



HAL
open science

Solving Incompressible Navier-Stokes Equations on Octree grids: towards Application to Wind Turbine Blade Modelling

Claire Taymans

► **To cite this version:**

Claire Taymans. Solving Incompressible Navier-Stokes Equations on Octree grids: towards Application to Wind Turbine Blade Modelling. Numerical Analysis [math.NA]. Université de Bordeaux, 2018. English. NNT : 2018BORD0157 . tel-01934807v2

HAL Id: tel-01934807

<https://theses.hal.science/tel-01934807v2>

Submitted on 12 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

pour l'obtention du grade de

DOCTEUR DE L'UNIVERSITÉ DE BORDEAUX

École doctorale de Mathématiques et Informatique de Bordeaux

Spécialité : **Mathématiques appliquées et calcul scientifique**

présentée par

Claire TAYMANS

**Solving Incompressible Navier-Stokes Equations
on Octree Grids: Towards Application to Wind
Turbine Blade Modelling**

Under the supervision of:

Angelo IOLLO, Professor, University of Bordeaux
Michel BERGMANN, Research Scientist (CR), INRIA

Committee members:

Mr. Charles-Henri BRUNEAU	Professor, University of Bordeaux	President
Mr. Eric LAMBALLAIS	Professor, University of Poitiers	Referee
Mr. Kai SCHNEIDER	Professor, Aix-Marseille University	Referee
Mr. Haysam TELIB	PhD, CEO at OPTIMAD Engineering	Examiner
Mr. Jean-Yves GRANDIDIER	PhD, President of VALOREM	Guest
Mr. Andrea FERRERO	Research Engineer, Politecnico di Torino	Guest

28 September 2018

Remerciements - Acknowledgments

Je souhaite remercier mes directeurs de thèse Angelo Iollo et Michel Bergmann. J'ai beaucoup apprécié de travailler avec vous pendant ces années de thèse. Nos échanges professionnels et personnels ont été très enrichissants.

Je remercie vivement Kai Schneider et Éric Lamballais pour avoir accepté de rapporter mon mémoire de thèse. Vos remarques ont été très précieuses pour améliorer le manuscrit de la thèse. Je remercie également Charles-Henri Bruneau pour avoir présidé ce jury de thèse, ainsi que Haysam Telib, Jean-Yves Grandidier et Andrea Ferrero pour y avoir participé.

J'ai pu profiter, pendant ces années à l'IMB, d'une ambiance de travail excellente avec l'ensemble des membres des équipes Memphis et Monc. J'ai apprécié énormément les pauses café avec les collègues et amis italiens où nous avons échangé sur nos différentes cultures. J'ai également trouvé dans le bureau d'Antoine et Stefano une aide importante et des discussions intéressantes et amusantes. Je remercie particulièrement Emanuela avec qui j'ai partagé mon bureau, mes coups durs lorsque la thèse n'avancait pas et mes joies (parfois ça arrive tout de même en thèse !). Son soutien m'a été très précieux.

Je remercie également mes collègues chez Valorem. J'ai beaucoup appris sur le domaine éolien à vos côtés. J'ai pu effectuer cette thèse grâce à l'équipe Valeol de Serge, Benjamin, Karine, Bastien et Nicolas et je leur suis très reconnaissante. L'équipe du Bureau d'Études, auquel je suis à présent rattachée, m'a beaucoup aidé dans le travail expérimental décrit dans la thèse. Je les remercie chaleureusement pour leur aide et pour la très bonne ambiance dans les bureaux. Toute l'entreprise Valorem constitue un environnement amical et de travail excellent et cela m'a permis de travailler sereinement sur ma thèse.

Ma famille m'a apporté un grand soutien pendant ces années de thèse. Merci à vous et tout particulièrement à mon mari Alexandre.

Computing Platforms

Several experiments presented in this paper were carried out using the PlaFRIM experimental testbed, supported by Inria, CNRS (LABRI and IMB), Université de Bordeaux, Bordeaux INP and Conseil Régional d'Aquitaine (see <https://www.plafrim.fr/>).

Allocated computational hours on the supercomputer Occigen, operated by CINES (the computing centre for the French Ministry of Higher Education and Research) and acquired by GENCI (Grand Equipement National de Calcul Intensif) allowed to perform major simulations of the present work.

Finally, the supercomputer Turing, operated by IDRIS (Institute for Development and Resources in Intensive Scientific Computing) has been used for the simulations with Cartesian meshes. IDRIS is the major centre of very high performance intensive numerical computation

for the French National Centre for Scientific Research (CNRS).

Contents

Introduction	1
Chapter 1. Wind Turbine Aerodynamics and Existing Modelling Tools	5
1 Wind Turbine Aerodynamics	5
2 Existing Simplified Numerical Models for Preliminary Design	7
2.1 Blade Element Momentum (BEM)	8
2.2 Generalized Actuator Disc Model	12
2.3 Actuator Lines	12
3 Computational Grid	13
4 Existing Numerical Models in Computational Fluid Dynamics	14
4.1 Numerical Schemes for Wind Turbine Modelling	16
4.2 Numerical Schemes with Octree Data Structure	16
4.3 Numerical Schemes for the Advection Term	17
Chapter 2. Numerical Modelling of Flow around a Wind Turbine on Octree Grids	19
1 Immersed Boundary Method	20
1.1 Penalized Equations	20
1.2 Temporal Discretization	21
1.3 Boundary Conditions	22
2 Two-Dimensional Finite Volume Solver	22
2.1 Numerical Scheme for the Divergence Operator	23
2.2 Numerical Scheme for the Laplace Operator	26
2.3 Semi-Lagrangian Scheme for the Advection Term	29
3 Extension of the Methods to the Three-Dimensional Case	31
3.1 Numerical Scheme for the Divergence Operator	31
3.2 Numerical Scheme for the Laplace Operator	32
3.3 Semi-Lagrangian Method	33
4 Wall Functions for High-Reynolds Number Flow	34
4.1 Large Eddy Simulations	35
4.2 Near Wall Modelling	37
Chapter 3. Validation of the Numerical Tool	41
1 Consistency and Accuracy of the Navier-Stokes Solver on Octree Grids	41
2 Validation with Data from the Literature for the Two-Dimensional Solver	43
2.1 Fixed Cylinder Case	43
2.2 Airfoil Case	50
3 Stability Issue with the Semi-Lagrangian Scheme for Three-Dimensional Solver	50

4	Validation with Data from Literature for Three-Dimensional Case	51
4.1	Sphere Case	53
4.2	Cylinder Case	54
5	Comparison between Cartesian and Octree Solvers	57
Chapter 4.	Collection and Analysis of Operating Wind Turbine Data	65
1	Specifications	66
1.1	Wind Measurements	66
1.2	Blade Data	67
2	Technical Solutions for Measurements	68
2.1	Meteorological Mast	68
2.2	Blade Measurements	69
2.3	Position of Sensors on the Wind Blade	70
3	On-site Installation	71
3.1	Meteorological Mast Installation	71
3.2	Sensors on Blade	74
4	Analysis of Experimental Results	76
4.1	Methodology	76
4.2	Synthesis of the Experimental Work	79
Chapter 5.	Towards Application to the Study of Wind Gust Effects on a Blade	83
1	Coupling with the Solid Model of the Wind Blade Apx48	83
1.1	Structural Model	83
1.2	Fluid-Structure Coupling in the Three-Dimensional Solver	86
2	Rotating Framework for Wind Blade Modelling	88
2.1	Boundary Conditions and Rotating Models	88
2.2	Preliminary Results on a Rotating Wind Blade	90
3	Adaptive Mesh Refinement Process	92
3.1	Two-Dimensional Cases	94
3.2	Three-Dimensional Cases	95
4	Rotating blade submitted to a gust	99
	Conclusion	103
	References	107
	List of Figures	113

Introduction

Industrial Context

Almost all states worldwide agreed during the 2015 United Nations Climate Change Conference that was held in Paris on the reduction of climate change. One of the goal is the limiting of global warming to "well below 2°C" compared to pre-industrial levels. The agreement calls also for zero net anthropogenic greenhouse gas emissions to be reached during the second half of the 21st century. This conference didn't envisage concrete actions to implement, no sanction is moreover planned in the case where a country doesn't make any effort. However, this agreement proves that the world is realizing the importance to limit climate change. To reach this objective of reducing greenhouse gas emissions, the development of renewable energies will play a fundamental role and a likely winning scenario is a mix of several sources of energy (wind, solar, hydraulic, methanation, ...). Inside this mix, wind energy has several significant advantages that will certainly lead to its further development in the next years. Wind is a local resource and its cost is becoming cheaper and cheaper thanks to technological improvements. Moreover, unlike the solar energy, wind can be available in a more regular way all along the day and night. France has also the advantage of having several main windy corridors that allow to have almost always somewhere the wind resource available. The possible extension to offshore wind plants could also lead to a significant development of wind energy. Indeed, the impacts are lower than onshore, while having a constant and regular wind. This could occur in the near future since costs of installations are constantly decreasing and the cost of offshore wind is currently lower than the new generation of nuclear power plant (EPR) under development in Flamanville (France).

This development of the wind energy sector is thus leading to technological advances. It can be seen in figure 1 that shows the evolution of the size of wind turbines over time. Thanks to the large blades, wind turbines are now able to catch wind even in areas where the wind resource is quite small. Concerning Valorem, the company is involved in the whole value chain of renewable energy with the development part, the construction of plants, the operation of wind farms and maintenance. The development of a wind plant consists in identifying the potential areas, the associated constraints and in obtaining the administrative authorization. Also, at this stage, some studies should be done to have an estimation of the wind resources and of the future energy production (as a function of layout of wind turbines and wind turbines themselves). Environmental studies should also be done to estimate the impact of the future wind farm on the landscape and the fauna. Finally, electrical connection studies need to be integrated into the development phase to plan the grid connection of the wind farm. Valorem owns also several power plants (solar, wind or hydroelectric). Concerning the wind sector, which is the major energy developed at Valorem, it is essential to predict in an accurate way the future production of the wind plant. Indeed, the business model of the future wind farm is highly dependent on that element and a valid business model is necessary for the viability of Valorem. Therefore, the use of adequate tools is essential for the wind assessment studies. For a company as Valorem,

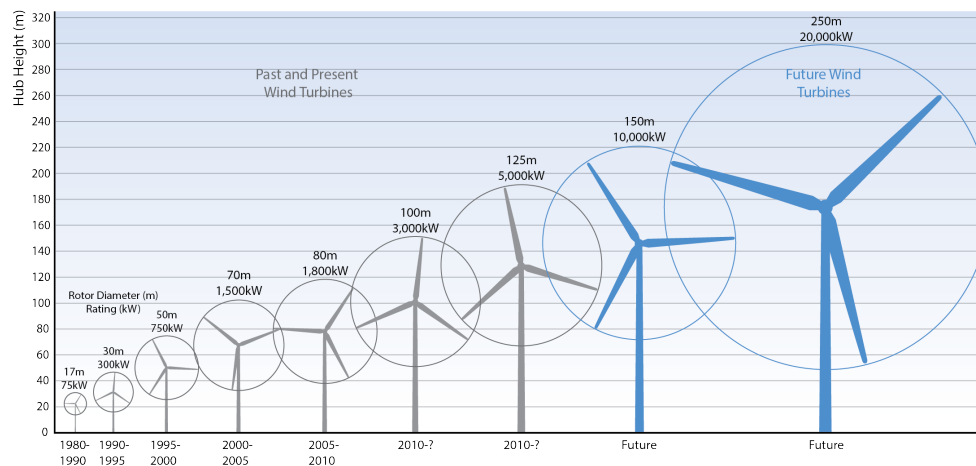


Figure 1: Evolution of the size of wind turbines over time. Source *www.windeurope.org*

two subjects are of high interest: lifetime of wind blades, that are the most sensitive parts of the turbine and production performance.

The simulation of the effects of a gust on a wind turbine is a fundamental step in the determination of the maximum possible load encountered by the wind turbine during its life cycle. Indeed, the magnitude of a gust can be sufficiently high to increase the loads up the point of fracture. Catastrophic failures are reported in the literature for both wind turbine blades and towers (see for example [23] and [22]). Therefore, Valorem was part of the European Horizon 2020 project AeroGust ("Aeroelastic Gust Modelling"). The aim of the project is to better understand the effects of wind gusts on both the aircraft and on wind turbine blades. As lifetime extension of the blades is of high interest for Valorem, being part of that project was a great opportunity.

Background to the Research

Since the Reynolds number of the involved bodies is quite large (with an order of several millions) and the flow field is fully turbulent, a large range of turbulent structures is present. In order to achieve a good representation of the flow by using a reasonable amount of computational power, a turbulence model is required. Recently, several promising simulations of the flow field around wind turbines have been carried out by using Large Eddy Simulations (see for example [42], [63] and [49]). This is the path which will be followed in the present work devoted to the simulation of a full wind turbine. Finally, very few existing numerical tools are able to compute these analyses. Moreover, a key parameter is the time needed by engineers to define an appropriate grid discretization for a given problem. This is a major issue explaining why INRIA and VALOREM chose to develop an immersed boundary code based on octree discretization where the definition of a mesh is straightforward.

An immersed boundary code on Cartesian grids has been developed in-house by INRIA, prior to the present thesis, with high-order numerical schemes. This code has been widely validated, so, this Cartesian code has been used as a proof of concept of new methods before implementing them in the octree solver. These new methods that have been validated with the Cartesian code are the following: Large Eddy Simulation with the Vreman subgrid model, the rotating frame of reference (to simulate the wind blade), the wall law (to describe the boundary layer) and the

coupling with the structural model of the blade.

The aim of the present thesis is to develop a first base of a numerical tool that will allow to better understand the behaviour of wind blades. This tool could then be improved and adapted to different applications as wished by Valorem.

Outline of the Thesis

The first chapter describes the basic aerodynamics of wind turbine to highlight the data of interest to compute. The existing numerical models used for preliminary design are then discussed. A particular attention is brought on the Blade Element Momentum theory which is widely used in industrial process. Some numerical models in Computational Fluid Dynamics are then considered. We focused on those used for wind turbine modelling as it is our final application, and on existing schemes that are based on an octree discretization. Different existing computational grids have indeed been studied and the octree configuration shows several advantages for our application. Advection schemes have also been studied since it led to an issue as realized during the thesis.

All the numerical methods developed during this work are explained in the chapter 2. Firstly, the choice of an immersed boundary method is commented and the solution of Navier-Stokes equations in that context is described. The two-dimensional finite volume solver developed with quadtree discretization is then widely explained with the chosen stencils and interpolation methods. Each operator implemented in the code in order to solve the Navier-Stokes equations is validated thanks to the use of analytical functions and the order of convergence is computed. The extension of these original methods to three dimensional case follows with the same detailed explanations and the operators are preliminary validated by the use of analytical functions and the computation of the order of convergence. To end this chapter, the numerical models used to deal with high-Reynolds flow in our work are presented. Large Eddy Simulation with a Vreman subgrid model has been implemented to model turbulence. Wall functions are also employed to model the behaviour of the flow around the obstacles. Indeed, the octree discretization prevents us to be able to properly model the boundary layer near walls.

The third chapter explains the steps performed to validate the different numerical models developed. These validations are indeed the key part of the thesis. First of all, the space accuracy of the 2D and 3D solvers is evaluated. The two-dimensional code has then been widely tested for the case of the flow around a 2D circular cylinder at different flow regimes and for the case of the flow around an airfoil. Thanks to well-known benchmark for these test cases, we compared the results obtained with data from literature at different flow regimes and with different geometries. The three-dimensional solver has then been validated at different Reynolds numbers. Our final application being the case of a rotating blade, it was therefore needed to validate the solver for the case of high Reynolds flows. The steps of comparison with existing benchmarks in the literature occurred progressively. The case of the flow around a sphere has been firstly considered at low Reynolds number. The next step was the flow around a cylinder at higher Reynolds. The turbulence model as well as the wall function have been included in the simulations. The flow at $Re = 3900$ has been studied before a higher Reynolds of 140 000. This high Reynolds flow benchmark has also been used to compare the code developed in the present work with a Cartesian solver.

Experiments have been conducted in the present work and are presented in chapter 4. Very

little experimental data is indeed available with the geometry of blades, the structural model and measurements on the blade. And none of this correlates that with the incoming wind. The context of the European project AeroGust¹ allowed to collect this experimental data on an in-service wind turbine. All the specifications needed to conduct the experimental work have been determined. Different technical solutions have been investigated to measure both the wind and to get data from the blade and the final choice is presented. Then, on-site installation of the sensors has been planned and organized. Finally, the experimental data collected has been studied in order to better understand the effects of wind on the blade.

The final chapter describes the application of the solver developed which is the simulation of the flow around a wind turbine blade. Since we have available experimental deformation data of the wind blade APX48, the structural model of this blade has been also implemented in the octree solver. The way the coupling between the fluid and the structure is done in present work is also described. Studying the flow around a whole wind turbine required too many computational time. Therefore, to reduce that, it was chosen in the thesis to study only one blade and moreover, to use a rotating framework to focus the computational domain on the wind blade. An interesting feature of our developed solver on octree is the possibility to perform Adaptive Mesh Refinement of the grid to optimize the grid size while being as precise as possible on the wake of the rotating wind blade. The implementation of this process is described together with the preliminary results obtained. Finally, all the features implemented have been put together to perform simulations of the flow over a rotating blade with real operating conditions of the wind turbine.

¹www.aerogust.eu

Chapter 1

Wind Turbine Aerodynamics and Existing Modelling Tools

This chapter explains the operation of a wind turbine with the aerodynamic forces that act on the blades. In order to compute these quantities and estimate the power extracted by some wind turbine, several kind of modelling exist. Simplified numerical models can be used for preliminary design. These models are mostly based on the Blade Element Momentum theory as described in [54, 83, 76]. These models are bi-dimensional whereas the flow around wind turbines is intrinsically tri-dimensional. Therefore, a significant work of modelling is necessary to accurately reproduce the physics of the flow with such models. An other approach is the Computational Fluid Dynamics (CFD) that rest on the Navier-Stokes equations and their approximation. As this subject is very wide, a bibliographic study has been performed in order to highlight the numerical schemes adapted to the problem of the flow around a rotating wind turbine.

1 Wind Turbine Aerodynamics

Aerodynamics of the blades has a significant importance on the energy efficiency of the wind turbines. The aerodynamic forces exerted on the blades can be seen in figure 1.1. U is the incoming wind speed and U_ω corresponds to velocity resulting from the rotational speed of the wind turbine. So, composition of wind speed and rotational speed is named relative speed (U_r) and corresponds to the velocity arriving on the blade. At blade root, the rotational speed is low which implies that the relative speed is really close to the wind speed. On the contrary, at tip blade, the rotational speed is significantly higher than the wind.

The parameter β represents the sum of the twist angle at considered radius and of the pitch angle, which is the orientation angle of the blade around its axis (this angle varies along time). β , named the angle of incidence is the angle between the chord line and the rotational plane of the rotor. α is the angle of attack, being the angle between the chord line and the relative speed (U_r). The flow angle, Φ can then be calculated as: $\Phi = \alpha + \beta$.

Due to the air flow around the airfoil, two forces exerted on the blade: drag (D) that act opposite to the relative flow and lift (L). Lift is the component perpendicular to the oncoming flow direction and corresponding to the depression due to acceleration of the fluid on the suction side of the airfoil. The sum of these two vectors forms the aerodynamic force F and multiplying it with the distance to the rotor's center point allows to get the torque. Its tangential component (F_T) contributes to the rotation of the wind turbine and the normal component (or axial component, referred to the rotor axis) of the aerodynamic force (F_N) corresponds to the thrust,

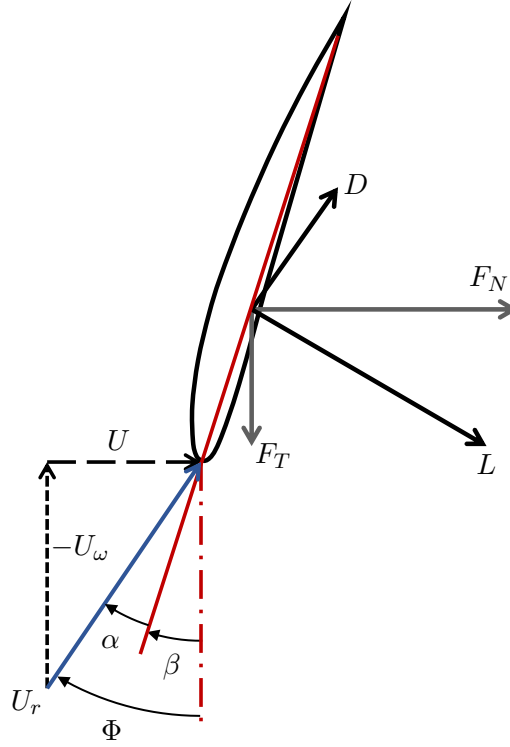


Figure 1.1: Sketch of the aerodynamic forces exerted by the flow around a wind blade airfoil

that subjects the rotor to mechanical stresses. The computation of these aerodynamic forces is necessary to evaluate the energy efficiency of the wind turbine.

The power extracted by a wind turbine is a significant variable to analyse and the power coefficient can be defined as the ratio between power extracted by the turbine and available wind power:

$$C_P = \frac{P_{extract}}{P_{available}} = \frac{P}{\frac{1}{2}\rho AV^3}, \quad (1.1)$$

with $A = \frac{\pi D^2}{4}$, the rotor swept area. In the same way, the thrust coefficient (notion defined above) can be introduced:

$$C_{Th} = \frac{F_N}{\frac{1}{2}\rho AV^2}. \quad (1.2)$$

The power coefficient C_P depends on the rotational speed of the rotor and can be expressed as a function of the tip speed ratio λ that represents the ratio between rotational speed at tip blade and the wind speed:

$$\lambda = \frac{\Omega R}{U}, \quad (1.3)$$

Ω being the rotational speed of the rotor (in $rad.s^{-1}$) and R the blade radius. The wind turbine torque T , as previously seen can be written:

$$T = \frac{P}{\Omega} = \vec{r} \wedge \vec{F}, \quad (1.4)$$

with \vec{r} being the position vector (vector from the origin of the coordinate system defined, to the point where the force is applied) and \vec{F} is the aerodynamic force vector.

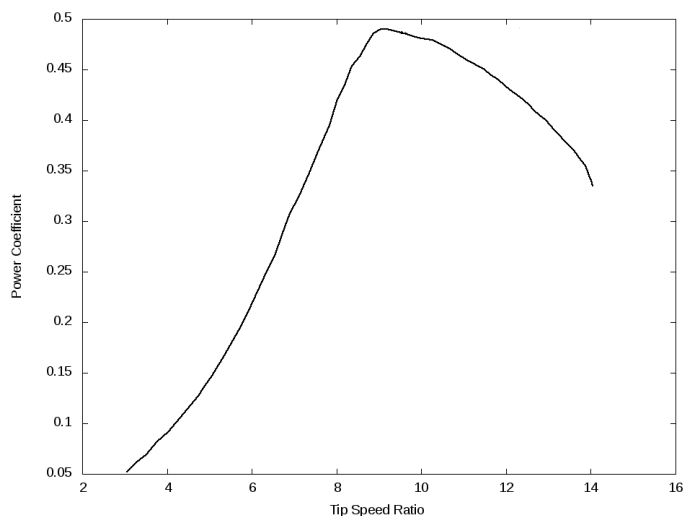


Figure 1.2: $C_P - \lambda$ curve for the Euros EU120 wind blade. Source *www.euros.de*.

The $C_P - \lambda$ curves are used in wind turbine design to determine the rotor power for any combination of wind and rotor speed. In figure 1.2 can be seen the corresponding curve for the Euros EU120 wind blade. For this blade, the maximal power coefficient is around 0.49 and is obtained for $\lambda_{opt} = 9.1$. That means, so, for a blade tip speed equal to 9.1 times wind speed. Using this curve as well as data from the wind turbine (its rated capacity, its minimal and maximal wind speed, ...) it is possible to retrieve the power curve of the wind turbine. The power curve of a wind turbine is a graph that indicates how large the electrical power output will be at different wind speeds. An example can be seen in figure 1.3.

Introducing the well-known Mach number as the ratio of flow velocity to the speed of sound, the incompressibility condition of flow around a wind turbine can be studied. Indeed, compressibility effects are small if the Mach number is less than 0.3. The speed of sound is equal approximately to 343 m.s^{-1} for air at 20°C . The maximal wind speed around a wind turbine appears at tip blade. Depending on the wind turbine model, its value is ranged between 70 m.s^{-1} for some 2 MW wind turbine and until 100 m.s^{-1} for some offshore 5 MW wind turbine. That implies Mach number included in the range $0.2 - 0.3$ for nominal tip speeds which supports the incompressibility assumption for the simulation of the flow around a wind turbine.

2 Existing Simplified Numerical Models for Preliminary Design

The existing tools to simulate wind turbine aerodynamics can be divided in two main categories: Computational Fluid Dynamics (CFD) and simplified models (or hybrid models). CFD computes the whole Navier-Stokes equations in a three-dimensional domain containing the wind turbine and evaluates the aerodynamic forces on the blades dynamically. Hybrid models replace the real wind turbine geometry by simplified models of the flow around the wind turbines. All the hybrid models are based on blade element momentum theory to compute the forces on the wind blades and are coupled with Reynolds' averaged Navier-Stokes equations (RANS). This section explains the blade element momentum (BEM) theory before describing actuator disc and actuator line models. These two models are named the way the blade geometry is replaced with.

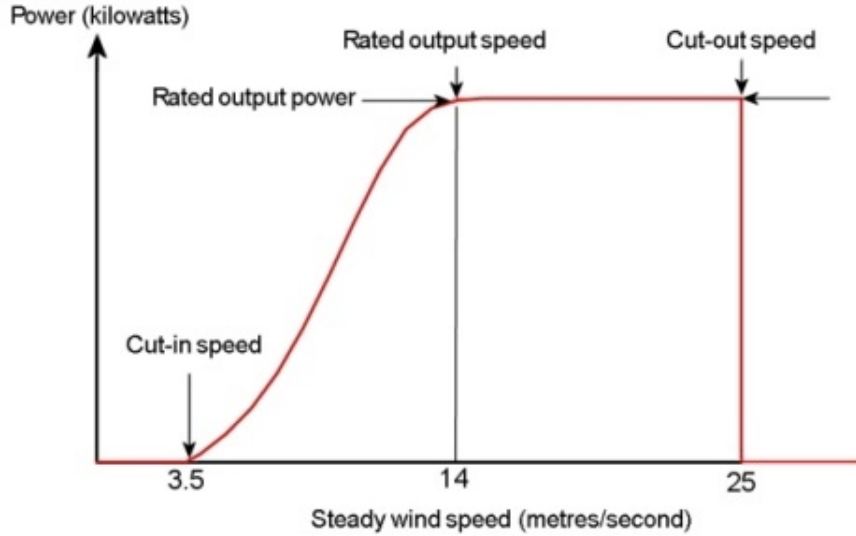


Figure 1.3: Example of power curve for some usual wind turbine.

2.1 Blade Element Momentum (BEM)

BEM theory is one of the oldest aerodynamic model in the field of wind engineering and is widely used in industrial world for the design of wind blades [82, 59, 11]. This model is a combination of two theories: axial momentum and blade element theory. This method has been developed by Betz in 1926 and is explained more recently in many articles [54, 83, 76]. Basically, in axial momentum theory, a unidimensional model is used to express the wind extracted energy and the flow is supposed to be incompressible. The blade element theory takes into account the fluid's rotational component that comes from blade's rotation. These theories are detailed below.

Axial momentum theory

In this theory, only an axial, incompressible flow is assumed, with airflow passing through the rotor without friction. The modelling consists in considering the wind turbine's rotor as a permeable disc without thickness which is set in motion by the airflow passing through it. As shown in figure 1.4, the air passing through the wind turbine rotor is subjected to a change of speed. The axial induction factor (often used in literature) is introduced and corresponds to the wind speed decrease fraction between upstream value V_1 and the value of wind speed going through rotor's plane V_2 :

$$a = 1 - \frac{V_2}{V_1}. \quad (1.5)$$

The variable V_4 corresponds to the downstream wind speed. Using the law of conservation of momentum, the theory considers that the force acting on the actuator disc reduces to the variation of axial momentum between upstream and downstream sections of the stream tube:

$$F_N = \dot{m}(V_1 - V_4). \quad (1.6)$$

Here, the mass flow through the rotor can be written $\dot{m} = \rho AV_2$, with A being the rotor area. In the same manner, using conservation of energy, the energy extracted by the disc can be written as:

$$E = \frac{1}{2} \dot{m}(V_1^2 - V_4^2). \quad (1.7)$$

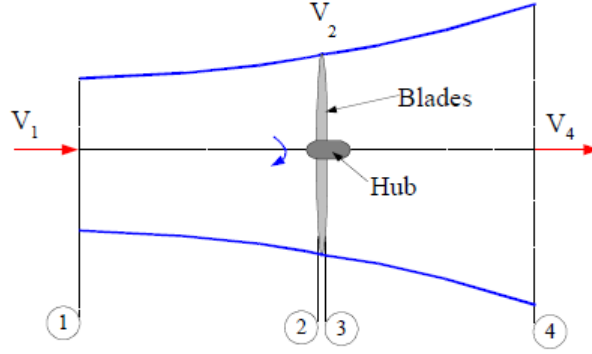


Figure 1.4: Axial stream tube around a wind turbine [41]

The power extracted from the flowfield by the rotor can thus be written in two manners. On the one hand, the power, by definition, can be written as the energy transferred per unit time. Using eq. (1.7), we finally have:

$$P = \frac{1}{2} \dot{m} (V_1^2 - V_4^2). \quad (1.8)$$

On the other hand, from the aerodynamics, with the axial flow condition, we have the definition: $P = F_N V_2$. With the use of eq. (1.6), we get:

$$P = \dot{m} (V_1 - V_4) V_2. \quad (1.9)$$

From eqs (1.9) and (1.8), can be deduced:

$$V_2 = \frac{1}{2} (V_1 + V_4). \quad (1.10)$$

By replacing V_4 in eq. (1.6), we get:

$$F_N = 2\rho A V_1^2 a(1 - a). \quad (1.11)$$

Energy efficiency

With eqs (1.1), (1.5), (1.9) and (1.10), the non-dimensional power coefficient is established as:

$$C_P = \frac{\rho A V_2^2 (V_1 - V_4)}{\frac{1}{2} \rho A V_1^3} = 4a(1 - a)^2. \quad (1.12)$$

By solving $\frac{dC_P}{da} = 0$ for eq. (1.12), the optimal conversion of energy possible can be found for $a = \frac{1}{3}$, which corresponds to $C_{P_{max}} = \frac{16}{27} \simeq 0.59$. The highest energy efficiency of a ideal wind turbine is so around 59%. This value is usually reffered to as the Betz limit. The axial momentum theory is a simple unidimensional model that doesn't take into account the air rotational flow.

Blade element theory

Blade element theory assumes an axial upstream flow far from rotor but adds the rotational speed of the flow at rotor's plane level. This model considers that the blade is constituted of several elements. The theory relies on the assumption that there is no aerodynamic interaction between different elements, so each element is independent from the others. Figure 1.5 shows an

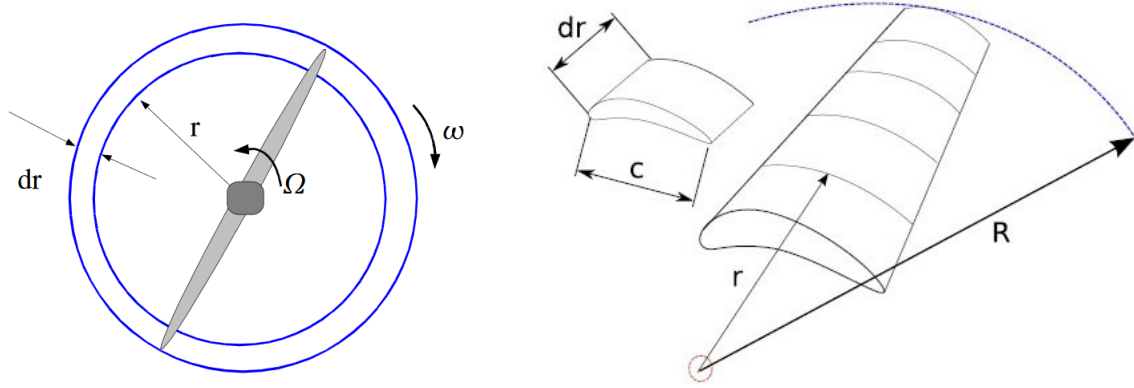


Figure 1.5: Blade element theory [41].

element with a thickness of dr that constitutes a blade and fictitious ring formed during blade's rotation. An other assumption of the theory is that the flow around each element is only bidirectional, thus, interaction forces on a blade element reduces to lift and drag. After computing the forces on each element, the global blade performance is evaluated by integrating the stresses exerted on each element.

The tangential flow induction factor a' is introduced in blade element theory. Its definition is $a' = \frac{\omega}{\Omega}$, with ω the angular velocity of the flow on the rotor's plane and Ω , the rotational speed of the rotor. Using notation of figures 1.1 and 1.5, these both relations can be written, according to the definition of a (see above) and a' :

$$U = U_\infty(1 - a), \quad (1.13)$$

$$U_\omega = \Omega + \omega r = \Omega r(1 + a'), \quad (1.14)$$

with U_∞ that corresponds to the far upstream wind speed. Then, geometrically, the following relations can be written:

$$\tan(\Phi) = \frac{U_\infty(1-a)}{\Omega r(1+a')}, \quad (1.15)$$

$$U_r = \sqrt{U_\infty^2(1-a)^2 + \Omega^2 r^2(1+a')^2}. \quad (1.16)$$

The basic hypothesis of the theory is that the force exerted on a blade element equals variation of momentum of the air passing through the blade. Thus, lift and drag are:

$$dL = \frac{1}{2}\rho U_r^2 B C_L dS, \quad (1.17)$$

$$dD = \frac{1}{2}\rho U_r^2 B C_D dS, \quad (1.18)$$

with C_L and C_D being respectively the lift and drag coefficients. $dS = c dr$ is the surface of the considered blade element and B corresponds to the number of blades. By projecting, the tangential force (related to torque) acting on a blade element can be obtained:

$$dF_T = \frac{1}{2}\rho c dr U_r^2 B (C_L \sin(\Phi) - C_D \cos(\Phi)). \quad (1.19)$$

The torque exerted on a blade element (dT) being the tangential force time the radius, we can write:

$$dT = \frac{1}{2}\rho c r dr U_r^2 B (C_L \sin(\Phi) - C_D \cos(\Phi)). \quad (1.20)$$

This is the force that need to be computed in order to get the power developed by the wind turbine. In the same way, the normal force (related to thrust) can be written:

$$dF_N = \frac{1}{2}\rho c dr U_r^2 B (C_L \cos(\Phi) + C_D \sin(\Phi)). \quad (1.21)$$

By introducing the definition of local solidity $\sigma = \frac{Bc}{2\pi r}$, eqs (1.20) and (1.21) become:

$$dT = \sigma \pi \rho r^2 dr U_r^2 (C_L \sin(\Phi) - C_D \cos(\Phi)), \quad (1.22)$$

$$dF_N = \sigma \pi \rho r dr U_r^2 (C_L \cos(\Phi) + C_D \sin(\Phi)). \quad (1.23)$$

Combination of both theories - Blade element momentum

Eq. (1.11) can be rewritten with notation from blade element and taking an area of the blade element $A = 2\pi r dr$ allows to obtain:

$$dF_N = 4\rho \pi r dr U_\infty^2 a(1-a). \quad (1.24)$$

In the same way, using notation of figure 1.4 and extending axial momentum theory, variation of momentum in tangential direction gives us the tangential force that is exerted on the blade:

$$dF_T = \rho A V_2 V_T, \quad (1.25)$$

with $V_T = \omega_3 r$, V_T being the tangential velocity of flow just after the rotor and ω_3 is defined as the angular velocity. In order to determine ω_3 , the conservation law of angular momentum is applied between section 1 and 2 of figure 1.4.

$$\omega_1 r_1^2 = \omega_2 r_2^2. \quad (1.26)$$

As the upstream flow far from the turbine is only axial, we have $\omega_1 = 0$. That implies from eq. (1.26) that $\omega_2 = 0$. It can be deduced that angular velocity is subjected to a discontinuity at rotor's plane, this is due to the torque applied on the rotor. The angular velocity of the flow on the rotor's plane is the average of those just before and after the rotor.

$$\omega = \frac{\omega_2 + \omega_3}{2} = \frac{\omega_3}{2}. \quad (1.27)$$

Finally, using definition of a and a' , eq. (1.25) becomes:

$$dF_T = \rho(2\pi r dr) V_1 (1-a) 2a' \Omega. \quad (1.28)$$

We obtain then the torque, noting that V_1 corresponds to U_∞ using notation of figure 1.1.

$$dT = 4\rho U_\infty (1-a) a' \Omega r^3 \pi dr. \quad (1.29)$$

With both theories, torque and thrust have been expressed in two different forms. We can thus solve the problem and obtain with eqs (1.23) and (1.24):

$$a = \frac{1}{\frac{4\sin^2(\Phi)}{\sigma(C_L \cos(\Phi) + C_D \sin(\Phi))} + 1}. \quad (1.30)$$

Use of eqs (1.22) and (1.29) allows to get:

$$a' = \frac{1}{\frac{4\sin(\Phi)\cos(\Phi)}{\sigma(C_L\sin(\Phi) - C_D\cos(\Phi))} + 1}. \quad (1.31)$$

Blade element momentum theory is thus a simplified model that allows to compute the forces that are applied on a rotor and deduce the power extracted from the rotor. The data needed for BEM are the following:

- radius of blade R ,
- rotational speed of the rotor Ω ,
- number of blades B ,
- distribution of airfoils with associate chord length c and twist angle β ,
- polar curves of each airfoil along the blade, that allows to know drag and lift coefficient of each airfoil.

As no analytical expression exists for the flow induction factors a and a' , an iterative process is used to determine them. Convergence of results of a and a' means that axial momentum and blade element theories converge.

2.2 Generalized Actuator Disc Model

Generalized actuator disc (GAD) method can be seen as an extension of BEM theory. Indeed, the wind turbine's rotor is modeled as a zero thickness disc in a fluid flow. But, instead of considering each blade element independent from each other as in BEM, GAD model computes Navier-Stokes equations to get the velocity field, especially on the blades. The aerodynamic forces are then obtained with BEM method as described above. Axisymmetric versions of Navier-Stokes equations are developed for GAD model. Numerical implementation can be based on finite difference as developed by Sorensen and Mikkelsen [83] and presented by Mikkelsen [54]. The model consists in supposing an axisymmetric tube, thus, a 2D plane (r, z) is considered, discretized with Cartesian mesh, where Navier-Stokes equations are solved.

Previously to this thesis, a work has been done with development of two codes based respectively on GAD and BEM theory. These models have been compared with experimental results obtained in a wind tunnel on a two blade wind turbine of 5 m diameter. The experimental process is described in [28]. The predicted power as a function of wind speed computed with BEM and GAD models have been compared with these experimental results and can be seen in figure 1.6. The results show that for high wind speeds, both models are inaccurate. This explains why an other approach with full 3D Computational Fluid Dynamics (CFD) has been studied.

2.3 Actuator Lines

The actuator line model consists in replacing the actual geometry of the rotor blades by lines carrying body forces corresponding to the loading of rotor blades. This technique was originally developed by Sørensen and Shen [81] and presented later in [27]. The method is based on the same ideas as GAD. Differences are, first, the solving of three-dimensional Navier-Stokes equations in the domain. Large Eddy Simulation is employed, that applies a low-pass filter on

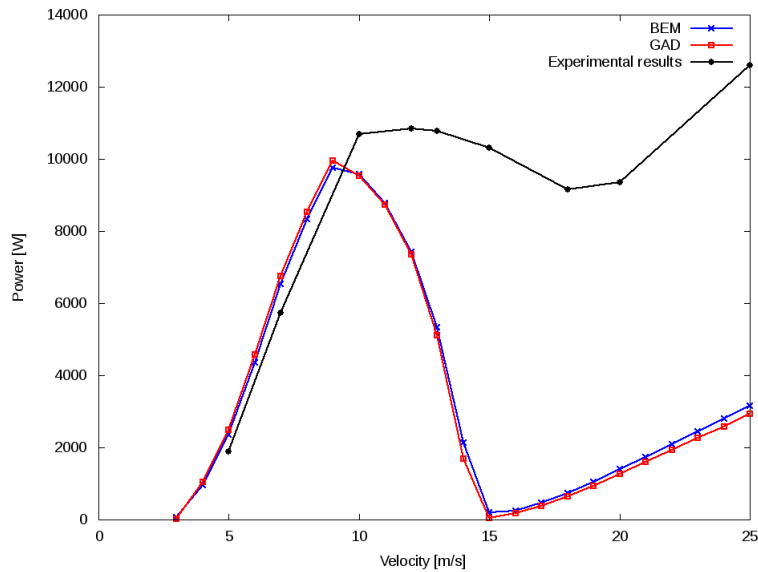


Figure 1.6: Comparison of predicted power for BEM and GAD models with experimental data of NREL UAE phase 6 (experimental process described in [28]).

the equations. So, the scales below the grid scale are modelled with a sub-grid model. Secondly, the influence of the wind turbine is modelled with lines instead of elements. The actuator lines are fully coupled with the CFD solver so that they can deflect and rotate. This allows an interesting simulation of the rotor’s wake. However, as in GAD, the forces applying on the blade are obtained with the polar curves that gives the drag and lift coefficients for a well-known airfoil geometry as a function of the angle of attack. These models are thus adapted only when simulating an existing blade with a well-known airfoil distribution.

3 Computational Grid

Different categories of grids exist to discretize Navier-Stokes equations. Structured grids are characterized by regular connectivity with quadrilateral elements in 2D and hexahedra in 3D. This kind of mesh has a better convergence and higher resolution compared to unstructured grids [17, 33]. Unstructured meshes typically employ triangles in 2D and tetrahedra in 3D. They are well adapted for complex geometries so that the mesh can fit the obstacles. Finally, hybrid grids exist that are mixing structured and unstructured blocks.

The octree data structure

The octree structure can be viewed in its dual nature, the tree and the grid, as it is represented in figure 1.7. A quadtree is a data structure in which each internal node has exactly four children. Extension in 3D gives octree data structure where eight children exist. This data structure was named a quadtree by R. Finkel and J. L. Bentley in 1974 [31]. Domain discretization using square octree grids is also explained in [74]. The tree is defined as a collection of interconnected cells, also called octants. Each refinement generates disjointed subtrees, whose nodes are called children. Nodes without children are named leaves.

An octree data structure is said to be linear when only the leaves of the tree are stored in memory, which means that the usage of computational resources is optimal. While ordering the cells of a Cartesian grid is straightforward, the same cannot be said about block structured

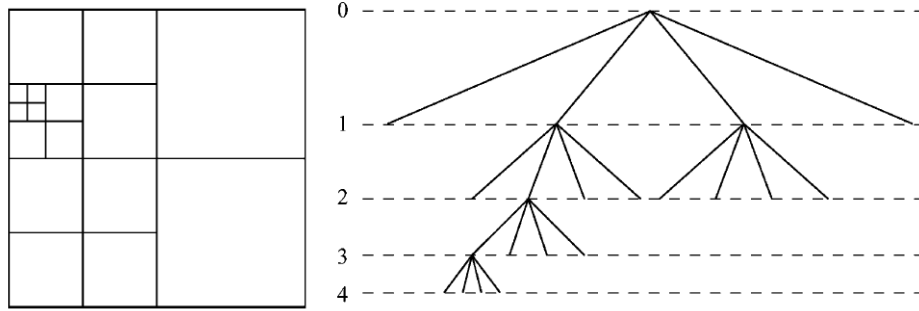


Figure 1.7: Example of quadtree discretization and corresponding tree representation. Found in [74].

meshes, sorting is achieved using space-filling curves that cross each element of the grid once. There are many different examples of space-filling curves, one of these is the Z-order, introduced by Morton [60], who also proposed a procedure, commonly known as the Morton code, to identify cells neighbours, based on simple manipulations of binary numbers. This algorithm avoids the evaluation and the storage of complex connectivity matrices, contributing to the low memory usage of octrees. The Z-order is also used to compute the parallel partition of the domain thanks to the linearity of the Z-curve. The communication between two different subdomains is made thanks to one layer of ghost cells where the information is shared between two processors.

Quadtree and octree allow a flexibility in mesh refinement and coarsening with the use of different levels of refinement in a unique grid and also have the advantages of regular meshes as seen previously. Indeed, Cartesian grids imply a high computational effort since the whole domain has the same refinement level. On the other hand other discretizations as octrees allow to refine on areas of high interest (on one body's wake, close to obstacles ...) and coarsen elsewhere. Meshing a computational domain in a complex flow context and dealing with complex geometries while making the grid fit the obstacles can rapidly become very time-consuming. Moreover, changing a small part of the geometry implies doing the whole mesh again. However body-fitted grids allow to get a high accuracy around obstacles which is a significant advantage. In the context of the present work, one aim is to develop a tool that can be used by an engineer that would not be necessary a specialist in CFD. Also, for the application to the simulation of wind turbines with blades that rotate along time the use of an adaptive mesh that is not too long to compute is well adapted. Therefore, the choice to use octree grids has been done in the present work.

4 Existing Numerical Models in Computational Fluid Dynamics

We consider the incompressible Navier-Stokes equations in a general and bounded domain Ω that can be seen in figure 1.8. Let Ω_f be the fluid domain and Ω_s be the solid domain, $\Omega = \Omega_f \cup \Omega_s$, $\Gamma_s = \partial\Omega_s$ is the boundary of the solid domain and $\Gamma_f = \partial\Omega_f$ is the boundary of the fluid domain.

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \vec{\nabla}) \mathbf{u} = -\frac{1}{\rho} \vec{\nabla} p + \nu \Delta \mathbf{u} \text{ in } \Omega_f, \\ \nabla \cdot \mathbf{u} = 0 \text{ in } \Omega_f, \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) \text{ in } \Omega_f, \\ \mathbf{u} = \mathbf{u}_b(\mathbf{x}, t) \text{ on } \Gamma_s, \\ \mathbf{u} = \mathbf{u}_f(\mathbf{x}, t) \text{ on } \Gamma_f, \end{cases} \quad (1.32)$$

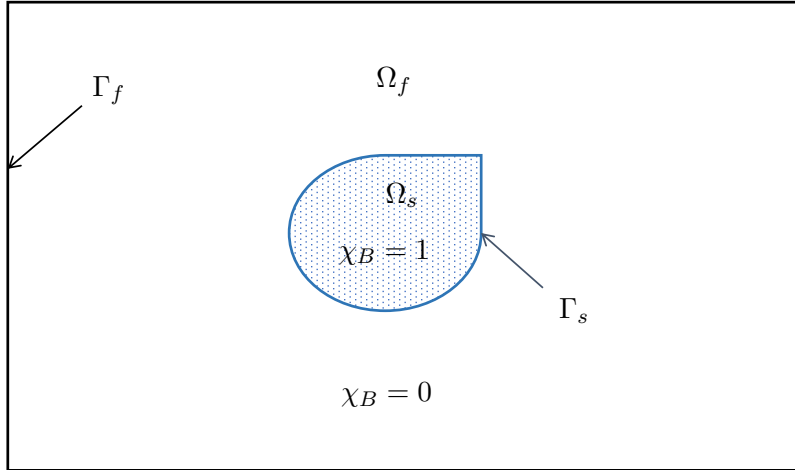


Figure 1.8: Sketch of the computational domain

where p is the pressure, \mathbf{u} the velocity field, ρ is the density of fluid and ν the kinematic viscosity.

