POLITECNICO DI TORINO

Dipartimento di Ingegneria Meccanica e Aerospaziale

Corso di Laurea Magistrale
in Ingegneria Aerospaziale

Tesi di Laurea Magistrale

Aeroelastic Modelling of a Wind Turbine Blade

Relatore
Prof. Domenic D'Ambrosio
Corelatore
Prof. Angelo Iollo

Candidato
Stefano Pezzano

Marzo 2018

Contents

List of Figures

List of Tables

Introduction

Context

This project has been carried out in collaboration with the French Institute for Research in Computer Science and Automation (INRIA) in the framework of the European Horizon 2020 project AeroGust.

One of the most important source of structural load for both aircrafts and wind turbines is given by wind gusts, knowing with accuracy the aeroelastic response is therefore essential in the design phases of a new project. Rapid and precise tools need to be developed in order to aid the engineers in choosing the best configuration, the one that minimizes the gusts induced loads. The AeroGust project aims to improve the knowledge in the field of fluid-structure interaction in order to increase the competitivity of the aeronautical and wind energy industry, reducing the level of conservatism and the number of wind tunnel tests necessary to understand the aeroelastic response.
In this context, computational fluid dynamics (CFD) plays a major role, especially thanks to the ever increasing performance of modern CPUs, which allow to perform quick and fairly accurate simulations. In order to further reduce the costs related to CFD, innovative solutions can be employed, such as the immersed boundary method, with the aim of gradually reduce the time spent on mesh generation, and, moreover, fully exploit massively parallel architectures of supercomputers, using cartesian grids and avoiding remeshing at each timestep.
The main goal of this thesis is thus the investigation of a mathematical model for wind turbine simulation, based on the immersed boundary method, that could allow a considerable reduction in CFD costs while maintaining the accuracy of a Navier-Stokes solver, without recurring to low fidelity approaches for fluid-structure interaction. In this sense, the present work has to be seen as a small part of an ambitious project, that aims to change the methodologies in which aeroelasticity is investigated in both the aviation and wind energy worlds.

Structure of the thesis

The first chapter deals with the mathematical model proposed for the study of fluid-structure interaction, the immersed boundary method is presented with a particular attention to the penalized Navier-Stokes equations. Then the turbulence model is introduced, with an explanation of the Wall Modelled Large Eddy Simulation methodology. As a conclusion the Finite Element structural model is presented and the coupling between the two solver is illustrated.
The second chapter explains the validation of the fluid model, that has been implemented in a cartesian code and was already validated for low Reynolds flows and needed to be tested at high Reynolds conditions, using the popular flow past a cylinder benchmark, for which numerous experimental and numerical results can be found in literature. The validation of the structural model is shown too. In the third chapter the proposed model is extended to Octree grids, which are hierarchical cartesian grids, that allow local mesh refinements, increasing the efficiency and accuracy of simulations. The introduction of local refinements leads to the presence of level-jumps in the domain, requiring special treatments of these regions. where numerical instabilities can develop, causing simulations to diverge. The proposed Octree solver is tested on low and high Reynolds flows in order to understand the characteristics of the employed discretization.
In the fourth and last chapter a model for the simulation of a wind turbine blade in a rotating frame of reference is presented and discussed. The geometry of the blade is discretized starting from a CAD model, then the mathematical model is improved in order to take into account inertial reactions in both the structure solver and the Navier-Stokes model. As a conclusion, some preliminary simulations are run and their results are presented.

Chapter 1

The mathematical model

Over the last decades, many efforts have been made towards accurate numerical simulation of flows around moving bodies and fluid structure interaction in general. All the models developed can be divided in two distinct families, the first category is made by Arbitrary Lagrangian Eulerian methods, which are based on unstructured body fitted meshes that are deformed after each timestep. These methods are very accurate, but very hard to implement highly expensive from a computational viewpoint due to constant dynamic mesh adaptation and partitioning in case of massively parallelized simulations. The second family is represented by Immersed Boundary methods, where the computational grid is not body fitted, therefore the interface between the body and the fluid is buried within the cells of the mesh, allowing the use of simpler cartesian or octree grids, whose parallelization is straightforward and requires a low amount of memory. Furthermore, since the body is not a boundary of the domain anymore, there is no need of a constant dynamic adaptation.
Among the last category of models is possible to identify two approaches to deal with the presence of an immersed boundary, the first is the discrete forcing, which is applied to discretized Navier-Stokes equation in a manner that strongly depends on the spatial discretization, the second way is the continuous forcing, which is applied to the equations before discretization, and is almost independent of the spatial discretization. The main advantage of the discrete forcing, such as in the ghost-cell method proposed by Mittal et al. [1], 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 timestep 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. [2]. For the present work, a continuous forcing model was employed, the penalization method [3], which will be briefly described in the following section.

1.1 Governing equation of fluid motion

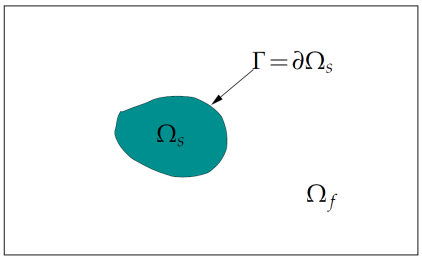


Figure 1.1: Domain of the problem

Since the present work deals with low Mach number flows, incompressible viscous Navier-Stokes equations are considered. Let Ω_f be the fluid domain and Ω_s be the solid domain, $\Omega = \Omega_f \mathrm{U} \Omega_s$ and $\Gamma = \partial\Omega_s$ be the boundary of the solid domain. The governing equations are:

$$\begin{cases} \nabla \cdot \mathbf{V} = 0 \text{ in } \Omega_f \\ \dfrac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \nabla)\mathbf{V} = -\dfrac{\nabla p}{\rho} + \nu \nabla^2 \mathbf{V} \text{ in } \Omega_f \\ \mathbf{V}(\mathbf{x}, 0) = \mathbf{V}_0(\mathbf{x}) \text{ in } \Omega_f \\ \mathbf{V} = \mathbf{V}_b(\mathbf{x}, t) \text{ in } \Omega_s \\ \mathbf{V} = \mathbf{V}_\Gamma(t) \text{ on } \Gamma \end{cases} \tag{1.1}$$

where $\mathbf{V}$ is the velocity vector, ρ is the fluid density and ν is the kinematic viscosity. The body velocity can be either imposed in case of an infinitely stiff solid, otherwise it is the result of the forces exerted by the fluid and therefore a proper structural model is necessary in order to evaluate body kinematics.
The entire system can be described as a single flow using the penalization method, where the solid body is considered as a porous fluid with very low permeability K:

$$\begin{cases} \nabla \cdot \mathbf{V} = 0 \\ \dfrac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \nabla)\mathbf{V} = -\dfrac{\nabla p}{\rho} + \nu \nabla^2 \mathbf{V} + \dfrac{\chi_B}{K}(\mathbf{V}_b - \mathbf{V}) \end{cases} \tag{1.2}$$

where χ_B is the characteristic function:

$$\chi_B = \begin{cases} 0 \text{ in } \Omega_f \\ 1 \text{ in } \Omega_s \end{cases} \tag{1.3}$$

It has been demonstrated by Angot et al. [3] that the system of equations 1.2 converges to the system 1.1 as $K \to 0$. The surface of the body is described using several points called Lagrangian markers, through which is possible to reconstruct a signed distance function φ, called level set. The relationship between the level set and the characteristic function is:

$$\chi_B = 1 - H(\varphi) \tag{1.4}$$

where H is the Heaviside function.
Using the level set function, it is possible to define the outward normal vector of the fluid-solid interface:

$$\mathbf{n} = \left(\frac{\nabla \varphi}{\|\nabla \varphi\|} \right)_{\varphi=0} \tag{1.5}$$

The penalized Navier-Stokes equations are solved numerically using a classical fractional step projection method. Space and time discretization will be discussed in detail in chapter 2 and chapter 3.

1.2 Turbulence model

As stated in [4], characteristic Reynolds numbers of a wind turbine flow in operative conditions are of the order of millions, which means that the fluid motion is characterized by the presence of turbulent boundary layers and multiscale vorticous structures in the wake, thus performing a direct numerical simulation is not possible. Given the highly unsteady nature of the flow that will be simulated, the most promising approach is the so-called Large Eddy Simulation (LES), that allows to resolve only the largest structures of the turbulence, which are the ones that carry the largest part of the energy and give the biggest contribution to the transport phenomena, whereas small structures, which are associated with turbulent energy dissipation, are modelled using a sub-grid model. The mathematical formulation of LES is based on a low pass filter operator G, so that filtered variables are defined as:

$$\overline{\psi} = \int_{x_1} \int_{x_2} \int_{x_3} G(\mathbf{r}, \mathbf{x}) \psi(\mathbf{x}) dx_1 dx_2 dx_3 \tag{1.6}$$

applying 1.6 to the system 1.1, filtered equations are obtained. 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:

$$\overline{\frac{\partial U_i}{\partial x_i}} = \frac{\partial \overline{U_i}}{\partial x_i} = 0 \tag{1.7}$$

where Einstein notation is employed. The filtered momentum equation is

$$\frac{\partial \overline{U_j}}{\partial t} + \frac{\partial \overline{U_i U_j}}{\partial x_i} = -\frac{1}{\rho} \frac{\partial \overline{p}}{\partial x_j} + \nu \frac{\partial^2 \overline{U_j}}{\partial x_i \partial x_i} \tag{1.8}$$

Due to the nonlinearity of the convective term, the form of the equation is different from the non-filtered one, 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:

$$\tau_{ij} = \overline{U}_i \overline{U}_j - \overline{U_i U_j} \tag{1.9}$$

so that equation 1.8 can be rewritten as:

$$\frac{\partial \overline{U_j}}{\partial t} + \frac{\partial \overline{U}_i \overline{U}_j}{\partial x_i} = -\frac{1}{\rho} \frac{\partial \overline{p}}{\partial x_j} + \nu \frac{\partial^2 \overline{U_j}}{\partial x_i \partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} \tag{1.10}$$

From a phisycal point of view the residual-stress tensor can be interpreted as the exchange of momentum at the filtered scale exerted by the subgrid turbulent structures. Let k_T be the residual kinetic energy:

$$k_T = \frac{1}{2}\tau_{ii} \tag{1.11}$$

then the residual-stress tensor can be decomposed into an isotropic component and an anisotropic one 1.12:

$$\tau_{ij}^s = \tau_{ij} - \frac{2}{3}k_T\delta_{ij} \tag{1.12}$$

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

$$p^* = \overline{p} + \frac{2}{3}\rho k_T \tag{1.13}$$

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

$$\frac{\partial\overline{U_j}}{\partial t} + \frac{\partial\overline{U}_i\overline{U}_j}{\partial x_i} = -\frac{1}{\rho}\frac{\partial p^*}{\partial x_j} + \nu\frac{\partial^2\overline{U_j}}{\partial x_i\partial x_i} + \frac{\partial\tau_{ij}^s}{\partial x_j} \tag{1.14}$$

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

$$\tau_{ij}^s = \nu_e\overline{S_{ij}} = \nu_e\left(\frac{\partial\overline{U_i}}{\partial x_j} + \frac{\partial\overline{U_j}}{\partial x_i}\right) \tag{1.15}$$

where ν_e is the eddy viscosity. The first and the simplest subgrid model is the Smagorinsky one [5]:

$$\nu_e = \left(C_S\Delta\right)^2\overline{\mathbf{S}} \tag{1.16}$$

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

$$\overline{\mathbf{S}} = \sqrt{2\overline{S}_{ij}\overline{S}_{ij}} \tag{1.17}$$

The model is 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, one of the most notable is the dynamic model proposed by Germano et al. [6], which is capable of predict with accuracy region of laminar, transitional and fully developed turbulent flow, but it is not easy to apply to complex tridimensional geometries. For the present work, the Vreman model [7] has been employed:

$$\nu_e = c\sqrt{\frac{B_\beta}{\alpha_{ij}\alpha_{ij}}} \tag{1.18}$$

with:

$$\alpha_{ij} = \frac{\partial \overline{U_j}}{\partial x_i} \tag{1.19}$$

$$\beta_{ij}\Delta_m^2 \alpha_{mi}\alpha_{mj} \tag{1.20}$$

$$B_\beta = \beta_{11}\beta_{22} - \beta_{12}^2 + \beta_{11}\beta_{33} - \beta_{13}^2 + \beta_{22}\beta_{33} - \beta_{23}^2 \tag{1.21}$$

$$c \approx 2.5 C_S \tag{1.22}$$

thi model is simpler to implement than the dynamic one, and it is equally capable of predicting the various characteristic of different flow regimes, furthermore ν_e is always positive, which means that local backscatter is not predicted, but there are not stability issues caused by negative turbulent dissipation.
The filtered system of equations is thus closed by applying the Boussinesq hypotesis to the anisotropic residual stress tensor, whose contribution to the momentum equation is:

$$\frac{\partial \tau_{ij}^s}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\nu_e\left(\frac{\partial \overline{U_i}}{\partial x_j} + \frac{\partial \overline{U_j}}{\partial x_i}\right)\right] \tag{1.23}$$

using Vreman model, eddy viscosity is a function of filtered velocities, hence,

$$\frac{\partial \tau_{ij}^s}{\partial x_j} = \frac{\partial \nu_e}{\partial x_j}\frac{\partial \overline{U_i}}{\partial x_j} + \frac{\partial \nu_e}{\partial x_j}\frac{\partial \overline{U_j}}{\partial x_i} + \frac{\partial^2 \overline{U_j}}{\partial x_i \partial x_i} \tag{1.24}$$

the turbulent stress term is decomposed into an additional viscous term and two higly non-linear terms related to the gradient of the eddy viscosity, whose contribution is, as it will be shown in chapter 2, usually very small and they are therefore negligible.

1.3 Near Wall Modelling

Despite the growth in computer performances, wall resolved Large Eddy Simulation still remains unfeasible for flows of industrial interest. It has been estimated by Choi et al. [8] that the number of grid points required by a wall resolved LES is of the order of $Re_L^{\frac{13}{7}}$, whereas it is approximately equal to Re_L for wall modelled LES.
The simplest approach to wall modelling is the use of the so-called wall functions, by which is possible to impose a correction on the predicted velocity 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, defined as:

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} \tag{1.25}$$

where τ_w is the wall shear stress:

$$\tau_w = \mu \left(\frac{\partial U}{\partial y} \right)_w \tag{1.26}$$

where U is the tangential velocity and y is the normal direction. Using the friction velocity it is possible to define two adimensional variables:

$$y^+ = \frac{y u_\tau}{\nu}, \; U^+ = \frac{U}{u_\tau} \tag{1.27}$$

by which an universal velocity profile can be described. The turbulent boundary layer can be divided into four subregion:

- viscous sublayer,
- buffer layer,
- logarithmic layer,
- wake region,

in the outer layer, which is composed by part of the logarithmic layer and the wake region, the flow is strongly dependent on the geometry and on the Reynolds number, whereas in the inner layer, formed by the other regions, it is possible to define a function f so that $U^+ = f\left(y^+\right)$, as it is represented in figure 1.2.

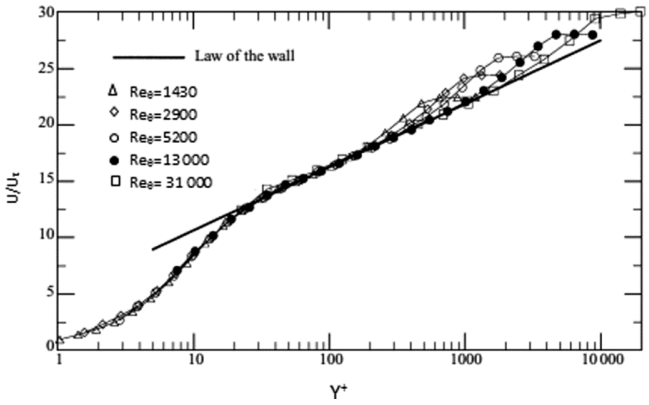


Figure 1.2: Turbulent boundary layer profiles for different Reynolds numbers

Traditionally, the universal profile has been described using two distinct analytical functions:

$$u^+ = f\left(y^+\right) = \begin{cases} y^+, & \text{if } y^+ < 5 \\ \dfrac{1}{k} \log y^+ + B, & \text{if } y^+ > 30, \; \dfrac{y}{\delta} < 0.3 \end{cases} \tag{1.28}$$

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 literature many different efforts to describe the profile with a unified law, such as the one by Spalding [9].

For the present work the Reichardt's wall law [10] is considered:

$$u^+ = f_w(y^+) = \frac{1}{k}\log\left(1+ky^+\right) + 7.8\left(1 - e^{-\frac{y^+}{11}} - \frac{y^+}{11}e^{-0.33y^+}\right) \tag{1.29}$$

As it can be noticed in figure 1.3, the advantage of using Reichardt's wall law is the possibility of capturing with a reasonable level of accuracy the three different regions of the inner layer using only one analytical expression.

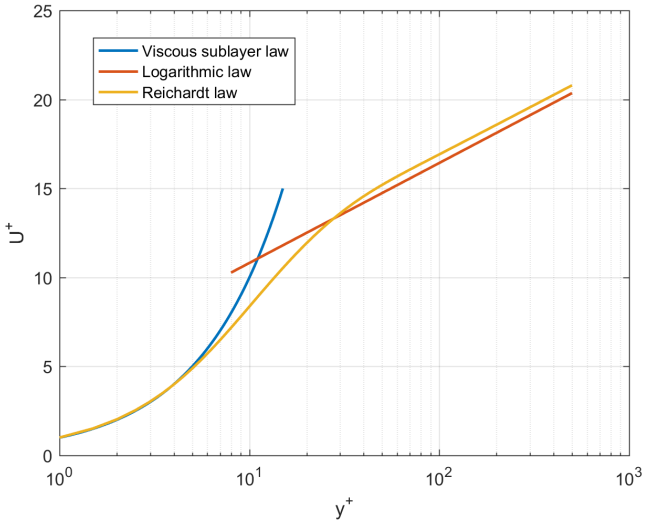


Figure 1.3: Reichardt's wall law

The numerical implementation follows the guideline of De Tullio [11]: in a first loop, all the interface cells in the fluid are identified, in these cells the velocity will be imposed using 1.29, 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 1.4. The position of point 2 is:

$$\mathbf{x_2} = \mathbf{x_1} + \left[\Delta - \varphi(\mathbf{x_1})\right]\frac{\nabla\varphi(\mathbf{x_1})}{\|\nabla\varphi(\mathbf{x_1})\|} \tag{1.30}$$

Using U_2, it is possible to obtain a first approximation of the wall shear stress:

$$\tau_w \approx \mu\frac{U_2}{\Delta} \tag{1.31}$$

therefore it is possible to evaluate the friction velocity:

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} \approx \sqrt{\frac{\nu U_2}{\Delta}} \tag{1.32}$$

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

$$y_2^+ = \frac{\Delta u_\tau}{\nu} \Rightarrow u_2^+ = f_w(y_2^+) \Rightarrow u_\tau = \frac{U_2}{u_2^+} \tag{1.33}$$

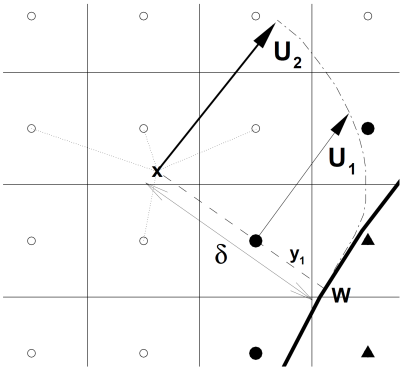


Figure 1.4: Implementation of the wall law

the 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:

$$y_1^+ = \frac{\varphi(y_1)}{\nu} \tag{1.34}$$

where φ is the level set function. The adimensional velocity is evaluated using the wall law:

$$u_1^+ = f_w(y_1^+) \tag{1.35}$$

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

$$U_1 = u_\tau u_1^+ \tag{1.36}$$

The choice of the distance Δ results critical for the well behaviour of the wall correction, if the distance is too large, the point where U_2 is evaluated may be in the outer layer and the law of the wall does not allow to predict the flow with accuracy, especially in the wake region. On the other hand, if Δ is too small, U_2 is calculated too close to the interface. All the results that will be presented are obtained imposing Δ equal to twice the biggest value of the level set function in all the interface points.
Since the point where U_2 is evaluated is not necessarily the center of a cell, velocity values need to be interpolated in order to obtain U_2. The interpolation is made by looking for neighbour cells through the closest node of the cell which owns the point 2 and then using Radial Basis Funtions.

1.4 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. 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:

- 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).

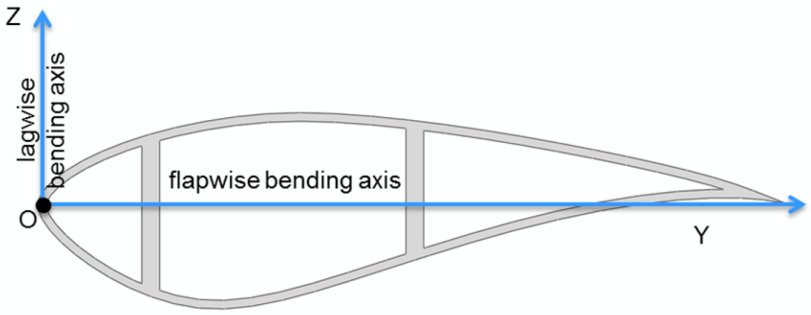


Figure 1.5: Blade cross section and global reference system

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

$$\begin{cases} u(x,y,z) = u_t(x) + (z - z_t)\theta_y(x) - (y - y_t)\theta_z(x) \\ v(x,y,z) = v_c(x) - (z - z_c)\theta_x(x) \\ w(x,y,z) = w_c(x) + (y - y_c)\theta_x(x) \end{cases} \tag{1.37}$$

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 centered in T and parallel to the global one, the deformation field is:

$$\begin{cases} \varepsilon_{xx} = u_{t,x} + (z' - z'_t)\theta_{y,x} - (y' - y'_t)\theta_{z,x} \\ \varepsilon_{xy} = -(z' - z'_c)\theta_{x,x} \\ \varepsilon_{xz} = (y' - y'_c)\theta_{x,x} \end{cases} \tag{1.38}$$

Since modern wind turbine blades are made of lightweight carbon fiber reinforced polymers, the stress-strain relation is that of an anisotropic materials:

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{xy} \\ \sigma_{xz} \end{Bmatrix} = \begin{pmatrix} E_{11} & E_{12} & E_{13} \\ E_{12} & E_{22} & E_{23} \\ E_{13} & E_{23} & E_{33} \end{pmatrix} \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{xy} \\ \varepsilon_{xz} \end{Bmatrix} \tag{1.39}$$

The internal reactions are defined as:

$$\begin{cases} N = \int_A \sigma_{xx} dA \\ M_x = \int_A \left[(y' - y'_c)\sigma_{xz} - (z' - z'_c)\sigma_{xy} \right] dA \\ M_y = \int_A z' \sigma_{xx} dA \\ M_z = -\int_A y' \sigma_{xx} dA \end{cases} \tag{1.40}$$

Substituting 1.39 and 1.38 in 1.40 and integrating on the cross sectional area, the following system of equations is obtained:

$$\begin{Bmatrix} N \\ M_x \\ M_y \\ M_z \end{Bmatrix} = \begin{pmatrix} EA & AT & 0 & 0 \\ AT & GJ & FT & -LT \\ 0 & FT & EI_{y'y'} & -EI_{y'z'} \\ 0 & -LT & -EI_{y'z'} & EI_{z'z'} \end{pmatrix} \begin{Bmatrix} u_{t,x} \\ \theta_{x,x} \\ \theta_{y,x} \\ \theta_{z,x} \end{Bmatrix} \tag{1.41}$$

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 fibers.

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

$$W_{int} = W_{ext} \tag{1.42}$$

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

$$W_{int} = \int_0^L \left(u_{t,x} N + \theta_{x,x} M_x + \theta_{y,x} M_y + \theta_{z,x} M_z \right) dx \tag{1.43}$$

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

$$\{u\} = [N] \{s\} \tag{1.44}$$

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

$$\{u\} = \begin{Bmatrix} u_t & v_c & w_c & \theta_x & \theta_y & \theta_z \end{Bmatrix}^T \tag{1.45}$$

$\big[N\big]$ is the shape function matrix:

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

with L_i and H_i respectively the Lagrange polynomials of the first order and the Hermite polynomials of the third order. $\big\{s\big\}$ is the vector of nodal displacements:

$$\big\{s\big\} = \big\{\big\{s_1\big\}\,\big\{s_2\big\}\big\}$$

$$\big\{s_i\big\} = \big\{u_{t,i} \quad v_{c,i} \quad w_{c,i} \quad \theta_{x,i} \quad \theta_{y,i} \quad \theta_{z,i}\big\}$$

Using the given definitions, equation 1.44 can be expressed as a function of the nodal displacements:

$$W_{int} = \big\{s\big\}^T \big[K_{el}\big]\,\big\{s\big\} \tag{1.46}$$

with $\big[K_{el}\big]$ 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:

$$W_{ext} = W_{in} + W_{lds} = -\big\{s\big\}^T \big[M_{el}\big]\,\big\{\ddot{s}\big\} + \big\{s\big\}^T \big\{F_{el}\big\} \tag{1.47}$$

where $\big[M_{el}\big]$ is the mass matrix of the element and $\big\{F_{el}\big\}$ is the vector of nodal loads. The equation 1.42 can be simplified using expressions 1.46 and 1.47, obtaining the final spatial discretization of the elasticity problem for a single element of the structure:

$$\big[M_{el}\big]\,\big\{\ddot{s}\big\} + \big[K_{el}\big]\,\big\{s\big\} = \big\{F_{el}\big\} \tag{1.48}$$

For an accurate description of the procedure to obtain the mass and stiffness matrices and their elements, see [12]. The linear system 1.48 can be extended to the whole structure by properly assembling the two matrices and the loads vector, hence:

$$\big[M\big]\,\big\{\ddot{d}\big\} + \big[K\big]\,\big\{d\big\} = \big\{F\big\} \tag{1.49}$$

where $\big\{d\big\}$ is the vector of nodal displacements for the entire structure. Equation 1.49 is discretized in the time domain using the Newmark method [13]:

$$\begin{cases} \dot{d}\big(t+\Delta t\big) = \ddot{d}(t) + \Delta t\big[(1-\gamma)\ddot{d}(t) + \gamma\ddot{d}\big(t+\Delta t\big)\big] \\ d\big(t+\Delta t\big) = d(t) + \Delta t\dot{d}(t) + \dfrac{\Delta t^2}{2}\big[(1-2\beta)\ddot{d}(t) + 2\beta\ddot{d}\big(t+\Delta t\big)\big] \end{cases} \tag{1.50}$$

with $0 \leq \beta \leq \frac{1}{2}$ and $0 < \gamma \leq 1$. If $\gamma = 0.5$ the method is of the second order 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 1.49 into 1.50, the fully discretized system of equation for the elasticity problem is obtained:

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

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

1.5 Fluid-Structure Coupling

It is possible to distinguish two distinct methodologies to deal with numerical simulation of the fluid structure interaction [14]: the monolithic approach and the partitioned approach. The first 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. The latter separates the elasticity problem from the Navier-Stokes equation and solves two distinct systems of equations, allowing to split the two domains and threat them with two different codes. While the monolithic approach can achieve better accuracy and it is very well suited for immersed boundary methods, the partitioned one is more flexible and easier to implement, since requires little modifications to both the structure and the fluid codes. Furthermore, splitting the domains allows to use different meshes for the two, not only in term of number of computational cells, but also in term of topology, such as coupling a 3D finite volume method with a 1D finite element model.

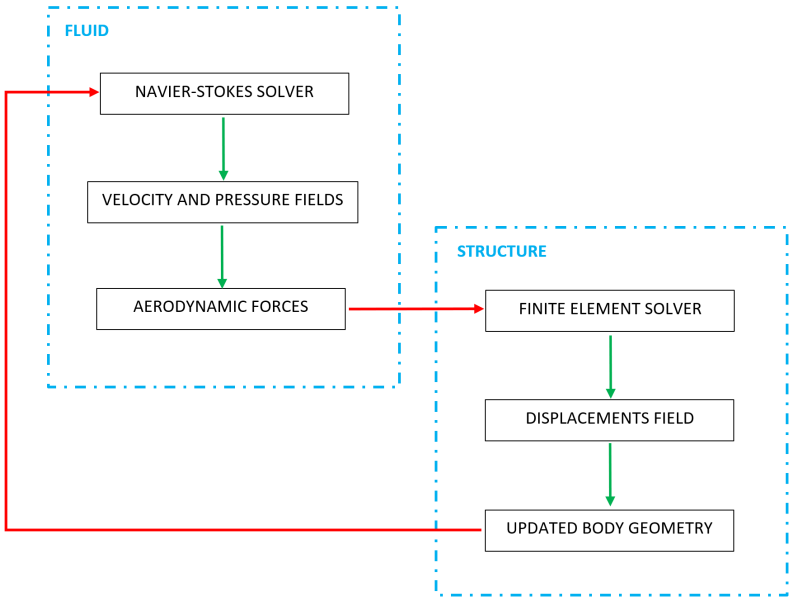


Figure 1.6: Partitioned scheme

In order to exploit the simplicity of the beam structure model, a partitioned scheme has been employed, which means that, as represented in figure 1.6, during each time iteration there is an exchange of information between the Navier-Stokes algorithm and the FEM code, which needs to know the aerodynamics forces to apply to the blade to evaluate the displacements, that are communicated back to the fluid solver to update the geometry and the body velocity.

