



HAL
open science

CFD simulation with anisotropic mesh adaptation : application to floating offshore wind turbines

Louis Douteau

► **To cite this version:**

Louis Douteau. CFD simulation with anisotropic mesh adaptation : application to floating offshore wind turbines. Fluids mechanics [physics.class-ph]. École centrale de Nantes, 2020. English. NNT : 2020ECDN0003 . tel-02972319

HAL Id: tel-02972319

<https://theses.hal.science/tel-02972319v1>

Submitted on 20 Oct 2020

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

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

THESE DE DOCTORAT

ÉCOLE CENTRALE DE NANTES

ÉCOLE DOCTORALE N° 602
Sciences pour l'Ingénieur
Spécialité : *Mécanique des milieux fluides*

Par

Louis DOUTEAU

CFD simulation with anisotropic mesh adaptation: application to floating offshore wind turbines.

Thèse présentée et soutenue à Nantes, le 5 mars 2020
Unité de recherche : Institut de Calcul Intensif

Rapporteurs avant soutenance :

Elie Hachem
Grégory Pinon

Professeur, Mines ParisTech
Maître de conférences HDR, Université du Havre

Composition du Jury :

Président : Thierry Coupez
Examineurs : Sandrine Aubrun
Thierry Coupez
Jean-Christophe Gilloteaux
Elie Hachem
Grégory Pinon
Dir. de thèse : Luisa Rocha da Silva

Professeur des universités, Mines ParisTech, Sophia Antipolis
Professeur des universités, Ecole Centrale de Nantes
Professeur des universités, Mines ParisTech, Sophia Antipolis
Ingénieur de Recherche, Ecole Centrale de Nantes
Professeur de l'école des Mines, Mines ParisTech, Sophia Antipolis
Maître de conférences HDR, Université du Havre
Chargée de recherche HDR, Ecole Centrale de Nantes

Résumé

Cette thèse réalisée à l'Institut de Calcul Intensif (ICI) de Centrale Nantes, en collaboration avec le Laboratoire de recherche en Hydrodynamique, Énergétique et Environnement Atmosphérique (LHEEA), est dédiée à la simulation numérique d'éoliennes flottantes. Elle a été financée par WEAMEC (West Atlantic Marine Energy Community), et s'intègre dans la stratégie de recherche et développement de la région Pays de la Loire en énergies marines renouvelables.

L'objectif de cette thèse a été le développement d'un simulateur d'éoliennes flottantes utilisant des méthodes d'une haute précision. Cette approche est complémentaire des technologies actuellement utilisées dans l'industrie pour la simulation d'éoliennes flottantes. Celles-ci sont bien plus efficaces grâce aux approximations importantes réalisées dans la modélisation de la physique. Les coûts de calcul engendrés par l'approche nouvellement développée sont rédhibitoires dans le cadre d'une utilisation industrielle, mais offrent d'autres perspectives. La simulation haute performance permet une description fine de la physique des phénomènes fluides apparaissant autour des éoliennes flottantes. La validation du design des structures peut être effectuée à l'aide de tels outils. Les approches fines permettent également une calibration des méthodes de simulation industrielles, qui intègrent des termes définis empiriquement. Ces enjeux sont présentés dans le Chap. [1](#).

La thèse a eu pour objectif le développement d'un simulateur d'éoliennes flottantes à partir de la bibliothèque logicielle ICI-tech, dédiée à la simulation d'écoulements avec des approches haute-performance. Une résolution des équations de Navier-Stokes dans un formalisme Variational MultiScale est effectuée pour obtenir une description fine des écoulements fluides. Un maillage de calcul unique est généré grâce à un processus d'adaptation de maillage automatique, les différentes phases du domaine de calcul étant représentées à l'aide de fonctions de phase de type "level-set". La résolution des équations de Navier-Stokes est réalisée par éléments finis, dans une formulation monolithique. Un schéma de convection-réinitialisation est utilisé pour transporter les différentes phases fluides dans le domaine de calcul. Le maillage automatiquement généré est anisotrope, ce qui permet de limiter le nombre de points dans le maillage de calcul, tout en conservant une bonne représentation des géométries simulées. Les coûts de calcul engendrés restent malgré tout importants, et un formalisme hautement parallélisé permet de les distribuer, via un déploiement sur des supercalculateurs.

La méthodologie introduite peut être séparée en deux étapes principales. En premier lieu, la représentation des différentes géométries dans le domaine de calcul est nécessaire pour l'établissement d'écoulements précis. Le paradigme de frontières immergées est détaillé dans le Chap. [2](#) à partir des fonctions de phase. L'optimisation de cette méthode, effectuée à l'aide d'une structure "octree", est présentée en détail. La dépendance de cette définition vis-à-vis du maillage de calcul est discutée, et la méthode d'adaptation de maillage *a posteriori*, automatique et anisotrope est introduite. La gestion du parallélisme est présentée ensuite, avant une application à des cas d'éoliennes flottantes. Les résultats d'immersion considèrent la précision de la reconstruction et l'efficacité des

méthodes implémentées. Des tests de scalabilité ont été réalisés, ainsi qu’une caractérisation des résultats d’immersion en fonction des principaux paramètres définissant la reconstruction. Les performances parallèles de l’immersion sont satisfaisantes.

La résolution des écoulements est la deuxième grande phase des calculs, qui peut être effectuée à partir d’un maillage initial correctement défini. La méthodologie suivie est détaillée dans le Chap. 3. Après une introduction présentant les enjeux des équations de Navier-Stokes, le paradigme éléments-finis pour leur résolution est exposé, et la formulation Variational MultiScale incompressible en vitesse et pression est décrite. Afin de déduire les écoulements, la viscosité et la masse volumique du fluide doivent être connues en chaque point du maillage de calcul. L’établissement de lois de mélange est détaillé, ainsi que le transport des phases. La description de la méthodologie suivie pour obtenir les écoulements se termine avec des détails sur le processus de résolution en lui-même. Une deuxième section de ce chapitre se concentre sur le développement d’outils numériques dédiés à la simulation d’éoliennes flottantes. Un dispositif de calcul de forces *a posteriori* est défini, et un bassin de houle numérique est mis en place. Pour finir, un formalisme d’interaction fluide/structure implémenté dans la cadre de la thèse est détaillé.

ICI-tech a été utilisé dans d’autres travaux pour simuler des écoulements visqueux. À l’inverse, le développement d’un simulateur d’éoliennes flottantes implique des flux aérodynamiques et hydrodynamiques plus turbulents. Les comportements des flux dans ces circonstances sont beaucoup plus erratiques, et engendrent une complexité plus importante dans les simulations. Des étapes de vérification et validation ont été nécessaires, et sont exposées dans le Chap. 4. Durant une première phase, les écoulements autour de profils ont été étudiés, afin notamment de vérifier et valider l’aérodynamique. Les cas tests utilisés ont été définis dans cet objectif, avec des écoulements de plus en plus turbulents et des profils représentatifs de pales d’éoliennes. Une deuxième phase a été dédiée à la vérification de la propagation de champs de houle. L’influence du maillage anisotrope a été particulièrement étudiée. Une dernière phase de validation s’est focalisée sur le formalisme d’interaction fluide/structure. Les résultats ont été globalement décevants, comme le montre la liste non exhaustive suivante. Les forces aérodynamiques évaluées sur les profils ont été surestimées, tandis que la dissipation numérique a été importante lors de la propagation de champs de vagues. L’adaptation de maillage anisotrope a notamment eu un impact important sur la qualité des résultats. Les tests de flottaison destinés à valider l’interaction fluide/structure ont aussi abouti sur des différences notables avec les mesures expérimentales, mais proches de résultats obtenus avec des méthodes similaires. Différentes pistes sont proposées en vue d’améliorer la qualité des simulations, se concentrant sur l’adaptation de maillage ou l’ordre d’interpolation des éléments finis.

Les conclusions principales des différents chapitres ont été résumées dans le Chap. 5. Des résultats préliminaires ont présenté les enjeux du développement d’un simulateur d’éoliennes flottantes. Dans ce cadre, des ébauches de cas 3D ou de champs de vagues irréguliers ont été simulés. À partir des résultats du chapitre précédent, les limitations actuelles du solveur ont été discutées. Des perspectives à court, moyen et long terme pour le développement d’un simulateur plus précis ont été proposées.

Remerciements

La réalisation de cette thèse a été possible grâce aux contributions de beaucoup de personnes, qu'il convient de remercier !

Dans un premier temps, je tiens à remercier les membres du jury, qui ont accepté d'évaluer mes travaux. Merci à Elie Hachem et Grégory Pinon d'avoir accepté la charge de rapporteur, et d'avoir apporté de nombreux commentaires constructifs sur beaucoup de points différents. Merci à Sandrine Aubrun, pour avoir examiné régulièrement mon travail, et à Thierry Coupez, pour avoir présidé le jury.

Je tiens à remercier grandement Luisa Rocha da Silva et Jean-Christophe Gilloteaux pour m'avoir permis d'effectuer ce doctorat. Leur encadrement a été extrêmement précieux, à la fois précieux et confortable. Il a contribué à la mise en place d'un environnement de travail en relative autonomie qui m'a grandement bénéficié au cours de ces trois ans.

Je tiens également à remercier chaleureusement toutes les personnes avec qui j'ai partagé ces trois années de travail acharné, mais aussi de bonne rigolade. Merci à tous les membres de l'ICI que j'ai côtoyé au quotidien, avec qui les échanges ont toujours été très enrichissants. En particulier, merci à Simon, le tonton, à Nesrine, avec qui on a partagé toutes les péripéties de l'ICI, à Hugues, qui mérite une statue, et à Manu, toujours dispo pour un bon débat.

Je veux également créditer tous ces gens avec qui on a fait la fête au cours de ces 3 ans. Petite pensée émue à Alex le coloc, parti trop tôt vers d'autres univers ensoleillés. Aux collègues doctorants Nathan/Marie/Thomas/et al, avec qui on su profiter des bons moments au quotidien ou en conf. Aux anciens colocs, même si rien n'est fini. Aux amis de Centrale à Nantes. Aux copains du basket d'ici ou d'ailleurs. À tous ceux qui ont connu la goutte à papi. Et aux autres oubliés sûrement, et qui pensent qu'ils auraient dû se trouver dans cette liste non exhaustive. Cette formulation est empruntée à Nicolas, toujours de bon conseil, et que j'avais oublié de remercier.

Un immense merci doit aller à toute ma famille, pour m'avoir toujours soutenu dans tout ce que j'ai entrepris et pour m'avoir mis sur des bons rails. Merci notamment au chat, que je devais citer le jour de la soutenance, mais qui s'est éclipsé.

Pour finir, bon courage dans la lecture de ce manuscrit à tout ceux qui ont préféré commencer par lire ces remerciements, par pure procrastination probablement.

Contents

1 Introduction	1
1.1 State of the art on floating offshore wind turbines	2
1.1.1 Why floating offshore wind turbines ?	2
1.1.2 Horizontal-axis wind turbines	5
1.1.3 Floating offshore wind turbines	9
1.2 Numerical simulation of FOWTs	12
1.2.1 Aerodynamic of wind turbines	12
1.2.2 Hydrodynamics	17
1.2.3 Existing numerical simulation tools for FOWTs	20
1.3 Aim and scope of this thesis	24
2 Implicit boundary representation	25
2.1 Reconstruction and level-set method	26
2.1.1 Theory of the mesh immersion using level-set functions	26
2.1.2 Immersion of objects represented by a set of elements	29
2.1.3 Tree-optimization of the reconstruction procedure	31
2.2 Mesh adaptation	35
2.2.1 Anisotropic mesh adaptation with a metric field	36
2.2.2 Metric construction using <i>a posteriori</i> error estimator	38
2.2.3 Mesh adaptation in practice	40
2.3 Highly-parallelized framework in ICI-tech	41
2.3.1 Parallel meshing in ICI-tech	41
2.3.2 Parallel reconstruction in ICI-tech	42
2.3.3 Mesh adaptation in the highly parallelized context of ICI-tech	42
2.4 Representation of WTs in ICI-tech	43
2.4.1 Reconstruction of a single WT	43
2.4.2 Reconstruction of a large number of wind turbines	48
2.4.3 Scalability of the immersion of FOWTs	50
2.5 Conclusion on the reconstruction of FOWTs	51
3 Numerical framework for the resolution of the flows around FOWTs	53
3.1 Generalities on the resolution of the incompressible Navier-Stokes problem	54
3.1.1 Modeling the flow with the Navier-Stokes equations	55
3.1.2 Resolving numerically the flow from the Navier-Stokes equations	58
3.1.3 Navier-Stokes equations for high-resolution simulations of FOWTs	63
3.2 Monolithic finite element formulation for the incompressible Navier-Stokes equations	65
3.2.1 Finite Element formulation	65
3.2.2 Variational MultiScale of the Navier-Stokes equations	67
3.2.3 Mixing laws	69
3.2.4 Movements of phases	73
3.2.5 Monolithic resolution of the Navier-Stokes equations	74
3.3 Numerical utilities of ICI-tech for the simulation of FOWTs	75

3.3.1	A posteriori computation of forces applied on an immersed body	76
3.3.2	Numerical wave tank	78
3.3.3	Buoyancy and Fluid-Structure Interaction	84
3.4	Conclusion of the numerical framework	86
4	Verification and validation towards the simulation of operating FOWTs	87
4.1	Monophasic verification and validation – Towards high-Reynolds flows	88
4.1.1	Low-Reynolds steady flow around a confined cylinder	88
4.1.2	Moderate-Reynolds flows around NACA profiles	93
4.1.3	High-Reynolds, steady flows around a section of WT blade	99
4.2	Hydrodynamic verification – Generation and propagation of monochromatic wave fields	102
4.2.1	Verification of the solver using constant, isotropic meshes	103
4.2.2	Propagation of monochromatic wave fields using automatic, anisotropic mesh adaptation	108
4.2.3	No-flow tests	116
4.3	Verification and validation of the FSI and application to buoyancy	118
4.3.1	Verification of the buoyancy force applied on a cylinder	118
4.3.2	Validation of the FSI solver for driven motions of floating structures	120
4.4	Conclusion on the numerical results	126
5	Conclusion and perspectives	128
5.1	Conclusion	129
5.1.1	Representation of the geometries	129
5.1.2	Solver verification and validation	129
5.1.3	Influence of the anisotropic meshing on the propagation of waves	130
5.2	First developments towards a demonstrator of FOWTs	130
5.2.1	Rotation of rigid WT blades	130
5.2.2	Preliminary works on IEAWind Task 29	131
5.2.3	Simplified FOWT demonstrator	131
5.2.4	Towards a representation of real sea states	132
5.3	Perspectives	135
5.3.1	Higher-order FEM	135
5.3.2	Mesh adaptation for the propagation of waves	136
5.3.3	Mid- to long-term perspectives for an accurate FOWT simulator	137
	Bibliography	139

Chapter 1

Introduction

Contents

1.1 State of the art on floating offshore wind turbines	2
1.1.1 Why floating offshore wind turbines ?	2
1.1.2 Horizontal-axis wind turbines	5
1.1.3 Floating offshore wind turbines	9
1.2 Numerical simulation of FOWTs	12
1.2.1 Aerodynamic of wind turbines	12
1.2.2 Hydrodynamics	17
1.2.3 Existing numerical simulation tools for FOWTs	20
1.3 Aim and scope of this thesis	24

This thesis was realized within the scope of the EOS project, funded by the West Atlantic Marine Energy Community (WEAMEC). The objective of WEAMEC is to federate the Marine Renewable Energy (MRE) ecosystem of “Pays de la Loire” French Region, in the fields of research, innovation and training. More than 60 industrialists and 30 research institutions are now involved in the development of MRE at the regional scale. Around 45 projects are supported, and several PhDs have already been defended. All the themes related to MREs are considered, ranging from offshore law and techno-economics, to topics focusing on the systems operating offshore. All the technologies of MREs are regarded: tidal and wave energy, Ocean Thermal Energy Conversion, and offshore wind energy, either bottom-fixed or floating.

The EOS project associates the High Performance Computing Institute (ICI) of Centrale Nantes and the Research Laboratory in Hydrodynamics, Energetics and Atmospheric Environment (LHEEA) of Centrale Nantes. It is dedicated to the development of a high-fidelity numerical model for the simulation of Floating Offshore Wind Turbines (FOWTs). The software platform ICI-tech, developed at ICI, initially in the context of viscous flows, is used in this project. The works realized during the PhD and detailed in this thesis consisted in different developments, verification and validation steps aiming at adapting the software platform to the simulation of FOWTs. This Chap. provides a context and introduces the issues raised by this new paradigm. FOWTs are presented in Sec. [1.1](#), which details its main components. A review of the literature concerning the simulation of FOWTs is realized in Sec. [1.2](#). The scope of this thesis and a comprehensive description of its outcomes are then detailed in Sec. [1.3](#).

1.1 State of the art on floating offshore wind turbines

1.1.1 Why floating offshore wind turbines ?

If wind energy has been exploited by humanity for a millennium, its development on an industrial scale only started at the end of the 20th century. In the literature, two main topologies prevailed: Horizontal-Axis WTs (HAWTs) and Vertical-Axis WTs (VAWTs), presented in Fig. [1.1](#). Nevertheless, only HAWTs have been deployed on an industrial-scale, thanks to a better power absorption performance as well as fewer vibration issues. Over the last 20 years, the installed capacity of WTs has been multiplied by 40 (17.4 GW in 2000, 700 GW in 2019). The evolution of the size of WTs is presented in [Yaramasu et al. \(2015\)](#). The rotor diameter has almost tripled (80 m in 2000 for 2 MW HAWTs, 220m for the Haliade-X 12MW WT announced by [GeneralElectric \(2019\)](#)), while the turbine capacity was multiplied by six, from 2 MW to 12MW. In this thesis, the focus is placed on horizontal-axis WTs.

The number of WT Megawatts installed each year in Europe has progressively increased since 2000 according to [WindEurope \(2018a\)](#), and wind energy now provides 18% of the energy production capacities in Europe. Onshore WTs have been operating for several decades, and thus represent about 90% of the whole wind capacity. Technology research WTs has essentially focused over the last decades on an optimization of the design, especially for blades, and rely on a viable industrial base. Experience in installation, maintenance or connection to the electrical grid contributed to bringing wind energy to a leading spot among renewable energies in Europe.

However, the production capacity of onshore WTs is limited by other factors. First of all, onshore wind is rather turbulent, which decreases the production capacity and can lead to more fatigue on blades. The impact of the turbines on their environment is also a strong limitation, as larger rotors generate heavier nuisances that can interact with wildlife or human activities. The intermittency of wind energy production is another

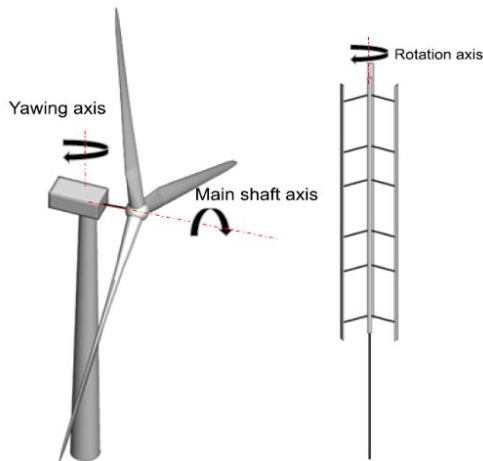


Figure 1.1: Horizontal-Axis Wind Turbine (left) and Vertical-Axis Wind Turbine (right), from [Bazilevs et al. \(2015\)](#).

issue often raised by opponents, as both too weak or too strong winds lead to a stop in production. Fig. [1.2](#) presents the wind energy capacity factor, the ratio between the electrical power produced and the total wind power capacity. This factor for onshore wind is of 22%, which means that, on average, the electricity produced corresponds to about a fourth of the installed onshore wind capacity. Due to poor wind quality preventing the turbines from operating at their nominal power conversion speed, this factor may be slightly increased through modernization of ancient blades but will remain rather low.

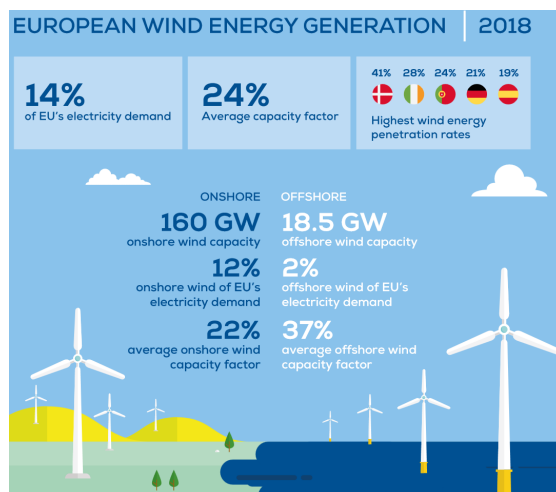


Figure 1.2: Statistics about wind energy in Europe in 2018, from [WindEurope \(2019\)](#).

For all those reasons, wind farms have been installed at sea since the early 2000s. The first farms used bottom-fixed foundations, directly installed on the seabed, and today all the commercial offshore wind farms use this concept in Europe. Offshore winds are stronger, more constant and less turbulent, as illustrated by Fig. [1.3](#). Fig. [1.2](#) shows that the better winds found offshore enable to improve drastically the capacity factor over the onshore wind, at 37%, while the turbulence is usually weaker offshore than onshore. Moreover, even if the design of the turbines is similar in offshore or onshore situations, [WindEurope \(2018b\)](#) noted in 2018 that 6MW-turbines are commonly chosen for offshore wind farms installation, while 2MW-WTs are installed onshore. Given the capacity factors and the nominal power of turbines installed, a rapid computation shows that, in 2017, the average production of 10 offshore turbines was equivalent to the one of 32 onshore turbines. As the actual trend is to transition to even bigger offshore turbines,

this gap may grow very rapidly.

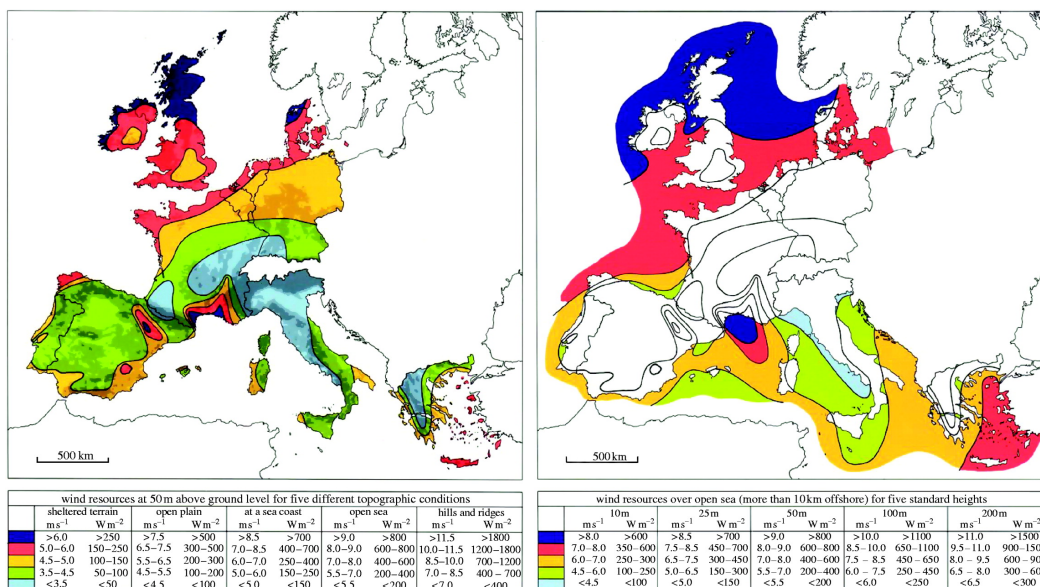


Figure 1.3: Wind resource in Western Europe, from European Wind Atlas of Troen and Petersen (1989).

In 2018 the contribution of offshore wind to European energy production was essentially provided by the bottom-fixed industry. The major part of the commercial farms currently installed is located in Northern Europe, in the North Sea or offshore Denmark. The nature of the foundations used for bottom-fixed WTs, hammered or lying on the seabed, are well suited for the relatively small depths found in those regions. However, the loads applied to the structures, as well as the complexity of the installation procedure, reduce the competitiveness of bottom-fixed offshore WTs when the water depth is superior to 60 meters, as stated in James and Ros (2015). Fig. 1.4 presents the bathymetry in Western Europe, and must be regarded in combination with the wind resource map depicted in Fig. 1.3. Very few sites feature the combination of strong winds and shallow waters, which limits the potential of fixed offshore wind. As an example, the strong winds found on the Mediterranean coast of France cannot be exploited with bottom-fixed WTs. Other, more insidious factors limit the potential of the bottom-fixed wind energy. The repetitive impact of waves on the structures is inducing fatigue, which may limit the operating lifetime of the WTs. The impact of biofouling on the structures is also a concern, currently addressed through several research projects as it can modify the resonance frequencies of the supports. Finally, the major part of the WTs installed being monopiles, they are hammered inside the seabed. This procedure is very noisy, particularly dangerous for marine mammals, which imposes the use of sound protections rather complicated to set up.

These limitations can be tackled by the installation of FOWTs, which is seen as an important way of development for wind energy. The major technological breakdown brought by the floaters offers new possibilities for the installation of wind farms, which could take part in the energy transition. The exploitation of wind using FOWTs has already started in Scotland, where a first pilot wind farm was installed in 2017. Prototypes are also operating offshore France, UK, Norway or Japan, see James and Ros (2015). Floaters enable the installation of WTs in areas where the water depth reaches 300 meters, which increases the number of sites available for FOWT settling. Using floaters enables the towing of structures, which can reduce the complexity of the WT installation compared to bottom-fixed turbines. Heavy-going maintenance procedures

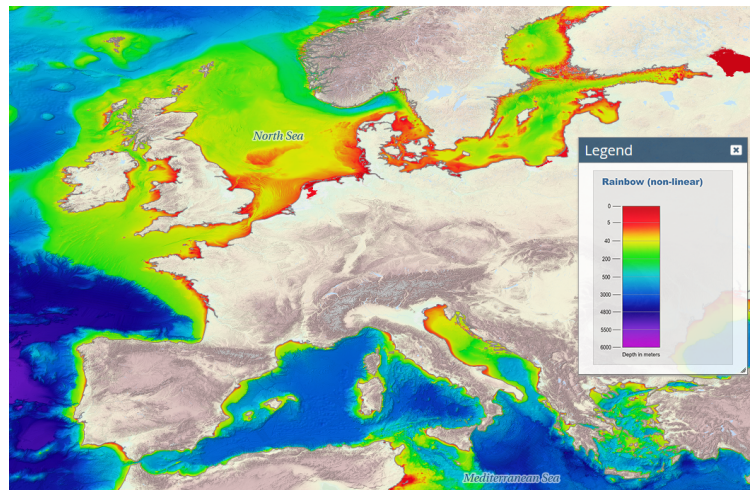


Figure 1.4: Bathymetry in Europe, from [EMODnet](#) ([2019](#)).

can be optimized in the same way. The major counterpoints to the use of floaters are related to motions induced by the combination of sea and wind effects, and its impact on both the performances of the turbines and the fatigue induced.

As a conclusion, FOWTs are the next instruments of the wind energy to be industrially deployed. They allow the deployment in offshore areas where strong, constant winds offer a potential for an important production, but where the depth prevent the installation of bottom-fixed turbines. The technologies are not mature yet, which justifies the development of high-fidelity simulation engines allowing to optimize the design of the different prototypes available.

1.1.2 Horizontal-axis wind turbines

Composition of a WT

Nowadays, a consensus has been established around horizontal-axis WTs. [Rapin and Noël](#) ([2010](#)) recalled that starting researches on the subject concerned both upwind and downwind concepts, with various numbers of blades. The downwind concepts were supplanted because of aerodynamic instabilities, the rotor being strongly impacted by the wake generated by the mast of the turbine. Similarly, the fatigue generated by the two-bladed rotor concepts condemned their industrial use. The industrial convergence is now established around upwind, three-bladed WTs, known as the Danish concept. This technology presents several advantages since, thanks to its geometry, the loads are well balanced over the whole rotor. Its energy production is also among the most efficient.

Today, most horizontal-axis WTs share the topology presented in [Fig. 1.5](#). Three composite blades are composed of airfoils mounted on a shaft. This forms the rotor, which is put in rotation by the wind. A gearbox is connected to the rotor, and placed in a nacelle disposed at the top of a mast. The gearbox accelerates the rotation speed provided by the wind energy, and its high-speed shaft is connected to a generator. Another technology, known as direct drive, connects directly the rotor to the generator. Getting rid of the gearbox reduces the electrical chain and the risk of failure, but increases the complexity of the installation procedure. For further details about gearboxes and the direct drive concept, see [Polinder et al.](#) ([2006](#)). The rotation speed of the rotor can be managed thanks to pitch controllers correcting the inclination of the blades. Electricity is produced and transmitted to a transformer located down the mast. This ends the energy conversion chain at the scale of a turbine. The electricity produced by each WT is gathered at the scale of the wind farm, which generates issues that are not addressed

in this document.

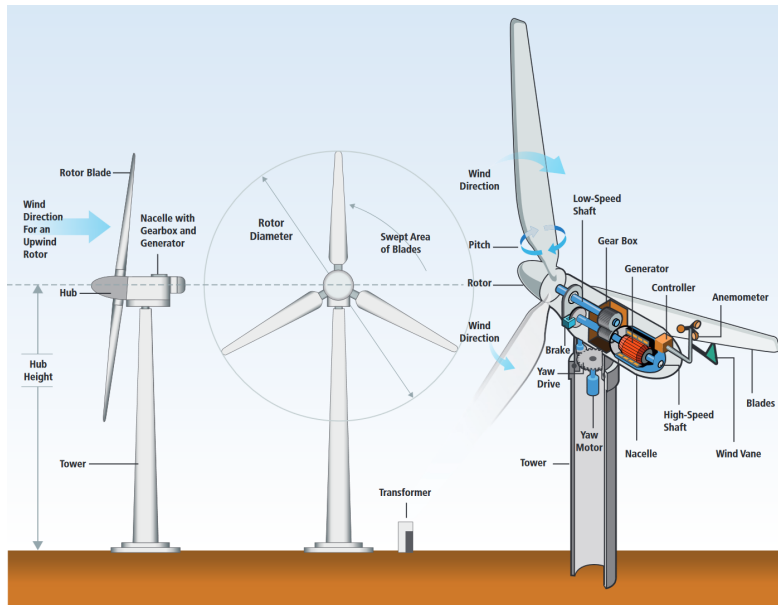


Figure 1.5: Description of a WT, from National Renewable Energy Laboratory (NREL).

Several challenges have been related to WTs. First of all, the elements composing them are disposed at the top of the mast. The center of gravity of the turbine is very high, which raises issues concerning its stabilization, especially in the floating wind context, where huge turbines are fixed on moving structures.

Aerodynamics of WTs and farm effects

An operational WT extracts energy from the ambient wind thanks to its blades. The ambient wind is turbulent and unsteady. Thus, the behavior of the boundary layers developing on the blades is very hard to predict. The interaction between the wind and the rotor generates vortices, which have a dynamic influence on the turbine and on the wake. The tower shadow modifies the wind distribution and complicates the studies. At the scale of the blade, dynamic stall, which corresponds to boundary layer detachment, may be observed. This particular effect deteriorates the lift applied on the blade locally. For all those reasons, the phenomena occurring around the turbine are rather unpredictable.

The rotational speed of the turbine can be determined from its design and from the wind speed observed at the rotor. A simple analysis is sufficient to show that the rotation of blades is extremely fast. Leroy (2018) realized a study on a horizontal-axis FOWT, whose TSR is 7 for a wind speed of $11.6m.s^{-1}$. The speed of the tip of the blades for this test case is thus 7 times faster than the ambient wind speed, at $81.2m.s^{-1}$. Consequently, the aerodynamic velocity observed at the tip of blades is represented mostly by the relative wind speed, generated by the rotation of the blades. The velocity of the flow is considerable, which tends to increase the computational effort required to represent the aerodynamics.

The flow around WT blades can be characterized using the Reynolds number Re , a very classical dimensionless measure in aerodynamics. For a fluid of dynamic viscosity η and of density ρ , its flow of velocity u around an object of characteristic length L is written in Eq. (1.1). The Reynolds number provides information about the boundary layers developing around objects under the action of fluid flows. For $Re \approx 1$, the flow is mainly attached to the geometries. When Re increases, turbulent structures can be

observed, even if the transition between laminar and turbulent flow directly depend on the geometries simulated. For $Re > 10^3$, the flow is usually unsteady. In the case of WT, the Re numbers around a WT blade is computed from the properties of the surrounding ambient flow, and uses the local chord of blade sections as characteristic length. The Re observed around the tip of blades reach orders of magnitude of 10^6 . This results in the generation of highly turbulent flows in the wake of the turbine, which are hard to predict.

$$Re = \frac{\rho u L}{\eta} \quad (1.1)$$

The WTs operating for industrial use are installed in wind farms. Aside from grid integration and other electrical considerations, the wake interactions effects occurring at the farm's scale are a very important topic. Whenever a WT is operating, the aerodynamic flows around it are highly perturbed. Consequently, the operational behavior of the WTs located in the wake is strongly modified. The incoming airflow is much more turbulent, which reduces the energy production and generates fatigue on the structures, thus reducing the life span of the turbines. A famous illustration of this effect is presented in Fig. 1.6. This picture, taken at the Horns Rev wind farm, depicts clouds appearing in the wake of WTs because of the pressure drop. Strong interactions can be highlighted between the downwind turbines and the wake of the upwind one. As mentioned before, a majority of the turbines are no longer operating in the optimal conditions, and limit the production of the whole wind farm.



Figure 1.6: Wind farms wake effects: clouds appearing in the wake of WTs at the Horns Rev wind farm (Denmark). Picture taken by Christian Steiness, 12 February 2008.

As a wind farm intends to produce as much energy as possible, the operational time spent in the situation of Fig. 1.6 must be reduced as much as possible. To that extent, the design of wind farms needs to be optimal. Wind roses are measured at the wind farm location, and turbines are placed depending on the prevailing wind. Active control strategies can be implemented, where an upwind turbine is yawed in order to deflect the wake, which can avoid the downwind turbines.

Controlling WTs

The objective of WTs is to extract electricity from the wind through a mechanical chain. The wind impacting the blades generate a lift force, which initiates and maintains the rotation of the WT rotor through the torque generated. The intensity of the lift force depends on wind speed and on the orientation of the blade. The action of the wind on the blades also generate a drag force, which is aligned with the inflow and induces a thrust. Control is applied to the WTs to both optimize the energy production and the lifetime of the machine. The objective of control is to keep torque and thrust as close as

possible from the optimal WT configuration.

An example of the control strategy is displayed in Fig. 1.7 for a 2MW-WT. Different operating modes can be identified from this curve: when the wind is too weak, typically lower than $3m.s^{-1}$, the WT is stopped. When the wind increases but remains inferior to the rated wind speed, commonly around $15m.s^{-1}$, the production is maximized. The rated wind speed is the wind speed which optimizes the performances of the WT, i.e., the optimal operational situation. The blades of the WT are actuated to produce as much as possible, which returns a production at the nominal power of the WT. The tip speed ratio (TSR), which characterizes the rotation speed of the tip of the blades regarding the ambient wind speed, is optimized at the rated wind speed. It reaches a value of around 7, which corresponds to the ideal TSR for 3-bladed, upwind WTs. When the wind speed increases again, the performances of the WT have to be altered by controlling the pitch angles of blades to keep producing at the nominal power. The reduction of production is necessary to preserve the integrity of the electrical chain. In this operating mode, the TSR of the rotor is progressively reduced. When too strong winds are observed, with, e.g., stormy winds reaching $25m.s^{-1}$, the turbine is stopped.

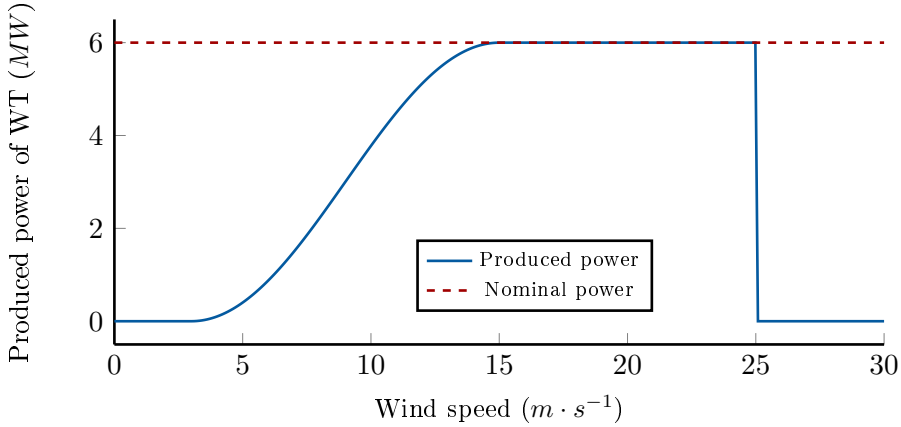


Figure 1.7: Theoretical power curve of a 6MW-WT.

The rotor needs to be aligned as well as possible with the incoming wind, to reach optimal production and to better balance the loads, and thus to limit fatigue. This alignment is obtained with active control of the nacelle’s yaw angle, as depicted in Fig. 1.5. Once correct alignment is observed, a common strategy to limit the production of the turbine is to control the blades in pitch. This idea can be regarded from a 2D point of view, as the sections of the blades are aerodynamic profiles. Increasing the pitch angle increases the relative wind angle of attack of the blade sections having the major contributions for power production. This correction facilitates the apparition of the stall, situation for which the lift is largely decreased. Applied individually or collectively on each of the blades for a uniform wind, it allows distributing the efforts applied on the rotor. When the turbine needs to be stopped, for high wind speeds or during maintenance, the pitch of the blade is brought to a situation where the production is minimal, tending towards zero. A security brake is also activated on the shaft, to guarantee an immobile rotor.

One of the challenges of the control laws is to identify the mode in which the turbine is operating. Even if the nacelle is equipped with an anemometer and a wind vane, or sometimes a Lidar, those sensors are located in the wake of the turbine. Their signals are largely perturbed by the wake generated by the rotor, and are not particularly reliable. Moreover, the relative wind impacting the blades is not uniform, and the information

obtained at the rotor may not be sufficient for accurate active control. Consequently, the pitch of the blade is commonly ruled by the loads measured on the blades. This technique allows to account for non-uniform distribution of wind, as the pitch is controlled separately on blades, through a balance of loads among the whole rotor.

1.1.3 Floating offshore wind turbines

If the conception of WTs is the same for floating purposes as for onshore one, the stronger winds found offshore, the smaller interaction with human activities and the technological advances pave the way for the use of giant WTs. As an answer to resource, transport or social acceptance issues, the industry tends to favor offshore wind.

While onshore turbines usually have a 2MW nominal power, the floating farm installed offshore Scotland during the second semester of 2017 is composed of 6MW turbines. However, the current development trend is to design even larger turbines, e.g., with General Electrics announcing in 2018 the commercialization of a 12MW turbine. [James and Ros \(2015\)](#) noted that the industry currently think of bigger WTs as a way to reduce the Levelized Cost Of Energy (LCOE) of offshore wind.

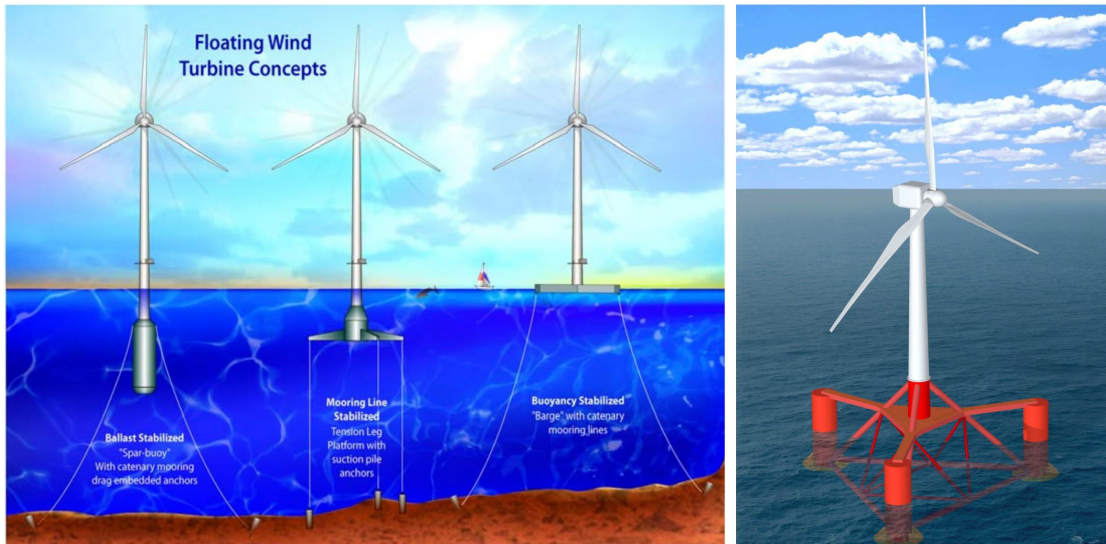
The development of FOWTs has accelerated over the last decade, as the first full-scale prototypes were installed around 2010. [James and Ros \(2015\)](#) did an overview of the different designs currently developed or tested, with emphasis put on technology readiness and on the different challenges to meet before industrial production. While many prototypes have been studied, they were all composed of: a floater to ensure stability, a WT disposed on the floater, several mooring lines for stationkeeping, and an electric cable to connect the turbine to the grid. The different elements are presented in the upcoming Subsecs.

Floaters

There are a lot of challenges to answer when designing a WT floater. First of all, the floater needs to provide buoyancy and stability for the whole structure, and both the massive size of the engines and the gravity center of the turbine located upwards makes it very challenging. As the efforts applied on the blades are transmitted to the floating support through the tower, a strong overturning moment can be imposed, especially on huge turbines. These aerodynamic effects are coupled to the wave-induced loads, generated by the swell and the waves directly on the floater. These multiple complex effects impacting the dynamics of the floater are hard to predict accurately.

Another objective is to limit the uncontrolled motion of the FOWTs, to keep the best alignment possible between the rotor and the wind. The idea, in the context of FOWTs, is to optimize the LCOE. Reducing the cost of the floater is a way to reduce the costs, and should be an industrial objective to help the floating wind energy reaching maturity. The understanding of the phenomena acting on FOWTs is necessary for this optimization. Today, the difficulty to model meteorological states with precision, along with the strength of extreme events encountered offshore, e.g., centennial waves, often lead to an oversizing of the floater. The same issue needs to be solved for the operational expenditures (OPEX), as the lifespan of a floater will be counted in decennials. More generally, the initial and recurrent costs of the floater are, as well as the production capacity of the WT, key indicators for the feasibility of an offshore wind farm. Consequently, the development of optimal stability criteria for floaters is critical for the floating wind industry, as a compromise between low expenses and robustness will need to be found.

Several types of floaters have been proposed since the beginning of the researches on



(a) Three different concepts of floaters, spar (left), TLP (middle) and barge (right), from [Butterfield et al. \(2005\)](#). (b) Example of a semi-submersible floater, the Dutch Trifloater.

Figure 1.8: Different concepts of floaters.

the subject. They are largely inspired by the technologies used in the oil industry. Three main types of floaters exist, as listed by [Butterfield et al. \(2005\)](#) and depicted in Fig. 1.8. On the left of the picture, a spar buoy is presented. This ballast-stabilized concept relies on a ballast, disposed deep down under the water surface, for its stability. This mass enables to bring down the gravity center of the whole structure under its center of buoyancy, which guarantees stability. In the middle, the family of mooring-lines-stabilized floaters is presented. This concept is very different from the spar one, as the standalone assembly of the turbine and the floater is unstable. Very good stability is provided by the tension in the mooring lines, and this concept is thus named Tension Leg Platform (TLP). The third concept of floater is the barge. This technology is designed to have a huge waterplane area, which compensates for the tilting moment induced by the efforts applied on the rotor. A last design, derived from barges, is also presented in Fig. 1.8. This concept, known as semi-submersible (semi-sub), has hydrodynamic properties close to those of the barges. It usually comes with three columns interconnected, forming a support on which the turbine can be mounted. Compared to barges, the design of a semi-sub is more complex, but reduces the quantity of materials required for its construction.

To classify floaters, a methodology based on a stabilization triangle can be used, as presented in Fig. 1.9. Each edge of the triangle corresponds to one of the three stabilization modes available for a floating structure: using a ballast, the tension in the mooring lines or buoyancy. While an approximative positioning is proposed on the illustration, geometries, materials or mooring choices have an impact on the position of each floater inside this triangle. This methodology enables to store the different technologies while defining the influence of any type of stabilization, and [Butterfield et al. \(2005\)](#) showed that the contribution of the three types should be approximatively the same to minimize costs.

The prototypes developed initially were composed of steel for the major part. This design was inherited from marine experience, bottom-fixed foundations and offshore oil engineering, but a major drawback can be found for steel. The cost of a steel-floater tends to be very important, and some prototypes now start to consider concrete as an alternative. Concrete is also lighter, which offers the possibility to design smaller floaters. Even if floaters designed in concrete are heavier, their cost remains lower, which

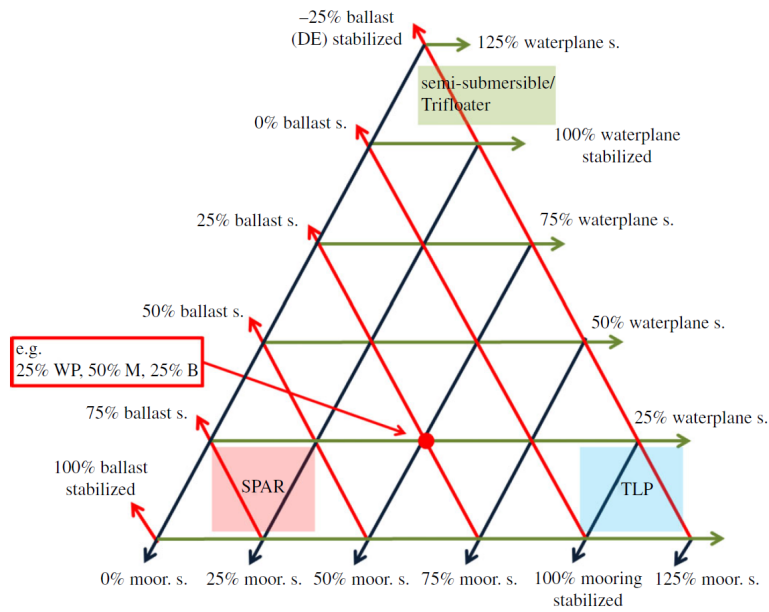


Figure 1.9: Classification of different floaters in a stabilization triangle, by [Borg and Collu \(2015\)](#).

is profitable in a context where CAPEX has to be reduced as much as possible. Some prototypes have even been developed to be built either in steel or in concrete, depending on the opportunities occurring for each project, e.g. the barge from IDEOL.

Mooring lines

The mooring lines are critical for the stabilization of the FOWTs. They guarantee the stationkeeping, which preserves the electrical cable to encounter high curvatures, but can also provide necessary stability in the case of TLP. Three different designs of mooring lines are used in the context of FOWT, depending on the prototypes installed and on the seabed. They are presented in Fig. [1.10](#).

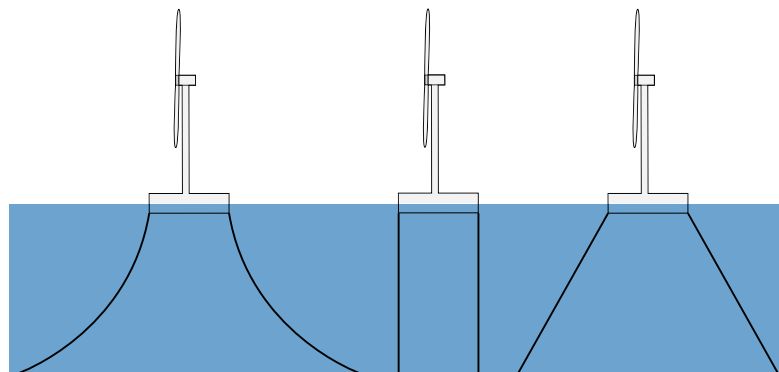


Figure 1.10: Different mooring concepts: catenary lines (left) ; tension-leg moorings (center) ; taut-leg moorings (right).

The first concept uses catenary lines, which stabilize the FOWTs with its own weight. At least three massive lines, usually crafted in metal, are attached to the floater on one hand, while the other end is anchored to the seabed. When loads are applied to the floating structure, the inertia of the mooring lines limits its movements. This technology is not suited for TLP FOWTs, whose structural definition requires mooring lines bringing tension as a matter of stability. This tension-leg concept relies on shorter lines anchoring

the floater to the seabed, commonly using synthetic fiber or wire ropes, which can largely reduce the CAPEX invested in mooring lines. The last technology, the taut-leg mooring, has been developed as an intermediate between catenary and TLP. The taut-leg concept features tension in the mooring lines, which are not oriented vertically as for TLP but form an angle of about 45° . This technique provides interesting stability, uses cheaper lines than catenary, and is less challenging to install than the TLP concept.

Electrical connection

The connection of a wind farm to the grid is performed using high-tension electrical cables. Those cables are buried into the seabed to limit potential degradations. The most common technology used for electricity transport is alternative current (AC), as commonly used onshore. However, the losses when electricity is transported on long distances using AC increase rapidly. A solution is to use transformers to regularly reamplify the electrical signal received, but this solution makes the energy distribution chain more complex, thus increasing the risk of failure. As the tendency is now to install wind farms far offshore, high-voltage direct current (HVDC) is another option, detailed by [Kirby et al. \(2001\)](#). HDVC lines are less expensive, as shorter cables are needed, and encounter limited losses during electricity transport. However, this strategy requires the construction of appropriated energy conversion installations onshore. This research subject is not further explored in the future developments.

1.2 Numerical simulation of FOWTs

Numerical simulation of wind turbines is a topic that has emerged over the last decades, following an important rise in the number of wind turbines installed onshore. The development of accurate and computationally efficient techniques helped the industry to reach maturity, while several codes were validated against experimental data. The first offshore wind turbines were bottom-fixed, which established a convergence point between wind and marine renewable energy sectors. In this bottom-fixed context, the aerodynamics and hydrodynamics can be studied separately, given the minor actions of air on the supports and of water on the rotor. The impact of the waves on the structure require appropriated dimensioning, which is realized thanks to an accurate comprehension of the hydrodynamics. The motions of the offshore bottom-fixed turbines are comparable to the ones of onshore WTs, and, consequently, the aerodynamic models designed for onshore wind are still applicable. This ability to split the coupled problem between aerodynamic and hydrodynamic ones reduces the complexity of the studies. On the contrary, FOWTs are much more impacted by environmental conditions, and the air/water interactions cannot be neglected.

The difficulties encountered with basin experimentations highlight the importance of numerical simulation of FOWTs, which is presented in this Sec. The resolution of the aerodynamics of WTs is presented in Sec. [1.2.1](#). The simulation techniques used for the hydrodynamics are presented in Sec. [1.2.2](#). Different methodologies applied to the simulation of FOWTs are presented in Sec. [1.2.3](#).

1.2.1 Aerodynamic of wind turbines

The simulation of FOWTs is a growing field, which benefits from the experiences gathered by the onshore wind community. Similarities can be found in the modeling of the rotor. Even if the floating context increases the interdependency between aero- and hydrodynamics, the reduced computational effort justified the application of the efficient methodologies developed for onshore wind to the offshore context. This Subsec. focuses

on the numerical methods used to simulate WTs in the onshore context. Some of those methods are directly applicable to the simulation of FOWTs.

Wind modeling

In a prelude to the simulation of WTs, the representation of accurate aerodynamics is necessary to guarantee the quality of the results obtained. The wind profile as a function of the altitude, the Atmospheric Boundary Layer (ABL), has been particularly studied, and is still under particular attention as the WTs grow bigger. The wind speeds are null at the ground and progressively increase, before reaching a plateau. The behavior of the ABL directly depends on the terrain surrounding the area observed and of the air stability in the atmosphere. The general description of ABLs focuses essentially on the definition of correct wind speeds and directions. As presented in [Sempreviva et al. \(2010\)](#), some differences can be found between onshore and offshore ABLs. The simulation of ABLs relies on the imposition of boundary conditions as input when Navier-Stokes (NS) solvers are considered. A focus was placed on the generation of coherent boundary conditions in [Blocken et al. \(2007\)](#).

Wind also comes with small, vanishing aerodynamic effects, which nonetheless have an important impact on WTs. Those aggregated effects form the turbulence, which has an impact on energy production and on the lifespan of WTs. The characteristics of turbulence, small and vanishing, makes it very expensive to compute. The size l_K and the duration τ_K of the smallest events, known as the Kolmogorov scales, are expressed in Eqs. [1.2](#) with η viscosity and ϵ the average rate of dissipation of turbulence kinetic energy.

$$l_K = \left(\frac{\eta^3}{\epsilon}\right)^{1/4} \quad (1.2a)$$

$$\tau_K = \left(\frac{\eta}{\epsilon}\right)^{1/2} \quad (1.2b)$$

A coarse approximation of ϵ writes $\epsilon = U^3/L$, with U and L respectively the reference velocity and length. For a flow of $Re = 1M$ around a tip of WT blade of section $L = 1m$, the order of magnitude of the small scales are $l_K = O(10^{-5})m$ and $\tau_K = O(10^{-4})s$. In order to capture all the physics, mesh cells of size inferior to l_K are necessary, which requires huge computational resources. Consequently, several wind modeling techniques are either considering inviscid wind or intending to capture the effects characterized by a long reference time. The representation of turbulence is possible with an accurate resolution of the NS equations, i.e., with LES- or DNS-type formulations, as described in Sec. [3.1](#).

Rotor representation

Different modeling strategies have been developed. The most accurate one consists in the representation of the whole blades, which are tracked during their revolutions. A first approximation simplifies the geometry of the blades, to reduce the computational effort associated. Simpler, quasi-steady approaches only consider the pressure drop occurring around the WTs, and forget the motion of the blades. All these models aim at measuring the efforts applied on the rotor by the wind, to evaluate the aerodynamic perturbations generated by the WTs. To that extent, rotor models can be used in a coupling with a flow solver. The definition of the relative wind at the rotor is better described with coupled simulations, and the perturbations form a wake of particular interest for farm considerations. The next paragraphs present different rotor models.

The simplest rotor model is the Actuator Disk (AD) method, used in [Sørensen and Myken \(1992\)](#). An axial, steady potential flow is considered. The rotor is represented as a porous disk, through which the flow can travel, at the cost of a pressure drop. An evaluation of the thrust and power coefficients measured at the turbine can be obtained, which allows measuring the energy produced. They are labeled C_T and C_P , and are presented in Eqs. [\(1.3\)](#) with u_∞ the ambient velocity and u_d the velocity traveling through the disk. This approximation is particularly used in the context of wind farms, where a huge number of turbines have to be simulated. However, the wake induced by an AD does not take into account the rotation of the rotor.

$$C_T = 4a(1 - a) \quad (1.3a)$$

$$C_P = 4a(1 - a)^2 \quad (1.3b)$$

$$a = \frac{u_\infty - u_d}{u_\infty} \quad (1.3c)$$

Another technique, the Blade Element Momentum (BEM) method, is an efficient approach computationally speaking to obtain an estimation of the loads applied on a WT rotor. An extensive study of BEM can be found in [Lanzafame and Messina \(2007\)](#), and a comparison between BEM and AD techniques can be found in [Madsen et al. \(2007\)](#). This method operates as a standalone method, combining the AD theory and blade element considerations. Consequently, this method is valid for steady and axial flows only. From the specification of blade geometry, especially the aerodynamic profiles defining the blade sections, drag and lift coefficients are computed, which enable the determination of forces. They are respectively named C_D and C_L , and are expressed in Eqs. [\(1.4\)](#), with u the velocity of the flow and A the surface of the airfoil. The drag and lift coefficients of the airfoil $C_{D,a}$ and $C_{L,a}$, which depend on α the orientation of the airfoil and on Re the Reynolds number of the flow, are determined from tabulated data. This method has been widely used, but the results presented in Eqs. [\(1.4\)](#) are valid for aligned, steady flow. This situation is utopian, and several corrections have thus been proposed from empirical considerations.

$$C_D = \frac{1}{2} \rho u A C_{D,a}(\alpha, Re) \quad (1.4)$$

$$C_L = \frac{1}{2} \rho u A C_{L,a}(\alpha, Re)$$

The empirical corrections, presented by [Hansen et al. \(2006\)](#), provide an extension of the BEM towards real operational situations. Among the major corrections, the dynamic inflow model provides an adaptation of BEM when the wake is not established. It allows to correct the evaluation of aerodynamic coefficients when brutal angle shifts are observed. A second correction is the yaw/tilt model, which considers the misalignment between the wind and the rotor. The forces applied on the blades are pondered depending on their position, and the blades positioned upwind induce weaker velocities than those located in the wake. A third correction considers the dynamic stall. When the pitch of blades is modified, a detachment of the boundary layers may occur. In classical BEM, the drop in C_L is directly applied. The dynamic stall technique provides a better smoothing to C_L , through a repartition of the shift along a certain time. Another correction, known as Prandtl's tip loss, takes into account the vortices generated at the tip of the blades. A last addition to the BEM can be used to generate realistic wind profiles. This method has progressively lost influence, as accurate wind repartition can now be obtained thanks to a coupling between BEM and CFD solvers. Those different corrections enable to adapt BEM to operational WT situations. However, this application keeps considering a potential flow, and misses turbine/wake interactions.

Another set of methods rely on an inviscid characterization of the flow around the WT blades, and on the propagation of the aerodynamic effects. The BEM can be used on a blade section to generate vortices at the tip of the blade, which will then form the wake. The vortices are propagated, which allows performing dynamic simulation of the WT wake. A prescribed motion of the vortices can be defined, using a constant wake velocity for the advection. This method is known as Prescribed Vortex Wake (PVW). An extension of this method considers free motion for the vortices, where each vortex undergoes the influence of all the others. This method, known as Free-Vortex Wake (FVW), provides an interesting behavior in the sense that a vortex previously generated has an impact on the newest. The dynamic behavior induced offers the potential to simulate accurate wakes when the relative wind at the rotor is varying, e.g., when a floating WT is considered. The free-vortex methods provide better precision than BEM, especially for thin aerodynamic effects, as stated in [Blondel et al. \(2016\)](#). The authors noted that this limitation may disqualify BEM for floating wind applications, where the fluid-structure interactions are considerable. However, the computational effort required by the FVW approach is considerable, as the wake generation requires the propagation of many vortices. A FVW method is presented in [Sebastian and Lackner \(2012b\)](#). Extensions of this approach can be realized to get an enhanced precision. This was done, e.g., by [Pinon et al. \(2017\)](#) to account for the turbulence in the wake. Their study was oriented towards the simulation of tidal turbines, for which the turbulence issues are more important. Still, this modified FVW method is applicable to WT simulation.

All the previous methods considered inviscid flows. To account for the viscosity of the flow, the Navier-Stokes equations have to be solved. The flow solvers developed can be coupled with the AD or BEM methods previously presented. To get a better representation of WTs, the Actuator Line (AL) method was developed. The AL models the rotor as a set of lines, as presented in [Shen \(2002\)](#). Different sections of the blade are studied, and the theoretical force applied by the flow on each section is evaluated using BEM approaches. Those forces are sent back to the flow solver, and create perturbations in the wake of the turbine. This technique generates a dynamic, rotating wake. The additional computational cost compared to AD is notable, but for the study of a single WT, several authors preferred AL which provides unsteadiness. For pure aerodynamic simulations, this formulation allows to study the unsteady flow behind a WT using monophasic simulations, as realized by [Benard et al. \(2018\)](#). An extension of this method is performed with surfaces instead of sections for the force evaluation, as in [Shen et al. \(2007\)](#), to create the Actuator Surface (AS) method. Even if the idea is similar, AS is supposed to bring additional precision. However, little improvements are observed from AL to AS, and the increased computational effort tends to limit the propagation of this method.

The last rotor representation technique requires an exact reconstruction of the blade geometries, along with an accurate assembly of the rotor. This approach is extremely expensive computationally, as the aerodynamic boundary layers have to be captured on all the blades. A WT is simulated using this approach in [Bazilevs et al. \(2011a\)](#). While the BEM can be used as an industrial technology in terms of rapidity, the FVW can be used only in R&D processes, and the full-representation technique may be affordable just for validation of softwares computationally cheaper. However, it offers the opportunity to couple an aeroelastic solver with the fluid solver used, which allows observing very thin effects completely overlooked by the other approaches. This strategy was developed, e.g., in [Bazilevs et al. \(2011b\)](#) of [Quallen and Xing \(2016\)](#).

Coupling with CFD solvers

As already mentioned, the less expensive numerical methods used to represent the solver consider inviscid flow. This approximation is acceptable to get approximative results, but CFD solvers are required when accurate numerical simulations are realized. Those applications need a precise representation of the wind at the rotor, which sometimes interacts with the wake. Complex terrains require the modeling of the obstacles perturbing the flows, and turbulence may be important under certain conditions. All of these characteristics require the use of a flow solver.

The flow solver refers to the resolution of the NS equations, which is introduced in Sec. 3.1. In a WT context, the simulation aims at obtaining a correct representation of the wind distribution in time and space. Depending on the precision required in the simulations, different approaches can be used for the NS resolution. Among the different options presented in Krogstad and Eriksen (2013), a major part used a RANS formulation of the NS equations, while some used LES. These formulations are presented in Sec. 3.1.

The simulation of a FOWT needs to be regarded as a fluid-structure interaction problem. The presence of the structures modify the flows, which have an impact on the motion, and potentially deformations, of the aforementioned structures. A coupling needs to be established between the WT model and the flow solver to realized WT simulation. The coupling strategy directly depends on the kind of rotor modeling selected. When standalone methods such as BEM are used for WT simulation, communication is established between the flow solver and the WT model. The ambient wind computed thanks to the flow solver are used as input of the rotor model. The outputs from WT modeling are brought back to the flow solver, in particular to perturb the WT wake. On the contrary, when intrusive methods such as full-rotor reconstruction are used, the forces and wake induced by the WT are directly deduced from the aerodynamic circulations, i.e., from the flow solver results. Different coupling strategies can then be implemented, which are presented with more details in Sec. 3.1.

Particular attention needs to be placed in the selection of the models used for both the flow solver and the WT modeling. The accuracy of these models in space and time needs to be similar, to obtain numerical errors balanced between the two models used. Along with the coupling strategies, the definition of the models must be representative of the test case simulated too.