Spectral methods can be used in Computational Fluid Dynamics to solve Navier-Stokes equations. These methods are highly accurate in the context of incompressible flows with a simple geometry. But, when simulating with more complex geometries, a full spectral approach is not adequate. Laizet and Lamballais [46], developed an intermediate tool between fully spectral Navier-Stokes solvers and more versatile codes based on standard numerical schemes. They indeed used a spectral space with a Fourier representation to solve the Poisson equation, which allows to have a very accurate incompressibility condition. The other terms (convective and viscous) are discretized with high-order compact schemes. All this work is performed in the context of Cartesian grids with an immersed boundary method (this method will be discussed in the next chapter). The use of Cartesian meshes implies high computational cost when dealing with large computational domains as for the application of the flow around wind blades.

A pseudo-spectral method has also been developed by Schneider [77]. The penalized Navier-Stokes equations in vorticity-velocity formulation are discretized in space and time. For more information about penalized equations, the reader is referred to the next chapter. The pseudo-spectral method is used for the space discretization, whereas for the time discretization, a variable time-stepping semi-implicit scheme is employed for the diffusion term and a second-order Adams-Bashforth scheme is developed for the convective and the penalization term. The pseudo-spectral method consists in transforming the vorticity and other variables to Fourier space for the computation of spatial derivatives. On the other hand, the terms containing products, as the convective and penalization terms are calculated in physical space. This work has been performed in a two-dimensional case and a three-dimensional extension is required to fit our present application to wind blade modelling.

The different discretization methods existing to solve the Navier-Stokes equations is discussed below. Schemes used for flow around wind turbines is first investigated. Then, existing discretizations based on quadtree/octree data structure is explained. The schemes used for the advection term in these contexts are then described.

4.1 Numerical Schemes for Wind Turbine Modelling

The three-dimensional flow solver Ellipsys3D was developed in collaboration between the Technical University of Denmark [53] and the former National Laboratory for Sustainable Energy (Risø) [84]. This code has been developed for the purpose of modelling the wind turbine's wakes. Incompressible Navier-Stokes equations are solved with a finite volume method. Pressure and velocity variables are collocated. A block structured mesh is used with general curvilinear coordinates. Large Eddy Simulation (LES) is performed by applying a low-pass filter to the Navier-Stokes equations, resulting in filtered velocity field. A sub-grid model is then employed to model the scales below the grid size. This solver is well documented and is open-source under the MIT License.

The solver Overflow CFD has been studied since a lot of features are available in this research code developed by NASA Ames Research Center. Chaderjian described in [19] the solver and its application for the modelling of helicopter's rotor which is quite close to the wind turbine application. Navier-Stokes equations are solved with central finite difference discretizations. This code employs an overset grid approach with totally structured meshes. An overset grid system means that a near-body mesh exist in the vicinity of the rotor, body-fitted and with curvilinear coordinates allowing a thin grid on the blades to resolve the boundary layer. This near-body mesh is coupled with an off-body grid system to extend the computational domain in the far field. In the article the off-body mesh is itself composed of several Cartesian grids as "brick grids". An Adaptive Mesh Refinement process (AMR) is implemented in Overflow, that allows to refine the off-body grid system in areas of interest, thus, the rotor wakes can be highly resolved. An other feature of Overflow is the coupling with a structural finite element model.

Commercial multi-purpose solvers as ANSYS CFX, Fluent or Star CCM⁺ are also used in the context of flow around wind turbines. These solvers are not well documented which can be problematic in the understanding of obtained results. A multi-grid structured mesh was used in [47] for the simulation of flow around a wind blade and nacelle, except at the rotational centre of the domain, where an unstructured grid is employed. The results obtained showed good results but some knowledge in Computational Fluid Dynamics are needed to be able to analyse the results obtained and to use adequate numerical models depending on the flow characteristics.

For all solvers described above, the meshing of the domain is not trivial and can be very time-consuming. A particular attention has then be accorded to the discretizations of Navier-Stokes equations using an octree data structure.

4.2 Numerical Schemes with Octree Data Structure

An immersed boundary method was introduced by Peskin [69] to simulate fluid-structure interactions. This method with fractional-step scheme to solve incompressible Navier-Stokes equations was proposed later by Bergmann and Iollo in [10] in the context of Cartesian grids. Fractional-step methods are based on the Chorin-Temam schemes [21, 87]. The immersed boundary allows to avoid the use of bodyfitted grids, thus, this method results in a significantly easier mesh process. Finite difference methods are equivalent to finite volume when dealing with Cartesian grids and, in this article, space discretization is performed using a centred second order finite-difference approximation for the diffusive term. This discretization is straight-forward to implement on a Cartesian grid.

Popinet [74] introduced a Navier-Stokes solver using an octree data structure. He proposed a second-order accurate in space and time solver on octrees. All variables are stored at the cell-centre. A Poisson equation for pressure is solved in order to maintain the second-order accuracy of the solution but the gradients have a first-order accuracy. A large stencil is used when a level jump occurs. The variables are indeed rebuilt as ghost values using neighbours of neighbours of

the cell. A constraint named grading condition is also imposed on the grid with the fact that levels of diagonally neighbouring cells can not differ by more than one.

Min and Gibou [55] developed a solver for Navier-Stokes equations on quadtree and octree data structures. They used the supra-convergent Poisson solver described in [56] with the same authors. Variables are stored at the cell's node which imply that schemes are straightforward to implement. This node-based discretization has a quite wide stencil with the use of neighbours of neighbours. It is demonstrated that second order accuracy in the L_∞ norms is obtained for the solution of Poisson equation and its gradients. The diffusion term is treated with a stable backward finite difference scheme.

In [65], Olshanskii et al. developed a finite difference solver on staggered grid. The pressure variable is thus defined cell-centered and velocity component are at cell face. The advantages of using staggered location of unknowns are the enforcement of the incompressibility condition and the well-known pressure stability of such schemes. But, that implies more difficulty in building higher order methods on octree grids. When a level jump occurs, the finite difference pressure gradient uses a stencil with only the four smaller neighbours and the larger cell sharing the face.

Losasso et al. [50] developed an octree data structure method where pressure variable is stored at the cell centre, while the velocity components are defined on the cell faces. At the level jumps, a ghost value of pressure is computed in order to get pressure gradient at face center. A standard central differencing is used even if it does not define the pressure gradient exactly at cell face. It is still shown that this approximation is convergent with first order accuracy. It has however a strong mesh constraint, since this method only allows one face with a level jump.

Another approach with a finite volume scheme was described in [6] by C. Batty. Batty's stencil is based on diagonals instead of axis directions as previously seen and is wide with the use of neighbours of neighbouring cells. This method allows to compute directly the ghost value at level jumps instead of doing an interpolation first. Quadratic interpolations are constructed, which allow to get a second-order accuracy of the solution and its gradients. This method is however only described in the two-dimensional case and can not be directly applied in 3D.

4.3 Numerical Schemes for the Advection Term

The left-hand side of eq. (1.32) corresponds to the advection term of Navier-Stokes equations. Different schemes for its treatment exist in the literature. The semi-Lagrangian method is a widely used model. It consists of being in an Eulerian framework but with discrete equations coming from the Lagrangian perspective. It involves backward time integration of a characteristic equation to find the departure point of a fluid particle arriving at an Eulerian grid point. Interpolation formulas are then used to recover the value of solution at such points. The advantage of semi-Lagrangian compared to Lagrangian framework is the fact that a well-defined grid is conserved. Indeed, a Lagrangian method, that follows the flow along time, could result in large distortions of the mesh. Then, semi-Lagrangian methods allow to avoid computing the convective term. A feature of semi-Lagrangian schemes is the unconditional stability. As we have a time step restriction, when using explicit schemes, imposed by the CFL condition to be proportional to the smallest grid cell, the stability of semi-Lagrangian scheme removes that restriction. Several examples of semi-Lagrangian methods implemented are reported in the literature [93, 64, 55, 36, 48].

Xiu and Karniadakis [93] have developed a second order accurate semi-Lagrangian method. Indeed, according to Oliveiro and Baptista [64], the simplest semi-Lagrangian scheme with linear interpolation is equivalent to the classical first-order upwinding scheme, which is excessively dissipative. Their second order accurate scheme shows an accuracy of the backward integration as well as the accuracy of the interpolation method. An implicit midpoint rule is used for backward integration. That means that the middle point in space between the arrival and the departure

point is considered. And also the mid time step $t^{n+1/2}$ is employed for the characteristic curve. The velocity at the mid time step is defined as a linear combination of the velocities at the two previous time steps t^n and t^{n-1} . The interpolation procedure to compute the departure point should then be at a second-order accuracy so that the whole semi-Lagrangian method described is itself second-order accurate.

Min and Gibou [55] used the aforementioned method in the context of computations of Navier-Stokes equations on an octree data structure. A similar semi-Lagrangian second-order backward scheme is developed in [36] for the computation of advection term on octrees. But with an adaptive time step, which requires some slight adaptation of previously described method.

Lentine et al. in [48] noticed that semi-Lagrangian schemes are based on characteristic tracing and interpolation, thus a fully conservative implementation is not necessarily guaranteed. They proposed a novel technique that applies a conservative limiter to the typical semi-Lagrangian interpolation step in order to guarantee that the amount of the conservative quantity does not increase during this advection.

The other common method consists in an Eulerian discretization of the advection term. Popinet in [74], used a first-order upwind scheme with finite differences. Olshanskii et al. [65], described a higher order method with the use of a third-order upwind discretization stencil with a finite difference scheme. The results obtained show that the method used is both stable and low-dissipative.

The solver Overflow, based on curvilinear or Cartesian grid systems employs discretizations of the convective term with fifth-order accurate central differences [19]. The Ellipsys3D code also discretized the convective term thanks to hybrid scheme combining a third-order QUICK scheme and fourth-order CDS scheme [27]. This higher-order accuracy on both solvers can be reached without too many efforts thanks to the block structured grid approach.

Chapter 2

Numerical Modelling of Flow around a Wind Turbine on Octree Grids

As described in the previous section, quadtree (in 2D) and octree (in 3D) allow a flexibility in mesh refinement and coarsening with the use of different levels of refinement on a unique grid and also have the advantages of regular meshes with a better convergence and higher resolution compared to unstructured grids. This octree-based discretization allows to refine on areas of high interest (on the body's wake, close to obstacles ...) and coarsen elsewhere, which is of particular interest for high-fidelity modelling of the flow around a wind turbine.

In this work, discretization and parallelization are based on a octree library named bitpit.¹ More precisely, bitpit is an open source modular library for scientific computing developed by Optimad Engineering Srl. The goal of bitpit is to ease the burden of writing scientific programs providing the common building blocks needed by every scientific application. In bitpit, only the PABLO (PARallel Balanced Linear Octree) module was used. This module allows to deal with parallel linear quadtrees/octrees, thanks to the Z-order space-filling curve and dynamical adaptive mesh refinement.

This module permitted an easy handling of parallel variables. Indeed, each cell surrounded by cells known by other processors (named ghost cells), has access to the so-defined ghost data (variables stored in the ghost cells). PABLO's routines allow immediate inclusion within a code without needing important adaptation. Another advantage of PABLO module is the fact that it manages linear octrees which allows a relevant gain of memory compared to other existing open-source libraries. It was also chosen in the present work that a cell has only access to the first cells close to it (let's say the first "layer" of surrounded cells). This implies in the context of parallelization that less communications and thus less computational time. The stencil used for the discretization is not wide and can only use the first neighbours surrounding each cell.

This chapter describes the implemented Navier-Stokes solver in the context of octree grids. The outline is, first, the explanation of the immersed boundary method for the computation of the equations. Then, the two-dimensional finite volume solver is described, followed by the extension to the three-dimensional case. Finally, the methods developed to deal with high-Reynolds number flows are explained.

¹Optimad, bitpit <https://github.com/optimad/bitpit>

1 Immersed Boundary Method

Immersed boundary method is included in the larger scope of the Fictitious domain method. The basic idea of fictitious domain method is to substitute a given problem posed on a domain D , with a new problem posed on a simple shaped domain Ω containing D . As previously introduced, the immersed boundary method allows to have a computational grid that is not body-fitted, which is simpler to implement. Indeed, as the interface between the fluid and the body is included in the cell meshes, simple grids like Cartesian or octree grids can be used, which implies a low amount of memory and an easy way to parallelize, compared to unstructured body-fitted meshes. Two approaches exist in the immersed boundary method: continuous forcing and discrete forcing. In the former, a force term is added to the continuous Navier-Stokes equations before discretization, whereas in the latter, the forcing is applied (explicitly or implicitly) to the discretized equations. The main advantage of the discrete forcing, such as in the ghost-cell method proposed by Mittal et al. [57], is the possibility to achieve a sharp representation of the fluid-solid interface, exactly as if the mesh were body fitted. On the other hand, its drawback is the problem of the so-called "fresh cells", a situation that is encountered dealing with moving interfaces: some solid cells might emerge into the fluid between one time step and another as a result of the boundary motion. Continuous forcing methods are not sharp, indeed the immersed boundary is diffused, leading to a loss of accuracy in the proximity of the body, but they allow to bypass the special treatment of fresh cells, as remarked by Bergmann et al. [9]. For the present work, a continuous forcing model was employed, which corresponds to the volume penalization method described in [3].

1.1 Penalized Equations

Using the notation of figure 1.8, the previously described Navier-Stokes equations (1.32) can be written using the penalization method, where the solid body is considered as a porous medium with very low permeability κ :

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \vec{\nabla})\mathbf{u} = -\frac{1}{\rho}\vec{\nabla}p + \nu\Delta\mathbf{u} + \frac{\chi_B}{\kappa}(\mathbf{u}_B - \mathbf{u}) & \text{in } \Omega, \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega, \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) & \text{in } \Omega, \\ \mathbf{u} = \mathbf{u}_f(\mathbf{x}, t) & \text{on } \Gamma_f, \end{cases} \quad (2.1)$$

where χ_B is the characteristic function:

$$\chi_B = \begin{cases} 0 & \text{in the fluid domain} \\ 1 & \text{in the solid domain} \end{cases} \quad (2.2)$$

It has been demonstrated by Angot et al. [3], that the system (2.1) converges to the system (1.32) when $\kappa \rightarrow 0$. More precisely, Nguyen van yen et al. [89] proved that the convergence of the model is in $\sqrt{\kappa}$.

In the present work, the solid body geometry is represented by Lagrangian markers. Basically, the body is divided in many sections and each section is formed by many points. Thanks to these markers, a signed distance function is computed for each cell of the grid. Indeed, for each point of our regular octree grid, the closest marker is searched in order to determine the distance from the surface. The Lagrangian markers have been organized, so that the sign of the distance function can be computed with the sign of the scalar product between the outward

surface normal and the vector that links the surface to the point in the domain. In order to simplify the notation, the signed distance function is called in present work the level-set. The relationship between the level-set function φ and the characteristic function is the following:

$$\chi_B = 1 - H(\varphi), \quad (2.3)$$

with $H(\varphi)$, the Heaviside function.

In the thesis, a first-order penalization is used in the solver developed, as a first approach. By the use of the level-set function gradients, the outward normal to the body is also computed at the fluid-solid interface:

$$\mathbf{n} = \left(\frac{\vec{\nabla}\varphi}{\|\vec{\nabla}\varphi\|} \right)_{\varphi=0}. \quad (2.4)$$

The penalized Navier-Stokes equations are then solved numerically using a fractional-step method as will be explained below.

1.2 Temporal Discretization

A semi-Lagrangian scheme has been considered in the present work which means that a Lagrangian perspective is used in the Eulerian framework. The left hand side term of the momentum equation in eq. (2.1) can be written:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \vec{\nabla})\mathbf{u} = \frac{D\mathbf{u}}{Dt}. \quad (2.5)$$

As we have a decoupling between the pressure and the velocity fields, nearly all numerical methods to solve the Navier-Stokes equations (2.1) use a fractional-step approach. This approach, referred to as the projection method, has been pioneered by Chorin [21] and Temam [87]. For a complete review of the different projection methods, the reader can refer to the article of Brown, Cortez and Minion [14]. In the present work, three fractional steps are solved with the computation of two intermediate velocity fields \mathbf{u}^* and \mathbf{u}^{**} . The subscript a means the arrival point, that fits with the mesh, whereas the subscript d corresponds to the departure point. The momentum equation can indeed be written:

$$\frac{\mathbf{u}_a^{n+1} - \mathbf{u}_d^n - \mathbf{u}_a^* + \mathbf{u}_a^* + \mathbf{u}_a^{**} - \mathbf{u}_a^{**}}{\Delta t} = -\frac{1}{\rho}\vec{\nabla}p^{n+1} + \frac{\chi_B}{\kappa}(\mathbf{u}_B^{n+1} - \mathbf{u}_a^{n+1}) + \nu\Delta\mathbf{u}_a^*. \quad (2.6)$$

An intermediate velocity \mathbf{u}^* is then computed as a prediction value of the velocity field \mathbf{u} . This first step, named the prediction step, consists in solving the following linear system:

$$\frac{\mathbf{u}_a^* - \mathbf{u}_d^n}{\Delta t} = \nu\Delta\mathbf{u}_a^*. \quad (2.7)$$

The diffusive term of the Navier-Stokes equations is implicitly computed. \mathbf{u}_d^n represents the velocity field interpolated at the root of the characteristic curve. This interpolation will be detailed in the next section. For easy handling, no initial guess of the pressure field is included for this prediction step. This algorithm is said to be non-incremental and is more precise, since there is no cumulative error on the pressure field. However, it is less stable than the incremental method, due to bigger correction imposed by the pressure. As far as we tested our code, no issue related to our non-incremental scheme occurred. Coming back to the prediction step, the computed predicted velocity field \mathbf{u}^* is not supposed to satisfy the divergence constraint. The correction step consists in solving a second fractional step:

$$\frac{\mathbf{u}_a^{**} - \mathbf{u}_a^*}{\Delta t} = -\frac{1}{\rho}\vec{\nabla}p^{n+1}. \quad (2.8)$$

Beforehand, in order to enforce the divergence constraint, the divergence operator is applied to eq. (2.8) and we obtain the following elliptic equation referred to as the projection step:

$$\nabla \cdot \mathbf{u}_a^* = \frac{\Delta t}{\rho} \Delta p^{n+1}. \quad (2.9)$$

Computing this equation allows to determine the pressure field. Then, we come back to eq. (2.8) (correction step) and the computation of the velocity field can be done. In order to apply the penalization term, a third fractional step is performed:

$$\mathbf{u}_a^{n+1} = \mathbf{u}_a^{**} + \Delta t \frac{\chi_B}{\kappa} (\mathbf{u}_B^{n+1} - \mathbf{u}_a^{n+1}). \quad (2.10)$$

1.3 Boundary Conditions

Boundary conditions for the pressure and the velocity field need to be imposed to correctly solve the problem. Considering the pressure, Neumann boundary conditions on all edges of the domain borders fit our application to study the flow past an obstacle. However, that leads to a singular linear system for the Poisson equation (2.9). It was thus chosen to remove the null space (i.e. the solution with a null average) to solve the system with Neumann boundary conditions. Indeed, the equations can be solved up to a constant. Another possibility would have been to fix the pressure at one edge of the border.

The boundary conditions for the velocity field will now be described. The inlet boundary condition will consist in imposing an incoming flow. So, Dirichlet boundary conditions are used at inlet corresponding to wind speed for our application to wind turbine modelling. On the contrary, at the outlet, Neumann boundary conditions are imposed to allow the flow to get out of the domain. The lateral borders should not let other flow getting in on the normal direction, therefore, a zero velocity Dirichlet condition is imposed for the normal component of the velocity field. For the tangential components (one component in two-dimensional case), it was chosen to impose a zero gradient condition. The initial condition of the velocity field is chosen to be the same as the inlet boundary condition. The Neumann boundary conditions imposed at the outlet constitute a first approach for simplicity, but numerical results can be improved by imposing non-reflecting boundary conditions instead. Especially, the size of the computational domain could be reduced.

Space and time discretizations of the Navier-Stokes solver are now described in detail for the two-dimensional case before extending the method to the three-dimensional case.

2 Two-Dimensional Finite Volume Solver

A Navier-Stokes solver with immersed boundaries in the framework of octree grids has been developed during this thesis. As explained in the previous section, a fractional-step method is used and the code solves equations (2.7), (2.9) and (2.8). The prediction and projection steps consist in solving linear systems. The open-source PETSc library is used to solve the linear systems [4]. Krylov subspace methods are employed, since these methods are known to be very successful in numerical linear algebra. Since we are using a discretization on octree grids with the Z-order space-filling curves, the variables in two neighbouring cells can be located far from each other in the matrices of the linear systems. Thus, two different iterative methods, that are part of Krylov subspace theory, have been tested for our Navier-Stokes solver: biconjugate gradient stabilized and generalized minimal residual methods. Both gave us the same results when compared with an analytical solution. This will be explained in more details in the next

chapter dedicated to the validation of the solver.

The finite volume method has been widely used in the present work for the octree-based discretization of the Navier-Stokes problem since it allows an easy formulation based on the evaluation of fluxes at adjacent cell surfaces. All the pressure and velocity variables are here only stored at cell-centres, since the PABLO module, used to handle parallel discretization, is more adapted to this configuration.

The prediction step (2.7) consists in solving two linear systems (for the two-dimensional case). A semi-Lagrangian scheme allows to compute the advection term. The left-hand side is constituted by a matrix constructed by the Laplace operator for the discretization of the implicit diffusive term multiplied by the vector of unknowns constituted with the predicted velocity field \mathbf{u}^* . The projection step (2.9) is an elliptic equation to obtain the pressure field. A linear system should thus be solved with a right-hand side corresponding to the discretization of the divergence of \mathbf{u}^* . The other side of the equation is similar as previously seen since a matrix is constructed that corresponds to the implicit discretization of the Laplace operator for pressure. Finally, the correction step (2.8) is quite simple since \mathbf{u}^* and p are known. The gradient of the pressure is computed by the use of the finite volume method as will be explained below and the velocity field \mathbf{u}^{n+1} can be obtained.

2.1 Numerical Scheme for the Divergence Operator

For the correction step, the computation of the pressure gradient is needed at the cell-centre in order to get the velocity field \mathbf{u}^{n+1} . In a similar way, the divergence of \mathbf{u}^* should also be computed in the projection step. The explanation of the method used will be described for the case of the divergence of some vector \mathbf{V} (totally similar to the divergence of \mathbf{u}^*) for simplicity. Exactly the same method is employed for the gradient computation. As explained above, a finite volume method is used, which means that we integrate $\nabla \cdot \mathbf{V}$ over each cell volume Ω_c . The divergence theorem and its corollaries are then employed to convert the volume integral into surface integrals:

$$\frac{1}{\Omega_c} \int_{\Omega_c} (\nabla \cdot \mathbf{V}) d\Omega_c = \frac{1}{\Omega_c} \oint_{\partial\Omega_c} (\mathbf{V} \cdot \mathbf{n}) dS, \quad (2.11)$$

where \mathbf{n} represents the outward normal of the cell. The right-hand side of eq. (2.11) is then evaluated as the sum of all fluxes at the edges of each cell. Thus, we can write for each cell i of the domain:

$$(\nabla \cdot \mathbf{V})_i = \frac{1}{\Delta\Omega_i} \sum_j \mathbf{V}_j \cdot \mathbf{n}_j \Delta S_j, \quad (2.12)$$

with j representing the edge index. For the two-dimensional case ΔS_j corresponds to the edge length and $\Delta\Omega_i$ to the area of i^{th} cell. \mathbf{V}_j needs to be evaluated for each cell at the edge centre. In order to optimize the operations, the developed solver computes only once the flux at each edge and uses this flux for the two surrounded cells sharing the edge.

When no level differences exist between the two cells sharing the edge (so when dealing with a Cartesian case), $\mathbf{V}_j \cdot \mathbf{n}_j$ is basically computed as the average of the two neighbouring cells where \mathbf{V} is known at cell-centre. In case of a level jump, for example between the cells c_4 and c_1 as can be seen in figure 2.1, $\mathbf{V}_j \cdot \mathbf{n}_j$ is evaluated with the vertex belonging to the edge:

$$\mathbf{V}_j \cdot \mathbf{n}_j = \frac{V_{v_1} + V_{v_2}}{2} \text{ at level jump.} \quad (2.13)$$

But V_{v_1} and V_{v_2} are unknowns at cell vertices, thus, their values need to be interpolated. All the neighbours surrounding each vertex are searched and linear or bilinear interpolation is then

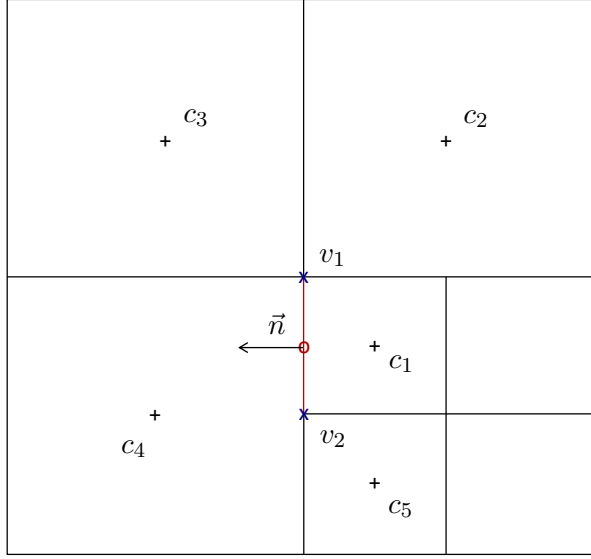


Figure 2.1: Example of mesh configuration in 2D with stencils

computed with the available neighbours. Indeed, only two different configurations exist: either the vertex v_1 has 4 neighbours which implies that vertex v_2 has 3 neighbours, or the opposite. A special treatment is done when the level jump has a vertex located at the border of the domain. In this case, the interpolation is computed at the edge centre with a stencil composed of the neighbours sharing the edge. In the present case with a level jump, we can write:

$$\begin{cases} V_{v_1} = \alpha_{v_1} + \beta_{v_1} \cdot x_{v_1} + \gamma_{v_1} \cdot y_{v_1} + \zeta_{v_1} \cdot x_{v_1} \cdot y_{v_1}, \\ V_{v_2} = \alpha_{v_2} + \beta_{v_2} \cdot x_{v_2} + \gamma_{v_2} \cdot y_{v_2}, \end{cases} \quad (2.14)$$

with x_{v_i} and y_{v_i} that represent the coordinates of the vertex v_i . Small linear systems are solved to obtain the unknown coefficients α_{v_i} , β_{v_i} , γ_{v_i} and ζ_{v_i} .

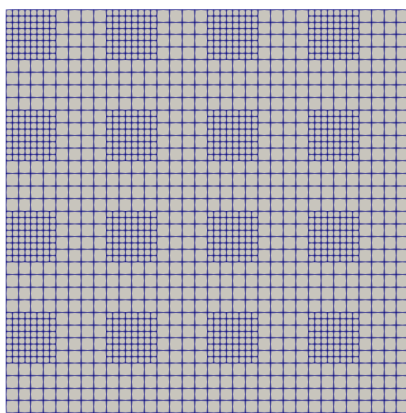
$$\begin{pmatrix} 1 & x_{c_1} & y_{c_1} & x_{c_1} \cdot y_{c_1} \\ 1 & x_{c_2} & y_{c_2} & x_{c_2} \cdot y_{c_2} \\ 1 & x_{c_3} & y_{c_3} & x_{c_3} \cdot y_{c_3} \\ 1 & x_{c_4} & y_{c_4} & x_{c_4} \cdot y_{c_4} \end{pmatrix} \begin{pmatrix} \alpha_{v_1} \\ \beta_{v_1} \\ \gamma_{v_1} \\ \zeta_{v_1} \end{pmatrix} = \begin{pmatrix} V_{c_1} \\ V_{c_2} \\ V_{c_3} \\ V_{c_4} \end{pmatrix}, \quad (2.15)$$

$$\begin{pmatrix} 1 & x_{c_1} & y_{c_1} \\ 1 & x_{c_4} & y_{c_4} \\ 1 & x_{c_5} & y_{c_5} \end{pmatrix} \begin{pmatrix} \alpha_{v_2} \\ \beta_{v_2} \\ \gamma_{v_2} \end{pmatrix} = \begin{pmatrix} V_{c_1} \\ V_{c_4} \\ V_{c_5} \end{pmatrix}. \quad (2.16)$$

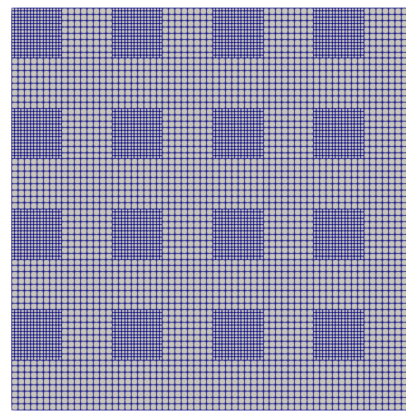
In order to proof the consistency and to evaluate the order of this method, an analysis has been carried out. The function $f(x, y) = \sin(x^2 + y^2)$ has been chosen in the two-dimensional square domain $\Omega = [-0.5, 0.5] \times [-0.5, 0.5]$. Exact Dirichlet boundary conditions have been imposed at the cell centres along the border of the domain. The computation of $\nabla \cdot f$ has so been performed by the use of the previously described solver and has been compared to the exact solution of the divergence of f . This analysis has been made for a Cartesian grid as a first validation and the results can be seen in table 2.1. As expected, an order of convergence of 2 is obtained for both the L_2 and the L_∞ norms. The same study has then been performed with a quadtree grid. This grid has been chosen with a high number of level jumps in order to prove the consistency

Tree level	L_∞	Order	L_2	Order
5	$6.972 \cdot 10^{-4}$		$1.680 \cdot 10^{-4}$	
6	$2.007 \cdot 10^{-4}$	1.80	$4.688 \cdot 10^{-4}$	1.84
7	$5.366 \cdot 10^{-5}$	1.90	$1.238 \cdot 10^{-5}$	1.92
8	$1.387 \cdot 10^{-5}$	1.95	$3.179 \cdot 10^{-6}$	1.96
9	$3.523 \cdot 10^{-6}$	1.98	$8.057 \cdot 10^{-7}$	1.98
10	$8.88 \cdot 10^{-7}$	1.99	$2.028 \cdot 10^{-7}$	1.99

Table 2.1: Norm of the error and order of the divergence operator for 2D Cartesian grid



(a) Tree level 6



(b) Tree level 7

Figure 2.2: Examples of quadtree grids used for validation

of the solver developed for every kind of quadtree grid. Examples of such grids can be seen in figure 2.2.

The results can be seen in table 2.2. Thus, the L_∞ norm is consistent with an order of 1. L_2 norm has a 1.5 order accuracy. We even get the second-order accuracy when looking at the L_1 norm of the error. This divergence operator scheme is so consistent even with highly irregular quadtree grid configuration.

Tree level - Number of cells	L_∞	Order	L_2	Order
5 - 1792	$3.124 \cdot 10^{-2}$		$4.443 \cdot 10^{-3}$	
6 - 7168	$1.562 \cdot 10^{-2}$	1.00	$1.540 \cdot 10^{-3}$	1.53
7 - 28672	$7.810 \cdot 10^{-3}$	1.00	$5.391 \cdot 10^{-4}$	1.51
8 - 114688	$3.905 \cdot 10^{-3}$	1.00	$1.900 \cdot 10^{-4}$	1.51
9 - 458752	$1.952 \cdot 10^{-3}$	1.00	$6.687 \cdot 10^{-5}$	1.50
10 - 1835008	$9.762 \cdot 10^{-4}$	1.00	$2.361 \cdot 10^{-5}$	1.50

Table 2.2: Norm of the error and order of the divergence operator for 2D quadtree grid

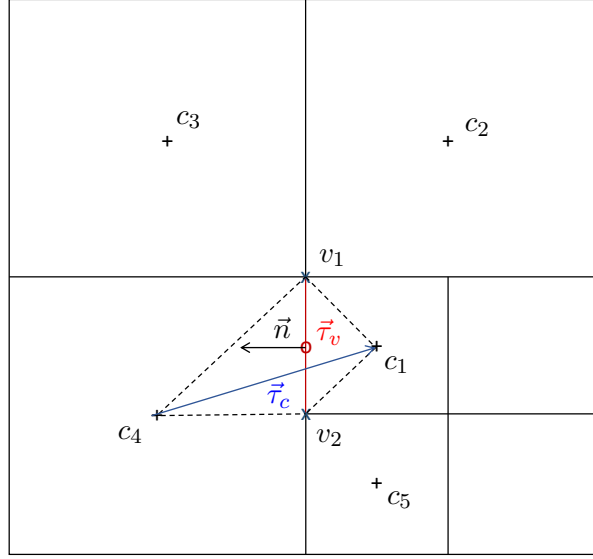


Figure 2.3: Example of mesh configuration in 2D with stencils for Laplacian operator

2.2 Numerical Scheme for the Laplace Operator

The linear systems that should be solved for both the prediction step (2.7) and the Poisson equation (2.9) imply the discretization of a Laplace operator. To explain the method developed in the present work, the Laplacian of a variable Φ will be described for easier writing. The reader is reminded that exactly the same method is used for the discretization of the pressure Laplacian in (2.9) and of the predicted velocity field Laplacian in (2.7). A finite volume method is employed and we integrate $\nabla^2\Phi$ on each cell volume. The divergence theorem is then applied:

$$\frac{1}{\Omega_c} \int_{\Omega_c} \nabla \cdot (\nabla\Phi) d\Omega_c = \frac{1}{\Omega_c} \oint_{\partial\Omega_c} (\nabla\Phi \cdot \mathbf{n}) dS. \quad (2.17)$$

The right-hand side of eq. (2.17) is then evaluated as the sum of fluxes for each edge of cells, which can be written for each cell i of the domain:

$$(\nabla^2\Phi)_i = \frac{1}{\Delta\Omega_i} \sum_j (\nabla\Phi)_j \cdot \mathbf{n}_j \Delta S_j. \quad (2.18)$$

The problem comes down to the evaluation of the gradient of Φ at each cell edge. Discrete Duality Finite Volume (DDFV) schemes have been investigated. The aim of such methods is therefore to provide a discrete reconstruction of gradients. The idea of these methods is to integrate the equations on both a primal and a dual mesh. Coudière et al. [24] developed the reconstruction of discrete gradients on a diamond mesh and this method has also been implemented on the present work since the mesh configuration with quadtrees could fit with diamond dual mesh. An example of grid can be seen in figure 2.3. When having no level difference between the two cells sharing the edge, for example, when computing the gradient on the edge shared by c_3 and c_2 , the gradient can simply be computed:

$$\left(\vec{\nabla}\Phi\right)_j \cdot \mathbf{n}_j = \frac{\Phi_{c_3} - \Phi_{c_2}}{\Delta l} \cdot \mathbf{n}_j \text{ for a Cartesian configuration} \quad (2.19)$$

with Δl corresponding to the length of the edge.

When dealing with a quadtree configuration, a diamond cell is used in order to reconstruct the discrete gradient of this dual mesh. The following system should so be solved:

$$\begin{cases} \vec{\nabla}\Phi \cdot \vec{\tau}_c = \frac{\Phi_{c_1} - \Phi_{c_4}}{\Delta c}, \\ \vec{\nabla}\Phi \cdot \vec{\tau}_v = \frac{\Phi_{v_1} - \Phi_{v_2}}{\Delta l}, \end{cases} \quad (2.20)$$

with Δc being the length between the centres of cells c_1 and c_4 . It can then be written that:

$$\vec{\nabla}\Phi = \nabla\Phi_1 \cdot \vec{e}_1 + \nabla\Phi_2 \cdot \vec{e}_2 = \nabla\Phi_1 \cdot \vec{n} + (\vec{\nabla}\Phi \cdot \vec{\tau}_v) \cdot \vec{e}_2, \quad (2.21)$$

with \vec{e}_i that represents the two unit vectors in x or y direction. When dealing with quadtree grids, $\vec{\tau}_v$ is either oriented in x or y direction and \vec{n} in the other direction. In the present case we chose to write $\vec{e}_1 = \vec{n}$. That implies also that:

$$\vec{\nabla}\Phi \cdot \vec{n} = \nabla\Phi_1.$$

Thanks to the use of (2.20) and (2.21), we finally obtain:

$$(\nabla\Phi)_j \cdot \mathbf{n}_j = \frac{1}{\vec{\tau}_c \cdot \mathbf{n}_j} \left(\frac{\Phi_{c_1} - \Phi_{c_4}}{\Delta c} - \frac{\Phi_{v_1} - \Phi_{v_2}}{\Delta l} \cdot \vec{\tau}_v \cdot \vec{\tau}_c \right) \text{ at level jump.} \quad (2.22)$$

In the present work, it was chosen to store only variables at the cell-centre. Moreover, we are dealing with an implicit variable. So, Φ_{v_1} and Φ_{v_2} should be evaluated from cell-centred variables. In order to use the same stencil as the previously described divergence operator, linear or bilinear interpolations were also employed for the present case. The same linear systems as eqs (2.15) and (2.16) can thus be written for the variable Φ but they can't be directly solved since we are dealing with implicit variables. The unknown coefficients are expressed from Φ variable and the matrix corresponding to the global linear system of the prediction or projection steps can then be constructed.

This scheme for discretization of the Laplace operator has then been analysed to evaluate its consistency and accuracy. The same analytical function, same square domain and the same grid configurations as in the previous section have been used and the equation $\nabla^2 f = s$ is solved with $s(x, y)$ being the analytical Laplacian of $f(x, y) = \sin(x^2 + y^2)$. Exact Dirichlet boundary conditions have been imposed at the cell centres along the border of the domain. A first analysis has been carried out with a Cartesian grid to proof the consistency of the method. The results can be seen on table 2.3. We reach an order of convergence of 2 with both the L_2 and L_∞ norms, which is the expected result.

The same study has also been made with the quadtree grid configuration previously used (figure 2.2). The results obtained can be observed on table 2.4. This analysis shows a second-order accuracy with both the L_2 and L_∞ norms for the Laplace operator discretization even on a highly irregular grid configuration. However, the level of errors are bigger compared to the Cartesian configuration. That is explained with the fact that no interpolation is needed for the Cartesian grid compared to quadtree. The results obtained with the L_1 norm are not presented here but similar results as for the L_2 norm are noticed. Coudière et al. [24] explained however that on general meshes, an order of convergence of 1 is reached for the Diamond's method. Thus, we might have a superconvergence phenomenon with the Laplace operator discretization. An example of the error distribution on a refined grid is presented in figure 2.4. The exact solution is symmetric in the computational domain but the error distribution is not. This can be explained by the fact that the grid configuration is not symmetric in that domain. The Laplace operator scheme is thus consistent and shows a second-order accuracy even with highly irregular grid configuration.

Tree level	L_∞	Order	L_2	Order
5	$2.837 \cdot 10^{-5}$		$1.989 \cdot 10^{-5}$	
6	$7.574 \cdot 10^{-6}$	1.91	$5.394 \cdot 10^{-6}$	1.88
7	$1.973 \cdot 10^{-6}$	1.94	$1.410 \cdot 10^{-6}$	1.94
8	$5.095 \cdot 10^{-7}$	1.95	$3.564 \cdot 10^{-7}$	1.98
9	$1.252 \cdot 10^{-7}$	1.98	$9.033 \cdot 10^{-8}$	1.98
10	$3.150 \cdot 10^{-8}$	1.99	$2.272 \cdot 10^{-8}$	1.99

Table 2.3: Norm of the error and order of the Laplace operator for 2D Cartesian grid

Tree level - Number of cells	L_∞	Order	L_2	Order
5 - 1792	$1.028 \cdot 10^{-3}$		$4.254 \cdot 10^{-4}$	
6 - 7168	$2.522 \cdot 10^{-4}$	2.03	$1.085 \cdot 10^{-4}$	1.97
7 - 28672	$6.272 \cdot 10^{-5}$	1.93	$2.750 \cdot 10^{-5}$	1.98
8 - 114688	$1.566 \cdot 10^{-5}$	1.97	$6.928 \cdot 10^{-6}$	1.99
9 - 458752	$3.917 \cdot 10^{-6}$	1.97	$1.739 \cdot 10^{-6}$	1.99
10 - 1835008	$9.658 \cdot 10^{-7}$	2.01	$4.306 \cdot 10^{-7}$	2.01

Table 2.4: Norm of the error and order of the Laplace operator for 2D quadtree grid

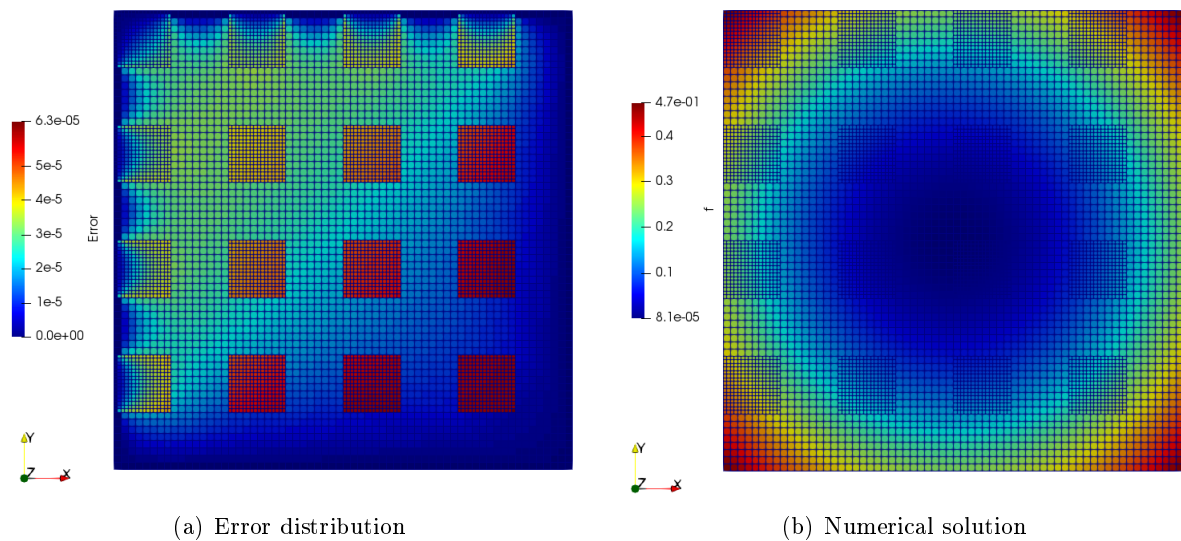


Figure 2.4: Example of error distribution between a Laplace operator discretization and an analytical expression and numerical solution with a quadtree grid configuration corresponding to tree level 7

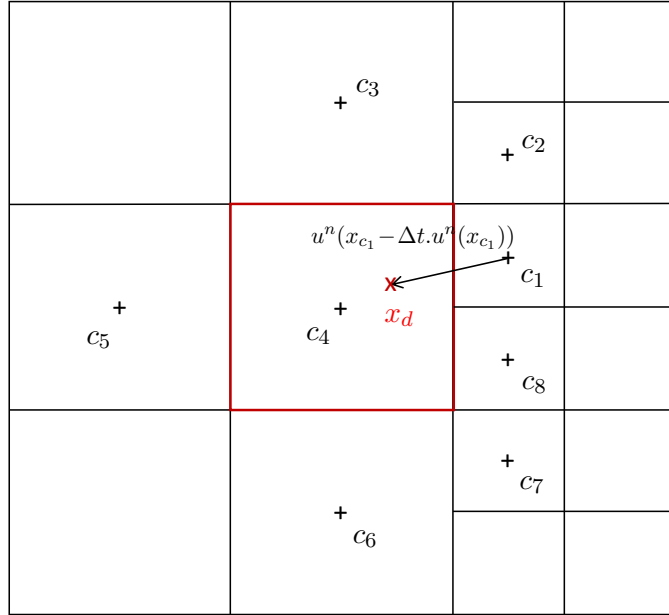


Figure 2.5: Example of mesh configuration in 2D with stencils for semi-Lagrangian scheme

2.3 Semi-Lagrangian Scheme for the Advection Term

The Lagrangian description of a flow field requires to follow the trajectories of the flow particles whereas the Eulerian description fixes a particular point in space and considers the time evolution of variables at this point. In this work, a semi-Lagrangian description of the flow is adopted. It avoids the direct computation of the convective terms which is not straightforward with an Eulerian scheme in the presence of non-conforming meshes as quadtree grids.

The semi-Lagrangian scheme consists in computing the predicted velocity field by taking into account the diffusive term and the velocity computed at the previous time step by following the trajectory of the field. So, a semi-Lagrangian scheme uses an Eulerian framework but the discrete equations come from the Lagrangian perspective. Thus, in the prediction step, the characteristic curve is computed for each cell i as:

$$(\mathbf{x}_d^n)_i = \mathbf{x}_i - \Delta t \mathbf{u}_i^n, \quad (2.23)$$

with \mathbf{x}_i being the position of i^{th} cell centre, \mathbf{u}_i^n corresponds to the velocity field at cell centre and $(\mathbf{x}_d^n)_i$ represents the position at the root of the characteristic. This quantity needs to be interpolated since there is no reason that this position coincides with the grid. As this characteristic curve uses only a single time-step backward integration, the time discretization of the Navier-Stokes solver will be only first-order accurate in time.

As previously explained, by the use of the bitpit library, we have a constraint in parallel computing, which is the fact that each cell knows only its first neighbouring cells. Indeed, the implicit computation of the diffusive term in the Navier-Stokes equations (2.1) prevents us to be constraint by the stability condition and allows us to use large time steps. Moreover the semi-Lagrangian scheme is unconditionally stable. However, the parallel computing imposes that the root of the characteristic stands at least in a ghost cell in order to interpolate the velocity.

The method developed for the implementation of the semi-Lagrangian scheme in the solver will now be described. For more simplicity the description will be done with the grid configuration

Tree level - Number of cells	L_∞	Order	L_2	Order
5 - 1792	$1.459 \cdot 10^{-3}$		$3.875 \cdot 10^{-4}$	
6 - 7168	$7.025 \cdot 10^{-4}$	1.05	$9.856 \cdot 10^{-5}$	1.98
7 - 28672	$3.453 \cdot 10^{-4}$	1.02	$2.486 \cdot 10^{-5}$	1.99
8 - 114688	$1.707 \cdot 10^{-4}$	1.02	$6.241 \cdot 10^{-6}$	1.99
9 - 458752	$8.488 \cdot 10^{-5}$	1.01	$1.564 \cdot 10^{-6}$	2.00
10 - 1835008	$4.232 \cdot 10^{-5}$	1.00	$3.914 \cdot 10^{-7}$	2.00

Table 2.5: Norm of the error and order of the semi-Lagrangian for 2D quadtree grid

shown in figure 2.5. The idea to interpolate the velocity field at the characteristic root is to have a loop on each cell, then obtaining the index of the cell where to interpolate and finally having a plane reconstruction of the velocity field in that cell. In the present configuration, our loop is currently on cell c_1 and the characteristic curve root is located on cell c_4 . So, a plane equation is used to rebuild the velocity field in the cell:

$$\mathbf{u}^n \cdot \mathbf{e}_i = \alpha + \beta x + \gamma y, \quad (2.24)$$

with $\mathbf{u} \cdot \mathbf{e}_i$ representing the two components of the velocity field. It will be simplified by looking at one component. The three parameters are then found thanks to the use of the velocity at the cell-centre (which is known) and with the gradient computation at cell-centre. We indeed have:

$$\begin{cases} \left(\frac{\partial u}{\partial x} \right)_{c_4} = \beta, \\ \left(\frac{\partial u}{\partial y} \right)_{c_4} = \gamma, \end{cases} \quad (2.25)$$

$$\text{and} \quad u_{c_4}^n = \alpha + \left(\frac{\partial u}{\partial x} \right)_{c_4} \cdot x_{c_4} + \left(\frac{\partial u}{\partial y} \right)_{c_4} \cdot y_{c_4}. \quad (2.26)$$

Finally, we obtain:

$$u^n(\mathbf{x}) = u_{c_4}^n - \left(\frac{\partial u}{\partial x} \right)_{c_4} \cdot x_{c_4} - \left(\frac{\partial u}{\partial y} \right)_{c_4} \cdot y_{c_4} + \left(\frac{\partial u}{\partial x} \right)_{c_4} \cdot x + \left(\frac{\partial u}{\partial y} \right)_{c_4} \cdot y. \quad (2.27)$$

The computation of the gradients at each cell-centre is done with a finite volume method. Exactly the same method as previously described for the divergence operator is used. This computation is done previously to the loop for semi-Lagrangian and the gradient values are stored. This scheme to treat the advection term in the Navier-Stokes solver has been tested with an analytical function which is transported in a quadtree grid configuration. The aim is to evaluate the interpolation error at the root of the characteristic curve. The same meshes have been used and the analytical function is $g(x, y, t) = f(x, y) \cdot e^t$ with t being the time and $f(x, y)$ the previously described sinus function. The results of errors and order of convergence for the space discretization of the semi-Lagrangian can be observed on table 2.5. A first-order accuracy is reached for the L_∞ norm for a highly non-conformal grid and we reach even a second-order accuracy with other norms. Thus, we validated all the operators needed to solve the whole Navier-Stokes equations for the two-dimensional case. In the next chapter, the tool has been widely tested with several two-dimensional benchmark cases existing in the literature.

