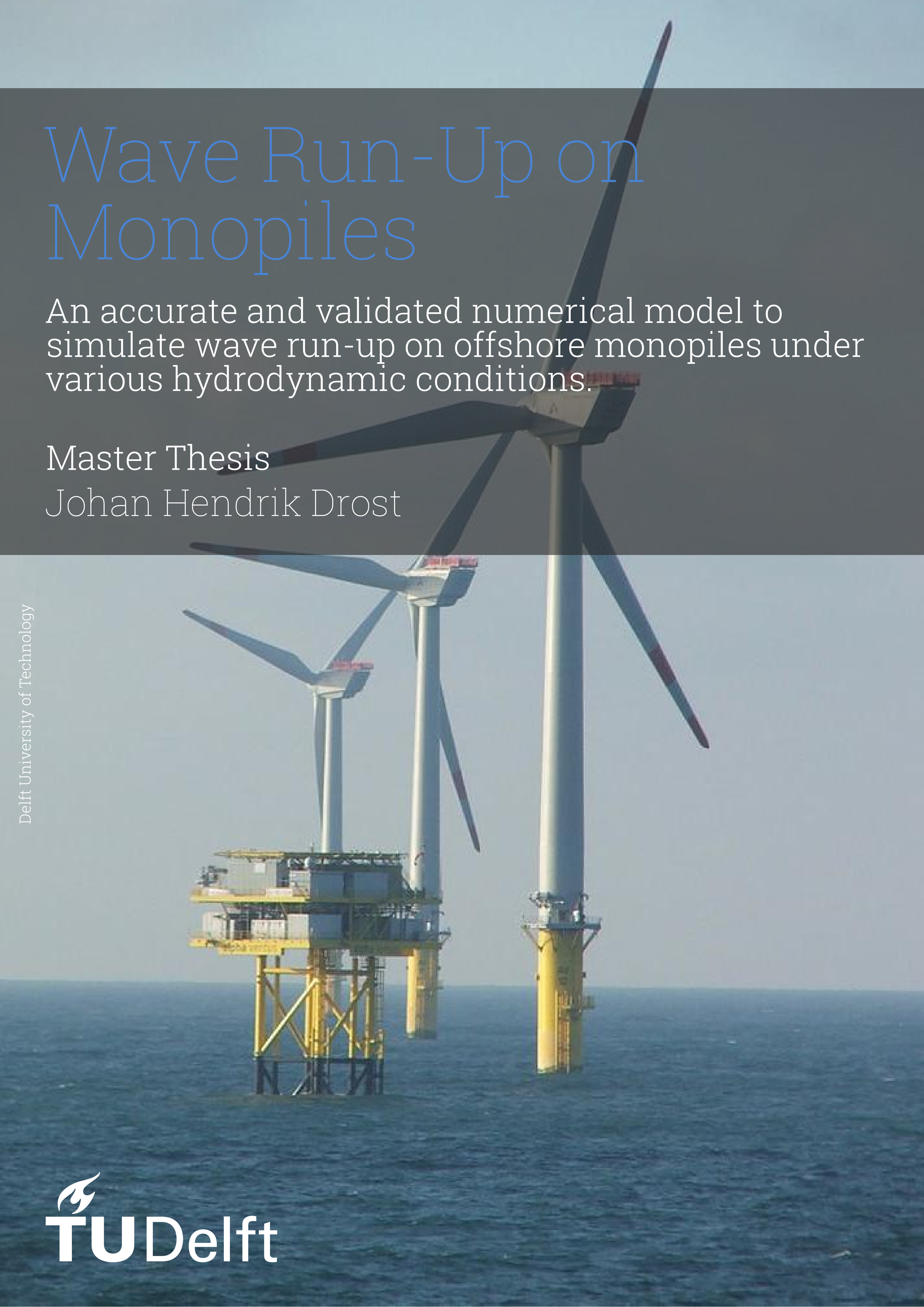


Wave Run-Up on Monopiles

An accurate and validated numerical model to simulate wave run-up on offshore monopiles under various hydrodynamic conditions.

Master Thesis

Johan Hendrik Drost

Delft University of Technology

Wave Run-Up on Monopiles

An accurate and validated numerical model to simulate wave run-up on offshore monopiles under various hydrodynamic conditions.

by

Johan Hendrik Drost

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Monday April 28, 2025 at 10:00 AM.

Student number:	5174112	
Project duration:	September, 2024 – April, 2025	
Thesis committee:	Dr. ir. O. Colomés Gené,	TU Delft, Chairman & Examiner
	Dr. ir. P. Mares Nasarre,	TU Delft, Supervisor & Examiner
	Dr. Ir. S. Agarwal,	TU Delft, Supervisor
	Ir. M. Vergassola,	TU Delft, Supervisor
	Ir. J. Modderman	TU Delft, Supervisor

Cover: Wind turbines and electrical substation of Alpha Ventus Offshore
Wind Farm in the North Sea (Wikipedia, 2009).

An electronic version of this thesis is available at http://repository.tudelft.nl/.
The model setups used in this thesis are publicly available via the Github repository:
https://github.com/hiddded/MSc-thesis-wave-run-up-on-monopiles

Preface

This master's thesis marks the end of my Master's degree in Civil Engineering at the Technical University of Delft, with a specialization in Hydraulic and Offshore Structures. When I began my Bachelor's in 2019 I never imagined that one day I'd spend almost half a year building and refining a numerical model. At the time, it sounded far from what I thought would be possible and interesting. But now, at the end of this MSc thesis, I can honestly say I've really enjoyed the process, despite all the errors, failed simulations and mysteriously disappearing waves. Diving into the complexities of numerical modeling and understanding wave-structure interactions has been both challenging and genuinely rewarding.

First and foremost, I would like to express my gratitude to Oriol for his supervision during my MSc thesis, despite his busy schedule. I truly appreciate his guidance and support over the past year, not only during this thesis but also through his courses, which got me interested in this field. I've gained valuable knowledge and insights from him and I'm grateful for that.

A special thanks to Shagun for his time, quick responses to my endless questions and his help with debugging the model. I still remember watching him navigate his computer using only his keyboard during the course Floating and Submerged Structures and our first meetings. I like to think that now I can (kind of) keep up with him and understand what he is doing.

I also want to thank Patricia for her valuable and detailed feedback and her time as a member of my thesis committee. I am also grateful to Marco and Jan, for their feedback, insights and time they dedicated to my work.

To my family, thank you for your constant support throughout my academic journey. You've always motivated me to follow my ambitions, and I'm incredibly grateful for that. And finally, I want to thank my friends for their mental support. I apologize for making you listen to endless stories about waves hitting monopiles, the complexities of numerical modeling and my frustrations about the errors. I promise I'll never talk about another wave hitting a monopile!

Johan Hendrik Drost
Delft, April 2025

Abstract

The rapid expansion of offshore wind energy, particularly into intermediate depths and more non-linear wave environments, has increased the need for accurate prediction of wave-structure interactions, especially wave run-up and the associated hydrodynamic forces on monopile foundations. These predictions are essential for ensuring the structural integrity, safety, and cost-efficiency of offshore wind turbines. Existing semi-empirical formulas are limited in their applicability, often overestimating run-up heights and failing to capture the complex non-linear interactions that occur under steep ($S_o > 0.03$) or near-breaking wave conditions typical of intermediate water depths.

This thesis presents a numerical model based on Computational Fluid Dynamics (CFD), developed using *OpenFOAM* and the *waves2Foam* toolbox to predict wave run-up and the associated hydrodynamic forces on monopiles under varying hydrodynamic conditions. The model was validated against experimental data from de Vos et al. (2007), achieving excellent agreement with less than 5% error in peak wave run-up predictions. Following validation, the model was applied to assess the influence of scale effects, turbulence, monopile geometry and key wave characteristics on run-up behavior and force magnitude.

The results confirm that scale effects on surface elevation and wave run-up are limited and hydrodynamic forces, while viscous forces in the vertical direction show some sensitivity. Steep, high-energy waves, near the breaking thresholds ($S \gtrsim 0.04$ to 0.05), produce significantly larger run-up heights and slamming forces, highlighting their importance in design considerations. Additionally, the geometry of the monopile impacts localized flow patterns and wave wrapping, while turbulence contributes to increased lateral and viscous forces. However, its effect on the critical run-up at the front of the structure remains minimal.

By integrating turbulence modeling, scaling, and mesh optimization, this study establishes a robust, scalable and accurate numerical framework for simulating complex wave-structure interactions. The model demonstrates a significant improvement over existing empirical approaches, particularly in intermediate water depths where non-linearity, shoaling and breaking are present.

While the model demonstrated excellent predictive capabilities, several areas for improvement remain. These include local mesh refinement around the monopile, the use of higher-order discretization schemes to better capture breaking waves and more extensive validation using recent experimental datasets. However, these improvements come at a significant computational cost. Given the already high level of accuracy in capturing peak wave run-up (>95%), the current model offers a practical balance between precision and efficiency. Future efforts should therefore focus less on fine-tuning and more on applying the model to investigate irregular wave conditions and additional hydrodynamic parameters. Future researchers with greater computational resources may consider these improvements, but should weigh the trade-off between added accuracy and computational expense.

The model configurations are publicly available to support future research and industry adoption: https://github.com/hiddded/MSc-thesis-wave-run-up-on-monopiles

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

Abbreviation	Definition
CFD	Computational Fluid Dynamics
DNS	Direct Numerical Simulation
Fr	Froude number
FVM	Finite Volume Method
GW	Gigawatts
KC	Keulegan-Carpenter number
LES	Large Eddy Simulation
LSM	Least Squares Method
MSL	Mean Sea Level
RANS	Reynolds-Averaged Navier-Stokes
Re	Reynolds number
URANS	Unsteady Reynolds-Averaged Navier-Stokes
VOF	Volume of Fluid

Symbols

Symbol	Definition	Unit
A	Cross-sectional area	[m^2]
C_D	Drag coefficient	[-]
C_M	Inertia coefficient	[-]
C_o	Courant number	[-]
D	Diameter of the structure monopile	[m]
d	Water depth	[m]
e	Turbulent dissipation rate	[m^2s^{-3}]
F	Force	[N]
F_D	Drag force component	[N]
F_I	Inertia force component	[N]
f	Face of a control volume	[-]
H	Wave height	[m]
H_{m0}	Spectral significant wave height	[m]
H_s	Significant wave height	[m]
$H_{1/3}$	Average height of the highest one-third of waves	[m]
$H_{2\%}$	2% Excess wave height	[m]
k	Wave number	[m^{-1}]
L	Wavelength of the incoming wave	[m]
m_0	zeroth moment	[m^2D]
p	Pressure	[Pa]
p_{rgh}	Hydrostatic pressure	[Pa]
R_u	Wave run-up height	[m]
$R_{u,2\%}$	2% Excess run-up height	[m]
R_{max}	Maximum wave run-up height	[m]
S	Wetted surface area	[m^2]
S_{op}	Wave steepness	[-]
T	Wave period	[s]
T_p	Peak wave period	[s]
t	Time	[s]

Symbol	Definition	Unit
U	Flow velocity	[m/s]
u	Water particle velocity at the wave crest	[m/s]
$\dot{u}$	Water particle acceleration at the wave crest	[m/s^2]
V	Volume	[m^3]
Z_i	Incident wave complex amplitude	[m]
Z_r	Reflected wave complex amplitude	[m]
α	Volume fraction	[-]
η_{max}	Maximum surface elevation of the wave	[m]
λ	First order wave amplitude	[m]
μ	Dynamic viscosity	[Nsm^-2]
ν	Kinematic viscosity	[m^2/s]
ρ	Density	[kg/m^3]
σ	Radian wave frequency	[s^{-1}]
τ	Viscous stress	[Pa]
ϕ	Velocity potential	[ms^{-1}]
ω	Angular wave frequency	[rad/s]

1 Introduction

The demand for renewable energy is growing rapidly, driven by the urgent need to mitigate climate change and transition away from fossil fuels. The European Union has committed to sourcing at least 42.5% of its energy from renewable resources by 2030 (Commission, 2025) and offshore wind is set to play a pivotal role in meeting this target. The Netherlands experienced a 55% increase in offshore wind energy production in 2023, now generating 4 gigawatts (GW). This is just the beginning, as the Dutch government aims to increase offshore wind capacity to 21 GW by 2032, enough to supply 16% of national energy demand and 75% of the current electricity consumption (Rijksoverheid, 2025).

Achieving this ambitious target will require a significant expansion of the offshore wind infrastructure, including the installation of many new wind farms (Agency, 2025). The monopile has been widely applied at intermediate water depths of 25-45 m and occupies approximately 80% of the wind turbine foundations (Zhu et al., 2022). In these intermediate water depths, the hydrodynamic conditions are highly variable, influenced by wave height, steepness, period and depth. As a result, the wave environment in these regions consists of a wide range of wave types, from linear, small-amplitude waves to steep, non-linear and breaking waves. With the growing ambition to expand offshore wind farms and move them into deeper ($\approx$ 35- 55 m) and more energetic environments, this variability in wave environment highlights the importance of understanding the wave-structure interactions, especially wave run-up and slamming forces (Murphy et al., 2018).

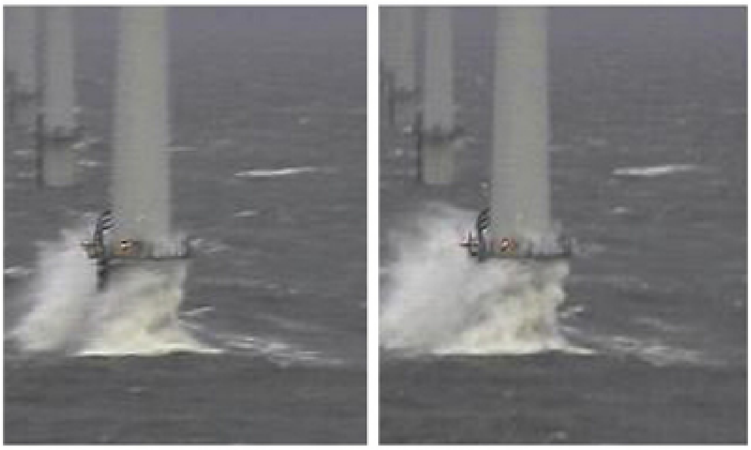


Figure 1.1: Wave run-up at Horns Reef 1 for $H_s \approx 2.5$ m, while the platform level is 9.0 m above MWL. $H_{s,design}$ = 5.3 m (Lykke Andersen et al., 2011).

Wave run-up refers to the vertical extent to which waves climb up the structure beyond the still water level (Fig. 1.1). Accurate prediction of wave run-up is essential for the correct elevation of access platforms, boat landings and maintenance decks. Underestimating run-up can lead to unsafe working conditions while overestimating results in unnecessarily high platforms and increased installation costs. In particular, non-linear wave effects, which occur in intermediate water depths, can significantly increase the run-up height beyond what linear wave theory would suggest. Slamming forces refer to the highly transient, localized impact loads caused by waves striking the structure. These forces can reach magnitudes 8 to 9 times higher than those associated with linear non-breaking waves (Fig. 1.3) and pose a serious risk to the structural integrity and fatigue life of monopiles (Zhu et al., 2022). Therefore, predicting wave run-up and slamming forces accurately are essential for safe and cost-effective design.

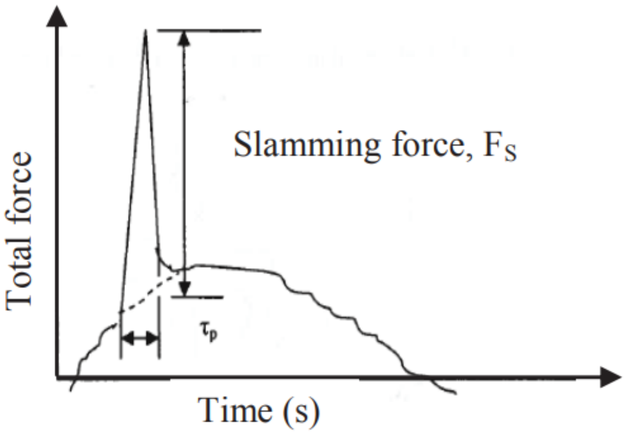


Figure 1.2: Magnitude of slamming force (F_s) compared to the total force (Chella et al., 2012).

The need for precise wave run-up and load predictions has become even more critical with the development of innovative foundation connection methods such as the slip joint monopile. The slip joint technology represents an alternative connection based on the overlap between two conical pipes with a circular cross-section. The reliability of the connection depends on the proper transfer of forces from the upper cone to the lower cone (Mojto & Cabboi, 2024). Since this connection method does not require bolts or welds, it offers a significant reduction in installation time and complexity. A notable example underscoring the importance of accurately predicting wave run-up and forces occurred in 2009, when 600 offshore wind turbines exhibited structural failures (Mojto & Cabboi, 2024). These failures, characterized by cracks and downward slip, emphasized the critical role that wave run-up and loads play in the design and long-term integrity of offshore structures. Such cases have driven the industry to not only consider alternative connection designs like the slip-joint, but mainly to prioritize precise modeling of wave-structure interactions to ensure safety and durability.

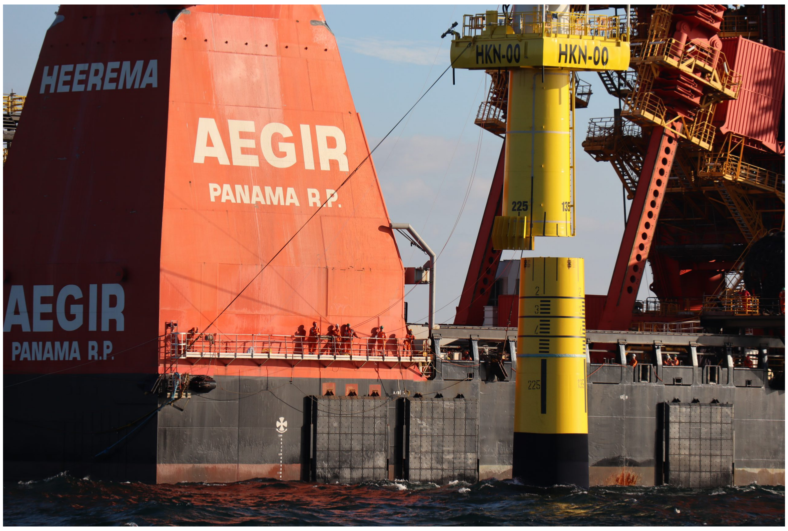


Figure 1.3: Installation of a slip joint monopile (DOT, 2025).

To address these design challenges, researchers have proposed various semi-empirical and analytical formulae to estimate wave run-up and slamming forces. These formulae are often based on experimental data combined with theoretical principles. However, experiments have a limited range of application, are time-consuming and often expensive. Most existing semi-empirical and analytical models are only valid within limited parameter ranges and often lack accuracy due to the highly non-linear and complex nature of wave–structure interactions. In addition, such formulae typically do not account for turbulence effects, which can play a significant role in offshore environments. As a result, these formulae may fall short when applied to varying hydrodynamic conditions. Numerical modeling and in particular Computational Fluid Dynamics (CFD), might offer a more efficient and flexible solution. CFD has the potential to capture a broader range of parameters and provide deeper insights into wave run-up and forces. Furthermore, CFD simulations can incorporate turbulence modeling, which may be important in scenarios involving high Reynolds number flows, wave breaking, and flow separation. These turbulence-related phenomena are particularly relevant in intermediate water depths around monopile foundations, where the interaction between waves and the cylindrical structure often results in localized turbulent flow. Therefore, this thesis will explore whether CFD is a suitable and effective method for predicting wave run-up and forces on offshore foundations.

This MSc thesis is structured as follows: Chapter 1 starts with the current state of the art, the knowledge gap, research questions and the method used to answer these research questions. Then Chapter 2 follows up with some theoretical background of the governing equations, wave theories, forces, turbulence and scaling. Chapter 3 contains the model configuration, including a full description of the model, tools and techniques used. Chapter 4 presents the results of the study, followed by a discussion in Chapter 5. Finally, Chapters 6 and 7 contain respectively the conclusion and recommendations for further work.

1.1. State of the Art in Wave Run-Up Modeling

Many considerable studies have been dedicated to the prediction of wave run-up on monopiles and offshore structures, including analytical, semi-empirical and numerical methods. Each of these methods has the goal of fully capturing the complexity of wave-structure interaction. However, the accuracy and applicability of each of these methods depends on the wave conditions, as most formulae are developed for specific cases such as non-breaking or small-amplitude waves. As a result, their reliability decreases when applied outside these conditions. In the following sections, the analytical and semi-empirical methods developed over time will be presented and discussed to provide context for the evolution of wave run-up prediction techniques.

1.1.1. Diffraction theory

Diffraction theory is one of the classical approaches used to calculate wave-structure interactions, particularly for large, fixed structures, such as monopiles. It is based on linear wave theory, which solves the potential flow around structures of arbitrary shape. However, its validity is limited to cases where the wave steepness (wave height to wavelength ratio) is small, typically when $\frac{H}{L} \lesssim 0.02$, meaning non-linear effects can be neglected. The approximate result for run-up on the up-wave side of a circular cylinder derived from linear diffraction theory is (MacCamy & Fuchs, 1954):

$$\frac{R_u}{\eta_{max}} = \left[1 + \left(\frac{2\pi D}{L}\right)^2\right]^{1/2} \tag{1.1}$$

- R_u: Wave run-up height $[m]$
- η_{max}: Maximum surface elevation of the wave $[m]$
- D: Diameter of the structure monopile $[m]$
- L: Wavelength of the incoming wave $[m]$

Diffraction theory is, in addition to being limited to small wave steepness, only applicable when $\frac{D}{L} > 0.2$, meaning the structure is large relative to the wavelength. In this case, the structure significantly disturbs the incoming wave field, causing the amount of scattered wave energy, the portion of energy redirected due to the interaction with the structure, to become non-negligible. As a result, this scattering effect must be explicitly accounted for in the analysis. Several authors (Kim & Yue, 1990; Kriebel, 1990; Stansberg et al., 2005) extended the diffraction theory to the second order and demonstrated that non-linear contributions can have a significant influence on wave-structure interactions. This implies that simple extrapolation from linear diffraction theory is often insufficient. They also found that the second order diffraction theory compares reasonably well for many cases, although this is only the case for moderate steep waves ($0.02 \lesssim \frac{H}{L} \lesssim 0.05$) since higher steepnesses ($\frac{H}{L} > 0.05$) introduce significant non-linear contributions.

1.1.2. Velocity stagnation head theory

Hallermeier (1976) proposed a formula for the run-up by considering the stagnation head of the wave crest at the point of impact with the cylinder. It assumes that the water particles at the wave crest are forced to convert their kinetic energy entirely into potential energy. This conversion causes the water to rise along the structure to a height corresponding to the wave run-up, which is given by $\frac{u^2}{2g}$. Therefore, the run-up is predicted to be:

$$R_u = \eta_{max} + \frac{u^2}{2g} \tag{1.2}$$

- u: Water particle velocity at the wave crest η_{max} $[m]$

Niedzwecki and Duggal (1992) performed small-scale experiments on circular cylinders under regular and irregular waves. They again found that linear diffraction theory underestimates the run-up for all but very low wave steepness. When applying the wave stagnation head theory, using linear theory for wave kinematics, Niedzwecki and Duggal (1992) found that the wave run-up from Hallermeier (1976) was under-predicted. They employed a semi-empirical variation of the formula using a coefficient $m = 6.83$.

$$R_u = \frac{H}{2} + m\frac{u^2}{2g} \tag{1.3}$$

- H: Wave height $[m]$

de Vos et al. (2007) performed new experiments, for wave steepnesses ranging from 0.02 - 0.12 on run-up to develop a new formula for both regular and irregular waves. They reviewed the applicable theories and previous experimental studies. By using Niedzwecki and Duggal (1992) theory, it was found that it both overestimates the run-up for smaller wave heights and underestimates it for large wave heights. Furthermore, it was not possible to get a good estimate of the run-up by adjusting the parameter m to fit the data. Therefore, the second-order Stokes theory was used to calculate the wave kinematics. These equations (Eq. 1.4 and 1.5) can then be filled into Eq. 1.2 to calculate the run-up (de Vos et al., 2007).

$$\eta_{max} = \frac{H}{2} + k\frac{H^2}{8}\frac{\cosh(kd)}{\sinh^3(kd)}\left(2 + \cosh(2kd)\right) \tag{1.4}$$

$$u = \frac{Hgk\cosh(k(\eta_{max}+d))}{2\omega\cosh(kd)} + \frac{3}{4}\frac{kH^2}{4\omega}\frac{\cosh(2k(\eta_{max}+d))}{\sinh^4(kd)} \tag{1.5}$$

- k: Wave number ($k = \frac{2\pi}{L}$) $[m^{-1}]$
- d: Water depth $[m]$
- ω: Angular wave frequency ($\omega = \frac{2\pi}{T}$) $[rad/s]$

For irregular waves, de Vos et al. (2007) introduced the 2% excess run-up height $R_{u,2\%}$, which represents the run-up height exceeded by only 2% of the incoming waves. They found that $R_{\text{max}} = 1.23R_{u,2\%}$, which closely matches the value reported by Mase et al. (2001), who found $R_{\text{max}} = 1.22R_{u,2\%}$. Therefore, only $R_{u,2\%}$ was considered in their study of irregular waves. The second-order Stokes theory was applied to calculate the surface elevation and horizontal particle velocity at the wave crest (Eqs. 1.4 and 1.5), with $H = H_{2\%}$ and $T = T_p$. Based on their analysis, de Vos et al. (2007) proposed Eq. 1.6 to calculate the $R_{u2\%}$:

$$R_{u,2\%} = \eta_{max} + 2.71\frac{u^2}{2g} \tag{1.6}$$

However, in most cases only the significant wave height H_s is available for irregular waves. For Rayleigh distributed waves, $H_{2\%}$ can be estimated using Eq. 1.7. However, one must be careful to use Eq. 1.7 because when H_s is on the limit of breaking, Eq. 1.6 will overestimate the run-up, because the shallow-water wave distribution differs from the Rayleigh distribution.

$$H_{2\%} = 1.40H_s \tag{1.7}$$

Lykke Andersen et al. (2011) applied Dean's stream function theory to calculate the wave kinematics instead of the second-order Stokes theory and reanalyzed the data of de Vos et al. (2007) as a function of $\frac{d}{D}$ and $\frac{H_s}{d}$. An increase in m was observed with increasing wave height to water depth ratio. To further investigate this relationship, additional small scale physical experiments (1:50) were performed using regular and irregular waves for $\frac{d}{D} =$ 2 - 4, $\frac{H_s}{d} =$ 0.35 - 0.46 and wave steepnesses (s_o) = 0.02 and 0.035, calculated using Eq. 1.8 and parameters at the monopile:

$$s_o = \frac{H_s}{(g/2\pi \cdot T_p^2)} = 0.02 \text{ and } 0.035, \tag{1.8}$$

- H_s: Significant wave height = H_{m0} for irregular waves $[m]$
- T_p: Peak wave period $[s]$

For each single wave, the parameter m was individually determined and a unified solution was found applicable across different scenarios using linear interpolation across the m values. An additional factor of 1.4 was included because of the underestimation of run-up heights for steep or breaking waves (Lykke Andersen & Brorsen, 2007).

$$R_{u,2\%} = \eta_{\text{max,2\%}} + 1.4 \cdot m \cdot \frac{u_{2\%}^2}{2g} \tag{1.9}$$

$$R_{u,\text{max}} = \eta_{\text{max}} + 1.4 \cdot m \cdot \frac{u_{\text{max}}^2}{2g} \tag{1.10}$$

where:

- $m = 4$ for $s_o = 0.02$
- $m = 3$ for $s_o = 0.035$.

Later Ramirez, Andersen, and de Vos collaborated in 2013 to test irregular breaking ($0.02 \lesssim s_o \lesssim 0.078$) wave conditions on a large scale (1:6) to quantify scale effects and to develop new formulae for the run-up heights (Ramirez et al., 2013). They introduced three different run-up levels, shown in Figure 1.4. This captures the variability in the run-up and provides more detailed insights.

- Level A: Level for green water run-up (thick layer). The most significant form of run-up, causing the highest loads on the monopile.
- Level B: Thin layer of water and air mixture, which is no longer attached to the surface of the pile.
- Level C: Level for maximum spray.

Eqs. 1.11 - 1.13 show the results, where m is calculated using the same method as Lykke Andersen et al. (2011). By comparing the experiments to the small scale experiments conducted by Lykke Andersen et al. (2011), they found no significant scale effects on run-up heights. However, their analysis indicated that extreme run-up events primarily occur under breaking or nearly breaking waves with steep fronts. Therefore, calculating run-up for individual breaking or nearly breaking waves remains challenging without considering the detailed wave shape.

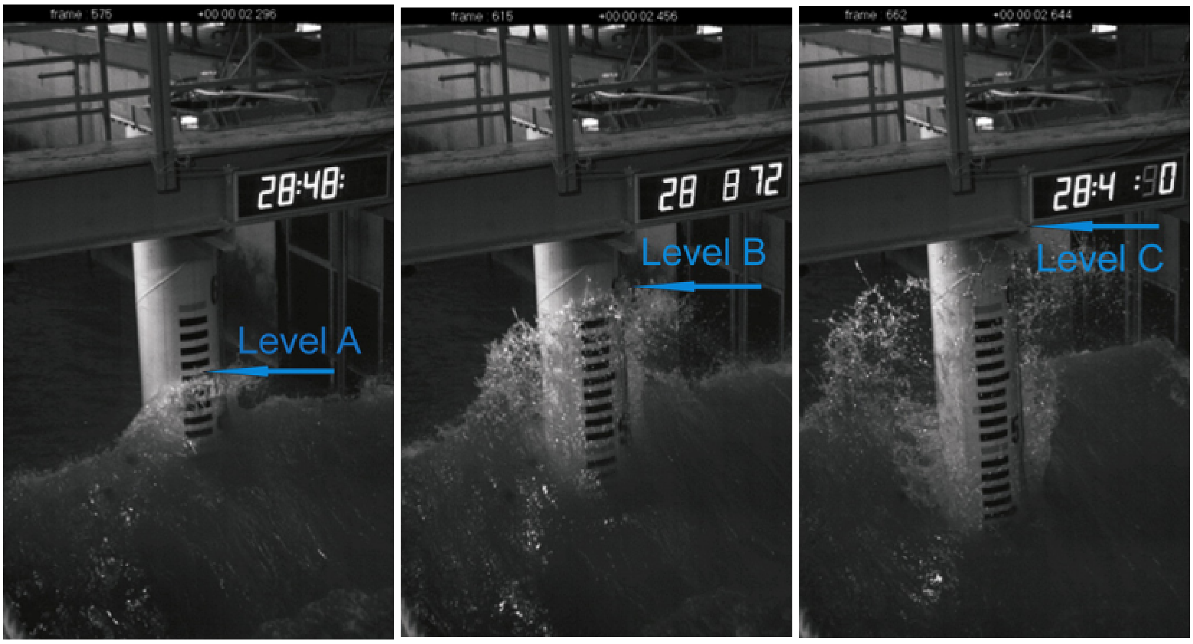


Figure 1.4: Different run-up levels from Ramirez et al. (2013).

- **Level A**

$$m = \begin{cases} -66.667 s_o + 5.33 & \text{for } s_o < 0.035 \\ 3 & \text{for } s_o > 0.035 \end{cases} \tag{1.11}$$

- **Level B**

$$m = \begin{cases} -93.333 s_o + 7.47 & \text{for } s_o < 0.035 \\ 4.2 & \text{for } s_o > 0.035 \end{cases} \tag{1.12}$$

- **Level C**

$$m = \begin{cases} -200 s_o + 16 & \text{for } s_o < 0.035 \\ 9 & \text{for } s_o > 0.035 \end{cases} \tag{1.13}$$

1.1.3. Non-dimensional parameters

Kazeminezhad and Etemad-Shahidi (2015) investigated the wave run-up from regular and irregular waves on vertical piles using the laboratory data of de Vos et al. (2007) and Ramirez et al. (2013). In previous studies, the wave run-up was based on the diffraction and velocity stagnation head theory. However, Kazeminezhad and Etemad-Shahidi (2015) estimated the run-up for non-breaking waves based on non-dimensional parameters: the wave height to water depth ratio $\frac{H}{d}$ and the wave height to wavelength ratio $\frac{H}{L}$ for the limited range of $0.02 \leq \frac{D}{L} \leq 0.07$. Based on these parameters and using a combination of the M5 model tree and non-linear regression, they found the following formulas for predicting wave run-up for regular waves:

$$\frac{R_u}{H} = 0.76 \left(\frac{H}{d}\right)^{0.15} \left(\frac{H}{L_0}\right)^{-0.055} \quad \text{for} \quad \frac{H}{d} \leq 0.41 \tag{1.14}$$

$$\frac{R_u}{H} = 0.65 \left(\frac{H}{L}\right)^{-0.055} + 3.2 \times 10^{-3} \left(\frac{H}{d} - 0.41\right)^{0.15} \left(\frac{H}{L}\right)^{-1.5} \quad \text{for} \quad \frac{H}{d} > 0.41 \tag{1.15}$$

The formulae developed by Kazeminezhad and Etemad-Shahidi (2015) are limited to deep and intermediate water conditions and do not account for shallow water scenarios, where higher run-up heights can occur. To address this gap, Bonakdar et al. (2016) carried out small-scale experiments with non-breaking regular waves. The experiments covered a wide range of hydrodynamic conditions, allowing identification of the most relevant non-dimensional parameters: $\frac{H}{d}$, $\frac{d}{L}$, and $\frac{D}{L}$. Based on these, new run-up prediction formulae were derived again using the M5 model tree and non-linear regression, for non-breaking waves in the range of $\frac{H}{d} = 0.070 - 0.517$, $\frac{d}{L} = 0.042 - 0.640$ and $\frac{D}{L} = 0.003 - 0.050$.

$$\frac{R_u}{H} = 0.863 \left(\frac{H}{d}\right)^{0.17} \left(\frac{H}{L}\right)^{-0.206} \left(\frac{D}{L}\right)^{0.106} \quad \text{for} \quad \frac{H}{d} \leq 0.41 \tag{1.16}$$

$$\frac{R_u}{H} = 0.777 \left(\frac{d}{L}\right)^{-0.206} \left(\frac{D}{L}\right)^{0.106} + 0.138 \left(\frac{H}{d} - 0.41\right)^{0.316} \left(\frac{d}{L}\right)^{-2.6} \left(\frac{D}{L}\right)^{1.16} \quad \text{for} \quad \frac{H}{d} > 0.41 \tag{1.17}$$

1.2. State of the Art for Wave-Induced Forces

Understanding wave-induced forces is crucial for the safe and cost-effective design of offshore monopiles. Accurate prediction of these forces ensures structural stability, prevents failures and helps optimize the overall design process, enhancing the safety and durability of offshore wind turbines. These forces can be evaluated using analytical, experimental or numerical methods, each with its own strengths and limitations.

Analytical methods are commonly used for initial design stages, but the choice of method depends on the dominant force regime (Fig. 1.5). Methods such as diffraction theory, Froude-Krylov theory and Morison's equation provide simplified estimations based on certain assumptions. However, their applicability is limited, particularly under highly non-linear wave conditions. Therefore, numerical methods, especially CFD, can provide detailed simulations and results of wave-structure interaction across a wide range of conditions. This chapter introduces the analytical formulae to determine the wave-induced forces.

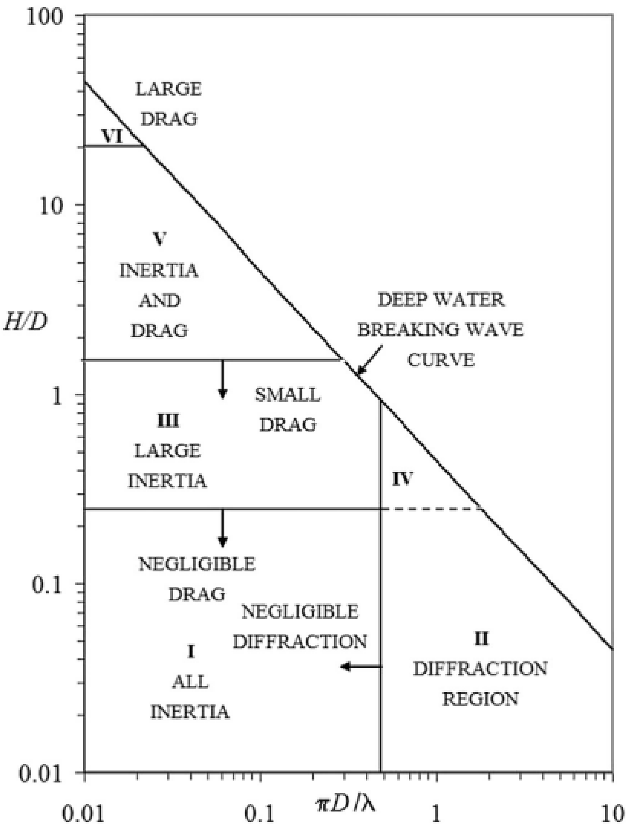


Figure 1.5: Different wave force regimes (Negro et al., 2014).

Most commonly used is the Morison's equation (Eq. 1.18), which is applicable for the inertia and drag regions (I, III and V), where diffraction is negligible. It assumes that the structure is small relative to the wavelength ($\frac{D}{L} < 0.2$). However, it does not account for non-linear wave effects, breaking waves or wave diffraction (Benitz et al., 1990). The equation consists of two terms representing the drag and inertial forces, each scaled by their drag and inertia coefficient. These coefficients must be estimated from experiments or design codes, and their accuracy can significantly influence the reliability of the force prediction.

$$F = F_D + F_I = \frac{1}{2}C_D\rho AU|U|+C_M\rho V\frac{dU}{dt} \tag{1.18}$$

where:

- F_D = Drag force component $[N]$
- F_I = Inertia force component $[N]$
- C_D = Drag coefficient [-]
- C_M = Inertia coefficient [-]
- ρ = Water density $[kg/m^3]$
- A = Cross-sectional area of the monopile $[m^2]$
- V = Displaced volume $[m^3]$
- U = Flow velocity $[m/s]$
- $\frac{dU}{dt}$ = Flow acceleration $[m/s^2]$

There are several methods to determine the appropriate C_D and C_m values (Journée & Massie, 2001). The first one is the Morison's method, which is a simple method to determine the coefficients. The method assumes that the wave motion is harmonic, i.e. it follows a sinusoidal form. In such motion, the water particle velocity and acceleration are out of phase, meaning when one reaches its maximum, the other is zero. The water particle velocity and acceleration can be described by:

$$u(t) = U\sin(\omega t) \quad \text{(velocity)} \tag{1.19}$$

$$\dot{u}(t) = \omega U\cos(\omega t) \quad \text{(acceleration)} \tag{1.20}$$

From these expressions, it follows that the velocity is at its maximum when the acceleration is zero (e.g., at the wave crest or trough), and conversely, the acceleration is at its maximum when the velocity is zero. This behavior arises because the velocity and acceleration vary as sine and cosine functions, where the sine function reaches its maximum when the cosine function is zero, and vice versa. This behavior allows the Morison equation to be simplified at specific time instants: when the acceleration is zero, the inertia term vanishes, allowing the drag coefficient C_D to be isolated; and when the velocity is zero, the drag term disappears, enabling calculation of the inertia coefficient C_M. This results in the following simplified expressions for the coefficients:

$$C_D = \frac{2F}{\rho D u|u|} \quad \text{at an instant } t_1 \text{ when } \dot{u}(t_1) = 0 \tag{1.21}$$

$$C_M = \frac{4F}{\pi\rho D^2 \cdot \omega u} \quad \text{at an instant } t_2 \text{ when } u(t_2) = 0 \tag{1.22}$$

However, this method lacks accuracy since $F(t)$ can be steep, and a small error in the velocity record can lead to a significant phase error, resulting in inaccuracies in the calculated coefficients. Another method is the Least Squares Method (LSM), which minimizes the residual force error by fitting the Morison equation to measured data. The method solves iteratively for C_D and C_M. However, while LSM is widely used to determine the coefficients, it is important to recall that this method does not predict hydrodynamic forces but relies on measured force data. This makes it unsuitable to predict wave-induced forces.

It is therefore better to determine the coefficients based on laboratory tests. Sarpkaya and Isaacson (1981) conducted experiments to investigate the drag and inertia coefficients on smooth cylinders. Clauss et al. (1992) provided recommendations for selecting the drag and inertia coefficients in the Morison equation for smooth cylinders, based on the Keulegan–Carpenter (KC) number and the Reynolds number (Re). These two dimensionless parameters are essential in characterizing the wave-structure interaction. The KC number is defined as:

$$KC = \frac{UT}{D} \tag{1.23}$$

where U is the maximum horizontal flow velocity, T is the wave period, and D is the diameter of the structure. This parameter helps determine the relative importance of drag and inertia forces in oscillatory flow conditions. The Reynolds number characterizes the flow regime and indicates whether the flow around the structure is laminar or turbulent, and is defined as:

$$Re = \frac{UD}{\nu} \tag{1.24}$$

where ν is the kinematic viscosity of the fluid. The values of the drag and inertia coefficients suggested by Clauss et al. (1992), based on ranges of KC and Re, are presented in Table 1.1.