Chapter 2

Validation of the model

Before performing simulations on complex geometries such as a wind turbine blade, the model has to be validated using well documented reference test cases, which allow to compare the results in term of aerodynamic coefficients and flow characteristics. Concerning wall modelled large eddy simulation (WMLES), one of the most popular benchmarks is the flow past a circular cylinder, thanks to the high number of experimental results in literature (see [15], [17] and [18]) and to the characteristic of the flow, highly unsteady with the development of a turbulent wake due to the separation of the boundary layer.
A first validation run was performed at a Reynolds number of 3900 in order to assess the performance of the mathematical model for a low Reynolds flow. Average wake velocity profiles and aerodynamic coefficients are compared with experimental data and results from other well documented CFD simulations.

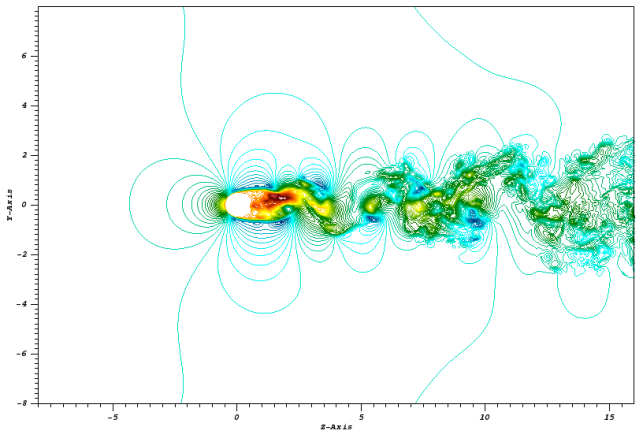


Figure 2.1: Isolines of streamwise velocity at x=2, 100 levels, $Re = 3900$

In figure 2.2, mean wake velocity profiles for three different values of the streamwise coordinate are presented, dashed lines values are obtained neglecting the contribution of the gradient of the eddy viscosity to the residual stress tensor, whereas the continuous lines are obtained using the full LES model. The dots represent the experimental data used as a reference. As anticipated in chapter

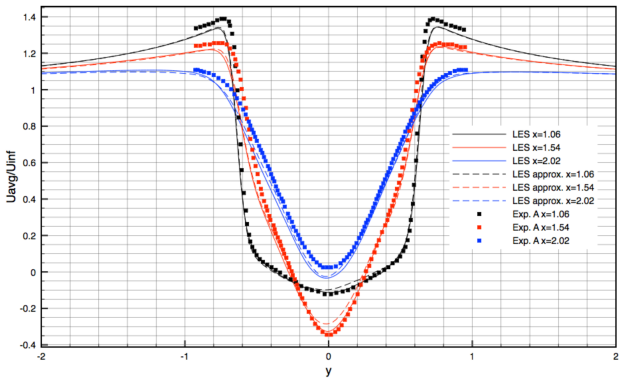


Figure 2.2: Comparison of average wake velocity profiles

1 the gradient of the eddy viscosity has a small impact on the solution and the profiles obtained with both of the LES models are well correlated with the experimental values of Parnaudeau et al. [19]. The streamwise velocity isolines are illustrated in figure 2.1, it is clearly visible that the separation is laminar and the transition occurs in the wake, with the formation of small turbulent structures, which means that the topology of the flow at low Reynolds numbers is correctly predicted by the model.
In this chapter, the methodology of the validation will be presented, and the results obtained for a high Reynolds number flow will be discussed.

2.1 Discretization of the Navier-Stokes equation

The validation of the model the implementation was made on a stable in-house developed cartesian finite difference code, called NaSCAR3D, in order to use a well estabilished numerical discretization of the penalized Navier-Stokes equation and concentrate the testing efforts only on the WMLES model. The penalized incompressible Navier-Stokes 1.2 are solved using the classical Chorin algorithm [20], also known as predictor-corrector scheme, which consists in three steps: prediction, projection and correction. In the prediction step a virtual velocity field is evaluated by solving an unsteady advection-diffusion equation:

$$\frac{u_i^* - u_i^n}{\Delta t} = -\frac{1}{2}\left(3C^n - C^{n-1}\right) - \frac{1}{\rho}\frac{\delta p^n}{\delta x_i} + \frac{\nu + \nu_e}{2}\left(D^n + D^*\right) + \frac{\chi_B}{K}\left(u_{iB} - u_i^*\right) \quad (2.1)$$

where:

$$C = \frac{\delta\left(U_j u_i\right)}{\delta x_j} \quad (2.2)$$

$$D = \frac{\delta^2 u_i}{\delta x_j \delta x_j} \quad (2.3)$$

are respectively the convective and diffusive terms. u is the cell-centered velocity, U is the face-centered velocity. The operator $\delta/\delta x_i$ is the central difference

approximation of the first derivative. A second-order Adams-Bashfort scheme has been used for the convective term and a second order Crank-Nicholson integration has been chosen for the viscous term. Since the equation 2.1 does not take into account the incompressibility constraint, the predicted velocity fields are not divergence free. Let p' be the pressure correction, defined as $p^{n+1} - p^n$, the relationship between the virtual velocity and the velocity at the (n+1)-th step is:

$$u_i^{n+1} = u_i^* - \frac{\Delta t}{\rho} \frac{\partial p'}{\partial x_i} \tag{2.4}$$

The virtual velocity is projected into a divergence free space applying the divergence to equation 2.4 and imposing the incompressibility condition to u^{n+1}:

$$\frac{\partial u_i^*}{\partial x_i} = \frac{\Delta t}{\rho} \frac{\partial^2 p'}{\partial x_i \partial x_i} \tag{2.5}$$

which is a Poisson equation for the pressure correction. The discretization of the equation is straightforward:

$$\frac{\delta^2 p'}{\delta x_i \delta x_i} = \frac{\rho}{\Delta t} \frac{\delta u_i^*}{\delta x_i} \tag{2.6}$$

Using cell-centered velocity to compute the divergence may lead to pressure oscillation, hence the equation 2.6 is solved using a face-centered discretization. The face-centered velocity is evaluated using the following scheme:

$$u_i^{**} = u_i^* + \Delta t \left(\frac{\delta p^n}{\delta x_i} \right)_{cc} \tag{2.7}$$

$$U_i^{**} = F\left(u_i^{**}\right) \tag{2.8}$$

$$U_i = U_i^{**} - \Delta t \left(\frac{\delta p^n}{\delta x_i} \right)_{fc} \tag{2.9}$$

with F being an interpolation function. The equation 2.6 becomes:

$$\left(\frac{\delta^2 p'}{\delta x_i \delta x_i} \right)_{fc} = \frac{\rho}{\Delta t} \left(\frac{\delta U_i}{\delta x_i} \right)_{fc} \tag{2.10}$$

The equation 2.10 is solved using the Generalized Minimum Residual algorithm [21] and homogeneous Neumann boundary conditions for p'. The last step consists in updating the pressure and velocity fields:

$$\begin{cases} u_i^{n+1} = u_i^* - \dfrac{\Delta t}{\rho} \left(\dfrac{\delta p'}{\delta x_i} \right)_{cc} \\ U_i^{n+1} = U_i - \dfrac{\Delta t}{\rho} \left(\dfrac{\delta p'}{\delta x_i} \right)_{fc} \\ p^{n+1} = p^n + p' \end{cases} \tag{2.11}$$

The Vreman subgrid model has been directly introduced in the prediction step, whereas the wall function can be seen as a second correction step, applied directly to u^{n+1}, in order to avoid strong perturbations of the divergence of the virtual velocity before the projection step.

2.2 Validation run set-up

Thanks to the use of the MPI standard, the NaSCAR3D code is able to run simulation on massively parallelized supercomputers, its scalability has been proven up to thousands of cores. For the high Reynolds number simulation, two different machines have been used:

- Occigen: each node is composed by two 12-core Intel Haswell CPUs with a clock speed of 2.6GHz, with 64 or 128 GB of shared RAM
- Turing: IBM Blue Gene/Q, each node is made by 16 POWERPC A2 CPUs with a clock frequency of 1.6GHz, with 16 GB of shared RAM

The axis of the cylinder corresponds to the x-axis of the domain, the streamwise direction is z, the diameter of the cylinder is equal to 1. The levelSet function for the cylinder is evaluated in an analytical way:

$$\varphi = \sqrt{x^2 + y^2} - R \tag{2.12}$$

The size of the computational domain is $[0, 4] \times [-8, 8] \times [-8, 16]$. The boundary conditions are:

- inlet: Dirichlet condition for the velocity
- Outlet: homogeneous Neumann condition for the velocity
- x-boundaries: periodic conditions
- y-boundaries: homogeneous Neumann condition for the velocity

The velocity of the undisturbed flow is: $U_\infty = 25$, the fluid density is $\rho = 1$ and the dynamic viscosity is: $\mu = 1.7857 \cdot 10^{-4}$, in order to obtain:

$$Re = \frac{\rho U_\infty D}{\mu} = 1.4 \cdot 10^5$$

The convective time is defined as:

$$t = \frac{L}{U_\infty} = 0.04$$

and it represents the time needed to a fluid particle travelling at U_∞ to cover a distance equal to one diameter. The Smagorinsky constant C_S has been set to 0.4, which is a higher value than the standard one, which is around 0.17, but, during the low Reynolds tests, it has been found that a lower value of C_S was not enough to dissipate the turbulent energy after the formation of the first eddies, making simulations diverge. 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 a sort of laminar wall function. The initial velocity field is the fully developed flow for $Re = 3900$, in order to save the time necessary to trigger the vortex shedding phenomenon, which is very slow to develop.

2.3 First run results

A first run has been performed on a $100 \times 800 \times 1200$ computational grid, with a complessive number of 96 millions of cells, using the 32 nodes of the Occigen machine for a total number of 768 cores. In this configuration, a simulation of 60 convective times requires about 20 wall clock hours.

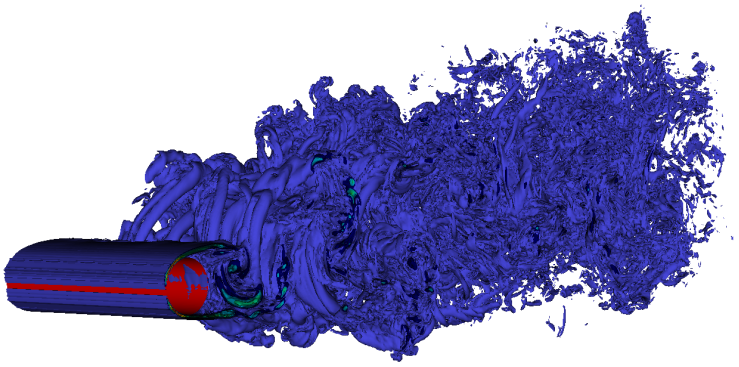


Figure 2.3: Contour surfaces of the norm of the vorticity, 10 levels

The contour lines of the norm of the vorticity are represented in figure 2.3, it is possible to observe the strong vorticity in the boundary layer, moreover, the separation is still laminar and the transition occurs in the wake.

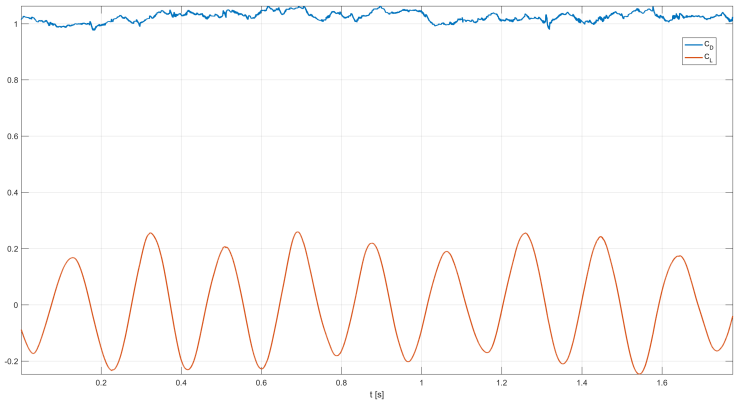


Figure 2.4: Time evolution of the aerodynamic coefficients

In figure 2.4 is reported the time evolution of both the lift and the drag coefficients, the C_L oscillates quite smoothly around the zero line, whereas the drag coefficient presents irregular fluctuations with a mean value of 1.02, which

is slightly lower than the experimental and numerical results found in literature.

Average wake velocity profiles were evaluated in two different position, $z = 1$ and $z = 3$, the averaging operation is made both along the x-axis and in time. The comparison is made using the experimental results of Cantwell et al. [18]. The profiles are presented in figure 2.5 and 2.6, for $z = 1$ the streamwise and the normal velocities are plotted, on the other hand, in the second position, only the streamwise velocity is used for the comparison.

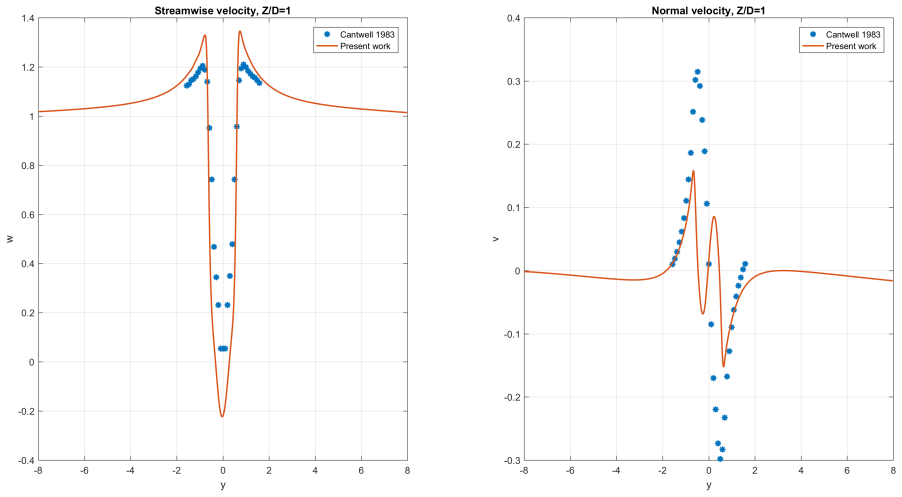


Figure 2.5: Normalized velocity profiles, $z = 1$

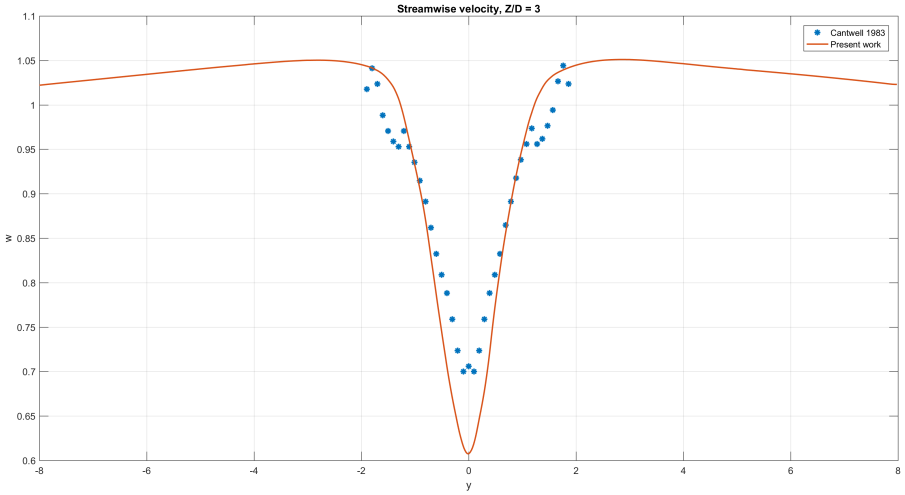


Figure 2.6: Normalized velocity profile, $z = 3$

The profiles are obtained by averaging the field over a span of 50 convective times (a first simulation of 60 convective times was performed to remove the influence of the initial transient). The streamwise velocity seems to fit pretty

well the experimental data, even though the velocity in the centerline is too small; on the other hand the normal velocity is totally different. Since the centreline velocity is negative at $z = 1$, the discrepancy in the profile could be explained as overestimation of the recirculation bubble in the mean flow downstream the cylinder. In an effort to better understand the behaviour of the mean flow in the wake, the average centreline velocity has been plotted, as reported in figure 2.7.

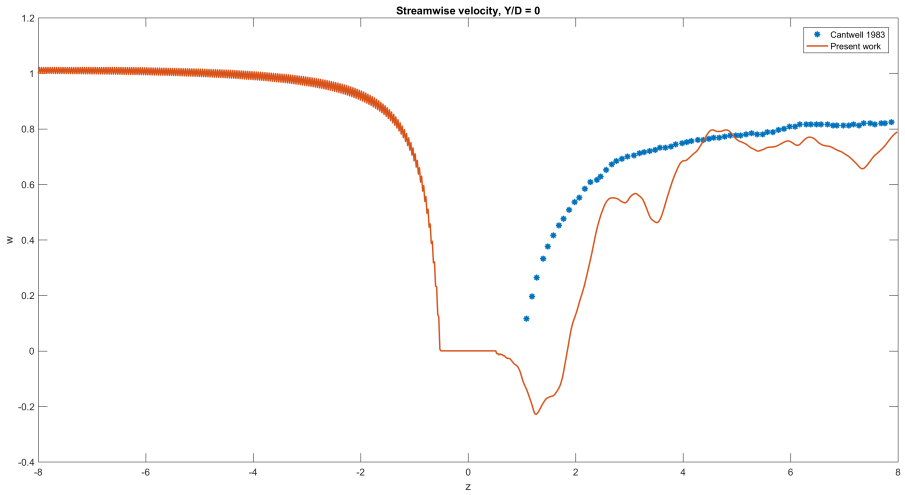


Figure 2.7: Centreline velocity profile

The recirculation region seems to be too large, furthermore upstream the cylinder it is possible to notice some spurious fluctuations with an amplitude of 2% of the inlet velocity.

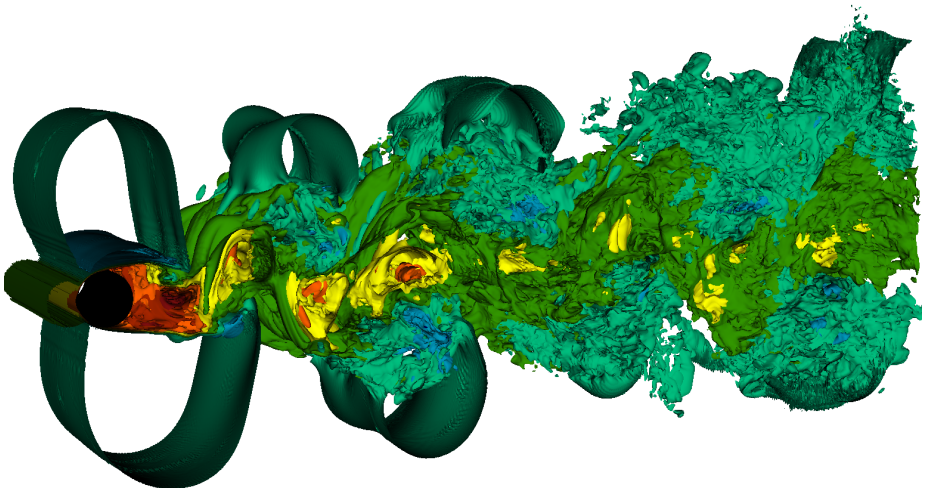


Figure 2.8: Contour surfaces of the streamwise velocity, 10 levels

Those fluctuations cannot be observed in figure 2.8 with 10 contour levels, but

they are clearly visible in figure 2.9, using 100 isolines.

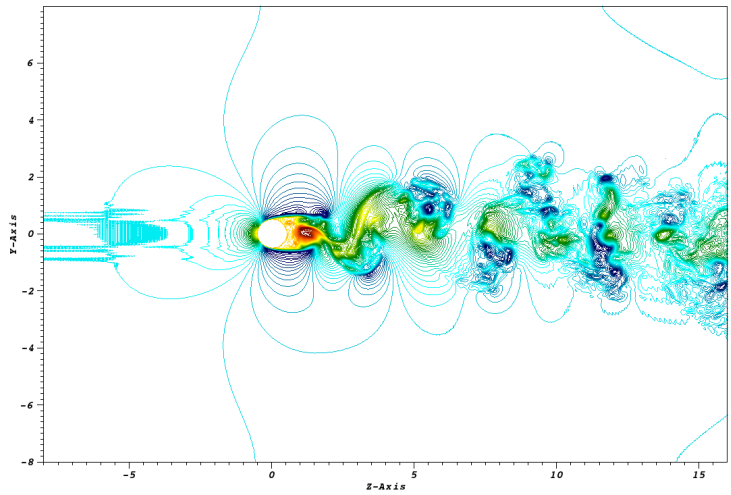


Figure 2.9: streamwise velocity contour lines for $x = 2$, 100 levels

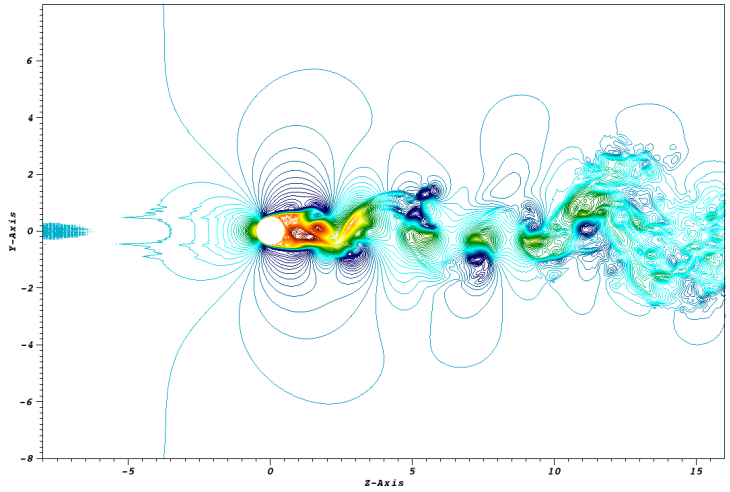


Figure 2.10: streamwise velocity contour lines for $x = 2$ and $C_S = 1$, 100 levels

In order to figure out the cause of those oscillation, several small runs were performed, in which it was tested the response of the code to the variation of some simulations parameters, such as:

- CFL coefficient, which limits the Δt;
- order of the time discretization;
- absolute tolerance of the GMRES solver for the Poisson equation;
- boundary conditions;

- Smagorinksy constant.

It was found that the only parameter that had an influence on the fluctuations is the Smagorinsky constant, indeed, increasing the value to 1, the oscillations are smoothed, as it can be notice in figure 2.10 and 2.11, but, on the other hand, a higher eddy viscosity leads to the dissipation of the small turbulent structures in the wake.

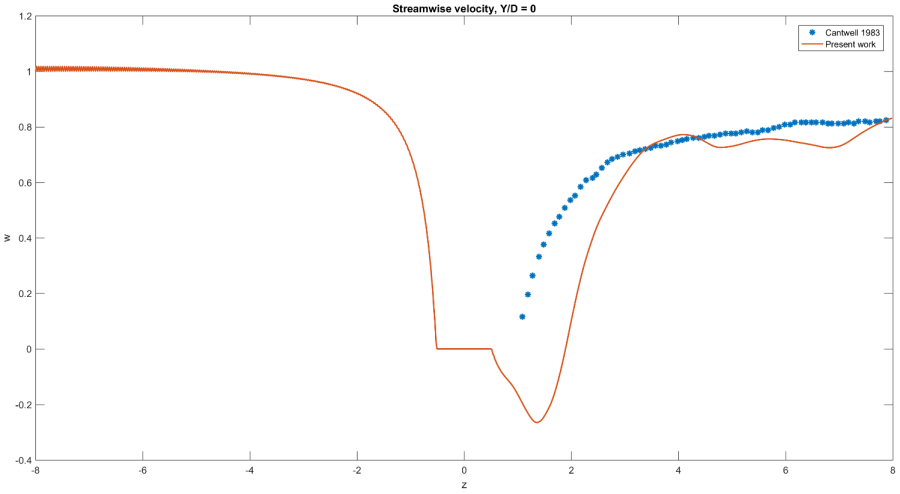


Figure 2.11: Centreline velocity profile, $C_S = 1$

It has been proven that the cause of the spurious oscillations is the lack of turbulent dissipation, which can be compensated by using a higher Smagorinsky constant. It is as well known that the action of dissipation is dominant at the smallest scales of the flow, therefore a lack of it might be triggered by an underresolved simulation, hence a more refined grid might be needed in order to obtain better results.

2.4 Second run results

A second run was made on the Turing machine, using a refined mesh with double the number of point in each direction, which means that the total number of cells is 8 times the one used for the first run, leading to 768 millions computational points and over 3 billions of unknowns. Due to the smaller clock speed of the Turing machine, a very high number of processors was necessary: 8192 physical cores were employed. The IBM Blue Gene/Q architecture supports the multithreading, the optimal performance of the NaSCAR3D is achieved with 2 threads per core, so that a total amount of 16384 threads was reached. In this configuration 20 wall clock hours were just enough to simulate slightly less than 3 convective times, not only due to the worse performance of the machine, but also due to the CFL condition that impose a smaller time step for each Navier-Stokes iteration when the grid is refined. A preliminary 20 wall clock hours run was made in order to understand if the finer mesh is able to dissipate

the spurious oscillation, the result is reported in figure 2.12 and it proves the hypothesis: there are very less fluctuations upstream the cylinder.

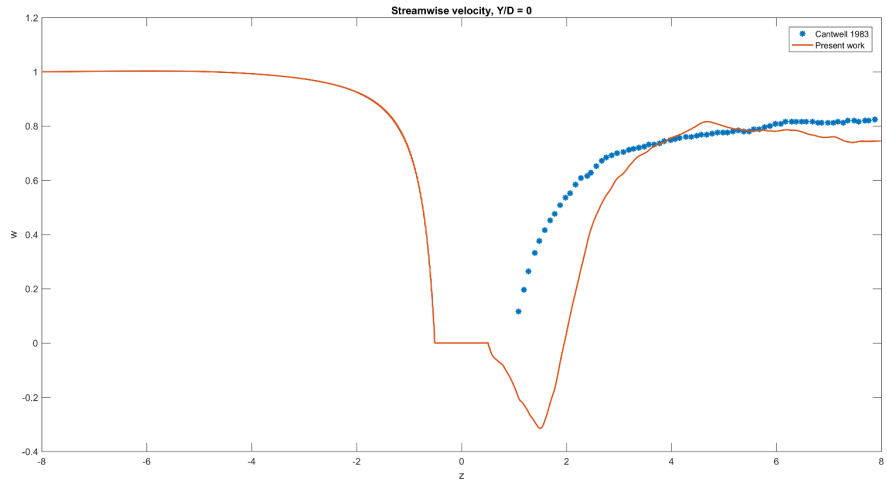


Figure 2.12: Centreline velocity profile, finer mesh, $C_S = 0.4$

An additional period of approximately 42 convective times has been simulated, the first 20 have not been taken into account for the computation of average profiles, in order to let the turbulence develop on the fine grid, therefore the statistics are made on the last 22 convective periods of simulation. The profiles are reported in figures 2.13, 2.14 and 2.15.

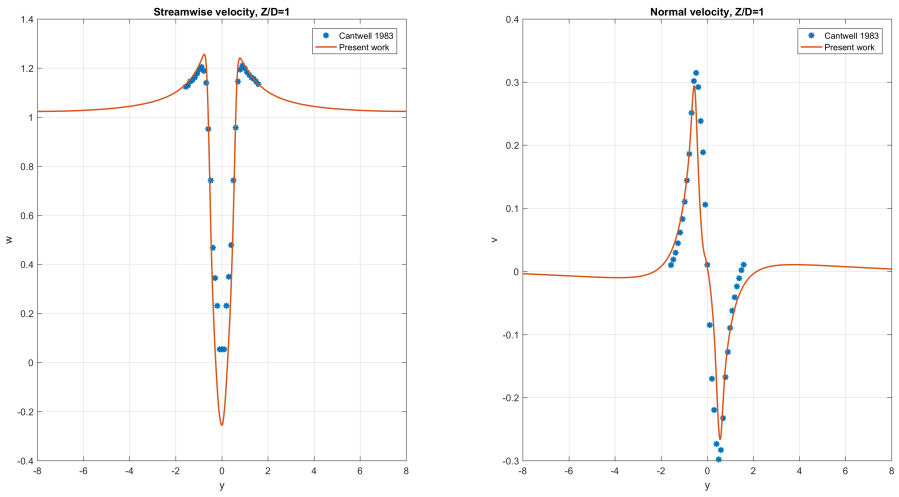


Figure 2.13: Average velocity profile, finer mesh, $z = 1$

In figure 2.13 are represented the mean velocity profiles at $z = 1$, both the streamwise and the normal component of the velocity are well correlated with the experimental measurements, the centreline velocity although is still negative and quite far from the values of Cantwell and Coles [18]. On the other hand

the normal profile is totally different from the one obtained in the first run 2.5, confirming that the first series of simulation was made on a mesh that was too coarse.