3 Extension of the Methods to the Three-Dimensional Case

The methods previously described have been chosen in order that the extension to the three-dimensional case can be straight-forward. In the same way as for the two-dimensional solver, the different operators have been developed separately and validated thanks to the comparison with an analytical function before putting all together to solve the whole Navier-Stokes equations (2.1). The main change done in the three-dimensional solver compared to the two-dimensional one is the way we deal with interpolations. Indeed, polynomial interpolations in 3D can be painstaking because of the large number of possible configurations that can be created depending of the number of neighbours. Therefore, Gaussian Radial Basis Function (RBF) interpolation has been chosen:

$$f(x) = \sum_i w_i e^{\epsilon r_i^2}, \quad (2.28)$$

where w_i are the weights of the interpolation, $r_i = \|x - x_i\|$ and ϵ is a scaling parameter proportional to $\frac{1}{\Delta l_i}$ (inverse of the cell sizes). This method is second-order accurate and is independent on the grid configuration, which implies an easy implementation. One of the drawbacks is the fact that it relies only on the distance between the desired point and the interpolation points. The position is not taken into account. However, our tests showed satisfying results with this method.

3.1 Numerical Scheme for the Divergence Operator

The method used to discretize the divergence operator in 3D is the same as for the two-dimensional case. That means that a finite volume method is used (see eq. (2.11)). We can write for each cell i of the domain:

$$(\nabla \cdot \mathbf{V})_i = \frac{1}{\Delta \Omega_i} \sum_j \mathbf{V}_j \cdot \mathbf{n}_j \Delta S_j, \quad (2.29)$$

with j representing the face index. For the three-dimensional case ΔS_j corresponds to the face area and $\Delta \Omega_i$ to the volume of the i^{th} cell. \mathbf{V}_j needs to be evaluated at face centre for each cell. When no level difference occurs between the two cells surrounding the face, $\mathbf{V}_j \cdot \mathbf{n}_j$ is basically computed as the average of the neighbouring cells where \mathbf{V} is known at the cell-centre. In case of octree configuration, the first idea tested was based on the two-dimensional method (see eq. (2.13)) that consists in, firstly, interpolating on the vertices composing the face (with RBF-based interpolations) and secondly, coming back to face-centre interpolation. However, when tested, this method showed inaccurate results. This was perhaps due to the fact that RBF interpolations don't take into account the position of the point used in the stencil. So, the method developed in the solver is a direct interpolation of the variable at face-centre with a single RBF interpolation. The stencil used is quite wide since all the neighbours of each vertex constituting the face are used. A criterion is however added to retrieve from the stencil the neighbours that are too far from the face-centre ($> 1.5\Delta l$ with Δl the length of the face). It was indeed noticed with the use of RBF interpolations that a stencil too large implied less accurate results.

This operator has been analysed in order to proof the consistency and to evaluate the order of convergence with irregular octrees. The mesh configurations used can be seen in figure 2.6. The function $f(x, y, z) = \sin(x^2 + y^2 + z^2)$ has been chosen in the three-dimensional domain $\Omega = [-0.5, 0.5] \times [-0.5, 0.5] \times [-0.5, 0.5]$. Exact Dirichlet boundary conditions have been imposed at the cell centres along the borders of the domain. On a Cartesian grid, the error norms show second-order accuracy. The results obtained with the octree grids can be seen in table 2.6. It

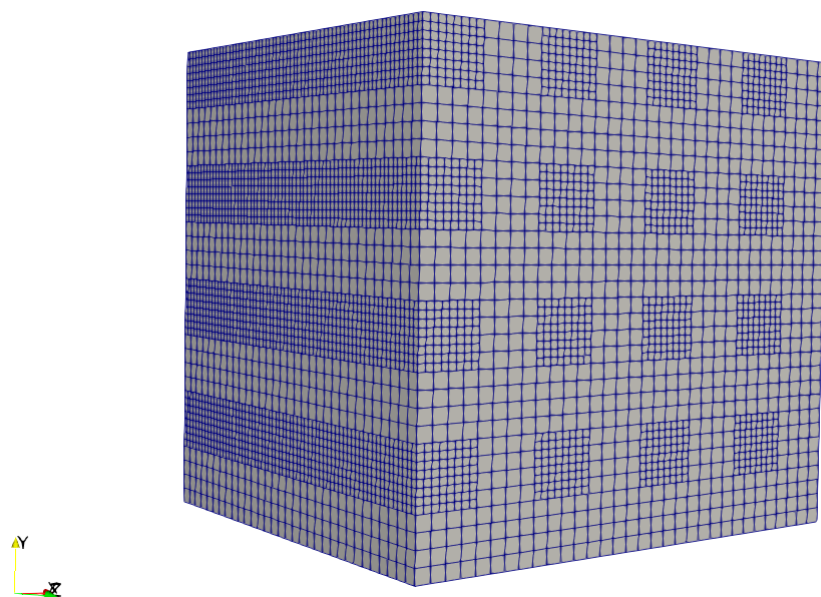


Figure 2.6: Example of octree grid configuration used for validation of the solver

Tree level - Number of cells	L_∞	Order	L_2	Order
5 - 11264	$1.627 \cdot 10^{-2}$		$2.817 \cdot 10^{-3}$	
6 - 90112	$7.965 \cdot 10^{-3}$	1.03	$1.120 \cdot 10^{-3}$	1.33
7 - 720896	$3.974 \cdot 10^{-3}$	1.00	$4.149 \cdot 10^{-4}$	1.43
8 - 5767168	$2.020 \cdot 10^{-3}$	1.02	$1.498 \cdot 10^{-4}$	1.47
9 - 46137344	$1.054 \cdot 10^{-3}$	1.01	$5.357 \cdot 10^{-5}$	1.48

Table 2.6: Norm of the error and order of the divergence operator for 3D octree grid

can be observed that the L_∞ norm of the error is first-order accurate. L_2 norm has a 1.5 order of accuracy and a second-order accuracy is obtained for the L_1 norm. This allows to prove the consistency of the divergence operator scheme with highly irregular octree grid configurations.

3.2 Numerical Scheme for the Laplace Operator

The linear systems that should be solved for both the prediction step (2.7) and the elliptic equation (2.9) imply the discretization of a Laplace operator. An extension of the method used for the two-dimensional case has been done and a finite volume scheme is used (see eqs. (2.17) and (2.18)). The gradients need so to be evaluated at each face centre. For a Cartesian configuration, exactly the same method as in 2D case is used with the use of the two surrounding neighbours (see eq (2.19)). When dealing with an octree configuration, a diamond cell is used to reconstruct the discrete gradient of this dual mesh at the centre of each face. An example of

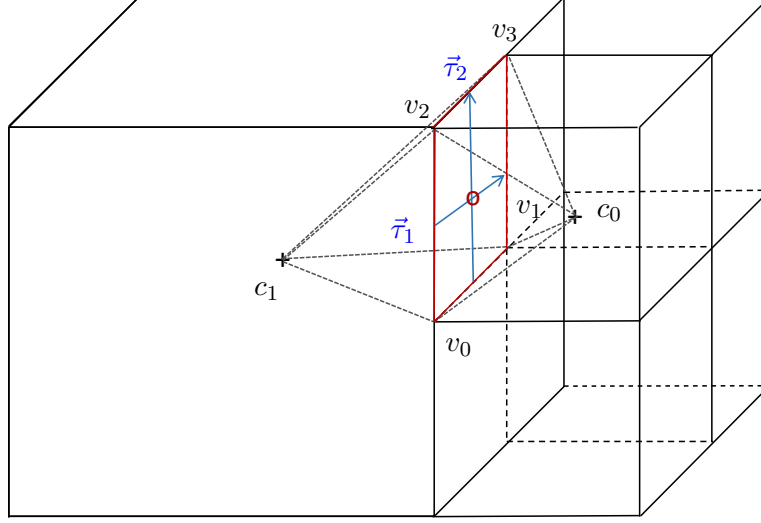


Figure 2.7: Example of mesh configuration in 3D with stencils for Laplace operator

the grid can be seen in figure 2.7. The following system should be solved:

$$\begin{cases} \vec{\nabla}\Phi \cdot \vec{\tau}_c = \frac{\Phi_{c_0} - \Phi_{c_1}}{\Delta c}, \\ \vec{\nabla}\Phi \cdot \vec{\tau}_1 = \frac{1}{2} \left(\vec{\nabla}\Phi \cdot \vec{\tau}_{v_3v_1} + \vec{\nabla}\Phi \cdot \vec{\tau}_{v_2v_0} \right) = \frac{1}{2} \left(\frac{\Phi_{v_3} - \Phi_{v_1}}{\Delta l} + \frac{\Phi_{v_2} - \Phi_{v_0}}{\Delta l} \right), \\ \vec{\nabla}\Phi \cdot \vec{\tau}_2 = \frac{1}{2} \left(\vec{\nabla}\Phi \cdot \vec{\tau}_{v_3v_2} + \vec{\nabla}\Phi \cdot \vec{\tau}_{v_1v_0} \right) = \frac{1}{2} \left(\frac{\Phi_{v_3} - \Phi_{v_2}}{\Delta l} + \frac{\Phi_{v_1} - \Phi_{v_0}}{\Delta l} \right), \end{cases} \quad (2.30)$$

with Δc being the length between the cell centres c_0 and c_1 , and Δl the size of the face.

Using the same methodology as for the two-dimensional case, we can finally write:

$$(\nabla\Phi)_j \cdot \mathbf{n}_j = \frac{1}{\vec{\tau}_c \cdot \mathbf{n}_j} \left[\frac{\Phi_{c_0} - \Phi_{c_1}}{\Delta c} - \frac{\vec{\tau}_1 \cdot \vec{\tau}_c}{2\Delta l} (\Phi_{v_3} - \Phi_{v_1} + \Phi_{v_2} - \Phi_{v_0}) - \frac{\vec{\tau}_2 \cdot \vec{\tau}_c}{2\Delta l} (\Phi_{v_3} - \Phi_{v_2} + \Phi_{v_1} - \Phi_{v_0}) \right]. \quad (2.31)$$

The variables are only known at cell centres, so, the variables at nodes should be interpolated to apply this scheme. All the neighbours of each vertex are used in the stencil and a RBF interpolation is performed at each node with the method described previously. The consistency and accuracy of this scheme in 3D has also been evaluated by solving a heat equation with an analytical function: $\nabla^2 f = s$ with $s(x, y, z)$ being the analytical Laplacian of the previously used function $f(x, y, z) = \sin(x^2 + y^2 + z^2)$. Exact Dirichlet boundary conditions are imposed at the borders of the domain Ω . The results on error can be observed on table 2.7. The orders of convergence with both the L_2 and L_∞ norms are second-order accurate. This allows to validate our Laplacian operator even on highly irregular octree grids.

3.3 Semi-Lagrangian Method

For the three-dimensional solver, a semi-Lagrangian scheme has been chosen to deal with the advection term, as for the 2D code (see eq. (2.23)). The velocity needs to be interpolated at the root of the characteristic curve since there is no reason that this point coincides with the grid. For the interpolation it was chosen to perform RBF interpolations as it seemed to be the easier

Tree level - Number of cells	L_∞	Order	L_2	Order
5 - 11264	$5.764 \cdot 10^{-3}$		$5.186 \cdot 10^{-3}$	
6 - 90112	$3.415 \cdot 10^{-4}$	2.03	$1.339 \cdot 10^{-4}$	1.95
7 - 720896	$8.492 \cdot 10^{-5}$	2.01	$3.409 \cdot 10^{-5}$	1.97
8 - 5767168	$2.121 \cdot 10^{-5}$	2.00	$8.608 \cdot 10^{-5}$	1.99
9 - 46137344	$5.312 \cdot 10^{-6}$	2.00	$2.164 \cdot 10^{-6}$	1.99

Table 2.7: Norm of the error and order of the Laplace operator for 3D octree grid

Tree level - Number of cells	L_∞	Order	L_2	Order
5 - 11264	$2.681 \cdot 10^{-3}$		$1.599 \cdot 10^{-3}$	
6 - 90112	$6.686 \cdot 10^{-4}$	2.00	$4.702 \cdot 10^{-4}$	1.77
7 - 720896	$1.671 \cdot 10^{-4}$	2.00	$1.255 \cdot 10^{-4}$	1.91
8 - 5767168	$4.178 \cdot 10^{-5}$	2.00	$3.232 \cdot 10^{-5}$	1.96
9 - 46137344	$5.312 \cdot 10^{-6}$	2.00	$2.164 \cdot 10^{-6}$	1.98

Table 2.8: Norm of the error and order of the semi-Lagrangian scheme for 3D octree grid

method. Indeed, the closest vertex to the root of the characteristic curve is obtained and all the cells surrounding this vertex are used in the stencil. This scheme has then be tested with an analytical function which is transported in a octree mesh configuration. The same meshes are used and the analytical function is $g(x, y, z, t) = f(x, y, z) \cdot e^t$ with t being the time and $f(x, y, z)$ the previously described sinusoidal function. The results of errors and order of convergence of the space discretization can be seen on table 2.8. A second-order accuracy is reached for all norms. A CFL condition of 0.7 has been chosen since it allowed to obtain a good compromise between accuracy and future time computing.

The time discretization of our 3D solver is however only first-order accurate considering the semi-Lagrangian scheme used as explained by Falcone and Ferretti [30]. Finally, all the operators needed for solving the Navier-Stokes in 3D have been validated. The whole solver has then been tested with several benchmark cases existing in the literature as will be explained in the next chapter.

4 Wall Functions for High-Reynolds Number Flow

Characteristic Reynolds numbers of a wind turbine flow in operative conditions are of the order of millions [18]. That means that the fluid motion is characterized by the presence of turbulent boundary layers and multiscale vortical structures in the wake. Thus, performing a direct numerical simulation won't be possible. Given the highly unsteady nature of the flow that will be simulated, a promising approach is the use of Large Eddy Simulation (LES) that permits to resolve only the largest structures of the turbulence. Indeed they carry the largest part of the energy and give the biggest contribution to the transport phenomena. The smallest structures, that are associated with turbulent energy dissipation are modelled using a sub-grid model. That method will now be explained.

4.1 Large Eddy Simulations

Based on the book of S. Pope [73], the mathematical formulation of LES is based on a low pass filter operator (noted G), so that filtered variables are defined as:

$$\bar{\psi} = \int_{x_1} \int_{x_2} \int_{x_3} G(\mathbf{r}, \mathbf{x}) \psi(\mathbf{x}) dx_1 dx_2 dx_3. \quad (2.32)$$

Applying (2.32) to Navier-Stokes equations (1.32) allows to obtain filtered equations. Considering a spatially uniform filter, filtering operation commutes with differentiation. The continuity equation is linear, therefore filtering the equation is equivalent to apply the divergence operator to the filtered variables:

$$\frac{\partial \bar{u}_i}{\partial x_i} = \frac{\partial \bar{u}_i}{\partial x_i} = 0, \quad (2.33)$$

where Einstein notation is applied. The filtered momentum equation is:

$$\frac{d\bar{u}_j}{dt} + \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_i} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_j} + \nu \frac{\partial^2 \bar{u}_j}{\partial x_i \partial x_i}. \quad (2.34)$$

Due to the non-linearity of the convective term, the form of the equation is different from the non-filtered one. Indeed, the filtered product $\overline{U_i U_j}$ is not equal to the product of filtered velocities. The residual-stress tensor is defined as the difference of the two elements:

$$\tau_{ij} = \overline{u_i u_j} - \bar{u}_i \bar{u}_j, \quad (2.35)$$

so that equation (2.34) can be rewritten as:

$$\frac{d\bar{u}_j}{dt} + \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_i} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_j} + \nu \frac{\partial^2 \bar{u}_j}{\partial x_i \partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j}. \quad (2.36)$$

From a physical point of view, the residual-stress tensor can be interpreted as the exchange of momentum, at the filtered scale, exerted by the sub-grid turbulent structures. Let k_r be the residual kinetic energy:

$$k_r = \frac{1}{2} \tau_{ii}. \quad (2.37)$$

Then, the residual-stress tensor can be decomposed into an isotropic component and an anisotropic one:

$$\tau_{ij}^r = \tau_{ij} - \frac{2}{3} k_r \delta_{ij}, \quad (2.38)$$

where δ_{ij} represents the Kronecker delta. The isotropic component is included in the modified filtered pressure term:

$$p^* = \bar{p} + \frac{2}{3} \rho k_r. \quad (2.39)$$

The filtered momentum equation can now be rewritten into its final form:

$$\frac{\partial \bar{u}_j}{\partial t} + \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_i} = -\frac{1}{\rho} \frac{\partial p^*}{\partial x_j} + \nu \frac{\partial^2 \bar{u}_j}{\partial x_i \partial x_i} + \frac{\partial \tau_{ij}^r}{\partial x_j}. \quad (2.40)$$

In order to solve the equation, the problem needs to be closed using what is called a sub-grid model, which express the anisotropic residual-stress tensor as a function of the filtered variables. Most of the sub-grid models rely on the Boussinesq hypothesis, which states that τ_{ij}^r is proportional to the filtered rate of strain tensor:

$$\tau_{ij}^r = \nu_e \overline{S_{ij}} = \nu_e \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right), \quad (2.41)$$

Chapter 2. Numerical Modelling of Flow around a Wind Turbine on Octree Grids

where ν_e is the eddy viscosity. The first and the simplest sub-grid model is the Smagorinsky model [79]:

$$\nu_e = (C_S \Delta)^2 \bar{S}, \quad (2.42)$$

where C_S is the Smagorinsky constant, whose classical value is 0.17, Δ is the length of the filter, with:

$$\bar{S} = \sqrt{2\bar{S}_{ij} \bar{S}_{ij}}. \quad (2.43)$$

This model is thus very simple to implement, but its main drawback is the poor performance in transitional flows. Indeed, the appropriate value of C_S depends on the flow regime. Many efforts have been made to extend this model and one of the most notable is the dynamic model proposed by Germano et al. [34]. This model is able to predict with accuracy regions of laminar, transitional and fully developed turbulent flow, but it is not easy to implement for complex three-dimensional geometries. For the present work, it was chosen to employ the Vreman model [90]:

$$\nu_e = c \sqrt{\frac{B_\beta}{\alpha_{ij} \alpha_{ij}}}, \quad (2.44)$$

with:

$$\alpha_{ij} = \frac{\partial \bar{u}_j}{\partial x_i}, \quad (2.45)$$

$$\beta_{ij} = \Delta_m^2 \alpha_{mi} \alpha_{mj}, \quad (2.46)$$

$$B_\beta = \beta_{11} \beta_{22} - \beta_{12}^2 + \beta_{11} \beta_{33} - \beta_{13}^2 + \beta_{22} \beta_{33} - \beta_{23}^2, \quad (2.47)$$

$$\text{and } c \simeq 2.5 C_S^2. \quad (2.48)$$

This model is simpler to implement than the dynamic one and it is equally capable of predicting the various characteristics of different flow regimes.

The filtered system of equations is thus closed by applying the Boussinesq hypothesis to the anisotropic residual-stress tensor, whose contribution to the momentum equation is:

$$\frac{\partial \tau_{ij}^r}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\nu_e \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \right]. \quad (2.49)$$

Using the Vreman model, the eddy viscosity is a function of filtered velocities. Hence, the turbulent stress term is decomposed into an additional viscous term and two highly non-linear terms related to the gradient of the eddy viscosity:

$$\frac{\partial \tau_{ij}^r}{\partial x_j} = \frac{\partial \nu_e}{\partial x_j} \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \nu_e}{\partial x_j} \frac{\partial \bar{u}_j}{\partial x_i} + \frac{\partial^2 \bar{u}_j}{\partial x_i \partial x_i}. \quad (2.50)$$

These contributions of the gradient of the eddy viscosity are usually very small and they are therefore neglected in the global computation.

The model has been implemented in the three-dimensional solver as an additional prediction step:

$$\mathbf{u}_i^{**} = \mathbf{u}_i^* + \frac{\Delta t}{\Delta \Omega_i LU_\infty} \sum_j \left(\frac{\partial \mathbf{u}^*}{\partial n} \right)_j \Delta S_j. \quad (2.51)$$

The model comes down to compute discrete gradients of velocity field at each face centre. The diamond's method, as previously explained, has been used. In the simulations performed in the present work, the classical value of the Smagorinsky constant, equal to 0.17 has been used.

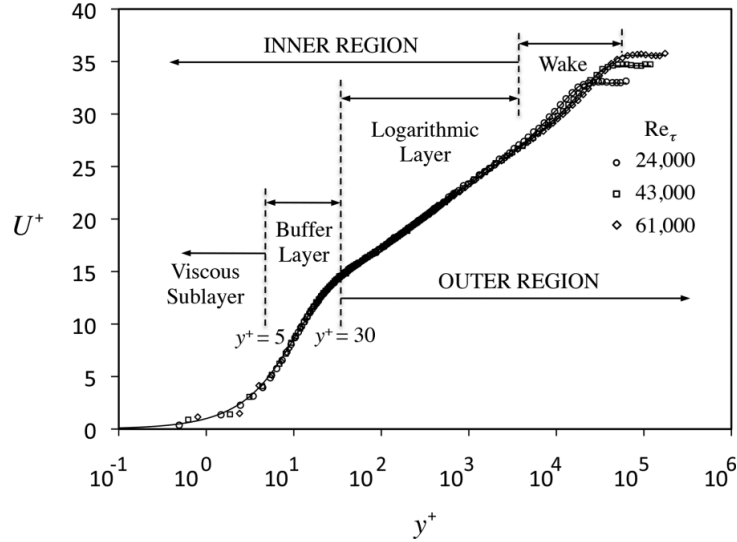


Figure 2.8: Turbulent boundary layer profiles for different Reynolds numbers

4.2 Near Wall Modelling

Despite the growth in computer performances, wall resolved Large Eddy Simulation still remains unfeasible to model the flow around wind turbines. It has been estimated by Choi et al. [20] that the number of grid points required by a wall resolved LES is of the order of $Re^{\frac{13}{7}}$, whereas it is approximately equal to Re for wall modelled LES. Moreover, the use of octree discretization in the developed solver, prevents to properly model the boundary layer near walls. Indeed, grid refinement with octrees allows only to divide by 2 the size of the cells whereas the anisotropic boundary layer around the wind turbine blades would require to have very thin cells around the obstacle. Wall modelling has thus been investigated in the present work.

The simplest approach for wall modelling is the use of the so-called wall functions, by which it is possible to impose a correction on the predicted velocity field in the first cells close to the fluid-body interface. The idea behind this technique is that the turbulent boundary layer velocity profile on a flat plate and with no external pressure gradient, is universal if properly normalized. Let u_τ be the friction velocity (or shear velocity), defined as:

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}}, \quad (2.52)$$

with τ_w the wall shear stress, that express:

$$\tau_w = \mu \left(\frac{\partial U}{\partial y} \right)_w, \quad (2.53)$$

where U is the tangential velocity and y is the normal direction. Using the friction velocity it is possible to define two adimensional variables by which an universal velocity profile can be described:

$$y^+ = \frac{y u_\tau}{\nu}, \quad U^+ = \frac{U}{u_\tau}. \quad (2.54)$$

As can be observed in figure 2.8, the turbulent boundary layer can be divided into four sub-regions:

- viscous sublayer,

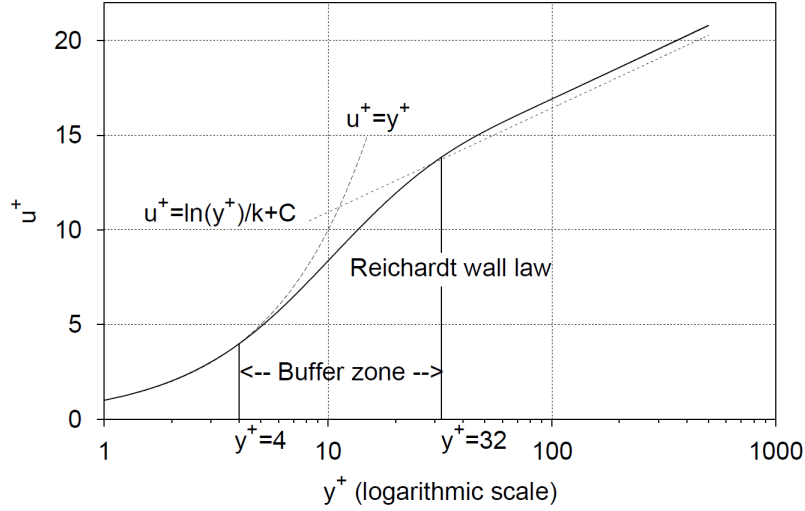


Figure 2.9: Reichardt's wall law. Found in [2].

- buffer layer,
- logarithmic layer,
- wake region.

The outer layer is composed by part of the logarithmic layer and the wake region and the flow is there strongly dependant on the geometry and on the Reynolds number. On the other hand, the inner layer, formed by the viscous sublayer and the buffer layer, allows to define a function f so that $U^+ = f(y^+)$. Traditionally, the universal profile has been described using two distinct analytical functions:

$$u^+ = f(y^+) = \begin{cases} y^+, & \text{if } y^+ < 5 \\ \frac{1}{k} \log y^+ + B, & \text{if } y^+ > 30, \frac{y^+}{\delta} < 0.3 \end{cases} \quad (2.55)$$

where $k = 0.41$ is the Von Karman constant, $B = 5.2$ and δ is the boundary layer thickness, which is a function of the Reynolds number. The buffer layer is a transition region between the two laws. It is possible to find in the literature many different functions to describe the profile with a unified law, such as those developed by Spalding [85]. For the present work, the Reichardt's wall law is considered as described in [2]:

$$u^+ = f_w(y^+) = \frac{1}{k} \log(1 + ky^+) + 7.8 \left(1 - e^{-\frac{y^+}{11}} - \frac{y^+}{11} e^{-0.33y^+} \right). \quad (2.56)$$

As it can be noticed in figure 2.9, the advantage of using Reichardt's wall law is the possibility to capture with a reasonable level of accuracy the three different regions of the inner layer using one analytical expression.

The numerical implementation has been done in the three-dimensional solver by following the guideline of De Tullio [26]. In a first loop, all the interface cells in the fluid are identified. In these cells, the velocity will be imposed using (2.56). Then, for each interface cell, the tangential velocity U_2 is evaluated in a point located on the same local normal, at a distance δ , as shown in figure 2.10. The position of point 2 is:

$$x_2 = x_1 + [\delta - \varphi(x_1)] \frac{\nabla \varphi(x_1)}{\|\nabla \varphi(x_1)\|}. \quad (2.57)$$

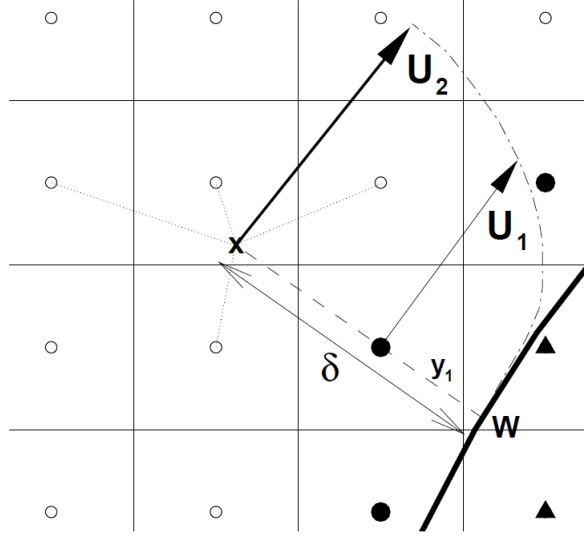


Figure 2.10: Implementation of the wall law. Found in [26].

As a remind, φ represents the level-set function. Using U_2 , it is possible to obtain a first approximation of the wall shear stress:

$$\tau_w \simeq \mu \frac{U_2}{\delta}. \quad (2.58)$$

Therefore, we can evaluate the friction velocity:

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} \simeq \sqrt{\frac{\nu U_2}{\delta}}, \quad (2.59)$$

by which the distance in wall units of the second point can be calculated and used to update u_τ :

$$y_2^+ = \frac{\delta u_\tau}{\nu} \Rightarrow u_2^+ = f(y_2^+) \Rightarrow u_\tau = \frac{U_2}{u_2^+}. \quad (2.60)$$

This iterative process stops when convergence is reached. Once the final value of the friction velocity has been calculated, the distance of the interface point in wall units is:

$$y_1^+ = \frac{\varphi(y_1)}{\nu}. \quad (2.61)$$

The non-dimensional velocity is evaluated using the wall law:

$$u_1^+ = f_w(y_1^+). \quad (2.62)$$

Thus, the tangential velocity to impose at the interface cell is:

$$U_1 = u_\tau u_1^+. \quad (2.63)$$

The choice of the distance δ is critical for the correct behaviour of the wall correction. Indeed, if the distance is too large, due to the parallelization (the choice has been done to have access to the first layer of neighbours for each cell), the point where U_2 is evaluated may be outside the ghost layer. On the other hand, if δ is too small, U_2 would be calculated too close to the interface which is not accurate. So, we choose to impose δ equal to twice the biggest value of

Chapter 2. Numerical Modelling of Flow around a Wind Turbine on Octree Grids

the level-set function at all the interface points. Finally, since the point x_2 is not necessarily at the cell-centre (where the variables are stored), velocity values should be interpolated in order to obtain U_2 . The interpolation is done by looking for neighbour cells through the closest node of the cell which owns point 2. Then Radial Basis Functions are used for the interpolation with this stencil. The wall function adjusts the velocity field after the penalization step.

Chapter 3

Validation of the Numerical Tool

As described in the previous chapter, a numerical tool for solving incompressible Navier-Stokes equations on octree grids has been developed for two and three dimensional cases. Each main operator has been independently tested and compared with an analytical solution to confirm the space accuracy and the consistency. This chapter is now dedicated to the validation of the whole solver. First of all, the space consistency and accuracy of the 2D and 3D solvers have been evaluated. Then, the tool has been widely tested with benchmarks of two-dimensional cases found in the literature. The 3D code is then considered and a stability issue is reported. We found that the origin of this issue is the semi-Lagrangian scheme. Another scheme has thus been implemented and the tool was then tested with a benchmark of three-dimensional cases found in the literature. Finally, a comparison has been done between the octree code and a high-order Cartesian code.

1 Consistency and Accuracy of the Navier-Stokes Solver on Octree Grids

The first validation performed for the whole Navier-Stokes solver is the analysis of the consistency and accuracy in space.

Taylor-Green vortex

The Taylor-Green vortex is an unsteady flow of a decaying vortex, which is considered in the present work since it has an exact solution in two spatial dimensions. A domain $\Omega = [-\pi, \pi] \times [-\pi, \pi]$ is used and the solution of the Taylor-Green vortex is given by:

$$u = \cos x \sin y F(t), \quad v = -\sin x \cos y F(t), \quad (3.1)$$

where $F(t) = e^{-2\nu t}$, ν being the kinematic viscosity of the fluid and t the time. The pressure field p can be obtained by substituting the velocity solution in the momentum equation and is given by

$$p = \frac{\rho}{4} (\cos 2x + \cos 2y) F^2(t). \quad (3.2)$$

Dirichlet boundary conditions have been imposed as ghost cells just outside of the borders of the domain thanks to analytical expressions. The computed solution as well as the mesh used for the analysis are shown in figure 3.1. Indeed, in order to get the consistency and the order of convergence in space of the whole solver, the computed velocity field has been compared with the exact solution and the results can be seen on table 3.1. The time step used for this simulation

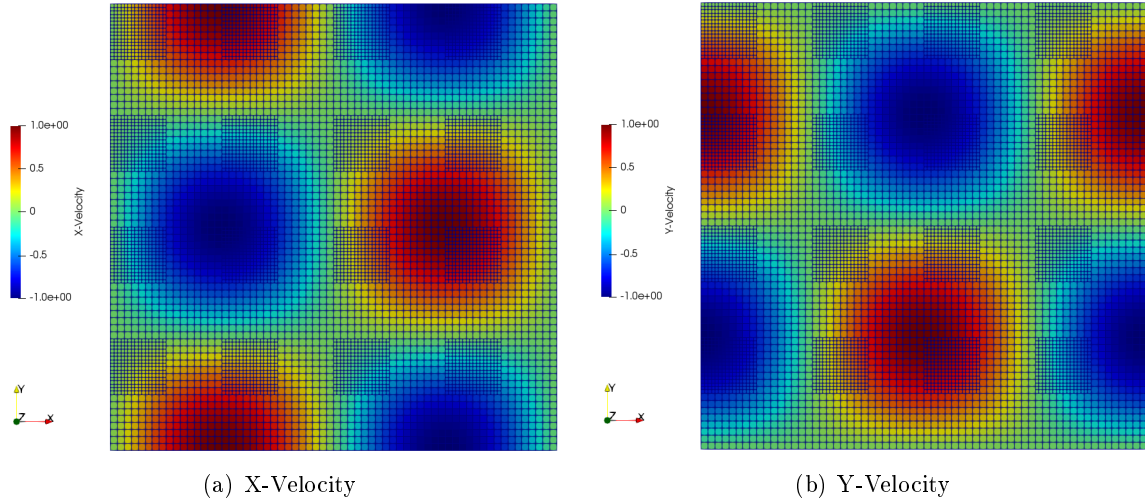


Figure 3.1: Velocity field of the Taylor-Green vortex on a quadtree grid (tree level = 7) after 0.1 s of simulation

Tree level - Number of cells	L_∞	Order	L_2	Order
5 - 1792	$4.800 \cdot 10^{-2}$		$1.970 \cdot 10^{-3}$	
6 - 7168	$1.29338 \cdot 10^{-2}$	1.89	$3.573 \cdot 10^{-4}$	2.46
7 - 28672	$2.573 \cdot 10^{-3}$	2.33	$4.984 \cdot 10^{-5}$	2.84
8 - 114688	$5.608 \cdot 10^{-4}$	2.20	$8.728 \cdot 10^{-6}$	2.51
9 - 458752	$1.320 \cdot 10^{-4}$	2.09	$1.731 \cdot 10^{-6}$	2.33
10 - 1835008	$3.400 \cdot 10^{-5}$	1.96	$3.614 \cdot 10^{-7}$	2.26

Table 3.1: Norm of the spatial error and orders of the Taylor-Green vortex for 2D quadtree grid

was $\Delta t = 10^{-4}$ and the final time was $t_{max} = 0.1$.

A second-order accuracy is obtained for the L_∞ norm and a few more for the L_2 (and L_1) norm. These results allow a first validation of our two-dimensional Navier-Stokes solver before going further with a comparison with data from literature in the next section.

Three-dimensional solver

Concerning the numerical tool in 3D, no exact solution exists for the whole computation of the Navier-Stokes equations to the best of our knowledge. Therefore, a partial analysis has been performed. An analytical function $f_{exact}(x, y, z) = \sin(x^2 + y^2 + z^2)$ is considered and the Poisson equation is solved using the developed methods explained in the previous chapter:

$$\nabla^2 f_c = \nabla \cdot s, \quad (3.3)$$

with $s(x, y, z)$ being the analytical gradient of f_{exact} . f_c is so computed. The next step consists in computing with the developed methods the gradient of f_c and comparing its values with the analytical expression $s(x, y, z)$. The consistency and the order of convergence in space of these steps, corresponding to the projection and correction steps of the 3D Navier-Stokes solver, can be seen on table 3.2.

Tree level - Number of cells	L_∞	Order	L_2	Order
5 - 11264	$3.027 \cdot 10^{-2}$		$1.111 \cdot 10^{-2}$	
6 - 90112	$1.385 \cdot 10^{-2}$	1.13	$3.948 \cdot 10^{-3}$	1.49
7 - 720896	$6.796 \cdot 10^{-3}$	1.03	$1.384 \cdot 10^{-3}$	1.51
8 - 5767168	$3.420 \cdot 10^{-3}$	1.01	$4.842 \cdot 10^{-4}$	1.51
9 - 46137344	$1.764 \cdot 10^{-3}$	0.99	$5.357 \cdot 10^{-5}$	1.51

Table 3.2: Norm of the spatial error and orders for 3D octree grid

Thus, we can validate the consistency of these parts of the solver. A first-order accuracy is obtained for the L_∞ norm and 1.5 for the L_2 norm. A further analysis is indeed necessary to validate the whole solver and a benchmark of test cases using data from literature will be used.

2 Validation with Data from the Literature for the Two-Dimensional Solver

In this section, the 2D solver is widely tested thanks to benchmarks found in the literature. It was indeed necessary to completely validate the methods developed for the two-dimensional case before extending these methods to a 3D framework. The computation of the aerodynamic forces exerted on a body is necessary for this work to validate the solver. The aerodynamic forces have the following expression:

$$\mathbf{F} = \int_{\partial\Omega_{body}} \left(-pI + \nu\rho \left(\vec{\nabla} \cdot \mathbf{u} + \vec{\nabla} \cdot \mathbf{u}^T \right) \right) \mathbf{n} dS, \quad (3.4)$$

I being the identity matrix. The chosen method of computation uses the Lagrangian markers that define the geometry of the body. A loop is carried out over the markers and for each marker, the cell that owns the marker is detected. Then, neighbours through the closest node of the cell to the marker are identified and a bilinear interpolation is performed with this stencil. Around the body, a Cartesian grid configuration is used in order to have the best accuracy possible in the computation of the aerodynamic forces. The pressure value is so obtained at the marker as well as the velocity derivatives, thanks to the bilinear expression whose gradient is straight forward to obtain.

The simulations performed in this section used the PlaFRIM cluster, whose nodes are composed by a couple of 12-core Intel Haswell CPUs with a clock speed of 2.5 GHz and a shared memory of 128 GB (see <https://www.plafrim.fr/>).

2.1 Fixed Cylinder Case

The 2D cylinder is a test-case that is fully studied in the literature. It was chosen in the present work to focus on three different Reynolds numbers that correspond to three different flow regimes. Laminar flow with periodic vortex shedding is studied at $Re = 100$, transition-in-wake regime is observed at $Re = 200$ when instabilities lead to the beginning of formation of streamwise vortex structure and finally unsteady flow with low turbulence is seen at $Re = 600$.

Geometry and Mesh

All simulations performed in this section used a cylinder with diameter equal to 1. The size of

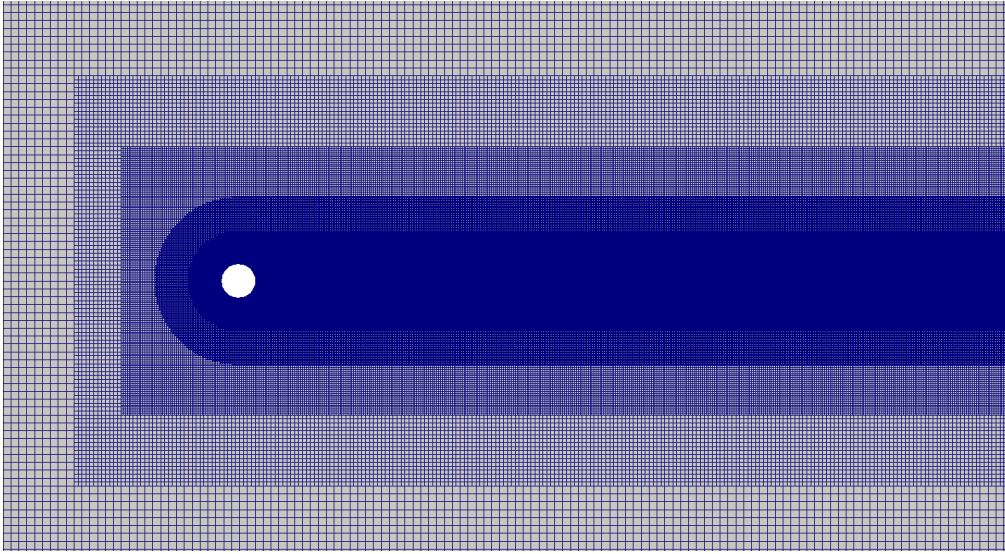


Figure 3.2: Quadtree mesh around a two-dimensional cylinder (minimum tree level = 7)

the domain is 30 and the lateral boundaries are located at 15 diameters from the centre of the cylinder. The inlet is located at 7 diameters in front of the cylinder and the outlet is thus at 23 diameter behind. The mesh used can be seen in figure 3.2. Five different levels of refinement are employed in this grid configuration. The aim is to simulate accurately the cylinder's wake. In order to choose the right compromise between accuracy and computing time, a grid convergence study has been performed.

Grid convergence study

In order to analyse the grid convergence, the case of a flow past an impulsively started cylinder has been chosen. Indeed, there exists a numerical solution of reference valid for short times as explained by Bar-Lev and Yang [5] and, especially, Ploumhans et al [71] studied the case of $Re = 550$. Thus, this flow has been studied in our work with three different meshes similar to figure 3.2 with different minimum level of refinement. The characteristics of each grid and the computational time spent to simulate 6 s of flow are explained in table 3.3. The chosen time step for the simulations is adaptive and is computed at each time step. It depends on the size of each cell, on the velocity field in that cell and on the CFL (Courant-Friedrichs-Lewy) parameter. This parameter has been chosen equal to 0.3.

The drag coefficient history has then been compared with the analytical solution and can be seen in figure 3.3. This comparison allows to validate the developed solver. With a minimum tree level of 9 (which corresponds to around 270 cells on the diameter of the cylinder) the results obtained are very close to the analytical ones but this mesh implies a high computational time. Therefore the grid with a minimum tree level of 8 has been used for the results described below.

The Strouhal number is a dimensionless variable describing oscillating flow mechanisms. It can be defined as:

$$S_t = \frac{fD}{U_\infty} \quad (3.5)$$

with f the frequency of the vortex shedding, D the diameter of the cylinder and U_∞ the inflow velocity at inlet.

2. Validation with Data from the Literature for the Two-Dimensional Solver

Minimum tree level	Minimum size	Number of cells on the diameter	Global number of cells	Number of processors	Computational time
7	$1.465 \cdot 10^{-2}$	68	442 516	72	36 min
8	$7.324 \cdot 10^{-3}$	136	1 764 136	120	6 hours 56 min
9	$3.662 \cdot 10^{-3}$	273	7 091 050	480	35 hours 49 min

Table 3.3: Two-dimensional grid characteristics and computational time for a simulation of 6 s

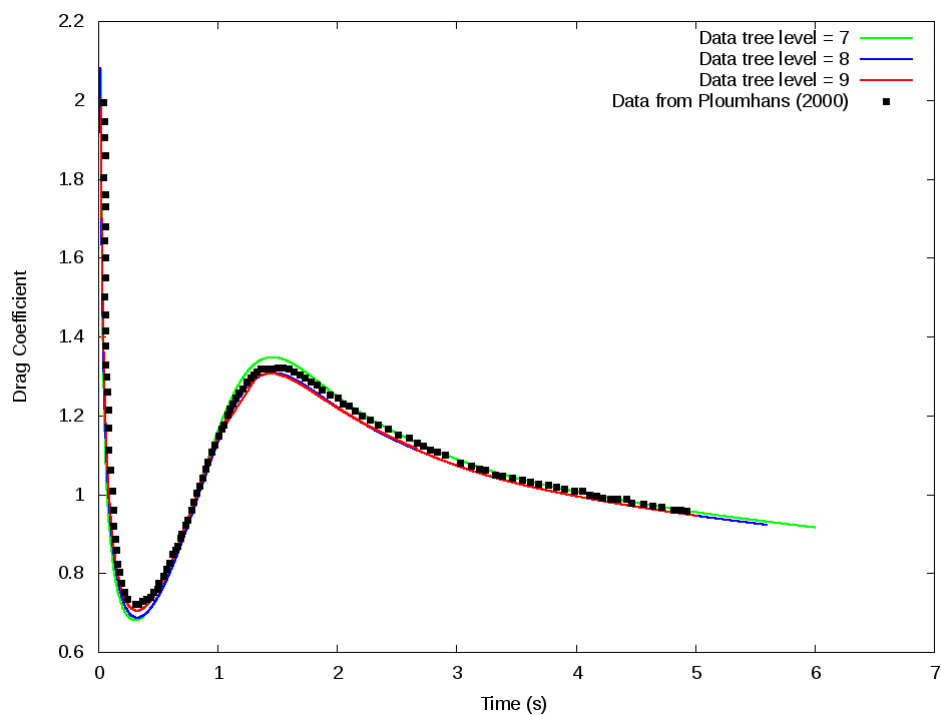


Figure 3.3: Drag coefficient history at $Re = 550$ with the numerical solution of reference and the computed results with 3 different grid sizes

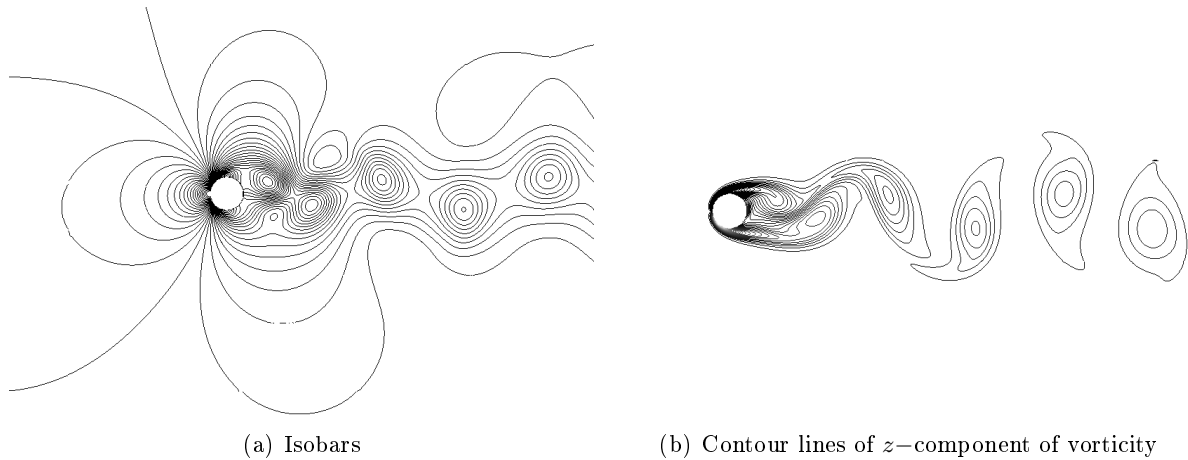


Figure 3.4: Isobars and contour lines of z -component of vorticity obtained at $t = 100$ and $Re = 100$ for the flow around a circular cylinder

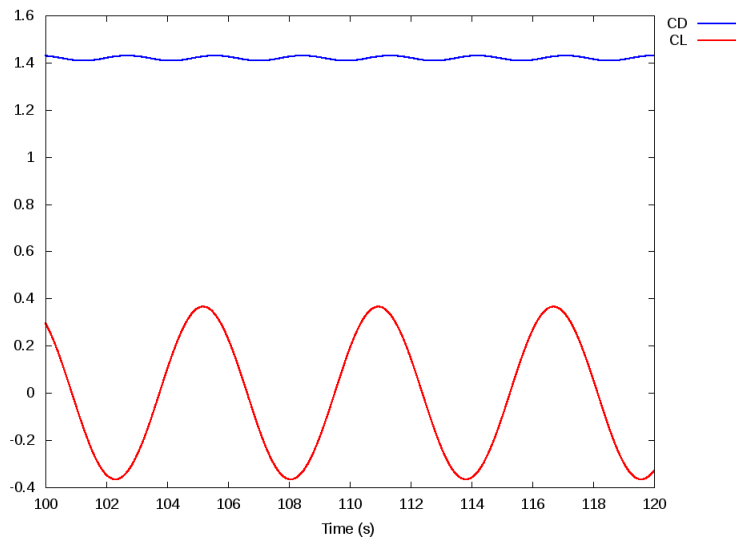


Figure 3.5: Drag and Lift coefficients history at $Re = 100$

Flow at $Re = 100$

In order to have several points for the comparison with data from the literature, the flow at $Re = 100$ has been simulated. Figure 3.4 shows the isobars and the contour lines of z -component of vorticity obtained at $Re = 100$ and $t = 100$ s. This flow shows a periodic von Karman vortex street and the figure can be compared with the work done by Bergmann [8]. This periodicity can indeed be observed in figure 3.5 with the time history of aerodynamic coefficients (lift and drag, named respectively C_L and C_D).

