Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Civil and Environmental Engineering

Vilde Malmei

Floating offshore wind and mooring systems modelling using CFD

Master's thesis in Civil and Environmental Engineering
Supervisor: Hans Bihs
Co-supervisor: Ahmet Soydan and Widar Weizhi Wang
June 2023

NTNU
Norwegian University of
Science and Technology

Vilde Malmei

Floating offshore wind and mooring systems modelling using CFD

Master's thesis in Civil and Environmental Engineering
Supervisor: Hans Bihs
Co-supervisor: Ahmet Soydan and Widar Weizhi Wang
June 2023

Norwegian University of Science and Technology
Faculty of Engineering
Department of Civil and Environmental Engineering

NTNU
Norwegian University of
Science and Technology

Floating offshore wind and mooring systems modelling using CFD

A Thesis
submitted to the Department of Civil and Environmental Engineering
at the Norwegian University of Science and Technology

Master Thesis 2023

by

Vilde Malmei

Abstract

Floating offshore wind is essential in transitioning towards renewable energy and reaching net zero emissions. As these turbines can be placed in the deeper ocean, they also can provide more power from the higher and consistent wind speed. However, this also comes with the challenges of the harsh environment, making the structures more expensive as they need to withstand these large forces. In order to make the offshore wind industry competitive, cost reduction and safe design innovation are crucial. Numerical simulations with CFD play a significant role under these circumstances due to the possibility of accurate prediction of the physical loads, making the optimization of the structure better. Still, to make the CFD model reliable, validation is required by comparing the numerical simulation to experimental data.

This thesis aims to validate floating bodies, particularly focusing on a floating offshore wind turbine, using the numerical model REEF3D with the implemented algorithm of six degrees of freedom (6DOF), including a new fluid-structure interaction(FSI) method. REEF3D is an open-source numerical CFD solver developed at the Department of Civil and Environmental Engineering at NTNU, which solves the three-dimensional Reynolds-Average Navier-Stokes (RANS) equation. The domain is discretized using a finite difference method and level set method to obtain the floating body and the free surface within water and air. This is executed on a cartesian staggered grid with a direct forcing immersed boundary method. The convective term in the RANS equation is discretized by applying a fifth-order WENO scheme, while the term of pressure is obtained by Chorin's Projection resulting in a Poisson equation that is solved with the Bi-Conjugate Gradient Stabilized method. The Total Variation Diminishing Runge-Kutta scheme of third-order is utilized for both the free surface's time scheme and the momentum equations. For validation, the waves are simulated in a numerical wave tank with second-order Stokes theory, where the numerical wave tank is divided into three parts: a generation zone, a working zone, and a numerical beach, explained in the chapter on the numerical model.

Two 2D cases of simple geometry are first investigated before a more complex 3D case of a floating offshore wind turbine with mooring lines developed by SINTEF is

studied under the influence of three different regular waves. These simulations are then compared to experimental data, which involves heave, roll, and surge motions in 2D, and heave, pitch, and surge in 3D. Also, the simulated wave elevation is compared with the measurements. Simplified spring mooring lines are used for the floating platform, which is investigated in conjunction with the motions. A grid convergence study is performed on the cases to optimize the grid size, combining accuracy and computational efficiency. Further, a parameter analysis of the least favorable cases is conducted to improve the outcomes. Considering all the validation cases, the results show good agreement with the measurements.

Acknowledgments

This master's thesis was written in the spring of 2023 at the Department of Marine Civil Engineering within the area of floating offshore wind.

Firstly, I'm incredibly grateful to my supervisor, Hans Bihs, who afforded me the chance to explore this topic and his guidance and expertise on numerical wave modeling. In addition, I am thankful for the opportunity he gave me to present my work at the Coastal Engineering Day.

I would also like to express my sincere appreciation to my co-supervisors, Ahmet Soydan and Widar Weizhi Wang, for their excellent assistance through this process and for always being available for questions and discussion.
Thanks should also go to SINTEF, which gave me the privilege of accessing experimental data from the WINDMOOR project. Without this, the thesis would not be the same.

Finally, a special thanks to my friends and family for valuable discussions and for always showing their support and guidance in both good and more stressful times.

The simulations were performed on the supercomputer Betzy provided by Sigma2 - the National Infrastructure for High-Performance Computing and Data Storage in Norway.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

The rapid climate changes in recent years have increased the demand for renewable technologies to fulfill the Paris Agreement's objectives and reach net-zero emissions[47]. In this regard, the wind industry is a great contributor, whereas the offshore wind industry's interest has grown rapidly[125][76][32]. Removing water depth constraints for Offshore Wind opens access to the best sites for wind energy production, as nearly 80% of the world's offshore wind resource potential is located at water depths exceeding 60 meters[31]. In addition, the floating offshore wind is capable of supplying large-scale energy production directly to global markets, as 2.4 billion of the population are settled within 100km of the coastal zone[31]. However, deeper water also comes with challenges with harsher metocean conditions and more extreme loads, leading to higher costs to make reliable structures[81]. In order to make this industry competitive, a cost reduction is needed, which can be obtained by high facility resources, like Computational Fluid Dynamics (CFD)[77][134][133].

Today bottom-fixed wind farms are competitive with other sources of energy[77]. However, suitable shallow areas for bottom-fixed wind farms are scarce and often already used by others[67]. Hence, the potential for larger wind farms is better offshore with the advantage of explication the stronger and more consistent winds[91][67]. However, deeper water stops bottom-fixed developments, as fixed-bottom turbines are inefficient at a depth of more than 60m[81][124][76]. A floating offshore wind turbine with mooring lines is then required[134][133][81].

For a long time, analytical methods using potential theory have dominated the calculations of hydrodynamic loads when designing in marine environments[3][68]. However, this has a disadvantage due to its difficulties in calculating the effects of

non-linearity and viscosity of the fluid[34][3]. The complex free surface surrounding floating structures in the ocean consists of high non-linearity and phenomena such as wave breaking and turbulence, giving an intense fluid-structure interaction[68]. Therefore, an accurate hydrodynamic load prediction is essential to design safe and lower costs, especially for small-scale structures like offshore floating wind[27]. A powerful tool for understanding these phenomena is Computational Fluid Dynamics (CFD)[77]. This numerical method is very computationally demanding, but in return, it can determine the majority of the fluid physics with just a small number of assumptions using the Navier-Stokes equation[12]. With these accurate results of the hydrodynamic forces, a deeper understanding and new innovative design solutions can be evolved to further development of floating offshore wind[7]. Traditionally these studies were done by physical experiments, but with increasing computational power, numerical simulations can obtain more accurate results and save costs[39]. However, the numerical simulations must be validated against experimental data to obtain reliable results[13][27][39].

1.2 State-of-the-Art

1.2.1 Numerical wave tanks based on CFD

Within the area of CFD, several models are developed based on the Navier-Stokes equations to simulate waves with proper assumptions[70][12], and various methods are available to represent this[35]. The free surface can be obtained by discretization methods, like finite difference, finite volume, or finite difference methods, and techniques like the Level Set Method (LSM)[75] or Volume Of Fluid (VOF)[43]. Furthermore, the calculations can be performed in different ways based on the inclusion of air, where a single-flow calculation does not include the air[108], while the two-phase includes it[12]. For the approach of fluid-structure interactions, the Arbitrary Lagrangian-Eulerian (ALE) methods, which is a one-phase solver, have earlier been used[87][117][107][68]. These methods follow the body's movement by modifying the underlying mesh when the floating body moves[13]. However, when the free surface gets complex, as it often does in the deep ocean, this approach will have limitations in usability[68]. With an overlapping mesh implementation, where the fluid and the object have two different meshes, re-meshing is avoided[17]. Peskin[84] presented another technique for fluid-structure interactions, named The Immersed Boundary (IB) method, that can have different applications of boundary conditions[68]. These boundary conditions can either take form as an extension of the momentum equation, with an extra source term[33], or with an approach of direct forcing[113], which has a fictive force field that illustrates the boundary effects in the solution of the flow equa-

tions. Combining this with the level set function introduced by Sanders et al.[94], the boundary can be represented as a zero-contour at the water and air interface. The level-set immersed boundary method is a two-phase solver and is a good approach to investigating complex geometries[7][13][135], such as in Kan et al.[57], and Calderer et al.[16]. Moreover, also the Volume of Fluid method (VOF) has been used to implement different CFD codes, like the toolboxes of OpenFOAM from Jacobsen et al.[50] and Higuere et al.[42] and ReFRESCO[115]. For turbulence modeling, two models of RANS models are developed, one and two-equations models[98]. Examples of one-equation models that use the average time transport equation are SA (spalrat-Allmaras)[129], Baldwin-Barth[5], and Prandtl's[109][60], where the SA-model is the most frequently used[98]. For CFD modeling of wind turbines, the two-equation model of turbulence, $k - w$, and $k - \epsilon$, is commonly applied[30]. Also, $k - \omega$ SST is an accurate and popular submodel of $k - \omega$ [127][92][98].

This thesis utilizes the CFD module in the open-source hydrodynamics solver REEF3D [12], which uses the level set method for the representation of both the boundary around the structure and the free surface with the direct forcing immersed boundary method for fluid-structure interactions[12][70]. The application shown by Martin et al.(2021) [70] is used for the mooring systems and for the 6DOF algorithm with the direct immersed boundary method. REEF3D also has more implemented modules than the CFD module, based on the different flow and wave conditions[119]. The SFLOW[120] module can be used for shallow conditions, and for large-scale phase resolving, the fully nonlinear potential flow solver, FNPF[14], is considered as beneficial. All of these have been used for diverse scenarios in the oceanic environment with good results, for instance, investigation of the breaking wave concept using Large eddy simulation for turbulence modeling[1], a quasistatic mooring model for floating structures[71], wave propagation over a sloping bed with breaking waves[120], and transformation and propagation of waves over complex topography[123].

A coupling technique for REEF3D is also presented in [122]. Here, a combination of the fully nonlinear potential flow solver (FNPF) and the model of computational fluid dynamic (CFD) is possible. This combination is valuable when conducting simulations over a large area, requiring precise values of a particular location. The CFD can provide accurate results. However, this requires a high computational effort, making it unfavorable for a larger scale. Coupling between these two modules leads to an optimization between productivity and accuracy, which could be an advantage for upscaling wind turbines and wind farms[79].

1.2.2 Numerical simulations of Wind Turbines using CFD

Multiple numerical simulation approaches on Wind Turbines are utilized in diverse fields with the advantages of high accuracy and turbulence simulation[77][98]. One of the first to use CFD models on Wind Turbines was Sorensen and Hansen [101][23], which used the incompressible RANS equation together with the turbulence model $k-\omega$ SST in two different models[98]. However, the unpredicted flow separation of wind speed higher than 10 m/s limited the usage. Therefore, Bazilevs et al. [101] established a model of FSI that was invented to predict the torque accurately. Another CFD performance was done in order to determine an appropriate shape for the airfoil, Maizi et al.[69], that used a solver influenced by the compressible URANS of 25 different airfoils. This study pointed out that $k-\omega$ SST had the maximum exactitude of all the turbulence models[98].

For turbines placed on the ocean, Stevenes et al.[103] have used CFD modeling to optimize the arrangement of Wind Turbines in an offshore wind farm with Large eddy simulations[98]. Focusing on how the inflow orientation of the water affects the turbines. In another case, a flow motion above two floating wind turbines was studied by Yu and Agel[128], using Reynolds Averaged Navier Stokes equations (RANS) in the CFD estimation with $k-\omega$ SST and SA as turbulence models. The outcome was that $k-\omega$ SST was the most suitable model for accurate turbulence modeling. The simulation also showed a good match with measurements in the model and full scale[98].

Under the conditions of yaw settings, this has also been investigated to optimize and build a new numerical model[90]. This model can give the yaw settings good estimations of the power output when it is both conventional and optimized. The alignment angles were examined by a simulation performed by Gebrad et al.[38] using an LES model, which showed promising results when compared with measured data[98].

1.3 Objectives of the Study

This thesis aims to validate the implemented algorithm of 6DOF with a particular focus on a floating offshore wind platform using REEF3D with its new approach of fluid-structure interaction(FSI) and mooring model[70]. First, there are studied two 2D cases for validations purpose. This is of a heaving cylinder and a floating box in waves. For the heaving cylinder, the heave motion is investigated in a simulation from a release of an initial position to the water and compared to experimental data. For the following case, a rectangular box is simulated in regular waves, where heave, roll, and surge motions are studied and compared to measurements.

The third and primary investigation is of a floating offshore wind platform, WINDMOOR, from SINTEF. Three cases of different regular waves are simulated and compared to measurements carried out of SINTEF in the SINTEF's oceans basin. Heave, pitch, and surge motions are investigated to validate the 6DOF algorithm and mooring lines. Mooring lines are simplified with springs, holding the platform stable.

The CFD module of the open-source code REEF3D is used in all cases. This code is developed at the Norwegian University of Science and Technology (NTNU) in the Department of Civil and Environmental Engineering[9][12]. The structure of this thesis starts with an explanation of the governing equations and the framework of the model as a methodology and theory chapter. Next, the validation cases are introduced and discussed. The WINDMOOR experiment is then explained, and the simulation results are examined and discussed. Finally, a conclusion and outlook are given at the end of the document.

1.4 Definition of the movements of a floating body

The movement of a floating body is represented by 6 degrees of freedom (6DOF)[34]. These motions consist of three displacements and rotations, making up all the movements necessary to present the body in three dimensions. Along the x, y, and z-axis, the three translations are defined as surge, sway, and heave. The three rotations are defined as roll, pitch, and yaw around the x, y, and z-axis, respectively[52][68]. These movements are illustrated in Fig. 1.1.

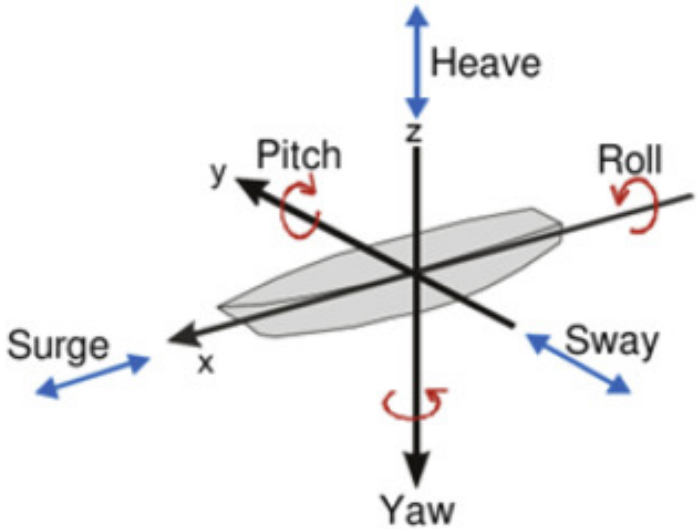


Figure 1.1: The six degrees of freedom[64]

Chapter 2

Numerical Model

This chapter represents the numerical solution methods used in this thesis, where the open-source hydrodynamics framework software, REEF3D[12], is utilized. The CFD module of REEF3D is applied to simulate the floating structure. This module solves the three-dimensional Navier-Stokes equations [12]. Furthermore, it will give detailed information on the flow process, which makes it an appropriate model for the complex case of floating structures[12].

2.1 Governing equations

The three-dimensional, incompressible Navier-Stokes equations are the governing equation[12]. These equations describe the incompressible fluid flow with the base of mass conservation and linear momentum conservation[12]. The continuity equation represents the conservation of mass:

$$\frac{\partial u_i}{\partial x_i} = 0, \tag{2.1}$$

Navier Stokes equation is established from the conservation of linear momentum[116].

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = \frac{\partial}{\partial x_j}(\mu \frac{\partial u_i}{\partial u_j}) - \frac{1}{\rho}\frac{\partial p}{\partial x_i} \tag{2.2}$$

If Eq. (2.1) and (2.2) are put together, it will then express the Reynolds-averaged Navier-Stokes equations (RANS equations)[68]:

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[(\nu + \nu_t)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] + g_i, \tag{2.3}$$

with u_i the velocity vector, ρ the fluid density, p the pressure, ν and ν_t the kinematic and turbulent viscosity, and g_i the gravity acceleration vector[12].

The first term in Eq. (2.3) is a velocity variation over time, and the second is a convection term[68]. These two terms constitute the equation's left side and the inertial forces[68]. The right side consists of three terms, where the first is a pressure force, the second is a viscous force, and the third is an external force[68][35].

2.1.1 Discretization Approaches

An appropriate discretization method must be selected to solve the Navier-stokes equations[35]. The goal is to replace the continuous problem with a discrete domain with nodes of a finite number so that the differential equation can be described as equations of an algebraic system[39][35]. This method will thus approximate the differential equations[39][35].

The three main approaches are: Finite Element Method, Finite Volume Method and Finite Differences Method[68].

Finite Volume Mehod

The starting point of the finite volume method is the integral form of the conservation equations, where the grid of the solution domain is divided into control volumes of a finite number[35]. First, the conservation equation is calculated at the computational node in the centroid. Then, the algebraic equation for each control volume is obtained by interpolation and integral approximations. An interpolation is performed to get the values from the centroid to the surface. Then approximation integrals of the surface and volume are done using suitable quadrature formulas[35]. This method fits any grid and is thus appropriate for complex geometries, making it a preferred choice for a CFD simulation[35]. One disadvantage of the finite volume method is that using higher-order schemes can cause trouble in 3D[35]. In such cases, another method should be used.

Finite Element Method

The finite element method is divided into unstructured discrete volumes or finite elements[35]. These volumes and elements are often formed as triangles or quadrilaterals in 2D and tetrahedra and hexahedra in 3D. A weight function makes this method different from the finite difference method because the integration over the domain is now with other weighted values of the equation[35].

Finite Difference Method

The Finite Difference Method was established in the 18th century by Euler, making it the oldest numerical method for solving PDEs[35]. This is also the easiest method to use with simple geometries[35]. The method is based on the differential form of the conservation equation, where a grid covers the solution domain with many grid points[35]. This makes it possible to create algebraic equations that replace the differential equation at each grid point by approximations using Taylor series expansions[8]. The structural grid is preferable when using the Finite Difference Method, but other grids can also be used.

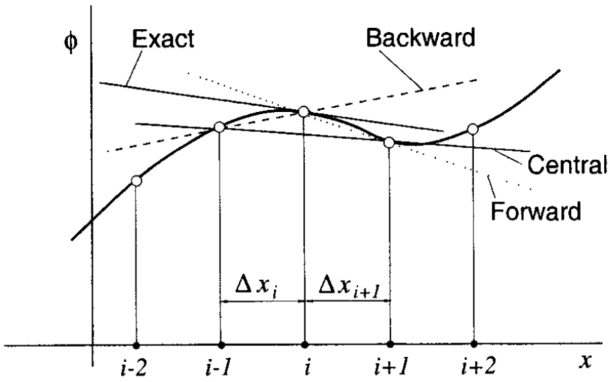


Figure 2.1: Spatial discretization approaches showing the approximation of the derivatives definition[35]

In REEF3D, the Finite difference approach is used[12]. In this approach, the geometric domain must be discretized by defining a numerical grid, often locally structured[68]. Nodes with a selected approximation of the derivatives form this structured grid[35][68]. The differential form of the generic scalar conservation equation is a linear equation and the starting point for each node's approximation, which is also uniquely identified[35]. This approximation of all the grid nodes entails a system of linear equations where the unknowns is defined as the variable values and gives the approximated solution to the partial differential equation[35].

As the solution to the differential equation is an approximation, this also involves an error at each step[68][35]. The truncation error is an example of this, which occurs from the truncation of the Taylor series[35]. Therefore, specific properties must be fulfilled to determine the solution's quality[35][68].

The most important ones are:

- Consistency: Consistency is obtained when the discretized equation and the exact one become the same as the grid size, and the time step tends toward zero[35].
- Stability: The solution is stable if errors do not increase as the grid size reduces [35].
- Convergence: The solution is convergent by fulfilling the consistency and stability properties[35]. In other words, a solution is convergent if the discretized differential equation is approaching the exact solution, as the grid size is made smaller and smaller towards zero[35].

2.2 Convection discretization

Eq. (2.3) contains a convection term that needs to be discretized before it can be solved numerically[68][35]. There are multiple ways to do this. In REEF3D, the finite difference method is used to trait the convection discretization[12][68]. The number of nodes used for approximation is also called a stencil and is used to check the accuracy[12][68]. A wider stencil presents a higher order of the schemes and, thus, higher accuracy[35][55][68][12].

Some of the schemes that are implemented in the code are:

- First-Order Upwind (FOU)
- Central Difference Scheme (CDS)
- Sharp and Monotonic Algorithm for Realistic Transport (SMART) scheme
- Quadratic Upwind Interpolation of Convective Kinetics (QUICK) scheme
- Weighted Essentially Non-Oscillatory (WENO) scheme

First-Order Upwind (FOU)

Courant et al.[25] introduced this scheme in 1952 [68]. It is a first-order scheme that calculates the first derivative from the values of the cells upstream of the flow relative to the cell considered in the center[35][68][55].

$$\frac{\partial \Phi}{\partial x_j} = \frac{\Phi_i - \Phi_{i-1}}{\Delta x_j} \tag{2.4}$$

Central Difference Scheme (CDS)

The Central Difference Scheme uses two points to calculate the cell with the central node and is, therefore, of second order[35]. Furthermore, the two points are taken from both sides of the cell, one upstream and one downstream, making the implementation of this scheme easy compared to other upwind schemes due to the advantage of not having to check the flow direction[35][68][55].
The second-order scheme is given by:

$$\frac{\partial \Phi}{\partial x} = \frac{\Phi_{i+1} - \Phi_{i-1}}{2\Delta x} \tag{2.5}$$

CDS can also be implemented with two points at each side and will then be a fourth-order scheme[68][35].

$$\frac{\partial \Phi}{\partial x} = \frac{-\Phi_{i+2} + 8\Phi_{i+1} - 8\Phi_{i-1} + \Phi_{i-2}}{12\Delta x} \tag{2.6}$$

Sharp and Monotonic Algorithm for Realistic Transport (SMART) scheme

This scheme was developed in 1988 by Gaskell and Lau [37] and is used on the steady-state transport equation for the convection term approximation[55]. The technique used is a curvature-compensated convective transport equation[55][37].

Quadratic Upwind Interpolation of Convective Kinetics (QUICK) scheme

This high-order scheme was developed by Leonard in 1975 [66]. It is based on an approximation considering three points for calculating the cell value[68]. Depending on where the flux arrives, the scheme can vary on which side it takes its values[68]. This results in a parabola shape from either two points taken from the upstream or downstream[68].

Weighted Essentially Non-Oscillatory (WENO) scheme

In 1996 Jiang and Shu[51] introduced the Weighted Essentially Non-Oscillatory (WENO) Scheme, which is also used in this thesis. It is of a higher order, making it computationally demanding. However, it will be a good choice for extensive gradient calculation because of its high accuracy. Three weighted ENO stencils make the base of this scheme with a local smoothness, where the highest weighted stencil is assigned with the highest smoothness and will thus count the most in the scheme[12][68].
For the implementation of the scheme for the level-set function, the equation below can be utilized[75].

$$\Phi_x = \begin{cases} \Phi_x^- \ if \ \ U_1 > 0 \\ \Phi_x^+ \ if \ \ U_1 < 0 \\ 0 \ if \ \ U_1 = 0 \end{cases} \tag{2.7}$$

In the x-direction, the discretization is formulated as the following:

$$\Phi_x^{\pm} = \omega_1^{\pm}\Phi_x^{1\pm} + \omega_2^{\pm}\Phi_x^{2\pm} + \omega_3^{\pm}\Phi_x^{3\pm}. \tag{2.8}$$

The three stencils are defined as:

$$\begin{aligned} \Phi_x^{\pm} &= \frac{1}{3}q_1^{\pm} - \frac{7}{6}q_2^{\pm} + \frac{11}{6}q_3^{\pm}, \\ \Phi_x^{\pm} &= -\frac{1}{6}q_2^{\pm} + \frac{5}{6}q_3^{\pm} + \frac{1}{3}q_4^{\pm}, \\ \Phi_x^{\pm} &= \frac{1}{3}q_3^{\pm} + \frac{5}{6}q_4^{\pm} - \frac{1}{6}q_5^{\pm}. \end{aligned} \tag{2.9}$$

with,

$$\begin{aligned} q_1^- &= \frac{\Phi_{i-2} - \Phi_{i-3}}{\Delta x}, q_2^- = \frac{\Phi_{i-1} - \Phi_{i-2}}{\Delta x}, q_3^- = \frac{\Phi_i - \Phi_{i-1}}{\Delta x}, \\ q_4^- &= \frac{\Phi_{i+1} - \Phi_i}{\Delta x}, q_5^- = \frac{\Phi_{i+2} - \Phi_{i+1}}{\Delta x} \end{aligned} \tag{2.10}$$

and,

$$\begin{aligned} q_1^+ &= \frac{\Phi_{i+3} - \Phi_{i+2}}{\Delta x}, q_2^+ = \frac{\Phi_{i+2} - \Phi_{i+1}}{\Delta x}, q_3^+ = \frac{\Phi_{i+1} - \Phi_i}{\Delta x}, \\ q_4^+ &= \frac{\Phi_i - \Phi_{i-1}}{\Delta x}, q_5^+ = \frac{\Phi_{i-1} - \Phi_{i-2}}{\Delta x} \end{aligned} \tag{2.11}$$

The weights are written as:

$$\omega_1^{\pm} = \frac{\alpha_1^{\pm}}{\alpha_1^{\pm} + \alpha_2^{\pm} + \alpha_3^{\pm}}, \omega_2^{\pm} = \frac{\alpha_2^{\pm}}{\alpha_1^{\pm} + \alpha_2^{\pm} + \alpha_3^{\pm}}, \omega_3^{\pm} = \frac{\alpha_3^{\pm}}{\alpha_1^{\pm} + \alpha_2^{\pm} + \alpha_3^{\pm}} \tag{2.12}$$

and

$$\alpha_1^{\pm} = \frac{1}{10}\frac{1}{\left(\widetilde{e} + IS_1^{\pm}\right)^2}, \alpha_2^{\pm} = \frac{6}{10}\frac{1}{\left(\widetilde{e} + IS_2^{\pm}\right)^2}, \alpha_3^{\pm} = \frac{3}{10}\frac{1}{\left(\widetilde{e} + IS_3^{\pm}\right)^2} \tag{2.13}$$

with the regularisation parameter $\widetilde{e} = 10^{-6}$ and the following smoothness indicators:

$$\begin{aligned} IS_1^{\pm} &= \frac{13}{12}\left(q_1 - 2q_2 + q_3\right)^2 + \frac{1}{4}\left(q_1 - 4q_2 + 3q_3\right)^2, \\ IS_2^{\pm} &= \frac{13}{12}\left(q_2 - 2q_3 + q_4\right)^2 + \frac{1}{4}\left(q_2 - q_4\right)^2, \\ IS_3^{\pm} &= \frac{13}{12}\left(q_3 - 2q_4 + q_5\right)^2 + \frac{1}{4}\left(3q_3 - 4q_4 + q_5\right)^2, \end{aligned} \tag{2.14}$$

2.3 Turbulence treatment

Turbulence is a phenomenon that occurs in almost all hydraulic flows and causes difficulty in obtaining reliable predictions of numerical methods[93]. The difficulty is related to the motion of the eddies, which are unsteady and irregular, leading to an increase in momentum[93]. This motion influences both the friction of the boundary and flow development, which results in flow losses (dissipation) through pipes and around structures, together with an impact on the velocity and pressure distribution, thus, the forces on the structure[93]. Therefore, getting a realistic simulation of the turbulence effect is very important[93]. One example of a turbulence scenario is wave breaking[12], which can occur when the wave steepness gets too high[121]. There are three main approaches for calculating the turbulence flow in a numerical model: Direct numerical simulation (DNS), Large eddy simulation (LES), and Reynolds-Average Navier-Stokes (RANS)[93].