Limitations towards direct offshore applications

The floating technologies are highly impacted by both wind and waves. Strong meteorological events induce the motion of the floating structure, which modifies the relative wind at the rotor's hub. Sebastian and Lackner (2011) showed that under weak wind conditions, a pitch-back of the floating structure generates interactions between the rotor and its wake. Those aerodynamic effects are not observed in onshore context, and consequently, the classical onshore simulation engines do not handle it. Moreover, with the trend of going far offshore, the size of the WTs tends to rise. Wind speeds defined by the atmospheric boundary layer cannot be considered constant over the whole rotor traveling area. This observation limits the application of models considering only the wind speed at the hub, even if this statement shall be moderated concerning offshore situations, as stated by Sark et al. (2019). The models widely used for bottom-fixed wind need to be adapted to account for the structure motion. The hydrodynamic loads applied on the floater have to be captured accurately, to guarantee a good estimation of the displacements encountered.

1.2.2 Hydrodynamics

The influence of the waves on the structures supporting FOWTs is a critical point for the development of the technologies. The understanding of the dynamics of waves, along with a correct numerical representation, is crucial to obtain an accurate simulation of FOWTs. The challenges related to hydrodynamics are presented in this Subsec.

Wave theories

The characterization of waves is necessary to measure correctly the movements of a FOWT. The hydrodynamics are ruled by the NS equations, which have long been prohibitively expensive to solve. Several approaches allowing to obtain approximative results at a moderate cost have been developed, and are briefly presented in the following.

The first approach historically considered is the potential theory. This expression of the velocity potential is obtained thanks to the resolution of a Laplace equation, which is simplified from the NS equations considering an inviscid and irrotational flow. The linearized potential theory applies to waves of small amplitude only, but non-linear approaches consider any kind of waves. The Boundary Element Method provides an efficient resolution, using Green functions to simplify the problem. This method returns kinematic and dynamic description of the free-surface. A linearization of the problem obtained enables to obtain a first approximation of the velocity potential. This results in an expression of the free-surface elevation.

An extension can be performed towards high order methods, taking advantage of a spectral basis and a Taylor expansion in the resolution, as presented in [Ducrozet et al. \(2016\)](#). While these methods provide an efficient and accurate representation of the targeted free-surface, the perturbations induced by a floating structure does not perturb the free-surface elevation in the wake.

The resolution of the NS equations is the upgrade of the potential flow formulation in terms of precision. It handles FSI but comes with much higher computational expenses. However, no expression describing the Free-Surface (FS) can be directly derived from the NS equations. Consequently, the generation of waves has to be handled differently. It imposes to go back to the sources of waves in nature: the wind and the swell.

Numerical generation of waves for CFD

The computational cost imposed by the generation of waves from natural sources is considerable. A common approach to tackle this limitation is to simulate smaller domains, where targeted wind/wave conditions can be reproduced. The development of Numerical Wave Tanks (NWTs) has enabled the generation of high-precision wave fields, critical to measure the hydrodynamic impacts on floating structures. Inspired by the experimental studies, the NWTs reproduce the main characteristics of water tanks: a wave-maker transmits energy to the water for the generation of waves ; a numerical beach limits the reflexion of the waves on the basin end. Coherent boundary conditions along with an appropriated domain definition enable to generate a wave field, obtained thanks to a resolution of the NS equations, in a preferential direction. A literature review of numerical wave tanks is proposed by [Windt et al. \(2018\)](#) in the context of wave energy converters, where similar challenges are observed when compared to the floating wind context.

Five wave generation techniques can be distinguished, and are presented in Fig. [1.11](#). In (a), the relaxation area methods acquire the solution of the NS equations as a combination of a targeted solution and a computed one. The waves are generated thanks to the targeted solution, which is non-zero only in the relaxation area and decays from

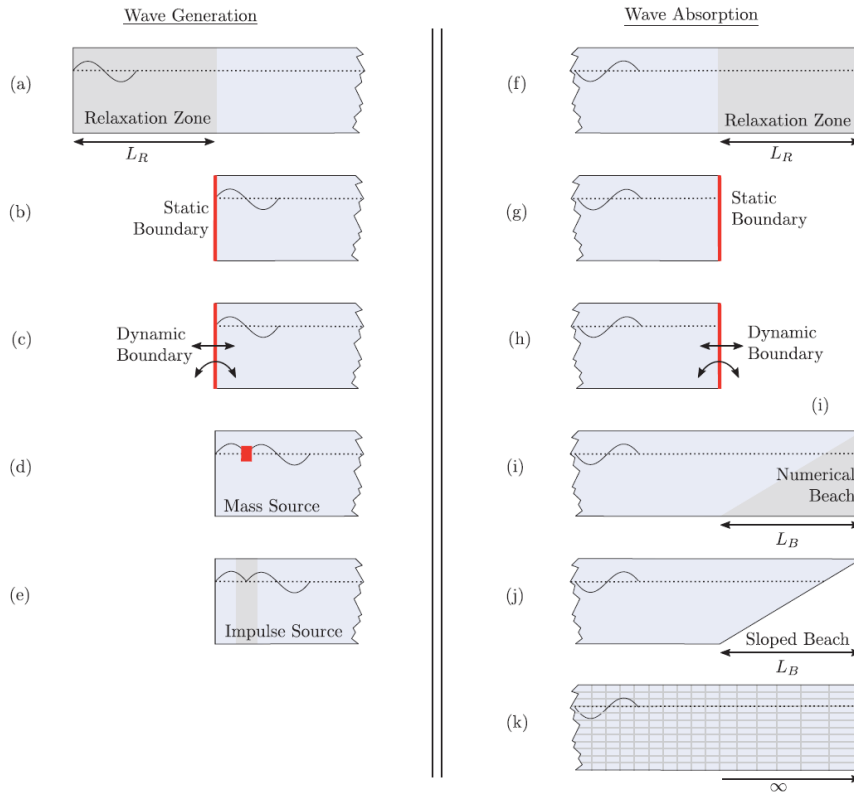


Figure 1.11: Wave generation (a)–(e) and absorption (f)–(k) in numerical wave tanks, from Windt et al. (2018).

the sides of the domain, defined from wave theory. In (b) and (c), boundary conditions are applied on domains and provide energy to the neighboring water. Those domains are respectively fixed or moving, which characterizes the static boundary or dynamic boundary methods. In (d), the mass source method consists in successive addition and removing of mass. In (e), the impulse source approach uses a source term which is integrated into the momentum equation. Those last two techniques necessarily generate two waves propagated in both directions.

The reflexion of waves in NWTs can be limited using different approaches. Six families of techniques are identified by Windt et al. (2018), and presented in Fig. 1.11. In (f)–(h), the three methods presented are similar to (a)–(c), mentioned in the previous paragraph concerning wave generation. This time, the target is obviously to extract all the energy from the incoming waves. In (i), the numerical beaches define regions where the energy of waves is dissipated. This can be achieved with viscosity increase, velocity canceling, or damping terms in the momentum equation. In (j), the physical definition of the domain integrates a geometry reproducing a beach, where waves break and, consequently, dissipate their energy. Finally, in (k), the last technique is to define a larger domain, where mesh cells can be coarsened at the rear. The numerical dissipation induced allows limiting the reflexion observed.

The generation of waves needs to be done regarding a particular sea state. The energy transported by the waves is highly dependent on the amplitude and frequency of the wave train studied. The excitation of the structures impacted by this particular wave train can have a huge influence on the motion or deformations observed, as resonance may appear. The validation of flow solvers and tank definitions is necessary, as well as the characterization of the sea states modeled. To that extent, the wave distribution observed at certain locations has been widely studied, e.g. the JONSWAP spectrum in

the North Sea.

FSI of floating structures

The subject of FSI is very important for the description of the dynamic of a FOWT. The objects on which FSI is realized need to be represented as accurately as possible, and the resolution should return the motions of the bodies under the action of hydrodynamic forces. The major issues of FSI are 1) to evaluate the forces applied by the waves on the structures, 2) to calculate the displacements of the floating objects, and 3) to compute the action of the structures on the surrounding water. Different methods used for floating FSI are presented in the following.

For slender bodies, wave-induced loads may be calculated through the Morison equation, proposed by [Morison et al. \(1950\)](#). This empirical formulation combines inertia and drag terms to evaluate the forces applied by waves on a structure. This method introduces simplifications, which reduce largely the computational costs. However, the viscous effects are completely canceled, which prevents its application to turbulent flows. While the Boundary Element Method defines the waves from the water fluxes, the Morison equation evaluates the forces applied to immersed structures. An important number of approaches returning the forces have been developed, some of them are listed by [Borg and Collu \(2015\)](#). They all come with their pros and cons, but often focus on simple geometries. This can become a limitation whenever the structures studied are complex, e.g. when a semi-submersible floater is modeled. This expression of the forces is standalone, but can also be coupled with a linear potential theory.

The movements of the structures may also be obtained from the linear potential flow theory. The resolution is performed in the frequency domain, for both waves and body movements of small amplitudes. The Cummins equation, detailed by [Cummins \(1962\)](#), enables to bring back the results to the temporal domain. The movements of oscillatory objects under a wave field can be determined from this resolution. Terms inherited from the Morison theory can be included in the Cummins equation. This formulation is supposed to bring better accuracy to the dynamic of the floater. However, this equation contains a memory term, which needs the knowledge of all the past velocities of the structure. Consequently, this equation is not solved in its original form, but must be treated to eliminate this particular term.

Another FSI approach considers the potential theory of the second order. The construction of Quadratic Transfer Functions (QTFs) is possible from this theory. Two main types of transfer functions can be built, which focus on particular wave frequencies. Consequently, a family will be able to represent accurately the high frequencies applied on a structure, while the other models efficiently the low frequencies. As an example, [Paik and Roesset \(1996\)](#) dedicated a hydrodynamic study using QTFs to tension-leg platforms, application close from the problematic of FOWTs.

CFD models can be used to get a flow field perturbed by the movements of the structure, at a much higher computational expense. CFD can be realized in different ways, using a single solver for both hydrodynamic and structural movements, or with coupled solvers. A particular requirement is the capture of both the free-surface and the structures. The evaluation of the forces can be obtained from one of the techniques detailed previously. Another approach is to determine the forces directly from the pressure applied by the fluid on the object, e.g., from a structural solver.

Once the floating structure is represented, the dynamic of the system is solved. The movements of the structure can be either prescribed or solved. Prescribed motions are

user-defined, and do not respect the efforts provided by the surrounding flows. On the contrary, the solved motions directly depend on the forces applied to the floater. The efforts are deduced from the flow, and may be measured locally from the fluid dynamics. The contributions induced by the displaced object can then be used to impact the surrounding water.

In the context of rigid structures, the force applied on 3 different points of an object is required to solve its movements. However, hydrodynamic loads may induce deformations of the object, and elasticity could be required to get accurate simulations. The integration of a structural solver has not been found in the literature in a floating wind context. The relevance of this study regarding the small deformations observed is the main reason, along with the important computational effort induced.

A key aspect of FOWT is the interaction with the mooring lines, as evoked in Subsec. [1.1.3](#). To prevent the floating structures from drifting, the moorings must absorb important hydrodynamic loadings. Different modeling approaches have been proposed, and are presented in an exhaustive study by [Davidson and Ringwood \(2017\)](#). The simplest one consists in representing each mooring line as a unique stiffness matrix, in a weak coupling. Better precision is achieved with springs modeling the different mooring lines, whose stiffness defines the span of movements allowed for the floating object. A more advanced approach considers the dynamics of a catenary line, modeled as a succession of weights interconnected using springs. These different models interact with the floating structure by communicating the forces applied by the mooring lines on the attach points placed on the structure itself.

1.2.3 Existing numerical simulation tools for FOWTs

This Sec. intends to make an inventory of the methods currently used to simulate FOWTs. In this particular Subsec., a focus is placed on the solutions used by the industry as design tools. The discussion is extended by an overview of the methods developed during the last decade, which targeted high accuracy in the results. A particular focus is placed on the models at the edge in terms of accuracy, basically CFD ones.

Engineering solutions for the simulation of FOWTs

The solutions used in the industry to design and certify FOWTs need to be as efficient as possible. The simulations tools providing rapid answers are favored compared to accurate and computationally expensive codes. An overview of the different softwares used for FOWT simulation in the industry is presented by [DNV-GL \(2019\)](#): 9 simulate complete FOWTs, and 4 are pure hydrodynamic softwares. The methods listed in the following paragraphs for industry use are described in Subsec. [1.2.1](#) for aerodynamics and in Subsec. [1.2.2](#) for hydrodynamics. The presence of hydrodynamic codes in this list allows reducing the computational expenses. They come with a precomputed hydrodynamic database, providing for every simple geometry the corresponding hydrodynamic coefficients.

The aerodynamics are handled in a majority of codes by BEM, using empirical corrections. Half of the aerodynamic codes also have a PVW method implemented. The other methods are rather confidential: only one code features FVW, and CFD is available for aerodynamics in two softwares. Even if they could bring a higher fidelity to physics, those methods are not favored over BEM, because of their longer computational times.

The evaluation of hydrodynamic loads can be realized in almost every software using a Morison equation. For better accuracy, FOWT aero/hydro softwares can solve a Cum-

mins equation with Morison drag. The calibration required by those methods for any geometry or sea state is emphasized by [DNV-GL \(2019\)](#). Experimental studies or CFD may be used as references. The pure hydrodynamic codes consider QTF evaluated from potential flow theories, which bring more flexibility, especially when the influence of the wind is negligible. However, in the context of FOWTs, the influence of the wind on the structures is rarely minor. Consequently, calibration is often required for QTF models too. The authors noted that, despite their high computational expenses, CFD provides a far better representation and understanding of the hydrodynamics around FOWTs. The study of extreme sea states using CFD instead of multiple situations with simpler models is a way to validate the behavior of floating supports.

The structural modeling of FOWTs is realized in engineering simulation tools using either multi-body system or finite element formulations. The reason for using rather accurate models for structures is advocated by the aeroelasticity, which is crucial to get a precise understanding of the dynamics of WT blades.

The moorings can be handled using stiffness approaches in every simulation code. A large majority of softwares can take advantage of quasi-static models, while finite element methods and multi-body system approaches have been implemented in half of the models. Results with an interesting fidelity can be obtained thanks to finite elements or multi-body systems. These representations allow understanding precisely the behavior of catenary lines, using a representation as connected beams. The efficiency of the computations is guaranteed by the 1D characterization of the mooring lines, which still provides interesting reliability.

As a summary, the FOWT industry intends to simulate as precisely as possible, given particularly strict time constraints. In a 3D, multi-physics context, fast and accurate are currently incompatible. Consequently, the computational expenses are reduced thanks to simplifications in the most expensive computational physical models: the aero- and hydrodynamics. The most used aerodynamic model, the BEM, has been corrected extensively, but keeps important physical shortcomings. The hydrodynamic models rely on a potential wave theory, which considers the water inviscid, and need to be calibrated before application to FOWTs. The need to get reference solutions to tune the engineering models brought researchers to more accurate methods, which are presented in the following.

High-precision simulation of FOWTs in the research community

The different limitations and approximations induced by the engineering codes lead to further studies, providing higher precision. Experimental studies are a way to validate the behavior of those codes. This is also the case of new numerical methods, usually more expensive but benefiting of successive technological advances to become affordable. CFD is an approach which attracted attention in the context of FOWTs.

The simulation of FOWTs using CFD is a fairly new topic, which has emerged during the last decade. It comes as a combination of challenges, inherited from WT simulation, ocean engineering or fluid-structure interaction. The simulations require huge computational domains, while small-scale effects need to be represented. The same behavior is computed on the time frame, as long simulations are needed, while vanishing effects can be observed. The meshing of the computational domain can become a burden, as the geometries of the floaters used are often complex. For all those reasons, CFD simulation of FOWTs remained a rather confidential topic. Several different CFD models exist, and each has its characteristic precision. Following the increase in the computing resources available, CFD models have been enriched to simulate the physics as accurately as pos-

sible. The coming paragraphs present the evolution of CFD-simulation codes towards a higher level of fidelity to physics.

The better known and one of the first numerical tools used for the simulation of FOWTs is OpenFAST (previously FAST), from the NREL. OpenFAST has been developed for various wind energy usages, from wind farms to FOWTs. It is thought of as a set of standalone codes, which can be easily coupled. Add-ons specific to each utilization have been developed, e.g. Aerodyn, Moordyn or HydroDyn for respectively aero-, moorings and hydro-dynamics. The study of [Jonkman \(2007\)](#) and the distribution of FAST has accelerated the development of the studies on FOWTs. Since then, a large number of researches have been realized using FAST, alone or coupled with other codes, e.g., by [Sebastian and Lackner \(2012a\)](#), [Kvittem et al. \(2012\)](#) or [Coulling et al. \(2013\)](#). The versatility of OpenFAST, illustrated by the number of add-ons available, allows generating different type of data. The most commonly regarded output are the movements of the FOWTs, the loads applied on the structures and the aerodynamic torque. These datasets provide critical information concerning the energy production and the stability of the platforms. However, the numerical approximations realized in OpenFAST tend to provide results with errors due to the modelization of the physical phenomena. The comparison against other codes, which provide a better accuracy at a higher computational cost, has been a recent research topic. As an example, [Leroy et al. \(2019\)](#) coupled a seakeeping software with OpenFAST's AeroDyn and a FVW code, and compared the results obtained with each method.

To tackle the computational effort required for a complete FOWT simulation, an option is to simulate the dynamics of only one component of the simulation. Many authors adapted their developments realized on aerodynamics in the context of onshore WTs to FOWTs. A correction of a FVW method is proposed by [Sebastian and Lackner \(2012b\)](#). A coupling between a RANS solver from the commercial code FLUENT and an unsteady BEM is realized in [Tran and Kim \(2015\)](#). The representation of an exact rotor is achieved by [Wu and Nguyen \(2016\)](#), and a RANS solver implemented in OpenFOAM is used. These studies focuses on the aerodynamics, particularly on the wind at the rotor and on the power production. Compared to onshore wind, the topic of wake effects is less present in the literature. The current research seems to focus more on the prototypes and their dynamic. The possibility to draw analogies between onshore and offshore wind farms is probably be the reason for it.

Another approach focused on hydrodynamics, to study the flows around structures or to characterize the floaters, e.g., [Nematbakhsh et al. \(2015\)](#) or [Beyer et al. \(2015\)](#). These hydrodynamic studies characterized the dynamic of the floating structures. The motion of the floater and its harmonics are the most common information presented. Some studies even studied the tension in the mooring lines.

More advanced simulations arose during the past years. They consist in coupled aero/hydro studies of a rigid FOWT, realized using CFD approaches. The coming paragraphs present the numerical methods used in these particular situations. The abbreviations presented in the coming paragraphs are explicated in Chap. [3.1](#).

The approach developed by [Calderer \(2015\)](#) consists in the coupling of a far-field solver, generating realistic environmental data, with a near-field solver, where the FOWT is studied. The free-surface interface is described using a level-set function, and a LES-FVM monolithic approach solves both the aerodynamics and hydrodynamics. The WT rotor is represented using an AL method, and the forces applied by catenary moorings are obtained from a linearization of the lines. The FOWT is placed in a wave tank,

where the generation of waves is realized thanks to a source-term wave-maker. A sponge layer is used as a numerical beach. A non-uniform structured mesh is used for the fluid domain, while the FSI is enabled by a meshing of the structure. The simulations realized focused on the dynamic of the FOWT under realistic sea states, but no error estimation is provided.

[Leble and Barakos \(2016\)](#) coupled three different solvers, allowing to get a test case representative of a FOWT. The hydrodynamics were ruled by a SPH solver, while a Multi-Body Model and moorings representation account for rigid floater movements. A Finite Volume solver using a RANS formulation of the NS equations (RANS-FVM) was used for aerodynamics, and a full description of the WT rotor is performed. The waves are generated using a hinged wave maker, and dissipated using a slope on the other end. A body-fitted mesh describes the rotor geometries, and a sliding plan is used to prevent the remeshing.

A different strategy is chosen by [Quallen and Xing \(2016\)](#), where the rotor is again exactly represented. A Delayed Detached-Eddy Simulation formulation of the NS equations is chosen, i.e., a LES approach is used far from WT, while a RANS one rules near-flow. A FVM solver permits the resolution of the airflow, while the free-surface, defined using a level-set approach, is enforced using BCs. A coupling is established with a mooring line model, and a controller is implemented to control the WT. The meshing is handled using an overset of regular grids of various precision. The aerodynamic forces and the motion of both the rotor and the platform are presented. Once again, no errors have been measured, as this study operates as a proof of concept for the coupling of the different solvers used.

[Yan et al. \(2016\)](#) took advantage of an isogeometric analysis to fully represent a FOWT, and used a level-set function to model the free-surface. A virtual-work structural solver is implemented, and coupled to the fluid one. The full system was solved using a monolithic LES-FEM approach. The mooring lines are represented using sets of articulated beams. Particular attention was given to the hydrodynamics, and a piston-type wave-maker is used for the wave generation. An ALE framework is used for the meshing of the computational domain, which allows accounting for the displacements of the different phases represented. The dynamics of the platform are verified against reference data obtained using a simplified approach, which highlighted a good overall shape and notable differences in amplitude. The dynamics of the platform are then presented.

[Liu et al. \(2017\)](#) computed the aerodynamic and hydrodynamic effects using OpenFOAM with a RANS-FVM solver, where a VoF approach defines the free-surface. The WT rotor is represented exactly thanks to a body-fitted meshing. A static modeling of the moorings is performed with a connection of weights. The waves are generated from BCs, while a sponge layer prevents the reflexion of waves. A rotating mesh handles the motion of the WT, and a chimera grid approach enables the communication with a background fixed mesh. This study is the most complete in terms of results presented, as the simulator is validated against downscaled FOWT experiments. The pressure coefficient around FOWT blades are compared against experimental data, which validated the aerodynamics. The force measured on the mooring lines and the motion of the floating platform are also validated. Aerodynamic thrust, torque, and FOWT motion are then presented for a steady wind and regular waves.

1.3 Aim and scope of this thesis

The developments presented in Sec. [1.2](#) introduced the principles governing the numerical simulation of FOWTs. An overview of the different methods used during engineering processes and the evolution of highly-accurate simulations was performed in Subsec. [1.2.3](#). Among the different approaches available for the simulation of FOWTs, this thesis aims at developing high-accuracy numerical tools at the edge. This answers to a need to bring more physics in this field, and intends to provide thin representations of the flows around FOWTs. The results obtained with such methods enable to calibrate, or extend the engineering models currently used. It may highlight effects of particular importance, especially for the hydrodynamics around floating supports with complex geometries. The high-accuracy results obtained will also have the potential to feed machine learning approaches.

CFD is a way to reach a high level of understanding of the dynamics of FOWTs, through the resolution of modified NS equations. Among the different CFD techniques available, it is important to distinguish the levels of accuracy offered by the NS formulations available. This thesis focuses on a formulation providing the best precision achievable when working at the scale of a FOWT. The shortcomings of the CFD approach are linked to the computational expenses required to run it, especially as high precision requires expensive computations. The application of high-accuracy CFD to the simulation of FOWTs is dependent on the optimization of both the numerical methods and the computational platforms used, such as supercomputers. These topics are discussed in this thesis.

During this thesis, developments have been realized towards the adaptation of the software platform ICI-tech to the simulation of FOWTs. ICI-tech proposes a full-CFD approach, where all the geometries immersed in the computational domain are represented. The reconstruction of elements takes advantage of an implicit boundary approach and of an automatically-adapted, anisotropic meshing engine. The anisotropy enables to reduce the number of mesh points used to precisely represent the geometries, and thus limits the computational footprint of the simulations. The NS problem is then solved under a VMS-FEM formulation, in a monolithic approach. These numerical tools are massively parallelized, which allows the deployment of ICI-tech on supercomputers. These methods allow simulating accurately the flows, while the parallelism and numerical optimization enable the realization of large-scale simulations. This highlights the potential for ICI-tech to provide an accurate representation of the loads applied on FOWTs.

The numerical tools implemented in ICI-tech can be split into two major parts. The first set is composed of the methods related to the reconstruction of immersed geometries. They are presented in Chap. [2](#), where an application to the reconstruction of FOWTs is performed. The second set of methods corresponds to the resolution of the NS problem. The numerical tools involved are presented in Chap. [3](#), along with the different developments related to the simulation of FOWTs. ICI-tech has long been used for the accurate modeling of viscous flows, while a FOWT operates in a quasi-inviscid environment. In Chap. [4](#), validation steps towards the simulation of accurate high-Reynolds flow and hydrodynamics are realized. The perspectives and developments further required for the adaptation of ICI-tech to the simulation of FOWTs are presented in Chap. [5](#).

Chapter 2

Implicit boundary representation

Contents

2.1 Reconstruction and level-set method	26
2.1.1 Theory of the mesh immersion using level-set functions	26
2.1.2 Immersion of objects represented by a set of elements	29
2.1.3 Tree-optimization of the reconstruction procedure	31
2.2 Mesh adaptation	35
2.2.1 Anisotropic mesh adaptation with a metric field	36
2.2.2 Metric construction using <i>a posteriori</i> error estimator	38
2.2.3 Mesh adaptation in practice	40
2.3 Highly-parallelized framework in ICI-tech	41
2.3.1 Parallel meshing in ICI-tech	41
2.3.2 Parallel reconstruction in ICI-tech	42
2.3.3 Mesh adaptation in the highly parallelized context of ICI-tech	42
2.4 Representation of WTs in ICI-tech	43
2.4.1 Reconstruction of a single WT	43
2.4.2 Reconstruction of a large number of wind turbines	48
2.4.3 Scalability of the immersion of FOWTs	50
2.5 Conclusion on the reconstruction of FOWTs	51

2.1 Reconstruction and level-set method

To perform simulations on FOWTs, the geometries describing the FOWTs must be represented in the computational domain. This process will be described as reconstruction or immersion in this document. In order to reconstruct the FOWTs, two major techniques can be used. The first one uses meshes fitting exactly to the geometries. This methodology, followed in the context of FOWT e.g., by [Yan et al. \(2016\)](#) or [Quallen and Xing \(2016\)](#), positions points of the computational mesh on the frontiers of the structures. This allows computing efficiently the loads applied to the FOWTs. However, the definition of the body-fitted mesh is challenging, especially when moving objects are simulated. This limitation is better handled by the other reconstruction approach, which considers implicit boundaries. With this method, the geometries to represent are defined from the positioning of mesh points or mesh cells inside or outside the object to immerse. The consideration of the mesh areas detected inside the body defines the reconstructed body. The main advantage of this technique is that any background mesh can be used. The reconstruction is dependent on the mesh used, which needs to be refined around the frontiers of the immersed geometries to get accurate results. However, the computation of the efforts applied to the structures becomes more challenging, as, statistically, no mesh point will be placed on the frontier of the objects. An immersed method is used for the simulation of FOWTs by, e.g., [Liu et al. \(2017\)](#) or [Tran and Kim \(2018\)](#). This implicit boundary method is used in ICI-tech too. The numerical framework linked to the reconstruction of immersed geometries is presented in this Sec.

2.1.1 Theory of the mesh immersion using level-set functions

The reconstruction of floating wind turbines in ICI-tech is achieved thanks to a mesh immersion procedure, which uses level-set functions. This technique is a way to define interfaces implicitly in a computational domain, which can be applied to any phase of the domain. A level-set function Φ defines the frontier of the immersed phase it represents through the points x for which $\Phi(x) = 0$. It shall also respect $\nabla\Phi = 0$ at the interface of the phases, in order to get a proper resolution of transport problems, necessary for the simulation of fluid phases. This method is indeed particularly suited for the representation of fluid interfaces. An early application to two-phase flows was proposed by [Sussman et al. \(1994\)](#).

This approach is implemented in ICI-tech as following. The reconstruction process considers the position of each point of the computational mesh, and compares it to the objects to immerse. The only requirement for this methodology is the existence of a frontier for each object, from where an interior and an exterior can be defined. This authorizes the evaluation of a signed-distance function α to this frontier at each point of the computational domain. The definition of α is presented in Eq. (2.1), for the immersion of an object ω of frontier Γ in a computational domain Ω . Note that no restrictions exist for the geometry of ω , which can even be defined as an assembly of non-connected elements.

$$\alpha = \begin{cases} d(x, \Gamma) & \text{if } x \in \omega \\ -d(x, \Gamma) & \text{if } x \notin \omega \end{cases}, \quad x \in \Omega. \quad (2.1)$$

Γ is defined by the locations where $\alpha = 0$, while the interior corresponds to $\alpha > 0$ and the exterior to $\alpha < 0$. The reconstruction of ω can be achieved using a function filtering the values of α , e.g., with a Heaviside function returning 1 for the interior and 0 for the exterior. Such a function introduces discontinuities, through the definition of a sharp interface. This formulation is not problematic for visualization purposes, but gets more challenging as numerical simulations must be conducted. In a multiphase context, when an implicit boundary method is used for the reconstruction, the definition of sharp

interfaces generates singularities in the cells crossing Γ . The application of mixing laws on these cells will be problematic. As an illustration, for a high-density ratio between ω and the outside, the highest value of density will have much higher influence, independently on the proportion of the cell located inside ω . A solution to suppress the discontinuity is to spread the transition from ω to the outside along several cells. A level-set function Φ_ε , presented in Eq. (2.2), can be designed to characterize this transition. It takes advantage of the properties of the hyperbolic tangent function, quasi-linear if α is close from zero and quasi-constant if $|\alpha| \rightarrow \infty$. This formulation concentrates the gradient of the level-set function around Γ . This will be a clear advantage over α , for which $|\nabla\alpha| = 1$, once the transport of immersed phases will be required.

$$\Phi_\varepsilon(\alpha) = \varepsilon \tanh\left(\frac{\alpha}{\varepsilon}\right). \quad (2.2)$$

This level-set function allows the definition of a smoothed Heaviside function H_ε , which is presented in Eq. (2.3). A transition is established around Γ , while H_ε ranges from 0 to 1, to conserve the properties of a Heaviside function.

$$H_\varepsilon(\alpha) = \frac{1}{2} \left(1 + \frac{\Phi_\varepsilon(\alpha)}{\varepsilon} \right) \quad (2.3)$$

An example of reconstruction is proposed for a complex geometry in a $1m \times 1m$ square. The distance function α , drawn in Fig. 2.1a, has a gradient of constant norm in space, which translates to a uniform evolution of the distance field. A comparison can be made with H_ε built for $\varepsilon = 10^{-2}m$, shown in Fig. 2.1b. A clear transition is observed around Γ in this case, while the isosurfaces $\alpha = 0$ and $H_\varepsilon = 0$ are identical. Both α and H_ε have been obtained with the mesh from Fig. 2.1c. This mesh was generated for this particular test case and features the expected precision in the transition area. The guidelines used for the generation of meshes propose to use 10 mesh cells across the transition area, which correspond to a minimum mesh size of $\varepsilon/10$.

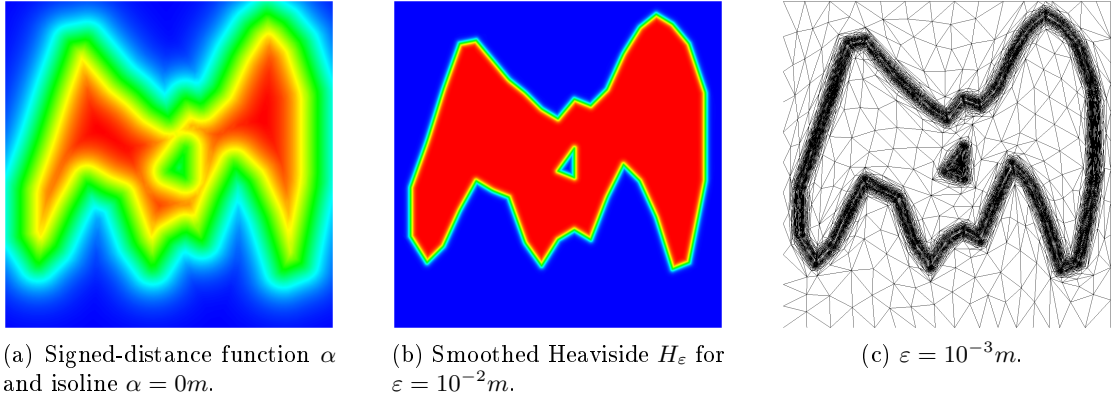


Figure 2.1: Immersion of a geometry ω .

The parameter ε operates as a controller of the transition area: increasing ε widens the smoothing region, while decreasing it tends towards a sharp interface. This statement is valid only if the computational mesh features enough points in the transition area. An illustration of the reconstructions achieved with three different values of ε is presented in Figs. 2.2a, 2.2b and 2.2c. The reconstruction of the immersed object is clearly influenced by ε , as Fig. 2.2c has a behavior closer to a sharp interface than the other test cases. These results were obtained using meshes containing enough mesh points in the transition areas. Consequently, the size of mesh cells around Γ is smaller for $\varepsilon = 10^{-3}m$ than for $\varepsilon = 10^{-1}m$.

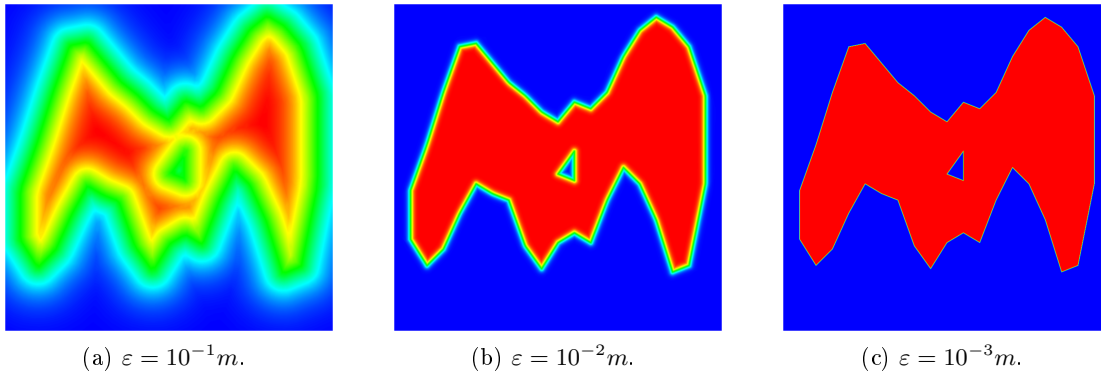


Figure 2.2: Smoothed Heaviside functions, for various ε .

This level-set approach for mesh immersion is heavily dependent on the precision of the mesh used. As previously mentioned, an important number of mesh points is required in the transition area to represent accurately the geometries, as well as the physic properties. The construction of Φ_ε is realized from α , which is evaluated at each point of the computational mesh. Hence, a large majority of points needs to be concentrated around Γ , i.e., in the region where the representation of the level-set function is critical for the quality of the reconstruction. Coupling this approach with an automatic mesh generation procedure handles this dependency. This utility is presented in Sec. 2.2. An illustration is provided by the meshes drawn in Figs. 2.3, related to the reconstructions found in Figs. 2.2

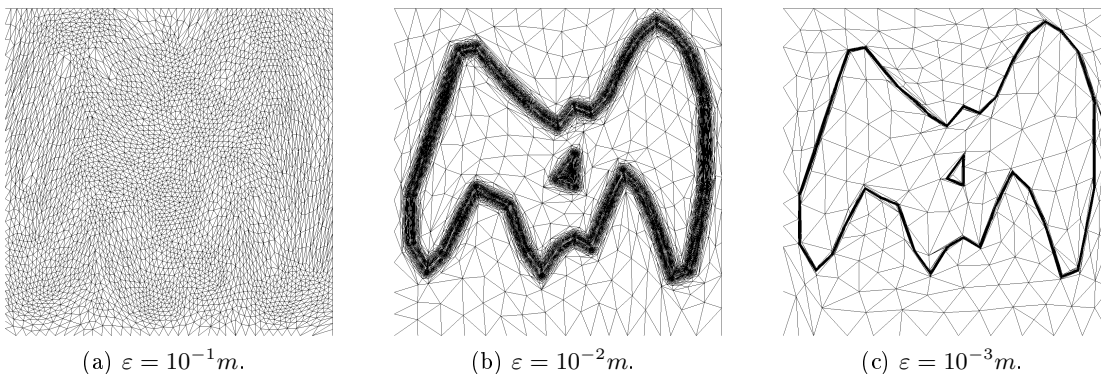


Figure 2.3: Computational meshes related to Figs. 2.2, for various ε .

The illustrations presented in this Subsec. highlight the importance of ε and the tight relation established between it and the meshing. This now needs to be regarded with a focus on the geometries simulated. A first observation concerns the size of the computational mesh. The number of mesh points required in the transition around Γ is constant, while the area of this region decreases. To conserve the resolution of the background mesh while decreasing ε , the number of points in the computational mesh thus needs to be increased. This behavior is confirmed by the meshes presented in Figs. 2.3a-2.3c. The number of mesh points is constant in those meshes, and decreasing ε reduces largely the size of cells far from Γ . Consequently, a very thin description of the geometries may lead to a prohibitive computational effort. A second remark concerns the choice of ε in comparison with the size of the objects to represent. When small geometries need to be represented, using a too big ε and its corresponding mesh, the minimal size of cells can be too important to allow a good capture of the thin effects. An example is presented for a modification of the geometry used in this Subsec., as a little hole is added in the object. The reconstruction is realized using $\varepsilon = 10^{-1}m$ in Fig. 2.4a and

with $\varepsilon = 10^{-2}m$ in Fig. 2.4b. The results obtained with the larger ε show a perturbation of H_ε around the hole, but no hole can be detected using the isoline $H_\varepsilon = 1/2$. Using a smaller ε and thus a thinner mesh, the hole is perceived. The same type of error can be observed for small variations at the surface of immersed objects or for sharp angles. The level-set tends to blur the interface, and the isoline $H_\varepsilon = 1/2$ can finally be closer from an "averaged" value of H_ε than from the real interface. These observations remain valid for any ω . The choice of ε will always be an important step in the reconstruction of a geometry in ICI-tech.

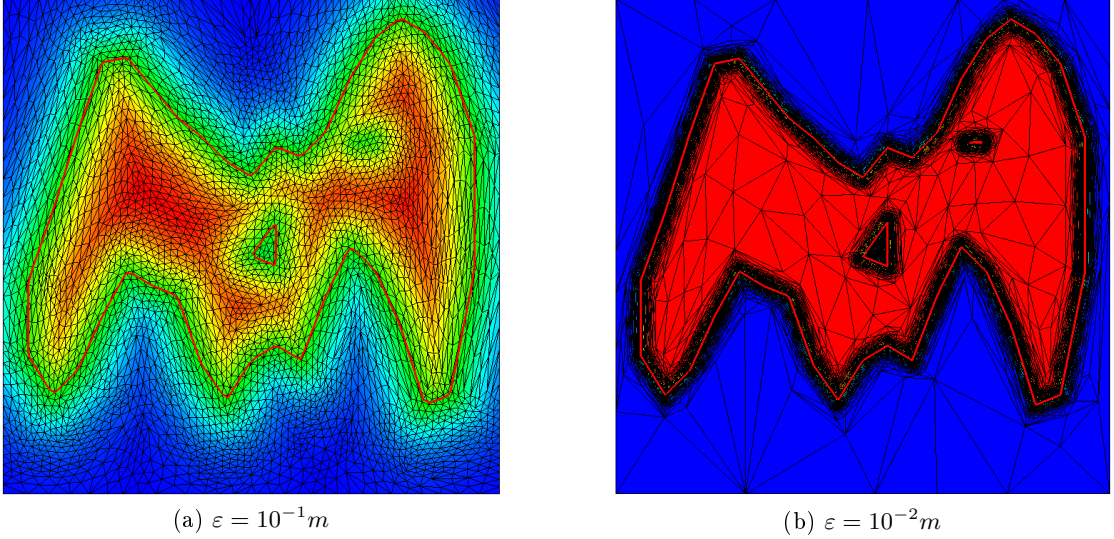


Figure 2.4: Reconstruction of modified object with mesh, H_ε and isoline $H_\varepsilon = 1/2$, for various ε .

2.1.2 Immersion of objects represented by a set of elements

The reconstruction of immersed objects relies on ω . A lot of options are available, as the only requirement is the definition of a frontier Γ splitting the space between interior and exterior. The simplest technique relies on α being a function of the space, returning a scalar field. This approach gives easily H_ε but is inapplicable to complex geometries. FOWTs fall in the second category, and thus must be represented by different manners. A common approach is to discretize the surface of the FOWT Γ . A set of elements, e.g., NURBS or facets, is connected to form a mesh, can be used to characterize ω . The definition of each element determines its orientation. Thanks to a proper positioning of the elements, an interior can be obtained. The distance α is then deduced from it.

The determination of α is necessary to get H_ε . As a scalar field is required, this evaluation needs to be performed at each point x of the computational mesh. When α is defined using a function of the space $f(x)$, H_ε is easily determined, as $H_\varepsilon(\alpha) = H_\varepsilon(f(x))$. But when ω is defined using a set of elements, the evaluation of α must be done in two steps. At first, the distance $|\alpha|$ has to be measured. The closest element among the set has to be found and selected, for each point x . This forces the evaluation of many distances if no optimization is conducted, which causes the computational effort to increase very rapidly with the number of points in the domain. On a second time, the sign of α has to be determined. This step is almost immediate, thanks to the orientation of the closest element. In particular situations, e.g., when considering a mesh cloud, no information on the sign of alpha can be obtained from the closest element, here point. The determination of the surface's orientation becomes much trickier. Neighborhoods around the closest point need to be considered, which brings considerable numerical challenges.

A method is proposed in [Santoso \(2018\)](#).

When a set of elements defines ω , the detection of the closest element must be done for each point of the domain mesh. The computational burden linked to the number of distances evaluated is accentuated by the fact that, for some points, several candidates are selected as closest. Their orientation may be different, which can imply for the computational point a switch between interior or exterior. An illustration is presented for the distance to a set of oriented elements, here segments, when such a situation is observed. Two segments are considered: B, bold horizontal blue, and R, bold inclined red, both featuring their normal pointing towards the exterior. From the point of view of a single segment, its interior is defined by the hatched semi-plan positioned opposite of the normal (slash direction for B, backslash one for R). The double-hatched and un-hatched regions explicitly determine respectively an interior and an exterior for both B and R. On the contrary, single-hatched regions present an ambiguity. It is necessary to choose between interior and exterior without falling into default choices. The first criterion can be the distance to B or R, considering that the closest element provides the most interesting information. However, even with this segregation, two areas, colored in green, remain undetermined. For a point of the computational mesh located in these regions, the distance to B is identical to the distance to R. To finally make a choice, the quality of the projection onto the elements can be evaluated, and the most important one corresponds to the most representative element, thus allowing to segregate between interior or exterior.

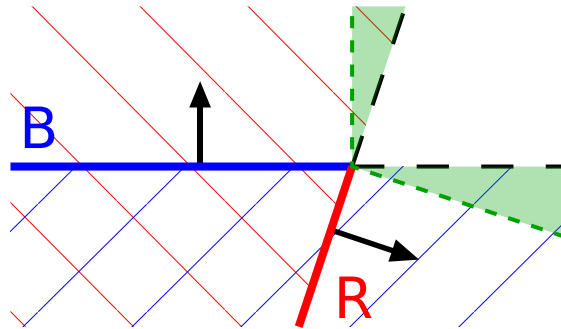


Figure 2.5: Illustration of the distance to a set of two segments B and R as a function of a point position in the plan. Hatched: interiors defined by each segment. Green, colored: Regions where the quality of projection is required.

The definition of the immersion procedure considers an object to reconstruct ω which can be composed of a union of separate geometries. The level-set function representing ω needs to account for all of these objects, which requires the construction of a signed-distance α . Some collisions can be found between these geometries, which need to be accounted for during the definition of α . A simple illustration can be made with a naive schema of a bus drawn in [Fig. 2.6](#): a rectangle represents the body of the bus, while tires are modeled thanks to small cylinders. The level-set function Φ_ε of ω needs to account only for Γ , and not for the frontiers of the immersed shapes. The definition of Φ_ε thus needs to exclude the dashed boundaries in [Fig. 2.6](#). This is done simply through the evaluation of several signed-distances, one for each geometry composing ω . During the construction of α , the highest value returned by the distance computations is favored, i.e., the innermost position found by the union of geometries. This paradigm enables to define ω as a union. Other options, such as intersections or subtractions could be considered, but have not been implemented for the moment.

The procedure detailed above is not complicated, but can rapidly become a burden



Figure 2.6: Illustration of an object formed by the union of distinct geometries. Plain: isoline $\Phi_\epsilon = 0$ defining Γ . Dashed: Frontiers of the geometries excluded from Γ .

when the meshes provided come with defaults. If the orientation of either B or R is wrong, the interior defined will be completely unrealistic. However, with the meshing engines used nowadays, those serious mistakes are barely encountered. More common are the joint issues, when the point presumably shared by B and R is slightly deflected for one of those elements. This leads to major errors, e.g., if R's top point is positioned slightly above B's one. Following the algorithm presented in the previous paragraph, the green-colored region located on top of Fig. 2.5 is consequently considered inside the object. Alignment errors occur too, when mesh cells are slightly deflected, which creates asperities on presumably smooth surfaces. All these sources of error impose the set-up of securities, tuned for each object reconstructed, which complicate the immersion procedure.

Moreover, the computational complexity brought by this methodology becomes considerable when a M -elements mesh is immersed in a N -points computational mesh. The complexity of the level-set approach as presented in the previous paragraphs is $N \times M$, mostly due to the distances to evaluate. The determination of the interior being essentially composed of if-loops, its cost is reduced. The optimization of the immersion procedure can be reduced in two ways. The first one is to parallelize the computations, which allows speeding-up the evaluation of α by measuring distances on several cores. The set of elements can also be split, allowing each processor to load only the elements meaningful in its computational area. This enables to consider a huge set of elements, which could hardly be loaded by a single processor, and therefore should not be duplicated on each core. The opportunities offered by the parallelism are interesting if either M or N is reasonably increased, and are available in ICI-tech. Yet, for extremely expensive cases, e.g., when both M and N rocket, the complexity becomes too important even for a highly-parallelized platform. The computational costs can then be reduced through the construction of a tree structure. The implementation of a tree, named "octree" for 3D applications, is detailed in Subsec. 2.1.3. Note that the usage of the name octree is generalized in the following to describe the tree structure implemented.

2.1.3 Tree-optimization of the reconstruction procedure

The mesh immersion procedure comes at a high computational cost whenever the mesh immersed is composed of a huge number of elements, as stated in Subsec. 2.1.2. As α still has to be determined for each point of the computational mesh, optimization can essentially be conducted on the evaluation of α for a given point. This was realized through the implementation of a tree-structure. These algorithmic constructions are commonly used to reduce the computational complexity in numerical codes. Tree-structures have various fields of application, and are commonly used for mesh generation, e.g., by Ahrabi et al. (2017) or Kudela et al. (2016).

This study proposes to use tree-structures for the optimization of the reconstruction process. The elements composing the object to represent are segregated depending on their position. For a given point, the tree-browsing then allows selecting the closest el-

elements of the immersed mesh, and distances can be evaluated on them only. Both the construction of the tree and the determination of the closest element is realized using bounding boxes, to reduce the computational expenses. Different definitions are possible for the bounding boxes, which are detailed by [Ding et al. \(2004\)](#). The implementation presented here uses axis-aligned bounding boxes, the least expensive method computationally, which is efficient for small, convex elements.

The definition of the tree-structure allows largely reducing the number of distances to compute during the evaluation of α . The complexity of the algorithm is brought down, within a theoretical minimum of $N \times \log(M)$. The addition of the tree structure into the mesh immersion procedure can be split into two steps. The first one consists in the construction of the octree, while the second one focuses on the evaluation of α . Both are detailed in the following.

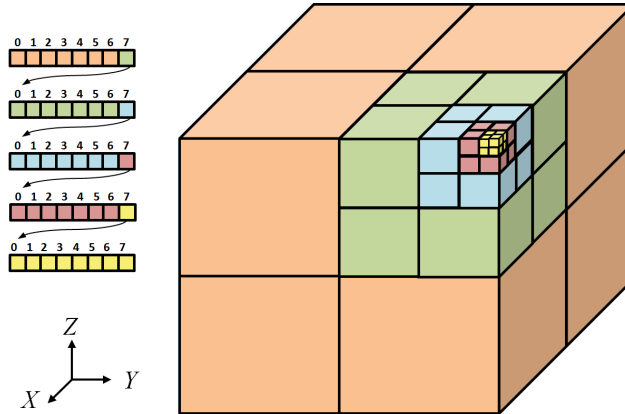


Figure 2.7: Octree refinement procedure, through recursive divisions of the computational domain. On the left is highlighted the refinement of a tree element into 8 new elements. The cube on the right presents the geometrical positions of the octree elements.

Construction of the tree structure

This data storage concept is built recursively from the set of M elements defining the immersed geometry, independently from the computational mesh. The bounding box of the whole set of elements is constructed, and all the elements are allocated to the domain created. This composes the initial tree. Refinement steps are then processed, and the tree domain considered is split into two segments of equal length along each dimension. Subdomains, or children, are generated, as presented in Fig. [2.7](#) in 3D. Note that the name octree comes from the 3D behavior of the tree, which splits each domain into 8 subdomains during the refinement procedure. After a refinement step, the elements must not be contained in the initial octree domain, but are allocated to every child they intersect. This allows elements to be duplicated if they intersect several children. After a refinement step, each child is examined with emphasis on the numbers of elements it contains. If a subdomain remains empty, i.e., no elements intersect it, it is immediately deleted. If too many elements are found in this child, the refinement procedure is repeated in this particular subdomain. The recursivity continues until, either an acceptable number of elements is obtained in the deepest subdomains (leaf), or the maximal depth of the octree is reached.

The number of elements allowed per leaf N_{max} and the depth of the octree needs to be determined for each set. The objective of this procedure is to split the elements between octree leaves. The best efficiency of the tree structure is obtained with a small

N_{max} , even if the number of leaves is considerable. The idea behind this choice is that the number of distance to elements is reduced when small leaves are considered. The filters brought on the evaluation of the distance perform better in this configuration. The depth of the octree should be chosen correspondingly, as smaller leaves are acquired from more refinements. At the same time, N_{max} needs to account for the geometry of the set considered. The concentration of elements in certain areas limits the potential of the octree, e.g., when more than N_{max} elements are connected to a single point. The octree leaves located in this region will reach the maximal depth of the octree, and the elements can be duplicated onto a huge number of neighbor leaves. Consequently, the definition of the octree depth can be done from the relative size of the elements compared to the bounding box on which the set is located. The determination of N_{max} can then be realized from the concentration of cells expected in those areas.

An illustration of the methodology generating a tree structure of maximum depth 4 is drawn in Figs. 2.8, from the example of a set of elements presented in Fig. 2.8a. For a matter of clarity, the initial octree domain is defined using the black square, and not from the bounding box of the elements. The rest of the procedure remains identical, and Fig. 2.8b presents the octree obtained after the first refinement. The subdomain containing a number of elements inferior to the one required is not refined, and the result obtained after a second refinement is shown in Fig. 2.8c. The empty subdomains are deleted. After 2 more refinements, the final octree is obtained. It is presented in Fig. 2.9a. The cells drawn in red correspond to the octree leaves.

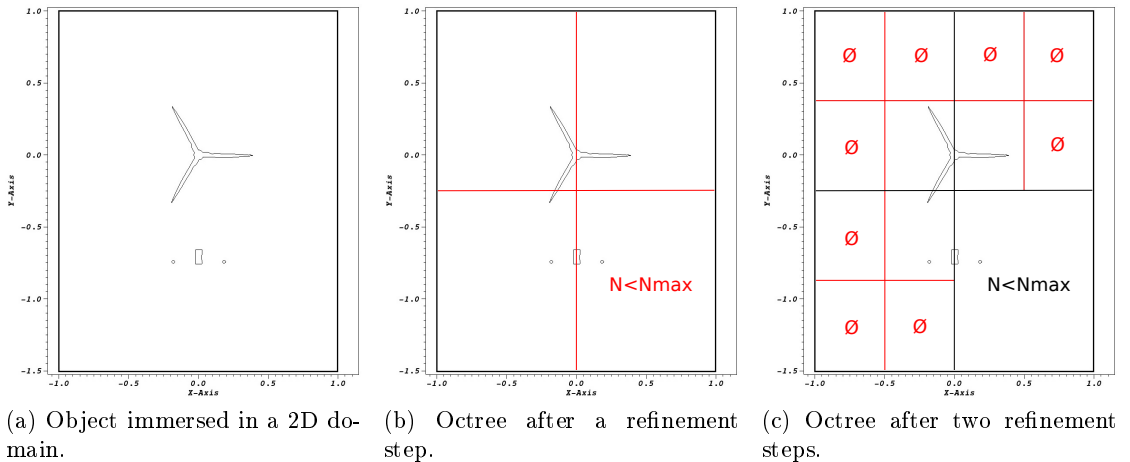


Figure 2.8: Refinement for the construction of an octree.

The repartition of elements into the children is handled using axis-aligned bounding boxes. This definition consists in determining the maximal and minimal positions of each element along each axis, which offer different advantages. First of all, these boxes are very easy to determine, both computationally speaking and considering data access. It also allows reducing the computational effort for the determination of the intersections with the octree domains, as boxes and tree domains are oriented along the same axis. Finally, this choice enables to generalize the octree to very different usage, from, e.g., fibers to 3D facets used to define surface meshes. The main drawback brought by these bounding boxes lies in the intersections with the octree domains when huge and/or highly-stretched elements are considered. This leads to multiplying duplicated elements in leaves they may not intersect. Another limitation is found when very long elements are positioned, proportionally to the size of the computational domain, as again the elements may be highly duplicated.

This strategy characterizes the "tree" class used in object-oriented programming. It is composed of a spatial domain and, either a set of elements contained inside the class, or pointers towards its children, which are obtained after refinement. The final tree has as entry point an object of type "tree", which contains the bounding box of the geometry to reconstruct as spatial domain and pointers towards each of its children. The elements composing the immersed geometry are split between all the leaves of the octree, hence none are contained at the depth zero if children exist. Every child contains its spatial domain, and either a set of elements or pointers towards its children. This pattern is reproduced recursively.

Evaluation of the signed-distance using the tree structure

Once the tree structure is built, the evaluation of α can start. Two major improvements are observed. First of all, for a given point of the computational mesh, the octree enables to select the closest elements from the point considered, i.e., the "best candidates" for the determination of α . To that extent, the notion of leaves is essential. The second advantage is brought by the bounding boxes. During the process of selecting the best candidates, no exact evaluation of the distance from the current mesh point to any element of the immersed geometry is performed. An upper bound of this distance, obtained thanks to the bounding box of the element, is preferred. The computational cost of an exact evaluation can indeed be very expensive to compute when the encapsulated element is complex. This choice allows computing a very limited number of exact distances during the evaluation of α , especially when small elements compose the object to reconstruct. As already mentioned, it also guarantees the versatility of the mesh immersion procedure, as any geometry can be immersed in the octree using its bounding box.

The methodology used for the evaluation of α begins with the determination of the closest leaf L_c from the computational point P considered. This is achieved using a recursive browsing of the octree. For an object of type "tree", the presence of geometry elements is tested. If some are found, the minimal distance from the spatial domain of the current tree object to P is measured. If none exists, then the tree object contains children, on which the same procedure is recursively applied. Using this function at the entry point of the octree allows finding the closest leaf from the point considered. The application of this approach on the test case introduced in Figs. 2.8 is detailed in Figs. 2.9. The octree presented in Fig. 2.9a is returned at the end of the refinement process. For the point P considered on Fig. 2.9b, the leaf drawn in green is the closest one.

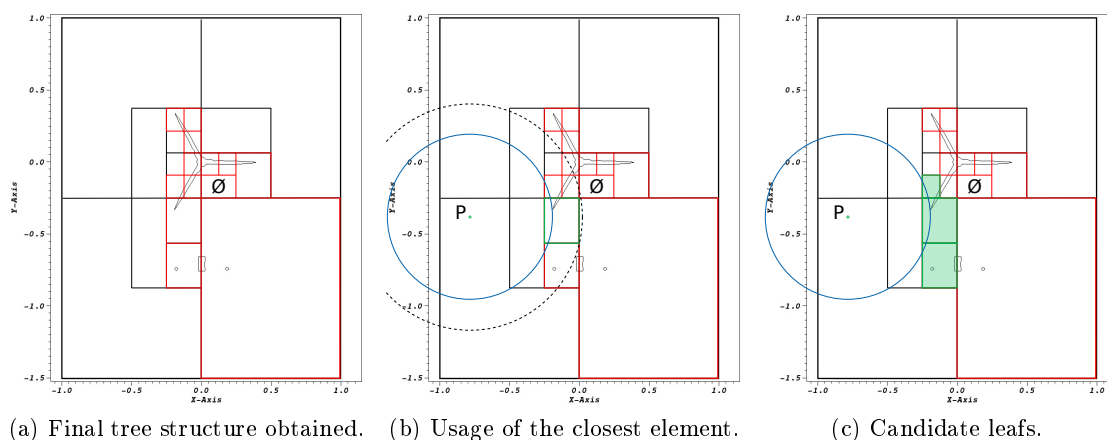


Figure 2.9: Determination of the closest element for a given computational point.

Once the closest leaf L_c has been determined, there is still uncertainty about the

leaf in which is located the closest element. Theoretically, it can be contained in any of the leaves intersecting a disc of center P and of radius the maximal distance to the spatial domain of the leaf. Graphically, it corresponds to the dashed black circle found on Fig. 2.9b, which intersect 7 leaves. However, using a browsing of the elements contained inside L_c , the size of this circle can be reduced. The maximal distance to the bounding boxes of the elements contained in L_c is evaluated, and the minimal one d_c is selected. Considering d_c , as it is initially defined as a maximal distance to a bounding box, at least one element has a distance to P which is smaller than d_c . Consequently, the closest element is located in a disc of center P and of radius d_c , which corresponds to the plain blue circle drawn in Fig. 2.9c. The number of leaves potentially containing the closest element is reduced. In the example provided on the illustrations, only 3 leaves are intersected. They are colored in green. The elements contained in the leaves selected are browsed, and the minimal distance between the mesh point and their bounding box is measured. When the value obtained is inferior to d_c , the real distance from the mesh point to the element is measured. The closest element, and then α , are deduced from the real distances evaluated.

The procedure detailed above highlights the interdependence between the tree structure and the localization of the elements representing ω . The octree is built from the positioning of the different elements, and relies heavily on it to evaluate the distances to mesh points. Displacing some elements may move them from octree leaves to others. In these conditions, the octree needs to be updated to guarantee the quality of the distance evaluation procedure. After one iteration, the elements representing the geometry immersed are examined, to identify if they moved from octree leaves to others. Under particular situations, when the displacements of the elements have a limited impact on the structure of the octree, basic modifications focusing essentially on the assignment of elements to children can be sufficient. But when the whole tree structure is modified, the simplest approach is to reconstruct the octree. The computational costs induced by the reconstruction procedure have to be taken into account, even if the results presented in Sec. 2.4 tend to show that they are quasi negligible.

2.2 Mesh adaptation

Mesh adaptation is essential for the mesh immersion procedure detailed in Sec. 2.1, but is also a self-standing software unit of ICI-tech. The mesh adaptation intends to increase the precision of the simulations through modifications of the computational mesh. A focus is placed on areas of interest, which can be prescribed or automatically generated. They can be defined either *a priori*, from the specification of a physical problem, or *a posteriori*, from the results of simulations.

Several mesh adaptation strategies can be found in the literature. The h-adaptation strategy consists in local refinements of the computational mesh, with additions of mesh points. Smaller cells are designed, e.g., through successive divisions of cells along each dimension, as in Wackers et al. (2014). A mesh of higher resolution is found in the zones of interest, which improves the results of the simulations. The r-adaptation keeps an identical number of point, but tends to move points towards the areas of interest of the mesh, as in Béal et al. (2001). This method is lighter computationally speaking, but does not interfere with the number of points in the mesh, which has to be correctly chosen since the beginning of the simulations. A last approach is known as p-refinement. Instead of displacing the mesh points, it focuses on the order of interpolation, which is enriched in areas of interest. This strategy can be combined with the h-adaptation, as in Ahraabi et al. (2017), in order to capture very small effects without using extremely thin meshes.

The different references proposed to illustrate the meshing strategies refine the mesh identically along each dimension, thus generating isotropic meshes. Another approach evaluates the requirement to refine the mesh along each dimension, and deform the mesh correspondingly. The mesh obtained after adaptation is anisotropic. Very small mesh sizes are observed along the dimensions where events occur, while large lengths are found when the domain is uniform locally along straight lines. Anisotropic meshes have attracted attention, as it allows to largely reduce the number of points in the discretization of computational domains. This meshing technique is used, e.g., by [Formaggia and Perotto \(2001\)](#) or [Frey and Alauzet \(2005\)](#).

In this thesis, an anisotropic *a posteriori* h-adaptation strategy, proposed by [Coupez \(2011\)](#), is used in a P1-Finite Element paradigm. From the results of the simulation, this method builds an error estimator, which is then post-treated to deform the mesh. The construction of the error estimator is presented in Subsec. [2.2.2](#), while the adaptation procedure itself is detailed in Subsec. [2.2.1](#). A more comprehensive description of the mesh adaptation procedure, and of its usage, is described in Subsec. [2.2.3](#).

2.2.1 Anisotropic mesh adaptation with a metric field

The mesh adaptation procedure used in this thesis takes advantage of a Riemannian metric field \mathcal{M} , built on the computational domain. At each point of the domain's mesh, a base related to the local mesh size in the Euclidean space can be considered. The metric operates as a distorting mirror: using a transition to a Riemann space, it deforms the perception of the distances observed in the Euclidean base. In practice, the Riemann space comes with a scalar product that is different from the classical Euclidean one. The scalar products are presented in Eq. [\(2.4\)](#) for two vectors v and w of the Euclidean space.

$$\text{Euclidean} \quad \langle v, w \rangle = {}^t v w \quad (2.4a)$$

$$\text{Riemannian} \quad \langle v, w \rangle_{\mathcal{M}} = {}^t v \mathcal{M} w \quad (2.4b)$$

The evaluation of the distances in both spaces can be deduced from these expressions, and are given in Eq. [\(2.5\)](#).

$$\text{Euclidean} \quad \|v\| = \sqrt{{}^t v v} \quad (2.5a)$$

$$\text{Riemannian} \quad \|v\|_{\mathcal{M}} = \sqrt{{}^t v \mathcal{M} v} \quad (2.5b)$$

A Riemann base is obtained, which rescales the mesh locally, from the eigenvalues λ_1 and λ_2 of \mathcal{M} . An illustration is presented in Fig. [2.10](#) for the perception in the Riemann space of a reference element. The same behavior can be found in a 3D context. However, even if the reference is rescaled from the Riemann point of view, the mesh in the Euclidean space is not modified.