Table 3.4 shows a comparison of the Strouhal number and averaged drag coefficient obtained numerically in the present work with data from literature. The results obtained for both the Strouhal number and the averaged drag coefficient are slightly overestimated (around 5 % of difference concerning the mean Strouhal number) compared with the literature. We can conclude that the numerical tool shows accurate results for the flow at $Re = 100$.

Flow at $Re = 200$

Re	<i>Authors</i>	S_t	C_D
100	Braza et al. [12]	0.1600	1.3600
	Williamson (experimental) [91]	0.1640	-
	Henderson [39]	0.1664	1.3500
	He et al. [37]	0.1670	1.3528
	Bergmann [8]	0.1660	1.4100
	Present work	0.1736	1.4193

Table 3.4: Comparison of Strouhal number and drag coefficient for the flow past a circular cylinder at $Re = 100$

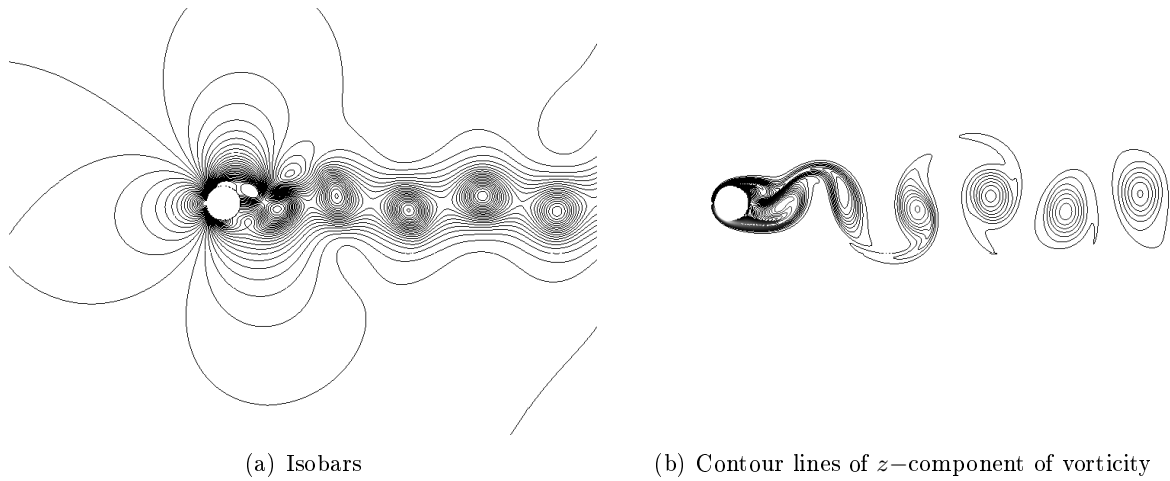


Figure 3.6: Isobars and contour lines of z -component of vorticity obtained at $t = 100$ and $Re = 200$ for the flow around a circular cylinder

In order to characterize the transition-in-wake regime, the flow at $Re = 200$ has been simulated. Indeed, as explained by Williamson [92] this flow configuration corresponds to the beginning of formation of streamwise vortex structure. The flow can be seen in figure 3.6 with the isobars and the contour lines of z -component of vorticity and a von Karman vortex street can be observed.

The drag and lift coefficient history of this simulation can be seen in figure 3.7. This figure was used to compute the averaged drag coefficient and the Strouhal number of this flow. The results and the comparison with the other data from literature can be observed on table 3.5. The mean drag coefficient shows a good agreement with the literature. As this two-dimensional simulation shows good results in simulation of a flow that could a priori be three-dimensional, that implies that these effects should be small enough, so that our two-dimensional solver is validated.

Flow at $Re = 600$

This flow regime at $Re = 600$ is faintly turbulent with appearance of three-dimensional effects (as explained by Williamson [91] [92]). The figure 3.8 presents the pressure field and the z -component of the vorticity at $t = 100$. It can be observed that the vortices constituting the von Karman vortex street have a higher frequency than the previously studied flows. That

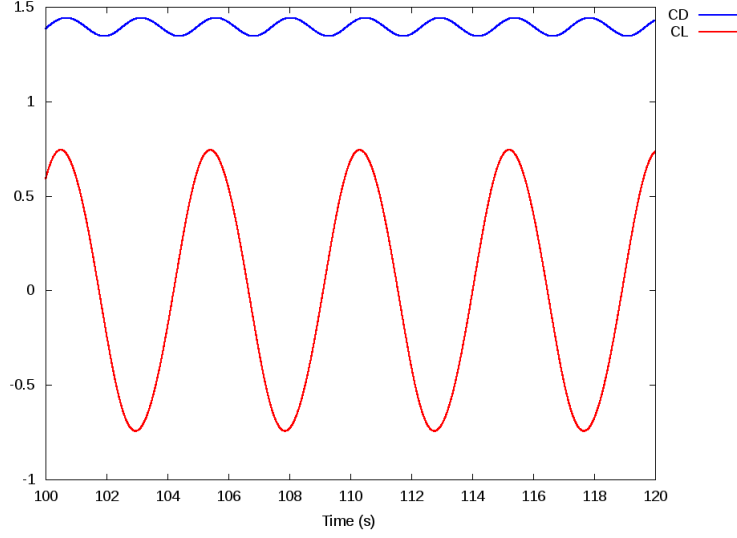


Figure 3.7: Drag and Lift coefficients history at $Re = 200$

Re	<i>Authors</i>	S_t	C_D
200	Braza et al. [12]	0.2000	1.4000
	Henderson [39]	0.1971	1.3412
	He et al. [37]	0.1978	1.3560
	Bergmann [8]	0.1999	1.3900
	Present work	0.2039	1.3951

Table 3.5: Comparison of Strouhal number and drag coefficient for the flow past a circular cylinder at $Re = 200$

can indeed be seen with the Strouhal number, whose value is higher as confirmed in table 3.6. The results obtained with the developed quadtree solver overestimated slightly the data from literature in this flow regime. The drag and lift coefficient history of this simulation can be seen in figure 3.9. This figure was used to compute the averaged drag coefficient and the Strouhal number of this flow.

Re	<i>Authors</i>	S_t	C_D
600	Henderson [39]	0.2294	1.4682
	He et al. [37]	0.2306	1.4641
	Henderson [38] (bi-dimensional tendency curve)	-	1.47
	Present work	0.2351	1.5096

Table 3.6: Comparison of Strouhal number and drag coefficient for the flow past a circular cylinder at $Re = 600$

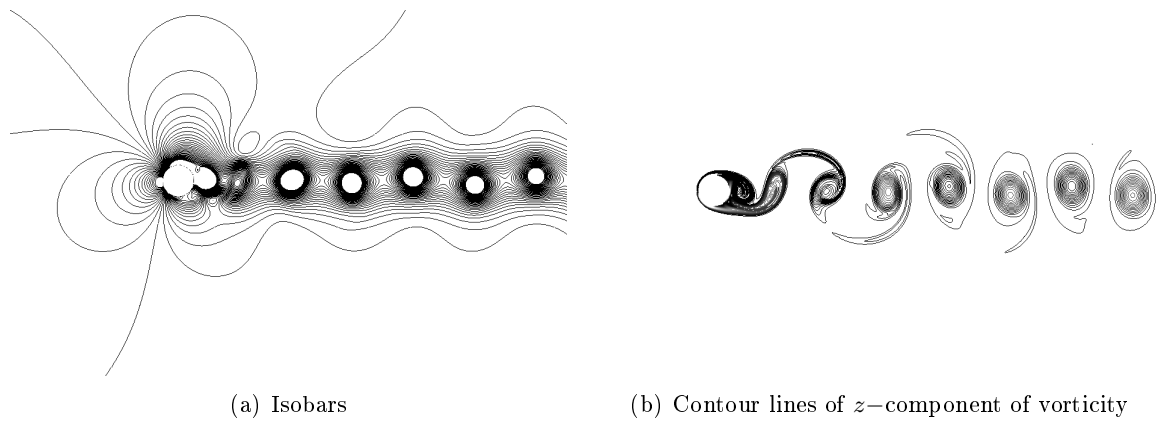


Figure 3.8: Isobars and contour lines of z -component of vorticity obtained at $t = 100$ and $Re = 600$ for the flow around a circular cylinder

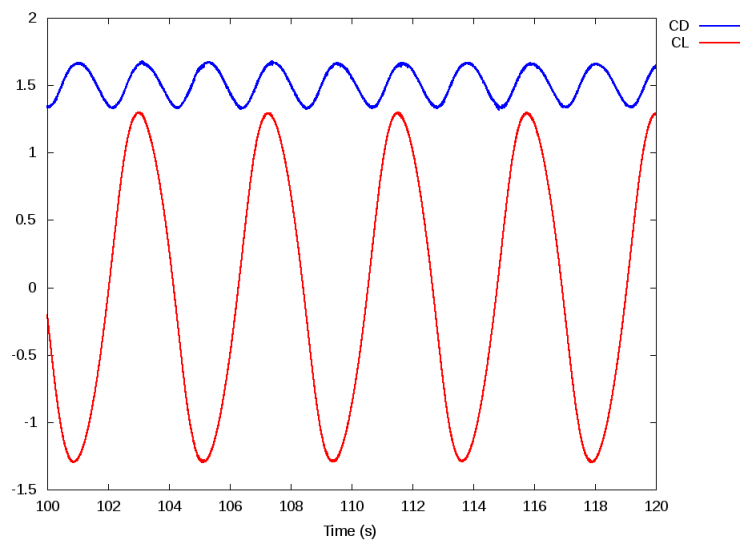


Figure 3.9: Drag and Lift coefficients history at $Re = 600$

2.2 Airfoil Case

The solver developed has then been tested with the case of the flow around a 2D airfoil, that shows a more complex geometry than the cylinder. Data from literature have been used to compare with our solver as can be observed on table 3.7 where the averaged lift coefficient at different angles of attack is reported. The quadtree solver shows accurate results in this flow configuration.

Re	<i>Authors</i>	$\alpha = 10^\circ$	$\alpha = 15^\circ$
1000	Mittal et al [58]	0.42	-
	Khalid et al [43]	0.39	0.58
	Suzuki et al [86]	-	0.72
	Kurtulus [45]	0.42	0.70
	Present work	0.38	0.66

Table 3.7: Comparison of lift coefficient for the flow past a NACA0012 airfoil at $Re = 1000$ with different angles of attack

Thus, the two-dimensional quadtree code has been validated for different flow configurations and for different geometries. The three-dimensional solver on octree grids is now studied and compared with data from literature for validation.

3 Stability Issue with the Semi-Lagrangian Scheme for Three-Dimensional Solver

While performing a low Reynolds simulation of the flow past a sphere on a hemispherically refined mesh, some numerical instabilities were detected in the proximity of the spherical level-jump, making the simulation diverge in a few seconds. An investigation has been performed to understand which part of the solver could imply such instabilities. Moreover, every operator was validated with an analytical expression. Every part of the Navier-Stokes solver has been tested and stabilized operators were implemented in order to stabilize the treatment of level jumps. But nothing changed and finally it turned out that only the advection term of the discretized Navier-Stokes equations was involved. The use of the semi-Lagrangian scheme as described in the previous chapter was unstable with some octree grids (not all kind of them). This instability is due to the fact that in the first hand, we are using non-uniform grids and in the second hand, when we interpolate the velocity field at the root of the characteristic curve the stencil used is not necessarily upwind. It can be noticed that, when stable, the semi-Lagrangian shows accurate results when compared to the data from literature.

An explicit Eulerian scheme has been implemented with a first order upwind reconstruction. This was indeed the easiest way to proceed considering the remaining time of the thesis. The numerical tool developed constitutes actually a basic structure that will lead to further improvements. The prediction step (2.7) is so rewritten with the Eulerian scheme:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + \nabla \cdot (\mathbf{u}^n \otimes \mathbf{u}^n) = \nu \Delta \mathbf{u}^*. \quad (3.6)$$

A finite volume discretization has been used to compute the convective term on each cell volume Ω_c .

$$\nabla \cdot (\mathbf{u}^n \otimes \mathbf{u}^n) = \frac{1}{\Omega_c} \oint_{\partial\Omega_c} (\mathbf{u}^n \otimes \mathbf{u}^n) \cdot \mathbf{n} dS, \quad (3.7)$$

where \mathbf{n} represents the outward normal of the cell. Hence, we can write for each cell i of the domain:

$$\nabla \cdot (\mathbf{u}^n \otimes \mathbf{u}^n)_i = \frac{1}{\Omega_c} \sum_j (\mathbf{u}^n \otimes \mathbf{u}^n)_j \cdot \mathbf{n}_j \Delta S_j. \quad (3.8)$$

The flux on the cell faces are then evaluated through a first order upwind reconstruction:

$$(\mathbf{u}^n \otimes \mathbf{u}^n)_{j \cdot \mathbf{n}_j} = u_i \max(\mathbf{u}_j \cdot \mathbf{n}_j, 0) + u_e \min(\mathbf{u}_j \cdot \mathbf{n}_j, 0), \quad (3.9)$$

where e is the cell pointed by the outward normal vector of the i -th cell, and the velocity at the face is basically:

$$\mathbf{u}_j = \frac{\mathbf{u}_i + \mathbf{u}_e}{2}. \quad (3.10)$$

Once this method has been implemented, the instability issue was totally solved as can be seen in figure 3.10. The three-dimensional solver has then been fully tested by comparing with benchmark from literature.

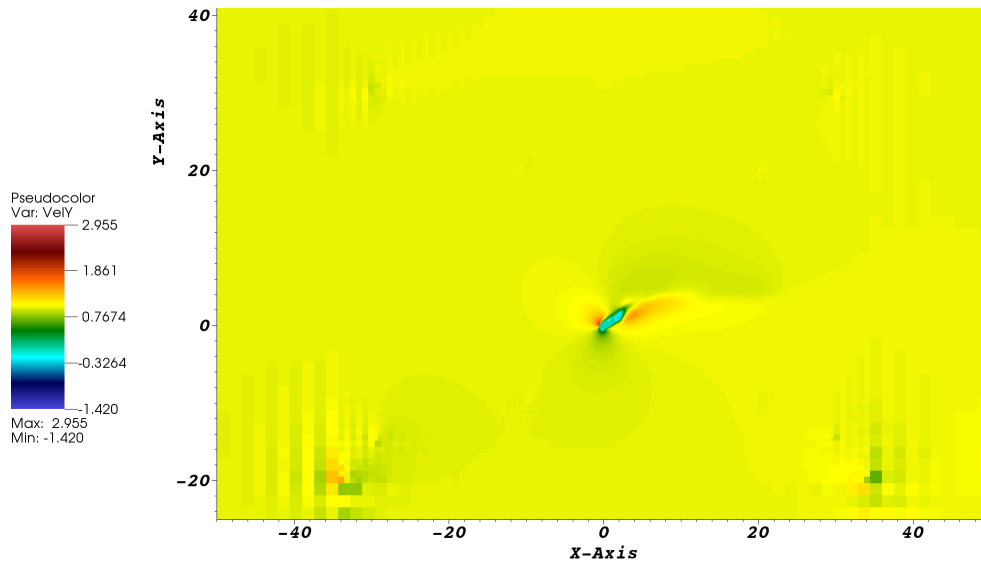
4 Validation with Data from Literature for Three-Dimensional Case

This section explains the work done in order to validate the developed 3D solver on octree grids. Two popular benchmarks have been used with the flow past a sphere at $Re = 500$ and the flow past a cylinder at $Re = 3900$. The flow past a sphere at low Reynolds number has been firstly studied to validate the numerical methods on a simple test case before dealing with higher Reynolds number for the case of the flow past a cylinder.

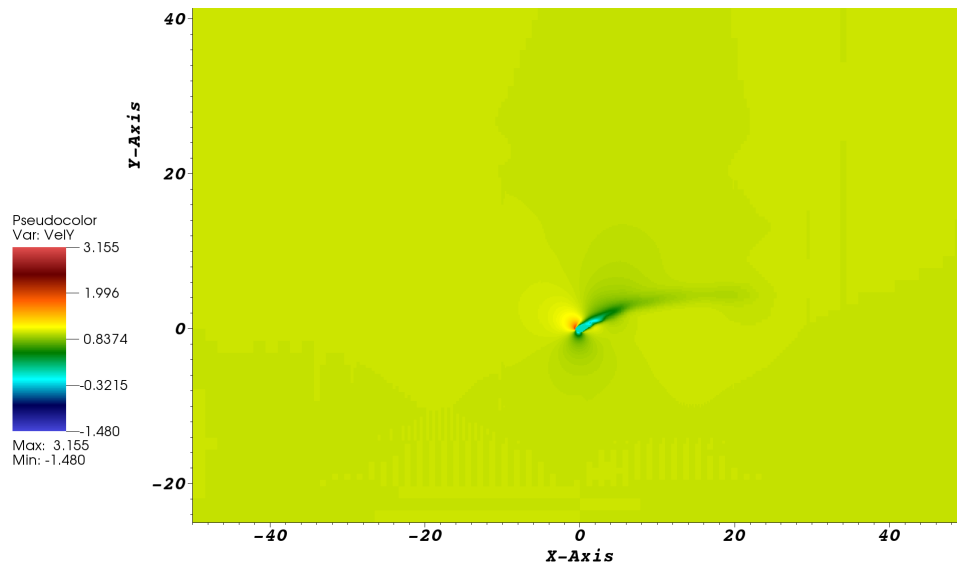
The computation of the aerodynamic forces that exerted on a 3D body has been implemented in the solver. The method is based on the Lagrangian markers (that define the geometry of the body), but the method is slightly different than the two-dimensional case where bilinear interpolations were employed. A loop is carried out on the markers and for each marker, the cell that owns the marker is detected. Then, neighbours through the closest node of the cell to the marker are identified. A RBF interpolation is then performed with this stencil to directly get the pressure value at the Lagrangian marker position. The velocity gradients are also needed at these positions. The same stencil as previously explained is used. A Moving Least Square interpolation has been used as described in [35]. That method comes down to a localized least-squares procedure, in which the error E is expressed as:

$$E = \sum_{n=1}^N L_n [u_n - q(\mathbf{x}, t)]^2 \quad (3.11)$$

with N representing the number of neighbours identified, u_n is the function to interpolate (the x-velocity for instance) where values are known at neighbour cell-centres. $q(\mathbf{x}, t)$ is the local first-order polynomial interpolation whose components are the position differences between each neighbour and the Lagrangian marker. The coefficients of the local polynomial are thus directly the local gradients. L_n is named the "localization parameter" and weights the contribution of the different points to the error. In the present work, we took $L_n = \frac{1}{Dist^3}$ equals to the inverse



(a) Case with semi-Lagrangian prediction



(b) Case with Eulerian prediction

Figure 3.10: Simulation of a blade in a rotating frame, y-Velocity at plane $z = -5$

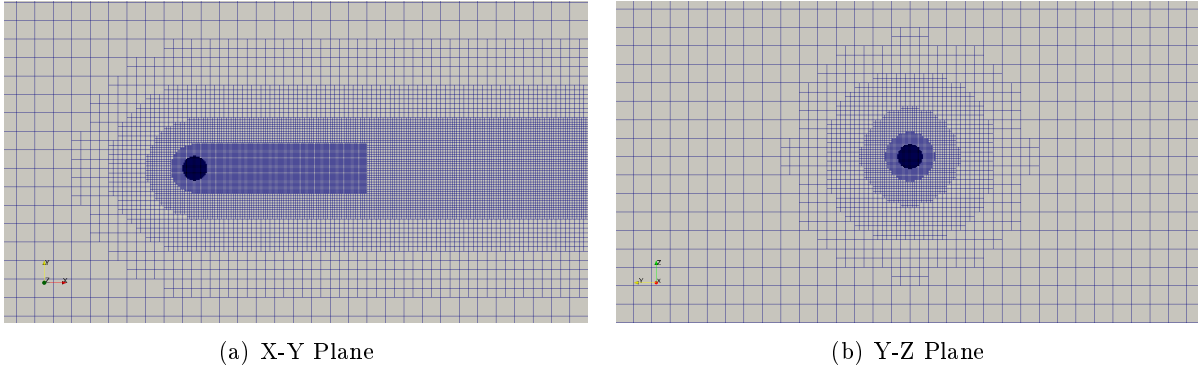


Figure 3.11: Octree mesh used for the flow around a sphere - Tree level 5

of the distance with a power of 3 after a preliminary study on this parameter. Minimization of E with respect to each of the coefficient (local gradients) yields a 3×3 system of linear equations. The reader is referred to the report in [35] for the detailed linear system. Finally, solving these systems allow to obtain directly the gradients at each of the Lagrangian markers.

4.1 Sphere Case

This benchmark has been chosen to validate the operators developed in our solver with a low Reynolds flow configuration. All the simulations performed used a sphere with a diameter equal to 1. The size of the computational domain is 24 diameters and the lateral boundaries are located at 12 diameters from the centre of the sphere. The inlet is located at 8 diameters in front of the sphere and the outlet is thus at 16 diameters behind. An example of the mesh used can be seen in figure 3.11. Four different levels of refinements are employed and a grid convergence study has been performed with different minimum level of octree. The characteristics of each grid can be seen in table 3.8 as well as the conditions of the simulations. The PlaFRIM cluster has been employed for these simulations. An adaptive time step has been used.

Minimum tree level	Minimum size	Number of cells on the diameter	Global number of cells	Number of processors	Computational time
5	$4.688 \cdot 10^{-2}$	21	590 892	24	8 hours
6	$2.344 \cdot 10^{-2}$	42	4 711 596	144	25 hours
7	$1.172 \cdot 10^{-2}$	85	37 597 820	960	36 hours

Table 3.8: Three-dimensional grid characteristics and computational time for a simulation of 100 s

The choice of the CFL number is very important and the results show really different behaviours depending on its value. The figure 3.12 describes the time history of drag coefficient with the 3 different octree grids and with a CFL condition of 0.5 compared to a CFL of 0.3. The higher CFL implies drag coefficients too high compared to existing value in the literature. Therefore for the future simulations a CFL condition of 0.3 has been imposed.

The averaged drag coefficient obtained is in line with results found in literature as remarked in table 3.9. With the minimum level of 7, it seems like the grid convergence is reached since the

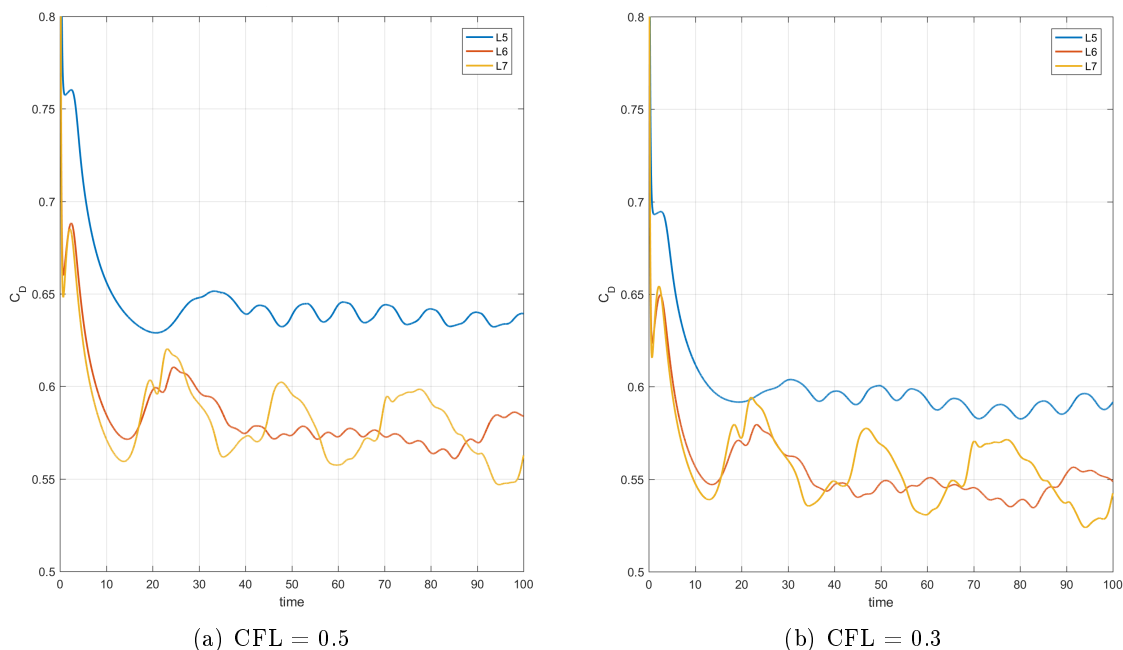


Figure 3.12: Time history of drag coefficients for the flow at $Re = 500$ with 3 different grid refinements and 2 different CFL conditions

drag coefficient is stabilized compared to the level 6. Visualizations of the flow obtained with the CFL condition of 0.3 and the minimum tree level of 7 can be seen in figure 3.13.

Re	<i>Authors</i>	C_D
500	Campregher et al. [15]	0.5200
	Fornberg [32]	0.4818
	Fadlun et al [29]	0.476
	Present work - tree level 5	0.5952
	Present work - tree level 6	0.5456
	Present work - tree level 7	0.5545

Table 3.9: Comparison of averaged drag coefficient for the flow past a sphere at $Re = 500$

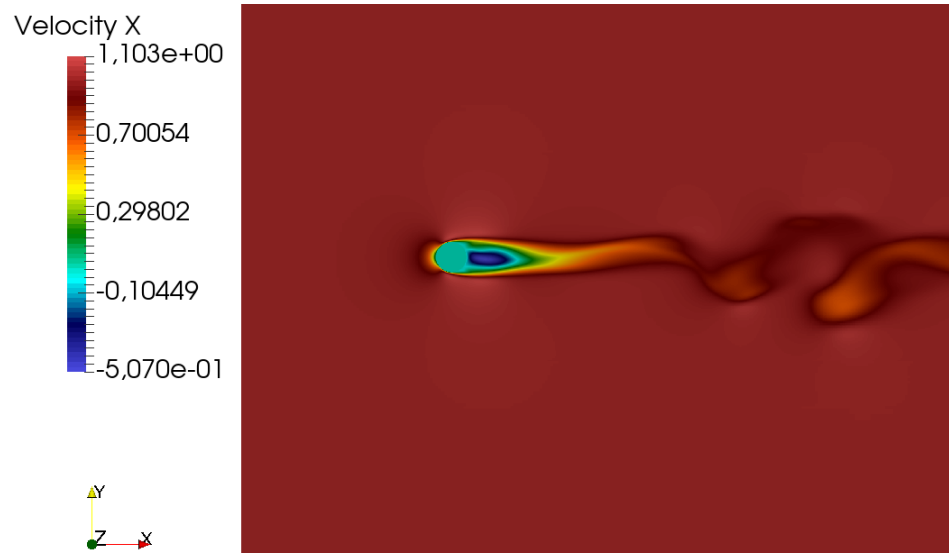
4.2 Cylinder Case

In the following sections, the simulations have been performed with the supercomputer Occigen¹.

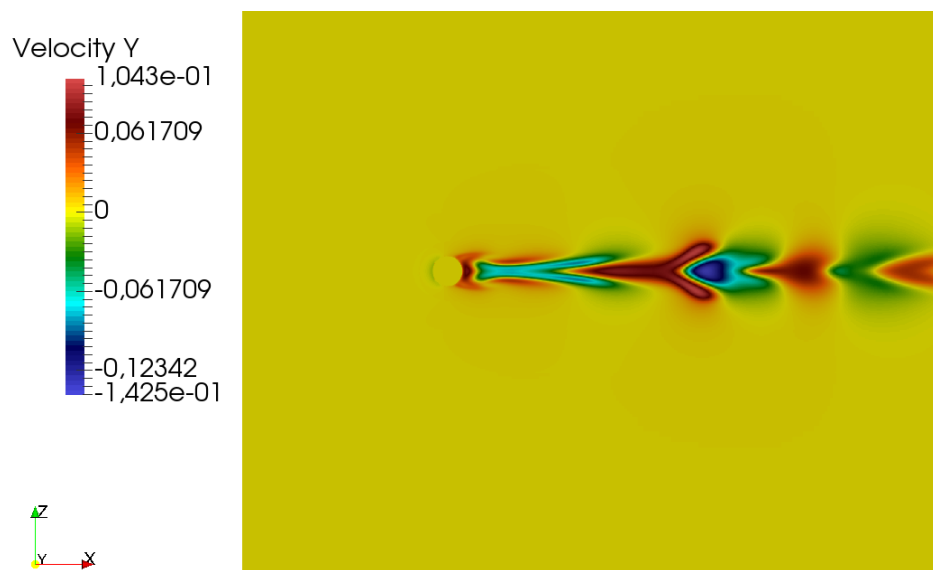
The flow past a cylinder benchmark at $Re = 3900$ has been used to test the high Reynolds behaviour of the octree Navier-Stokes solver, coupled with the Wall Modelled Large Eddy Simulation approximation (WMLES). Since the boundary layer is laminar at the considered Reynolds number, the Reichardt wall function is not suited. Therefore, a linear correction is implemented: $u^+ = y^+$, which can be considered as a sort of laminar wall function. As a reminder, u^+ represents the non-dimensional tangential velocity and y^+ is the dimensionless wall distance. This

¹www.cines.fr

4. Validation with Data from Literature for Three-Dimensional Case



(a) X-Velocity in the X-Y plane



(b) Y-Velocity in the X-Z plane

Figure 3.13: Flow past a sphere at $Re = 500$

case constitutes a first test and validation of all the models developed for the 3D code before studying a higher Reynolds flow in the next section. The flow regime at $Re = 3900$ is still challenging since transition waves appear in the free shear layer in the wake. One difficulty with our solver to simulate properly this test case is the fact that we are constrained to use cubic domains. Thus, if we want to have a large wake visualization, that implies also large lateral borders. As we are studying an infinite cylinder, no coarsening can be done in the direction of the cylinder length and a high number of cells arrives quickly. A compromise should be accepted between the accuracy in the wake of the cylinder and the number of cells. The chosen grid has the following characteristics:

- 5 level jumps,
- domain $\Omega = [-8; 16] \times [-12; 12] \times [-12; 12]$ respectively in the x , y and z directions,
- minimum size of the grid $\Delta x = 1.172 \cdot 10^{-2}$,
- number of points on the diameter of the cylinder = 85,
- total number of cells = 138 011 392.

Several experiments and computations have been carried out for the flow past a circular cylinder at $Re = 3900$ which will allow to compare the different data from literature with the developed solver. But we should be aware that this numerical tool, allows to have an interesting compromise between accuracy and computational time but we do not expect to be as accurate as high-order numerical methods. A simulation of 90 s of flow has been performed with 2880 processors. The associate computational time was around 72 hours. An adaptive time step has been used with a CFL condition of 0.3. For the first comparison, we used the computation of the averaged drag coefficient as can be seen in table 3.10. Our solver shows to be in line with data from literature for the mean drag coefficient.

Re	<i>Authors</i>	C_D
3900	LES of Rajani et al. (2016) [75]	1.01
	LES of Snyder and Degrez (2003) [80]	1.09
	LES DM of Park et al (2006) [67]	1.04
	LES DM1 of You and Moin (2006) [94]	1.01
	LES WALE of Ouvrard et al. (2010) [66]	1.02
	VMS-LES WALE of Ouvrard et al. (2010) [66]	0.94
	Experiments of Norberg (1987) [62]	0.98 ± 0.05
	Present work	0.97

Table 3.10: Comparison of averaged drag coefficient for the flow past a cylinder at $Re = 3900$

The pressure coefficient on the cylinder has then been studied and compared with experimental values of Norberg [62]. This computation has been done after a preliminary simulation for removing the initial transient. Pressure values have been computed at the middle of the cylinder and are averaged over 9 vortex sheddings (corresponding to around 40 seconds of simulations). We were indeed limited by the computational time of the simulation since we had a limited number of hours of simulation on the supercomputer Occigen. The results obtained can be seen

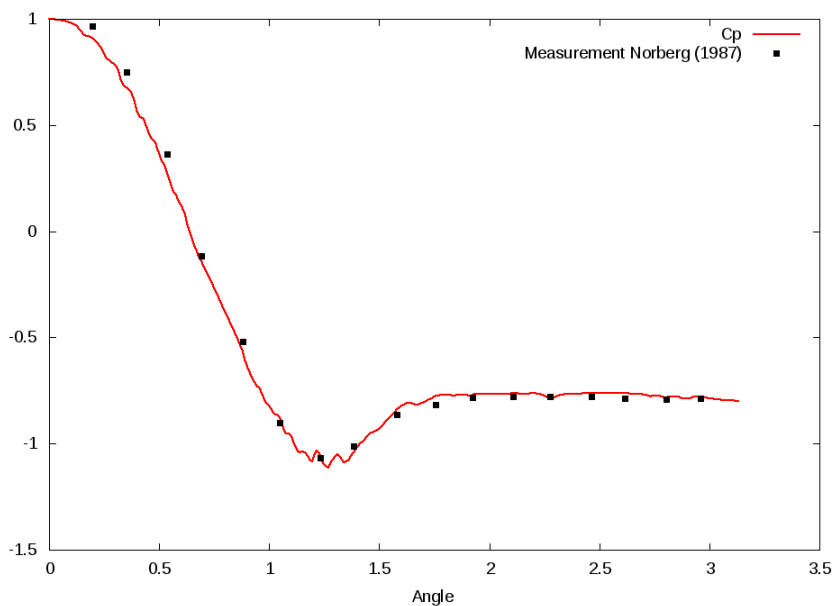


Figure 3.14: Pressure coefficient around the cylinder at $Re = 3900$. Angle of π corresponds to the wake of the cylinder, whereas angle of 0 is the incoming flow.

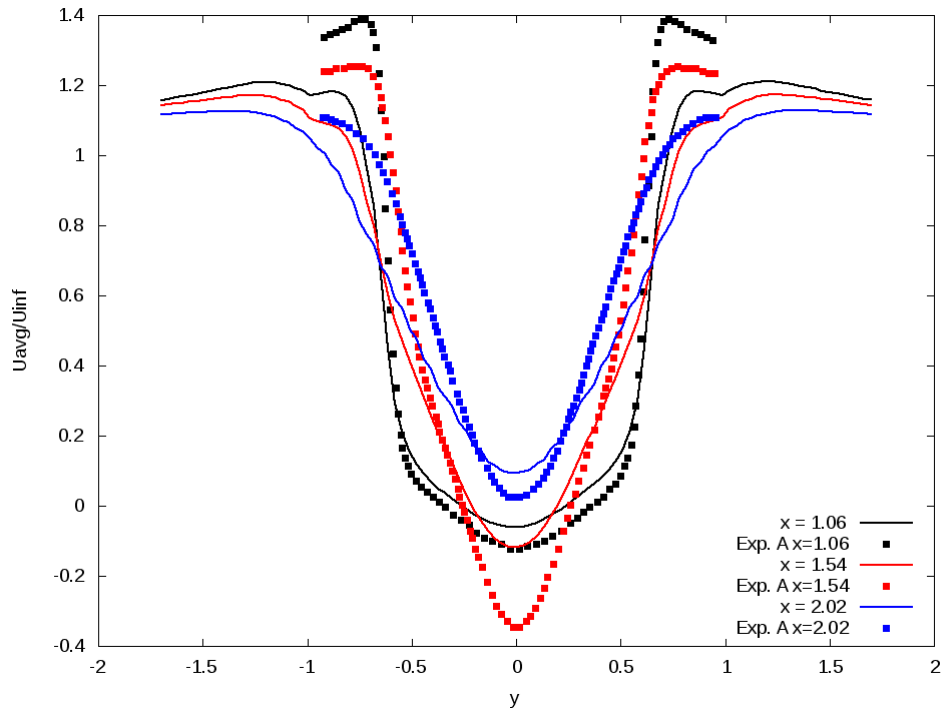
in figure 3.14 and show accurate results compared to the experimental values.

Then, a comparison has been performed between the present results and the experimental results obtained by Parnaudeau et al. in 2008 [68] (noted experiment A) and Lourenco and Shih in 1993 [51] (noted experiment B). In the same way as for the pressure coefficient, the numerical results are obtained after a preliminary simulation and by performing an average over a period of 9 vortex sheddings. Figure 3.15 shows the wake profile at three different locations ($x = 1.06$, $x = 1.54$ and $x = 2.02$). The results from the octree solver are in-line with the experimental data of Parnaudeau et al. [68] for the position close to the cylinder (at $x = 1.06$). The results obtained for the two other positions, farer from the cylinder, are then closer to the experimental data from Lourenco and Shih [51]. Indeed, a level jump is present in the wake at $x = 2$ which should imply a loss of accuracy in this area. Also, the period used for averaging the wake profile data could not have been very wide due to computational limits.

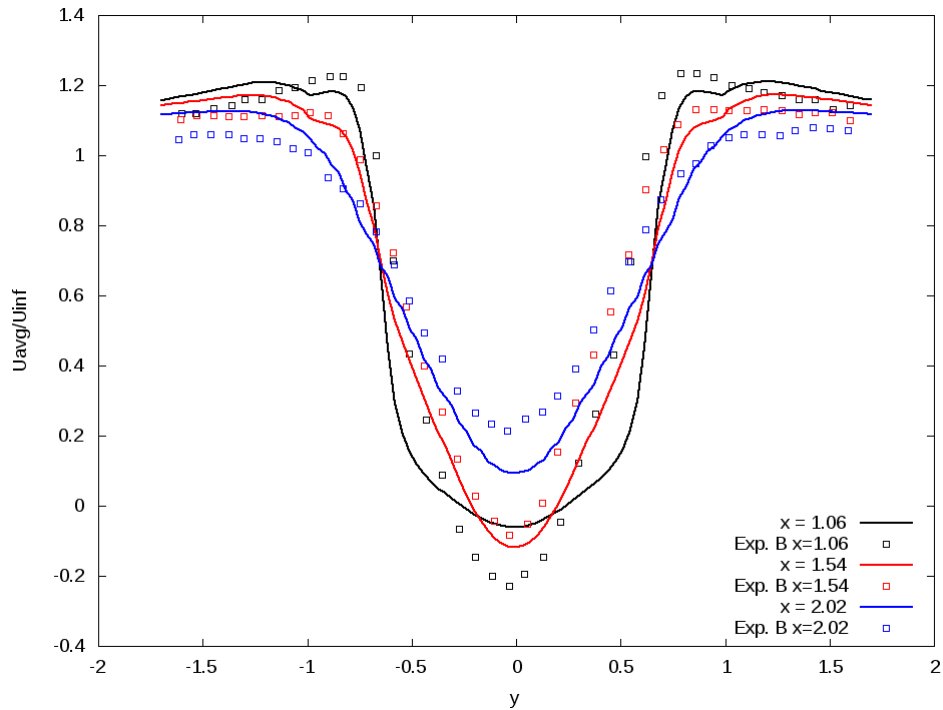
Finally, the results obtained with the developed solver are in accordance with the data from literature. They are not as accurate when looking at the velocity profile in the wake but this is probably due to the numerical dissipation implied by the first-order accuracy of the numerical tool. The next section will focus on a higher Reynolds number flow and take particular attention to compare the simulations performed and the results obtained with a high-order solver on Cartesian grids.

5 Comparison between Cartesian and Octree Solvers

The flow past a cylinder benchmark at $Re = 140000$ has been used to compare simulations performed with the developed octree solver and a Cartesian code that was developed before the present work. This solver has been developed and validated in-house by the MEMPHIS Team of INRIA and is called NaSCAR. High-order finite difference model was implemented with an immersed boundary method. Considering the simulation with the octree code, the same domain



(a) Comparison with experimental results of Parnaudeau et al. (2008)



(b) Comparison with experimental results of Lourenco and Shih (1993)

Figure 3.15: Wake profile of streamwise velocity at different positions obtained by an average over 9 vortex sheddings after a preliminary simulation for the flow past a cylinder at $Re = 3900$.

size and mesh configuration as previously described for the case at $Re = 3900$ has been used. The Cartesian code is not submitted to the constraint of having a cubic computational domain. Therefore the size of the domain chosen for the simulation was the following: $[-8, 16] \times [-8, 8] \times [0, 4]$. The boundary conditions are:

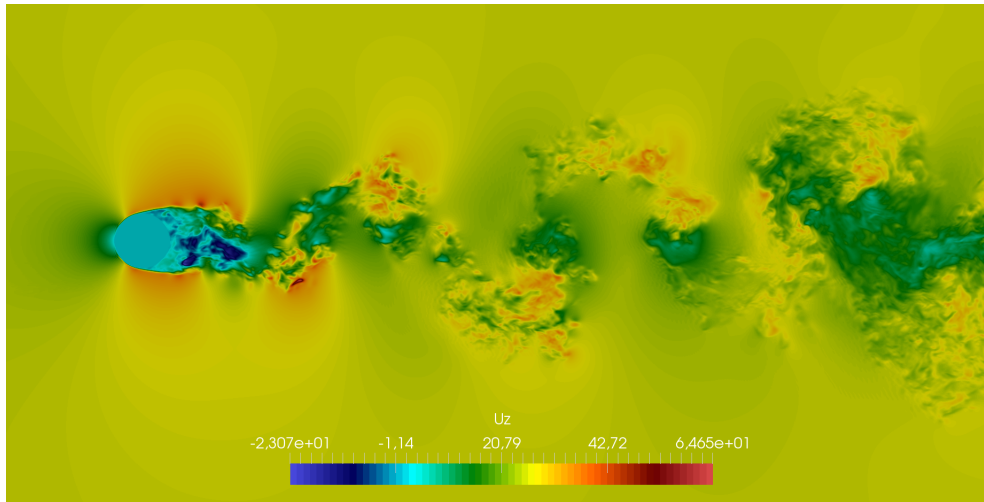
- Dirichlet boundary condition for the velocity at the inlet (x -direction)
- homogeneous Neumann condition for the velocity at outlet
- periodic conditions in the lateral z -directions that correspond to the cylinder length
- homogeneous Neumann condition for the velocity in the lateral y -direction.

The grid used has $2400 \times 1600 \times 200$ cells respectively in the x , y and z -directions which implies a total of 768 millions of cells. Around 100 cells are located on the cylinder diameter which is a few more than with the octree grid configuration. If we resume, the octree configuration has a domain around 12 times bigger than the Cartesian domain, the number of cells is around 5 times lower than the Cartesian grid and the number of cells on the cylinder diameter is 1.18 lower than for the Cartesian configuration. The number of unknowns is not the only factor of computational time since the performance of the solver has also a significant impact. The Z -ordering used to deal with the discretization on the octree grid implies indeed matrices less well conditioned than the discretization with the Cartesian solver NaSCAR. No test has been performed with exactly the same domain, same grid, same flow and same computational machine but we assume that the Cartesian solver is around 3 times faster to solve the Navier-Stokes equations than the octree solver. For this case, for example, the machines used for the simulations were not the same and especially the CPUs had different clock frequencies. The simulations on the Cartesian mesh with NaSCAR have been performed with the supercomputer Turing².

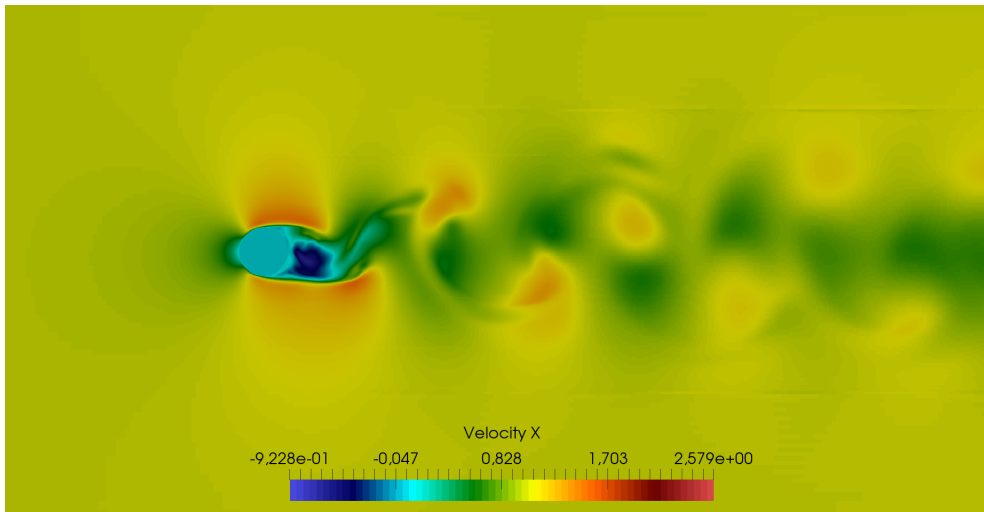
The first comparison performed is visual and can be seen in figure 3.16. It can be seen that small structures of turbulence are captured with the high-order Cartesian code, whereas the octree code can't resolve that. But the flow patterns seem however to be correct with the octree code. Further comparisons have been performed by comparing both codes with data from literature. The averaged drag coefficient has been computed for both simulations and compared with data from literature. This can be observed on table 3.11. If looking at the experimental value and at the most recent simulation performed with LES done by Kim (2006) [44], the result obtained with the octree code is slightly more accurate than with the Cartesian code. Both results are in line with the data from literature.