2.3.1 Direct numerical simulation (DNS)

Direct numerical simulation (DNS) is the most computationally demanding method because it resolves all the scales of the turbulence motion[86]. This means that all the sizes of the eddies are calculated without any model. The limitation of the applicability of this approach is given by low or moderate Reynold number flow because of the rapidly increasing computational cost with increasing Reynolds number[86].

2.3.2 Large eddy simulation (LES)

Large Eddy Simulation proposed from Smagorinsky[100] is a method where the small scale of turbulence is approximated while the largest scales are resolved, making this method faster on the computer than the DNS[68][93].

2.3.3 Reynolds-Average Navier-Stokes method (RANS)

The RANS method is based on average quantities of the flow where the turbulence's fluctuation is average out and will hence have less computational time than DNS and LES[35][68]. This reduction of computational time is an advantage for many engineering perspectives because of the need for simulation on a large scale. It has therefore been the primary approach in this industry for years[136][68].
When the average velocity field is used to solve the Reynolds equations, some unknown parameters appear, called the Reynolds stresses[86]. These unknowns need a turbulence model to be solved, which can be a turbulent viscosity hypothesis or from Reynold-stress transport equations[86]. In the code of REEF3D, there are implemented multiple turbulent models[68][12]: Explicit Algebraic Reynold Stress Models (EARSM) [118], $k-\epsilon$ [63], Unsteady Reynold-Averaged Navier-Stokes (URANS)[68], LES and $k-\omega$ model [62].

2.3.4 Turbulence model k-w model

A popular and often used turbulence model is the $k-\omega$, and it will therefore be explained in more detail. This is a two-equation model of turbulence[126], which entails that both convection and diffusion effects can be accounted for in the turbulent flow[22]. The turbulent kinetic energy, k, is the first transported variable, and the specific turbulent dissipation ω is the second transported equation[22].

The Boussinesq-approximation [15] of the Reynolds stress replaces the term in Eq. (2.3)[68]:

$$\overline{u_i u_j} = \nu_t \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right) - \frac{2}{3} k \delta_{ij} \tag{2.15}$$

and a new RANS equation is obtained:

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho} + \frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\nu + \nu_t \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} k \delta_{ij} \right] + g_i \tag{2.16}$$

the eddy viscosity ν_t from Eq. (2.15) can be calculated from the subsequent equation:

$$\nu_t = c_\mu \frac{k}{\omega} \tag{2.17}$$

which includes two unknown variables, the specific turbulent dissipation ω and the turbulent kinetic energy k. These variables can be obtained by:

$$\frac{\partial \omega}{\partial t} + u_j \frac{\partial \omega}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_\omega}\right)\frac{\partial \omega}{\partial x_j}\right] + \frac{\omega}{k}\alpha P_k - \beta\omega^2 \tag{2.18}$$

$$\frac{\partial k}{\partial t} + u_j \frac{\partial k}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + P_k - \beta_k k\omega \tag{2.19}$$

Here, the values of the coefficient is $\sigma_w = 2$, $\beta = \frac{3}{40}$, $\sigma_k = 2$, $\beta_k = \frac{9}{100}$ and $\alpha = \frac{5}{9}$ and the turbulent production rate is P_k[68].

The mean rate of strain S can be significant in the wave flow field, characterized by the oscillatory flow motion[68]. Wall laws are used in the turbulence model to account for the large stain in the boundary layers because the model does not resolve these layers explicitly[35][68]. A limited formulation restricts the eddy viscosity ν_t to avoid the overproduction of turbulence outside the boundary layer in the highly strained flow[68]. This formulation is obtained after Durbin [29][68], presented in Eq. (2.20):

$$\nu_t = min(\frac{k}{\omega}, \sqrt{\frac{2}{3}}\frac{k}{|S|}) \tag{2.20}$$

Additional turbulent damping is added to avoid inaccurate results of the strain S near the boundary between air and water[68]. The free surface effects on the turbulence will then get a more realistic representation[74]. The dissipation of specific turbulent at the free surface is determined[68] by Eq. (2.21):

$$\omega_s = \frac{c_\mu^{\frac{-1}{4}}}{\kappa} k^{\frac{1}{2}} \cdot (\frac{1}{y'} + \frac{1}{y^*}) \tag{2.21}$$

where $\kappa = 0.4$ and $c_\mu = 0.07$. The turbulent length scale has a virtual origin y' at 0.07 times the mean water depth, which was found empirically[46][68]. y^* is the distance from the nearest wall, and when including this variable, a smooth transition from the free surface value to the boundary value of w is obtained[68].

The Dirac delta function $\delta(\phi)$ is used for the damping, which is only carried out around the interface[54][68].

$$\delta(\phi) \begin{cases} \frac{1}{2\epsilon}(1 + cos(\frac{\pi\phi}{\epsilon})) & if \ \|\phi| < \epsilon \\ 0 & else \end{cases} \tag{2.22}$$

2.4 Pressure treatment

In the Navier-Stokes equation of an incompressible flow, the pressure-velocity coupling makes it difficult to obtain the pressure value directly[12][68]. Furthermore, the non-linearity also makes the process very computationally demanding[35][68]. So when solving this equation, a guess for the pressure is needed to give a velocity, where the velocity also depends on satisfying the continuity equation[35][68]. To ensure this requirement, a pressure velocity coupling algorithm must be used[55]. In REEF3D, there are different methods of such an algorithm, which are[12][68]: Semi Implicit Method for Pressure Linked Equation (SIMPLE)[82], Pressure Implicit with Split Operator (PISO)[48], and the Projection Method (PJM)[24].

2.4.1 Semi Implicit Method for Pressure Linked Equation (SIMPLE)

Patankar and Spalding developed Simple[82] in 1972[68]. This method is an iterative algorithm that takes the pressure into the continuity equation[68]. Then, the solution of the pressure field is used to correct the velocities[68]. The goal is to achieve convergence, so this step needs to be repeated until this is accomplished[68]. An implicit calculation is done to obtain the discretized momentum and pressure correction equations, and an explicit calculation is done for the velocity correction[21]. Therefore this method is called the "Semi-Implicit Method"[21].

2.4.2 Pressure Implicit with Split Operator (PISO)

PISO is an extension of the SIMPLE algorithm and was proposed in 1986 by ISSA[48][68]. This algorithm has lesser computing effort and is without iterations and with large time steps[68]. It uses predictor-corrector steps to satisfy mass conservation, with one predictor step and two corrector steps[20][68].

2.4.3 Projection Method (PJM)

Chorin's Projection Method[24] was developed in 1968 and is utilized in this thesis. This method is also the most used of these three methods presented[68]. It uses an intermediate velocity u_i^* to compute the momentum equation, neglecting the pressure term[68]:

$$\frac{\partial u_i^*}{\partial t} + u_j^n \frac{\partial u_i^n}{\partial x_j} = + \frac{\partial}{\partial x_j} \left[(\nu + \nu_t) \left(\frac{\partial u_i^n}{\partial x_j} + \frac{\partial u_j^n}{\partial x_i} \right) \right] + g_i \tag{2.23}$$

The intermediate velocity u_i^* is then corrected when taking the projection step, and the solution of the time-step u_i^{n+1} is obtained together with the pressure term[68]:

$$\frac{\partial (u_i^{n+1} - u_i^*)}{\partial t} = - \frac{1}{\rho(\phi^n)} \frac{\partial \rho^{n+1}}{\partial x_i} \tag{2.24}$$

The pressure term p should be known at time-step $n+1$ for calculating the terms on the right side of Eq. (2.24)[68]. Following the fact that the divergence of the continuity equation of an incompressible flow is zero, and therefore also u_i^{n+1} should be zero, this is obtained and results in a Poisson equation[68]:

$$\frac{\partial}{\partial x_i} \left(\frac{1}{\rho(\phi^n)} \frac{\partial \rho^{n+1}}{\partial x_i} \right) = \frac{1}{\Delta t} \frac{\partial u_i^*}{\partial x_i} \tag{2.25}$$

2.5 Iterative solver

Direct methods, iterations, and multi-grid methods are all methods that can be used to solve the Poisson equation in Eq. (2.25)[68][35]. However, the computational costs of direct methods are quite high, so in CFD problems, iterative methods are a good alternative, which is used for non-linear problems and sparse linear systems[55]. Furthermore, with an iterative method, the equation can be used to systematically improve a made guess for the solution[55]. In REEF3D, multi-grid and iterative solvers are implemented with pre-conditioners, and the library of HYPRE (Parallel High-Performance Pre-conditioners) makes it possible to solve large linear systems on parallel computers[12][68]. Some of the implemented systems are explained below[68]:

PFMG

Ashby and Falgout[4] introduced the semi-coarsening multigrid method PFMG in 1996. This method uses 9 and 27-point stencils, in 2D and 3D, respectively, to solve scalar diffusion equations discretized on rectangular grids[68].

SMG

In 1968, Stone[105] introduced SMG. This method is very similar to the PFMG but is more robust, which also leads to more computational costs per iteration[68]. The advantage is for anisotropies that vary across the domain, either in direction or strength[68].

AMG(BoomerAMG)

Henson and Tang[132] developed AMG(BoomerAMG) in 2002. This is an unstructured Algebraic Multi-Grid (AMG) that doesn't need explicit grid geometry, and the system can therefore solve more types of problems[68]. The disadvantage is that this method is a complex algorithm, and low memory, computational requirements, and good convergence become harder to achieve[68].

Bi-Conjugate Gradient Stabilized (BiCGSTAB)

In this thesis, the Bi-Conjugate Gradient Stabilized (BiCGSTAB)[41] method is used with preconditioning to get faster convergence[114]. The BiCGSTAB is an improvement of the Conjugate Gradient (CG) solver and the Bi-Conjugate Gradient method (BiCG) solver, which are all global solvers used for non-linear equations[114][55]. The CG was first developed by Hestenes and Stiefel in 1952[41], with the advantage of searching for the minimum in one direction while minimizing a function in many directions[55]. This method made the convergence faster but was limited to symmetric systems. Van der Vorst then developed the BiCG method in 1992[36], where a transpose matrix converted the non-symmetric system to a symmetric one[55]. However, this conversion required more computational cost and did not converge faster[55]. Therefore, the BiCGSTAB method was made to improve the solutions and increase the convergence[55][68].

2.6 Time treatment

In an unsteady flow, time-dependent terms play an essential role. In such cases, an accurate discretization scheme is essential[55][35]. The main difference between the time and space coordinates is that the force will influence the future flow for the time step and not backward as it will in space[35]. There are both explicit and implicit methods for the time treatment[35].

In REEF3D there are implemented several explicit methods of discretization schemes for time treatment, which are advantageous for using higher-order schemes due to

their easy construction and application[12]. The schemes in REEF3D are Adams-Bashforth, second, third, and fourth-order Total Variation Diminishing (TVD) Runge-Kutta[55][12].

Adam-Baschforth scheme

The Adams-Bashforth method is of second order[10] and, therefore, second-order accuracy[56]. It uses values of the solution of the two earlier time steps to approximate a solution of the next[55]. The equation below, Eq. (2.26), is an example of how the method can be applicated to the level set function[55]:

$$\phi^{(n+1)} = \phi^n + \frac{\Delta t_n}{2}\left(\frac{\Delta t_n + 2\Delta t_{n-1}}{\Delta t_{n-1}} L(\phi^n) - \frac{\Delta t_n}{\Delta t_{n-1}} L(\phi^n)\right) \tag{2.26}$$

Total Variation Diminishing (TVD) Runge-Kutta

Robustness, monotonicity preserving, and high-order accuracy are highly interesting when discretizing spatial terms[28]. Total variation diminishing (TVD) schemes have the strength of preventing violating monotonicity in the solution that might occur if the differencing schemes cause oscillations in the solutions, which is often the case when using high-order differencing schemes[28]. Therefore a TVD scheme developed by Harten[40] is a good solution to this problem[28]. Runge-Kutta methods use points between t_n and t_{n+1} rather than earlier points to overcome the starting multipoint method that can occur in non-physical solutions[35]. The method of Runge-Kutta consists of different orders of accuracy, where the computational cost increases with the higher order[68].

This thesis uses a third-order TVD Runge-Kutta for momentum equations and the free surface level set method. A Runge-Kutta scheme after Schu, C. W., and Osher in 1988[99] with three Euler steps is shown below in Eq. (2.27). This scheme uses three steps for calculation and is more computationally demanding than the second-order Runge Kutta and Adam-Baschforth schemes[12][55].

$$\begin{aligned}
\phi^{(1)} &= \phi^n + \Delta t L(\phi^n) \\
\phi^{(2)} &= \frac{3}{4}\phi^n + \frac{1}{4}\phi^{(1)} + \frac{1}{4}\Delta t L\left(\phi^{(1)}\right) \\
\phi^{n+1} &= \frac{1}{3}\phi^n + \frac{2}{3}\phi^{(2)} + \frac{2}{3}\Delta t L\left(\phi^{(2)}\right)
\end{aligned} \tag{2.27}$$

2.6.1 Adaptive Time Stepping

The Courant-Friedrichs-Lewy[80] (CFL) condition holds stability in the simulation, which is important for time discretization[68]. This condition entails that the time step is sufficiently small and is within the computational grid size[27][68]. An example of a CFL can be:

$$C = \frac{u\Delta t}{\Delta x} \leq C_{max} \tag{2.28}$$

where the dimensionless Courant number is C, Δt is the time-step, u is the velcoity magnitude and Δx is the length of the interval. C_{max} can vary but often has the value of 1[68]. Adaptive time stepping is used to make sure the limit of the Courant number is held[68]. With this adjustable time step, the CFD code will automatically reduce the time step if it exceeds the limit so that the Courant number gets lower than the limit for the next iteration[55]. This method is based on the diffusion D, velocity u and the source term S, where the time-step size Δt is calculated as follows[68][55]:

$$\Delta t \leq 2 \left(\left(\frac{|u|_{max}}{dx} + D \right) + \sqrt{\left(\frac{|u|_{max}}{dx} + D \right)^2 + \frac{4|S_{max}|}{dx}} \right)^{-1} \tag{2.29}$$

with

$$D = max(\nu + \nu_t) \cdot \left(\frac{2}{(dx)^2} + \frac{2}{(dy)^2} + \frac{2}{(dz)^2} \right) \tag{2.30}$$

2.7 Numerical grid

A Cartesian grid is used in the numerical model to employ high-order discretization schemes[35][68]. This grid is one of the most straightforward systems[18]. It is built of bricks or cubes with a cartesian axis system, making it easy to implement numerical algorithms and define coordinates to geometries[39][18]. In this model, the grid is staggered, meaning that the velocity and pressure scalar variables are placed at different cell locations[39][68]. Fig. 2.2 shows the setup of the staggered cartesian grid, with the dots illustrating the pressure and the dash the velocity. The velocity or momentum is located at the cell faces, and the pressure is stored in the cell centers of the control volumes, preventing error in the discretization [68][35][19].

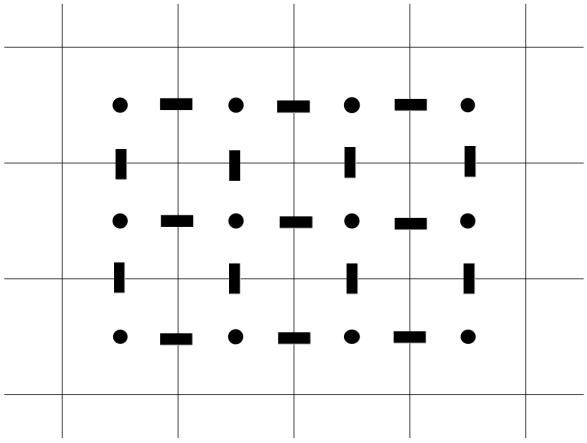


Figure 2.2: Illustration of the Staggered Cartesian grid[9].

High accuracy can be provided if all the solid geometry consists of orthogonal surfaces[18]. The disadvantage of this grid happens when the geometry of the solid boundary is irregular and non-orthogonal[18][35]. Then, higher meshing is needed to ensure accuracy, which also entails higher computational effort[18][65]. Floating bodies can also be challenging to generate a high-quality grid because of their irregularities and sharp edges[68]. An immersed boundary method is used to overcome the problem of irregularities and non-orthogonality where the boundary crosses the underlying Cartesian grid[68][55]. Hence, this cut creates a new boundary of the grid cells[68][55]. The ghost cell immersed boundary method (GCIBM)[6] is used in REEF3D and was developed by Berthelsen and Faltinsen[6] in 2008[68]. Here, fictitious ghost cells are updated by extrapolation near the solid boundary to update the solution, and the boundary conditions are then enforced implicitly instead of accounting for them explicitly[6][12][68]. An illustration of the complications with sharp geometry and the ghost cell approach is shown in Fig. (2.3).

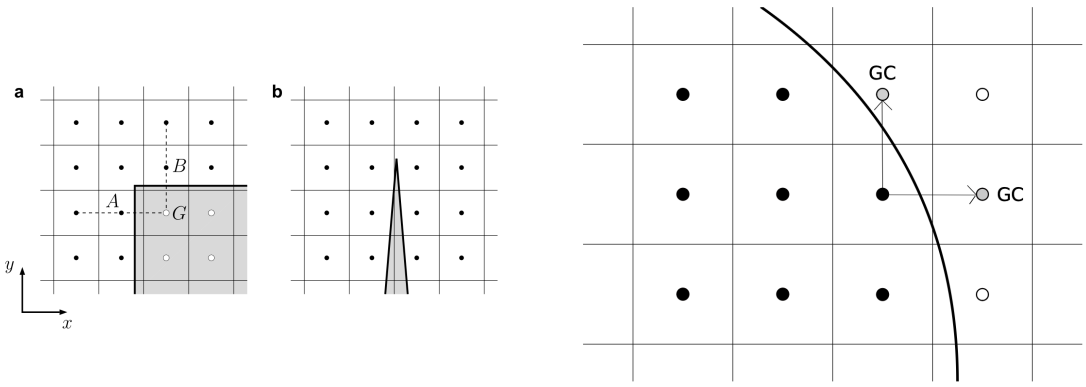


Figure 2.3: Pictures of the difficulties with irregularly shaped geometries and the ghost cells approach[9].

REEF3D can generate grids of several types[12]. For the easiest geometries, it can generate grids based on geometric primitives, like cylinders, wedges, and boxes[12]. With more complex geometry, a strategy presented in [131] is followed, where a .STL format can be read and immersed into the Cartesian grid[12].

2.8 Numerical Wave Tank

A numerical wave tank (NWT) is used to simulate waves, built up by a Cartesian grid to make an effortless definition of the geometry[12][68]. This tank is an imitation of a physical laboratory wave flume[68]. The purpose is to use the numerical wave tank to simulate waves that can mimic physical experiments from the wave flume[68][35]. A Numerical wave tank can be a good alternative to experiments as the CFD advances and computational power have increased[35][68]. With a validated CFD code, the costs, space, and time can be reduced compared to a laboratory experiment[12][27][68].

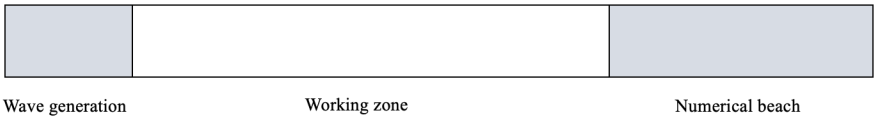


Figure 2.4: The numerical wave tank with zones.

The numerical wave tank comprises three zones: wave generation, a working zone (the wave tank)[9][68], and a numerical beach as illustrated in Fig. 2.4. The wave generation zone is of a length of one wavelength and will gradually generate the waves up to the input parameters before it goes over to the working zone[12]. In the working zone, the waves propagate over the defined length[12]. At the end of the tank, the numerical beach is placed with a length of two wavelengths[12]. Here the waves will gradually dissipate so that no wave reflections will occur in the tank[12][68][58].

2.8.1 Boundary treatment

Boundary conditions are needed to solve the partial differential equations with a unique solution[35]. Several types of boundary conditions are available. It can be treated as a solid boundary, inflow, outflow, wave generation, numerical or active beach[35][68]. The inflow and outflow can be a constant or hydrograph; if the boundary is solid, it is characterized as a wall[68]. There are also some conditions just for the flow velocity, where the two main types are: slip and non-slip[68]. These conditions are also illustrated in Fig. 2.5.

The non-slip conditions have zero velocity at the wall, so the fluid and boundary will have the same velocity[35][68]. For the non-slip condition, a relative moment exists between the fluid and the boundary. It then arises a discontinuity of the velocity function that provides a slip[88]. For this thesis, the non-slip condition is used.

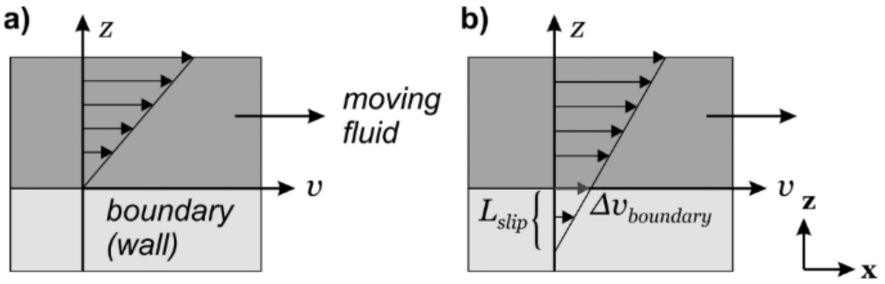


Figure 2.5: Illustration of the non-slip and slip conditions of the velocity flow[88]. a) is non-slip and b) is slip

In many cases, the boundary is not totally smooth but has some roughness[27]. Therefore, a roughness value and wall law should also be counted for this in the input[68]. Schlichting's formula[96] is used for the wall roughness and is calculated as shown below:

$$u^+ = \frac{1}{\kappa} \ln\left(\frac{30y}{k_s}\right) \tag{2.31}$$

where u^+ is the dimensionless wall velocity, κ is the Kàrmàn constant which is set to 0.4 based on experiments, y is the water depth, and k_s is the equivalent sand roughness. k_s is also the value used in REEF3D for the wall roughness[68]. The turbulent production near the wall is assumed to be the same as the dissipation of k[68]. Then the specific turbulent dissipation w wall function for a bed cell is obtained with the distance Δy_p from the wall to the center of the cell[68]:

$$\epsilon_{wall} = -\frac{c_\mu^{\frac{3}{4}} k_w^{\frac{1}{2}} U_w^+}{\Delta y_p} \tag{2.32}$$

Here, the value for the turbulent dissipation ϵ is given directly[68]. To treat the turbulent kinetic energy k at the wall, an integrating of the source terms of Eq. (2.18) over the bed cell must be done[68]:

$$\int (P_k - \epsilon_{wall})\rho = \left[\frac{\tau_w u_w}{\Delta y_p} - -\frac{c_\mu^{\frac{3}{4}} k_w^{\frac{1}{2}} U_w^+}{\Delta y_p}\right] \tag{2.33}$$

The shear stress τ_w and the dimensionless u^+ can then be determined from the rough wall law[68]. The transport equation for k discretizes the terms in Eq. (2.33) as source terms[68].

2.8.2 Wave generation and absoption

The numerical wave tank consists of three zones, a wave generator, a working zone, and a numerical beach[12]. The waves are generated in the wave generator zone, which is also the first zone of the tank. Further, the waves move to the next zone, a working zone or the wave tank, where the problem is solved, and the waves propagate[12][68]. Finally, a numerical beach will gradually dissipate the waves for the outlet and the last zone[12][68].

The wave generator can be of Dirichlet type or relaxation methods[12][68]. With a Dirichlet type, the generated waves are in constant motion, and the flow motion will periodically change, leading to waves of less quality[12]. In REEF3D, the relaxation method is used to generate waves and for wave dissipation[61], presented in [72] and extended for CFD models in [50][12]. The velocities and the free surface will then be ramped up over one wavelength to the values obtained from wave theory[12]. As a result, there will not be any disturbance in the interface, and reflected waves are absorbed, leading to higher-quality waves[12].

At the outlet, one crucial factor is to avoid reflecting waves that can be negative for the numerical results[12]. A relaxation method is a good option to avoid this and ensure effective wave energy absorption in the numerical beach[12][55]. It must be of two wavelengths, which, together with the length requirement of the relaxation method in the generation zone, can be a disadvantage due to the need for an extension of the numerical wave tank[68]. With the use of this method in the numerical beach relaxation zone, a smooth reduction of the computational values for the horizontal and vertical velocities to zero can be made[12]. Also, the free surface can be reduced smoothly to the still water level and the pressure to hydrostatic distribution for the still water level[12].

$$\begin{aligned} u(\tilde{x}) &= \Gamma(\tilde{x})u_{analytical} + (1-\Gamma(\tilde{x}))u_{computational} \\ w(\tilde{x}) &= \Gamma(\tilde{x})w_{analytical} + (1-\Gamma(\tilde{x}))w_{computational} \\ p(\tilde{x}) &= \Gamma(\tilde{x})p_{analytical} + (1-\Gamma(\tilde{x}))p_{computational} \\ \phi(\tilde{x}) &= \Gamma(\tilde{x})\phi_{analytical} + (1-\Gamma(\tilde{x}))\phi_{computational} \end{aligned} \tag{2.34}$$

The location of the relaxation zone is a variable of in $\Gamma(\tilde{x})$ and will transform the values in the generation zone and the numerical beach[68]. In the generation zone, the analytical values are transformed into computational ones, and in the numerical beach, the computational values are transformed into the analytical ones, dissipating the waves[68].

An active absorption method can be employed for the numerical beach zone to avoid the need for an extension of the tank[12][68]. This method will cancel out the incoming wave by producing opposite waves[68]. The velocity needed to match and absorb the waves can be obtained using linear long-wave theory[95][53].

$$U(t) = -\sqrt{\frac{g}{h}}\xi(t) \tag{2.35}$$

where $\xi(t)$ represent the free surface elevation from:

$$\xi(t) = \eta(t) - d \tag{2.36}$$

where $\eta(t)$ is the actual free surface location along the downstream boundary, and d is the water depth.

