearList();
    intrsctData.findCellIntersctPts(pStart,pEnd);
    label oldMeshLbl;
    forAll(intrsctData.midPts(),i)
    {
        oldMeshLbl=MeshTurbine_.cellMap()[intrsctData.intersctCell()[i]];
        Vsamp+=velInterp(intrsctData.midPts()[i],oldMeshLbl);
        totLength +=intrsctData.lenLine()[i];
        count_samp+=1;
    }
}

if(parallelCalc)
{
    // collect and sum V and totLength of all processes
    reduce(totLength, sumOp<scalar>());
    reduce(Vsamp, sumOp<vector>());
    reduce(count_samp, sumOp<scalar>());
    Vsamp /= count_samp;
}
else
{
    if(totLength<0.95*length*iUnitLength)//if the sampled line is shorter
    //than 0.95*length then probably the line belongs to two processors
    {
        oLocalLine_samp=false;

    }
    else
    {
        oLocalLine_samp=true;
            Vsamp /= count_samp;
    }
}

//computation of the induced velocity using a circumference around the ALpoint
// circumference on the local plane with r=0
scalar r_gamma=radius_gamma*iUnitLength;

//vector of circumference angles phi
float phi_gamma[N_gamma];
float delta_phi_gamma=(2*Foam::constant::mathematical::pi)/N_gamma;
phi_gamma[0]=0;
for (int i=1 ; i<N_gamma; ++i)
{
    phi_gamma[i]=phi_gamma[i-1]+delta_phi_gamma;
}
```

```
point puntiGammaCrf_Bld[N_gamma];
for (int i=0; i<N_gamma; i++)
{
    puntiGammaCrf_Bld[i].x()=r_gamma*cos(phi_gamma[i]);
    puntiGammaCrf_Bld[i].y()=r_gamma*sin(phi_gamma[i]);
    puntiGammaCrf_Bld[i].z()=0;
}

//circumference on the plane perpendicular to the blade
//points obtained in cartesian coordinates (x,y,z)
point puntiGammaCrf_cart[N_gamma];
for (int i=0; i<N_gamma; i++)
{
    puntiGammaCrf_cart[i]=(inv(AFtensCart2Pitch)&inv(AFtensCart2Bld)&
    puntiGammaCrf_Bld[i])+iEffForcePoint;
}

//increments between circle points along axial direction
double delta_ax[N_gamma];
delta_ax[N_gamma-1]=puntiGammaCrf_Bld[0].y()-puntiGammaCrf_Bld[N_gamma-1].y();
for (int i=0; i<N_gamma-1; i++)
{
    delta_ax[i]=puntiGammaCrf_Bld[i+1].y()-puntiGammaCrf_Bld[i].y();
}

//increments between circle points along tangential direction
double delta_tg[N_gamma];
delta_tg[N_gamma-1]=puntiGammaCrf_Bld[0].x()-puntiGammaCrf_Bld[N_gamma-1].x();
for (int i=0; i<N_gamma-1; i++)
{
     delta_tg[i]=puntiGammaCrf_Bld[i+1].x()-puntiGammaCrf_Bld[i].x();
}

vector Vind(0,0,0);
vector Vind_Bld(0,0,0);
vector Vinterp(0,0,0);
vector Vinterp_Bld(0,0,0);
double circ(0);
bool oLocalLine_circ=false;
int count_gamma=0;
if(mesh_.C().size()>0)
{
    for (int i=0; i<N_gamma; i++)
    {
        label Label=mesh_.findCell(puntiGammaCrf_cart[i]);
        if (Label > -1)
        {
            label oldMeshLbl_circ=MeshTurbine_.cellMap()[Label];
            Vinterp=velInterp(puntiGammaCrf_cart[i],oldMeshLbl_circ);
```

```
            Vinterp_Bld=AFtensCart2Pitch&AFtensCart2Bld&Vinterp;
            circ+=Vinterp_Bld[1]*delta_ax[i]+Vinterp_Bld[0]*delta_tg[i];
            count_gamma+=1; //it counts how many circle points are
            //inside the single processor
        }
    }
}
if(parallelCalc)
{
    // collect and sum V of all processes
    reduce(circ, sumOp<double>());
    reduce(count_gamma,sumOp<scalar>());
}
else
{
    if(count_gamma<N_gamma) //if the sampled points are lower than N
    //then probably the line belongs to two processors
    {
        oLocalLine_circ=false;
    }
    else
    {
        oLocalLine_circ=true;
    }
}

if (oLocalLine_samp==false || oLocalLine_circ==false)
{
    oLocalLine=false;
}
else
{
    oLocalLine=true;
}

//computation of induced velocity in local coordinates (2D method)
double u_ind[intrsctData.midPts().size()]; //vector containing axial
//components of induced velocity
double v_ind[intrsctData.midPts().size()]; //vector containing tangential
//components of induced velocity

//passage of ALP coordinates from blade to cartesian
point iEffForcePoint_Bld=AFtensCart2Pitch&AFtensCart2Bld&
(iEffForcePoint-iEffForcePoint);

//passage of sampling line points coordinates from blade to cartesian
point midPts_Bld[intrsctData.midPts().size()];
```

```
    forAll(intrsctData.midPts(),i)
    {
        midPts_Bld[i]=AFtensCart2Pitch&AFtensCart2Bld&(intrsctData.midPts()[i]
        -iEffForcePoint);
    }