The mesh adaptation procedure used in ICI-tech intends to use very thin mesh cells in the regions of interest. As some phenomena are developed with preferential directions, anisotropic meshing is a good opportunity to reduce the computational effort while keeping a similar precision. The mesh adaptation procedure relies heavily on Riemannian distance evaluation. A metric field \mathcal{M} is built to account for the variations observed in the domain. At each point of the computational mesh, the objective of the mesh adaptation is to visualize a length of 1 for every edge of the mesh in the Riemannian space. From an analogy with Fig. [2.10](#), the transformation drawn in Fig. [2.11](#) is obtained. To

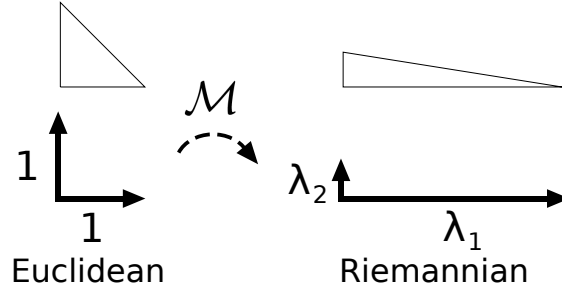


Figure 2.10: Deformation of a reference element using the metric field \mathcal{M} .

see an orthonormal base in the Riemannian space, a deformed base is required in the Euclidean space. The mesh cell obtained is anisotropic, small in the horizontal direction, large in the vertical one. These characteristics are typical of, e.g., horizontally-oriented flows.

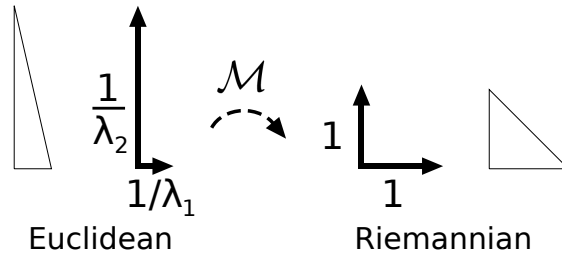


Figure 2.11: Riemannian distance evaluation for the anisotropic mesh adaptation.

The mesh adaptation tools apply this framework to every edge of the computational mesh. At a given point X_i of the mesh, each edge connected to X_i needs to be considered. This condition would write as in Eq. (2.6), with $\Gamma(i)$ the set of points connected to X_i , $X_{ij} = X_j - X_i$ for $j \in \Gamma(i)$ and \mathcal{M} a metric to determine.

$$\forall j \in \Gamma(i), \quad \langle X_{ij}, X_{ij} \rangle_{\mathcal{M}} = 1 \quad (2.6)$$

As X_{ij} is split between X_i and X_j , and because the metric field is built at the points of the computational mesh, this condition is formulated as a sum over the neighborhood of each point. The condition obtained is written in Eq. (2.7), with $|\Gamma(i)|$ number of points in $\Gamma(i)$.

$$\sum_{j \in \Gamma(i)} \langle X_{ij}, X_{ij} \rangle_{\mathcal{M}_i} = |\Gamma(i)| \quad (2.7)$$

By deriving this equation, an expression can be obtained for the metric at each node \mathcal{M}_i , presented in Eq. (2.8) for d dimension of the domain considered and \otimes tensor product operator.

$$\mathcal{M}_i = \frac{1}{d|\Gamma(i)|} \left(\sum_{j \in \Gamma(i)} X_{ij} \otimes X_{ij} \right)^{-1} \quad (2.8)$$

An exact solution of this equation is expensive to determine, as a matrix inversion is required for each point of the computational mesh. The value of \mathcal{M} can be determined at a lower computational cost through the resolution of an optimization problem, as in Eq. (2.9).

$$\mathcal{M}_i = \arg \min_{\mathcal{M}} \left(\sum_{j \in \Gamma(i)} (\langle X_{ij}, X_{ij} \rangle_{\mathcal{M}} - 1)^2 \right) \quad (2.9)$$

The theory developed in [Coupez \(2011\)](#) precises that the Eqs. [\(2.8\)](#) and [\(2.9\)](#) are equivalent when at least one non-degenerate element is connected to X_i . This condition is almost trivial, as it can be fulfilled despite a very low mesh regularity. These different steps detailed in this Subsec. allow, for any point X_i of a non-degenerated mesh, to build a metric field measuring any edge connected to X_i as unity in the Riemannian space. The formulation obtained depends only on the topology of the mesh, but does not account for other parameters, e.g. fluid flows. The coming Subsec. presents the construction of the metric that minimizes the interpolation error of a solution field u .

2.2.2 Metric construction using *a posteriori* error estimator

The adaptation of mesh cells as presented previously rely on the definition of areas of interest. These areas can be user-defined, when the evolution of the physical problem in space is small. A more polyvalent approach uses error estimators, which evaluate the error produced in the computational domain. Error estimators can be defined *a priori*, from the interpolation error produced on a mesh regarding a reference solution, strategy followed by, e.g., [Formaggia and Perotto \(2001\)](#). This implies to possess information about the output required, which is not evident in the context of complex flows. Another approach considers *a posteriori* error estimators, which are built from the outputs of simulations. The interpolation error produced by the resolution is evaluated. Some of these methods use a Hessian of a scalar field, see, e.g., [Frey and Alauzet \(2005\)](#). Others prefer a recovery gradient from the given scalar field, such as proposed by [Almeida et al. \(2000\)](#).

The mesh adaptation process presented here is based on the equidistribution of the error in all the computational domain. A first, intuitive parallel with unity vectors in Riemannian space can be drawn. An *a posteriori* error estimator is built at each node of the computational mesh. The mesh is adapted in areas of interest defined from this estimator: high-error areas need to be refined, while low error regions can be coarsened. The evolution of the error is examined along the edges of the mesh, where points can be inserted if the values found are too large. A metric field is built from edge errors in order to deform the mesh, always with an error homogenization goal. Scaling factors are defined for each dimension, as well as the metric field, from which the anisotropic characteristic of the mesh arises.

The adaptation depends on the construction of the error estimator. A generic introduction of the mesh adaptation procedure, realized in [Coupez \(2011\)](#), defines the error estimator e as a function of a scalar field u . This definition arises from C ea's lemma, which bounds the interpolation error, as written in Eq. [\(2.10\)](#). The value of e and u at the point X_i are expressed, respectively, e_i and U_i .

$$\|e_i\| < c\|u - U_i\| \quad (2.10)$$

In the context of P1 meshes, u is interpolated using first-order polynomials. The scalar field obtained is written u_h . The gradient of the u_h , written ∇u_h , is discontinuous, constant per mesh cell only. However, the projection of the gradient on an edge remains continuous. An evaluation of the error produced along each edge is proposed in Eq. [\(2.11\)](#). This formula paves the way for the construction of a Cl ement interpolant, as explained by [Carstensen \(2006\)](#), and uses the Hessian matrix, which is expensive to evaluate. This can be avoided through the definition of a recovery gradient, built from a projection of ∇u on the edges of the mesh. This correction is dedicated to the definition of an error estimator avoiding the construction of a Hessian.

$$|(\nabla u_h - \nabla u(X_i)) \cdot X_{ij}| \leq \max_{Y \in [X_i, X_j]} |\nabla^2 u(Y) X_{ij} \cdot X_{ij}| \quad (2.11)$$

The first step consists in the reconstruction of a P1 gradient G of u_h , defined at each point of the mesh. The error estimator expressed on an edge, presented in Eq. (2.13), is deduced from the reconstructed gradient, given in Eq. (2.12).

$$G_i = d\mathcal{M}_i U_i \quad (2.12)$$

$$e_{ij} = |G_{ij} \cdot X_{ij}| \quad (2.13)$$

This formulation of the error is valid only on a complete edge. A generalization onto portions of edges is possible using the P1 interpolation characteristics, following the expression proposed in Eq. (2.14). A maximization of the error produced on an edge can be performed at the same time.

$$e_{ij}(s) = |G_{ij}(s) \cdot sX_{ij}| \leq s^2 e_{ij}, \quad s \in \mathbb{R}^+ \quad (2.14)$$

The whole error ϵ generated on an element \mathcal{K} is then evaluated, from the difference between u and u_h . The different equations proposed in this Subsec. translate to an estimation of the error produced on an element, as a function of e_{ij} and of the area/volume of \mathcal{K} , written $|\mathcal{K}|$. The expressions obtained for 2D and 3D applications are presented in Eq. (2.15).

$$\epsilon_{\mathcal{K}, 2D} = \frac{1}{8} |\mathcal{K}| \sum_{i,j=1,\dots,d} e_{ij} \quad (2.15a)$$

$$\epsilon_{\mathcal{K}, 3D} = \frac{1}{30} |\mathcal{K}| \sum_{i,j=1,\dots,d} e_{ij} \quad (2.15b)$$

The equation obtained states that controlling e_{ij} for all the edges of an element implies the control of the interpolation error realized on this element. This is highly compatible with the metric formulation presented in Subsec. 2.2.1. The metric \mathcal{M}_i can be built using the error estimator e_{ij} . Whenever the Riemann scalar product is used, the target of the adaptation is to obtain an error of 1 along each edge, i.e., to equidistribute the error. A well-adapted mesh consequently positions nodes regarding the error, and not the topology or mesh size. Moreover, the distribution of the error in all the domain allows generating errors of similar orders of magnitude for each cell. The metric used in this formulation is presented in Eq. (2.16).

$$\mathcal{M}_i = \left(\frac{1}{d} \sum_{j \in \Gamma(i)} s_{ij}^2 X_{ij} \otimes X_{ij} \right)^{-1}$$

$$\text{with } \begin{cases} s_{ij} = \left(\frac{\lambda}{e_{ij}} \right)^{-1} \\ \lambda = \left(\frac{\sum_i \sum_{j \in \Gamma(i)} e_{ij}^{\frac{p}{p+2}}}{N_n} \right)^{\frac{p+2}{p}} \end{cases} \quad (2.16)$$

The parameter N_n corresponds to the maximal number of nodes allowed in the mesh. This correction provides a safeguard to the mesh adaptation procedure. In the case when the error rockets locally, the adaptation tends to refine the mesh towards a very thin network. This may require a very important number of nodes in the mesh, which may be unaffordable. Thanks to the limitation of the number of nodes, the computational effort can be reduced. However, this correction prevents the control of the error as detailed previously. The quality of the mesh generated thus needs to be examined with careful attention. Limiting the number of nodes in the mesh does not prevent the adaptation to diving into the error-rocketing region. Adding a maximal error tends to limit the size of mesh cells, and is a way to avoid infinitely small cells. Similarly, a null error over an edge

e_{ij} tends to resize this edge towards an infinite one. In order to tackle these limitations, the corrections presented in Eq. (2.17) are applied to e_{ij} , with ϵ_{\min} and ϵ_{\max} parameters ruling respectively the minimal and maximal values allowed for e_{ij} .

$$\begin{cases} e_{ij} = \max \left(|G_{ij} \cdot X_{ij}|, \epsilon_{\min} |X_{ij}|^2 \right) \\ e_{ij} = \min \left(|G_{ij} \cdot X_{ij}|, \epsilon_{\max} |X_{ij}|^2 \right) \end{cases} \quad (2.17)$$

During the adaptation procedure, the mesh is either refined or derefined locally. Depending on the value of s_{ij} , points can be inserted or suppressed. As the stretching coefficients s_{ij} are evaluated along the edges, the point addition process is simplified. The edge concerned is split, and the elements adjacent to this edge are split too, in order to integrate the newly created point. The node suppression process applied to a point X_i considers its neighborhood $\Gamma(i)$. A point X_j of $\Gamma(i)$ is selected, and the new mesh cells are created through a connection of the different points of $\Gamma(i)$ to X_j . Illustrations of these procedures are presented in Fig. 2.12

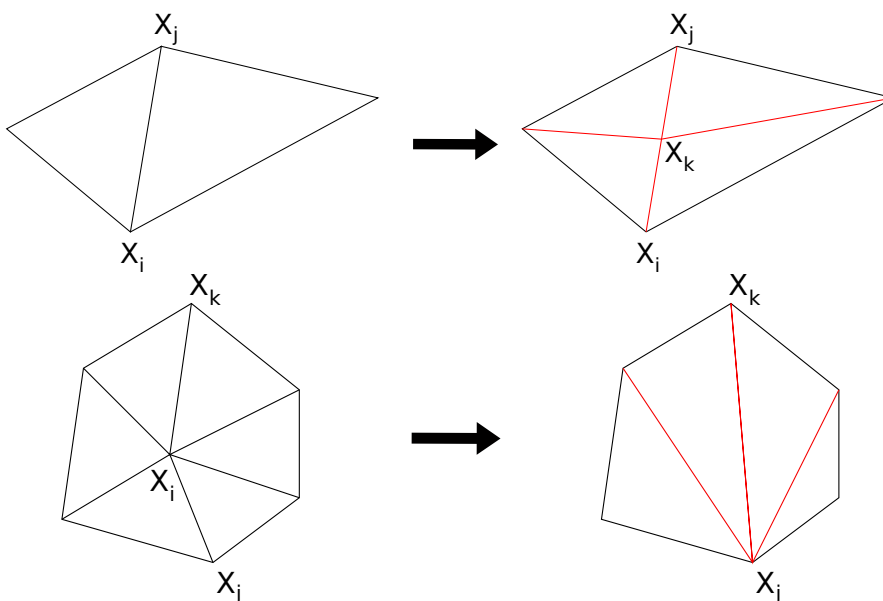


Figure 2.12: Procedure for point addition (up) and removal (down).

2.2.3 Mesh adaptation in practice

The theory presented in the previous Subsecs. details the adaptation as introduced by Coupez (2011). Mathematically, the metric formulation shows the potential to adapt a mesh through a single execution of the process presented above. However, this ideal view is contradicted by computational considerations. Corrections brought to e_{ij} bring new constraints, with a maximal number of nodes N_n allowed, or through addition and removal of mesh nodes. Hence, iterations are required to get a fully adapted mesh, automatically generated from a potentially coarse initial mesh. A huge share of mesh nodes tends to be gathered in the areas of interest, defined using a combination of the level-set functions obtained from mesh immersion. More generally, any scalar field derivable twice can be used for the adaptation. As an example, when the resolution of the NS equations is achieved, the velocities found in the flow can be taken into account too.

The mesh adaptation procedure is used to generate meshes adapted on the different level-set functions represented in the domain. This operates as a prelude to NS resolutions, which requires adapted meshes to guarantee the convergence of the schemes. The adaptation procedure starts from a potentially coarse initial mesh. The immersion of the

different geometries represented in the domain is performed. Level-set functions of poor precision are constructed, due to the unadapted mesh. The adaptation is realized using an error estimator established from the level-set functions, as presented previously. The inaccurate reconstruction of the phases generates high gradients in the level-set functions, which are identified by the adaptation procedure as areas to refine. A new reconstruction is performed, and the same procedure is repeated. After several iterations, an adapted, anisotropic mesh is obtained.

An example of adapted meshes generated from a level-set function was presented in Figs. [2.3a](#)-[2.3c](#). These Figs., generated with three different ε , show three levels of precision. The minimal size of mesh cells is fixed to $\varepsilon/10$, which translates to a coarser representation in Fig. [2.3a](#), while Fig. [2.3c](#) provides the best precision in a very thin layer only. This example featured only one level-set function. Yet similar results are obtained when adaptation is performed on several geometries. The different level-set functions just need to be linearly weighted, to form a single scalar field.

This aforementioned example is presented for a sequential application. The reconstruction procedure is integrated into ICI-tech, a massively parallelized software platform. The parallelism consists in the distribution of the computations on a cluster, composed of several processors. The computational mesh is split among the different cores implied in the simulation, which requires several implementations, introduced in the coming Sec.

2.3 Highly-parallelized framework in ICI-tech

The development of ICI-tech relies on a highly-parallelized architecture, which allows deploying it on clusters. The computational domain is partitioned, to take advantage of the important number of cores. The parallel meshing as implemented in ICI-tech is presented in Subsec. [2.3.1](#). To guarantee the efficiency of the parallelism, computational bottlenecks need to be suppressed, or at least reduced. The different utilities composing ICI-tech thus need to be optimized. The parallel implementation of the mesh immersion and mesh adaptation procedures into ICI-tech is presented, respectively, in Subsecs. [2.3.2](#) and [2.3.3](#).

2.3.1 Parallel meshing in ICI-tech

The parallelism relies on a partitioning to separate the computational domain into different areas, allocated to the processors on which the simulation platform is deployed. The computational mesh is split, and the mesh points and cells are allocated to the different cores, without any geographical bias. This technique allows spreading the computational load, each processor getting a sensibly equal proportion of the mesh. Each core contains a local mesh, and the association of all local meshes forms the whole computational mesh. An illustration is proposed in Fig. [2.13](#), where a mesh is colored by the processor simulating locally.

During the deployment on a cluster, the partitioning process separates the mesh points and cells among the different CPUs used. Each point or cell is positioned on a single processor only, with points defined by its coordinates and cells by the identifiers of the points composing it. However, when a point is used to define cells allocated to distinct CPUs, one of these cells is located on a core where the point is not. To get a self-standing representation of this cell and to limit the communications between the processors, this point has to be duplicated. A "ghost" point is created, which is distinguished from the local points of the core. The creation of local points is realized at the interfaces between the areas of different processors. The parallelism implies a rise in the number of points in memory, through the addition of ghost points. This increase directly depends on the

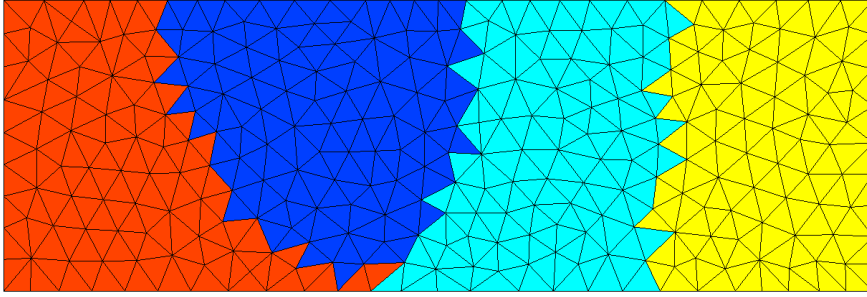


Figure 2.13: Computational mesh colored by the processor on which mesh points are stored.

ratio between the number of points in the computational domain and the number of CPUs on the cluster.

2.3.2 Parallel reconstruction in ICI-tech

The optimization of the mesh immersion procedure is presented in Subsec. [2.1.3](#), for a computational mesh of N points and for a set of elements M . The implementation of an octree enables to reduce the expenses linked to the immersion, without any considerations regarding M or N . The parallelism allows splitting the different meshes, in order to limit N and M through the use of multiple cores.

The limitation of N is done naturally, following a simple partitioning on n_p processors as presented in Subsec. [2.3.1](#). The immersion is done for each computational point through the evaluation of H_ε , without considering the ghosts. Consequently, each processor will need to compute H_ε for approximately N/n_p points, instead of N in a sequential run. This leverage gains importance and influence when N rises.

A complementary approach acts on M . The simplest parallel mesh immersion algorithm would duplicate the set of M elements onto each core, before the reconstruction starts. However, for a given computational point, the mesh immersion procedure as presented in Sec. [2.1](#) needs only the closest element from the set of M . Geographical segregation of the set of elements can be done for each CPU, based on the location of the computational points it contains. This offers the potential to select only the meaningful elements, to move from M to $M_p < M$. This reduction in the load of the set has an impact on the evaluation of H_ε , even if the elements suppressed are mostly located far from the computational point, and are thus filtered by the octree. The other impact of this reduction is the distribution of the set of M elements into sets of smaller size, located in interest areas regarding each processor. This may allow the reconstruction when M is too important to be loaded on a single core, and therefore can not be shared with each CPU of the cluster.

2.3.3 Mesh adaptation in the highly parallelized context of ICI-tech

The mesh adaptation procedure is parallelized with a particular regard on the different areas belonging to each processor. Each CPU refines automatically the mesh in its region, excepted its frontier cells. As previously noted, the points linked to these cells can be duplicated among different processors. As the adaptation realized by each CPU may not be identical, to avoid ambiguities, these cells are not adapted. After a remeshing step, the domain is repartitioned to redistribute the computational loads, as all the processors need to handle a similar number of nodes. During the repartitioning, the frontier cells, which have not been modified, are preferably moved inside a CPU. This allows redimensioning these cells in the next remeshing iterations, which is necessary for a well-adapted

mesh. After several adaptation increments, the different iterations ensure that all the cells have been adapted.

During the remeshing process, some mesh points and cells are moved from a CPU to another one. This step requires communications between the different processors, to transfer the missing points and cells to the receiving CPU.

2.4 Representation of WTs in ICI-tech

This Sec. presents different test cases considering the immersion of FOWTs in ICI-tech. The surface mesh of a small-scaled FOWT is used in the different studies presented hereafter. Successively, the immersion of one FOWT, of 100 FOWTs, and a scalability study are performed.

2.4.1 Reconstruction of a single WT

A first application case considers the immersion of a single, small-scaled FOWT, composed of a WT and its semi-submersible floater of approximative size $2m \times 2m \times 2.5m$. The Figs. 2.14 present the steps followed. The surface mesh of a prototype studied in wind flume by Lacaze et al. (2014), composed of approximately $75K$ oriented triangles, is immersed in a computational mesh of about $120K$ points. A view of this mesh is proposed in Fig. 2.14a. A slice of the mesh obtained at the end of the immersion procedure is presented in Fig. 2.14b. This slice intersects the rotor, and consequently, the high concentration of points affords to guess the position of the blades. The parameter ε used to generate the mesh, to define the width of the adaptation area, is of about $1/1000$ -th of the overall WT size. The order of precision of the immersion is of the centimeter for a full-scale WT, which is appropriated for visualization purposes. The reconstructed WT depicted in Fig. 2.14c looks similar to the surface mesh immersed, even if some imprecisions can be found in challenging regions, such as the tip of blades.

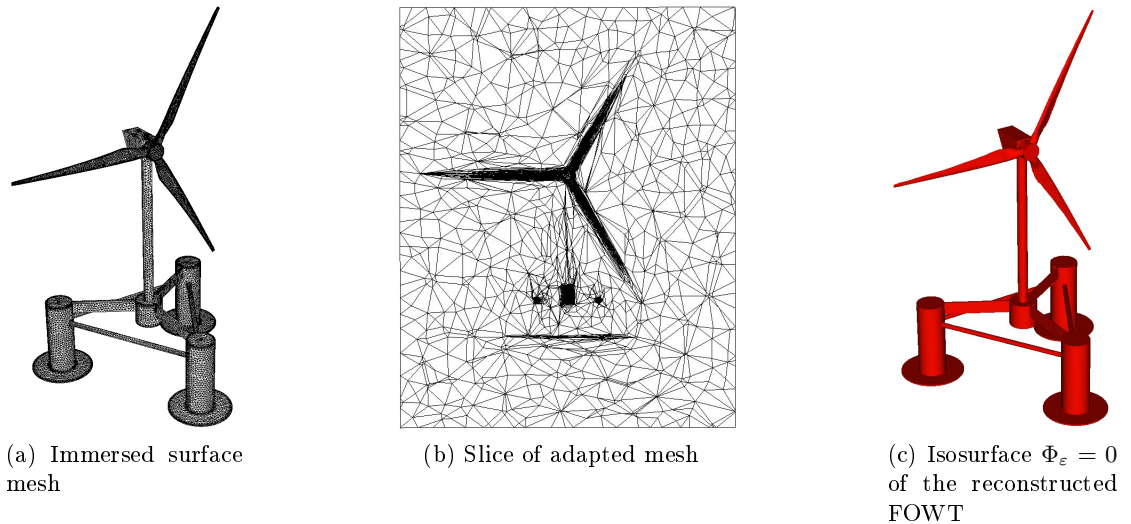


Figure 2.14: Reconstruction of a scaled-down FOWT

As explained in Sec. 2.2, the computational mesh is generated from the level-set Φ_ε of the reconstructed geometries, here the down-scaled FOWT. An error estimator is built from Φ_ε , to detect the areas requiring more adaptation. The errors observed are due to the discretization of the computational domain. Even if the solutions are exact at the

nodes of the mesh, these positions are discrete. The Finite Element framework implemented in ICI-tech uses linear interpolation to define the values of the different fields outside of these points. The linear description is suited for an exact representation of planes only. In a context where complex geometries are represented, especially curved surfaces, the exact description of the immersed objects then relies on the addition of points in the domain. As the minimal size of cells is limited to $\varepsilon/10$, the description of complex geometries, e.g., sharp angles or regions of strong curvature, can not be indefinitely improved. The highest values ($e > 2$) provided by the error estimator for a fully adapted mesh of $200K$ points, with $\varepsilon = 2 \cdot 10^{-2}m$ and $\varepsilon = 2 \cdot 10^{-3}m$, can be viewed in Figs. 2.15. These areas are concentrated around the surface of the immersed FOWT. For $\varepsilon = 2 \cdot 10^{-2}m$, the errors are exclusively found around the sharp interfaces of the immersed mesh, e.g., the trailing edge of the WT blades. Thin regions such as the heave plates of the floater or the tip of blades also generate errors around it, up to a distance corresponding approximatively to ε . With $\varepsilon = 2 \cdot 10^{-3}m$, the errors are much better spread along the frontiers of the FOWT, in all the regions where the surfaces are not plane.

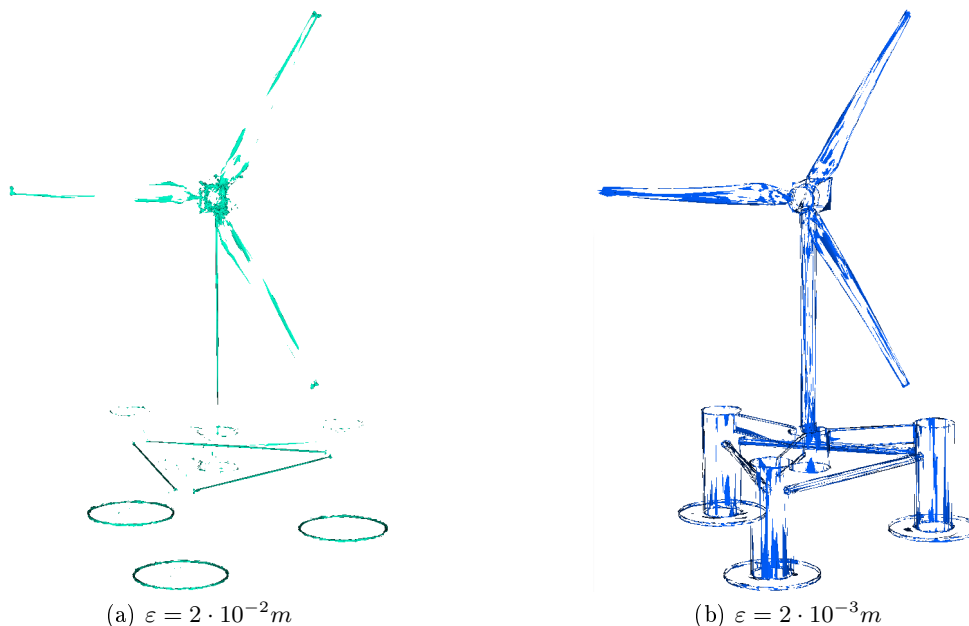


Figure 2.15: Isovolume of the error estimator $e > 2$, for different ε .

Two observations can be made from the error estimator fields printed. First of all, a coarse ε struggles to capture accurately the sharp interfaces, as the minimal size of mesh cells is limited. The errors are naturally concentrated in these regions. With a thinner ε , these areas are better represented, thus the adaptation of the mesh can focus on other regions. The second observation concerns the value of the error estimator for each test case. Intuitive thinking could state that, as the reconstruction is less precise with a coarser ε , the error estimator would reach higher values. On the contrary, the number of areas concerned by $e > 2$ is more important with $\varepsilon = 2 \cdot 10^{-3}m$ than with $\varepsilon = 2 \cdot 10^{-2}m$. This can be related to the second derivatives on the level-set functions, from which the error estimator is built. In particular, the derivation makes a ε^{-1} term appear, which tends to increase e when ε decreases. Moreover, the error is dimensionless and measured along the edges of the mesh. The adimensioning process forces a division by the length of the edge, which also increases the estimation of the error.

The evolution of the errors estimated with the parameter ε can be studied more precisely. Several reconstructions of the FOWT represented by the surface mesh from

Fig. 2.14a have been performed, with a constant load for the mesh, at $200K$ points. The width of adaptation ε is varied, and the errors obtained are measured. The L2- and infinite-errors, respectively written $\|e\|_2$ and $\|e\|_\infty$, produced on the computational points of Ω are considered. The definitions are presented in Eq. (2.18), with e_i the value of the error estimator at a point i .

$$\|e\|_2 = \left(\int_{\Omega} e_i^2 \right)^{1/2} \quad (2.18a)$$

$$\|e\|_\infty = \max |e_i| \quad (2.18b)$$

The errors are drawn in Fig. 2.16 as a function of ε . The error curves obtained confirm the behavior observed in Figs. 2.15, with the error estimator e tending to return higher values when ε is decreased. Even if this statement can be tampered. The variation of both the L2- and infinite-errors measured remain in an order of magnitude, while ε is divided by 100. The variation is particularly limited for $\|e\|_\infty$, which characterizes the highest error detected by the error estimator. The rise in $\|e\|_2$ depicts a global increase in the errors found in the computational domain. This can be linked with the explanations previously proposed, with the ε^{-1} term and mesh size involved. A drop is observed in $\|e\|_\infty$ with $\varepsilon = 10^{-3}m$, while the evolution of $\|e\|_2$ plateaued. This probably finds its origin in the number of computational points available in the mesh. The adaptation starts to struggle to generate a well-adapted mesh, even if the reconstruction provided correct results.

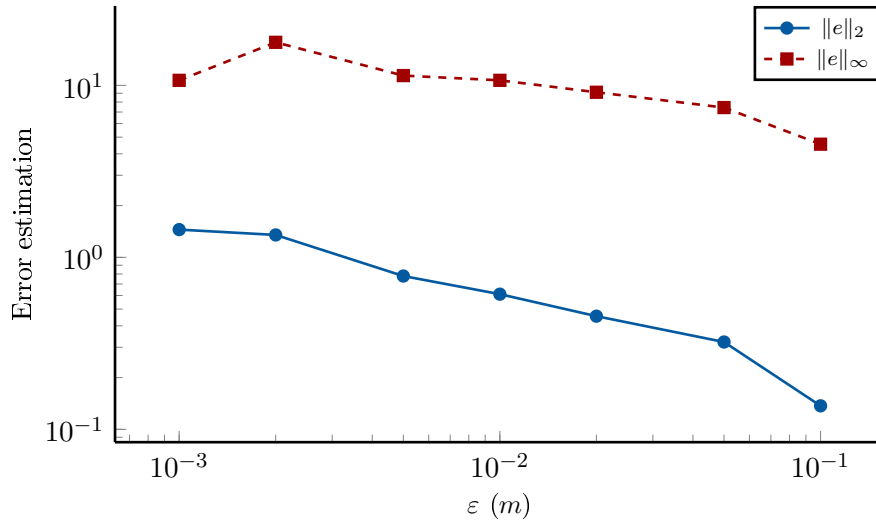
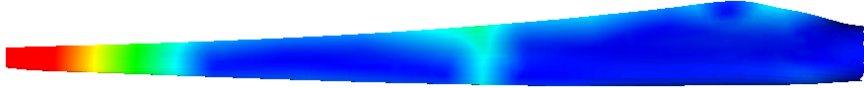


Figure 2.16: Estimation of the error produced during the immersion of a WT, with computational meshes of $200K$ points and a varying ε .

It is important to note that the error estimator, for all the information it provides, is highly dependent on the mesh generated. Moreover, it can not be used to evaluate the precision of the reconstruction performed. To that extent, a criterion has been defined to measure the difference between the immersed geometry and the iso-surface $\Phi_\varepsilon = 0$ reconstructed. This criterion is inspired by the Hausdorff distance, which characterizes the difference between two surfaces A and B , as written in Eq. (2.19).

$$d(A, B) = \max \left\{ \sup_{x \in A} \left(\inf_{y \in B} d(x, y) \right), \sup_{y \in B} \left(\inf_{x \in A} d(y, x) \right) \right\} \quad (2.19)$$



(a) Distance from A computational mesh to B immersed geometry.



(b) Distance from B computational mesh to A immersed geometry.

Figure 2.17: Distances measured for the construction of the Hausdorff distance with A mesh representing the immersed FOWT and B iso-surface $\Phi_\varepsilon = 0$ with $\varepsilon = 10^{-1}m$. Plot of the distance scaling from 0 to $10^{-1}m$, zoom around a blade.

The construction of a Hausdorff distance would require the computation from A to B and from B to A . Following this approach, A corresponds to the mesh representing the immersed object, and B to the iso-surface $\Phi_\varepsilon = 0$ reconstructed. In practice, both A and B are discretized using surface meshes of triangles. One of these meshes is considered as computational, and the distances are measured from each point to the set of triangles composing the other mesh. Even if the distance from the points of A to the surface of B is not equivalent to the distance from A to B , the position of mesh points concentrated around the interfaces allows this approximation. During an application to the immersion of a FOWT, the most challenging region to represent is the rotor, as thin geometries need to be represented. A Hausdorff distance is computed between the immersed mesh of a FOWT A , drawn in Fig. 2.14a, and the meshed iso-surface $\Phi_\varepsilon = 0$ B , obtained thanks to Paraview. When a poor reconstruction is performed, e.g., with $\varepsilon = 10^{-1}m$ in Figs. 2.17, the tip of blades are rarely represented. The evaluation of the distances as presented in Fig. 2.17a is much more representative of the quality of the reconstruction than the results obtained in Fig. 2.17b.

Similarly, even when a much thinner representation is achieved, e.g., with $\varepsilon = 10^{-3}m$ drawn in Figs. 2.18, the evaluation of the distances between A and B returns higher values. The regions where these errors are important are located in the most challenging places, e.g., at the trailing edge. For these reasons, only the distance from A to B is considered in the following. The "error" produced during the reconstruction is thus represented using the distance from the immersed mesh of a FOWT to the reconstructed iso-surface $\Phi_\varepsilon = 0$. The expression written in Eq. (2.20) is used.

$$e_{AB} = \sup_{x \in A} \left(\inf_{y \in B} d(x, y) \right) \quad (2.20)$$

The evolution of the reconstruction error e_{AB} with ε is plotted in Fig. 2.19 for the immersion of the FOWT presented Fig. 2.14a. Two different phases are observed on the curve. From $\varepsilon = 10^{-1}m$ to $\varepsilon = 2 \cdot 10^{-2}m$, a very rapid decrease is observed. This effect is related to the minimal size of mesh cells diminishing, allowing to better capture the challenging areas, e.g., the tip of blades. From $\varepsilon = 2 \cdot 10^{-2}m$ to $\varepsilon = 10^{-3}m$, the decrease is much slower. The geometries are globally well represented, and decreasing ε allows small improvements, which tend to better model small, local geometries. This whole development can be summarized from the identification of the red, dashed line. On the right of this line, i.e., with large ε , huge errors are made during the reconstruction, and whole sections of the geometry are not captured. On the left of this line, with small ε , local improvements are performed at each decrease of ε , which provides a slow evolution of e_{AB} .

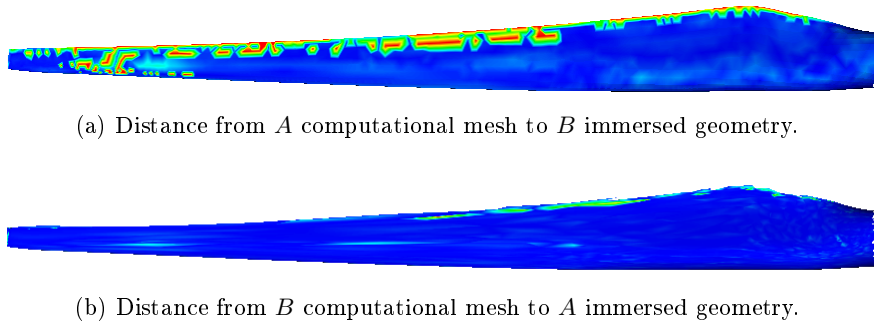


Figure 2.18: Distances measured for the construction of the Hausdorff distance with A mesh representing the immersed FOWT and B iso-surface $\Phi_\varepsilon = 0$ with $\varepsilon = 10^{-3}m$. Plot of the distance scaling from 0 to $10^{-3}m$, zoom around a blade.

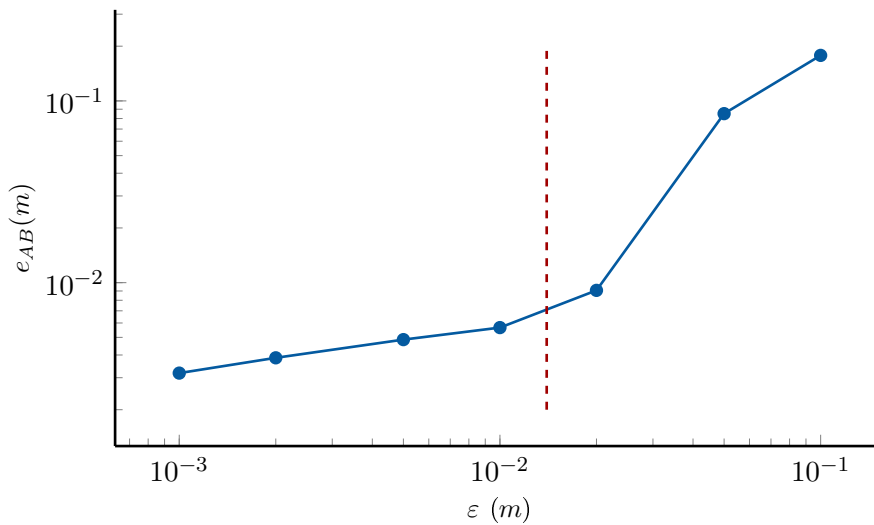


Figure 2.19: Error e_{AB} measured after the reconstruction of a FOWT, for different ε .

As this function is based on an upper bound of the distance, local imprecisions in the mesh can have a huge impact on e_{AB} , especially when ε is high. An illustration can be given from the example used for $\varepsilon = 5 \cdot 10^{-2}m$. The size of mesh cells is not thin enough to get a good representation of the tip of blades, which has distances on the order of $10^{-2}m$. When mesh points happen to be correctly positioned, the tip of blades can be properly reconstructed. On the contrary, an ill-positioned point has a huge influence on e_{AB} . The differences observed on Fig. 2.20 between the blade located on the left and the ones on the right provide a good example. The blade on the left is rather well reconstructed, as the distance is higher than $10^{-2}m$ only for challenging areas, such as the tip or the trailing edge. On the contrary, the tip of the other blades is not reconstructed, which leads to a considerable distance in these regions and finally affects on e_{AB} . An important variability is expected for e_{AB} . The estimation of the error realized during the mesh adaptation procedure will be much lower for the blade located on the left. As equidistribution of the error is targeted, the mesh around this blade will be coarsened during the next adaptation step, while refinement is expected around the two other blades. This justifies the rapid variations of e_{AB} between adaptation increments, which questions the accuracy of the positions of points with the largest ε on Fig. 2.19.

A remark concerning the generation of an adapted mesh can be done, from the study of the computational mesh presented in Fig. 2.14b. The slice of the mesh highlights the



Figure 2.20: Distance e_{AB} with A mesh representing the immersed FOWT and B isosurface $\Phi_\varepsilon = 0$ with $\varepsilon = 5 \cdot 10^{-2}m$. Plot of the distance scaling from 0 to $10^{-2}m$.

presence of coarse cells far from the turbines. This mesh, which is too coarse for flow simulations, can be refined to be used at the beginning of a computation. The adaptation procedure coupled with the resolution of the NS problem generates refinement based on the velocity of the flow, which will lead to thinner cells in areas of interest. Note that more than one billion points are required to obtain the same precision with a regular grid.

This case allows highlighting the computational savings provided by the octree. The construction of the octree was performed as presented in Subsec. 2.1.3 with a limit of 200 elements per subdomain and a max depth of 12. The cost of the evaluation of distances during the immersion of the wind turbine mesh in the adapted computational mesh is presented in Tab. 2.1. The focus has been placed on the computation of the distances because this step features the highest complexity. Moreover, the evaluation of the distances is often required during a simulation cycle, as explained in Subsec. 2.1.1, and its optimization is an absolute need. The computational time required for distance evaluations is divided by about 235 only by the addition of the octree. The browsing of the octree to select the closest neighbors for each point enabled to largely reduce the number of distances to evaluate. In this application case, only 0.081% of the exact distances from computational points to mesh facets are processed. The cost of the octree browsing increases the average time needed to compute a distance by a factor of about 5, which is reasonable considering the large reduction in the number of evaluations done. This example assesses the huge interest of the octree structure for distance computation algorithm, and by extension for mesh immersion procedure.

	Distances evaluated	Time for distance evaluation	Time per distance (s)
No Octree	$\sim 8693M$ (100%)	9min, 51s (100%)	$6.8 \cdot 10^{-8}$
Octree	$\sim 7M$ (0.081%)	2.5s (0.4%)	$3.5 \cdot 10^{-7}$

Table 2.1: Influence of octree for wind turbine immersion

2.4.2 Reconstruction of a large number of wind turbines

The octree implementation intended to limit the complexity of the distance algorithm. An idea of the scalability of the immersion procedure is obtained with a test case of 100 FOWTs. The WT mesh presented in Fig. 2.14a was duplicated 100 times, and the meshes obtained were randomly dispersed in a computational domain of size $50m \times 50m \times 2.5m$.

As a matter of comparison, the dimensions of the computational domain used for the reconstruction of a single wind turbine were $2m \times 2m \times 2.5m$. The immersed mesh consequently features 100 times more facets, the results of immersion are presented in Figs. 2.21 and 2.22.

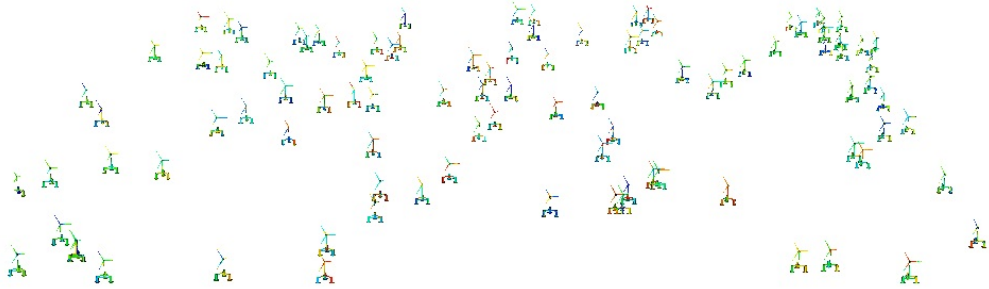


Figure 2.21: Reconstruction of 100 floating wind turbines

This test case is not representative of an operational wind farm in terms of positioning or of meshing. Here the level-set functions are used only for the representation of the FOWTs, without any free-surface or dynamic considerations. However, this effort was realized in order to prove the consistency of the mesh adaptation procedure in the situation of a large number of WTs. Very small mesh elements are required to mesh the tip of wind turbines, inside a large computational domain. The mesh adaptation procedure can detect these effects on randomly-positioned WTs, which would be impossible with traditional approaches refining the computational mesh in identified interest areas.

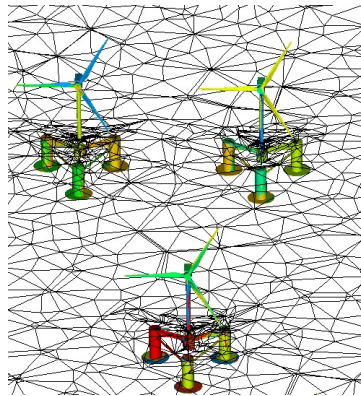


Figure 2.22: Zoom on 3 WTs and mesh slice

The computational mesh used for the immersion contains $18M$ nodes for the same precision as in the single turbine test case. The number of nodes used has been multiplied by about 150, while the immersed mesh was 100 times bigger. The main reason for this rise is the increased size of the computational domain. Despite these test case-related considerations, the rise in the number of nodes needed has been well limited. Moreover, the rise of computational times can be controlled with a corresponding increase in the number of processors used.

The colors depicted on the wind turbines in Fig. 2.21 correspond to the partitions used for the immersion. The location of the areas considered by the different CPUs is not predefined, which explains that small, divided-up color regions appear on the FOWTs. This test case was quantitative, and do not present the computational times.

2.4.3 Scalability of the immersion of FOWTs

The scalability of the immersion procedure is studied in this Subsec. from the test cases of immersed FOWTs. Two different studies are presented: they consist, respectively, in weak and hard speed-up studies. The speed-up characterizes the acceleration of the simulations when parallelism is involved, to evaluate the potential gains and losses due to the size of the cluster used. The perfect speed-up would be obtained by dividing the computational times by N_{CPU} when the number of CPUs is multiplied by N_{CPU} . This value is purely theoretical. As the number of processors rises, the cost due to the communications increases. The problem can be ill-suited for parallel computations, e.g., because of a small number of points in the computational mesh. All those reasons tend to provide, when the number of CPU is multiplied by N_{CPU} , a computational time divided by a factor $M < N_{CPU}$. the objective is obviously to get a factor M/N_{CPU} as close from 1 as possible.

Several parameters can be tuned to optimize parallel simulations. Among them, the most important is the number of CPUs involved. Speed-up studies intend to evaluate the influence of the number of processors used in the simulations. There is a difference in paradigm between weak and hard speed-up. The weak speed-up intends to keep the computational load similar on the CPUs, and to run different test cases of various sizes. The number of points in the computational mesh is thus multiplied by N_{CPU} , exactly like the number of processors. Weak speed-up then presents the time required to solve each test case. An ideal parallelized test case will return the same time expends for the different values of N_{CPU} . However, whenever a simulation is distributed on several processors, the efficiency of the resolution is never ideal. Communication between the processors, partitioning and parallel bottlenecks limit the scalability. Thanks to weak speed-up studies, the computational effort put into the set-up of the parallelism, e.g. due to the communications or the partitioning, can be evaluated. The efficiency of the resolution when the size of the test case rises is also crucial information before running massive simulations. The hard speed-up uses another paradigm. The test case is now fixed, and only the number of processors varies. The computational time would ideally be divided by N_{CPU} when the number of CPUs is multiplied by N_{CPU} . This process presents the acceleration observed on a given test case with different N_{CPU} , which offers the potential to evaluate the parallelization procedure as a whole. The ε was identical for the different simulations, in order to reach the same level of precision. A converged mesh was reached in both case.

Case	Nodes/core	Adaptation (s)	Immersion (s)	I/O (s)
1 WT	25K	6.2K	62	7.1
10 WTs	24.9K	8.2K ($\times 1.3$)	119 ($\times 1.9$)	20.1 ($\times 2.8$)
100 WTs	26.5K	9.8K ($\times 1.6$)	667 ($\times 10.8$)	140 ($\times 19.7$)

Table 2.2: Weak speed-up study with 6 WTs per processor

A reconstruction of FOWTs described by the mesh depicted in Fig. 2.14a is realized with different numbers of processors and a constant precision of the reconstruction, again on the order of the decimeter for a full-scale turbine. The simulations started from a coarse initial mesh and took 50 increments to perform full adaptation, leading to a fully adapted mesh. The parameters used to generate the octree were also kept constant. The data in Tab. 2.2 form a weak speed-up study, performed with a constant number of 6 CPUs per FOWT, for 1, 10 and 100 turbines. The adaptation, which corresponds to about 95% of the overall computational time, has proven to be very scalable. This observation was already done in Digonnet et al. (2017), for massively parallel isotropic and anisotropic meshing, on simpler test cases. The immersion, which showed interesting scalability for 10 turbines, tended to be limited by the octree parameters for the last

case. However, an interesting behavior is observed between the 1WT and 10WTs test cases. Finally, the I/Os seem to be the least scalable part of the reconstruction procedure. The writing of the output files is at stake here, and the only option to accelerate it would be to use binary files. This weak speed-up study highlights the scalability of the reconstruction procedure. It also emphasizes the importance of a good tuning of the octree, suited to the test case considered. The addition of elements in the representation of the immersed geometries shall lead to a deeper octree, in order to remain efficient.

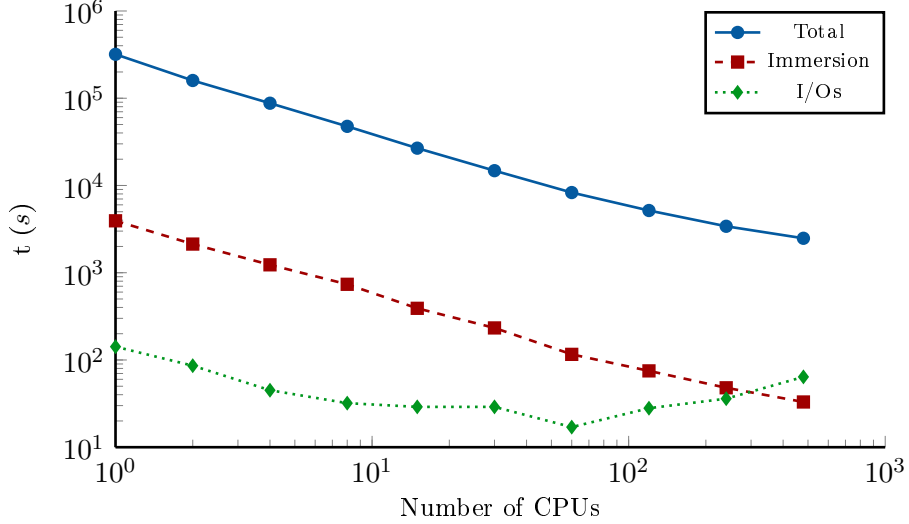


Figure 2.23: Time required to reconstruct 10 WT cases with a varying number of processors. The total time presented is composed of both the adaptation, immersion and I/O's expenses.

A hard speed-up study was also performed, with the reconstruction of 10 WT cases using a mesh of 250K points and a varying number of processors. The results are presented in Fig. 2.23. Even if the computational costs for the adaptation are not presented, the logarithmic scale view make it similar to the "total" one. The scalability of the reconstruction procedure is again interesting, as the same tendency was observed for both the adaptation and the immersion. While an ideal slope of -1 can be reached for the scalability curves, a slope of approximately -0.85 is obtained, which is satisfying. This means that, when the number of CPUs is multiplied by k_{CPU} , the computational time is divided by $k_{CPU}^{0.85}$. These studies prove the capacity of ICI-tech to handle large computational loads. However, in order to evaluate its potential for offshore wind applications, the focus must also be placed on the solver, which is done in the following.

2.5 Conclusion on the reconstruction of FOWTs

This Chap. presents the implicit boundary technique implemented in ICI-tech. An exhaustive presentation of the methodology used for the reconstruction of objects can be found in Sec. 2.1. The mesh adaptation on which the mesh immersion relies for the accuracy of the results is detailed in Sec. 2.2. The integration of those bricks inside the highly-parallelized framework of ICI-tech is detailed in Sec. 2.3. Finally, an application of these techniques to the simulation of FOWTs is proposed in Sec. 2.4.

During the developments of all the utilities presented hereafter, an orientation towards the optimization of the computational costs has clearly been given. This thesis followed these guidelines, particularly during the development of the reconstruction procedure onto a new ICI-tech platform. The octree has been developed for the immersion

of random elements, with a focus on the efficiency and reliability of the methods used. The reconstruction of FOWTs shows interesting results, both in terms of precision or of scalability. The representation of a large number of FOWTs show potential for ICI-tech's mesher to represent very small objects dispersed in large computational domains.

The developments also intended to produce a versatile tree-structure, suited for various fields of application. The octree produced can be used for the reconstruction of any geometry defined using different types of elements, thanks to a systematic recourse to bounding boxes. The development of the octree was realized as a software brick, to again increase its potential. This technological choice allows the call to the tree-structure for other HPC requirements, e.g., to cut huge meshes. This allows splitting large sets of elements, impossible to load by each processor, into several smaller sets.

Chapter 3

Numerical framework for the resolution of the flows around FOWTs

Contents

3.1	Generalities on the resolution of the incompressible Navier-Stokes problem	54
3.1.1	Modeling the flow with the Navier-Stokes equations	55
3.1.2	Resolving numerically the flow from the Navier-Stokes equations	58
3.1.3	Navier-Stokes equations for high-resolution simulations of FOWTs	63
3.2	Monolithic finite element formulation for the incompressible Navier-Stokes equations	65
3.2.1	Finite Element formulation	65
3.2.2	Variational MultiScale of the Navier-Stokes equations	67
3.2.3	Mixing laws	69
3.2.4	Movements of phases	73
3.2.5	Monolithic resolution of the Navier-Stokes equations	74
3.3	Numerical utilities of ICI-tech for the simulation of FOWTs	75
3.3.1	A posteriori computation of forces applied on an immersed body	76
3.3.2	Numerical wave tank	78
3.3.3	Buoyancy and Fluid-Structure Interaction	84
3.4	Conclusion of the numerical framework	86

The simulation of FOWTs requires the modeling of both the aerodynamic and hydrodynamic effects acting around the structures. The correct representation of these phenomena is critical for the design of a FOWT, which will frequently be under the effect of extreme environmental conditions. Currently, simplified approaches are used in the industry for the simulation of FOWTs. As explained in Chap. 1, rotor models are currently used, to limit the computational expenses linked to the resolution of the aerodynamics. The hydrodynamics are also solved using simplified, cheaper methods, e.g., with the Morison equation. These approaches can provide satisfying results for particular measures, e.g., loads applied to the structures, under specific environmental conditions and once they are calibrated.

The definition of accurate simulators of FOWTs, on which the flows are accurately solved, is a complementary approach to the industrial solutions. The data required for the calibration can be obtained at a reduced cost thanks to thin numerical simulation, especially when experimentations are too hard and/or expensive to set up. The accurate resolution of flows around FOWTs is also interesting to highlight specific, unsteady and dynamic effects appearing in particular conditions, that may be hard to reproduce in experiments. Moreover, a precise resolution of the flows is necessary to consider the fluid-structure interaction of high accuracy, especially with floating structures. For all those reasons, the developments of softwares allowing a thin resolution of the flows around FOWTs is interesting.

The description of the flows is obtained in theory from the Navier-Stokes equations. This problem is considered in an incompressible formulation in this thesis. Under this configuration, it solves the velocity and pressure of the flow, depending on the properties of the fluid and on the environmental conditions specified for the computational domain. Some generalities about the resolution of the Navier-Stokes equations are proposed in Sec. 3.1

The resolution of the Navier-Stokes equations is coupled in this work to the reconstruction techniques and mesh adaptation introduced in Chap. 2. Interesting compatibility is found between those methods, with the potential to simulate accurately the behavior of FOWTs. The geometries are represented exactly, using the reconstruction technique and the mesh adaptation procedure. A high concentration of computational points is found around the surface of the FOWT, which allows capturing the fluid flows up to the boundary layers. The computational costs induced by this approach are considerable, as a thin meshing is required for both the geometries and the fluid effects. However, the level of accuracy enabled by this methodology is very important. The temporal and spatial discretization, along with the type of resolution of the Navier-Stokes equations, are critical for the precision of the results. The methodology followed for the resolution and the issues related to it are presented in Sec. 3.2.

The numerical simulation of FOWTs requires the representation of the environmental conditions under which the systems are operating. Moreover, the movements of the FOWT need to be representative of the one observed in reality. To that extent, several implementations have been realized within the scope of the thesis, to make the simulations of FOWTs possible. The different additions are presented in Sec. 3.3

3.1 Generalities on the resolution of the incompressible Navier-Stokes problem

The Navier-Stokes (NS) equations rule the evolution of fluid flow. Their resolution is a challenge, that is addressed in this Sec. Different approaches to simulate incompressible

fluid flows using the NS equations expressed in velocity in pressure are presented. The approximations required in the formulation of the NS problem for its resolution are presented in Subsec. [3.1.1](#), while the different issues related to the resolution itself are evoked in Subsec. [3.1.2](#).

3.1.1 Modeling the flow with the Navier-Stokes equations

This thesis is dedicated to the simulation of FOWTs, which imposes the consideration of multiphase flows. The level-set approach presented in Chap. [2](#) allows the determination of the properties of the fluid everywhere in the computational domain. The next step consists in the resolution of the flows. This Subsec. intends to present the different techniques used to represent the flows. The application of the Navier-Stokes problem to continuum media in an incompressible formulation and the equations at stake are presented. The issues raised by turbulence are examined in a second time.

Navier-Stokes equations

The Navier-Stokes (NS) problem defines the behavior of the fluids, and is composed of two equations: one guarantees the continuity of the solutions obtained, while the second one states that the momentum will be conserved. The problem is presented in the following in velocity and pressure, as this formulation is used in the context of floating wind turbines.

The continuity equation states that, for any volume inside the computational domain, the mass entering the volume is equal to the mass leaving it. The continuity equation for a fluid is presented in Eq. [\(3.1\)](#), with ρ fluid density, \mathbf{u} fluid velocity, t time and ∇ differential operator.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0. \quad (3.1)$$

This document is focusing on flows around floating wind turbines. The fluids simulated are water and air which, in this context, may be considered incompressible. In this case, the density of the fluids is constant, and the continuity condition can be rewritten. A physical interpretation of this phenomenon is possible with water going through a tunnel: no compression is allowed, so, to satisfy the continuity condition, the velocity outwards is increased. Hence, Eq. [\(3.1\)](#) is reduced to Eq. [\(3.2\)](#).

$$\nabla \cdot \mathbf{u} = 0. \quad (3.2)$$

The momentum equation is the second one composing the NS problem. It states that the momentum is conserved locally at each point of the domain. This equation is presented in Eq. [\(3.3\)](#) for an incompressible formulation with p the pressure, η the viscosity and f the exterior force, e.g. gravity or another mass force.

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) - \eta \Delta \mathbf{u} + \nabla p = f. \quad (3.3)$$

The comprehension of fluid flows requires the resolution of the NS equations. The problem is solved on a domain Ω of frontier Γ , where the NS equations are verified. This formulation is completed by an initial solution \mathbf{u}_0 and by Dirichlet and Neumann boundary conditions (BCs), applied respectively on Γ_D and Γ_N , verifying $\Gamma_D \cup \Gamma_N = \Gamma$ and $\Gamma_D \cap \Gamma_N = \emptyset$. The viscous tensor σ and the unit vector normal to Γ_N appear in this

expression. The whole incompressible NS problem then reads:

$$\begin{cases} \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega \\ \rho(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}) - \eta \Delta \mathbf{u} + \nabla p = f & \text{in } \Omega \\ \mathbf{u} = \alpha & \text{in } \Gamma_D \\ \sigma \cdot \mathbf{u} = \beta & \text{in } \Gamma_N \\ \mathbf{u}(t=0) = \mathbf{u}_0 & \text{in } \Omega \end{cases} \quad (3.4)$$

However, no trivial solution can be found for the NS problem. The existence of smooth solutions to the NS problem in 3D with given initial solution has even been named as a Millennium Prize problem in mathematics. This limitation is due to the non-linear term $\mathbf{u} \cdot \nabla \mathbf{u}$ appearing in the momentum equation. Consequently, as continuous solutions of the NS problem are not known, the domain Ω has to be discretized to enable the acquisition of approximate solutions. Another issue intrinsic to the NS problem is the turbulence. This phenomenon becomes very complex when the Reynolds number of the flow rises, which generates an important computational load. A common approach is to modify the formulation of the NS equations to handle the turbulence. Different levels of accuracy can be achieved, depending on the alternative problem solved. These formulations are tightly linked with the discretization chosen for Ω . These topics are detailed in the following.

Modeling turbulence with the Navier-Stokes equations

The turbulence is a physical phenomenon characterized by vortices appearing in a disturbed fluid flow. If a viscous fluid is considered, its contribution will be minor compared to the viscous effects. However, when high Reynolds-number flows are considered, e.g. aerodynamic effects, the comprehension of turbulence is critical for a good representation of the phenomena occurring.

The vortices initially induced by fluid effects have a direct influence on its vicinity, and progressively lose energy, dispersed in the neighboring fluid. Smaller vortices are generated, create even smaller eddies, which finally vanish as the viscosity of the fluid becomes dominant. This cascade of energy has been characterized by [Kolmogorov \(1941\)](#). From this observation, an accurate simulation of the fluid imposes the capture of all the vortices, i.e. to use a discretization thinner than the size of the smallest eddies. However, in the case of high-Reynolds flows the computational cost required for both the meshing and the resolution of flows imposes the usage of a simplified formulation of the NS problem, as depicted in [Fig. 3.1](#). The biggest challenge comes with the non-linear term in $\mathbf{u} \cdot \nabla \mathbf{u}$, which generates the turbulence and is complex to approximate. In the next paragraphs, three main approaches used for the resolution of the NS problem are presented.

The Direct Numerical Simulation (DNS) formulation of the NS problem is an approach that intends to solve all the scales of the flow, followed, e.g., by [Scardovelli and Zaleski \(1999\)](#). This approach requires the capture of each eddy contained in the flow, to measure its evolution and decay. The precision required is deduced from the Kolmogorov cascade, with a characteristic length depending on the Reynolds number of the flow.

DNS is currently unaffordable for engineering cases featuring complex geometries and high-Reynolds flows such as floating wind, as the resolution required in the discretization of the computational domain leads to prohibitive meshing. This technique is nowadays most commonly used for turbulence studies on academic test cases. Some authors now focus on WT blades, e.g., [Ducoin et al. \(2017\)](#) for a Savonius-type VAWT. However, future advances in the computational area could pave the way for highly accurate engineering simulations.