2.8.3 Wave theories

In order to understand the impact of the ocean on structures, an understanding of waves and their properties is essential[55][34]. Waves are made up of wind or gravity, and the water depth and slope of the seabed will influence the shape and size[68][44]. By using wave theories, it is possible to get a simplified representation of the waves and obtain their characteristic properties[68]. These theories are divided into different categories of linear and non-linear[55][44]. This thesis uses the non-linear theory of second-order Stokes wave theory. Still, a short introduction to linear wave theory will be presented as the Stokes theory is based on this.

2.8.4 Linear wave theory

Linear wave theory is also often known as Airy wave theory[2][45]. This theory describes linearized gravity wave propagation and implies that the wave is unaffected by each other while moving across the water's surface[45]. The linear wave theory is also characterized as waves with small amplitudes and can not be used on waves other than those called the small-amplitude approximation[45]. The horizontal and vertical velocity of an arbitrary depth given by Krogstad and Arntsen [59] are:

$$u(x,z,t) = aq\frac{cosh(k(d+z))}{sinh(kd)}sin(wt-kx) \tag{2.37}$$

$$w(x,z,t) = aw\frac{cosh(k(d+z))}{sinh(kd)}cos(wt-kx) \tag{2.38}$$

Here, a is the wave amplitude, $w = \frac{2\pi}{T}$ is the observed wave angular frequency, $k = \frac{2\pi}{\lambda}$ is wave number with λ wavelength, d is the waterdepth, t is time and x is the direction of propagation[12][68].

2.8.5 Non-linear Wave Theories

To get a complete statistical and physical description of the spectrum when the water depth becomes shallow or when there are steep waves, a non-linear wave theory must be used[45]. This is due to the limits of the linear theory that entails that this theory will no longer be valid under these conditions[45].

Second-order Stokes theory

In 1847, the second-order Stokes theory was developed by Stokes[104][45]. This theory improves the linear wave theory by adding an extra harmonic wave to the fundamental harmonic[45]. The wave velocities of the second-order Stoke theory after Dean and Dalrymple[26] are:

$$\begin{aligned} u = -\frac{\partial \rho}{\partial x} = \frac{H}{2}\frac{gk}{w}\frac{cosh(k(d+z))}{cosh(kd)}cos(kx - wt) \\ + \frac{3}{16}\frac{H^2 wk cosh(2k(d+z))}{sinh^4(kd)}cos(2(kx - wt)) \end{aligned} \tag{2.39}$$

$$\begin{aligned} w = -\frac{\partial \rho}{\partial z} = \frac{H}{2}\frac{gk}{w}\frac{sinh(k(d+z))}{cosh(kd)}cos(kx - wt) \\ + \frac{3}{16}\frac{H^2 wk sinh(2k(d+z))}{sinh^4(kd)}sin(2(kx - wt)) \end{aligned} \tag{2.40}$$

Compared to the linear wave, the second-order Stokes wave has a sharper crest and flatter trough, resulting in a higher velocity at the crest and a reduced velocity at the trough[45][68]. This change in velocity will produce a drift, pointed out by Stokes, and named Stokes drift[104]:

$$\bar{u} = \frac{wka^2 cosh(2k(d+z))}{2sinh^2(kd)} \tag{2.41}$$

2.9 Free surface treatment

The free surface is modeled by the Level Set Method, which uses a signed distance function $\phi(\vec{x}, t)$ called the level set function to capture the free surface[68]. The output from this function gives the closest distance to the interface Γ, and the zero level set is the location of the free water surface[12][55]. A change in the sign of the level set function distinguishes the two phases of water and air[75], which results in the following[83]:

$$\phi(\vec{x}, t) \begin{cases} > \ 0 \ if \ \vec{x} \ \epsilon \ phase \ 1 \\ = \ 0 \ if \ \vec{x} \ \epsilon \ \Gamma \\ < \ 0 \ if \ \vec{x} \ \epsilon \ phase \ 2 \end{cases} \tag{2.42}$$

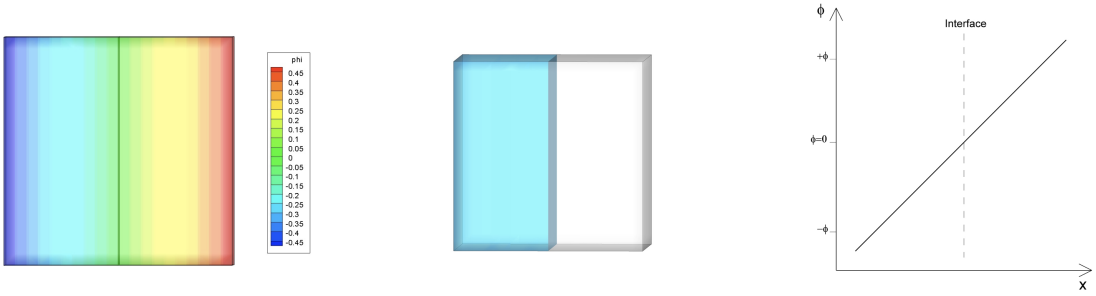


Figure 2.6: Illustration of the signed distance function, with the distance from the interface at zero level set, and an illustration of just the interface and the two phases [9].

The level set function is obtained from a convection equation when the interface is moving[68].

$$\frac{\partial \phi}{\partial t} + u_j \frac{\partial \phi}{\partial x_j} = 0 \tag{2.43}$$

After each time step, the level set function is reinitialized to ensure that the signed distance property holds when the interface starts to move, which is based on a PDF equation[83][68]:

$$\frac{\partial \phi}{\partial t} + S(\phi) \left(\left| \frac{\partial \phi}{\partial x_j} \right| - 1 \right) = 0 \tag{2.44}$$

Here, the $S(\phi)$ is the smoothed signed distance function[83]:

$$S(\phi) = \frac{\phi}{\sqrt{\phi^2 + |\frac{\partial \phi}{\partial x_j}|^2 \Delta x^2}} \tag{2.45}$$

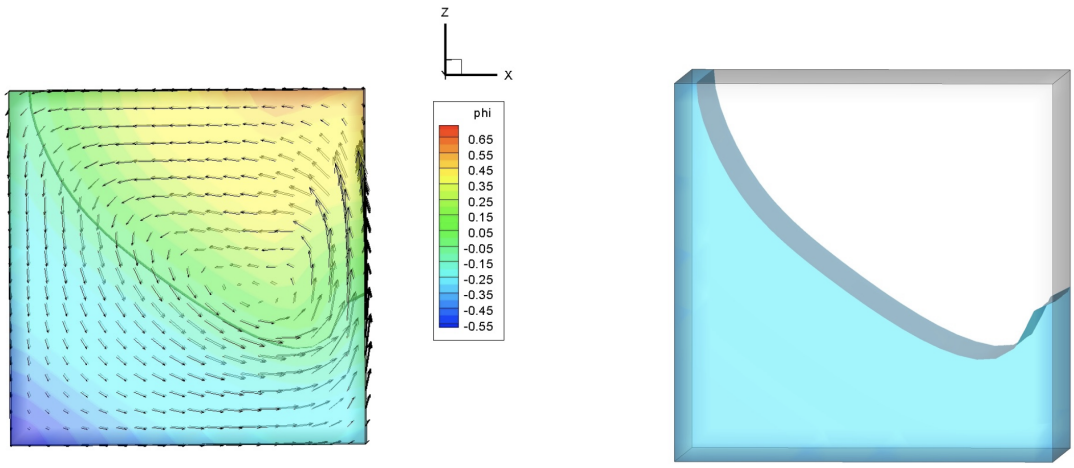


Figure 2.7: The coupling with the flow solver when the interface starts moving[9].

After the reinitialization, the density and viscosity are calculated[70].

$$\begin{aligned} \rho(\Phi) &= \rho_w H(\Phi) + \rho_a(1 - H(\Phi)), \\ \nu(\Phi) &= \nu_w H(\Phi) + \nu_a(1 - H(\Phi)), \end{aligned} \tag{2.46}$$

where w indicating water and a indicating air. H is the smoothed Heaviside step function, which is defined as:

$$H(\Phi) \begin{cases} 0 & if\ \Phi < -\epsilon \\ \frac{1}{2}\left(1 + \frac{\Phi}{\epsilon} + \frac{1}{\pi} sin(\frac{\pi\Phi}{\epsilon})\right) & if\ \Phi \leq \epsilon \\ 1 & if\ \Phi > \epsilon \end{cases} \tag{2.47}$$

Close to the interface, there is a jump in the values, which leads to numerical instability[55]. The smoothed Heaviside step function calculates the density at the cell face of a transition zone, ϵ is defined as $2.1\Delta x$ that smoothes out these properties to solve this problem, where Δx is the length between the grid points in the vicinity of each evaluation point[70][12]. For the fluid-structure interactions, an incremental pressure-correction algorithm[112] is used. This algorithm also solves the continuity Eq. (2.1) together with the momentum Eq. (2.3), where the previous step of the pressure gradients from the momentum equation in each k-th Runge-Kutta sub-step predicts the velocity field[70]:

$$\frac{u^{(*)} - \beta_k u^{(n)}}{\alpha_k \Delta t} = \frac{1}{\Delta t} u^{(k-1)} - u^{(k-1)} \cdot \nabla u^{(k-1)} - \frac{\nabla p^{(k-1)}}{\rho} + \frac{1}{\alpha_k} \nabla \cdot (\nu[\nabla u + \nabla u^T])^{(*)} + g, \tag{2.48}$$

with $\alpha_k = 1.0, 1/4, 2/3$, $\beta_k = 0.0, 3/4, 1/3$ and $k = 1, 2, 3$
Then the Poisson equation can be solved for the pressure correction term p_{corr}

$$\nabla \left(\frac{1}{\rho} \nabla p_{corr} \right) = \frac{1}{\alpha_k \Delta t} \nabla \cdot u^{(*)}, \tag{2.49}$$

Finally, the pressure and divergence-free velocity fields are calculated:

$$\begin{aligned} p^{(k)} &= p^{(k-1)} + p_{corr} - \rho \nu \nabla^{(*)}, \\ u(k) &= u^{(*)} - \frac{\alpha_k \Delta t}{\rho} \nabla p^{(k)}. \end{aligned} \tag{2.50}$$

2.10 6 DOF algorithm

A continuous, direct forcing immersed boundary method is used for the floating body in the fluid solver and is presented in [70], which also is followed in the further summary presentation of the 6 DOF algorithm below. First, a STL geometry representing the floating body, made by many non-connected triangles, is transferred to the solver[70]. Further, it is necessary to represent the geometry in the Eulerian fluid domain. This representation is possible by creating a signed distance field from the STL geometry, where a ray casting algorithm[11] ensures that the inside-outside information close to the object and Sussman et al.[106] reinitialization algorithm is obtained[70]. The generated level set function Φ_s is then utilized to differentiate the fluid and solid[70]. For this, an extension of the Eq. (2.46) is done:

$$\begin{aligned} \rho(\Phi, \Phi_s) &= \rho_s H(\Phi_s) + (1 - H(\Phi_s)) \cdot (\rho_w H(\Phi) + \rho_a (1 - H(\Phi))), \\ \nu(\Phi, \Phi_s) &= (1 - H(\Phi_s) \cdot (\nu_w H(\Phi) + \nu_a (1 - H(\Phi))). \end{aligned} \tag{2.51}$$

Examining the conservation law, a guarantee of the continuity of the forcing field can be completed[70].

$$\begin{aligned} \Delta \cdot u &= 0, \\ \frac{\partial u}{\partial t} + u \cdot \Delta u &= -\frac{1}{\rho} \Delta \rho + g + f, \end{aligned} \tag{2.52}$$

with the forcing term

$$f = \frac{\partial P(u)}{\partial t} + P(u) + \frac{1}{\rho} \Delta \rho - g, \tag{2.53}$$

Where $P(u)$ is the operator predicting the velocity field into an incompressible rigid body velocity field[70].

With Eq. (2.51) and the additional term f, equations of a single set throughout the entire domain can be determined, ensuring that the three phases' material properties are distinguished and that the interface boundary conditions are exact[70].
A new discrete time step prediction $n+1$, f is found to be[70]:

$$f^{n+1} = H(\Phi_s^{(n+1)}) \cdot \left(\frac{P(u^{(n+1)} - P(u^{(n)})}{\Delta t} + P(u^{(n)}) \cdot \Delta P(u^{(n)}) + \frac{1}{\rho} \Delta p^{(n+1)} - g \right). \tag{2.54}$$

A satisfactory approximation of $P(u^u) = u^{(n)}$ is made to find the unknown velocity at the new time step and avoid implicit calculations that are very expensive[70]. Further, an approximation for the pressure is obtained from the earlier time step[70]. So, f at the new time step becomes:

$$f^{n+1} = H(\Phi_s^{(n+1)}) \cdot \left(\frac{P(u^{(n+1)} - u^{(n)}}{\Delta t} + u^{(n)} \cdot \Delta u^{(n)} + \frac{1}{\rho} \Delta p^{(n)} - g \right). \tag{2.55}$$

By evaluating this with Eq. (2.48), $f^{(n+1)}$ can be represented as:

$$f^{(n+1)} = H(\Phi_s^{(n+1)}) \cdot \left(\frac{P(u^{(n+1)}) - u^{(*)}}{\Delta t} \right). \tag{2.56}$$

The updated velocity field can be represented as $u^{(*)}$, which entails that the predictor step in Eq. (2.48) is first carried out without the forcing term $f^{(*)}$. Before the Poisson equation can be determined, this forcing term must be calculated and included in the estimated velocity field[70]. The calculation of $f^{(*)}$ is done from:

$$f^{(*)} = H(\Phi_s^{(*)}) \cdot \left(\frac{P(u^{(*)}) - u^{(*)}}{\alpha_k \Delta t} \right), \tag{2.57}$$

Calculation of the rigid body velocity field

The rigid body's position after moving x_i and rotational motion must be described to determine its velocity field[70]. The motion position x_i is accounted for in the inertia system. A coordinate system attached to the body describes the rotational motion, where the Euler parameter vector $e = (e_0, e_1, e_2, e_3)^T$ is utilized, with $e^T e = 1$. A more appropriate way to describe these parameters is with the Tait-Bryan physical angles ϕ, θ, ψ, and their relation are given as[85]:

$$
\begin{aligned}
e_0 &= c(\frac{\phi}{2}) \cdot c(\frac{\theta}{2}) \cdot c(\frac{\psi}{2}) + s(\frac{\phi}{2}) \cdot s(\frac{\theta}{2}) \cdot s(\frac{\psi}{2}),\\
e_1 &= s(\frac{\phi}{2}) \cdot c(\frac{\theta}{2}) \cdot c(\frac{\psi}{2}) - c(\frac{\phi}{2}) \cdot s(\frac{\theta}{2}) \cdot s(\frac{\psi}{2}),\\
e_2 &= c(\frac{\phi}{2}) \cdot s(\frac{\theta}{2}) \cdot c(\frac{\psi}{2}) + s(\frac{\phi}{2}) \cdot s(\frac{\theta}{2}) \cdot s(\frac{\psi}{2}),\\
e_3 &= c(\frac{\phi}{2}) \cdot c(\frac{\theta}{2}) \cdot s(\frac{\psi}{2}) - s(\frac{\phi}{2}) \cdot s(\frac{\theta}{2}) \cdot s(\frac{\psi}{2}),
\end{aligned}
\tag{2.58}
$$

Here, c is *cos*, and s is *sin*.
The definition of the back-transformations are[70]:

$$
\begin{aligned}
&\psi = arctan2(2 \cdot (e_1 \cdot e_2 + e_3 \cdot e_0), 1 - 2 \cdot (e_2 \cdot e_2 + e_3 \cdot e_3)),\\
&\theta = arcsin(2 \cdot (e_0 \cdot e_2 - e_1 \cdot e_3)),\\
&\phi = arctan2(2 \cdot (e_2 \cdot e_3 + e_1 \cdot e_0), 1 - 2 \cdot (e_1 \cdot e_1 + e_2 \cdot e_2)).
\end{aligned}
\tag{2.59}
$$

An orthogonal rotation matrix presents the vector transformation in the body-fixed coordinate system to a comparable vector within the inertial system[70].

$$
\mathbf{R} = 2 \begin{bmatrix}
\frac{e_0^2+e_1^2-e_2^2-e_3^2}{2} & e_1e_2 - e_0e_3 & e_0e_2 - e_1e_3\\
e_0e_3 + e_1e_2 & \frac{e_0^2+e_1^2-e_2^2-e_3^2}{2} & e_2e_3 - e_0e_1\\
e_1e_3 - e_0e_2 & e_0e_1 + e_2e_3 & \frac{e_0^2+e_1^2-e_2^2-e_3^2}{2}
\end{bmatrix}
\tag{2.60}
$$

In relation to the Euler parameters, the body's rotational kinematic equations are provided as[70]:

$$
\dot{e} = \frac{1}{2} G^T w,
\tag{2.61}
$$

This equation consists of the factors of the angular velocity ω in the body-fixed coordinate system and G, which is defined as:

$$
\mathbf{G} = \begin{bmatrix}
-e_1 & e_0 & e_3 & -e_2\\
-e_2 & -e_3 & e_0 & e_1\\
-e_3 & e_2 & -e_1 & e_0
\end{bmatrix}
\tag{2.62}
$$

With the momentum vector $h = Iw$, where I is the moment of inertia tensor, Eq. (2.61) can be defined as:

$$
\dot{e} = \frac{1}{2} G^T I^{-1} h,
\tag{2.63}
$$

A first-order ODE for $\dot{h}$ is then calculated by applying the development of Shivarama and Fahrenthold[97] and a system Hamiltonian[70].

$$\dot{h} = -2G\dot{G}^T h + M_b \tag{2.64}$$

with assuming that M_b are presenting imposed moments in the body-fixed system, together with a zero-assumption for the potential energy function, an automatical fulfillment of the Euler parameters is provided[70].

On the triangulated surface of N triangles, the body forces and momenta are calculated by integration of the fluid properties over the structure surface Ω together with a trilinear interpolation[70][27]:

$$\begin{aligned} F_i &= \int_\Omega (-np + \rho\nu n\tau) d\Omega(x) = \sum_{i=1}^{N} (-np + \rho\nu n\tau)_i \cdot \Delta\Omega_i, \\ M_i &= \int_\Omega r \times (-np + \rho\nu n\tau) d\Omega(x) = \sum_{i=1}^{N} r_i \times (-np + \rho\nu n\tau)_i \cdot \Delta\Omega_i, \end{aligned} \tag{2.65}$$

where, n is the surface normal vector, τ is the viscous stress tensor and r represent the distance vector to the centre of gravity[70]. The projection can first be calculated after the transformation matrix in Eq. (2.60) has transferred the moments to the body-fixed coordinate system and the body velocities are found[70]. Then the calculation can be carried out from:

$$P(u^{(*)}) = \dot{x_i} + \omega_i \times r, \tag{2.66}$$

here, $\dot{x_i}$ is the translational, and ω_i is the rotational rigid body velocity vector in the inertial reference frame[70].

2.11 Mooring model

The mooring forces are found in the mooring dynamic solver, following the work presented in [70]. The overview and explanation of the mooring model below are also following this study [70]. First, the bending stiffness is neglected[78] in the dynamics of a mooring line explained as:

$$\gamma \frac{\partial^2 r}{\partial t^2} = \frac{\partial F_T f}{\partial s} + F_e, \tag{2.67}$$

here, γ is the material-specific weight, r is the line coordinates in the inertial system, F_T is the tension force magnitude where f pointing the direction of this force as a unit vector, s is the coordinate along the line and F_e is the external force, which

includes the effect of hydrodynamic (F_H) and gravitation (F_G)[70]. A simplification of the Eq. (2.67) can be done by assuming that the line motion in time is small and the fluid is of steady-state flow, then the equation can be presented as[70]:

$$\frac{\partial F_T f}{\partial s} = -F_e \tag{2.68}$$

Each mooring line has to be divided into N bars with the same distance of length l_t and knots P in between to determine this force equilibrium[27][70]. This discretization is shown in Fig. 2.8, also illustrating the connection to the bottom by the first knot $P^{(0)}$ and to the floater by the last knot $P^{(N)}$.

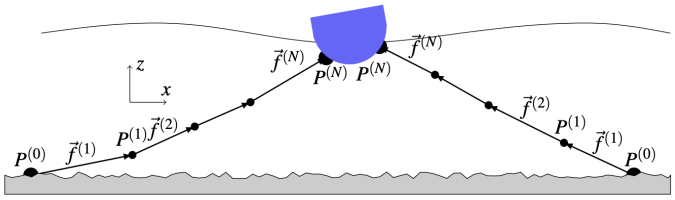


Figure 2.8: Discretization of mooring lines[70]

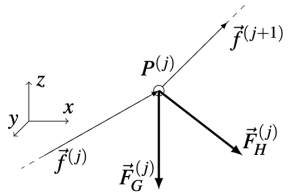


Figure 2.9: Force equilibrium [70]

The contribution of the gravity force at any knot $P^{(j)}$ is given as:

$$F_G^j = \gamma g \cdot \frac{\rho_m - \rho}{\rho} \cdot \frac{l_t^{(j)} + l_t^{(j+1)}}{2}, j = 1, ..., N-1, \tag{2.69}$$

Morison's formula[73] is used to calculate the hydrodynamic forces F_H as drag forces at each bar[70].

$$F_H^{(j)} = l_t^{(j)} d_t^{(j)} \frac{\rho}{2} \cdot [c_t(u \cdot f)|u \cdot f| \cdot f + c_n(u - (u \cdot f)f|u - (u \cdot f)f|]^{(j)}, j = 1, ..., N, \tag{2.70}$$

Then, the force equilibrium can finally be solved at each inner knot $P^{(j)}$ with a discretized expression[27][70], as shown in Fig. 2.9.

$$f^{(j+1)} F_T^{(j+1)} - f^{(j)} F_T^{(j)} = -\left(F_H^{(j)} + F_G^{(j)}\right), j = 1, ..., N-1. \tag{2.71}$$

In this thesis, a simplified mooring system is used to hold the floating object in place[7]. It is a horizontal mooring system based on the spring Eq. (2.72).

$$F = F_0 + k \cdot \delta l, \tag{2.72}$$

For the application in the simulation, a comparison must be made with the experiment to match the force-displacement of the actual mooring, which then determines the stiffness of the spring[7].

2.12 Parallelization

Domain decomposition is how REEF3D achieves parallelization, which splits the simulation domain into smaller parts that use ghost cells, implemented by the message-passing interface (MPI), to communicate and exchange values[12]. This parallelization process improves the code performance because it can simultaneously solve the smaller problems instead of just solving one large problem[12][55]. Fig. 2.10 shows how this parallelization can be done.

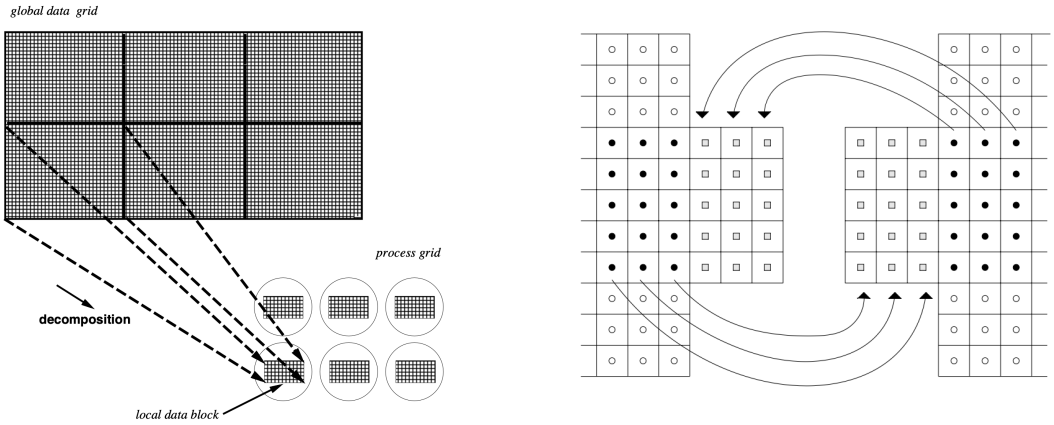


Figure 2.10: Illustration of how the domain is decomposite and the exchange of values using ghost cells[9].

Chapter 3

Verification and Validation of the Numerical Tank

In this chapter, the verification and validation of two different 2D cases are presented and discussed to ensure the accuracy and reliability of further simulations. The 6DOFs algorithm will first be verified and validated by a simulation of a heaving cylinder, considering the damping heave motion. Then, a free-floating box in waves will be examined and discussed, looking at the motions of the box; heave, roll, and surge in different wave conditions[13].

3.1 Heaving cylinder

A cylinder is simulated considering only the vertical motion and then compared to experimental data from Ito [49]. Another study of this case is also done in [13], where numerical simulations are performed by Yang[130]. In this study, the numerical setup and simulation are followed by Ito [49] and [13], leading to a domain of a numerical wave tank with an initial water depth of $d = 1.22m$ and water density $\rho = 1000kg/m^3$. The tank's geometry has a length of $10m$ and a height of $1.8m$, also shown in Fig. 3.1.

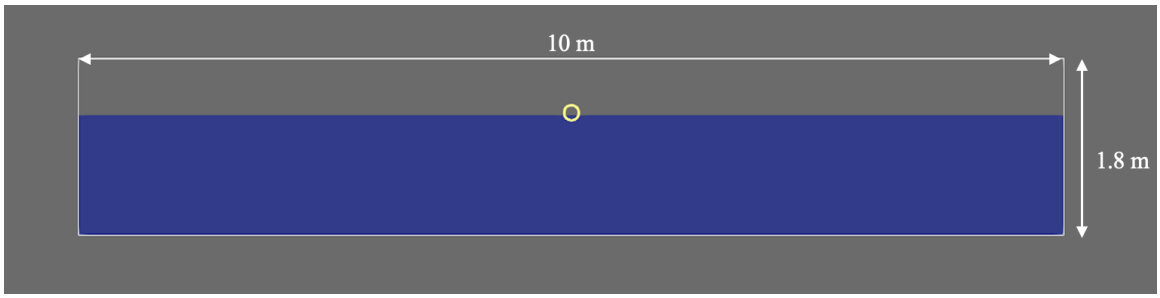


Figure 3.1: The domain and dimensions of the numerical wave tank with the cylinder.

In the middle of the tank, partly submerged, the horizontally circular cylinder is placed, equal to Ito's[49], with a radius of $r = 0.0762m$, centroid position at $h_1 = 0.02454m$ above the free surface, and density $\rho = 500kg/m^3$. Three grid sizes, $dx = 0.025m$, $dx = 0.01m$, and $dx = 0.005m$, are used for investigating this case, where a free fall from the initial position makes the starting point of the simulation[13]. The cylinder is then attached to the water[13].