Table 1.1: Morison Coefficients Suggested by Clauss et al. (1992)

	$Re < 10^5$		$Re > 10^5$	
KC	C_D	C_M	C_D	C_M
< 10	1.2	2.0	0.6	2.0
≥ 10	1.2	1.5	0.6	1.5

Figure 1.6 presents the experimental results from Sarpkaya and Isaacson (1981) as graphs of the drag and inertia coefficient plotted against KC number for various values of the parameter β, which is defined as:

$$\beta = \frac{Re}{KC} = \frac{D^2}{\upsilon T} \tag{1.25}$$

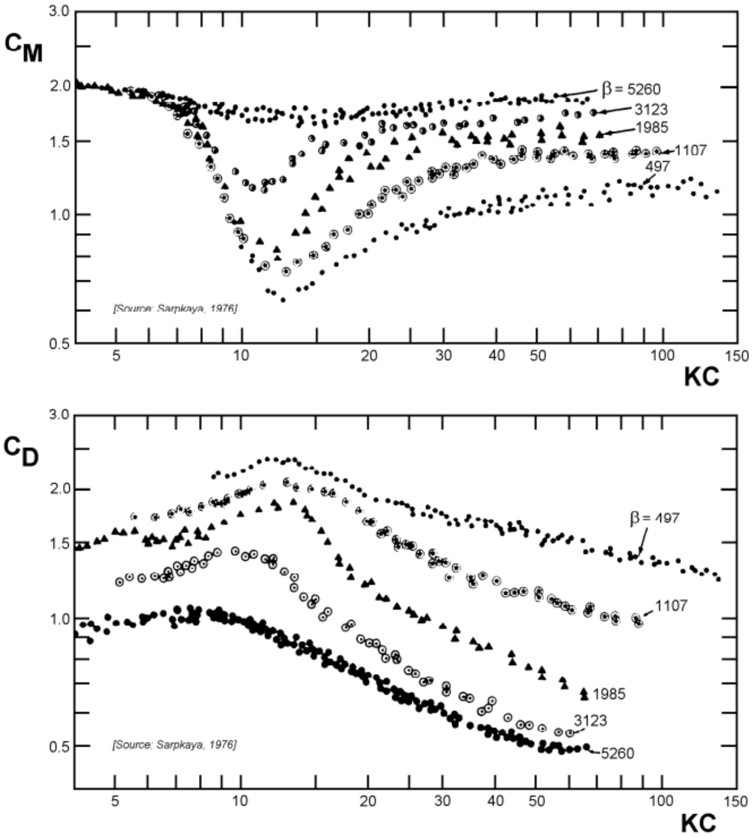


Figure 1.6: Result from Sarpkaya and Isaacson (1981) experiments.

Next is the Froude-Krylov theory (region I), which estimates the wave-induced force on a structure by integrating the pressure from the undisturbed wave field over the wetted surface (Eq. 1.26). It assumes that the structure is small enough that it does not disturb the incoming wave, allowing diffraction and viscous effects to be neglected, which is applicable for $KC < 3$, $H/D < 0.3$ and $\frac{\pi D}{L} < 0.5$ (Benitz et al., 1990; Journée & Massie, 2001).

$$F = \int_S p\, dS \tag{1.26}$$

Where:

- p = the pressure on the body in the undisturbed wave $[Pa]$
- S = the wetted surface area $[m^2]$

Lastly, when the monopile diameter is large compared to the wavelength (region II, $\frac{\pi D}{L} > 0.5$), diffraction effects must be considered. MacCamy and Fuchs (1954) analyzed the wave interaction with a vertical cylinder fixed to the seabed, assuming incompressible and irrotational flow. By applying 5 boundary conditions to the governing Laplace equation and using the Jacob-Angler expansion and Bessel/Hankel functions, an analytical solution was derived for determining wave forces on a fixed structure with a $\frac{\pi D}{L} > 0.5$ ratio. This analytical approach forms the basis of the diffraction theory (MacCamy & Fuchs, 1954; Walker, 2024).

1.3. Knowledge Gap and Research Scope

Extensive semi-empirical research has been dedicated to predicting wave run-up on monopiles. However, these methods fail to account for the non-linear contributions that occur in more extreme wave conditions, particularly for mild to steep waves ($\gtrsim 0.03$). Furthermore, although semi-empirical models, such as those proposed by de Vos et al. (2007), Kazeminezhad and Etemad-Shahidi (2015) and Bonakdar et al. (2016), have been developed for certain wave conditions. There is still a lack of a universally applicable method for predicting wave run-up across all wave conditions, especially in shallower waters, where there are nearly breaking and breaking waves. These limitations highlight the need for further research to develop an accurate, validated and generalized model that can predict the wave run-up under a wider range of conditions.

The scope of this research is to explore the use of OpenFOAM, an open-source Computational Fluid Dynamics (CFD) toolbox, to bridge the existing gaps in wave run-up prediction. The study will first focus on recreating the experiment from de Vos et al. (2007), using the same hydrodynamic conditions to validate the CFD approach. After successful validation, the model will be extended to investigate steeper waves with a wave steepness greater than 0.03, as well as scaled-up scenarios, to assess their influence on wave run-up predictions. The scope of this study is limited to regular waves. Ultimately, the goal is to develop an accurate and reliable model for predicting wave run-up and the associated forces on monopiles, while improving the understanding of how varying wave characteristics impact both run-up and loading.

1.4. Research Questions

The main focus of this research is to develop a reliable numerical model that can simulate wave run-up on monopiles, to gain more insights into this phenomena. Therefore, the main research question of this study is:

How can wave run-up and hydrodynamic forces on a monopile be accurately predicted under varying hydrodynamic conditions using computational fluid dynamics?

To answer the main research question, the following sub-questions will be answered and investigated to create the optimal model:

- What is the current state of the art in calculating wave run-up and the associated hydrodynamic forces on monopiles?
- Which types of waves result in the highest wave run-up and forces on the monopile, and how should they be modeled?
- How can the wave generation, reflection, and absorption be modeled within computational fluid mechanics to ensure a continuous and realistic wave environment?
- How do scale effects impact the prediction of wave run-up in model-scale vs full-scale simulations?
- What is the influence of different monopile diameters on the wave run-up under the same hydrodynamic conditions?
- How do wave characteristics influence the magnitude of wave run-up and forces on monopiles?
- How does turbulence affect the wave run-up and forces on monopiles and what turbulence model should be used to simulate these effects accurately?

1.5. Research Methodology

As stated before, modeling wave run-up is crucial for the design and safety of offshore wind turbines. But, it is challenging due to the complex non-linear interaction between waves and monopiles. Previous research often focused on linear wave theory and simplified computational models, neglecting non-linear effects. Due to the increase of computational power and advanced computational fluid dynamics tools, like OpenFOAM, it is now possible to simulate this complex interaction (Ramirez et al., 2011). However, many models are still simplified or lack validation and accuracy.

Therefore, the primary aim is to develop and validate an accurate OpenFOAM model for simulating wave run-up on monopiles under varying hydrodynamic conditions as mentioned in section 1.3. This will be done by initially recreating the experiment by de Vos et al. (2007) in OpenFOAM, which will then be validated against the data from the experiment using Python for data processing and ParaView for visualization. After validation, the model will be scaled up to real-life dimensions to investigate the scale effects. Followed by a series of simulations with varying wave characteristics to evaluate their influence. The outcomes of these simulations will be compared against existing analytical methods to assess their accuracy and relevance. The validation experiment of de Vos et al. (2007), the complete model setup and validation process, are described in Chapter 3. A brief overview is provided below:

1. Recreate the validation model of de Vos et al. (2007)
2. Perform a sensitivity analysis to identify the optimal model configuration
3. Scale the model to real-life dimensions to assess potential scale effects
4. Simulate different scenarios with varying wave characteristics
5. Evaluate and interpret the simulation results
6. Compare the results with existing analytical methods

The next chapter provides an overview of OpenFOAM and the waves2Foam toolbox, which are both used to simulate the wave run-up on monopiles in this research. OpenFOAM is a C++ toolbox for the solution of continuum mechanics problems, including computational fluid dynamics (CFD). It will describe its general layout and working, including its directory structure and utilities. Following this, the waves2Foam toolbox will be introduced, which extends OpenFOAM's wave generation, propagation, and absorption capabilities. Finally, the waveIsoFoam solver, which uses the isoAdvector method for improved accuracy in tracking the free surface, will be discussed.

1.5.1. OpenFOAM & waves2Foam

In this research, CFD is used to analyze wave run-up on monopiles. For this purpose, the open-source CFD toolbox, OpenFOAM, is used. OpenFOAM is a free, open-source CFD software package developed by OpenCFD Ltd. and distributed by the OpenFOAM Foundation. In the field of fluid mechanics, it solves the Reynolds averaged Navier-Stokes equations used to model turbulent flows and captures the free surface via the extensively used VOF method (Peng, 2014). It is a powerful platform to simulate complex fluid-structure interactions, making it well-suited for modeling wave run-up on monopiles. It is a fully non-linear model that can resolve higher-order effects neglected by the linear methods (Edesess, 2018).

OpenFOAM is fundamentally a C++ library used to build executables called applications. A C++ library is a collection of pre-written code that provides reusable functionalities for developers, allowing them to build upon or modify the code without starting from scratch (Josuttis, 2012). These applications are divided into two types: solvers, designed to solve a specific problem in computational continuum mechanics, and utilities, that perform pre- and post-processing tasks, mesh generation, data conversion, field data manipulation, and visualization of results. They complement solvers by preparing input data and processing simulation results. Lastly, OpenFOAM also allows for the development of custom solvers and utilities. These are modifications or additions to the core OpenFOAM framework, designed to address specialized tasks that go beyond the standard solvers and utilities. Many researchers and engineers develop their own solvers for applications such as offshore wind energy, wave dynamics, and fluid-structure interactions, extending OpenFOAM's capabilities. Additionally, various third-party modules, such as waves2Foam for wave generation and absorption, provide enhanced functionalities tailored to specific simulation needs. While these modifications integrate into OpenFOAM's workflow, they are typically shared within the user community rather than being classified as official ”extensions” of OpenFOAM (OpenFOAM, 2024).

In OpenFOAM, the main directory structure for each simulation follows a standard format to ensure organized storage of input parameters, boundary conditions, solver settings and numerical schemes. The main directory typically consists of three key subdirectories: *0*, *system*, and *constant*.

The **0** folder contains the initial and boundary conditions for the flow variables, including velocity, pressure, and phase fraction in multiphase simulations. Each field file within this directory specifies the boundary conditions applied to different patches and walls, with predefined initial values where necessary. For instance, Dirichlet boundary conditions require an explicit specification of initial values.

Within the **system** directory, numerical schemes for solving the governing equations are detailed in the *fvSchemes* file. Solver settings, including time-stepping parameters, PISO, SIMPLE or PIMPLE algorithm settings and tolerance levels, are defined in the *fvSolution* file. Additionally, the *blockMeshDict* file, defines the mesh generation parameters for the simulation when using the blockMesh utility (Appendix A.1). This file contains information on the domain dimensions, number of cells, and mesh grading. Another important file within this directory is *controlDict*, which controls global simulation parameters such as time step settings, write intervals, and solver execution settings.

The **constant** folder stores the mesh definition from the blockMesh utility, which is structured inside the *polyMesh* subdirectory. Additionally, it contains the physical and transport properties of the fluids. The *transportProperties* file, located in this directory, defines key parameters such as viscosity, which are crucial for modeling fluid behavior in multiphase simulations. Other environmental conditions, acting as external source terms, are also specified in this directory and located in the *environmentalProperties*. Furthermore, turbulence modeling is configured in the *turbulenceProperties* file, where users can specify whether the flow should be treated as laminar or turbulent. The specific details of the model can be found in Chapter 3 and all the files can be found in the project folder (Drost, 2025).

Building on this OpenFOAM framework, waves2Foam extends its capabilities by introducing specialized tools for wave generation, propagation, and pre- and post-processing. It enables the simulation of complex free-surface flows and wave-structure interaction. The waves2Foam toolbox was originally developed at the Technical University of Denmark by Niels Gjøl Jacobsen under the supervision of Prof. Jørgen Fredsøe. The waves2Foam toolbox is a plug-in to the open-source general-purpose CFD-package OpenFOAM and was released as a toolbox for the generation, propagation and absorption of free surface waves as well as modeling the interaction between free-surface waves and bodies (Jacobsen, 2017). By building on OpenFOAM's VoF method, it enables accurate tracking of the air-water interface, making it highly effective for free-surface flow. OpenFOAM takes care of the underlying CFD framework, including general meshing, discretization and solving governing equations. Waves2Foam extends this functionality by introducing specialized solvers and additional boundary equations options for wave-related phenomena (Scholte, 2020).

The governing equations solved in waves2Foam are based on the principles of incompressible, immiscible two-phase flow. These include the continuity equation to ensure mass conservation, the Navier-Stokes equations to account for momentum conservation and the VOF equation to track the evolution of the air-water interface. Combined, these equations describe the flow dynamics. All of them are discussed in detail in Chapter 2.1. Additionally, waves2Foam provides advanced pre- and post-processing capabilities. It contains the utility *waveGaugesNProbes* which allows for the definition of probes and wave gauges used to extract time-series data of the free surface elevation at specific locations. These locations are referred to as wave gauge points or probes. It acts similar to physical sensors in experimental setups. It can also extract a variety of different physical quantities (e.g. pressure or velocity) beyond just free surface elevation.

In addition, waves2Foam supports various wave theories. The available wave theories in the waves2Foam toolbox are algebraic, bichromatic, first-order irregular, potential current, solitary wave theory, combined waves and external wave theories (Jacobsen, 2017). The flexibility of waves2Foam lies in its ability to implement these wave theories through parameterized input files. Users can define key wave parameters, such as wave height, period, direction and/or spectrum type, easily without needing to modify the underlying code. This ensures that simulations can be quickly adapted to different scenarios. It simplifies the implementation through its wave generation and absorption zones, also called relaxation zones. These zones ensure that the desired wave characteristics are generated accurately at the boundaries and minimize unwanted reflections, ensuring realistic wave behavior. Both the wave theory and the key wave parameters, as well as the relaxation zones, are specified in the *waveProperties.input* file (Appendix A), making it very user-friendly.

Solver

To accurately model multiphase (e.g. water and air) flows in a wave flume, the solver interFoam is widely used in CFD. Its flexibility in handling incompressible, immiscible two-phase flows makes it a reliable choice for many applications, including general free surface flow problems (Deshpande et al., 2012; Larsen et al., 2019). However, when the focus is on simulating waves in a wave flume, waveIsoFoam offers distinct advantages. This solver is specifically designed for wave-focused applications. It addresses the limitations of interFoam and provides a more accurate and efficient solution for wave propagation, breaking, and interaction with structures (Chen et al., 2021; Jacobsen, 2017).

Roenby et al. (2016) tested the effect of replacing the existing interface advection method in interFoam, based on MULES, with the isoAdvector method and found that it results in a sharper and smoother surface and that it is less sensitive to cell aspect ratio, illustrated in figure 1.7 (Roenby et al., 2016). The primary difference between the two solvers lies in how they track the free surface. InterFoam uses the VOF method to capture the phase interface by solving the volume fraction field (α) through the advection equation. To maintain boundedness ($0 \leq \alpha \leq 1$) and stability of the VOF, it often employs the Multidimensional Universal Limiter with Explicit Solution (MULES). However, since the standard VOF approach relies on algebraic schemes to solve the advection term ($\nabla \cdot (\alpha \mathbf{u})$) (Eq. 1.27), numerical diffusion can occur, particularly in regions with steep gradients or irregular geometries. To counteract this, an additional compression term ($\nabla \cdot (\alpha(1-\alpha)\mathbf{u}_r)$) is explicitly added to sharpen the interface, which results in the following advection equation:

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\mathbf{u}\alpha) + \nabla \cdot (\mathbf{u}_r \alpha(1-\alpha)) = 0 \tag{1.27}$$

While this ensures sharper interfaces, it can still lead to inaccuracies due to the excessive reliance on numerical corrections (Damian, 2013).

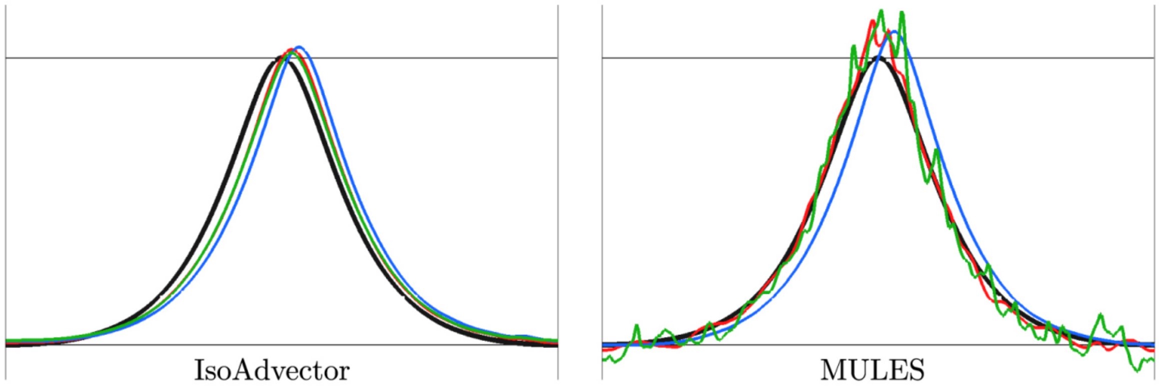


Figure 1.7: Results from Roenby et al. (2016): Surface elevation after 10 wave periods. Black: Exact. Red: square cells, $H/dx = H/dy = 20$. Blue: Flat cells, $H/dx = 10$, $H/dy = 20$. Green: Tall cells, $H/dx = 20$, $H/dy = 10$. Left panel: IsoAdvector, Right panel: interFoam/MULES

.

The isoAdvector method, which is implemented in waveIsoFoam, eliminates the need for the additional compression term by fundamentally changing how the advection equation is handled. Rather than solving the advection term ($\nabla \cdot (\alpha \mathbf{u})$) using algebraic discretization schemes such as vanLeer or interfaceCompression, isoAdvector replaces this process with a geometric reconstruction of the interface. Specifically, isoAdvector reconstructs the interface as a geometric iso-surface (a surface of constant $\alpha = 0.5$) within each computational cell. The position and orientation of the interface are determined directly from the volume fraction field (α), and the flux across the interface is calculated based on this reconstructed geometry rather than through traditional numerical solutions to the advection equation (Eq. 1.27) (Roenby et al., 2017).

This geometric reconstruction eliminates numerical diffusion by ensuring that the interface sharpness is maintained without relying on iterative corrections or artificial compression terms. Unlike VOF with MULES, which can smear the interface over time due to errors in solving the advection equation, isoAdvector inherently preserves a sharp boundary. By directly aligning the flux calculation with the interface geometry, isoAdvector avoids the need to approximate the interface location or apply correction factors, making it more accurate and robust for free-surface flows. The absence of the additional compression term in isoAdvector also simplifies the governing equation, reducing computational cost while simultaneously improving accuracy. Furthermore, boundedness ($0 \leq \alpha \leq 1$), which is explicitly enforced in MULES through iterative corrections, is inherently satisfied in isoAdvector due to the constraints imposed by the geometric reconstruction. This ensures that the volume fraction field remains physically realistic throughout the simulation (Roenby et al., 2017).

By reconstructing the interface instead of relying on numerical solutions to the advection equation, isoAdvector provides significant advantages in terms of accuracy and efficiency. It removes numerical diffusion and ensures sharp, stable interfaces, even in challenging scenarios such as wave dynamics or irregular grid geometries. This makes isoAdvector particularly well-suited for applications involving free-surface flows, where a precise representation of the interface is critical (Roenby et al., 2016).

Another critical advantage of waveIsoFoam is its built-in capability for wave generation and absorption. Unlike the standard interFoam solver, which requires additional utilities or frameworks such as waves2foam to simulate wave generation and damping zones, waveIsoFoam is a specialized solver developed within the waves2Foam framework itself. This results in the direct integration of wave generation, propagation and absorption functions, into its core, eliminating the need for coupling external solvers to handle these processes. From a computational perspective, while the isoAdvector method in waveIsoFoam may marginally increase computational cost compared to the VOF method in interFoam, the gains in accuracy and interface sharpness make it the superior choice for wave-specific applications.

2
Theoretical Framework

2.1. Governing Equations

As mentioned in the section 1.5, OpenFOAM and its extension, waves2Foam, provide advanced numerical techniques to model wave generation, propagation and absorption in two-phase flow systems. Its theoretical foundation combines the principles of the Finite Volume Method (FVM), the Volume of Fluid (VOF) approach and the PIMPLE algorithm to solve the governing equations for two-phase incompressible flows. OpenFOAM uses the Navier-Stokes equations to describe the motion of the two-phase incompressible fluid (air and water). These equations consist of the momentum and the continuity equation (Jacobsen, 2017). They are applied in their conservative form, where the momentum equation is expressed in terms of momentum fluxes. This ensures mass and momentum conservation, commonly used in the FVM for two-phase incompressible flow.

1. Momentum Equation (Navier-Stokes for Two-Phase Flow):

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot \rho \mathbf{u}\mathbf{u}^T = -\nabla p^* + \mathbf{g} \cdot (\mathbf{x} - \mathbf{x_r})\nabla \rho + \nabla \cdot \mu_{tot} \nabla \mathbf{u}. \tag{2.1}$$

Where:

1. **Unsteady and Convective Terms:**
   - $\frac{\partial \rho \mathbf{u}}{\partial t}$: Represents the time-dependent changes in momentum, also known as the transient term.
   - $\nabla \cdot \rho \mathbf{u}\mathbf{u}^T$: Accounts for the transport of momentum due to convection in the flow.
     - ρ: Effective density (Eq. 2.15)
     - $\mathbf{u}$: Velocity vector
   - $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)$
2. **Excess Pressure Gradient:**
   - $-\nabla p^*$: The pressure gradient term, where $p^* = p - \rho g z$, represents the pressure in excess of the hydrostatic pressure. Separating hydrostatic effects improves numerical stability.
3. **Gravitational Force:**
   - $\mathbf{g} \cdot (\mathbf{x} - \mathbf{x_r})\nabla \rho$: Accounts for the buoyancy effects caused by density differences, where:
     - $\mathbf{g}$: Gravitational acceleration vector.
     - $\mathbf{x}$: Spatial coordinates.
     - $\mathbf{x_r}$: Reference location for the hydrostatic pressure.
4. **Viscous Stresses:**
   - $\nabla \cdot \mu_{tot} \nabla \mathbf{u}$: Represents the viscous diffusion of momentum.
     - μ_{tot}: Effective viscosity (Eq. 2.16)

2. Continuity Equation:

$$\nabla \cdot \mathbf{u} = 0, \tag{2.2}$$

- Which ensures mass conservation for incompressible flow.

2.1.1. Finite Volume Method

The Finite Volume Method (FVM) is a numerical approach used to convert partial differential equations (PDE), which describe conservation laws over continuous domains, into discrete algebraic equations defined over finite volumes. Similar to the finite difference and finite element methods, the process begins with the discretization of the geometric domain. In FVM, the domain is divided into non-overlapping finite volumes that collectively cover the entire domain. The governing partial differential equations are then transformed into algebraic equations by integrating them over each finite volume. The resulting system of algebraic equations is then solved to determine the values of the dependent variables within each finite volume (Jasak, 1996; Moukalled et al., 2012). Below is a detailed explanation of how the FVM is applied to the momentum equation, followed by the discretized continuity equation.

The momentum equation (Navier-Stokes for two-phase flow) is expressed in Eq. 2.1. The FVM transforms the PDE into a set of algebraic equations by integrating it over a control volume, V, and applying Gauss's divergence theorem to convert surface integrals into volume integrals.

1. **Integration of the Momentum Equation over a Control Volume:**

$$\int_V \frac{\partial(\rho \mathbf{u})}{\partial t}\, dV + \int_V \nabla\cdot(\rho\mathbf{u}\mathbf{u}^T)\, dV = -\int_V \nabla p^*\, dV + \int_V \mathbf{g}\cdot(\mathbf{x}-\mathbf{x}_r)\nabla\rho\, dV + \int_V \nabla\cdot(\mu_{tot}\nabla\mathbf{u})\, dV \tag{2.3}$$

2. **Application of Gauss's Divergence Theorem:**
The Gauss's divergence theorem states that the net flux of a vector field through the surface of a volume equals the divergence of the field integrated over the volume (Moukalled et al., 2012). In other words, this means that instead of evaluating the divergence of a vector field over an entire control volume, we compute the net flux of the field across all the faces that enclose the volume. In FVM it allows for the replacement of complex volume integrals with surface integrals, which can be evaluated as fluxes across the faces of finite volumes:

$$\int_V \nabla\cdot\mathbf{F}\, dV = \int_S \mathbf{F}\cdot\mathbf{n}\, dS = \sum_f \mathbf{F}\cdot\mathbf{n}_f \Delta S_f \tag{2.4}$$

where:

- V Is a volume (or domain) in space
- S Is the closed surface that bounds the volume V
- $\mathbf{F}$ Is a vector field (e.g., velocity, momentum flux, or diffusive flux)
- $\mathbf{n}$ Is the outward-facing unit normal vector to the surface S
- f Refers to each face surrounding the control volume

The left-hand side $\left(\int_V \nabla\cdot\mathbf{F}\, dV\right)$ computes the volume integral of the divergence of the vector field $\mathbf{F}$ within the volume V. The right-hand sides $\left(\int_S \mathbf{F}\cdot\mathbf{n}\, dS\right)$ and $(\sum_f \mathbf{F}\cdot\mathbf{n}_f\Delta S_f)$ compute the net flux of $\mathbf{F}$ through the surface S.

The equation is now expressed as an integral form over the volume V and its enclosing surface S:

$$\int_V \frac{\partial(\rho\mathbf{u})}{\partial t}\, dV + \int_S (\rho\mathbf{u}\mathbf{u}^T)\cdot\mathbf{n}\, dS = \int_S (-p\mathbf{n})\, dS + \int_S (\mu_{tot}\nabla\mathbf{u})\cdot\mathbf{n}\, dS + \int_V \mathbf{g}\rho\, dV \tag{2.5}$$

3. **Discretization:** The governing equation is now transformed from its integral form into a surface integral over the control volume's boundaries. The discretization process converts the integral form into a system of algebraic equations for numerical solution. Each term in the momentum equation is discretized as follows:

The transient term $\frac{\partial(\rho\mathbf{u})}{\partial t}$ is discretized using a time-stepping scheme. In this case using the implicit Euler method (first-order accurate):

$$\frac{\partial(\rho\mathbf{u})}{\partial t} \approx \frac{\rho\mathbf{u}_P^{n+1} - \rho\mathbf{u}_P^n}{\Delta t} \tag{2.6}$$

where:

- $\mathbf{u}_P^{n+1}$: Velocity at the next time step
- $\mathbf{u}_P^n$: Velocity at the current time step
- Δt: Time step size

Convective Term:

$$\int_S (\rho \mathbf{u}\mathbf{u}^T) \cdot \mathbf{n}\, dS \approx \sum_f (\rho_f \mathbf{u}_f \mathbf{u}_f) \cdot \mathbf{n}_f A_f \tag{2.7}$$

where:

- A_f: Area of the face

Interpolation schemes (e.g., upwind, central differencing) are used to calculate face-centered values ρ_f and $\mathbf{u}_f$.

Pressure Gradient Term:

$$\int_S (-p\mathbf{n})\, dS \approx \sum_f (-p_f \mathbf{n}_f) A_f \tag{2.8}$$

where:

- p_f: Pressure at the face, interpolated from cell values

Diffusive Term:

$$\int_S (\mu_{\text{tot}} \nabla \mathbf{u}) \cdot \mathbf{n}\, dS \approx \sum_f (\mu_f \nabla \mathbf{u}_f) \cdot \mathbf{n}_f A_f \tag{2.9}$$

where:

- $\nabla \mathbf{u}_f$: Velocity gradient at the face
- μ_{tot}: Total viscosity (Eq. 2.29)

Body Force Term:

$$\int_V \mathbf{g}\rho\, dV \approx \mathbf{g}\rho_P V_P \tag{2.10}$$

where:

- ρ_P: The density at the cell center

Combining these terms, the discretized momentum equation for each cell becomes:

$$a_p u_P = \sum_f a_f u_f - \sum_f p_f n_f A_f + g\rho_P V_P, \tag{2.11}$$

Where a_P is the coefficient for the central cell, a_f are the coefficients for neighboring cells and $\mathbf{u}_P, \mathbf{u}_f$ are the unknown velocities at the cell center and faces, respectively.

To solve the momentum equation, each term in the momentum equation is discretized individually (Eq. 2.6 - Eq. 2.10) and the process uses the appropriate numerical schemes specified in the fvSchemes file (section 3.2.7). These schemes ensure that the fluxes, gradients and other spatial and temporal derivatives are accurately represented, maintaining numerical stability and physical consistency. Discretizing each term in the momentum equation yields coefficients that represent how the current cell interacts with itself (a_P) and with neighboring cells (a_f). These coefficients form the structure of the matrix equation that is solved to compute the velocity field. Assembling the momentum equations for all cells results in a matrix representation:

$$A\mathbf{u} = \mathbf{R} \tag{2.12}$$

1. **Coefficients of the Matrix A:**
   - The matrix A, size $3n_{cells} \times 3n_{cells}$ contains the coefficients a_P (diagonal) and a_f (off-diagonal), which depend on the velocity $\mathbf{u}$, size $(3n_{\text{cells}} \times 1)$.

- These terms represent the coupling between the velocity of the central cell $\mathbf{u}_P$ and its neighbors $\mathbf{u}_f$:

$$\mathbf{A}\mathbf{u} = [a_P]\mathbf{u}_P + [a_f]\mathbf{u}_f \tag{2.13}$$

2. **Terms Contributing to $\mathbf{R}$ (the Source Vector):** The source term $\mathbf{R}$, size $(3n_{\text{cells}} \times 1)$, contains contributions that are independent of velocity $\mathbf{u}$:
   - **Pressure Gradient ($-\sum_f p_f n_f A_f$):**
     - This term depends on the pressure p_f, which is solved separately (via PIMPLE pressure correction steps).
     - It does not directly contribute to the velocity coefficient matrix $\mathbf{A}$ during the momentum equation solution.
   - **Body Force ($g\rho_P V_P$):**
     - This gravitational force term directly adds to the source vector $\mathbf{R}$ and is constant for a given time step.

The continuity equation is discretized as:

$$\sum_f \mathbf{u}_f \cdot \mathbf{n}_f A_f = 0 \tag{2.14}$$

Where:

- $\mathbf{u}_f \cdot \mathbf{n}_f$ is the velocity component normal to face f, calculated from the velocity field.
- A_f is the area of face f.

Now that the momentum and continuity equations are discretized, the velocity (**U**) and pressure (p) are solved in a loosely coupled but iterative manner using the PIMPLE algorithm, explained in section 2.1.3. The momentum equation is solved using the smoothSolver and the pressure equation with PCG, explained in section 3.2.7. This framework ensures a consistent solution for multiphase flows, maintaining mass conservation, accurate interface tracking and computational efficiency.

2.1.2. Volume of Fluid method

The VOF method is a numerical approach used to track and model the interface between two immiscible fluids. It introduces a scalar field, α, known as the volume fraction, which indicates the proportion of each fluid present within a computational cell. The VOF method is particularly effective for simulating free surface flows, as it allows for the accurate representation of sharp interfaces without explicitly tracking surface geometry.

The scalar field α, which in this case represents the local volume fraction of water in a cell, is interpreted as follows:

- $\alpha = 1$: The cell is fully occupied by water
- $\alpha = 0$: The cell is fully occupied by air
- $0 < \alpha < 1$: The cell contains the interface between water and air

The evolution of α is governed by the advection equation:

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\mathbf{u}\alpha) = 0 \tag{2.14}$$

where:

- $\mathbf{u}$ is the velocity field

In this study, the advection of the volume fraction α is handled using the isoAdvector method, a geometric variant of the traditional VOF formulation. While standard implementations use algebraic discretization schemes (e.g., MULES) to solve the advection equation, the isoAdvector method reconstructs the interface as a geometric isosurface (at $\alpha = 0.5$) within each surface cell. This interface is then tracked over the time step Δt using the local velocity field, which determines how the surface moves through the mesh (Roenby et al., 2016). This motion allows for the direct calculation of the fluid volume fluxes across each cell face, avoiding the need for numerical approximation of derivatives. These fluxes are then used to update the volume fraction in each cell using a first-order explicit Euler time-stepping scheme (see Section 3.2.7):

$$\alpha^{n+1} = \alpha^n + \frac{V_{\text{in}} - V_{\text{out}}}{V_{\text{cell}}}$$

where:

- V_{in}: Volume of water entering the cell [m^3]
- V_{out}: Volume of water leaving the cell [m^3]

This geometric treatment significantly reduces numerical diffusion and maintains a sharp, physically realistic free surface, without requiring artificial compression terms or iterative corrections (Roenby et al., 2016, 2017).

Once the volume fraction field α is known, it can be used to compute the effective fluid properties in each cell. The density ρ and dynamic viscosity μ, required for solving the momentum equation, are interpolated based on α to distinguish between water and air:

1. **Density (ρ)**: The effective density is a volume-weighted average of the densities of water and air:

$$\rho = \alpha\rho_w + (1-\alpha)\rho_a \tag{2.15}$$

   where:

   - ρ_w: Density of water
   - ρ_a: Density of air

2. **Viscosity (μ)**: The effective dynamic viscosity is similarly interpolated:

$$\mu = \alpha\mu_w + (1-\alpha)\mu_a \tag{2.16}$$

   where:

   - μ_w: Viscosity of water
   - μ_a: Viscosity of air

2.1.3. Pressure and Velocity Coupling

The primary challenge in solving the Navier-Stokes equation lies in ensuring that the velocity field satisfies the continuity equation while solving the momentum equation for velocity. The PIMPLE algorithm combines elements of the SIMPLE and PISO algorithms, balancing stability and computational efficiency. While SIMPLE is a steady-state algorithm with outer-loop corrections, PISO is a transient algorithm capable of correcting pressure and velocity within a single time step. By merging these approaches, PIMPLE introduces outer-loop corrections for enhanced stability and convergence in transient simulations with large Courant numbers (section 3.2.7). The PIMPLE algorithm solves the Navier-Stokes equations iteratively in the following sequence (wiki, 2025):

1. **Predictor Step:**
   - Solve the momentum equation with the pressure field from the previous iteration or time step.
   - Obtain an intermediate velocity field, $\mathbf{U}^*$.
2. **Pressure-Correction Loop:**
   - Solve the pressure-Poisson equation to update the pressure field:

$$\nabla \cdot \left(\frac{1}{a_P}\nabla p\right) = \nabla \cdot \mathbf{U}^*.$$

   - Correct the velocity field using the updated pressure gradient:

$$\mathbf{U} = \mathbf{U}^* - \frac{\Delta t}{\rho}\nabla p.$$

3. **Non-Orthogonal Correction (Optional):**
   - For meshes with non-orthogonal faces (Appendix A.1.2), errors in flux evaluation caused by face angles are corrected iteratively.

- The pressure-correction equation is modified to account for these errors, refining the pressure field for more accurate flux computations:

$$\nabla \cdot \left(\frac{1}{a_P} \nabla p \right) + \text{Non-Orthogonal Terms} = \nabla \cdot \mathbf{U}.$$

- The number of corrections is controlled by the parameter $n_{\text{NonOrthogonalCorrectors}}$.

4. **Outer Correction Loop (Optional):**
    - The outer corrector loop iterates the entire predictor and pressure-correction sequence. This step handles non-linearities in the governing equations and ensures convergence for highly transient or complex flow conditions.
    - By solving the momentum and pressure-correction equations multiple times within a time step, the algorithm refines the solution, particularly for simulations with large time steps (high Courant numbers).
    - The number of outer corrections is controlled by the parameter $n_{\text{OuterCorrectors}}$.

Mathematically, this approach can be summarized as:

$$\mathbf{U}^* = solve\left(\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U}\mathbf{U}) = -\nabla p + \nu \nabla^2 \mathbf{U} + \mathbf{f} \right) \tag{2.17}$$

$$\nabla \cdot \left(\frac{1}{a_P} \nabla p \right) = \nabla \cdot \mathbf{U}^* \tag{2.18}$$

$$\mathbf{U} = \mathbf{U}^* - \frac{\Delta t}{\rho} \nabla p \tag{2.19}$$

2.2. Mathematical Formulation of Wave Theories

To accurately predict the wave run-up on offshore monopiles, a detailed understanding of wave dynamics and their structure interaction is required. Wave theories are fundamental to modeling wave propagation, interactions and the resulting forces and run-up on structures. They serve as the basis for numerical methods used to predict wave run-up.

The selection of an appropriate wave theory is influenced by wave steepness H/L and relative depth d/L, as these parameters reflect the degree of non-linearity and dispersion in wave behavior. In deep water, these ratios are typically sufficient to differentiate between linear and non-linear theories. However, their reliability decreases in intermediate to shallow waters $d/L < 0.5$, where wave dynamics become more complex (Zhao et al., 2024).

In such cases, a more reliable approach involves using dimensionless dynamic parameters such as H/gT^2 and d/gT^2, which incorporate the influence of wave period and gravitational acceleration. These parameters enable a clearer distinction between wave conditions across various depth regimes. The diagram in Figure 2.1 illustrates how different combinations of H/gT^2 and d/gT^2 serve as a basis for selecting the appropriate wave theory, ranging from linear theory to higher-order Stokes and stream function theories (Negro et al., 2014; Zhao et al., 2024).

Wave generation in waves2Foam is implemented inside a relaxation zone, as mentioned in section 3.2.4 to ensure smooth and physically consistent wave formation without introducing numerical instabilities and avoid internal reflection from the monopile in the computational domain to interfere with the wave maker boundaries (Jacobsen et al., 2012). The selection of an appropriate wave theory is determined by water depth and wave steepness, which influence whether linear or non-linear effects prevail (Fig. 2.1). It should be noted however, that within waves2Foam, Stokes second- and third-order are not available. Below, the key wave theories relevant to this research are discussed, emphasizing their governing equations, underlying assumptions, limitations, and suitability for predicting wave run-up.

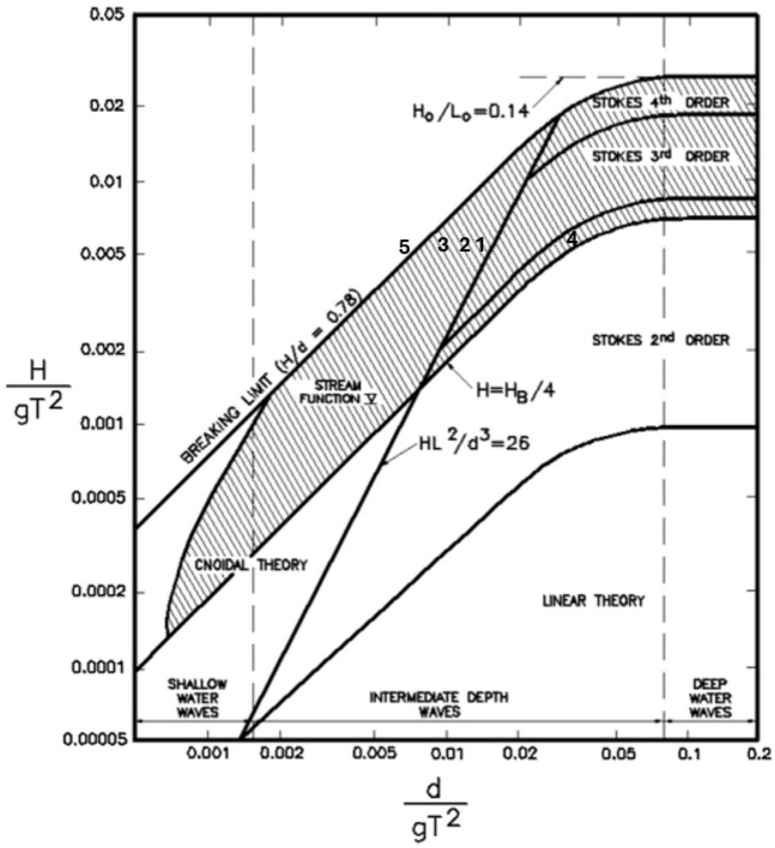


Figure 2.1: Applicability Ranges of Wave Theories in Terms of $\frac{H}{gT^2}$ and $\frac{d}{gT^2}$ (Negro et al., 2014).