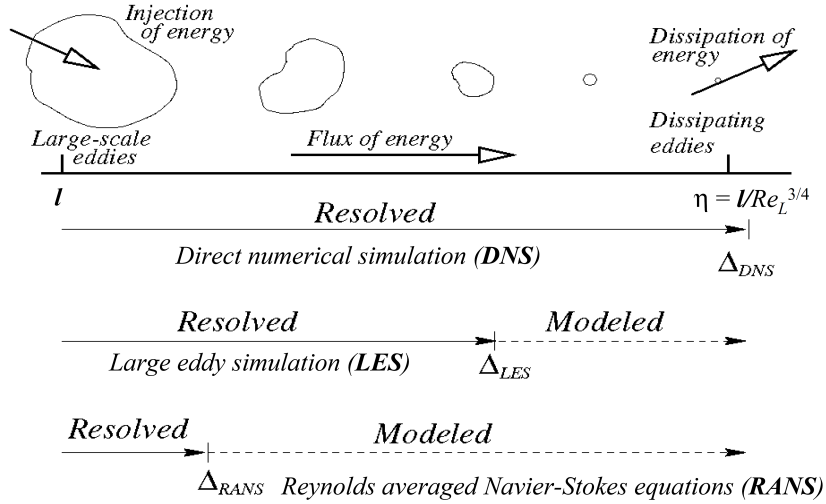


Figure 3.1: Kolmogorov cascade and formulations of the NS problem, from Bakker (2006). DNS solves all the scales of the flow, while RANS models the major part of turbulence. LES operates as an intermediate approach.

The RANS formulation of the NS equations is a paradigm that aims at modeling all the turbulence in the flow, at reasonable computational expenses. The strategy is to reduce the computational expenses through filtering of the turbulent structures, to keep only the constant part of the flow. This can be viewed as a time-averaging of the NS equations, i.e. a resolution of the NS problem ignoring the $\partial \mathbf{u} / \partial t$ term. The influence of turbulence is modeled, through a term included in the NS problem. RANS can be modified to make it time-dependent, when the evolution of the system can be expressed using boundary conditions varying slowly. This approach is called URANS, for Unsteady RANS, and was used, e.g., by Quallen and Xing (2016) in the context of FOWTs.

The computational expenses are largely reduced thanks to the RANS formulation of the NS problem. The time-averaging allows getting an idea of the flow with the simulation of a single time step. Vortices are not tracked, which allows a coarser discretization of the domain, and consequently faster simulations. However, the results obtained do not highlight the perturbations observed in the flow, as proposed in Fig. 3.2. URANS tackles the time-constant limitation, even if the flow resolved will still have a smooth behavior. Some phenomena are also impossible to represent using time-averaged NS equations, one of the best known being the Karman vortices.

LES is an intermediate approach between RANS and DNS in terms of precision and computational effort. The turbulent structures are resolved up to a certain characteristic length. The end of the energy cascade is matched using viscous terms representing the subgrid-energy decay of the vortices. An extensive study of the turbulence modeling in LES is performed by Meneveau and Katz (2000).

Mathematically, the LES formulation of the NS equations is created by a scale separation in velocity $\mathbf{u} = \mathbf{u}_h + \mathbf{u}'$, \mathbf{u}_h being the resolved velocity and \mathbf{u}' the small scale one. The non-linear term of the NS equations generates a closure problem, where the subgrid-scale stress needs to be modeled. More information can be found in Hughes et al. (1998).

VMS is an alternative approach proposed by Hughes et al. (1998) and Codina et al.

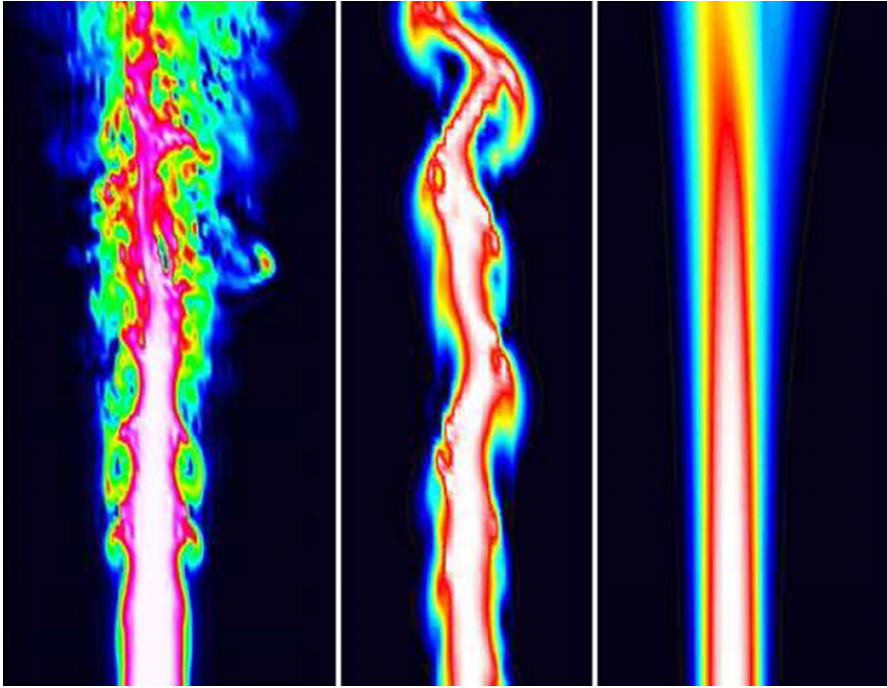


Figure 3.2: Turbulent jet solved with different formulations of the NS equations from [Maries et al. \(2012\)](#): DNS (left), LES (center), and RANS (right).

[\(2017\)](#), operating as an implicit LES. A similar scale decomposition is processed in VMS, and two distinct problems are built: a large-scale one and a subscale one, both of them including each velocity components \mathbf{u}_h and \mathbf{u}' . The subscale velocity \mathbf{u}' is approximated analytically using the fine-scale problem, and replaced back into the coarse-scale one. This allows building a formulation providing a similar precision to LES, without any explicit correction of the problem.

Mixed techniques have also been developed to obtain precision and computational effort located somewhere in-between LES and RANS. E.g. Detached Eddy Simulation (DES) is a combination of the RANS and LES paradigms, used in particular for simulation around bodies. To limit the precision required for the mesh around the body, a RANS approach is used in this region. On the contrary, the resolution of accurate far flow is achieved thanks to LES.

3.1.2 Resolving numerically the flow from the Navier-Stokes equations

As mentioned previously, the resolution of the NS problem is not straightforward. Whenever complex geometries are simulated, approximated formulations of the NS equations are required. But even with those modified formulations, no immediate solutions can be obtained. The resolution is achieved in practice thanks to a discretization of the area where the NS equations are solved. Similarly, a discretization in time is required. Moreover, as this thesis focuses on the simulation of FOWTs, the problematic of fluid-structure interaction (FSI) needs to be discussed. The upcoming Subsecs. present those issues.

Spatial discretization techniques and meshing

The discretization of the geometries is critical for an accurate resolution of the NS equations. The first step is the definition of an appropriated computational domain, depicting as well as possible the shape and characteristics of the problem simulated. A point set is built to represent the computational domain, and transformed into a mesh. The

distribution of the points define cells of various size in the domain, which translate to fluctuating levels of precision achievable. Yet, as mentioned in Subsec. [3.1.1](#), the choice of a formulation for the NS equation also has a huge impact on the accuracy of the results obtained. Consequently, as for any simulation, the numerical framework chosen for the resolution of the NS problem and the mesh generated needs to be compatible. To that extent, different sources of errors must be regarded. First of all, the size of the mesh cells need to be appropriated everywhere in the computational domain. The characteristics of the flow being different depending on many factors, e.g., geometries or boundary conditions, this target length can vary in the domain. Moreover, the results obtained with a given formulation written need to converge towards a solution when the mesh becomes thinner. This converged solution needs to be validated, compared with experimental results, to prove the accuracy of the solver used. Successive refinements of an initial coarse mesh should then provide solutions converging towards the "reality". Finally, the numerical error produced by the formulation of the NS equations must be put in perspective with the spatial error generated by the mesh. To illustrate these statements, the NS frameworks described in Subsec. [3.1.1](#) can be considered. A time-averaged, turbulence-modeled RANS formulation evaluated with a very thin DNS mesh makes no sense, as it leads to prohibitive computational effort for only minor improvements. Likewise, a DNS formulation evaluated on a RANS mesh will provide poor results.

Different types of connections can be used. The most common are composed in 2D of triangles or quadrilaterals, while 3D meshes use for the major part tetrahedra or hexahedra. Aside from these different mesh types, two major families of meshes exist: a structured mesh reproduces a particular pattern, and its connectivity is defined spatially ; on the contrary, unstructured meshing generates irregular grids. Different approaches can be used to solve the NS problem, suited for certain kinds of discretization. Some of them are detailed in the coming paragraphs.

The Finite Difference Method takes advantage of a structured mesh to optimize the computations. The equation is written at each of the computational points, and accounts for the values at the neighbor nodes. This technique is applicable with structured meshes only, when the distance between two connected grid points is easily evaluable. The whole set of equations can be reformulated into a matricial formulation, which is then solved. This method allows evaluating the equations considered at every computational points, and is relatively affordable. However, the formulation is solved exactly only at the grid points, and do not account for the phenomena occurring elsewhere. This type of results is reliable when thin grids are considered. The combination of small distances between points and homogeneous grid translates to an important computational effort. A finite difference solver with a LES formulation of the NS equations is used for the simulation of wind farms by [Joulin et al. \(2019\)](#).

The Finite Volume Method does not consider the point net, but instead relies on control volumes defined around each node for the resolution. The equations to be solved are written on the whole control volume, and transferred to its frontiers. A flux formulation appears, which is conservative and consequently allows the resolution. The problem is constructed under a matricial format for the resolution. Both structured or unstructured meshes can be used with this framework. This achieves mass conservation during the simulation, making this technique rather popular for the resolution of the fluid flows. [Calderer \(2015\)](#) used a finite volume solver for the resolution of the NS equations under a LES formulation in the context of FOWTs.

The Finite Element Method (FEM) also considers cells for its resolution. Again, the equations are written on a whole element. The approach relies on the linearization of the

solution on each element, using solutions at the nodes of the mesh and base functions for interpolation. The initial equation, also named strong formulation, is not directly solved in FEM. Weak formulations, built from the integral over an element of the strong formulation multiplied by test functions. The whole problem, applied to all the elements simultaneously, is written under a matrix formulation and solved. FEM meshes can be either structured or unstructured. [Yan et al. \(2016\)](#) used a FEM framework to solve the NS equation in a LES formulation for the simulation of FOWTs.

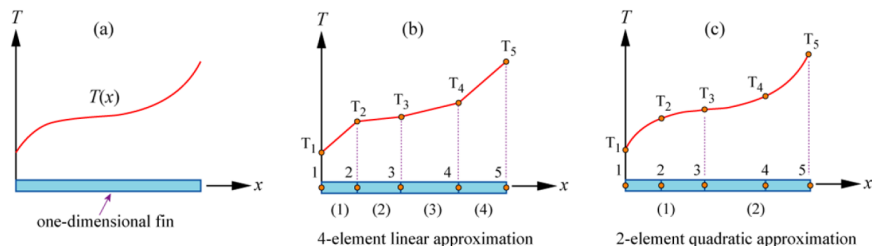


Figure 3.3: Comparison of FEM results obtained with first- and second-order base functions.

The precision of the results obtained with the FEM is directly dependent on the linearization chosen for each element. To that extent, the order of the interpolation is critical. Higher-order schemes will require the resolution at more Gauss interpolation points and uses more base functions. However, the precision achieved with higher order can be largely improved, which allows coarsening the mesh used. A 1D illustration of this statement can be found in Fig. [3.3](#). The results obtained enable a much better fitting of the desired solution, while the function is evaluated at the same number of points. This technique provides another possibility in the control of the discretization error when using FEM.

Several techniques using particle approaches have been developed over the years. These methods do not require a mesh to solve the flows, but only a set of points. In particular, the Smoothed-particle hydrodynamics (SPH) and the Lattice Boltzmann method (LBM) have attracted a lot of attention. In SPH the displacement of the fluid is inherited from the movements of the particles, which are solved from the NS equations. SPH is efficient to solve free-surface flows, particularly for fluids of low viscosity. However, in an air/water context, no aerodynamic results are provided by SPH, and another solver needs to be coupled to it for FOWT simulation. [Leble and Barakos \(2016\)](#) used a SPH solver to solve the hydrodynamics for the simulation of a FOWT. In LBM the NS problem is not solved. The movement of the particles is obtained after the resolution of the discrete Boltzmann equation, which includes a collision model that can be adapted to the context simulated. The Boltzmann method, similarly to the Finite Difference approach, needs a regular grid, which limits its application in the context of FOWTs. An example of WT simulation using the Boltzmann method is proposed by [Deiterding and Wood \(2016\)](#).

Time discretization techniques

Just like the computational domain has to be discretized, the presence of a derivative over time in the NS equations imposes a time discretization. The NS problem will be solved at distinct instants of the time, separated by a time step. The choice of the time step at each instant needs to be well thought to capture the fluid effects required by both the numerical formulation and the spatial discretization. To that extent, the characteristic time ruling the evolution of phenomena need to be regarded in correspondence with the meshing. As an example, a DNS formulation requires the capture of small,

vanishing vortices, thus requiring a very thin meshing, and consequently very small time steps. The time step can be optimized during a computation. Time adaptation tools evaluate the flow in the computational domain at a given instant, and define a time step guaranteeing both a good control of the time error and an optimized computational effort.

When discretizing in time, two different paradigms can be chosen for the resolution at the time step $n + 1$: the solution can either depend on the results at the time step n , or anticipate the evolution towards the timestep $n + 1$. This writes as presented in Eqs. (3.5) for the resolution of $Ax + b$. The first option is known as explicit, while the second is implicit.

$$\begin{cases} x_{n+1} = Ax_n + b \\ x_{n+1} = (I - A)^{-1}b \end{cases} \quad (3.5)$$

When the problem to solve is non-linear in time, the influence of the time-discretization is crucial for the accuracy of the simulation. In particular, the numerical schemes are known to provide convergence issues when the time step chosen is too large. The respect of the Courant–Friedrichs–Lewy (CFL) condition is necessary to ensure convergence in this situation. The CFL condition is presented in Eq. (3.6), with Δt time step, Δx_i mesh size along each dimension, u_{x_i} velocity of the flow along each dimension, C_{max} a constant.

$$\Delta t \sum_i \frac{u_{x_i}}{\Delta x_i} < C_{max} \quad (3.6)$$

The value of the constant C_{max} depends on the schemes used. If explicit schemes are used, C_{max} is about 1, but its value for implicit schemes is larger. Implicit schemes consequently allow higher time steps than explicit ones. The schemes presented in Eq. (3.5) share the same convergence rate, 1, which means that when the time step is divided by N , the error is too. Schemes with higher convergence rates can be defined, e.g. Runge-Kutta ones, which can reach order 4. The higher-order schemes are usually built as mixed explicit-implicit, as the non-linear evolution of the solutions are better approximated when both the previous and the current solutions are considered. An example which provides a second-order accuracy, known as the Crank-Nicholson scheme, is given in Eq. (3.7).

$$x_{n+1} = (I - A_i)^{-1}(A_x x_n + b) \quad (3.7)$$

Boundary conditions

The NS problem presented in Eq. (3.4) requires boundary conditions (BCs). BCs are imposed in particular regions of the computational domain considered, to impose the flow in those areas. The most common types of conditions are, namely: the Dirichlet ones, which impose a value for a variable of the problem ; the Neumann ones, which characterize a flux for one of those variables. The BCs are commonly imposed on the borders of the domain, to define the inflow and outflow. They can also be applied in the inside, especially when geometries are dispersed in the domain. The BCs are always applied on the variables of the problem considered, and consequently, NS resolutions involve conditions in velocity and flux.

The BCs are necessary to build the problem, thus they have to be imposed before the resolution of the NS equations. The integration of the BCs is then performed through the discretization schemes. The resolution uses the information provided to solve the problem. Each discretization method uses the BCs in a different manner, but the concept behind the resolution is always to communicate them through the neighboring mesh

points or elements. The matricial formulation of the NS problem then allows the full resolution, with influence accorded to the BCs.

Fluid-structure interaction

When a fluid flow around geometries is simulated, the correct representation of the objects dispersed in the domain is critical to get accurately resolved flows. The physical shape of the objects must be well reproduced, and their influence on the flow has to be correctly taken into account. If the objects have a flow-driven motion, the influence of the flow on their movements is critical, but the perturbations induced by the objects on the neighboring flow are very important to capture too. If the objects are deformable, the forces applied by the flow on the boundaries of the objects are required to measure the deformations induced.

Two different methodologies can be followed for the resolution of FSI. The partitioned approach relies on two different solvers, a solid and a fluid one, which are used independently. The fluid solver is used to generate a fluid flow, whose results are inputted inside the structural solver. In particular, the forces applied by the fluid on the structures are necessary to measure the deflection of the geometries. A weakly-coupled partitioned approach conserves the results obtained by both solvers and advances to the next time step. On the contrary, a strongly-coupled one continues to iterate alternatively on fluid and structural solver, until a convergence condition is achieved. The use of two standalone solvers imposes the coupling of softwares already existing. The only challenge is the organization of the coupling, which can become a real challenge. Weakly-coupled FSI solutions tend to be unstable, while the convergence of hard-coupled partitioned procedures can be very time-consuming. This strategy was followed by [Leble and Barakos \(2016\)](#) in the context of FOWTs. A monolithic approach considers both the fluid and structural effects using a single solver. A solution needs to be implemented, which solves the NS equations in the fluid, and a structural problem in the solid. This direct solving can be very expensive computationally, as the system may be hard to build and/or slow to converge. However, this approach allows to use a single mesh, and largely reduces the communications between the solvers. [Yan et al. \(2016\)](#) used a quasi-direct coupling, derived from a monolithic approach, to handle the FSI of FOWTs.

The representation of the geometries can be realized in several ways. The most straightforward one is to use body-fitted meshes. Those meshes of the computational domain feature holes at the position of the different objects, whose boundary is meshed exactly. An example can be found on [Fig. 3.4](#) for a NACA 0008 profile. Boundary conditions can be applied on the borders of the object to deflect the flow, and obtain a resolution around a body. This technique enables to accurately define a geometry, and consequently to obtain a correct representation of the flow. Once the flow is solved, if needed, the forces can be computed directly on the frontier of the object. The body-fitted mesh is, in this situation, also solution-fitted. This choice imposes, when the object is deformed or displaced, to remesh all the domain has from scratch. To tackle these limitations, techniques such as the Arbitrary Lagrangian-Eulerian (ALE) approach have been developed. The ALE permits remeshing when the displacements of the mesh points are small. When an appropriated time discretization is used, ALE makes the mesh follow the motion of the structure. FSI simulation of WT blades was realized by [Bazilevs et al. \(2011b\)](#) with a ALE formulation moving the fluid mesh.

An alternative to body-fitted meshing considers bodies reconstructed implicitly, e.g. using the immersed boundary approach. This methodology uses a unique mesh and a signed-distance function to reconstruct the object. The points of the computational mesh are segregated between inside and outside the object to represent. Hence, the

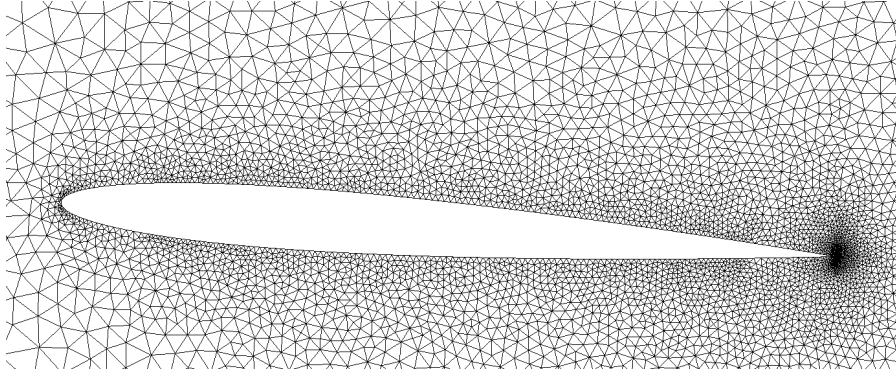


Figure 3.4: Body-fitted mesh around a NACA 0008 profile, for a low Reynolds flow ($Re = 100$).

geometry obtained in the computation does not exactly represent the object, but is dependent on the quality of the meshing around its frontiers. Remeshing is still necessary, but can be achieved using the signed-distance function. Once the representation of the immersed object is accurate, mixing laws can be applied to characterize the fluid or solid at each computational point. The fluid and solid problems can then be solved. [Hachem et al. \(2012\)](#) used an immersed volume method for fluid/solid simulations.

This formulation is appropriated for both the monolithic or partitioned FSI approaches. The meshing inside the computational domain allows solving directly the structure equation on this mesh inside a monolithic formulation, but an overset grid can also be used. Two different frameworks can be used to define implicit boundaries, which are exemplified in the following for the immersion of a fluid. The first one is the Volume of Fluid (VoF) approach. The computational mesh is examined, and the proportion of fluid found in each cell is measured. This method is efficient, but the cells with a very low proportion of fluid tend to complicate the resolution of the flows. The second approach uses level-set functions, which are based on the signed-distance from each point of the mesh to the interface of the fluid represented. This signed-distance obtained is rearranged to obtain a smooth-Heaviside function, which is able to represent accurately the interfaces of the fluid immersed. This approach increases the computational costs, and is challenging when the interior of the object is hard to define. It also represents more precisely the interfaces. In the context of fluid flows, both methods require the resolution of a transport equation to solve the position of the fluid's free-surface iteratively in time.

A last consideration about FSI meshing can be done on the definition of the boundary conditions at the borders of the computational domain. The flow is naturally deflected when geometries are positioned in an airflow. If the computational domain is too small, the definition of the BCs can provide additional perturbations to the flow, thus generating imprecise results, even if a good convergence is achieved for the solver. Those confinement effects must be avoided, as they compromise the good representation of the system simulated.

3.1.3 Navier-Stokes equations for high-resolution simulations of FOWTs

This thesis focused on accurate simulations of FOWTs. This Subsec. highlight the considerations behind the development of a simulator providing high-accuracy. As mentioned in Sec. [1.2](#), the resolution of the Navier-Stokes equations provides the best precision among the different strategies followed for the simulation of FOWTs. In Sec. [1.1](#), the issues associated to the simulation of a realistic operational FOWTs are presented. As a

recall, sea states need to be reproduced, as well as the aerodynamics. The interactions of the FOWT with its environment are required, and several different situations need to be distinguished. The aerodynamic efforts applied on the blades, and the rotor behavior are important to characterize the energy produced by the FOWT. The motion of the whole structures, and the deformation linked, e.g., to aeroelasticity, would be needed to get the best precision. The dynamic of the moorings is also very important, to ensure that the stationkeeping of the FOWT is assured.

The set-up of an accurate framework for the resolution of the dynamics of a FOWT requires the handling of a three-phase simulation. Two fluids (air and water) define a free-surface, which intersect a solid (the FOWT) evolving at its surface. The FOWT endures a flow-driven motion, which is limited by moorings. The resolution needs to handle both the fluid and solid behaviors, hence a coupling is required. The definition of a hard coupling strategy, based on a monolithic approach, has the best potential in terms of precision.

To limit the coupling between fluid softwares, the aerodynamics and hydrodynamics can be solved using a single software. This choice requires the capture of the air/water interface, and its tracking in time. To that extent, an immersed volume approach is required, along with an appropriated meshing of the free-surface. Either the same methodology can be used for the definition of the solid structures, or a body-fitted mesh needs to be defined. In the first situation, a lot of effort needs to be placed in the positioning of computational point around the fluid/solid interface, while the other strategy imposes the definition, at each time iteration, of a FOWT-fitted computational mesh.

The accuracy of the flow resolved depends on the formulation chosen for the NS equations. A DNS paradigm is unachievable computationally, which pushes LES-type formulations forward. The computational mesh cells thus need to capture the largest eddies characterizing the Kolmogorov cascade, which impose the use of relatively thin meshes. As high-Reynolds flows are observed for FOWT simulations, the turbulence generated reduces the size locally required for mesh cells. This small size of mesh cells cannot be used everywhere in the computational domain, because of the efforts it generates in terms of memory and resolution. Refined areas need to be defined, in interest areas where the fluid and solid effects will be important. The generation of this mesh can be done using user-defined interest zones, generating higher-resolution patches. Another strategy consists in automatic adaptation, which refines the mesh in regions defined from simulation results.

The resolution of the NS problem is performed on the computational mesh generated. Among the spatial discretization methods evoked in this Sec., an optimal accuracy is obtained using Finite Volumes or Finite Elements Methods. The Finite Difference and Lattice-Boltzmann approaches are competitive only on regular grids, which compromises their accuracy for multiphase flows, while the SPH method can solve only the hydrodynamics. From the order of convergence of the error produced spatially, a time discretization scheme of corresponding order can be selected.

The representation of the rotor is another issue for a simulator of FOWT. In a framework solving the NS equations, two different alternatives are found. A model can be used to account for the effect of the WT rotor, among the solutions proposed in Subsec. [1.2.1](#). The other option consists in the representation of all the geometries, including the blades. This option induces a higher computational effort due to blade meshing and all the aerodynamic effects developing around. It also offers the potential to simulate all the physics with better precision. Similarly, models can be used to account for moorings

in the simulations, which provide various levels of precision. An optimal simulation would still consider each mooring line and conduct an elasticity study on the fully meshed on it.

In the set-up of a simulator of FOWTs, a consideration can be made about the boundary conditions. As realistic sea states and wind profiles have to be generated, a lot of attention needs to be placed in the definition of BCs at the inlet. A propagation of the wave field must be realized in order to reproduce sea conditions before it impacts the FOWT. This allows getting rid of perturbations that may produced by wave generation techniques. The implementation of slip BCs on the sides of the domain constrains well the water flux. However, the dynamics at the free-surface may be impacted near the sides. The FOWT simulated thus needs to be placed in the middle of a rather large computational domain. Similarly, the aerodynamic effects are impacted by the presence of BCs too close from the rotor. In the meantime, constraining all the computational domain to impose the fluxes is not affordable, as it would break the incompressibility condition. Areas must be left "open", where recirculation may be established during the resolution. These perturbations must be limited, e.g., through the addition of viscous layers.

The numerical methodology followed during this thesis for the development of a platform dedicated to the simulation of FOWTs use several prescriptions detailed in this Subsec. The preliminary works realized up to now imply that some simplifications are still used in the design of the simulator, while other topics are not considered at all. In the following, Sec. 3.2 details the numerical framework for the resolution of the NS equations in ICI-tech, while Sec. 3.3 presents the implementations and advances relative to the simulation of FOWTs.

3.2 Monolithic finite element formulation for the incompressible Navier-Stokes equations

The incompressible Navier-Stokes equations, whose resolution is introduced in Sec. 3.1, are recalled in Eq. (3.8).

$$\left\{ \begin{array}{ll} \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega \\ \rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) - \eta \Delta \mathbf{u} + \nabla p = \mathbf{f} & \text{in } \Omega \\ \mathbf{u} = \alpha & \text{in } \Gamma_D \\ \sigma \cdot \mathbf{u} = \beta & \text{in } \Gamma_N \\ \mathbf{u}(t = 0) = \mathbf{u}_0 & \text{in } \Omega \end{array} \right. \quad (3.8)$$

As already mentioned, the resolution of the NS problem requires advanced computational techniques. This Sec. presents the Finite Element framework used in ICI-tech.

3.2.1 Finite Element formulation

A Finite Element (FE) approach is used in ICI-tech to handle the resolution of the NS problem. The theory of FE relies on the definition of Sobolev spaces, presented in Eqs. (3.9), on which the solution is approximated. These spaces are defined on the computational domain Ω , with $L^2(\Omega)$ the space of functions whose square is integrable on Ω .

$$H^1(\Omega) = \{u \in L^2(\Omega) \mid Du \in L^2(\Omega)\} \quad (3.9a)$$

$$\mathcal{V} = \{u \in L^2(\Omega) \mid u = \alpha \text{ in } \Gamma_D\} \quad (3.9b)$$

$$\mathcal{Q} = \left\{ u \in L^2(\Omega) \mid \int_{\Omega} u d\Omega = 0 \right\} \quad (3.9c)$$

$$\mathcal{V}_0 = \{u \in H^1(\Omega) \mid u = 0, \text{ at } \partial\Omega\} \quad (3.9d)$$

The scalar product of $L^2(\Omega)$ (\cdot, \cdot) is defined as following.

$$\forall (u, v) \in L^2(\Omega), (u, v) = \int_{\Omega} uv d\Omega \quad (3.10)$$

The FE method relies on a weak formulation of the equations considered. As opposed to the strong formulation, which writes $R(u) = f$ for R differential operator and u, f defined on Ω , the weak formulation considers the scalar product by a function defined on Ω . The weak formulation is written in Eqs. (3.11) for, respectively, the integral or the scalar product formulations.

$$\forall v \in L^2(\Omega), \int_{\Omega} R(u)vd\Omega = \int_{\Omega} fvd\Omega \quad (3.11a)$$

$$\forall v \in L^2(\Omega), (R(u), v) = (f, v) \quad (3.11b)$$

The resolution of Eqs. (3.11) consists in obtaining a solution u respecting the weak formulation, which is equivalent to solving $R(u) = f$. However, it is almost impossible to find an analytical solution to this type of problem, especially in the context of the NS equations. To that extent, a discretization of the problem is proposed. The computational domain Ω is discretized and split into simplexes \mathcal{K} . The set of \mathcal{K} , written \mathcal{T}_h , forms the computational mesh. New FE spaces can be defined, V_h and Q_h , presented in Eqs. (3.12), correspond to first-order FE written for a velocity/pressure problem. However, a FE formulation being P1 in velocity and in pressure does not respect the Brezzi-Babushka condition, which generates numerical instabilities. An enrichment of the velocity is performed thanks to the space $V_{0,h}$. This defines a P1+/P1 FE formulation, abbreviated P1-FE in the following. The spaces use the notations $C^0(\Omega)$ and $P^1(\mathcal{K})$ describing respectively the space of functions continuous on Ω and the space of polynomials of degree 1 on \mathcal{K} .

$$V_h = \{u_h \mid u_h \in C^0(\Omega), \forall \mathcal{K}, u_{h|\mathcal{K}} \in P^1(\mathcal{K})^n\} \quad (3.12a)$$

$$V_{0,h} = \{u_h \mid u_h \in V_h, \forall \mathcal{K}, u_h = 0 \text{ on } \partial\mathcal{K}\} \quad (3.12b)$$

$$Q_h = \{p_h \mid p_h \in C^0(\Omega), \forall \mathcal{K}, p_{h|\mathcal{K}} \in P^1(\mathcal{K})\} \quad (3.12c)$$

The resolution of the FE problem is achieved on each element \mathcal{K} through a linear combination of the contributions of each point i of \mathcal{K} . Shape functions N_i of $P^1(\mathcal{K})$ are defined, respecting $N_i(j) = 1$ if $i = j$ and $N_i(j) = 0$ if $i \neq j$, for i and j of \mathcal{K} . The interpolated solution at a point X located in \mathcal{K} , simplex composed of $D_{\mathcal{K}}$ nodes, is written in Eqs. (3.13). The value of u at the node i is written u_i . In the context of P1-FE, $u_{h|\mathcal{K}}$ for a random $x \in \mathcal{K}$ depends only on node values. In the context of higher-order FE, more interpolation points are required. The same behavior is found for p_h .

$$u_{h|\mathcal{K}}(x) = \sum_{i=1}^{D_{\mathcal{K}}} u_i N_i(x) \quad (3.13)$$

The FE approach considers an interpolated field in each element \mathcal{K} . The precision of the results directly depends on the order of interpolation, i.e., the degree of the polynomials N_i . In the context of P1-FE, the shape functions N_i are of order 1, which translates to ∇u_h and ∇p_h being constant in \mathcal{K} . This approximation generates errors, which can be tackled in different ways. The interpolation order can be increased, the size of \mathcal{K} can be reduced.

The resolution of the Navier-Stokes equations can not be directly realized with the P1+/P1 formalism introduced. First of all, as mentioned in Subsec. 3.1.1, the turbulence of the flow needs to be modeled. Several approaches have been proposed, which rely on the modification of the NS equations to characterize the subgrid-scale flow. The second issue with the current P1+/P1 paradigm appears when convective flows are considered. The solutions tend to integrate numerical oscillations, which require the addition of stabilization terms to be suppressed. As Re numbers around WT blades have an order of magnitude of 10^6 , the flows are clearly convection-dominant. A resolution performed in a P1+/P1 formulation would thus generate numerical oscillations. Instead of considering modified NS equations, the methodology followed in this thesis uses a Variational MultiScale (VMS) paradigm. This formulation proposes an analytical approximation of the subgrid-scale turbulence, which integrates stabilization terms suited for a FE resolution. In the following, the resolution of the incompressible NS equations in the VMS framework is described.

3.2.2 Variational MultiScale of the Navier-Stokes equations

The weak form of the incompressible NS problem is presented in a variational form in Eq. (3.14) with \mathbf{u} the velocity and p the pressure. From here on, a differentiation is established in the notations between velocity and pressure, as they are represented respectively by vector and scalar fields. Compared to the NS problem written in Eq. (3.8).

$$\begin{cases} \forall(\mathbf{v}, q) \in V_0 \times Q, \\ \rho(\partial_t \mathbf{u}, \mathbf{v}) + \rho(\mathbf{u} \cdot \nabla \mathbf{u}, \mathbf{v}) + 2\eta \varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v}) - (p, \nabla \cdot \mathbf{v}) = (\mathbf{f}, \mathbf{v}) \\ (\nabla \cdot \mathbf{u}, q) = 0 \end{cases} \quad (3.14)$$

ICI-tech uses a Variational MultiScale (VMS) formulation as presented by Hachem et al. (2010) to solve the NS problem. The variational formulation considers a FE expression of the NS equations, with the weak form presented previously. The VMS paradigm operates as an implicit-LES (Large Eddy Simulation). Velocity and pressure are split between the coarse scales, which are solved, and sub-grid scales, which are modeled. To link this with the developments proposed in Subsec. 3.2.1, the coarse-scale contributions correspond to the results obtained from interpolation on each simplex \mathcal{K} . The subgrid-scale ones correspond to the error produced by the FE approximation. The complementary spaces to V , V_0 and Q can be defined. They are presented in Eqs. (3.15). The scale separation induced in velocity in pressure is written in Eqs. (3.16).

$$V = V_h + V' \quad (3.15a)$$

$$V_0 = V_{0,h} + V'_0 \quad (3.15b)$$

$$Q = Q_h + Q' \quad (3.15c)$$

$$\mathbf{u} = \mathbf{u}_h + \mathbf{u}' \quad (3.16a)$$

$$p = p_h + p' \quad (3.16b)$$

The NS problem obtained after this scale separation is presented in Eq. (3.17). This set of equations features both coarse and small scales. In LES, a viscous term is incorporated in the NS equations to compensate for the effect of the subgrid-scale turbulence. In VMS, the NS problem is reformulated to add implicitly the contributions from the subgrid-scale velocities in the formulation.

$$\begin{aligned} \forall(\mathbf{v}_h, q_h) \in V_h \times Q_h, \forall(\mathbf{v}', q') \in V' \times Q', \\ \left\{ \begin{aligned} &\rho(\partial_t(\mathbf{u}_h + \mathbf{u}'), (\mathbf{v}_h + \mathbf{v}'))_{\Omega} + \rho((\mathbf{u}_h + \mathbf{u}') \cdot \nabla(\mathbf{u}_h + \mathbf{u}'), (\mathbf{v}_h + \mathbf{v}'))_{\Omega} \\ &+ (2\eta\varepsilon(\mathbf{u}_h + \mathbf{u}') : \varepsilon(\mathbf{v}_h + \mathbf{v}'))_{\Omega} - ((p_h + p'), \nabla \cdot (\mathbf{v}_h + \mathbf{v}'))_{\Omega} = (\mathbf{f}, (\mathbf{v}_h + \mathbf{v}'))_{\Omega} \\ &(\nabla \cdot (\mathbf{u}_h + \mathbf{u}'), (q_h + q'))_{\Omega} = 0 \end{aligned} \right. \end{aligned} \quad (3.17)$$

The orthogonality between coarse and small scales offer to split Eq. (3.17) into a coarse-scale and a sub-scale problem. They are written respectively in Eq. (3.18) and Eq. (3.19).

$$\begin{aligned} \forall(\mathbf{v}_h, q_h) \in V_h \times Q_h, \forall(\mathbf{v}', q') \in V' \times Q', \\ \left\{ \begin{aligned} &\rho(\partial_t(\mathbf{u}_h + \mathbf{u}'), \mathbf{v}_h)_{\Omega} + \rho((\mathbf{u}_h + \mathbf{u}') \cdot \nabla(\mathbf{u}_h + \mathbf{u}'), \mathbf{v}_h)_{\Omega} \\ &+ (2\eta\varepsilon(\mathbf{u}_h) : \varepsilon(\mathbf{v}_h))_{\Omega} - (p_h + p', \nabla \cdot \mathbf{v}_h)_{\Omega} = (\mathbf{f}, \mathbf{v}_h)_{\Omega} \\ &(\nabla \cdot (\mathbf{u}_h + \mathbf{u}'), q_h)_{\Omega} = 0 \end{aligned} \right. \end{aligned} \quad (3.18)$$

$$\begin{aligned} \forall(\mathbf{v}_h, q_h) \in V_h \times Q_h, \forall(\mathbf{v}', q') \in V' \times Q', \\ \left\{ \begin{aligned} &\rho(\partial_t(\mathbf{u}_h + \mathbf{u}'), \mathbf{v}')_{\Omega} + \rho((\mathbf{u}_h + \mathbf{u}') \cdot \nabla(\mathbf{u}_h + \mathbf{u}'), \mathbf{v}')_{\Omega} + (2\eta\varepsilon(\mathbf{u}') : \varepsilon(\mathbf{v}'))_{\Omega} \\ &- (p_h + p', \nabla \cdot \mathbf{v}')_{\Omega} = (\mathbf{f}, \mathbf{v}')_{\Omega} \\ &(\nabla \cdot (\mathbf{u}_h + \mathbf{u}'), q')_{\Omega} = 0 \end{aligned} \right. \end{aligned} \quad (3.19)$$

The VMS formulation relies on a reformulation of the fine scale problem. Some simplifications can be considered in Eq. (3.19). The crossed viscous terms can be neglected. The pressure subscales are modeled, see Eqs. (3.20). Two more approximations are considered by Hachem et al. (2010). The subscales are treated as quasi-static, which returns $\partial_t \mathbf{u}' = 0$. The subscale velocity is neglected in convective velocity, so that $(\mathbf{u}_h + \mathbf{u}') \cdot \nabla(\mathbf{u}_h + \mathbf{u}') \approx \mathbf{u}_h \cdot \nabla(\mathbf{u}_h + \mathbf{u}')$. These last two approximations are mentioned in Codina et al. (2017) as perturbing the NS problem.

Stabilization terms and residuals are introduced in Eqs. (3.20), along with the approximation realized for the fine-scale pressure p' . The expression of τ_C is inherited from Codina (2000), where c_1 and c_2 are two constants and h defines a characteristic length of the element.

$$p' \approx \tau_C \mathcal{R}_C \quad (3.20a)$$

$$\mathcal{R}_M = \mathbf{f} - \rho \partial_t \mathbf{u}_h - \rho(\mathbf{u}_h + \mathbf{u}') \cdot \nabla \mathbf{u}_h - \nabla p_h \quad (3.20b)$$

$$\mathcal{R}_C = -\nabla \cdot \mathbf{u}_h \quad (3.20c)$$

$$\tau_C = \left(\left(\frac{\eta}{\rho} \right)^2 + \left(\frac{c_2 \|\mathbf{u}\|_{\mathcal{K}}}{c_1 h} \right)^2 \right)^{1/2} \quad (3.20d)$$

A simplified subscale problem is obtained.

$$\begin{aligned} \forall(\mathbf{v}_h, q_h) \in V_h \times Q_h, \forall(\mathbf{v}', q') \in V' \times Q', \\ \left\{ \begin{aligned} &\rho(\mathbf{u}_h \cdot \nabla \mathbf{u}', \mathbf{v}')_{\Omega} + (2\eta\varepsilon(\mathbf{u}') : \varepsilon(\mathbf{v}'))_{\Omega} + (\nabla p', \mathbf{v}')_{\Omega} = (\mathcal{R}_M, \mathbf{v}')_{\Omega} \\ &(\nabla \cdot \mathbf{u}', q')_{\Omega} = (\mathcal{R}_C, q')_{\Omega} \end{aligned} \right. \end{aligned} \quad (3.21)$$

The resolution of velocity/pressure equations requires an enrichment of the formulation. Residual-free bubble shape functions $b_{\mathcal{K}}$ are defined. The small scale velocities can be rewritten as in Eq. (3.22).

$$\mathbf{u}' = \sum_{\mathcal{K} \in \mathcal{T}_{\mathcal{K}}} \mathbf{u}'_{\mathcal{K}} b_{\mathcal{K}} \quad (3.22)$$

The P1 characteristics of the FE considered here offer to express \mathbf{u}' at the scale of each simplex \mathcal{K} as a function of \mathcal{R}_M . This approximation can be combined with an upwind correction of the bubble shape functions, to stabilize the convective terms of the Navier-Stokes equations. The formulation obtained is similar to a Streamline Upwind Petrov-Galerkin (SUPG) framework, with small-scale velocities \mathbf{u}' , which can finally be expressed as depicted in Eqs. (3.23). The term $b_{\mathcal{K}}^u$ refers to the upwind corrections brought to the bubbles.

$$\mathbf{u}'_{\mathcal{K}} = \tau_{\mathcal{K}} \mathcal{R}_M \quad (3.23a)$$

$$\tau_{\mathcal{K}} = \frac{b_{\mathcal{K}} \int_{\mathcal{K}} (b_{\mathcal{K}} + b_{\mathcal{K}}^u) d\Omega}{\rho (\mathbf{u}_h \cdot \nabla b_{\mathcal{K}}, b_{\mathcal{K}}^u)_{\mathcal{K}} + (2\mu \varepsilon (b_{\mathcal{K}}) : \varepsilon (b_{\mathcal{K}}))_{\mathcal{K}}} \quad (3.23b)$$

Substituting back the fine-scale equations obtained into the coarse-scale ones, the final NS VMS problem can be defined. The full problem, expressed with P1-FE and solved in ICI-tech, is presented in Eq. (3.24)

$$\begin{aligned} \forall (\mathbf{v}_h, q_h) \in V_{0,h} \times Q_h, \\ \left\{ \begin{array}{l} \rho (\partial_t \mathbf{u}_h + \mathbf{u}_h, \nabla \mathbf{u}_h, \mathbf{v}_h)_{\Omega} + (2\eta \varepsilon (\mathbf{u}_h) : \varepsilon (\mathbf{v}_h))_{\Omega} - (p_h, \nabla \cdot \mathbf{v}_h)_{\Omega} + (\nabla \mathbf{u}_h, q_h)_{\Omega} \\ - (\mathbf{f}, \mathbf{v}_h)_{\Omega} + \sum_{K \in \mathcal{T}_h} \tau_K (\rho (\partial_t \mathbf{u}_h + \mathbf{u}_h \cdot \nabla \mathbf{u}_h) + \nabla p_h - \mathbf{f}, \rho \mathbf{u}_h \nabla \mathbf{v}_h)_K \\ + \sum_{K \in \mathcal{T}_h} \tau_K (\rho (\partial_t \mathbf{u}_h + \mathbf{u}_h \cdot \nabla \mathbf{u}_h) + \nabla p_h - \mathbf{f}, \nabla q_h)_K \\ + \sum_{K \in \mathcal{T}_h} (\tau_C \nabla \cdot \mathbf{u}_h, \nabla \cdot \mathbf{v}_h)_K = 0 \\ (\nabla \cdot \mathbf{u}_h, q_h) - \sum_{\mathcal{K} \in \mathcal{T}_h} (\tau_{\mathcal{K}} \mathcal{R}_M, \nabla \cdot q_h)_{\Omega} = 0 \end{array} \right. \quad (3.24) \end{aligned}$$

The characteristic length of the element h is evaluated locally. In the context of anisotropic meshing, the determination of h is a challenge. For an element \mathcal{K} , edges of various sizes can be found, without any guarantee on their orientations. To account for this effect, Coupez and Hachem (2013) used a local definition of h on each simplex \mathcal{K} , where the local velocity of the flow is considered. This expression is presented in Eq. (3.13), with $\mathcal{V}_{\mathcal{K}}$ the vertices of \mathcal{K} and N_i the shape functions of \mathcal{K} .

$$h_{\mathcal{K}} = \frac{2 |\mathbf{u}_h|}{\sum_{i=1}^{\mathcal{V}_{\mathcal{K}}} |\mathbf{u}_h \cdot \nabla N_i|} \quad (3.25)$$

3.2.3 Mixing laws

The incompressible NS equations require the knowledge of the characteristics of the flow at each point of the computational mesh. In particular, the viscosity η and the density ρ define the behavior fluid flows for a given problem. Mixing laws are applied to interpolate them from the properties of the different materials or fluids in the simulation.

The context of multiphase flows requires the representation of several phases. To that extent, different mixing laws have to be applied. In ICI-tech, these mixing laws are resolved successively. For the immersion of two phases of level-set Φ_1 and Φ_2 , the procedure presented in Eqs. (3.26) is followed. The viscosities η_1 , η_2 and η_d characterize the two phases and the computational domain. The level-set functions allow to design

Heaviside functions H_1 and H_2 . The procedure presented hereafter is valid for a three-phases flow using a linear law for each immersion. A generalization can obviously be realized to more complicated multiphase flows, with more phases and different laws.

$$\eta' = H\eta_1 + (1 - H)\eta_d \quad (3.26a)$$

$$\eta = H\eta_2 + (1 - H)\eta' \quad (3.26b)$$

A linear law based on a Heaviside function H is presented in Eq. (3.27) for the immersion of an object of viscosity η_o in a domain of viscosity η_d . The construction of H is performed at each computational point, which helps to build a scalar field for η . The definition of H tends to characterize the affiliation of the current point to the phases considered, which allows a correct interpolation.

$$\eta = H\eta_o + (1 - H)\eta_d \quad (3.27)$$

Different choices are available concerning the Heaviside function used for the immersion. The only requirement for H is to define a bijection in the transition area returning results between 0 and 1. However, a constraint arises from the numerical framework used in ICI-tech. For the case of two fluids, the only information known at each time step about their interface is provided by the level-set function Φ_ε . Consequently, H needs to be defined from Φ_ε for the construction of fluid/fluid mixing law. For the sake of generalization, the definition of H is performed in ICI-tech from Φ_ε , and is thus written H_ε .

The compatibility of the mixing laws with the adaptive meshing presented in Sec. 2.2 is very interesting. The mixing law rules the evolution of η and ρ , and require a fine meshing to be represented precisely at the interfaces. In ICI-tech, the adaptive meshing is, in particular, defined by Φ_ε . This implies the mesh to be adapted in the region where the level-set function evolves.

The definition of the mixing law needs to account for this, as the evolution of η and ρ needs to be concentrated in the region where the gradient of Φ_ε is the highest. If not, the smoothing of the fluid properties can be imprecise, does not represent well the problem simulated, and computational issues may arise. In ICI-tech, this is achieved through the introduction of a parameter e_{mix} , which characterizes the width of the mixing region. The level-set function Φ_ε , defined in Sec. 2.1 for α signed-distance, is recalled in Eq. (3.28). An option for H_ε , built using Φ_ε to answer the meshing issues, is presented in Eq. (3.29) and plotted in Fig. 3.5.

$$\Phi_\varepsilon(\alpha) = \varepsilon \tanh\left(\frac{\alpha}{\varepsilon}\right) \quad (3.28)$$

$$H_\varepsilon(\Phi_\varepsilon) = \begin{cases} 0 & \text{if } \frac{1}{2} \left(1 + \frac{\Phi_\varepsilon}{e_{mix}}\right) < -e_{mix} \\ \frac{1}{2} \left(1 + \frac{\Phi_\varepsilon}{e_{mix}}\right) & \text{if } -e_{mix} < \frac{1}{2} \left(1 + \frac{\Phi_\varepsilon}{e_{mix}}\right) < e_{mix} \\ 1 & \text{if } \frac{1}{2} \left(1 + \frac{\Phi_\varepsilon}{e_{mix}}\right) > e_{mix} \end{cases} \quad (3.29)$$

The illustration highlights the importance of a thinner mixing area. The level-set figures the regions where the mesh is refined, mostly where the Hessian of Φ_ε is important. These areas coincide for $e_{mix} = \varepsilon/2$, which guarantees a good representation of the interfaces. On the contrary, H tends to widen as e_{mix} gets closer from ε , which opens the door to variations in η in coarsely-meshed zones. With $e_{mix} = \varepsilon$, H_ε defines mixing laws of width ε . As the transition in fluid properties must be spread on several mesh

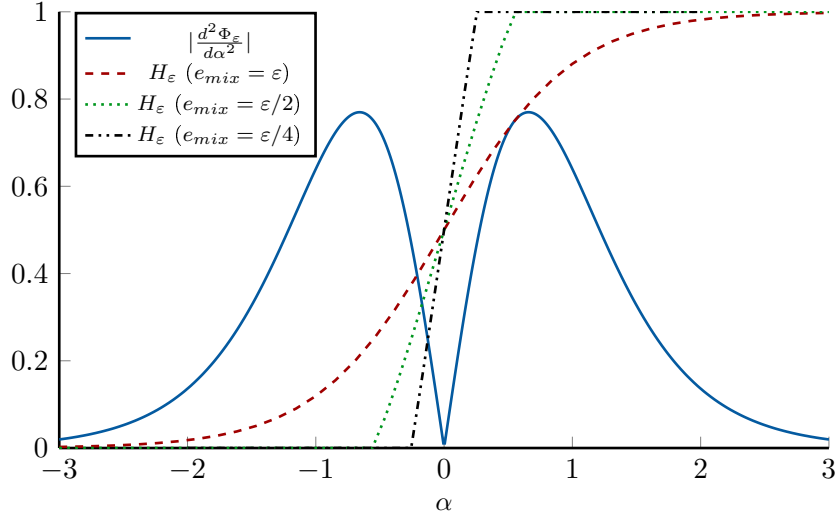


Figure 3.5: Second derivative of Φ_ε (Eq. (3.28)) and H_ε (Eq. (3.29)) for $\varepsilon = 1$ and various e_{mix} .

cells, $e_{mix} = \varepsilon/4$ is rejected. Finally, $e_{mix} = \varepsilon/2$ seems to be the most efficient option among the three proposed in Fig. 3.5.

The drop of $\frac{d^2\Phi_\varepsilon}{d\alpha^2}$ around the $\alpha = 0$ shows the limits of the adaptation based on Φ_ε only. The refinement of mesh cells in this region is not constrained, which can lead to important errors. However, the mixing law is quasi-linear in this region, thanks to this characteristic of the hyperbolic tangent. Consequently, only a few mesh cells allow producing a small interpolation error. Moreover, this area is located in-between two highly refined regions. The coherence of the mesh generated tends to reduce the target size of cells, which generates a "proximity refinement". If the precision achieved using the mesh generated automatically using Φ_ε is not sufficient, the adaptation can be enriched thanks to recourse to a Dirac function δ_{adapt} , defined as a derivative of H_ε . The adaptation is then done on $\Phi_\varepsilon + \delta_{adapt}$. The expression of δ_{adapt} is presented in Eq. (3.30), and a comparison between Φ_ε and δ_{adapt} is drawn in Fig. 3.6. The addition of the Dirac function in the adaptation procedure allows to consider the plain curve from Fig. 3.6 to obtain the mesh concentration. The adapted area will feature more points around $\alpha = 0$, which improves the description of the interfaces.

$$\delta_{adapt} = \frac{1}{2\varepsilon} \left(1 - \left(\frac{\Phi_\varepsilon}{\varepsilon} \right)^2 \right) \quad (3.30)$$

The construction of η proposed in Eq. (3.27) uses a linear interpolation. This law is the most straightforward to build. However, when large disparities are encountered, other laws may be more appropriated. This approach may be required for, e.g., the density drop observed at the surface of the sea, where the order of magnitude of the ration ρ_{water}/ρ_{air} is of 10^3 . Different laws can be used for the interpolation. Logarithmic and inverse laws are presented in Eqs. (3.31), and drawn in Fig. 3.7. The variety of interpolations available, along with the different options for H_ε , enable to design mixing laws suited for various problems. The inverse and logarithmic laws are valid for strictly positive values, which is always the case for the physical properties η and ρ .

$$\log(\eta) = H_\varepsilon \log(\eta_o) + (1 - H_\varepsilon) \log(\eta_d) \quad (3.31a)$$

$$\frac{1}{\eta} = H_\varepsilon \frac{1}{\eta_o} + (1 - H_\varepsilon) \frac{1}{\eta_d} \quad (3.31b)$$

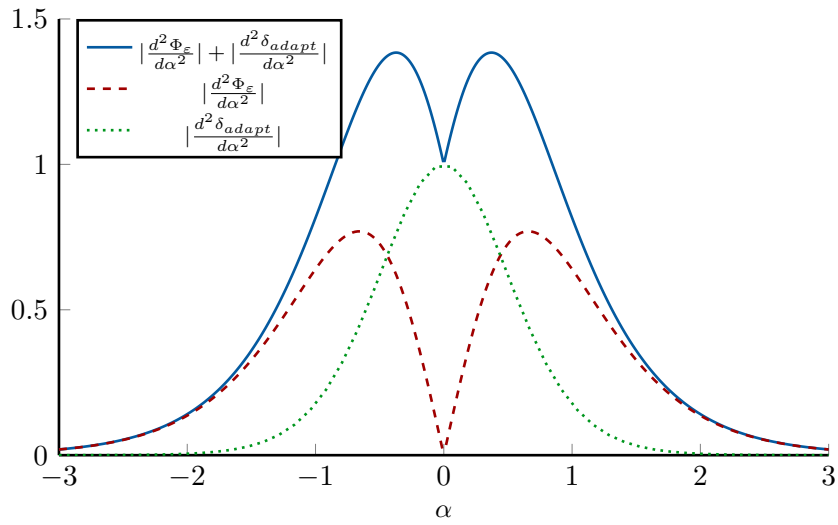


Figure 3.6: Second derivatives of Φ_ε (Eq. (3.28)) and δ_{adapt} (Eq. (3.30)) for $\varepsilon = 1$.

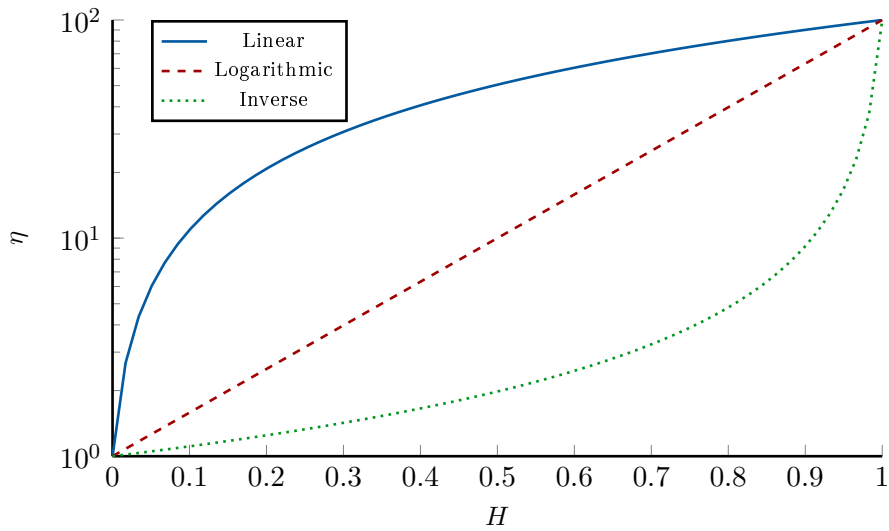


Figure 3.7: Different interpolation techniques applicable to mixing laws.

3.2.4 Movements of phases

During a simulation of FOWTs, the phases represented in the domain, e.g., the floating structure, the rotor or the sea, need to be moved. Two different situations need to be distinguished, the prescribed movements and the solved ones. To draw a comparison with a simulation of FOWTs, the rotation of WT blades can be user-defined, i.e., prescribed since the beginning of the simulation, or solved, i.e., deduced from the aerodynamic efforts applied on it. Independently of the type of movements considered, the determination of the new phase follows the same objective: the level-set Φ_ε at the next time iteration needs to be computed. To that extent, the new Φ_ε needs to respect all the mathematical requirements of a level-set function, but also has to accurately represent the phase. Among the properties, the gradient of the level-set near the interface Γ has to respect $\|\nabla\Phi_\varepsilon\|_\Gamma = 1$, which can require particular numerical treatment. The accuracy of the representation is mostly dependent on good compatibility between the physical problem simulated, the simulation parameters and the mesh adaptation.

When the movements of the phase to displace are prescribed, this step is straightforward. The new phase is known explicitly, with a variety of options available. The signed-distance α or the level-set function Φ_ε can be directly prescribed, or the movements can be defined from the coordinate system of the object. This paradigm is often followed for rigid objects, which follow a simple path. In this situation, the tracking of the interface with time is simplified. The prescribed definition of the parameters allow a complete reconstruction of the phases at each time increment, which respects $\|\nabla\Phi_\varepsilon\|_\Gamma = 1$ by nature.

When the movements of the phases need to be solved, the determination of the new positions is more challenging. The geometry represented by a Heaviside function needs to be moved, and eventually deformed, from the velocities of the surrounding flow. Once again, two different situations can be considered: the solved movements of rigid solids, whose treatment is detailed in the context of buoyancy in Subsec. 3.3.3, and the movements of fluids. The movement of deformable solids is a last case, which is not considered in this thesis. Even if, for accurate simulation of FOWTs, the aeroelasticity corresponds to this type of solved movements.

The fluid flows impose the tracking of the interface, represented by Φ_ε , during the simulation. The movements of fluids are realized in ICI-tech through the resolution of an advection problem, which solves the transport of level-set functions. The advection equation is presented in Eq. (3.32) for a fluid phase of frontier Γ represented by a level-set function Φ_ε of characteristic width ε , and with \mathbf{u} the velocity of the flow. The transport is solved by considering the hyperbolic equation.

$$\begin{cases} \frac{\partial\Phi_\varepsilon}{\partial t} + \mathbf{u} \cdot \nabla\Phi_\varepsilon = 0 \\ \Phi_\varepsilon(t=0) = \Phi_{\varepsilon,0} \end{cases} \quad (3.32)$$

By construction, the gradient of Φ_ε needs to be 1 around Γ . However, the resolution of this advection equation does not conserve this behavior. The gradient of the level-set advected Φ_ε^a does not respect $\|\nabla\Phi_\varepsilon^a\|_\Gamma = 1$. A correction step needs to be added for Φ_ε to respect the gradient condition at the next time iteration. A common approach is to reinitialize Φ_ε^a through the resolution of a Hamilton–Jacobi equation, which introduces a virtual time to reconstruct Φ_ε from Φ_ε^a . This reinitialization step considers Φ_ε^a , and performs a redistancing, to respect the condition on the gradient at Γ . This procedure allows obtaining a level-set at the next time step respecting all the condition enumerated previously. However, the resolution of multiple problems, the advection one and some reinitialization steps, is required before obtaining the distribution of Φ_ε at the next time step.

In ICI-tech, the transport and the reinitialization are combined in a single problem. The convective-reinitialization method, developed by [Ville et al. \(2011\)](#), is used. This approach allows to both transport and reinitialize the gradient of the level-set, through the resolution of a single problem. A virtual time τ and a function Φ_ε^v , sharing the zeros of Φ_ε^a , are introduced. The convective reinitialization problem, written in Eq. [\(3.33\)](#) for Φ_ε^v , contains transport and redistancing features. The sign function S is defined, and $g(\Phi_\varepsilon) = 1 - (\Phi_\varepsilon/\varepsilon)^2$ corresponds to the analytical gradient of the level-set function, computed from a derivation of the theoretical expression of Φ_ε . This new problem is written with a total derivative instead of a partial one, as an Eulerian framework is used.

$$\begin{cases} \frac{d\Phi_\varepsilon^v}{d\tau} + S(\Phi_\varepsilon^a) (\|\nabla\Phi_\varepsilon^v\|_2 - g(\Phi_\varepsilon^v)) = 0 \\ \Phi_\varepsilon^v(\tau = 0) = \Phi_\varepsilon^a(t) \end{cases} \quad (3.33)$$

A velocity \mathbf{U} and a parameter λ are introduced in Eq. [\(3.34\)](#).

$$\mathbf{U} = S(\Phi_\varepsilon^a) \frac{\nabla\Phi_\varepsilon^v}{\|\nabla\Phi_\varepsilon^v\|_2} \quad (3.34a)$$

$$\lambda = \frac{d\tau}{dt} \quad (3.34b)$$

A reformulation of Eq. [\(3.33\)](#) to bring \mathbf{U} out allows reformulating this set of equations as an advection problem, set down with τ . A multiplication of this advection problem by λ enables to come back to time derivatives. The problem obtained is presented in Eq. [\(3.35\)](#).

$$\begin{cases} \frac{d\Phi_\varepsilon^v}{dt} + \lambda\mathbf{U} \cdot \nabla\Phi_\varepsilon^v = \lambda S(\Phi_\varepsilon^a) g(\Phi_\varepsilon^v) \\ \Phi_\varepsilon^v(\tau = 0) = \Phi_\varepsilon^a(t) \end{cases} \quad (3.35)$$

The recourse to Φ_ε^v is not necessary, as the resolution is performed in one step. The final convective-reinitialization is presented in Eq. [\(3.37\)](#). This formulation integrates the expression of the derivative of Φ_ε drawn in Eq. [\(3.36\)](#). This final problem allows solving the transport, while conserving the correct description of $\nabla\Phi_\varepsilon$.

$$\frac{d\Phi_\varepsilon}{dt} = \frac{\partial\Phi_\varepsilon}{\partial t} + \mathbf{u} \cdot \nabla\Phi_\varepsilon \quad (3.36)$$

$$\begin{cases} \frac{\partial\Phi_\varepsilon}{\partial t} + (\mathbf{u} + \lambda\mathbf{U}) \cdot \nabla\Phi_\varepsilon = \lambda S(\Phi_\varepsilon^a) g(\Phi_\varepsilon^v) \\ \Phi_\varepsilon(t = 0) = \Phi_{\varepsilon,0} \end{cases} \quad (3.37)$$

The term $d\Phi_\varepsilon/dt$ is linearized using a backward Euler time scheme, which allows building a linear system. The numerical instabilities due to the convection, discussed in Subsec. [3.2.2](#), are eliminated thanks to the addition of SUPG terms. The stabilized linear system is finally solved using PETSc.

