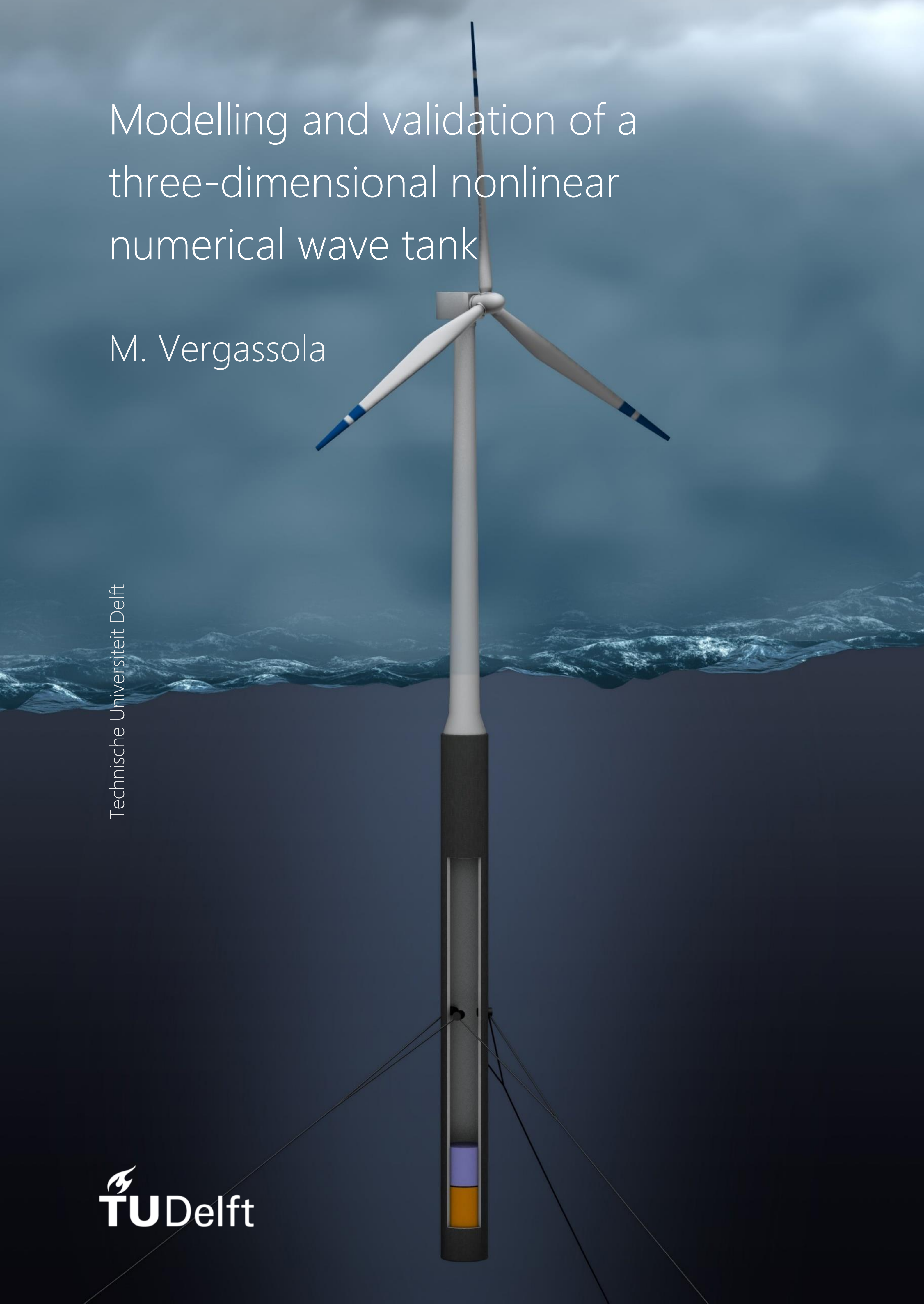


Modelling and validation of a three-dimensional nonlinear numerical wave tank

M. Vergassola

Technische Universiteit Delft

TUDelft

Modelling and validation of a three-dimensional nonlinear numerical wave tank.

by

M. Vergassola

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday July 5, 2018 at 12:30 PM.

Student number:	4630718	
Project duration:	September 1, 2017 – June 25, 2018	
Thesis committee:	Dr. Ing. R. Schmehl,	TU Delft
	Dr. Ir. A. C. Viré,	TU Delft
	Dr. Ir. A. Jarquin Laguna,	TU Delft
	MScEng. J. D. Brandsen,	TU Delft
Supervisors:	Dr. Ir. A. C. Viré,	TU Delft
	MScEng. J. D. Brandsen,	TU Delft

Cover picture retrieved from
https://www.statoil.com/en/news/archive/2008/05/22/hywindfullscale.html.

TU Delft

Preface

This master thesis represents the last step of a two year path here at TU Delft. I would like to thank my supervisors, Axelle Viré and Jaco Brandsen, for giving me support during this nine months project. There have been some very psychologically challenging moments and it is also thanks to your passion if I managed to overtake those moments and learn from them.

I would like to thank all my friend here in Delft. You have been my family for the last two years and I never felt alone because of that. A special thanks goes to Andrea, Filippo, Livia, Giacomo, Juan, Irene and Julia. A note is well deserved also by my friend Andrea, I left you behind in Genova but your support reached me every day, thank you.

Finally, I think my family deserve a personal note of thanks. This couldn't be possible without your support and, even though "I never call", I will always be grateful towards you.

M. Vergassola
Delft, June 2018

Abstract

Offshore wind energy is a fast-growing sector and it is quickly increasing its share in the European electricity mix. However, the costs are still high and the economic feasibility of offshore wind turbines is limited to shallow waters. This limitation plays a relevant role in the development of the sector and limits the offshore wind exploitation area to a small percentage of the potential one. In this scenario, floating support structures may represent a solution. Nevertheless, these type of support structures are not yet economically feasible as their design is challenging. In fact, with respect to bottom fixed structures, they present more complex dynamics and more degree of freedom. In order to cut the cost of floating support structures, it is necessary to be able to predict the wave- and wind-induced loads and motions of the floater and understand the coupling between them.

The aim of this project is to developed a three-dimensional numerical model of a floating cylinder in order to study the Fluid-Structure Interaction (FSI) of a spar-buoy support structure for offshore wind applications. The model is based on the coupled use of the CFD solver *fluidity*, to resolve the fluid dynamics, and an in-house python-developed code, to numerically solve the equations of motion of a rigid body in three dimensions. To achieve this goal, firstly, the python code is developed. Secondly, a Numerical Wave Tank (NWT) containing both air and water is generated and validated with both linear and nonlinear waves. Here, both an unstructured adaptive mesh and a simplex structured mesh are used to describe the domain. Finally, a free heave decay test of a floating cylinder is performed to investigate the accuracy of the model in computing the hydrodynamic coefficients of the floater. Here, the immersed-body method is used to model the presence of the body in the fluid domain.

The numerical wave tank developed in this work resulted to be quite accurate and capable of correctly describing both the linear and nonlinear waves propagation. The final model is developed with the use of a simplex structured mesh. In fact, mesh adaptivity resulted to be very challenging and the cost of its implementation exceeded the benefits. Also, a finite element based scheme including a Sweby slope limiter is used to limit advective fluxes in the setup of the NWT. Thanks to this, a less computationally demanding mesh could be used. Finally, the FSI analysis showed that, with the developed setup, the CFD solver is able to accurately predict the natural period of the floater but it underestimates the hydrodynamic damping. The cause of this was attributed to the use of the slope limiter aforementioned. In fact, it smooths the velocity field by means of numerical diffusion and this affects the resulting damping.

Contents

List of Figures ix

List of Tables xiii

1 Introduction 1

2 Floating wind turbines 5

2.1 FOWT concepts and challenges . . . 5

2.1.1 Wind turbine . . . 6

2.1.2 Support structure . . . 7

2.1.3 Mooring system . . . 8

3 Hydromechanics of floating rigid bodies 11

3.1 Potential wave theory . . . 11

3.2 Linear Airy wave theory . . . 12

3.3 Hydrostatic analysis . . . 15

3.4 Hydrodynamic analysis . . . 18

3.4.1 Hydrodynamic forces . . . 18

3.4.2 Dynamics of a floating rigid cylinder . . . 20

4 The rigid body code 29

4.1 Non-uniform rational B-splines . . . 29

4.1.1 The isogeometric concept . . . 29

4.1.2 NURBS as a geometry and mesh generator tool . . . 29

4.2 Code description . . . 32

4.2.1 Geometry . . . 32

4.2.2 Physics and time integration scheme . . . 34

4.3 Validation . . . 35

5 The CFD solver *fluidity* 39

5.1 Multi-material problems and governing equations . . . 39

5.2 Discretization . . . 40

5.2.1 Velocity and pressure discretization pairs . . . 44

5.3 Immersed-body method . . . 44

5.4 Adaptive remeshing . . . 45

5.4.1 Mesh optimisation . . . 46

5.4.2 Metric formation and norm selection . . . 47

5.4.3 Mesh interpolation . . . 48

5.4.4 Mesh adaptivity parameters . . . 48

6 Simulations and results 51

6.1 Multi-material interface tracking in a NWT . . . 51

6.1.1 Mesh adaptivity: tuning and results . . . 52

6.1.2 Interpretation of pressure and velocity profiles at the interface . . . 59

6.2 Validation of the 3-dimensional NWT for linear waves propagation . . . 60

6.2.1 Numerical setup . . . 60

6.2.2 Spurious velocity modes . . . 61

6.2.3 Results . . . 64

6.3 Further validation of the 3-dimensional NWT . . . 68

6.3.1 Numerical setup . . . 68

6.3.2 Results . . . 68

6.4 Free heave decay test 71
6.4.1 Numerical setup 71
6.4.2 The FSI algorithm 72
6.4.3 Results 73

7 Conclusions and recommendation 77
7.1 Conclusions. 77
7.2 Further Research 78

Bibliography 81

List of Figures

1.1 Global annual installed wind capacity 2001-2016 [8]. 1
1.2 Global cumulative installed wind capacity 2001-2016 [8]). 1
1.3 Wind energy resources [17] and bathymetric data [24] of Europe. 2

2.1 Three main floater concepts [35]. 6
2.2 A typical HAWT and VAWT [15]. 6
2.3 VAWT operation is here presented. In this figure, α is the angle of attack, U the wind velocity, ω is the rotational speed, L is the lift, D is the drag and N and T are normal and thrust force respectively. Differently from a HAWT, the angle of attack is periodically changing leading to cyclic forces as well [15]. 7
2.4 Quasi-static and dynamic models are compared showing that the two methods give different results when computing load peaks [15]. 9
2.5 Kinematic chain of elements used to describe the mooring line in dynamic mooring models [15]. 9

3.1 Particle trajectories and kinematics of a linear progressive wave with $H/L = 0.14$ [7]. 14
3.2 The figure shows how a wave train composed by two harmonic waves with slightly different phase propagates [7]. 15
3.3 Range of validity of different wave theories [7]. 16
3.4 Comparison between floating and submerged structures stability [7]. 17
3.5 Stability of a floater at large φ [7]. 17
3.6 Vertical pile in a wave field [7]. 20
3.7 The figure presents the loading regimes of the horizontal force acting on a vertical pile as function of relative wave height and relative size of the structure [7]. 21
3.8 Hydromechanical and waves forces are superposed to obtain the motions of the body in a wave field [14]. 22
3.9 Example of record obtain from a decay test. Here, the determination of the logarithmic decrement is presented [14]. 24
3.10 Rao and phase characteristic for heaving cylinder [14]. 26
3.11 The figure shows the three frequency areas related to the motions of the cylinder [14]. 27

4.1 Schematic illustration of NURBS basic concepts [12]. 30
4.2 Quartic basis functions for the knot vector $\Xi = \{0,0,0,0,0,1,2,2,3,3,3,4,4,4,4,5,5,5,5,5\}$. Note that the knot vector is open and non-uniform [12]. 31
4.3 Figure (a) shows a typical circle template. In (b) a template for an annulus is represented while in (c) the template used to generate the circular NURBS surface can be seen. 33
4.4 In (a), the cylindrical NURBS template is shown. In (b), the cylinder generated by the python code is presented. 34
4.5 Test case 1: displacement, velocity and acceleration computed by the new and the old code compared with the analytical solution. 36
4.6 Test case 1: relative error in the displacement prediction for both the old and new code. 37
4.7 Test case 1: relative error for both the old and new code. 37
4.8 Test case 1: relative error in the displacement prediction for both the old and new code. 38
4.9 Test case 7: Linear and angular displacement along x- and y-axis and about z-axis respectively. . 38

5.1 Schematic of piecewise linear discontinuous (a,c) and continuous (b,d) shape functions in one (a,b) and two (c,d) dimensions. In both cases the shape function as a value of 1 at the node A and decreases to 0 at the neighbouring nodes. In CG the support s is extended to the surrounding nodes while for DG it is restricted to the element area Ω_e. Modified figure from [3]. 41

5.2 In CG method, an arbitrary variable $u = u(x)$ is approximated globally in a continuous way. On the other hand, in DG method, the same variable is approximated globally in a discontinuous manner and locally in a continuous way. 42
5.3 An example of the schematic of piecewise linear (a,c) and quadratic (b,d) continuous shape functions in one (a,b) and two (c,d) dimensions [3]. 42
5.4 Schematic of the finite volume method in one (a) and two (b) dimensions [22]. 43
5.5 Difference between a finite volume mesh (a) and a control volume mesh (b) constructed over a finite element mesh (dashed lines in (b)) [22]. 43
5.6 Schematic of the computational domain for the immersed body method (left) and example of the fluid mesh containing a solid concentration field used to represent the body (right). Modified figure from [33]. 45
5.7 Graphic representation of the topological operations that are allowed in fluidity [3]. 46
5.8 Information transfer via consistent interpolation (top) and Galerkin projection (bottom). Modified figure from [3]. 49

6.1 Set-up and dimensions of the numerical wave tank used in this project to simulate wave propagation. 51
6.2 Two snapshots of the the longitudinal cross-section of the domain showing the adaptive remeshing for the case of single field adaptation and interpolation error of 8%. The figures show the mesh obtained after the first optimization cycle at $t = 0s$ (a) and at $t = 10s$ (b). 53
6.3 Three snapshots of the the longitudinal cross-section of the domain showing the adaptive remeshing for the case of multi-fields adaptation and interpolation error of 0.008 for the velocity field. The figures show the mesh obtained after the first optimization cycle at $t = 0s$ (a), at $t = 10s$ (b), at $t = 20s$ (c) and $t = 40s$ (d). Here the domain is zoomed to permit a better visualization of the mesh however, the absorption layer is consequently cut out from the picture. 54
6.4 Three snapshots of the the longitudinal cross-section of the domain showing the adaptive remeshing for the case of multi-fields adaptation and interpolation error of 0.05 for the velocity field. The figures show the mesh obtained after the first optimization cycle at $t = 0s$ (a), at $t = 6s$ (b) and at $t = 14s$ (c). Here the domain is zoomed to permit a better visualization of the mesh however, the absorption layer is consequently cut out from the picture. 55
6.5 Vertical particle velocity $3m$ below the air-water interface at $t = 40s$. 55
6.6 Figures (a,b), (c,d) and (d,e) present the time series of horizontal particle velocity, vertical particle velocity and dynamic pressure respectively. The set of figures (a), (c) and (d) shows velocity and pressure fields at the interface level in $P_1(0,0,0)$ while the set (b), (d) and (e) is extracted at $P_2(0,0,-1)$. Here, results from the multi-field adaptation simulations are compared with the linear Airy analytical solution. 56
6.7 Figures from (a) to (d) show the longitudinal cross-section of the mesh at $t = 39s$, $t = 99s$, $t = 123s$ and $t = 267s$ respectively. From these, it is possible to see how the mesh resolution nicely moves to the right while the wave field propagates in the domain. 57
6.8 This set of figures present the velocity (a,b) and pressure (c,d) profiles at the points $P1(0,0,0)$ (a,c) and $P2(30,0,-3)$ (b,d). 58
6.9 The material volume fraction at the interface level is presented in (b). Figure (a) shows the region of invalidity (grey region) of the solution computed at the interface and extracted with the use of detectors. 59
6.10 Set-up and dimensions of the numerical wave tank used in this project to simulate linear waves propagation. 60
6.11 Example of the mesh used to discretized the NWT. 61
6.12 Staggered arrangement of the unknowns obtained for pressure and velocity in the case of P_0P_{1CV}. 61
6.13 Figures (a) and (c) present the velocity field at $z = 0m$ and $z = -0.15$ respectively obtained with the coarse mesh. Figures (b) and (d) present the velocity field for the same water depths of (a) and (c) respectively but obtained with the fine mesh. 62
6.14 Figures (a,b,e,f) and (c,d,g,h) present the dynamic pressure and the horizontal particle velocity computed at $z = -0.025m$ and $z = -0.15$ respectively. Figures (a,b,c,d) are relative to what is extrapolated from the field at the inlet while figures (e,f,g,h) present results from a detector placed at $x = 10m$ from the wave-maker. Results obtained with the HyperC (green) and the FE (black) approach are compared with linear Airy (dashed red). 63

6.15 Two snapshots of the domain are here presented. Figure (a) shows the adimensional horizontal velocity while (b) presents the adimensional vertical velocity at $z = -0.5$. Both velocity components have been normalized with respect to the maximum of particle velocity components predicted by the linear wave theory. 64
6.16 Figures (a), (c) and (e) present the dynamic pressure computed at $0m$ from the wave-maker and at $z = 0m$, $z = -0.025m$ and $z = -0.15$ respectively. Figures (b), (d) and (f) present the pressure profile at $x = 20m$ from the inlet for the same water depths of (a), (c) and (e) respectively. Results obtained with the coarse (green), intermediate (blue) and fine (black) mesh are compared with linear Airy (dashed red). 65
6.17 Figures (a), (c) and (e) present the particle velocity computed at $0m$ from the wave-maker and at $z = 0m$, $z = -0.025m$ and $z = -0.15$ respectively. Figures (b), (d) and (f) present the velocity profile at $x = 20m$ from the inlet for the same water depths of (a), (c) and (e) respectively. Results obtained with the coarse (green), intermediate (blue) and fine (black) mesh are compared with linear Airy (dashed red). 66
6.18 Figures (a), (c) and (e) present the particle velocity computed at $60m$ from the wave-maker and at $z = 0m$, $z = -0.025m$ and $z = -0.15$ respectively. Figures (b), (d) and (f) present the dynamic pressure profile at $x = 60m$ from the inlet for the same water depths of (a), (c) and (e) respectively. The results (black) are compared with linear Airy (dashed red). 67
6.19 Set-up and dimensions of the numerical wave tank used in this project to simulate nonlinear waves propagation. 68
6.20 Two snapshots of the domain are here presented. Figure (a) shows the adimensional horizontal velocity while (b) presents the adimensional vertical velocity at $z = -0.5$. Both velocity components have been normalized with respect to the maximum of the particle velocity components predicted by the linear wave theory. 69
6.21 Figures (a,b,c,d) present the dynamic pressure (a,c) and the horizontal particle velocity (b,d) computed at distance $x = 5m$ from the wave-maker and at $z = 0m$ (a,b) and $z = -0.025$ (c,d). Figures (e,f,g,h) present the dynamic pressure and the velocity profile at distance $x = 10m$ from the inlet for the same water depths of (a,b) and (c,d) respectively. Results obtained with the CFD simulation (black line) are compared with the analytical solution provided by linear Airy (dashed-red line). 70
6.22 Set-up and dimensions of the numerical wave tank used in this project to simulate a heave decay test. 71
6.23 Two snapshots of the mesh are here presented. Figure (a) shows the longitudinal cross-section of the mesh while (b) presents a cross-section of the mesh at quote $z = 0m$. The mesh is relative to the case of $20p.p.d.$ but it is representative of both meshes as they differ only in the resolution. 72
6.24 The solid concentration field describing the floater in the domain is compared with the actual unstructured solid mesh. 73
6.25 Results obtained for the free decay test by using $15p.p.d.$ (blue-dashed line), $20p.p.d.$ (green-dashed line) and OpenFOAM (black line). 74

List of Tables

2.1 Assessment of floater concepts. 8

4.1 Summary of the 9 test cases performed to validate the python code. 35

6.1 Wave parameters used in the mesh adaptivity analysis. 52
6.2 Wave parameters used in the validation of the NWT. 60
6.3 Mesh characteristics based on wavelength and wave height. 60
6.4 Wave parameters used in the validation of the NWT. 68
6.5 Characteristics of the cylindrical floater. 71
6.6 Mesh characteristics based on the diameter of the floater. 72
6.7 Natural period, damping coefficient and hydrodynamic added mass and damping obtained from the free decay test are presented. The first three columns show the results from *fluidity* for the three different resolutions, the fourth columns presents the results obtained with OpenFOAM in [11] and the last column presents the period obtained via experimental free decay in [25]. 73
6.8 Error in the evaluation of natural period, damping coefficient and hydrodynamic added mass and damping with respect to the results obtained in [11]. 74

1 Introduction

Wind power is a promising electricity production player that is quickly increasing its share in the European electricity mix and it will meet the 4% of the European Union's total demand in 2020 and the 14% in 2030 [1]. The increasing size of wind turbines and wind farms in addition to the limited empty space and the more restrictive laws are pushing the wind industry to develop offshore. Offshore wind is nowadays a strong industry especially in the North Sea, however, the costs are still high and the economic feasibility of offshore wind turbines is limited to a water depth of around 50 m [15]. In fact, the most commonly adopted support structures, monopiles and jackets, are all bottom-mounted structures and they are economically limited to shallow water installations.

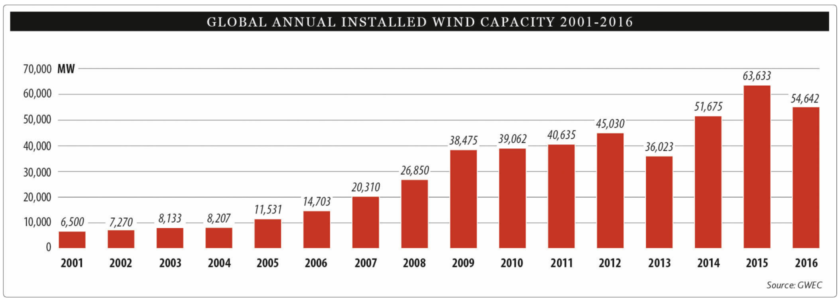


Figure 1.1: Global annual installed wind capacity 2001-2016 [8].

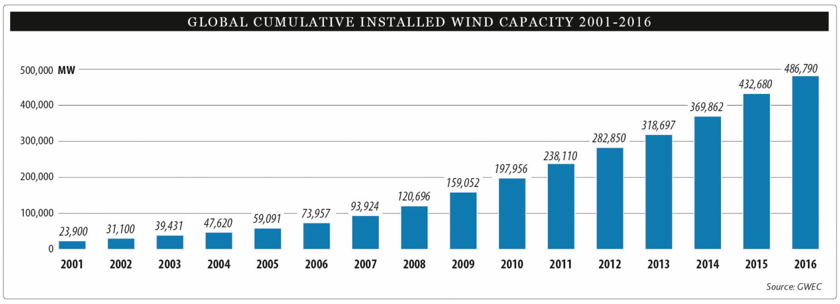


Figure 1.2: Global cumulative installed wind capacity 2001-2016 [8]).

This problem plays a relevant role in the development of the sector and limits the offshore wind exploitation area to a small percentage of the potential one. Moreover, it makes impossible for some countries to develop an offshore wind market although they may have wind resources and engineering knowledge to do so. Figure 1.3 shows that some of the most promising areas for wind energy exploitation, for instance in the Mediterranean sea and in the north of UK, are located in deep water areas.

In this scenario, floating support structures may represent a solution. Offshore oil and gas industry has already largely implemented floating support structures in the past decades in order to exploit deeper waters

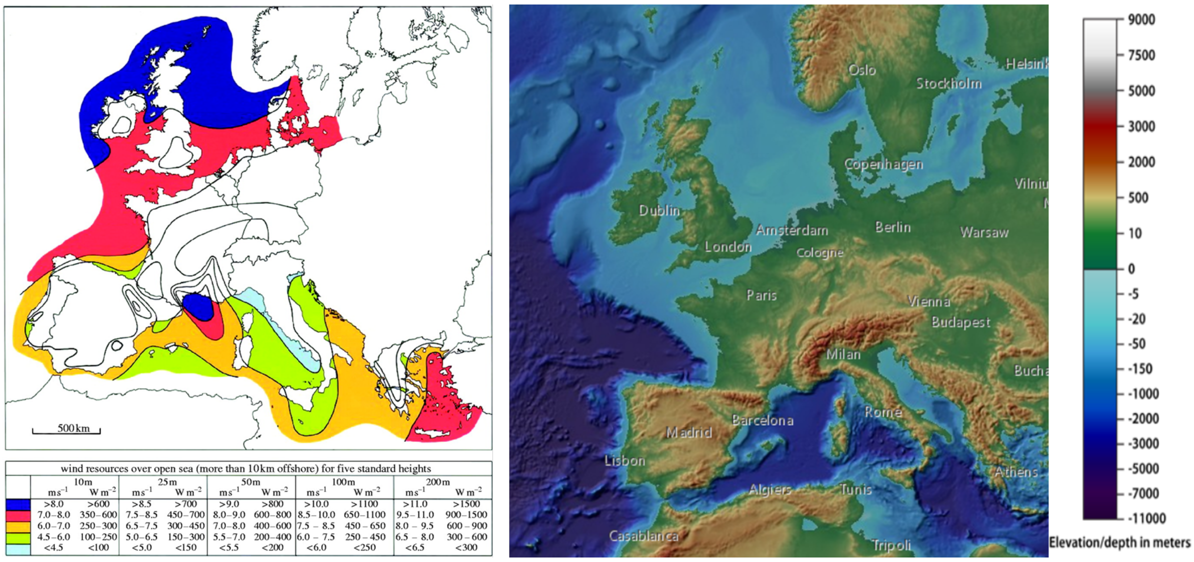


Figure 1.3: Wind energy resources [17] and bathymetric data [24] of Europe.

and now it seems time for the offshore wind industry to follow the same path [15]. Floating offshore wind turbines, however, are not yet economically feasible as their development is challenging. With respect to a fixed structure, they present different and more complex dynamics due to the fact that the support structure is moving because of waves and wind. For the same reasons, also the control system needs to be more sophisticated and able to deal with these wind-wave-structure interactions [15]. In this scenario, it is clearly necessary to be able to compute the wave-induced loads and motions of the floater [14] and the coupling between the formers and the wind-induced loads and motions.

Wave Structure Interaction (WSI) problems for both fixed and floating bodies are, in the simple cases, well-known problems as they have been studied for a long time. For simple problems, it is usually possible to analytically solve a boundary value problem assuming the fluid to be inviscid and applying potential flow theory [2]. However, for more complex cases, numerical methods, such as Boundary Element Methods (BEMs) and Finite Element Methods (FEMs), need to be applied in order to solve the WSI problem. BEMs are widely used at industry level because, in their linearised form, they work in the frequency domain and are, therefore, not computationally demanding [32]. Nevertheless, whenever transient phenomena and non-linearities are important, methods which work in the time domain, such as FEMs, have to be applied [32]. For these reasons, non-linear methods are increasing their appeal and Computational Fluid Dynamics (CFD) models have begun to be more frequently applied especially for extreme waves-structure interaction calculations where the hypotheses of irrotational and inviscid fluid are no longer acceptable [32].

The purpose of this Master thesis is to develop a three-dimensional numerical model of a floating cylinder in order to study the fluid-structure interaction of a spar-buoy support structure for offshore wind applications. The starting point of this project is represented by the work of the MSc. M. Baudino Bessone [2] who developed a two-dimensional model of a floating cylinder during his Master thesis at the Delft University of Technology. In order to solve the wave-structure interaction problem, the CFD solver *fluidity* is coupled with an in-house python-based code capable of numerically solving the equations of motion of a rigid body. The immersed-body approach is used to simulate the presence of the floater in the fluid domain through a solid concentration field and a penalty term [34]. In order to allow motions of the floater, a multi-material approach is used to generate a domain which contains both air and water. In this context, the potential of the mesh optimization library, available in *fluidity*, as a tool to track the air-water interface in a Numerical Wave Tank (NWT) is explored. The NWT is also validated with the use of a simplex structured mesh that is then used to perform the fluid structure interaction analysis as well.

In order to achieve the final goal of this project, firstly, the python code used in [2] and developed by MScEng. J.D. Brandsen from Delft University of Technology needs to be extended in order to numerically solve the equations of motion for a six degree of freedom body. Secondly, a Numerical Wave Tank (NWT) containing both air and water is generated and waves propagation through it is compared with linear Airy wave theory. Finally, the floater is introduced in the domain and the accuracy of the solver in computing the hydrodynamic damping and the added mass of the body is evaluated via free decay test. Research questions are formulated for each step of the project:

- How accurate is the solution of the equations of motion computed by the python code for a six degree of freedom rigid body subjected to external loads and constrained only by a spring and a damper?
- How efficient and precise is the adaptive remeshing in capturing the air-water interface? What is the cost of mesh adaptivity and what are the main issues related to its implementation?
- For the case of a three-dimensional NWT containing both air and water, discretized with $P0-P1_{CV}$ and represented by a simplex structured mesh, how precise is the waves propagation predicted by the CFD solver when compared to linear Airy wave theory? And what is the accuracy of the solver when nonlinear waves are generated?
- How accurate are the hydrodynamic damping and added mass computed by the solver performing a heave free decay test? To which extend is the accuracy of the solver affected by the presence of a mooring line?

The report is organized as follow. Section 2 introduces the concept of floating wind turbine and the related design challenges. Section 3 deals with wave theories and hydromechanics of floating structures. In section 4 the python-developed rigid body code is described and validated. Section 5 describes the CFD solver *fluidity*, introduces the discretization adopted in this work, presents the immersed body method and briefly analyses the functioning of the optimization routine performed by *fluidity* when adaptive remeshing is used. Finally, Sections 6 and 7 present the results of the simulations and conclusions respectively.

2

Floating wind turbines

This work aims to develop a numerical model of a Floating Offshore Wind Support Structure (FOWSS). Thus, an overview of the main components of such a device and the related engineering challenges is necessary. Firstly, a general description of Floating Offshore Wind Turbine (FOWT) concepts is provided together with the main challenges that this novel technology brings into the design process. Finally, the main components of a FOWT are presented and different design concepts are analysed and compared. Only a brief overview of FOWTs is presented in the following sections, thus, for a complete and in-depth literature review, the reader is referred to [15], [27], [21].

2.1. FOWT concepts and challenges

Offshore wind energy is a fast-growing sector, however, its development is limited to shallow water areas. In fact, the commonly used support structures, Jacket, Monopile and Gravity Based (GB), are economically feasible only for water depths smaller than around $50m$. This limitation is mainly due to the fact that for an increasing water depth, the entire support structure needs to be increased in dimensions. In this sense, FOWTs can represent a solution as the dimensions of the floating support structure are not relevantly affected by the water depth since only the mooring design is sensitive to it [15]. The reason why the cost of the floater is mainly indifferent to water depth is that, while for bottom-mounted structures loads are transferred to the soil through a rigid structure causing bending loads around the mudline that increase with the water depth, for floating structures the wind turbine loads are transferred to the water. This leads also to two benefits, (1) because the water is closer to the turbine with respect to soil, the bending moments are extremely reduced and (2) as water is compliant, peaks in the loads are generally lower [15]. This characteristic of FOWSSs may also lead to other advantages such as an easier design and installation process for all those areas which suffer from a great variation in water depth and, in general, an easier standardization of the production chain of the floater [15]. However, as is typical for new technologies, FOWSSs come along with some new challenges with respect to bottom-mounted support structures [27]. First of all, the use of a floating structure leads to a larger number of Degree of Freedom (DoF) and, in general, to larger motions. Secondly, a mooring system is required in order to keep the floater in position and to generate a restoring force which opposes the wave drift force [15]. Finally, as the turbine is subjected to wind forces and these loads are transferred to the floater, the development of a control system that is capable of dealing with this wave-wind-structure interaction is needed and its complexity poses important challenges which may affect the entire design process [27].
Out of the great number of floater concepts which has been designed, three main foundation types can be identified, Spar, Semi-submersible and Tension Leg Platform (TLP). This classification arises from the way in which these different floaters react to the overturning moment due to the thrust force acting on the wind turbine and achieve stability [15]:

- **Spars**, use gravity to counteract the external forces,
- **Semi-submersibles**, rely on distributed buoyancy,
- **TLPs**, make use of taut moorings to keep the floater stable.

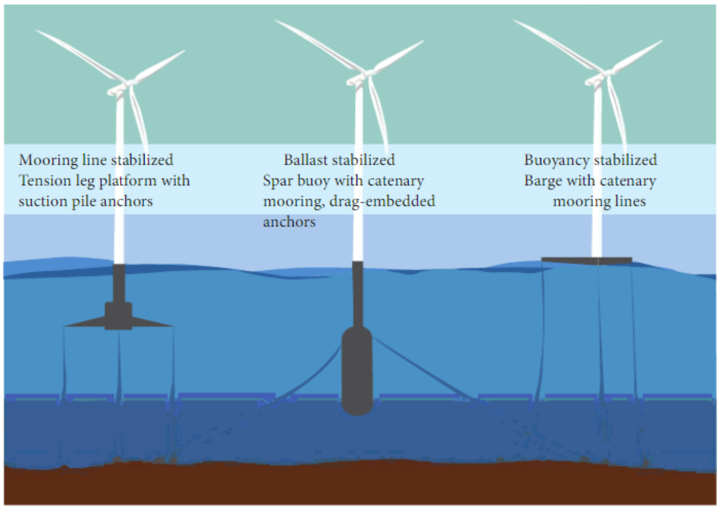


Figure 2.1: Three main floater concepts [35].

In general, a FOWT can be divided in three sub-components, moving from top to bottom, the wind turbine, the support structure (or floater or platform) and the mooring system. In the next sections, each component is individually briefly described and the main features regarding the design of each of these elements are outlined.

2.1.1. Wind turbine

Wind turbines can be divided into two main concepts, Horizontal Axis Wind Turbines (HAWTs) and Vertical Axis Wind Turbines (VAWTs). Despite the fact that HAWTs are a well-established technology and are largely adopted in offshore wind applications based on bottom mounted support structures, VAWTs can play an important role in the development of FOWTs.

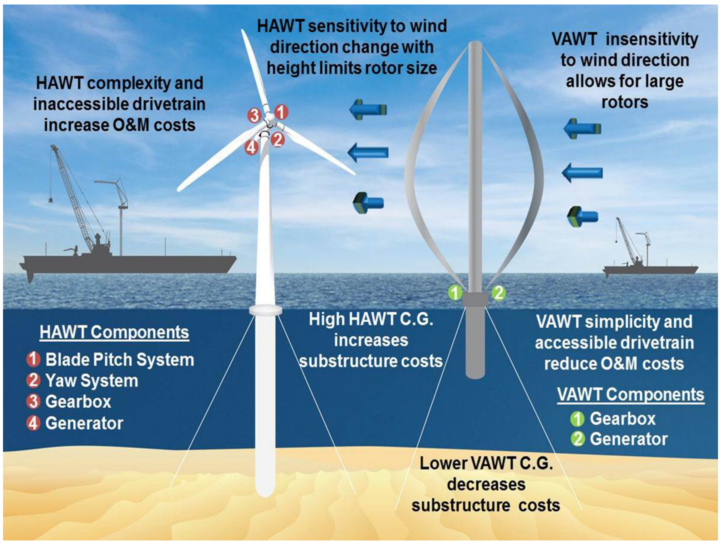

HAWT sensitivity to wind direction change with height limits rotor size
VAWT insensitivity to wind direction allows for large rotors
HAWT complexity and inaccessible drivetrain increase O&M costs
3
1
4
2
HAWT Components
1 Blade Pitch System
2 Yaw System
3 Gearbox
4 Generator
High HAWT C.G. increases substructure costs
1
2
VAWT simplicity and accessible drivetrain reduce O&M costs
VAWT Components
1 Gearbox
2 Generator
Lower VAWT C.G. decreases substructure costs

Figure 2.2: A typical HAWT and VAWT [15].

The two concepts are very different (see figure 2.2) and present some advantages and disadvantages with respect to floating offshore wind implementations. By looking at Figure 2.3, which describes the operation of a VAWT, it can be understood how from an aerodynamic point of view, VAWTs present some weaknesses with respect to HAWTS [15]. Firstly, VAWTs do not continuously operate at the optimum angle of attack, α in the figure, leading to an oscillating thrust force, T, with maxima that can reach largely higher values than for HAWTs. Moreover, this intrinsic oscillatory nature of the aerodynamic loads increases fatigue issues [21]. However, fatigue is an important topic also for HAWTs as they suffer from gravitational fatigue problems, especially for large rotors.

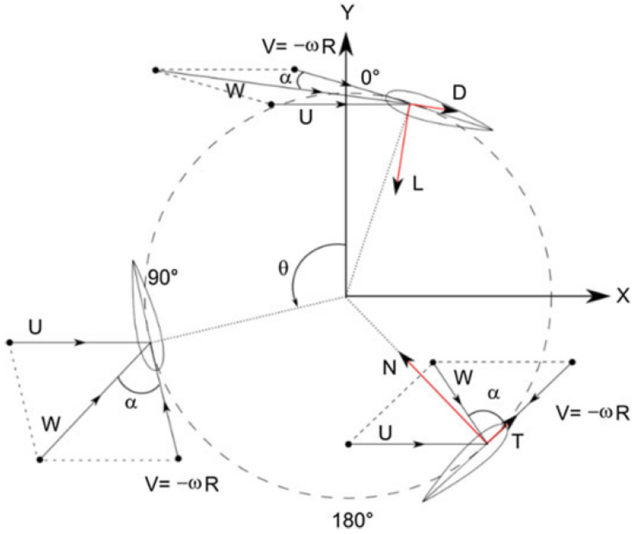


Figure 2.3: VAWT operation is here presented. In this figure, α is the angle of attack, U the wind velocity, ω is the rotational speed, L is the lift, D is the drag and N and T are normal and thrust force respectively. Differently from a HAWT, the angle of attack is periodically changing leading to cyclic forces as well [15].