    //computation of induction velocity
    for (int i=0; i<intrsctData.midPts().size(); i++)
    {
        double r=sqrt(pow(midPts_Bld[i].y()-iEffForcePoint_Bld[1],2)+
        pow(midPts_Bld[i].x()-iEffForcePoint_Bld[0],2));

        u_ind[i]=(circ*(midPts_Bld[i].x()-iEffForcePoint_Bld[0]))/
        (2*Foam::constant::mathematical::pi*pow(r,2));

        v_ind[i]=-(circ*(midPts_Bld[i].y()-iEffForcePoint_Bld[1]))/
        (2*Foam::constant::mathematical::pi*pow(r,2));
    }
    //computation of the mean induction velocity
    for (int i=0; i<intrsctData.midPts().size(); i++)
    {
        Vind_Bld[0]+=v_ind[i];
        Vind_Bld[1]+=u_ind[i];
    }

    if (count_samp==0)
    {
        int count_samp_temp=1;
        Vind_Bld[0]/=count_samp_temp;
        Vind_Bld[1]/=count_samp_temp;
    }
    else
    {
        Vind_Bld[0]/=count_samp;
        Vind_Bld[1]/=count_samp;
    }

    //computation of the effective velocity (removing the bound vortex influence)
    vector Veff_Bld(0,0,0);
    Vsamp_Bld=AFtensCart2Pitch&AFtensCart2Bld&Vsamp;

    Veff_Bld[0]=Vsamp_Bld[0]-Vind_Bld[0];
    Veff_Bld[1]=Vsamp_Bld[1]-Vind_Bld[1];
    Veff_Bld[2]=Vsamp_Bld[2];//neglecting the bound vortex influence on the radial component

    //coming back to cartesian coordinates
    Veff=inv(AFtensCart2Pitch)&inv(AFtensCart2Bld)&Veff_Bld;
    return (Veff);

}
```

List of Figures

List of Tables

List of Symbols

Symbol	Description
CFD	Computational fluid dynamics
ALM	Actuator Line Model
CO_2	Carbon dioxide
CfD	Contracts for Difference
CF	Capacity factor
LCOE	Levelized Cost of Electricity
TLP	Tension Leg platform
DOF	Degree of freedom
BEM	Blade element momentum
OC6	Offshore Code Comparison Collaboration Continuation, with Correlation and un-Certainty
IEA WIND	International Energy Agency Wind Technology Collaboration Program
NREL	National Renewable Energy Laboratory
LCs	Load cases
HAWT	Horizontal axes wind turbine
c	Chord length
Ω	Rotational speed
r	Radius
D	Rotor diameter
U	Absolute velocity
RANS	Reynolds Average Navier Stokes
ASM	Actuator Surface Model
2D	Two dimensions
3D	Three dimensions
a	Axial induction factor
C_T	Thrust coefficient

Symbol	Description
T	Thrust Force
A	Rotor swept area
ρ	Air density
α	Attack angle
ϕ	Relative flux angle
γ	Local pitch angle
Re	Reynolds Number
μ	Freestream dynamic viscosity
C_l	Lift coefficient
C_d	Drag coefficient
t	Time
p	Pressure
ν	Kinematic viscosity
AoA	Attack angle
ϵ	Gaussian shape parameter
η	Regularization Kernel factor
Δ	Characteristic cell size
c^*	Equivalent elliptic planform
AAT	Average Azimuthal method
EVM	Effective velocity method
AF	Actuator forse
FOWT	Floating offshore wind turbine
Λ	Tip speed ratio
θ_p	Turbine pitch angle
C_P	Power coefficient
Co	Courant number
R	Rotor radius

Subscript	Description
rel	relative
tg	tangential
ax	axial
∞	freestream
ker	kernel
ind	inducted
eff	effective
samp	sampled
trasl	transnational
rad	radial
circ	circulation
s	surge
p	pitch

Acknowledgements

Ringrazio innanzitutto il prof. Persico e l'Ing. Sanvito per avermi dato la possibilità di realizzare questo progetto di tesi. Vi ringrazio per la disponibilità e per l'aiuto che mi avete fornito durante questo lavoro. Attraverso il continuo e proficuo confronto, mi avete permesso di imparare molto e di entrare a diretto contatto con il mondo della ricerca.

Un grazie fondamentale va alla mia famiglia, mamma, papà e Matilde, senza i quali non sarei arrivata a questo importante obiettivo. Grazie per avermi sempre fornito un enorme sostegno, per aver sempre creduto nelle mie ambizioni e per avermi insegnato che, attraverso il lavoro e la perseveranza, si può raggiungere ogni traguardo.

Grazie a tutti i miei amici di sempre Debora, Francesca, Elisabetta, Eleonora, Luca T., Federico, Matteo e Luca V. per avermi accompagnato in ogni momento di divertimento e spensieratezza e per aver sempre creduto in ogni mia aspirazione.

Grazie a Eleonora, Marta e Rachele con cui ho condiviso gli ultimi anni di questo percorso. È anche grazie ai progetti che abbiamo portato a termine insieme, dopo tanto duro lavoro, che sono riuscita ad arrivare a questo punto.

E infine, grazie a Matteo che, dal primo esame di questo percorso universitario, mi ha incoraggiato in ogni momento di sconforto e ha gioito per ogni mio piccolo traguardo. Grazie per aver sempre creduto in me.