3.2.5 Monolithic resolution of the Navier-Stokes equations

The resolution of the flow problem is realized in ICI-tech using the utilities previously presented in this Sec. The problem of Eq. [\(3.24\)](#) needs to be transformed into a linear system $Ax = b$, which is solved using PETSc. Several terms presented in this Sec. are not suited for a resolution. In particular, the time derivatives and non-linear terms found in Eq. [\(3.24\)](#) have to be adapted to be handled numerically.

The time dependency is handled through time discretization. The formulation proposed in ICI-tech uses an Euler implicit scheme. This means that, to compute $x_n = x(t = t_n)$, any x in $Ax = b$ is considered as x_n itself. This behavior tends to enrich the

matrix A , and leads to a higher complexity during the implementation, compared to an explicit scheme considering x_{n-1} . However, much better robustness is obtained with this choice. The order of evolution of the error is of 1, which means that, when the time step is divided by N , the error is divided by N too. The time discretization should always be regarded in perspective with the spatial discretization. Those two methods enable to get a solution to the NS equations, but both produce errors, which have to be kept in the same orders of magnitude.

The advection term $\mathbf{u} \cdot \nabla \mathbf{u}$ in Eq. (3.24) is non-linear. To build a linear problem, this term must be linearized. In ICI-tech, a linearization in time is performed, based on the implicit Euler time scheme discussed previously. The details of the linearization are presented in Eqs. (3.38).

$$\begin{aligned}
\mathbf{u}_{n+1} \cdot \nabla \mathbf{u}_{n+1} &= (\mathbf{u}_n + (\mathbf{u}_{n+1} - \mathbf{u}_n)) \cdot \nabla (\mathbf{u}_n + (\mathbf{u}_{n+1} - \mathbf{u}_n)) \\
&= \mathbf{u}_n \cdot \nabla \mathbf{u}_n + (\mathbf{u}_{n+1} - \mathbf{u}_n) \cdot \nabla \mathbf{u}_n + \mathbf{u}_n \cdot \nabla (\mathbf{u}_{n+1} - \mathbf{u}_n) \\
&\quad + (\mathbf{u}_{n+1} - \mathbf{u}_n) \cdot \nabla (\mathbf{u}_{n+1} - \mathbf{u}_n) \\
&= \mathbf{u}_{n+1} \cdot \nabla \mathbf{u}_n + \mathbf{u}_n \cdot \nabla \mathbf{u}_{n+1} - \mathbf{u}_n \cdot \nabla \mathbf{u}_n
\end{aligned} \tag{3.38}$$

In the last step of the procedure, the term $(\mathbf{u}_{n+1} - \mathbf{u}_n) \cdot \nabla (\mathbf{u}_{n+1} - \mathbf{u}_n)$ disappears. This approximation is realized because the term generated is of the second order in time, as it can be verified with a Taylor expansion.

Once the matricial problem has been set up, ICI-tech takes advantage of PETSc to solve the final system in parallel. A preconditioner is used to optimize the conditioning of the matrix, here an Incomplete LU by blocks. Iterative solvers are used to solve the NS VMS problem in a P1-FE formulation. A Biconjugate Gradient Stabilized solver has been used during the thesis.

3.3 Numerical utilities of ICI-tech for the simulation of FOWTs

To simulate FOWTs in real operational conditions, the resolution of the NS equations is only a small part of the problem. The rotor, the hydrodynamic behavior of the floating platform or the moorings are among the subjects that have to be addressed.

In ICI-tech, the representation of a rigid rotor is realized using a full-geometry reconstruction. The numerical assets presented in Chap. 2 handle this step by themselves, and no further implementations have been needed to accurately model a static WT rotor. The developments included in ICI-tech focused on the rotation of the whole structure. The reconstruction of the moving rotor relies on successive immersions of the rotor mesh. The implementation of the octree was able to largely reduce the costs linked to mesh immersion. This postponed the implementation of a utility rotating the computational mesh to follow the rapid movements of the rotor. Prescribed rotation of the WT blades only requires the correction of the positioning of the rotor immersed mesh. The resolution of solved movements for the rotor requires the computation of the efforts applied on the blades. This evaluation of forces is described in Subsec 3.3.1

The hydrodynamics can be viewed from two different points of view. The first one considers the impact of the sea on the FOWT. To that extent, an accurate representation of real sea states is important. In particular, the numerical generation of waves is a key feature of a FOWT simulator. Simulation tools can only work with domains of finite size, and small mesh cells are required for the representation of the free-surface. FSI is required in the context of FOWTs, which is expensive computationally. All these

reasons force the recourse to small, bounded domains, where waves have to be generated artificially. The different developments realized concerning numerical wave tanks are presented in Subsec. [3.3.2](#).

The second one focuses on the motion induced on and by the floating structure. The incoming wave field has an impact on the FOWT, which generates loads. The floating structure is thus put in movement. The definition of those displacements is necessary to obtain an accurate simulation of a FOWT. The displacements of the FOWT then have a direct impact on the surrounding water, which perturbs the hydrodynamics. The buoyancy and FSI simulation tools are presented in Subsec. [3.3.3](#). Even if this thesis has not been focused on moorings, this subject is briefly discussed in this Subsec. too.

3.3.1 A posteriori computation of forces applied on an immersed body

The computation of the force applied on an object is not trivial in the context of mesh immersion, as the adapted mesh obtained is not body-fitted. The absence of points located exactly on the frontier of the immersed geometries prevents the direct acquisition of the loads. However, the concentration of points around the interfaces enables to measure those forces rather accurately. Instead of measuring the local normal constraint T_{local} applied on the exact boundary of an object ω of interface Γ , as in Eq. [\(3.39\)](#), the force needs to be measured on a volume surrounding ω . This is done using the approach developed by [Brackbill et al. \(1992\)](#), which uses a Dirac function δ_ε , presented in Eq. [\(3.41\)](#). This Dirac function is built as a derivative of the Heaviside function H_ε , which respects Eq. [\(3.42\)](#). The function peaks for $\Phi_\varepsilon = 0$, and progressively decays as it moves away from Γ . The integral of T_{local} pondered by δ_ε is considered, which allows to measure an approximation \mathbf{F}_ε of \mathbf{F} , written in Eq. [\(3.40\)](#). The influence of the flows on Γ follows the trend of δ_ε . The contributions of T_{local} are maximal around Γ , and progressively decay.

$$\mathbf{F} = \int_{\Gamma} \mathbf{T}_{local} dS \quad (3.39)$$

$$\mathbf{F}_\varepsilon = \int_{\Omega} \delta_\varepsilon \mathbf{T}_{local} dV \quad (3.40)$$

$$\delta_\varepsilon = \frac{1}{2\varepsilon} \left(1 - \left(\frac{\Phi_\varepsilon}{\varepsilon} \right)^2 \right) = \nabla H_\varepsilon \quad (3.41)$$

$$\int_{\Omega} \delta_\varepsilon dV \xrightarrow{\varepsilon \rightarrow 0} \int_{\Gamma} dS \quad (3.42)$$

Different Dirac functions δ can be defined for the computation of forces. Several constraints exist though. If ω is deformable, the only information characterizing Γ known at each time step is Φ_ε . Consequently, the definition of δ is generalizable if its expression is built from Φ_ε . The connection between the computation of forces and the mesh adaptation procedure needs to be regarded too. The support of δ can be defined as the width of the region where $\delta \neq 0$. The support of the Dirac function used needs to be included inside the highly adapted regions. Regarding this definition, the support of δ_ε is infinite, even if δ_ε rapidly tends towards 0. A virtual support can thus be defined for δ_ε using a threshold. This could help to evaluate the compatibility of δ_ε with the mesh adaptation based on Φ_ε . However, as $\delta_\varepsilon = \nabla H_\varepsilon$, the mesh generated will naturally be adapted in the areas where δ_ε has influence. Finally, even if δ does not need to be even by definition, this characteristic facilitates some developments.

The forces are computed *a posteriori* from the results of velocity \mathbf{u} and pressure p obtained after the NS resolution. A reconstructed velocity gradient is built and the

stress tensor σ is computed at every point of the mesh, with η the viscosity and \mathbb{I} the identity matrix. The local normal stress is computed in Eq. (3.43a). The definition of a vector normal to ω is necessary at that point, but as the computational points are not disposed along Γ , its determination is not immediate. It has been chosen to define \mathbf{n} as a normalized gradient of Φ_ε . This allows computing \mathbf{n} at every point of the mesh using existing data. This also sets a natural filter, as \mathbf{n} is non-zero around Γ only. The force F_ε applied on ω is then deduced from an integration of the local normal constraints ponderated by δ_ε over Ω , as presented in Eq. (3.39).

$$\mathbf{T}_{local} = \sigma \cdot \mathbf{n} \quad (3.43a)$$

$$\sigma = \eta(\nabla \mathbf{u} + {}^t \nabla \mathbf{u}) - p \mathbb{I} \quad (3.43b)$$

$$\mathbf{n} = -\frac{1}{\|\nabla \Phi_\varepsilon\|} \nabla \Phi_\varepsilon \quad (3.43c)$$

This formulation is inherited from the smoothed Heaviside function definition. The ε parameter, characterizing the width of the transition area, is defining the span of the integration zone. The convergence of \mathbf{F}_ε towards \mathbf{F} when the width of the transition area tends towards zero has been established by Brackbill et al. (1992). Consequently, the precision of the force representation is directly dependent on ε , i.e., the quality of the force obtained is determined by the accuracy achieved within the immersion process.

When the immersion of a solid object ω is considered, the velocity of the flow inside the object is null. The only contributions of the flow are coming from the outside of ω . Consequently, the evaluation of the force applied on an object by a surrounding flow must focus on the exterior of ω only. The definition proposed in Eq. (3.41) is not suited for this usage, as δ_ε provides the same influence to the interior or the exterior. To account for the exterior flow only, a truncated Dirac function δ_ε^t is defined. The expression of δ_ε^t is written in Eq. (3.44), and a comparison of δ_ε and δ_ε^t is drawn in Fig. 3.8.

$$\delta_\varepsilon^t(\Phi) = \begin{cases} 2\delta_\varepsilon & \text{if } \Phi \leq 0 \\ 0 & \text{if } \Phi > 0 \end{cases} \quad (3.44)$$

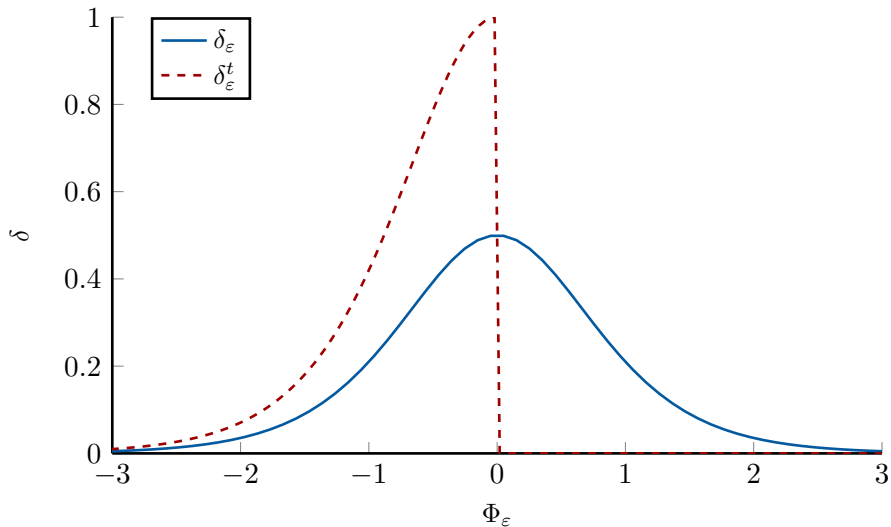


Figure 3.8: Dirac and truncated Dirac functions for $\varepsilon = 1$.

The truncated Dirac must respect the condition written in Eq. (3.42) to be used for the force computation. This definition is enabled by the even characteristic of δ_ε . For a

1D test case as plotted in Fig. 3.8, the development presented in Eq. (3.45) is trivial.

$$\int \delta_\varepsilon^t d\Phi = \int_{\Phi \leq 0} 2\delta_\varepsilon d\Phi \quad (3.45a)$$

$$= \int \delta_\varepsilon d\Phi \quad (3.45b)$$

In the context of a random, closed object in 2D or 3D, this statement becomes wrong. The regions where δ has high influence and corresponding, respectively, to the interior or the exterior of ω , do not have the same area. This statement is illustrated in Fig. 3.9 for the case of a circle (hatched). The support of δ defines the support area, between the dashed circles. The interior zone, hatched and colored in red, is smaller than the exterior one, colored in blue. Moving from δ_ε to δ_ε^t skips the red interior area, to evaluate the force in the blue exterior region only. The reasoning of Eq. (3.45) can be reproduced, this time using limits. The developments are presented in Eqs. (3.46).

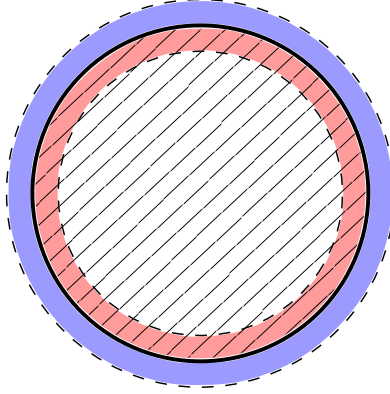


Figure 3.9: Dirac support around on circle (bold line) defining the interior (red, hatched) and the exterior (blue).

$$\int_{\Omega} \delta_\varepsilon dV = \int_{\Omega, \Phi \leq 0} \delta_\varepsilon dV + \int_{\Omega, \Phi \geq 0} \delta_\varepsilon dV \quad (3.46a)$$

$$\int_{\Omega, \Phi \leq 0} \delta_\varepsilon dV \xrightarrow{\varepsilon \rightarrow 0} \int_{\Omega, \Phi \geq 0} \delta_\varepsilon dV \quad (3.46b)$$

$$\int_{\Omega} \delta_\varepsilon dV \xrightarrow{\varepsilon \rightarrow 0} \int_{\Gamma} dS \quad (3.46c)$$

From the first equation of Eq. (3.45), which remains valid, and from Eqs. (3.46), the respect of the condition from Eq. (3.42) is established for δ_ε . This allows to compute the force applied by the fluid on an immersed geometry. A validation of the method presented in this Subsec. is proposed in Sec. 4.3.

3.3.2 Numerical wave tank

The numerical generation of waves is a critical point for the simulation of FOWTs. This is realized in ICI-tech in a Numerical Wave Tank (NWT) defined as a closed computational domain, rectangle in 2D and cuboid in 3D. Free-slip boundary conditions are used to retain the water located inside, except on top of the computational domain, where the flows are freed to respect the incompressibility. The initial pressure is fixed in this area to the atmospheric one. A 2D example of NWT is found in Fig. 3.10, for a water depth of 5m and 10m of air. At a point, parasitic velocities disturbed the computations,

forcing the implementation of a viscous area on top of the computational domain. In this region, where the viscosity of the flow was higher, the velocity of the flow is also prescribed horizontal. This correction prevented the creation of recirculation going through the borders of the domain.

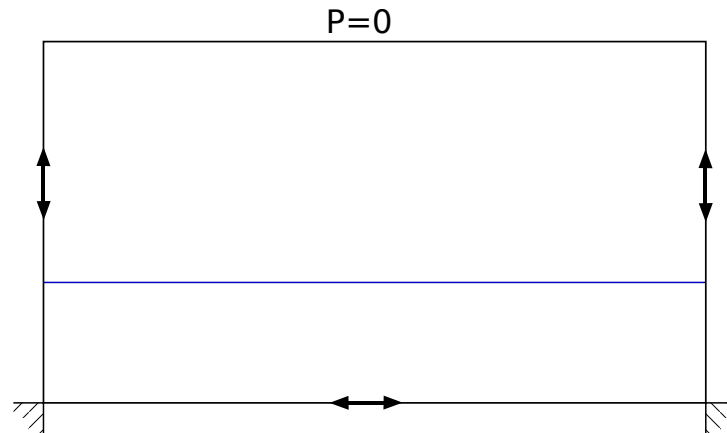


Figure 3.10: NWT with imposed pressure on top and with free-slip conditions on bottom, left and right.

As evoked in Subsec. [1.2.2](#), different methods can be used to generate waves. All of them focus on energy supply. Among them, two have been selected and implemented in ICI-tech: the dynamic boundary and the source-term approaches. A sponge layer method is also used to prevent the reflexion of waves at the end of the domain. The sponge layer approach consists in the creation of a numerical absorption area, where the energy of the incoming waves is dissipated. The flow entering the sponge layer progressively loses its energy, which results in smaller velocities. An illustration of the interest of a sponge layer is presented in Fig. [3.11](#). The blue curve, which corresponds to a case when no absorbing area was implemented, is highly perturbed at the right-hand side of the domain. These effects arise when a transport equation is solved near a wall, and generate numerical issues. The convergence in the resolution of the flows is harder to establish, and a lot of computational effort is required in terms of meshing. On the contrary, the red curve is progressively damped, which facilitates the computations. These utilities are presented in the following.

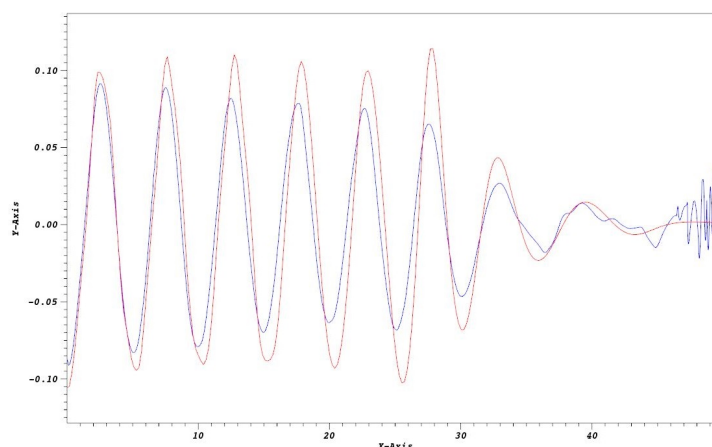


Figure 3.11: Comparison between two wave field. Red: With sponge layer on the right-hand side. Blue: Without sponge layer.

The tools presented in the following have been implemented in ICI-tech to aim at

developing a demonstrator of FOWT. Examples of NWTs can be found in Sec. 5.2 for preliminary works for the set-up of a FOWT demonstrator. The validation of the hydrodynamics, especially focusing on the propagation of wave fields, is presented in Sec. 4.2.

Wave generation

Several wave generation methods exist in the literature, which have been listed in Subsec. 1.2.2. Two of them have been implemented in ICI-tech: a physical wave maker, and a source-term wave generator. They are presented in the following.

The first wave generation method implemented consists in displacing a moving rigid solid, on which boundary conditions are applied. The solid is located at the left-hand side of the domain, to generate a wave field going through all the domain. The BCs defined on Fig. 3.10 are conserved, and a sponge layer is added. The NWT obtained is presented in Fig. 3.12. The sponge layer is represented using a gradation to highlight the progressive absorption of the energy in this area.

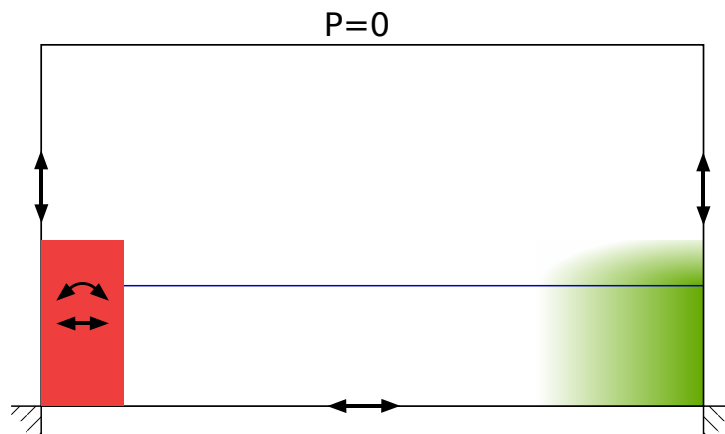


Figure 3.12: NWT integrating a dynamic boundary wave generator (left, red) and a sponge layer (right, gradation, green).

The first category of wave generators implemented in ICI-tech is the dynamic boundaries one, composed of two types of wave makers. The first type corresponds to a piston, moved horizontally only. This representation is simple and efficient, but lacks precision as the whole column of water in contact with the piston is excited. A more advanced model considers a hinged wave maker, which mostly energizes the free-surface (FS). This model provides most of the energy to the water located near the FS, which is closer from reality. The definition of the boundary conditions applied on the wave generator is obtained using the software HOS-NWT (for High Order Spectral - Numerical Wave Tank), developed by Ducroz et al. (2012). Using as input a prescribed wave field, HOS-NWT is able to provide the position and velocity of a wave maker, either of piston or hinged type. In the NWT as considered by HOS-NWT, presented in Figs. 3.13, different wave fields can be represented, from monochromatic to irregular ones representing real sea states. The imposition of the BCs follows a simple procedure. At a given time step, the position of the wave generator is updated. The velocity is imposed from the boundary conditions of the moving solid, and the waves are generated and propagated through the resolution of the Navier-Stokes equations and of an advection problem.

This procedure offers important flexibility. The construction of a moving solid in the domain is rather straightforward, and its motion is directly imported from HOS-NWT. The main limitation comes from the characteristics of physical wave generators. They tend to create water perturbations near the wake maker, which rapidly decay. However,

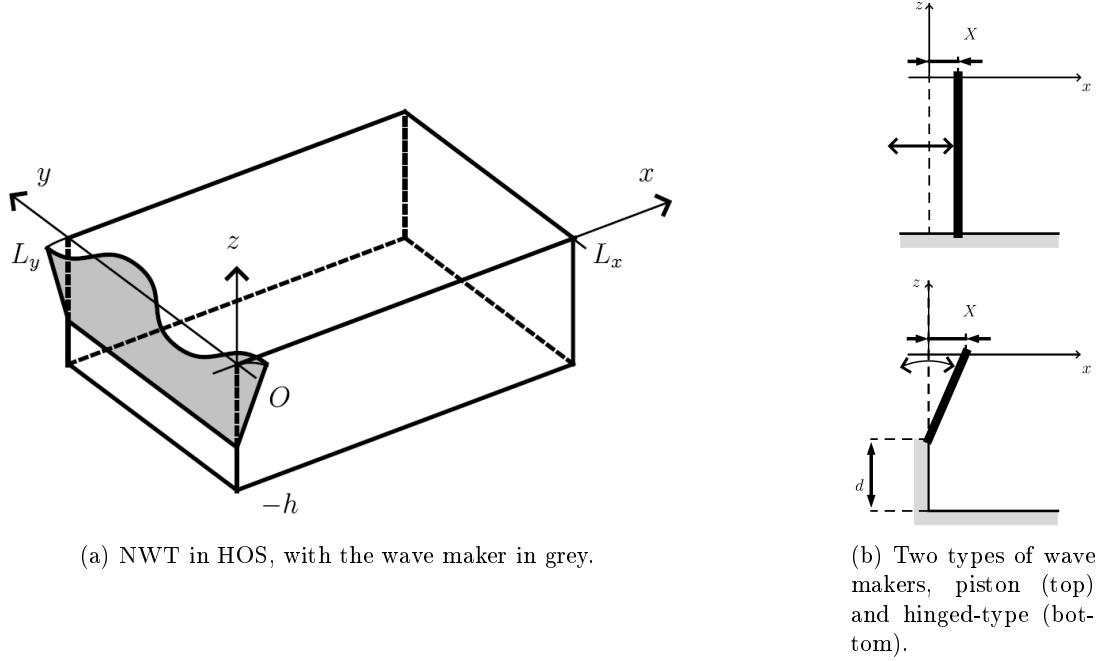


Figure 3.13: Description of the NWT defined in HOS-NWT, from [Ducroz et al. \(2012\)](#).

these effects are time-consuming, and may lead to inaccuracy in the simulations. A way to tackle those turbulent issues is to increase the viscosity of the fluids in the computation. This improves the convergence of the schemes and thus reduces the computational expenses, but modifies the flow. Even if the differences obtained are minor, the perturbations induced in the flows around structures are largely modified. The FSI, which is a key element of an accurate study of FOWT, is deeply altered. Consequently, this technique can be used to model accurate wave fields, but struggles once FSI studies have to be realized.

A source-term wave generation process, used by [Calderer \(2015\)](#), has been implemented too. Contrarily to the dynamic BCs method, this approach does not need any physical wave makers. The waves are this time generated by source-terms, which act directly in the momentum equation of the NS problem. The modified momentum equation is presented in Eq. [\(3.47\)](#).

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) - \eta \Delta \mathbf{u} + \nabla p = \mathbf{f} + \rho \mathbf{S} \quad (3.47)$$

The source term \mathbf{S} which appears in this Eq. can be viewed as a modulator of gravity along the FS. In the context of a monochromatic wave field of amplitude A , of wavelength k , of frequency ω and of phase shift θ , the source terms applied directly on the FS \mathbf{S}_{FS} can be expressed at any time t and position x . The relation can be found in Eq. [\(3.48\)](#). The terms \mathbf{n}_{FS} and C_{FS} correspond, respectively, to the outward normal to the FS and a constant depending on the wave parameters.

$$\mathbf{S}_{FS} = C_{FS} \sin(\omega t - \theta) \mathbf{n}_{FS} \quad (3.48)$$

However, the meshing of the computational domain prevents from imposing the source function at a unique point. The excitations of the source have to be shared among several points located around the FS, but also in an interest zone. In a similar approach to the one developed in Eq. [\(3.42\)](#) for the evaluation of forces, the source term can be applied to the FS using a Dirac function δ_ε . This Dirac function can be built from the level-set representing the FS Φ_ε , e.g., using Eq. [\(3.41\)](#). Similarly to the procedure followed in

Subsec. [3.3.1](#), this smoothing function allows moving from integration over the whole volume to a limit towards an integration along the FS. This can be summarized by Eq. [3.42](#), with Ω the computational domain and Γ the FS.

The source-term theory considers an energy supply concentrated around the FS, from which waves can be propagated. To that extent, the source term has to be applied only to a certain region of the FS. In the context of a 2D wave maker, the theoretical source needs to be nodal. A new discretization has to be processed, this time along x . Had a 3D case being considered, the source term could have been nodal, but it could also be applied along a line. For a nodal source, two discretizations would have been needed, along the axis x and y as defined in Fig. [3.13a](#). For a line-source, the 2D expression can be kept, with a normalization linked to the length of the NWT along the transversal direction. The development proposed hereafter consider a 2D NWT only.

A Dirac function characterizing the width of the source area along the x direction needs to be defined. For the application of a source term S_{FS} around a point located on the FS, at $x = x_S$, the Eq. [\(3.49a\)](#) needs to be respected, which implies the limit presented in Eq. [\(3.49b\)](#).

$$\int_{\Gamma} \delta_x \mathbf{S}_{FS} dS \xrightarrow{x \rightarrow 0} \mathbf{S}_{FS}(x_S) \quad (3.49a)$$

$$\int_{\Omega} \delta_{\varepsilon} \delta_x \mathbf{S}_{FS} dV \xrightarrow{\varepsilon, x \rightarrow 0} \mathbf{S}_{FS}(x_S) \quad (3.49b)$$

Several different formulations are available for δ_x . The one proposed by [Calderer \(2015\)](#) is used, and is presented in Eq. [\(3.50\)](#). The parameter ε_x represents half the width of the source area, while x' is the translation of x being null in the middle of the source area.

$$\delta_x = \begin{cases} \frac{1}{2\varepsilon_x} \left(1 + \cos \left(\frac{\pi x'}{\varepsilon_x} \right) \right) & \text{if } -\varepsilon_x < x' < \varepsilon_x \\ 0 & \text{otherwise.} \end{cases} \quad (3.50)$$

These two Dirac functions allow to define a region, in which the source function \mathbf{S} can be applied. The expression of \mathbf{S} is presented in Eq. [\(3.51a\)](#). The vector \mathbf{n} is built at each point as in Eq. [\(3.43c\)](#). The terms g , ρ_w and ρ_{FS} of Eq. [\(3.51b\)](#) correspond respectively to the gravity constant, the density of water, and $\rho(\Phi_{\varepsilon} = 0)$. It forces the width of the source region along x to be smaller than k_x .

$$\mathbf{S} = C \sin(\omega t - \theta) \delta_{\varepsilon} \delta_x \mathbf{n} \quad (3.51a)$$

$$C = \frac{A \rho_w g^2 \varepsilon_x k (\pi^2 - \varepsilon_x^2 k^2)}{\rho_{FS} \omega^2 \pi^2 \sin(k \varepsilon_x)} \quad (3.51b)$$

The generation of waves is less expensive computationally compared to the dynamic BCs method. There is no need to represent a solid in the domain, which leads to a reduction in both the number of nodes in the mesh and the cost of remeshing. A higher cost is required in the implementation, as the momentum equation has to be modified. A correction of the NS implementation is required for this method. In ICI-tech, the P1-FE VMS NS problem presented in Eq. [\(3.24\)](#) has to be modified, to integrate the corrections of the momentum equation proposed in Eq. [\(3.47\)](#). In practice, this step is not that challenging, as \mathbf{S} can be treated in a similar way as \mathbf{f} ; instead of considering \mathbf{f} in Eq. [\(3.24\)](#), Eq. [\(3.47\)](#) requires the use of $\mathbf{f} + \rho \mathbf{S}$. Consequently, after implementation, this method is efficient to generate a monochromatic wave field. A first limitation of this method arises when a complex wave field is required. Theoretically, this method offers to generate complex wave fields, using a decomposition in monochromatic waves. Consider a complex

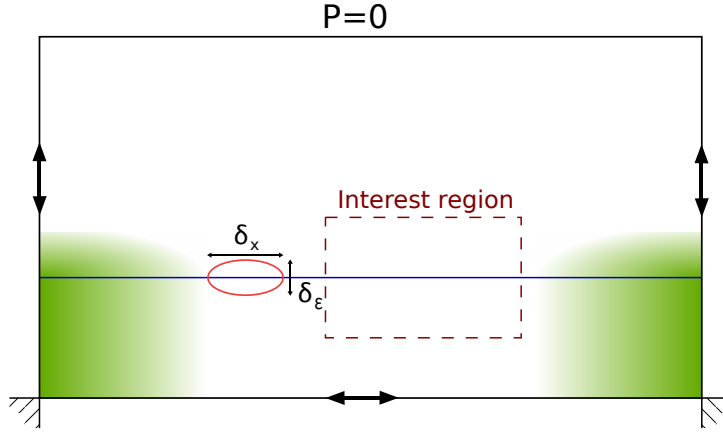


Figure 3.14: NWT integrating a source-term wave generator (circled, red) and sponge layers (gradation, green).

wave field composed of N monochromatic waves of parameters $(A_i, k_i, \omega_i, \theta_i)_{i \in \mathbb{N}, i < N}$. The source term related to the complex wave field \mathbf{S}_c writes as in Eq. (3.52). The constant C_i is similar to the one defined in Eq. (3.51b).

$$\mathbf{S}_c = \sum_{i < N} (C_i \sin(\omega_i t_i - \theta_i)) \delta_\epsilon \delta_x \mathbf{n} \quad (3.52)$$

This definition seems to offer great potential for the simulation of complex waves. However, Choi and Yoon (2009) and Ha et al. (2013) obtained dissipation at certain frequencies. More generally, they struggled to conserve the spectrum used in the generation after propagation of the wave field. Consequently, this methodology can be used to simulate simple test cases, when a monochromatic wave field is considered. Preliminary results can be obtained at a reduced cost, and basic FSI can be realized. However, the source-term wave generation method may be disqualified when complex wave fields are simulated.

The different options considered for NWTs in ICI-tech have been illustrated using 2D test cases, for the sake of simplicity. Note that 3D implementation has also been realized for all the different NWTs. In the context of physical wave makers, the definition of a 3D wave tank is straightforward. A single wave maker can generate a monochromatic wave field independent from the transversal position in the NWT. To generate more complex flows, HOS-NWT enables to discretize the physical wave maker, and provides the motions of each wave maker. For source-term wave makers, the generation of wave fields can be realized from either a source point or a source line positioned on the FS.

Absorption of waves

The generation of waves in a small, closed domain also imposes to damp the waves arriving at the right end of the NWT. If it is not done, a reflexion of the incoming wave train would perturb the flows in all the computational domain. Sponge layers have to be designed, as already mentioned in this Subsec. Several approaches have been tested, which intended to dissipate the energy of the waves. The most promising one is the relaxation approach, which relies on the blending of a target solution β_t , determined analytically, into the computed one β_c . In the general theory, the function β can be the velocity of the flow or the FS elevation. The relaxation function γ characterizes the relative positioning of the current point into the sponge layer. It varies from 0 at the beginning of the relaxation area to 1 at its end, and is 0 elsewhere. The relation solved

is written in Eq. (3.53).

$$\beta = \gamma\beta_t + (1 - \gamma)\beta_c \quad (3.53)$$

For the design of an absorbing area, the relaxation of the velocities of the flow is realized towards a target solution $\mathbf{u}_t = 0$. The determination of γ leads to many different possibilities. A theoretical requirement noted by Engsig-Karup et al. (2007) is that the derivative of γ should vanish at the borders of the relaxation area. In this thesis, the function defined by Bonnefoy et al. (2006) and written in Eq. (3.54) is used, as it answers the condition presented for $x_s \in [0; 1]$ in the sponge layer, with $x_s = 0$ where $\beta = \beta_c$ is required.

$$\gamma = x_s^2(3 - 2x_s) \quad (3.54)$$

3.3.3 Buoyancy and Fluid-Structure Interaction

The floating ability is a necessity for a FOWT, as its name suggests. Fluid-Structure Interaction (FSI) tools are required, to account for the effect of the environment on the FOWT, but also the response of the floating structure to perturb its near flow. A rigid-body motion has been implemented in ICI-tech, within a master thesis integrated into the scope of the PhD. The movement of a FOWT is solved in a 6 Degrees of Freedom (DoF) way. The resolution of the movements is integrated into the monolithic framework developed for the resolution of the Navier-Stokes equations.

Rigid-body motion

The movements of a rigid solid ω can be split between translation and rotation. The relation ruling it is presented in Eq. (3.55), for u and x respectively velocity of the flow and position at a point of the solid. U and R correspond to the translation and rotation vectors characterizing the rigid-body motion, and \wedge is the cross-product operator.

$$u(x, t) = U + x \wedge R \quad (3.55)$$

The evaluation of U and R can be realized thanks to a minimization problem, presented in Eqs. (3.56). Theoretically, the couple (U, R) determined using three points or using all the points of the computational domain located inside ω should be identical, as the motion of three points defines the dynamics of a rigid body. The consideration of more than three points is a way to eliminate numerical bias, that could arise in particular situations, with e.g., aligned points.

$$\min_{U, \omega} \left(\int_{\omega} \|\varphi(u, x)\|^2 \right) \quad (3.56a)$$

$$\varphi(u, x) = U + x \wedge R - u \quad (3.56b)$$

The minimal values of U and R can be found when the conditions presented in Eq. (3.57) are fulfilled.

$$\begin{cases} \frac{\partial \varphi}{\partial U} = 0 \\ \frac{\partial \varphi}{\partial R} = 0 \end{cases} \quad (3.57)$$

Once the couple (U, R) has been determined, the resolution of the new position of the rigid body can be determined. Each point defining the object could be displaced individually. However, as the representation of the object may be represented by a mesh gathering a huge number of points, this step can be time-consuming. The properties of a

rigid-body motion are exploited, and the referential related to the object is updated from the initial coordinate system. The new position is detailed in Eq. (3.58), for a referential of origin O and of axis e^i .

$$\begin{cases} O_{t+\Delta t} = O_0 + \Delta t (U + O_0 \wedge R) \\ e_{t+\Delta t}^i = \frac{1}{\|e_0^i + \Delta t e_0^i \wedge R\|} (e_0^i + \Delta t e_0^i \wedge R) \end{cases} \quad (3.58)$$

This procedure takes advantage of the rigid characteristics of the immersed object. This inherits from the position of a rigid geometry, which can be split between a translation and a rotation from the original coordinate system. The reconstruction can thus be performed in a reference frame, before the object is translated and rotated into its real position. This procedure allows using an octree, which does not need to be rebuilt at each time increment. More generally, the savings highlighted in Sec. 2.4 are of particular interest here, as the object studied needs to be reconstructed at each time increment.

A last remark considers the data required for the resolution of the rigid-body motion. The velocity and pressure need to be computed inside the object to immerse, to be able to determine accurately U and R . Hence, the presence of the solid must be considered in the resolution of the flow, to get velocity and pressure distributions representative of a rigid solid.

Determination of the flow inside a rigid-body

The focus is placed now on the evaluation of the velocity and pressure inside the immersed solid ω , of level-set Φ_ϵ . An additional rigidity constraint is added to the classical incompressible Navier-Stokes problem presented in Eq. (3.4). This constraint is written in Eq. (3.59), with ϵ_s the deformation-rate tensor of the solid. The momentum equation of the Navier-Stokes problem is also modified to integrate a rigidity constraint τ_s , for the computational points located inside ω .

$$\epsilon_s(u) = 0 \quad (3.59)$$

The new Navier-Stokes equations dedicated to FSI with a rigid body ω , which includes this new condition, is presented in Eq. (3.60). This formulation uses the Heaviside description of ω , H_ϵ . The addition of this function enables to create a generic expression, valid everywhere in the computational domain.

$$\begin{cases} \rho(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}) - \eta \Delta \mathbf{u} - H_\epsilon \nabla \cdot \tau + \nabla p = f & \text{in } \Omega \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega \\ H_\epsilon \epsilon_s(\mathbf{u}) = 0 & \text{in } \Omega \end{cases} \quad (3.60)$$

This modified Navier-Stokes problem is then expressed in a VMS paradigm, following the procedure introduced in Subsec. 3.2.2. However, the addition of the rigidity constraint complicates the resolution of the flows. This difficulty is overcome through the addition of an augmented Lagrangian coupled to an iterative Uzawa scheme. The computation of \mathbf{u}_{t+1} and p_{t+1} velocity and pressure at the time increment $t + 1$ is done iteratively, from the values of \mathbf{u}_t , p_t and τ_t . The Uzawa iterations consider successive resolutions of Eq. (3.61), with $\mathbf{u}^0 = \mathbf{u}_t$, $p^0 = p_t$ and $\tau^0 = \tau_t$.

$$\begin{aligned} & \forall (\mathbf{v}, q) \in V_0 \times Q, \\ & \begin{cases} \rho(\partial_t \mathbf{u}^{k+1}, \mathbf{v}) + \rho(\mathbf{u}^{k+1} \cdot \nabla \mathbf{u}^{k+1}, \mathbf{v}) + (2\eta \epsilon(\mathbf{u}^{k+1}) + H_\epsilon(\alpha)(\tau^k + 2\eta r P_\tau(\epsilon(\mathbf{u}^{k+1}))), \epsilon(\mathbf{v})) \\ - (p^{k+1}, \nabla \cdot \mathbf{v}) = (\mathbf{f}, \mathbf{v}) \\ (\nabla \cdot \mathbf{u}^{k+1}, q) = 0 \end{cases} \end{aligned} \quad (3.61)$$

After each resolution of Eq. (3.61), \mathbf{u}^{k+1} and p^{k+1} have been determined. Following the paradigm of the augmented Lagrangian theory, the value of τ^k is updated, as presented in Eq. (3.62). The parameter r is a penalty coefficient, and P_τ is a projection operator.

$$\tau^{k+1} = \tau^k + 2\eta r P_\tau(\epsilon(\mathbf{u}^{k+1})) \quad (3.62)$$

The execution moves outside the Uzawa loop once the convergence condition $\|\mathbf{u}^{k+1} - \mathbf{u}^k\| < \varepsilon_{tol}$ is met, with ε_{tol} specified tolerance. The converged values of velocity and pressure define to the solved flows at the next time iteration, i.e., $\mathbf{u}_{t+1} = \mathbf{u}^{k+1}$ and $p_{t+1} = p^{k+1}$. The determination of velocity and pressure at the next time step allows computing the motions of the immersed object, following the rigid-motion theory previously exposed.

The utilities presented above allow performing FSI with rigid-bodies. This is applied in this context to a FOWT floating freely under the action of buoyancy. In this particular situation, a FOWT under the action of a wave field is subject to drift. The correct behavior of a FOWT, in particular the stationkeeping, would be obtained through the addition of mooring models. The representation of moorings lines have not been considered in this thesis, and should be regarded in further studied extending the simulator of FOWTs. Different approaches dedicate to their modeling have been briefly discussed in Subsec. 1.2.2.

3.4 Conclusion of the numerical framework

This Chap. presented the numerical framework used in ICI-tech for the resolution of the flow problem. A particular focus has been placed on the incompressible Navier-Stokes equations. Generalities about the resolution of a flow problem using the Navier-Stokes equations have been provided in Sec. 3.1. The monolithic resolution of the Navier-Stokes problem, formulated in a VMS paradigm for a FE framework, is detailed in Sec. 3.2. The developments and implementations realized in the context of FOWTs within the scope of this thesis have been presented in Sec. 3.3.

Only minor modifications were brought into the monolithic framework for the flow resolution during this thesis. The VMS formulation of the Navier-Stokes problem, based on P1-FE, has not been modified. The different approximations detailed in Subsec. 3.2.2, concerning, e.g., quasi-static sub-scales, have been questioned in the context of FOWTs. However, no reimplementation was realized within the scope of this thesis, even if the solver as currently developed seemed more suited for low- Re flows.

The most notable contributions to the development of the software concerned the implementation of bricks intending to enable the definition of a FOWT simulator. A method for the evaluation of the loads applied on the surface of an immersed object has been implemented. This development gives the potential to measure the efforts applied on a FOWT, which is both critical for design purposes. A FSI framework has been implemented within a master thesis for rigid bodies of complex geometries, which will handle the buoyancy, a necessary tool for the simulation of FOWTs. These additions to the software platform are wider than the particular application of FOWTs.

The final contribution of this thesis to the numerical extension of ICI-tech came with the development of a NWT. Several wave generation techniques have been considered, with physical and source-term wave makers. A sponge layer method was implemented for the absorption of the energy at the borders of the NWT, to reduce the computational expenses and facilitate the convergence of the schemes.

Chapter 4

Verification and validation towards the simulation of operating FOWTs

Contents

4.1 Monophasic verification and validation – Towards high-Reynolds flows	88
4.1.1 Low-Reynolds steady flow around a confined cylinder	88
4.1.2 Moderate-Reynolds flows around NACA profiles	93
4.1.3 High-Reynolds, steady flows around a section of WT blade	99
4.2 Hydrodynamic verification – Generation and propagation of monochromatic wave fields	102
4.2.1 Verification of the solver using constant, isotropic meshes	103
4.2.2 Propagation of monochromatic wave fields using automatic, anisotropic mesh adaptation	108
4.2.3 No-flow tests	116
4.3 Verification and validation of the FSI and application to buoyancy	118
4.3.1 Verification of the buoyancy force applied on a cylinder	118
4.3.2 Validation of the FSI solver for driven motions of floating structures	120
4.4 Conclusion on the numerical results	126

4.1 Monophasic verification and validation – Towards high-Reynolds flows

The adaptation of ICI-tech to the context of floating wind energy requires the simulation of high-Reynolds flows, with Re reaching several million. On the contrary, ICI-tech was commonly used for the simulation of viscous flows, e.g., by [Ville et al. \(2011\)](#) of [Betancourt \(2017\)](#). A validation process is required, to guarantee the quality of the simulations. This Sec. presents the different steps followed for this validation from low- Re to high- Re flows in Subsecs. [4.1.1](#) to [4.1.3](#).

4.1.1 Low-Reynolds steady flow around a confined cylinder

In the study of [Schäfer et al. \(1996\)](#), test cases of fixed objects under the effect of confinement were submitted to several research teams to produce a numerical benchmark. The first test case considers a cylinder in a water tunnel. The objective behind this study is to evaluate the drag and lift forces generated by a low-Reynolds flow ($Re = 20$) on a simple geometry. The action of the fluid on the bodies creates a drag force. Confinement is established, as the walls are located near the geometries. A slight deflection of the object, which is not positioned in the middle of the wind flume, also creates a lift force. Steady and unsteady test cases are detailed in the study.

This Subsec. focuses on steady test cases only. This choice is motivated by the verification required for the evaluation of forces detailed in Subsec. [3.3.1](#), which is easily performed on steady test cases. These applications also verifies the behavior of the solver for low-Reynolds flows. The study from [Schäfer et al. \(1996\)](#) intended to measure the drag and lift coefficients applied by the flow on the cylinder, along with the length of the recirculation area and the pressure drop observed. A focus is here placed on drag and lift coefficients C_d and C_l . They are written in Eqs. [\(4.1\)](#), with ρ the fluid density, u the freestream velocity, A the cross-sectional area and F the force computed with the method presented in Subsec. [3.3.1](#). The vector e_x is aligned with the inflow, and e_y is the vector normal to it, see Fig. [4.5](#).

$$C_d = \frac{2}{\rho u^2 A} F \cdot e_x \quad (4.1a)$$

$$C_l = \frac{2}{\rho u^2 A} F \cdot e_y \quad (4.1b)$$

The steady test cases selected consist in a cylinder disposed inside a water tunnel. The flow is propagated rightwards at a Reynolds number of 20, with a Poiseuille inflow. In this situation, the cross-sectional area A corresponds to the product of the diameter of the cylinder by the width of the tunnel. To reproduce the test case, no-slip boundary conditions are imposed at the borders of the domain and inside the cylinder. The inlet imposes a Poiseuille flow, while the outlet is opened. The drag and lift coefficients are measured around the cylinder. In their study, [Schäfer et al. \(1996\)](#) compared the results from several CFD codes, which used a LES formulation of the Navier-Stokes equations with different discretizations in time and space. They defined bounds for C_d and C_l for the different test cases. A comparison is drawn between those bounds and the results obtained using ICI-tech.

The simulations realized with ICI-tech start from an initial mesh refined around the cylinder. The mesh is regularly adapted during the transitory regime, until a converged, steady flow is obtained. The objective being to evaluate a steady flow, the transitional results are not considered. To that extent, the choice of the time step is not critical, as

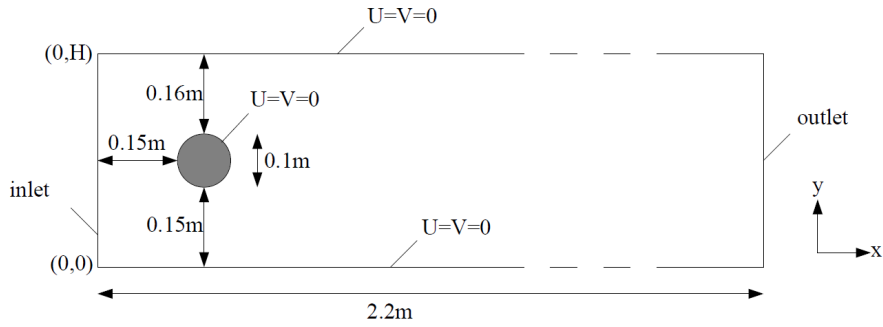


Figure 4.1: 2D steady test case of cylinder in confinement, from Schäfer et al. (1996).

soon as the aerodynamic effects, limited at $Re = 20$, are well captured. The number of nodes in the computation is fixed, to limit the influence of this parameter on the simulations. This study then allows evaluating the action of the width of the level-set ε , presented in Sec. 2.1, on the precision of the steady values of C_d and C_l . Mesh convergence studies are realized. The width of the adapted area varies with ε , thus the automatically-generated meshes provide a varying precision around the cylinder. Corresponding time steps are used, depending on the minimal mesh size to respect the CFL condition.

2D confined cylinder

The first test case consists in a 2D cylinder confined in a water tunnel. The specifications of the test case are presented in Fig. 4.1. In this situation, the cross-sectional area A is restricted to the diameter of the cylinder, hence $A = 0.1m$. The expression of the Poiseuille inflow is presented in Eq. (4.2), with $u_M = 0.3m.s^{-1}$ the maximal velocity of the flow.

$$u_{inlet} = 4u_M y(H - y)/H^2 \cdot e_x \quad (4.2)$$

The configuration of the test case simulated with ICI-tech is presented in Figs. 4.2. The velocity profile of the converged steady flow is drawn, with the corresponding computational mesh. The mesh adaptation procedure tended to refine the cells in the areas where important variations are observed. In particular, the surroundings of the cylinder and the boundary layers developing at $y = 0m$ and $y = 0.41m$ need many mesh points for their representation. The aerodynamic coefficients are evaluated around in the cylinder, in an area defined from the level-set function. The evaluation of the forces acting on the cylinder presented in Subsec. 3.3.1 requires an important number of mesh cells around the free-surface. The zoom presented for a configuration of $\varepsilon = 10^{-4}m$ and for $30K$ points in the computational mesh shows that the transition area is properly represented. An example of simulation conducted with $\varepsilon = 10^{-2}m$ is presented in Figs. 4.3, where transitional and steady regimes can be highlighted.

A mesh convergence study is proposed, which focuses on the aerodynamic coefficients evaluated as a function of ε . For values of ε scaling from $10^{-2}m$ to $10^{-4}m$, the converged C_d and C_l are compared to evaluate their dependency towards the level-set representation. The zoom presented in Fig. 4.2b corresponds to the smallest ε simulated in the upcoming mesh convergence study. The number of mesh cells around the interface of the cylinder guarantees that, independently of the ε used for the test cases simulated, the meshing of the object will be correctly realized. The values obtained using ICI-tech are compared to bounds provided by Schäfer et al. (1996) in Figs. 4.4. The evaluation of the drag coefficient provides an acceptable accuracy independently of the ε chosen. Compared to the bounds of reference, the maximal error generated is inferior to 1%, and diminishes with ε . The evaluation of the lift coefficient is much more challenging. The

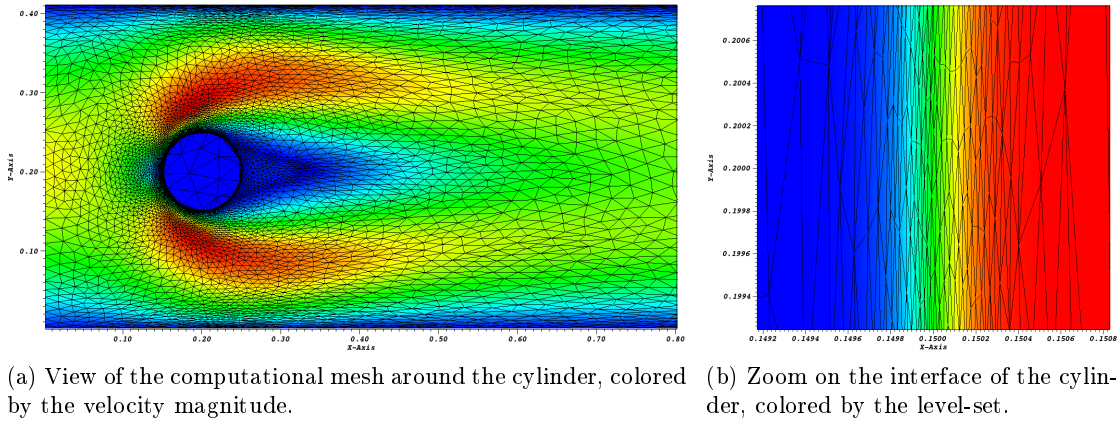


Figure 4.2: View of the computational mesh for the test case of a 2D confined cylinder at $Re = 20$ and with $\varepsilon = 10^{-4}m$, for a 30K-points mesh.

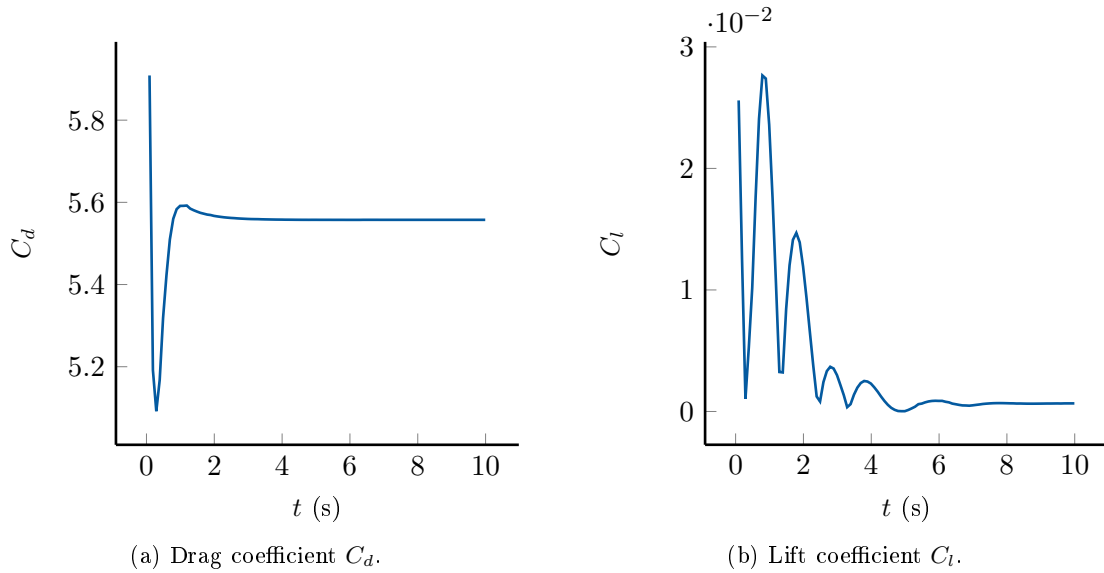


Figure 4.3: Evolution of the drag and lift coefficients as a function of the time, for a simulation realized with $\varepsilon = 10^{-2}m$ and $\Delta t = 10^{-2}s$.

order of magnitude of C_l is of 10^{-2} . The force to evaluate is of about $10^{-5}N$, which gives more influence to numerical errors arising. Still, the evaluation of C_l realized with ICI-tech is on the right order of magnitude, even if no convergence is established. A case could be made for simulation with lower ε , to check if a convergence is established. These test cases are not presented here, as a transition to 3D with those parameters is unrealistic. The level of accuracy achieved is considered interesting, especially for the drag coefficient. The dependency of the force evaluation method seems to be dependent on the mesh used, particularly when small loads need to be measured.

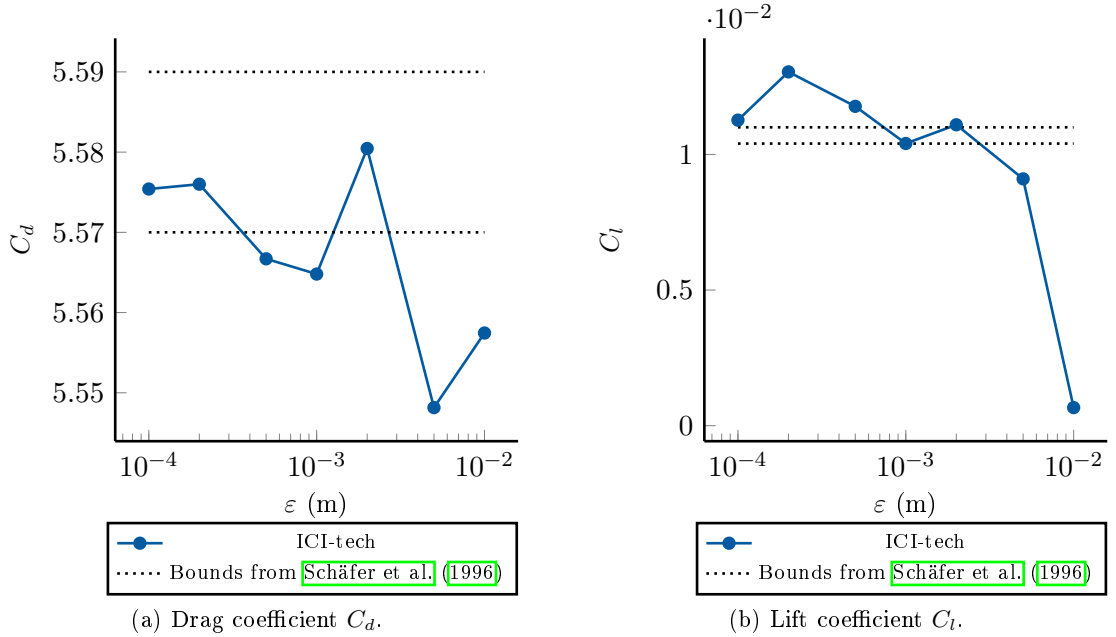


Figure 4.4: Steady values of aerodynamic coefficients measured around a 2D confined cylinder, with meshes of $30K$ nodes and various ε .

3D confined cylinder

The 3D test case is an extension of the 2D simulation presented previously, as detailed in Fig. 4.5. The geometry of the test case is similar, except for the cylinder positioned $30cm$ further from the inlet. The Poiseuille flow is modified to keep the Reynolds number of the flow constant. Its expression is written in Eq. (4.3), with $u_M = 0.45m.s^{-1}$ the maximal velocity of the flow and H the height and width of the tunnel.

$$u_{inlet} = 16u_M yz(H - y)(H - z)/H^4 \cdot e_x \quad (4.3)$$

An illustration of the simulation processed is presented in Figs. 4.6, with an example of mesh used during the computations as depicted in Fig. 4.6a. The mesh presented is refined in the same areas than the 2D one from Fig. 4.2a. The differences in the velocity profile are related to the position of the cylinder, further from the inlet in this second situation.

Mesh convergence studies are realized with meshes containing respectively $60K$ and $300K$ nodes, and with ε in the same range as in the 2D test case. Only the test cases with $\varepsilon = 10^{-4}m$ became unaffordable for the 3D test cases considered. The results obtained are presented in Figs. 4.7. The evolution of C_d and C_l as a function of ε , for different mesh sizes, are examined. They are compared to the bounds provided by Schäfer et al. (1996), which are plotted on each Fig. The different C_d measured are in the bounds provided with large values of ε , which suggests that relatively accurate results can be

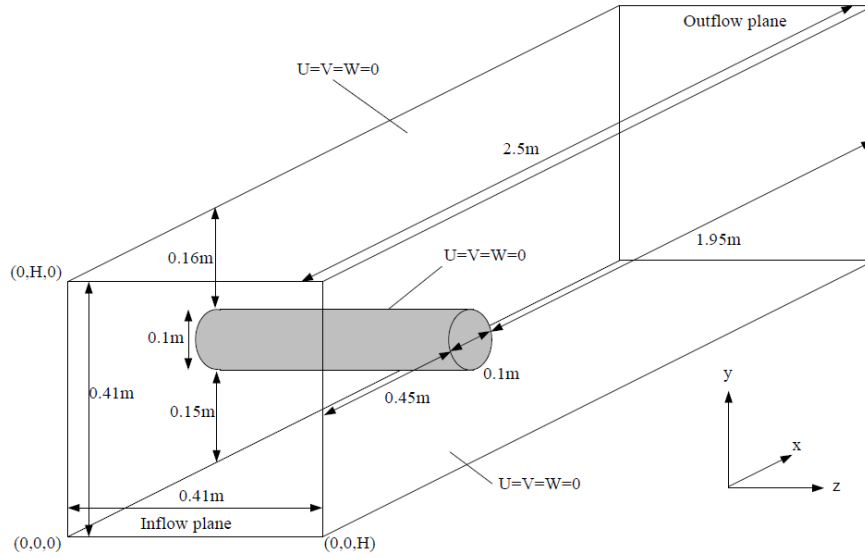
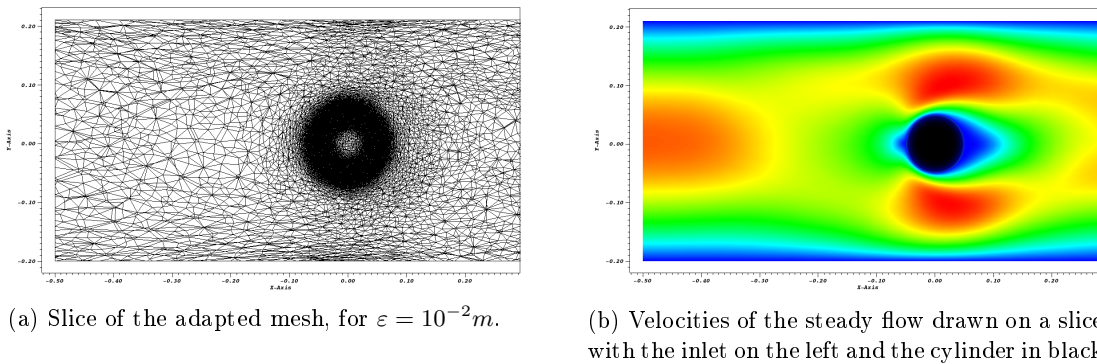


Figure 4.5: 3D steady test case of cylinder in confinement, from Schäfer et al. (1996).



(a) Slice of the adapted mesh, for $\varepsilon = 10^{-2}m$.

(b) Velocities of the steady flow drawn on a slice, with the inlet on the left and the cylinder in black.

Figure 4.6: Steady flow around a 3D cylinder at $Re = 20$ simulated with a $300K$ -points mesh, focusing on the inlet and cylinder region.

obtained with $\varepsilon/2r = 0.1$, r being the radius of the cylinder. While ε reaches smaller values, the results become approximative, and out of the bounds. This effect is observed more rapidly with 60K points, which suggests that it corresponds to a lack of points in the computational mesh. For $\varepsilon = 2 \cdot 10^{-4}m$, both meshes of 60K and 300K points provide C_d evaluation outside the bounds. This observation is due to the number of points in the mesh, which is not sufficient to reach mesh sizes of $2 \cdot 10^{-5}m$ around the whole surface of the cylinder. Concerning C_l , the values measured are much more erratic. With both test cases, the coefficient varies a lot outside the bounds prescribed. The small value of C_l , of about 10^{-2} , is partly responsible for this, as the total force measured around the cylinder is on the order of $10^{-5}N$. The evaluation of such a small force from the integration of the constraint around the cylinder gives a lot of importance to the disposition of the mesh. The remeshing thus has a huge influence on the measurements. This dependency will have to be examined with more attention in further studies. Increasing the number of points in the simulations may be a way to improve the evaluation of C_l . As the focus of this thesis goes to higher- Re flows, representative of operational WT, the lift coefficients measured will be much more important. The results of C_d were thus considered sufficient to transition towards flows at higher Re .