With respect to the aerodynamic behaviour of the two concepts, the main advantage of VAWTs lies in the lower position of the centre of thrust pressure [21]. This result leads to a much lower overturning moment for VAWTs than for HAWTs. This is very important while dealing with floating structures as the overturning moment caused by wind loads is a major aspect which drives the design of the floater. Thus, VAWTs should need a smaller and consequently cheaper platform [15]. Another potential advantage of VAWTs lies in the conversion efficiency of this concept for skewed flow conditions [15]. In fact, because the floater is not fixed but oscillates, a FOWT often operates in a non-upright condition which is a problem for HAWTs but not for VAWTs. Actually, for the case of H-type VAWTs, a skewed flow condition can increase the performance of the turbine as a part of the blade in the downwind cycle will no longer be affected by the upwind blade, leading to a higher quantity of energy being captured by the turbine [15] [20]. Lastly, the position of the drive train plays an important role and significantly differentiates a HAWT from a VAWT. In fact, while for the former it is allocated at the hub height, generating large bending moments and motions of the tower, for the latter it is at the bottom, requiring a smaller platform. An important result which derives from the position of the drive train is that for VAWTs the Centre of Gravity (CoG) of the structure is lower than for HAWTs, leading to better stability as explained in Section 3.3.

2.1.2. Support structure

The goal of the support structure is to counter and transfer the environmental loads while retaining stability and supporting the turbine. Going a bit more in depth, the loads that a floater experiences are [15]:

- **Wind loads**: loads applied on both the rotor and the tower including both the dynamic and mean components need to be considered. The main source of stress is represented by the thrust force.
- **Wave loads**: these forces and moments are caused by the incident wave train and the motions of the floater (plus turbine) itself and they have both a static part, the drift force that is mainly countered by the mooring, and a dynamic part. Wave loads will be discussed in more detail in Chapter 3.
- **Gravitational loads**: loads caused by the weight of the platform (plus turbine) itself. For a floating body, they may be strongly dependent on the motions of the entire structure.
- **Environmental loads**: these represent loads induced by currents, tides, wind-waves misalignment and ice.

As beforehand introduced in section 2.1, the thrust force and the related overturning moment is by far the highest load acting on the support structure. Thus, platform concepts are usually classified depending on how they react to this load and achieve stability [15]. Based on this criteria, three main classes are defined, Spar Class, TLP Class and Semi-Submersible Class. Each of these classes presents advantages and disadvantages and, so far, none of them is clearly superior. However, with respect to technological maturity of the design, the advantage of spar and semi-submersible structures lie in the fact that they are both based on well-known technologies that have already been largely implemented in the offshore oil and gas sector.

A relevant topic while dealing with floaters is the analysis of the dynamic response of the support structures itself. In fact, differently from what someone could expect, in general, a good design can reduce the amplitude of the motions experienced by a platform to the same order of magnitude of those exhibited by bottom-mounted structures. This is because very high waves have long periods resulting in slow motions of the platform and waves with smaller periods do not excite the floater in a severe way as the wavelength is similar to the dimensions of the structure [15].

A proper feature of spar and semi-submersible classes is that they achieve stability by making use of the heeling restoring forces. This results in a wind turbine which constantly operates in a tilted position which can lead to a loss of produced power. While semi-submersibles and TLPs can be adopted also in shallow water conditions, spar floaters need a higher water depth because of the extremely deeper draft required to counter the overturning moment [15]. Another relevant difference between spar and semi-submersible with respect to the TLP concept lies in the use of mooring lines. These are slack and mainly used to resist drifting for the former concepts, but they are taut and fully involved in the stability of the floater for the latter. Finally, Table 2.1 summaries the main features of these three different concepts.

Table 2.1: Assessment of floater concepts.

Concept	**Spar**	**TLP**	**Semi-submersible**
Stability	Gravity (in the form of ballast)	Moorings	Buoyancy (hydrostatics)
Min. water depth	Deeper	Shallower	
Fabrication	Simple structure, similar to a monopile	Complex structure	
Installation	Complex operation		Good
Decommissioning	Potentially simpler than bottom-fixed support structures		

The complexity of the installation differs depending on the concept. For spars, the main issues are related to the transportation and upending of the structure while for TLP, the challenges lie in the anchoring process as a TLP should not move vertically with the waves [15]. In this sense, the TLP also represents the platform concept which exhibits behaviour that most resembles bottom-fixed support structures [15]. In this project, a spar-type floater is used to validate the numerical wave tank in Section 6.4. This choice is mainly due to the simpler geometry of this type of support structure.

2.1.3. Mooring system

Mooring lines are necessary for every concept of platform, however, while for spars and semi-submersibles they only resist the mean drift force generated by the incident waves, they play a major role in TLP stability. In general, mooring lines and anchoring design should aim to develop a reliable station-keeping system which is able to resist the environmental loads and possesses a natural frequency that is outside the wave frequency range [15]. The design of a station keeping system usually starts from the anchoring selection which depends on the soil holding capacity. Some of the different types of anchoring systems are drag embedment anchors, pile and screw anchors, plate anchors and gravity anchors. The number of anchors needed to resists the design environmental loads is determined by the bearing capacity of the soil [15].

Mooring lines may be made out of a combination of chains, wire ropes and fibre ropes. The advantage of introducing fibre ropes consists mainly of their neutral buoyancy which allows achieving deeper waters [15]. In fact, although chains have been extensively used in offshore engineering, if mooring lines are made of only chains, they suffer from self-weight related problems for deep waters [15]. However, even though fibre ropes seems promising for floating offshore wind applications, they suffer from creep and non-linear stiffness introducing new challenges into the mooring lines design [36].

Because in this work a single mooring line will be used in the heave decay test to asses the influence of a mooring line on the accuracy of the CFD solver (see Section 6.4), it is relevant to see how a mooring line can be modelled. Generally speaking, two main modelling approaches are possible, a quasi-static model and a dynamic model. The major limitation of the quasi-static model is its inability to capture the interactions between mooring line and surrounding environment and the underestimation of load peaks [15], as shown in figure 2.4.

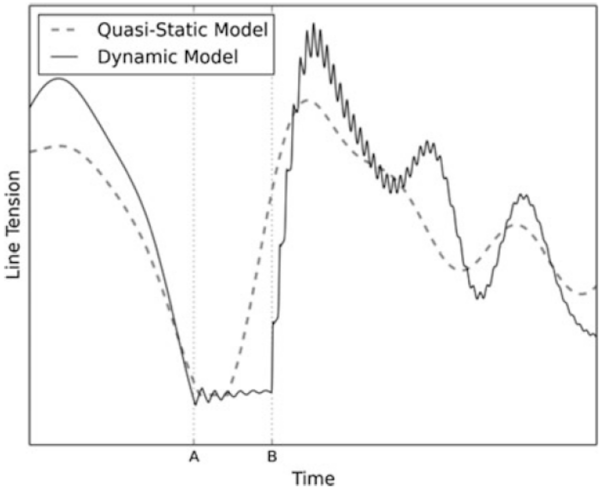


Figure 2.4: Quasi-static and dynamic models are compared showing that the two methods give different results when computing load peaks [15].

A quasi-static model can be formulated in different ways, however, the simplest approach is to describe the mooring line as a linear spring which follows Hooke's law [15]. The equation that describes the restoring force of the mooring line is

$$\mathbf{F} = \mathbf{K}\mathbf{x} \tag{2.1}$$

where **x** is the generalized floater displacement, **K** is the linearised stiffness matrix and **F** is the restoring force matrix. In general, F will be a $N \times N$ matrix where N is the number of DoF of the system.
For a dynamic mooring model, the mooring line is described by a kinematic chain of elements (see Figure 2.5). In general, three main groups of dynamic models can be identified: lumped-mass models, finite element models and finite-difference models. The governing equation of these dynamic models can in general be expressed as [15]

$$\mathbf{M}_i \ddot{\mathbf{r}}_i = \sum f_{i_{ext}} + \sum f_{i_{int}} \tag{2.2}$$

where $\mathbf{M}_i$ stands for the mass matrix of the chain element i, $\ddot{\mathbf{r}}_i$ is a vector containing the acceleration at the nodes of the element i and $\sum f_{i_{ext}} + \sum f_{i_{int}}$ represents the summation of external and internal forces acting on the element.

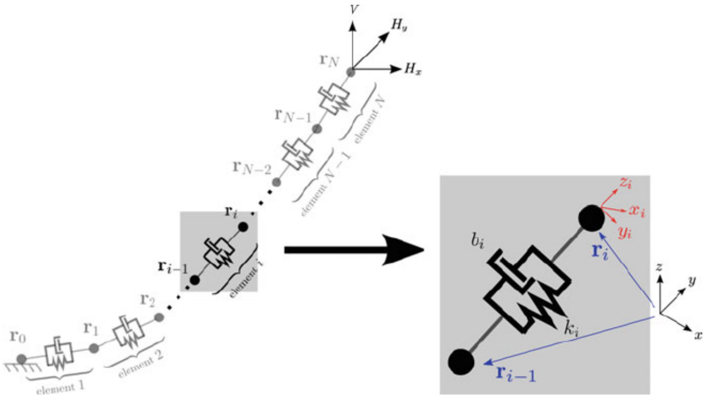


Figure 2.5: Kinematic chain of elements used to describe the mooring line in dynamic mooring models [15].

Electrical connections are not part of the mooring system, as they do not provide any help in maintaining stability and avoiding drift of the platform. However, for completeness, their main design challenges are still introduced here. Because of the larger motions of floating platforms with respect to bottom-fixed support structures, the electrical cabling is subjected to significant dynamics [2]. Moreover, as FOWTs are designed to be implemented in deeper waters, the location of these may be quite far from shore with respect to most of the offshore wind farms that have been installed. These can cause severe losses due to power transmission which has to be solved by using DC cables in order to transport the electricity to shore. The use of DC transmission, however, can significantly increase the cost of the electrical infrastructure [16].

3

Hydromechanics of floating rigid bodies

In this chapter, both hydrostatic and hydrodynamic analysis are treated together with wave theory. The comprehension of waves related phenomena is of great importance while analysing the dynamic behaviour of offshore structures. Thus, in Section 3.1 potential wave theory is briefly introduced together with some basic concepts of hydromechanics. In Section 3.2, out of the many wave theories that have been developed, such as linear Airy, Stokes, Dean's stream function, only the first order linear Airy wave theory is treated as this work will mainly deal with linear waves. Once that the hydrodynamic characteristics of surface waves have been introduced, Sections 3.3 and 3.4 will deal with the hydrostatic and hydrodynamic analysis of floating structures. In Section 3.4, only hydrodynamically transparent structures are considered as this is the condition in which the analysed floater falls. In this chapter, only what is relevant for this work is treated, thus, for a more in-depth literature review regarding wave theories and hydromechanics the reader is referred to [7] and [14].

3.1. Potential wave theory

Potential wave theory is based on the assumption that the fluid of interest is incompressible, irrotational and inviscid. Thus, gradually introducing these assumptions, the governing equation and the boundary conditions related to potential waves are derived. For an incompressible fluid of density ρ, the mass flow balance of a volume element $dV = dxdydz$ results in [7]

$$\nabla^T \mathbf{v} = 0 \tag{3.1}$$

with $\mathbf{v}$ being the velocity vector. If now an irrotational fluid is considered and the scalar function known as **velocity potential** Φ is introduced [7]

$$\mathbf{v} = \nabla\Phi \tag{3.2}$$

by combining equation 3.1 and 3.2, the **Laplace differential equation** is obtained as

$$\nabla\nabla\Phi = \nabla^2\Phi = 0 \tag{3.3}$$

If viscosity and the related stress-strain effect may be neglected, the Navier-Stokes equation takes the simplified form known as **Euler equation** [7]

$$\rho\frac{d\mathbf{v}}{dt} = \rho\left[\frac{\partial\mathbf{v}}{\partial t} + (\mathbf{v}^T\nabla)\mathbf{v}\right] = -\nabla(p + \rho g z) \tag{3.4}$$

where the total acceleration has been decomposed in its local, $\frac{\partial \mathbf{v}}{\partial t}$, and convective, $(\mathbf{v}^T\nabla)\mathbf{v}$, components and g stands for the gravitational acceleration. Substituting the local acceleration with the velocity potential, the Euler equation can be integrated along a streamline resulting in the **Bernoulli equation** [7]

$$\rho\frac{\partial \Phi}{\partial t}+\frac{\rho}{2}|\mathbf{v}|^2+p+\rho g z=p_0 \tag{3.5}$$

where p_0 is an integration constant which is usually assumed to be equal to the atmospheric pressure p_0 on the free surface [7].
In case of **free surface waves**, at seabed and free surface some boundary conditions need to be satisfied. Here, a limiting surface S, to which fluid particles are bound, is introduced and it can be described by $S(x,y,z,t)=const$. For a 2-dimensional flow, the surface S is a streamline, in fact, if S is moving it has to satisfy $S(x+dx,y+dy,z+dz,t+dt)=const$ which implies

$$\frac{dS}{dt}=0 \tag{3.6}$$

At the seabed, $z=-d$, where d represents the water depth, $S:=z+d=0$ and

$$\frac{dS}{dt}=\frac{\partial S}{\partial t}\frac{dz}{dt}=\frac{dz}{dt}=0 \quad at \quad z=-d \tag{3.7}$$

which means that the normal component of the velocity, w, has to be equal to zero [7]. As already introduced, in order to describe the time-dependent contour of the free surface, the wave profile, a kinematic and a dynamic boundary conditions have to be satisfied [7]. First of all, the wave profile is defined as $z=\zeta(x,y,t)$ leading to $S:=\zeta(x,y,t)-z=0$, where ζ represents the wave elevation. Now, starting from equation 3.6, the **kinematic boundary condition** is [7]

$$\frac{dS}{dt}=\frac{\partial \zeta}{\partial t}+\frac{\partial \zeta}{\partial x}u+\frac{\partial \zeta}{\partial y}v-w=0 \quad at \quad z=\zeta \tag{3.8}$$

The dynamic boundary condition derives from the imposition of $p=p_0$ at every point of the free surface [7], from which the Bernoulli equation 3.5 becomes

$$\rho\frac{\partial \Phi(x,y,\zeta,t)}{\partial t}+\frac{\rho}{2}\left[\left(\frac{\partial \Phi}{x}\right)^2+\left(\frac{\partial \Phi}{y}\right)^2+\left(\frac{\partial \Phi}{z}\right)^2\right]_{z=\zeta}+\rho g\zeta(x,y,t)=0 \tag{3.9}$$

By imposing these boundary conditions and making use of the Laplace equation, Φ, $\zeta(x,y,t)$, dynamic pressure and particle velocity and acceleration can be determined over the entire domain [7]. However, since these conditions are non-linear and ζ is unknown, the problem does not present a closed analytical solution [7]. Fortunately, approximate solutions for velocity potential and free surface elevation can be achieved by power series development of the **wave steepness**, $\frac{wave\ height\ H}{wavelength\ L}$, for deep water waves and **relative wave height**, $\frac{wave\ height\ H}{water\ depth\ d}$, for shallow water waves condition [7]. The first order linear wave theory based on series development in powers of H/L is known as **Airy wave theory** and it will be described in Section 3.2.

3.2. Linear Airy wave theory

The condition under which linear Airy wave theory is applicable is that water depth and wavelength are so big if compared to wave amplitude that the free surface boundary conditions are approximated also at the

wave contour, $z = \zeta = 0$ [7]. Moreover, the wave surface is assumed to be unchanged along the y-direction, i.e. the transverse direction with respect to the wave development, so that the flow is actually two-dimensional. This condition is referred to as long-crested waves.
The boundary condition at the seabed is exactly satisfied while for the free water surface boundary condition, only a generalized surface condition is enforced [7]. The bottom condition is the same as in 3.7

$$w = \frac{\partial \Phi}{\partial z} = 0 \quad at \quad z = -d \tag{3.10}$$

In order to derive the generalized surface condition at the wave contour, only the linear terms of 3.8 and 3.9 are considered [7]

$$\frac{\partial \zeta}{\partial t} - \frac{\partial \Phi}{\partial z} = 0 \tag{3.11}$$

$$\frac{\partial \Phi}{\partial t} + g\zeta = 0 \quad at \quad z = -d \tag{3.12}$$

By derivation of 3.12 and subsequent substitution of 3.11 into 3.12, the generalized condition is obtained as

$$\frac{\partial^2 \Phi}{\partial t^2} + g\frac{\partial \Phi}{\partial z} = 0 \quad at \quad z = -d \tag{3.13}$$

Now, from the Laplace's equation and the boundary conditions, Φ can be computed by separation of variables [7]. The partial sinusoidal solution as function of x and t is expressed as

$$\Phi = \frac{\zeta_a g}{\omega} \frac{coshk(z+d)}{coshkd} sin(kx - \omega t) \tag{3.14}$$

where ω is the **circular velocity** which is defined as

$$\omega = \frac{2\pi}{T} \tag{3.15}$$

and k represents the **wave number**

$$k = \frac{2\pi}{L} = \sqrt{kgtanhkd} \tag{3.16}$$

From equation 3.15 and 3.16 a relation between wave number and circular velocity can be obtained and this is known as **linear dispersion**

$$\omega = \sqrt{kgtanhkd} \tag{3.17}$$

Equation 3.14 is valid for every water depth condition, deep, intermediate and shallow, however, for deep water and shallow water conditions some approximations can be introduced [7]. Finally, from the computed velocity potential and the linearised dynamic surface condition, the wave profile can be expressed as [7]

$$\zeta = \zeta_a cos(kx - \omega t) = \zeta_a cos\theta \tag{3.18}$$

where ζ_a and θ stand for wave amplitude and phase respectively.
A closer look at particle velocities and particles path will show how these water particles travel in closed elliptical orbits. Knowing the velocity potential, particle velocities, u and w, can be obtained through a differentiation of Φ with respect to x and z respectively. Once u and w are known, each particles path can be obtained by integration of particle velocities about the mean particle position $M(x_0, y_0)$ [7]

$$x - x_0 = \int \frac{\partial \Phi}{\partial x} dt = -\zeta_a \frac{coshk(z+d)}{sinhkd} sin\theta \tag{3.19}$$

$$z - z_0 = \int \frac{\partial \Phi}{\partial z} dt = \zeta_a \frac{sinhk(z+d)}{sinhkd} cos\theta \tag{3.20}$$

which leads to

$$\left[\frac{x - x_0}{\zeta_a \frac{coshk(z+d)}{sinhkd}}\right]^2 + \left[\frac{z - z_0}{\zeta_a \frac{sinhk(z+d)}{sinhkd}}\right]^2 = 1 \tag{3.21}$$

from which it is clear why the particle path is elliptical. Moreover, it can be seen that in the special case of deep water condition the orbits are circular and present a radius which exponentially decreases with depth, i.e. the particles motion is reduced for increasing depth [7]. Figure 3.1 clearly shows what just stated.

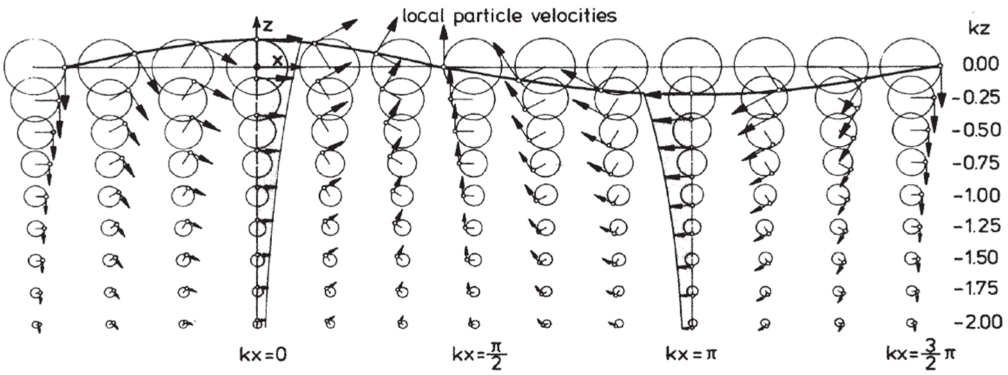


Figure 3.1: Particle trajectories and kinematics of a linear progressive wave with $H/L = 0.14$ [7].

From figure 3.1, which presents a progressive wave, it can also be observed that while water particles move in elliptical orbits about their mean position, the wave translates horizontally with a **phase velocity**

$$c = \sqrt{\frac{g}{k} tanhkd} \tag{3.22}$$

which corresponds to the velocity of a moving observer who sees an unchanged wave profile. This phase velocity, also known as **celerity**, however, is, apart for shallow water conditions, different from the wave energy velocity, or **group velocity**, c_{gr}. By superimposing two waves with a slightly different phase and by analysing the resulting wave train phase, the velocity of the wave envelope, the group velocity, can be derived through the dispersion equation 3.17[7].

$$c_{gr} = \frac{c}{2}\left[1 + \frac{2kd}{sinh2kd}\right] \tag{3.23}$$

From the definition of c_{gr}, it can be seen that in general singular waves propagate faster than the wave envelope. Particular cases are represented by shallow water, where $c_{gr} = c$, and deep water, where $c_{gr} = c/2$. The difference between c and c_{gr} is of relevance for fluid-structure interaction problem in presence of a wavemaker. In fact, as the wavefront advances at group velocity, it will reach the obstacle later than the phase[7]. Figure 3.2 shows the propagation of a wave train as previously described.

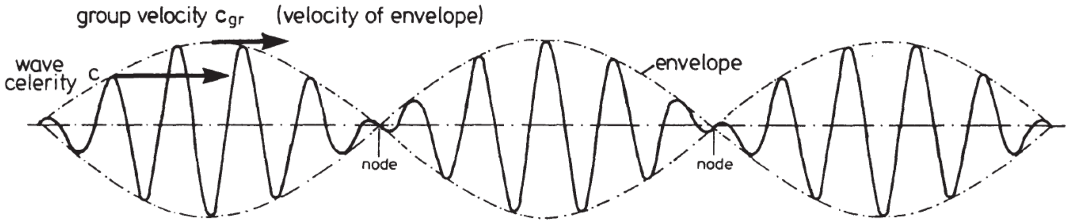


Figure 3.2: The figure shows how a wave train composed by two harmonic waves with slightly different phase propagates [7].

Finally, it is relevant to analyse the range of applicability of Airy wave theory in order to identify the limitations that such a linear theory introduces. In this sense, figure 3.3 presents the region of validity of different wave theories using a relative wave height, H/gT^2, and a relative water depth, d/gT^2, as limiting parameters.

3.3. Hydrostatic analysis

In this section, some hydrostatic concepts which are relevant for this work are introduced and briefly discussed. Thus, two main topics are treated here: **buoyancy** and **stability** of floating offshore structures.
First of all, an important characteristic that distinguishes floating structures from the typical bottom mounted ones is that the atmospheric pressure, p_0, can be ignored as it equally acts on all surfaces. Thus, the resultant force acting normal to a surface dS is,

$$d\mathbf{F} = -p_h \mathbf{n} dS \tag{3.24}$$

where p_h is the hydrostatic pressure and **n** is the normal vector of dS. From equation 3.4, by neglecting the convective velocity term, it is possible to obtain a linearised form of the Bernoulli equation 3.5, from which dynamic and hydrostatic pressure terms can be identified [7].

$$p - p_0 = -\rho\frac{\partial\Phi}{\partial t} - \rho g z \tag{3.25}$$

$$p_h = -\rho g z \tag{3.26}$$

$$p_{dyn} = -\rho\frac{\partial\Phi}{\partial t} \tag{3.27}$$

Introducing 3.26 into 3.24 and integrating over the surface S, the buoyancy force acting on a floating structure is derived as follow [7]

$$\mathbf{F} = \rho g \int_{(V)} (\nabla z) dV = (0, 0, \rho g V)^T \tag{3.28}$$

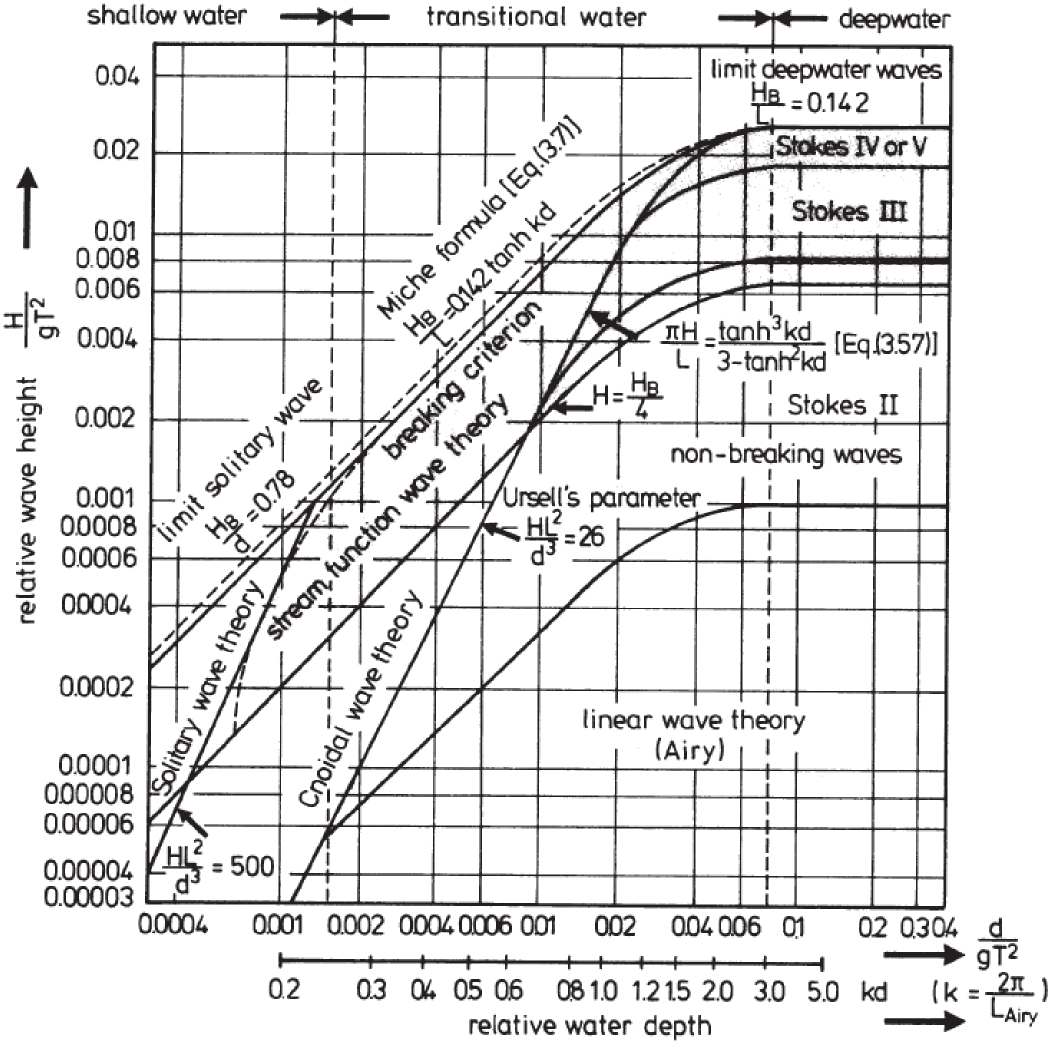


Figure 3.3: Range of validity of different wave theories [7].

As already described by Archimedes, this static vertical force corresponds to the weight of the displaced fluid, is pointing upwards and is acting at the centroid of the displaced volume, also known as the center of buoyancy, $B_0(x_{B_0}, y_{B_0}, z_{B_0})$. The buoyancy force opposes the weight and its line of action passes through the center of gravity of the body. In a condition of static equilibrium, these two forces are aligned and equal in magnitude and the structure is neutrally buoyant. However, when the floater rotates of an angle φ under the effect of a load, the geometry of the displaced volume changes and so also the position of its center of buoyancy. The line of action of the new buoyancy intersects the previous one in a point called metacenter M_0 [7]. As a consequence, the line of action of the new buoyancy is displaced with respect to the weight action line by a so-called righting arm. The resulting **righting moment** is given by [7]

$$M_R = \rho g V \overline{GM_0} sin\varphi \tag{3.29}$$

where G is the center of gravity of the floater. If M_0 is above G, then M_R rotates the structure back to the initial position, however, if M_0 is below G, the righting moment actually increases the heeling angle. A submerged, neutrally-buoyant structure would flip over under this conditions (see figure 3.4), however, for a floating structure, B_0 moves as φ increases and this results into an additional stabilizing moment which leads the structure into a new equilibrium state.

Figure 3.5 shows that for large φ, the metacenter does not lie anymore on the line of vertical symmetry

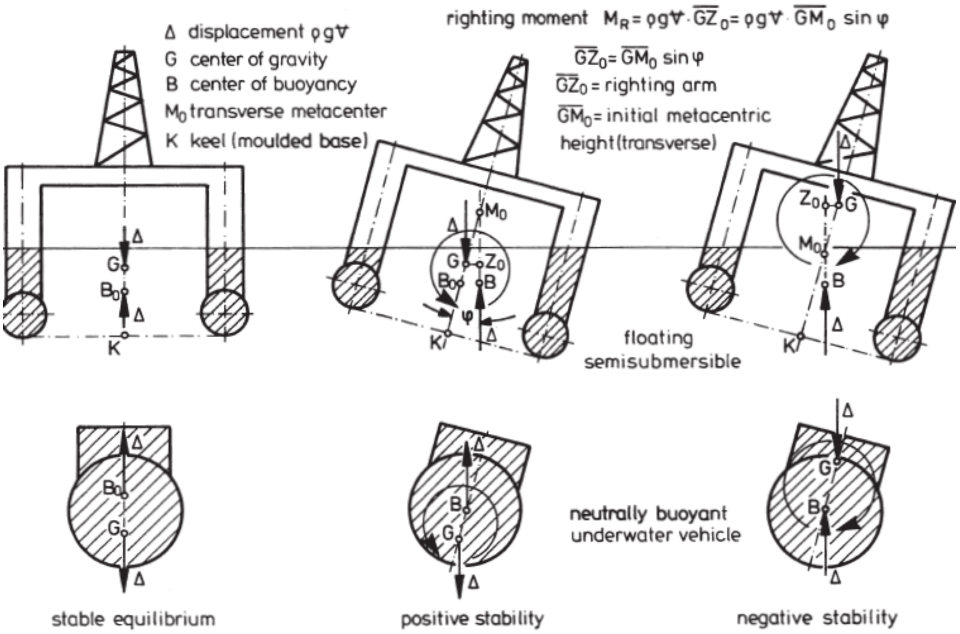


Figure 3.4: Comparison between floating and submerged structures stability [7].

of the body. The point identified by the intersection between the plane of symmetry of the floater and the action line of the buoyancy force, N_φ in figure 3.5, is called apparent metacenter. Because the metacenter is identified by the intersection of two near buoyancy line of action, the M_φ-curve is the evolute of the B_φ-curve.

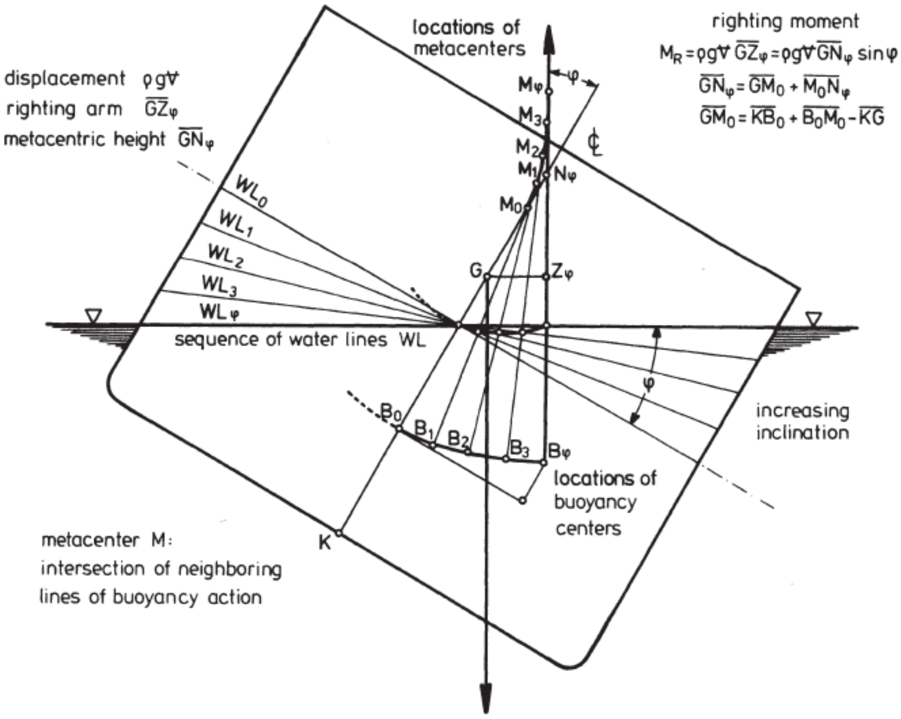


Figure 3.5: Stability of a floater at large φ [7].

3.4. Hydrodynamic analysis

In this section the hydrodynamic analysis of a 6-DoF floating body under the action of a wave train is presented. First, the hydrodynamic forces caused by an incident wave are introduced, then, the dynamics of a floating body due to wave-structure interaction are analysed. As is typical in this kind of dissertations, first, an easier case of a purely heaving cylinder is presented, secondly, the case of a freely floating body with six DoF is treated. Particular attention is posed on the description of the heave decay test, e.g. how to perform one and what information can be extracted from the record. This is of importance as this test will be carried out in the non-linear NWT in Section 6.4.

3.4.1. Hydrodynamic forces

In order to compute the hydrodynamic forces caused by waves and current, integration of the pressure field over the wetted surface of the floating structure is needed. In the easier case of slender structures, when the wavelength L is relatively large with respect to the characteristic dimension of the floater D (i.e. $D/L < 0.2$ [7]), the main forces are [7] [14]:

- **Froude-Krylov force**: this force is due to the pressure field of the undisturbed incident wave;
- **Hydrodynamic mass force** ('added mass'): this force is caused by the relative acceleration of the body with respect to the fluid;
- **Drag force**: this force is caused by the relative velocity of the body with respect to the fluid in a viscous flow.

When D is relatively small with respect to L, then the structure is said to be 'hydrodynamically transparent' as it does not relevantly deform a wave passing through it. In this condition, **diffraction** and **reflection** phenomena are marginal and they can be disregarded. However, for bigger structures, a wave passing the floater is indeed deformed by the presence of the body and those phenomena have to be taken into account. For a submerged body, the pressure force, in the direction of the wave development, can be computed as the pressure difference across a volume element, $dV = dxdydz$, considering a linear Taylor series development of the pressure [7].

$$dF_x = -\frac{\partial p}{\partial x} dV \tag{3.30}$$

The Froude-Krylov force can be computed by means of pressure integration over the wetted surface S. Applying Gauss theorem, the integral can be expressed as a volume integral and, if the unsteady term of Euler equation 3.4 is introduced, the resulting force is given by

$$F = \int_{(S)} p\mathbf{n}dS = -\int_{(V)} \nabla p dV = \int_{(v)} \rho \frac{d\mathbf{v}}{dt} dV \tag{3.31}$$

In general, for a non-fully-submerged structure, the Froude-Krylov force is expressed in terms of velocity potential Φ [7]

$$\mathbf{F} = \rho \int_{(S)} \frac{\partial \Phi}{\partial t} \mathbf{n} dS \tag{3.32}$$

It should be noticed that this force depends only on the characteristics of the external flow and it is no affected by the motions of the body.

The hydrodynamic mass force, also called added mass, can be computed by means of integration of the pressure field which derives from the presence of a relative acceleration between structure and fluid [7]. In other words, the motions of the body generate a pressure field around the body itself which accelerates the fluid in the region close to the body. In the case of body moving with velocity u_b in a fluid at rest, the added mass force can be computed as [7]

$$F_{x_{AM}} = -m_{11}\dot{u}_b = -C_a \rho V \dot{u}_b \tag{3.33}$$

where $\dot{u}_b$ is the acceleration of the body and C_a is the **added mass coefficient**. For the case of a floating body subjected to a wave train, the fluid velocity is non zero and the added mass force is given by [7]

$$F_{x_{AM}} = C_a \rho V (\dot{u}_f - \dot{u}_b) \tag{3.34}$$

where $\dot{u}_f$ stands for the fluid acceleration or water particle acceleration. It should be noticed that this force, unlike the Froude-Krylov one, is affected by both the external flow and the body motions.
The drag force derives from the viscous effects and the related downstream wake and it is given by [7]

$$F_D = C_d \frac{\rho}{2} A |u| u \tag{3.35}$$

where C_d is the drag coefficient, A is the cross-sectional area of the body and the term $|u|u$ ensures that drag force and velocity always act in the same direction [7]. Also in this case, as for the added mass force, both body and external flow have a contribution on the force definition.
For slender structures, i.e. hydrodynamically transparent, the **Morison Equation** can be applied to compute the horizontal force acting on a section, dz, under the effect of an incident wave. For a vertical pile of diameter D (Figure 3.6), this force is defined as the sum of a non-linear drag force and an inertia force and, in the case of a fixed structure immersed into a moving flow, it is given by [7]

$$dF_x = (f_m + f_d)dz = C_m \rho \frac{\pi D^2}{4} dz \frac{\partial u}{\partial t} + C_d \frac{\rho}{2} D dz |u| u \tag{3.36}$$

where u and $\frac{\partial u}{\partial t}$ stand for horizontal particle velocity and acceleration respectively and $C_m = 1 + C_a$ is the inertia coefficient which is dependent on the added mass coefficient. Here, as the structure is hydrodynamically transparent, the local acceleration $\frac{\partial u}{\partial t}$ is used instead of the substantial acceleration $\frac{du}{dt}$ [7]. This substitution is in agreement with linear theory which disregards the convective term of the velocity [7]. It is important to note that, as u and $\dot{u}$ are shifted by 90°, then the same happens for the two components of the Morison force.

In the case of a structure which is not fixed, the Morison Equation is modified in order to take into account the relative velocity between floater and fluid [7]

$$dF_x = \rho dV \frac{\partial u}{\partial t} + C_a \rho dV \left(\frac{\partial u}{\partial t} - \dot{u}_b \right) + C_d \frac{\rho}{2} dA |u - u_b| (u - u_b) \tag{3.37}$$

Finally, Figure 3.7 shows the region of applicability of the Morison Equation as a function of relative wave height $\pi H/D$ and relative structure size $\pi D/L$.