Other comparisons have been performed to evaluate the accuracy of the solvers in the wake of the cylinder. For this purpose, experimental results from Cantwell (1983) [16] have been used. This has been computed after preliminary simulations in both the Cartesian and the octree codes to remove the initial transient. The following results have been averaged over 5 vortex shedding (corresponding to around 20 seconds of simulations) for both the Cartesian and the octree code. Indeed the computational time needed was very high for the Cartesian case, therefore it has been chosen to perform the comparison on the same number of vortex shedding. The results can be observed in figure 3.17. At the location $x = 1$ behind the cylinder, the results from Cartesian code are closer to the experimental values. However, farer from the cylinder, the results from the octree code seems to better represent the experimental work. It was not possible to find the number of vortex shedding used to perform the mean experimental wake profiles but we expect that a very large time has been used. We know, that averaging on

²<http://www.idris.fr/>



(a) Cartesian code



(b) Octree code

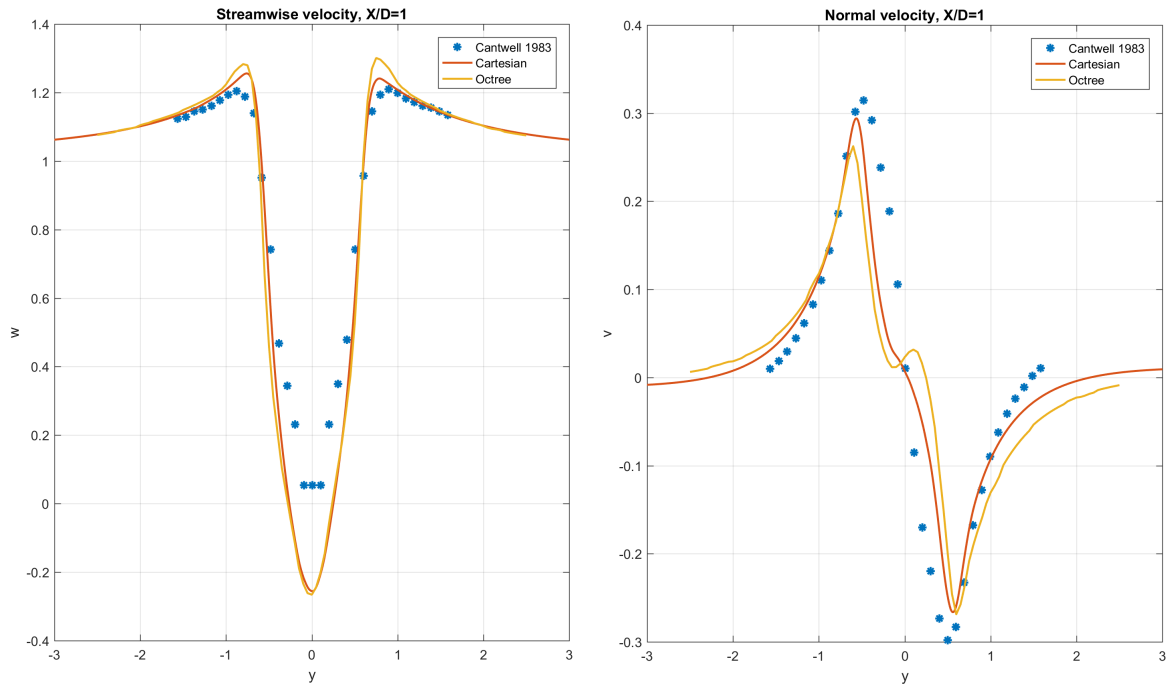
Figure 3.16: Comparison of x-Velocity field obtained with the Cartesian and the octree codes for the flow around a cylinder at $Re = 140000$

Re	<i>Authors</i>	C_D
140 000	PANS of Luo et al. (2014) [52]	0.82 - 1.28
	DES of Luo et al. (2014) [52]	0.84
	DES of Travin et al (2000) [88]	1.08
	LES of Breuer (1998) [13]	0.971 - 1.454
	LES of Kim (2006) [44]	1.21
	Experiments of Cantwell et al. (1983) [16]	1.237
	WMLES, Cartesian	1.0248
	Present work, WMLES octree	1.1846

Table 3.11: Comparison of averaged drag coefficient (over 9 vortex sheddings) for the flow past a cylinder at $Re = 140000$

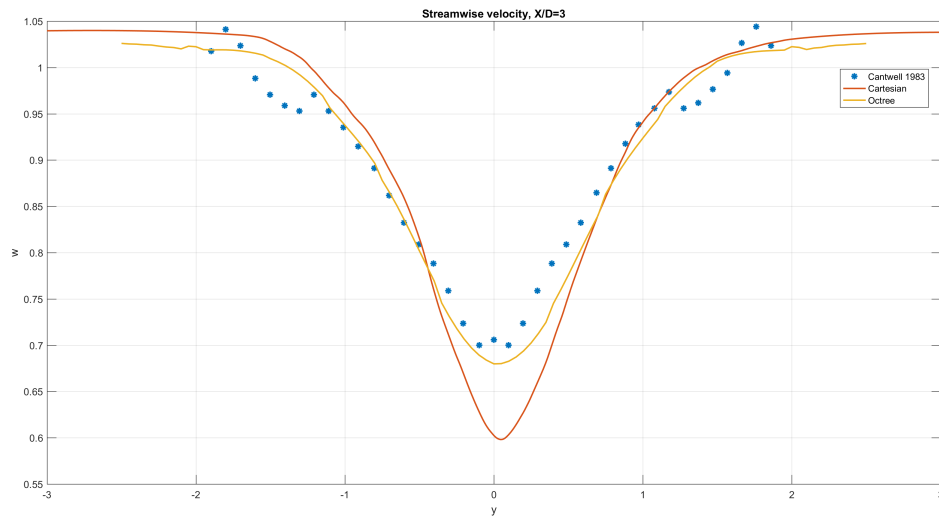
5 vortex shedding is very little. Therefore, for the octree solver, other simulations have been performed to see if the wake profile remains stable. The results obtained after 5 other vortex shedding can be seen in figure 3.18. It can be observed that the streamwise velocity at $x = 1$ does not change much. However, at $x = 3$, the simulated results are farer from the experimental one, while staying in-line, especially compared to the results seen previously from the Cartesian solver. The normal velocity shows a significant difference between the two different averages. We don't know yet how to explain this behaviour, thus, an other simulation has been launched with the Cartesian solver with another 5 vortex shedding to see if the same behaviour occurs. In the mean time, a very long simulation is performed with the octree code in order to average with a high number of vortex shedding. Perhaps this effect is due to a long period oscillations and necessitates a long time averaging to be compensated.

Finally, the code developed allows to get an interesting compromise between computational time and accuracy. The benchmark of test cases well described in literature allows to show that with the octree solver accurate results can be obtained when dealing with a reasonable number of degrees of freedom. The cylinder case has involved all the numerical models developed. The constraint of having a cubic domain prevents us to have a grid fine enough in the wake. Thus, the results obtained are not as accurate as a high-order solver, especially the Cartesian code developed in-house that is able to catch the small structures of turbulence. But this chapter allowed to validate the developed numerical tool with non-trivial test cases at high Reynolds number flows. The implementation of the application of our solver to the case of a rotating wind turbine will be described in the final section.



(a) Streamwise velocity at $x = 1$

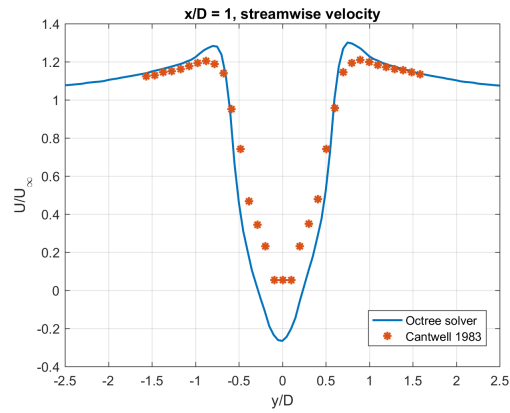
(b) Normal velocity at $x = 1$



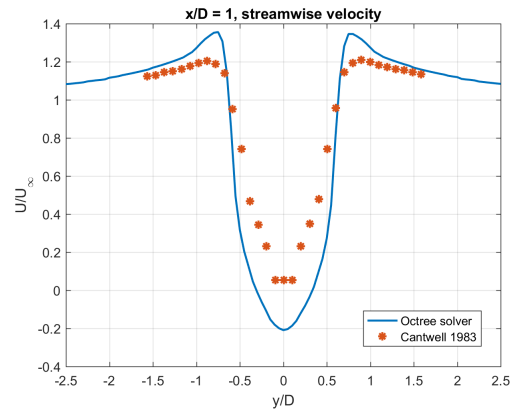
(c) Streamwise velocity at $x = 3$

Figure 3.17: Wake profile for the flow past a cylinder at $Re = 140000$ at different locations obtained by an average over 5 vortex shedding after a preliminary simulation

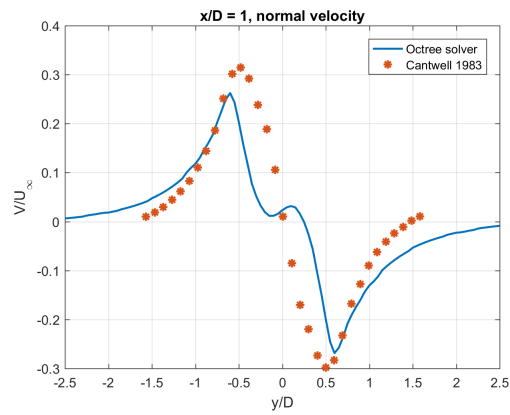
5. Comparison between Cartesian and Octree Solvers



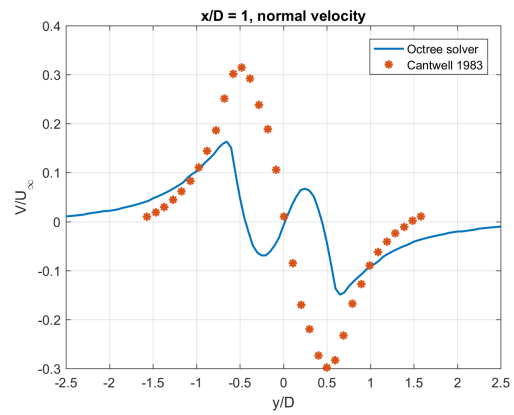
(a) Streamwise velocity at $x = 1$, first 5 vortex shedding



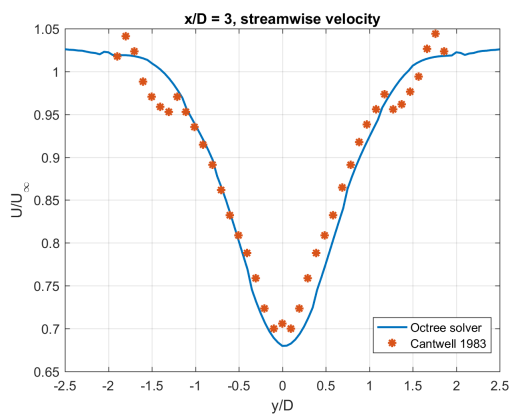
(b) Streamwise velocity at $x = 1$, next 5 vortex shedding



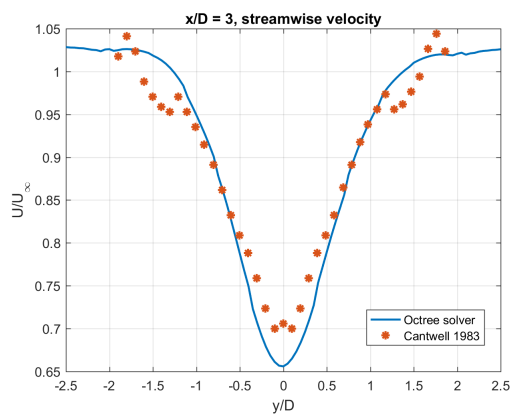
(c) Normal velocity at $x = 1$, first 5 vortex shedding



(d) Normal velocity at $x = 1$, next 5 vortex shedding



(e) Streamwise velocity at $x = 3$, first 5 vortex shedding



(f) Streamwise velocity at $x = 3$, next 5 vortex shedding

Figure 3.18: Wake profile for the flow past a cylinder at $Re = 140000$ at different locations obtained after a preliminary simulation and by an average over the 5 first vortex shedding (a, c and e), and over 5 next vortex shedding (b, d and f)

Chapter 4

Collection and Analysis of Operating Wind Turbine Data

Wind shear and gusts are both wind situations that create the largest loads on wind turbines. Indeed, the magnitude of a gust can be sufficiently high to increase the loads up to the point of fracture. Catastrophic failures are reported in the literature for both wind turbine blades and towers (see for example [23] and [22]).

Thus, the task of collection of experimental data on an in-service wind turbine has been planned in the context of the European funded project AEROGUST (More informations on www.aerogust.eu). Interests will be first to have real data and to use it to better understand the effects of wind and more precisely of gusts, on wind blades. A second interest is to use experimental data to calibrate our numerical schemes in the high-fidelity CFD code described above. Indeed, very little experimental data of wind turbines is accessible which includes the geometry, structural model and measurements on blade (pressure, strain). Further, none of the available data correlate these measurements with upstream gusts coming on to the turbines, to the best of our knowledge. As an extension for future application, wind turbine aeroelastic ROMs (Reduced Order Models) could be created from a combination of high fidelity CFD and data collected from an in-service turbine. The initial plan (before the beginning of AEROGUST) was to obtain experimental data from wind tunnel measurements but the costs were prohibitive. As Valorem Group owns and operates wind farms and has many contacts with French wind farm owners, the idea of obtaining experimental data from an in-service wind turbine was deemed more viable.

The motivations of this experimental work are thus to:

- Analyse effects of wind on blade loads.
- Compare experimental data with those obtained with our CFD code.

A requirement for the choice of the wind turbine to measure, is that the geometry and the structural model of the blades are known. This is needed to meaningfully compare experimental measurements and numerical simulations. As Valorem only has those data for one wind blade model (APX48 model), which is not very common, the choice of the wind turbine to measure was limited. The Valorem Group, on the other hand, doesn't own this kind of wind turbine but has already worked in partnership with a wind plant owner having Jeumont J48 wind turbines (a model with APX48 blades). APX48 was therefore the blade model of choice for this project.

On the following sections, the work done by Valorem and INRIA to specify the data to be

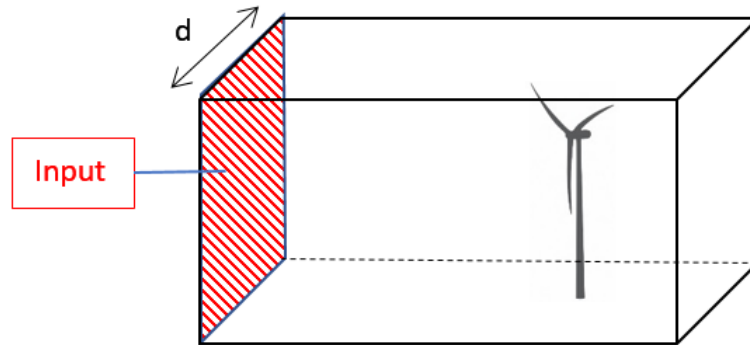


Figure 4.1: Computational domain with $d = \text{rotor diameter} + \text{a few meters}$

measured on a wind blade as well as the measurements of upstream wind is reported. A description of the technical solutions allowing to achieve this, is done. Then, the on-site sensor's installation process is explained and finally, the analysis of the data is described.

1 Specifications

Global specifications for the experimental work are first to use a Jeumont J48 wind turbine because this is the only model on which we have access to geometrical and structural data. Some characteristics of this wind turbine are a hub height of 46 m; a rotor diameter of 48 m; and blades measuring 23.5 m long. Then, the duration of these measures should allow obtaining enough wind data, which should include several strong wind conditions. So, the planned period of measurements will be between 4 and 7 months. Indeed, when the specifications were written, we had no idea of the cost and of the resistance along time of future sensors on the wind blade. In order to reach the previously explained objectives of this experimental work, discussions were held with INRIA to enable identifying the data of interest to measure.

1.1 Wind Measurements

The measurement of the wind was considered as the most important data to be obtained from the very start of the work. Indeed, this data will be used as a key input for the numerical simulations. This is needed to represent the wind as it arrives at the wind turbine. Figure 4.1 shows an example of computational domain for numerical simulations. Thus, the measurement should be done in a wind sector with a high occurrence in the wind rose of the site. In order to have representative data of the wind arriving at the turbine, no perturbations from the terrain should occur between the measuring tool and the wind turbine. So, the terrain should be quite flat and without many trees and obstacles. Installing the measuring tool in the wake of another wind turbine should also be avoided.

Another requirement is to enable estimation of the vertical wind profile. Indeed, the wind velocity profile in the vicinity of the ground generally shows a strong gradient (in the direction normal to the ground). This phenomenon is known as the atmospheric boundary layer and depends on many parameters, such as the roughness of the surface, the weather conditions (pressure and temperature) and also thermal stability changes between day and night. So, several measurement points are needed at different heights. In order for the future numerical simulations to be accurate and similar to the real experiment, the turbulent flow should be measured.

Moreover, the wind during a gust phenomenon is characterized by high variations in wind velocities in a few seconds as explained in the IEC standard [1]. Thus, the acquisition frequency of wind data should be small enough to accurately characterise these changes.

1.2 Blade Data

The main goal of the wind turbine data collection is to observe the aero-elastic behaviour of wind blades. The decision was made to collect data from the maximum number of measurement points, given the budget in the AEROGUST project. Thus, it was decided to fully instrument one wind turbine blade (instead of all three blades).

A global requirement on instrumentation of the wind blade is, first, to minimize the impact of measurement tools on the wind blade. Indeed, the wind turbine is supposed to keep working well after the experimental phase. So, for example, no hole is allowed to be made. Then, the Valorem company does not have the resources to develop an inquiry system to get data back from sensors to the office. Thus, we looked for a complete technological solution with regards sensors supply, installation, maintenance and communication system – via a GSM modem to collect the data. In order to have experimental data showing the aero-elastic behaviour of a wind blade, the measurement of the blade deformations will allow to check the structural beam model of the blade and to quantify its structural behaviour. To observe the aerodynamic load on the wind blade, the measurement of pressure of air on the blade will be of significant interest.

The aim of the blade deformation data will be, first, to check that the beam model used in the numerical simulations is in accordance with the real deformations of the blade and second, to observe the effects of the wind on the blade deformations. Strain gauges are considered for this measurement of blade deformations. For the purpose of validating the beam model of the blade, a requirement is to put several strain gauges on the blade root, since the strain at the root is the sum of all strains along the blade. The measurement of experimental pressure data will be used for comparison between experimental data and numerical simulations.

Moreover, we realized that the correlation of the wind on the blade and the analysis of the blade deformation will need to be spatially synchronised. Indeed, as this will be the first access of such structural data, the different kinds of deformation that the blade undergoes in reality is to date relatively unknown, while being related to its position on the rotation circle of the wind turbine (for example, when the blade passes along the tower, the behaviour of the deformation is probably not the same as when the blade is clear of the tower). So, another requirement for the experimental work is to sense the position of the blade relative to the tower (rotational position).

The operating range encountered by the sensors necessitated writing specifications to our suppliers. For the blade deformation, data from the blade manufacturer was used to determine the expected range of deformation. Thus, the local deformations of the blade are of the order of -1.5 % to 1.5 % and the tip deflection of the blade can reach 2 m. For the pressure, the Bernoulli equation was used and applied to the approximation of the mean air density and expected normal wind velocity range. So, the operating range of the pressure sensors is of the order of $1/10^{th}$ of the atmospheric pressure (on average 104 Pa). The positioning of the sensors was not decided during the specification step, since it was unclear how many measurement points the budget would allow.

A requirement was written to specify the targeted accuracy of experimental data. Indeed,

the numerical results from our computational code are expected to have an error of the order of 5 %, so a similar accuracy is required from the experimental results. The final requirement for all the blade data is to have the same acquisition frequency as the wind data in order to ease the synchronisation and the correlation between both kinds of experimental data.

Now, the manner in which the technical solutions could meet the specifications will be explained in the next section.

2 Technical Solutions for Measurements

2.1 Meteorological Mast

The aim is to have the most accurate measures and to be in accordance with the budget. Valorem has experimented 3 different ways to measure wind velocity and turbulence at different heights. The measurement with a meteorological mast, with a SODAR (SOmic Detection And Ranging) which measures the scattering of sound waves or with a LIDAR (LIght Detection And Ranging) which measures the reflected pulses of a laser light with a sensor. Valorem owns several meteorological masts and Sodars and the cost of a Lidar is very high. The IEC 61400-12-1 standard [1] is the reference for wind assessment studies. The standard prescribes the use of a meteorological mast in such studies. Moreover, the Sodar has a maximum acquisition frequency of several seconds which is not adequate for our specifications. That's why the final choice was the use of a meteorological mast (met mast).

Valorem had a 50m height meteorological mast available. As the height of the hub of the wind turbine is 46 m, an extrapolation of the wind to the top of the blades will be done thanks to the measured wind profile. All the sensors will be oriented to minimize the wake effect from the mast according to the IEC 61400-12-1 standard. According to this same standard the met mast should be located between 2D and 4D (D being the turbine rotor diameter) upstream from the wind turbine, ideally at 2.5D, which corresponds to approximately 120m.

Then, work has been done to define the setting-up of the met mast. Indeed, requirements are to (a) find a flat terrain without perturbations (trees, wakes of other wind turbines...) between the wind turbine and the mast in the main direction of wind; which (b) the owner of that terrain accepts grant for this purpose. A wind rose defines the main wind directions. It has been obtained thanks to a wind measurement campaign by Valorem in another terrain located a few kilometres away of our site. As can be seen in figure 4.2, the wind plant is located on a hill so the met mast should also be on the same height. Indeed, putting the met mast upstream from wind turbines n°4 and 8 should be avoided because of the wake of other turbines. Turbines n°1, 2 and 7 have strong terrain variations so are not good choices either. This leaves wind turbines 3, 5 and 6 as possible options.

Going forward, a site-inspection was conducted to confirm the possibility for the installation of the met mast and to investigate whether the owners of fields would agree. Compensation was planned for owners and farmers of the terrain where the met mast would be placed. But as wind turbines are not always well accepted in France, it was not easy to obtain an agreement from the owners to install the met mast. Finally, a terrain was found upstream from wind turbine n°5. The next step was then to obtain an administrative permit for the setting up of the met mast as will be explained in the next section

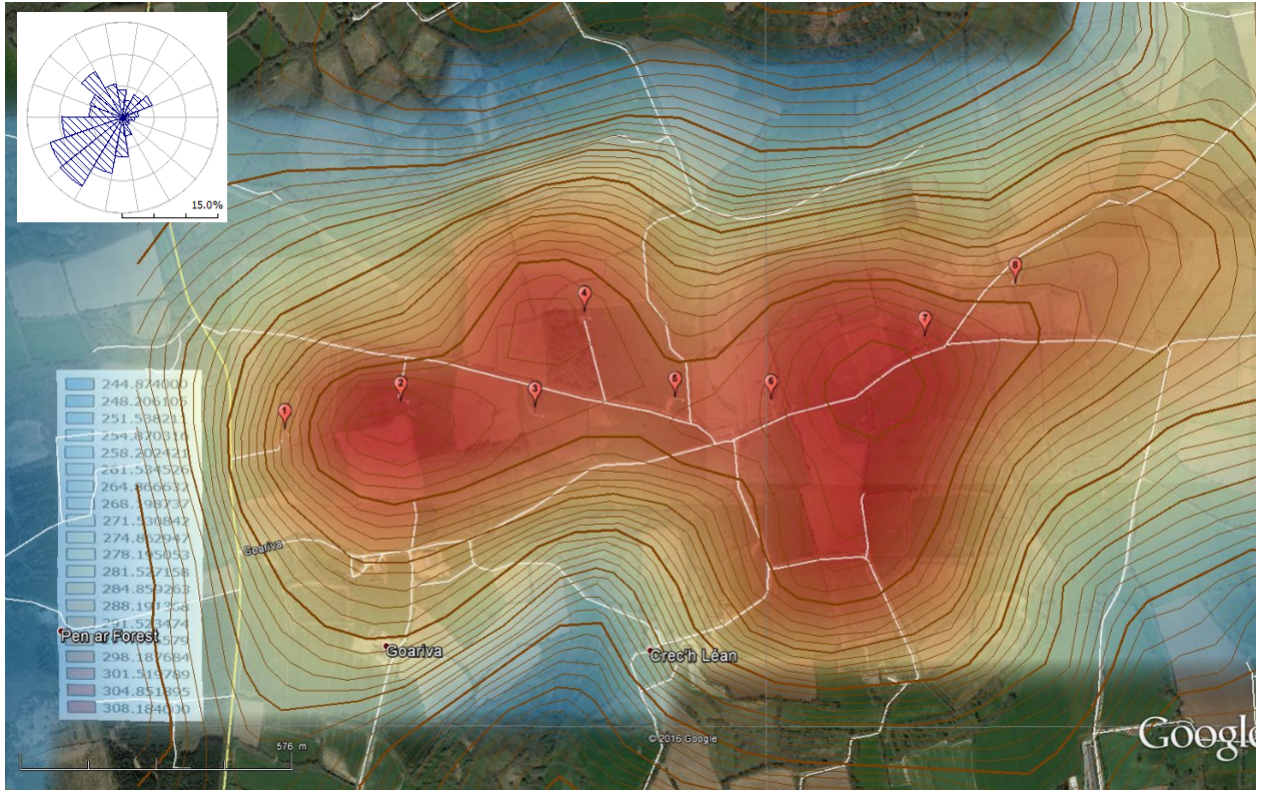


Figure 4.2: Topographic view of the wind plant obtained with Google Earth [®] and wind rose corresponding to the site.

2.2 Blade Measurements

The specifications for the blade measures have been sent to several companies. The only viable solution was suggested by a company named Febus Optics, based in Pau (France). They suggest the use of optical fibre with Fiber Bragg Gratings (FBG) sensors. Achieved by creating a periodic variation in the refractive index of the fibre core, FBG are short segments of an optical fibre that reflect particular wavelengths of light and transmits all others as can be seen in figure 4.3. This wavelength can be measured by injecting an incident light having a broad optical spectrum and analysing the wavelength of the reflected light with an OSA (Optical Spectrum Analyser). It can be used as a sensor because any change in the period of the refractive index will induce a change in the specific wavelength (so-called Bragg wavelength) reflected by the FBG. Thus, strain or temperature will induce a measurable wavelength change. Many types of sensors can be addressed by this technology as soon as the body of the sensor converts the measure in wavelength change in the FBG.

Moreover, using different Bragg gratings with different wavelengths, it is possible to connect several sensors inline. That means that every sensor will be connected to the others in the fibre and the different fibres will then be connected to an inquiry system located inside the rotor. More precisely, on the rotating part of the rotor, so that no problem occurs with the fibre arriving in the hub (it would be very complicated to pass from the rotating blade to the fixed part of the rotor). The company also suggested the installation of an inclinometer inside the inquiry system to measure the angular position of the instrumented blade. Figure 4.4 shows an illustration of the instrumentation of the blade with this technology.

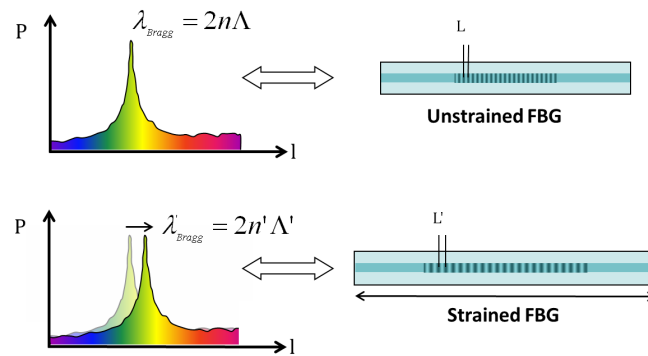


Figure 4.3: Fiber Bragg Grating principle. n corresponds to the index of refraction and Λ is the period of n variation. Found on www.fbgs.com.

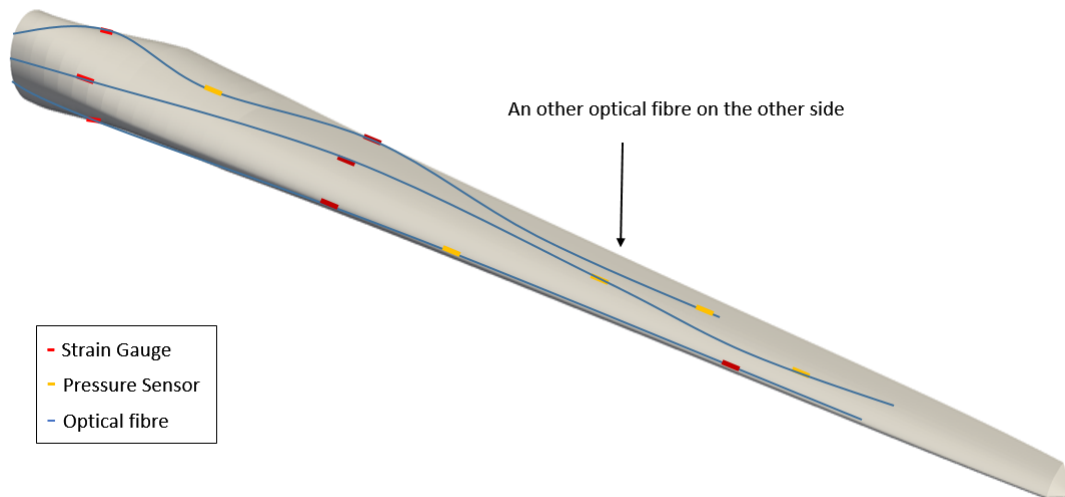


Figure 4.4: Optical fibre technology

This technological solution is very non-intrusive since all the sensors and the fibres would be stuck to the blade surface. The fibre is also flexible which allows a wide range of sensor locations. The inquiry system will collect the data and send them via 3G to the office. Finally, this technological solution fits all the specifications and the cost is in accordance with the budget. This was therefore the technological solution of choice. Febus Optics works with another French company, based in Brittany (France) close to the wind plant, which manufactures the optical fibres, pressure sensors and strain gauges.

2.3 Position of Sensors on the Wind Blade

Concerning the number of sensors (for pressure and strain) to put on the blade, the main constraint was the budget. Indeed, the more sensors installed, the better our experimental work. Strain gauges were less expensive than pressure sensors, so more of these were chosen. This allowed the purchase of 10 strain gauges and 6 pressure sensors.

For strain gauges, as explained previously, one of the goals is to check the beam model. So, strain gauges should be put on the blade root, where the strain is the sum of all the strains on the blade. It was decided to place 4 strain gauges every 90° at the cylindrical blade root. Then,

3. On-site Installation

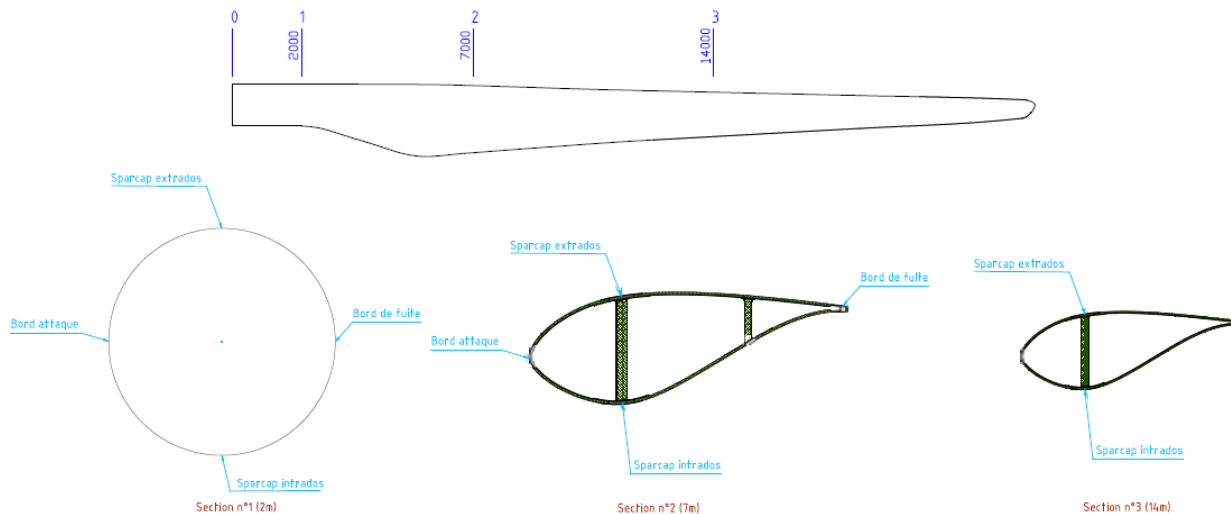


Figure 4.5: Position of the strain gauges on the blade

to optimize the use of the remaining strain gauges, their positions were chosen to be where only longitudinal strains occurs. Indeed, if we would like to measure the strains in several directions, several strain gauges would be necessary at each location. The other locations have been chosen to have a kind of equipartition along the length of the blade. The locations of the strain gauges can be seen on figure 4.5. At section n°2 we put the gauges at the same locations as the blade root i.e. at the leading edge, trailing edge, and on the sparcaps at the pressure and suction sides. At section n°3, two strain gauges were located on the sparcaps at the pressure and suction sides.

As we only had 6 available pressure sensors, their locations were chosen in order to obtain a pressure profile around a section, and in a manner similar to experiments conducted by the NREL (National Renewable Energy Laboratory) on a small wind turbine [78]. The sections where the pressure sensors would be placed was chosen as a compromise between two requirements. First, the sensors should be located far from the nacelle since the nacelle geometry is not well known and can't be reproduced accurately in our simulation. Then, we cannot put the sensors where the chord is too small, otherwise the geometric perturbations induced by the sensor on the airfoil would become significant (the sensor has a thickness of 1 cm, a length of 5 cm and a width of 2 cm). The chosen location for the pressure sensors can be seen on figure 4.6.

In the next section, work done before and during the installation on site of the met mast and of the sensors on the blade will be reported.

3 On-site Installation

3.1 Meteorological Mast Installation

Prior to the met mast installation, some administrative tasks should be performed. First, the civil aviation services must be consulted and should agree. Then, an administrative permit (named "preliminary declaration of work") should also be delivered by the city council. This file should include geographic location of the future met mast, photographs of the mast in its environment, drawings of the mast with its dimensions (2D and 3D) and how long the mast

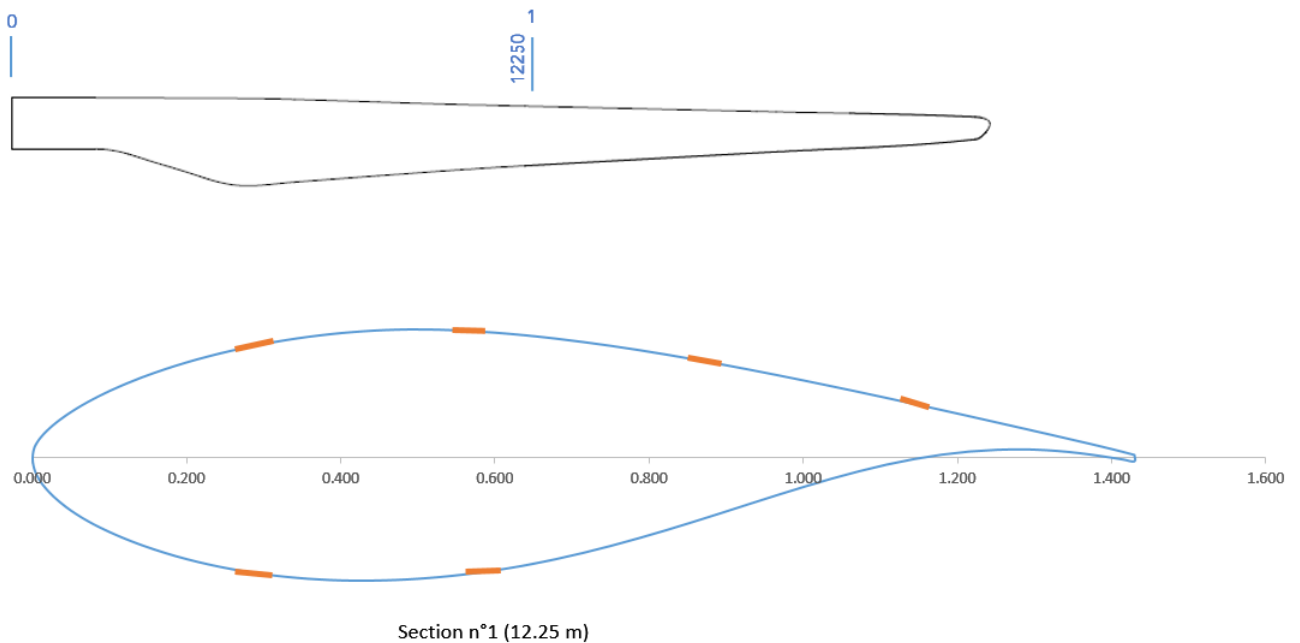


Figure 4.6: Position of pressure sensors on the blade

will stay. The agreement of the civil aviation should be included in the file. Once this administrative permit has been obtained, the military aviation should also be consulted for agreement. The final step is to submit a DICT (Declaration of the intention to begin work). It consists of checking that the future mast will not impact on power lines, gas pipes, water network, etc. All this was subsequently done by Valorem for the project.

The final position of the met mast can be observed on figure 4.7 and it can be observed thanks to the wind rose that the wind turbine and the met mast are aligned with the main wind direction occurrence on the site.

A brief description of the installation process for the met mast follows:

- The mast is composed of several tubes that fit together on the ground.
- 8 anchoring points are made in 4 orthogonal directions (in order to have a high stability of the mast).
- Installation of wind vanes, cup anemometers, logger, photovoltaic panels, nocturnal runway lights is done on the ground.
- The mast is raised thanks to a motorised winch.
- Adjustments are made on the guide wires so that the mast stands vertical and straight.

The certification of the cup anemometers is done thanks to wind tunnel measurements. The calibration laboratory certified that the value of the measured wind lies within the assigned range of values with a probability of 95%.

Figure 4.8 shows a photograph of the whole mast after its installation. Figure 4.9 presents the sensors of the met mast with anemometers and wind vanes.

3. On-site Installation



Figure 4.7: Final position of the met mast and wind rose of the area.



Figure 4.8: Photo of the met mast after its installation



Figure 4.9: Photo of the sensors on the met mast

3.2 Sensors on Blade

For on-site installation of the blade sensors two choices exist: using steeplejacks or an aerial work platform. Although an aerial work platform has a significantly higher cost, the time needed with steeplejacks for installation would be too long and so too complicated to implement the sensors on the blade.

The supplier of the sensors had no previous experience in wind blade instrumentation. Thus, preliminary checks were done. Indeed, Valorem has available nearby its offices, a 48m long wind blade of the same model than the blade that will be instrumented. The blade is located on the ground. A test of implementation of sensors on the blade was therefore done at Valorem before the real aerial installation. This test revealed that the use of the high adhesive tapes as firstly planed wasn't a good option. Indeed, thanks to the previous experience from Valorem, the fact that a step exists between the adhesive tape and the blade, renders the surface not smooth enough, and is likely to cause a problem. The blade is submitted to significant load and apparent wind, so, these kinds of steps with an obstacle on the blade could lead to the loosening of the tape

The methodology suggested by the supplier was then the following:

- Sanding of the painting of the blade where the sensors and the optical fibres will be located from the top (the nacelle) to the bottom along the blade.
- Cleaning of this surface with acetone.
- Passing the fibre through the hub of the wind turbine for its connection to the inquiry system.
- Positioning of the fibre and/or sensor and gluing with epoxy glue.
- Protecting with low adhesion wove tape.
- Painting with protecting polyurethane paint designed for wind blades.



Figure 4.10: Aerial work platform positioning

Once the met mast has been installed and it has been checked that the wind data was correctly collected, the installation of the sensors on the blade began. Necessary conditions for installation are complete absence of both rain and strong wind. Indeed, rain is a problem for gluing with epoxy and a strong wind prevents the use of the aerial work platform. The prevailing weather was therefore monitored prior beginning the installation. We had to be sure that as soon as we began the implementation, we would be able to finish it. Finally, a half day delay occurred because of strong wind and the whole installation lasted almost 3 days. A view of the work on the aerial platform can be seen on figure [4.10](#).

In figure [4.11](#), the 2 lines of sensors going along the pressure side and the leading edge of the wind blade are shown. The setup of the 4 cables is:

- Pressure side: 3 strain gauges, 1 pressure sensor.
- Leading edge: 2 strain gauges, 2 pressure sensors.
- Suction side: 3 strain gauges, 1 pressure sensor.
- Trailing edge: 2 strain gauges, 2 pressure sensors.

As a reminder, the box containing the inquiry system is located on the rotating part of the rotor and contains an inclinometer to measure the angle of the blade.

Finally, organization of the on-site installations was successfully conducted for both the met mast and the sensors on the blade. Work has then be done with the experimental data in order to identify different gust conditions in the field and to analyse the effects on the blade deformations.



Figure 4.11: Photo of the pressure side of the wind blade after installation of sensors

4 Analysis of Experimental Results

The measurement campaign lasted 9 months from April 2017 to January 2018. At the end, a huge number of experimental data have been collected. Considering the wind speed measurements, 26 939 646 data points exist for each 5 anemometers. For the blade data, 11 806 822 measurements are available for each of the 16 sensors (10 deformation data and 6 pressure data). This corresponds to an availability for the meteorological mast of 88 % and 43 % for the blade's sensors. Indeed, several issues arose along the campaign that implied some data loss. The methodology used will so be described and then the most important outcomes of the study will be explained.

4.1 Methodology

Considering the amount of data, the first step in the analysis of this data was filtering to ease the handling of data. Indeed, the wind speed measurement should be representative of the wind arriving on the turbine so the wind direction should be lined up with the met mast and the wind turbine. Direction of the wind has so been filtered at $\pm 10^\circ$ around the optimal direction. We also wanted to have a rotational speed of the rotor representing a normal behaviour of the wind turbine. That means for example to avoid the start of the turbine. Therefore, the blade rotational speed has been filtered to keep only the value between 11 and 25 rounds per minute (25 rpm corresponding to the maximum rotational speed of the turbine).

Once, this work of filtering was performed, the first idea was to gather all the remaining data along time and to see if a correlation between wind speed and sensor's measurement could be

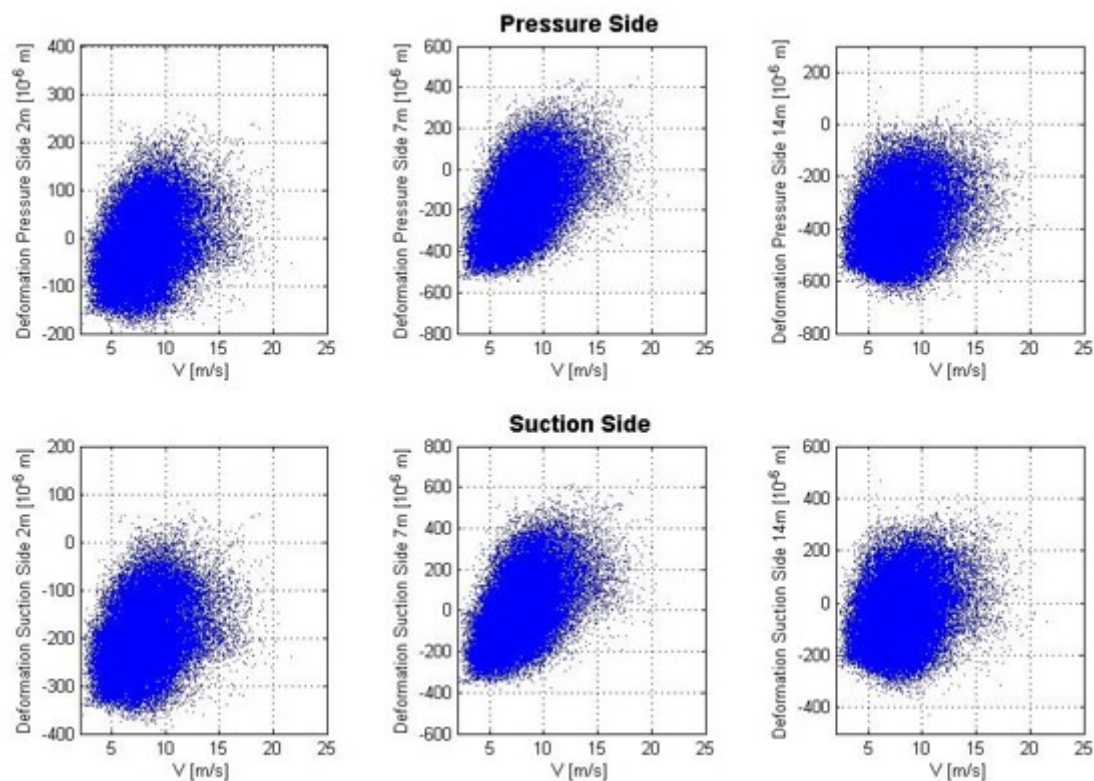


Figure 4.12: Longitudinal deformations from sensors at different locations as a function of wind speed

observed. Figure 4.12 shows the deformation data from the strain gauges located at pressure and suction side of the blade as a function of wind speed. The influence of the wind speed intensity in the values of deformations can be seen but no significant correlation has been highlighted with this methodology. Therefore, we change our strategy for the analysis of experimental results.

Another idea was to find a methodology to identify the highest deformations occurring on the blade and look then how it could be related to the wind speed. Indeed, we are finally interested in knowing what produces the highest loads that exerted on a wind blade. However we didn't find an adequate strategy to do that, since the strain gauges data show high variations along time whatever the wind is. Also, at Valorem, we are not specialists enough to characterize the deformations that would imply high loads on the blade. The choice was thus to focus on wind data first and to identify gust events.

Identification of "gust events"

The IEC standard [1] defines a theoretical gust as a high variation with a Gaussian shape of wind speed on a period of several seconds (around 10 s). Therefore, the basic idea tested is to compute a moving average of wind speed on 10 s period. Then, on each period, the computation of the variation of wind speed is done ($V_{max} - V_{min}$). A second method has also been tested, that consisted in using definitions of the "Extreme Operating Gust" (EOG) in the IEC standard. This concept defines an extreme wind condition due to a gust and an empirical equation exist to compute a value of the EOG from 10 minutes averaged wind speed data (and other parameter as the turbulence, the diameter of the rotor, ...). Using this equation, Extreme Operating Gusts have been computed from 10 minutes averaged wind speed data, which gave us a curve with a length of 10 seconds describing the EOG during each 10 minutes. A comparison of this theo-


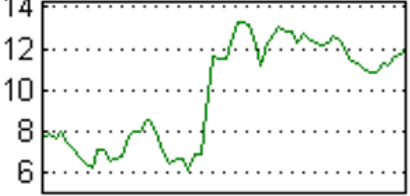
	Category 1	Category 2
Number of gust events	28	10
Example	<p>Gust : 08-Sep-2017 00:08:18</p> 	<p>Gust : 08-Sep-2017 09:56:25</p> 

Table 4.1: Characterization of gust events

retical gust to the 1 second experimental wind speed data is then done to find the correlation between the two. The two different methods have been tested on 1 month of experimental data and the method based on the IEC standard (i.e. based on the EOG definition) showed poor correlation coefficients. Indeed, we analysed after, that this equation defines an extreme event but we had a few chance that such an extreme event occurred during our experimental work. We focused thus on the first method (based on the variation of wind speed). It was noticed that, only doing $V_{max} - V_{min}$ only finds gusts with high wind speeds, whereas normalizing with the minimum velocities $\frac{V_{max} - V_{min}}{V_{min}}$ keeps high variations whatever the velocity is. These two variants of the method have been used, since we are interested in every kind of gust event.

This methodology has been applied to the whole period of available experimental data. The threshold used for our parameter to define whether the variation corresponds to a gust or not, has been chosen so that, at the end, only a few events are left. Indeed we had no expertise to define such a value otherwise. Finally we identified 38 gust events on 1 minute periods. It was noticed that the gusts can be characterized and classified into two different categories. We observed that two different behaviours of the gust event could occur, as summarized in table 4.1. On the one hand, we can see gusts that consist in a high increase of wind speed followed by a decrease to come back to the initial value of wind (category 1). This category shows gust events quite similar from what is described in the IEC standard. On the other hand, we can also observe a high increase of wind speed but then, no decrease occurs right after (category 2).