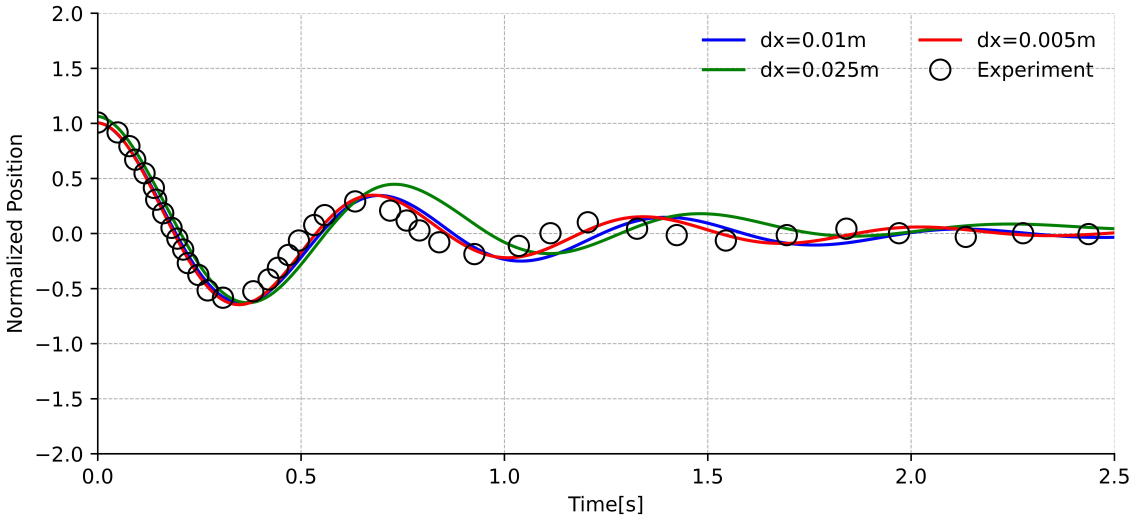
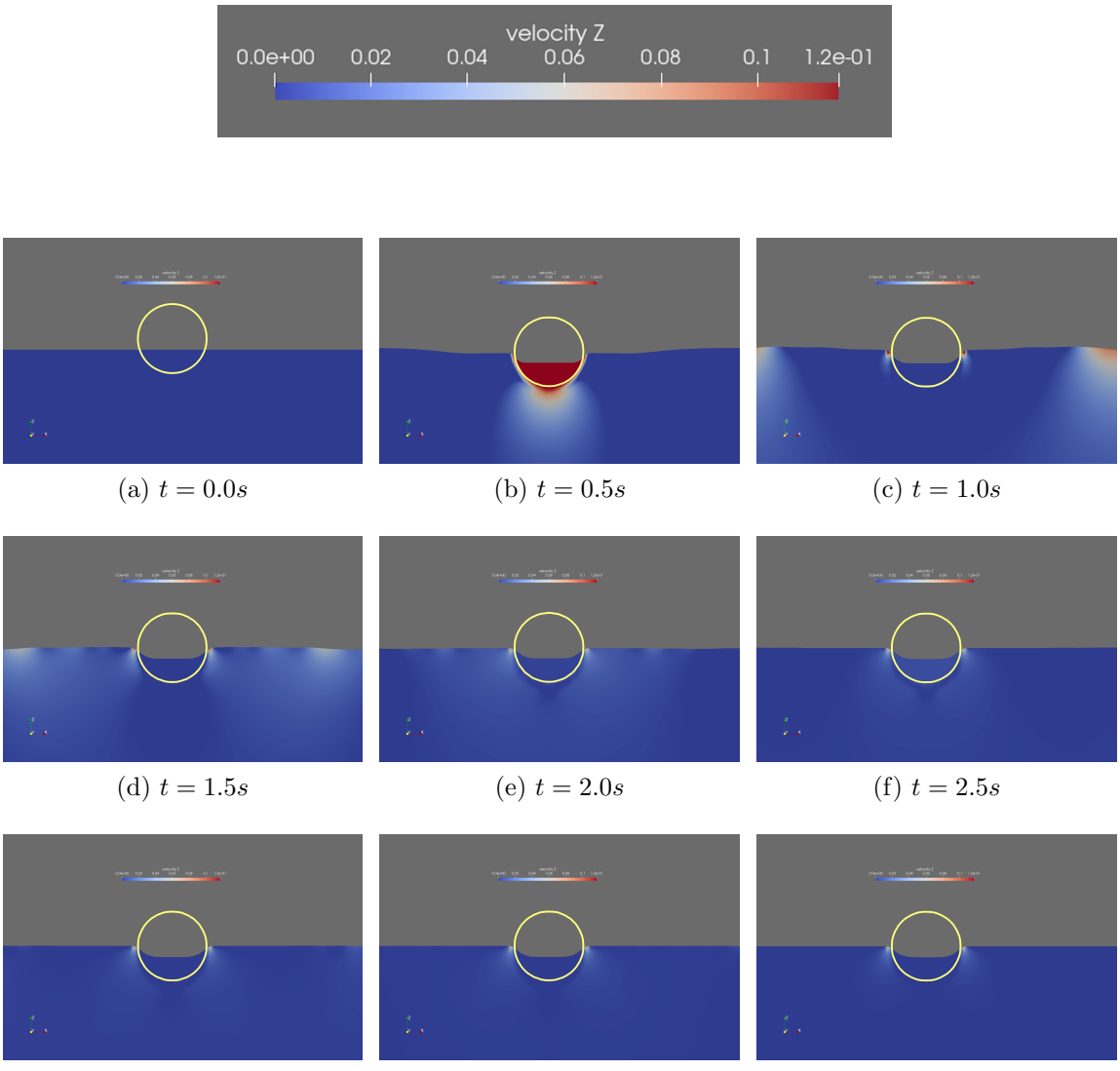


Figure 3.2: Cylinder heave motion compared to the experimental data using three different meshes.

The plot in Fig. 3.2 shows the results of the heave motion of three different grid sizes and the experimental data. The x-axis represents time, and the z-axis is a normalized position calculated from Eq. (3.1).

$$Normalization = \frac{Z - d}{h_i} \tag{3.1}$$

The grid size of $dx = 0.025m$ is the coarsest and the one with the most deviation, showing a good match for the first 0.5 seconds but tends to diverge after the first period[13]. The simulation has a better approximation for the grid sizes of $dx = 0.01m$ and $dx = 0.005m$ compared to the experimental data. However, as mentioned in [13], the reflection of the waves from the side walls makes the simulation more out of position after $t = 1.5s$. The grid sizes $dx = 0.01m$ and $dx = 0.005m$ are relatively similar, with $dx = 0.005m$ showing a slightly more precise approximation. Due to the increasing computational time using $dx = 0.005m$, the grid size of $dx = 0.01m$ should be used in further simulations[13].



(a) $t = 0.0s$ (b) $t = 0.5s$ (c) $t = 1.0s$

(d) $t = 1.5s$ (e) $t = 2.0s$ (f) $t = 2.5s$

(g) $t = 3.0s$ (h) $t = 3.5s$ (i) $t = 4.0s$

Figure 3.3: Illustration of the vertical motion and velocity of the cylinder entering the water.

Fig. 3.3 shows the vertical velocity and motion of the cylinder when it is released and enters the water at different timesteps of $t = 0.5s$. The cylinder gets its highest velocity in the first period of breaking the water at $t = 0.5s$, similar to the case of the heaving cylinder simulated in [13]. As the cylinder enters the water, it also creates waves at each side[13], starting to dissipate after approximately $t = 2.5s$. At the time step of $t = 3.0s$, these waves are totally spread out, and no motion is no longer observed around the cylinder. Since the cylinder is only released from its initial position, without an additional height, it will thus not have enough power to be immersed, which also is the case in [13].

3.2 Free Floating Box in Waves

A simulation of a floating box in 2D under wave conditions is done to validate the 6DOF algorithm, looking at the motion of heave, roll, and surge. These movements are similar to the degrees of freedom of rotation around the Y axis and movement along the X and Z axis[13]. The experiment from Dalian University by Ren et al.[89] is used for the dimensions of the box and the comparison data. The numerical setup follows the dimensions from the investigation in [13]. The box is rectangular with a height of $h = 0.20m$, length of $l = 0.30m$, and density of $\rho = 500kg/m^3$. The numerical tank used in the simulation is $1.5m$ high and $20m$ long, where the position of the floating box is $7.0m$ from in from the start of the tank, which is illustrated in Fig. 3.4. Also, in [71] and [68], a similar simulation of the floating box is performed to validate a new quasi-static mooring model and the 6DOF algorithm.

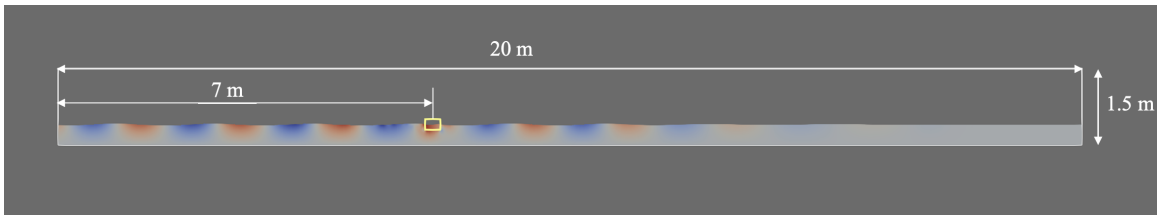


Figure 3.4: The spatial domain of the numerical wave tank with the box

For wave generation and absorption, the tank has three zones: a generation zone, a working zone, and a numerical beach, where the first zone is of one wavelength and the last of two wavelengths. In the simulation, the 2nd-order Stokes theory is used for the waves. These waves are generated in the generation zone. It often takes some time before the waves are stabilized[13]. Therefore, the comparison is made after this is achieved and when the box is in motion. The 2nd-order Stokes waves consist of a wave height of $H = 0.04m$, wave period $T = 1.2s$, a wavelength of $L = 1.936$, and a wavenumber of $k = 3.245m^{-1}$[13]. The total modeling time of the simulation is 20 seconds. All the variables from the results are normalized:

$$time = \frac{t}{T}$$

$$heave\ and\ surge = \frac{result - d}{d}$$

$$roll = \frac{\theta}{k \cdot d}$$

A grid convergence test is done with three different grid sizes and compared with the experimental data. These are $dx = 0.025m$, $dx = 0.01m$, and $dx = 0.005m$.

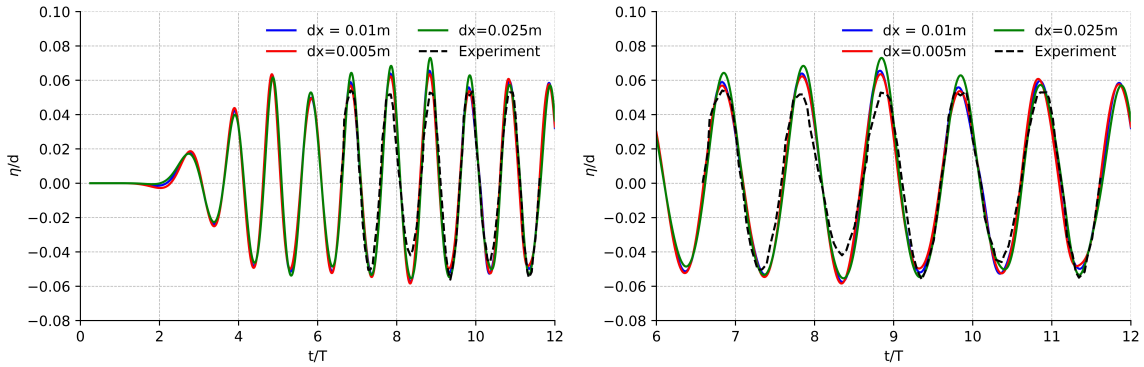


Figure 3.5: Grid convergence study of the wave elevation

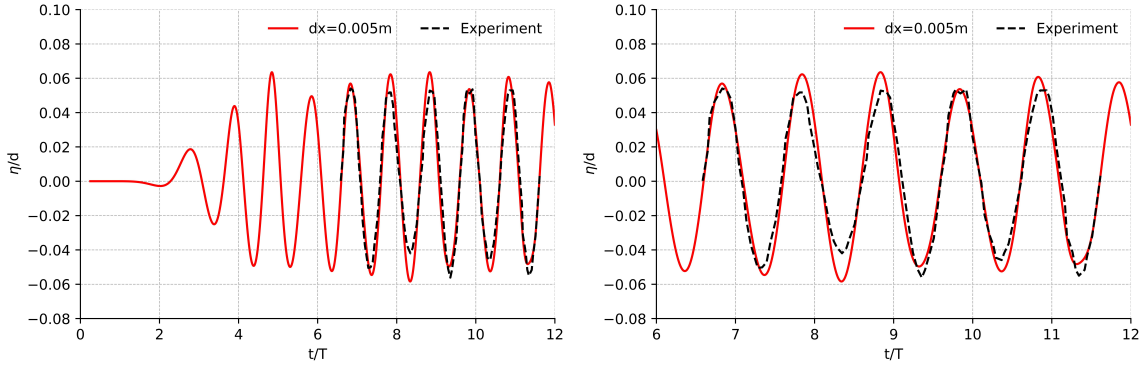
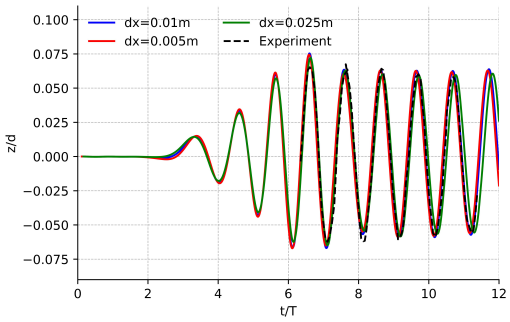


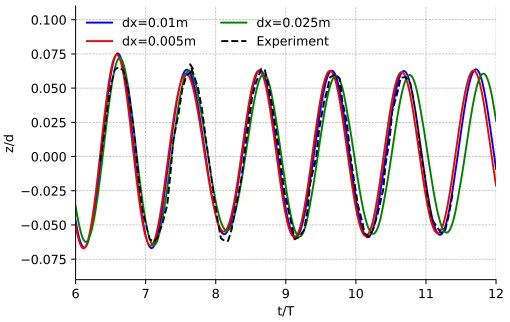
Figure 3.6: Wave elevation of dx=0.005m compared with experimental data

In Fig. 3.5, the wave elevation is plotted with the three grid sizes and compared to the experimental data. All of the grid sizes have a good match with the experimental data. However, the finer the grid size of $dx = 0.005m$ is a slightly better result, illustrated in Fig. 3.6. For all grid sizes, the crest has a minor overestimation through the simulation, with the most significant deviation at about $t/T = 8$. At this time, there is also an overestimation of the trough. Otherwise, throughout the simulation, the trough perfectly matches the experiment.

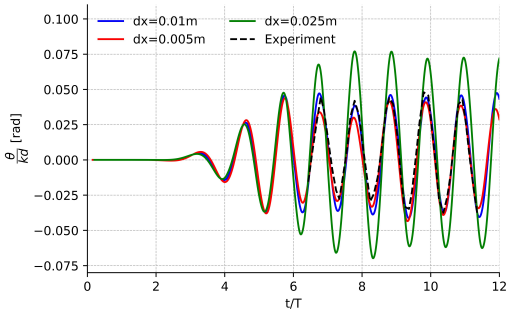
The motions of heave, roll, and surge are plotted against the experimental data in Fig. 3.7 with a grid convergence study with the grid sizes $dx = 0.025m$, $dx = 0.01m$, and $dx = 0.005m$.



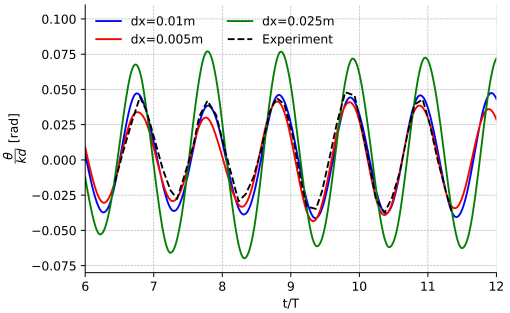
(a) Heave motion comparing three different grid sizes between $t/T = 0$ and $t/T = 12$



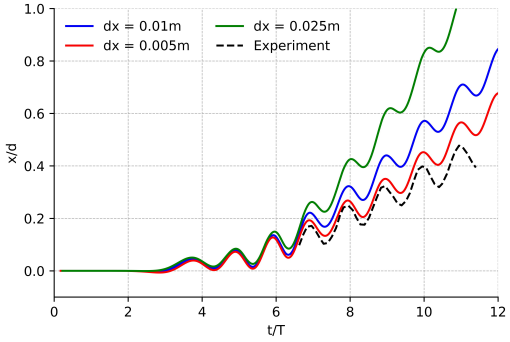
(b) Heave motion comparing three different grid sizes between $t/T = 6$ and $t/T = 12$



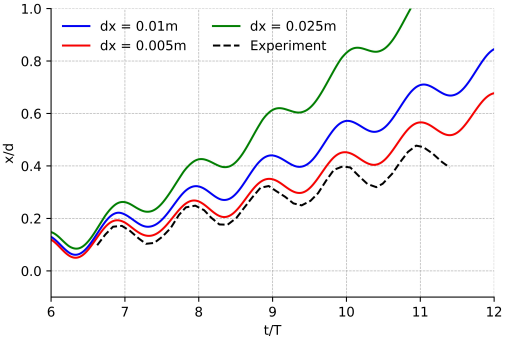
(c) Roll motion comparing three different grid sizes between $t/T = 0$ and $t/T = 12$



(d) Roll motion comparing three different grid sizes between $t/T = 6$ and $t/T = 12$



(e) Surge motion comparing three different grid sizes between $t/T = 0$ and $t/T = 12$



(f) Surge motion comparing three different grid sizes between $t/T = 6$ and $t/T = 12$

Figure 3.7: Grid convergence study of heave, roll and surge motion.

Fig. 3.7 a) and b) show the heave motion. This is in excellent agreement with the experimental data. All the grid sizes show approximately the same results, so this parameter is not decisive. The first crest has an overestimation but decreases for the following oscillations and gives a good match. The second trough is slightly underestimated, while the rest is the same value as the experiment.

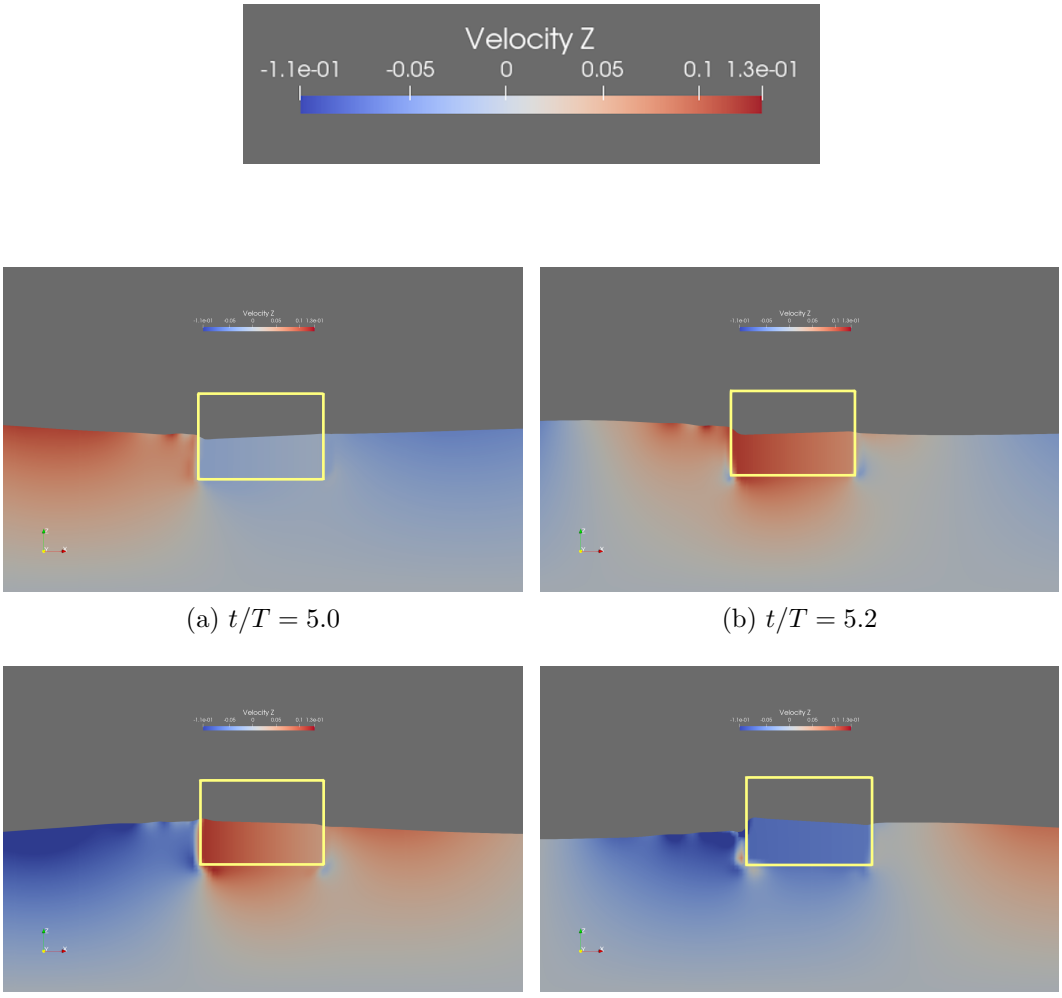
The roll motion is illustrated in Fig. 3.7 c) and d). Here, the simulation depends on the grid size, as they show different results with higher grid resolution. The coarsest grid size of $dx = 0.025m$ is nonfunctional, with an overestimation of each oscillation of the crest and trough. The grid sizes of $dx = 0.01m$ and $dx = 0.005m$ are in better agreement with the experimental data and show very similar results. For the grid size of $dx = 0.01m$, the result starts to better agree with the experimental data gradually over time. However, the first crests have an underestimation. After about $t/T = 8$, the crest decreases to the experimental crest and gives good results. The trough has a perfect match throughout the whole simulation. The grid size of $dx = 0.005m$ switches between overestimation and underestimation every other of the top. The bottom is slightly overestimated for the three first troughs, stabilizing with a good match with the experimental data. Overall, a very good match for $dx = 0.01m$ and $dx = 0.005m$.

In Fig. 3.7 e) and f), the surge motion is shown. The result from the grid sizes has very different results, where the coarsest of $dx = 0.005m$ has the most deviation from the experiment, while the grid size of $dx = 0.01m$ shows the best result. This indicates that the simulation is independent of the grid size. The motion of the surge shows that the box is moving through the numerical tank, which is obvious, as the box is not held in place. The grid size of $dx = 0.025m$ starts following the other grid sizes, but after about $t/T = 6m$, it has a sudden increase and continues to increase at a much higher rate than the others, which gives the floating box a more significant movement in the x-direction. The grid size of $dx = 0.01m$ and $dx = 0.005m$ is closer to the experimental data, but none match perfectly. Both start to deviate after the first oscillation, but $dx = 0.005m$ deviates with a higher slope than $dx = 0.01m$, resulting in a better match with the latter.

In response to unsatisfactory results obtained from the surge-motion simulation, a parameter exploration was undertaken to gain deeper insights into the factors influencing the surge-motion simulation and identify optimal parameter settings that could potentially enhance the overall outcomes. Particular attention was given to the direct forcing and the CFL number to investigate the 6DOF and the time treatment. The tangential velocity and the floating body density treatment were activated for direct forcing. The tangential velocity can be standard or reduced force, and the

floating body density can be either discontinuous solid or continuous fluid density. Four combinations of these were investigated to assess the impact comprehensively. However, none of these showed any change in the result.

Continuing the investigation, the CFL number was studied by modifying this number while the other parameters remained constant. For the original simulation, this number was 0.3. A change to lower and higher values was conducted but showed no significant improvement.



(a) $t/T = 5.0$ (b) $t/T = 5.2$

(c) $t/T = 5.4$ (d) $t/T = 5.6$

Figure 3.8: The motion of the floating box in waves, showing the vertical velocity in $t/T = 0.2$ intervals

Fig. 3.8 shows the motion of the floating box in waves from $t/T = 5.0$ to $t/T = 5.6$, with values of the wave's vertical velocity. This illustration also demonstrates how the changes in velocity vary during the wave formation and interaction with the box.

Chapter 4

Windmoor

In this chapter, a case study of a floating offshore wind turbine is done using SINTEF's INO WINDMOOR semisubmersible floating[102] wind turbine, where the purpose is to simulate the response to hydrodynamics loads and motion with REEF3D::CFD, which has also been explored in [7]. Furthermore, from the WINDMOOR project, there are carried out experimental data that is used for comparison with the numerical simulations for validation of the results[111]. Three regular waves are investigated with different steepness, wave height, and wave period, where the response motion for heave, pitch, and surge are studied.

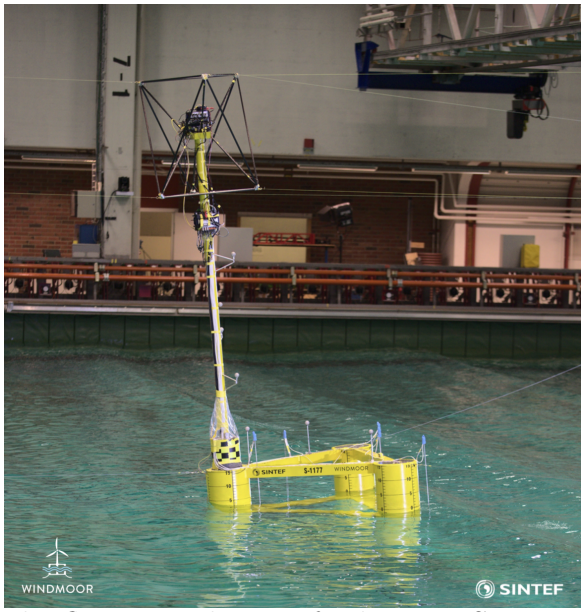
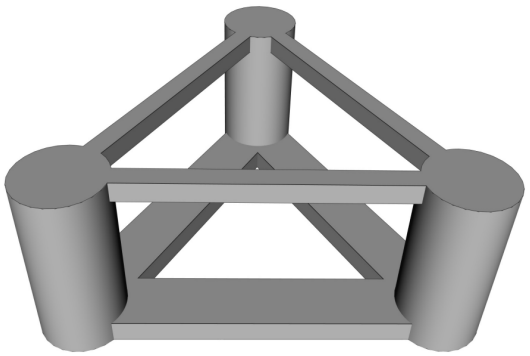


Figure 4.1: The floater in the test facilities at SINTEF Ocean[110]

4.1 Geometry

The geometry description of the INO WINDMOOR semisubmersible floating wind turbine follows the presentation in [7]. Three columns with pontoons and a deck beam in between create the platform's framework, forming it into a triangle[7]. On top of the triangular-shaped platform, there is a wind turbine tower. This tower is not included in the numerical simulation, so the simulation is performed with only the platform, similar to the study in [7]. Therefore, the loads on the tower and rotor from the wind are not considered. A dimension overview of the geometry is shown below in Tab. 4.1.