As aforementioned and here presented in Figure 3.7, Morison equation is valid only for slender structures. In fact, when a hydrodynamically compact structure is analysed, phenomena such as diffraction and reflection cannot be neglected anymore [14]. In particular, while for the Morison equation we assumed that the water particle velocity in the proximity of the structure does not considerably change with respect to the velocity at the pile axis, for compact structures this assumption is no more valid [7]. Here, as the incident wave is significantly deformed by the structure, the wave field potential is derived as superposition of two potentials, one associated with the undisturbed incident wave and the other related to a wave field generated by and reflecting from the body [7]. The resulting potential is usually written as [7] [14]

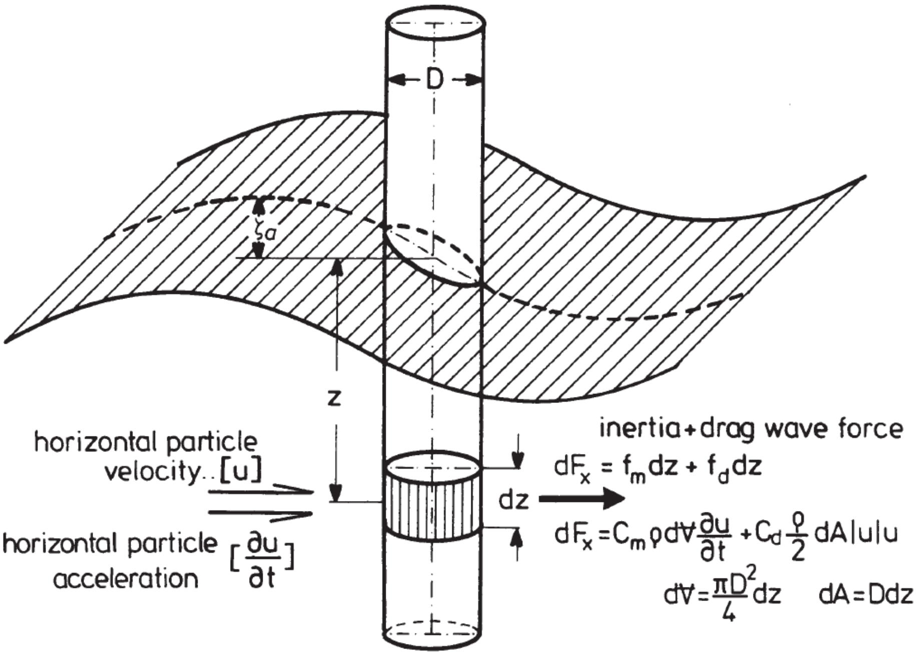

D
ζ_a
z
horizontal particle velocity..[u]
horizontal particle acceleration $[\frac{\partial u}{\partial t}]$
inertia+drag wave force
$dF_x = f_m dz + f_d dz$
dz
$dF_x = C_m \rho dV \frac{\partial u}{\partial t} + C_d \frac{\rho}{2} dA|u|u$
$dV = \frac{\pi D^2}{4} dz$
$dA = Ddz$

Figure 3.6: Vertical pile in a wave field [7].

$$\Phi = \Phi_0 + \Phi_7 + \sum_{j=1}^{6} \dot{s}_j \varphi_j \tag{3.38}$$

where Φ_0 is the wave field potential and Φ_7 and $\sum_{j=1}^{6} \dot{s}_j \varphi_j$ represent the potential of the generated waves due to the presence of the body. In particular, Φ_7 is related to the boundary conditions at the water-body interface while $\sum_{j=1}^{6} \dot{s}_j \varphi_j$ is related to the harmonic motions of the body [7]. The terms, $\dot{s}_j$ and φ_j represent harmonic velocity of the body and local potential for every degree of freedom respectively.

3.4.2. Dynamics of a floating rigid cylinder

The dynamics of a rigid body in a wave field results from external forces and moments combined with the inertia of the body itself. As typical in literature, first a simpler case of a purely heaving cylinder is presented in order to introduce most of the floating body theory while avoiding complications resulting from a higher number of DoF.

In general, for a floating body with a zero forward speed, the six rigid motions of its center of gravity related to translation along and rotation about the axis, are given by [14]

$$\textbf{Surge:} \quad x = x_a cos(\omega t + \epsilon_{x\zeta}) \tag{3.39a}$$

$$\textbf{Sway:} \quad y = y_a cos(\omega t + \epsilon_{y\zeta}) \tag{3.39b}$$

$$\textbf{Heave:} \quad z = z_a cos(\omega t + \epsilon_{z\zeta}) \tag{3.39c}$$

$$\textbf{Roll:} \quad \phi = \phi_a cos(\omega t + \epsilon_{\phi\zeta}) \tag{3.39d}$$

$$\textbf{Pitch:} \quad \theta = \theta_a cos(\omega t + \epsilon_{\theta\zeta}) \tag{3.39e}$$

$$\textbf{Yaw:} \quad \psi = \psi_a cos(\omega t + \epsilon_{\psi\zeta}) \tag{3.39f}$$

where $\epsilon_{i\zeta}$ represents the phase shift between the wave elevation ζ and the motion i. Because the treated system is linear, then motions superposition is possible. An important assumption here is that motions are of

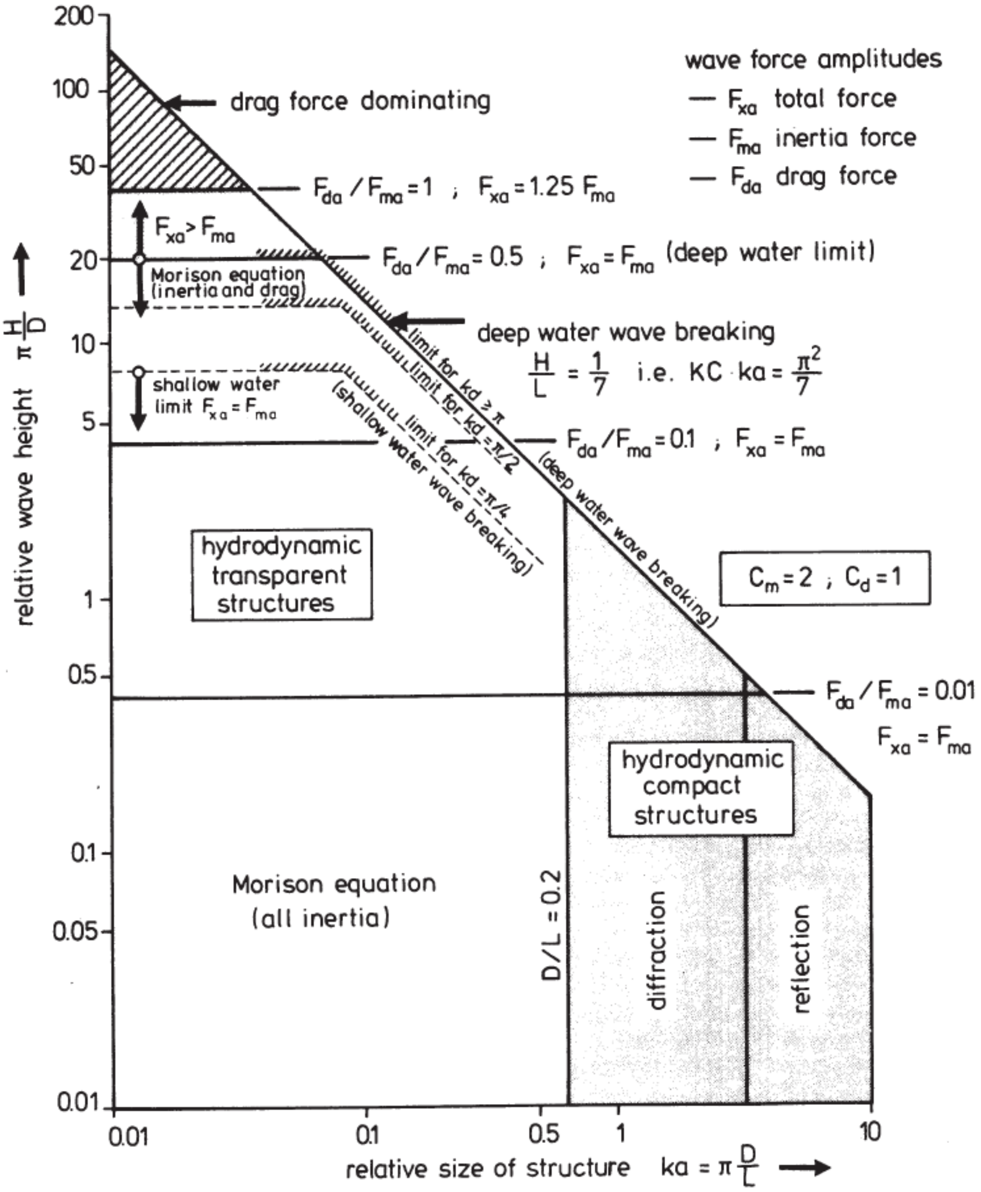


Figure 3.7: The figure presents the loading regimes of the horizontal force acting on a vertical pile as function of relative wave height and relative size of the structure [7].

a small amplitude, thus, the angular displacements, ϕ, θ and ψ, are so small that it can be assumed that [14]

$$sin\phi \simeq \phi \quad \text{and} \quad cos\phi \simeq 1 \tag{3.40a}$$

$$sin\theta \simeq \theta \quad \text{and} \quad cos\theta \simeq 1 \tag{3.40b}$$

$$sin\psi \simeq \psi \quad \text{and} \quad cos\psi \simeq 1 \tag{3.40c}$$

This assumption leads to the possibility to express the harmonic motions of a point $P(x_b, y_b, z_b)$ on the rigid body as [14]

$$x_P = x - y_b\psi + z_b\theta \tag{3.41a}$$
$$y_P = y + x_b\psi - z_b\phi \tag{3.41b}$$
$$z_P = z - x_b\theta + y_b\phi \tag{3.41c}$$

From equation 3.41 it can be seen how motions are coupled and, for instance, the vertical displacement is affected by heave, roll and pitch. The possibility to superpose motions for linear systems is very useful also while dealing with irregular waves. In fact, it is possible to consider an irregular wave as the resulting summation of multiple regular waves, with different amplitude, wavelength and direction [14]. With respect to a heaving cylinder in a wave field, this property of linear systems leads to the possibility of evaluating this case as a combination of two different sub-cases: a freely floating cylinder in still water conditions and a restrained body under the effect of an incident wave train [14]. As a result, from Newton's second law, the equation of motion for the heaving cylinder is given by [14]:

$$\frac{d}{dt}(\rho V \cdot \dot{z}) = F_h + F_w \tag{3.42}$$

where F_h and F_w are the hydromechanical and waves forces respectively. Figure 3.8, shows the scheme of this superposition.

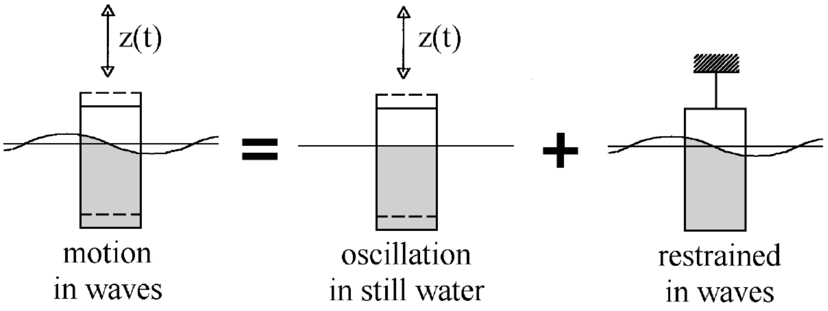


Figure 3.8: Hydromechanical and waves forces are superposed to obtain the motions of the body in a wave field [14].

For the first analysed case, a simple **free decay test** in still water, the equation of motions of the heaving cylinder can be written as [14]

$$m\ddot{z} = -P + \rho g(T - z)A_w - b\dot{z} - a\ddot{z} \tag{3.43}$$

From Archimedes' law, the mass force $P = mg$ is equal to the buoyancy, $\rho g T A_w$, with A_w being the water plane area, and the heave motion can be rewritten as [14]

$$(m + a_{zz})\ddot{z} + b_{zz}\dot{z} + c_{zz}z = 0 \tag{3.44}$$

The coefficient a_{zz} represents the hydrodynamic mass already introduced in section 3.4.1 for the heave motion. The cylinder, through its vertical motion, generates a wave field which radiates from it. Because these waves carry energy, the vertical motion dies out after some time. This damping effect is proportional to to $\dot{z}$ and for this reason, b_{zz} is called hydrodynamic damping coefficient. While dealing with a viscous fluid, damping can be caused by friction and other viscous effects and, especially for roll motions, this can be really relevant resulting in non-linear damping coefficients [14]. The term c_{zz} is the restoring spring coefficient,

$c_{zz} = \rho g A_w$, and it is present only for heave, roll and pitch. For angular motions, this coefficient derives from static stability considerations and is defined as [14]

$$c_{\phi\phi} = \rho g V \cdot \overline{GM} \tag{3.45}$$

$$c_{\theta\theta} = \rho g V \cdot \overline{GM_L} \tag{3.46}$$

where $\overline{GM}$ and $\overline{GM_L}$ are respectively transverse and longitudinal metacentric heights. For the case of vertical motions, Equation 3.44 can be expressed as [14]:

$$\ddot{z} + 2\nu\dot{z} + \omega_0^2 z = 0 \tag{3.47}$$

where

$$2\nu = \frac{b_{zz}}{m + a_{zz}} \tag{3.48}$$

and

$$\omega_0^2 = \frac{c_{zz}}{m + a_{zz}} \tag{3.49}$$

stand for damping coefficient and undamped natural frequency respectively. Now, an **adimensional damping coefficient**, κ, can be defined as the ratio of ν over ω_0.

$$\kappa = \frac{\nu}{\omega_0} = \frac{b_{zz}}{2\sqrt{(m + a_{zz})c_{zz}}} \tag{3.50}$$

where the term $b_{cr} = 2\sqrt{(m + a_{zz})c_{zz}}$ is known as critical damping coefficient. Introducing κ in Equation 3.47, it can be rewritten as [14]:

$$\ddot{z} + 2\kappa\omega_0\dot{z} + \omega_0^2 z = 0 \tag{3.51}$$

In a heave decay test, the floater is initially displaced with respect to its equilibrium position and successively released. Assuming an initial displacement $z(t = 0s) = z_a$, then the solution of the equation of motion, Equation 3.47, can be expressed as [14]

$$z = z_a e^{-\nu t}\left(cos\omega_z t + \frac{\nu}{\omega_z} sin\omega_z t\right) \tag{3.52}$$

where ω_z is the natural frequency and $z_a e^{-\nu t}$ represents the decrease of the pick after one period. According to [14], the **logarithmic decrement** of the heave motion is given by

$$\nu T_z = \kappa\omega_0 T_z = ln\left\{\frac{z(t)}{z(t + T_z)}\right\} \tag{3.53}$$

where T_z is the natural period and $z(t)$ stands for the amplitude of the heave motion at the instant t in time. In general, ν is small and thus $\nu^2 \ll \omega_0^2$. Because $\omega_z^2 = \omega_0^2 - \nu^2$, this leads to the assumption that $\omega_z \simeq \omega_0$. Consequently, $\omega_0 T_z \simeq \omega_z T_z = 2\pi$ and Equation 3.53 can be rewritten, isolating κ, to obtain the non-dimensional damping coefficient [14]:

$$\kappa = \frac{1}{2\pi} ln\left\{\frac{z(t)}{z(t+T_z)}\right\} \tag{3.54}$$

Therefore, the non-dimensional damping can be easily found if results from a decay test are available. When small amplitudes are involved, then it is preferable to used a double amplitude approach to determine κ [14]. In this case, κ is computed as

$$\kappa = \frac{1}{2\pi} ln\left\{\frac{z_{a_i} - z_{a_{i+1}}}{z_{a_{i+2}} - z_{a_{i+3}}}\right\} \tag{3.55}$$

where z_{a_i} is the amplitude of the motion for the i^{th} pick in Figure 3.9.

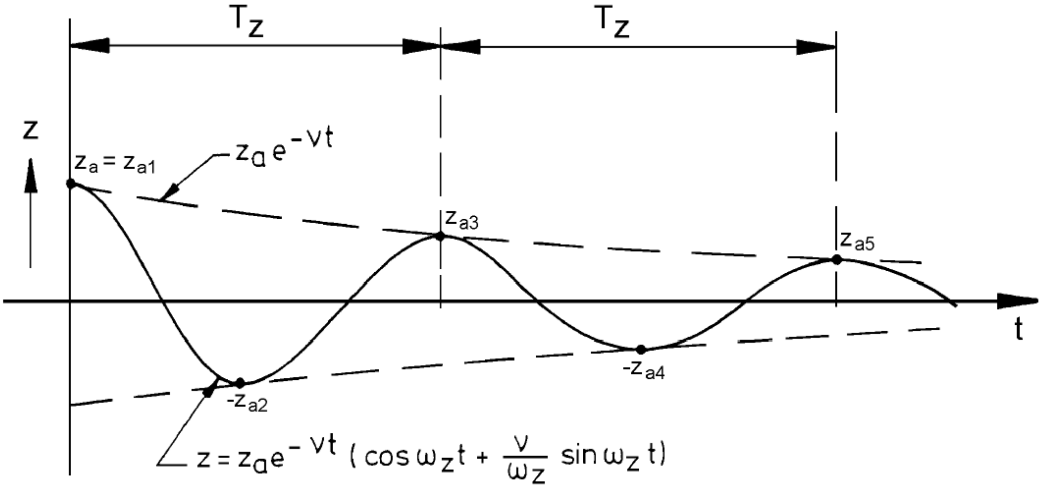


Figure 3.9: Example of record obtain from a decay test. Here, the determination of the logarithmic decrement is presented [14].

Once the non-dimensional damping is known, the hydrodynamic added mass and damping can be computed as:

$$a_{zz} = \frac{c_{zz}}{\omega_0^2} - m \tag{3.56}$$

$$b_{zz} = \frac{2\kappa c_{zz}}{\omega_0} \tag{3.57}$$

where ω_0 is the natural frequency obtain from the record of the heave motion while m and c_{zz} are known values computed from the geometry of the body. The importance of a decay test is that it provides the hydrodynamic coefficients which are necessary to predict the motions of a body [14]. Moreover, once added mass and damping coefficients are determined, either through experiments or CFD simulations, the motions of a body in a wave field can be predicted with less (computationally) expensive approaches, such as potential flow. However, since the free decay test is, necessarily, performed at natural frequency, it does not provide any information regarding the variation of a and b with respect to frequency and the so-obtained coefficients are valid only at ω_z. Thus, a **forced oscillation test** should be carried out in order to describe this relation. By applying a vertical sinusoidal force, such as $F_a sin(\omega t + \epsilon_{Fz})$, after some mathematics, it is possible to obtain [14]

$$a_{zz} = \frac{c_{zz} - \frac{F_a}{z_a} cos\epsilon_{Fz}}{\omega^2} - m \tag{3.58}$$

$$b_{zz} = \frac{\frac{F_a}{z_a} sin\epsilon_{Fz}}{\omega} \tag{3.59}$$

$$c_{zz} = \rho g A_w \tag{3.60}$$

Here, c_{zz} is obtained from geometry while a_{zz} and b_{zz} are computed from the equation of motion for $\omega t = \pi/2$ and $\omega t = 0$ respectively. The choice of these particular cases derives from the relation between exciting force and induced heave motion. In fact, the component of the applied force which is in-phase with the motion is related to inertia, i.e. added mass, while the out-of-phase component is linked to damping [14]. Thus, when the equation of motion is evaluated for $\omega t = \pi/2$, the added mass can be isolated from the damping coefficient and Equation 3.58 is obtained. In the same manner, solving for $\omega t = 0$, a formulation of the damping coefficient versus ω can be found.
For the case of a restrained cylinder in a wave field, the previously introduced Froude-Krilov force 3.31, is redefined as

$$F_{FK} = c_{zz} \cdot \zeta^* \tag{3.61}$$

where c_{zz} is the spring coefficient and $\zeta^* = e^{-kT} \cdot \zeta_a cos(\omega t)$ is called reduced wave elevation. It is important to underline that this formulation is valid only in deep water condition [14]. The total wave force is given by [14]

$$F_w = a_{zz}\ddot{\zeta}^* + b_{zz}\dot{\zeta}^* + c_{zz}\zeta^* \tag{3.62}$$

where $a_{zz}\ddot{\zeta}^*$ and $b_{zz}\dot{\zeta}^*$ are introduced as correction factors of the Froude-Krilov force in order to take into account diffraction phenomena due to the presence of the body as explained in Section 3.4.1.
Superimposing the two introduced cases, the equation of motion of a heaving cylinder under the action of an incident wave train can be written as [14]

$$(m + a_{zz})\ddot{z} + b_{zz}\dot{z} + c_{zz}z = a_{zz}\ddot{\zeta}^* + b_{zz}\dot{\zeta}^* + c_{zz}\zeta^* \tag{3.63}$$

Substituting the heave response for regular waves (3.39) into Equation 3.63, after some mathematics, the **Response Amplitude Operator (RAO)** and the **phase characteristic** can be written as [14]

$$\frac{z_a}{\zeta_a} = e^{-kT}\sqrt{\frac{(c_{zz} - a_{zz}\omega^2)^2 + (b_{zz}\omega)^2}{[c_{zz} - (m + a_{zz})\omega^2]^2 + (b_{zz}\omega)^2}} \tag{3.64}$$

$$\epsilon_{z\zeta} = arctan\left\{\frac{-mb_{zz}\omega^3}{(c_{zz} - a_{zz}\omega^2)[c_{zz} - (m + a_{zz})\omega^2]^2 + (b_{zz}\omega)^2}\right\} \qquad 0 \leq \epsilon_{z\zeta} \leq 2\pi \tag{3.65}$$

The amplitude characteristic, or RAO, and the phase characteristic are together referred to as **frequency characteristic**. Figure 3.10 shows the frequency characteristic of a heaving cylinder.

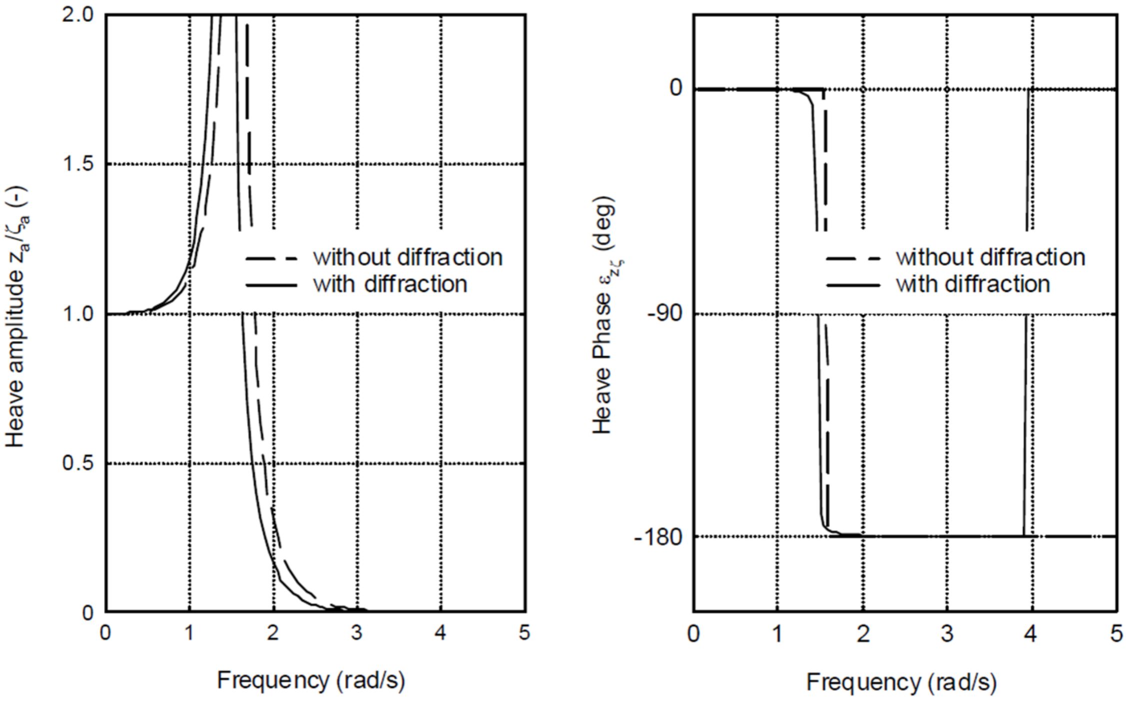


Figure 3.10: Rao and phase characteristic for heaving cylinder [14].

From Figure 3.10, the effect of the diffraction corrections, $a\ddot{\zeta}^*$ and $b\dot{\zeta}^*$, can be seen. From the left-hand side of Figure 3.10 is also possible to notice that the diffraction term is not really relevant for low frequency waves while its effect increases significantly for high frequency waves. Analysing Equation 3.64, three regions in the frequency domain can be identified [14]:

- **Low frequency region**: in this area, motions are dominated by the restoring spring term and the body behaves like a small ball in a wave field. The RAO tends to one and the phase to zero and consequently $\omega^2 \ll c_{zz}/(m + a_{zz})$.

- **Natural frequency region**: here, since $\omega^2 \approx c_{zz}/(m+a_{zz})$, damping is the dominant term. Thus, if c_{zz} is small, large amplitudes may occur at resonance. A phase shift of $-\pi$ is present here and the abruptness of this shift is determined by b_{zz}. The smaller b_{zz} is the less smooth the shift is.

- **High frequency region**: Inertia term is the dominating figure in this area and $\omega^2 \gg c_{zz}/(m + a_{zz})$. As the wavelength is small with respect to the diameter of the cylinder, the effect of the wave field is less noticeable on the motions of the floater. A second phase shift, resulting from a phase shift in the wave load, is presented.

Figure 3.11 schematically presents the three just described regions. Let's now consider again Equation 3.41 previously introduced in this section. As stated before, because the system under consideration is linear, the motions of the body can be derived by means of superposition of the six motions of a body, three linear displacements, surge, sway and heave, and three angular displacements, roll, pitch and yaw. Moreover, Equation 3.41, shows how motions in different directions are coupled. In general, the equations of motion of the body can be written as [14]

$$\sum_{j=1}^{6} m_{i,j} \cdot \ddot{x}_j = F_i \quad for \;\; i = 1, \ldots 6 \tag{3.66}$$

where $m_{i,j}$ is the 6x6 mass matrix, $\ddot{x_j}$ is the body acceleration in the direction j^{th} and F_i is the load applied in the direction i^{th}. For a freely floating cylinder in a wave field, introducing added mass, hydrodynamic damping and restoring spring terms, the equations of motion can be written as follow [14]:

$$(\rho V + a_{11})\ddot{x} + b_{11}\dot{x} + c_{11}x + a_{13}\ddot{z} + b_{13}\dot{z} + c_{13}z + a_{15}\ddot{\theta} + b_{15}\dot{\theta} + c_{15}\theta = F_{w_x} \qquad (\textbf{Surge}) \tag{3.67a}$$

$$a_{31}\ddot{x} + b_{31}\dot{x} + c_{31}x + (\rho V + a_{33})\ddot{z} + b_{33}\dot{z} + c_{33}z + a_{35}\ddot{\theta} + b_{35}\dot{\theta} + c_{35}\theta = F_{w_z} \qquad (\textbf{Heave}) \tag{3.67b}$$

$$a_{51}\ddot{x} + b_{51}\dot{x} + c_{51}x + a_{53}\ddot{z} + b_{53}\dot{z} + c_{53}z + (I_{yy} + a_{55})\ddot{\theta} + b_{55}\dot{\theta} + c_{55}\theta = F_{w_\theta} \qquad (\textbf{Pitch}) \tag{3.67c}$$

$$(\rho V + a_{22})\ddot{y} + b_{22}\dot{y} + c_{22}y + a_{24}\ddot{\phi} + b_{24}\dot{\phi} + c_{24}\phi + a_{26}\ddot{\psi} + b_{26}\dot{\psi} + c_{26}\psi = F_{w_y} \qquad (\textbf{Sway}) \tag{3.67d}$$

$$a_{42}\ddot{y} + b_{42}\dot{y} + c_{42}y + (I_{xx} + a_{44})\ddot{\phi} + b_{44}\dot{\phi} + c_{44}\phi + (-I_{xz} + a_{46})\ddot{\psi} + b_{46}\dot{\psi} + c_{46}\psi = M_{w_\phi} \qquad (\textbf{Roll}) \tag{3.67e}$$

$$a_{62}\ddot{y} + b_{62}\dot{y} + c_{62}y + (-I_{zx} + a_{64})\ddot{\phi} + b_{64}\dot{\phi} + c_{64}\phi + (I_{zz} + a_{66})\ddot{\psi} + b_{66}\dot{\psi} + c_{66}\psi = M_{w_\psi} \qquad (\textbf{Yaw}) \tag{3.67f}$$

where a_{ij}, b_{ij} and c_{ij} are the components of the 6x6 added mass matrix, 6x6 hydrodynamic damping matrix and 6x6 restoring spring coefficient matrix respectively. For linear displacement, the terms F_{w_i} represent the wave forces in the i direction. For the rotations, the mass moments of inertia, I_{ij}, and the wave moments, M_{w_i}, are introduced. In most cases, by applying some geometry considerations, the equations of motion can be simplified [13] [14]. The six equations of motion can be divided into two sets of three equations: (1) a set of symmetric motions, surge, heave and pitch and (2) a set of anti-symmetric motions, sway, roll and yaw [14]. The equations of motion can be simplified also considering that symmetric motions cannot affect antisymmetric ones and vice versa [26].

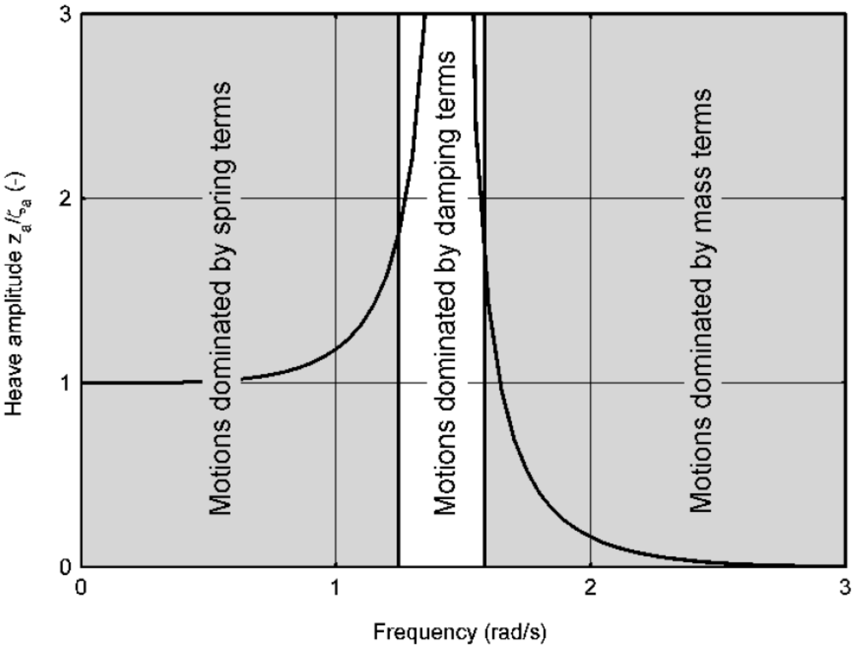


Figure 3.11: The figure shows the three frequency areas related to the motions of the cylinder [14].

4

The rigid body code

This section introduces the in-house developed 3-dimensional rigid body code which has been used to evaluate the dynamics of the rigid floater. This code is the resulting extension of an already existing 2-dimensional version of it developed by the MScEng. J.D. Brandsen from Delft University of Technology. This python-developed code utilizes non-uniform rational B-splines (NURBS) to generate the geometry of the body, thus first a brief introduction to isogeometric analysis and NURBS theory is necessary. Secondly, the code is described and validated.

4.1. Non-uniform rational B-splines

In the following sections, the isogeometric analysis is introduced together with the NURBS concepts which have been implemented in the python code. For a detailed literature review regarding these concepts, the reader is referred to [12] and [18].

4.1.1. The isogeometric concept

The aim of isogeometric analysis is to fill the gap between CAD (Computer Aided Design) and FEA (Finite Element Analysis). In fact, when moving from the CAD geometry to the FE mesh, there is a loss of accuracy which may represent a bottleneck in many applications. For instance, shape optimization requires strong automated integration between CAD geometry and mesh generation to be effective. Thus, in order to break down this barrier, isogeometric analysis recycles an already existing concept, the isoparametric finite element analysis, reversing the process. In fact, while in FEA the basis chosen to estimate the unknown solution is subsequently adapted to approximate the known geometry, the isogeometric concept selects an exact representation of the known geometry and uses the same basis to approximate the solution field [12].
Among the computational geometry technologies which may be adopted in isogeometric analysis, NURBS represents the industry standard and seems to be the best candidate to fill this gap [12]. Its major advantages are the ability to exactly represent all conic sections, the presence of many stable algorithms to generate NURBS shapes, the ability to straightforwardly perform refinement through knot insertion, the C^{p-1} continuity for pth-order NURBS and the convex hull property [12].

4.1.2. NURBS as a geometry and mesh generator tool

When working with NURBS, two different notions of mesh are encountered, the **control mesh** and the **physical mesh**, see Figure 4.1. The control mesh looks very similar to a classical FE mesh and does not represent the geometry itself. The control variables are stored at the **control points** which are located on the intersections of the control grid. The physical mesh is what actually represents the shape, it is a decomposition of the real geometry. Two concepts arise from the definition of the physical mesh: **patches** and **knot spans**. A patch can be seen as a sub-domain which is decomposed into knot spans which, in turn, are bounded by **knots**. The geometry used for this work consists of only a single patch. Both patches and knot spans have two representations, one in the **parent domain** and one in the **physical space**. At knots the basis functions have C^{p-m} continuity where p is the polynomial order and m is the multiplicity of the knot [12]. Knots can be represented in the **index domain** where knots with multiplicity larger than 1 have a unique representation. Figure 4.1 schematically presents the above-introduced concepts.

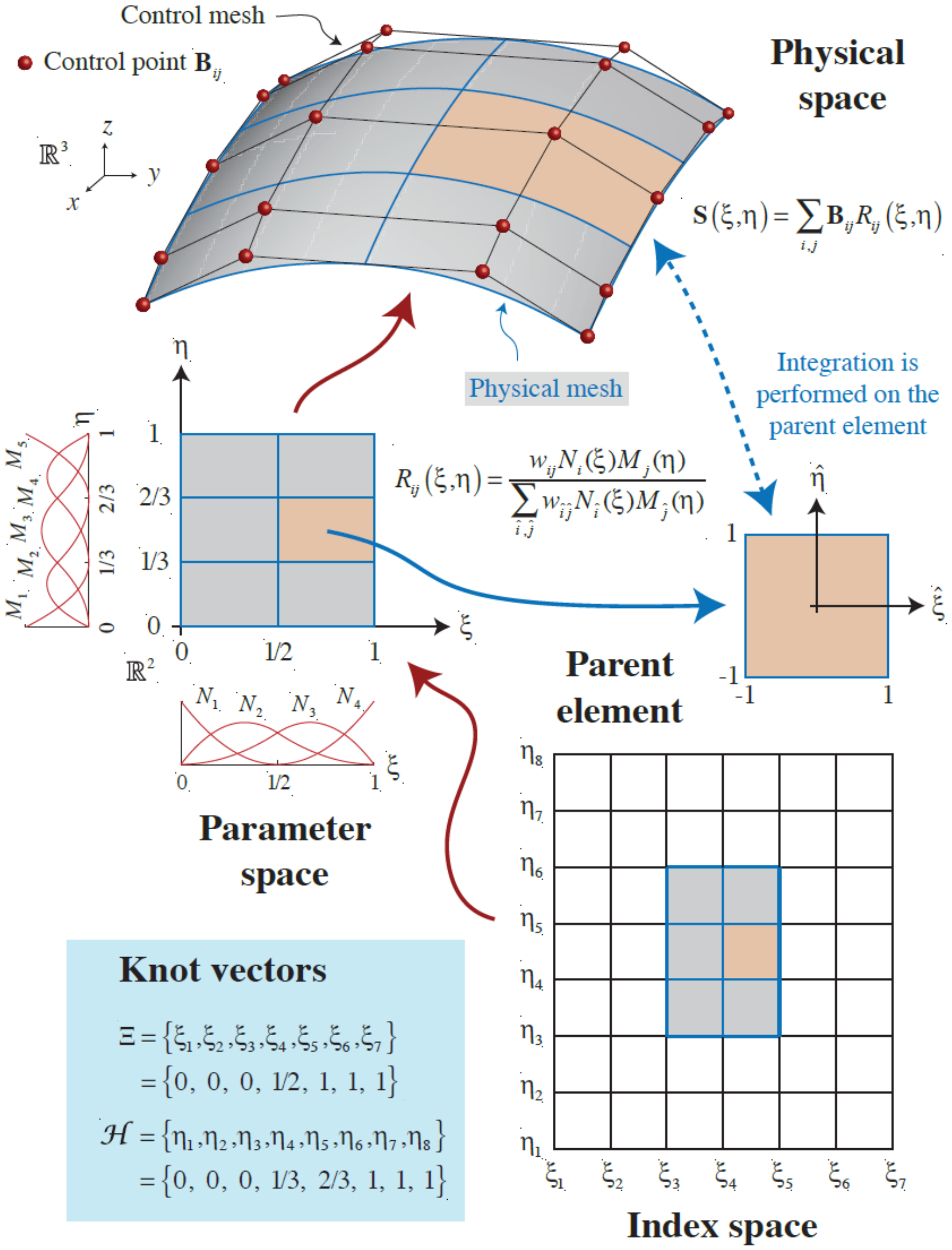

Control mesh
Control point $\mathbf{B}_{ij}$
$\mathbb{R}^3$
z
y
x
Physical space
$\mathbf{S}(\xi,\eta)=\sum_{i,j}\mathbf{B}_{ij}R_{ij}(\xi,\eta)$
Physical mesh
Integration is performed on the parent element
η
1
2/3
1/3
0
ξ
0
1/2
1
$\mathbb{R}^2$
M_1 M_2 M_3 M_4 M_5
N_1 N_2 N_3 N_4
$R_{ij}(\xi,\eta)=\frac{w_{ij}N_i(\xi)M_j(\eta)}{\sum_{\hat{i},\hat{j}}w_{\hat{i}\hat{j}}N_{\hat{i}}(\xi)M_{\hat{j}}(\eta)}$
$\hat{\eta}$
$\hat{\xi}$
1
-1
-1
1
Parent element
Parameter space
Knot vectors
$\Xi=\{\xi_1,\xi_2,\xi_3,\xi_4,\xi_5,\xi_6,\xi_7\}$
$=\{0,\ 0,\ 0,\ 1/2,\ 1,\ 1,\ 1\}$
$\mathcal{H}=\{\eta_1,\eta_2,\eta_3,\eta_4,\eta_5,\eta_6,\eta_7,\eta_8\}$
$=\{0,\ 0,\ 0,\ 1/3,\ 2/3,\ 1,\ 1,\ 1\}$
η_8 η_7 η_6 η_5 η_4 η_3 η_2 η_1
ξ_1 ξ_2 ξ_3 ξ_4 ξ_5 ξ_6 ξ_7
Index space

Figure 4.1: Schematic illustration of NURBS basic concepts [12].