2.2.1. Linear waves

Linear wave generation at the inlet can be achieved using potential flow theory (Dean & Dalrymple, 1991), a theory that simplifies the wave motion by assuming the flow is irrotational, inviscid, and incompressible. This allows the fluid motion to be described by a scalar velocity potential ϕ, which is defined such that the velocity field, **u**, can be expressed as:

$$\mathbf{u} = \nabla\phi \tag{2.20}$$

Where:

- $\mathbf{u} = (u, w)$ is the velocity vector.
- $u = \frac{\partial\phi}{\partial x}$ and $w = \frac{\partial\phi}{\partial z}$ are the horizontal and vertical velocity components, respectively.

This formulation satisfies the irrotational condition ($\nabla \times \mathbf{u} = 0$) and provides a solution for the flow field beneath the wave surface. The free surface elevation, η, for a linear wave is represented as:

$$\eta = \frac{H}{2}\cos(kx - \omega t + \phi) \tag{2.21}$$

This equation describes a sinusoidal wave propagating in the x-direction with time-dependent oscillations, where H is the wave height, ω the angular frequency, t the time, and k the wavenumber, which is obtained by solving the dispersion relation for water waves (Appendix D.1).

The velocity potential, $\phi(x, z, t)$, corresponding to the linear wave theory, is used to compute the horizontal (u) and vertical (w) velocity components. These are derived as:

$$u = \frac{\partial\phi}{\partial x}, \quad w = \frac{\partial\phi}{\partial z}. \tag{2.22}$$

where the potential $\phi(x, z, t)$ for linear waves is expressed as:

$$\phi(x,z,t) = \frac{H\omega}{2k}\frac{\cosh(k(z+h))}{\cosh(kh)}\cos(kx-\omega t), \tag{2.23}$$

This potential, accurately captures the oscillatory motion of fluid particles beneath the wave surface. The wave generation utility, waves2Foam, then calculates the theoretical values of the velocity components (u, w) and surface elevation (η) at the inlet to generate the desired wave kinematics.

2.2.2. Non-linear waves

When switching from linear wave theory to higher-order Stokes wave theories, which are non-linear wave theories, several aspects of the wave generation process are modified to include second-order effects. While the general approach of using a velocity potential remains, the surface elevation, velocity potential and resulting velocity field are adjusted to account for higher-order harmonics and non-linear interactions. This transition occurs when the steepness of the wave increases (H/L) or the relative depth decreases (d/L) so that linear wave theory can no longer accurately represent the wave dynamics. Despite these changes, the fundamental relationship between the velocity field and the velocity potential remains unchanged (Dean & Dalrymple, 1991):

$$\mathbf{u} = \nabla\phi \tag{2.24}$$

This means that the velocity field is still deriveed from the gradient of the scalar potential, ϕ, ensuring that the flow remains irrotational ($\nabla \times \mathbf{u} = 0$). The adjustments when adopting higher-order stokes theory primarily involve changes to the surface elevation, velocity potential and therefore the resulting velocity components. The free surface elevation, η, now includes a correction term to account for wave asymmetry. The parameters are again obtained by using the dispersion relation (Appendix D.1). The modified surface elevation for Stokes second-order waves, is therefore given by:

$$\eta = \frac{H}{2}\cos(kx-\omega t+\phi) + k\frac{H^2}{4}\frac{3-\sigma^2}{4\sigma^3}\cos(2(kx-\omega t+\phi)) \tag{2.25}$$

And for Stokes fifth-order waves the surface elevation becomes (Dean & Dalrymple, 1991):

$$\begin{aligned}\eta = \frac{\lambda}{k}\cos(kx-\omega t+\phi) + \frac{\lambda^2 B_{22}+\lambda^4 B_{24}}{\lambda}\cos(2(kx-\omega t+\phi)) + \frac{\lambda^3 B_{33}+\lambda^5 B_{35}}{\lambda}\cos(3(kx-\omega t+\phi)) \\ + \frac{\lambda^4 B_{44}}{\lambda}\cos(4(kx-\omega t+\phi)) + \frac{\lambda^5 B_{55}}{\lambda}\cos(5(kx-\omega t+\phi))\end{aligned} \tag{2.26}$$

where:

- σ = radian wave frequency [m]
- λ = first order wave amplitude [s^{-1}]
- B_{xx} = coefficients in the fifth-order solution, obtained through a perturbation expansion by expanding the governing equations and boundary conditions in terms of the wave steepness.

The velocity potential, $\phi(x,z,t)$, in Stokes higher-order theories includes both first-order and higher-order components. For the second-order theory this becomes:

$$\phi(x,z,t) = \phi^{(1)}(x,z,t) + \phi^{(2)}(x,z,t), \tag{2.27}$$

When transitioning to Stokes fifth-order theory, the velocity potential, ϕ, becomes even more complex, as it includes up to fifth-order, $\phi^{(5)}$, terms to account for more non-linear interactions. The horizontal (u) and vertical (w) velocity components at the inlet are still derived from the velocity potential (ϕ) using the fundamental relationships in equation 2.22. Again, the wave generation utility (e.g. waves2Foam) calculates the theoretical values to generate the desired wave kinematics. The inclusion of higher-order terms ensures that the wave generation captures additional harmonics and more detailed asymmetry of the wave profile.

For highly non-linear waves or breaking waves with significant steepness, streamfunction wave theory can be used (Dean & Dalrymple, 1991). This theory better captures the non-linearity and complexity of the wave. The stream function also satisfies the incompressibility condition and follows the fundamental relationship between the velocity field and the velocity potential (Eq. 2.24). To solve for the surface elevation (η), the streamfunction is expanded to include higher-order terms, which help account for wave asymmetry and non-linear interactions. The surface elevation is then calculated from the streamfunction at each order using a Fourier series expansion.

$$\eta = h \sum_{j=1}^{N} E_j \cos(jkx - \omega t + \phi) \tag{2.28}$$

Where:

- E_j = Fourier coefficients of the computed free surface,
- N = number of terms in the Fourier expansion (chosen based on the wave steepness).

Eq. 2.28 represents a Fourier series for the surface elevation as a sum of sinusoidal terms. Where each term represents a different harmonic of the wave, with N depending on the wave steepness. This surface elevation is then reconstructed using a series of Fourier coefficients, which are determined based on the initial conditions.

2.3. Turbulence

When models upgrade to real-life dimensions or waves become steeper, the flow characteristics transition from laminar to turbulent. The turbulence generated by this plays a factor in the wave-structure interaction. It becomes essential to include a turbulence model in the simulation to capture this non-linear behavior. There are several turbulence models available in OpenFOAM, each offering a big trade-off between accuracy and computational cost (Sodja, 2007). The different turbulence modeling approaches are shown in Figure 2.2 and explained below, ranging from the most accurate and computationally expensive method (DNS) to the more approximate but computationally efficient RANS models.

1. **DNS (Direct Numerical Simulation):**

DNS solves the Navier-Stokes equations without any turbulence modeling. It offers the best accuracy by resolving all turbulence scales. However, it is so computationally expensive and impractical for large-scale or complex flows, that it is primarily used for fundamental research (Vijayakumar, 2023).

2. **LES (Large Eddy Simulation):**

LES is based on a scale separation. The large scales which contain the most energy and transport the flow properties are solved by solving the Navier-Stokes equation, but the small scales which behave universally are simply modeled (Ghasemian et al., 2017). The selection method is based on the size of the scale, if this is smaller than the grid size it is modeled. Les is more accurate than the above models, however the computational costs are high.

3. **Hybrid RANS-LES:**

The hybrid RANS-LES models (e.g. DES) is a blending technique between both methods. It uses the RANS models to model turbulence in regions close to walls or boundaries and LES in areas where large-scale turbulence dominates (Ghasemian et al., 2017).

4. **URANS (Unsteady RANS):**

URANS is an extension of RANS that allows for time-dependent changes in the flow field, making it capable of capturing unsteady phenomena. It still utilizes the Reynolds averaging of the Navier-Stokes equations but does not average the flow in time as strictly as RANS (Kazemian, n.d.). This means it captures transient phenomena while still modeling turbulence at a low computational cost.

5. **RANS (Reynolds-Averaged Navier-Stokes):**

RANS refers to a class of turbulence models (e.g. (RNG) $k - e$, (SST) $k - w$) that involve averaging the Navier-Stokes equations over time, to describe the mean flow behavior. It is typically used for steady-state simulations, where the time evolution of the flow is not important. Additionally, it has the highest model influence since none of the turbulence is directly resolved by the numerical simulation itself (Kazemian, n.d.).

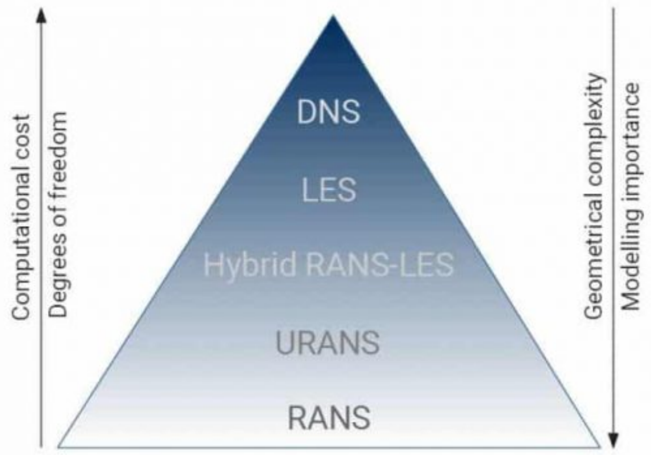


Figure 2.2: Different turbulence models (Vijayakumar, 2023).

The choice of turbulence model depends on the level of accuracy required, the computational resources available and the nature of the flow. For preliminary studies, RANS models are often sufficient to capture the basic flow behavior. Introducing turbulence modeling allows the simulation to better capture the enhanced momentum transfer caused by turbulent eddies. This results in a more accurate prediction of forces acting on the structure. Turbulence modeling typically leads to increased energy dissipation within the flow, which reduces wave run-up and forces on the monopile.

For this study, most of the runs will be done under laminar flow, however the influence of a RANS model will be evaluated. This introduces turbulent viscosity μ_t, which along with the standard dynamic viscosity results in the following relation:

$$\mu_{tot} = \mu + \mu_t \tag{2.29}$$

Which is then placed inside the Navier-Stokes Equation (Eq. 2.1). This research will use the $k - \omega$ - model to simulate the turbulence since it provides better accuracy near walls. If it turns out that the influence of turbulence is significant on the wave run-up then higher accuracy models should be used.

2.4. Model scaling and Normalisation

When scaling wave run-up on monopiles, it is essential to maintain similarity between the model and full-scale conditions. This can be achieved by using dimensionless numbers such as Reynolds number (Re), Froude number (Fr) and Keulegan-Carpenter number (KC) to ensure that the scaled model accurately replicates real-world hydrodynamic behavior. For wave-dominated phenomena like wave run-up, free surface effects and wave-induced forces, Froude scaling is typically used because gravity is dominant in these processes. However, this affects the Reynolds and Keulegan-Carpenter number, which may introduce differences in turbulence and viscous effects between the models (Journée & Massie, 2001).

The Reynolds number expresses the ratio of inertial forces to viscous forces. Reynolds scaling ensures that turbulence and viscous effects remain consistent between model and full scale. However, maintaining Reynolds similarity in large-scale flows is challenging due to high viscosity constraints in laboratories. Because if the same fluid is used, e.g. water, changes in L and U make it impossible to match Re. However, turbulence modeling methods (RANS, LES, DNS), explained in chapter 2.3 can help compensate for scaling differences. Reynolds number is given by:

$$Re = \frac{UL}{\nu} \tag{2.30}$$

where:

- U = characteristic velocity, typically wave celerity $[m/s]$
- L = characteristic length scale, typically wave height or monopile diameter $[m]$
- ν = kinematic viscosity of the fluid $[m/s]$

Froude number is the ratio between inertial and gravitational forces, defined as:

$$Fr = \frac{U}{\sqrt{gL}} \tag{2.31}$$

Froude scaling ensures that gravitational effects remain consistent between scaled models and full-scale structures. It is the dominant scaling law for free-surface flows, where gravity plays a major role. It is commonly used for experimental modeling in wave tanks (Andersen et al., 2020; de Vos et al., 2007; Tang et al., 2020). Therefore, Froude scaling laws of similitude can be seen in Table 2.1 below.

Table 2.1: Froude Scaling Laws of Similitude (Journée & Massie, 2001).

Parameter	Dimension	Similarity Relation
Length	L	λ
Mass	M	λ^3
Velocity	LT^{-1}	$\lambda^{1/2}$
Time	T	$\lambda^{1/2}$
Force	MLT^{-2}	λ^3
Moment	ML^2T^{-2}	λ^4

In wave-structure interaction, selecting the appropriate scaling method is essential for accurate modeling. The choice between Froude scaling and Reynolds scaling depends on the dominant forces acting on the structure, which depend on the wave regime. Figure 1.5 categorizes wave force regimes based on the non dimensional parameters H/D and $\pi D/\lambda$. This classification helps determine whether inertia, drag, or diffraction effects dominate and what scaling method is the most suitable.

In region I, inertia forces are the only significant contributor, with negligible drag and diffraction. Region II is characterized by wave diffraction effects, meaning that the structure is relative large compared to the wavelength, and forces must be calculated using diffraction theory. In region III, inertia remains dominant, but with some drag influence, requiring a careful consideration of the Keulegan-Carpenter number. Region IV is the transition zone where both diffraction and drag effects start to play a role, making it necessary to account for some breaking wave behavior and potential turbulence effects. In region V, both inertia and drag forces are significant and region VI is dominated entirely by drag forces, meaning that flow separation and turbulent wake formation govern the interaction. The deep-water breaking wave curve indicates conditions where wave breaking occurs (Negro et al., 2014).

To determine the appropriate scaling method, the wave-structure interaction regime needs to be determined. If inertia and gravity forces are dominant (region I, II, III), Froude scaling is the best choice. If drag effects are significant (regions V and VI), Reynolds scaling should be considered. In regions where both inertia and drag play a significant role (region IV and V), a combined approach using both scaling methods is necessary.

2.4.1. Scale Effects

Scaling models introduces scale effects, which are critical in wave modeling as they influence the accuracy and reliability of the model. Froude scaling ensures gravitational similarity, but it does not account for viscous and turbulent influences, which are governed by Reynolds scaling. A key consequence of this limitation is the increased wave damping observed in smaller-scale models, primarily due to lower Reynolds numbers. Since viscosity ν remains constant when scaling models with the same fluid, the Reynolds number (Eq. 2.30) decreases as the characteristic velocity U and length scale L are reduced. As a result, the Reynolds number in small-scale models is significantly lower than in real-world conditions, leading to exaggerated viscous effects.

A lower Reynolds number means that viscous forces become more dominant relative to inertial forces. Increased viscosity leads to an increase in wave damping, causing a reduction in wave heights as energy dissipates more rapidly. This effect is particularly relevant in wave-structure interactions, where higher viscous influence in small-scale models may result in an overestimation of drag forces. Understanding these effects is crucial for interpreting results from scaled models and ensuring that appropriate corrections or numerical approaches, such as turbulence modeling, are applied to account for these limitations.

2.4.2. Normalisation

In numerical modeling of wave-structure interactions, normalisation is essential for comparing different scale models and wave characteristics. It simplifies the analysis by removing scale-dependent variables and helps identify universal relationships. Run time normalisation involves dividing the total time t by the wave period T. This ensures that time is expressed as a dimensionless quantity relative to the wave period:

$$T_{\text{norm}} = \frac{t}{T} \tag{2.32}$$

Similarly, wave run-up (RU) and surface elevation (SE) are often normalised by dividing by the generated wave height H:

$$\text{RU,SE}_{\text{norm}} = \frac{\text{RU,SE}}{H} \tag{2.33}$$

In inertia-dominated flows, the total force can be normalised by a characteristic inertial force scale, defined as:

$$F_{\text{inertia}} = \rho g D^2 H \tag{2.34}$$

where ρ is the fluid density, g is the acceleration due to gravity, D is the diameter, and H is the wave height. The normalised force is then obtained by dividing the actual force by this characteristic inertia term to make it dimensionless.

For flows where viscous effects are dominant, the force is typically normalised by a characteristic viscous force scale, given by:

$$F_{\text{viscous}} = \mu U D \tag{2.35}$$

where μ is the dynamic viscosity of the fluid, U is a characteristic velocity, and D is the diameter. As a result, normalising forces in one way focuses on that dominant physical process, while reducing the effect of the other (either inertia or viscosity). This approach is needed to reduce differences caused by varying wave characteristics, making it easier to compare different simulations, independently of their scale. By normalising these parameters, dimensionless values are obtained that focus on the key physical behaviors. Which allows to compare and confirm results across different conditions and sizes.

3 Model Configuration

The methodology outlines the approach taken to create and validate the numerical model used in this study. It begins by describing the experiment that serves as a benchmark for validation in section 3.1. This is then followed by a detailed breakdown of the numerical model implementation in section 3.2. Key aspects such as mesh, boundary conditions, wave generation and absorption and probe placement are discussed to ensure accurate representation of the physical setup. Additionally, this section covers the functions and numerical solvers employed in the simulation. Lastly, the model validation process is presented, comparing the the numerical results with the experimental data from de Vos et al. (2007) to assess accuracy and reliability.

3.1. Initial scenario

The numerical method is validated by replicating an experimental study conducted in a wave flume at Aalborg University, Denmark from de Vos et al. (2007). The flume, measuring 30 m in length, 1.5 m in width, and 1 m in depth, is equipped with a piston-type wave paddle at one end to generate waves and an absorbing beach at the other end to minimize reflections. In the experiment they analyzed monopiles with either a monopile foundation or a cone shaped gravity type foundation. For this research only the monopile foundation is taken into account. The diameter of the pile is 0.12 m and the water depth at the monopile is 0.5 m. The offshore slope is fixed at 1:100. For the regular waves, wave heights vary between 0.01 and 0.26 m with a period varying between 0.40 and 2.78 s, corresponding with a wave steepness varying between 0.02 and 0.12 (de Vos et al., 2007). For the validation, the case of H = 0.12m and T = 1.6s was selected since for this case the most data was available. This experiment provides the data used to validate the accuracy of the numerical model. Figure 3.1 gives an overview of the test set up.

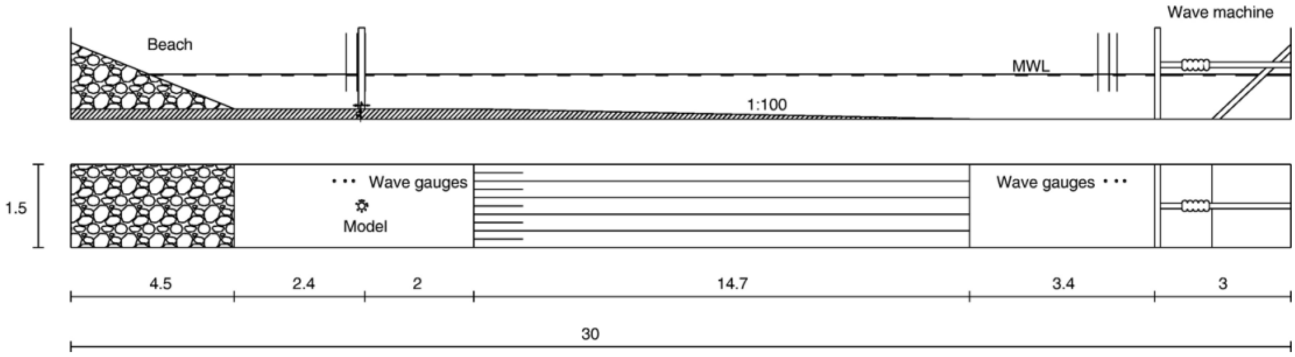


Figure 3.1: Test set up, values in meters (de Vos et al., 2007).

3.2. Model description

This section describes the numerical model developed to simulate wave run-up on monopiles using OpenFOAM and the waves2Foam toolbox. The model aims to capture complex wave-structure interactions with a high degree of physical accuracy. It includes the computational mesh, boundary conditions, wave generation and absorption methods, placement of wave gauges, solver settings and function objects. Each component has been chosen to ensure numerical stability, minimize reflection and accurately represent the physical conditions.

3.2.1. Mesh

In Computational Fluid Dynamics (CFD), the quality of the mesh plays a crucial role in the accuracy of the numerical simulation. A well-constructed mesh ensures that the numerical solver can accurately resolve the problem while maintaining computational efficiency. Key considerations when creating the mesh include grid resolution and cell shape. In wave simulations, the mesh must be sufficiently refined to accurately capture essential wave characteristics. However, it cannot be excessively fine, as this would significantly increase the computational cost and simulation time. In OpenFOAM, the mesh is defined in the *system* folder where the files *blockMeshDict* and *snappyHexMeshDict* control the mesh generation process. A detailed description of how meshes are defined in OpenFOAM can be found in Appendix A.

To determine the optimal mesh sizes for the model in all three spatial directions, a two-dimensional convergence study has been conducted to assess the accuracy of the mesh.

Convergency study Δz

To evaluate the impact of the mesh size in z-direction, simulations were conducted using Δx = L/100 and different Δz values while keeping all other numerical parameters constant. The study compares the surface elevation at two critical locations:

1. Right after leaving the inlet zone - to check wave generation accuracy.
2. Right in front of the monopile - to check the wave-structure interaction.

The simulations were run in the x-z plane of the validation model (Figure 3.1) for the following cases: Δz = H/20, H/25, and H/30 where H is the characteristic wave height. The precision of each size Δz was evaluated based on the surface elevation, wave dissipation across the domain and the computational cost. Figure 3.2 shows the surface elevation at both locations for different Δz values.

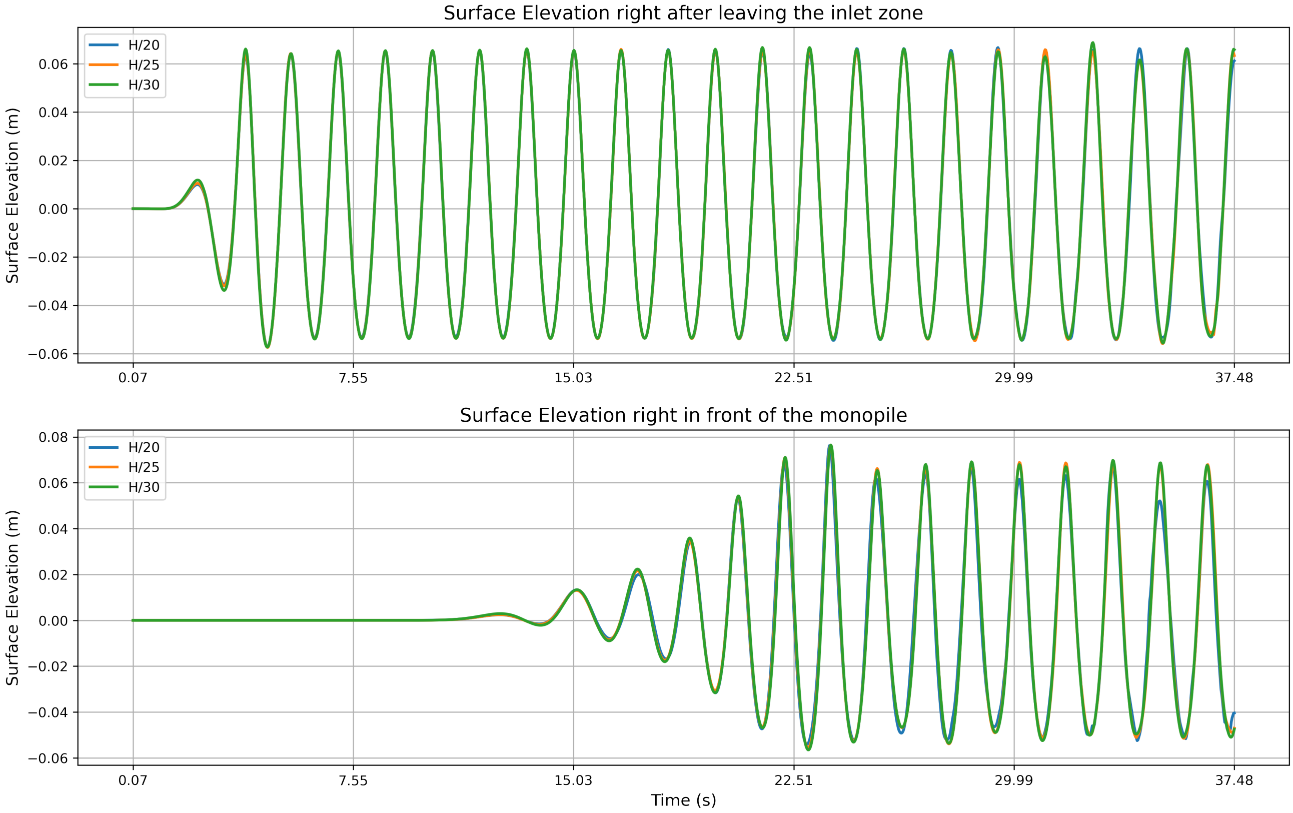


Figure 3.2: Comparison of surface elevation at two critical locations for different Δz, with Δx = L/100.

All cases show similar wave patterns right after the inlet zone with minor deviations. This indicates that the wave generation and propagation are captured almost perfectly for all cases. However, right in front of the monopile, H/20 shows deviation up to 0.02m compared to H/25 and H/30. Therefore, only H/25 and H/30 were analyzed. The absolute errors of the crests and troughs for the H/25 vs H/30 scenarios at both locations are shown in Table 3.1.

Table 3.1: Crest and Trough Errors for Δz H/25 vs H/30 at Different Gauge Locations.

Abs. Error Type	Inlet zone	Monopile
Max Crest Error	0.003817	0.001776
Avg Crest Error	0.001083	0.000907
Max Trough Error	0.001372	0.001650
Avg Trough Error	0.000443	0.000714

Since the absolute errors are minimal and the relative errors remain below 3.5% at the monopile, and because reducing the mesh size from H/25 to H/30 in the z-direction increases the number of cells by 20% and the computational time by approximately the same amount, a mesh size of H/25 is used for all simulations.

Convergency study Δx

Next to the vertical mesh resolution, the horizontal mesh resolution (Δx) also plays a critical role in capturing the wave propagation. Again, a coarse resolution leads to wave underestimation and excessive refinement increases computational cost. The optimal Δx is determined by conducting two-dimensional simulations using a fixed vertical mesh size (Δz = H/25), while testing three different horizontal resolutions: $L/50$, $L/66.67$ and $L/100$, where L represents the characteristic wavelength. Figure 3.3 shows the surface elevation of the three different mesh sizes at the two critical locations.

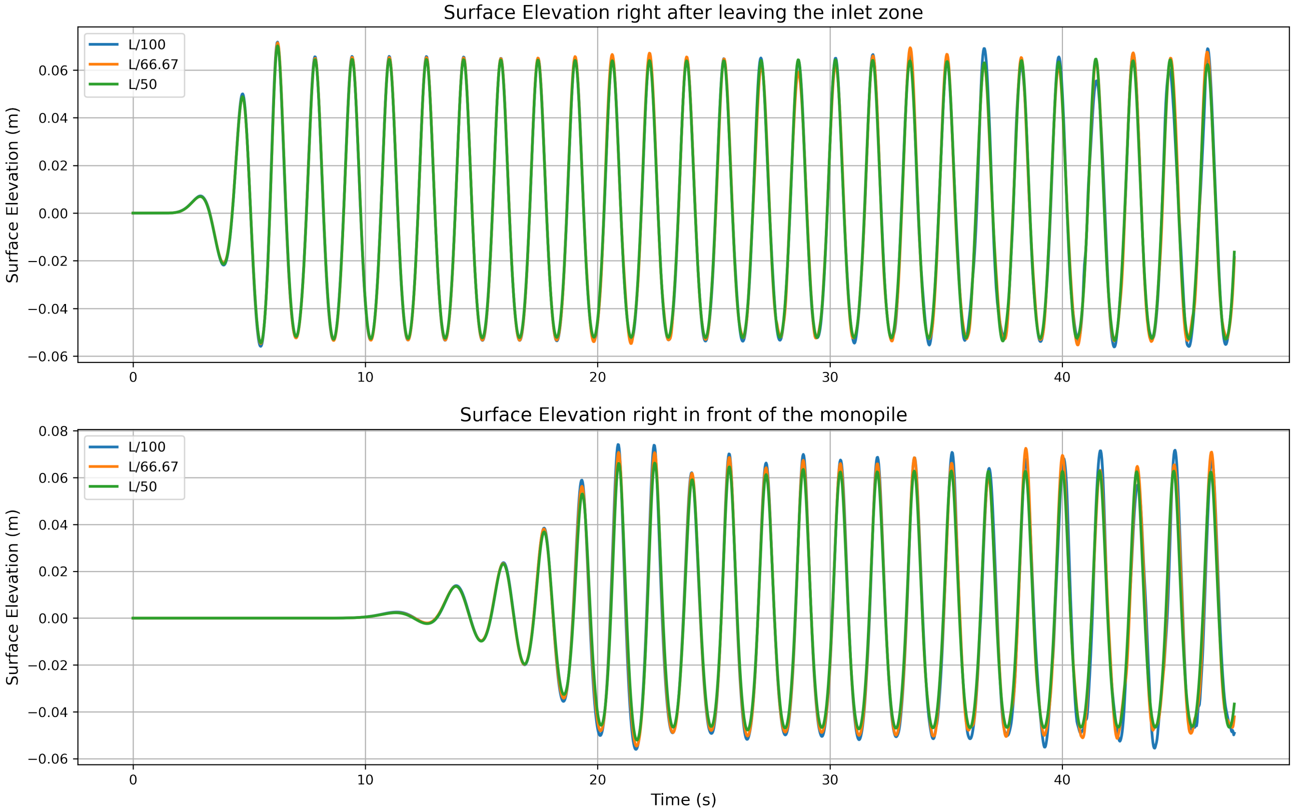


Figure 3.3: Comparison of surface elevation at two critical locations for different Δx, with Δz = H/25.

L/50 significantly underestimates the wave elevation, especially near the monopile. The finer resolutions (L/66.67 and L/100) show higher wave crests and deeper troughs, indicating that L/50 introduces numerical dissipation (Appendix A.1.1). However, OpenFOAM, also has the possibility for local refinement by using *snappyHexMesh* (Appendix A). This improves accuracy where needed without excessively increasing the total number of cells. Therefore, the L/100 case was also compared to the L/50 case with a refinement of level 1 in MSL (0m) $\pm H$ (Scholte, 2020; Tang et al., 2020). These results can be seen in Figure 3.4.

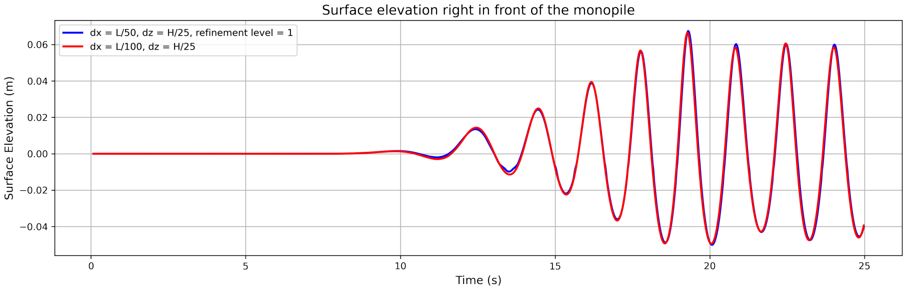


Figure 3.4: Comparison of surface elevation for L/100 and L/50 + refinement level 1.

Despite the L/50 resolution underestimating waves, the introduction of snappyHexMesh refinement in a region of $\pm H$ around the free surface significantly improves accuracy, making the results nearly identical to L/100. The errors can be seen in the following Table.

Table 3.2: Crest and Trough Errors for Δx=L/100 vs Δx=L/50 + refinement level = 1.

Abs. Error Type	Error (m)
Avg Crest (Peak) Error (t $\geq$ 15s)	0.00031
Avg Trough Error (t $\geq$ 15s)	0.00022

The refinement is done by dividing each cell in the refinement region by a factor of 2, resulting in a local resolution of $\Delta x = L/100$ and $\Delta z = H/50$. This finer resolution better captures the wave steepness, surface gradients and flow characteristics near the free surface. In contrast, refining the entire domain to this fine level would increase the total number of cells from 9,490,848 to 32,298,000. Since the errors are significantly small (<0.5%), the final configuration of Δx is $L/50$ with a local refinement level 1 applied in a vertical band of $\pm H$ around the free surface.

Convergency study of Δy

In contrast to the x- and z-direction, where wave propagation requires fine resolution, the y-direction represents the spanwise direction. Since in this study only regular waves were taken into account, excessive refinement in the y-direction is unnecessary. Instead, the goal is to maintain a structured grid with square or cubical cells, since OpenFOAM performs optimally in these scenarios. To ensure this, the smallest block in the y-direction of the domain was analyzed. This block extends from 0.7076 to 0.7924 m. The difference therefore is 0.08485 m. Given the x-direction cell size of $\Delta x = 0.03$, we evaluated three possible discretization strategies (1, 2, or 3 cells):

$$\frac{\Delta y}{1} = 0.08485m$$

$$\frac{\Delta y}{2} = 0.04243m$$

$$\frac{\Delta y}{3} = 0.02828m$$

The corresponding ratios relative to Δx are 2.83, 1.41, and 0.94 respectively. Since OpenFOAM performs best with square elements, the optimal choice would have been $\frac{\Delta y}{3} = 0.02828m$. However, compared to the case of $\frac{\Delta y}{3}$ this would increase the number of cells by 50%. Since OpenFOAM performs very well with cubical cells and since only regular waves are simulated the mesh size in the y-direction was chosen as $\frac{\Delta y}{2} = 0.04243m$ which is equal to 1.41 * Δx.

3.2.2. Final mesh

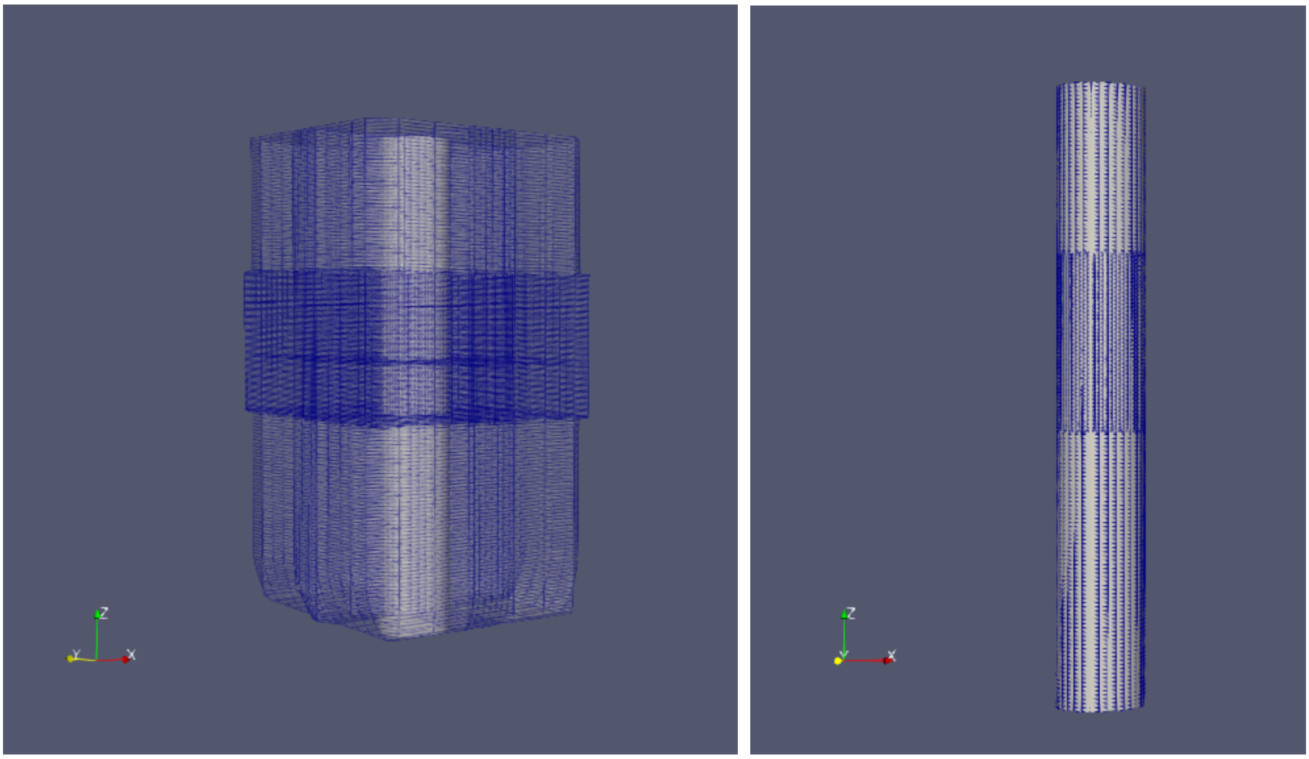
Based on the convergency studies above. The final mesh sizes in the three spatial directions are:

$$\Delta x = \frac{L}{50} + \text{ a refinement of level 1 in a range of } \pm H \text{ around the free surface}$$

$$\Delta y = 1.41\Delta x$$

$$\Delta z = \frac{H}{25}$$

Figure 3.5 shows the mesh around and on the monopile, based on the previously described resolutions. As shown, an additional refinement level of 2 was applied around the monopile in x- and y-direction to better capture the wave-structure interactions. The use of smaller cells in this area results in a more accurate prediction of wave run-up and the associated forces. This refinement significantly improved the overall accuracy of the simulation results without requiring excessive refinement throughout the entire domain.



(a)

(b)

Figure 3.5: Local mesh refinement around the monopile. **(a)** Clip of the mesh surrounding the monopile (close up of the monopile from Figure 3.6). **(b)** Mesh on the monopile surface.

During the simulations, the outlet boundary was placed far downstream from the monopile to ensure that reflected waves would not interfere with the incident wave field. This decision was based on an initial overestimation of the reflection coefficient, which was believed to indicate significant reflection at the outlet. However, it was later found that wave reflections were minimal, when using the waves2Foam outlet zone. As a result, in future simulations, the outlet can be positioned much closer to the monopile without losing accuracy. This change would reduce the total number of cells in the domain, thereby decreasing simulation time.

The final mesh is shown in Figure 3.6. Everything is identical to the initial scenario (Fig. 3.1) but the mesh also includes an inlet zone of 2L (=6m) in front of the wave maker from the experiment, allowing the wave to develop and propagate 6.4 m before interacting with the bottom slope. Additionally, the location of the wave-absorbing beach was extended to a length of 8L (=24m) including the outlet zone of 2L.

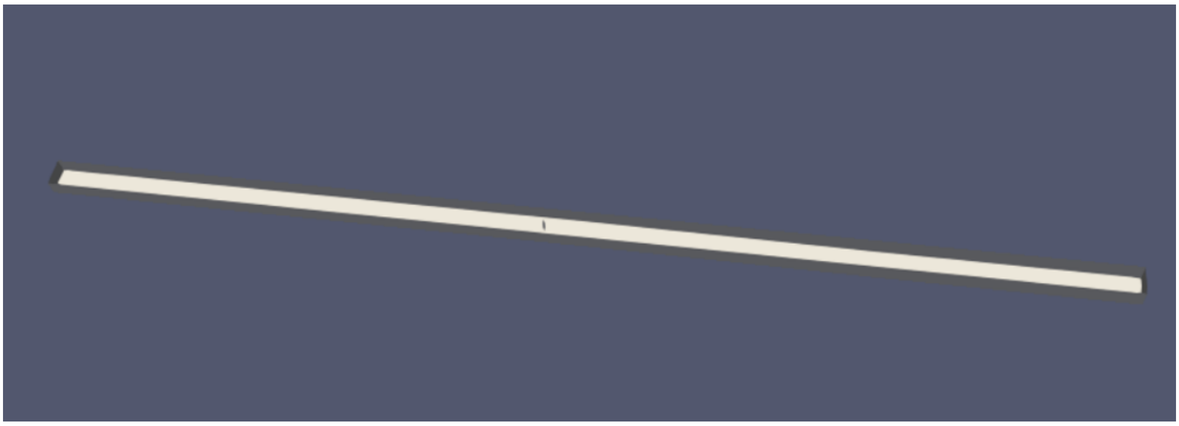


Figure 3.6: Final mesh for the small scale model (D0.12) with in the middle the monopile.