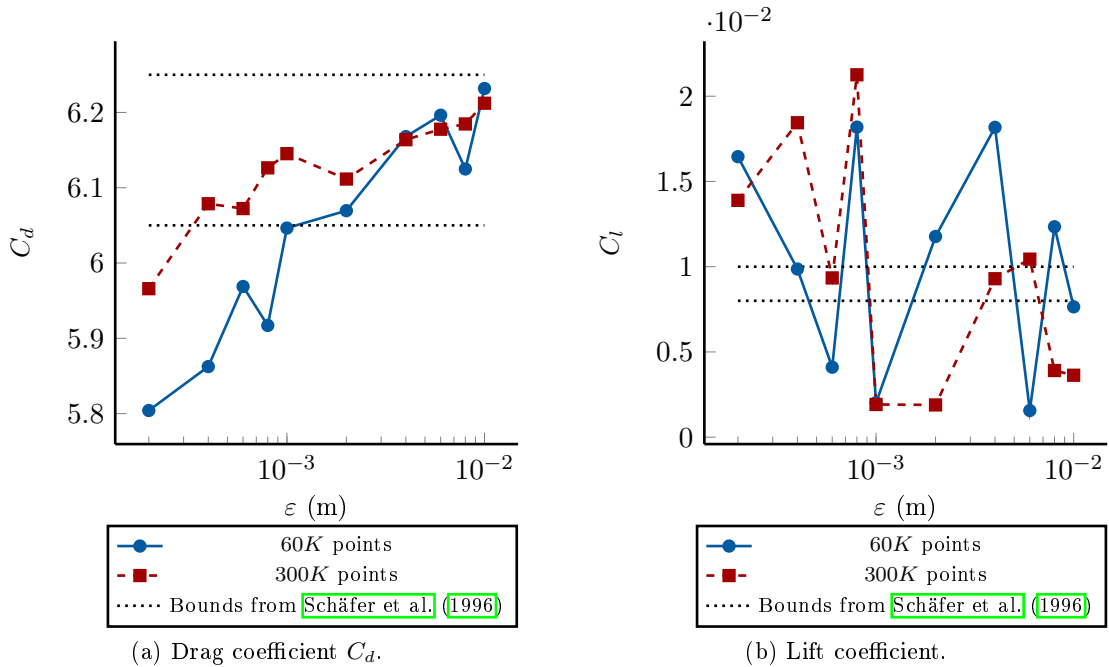
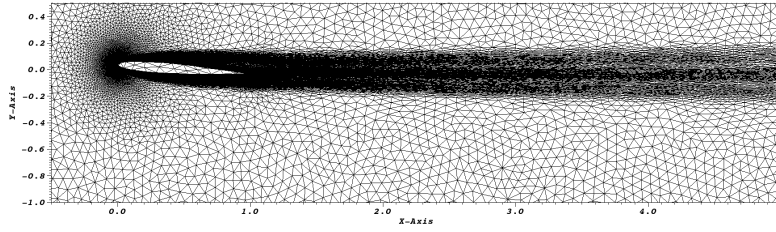


Figure 4.7: Aerodynamic coefficients in a mesh convergence study around a 3D cylinder, with meshes of 60K and 300K nodes.

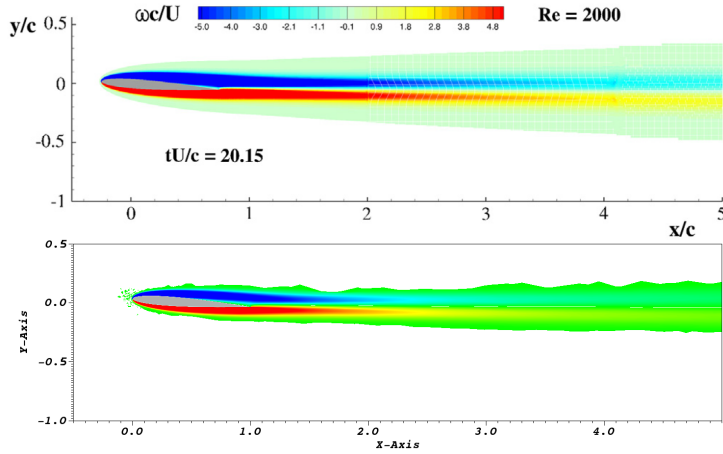
4.1.2 Moderate-Reynolds flows around NACA profiles

Steady test case at $Re = 2000$

A verification with higher-Reynolds flows is now realized, based on the test cases proposed by Rossi et al. (2016). The first test case studied concerns simulations of a fixed 2D NACA 0008 airfoil in a constant, uniform flow of $Re = 2000$. An orientation of 4° is prescribed for the NACA profile, which enables the installation of a steady, laminar flow around it. The flow is directed rightwards. The results provided by ICI-tech are compared with the one from Rossi et al. (2016), which are obtained thanks to a 2D Diffused Vortex Hydrodynamics (DVH) method. This pure meshless approach proved to be accurate for those applications. In particular, the numerical diffusion observed in the DVH simulations is particularly reduced. The DVH approach was validated against



(a) Adapted mesh generated.



(b) Vorticity field obtained with a DVH method by Rossi et al. (2016) (up), and with ICI-tech (down).

Figure 4.8: Steady flow around a NACA 0008 profile with an incidence of 4° at $Re = 2000$, with $\varepsilon = 2 \cdot 10^{-2}m$ and $60K$ nodes in the computational mesh.

a FVM code for the particular applications presented in this Subsec.

The mesh used for the simulation with ICI-tech and the results obtained are presented in Figs. 4.8. From an initial mesh refined around the airfoil, the successive adaptation steps and the formation of aerodynamic effects create concentrations of mesh points in the wake of the profile. The mesh visualized features a high concentration of nodes at the leading edge of the NACA profile, as well as in the wake. Despite this, the FE approach used in ICI-tech still induces too much dissipation, even if similar profiles of vorticity are observed. This phenomenon appears despite the mesh adaptation procedure, which tends to position the mesh cells in the areas of the domain where events occur.

Aside from visual considerations, the comparison with the DVH method enables to characterize the dynamic performances of ICI-tech for moderate-Reynolds flows. The drag and lift coefficients, written in Eqs. (4.1), can be studied as a function of the time by replacing A cross-section area with c chord of the airfoil. A comparison between the results obtained with the DVH method and with ICI-tech is presented in Fig. 4.9. The set of results obtained using ICI-tech presented in these curves was generated thanks to meshes of $300K$ points, with a varying definition of the level-set around the airfoil. A fixed number of nodes in the computational mesh is used in the simulations, and the meshes are adapted correspondingly to ε . The results show only a little dependence on ε for C_d . The converged value is different from the one provided by the DVH method, and decreasing ε seems to have a reduced impact on the quality of the drag evaluated. The transitional behavior of the different C_d is similar to the DVH results. Concerning C_l , the dependence towards ε is more important. However, the convergence in ε tends to increase the differences with the DVH results. Moreover, the transitional regime highlights important differences between the results from ICI-tech and the DVH approach, independently from the ε chosen. This observation questions the ability to simulate

accurately unsteady flows at moderate Reynolds with the numerical methods currently implemented in ICI-tech.

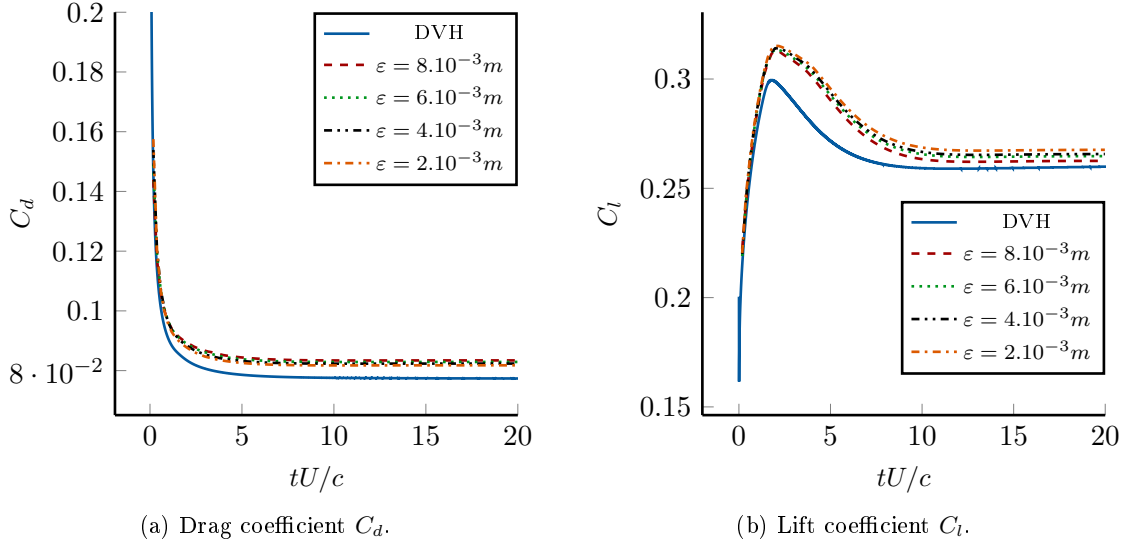


Figure 4.9: Aerodynamic coefficients measured as a function of the time around a 3D NACA 0008 with meshes of 300K nodes, with various ε .

The evolution of the converged drag and lift coefficients with the meshing has been studied more precisely. Mesh convergence studies are presented in Fig. 4.10, with a constant number of nodes. The width ε around the airfoil is progressively decreased, and the convergence of C_d and C_l is overlooked. Two different computational loads are studied, one with 60K nodes, the second one with 300K nodes. The parameter ε scales from $10^{-1}m$ to $10^{-4}m$, except for the test case with $\varepsilon = 10^{-4}m$ and 60K mesh points, which reached the limits of the meshing. The definition of small cells would require an increase in the number of points in the mesh, which is constrained. These studies confirm the trend shown on Figs. 4.9. The sensitivity of the converged C_d to ε is limited, as the error produced compared to the DVH method goes from around 11% at $\varepsilon = 10^{-1}m$ to about 5% at $\varepsilon = 10^{-4}m$. Concerning C_l , high values of ε produce a poor evaluation of C_l , while a convergence seems to be established for lower ε , with a 4% error compared to the DVH approach. The evolution of both C_d and C_l for 300K-nodes meshes is relatively smooth, while with 60K points, reaching $\varepsilon \approx 10^{-3}$ provide irregular values. The existence of convergence could be hard to prove with meshes of 60K mesh points only. Yet the precision reached with 60K points is quite similar to the one achieved with 300K points.

The differences observed in the transitory regime still need to be investigated. An explanation to the overestimation of the aerodynamic coefficients may be found in Fig. 4.11, which depicts a comparison of the pressure coefficient C_p measured around the NACA profile using the DVH method and ICI-tech. The expression of C_p is written in Eq. (4.4), with p_∞ , ρ_∞ and u_∞ respectively pressure, density and velocity of the far, unperturbed flow. For a matter of clarity of the Fig., only the curve corresponding to $\varepsilon = 4 \cdot 10^{-3}m$ is represented.

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty u_\infty^2} \quad (4.4)$$

The profile of C_p computed with this ε is representative of the results obtained with ICI-tech. In particular, the different simulations show differences at the trailing edge, which corresponds to $x/c = 0.5$. The results obtained using ICI-tech compared to the DVH ones underestimate the pressures over the area of the profile where $x/c > -0.3$,

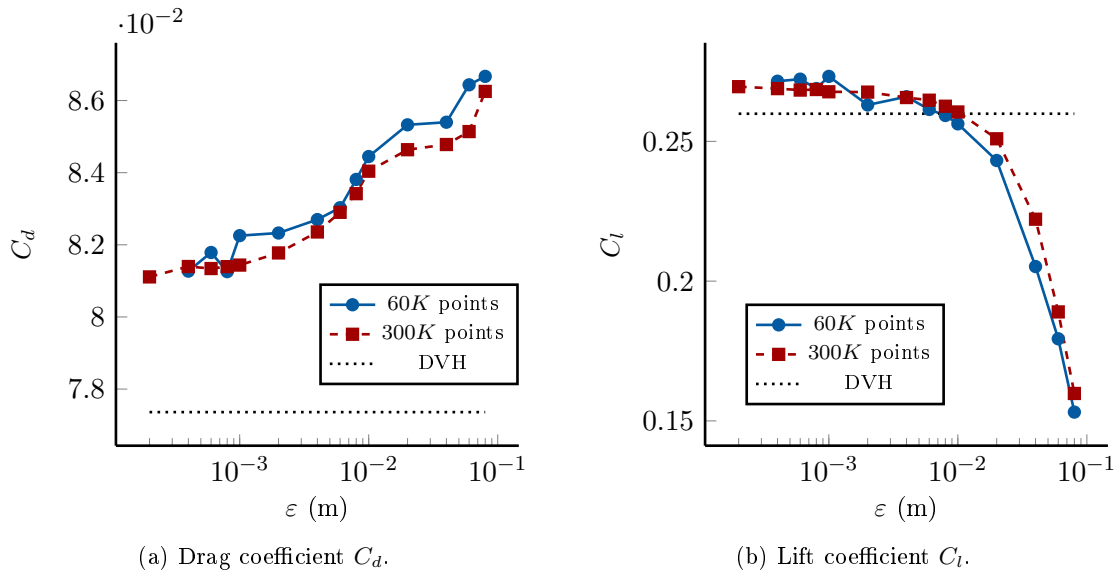


Figure 4.10: Steady values of aerodynamic coefficients measured around a 3D NACA 0008 with meshes of 60K and 300K nodes, with various ε .

while the minimum pressure measured at the leading edge is overestimated. Those differences in the distribution of pressure around the airfoil imply modifications in the flows, which may explain the transitional behavior of the lift coefficient. Different reasons may justify the imprecision of the C_p measurements, as the airfoil is reconstructed from a mesh representing the NACA 0008. The definition provided by this mesh is highly dependent on the number of points it contains, and on the position of these points on the surface. A linear interpolation is realized between those points, which tends to add imprecisions. Several datasets, for some of them featuring a considerable number of points, have been immersed to limit the dependency towards this meshing. The reconstruction of the airfoil also introduces some errors, as the level-set function is linearly interpolated in the finite elements. The errors due to the reconstruction vanish as ε tends towards zero. The pressure coefficients measured around the airfoil for different ε did not highlight this expected behavior. The flow around the airfoil can also be incorrect due to the meshing around the airfoil. However, the important number of points combined to constant results obtained with various ε disqualified this explanation. Another proposition is valid for the trailing edge only. The mesh representing the NACA 0008 is sharp at the trailing edge, and needs to be reconstructed. More than elsewhere, the minimal mesh size has a huge impact on the representation of the airfoil. At the tip of the NACA profile, the sharp edge reaches a size inferior to the minimal mesh size. Thus, the reconstruction will encounter an error on the order of magnitude of h_{min} , even for converged adapted meshes. An illustration is proposed on Figs. [4.12](#), for the immersion of the NACA 0008 with two different ε . The smallest one present a much better reconstructed airfoil, which generate a more accurate flow. If this observation may explain the C_p differences at the trailing edge, it is hard to imagine it having an influence all along the suction side of the airfoil. The last proposition, perhaps the more promising one, would consider the solver, which may not be suited for Re on the order of the thousand.

Unsteady test case at $Re = 900$

To evaluate the influence of the differences observed on the lift coefficient between ICI-tech and the DVH code, unsteady test cases can be considered. A comparison is presented between DVH results from [Rossi et al. \(2018\)](#) and ICI-tech. A NACA 0010 profile is positioned in a constant, uniform inflow at a Re of 900, with an orientation angle $\alpha = 30^\circ$. This test case has several advantages. It remains in a field of Re similar to the one pre-

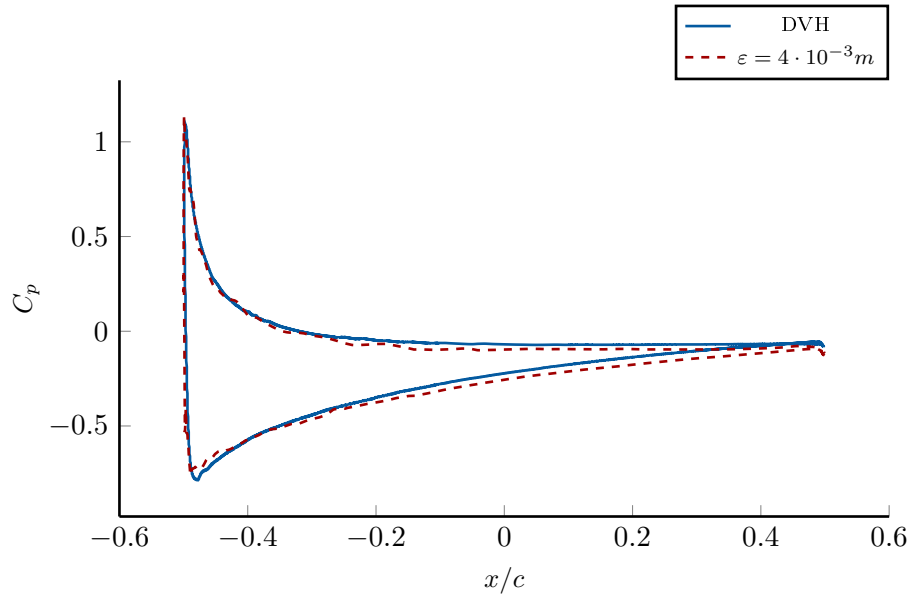


Figure 4.11: Pressure coefficient measured around the NACA 0008 (leading edge at $x/c = -0.5$, trailing edge at $x/c = 0.5$) for $\varepsilon = 4 \cdot 10^{-3}m$ and $300K$ computational points. Comparison between ICI-tech data simulated with $\varepsilon = 4 \cdot 10^{-3}m$ and DVH results.

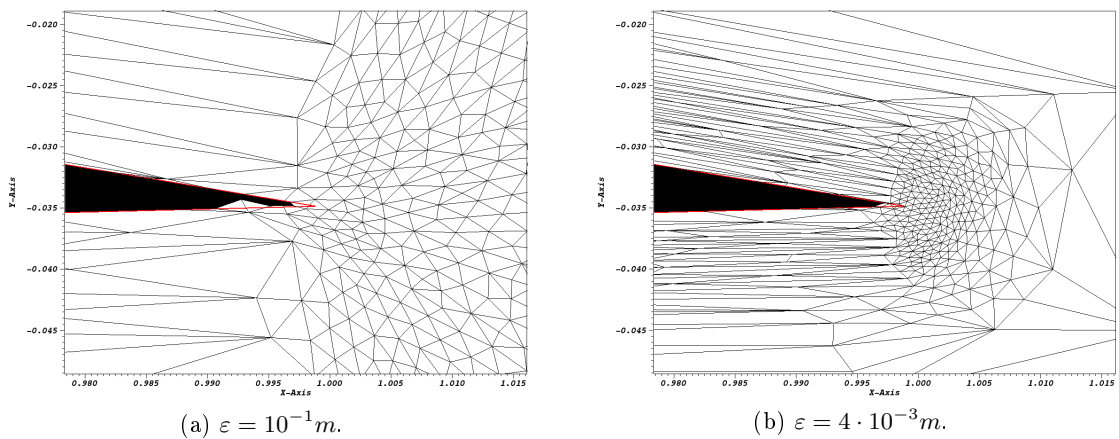


Figure 4.12: Mesh around the trailing edge of a NACA 0008 reconstructed from an immersed mesh (red, bold). Reconstruction performed on the background mesh, reconstructed NACA in black.

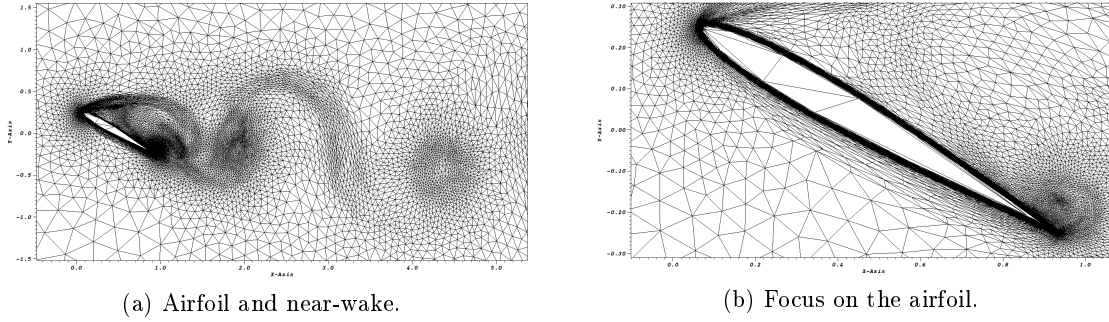
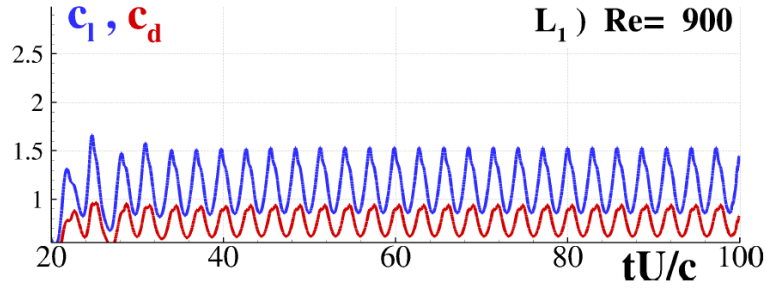


Figure 4.13: Meshing around a NACA 0010 with an orientation $\alpha = 30^\circ$ at $Re = 900$, at $tU/c = 100$. Computational mesh of $30K$ points, with $\varepsilon = 2 \cdot 10^{-2}m$

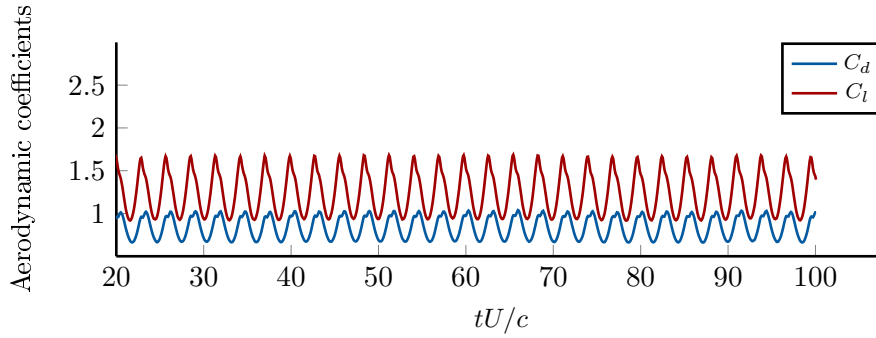
viously considered. Another advantage concerns the results obtained, which are hardly periodic. A simulation has been run on a 2D example with $\varepsilon = 2 \cdot 10^{-3}m$ and $30K$ points in the computational mesh. The meshes obtained during the computations after automatic adaptation are presented in Figs. 4.13. The objective of the simulation is to measure the aerodynamic coefficients in an unsteady context. Consequently, the refinement focuses mostly on the surroundings of the airfoil, and gathers there the major part of the mesh points. This type of meshing is acceptable to focus on the airfoil. In the wake, the lack of nodes forms larger cells, which induce numerical dissipation. The accuracy of the far wake measured with this mesh is thus questionable. As no recirculation is observed, the low density of mesh points in this region deteriorates the quality of the wake, which will be advected and does not impact the airfoil. The measurement of the aerodynamic coefficients is, consequently, independent from the representation of the far wake. The width $\varepsilon = 2 \cdot 10^{-3}m$ has been selected from the results of Figs. 4.10, as it provides interesting precision at a reduced cost.

The drag and lift measured are plotted in Figs. 4.14, where they are compared to the results from Rossi et al. (2018). The plots are presented on two different figures to depict clearly the overall dynamics of the aerodynamic coefficients. A quasi-oscillatory behavior is expected for both C_d and C_l after the transitional regime. The results computed with ICI-tech compare well with the one obtained with the DVH method in terms of shape. The slight bump observed before each peak of C_d is well reproduced over the span of time, so are the slope variations observed at the beginning of each C_l decrease. The phase is well reproduced too. For tu/c varying between 40 and 100, 21 periods are observed on both the curves provided by ICI-tech and the DVH approach. The Strouhal number, which characterizes the frequency of vortex shedding, is $St = 0.35$ for both ICI-tech and the DVH method, which is satisfying. A small, constant phase shift is observed, which is due to the time required for the establishment of the permanent regime, less important with ICI-tech. The major difference observed corresponds to the amplitudes of C_d and C_l , which are overestimated by ICI-tech. The mean drag and lift in the permanent regime are respectively $\bar{C}_d = 0.84$ and $\bar{C}_l = 1.26$, while the DVH method returned $\bar{C}_{d,DVH} = 0.77$ and $\bar{C}_{l,DVH} = 1.15$. The difference produced by ICI-tech compared to the DVH approach is of 9.1% for \bar{C}_d and 9.5% for \bar{C}_l .

A zoom on the steady state results is presented in Fig. 4.15 for tU/C in $[60, 80]$, where a phase shift is introduced in the results from ICI-tech for the sake of visualization. The overestimation of the aerodynamic coefficients is notable on this zoom. The overall shape of the curves remains satisfying. The results progressively become out of phase during this time lapse, which suggests that the dynamic of the vortex sheddings could not be correctly reproduced. However, the observations realized based on Figs. 4.14 show that it has only a minor global impact.



(a) DVH results from Rossi et al. (2018).



(b) Results from ICI-tech with $\varepsilon = 2 \cdot 10^{-3}m$ and 30K mesh points.

Figure 4.14: Aerodynamic coefficients around a NACA 0010 with an orientation of 30° , for an unsteady flow at $Re = 900$.

The results presented can be linked with the overestimation of C_d and C_l already observed in the transitory regime on Figs. 4.9, and on the steady-state on Figs. 4.10. The 9%-error approximatively found in this unsteady test case for C_d is similar to the one observed on steady lows. Concerning C_l , the gap probably has its origin in the transitional errors observed on Figs. 4.9.

To conclude on the results obtained with $Re = O(10^3)$, two distinct behaviors have been highlighted for C_d and C_l . The transitional behavior of C_d does not seem to be a problem. The main question for C_d is the convergence of the steady-state, which is very slow with ε . Concerning C_l , the convergence of the steady-state is highly dependent on ε , yet an appropriated choice leads to smaller steady-state errors than for C_d . However, ICI-tech overestimates C_l in the transitional regime. Those observations lead to a 10% overestimation of both aerodynamic coefficients measured on a simple unsteady test case. The number of nodes in the computations does not seem to be the blocking point, same for the choice of ε . The influence of the time step or a thinner mesh adaptation on the computations may be a starting point for future studies.

4.1.3 High-Reynolds, steady flows around a section of WT blade

Even if the results obtained with moderate-Reynolds flows were far from those expected, a test case more representative of operating WTs was tested. A test case with a Reynolds number $Re = 1.6M$ is considered here, which is on the order of magnitude of the flows developing around the tip of WT blades. The test case of Bak et al. (2000) is reproduced, which studied a NACA 63-415 profile with different angles of attack α in a wind flume. The geometry of the airfoil allows keeping the boundary layers attached to the profile at high-Reynolds number flows.

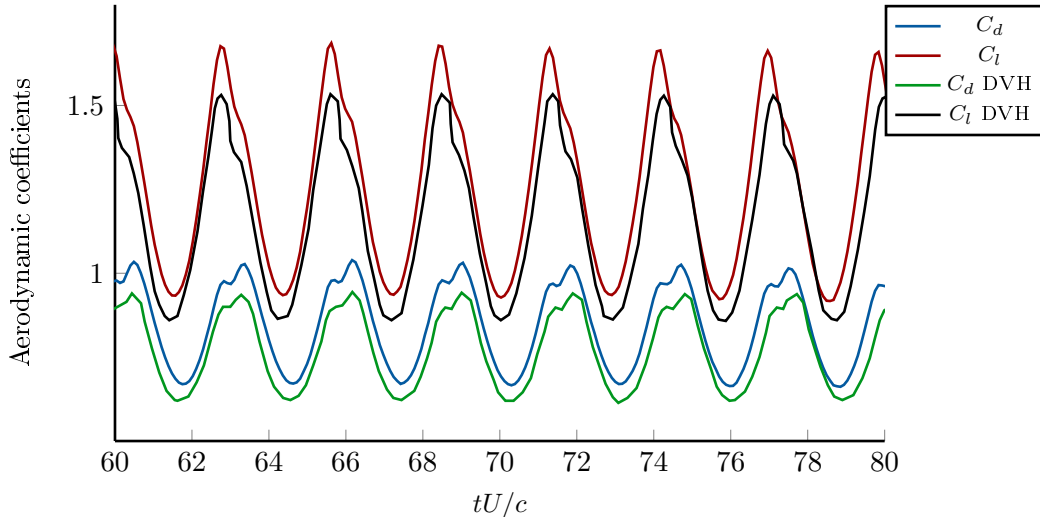
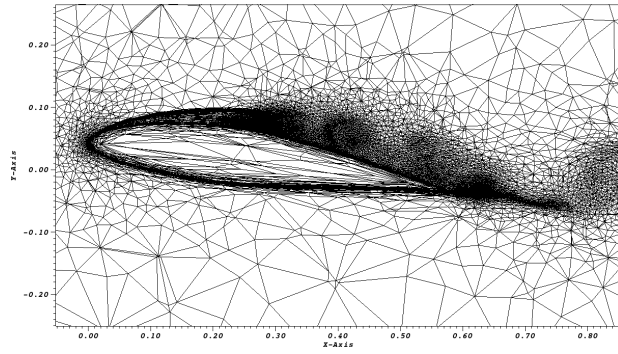


Figure 4.15: Aerodynamic coefficients around a NACA 0010 with an orientation of 30° , for an unsteady flow at $Re = 900$. Zoom on the area where $tU/c \in [60; 80]$, with a phase shift on ICI-tech’s results to present a comparison on phase.

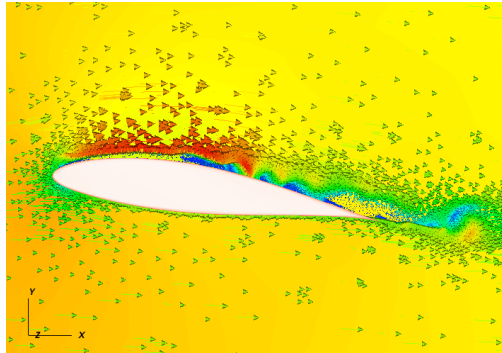
The computational test case models a 3D airfoil, of section defined as a NACA 63-415 of chord $0.6m$, positioned inside a wind tunnel where a constant, uniform inflow is prescribed. A mesh of $500K$ nodes is used to discretize the computational domain of dimensions $20mm \times 1.9m$, with $\varepsilon = 2 \cdot 10^{-3}m$, while the airfoil is represented using 100 points. Illustrations presenting the mesh used and an example of result can be found in Figs. 4.16. Converged C_d and C_l are measured on the airfoil, for different inclinations. In comparison with the experimental study, only several angles of attacks have been tested with ICI-tech. While Bak et al. (2000) sampled α in $[-5; 25]$ with steps of 1, the study conducted with ICI-tech considered only the multiples of 4 in $[-4; 20]$. Even if the set of results generated is reduced, a clear tendency can be highlighted.

The aerodynamic coefficients C_d and C_l computed are presented in Figs. 4.17. The drag provided by ICI-tech is not in accordance with the experimental measurements. For $\alpha = -4^\circ$ or $\alpha = 0^\circ$, the order of magnitude of C_d is correct, but the tendency of a decreased effort is not observed. When the angle of attack increases, from $\alpha = 4^\circ$ to $\alpha = 12^\circ$, the differences between the numerical and the experimental data reach almost an order of magnitude. As higher angles of attack are reached, e.g., for $\alpha = 20^\circ$, the differences measured are reduced. Concerning the lift coefficient, the simulations from ICI-tech compare well with the experimental campaign for $\alpha \leq 4^\circ$. However, as the angle of attack is increased, the airfoil is placed in a stall configuration, which is missed by ICI-tech. The C_l evaluated continues to increase, while the experimental measurements identified a decrease.

The large errors identified can be partly linked to the discretization of the NACA 63-415. The value chosen $\varepsilon = 2 \cdot 10^{-3}m$ is particularly large in the context of a $Re = 1.6M$ flow. However, the quality of the discretization is such that decreasing ε highlights the errors produced by the mesh representing the airfoil, as drawn in Fig. 4.18. This illustration, proposed in a 2D case with $\varepsilon = 2 \cdot 10^{-3}m$, shows the perturbations appearing at the surface of the airfoil when ε decreases. The imprecisions of the discretization of the NACA profile are of about $10^{-3}m$, which is largely superior to the minimal mesh size $h_{min} = \varepsilon/10$ for $\varepsilon = 2 \cdot 10^{-3}m$. Numerical roughness is generated, mostly around the leading edge where the influence of the discretization is the highest. This lack of detail tends to increase the drag, independently of the solver or representation method used.



(a) Slice of the 3D mesh around the airfoil.



(b) Velocity vectors and colored by pressure.

Figure 4.16: Illustration of a NACA 63-415 with an orientation $\alpha = 8^\circ$ at $Re = 1.6M$.

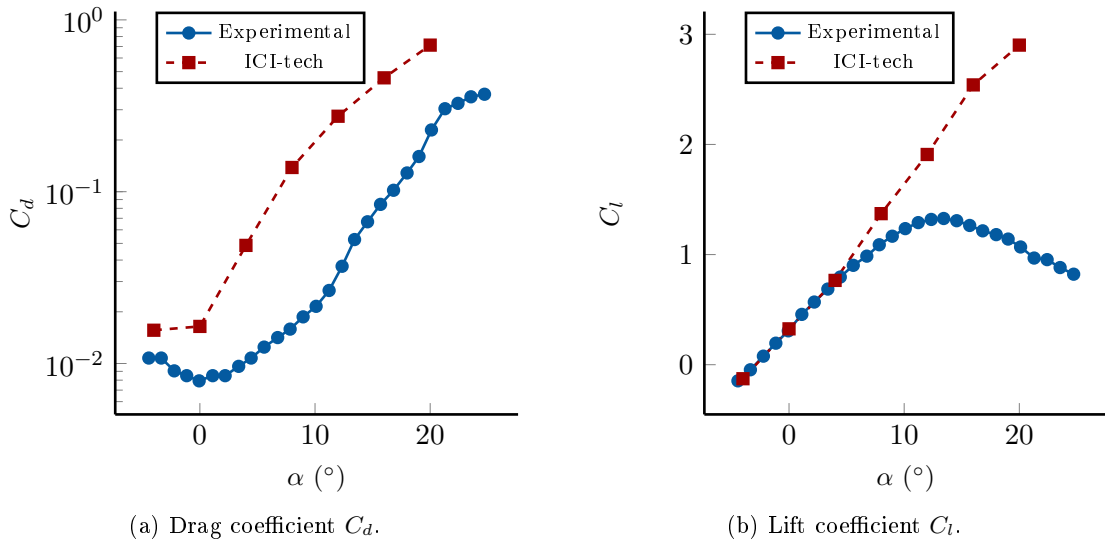


Figure 4.17: Aerodynamic coefficients measured using ICI-tech around a 2D NACA 63-415, for different angles of attack and with $\varepsilon = 5 \cdot 10^{-2}m$, compared against experimental data from [Bak et al. \(2000\)](#).

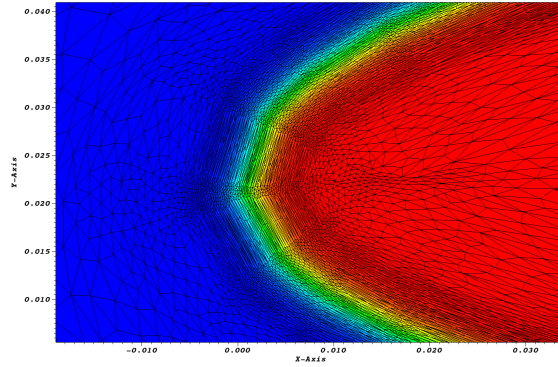


Figure 4.18: Reconstruction of a NACA 63-415 immersed using a cloud of 100 points, with $\varepsilon = 2 \cdot 10^{-3}m$, and mesh used. Scales in meters.

Moreover, when the angle of attack increases, the highest pressure around the airfoil is concentrated in this area. As the discretization is imprecise, the flows generated are not exact, and thus the pressure distribution is not appropriated. This roughness will make the lift coefficient measured questionable, especially when high angles of attack position the poorly-discretized leading edge in high pressure regions. The generation of a representation of the airfoil with a higher resolution is critical for the realization of high- Re studies on a NACA 63-415. The definition of the geometry of the airfoil is not straightforward, as it is derived from the potential flow desired around a theoretical profile, see [Theodorsen \(1933\)](#). The implementation of such a generator is still to be done. The better definition of the airfoil will improve the quality of the reconstruction, and pave the way for the use of thinner ε .

A smaller ε for the representation of an airfoil of enhanced precision should bring higher accuracy to the simulations. This will allow the generation of smaller mesh cells, which enables the capture of lower turbulent scales. The representation of the boundary layers will be improved through a more appropriated tracking of their dynamics. This is necessary, e.g., for the detection of stall, and more generally, for a precise evaluation of the aerodynamic coefficients.

A FOWT operates under the action of both the air and the sea. For the simulation of FOWTs, the verification and validation steps presented in this Sec. are necessary to guarantee the accuracy of the aerodynamics. At the same time, the generation of wave fields is important, to quantify the loads applied on the floater and to measure its displacements. The upcoming Sec. focuses on the generation and propagation of wave fields with ICI-tech.

4.2 Hydrodynamic verification – Generation and propagation of monochromatic wave fields

This Sec. intends to use the numerical tools of ICI-tech to represent sea states as accurately as possible. The generation of wave fields has been an important part of the works realized within the scope of this thesis. The validation steps processed with different wave makers are presented in the following. Among the objectives of this study, the quantification of the effects produced by the anisotropic mesh adaptation procedure is important, as it is not classical for marine applications in the literature. Comparisons are drawn between wave fields depending on tuned simulation parameters, in domains meshed using isotropic (Subsec. [4.2.1](#)) or anisotropic (Subsec. [4.2.2](#)) paradigms. Source-term and physical wave makers are compared for each type of meshing, from the implementations proposed in Subsec. [3.3.2](#). The realization of no-flow tests for hydrostatic cases is a con-

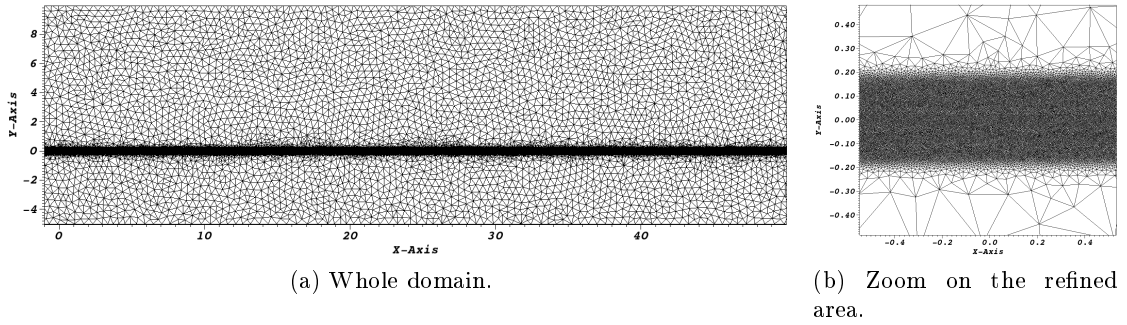


Figure 4.19: Mesh of the computational domain used for wave generation using a source-term wave maker. Scales in meters.

sequence of those studies, and the results are presented in Subsec. [4.2.3](#)

In this Sec., multiphase flows are considered, which force the representation of several phases. Different interface thicknesses may be used for the phases. As this Sec. focuses on the propagation of wave fields, the interface thickness of the free-surface is always named ε . When other phases will be considered, their characteristics will be detailed on a case-by-case basis.

It must be noted that the simulations presented in this Sec. have been realized with erroneous definition for the sponge layer, lately detected. The simulations could not be generated again, because of the considerable time expenses required. An impact can be identified on the simulations, especially around the sponge area. On the contrary, the early propagation of waves is only slightly modified by this error.

4.2.1 Verification of the solver using constant, isotropic meshes

Wave field generated using a source-term wave maker

The accurate simulation of sea states requires the validation of the solver under certain conditions. The generation and propagation of waves is studied using the utilities presented in Subsec. [3.3.2](#), i.e., the source-term and physical wave makers. The resolution of the NS equations always tend to create dissipation, especially with low orders in space and time. A particular focus is thus placed on the propagation of wave fields, in order to limit the numerical dissipation.

The damping observed on propagated waves is studied in NWTs. In order to identify the contributions of the solver to the damping observed, a constant mesh, uniformly and isotropically adapted around the free-surface, is used. The mesh adaptation procedure is unactivated, to suppress its eventual effects on the simulations.

A first test case considers a source-term wave generator. The considered mesh features cells of size $h = 5 \cdot 10^{-3}m$, concentrated in a region of $0.2m$ -width around the $y = 0$ -axis. The corresponding mesh contains $426K$ nodes, and is drawn in Figs. [4.19](#). The mesh used is coherent with the test case simulated. A level-set function of $\varepsilon = 5 \cdot 10^{-2}m$ is defined to represent a monochromatic wave field of amplitude $2.5 \cdot 10^{-2}m$, of frequency $1s^{-1}$ and of wavelength $2m$, while a water depth of $5m$ is considered. These parameters imply a crest-to-crest vertical distance of ε . Consequently, the transition of the level-set function is represented using 10 cells, while a period of the wave is split among 400 cells. The source terms being applied in the transition area of the level-set function, see Subsec. [3.3.2](#), a correct representation can be obtained using the mesh presented. The

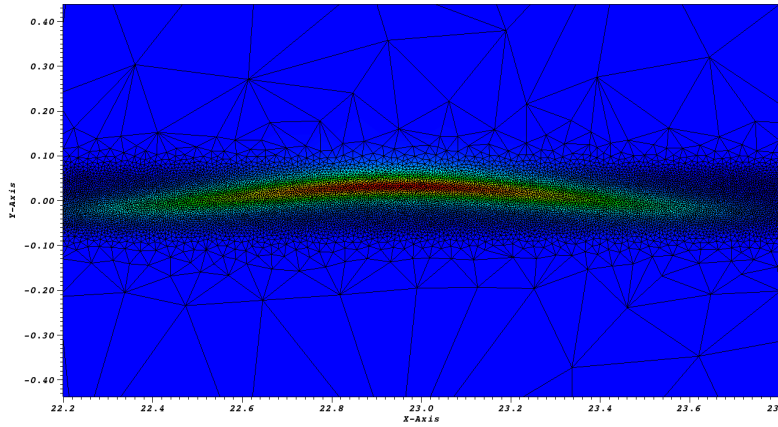


Figure 4.20: Source area generating the wave field, scaling from zero in blue regions to an oscillating maximal value in the red area, and mesh used for the representation. Scales in meters.

source term for an extreme positioning of the free-surface is presented in Fig. 4.20 for a wave crest. The source area remains inside the densely-meshed region, which guarantees the correct generation of the wave field.

The propagated wave fields from the simulations realized with ICI-tech are presented in Fig. 4.21. Two curves are plotted, for $\Delta t = 10^{-2}s$ and $\Delta t = 10^{-3}s$, both have been obtained with the same mesh. The source term is acting on the free-surface between $x = 22m$ and $x = 24m$, and the waves are then propagated in both the x - and $(-x)$ -directions. A first observation concerns the damping, which is much more important with $\Delta t = 10^{-2}s$. The numerical dissipation, with the highly-refined mesh used, is tackled using $\Delta t = 10^{-3}s$. The phase does not seem to be very sensitive to the numerical dissipation, which mostly acts on the amplitude of the oscillations. Two other observations can be made. The waves generated are not oscillating exactly around $y = 0$, i.e., the mean free-surface height is upper than $y = 0$. This tendency is probably coming from the wave generation technique, which modulates the gravity in the source area. The simulation starts with an upwards-directed wave, obtained thanks to a decrease in gravity. Thus, an integration over the time of the gravity force will return smaller values than the reality, except at the end of the generation periods. The second observation concerns the $\Delta t = 10^{-2}s$ -curve. The oscillations have a similar amplitude in the source region, but the free-surface solved is located slightly below the $\Delta t = 10^{-3}s$ one. This may be due to mass loss induced by the numerical dissipation, which tends to reduce the energy provided by the source to the free-surface. Finally, even if they can hardly be identified on the Fig., parasitic velocities are found on the sides of the domain, where the free-surface is not supposed to move. These velocities are concentrated around the air/water interface, where a density drop is found. They tend to disturb the transport of the free-surface, even if they disappear once waves are propagated.

A more detailed study of the wave field generated is proposed in Figs. 4.22, based on the contributions propagated rightwards. The position of wave crests and their height are studied in comparison to a reference number given to each peak. This procedure allows to study specifically the propagation, to check if the velocity of the wave field is constant, and the numerical dissipation. The first seven peaks are studied, i.e., those located between $x = 23m$ and $x = 33m$. The position of peaks found in Fig. 4.22a is found to be linear, even if the prescribed wavelength of $2m$ is not respected. A measured wavelength of $1.54m$ is found, which may be due to the wave generation technique. The amplitudes of the peaks, presented in Fig. 4.22b, needs to be regarded closely. The first

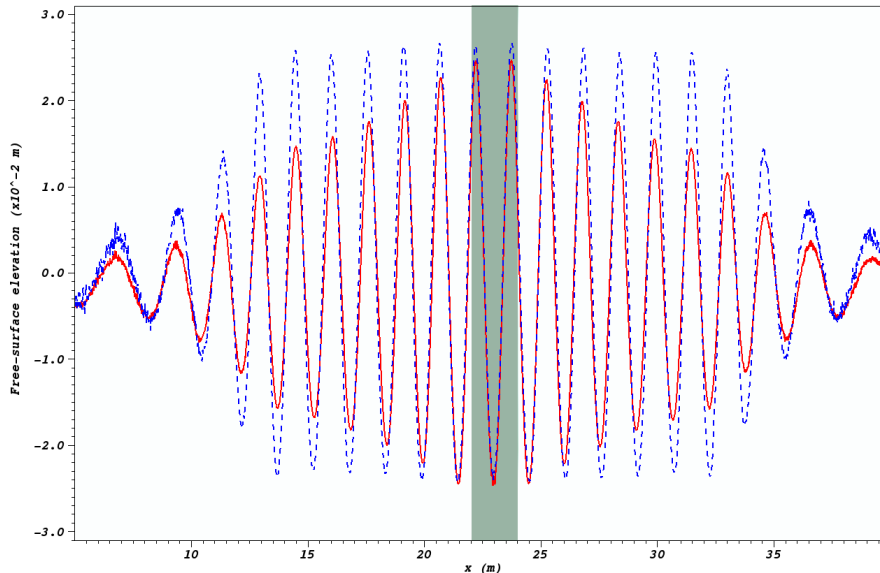


Figure 4.21: Wave field of amplitude $2.5 \cdot 10^{-2}m$, of frequency $1s^{-1}$ and of wavelength $2m$ simulated with ICI-tech using an isotropic mesh, with $\varepsilon = 5 \cdot 10^{-2}m$ and different time steps. Red, plain: $\Delta t = 10^{-2}s$. Blue, dashed: $\Delta t = 10^{-3}s$. Green area: source generation area.

crest, with a peak number of 1, is still located in the generation area, which tends to increase its amplitude. Between the next five crests, only a 2% difference is observed between the extreme positions of the peaks. The last crest, which also provides the smallest amplitude, is due to the hydrodynamic effects observed at the beginning of the simulation, which tend to prevent the generation of waves due to inertia. The wave field generated with the source-term wave maker tends to overestimate the amplitude of the wave field, while the wavelength is not properly respected. This questions the accuracy of the wave maker implemented, even if such a study allows getting first evaluations of the numerical dissipation observed during the propagation of wave fields.

Wave field generated using a physical wave maker

A new test case is presented, which uses this time a physical, hinged wave maker as defined in Subsec. [3.3.2](#). A monochromatic wave field of amplitude $0.1m$, of frequency $0.5s^{-1}$ and of wavelength $6m$ is created by a wave maker in a $5m$ -depth NWT, actioned thanks to HOS-NWT's outputs. The physical wave maker relies on the enforcing of boundary conditions using a physical object defined inside the computational domain. This object is moved, and BCs are imposed regarding data provided by HOS-NWT. The computational mesh has to take these elements into account. The mesh generated is presented in Figs. [4.23](#). The meshing around the free-surface is comparable to the one used for the source-term NWT, except that it is wider to account for the larger waves generated. The interface thickness ε of the level-set representing the water phase is defined smaller than the one used with the source-term wave generator, at $1 \cdot 10^{-2}m$ instead of $5 \cdot 10^{-2}m$. This choice is motivated by the behavior of the free-surface around the wave maker, which needs to be refined to guarantee good accuracy of the results. The precision of the results can still be ensured by the number of mesh cells found in the transition area, which is of about 10. As explained in Subsec. [3.3.2](#), complex hydrodynamic effects appear near the piston, and rapidly vanish as they move away. To limit the computational load linked with those turbulent effects and to increase the stability of the simulations, the viscosity of the fluids are multiplied by 100. As the free-surface flows are inertia-dominated, this correction is supposed to produce limited errors. This increased viscosity is used in all the simulations involving a physical wave maker presented in this

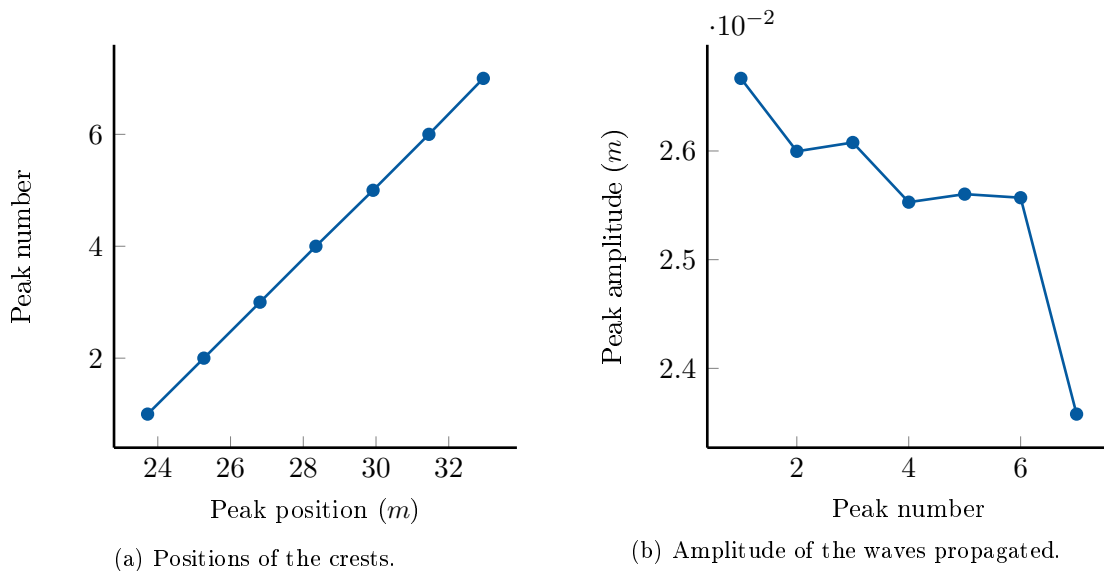


Figure 4.22: Study of the wave crests propagated rightwards on Fig. 4.21.

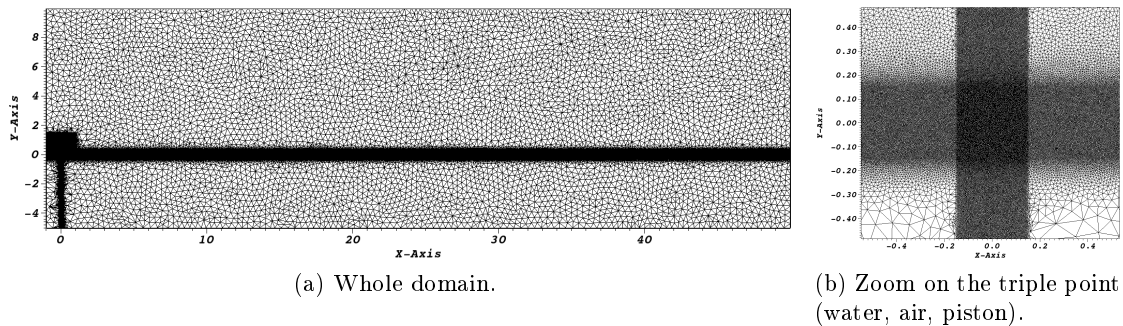


Figure 4.23: Mesh of the computational domain used for the generation of waves using a physical wave maker. Scales in meters.

thesis.

Compared to the mesh from Figs. 4.19, the one generated is more sophisticated. As the piston is placed on the left-hand side of the domain, its geometry needs to be meshed. The regions around the piston also need to be refined specifically, to avoid extreme aerodynamic events. These modifications imply a higher computational load, highlighted by the size of the mesh generated, of 1.19M points. The presence of a physical wave maker in the domain also creates a triple point, at the interface of three different phases. Here the free-surface, i.e., the air/water interface, is excited by the solid. The convergence of the schemes is more challenging in this area, thus a thin meshing is required in this region, and the computational cost increases. The addition of patches is justified by the study focusing on the free-surface on a fixed mesh. The focus is placed on the accuracy of the wave field generated, and not on the reduction of the computational load.

The mesh presented is used in simulations realized with ICI-tech. The results obtained are presented in Fig. 4.24. The simulations performed with $\Delta t = 10^{-2}s$ come with a non-negligible damping, along with important mass losses. This results in an averaged free-surface level progressively going down, which considerably deteriorates the quality of the resolution. Yet, the wave field obtained is in phase with the prescribed one. These mass losses may be directly linked with the temporal discretization of the problem, as the motion of the wave maker is more important between two time increments with a

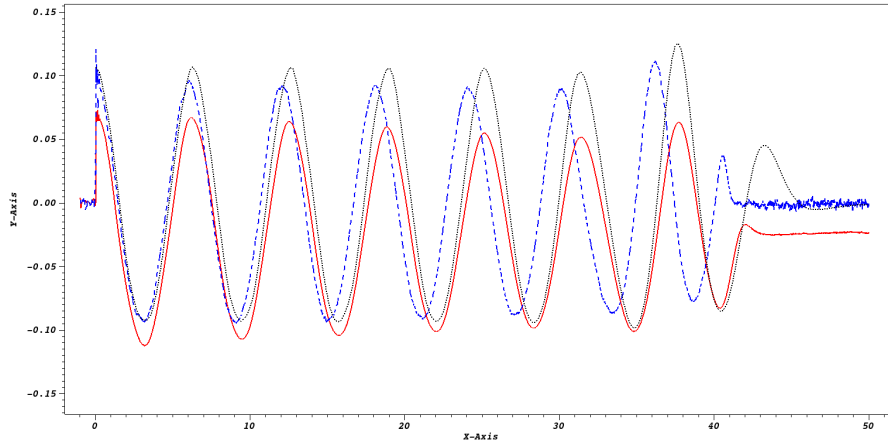


Figure 4.24: Monochromatic wave field of amplitude $0.1m$, of frequency $0.5s^{-1}$ and of wavelength $6m$ at $t = 30s$ simulated with ICI-tech using an isotropic mesh, with $\varepsilon = 1 \cdot 10^{-2}m$ and different time steps. Black, dotted: wave field from HOS-NWT. Red, plain: $\Delta t = 10^{-2}s$. Blue, dashed: $\Delta t = 10^{-3}s$. Scales in meters.

large time step. The transition between water to wave maker in cells may be too fast for a good resolution of the flows. The simulation that ran with $\Delta t = 10^{-3}s$ features smaller damping, and no mass losses are observed. A phase shift appears in this configuration, which has not been found in any other simulation. More investigations are conducted to confirm or infirm this observation. The incorrect implementation of the sponge layer used when those results were generated may be the origin of this phase shift. This tendency was not observed on the previous test case, as the propagation of the wave field did not reach the sponge layers. The free-surface at the right-hand side of the domain is perturbed, because of parasitic velocities generated in this area. Still, only a minor impact is found on the wave field generated, as these velocities tend to disappear when the wave train is propagated.

To conclude on the quality of the waves generated using isotropic meshes, the different results presented in this Subsec. highlight the potential to simulate accurate monochromatic wave fields using ICI-tech. Depending on the time step chosen, different levels of damping were observed. Mass losses could be identified when a physical wave maker was used, in a situation featuring damping too. Phase shift has been highlighted in only one configuration, where it looked like a perturbation impacted strongly the results. The source-term simulations also proved that rather large ε regarding the wave amplitude are acceptable. Those different observations are rather satisfying, as accurate wave trains could be generated. However, the simulations presented have been realized using massive meshes, between $500K$ and $1M$ computational points in 2D. Getting the same level of refinement in 3D involves an extreme computational load, especially with one or several FOWTs in the domain. Moreover, the time step required to limit the damping is small for cases where monochromatic wave fields are simulated. The influence of the time step for the simulation of monochromatic wave fields of higher frequency or smaller wave length is still to be investigated. These studies have not been realized partly because of the time needed to process the computations. To simulate $30s$ of physical time, with a load of $10K$ computational points per core and a time step $\Delta t = 10^{-3}s$, about $72h$ are required. Still, further observations could change the interpretation of the results presented. In particular, damping and/or mass losses may increase from monochromatic to irregular wave field simulations if Δt is kept identical. Irregular wave fields represent real sea states, where waves of small wavelength or high frequency may occur. The dependency towards the time step and mesh size seems to be important. Consequently, the time and space discretizations will need to be designed for the most challenging waves, leading to

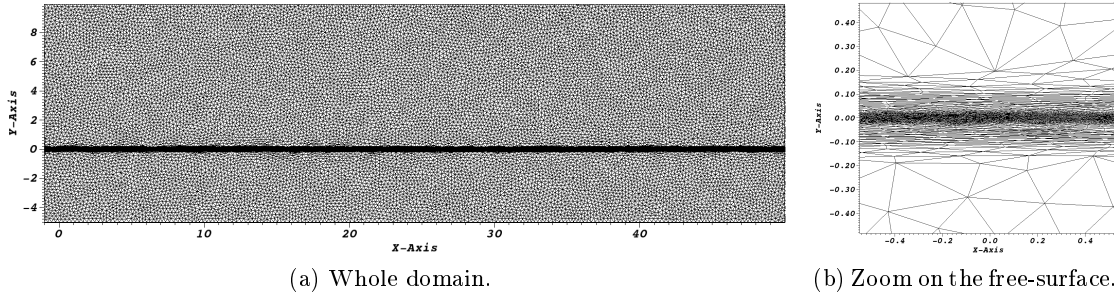


Figure 4.25: Anisotropic mesh of the computational domain used for the generation of waves using source terms. Scales in meters.

small mesh sizes and time steps. This kind of observation would question the capacity of the current solver to solve accurately irregular wave fields, especially for 3D situations, where diminishing the mesh size increases rapidly the computational effort.

4.2.2 Propagation of monochromatic wave fields using automatic, anisotropic mesh adaptation

The isotropic simulations presented in the previous Subsec. do not exploit all the potential of ICI-tech. Meshes of $426K$ to $1M$ nodes are considerable, especially for 2D applications. If it is affordable in 2D, such a precision becomes prohibitive in 3D. Consequently, a transition towards an anisotropic meshing is realized. During the simulation, the mesh will be adapted on the free-surface and on the velocities. This aims at capturing the physical effects as accurately as possible, and thus to transport the free-surface with the best precision. The test cases from Subsec. 4.2.1 were replicated, this time with an activation of the mesh adaptation procedure.

Wave field generated using a source-term wave maker

The test case using a source-term wave generator is reproduced, this time using an automatically-generated anisotropic mesh. An important number of nodes, $100K$, is used to guarantee the quality of the meshing of the free-surface. The mesh obtained is presented in Figs. 4.25, in particular with a focus on the free-surface. The meshing around the free-surface is not uniform. The "intensity" of the meshing is defined as depicted in Fig. 3.5. By default, the adaptation is performed from the error estimator, built on the second derivative of the level-set, which tends to create an unadapted area around the free-surface. A Dirac function then needs to be defined. The adaptation is pondered between the level-set and the Dirac functions to concentrate mesh cells in this zone. Similarly to the test case presented in Subsec. 4.2.1, a monochromatic wave field of amplitude $2.5 \cdot 10^{-2}m$, of frequency $1s^{-1}$ and of wavelength $2m$ is generated. In order to reproduce the study done using an isotropic mesh, the same width of the water level-set $\varepsilon = 5 \cdot 10^{-2}m$ and the identical two time steps $\Delta t = 10^{-2}s$ and $\Delta t = 10^{-3}s$ are used. In both of the simulations, the remeshing is performed every 11 increments, i.e., every $0.11s$ for $\Delta t = 10^{-2}s$ and every $1.1 \cdot 10^{-2}s$ for $\Delta t = 10^{-3}s$.

The results are presented in Fig. 4.26. Despite a thin meshing around the free-surface, featuring a precision similar to the one obtained in the isotropic test case, important damping is observed with both time steps. While $\Delta t = 10^{-2}s$ generates waves of correct amplitude in the source area, rapid damping deteriorates the wave field. With $\Delta t = 10^{-3}s$, the damping is slightly reduced. However, the waves generated are of

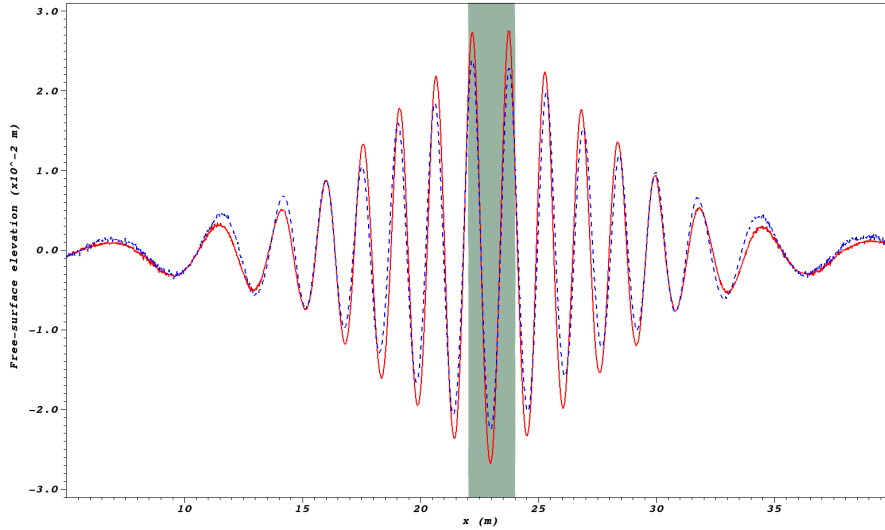


Figure 4.26: Wave field of amplitude $2.5 \cdot 10^{-2}m$, of frequency $1s^{-1}$ and of wavelength $2m$ simulated at $t = 10s$ with ICI-tech using an anisotropic mesh, with $\varepsilon = 5 \cdot 10^{-2}m$ and different time steps. Blue, dashed: $\Delta t = 10^{-3}s$. Red, plain: $\Delta t = 10^{-2}s$. Green area: source generation area.

smaller amplitude, which is surprising. The effects of remeshing may induce additional dissipation, which perturbs the generation of the wave field.