As suggested by their name, NURBS are built from **B-splines**. Thus, the definition of a B-spline is first introduced. In order to define a B-spline, a knot vector is necessary. A one-dimensional knot vector is a set of coordinates in the parameter space, defined as [12][18]:

$$\Xi=\{\xi_1,...,\xi_{n+p+1}\} \tag{4.1}$$

where $\xi_i \epsilon \mathbb{R}$ is the i^{th} knot, $i=1,..,n+p+1$, n is the number of basis functions and p is the polynomial order. Knot vectors are said to be **uniform** when knots are equally spaced and **open** when the multiplicity of the first and last knot equals $p+1$. Once a knot vector is defined, the B-spline basis functions are defined through the **Cox-de Boor recursion formula** [12][18]:

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \tag{4.2}$$

which, for $p = 0$, can be simplified to a piecewise constant function:

$$N_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

B-spline basis functions of order p have generally $p - m_i$ continuous derivatives across ξ_i, where m_i is the multiplicity of ξ_i. If $m = p+1$ the basis functions are discontinuous at the knot and they represent a patch boundary [12]. Figure 4.2 shows the relationship between continuity and m for the case of quadratic basis functions: at the patch boundary, first and last knots, $m = p+1 = 5$ and the basic function are fully discontinuous; going from knot $\xi = 1$ to $\xi = 4$, the multiplicity increases and consequently the number of continuous derivatives decreases. The minimum possible number of continuous derivatives for a non-boundary knot is reached at $\xi = 4$ where the basis is interpolatory.

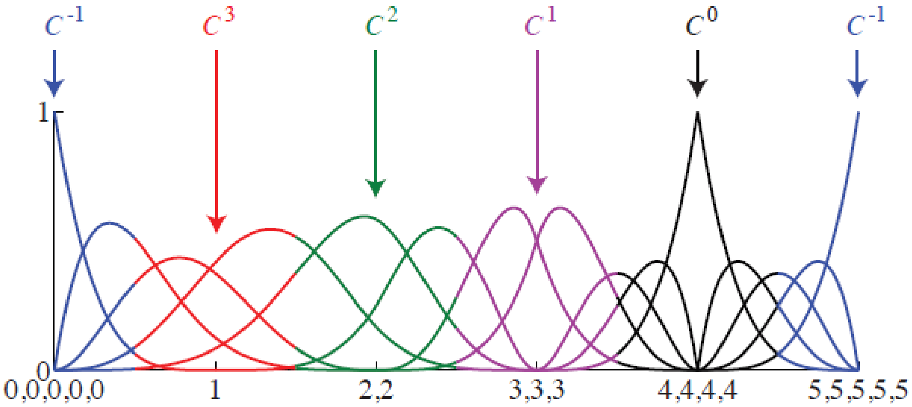


Figure 4.2: Quartic basis functions for the knot vector $\Xi = \{0,0,0,0,0,1,2,2,3,3,3,4,4,4,4,5,5,5,5,5\}$. Note that the knot vector is open and non-uniform [12].

Given a knot vector Ξ and a polynomial order p, the k^{th} derivative of the i^{th} basis function can be computed as [12][18]

$$\frac{d^k}{d^k\xi} N_{i,p}(\xi) = \frac{p}{\xi_{i+p} - \xi_i} \left(\frac{d^{k-1}}{d^{k-1}\xi} N_{i,p-1}(\xi)\right) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} \left(\frac{d^{k-1}}{d^{k-1}\xi} N_{i+1,p-1}(\xi)\right) \tag{4.4}$$

which, for $k = 0$, becomes

$$\frac{d}{d\xi} N_{i,p}(\xi) = \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \tag{4.5}$$

A **B-spline curve**, defined in $\mathbb{R}^d$, is the result of a linear combination of B-splines basis functions. Given a n number of basis functions, $N_{i,p}$, $i = 1, ..., n$, and control point $B_i \epsilon \mathbb{R}^d$, $i = 1, ..., n$, a piecewise polynomial B-spline curve is defined as [12] [18]

$$\mathbf{C}(\xi) = \sum_{i=1}^{n} N_{i,p}(\xi) \mathbf{B}_i \tag{4.6}$$

Some of the most important properties of B-splines curves are, the **affine invariance** property, the **convex hull** property and the **variation diminishing** property. For a detail explanation of these properties, the reader is referred to [12] and [18]. In $\mathbb{R}^d$, Non-Uniform Rational B-Splines (NURBS) are derived from projecting a B-spline curve in $\mathbb{R}^{d+1}$. At this stage, the concepts of **weight**, w_i, associated to the i^{th} control point, B_i, are introduced. In the $\mathbb{R}^3$ space, weights can be seen as the z-component of the projective control points. In a generic $\mathbb{R}^d$ space, they are defined as follow [12]:

$$\begin{aligned} (\mathbf{B}_i)_j &= \frac{(\mathbf{B}_i^w)_j}{w_i}, \quad j = 1, ..., d \\ w_i &= (\mathbf{B}^w)_{d+1} \end{aligned} \tag{4.7}$$

Once the notion of weight is defined, the algebraic definition of a **NURBS** basis is given by [12]

$$\mathbf{R}_i^p(\xi) = \frac{N_{i,p}(\xi)\, w_i}{\mathbf{W}(\xi)} = \frac{N_{i,p}(\xi)\, w_i}{\sum_{\hat{i}=1}^{n} N_{\hat{i},p}(\xi)\, w_{\hat{i}}} \tag{4.8}$$

which, based on the B-spline curve definition and making use of Equations 4.7 and **??**, naturally leads to the equation for a **NURBS curve**

$$\mathbf{C}(\xi) = \sum_{i=1}^{n} \mathbf{R}_i^p(\xi)\mathbf{B}_i \tag{4.9}$$

Analogously, a NURBS solid can be defined as [12]

$$\mathbf{R}_{i,j,k}^{p,q,r}(\xi,\eta,\zeta) = \frac{N_{i,p}(\xi)M_{j,q}(\eta)L_{k,r}(\zeta)\, w_{i,j,k}}{\sum_{\hat{i}=1}^{n}\sum_{\hat{j}=1}^{m}\sum_{\hat{k}=1}^{l} N_{\hat{i},p}(\xi)M_{\hat{j},q}(\eta)L_{\hat{k},r}(\zeta)\, w_{\hat{i},\hat{j},\hat{k}}} \tag{4.10}$$

It is important to understand that all the properties of B-spline are retained by NURBS. In fact, if $w_i = w_j$ for every i, j, then $\mathbf{R}_i^p(\xi) = N_{i,p}(\xi)$. Thus, B-splines are a particular subset of NURBS. The advantage of NURBS with respect to B-splines is that they are capable of exactly representing a large number of objects which cannot be accurately described by polynomial splines.

4.2. Code description

In this section, the rigid body code is described starting with the geometry. The second section of this paragraph deals with the implemented physics and describes the time integration scheme. Finally, validation of the code is performed by applying 9 simple load combinations and comparing the response with analytical solutions and results provided by the previous code.

4.2.1. Geometry

The geometry adopted for the spar-type floater in this work is a simple circular cylinder of diameter d and height h. In order to generate this shape by using NURBS only a single patch is necessary. The first step to produce a cylinder is to generate a circle in the x-y plane that will be successively extruded along the z-axis. Many different templates to generate a circle are available in literature and the reader is refereed to [18] for a detailed analysis of them and their properties and limitations. In this work, the circle is generated from a template consisting of a nine-point square control polygon and quadratic basis functions, $p = 2$, as shown in Figure 4.3(a).

The knots vector and the weights used to generate the circle in the code are

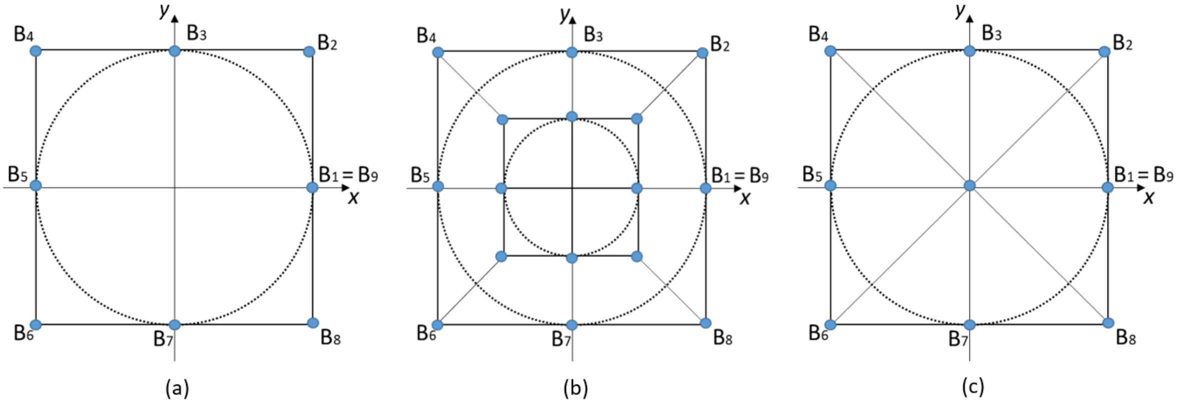


Figure 4.3: Figure (a) shows a typical circle template. In (b) a template for an annulus is represented while in (c) the template used to generate the circular NURBS surface can be seen.

$$\Xi = \{0,0,0,1,1,2,2,3,3,4,4,4\}$$

$$w_i = \begin{cases} 1/\sqrt{2} & \text{for } i \text{ even} \\ 1 & \text{otherwise} \end{cases} \tag{4.11}$$

The multiplicity of the first and last knots are set equal to three. In this way, the curve is closed and a patch is defined. The reason why the other knots are of multiplicity two arises from the fact that this template is the resulting assembly of four 90° arcs [18].

Now that the circle is constructed, two things need to be done in order to obtain a 3-dimensional cylindrical **NURBS mesh**. Firstly, the circle needs to be transformed into a circular surface and a linear element in x-y plane is used to do this, see Figure 4.3(c). Secondly, the generated surface is swept along a linear element that is aligned with the z-axis and the cylindrical mesh is obtained. In both cases, linear basis functions are used, $q = r = 1$, and the weights and knot vectors are

$$H = Z = \{0,0,1,1\}$$

$$w_j = w_k = 1 \tag{4.12}$$

The knot vectors are identical because they both represent a linear element that is interpolatory at its end points. The weights are equal to 1 for every j, k because the linear NURBS curve will pass through the associated control points. An important step in the process is to define the system of reference in the parametric space of the cylinder which, in this case, is defined using a cylindrical system of coordinates (ξ,η,ζ), where ξ is the circumferential direction and is thus associated with the quadratic basis functions along the circle, η is the radial coordinate and ζ is aligned with the height of the cylinder.

The last step in the geometry definition is to define the control grid. Given the nine control points that make up the circle, the same template can be repeated at distance h, normal to the x-y plane containing the circle, in order to create a rectangular box of dimensions $d \times d \times h$. At this stage, two extra control points are added at the centre of top and bottom surfaces of the box. These two control points are associated to the knots in the η dimension in the parametric space. Figure 4.3 shows the control points associated with the transformation of the circle into a surface. In figure 4.3 (b), a template for the top (bottom) face of an empty cylinder with a thickness smaller than the radius is shown. In figure 4.3 (c), the template implemented in the code is presented and it is derived from the degeneration of an annulus with a inner radius of zero. Finally, figure 4.4 shows the template created for the generation of the 3D cylindrical NURBS mesh (a) and the cylinder representation produced by the python code (b).

In order to generate a 3D NURBS cylinder, only two knot vectors actually need to be used: one in ξ-direction and one in ζ-direction. However, three were used here as the aim of this work was to implement in the python code a NURBS mesh which can be used in future for object of other shapes, distributed loads

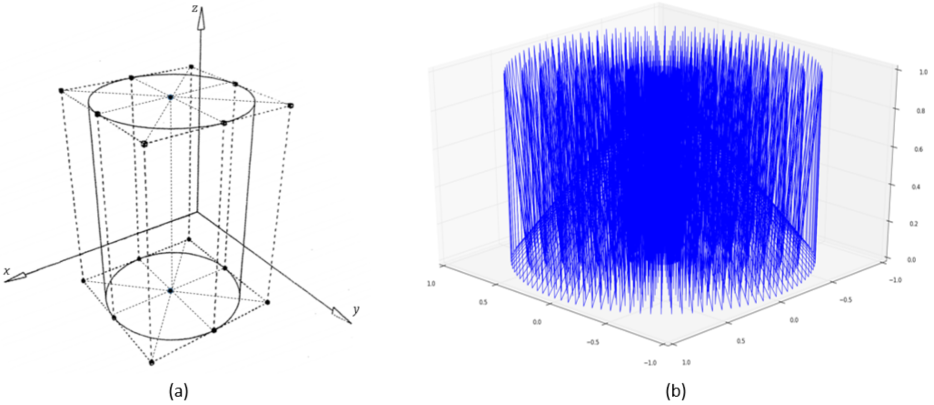


Figure 4.4: In (a), the cylindrical NURBS template is shown. In (b), the cylinder generated by the python code is presented.

and/or tensor fields. Towards this idea, some of the already implemented features will be introduced in section 4.2.2.

4.2.2. Physics and time integration scheme

In addition to generating a rigid body geometry, the code allows a user to apply springs, dampers, loads and constraints to the body. Until now, only concentrated loads have been validated. However, thanks to the NURBS mesh and the implementation of a three-dimensional gauss quadrature integration scheme in the code, the application of a tensor field or distributed loads is theoretically possible even though it is not functional yet. In the same way, the use of non-homogeneous objects can be simply implemented. This last feature may be especially interesting while dealing with floating rigid bodies as the centre of gravity of an object can be moved down by adding weight to the bottom section of a floater.

Both linear and torsional springs and dampers are implemented in the code. In this work, they will be used to simulate the effect of mooring lines connected to the floater in Chapter 6.4. Constraints can also be applied on the body in order to fix its translation or rotation or both. However, until now, it is not possible to constrain a translation along or rotation about a specific axis. In fact, only the full set of translation or rotation can be constraint. Even if it represents a limitation, this approach has been followed because it simplifies the implementation of point constraints in the code without affecting the rest of the project. Constraints can be used, for instance, in the wave-structure interaction analysis of a fixed truncated cylinder under the action of a regular wave train.

Finally, the code is capable of solving Newton's equations of motion for a rigid body in order to compute its linear and angular displacements, velocities and accelerations. An implicit **Newmark-beta scheme** is used as the time integration scheme to solve the second-order differential equations of motion. Linear and angular velocities and displacements are computed using

$$\dot{u}_{n+1} = \dot{u}_n + (1-\gamma)\Delta t \ddot{u}_n + \gamma \Delta t \ddot{u}_{n+1} \tag{4.13}$$

$$u_{n+1} = u_n + \Delta t \dot{u}_n + \frac{1-2\beta}{2}\Delta t^2 \ddot{u}_n + \beta \Delta t^2 \ddot{u}_{n+1} \tag{4.14}$$

where u_n, $\dot{u}_n$ and $\ddot{u}_n$ are respectively displacement, velocity and acceleration at $t = n\Delta t$, where Δt is the integration timestep size, and u_{n+1}, $\dot{u}_{n+1}$ and $\ddot{u}_{n+1}$ represents displacement, velocity and acceleration at $t = (n+1)\Delta t$. The two coefficients, γ and β, are chosen, in this work, to be equal to 0.5 and 0.25 respectively leading Newmark's scheme to reduce to the trapezoidal rule.

4.3. Validation

The original code by itself has been validated in [2] and it was producing good results, however, due to the intensive modifications implemented in the code to extend it to three dimensions, a validation of the new updated rigid body code is necessary. It is important to check the implementation of the linear and torsional springs and dampers as well as the dynamic response of the code. In order to do that, the following nine test case have been generated:

- **(i)** Six different single DoF mass-damper-spring systems compared with analytical solution:
  - Body free to translate along x-, y- and z-axis respectively, constraint by a spring and a damper;
  - Body free to rotate around x-, y- and z-axis respectively, constraint by a spring and a damper;
- **(ii)** Three DoF mass-damper-spring system compared with previous rigid body code: three combinations of two translations and the respective in-plane rotation;

Table 4.1 summarized the set-up of the nine simulations used to validate the rigid body code. For all the simulations, a cylinder of radius, $r = 1m$, height, $h = 1m$, and density, $\rho = 100kg/m^3$ has been assumed. All the test cases present almost uncoupled motions because all the loads are applied on the CoG of the body.

Test case	**DoF**	**Force**		**Moment**		**k**		**c**	
		direction and magnitude		**direction and magnitude**		**linear and torsional spring**		**linear and torsional damper**	
1	1	x	100 N	*None*	*None*	20 N/m	*None*	10 Ns/m	*None*
2	1	y	100 N	*None*	*None*	20 N/m	*None*	10 Ns/m	*None*
3	1	z	100 N	*None*	*None*	20 N/m	*None*	10 Ns/m	*None*
4	1	*None*	*None*	x	100 Nm	*None*	20 N/m	*None*	10 Ns/m
5	1	*None*	*None*	y	100 Nm	*None*	20 N/m	*None*	10 Ns/m
6	1	*None*	*None*	z	100 Nm	*None*	20 N/m	*None*	10 Ns/m
7	3	x, y	100 N, 100 N	z	100 Nm	20 N/m	20 N/m	10 Ns/m	10 Ns/m
8	3	y, z	100 N, 100 N	x	100 Nm	20 N/m	20 N/m	10 Ns/m	10 Ns/m
9	3	x, z	100 N, 100 N	y	100 Nm	20 N/m	20 N/m	10 Ns/m	10 Ns/m

Table 4.1: Summary of the 9 test cases performed to validate the python code.

The magnitude of the loads as well as the spring and damper coefficients have been arbitrarily chosen. The first six cases which has been tested are represented by single DoF mass-damper-spring systems under the effect of a constant load f, force or moment. This kind of assemble can be analytically solved through a scalar equation such as

$$m\ddot{x} + c\dot{x} + kx = f \tag{4.15}$$

where m is the mass of the body, c is the damping coefficient, k is the stiffness coefficient and $\ddot{x}$, $\dot{x}$ and x are respectively acceleration, velocity and displacement of the body. Figure 4.5 presents the computed $\ddot{x}$, $\dot{x}$ and x for the test case number 1 showing very good agreement between analytical solution and numerical solution. Here, also the old code has been introduced in the comparison in order to check if the new python code was more or less accurate then the old one. In the plots, all the quantities on the vertical axis have been normalized with the maximum of the analytical solution. Time is normalized with the period of the oscillation computed as:

$$T = \frac{1}{\omega_n \sqrt{1 - \zeta_c^2}} \tag{4.16}$$

where ω_n is the natural frequency of the system and ζ_c is the damping ratio that are respectively define as:

$$\omega_n = \frac{1}{2\pi}\sqrt{\frac{k}{m}} \tag{4.17}$$

and

$$\zeta_c = \frac{c}{2\sqrt{km}} \tag{4.18}$$

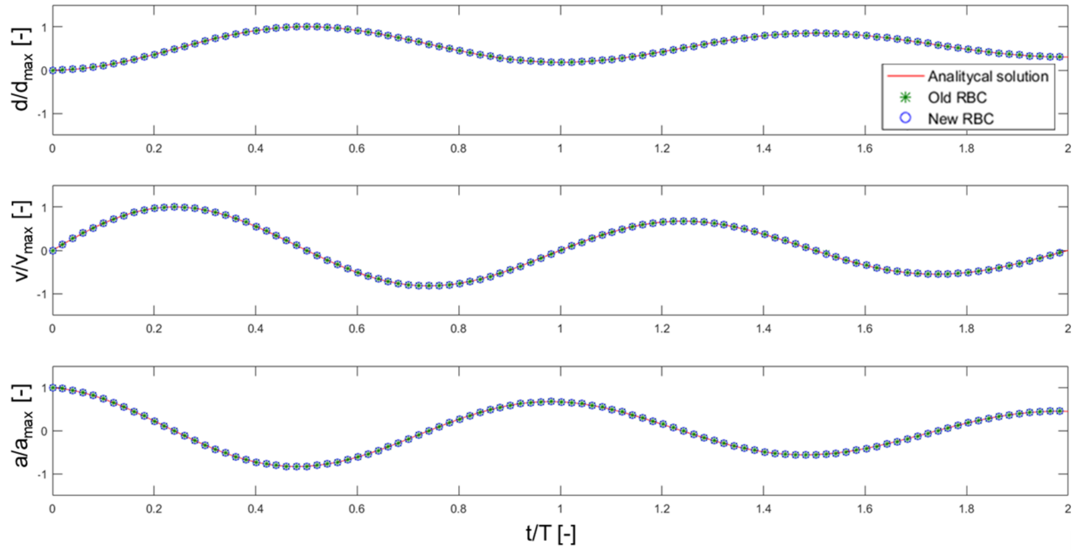


Figure 4.5: Test case 1: displacement, velocity and acceleration computed by the new and the old code compared with the analytical solution.

Same results as those presented in Figure 4.5 have been found for the other two 1DoF cases, namely test case 2 and 3. In all these simulations, the computed relative error, e_r, in the displacement of the CoG was found to be always smaller than 1.0%. From figure 4.6, e_r in time can be seen and it is also possible to see how the new code is slightly more accurate than the original code. Here, the relative error has been computed as:

$$e_r = \frac{\|x - x_{th}\|}{\|x_{th}\|} \times 100\% \tag{4.19}$$

where x and x_{th} are the numerical and analytical solution respectively. For both the codes the relative error has been found to be always smaller than 1%. The small difference in between the two numerical solutions can be attribute to the fact that the mass of the body is computed differently in the two codes. This leads to a small difference in the body mass used in the two codes which is then reflected in the motion. In fact, imposing the same mass in the two codes, the obtained motion perfectly matches in both amplitude and phase.

Similar results have been found also for the set of test cases, 4, 5 and 6. Figure 4.7 shows the results from test case 6 while Figure 4.8 presents $e_r = e_r(time)$ for the angular displacement. The relative error for the new code never exceeds 1.5% while for the old code e_r reaches a maximum of 5.3% at about $t/T = 1.2$.

This difference is caused by a different approach in the integration of the mass moment of inertia, I, in the two codes. Here, the difference in I introduces a phase shift with respect to the analytical solution which consequently lead to a higher error. Again, imposing the same mass moment of inertia the results were exactly matching.

After the single DoF systems have been tested, three more simulations have been performed combining multiple motions. The resulting displacement and rotations for one of these test cases, namely test case 7, are presented in Figure 4.9. For the three DoF cases, results have been compared only with those provided by the previous RBC. Moreover, in order to avoid the issue related to the different integration approach, the same mass and moment of inertia have been used in both codes. As it can be seen the two solutions match perfectly and as expected, thanks to the introduced correction, the relative error reduces to a maximum of 0.1%.

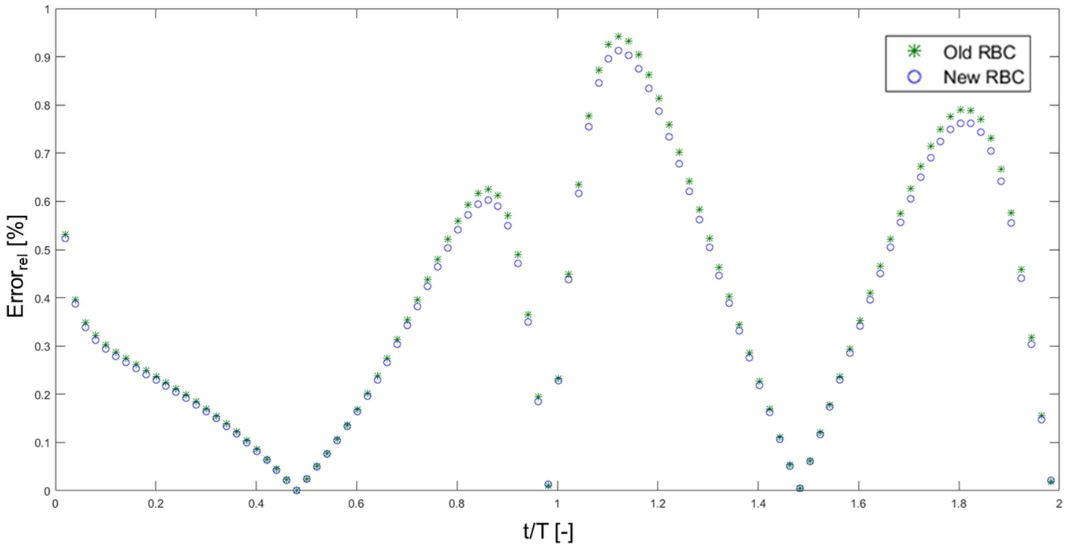


Figure 4.6: Test case 1: relative error in the displacement prediction for both the old and new code.

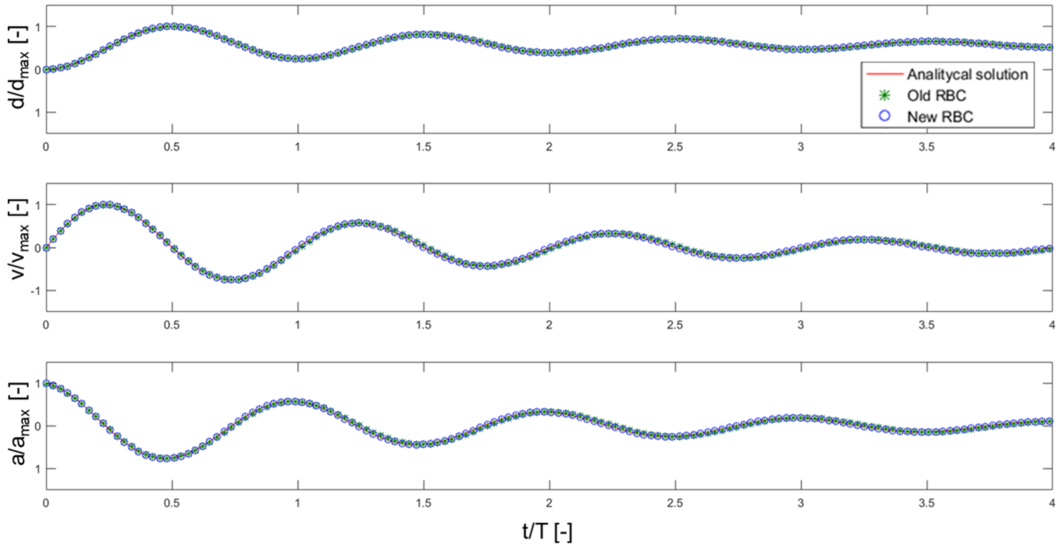


Figure 4.7: Test case 1: relative error for both the old and new code.

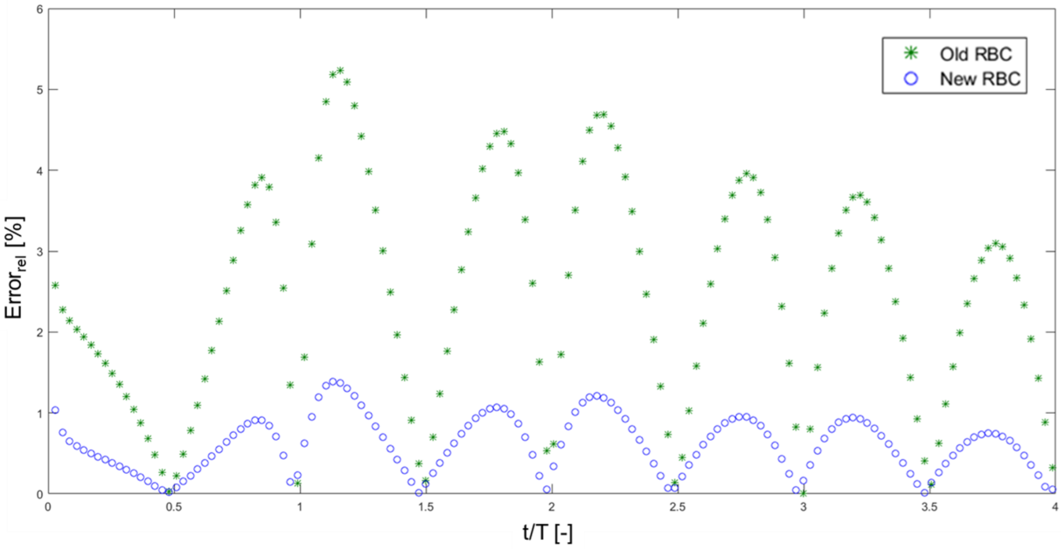


Figure 4.8: Test case 1: relative error in the displacement prediction for both the old and new code.

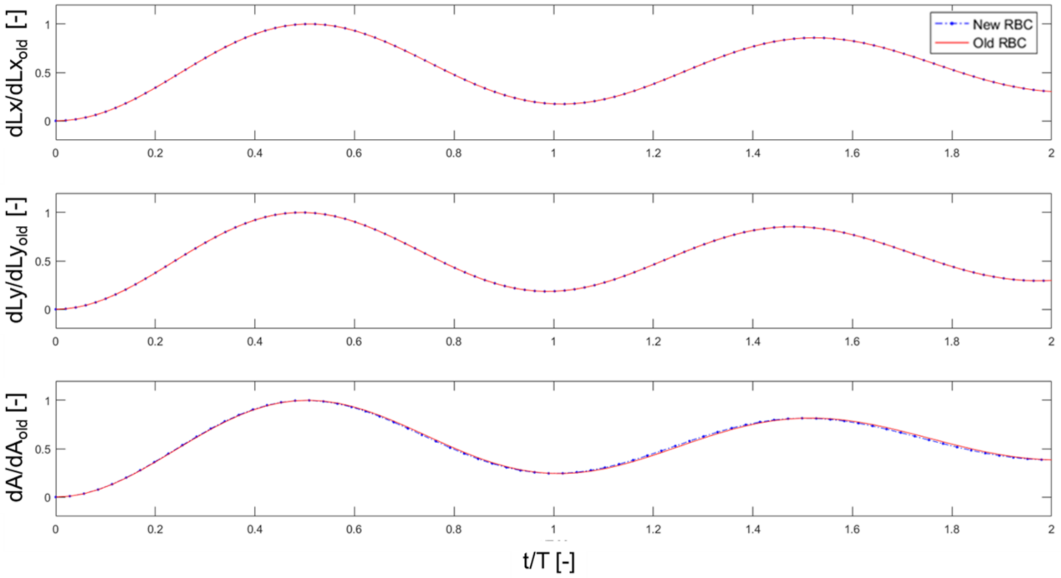


Figure 4.9: Test case 7: Linear and angular displacement along x- and y-axis and about z-axis respectively.

5

The CFD solver *fluidity*

The CFD solver used in this project is the open-source CFD code *fluidity* developed by the Applied Modelling and Computation Group (AMCG) at Imperial College London. It is capable of numerically solving the Navier-Stokes equations on both structured and unstructured 1-, 2- and 3-D finite element meshes [22]. Its strength lies on the possibility to use both finite element and control volume discretization methods and also combinations of them, as is the case for this project (see Section 5.2). *Fluidity* is also capable of dealing with multi-material problems and it is coupled with a mesh optimisation library (*libmba2d* for 2D and *libadaptivity* for 3D) that allows the use of adaptive meshes [22]. Moreover, a python interface is implemented making the coupling between fluidity and the python code described in Section 4 easier. It also allows to prescribe fields, set initial conditions and tune some parameters by means of simple python scripts which has revealed to be a very useful feature for many different purposes.

The chapter is structured as follow: firstly, the multi-material approach is introduced together with the governing equations of purely hydrodynamic problems. Secondly, the adopted discretization pair is described. Then, the immersed body method used to model the floater is presented. Finally, mesh optimization is discussed and the main parameters related to adaptive remeshing in *fluidity* are introduced.

5.1. Multi-material problems and governing equations

In this project, the fluid-structure interaction of a floating cylinder in a numerical wave tank is analysed. In order to allow the motion of the floater, a domain containing both air and water has to be used and thus, a multi-material simulation has to be performed. Multi-material simulations are known to be challenging due to the presence of an interface between the two materials, or fluids in this case. It is important to be able to track the interface development and also to avoid unphysical mixtures at the interface, e.g. bubbles of one material suspended into the other. A lot of different interface tracking techniques are described in literature, however, they can be divided into two main categories, interface-tracking methods and interface-capturing methods respectively [19].

Interface-tracking methods rely on an explicit representation of the interface which is then localized by means of interface markers. They present problems whether the interface can substantially deform [19].

Interface-capturing methods make use of a scalar field to implicitly describe the interface and they are suitable also for cases where a large deformations of the interface is present. Methods that rely on implicit description of the interface are, for example, volume of fluid method (VOF) [10] [22] [3], phase field method (PF) [22] and level-set method [19].

Fluidity presents two different interface tracking methods, a VOF based method, called multi-material approach in *fluidity*, and PF based method, called multi-phase approach and currently under development [22]. The multi-material method, the approach used in this work to track the interface, has the advantage that, for incompressible flows, exact conservation of mass is guaranteed [19].This approach assumes all the materials to be immiscible and separated by a sharp interface which also allows the use of a single velocity field to describe the flow and consequently, the need to solve only one momentum equation [22]. In order to be able to solve the conservation equation for multiple material flows, the bulk properties of the fluid in each point have to be define as [22]:

$$\rho = \sum_{i=1}^{n} \varepsilon_i \rho_i \tag{5.1a}$$

$$\mu = \sum_{i=1}^{n} \varepsilon_i \mu_i \tag{5.1b}$$

where ε_i, ρ_i and μ_i are the material volume fraction, the density and the viscosity of the i^{th} material of a flow that comprehends n materials. This definition of density and viscosity implies that the local dynamics of the flow can be determined either by the properties of a single material or by the average of the properties of different materials [3]. This last possibility represents what concerns the region of the domain close to the interface. The transport of the ε_i, $i = 1,..,n-1$, can be described by solving $n-1$ advection equations (5.2) while the ε_n is determined with equation 5.3.

$$\frac{\partial \varepsilon_i}{\partial t} + \nabla \cdot (\varepsilon_i \mathbf{u}) = 0 \tag{5.2}$$

$$\varepsilon_n = 1 - \sum_{i=1}^{n-1} \varepsilon_i \tag{5.3}$$

The dynamics of a viscous flow is generally described by the Navier-Stokes equations. These, in their conservative form and for a compressible fluid, are written as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{5.4a}$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u} - \bar{\bar{\sigma}}) = \rho \mathbf{F} \tag{5.4b}$$

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot (\rho E \mathbf{u} - \bar{\bar{\sigma}} \mathbf{u} + \mathbf{q}) = \rho \mathbf{F} \mathbf{u} \tag{5.4c}$$

where $\mathbf{u}$ is the fluid velocity vector, ρ is the fluid density, $\bar{\bar{\sigma}}$ is the stress tensor, $\mathbf{F}$ is the source term, $\mathbf{E}$ is the total specific energy and $\mathbf{q}$ is the heat flux.
For a purely hydrodynamic problem, the equations of motion of the fluid can be simplified considering only continuity (5.4a) and momentum (5.4b) equations and assuming the fluid to be incompressible [32]. Introducing also the definition of the bulk properties for multi-material problems, Equations 5.1a and 5.1b, then the Navier-Stokes equations for purely hydrodynamic problems involving multiple immiscible materials can be expressed as:

$$\nabla \cdot (\varepsilon_i \mathbf{u}) = 0 \tag{5.5a}$$

$$\varepsilon_i \rho_i \frac{\partial \mathbf{u}}{\partial t} + \varepsilon_i \rho_i (\mathbf{u} \cdot \nabla) \mathbf{u} = -\varepsilon_i \nabla p + \nabla \cdot (2 \mu_i \varepsilon_i \bar{\bar{S}}) + \mathbf{F} \tag{5.5b}$$

where $\bar{\bar{S}}$ is the deviatoric part of $\bar{\bar{\sigma}}$ resulting from the extraction of the pressure term, $-\nabla p$, from the stress tensor.