3.2.3. Boundary conditions

The boundary conditions for each field and boundary are defined in the 0-time directory used by OpenFOAM and are detailed in Table 3.3. These boundary conditions are chosen to accurately represent the physical scenario.

Table 3.3: Boundary conditions for the simulation.

Boundary	*alpha.water*	*U*	$\mathbf{p_{rgh}}$
inlet	zeroGradient	zeroGradient	zeroGradient
outlet	zeroGradient	zeroGradient	zeroGradient
sides	zeroGradient	slip	zeroGradient
atmosphere	inletOutlet	pressureInletOutletVelocity	totalPressure
bottom	zeroGradient	slip	zeroGradient
monopile	zeroGradient	fixedValue (0 0 0)	zeroGradient

Below is a brief description of each boundary condition:

1. **zeroGradient**:
   - This boundary condition assumes the gradient of a variable is zero at the boundary:

   $$\frac{\partial \phi}{\partial n} = 0 \tag{3.1}$$

   Where ϕ represents the variable (e.g., $\alpha.water$, U, or p_{rgh}) and n is the normal direction to the boundary. It ensures that there is no net flux of the quantity across the boundary. This condition maintains continuity by making the value at the boundary equal to the value in the adjacent interior cell, allowing the variable to transition smoothly across the boundary (OpenFOAM, 2025i).

2. **inletOutlet**:
   - This condition switches behavior based on the flow direction (where the direction of the inflow is normal to the boundary and into the domain. The outflow direction is normal and out of the domain):

   $$\phi = \begin{cases} \phi_{\text{fixed}}, & \text{if inflow} \\ \frac{\partial \phi}{\partial n} = 0, & \text{if outflow} \end{cases} \tag{3.2}$$

   For inflow, it applies a fixed value (air: $alpha.water$ = 0) and a zero-gradient condition during outflow. Ensuring that air enters the domain when water recedes and water exits smoothly when waves propagate upward (OpenFOAM, 2025b).

3. **slip**:
   - This boundary condition enforces zero normal velocity and allows tangential velocity:

   $$U \cdot n = 0, \quad \frac{\partial U_t}{\partial n} = 0 \tag{3.3}$$

   where U_t is the tangential velocity. It models a frictionless surface (OpenFOAM, 2025f).

4. **pressureInletOutletVelocity**:
   - This condition computes the velocity at the boundary using the pressure gradient:
   $$\frac{\partial U}{\partial n} = \frac{1}{\rho}\frac{\partial p_{rgh}}{\partial n} \tag{3.4}$$
   It ensures a realistic relationship between pressure and velocity and allows air movement to match the wave-induced pressure dynamics, maintaining wave shape and energy (OpenFOAM, 2025e).
5. **fixedValue**:
   - The fixedValue condition prescribes a constant value for the variable at the boundary:
   $$\phi = \phi_{\text{fixed}} \tag{3.5}$$
   In this case setting velocity to zero to represent a stationary structure (OpenFOAM, 2025a).
6. **totalPressure**:
   - This condition accounts for both static and dynamic pressures at the boundary:
   $$p = \begin{cases} p_0, & \text{for outflow} \\ p_0 - \frac{1}{2}|\mathbf{u}|^2, & \text{for inflow (incompressible).} \end{cases} \tag{3.6}$$
   where p_{total} is the prescribed total pressure, ρ is the fluid density, and $|\mathbf{U}|$ is the velocity magnitude. The p_0 value is set to 0, which serves as a gauge pressure relative to atmospheric pressure. It allows flow to enter or exit based on pressure gradients (OpenFOAM, 2025h).

3.2.4. Wave generation and absorption

In waves2Foam, wave generation and absorption settings are defined in the *waveProperties* file. This includes the specification of the wave type and parameters (e.g. H, T, direction) used in the simulation, as well as the configuration of the relaxation zones, including their location, length, and relaxation method. In this study, the wave types used are Stokes fifth-order and stream function waves. For the relaxation zones, the rectangular relaxation shape and an exponential weighting function are applied. These configurations are crucial for accurately generating the desired wave conditions and minimizing unwanted reflections at domain boundaries. A detailed explanation of how these inputs are defined and how OpenFOAM handles relaxation zones is provided in Appendix A.2. Below, a convergence study on the relaxation zone lengths is presented to determine their optimal length. With the mesh resolution from section 3.2.1, $H = 0.12m$ and $T = 1.6$.

For the inlet relaxation zone, wave generation simulations were conducted using three different lengths: 1,2, and 3 times the wavelength (L). The criteria for the optimal wavelength was whether increasing the inlet zone length led to a more stable and accurate wave profile. The results at half and the full length of the inlet zone are visualized in Figures 3.7 and 3.9, with the absolute errors in Figures 3.8 and 3.10.

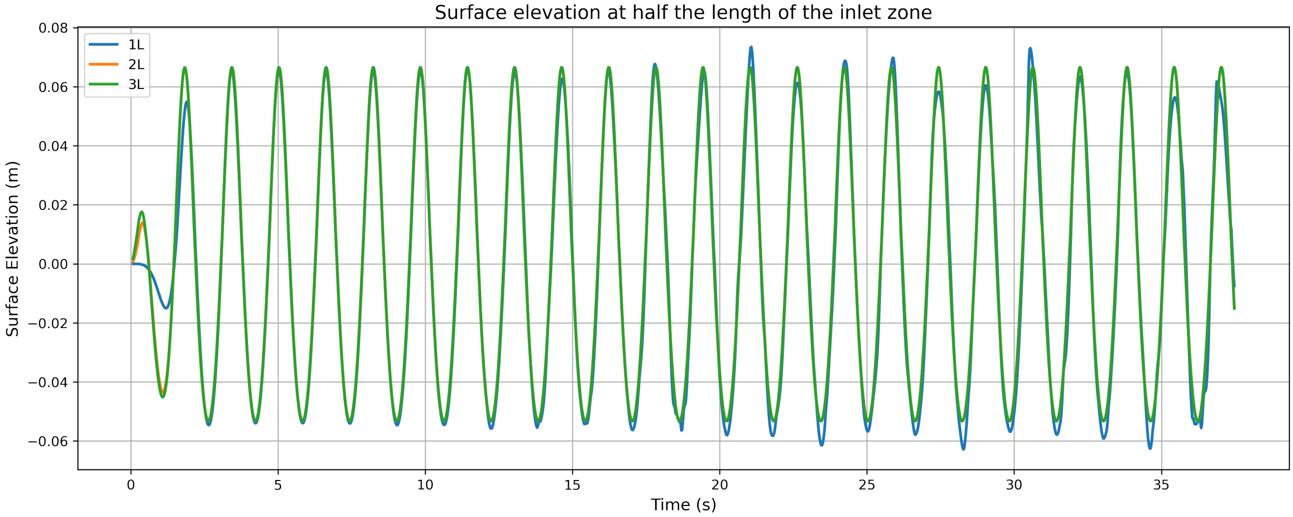


Figure 3.7: Comparison of surface elevation at half the length of the inlet zone.

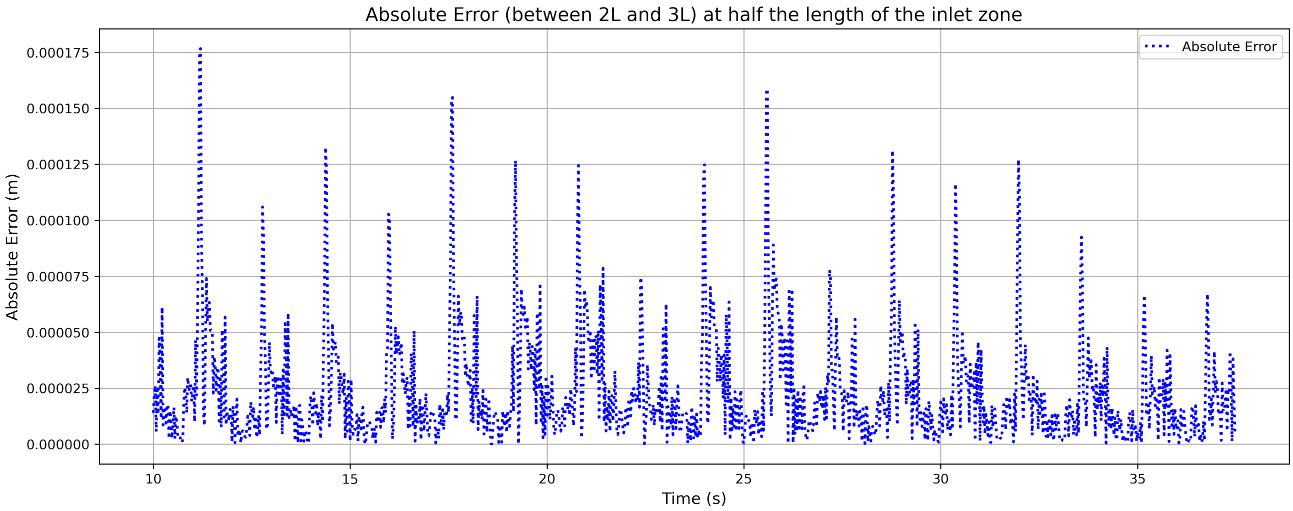


Figure 3.8: Absolute error between 2L and 3L at half the length of the inlet zone.

As can be seen in Figures 3.7 and 3.9, an inlet zone with a length of 1L differs significantly from 2L and 3L and does not generate the wave characteristics specified in the *waveProperties* file. On the other hand, the absolute errors, between 2L and 3L are negligible small across both gauge locations. At half the length of the inlet zone, the absolute error is of the order of $10-4$ to $10-5$. At the end of the inlet zone the error is slightly larger, averaging at $10-4$ but peaking around $10-3$ which is still relatively small. The increase in absolute error towards the end of the inlet zone does suggest that longer inlet zones help stabilize wave profiles before they enter the main computational domain. Since the differences between 2L and 3L are minimal, increasing the length does not provide a significant improvement in wave generation. However, it does give a significant increase in computational cost since the number of cells and the length of the domain increases. Therefore, the optimal inlet length is 2L, as it balances wave stability and computational cost.

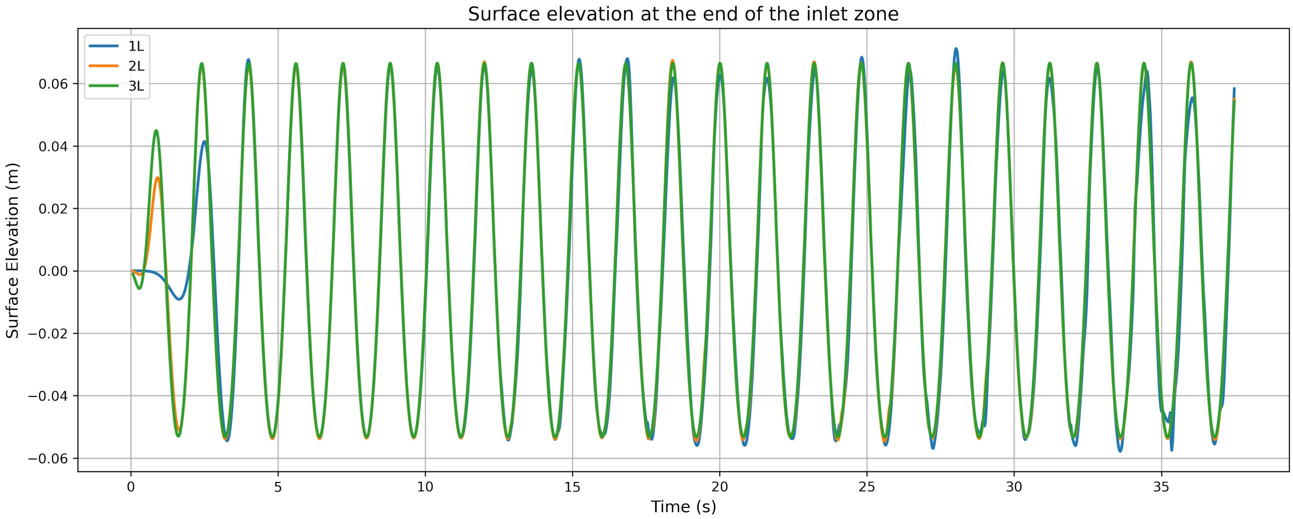


Figure 3.9: Comparison of surface elevation at the end of the inlet zone.

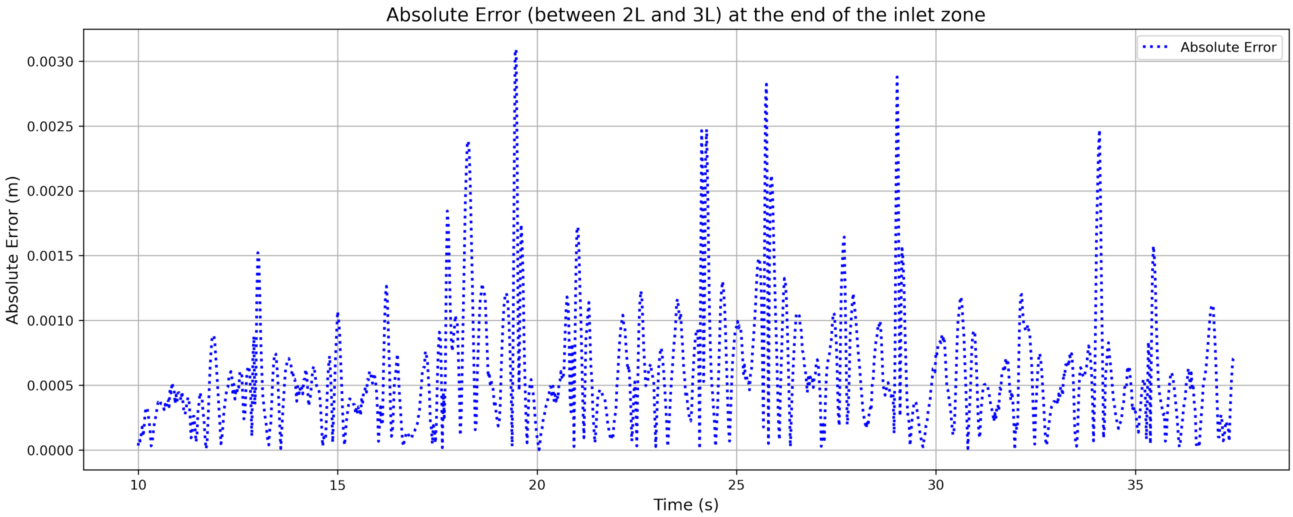


Figure 3.10: Absolute error between 2L and 3L at the end of the inlet zone.

The length of the outlet zone is determined by conducting a reflection analysis using the Mansard and Funke method (Mansard & Funke, 1980). This method requires amplitude and phase information at different gauge locations. As the simulations focus on regular wave propagation, it is logical to apply the Mansard and Funke (1980) in a monochromatic framework, where the dominant frequency is extracted from the wave signal. The surface elevation data from the simulations is first processed using a Fast Fourier Transform (FFT) to extract the dominant frequency, amplitude, and phase at each wave gauge location. The extracted amplitudes and phases are then used as input parameters for the Mansard and Funke (1980) method, which decomposes the incident and reflected wave components. The reflection coefficient is derived using the following formula:

$$\text{Reflection Coefficient} = \frac{|Z_r|}{|Z_i|} \tag{3.7}$$

Where Z_i is the incident wave complex amplitude and Z_r is the reflected wave complex amplitude. The full Mansard and Funke (1980) method within a monochromatic framework can be found in Appendix C. The reflection analysis is conducted across four different outlet zone lengths: 1L, 2L, 3L, and 4L. The reflection coefficient indicates how effectively each zone length minimizes wave reflection. Table 3.4 shows the different amplitudes and phases for each gauge and the corresponding reflection coefficient.

Table 3.4: Amplitude, phase, and reflection coefficient values for different absorption zone lengths.

	Gauge 1		Gauge 2		Gauge 3		Refl. Coeff.
Scenario	**Ampl.**	**Phase (rad)**	**Ampl.**	**Phase (rad)**	**Ampl.**	**Phase (rad)**	
1L	3.818	-1.660	4.177	-3.058	3.720	1.205	0.057
2L	4.012	-1.732	4.034	3.136	4.000	1.167	0.003
3L	4.024	-1.887	4.026	3.013	4.010	1.039	0.012
4L	4.082	-2.042	4.134	2.854	4.058	0.878	0.013

When comparing the reflection coefficients the 2L configuration achieves the lowest reflection coefficient (0.003). It can also be seen that increasing the length beyond 2L does not result in any improvements. Therefore, both the inlet and outlet zone lengths are equal to 2 * wavelength.

In addition to the numerical values presented in Table 3.4, the wave damping is also clearly visible in Figure 3.11. The image shows the pressure field at $t = 49\,\text{s}$, with a wave height of $H = 3.5\,\text{m}$, period $T = 7\,\text{s}$, and wavelength $L = 70\,\text{m}$ using the turbulent model (section 4.4) with laminar flow. The low reflection coefficient of 0.003 corresponds well with the visual observation that the wave pressure fades toward the outlet. The pressure field decreases towards the outlet boundary, indicating that the wave energy has been effectively absorbed and unwanted reflections have been successfully minimized.

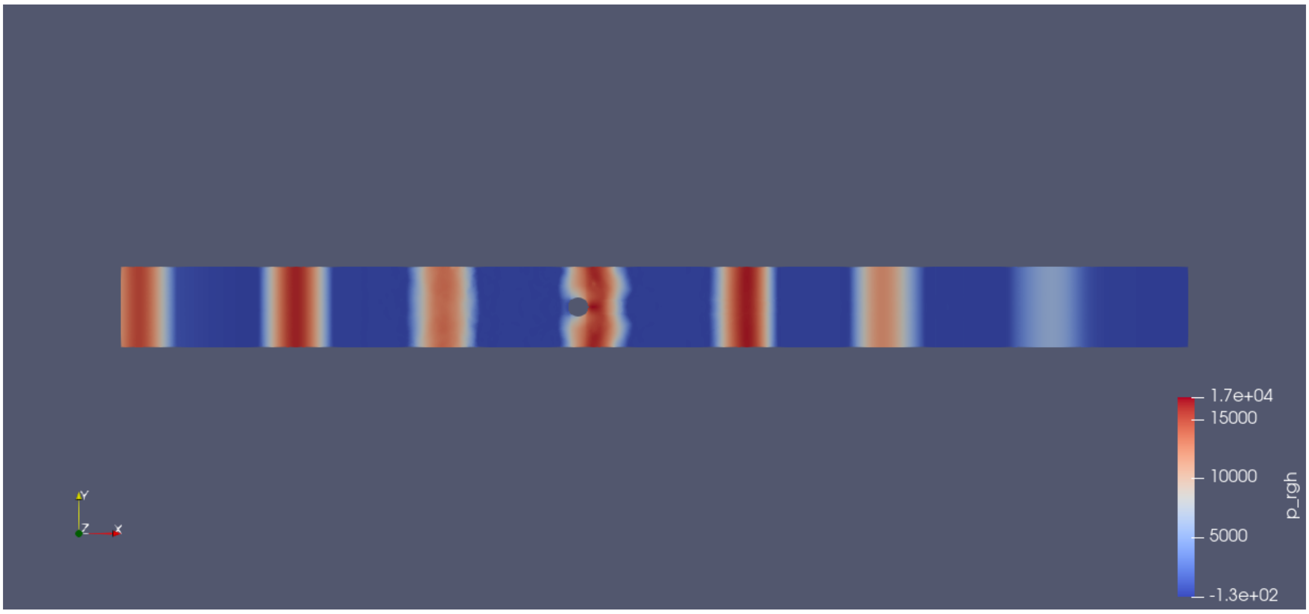


Figure 3.11: Visualization of the pressure field (p_{rgh}) at $t = 49$ s..

3.2.5. Wave gauges and probes

Waves2Foam has a built-in utility that allows for the definition of probes and wave gauges, called *waveGaugesNProbes*. These are essential tools in CFD simulations, for extracting critical flow parameters at predefined locations within the computational domain. Within waves2Foam two types of gauges can be specified. Either wave gauges, which extract the alpha.water parameter to determine the surface elevation. Or probes, which can extract every parameter that the simulation solves for (e.g. velocity, pressure, turbulence parameters).

In Table 3.5 the locations (x, y, z) and the function of the wave gauges for the validation model are given. Where X = -3, 6 indicates that the wave gauges are positioned between -3 and 6 m, with N gauges distributed at equal intervals.

Table 3.5: Wave Gauge Locations.

Name	X (m)	Y (m)	Z (m)	N	Function
surfaceElevation1	-3, 6	0.75	-0.2, 0.6	4	Check generated waves
surfaceElevation2	21, 23	0.75	-0.2, 0.6	10	Check reflection of monopile
surfaceElevation3	22.1, 24.1	1.125	-0.2, 0.6	10	Validate model against existing data
surfaceElevation4	39.5, 44	0.75	-0.2, 0.6	6	Check reflection of damping zone
surfaceElevation5	23.2, 30.2	0.75	-0.2, 0.6	7	Check wake
Name	**Radius**	**Center**	**Z (m)**	**N**	**Function**
monopileElevation	0.062	(23.1, 0.75, -0.2)	-0.2,0.6	16	Check wave run-up

The gauges around the monopile are positioned 2mm in front of its surface, similar to the validation experiment (de Vos et al., 2007). They are arranged in a counter-clockwise direction, starting at the rear of the monopile ($180°$) and spaced evenly at $22.5°$ intervals around the circumference. The placing can be seen in Figure 3.12.

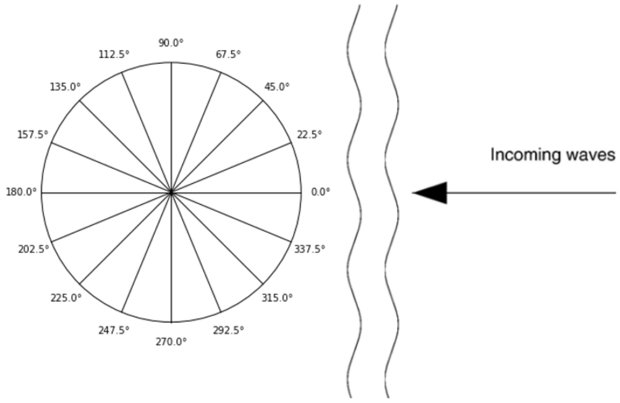


Figure 3.12: Position of the wave gauges around the monopile: 16 wave gauges placed 2mm in front of the pile surface.

3.2.6. Functions

OpenFOAM has a built-in library to compute the forces and moments acting on solid bodies. This is achieved using the *libforces.so* library, which integrates pressure and shear stress over the surface of the object. Within the *libforces.so* library, a specific *patch* must be defined. This refers to a named boundary surface in the mesh over which the force and moment calculations are performed. In this study, the relevant patch corresponds to the surface of the monopile. Additionally, to compute the moment (Eq. 3.9), a reference point must be specified as the center of rotation. Here, the bottom of the monopile is used as the moment center to capture the moment generated around its base. To enable this functionality, the forces function, shown below, must be added to the `controlDict` file under functions.

```
forces
{
    type                forces;
    functionObjectLibs ("libforces.so");
    outputInterval      <label>;

    patches            (<list of patch names>);
    CofR               (x y z);     \\ Centre of rotation for moment calculations
}
```

Code Fragment 3.1: Common basic set of input entries

The total force on the body is then calculated using:

$$F = \int_S (-p\mathbf{n} + \boldsymbol{\tau} \cdot \mathbf{n})\, dS \tag{3.8}$$

where:

- F is the total force on the body $[N]$
- p is the pressure acting normal to the surface $[Pa]$
- $\mathbf{n}$ is the unit normal vector of the surface $[-]$
- $\boldsymbol{\tau}$ is the viscous stress tensor $[Pa]$
- S is the surface area where the force is computed $[m^2]$

The pressure force component, given by $-p\mathbf{n}$, arises from the normal pressure acting on the surface. The viscous force component, represented by $\boldsymbol{\tau} \cdot \mathbf{n}$, is due to tangential shear stresses induced by fluid viscosity. Additionally, the total moment acting on the body is calculated as:

$$M = \int_S \mathbf{r} \times (-p\mathbf{n} + \boldsymbol{\tau} \cdot \mathbf{n})\, dS \tag{3.9}$$

where:

- $\mathbf{r}$ is the position vector relative to the specified center of rotation $[m]$

3.2.7. Numerical Discretization and Solver Settings

Numerical accuracy and solver performance in OpenFOAM are governed by two files: *fvSchemes* and *fvSolution*. These files define how the governing equations are discretized and solved using the finite volume method. Proper configuration of these settings is crucial for achieving accuracy, stability and computational efficiency.

The *fvSchemes* file, located in the *system* directory, specifies the numerical schemes used for spatial and temporal discretization. It defines how derivatives such as gradients (∇), divergence ($\nabla\cdot$) and Laplacian (∇^2) terms are calculated, as well as how field values are interpolated between mesh points (OpenFOAM, 2025d). The set of terms, for which the numerical schemes must be specified, are subdivided within the *fvSchemes* dictionary as categorized in Table 3.6.

Table 3.6: Mathematical terms associated with OpenFOAM schemes.

Scheme Name	Mathematical Meaning
ddtSchemes	First and second time derivatives $\partial/\partial t, \partial^2/\partial t^2$
gradSchemes	Specifies the calculation of gradient terms ∇
divSchemes	Determines the divergence discretization $\nabla\cdot$
laplacianSchemes	Controls Laplacian ∇^2 used in diffusion terms
interpolationSchemes	Defines how values are interpolated between points in mesh
snGradSchemes	Component of gradient normal to a cell face

OpenFOAM provides flexibility in selecting discretization methods, ranging from first-order upwind schemes that ensure numerical stability but introduce diffusion, to higher-order schemes that enhance accuracy while requiring more careful attention to stability. The choice of these schemes affects the trade-off between accuracy and computational cost (OpenFOAM, 2025d). The list of all available schemes in OpenFOAM can be found on the official website OpenFOAM (2025d). Table 3.7 provides an overview of the used schemes in the simulations.

Table 3.7: Selected numerical schemes for wave run-up simulation in OpenFOAM.

Scheme Name	Selected Scheme
`ddtSchemes`	Euler
`gradSchemes`	Gauss linear
`divSchemes`	div(rhoPhi, U) - Gauss linearUpwind Gauss linear phi
	div(((rho*nuEff)*dev2(T(grad(U))))) - Gauss linear
`laplacianSchemes`	Gauss linear corrected
`interpolationSchemes`	Linear
`snGradSchemes`	Corrected

In the wave run-up simulation, ddtSchemes (Euler), gradSchemes (Gauss linear), laplacianSchemes (Gauss linear corrected) and interpolationSchemes (linear) are selected by default. These schemes are typical schemes for multiphase simulations using VOF and provide a balance between accuracy and stability while ensuring computational efficiency (Alsudani, 2024; Edesess, 2018). Euler is a stable first-order time scheme suitable for transient simulations, while Gauss linear provides second-order accuracy for gradients and interpolations. The Laplacian terms make use of corrected schemes to improve the accuracy for non-orthogonal meshes, which is essential for complex monopile geometries. The divSchemes are carefully selected to ensure accurate wave propagation: linearUpwind is used for velocity convection to reduce numerical oscillations by introducing a slight upwind bias(Edesess, 2018). The viscous stress tensor (rho*nuEff term) is Gaussian linear, ensuring smooth and accurate diffusion calculations. While minimizing numerical diffusion and instability, these choices allow the model to properly capture wave propagation, breaking and interaction with the monopile.

The *fvSolution* file, also located in the *system* directory, manages the solution strategy, including solvers, tolerances and iterative controls. It consists of two key entries:

- *Solvers:* Defines the numerical solver for all variables. Each solver entry specifies the type of solver, solver tolerance, and additional parameters such as the preconditioner and/or smoother.
- *Pressure-velocity coupling settings:* Defines the coupling method (PISO, SIMPLE, PIMPLE).

In this case, solvers are set for the volume fraction field (*alpha.water*), pressure (*p_rgh*), velocity (*U*), and if turbulence is active, also for turbulence variables *k* and *omega*. For solving these variables, different numerical solvers are used depending on the parameter. The GAMG (generalized Algebraic Multigrid) solver is used for the pressure, which helps speed up convergence, especially for large-scale simulations. The PCG (Preconditioned Conjugate Gradient) solver is used for pressure correction fields (*pcorr* and *pcorrFinal*) to ensure mass conservation and stability in pressure-velocity coupling. The PBiCG (Preconditioned Bi-Conjugate Gradient) solver is used for velocity and turbulence variables, which is well-suited for asymmetric systems like those found in free-surface flows (OpenFOAM, 2025g).

In addition to solvers and smoothers, preconditioners play a crucial role in ensuring numerical stability and efficient convergence. Smoothers, such as DIC (Diagonal Incomplete Cholesky) and symGaussSeidel (Symmetric Gauss-Seidel), help reduce numerical errors and stabilize iterative solvers, particularly for pressure-based systems. Preconditioners, such as DILU (Diagonal Incomplete LU decomposition), accelerate convergence by improving matrix conditioning, making them essential for handling the strong asymmetry and stiffness of momentum and turbulence equations (OpenFOAM, 2025g). All available solvers can be found at OpenFOAM (2025g).

Another essential aspect of solver performance is tolerance settings. The absolute tolerance (*tolerance*) defines the minimum acceptable residual error before the solver stops iterating, ensuring accuracy in pressure and velocity fields. For example, low tolerance values like 1e-10 for pressure and 1e-7 for alpha.water ensure a precise resolution of free-surface behavior and wave-induced forces. The relative tolerance (*relTol*) determines how much the residual error must decrease within each iteration before convergence is assumed. A value of 0 ensures that the solver does not stop prematurely, maintaining accuracy throughout the simulation. An overview of all settings can be found in Table 3.8.

Table 3.8: Solver settings from `fvSolution`.

Variable	Solver	Preconditioner	Tolerance	relTol
alpha.water.*	smoothSolver	symGaussSeidel	1×10^{-7}	0
pcorr	PCG	DIC	1×10^{-10}	0
pcorrFinal	PCG	DIC	1×10^{-10}	0
p_rgh	GAMG	DIC	1×10^{-10}	0
p_rghFinal	GAMG	DIC	1×10^{-10}	0
(U | k | omega)	PBiCG	DILU	1×10^{-10}	0
(U | k | omega)Final	PBiCG	DILU	1×10^{-10}	0

The *fvSolution* file defines not only the solver settings, but also the parameters used for pressure-velocity coupling. For this study, the *PIMPLE* algorithm is used, as explained in section 2.1.3. The mandatory corrector parameters for the PIMPLE algorithm and their explanation are given in Appendix A.2.3.

Time stepping

Determining an appropriate time step is crucial for achieving both numerical stability and computational efficiency. A time step that is too large can lead to skipped cells, causing loss of information and inaccuracies, while a time step that is too small significantly increases computational time. OpenFOAM employs an *adjustable time step* function that dynamically modifies the time step to optimize the trade-off between accuracy and efficiency.

In wave simulations, such as wave run-up on monopiles, high-velocity gradients occur near the free surface, making it essential to control the time step adaptively. By enforcing a constraint on the Courant-Friedrichs-Lewy (CFL) condition, the simulation avoids skipping cells while maintaining computational speed. This method enables accurate wave propagation and interaction with the monopile while significantly reducing unnecessary computations in lower-velocity regions.

The Courant number (C_O) is a dimensionless quantity and is defined as:

$$C_O = \frac{||\mathbf{u}|| \cdot \Delta t}{\Delta x} \leq C_{\text{max}} \tag{3.10}$$

The maximum Courant number, C_{max}, determines the allowable timestep. If $C_O > 1$, the numerical scheme may skip cells, leading to errors, which is shown in Figure 3.13. By ensuring $C_O \leq 0.6$ (Chen et al., 2021; Edesess, 2018), the solver avoids information loss while optimizing computational speed. This is particularly beneficial for long-duration wave simulations, where maintaining accuracy without excessive computation time is critical.

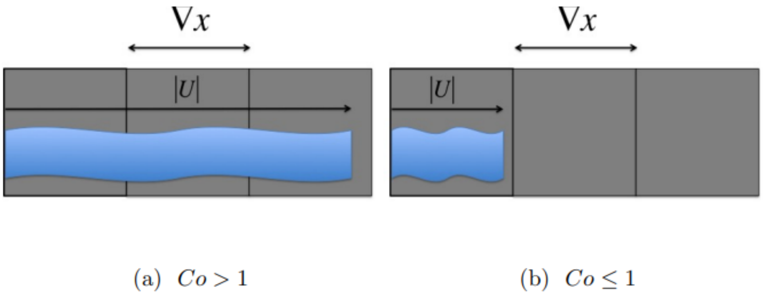


Figure 3.13: Distance that fluid flow travels per time step, depending on the Courant number (Edesess, 2018).

3.3. Model validation

In this chapter, the performance of the developed model is validated against experimental data to assess its accuracy and reliability in simulating wave run-up on monopiles. The validation is carried out by comparing the results from the simulation against the data from the validation experiment for the situation as mentioned in chapter 3.1. The simulation parameters are listed in Table 3.9. The validation data is provided by de Vos et al. (2007). However, since the experiment was conducted in 2007 some of the notes and descriptions of the data were lost. Therefore, the model could be compared on several key wave characteristics and model predictions, which include:

- Surface Elevation: the surface elevation predicted by the model is compared to the measured surface elevation next to the monopile.
- Wave Run-Up: the simulations predicted wave run-up height is compared to the experimental results, measured at the angles mentioned in section 3.2.5. However, due to measurement errors in the data, the surface elevation at $112.5°$ is left out.
- Wave characteristics: Key wave parameters are compared.

Table 3.9: Simulation parameters used for the validation simulation.

T	H	flow	Timestep	Co_{max}	ΔT_{max}	#Cells	End Time	Total Runtime
1.6	0.12	laminar	adjustable	0.6	0.05	9,490,848	50	$\approx$ 54 hours

The wave height in Figure 3.14, which represents the peak-to-trough variation, closely follows the experimental data, confirming that the model accurately reproduces the wave energy and propagation characteristics. Additionally, the phase shift between the two curves is minimal, indicating that the numerical scheme correctly resolves wave speed and propagation over time.

The deviation between the experiment and the simulation can be partly attributed to the mesh resolution. Accurate representation of wave steepness, especially near breaking points or steep fronts, requires a sufficiently refined mesh. Coarser meshes may introduce numerical dissipation, leading to slight reductions in wave amplitude. As demonstrated in Appendix A.1.1, mesh refinement is crucial in capturing the physical behavior of wave propagation and interaction.

Additionally, experimental uncertainties contribute to the deviation. Factors such as minor inaccuracies in wave gauge positioning, variations in wave maker performance or reflection of the boundaries may affect the surface elevation measured. Experimental uncertainties usually introduce minor variations in phase and amplitude due to mechanical limitations. These irregularities cause slight inconsistencies, especially over multiple wave cycles. In contrast, numerical simulations are based on idealized and precise boundary conditions, resulting in a smoother and more uniform wave train compared to experiments.

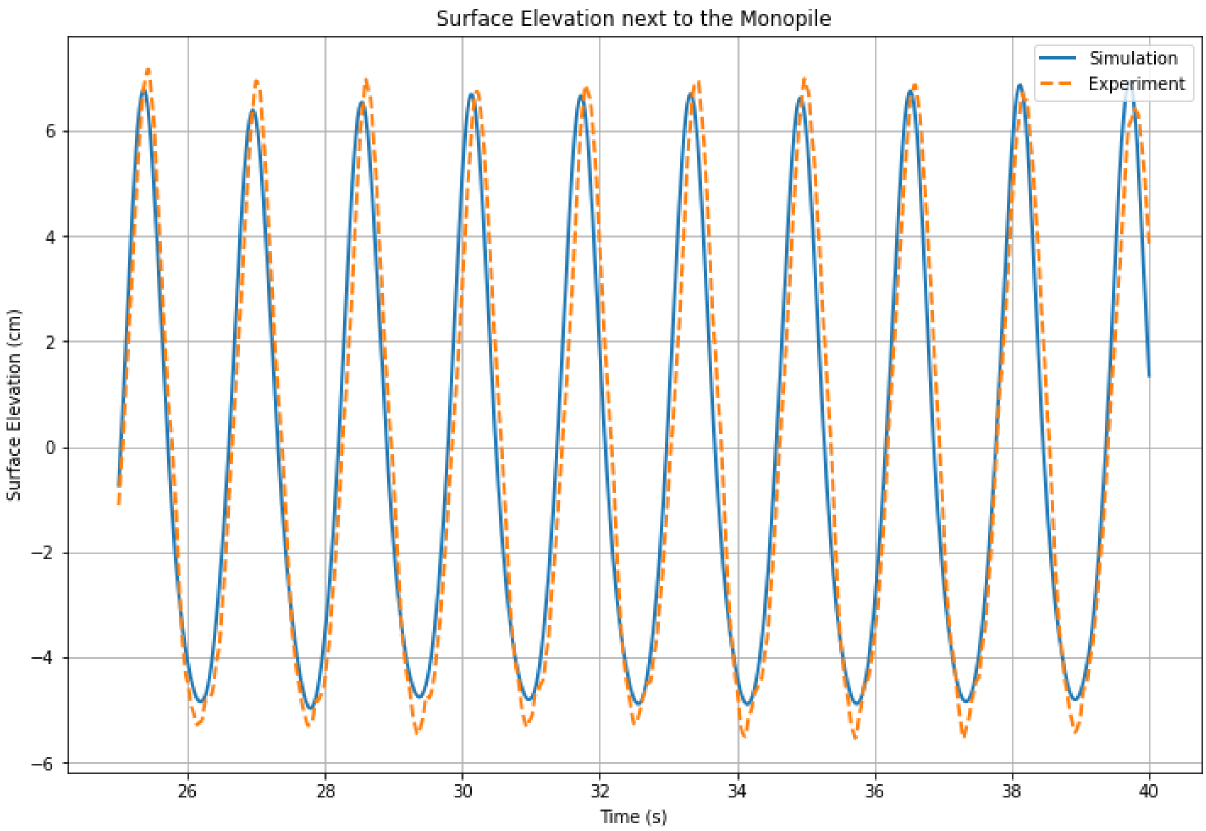


Figure 3.14: Surface Elevation next to the monopile.

Table 3.10 shows the absolute and relative errors compared to the experiment between the peaks (P) and troughs (T) for 10 fully generated waves (W). The accurate reproduction of the free surface elevation next to the monopile (Fig. 3.14) confirms that the numerical model is a reliable tool for simulating waves, making it suitable for further hydrodynamic analysis, including wave loads and run-up predictions on the monopile.

Table 3.10: Peak-to-peak and Trough-to-trough Differences for Simulation and Experiment.

Parameter	W 1	W 2	W 3	W 4	W 5	W 6	W 7	W 8	W 9	W 10
P-to-P Diff. [cm]	0.396	0.557	0.434	0.060	0.207	0.231	0.361	0.115	0.162	0.510
T-to-T Diff. [cm]	0.436	0.340	0.629	0.506	0.367	0.534	0.639	0.606	0.608	0.607
Rel. P Diff. [%]	5.518	8.009	6.214	8.895	3.009	3.343	6.213	6.039	7.945	7.944
Rel. T Diff. [%]	8.297	6.440	11.700	9.552	7.032	9.853	9.551	11.620	11.131	11.261

To further evaluate the overall accuracy of the simulation, the mean absolute error (MAE) and mean absolute percentage error (MAPE) are calculated using the following formulas and the results can be seen in Table 3.11.

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{3.11}$$

$$\text{MAPE} = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{\max(y_i)}\right| \tag{3.12}$$

where:

- y_i = actual values (experiment data)
- $\hat{y}_i$ = predicted values (simulation data)
- N = total number of waves

Table 3.11: Mean Absolute Error and Mean Absolute Percentage Error for Peaks and Troughs.

Parameter	MAE [m]	MAPE [$\%$]
Peak	0.3033	6.3129
Trough	0.5272	9.6437

Despite the minor deviations in surface elevation and taking into account the mesh resolution and experimental uncertainties, the results validate the numerical model and demonstrate that it captures the overall surface elevation with a high level of accuracy.

To prove that the model is suitable for wave run-up predictions, the wave run-up on the monopile from the experiment is compared to the wave run-up predictions from the model. Figures 3.15 and 3.16 show the wave run-up at $0°$ and $180°$. While the overall trend and wave patterns align well, the simulation slightly underpredicts the wave run-up in the trough as was the case for the surface Elevation. The underlying reasons are consistent with those explained for the surface elevation. The wave run-up for all angles besides $0°$ and $180°$ can be found in Figure B.1 in Appendix B.