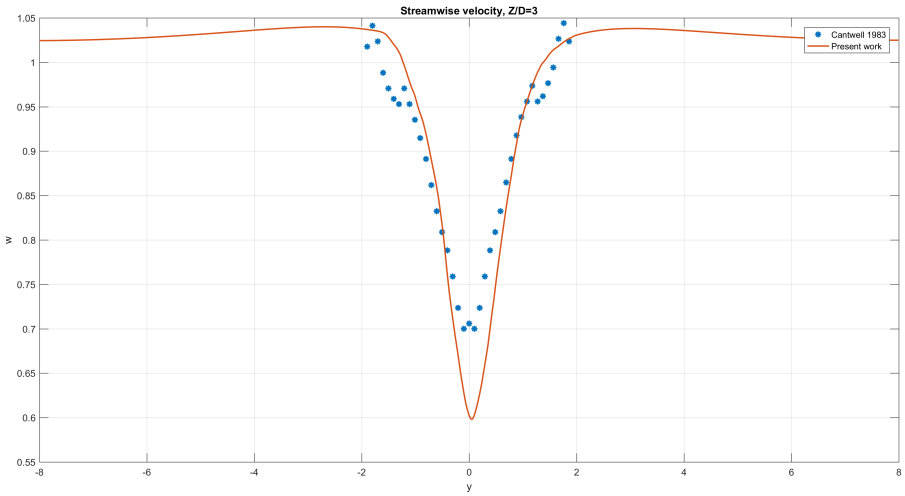


Figure 2.14: Average streamwise velocity profile, finer mesh, $z = 3$

Figure 2.14 shows the profile of the streamwise velocity at $z = 3$, except in central region of the wake, the values are comparable with measurements. In figure 2.15 the centreline velocity is plotted, the oscillations upstream the cylinder are still presents, but their amplitude has decreased to the 0.2% of the undisturbed velocity. Downstream the cylinder, within the wake, the velocity is underestimated when compared to the experimental reference.

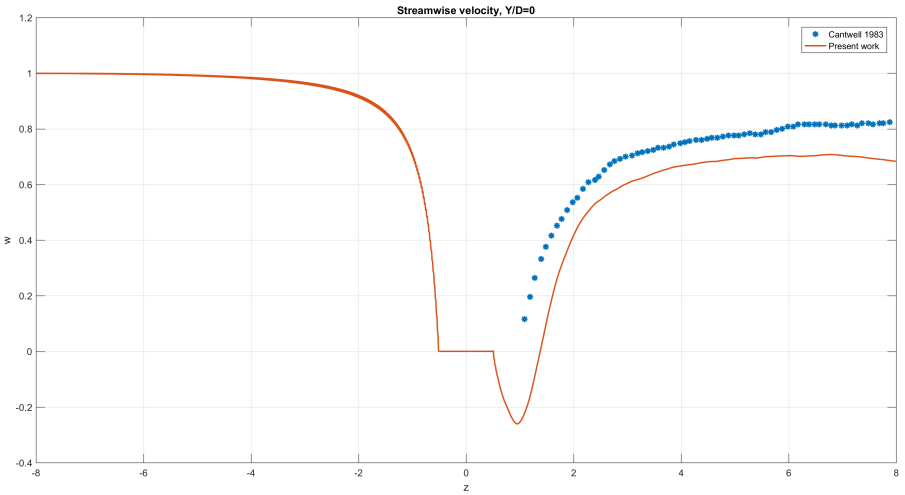


Figure 2.15: Average streamwise velocity profile, finer mesh, $y = 0$

It has to be said that the timespan over which the statistics were evaluated was not long enough to reach full convergence, indeed, considering the Strouhal

number:

$$St = \frac{fD}{U_\infty} \tag{2.13}$$

where f is the frequency of the vortex shedding. For high Reynolds number flows past bluff bodies, the St is approximately equal to 0.2, which means that each vortex shedding cycle has a period roughly equal to 5 convective times. As remarked by Breuer [22], statistics should be evaluated over at least 100 convective times, which correspond to about 20 vortex shedding cycles. A similar runtime would have required an unaffordable number of computational time, given that the present simulation already consumed more than 2.5 million hours (evaluated as the product of wall clock hours and number of physical cores employed). Even though 22 convective times is not sufficient for the full convergence of the average profiles, the results are close to the experimental data, confirming that the mathematical model presented in chapter 1 is able to predict with accuracy unsteady turbulent phenomena.

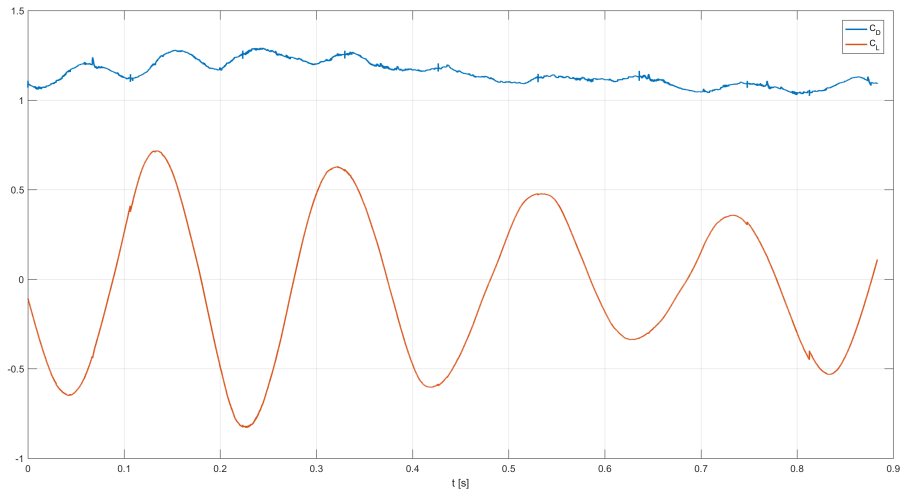


Figure 2.16: Time evolution of the aerodynamic coefficients, finer mesh

The value of the average drag coefficient is in the same range of the results found in literature, as it is shown in table 2.1.

Case	C_D
Present work, WMLES	1.084
Luo et al. [25], PANS	$0.82 - 1.28$
Luo et al. [25], DES	0.84
Kim [23], LES	1.21
Breuer [22], LES	$0.971 - 1.454$
Travin et al. [24], DES	1.08
Cantwell et al. [18], exp.	1.237

Table 2.1: Comparison of drag coefficient values

In figure 2.17, the contour plot of the instantaneous streamwise velocity field at $x = 2$ is represented, using 100 levels; it is possible to appreciate the high number of turbulent structures in the wake, whose size is significantly smaller compared to what can be observed in figure 2.1, for the low Reynolds simulation.

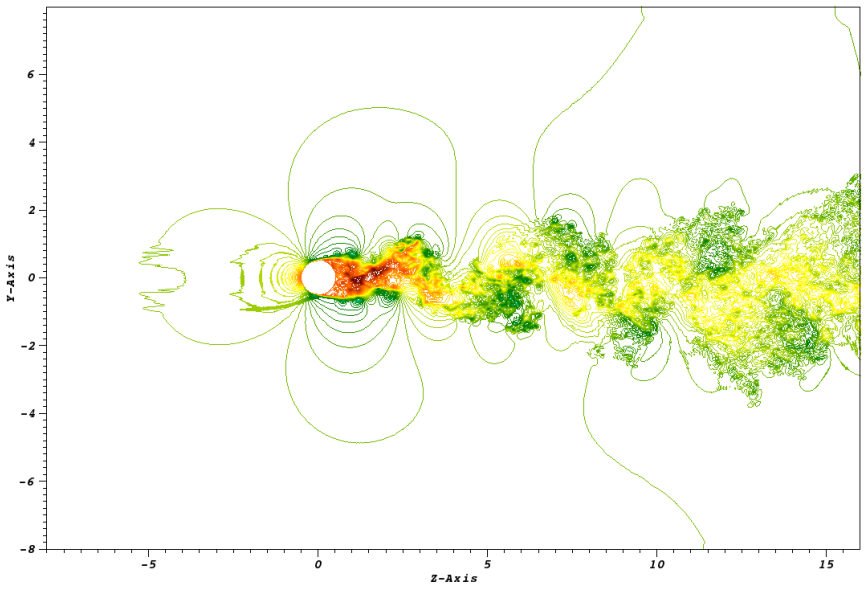


Figure 2.17: Isolines of instantaneous streamwise velocity, 100 levels

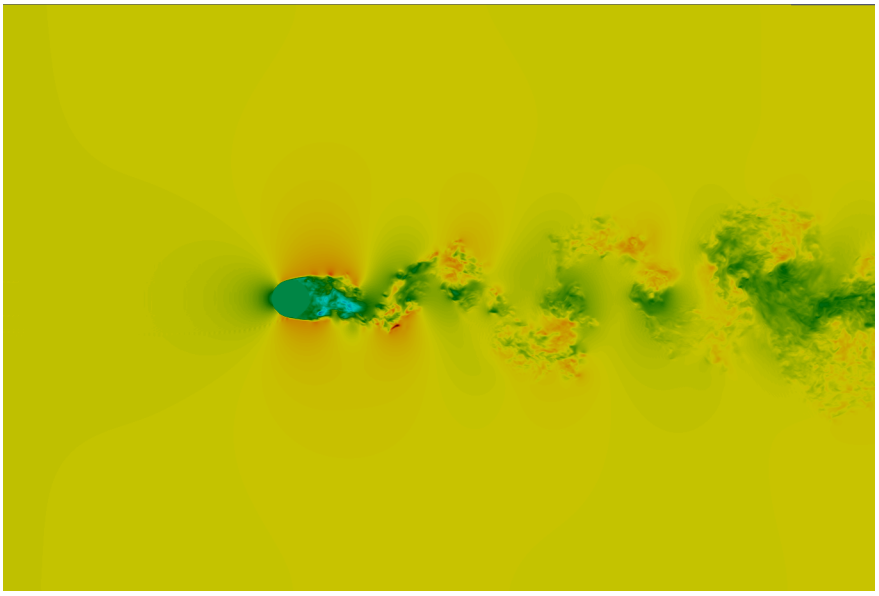


Figure 2.18: Pseudocolor plot of instantaneous streamwise velocity, $x = 2$

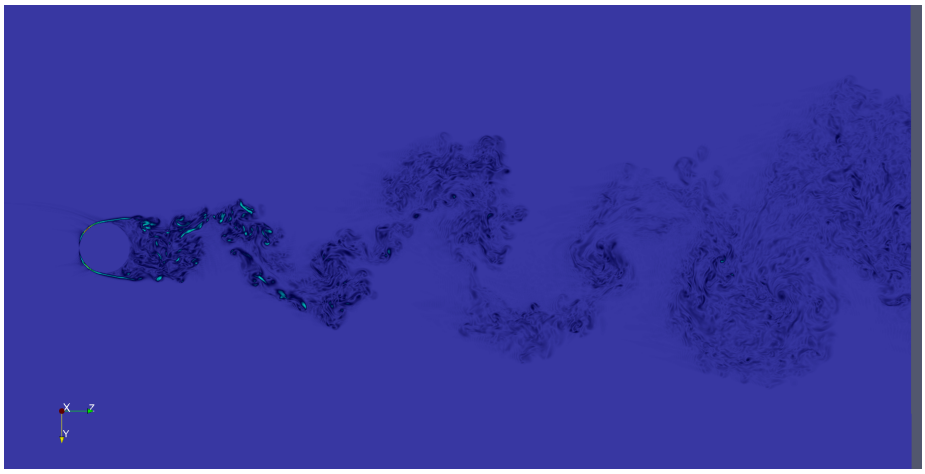


Figure 2.19: Pseudocolor plot of instantaneous norm of vorticity, $x = 2$

In figure 2.19 the plot of the instantaneous vorticity norm is reported, the maximum vorticity is detected in the boundary layer, then laminar separation occurs. The fluid dynamic instability causes the generation of large scale vortices in the near wake, where the transition from laminar to turbulent flow is observed. The stretching phenomenon gives rise to the inertial cascade, where turbulent energy is transferred to small scales and then dissipated, the flow has become fully turbulent.

2.5 Validation of the structure model

The finite element model has been validated using the cantilever beam test case, whose results can be compared with the Euler-Bernoulli beam theory, that allows to obtain simple analytic expressions of displacements and natural frequencies of the structure.

Figure 2.20: Cantilever beam

For the validation, fictitious structural parameters are used:

$$EI_{yy} = EI_{zz} = 10^6 \ \ Nm^2, L = 22.25 \ \ m, EA = 10^9 \ \ N, \rho A = 1 \ \ Kg/m$$

where E is the young modulus, I_{yy} and I_{zz} are the inertia with respect to the y and z axis, L is the length of the beam, A is the area of the cross section and ρ is the density of the material. Three test cases are considered:

- case 1: static test, uniform axial load,
- case 2: static bending test, uniform transverse load,
- case 3: free bending vibrations.

For the first test case, the analytic solution is given by the equation:

$$\frac{d^2u}{dx^2} = -\frac{q}{EA} \tag{2.14}$$

where u is the axial displacement and q is the load. The clamped-free constraints are:

$$\begin{cases} u(0) = 0 \\ \dfrac{du}{dx}(L) = 0 \end{cases}$$

hence:

$$u = \frac{qL^2}{EA}\left(\xi - \frac{\xi^2}{2}\right) \tag{2.15}$$

where $\xi = x/L$. The load q has been chosen in order to obtain:

$$u(L) = \frac{qL^2}{2EA} = 1 \tag{2.16}$$

therefore, $q = 4.04 \cdot 10^6 \ \ N/m$.

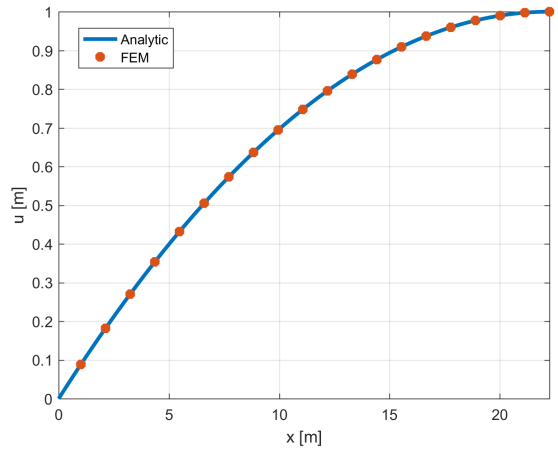


Figure 2.21: Test case 1: comparison of analytic and FEM results

For the second test case, the analytic solution is provided by the equation:

$$\frac{d^4v}{dx^4} = \frac{p}{EI_{zz}} \tag{2.17}$$

where v is the deflection and p is the transverse load. The clamped-free constraints for bending are:

$$\begin{cases} u(0) = 0 \\ \dfrac{du}{dx}(0) = 0 \\ \dfrac{d^2u}{dx^2}(L) = 0 \\ \dfrac{d^3u}{dx^3}(L) = 0 \end{cases}$$

hence:

$$\begin{cases} v = \dfrac{pL^4}{24EI_{zz}}\xi^2\left(\xi^2 - 4\xi + 6\right) \\ \theta_z = \dfrac{dv}{dx} = \dfrac{pL^3}{6EI_{zz}}\xi\left(\xi^2 - 3\xi + 3\right) \end{cases} \tag{2.18}$$

The load q has been chosen in order to obtain:

$$u(L) = \frac{pL^4}{8EI_{zz}} = 1 \tag{2.19}$$

therefore, $p = 32.64\ N/m$. The analytic tip rotation for this test case is:

$$\theta_z(L) = \frac{pL^3}{6EI_{zz}} = 0.0599 \tag{2.20}$$

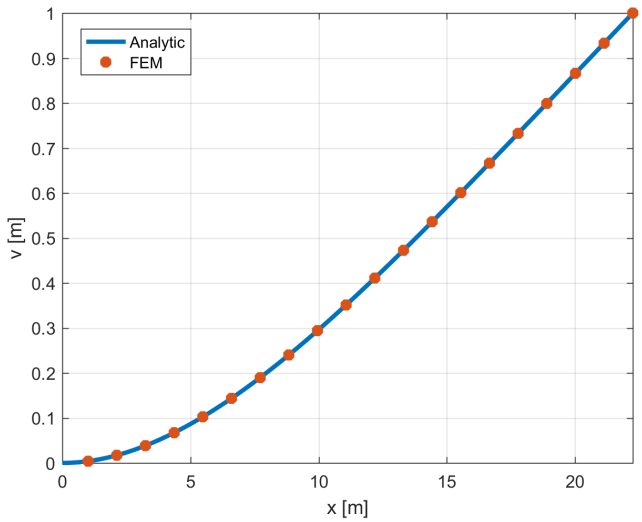


Figure 2.22: Test case 2: comparison of analytic and FEM results, deflection

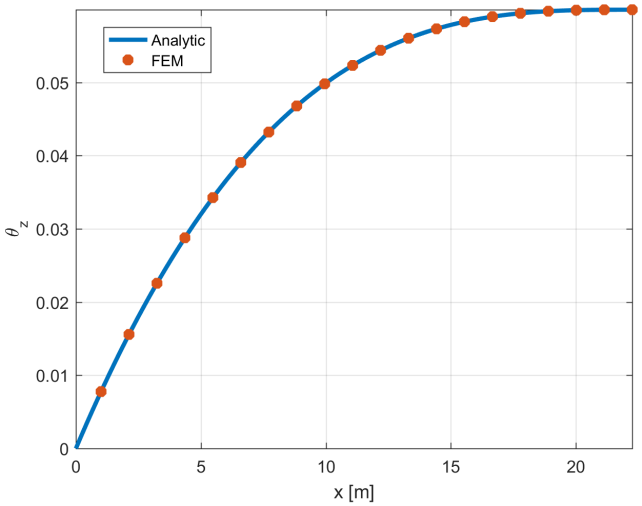


Figure 2.23: Test case 2: comparison of analytic and FEM results, rotation

In figure 2.21, 2.22 and 2.23 are shown the comparisons between the FEM model and the analytical results, the numerical values lie exactly on the curve obtained using the Euler-Bernoulli theory, certifying the quality of the static model. All the tests are performed using 200 elements.
The free bending vibration case aims to evaluate the eigenfrequency of the system, which are the natural frequencies of the structure in absence of external forces. It is a dynamic test, therefore it involves the mass matrix and allows to validate the inertial response of the model. The dynamic equilibrium equation is:

$$\rho A \frac{\partial^2 v}{\partial t^2} + EI_{zz} \frac{\partial^4 v}{\partial x^4} = 0 \tag{2.21}$$

The natural frequencies can be evaluated analytically by means of the following expression:

$$f_i = \frac{(\alpha_i L)^2}{2\pi} \sqrt{\frac{EI_{zz}}{\rho A L^4}} \tag{2.22}$$

where α_i depends on the boundary conditions. From a numerical viewpoint the free vibration is recreated by imposing an initial deformation and then releasing the structure. Then applying the Fourier transform to the tip displacement time history, the spectrum of the response is obtained, the peaks of the spectrum represent the natural frequencies of the beam.

	f_1	f_2	f_3	f_4
Analytic	1.130	7.084	19.835	38.868
FEM	1.130	7.081	19.770	38.395

Table 2.2: Comparison of FEM and Analytic eigenfrequencies (in Hz)

In figure 2.24 the spectrum of the tip displacement is shown, it can be noticed that natural frequencies obtained sampling the response of the FEM model

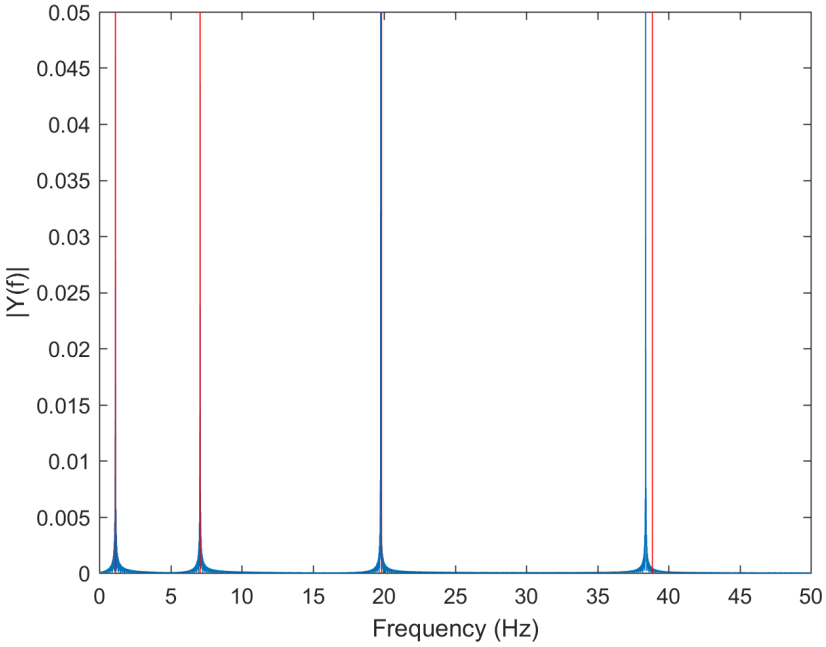


Figure 2.24: Test case 3: single side amplitude spectrum

are well correlated with the analytic values, the comparison of the results is reported in table 2.2. There is a slight loss of accuracy in the values for the fourth structural mode that can be explained with the choice of a too large timestep for the Newmark time integration.

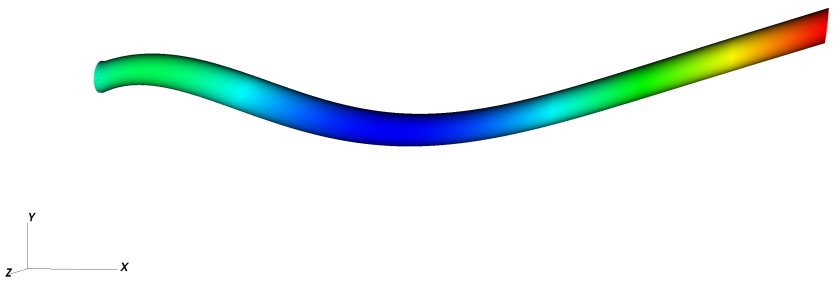


Figure 2.25: Test case 3: propagation of transverse waves in the beam

Chapter 3

An Octree solver for incompressible flows

Using uniform cartesian grids allows easy parallelization and a low memory usage, since there is no need to store information about the connectivity and the geometry of finite volumes, but, on the other hand, the impossibility of making local refinements makes the mesh inefficient, such as the one used for the high Reynolds cylinder simulation. Block structured grids permit to maintain the advantages of the cartesian ones, with the possibility of creating an adaptive mesh by refining where more detail is needed or coarsening where it is not. This approach fits very well with the immersed boundary method, where the mesh is not body fitted, and thus needs to be properly refined in the proximity of the fluid-solid interface.

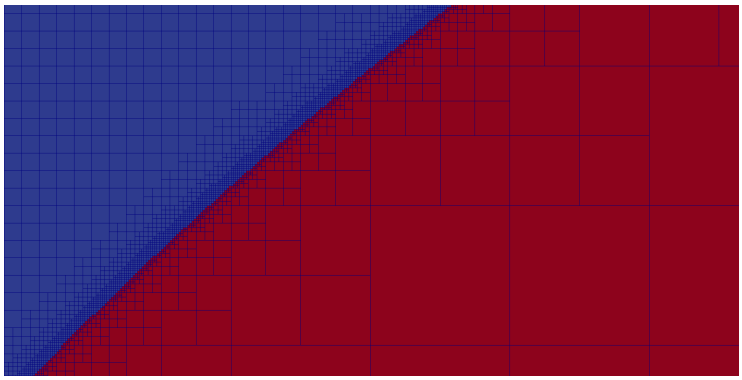


Figure 3.1: Local refinement close to the surface of a cylinder

One of the most popular approach to generate and handle block structured meshes is the use of hierarchical data structures named Octree (Quadtree in 2D) where each element is a cube, or squares, and refinement is made by halving the size of the cell, creating thus a nested grid. Thanks to their simplicity, Octrees are well suited for automatic mesh refinement (AMR) techniques, such as the one used to obtain the grid represented in figure 3.1, where levelset-based refinement and coarsening criteria have been employed.

Local refinements lead to a globally non-uniform grid, which means that a finite difference discretization is not straightforward, whereas a finite volume method is much more suited. In this chapter an innovative algorithm for the solution of the Navier-Stokes equations on Octree grids will be presented and discussed, with a particular attention to the treatment of level jumps.

3.1 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 3.2. The tree is defined as a collection of interconnected cells, also called octants. Each refinement generates 8, 4 in 2D, disjointed subtrees, whose nodes are called **children**. Nodes without children are named **leaves**.

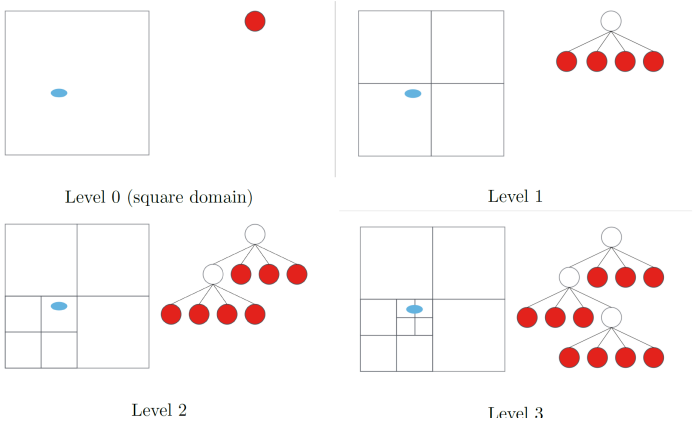


Figure 3.2: Dual nature of the data structure

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 meshes, sorting is achieved using space-filling curves that cross each element of the grid once.

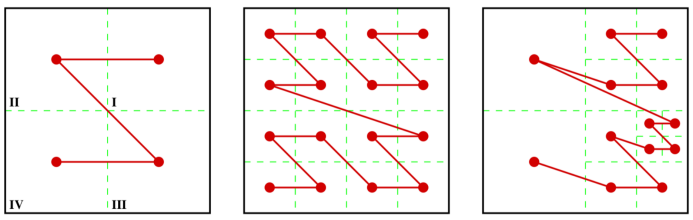


Figure 3.3: Z-order space-filling curve

As remarked by Raeli [26], there are many different examples of space-filling curves, one of these is the Z-Order, introduced by Morton [27], who also pro-

posed 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 Otrees. The Z-order is also used to compute the parallel partition of the domain and the communication between two different subdomains is made thanks to one layer of ghosts cells where the information is shared between two processors.

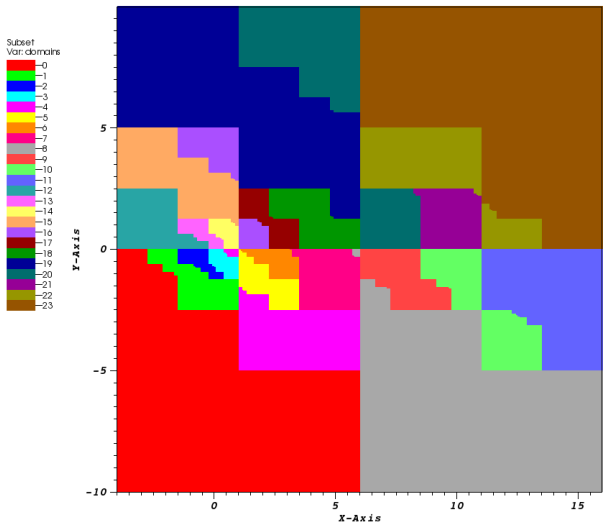


Figure 3.4: Example of Z-order based parallel partition obtained on 24 cores

For the present work, the generation and the handling of the Octree structure was made possible by the PABLO library[1], which deals with Z-ordered, linear Octrees, parallelized using the MPI paradigm.