5.2. Discretization

Apart from some particular conditions, to solve the Navier-Stokes equations use has to be made of a numerical discretization scheme. Various numerical schemes which make use of a mesh are described in literature and, in general, they can be divided into three groups, namely finite difference methods, finite or control volume (CV) methods and finite element methods [3]. Finite difference methods are typically suitable for structured grids where the nodes are aligned with the system of coordinates. On the other hand, finite volume and element methods are also suitable for unstructured grids as the arbitrary distribution of the nodes

in the domain is integrated in their formulation [3].
In this project, both the continuous Galerkin (CG) finite element discretization method and the zero order discontinuous Galerkin (DG) method or finite volume method, are used to discretize pressure and velocity. In both cases, CG and DG, the discretization process starts with the definition of the weak form of the partial differential equations that have to be solved. The weak form of a differential equation can be obtained by multiplying it with a test function and then integrating over the domain. Integration by parts is successively performed so that the required smoothness is lowered. At this stage, boundary conditions are introduced in the formulation while integrating by parts. Solutions of this equation are called weak solutions and they only need to have a continuous first derivative. Of course, for such a solution, if it doesn't satisfy extra smoothness conditions, then the strong form of the equation is not well defined as it incorporates second derivatives of the field of interest.
In finite element discretization, a solution of the weak form of the equation is searched only in a finite-dimensional subspace. Consequently, a solution c can be written as a linear combination of a finite number of trial functions ς as [22]:

$$c(x) = \sum_i c_i \varsigma_i(x) \tag{5.6}$$

where the coefficient c_i can be expressed in a vectorial form with dimension equal to that of the trial space. The finite element methods in which the same space for trial and test functions is selected are referred to as Galerkin methods[22]. CG and DG methods both restrict the test and trial functions to arbitrary polynomials, P_N, of degree N, however, while CG consider polynomials which are continuous over and across the elements, DG uses polynomials that are continuous over the element but discontinuous across elements [23][22]. In other words, in the CG method each basis function has compact support over neighbouring elements while in the DG method each function has support only within the element [23]. Figure 5.1 shows the schematic of piecewise linear shape functions for both CG and DG while Figure 5.2 exemplifies the different approaches used by CG and DG in approximating a variable.

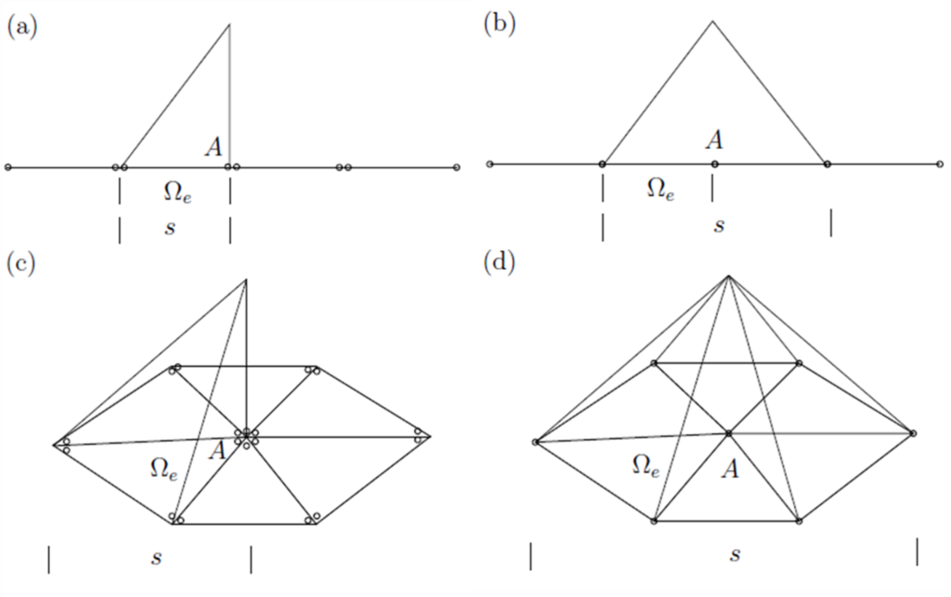


Figure 5.1: Schematic of piecewise linear discontinuous (a,c) and continuous (b,d) shape functions in one (a,b) and two (c,d) dimensions. In both cases the shape function as a value of 1 at the node A and decreases to 0 at the neighbouring nodes. In CG the support s is extended to the surrounding nodes while for DG it is restricted to the element area Ω_e. Modified figure from [3].

From Figure 5.1 it is possible to see that in CG the solution nodes are shared in between elements (single dot at the element boundary), while in DG method these are not shared and it is possible to have multiple solution nodes coexisting at the same point in space and belonging to different elements (double dot at the element boundary) [3]. Also, the number of solution nodes in an element depends on the order of the shape function as shown by Figure 5.3. The fact that the solution nodes are not shared in DG leads to a problem: on the boundaries, where surface integrals have to be computed, test and trial functions are not defined and the

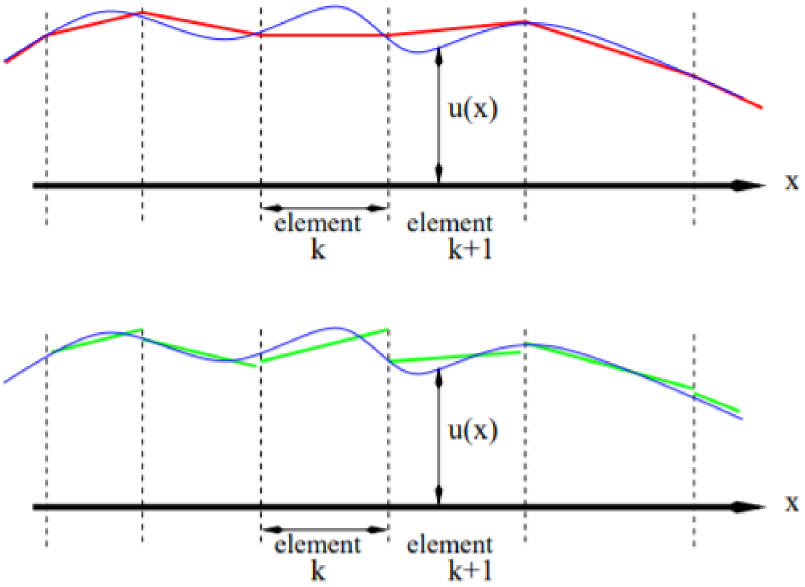


Figure 5.2: In CG method, an arbitrary variable $u = u(x)$ is approximated globally in a continuous way. On the other hand, in DG method, the same variable is approximated globally in a discontinuous manner and locally in a continuous way.

same happens with their derivatives [3]. The way DG tackles this problem consist in dealing with the undefined surface integrals as boundary conditions on the element itself. Here the advantage is that the elements in the domain can be coupled with each other by adopting the values of neighbouring elements as boundary condition for an element [3]. Of course, whenever an element has no neighbouring elements, i.e. it is at the domain boundary, the value used is that of the boundary condition. However, while dealing with derivatives, the procedure becomes more complex and usually an auxiliary variable needs to be introduced and different approaches can be used to either implicitly or explicitly do so. For a more detailed discussion about it the reader is referred to [3].

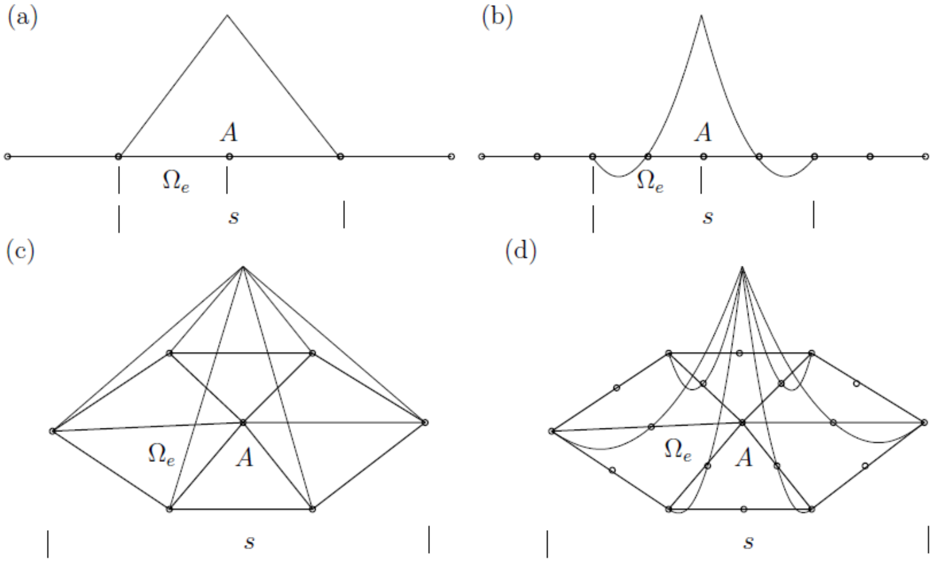


Figure 5.3: An example of the schematic of piecewise linear (a,c) and quadratic (b,d) continuous shape functions in one (a,b) and two (c,d) dimensions [3].

When constant shape functions are used, the DG method practically reduces to an upwinded finite volume method [3] and this is the discretization method used in this project to discretize the velocity. Figure 5.4 shows the schematic of the finite element method. Here, it is possible to see how the support, s, corresponds to the element region, e, as for the other DG methods and that the shape function is equal to 1 everywhere but at the boundaries where it has a value of 0.

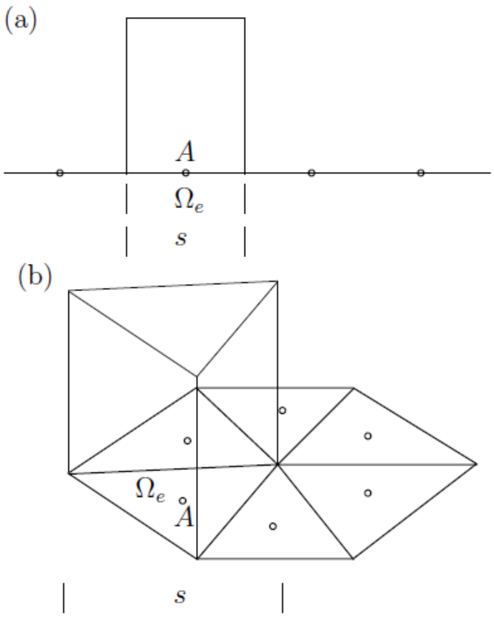


Figure 5.4: Schematic of the finite volume method in one (a) and two (b) dimensions [22].

In this work, also the pressure is discretized with a finite volume method, however, the approach used here is different and referred to as control volume discretization in *fluidity*. The control volume method is based on a dual mesh which is constructed over a finite element mesh, called parent mesh, that, in the specific case of this work, makes use of continuous linear shape functions. In three dimensions, the dual mesh is built connecting element and face centroids to the edge midpoints [22]. Figure 5.5 shows the difference between a finite volume mesh and the dual mesh used in the control volume discretization.

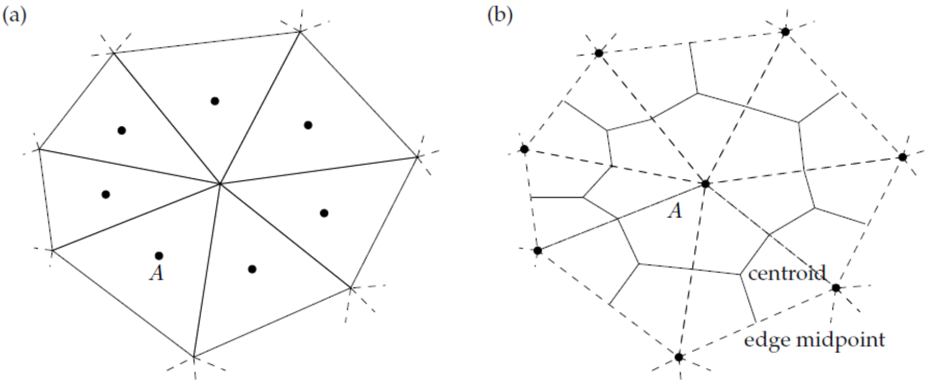


Figure 5.5: Difference between a finite volume mesh (a) and a control volume mesh (b) constructed over a finite element mesh (dashed lines in (b)) [22].

On the dual mesh, as for the DG method, the advection-diffusion equation has to be integrated by parts to be well defined [22] as the integral of the velocity divergence cannot be evaluated when the velocity is piecewise constant [3]. Moreover, all the terms involving derivatives of c disappears as now it is represented by constant functions. Here, at the element boundaries the same problem introduced for DG is also present and it is again solved by imposing boundary conditions at the element boundaries based on the values of neighbouring elements. For a more detailed description of the method the reader is referred to [22].

5.2.1. Velocity and pressure discretization pairs

As aforementioned, various spatial discretization methods are available in *fluidity*. Precisely, continuous and discontinuous Galerkin are available to discretize the velocity while continuous Galerkin and control volume, or finite volume, can be used for the pressure. Both continuous and discontinuous Galerkin allow the use of arbitrary degree polynomials and they are indicated as P_N where N is the polynomial degree. Control volume discretization is indicated as P_{1CV} [22]. However, not every velocity-pressure discretization pair is appropriate and thus available in *fluidity*. For instance, all the $P_N P_N$ pairs, same discretization order for both velocity and pressure, suffer from spurious pressure modes due to the fact that they do not satisfy the LBB stability condition [22], see [6] for a detailed description of it. The element pairs available in *fluidity* are [22]:

- $P_1 P_1$: both velocity and pressure are piecewise linear and the pressure equation needs to be stabilized. It is the only unstable element pair available in *fluidity*. Simple and thus very popular.
- $P_{1DG} P_2$: the velocity is linear and discretized with discontinuous Galerkin while the pressure is quadratic. It is stable and in *fluidity* it is recommended for large oceanic simulations.
- $P_0 P_1$: lower order version of the $P_{1DG} P_2$ and thus less accurate but computationally cheaper.
- $P_0 P_{1CV}$: it is a $P_0 P_1$ discretization with the advantage that the velocity is divergence free in the advection equation for P_{1CV} tracers. This is due to the fact that P_{1CV} test functions are used to test the continuity equation. It is consequently the element pair used for multi-material problems and thus adopted in this work.
- $P_2 P_1$: it is stable and suggested for problems where the viscosity term is dominant.

The element pair used in this work to spatially discretize the velocity and pressure is the $P_0 P_{1CV}$. It is a mixed finite element/control volume method in which the velocity is discretized with a zero order discontinuous Galerkin approach. The advantage of this element pair is that the velocity is exactly divergence free in the advection equation of the tracers (5.2) because the continuity equation is tested with P_{1CV} test functions [22] [32].

5.3. Immersed-body method

Fluid-structure interactions (FSI), in general, can be modelled with two different approaches, the mesh-conforming defined-body method and the immersed-body method. The first approach models the interactions between body and fluid by excluding the body from the computational domain and solving the fluid dynamics only in the region covered by the fluid [32]. In other words, the physical domain presents a hole where the body is located and the presence of the solid is simulated by the boundaries of the domain itself. This method shows good results when the body, which is immersed in the fluid, is fixed, however, when dealing with moving structures as in this project, the mesh-conforming defined-body method may lead to highly distorted grids and consequent numerical diffusion [32]. Moreover, a moving body would require continuous remeshing in the case of the defined-body method increasing the computational demand. By contrast, the immersed-body method does not require remeshing at every time step as the equations of motion are solved on an extended mesh, Ω, which covers both fluid, Ω_f, and solid, Ω_s regions such that $\Omega = \Omega_f \cup \Omega_s$. The function of the solid mesh is to define the geometry of the body and locate the body in the fluid mesh. Intersections between Ω_f and Ω_s are found by superposition and in the region where the body is located a penalty term is applied in the fluid mesh to relax fluid and structural velocity to one another and simulate the presence of the body.

In the immersed-body method, the Navier-Stokes equation are therefore modified to take into account both fluid and solid velocity and to include the relaxation between these two. The first goal is achieved by defining a monolithic velocity such as [33]:

$$\mathbf{v} = \alpha_f \mathbf{v}_f + \alpha_s \mathbf{v}_s \tag{5.7}$$

where $\mathbf{v}_f$ is the fluid velocity, $\mathbf{v}_s$ stands for the solid velocity and $\alpha_f = \Omega_f/\Omega$ and $\alpha_s = \Omega_s/\Omega$ are the fluid and solid concentration fields respectively. On the entire domain, the condition $\alpha_f + \alpha_s = 1$ is satisfied by definition [33]. Figure 5.6 shows the schematic of the computational domain (left) and the fluid mesh where a solid concentration field is used to represent the body (right).

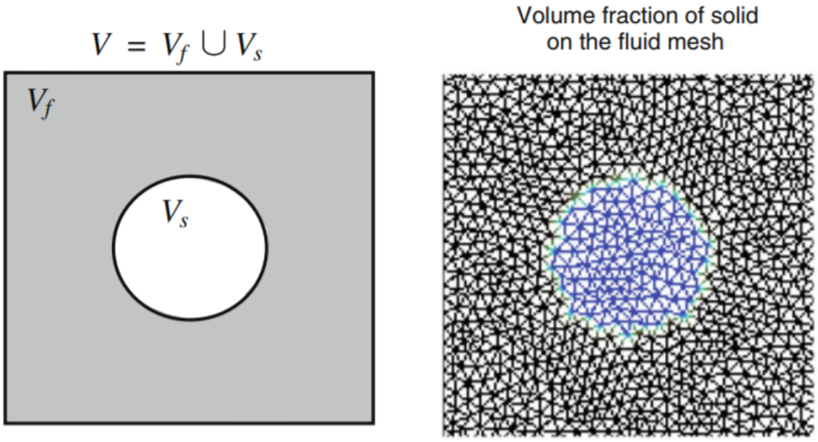


Figure 5.6: Schematic of the computational domain for the immersed body method (left) and example of the fluid mesh containing a solid concentration field used to represent the body (right). Modified figure from [33].

In order to accomplish the second objective, a penalty term, in the form of a force **F**, is added in the momentum equation. Consequently, for a purely hydrodynamic problem, Equation 5.4 can be rewritten as [32]

$$\nabla \cdot \mathbf{v} = 0 \tag{5.8a}$$

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \rho(\mathbf{v} \cdot \nabla)\mathbf{v} = -\nabla p + \nabla \cdot (2\mu \bar{\bar{\mathbf{S}}}) + \mathbf{B} + \mathbf{F} \tag{5.8b}$$

where

$$\mathbf{F} = \beta \alpha_s (\mathbf{u}_s - \mathbf{u}) \tag{5.9}$$

The penalty force, **F**, weakly imposes that solid and monolithic velocity are equal in the structure. Here, the relaxation factor is defined as [33]

$$\beta = max\left(\frac{\rho_f}{\Delta t}, \frac{\nu}{L^2}\right) \tag{5.10}$$

where Δt is the time step and L is the local edge length. The function of β is to determine the rate at which fluid and solid velocities converge to an equal value at the interface. As in [32] and [33], it is assumed that inertia effects are dominating with respect to viscous effects, $\rho_f/\Delta t \gg \nu/L^2$, and thus $\beta = \rho_f/\Delta t$. From Equation 5.9, it is clear that the penalty term is nonzero only over (and in the proximity of) the body where α_s is nonzero as well.

The solid concentration field is computed by projecting a unitary field from solid to fluid mesh through a Galerkin projection. Moreover, α_s is used to locate the body on the fluid mesh [34]. The computation of the solid concentration field starts with finding the intersections between structural and extended mesh and constructing a so-called super-mesh, S. Successively, the unitary field is projected from structural- to super-mesh via Galerkin projection and α_s^S is obtained [34]. Finally, α_s^S is projected on the extended mesh through a second Galerkin projection.

5.4. Adaptive remeshing

As aforementioned, *Fluidity* is coupled with a mesh optimization library which has the capability to adapt the mesh to specific, user-defined flow features. In principle the mesh can be adapted to any field, independently on its nature (scalar, vectorial, constant, variable, computed, prescribed, etc.) and multi-field adaptation is allowed. A typical application of the adaptive remeshing is the tracking of vortices in a flow. In this work, the use of the mesh adaptivity as a tool to capture the interface between different materials is explored. Mesh adaptivity is typically performed in three steps, metric formation, mesh optimization and mesh interpolation. In this section, firstly, these steps are briefly presented, successively, the adaptivity parameters which can be tuned by the user in *fluidity* are described.

5.4.1. Mesh optimisation

Given an initial simplex mesh, the optimization library of fluidity attempts to improve the local mesh quality going through every element one-by-one. Typically, mesh optimization is performed on simplex meshes as they permit local reshaping while cube meshes only allow global modifications where hanging nodes are not permitted. Many topological operations, which involve a local cluster of elements, are allowed, namely, vertex movement, edge splitting, edge collapsing, edge swapping. All the permitted different operations are graphically explained in Figure 5.7.

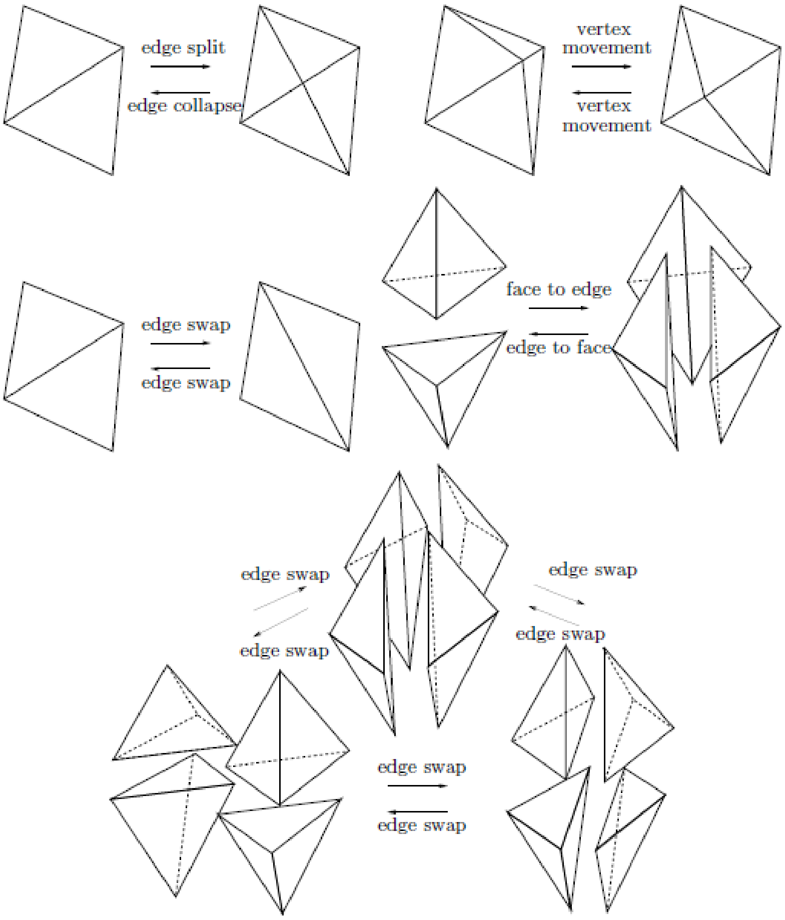


Figure 5.7: Graphic representation of the topological operations that are allowed in fluidity [3].

Performing one or more of these operations, the mesh quality can be locally modified. In order to ensure that this modification represents an improvement, a mesh quality functional has to be defined [3]. For the three dimensional case, the optimization library of fluidity defines this as [3]:

$$Q_{\Omega_e}|_M = \left(\frac{1}{2\sqrt{\sigma}r_{\Omega_e}|_M} - 1\right)^2 + \frac{1}{2}\sum\left(l_{\partial\Omega_{ek}}|_M - 1\right) \tag{5.11}$$

where $l_{\partial\Omega_{ek}}$ and r_{Ω_e} are the edge length k and the radius of the inscribed sphere of element Ω_e respectively. Two terms can be identified in Equation 5.11, the first is linked to the shape of the element and it equals zero for equilateral tetrahedra, the second term instead is related to the element size and it equals zero when all the elements have unitary edge length. Direct use of Equation 5.11 would however lead to a uniform mesh in which all the elements have unit edge length [3]. Thus, it is necessary to evaluate Q_{Ω_e} in a non-Euclidean spatially varying metric, M, which guides the optimization process. Given a user-defined error bound, the metric is constructed in such a way that if the error in an element is larger (or smaller) than the error bound then an edge length larger (or smaller) than 1 is associated to that element [3]. Now, Equation 5.11 can be evaluated on the metric to ensure:

$$\|\mathbf{k}\|_M = \sqrt{\mathbf{k}^T M \mathbf{k}} = 1 \quad \forall \mathbf{k} \in \text{Mesh} \tag{5.12}$$

In this way, all the elements of the metric that have non-unity edge length are optimized in order to obtain a uniform metric and satisfy Equation 5.12. The mesh is then optimized when all the elements have unity edge length with respect to the metric [9] and this allows the optimized mesh to be anisotropically refined.

5.4.2. Metric formation and norm selection

Given a continuous field, f, and its representation on the mesh, f_h, the interpolation error, ϵ, that is used to generate the metric is generally defined as $|f - f_h|$ [22]. This interpolation error can be evaluated with different type of norms, i.e. L_∞-, L_R- and L_p-norm. By making use of the L_∞-norm, the metric M_∞ takes the form [3]:

$$M_\infty(\mathbf{x}) = \frac{|H(\mathbf{x})|}{\epsilon(\mathbf{x})} \tag{5.13}$$

where $|H(\mathbf{x})|$ is defined as [9]

$$|H(\mathbf{x})| = Q(\mathbf{x})^T |\Lambda(\mathbf{x})| Q(\mathbf{x}) \tag{5.14}$$

where,

$$|\Lambda(\mathbf{x})|_{ij} = \begin{cases} |\lambda_i(\mathbf{x})| & i = j \\ 0 & i \neq j \end{cases} \tag{5.15}$$

with λ_i and Q being the eigenvalues of H and the matrix of normalized eigenvectors respectively [9]. Thus, the mesh is refined in those regions where there is a high curvature of the field (or high eigenvalues) and coarsened in the areas of low curvature of the field. A problem related to this specific norm is that it tends to focus on large-scale dynamics [22] since the local magnitude of the field is not included in the formulation of M_∞ [9]. In order to solve this issue and give more weight to small-scale dynamics, the L_p-norm, $p \in [1, \infty)$, has been introduced. By adopting this norm, the new metric, M_p, is formulated as [9]

$$M_p(\mathbf{x}) = (\det(|H(\mathbf{x})|))^{-\frac{1}{2p+n}} M_\infty \tag{5.16}$$

where n is the physical dimension of the problem. With the introduction of determinant of the Hessian, the metric now takes into account the local magnitude of the field into its formulation [9]. Moreover, the influence of this term can be tuned by changing the parameter p, in fact, if p is small, then a high weight is associated in the metric to small-scale phenomena [9]. In this work, the L_2-norm ($p = 2$) is used as suggested in the manual of *fluidity* and numerous papers in literature. This is done in order to be able to take into account both small-scale and large-scale phenomena as in this work they are both important. In a three dimensional problem, for this choice the metric takes the form:

$$M_2(\mathbf{x}) = (\det(|H(\mathbf{x})|))^{-\frac{1}{7}} M_\infty \tag{5.17}$$

As aforementioned, the mesh can be adapt to multiple fields at the same time and for any of these an interpolation error bound has to be provided. When n fields are selected, then n metrics, $M_{f_1}, M_{f_2}, .., M_{f_n}$, are formed and the final metric, M, is obtained via superposition of the n metrics [9][3]. Basically, the minimum local edge length present in the different metrics is used to form the final single combined metric [3].

5.4.3. Mesh interpolation

Once the mesh optimization process has been performed, two meshes exist, the original mesh, mesh I, and the optimized one, mesh II. They both describe the same domain but they do not share information. In fact, mesh I is optimized for the condition at the previous adaptivity timestep and contains all the information from that time until the current time while mesh II is optimized for the current condition but it does not contain any information regarding the fields. Thus, it is necessary to transfer the information from mesh I to mesh II and this can be done via consistent interpolation or Galerkin projection[3]. Consistent interpolation is simple, fast and bounded but it is not conservative and, consequently, the integral of a field over the entire domain is not preserved when moving from mesh I to mesh II. Moreover, when dealing why discontinuous meshes, the field is smoothed during the transfer as all the co-located nodes will assume the same value [3]. Conservation can be achieved if Galerkin projection is used to transfer the information from one mesh to the other. However, this requires the construction of a mesh containing all the nodes and edges of both the original and optimized mesh, this mesh is usually called supermesh [3]. Once the supermesh is constructed, the information can be transferred to the new nodes by means of interpolation of the basis function of mesh I on the supermesh. This process is now conservative as the supermesh contains both nodes from mesh I and II. Finally, a second projection is performed, from the supermesh to the new mesh, through which only nodes and edges from mesh II are conserved. This process guarantees conservation of the integral of the transferred field however, it does not necessarily ensure the boundedness of the field itself. However, a bounded version of the Galerkin projection is available in *fluidity* and it is described in [4] and [3]. In this project, the bounded version of the Galerkin projection is used for the transfer of the material volume fraction described in Section 5.3 from Mesh I to Mesh II. Galerkin projection is computationally demanding with respect to consistent interpolation as it requires the construction of a supermesh however, it is used in this work because it is very important to ensure a conservative and bounded transfer of the material volume fraction. Moreover the bounds of this field are known as it has to be in between 0 and 1 by definition and this makes the use of the bounded Galerkin projection straightforward. In general, velocity and pressure fields could be transfer with both consistent interpolation and Galerkin projection however, Galerkin projection is used because a supermesh is anyway constructed in fluidity and thus, it is not computationally cheaper to use consistent interpolation to transfer the other fields.

5.4.4. Mesh adaptivity parameters

As it can be deduced from what is presented in the previous sections, the use of adaptive remeshing requires quite some effort from the user. Many parameters have to be tuned and most of them require some iterations to really understand their influence in the optimization process. In this section, the parameters that have to be set by the user in *fluidity* are presented and described. The use of mesh adaptivity requires the setting of:

- **Field selection:** first of all, one (or more) field has to be selected. In this project, the mesh is primarily adapted to the material volume fraction field, however, a multi-field adaptation involving both material volume fraction and velocity fields is explored.

- **Interpolation method:** the interpolation method for each field can be chosen between consistent interpolation, continuous and discontinuous Galerkin projection and continuous bounded Galerkin projection. In this work, continuous bounded Galerkin projection is used to interpolate the material volume fraction field and the pressure while discontinuous Galerkin projection is used for the Velocity field as it is discretized with P_0.

- **Norm selection:** a norm has to be selected between L_∞-norm and L_p-norm, $p \in [0,\infty)$. In this project, as explained in Section 5.4.2, the L_2-norm is selected.

- **Interpolation error bound:** an error bound has to be provided in order to guide the optimization process. In general, a good rule of thumb is to start with a value equal to 10% of the norm of the field [22]. This interpolation error represents the most important parameter in the optimization and ideally one should be able to obtain the right refinement by only setting this value correctly. The error bound used in this project has a constant value, however, this can also be a function of time and space and, in this case, it can be defined by making use of the python interface.

- **Gradation parameter:** this value is used to smooth the mesh generated by the optimization process. In fact, the algorithm used in the mesh adaptivity always aims at the lowest numbers of nodes by means

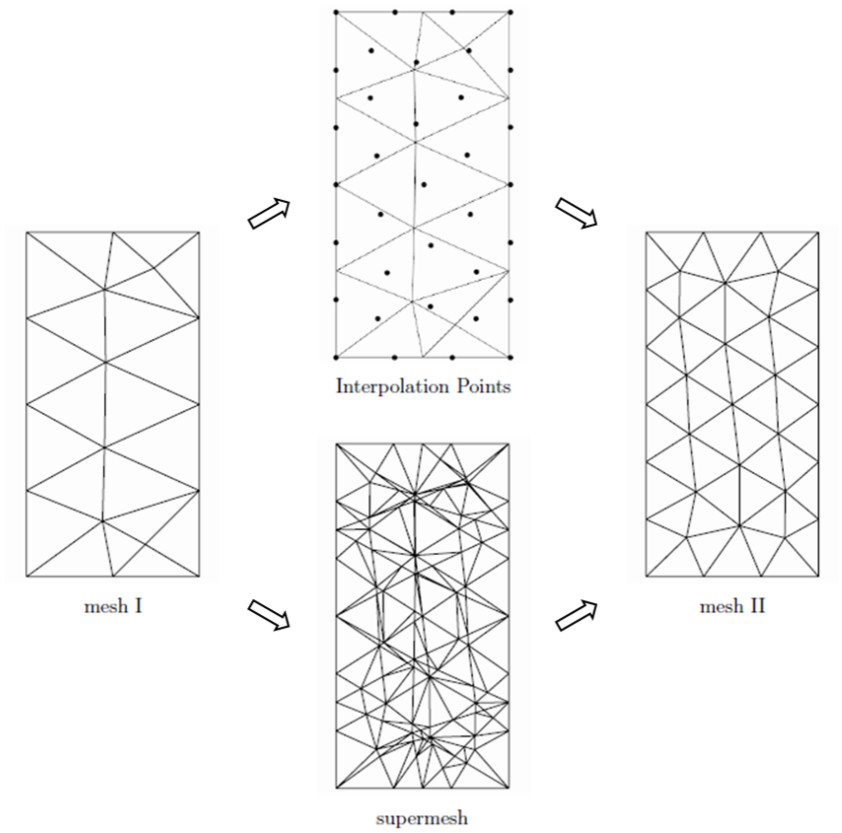


Figure 5.8: Information transfer via consistent interpolation (top) and Galerkin projection (bottom). Modified figure from [3].

of coarsening the mesh where the curvature of the field is small. This means that without an appropriate gradation parameter the mesh may be coarsened too quickly leading to jumps in the edge length which are typically undesired. By using this tool, the user can define the maximum increment in edge length for two adjacent elements. The gradation parameter can be defined as either a constant or an anisotropic constant symmetric tensor or an anisotropic symmetric tensor which is a function of the position in space. All these possibilities are explored in this work.

- **Maximum (minimum) edge length tensor:** by superimposition of a maximum (minimum) metric with the optimized metric the edge length can be bounded. A more detailed explanation of this procedure can be find in [3]. This symmetric tensor is usually anisotropic and can be either constant or function of the position in space. In this project, the use of an anisotropic tensor which is function of space is use in order to locally limit the maximum edge length in different regions of the domain depending on their nature. This will be explained in more detail in Section 6.1.

- **Maximum and minimum number of nodes:** this range can be used to set a upper limit to the number of nodes if, for instance, there is a limitation given by the available computational power or to prevent the mesh from coarsening too much, e.g. setting a lower limit. In this last case, however, the use of a gradation parameter is advisable as it gives to the user a finer control of the optimization process. Ideally, this range should be intentionally very large in order not to limit the optimization process [22]. In this work, the minimum number of nodes is arbitrarily set to 1000 while the maximum is defined as 200000 nodes times the number of cores used to run the simulation. The aim is to fix limits that will never be reached by the mesh optimization routine.

- **Metric advection:** this option can be activated in *fluidity* to advect the metric. Based on the velocity of the flow, an addition advection equation can be solved in order to move forward the mesh. This should help keeping the dynamics of interest inside of the refined region until the next optimization is performed. It can also allow the use of a larger period for the mesh adaptivity.

- **Period:** it indicates how often the mesh has to be optimized. Depending on the dynamics involved it can be necessary to optimize the mesh more or less often. In general, metric advection can help reducing the optimizations however, when large dynamics are involved, it is usually necessary to adapt every few timesteps. This value can be defined in either simulation time or number of timesteps. In this project, the mesh is adapted every 10 – 20 timesteps of the simulation as suggested in [22].

In Section 6.1, the effect of some of these parameters on the optimization routine is analysed. Moreover, advantage and disadvantage related to the use of this potentially powerful tool are explored for the case in which it is adopted to capture the interface in between air and water in a numerical wave tank.

6
Simulations and results

In this section, the different CFD simulations perform in this project, together with the setups and results, are presented. Firstly, the use of mesh adaptivity in a linear numerical wave tank as a tool for capturing the air-water interface is explored. Successively, validation of a NWT based on a structured mesh is performed for both linear and non-linear waves and results are compared with linear Airy wave theory. Finally, the capabilities of the CFD solver *fluidity* in computing the hydrodynamic coefficients of a floating cylinder are tested and results are compared with experimental data and CFD simulations performed in OpenFOAM and presented in [25] and [11] respectively.

6.1. Multi-material interface tracking in a NWT

Multi-material simulation are known to suffer from numerical diffusion when propagating the interface between the different fluids. A typical approach to deal with this issue is to refine the mesh in the proximity of this interface. The idea is that the increase in the mesh resolution should prevent the interface region from being under-resolved [3]. However, this approach considerably increases the number of nodes in the domain if the mesh resolution is kept constant everywhere else. Moreover, if a fixed mesh is used and the interface moves in time, then the resolution has to be increased in a potentially very large region of the mesh in order to keep the interface inside the refined region for the entire duration of the simulation. In this scenario, mesh adaptivity may be a very powerful tool as it would allow refinement only in the real vicinity of the interface and coarsening in those regions which are far from it. Here, there capabilities of the mesh adaptivity algorithm implemented in *fluidity* are explored in an attempt to validate a numerical wave tank with both air and water. The NWTs used in this analysis have the dimensions indicated in Figure 6.1 which are based on the wave characteristics presented in Table 6.1.

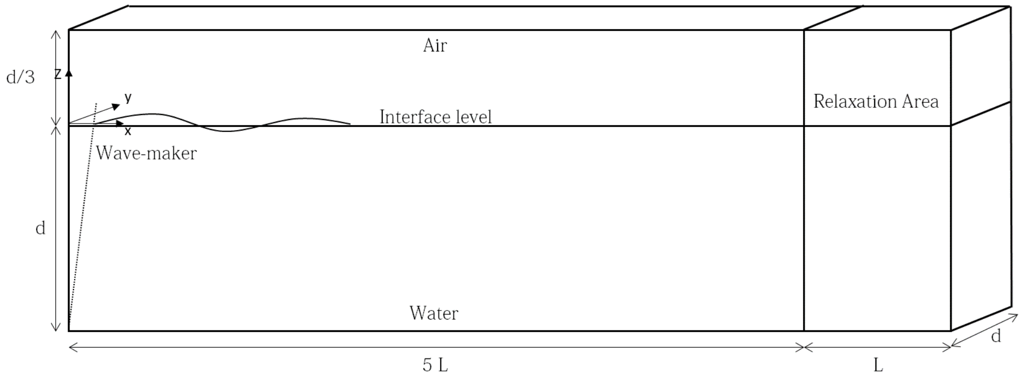


Figure 6.1: Set-up and dimensions of the numerical wave tank used in this project to simulate wave propagation.

f [Hz]	T [s]	H [m]	L [m]	d [m]	H/g T^2 [-]	d/g T^2 [-]
1.256	5.0	0.2	38.9	20.0	8.15e-4	0.082

Table 6.1: Wave parameters used in the mesh adaptivity analysis.