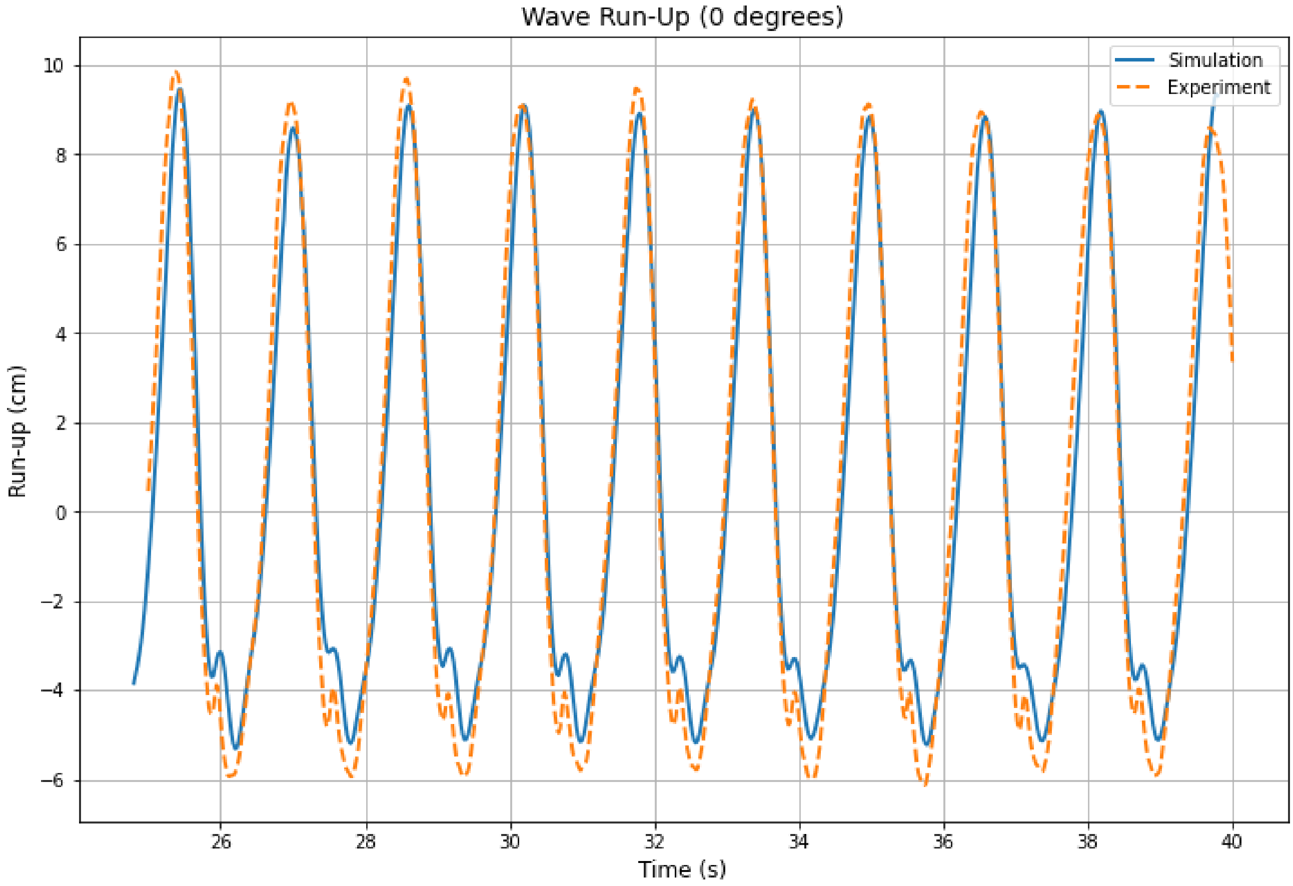


Figure 3.15: Wave run-up at $0°$ on the monopile.

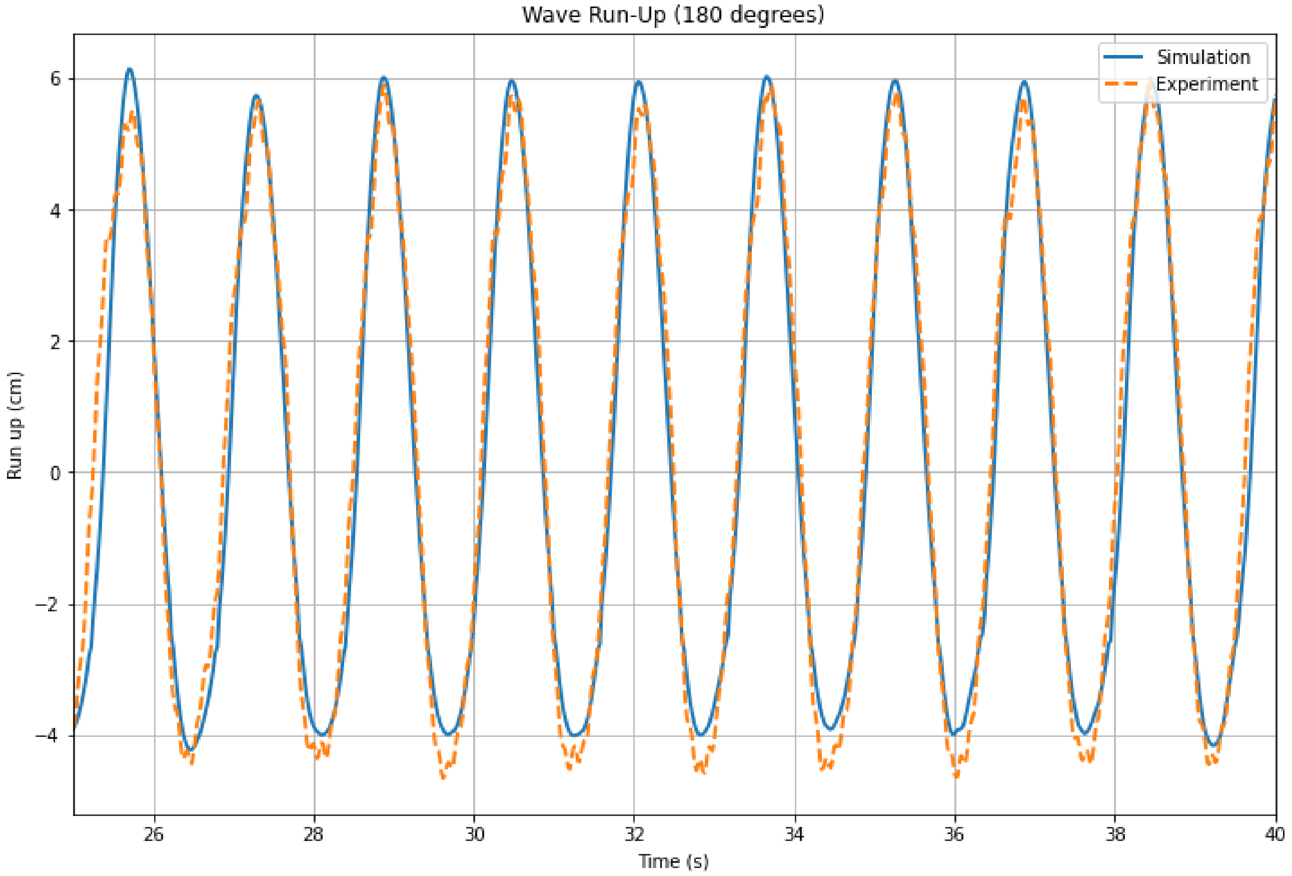


Figure 3.16: Wave run-up at $180°$ on the monopile.

For the wave run-up on the monopile, the MAE and MAPE values are presented in Table 3.12. The results indicate that the errors remain relatively low, confirming that the simulation closely follows the experimental measurements. The MAPE values further support this, with trough errors being slightly higher than peak errors but generally staying below 13%. The MAPE for the peaks is particularly low (<5%). When excluding the first or first three peaks to account for potential transient effects during wave generation, the values decrease even further (<3.5%).

Table 3.12: Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) for Peaks and Troughs.

Parameter	$0°$	$180°$
MAE Peaks $[m]$	0.3234	0.2774
MAPE Peaks [%]	3.4384	4.9279
MAE Troughs $[m]$	0.7518	0.4097
MAPE Troughs [%]	12.6635	9.1534
MAE Peaks (Excluding First) $[m]$	0.2152	-
MAPE Peaks (Excluding First) [%]	2.3228	-
MAE Peaks (Excluding First 3) $[m]$	-	0.2003
MAPE Peaks (Excluding First 3) [%]	-	3.5062

Finally, some key wave parameters between the experiment and simulation are compared and presented in Table 3.13. All parameters show small deviations between the experiment and the simulation. Only the the average height of the highest one-third of waves ($H_{1/3}$) and maximum wave height (H_{max}) show deviations around 10%. Again, these deviations arise from the same underlying causes as those affecting the surface elevation and run-up results (e.g. mesh resolution, experimental uncertainties, idealized boundary conditions).

Table 3.13: Comparison of wave characteristics between experiment and simulation.

Parameter	Experiment	Simulation
T_p [s]	1.707	1.669
H_{m0} [m]	0.1659	0.165
$H_{1/3}$ [m]	0.1285	0.117
T_s [s]	1.604	1.600
H_{max} [m]	0.1322	0.119

The difference in spectral significant wave height (H_{m0}) and ($H_{1/3}$) is because of their differing methods of computation, even though both are used extensively for the definition of the significant wave height (H_s), where the significant wave height represents the average wave height of the highest one-third of waves in a sea-state. It can be estimated in the time domain as $H_{1/3}$, the mean of the highest one-third of measured wave heights, or in the frequency domain as the spectral significant wave height $H_{m0} = 4\sqrt{m_0}$. Here m_0 is the zeroth moment, defined by:

$$m_0 = \int_0^\infty S(f)\,df \tag{3.13}$$

Where $S(f)$ is the wave energy spectrum as a function of frequency (Langton, 2000). In irregular waves, energy is distributed over a range of frequencies and m_0 must be determined by integrating over the entire spectrum. But for a monochromatic and regular wave, where the surface elevation assumes a pure sine form, all of the wave energy is concentrated in one frequency f_0. In this case, the energy spectrum $S(f)$ collapses to a delta function at f_0, and the total energy simplifies to $m_0 = A^2/2$, where A is the wave amplitude. Substituting this into the formula for H_{m0} gives (Langton, 2000):

$$H_{m0} = 4\sqrt{m_0} = 2\sqrt{2}A \approx 1.414H \tag{3.14}$$

This means that H_{m0} will be greater than both the actual wave height H and H_s, which in regular wave conditions tends to H. So, while both H_s and H_{m0} are valid definitions of significant wave height, they will have different values in regular waves due to the energy-based definition of H_{m0} versus the statistical definition of H_s.

Overall, the comparison between the numerical and experimental results demonstrates that the surface elevation, wave run-up and wave characteristics are well captured by the simulation, with relatively low errors in both peak and trough measurements. The wave period and significant wave height exhibit only minor deviations, confirming that the model provides a reliable estimation of wave dynamics. However, some discrepancies remain due to limitations in mesh resolution and numerical diffusion, which can introduce small variations in wave steepness and run-up heights. While further refinement will improve accuracy, the current setup offers a balance between computational cost and precision, making it a practical approach for simulating wave-structure interactions.

4 Results

4.1. Different scales

The effect of scale on wave-structure interactions is crucial in validating numerical models and understanding run-up and hydrodynamic forces on offshore structures. This chapter analyzes the results obtained from simulations at different scales, including a small-scale (D = 0.12m), an intermediate-scale (D = 6m), and a large-scale model (D = 10m). All models are scaled using Froude scaling (section 2.4), with corresponding Froude scaling factors of 50 and 83.33. Their run parameters are given in Table 4.1. The influence of scaling is assessed by comparing the non-dimensionalised surface elevation, wave run-up, and hydrodynamic forces on the monopile across the three model scales. This approach enables a direct evaluation of the consistency between the models and provides insight into the validity of the applied scaling laws.

Table 4.1: Simulation parameters used for the different scale simulations. All other parameters and settings are explained in section 3.2.

Run	T	H	Flow	Timestep	Co_{max}	ΔT_{max}	#Cells	End Time	Total Runtime
D0.12	1.6	0.12	laminar	adjustable	0.6	0.05	9,490,848	50	$\approx$ 58 hours
D6	11.3	6	laminar	adjustable	0.6	0.05	9,486,529	295	$\approx$ 80 hours
D10	14.6	10	laminar	adjustable	0.6	0.05	9,491,352	380	$\approx$ 78 hours

4.1.1. Surface Elevation

When comparing the surface elevation at different scales, it is important to consider hydrodynamic similarity and scaling effects. Under Froude scaling, wave parameters such as period (T) and height (H) are scaled accordingly to maintain dynamic similarity across different scales (section 2.4). Ideally, when surface elevation is normalized by H and time by T, the results should collapse onto a single trend if scale effects are negligible.

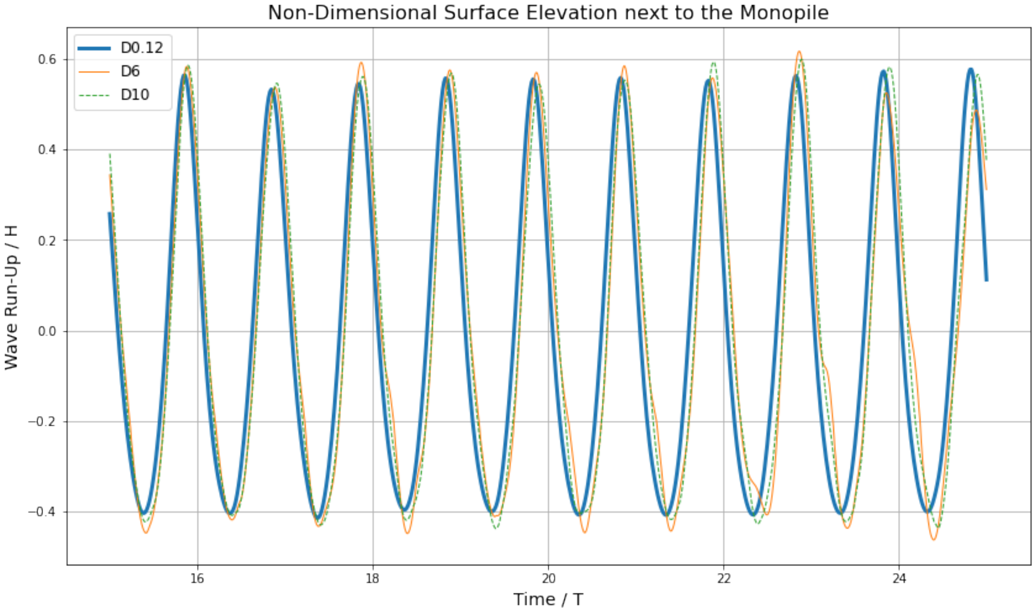


Figure 4.1: Non-dimensional surface elevation next to the monopile for different scales.

Figure 4.1 shows the normalized surface elevation measured next to the monopile for the three different scales. The wave profiles across these cases demonstrate strong consistency, with minor variations in the crests and troughs, indicating small numerical errors and minimal scale effects on surface elevation dynamics. This suggests that the numerical model successfully preserves hydrodynamic similarity across different scales by following Froude scaling principles.

A more detailed quantitative comparison is provided in Table 4.2, where several non-dimensionalised wave parameters are shown for each scale. The dominant period (T_p) and mean zero-crossing period (T_s) show negligible variations, proving the consistency of wave propagation dynamics and confirming that the frequency characteristics of the wave field remain unchanged across different scales. The different wave heights are also very consistent with some minor deviations, except for the maximum wave height for D6. The deviation in maximum wave height, caused by wave eight, is most likely the result of a minor numerical error. Given that all models are Froude scaled and the rest of the wave train agrees well, the deviation is unlikely to originate from scale effects.

Table 4.2: Non-dimensionalised (divided by H and T) wave parameters for D0.12, D6, and D10 right after the slope.

Parameter	D0.12	D6	D10
Non-dimensional Dominant Period (Tp)	1.042	0.994	0.996
Non-dimensional Significant Wave Height (Hm0 - Spectral)	1.386	1.397	1.414
Non-dimensional Significant Wave Height (Hs - Highest 1/3)	0.957	1.041	1.000
Non-dimensional Mean Zero-Crossing Period (Ts)	0.999	0.996	0.997
Non-dimensional Maximum Wave Height (Hmax)	0.971	1.120	1.045

4.1.2. Wave Run-Up

Figures 4.2 and 4.3 show the non-dimensionalised wave run-up at 0° and 180° for the different scales. The wave run-up at the remaining angles is presented in Figure B.2 in Appendix B. The results provide valuable insight into the influence of scale effects on wave run-up behavior. The pattern of wave run-up is well captured, demonstrating that Froude scaling effectively captures the overall wave-structure interaction. However, small deviations in peak run-up values and small phase shifts are observed between different scales. These differences are attributed to the influence of viscous forces, which are not accounted for in Froude scaling. At the smallest scale (D0.12), the lower Reynolds number leads to increased viscous dissipation of wave energy, resulting in smaller peak run-up heights. In contrast, at larger scales (D6 and D10), inertial forces dominate, resulting in more wave energy and higher run-up values. Overall, the strong agreement across scales confirms the effectiveness of Froude scaling. However, including Reynolds number effects could further improve the accuracy of small scale modeling.

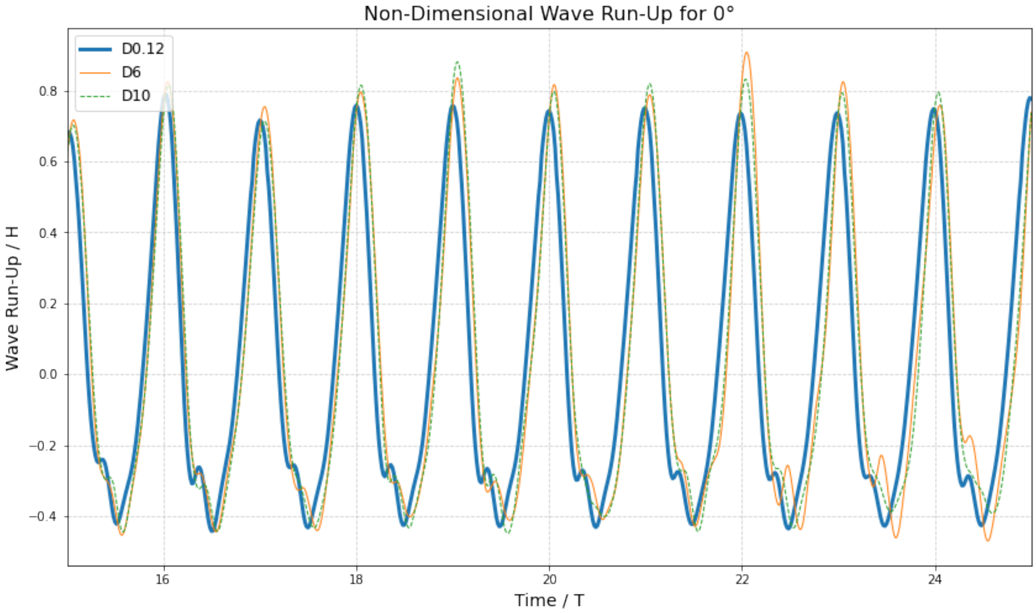


Figure 4.2: Non-dimensional run-up for 0° for different scales.

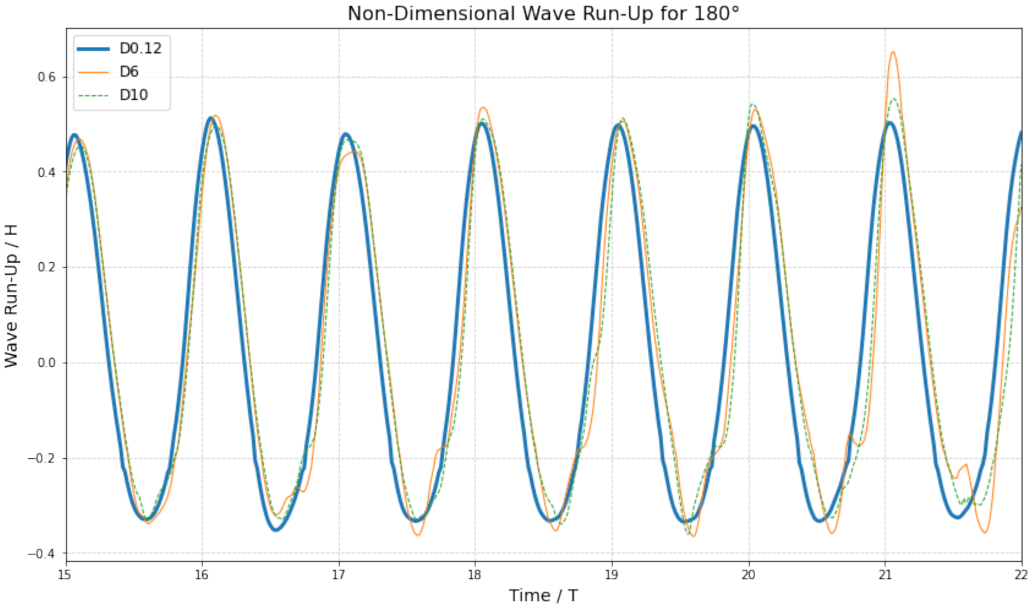


Figure 4.3: Non-dimensional run-up for 180° for different scales.

4.1.3. Forces

Figure 4.4 shows the non-dimensionalised total force in x-direction, based on Froude similarity (Eq. 4.1). A clear and consistent collapse of the force signal is observed for all three scales after the transient phase. This confirms that the total X force is primarily governed by inertial and gravitational effects, which are correctly preserved under Froude scaling. The excellent agreement between the different scales supports the assumption that Froude similarity is valid for the prediction of total wave loading in the x-direction. This also confirms that viscous effects in the x-direction are negligible compared to the pressure-induced inertial loads.

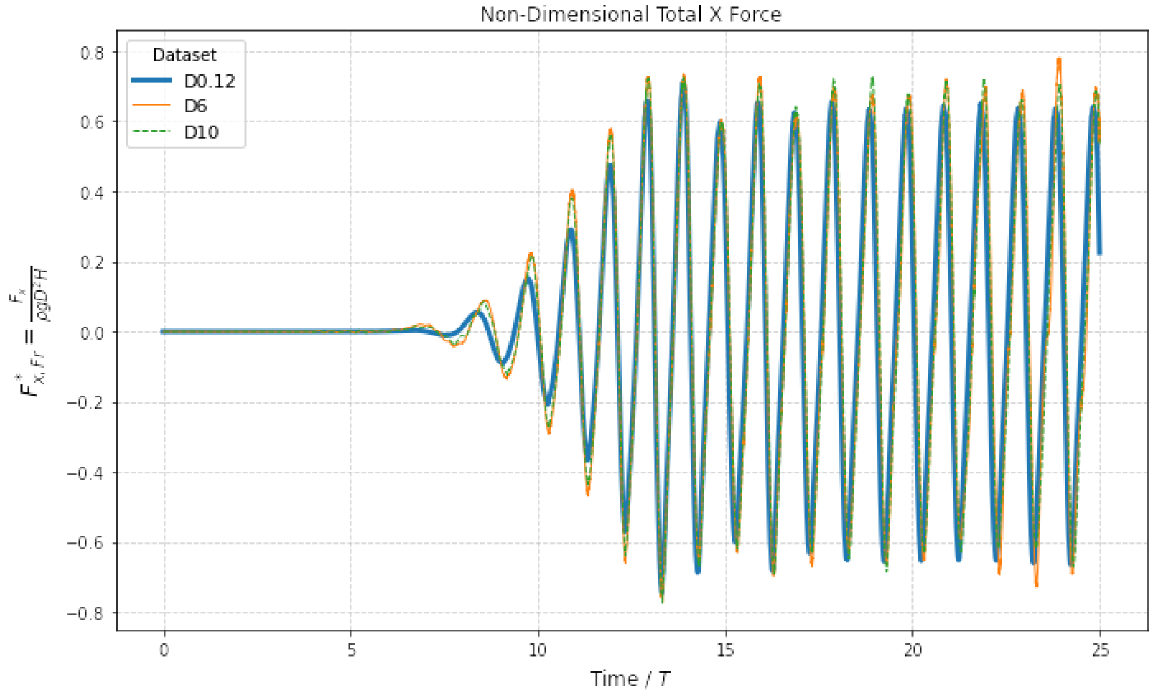


Figure 4.4: Non-Dimensional total force in x-direction (Eq. 4.1) vs non-dimensional time for different scales.

$$F^*_{,Fr} = \frac{F}{\rho g D^2 H} \tag{4.1}$$

where:

- F is the force [N]
- $F^*_{,Fr}$ is the non-dimensionalised force by the characteristic inertial force scale

Figure 4.5 shows the non-dimensional transverse force (F_y), which is much smaller in magnitude and more sensitive to local flow fluctuations, such as vortex shedding or asymmetries in wave impact. While all scales are centered around zero, the larger-scale models exhibit greater scatter and divergence. This is due to increased turbulence and unsteady wake behavior at higher Reynolds numbers, which are not accounted for under Froude scaling. Additionally, numerical resolution may influence the observed fluctuations, particularly because small-magnitude forces like F_y are more prone to discretization errors. Since transverse forces in regular waves are generally small, even minor physical or numerical imbalances can lead to differences. These effects are more significant at larger scales due to stronger flow gradients and more complex wake dynamics.

Therefore, while part of the variation is caused by real physical effects such as turbulence, some discrepancies also arise from numerical limitations. This highlights a key limitation of Froude scaling, it preserves gravitational similarity, but does not account for Reynolds number effects, which can be important for accurately capturing transverse forces.

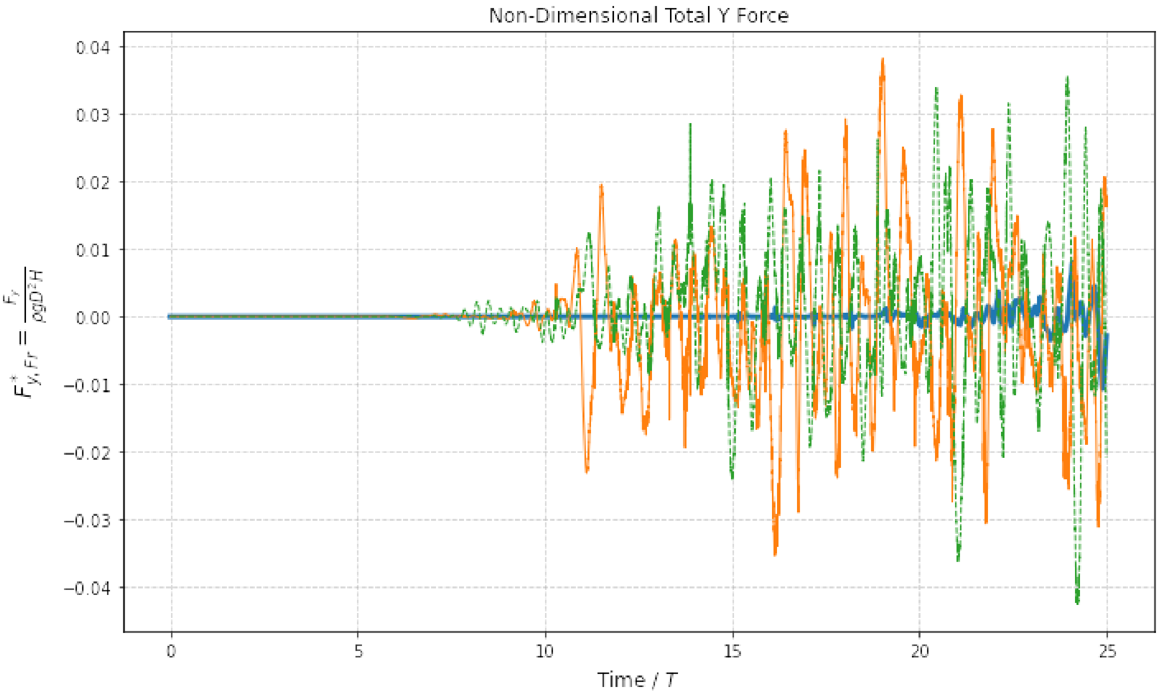


Figure 4.5: Non-Dimensional total force in y-direction (Eq. 4.2) vs non-dimensional time for different scales.

However, since F_y has minimal influence on the overall wave run-up or the total force distribution on the monopile, these differences are not critical. Therefore, the scale effects observed in the transverse direction do not affect the outcome and reliability of the simulation.

Figure 4.6 presents the non-dimensional total and pressure forces in the vertical (z) direction for three geometrically scaled models (D0.12, D6, and D10). These forces have been non-dimensionalised based on Froude similarity. As mentioned earlier, Froude scaling ensures that gravitational and free-surface dynamics are preserved across scales, which is critical in modeling wave-structure interaction. However, as the plots show, the total and pressure forces in z-direction do not collapse onto a single curve when normalized using this approach. This deviation indicates the influence of Reynolds number effects.

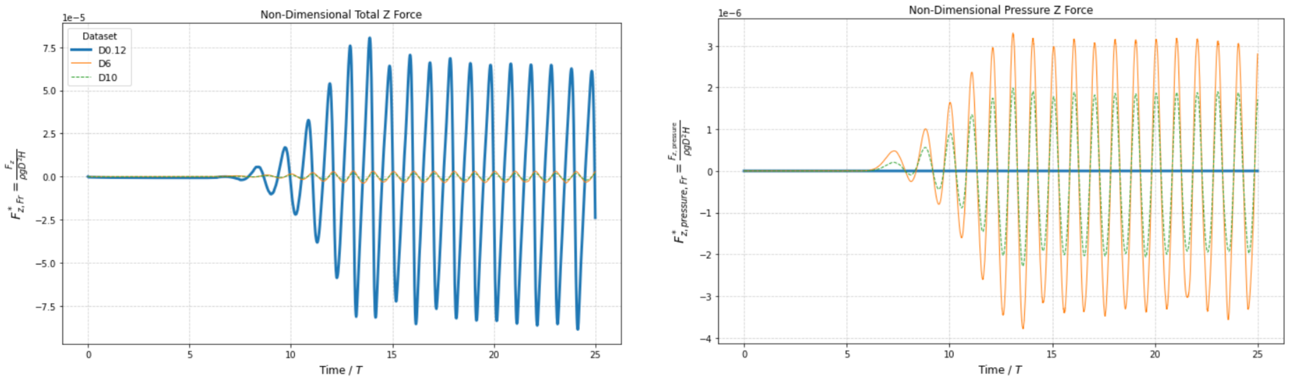


Figure 4.6: Non-dimensionalised total and pressure force in z-direction (Eq. 4.1) vs non-dimensional time for different scales.

In contrast, Figure 4.7 shows the non-dimensionalised viscous force in z direction ($F^*_{z,viscous,Re}$) based on Reynolds similarity (Eq. 4.2). This proves that viscous forces depend on shear, which scale with velocity and length. Under this Reynolds-based normalization, the viscous force curves collapse much more closely across the three scales, confirming that viscous forces indeed follow Reynolds similarity.

$$F^*_{,\text{Re}} = \frac{F}{\mu U D} \tag{4.2}$$

where:

- $F^*_{,Re}$ is the non-dimensionalised force by the characteristic viscous force scale

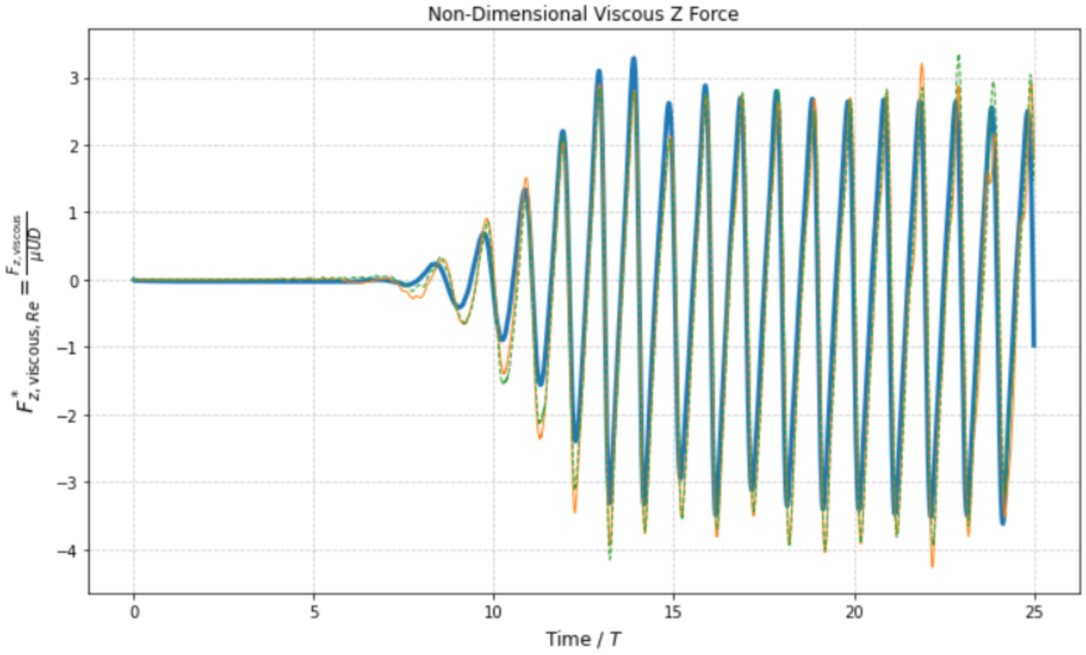


Figure 4.7: Non-dimensional viscous z force (Eq. 4.2) vs non-dimensional time for different scales.

This behavior can be better understood by considering how the contribution of viscous and inertial forces varies with Reynolds number across the different model scales. In the small-scale model, the Reynolds number is lower, indicating that viscous forces play a relatively larger role in the total force response. As a result, when total forces are non-dimensionalised using Froude scaling, which does not account for viscous effects, the viscous contributions in the small-scale model become more prominent. This explains why the smaller-scale model (D0.12) exhibits a noticeably larger relative force response in the Froude-scaled plot (Fig. 4.6).

In contrast, when the total force is non-dimensionalised using Reynolds scaling (Fig. 4.8), the larger-scale models (D6 and D10) dominate. This is because their higher Reynolds numbers correspond to a flow regime where inertial forces dominate viscous effects. Consequently, in Reynolds-scaled plots, the influence of the small-scale model is reduced, while the larger-scale models collapse more closely, reflecting the dominance of inertia and the diminishing role of viscosity.

It should also be noted that an initial pressure force in the vertical (z) direction was observed, which affected the comparability between the total and pressure force components. This initial force is attributed to non-zero surface normals at the boundaries between the refinement region and the surrounding mesh. Even when this initial force is subtracted, the resulting pressure force magnitudes remain inaccurate due to the non-zero surface normals. The initial force can be seen in Appendix B.1. As a result, only the viscous force component is considered in the following chapters. It is also important to note that these differences are particularly evident in the vertical force component (F_z), where viscous effects are greater due to vertical flow separation and near-bed shear. In contrast, the horizontal force component (F_x) is primarily governed by wave-induced pressure forces and inertia, which scale well under Froude similarity. This is reflected in the good agreement between the models when non-dimensionalised using Froude scaling (Fig. 4.4).

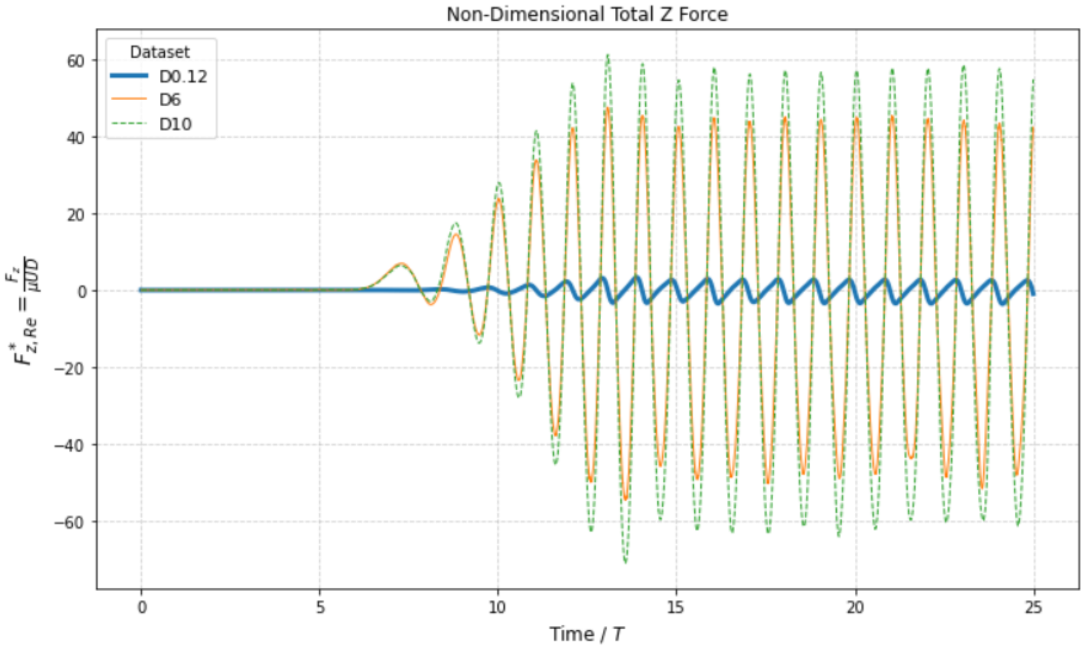


Figure 4.8: Non-dimensional total force in z-direction (Eq. 4.2) vs non-dimensional time for different scales.

These findings highlight a key limitation in using Froude scaling for all force components. While it accurately captures pressure-driven effects and wave behavior, it does not properly scale viscous phenomena. For most engineering purposes, this is acceptable because the viscous forces are very small and their influence on wave run-up and global force response is negligible. However, for detailed analysis of local flow features, such as separation, or downflow behind the monopile, these limitations must be considered.

4.2. Different Monopile Diameters

Since wave loading and water elevation around a monopile depend on both hydrodynamic conditions and the geometry of the monopile, it is important to understand how variations in diameter influence run-up behavior. Here, the effect of the monopile diameter is isolated by using the D10 model and keeping all wave, mesh and numerical parameters constant. Only the monopile diameter varies, other parameters are given in Table 4.3. This will give insights into the influence of geometry on wave-structure interaction. By keeping everything the same, scale effects are eliminated.

Table 4.3: Simulation parameters used for the different monopile diameter simulations. All other parameters and settings are explained in section 3.2.

Run	T	H	Flow	Timestep	Co_{max}	ΔT_{max}	#Cells	End Time	Total Runtime
D10	14.6	10	laminar	adjustable	0.6	0.05	9,490,848	380	≈ 78 hours
D14	14.6	10	laminar	adjustable	0.6	0.05	9,491,352	380	≈ 84 hours

4.2.1. Wave Run-Up

Figure 4.9 and Appendix B.3 show the wave run-up on monopiles with two different diameters under identical hydrodynamic conditions for different angles around the monopile. This allows for the assessment of the diameter on wave-structure interaction. At 0°, the wave run-up is nearly identical for D10 and D14, which is logical since this is the direct wave impact zone and the hydrodynamic conditions are identical.

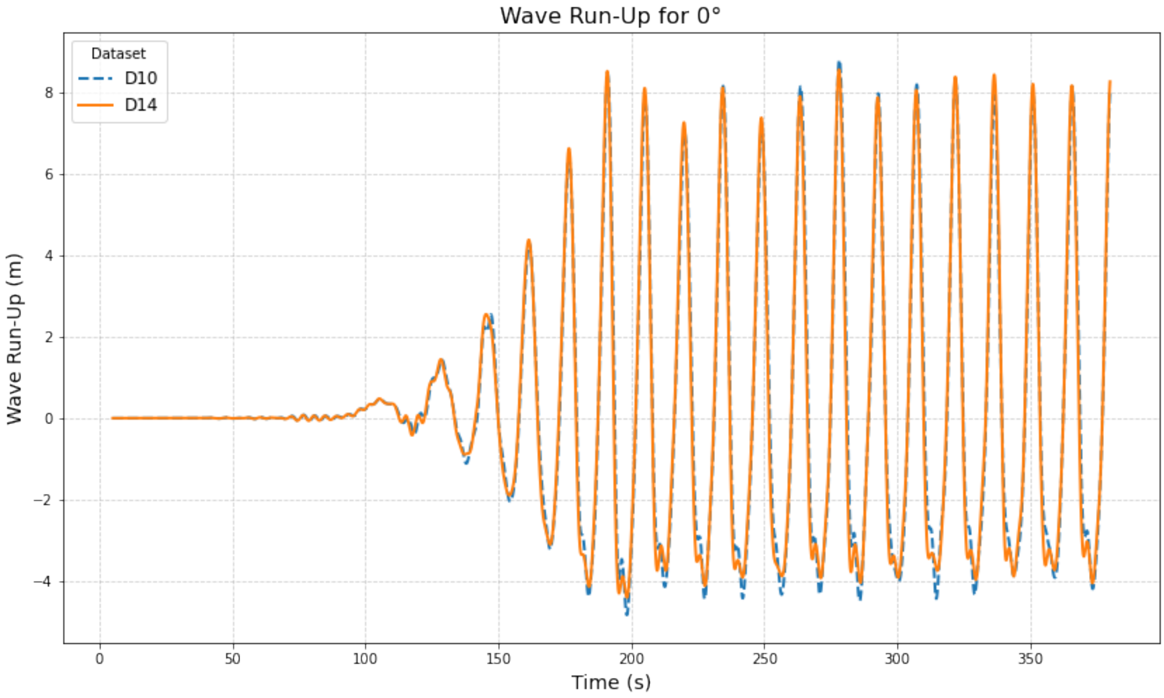


Figure 4.9: Wave run-up for different diameters under the same hydrodynamic conditions at 0°.