3.2 Finite volume discretization

The penalized Navier-Stokes equation 1.2 can be rewritten in the Lagrangian form:

$$\begin{cases} \nabla \cdot \mathbf{V} = 0 \\ \dfrac{D\mathbf{V}}{Dt} = -\nabla p + \dfrac{1}{Re}\nabla^2 \mathbf{V} + \dfrac{\chi_B}{K}(\mathbf{V}_b - \mathbf{V}) \end{cases} \tag{3.1}$$

where:

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{V} \cdot \nabla \tag{3.2}$$

is the Lagrangian derivative, which is formed by the time rate of change plus a convective contribution. Equation 3.1 is written in a non dimensional form, all variables are normalized with respect to some arbitrary reference values. A semi-Lagrangian scheme is employed, which means that equations are still solved

[1]Optimad, PABLO https://github.com/optimad/PABLO

using an Eulerian point of view, but, instead of approximating the partial time derivative and the convective term separately, time discretization is achieved by approximating the Lagrangian derivative:

$$\frac{D\mathbf{V}}{Dt} \approx \frac{\mathbf{V}^{n+1} - \mathbf{V}_\lambda^n}{\Delta t} \tag{3.3}$$

where $\mathbf{V}_\lambda^n$ is the velocity at the root of the characteristic, defined as:

$$\mathbf{V}_\lambda^n = \mathbf{V}\big(\mathbf{x} - \Delta t \mathbf{V}(\mathbf{x}, t^n), t^n\big) \tag{3.4}$$

using an explicit first order scheme to propagate the characteristic backwards. The finite volume method is applied to the integral form of equations 3.1, which is:

$$\begin{cases} \oint_{\partial\Omega} \mathbf{V} \cdot \mathbf{n} d\sigma = 0 \\ \dfrac{D}{Dt} \displaystyle\int_\Omega \mathbf{V} d\Omega = -\oint_{\partial\Omega} p\mathbf{n} d\sigma + \frac{1}{Re} \oint_{\partial\Omega} \frac{\delta \mathbf{V}}{\delta n} d\sigma + \frac{\chi_B}{K} \int_\Omega \big(\mathbf{V}_B - \mathbf{V}\big) d\Omega \end{cases} \tag{3.5}$$

Equation 3.5 is valid for each cell of the domain, thus, applying numerical integration and writing for one finite volume:

$$\begin{cases} \displaystyle\sum_i \mathbf{V}_i \cdot \mathbf{n}_i \Delta S_i = 0 \\ \dfrac{D\mathbf{V}_i}{Dt} \Delta\Omega_i = -\displaystyle\sum_j p_j \mathbf{n}_j \Delta S_j + \frac{1}{Re} \sum_j \left(\frac{\delta \mathbf{V}}{\delta n}\right)_j \Delta S_j + \frac{\chi_B}{K} \big(\mathbf{V}_b - \mathbf{V}\big) \Delta\Omega_i \end{cases}$$

where $\delta/\delta n$ is a centered approximation of the derivative in the face normal direction. Combining 3.4 with the space-discretized equations and using the predictor-corrector scheme, a classical three stage algorithm is obtained, whose prediction step consists in solving the following linear system:

$$\mathbf{V}_i^* - \frac{\Delta t}{\Delta\Omega_i} \frac{1}{Re} \sum_j \left(\frac{\delta \mathbf{V}^*}{\delta n}\right)_j \Delta S_j = \mathbf{V}_{\lambda i}^n \tag{3.6}$$

where for the viscous term an implicit Euler scheme has been employed. The projection step is obtained imposing the divergence free condition on the virtual velocity, leading to a finite volume discretized Poisson equation:

$$\sum_j \left(\frac{\delta p}{\delta n}\right)_j^{n+1} \Delta S_j = \frac{1}{\Delta t} \sum_j \mathbf{V}_j^* \cdot \mathbf{n}_j \Delta S_j \tag{3.7}$$

The linear systems in the first two steps are solved using GMRES [21] or the stabilized biconjugate gradient (BCGSTAB) [28] algorithm. The computed pressure is used in the correction step:

$$\mathbf{V}_i^{**} = \mathbf{V}_i^n - \frac{\Delta t}{\Delta\Omega_i} \sum_j p_j^{n+1} \mathbf{n}_j \Delta S_j \tag{3.8}$$

The penalization is taken into account as an additional correction, in order to finally obtain the velocity at the (n+1)-th step:

$$\mathbf{V}_i^{n+1} = (1 - \chi_{Bi})\mathbf{V}_i^{**} + \chi_{Bi}\mathbf{V}_{Bi} \tag{3.9}$$

In this algorithm the pressure is not involved in the prediction step, therefore during each projection step the pressure is completely re-evaluated. This scheme is said to be **non-incremental**, and it is more precise, since there is no cumulative error on the pressure field, but it is less stable than the incremental one, due to be bigger correction imposed by the pressure.

3.3 Treatment of level-jumps

Octrees can be described as hierarchical cartesian grids, which means that each refinement block is made by uniformly spaced cells, the transition regions where the mesh is non-uniform are called level-jumps and the correct discretization of the equations around these portions of the grid is fundamental for the accuracy of the whole numerical scheme. Several different approach can be employed in order to cope with level-jumps, in this work two methods will be presented, the first one is the diamond's method, introduced by Coudière [29], which is illustrated in figure 3.5.

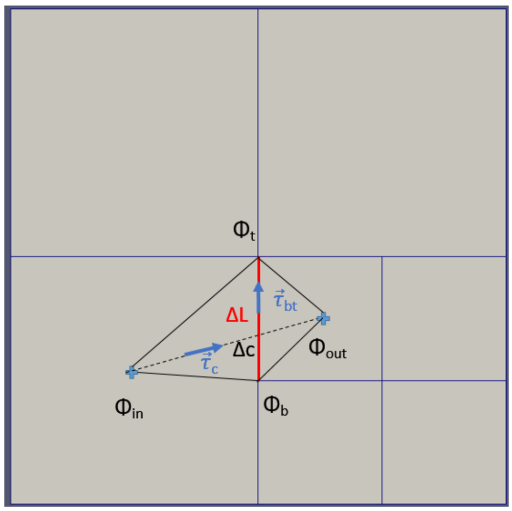


Figure 3.5: The diamond's method

The value of Φ in the center of the intersection is given by:

$$\Phi_i = \frac{1}{2}\left(\Phi_t + \Phi_b\right)$$

where Φ_t and Φ_b are the values on the nodes of the intersection. Since a fully collocated scheme is considered, nodal values are obtained by interpolating Φ among surrounding cells.

The gradient of Φ on the intersection is obtained solving the following linear system:

$$\begin{cases} \nabla\Phi \cdot \mathbf{t}_c = \dfrac{\Phi_{out} - \Phi_{in}}{\Delta c} \\ \nabla\Phi \cdot \mathbf{t}_{bt} = \dfrac{\Phi_t - \Phi_b}{\Delta l} \end{cases} \tag{3.10}$$

An order of convergence of 2 has been proven on the heat equation, as reported in [30].
Although the diamond's method results very accurate, it has been shown, by Olshanskii et al. [31], that some spurious discretely divergence-free modes may occur at level-jumps when a high number of degrees of freedom is used to compute fluxes. These modes can be suppressed by locally reducing the amount of degrees of freedom required by the discretization, forcing the modes not to be divergence-free anymore.

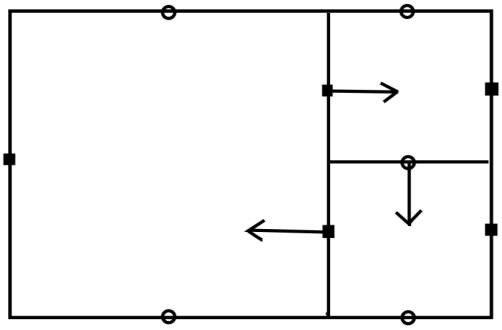


Figure 3.6: A divergence-free level-jump mode

In order to avoid level-jump modes, an alternative scheme has been developed, with the goal of using the lowest possible number of degrees of freedom in the proximity of the interface between the coarse and the fine grid.

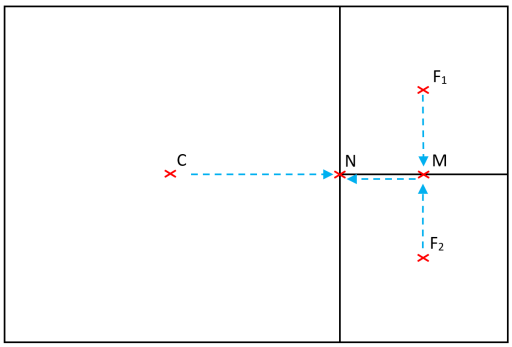


Figure 3.7: Alternative scheme for evaluating fluxes at level-jumps

The proposed discretization is illustrated in figure 3.7 for a bidimensional grid,

the value of the considered variable in the hanging node N is obtained by linear interpolation of the values in the three cells involved in the level-jump:

$$\Phi_N = \frac{1}{3}\Phi_C + \frac{2}{3}\Phi_M = \frac{1}{3}\Phi_C + \frac{2}{3}\frac{1}{2}\sum_i \Phi_{Fi} = \frac{1}{3}\big(\Phi_C + \Phi_{F1} + \Phi_{F2}\big) \tag{3.11}$$

in a similar manner the normal discrete gradient can be evaluated in N:

$$\left(\frac{\delta\Phi}{\delta n}\right)_N = \frac{\Phi_M - \Phi_C}{\frac{3}{2}\Delta l} = \frac{1}{3\Delta l}\big(\Phi_{F1} + \Phi_{F2}\big) - \frac{2}{3\Delta l}\Phi_C \tag{3.12}$$

Where a first order approximation of the normal derivative has been employed and Δl is the size of the finer cells. The values obtained for the node N are used to compute the fluxes between the coarse cell and both the finer ones, meaning that the flux entering the two small cells is the same. This is equivalent to treat the finer cells as a unique rectangular volume centered on the point M, in this way the spurious level-jump modes are suppressed, since they can be divergence-free only if all the three fluxes are equal to zero. Expressions 3.11 and 3.12 can be easily extended to a tridimensional geometry, where a coarse cell has four finer neighbours. The hanging node value is:

$$\Phi_N = \frac{1}{3}\Phi_C + \frac{2}{3}\frac{1}{4}\sum_i \Phi_{Fi} = \frac{1}{3}\Phi_C + \frac{1}{6}\sum_i \Phi_{Fi} \tag{3.13}$$

and the normal gradient:

$$\left(\frac{\delta\Phi}{\delta n}\right)_N = \frac{\Phi_M - \Phi_C}{\frac{3}{2}\Delta l} = \frac{1}{6\Delta l}\sum_i \Phi_{Fi} - \frac{2}{3\Delta l}\Phi_C \tag{3.14}$$

This approach has been validated by solving the 3D Poisson's equation:

$$\nabla^2\Phi = q \tag{3.15}$$

with forcing term:

$$q = 6\cos\left(x^2 + y^2 + z^2\right) - 4\big(x^2 + y^2 + z^2\big)\sin\left(x^2 + y^2 + z^2\right) \tag{3.16}$$

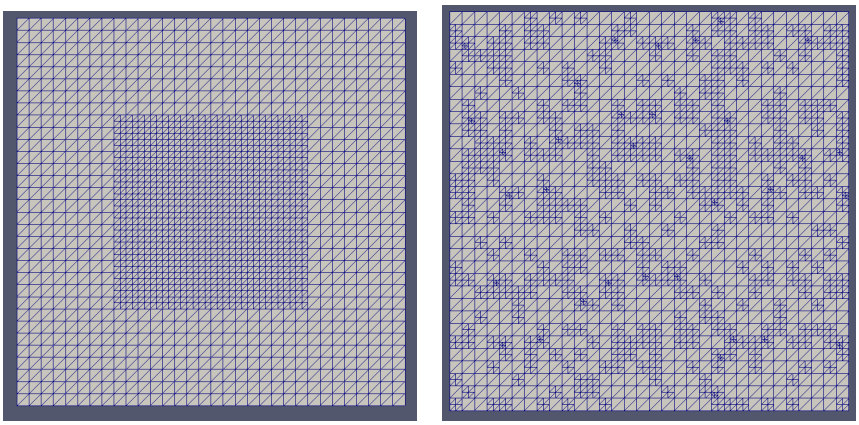
The analytical solution is known and it is:

$$\Phi_{exact} = \sin\left(x^2 + y^2 + z^2\right) + C \tag{3.17}$$

where C is a constant dependent on the boundary condition. A convergence analysis has been performed in three different configurations:

- case 1: domain size $L = 2$, single cubic refinement of size 1
- case 2: domain size $L = 4$, single cubic refinement of size 2
- case 3: domain size $L = 2$, multiple "random" refinements

Let ε be the absolute error: $\varepsilon = |\Phi - \Phi_{exact}|$, it is possible to evaluate the three norms of ε:



(a) *Single cubic refinement* (b) *Multiple random refinements*

Figure 3.8: Comparison of refinements

$$\|\varepsilon\|_{L_1} = \frac{1}{\Omega}\int_{\Omega} \varepsilon d\Omega \approx \frac{1}{\Omega}\sum_i \varepsilon_i \Delta\Omega_i \tag{3.18}$$

$$\|\varepsilon\|_{L_2} = \frac{1}{\Omega}\sqrt{\int_{\Omega} \varepsilon^2 d\Omega} \approx \frac{1}{\Omega}\sqrt{\sum_i \varepsilon_i^2 \Delta\Omega_i} \tag{3.19}$$

$$\|\varepsilon\|_{L_\infty} = \max_i \varepsilon_i \tag{3.20}$$

Defining $\Delta l_{min} = \min_i \Delta l_i$ and considering a 1n-grid and a 2n-grid such that $\Delta l_{min,2n} = 2\Delta l_{min,1n}$, the order of convergence ν is defined as:

$$\nu_{L_i} = \log_2\left[\frac{\left(\|\varepsilon\|_{L_i}\right)_{2n}}{\left(\|\varepsilon\|_{L_i}\right)_{1n}}\right] \tag{3.21}$$

Δl_{min}	$\|\varepsilon\|_{L_1}$	ν_{L_1}	$\|\varepsilon\|_{L_2}$	ν_{L_2}	$\|\varepsilon\|_{L_\infty}$	ν_{L_∞}
1,25E-01	4,43E-04	-	1,25E-03	-	3,80E-03	-
6,25E-02	1,50E-04	1,56	2,68E-04	2,23	7,88E-04	2,27
3,13E-02	5,35E-05	1,49	7,05E-05	1,93	1,73E-04	2,18
1,56E-02	1,61E-05	1,73	1,92E-05	1,87	4,10E-05	2,08
7,81E-03	4,42E-06	1,87	5,09E-06	1,92	1,01E-05	2,02
3,91E-03	1,16E-06	1,93	1,32E-06	1,94	2,49E-06	2,02

Table 3.1: Convergence analysis, case 1

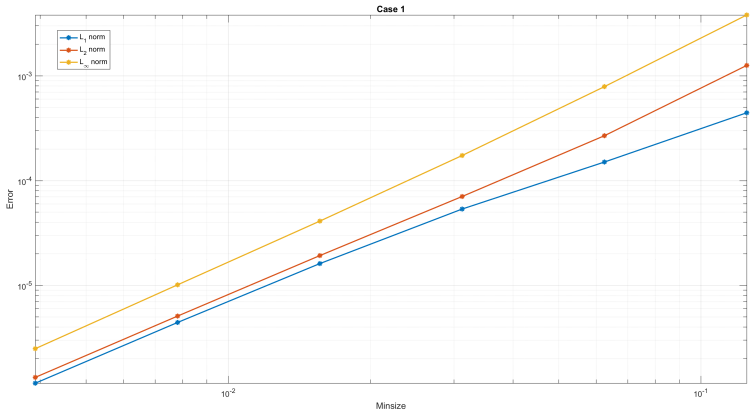


Figure 3.9: Convergence Analysis, case 1

Δl_{min}	$\|\varepsilon\|_{L_1}$	ν_{L_1}	$\|\varepsilon\|_{L_2}$	ν_{L_2}	$\|\varepsilon\|_{L_\infty}$	ν_{L_∞}
1,25E-01	3,76E-02	-	6,33E-03	-	1,09E-01	-
6,25E-02	1,04E-02	1,85	1,63E-03	1,96	2,46E-02	2,14
3,13E-02	2,70E-03	1,95	4,10E-04	1,99	6,24E-03	1,98
1,56E-02	6,86E-04	1,98	1,03E-04	2,00	1,59E-03	1,97
7,81E-03	1,73E-04	1,99	2,56E-05	2,00	4,08E-04	1,97
3,91E-03	4,28E-05	2,01	6,32E-06	2,02	1,04E-04	1,98

Table 3.2: Convergence analysis, case 2

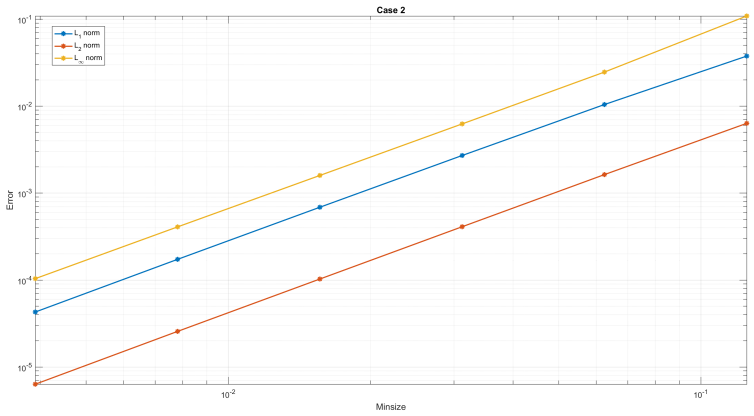


Figure 3.10: Convergence Analysis, case 2

Δl_{min}	$\|\varepsilon\|_{L_1}$	ν_{L_1}	$\|\varepsilon\|_{L_2}$	ν_{L_2}	$\|\varepsilon\|_{L_\infty}$	ν_{L_∞}
1,56E-02	8,56E-05	-	1,03E-04	-	4,25E-04	-
7,81E-03	2,36E-05	1,86	2,70E-05	1,94	1,15E-04	1,89
3,91E-03	6,26E-06	1,92	7,10E-06	1,93	3,15E-05	1,87
1,95E-03	1,61E-06	1,95	1,87E-06	1,92	8,59E-06	1,88
9,77E-04	4,99E-07	1,69	6,41E-07	1,55	2,31E-06	1,89

Table 3.3: Convergence analysis, case 3

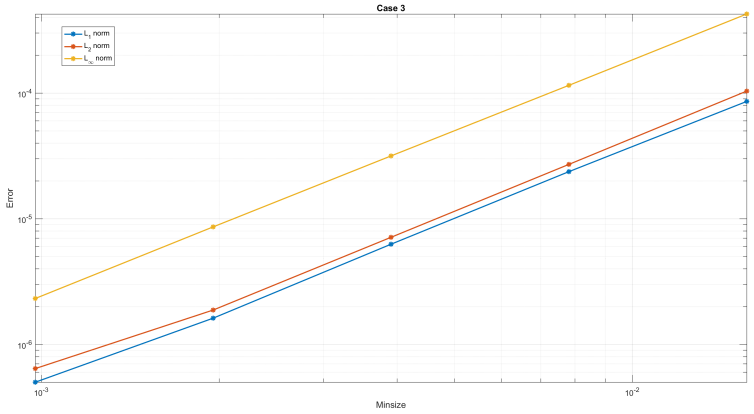
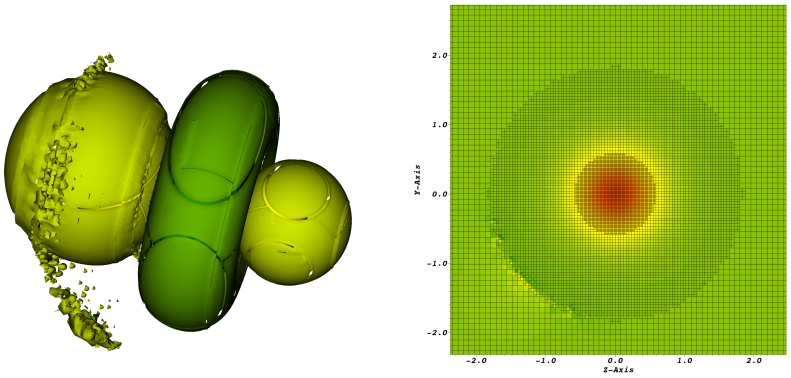


Figure 3.11: Convergence Analysis, case 3

Surprisingly, the order of convergence is superior than 1 in all of the test cases, for the first two a second order of convergence is achieved, whereas for the third ν suddenly falls once Δl_{min} becomes smaller than 0,001. The developed level-jump operator performs better than expected when the number of refinements is small when compared to the total number of cells, when the number of level-jumps becomes comparable with the total amount of grid points, such as the case of the randomly refined mesh, a loss of accuracy is registered.

3.4 Analysis of the 3D advection equation

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 few seconds. The numerical noise is clearly visible in figure 3.12, and it is located just outside the outer spherical refinement. The same result was obtained with both the diamond's method and the stabilized operators, furthermore, since the Poisson's equation is an elliptic problem, a perturbation is immediately felt in the whole domain, which means that, if there is a strong instability in one region, in another region there is a second instability that compensates for the first. In figure 3.12 there is no evidence of such a behaviour, therefore the focus has been directed on the analysis of the convective term of the discretized Navier-Stokes equations.



(a) *Contour plot of the pressure* (b) *Slice of the pressure pseudocolor plot*

Figure 3.12: Numerical instability, low Reynolds flow past a sphere

In order to better understand the behaviour of the semi-Lagrangian scheme, a simple test case was made on the 3D transport equation:

$$\begin{cases} \dfrac{D\varphi}{Dt} = \dfrac{\partial\varphi}{\partial t} + \mathbf{V}\cdot\nabla\varphi = 0 \\ \varphi_0 = \sin x \\ \varphi_{inlet} = \sin\left(x_{inlet} - 2t\right) \end{cases} \tag{3.22}$$

where $\mathbf{x} = (x, y, z)$ and $\mathbf{V} = (2, 0, 0)$. The exact solution of the problem is:

$$\Phi(\mathbf{x}, t) = sin(x - 2t) \tag{3.23}$$

From a numerical viewpoint, equation 3.22 discretized with a first order semi-Lagrangian schemes becomes:

$$\varphi_i^{n+1} = \varphi_\lambda^n = \varphi^n\left(\mathbf{x}_i - \Delta t\mathbf{V}_i^n\right) \tag{3.24}$$

which can be seen as a discrete method of characteristics. Since the root of the characteristic does not forcibly coincide with the center of a cell, its value has to be obtained by interpolation. It has been proved by Falcone et al. [32] that the L_∞ norm of the error of a semi-Lagrangian scheme has the following property:

$$\|\varepsilon\|_{L_\infty} \leq C\left(\Delta t^p + \frac{\Delta l^r}{\Delta t}\right) \tag{3.25}$$

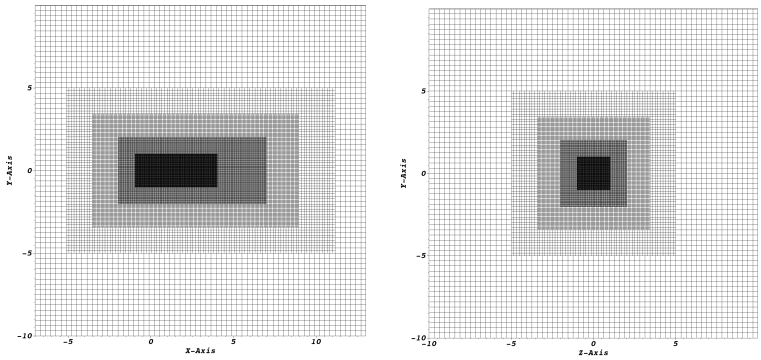
where p is the order of convergence of the time semi-discretization and r is the order of convergence of the interpolation. Since the CFL condition imposes that $\Delta t \sim \Delta l$, the interpolation must be one order more precise than the scheme in order to be consistent, which means that a first order semi-Lagrangian scheme requires a second order interpolation. Using polynomial in 3D can be painstaking because of the high number of neighbours that need to be considered for the stencil and also because of the large amount of possible configurations that can be created using adaptive mesh refinement techniques, in each of these cases the extension of the stencil is different, meaning that the polynomial basis has to

be adapted case-by-case. Due to these issues, Gaussian Radial Basis Function (RBF) interpolation is chosen:

$$\varphi(\mathbf{x}) = \sum_i w_i e^{\varepsilon r_i^2} \tag{3.26}$$

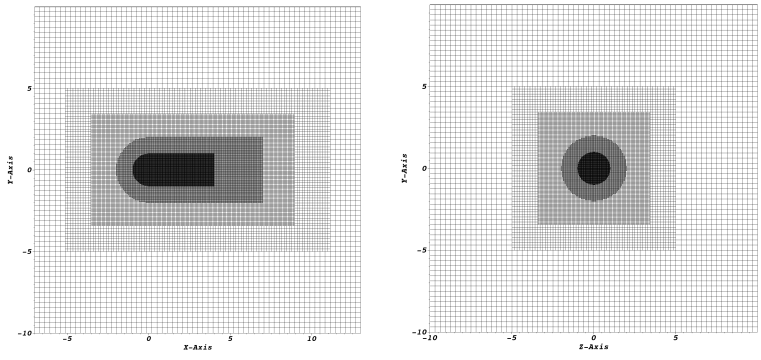
where w_i are the weights of the interpolation and $r_i = \|\mathbf{x} - \mathbf{x}_i\|$ and ε is a scaling parameter proportional to $1/\Delta l_i$. This method is second-order accurate, does not require adaptation of the basis and relies only on the distance between the desired point and the interpolation points.
The test case 3.22 is investigated using two different meshes:

- Case 1: only cartesian patch refinements are used
- Case 2: mesh with hemispherical refinements, such as the one used in figure 3.12



(a) *Slice of the grid, z=0* (b) *Slice of the grid, x=0*

Figure 3.13: Mesh for case 1



(a) *Slice of the grid, z=0* (b) *Slice of the grid, x=0*

Figure 3.14: Mesh for case 2

For each of the simulation the evolution of the L_1, L_2 and L_∞ norm of the error and the L_∞ norm of the solution are tracked. Figure 3.15 reports the time history of the L_∞ norm of φ for the two test cases, it is clear that both simulations diverge. The first test case remains stable for nearly a period of 0.2, then the maximum of the solution starts to increase. On the other hand, Case 2 is highly unstable, the solution starts diverging after less than 0.05 seconds, after then the L_∞ norm rises abruptly.

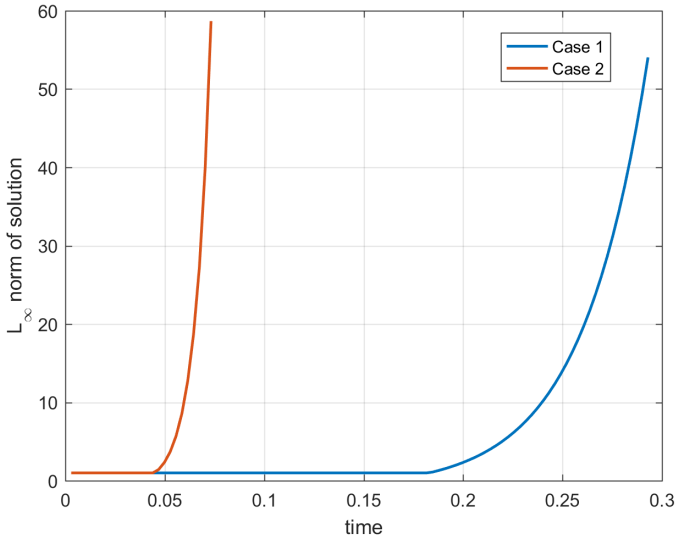
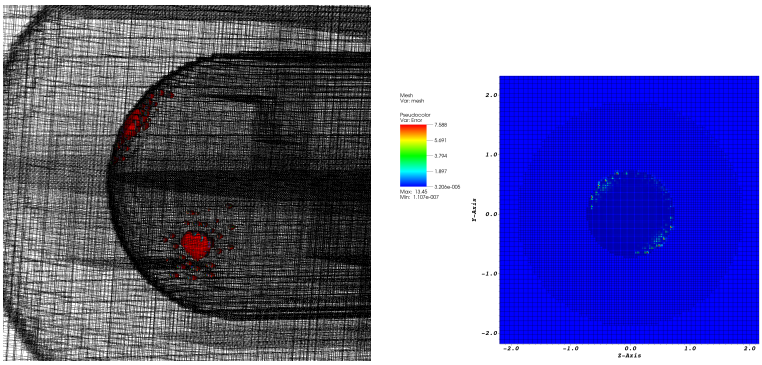


Figure 3.15: Evolution of L_∞ norm of the solution for the two test cases



(a) *3D view of the error, zoom on the inner spherical level-jump*

(b) *Slice of the error plot, x=-0.7*

Figure 3.16: Case 2, error plot

In figure 3.16 it is represented the local error for case 2, on the left, in red, the regions where the error is more than 1 are shown, whereas on the right the pseudocolor plot at $x = 0.7$ is reported. It is clearly visible that the error is concentrated in the proximity of the spherical level-jump, exactly as figure 3.12, which means that the cause of the instability of the Navier-Stokes algorithm

lies in the convection term. The result obtained on the cartesian-patch grid is actually quite surprising, since simulations on the low Reynolds flow past a sphere on those meshes did not present any sort of instability. Figure 3.16 although, might provide an explanation for this contradiction: the instability observed on the cartesian-patch grid is more gradual than the one occurring in case 2, it is thus possible that the viscous term in the prediction step is sufficiently strong to stabilize the scheme at low Reynolds numbers on cartesian-patch grids, whereas the error on the hemispherically refined mesh might grow too rapidly to be regularized by the diffusive term.
In order to better understand the instability mechanism and to find an appropriate fix, a convergence test of Gaussian RBF interpolations has been performed on the case 2 grid. In each cell a point is chosen by shifting along the x-direction using a random distance between 0 and $\Delta l_i/2$, the reference function is $\varphi = sin(x)$. The results are reported in table 3.4, where the relationship with the maximum size of the grid, $\Delta l_{max} = \max_i \Delta l_i$, is summarized.