Parameter	Unit	Value
Column diameter	m	15.0
Column height	m	31.0
Pontoon width	m	10.0
Pontoon height	m	4.0
Column centre-centre distance	m	61.0
Deck-Beam width	m	3.5
Deck-Beam height	m	3.5
Draft	m	15.5

Table 4.1: An overview of the INO WINDMOOR platforms geometry and main dimensions[102][7]

4.2 Experimental Data

The experimental data were carried out by the WINDMOOR KPN project (NFR grant 294573) [111] in SINTEF Ocean's Ocean Basin on a 1:40 froude scale in 2020 [110]. The basin has dimensions of 80m by 50m with a full-scale water depth of 150m[7]. However, during the experiment, this water depth was 3.75m[7]. This thesis uses only the wave experimental data without aerodynamic loads to compare the numerical simulations.

The floater was held in place with mooring lines of a simplified horizontal system above the water, consisting of three wire-chain spring lines connected to the floater's columns and anchored at the end of the basin[7], as shown in Fig. 4.1. Based on as-built data[110], Tab. 4.2 gives the floating wind turbine's mass properties in full scale, where the still water level is the basis of the center of gravity (COG) location[7].

Parameter	Unit	Value
Mass	t	14124.0
r_{xx}	m	43.62
r_{yy}	m	44.01
r_{zz}	m	29.87
COG_x	m	0.00
COG_y	m	0.00
COG_z	m	3.94

Table 4.2: Mass properties of the floating platform, WINDMOOR[7]

4.3 Numerical Setup

The numerical wave tank

The numerical simulations for the regular waves are explored in a rectangular numerical wave tank, following the setup from [7], performed with a model scale (1:40). However, the results are scaled up and presented in a prototype scale[7]. For the numerical wave tank, a simplification was done to reduce the computational time by decreasing the dimensions compared to the experiment[7]. This resulted in a domain length of 18.2m and width of 7.6m with a height of 5.0m and a water depth of 3.75m. In addition, there are added wall boundaries and two wave relaxation zones for the inlet and outlet[7]. At the inlet, there is a wave generation zone of 3.0m, and at the outlet, there is a numerical beach that absorbs the waves at a distance of 6.0m[7]. The position of the floater is 9.078m from the inlet, shown in Fig. 4.2, together with the mooring lines and dimensions[7]. WAVE1 and WAVE2 from this figure represent where the experimental data are measured[7]. WAVE1 is experimental data carried out at the beginning of the tank, and WAVE2 is experimental data carried out at the position of the floating platform[7].

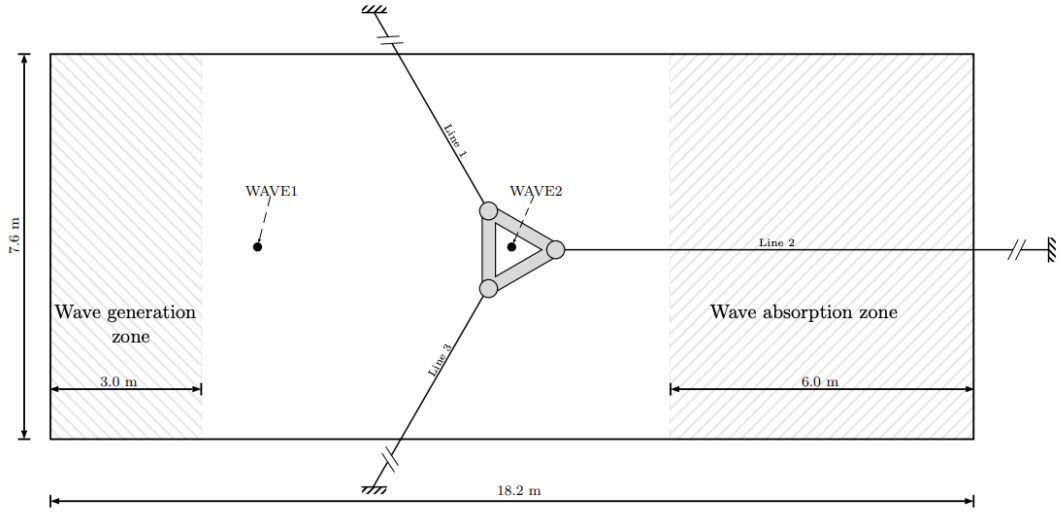


Figure 4.2: The numerical wave tank[7]

Wave gauges

In the numerical setup, three different wave gauges are placed to measure the numerical results at the same places as the experimental measurements[7]. The first wave gauge is placed at the inlet of the numerical wave tank, the same place as WAVE1, while wave gauges 2 and 3 are placed at the same place as WAVE2, at the floating

platform[7]. Wave gauge 2 has the position of $x = 9.09227m$ and $y = 4.13636m$, and wave gauge 3 of $x = 909227m$ and $y = -0.86364m$, so both are placed where the floating platform is but at different hight.

Mooring lines

The spring equation from Eq. (2.72) is used as horizontal mooring lines in the numerical model as a simplification. To get this simplification to be similar to the actual mooring lines, the measured force-displacement of the mooring in the experiment determines the value of the equivalent linear spring stiffness to match this[7], which is also shown in Fig. 4.3. In the numerical setup, there is also defined a pretensioning value of the three mooring lines with the values of $36.8902kN$, $37.1951kN$, and $36.8902kN$, in all cases.

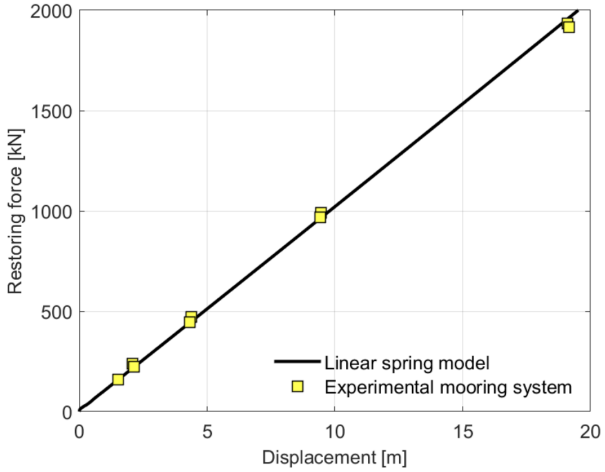


Figure 4.3: Force-displacement of the mooring[7]

Grid resolution

As the computational fluid dynamic simulation requires that the domain is discretized to solve the governing equation[35][39], also the domain of this numerical wave tank needs to be discretized into a grid or mesh. A cell size-based grid stretching approach is used to reduce the computational effort, allowing the discretized domain to have different cell sizes throughout the domain[7]. This technique works so that a high-resolution mesh is applied within a certain area, defined by a chosen center point and a distance in the x, y, and z directions, making up a small domain. Then, outside this boundary, a stretching factor is utilized, leading to coarser grid sizes further away from the high-resolution domain until it reaches a defined coarse grid cell size[7].

In this particular case, the center point of the high-resolution domain is set to $x = 9.2981m$, $y = 3.7825m$, and $z = 3.75m$ with a distance of 2.0m in the x and y directions and 1.4m in the z-direction[7]. This results in a high-resolution domain around the floating platform[7].

This method of high grid resolution around the platform and coarser away will ensure that the propagation and generation of the waves will be good in the area around the platform, which also is the most important area to capture accurate values[7]. The mesh around the platform is illustrated in Fig. 4.4.

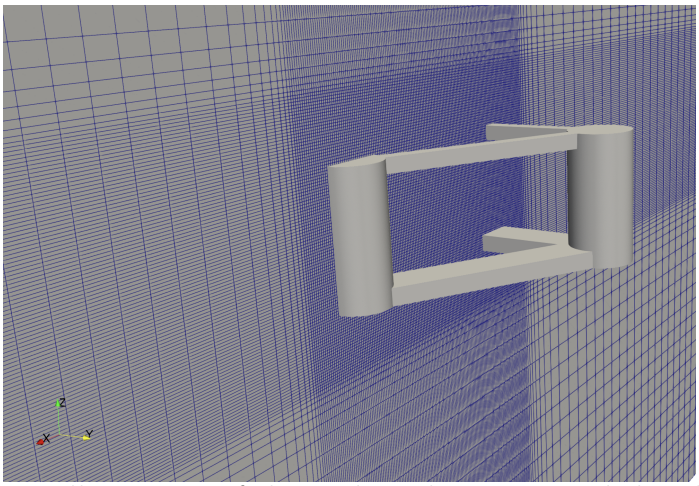


Figure 4.4: Illustration of the grid resolution around the platform.

Simulation time

For the regular wave cases, an adaptive time step is chosen as the time step, Δt of the simulation, to fulfill the criterion of CFL = 0.1 in the whole simulation[7].

4.4 Regular Waves

Three regular waves of different steepness, wave height, and wave period are simulated in a numerical wave tank with the floating platform and compared with the experiments of the regular waves from SINTEF. The non-linear wave theory, 2nd-order Stokes theory, is used in all cases with the same numerical tank dimensions, as explained before. The maximum modeling time is set to 150 seconds for the three investigated cases, 2030, 2050, and 2080. The case of 2030 is the mildest wave, and 2080 is the steepest. Water density is set to $\rho = 1000kg/m^3$ and the gravity acceleration of the z-component is $g = 9.81m/s^2$. For all the cases, the simulation is done with different grid sizes. This aims to look at the grid sensitivity and ensure accurate and reliable results for an optimized computational effort[65]. An illustration of the numerical wave tank with the floating platform is shown in Fig. 4.5.

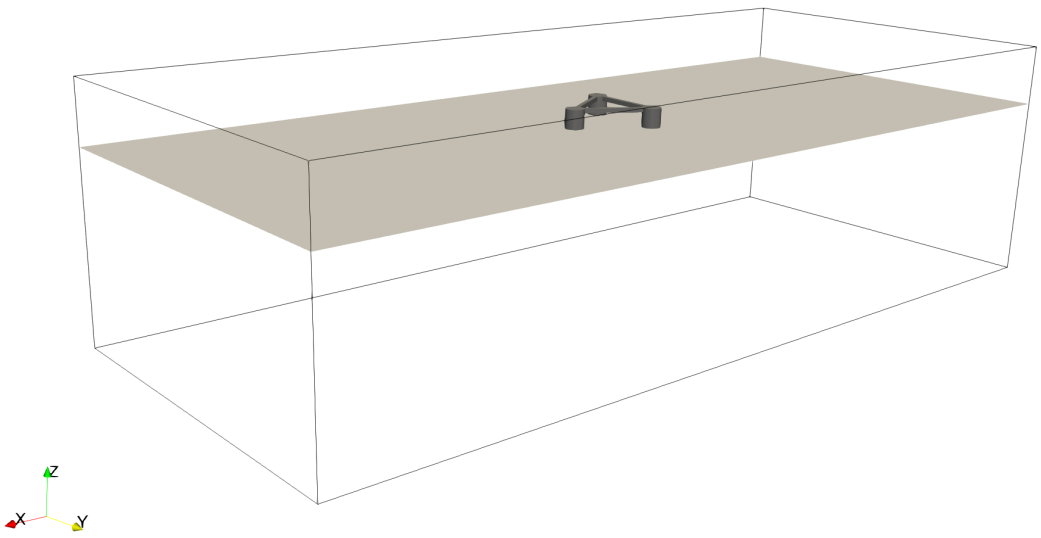


Figure 4.5: Illustration of the numerical wave tank and the floating platform.

4.4.1 Case 2030

The first case is 2030. This is the mildest wave with a steepness of 1:120. It has a wave height of $H = 2.55m$ and a wave period of $T = 14.0s$ in full scale. As the experimental model is scaled, the numerical simulation also is scaled with the model values of the wave height equal to $H = 0.06375m$ and wave period $T = 2.213594s$. These values are also shown in an overview in Tab. 4.3.

Steepness: 1:120	Full scale	Model scale
H[m]	2.55m	0.06375m
T[s]	14.0s	2.213594s

Table 4.3: Overview of the wave parameter values in case 2030[7]

Three different grid sizes are used to investigate the case, $dx = 0.025m$, $dx = 0.02m$, and $dx = 0.015m$. The analyses include wave elevation, heave, pitch, and surge motion.

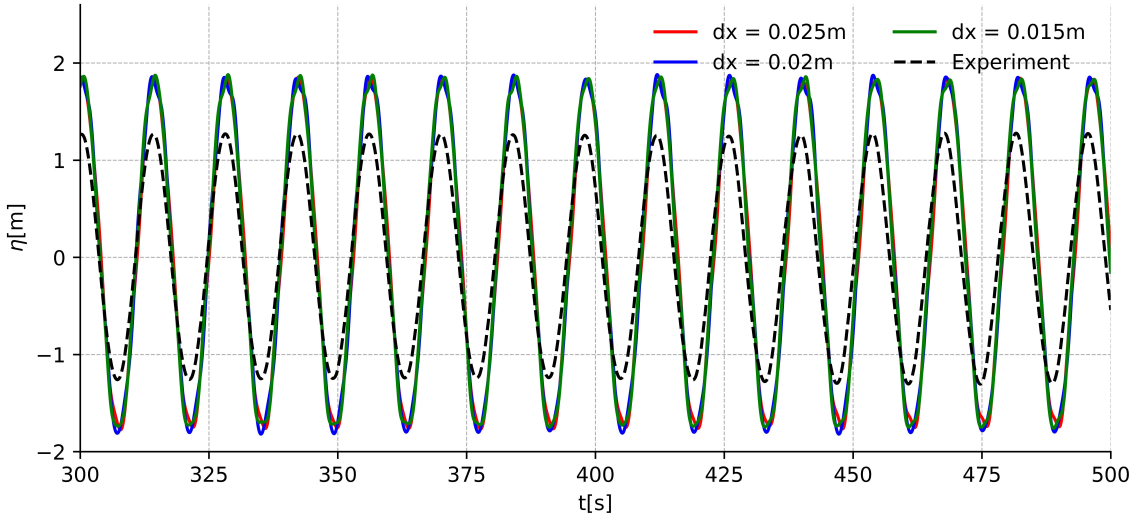


Figure 4.6: Wave elevation of 2030

The wave elevation in Fig. 4.6 shows a good agreement with the slope and period. However, it has some deviation from the experimental data at the crest and trough, with about the same value of overestimation in both, implying that the wave simulated is higher than the experimental. The grid size looks to be independent, meaning that the convergence of the simulation is not dependent on the grid size[65]. Therefore, a coarser grid size can be chosen, reducing the computational effort compared to a finer one.

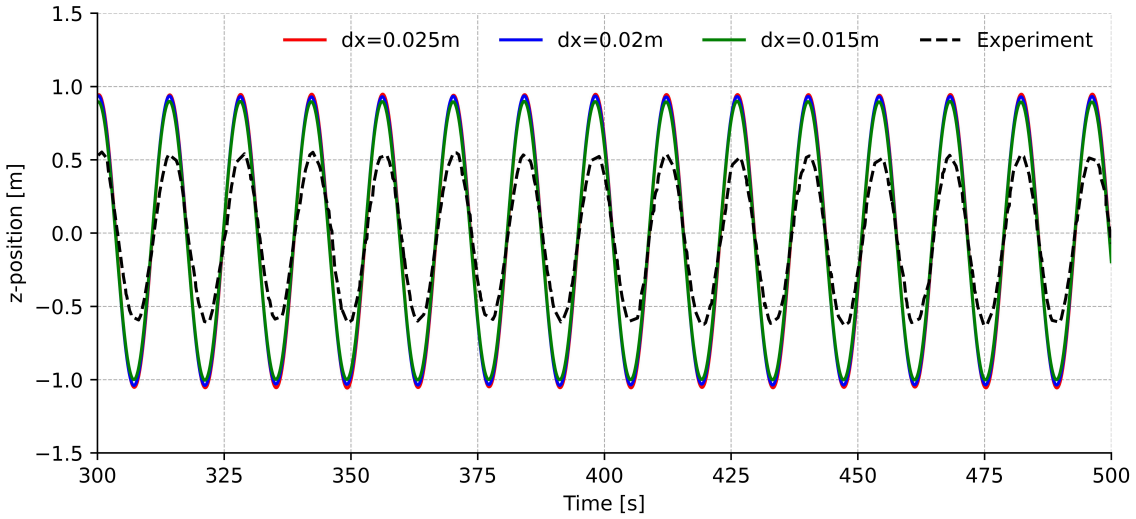


Figure 4.7: Heave motion of 2030

Fig. 4.7 presents the heave motion against the experimental data. This result agrees well with the slope and period but has significant symmetrical deviations at the top and bottom throughout the simulation. In addition, it can be observed that resizing the grid cells exhibits a nearly identical outcome. Consequently, the final result, similar to the wave elevation, is not affected by a specific grid size, allowing for a coarser grid size selection[65].

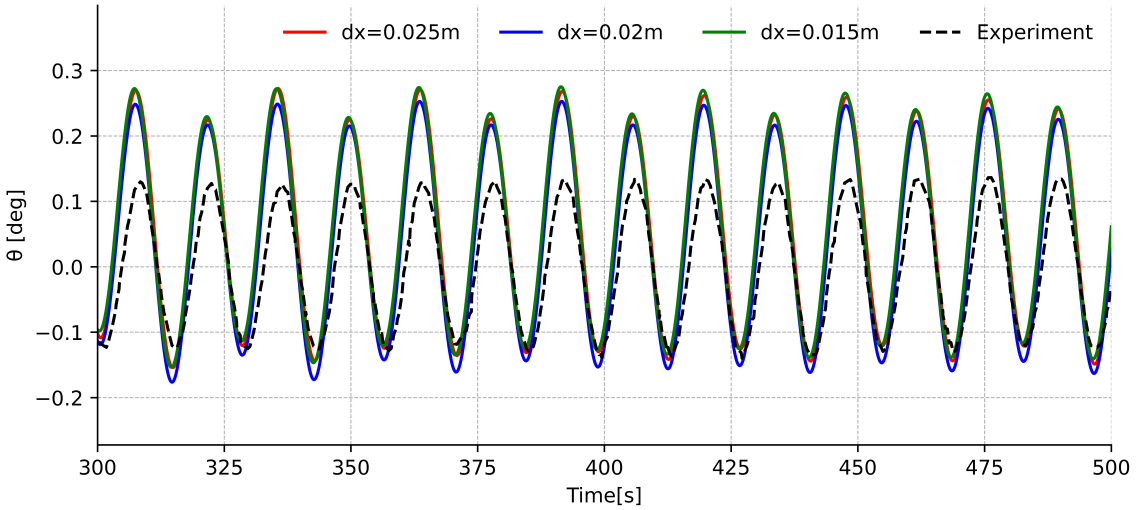


Figure 4.8: Pitch motion of 2030

The pitch motion from Fig. 4.8 shows a good agreement with the value of the troughs, but the crest is higher than the experimental data. As the values of the pitch motion are very small, it can be sensitive to changes. Just a slight irregularity in an external factor could result in significant variations, making accurate results challenging to

obtain. Another aspect to consider is that only one side of the simulation is overestimated, indicating that there may be some unequal distribution of the mass of the floating platform that causes it to tilt more over to one side. This can, for example, be that the tower's mass is not neglected in the simulation. Like the previous motions, the grid sizes for the pitch motion show roughly the same results and can be considered grid independent.

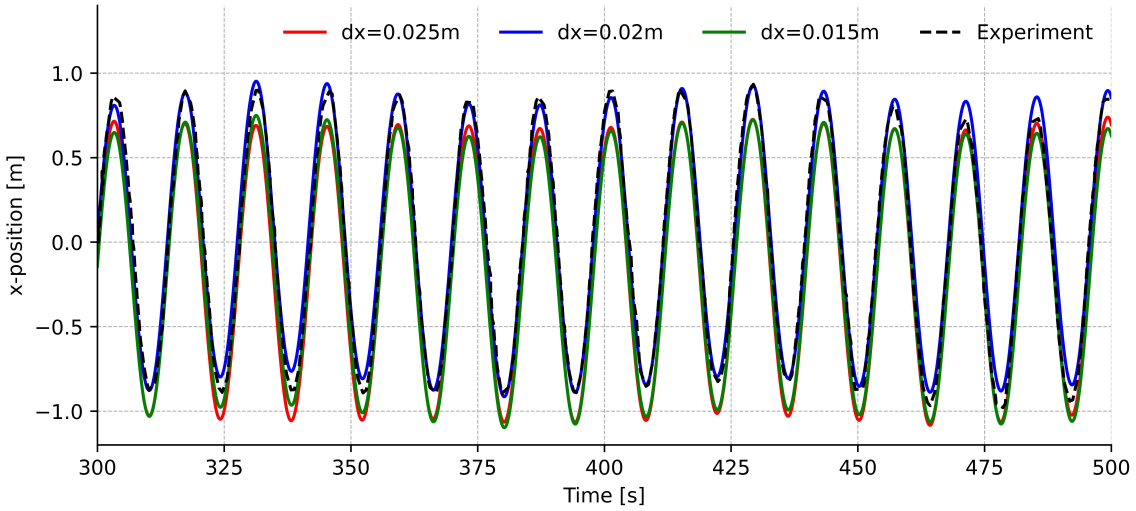
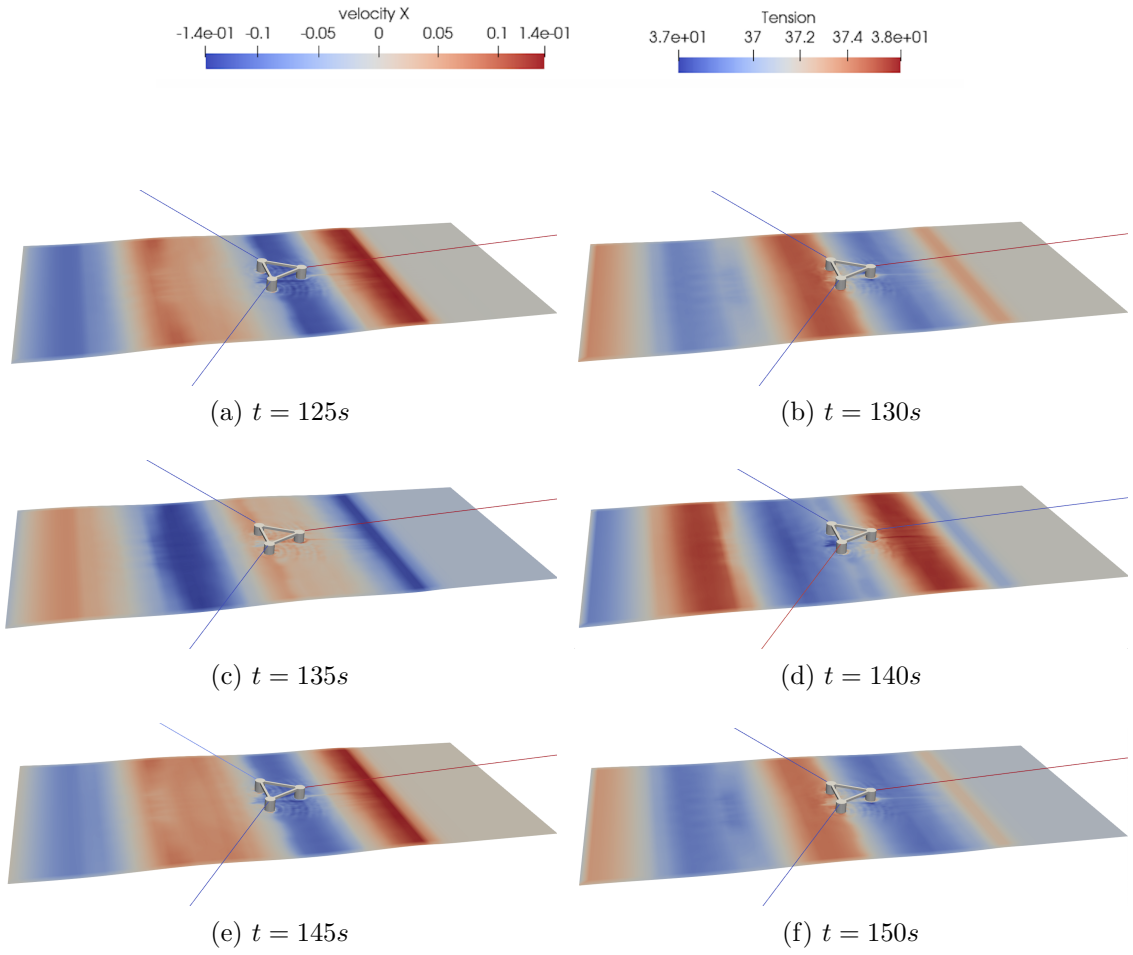


Figure 4.9: Surge motion of 2030

Fig. 4.9 represents the surge motion. Here, the numerical simulation shows an acceptable agreement with the experimental data. However, the simulation of the different grid sizes shows minor differences. The coarsest grid size of $dx = 0.025m$ has a smaller crest than the experiment and an overestimation of the trough, the same is for the finest grid size of $dx = 0.015m$, but here the deviation is slightly smaller. In comparison, the grid size in between has a small underestimation of the trough and matches the crest very well. Thus, the best match is for the grid size of $dx = 0.02m$, which is also a very good match. Furthermore, it oscillates constantly around 0 meters, meaning that the floating fundament is not drifting away but standing still in the water with help from the mooring lines.



(a) $t = 125s$ (b) $t = 130s$

(c) $t = 135s$ (d) $t = 140s$

(e) $t = 145s$ (f) $t = 150s$

Figure 4.10: Wave propagation of 2030 and interactions with the floating platform at different timesteps

In Fig. 4.10, the wave propagation of 2030 at different time steps, from $t = 125s$ to $t = 150s$, can be seen, together with the interaction of the floating platform, showing the values of the horizontal velocity and tension in the mooring lines. When the waves hit the platform, forces are exposed on the structure and the mooring lines. It can be observed that the highest horizontal velocity occurs at the wave crest. The mooring line closest to the absorption zone gets high tension before the wave hits, while the mooring lines on the other side get high tension after the wave has interacted.

4.4.2 Case 2050

The second case with regular waves is 2050. This case has steeper waves than 2030 with a steepness of 1:30. The waves are of wave height $H = 6.30m$ and wave period $T = 11s$ in full scale, and $H = 0.1575m$ and $T = 1.739252s$ in model scale, also shown in Tab. 4.4.

Steepness: 1:30	Full scale	Model scale
H[m]	6.30m	0.1575m
T[s]	11.0s	1.739252s

Table 4.4: Overview of the values in case 2050[7]

To analyze the wave elevation, heave, pitch, and surge motion, the same three grid sizes as in 2030 are used for simulation, $dx = 0.025m$, $dx = 0.02m$, and $dx = 0.015m$.