Figure 4.10 shows the maximum normalized run-up at different angles around the monopile, which does show differences.

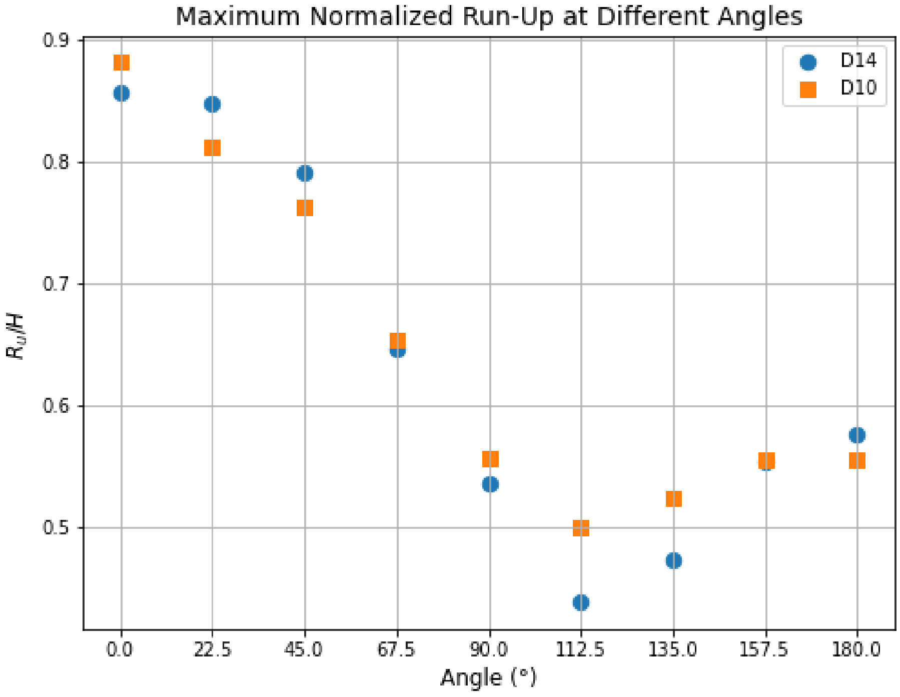


Figure 4.10: Maximum normalized run-up at different angles for different diameters.

The differences in normalized run-up between the D10 and D14 monopiles around the circumference of the structure are explained by how each diameter interacts with the incoming wave. At the front of the monopile (0°), the run-up values for D10 and D14 are nearly identical, indicating similar direct wave impact. However, at 22.5° and 45°, D14 shows higher run-up than D10. This is due to the larger pile capturing and redirecting more wave energy around its sides, causing local amplification of the run-up in those off-front regions.

When moving further along the sides (67.5° to 135°), the run-up on D14 becomes lower. This is due to the larger diameter causes stronger diffraction and more pronounced shadow zones at those angles. The smaller D10 structure allows more wave wrapping, so wave energy can more easily reach the side and rear surfaces. However, toward the back side (180°), the D14 monopile still shows slightly higher run-up. This suggests that although the larger diameter disperses and blocks wave energy more effectively, some recirculation behind the pile may still create a slightly elevated run-up in that region compared to D10.

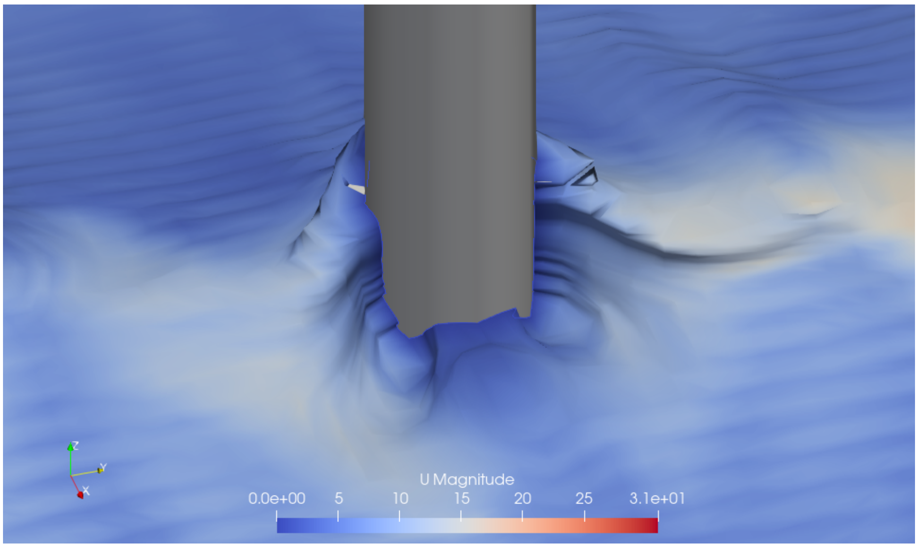


Figure 4.11: Visualization of a wave hitting the D14 monopile and the shadow zones behind the monopile.

The results clearly show that monopile diameter influences the spatial distribution of wave run-up. Differences between the D10 and D14 cases highlight the role of geometry in shaping how wave energy is reflected, diffracted, and redistributed around the structure. Larger diameters lead to stronger wave obstruction and more pronounced shadow zones, while smaller diameters allow more wave wrapping and energy to reach the rear and sides of the pile. However, this comparison is limited to two diameters under identical hydrodynamic conditions. The findings may not hold for larger diameter variations or different wave parameters. Additionally, the use of laminar flow without turbulence modeling simplifies the flow behavior, showing the geometric effects but limiting applicability to real-world conditions. To fully understand the influence of monopile diameter on wave run-up, a broader parametric study involving multiple diameters and varying wave characteristics is required.

4.2.2. Forces

The total x and total y force are again non-dimensionalised using Froude-based scaling, where viscous z is non-dimensionalised using Reynolds scaling. Figure 4.12 demonstrates that when these forces are non-dimensionalised, the resulting force profiles align very well across all components. The fluctuations in F_y are again due to small numerical errors. Since all other hydrodynamic conditions are held constant and only the diameter of the monopile varies, this setup isolates the effect of geometry on force generation. However, the force must be non-dimensionalised before comparison, because force is equal to pressure times area. Without normalization, differences in absolute force values would reflect geometric scaling rather than underlying changes in physical behavior. The force behavior over time is the same for both diameters, confirming that changes in force are primarily due to geometry rather than changes in physical response and that the wave-induced force behavior remains consistent across different diameters.

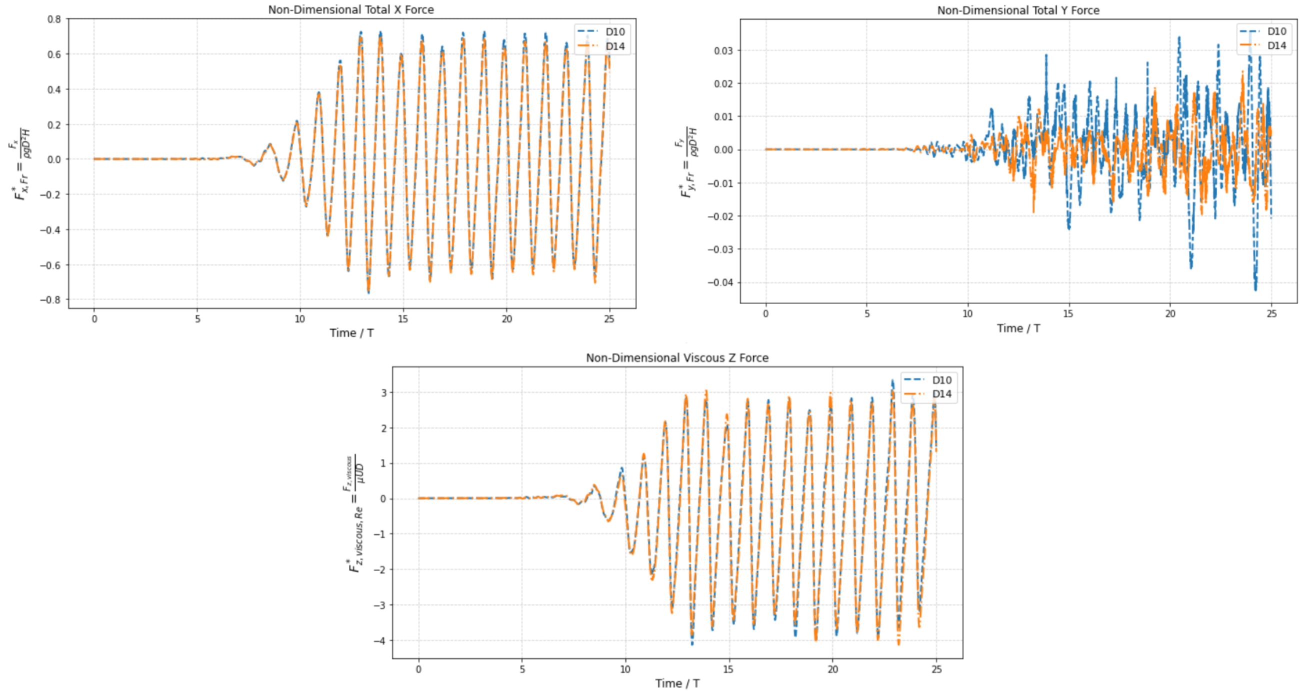


Figure 4.12: Non-dimensionalised Total X, Total Y and Viscous Z force (Eq. 4.1) vs non-dimensional time for two different diameters under the same hydrodynamic conditions.

4.3. Different Wave Characteristics

This section analyzes the results of five different regular waves simulated using the D10 model to investigate the influence of wave characteristics on wave run-up around a monopile. The wave height H and period T are Froude-scaled to ensure dynamic similarity across cases, while the wave celerity c is computed using the dispersion relation. Since the wave number k must be solved iteratively from the dispersion equation (Appendix D), this introduces some slight variations in wave speed between the cases. The selected wave conditions span a range from a little non-linear ($S_0 = 0.032$) to near-breaking conditions ($S_o = 0.064$). The run parameters are given in Table 4.4.

Table 4.4: Simulation parameters used for the different wave characteristics simulations. All other parameters and settings are explained in section 3.2.

Wave	T	H	Flow	Timestep	Co$_{max}$	ΔT$_{max}$	#Cells	End Time	Total Runtime
1	10.32	5.00	laminar	adjustable	0.6	0.05	10,475,440	300	≈ 68 hours
2	14.61	10.00	laminar	adjustable	0.6	0.05	9,491,352	380	≈ 80 hours
3	15.79	11.67	laminar	adjustable	0.6	0.05	10,475,440	237	≈ 42 hours
4	17.98	15.00	laminar	adjustable	0.6	0.05	11,974,196	270	≈ 68 hours
5	20.63	20.00	laminar	adjustable	0.6	0.05	18,575,196	310	≈ 216 hours

The increase in number of cells for wave 5 is due to Δz being 25% smaller. Meaning that there are more cells refined between $\pm H$ to better capture the breaking wave (section 3.2.1). The corresponding wave parameters right after the inletzone for each case are summarized in Table 4.5, including several key non-dimensional parameters: $\frac{H}{gT^2}$, $\frac{d}{gT^2}$, and $\frac{\pi D}{L}$. These non-dimensional quantities are useful for classifying wave regimes and determining the appropriate theoretical models for wave behavior. Figure 4.13 presents two diagrams used for this classification. This diagrams categorize the wave force regime and the wave theory. For Wave 1, which falls between 2^{nd} and 3^{rd} order, the 5^{th} order is chosen since 3^{rd} is not available in waves2Foam and 5^{th} order includes all terms from the 3^{rd} order theory. For Wave 2,3,4, the 5^{th} order theory was selected since it reduces the computational costs and the waves are not so non-linear that they require the complexity of Stream Function theory (Fenton, 1985; van den Bos & Sanderse, 2014). Each wave case is marked with its number in both diagrams.

Table 4.5: Wave parameters right after the inlet zone with corresponding theoretical classifications.

Wave	H	T	d	k	L	c	H/gT^2	d/gT^2	$\pi D/L$	So	Theory
1	5.00	10.32	41.67	0.041	155.10	15.04	0.0048	0.03992	0.203	0.032	Stokes 5^{th}
2	10.00	14.61	41.67	0.025	256.78	17.56	0.0048	0.01991	0.122	0.039	Stokes 5^{th}
3	11.67	15.79	41.67	0.022	283.04	17.94	0.0048	0.01703	0.111	0.041	Stokes 5^{th}
4	15.00	17.98	41.67	0.019	332.68	18.46	0.0047	0.01313	0.094	0.045	Stokes 5^{th}
5	20.00	20.63	25.64	0.020	313.94	15.22	0.0048	0.00614	0.100	0.064	Stream func.

Where:

- H = Wave Height [m]
- T = Wave Period [s]
- d = depth [m]
- L = Wave Length [m]
- D = Diameter = 10 [m]
- S_o = Steepness = H/L [-]

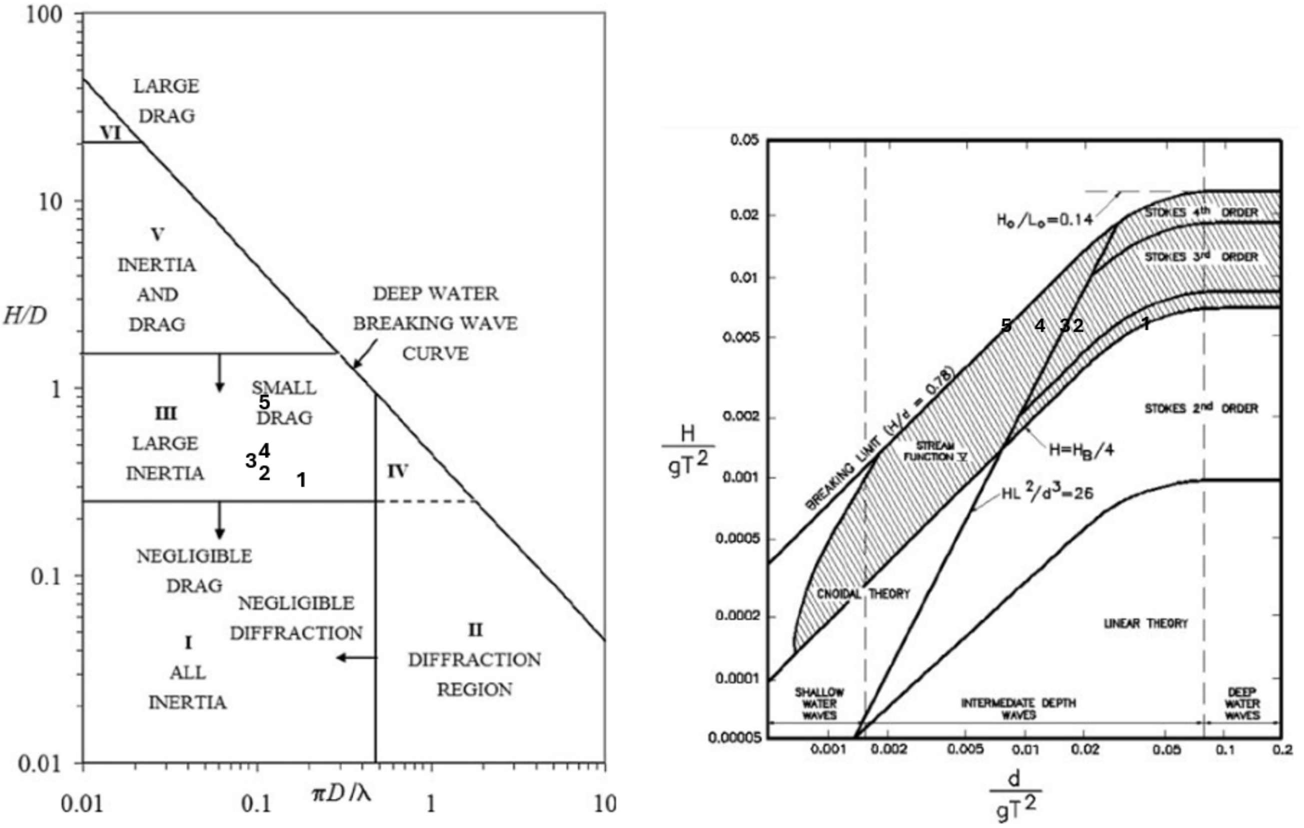


Figure 4.13: Wave force regimes and wave theory classifications for the different waves.

4.3.1. Surface Elevation

The surface elevation plots in Figure 4.14 compare the wave profiles at two locations: one wavelength (1L) after the inlet zone and directly next to the monopile. Across most scenarios, the wave signals remain consistent. However, Wave 5 exhibits a clear deviation from the others, especially next to the monopile.

This behavior can be attributed to the high relative wave height in Wave 5, where the ratio $H/d = 0.78$. At this level, the wave starts breaking, especially when traveling over a bottom slope. In the simulation, a slope of 1:100 is present between the inlet and the monopile, causing local shoaling. This promotes wave steepening and eventual breaking, which dissipates energy and results in the observed decrease in surface elevation amplitude

(Fig. 4.15). This shoaling is, next to the decreasing wave crest due to breaking, also the reason for the decrees in trough (i.e. less deep).

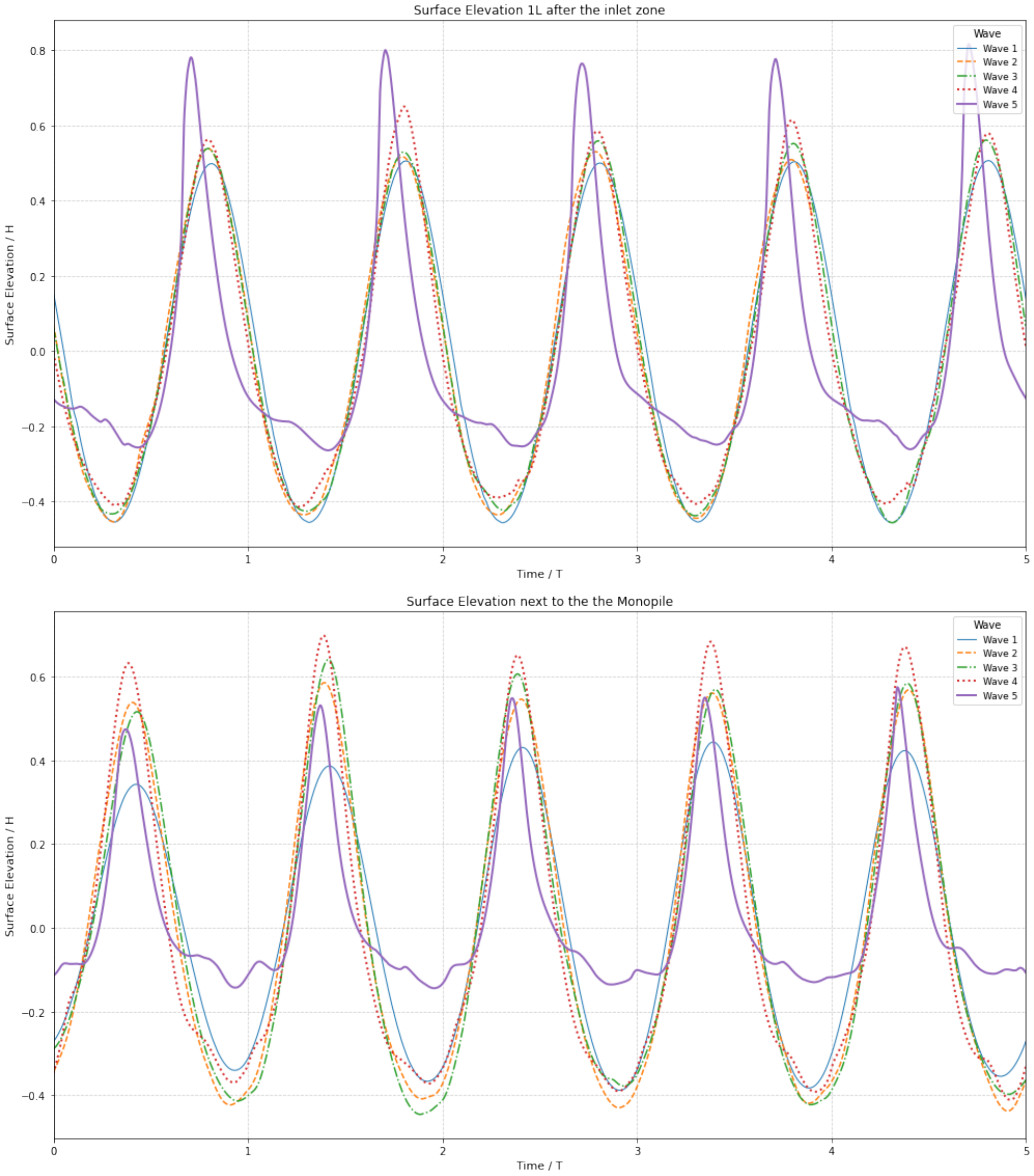


Figure 4.14: Surface elevation for different wave characteristics 1L after the inlet zone and next to the monopile.

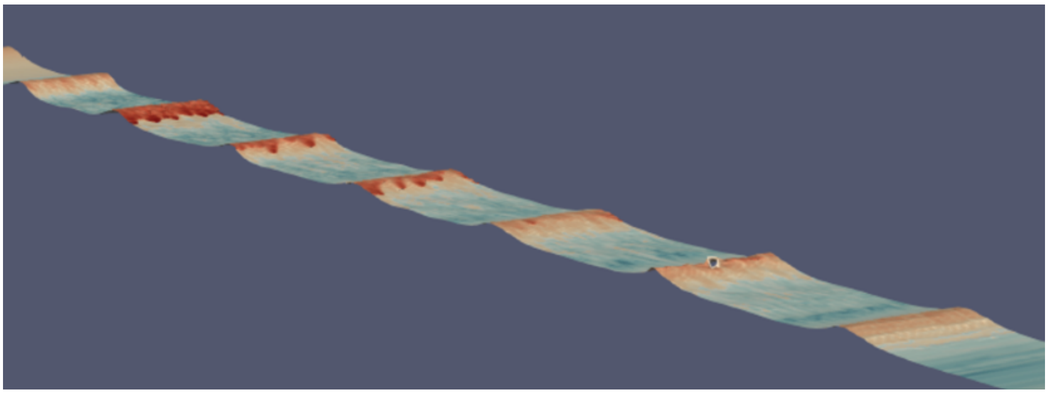


Figure 4.15: Wave propagation of Wave 4 approaching the monopile. Visualized is the horizontal velocity field **u**. The red regions indicate high velocities near the crest, which decrease as the wave breaks and energy dissipates.

Wave 1 (Fig. 4.14) shows a gradual reduction in wave height as the wave propagates through the domain, especially evident when comparing the surface elevation at 1L after the inlet zone and next to the monopile. This is because the vertical mesh resolution was not refined as outlined in section 3.2.1. This coarser resolution has also led to increased numerical diffusion, resulting in the slight decrease of the wave height.

In contrast, scenarios such as Wave 3, with a more moderate $H/d = 0.36$, show an increase in wave amplitude as the wave shoals over the slope. This is because when a wave travels into shallower regions without breaking, its height increases due to energy conservation over decreasing depth. This amplification is clearly visible in the plot, where Wave 4 shows higher crests near the monopile compared to the location after the inlet zone. These results show the sensitivity of waves to the H/d ratio. While Wave 5 illustrates how large relative wave heights can trigger early breaking and energy loss, more moderate conditions such as Wave 3 and 4 experience constructive shoaling, increasing wave-induced loading on structures. This proves that even small seabed slopes can significantly affect incoming wave energy depending on the wave conditions.

4.3.2. Wave Run-Up Analysis and Comparison to Existing Theories

To further analyze the hydrodynamic behavior of the wave scenarios, the wave run-up, surface elevation, wave height and steepness values were extracted directly after the slope transition ($S_{o,slope}$) and can be seen in Table 4.6. These parameters show the influence of the slope on the generated waves and the influence of wave characteristics on the resulting wave run-up around the monopile. It is important to re-evaluate the steepness at this point because the presence of the slope alters the wave shape.

Table 4.6: Wave run-up, surface elevation, wave height and steepness values for different scenarios right after the slope and in front of the monopile. RU = Run-Up, El = Elevation. H is generated wave height at inlet zone and $S_{o,slope}$ = Avg H / L (L from Table 4.5).

Wave	H	Max Wave RU. (0°)	Avg Max RU. (0°)	Max El.	Avg El.	Avg H	$S_{o,slope}$
1	5.00	3.040	2.797	2.259	2.179	3.916	0.025
2	10.00	8.808	8.111	5.815	5.398	10.052	0.039
3	11.87	10.729	9.510	7.310	6.898	11.610	0.041
4	15.00	16.401	14.285	10.416	10.004	15.813	0.048
5	20.00	18.588	17.383	11.254	10.716	13.513	0.043

Now that key parameters are determined at the location right in front of the monopile, the wave run-up on the monopile can be determined. In Appendix B.4 the wave run-up for five fully generated waves at all angles around the monopile can be seen and in Figure 4.16 the maximum normalized wave run-up for all angles for all waves are given.

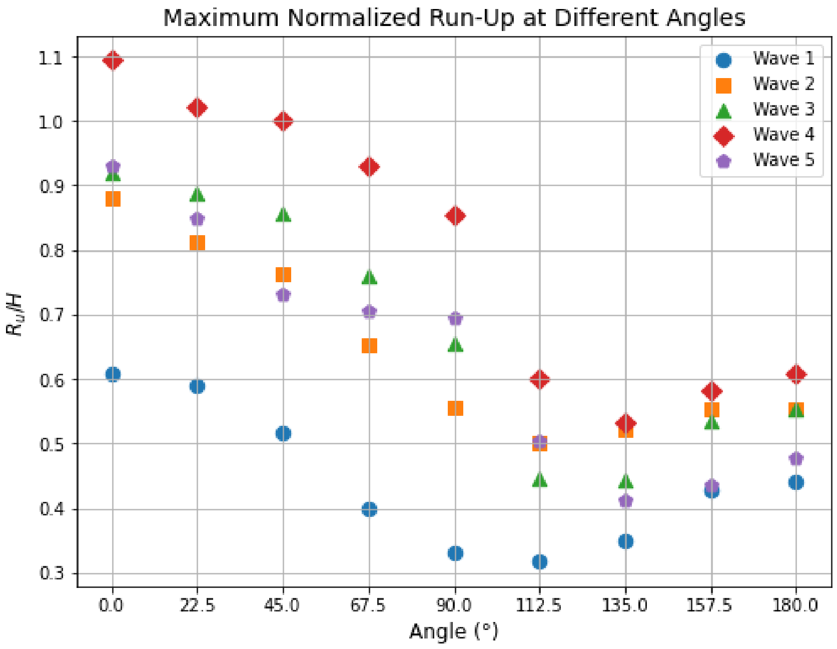
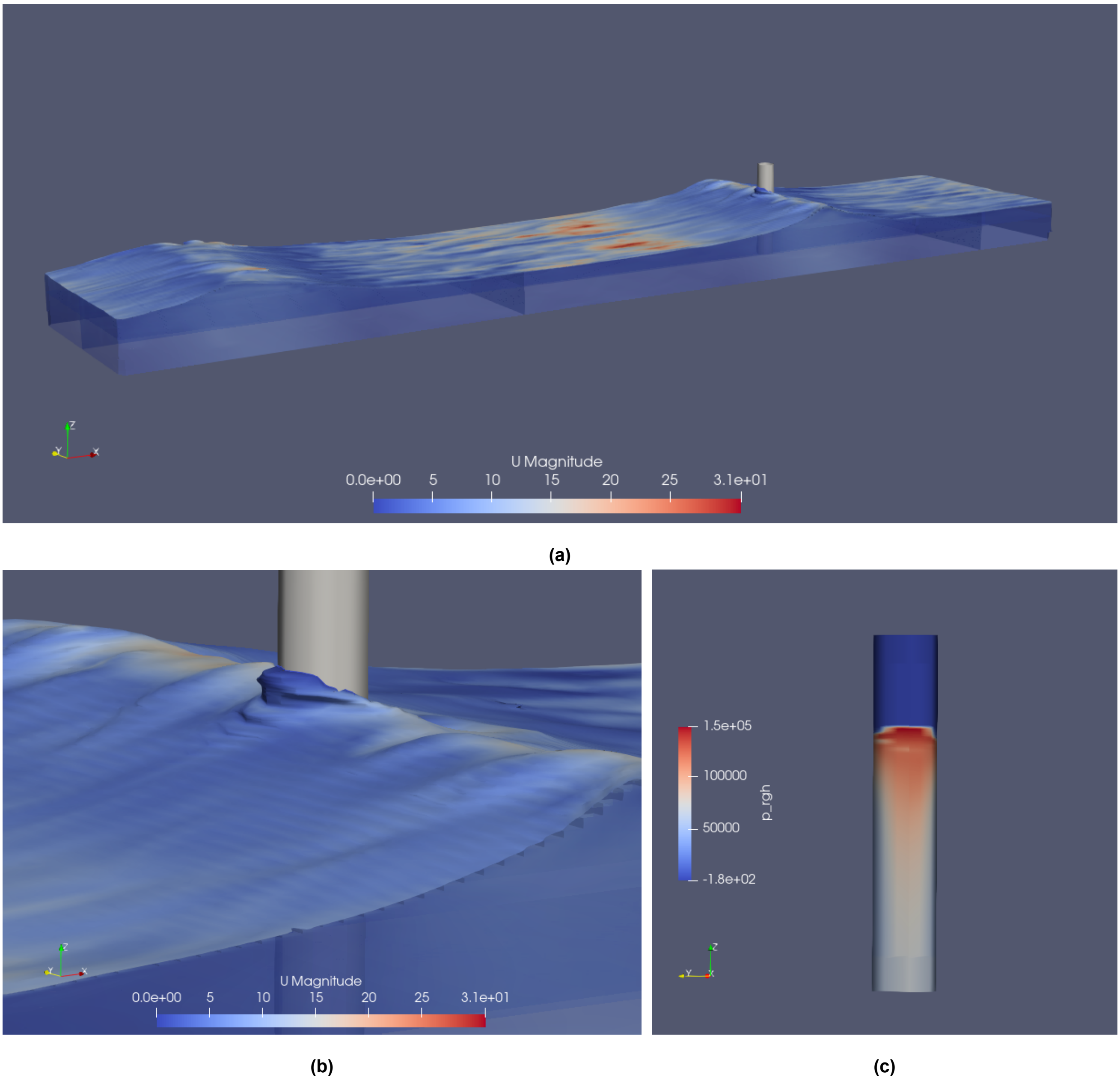


Figure 4.16: Maximum normalized run-up at different angles for different wave characteristics.

Figure 4.16 shows that the steepest wave (Wave 3) produces the highest normalized run-up values, especially in the directions 0°- 90°, where the wave impact is most direct. This is in line with the fact that steeper waves carry more energy on the structure resulting in higher run-up heights. Wave 3 also has more of an irregular pattern compared to the other cases, with a sharper decrease in run-up further down the monopile (especially at 112.5°). This trend is also observed in Wave 2 and Wave 4, which are both steeper waves ($S_{o,slope} \approx 0.042$). This is due to stronger reflection, more concentrated energy on the front face and less diffraction around the monopile. On the other hand, flatter waves like Wave 1 and 2 maintain a more consistent run-up across the full range. This is due to their more linear nature and less complex interaction with the monopile. This difference highlights the impact of wave steepness on the run-up on monopiles. Figure 4.17 shows a 3D visualization of the model and the pressure on the monopile.



(a)

(b)

(c)

Figure 4.17: 3D visualization of Wave 4 hitting the monopile at $t = 206.31\,\mathrm{s}$. **(a)** presents the velocity magnitude from $\approx 1L$ before to $\approx 0.5L$ after the monopile, **(b)** shows a zoomed-in view of the wave–structure interaction and **(c)** shows the dynamic pressure (p_{rgh}) along the monopile.

Now that the run-up is known, the observed results can be compared to the existing analytical methods. Therefore, Figure 4.18 shows the observed run-up vs the older analytical methods from section 1.1. The parameters from Table 4.6 were used to calculate the run-up. The breaking wave (Wave 5) is left out of this comparison since non of the formulas are valid for breaking waves.

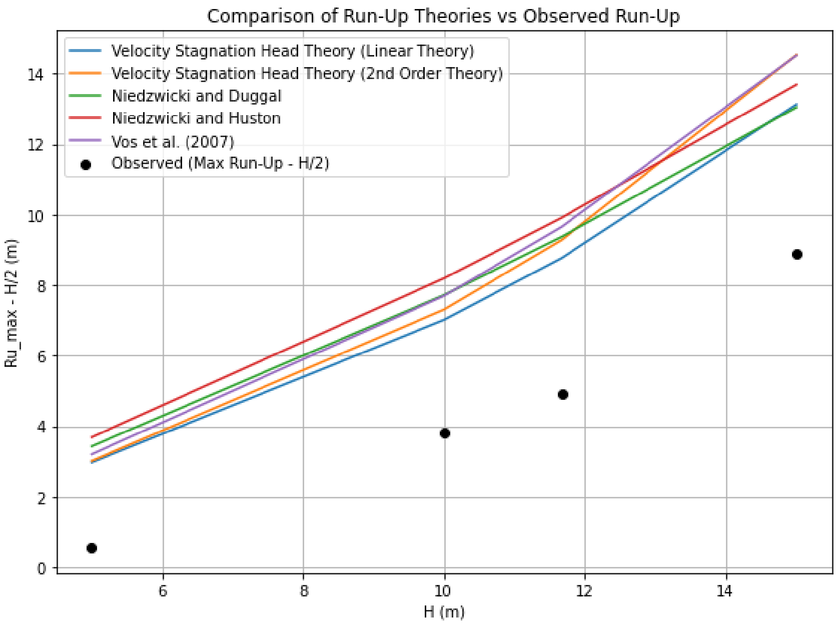
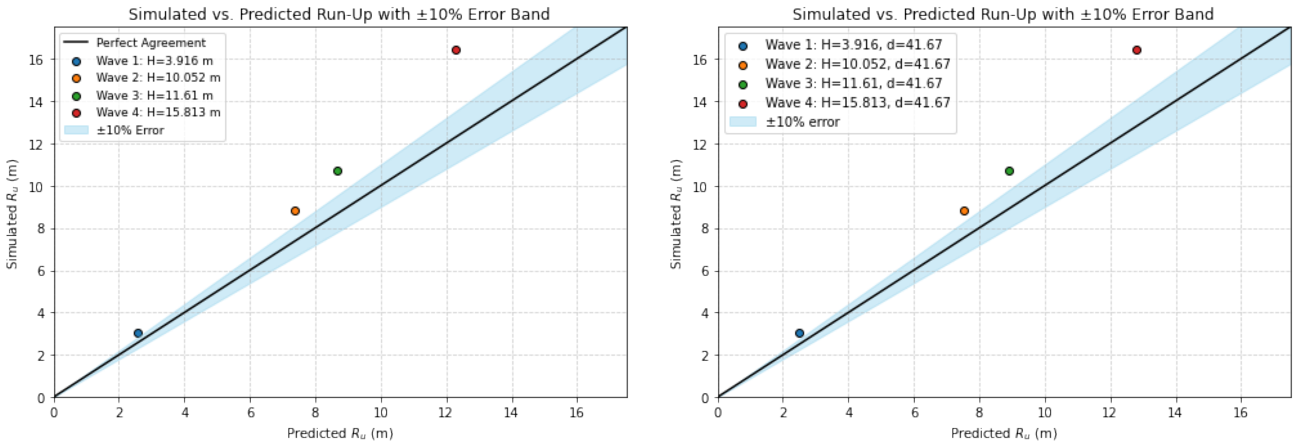


Figure 4.18: Comparison of wave theories vs Observed run-up

It becomes evident that all traditional analytical methods follow the trend of the run-up, but overestimate in all cases. The results reinforce that linear and second-order theory are insufficient for predicting run-up under non-linear wave conditions and that more advanced or fully non-linear approaches are necessary for accurate estimations.

The more advanced methods of Kazeminezhad and Etemad-Shahidi (2015) and Bonakdar et al. (2016) are compared to the observed run-up in Figure 4.19. Both formulae are independent of any wave theory and based on non-dimensional parameters. However, the formula of Kazeminezhad and Etemad-Shahidi (2015) is based on data that is limited to deep and intermediate water conditions and the formula of Bonakdar et al. (2016) is based on experiments with only non-breaking regular waves.



(a)

(b)

Figure 4.19: Comparison of wave theories vs Observed run-up. **(a)** Kazeminezhad and Etemad-Shahidi (2015) **(b)** Bonakdar et al. (2016)

The results show almost identical results for the formulas of Kazeminezhad and Etemad-Shahidi (2015) and Bonakdar et al. (2016). However, for both formulas the deviations start to grow for the larger, more non-linear waves due to the limitations and simplifications in their formulas. Therefore the the differences between the run-up height predicted by the formulas and the simulation are compared using statistical indicators, including bias, consistency index (I_a), squared correlation coefficient (R^2), and dispersion index (S_I). Their formulas can be found in Appendix D.2. Table 4.7 shows the statistical results between the predicted values of the two formulas and the simulation results.

Table 4.7: Statistical indicators for different cases.

Theory	**Bias**	I_a	R^2	S_I
Kazeminezhad and Etemad-Shahidi (2015)	2.0193	0.9141	0.9931	0.3130
Bonakdar et al. (2016)	1.8082	0.9365	0.9946	0.2685

Table 4.7 shows that both formulas exhibit high correlation with the numerical results, as indicated by R^2 values exceeding 0.99. This high correlation reflects that the predicted and simulated run-up values follow a similar trend, despite that the absolute values deviate. However, Bonakdar et al. (2016) demonstrates slightly better performance across all statistical indicators, including a lower bias (1.8082), a higher index of agreement ($I_a = 0.9365$), and a lower scatter index ($S_I = 0.2685$).

The index of agreement I_a assesses the degree to which predicted and observed values deviate from the observed mean, with values closer to 1 indicating better agreement. The scatter index S_I, represents the normalized standard deviation of the errors and is used as a measure of relative dispersion, lower values denote tighter clustering around the line of perfect agreement. This suggests that the model of Bonakdar et al. (2016) provides a slightly more accurate and consistent prediction of wave run-up, however both formulations are considered useful in predicting the run-up.

4.3.3. Force Analysis and Comparison to Existing Theories

Figures 4.20 - 4.22 show the non-dimensional total force in X and Y direction and the viscous force in Z direction. It clearly shows that the non-linearity of the horizontal wave force increases with the increase in steepness of the generated wave (waves 3 - 5). It also shows that under similar steepnesses the increase in period results in an increased amplitude of the wave force. Additionally, it can be seen that when increasing the period for mild steep waves the absolute value of the peak differs from the absolute value of the through. Lastly, when the non-linearity increases (Wave 5) there is a trend of a secondary peak. Both of these can be seen in Figure 4.23.