Δl_{max}	$\|\varepsilon\|_{L_1}$	ν_{L_1}	$\|\varepsilon\|_{L_2}$	ν_{L_2}	$\|\varepsilon\|_{L_\infty}$	ν_{L_∞}	$\|\varphi_{int}\|_{L_\infty}$
6,25E-01	1,58E-03	-	2,39E-05	-	5,80E-03	-	1,00208
3,13E-01	4,15E-04	1,93	6,21E-06	1,95	1,47E-03	1,98	1,00005
1,56E-01	1,06E-04	1,97	1,57E-06	1,98	3,71E-04	1,99	1,00005
7,81E-02	2,87E-05	1,88	3,99E-07	1,98	1,12E-04	1,73	1,00003

Table 3.4: RBF convergence analysis

The convergence characteristics show that the interpolation is working properly, the error decreases with an almost second order slope, but the L_∞ norm of φ is higher than the unity in every case, this means that the Gaussian RBF interpolation does not conserve the maximum and the minimum of a discrete function, hence the semi-Lagrangian scheme employed does not result to be Total Variation Diminishing (TVD). In computational fluid dynamics, using flux limiters is a common practice that enables the formulation of high-order TVD schemes [33]; a similar fix is proposed in order to force the solution of the transport equation to be limited. Let a and b be the maximum and the minimum values of the transported variable among the interpolation points of the stencil:

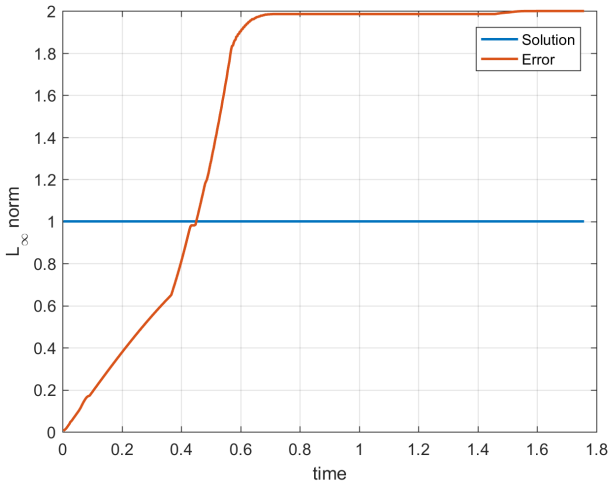
$$a = \max_i \varphi_i \tag{3.27}$$

$$b = \min_i \varphi_i \tag{3.28}$$

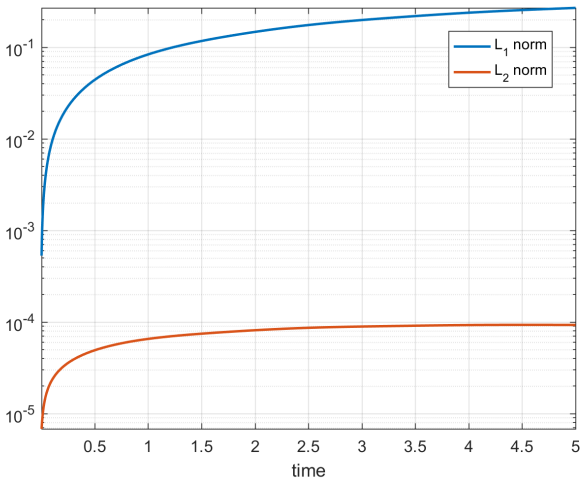
The proposed limiter function is:

$$\mathcal{F} = \max\big(\min\big(a, \varphi_{int}\big), b\big) \tag{3.29}$$

where φ_{int} is the result of the Gaussian RBF interpolation. $\mathcal{F}$ enables the conservation of the L_∞ norm of the solution and, as a natural consequence, it prevents the instability of the semi-Lagrangian scheme.
In figure 3.17 the results for the TVD semi-Lagrangian scheme on the case 2 grids are summarized, the L_∞ norm of the solution is stable thanks to the limiter function, whereas the L_∞ norm of the error increases until reaching a



(a) *Error and solution L_∞ norms*



(b) *Error L_1 and L_2 norms*

Figure 3.17: Results, Case 2, TVD semi-Lagrangian scheme

stationary value of 2, which means that the phase of the sinusoidal solution gradually shifted. The same behaviour is found for the L_1 and L_2 norms of the error.

3.5 Flow past a sphere

In order to assess the performance of the Navier-Stokes Octree solver two popular benchmarks have been used:

- flow past a sphere at $Re = 500$
- flow past a circular cylinder

Firstly, the sphere flow at a very low Reynolds number has been investigated using three different test cases, in this way all the features proposed in this chapter have been tested and validated within the fractional step projection algorithm. These simulations were performed using the PLAFRIM cluster, whose nodes are composed by a couple of 12-core Intel Haswell CPUs with a clock speed of 2.5GHz and a shared memory of 128GB. The set-up of first test case is the following:

- origin of the domain: (-5,-10,-10)
- length of the domain: 20
- 2.57 million cells
- cartesian-patch mesh
- non-TVD convection scheme
- diamond's method based differential operators
- number of cores: 24
- wall clock hours of simulation: 72

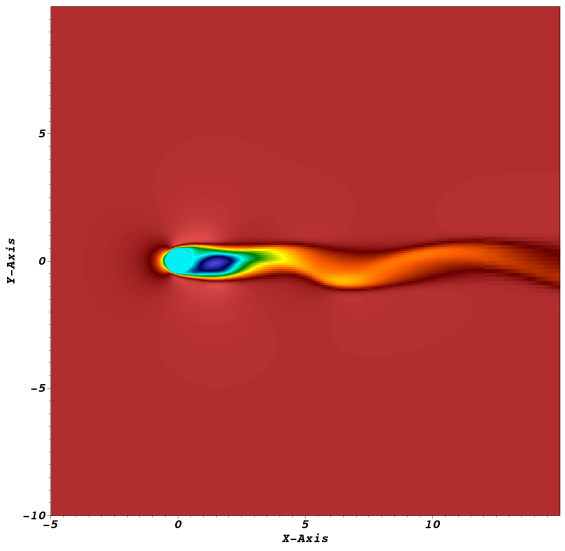


Figure 3.18: X Velocity plot at $t = 62$, test case 1

The set-up of the second test case is quite similar to the first one:

- origin of the domain: (-5,-10,-10)
- length of the domain: 20
- 2.57 million cells
- cartesian-patch mesh
- non-TVD convection scheme
- stabilized differential operators
- number of cores: 24
- wall clock hours of simulation: 72

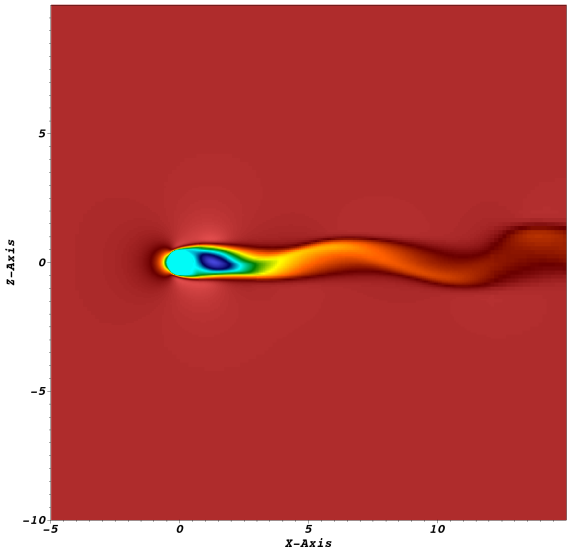


Figure 3.19: X Velocity plot at $t = 73$, test case 2

The only difference between the first two test cases is the use of the stabilized operators described in this chapter, allowing a fair comparison of the performance of the two different level-jump discretizations. The third test case differs from the first two, the mesh is more refined, spherical level-jumps are introduced, in order to validate the limiter function for the convective term. The set-up is:

- origin of the domain: (-7,-10,-10)
- length of the domain: 20
- 4.98 million cells
- hemispherically refined mesh
- TVD convection scheme

- stabilized differential operators
- number of cores: 72
- wall clock hours of simulation: 72

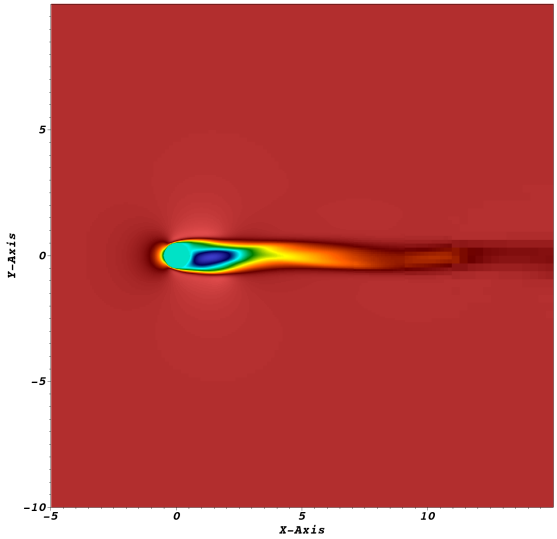


Figure 3.20: X Velocity plot at $t = 46$, test case 3

Figure 3.18, 3.19 and 3.20 show that the structure of the wake is similar for all the test cases. For the test case 3, it is possible to see that the wake is not fully developed yet. A comparison of the values of the drag coefficient C_D has been made too, both in term of time evolution and mean values.

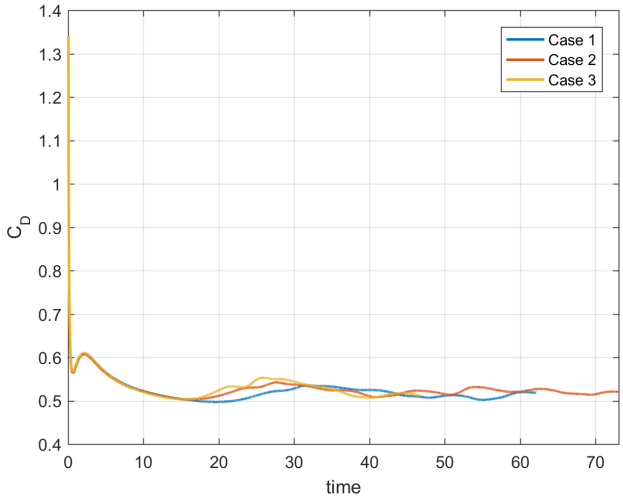


Figure 3.21: Evolution of the drag coefficient

Figure 3.21 reports the time evolution of the drag coefficient for the three test cases, it is possible to observe that the three C_D curves are identical until 15 seconds of simulations, after that some differences are noticeable in the time history, but overall the behaviour of the three simulations is comparable. After 40 seconds, the numerical transitory phase ends, and the shedding phenomenon starts to develop, curves stabilize and start oscillating around their mean values with a definite frequency. The average C_D obtained are in line with results found in literature, as remarked in table 3.5.

Case	C_D
Present work, case 1	0.5125
Present work, case 2	0.5203
Present work, case 3	0.5138
Campregher et al. [34]	0.520
Fornberg [35]	0.4818
Fadlun et al. [36]	0.4758

Table 3.5: Comparison of drag coefficient values

3.6 Flow past a cylinder

The flow past a cylinder benchmark has been used to test the high Reynolds behaviour of the Octree Navier-Stokes solver, coupled with the Wall Modelled Large Eddy Simulation approximation. The Vreman model [7] has been introduced in the Octree solver as an additional prediction step:

$$\mathbf{V}_i^{**} = \mathbf{V}_i^{*} + \frac{\Delta t}{\Delta \Omega_i} \frac{\nu_e}{L U_\infty} \sum_j \left(\frac{\delta \mathbf{V}^*}{\delta n}_j \right) \Delta S_j \tag{3.30}$$

where L is the reference length of the problem. The wall function adjusts the velocity field after the penalization step.
The Reynolds number for the test case is:

$$Re = \frac{U_\infty D}{\nu} = 140000 \tag{3.31}$$

The following set up has been employed:

- Domain: [-8,16]x[-12,12]x[-12,12],
- Grid size: 76 millions of cells,
- 85 points on the diameter of the Cylinder,
- streamwise direction: X-axis,
- Cylinder axis parallel to the Z-axis,
- Smagorinksy Constant: 0.17,
- CFL constant: 0.7,
- viscous wall function.

The biggest downside of using the PABLO Octree library is being limited to square domains in 2D and cubic domains in 3D, therefore a cylinder of length 24 has been simulated. The consequence of this limitation is that, even though it is possible to employ local grid refinements, these advantage cannot be fully exploited, since the length of the cylinder impose a very refined grid along the whole Z-axis, therefore, there is not a big decrease in the number of cells compared to the Cartesian code.
The simulation has been performed on the OCCIGEN machine, using 60 nodes composed by two 12-core Intel Haswell CPUs @2.6 GHz, for a total of 1440 cores. In this configuration for each 30 convective times of simulations, 24 hours of wall clock time are required. Six day-long runs were carried out, for a total of roughly 200000 computation hours, in which 166 convective times were simulated.

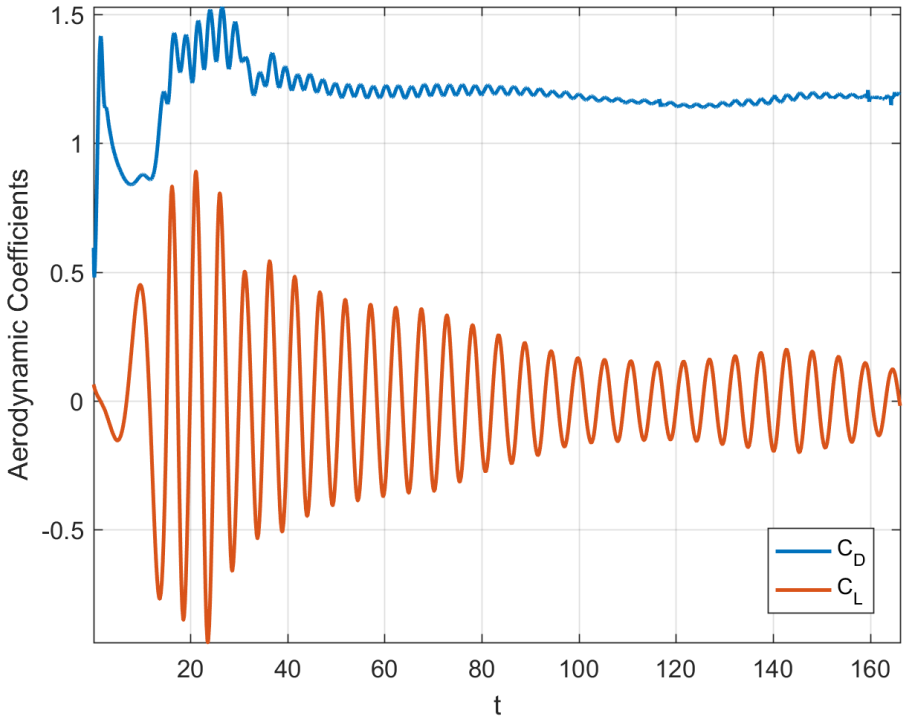
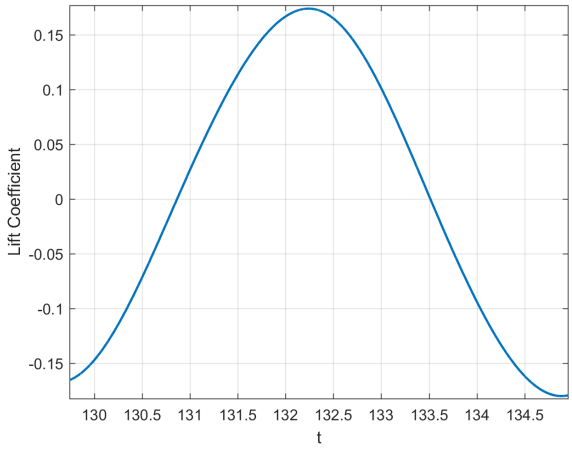
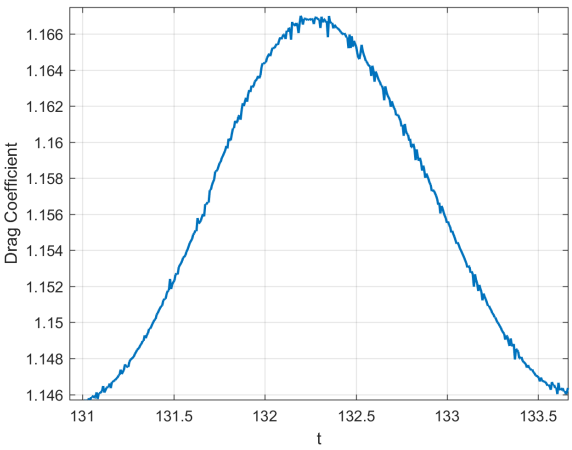


Figure 3.22: Evolution of the aerodynamic coefficients

Figure 3.22 reports the evolution of the lift coefficient C_L and the drag coefficient C_D as a function of the adimensional time. A numerical transient is observed in the first 20 convective times, then the shedding phenomenon starts to develop and periodic fluctuations in forces are visible. Short period oscillations start to stabilize after 40 convective times and start to assume a regular pattern. Some long period amplitude oscillation are present, they are clearly visible in the evolution of the lift coefficient. Overall, the fluctuation of C_L is smoother than the evolution of C_D, that presents a more random character, as it is observable in figure 3.23.



(a) *Oscillation of* C_L



(b) *Oscillation of* C_D

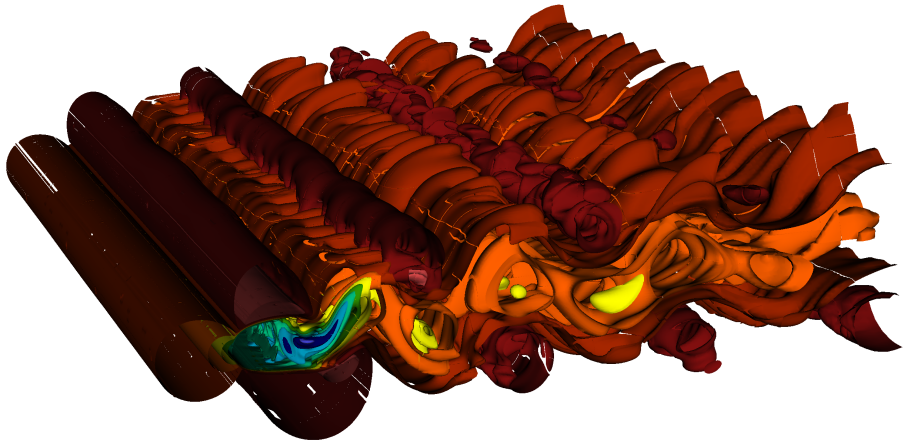
Figure 3.23: Detail of an oscillation of the aerodynamic coefficients

Case	C_D
Octree code, WMLES	1.182
NASCAR3D code, WMLES	1.084
Luo et al. [25], PANS	$0.82 - 1.28$
Luo et al. [25], DES	0.84
Kim [23], LES	1.21
Breuer [22], LES	$0.971 - 1.454$
Travin et al. [24], DES	1.08
Cantwell et al. [18], exp.	1.237

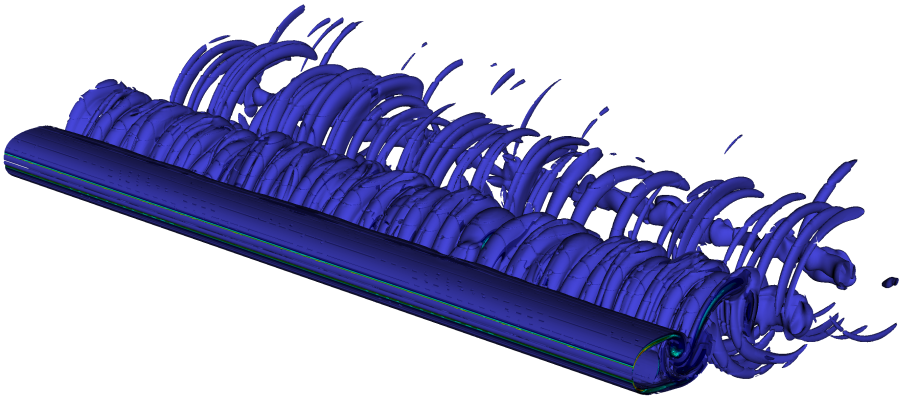
Table 3.6: Comparison of drag coefficient values

An average value of the drag coefficient has been evaluated over the last 120 convective times. Table 3.6, reports the comparison between the results obtained

with the Octree code, the NASCAR3D code and the one found in scientific literature. The value obtained with the Octree solver is slightly higher than the one discussed in chapter 2, but they are both in line with the ones from other authors.



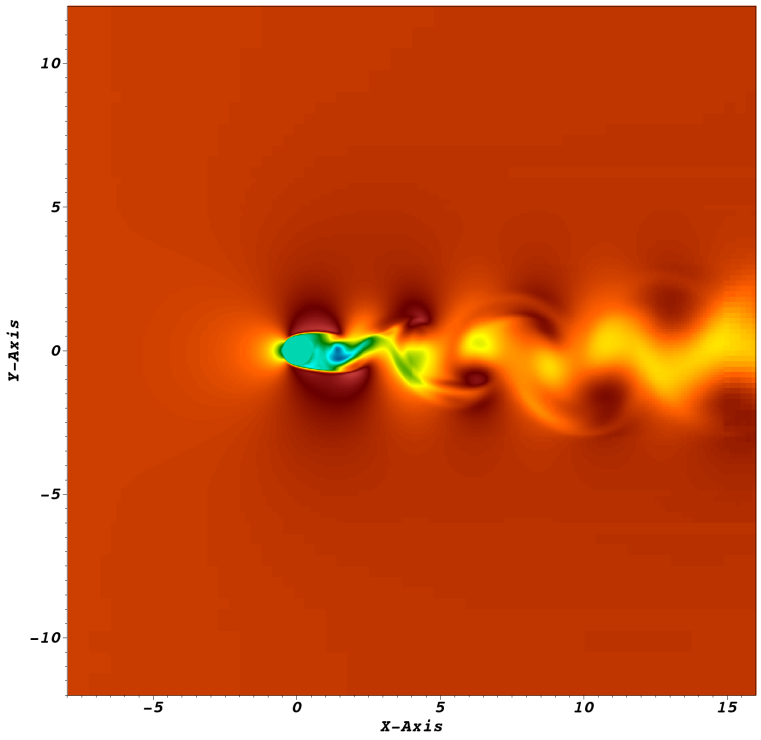
(a) *Contour surfaces of the x velocity, 10 levels*



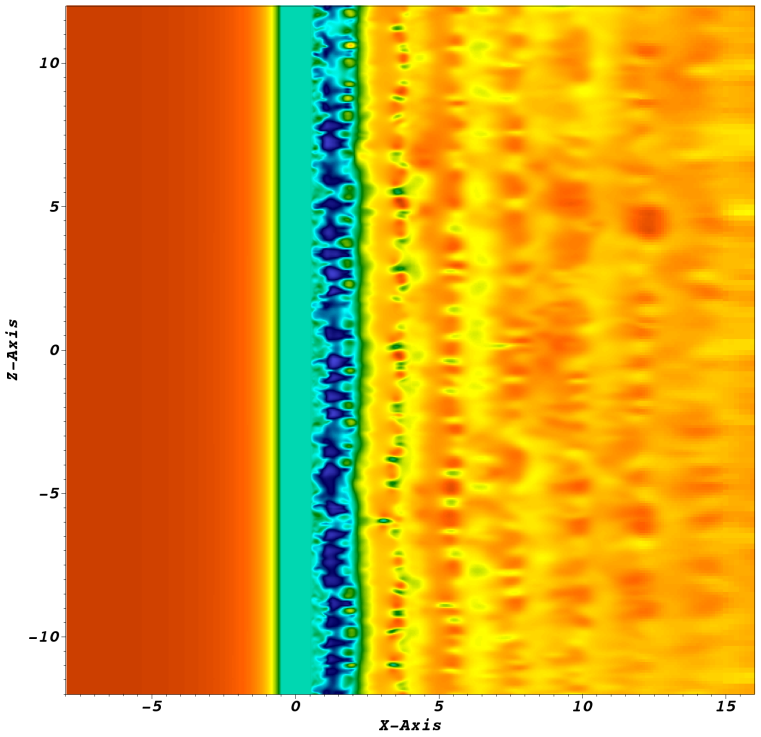
(b) *Contour surfaces of the norm of vorticity, 20 levels*

Figure 3.24: Flow past a cylinder at $Re = 140000$, Octree solver

In figure 3.24 are represented the contour surfaces of the x velocity and of the norm of the vorticity, comparing the images with the one obtained with the NASCAR3D code, it is possible to notice that there are very few small turbulent structures in the wake.

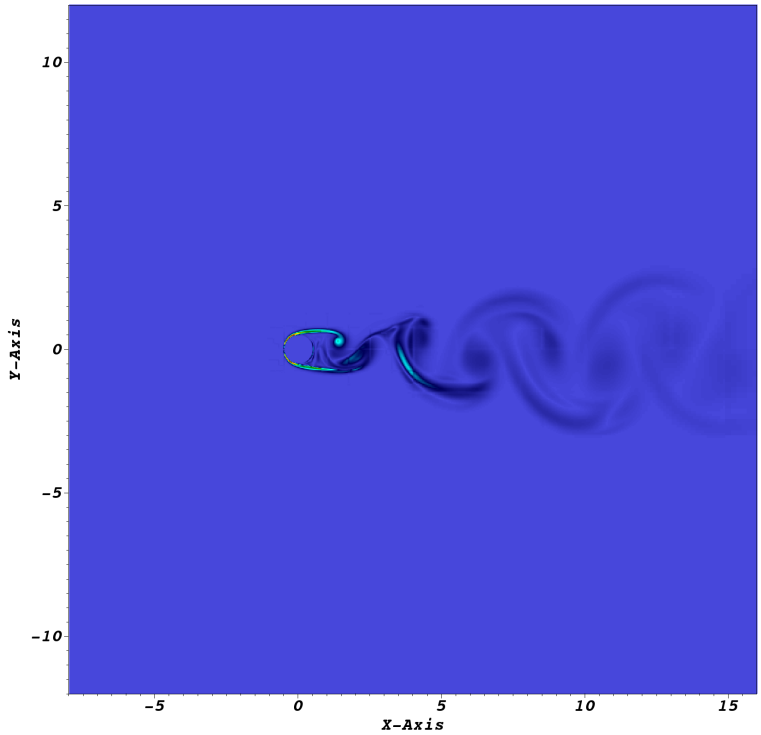


(a) *Plot of x velocity,* $z = 0$

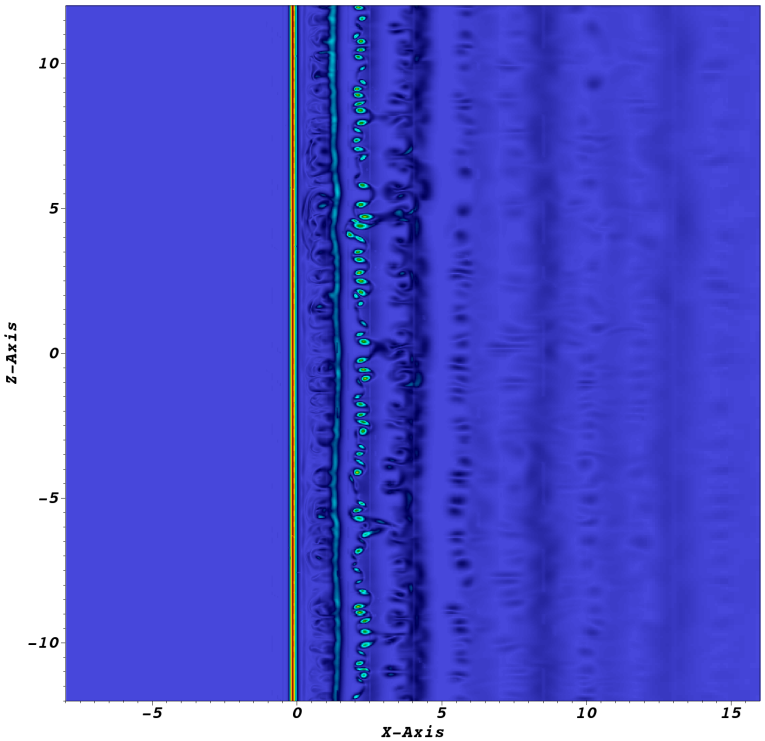


(b) *Plot of x velocity,* $y = 0$

Figure 3.25: Flow past a cylinder at $Re = 140000$, Octree solver



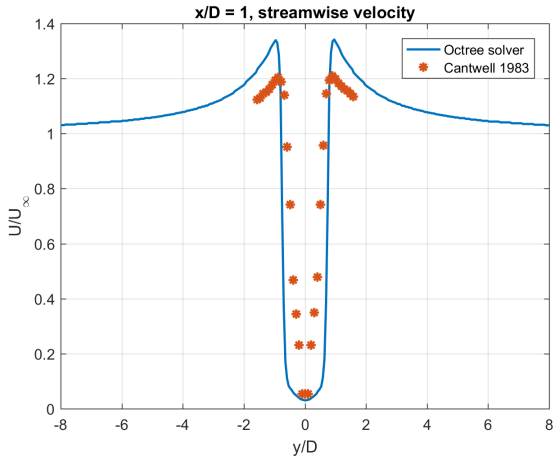
(a) *Plot of the norm of vorticity,* $z = 0$



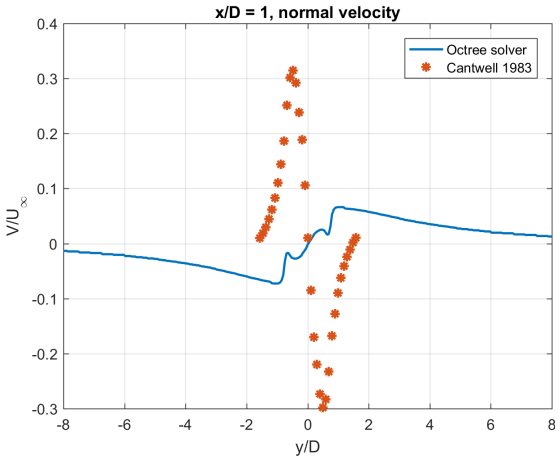
(b) *Plot of the norm of vorticity,* $y = 0.5$

Figure 3.26: Flow past a cylinder at $Re = 140000$, Octree solver

Figures 3.25 and 3.26 report the plots of the x velocity and the norm of the vorticity for $x = 0$ and $y = 0$, the first impression, given by the contour plots, is confirmed, the number of small eddies in the wake is much inferior when the Octree code is used, this can be explained considering that the NASCAR3D code is second order accurate in time and space, whereas the Octree solver is only first order in time and also in space, because of the presence of level jumps. The numerical dissipation of the first order approximation kills the small turbulent structures, thus the simulation is dominated by large eddies. Another factor that contributes to this lack is the resolution of the grid, which is comparable to the coarse one used in chapter 2 only in proximity of the cylinder, in the far wake the mesh is coarser, therefore the simulation is a bit underresolved far away from the cylinder. Although the worse resolution, the overall structure of the flow is well described, the separation is laminar and the transition occurs in the wake where large eddies are clearly visible and the evolution of the turbulence is proper, transition, stretching and dissipation can be identified in figure 3.26.