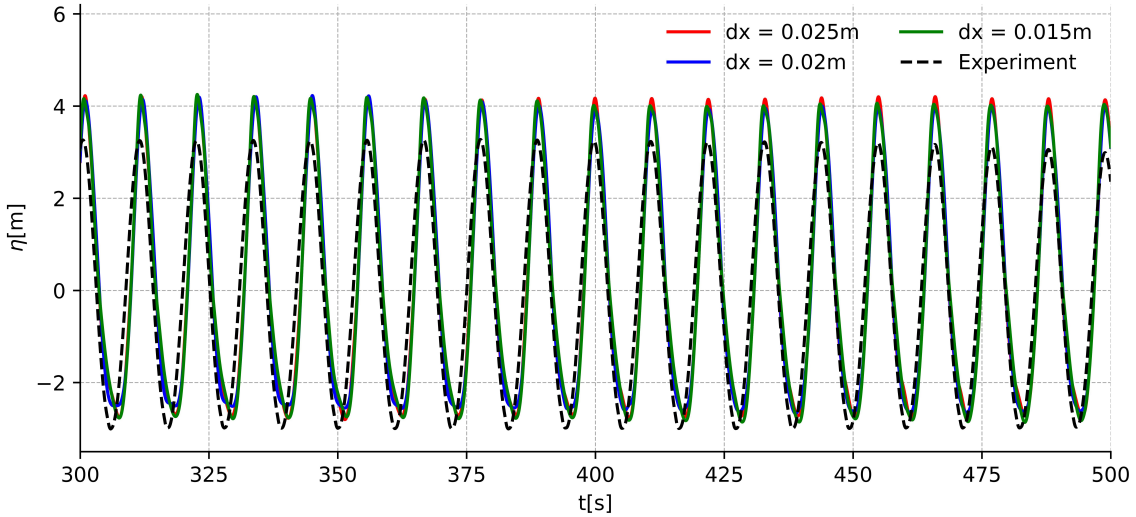


Figure 4.11: Wave elevation of 2050

The wave elevation of case 2050 from Fig. 4.11 agrees well with the trough and slope. Compared to 2030, it is clear that this case has a steeper slope. However, the crest is overestimated by about 1 meter, leading to more considerable wave heights in the numerical simulation compared to the experiment.

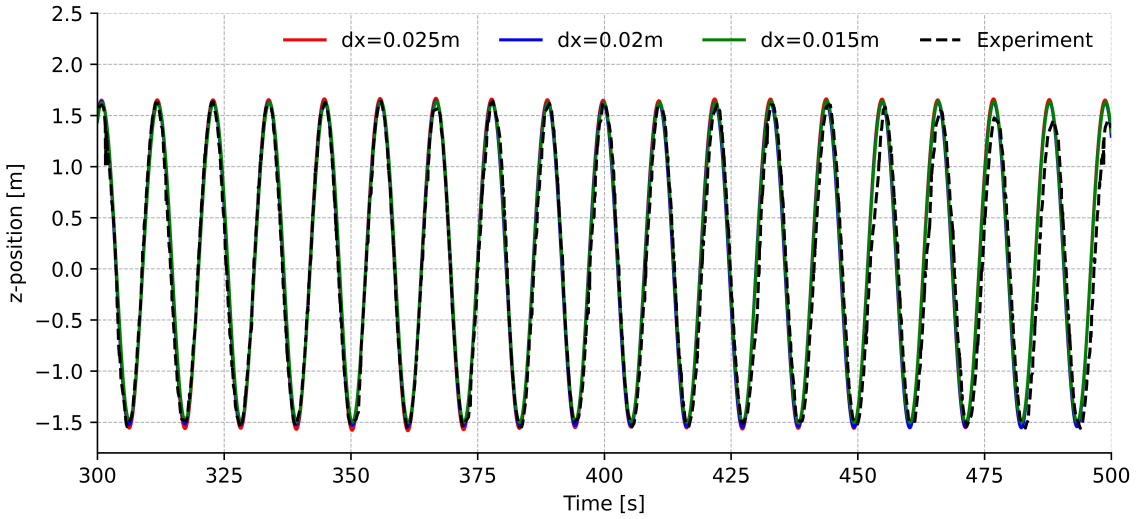


Figure 4.12: Heave motion of 2050

The heave elevation in Fig. 4.12 has a perfect match with all grid sizes compared with the experimental data. This level of agreement between the simulated heave elevation and the experimental measurements can suggest a high degree of fidelity in the model's representation of the physical phenomenon.

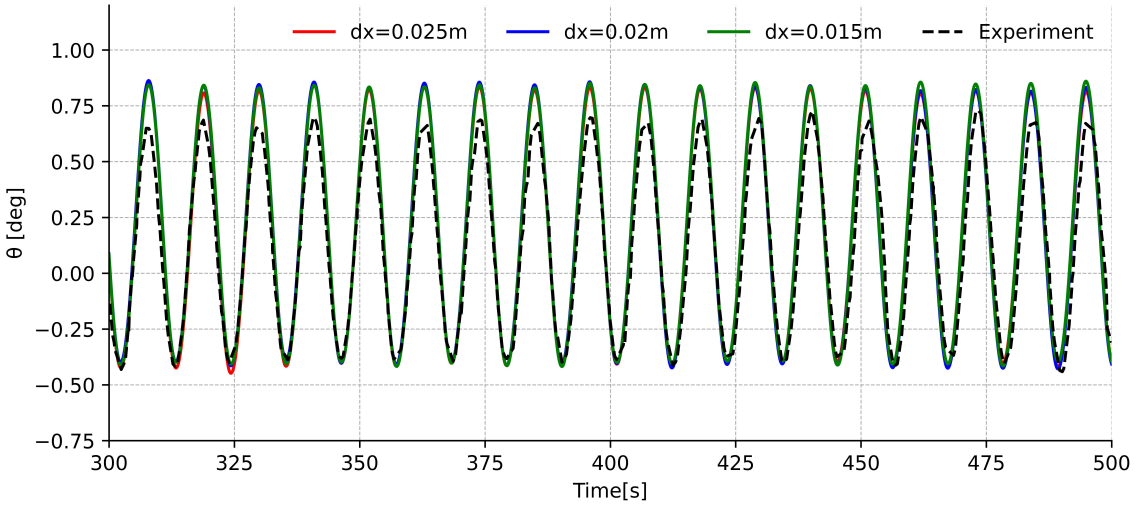


Figure 4.13: Pitch motion of 2050

The pitch motion in Fig. 4.13 is almost a perfect match but with a deviation of the crest of about 0.2 degrees. Overall, the rotation of the platform is minimal, so it is difficult to get the exact values of a simulation because of the significant effects of small changes, alike the pitch motion in 2030.

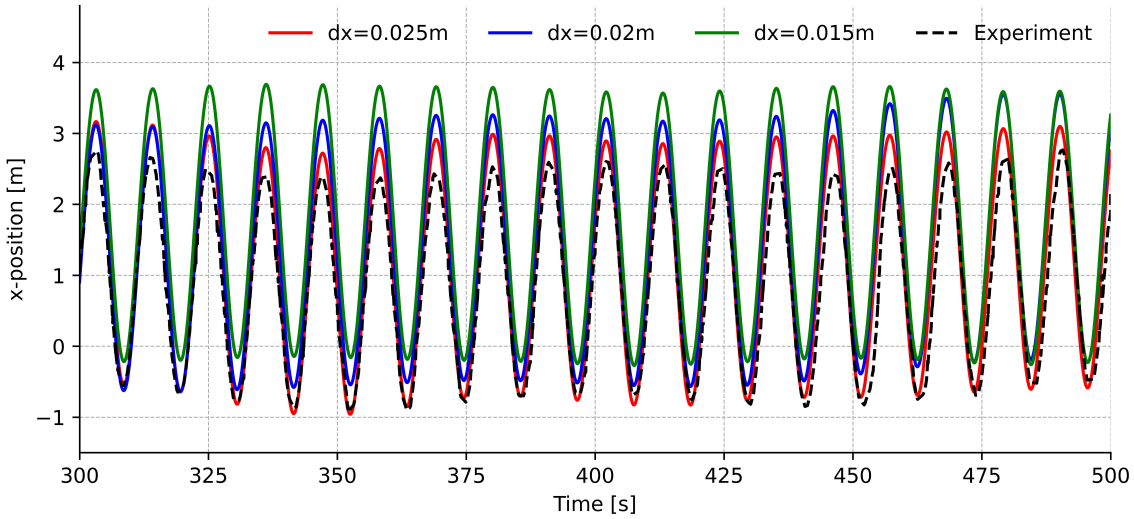
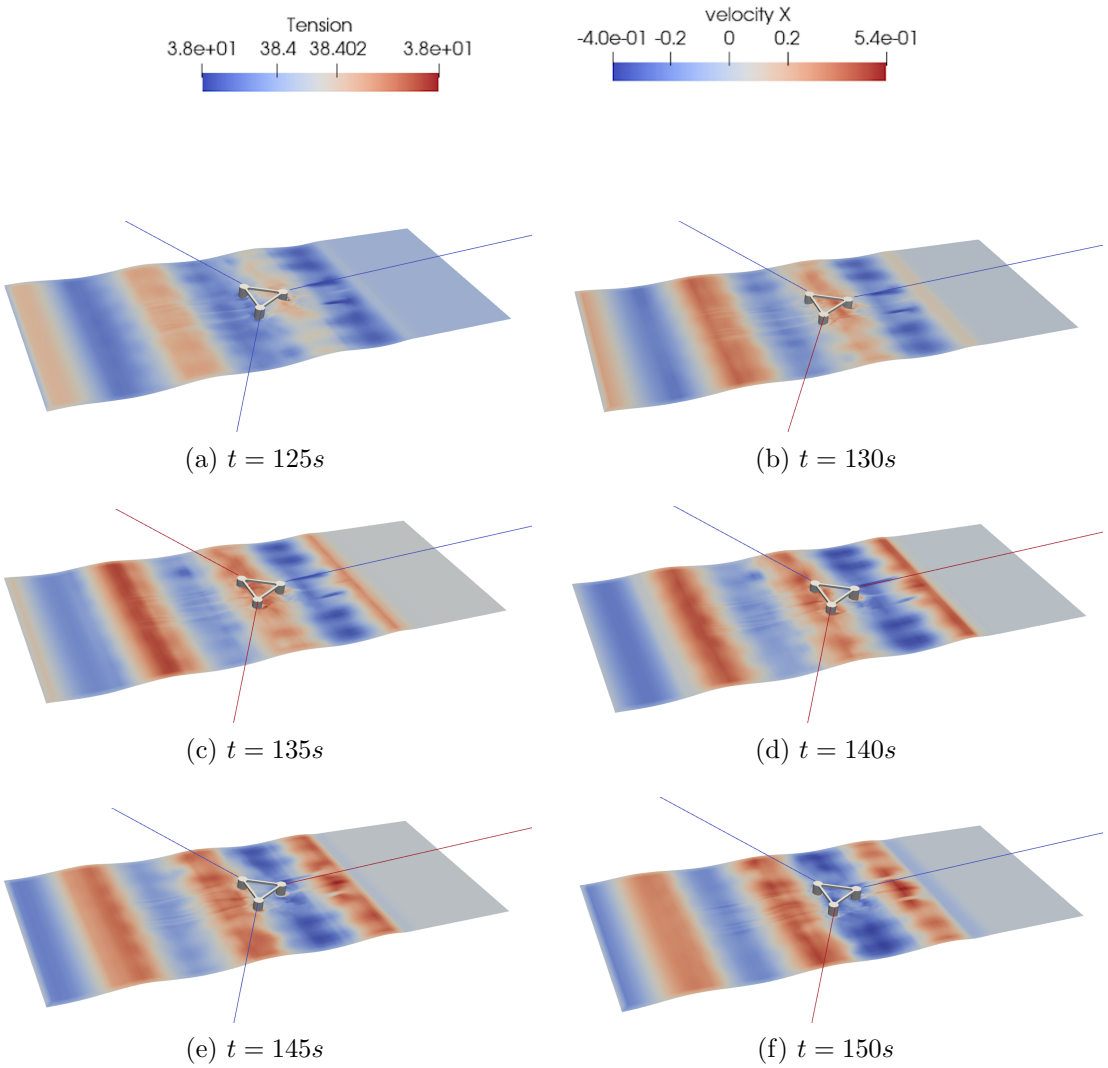


Figure 4.14: Surge motion of 2050

Fig. 4.14 represents the surge motion. This is the only of the analyzed motions in the case of 2050 that is not grid independent. Here it is clear that the result varies with different grid sizes. Strangely, the simulation does not converge to the experimental result with a higher resolution but with a coarser grid size. The best match is with the coarsest grid size of $dx = 0.025m$, while the worst is with the finest grid size of $dx = 0.015m$. In the simulation with $dx = 0.025m$, the result is very good. The trough is almost a perfect match but with a slight crest deviation. Compared to 2030, there are higher values of the surge motion, which can indicate that larger forces are leading to more significant motions in the x-direction, coming from the steeper and higher regular waves.



(a) $t = 125s$ (b) $t = 130s$

(c) $t = 135s$ (d) $t = 140s$

(e) $t = 145s$ (f) $t = 150s$

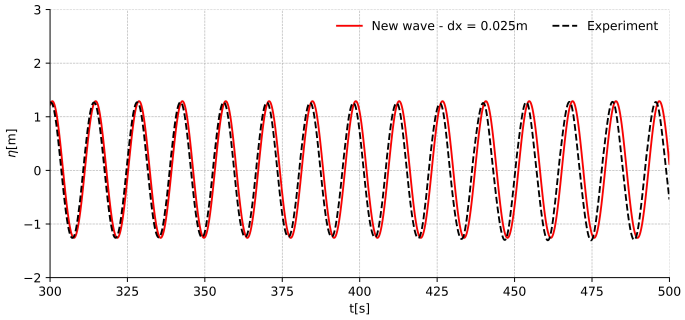
Figure 4.15: Wave propagation of 2050 and interaction with the floating platform

Fig. 4.15 shows the wave propagation of 2050 and its interaction with the floating platform. Compared to 2030, horizontal velocity and tension values are higher, which can align with the steeper wave of this case.

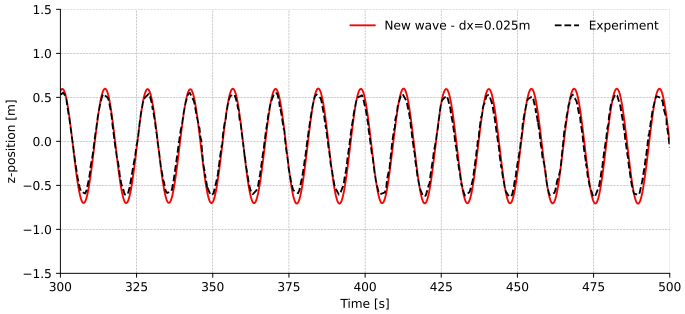
Adjustment of the wave height of 2030 and 2050

The results from 2030 and 2050 are relatively good but have some deviations in the wave elevations and the motions. As an inaccurate wave elevation can greatly impact the motions, the incoming waves are investigated[7]. By comparing with the experimental data, it indicates that the wave height should be decreased to achieve a sufficient alignment. For 2030, the wave height is then set to $H = 0.05025m$, and for 2050 the wave height is set to $H = 0.1375m$ in model scale. A new simulation with these values is carried out with the grid size of $dx = 0.025m$. The results are shown below in Fig. 4.16 and 4.17 for wave elevation and heave, pitch, and surge motion.

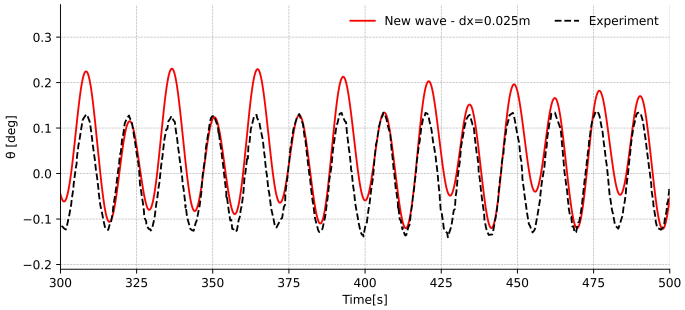
Significant enhancements are exhibited in the outcome through the adjustment of wave height. The first perspective for consideration is that the simulation of the wave elevation now accurately replicates the wave from the experiment in both 2030 and 2050, which appear as a crucial factor with the goal of recreating comparable platform motion responses[7]. This is reflected in the new result in Fig. 4.16 and 4.17, especially the heave, pitch, and surge motion of 2030, and the pitch and surge motion for 2050 showed significant improvements.



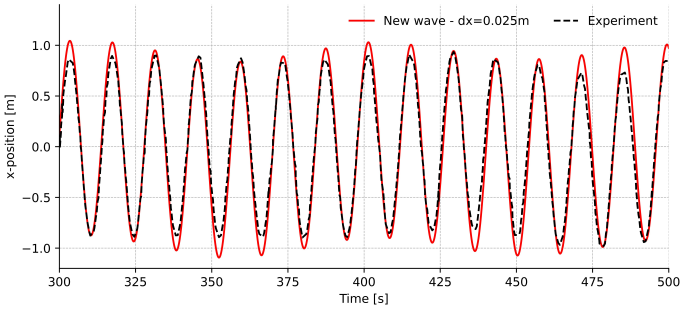
(a) Wave elevation of 2030 with decreased wave height



(b) Heave motion of 2030 with decreased wave height

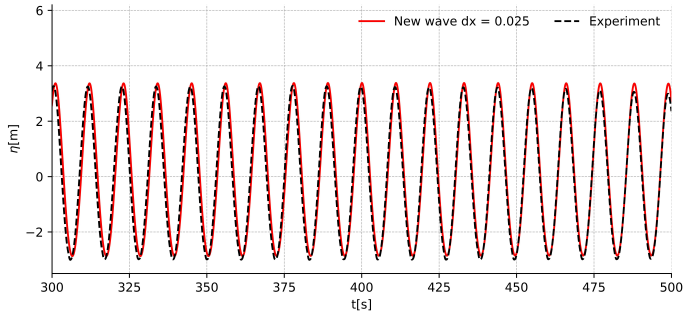


(c) Pitch motion of 2030 with decreased wave height

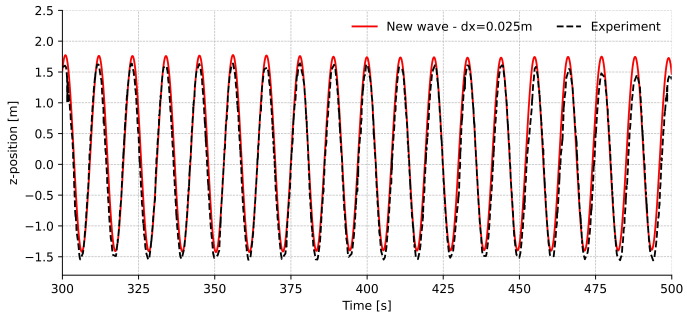


(d) Surge motion of 2030 with decreased wave height

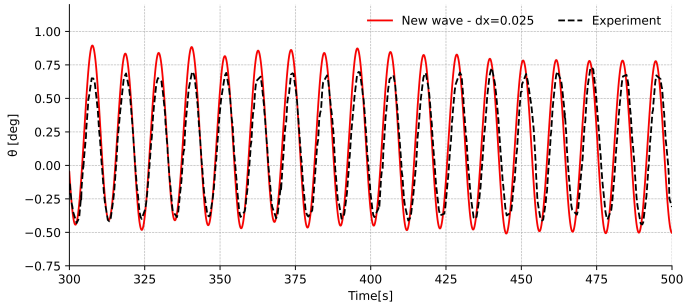
Figure 4.16: Wave elevation and motions of 2030 with a new wave height



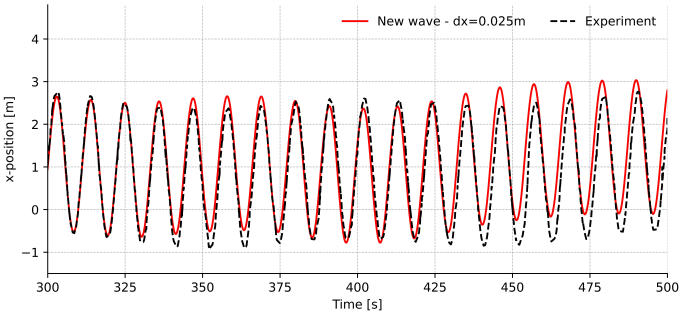
(a) Wave elevation of 2050 with decreased wave height



(b) Heave motion of 2050 with decreased wave height



(c) Pitch motion of 2050 with decreased wave height



(d) Surge motion of 2050 with decreased wave height

Figure 4.17: Wave elevation and motions of 2050 with a new wave height

4.4.3 Case 2080

The last case is 2080. This is also the case with the highest wave steepness of 1 : 15. The wave height $H = 12.59m$ and wave period $T = 11.0s$ in full scale. In the model scale, the wave height $H = 0.31475m$ and period $T = 1.739252s$. A complete overview is given in Tab. 4.5.

Steepness: 1:15	Full scale	Model scale
H[m]	12.59m	0.31475m
T[s]	11.0s	1.739252s

Table 4.5: Overview of the values in case 2080[7]

Three different grid sizes are used for a grid convergence study of the wave elevation, heave, pitch, and surge motion, $dx = 0.035m$, $dx = 0.03m$, and $dx = 0.025m$.

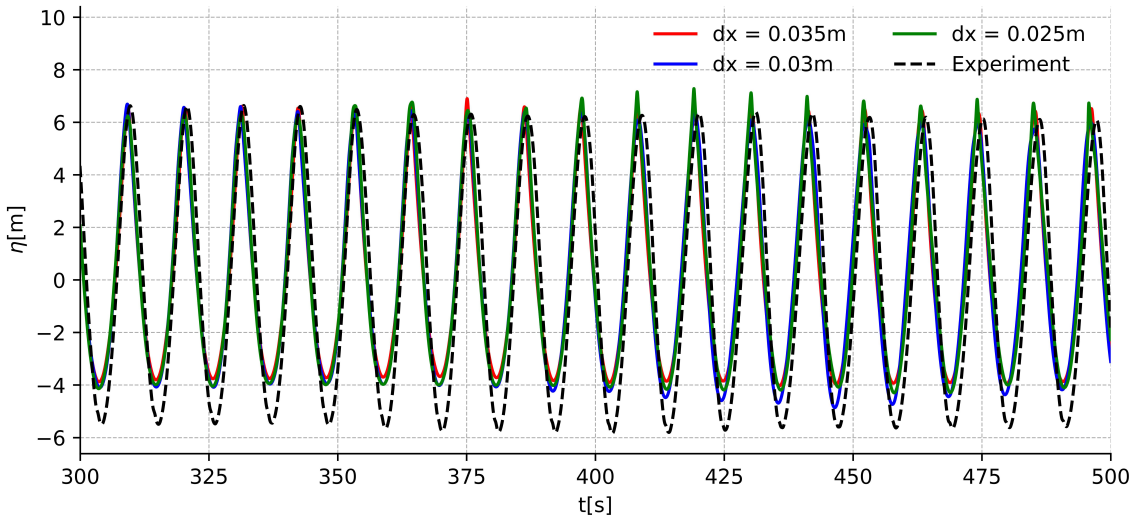


Figure 4.18: Wave elevation of 2080

Fig. 4.18 shows the simulated wave elevation compared with the measurements. The simulation has a good agreement with the measurements. Some deviation occurs at the crest and bottom. From $t = 300s$ to about $t = 400s$, the simulated waves are underpredicted, but after about 400 seconds to 500 seconds, the simulation almost matches the measurements. The different grid sizes show almost the same results, so the grid size does not significantly impact the result.

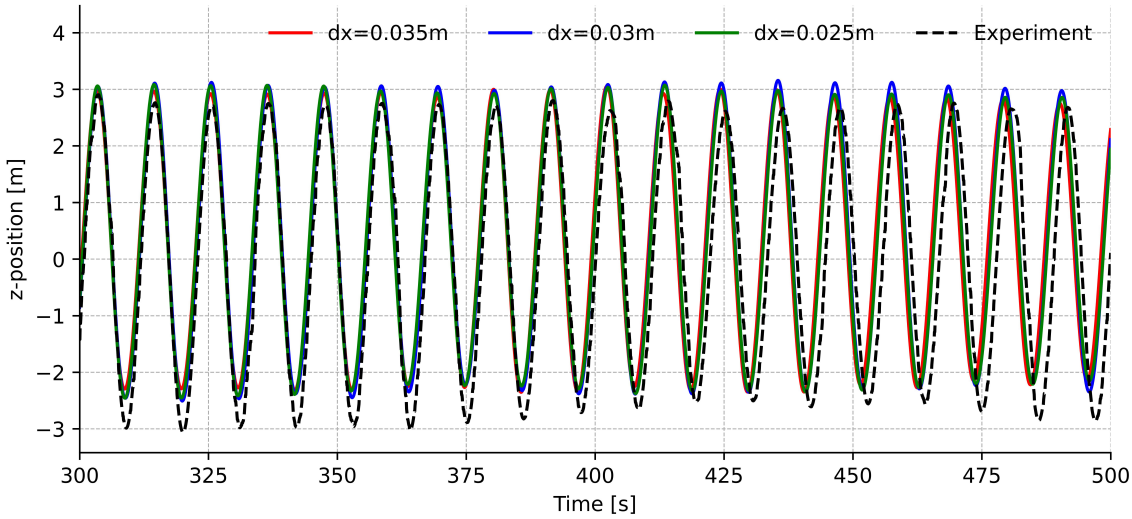


Figure 4.19: Heave motion of 2080

The heave motion of the floating platform is shown in Fig. 4.19. The result has a good match with the experimental data. From $t = 300s$ to $t = 400s$, the simulation has a slight overprediction of the motion and an underprediction of the trough. After $t = 400s$, there is a better match with the trough but slightly more overestimating of the crest. It also appears to be a small shift in the period.

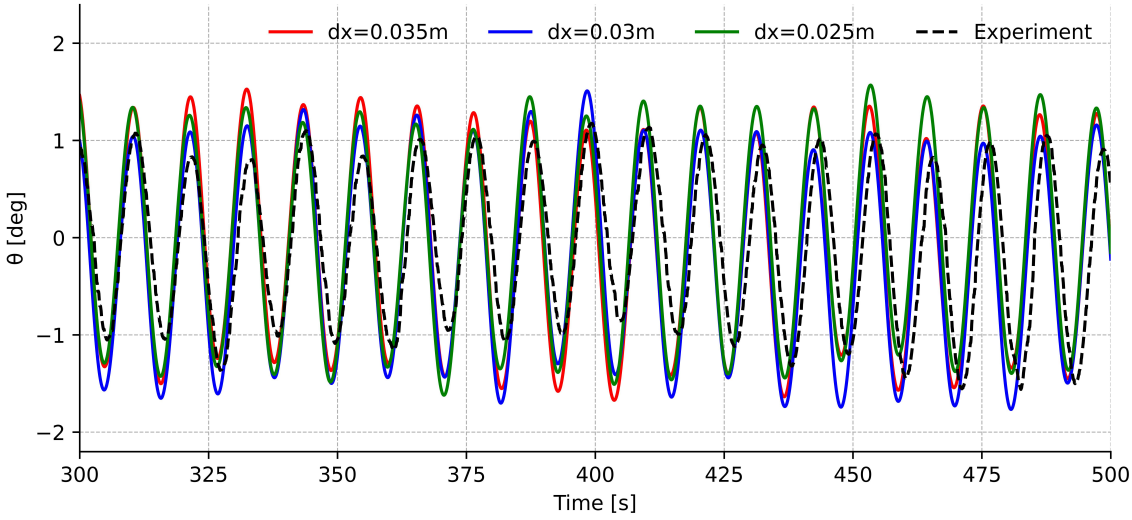


Figure 4.20: Pitch motion of 2080

In Fig. 4.20, the pitch motion is plotted. This has a relatively good agreement with the measurement but with some deviations, mainly overestimations. The pitch motion oscillates around 0 degrees and with an amplitude of approximately 1 degree. Similar to the heave motion, the pitch motion also gets a small offset in the period after about $t = 400s$. The grid sizes show small variations, but there is no apparent

convergence of the higher-resolution grid size.