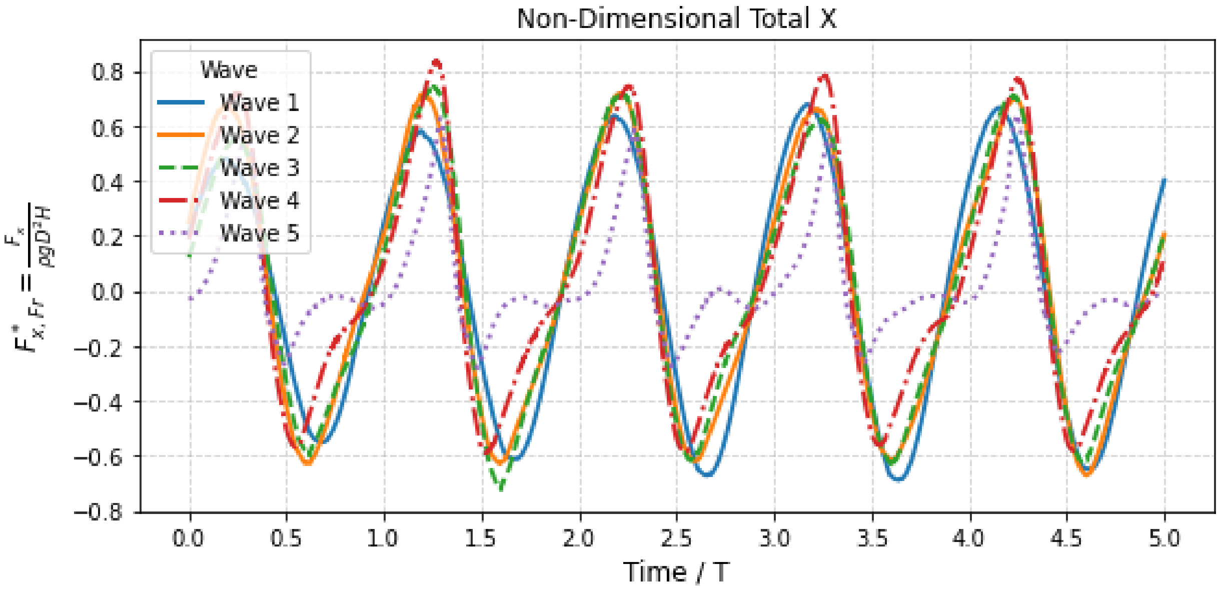


Figure 4.20: Total X force for different hydrodynamic conditions.

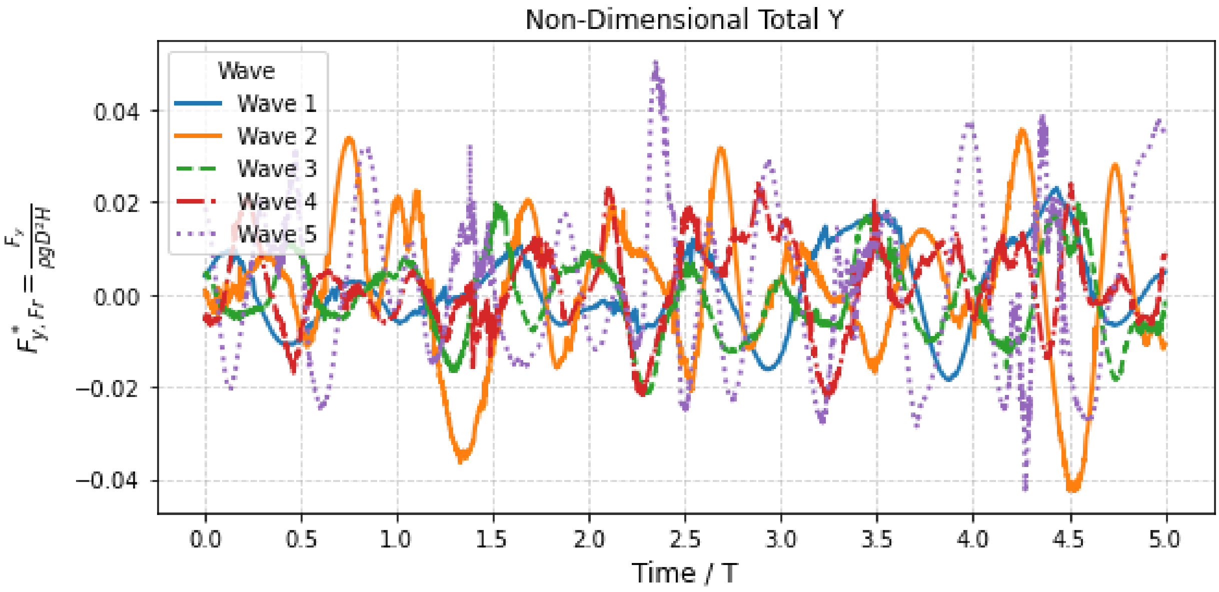


Figure 4.21: Total Y force for different hydrodynamic conditions.

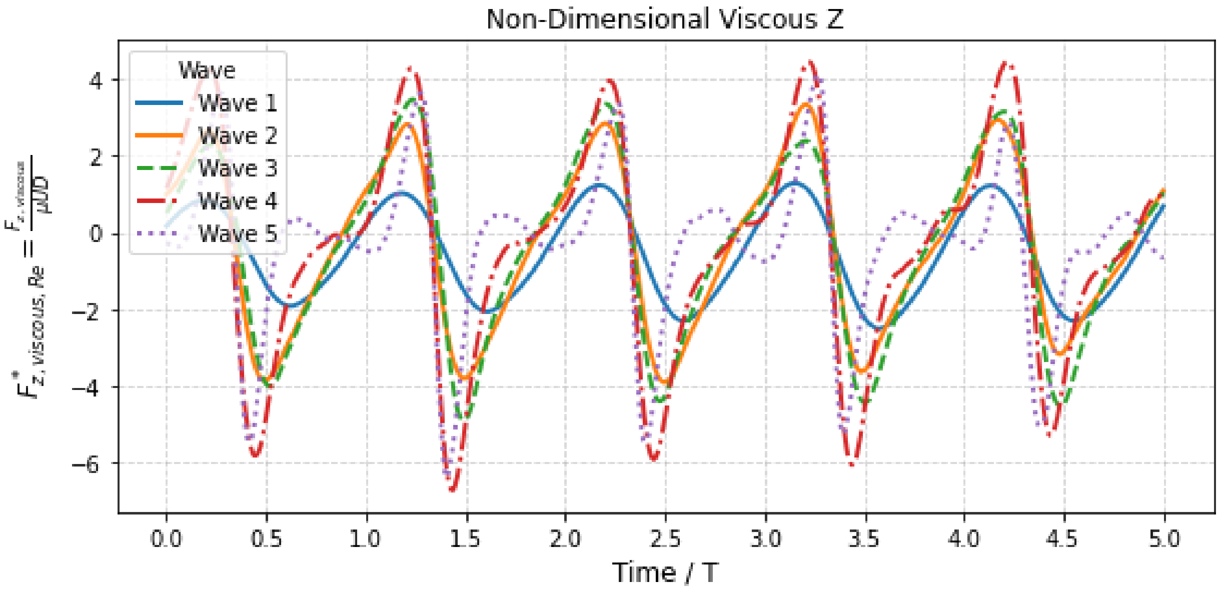


Figure 4.22: Viscous Z force for different hydrodynamic conditions.

For the total Y force, the plots show some signs of vortex shedding, especially for wave 2. However, there is still a lot of noise in the graph, which introduces a lot of irregularities in the total Y force making it hard to find any patterns. Lastly, it is visible that for an increase in wave steepness and therefore nonlinearity the viscous force increases. This is in line with the expectations since viscous forces arise due to the friction between the fluid and the surface, which becomes more significant at high velocities and larger gradients. For larger and steeper waves the wave energy is higher, which results in more turbulence and more complex flow dynamics around the monopile. This in combination with the thicker boundary layer of water results in higher and more non-linear viscous forces when compared to small and low steepness waves (wave 1).

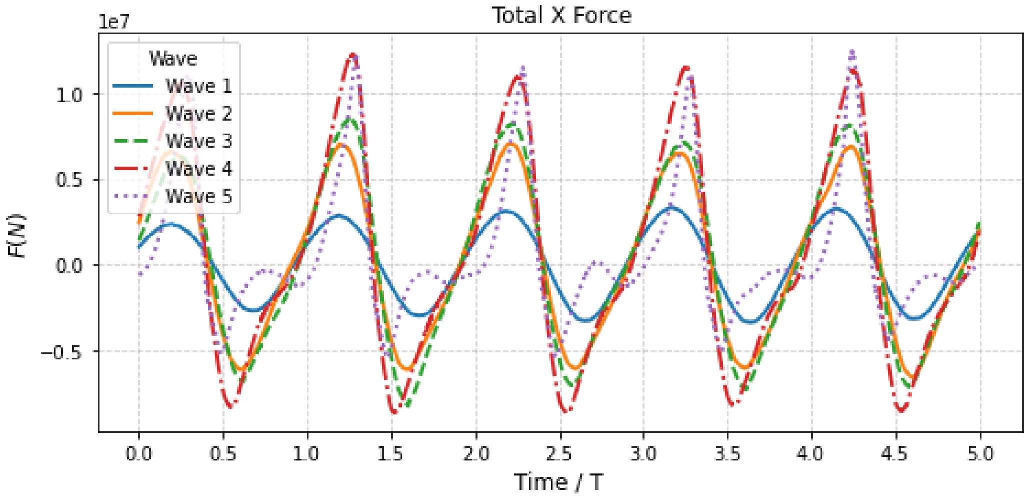


Figure 4.23: Total X force for different hydrodynamic conditions.

Now that all forces are known, they can be compared to analytical methods. All modeled wave cases fall within Region III of the force regime (Fig. 4.13), where both inertia and drag forces are significant. Therefore, Morison's equation is used to calculate the forces. The forces on the monopile were calculated using the d and T from Table 4.5 and the Avg H from Table 4.6, applying $C_D = 0.6$ and $C_M = 1.9$ from Sarpkaya and Isaacson (1981) (section 1.2). Velocity and acceleration were calculated using linear wave theory and second-order Stokes wave theory (Eqs. 4.3 - 4.6). And the results can be seen in Table 4.8.

Linear theory:

$$U(z) = \frac{\pi H}{T}\frac{\cosh\left(k(h+z)\right)}{\sinh(kh)} \tag{4.3}$$

$$\frac{dU}{dt} = \frac{2\pi^2 H}{T^2}\frac{\cosh\left(k(h+z)\right)}{\sinh(kh)} \tag{4.4}$$

Second-order Stokes theory:

$$U(z) = \frac{\pi H}{T}\frac{\cosh\left(k(h+z)\right)}{\sinh(kh)} + \frac{3\pi^2 H^2}{4T^2}\frac{\cosh\left(2k(h+z)\right)}{\sinh^3(kh)} \tag{4.5}$$

$$\frac{dU}{dt} = \omega\left(\frac{\pi H}{T}\frac{\cosh\left(k(h+z)\right)}{\sinh(kh)}\right) + \omega\left(\frac{3\pi^2 H^2}{2T^2}\frac{\cosh\left(2k(h+z)\right)}{\sinh^3(kh)}\right) \tag{4.6}$$

Table 4.8: Comparison of Maximum Forces in x-direction (Fx) and Velocities (U) for Linear (Lin.) and Stokes 2nd Order Wave Theory (2nd).

Wave	Max Fx model [MN]	Max Fx Lin. [MN]	Max Fx 2nd [MN]	Max U Lin. [m/s]	Max U 2nd [m/s]
1	3.33	2.68	3.66	1.27	2.13
2	7.12	5.67	20.08	2.80	10.52
3	8.51	6.19	26.78	3.16	13.96
4	12.84	7.61	52.27	4.19	25.75
5	12.46	4.67	72.63	4.35	36.69

The linear theory underestimates forces because it neglects non-linear effects, leading to an under prediction of horizontal velocities. In contrast, second-order Stokes wave theory overestimates forces near the surface by predicting unrealistically high velocities and accelerations without accounting for energy dissipation and wave breaking. These discrepancies reflect the limitations of Morison's equation, which assumes small steepness and amplitude, non-breaking waves, and linear kinematics, making it unsuitable for strongly non-linear or breaking conditions. Similarly, Froude–Krylov theory cannot be accurately applied, as it requires knowledge of the undisturbed pressure field around the structure, which is not known beforehand and cannot be determined solely from wave characteristics. Consequently, predicting hydrodynamic forces for non-linear waves through analytical methods alone becomes highly challenging. However, this model, which solves the full Navier–Stokes equations, offers a more reliable solution by capturing the non-linear wave-structure interactions.

4.4. Turbulence

Figure 4.24 shows the wave run-up at $0°$ on the monopile and the total X, Y, and viscous Z force. When running the same model with laminar versus turbulent flow (using the $k-\omega$-model), while keeping all other parameters the same, several small differences in the results are observed. The run parameters are given in Table 4.9. The computational domain has a length of 7.5L, which includes an inlet and outlet zone of 2L each. The domain has a depth of 20m and a width of 0.6L. All other parameters and setting are consistent with section 3.2.

Table 4.9: Simulation parameters used for the different flow types. All other parameters and settings are explained in section 3.2.

Run	T	H	S_o	Flow	Timestep	Co_{max}	ΔT_{max}	#Cells	End Time	Total Runtime
laminar	7	3.5	0.05	laminar	adjustable	0.6	0.05	3,211,800	105	$\approx$ 18.7 hours
$k-\omega$	7	3.5	0.05	turbulent	adjustable	0.6	0.05	3,286,200	77	$\approx$ 25.6 hours

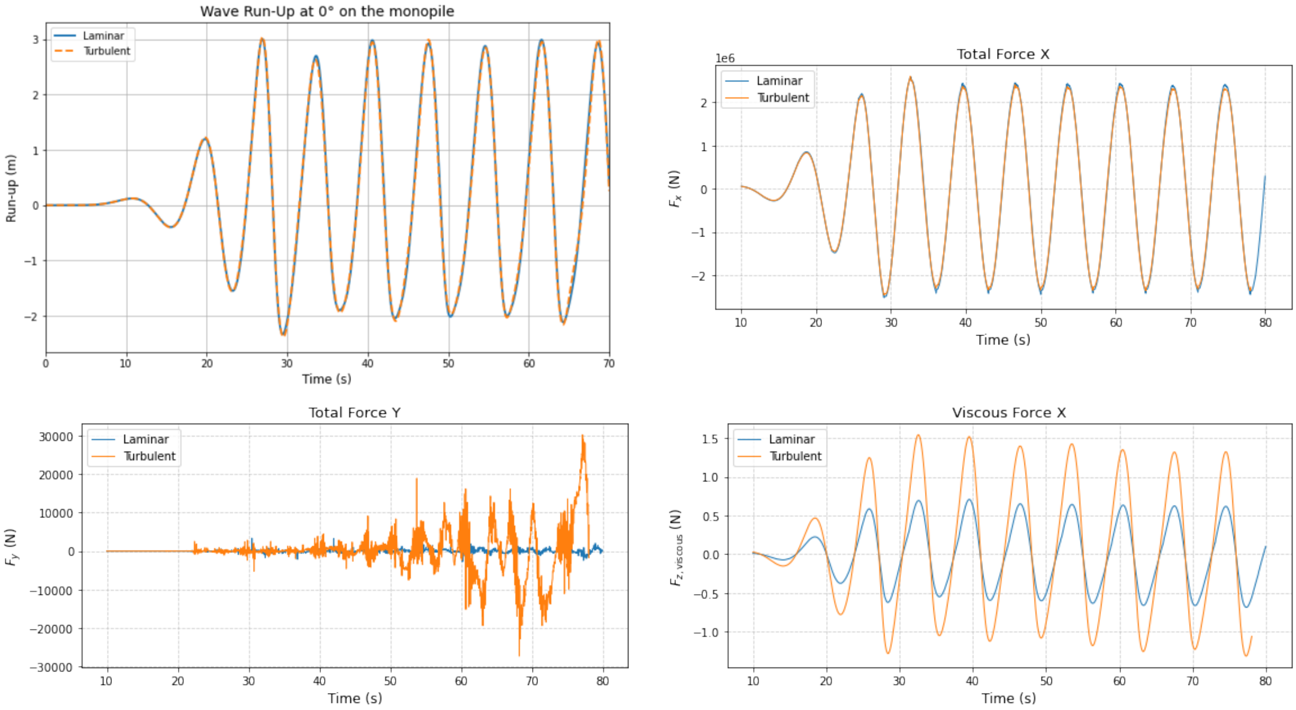


Figure 4.24: Wave run-up at 0° on the monopile, Total X, Total Y and Viscous Z force for laminar and turbulent ($k-\omega$-model) flow.

The wave run-up at the front of the monopile (0°) does not show significant differences. This suggests that, at this part of the structure, the flow behavior does not dramatically influence the wave dynamics in terms of run-up. This is because at the front the flow behavior and thus the wave run-up is dominated by inertial forces. This can also be seen in the plot of the total X force. This force is slightly smaller under turbulent flow conditions, because of the dissipation of energy that occurs in turbulent flows. However, since the total X force is dominated by the pressure force the differences are relatively small.

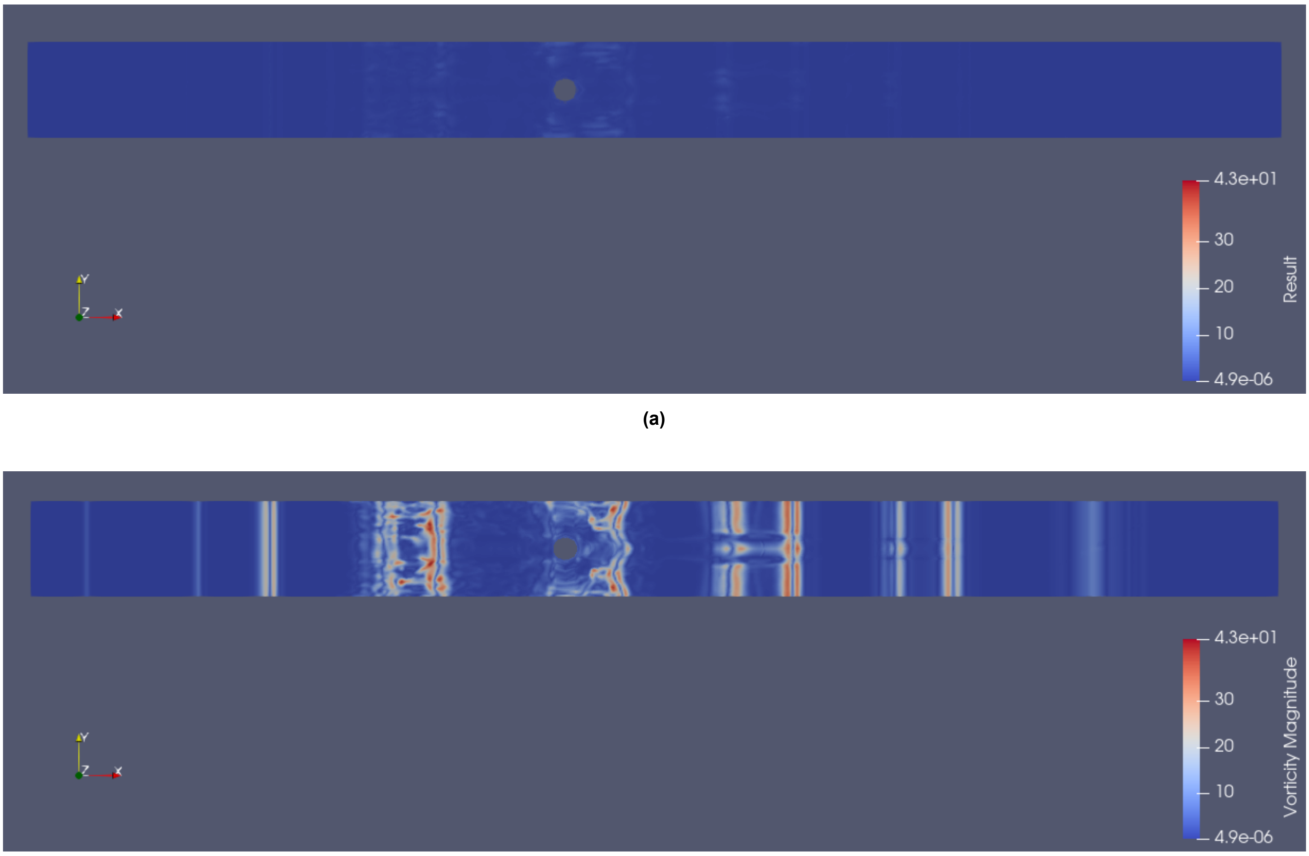
On the other hand, the total force in the Y-direction increases in turbulent flow. This increase occurs due to extra momentum transfer caused by chaotic, turbulent eddies, which enhance lateral forces acting on the structure. In turbulent flows, pressure forces become significant and more effective in the lateral direction because turbulence creates strong fluctuations and uneven pressure distributions around the monopile surface. Viscous forces also increase under turbulent conditions (Fig. 4.24). The turbulence model accounts for the additional momentum transfer from turbulent eddies and this effect is seen in the increase in viscous force. And since turbulence generates more irregularities in the flow, the forces generated due to these irregularities also increase.

Finally, in Appendix B, Figure B.5, it is evident that the run-up in the back, especially at $180°$, is higher in turbulent flow compared to laminar flow. This is because turbulence creates random eddies at the rear of the structure. These eddies trap and redistribute wave energy, causing water to pile up locally at the back side of the monopile. These eddies cause higher localized wave elevations at the back of the monopile, increasing the run-up in this region compared to the laminar flow.

Overall, the difference between laminar and turbulent flow in this context highlights how the complexity of turbulent flow dynamics can influence the distribution of forces and the wave run-up. The increased total Y force and viscous forces and the higher run-up at the back are directly related to the turbulent flows. However, in most cases, turbulent flow will reduce the run-up at the front of the structure, because of the dissipation of energy in the turbulent flow.

To simplify simulations and save massively on computational costs, laminar flow is often sufficient, as it provides a good approximation in scenarios where wave breaking is not significant and turbulence models reduce the critical run-up value. By avoiding the computational expense of turbulent flow simulation, e.g. the turbulent run took 4 times as long as the laminar run, one can still capture the essential behavior of wave run-up and forces in many cases. However, when simulating highly breaking waves or very steep waves, the turbulence model should be included to better capture the wave breaking physics. In these scenarios, turbulent flow provides more accurate results, as it accounts for the energy dissipation and chaotic fluid motion that occur during wave breaking.

Figure 4.25 shows the vorticity fields around the monopile for both the laminar and turbulent flow. It clearly shows the smooth and organized flow structures in the laminar flow and the strong chaotic eddies and irregular vortices in the turbulent flow. These turbulent vortices explain the increase in total Y force, viscous forces, and elevated run-up behind the monopile compared to laminar conditions. It also supports the similar run-up at the front of the monopile since the vorticity field does not differ significantly at that location.



(a)

(b)

Figure 4.25: Comparison of vorticity fields around the monopile for (a) laminar and (b) turbulent flow.

5 Discussion

The numerical model developed in this study shows a high level of accuracy, capturing peak wave run-up with over 95% accuracy when validated against experimental data, and proves to be a reliable tool for predicting both wave run-up and force distribution on monopiles. There are, however, always aspects of the model that can still be enhanced to further improve the accuracy of the simulations. The first area that can be improved is the mesh, although the current mesh arrangement is successful in capturing the peak run-up with less than 5% error. More mesh refinement around critical regions, such as the monopile surface and the free surface interface, would capture even more wave complexities. A finer mesh would better capture turbulence features and complex wave interactions, potentially increasing the model's predictive capability for highly non-linear and breaking waves. The problem with refining the mesh is that it would increase the number of total cells and lead to significantly longer computation times, increased data storage and additional computational costs. Given the already very small margin of error, the additional accuracy does not outweigh the large increase in computational resources required.

In addition, the numerical methods used within this research are mostly linear discretization schemes. These schemes offer stability, are less likely to cause divergency, computationally efficient and accurate for many cases. However, the application of higher-order numerical schemes has the potential to better represent non-linear processes of very steep and breaking waves. Higher-order numerical schemes may improve the sharpness of wave crests and improve the overall precision for complex and extreme wave conditions. However, employing these schemes would also be associated with greater computational costs and a greater chance of numerical instability. Additionally, this study utilized a Reynolds-Averaged Navier-Stokes (RANS) approach with the k-ω SST turbulence model. RANS efficiently resolves the average turbulence effects but does not resolve fine-scale turbulent eddies. More advanced methods, such as Large Eddy Simulation (LES), could improve accuracy, particularly for breaking waves. However, these methods are extremely computationally expensive. Since turbulence was found to have minimal impact on peak wave run-up, and given the model's high accuracy, the laminar or RANS-based approach is acceptable for predicting run-up since it is mostly dominated by large-scale wave dynamics rather than small-scale turbulent effects. Nevertheless, if detailed localized flow is of interest, a more advanced method should be considered.

The accuracy of the validation data itself is also worth noting. Data from de Vos et al. (2007) was used for validation and lacked labeling and descriptions since the experiment was conducted in 2007. This introduced uncertainty regarding the accuracy and reproducibility of the reference data. Errors between the validation data and the model could also stem from errors in the experiment or issues during data storage. This concern grew when some inconsistencies and unexpected values were discovered in the output of some wave gauges, suggesting potential experimental errors. This suggest that the model may actually be more accurate than it currently appears. Therefore, validating the model a second time against other confirmed experimental data would improve the reliability of the model even further (e.g. Ramirez et al. (2013)).

Also, a non-zero initial pressure force in the z-direction was observed in the simulations (Appendix B.1), caused by mesh discontinuities between the finer free surface region and the coarser mesh near the base. These discontinuities result in non-uniform cell normals, which affect the accuracy of the pressure force integration around the monopile. By applying a uniform mesh refinement around the whole length of the monopile, the issue can be resolved, resulting in a model that provides physically consistent and numerically stable predictions of the total vertical force.

As explained in section 3.2.1, the outlet zone was initially placed far from the monopile due to an overestimation of the reflection coefficient. This was done to avoid reflected waves interfering with the simulation results. However, it was later found that wave reflections in OpenFOAM are minimal when using the outlet zone of waves2Foam.

Therefore, in further research, the outlet can be placed much closer to the monopile without losing accuracy. This adjustment would reduce the total number of cells and lower the simulation time. Within the project folder (Drost, 2025), a model named 'Regular' is provided, which contains this adjustment and is therefore computationally less expensive. A description of this model is also included in the folder.

Lastly, due to limited time, computing power and available storage, only a finite number of simulations were performed for different scenarios and wave steepnesses. Additional simulations would further verify and confirm the conclusions drawn by this research, particularly in the comparison with analytical methods and detailed analysis of wave steepness effects. Increasing the number of runs over a larger set of conditions would produce even better insight into the sensitivities of the parameters.

It should however be taken into account that the implementations of these improvements stated above, such as mesh refinement or improved numerical schemes, would massively increase the computational costs and simulation times, as mentioned before. Researchers with greater computational power should consider these improvements. However, they should make a sincere judgment on the trade-off between the increase in accuracy and the significant increase in computational costs to ensure effective use of the model. Since it already captured the critical peak wave run-up at an accuracy of more than 95%.

The objective of this research was to create an accurate model to simulate wave run-up on monopiles and look at the influence of different hydrodynamic conditions. Traditional semi-empirical formulas over predict the run-up and are often not valid for steeper waves. Through the use of OpenFOAM, waves2Foam and computational fluid dynamics, this research developed an adaptive model that is capable of simulating complex wave-structure interactions with great accuracy, thereby improving the understanding and especially the prediction of wave run-up.

6 Conclusion

This research aimed to develop a numerical model using computational fluid dynamics to predict wave run-up and the associated hydrodynamic forces on monopiles. Due to the growing demand for offshore wind farms and the shift towards intermediate and more energetic waters, there is an increasing need for more accurate modeling approaches, as traditional semi-empirical methods often become invalid under these conditions. Below, the sub-questions of this thesis are addressed, followed by a conclusion regarding the main research question.

What is the current state of the art in calculating wave run-up and the associated hydrodynamic forces on monopiles?

The current state of the art in calculating wave run-up and the associated hydrodynamic forces on monopiles is discussed in section 1.1 and 1.2. Numerous semi-empirical methods exist for predicting wave run-up, however, most tend to overestimate run-up heights and are not valid for near-breaking or breaking wave conditions. These limitations reduce their applicability in more extreme sea states. Among the available approaches, the most accurate is the method proposed by Bonakdar et al. (2016), which provides improved predictions within its range of validity. With a bias of 1.8082, agreement index of 0.9365, R^2 value of 0.9946 and a scatter index S_I of 0.2685.

Morison's and Froude–Krylov theory are fundamentally limited in predicting hydrodynamic forces by their assumptions of linear wave behavior and non-breaking conditions. As a result, when applied to steep or near-breaking waves, linear wave theory underestimates hydrodynamic forces due to neglected non-linearities, while second-order Stokes theory tends to overestimate forces by predicting unrealistically high velocities and accelerations near the surface.

Which types of waves result in the highest wave run-up and forces on the monopile, and how should they be modeled?

Monopiles are most commonly installed in intermediate water depths, typically ranging from 25 to 45 meters. In these environments, the wave conditions are highly variable and often characterized by steep, non-linear waves. Due to the depth and wave characteristics, linear wave theory is no longer sufficient, and the wave behavior must be described using higher-order wave theories, such as Stokes 5th order theory. These non-linear wave effects significantly increase the wave run-up on monopiles and must be accurately captured to ensure safe and efficient structural design.

Additionally, steep waves and near-breaking conditions can generate slamming forces that are 8 to 9 times greater than those induced by linear waves. These highly transient forces pose serious risks to the design of offshore monopiles. Therefore, to capture the most critical loading scenarios and ensure accurate predictions of wave run-up and hydrodynamic forces, steep and near-breaking waves ($S_0 \approx 0.04 - 0.05$) should be modeled using appropriate non-linear wave theories, which can be selected in the *waveProperties.input* file in this model.

How can the wave generation, reflection, and absorption be modeled within computational fluid mechanics to ensure a continuous and realistic wave environment?

In this study, waves2Foam was used to generate and absorb waves, which proved to be an excellent toolbox for these purposes. It accurately recreates waves, for an inlet zone length of two-wavelengths, based on the defined wave theory and input parameters with reflection coefficients less than 1% for a two-wavelength length absorption zone.

How do scale effects impact the prediction of wave run-up in model-scale vs full-scale simulations?

Scale effects mainly impact viscous forces, which are not preserved under Froude scaling. While horizontal forces (F_x), governed by inertia and pressure, scale well across models, deviations in vertical and lateral forces arise due to Reynolds number sensitivity, especially at smaller scales.

However, for wave run-up and surface elevation, these effects are negligible, as viscous forces are typically several orders of magnitude ($\approx 10^5 - 10^6$) smaller than the dominant pressure forces. As a result, Froude scaling remains appropriate for predicting run-up and forces, as the influence on these quantities is negligible. Reynolds-based scaling can be used if detailed viscous effects are of interest.

What is the influence of different monopile diameters on the wave run-up under the same hydrodynamic conditions?

The analysis of the monopile diameter showed that increasing the diameter resulted in greater wave run-up at the front and reduced run-up in the shadow zones of the monopile. This is because smaller diameters allow more wave wrapping, enabling more energy to reach the sides of the monopile. Although larger diameters naturally result in higher absolute forces due to geometric scaling, non-dimensionalizing the forces effectively removes this dependency. As a result, the total non-dimensionalized forces remain largely consistent across different diameters. This indicates that monopile diameter primarily affects local run-up behavior and the absolute magnitude of the force, but has limited influence on the normalized total force.

How do wave characteristics influence the magnitude of wave run-up and forces on monopiles?

Different wave characteristics have a significant influence on the magnitude of wave run-up and hydrodynamic forces on monopiles, as shown in section 4.3. Among the various parameters, wave steepness ($S_o = H/L$) was found to be the most critical factor. Higher steepness leads to greater energy concentration, which results in higher wave run-up and increased hydrodynamic forces on the monopile.

The simulations demonstrated that waves with higher steepness and velocity (Wave 4), induce the highest run-up and force magnitudes. This is due to stronger wave-structure interactions and, in some cases, the occurence of wave breaking near the monopile. The influence of relative wave height (H/d), in combination with the bottom slope was also evident. These factors contribute to wave shoaling and breaking, which affect the amount of energy in the wave crest and, therefore, the run-up and hydrodynamic forces. Additionally, for waves with the same steepness, a greater wave period (T) was observed to increase the magnitude of the hydrodynamic forces.

While all wave parameters play a role, the results confirm that steep waves with high energy and velocity impose the most critical loading conditions.

How does turbulence affect the wave run-up and forces on monopiles and what turbulence model should be used to simulate these effects accurately?

The inclusion of a turbulence model is essential for accurately capturing the complex flow physics observed in reality, particularly the formation of eddies and momentum transfer behind the monopile. These effects influence the lateral forces (F_y), viscous forces ($F_{z,viscous}$), and wave run-up at the rear of the structure. In contrast, the impact on the maximum run-up at the front ($0°$) and the maximum in-line force (F_x) remains minimal, as these are largely governed by inertial and pressure-driven effects. Therefore, when computational resources are constrained, laminar flow simulations remain a reliable alternative, as turbulence generally leads to a slight reduction in run-up while significantly increasing the computational cost.

How can wave run-up and hydrodynamic forces on a monopile be accurately predicted under varying hydrodynamic conditions using computational fluid dynamics?

Wave run-up and hydrodynamic forces on a monopile can be accurately predicted under varying hydrodynamic conditions using computational fluid dynamics (CFD), as demonstrated in this study through the implementation of a numerical model in OpenFOAM. The model simulates regular wave conditions and computes both the wave run-up and associated forces on a monopile with high accuracy. Validation against experimental data from de Vos et al. (2007) confirmed that the model achieves a prediction error of less than 5% for peak wave run-up.

Unlike semi-empirical methods, which are limited in applicability and over-predict run-up and the associated forces, the CFD-based method developed here captures the complex, non-linear wave-structure interactions, including wave wrapping, and localized run-up effects. The model is robust and flexible. Additionally, it can simulate a wide range of wave conditions, from moderate to steep and near-breaking waves. While additional refinement in turbulence modeling, meshing, or numerical schemes may slightly increase the accuracy, the framework presented here remains a practical and accurate tool for offshore engineering applications.

In conclusion, this study demonstrates that computational fluid dynamics, when properly implemented and validated, can result in a reliable and accurate model, achieving 95% agreement with validation data, for predicting wave run-up and hydrodynamic forces on monopiles under realistic and varied hydrodynamic conditions.

7 Recommendations

While the validated model in this paper has demonstrated to be effective in accurately predicting wave run-up and force distribution on monopiles for a variety of cases, there are a few options that could prove worth exploring within future work toward improving the model's performance, especially in capturing very steep or breaking waves. Potential refinements, as discussed in Chapter 5, involving more advanced turbulence models, particularly for cases of high wave steepnesses and wave breaking, could improve the accuracy. Additionally, further mesh refinement near critical regions, such as around the monopile and at the free surface interface and the use of higher-order numerical schemes, could improve the model and its precision in predicting wave-structure interaction.

However, as previously mentioned, it is important to note that the current model already delivers accurate results for steep, near breaking waves, achieving over 95% accuracy when compared to experimental data. The above-described improvements are expected to provide no more than small increases in accuracy. While the associated computational time and costs would increase significantly. As an example, the addition of the RANS turbulence model already led to an increase of the computational time by a factor of 2.5.

It is therefore more interesting to use this model to analyze a broader variety of parameters influencing run-up, such as steepness, wavelength-to-diameter ratio, wavelength-to-depth ratio, and wave speed, among others. This study primarily focuses on the effect of wave steepness, however, extending the analysis to other parameters could lead to the development of generalized analytical relationships and a deeper understanding of wave-structure interaction.

Despite these challenges, the model remains a valuable tool for both industry and research applications. In cases where a smaller length of the domain is sufficient, the model can generate highly accurate predictions in relatively short computation times. This makes it an excellent alternative to analytical methods, especially for steep wave conditions, where existing analytical methods are unreliable and experiments are expensive. Given the model's capacity to provide more precise results, it could be a powerful tool for more detailed assessments of run-up on monopiles.

Furthermore, it is important to recognize that while this model currently runs regular waves, real-world conditions involve irregular waves, which can be even more complex. It would be interesting to see future researchers implement irregular waves in this model and investigate the impact of irregular steep or breaking waves. This can be done by changing the `waveProperties.input` file according to section 4.2.3.1 from the waves2Foam manual (Jacobsen, 2017). This would help provide a better understanding of wave run-up in real-life scenarios.

In addition to the monopile model in this research, a secondary model incorporating a perforated monopile was developed. This concept could offer a promising solution to reduce wave run-up and the associated forces acting on monopiles. The perforations could help dissipate some of the wave energy by allowing partial flow-through, resulting in lower run-up and smaller forces. However, this design introduces new structural challenges, including reduced strength and potential for fatigue. The perforated monopile is available in the project folder (Drost, 2025), but due to its high computational cost and the limited time and memory available in this study, it was not possible to fully run the model. While the model is functional and tested, additional convergence studies are needed to refine the mesh, reduce the number of cells, and decrease the run time.

In summary, future research should prioritize the use of the current model and extending it to simulate irregular wave conditions, as this would have a great impact on the model's real-world implementation. Only after these steps should computationally expensive upgrades, such as turbulence modeling and mesh refinements, be considered.

References

Agency, N. E. (2025). New offshore wind farms. Retrieved April 4, 2025, from https://english.rvo.nl/topics/offshore-wind-energy/new-offshore-wind-farms

Alsudani, A. (2024). *Reynolds-averaged navier-stokes simulations of unyawed and yawed rotating wheels* [Master's thesis, Technical University Delft]. https://repository.tudelft.nl/file/File_6022960d-9eee-4527-acff-46631bae92ac?preview=1

Andersen, J., Abrahamsen, R., & Andersen, T. L. (2020). Wave load mitigation by perforation of monopiles. *Marine Science and Engineering*. https://doi.org/10.3390/jmse8050352

Benitz, M., Lackner, M., & Schmidt, D. (1990). Hydrodynamics of offshore structures with specific focus on wind energy applications. *Renewable and Sustainable Energy Reviews*, *44*, 692–716. https://doi.org/https://doi.org/10.1016/j.rser.2015.01.021

Bonakdar, L., Oumeraci, H., & Etemad-Shahidi, A. (2016). Run-up on vertical piles due to regular waves: Small-scale model tests and prediction formulae. *Coastal Engineering*, *118*. https://doi.org/https://doi.org/10.1016/j.coastaleng.2016.08.008

Chella, M., Torum, A., & Myrhaug, D. (2012). An overview of wave impact forces on offshore wind turbine substructures. *Energy Procedia*. https://doi.org/https://doi.org/10.1016/j.egypro.2012.03.022

Chen, W., Warmink, J., & van Gent, M. (2021). Numerical modelling of wave overtopping at dikes using openfoam®. *Coastal Engineering*. https://doi.org/https://doi.org/10.1016/j.coastaleng.2021.103890

Clauss, G., Lehmann, E., & Östergaard, C. (1992). *Offshore structures* (1st edition). Springer-Verlag.

Commission, E. (2025). Renewable energy targets. Retrieved April 4, 2025, from https://energy.ec.europa.eu/topics/renewable-energy/renewable-energy-directive-targets-and-rules/renewable-energy-targets_en

Damian, S. (2013). *An extended mixture model for the simultaneous treatment of short and long scale interfaces* [Doctoral dissertation, Universidad Nacional Del Litoral].

de Vos, L., Frigaard, P., & de Rouck, J. (2007). Wave run-up on cylindrical and cone shaped foundations for offshore wind turbines. *Coastal engineering*, *54*, 17–29. https://doi.org/https://doi.org/10.1016/j.coastaleng.2006.08.004

Dean, R., & Dalrymple, R. (1991). *Water wave mechanics for engineers and scientists*. World Scientific Publishing Company.

Deshpande, S., Anumolu, L., & Trujilo, M. (2012). Evaluating the performance of the two-phase flow solver interfoam. *Computational Science Discovery*. https://doi.org/10.1088/1749-4699/5/1/014016

DOT. (2025). Slip joint. Retrieved April 8, 2025, from https://dotpower.nl/innovation/slipjoint/

Drost, J. (2025). Github: Msc thesis wave run-up on monopiles [Github repository]. https://github.com/hiddded/MSc-thesis-wave-run-up-on-monopiles

Edesess, A. (2018). *Simulated wave hydrodynamics and loading on an offshore monopile* [Doctoral dissertation, University College Cork, Ireland].

Fenton, J. (1985). A fifth-order stokes theory for steady waves. *Journal of Waterway, Port, Coastal and Ocean Engineering*, *111*. https://doi.org/https://doi.org/10.1061/(ASCE)0733-950X(1985)111:2(21

Fuhrman, D., Madsen, P., & Bingham, H. (2006). Numerical simulation of lowest-order short-crested wave instabilities. *Journal of Fluid Mechanics*, *145*. https://doi.org/10.1017/S0022112006001236

Ghasemian, M., Ashrafi, Z., & Sedaghat, A. (2017). A review on computational fluid dynamic simulation techniques for darrieus vertical axis wind turbines. *Energy Conversion and Managment*, *149*. https://sciencedirect.com/topics/engineering/large-eddy-simulation

Hadane, A. (n.d.). Openfoam tutorial: Simulation of the flow around a cylinder. Retrieved October 2024, from https://github.com/AsmaaHADANE/Youtube-Tutorials/tree/master

Hallermeier, R. (1976). Nonlinear flow of wave crests past a thin pile. *Journal of the Waterways, Harbors and Coastal Engineering Division*, *102*. https://doi.org/https://doi.org/10.1061/AWHCAR.000034

Jacobsen, N. (2017). *Waves2foam manual* (tech. rep.). Deltares. https://www.researchgate.net/profile/Niels-Jacobsen/publication/319160515_waves2Foam_Manual/links/5995c1e7aca27283b11b21a2/waves2Foam-Manual.pdf

Jacobsen, N., Fuhrman, D., & Fredsoe, J. (2012). A wave generation toolbox for the open-source cfd library: Openfoam (r). *International Journal for Numerical Methods in Fluids*. https://doi.org/10.1002/fld.2726

Jasak, H. (1996). *Error analysis and estimation for the finite volume method with applications to fluid flows* [Doctoral dissertation, University of London].