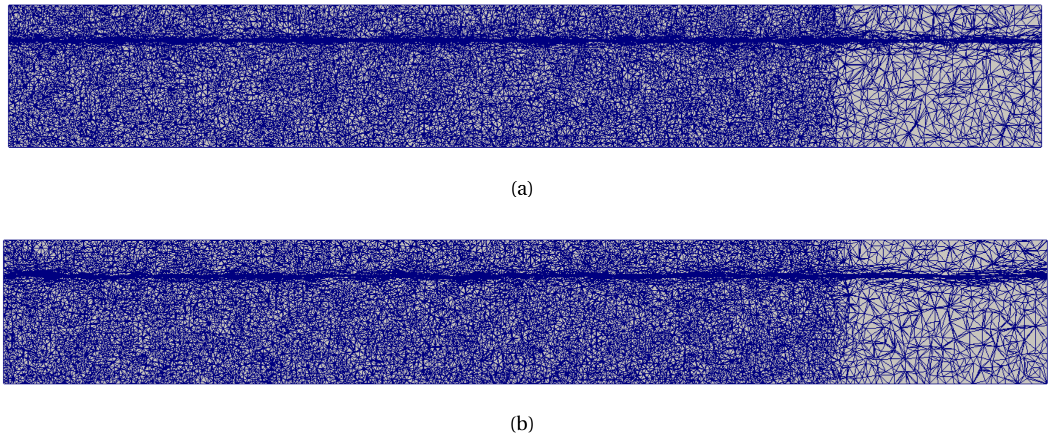
The waves generated in this simulation are linear waves in deep water condition as it can be derived from the relative wave height, H/gT^2, and the relative water depth, d/gT^2 presented in Table 6.1. These are generated with the use of a piston type wave-maker which is based on the work presented in [28]. This wave-maker is capable of generating both linear and non-linear waves up to the fifth order of the Stokes wave theory. For the case of linear waves, the wavenumber k is computed via linear dispersion relation, Equation 3.17. The wave-maker is then modelled as a weakly applied Dirichlet boundary condition on the x-component of the velocity at the inlet of the domain in the region covered by the water. The exact formulation of the imposed velocity can be found in [29]. The applied velocity presents a ramp in the first 3 second in order to smoothly apply the boundary condition at the inlet. One boundary condition has to be applied on each side of the domain for the numerical problem to be well-posed. Thus, at the inlet in the region covered by air, at the back of the domain, on the left and right side and on the bottom of the domain, a wall boundary condition is applied and this is done prescribing a no normal flow condition on the velocity field. At the top of the domain, a homogeneous Neumann boundary condition is imposed and the air can flow in and out from the domain. From Figure 6.1, it can be seen that an absorption layer is introduced at the end of the domain in the region $5L \leq x \leq 6L$. This is done in order to damp the waves and avoid reflection from the back of the domain which is modelled as a wall. The absorption layer implemented in this work have been already described in [34] and [32] and it relies on the introduction of an absorption term, σu, on the right side of the momentum equation, Equation 5.4b. The absorption coefficient, σ, is defined as [34] [32]

$$\sigma = \begin{cases} \frac{1}{4}\left(\tanh\left[\frac{\sin(\pi(4\tilde{x}-1))/2}{1-(4\tilde{x}-1)^2}\right]+1\right) & \text{if } 0 \leq \tilde{x} \leq \frac{1}{2} \\ \frac{1}{4}\left(\tanh\left[\frac{\sin(\pi(3-4\tilde{x}))/2}{1-(3-4\tilde{x})^2}\right]+1\right) & \text{if } \frac{1}{2} \leq \tilde{x} \leq 1 \end{cases} \tag{6.1}$$

where $\tilde{x} = (x - l_0)/l$, $l_0 = 5L$ is the starting point of the sponge layer and $l = 1L$ is the length of it. As aforementioned, velocity and pressure are spatially discretized with the P_0P_{1CV} pair and the mesh interpolation is performed via Galerkin projection. This is implemented in its bounded form in the case of pressure and material volume fraction with the difference that in the pressure no user defined bounds are provided but they are computed by the solver as described in [22] and [3]. In both cases, based on the findings of [4], a maximum of 1000 iterations is used to bound the solution. The material volume fraction is spatially discretized with a control volume approach. To precisely describe the air-water interface, the HyperC approach is used to limit advective fluxes while the Bassi-Rebay scheme is used for the diffusion term [32]. In order to be able to compare the waves generated in the NWT with the Airy solution, some detectors are placed in the domain every $5m$ and at different water depths. A detector is essentially a point in the domain where the fields of interest are computed and saved at every timestep. They are very useful in this kind of simulations as they provide the user with a field time series, for instance velocity or pressure time series, that can be easily compared with the analytical solution of the same field.

6.1.1. Mesh adaptivity: tuning and results

Initially the mesh is adapted to the material volume fraction only and the interpolation error ϵ is set to 0.09 (see Section 5.4.2) and it is evaluated with the L_2-norm. This essentially means that the optimization algorithm aims to achieve a 9% error in the solution by increasing the mesh resolution. Of course, an higher value would lead to a coarser mesh while a lower value would result in a finer mesh. In the entire domain, the minimum edge length is 0.01 in x- and y- direction and 0.0025 in the z-direction. This difference depends on the fact that the field to which the mesh is adapted, mainly varies in the z-direction and consequently the mesh will be finer in that direction. The maximum edge length is instead defined as a function of space. In fact, in the region occupied by the absorption layer the mesh can be coarsen as this may actually help damping the waves by means of numerical diffusion. Therefore, the maximum edge length is set as 2.0 for $x \leq 5L$ and 5.0 for $x > 5L$. An anisotropic gradation parameter is also implemented in order to control the increment in the element's dimensions. In particular, this parameter is set equal to 1.05 in the region close to the interface, a band across the interface of height equals two times the wave height, and 1.5 everywhere else. This is done because, in initial tests, it has been found that the mesh was growing too quickly still very close to the free surface due to the fact that this occupies a very tiny region of the domain, ideally it is a surface of one element of thickness. The mesh is optimized every 20 timesteps in order to keep the dynamics constantly inside the optimized region. An adaptive timestep is used and it is based on a maximum Courant number, C, of 0.8.



(a)

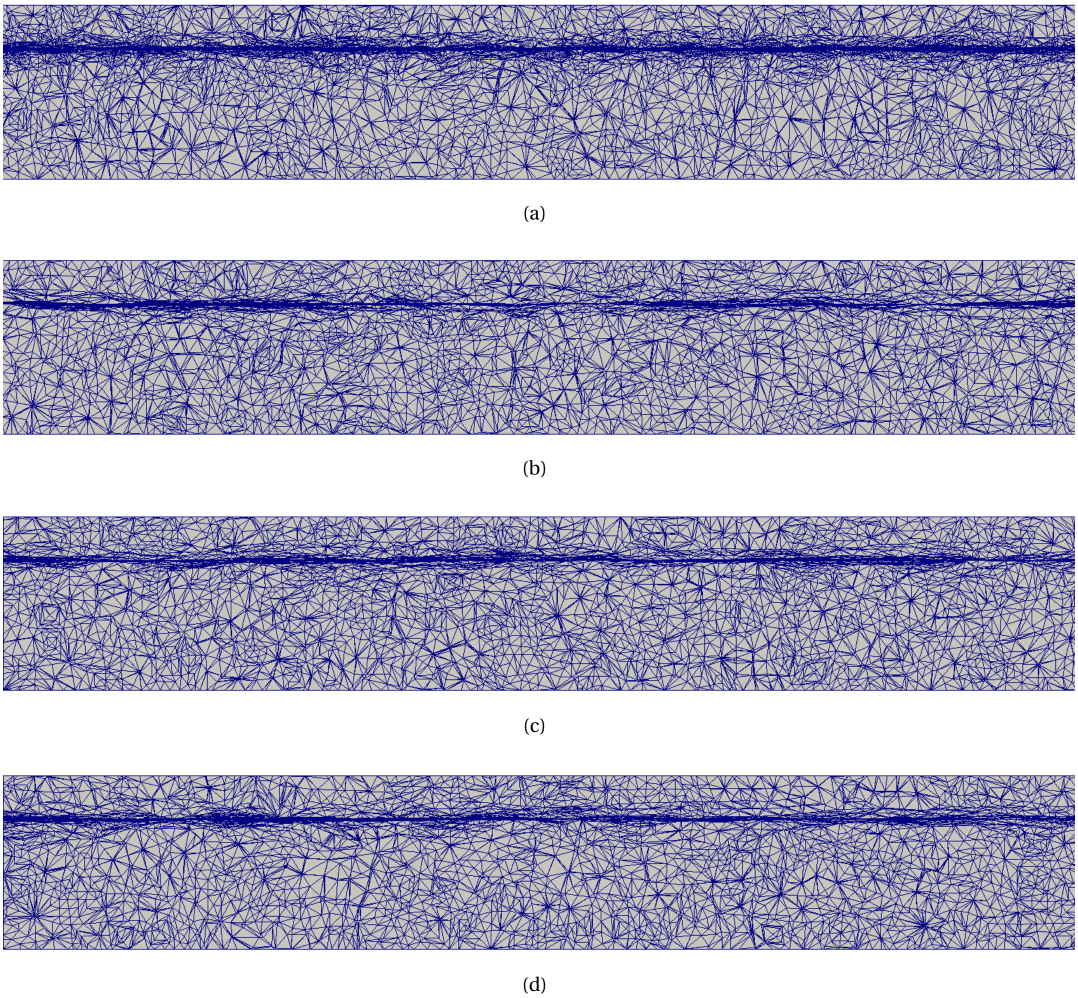
(b)

Figure 6.2: Two snapshots of the the longitudinal cross-section of the domain showing the adaptive remeshing for the case of single field adaptation and interpolation error of 8%. The figures show the mesh obtained after the first optimization cycle at $t = 0s$ (a) and at $t = 10s$ (b).

Figures 6.2(a) and 6.2(b) show the longitudinal cross-section of the mesh at the instants $t = 0s$ and $t = 10s$. It can be seen that the mesh is correctly adapting to the material volume fraction as a refinement is clearly present at the interface were the curvature of the field is high. It can be notice as well the role of the maximum edge length, in fact, in the absorption layer region, the mesh is coarser than in the first part of the domain. Moreover, focusing on the region close to the inlet, it is possible to notice that the refined region moves from being almost a straight line at $t = 0s$ to presenting some small bumps where crests and troughs of the waves are developing at $t = 10s$. However, the results obtained with this configuration are still very inaccurate because, even if the mesh is fine enough to ensure a maximum 9% error in the solution of the material volume fraction field, it is too coarse to correctly represent velocity and pressure. Another important aspect observed in the time series of the dynamic pressure, and consequently velocity, is the presence of spikes in the solution after every mesh optimization cycle (see Figure 6.6). This can be explained by the fact that, even if the pressure field is interpolated with the bounded version of the Galerkin projection, this ensures only a global boundedness of the field and thus, local deviations are allowed if there are enough nodes that can absorb this deviation [3]. In this situation, where pressure time series are extrapolated from a fix point in time, then spikes are possible. Moreover, projection from mesh I to mesh II (see Section 5.4.3) is perform with the total pressure field and this can also explain the presence of these spikes in the dynamic pressure. In fact, a large value of dynamic pressure is relatively small if compared to total pressure and thus, the value of these spikes may still be inside the bounds when total pressure is evaluated.

Based on the results of the first setup, one main modification is introduced in the setup and the mesh is adapted to both material volume fraction and velocity. From the first run, it has been noticed that the mesh was too coarse to correctly describe the dynamics already close to the air-water interface. In fact, the material volume fraction has a non-zero gradient only in a region of approximately two cells of height across the interface and this necessarily drives the optimization towards increasing the mesh size everywhere else. Thus, the introduction of the velocity field in the optimization routine aims to provide a field with a more globally varying gradient. In particular, in a numerical wave tank the velocity should have a maximum at the interface and decrease with the water depth and this should lead to a smoother grid. Initially, the metric error used for the velocity is 0.8%.

Figure 6.3 shows the mesh at four different instants in time, precisely $t = 0s$, $t = 10s$, $t = 20s$ and $t = 40s$. The first thing to be noticed is that the mesh gets very coarse very quickly, in fact, the number of nodes in the domain rapidly decreases down to 30000. This is due to the small interpolation error used to construct the metric of the velocity field. In fact, when an error bound is significantly smaller than the other, the optimization algorithm tends to prioritise the metric that requires the smaller error. The problem is that at



(a)

(b)

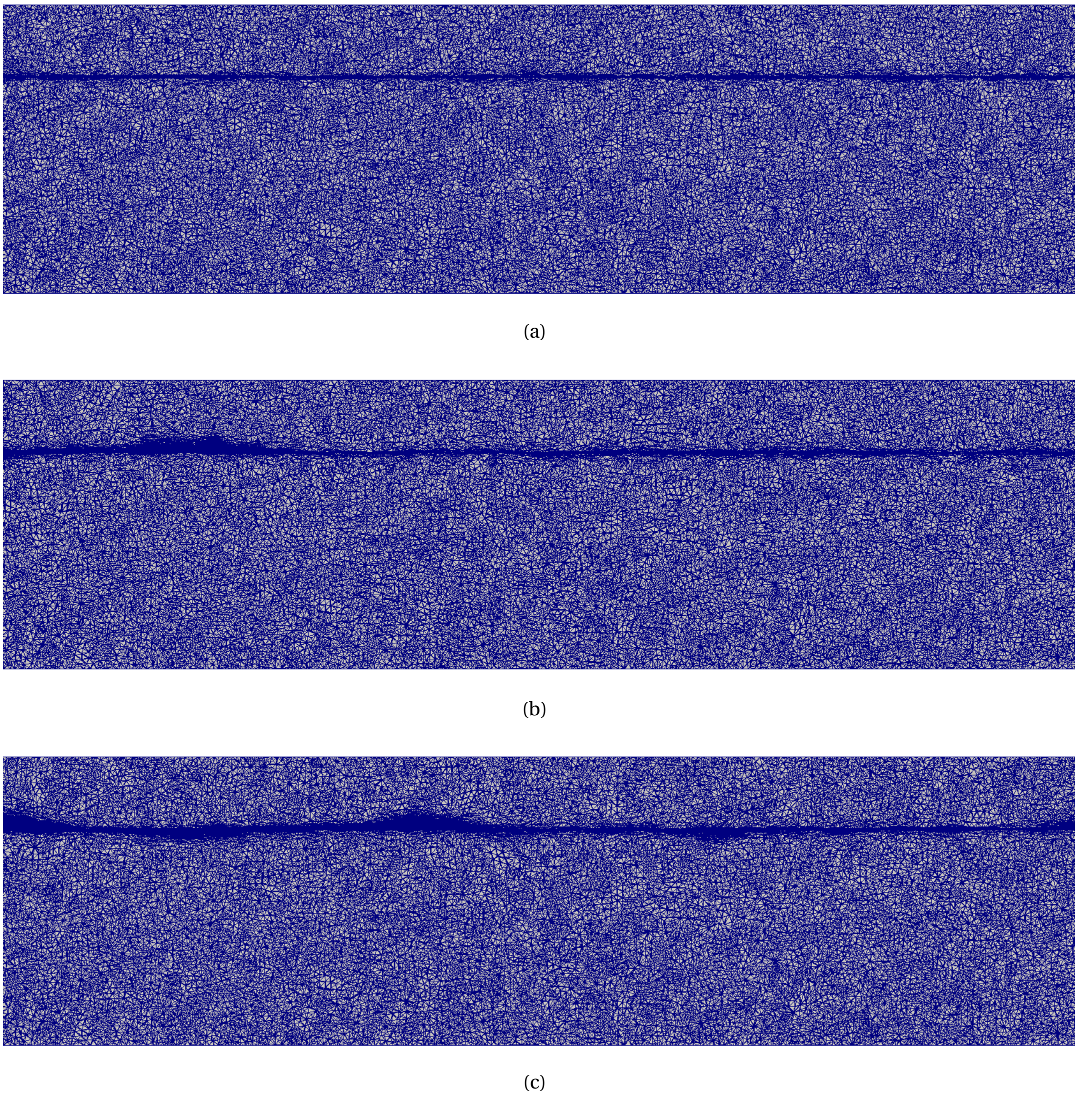
(c)

(d)

Figure 6.3: Three snapshots of the the longitudinal cross-section of the domain showing the adaptive remeshing for the case of multi-fields adaptation and interpolation error of 0.008 for the velocity field. The figures show the mesh obtained after the first optimization cycle at $t = 0s$ (a), at $t = 10s$ (b), at $t = 20s$ (c) and $t = 40s$ (d). Here the domain is zoomed to permit a better visualization of the mesh however, the absorption layer is consequently cut out from the picture.

the beginning of the simulation the velocity field is non-zero only very close to the inlet boundary and this means that, if priority is given to the velocity metric, the mesh becomes very coarse very quickly. As a result, the dynamics escape the high resolution region and the waves are numerically damped out. Consequently the mesh keeps being refined only close to the inlet and waves never propagate in the domain. This can be seen by visualizing the velocity field in the domain at $t = 40s$ and it is shown in Figure 6.5. In fact, after $40s$, with a celerity, $c = L/T$, of $7.8 m/s$, the waves should have already propagated in the entire domain.

This issue could be solved advecting the metric, however, for unknown reasons, after some iterations the solver was constantly failing in the attempt to solve this extra advection equation. Thus, in order to overtake this problem, another approach is used and the velocity interpolation error is increased from 0.008 to 0.02. Moreover, the period of the mesh adaptivity and the timestep are delinked by introducing a constant period equal to $1s$ of simulation time. This has a triple aim, it provides more control over the adaptivity period, it reduces the presence of spikes in the pressure and it can make the simulation faster. The use of an adaptivity period as function of a certain number of timesteps does not provide a great control over it if, as in this case, an adaptive timestep is used. This could be achieve also by imposing a constant timestep, however, in order to ensure $C < 1$ for the entire duration of the simulation, the use of a constant timestep is not recommended. In fact, as C is function of the mesh size, the only way to ensure $C < 1$ in the entire domain would be to compute the timestep based on the user-defined minimum edge length. However, this dimension should never be reached by the optimizer and consequently this would unnecessarily slow down the simulation. Another


(a)

(b)

(c)

Figure 6.4: Three snapshots of the the longitudinal cross-section of the domain showing the adaptive remeshing for the case of multi-fields adaptation and interpolation error of 0.05 for the velocity field. The figures show the mesh obtained after the first optimization cycle at $t = 0s$ (a), at $t = 6s$ (b) and at $t = 14s$ (c). Here the domain is zoomed to permit a better visualization of the mesh however, the absorption layer is consequently cut out from the picture.

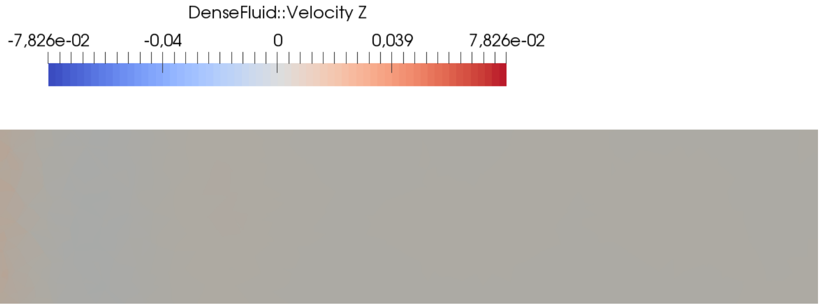
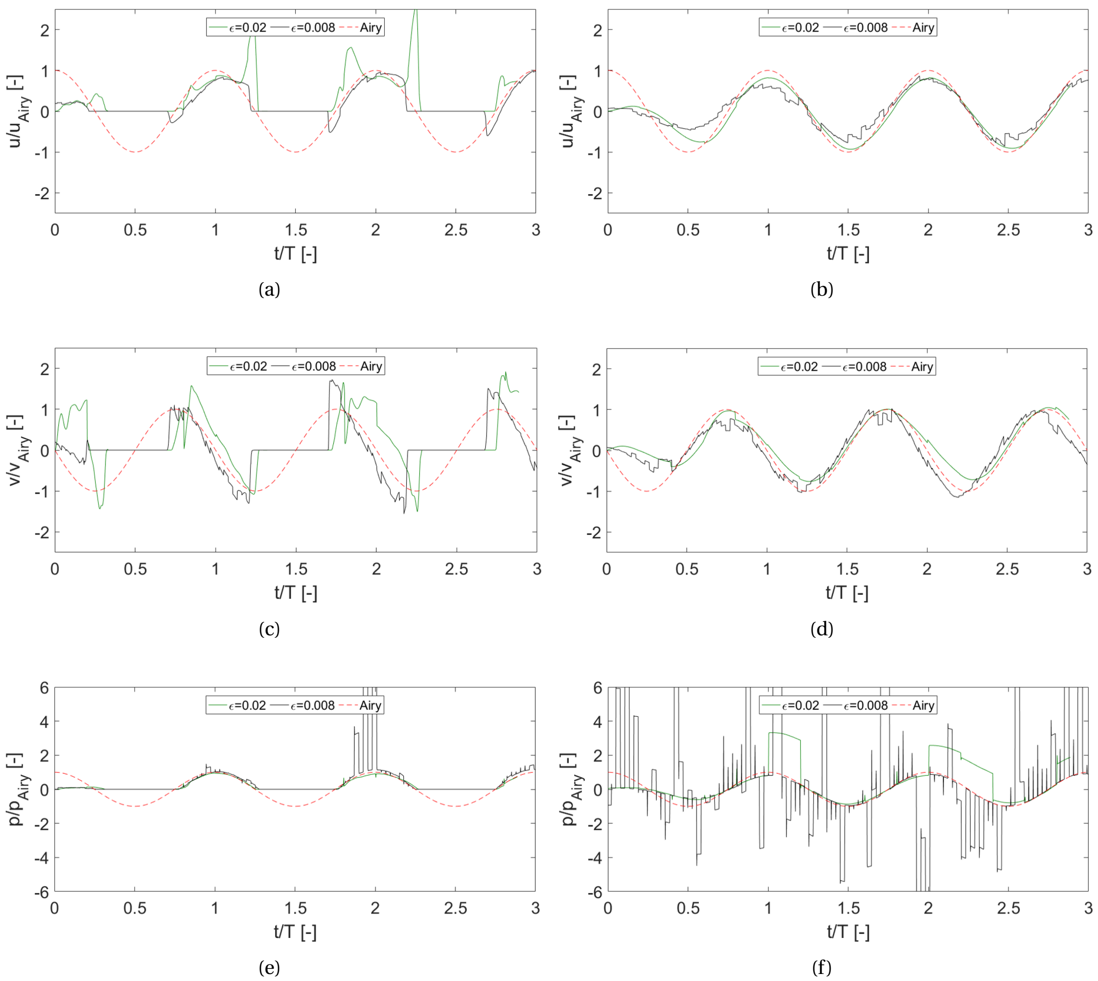


Figure 6.5: Vertical particle velocity $3m$ below the air-water interface at $t = 40s$.

aspect is instead represented by the presence of spikes in the dynamic pressure. These are undesirable as they may relevantly affect the dynamics of an hypothetical body which interacts with the wave field even though their effect may be limited if total pressure is considered. Moreover, they affect the velocity field, as pressure and velocity are linked by the continuity equation 5.4a, and this can be seen in Figure 6.6. Because of this, the adaptivity period is intentionally increased with respect to the previous simulation. The effect of this choice is evident in Figures 6.6(c) and 6.6(f). The last aspect related to the use of a constant and higher period is related to the required computational power. In fact, the first simulation was running at a ratio elapsed-time over wall-time of $7.3e-05$ despite the low number of nodes, showing a relevant impact of the mesh optimization on the economy of the simulation. For a more detailed analysis of the cost of mesh adaptivity, the reader is referred to [22].



(a)

(b)

(c)

(d)

(e)

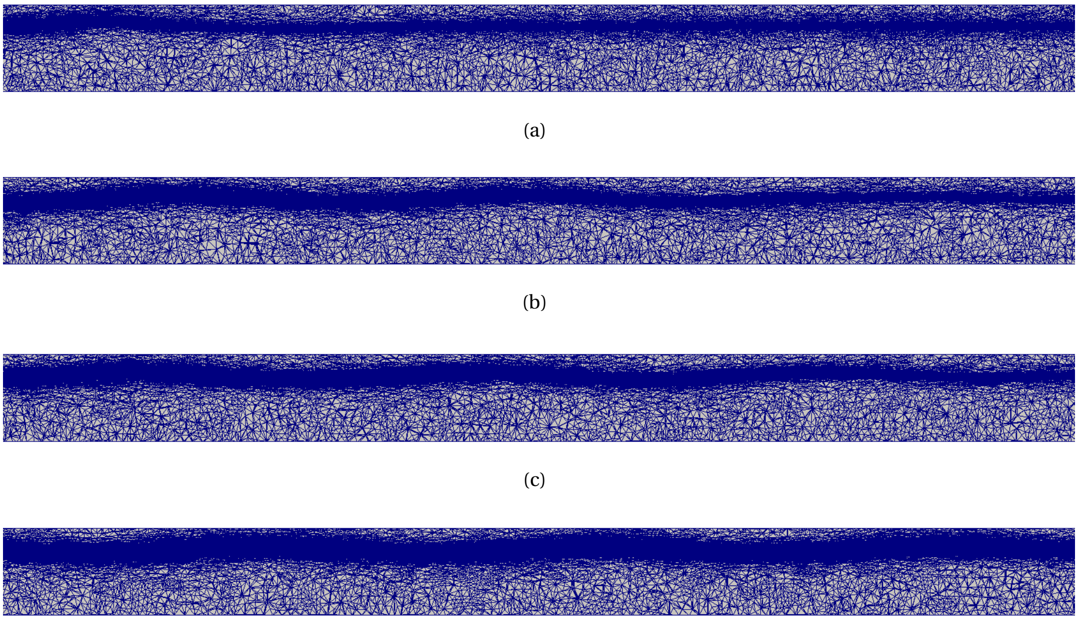
(f)

Figure 6.6: Figures (a,b), (c,d) and (d,e) present the time series of horizontal particle velocity, vertical particle velocity and dynamic pressure respectively. The set of figures (a), (c) and (d) shows velocity and pressure fields at the interface level in $P_1(0,0,0)$ while the set (b), (d) and (e) is extracted at $P_2(0,0,-1)$. Here, results from the multi-field adaptation simulations are compared with the linear Airy analytical solution.

In Figure 6.4, three snapshots of the longitudinal cross-section of the mesh at $t = 0s$ (a), $t = 6s$ (b) and $t = 14s$ (c) are shown. With respect to Figure 6.3, it can be seen that the higher metric error in the velocity has led to a finer and more homogeneous mesh thanks to the fact that now both fields play a relevant role in the optimization process. Consequently, even if the velocity is non-zero only at the inlet, now the mesh is always refined at the interface and, thanks to the gradation parameter, it results in a globally finer and smoother mesh. This is clearly visible in Figure 6.4(a) which shows the mesh after the first optimization routine, at $t = 0s$. From a computational point of view, the resulting mesh is necessarily more demanding and, in fact, the number of nodes in the domain is constantly oscillating around 450000 nodes. This is also an interesting result as it means that the optimization converges to an almost stable number of nodes from the beginning of the simulation. Now, from Figures 6.4(b) and 6.4(c), it is finally possible to clearly see the deformation of the mesh as the first generated wave propagates in the domain. Analysing the mesh, the accuracy of the solution can be already qualitatively evaluated, however, for a quantitative assessment, particle velocity and dynamic pressure profiles, extracted in specific points of the field, are compared with the analytical solution provided by the linear wave theory. The results of this comparison are presented in Figure 6.6, where the Airy solution is represented by the red-dashed line while the results provided by the simulation with interpolation error of 0.008 and 0.02 are represented by the black and green lines respectively. An attempt to estimate the error is done by defining this as:

$$error(t_i) = \left\| \frac{x(t_i) - x_{Airy}(t_i)}{max(x_{Airy}(t))} \right\| \tag{6.2}$$

where x stands for either vertical velocity or horizontal velocity or pressure and $max(x_{Airy}(t))$ is the maximum in the Airy solution for the variable x. The error is normalized with the maximum of the analytical solution to avoid infinite error when the solution approaches zero. The error for the pressure has not been estimated because of the presence of spikes in the solution which makes the computed error meaningless. Regarding the velocity, the average error evaluated at $P(0,0,-1)$ equals 12% and 26% for horizontal and vertical velocity respectively in the case of $\epsilon = 0.008$ and 8.6% and 18.2% for the case of $\epsilon = 0.02$. However, the maximum error can be even greater than 100% at the interface as it can be seen from Figures 6.6(a) and 6.6(b). Apart from the maximum error, which is very large, the results obtained with these setups are in general very poor. In fact, even if the use of a larger period for the optimization has improved the pressure solution (green line), the results obtained at the interface for the velocity are still very far from the analytical solution.



(a)

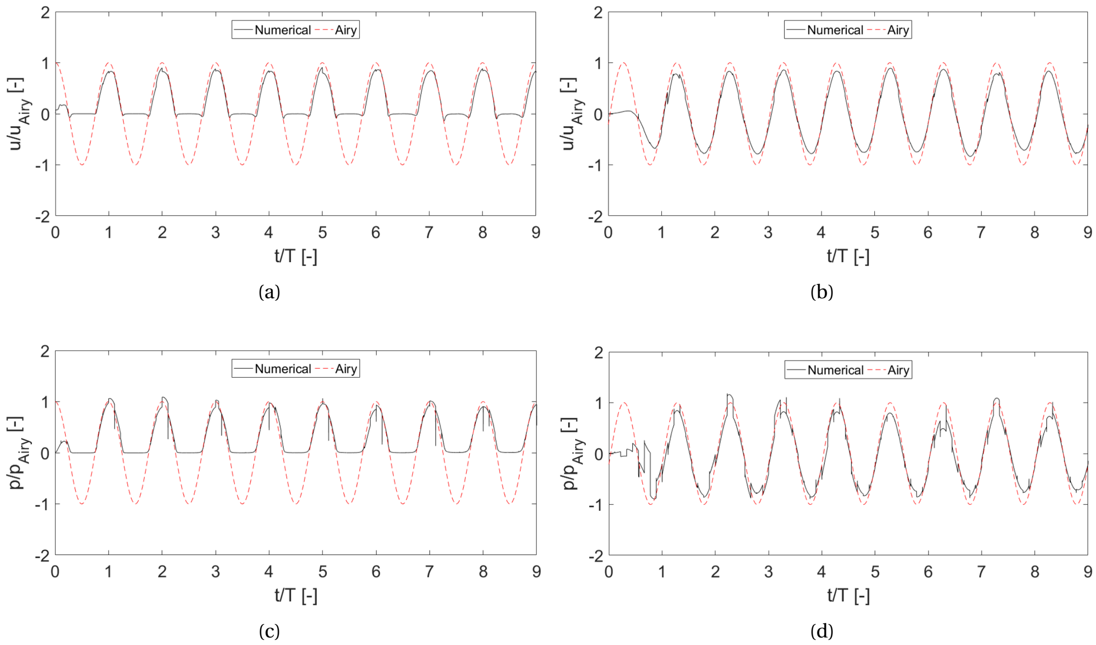
(b)

(c)

(d)

Figure 6.7: Figures from (a) to (d) show the longitudinal cross-section of the mesh at $t = 39s$, $t = 99s$, $t = 123s$ and $t = 267s$ respectively. From these, it is possible to see how the mesh resolution nicely moves to the right while the wave field propagates in the domain.

The introduction of the velocity in the remeshing process, showed that a more globally varying field leads to a generally finer mesh. This result improved the accuracy of the NWT, however, it also increased the computational cost of the simulation and led to an almost constant number of nodes. On the other side, when the mesh was adapted to only the material volume fraction, it was resulting in a too coarse mesh due to the very small region of non zero curvature of the field. Thus, one last simulation was performed to understand if higher waves can be better resolved by using mesh adaptivity. The idea is that if the domain has the same dimensions and the wave height is increased, then the region where the material volume fraction gradient is non zero becomes larger. For this purpose, waves with a wave height $H = 1.5m$ and period $T = 5.0s$ were generated at the inlet and the mesh was again adapted only to the material volume fraction with an interpolation error $\epsilon = 0.08$. This type of waves falls in the region of validity of the second order Stokes theory (see Section 3.2, Figure 3.3) and thus, they could be generated by the implemented wave-maker. Snapshots of the mesh at different stages of the simulation and results from the detectors are presented in Figures 6.7 and 6.8 respectively.



(a)

(b)

(c)

(d)

Figure 6.8: This set of figures present the velocity (a,b) and pressure (c,d) profiles at the points $P1(0,0,0)$ (a,c) and $P2(30,0,-3)$ (b,d).

As expected, for higher waves the mesh is refined in a region which is larger. However, for this type of waves, the gradation parameters in now able to maintain an acceptable resolution everywhere in the domain. The number of nodes gradually decreases from 700000 nodes at $t = 0s$ to 200000 nodes at $t = 10s$ to then settle at 60000 from $t = 25s$ on. This setup showed better results also in terms of computed velocity and pressure and in terms of numerical diffusion. In fact, while for the second setup the waves were not propagating and in the third they were losing phase, here they correctly propagate in the domain. This is shown by Figures 6.8(a) and 6.8(b) where the horizontal particle velocity at a distance of $0m$ and $30m$ from the wave-maker are respectively presented. However, still some spikes are present in the pressure due to the local unboundedness of the adopted interpolation method. Here, particle velocity and dynamic pressure are compared with linear wave theory even if they do not strictly fall in the region of validity of linear waves. However, for second-order Stokes waves if the water depth is relatively large then the non linearities lose of importance and linear wave theory can be used to describe particle velocity and dynamic pressure [7]. In other words, in deep water condition, waves that only slightly exceed the linear regime do not present higher velocities and larger pressure with respect to what is predicted by linear Airy. It should be point out that in this setup, another important modification has been introduced with respect to the previous setups. In fact, this simulation has been performed at a later stage of this project and consequently it includes the modification discussed in Section 6.2.2 regarding the approach used to limit the advective fluxes. Unfortunately, due to lack of time, the actual influence of this change on the obtained results could not be analysed.

In conclusion, the results obtained making use of the adaptive remeshing showed that some work has still to be done in order to obtain an accurate NWT. The latest results are encouraging and show that for larger waves the adaptivity algorithm of *fluidity* works better. However, the following issues have been observed:

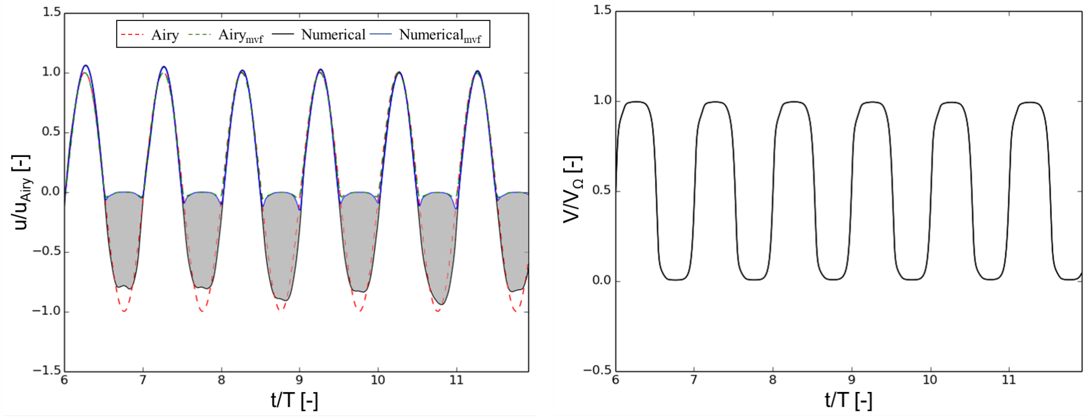
- The tuning of all the parameters related to the mesh adaptivity can be highly time consuming. Adaptive remeshing can indeed speed up a simulation once it is correctly configured, however, the tuning can take several weeks if the user is not an expert.

- Unlike what someone could expect, mesh optimization is typically highly simulation dependent.
- Spikes are present in the dynamic pressure and this may affect the dynamics of a body immersed in the wave field. However, if total pressure is considered, the effect of these spikes may become negligible.
- The optimization algorithm becomes quite unstable if many constraints are introduced. Too often simulations crashed because of optimizer failure even after having already performed hundreds of optimization cycles over the same mesh.
- Mesh optimization is ideal when a small number of nodes can be used to accurately resolve the dynamics involved. However, when larger meshes have been used, the time spent to adapt the mesh became dominant in the simulation. In this condition, a structured mesh can be less demanding under the computational time point of view. Moreover, the latter can be quickly generated for simple geometries like those involved in this work.

For this reasons, in the rest of the project a structured mesh has been used to validate the numerical wave tank. This is described in Sections 6.2, 6.3 and 6.4.

6.1.2. Interpretation of pressure and velocity profiles at the interface

In the previous section, the velocity and pressure profiles at the air-water interface have been presented. However, it is necessary to explain how these results are obtained as they may initially appear unconventional. From both Figure 6.6 and 6.8, it is possible to see that the solution computed at the interface periodically presents a constant zero value region. This result is obtained multiplying the velocity (or pressure) with the material volume fraction. The reason of this post-processing artefact is that for a fix point at the interface, i.e. a detector, the computed velocity (or pressure) will not always be the velocity of the water. In fact, for a fix point there will be a periodic alternation of water and air, see Figure 6.9(b). Thus, the velocity computed when the detector is in the air does not belong to the water particle velocity time series. This is shown in Figure 6.9(a) where the grey area represents the region of invalidity of the obtained solution and it is delimited by the Airy solution (red dashed line) and the Airy solution multiplied by the material volume fraction (green dashed line). Here, the horizontal particle velocity is shown, however, this applies also for pressure and vertical velocity with the only difference that in the case of vertical particle velocity, the region of invalidity is shifted by $\pi/2$ as the latter is not in phase with the surface elevation.



(a)

(b)

Figure 6.9: The material volume fraction at the interface level is presented in (b). Figure (a) shows the region of invalidity (grey region) of the solution computed at the interface and extracted with the use of detectors.

The main reason for doing this is that the poor results obtained when the detector is in the air are not really representative of the field. In fact, if the detector is in the air then the local dynamics are described by the bulk properties of the air as explained in Section 5.1. Thus, from now on, velocities and pressure at the interface will be always presented in this way.