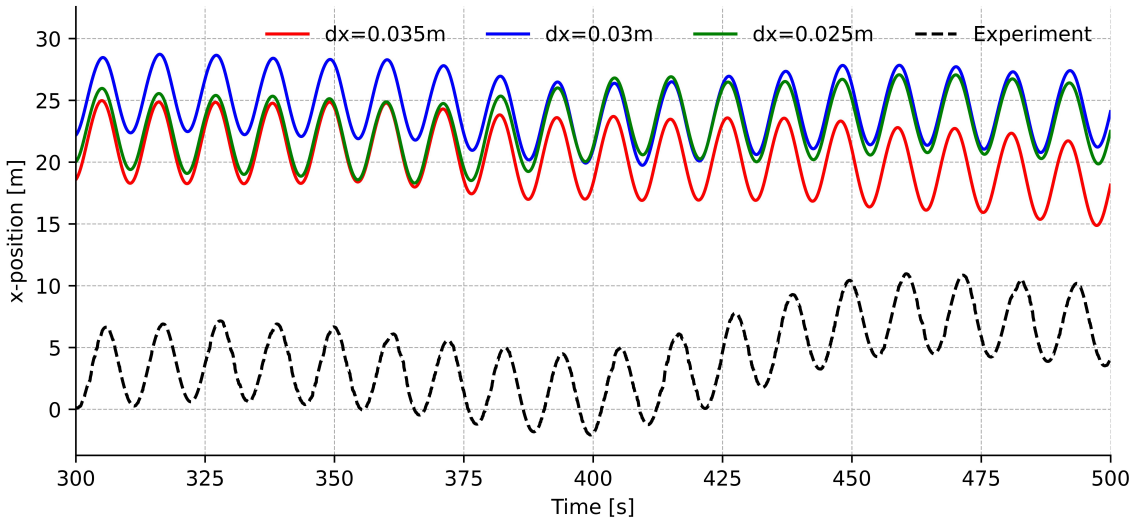


Figure 4.21: Surge motion of 2080

Fig. 4.21 shows the surge motion. This result presents a considerable deviation compared to the measurements. The simulated values of the surge motion are overpredicted during the whole simulation, independent of the grid size, consistently yielding approximately 20 meters above the experimental data. Moreover, no clear convergence pattern emerges from the analysis, as the coarsest grid size generates results relatively closer to the experimental measurements than the finest one. A possibility for this outcome could be that the grid sizes studied are too narrow in size, so there will not be a clear distinction between them. Another factor contributing to this result could be the uncertainty related to the pretension of the mooring lines. If this value does not align with the real one, this could lead to a significant difference in surge motion.

As this case is of a steep wave, it is also the most challenging case to get good results. This comes from the fact that a steeper wave has a flow field that consists of more significant gradients, which also require greater demand from the CFD simulation[58]. A consequence is that the numerical simulation can dissipate energy, for example, in the form of turbulence or breaking, leading to underestimated wave forces, which also affect the response and motion of the floating platform. This could also explain the results of the wave elevation. In addition, steeper waves are also more exposed to reflected waves from the boundary[58].

The surge motion shows an unsatisfactory outcome, so an investigation of different parameters is undertaken to improve the result.

Mooring

The first parameter to investigate is the mooring lines, which are supposed to prevent the platform from moving horizontally. In Fig. 4.22, mooring lines 0, 1, and 2 are plotted to look at their forces and how they act compared to each other.

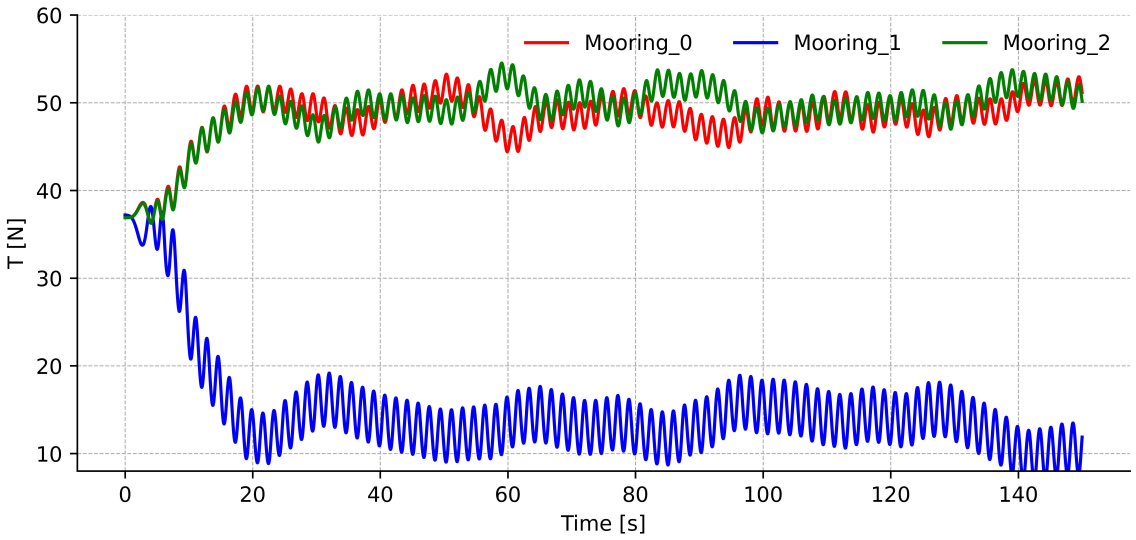


Figure 4.22: Mooring lines of 2080

Fig. 4.22 shows how the mooring lines are acting compared to each other. The distribution of the forces is plotted, starting from the pretension. Mooring 0 and 2 have the highest values and are in tension, while mooring line 1 has the smallest value, which can be seen as reasonably as the wave gives different forces to those mooring lines in front and behind the platform.

Wave forces

The next parameter to investigate are the wave forces acting on the platform in the x, y, and z directions. In order to get a clearer picture of what is influencing the mooring lines and the surge motion, there is done a simulation with no mooring (no springs) and a simulation with no waves to look at how that affects the resulting forces.

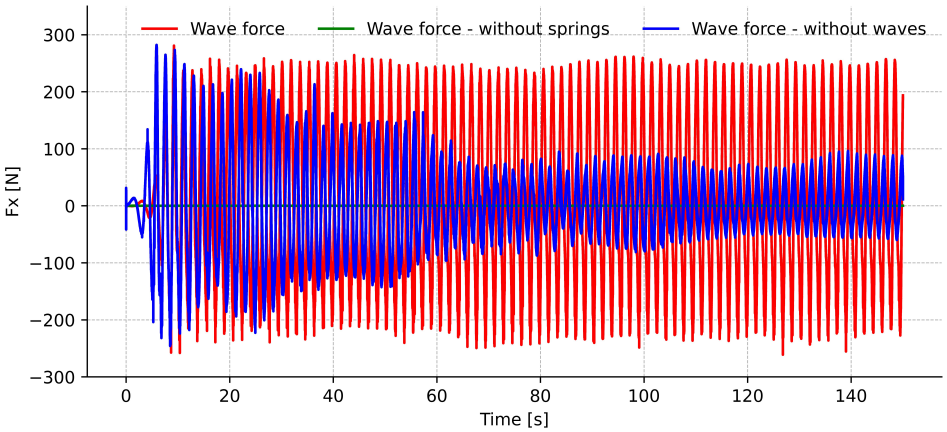


Figure 4.23: Wave force of 2080 in x-direction

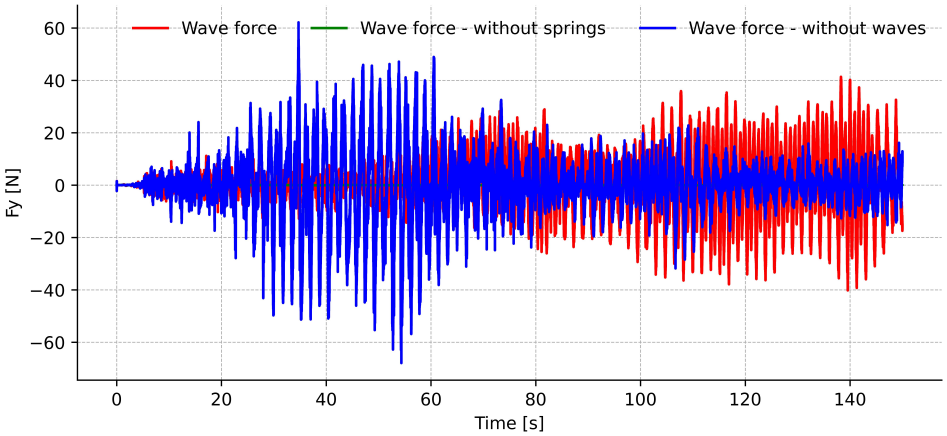


Figure 4.24: Wave force of 2080 in y-direction

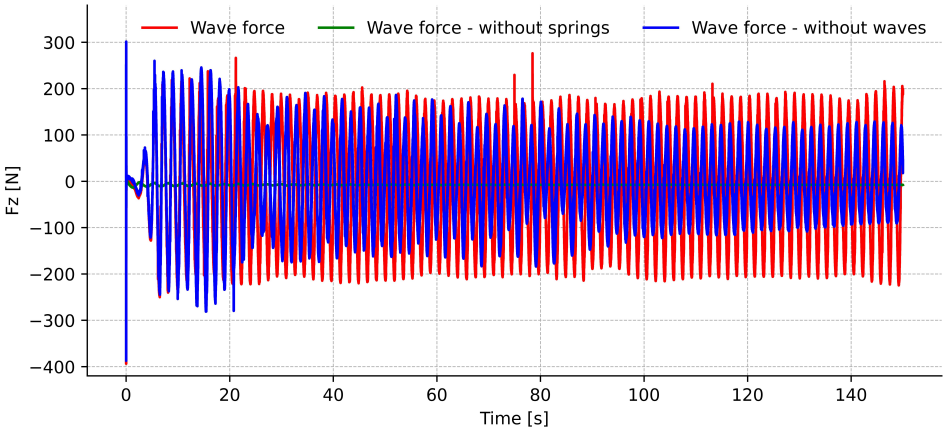


Figure 4.25: Wave force of 2080 in z-direction

The wave force in Fig. 4.23 shows the wave force in the x-direction, where the original simulated wave force has the most significant values, and the force on the mooring without waves has a relatively large force in the first 60 seconds. After 60 seconds, this force gets smaller. When there is no mooring (springs), there is no force.

From Fig. 4.24, the wave force in the y-direction is presented. When there are no waves, the forces on the mooring lines are large in the first 60 seconds, while the wave force of the regular wave has more significant forces after about 100 seconds. Similar to the x-direction forces, there are no wave forces in this direction when there are no mooring lines(springs).

The wave force of the z-direction is shown in Fig. 4.25, showing the largest wave force in the original simulation.

Immersed boundary method

A comparison between the force results obtained by the direct forcing immersed boundary method and the ghost cell immersed boundary method was done to check if the error in surge motion was affected. This did not show any interesting results.

Extention of the numerical wave tank

An extension of the numerical wave tank was conducted to check if reflected waves caused the surge error. The tank was then extended from 18.2m to 34.7m. In addition, the wave generation and absorption zones were also extended from 3.0m to 4.72m and from 6.0m to 9.44m, respectively. The length in the y-direction and the height of the tank were the same as before. The heave, pitch, and surge motions were then simulated again with the grid size $dx = 0.02m$ in this wave tank. The results are presented from $t = 300s$ to $t = 500s$.

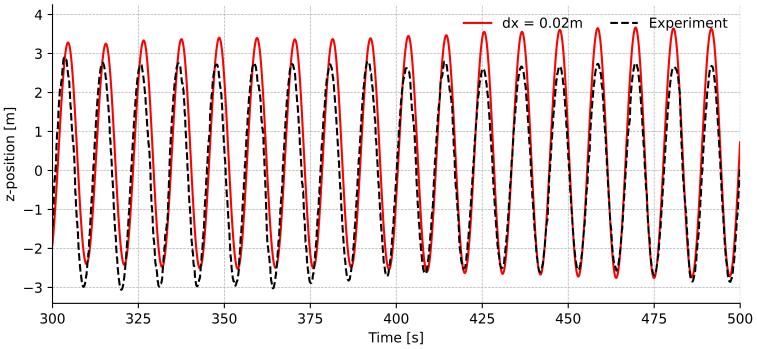


Figure 4.26: Heave motion of 2080 with extended tank

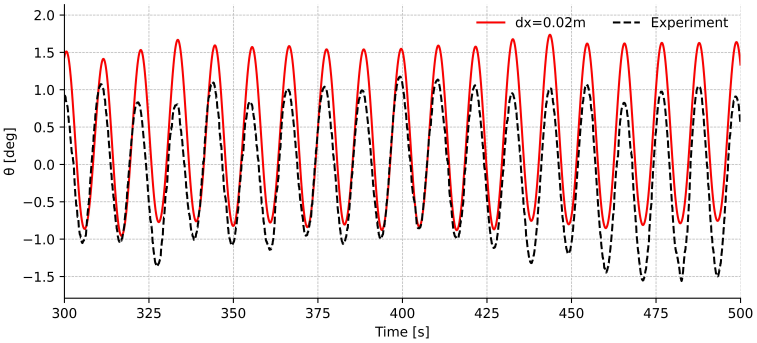


Figure 4.27: Pitch motion of 2080 with extended tank

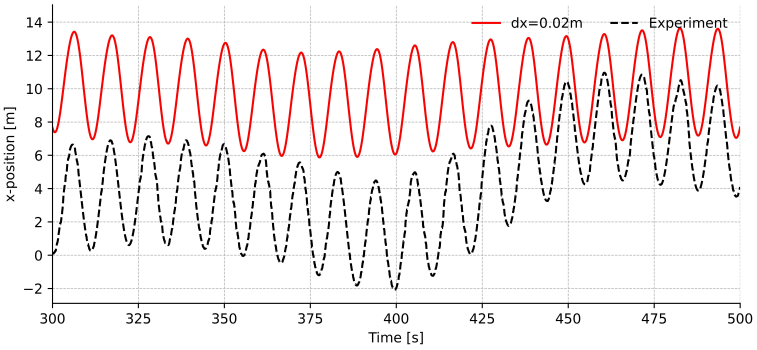


Figure 4.28: Surge motion of 2080 with extended tank

Fig. 4.26 shows the heave motion, which has a good match with the experimental data. It oscillates around 0 meters with an amplitude of about 3.0m. From $t = 300s$ to about $t = 400s$, the heave motion has some overprediction of the crest and underestimation of the bottom. After about 400 seconds, the crest is still overestimated, but the bottom is now almost a perfect match.

The pitch motion is represented in Fig. 4.27. This is also a relatively good match but has some deviations in the crest and trough. The pitch motion oscillates around 0 degrees, with an amplitude varying between about 0.5 degrees to 1.0 degrees. Up to 400 seconds, the trough has relatively good agreement with the measurements, while after 400 seconds, the trough is significantly underestimated. The crest has a slight overestimation throughout the whole time series.

The surge motion from Fig. 4.28 shows a much better result than the surge motion in the small tank in Fig. 4.21. Now the simulation is near the measurements, with a slight overestimation initially. However, after about 450 seconds, the simulation is starting to get good agreement with the experimental data. This result indicates that the reflected waves are the problem with the surge error from the smaller tank. Looking at the mooring forces from this simulation Fig. 4.29, it is also clear that the mooring lines now have smaller forces in the larger tank than in the smaller tank. This result shows that when the waves are reflected on the platform, there will also be a more significant impact on the mooring lines due to the higher and stronger waves.

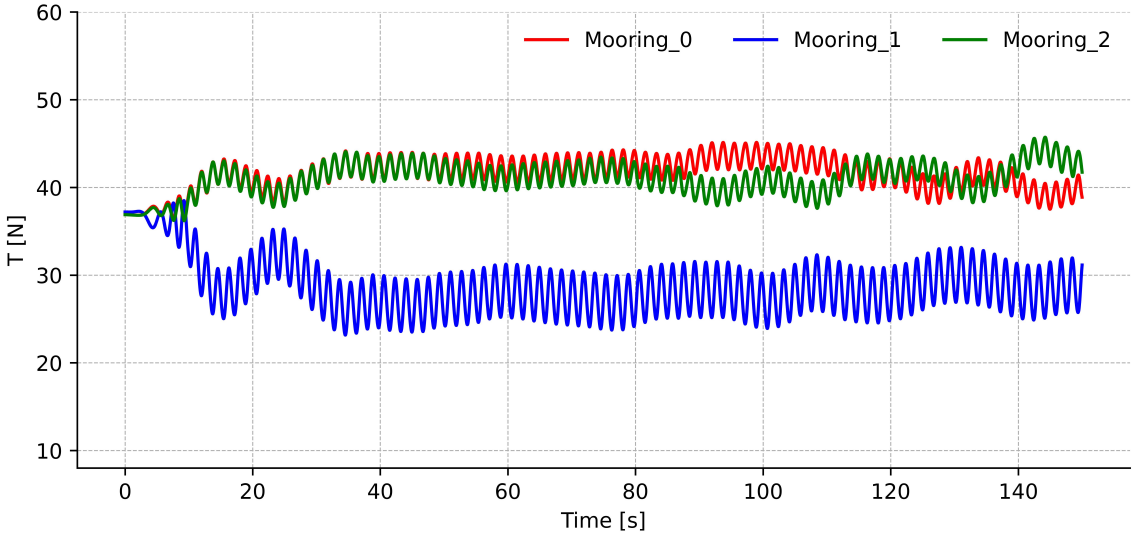


Figure 4.29: Mooring lines of 2080 after extending the tank

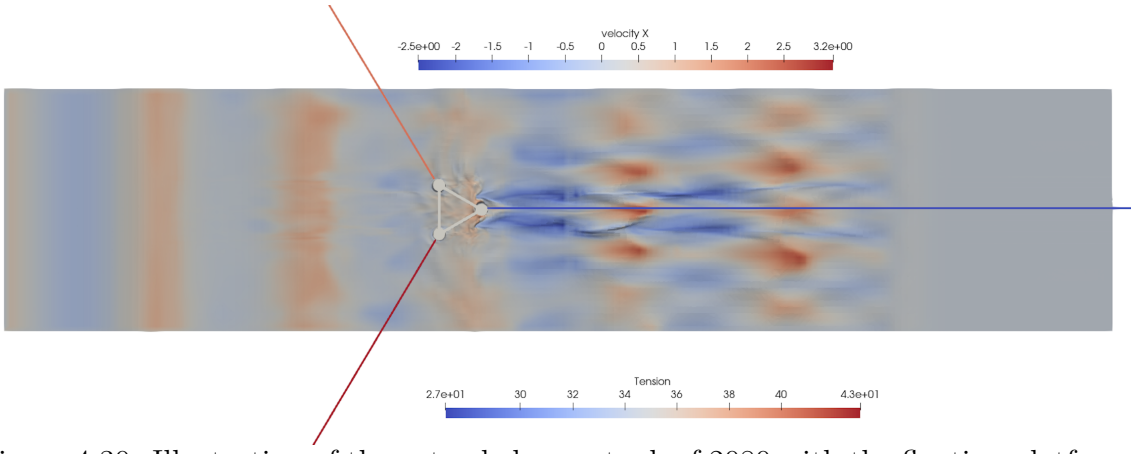


Figure 4.30: Illustration of the extended wave tank of 2080 with the floating platform and mooring lines.

An illustration of the floating platform in the regular waves of 2080 with an extended tank is shown in Fig. 4.30. The steep wave from 2080 will lead to high horizontal velocities and mooring tension, giving larger forces interacting with the platform. This can be observed with the free surface being split and deformed significantly after the interaction.

Chapter 5

Conclusion and Outlook

In the floating offshore wind industry, the challenging environment surrounding the platforms is of significant importance when designing the structures. By validating numerical methods, like CFD, floating offshore wind turbines can be cost-effectively optimized with safe design, an essential element for this industry to be competitive and actively contribute to the transition towards renewable energy. This thesis uses the numerical model REEF3D to validate the 6DOF algorithm. The first two cases for validation were in 2D with easy geometries to provide good results of simple cases before approaching the more complex geometry of a floating offshore wind turbine.

The first case was of a cylinder, realized from an initial position and to the water, where the heave motion was studied and compared with experimental data. A good agreement with the two finest grid sizes of dx = 0.01m and dx=0.005m was obtained, while the coarsest grid size of dx = 0.025m showed a significant deviation after the first oscillation. This observation indicated the most appropriate grid size, dx=0.01m, for further investigation, as it provides accurate results with an optimized need for computational effort.

The following case of the 2D investigation was of a rectangular box. In contrast to the cylinder, this case was investigated with a generation of waves, where a regular wave of second-order Stokes theory was used. A generation zone and absorption zone were activated at the start and end, respectively, for this purpose of study, dividing the numerical tank into three parts. As an external force affected the floating box due to the generated waves, this also included more possible movement of the floating box. Therefore, the three motions of heave, roll, and surge, together with the wave elevation, were studied and compared with measurements with three different grid sizes for a grid convergence study. The first simulation was of the wave elevation, a fundamental component that should be accurate for the rest of the simulations to be of good quality. This simulation had the best match with the measurements of

the finest grid size of dx=0.005m, with just a small overestimation at the crest and trough. Also, the heave, roll, and surge motion matched the experimental data at this grid size. However, some problems with larger overestimations occurred with the coarser grid sizes for roll and surge motion, implying a grid dependence of the simulations.

The primary investigation of SINTEF's WINDMOOR, the floating offshore wind turbine, was done as the last case. As this was a 3D case, the motion of heave, pitch, and surge were studied in three cases of different regular waves, 2030, 2050 and 2080. The simulations were compared to measurements, which were carried out by SINTEF. The cases of 2030 and 2050 showed very good agreement with measurements but with slight overestimations in some of the motions. Both cases marginally overestimated the crest and trough of wave elevation and pitch motion. The deviation in the pitch motion could be caused by its small value, leading to a great impact on the result with small changes, making it difficult for accurate simulations. Another uncertainty could be unequal mass distribution due to the neglected tower in the simulations. For the heave motion result, 2030 had larger overestimations than 2050, which nearly matched the experimental data, while the surge motion in 2030 had a slightly better result than 2050. However, also 2050 matched well with measurements in this motion. As the wave elevation, pitch, and surge for 2030 and 2050, in addition to heave in 2030, had a slight deviation from the experimental data, the wave height was decreased in order to match the measured waves better. This produced favorable results for all motions, indicating that calibrating the incoming waves is essential to receive a good agreement with the experimental data. In total, both cases had good results, but 2050 was somewhat better.

The wave of 2080 was the most difficult case to simulate due to the high wave steepness. However, a good match of the wave elevation and the motions were obtained, except for the surge motion, which had a significant overestimation and did not show any convergence in the grid size study. A reason for the non-convergence was discussed, and it was noted that the narrow grid investigation and the uncertainty of the pretension of the mooring lines could be a problem. For future investigations, a grid convergence study of $dx = 0.025m$, $dx = 0.05m$, and $dx = 0.1m$ can be suggested to see if this could change the convergence result. Further investigation was then done for improvement, with a parameter study of the mooring lines, wave forces, immersed boundary method, and extension of the tank. The decisive parameter was the length of the tank, which was too short, so the waves were reflected, giving more significant movements. An extension of the tank then gave a better result of the surge motion.

Overall, all cases showed a relatively acceptable agreement with the experimental data. For steeper waves, the free surface is observed to be of higher separation than the less steep waves behind the structure, so an extension must be considered in such cases to avoid reflecting waves. The validation of the 2D and 3D cases in this thesis has shown REEF3D is suitable for floating body simulation similar to this. The validation has considered the heave, roll, and surge motion for 2D cases and the heave, pitch, and surge motion for three regular waves in 3D. However, further research on irregular wave simulation should give a more realistic picture of the ocean. Also, a deeper investigation of the mooring lines and comparing them to experimental data could be of interest, as this is an essential parameter for the platform's safety and to be cost-efficient. Coupling with the CFD and FNPF modules of REEF3D could also be of high interest to increase efficiency combined with high fidelity of large-scale simulations, such as a wind farm. This could then contribute to lowering the cost and getting one step closer to a competitive floating offshore wind industry.

Bibliography

[1] Large eddy simulation of focused breaking waves with different wave steepness. *Ocean Modelling*, 179:102122, 2022.

[2] G. B. Airy. *Tides and waves*. B. Fellowes, 1845.

[3] J. D. Anderson, G. Degrez, E. Dick, and R. Grundmann. *Computational fluid dynamics: an introduction*. Springer Science & Business Media, 2013.

[4] S. F. Ashby and R. D. Falgout. A parallel multigrid preconditioned conjugate gradient algorithm for groundwater flow simulations. *Nuclear science and engineering*, 124(1):145–159, 1996.

[5] B. Baldwin and T. Barth. A one-equation turbulence transport model for high reynolds number wall-bounded flows. In *29th aerospace sciences meeting*, page 610, 1991.

[6] P. A. Berthelsen and O. M. Faltinsen. A local directional ghost cell approach for incompressible viscous flow problems with irregular boundaries. *Journal of computational physics*, 227(9):4354–4397, 2008.

[7] P. A. Berthelsen, M. Thys, A. M. Kamath, T. Martin, and H. S. Bihs. Numerical simulation and comparison with experiments of a floating wind turbine using a direct forcing method. In *Trends in Renewable Energies Offshore Proceedings of the 5th International Conference on Renewable Energies Offshore*. Taylor & Francis, 2022.

[8] R. Bhaskaran and L. Collins. Introduction to cfd basics. *Cornell University-Sibley School of Mechanical and Aerospace Engineering*, pages 1–21, 2002.

[9] H. Bihs. Three-dimensional numerical modeling of local scouring in open channel flow. 2011.

[10] H. Bihs and A. Kamath. A combined level set/ghost cell immersed boundary representation for floating body simulations. *International Journal for Numerical Methods in Fluids*, 83(12):905–916, 2017.

[11] H. Bihs, A. Kamath, M. Alagan Chella, and C. Pakozdi. Complex geometry handling for a cartesian grid based solver. In *MekIT'17-Ninth national conference on Computational Mechanics*. International Center for Numerical Methods in Engineering (CIMNE), 2017.