To validate the behavior of the wave generation procedure coupled to the automatic mesh adaptation, a comparison is drawn on the same test case, this time at $\Delta t = 10^{-3}s$, with different mesh adaptation frequencies. The results are presented in Fig. 4.27, where the dashed, blue curve corresponds to an adaptation every $1.1 \cdot 10^{-2}s$, as already presented. The green curve corresponds to an adaptation every $0.11s$, which limits the impact of the remeshing on the results. The waves measured in the source generation area confirm the impact of the mesh adaptation on the quality of the waves generated. With a decreased adaptation frequency, less damping is observed in the impulsing area. However, damping is still largely observed in the propagation of the waves, in proportions comparable to the test case with adaptation every $1.1 \cdot 10^{-2}s$.

In terms of numerical dissipation, the data plotted in Fig. 4.27 can be compared to the results obtained with an isotropic meshing presented in Fig. 4.21. A rapid comparison of the free-surface elevations for $\Delta t = 10^{-3}s$ and similar configuration, except for the computational mesh, highlight a major damping obtained with the anisotropic meshing. To verify that this dissipation is not produced by the wave-generation method, a similar study is conducted with a physical wave maker.

Wave field generated using a physical wave maker

The influence of the mesh adaptation is tested for a physical wave generation procedure too. The test case considered in Subsec. 4.2.1 is reproduced. A monochromatic wave field of amplitude $0.1m$, of frequency $0.5s^{-1}$ and of wavelength $6m$ is prescribed. The free-surface is represented using $\varepsilon = 10^{-2}m$. An anisotropic mesh of $50K$ nodes is used, refined around both the free-surface and the piston. The initial mesh is presented in Figs. 4.28a. The reduction of the number of cells is realized in the anisotropic mesh adaptation thanks to the deformation of mesh cells to follow the demarcation of the geometries immersed. Consequently, compared to an isotropic mesh, the concentration of cells around both the free-surface and the piston is similar only in the direction normal to the geometries. In the other directions, when the phases immersed have plane surfaces,

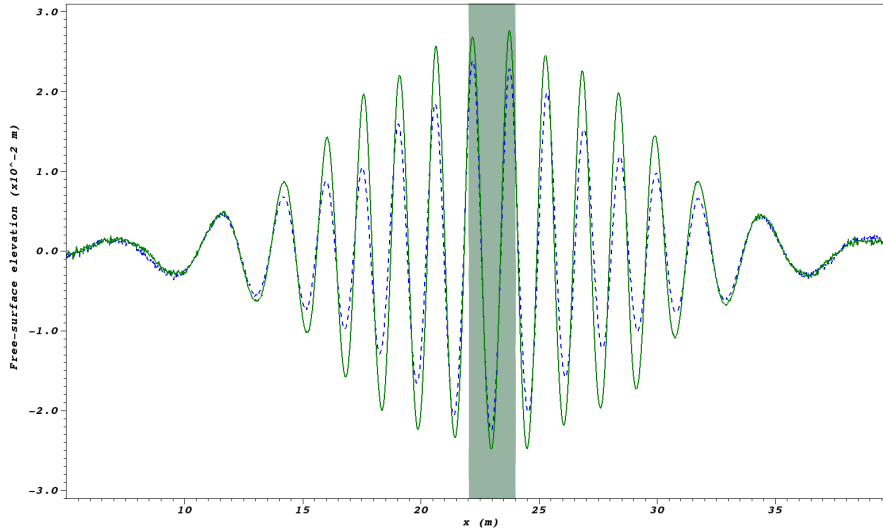


Figure 4.27: Wave field of amplitude $2.5 \cdot 10^{-2}m$, of frequency $1s^{-1}$ and of wavelength $2m$ simulated with ICI-tech at $t = 10s$ using an anisotropic mesh, with $\varepsilon = 5 \cdot 10^{-2}m$, $\Delta t = 10^{-3}s$ and adaptation every 11 time increments. Blue, dashed: adaptation every $1.1 \cdot 10^{-2}s$. Green, plain: adaptation every $0.11s$. Green area: source generation area.

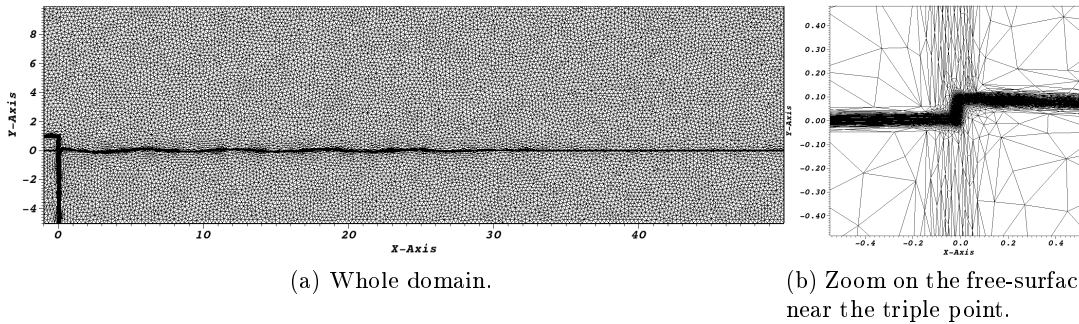


Figure 4.28: Anisotropic mesh of the computational domain used for the generation of waves using a physical wave maker. Scales in meters.

the concentration of cells tends to be largely reduced. An increase in the number of nodes tends to limit the deformations of mesh cells, as remaining mesh points are placed in the interest areas.

The results obtained for this test case, which are presented in Fig. [4.29](#), do not reproduce the level of accuracy from isotropic simulations. Different time steps of $10^{-2}s$ and $10^{-3}s$ are used, with adaptation every $7 \cdot 10^{-2}s$ or $7 \cdot 10^{-3}s$. For both of these simulations, important damping is observed. This damping is accentuated with $\Delta t = 10^{-3}s$ and adaptation every $7 \cdot 10^{-3}s$. On the contrary, in isotropic simulations, a reduced time step tended to reduce the damping. This highlights a direct link between the adaptation of the computational mesh and the damping. This simulation provides results in phase with the prescribed one, as well as the one obtained with $\Delta t = 10^{-2}s$. On the contrary, a phase shift is progressively established with $\Delta t = 10^{-3}s$ and adaptation every $7 \cdot 10^{-2}s$. The displacements of the mesh are not frequent enough to follow the wave being transported. The propagation of the wave reaches the limit of the adapted area, where the size of mesh cells tends to increase rapidly. The transport is less accurate when solved on large cells, which explains the progressive phase delay. At the right-hand side of the domain, the differences found around $x = 40m$ are due to the sponge layer, which is designed differently between HOS and ICI-tech. Except for the phase shift, whose origin

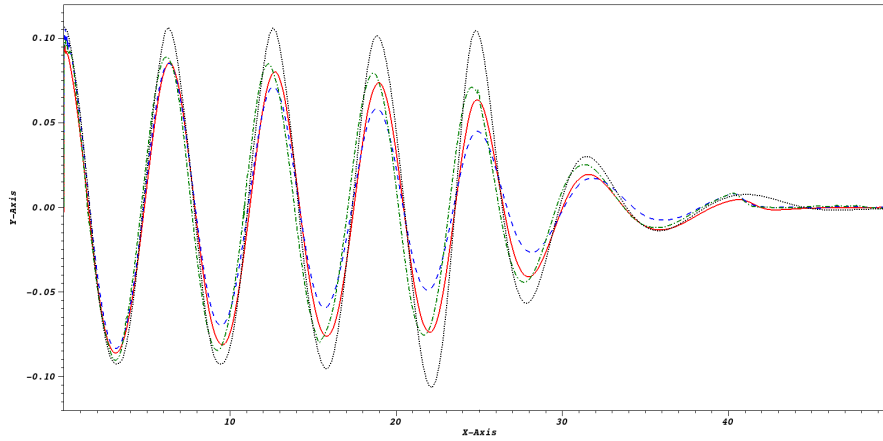


Figure 4.29: Monochromatic wave field of amplitude $0.1m$, of frequency $0.5s^{-1}$ and of wavelength $6m$ at $t = 20s$ simulated with ICI-tech using an anisotropic mesh, with $\varepsilon = 1 \cdot 10^{-2}m$ and different time steps. Black, dotted: wave field from HOS-NWT. Red, plain: $\Delta t = 10^{-2}s$, adaptation every $7 \cdot 10^{-2}s$. Blue, dashed: $\Delta t = 10^{-3}s$, adaptation every $7 \cdot 10^{-3}s$. Green, dash-dotted: $\Delta t = 10^{-3}s$, adaptation every $7 \cdot 10^{-2}s$. Scales in meters.

has already been discussed regarding the implementation of the sponge layer, the conclusions drawn here are in accordance with those from the simulations using a source-term wave maker and mesh adaptation.

The results obtained with the anisotropic meshing are globally disappointing. With a minimal mesh size similar to the one used with isotropic test cases and a huge number of nodes in the computational meshes, the accuracy achieved in isotropic situations could not be reproduced. Several corrections enable to limit the damping observed, e.g. through a reduction of the time step or works on the adaptation frequency, but remain insufficient. The origins of the numerical dissipation are hard to identified. The shares of the NS resolution at each time step or of the mesh adaptation itself have not been quantified. Moreover, if an isotropic mesh guarantees that the numerical dissipation will be more or less equivalent at each time step, this statement can not be transposed to the case of anisotropic meshes. The mesh adaptation is defined using an *a posteriori* error estimator, thus the mesh produced at an instant t is optimal for the previous time step $t - \Delta t$, with mesh cells located around the free-surface. When the characteristics of the simulations are correctly chosen, the physical effects reproduced have a slow evolution regarding the mesh sizes and time iterations, which guarantee the quality of the results. However, in the context of free-surface flows, the transport of the different phases moves the fluid interface towards the borders of the adapted area, which tends to increase the numerical dissipation per time step. To that extent, the increasing numerical dissipation created at each time step could play an important role in the deterioration of the wave field. Once again, a quantification of this effect would be required, along with a comparison with the dissipations created by the mesh adaptation procedure. A simple outlook on the dissipation produced per time increment is illustrated in Fig. 4.30. The height of the different peaks and curves and their evolution with time would be interesting to know. A numerical dissipation progressively increasing per time step is proposed, which is reinitialized at each adaptation. The fact that a newly adapted mesh reduces the dissipation produced at each time increment seems indubitable, which explains the shape of the dashed curve. This is not the case for the regularity, periodicity or relative sizes of the different contributions to numerical dissipation. A comprehensive comparison could give guidelines for the definition of an optimal adaptation frequency. The ideal would be to link it with physical parameters, which seems possible for monochromatic wave fields,

maybe less for real sea states.

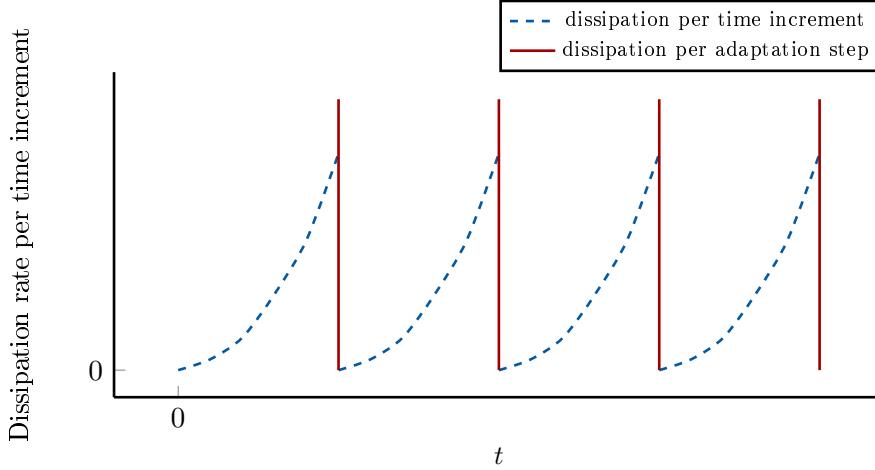


Figure 4.30: Simple view of the rate of dissipation per time increment during a simulation with periodic mesh adaptations.

Another limitation may be due to the definition of the anisotropic meshing as it is presently defined, and its relation with the physics. The adapted area is based exclusively on the level-set function representing the free-surface. Consequently, a thin meshing is obtained only in an area of width ε around the free-surface. This definition mostly ignores the water dynamics below the water level, which play an important role in the propagation of wave fields. Moreover, the anisotropic characteristic derived from the level-set function is able to detect the curvature of the free-surface to adapt on it. But when waves of small steepness are defined, mesh cells stretched in the direction of the propagation of the wave field tend to be generated. The reduction of the number of mesh cells in this direction reduces the definition of the waves, which tends to increase the numerical damping produced. An immediate correction is to limit the stretching of mesh cells. However, this leads to an increase in the number of mesh points used in the computations, which reduces the potential of the anisotropic meshing, in particular for 3D applications. As the mesh used in the anisotropic simulations is adapted in a quasi-isotropic way around the free-surface, it must not be the source of the damping observed in this Subsec.

This thesis focused on the development of a simulator for FOWTs. To that extent, 3D applications need to be considered, for which the use of isotropic meshes is unrealistic. The propagation of wave fields realized with anisotropic meshes thus needs to be improved. A preliminary study, aiming at developing test cases integrating the anisotropic mesh adaptation procedure to propagate wave fields more accurately, has been realized. A framework is developed for the definition of a refined mesh under the free-surface, i.e., where the hydrodynamic effects are expected to have the greatest impact on the shape of the waves simulated. This refinement is a way to conserve the anisotropic characteristic, while constraining the mesh size in an attempt to reduce the numerical dissipation. The procedure detailed in the following acts as a patch, positioned under the free-surface, where a thinner mesh is generated.

The condition presented in Eq. (4.5) defines the areas where the computational mesh needs to be refined. The level-set of the water phase Φ_ε is used, along with the amplitude of the wave field generated A_w . A threshold ratio r_{Thr} allows to easily tune the size of

the refined zone.

$$\begin{cases} \Phi_\varepsilon > 0 \\ y > -r_{Thr}A_w \end{cases} \quad (4.5)$$

The mesh can then be adapted thanks to a tuning of the metric field presented in Sec. 2.2. The modification of the mesh intends to control the size of the mesh cells along both the x and y direction. To that extent, a metric field similar to the one presented in Eq. (4.6) is required in the refined area, with h_x and h_y the sizes of the mesh in the x and y directions.

$$\mathcal{M}_i = \begin{bmatrix} 1/h_x^2 & 0 \\ 0 & 1/h_y^2 \end{bmatrix} \quad (4.6)$$

The methodology followed here takes advantage of both the adaptation on the free-surface and a further refinement based on geometrical criteria. The definition of the metric as realized in Subsec. 2.2.2 needs to be conserved, while supplementary constraints are defined in Eq. (4.6). To restrict the implementations to high-level encoding, a definition of the new metric field in two steps has been realized. The adaptation is first done exclusively on Φ_ε . The metric field \mathcal{M}_i obtained is then summed with the expression proposed in Eq. (4.6). In the area to refine, the initial eigenvalues of \mathcal{M}_i correspond to the local mesh size along each direction. With well-tuned prescribed mesh sizes, $1/h_x^2$ and $1/h_y^2$ are much higher than the eigenvalues, and thus get the most attention in the remeshing. In the areas being adapted by both steps, this procedure tends to generate thinner mesh cells, which increases the computational costs and the expected precision of the simulations.

The tuning of the parameters h_x and h_y is a key feature of this refinement area. This topic is commonly discussed, e.g., in Calderer et al. (2018). In order to get a first idea of the level of refinement required, a rather thin description of the refined zone is proposed, using Eqs. (4.7). This definition allows getting 100 cells along a wavelength, and 20 cells from crest to trough.

$$h_x = \frac{\lambda_w}{100} \quad (4.7a)$$

$$h_y = \frac{A_w}{10} \quad (4.7b)$$

This mesh generation process is suited for monochromatic wave fields, as an amplitude A_w has to be provided. It does not account for regions where the refinement is not required, e.g., in the sponge layer. Improvements could easily be designed and implemented, from a modification of the condition in Eq. (4.5). However, the point followed here was to improve the description of the propagated wave field on a 2D test case. Little attention was thus placed on the optimization of the meshing. Similarly, a thinner tuning of h_x and h_y could lead to either better precision achieved with the results, or to a reduced computational load.

A zoom on a mesh generated using this methodology is presented in Fig. 4.31. The test case considered here uses a physical wave maker, even if the mesh refinement proposed is independent of the wave generation tools. The computational domain, similar to the one presented previously, is meshed using about 150K points and uses $\varepsilon = 10^{-2}m$. Several comments can be made about the distribution of nodes. An important concentration is again found around the piston and the free-surface. In addition, the regions respecting Eq. (4.5) are refined too. The adapted area is widened, especially under wave

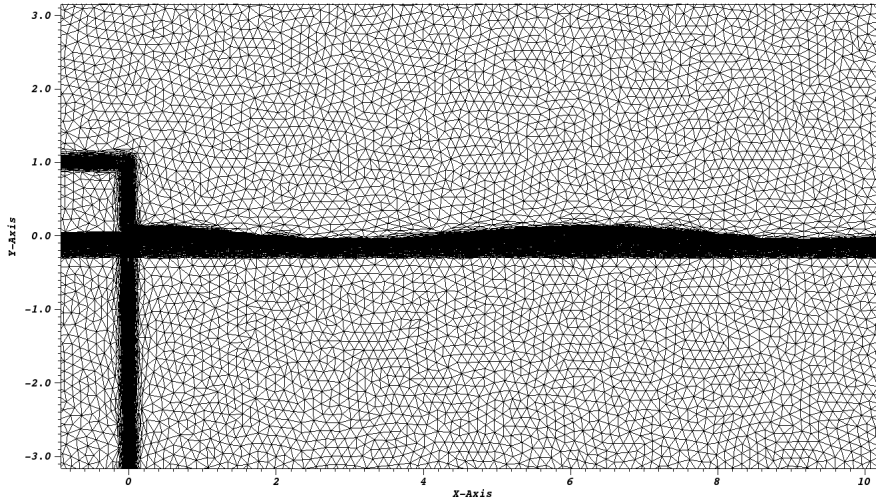


Figure 4.31: Anisotropic mesh for the simulation of a monochromatic wave field of amplitude $0.1m$, of frequency $0.5s^{-1}$ and of wavelength $6m$ at $t = 20s$, refined under the free-surface. Zoom on the free-surface. Scales in meters.

crests. The wave fields obtained with different configurations are presented in the following.

The first configuration tested consisted in the determination of r_{Thr} , threshold ratio from Eq. (4.5). A monochromatic wave field of amplitude $0.1m$, of frequency $0.5s^{-1}$ and of wavelength $6m$ is prescribed. The different wave fields presented in Fig. 4.32 compare the results of ICI-tech simulations against output data from HOS-NWT. A test case was run without the additional refinement, while the other test cases respected $r_{Thr} = 3$ and $r_{Thr} = 5$. The free-surfaces plotted show very similar data for $r_{Thr} = 3$ and $r_{Thr} = 5$, with reduced damping compared to the last configuration. For a matter of computational efficiency, these results have been obtained with a time step $\Delta t = 10^{-2}s$, which provided damping even for thin, isotropic meshes. The improvement observed in the propagation of the wave fields advocates the use of refinement under the free-surface. Still, the sensitivity of the results to the variations of r_{Thr} would need to be investigated. In the following, $r_{Thr} = 3$ is conserved, in more accurate studies.

Another influence of the refinement under the free-surface is to concentrate a high number of computational points around the free-surface. From this observation, the influence of the mesh adaptation is expected to be slightly less important than for a classical anisotropic mesh generation. Once again, a monochromatic wave field of amplitude $0.1m$, of frequency $0.5s^{-1}$ and of wavelength $6m$ is prescribed. The dependency of the results towards the adaptation frequency is tested for $\varepsilon = 10^{-2}m$ and $\Delta t = 10^{-3}s$. Both free-surfaces are found to have the same period. In terms of amplitude, the highest adaptation frequency, with mesh updates every $7 \cdot 10^{-3}s$, underperforms compared to the other test cases, considering, respectively, adaptation every $3.5 \cdot 10^{-2}s$ and $7 \cdot 10^{-2}s$. The last two cases provide very similar results. As only difference, the free-surface adapted every $7 \cdot 10^{-2}s$ is slightly less damped, but seems to encounter more mass losses. Despite the increasing precision of the results, the results obtained with the anisotropic adaptation are still a bit damped, even if the mesh is refined under the free-surface. A 8.5%-error is still found in terms of amplitude, from a rapid evaluation of the crest located next to $x = 25m$.

Further studies are required to increase again the precision of the wave fields propagated. The refinement realized under the free-surface follows the paradigm proposed in a previous paragraph. The size of cells along the x -direction is reduced thanks to

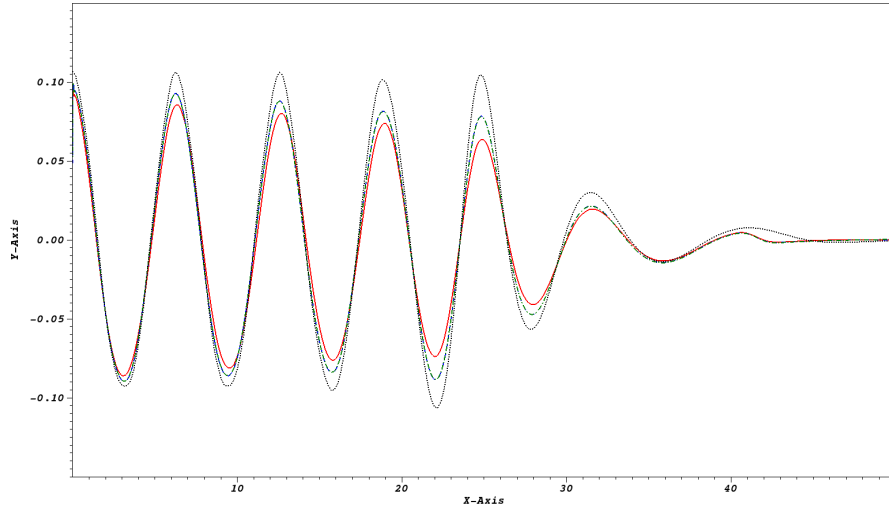


Figure 4.32: Monochromatic wave field of amplitude $0.1m$, of frequency $0.5s^{-1}$ and of wavelength $6m$ at $t = 20s$ simulated with ICI-tech using an anisotropic mesh refined under the free-surface. Simulations realized with $\varepsilon = 10^{-2}m$, $\Delta t = 10^{-2}s$, and adaptation every $7 \cdot 10^{-2}s$. Black, dotted: Wave field from HOS-NWT. Red, plain: No refinement. Blue, dashed: $r_{Thr} = 3$. Green, dash-dotted: $r_{Thr} = 5$. Scales in meters.

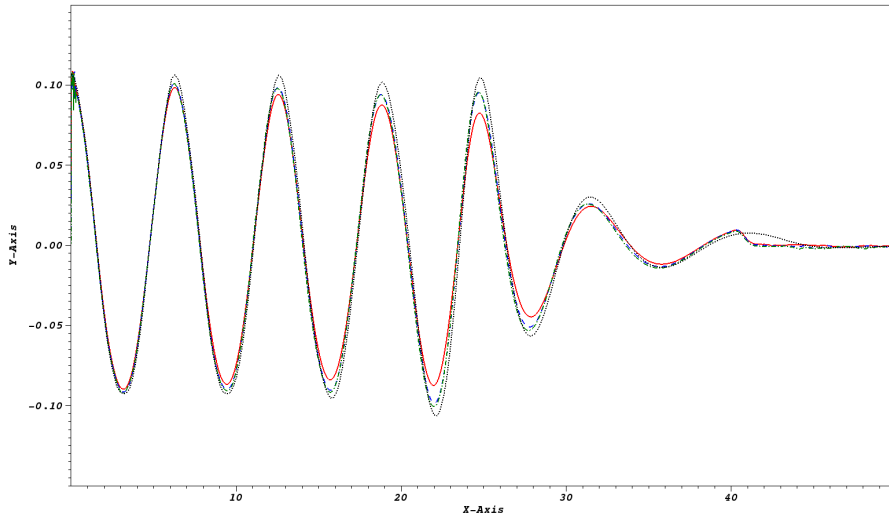


Figure 4.33: Monochromatic wave field of amplitude $0.1m$, of frequency $0.5s^{-1}$ and of wavelength $6m$ at $t = 20s$ simulated with ICI-tech using an anisotropic mesh refined under the free-surface ($r_{Thr}=3$). Simulations realized with $\varepsilon = 10^{-2}m$, $\Delta t = 10^{-3}s$, and different adaptation frequencies. Black, dotted: Wave field from HOS-NWT. Red, plain: Adaptation every $7 \cdot 10^{-3}s$. Blue, dashed: Adaptation every $3.5 \cdot 10^{-2}s$. Green, dash-dotted: Adaptation every $7 \cdot 10^{-2}s$. Scales in meters.

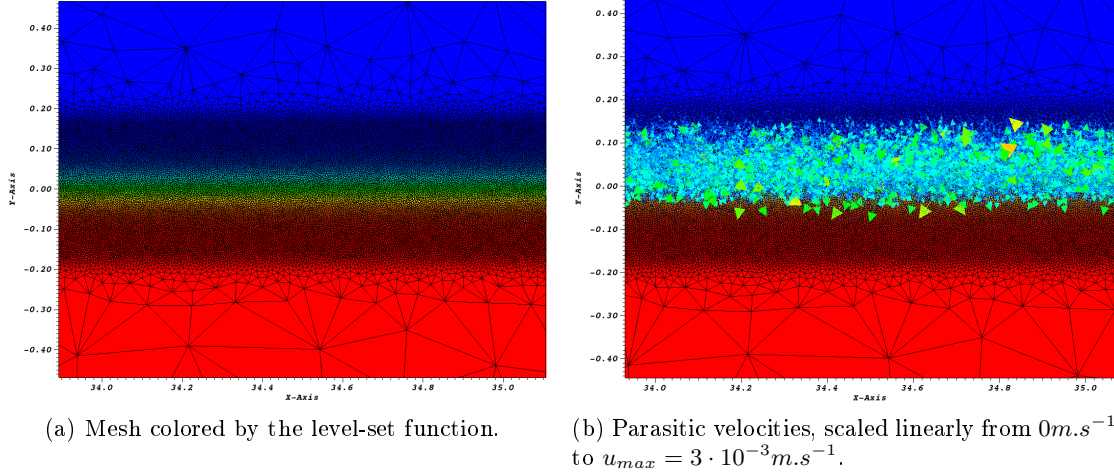


Figure 4.34: No-flow test with a constant isotropic mesh ($h_{min} = 5 \cdot 10^{-3}m$) after a single time increment, for $\Delta t = 10^{-3}s$ and $\varepsilon = 5 \cdot 10^{-2}m$. Scales in meters.

the parameter h_x introduced, while a discretization of the areas under the free-surface is performed. The improvement of the results is notable, but optimization could still be realized. Several options can be discussed. The first one considers the level of refinement prescribed in the studies realized. A unique discretization was tested, with 100 cells to represent a wavelength and 20 cells between crest and trough. Alternative distributions of cells in the refined areas would be necessary to get a good comprehension of the influence of the mesh under the free-surface.

4.2.3 No-flow tests

In a NWT, the regions located far from the wave generators follow a hydrostatic behavior at the beginning of the simulations. The free-surface is supposed to be at rest, but the simulation presented in this Sec. highlighted the creation of parasitic velocities. They appear in the areas where a density drop is observed, especially with isotropic meshes. They can be identified on Fig. 4.21 for $x < 10m$ and $x > 35m$, where the free-surface is disturbed. The presence of parasitic velocities is to avoid in the context of NWTs, as the propagation of waves is directly impacted by the perturbations. These velocities are due to the iterative solver used, which minimizes a residual. The minimization is stopped when a convergence condition is met. If the rate of convergence is not sufficient, perturbations of important magnitude may be generated.

The parasitic velocities generated by isotropic and anisotropic meshes are compared for an identical test case. A water basin of dimensions $60mm$ is defined, with a water depth of $5m$. An initially still free-surface is considered, and no external perturbations are prescribed. The simulations are run with $\varepsilon = 5 \cdot 10^{-2}m$, $h_{min} = 5 \cdot 10^{-3}m$ and $\Delta t = 10^{-3}s$. The adaptation is forced, to reach the minimal mesh size around the free-surface. Thus, the only difference between those test cases consist in the meshing, which is thinner along the horizontal direction in the isotropic case. The evolution of the velocities generated after a single time step are examined. The results obtained present an homogeneous distribution of the parasitic velocities along the free-surface. Zooms on the air/water interface present views of those velocities drawn for isotropic meshing, see Figs. 4.34, and anisotropic meshing, in Fig. 4.35. The scale of the velocities presented in those Figs. is linear from 0 to u_{max} . In the isotropic case, the magnitude of the highest velocities is $u_{max} = 3 \cdot 10^{-3}m.s^{-1}$, while $u_{max} = 7 \cdot 10^{-4}m.s^{-1}$ in the anisotropic case.

A first observation concerns the repartition of the velocities, that are all located in

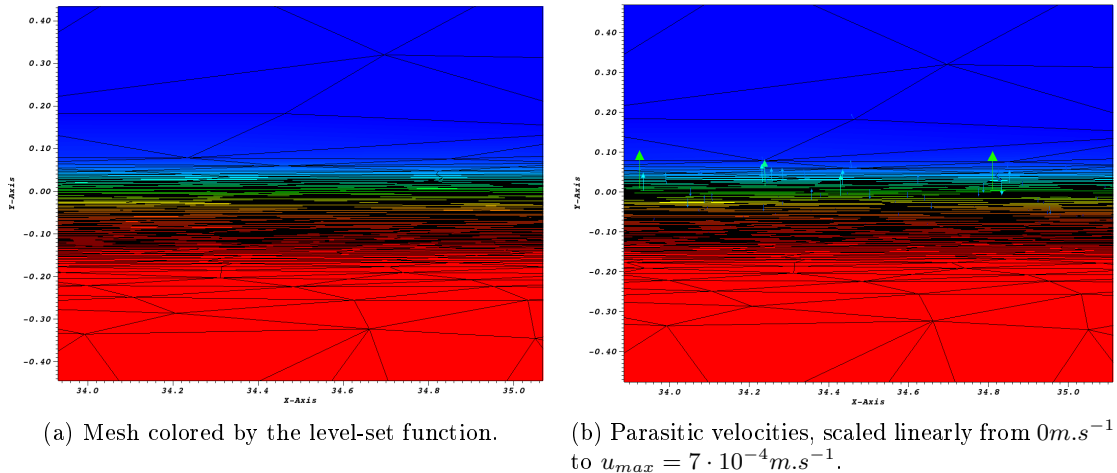


Figure 4.35: No-flow test with a constant anisotropic mesh ($h_{min} = 5 \cdot 10^{-3}m$) after a single time increment, for $\Delta t = 10^{-3}s$ and $\varepsilon = 5 \cdot 10^{-2}m$. Scales in meters.

the transition area of the level-set, around the free-surface. More precisely, the parasitic velocities appear when a density drop is observed. With the isotropic and anisotropic test cases, the difference between both u_{max} is of an order of magnitude. This is due to the meshing, much coarser in the x-direction in the anisotropic test case. This induces numerical dissipation, which tends to reduce the parasitic velocities observed. Moreover, while in the isotropic test case the velocities have no preferential direction, the anisotropic application generates a vertical bias. The reason is again the meshing in the x-direction.

Several studies have been realized reduce the impact of those velocities on the simulations using anisotropic meshes. In particular, their evolution with the time step Δt or the smallest mesh sizes h_{min} have been studied, and are plotted in Figs. 4.36. It is important to note that, due to the anisotropic characteristic of the mesh, the evolution of h_{min} involves variations of the mesh size following the level-set, and consequently the density drop. As soon as the simulations times are reduced, the evolution of h_{min} acts exclusively on the mesh size in the vertical direction. The results obtained show a small dependency in Δt , while a first-order convergence is measured in h_{min} . This means that, to divide u_{max} by 10, h_{min} must be divided by 10 too. Consequently, the reduction of the parasitic velocities needs to be realized through the use of a thinner mesh, at least in the vertical direction. This behavior is observed independently of the level-set width ε . The pressure drop measured on each mesh cell seems to rule the apparition of parasitic velocities: the highest the drop, the largest the perturbations. The convergence rate can be put in perspective with the orders of convergence of the schemes used for the NS simulations, of the first order both in time and space. Increasing the order of those schemes would be a way to reduce them without diminishing h_{min} too much.

The resolution along the horizontal direction has an impact on the results, as highlighted by the differences between isotropic and anisotropic simulations. The wave fields presented, e.g., in Fig. 4.21, suggest that the parasitic velocities mostly have an influence on solutions in areas where the waves are small and recently appeared. This behavior is observed both for anisotropic or isotropic simulations. As anisotropic meshes are refined horizontally when waves of strong curvature are found, the influence of the h_{min} on the horizontal refinement will be limited. The areas concerned by parasitic velocities will mostly be meshed as in Fig. 4.35 rather than Fig. 4.34. A different situation could be found with enhanced meshing under the free-surface, depending on the waves simulated and criteria considered. These situations will need to be regarded in a case-by-case basis.

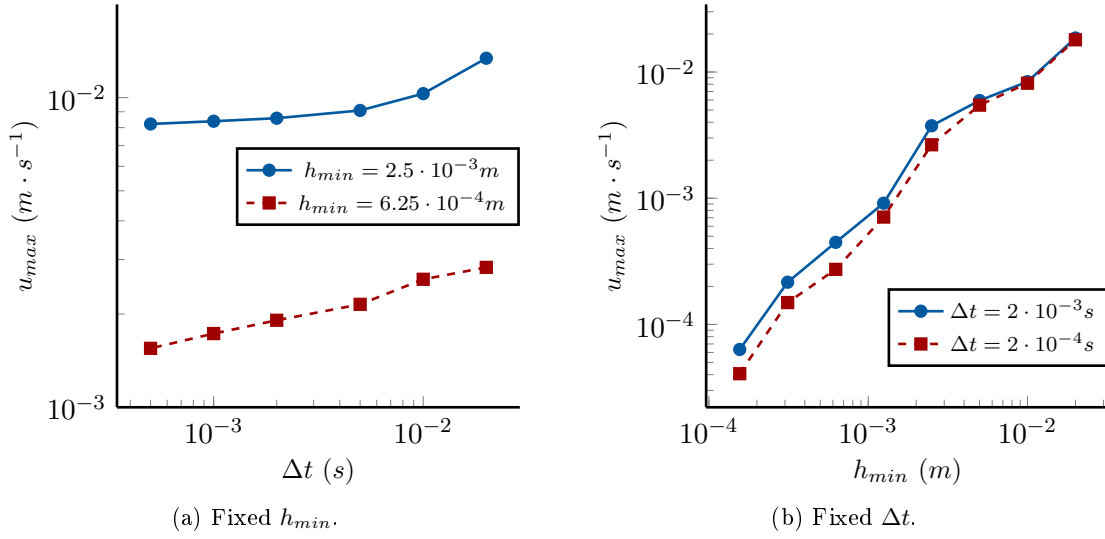


Figure 4.36: Residual velocities obtained using anisotropic meshes at $t = 2 \cdot 10^{-2} s$ for various Δt and h_{min} , with $\varepsilon = 5 \cdot 10^{-2} m$.

Another approach consists in increasing the rate of convergence achieved during the resolution of the problem, which is formulated in velocity and pressure. However, in a hydrostatic situation, the converged pressure is in the range of $10^5 Pa$ for a depth about $1m$, while the velocities are supposed to converge towards $0m \cdot s^{-1}$. The risk is for the solver to focus on the convergence on the pressure, and to overlook the errors on the velocity. Consequently, the preconditioning of the problem needs to be particularly efficient to obtain small residual velocities. This study will be important for the validation of a new solver dedicated to the simulation of free-surface flows.

4.3 Verification and validation of the FSI and application to buoyancy

This Sec. proposes a validation of the different utilities implemented in ICI-tech for the FSI of FOWTs and presented in Sec. 3.3. This study has been realized within a master thesis integrated into the scope of the PhD. Two main topics need to be regarded: the evaluation of forces applied on structures, and the driven motion of buoys. In a first time, a verification of the forces applied on a static object are measured, and convergence studies are presented in Subsec. 4.3.1. Free motions of floating solids intersecting a free surface are then studied in Subsec. 4.3.2. Two-dimensional test cases are presented, where vertical and rotational motions are considered separately.

4.3.1 Verification of the buoyancy force applied on a cylinder

The first case tested intends to validation the computation of the force presented in Subsec. 3.3.1, realized within a master's thesis. A cylinder of density $\rho = 500kg \cdot m^{-3}$ is completely submerged under water, defined as a fluid of density $\rho = 1000kg \cdot m^{-3}$ and of viscosity $\eta = 0.001kg \cdot m^{-1} \cdot s^{-1}$. Its diameter is $D = 0.1524m$ and the computational domain is of size $20D$, the cylinder being located at its center. The exact solution is given by the Archimedes' principle: the resultant buoyant force F_{an} is equal to the weight of the displaced fluid. The results obtained are compared to those from Borazjani et al. (2008) and Calderer (2015), which used a LES formulation of the Navier-Stokes equations and

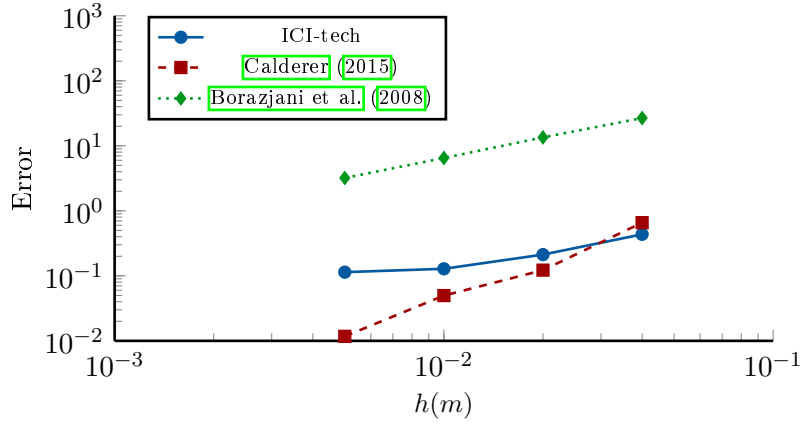


Figure 4.37: Static buoyant cylinder, convergence of the error of the computed buoyant force with the interface thickness, for the fully immersed case. Comparison with results from Calderer (2015) and Borazjani et al. (2008).

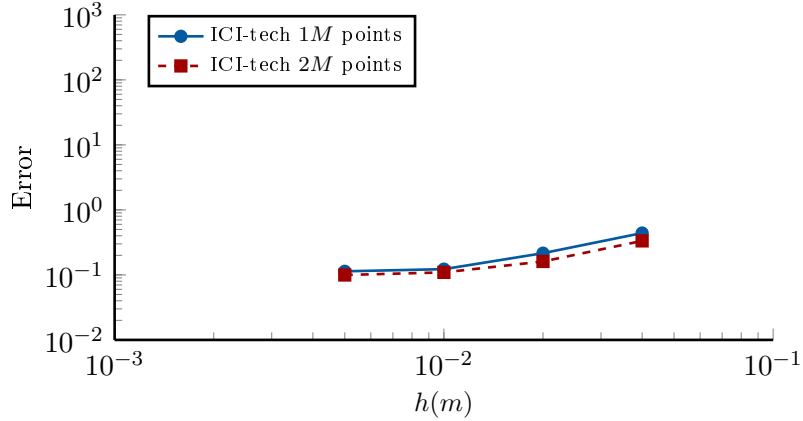


Figure 4.38: Static buoyant cylinder, convergence of the error of the computed buoyant force with the number of mesh nodes, for the fully immersed case.

a Finite Volume approach with immersed boundaries. The error E , written in Eq. (4.8), is measured.

$$E = \frac{|F_{fluid} - F_{an}|}{F_{an}} \quad (4.8)$$

A log-log plot showing the convergence of the error is printed in Fig. 4.37. Different adapted meshes of minimal mesh size h_{min} being $D/25$, $D/50$, $D/100$, and $D/200$ are considered. As explained in Chap. 2, the width of the level-set representing the immersed object respects $h_{min} = \varepsilon/10$.

The standard method from Borazjani et al. (2008) overpredicts the buoyancy force on the cylinder, and has a first-order rate of convergence. The PPBC method detailed in Calderer (2015) provides errors lower of one order of magnitude than the standard method, also with a better convergence rate, of 1.8. The approach followed in ICI-tech presents a good order of magnitude and a similar convergence rate than the standard method.

The number of mesh nodes has been increased to 200K. This avoids a limitation of the convergence rate due to the number of computational points. A similar convergence is highlighted by Fig. 4.38. As the order of the discretizations over time and space are

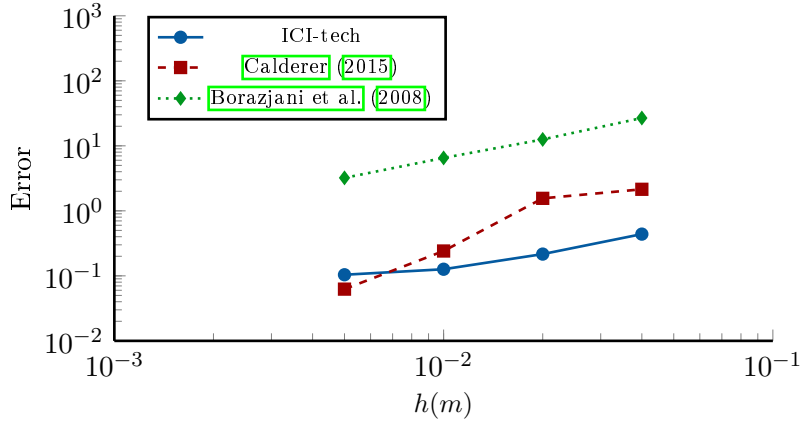


Figure 4.39: Static buoyant cylinder: convergence of the error of the computed buoyant force with the interface thickness, for the half immersed case. Comparison with results from Calderer (2015) and Borazjani et al. (2008).

both of 1, the rate of convergence obtained is satisfying.

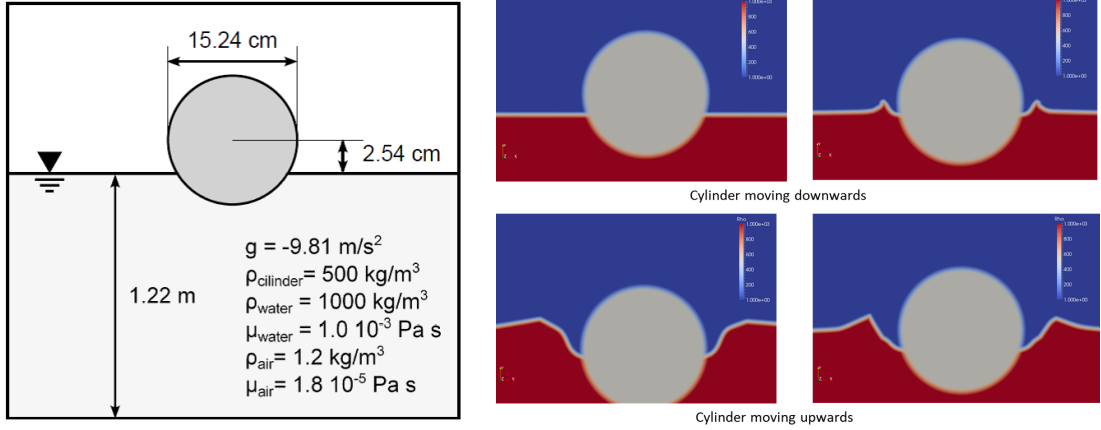
A similar study is proposed for the same cylinder, this time half immersed. An important modification of the test case is required, as a free-surface, defining the air/water interface, has to be represented. This three-phase is more challenging, as important drops in η and ρ are observed between air and water. In particular, the density variations were shown to generate parasitic velocities in Subsec. 4.2.3. The meshing of the free-surface also induces a higher computational effort. A triple point appears at the meeting point of the three phases, which also requires a particular treatment. Still, the error between the exact buoyancy force, given by Archimedes' principle, and the numerical evaluation can be computed as detailed in Eq. (4.8).

The convergence of the error is presented in Fig. 4.39 with different adapted meshes of $100K$ nodes, of minimal mesh size h being $D/25$, $D/50$, $D/100$, and $D/200$. A good order of magnitude and the same convergence rate are achieved, even if the mesh adaptation densifies the mesh around the free-surface. The same behavior is observed for meshes of $200K$ points, which shows that $100K$ nodes should be largely enough to test these types of cases. The comparison with the results presented by Borazjani et al. (2008) and Calderer (2015) provide a similar tendency compared to the fully-immersed case. The rate of convergence achieved with ICI-tech compares well against the standard method, while the PPBC method provides a higher convergence rate.

4.3.2 Validation of the FSI solver for driven motions of floating structures

This Subsec. intends to apply the FSI framework developed in Subsec. 3.3.3 to the case of buoyancy required in the context of FOWTs. This particular case requires the simulation of three-phases: the FOWT, the water, and the air. Two different level-set functions are required, one representing the FOWT, the second one modeling the free-surface. A triple point appears at each intersection of the three-phase and a higher complexity is induced. No particular treatment of these points has been realized in the FSI framework.

Two benchmarks are considered in the following, both focusing on driven motion. The free, vertical decay of a cylinder is studied, from an initial non-equilibrium position to an equilibrium one. The free decay of a rectangular solid in rotation, up to equilibrium, is also simulated. The results obtained using the methodology developed are compared with



(a) Free decay of a cylinder: schematic description of the cylinder configuration given in Calderer (2015) and studied experimentally in Ito (1977).

(b) Free decay of a cylinder: snapshots of the cylinder position and the water and air interface surface, for a mesh with $N = 100000$ nodes, $\varepsilon = 0.005D$ and $\Delta t = 5 \cdot 10^{-4} \text{ s}$.

Figure 4.40: Free decay of a cylinder released over a free-surface. Description of the test case and illustration of the results from ICI-tech.

other authors, whose results have been gathered in Calderer (2015). An experimental study from Ito (1977) is considered, along with two numerical methods. The first one is a standard immersed-boundaries approach, presented by Borazjani et al. (2008), which was extended in Calderer (2015) through corrections of the force evaluation around the solid. This Subsec. intends to demonstrate the capabilities of the FSI modified level-set formulation to interact with a free-surface.

Free decay of a cylinder

The test case considered here uses the cylinder defined in Subsec. 4.3.1, which is released from an initial position located over its equilibrium. To recall, the cylinder of density $\rho = 500 \text{ kg.m}^{-3}$ is horizontal and piercing the free-surface. Its diameter is again $D = 0.1524 \text{ m}$ and the computational domain is still of size $20D$, with no-slip conditions imposed at the walls. The test case is illustrated in Fig. 4.40a. The cylinder presents a single degree of freedom (vertical) and is going to oscillate until attaining its equilibrium position (counterbalance of Archimedes' force), resulting in no-velocity. Furthermore, the maximum Reynolds number attained, based on the cylinder diameter, the water properties, and the maximum velocity, is $30K$.

Two different interface thicknesses have been considered in our computations, corresponding to $D/200$ and $D/50$, with $100K$ -points meshes. The time step is computed from the CFL condition and thus different for the two cases. First trials indicated that the lowest value should be around $\Delta t = 5 \cdot 10^{-4} \text{ s}$. Calderer (2015) tested the same case with different mesh sizes on uniform and non-uniform meshes, the latter still being uniform around the solid object and having a mesh size that varies far from it.

Snapshots of the cylinder position are presented in Fig. 4.40b. The deformation of the free-surface is important, generating radiation waves. A close view of the triple point shows that there is a remaining air layer at the junction. The computed normalized position of the cylinder, determined by Eq. 4.9, is used to compare the results obtained by Calderer (2015) and Ito (1977). This Eq. considers y_{CG} , the vertical position of the center of gravity of the cylinder, tracked in time. Parameters defining the test case are used too, $d = 1.22 \text{ m}$ the water depth and $h_0 = 0.0254 \text{ m}$ initial position of the cylinder

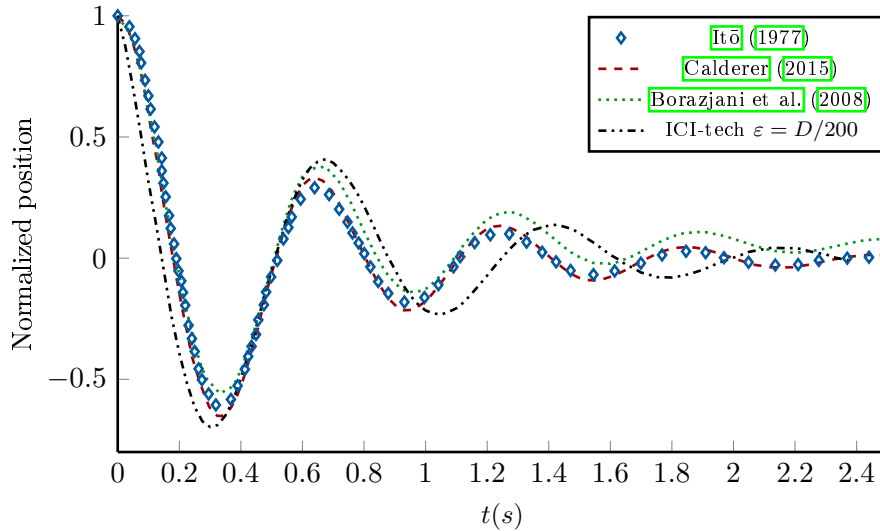


Figure 4.41: Free decay of a cylinder: normalized position of the cylinder, for an interface thickness of $D/200$, a mesh with $100K$ nodes, and $\Delta t = 5 \cdot 10^{-4} s$.

above the free-surface.

$$y^*(t) = \frac{y_{CG}(t) - d}{h_0} \quad (4.9)$$

The results from ICI-tech are compared in Fig. 4.41 with the experimental ones of Itō (1977), which have also been used to validate the immersed boundary methods of Calderer (2015). The frequency of the oscillation observed does not match the experimental one. Interestingly, the standard method from Borazjani et al. (2008) encounters the same problem, while the PPCB approach from Calderer (2015) performs better. The position of the cylinder oscillates with a higher amplitude than expected, showing that the radiation damping is underpredicted in this situation.

The damping observed on ICI-tech's simulations for interface thickness of $D/200$ can be compared to the one from Itō (1977) using the quantification of pseudo-oscillations. The first three peak positions of the cylinder are selected, as a strong damping limits the precision of the identification of further peaks. The amplitude of the peaks is plotted in Fig. 4.42 as a function of the time. The differences observed for the pseudo-period are notable, reaching 20% for the third peak. Even if the precision of the measures induce uncertainties, the pseudo-period measured is largely outside the error bounds. Based on the three first peaks identified, the average pseudo-period computed with ICI-tech is $1.10s$, while the experimental one is $0.92s$. This characterizes a slower evolution of the cylinder. Similarly, important differences are measured for the amplitude. The decays follow an exponential law $\exp(-\gamma t)$. An identification of the parameter γ for the decays returned $-1.41s^{-1}$ for ICI-tech's simulations and $-1.87s^{-1}$ for the experimental study. The smaller coefficient γ computed with ICI-tech characterizes a slower decay.

Further analysis can be conducted with a focus on the pseudo periods. In the context of a free decay, the expression of the pseudo-period, written in Eq. (4.10), depends on the mass of the object m , on the added-mass m_{add} , and on the hydrostatic stiffness K_H .

$$T = 2\pi \sqrt{\frac{m + m_{add}}{K_H}} \quad (4.10)$$

As the verification of K_H was realized in Subsec. 4.3.1 and the mass of the system is correctly specified, the differences arising in the decay of the cylinder certainly come

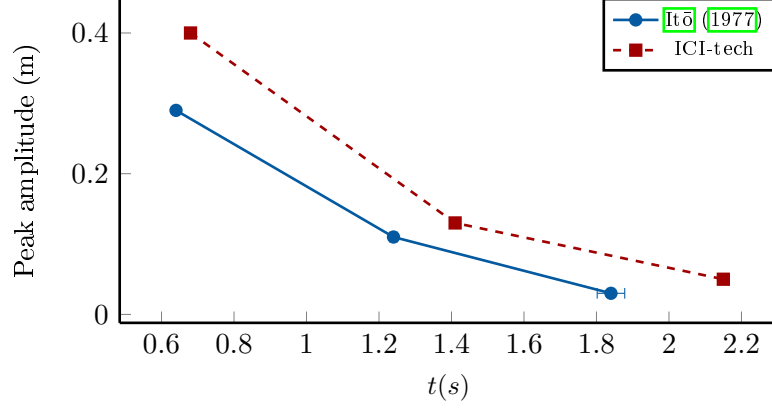


Figure 4.42: Study of the peak positions of the cylinder during the pseudo-oscillatory decay, as a function of the pseudo-period.

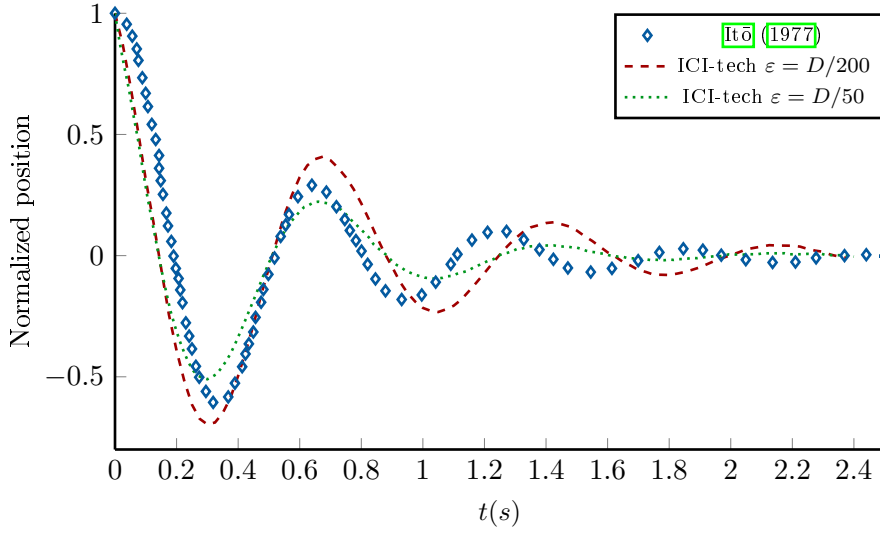


Figure 4.43: Free decay of a cylinder: normalized position of the cylinder, for an interface thickness of $D/200$ and $D/50$, a mesh with $100K$ nodes and $\Delta t = 5 \cdot 10^{-4}s$.

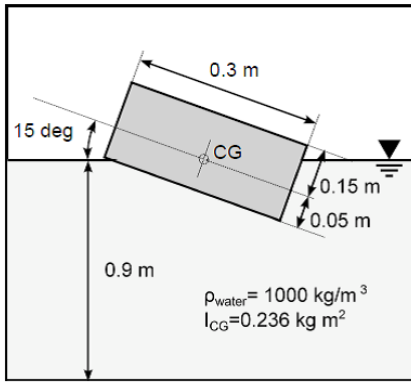
from the added-mass term. Further investigations on the FSI framework will be required to get a better comprehension of these decay errors.

Another important study concerns the sensitivity of the decay towards ε , i.e., the variations found with coarser or thinner meshes. Fig. 4.43 compares the results obtained for two different interface thicknesses, showing the degradation of the result with an increasing ε , in terms of amplitude. The frequency remains similar for both ε .

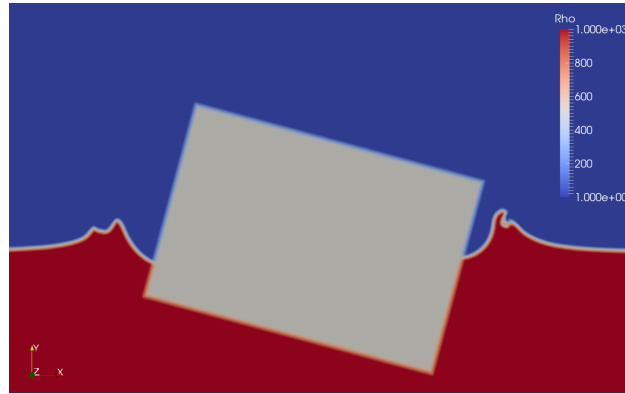
The average error can be computed from Eq. 4.11, with y_{exp} the experimental position of the vertical center of gravity, given in Itô (1977), and n is the total number of time steps.

$$E = \frac{1}{n} \sqrt{\sum_{i=1}^n (y_{CG}(t_i) - y_{exp}(t_i))^2} \quad (4.11)$$

The error obtained is $1.12 \cdot 10^{-3}m$ for $\varepsilon = D/200$ and $1.34 \cdot 10^{-2}m$ for $\varepsilon = D/50$. As a point of comparison, Calderer (2015) has obtained, with other immersed boundary methods, between $E = 6.13 \cdot 10^{-4}m$ and $E = 1.28 \cdot 10^{-3}m$ for interface thicknesses of



(a) Free decay of a rectangle bar: schematic description of the rectangular bar configuration given in Calderer (2015) and studied experimentally in Jung et al. (2006).



(b) Free decay of a rectangular bar: snapshot of the bar position and the water and air interface surface, for a mesh with 100K nodes, $\epsilon = D/200$ and $\Delta t = 5 \cdot 10^{-4} s$. Perturbation of the system is high and has required several trials of the numerical parameters to achieve convergent results.

Figure 4.44: Free decay of a rectangle bar through a free-surface. Description of the test case and illustration of the results from ICI-tech.

the order of $D/200$.

In conclusion, the differences between the simulations realized in ICI-tech and experiments show that the methodology has given promising, preliminary results. Correct orders of magnitude are met, with similar tendencies compared to the works from Borazjani et al. (2008), which share the same order of convergence in the error studies. Further investigation will be necessary to match more accurately the experimental results.

Free decay of a rectangular bar in rotation

To validate the rotational motion of the solid object, a second test case is used. It is described and simulated in Calderer (2015), and experimentally studied in Jung et al. (2006). A rectangular bar of length of $0.3m$, of height of $H = 0.2m$ and of depth of $1m$ is considered. The large width of the bar leads to the assumption that studying a 2D set-up is sufficient to evaluate the accuracy. The bar is placed in a $35m$ -long computational domain, with a water depth of $0.9m$. The center of gravity of the bar coincides with the position of the free-surface and the solid is, at the beginning, inclined with an angle of 15° . Boundary conditions imposed are slip conditions ($v \cdot n = 0$) on the borders of the computational domain. The different geometrical parameters of the test case are illustrated in Fig. 4.44a.

As for the cylinder case, two different interface thicknesses are considered, corresponding to $H/200$ and $H/50$, being H the height of the channel ($H = 0.2m$). Adapted meshes of 100K points are used, and the time step is computed from the CFL condition for both cases.

Stability was hard to establish in this test case, due to the complexity of the interface near the object and to its motion, as presented in Fig. 4.44b. Strong deformations of the free-surface are observed, while the rectangular shape is conserved over the simulation thanks to the mesh adaptation procedure.

Results provided by ICI-tech are compared with the experimental ones of Jung et al. (2006) in Fig. 4.45, by plotting the evolution in time of the angle of inclination. The

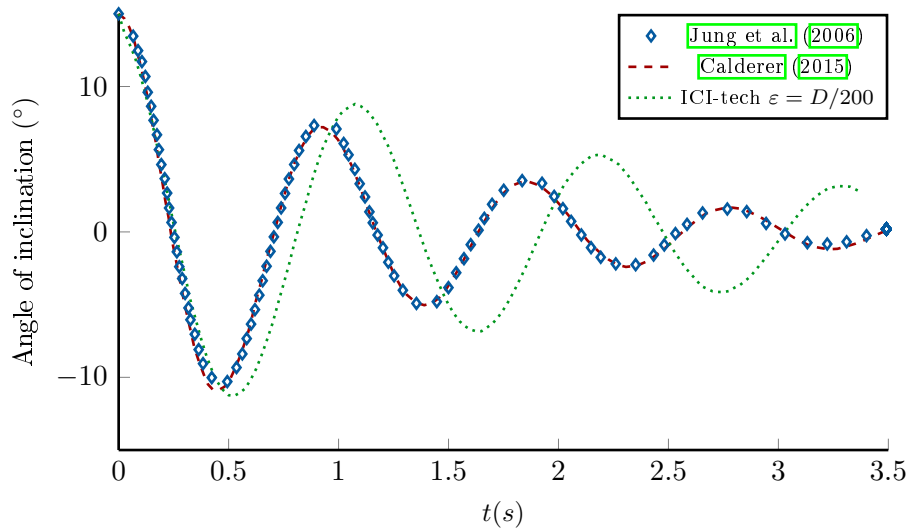


Figure 4.45: Free decay of a rectangular bar: numerical angle of inclination of the bar obtained with our code; experimental data of Jung et al. (2006) and numerical results from Calderer (2015). Mesh with $N = 100K$ nodes, $\epsilon = D/200$ and $\Delta t = 5 \cdot 10^{-4}s$.

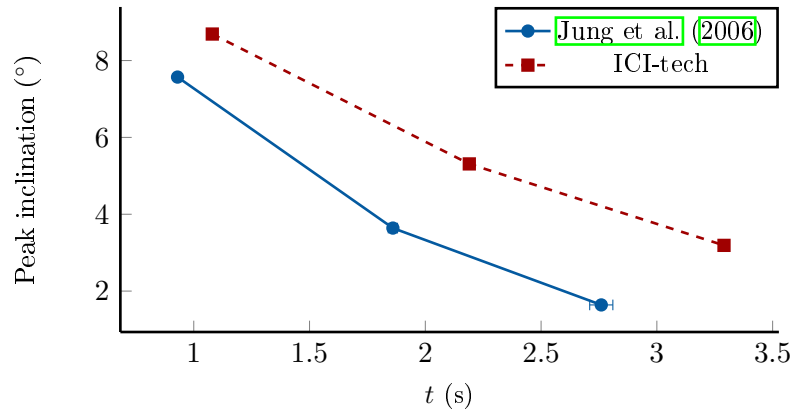


Figure 4.46: Study of the peak positions of the rectangle during the pseudo-oscillatory decay, as a function of the pseudo-period.

method predicts qualitatively the evolution of the angle of inclination, but still does not have the same amplitude. Numerical results of Calderer (2015) are also presented, showing that he gets better accuracy by introducing some damping in the equation of motion of the solid object. Important errors are observed in terms of frequency too.