6.2. Validation of the 3-dimensional NWT for linear waves propagation

In this section, validation of the numerical wave tank for the case of linear waves is performed. Three different structured meshes are used and results are compared with linear Airy wave theory.

6.2.1. Numerical setup

The setup of the numerical wave tank does not differ too much from what already presented in Section 6.1. In fact, pressure and velocity are discretized with the same mixed finite element pair P_0P_{1CV}, the same boundary conditions are applied and the same wave-maker is used. However, the generated waves are different and their relevant characteristics are presented in Table 6.2. Based on the wavelength and the water depth, the dimensions of the NWT are derived as presented in Figure 6.10.

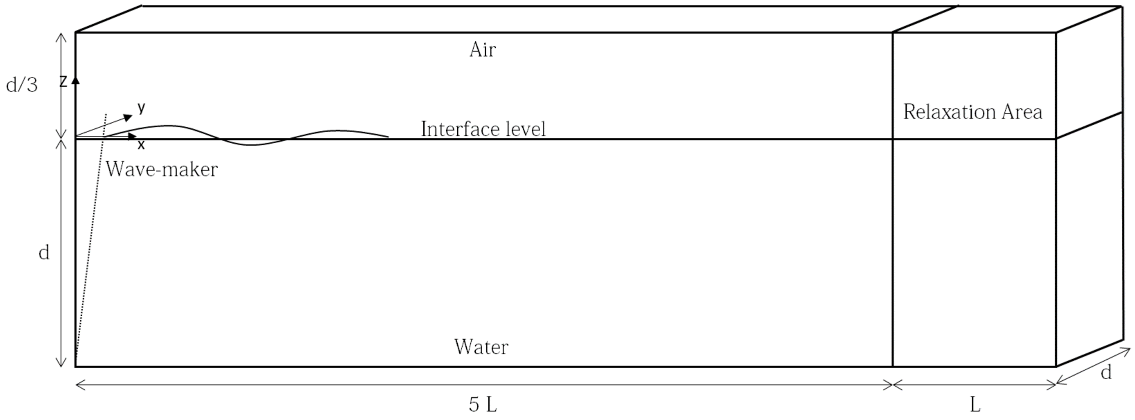


Figure 6.10: Set-up and dimensions of the numerical wave tank used in this project to simulate linear waves propagation.

f [Hz]	T [s]	H [m]	L [m]	d [m]	H/g T^2 [-]	d/g T^2 [-]
1.795	3.5	0.1	19.07	10.0	8.32e-4	0.083

Table 6.2: Wave parameters used in the validation of the NWT.

The NWT is now described with a simplex structured mesh and consequently a mesh convergence study is typically required in order to show that the results are mesh independent. In order to do so, three different meshes are used, namely coarse, intermediate and fine, characterized by a different resolution. In particular, the mesh resolutions are derived from the work presented in [2] and they are shown in Table 6.3. In the x-direction, the resolution presented in Table 6.3 is kept constant in the entire domain and linearly decreased in the absorption layer. In the z-direction, starting from the mesh size defined by the number of point per wave height, the elements dimension is linearly increased going towards the extremes of the domain. In the y-direction instead the mesh has a constant edge length of $0.5m$. In this direction, the mesh is quite coarse if compared to the mesh in the longitudinal cross-section, however, as the waves propagate in the x-direction, not much should occur in the y-direction.

Parameter	Coarse	Intermediate	Fine
P.p. wavelength	80	100	120
P.p. wave height	10	20	30

Table 6.3: Mesh characteristics based on wavelength and wave height.

Figure 6.11 shows the mesh used to discretized the domain. Here, the absorption layer can be identified on the right, the coarsest zone, while the interface region is recognizable in the zoom as the refined area. An adaptive timestep with a maximum Courant number of 0.9 is used in this setup. The number of nonlinear iterations performed in the time domain is set equal to 4 in order to better describe the air-water interface.

A FE based scheme which includes a Sweby slope limiter [30] is used to limit advective fluxes and the Bassi-Rebay scheme is adopted for the diffusion term.

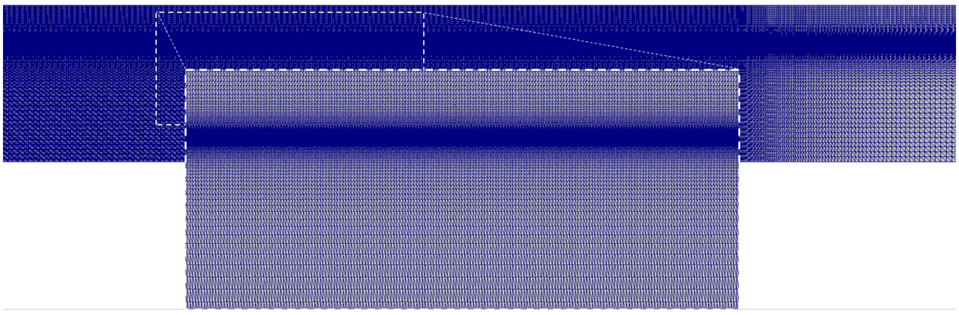


Figure 6.11: Example of the mesh used to discretized the NWT.

6.2.2. Spurious velocity modes

Initially the HyperC approach was used to limit advective fluxes as for the setups of the mesh adaptivity analysis, Section 6.1. However, with this configuration, spurious velocity modes at the interface arose. Figure 6.13(a) shows a cross-section of the domain at quote $z = 0$, i.e. interface level, from which it can be identified an instability in the velocity field. At first, because of its nature, the instability has been identified as a checker board modes (or LBB instability) problem. Checker board instability is characterised by spurious pressure modes and it occurs for finite volume spatial discretizations combined with the use of a structured mesh. In fact, spurious pressure modes may appear when velocity and pressure are co-located at the same node, i.e. when the same mesh is used for velocity and pressure. When this happens, pressure and velocity can became uncoupled and this results in a pressure field which can rapidly vary from cell to cell, as shown in Figure 6.13(a). However, in this work, velocity and pressure use two different meshes as they are discretized with the P_0P_{1CV} pair. Moreover, even if a mixed finite element/control volume discretization is used, the resulting mesh has a staggered arrangement of the unknowns, as shown in Figure 6.12, and, according to [22], the velocity-pressure pair is LLB stabilized.

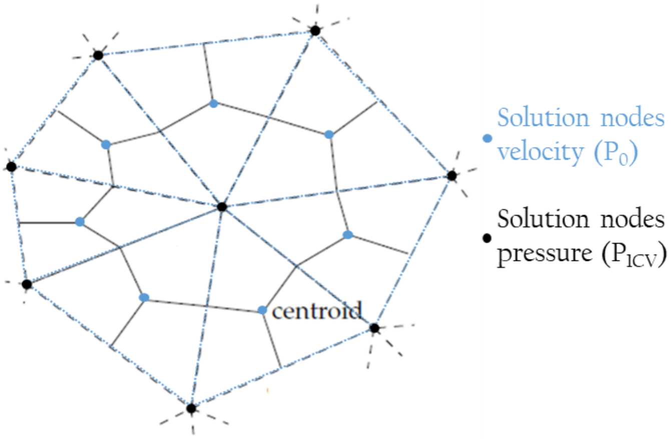
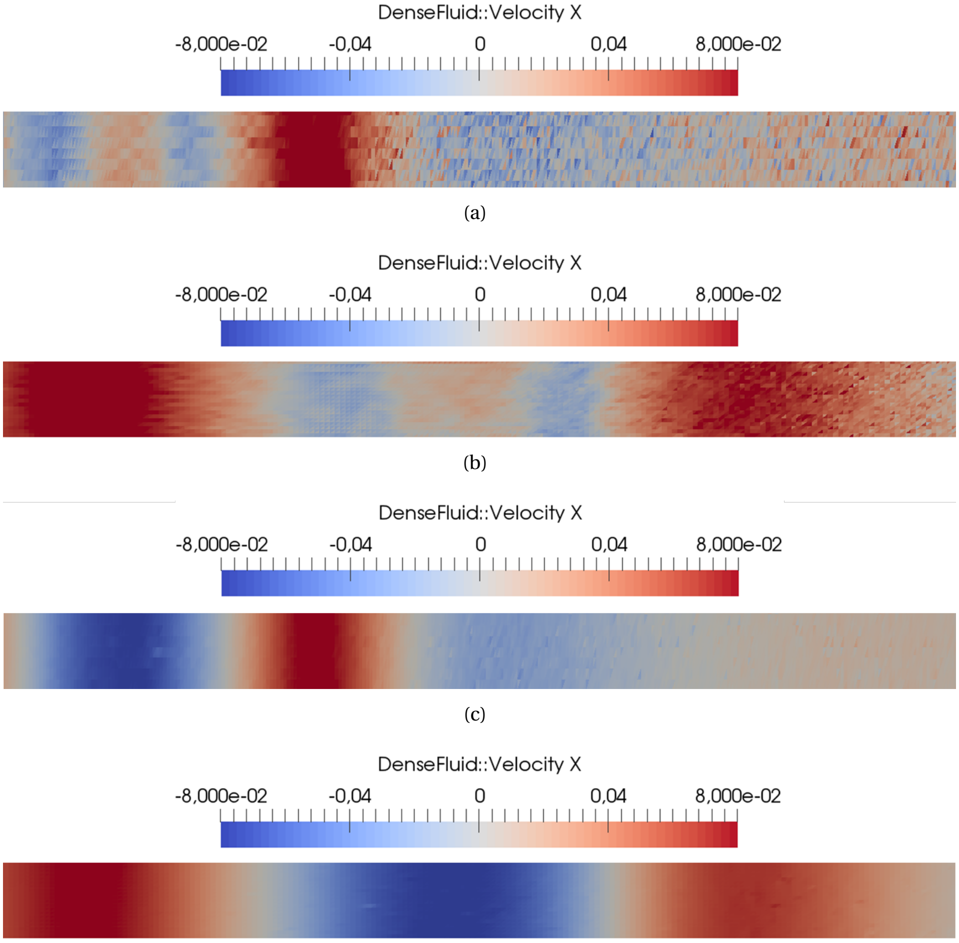

Solution nodes velocity (P_0)
Solution nodes pressure (P_{1CV})
centroid

Figure 6.12: Staggered arrangement of the unknowns obtained for pressure and velocity in the case of P_0P_{1CV}.

Also, it should be considered the fact that the spurious modes were present in the velocity instead of in the pressure as it should be for LBB instability. Thus, check board modes have been excluded as possible cause of instability. Finally, the cause of the spurious velocity modes has been identified in the choice of the edge length of the mesh in the y-direction. In fact, as found and described in [5], for large mesh Reynolds numbers, i.e. too coarse mesh, even stabilized Galerkin schemes may show spurious velocity modes. In order to validate this hypothesis, a new finer mesh has been generated. This was characterised by the same dimension as those described in Section 6.2.1 with the only difference that the edge length in the y-direction was equal to $0.1m$, i.e. five times smaller than before. Figure 6.13 presents the velocity field, at the interface (a,b) and at $z = -0.15m$ (c,d), obtained with the coarse (a,c) and fine (b,d) mesh. Two observation can be done here:

- The instability is reduced when the mesh is refined and this seems to confirm the hypothesis that the instability is caused by large mesh Reynold numbers at the interface.
- The instability almost disappear when moving below the interface. This also support the thesis as the particle velocity decreases with depth, see Section 3.2, and so the mesh Reynolds numbers do.



(a)

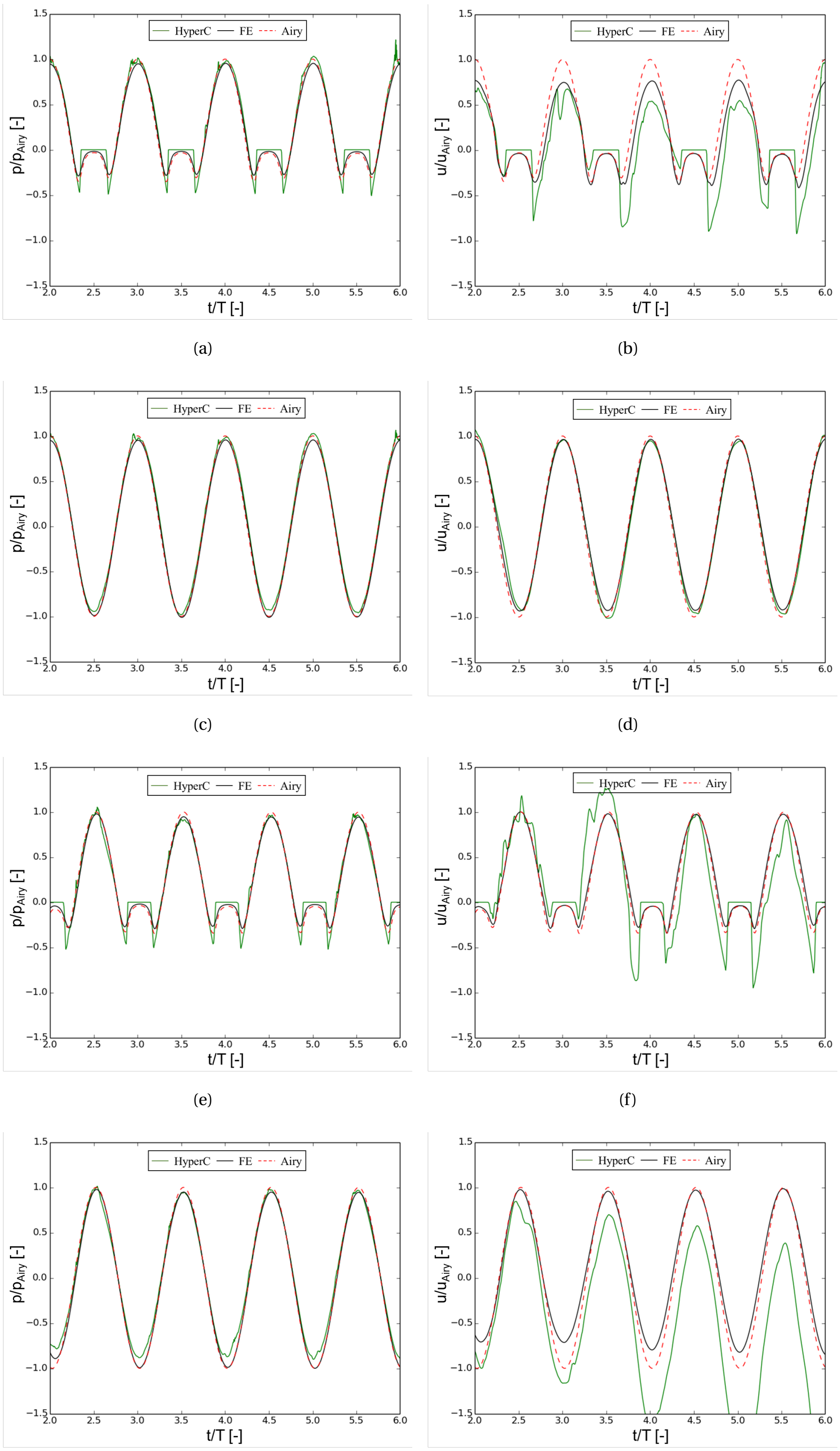
(b)

(c)

(d)

Figure 6.13: Figures (a) and (c) present the velocity field at $z = 0m$ and $z = -0.15$ respectively obtained with the coarse mesh. Figures (b) and (d) present the velocity field for the same water depths of (a) and (c) respectively but obtained with the fine mesh.

Even if the cause of this instability was identified, due to the limited available computational power, the mesh refinement in the y-direction has been kept as described in Section 6.2.1, i.e. edge length equal to $0.5m$. Another approach has been instead adopted in order to solve this issue and the control volume based HyperC approach has been substituted by a finite element based scheme for advective fluxes which includes a Sweby slope limiter. The use of a slope limiter did not completely solved the instability but it mitigated its effect. Figure 6.14 shows the improvement obtained with the use of a slope limiter. From these figures two main conclusions can be drawn: (1) the effect of the slope limiter over the pressure is almost negligible, (2) the improvement introduced by the different approach is greater at the interface and it becomes even more noticeable far from the inlet. The better results obtained for pressure and velocity, however, come at a cost. In fact, the use of a slope limiter smooths the material volume fraction field. This can be seen, for instance, from Figure 6.14(a) where the black line shows a smoother approach to the zero with respect to the green line when multiplied by the material volume fraction.



(a)

(b)

(c)

(d)

(e)

(f)

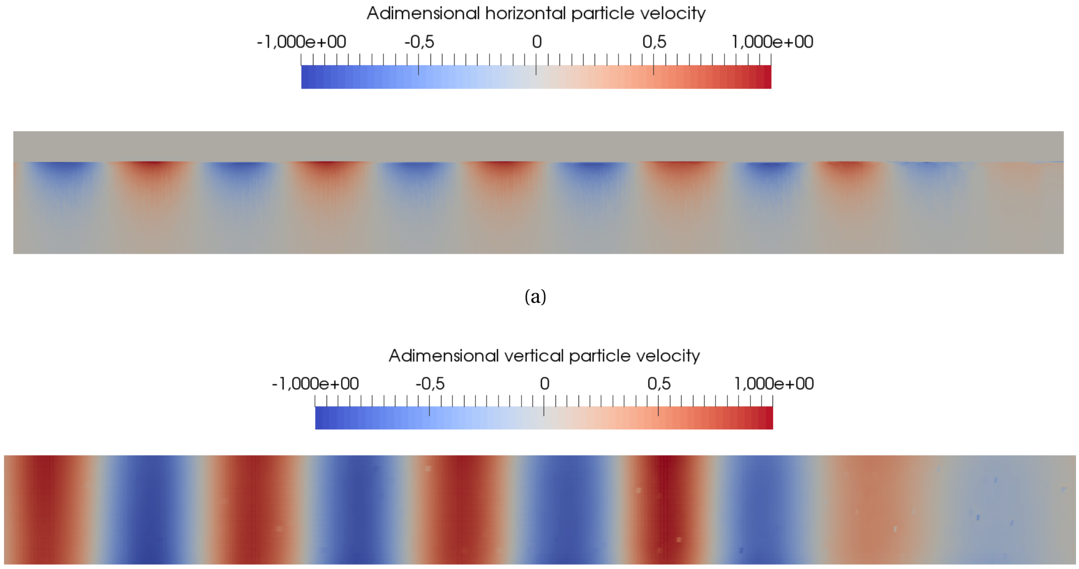
(g)

(h)

Figure 6.14: Figures (a,b,e,f) and (c,d,g,h) present the dynamic pressure and the horizontal particle velocity computed at $z = -0.025m$ and $z = -0.15$ respectively. Figures (a,b,c,d) are relative to what is extrapolated from the field at the inlet while figures (e,f,g,h) present results from a detector placed at $x = 10m$ from the wave-maker. Results obtained with the HyperC (green) and the FE (black) approach are compared with linear Airy (dashed red).

6.2.3. Results

The same setup has been used for the three different meshes presented above and good agreement with the linear Airy analytical solution has been found. Results obtained from the different meshes are presented in Figures 6.16 and 6.17 where the horizontal particle velocity and the dynamic pressure are presented for different water depths and distances from the inlet. Figure 6.15 shows the development of the waves in the numerical wave tank by means of adimensional horizontal and vertical particle velocities. Both velocity and pressure have been normalized with respect to the maximum velocity and pressure predicted by the linear wave theory. Moreover, it can be noticed the effect of the absorption layer at the end of the domain where the velocity reduces almost to zero.



(a)

(b)

Figure 6.15: Two snapshots of the domain are here presented. Figure (a) shows the adimensional horizontal velocity while (b) presents the adimensional vertical velocity at $z = -0.5$. Both velocity components have been normalized with respect to the maximum of particle velocity components predicted by the linear wave theory.

Here, some general trends can be identified:

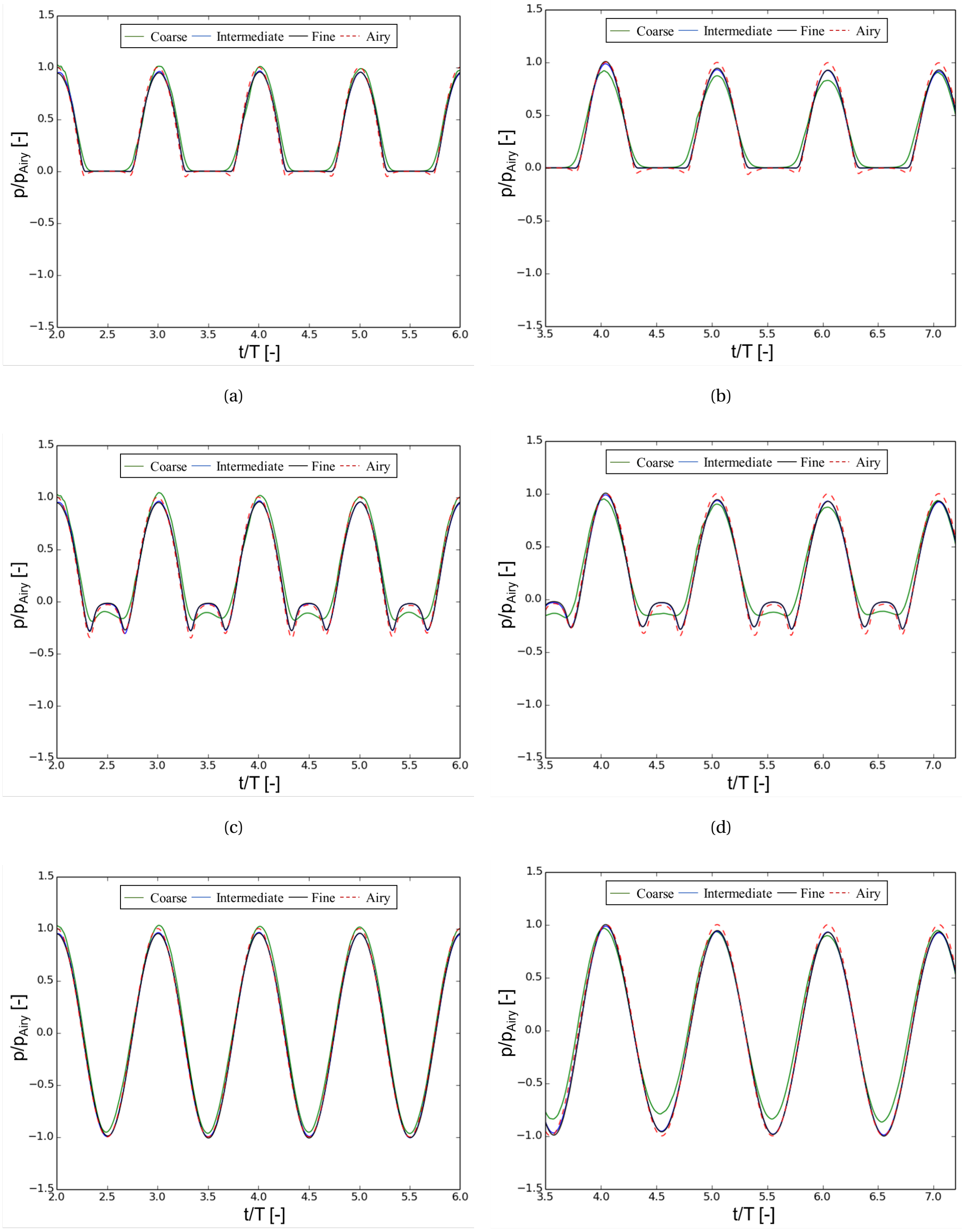
- The dynamic pressure is very well described by all the three meshes even at the air-water interface. Also, results obtained for the pressure with the intermediate and fine mesh are almost identical (Figure 6.16).

- With respect to the particle velocity, fine and intermediate meshes match the analytical solution. The coarse mesh is still capable of correctly describing the field in depth, however, at the interface it significantly loses accuracy with respect to the other two meshes (Figure 6.17(a,b)). This result underlines how the correct representation of the air-water interface is the most critical part of multi-material simulation. Also, it shows that the interface is the main source of numerical diffusion.

- At the wave-maker (Figure 6.17(a)) the velocity suffers from some numerical diffusion which seems not to be present at one wavelength from the inlet (Figure 6.17(b)). Therefore, the wave amplitude at the wave-maker is smaller than what is predicted by linear Airy theory while this is not the case at $x = 20m$. Because at $x = 5m$ the velocity matches with the findings at $x = 20m$, this problem at the inlet has to be caused by some numerical error.

- With respect to the findings of [2], the NWT developed in this work seems to be relatively less computationally demanding. In fact, in [2] similar agreement with linear theory could only be found with a mesh having 40 points per wave height. This result is attributed to the use of a slope limiter for the advective fluxes.

- Under the computational point of view, with the finest mesh and using 24 cores, the ratio elapsed time over walltime was equal to $6.6e-05$, being actually lower than what found in most of the simulations performed with the adaptive remeshing.



(a)

(b)

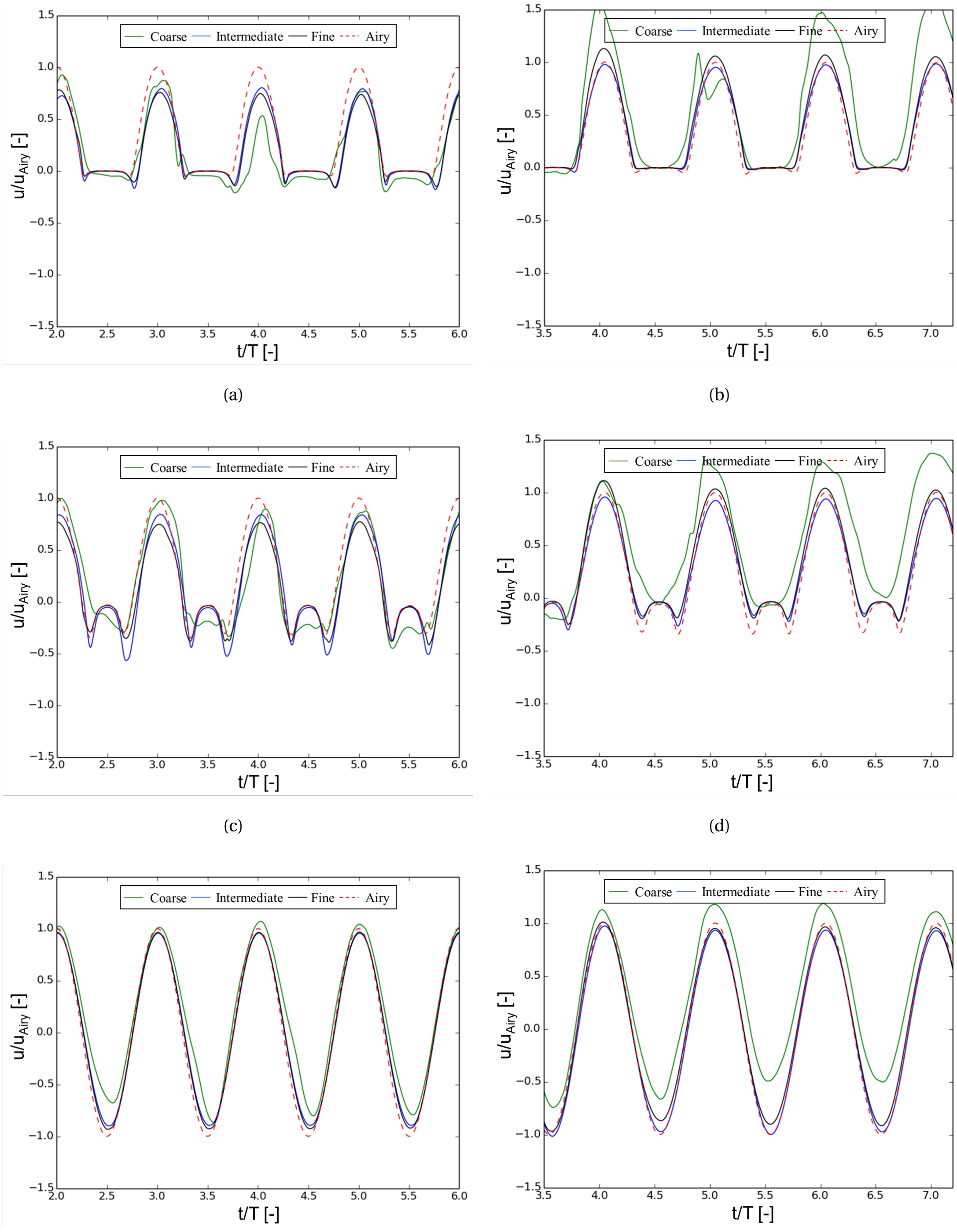
(c)

(d)

(e)

(f)

Figure 6.16: Figures (a), (c) and (e) present the dynamic pressure computed at $0m$ from the wave-maker and at $z = 0m$, $z = -0.025m$ and $z = -0.15$ respectively. Figures (b), (d) and (f) present the pressure profile at $x = 20m$ from the inlet for the same water depths of (a), (c) and (e) respectively. Results obtained with the coarse (green), intermediate (blue) and fine (black) mesh are compared with linear Airy (dashed red).



(a)

(b)

(c)

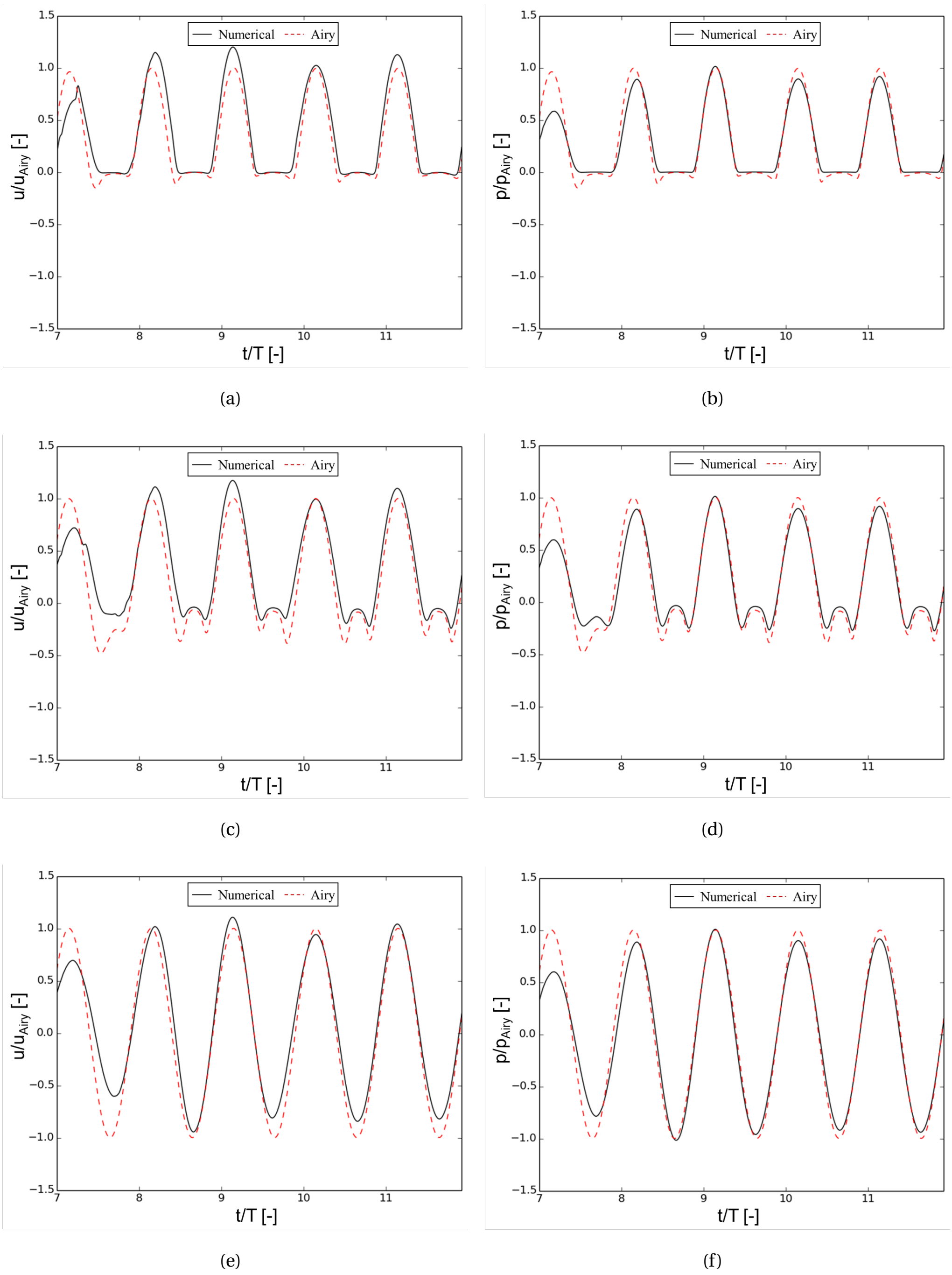
(d)

(e)

(f)

Figure 6.17: Figures (a), (c) and (e) present the particle velocity computed at $0m$ from the wave-maker and at $z = 0m$, $z = -0.025m$ and $z = -0.15$ respectively. Figures (b), (d) and (f) present the velocity profile at $x = 20m$ from the inlet for the same water depths of (a), (c) and (e) respectively. Results obtained with the coarse (green), intermediate (blue) and fine (black) mesh are compared with linear Airy (dashed red).

In conclusion, results from the fine mesh are presented at a distance from the wave-maker of around 2 and 3 wavelengths in order to show that no phase shift occurs even at large distance from the wave-maker. Thus, Figure 6.18 presents velocity (a,c,e) and pressure (b,d,f) profiles at a distance of $60m$ from the wave-maker. Once again, despite the presence of some overshoots and undershoots caused by a secondary low frequency component in the signal, good agreement is found with respect to the analytical prediction provided by the linear wave theory. Due to limited available computational power, the absorption layer implemented in these setups is relatively small when compared to literature and this may explain the presence of a secondary frequency in the velocity ad pressure profile. In fact, if the absorption layer is too short, waves may escape it and, consequently, pollute the results. However, a Fourier analysis should be performed to ensure the validity of this hypothesis.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 6.18: Figures (a), (c) and (e) present the particle velocity computed at $60m$ from the wave-maker and at $z = 0m$, $z = -0.025m$ and $z = -0.15$ respectively. Figures (b), (d) and (f) present the dynamic pressure profile at $x = 60m$ from the inlet for the same water depths of (a), (c) and (e) respectively. The results (black) are compared with linear Airy (dashed red).

6.3. Further validation of the 3-dimensional NWT

In this section, validation of the numerical wave tank for the case of nonlinear waves propagation is performed. The setup is the same as that presented for the case of linear waves and only a fine mesh is used. Convergence study hasn't been performed for the case of nonlinear waves due to time restrictions of the project.

6.3.1. Numerical setup

Apart from the waves characteristics, the setup of the numerical wave tank is the same as for the previous one. Here, mesh convergence study is not performed and thus, based on the result of Section 6.2, the fine mesh resolution is chosen. Consequently, the mesh adopted is a simplex structured mesh with 30 points per wave height and 120 points per wavelength. The dimensions of the domain are based on the wave characteristics presented in Table 6.4 and they are depicted in Figure 6.19. The main modification implemented in the domain is that the length of the absorption layer is here increased by a wavelength. This is done in order to avoid reflection from the back of the domain based on the results obtained from the linear wave tank.

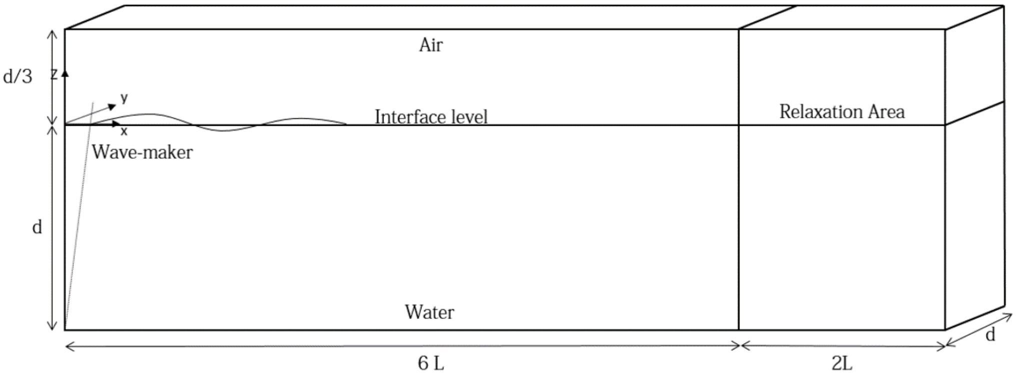


Figure 6.19: Set-up and dimensions of the numerical wave tank used in this project to simulate nonlinear waves propagation.

f [Hz]	T [s]	H [m]	L [m]	d [m]	H/g T^2 [-]	d/g T^2 [-]
5.236	1.2	0.04	2.22	1.1	2.83e-3	0.078

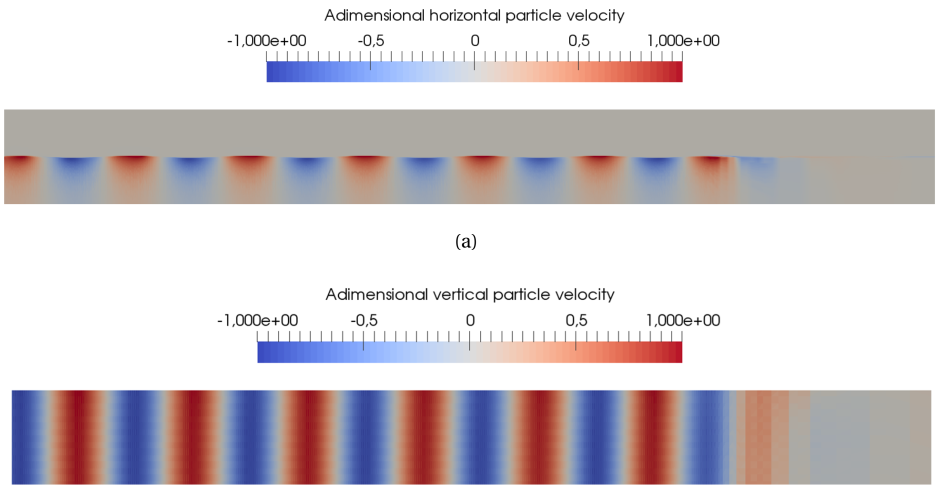
Table 6.4: Wave parameters used in the validation of the NWT.