[12] H. Bihs, A. Kamath, M. A. Chella, A. Aggarwal, and Ø. A. Arntsen. A new level set numerical wave tank with improved density interpolation for complex wave hydrodynamics. *Computers & Fluids*, 140:191–208, 2016.

[13] H. Bihs, A. Kamath, J. Z. Lu, and O. A. Arntsen. Simulation of floating bodies using a combined immersed boundary with the level set method in reef3d. In *MARINE VII: Proceedings of the VII International Conference on Computational Methods in Marine Engineering*, pages 921–931. CIMNE, 2017.

[14] H. Bihs, W. Wang, C. Pakozdi, and A. Kamath. Reef3d:: Fnpf—a flexible fully nonlinear potential flow solver. *Journal of Offshore Mechanics and Arctic Engineering*, 142(4), 2020.

[15] J. Boussinesq. *Essai sur la théorie des eaux courantes*. Impr. nationale, 1877.

[16] A. Calderer, S. Kang, and F. Sotiropoulos. Level set immersed boundary method for coupled simulation of air/water interaction with complex floating structures. *Journal of Computational Physics*, 277:201–227, 2014.

[17] P. M. Carrica, R. V. Wilson, R. W. Noack, and F. Stern. Ship motions using single-phase level set with dynamic overset grids. *Computers & fluids*, 36(9):1415–1433, 2007.

[18] C. CFD. Methods for grid generation in cfd simulations, 2023. `https://resources.system-analysis.cadence.com/blog/msa2021-methods-for-grid-generation-in-cfd-simulations` [Accessed: 07.05.2023].

[19] CFD-Online. Staggered grid, 2012. `https://www.cfd-online.com/Wiki/Staggered_grid` [Accessed: 29.04.2023].

[20] CFD-Online. Piso algorithm - pressure implicit with split operator, 2015. `https://www.cfd-online.com/Wiki/PISO_algorithm_-_Pressure_Implicit_with_Split_Operator` [Accessed: 29.04.2023].

[21] CFD-Online. Simple algorithm, 2016. `https://www.cfd-online.com/Wiki/SIMPLE_algorithm` [Accessed: 29.04.2023].

[22] CFD-Online. Two equation turbulence models, 2016. https://www.cfd-online.com/Wiki/Two_equation_turbulence_models#References [Accessed: 29.04.2023].

[23] Q. Cheng, X. Liu, H. S. Ji, K. C. Kim, and B. Yang. Aerodynamic analysis of a helical vertical axis wind turbine. *Energies*, 10(4):575, 2017.

[24] A. J. Chorin. Numerical solution of the navier-stokes equations. *Mathematics of computation*, 22(104):745–762, 1968.

[25] R. Courant, E. Isaacson, and M. Rees. On the solution of nonlinear hyperbolic differential equations by finite differences. *Communications on pure and applied mathematics*, 5(3):243–255, 1952.

[26] R. G. Dean and R. A. Dalrymple. *Advanced series on ocean engineering-volume 2: water wave mechanics for engineers and scientists*. World Scientific, 1991.

[27] L.-C. Dempwolff. Experimental and numerical investigation of a moored floating structure in waves. Master's thesis, NTNU, 2019.

[28] F. Denner and B. G. van Wachem. Tvd differencing on three-dimensional unstructured meshes with monotonicity-preserving correction of mesh skewness. *Journal of Computational Physics*, 298:466–479, 2015.

[29] P. Durbin. Limiters and wall treatments in applied turbulence modeling. *Fluid Dynamics Research*, 41(1):012203, 2009.

[30] A. El Kasmi and C. Masson. An extended k–ε model for turbulent flow through horizontal-axis wind turbines. *Journal of Wind Engineering and Industrial Aerodynamics*, 96(1):103–122, 2008.

[31] Equinor. Floating wind, 2023. https://www.equinor.com/energy/floating-wind [Accessed: 09.06.2023].

[32] M. D. Esteban, J. J. Diez, J. S. López, and V. Negro. Why offshore wind energy? *Renewable energy*, 36(2):444–450, 2011.

[33] E. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of computational physics*, 161(1):35–60, 2000.

[34] O. Faltinsen. *Sea loads on ships and offshore structures*, volume 1. Cambridge university press, 1993.

[35] J. H. Ferziger, M. Perić, and R. L. Street. *Computational methods for fluid dynamics*, volume 3. Springer, 2002.

[36] R. Fletcher. Conjugate gradient methods for indefinite systems. In *Numerical Analysis: Proceedings of the Dundee Conference on Numerical Analysis, 1975*, pages 73–89. Springer, 2006.

[37] P. Gaskell and A. Lau. Curvature-compensated convective transport: Smart, a new boundedness-preserving transport algorithm. *International Journal for numerical methods in fluids*, 8(6):617–641, 1988.

[38] P. M. Gebraad, F. W. Teeuwisse, J.-W. van Wingerden, P. A. Fleming, S. D. Ruben, J. R. Marden, and L. Y. Pao. A data-driven model for wind plant power optimization by yaw control. In *2014 American Control Conference*, pages 3128–3134. IEEE, 2014.

[39] M. Griebel, T. Dornseifer, and T. Neunhoeffer. *Numerical simulation in fluid dynamics: a practical introduction*. SIAM, 1998.

[40] A. Harten. High resolution schemes for hyperbolic conservation laws. *Journal of computational physics*, 135(2):260–278, 1997.

[41] M. R. Hestenes, E. Stiefel, et al. Methods of conjugate gradients for solving linear systems. *Journal of research of the National Bureau of Standards*, 49(6):409–436, 1952.

[42] P. Higuera, J. L. Lara, and I. J. Losada. Realistic wave generation and active wave absorption for navier–stokes models: Application to openfoam®. *Coastal Engineering*, 71:102–118, 2013.

[43] C. W. Hirt and B. D. Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of computational physics*, 39(1):201–225, 1981.

[44] L. H. Holthuijsen. *Waves in oceanic and coastal waters*. Cambridge university press, 2010.

[45] L. H. Holthuijsen. *Waves in oceanic and coastal waters*. Cambridge university press, 2010.

[46] M. Hossain. Mathematical modelling of vertical mixing in stratified channel flow. In *Proc. 2nd Int. Symp. on Stratified Flows, Trondheim, Norway, 1980*, 1980.

[47] U. IPCC. Special report: Global warming of 1.5° c—summary for policy makers, 2018.

[48] R. I. Issa. Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of computational physics*, 62(1):40–65, 1986.

[49] S. Itō. *Study of the transient heave oscillation of a floating cylinder.* PhD thesis, Massachusetts Institute of Technology, 1977.

[50] N. G. Jacobsen, D. R. Fuhrman, and J. Fredsøe. A wave generation toolbox for the open-source cfd library: Openfoam®. *International Journal for numerical methods in fluids*, 70(9):1073–1088, 2012.

[51] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted eno schemes. *Journal of computational physics*, 126(1):202–228, 1996.

[52] J. M. Journee and W. Massie. Introduction in offshore hydromechanics (ot3600). *TUDelft, Faculty of Marine Technology, Ship Hydromechanics Laboratory, Report No. 1267-K, Lecture Notes*, 2001.

[53] A. Kamath, M. Alagan Chella, H. Bihs, and Ø. A. Arntsen. Influence of the upstream cylinder and wave breaking point on the breaking wave forces on the downstream cylinder. In *Proceedings of the 12th International Conference on Computational Fluid Dynamics in the Oil & Gas, Metallurgical and Process Industries*. SINTEF Academic Press, 2017.

[54] A. Kamath, G. Fleit, and H. Bihs. Investigation of free surface turbulence damping in rans simulations for complex free surface flows. *Water*, 11(3):456, 2019.

[55] A. M. Kamath. Calculation of wave forces on structures using reef3d. Master's thesis, Institutt for bygg, anlegg og transport, 2012.

[56] A. M. Kamath et al. Wave forces on structures using reef3d. Master's thesis, 2012.

[57] K. Kan, Z. Yang, P. Lyu, Y. Zheng, and L. Shen. Numerical study of turbulent flow past a rotating axial-flow pump based on a level-set immersed boundary method. *Renewable Energy*, 168:960–971, 2021.

[58] E. Katsidoniotaki, Z. Shahroozi, C. Eskilsson, J. Palm, J. Engström, and M. Göteman. Validation of a cfd model for wave energy system dynamics in extreme waves. *Ocean Engineering*, 268:113320, 2023.

[59] H. Krogstad and Ø. Arntsen. Ocean surface waves–linear wave description and theory. *Compendium for TBA4265: Marine Physical Environment*, 2003.

[60] R. Kumar, K. Raahemifar, and A. S. Fung. A critical review of vertical axis wind turbines for urban applications. *Renewable and Sustainable Energy Reviews*, 89:281–291, 2018.

[61] J. Larsen and H. Dancy. Open boundaries in short wave simulations—a new approach. *Coastal engineering*, 7(3):285–297, 1983.

[62] B. E. Launder. Turbulence modelling for cfd. by d. c. wilcox. dcw industries inc., 1993. 460pp. 75. *Journal of Fluid Mechanics*, 289:406–407, 1995.

[63] B. E. Launder and D. B. Spalding. The numerical computation of turbulent flows. In *Numerical prediction of flow, heat transfer, turbulence and combustion*, pages 96–116. Elsevier, 1983.

[64] J.-M. Lee, C.-J. Lee, Y.-S. Kim, G.-G. Choi, and J.-M. Lew. Determination of global ice loads on the ship using the measured full-scale motion data. *International Journal of Naval Architecture and Ocean Engineering*, 8(4):301–311, 2016.

[65] M. Lee, G. Park, C. Park, and C. Kim. Improvement of grid independence test for computational fluid dynamics model of building based on grid resolution. *Advances in Civil Engineering*, 2020:1–11, 2020.

[66] B. P. Leonard. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Computer methods in applied mechanics and engineering*, 19(1):59–98, 1979.

[67] Y. Liu, Q. Xiao, A. Incecik, C. Peyrard, and D. Wan. Establishing a fully coupled cfd analysis tool for floating offshore wind turbines. *Renewable Energy*, 112:280–301, 2017.

[68] J. Z. Lu. Simulation of floating bodies in waves. Master's thesis, [NTNU], 2017.

[69] M. Maizi, M. Mohamed, R. Dizene, and M. Mihoubi. Noise reduction of a horizontal wind turbine using different blade shapes. *Renewable Energy*, 117:242–256, 2018.

[70] T. Martin. A new cfd-based framework for modelling the interaction of open ocean aquaculture structures and complex free surface hydrodynamics. 2021.

[71] T. Martin, A. Kamath, and H. Bihs. Accurate modeling of the interaction of constrained floating structures and complex free surfaces using a new quasistatic mooring model. *International Journal for Numerical Methods in Fluids*, 93(2):504–526, 2021.

[72] S. Mayer, A. Garapon, and L. S. Sørensen. A fractional step method for unsteady free-surface flow with applications to non-linear wave dynamics. *International Journal for Numerical Methods in Fluids*, 28(2):293–315, 1998.

[73] J. Morison, J. W. Johnson, and S. A. Schaaf. The force exerted by surface waves on piles. *Journal of Petroleum Technology*, 2(05):149–154, 1950.

[74] D. Naot and W. Rodi. Calculation of secondary currents in channel flow. *Journal of the Hydraulics Division*, 108(8):948–968, 1982.

[75] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988.

[76] A. I. Osman, L. Chen, M. Yang, G. Msigwa, M. Farghali, S. Fawzy, D. W. Rooney, and P.-S. Yap. Cost, environmental impact, and resilience of renewable energy under a changing climate: a review. *Environmental Chemistry Letters*, 21(2):741–764, 2023.

[77] A. Otter, J. Murphy, V. Pakrashi, A. Robertson, and C. Desmond. A review of modelling techniques for floating offshore wind turbines. *Wind Energy*, 25(5):831–857, 2022.

[78] J. Palm, C. Eskilsson, and L. Bergdahl. An hp-adaptive discontinuous galerkin method for modelling snap loads in mooring cables. *Ocean Engineering*, 144:266–276, 2017.

[79] F. Papi and A. Bianchini. Technical challenges in floating offshore wind turbine upscaling: A critical analysis based on the nrel 5 mw and iea 15 mw reference turbines. *Renewable and Sustainable Energy Reviews*, 162:112489, 2022.

[80] B. Pardis, H. Mahdyar, and M. Riazi. Streamline based modeling and simulation of carbonated water injection.

[81] S. Park, M. A. Lackner, P. Pourazarm, A. Rodríguez Tsouroukdissian, and J. Cross-Whiter. An investigation on the impacts of passive and semiactive structural control on a fixed bottom and a floating offshore wind turbine. *Wind Energy*, 22(11):1451–1471, 2019.

[82] S. V. Patankar and D. B. Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. In *Numerical prediction of flow, heat transfer, turbulence and combustion*, pages 54–73. Elsevier, 1983.

[83] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. A pde-based fast local level set method. *Journal of computational physics*, 155(2):410–438, 1999.

[84] C. S. Peskin. The immersed boundary method. *Acta numerica*, 11:479–517, 2002.

[85] C. P. Poole and J. L. Safko. *Classical mechanics*. Addison Wesley, 2001.

[86] S. B. Pope and S. B. Pope. *Turbulent flows*. Cambridge university press, 2000.

[87] B. Ramaswamy, M. Kawahara, and T. Nakayama. Lagrangian finite element method for the analysis of two-dimensional sloshing problems. *International Journal for Numerical Methods in Fluids*, 6(9):659–670, 1986.

[88] B. E. Rapp. Chapter 9 - fluids. In B. E. Rapp, editor, *Microfluidics: Modelling, Mechanics and Mathematics*, Micro and Nano Technologies, pages 243–263. Elsevier, Oxford, 2017.

[89] B. Ren, M. He, P. Dong, and H. Wen. Nonlinear simulations of wave-induced motions of a freely floating body using wcsph method. *Applied Ocean Research*, 50:1–12, 2015.

[90] A. Rezaeiha, I. Kalkman, and B. Blocken. Cfd simulation of a vertical axis wind turbine operating at a moderate tip speed ratio: Guidelines for minimum domain size and azimuthal increment. *Renewable energy*, 107:373–385, 2017.

[91] L. Roald, J. Jonkman, A. Robertson, and N. Chokani. The effect of second-order hydrodynamics on floating offshore wind turbines. *Energy Procedia*, 35:253–264, 2013.

[92] P. C. Rocha, H. B. Rocha, F. M. Carneiro, M. V. Da Silva, and A. V. Bueno. k–ω sst (shear stress transport) turbulence model calibration: A case study on a small scale horizontal axis wind turbine. *Energy*, 65:412–418, 2014.

[93] W. Rodi, G. Constantinescu, and T. Stoesser. Large-eddy simulation in hydraulics. 2013.

[94] J. Sanders, J. E. Dolbow, P. J. Mucha, and T. A. Laursen. A new method for simulating rigid body motion in incompressible two-phase flow. *International Journal for Numerical Methods in Fluids*, 67(6):713–732, 2011.

[95] H. A. Schäffer and G. Klopman. Review of multidirectional active wave absorption methods. *Journal of waterway, port, coastal, and ocean engineering*, 126(2):88–97, 2000.

[96] H. Schlichting and K. Gersten. *Boundary-layer theory*. springer, 2016.

[97] R. Shivarama and E. P. Fahrenthold. Hamilton's equations with euler parameters for rigid body dynamics modeling. *J. Dyn. Sys., Meas., Control*, 126(1):124–130, 2004.

[98] A. Shourangiz-Haghighi, M. A. Haghnegahdar, L. Wang, M. Mussetta, A. Kolios, and M. Lander. State of the art in the optimisation of wind turbine performance using cfd. *Archives of Computational Methods in Engineering*, 27:413–431, 2020.

[99] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of computational physics*, 77(2):439–471, 1988.

[100] J. Smagorinsky. General circulation experiments with the primitive equations: I. the basic experiment. *Monthly weather review*, 91(3):99–164, 1963.

[101] N. Sorensen and M. Hansen. Rotor performance predictions using a navier-stokes method. In *1998 ASME wind energy symposium*, page 25, 1998.

[102] C. Souza, P. Berthelsen, L. Eliassen, E. Bachynski, E. Engebretsen, and H. Haslum. Definition of the ino windmoor 12 mw base case floating wind turbine. *Tech. Rep. OC2020 A-044*, 2021.

[103] R. J. Stevens, D. F. Gayme, and C. Meneveau. Large eddy simulation studies of the effects of alignment and wind farm length. *Journal of Renewable and Sustainable Energy*, 6(2):023105, 2014.

[104] G. G. Stokes. On the theory of oscillatory waves. *Trans. Cam. Philos. Soc.*, 8:441–455, 1847.

[105] H. L. Stone. Iterative solution of implicit approximations of multidimensional partial differential equations. *SIAM Journal on Numerical Analysis*, 5(3):530–558, 1968.

[106] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational physics*, 114(1):146–159, 1994.

[107] N. Takashi. Ale finite element computations of fluid-structure interaction problems. *Computer methods in applied mechanics and engineering*, 112(1-4):291–308, 1994.

[108] S. Tanaka and K. Kashiyama. Ale finite element method for fsi problems with free surface using mesh re-generation method based on background mesh. *International Journal of Computational Fluid Dynamics*, 20(3-4):229–236, 2006.

[109] J. Thé and H. Yu. A critical review on the simulations of wind turbine aerodynamics focusing on hybrid rans-les methods. *Energy*, 138:257–289, 2017.

[110] M. Thys, C. Souza, T. Sauder, N. Fonseca, P. A. Berthelsen, E. Engebretsen, and H. Haslum. Experimental investigation of the coupling between aero-and hydrodynamical loads on a 12 mw semi-submersible floating wind turbine. In *International Conference on Offshore Mechanics and Arctic Engineering*, volume 85192, page V009T09A030. American Society of Mechanical Engineers, 2021.

[111] M. Thys, C. Souza, T. Sauder, N. Fonseca, P. A. Berthelsen, E. Engebretsen, and H. Haslum. Experimental investigation of the coupling between aero-and hydrodynamical loads on a 12 mw semi-submersible floating wind turbine, proceedings of the asme 2021 40th international conference on ocean, offshore and arctic engineering. *Applied Ocean Research*, Proceedings of the ASME 2021 40th International Conference on Ocean, Offshore and Arctic Engineering, 2021.

[112] L. J. Timmermans, P. D. Minev, and F. N. Van De Vosse. An approximate projection scheme for incompressible flow using spectral elements. *International journal for numerical methods in fluids*, 22(7):673–688, 1996.

[113] M. Uhlmann. An immersed boundary method with direct forcing for the simulation of particulate flows. *Journal of computational physics*, 209(2):448–476, 2005.

[114] H. A. Van der Vorst. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM Journal on scientific and Statistical Computing*, 13(2):631–644, 1992.

[115] G. Vaz, F. Jaouen, and M. Hoekstra. Free-surface viscous flow computations: Validation of urans code fresco. In *International Conference on Offshore Mechanics and Arctic Engineering*, volume 43451, pages 425–437, 2009.

[116] H. K. Versteeg and W. Malalasekera. *An introduction to computational fluid dynamics: the finite volume method*. Pearson education, 2007.

[117] E. Walhorn, A. Kölke, B. Hübner, and D. Dinkler. Fluid–structure coupling within a monolithic model involving free surface flows. *Computers & structures*, 83(25-26):2100–2111, 2005.

[118] S. Wallin and A. V. Johansson. An explicit algebraic reynolds stress model for incompressible and compressible turbulent flows. *Journal of fluid mechanics*, 403:89–132, 2000.

[119] W. Wang, A. Kamath, T. Martin, C. Pákozdi, and H. Bihs. A comparison of different wave modelling techniques in an open-source hydrodynamic framework. *Journal of Marine Science and Engineering*, 8(7):526, 2020.

[120] W. Wang, T. Martin, A. Kamath, and H. Bihs. An improved depth-averaged nonhydrostatic shallow water model with quadratic pressure approximation. *International Journal for Numerical Methods in Fluids*, 92(8):803–824, 2020.

[121] W. Wang, C. Pákozdi, A. Kamath, and H. Bihs. A fully nonlinear potential flow wave modelling procedure for simulations of offshore sea states with various wave breaking scenarios. *Applied Ocean Research*, 117:102898, 2021.

[122] W. Wang, C. Pákozdi, A. Kamath, T. Martin, and H. Bihs. Hydrodynamic coupling of viscous and nonviscous numerical wave solutions within the open-source hydrodynamics framework reef3d. *Journal of Offshore Mechanics and Arctic Engineering*, 144(4):041903, 2022.

[123] W. Wang, C. Pákozdi, A. Kamath, S. Fouques, and H. Bihs. A flexible fully nonlinear potential flow model for wave propagation over the complex topography of the norwegian coast. *Applied Ocean Research*, 122:103103, 2022.

[124] X. Wang and J. Li. Parametric study of hybrid monopile foundation for offshore wind turbines in cohesionless soil. *Ocean Engineering*, 218:108172, 2020.

[125] C. V. Weiss, R. Guanche, B. Ondiviela, O. F. Castellanos, and J. Juanes. Marine renewable energy potential: A global perspective for offshore wind and wave exploitation. *Energy conversion and management*, 177:43–54, 2018.

[126] D. C. Wilcox et al. *Turbulence modeling for CFD*, volume 2. DCW industries La Canada, CA, 1998.

[127] X. Wu. Inflow turbulence generation methods. *Annual Review of Fluid Mechanics*, 49:23–49, 2017.

[128] Y.-T. Wu and F. Porté-Agel. Modeling turbine wakes and power losses within a wind farm using les: An application to the horns rev offshore wind farm. *Renewable Energy*, 75:945–955, 2015.

[129] Z. Wu, Y. Cao, S. Nie, and Y. Yang. Effects of rain on vertical axis wind turbine performance. *Journal of Wind Engineering and Industrial Aerodynamics*, 170:128–140, 2017.

[130] J. Yang and F. Stern. Sharp interface immersed-boundary/level-set method for wave–body interactions. *Journal of Computational Physics*, 228(17):6590–6616, 2009.

[131] J. Yang and F. Stern. Robust and efficient setup procedure for complex triangulations in immersed boundary simulations. *Journal of Fluids Engineering*, 135(10), 2013.

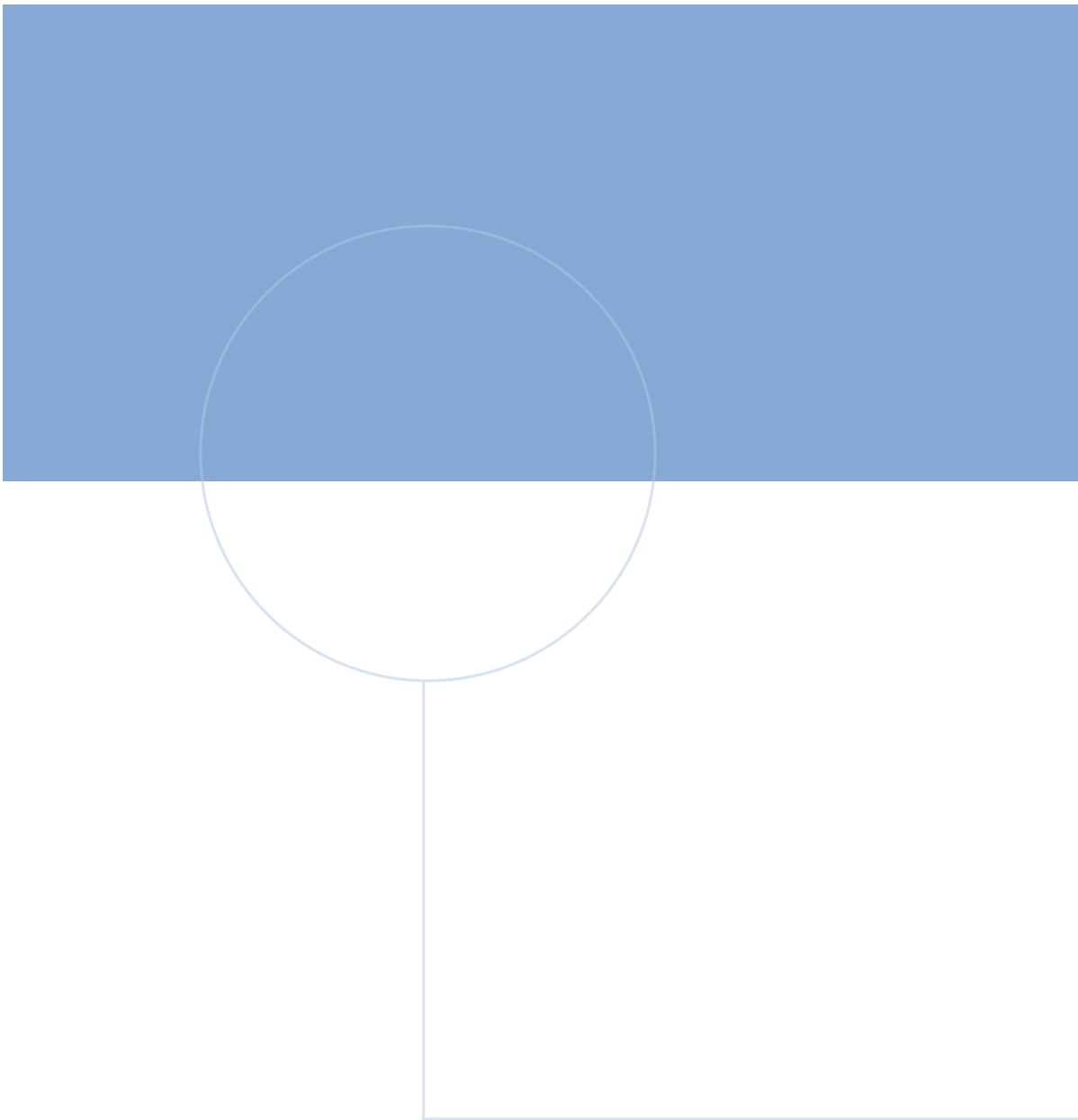
[132] U. M. Yang et al. Boomeramg: A parallel algebraic multigrid solver and preconditioner. *Applied Numerical Mathematics*, 41(1):155–177, 2002.

[133] C. W. Zheng, C. Y. Li, J. Pan, M. Y. Liu, and L. L. Xia. An overview of global ocean wind energy resource evaluations. *Renewable and Sustainable Energy Reviews*, 53:1240–1251, 2016.

[134] C.-w. Zheng, Z.-n. Xiao, Y.-h. Peng, C.-y. Li, and Z.-b. Du. Rezoning global offshore wind energy resources. *Renewable Energy*, 129:1–11, 2018.

[135] J. Zheng Lu. Simulation of floating bodies in waves using reef3d. Master's thesis, NTNU, 2017.

[136] Y. Zhiyin. Large-eddy simulation: Past, present and the future. *Chinese journal of Aeronautics*, 28(1):11–24, 2015.



NTNU

Norwegian University of
Science and Technology