As for the case of the cylinder, the pseudo-oscillations observed during the decay with ICI-tech simulations can be compared against experimentations, this time from Jung et al. (2006). The amplitudes of rotation are plotted as a function of the pseudo-period in Fig. 4.46. Similarly to the cylinder vertical decay, the rotation of the rectangle simulated with ICI-tech return longer pseudo-periods and higher amplitudes. As for the previous test case, an exponential law can be used to fit the decays, of the form $15\exp(-\gamma t)$. The coefficient γ of the simulations from ICI-tech is $-0.47s^{-1}$, which the experimental one is $-0.79s^{-1}$. As for the cylinder test case, a slower decay is identified, which can probably be related to the added-mass effects. The average pseudo-period from ICI-tech's simulations is $0.72s$ while Jung et al. (2006) identified a $0.61s$ one, which characterizes a slower evolution of the decay.

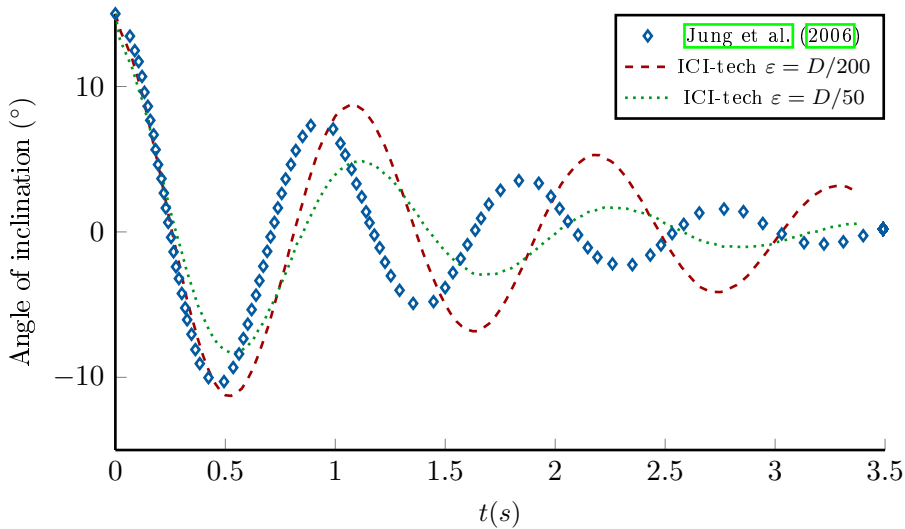


Figure 4.47: Free decay of a rectangular bar: influence of the interface thickness and comparison with the experimental data from Jung et al. (2006).

Once again, a comparison between two different ε done in Fig. 4.47 highlights an important variation in terms of amplitude, as observed for the free-decay of the cylinder previously studied. A higher ε leads to a damped motion. For this test case, a phase shift is also observed when ε is varied. Further studies would be needed to get a better comprehension of the phenomena generating these errors on the period. The pseudo-periods are increased with a smaller ε , which widens the gap between experimental and ICI-tech's results. This observation tends to show that decreasing ε optimizes the quality of the simulations. Further studies with smaller ε would be required to check this supposition.

However, $\varepsilon = D/200$ already generates a considerable computational effort, as a minimal mesh size of $D/2000$ needs to be achieved around the immersed object. A smaller mesh size will probably improve the results. At the same time, the simulation of 3D, operational FOWT test cases with such a precision is unachievable. The developments of other utilities or resolution methods will be required for an accurate simulation of floating structures.

4.4 Conclusion on the numerical results

This Chap. presented the simulations and results realized towards the development of a simulator of FOWTs. Two different paradigms were followed: the validation of the numerical framework was realized, besides of developments linked to the definition of a demonstrator. The validation of the flows was split between aerodynamics, in Sec. 4.1, and hydrodynamics, in Sec. 4.2. A validation of the FSI was proposed in Sec. 4.3.

The aerodynamic studies focused on flows around fixed geometries, represented using implicit boundary methods. The aerodynamic coefficients measured around the bodies were compared with numerical studies and experimental data, while the wake was not regarded. The results obtained during the aerodynamic validation deteriorated as the Re number of the flows considered rose, with errors arising. The consideration of first-order finite elements can be linked to the low precision achieved with rising Re . The linear interpolation of the phase in each finite element requires, in order to represent phenomena, a very thin meshing. In the context of rising Re , the characteristic length of turbulent structures decreases, which requires corresponding reductions in local mesh

sizes. Similarly, the first-order time discretization implies the use of smaller time steps.

The validation of the hydrodynamics focused on the generation and propagation of waves. Monochromatic wave fields were studied, mainly in 2D, and generated using two different methods. The recourse to thin, isotropic meshes tended to return wave fields approaching the target solution. Using the anisotropic, automatic meshing as presented in Sec. 2.2, the propagation of waves highlighted the presence of damping. Several studies were done to reduce this numerical dissipation, which could be reduced, but not eradicated. The influence of the mesh adaptation procedure on the damping produced was identified, but not quantified. Aside from the amplitude of the wave fields simulated, the frequencies observed were generally found satisfying. Similarly to the conclusions drawn for the aerodynamic validation, an extension of the finite element method to higher orders of interpolation has the potential to limit the errors produced.

The Sec. focusing on the FSI framework intended to validate the tools implemented into ICI-tech for the simulation of FOWTs. A first step considered the evaluation of forces, from two cases computing the buoyancy force on an immersed cylinder. Free-decay, driven test cases were then studied, to validate the FSI framework. The test cases simulated were representative of a FOWT, as a three-phase flow was considered, with a free-surface and triple points. The decay in translation provided results with some bias, that could be put in perspective with methods of similar precision. On the contrary, the test case focusing on rotational decay was subject to failures, and generated important errors, both in terms of amplitude and period. Still, the added-mass effects looked overlooked. A poor representation of these hydrodynamic contributions tended to limit the accuracy achieved with ICI-tech on buoyant test cases.

Chapter 5

Conclusion and perspectives

Contents

5.1 Conclusion	129
5.1.1 Representation of the geometries	129
5.1.2 Solver verification and validation	129
5.1.3 Influence of the anisotropic meshing on the propagation of waves	130
5.2 First developments towards a demonstrator of FOWTs	130
5.2.1 Rotation of rigid WT blades	130
5.2.2 Preliminary works on IEAWind Task 29	131
5.2.3 Simplified FOWT demonstrator	131
5.2.4 Towards a representation of real sea states	132
5.3 Perspectives	135
5.3.1 Higher-order FEM	135
5.3.2 Mesh adaptation for the propagation of waves	136
5.3.3 Mid- to long-term perspectives for an accurate FOWT simulator	137

In this thesis, the steps followed towards the adaptation of the software platform ICI-tech to the simulation of FOWTs have been presented. After an introduction on the simulation of FOWTs in the literature, the numerical methods for the representation and the simulation implemented in ICI-tech have been introduced. A paradigm for the resolution of the incompressible NS equations in a VMS formulation is presented, which takes advantage of level-set functions and P1-FEM. The numerical results have finally been presented in Chap. 4. The different conclusions from the thesis are presented in Sec. 5.1. A presentation of preliminary developments towards a simulator of FOWTs is presented in Sec. 5.2. Finally, the perspectives obtained from those results are presented in Sec. 5.3.

5.1 Conclusion

The numerical simulations realized within the scope of this thesis can be split between the representation of the geometries and test cases aiming at developing a FOWT demonstrator. The observations made from the results presented in Chaps. 2 and 4 highlighted several facts, which are detailed in the following.

5.1.1 Representation of the geometries

The methodology for the resolution of multiphase flows in ICI-tech uses a unique mesh. The representation of the geometries is a self-standing software unit, which uses level-set functions for the reconstruction of the immersed geometries. The mesh immersion process has been optimized using an octree, as detailed in Chap. 2. The computational gains have been quantified thanks to speed-up tests. The different simulations presented in this Chap. highlight the potential of ICI-tech to reconstruct one or several FOWTs. The level of accuracy of the representation can be adjusted, which is important given the geometry of WT blades. This tool looks optimized and suited for the representation of FOWT. The implementation oriented towards an optimization of the computational costs allows its use for the simulation of FOWTs. Its integration into ICI-tech's massively parallel framework allows deployments on clusters. All those characteristics pave the way for an accurate representation of one or several FOWTs, on large computational domains.

5.1.2 Solver verification and validation

The validation of the solver realized in Chap. 4 intends to simulate FOWTs as accurately as possible. To that extent, the aerodynamics and hydrodynamics have been considered. Different behaviors have been highlighted. The evaluation of the aerodynamic coefficients around several profiles provided results of precision decreasing with the rise of the Reynolds number. The VMS solver currently used is not suited for high- Re flows. Approximations made in the development of the VMS subscales, which are considered quasi-static and neglect the convective velocity, probably have the most influence on the imprecisions observed.

On the contrary, preliminary hydrodynamic studies considering an isotropic mesh showed a propagation of a monochromatic wave field with very limited damping. Even if this study is not sufficient to prove the capability of ICI-tech to simulate accurate wave fields, it shows that the numerical methods used already provide interesting precision for the simulation of free-surface flows. This can be tempered by the results obtained in Subsec. 5.2.4 on irregular wave fields. The different simulations regarding the propagation of wave fields in this thesis have been realized with an implementation of the sponge layer integrating an error. The results presented allowed to select the most meaningful simulations, which will be redone to evaluate the influence of this error. In particular,

this may explain the phase shift appearing on the waves located near the sponge layer for the simulations using a physical wave maker.

5.1.3 Influence of the anisotropic meshing on the propagation of waves

The influence of the anisotropic meshing on the generation and propagation of waves has been studied in Chap. 4. The errors produced by the meshing have been quantified through a comparison with results obtained using isotropic meshes, for which undamped wave fields could be generated. On the contrary, anisotropic meshes tended to induce numerical dissipation. Both the shape of the mesh generated, the frequency of adaptation and the time step used in the simulations seem to have a high impact on the quality of the results. If several developments have been realized, the damping could be reduced, but not eradicated. The influences of the time-stepping and of the spatial discretization have been examined. A mesh patch enhancing the precision of the results has been proposed, for an application to 2D test cases. The action of the mesh adaptation on free-surface flows have been identified and discussed, but not quantified.

5.2 First developments towards a demonstrator of FOWTs

The verification and validation of the solvers towards the simulation of FOWTs have been introduced in Chap. 4. This Sec. intends to present preliminary works and simulations realized for the development of a demonstrator of FOWT. A basic version of FOWT simulator is proposed, which considers a fixed FOWT in a monochromatic wave field. Several implementations have been realized with that objective in mind. The generation of complex wave fields and 3D wave makers have been considered, while the rotation of the rotor has been realized from the immersion procedure. Simple, prospective simulations proved the potential to represent a single FOWT in a wave field. Given the reconstruction of a huge number of WTs realized in Chap. 2, a case can be made for the simulation of several FOWTs in a NWT, in order to highlight their interactions. This would obviously require the validation of the solver on the aero- and hydrodynamics applications. More generally, this Sec. presents the challenges behind the development of a demonstrator of FOWTs.

5.2.1 Rotation of rigid WT blades

The rotation of WT blades has been implemented in ICI-tech, following the exact representation of the geometries. To that extent, the WT blades are reconstructed from their surface mesh. The mesh immersion techniques introduced in Chap. 2 are used. In order to accurately represent the blades, the rotation speed and the axis of rotation of the whole rotor are required. In the context of a rigid FOWT, no deformations of the structures are considered. The floater and the rotor are joined together, and thus share a coordinate system. Consequently, the rotation axis of the WT can be deduced from orientations of the floater, which are dependent on the buoyancy of the whole structure. The rotation speed can then be either prescribed or deduced from the torque generated by the forces applied on the WT blades.

This approach remains valid as soon as the different structures involved are rigid. In the context of flexible blades, aeroelasticity will be needed for both the tower and the blades of the WT. Note that only a few works have been realized concerning this topic, especially because the complexity of the physical problem is considerable. The validation of monophasic flows around airfoils tended to show that the aerodynamics are solved inaccurately at high Re using the $P1$ -VMS solver implemented in ICI-tech. Consequently, aside from the rotation utility, no further developments or simulations have been done. They will be required to reproduce the exact behavior of a FOWT.

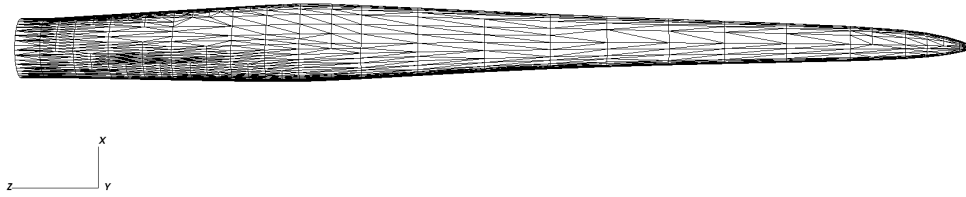


Figure 5.1: Surface mesh of a blade from a DanAero NM80 WT, used in the test case from the IEA Wind Task 29.

Once a solver will have been validated at high Reynolds number, the aeroelasticity will be a must-have for an accurate representation of WTs.

5.2.2 Preliminary works on IEAWind Task 29

During the thesis, ICI joined the Task 29 of the International Energy Agency dedicated to wind energy (IEA Wind). This task aims at improving the aerodynamic numerical models for WTs, with a focus on WT design codes. Experimental tunnel studies and on-site operating data have been acquired to provide comparison and validation data. The fourth phase of the IEA Wind Task 29 consists in the simulation of the aerodynamics of a WT blade, which are compared against full-scale field measurements. The test case proposed to the community considered the rotation of a blade from a DanAero NM80 WT, represented by the mesh presented in Fig. 5.1. The objective is to compare lifting line codes with CFD solvers, using the forces, torque and moments applied on the blade. The CFD solvers can also provide velocity and pressure fields. As the validation steps presented in Sec. 4.1 were not satisfying, no simulations realized with ICI-tech have been submitted to the Task 29.

The integration of the ICI inside the IEA Wind Task 29 will be profitable for future developments. The tunnel data provided consider airfoils and flows representative of WT operating. The different airfoils studied are defined as sections of the DanAero NM80 WT, and tested at flows whose Re is representative of operational conditions. This data set will enable the validation of a new VMS solver for high- Re flows. The validation will be possible on 2D airfoils, on different sections of the WT with their corresponding local Re . On-site data are also provided, but will only be profitable for long-term validation.

5.2.3 Simplified FOWT demonstrator

A simplified simulation of FOWT has been realized to operate as a proof of concept in terms of meshing. The FOWT presented in Sec. 2.4 is immersed in a NWT of length $50m$ and of width $20m$, where a still water of depth $5m$ is defined. A wave field of large amplitude is generated using a physical wave maker, and the FOWT is set immobile, with a fixed rotor. A mesh of $1M$ nodes, particularly refined at the free-surface and around the FOWT, is used. A view of the results obtained is presented in Fig. 5.2. No particular attention was placed on the quality and/or physical scales of the simulation. This test case was only dedicated to the observation of the interactions between the FOWT and the free-surface. Perturbations of the free-surface due to the FSI can be observed locally around the FOWT, and the forces applied to the structure could be measured to check the implementation of the utilities. This particular test case was realized with $\varepsilon = 10^{-2}m$ for both the FOWT, the free-surface and the piston, and a time step of $\Delta t = 10^{-2}s$ was used. Those parameters are particularly coarse, as the demonstrator was realized with computational efficiency in mind instead of accuracy. Despite this low-cost definition, the simulation of $1s$ of physical time required $2h40$, with $14K$ computational

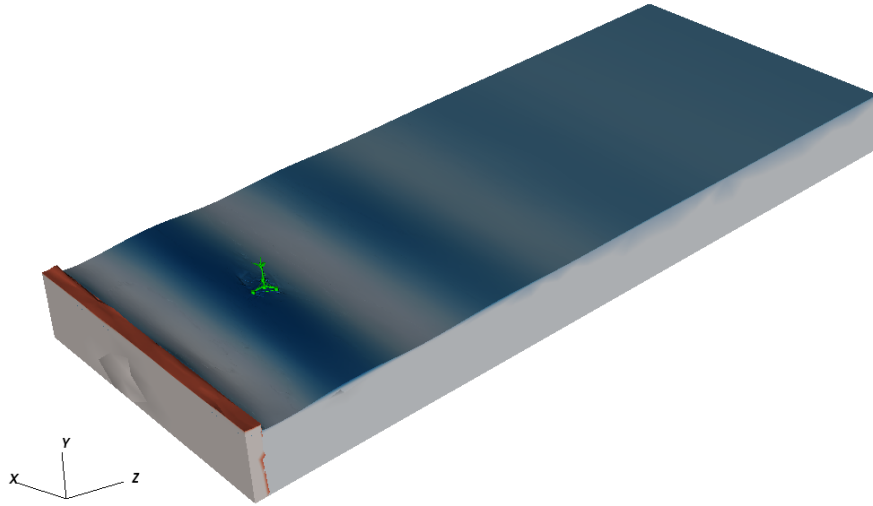


Figure 5.2: Proof of concept for the set-up of a FOWT demonstrator: fixed FOWT, with fixed rotor, in a monochromatic wave field.

points per core. Optimization are possible in the design of the test case, but ε and Δt will also need to be adjusted to get a better precision, especially around the FOWT for ε .

This work, coupled to the different developments presented in this Chap., enables to imagine the future NWT for FOWT testing based on ICI-tech. However, to simulate accurately the behavior of a FOWT, several advances will be required, aside from the validation of the solver. First of all, the WT must be operating. The rotor must be put in motion, which dramatically complicates the aerodynamics around it. The FOWT also needs to float, and moorings will be needed for stationkeeping. Finally, the waves generated must have a physical sense, and be representative of real sea states. Irregular wave spectra can be used, and reduced-scale test cases must consider similitude laws. If the moorings are not addressed in this thesis, the other two subjects are evoked in the following.

5.2.4 Towards a representation of real sea states

The definition of a FOWT demonstrator requires the positioning of the FOWT inside a wave field. The verification of the solver realized in Sec. 4.2 focuses on monochromatic wave fields only, which are not representative of complex sea states. The upcoming paragraphs focus on two different characteristics of the open sea: the 3D propagation of waves, and the combination of many waves to form an irregular wave field. These topics are briefly discussed, as the simulations introduced have been realized in a prospective way.

The development of a 3D NWT has thus been realized. The two different methods used in the validation process presented in Sec. 4.2, namely the source-term and physical wave generation methods, have been implemented. Using the source-term, 3D monochromatic wave fields can be generated thanks to minor modifications introducing a transverse wavelength in the formulation presented in Subsec. 3.3.2, as detailed in Calderer (2015). This method being suited for monochromatic waves only, the solution favored uses the physical wave maker, moved from HOS-NWT outputs. A NWT of length $50m$, of width $30m$ and of depth $5m$ has been designed, with 11 distinct wave makers positioned on the left of the domain. The results obtained are presented in Figs. 5.3, for the generation of a monochromatic wave field of amplitude $0.1m$, of frequency $0.5s^{-1}$ and of wavelength $6m$. A mesh of $800K$ nodes is anisotropically adapted in the computational domain,

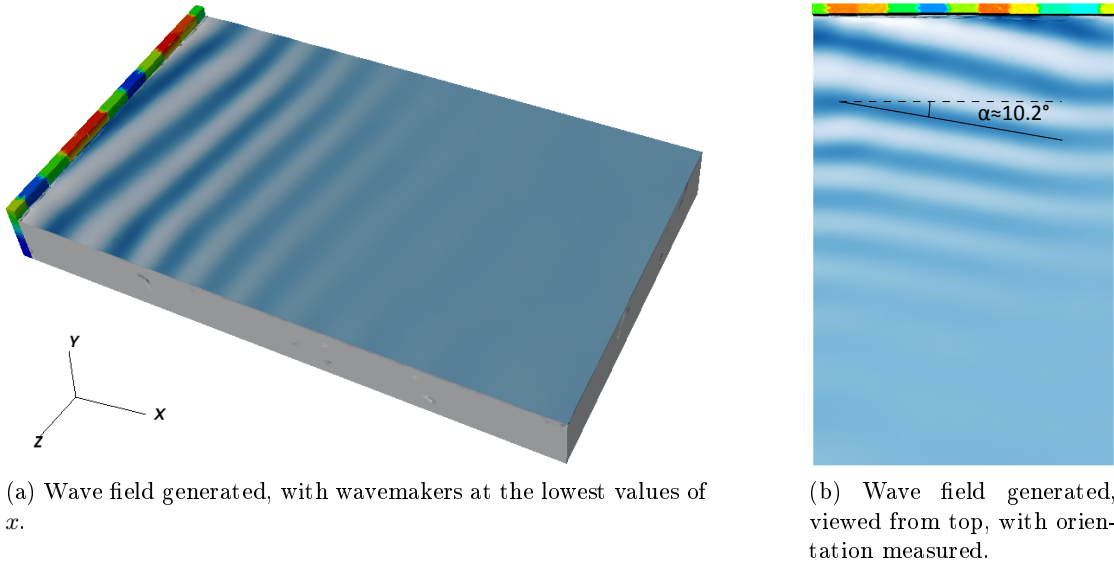


Figure 5.3: 3D NWT for the generation of a monochromatic wave field of angle $\alpha = 10^\circ$, with wave crests in blue and the wave makers on the left.

the free-surface is represented using $\varepsilon = 10^{-2}m$, and a time step of $10^{-2}s$ is used for the NS resolution. The wave field obtained is largely damped, which is due to both the meshing and the time stepping. The objective of this simulation was to validate the behavior of the 3D wave generator. The wave field obtained corresponded quite well to the prescribed wave field, as its approximative orientation is of 10.2° . Some expected perturbations are observed near the borders of the domain, because of the slip-conditions imposed. They only have a minor influence on the wave field in the middle of the domain, where the FOWT is supposed to be positioned, which is satisfying. More investigations will be needed once a validated monochromatic wave field will have been generated in 2D.

Aside from the 3D perspective, a complex sea state is formed by combined waves, of different amplitudes and frequencies. The representation of an irregular wave field is critical in the development of a NWT. In order to represent coherent waves, an irregular wave field has been represented from HOS-NWT following a JONSWAP spectrum of significant wave height $5 \cdot 10^{-2}m$, of peak period $1s$, and of shape factor 3.3 . For a 1/100th-scaled FOWT such as presented in Sec. 2.4, this wave field corresponds to a full-scale one with significant height $5m$ and peak period $10s$, i.e., a severe sea state. A 2D study has been realized, with an isotropic mesh similar to the ones used in Sec. 4.2 for the representation of the computational domain. The level-set of the free-surface is defined with $\varepsilon = 10^{-2}m$. The free-surface elevations obtained at $t = 20s$ for $\Delta t = 10^{-3}s$ are compared against HOS outputs in Fig. 5.4. This configuration is selected as it provides the best results on monochromatic wave fields, in particular with largely reduced damping. The curves immediately highlight the lack of precision achieved in this configuration. The wave field solved by ICI-tech is noisy, and those perturbations induce important errors for both the amplitude and the phase of the successive waves. The damping of the wave field is reduced, as the biggest waves can still be guessed on ICI-tech results. The order of magnitude of the wave field solved remains comparable with the one prescribed by HOS. However, the phase shift is increased as the waves are transported, which induces important errors. Moreover, the residual velocities discussed in Subsec. 4.2.3 induce here large perturbations. They tend to disturb the free-surface far from the wave maker, which then have an impact on the waves of small amplitude or wavelength. In order to simulate accurately irregular wave fields, the problem of the parasitic velocities will need to be solved.

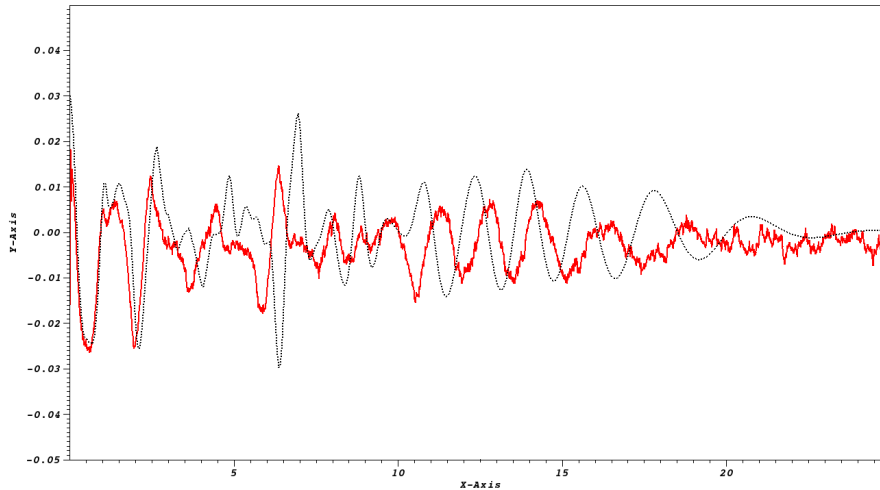


Figure 5.4: Irregular wave field from a JONSWAP spectrum, with significant wave height $0.05m$, peak period $1s$, and shape factor 3.3 . Black, dotted: Free-surface prescribed by HOS-NWT. Red, plain: Simulation realized with ICI-tech using an isotropic mesh, with $\Delta t = 10^{-3}s$. Scales in meters.

Even if the isotropic results are disappointing, a comparative study can be realized using anisotropic meshes of $50K$ nodes. Similarly to the studies realized in Sec. 4.2, the anisotropic characteristic enables to reduce the computational load induced by the meshing, and allows to translate to 3D test cases. The same time steps $\Delta t = 10^{-2}s$ and $\Delta t = 10^{-3}s$, along with adaptation every $7 \cdot 10^{-2}s$ and $7 \cdot 10^{-3}s$, are compared in Fig. 5.5. A rapid look at the results highlights the lack of accuracy of ICI-tech with an anisotropic meshing. The amplitude of the waves is largely reduced compared to HOS-NWT's prescriptions, and the waves of small wavelengths are not captured. The time step of $10^{-2}s$ is particularly not suited for such an erratic wave field, as even the amplitude of the last-generated wave is largely underestimated. The simulations ran with $\Delta t = 10^{-3}s$ seem to provide a better representation of the structures of huge wavelength, but do not capture the small waves too. With $x > 10m$, no simulation presented in Fig. 5.5 is able to give an idea of the target wave field. In comparison with the facts observed in Sec. 4.2, the frequency of mesh adaptation has here only a small impact. The poor quality of the free-surfaces generated overshadows its influence, which is limited to slight modifications of amplitude or phase. The poor results could find their origins in several factors. The influence of the time step can be linked with the difficulty to predict waves of different periods, and even smaller time steps could provide better results. The meshing may not be appropriated for this wave field, even if the minimum mesh size is similar to the one used for the isotropic meshing. The influence of the water dynamics below the surface, already discussed in Subsec. 4.2.2, could be a starting point.

In order to obtain real sea state in the NWT, a combination of the last two simulations, respectively 3D and irregular wave fields, is required. The generation of 3D wave fields using HOS-NWT seems to generate proper wave fields in terms of orientation, at least in the monochromatic situation tested. This provides a preliminary verification of the implementation of the 3D NWT. The concerns about numerical dissipation remain, in both of the simulations presented. Obviously, the level of accuracy is not acceptable to consider a coupling, and those simulations have not been realized for the moment. An accurate representation of irregular wave fields would first be needed in 2D, with an isotropic meshing. The criteria defined from this study could then be used for the simulation of accurate 2D irregular wave fields using anisotropic meshes. A coupling will be considered once this last step will be reached. Using the parameters defined in the

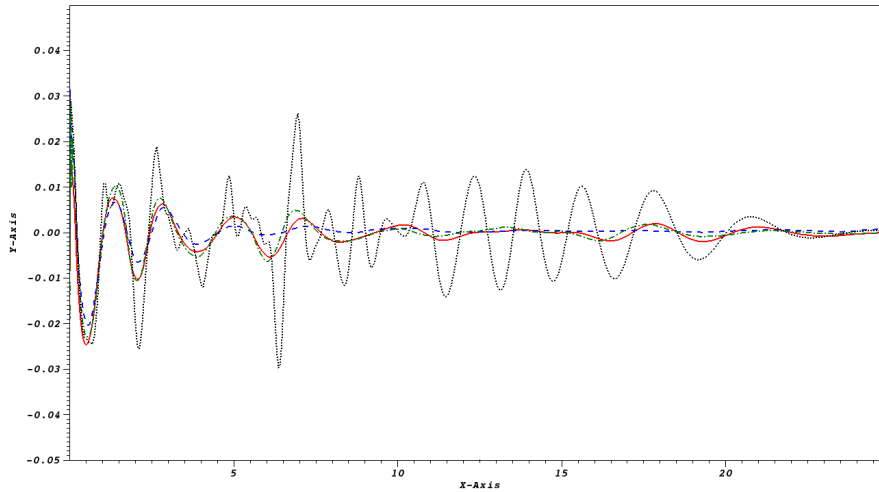


Figure 5.5: Irregular wave field simulated at $t = 20s$ with target from a JONSWAP spectrum, of significant wave height $0.05m$, peak period $1s$, and shape factor 3.3 . Results of the simulations obtained with ICI-tech using an anisotropic mesh. Black, dotted: Free-surface prescribed by HOS-NWT. Red, plain: $\Delta t = 10^{-2}s$, adaptation every $7 \cdot 10^{-2}s$. Blue, dashed: $\Delta t = 10^{-3}s$, adaptation every $7 \cdot 10^{-3}s$. Green, dash-dotted: $\Delta t = 10^{-3}s$, adaptation every $7 \cdot 10^{-2}s$. Scales in meters.

corresponding simulation, the resolution of real sea states will be on the way.

5.3 Perspectives

Given the numerical methods used by ICI-tech and the computational load induced, high accuracy in the representation of the physics simulated is required, as already stated in Chap. 1. ICI-tech has the potential to be used for the validation and calibration of engineering codes. Consequently, for an application of ICI-tech to the simulation of FOWTs, a lot of current limitations of the model developed will have to be addressed. Those modifications may be applied from short- to long-term basis, depending on the priorities defined for both this project and the other activities of ICI.

5.3.1 Higher-order FEM

The development of higher-order FEM is already in process within other projects at ICI. This study initially focused on the representation of FOWTs at a reduced cost, before considering the validation of a first-order finite element solver. A new solver is currently being implemented at ICI, which provides higher orders of precision both in space and time. In time, numerical schemes of higher complexity are used to compute the values of velocity and pressure provided by the NS equations at the next time step. In space, a higher order consists in changing from linear interpolation functions for polynomials of higher orders. This imposes using more interpolation points within each finite element. The computational load per element rises, but the accuracy of the resolution increases.

A higher order of interpolation of the FEM increases the accuracy of the problem solved, which finally improves the results obtained. As mentioned in Sec. 3.1, the orders of accuracy in time and space need to be similar, in order to balance the approximation errors between space and time. The numerical schemes will thus need to be modified, in order to provide a convergence similar to the one in space. The corollary to the rise in order is that bigger mesh cells and time steps can be used for the same accuracy than with P1-FEM.

The impact of higher-order FEM on the simulations of FOWT will probably be positive. The additional precision should have an impact on the quality of the results, while the increased computational at equivalent meshing will need to be justified. The solver will need to undergo validation representative of the operational cases for both the aero- and hydrodynamics. Concerning the aerodynamics, the higher-order solver will offer to represent more accurately the turbulent effects with an identical mesh. This will allow a better representation of the flows around WT blades, which should improve the evaluation of the aerodynamic coefficients. From a free-surface point of view, the higher order should reduce the numerical damping observed, which will provide flexibility to the simulation procedure.

5.3.2 Mesh adaptation for the propagation of waves

Currently, the simulation of wave fields with anisotropic mesh adaptation tends to create numerical damping, with origins in both the adaptation frequency and in the cell size. Further works will be needed to reach the accuracy achieved with isotropic meshing using anisotropic meshes. Several options may be considered on a short-term basis. The anisotropic mesher is an important tool developed at ICI, and its use is necessary for the realization of accurate, large-scale simulations of FOWTs.

Working on the frequency of adaptation is important, as it moves the mesh calls towards areas of the computational domain where they are needed. A small frequency of adaptation may lead to poorly positioned mesh cells, which translate to inaccurate results. On the contrary, too frequent adaptations may induce damping and/or approximations in the results, due to the interpolation realized during the adaptation process. The determination of the correct frequency of adaptation has to be done regarding the test case simulated before each simulation. The definition of guidelines in the context of wave generation is still to be done. The quantification of the numerical dissipation induced by the adaptation itself will be required. The numerical damping provoked by not adapting the mesh will also need to be addressed, i.e., when the mesh unadapted for a few time steps become less and less suited to the capture of the wave dynamics.

By nature, anisotropic meshing induces another limitation for wave propagation. The anisotropic characteristic tends to stretch the mesh cells appearing near the free-surface, which reduces the number of cells describing each wave period. However, a limited number of cells in the direction of the wave propagation tends to generate numerical dissipation, which turns into damping. The propagation of wave fields using anisotropic adaptation has been particularly regarded recently, but still, no satisfying results have been obtained. A continuity of this study should focus on the production of anisotropic meshes enabling an accurate propagation of the wave fields. The limitation of the stretching of mesh cells around the free-surface may be a starting point, even if it will partly tackle the advantages linked to anisotropic meshing. The evolution of the damping with the transition towards solvers of higher orders will need to be examined.

The anisotropic adaptation as realized at the moment may also have an importance. Even if the refined areas are, theoretically, exclusively defined using the error estimator, in practice several constraints are used. As an example, the minimal mesh size is constrained to prevent the mesher to request "infinitely" small mesh sizes at singularities. This criterion allows to generate an acceptable mesh even if, e.g., the definition of the test case features sharp interfaces. Some other constraints are related to the quality of the mesh, e.g. to guarantee a minimum background mesh size or to avoid highly-stretched cells. The number of points in the computational mesh is also constrained, for user-friendliness. The result is a mesh adaptation process which does not respect exactly the prescribed error estimator, and introduces some bias in the meshing. A comeback to

a lower-level mesh adaptation can be a viable option to optimize the representation of waves. This approach would consider exclusively the error estimator in the adaptation process, with a target error to respect on each element. This paradigm can not be followed for the simulation of a wave field generated using a source-term wave maker, as the meshing of the source area has to be enriched. The representation of a physical wave maker induces more computational constraints, and should not include any singularity, i.e., sharp angle. But no theoretical limitation shall arise from this test case. The propagation of wave fields in this configuration can be studied and compared to results obtained with isotropic meshing. A rapid comparison will provide first information about the potential of anisotropic meshing to not only represent the wave field, but also to propagate it.

5.3.3 Mid- to long-term perspectives for an accurate FOWT simulator

The realization of a FOWT simulator needs to represent one or several FOWTs in their operational configuration. To that extent, the definition of a wind/wave simulator is required, with both fluids being solved accurately. The different components of the FOWT must also be addressed. The investigations realized during this PhD concerned mostly the solver, for which validation test cases were run for aero- and hydrodynamics. Despite the poor validation results obtained, the design of a simple NWT has been realized during the thesis. Several utilities have been implemented, such as rigid-rotor rotation or buoyancy, but a lot of work is still to be done.

First of all, the validation of hydrodynamics seems to be more advanced than the aerodynamic one. Following this observation, it makes more sense to focus on the next developments on the design of an accurate hydrodynamic model. To that extent, a mid-term focus could be the development of an advanced NWT, enabling the simulation of accurate sea state and the reaction of a FOWT to it. This model requires the representation of moorings, and of their action on the floater. The NWT obtained could provide interesting load evaluation of the loads applied by waves on the structure, and which forces are applied on the moorings. This type of study could also model the electrical cable connected to the FOWT.

The NWT defined would not include the aerodynamics, which are rather important for FOWTs. Aside from the validation of the solver for high- Re flows, several works will be required for the simulation of an operational WT. Realistic wind conditions require a turbulent inflow, as well as the definition of an ABL. The aerodynamics of the ambient wind will need to be validated to be used for the simulation of FOWTs. These developments will allow the resolution of accurate flows around WTs with rigid rotors. This simulator will already provide interesting results, but is not exactly representative of an operational FOWT. The aeroelasticity will be needed, as well as control laws. These last developments may be needed on a long-term perspective for ICI-tech.

A simulator of FOWT integrating the different propositions formulated in this Subsec. would be able to provide an important precision for the simulation of one or multiple FOWTs. The validation and calibration of engineering codes could be realized with such a simulation engine, as already mentioned. The path proposed for the definition of this simulator with ICI-tech has been realized regarding the situation currently found at ICI. It is subject to evolve, if other opportunities are found.

Once the simulator of FOWTs will have been defined, several extensions will be available. Three propositions are made in the following paragraphs, with different targets. The first one presents a natural way to extend the simulator to study wake interactions. The second one focuses on the environmental conditions under which a FOWT is operat-

ing. The last one proposes a development towards the simulation of a floating wind farm.

The massively-parallel framework of ICI-tech will easily allow the simulation of several FOWTs. This transition will rely on the robust partitioning strategies detailed in 2.3 to represent the different FOWTs. Theoretically, no limitation exist for the number of FOWTs represented in the computational domain. A major constraint corresponds to the size of the cluster on which ICI-tech will be deployed, i.e., on the number of cores available for the simulation. The representation of a large number of FOWTs proved to be scalable, as presented in Sec 2.4. The resolution of the Navier-Stokes equation in the computational domain is also well parallelized, and the number of points realized for the resolution around each FOWT should not differ much. The most constraining factor will be the wake generated by the upwind WTs, which will need additional points at each instant to conserve a good representation of the aerodynamics propagated in the computational domain. This type of simulation would obviously be extremely expensive compared to the reduced-order approaches currently used in the industry. They could still have a utility for the evaluation of potential complex coupled aero/hydro efforts arising from farm effects.

The simulator of FOWT produced will also be dependent on the boundary conditions prescribed at the inlet. Currently, the generation of sea states in ICI-tech can be realized using physical wave makers. This type of definition allows defining realistic sea states, e.g., as proposed in Subsec. 5.2.4 following a JONSWAP spectrum. This approach generates an approximation of real sea states. A higher fidelity could be reached using a simulation of the interactions between the wind and the free-surface, which generate the sea states. Environmental data could be used as inlet of this wave generator, which would be much larger than the regular NWT. Time will be needed to generate waves from these air/water interactions, which could ultimately be used to feed the NWT in inlet. The aerodynamics and hydrodynamics would be more representative of real environmental conditions under which FOWTs are operating.

The last proposition focuses on the potential level of accuracy provided by the methodology detailed in this thesis. The numerical framework of ICI-tech has one of the highest potential to simulate precisely the flows around FOWTs. The development of a versatile simulator of FOWT would give the opportunity to generate a lot of highly-accurate data. This amount of data could be used to feed machine-learning codes. Many data-driven paradigms may be followed to reduce the complexity of the computations. A first approach could be to realize a series of computations focusing on the rotor of the FOWT, with different atmospheric conditions. The data produced could be used, e.g., to feed a neuron network, allowing the definition of a black box returning the outflow from a specified inflow. Future simulations could then skip the representation of the rotor, expensive step as the geometries and the evolution of the boundary layers are complex to represent. This reduction of the computational load would pave the way for large scale simulations, at the scale of a wind farm.

All the potential developments presented are dedicated to the development of an accurate framework for the simulation of FOWTs. Different points of views have been regarded, always with a focus on the level of precision provided by the framework. Accurate simulators will have a role to play in the floating wind industry, as a complementary approach to the set-up of prototypes, experimental studies or computationally-cheap design codes. They can help calibrating industrial softwares, and bring the ability to study specific parts of the structures, to get a better understanding of the dynamics of FOWTs.

Bibliography

- Ahrabi, B. R., W. K. Anderson, and J. C. Newman
2017. An adjoint-based hp-adaptive stabilized finite-element method with shock capturing for turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 318:1030 – 1065.
- Almeida, R. C., R. A. Feijóo, A. C. Galeão, C. Padra, and R. S. Silva
2000. Adaptive finite element computational fluid dynamics using an anisotropic error estimator. *Computer Methods in Applied Mechanics and Engineering*, 182(3):379 – 400.
- Bak, C., P. Fuglsang, J. Johansen, and I. Antoniou
2000. *Wind Tunnel Tests of the NACA 63-415 and a Modified NACA63-415 Airfoil*, number 1193. Forskningscenter Risoe, Risoe-R, Denmark.
- Bakker, A.
2006. Turbulence models. A. Bakker’s website:
www.bakker.org/dartmouth06/engs150/10-rans.pdf.
- Bazilevs, Y., M.-C. Hsu, I. Akkerman, S. Wright, K. Takizawa, B. Henicke, T. Spielman, and T. Tezduyar
2011a. 3d simulation of wind turbine rotors at full scale. part i: Geometry modeling and aerodynamics. *International Journal for Numerical Methods in Fluids*, 65(1-3):207–235.
- Bazilevs, Y., M.-C. Hsu, J. Kiendl, R. Wüchner, and K.-U. Bletzinger
2011b. 3d simulation of wind turbine rotors at full scale. part ii: Fluid–structure interaction modeling with composite blades. *International Journal for Numerical Methods in Fluids*, 65(1-3):236–253.
- Bazilevs, Y., A. Korobenko, X. Deng, and J. Yan
2015. Novel structural modeling and mesh moving techniques for advanced fluid–structure interaction simulation of wind turbines. *International Journal for Numerical Methods in Engineering*, 102(3-4):766–783.
- Béal, P., J. Koko, and R. Touzani
2001. Mesh r-adaptation for unilateral contact problems. *International Journal of Applied Mathematics and Computer Science*, 12:No. 1, 9–16.
- Benard, P., A. Viré, V. Moureau, G. Lartigue, L. Beaudet, P. Deglaire, and L. Bricteux
2018. Large-eddy simulation of wind turbines wakes including geometrical effects. *Computers & Fluids*, 173:133 – 139.
- Betancourt, L. S.
2017. Modeling of high performance smc behavior applications to 3d compression molding simulation. *Mines ParisTech*.

- Beyer, F., T. Choynet, M. Kretschmer, and P. W. Cheng
 2015. Coupled mbs-cfd simulation of the ideal floating offshore wind turbine foundation compared to wave tank model test data. In *Proceedings of the Twenty-fifth (2015) International Ocean and Polar Engineering Conference*.
- Blocken, B., T. Stathopoulos, and J. Carmeliet
 2007. Cfd simulation of the atmospheric boundary layer: wall function problems. *Atmospheric Environment*, 41(2):238 – 252.
- Blondel, F., R. Boisard, M. Milekovic, G. Ferrer, C. Lienard, and D. Teixeira
 2016. Validation and comparison of aerodynamic modelling approaches for wind turbines. *Journal of Physics: Conference Series*, 753:022029.
- Bonnefoy, F., D. L. Touzé, and P. Ferrant
 2006. A fully-spectral 3d time-domain model for second-order simulation of wavetank experiments. part b: Validation, calibration versus experiments and sample applications. *Applied Ocean Research*, 28(2):121 – 132.
- Borazjani, I., L. Ge, and F. Sotiropoulos
 2008. Curvilinear immersed boundary method for simulating fluid structure interaction with complex 3d rigid bodies. *Journal of Computational Physics*, 227(16):7587 – 7620.
- Borg, M. and M. Collu
 2015. Offshore floating vertical axis wind turbines, dynamics modelling state of the art. part iii: Hydrodynamics and coupled modelling approaches. *Renewable and Sustainable Energy Reviews*, 46:296 – 310.
- Brackbill, J., D. Kothe, and C. Zemach
 1992. A continuum method for modeling surface tension. *Journal of Computational Physics*, 100(2):335 – 354.
- Butterfield, S., W. Musial, J. Jonkman, P. Sclavounos, and L. Wayman
 2005. Engineering challenges for floating offshore wind turbines. In *Copenhagen Offshore Wind Conference, Copenhagen, Denmark*, Pp. 377–382. Citeseer.
- Calderer, A.
 2015. *Fluid-Structure Interaction Simulation of Complex Floating Structures and Waves*. PhD thesis, University of Minnesota.
- Calderer, A., X. Guo, L. Shen, and F. Sotiropoulos
 2018. Fluid–structure interaction simulation of floating structures interacting with complex, large-scale ocean waves and atmospheric turbulence with application to floating offshore wind turbines. *Journal of Computational Physics*, 355:144 – 175.
- Carstensen, C.
 2006. *Clément Interpolation and Its Role in Adaptive Finite Element Error Control*, Pp. 27–43. Basel: Birkhäuser Basel.
- Choi, J. and S. B. Yoon
 2009. Numerical simulations using momentum source wave-maker applied to rans equation model. *Coastal Engineering*, 56(10):1043–1060.
- Codina, R.
 2000. Stabilization of incompressibility and convection through orthogonal sub-scales in finite element methods. *Computer Methods in Applied Mechanics and Engineering*, 190(13):1579 – 1599.

- Codina, R., S. Badia, J. Baiges, and J. Principe
2017. *Variational Multiscale Methods in Computational Fluid Dynamics*, Pp. 1–28. American Cancer Society.
- Coulling, A. J., A. J. Goupee, A. N. Robertson, J. M. Jonkman, and H. J. Dagher
2013. Validation of a FAST semi-submersible floating wind turbine numerical model with DeepCwind test data. *Journal of Renewable and Sustainable Energy*, 5(2):023116.
- Coupez, T.
2011. Metric construction by length distribution tensor and edge based error for anisotropic adaptive meshing. *Journal of Computational Physics*, 230(7):2391 – 2405.
- Coupez, T. and E. Hachem
2013. Solution of high-reynolds incompressible flow with stabilized finite element and adaptive anisotropic meshing. *Computer Methods in Applied Mechanics and Engineering*, 267:65–85.
- Cummins, W.
1962. The impulse response function and ship motions. Technical report, David Taylor Model Basin Washington DC.
- Davidson, J. and J. Ringwood
2017. Mathematical modelling of mooring systems for wave energy converters—a review. *Energies*, 10(5):666.
- Deiterding, R. and S. L. Wood
2016. Predictive wind turbine simulation with an adaptive lattice boltzmann method for moving boundaries. *Journal of Physics: Conference Series*, 753:082005.
- Digonnet, H., T. Coupez, P. Laure, and L. Silva
2017. Massively parallel anisotropic mesh adaptation. *The International Journal of High Performance Computing Applications*.
- Ding, S., M. Mannan, and A. Poo
2004. Oriented bounding box and octree based global interference detection in 5-axis machining of free-form surfaces. *Computer-Aided Design*, 36(13):1281 – 1294.
- DNV-GL
2019. Coupled analysis of floating wind turbines. *Recommended practice*, DNVGL-RP-0286.
- Ducoin, A., M. Shadloo, and S. Roy
2017. Direct numerical simulation of flow instabilities over savonius style wind turbine blades. *Renewable Energy*, 105:374 – 385.
- Ducrozet, G., F. Bonnefoy, D. Le Touzé, and P. Ferrant
2012. A modified high-order spectral method for wavemaker modeling in a numerical wave tank. *European Journal of Mechanics-B/Fluids*, 34:19–34.
- Ducrozet, G., F. Bonnefoy, D. L. Touzé, and P. Ferrant
2016. Hos-ocean: Open-source solver for nonlinear waves in open ocean based on high-order spectral method. *Computer Physics Communications*, 203:245 – 254.
- EMODnet
2019. Bathymetry in europe. EMODnet website: portal.emodnet-bathymetry.eu/.
- Engsig-Karup, A., P. Madsen, H. Bingham, and P. Thomsen
2007. *Unstructured nodal DG-FEM solution of high-order Boussinesq-type equations*. PhD thesis, Technical University of Denmark.

- Formaggia, L. and S. Perotto
2001. New anisotropic a priori error estimates. *Numerische Mathematik*, 89(4):641–667.
- Frey, P. and F. Alauzet
2005. Anisotropic mesh adaptation for cfd computations. *Computer Methods in Applied Mechanics and Engineering*, 194(48):5068 – 5082.
- GeneralElectric
2019. Haliade-x offshore wind turbine platform. GE website:
www.ge.com/renewableenergy/wind-energy/offshore-wind/haliade-x-offshore-turbine.
- Ha, T., P. Lin, and Y.-S. Cho
2013. Generation of 3d regular and irregular waves using navier–stokes equations model with an internal wave maker. *Coastal Engineering*, 76:55–67.
- Hachem, E., T. Kloczko, H. Digonnet, and T. Coupez
2012. Stabilized finite element solution to handle complex heat and fluid flows in industrial furnaces using the immersed volume method. *International Journal for Numerical Methods in Fluids*, 68(1):99–121.
- Hachem, E., B. Rivaux, T. Kloczko, H. Digonnet, and T. Coupez
2010. Stabilized finite element method for incompressible flows with high reynolds number. *Journal of Computational Physics*, 229(23):8643 – 8665.
- Hansen, M. O. L., J. N. Sørensen, S. Voutsinas, N. Sørensen, and H. A. Madsen
2006. State of the art in wind turbine aerodynamics and aeroelasticity. *Progress in aerospace sciences*, 42(4):285–330.
- Hughes, T. J., G. R. Feijóo, L. Mazzei, and J.-B. Quincy
1998. The variational multiscale method—a paradigm for computational mechanics. *Computer methods in applied mechanics and engineering*, 166(1):3–24.
- Itō, S.
1977. *Study of the transient heave oscillation of a floating cylinder*. PhD thesis, Massachusetts Institute of Technology.
- James, R. and M. C. Ros
2015. Floating offshore wind: Market technology review. *The Carbon Trust, London, UK*.
- Jonkman, J. M.
2007. *Dynamics modeling and loads analysis of an offshore floating wind turbine*. University of Colorado at Boulder.
- Joulin, P., M. Mayol, F. Blondel, V. Masson, Q. Rodier, and C. Lac
2019. Coupling the actuator line method to the high order meteorological LES model meso-NH to study wind farm wakes impacts on local meteorology. *Journal of Physics: Conference Series*, 1256:012019.
- Jung, K. H., K.-A. Chang, and H. J. Jo
2006. Viscous effect on the roll motion of a rectangular structure. *Journal of engineering mechanics*, 132(2):190–200.
- Kirby, N., M. Lockett, L. Xu, and W. Siepmann
2001. HvdC transmission for large offshore windfarms. *IET Conference Proceedings*, Pp. 162–168(6).

- Kolmogorov, A. N.
1941. The local structure of turbulence in incompressible viscous fluid for very large reynolds numbers. *30(4):299–303*.
- Krogstad, P.-Å. and P. E. Eriksen
2013. “blind test” calculations of the performance and wake development for a model wind turbine. *Renewable energy*, 50:325–333.
- Kudela, L., N. Zander, S. Kollmannsberger, and E. Rank
2016. Smart octrees: Accurately integrating discontinuous functions in 3d. *Computer Methods in Applied Mechanics and Engineering*, 306:406 – 426.
- Kvittem, M. I., E. E. Bachynski, and T. Moan
2012. Effects of hydrodynamic modelling in fully coupled simulations of a semi-submersible wind turbine. *Energy Procedia*, 24:351 – 362. Selected papers from Deep Sea Offshore Wind R&D Conference, Trondheim, Norway, 19-20 January 2012.
- Lacaze, J.-B., B. Molin, F. Remy, H. Branger, and C. Luneau
2014. Small scale tests of floating wind turbines in the wind and wave flume of luminy. In *14th journées d’hydrodynamique, Val de Reuil, France, 18-20 Nov 2014*.
- Lanzafame, R. and M. Messina
2007. Fluid dynamics wind turbine design: Critical analysis, optimization and application of bem theory. *Renewable Energy*, 32(14):2291 – 2305.
- Leble, V. and G. Barakos
2016. Demonstration of a coupled floating offshore wind turbine analysis with high-fidelity methods. *Journal of Fluids and Structures*, 62:272–293.
- Leroy, V.
2018. *Aérodynamique instationnaire pour l’analyse de la tenue à la mer des éoliennes flottantes*. PhD thesis, Ecole Centrale de Nantes.
- Leroy, V., J.-C. Gilloteaux, M. Lynch, A. Babarit, and P. Ferrant
2019. Impact of aerodynamic modeling on seakeeping performance of a floating horizontal axis wind turbine. *Wind Energy*, 0(0).
- Liu, Y., Q. Xiao, A. Incecik, C. Peyrard, and D. Wan
2017. Establishing a fully coupled cfd analysis tool for floating offshore wind turbines. *Renewable Energy*, 112:280 – 301.
- Madsen, H. A., R. Mikkelsen, S. Øye, C. Bak, and J. Johansen
2007. A detailed investigation of the blade element momentum (BEM) model based on analytical and numerical results and proposal for modifications of the BEM model. *Journal of Physics: Conference Series*, 75:012016.
- Maries, A., A. Haque, S. L. Yilmaz, M. B. Nik, and G. E. Marai
2012. Interactive exploration of stress tensors used in computational turbulent combustion. In *Mathematics and Visualization*, Pp. 137–156. Springer Berlin Heidelberg.
- Meneveau, C. and J. Katz
2000. Scale-invariance and turbulence models for large-eddy simulation. *Annual Review of Fluid Mechanics*, 32(1):1–32.
- Morison, J., J. Johnson, S. Schaaf, et al.
1950. The force exerted by surface waves on piles. *Journal of Petroleum Technology*, 2(05):149–154.

- Nematbakhsh, A., E. E. Bachynski, Z. Gao, and T. Moan
 2015. Comparison of wave load effects on a tlp wind turbine by using computational fluid dynamics and potential flow theory approaches. *Applied Ocean Research*, 53:142–154.
- Paik, I. and J. M. Roesset
 1996. Use of quadratic transfer functions to predict response of tension leg platforms. *Journal of engineering mechanics*, 122(9):882–889.
- Pinon, G., C. Carlier, A. Fur, B. Gaurier, G. Germain, and É. Rivoalen
 2017. Account of ambient turbulence for turbine wakes using a synthetic-eddy-method. *Journal of Physics: Conference Series*, 854:012016.
- Polinder, H., F. F. Van der Pijl, G.-J. De Vilder, and P. J. Tavner
 2006. Comparison of direct-drive and geared generator concepts for wind turbines. *IEEE Transactions on energy conversion*, 21(3):725–733.
- Quallen, S. and T. Xing
 2016. Cfd simulation of a floating offshore wind turbine system using a variable-speed generator-torque controller. *Renewable Energy*, 97:230–242.
- Rapin, M. and J.-M. Noël
 2010. *Energie éolienne: principes-études de cas*. Dunod.
- Rossi, E., A. Colagrossi, D. Durante, and G. Graziani
 2016. Simulating 2d viscous flow around geometries with vertices through the diffused vortex hydrodynamics method. *Computer Methods in Applied Mechanics and Engineering*, 302:147–169.
- Rossi, E., A. Colagrossi, G. Oger, and D. Le Touzé
 2018. Multiple bifurcations of the flow over stalled airfoils when changing the reynolds number. *Journal of Fluid Mechanics*, 846:356–391.
- Santoso, S.
 2018. *Simulation d'écoulement fluide à partir de données réelles*. PhD thesis, Institut de Calcul Intensif.
- Sark, W. G. V., H. C. V. der Velde, J. P. Coelingh, and W. A. Bierbooms
 2019. Do we really need rotor equivalent wind speed? *Wind Energy*, 22(6):745–763.
- Scardovelli, R. and S. Zaleski
 1999. Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics*, 31(1):567–603.
- Schäfer, M., S. Turek, F. Durst, E. Krause, and R. Rannacher
 1996. *Benchmark Computations of Laminar Flow Around a Cylinder*, Pp. 547–566. Wiesbaden: Vieweg+Teubner Verlag.
- Sebastian, T. and M. Lackner
 2011. Offshore floating wind turbines - an aerodynamic perspective. In *49th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, P. 720.
- Sebastian, T. and M. Lackner
 2012a. Characterization of the unsteady aerodynamics of offshore floating wind turbines. *Wind Energy*, 16(3):339–352.

- Sebastian, T. and M. Lackner
 2012b. Development of a free vortex wake method code for offshore floating wind turbines. *Renewable Energy*, 46:269 – 275.
- Sempreviva, A., M. Schiano, S. Pensieri, A. Semedo, R. Tome, R. Bozzano, M. Borghini, F. Grasso, L. Sørensen, J. Teixeira, and C. Transerici
 2010. Observed development of the vertical structure of the marine boundary layer during the lasie experiment in the ligurian sea. *Annales Geophysicae*, 28(1):17–25.
- Shen, W. Z.
 2002. Numerical modeling of wind turbine wakes. *J. Fluids Eng*, 124(2):393–399.
- Shen, W. Z., J. N. Sørensen, and J. Zhang
 2007. Actuator surface model for wind turbine flow computations. In *2007 European Wind Energy Conference and Exhibition*.
- Sørensen, J. N. and A. Myken
 1992. Unsteady actuator disc model for horizontal axis wind turbines. *Journal of Wind Engineering and Industrial Aerodynamics*, 39(1-3):139–149.
- Sussman, M., P. Smereka, and S. Osher
 1994. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114(1):146 – 159.
- Theodorsen, T.
 1933. Theory of wing sections of arbitrary shape. Technical report, NACA.
- Tran, T.-T. and D.-H. Kim
 2015. The platform pitching motion of floating offshore wind turbine: A preliminary unsteady aerodynamic analysis. *Journal of Wind Engineering and Industrial Aerodynamics*, 142:65–81.
- Tran, T. T. and D.-H. Kim
 2018. A cfd study of coupled aerodynamic-hydrodynamic loads on a semisubmersible floating offshore wind turbine. *Wind Energy*, 21(1):70–85.
- Troen, I. and E. L. Petersen
 1989. *European wind atlas*. Risø National Laboratory.
- Ville, L., L. Silva, and T. Coupez
 2011. Convected level set method for the numerical simulation of fluid buckling. *International Journal for numerical methods in fluids*, 66(3):324–344.
- Wackers, J., G. Deng, E. Guilmineau, A. Leroyer, P. Queutey, and M. Visonneau
 2014. Combined refinement criteria for anisotropic grid refinement in free-surface flow simulation. *Computers & Fluids*, 92:209 – 222.
- WindEurope
 2018a. Offshore wind in europe. Technical report, WindEurope.
- WindEurope
 2018b. Wind in power 2017. Technical report, WindEurope.
- WindEurope
 2019. Wind energy in europe in 2018. Technical report, WindEurope.
- Windt, C., J. Davidson, and J. V. Ringwood
 2018. High-fidelity numerical modelling of ocean wave energy systems: A review of computational fluid dynamics-based numerical wave tanks. *Renewable and Sustainable Energy Reviews*, 93:610 – 630.

Wu, C.-H. K. and V.-T. Nguyen

2016. Aerodynamic simulations of offshore floating wind turbine in platform-induced pitching motion. *Wind Energy*, 20:835–858.

Yan, J., A. Korobenko, X. Deng, and Y. Bazilevs

2016. Computational free-surface fluid–structure interaction with application to floating offshore wind turbines. *Computers & Fluids*, 141:155–174.

Yaramasu, V., B. Wu, P. C. Sen, S. Kouro, and M. Narimani

2015. High-power wind energy conversion systems: State-of-the-art and emerging technologies. *Proceedings of the IEEE*, 103(5):740–788.

Titre : Simulations CFD avec adaptation de maillage anisotrope : application à la simulation d'éoliennes offshore.

Mots clés : CFD ; équations de Navier-Stokes ; frontières immergées ; éoliennes flottantes

Résumé : Le calcul des performances et des efforts appliqués sur une éolienne offshore est actuellement réalisé à l'aide d'outils basés sur des approches quasi-statiques. Ces approches sont intéressantes pour leur vitesse de calcul, elles sont cependant perfectibles suivant la méthode de mise en œuvre et suivant les cas de chargement étudiés. Une approche alternative consiste à utiliser la modélisation CFD. Cette thèse s'intéresse à des méthodes d'une haute précision, ayant le potentiel de fournir des écoulements et efforts précis.

La plateforme logicielle hautement parallélisée ICI-tech est utilisée dans cette thèse. Elle se base sur une résolution des équations de Navier-Stokes dans une approche multi-échelle, effectuée à l'aide d'éléments finis stabilisés. La

représentation des phases dans le domaine de calcul est réalisée grâce à une méthode frontières immergées. Des implémentations ont été réalisées dans ICI-tech afin de pouvoir simuler des éoliennes flottantes. L'interaction fluide-structure et un bassin de houle numérique ont notamment été considérés.

Un processus de vérification et validation s'est intéressé au comportement du solveur dans des conditions reproduisant celles impactant des éoliennes flottantes. Le niveau de précision atteint par les écoulements à haut Reynolds et la propagation de champs de houle s'est avéré être décevant. L'influence du maillage anisotrope sur les résultats obtenus a été quantifiée. Plusieurs pistes visant à améliorer la précision des simulations ont été introduites.

Title : CFD simulation with anisotropic mesh adaptation: application to floating offshore wind turbines.

Keywords : CFD ; Navier-Stokes equations ; immersed boundaries ; floating wind turbines

Abstract : The simulation of Floating Offshore Wind Turbines (FOWTs) is a tool to help this technology reach an industrial scale. Nowadays, low-precision numerical methods are used for the dimensioning of the structures, as they involve a reduced computational effort. This PhD thesis focused on the development of highly-accurate numerical methods, with a potential to provide a thin description of the flows and efforts around FOWTs.

The simulations presented in this thesis have been realized on the highly-parallelized software platform ICI-tech. A resolution of the Navier-Stokes equations in a Variational MultiScale formulation is performed using Stabilized Finite Elements. The representation of the different phases in the computational domain is achieved

using immersed boundary methods. Several numerical tools have been implemented in ICI-tech towards an application to the simulation of FOWTs. A fluid-structure interaction paradigm has been set up, and a numerical wave tank has been defined.

Verification and validation studies have been realized to assess the solver results for environmental conditions representative of those observed for operating FOWT. The accuracy achieved for both the aerodynamics at high Reynolds numbers and the propagation of wave fields has been disappointing. The influence of the anisotropic meshing on the results presented has been quantified. Several options aiming at increasing the accuracy of the simulations have been discussed.