(a) *Average streamwise velocity profile*



(b) *Average normal velocity profile*

Figure 3.27: Comparison of numerical results with experimental values

In an effort to reproduce the good results obtained with the cartesian solver, average velocity profiles have been evaluated and compared with the experimental results of Cantwell et al. [18]. The profiles are reported in figure 3.27, while the streamwise velocity might be not too far from the reference data, the normal velocity is totally off, exactly as it happened with the coarse mesh simulation examined in chapter 2. It is therefore clear that the simulation is not resolved enough to evaluate the mean flow field with accuracy, even though no spurious oscillation is observed and aerodynamic coefficient are in line with those found in literature.

Chapter 4

Wind turbine blade modelling

This chapter deals with the creation of a virtual model of the turbine blade in order to perform fluid dynamics, structural and coupled simulations. The first major difficulty is the conversion of 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 discussed in chapter 3 and the finite element code presented in chapter 1. The other significant issue is dealing with the movement of the blade, that rotates around the hub at a known rotational speed.

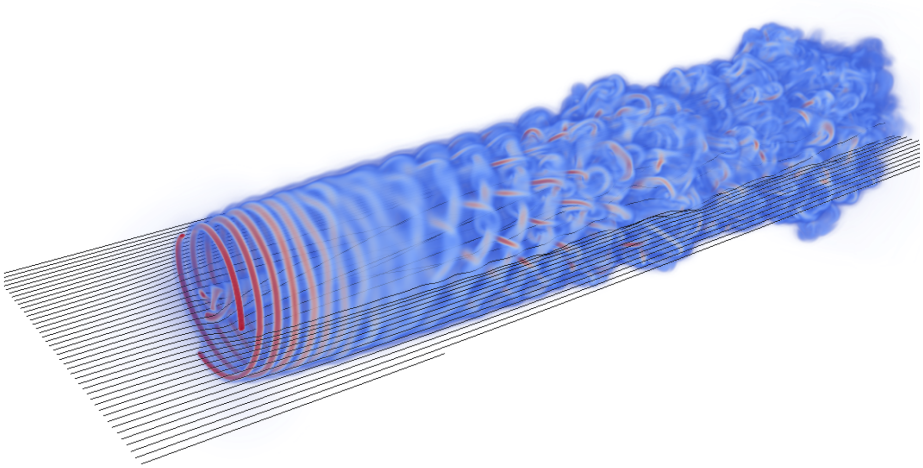


Figure 4.1: Norm of vorticity plot, wind turbine wake (Chatelain et al. [37])

There are two different ways to simulate a rotating object:

- using a fixed reference frame and update the position of the blade at each time step in order to take into account the body motion;
- considering a rotating reference frame in which the body is fixed.

Thanks to the immersed boundary method both the approaches are feasible, but the second one remains more affordable, since there is no need to constantly update the level-set function, which is a time consuming operation, especially for a complex geometry such as the one that will be considered. Furthermore, since the blade is fixed, an optimized mesh can be built around it with no needs of dynamic AMR techniques.
In order to operate in a non-inertial reference frame, the equations of motion for both the structure and the fluid needs to be modified due to the presence of two apparent forces: Coriolis and centrifugal force. In this chapter the inclusion of these two effects in the finite element model and in the Navier-Stokes code will be discussed, then some preliminary results will be presented.

4.1 Geometric model

One of the most time consuming operations in modern computational fluid dynamics is the pre-processing of complex CAD models in order to generate a proper unstructured mesh that preserves the geometry. The quality of the grid has a strong influence on the fidelity of the simulation, therefore, meshing is iteratively optimized and this process still heavily relies on human expertise. The immersed boundary method, coupled with Octrees, aims to automatize this operation, using mesh that are not body fitted and cubic cells, whose isotropy represents the best option for highly accurate simulations. On the other hand, thanks to years of experience in the use of unstructured mesh generators, standard CAD formats can be easily imported and used to generate grids, without any sort of conversion. The same cannot be said for the present approach, where the surface of the object needs to be described by a structured lattice of Lagrangian markers.

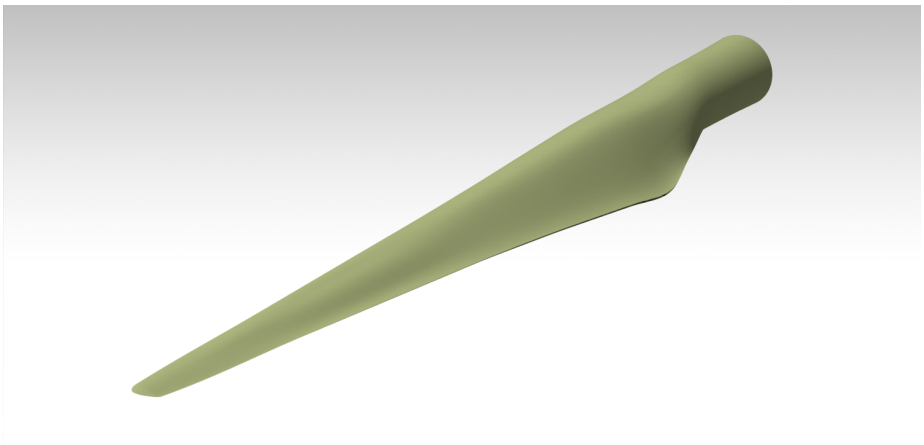


Figure 4.2: 3D rendering of the blade

CAD models are described thanks to non-uniform rational b-splines (NURBS). STEP-files contain all the necessary information to generate the NURBS representation of an object, stored in a standardized format. Creating a structured array of Lagrangian markers starting from a STEP-file is not straightforward,

first, the CAD surface needs to be cut into equally spaced sections, as represented in 4.3.

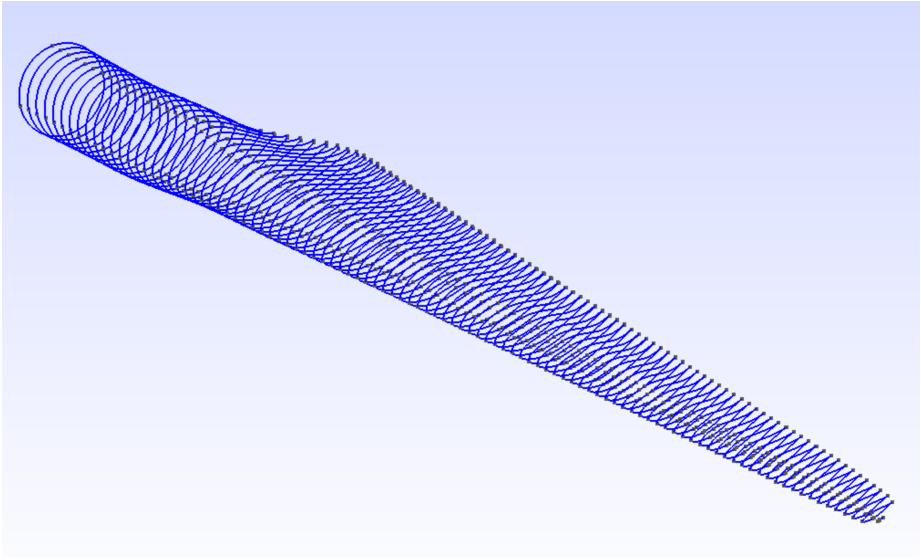
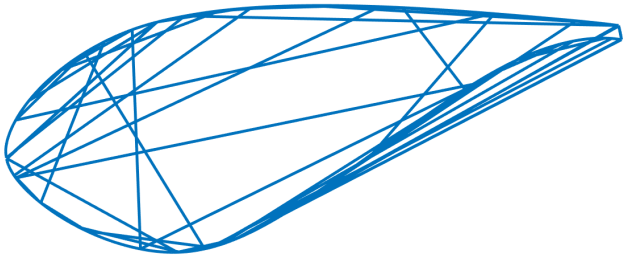
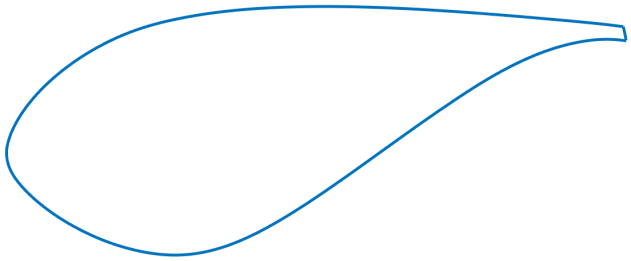


Figure 4.3: Sections of the blade, $N_S = 100$

Then using Gmsh[1], each section is be meshed using a large amount 1D rectilinear elements in order to generate an unstructured array of points.



(a) *Unordered set of airfoil points*



(b) *Reordered set of airfoil points*

Figure 4.4: Airfoil discretization

[1] http://gmsh.info/

The array obtained thanks to Gmsh is not properly ordered, as it is possible to observe in figure 4.4a, thus it is not directly usable to generate Lagrangian markers, the points need to be clockwise reordered. Moreover, some profiles present a very sharp trailing edge, where b-splines fail to accurately represent the geometry, as it is possible to observe in figure 4.5, therefore these sections need to be cropped.

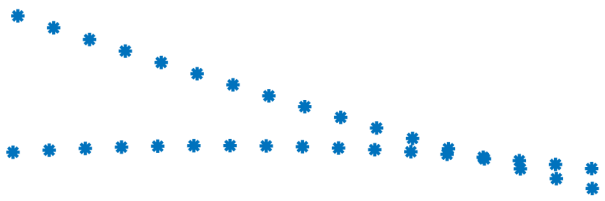


Figure 4.5: Detail of the trailing edge

Points generated by Gmsh are not equally spaced, so the one last step is needed to generate the structured grid of markers. Thanks to the reordering it is simple to define the curvilinear coordinate of each point, defined as:

$$\begin{cases} s_i = s_{i-1} + \int_{i-1}^{i} dl \approx s_{i-1} + \sqrt{\left(x_i - x_{i-1}\right)^2 + \left(y_i - y_{i-1}\right)^2 + \left(z_i - z_{i-1}\right)^2} \\ s_0 = 0 \end{cases}$$

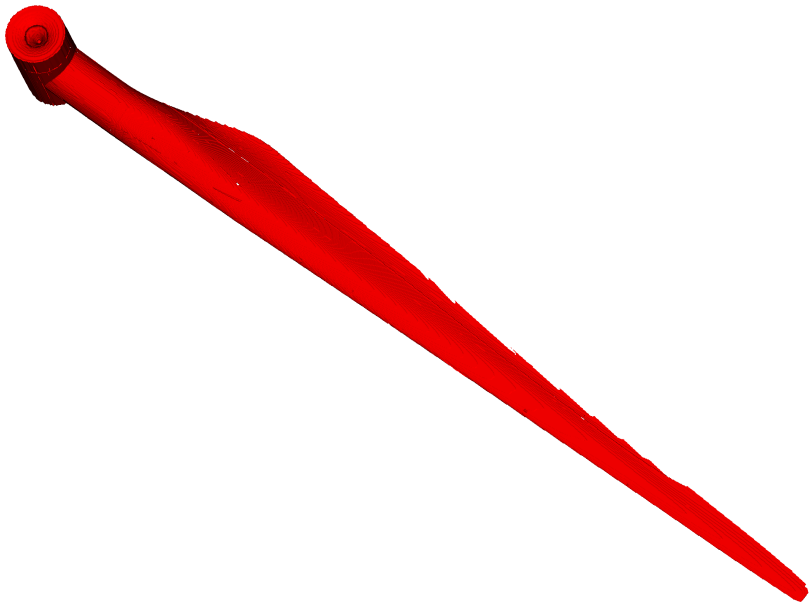
using a linear approximation. The initial point is the trailing edge of each profile. All sections are discretized with the same number of Lagrangian markers, once the amount is defined it is possible to evaluate the spacing:

$$\Delta L = \frac{s_{tot}}{N_{LM}} = \frac{1}{N_{LM}} \oint_l dl$$

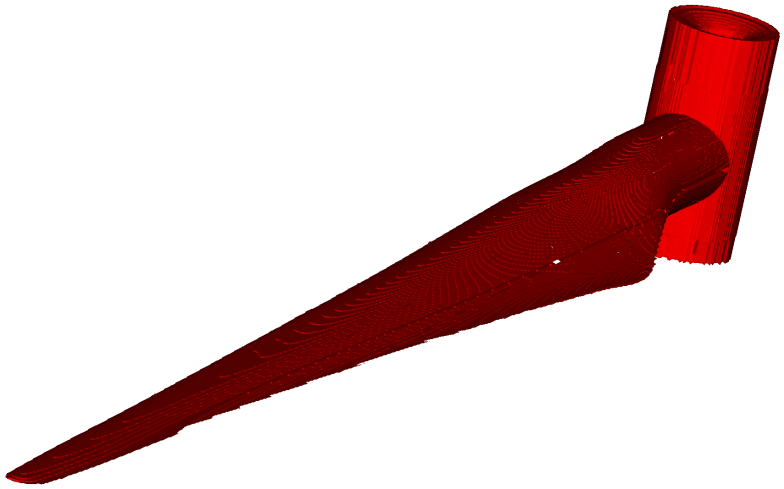
which is different for each profile. Thanks to the curvilinear coordinate each Lagrangian marker is defined as:

$$s_i = s_{i-1} + \Delta L$$

therefore it is possible to obtain the coordinate of the i-th marker by linearly interpolating between two points of the original set generated thanks to Gmsh. Within the Navier-Stokes solver the geometry is described thanks to the level set function φ which is evaluated using the Lagrangian markers, basically for each point in the domain the closest couple of markers is researched in order to determine the distance from the surface. The sign of φ is equal to the sign of the scalar product between the outward surface normal and the vector that links the surface and the point in the domain.



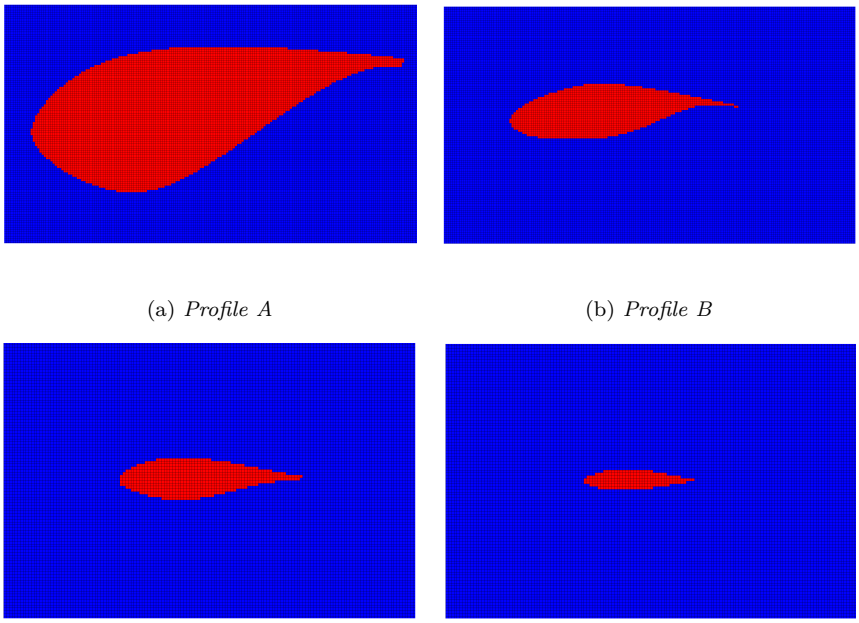
(a) *First point of view*



(b) *Second point of view*

Figure 4.6: Marker-based geometry of the blade

In figure 4.6 is reported the reconstruction of the surface of the blade using the array of Lagrangian markers created thanks to the implementation of the described procedure. It is possible to observe that there is a loss of accuracy in the proximity of the trailing edge, where the details of the geometry tend to be very small, therefore a very fine mesh is needed in order to capture them.
In figure 4.7 are represented four profiles in different position along the blade



(a) *Profile A* (b) *Profile B*

(c) *Profile C* (d) *Profile D*

Figure 4.7: Example of discretized blade profiles

span, from A to D the spanwise position increases. The reconstruction becomes more accurate as the profile is closer to the hub, since the chord becomes shorter in the proximity of the tip, and consequently the airfoils become less and less thick, which means that, in order to describe the geometry the grid needs to be refined.

4.2 Rotating frame structure model

The finite element model needs to be updated with the inclusion of the Coriolis and the centrifugal force. 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. In other words, the axis of rotation is fixed. Moreover, the dynamic effects caused by other rotational deformation speeds are neglected, gyroscopic torques are not considered, in order to keep the structure model as simple as possible.
The finite element describer in chapter 1 can be described as a combination of:

- rod element for axial deformations;
- beam element for bending;
- bar element for torsion.

The centrifugal force can thus be modelled considering a simple rod element with shape functions:

$$[N] = \begin{bmatrix} 1 - \frac{x}{L} & \frac{x}{L} \end{bmatrix} \tag{4.1}$$

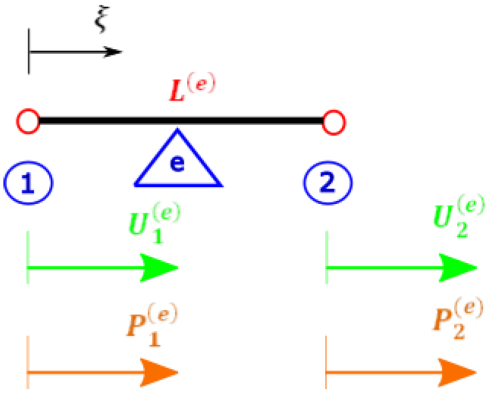


Figure 4.8: Schematization of a rod element

The work of the centrifugal force can be expressed as:

$$W_{cf} = \int_V u\rho\Omega^2(x+u)dV = \int_x \int_A u\rho\Omega^2(x+u)dAdx \tag{4.2}$$

introducing ρ_m, defined as:

$$\rho_m = \int_A \rho dA \tag{4.3}$$

expression 4.2 becomes:

$$W_{cf} = \int_x u\rho_m\Omega^2(x+u)dx = \int_x u\rho_m\Omega^2 u dx + \int_x u\rho_m\Omega^2 x dx \tag{4.4}$$

Expressing u as a function of the vector of nodal degrees of freedom:

$$W_{cf} = \{U\}^T \left[\left(\int_x [N]^T \rho_m\Omega^2 [N]\, dx \right) \{U\} + \int_x [N]^T \rho_m\Omega^2 x dx \right] \tag{4.5}$$

The centrifugal force generates two contributions, the second term, which is the more relevant, is the external load that would be applied to a infinitely stiff object, whereas the first term, also known as **spin softening**, is due to the elasticity of the rod: the tip of a rotating blade is pushed outwards by the centrifugal force, thus it is subject to a bigger load, compared to a non deformable object. In this sense the rotation makes apparently decreases the stiffness of the rod. The spin softening term can be combined with the stiffness matrix:

$$[\mathcal{K}] = [K] - \int_x [N]^T \rho_m\Omega^2 [N]\, dx \tag{4.6}$$

considering ρ_m constant within each element:

$$\int_x \left[N\right]^T \rho_m \Omega^2 \left[N\right] dx = \rho_m \Omega^2 \int_0^L \begin{bmatrix} 1 - \frac{x}{L} \\ \frac{x}{L} \end{bmatrix} \begin{bmatrix} 1 - \frac{x}{L} & \frac{x}{L} \end{bmatrix} dx \tag{4.7}$$

hence, integrating:

$$\int_x \left[N\right]^T \rho_m \Omega^2 \left[N\right] dx = \rho_m \Omega^2 \begin{bmatrix} \frac{L}{3} & \frac{L}{6} \\ \frac{L}{6} & \frac{L}{3} \end{bmatrix} \tag{4.8}$$

Reassembling the matrix for the complete element, the second row and corresponds to the sixth one and so for the column. Therefore:

$$\begin{cases} \mathcal{K}_{1,1} = K_{1,1} - \rho_m \Omega^2 \dfrac{L}{3} \\ \mathcal{K}_{1,6} = K_{1,6} - \rho_m \Omega^2 \dfrac{L}{6} \\ \mathcal{K}_{6,1} = K_{6,1} - \rho_m \Omega^2 \dfrac{L}{6} \\ \mathcal{K}_{6,6} = K_{6,6} - \rho_m \Omega^2 \dfrac{L}{3} \end{cases} \tag{4.9}$$

elsewhere:

$$\mathcal{K}_{i,j} = K_{i,j} \tag{4.10}$$

The right hand side contribution of the centrifugal force is:

$$\left\{F_{cf}\right\} = L \rho_m \Omega^2 \int_0^L \begin{bmatrix} 1 - \frac{x}{L} \\ \frac{x}{L} \end{bmatrix} (x + x_0) dx \tag{4.11}$$

where x_0 is the coordinate of the first node of the element. Thus, integrating:

$$\left\{F_{cf}\right\} = \rho_m \Omega^2 x_0 \begin{bmatrix} \frac{L}{2} \\ \frac{L}{2} \end{bmatrix} + \rho_m \Omega^2 \begin{bmatrix} \frac{L^2}{6} \\ \frac{L^2}{3} \end{bmatrix} \tag{4.12}$$

The work of the Coriolis force can be written as:

$$W_{cor} = -2 \int_V \left\{u\right\}^T \rho \left(\left\{\Omega\right\} \wedge \left\{\dot{u}\right\} \right) dV = -2 \int_x \left\{u\right\}^T \rho_m \left(\left\{\Omega\right\} \wedge \left\{\dot{u}\right\} \right) dx \tag{4.13}$$

The external product between the rotational speed and the velocity vector is:

$$\left\{\Omega\right\} \wedge \left\{\dot{u}\right\} = \begin{Bmatrix} 0 \\ 0 \\ \Omega \end{Bmatrix} \wedge \begin{Bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{Bmatrix} = \Omega \begin{Bmatrix} \dot{v} \\ -\dot{u} \\ 0 \end{Bmatrix} \tag{4.14}$$

The speed of the tension centre can be evaluated deriving the expression of the kinematics:

$$\begin{cases} \dot{u} = \dfrac{\partial u_t}{\partial t} \\ \dot{v} = \dfrac{\partial v_c}{\partial t} - (z_t - z_c) \dfrac{\partial \theta_x}{\partial t} \end{cases} \tag{4.15}$$

which can be expressed in matrix form:

$$\begin{Bmatrix} \dot{v} \\ -\dot{u} \\ 0 \end{Bmatrix} = \frac{\partial}{\partial t} \begin{pmatrix} 0 & 1 & 0 & -h & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{Bmatrix} u \\ v \\ w \\ \theta_x \\ \theta_y \\ \theta_z \end{Bmatrix} \tag{4.16}$$

with $h = z_t - z_c$. Naming the 6x6 matrix $\big[B\big]$, and expressing the displacement as a function of the nodal degrees of freedom:

$$\begin{Bmatrix} \dot{v} \\ -\dot{u} \\ 0 \end{Bmatrix} = \big[B\big] \big[N\big] \big\{\dot{s}\big\} \tag{4.17}$$

where $\big[N\big]$ is the matrix of the shape functions of the complete beam element:

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

with L_1 and L_2 being the first order Lagrange polynomials:

$$\begin{cases} L_1 = 1 - \xi \\ L_2 = \xi \end{cases} \tag{4.18}$$

and H_i are the third order Hermite polynomials:

$$\begin{cases} H_1 = 1 - 3\xi^2 + 2\xi^3 \\ H_2 = 3\xi^2 - 2\xi^3 \\ H_3 = L(\xi - 2\xi^2 + \xi^3) \\ H_4 = L(-\xi^2 + \xi^3) \end{cases} \tag{4.19}$$

with $\xi = x/L$. The nodal Coriolis Force vector can be therefore expressed as:

$$\big\{F_{cor}\big\} = \Big\{-2\rho_m \Omega \int_x \big[N\big]^T \big[B\big] \big[N\big]\, dx\Big\} \big\{\dot{s}\big\} = \big[C\big] \big\{\dot{s}\big\} \tag{4.20}$$

where $\big[C\big]$ is the Coriolis matrix.

Simplifying the matrix-matrix products and integrating, the Coriolis matrix is:

$$\left[C\right] = -2\rho_m\Omega \begin{pmatrix} 0 & -\frac{7L}{20} & 0 & \frac{hL}{3} & 0 & -\frac{3L^2}{20} & 0 & -\frac{3L}{20} & 0 & \frac{hL}{6} & 0 & \frac{L^2}{30} \\ \frac{7L}{20} & 0 & 0 & 0 & 0 & 0 & \frac{3L}{20} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{3L^2}{20} & 0 & 0 & 0 & 0 & 0 & \frac{13L^2}{30} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{3L}{20} & 0 & \frac{hL}{6} & 0 & -\frac{13L^2}{30} & 0 & -\frac{7L}{20} & 0 & \frac{hL}{3} & 0 & \frac{L^2}{20} \\ \frac{3L}{20} & 0 & 0 & 0 & 0 & 0 & \frac{7L}{20} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{L^2}{20} & 0 & 0 & 0 & 0 & 0 & -\frac{L^2}{30} & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Taking into account the newly added contributions, the space discretized equations of the elasticity becomes:

$$\left[M\right]\left\{\ddot{d}\right\} + \left[C\right]\left\{\dot{d}\right\} + \left[\mathcal{K}\right]\left\{d\right\} = \left\{F_{ext}\right\} + \left\{F_{cf}\right\} = \left\{F\right\} \tag{4.21}$$

Applying the Newmark method with $\gamma = \beta = 1/2$, the linear system for the solution of the dynamic structural problem is obtained:

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

4.3 Rotating frame fluid model

Considering the low rotational speed of a wind turbine blade, it is possible that the structural effects of the rotation might be negligible, although it is certain that the effects on the fluid will be very important. It is therefore essential the development of an accurate model capable of dealing with non-inertial coordinate systems. Considering an inertial frame (X, Y, Z) and a rotating one (X', Y', Z'), the relationship between the acceleration of a point expressed in the two references is:

$$\mathbf{a} = \mathbf{a_{ni}} + \mathbf{a}' + \omega \wedge \left(\omega \wedge \mathbf{r}'\right) + \frac{d\omega}{dt} \wedge \mathbf{r}' + 2\omega \wedge \mathbf{V}' \tag{4.22}$$

where $\mathbf{r}'$, $\mathbf{V}'$, $\mathbf{a}'$ are respectively the position, velocity and acceleration in the rotating frame, ω is the angular velocity vector, $\mathbf{a_{ni}}$ is the linear acceleration of the non inertial frame and $\mathbf{a}$ is the acceleration in the inertial coordinate system. Considering a purely rotating frame with constant revolution rate, equation 4.22 becomes:

$$\mathbf{a} = \mathbf{a}' + \omega \wedge \left(\omega \wedge \mathbf{r}'\right) + 2\omega \wedge \mathbf{V}' \tag{4.23}$$

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

$$\mathbf{a} = \frac{D\mathbf{V}}{Dt} \tag{4.24}$$

hence,

$$\frac{D\mathbf{V}}{Dt} = \frac{D\mathbf{V}'}{Dt} + \omega \wedge \left(\omega \wedge \mathbf{r}'\right) + 2\omega \wedge \mathbf{V}' \tag{4.25}$$

The momentum equation in the Lagrangian form is:

$$\frac{D\mathbf{V}}{Dt} = -\frac{\nabla p}{\rho} + \nu \nabla^2 \mathbf{V} \tag{4.26}$$

The relationship between $\mathbf{V}$ and $\mathbf{V}'$ is:

$$\mathbf{V} = \mathbf{V}' + \omega \wedge \mathbf{r}', \tag{4.27}$$

it is simple to demonstrate that the velocity inducted by the rotation is divergence-free, therefore:

$$\begin{cases} \nabla \cdot \mathbf{V} = \nabla \cdot \mathbf{V}' \\ \nabla^2 \mathbf{V} = \nabla^2 \mathbf{V}' \end{cases} \tag{4.28}$$

Substituting expressions 4.28 and 4.26 into 4.25 and considering the continuity equation, the incompressible Navier-Stokes equations for a rotating reference frame are obtained:

$$\begin{cases} \nabla \cdot \mathbf{V}' = 0 \\ \dfrac{D\mathbf{V}'}{Dt} = -\dfrac{\nabla p}{\rho} + \nu \nabla^2 \mathbf{V}' - \omega \wedge \left(\omega \wedge \mathbf{r}'\right) - 2\omega \wedge \mathbf{V}' \end{cases} \tag{4.29}$$

The two non inertial terms act as volume forces and they can be accounted in the prediction step of the Chorin algorithm:

$$\mathbf{V}_i^* - \frac{\Delta t}{\Delta \Omega_i} \frac{1}{Re} \sum_j \left(\frac{\delta \mathbf{V}^*}{\delta n} \right)_j \Delta S_j = \mathbf{V}_{\lambda i}^n - \omega \wedge \left(\omega \wedge \mathbf{r}_i\right) - 2\omega \wedge \mathbf{V}_i^n \tag{4.30}$$

all the vectors are referred to the rotating reference frame, apices are omitted for the sake of clarity. For the present work the y-axis of the domain is taken as the rotation axis of the wind turbine, therefore:

$$\omega = \begin{Bmatrix} 0 \\ \Omega \\ 0 \end{Bmatrix} \tag{4.31}$$

hence, the centrifugal and Coriolis acceleration are:

$$\omega \wedge \left(\omega \wedge \mathbf{r}_i'\right) = \omega \wedge \begin{Bmatrix} \Omega z' \\ 0 \\ -\Omega x' \end{Bmatrix} = -\Omega^2 \begin{Bmatrix} x' \\ 0 \\ z' \end{Bmatrix} \tag{4.32}$$

$$\omega \wedge \mathbf{V}' = -2\Omega \begin{Bmatrix} w' \\ 0 \\ -u' \end{Bmatrix} \tag{4.33}$$

In order to assess the scheme, a simple test case is investigated, assuming that the velocity in the fixed reference is uniform and parallel to the rotational axis of the turbine:

$$\mathbf{V} = \begin{Bmatrix} 0 \\ U_\infty \\ 0 \end{Bmatrix} \Rightarrow \mathbf{V}' = \mathbf{V} - \omega \wedge \mathbf{r}' = \begin{Bmatrix} -\Omega z \\ U_\infty \\ \Omega x \end{Bmatrix} \tag{4.34}$$

For this field of relative velocity, the Lagrangian derivative is:

$$\frac{D\mathbf{V}'}{Dt} = \frac{\partial \mathbf{V}'}{\partial t} + (\mathbf{V}' \cdot \nabla)\mathbf{V}' = \begin{Bmatrix} u' \frac{\partial u'}{\partial x} + v' \frac{\partial u'}{\partial y} + w' \frac{\partial u'}{\partial z} \\ u' \frac{\partial v'}{\partial x} + v' \frac{\partial v'}{\partial y} + w' \frac{\partial v'}{\partial z} \\ u' \frac{\partial w'}{\partial x} + v' \frac{\partial w'}{\partial y} + w' \frac{\partial w'}{\partial z} \end{Bmatrix} \tag{4.35}$$

Manipulating expression 4.35, the following relationship is obtained:

$$\frac{D\mathbf{V}'}{Dt} = \begin{Bmatrix} -\Omega w' \\ 0 \\ \Omega u' \end{Bmatrix} = \begin{Bmatrix} -\Omega^2 x \\ 0 \\ -\Omega^2 z \end{Bmatrix} \tag{4.36}$$

Since the relative velocity is linear, the viscous term is zero. It it thus found that the Coriolis acceleration and the convective term balance the centrifugal acceleration. Hence:

$$\nabla p = 0 \tag{4.37}$$

The rotating frame Navier-Stokes solver must be able to reproduce a uniform pressure and a stationary velocity field.