Comparison of deformation data between a gust event and with weak wind

A test has been done to compare the deformation data during a gust and with weak wind in order to validate that different behaviours occurred. First-of-all it was needed to reconstruct the time series of wind speed data measured at the meteorological mast so that we can evaluate when the gusts arrive on the wind turbine. This was done basically by transporting the wind speed along time. It implies of course uncertainties on the wind data but it gives an idea when the gust arrives at the turbine. The different behaviours of the deformation data during a gust or a period with weak wind can be seen in figure 4.13 with an example for one strain gauge located at the leading edge at 7 m from the blade root. When the wind is weak, it can be observed that the oscillations defining the deformations of the blade have a regular amplitude and frequency. The frequency has been computed with the time between two successive oscillations. However, during a gust, variation of the frequency of deformation can be seen around the gust for the sensors located at the leading edge of the blade. The sensors located at the leading and trailing

4. Analysis of Experimental Results

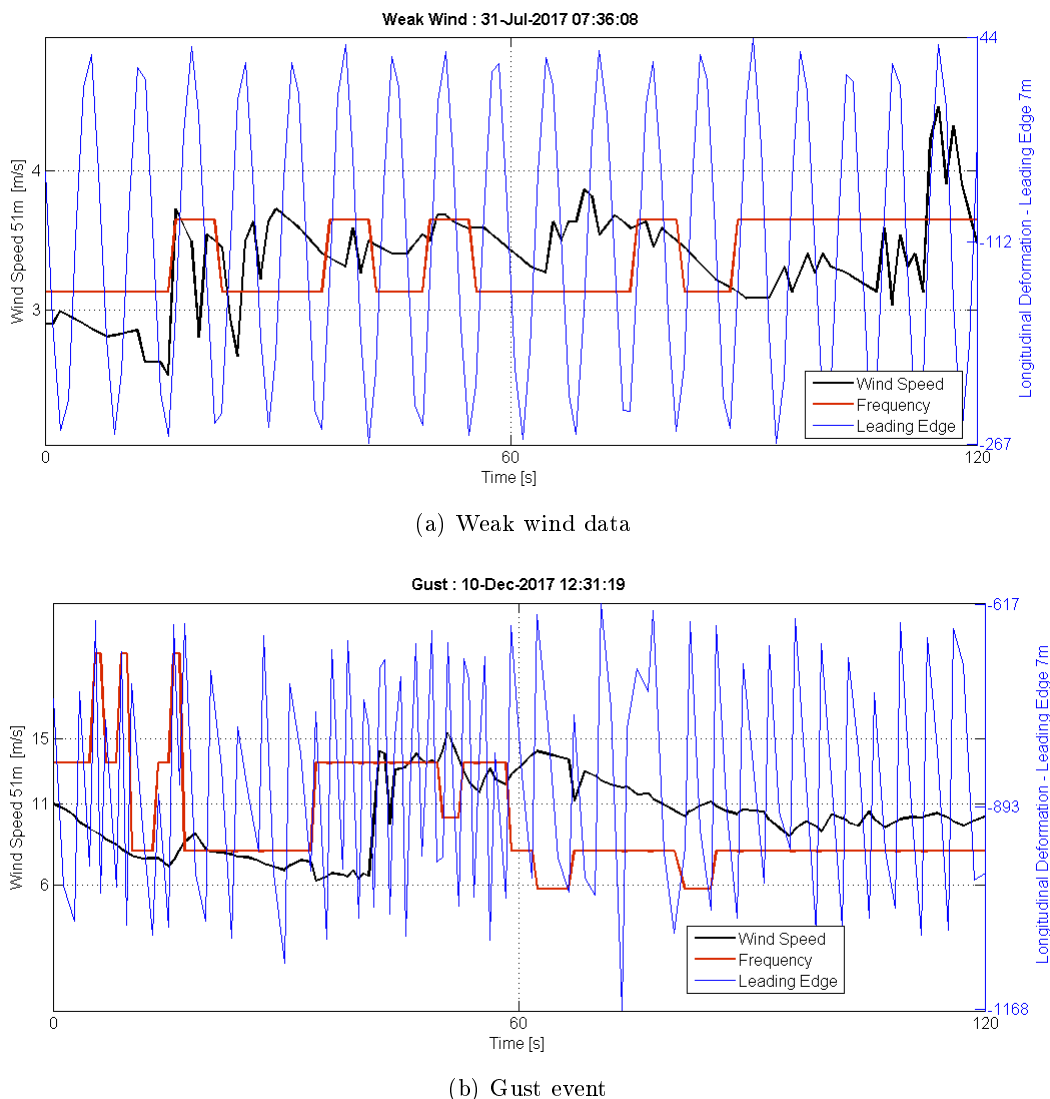


Figure 4.13: Time history of deformation data from sensor located on the leading edge at 7m of the blade length, of wind speed data reconstructed at the wind turbine and frequency (with a normalized scale) during a gust and with weak wind.

edges show similar results, and, in a same way for the sensors located at the pressure side and suction side, which makes sense. The sensors located in the pressure and suction sides are also subjected to variation of the amplitude of data around the gust and less effected by variation of frequency. These are so first remarks made from figures plotted for several events one by one.

4.2 Synthesis of the Experimental Work

In order to synthesize the results obtained from the experimental work and to better understand the effects of gusts on the wind blade, we came back to our very first test that was to gather the data along time as a function of several parameters and see if a correlation of gusts and deformations could be seen. But, we only used data of the identified gust events and instead of the wind speed measured at the meteorological mast, we used the time series reconstruction. Therefore, as we noticed above that the frequency of the deformation seemed to increase when dealing with gusts, the mean frequency during periods of weak wind and during gusts have been

	f gusts	f weak wind	Amp gusts	Amp weak wind
Pressure side 2m	0.325	0.161	23.2	43.1
Pressure side 7m	0.332	0.163	53.3	84.0
Pressure side 14m	0.326	0.163	48.0	78.1
Suction side 2m	0.321	0.162	23.4	44.6
Suction side 7m	0.337	0.183	52.8	70.4
Suction side 14m	0.323	0.163	48.1	72.9
Trailing edge 2m	0.270	0.172	31.1	74.5
Trailing edge 7m	0.266	0.171	87.2	180.1
Leading edge 2m	0.267	0.160	28.2	30.0
Leading edge 7m	0.264	0.160	86.0	146.4

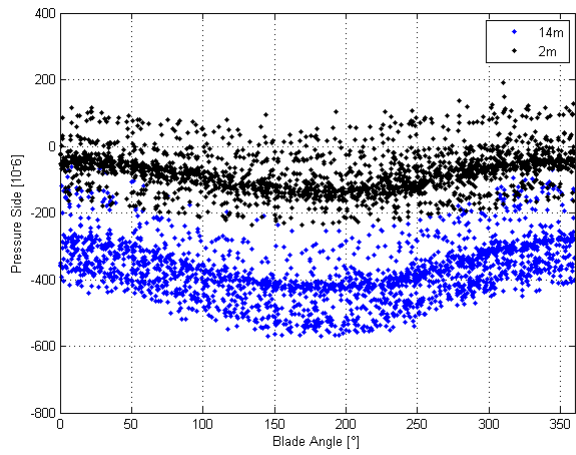
Table 4.2: Mean values of frequency (noted f) and amplitude (noted Amp) of deformations during gusts or with weak wind for each sensor

compared. Indeed, we identified periods of weak wind (same number of events as gust events) to allow a serious comparison. The results can be seen in table 4.2. We can thus confirm that the frequency of the deformation is higher for every sensor in the case of gust events. Concerning the amplitude, the analysis is more difficult to be conducted. Indeed, the mean amplitudes are higher when submitted to weak wind compared to gust. An explanation could be that, as the frequency is higher during gust events, and the acquisition frequency of the data being 1 s, it is likely that several data points at high amplitude are missing in the data collection during gusts.

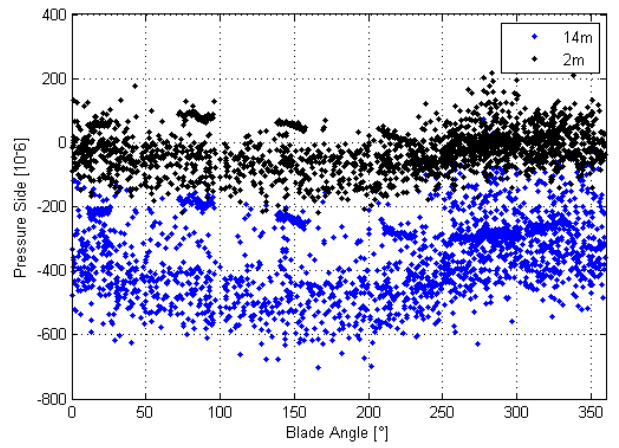
An interesting figure is the plot of the deformation as a function of the position of the blade. This can be seen in the figure 4.14 for the cases of gusts and during weak wind. The angle of 0 corresponds to the position where the blade is along the tower, the tip blade being at its closest position to the ground. The blade rotates clockwise. It can be observed that during regular and weak wind, the angle of the blade has a significant influence on the deformations located on the trailing and leading edge (same behaviour for both). This is also true at the pressure or suction side as we can see that points are less diffused than during gusts but slightly less significant. It can be remarked, that during gusts, the deformations of all the sensors seem to increase when the blade is going down. This phenomenon does not occur when the wind is weak. We can also notice, as expected, that for the sensors located at the blade root, the deformations at weak wind are very small.

Another remark is that, when plotting the deformations as a function of wind speed at the wind turbine (reconstructed from the measurements at the meteorological mast), no significant influence can be seen. The same kind of figure as the very first test (see above figure 4.12) is obtained, whereas we are only dealing with gust events. This is an unexpected result since we imagined that the wind speed intensity would have an impact on the amplitude of deformations. Another hypothesis is that, since the position of the blade has a significant impact on the deformation of the blade, we should subtract in a way this effect of the position (of the rotation of the blade, finally). Perhaps, this could lead to highlight the influence of wind speed on deformation data.

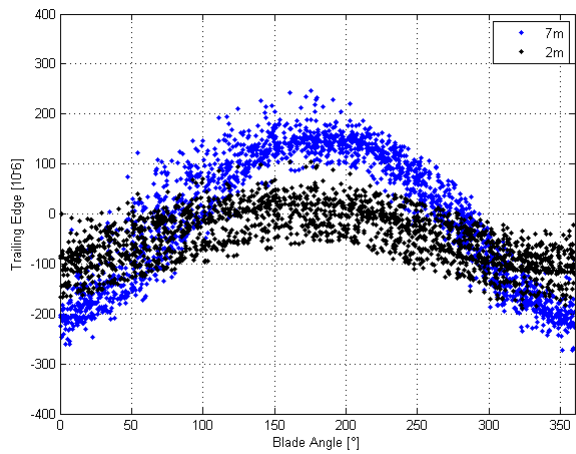
4. Analysis of Experimental Results



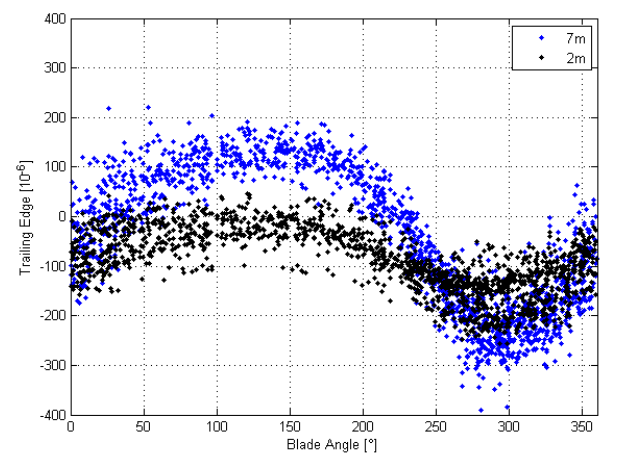
(a) Deformations at pressure side with weak wind



(b) Deformations at pressure side during gusts



(c) Deformations at trailing edge with weak wind



(d) Deformations at trailing edge during gusts

Figure 4.14: Deformations on the blade at several location as a function of blade position with weak wind (a and c) or during gust events (b and d)

Chapter 4. Collection and Analysis of Operating Wind Turbine Data

The results presented in this section constitute preliminary outcomes. Several gust events have been identified, that can be used as test cases for our numerical tool. Gusts that can occur on an operating wind farm have been characterized. This experimental work allowed a gain of experience for Valorem in terms of data treatment and knowledge on wind gusts. This analysis can be pursued in order to show the influence of wind intensity on the deformations. Indeed, the inertial effects can be subtracted from the deformation data in order to highlight the effects due to wind.

Chapter 5

Towards Application to the Study of Wind Gust Effects on a Blade

The aim of the thesis is to simulate the flow around wind turbine blades, especially when submitted to gusts. As we have available several experimental data with deformations at different locations and pressure, it is relevant to implement a structural model in our numerical tool. This can so be used to calibrate our numerical tool and to compare the data with the simulations.

Using a computational domain with the whole wind turbine and simulating the three blades, even with an octree grid, would require a huge amount a computational time. This would indeed imply a wide computational domain and fine cells around the three blades. Therefore, the approach chosen in the present work, is to simulate only one wind blade in a rotational framework. The rotational framework allows to use a significantly smaller computational domain compared to the domain that would require the simulation of the rotation of the blade. This simple approach is a base but, in the future, the whole wind turbine could be simulated with the tower and the nacelle.

This chapter describes the structural model used for the wind blade and the way it has been coupled with the Navier-Stokes solver. Then, the rotating framework and the required modifications are explained. An adaptive mesh refinement process has then been implemented and the results obtained are presented. Finally the simulations including all the numerical methods implemented are described with the results obtained.

1 Coupling with the Solid Model of the Wind Blade Apx48

1.1 Structural Model

Wind turbine blades are very slender structures with an internal configuration very similar to that of airplane wings, their structural response can therefore be described with a reasonable level of accuracy using beam models. In this work the model reported in [25] is used. The spatial discretization of the elasticity problem relies on the finite element method. The considered element possesses two nodes, each with six degrees of freedom: three displacements and three rotations.

Let X-Y-Z be the global reference system as can be seen in 5.1:

- X axis corresponds to the beam axis,
- Y axis is parallel to the lagwise movement direction (flapwise bending axis),
- Z axis is parallel to the flapwise movement directions (lagwise bending axis).

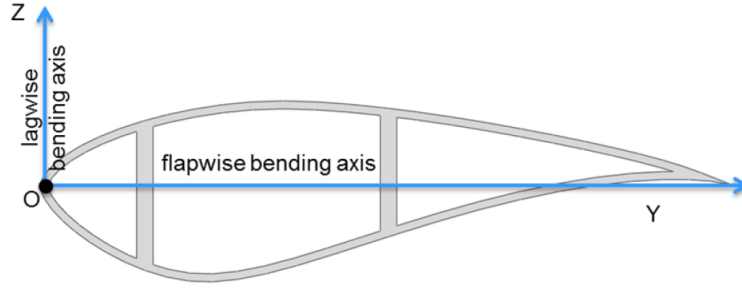


Figure 5.1: Blade cross section with suggested reference system including definition of flapwise and lagwise bending axes. Source [25].

The kinematics of the beam are described using the Euler-Bernoulli theory for the bending combined with the Saint-Venant hypothesis for the torsional motion. In the global system, the displacement field is:

$$\begin{cases} u(x, y, z) = u_t(x) + (z - z_t)\theta_y(x) - (y - y_t)\theta_z(x), \\ v(x, y, z) = v_c(x) - (z - z_c)\theta_x(x), \\ w(x, y, z) = w_c(x) + (y - y_c)\theta_x(x), \end{cases} \quad (5.1)$$

where y_t and z_t are the coordinates of the Tension center (T), y_c and z_c the coordinates of the Shear center (C), u_t is the axial displacement of T, v_c and w_c are the bending displacement of C. Let $X'-Y'-Z'$ be the reference system centred in T and parallel to the global one, the deformation field is:

$$\begin{cases} \epsilon_{xx} = u_{t,x} + (z' - z'_t)\theta_{y,x} - (y' - y'_t)\theta_{z,x}, \\ \epsilon_{xy} = -(z' - z'_c)\theta_{x,x}, \\ \epsilon_{xz} = (y' - y'_c)\theta_{x,x}. \end{cases} \quad (5.2)$$

Since modern wind turbine blades are made of glass-fibre and reinforced with lightweight carbon fibres, the stress-strain relation is that of an anisotropic materials:

$$\begin{pmatrix} \sigma_{xx} \\ \sigma_{xy} \\ \sigma_{xz} \end{pmatrix} = \begin{pmatrix} E_{11} & E_{12} & E_{13} \\ E_{21} & E_{22} & E_{23} \\ E_{31} & E_{32} & E_{33} \end{pmatrix} \begin{pmatrix} \epsilon_{xx} \\ \epsilon_{xy} \\ \epsilon_{xz} \end{pmatrix}. \quad (5.3)$$

The internal reactions are defined as:

$$\begin{cases} N = \int_A \sigma_{xx} dA, \\ M_x = \int_A [(y' - y'_c)\sigma_{xz} - (z' - z'_c)\sigma_{xy}] dA, \\ M_y = \int_A z' \sigma_{xx} dA, \\ M_z = \int_A y' \sigma_{xx} dA. \end{cases} \quad (5.4)$$

Substituting eqs. (5.2), (5.3) in (5.4) and integrating on the cross sectional area, the following system of equations is obtained:

$$\begin{pmatrix} N \\ M_x \\ M_y \\ M_z \end{pmatrix} = \begin{pmatrix} EA & AT & 0 & 0 \\ AT & GJ & FT & -LT \\ 0 & FT & EI_{y'y'} & -EI_{y'z'} \\ 0 & -LT & -EI_{y'z'} & EI_{z'z'} \end{pmatrix} \begin{pmatrix} u_{t,x} \\ \theta_{x,x} \\ \theta_{y,x} \\ \theta_{z,x} \end{pmatrix}. \quad (5.5)$$

where:

- EA is the axial stiffness;
- GJ is the torsional stiffness;
- $EI_{y'y'}$, $EI_{z'z'}$, $EI_{y'z'}$ are the bending stiffnesses, respectively around the Y' axis, Z' axis and the flap-lag coupling term;
- AT , FT and LT are respectively the axial-torsion coupling, the flap-torsion coupling and the lag-torsion coupling, they are strongly dependent on the orientation of carbon / glass-fibres.

The balance equation of elasticity problem can be written using the virtual work principle:

$$W_{int} = W_{ext}, \quad (5.6)$$

where W_{ext} is the work of the external forces and W_{int} that of the internal forces:

$$W_{int} = \int_0^L (u_{t,x}N + \theta_{x,x}M_x + \theta_{y,x}M_y + \theta_{z,x}M_z)dx. \quad (5.7)$$

The spatial variation of the displacements is described using the so-called shape functions:

$$\{u\} = [N] \{s\}, \quad (5.8)$$

where $\{u\}$ is the displacement vector:

$$\{u\} = \{u_t \quad v_c \quad w_c \quad \theta_x \quad \theta_y \quad \theta_z\}^T, \quad (5.9)$$

$[N]$ is the shape function matrix:

$$[N] = \begin{pmatrix} L_1 & 0 & 0 & 0 & 0 & 0 & L_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & H_1 & 0 & 0 & 0 & H_3 & 0 & H_2 & 0 & 0 & 0 & H_4 \\ 0 & 0 & H_1 & 0 & -H_3 & 0 & 0 & 0 & H_2 & 0 & -H_4 & 0 \\ 0 & 0 & 0 & L_1 & 0 & 0 & 0 & 0 & 0 & L_2 & 0 & 0 \\ 0 & 0 & -H_{1,x} & 0 & H_{3,x} & 0 & 0 & 0 & -H_{2,x} & 0 & H_{4,x} & 0 \\ 0 & H_{1,x} & 0 & 0 & 0 & H_{3,x} & 0 & H_{2,x} & 0 & 0 & 0 & H_{4,x} \end{pmatrix}, \quad (5.10)$$

with L_i and H_i being respectively the Lagrange polynomials of the first order and the Hermite polynomials of the third order.

$\{s\}$ is the vector of nodal displacements: $\{s\} = \{\{s_1\} \{s_2\}\}$,

$$\{s_i\} = \{u_{t,i} \quad v_{c,i} \quad w_{c,i} \quad \theta_{x,i} \quad \theta_{y,i} \quad \theta_{z,i}\}. \quad (5.11)$$

Using the given definitions, eq. (5.8) can be expressed as a function of the nodal displacements:

$$W_{int} = \{s\}^T [K_{el}] \{s\}, \quad (5.12)$$

with $[K_{el}]$ being the stiffness matrix of the element. W_{ext} is made by the contribution of inertial forces and external loads and it can be rewritten as a function of nodal displacements:

$$W_{ext} = W_{inertial} + W_{loads} = -\{s\}^T [M_{el}] \{\ddot{s}\} + \{s\}^T \{F_{el}\}, \quad (5.13)$$

where $[M_{el}]$ is the mass matrix of the element and $\{F_{el}\}$ is the vector of nodal loads. Equation (5.6) can be simplified using expressions (5.12) and (5.13). That allows to obtain the final spatial discretization of the elasticity problem for a single element of the structure:

$$[M_{el}] \{\ddot{s}\} + [K_{el}] \{s\} = \{F_{el}\}. \quad (5.14)$$

The procedure to obtain the mass and stiffness matrices and their elements will not be described here and the reader can refer to [25]. The linear system (5.14) can be extended to the whole structure by properly assembling the two matrices and the load vector, hence:

$$[M] \{\ddot{d}\} + [K] \{d\} = \{F\}, \quad (5.15)$$

where $\{d\}$ is the vector of nodal displacements for the entire structure. Eq. (5.15) is discretized in the time domain using the Newmark method [61]:

$$\begin{cases} \ddot{d}(t + \Delta t) = \ddot{d}(t) + \delta t \left[(1 - \gamma)\ddot{d}(t) + \gamma\ddot{d}(t + \Delta t) \right], \\ d(t + \Delta t) = d(t) + \Delta t \dot{d}(t) + \frac{\Delta t^2}{2} \left[(1 - 2\beta)\ddot{d}(t) + 2\beta\ddot{d}(t + \Delta t) \right], \end{cases} \quad (5.16)$$

with $0 \leq \beta \leq \frac{1}{2}$ and $0 \leq \gamma \leq 1$. If $\gamma = 0.5$ the method is second-order accurate regardless of β . If $\beta > 0.25$ the method is unconditionally stable, therefore no limits of δt are imposed by stability issues. For the present work $\gamma = \beta = 0.5$ is chosen.

Substituting equation (5.15) into (5.16), the fully discretized system of equation for the elasticity problem is obtained:

$$\begin{pmatrix} \frac{\Delta t}{2} [K] & [M] \\ [M] + \frac{\Delta t^2}{2} [K] & 0 \end{pmatrix} \begin{Bmatrix} \{d_{k+1}\} \\ \{\dot{d}_{k+1}\} \end{Bmatrix} = \begin{Bmatrix} [M] \{\dot{d}_k\} - \frac{\Delta t}{2} [K] \{d_k\} + \Delta t \{F_k\} \\ [M] (\{d_k\} + \Delta t \{\dot{d}_k\}) + \frac{\Delta t^2}{2} \{F_k\} \end{Bmatrix}, \quad (5.17)$$

where $d_k = d(t_k)$ and $d_{k+1} = d(t_{k+1})$.

This model has been implemented and validated for the case of uniform loads. Indeed, a comparison of the simulated results has been done by colleagues at Inria with the analytic theory of beams [70].

1.2 Fluid-Structure Coupling in the Three-Dimensional Solver

Two methodologies exist to deal with numerical simulation of the fluid-structure interaction [40]: the monolithic and the partitioned approaches. The monolithic approach consists in a total coupling of the fluid with the elastic body, leading to a single system of equations, whose solution describes the entire domain of the problem. On the other hand, the partitioned approach separates the elasticity problem from the Navier-Stokes equation and solves two distinct systems

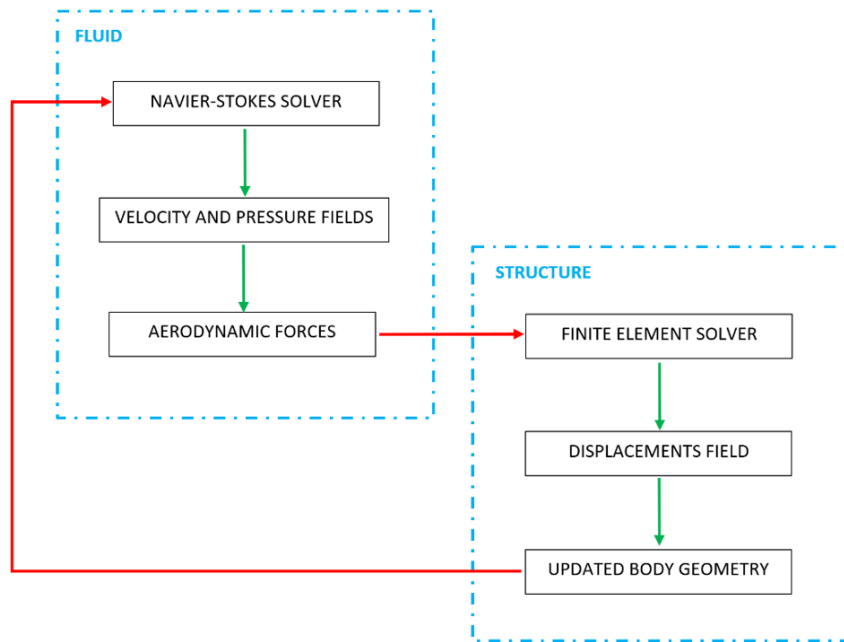


Figure 5.2: Partitioned scheme implemented in the 3D solver

of equations. That allows thus to split the two domains and to treat them with two different codes. While the monolithic approach can achieve better accuracy and is well suited for immersed boundary method, the partitioned one is more flexible and easier to implement. Indeed, it requires little modifications to both the structure and the fluid codes. Moreover, splitting the domains allows to use different meshes for the two, not only in term of computational cells, but also in term of topology, such as coupling a 3D finite volume method with a 1D finite element beam.

It was chosen to exploit the simplicity of the beam structure model and a partitioned scheme has been employed. That means, that, as represented in figure 5.2, during each time iteration, there is an exchange of information between the Navier-Stokes algorithm and the finite element model. The structural code needs indeed to know the aerodynamic forces to apply to the blade to evaluate the displacements. Those are then communicated back to the fluid solver to update the geometry and the body velocity. Moreover, the computation of the level-set function needs to be updated given the modification of the geometry. This implies a significant increase of the computational time, especially for complex geometries, as the blade, where a high number of Lagrangian markers is needed to properly describe it.

To summarize, a 6 degrees of freedom per node finite element beam model is employed with:

- linear axial deformation,
- bending, based on Euler-Bernoulli theory,
- linear torsion, based on Saint-Venant hypothesis,
- anisotropic stress-strain relationship for composite materials.

Indeed, a french company named Tensyl, specialist of composite structure and mechanical systems have been consulted for our wind blade model (APX48). They are able to compute high-fidelity structural simulations of wind blade deformations, which is not our aim in the present work. They explained that all the coupling phenomena are not significant for this blade. Therefore, the terms AT , FT and LT , that represent respectively the axial-torsion, the flap-torsion and the lag-torsion coupling are not considered in our beam model. The mass distribution, axial and bending stiffnesses used in the structural model are those of the APX48 wind blade.

2 Rotating Framework for Wind Blade Modelling

A first difficulty was to convert the geometry of the blade from a standardized CAD format into an array of properly organized Lagrangian markers that will be the input for both the Octree Navier-Stokes solver and the finite element code presented above. We had available the design of the wind blade only in a format describing the blade with non-uniform rational b-splines. A second obstacle has also been added with the presence of the hub of the rotor. The level-set function is so computed for both obstacles. The modifications operated in the fluid and structure models will now be described before presenting some preliminary results obtained in this configuration.

2.1 Boundary Conditions and Rotating Models

The rotating frame fluid model will now be explained. Considering an inertial frame (X, Y, Z) and a rotating one (X', Y', Z') , the relationship between the velocity of a point expressed in the two references is:

$$\mathbf{V}_i = \mathbf{V}'_r + \boldsymbol{\omega} \wedge \mathbf{r}, \quad (5.18)$$

with \mathbf{r} the position in a rotating reference, $\boldsymbol{\omega}$ is the rotational speed, subscript i means the inertial frame of reference and r means the rotating one. For the present work, the y-axis of the domain is taken as the rotation axis of the wind blade:

$$\boldsymbol{\omega} = \begin{pmatrix} 0 \\ \Omega \\ 0 \end{pmatrix}. \quad (5.19)$$

It is therefore straightforward to obtain:

$$\begin{cases} \nabla \cdot \mathbf{V}_i = \nabla \cdot \mathbf{V}'_r \\ \nabla^2 \cdot \mathbf{V}_i = \nabla^2 \cdot \mathbf{V}'_r \end{cases} \quad (5.20)$$

In the case of a fluid system, the acceleration is equal to the Lagrangian derivative:

$$\mathbf{a} = \frac{D\mathbf{V}}{Dt}, \quad (5.21)$$

and the relationship between the acceleration in the two references can be written:

$$\mathbf{a}_i = \mathbf{a}_r + 2\boldsymbol{\omega} \wedge \mathbf{V}'_r + \boldsymbol{\omega} \wedge (\boldsymbol{\omega} \wedge \mathbf{r}') + \frac{d\boldsymbol{\omega}}{dt} \wedge \mathbf{r}', \quad (5.22)$$

where \mathbf{r}' is the position in the rotating frame, $\boldsymbol{\omega} \wedge (\boldsymbol{\omega} \wedge \mathbf{r}')$ represents the centrifugal acceleration and the term $2\boldsymbol{\omega} \wedge \mathbf{V}'_r$ is the Coriolis acceleration. Since our blade is submitted to a constant revolution rate, and using the definition of eq. 5.21, equation 5.22 becomes:

$$\frac{D\mathbf{V}_i}{Dt} = \frac{D\mathbf{V}'_r}{Dt} + \boldsymbol{\omega} \wedge (\boldsymbol{\omega} \wedge \mathbf{r}') + 2\boldsymbol{\omega} \wedge \mathbf{V}'_r. \quad (5.23)$$

2. Rotating Framework for Wind Blade Modelling

The Navier-Stokes momentum equation, in the Lagrangian form is:

$$\frac{D\mathbf{V}_i}{Dt} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{V}_i. \quad (5.24)$$

Substituting expressions 5.20 and 5.24 into 5.23 and considering the continuity equation, the incompressible Navier-Stokes equations for a rotating reference frame are obtained:

$$\begin{cases} \nabla \cdot \mathbf{V}'_r = 0 \\ \frac{D\mathbf{V}'_r}{Dt} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{V}'_r - \omega \wedge (\omega \wedge \mathbf{r}') - 2\omega \wedge \mathbf{V}'_r \end{cases} \quad (5.25)$$

The two non inertial terms act as volume forces and they can be accounted in the prediction step of the Chorin algorithm. Using the expression of ω , the centrifugal and Coriolis accelerations are:

$$\omega \wedge (\omega \wedge \mathbf{r}') = \omega \wedge \begin{pmatrix} \Omega z' \\ 0 \\ -\Omega x' \end{pmatrix} = -\Omega^2 \begin{pmatrix} x' \\ 0 \\ z' \end{pmatrix} \quad \text{and} \quad (5.26)$$

$$2\omega \wedge \mathbf{V}'_r = 2\Omega \begin{pmatrix} w' \\ 0 \\ -u' \end{pmatrix}. \quad (5.27)$$

The boundary conditions used for the velocity field have also been modified to simulate the rotating blade compared to previous simulations done for code validation. Indeed, the following conditions have been used for the velocity:

- At the inlet (y-direction), Dirichlet boundary conditions with $u_y = U_\infty$, $u_x = -\Omega z$ and $u_z = \Omega x$ are imposed.
- At the outlet, homogeneous Neumann conditions are imposed.
- In the lateral x and z directions, Dirichlet boundary conditions are used, exactly the same as for the inlet.

Considering now the structural model of the blade, the Coriolis and centrifugal forces need to be included in the finite element model described above. The method used is widely described in the Master thesis of Pezzano [70] who worked at MEMPHIS team. It was chosen to keep the structural model as simple as possible. Therefore, the rotation is assumed to be only with respect to the z-axis, which means that tilting and rotation movement of the hub are neglected. Moreover, the dynamic effects caused by other rotational deformation speeds are also neglected and gyroscopic torques are not considered. The work of the Coriolis force and of the centrifugal force are written thanks to their definitions. The structural equations are solved in order to obtain the matrix of the shape functions of the complete beam model. From equation 5.15, the space discretized equations of elasticity are then obtained by taking into account the newly added contributions:

$$[M] \{\ddot{d}\} + [C] \{\dot{d}\} + [K] \{d\} = \{F_{ext}\} + \{F_{cf}\} = \{F\}, \quad (5.28)$$

with $[C]$, the Coriolis matrix and $\{F_{cf}\}$ is the contribution of the centrifugal force. Applying the Newmark method with $\gamma = \beta = 0.5$ allows to get the linear system for the solution of the

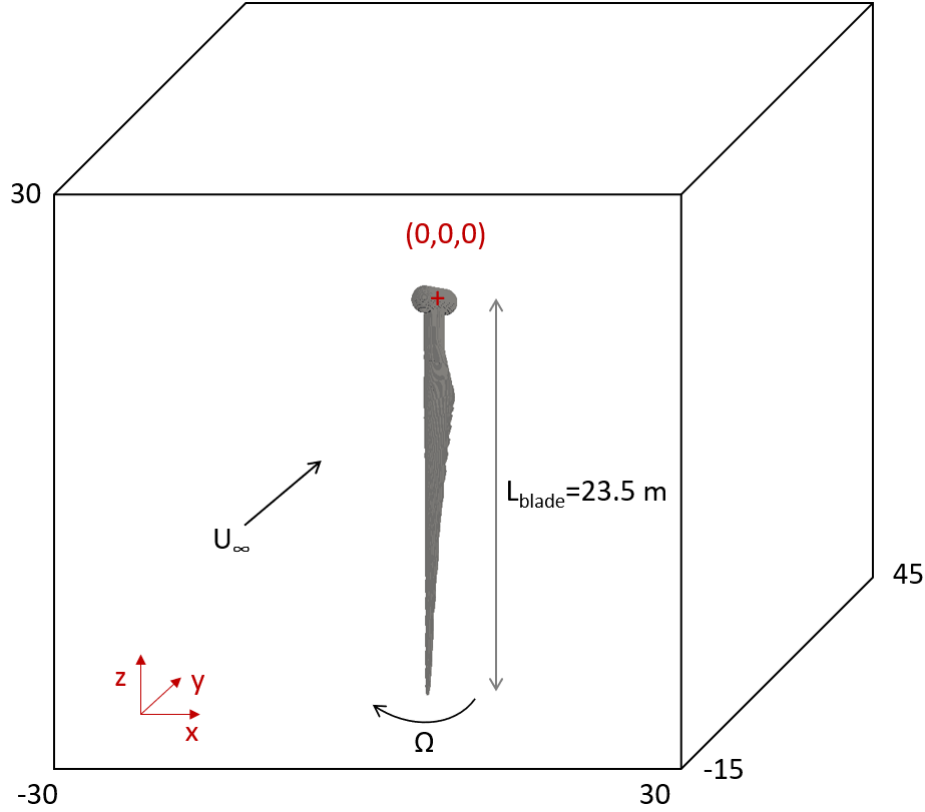


Figure 5.3: Sketch of the computational domain for the flow past a rotating blade.

dynamic structural problem:

$$\begin{pmatrix} \frac{\Delta t}{2} [K] & [M] + \frac{\Delta t}{2} [C] \\ [M] + \frac{\Delta t^2}{2} [K] & \frac{\Delta t^2}{2} [C] \end{pmatrix} \begin{Bmatrix} \{d_{k+1}\} \\ \{\dot{d}_{k+1}\} \end{Bmatrix} = \begin{Bmatrix} ([M] - \frac{\Delta t}{2} [C]) \{\dot{d}_k\} - \frac{\Delta t}{2} [K] \{d_k\} + \Delta t \{F_k\} \\ [M] (\{d_k\} + \Delta t \{\dot{d}_k\}) + \frac{\Delta t^2}{2} \{F_k\} \end{Bmatrix}. \quad (5.29)$$

2.2 Preliminary Results on a Rotating Wind Blade

Once the implementation of the rotating framework has been performed, all the features developed during the present work were ready to simulate a rotating blade with operating conditions corresponding to those of the real wind turbine. A thought has been done to define the computational domain and the mesh to use in order to perform the simulations. We are indeed dealing with a blade whose length is 23.5 meters. A compromise has been done on the computational domain size in order not to reach a high number of degrees of freedom while having a correct resolution on the blade. Moreover, it is needed to be far enough from the boundary conditions. Otherwise, abnormal behaviours of the flow around the blade were observed. The boundary conditions have indeed a key impact on the numerical tool developed. The computational domain chosen is $[-30, 30] \times [-15, 45] \times [-30, 30]$ and it is centred at the middle of the blade hub and the blade extends in the negative z -direction. The incoming wind arrives from the negative y -direction and y -axis corresponds to the rotation axis of the wind blade. This can be seen in figure 5.3.

The grid chosen is also a compromise to reduce the computational time. The minimum size

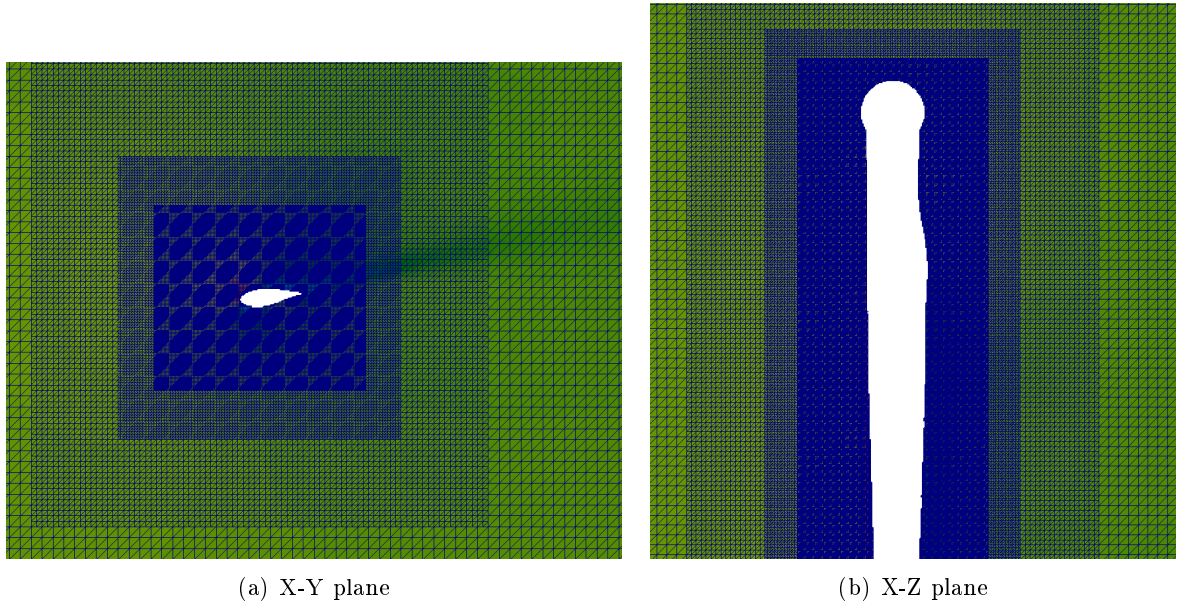


Figure 5.4: Display of the octree mesh used for the simulation of the flow past a rotating blade.

of the grid is $\Delta x = 0.0293$ and five different levels of refinements are employed. The objective being to simulate the fluid-structure interaction occurring during the rotation of the blade, it was necessary to reduce the number of degrees of freedom. As explained briefly previously, simulating with the structural model of the blade implies to compute the level-set function and its gradients at each time step which is very significantly time consuming. Indeed, the number of Lagrangian markers necessary to properly describe the blade geometry is around 200 000. Moreover, the gradient of the total level-set (hub+blade) should be evaluated for the computation of the wall law, whereas for the aerodynamic forces computation, the use of the gradient of the blade level-set is necessary. Therefore, a compromise has been done on the minimum grid size and on the number of cells in order to use a reasonable number of computational hours. The total number of cells of the chosen grid is around 28 million and the octree mesh is displayed in figure 5.4.

All the features developed during the present work have been used for this preliminary simulation with the wall law, the turbulence model, the rotating framework. This section indicates preliminary results since, now, only a uniform wind speed is imposed at inflow. So, this case does not constitute a real field condition of the blade. However, a real operating condition has been simulated in the sense that an incoming wind of $8 \text{ m}\cdot\text{s}^{-1}$ with a rotational speed of 21.6 rpm has been used, which corresponds to one working condition of the wind turbine. The viscosity has been imposed as the one of the air with a usual temperature. As a way to validate the results, the power extracted by the simulated wind turbine is recorded.

The figure 5.5 shows the streamwise velocity of the flow after 3.5 revolutions of the wind blade. The contour lines reveal the tip vortex phenomenon that is developing in the wake of the blade. When looking at the streamwise velocity in x, y and z-plane, no small structures can be seen and especially, in the z-plane, the wake behind the airfoil is not very extended. Indeed, as the Eulerian scheme to treat the advection term is only first order accurate, significant numerical dissipation occurs during the simulation. Moreover, the used mesh does not allow a very accurate simulation of the flow. The streamwise velocity in the y-plane shows the rotation movement

of the blade. In the x-plane, the flow pattern of the tip vortices can be seen in the wake.

Simulation of 22 s of flow (that corresponds to around 8 revolutions of the blade) has been performed on Occigen with 600 processors. The corresponding computational time was 338 hours. The power extracted by the wind turbine is computed during this simulation as a way to compare with the real operating data. The computation is done thanks to equation (1.4) that defines the power as a function of the torque of the three blades. Data from the wind turbine manufacturer inform us that the expected power extracted by this wind turbine with the previously detailed operating condition is 251 kW. After 8 revolutions of the blade, the simulated value begin to be stabilized and we obtained a power of around 114 kW as can be seen in figure 5.6. It can be concluded that a good order of magnitude is obtained. However, the simulation performed did not allow to get a good accuracy for this result. The possible improvements that could be done are the use of a second-order accuracy for the penalization and to refine in the area of the wind blade. Both possibilities would certainly imply a better accuracy in the computation of the aerodynamic forces. We are indeed simulating a very complex case. Also, even with the high numerical dissipation, the flow pattern with the tip-vortex phenomenon seems to represent correctly the flow around a wind blade.

3 Adaptive Mesh Refinement Process

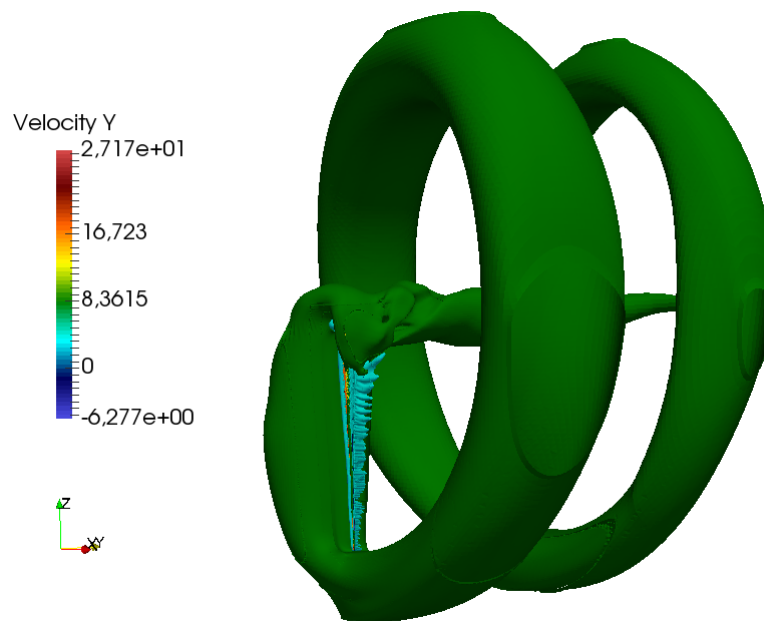
An adaptive mesh refinement (AMR) process has been implemented in the code in order to optimize the number of cells while controlling the accuracy. The aim is indeed to refine in the most interesting areas. For our present work, the areas of highest interest in the flows studied are basically the wakes of the different objects tested. We could thus be able to refine the grid on the zone that follows the wake and coarsen elsewhere. An adequate criterion should be defined in order to comply with this objective. Several criteria have been tested for the case of the flow around a sphere. This flow has been chosen since it deals with three-dimensional effects and this case does not require much computational time to perform tests. The different criteria tested were:

- based on the q-criterion,
- based on the vorticity,
- based on the L_2 norm of velocity gradients normalized by the L_2 norm of velocity. For

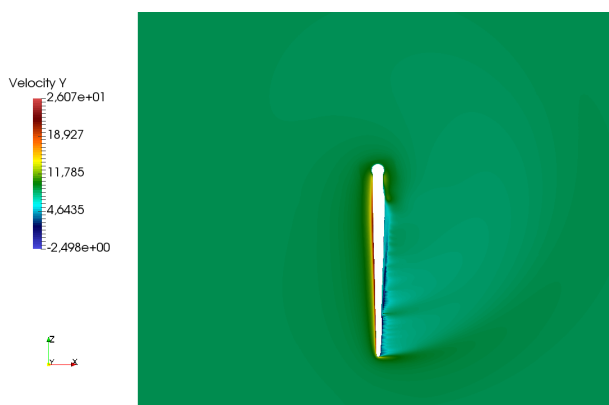
example, in 2D, the criterion is:
$$\frac{\sqrt{\left(\frac{\partial U_x}{\partial x}\right)^2 + \left(\frac{\partial U_x}{\partial y}\right)^2 + \left(\frac{\partial U_y}{\partial x}\right)^2 + \left(\frac{\partial U_y}{\partial y}\right)^2}}{\sqrt{U_x^2 + U_y^2}}.$$

The tests performed with a criterion for the AMR based on the q-criterion showed inadequate behaviours. Indeed, the q-criterion allows to highlight the vortices. Therefore, a grid adaptation based on it does not permit to follow the whole wake but only the vortices. The two other criteria show similar results and allow to adapt the grid along the wake as requested. The advantage of the criterion based on velocity gradients normalized by the norm of velocity is the fact that the threshold to indicate for coarsening and for refinement is less dependant of the studied flow when using the normalization by the velocity. Indeed, depending on the case to simulate (different obstacles, different Reynolds numbers,...), the parameters for adaptation will vary. The chosen criterion shows to be very less dependant to these flow configurations and the values to indicate for adaptation can be define with very few tests.

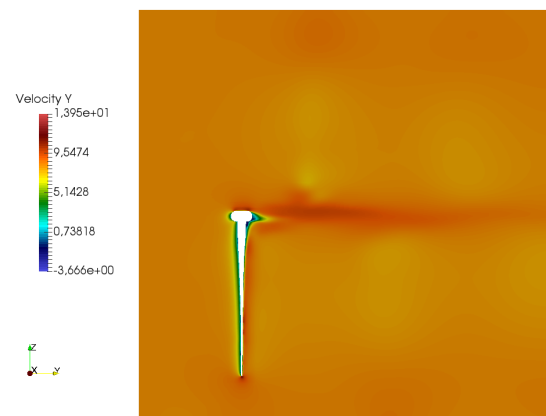
3. Adaptive Mesh Refinement Process



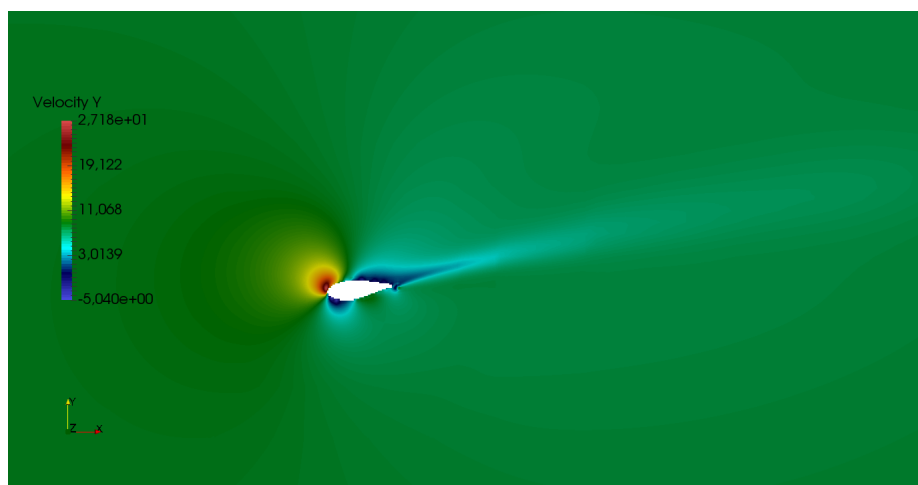
(a) Contours of the streamwise velocity



(b) Streamwise velocity in y-plane



(c) Streamwise velocity in x-plane



(d) Streamwise velocity in z-plane at 12 m from the blade root

Figure 5.5: Display of the flow around a rotating blade with $U_\infty = 8 \text{ m}\cdot\text{s}^{-1}$ and $\Omega = 21.6 \text{ rpm}$ after 3.5 revolutions of the blade.