Waves are generated with the same wave-maker used in the other wave tanks, however, a second-order dispersion relation is now used to compute the wavenumber k. In fact, from the adimensional wave height and water depth, it is possible to see that these waves fall in the region of applicability of the second-order Stokes wave theory (See Section 3.2). Nevertheless, waves are compared with linear Airy wave theory as they still satisfy the condition of small amplitude waves, $H/L \leq 0.03$, introduced by [31]. Moreover, as discussed in Section 6.1, for second-order Stokes waves in deep water condition, the non linearities lose of importance and linear wave theory can be used to predict particle velocity and dynamic pressure [7].

6.3.2. Results

Results obtained for the case of nonlinear waves propagation are here presented and discussed. Firstly, Figure 6.20 shows the waves propagation in the domain by means of adimensional horizontal (a) and vertical (b) particle velocities. With respect to what has been found for the linear waves simulations, here the absorption layer seems to be more effective. In fact, from both Figure 6.20(a) and 6.20(b), it is possible to see that at the end of the domain the velocity decreases to zero. The effect of this can be appreciated also in Figure 6.21 where the secondary low frequency observed in the case of linear waves is now not present. From these figures, it is also possible to see that, in general, good agreement is found for particle velocity and dynamic pressure when compared to the analytical solution provided by the linear wave theory. Here, waves are present at $x = 5m$, about 2.25 wavelengths, and at $x = 10$, about 4.5 wavelengths. From Figures 6.21(e,f,g,h), it is also possible to observe the development of the wave front. In fact, taking into account the initial ramp of $3s$, the first fully developed wave should reach $x = 10m$ at $t \approx 12T$ which corresponds to what is shown in the plot. As for the case of linear waves, no phase shift is detected even far from the inlet. Some conclusions can be drawn from this results:

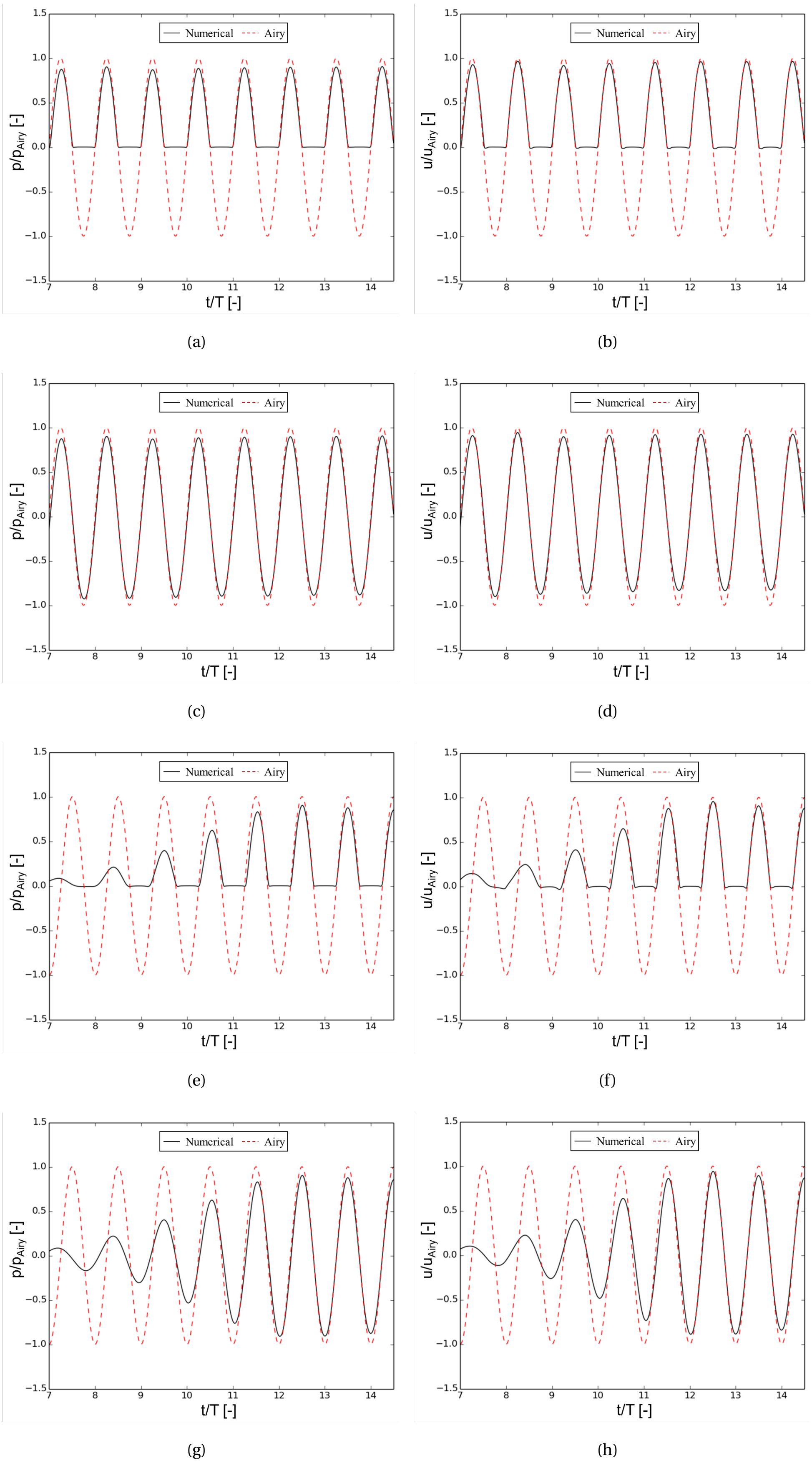
- The setup developed for the numerical wave tank is also capable of correctly describe nonlinear waves propagation if non-linearities are of small relevance.
- An absorption layer of length equal to 2 wavelength seems to be enough to damp the waves out and avoid reflection from the back of the domain. This is a very good result if compared with literature. In fact, most of the studies present a relaxation area which typically occupies half of the entire domain.
- In terms of both velocity and pressure, an almost constant error is present at every crest and trough showing that the generated waves are slightly smaller than expected. In general, this means that some numerical diffusion is still present at the air-water interface. However, the same simulation should be performed with a higher resolution mesh in order to ensure that the error is caused by numerical diffusion. In fact, because the error seems to be constant, it may be that it is actually introduced at the level of the wave-maker, in the wave generation.



(a)

(b)

Figure 6.20: Two snapshots of the domain are here presented. Figure (a) shows the adimensional horizontal velocity while (b) presents the adimensional vertical velocity at $z = -0.5$. Both velocity components have been normalized with respect to the maximum of the particle velocity components predicted by the linear wave theory.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 6.21: Figures (a,b,c,d) present the dynamic pressure (a,c) and the horizontal particle velocity (b,d) computed at distance $x = 5m$ from the wave-maker and at $z = 0m$ (a,b) and $z = -0.025$ (c,d). Figures (e,f,g,h) present the dynamic pressure and the velocity profile at distance $x = 10m$ from the inlet for the same water depths of (a,b) and (c,d) respectively. Results obtained with the CFD simulation (black line) are compared with the analytical solution provided by linear Airy (dashed-red line).

6.4. Free heave decay test

In this section, firstly the setup of the free decay test performed for heave motion is presented. Secondly, the fluid structure algorithm used to compute the motion of the floater is briefly described. Finally, results are compared with the work of [11]. In [11], a heave decay test is performed with the use of the CFD solver OpenFOAM 16.06+ and results are validated with experimental data from [25].

6.4.1. Numerical setup

As for the numerical wave tanks presented in Sections 6.2 and 6.3, the free decay test is simulate in a 3-dimensional domain. The dimensions on the domain are depicted in Figure 6.22, while Table 6.5 present the main characteristics of the floater. The water depth, d, is set to $0.9m$ in Figure 6.22. In order to be able to compare the results, all the dimensions are taken from [11].

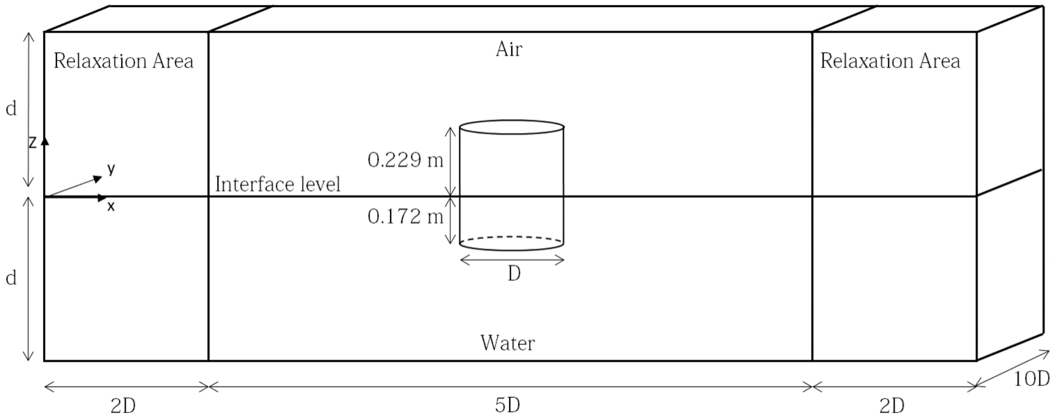


Figure 6.22: Set-up and dimensions of the numerical wave tank used in this project to simulate a heave decay test.

D [m]	H [m]	Draft [m]	Mass [kg]	$I_{yy} = I_{xx}$ [kg·m^2]
0.515	0.401	0.172	35.85	0.9

Table 6.5: Characteristics of the cylindrical floater.

Apart from the absence of the wave-maker, the boundary conditions are the same as those applied in the other NWTs. At the inlet, a no-normal flow condition is now applied. The absorption layer is here modified in order to have a relaxation zone on both sides of the domain. The advantage of this modification lies on the possibility to use a smaller domain. In order to obtained the double absorption layer showed in Figure 6.22, the absorption coefficient, Equation 6.1, is modified as follow

$$\sigma = \begin{cases} \begin{cases} \frac{1}{4}\left(\tanh\left[\frac{\sin(\pi(4\tilde{x}_1-1))/2}{1-(4\tilde{x}_1-1)^2}\right]+1\right) & \text{if } 0 \le \tilde{x}_1 \le \frac{1}{2} \\ \frac{1}{4}\left(\tanh\left[\frac{\sin(\pi(3-4\tilde{x}_1))/2}{1-(3-4\tilde{x}_1)^2}\right]+1\right) & \text{if } \frac{1}{2} \le \tilde{x}_1 \le 1 \end{cases} & \text{if } x < (l_{0_1} + l_1) \\ \\ \begin{cases} \frac{1}{4}\left(\tanh\left[\frac{\sin(\pi(4\tilde{x}_2-1))/2}{1-(4\tilde{x}_2-1)^2}\right]+1\right) & \text{if } 0 \le \tilde{x}_2 \le \frac{1}{2} \\ \frac{1}{4}\left(\tanh\left[\frac{\sin(\pi(3-4\tilde{x}_2))/2}{1-(3-4\tilde{x}_2)^2}\right]+1\right) & \text{if } \frac{1}{2} \le \tilde{x}_2 \le 1 \end{cases} & \text{otherwise} \end{cases} \tag{6.3}$$

where $\tilde{x}_1 = (l_{0_1} - x)/l_1$ and $l_{0_1} = 0$ and $l_1 = 1L$ are the starting point and the length of the first sponge layer respectively. In the lower part of Equation 6.3, $\tilde{x}_2 = (x - l_{0_2})/l_2$ and $l_{0_2} = 7D$ and $l_1 = 2D$ are the starting point and the length of the second sponge layer respectively. Numerical discretizations and schemes are the same used for the two NWTs. Three different meshes are used in order to ensure that the results are mesh independent. Based on the work of [11], three resolutions are used and they are presented in Table 6.6.

Parameter	Coarse	Intermediate	Fine
P.p. diameter	10	15	20

Table 6.6: Mesh characteristics based on the diameter of the floater.

As for the NWTs, a simplex structured mesh is used to discretize the domain. Starting from the resolution defined in Table 6.6, the elements size is linearly increased going from the body to the extremes of the domain in both x- and y-direction. In the z-direction, the domain is refined at the interface in order to be able to capture the deformation of the air-water interface caused by the motion of the cylinder. Figure 6.23 presents the side view (a) and the top view (b) of the mesh for the case of 20 points per diameter ($p.p.d.$) of resolution.

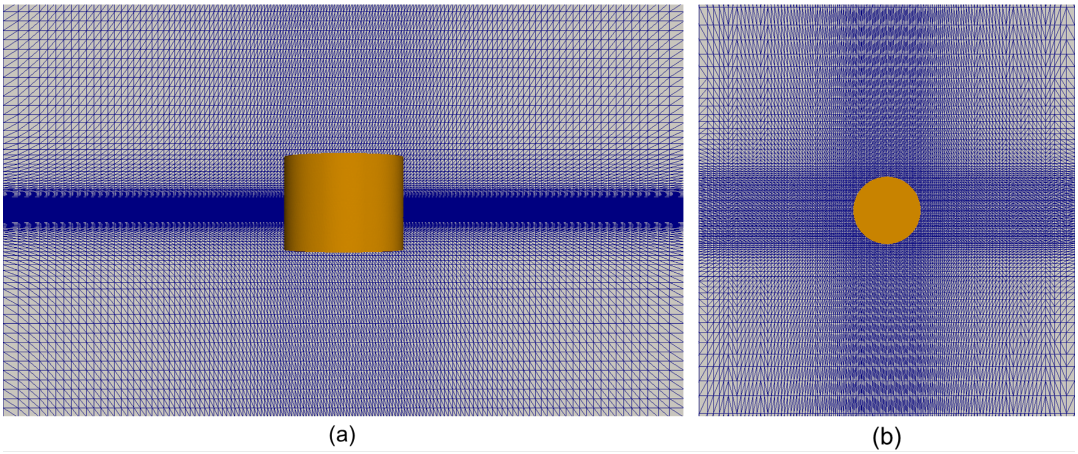


Figure 6.23: Two snapshots of the mesh are here presented. Figure (a) shows the longitudinal cross-section of the mesh while (b) presents a cross-section of the mesh at quote $z = 0m$. The mesh is relative to the case of $20p.p.d.$ but it is representative of both meshes as they differ only in the resolution.

Independently on the base mesh, the resolution of the unstructured mesh that defines the cylinder is $20p.p.d.$ in every direction. The floater is modelled with the immersed-body method described in Section 5.3. Figure 6.24 shows the solid concentration field that defines the body in the numerical domain. In the picture, the cross-section of the actual unstructured solid mesh is projected on top of this field in order to visually show the accuracy of the method. Despite the non-perfect representation, the relative error in the volume of the solid resulted to be less 0.13% for the entire duration of the simulation. The volume of the solid represented with the immersed body method can be computed at every timestep as the integral of the solid concentration field in the entire domain.

6.4.2. The FSI algorithm

The fluid structure interaction (FSI) algorithm used in this project has been acquired from the work of [2]. This is based on a staggered approach since the equations that describe the dynamics of fluid and solid are computed at asynchronous instants and solved with different methods [2]. The FSI algorithm developed in [2] can be divided in three main steps:

1. *Fluidity* solves the fluid equations and computes pressure and viscous forces and moments acting on the structure by means of integration of pressure and velocity fields on the wetted surface [2].
2. The computed forces are transferred to the RBC and the equations of motion are numerically solved. Then, the python code provides the CFD solver with the position of the CoG and both linear and angular displacement and velocity [2].
3. The computed velocities and displacements are applied to the structural mesh which is accordingly moved in the fluid domain. The new solid concentration field, the velocity of the body and the penalty term are then projected from the structural to the fluid mesh by means of Galerkin projection [2].

With respect to what is presented in [2], two main modifications have been implemented in the coupling between fluidity and the rigid body code:

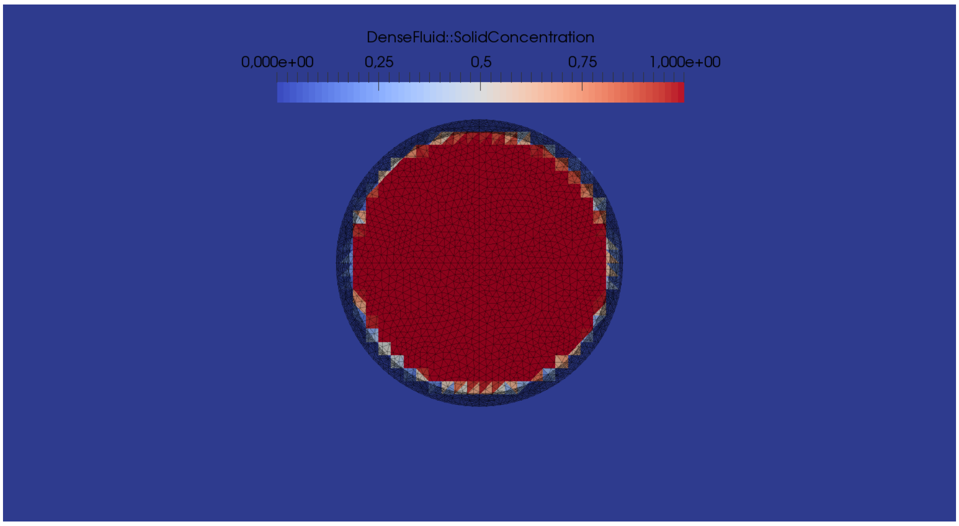


Figure 6.24: The solid concentration field describing the floater in the domain is compared with the actual unstructured solid mesh.

- The algorithm described in [2] is based on the old python-based RBC, thus, the new algorithm has been modified to make it compatible with the 3-dimensional RBC (see Section 4). This also included a modification on the *fluidity* side of the coupling. In fact, the internal algorithm has been modified in order to be able to deal with 6DoF systems.
- In [2], the forces acting on the structure were computed with a python script while now they are directly computed in *fluidity* by making use of the capability of the solver to integrate fields over the boundaries of the domain.

A disadvantage of the immersed-body method, with respect to the mesh-conforming defined-body method, lies on the fact that it is not possible to directly integrate a field over the surface of the body as CFD solvers are typically able to integrate fields only over boundaries. Thus, to overcome this problem, velocity and pressure fields are firstly projected from the fluid mesh to structural mesh via Galerkin projection. In fact, on the structural mesh, the surfaces that delimit the body are recognised as boundaries from the solver and both pressure and viscous force can be computed.

6.4.3. Results

For the free heave decay test, as described in both [11] and [25], the CoG of the floater is initially displaced $0.075m$ from its equilibrium position. The equilibrium position is obtained as static equilibrium between the buoyancy force, Equation 3.28, and the gravity force acting on the floating body. At the beginning of the simulation the body is released and it is left free to move until the oscillation dies. The response in heave for the three levels of resolution is presented in Figure 6.25 and it is compared with the response obtained with OpenFOAM.

Parameter	**10 p.p.d.**	**15 p.p.d.**	**20 p.p.d.**	**OpenFOAM (15 p.p.d.)**	**Experiments**
T_n $[s]$	1.145	1.137	1.127	1.114	1.112
ν $[1/s]$	0.0612	0.0617	0.0642	0.0811	n.d.
a_{zz} $[kg]$	2643.2	2608.2	2561.9	2500.1	n.d.
b_{zz} $[kg/s]$	328.33	326.30	333.77	411.75	n.d.

Table 6.7: Natural period, damping coefficient and hydrodynamic added mass and damping obtained from the free decay test are presented. The first three columns show the results from *fluidity* for the three different resolutions, the fourth columns presents the results obtained with OpenFOAM in [11] and the last column presents the period obtained via experimental free decay in [25].

The period experimentally computed equals $1.112s$ while the period obtain with OpenFOAM is $1.114s$. The natural periods computed with *fluidity*, for the three different meshes, are presented in Table 6.7 along with the obtained damping, added mass and hydrodynamic damping coefficients. The period is evaluated

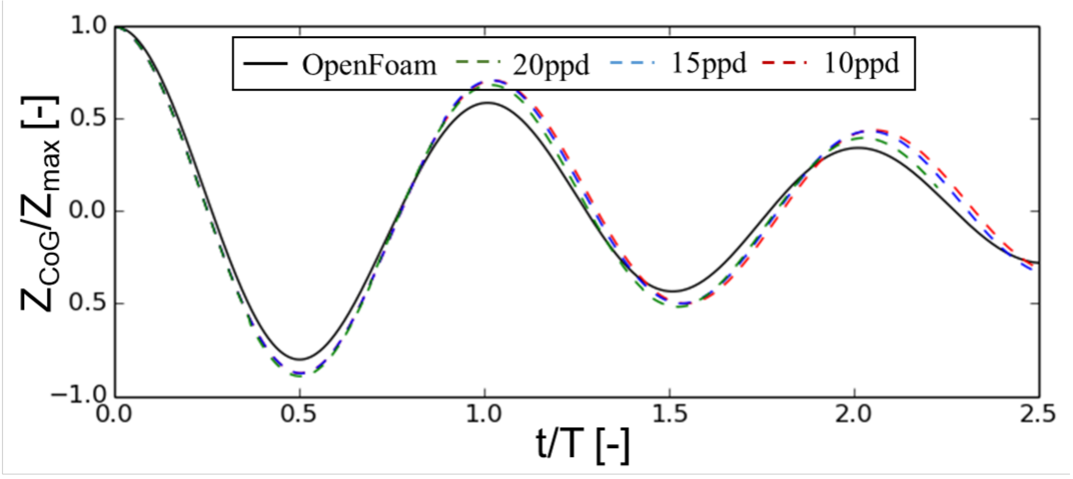


Figure 6.25: Results obtained for the free decay test by using 15$p.p.d.$ (blue-dashed line), 20$p.p.d.$ (green-dashed line) and OpenFOAM (black line).

Parameter	**20 p.p.d.**	**OpenFOAM (15 p.p.d.)**	**Error [%]**
T_n $[s]$	1.127	1.114	1.16
ν $[1/s]$	0.0642	0.0811	20.8
a_{zz} $[kg]$	2561.9	2500.1	2.47
b_{zz} $[kg/s]$	333.77	411.75	18.9

Table 6.8: Error in the evaluation of natural period, damping coefficient and hydrodynamic added mass and damping with respect to the results obtained in [11].

from the first three periods of the record. The damping coefficient is obtained from Equation 3.50 in which the adimensional damping coefficient, κ, has been obtained from the heave motion with the use of a double amplitude approach, Equation 3.55. Here, Equation 3.55 is used under the assumption that the damping can be considered to be linear. This is done because it simplifies the computation of the damping and because non-linearities in the damping are mainly relevant for roll motion [14]. The relative errors for all the computed quantities are reported in Table 6.8 for the case of the finest mesh. From the results obtained with the free decay test, some conclusion can be drawn:

- Increasing the resolution of the mesh, the error is reduced for all the computed quantities. However, while for period and added mass the error is relatively small, the error in the computation of the damping is still very high.

- A possible cause of the lower damping in *fluidity*, may be the use of the immersed-body method. In fact, because the field integration (either velocity or pressure) is performed slightly inside the body, the so-obtained forces may correspond to those acting on a smaller floater. However, it is relevant to say that the heave motion converges to zero meaning that in static (or semi-static) condition the body seen by the solver has the correct dimensions.

- Another possible cause of this phenomena, may be represented by the use of the Sweby slope limiter (see Section 6.2.2). In fact, as it is shown in Figure 6.14, the introduction of a slope limiter affects the material volume fraction field. In particular, the solution obtained with the use of the Sweby slope limiter is significantly more diffusive than the solution obtained with the HyperC approach. This phenomena is also described in detail in [3]. Moreover, as described in Section 6.2.2, also the velocity field were influenced by the slope limiter and in particular, it was smoothed. In that case, the obtained effect was beneficial as the aim was to limit the spurious velocity modes, however, this may be the cause of the low hydrodynamic damping computed from the free decay test. Nevertheless, other schemes for advective fluxes limitation should be tested in order to be able to determine whether this is the reason behind the discrepancy found between *fluidity* and OpenFOAM or not.

- With respect to the computation of the hydrodynamic damping, the results obtained within this work are in disagreement with what was found in [2]. In fact, in [2], it was found that *fluidity* was overestimating the damping with respect to OpenFOAM for the case of heave motion. However, it is important to underline that the FSI analysis performed in [2] was including also a wave field and that the HyperC approach was used to limit the advective fluxes. Thus, the results obtained in the two projects are not fully comparable.

A heave decay test for a body constraint by a mooring line has been performed as well. The properties of the mooring line implemented in this simulation are detailed in [11] and [25] and a quasi-static approach is used to model it (see Section 2.1.3). Due to lack of time the simulation hasn't been completed and, thus, it is not presented here. Nevertheless, it is still relevant to point out that, from the first period of the heave motion, no significant differences between free and constraint decay test are observed. In fact, both period and amplitude of the motion match with what is presented in Figure 6.25. Based on the physical parameters and supported by the findings of both [11] and [25], this result was expected.

7 Conclusions and recommendation

In this work, modelling and validation of a three-dimensional nonlinear numerical wave tank are presented. The starting point of this project was represented by a two-dimensional NWT developed in [2]. The model is based on the coupled use of the CFD solver *fluidity*, to resolve the fluid dynamics, and an in-house python-developed code, to numerically solve the equations of motion of a rigid body. In [2], a two-dimensional Rigid Body Code (RBC) developed by MScEng. J.D. Brandsen from Delft University of Technology was used. Thus, the first step was the development and validation of a three-dimensional version of this python code. The RBC allows the user to apply springs, dampers, point loads and constraints on the body and it is coupled with the CFD solver thanks to the python interface which is present in *fluidity*. As the second step of this work, the setup of the numerical domain containing both air and water has been tuned in order to obtain a NWT in which both linear and nonlinear waves could correctly propagate. Both an unstructured adaptive mesh and a simplex structured mesh have been used to discretize the domain and, based on the obtained results, the structured mesh has been selected for the rest of the project. Finally, a floater, modelled with the immersed-body method, has been added in the model and a free heave decay test has been performed. This last simulation had a double aim: (1) it analysed the accuracy of *fluidity* in computing the hydrodynamic coefficients of a floating body and (2) it tested the FSI algorithm developed to couple the CFD solver and the python code.

In Section 1, a research question has been formulated for each step of this project. Thus, in the next section, each of these questions will be answered along with a discussion of the results and conclusions. Finally, the recommended further research will be presented in Section 7.2.

7.1. Conclusions

The four research questions introduced in Section 1 have been addressed during the project. The first question is covered in Section 4.3, while the second, third and fourth questions are answered in Section 6.1, 6.2 and 6.4 respectively. Here, these are made explicit and treated one at the time.

- *How accurate is the solution of the equations of motion computed by the python code for a six degree of freedom rigid body subjected to external loads and constrained only by a spring and a damper?*

  In Section 4, the RBC developed in this project is presented. Even though it was the resulting extension of an already validated code, validation of the new code was necessary as the code has been largely modified during this work. In Section 4.3, the RBC is validated with analytical solutions, for the easier cases, and with the results produced by the previous RBC for more complex tests. The overall accuracy of the python code resulted to be very high as the maximum observed relative error with respect to the analytical predictions was about 1.5%. Moreover, it resulted to be more accurate than the previous version when compared with the theoretical solutions.

- *How efficient and precise is the adaptive remeshing in capturing the air-water interface? What is the cost of mesh adaptivity and what are the main issues related to its implementation?*

In Section 6.1, the mesh optimization algorithm implemented in *fluidity* is extensively tested. Mesh adaptivity resulted to be an efficient tool to capture the air-water interface, in fact, in almost every simulation that has been ran the interface was correctly represented. However, velocity and pressure were not accurately computed and, thus, waves propagation in the domain resulted to be poorly described. The tuning of all the parameters related to the mesh adaptivity resulted to be highly time consuming. Also, it is observed that the optimization algorithm becomes quite unstable if many constraints are introduced. Too often simulations crashed because of optimizer failure even after having already performed hundreds of optimization cycles over the same mesh. Under the computational point of view, it is concluded that mesh optimization is ideal when a small number of nodes can be used to accurately resolve the dynamics involved. However, when larger meshes have to be used, the time spent to adapt the mesh became dominant in the simulation and in this condition, a structured mesh can be less demanding under the computational time point of view. Moreover, the latter can be quickly generated for simple geometries like those involved in this work.

- *For the case of a three-dimensional NWT containing both air and water, discretized with $P0 - P1_{CV}$ and represented by a simplex structured mesh, how precise is the waves propagation predicted by the CFD solver when compared to linear Airy wave theory? And what is the accuracy of the solver when nonlinear waves are generated?*

  In Sections 6.2 and 6.3, the numerical wave tank is presented and results for both linear and nonlinear waves propagation are shown. The setup developed in this project resulted to be capable of correctly describing both linear and nonlinear waves propagation. With respect to both velocity and dynamic pressure profiles, good agreement with what is predicted by linear Airy theory was found. Moreover, despite the use of a coarser mesh, the obtained results are similar in accuracy to those found in [2]. This result is attributed to the use of a slope limiter as explained in Section 6.2.2. Reflection from the back of the domain was present in the case of linear waves, however, this issue has been solved by making use of a longer absorption layer as described in Section 6.3. For the case of nonlinear waves, apart from the presence of some numerical diffusion, no relevant differences have been observed in the results with respect to the linear case.

- *How accurate are the hydrodynamic damping and added mass computed by the solver performing a heave free decay test? To which extend is the accuracy of the solver affected by the presence of a mooring line?*

  In Section 6.4, the setup and results of the free heave decay test are presented. Here, after an initial displacement, the body is let free to move until the oscillation dies. From the record of the heave motion, natural period and hydrodynamic coefficients of the floater are estimated and compared to the finding of [11]. In terms of natural period, the observed relative error was 1.16% for the finest mesh, however, the error in the estimation of the damping was about 20%. The large error in the computation of the hydrodynamic damping is probably caused by the presence of a slope limiter in the scheme used to limit the advective fluxes. In fact, as shown in Section 6.2.2, the use of a slope limiter affects both material volume fraction and velocity fields. In particular, the solution obtained with the use of the Sweby slope limiter was significantly more diffusive than the solution obtained with the HyperC approach and this may be the cause of the low hydrodynamic damping computed from the free decay test. Nevertheless, further investigation is needed to ensure the validity of this hypothesis. Finally, a mooring line has been introduced in the system, however, due to lack of time it was not possible to fully investigate the effect of this on the resulting natural period and hydrodynamic coefficients. Still, it is relevant to point out that, from the first period of the heave motion, no significant differences between free and constraint decay test have been observed.

7.2. Further Research

Although the numerical wave tank developed in this project has shown to be able to correctly describe both linear and nonlinear waves, it did not perform very well in the FSI analysis. Moreover, the capabilities of the

RBC haven't been fully exploited in this project. Therefore, several areas that demand improvement and/or investigation have been identified. These are proposed here as possible future research topics.

- The results obtained with the use of adaptive remeshing for the case of higher waves showed that it is possible to describe a 3-dimensional NWT with very few nodes. In fact, even though the obtained velocity and pressure profiles were still inaccurate, the nodes used to discretize the domain were less than half of those used in the finest structured mesh. Thus, further research should be done in order to find a way to accurately describe the waves propagation in a NWT with the use of mesh adaptivity.
- FSI analysis should be further investigated. Firstly, different schemes for advective fluxes should be tested in order to identify the exact cause of the low estimated damping found in Section 6.4. Secondly, roll decay test should be performed to verify the accuracy of the CFD solver for a different type of motion. Finally, the capabilities of the solver in computing the motions of a floating cylinder in a wave field should be tested.
- The development of a directional absorption layer is necessary. In fact, the actual absorption layer is modelled to dump any wave that is inside it. However, if the absorption layer would be modelled in such a way that the directionality of the wave is included, then it could be possible to place a relaxation area also on the wave-maker side. This would drastically reduce the space needed in between the inlet and the floater when waves are generated as the waves reflected by the body would be absorbed.
- As mentioned in Section 4, the three-dimensional rigid body code is capable of dealing with any body geometry and thus, it would be interesting to perform the same FSI analysis with different types of floater.

Bibliography

[1] Arapogianni Athanasia, Genachte Anne-Bénédicte, and Moccia Jacopo. The offshore wind market deployment: Forecasts for 2020, 2030 and impacts on the european supply chain development. *Energy Procedia*, 24:2 – 10, 2012.

[2] M. Baudino Bessone. Wave-structure interaction study in context of floating wind turbines by mean of coupled rigid body code and fluidity cfd software, 2017.

[3] Wilson Cian. *Modelling multiple-material flows on adaptive unstructured meshes.* Phd dissertation, 2009.

[4] P.E. Farrell, M.D. Piggott, C.C. Pain, G.J. Gorman, and C.R. Wilson. Conservative interpolation between unstructured meshes via supermesh construction. *Computer Methods in Applied Mechanics and Engineering*, (198), October 2009.

[5] Tobias Gelhard, Gert Lube, Maxim Olshanskii, and Jan-Hendrik Starcke. Stabilized finite element schemes with lbb-stable elements for incompressible flows. *Elsevier*, (117), 2005.

[6] P.M. Gresho and S.T. Chan. Solving the incompressible navier–stokes equations using consistent mass and a pressure poisson equation.

[7] Clauss Gunther, Lehmann Eike, and Ostergaard Carsten. *Offshore Structures Volume I. Conceptual Design and Hydromechanics.* Springer, Berlin, 1992.

[8] Global Wind Energy Council (GWEC). Global statistics, 2016. URL `http://gwec.net/global-figures/graphs/`.

[9] H.R. Hiester, M.D. Piggott, P.E. Farrell, and P.A. Allison. Assessment of spurious mixing in adaptive mesh simulations of the two-dimensional lock-exchange. *Elsevier*, (73), October 2013.

[10] C. W. Hirt and B. D. Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of computational physics.*

[11] Rivera Arreba I. Computation of nonlinear wave loads on floating structures, 2017.

[12] Cottrell J. Austin, Hughes Thomas J.R., and Bazilevs Yuri. *Isogeometric Analysis. Towards Integration of CAD and FEA.* John Wiley and Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom, 2009.

[13] Vugts J.H. The hydrodynamic coefficients for swaying, heaving and rolling cylinders on a free surface, 1968.

[14] Journée J.M.J. and Massie W.W. *Offshore Hydromechanics.* Delft University of Technology, Delft, 2001.

[15] Cruz Joao and Atcheson Mairead. *Floating Offshore Wind Energy: The Next Generation of Wind Energy.* Springer, Gewerbestrasse 11, Cham, Switzerland, 2016.

[16] Padmavathi Lakshmanan, Jun Liang, and Nicholas Jenkins. Assessment of collection systems for hvdc connected offshore wind farms. *Electric Power Systems Research*, 129:75 – 82, 2015.

[17] W.E Leithead. Wind energy, 2007. URL `http://rsta.royalsocietypublishing.org/content/365/1853/957.figures-only`.

[18] Piegl Les and Tiller Wayne. *The NURBS Book.* Springer, Berlin, 1995.

[19] Jian Luo. *Conservative sharp interface methods for incompressible multi-phase flow.* Phd dissertation, 2016.

[20] Sander Mertens, Gijs van Kuik, and Gerard van Bussel. Performance of an h-darrieus in the skewed flow on a roof. 125:433–440, 11 2003.

[21] Borg Michael, Shires Andrew, and Collu Maurizio. Offshore floating vertical axis wind turbines, dynamics modelling state of the art. 08 2014.

[22] Applied Modelling and Computation Group. fluidity manual, 2015.

[23] Bokhove Onno and van der Vegt Jaap J.W. Introduction to (dis)continuous galerkin finite element methods, 2008.

[24] International Hydrographic Organization. Iho dcdb world bathymetry database, 2014. URL `https://maps.ngdc.noaa.gov/viewers/bathymetry/`.

[25] Johannes Palm, Claes Eskilsson, Guilherme Moura Paredes, and Lars Bergdahl. Coupled mooring analysis for floating wave energy converters using cfd: Formulation and validation. *Elsevier*, (16), 2016.

[26] Porter R. Calculation of the hydrodynamic coefficients for cylinders of rectangular cross-section oscillating in the free surface. Technical report, School of Mathematics, University of Bristol, 2008.

[27] Butterfield S., Musial W., Jonkman J., and P. Sclavounos. Engineering challenges for floating offshore wind turbines, 2005.

[28] J Spinneken, V Heller, S Kramer, M Piggott, and A Viré. Assessment of an advanced finite element tool for the simulation of fully-nonlinear gravity water waves. pages 1043–1050, 2012.

[29] J Spinneken, V Heller, S Kramer, M Piggott, and A Viré. Assessment of an advanced finite element tool for the simulation of fully-nonlinear gravity water waves. pages 1043–1050, 2012.

[30] P. K. Sweby. High-resolution schemes using flux limiters for hyperbolic conservation-laws. *Siam Journal on Numerical Analysis*, (21), 1984.

[31] F. Ursell, R. G. Dean, and Y. S. Yu. Forced small-amplitude water waves: a comparison of theory and experiment. *Journal of Fluid Mechanics*, (7), 1960.

[32] A. Viré, J. Spinneken, M.D. Piggott, C.C. Pain, and S.C. Kramer. Application of the immersed-body method to simulate wave–structure interactions. *European Journal of Mechanics - B/Fluids*, 55:330 – 339, 2016.

[33] Axelle Viré, Jiansheng Xiang, Frank Milthaler, Patrick Emmet Farrell, Matthew David Piggott, John-Paul Latham, Dimitrios Pavlidis, and Christopher Charles Pain. Modelling of fluid–solid interactions using an adaptive mesh fluid model coupled with a combined finite–discrete element model. *Ocean Dynamics*, 62(10):1487–1501, 2012. ISSN 1616-7228. doi: 10.1007/s10236-012-0575-z. URL `https://doi.org/10.1007/s10236-012-0575-z`.

[34] Axelle Viré, Johannes Spinneken, Matthew Piggott, Chris Pain, and Stephan C Kramer. Modelling of waves and wave-structure interactions using non-linear numerical models. 2014.

[35] Lei Wang, Shan Zuo, Y D. Song, and Zheng Zhou. Variable torque control of offshore wind turbine on spar floating platform using advanced rbf neural network. 2014:1–7, 03 2014.

[36] Sam Weller, Peter Davies, Philipp Thies, Violette Harnois, and Lars Johanning. Durability of synthetic mooring lines for ocean energy devices, 10 2012.