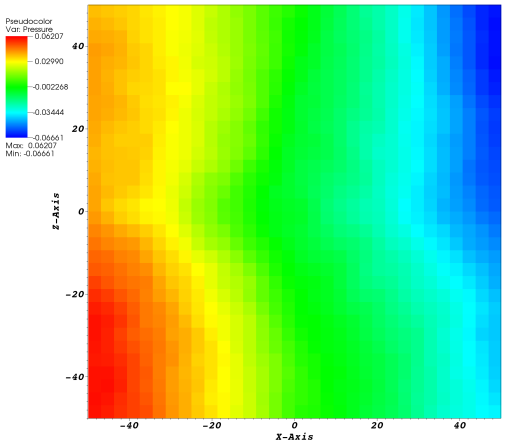


Figure 4.9: Pressure field, rotating frame test case

The test case has been carried out using a 32^3 uniform mesh on a domain of size 100, the rotational speed is $\Omega = 0.595$ and $U_\infty = 1$. Dirichlet conditions are imposed on every boundary except for the outlet, where a uniform Neumann condition is employed. In figure 4.9 the pressure field after one iteration is represented, whereas in figure 4.10 is plotted the norm of the pressure gradient.

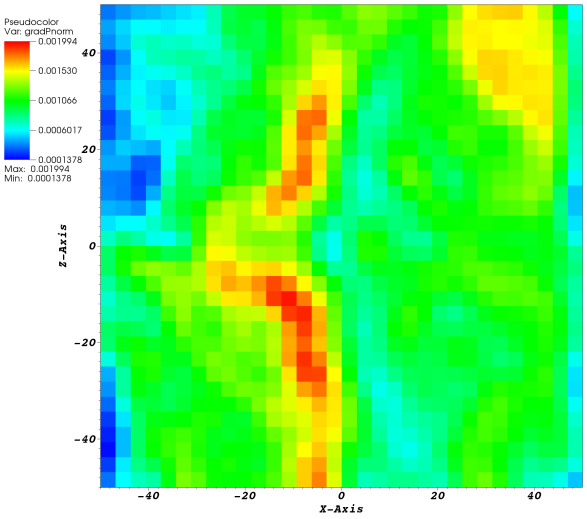


Figure 4.10: Norm of the pressure gradient, rotating frame test case

Even though the norm of the pressure gradient is of the order of 10^{-3}, a smaller value should be obtained, since only one iteration has been performed. In order to understand the origin of the spurious pressure gradient, the divergence of the virtual velocity $\mathbf{V}^*$ has been analyzed, the plot is reported in figure 4.11.

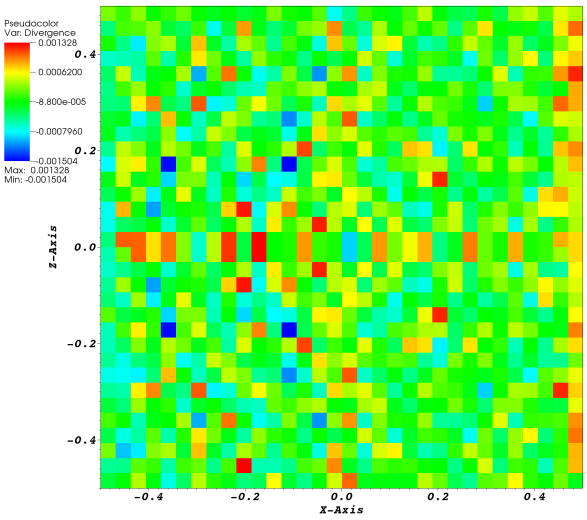


Figure 4.11: Virtual velocity divergence, rotating frame test case

It is possible to observe that there is a relevant numerical noise in the divergence, which means that the cause of the pressure gradient has to be searched in the prediction step. For this test case it is possible to analytically evaluate the velocity at the root of the characteristic:

$$\mathbf{V}_\lambda^n = \mathbf{V}^n\big(\mathbf{x} - \Delta t \mathbf{V}^n(\mathbf{x})\big) \tag{4.38}$$

hence,

$$\mathbf{x} - \Delta t \mathbf{V}^n(\mathbf{x}) = \begin{Bmatrix} x \\ y \\ z \end{Bmatrix} - \Delta t \begin{Bmatrix} -\Omega z \\ U_\infty \\ \Omega x \end{Bmatrix} = \begin{Bmatrix} x + \Delta t \Omega z \\ y - \Delta t U_\infty \\ z - \Delta t \Omega x \end{Bmatrix} \tag{4.39}$$

thus,

$$\mathbf{V}_\lambda^n = \mathbf{V}^n \left(\begin{Bmatrix} x + \Delta t \Omega z \\ y - \Delta t U_\infty \\ z - \Delta t \Omega x \end{Bmatrix} \right) = \begin{Bmatrix} -\Omega z + \Delta t \Omega^2 x \\ U_\infty \\ \Omega x + \Delta t \Omega^2 z \end{Bmatrix} \tag{4.40}$$

The analytic velocity can be forced into the prediction step, in order to compare the results obtained using the interpolated field at the root of the characteristics and the exact one. The divergence field, reported in figure 4.12, is uniform and equal to 0, the predicted velocity is already divergence-free, therefore no correction is required, which means that the responsible of the numerical noise in the rotating frame test case is the evaluation of the velocity at the root of the characteristics.

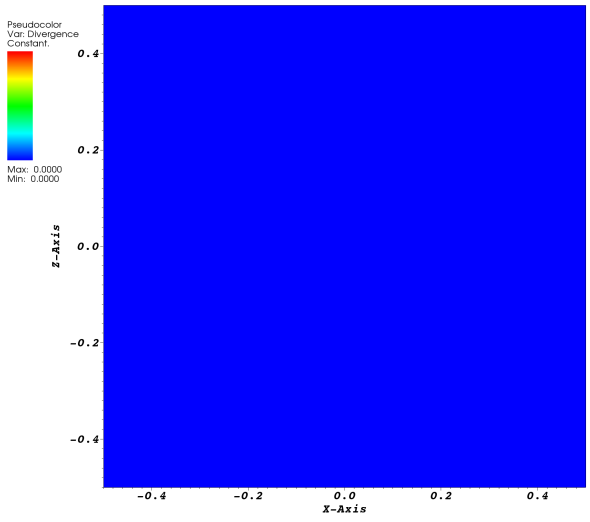
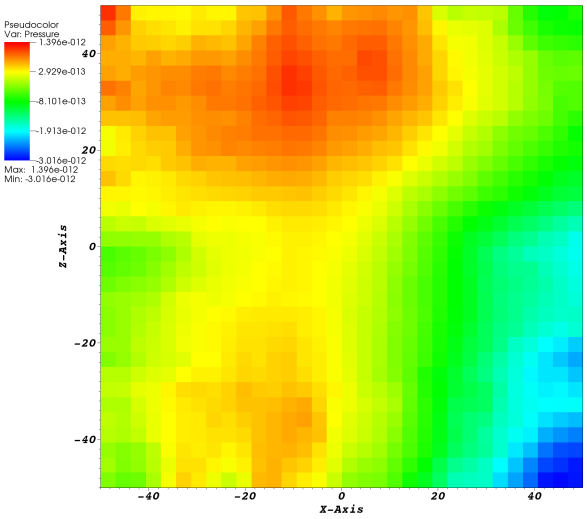
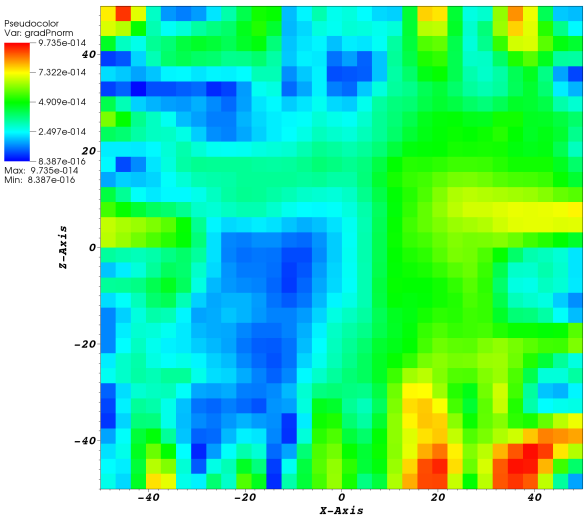


Figure 4.12: Virtual velocity divergence, rotating frame test case, analytic $\mathbf{V}_\lambda^n$

In figure 4.13 the pressure field and the norm of the pressure gradient are shown, there is not a perfect uniformity, but the noise is much smaller than the previously considered test and it is uniquely generated by the Poisson solver. Looking at the values of the pressure gradient norm it is possible to notice that they are close to the machine epsilon, which means that they are totally negligible.



(a) *Pressure field*



(b) *Norm of the pressure gradient*

Figure 4.13: Rotating frame test case, analytic $\mathbf{V}_\lambda^n$

In a real world problem it is not possible to recover the analytic velocity at the root of the characteristics, because the presence of the body in the domain disturbs the flow field, although part of the methodology can be used to enhance the semi-Lagrangian reconstruction.
The velocity field can be split into an undisturbed component and a perturbation:

$$\mathbf{V} = \mathbf{V}_{uf} + \mathbf{V}_p = \begin{Bmatrix} -\Omega z \\ U_\infty \\ \Omega x \end{Bmatrix} + \begin{Bmatrix} u_p \\ v_p \\ w_p \end{Bmatrix} \tag{4.41}$$

the perturbation $\mathbf{V}_p$ is caused by the presence of the body and it is negligible in the far field, where the undisturbed velocity $\mathbf{V}_{uf}$ is dominant.

Let $\mathbf{x}^*$ be the root of the characteristic:

$$\mathbf{x}^* = \mathbf{x} - \Delta t \mathbf{V}^n(\mathbf{x}) \tag{4.42}$$

then:

$$\mathbf{V}_\lambda^n = \mathbf{V}^n(\mathbf{x}^*) = \mathbf{V}_{uf}^n(\mathbf{x}^*) + \mathbf{V}_p^n(\mathbf{x}^*) \tag{4.43}$$

Splitting $\mathbf{V}_\lambda^n$ into the undisturbed field and the perturbation allows to analytically recover part of the information:

$$\mathbf{V}_{uf}^n(\mathbf{x}^*) = \begin{Bmatrix} -\Omega z^* \\ U_\infty \\ \Omega x^* \end{Bmatrix} \tag{4.44}$$

whereas the perturbation field is evaluated interpolating the information from neighbour cells:

$$\mathbf{V}_p^n(\mathbf{x}^*) = \sum_i w_i \mathbf{V}_{p,}(\mathbf{x}_i) \tag{4.45}$$

This procedure corresponds, from a physical point of view, to interpolate the velocity field in a non-rotating reference frame that moves along the y-direction with velocity U_∞.
In the considered test case the perturbation field is not present since there is not an obstacle inside the domain, therefore the proposed method is equivalent to the imposition of an analytic semi-Lagrangian reconstruction. A comparison of the original method and the enhanced one has been performed over 100 iterations, the results are summarized in figure 4.14 and in tables 4.1 and 4.2.

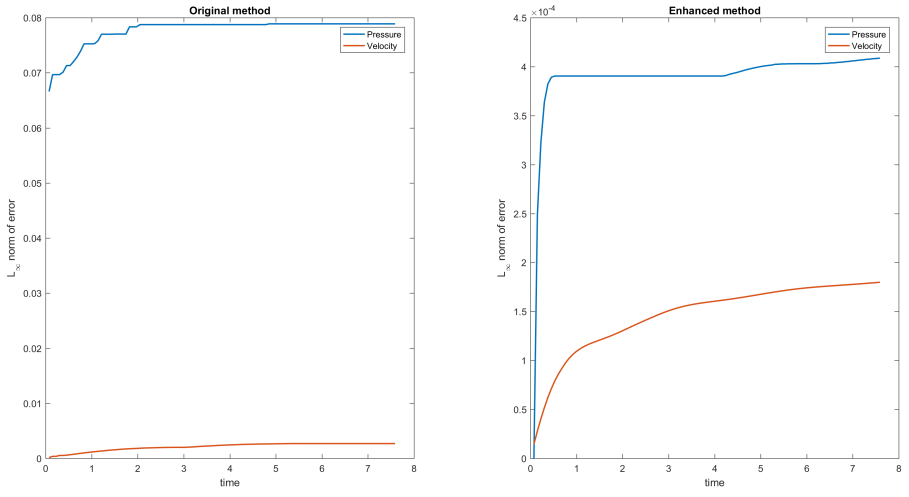


Figure 4.14: Time evolution of the L_∞ norm of error for the two methods

	Original method	Enhanced method
$\|\varepsilon_u\|_\infty$	1.6E-4	1.45E-5
$\|\varepsilon_v\|_\infty$	4.29E-5	4.66E-15
$\|\varepsilon_w\|_\infty$	1.58E-4	1.45E-5
$\|\varepsilon_p\|_\infty$	6.66E-2	3.05E-12
$\|\varepsilon_{\nabla p}\|_\infty$	1.99E-3	9.74E-14

Table 4.1: Comparison of the two methods after one iteration

	Original method	Enhanced method
$\|\varepsilon_u\|_\infty$	2.00E-3	1.80E-4
$\|\varepsilon_v\|_\infty$	1.66E-3	1.36E-6
$\|\varepsilon_w\|_\infty$	2.69E-3	1.80E-4
$\|\varepsilon_p\|_\infty$	7.89E-2	4.09E-4
$\|\varepsilon_{\nabla p}\|_\infty$	2.01E-3	4.09E-5

Table 4.2: Comparison of the two methods after 100 iterations

After the first iteration the pressure error of the two methods increases steeply, then the growth slows up. A similar behaviour is found for the velocity error. It is possible to observe that the error with the enhanced method is roughly reduced of a factor 100 with respect to the standard interpolation scheme.

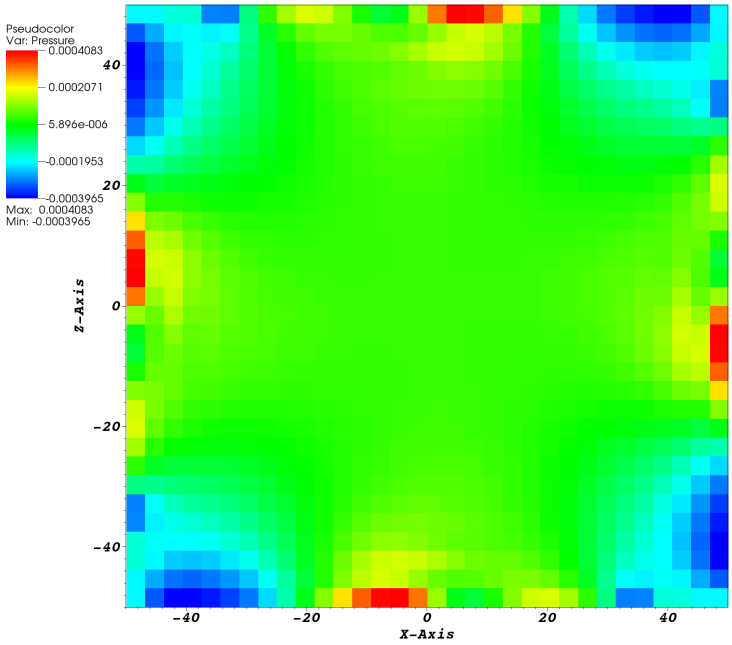


Figure 4.15: Pressure field after 100 iteration, enhanced method

In figure 4.15 and 4.16 the pressure and the norm of the pressure gradient after 100 iterations are represented, it is possible to notice that the error has assumed a less random pattern compared to the first iteration, the pressure perturbations

are propagating from the boundaries towards the centre of the domain, but overall the norm of the error remains acceptable for both the variables.

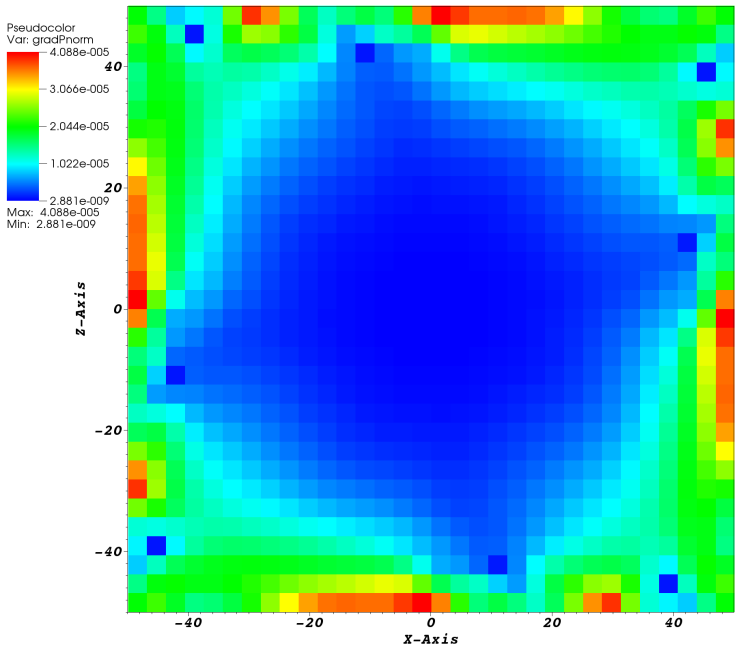


Figure 4.16: Norm of the pressure gradient after 100 iteration, enhanced method

4.4 Preliminary structure simulations

Some preliminary simulations are performed to understand the behaviour of the model in a rotating frame of reference. Firstly, a static test of the structural code is run in order to validate the effect of the centrifugal force. For this simulation it is possible to compare the numerical solution with the analytic one, which can be obtained solving the following equilibrium equation:

$$\frac{d^2u}{dx^2} = -\frac{\rho\Omega^2}{E}x \tag{4.46}$$

where u is the axial displacement and Ω is the rotational speed, the spin softening term has been neglected. The structural parameters are the same fictitious values used in chapter 2:

$$EI_{yy} = EI_{zz} = 10^6 \ Nm^2, L = 22.25 \ m, EA = 10^9 \ N, \rho A = 1 \ Kg/m$$

The clamped-free constraints are:

$$\begin{cases} u(0) = 0 \\ \dfrac{du}{dx}(L) = 0 \end{cases}$$

hence:

$$u = \frac{\rho\Omega^2 L^3}{E}\left(\frac{\xi}{2} - \frac{\xi^3}{6}\right) \tag{4.47}$$

where $\xi = x/L$. The rotational speed Ω has been chosen in order to obtain:

$$u(L) = \frac{\rho\Omega^2 L^3}{3E} = 1 \tag{4.48}$$

therefore, $\Omega = 521.9\ rad/s$.

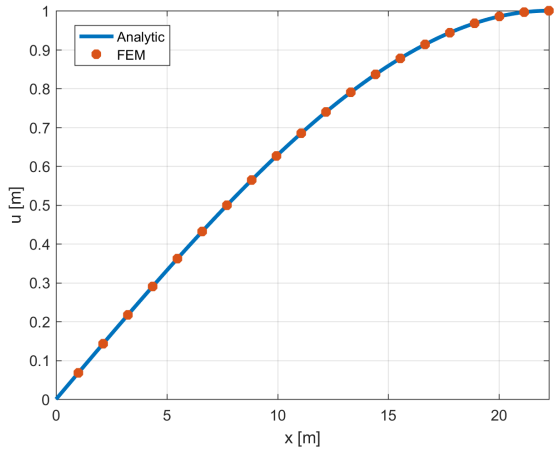


Figure 4.17: Comparison of FEM and analytic results

The number of beam elements employed is 200. In figure 4.17 the comparison between the exact solution and the one obtained with the FEM code is reported, it is possible to observe that the numerical results lie on the analytic curve, thus the method predicts correctly the centrifugal effect. A second static test case has been performed to verify the effect of the spin softening term.

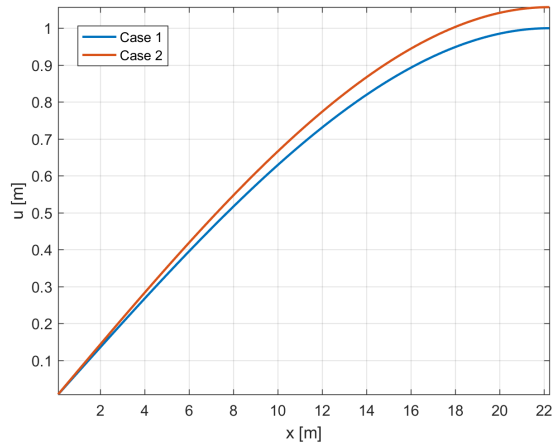


Figure 4.18: Displacements for the two test cases

Figure 4.18 reports the results obtained in the first test case, where the spin softening is neglected, and the second one, where the complete model has been employed. As expected the displacements is slightly higher when the spin softening is considered, it is also possible to observe that the contribution of this term becomes bigger as the distance from the clamped end, which corresponds to the axis of rotation, increases, this is also expected, since the spin softening is proportional to the displacement.
In order to evaluate the effect of the Coriolis force, two dynamic bending tests have been carried on, a constant transverse load has been employed for both of them, $p = 32.64\ N/m$.
Since wind turbines turn at relatively low speeds, a small influence of the Coriolis and centrifugal term is expected. In order to verify this thesis, in the first test case the blade does not rotate, whereas in the second one, a realistic angular velocity is adopted, $\Omega = 2\ rad/s$.

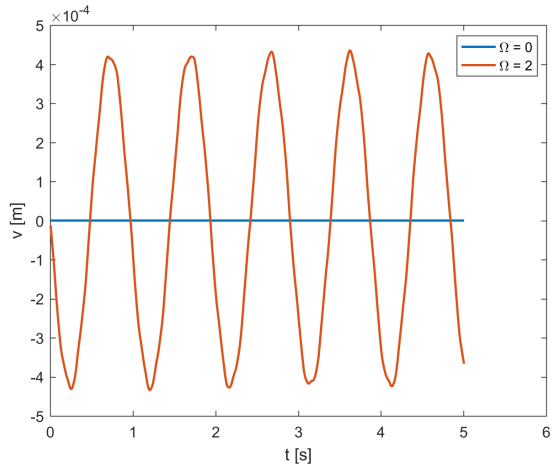


Figure 4.19: y-axis displacement comparison

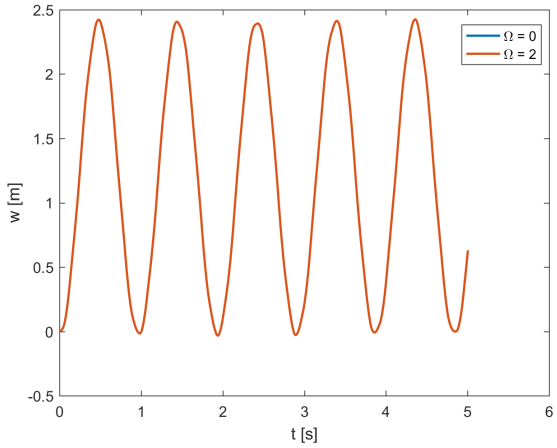


Figure 4.20: z-axis displacement comparison

In figure 4.19 and 4.20 the comparison of the two test cases is shown, it is impossible to distinguish the two curves in figure 4.20, whereas the y-axis displacement presents a sinusoidal response when $\Omega \neq 0$, but for low rotational speeds the contribution of the Coriolis force is quite small, as remarked in figure 4.19.

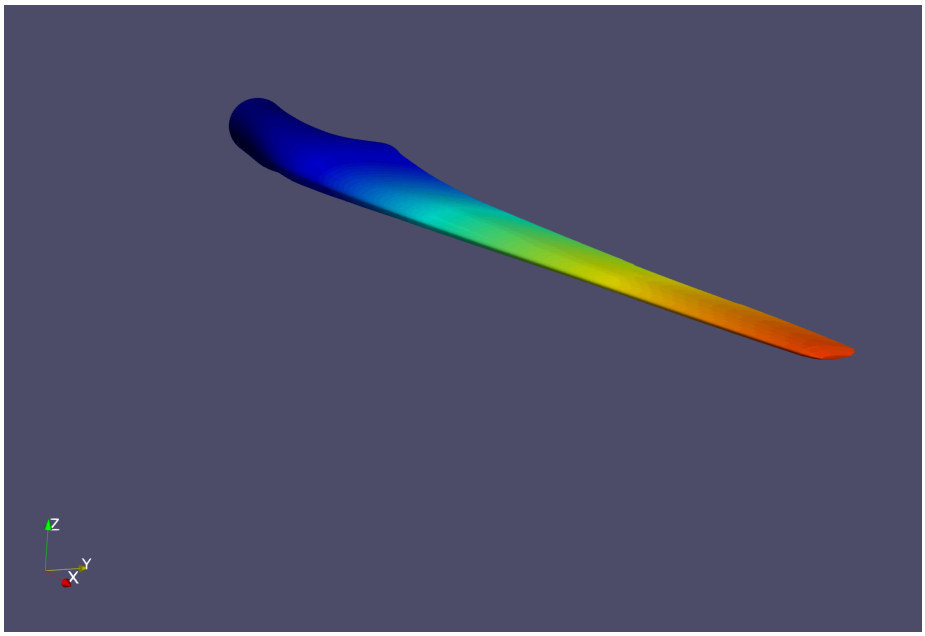


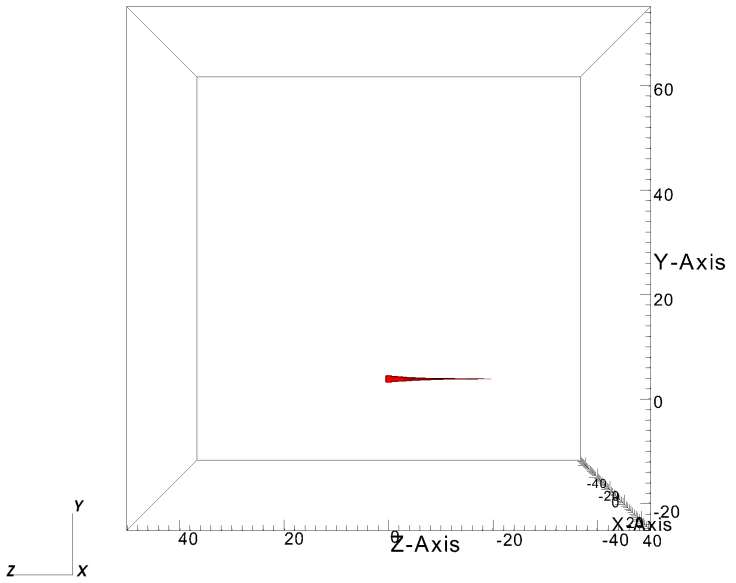
Figure 4.21: Deformed blade structure

4.5 Preliminary Navier-Stokes simulations

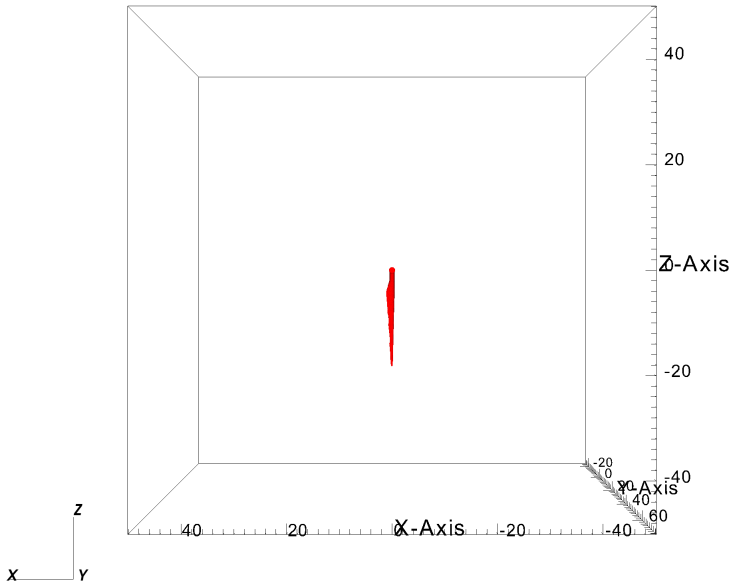
A qualitative simulation has been run using the incompressible Octree Navier-Stokes solver and the blade geometry. The incompressible flow induced by a rotating object can be described by means of two non-dimensional parameters:

- Reynolds number: $Re = \frac{U_\infty L}{\nu}$
- Rossby number: $Ro = \frac{U_\infty}{\Omega L}$

where L is, in the case of a turbine blade, the maximum value of the chord. In order to avoid spurious edge effects, the domain needs to be sufficiently large, enough tip clearance has to be taken into account. The radius of the turbine, which is approximately 23 metres and is the dominant size, imposes a domain which is at least twice as long, but, being limited to a cubic domain, enough space must be considered for the development of the wake, so that the resulting length is roughly four times the blade span. This size should ensure that the influence of the edge effects will be negligible.
A [-50,50]x[-25,75]x[-50,50] domain is employed, the y-axis corresponds to the streamwise direction and also to the axis of rotation, the z-axis is the spanwise direction and the x-axis is roughly parallel to the chord of the blade.