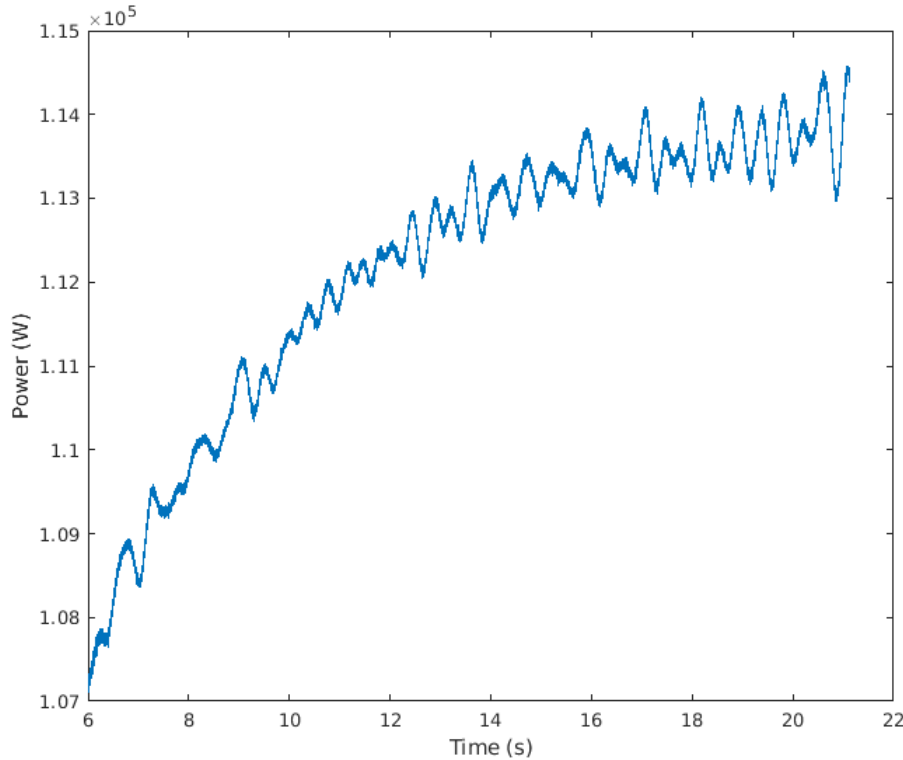


Figure 5.6: Time history of the power extracted by the simulated wind turbine.

The adaptive mesh refinement process has been implemented in the 2D and 3D solvers. This stage is performed after the prediction step so that the zero divergence condition is guaranteed at the end of the grid adaptation. Indeed, once the grid has been adapted, the predicted velocity field computed at the prediction step should be updated for the new grid configuration. This is done in the present work in a simple way. Thanks to the bitpit library, different functions exist so that each cell of the new grid configuration has information toward the old configuration. Three different possibilities exist for each cell:

- if the cell has the same size as in the former configuration, the value contained in the former cell is applied to the "new" one;
- if the cell has been refined since the former state, the value contained in the former bigger cell is applied;
- if the cell has been coarsened, a basic mean is performed with all the values contained in the smaller cells constituting this new bigger cell. For example, in 2D, the coarsened cell will have the value of $\frac{1}{4}$ of the sum of the 4 former smaller cells.

The level-set function and its gradient should also be computed again after each grid adaptation, which consist in the most time-consuming step of the AMR process. The different results obtained with several test-cases will now be described.

3.1 Two-Dimensional Cases

The computation of the velocity gradients is thus needed to perform the grid adaptation. The same method as used for the divergence and gradient operators in the Navier-Stokes solver as described in chapter 2 for two-dimensional case is used. The AMR process has been tested in

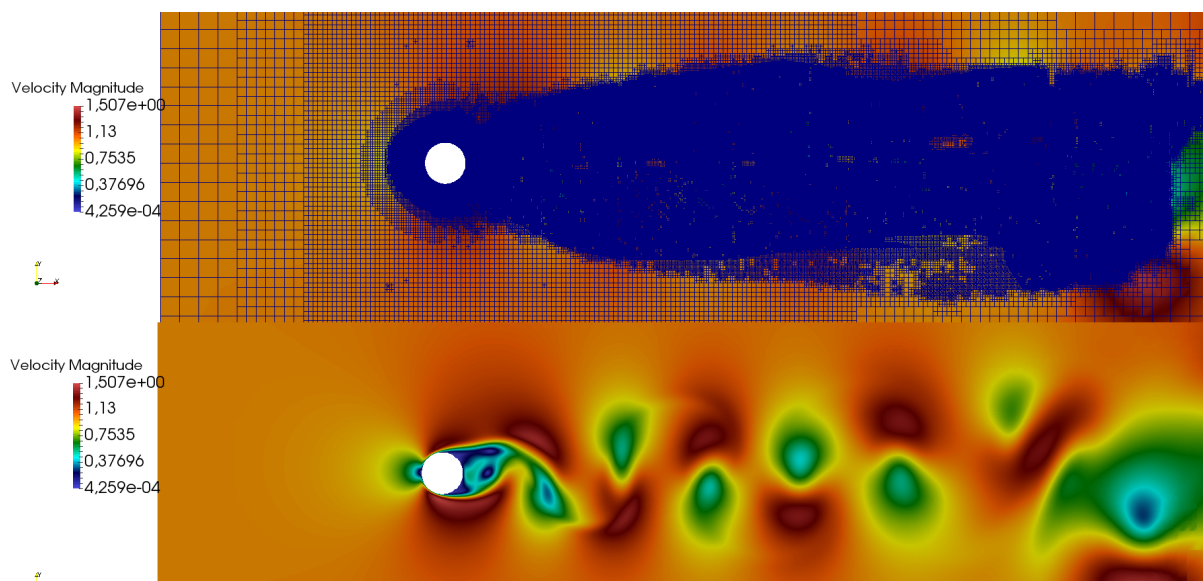


Figure 5.7: Velocity magnitude for the flow past a circular cylinder in 2D at $Re = 200$ showing the AMR process with superimposition of the mesh at the top

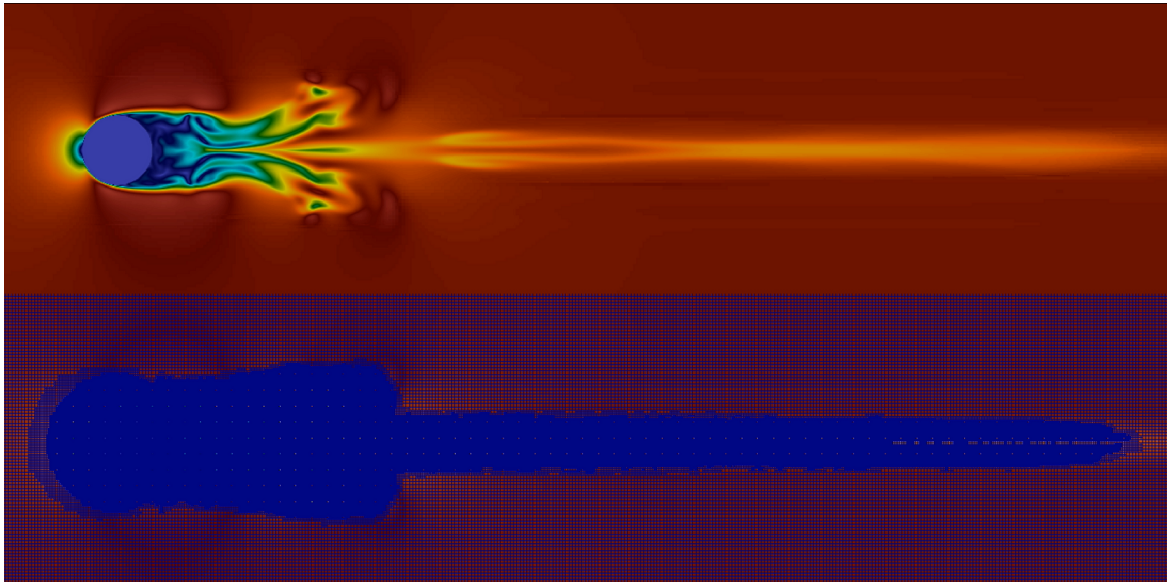
the two-dimensional solver with two different flows as a proof of concept. Indeed, for 2D cases, the use of grid adaptation is less pertinent as in 3D since the gain in term of cell numbers has less impact, moreover in light of the computational time needed for the AMR process.

Figure 5.7 shows the grid adaptation for the case of the flow around a circular cylinder at $Re = 200$. The same AMR process has been performed for the flow around a NACA0012 airfoil at $Re = 1000$. As an indication, the same values for the coarsening and refinement parameters have been used for both flow configurations. The grid is actually refined in the wakes but the coarsening can be improved. It has been chosen to perform the grid adaptation only for the 3 finer level of trees in order not to have too close level jumps. No further work has been pursued to find the right parameters for the two-dimensional case since we wanted to spend more efforts on the three-dimensional case. We validated the method on these two-dimensional cases before working in the 3D solver.

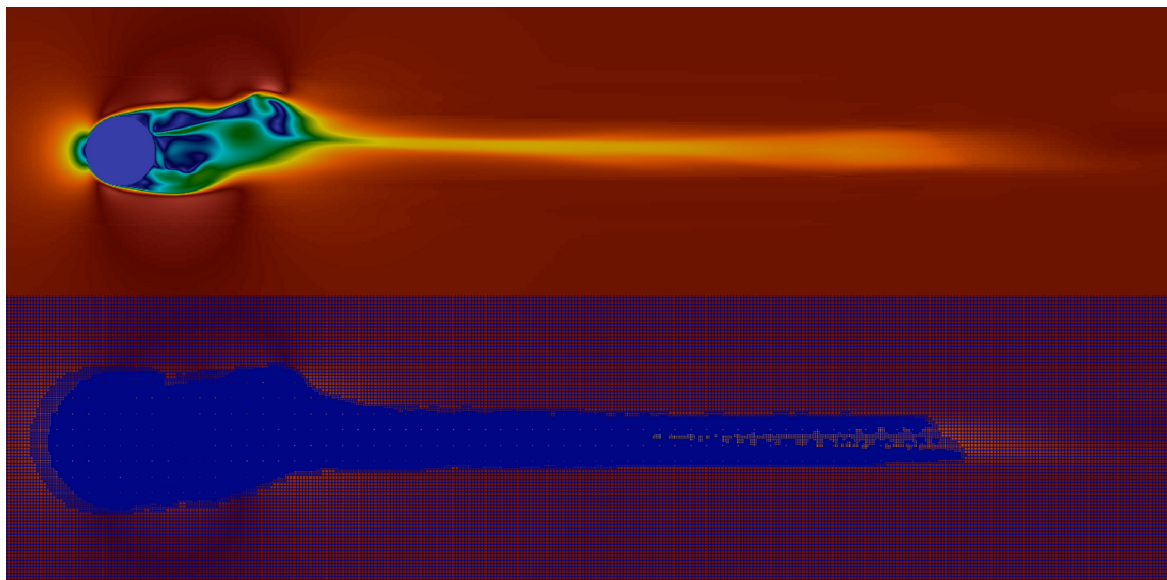
3.2 Three-Dimensional Cases

For the computation of the gradients in 3D case, the same method as used for the gradient and divergence operators in the Navier-Stokes equations, explained in the chapter 2, is employed. A particular attention has been brought to find adequate values for the parameters to refine and coarsen the grid. It has also been chosen to perform the AMR for 3D cases only with the two finer levels of octree to avoid several level jumps in the same area that could imply a significant loss of accuracy. The AMR has been tested firstly for the case of the flow around a sphere at $Re = 5000$. This quite high Reynolds number has been chosen in order to have a turbulent flow, more representative to the simulation of the rotating blade. The grid adaptation obtained for this case can be seen in figures 5.8 to 5.10 and we can observe that a fine optimization in the grid distribution is reached. Moreover, no loss of accuracy seems to occur at the level jumps.

The AMR process has then been tested for the case of the rotating blade which is our case of highest interest. A global quite coarse mesh has been used for the tests. It was noticed that the values to use for the grid adaptation needed to be changed compared to the case of the

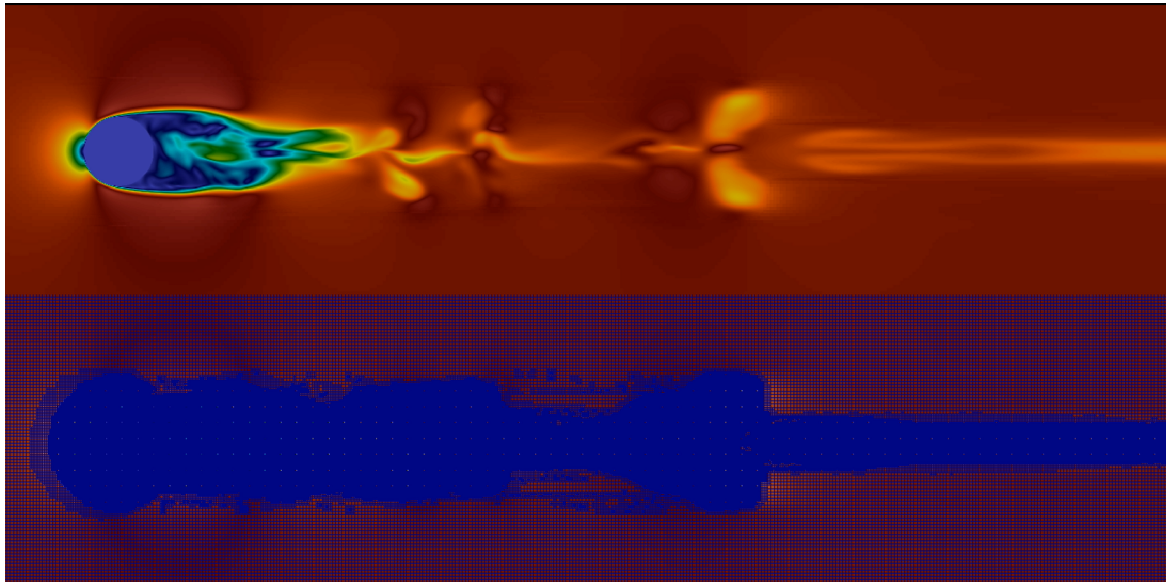


(a) Velocity-magnitude in y-plane

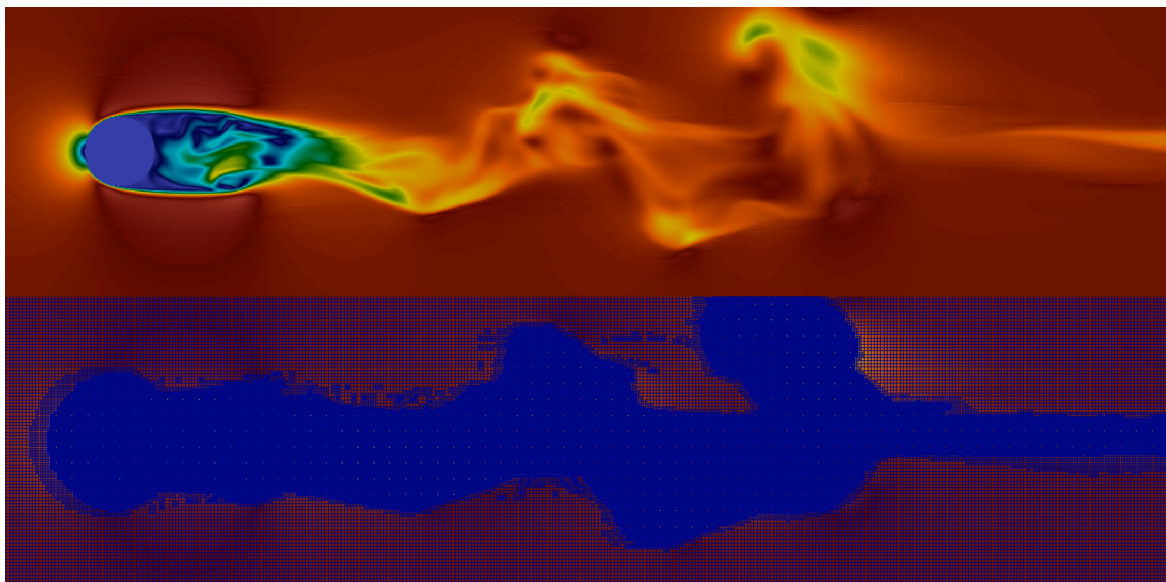


(b) Velocity-magnitude in z-plane

Figure 5.8: Beginning of the simulation with AMR process for the case of the flow around a sphere at $Re = 5000$

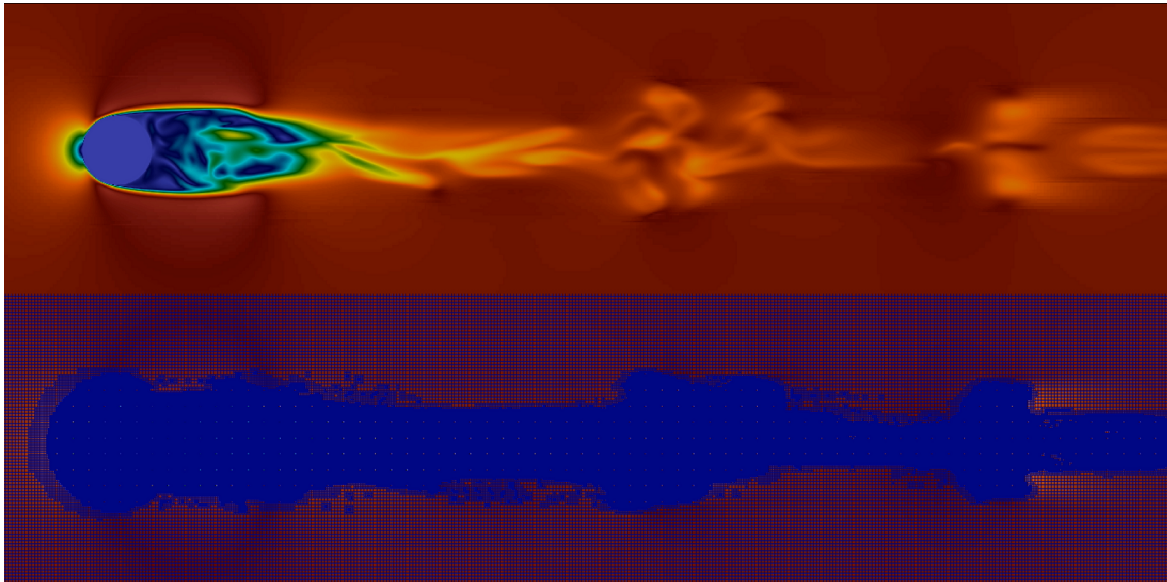


(a) Velocity-magnitude in y-plane

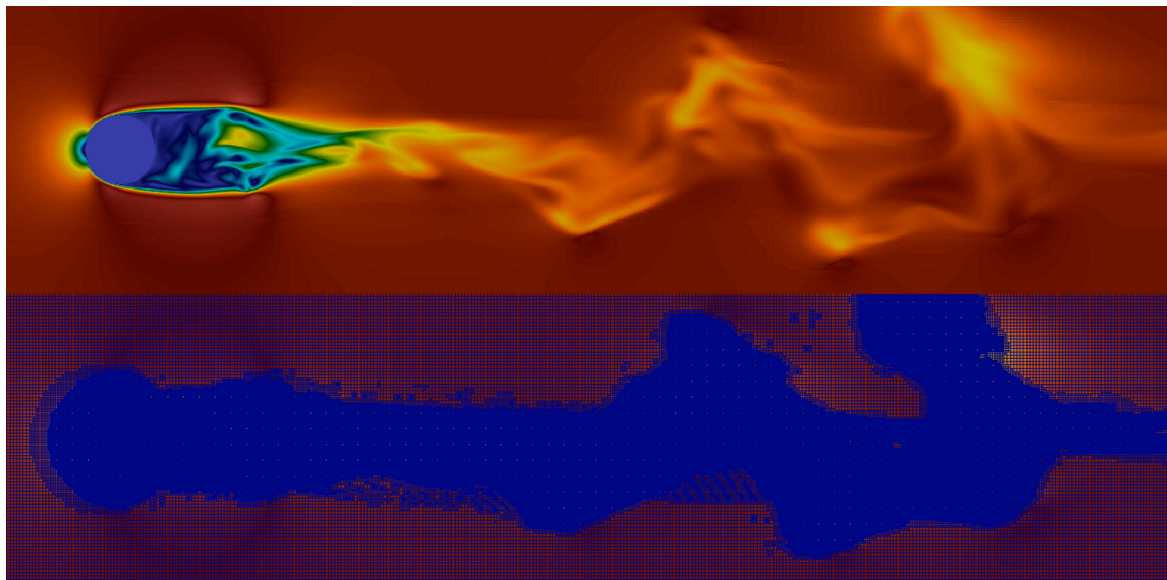


(b) Velocity-magnitude in z-plane

Figure 5.9: Middle of the simulation with AMR process for the case of the flow around a sphere at $Re = 5000$



(a) Velocity-magnitude in y-plane



(b) Velocity-magnitude in z-plane

Figure 5.10: End of the simulation with AMR process for the case of the flow around a sphere at $Re = 5000$

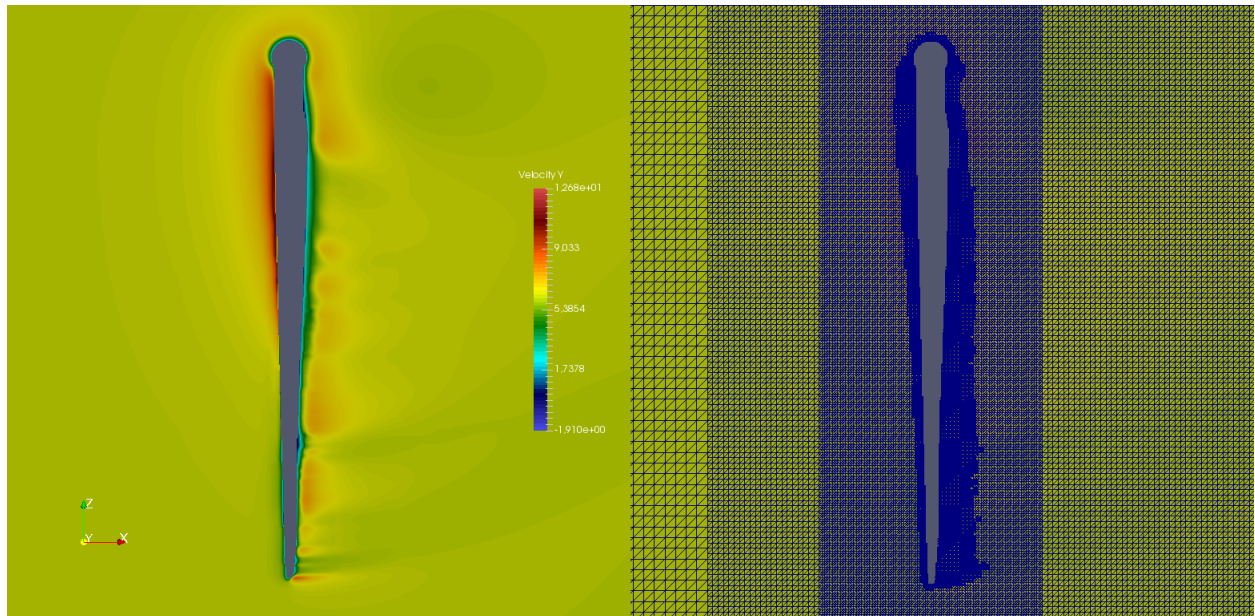
sphere. Indeed, the both flows are very different, especially since, for the blade, the rotating framework is employed. Figure 5.11 shows the AMR process for the flow around the rotating blade with a low Reynolds Number (around 1000). It can be observe that a good grid adaptation is reached. However, when increasing the Reynolds number and simulating with a real operating condition of the wind turbine, the mesh adaptation doesn't fit properly with the wake of the blade. It shows that the criteria need to be slightly adapted. This work will be extended in order to perform future simulations of the rotating blade with AMR.

4 Rotating blade submitted to a gust

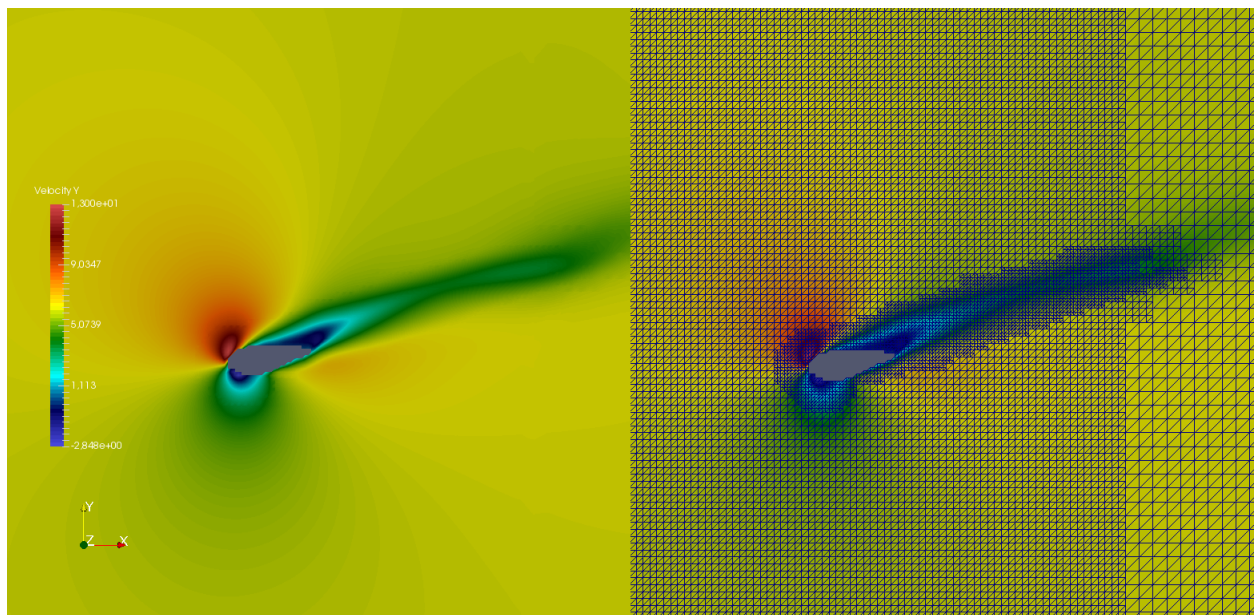
As a first approach, a fixed mesh has been used and a gust-like perturbation has been submitted to the rotating blade. The experimental gust events highlighted in the previous chapter show that the mean duration of an event is around 30 seconds. But we were not able with the remaining time (human and computational) to simulate such a long period. Therefore, we performed a simulation with a fast increase of the velocity to reproduce a gust-like behaviour. The preliminary simulation described above was used as an initial state, with a wind speed of 8 m.s^{-1} as boundary condition. The gust is then imposed as a linear increase of the wind speed. This linear increase lasts 0.5 s and the maximal wind speed of 13 m.s^{-1} is then reached. Gust events highlighted during the experimental work have been used to have ideas of reasonable values of the wind speed increase during a real gust. The boundary conditions in the lateral x and y directions have been adapted compared to those used for the preliminary simulation. As a reminder, Dirichlet boundary conditions were firstly imposed with $u_y = U_\infty$, $u_x = -\Omega z$ and $u_z = \Omega x$. But, as we have now an increase of U_∞ along time, it seemed more reasonable to impose Neumann boundary condition in the y direction for the 4 lateral boundaries of the computational domain.

The structural model of the blade has been included in the simulation as a first approach, as well as the wall law and the turbulence model. The objective would be to determine the effects of the gust on the structural behaviour of the blade. An extension of the present work will indeed be performed in order to obtain a Reduced Order Model (ROM) of the structural model of the blade. Both the experimental data of pressure collected on the wind blade and the pressure data from this simulation will be used to build the Proper Orthogonal Decomposition (POD) modes of the structural model of the wind turbine. ROMs could allow to reduce significantly the computational time needed for the fluid-structure interactions. The pressure data is so recorded at each Lagrangian marker that describes the blade geometry and at each time step. But the computational cost of the structural model with that mesh and flow configuration was way too important to be performed, even with a gust-like event faster than real gusts. As a comparison, with the same computational time, and in the case of a gust event, the simulation without structural model computes around 1 s whereas with the structural model only around 0.075 s was simulated. This was therefore not possible to use the structural model with that configuration. As previously explained, the cost of the fluid-structure interaction is not only due to the structural model but also to the computation of the level-set function and its gradients.

The structural model has so been removed to perform the simulation. When the streamwise velocity were looked up during the gust-like event, an incorrect behaviour of the flow is observed. This can be seen in figure 5.12. At the beginning of the wind speed increase, the behaviour stays similar as previously showed, while having an increase of velocity in front of the blade. But after a few times, the flow does not behave correctly anymore. This could be due either to the change in the boundary conditions, compared to the case without gust, that might be unsuitable for



(a) Y-Velocity in y-plane



(b) Y-Velocity in z-plane at 10 m from blade root

Figure 5.11: AMR process for the rotating blade with an operating condition at $U_\infty = 6 \text{ m.s}^{-1}$, $\Omega = 16.2 \text{ rpm}$ and $Re = 1000$

4. Rotating blade submitted to a gust

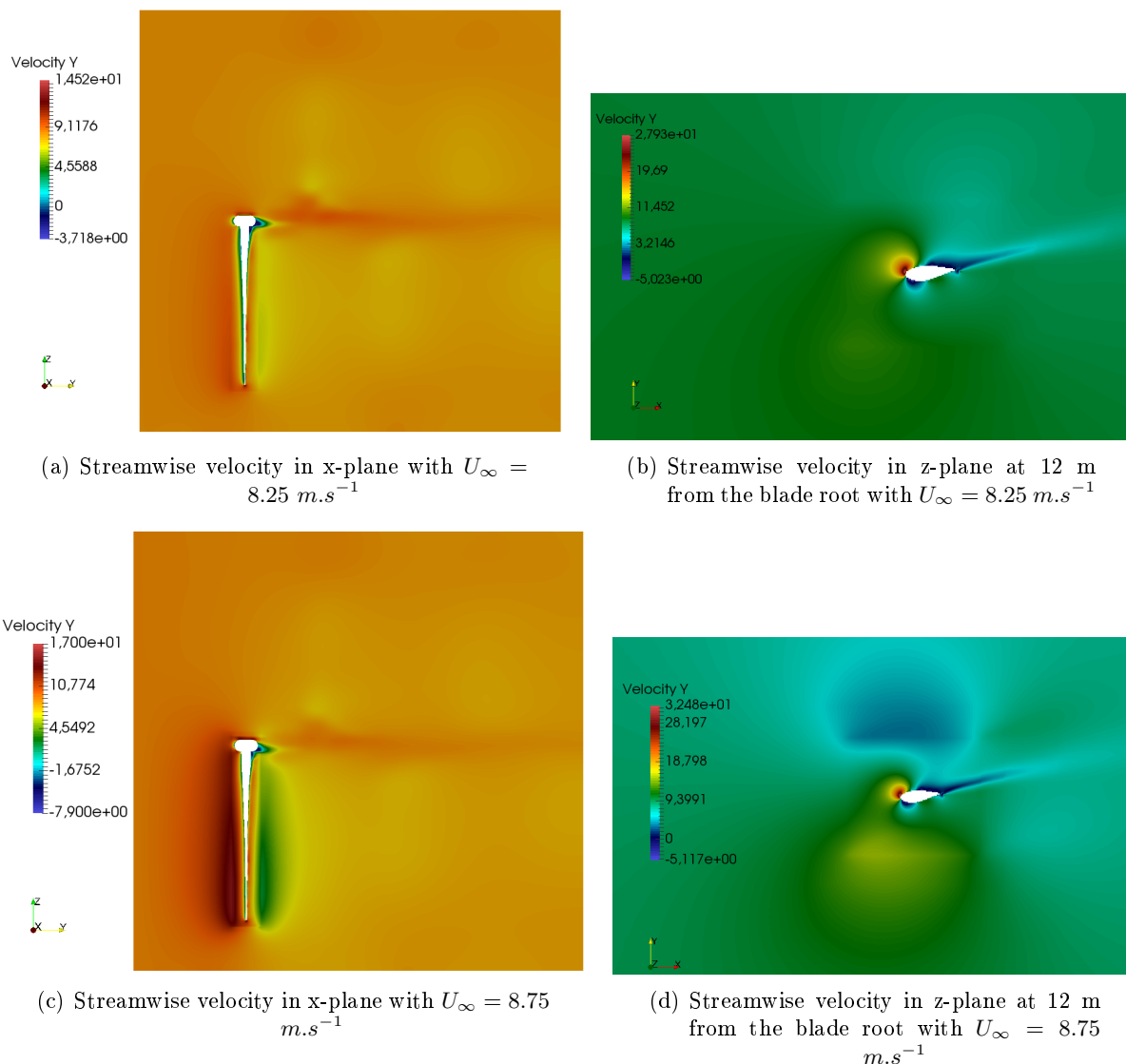


Figure 5.12: Visualization of the flow around a rotating blade with a gust-like event.

this complex flow. Indeed, the solver developed is very sensitive to the choice of the boundary conditions. An other possibility could be the fact that the increase of incoming wind is too fast for the solver.

Finally, even if we are not ready to simulate a gust event, the preliminary simulations show promising results with an octree solver which is only first-order accurate. The flow with the wall law and the turbulence model seems to reproduce the behaviour of the flow around a wind blade. The numerical tool has been validated in the previous chapter with several non-trivial test cases which is a promising result. The Adaptive Mesh refinement process needs slight adjustments of the used criteria before being employed in the simulations, but it could allow to reduce the numerical diffusion in the wake of the blade. We also have at our disposal a high number of experimental data with different kind of values as the wind speeds at different heights, that allows to have real boundary conditions, pressure data and deformations on the wind blade. This is of high interest to perform comparisons with several different simulations.

Conclusion

In this thesis we have proposed a numerical method to solve the incompressible Navier-Stokes equations on octree meshes. Two and three dimensional solvers have so been developed in that context with finite volume method. The application field of this work is the study of the flow around wind turbines. Therefore, as dealing with high Reynolds numbers, Large Eddy Simulations with a subgrid turbulence model has been implemented. We were aware that octree grids don't allow to properly define the anisotropic boundary layer around the wind blade. Indeed, an octree mesh can only refine by decreasing its size by two, which is not enough for the boundary layer. Consequently, the choice has been done to implement in the solver a wall law to model the smallest scales close to the blade. The structural behaviour of a blade plays a significant part in the flow around it, so a fluid-structure interaction has been developed with a beam model of the blade. The structural model, as well as the geometry of a real wind blade have been used.

The existing numerical models used for preliminary design of wind turbines are first described. These tools, that are often partly based on the Blade Element Momentum theory have the advantage of being fast to compute. But, the geometry of the blade should be known with the polar curves of the airfoils, the chords, the twist angles... It is therefore not possible to perform simulations of a new blade or of a blade with aerodynamic appendices, without a preliminary work. On the other hand, Computational Fluid Dynamics (CFD) allow the computation of new geometries. The grid to use is a key parameter for CFD since it needs to be fine enough to accurately model the flow, while having a significant importance on the computational time. Numerical schemes with a discretization based on octree grids have been investigated and these meshes show an interesting compromise between computational time and accuracy for our application to flow around wind turbines. It is indeed well adapted to Adaptive Mesh Refinement process, that allows to refine in the areas of high interest in the computational domain while coarsen elsewhere.

In the present work, the Navier-Stokes equations are solved with a penalization method which is well adapted to Cartesian or octree grids. In this framework of penalized Navier-Stokes equations, a two-dimensional solver based on a octree discretization is developed. Finite volume method is mainly used in the thesis. The numerical scheme developed to treat the Laplace operator is based on Discrete Duality Finite Volume methods with a Diamond dual mesh introduced to compute the discrete gradient at the face centre. For the advection term, a semi-Lagrangian scheme has been developed that have several advantages as it avoids the computation of the convective term and is unconditionally stable. All the operators developed are validated with the use of analytical functions and their consistency is proved. These methods are then extended to the three-dimensional case and the operators are again validated with the use of analytical functions that prove their consistency. As previously explained, with an application of the solver being the modelling of the flow around wind blades, wall functions for high Reynolds number flow have been implemented together with Large Eddy Simulations. A Vreman subgrid model

has been used and implemented in the context of octree grids.

As the models proposed in the present work are original and have the objective of being a compromise between accuracy and computational time, a wide validation was essential. The consistency and accuracy of the whole solver in 2D has been computed thanks to the analytical solution of the Taylor Green vortex. A second order accuracy is reached in space and a first order in time. Data from literature has then been used to perform comparisons with the solver developed. The flow around a circular cylinder has been studied at several Reynolds numbers. The drag coefficient and the Strouhal number have been compared to the literature and our results show to be in line with the literature. The flow around a NACA0012 airfoil has also been studied to perform simulations with a more complex geometry. The lift coefficient computed with our solver has thus been validated by comparing with literature. While performing simulations with the three-dimensional solver, a stability issue occurred with some grids. It has been realized that it was due to the semi-Lagrangian scheme. As we were not able to fix that issue, a first-order Eulerian scheme has been implemented instead to solve the advection terms of Navier-Stokes equations. The 3D solver has then been widely tested. The flow around a sphere has been simulated at low Reynolds number and the drag coefficient has been validated by comparing with data from literature. The flow around a cylinder at Reynolds 3900 has then been widely studied with a comparison of wake profiles simulated with the developed solver and with experimental data. After this flow configuration at a medium Reynolds number, a simulation with the flow around the cylinder at higher Reynolds, 140 000, has been performed. This flow simulation was performed with LES and a viscous wall function. This case has been compared with experimental data from literature and also with a well-known Cartesian code. The grid used for both simulations is compared and arguments are given concerning the computational time. The results obtained are in-line with literature, which allowed to validate the numerical tool developed.

In the framework of a H2020 European Project (AeroGust), we had the opportunity to perform an experimental work in order to collect in-service wind turbine data. The aim of the project is to study the effects of gust on a wind turbine blade. Specifications for this study were to get measurements on a wind blade where we know geometrical and structural data. Wind measurements are also needed in order to represent the wind as it arrives at the wind turbine. Valorem uses usually meteorological masts to perform wind assessment studies for wind farm development as prescribes in the IEC 61400-12-1 standard [1]. As the wind during a gust is characterized by high variations in wind velocities in a few seconds as explained in the IEC standard, the acquisition frequency of wind data should be small enough. In order to enable estimation of the vertical wind profile (strong gradients can exist in the vicinity of the ground), anemometers at 5 different heights have been installed. Concerning the blade data, Valorem has not the expertise to install sensors on a strong experimental case which is an operating wind blade. So, a complete technological solution was looked up with regards to sensors supply, installation, maintenance and communication system. A global requirement was to minimize the impact of measurement tools on the blade as the wind turbine should be working well during and after the experimental phase. In order to have experimental data showing the aero-elastic behaviour of a wind blade, the measurement of blade deformations will allow to observe its structural behaviour. To observe the aerodynamic load on the wind blade, the measurement of pressure of air will be of significant interest. The technological solution of choice uses optical fibre with Fiber Bragg Gratings (FBG) sensors. They measure the wavelength change induced by deformation or other data. The fibres have been stuck along the blade surface and connected to an inquiry system located in the hub of the wind turbine. The same sampling of 1 s is done.

Conclusion

Filtering and analysing of the data has been performed but it is still ongoing to identify the influence of wind speed on the deformations measured. Several gust events have been identified and we have now at our disposal several test cases with many measurements of wind speed, deformations and pressure at several locations.

The structural model of the wind blade that has been instrumented, has been implemented in the developed solver with a simple beam model. Indeed, one application of the present work is the study of the effects of gusts on blade deformations. A weak fluid-structure coupling is performed in the numerical tool, with two different solvers, one for the fluid and one for the structure, that exchange only the necessary informations. In order to reduce the size of the computational domain, the choice has been done to perform simulations of one rotating blade with an inertial domain. Adaptations have thus been done in the fluid and structural models to take into account the Coriolis and centrifugal forces in the computations. Simulations have so been performed in that framework with a uniform incoming wind speed and a real operating condition of the wind blade. A lot of numerical dissipation occurred in that simulation, which is due to the first-order accuracy of the solver and to the fact that the grid used was not very fine in the wake of the blade. An Adaptive Mesh Refinement process has been implemented in the solver and has been investigated with two and three dimensional cases. Promising results have been obtained, but a slight adaptation of the criterion should be done to find the appropriate thresholds of the criterion to refine and coarsen the grid for the rotating blade at high Reynolds flow. A simulation has then been performed with a gust-like event whose behaviour has been defined from the experimental data. The computational cost of the fluid-structure coupling is very high, therefore, a perspective of the present work is the use of Reduced Order Models with Proper Orthogonal Decomposition to model the structural beam model of the wind blade.

Perspectives

As we saw that the fluid-structure coupling is very costly, the use of Reduced Order Models (ROMs) would be of high interest. We could use both the experimental data and data from CFD to compute Proper Orthogonal Decomposition (POD) modes. The pressure at every Lagrangian marker that defines the geometry of the blade is indeed recorded during the simulation which allows to have at our disposal a high number of CFD data. Thanks to the experimental work, we also have significant experimental values with a lot of wind conditions. In a more general way, modelling of the future performance of a wind plant (taking into account the terrain, the wake of other wind turbines) is of high interest for Valorem. Since the computational time is a key parameter for the engineers that have to deal with a high number of wind resource assessments, ROMs could be an interesting field of research, coupled with the present numerical tool and with the experimental data collected.

The solver developed in the thesis constitutes a basic architecture with several features (LES, wall law, AMR, fluid-structure interaction). This base can now be improved quite easily, especially by having a second-order accurate Eulerian scheme to compute the advection term of the Navier-Stokes equations. A method is currently investigated in the Memphis team of INRIA. The velocity field is known at the cell-centre of each cell. The gradients of velocity are then computed on each cell centre with a Moving Least Square interpolation (that method is described in chapter 3, section 4). The velocity fluxes are thus rebuilt at every face centre thanks to the gradients and velocity field from cell centres. This is done in an upwind framework. Another upgrade would consist in getting second-order accuracy in time. For example, an Adams-Bashforth scheme could be implemented for the convective term and a Cranck-Nicholson method could be

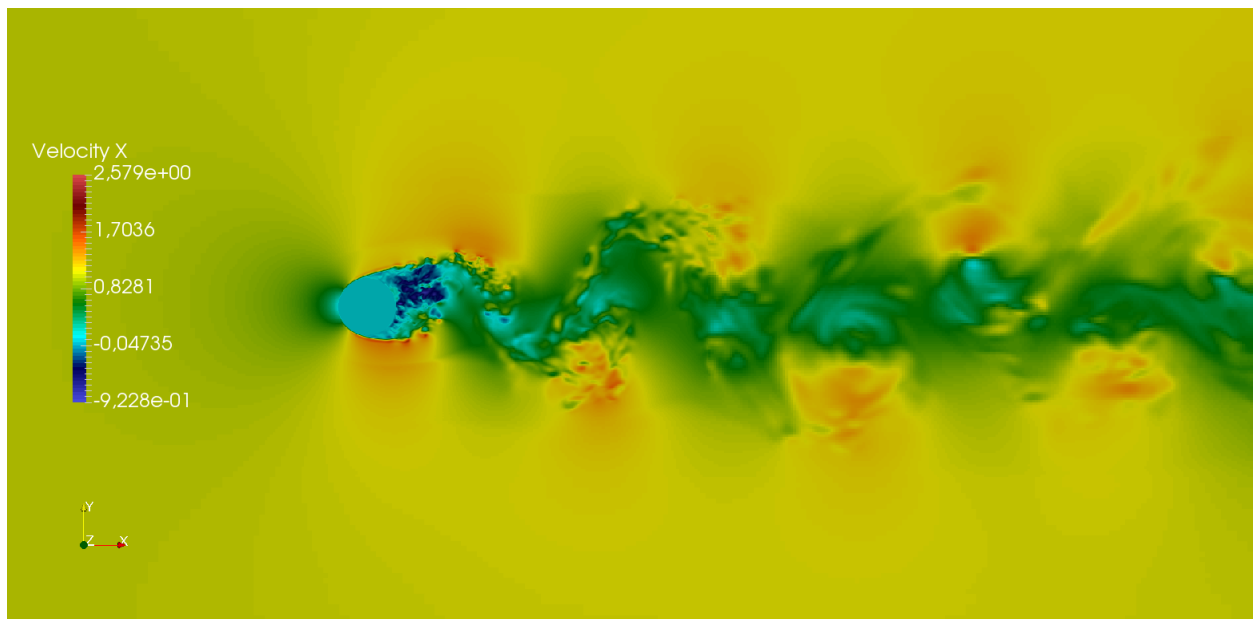


Figure 5.13: Streamwise velocity past a cylinder for the flow at $Re=140\,000$.

used to compute the viscous term. That would yield second-order accuracy in time for the prediction step. These upgrades have been implemented in the solver by colleagues in the Memphis team in a few weeks. New simulations have been performed and allowed to observe the small structures of the turbulence in the wake of cylinder at Reynolds 140 000 with the same octree mesh as presented in chapter 3. This can be seen in the figure below. That shows that the solver developed during the present thesis constitutes an interesting reliable numerical tool and continuous improvements can be conducted.