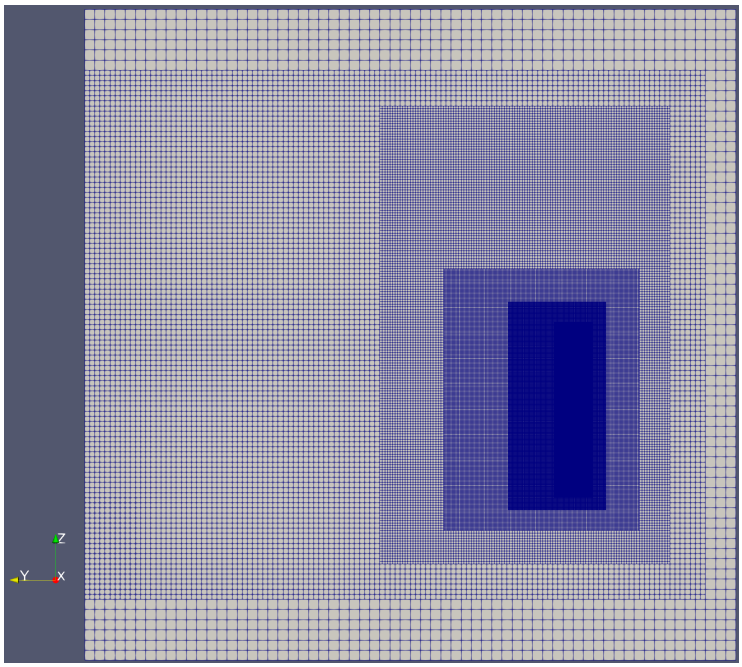
(a) *Domain, view 1*



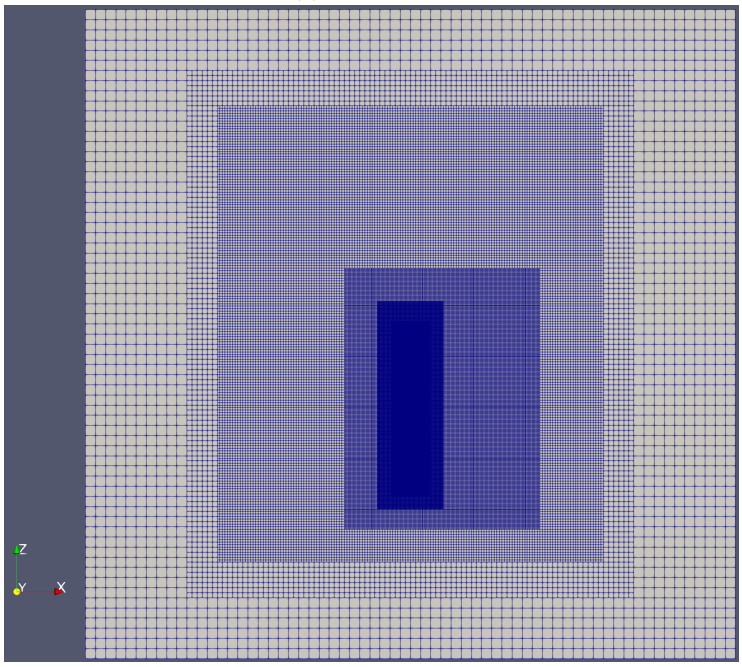
(b) *Domain, view 2*

Figure 4.22: Blade simulation, domain

In order to keep the test as simple as possible, a cartesian-patch mesh is used, the smallest grid size is slightly less than 5 centimetres, and the total number of cells is 20 millions. The mesh has been constructed in order to have the highest possible resolution on the blade and the near wake, taking into account the rotation of the fluid. The huge size of the domain allow to take full advantage of local refinement, having a sufficient spacing between two level-jumps is fundamental in order to preserve the accuracy of the solution.



(a) *Mesh*, $x = 0$



(b) *Mesh*, $y = 0$

Figure 4.23: Blade simulation, mesh

The adimensional parameters for the preliminary simulation are:

- Reynolds number: $Re = 1000$
- Rossby number: $Ro = 3.33$

A normal in service value of the Reynolds number is of the order of 10^6, but the aim of the considered simulation is not the reproduction of the real working condition of the blade, indeed a very low Reynolds number allows to rapidly assess the quality of the rotating frame model without using an exaggerate amount of computational resources, since there is no need for an extremely refined grid.
A short run of 5 non-dimensional times has been performed, using 16 nodes of the PlaFRIM cluster[2], each composed by two 12-cores Intel Haswell CPUs @2.5GHz. This kind of computational nodes are very similar to the one of the more powerful OCCIGEN machine, therefore they allow a straightforward portability of the code from the test cluster to the production one. The computational time of a short run is 4 wall clock hours.

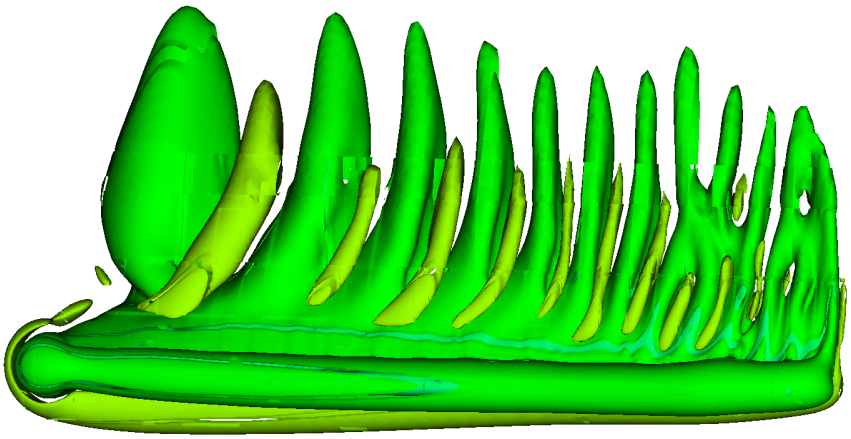
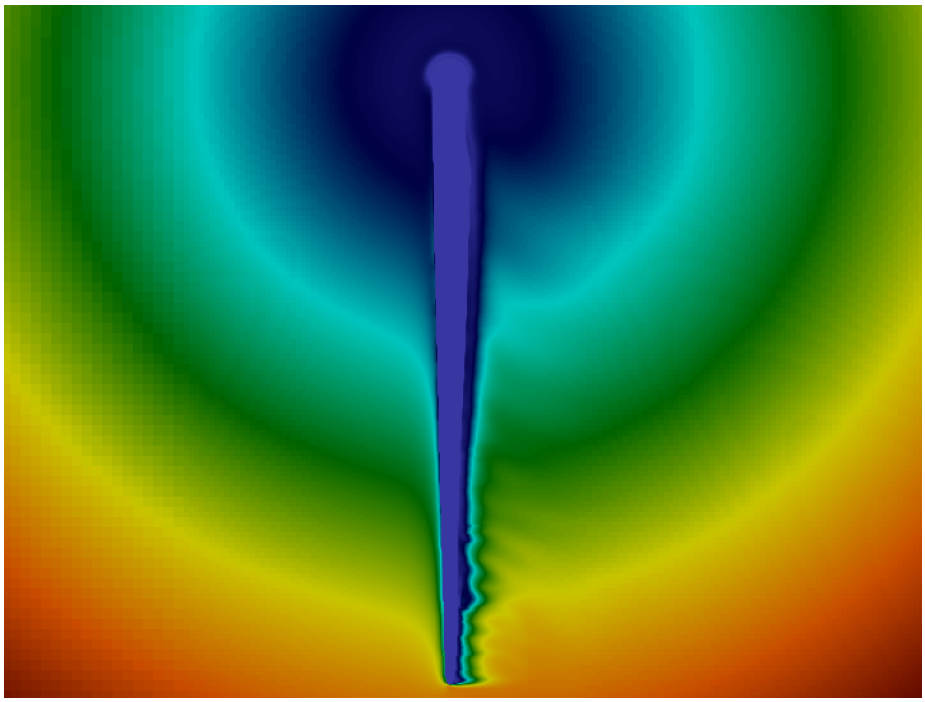


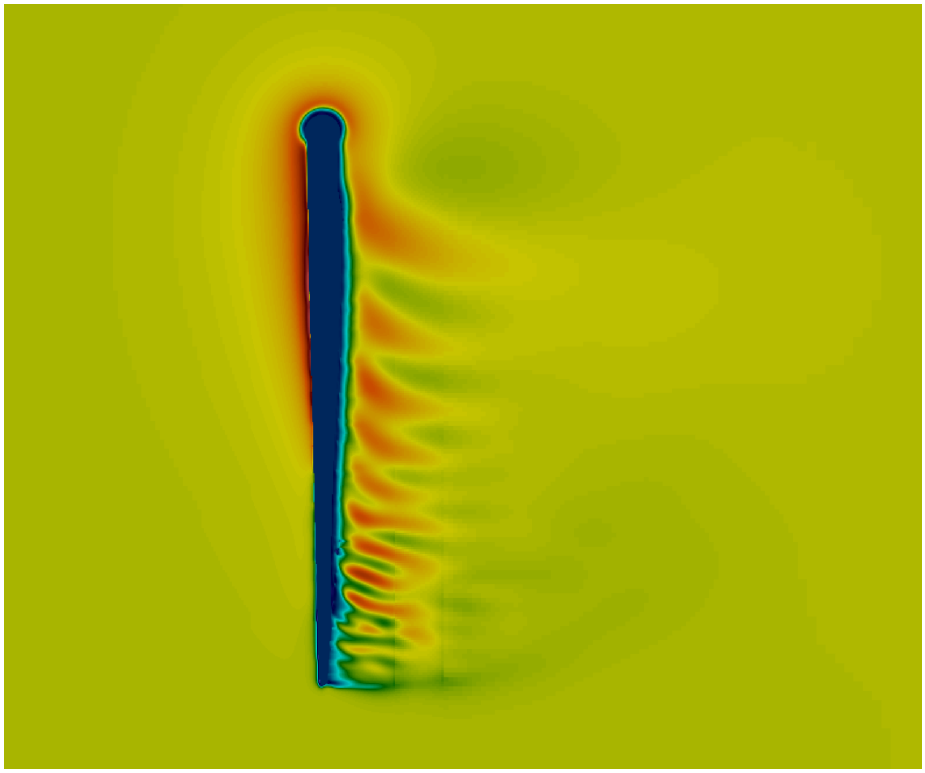
Figure 4.24: Contour surfaces of the y-velocity, 9 levels

In figure 4.24 the 3D contour plot is reported, the wake of the body can be observed, it is clearly visible the rotational effect imposed by the blade. Due to the low Reynolds and the low resolution of the geometry, the flow is severely separated, at higher Reynolds numbers and with a sufficiently fine mesh, the wake should become a thin layer and the turbulent structure should be concentrated in the tip region, where a tip vortex forms due to the pressure differential between the upside and the downside of the blade. The described phenomenon is well represented in figure 4.1.

[2]https://www.plafrim.fr/



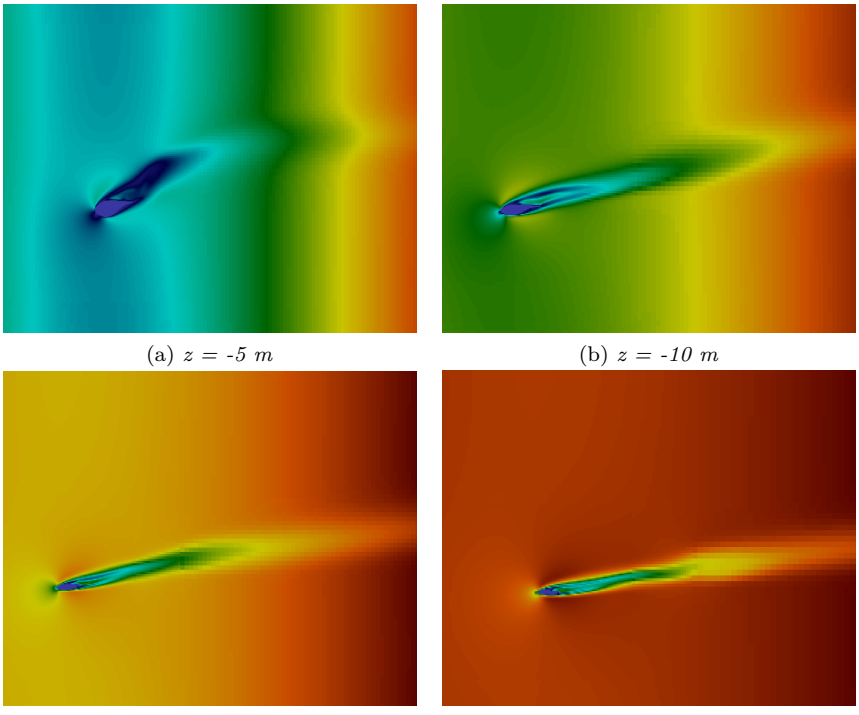
(a) *Relative velocity magnitude*



(b) *y-velocity*

Figure 4.25: 2D plot of velocities, $y = 0$

Since the flow is characterized by a strong three-dimensional character, interpreting the qualitative results of this preliminary simulation is not an easy task. It is useful to plot the velocity field in some planes, such as in figure 4.25, 4.26 and 4.27. Figure 4.25 reports the plots of the relative velocity magnitude and the y-velocity at $y = 0$. Since the y-axis is the direction of rotation, the first plot is dominated by the effect of the rotational speed, the contour lines are almost circular, but they are disturbed by the blade. The y-velocity field is characterized by the wake, observed also with the contour plot in figure 4.24, some inaccurate spots can be observed in correspondence of two level-jumps, which are placed too close to be blade.



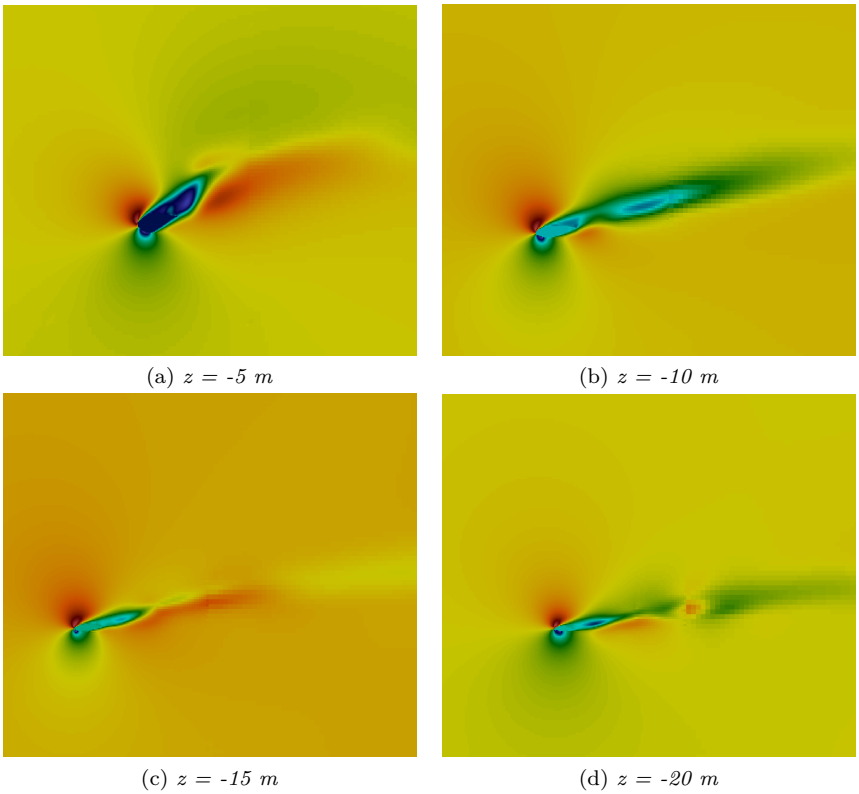
(a) z = *-5 m* (b) z = *-10 m*

(c) z = *-15 m* (d) z = *-20 m*

Figure 4.26: 2D plots, magnitude of relative velocity

Figure 4.26 reports the plots of the magnitude of the relative velocity for different position along the blade span; the most important effect of the rotation of the blade is visible: the tangential speed increases with the distance from the hub and the angle of attack decreases, therefore the fluid remains attached for longer near the tip, whereas the airfoils that are operating close to the hub present a massively separated flow, this is generally true in real world conditions too, even though the effect partially mitigated thanks to the negative twisting of the airfoil.
In figure 4.27 the y-velocity is represented in the same sections of image 4.26 and the same phenomenon can be observed. In this figure the recirculation regions behind the trailing edges of the airfoils are visible, the thickness of the wake decreases significantly as the distance from the hub increases, thus the resolution of the grid becomes less and less satisfying closer to the tip. In order

to capture all the small details a progressive refinement should be employed, but at the same time, introducing level-jumps across the surface of the body might lead to other kind of spurious results.
Overall, the preliminary simulation provided a flow field which is in line with the physics of the problem, the rotational effect is correctly represented. The numerical method needs to be furtherly improved across the level-jumps, and a globally finer mesh has to be employed in order to simulate the turbine real operative conditions, but the results presented in this chapter represents a good starting point.



(a) z = *-5 m*

(b) z = *-10 m*

(c) z = *-15 m*

(d) z = *-20 m*

Figure 4.27: 2D plots, y-velocity

Conclusion

In this thesis a innovative mathematical model has been proposed to deal with fluid-structure interaction problems. The penalized incompressible Navier-Stokes equations are solved using a predictor-corrector scheme, a Wall Modelled Large Eddy Simulation model has been introduced in order to simulate high Reynolds turbulent flows and a classic Euler-Bernoulli finite element formulation has been employed for the structure.
The fluid model has been tested to its limits, thanks to an in-house developed and stable second order cartesian code. Some spurious oscillation have been found when high Reynolds flows are not enough resolved, but it has been shown that refining the mesh these phenomena tend to disappear. The flow past a cylinder test case has been used as a benchmark to evaluate the performance of the model, experimental data found in literature have been compared to the results obtained in the present work, and a generally good accordance was found. The structural model has been validated comparing numerical results with the analytic theory of beams, no issues at all were encountered with the finite element solver, results were perfectly aligned with analytic formulae.
Once the fluid model was validated, its extension to Octree grid has been deeply investigated, in particular, the treatment of level-jumps required special attentions. The possibility of introducing local mesh refinements while keeping a cartesian frame is an attractive feature of Octrees, but on the other hand, level-jumps have revealed to be main sources of instabilities and numerical errors, an accurate and robust discretization at level-jumps is the key of a valid Octree solver. An alternative stencil has been proposed in order to deal with local refinements, the validation has shown that in some special cases the second order can be achieved, but in general, the order of convergence lies between 1 and 2, depending on the number of refinements and their configuration.
Another criticality has been found in the first order semi-Lagrangian scheme: the discretization of the advection equation with this approach is unstable, the method is not Total Variation Diminishing, and the reason has been found in the Radial Basis Function interpolation which does not preserve the maximum and the minimum of the interpolating data. A limiter function has been employed as a stabilization of the numerical scheme. The methodology has been tested on a low Reynolds flow past a sphere, and the results where similar for all the test cases performed. Moreover, the high Reynolds flow past a cylinder has been retested using the Octree solver, the simulation was slightly underresolved, but overall results were in line with results found in the scientific literature.
In the last chapter the modelling of a wind turbine blade has been discussed, the geometry has been discretized using a structured grid of Lagrangian markers, in order to make it compatible with the Octree solver. A non-inertial coordi-

nate system has been employed, thus the centrifugal and Coriolis effect had to be added to both the Navier-Stokes solver and the FEM model. The rotating frame of reference has been tested on both the structure of the blade and the fluid simulation, the results obtained in the preliminary simulations are in line with the physics of the problem.
There are still some minor adjustment to be done, but the present work helped in the detection of some critical aspects related to the model that has been proposed, in particular with the implementation of the method itself on Octree grids. Future developments will start from the solid basis estabilished during the validation phase, which certified that the proposed model is able to predict with accuracy challenging high Reynolds flows, which is an excellent result for the immersed boundary method.

Bibliography

[1] R. Mittal, H. Dong, M. Bozkurttas, F.M. Najjar, A. Vargas, A. von Loebbecke, *A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries*, J. Comput. Phys. 227 (2008) 4825-4852

[2] M. Bergmann, J. Hovnanian, A. Iollo, *An accurate cartesian method for incompressible flows with moving boundaries*, Commun. Comput. Phys. 15 (2014) 1266-1290

[3] P. Angot, C.H. Bruneau, P. Fabrie, *A penalization method to take into account obstacles in incompresible viscous flows*, Numer. Math. 81 (1999) 497-520

[4] O. Ceyhan, *Towards 20MW Wind Turbine: High Reynolds Number Effects on Rotor Design*, 50$^{\text{th}}$ AIAA ASM Conference, Nashville, Tennessee, USA 9-12 January 2012

[5] J. Smagorinksy, *General circulation experiments with the primitive equations: 1. The basic experiment*, Monthly Weather Review 91 (1963) 99-164

[6] M. Germano, U. Piomelli, P. Moin, W.H. Cabot, *A dynamic subgrid-scale eddy viscosity model*, Phys. Fluids 3 (1991) 1760-1765

[7] A.W. Vreman, *An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications*, Phys. Fluids 16 (2004) 3670-3681

[8] H. Choi, P. Moin, *Grid-point requirements for large eddy simulation: Chapman's estimates revisited*, Phys. Fluids 24 (2012)

[9] D.B. Spalding, *A single formula for the "Law of the Wall"*, J. Appl. Mech. 28 (1961) 455-458

[10] I. Abalakin, B. Koobus, *The behavior of two near-wall models for k-ε prediction of stall*, INRIA, Technical Report (2000), unpublished

[11] M.D. De Tullio, *Development of an immersed boundary method for the solution of the preconditioned Navier-Stokes equations*, Doctoral dissertation, Politecnico di Bari (2006)

[12] R. de Frias Lopez, *A 3D finite beam element for the modelling of composite wind turbine wings*, Master Thesis, KTH (2013)

[13] N.M. Newmark, *A method of computation for structural dynamics*, Journal of the Engineering Mechanics Division, 85 (1959) 67-94

[14] G. Hou, J. Wang, A. Layton, *Numerical methods for fluid-structure interaction - A review*, Commun. Comput. Phys. 12 (2012) 337-377

[15] L.M. Lourenco, C. Shih,(1993). *Characteristics of the plane turbulent near wake of a circular cylinder, a particle image velocimetry study*, (1993) published in [16]

[16] P. Beaudan, *Numerical experiments on the flow past a circular at a subcritical Reynolds number*, Doctoral disseration, Stanford University (1994)

[17] L. Ong, J. Wallace, *The velocity field of the turbulent very near wake of a circular cylinder*, Experiments in Fluids 20 (1996) 441-453

[18] B. Cantwell, D. Coles, *An experimental study of entrainment and transport in the turbulent near wake of a circular cylinder*, J. Fluid Mech. 136 (1983) 321-374

[19] P. Parnaudeau, J. Carlier, D. Heitz, E. Lamballais, *Experimental and numerical studies of the flow over a circular cylinder at Reynolds number 3900* Phys. Fluids 20 (2008) 1-14

[20] A. Chorin, *Numerical Solution of the Navier Stokes equation*, Math. Comput. 22 (1968) 746-762

[21] Y. Saad, M.H. Schultz, *GMRES: A Generalized Minimal Residual algorithm for solving nonsymmetric linear systems* J. Sci. and Stat. Comput. 7 (1986) 856-869

[22] M. Breuer. *Large eddy simulation of the subcritical flow past a circular cylinder: numerical and modeling aspects*, Int. J. Numer. Meth. Fluids 28 (1998) 1281-1302

[23] S.E. Kim, *Large eddy simulation of turbulent flow past a circular cylinder in subcritical regime*, AIAA Paper 1418 (2006)

[24] A. Travin, M. Shur, M. Strelets, P. Spalart, *Detached-eddy simulation past a circular cylinder*, Flow Turbul. Combust. 63 (1999) 293-313

[25] D. Luo, C. Yan, H. Liu, R. Zhao, *Comparative assessment of PANS and DES for simulation of flow past a circular Cylinder*, J. Wind Eng. Ind. Aerodyn. 134 (2014) 65-77

[26] A. Raeli, *Solution of the variable coefficient Poisson equation on cartesian hierarchical meshes in parallel: application to phase changing materials*, Doctoral dissertation, Université de Bordeaux (2017)

[27] G.M. Morton, *A computed oriented geodetic data base and a new technique in file sequencing*, IBM Ltd., technical report (1966)

[28] H.A. van der Vorst, *Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, J. Sci. Stat. Comp. 13 (1992) 631-644

[29] Y. Coudière, *Analyse de schemas volumes finis sur maillages non structures pour des problemes lineaires hyperboliques et elliptiques*, Doctoral dissertation, Université de Toulouse 3 (1999)

[30] M. Bergmann, A. Ferrero, A. Iollo, C. Taymans, *D2.7: Report on investigation of predicted non-linear behaviour of wind turbine blades gust behaviour*, INRIA, Technical report (2016), unpublished

[31] M.A. Olshanskii, K.M. Terekhov, Y.V. Vassilevski, *An octree-based solver for the incompressible Navier-Stokes equations with enhanced stability and low dissipation*, Computer & Fluids 84 (2013) 231-246

[32] M. Falcone, R. Ferretti, *Convergence analysis for a class of high-order semi-Lagrangian advection schemes*, SIAM J. Numer. Anal. 35 (1998) 909-940

[33] P.K. Sweby, *High resolution schemes using flux limiters for hyperbolic conservation laws*, SIAM J. Numer. Anal. 21 (1984) 995-1011

[34] R. Campregher, J. Militzer, S.S. Mansur, A.S. Neto, *Computation of the flow past a still sphere at moderate Reynolds numbers using an immersed boundary method*, J. Braz. Soc. Mech. Sci. & Eng. 31 (2009) 344-352

[35] B. Fornberg, *Steady viscous flow past a sphere at high Reynolds numbers*, J. Fluid Mech. 190 (1988) 471-489

[36] E.A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, *Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations*, J. Comput. Phys. 161 (2000) 35-60

[37] P. Chatelain, L. Bricteux, S. Backaert, G. Winckelmans, S. Kern, P. Koumoutsakos, *Vortex particle-mesh methods with immersed lifting lines applied to the Large Eddy Simulation of wind turbine wakes*, Wake Conference, 2011, Gotland

Acknowledgements

Firstly, I would like to thank professor Angelo Iollo for the opportunity he gave me and for his wise supervision, working under his direction to the AeroGust project at INRIA in Bordeaux has been an incredible experience. Special thanks to the whole MEMPHIS team, that made me feel home during the last six months, in particular to Andrea Ferrero, whose precious advices helped me to overcome many difficulties, Claire Morel, who worked alongside me to the AeroGust project and to my officemate Antoine Fondaneche.
Thanks to all my colleagues of the Bordeaux Institute of Mathematics, who were like a second family, and to my friends from the Erasmus group, for all the joyful moments we shared in our free time. Much appreciation goes to my closest friends in Italy, Emanuele Sanguineti, Alessandro Quarta and Walter Finocchiaro, for keeping in touch with me even when the distance was considerable, and to my fellow expats and classmates, Daniele Esposito and Mattia Bassignana, for the beautiful experiences spent together in the last two years.
Last, but not the least, I would like to thank my parents, without their continuous support, this would not have been possible.