Bibliography

- [1] International Electrotechnical Commission 61400-12, 2005. p. 67, 68, 77, and 104
- [2] I. Abalakin and B. Koobus. The behavior of two near-wall models for k-epsilon prediction of stall. Technical report, INRIA, Sophia Antipolis, France, 2000. p. 38 and 115
- [3] P. Angot, C. H. Bruneau, and P. Fabrie. A penalization method to take into account obstacles in incompressible viscous flows. *Numerische Mathematik*, 81:497–520, 1999. p. 20
- [4] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Dave A. May, Lois Curfman McInnes, Richard Tran Mills, Todd Munson, Karl Rupp, Patrick Sanan, Barry F. Smith, Stefano Zampini, and Hong Zhang. PETSc Web page. <http://www.mcs.anl.gov/petsc>, 2018. p. 22
- [5] M. Bar-Lev and H. T. Yang. Initial flow field over an impulsively started circular cylinder. *Journal of Fluid Mechanics*, 72:625–647, 1975. p. 44
- [6] C. Batty. A cell-centred finite volume method for the poisson problem on non-graded quadtrees with second order accurate gradients. *Journal of Computational Physics*, (331):49–72, 2017. p. 17
- [7] Patrick Beaudan and Parviz Moin. Numerical experiments on the flow past a circular cylinder at sub-critical Reynolds number. 12 1994. p.
- [8] M. Bergmann. *Aerodynamic optimization using Reduced Order Model and optimal control. Application to the laminar wake flow around a circular cylinder*. PhD thesis, Institut National Polytechnique de Lorraine, France, 2004. p. 46, 47, and 48
- [9] M. Bergmann, J. Hovnanian, and A. Iollo. An accurate Cartesian method for incompressible flows with moving boundaries. *Communications in Computational Physics*, (15):1266–1290, 2014. p. 20
- [10] M. Bergmann and A. Iollo. Modeling and simulation of fish-like swimming. *Journal of Computational Physics*, (230):329–348, 2011. p. 16
- [11] E.T.G. Bot and O. Ceyhan. *Blade Optimization Tool (BOT): User Manual*. Energy Research Centre of the Netherlands, 2011. p. 8
- [12] M. Braza, P. Chassaing, and H. Ha Minh. Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder. *Journal of Fluid Mechanics*, 165:79–130, 1986. p. 47 and 48
- [13] Michael Breuer. Large eddy simulation of the subcritical flow past a circular cylinder: Numerical and modeling aspects. 28:1281–1302, 12 1998. p. 60

-
- [14] D. L. Brown, R. Cortez, and M. L. Minion. Accurate projection methods for the incompressible Navier–Stokes equations. *Journal of Computational Physics*, (168):464–499, 2001. p. 21
- [15] Rubens Campregheer, Julio Militzer, Sérgio Mansur, and Aristeu Neto. Computations of the flow past a still sphere at moderate Reynolds numbers using an Immersed Boundary Method. 31, 10 2009. p. 54
- [16] Brian Cantwell and Donald Coles. An experimental study on entrainment and transport in the turbulent near wake of a circular cylinder. 136, 12 1983. p. 59 and 60
- [17] J. E. Castillo. *Mathematical Aspects of Numerical Grid Generation*. Society for Industrial and Applied Mathematics, San Diego, California, 1991. p. 13
- [18] O. Ceyhan. Towards 20MW wind turbine: High Reynolds number effects on rotor design. Energy research centre of the Netherlands, 2012. p. 34
- [19] N. M. Chaderjian. Advances in rotor performance and turbulent wake simulation using DES and Adaptive Mesh Refinement. NASA Ames Research Center, 2012. p. 16 and 18
- [20] H. Choi and P. Moin. Grid-point requirements for large eddy simulation: Chapman’s estimates revisited. *Physics of Fluids*, 24, 2012. p. 37
- [21] A. J. Chorin. Numerical solution of the Navier–Stokes equations. *Mathematics of Computation*, (22):745–762, 1968. p. 16 and 21
- [22] J. Chou, C. Chiu, I. Huang, and K. Chi. Failure analysis of wind turbine blade under critical wind loads. *Engineering Failure Analysis*, (27):99–118, 2013. p. 2 and 65
- [23] J. Chou and W. Tu. Failure analysis and risk management of a collapsed large wind turbine tower. *Engineering Failure Analysis*, (18):295–313, 2011. p. 2 and 65
- [24] Yves Coudière, Jean-Paul Vila, and Philippe Villedieu. Convergence rate of a finite volume scheme for a two dimensional convection-diffusion problem. *ESAIM: M2AN*, 33(3):493–516, 1999. p. 26 and 27
- [25] R. de Frias Lopez. *A 3D finite beam element for the modelling of composite wind turbine wings*. PhD thesis, KTH, Stockholm, Sweden, 2013. p. 83, 84, 86, and 116
- [26] M. D. de Tullio. *Development of an immersed boundary method for the solution of the preconditioned Navier–Stokes equations*. PhD thesis, Politecnico di Bari, Italy, 2006. p. 38, 39, and 115
- [27] J. N. Sorensen et al. Simulation of wind turbine wakes using the actuator line technique. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 373(2035), 2015. p. 12 and 18
- [28] M.M. Hand et al. Unsteady aerodynamics experiment phase VI: Wind tunnel test configurations and available data campaigns. Technical report, National Renewable Energy Laboratory, 2001. p. 12, 13, and 115
- [29] E.A. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof. Combined Immersed-Boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics*, 161(1):35 – 60, 2000. p. 54

BIBLIOGRAPHY

- [30] M. Falcone and R. Ferretti. Convergence analysis for a class of high-order semi-lagrangian advection schemes. *SIAM Journal on Numerical Analysis*, 35:909–940, 1998. p. 34
- [31] R. A. Finkel and J. L. Bentley. Quad trees. A data structure for retrieval on composite keys. *Acta Informatica*, 4:1–9, 1974. p. 13
- [32] Bengt Fornberg. Steady viscous flow past a sphere at high Reynolds numbers. *Journal of Fluid Mechanics*, 190:471–489, 1988. p. 54
- [33] P. L. George. *Automatic mesh generation: application to finite element methods*. Chichester, New York, 1991. p. 13
- [34] M. Germano, U. Piomelli, P. Moin, and W. H. Cabot. A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids*, 3:1760–1765, 1991. p. 36
- [35] A. Gossier. Moving Least-Squares: A numerical differentiation method for irregularly spaced calculation points. Technical report, Sandia National Laboratories, California and New Mexico, United States, 2001. p. 51 and 53
- [36] A. Guittet, M. Theillard, and F. Gibou. A stable projection method for the incompressible Navier–Stokes equations on arbitrary geometries and adaptive quad/octrees. *Journal of Computational Physics*, (292):215–238, 2015. p. 17 and 18
- [37] J.W. He, R. Glowinski, R. Metcalfe, A. Nordlander, and J. Periaux. Active control and drag optimization for flow past a circular cylinder. *Journal of Computational Physics*, 163:83–117, 2000. p. 47 and 48
- [38] R. D. Henderson. Details of the drag curve near the onset of vortex shedding. *Physics of Fluids*, 7:2102–2104, September 1995. p. 48
- [39] R. D. Henderson. Nonlinear dynamics and pattern formation in turbulent wake transition. *Journal of Fluid Mechanics*, 352:65–112, 1997. p. 47 and 48
- [40] G. Hou, J. Wang, and A. Layton. Numerical methods for fluid-structure interaction - A review. *Communications in Computational Physics*, 12:337–377, 2012. p. 86
- [41] Grant Ingram. Wind turbine blade analysis using the Blade Element Momentum method. Technical report, Durham University, 2011. v 1.1. p. 9, 10, and 115
- [42] M. Arjomandi J. O. Mo, A. Choudhry and Y. H. Lee. Large eddy simulation of the wind turbine wake characteristics in the numerical wind tunnel model. *Journal of Wind Engineering and Industrial Aerodynamics*, (112):11–24, 2013. p. 2
- [43] Muhammad Khalid and Akhtar Imran. Characteristics of flow past a symmetric airfoil at low Reynolds number; A nonlinear perspective. In *ASME International Mechanical Engineering Congress and Exposition, Proceedings (IMECE)*, volume 7. 11 2012. p. 50
- [44] Sung-Eun Kim. Large eddy simulation of turbulent flow past a circular cylinder in subcritical regime. In *Collection of Technical Papers - 44th AIAA Aerospace Sciences Meeting*, volume 22, 01 2006. p. 59 and 60
- [45] Dilek Kurtulus. On the unsteady behavior of the flow around NACA 0012 airfoil with steady external conditions at $Re=1000$. 7:301–326, 09 2015. p. 50

-
- [46] S. Laizet and E. Lamballais. High-order compact schemes for incompressible flows: A simple and efficient method with quasi-spectral accuracy. *Journal of Computational Physics*, 228:5989–6015, 2009. p. 15
- [47] J Laursen, P Enevoldsen, and S Hjort. 3D CFD quantification of the performance of a multi-megawatt wind turbine. *Journal of Physics: Conference Series*, 75(1):012007, 2007. p. 16
- [48] M. Lentine, J. T. Grétarsson, and R. Fedkiw. An unconditionally stable fully conservative semi-Lagrangian method. *Journal of Computational Physics*, (230):2857–2879, 2011. p. 17 and 18
- [49] Lorenzo E.M. Lignarolo, Dhruv Mehta, Richard J.A.M. Stevens, Ali Emre Yilmaz, Gijs van Kuik, Søren J. Andersen, Charles Meneveau, Carlos J. Ferreira, Daniele Ragni, Johan Meyers, Gerard J.W. van Bussel, and Jessica Holierhoek. Validation of four LES and a vortex model against stereo-PIV measurements in the near wake of an actuator disc and a wind turbine. *Renewable Energy*, 94:510 – 523, 2016. p. 2
- [50] F. Losasso, F. Gibou, and R. Fedkiw. Simulating water and smoke with an octree data structure. volume 23, pages 457–462. SIGGRAPH, 2004. p. 17
- [51] L. M. Lourenco and C. Shih. Characteristics of the plane turbulent near wake of a circular cylinder, a particle image velocimetry study. 1993. Published in Beaudan and Moin (1994). p. 57
- [52] Dahai Luo, Chao Yan, Hongkang Liu, and Rui Zhao. Comparative assessment of PANS and DES for simulation of flow past a circular cylinder. 134:65–77, 11 2014. p. 60
- [53] J.A. Michelsen. Basis3d—a platform for development of multiblock PDE solvers. Technical report, Technical University of Denmark, Lyngby, Denmark, 1992. p. 16
- [54] Robert Mikkelsen. *Actuator Disk Methods Applied to Wind Turbines*. PhD thesis, Technical University of Denmark, 2003. p. 5, 8, and 12
- [55] C. Min and F. Gibou. A second order accurate projection method for the incompressible Navier–Stokes equations on non-graded adaptive grids. *Journal of Computational Physics*, (219):912–929, 2006. p. 17 and 18
- [56] C. Min, F. Gibou, and H. Ceniceros. A supra-convergent finite difference scheme for the variable coefficient poisson equation on non-graded grids. *Journal of Computational Physics*, (218):123–140, 2006. p. 17
- [57] R. Mittal, H. Dong, and et al. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *Journal of Computational Physics*, (227):4825–4852, 2008. p. 20
- [58] S. Mittal and T. E. Tezduyar. Massively parallel finite element computation incompressible flows involving fluid-body interactions. *Computer Methods in Applied Mechanics and Engineering*, 112:253–282, 1994. p. 50
- [59] P.J. Moriarty and C. Hansen. *AeroDyn theory manual*. National Renewable Energy Laboratory, Colorado, USA, 2005. p. 8
- [60] G. M. Morton. A computed oriented geodetic data base and a new technique in file sequencing. Technical report, IBM Ltd, 1966. p. 14

BIBLIOGRAPHY

- [61] N. M. Newmark. A method of computation for structural dynamics. *Journal of the Engineering Mechanics Division*, (85):67–94, 1959. p. 86
- [62] C. Norberg. Effects of Reynolds number and a low- intensity free-stream turbulence on the flow around circular cylinder. Technical report, Department of Applied Thermodynamics and Fluid Mechanics, Chalmers University of Technology, Sweden, 1987. p. 56
- [63] S. E. Norris, J. E. Cater, K. A. Stol, and C. P. Unsworth. Wind turbine wake modelling using Large Eddy Simulation. University of Auckland, 2010. 17th Australasian Fluid Mechanics Conference. p. 2
- [64] A. Oliveiro and A. M. Baptista. A comparison of integration and interpolation Eulerian-Lagrangian methods. *International Journal for Numerical Methods in Fluids*, (3):183–204, 1995. p. 17
- [65] M. A. Olshanskii, K. M. Terekhov, and Y. V. Vassilevski. An octree-based solver for the incompressible Navier–Stokes equations with enhanced stability and low dissipation. *Computers and Fluids*, (84):231–246, 2013. p. 17 and 18
- [66] Hilde Ouvrard, Bruno Koobus, Alain Dervieux, and Maria Vittoria Salvetti. Classical and variational multiscale LES of the flow around a circular cylinder on unstructured grids. *Computers and Fluids*, 39(7):1083 – 1094, 2010. p. 56
- [67] Noma Park, Sungwon Lee, Jungil Lee, and Haecheon Choi. A dynamic subgrid-scale eddy viscosity model with a global model coefficient. *Physics of Fluids*, 18(12):109–125, 2006. p. 56
- [68] P. Parnaudeau, J. Carlier, D. Heitz, and E. Lamballais. Experimental and numerical studies of the flow over a circular cylinder at Reynolds number 3900. *Physics of Fluids*, 20:085101–085101–14, 2008. p. 57
- [69] Charles S Peskin. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10(2):252 – 271, 1972. p. 16
- [70] Stefano Pezzano. Aeroelastic modelling of a wind turbine. Master’s thesis, Politecnico di Torino, March 2018. p. 86 and 89
- [71] P. Ploumhans and G. S. Winckelmans. Vortex methods for high-resolution simulations of viscous flow past bluff bodies of general geometry. *Journal of Computational Physics*, 165:354–406, 2000. p. 44
- [72] P. Ploumhans, G.S. Winckelmans, J.K. Salmon, A. Leonard, and M.S. Warren. Vortex methods for Direct Numerical Simulation of three-dimensional bluff body flows: Application to the sphere at $Re=300$, 500, and 1000. *Journal of Computational Physics*, 178(2):427 – 463, 2002. p.
- [73] Stephen B. Pope. *Turbulent Flows*. Cambridge University Press, 2000. p. 35
- [74] S. Popinet. Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries. *Journal of Computational Physics*, (190):572–600, 2003. p. 13, 14, 16, 18, and 115
- [75] B. N. Rajani, A. Kandasamy, and S. Majumdar. LES of flow past circular cylinder at $Re = 3900$. *Journal of Applied Fluid Mechanics*, 9:1421–1435, 2016. p. 56

-
- [76] B. Sanderse. Aerodynamics of wind turbine wakes - Literature review. Technical report, Energy Research Centre of the Netherlands, 2009. p. 5 and 8
- [77] K. Schneider. Numerical simulation of the transient flow behaviour in chemical reactors using a penalisation method. *Computers and Fluids*, 34:1223–1238, 2005. p. 15
- [78] D.A. Simms, M.M. Hand, L.J. Fingersh, and D.W. Jager. Unsteady aerodynamics experiment phase II-IV: Test configurations and available data campaigns. Technical report, National Renewable Energy Laboratory, 1999. p. 71
- [79] J. Smagorinsky. General circulation experiments with the primitive equations. *Monthly Weather Review*, 91(3):99–164, 1963. p. 36
- [80] D. O. Snyder and G. Degrez. Large-eddy simulation with complex 2-d geometries using a parallel finite-element/spectral algorithm. *International Journal for Numerical Methods in Fluids*, 41(10):1119–1135, 2003. p. 56
- [81] J. N. Sorensen and W. Z. Shen. Numerical modelling of wind turbine wakes. *Journal of Fluids Engineering*, (124):393–399, 2002. p. 12
- [82] J.N. Sorensen. Aerodynamic aspects of wind energy conversion. *Annual Review of Fluid Mechanics*, (43):427–448, 2011. p. 8
- [83] J.N. Sorensen and A. Mikken. Unsteady actuator disc model for horizontal axis wind turbine. *Journal of Wind Engineering and Industrial Aerodynamics*, (39):139–149, 1992. p. 5, 8, and 12
- [84] N. N. Sorensen. *General purpose flow solver applied to flow over hills*. PhD thesis, Technical University of Denmark, Lyngby, Denmark, 1995. p. 16
- [85] D. B. Spalding. A single formula for the Law of the Wall. *Journal of Applied Mechanics*, 28:455–458, 1961. p. 38
- [86] Takao Suzuki, Akira Sanse, Takashi Mizushima, and Fujio Yamamoto. Unsteady PTV velocity field past an airfoil solved with DNS: Part 2. Validation and application at Reynolds numbers up to Re 10000. In *Experiments in Fluids - EXP FLUID*, volume 47, pages 977–994. 12 2009. p. 50
- [87] R. Témam. Sur l’approximation de la solution des équations de Navier–Stokes par la méthode des pas fractionnaires. *Archive for Rational Mechanics and Analysis*, (32):135–153, 1969. p. 16 and 21
- [88] A Travin, Michael Shur, Michael Strelets, and Philippe Spalart. Detached-eddy simulations past a circular cylinder. In *Flow Turbulence and Combustion*, volume 63, pages 293–313. 01 2000. p. 60
- [89] R. Nguyen van yen, D. Kolomenskiy, and K. Schneider. Approximation of the Laplace and Stokes operators with Dirichlet boundary conditions through volume penalization: a spectral viewpoint. *Numerische Mathematik*, 128:301–338, 2014. p. 20
- [90] A. W. Vreman. An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications. *Physics of Fluids*, 16:3670–3681, 2004. p. 36
- [91] C. H. K. Williamson. Oblique and parallel modes of vortex shedding in the wake of a circular cylinder at low Reynolds numbers. *Journal of Fluid Mechanics*, 206:579–627, 1989. p. 47

- [92] C. H. K. Williamson. Vortex dynamics in the cylinder wake. *Annual Review of Fluid Mechanics*, 28:477–539, 1996. p. 47
- [93] D. Xiu and G. E. Karniadakis. A semi-Lagrangian high-order method for Navier–Stokes equations. *Journal of Computational Physics*, (172):658–684, 2001. p. 17
- [94] Donghyun You and Parviz Moin. A dynamic global-coefficient subgrid-scale eddy-viscosity model for large-eddy simulation in complex geometries. *Physics of Fluids*, 19(6):065–110, 2007. p. 56

List of Figures

1	Evolution of the size of wind turbines over time. Source <i>www.windeurope.org</i> . . .	2
1.1	Sketch of the aerodynamic forces exerted by the flow around a wind blade airfoil	6
1.2	$C_P - \lambda$ curve for the Euros EU120 wind blade. Source <i>www.euros.de</i>	7
1.3	Example of power curve for some usual wind turbine.	8
1.4	Axial stream tube around a wind turbine [41]	9
1.5	Blade element theory [41].	10
1.6	Comparison of predicted power for BEM and GAD models with experimental data of NREL UAE phase 6 (experimental process described in [28]).	13
1.7	Example of quadtree discretization and corresponding tree representation. Found in [74].	14
1.8	Sketch of the computational domain	15
2.1	Example of mesh configuration in 2D with stencils	24
2.2	Examples of quadtree grids used for validation	25
2.3	Example of mesh configuration in 2D with stencils for Laplacian operator	26
2.4	Example of error distribution between a Laplace operator discretization and an analytical expression and numerical solution with a quadtree grid configuration corresponding to tree level 7	28
2.5	Example of mesh configuration in 2D with stencils for semi-Lagrangian scheme .	29
2.6	Example of octree grid configuration used for validation of the solver	32
2.7	Example of mesh configuration in 3D with stencils for Laplace operator	33
2.8	Turbulent boundary layer profiles for different Reynolds numbers	37
2.9	Reichardt's wall law. Found in [2].	38
2.10	Implementation of the wall law. Found in [26].	39
3.1	Velocity field of the Taylor-Green vortex on a quadtree grid (tree level = 7) after 0.1 s of simulation	42
3.2	Quadtree mesh around a two-dimensional cylinder (minimum tree level = 7) . . .	44
3.3	Drag coefficient history at $Re = 550$ with the numerical solution of reference and the computed results with 3 different grid sizes	45
3.4	Isobars and contour lines of z -component of vorticity obtained at $t = 100$ and $Re = 100$ for the flow around a circular cylinder	46
3.5	Drag and Lift coefficients history at $Re = 100$	46
3.6	Isobars and contour lines of z -component of vorticity obtained at $t = 100$ and $Re = 200$ for the flow around a circular cylinder	47
3.7	Drag and Lift coefficients history at $Re = 200$	48
3.8	Isobars and contour lines of z -component of vorticity obtained at $t = 100$ and $Re = 600$ for the flow around a circular cylinder	49
3.9	Drag and Lift coefficients history at $Re = 600$	49

3.10	Simulation of a blade in a rotating frame, y-Velocity at plane $z = -5$	52
3.11	Octree mesh used for the flow around a sphere - Tree level 5	53
3.12	Time history of drag coefficients for the flow at $Re = 500$ with 3 different grid refinements and 2 different CFL conditions	54
3.13	Flow past a sphere at $Re = 500$	55
3.14	Pressure coefficient around the cylinder at $Re = 3900$. Angle of π corresponds to the wake of the cylinder, whereas angle of 0 is the incoming flow.	57
3.15	Wake profile of streamwise velocity at different positions obtained by an average over 9 vortex sheddings after a preliminary simulation for the flow past a cylinder at $Re = 3900$	58
3.16	Comparison of x-Velocity field obtained with the Cartesian and the octree codes for the flow around a cylinder at $Re = 140000$	60
3.17	Wake profile for the flow past a cylinder at $Re = 140000$ at different locations obtained by an average over 5 vortex shedding after a preliminary simulation	62
3.18	Wake profile for the flow past a cylinder at $Re = 140000$ at different locations obtained after a preliminary simulation and by an average over the 5 first vortex shedding (a, c and e), and over 5 next vortex shedding (b, d and f)	63
4.1	Computational domain with $d =$ rotor diameter + a few meters	66
4.2	Topographic view of the wind plant obtained with Google Earth [®] and wind rose corresponding to the site.	69
4.3	Fiber Bragg Grating principle. n corresponds to the index of refraction and Λ is the period of n variation. Found on www.fbgs.com	70
4.4	Optical fibre technology	70
4.5	Position of the strain gauges on the blade	71
4.6	Position of pressure sensors on the blade	72
4.7	Final position of the met mast and wind rose of the area.	73
4.8	Photo of the met mast after its installation	73
4.9	Photo of the sensors on the met mast	74
4.10	Aerial work platform positioning	75
4.11	Photo of the pressure side of the wind blade after installation of sensors	76
4.12	Longitudinal deformations from sensors at different locations as a function of wind speed	77
4.13	Time history of deformation data from sensor located on the leading edge at 7m of the blade length, of wind speed data reconstructed at the wind turbine and frequency (with a normalized scale) during a gust and with weak wind.	79
4.14	Deformations on the blade at several location as a function of blade position with weak wind (a and c) or during gust events (b and d)	81
5.1	Blade cross section with suggested reference system including definition of flapwise and lagwise bending axes. Source [25].	84
5.2	Partitioned scheme implemented in the 3D solver	87
5.3	Sketch of the computational domain for the flow past a rotating blade.	90
5.4	Display of the octree mesh used for the simulation of the flow past a rotating blade.	91
5.5	Display of the flow around a rotating blade with $U_\infty = 8 \text{ m.s}^{-1}$ and $\Omega = 21.6$ rpm after 3.5 revolutions of the blade.	93
5.6	Time history of the power extracted by the simulated wind turbine.	94
5.7	Velocity magnitude for the flow past a circular cylinder in 2D at $Re = 200$ showing the AMR process with superimposition of the mesh at the top	95

5.8	Beginning of the simulation with AMR process for the case of the flow around a sphere at $Re = 5000$	96
5.9	Middle of the simulation with AMR process for the case of the flow around a sphere at $Re = 5000$	97
5.10	End of the simulation with AMR process for the case of the flow around a sphere at $Re = 5000$	98
5.11	AMR process for the rotating blade with an operating condition at $U_\infty = 6 \text{ m.s}^{-1}$, $\Omega = 16.2 \text{ rpm}$ and $Re = 1000$	100
5.12	Visualization of the flow around a rotating blade with a gust-like event.	101
5.13	Streamwise velocity past a cylinder for the flow at $Re=140\ 000$	106
5.14	Évolution de la taille des éolienne au cours du temps. Source www.windeurope.org	122
5.15	Comparaison de la vitesse selon l'axe x obtenue avec les codes Cartesien et octree pour le cas de l'écoulement autour d'un cylindre à $Re = 140000$	125
5.16	Profils de vitesses dans le sillage du cylindre à $Re = 140000$ obtenus à plusieurs positions et en moyennant sur 5 lachés tourbillonnaires après une simulation préliminaire	126

List of Tables

2.1	Norm of the error and order of the divergence operator for 2D Cartesian grid . . .	25
2.2	Norm of the error and order of the divergence operator for 2D quadtree grid	25
2.3	Norm of the error and order of the Laplace operator for 2D Cartesian grid	28
2.4	Norm of the error and order of the Laplace operator for 2D quadtree grid	28
2.5	Norm of the error and order of the semi-Lagrangian for 2D quadtree grid	30
2.6	Norm of the error and order of the divergence operator for 3D octree grid	32
2.7	Norm of the error and order of the Laplace operator for 3D octree grid	34
2.8	Norm of the error and order of the semi-Lagrangian scheme for 3D octree grid . . .	34
3.1	Norm of the spatial error and orders of the Taylor-Green vortex for 2D quadtree grid	42
3.2	Norm of the spatial error and orders for 3D octree grid	43
3.3	Two-dimensional grid characteristics and computational time for a simulation of 6 s	45
3.4	Comparison of Strouhal number and drag coefficient for the flow past a circular cylinder at $Re = 100$	47
3.5	Comparison of Strouhal number and drag coefficient for the flow past a circular cylinder at $Re = 200$	48
3.6	Comparison of Strouhal number and drag coefficient for the flow past a circular cylinder at $Re = 600$	48
3.7	Comparison of lift coefficient for the flow past a NACA0012 airfoil at $Re = 1000$ with different angles of attack	50
3.8	Three-dimensional grid characteristics and computational time for a simulation of 100 s	53
3.9	Comparison of averaged drag coefficient for the flow past a sphere at $Re = 500$. .	54
3.10	Comparison of averaged drag coefficient for the flow past a cylinder at $Re = 3900$	56
3.11	Comparison of averaged drag coefficient (over 9 vortex sheddings) for the flow past a cylinder at $Re = 140000$	60
4.1	Characterization of gust events	78
4.2	Mean values of frequency (noted f) and amplitude (noted Amp) of deformations during gusts or with weak wind for each sensor	80

Résumé détaillé

Résolution des équations de Navier-Stokes sur maillage octree : vers une application à la modélisation d'une pale d'éolienne

Contexte de la recherche

Pendant la Conférence de Paris de 2015 sur les changements climatiques, presque tous les états du monde ont trouvé un accord sur la réduction des changements climatiques. Un des objectifs annoncé est de limiter le réchauffement climatique à moins de 2°C comparé à l'ère pré-industrielle. L'accord vise également à atteindre une émission de gaz à effet de serre due à l'activité humaine à un niveau nul. Certes, la conférence ne planifie pas d'actions concrètes pour les pays qui ont validé l'accord, et il n'est pas prévu de sanction dans l'hypothèse où un état n'effectuerait pas d'effort suffisant. Tout de même, cet accord montre que le monde est en train de réaliser l'importance pour tous de limiter les changements climatiques. Afin d'atteindre les objectifs affichés, le développement des énergies renouvelables va jouer un rôle fondamental et il semble raisonnable de penser qu'un scénario gagnant serait le développement d'un mix énergétique avec du solaire, de l'éolien, de l'hydraulique, de la méthanisation ... À l'intérieur de ce mix, le secteur éolien présente plusieurs avantages qui le distingue des autres moyens de production. La ressource en vent est en effet locale et son coût baisse de plus en plus grâce aux améliorations technologiques. De plus, contrairement à l'énergie solaire, le vent est disponible à toute heure. La France a aussi l'avantage de disposer de nombreux couloirs de vent distincts et cela permet de disposer presque toujours au moins sur une partie du territoire de la ressource éolienne. Le développement de la filière éolienne offshore pourrait également permettre d'accélérer en France la production éolienne. L'éolien offshore est en effet en train de bénéficier d'avancées technologiques qui permettent de baisser les coûts d'installation petit à petit. Actuellement, le coût au MWh de l'éolien offshore est plus bas que le coût produit par l'EPR de Flamanville (France) en cours de construction.

Ce développement de la filière éolienne entraîne ainsi de nombreux avancements technologiques. La figure 5.14 illustre ce phénomène avec l'évolution de la taille des éoliennes au cours du temps. Grâce aux longues pales, les éoliennes sont en effet capables de produire même lorsque la ressource en vent est limitée. En ce qui concerne l'entreprise Valorem, elle intervient tout le long de la chaîne de valeur des énergies renouvelables, du développement des parcs jusqu'à l'exploitation et la maintenance en passant par la construction. Le développement d'un parc éolien consiste à identifier des zones d'implantations potentielles avec les contraintes associées et obtenir les différentes autorisations administratives nécessaires. Lors de cette étape, plusieurs études sont effectuées afin d'estimer la ressource en vent disponible et la future production d'énergie (en fonction des modèles d'éoliennes et de leur disposition sur le terrain). Des études environnementales sont faites pour estimer les impacts du parc. Également, des études de raccordement électriques sont intégrées lors de cette phase de développement pour préparer le raccordement du parc au réseau. Valorem possède plusieurs parcs (éoliens majoritairement,

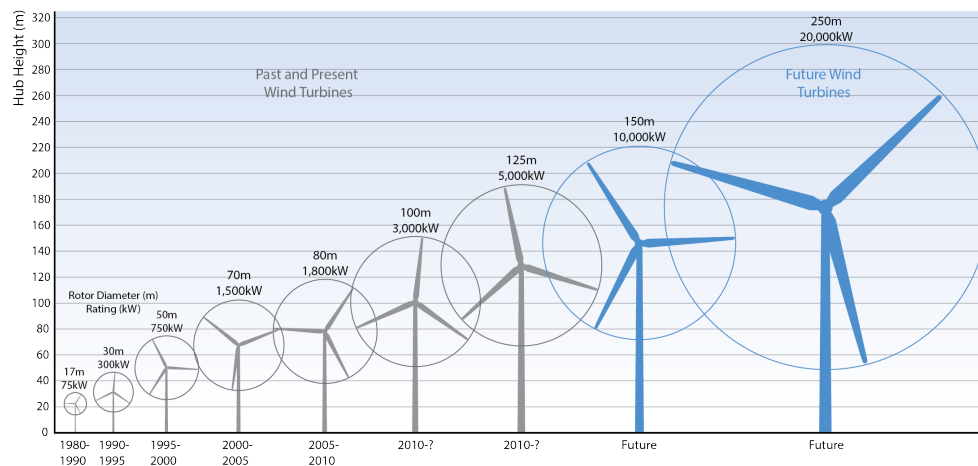


Figure 5.14: Évolution de la taille des éoliennes au cours du temps. Source *www.windeurope.org*

mais également solaires et hydraulique). Concernant l'éolien, il est essentiel pour Valorem de pouvoir prédire, de la manière la plus précise possible, la future production du parc. En effet, le modèle économique est très dépendant de cette information et un modèle économique valide est nécessaire pour la viabilité de Valorem. Ainsi, l'utilisation d'outils numériques adéquat est essentiel pour les études de vent. Pour une entreprise comme Valorem qui possède plusieurs parcs éoliens, deux sujets sont donc très importants : la durée de vie des pales d'éoliennes, qui sont la partie la plus fragile et l'optimisation de la performance.

Démarche adoptée

Dans un premier temps, l'aérodynamique des éoliennes est étudiée afin de comprendre les paramètres importants à calculer. Les outils numériques simplifiés permettant d'effectuer des calculs rapidement pour estimer la puissance extraite par une éolienne sont étudiés, en particulier la méthode Blade Element Momentum, qui est beaucoup utilisée par les industriels du domaine. Les modèles numériques de CFD (Computational Fluid Dynamics) sont ensuite considérés. Comme il existe de très nombreuses méthodes, nous nous sommes intéressées à celles dont l'application porte sur la modélisation des éoliennes, et également aux schémas numériques qui utilisent une discrétisation sur des maillages octree. Ces types de maillages ont en effet été étudiés et montrent des avantages intéressants pour notre application.

Lors de cette thèse, des méthodes numériques ont été développées afin de résoudre les équations de Navier-Stokes dans un contexte de frontières immergées. Le choix de cette méthode est justifié dans le chapitre 2. Un code de calcul utilisant des méthodes type volumes finis avec une discrétisation sur des maillages octree a été développé dans cette thèse. Dans un premier temps un code en deux dimensions a été développé. Les méthodes d'interpolations avec les stencils utilisés sont décrits. Chaque opérateur qui permet de résoudre les équations de Navier-Stokes est décrit et validé dans un premier temps, en utilisant des fonctions analytiques. Ces schémas numériques sur maillage octree sont ensuite modifiés afin de résoudre le problème en trois dimensions. Comme notre domaine d'application est l'écoulement de l'air autour des pales d'éoliennes, des grands nombres de Reynolds doivent être simulés. Une méthode Large Eddy Simulation (LES) avec un modèle de turbulence pour les petites structures de type Vreman a été implémentée. Afin de modéliser les petites échelles proche de la paroi, une méthode dite des wall functions a été développée. En effet, l'utilisation de maillages type octree ne permet pas de modéliser correctement la couche limite proche paroi.

Une validation détaillée des code 2D et 3D a ensuite été effectuée. Cette étape est une partie essentielle du travail de la thèse. Le code de calcul 2D a été testé pour le cas de l'écoulement autour d'un cylindre 2D pour plusieurs régimes d'écoulement et ensuite pour le cas de l'écoulement autour d'un profil aérodynamique. Grâce aux données provenant de la littérature, les résultats des simulations ont pu être comparés avec différentes géométries et différents nombre de Reynolds. Le code de calcul 3D a ensuite également été largement contrôlé en procédant par étapes. Dans un premier temps, le cas de l'écoulement autour d'une sphère à faible nombre de Reynolds (500) a été considéré. Ensuite, des simulations ont été effectuées pour simuler l'écoulement autour d'un cylindre à $Re = 3900$. Le modèle de turbulence, ainsi qu'une loi de paroi visqueuse ont été intégrés aux simulations. Enfin, un haut nombre de Reynolds de 140 000 a été simulé. Les données de la littérature sur ce cas ont également été utilisées pour discuter sur les différences entre un code de calcul d'ordre élevé sur maillage Cartésien et la présente thèse.

Un travail expérimental a été conduit pendant la thèse. En effet, très peu de données expérimentales sont disponibles avec à la fois la connaissance de la géométrie de la pale, le modèle de structure et des mesures sur la pale. Et aucun travail disponible ne permet d'avoir accès, dans le même temps, à la donnée de la vitesse du vent qui arrive sur la pale. Le contexte de la participation de Valorem au projet européen AeroGust¹ a permis de collecter des données expérimentales sur une éolienne en fonctionnement. Une réflexion a été menée afin de déterminer le cahier des charges permettant de mener à bien cette étude expérimentale. Plusieurs solutions techniques ont été investiguées afin d'être capables de mesurer le vent et également pour obtenir les données sur la pale et le choix final est présenté. L'installation sur le terrain a été organisée et effectuée. Les données ont ensuite été récupérés pendant plusieurs mois et analysées.

Une application du présent travail est la modélisation de l'écoulement autour des éoliennes. Comme nous avons à disposition des données expérimentales de déformations d'une pale d'éolienne en fonctionnement, le modèle de structure de cette pale a été implémenté dans le code octree. Un couplage fluide-structure est développé. Comme l'écoulement autour d'une éolienne complète aurait nécessité un coût de calcul trop important par rapport aux ressources disponibles, une seule pale a été étudiée dans nos simulations. De plus, un système de référence tournant a été mis en place pour focaliser le domaine sur la pale. Une fonctionnalité intéressante a également été développée avec un processus d'adaptation automatique du maillage (AMR). Cela permet d'optimiser le maillage en étant le plus précis possible dans les zones d'intérêt (comme le sillage) tout en diminuant le nombre de mailles loin de la pale. Les réelles conditions de fonctionnement de l'éolienne ont ainsi été simulées.

Principaux résultats obtenus

Les équations de Navier-Stokes sont écrites dans le contexte d'une frontière immergée avec une méthode de pénalisation de l'obstacle. On considère un domaine Ω qui comprend à la fois un fluide et un (ou plusieurs) obstacle(s). La frontière extérieure de ce domaine est notée Γ .

$$\begin{cases} \frac{d\mathbf{u}}{dt} + (\mathbf{u} \cdot \vec{\nabla})\mathbf{u} = -\frac{1}{\rho}\vec{\nabla}p + \nu\Delta\mathbf{u} + \frac{\chi_B}{\kappa}(\mathbf{u}_B - \mathbf{u}) & \text{dans } \Omega, \\ \nabla \cdot \mathbf{u} = 0 & \text{dans } \Omega, \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) & \text{dans } \Omega, \\ \mathbf{u} = \mathbf{u}_f(\mathbf{x}, t) & \text{sur } \Gamma_f, \end{cases}$$

¹www.aerogust.eu

où p est la pression, \mathbf{u} représente le champs de vitesse. La variable ρ est la masse volumique du fluide, ν sa viscosité cinématique et χ_B est la fonction caractéristique :

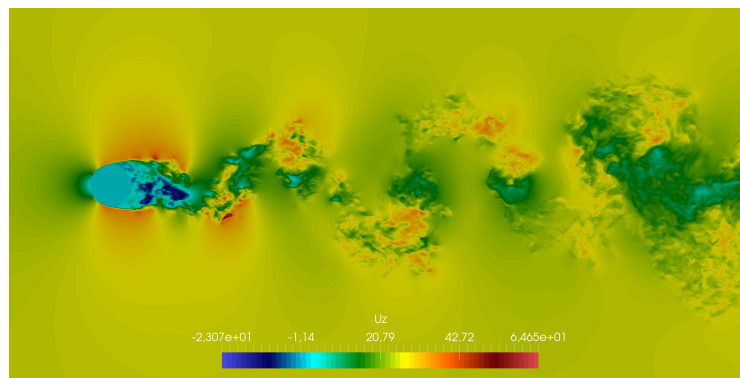
$$\chi_B = \begin{cases} 0 & \text{in the fluid domain} \\ 1 & \text{in the solid domain} \end{cases}$$

Ainsi, un code de calcul a été développé pour résoudre ces équations dans le contexte parallèle d'un maillage de type octree. La méthode des volumes finis a été principalement utilisée dans cette thèse. Le schéma numérique développé pour traiter l'opérateur de Laplace est basé sur une méthode DDFV (Discrete Duality Finite Volume) où un maillage dual en forme de diamant a été introduit et qui permet de calculer les gradients discrets au centre des faces de chaque cellule. Dans le cas du terme d'advection, un schéma semi-Lagrangien a été implémenté dans un premier temps. Cette méthode a en effet l'avantage d'éviter le calcul du terme de convection et de ne pas être soumis à la contrainte de stabilité où à la condition CFL (Courant-Friedrich-Lévy). Cependant, durant la phase de validation du solveur pour le cas 3D, un problème de stabilité s'est produit pour certains maillages octree. L'origine de ce problème de stabilité s'est avéré venir du schéma semi-Lagrangien. Il a donc été décidé de changer de méthode et de développer un schéma Eulérien upwind à l'ordre 1 à la place.

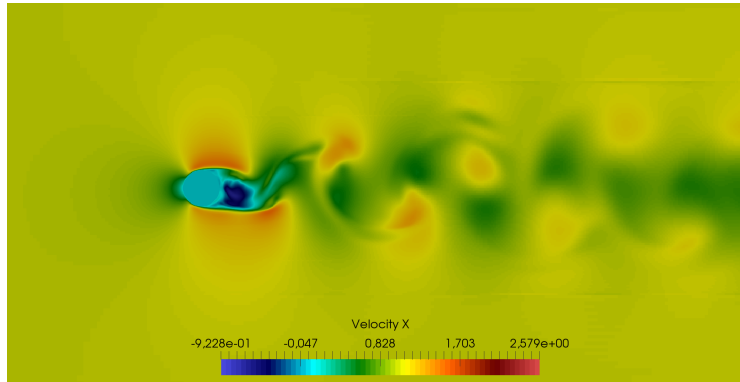
Une validation principale a été effectuée avec le cas de l'écoulement autour d'un cylindre à Reynolds 140 000. Le modèle de turbulence ainsi que la loi de paroi ont été intégrées dans les simulations et une comparaison est faite entre le code de calcul développée dans la thèse et un code de calcul Cartésien, d'ordre élevé qui a été développé au préalable de cette thèse. Ainsi, le code octree permet d'avoir un nombre de degrés de libertés significativement plus faible que dans le cas Cartésien. Le temps de calcul n'est toutefois pas seulement dépendant du nombre de mailles, le conditionnement des matrices a un impact également et elles sont bien mieux conditionnées dans le cas d'un maillage Cartésien. Une première visualisation sur la figure 5.15 permet d'observer que les petites échelles de la turbulence sont captées par le code Cartésien mais pas le code octree.

Des données expérimentales provenant de la littérature ont également été utilisées afin de comparer les profils de vitesse dans le sillage du cylindre avec les deux codes de calcul. Une première simulation a été effectuée avec les deux codes afin de passer la phase transitoire de démarrage des calculs. Les résultats visibles sur la figure 5.16 ont ensuite été obtenus avec une moyenne sur 5 lachés tourbillonnaires. Cela permet de valider le code de calcul développé dans cette thèse.

Le procédé d'adaptation automatique du maillage a été développé pour plusieurs applications. Un effort supplémentaire est encore nécessaire pour adapter le critère utilisé pour le rendre performant pour les hauts nombres de Reynolds dans le système de référence tournant. Des simulations préliminaires ont été effectuées sur une pale en conditions réelles de fonctionnement. Malgré une forte dissipation numérique due à l'ordre peu élevé des schémas numériques, la forme de l'écoulement observé est en adéquation avec la réalité. Le code de calcul développé constitue ainsi une base de travail avec de nombreuses fonctionnalités. Les schémas numériques développés pourront par la suite faire l'objet d'améliorations afin d'augmenter leur précision. La campagne de mesures expérimentales nous permet de disposer d'une importante base de données qui pourra être utilisée pour réaliser des futures simulations de conditions réelles de fonctionnement de l'éolienne et permettre de calibrer les modèles numériques.

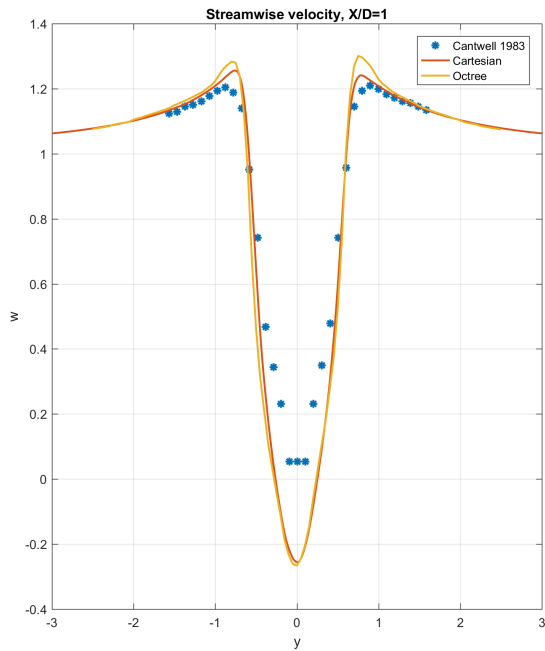


(a) Cartesian code

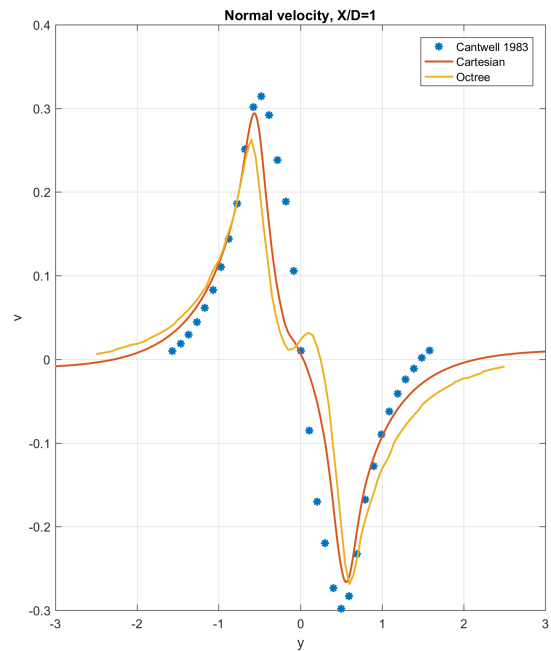


(b) Octree code

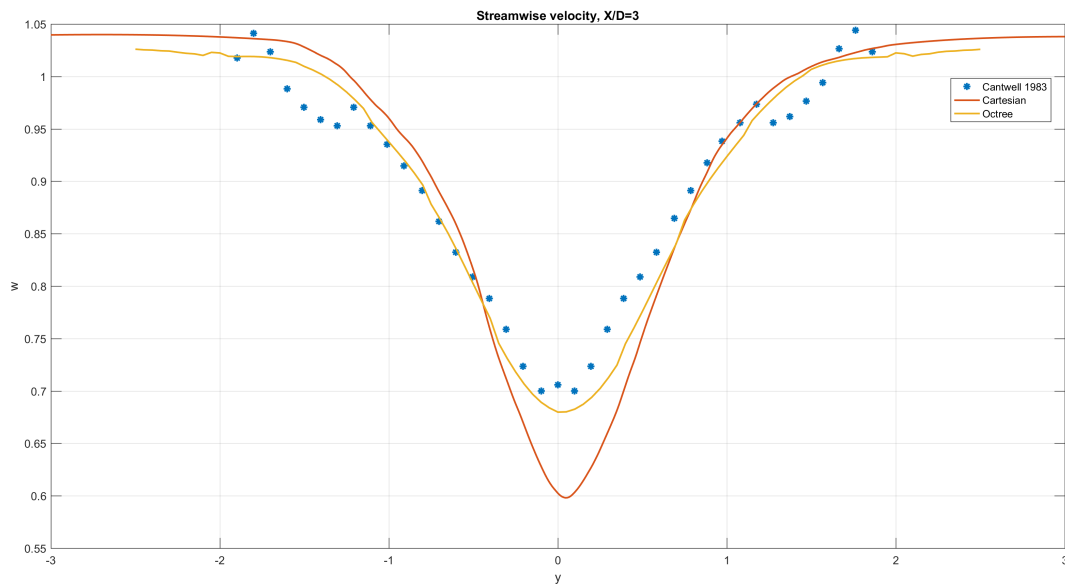
Figure 5.15: Comparaison de la vitesse selon l'axe x obtenue avec les codes Cartésien et octree pour le cas de l'écoulement autour d'un cylindre à $Re = 140000$



(a) Vitesse dans le sens de l'écoulement à $x = 1$



(b) Vitesse normale à $x = 1$



(c) Vitesse dans le sens de l'écoulement à $x = 3$

Figure 5.16: Profils de vitesses dans le sillage du cylindre à $Re = 140000$ obtenus à plusieurs positions et en moyennant sur 5 lachés tourbillonnaires après une simulation préliminaire

Titre : Résolution des équations de Navier-Stokes sur maillage octree : vers une application à la modélisation d'une pale d'éolienne

Résumé :

Le sujet de la thèse est le développement d'un outil numérique qui permet de modéliser l'écoulement autour des pales d'éoliennes. Nous nous sommes intéressés à la résolution des équations de Navier-Stokes en incompressible sur des maillages de type octree où les échelles plus petites en proche parois ont été modélisées par la méthode dite des wall functions. Un procédé d'adaptation automatique du maillage (AMR) a été développé pour affiner le maillage dans les zones où la vorticit  est plus importante. Le mod le de structure d'une pale d' olienne a  t   galement impl ment  et coupl  avec le mod le fluide car une application de l'outil num rique est l' tude des effets des rafales de vent sur les pales d' olienne. Un travail exp rimental a  t  men  sur une  olienne avec une mesure de vent en amont. Ces donn es permettent ainsi de calibrer et valider les mod les num riques d velopp s dans la th se.

Mots-cl s :

 coulements incompressibles; Maillage octree; Fronti re immerg e; Int ractions fluide-structure;  olienne

Title : Solving Incompressible Navier-Stokes Equations on Octree grids: Towards Application to Wind Turbine Blade Modelling

Abstract :

The subject of the thesis is the development of a numerical tool that allows to model the flow around wind blades. We are interested in the solving of incompressible Navier-Stokes equations on octree grids, where the smallest scales close to the wall have been modelled by the use of the so-called Wall Functions. An automatic Adaptive Mesh Refinement (AMR) process has been developed in order to refine the mesh in the areas where the vorticity is higher. The structural model of a real wind blade has also been implemented and coupled with the fluid model. Indeed, an application of the numerical tool is the study of the effects of wind gusts on blades. An experimental work has been conducted with an in-service wind turbine with the measurement of wind speed upstream. This data will allow to calibrate and validate the numerical models developed in the thesis.

Key words :

Incompressible flows; Octree grid; Immersed boundary; Fluid-Structure Interaction; Wind Turbine