Josuttis, N. (2012). *The c++ standard library: A tuturial and reference* (2nd edition). Pearson Education, Inc.

Journée, J., & Massie, W. (2001). *Offshore hydromechanics* (1st edition). Delft University of Technology.

Kazemian, M. (n.d.). Comparative analysis for rans, urans, and ddes turbulence models. Retrieved October 14, 2024, from https://www.dlubal.com/en/support-and-learning/support/faq/005488?srsltid=AfmBOorS9ZLwupG-66LzM9Q9UxWKeEAc7oy5nYSwB3xDKusi-Cy6jNY8

Kazeminezhad, M., & Etemad-Shahidi, A. (2015). A new method for the prediction of wave runup on vertical piles. *Coastal Engineering*, *98*, 55–64. https://doi.org/https://doi.org/10.1016/j.coastaleng.2015.01.004

Kim, M.-H., & Yue, D. (1990). The complete second-order diffraction solution for an axisymmetric body - part 2. bichromatic incident waves and body motions. *Journal of Fluid Mechanics*, *211*, 557–593. https://doi.org/https://doi.org/10.1017/S0022112090001690

Kriebel, D. (1990). Nonlinear wave interaction with a vertical circular cylinder. part i: Diffraction theory. *Ocean Engineering*, *17*, 345–377. https://www.sciencedirect.com/science/article/pii/0029801890900296

Langton, C. (2000). Signal processing simulation newsletter - tutorial 1: Basic concepts in signal processing. *Complex to Real*. http://www.ece.iit.edu/~biitcomm/research/references/Other/Tutorials%20in%20Communications%20Engineering/TUTORIAL%201%20-%20Basic%20concepts%20in%20signal%20processing.pdf

Larsen, B. E., Fuhrman, D. R., & Roenby, J. (2019). Performance of interfoam on the simulation of progressive waves. *Coastel Engineering Journal*. https://doi.org/10.1080/21664250.2019.1609713

Lykke Andersen, T., & Brorsen, M. (2007). Horns rev ii, 2d-model tests: Impact pressures on horizontal and cone platforms from irregular waves. *Department of Civil Engineering, Aalborg University*. https://vbn.aau.dk/en/publications/horns-rev-ii-2d-model-tests-impact-pressures-on-horizontal-and-co

Lykke Andersen, T., Frigaard, P., & Damsgaard, M. (2011). Wave run-up on slender piles in design conditions - model tests and design rules for offshore wind. *Coastal Engineering*. https://doi.org/https://doi.org/10.1016/j.coastaleng.2010.10.002

MacCamy, R., & Fuchs, R. (1954). Wave forces on piles: A diffraction theory. *Technical memorandum series of the Beach Erosion Board*. https://apps.dtic.mil/sti/tr/pdf/AD0699406.pdf

Mansard, E., & Funke, E. (1980). The measurement of incident and reflected spectra using a least squares method. https://repository.tudelft.nl/file/File_9de94948-0378-45b4-9b3c-f837f4eb9768?preview=1

Mase, H., Kosho, K., & Nagahashi, S. (2001). Wave runup of random waves on a small circular pier on sloping seabed. *Journal of Waterway, Port, Coastal and Ocean Engineering*, *127*. https://doi.org/https://doi.org/10.1061/(ASCE)0733-950X(2001)127:4(192

Mojto, M., & Cabboi, A. (2024). The mechanical behaviour of a slip joint for an offshore wind turbine: First monitoring and modelling results. *Thin-Walled Structures*, *196*. https://doi.org/https://doi.org/10.1016/j.tws.2023.111482

Moukalled, F., Mangani, L., & Darwish, M. (2012). *The finite volume method in computational fluid dynamics* (5th edition, Vol. 113). Springer.

Murphy, G., Igoe, D., & Doherty, P. (2018). 3d fem approach for laterally loaded monopile design. *Computers and Geotechnics*, *100*, 76–83. https://doi.org/https://doi.org/10.1016/j.compgeo.2018.03.013

Negro, V., Esteban, M. D., & Lopez-Gutierrez, J. S. (2014). Uncertainties in the design of support structures and foundations for offshore wind turbines. *Renewable Energy*. https://doi.org/10.1016/j.rene.2013.08.041

Niedzwecki, J., & Duggal, A. (1992). Wave runup and forces on cylinders in regular and random waves. *Journal of Waterway, Port, Coastal and Ocean Engineering*, *118*, 567–654. https://doi.org/https://doi.org/10.1061/(ASCE)0733-950X(1992)118:6(615)

OpenFOAM. (2024). Openfoam user guide. Retrieved September 26, 2024, from https://www.openfoam.com/documentation/user-guide

OpenFOAM. (2025a). Fixed value. https://www.openfoam.com/documentation/guides/latest/doc/guide-bcs-fixed-value.html

OpenFOAM. (2025b). Inlet outlet. https://www.openfoam.com/documentation/guides/latest/doc/guide-bcs-outlet-inlet-outlet.html

OpenFOAM. (2025c). Mesh generation with the blockmesh utility. Retrieved March 7, 2025, from https://www.openfoam.com/documentation/user-guide/4-mesh-generation-and-conversion/4.3-mesh-generation-with-the-blockmesh-utility

OpenFOAM. (2025d). Numerical schemes. Retrieved March 14, 2025, from https://www.openfoam.com/documentation/user-guide/6-solving/6.2-numerical-schemes

OpenFOAM. (2025e). Pressure-inlet outlet velocity. https://www.openfoam.com/documentation/guides/latest/doc/guide-bcs-outlet-pressure-inlet-outlet.html

OpenFOAM. (2025f). Slip. https://doc.openfoam.com/2306/tools/processing/boundary-conditions/rtm/derived/wall/slip/

OpenFOAM. (2025g). Solution and algorithm control. Retrieved March 14, 2025, from https://www.openfoam.com/documentation/user-guide/6-solving/6.3-solution-and-algorithm-control

OpenFOAM. (2025h). Total pressure. https://www.openfoam.com/documentation/guides/latest/doc/guide-bcs-inlet-outlet-total-pressure.html

OpenFOAM. (2025i). Zero gradient. https://www.openfoam.com/documentation/guides/latest/doc/guide-bcs-general-zero-gradient.html

Peng, Z. (2014). Wave slamming impact on offshore wind turbine foundations. *Coast. Eng. Proc.* https://doi.org/10.9753/icce.v34.waves.43

Ramirez, J., de Vos, L., & Lykke Andersen, T. (2013). Large scale model test investigation on wave run-up in irregular waves at slender piles. *Coastal Engineering*. https://doi.org/https://doi.org/10.1016/j.coastaleng.2012.09.004

Ramirez, J., Frigaard, P., & Lykke Andersen, T. (2011). Numerical modelling of waave run-up: Regular waves. *Proceedings of the Twenty-first (2011) International Offshore and Polar Engineering Conference*, 342–346.

Rijksoverheid. (2025). Wind energy capacity, 1990-2023. Retrieved April 4, 2025, from https://www.clo.nl/en/indicators/en038630-wind-energy-capacity-1990-2023

Roenby, J., Bredmose, H., & Jasak, H. (2017). A computational method for sharp interface advection. *MARINE*. https://doi.org/https://doi.org/10.1098/rsos.160405

Roenby, J., Larsen, B., & Bredmose, H. (2016). A new volume-of-fluid method in openfoam. *Royal Society Open Science*. https://upcommons.upc.edu/bitstream/handle/2117/331052/Marine-2017-20_A%20new%20volume-of-fluid%20method.pdf?sequence=1&isAllowed=y

Sarpkaya, T., & Isaacson, M. (1981). *Mechanics of wave forces on offshore structures* (1st edition). Van Nostrand Reinhold.

Scholte, S. (2020). *The validation of wave overtopping over complex dikes using cfd* [Master's thesis, University of Groningen]. https://fse.studenttheses.ub.rug.nl/22715/1/mAppM_2020_ScholteSIO.pdf

Sodja, J. (2007). Turbulence models in cfd. *Faculty for mathematics and physics University of Ljubljana*. https://ascelibrary.org/doi/10.1061/%28ASCE%290733-950X%282001%29127%3A4%28192%29

Stansberg, C., Baarholm, R., & Kristiansen, T. (2005). Extreme wave amplification and impact loads on offshore structures. *Offshore Technology Conference*. https://eadn-wc05-12201999.nxedge.io/cdn/wp-content/uploads/2014/08/Extreme-Wave-Amplification-and-Impact-Loads-on-Offshore-Structures.pdf

Tang, Y., Shi, W., & J.You. (2020). Effects of spilling and plunging type breaking waves acting on large monopile offshore wind turbines. *frontiers in Marine Science*. https://doi.org/10.3389/fmars.2020.00427

van den Bos, L., & Sanderse, B. (2014). Uncertainty quantification for wind energy applications - literature review. *Centrum Wiskunde Informatica*. https://ir.cwi.nl/pub/26650/wind_uq_overview_20170822.pdf

Vijayakumar, P. (2023). Week 5 - turbulence modelling challenge. https://skill-lync.com/student-projects/week-5-turbulence-modelling-challenge-39

Walker, M. (2024). *Computational fluid dynamics study of perforated monopiles* [Master's thesis, Florida Institute of Technology].

wiki, O. (2025). Openfoam guide/the pimple algorithm in openfoam. Retrieved January 20, 2025, from https://openfoamwiki.net/index.php/OpenFOAM_guide/The_PIMPLE_algorithm_in_OpenFOAM

Wikipedia. (2009). *Offshore wind power*. Retrieved October 2, 2024, from https://en.wikipedia.org/wiki/Offshore_wind_power

Zhao, K., Wang, Y., & Liu, P. (2024). A guide for selecting periodic water wave theories - le méhauté (1976)'s graph revisited. *Coastal Engineering*. https://doi.org/https://doi.org/10.1016/j.coastaleng.2023.104432

Zhu, J., Gao, Y., Wang, L., & Wei, L. (2022). Experimental investigation of breaking regular and irregular waves slamming on an o-ffshore monopile wind turbine. *Marine Structures*. https://doi.org/https://doi.org/10.1016/j.marstruc.2022.103270

A
OpenFOAM Numerical Model Configuration

All OpenFOAM files and the full model setups are publicly available at:
https://github.com/hiddded/MSc-thesis-wave-run-up-on-monopiles.

A.1. Mesh

The meshing in OpenFOAM starts with *blockMesh*, a utility that creates structured hexahedral meshes. The *blockMeshDict* file is the blueprint for the initial mesh layout. It allows to set up a grid where vertices, blocks, edges, and boundaries are defined. The first step is to define the vertices. These points act as reference markers for the structure of the domain. After they are set, they are used to define blocks, which specify how these points are connected to form hexahedral cells. It is important to define them in counterclockwise order, starting from the bottom four vertices and then defining the corresponding top four vertices. This ensures a right-handed coordinate system, which is essential for defining the mesh orientation. Instead of straight-line connections, arcs can be specified to capture the geometry. This is used to create the monopile. Lastly, once the block structure is in place, the boundaries are assigned to the outer faces of the mesh. Boundaries specify different regions in the mesh, such as inlet, outlet, bottom, walls and atmosphere. Furthermore, it is only necessary to specify the external faces of the blocks. Internal faces between multiple blocks do not need to be specified, as OpenFOAM automatically recognizes their connectivity based on the block definitions.

An additional feature that can be applied to structured grids is mesh grading using the *simpleGrading* parameter. This controls how the size of the cells varies along the edges of the block. The expansion ratio, which is defined for every direction, defines how cell sizes change between the start and the end of a block, as illustrated in Figure A.1

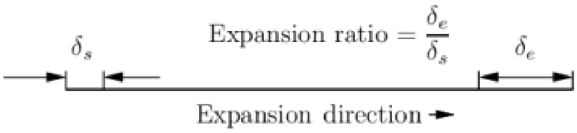


Figure A.1: Mesh grading along a block edge (OpenFOAM, 2025c).

It is particularly useful when simulating wave propagation, as it allows for gradual damping of the wave energy towards the damping zone by progressively increasing the cell size toward the end of the domain. This helps to prevent artificial reflections from the damping zone. A detailed explanation of all the blockMeshDict options can be found here (OpenFOAM, 2025c).

A.1.1. Numerical dissipation

Numerical dissipation is the loss of energy that occurs in simulations due to the discretization of the computational domain. One of the primary factors influencing numerical dissipation is the mesh size. The finer the mesh, the better it captures the physical processes, however, this is at a significantly higher computational cost. A finer mesh reduces numerical dissipation because it captures more details of the wave motion, ensuring that the energy is better preserved. To show this, a two-dimensional simulation, in the same x-z plane as the convergency studies (section 3.2.1), was run with $T = 2s$, $H = 0.05m$ and adjustable timestep ($Co < 0.6$), for two different mesh sizes. The coarse mesh is equal to the mesh resolution from section 3.2.1 and the finer mesh is the coarse mesh with a refinement of 2 (= #cells * 4). Figure A.2 shows the surface elevation at two locations (1L and 10L after the inlet zone) for the coarse and fine mesh. At 1L after the inlet, both meshes produce nearly identical wave profiles, indicating minimal numerical dissipation at the beginning. However, at 10L after the inlet, deviations appear between the coarse and fine meshes, with the coarse mesh showing slightly more damping of wave amplitude. This suggests that numerical dissipation builds up over longer distances, and finer resolution better preserves wave energy over propagation.

However, the runtime for the coarse mesh was 2131.6 s for 24 waves and the runtime for the 2D fine mesh was 4360.5 which is two times as long. This effect is even larger in 3D simulations, where if we refine the mesh by a factor of 2, the number of cells increases by a factor of 8 (2*2*2) instead of a factor of 4 (2*2). The computational demand in 3D is also significantly higher, and numerical dissipation is also more prominent because wave energy is spread across more degrees of freedom, making it harder to maintain accuracy over long simulations. While reducing numerical dissipation through mesh refinement improves accuracy, it requires longer run times and more computational resources. Therefore, finding the optimal balance between mesh resolution and the available computational power is crucial, especially in 3D cases, where dissipation is naturally more significant than in 2D.

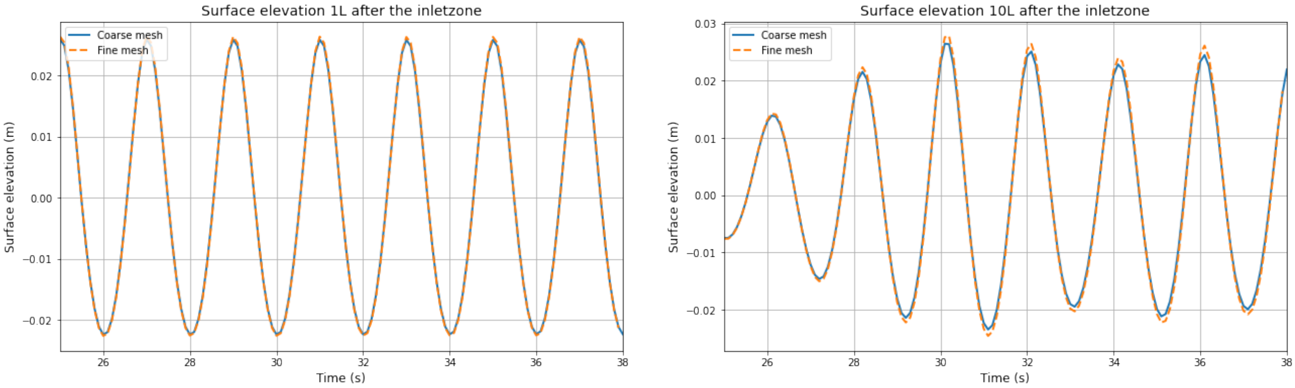


Figure A.2: Surface elevation at two different locations for the given mesh above (coarse) and a refined mesh with a refinement factor of 2 (fine).

A.1.2. Non orthogonality

The original mesh was based on a tutorial named Simulation of the flow around a cylinder and can be seen in Figure A.3 (Hadane, n.d.). By using this method a highly refined region around the monopile is created. This refinement captures the fluid-structure interactions very well since the cells are fine. However, it also introduces a high degree of non-orthogonality in the mesh.

A big lesson learned is that OpenFOAM does not work well when cells are non-orthogonal, even with the non-orthogonal correction (section 2.1.3). It is particularly problematic in regions close to boundaries and at mesh corners, where grid cells tend to be highly skewed due to geometric constraints. In these areas, the gradient and Laplacian approximations become inaccurate, leading to poor pressure resolution and unstable simulations. The primary reason for pressure instability in highly non-orthogonal meshes lies in how pressure gradients are computed using the Laplacian operator ($\nabla^2 p$). The discretized pressure equation involves flux interpolation between neighboring cells, assuming that face normal vectors align with the cell-to-cell direction. However, in non-orthogonal meshes, these vectors do not align, leading to miscalculated pressure gradients.

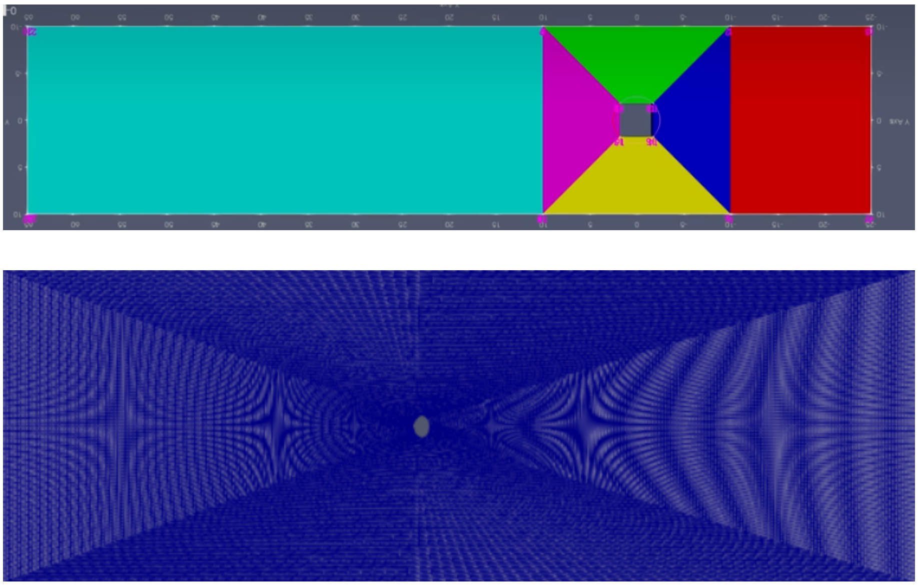


Figure A.3: Visualization of the original mesh: (a) the block decomposition used in the domain, and (b) shows the fine grid resolution.

A.2. Wave generation and absorption

Wave generation and absorption is one of the main functions of waves2Foam. The wave properties and conditions for wave generation are defined in the *waveProperties* file under *inletCoeffs*. Which includes the type of waves, their height, period, depth, direction and other wave specific parameters. In this research the most commonly used wave type is the stokes fifth order wave. The wave properties for this wave type are defined as follow, both the mandatory and non-mandatory parameters are given. The default values for non-mandatory parameters are specified.

```
waveType     stokesFifth;

height       <scalar>;     // [m]
period       <scalar>;     // [s]
depth        <scalar>;     // [m]
stokesDrift  <scalar>;     // [m/s]
direction    <vector>;     // [-]
Tsoft        <scalar>;     // [s] Default value: period
phi          <scalar>;     // [radians]
```

Code Fragment A.1: Stokes Fifth Order wave properties

Configurations for different wave scenarios used in this research (stokes second and streamfunction) are specified in Appendix A.2.1, where their Code Fragment is given. Other configurations can be found in the waves2Foam manual (Jacobsen, 2017).

To minimize unwanted reflections and ensure realistic wave behavior, relaxation zones are implemented. The inlet relaxation zone is required to smoothly introduce the generated waves into the computational domain. The inlet relaxation zone is specified in the *waveProperties* file under *relaxationZone* within the *inletCoeffs* as follows. The outlet relaxation zone is also under *relaxationZone* but is within the *outletCoeffs*.

```
relaxationZone
{
    relaxationScheme    <word>;
    relaxationShape     <word>;
    relaxationWeight    <word>;
}
```

Code Fragment A.2: Fundamental definition of a relaxation zone

The relaxation shape specifies the location of the relaxation zone within the computational domain. At this point there are four relaxation shapes available in waves2Foam: rectangular, semi-cylindrical, cylindrical and frozen (Jacobsen, 2017). Since the computational domain is rectangular, the rectangular relaxation shape is used in this research. It is defined using the coordinates of the two diagonal corner points of a rectangle, along with the direction of one of the lateral sides and the relaxation direction. The formatting of the relaxation shape is presented in the code fragment below.

```
relaxationShape   Rectangular;
relaxType         <word>;
startX            <point>;
endX              <point>;
orientation       <vector>;
```

Code Fragment A.3: Rectangular relaxation shape

The *relaxationScheme* determines if the relaxation is implemented. When specified as 'Empty' it turns off the relaxation. By specifying it as 'Spatial' the relaxation scheme creates the relaxation zone as stated in *relaxationZone*. In the inlet zone, OpenFOAM enforces wave conditions as stated in *waveProperties.input*. Within the outlet zone, OpenFOAM performs the relaxation according to Eq. A.1:

$$\phi = (1 - w_R)\phi_{\text{target}} + w_R\phi_{\text{computed}} \tag{A.1}$$

Here, the weighting function $w_R \in [0, 1]$ is defined as an exponential weight and explicitly corrects the fields α and u according to Eq. A.1 each time step before the solution to the pressure-velocity coupling (Jacobsen et al., 2012). The exponential weight distribution, as presented by Fuhrman et al. (2006), is defined as follows:

$$w_R = 1 - \frac{\exp \sigma^p - 1}{\exp 1 - 1} \tag{A.2}$$

The relaxation function satisfies $w_R(0) = 1$ at the interface between the non-relaxed part of the computational domain (where the fully computed solution is applied) and the relaxation zone and $w_R(1) = 0$ at the boundary (where the fully theoretical solution is applied), as illustrated in Figure A.4. This weighting function is crucial in wave generation and absorption zones. At the inlet, the solution is fully theoretical, meaning the prescribed wave conditions dictate the behavior of the flow, ensuring accurate wave generation. On the other hand, at the outlet, the weighting function smoothly transitions from the computed solution to the theoretical target solution, effectively damping outgoing waves and minimizing reflection back in the domain.

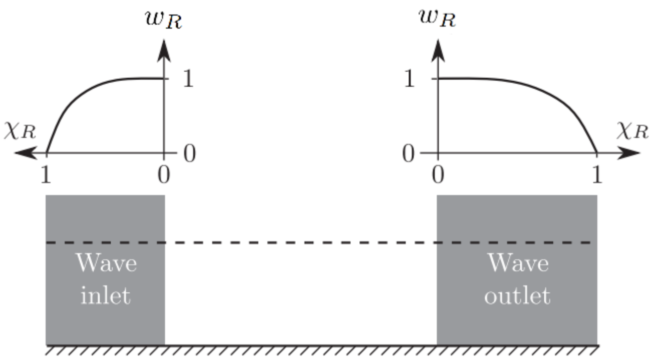


Figure A.4: Sketch of w_R for the inlet and outlet relaxation zones (Jacobsen et al., 2012).

A.2.1. OpenFOAM configurations for different wave theories

```
waveType     stokesSecond;

depth        <scalar>;     // [m]
period       <scalar>;     // [s]
direction    <vector>;     // [-]
height       <scalar>;     // [m]
phi          <scalar>;     // [radians]
Tsoft        <scalar>;     // [s] Default value: period
```

Code Fragment A.4: Stokes Second Order wave properties

```
waveType    streamFunction;

height      <scalar>;    // [m]
depth       <scalar>;    // [m]
N           <scalar>;    // [-] Number of components
Niter       <scalar>;    // [-] Number of iterations
phi         <scalar>;    // [radians]
direction   <vector>;    // [-]

EITHER:
specifyPeriod false;     // [boolean]
waveLength   <scalar>;   // [m]

OR:
specifyPeriod true;      // [boolean]
period       <scalar>;   // [s]

EITHER:
specifyEuler true;       // [boolean]
eulerVelocity <scalar>;  // [m/s]

OR:
specifyEuler false;      // [boolean]
stokesVelocity <scalar>; // [m/s]
```

Code Fragment A.5: Stream Function Theory wave properties

```
/*--------------------------------*- C++ -*----------------------------------*\
FoamFile
{
    version 2.0;
    format  ascii;
    class   dictionary;
    object  waveProperties;
}
\\ * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
seaLevelAsReference true;
seaLevel            0;
relaxationNames     ( inlet outlet );
initializationName  inlet;

inletCoeffs
{
    waveType            stokesFifth;
    height              0.12;
    period              1.6;
    depth               0.647;
    stokesDrift         0;
    direction           ( 1 0 0 );
    Tsoft               1.6;
    phi                 0;
    waveNumber          (1.86524 0 0);
    omega               3.92699;

    relaxationZone
    {
        relaxationScheme    Spatial;
        relaxationShape     Rectangular;
        beachType           Empty;
        relaxType           INLET;
        startX              ( -6 0 -0.647 );
        endX                ( 0 1.5 0.353 );
        orientation         ( 1 0 0 );
    }
}
outletCoeffs
{
    waveType            potentialCurrent;
    U                   ( 0 0 0 );
    Tsoft               1.6;

    relaxationZone
    {
        relaxationScheme    Spatial;
        relaxationShape     Rectangular;
        beachType           Empty;
        relaxType           OUTLET;
        startX              ( 44 0 -0.5 );
        endX                ( 50 1.5 0.353 );
        orientation         ( 1 0 0 );
    }
}
// ************************************************************************* //
```

Code Fragment A.6: OpenFOAM full Stokes Fifth Order waveProperties file for the validation experiment.

A.2.2. Wave gauge and probe configuration in OpenFOAM

In OpenFOAM, wave gauges and probes are defined in the *constant/probeDefinitions* file, where their locations and properties are specified. Their configuration is given below:

```
<name of wave gauges>
{
    type waveGauge;           // Definition of the type
    add       (0 0 1);        // Length and orientation of gauge
    axis       z;             // Vertical axis

    // Insert point distribution
}

<name of probes>
{
    type probeGauge;

    outputInterval <label>;  // The frequency of outputting
    fields (<name0> <name1>); // The fields to sample

    // Insert point distribution
}
```

Code Fragment A.7: Common settings for definition of wave gauges and probes

The point distributions to be inserted in the Code Fragment above are defined below:

circularDistribution Defines a circular distribution with a certain radius around a given centre. The point distribution is defined as:

```
pointDistribution circularDistribution;

N       4;              \\ number of gauges (equally distributed)
centre  (0 0 0);
radius  1;
```

Code Fragment A.8: A circular point distribution

lineDistribution Defines a linear distribution along a line defined by a starting and an end point. It is possible to have the distribution stretched along the line; resulting in a non-equidistant point distribution.

```
pointDistribution lineDistribution;
N       4;              \\ number of gauges (equally distributed)
linestart  (0 0 0);
lineend    (1 0 0);
stretch    1.;
```

Code Fragment A.9: A linear point distribution

A.2.3. Configuration of the PIMPLE algorithm

```
PIMPLE
{
    momentumPredictor        yes;        // Solve momentum eq. before pressure corr.
    nCorrectors              2;          // Number of correction steps for velocity
    nNonOrthogonalCorrectors 1;          // Number of non-orthogonal corrections
    nOuterCorrectors         1;          // Number of outer pressure corrections
    nAlphaCorr               1;          // Number of alpha corrections
    cAlpha                   1;          // Alpha correction scaling factor
    correctPhi               yes;        // Correct phi field after each iteration
    checkMeshCourantNo       true;       // Check Courant number for stability
}
```

Code Fragment A.10: Common basic set of input entries for PIMPLE

B

Wave Run-Up and Forces results

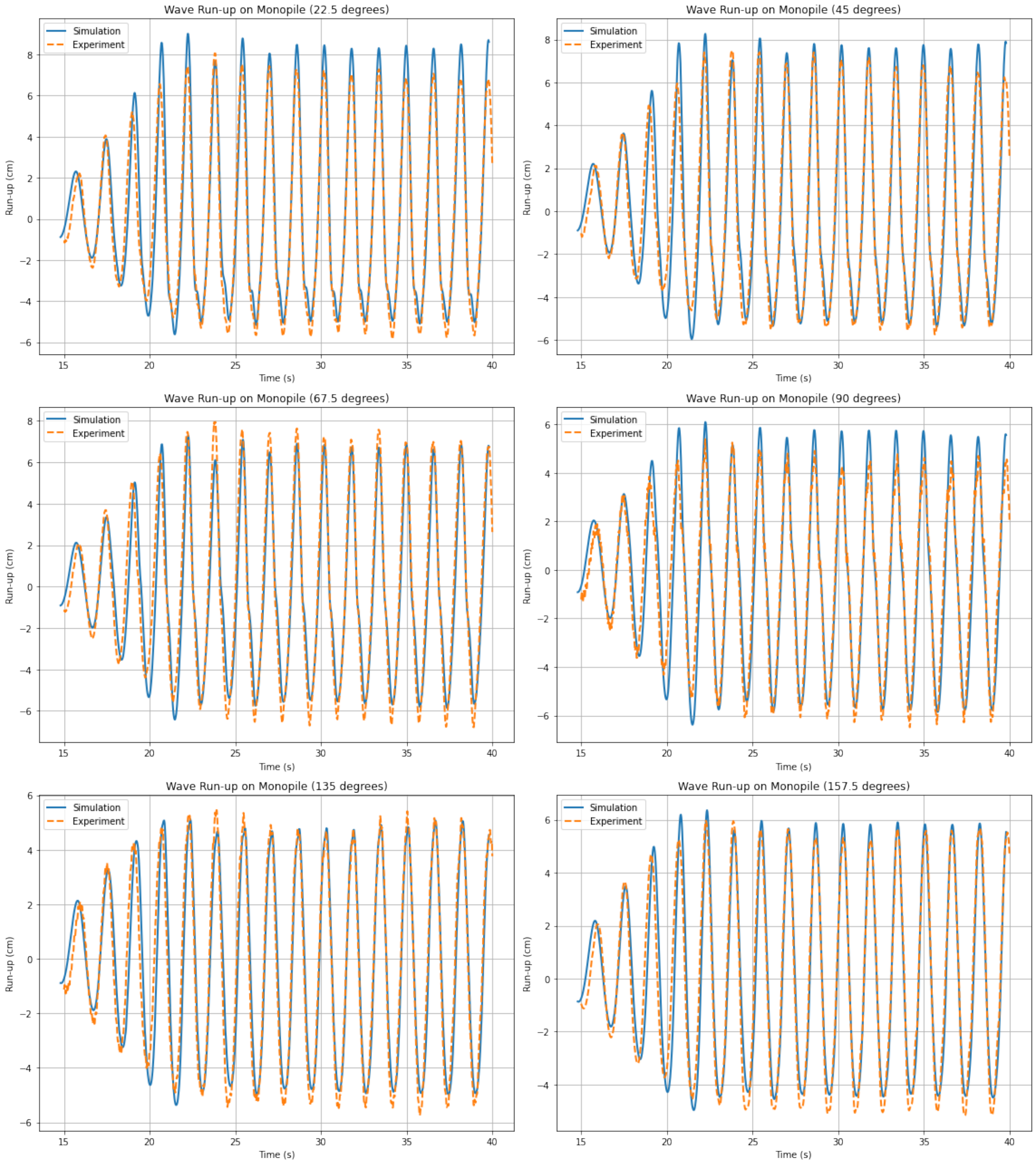
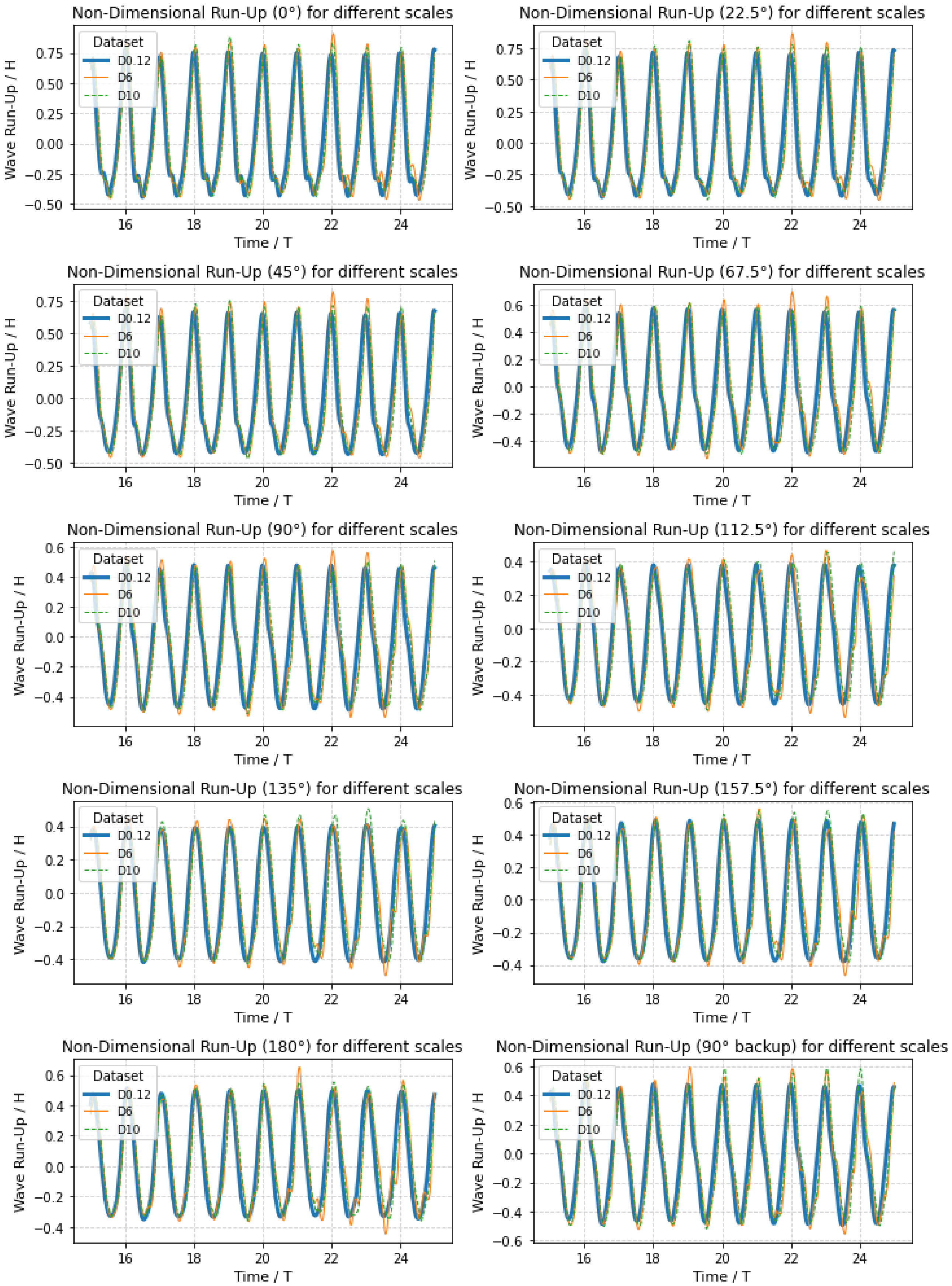


Figure B.1: Wave Run-Up on monopile comparison between experiment and simulation for different angles, $112.5°$ is left out due to measurement errors in the experiment.



Non-Dimensional Run-Up (0°) for different scales

Non-Dimensional Run-Up (22.5°) for different scales

Non-Dimensional Run-Up (45°) for different scales

Non-Dimensional Run-Up (67.5°) for different scales

Non-Dimensional Run-Up (90°) for different scales

Non-Dimensional Run-Up (112.5°) for different scales

Non-Dimensional Run-Up (135°) for different scales

Non-Dimensional Run-Up (157.5°) for different scales

Non-Dimensional Run-Up (180°) for different scales

Non-Dimensional Run-Up (90° backup) for different scales

Figure B.2: Non-dimensional wave run-up on a monopile at different scales. D0.12 represents the experiment from de Vos et al. (2007), while D6 and D10 are Froude-scaled by factors of 50 and 83.33, respectively.

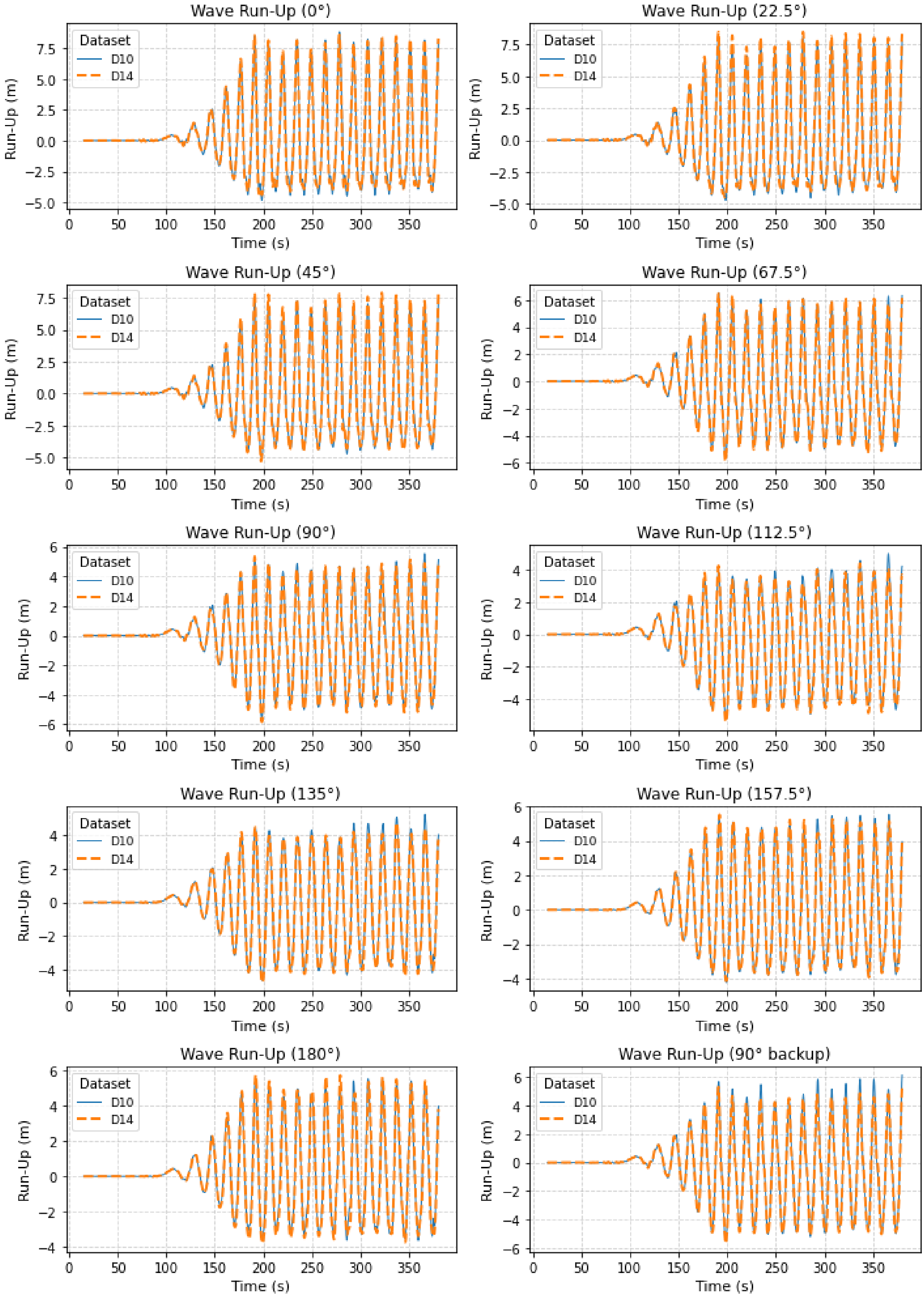


Figure B.3: Wave run-up for all angles on the monopile for two different diameters (D10 and D14) under the same wave characteristics (section 4.2.

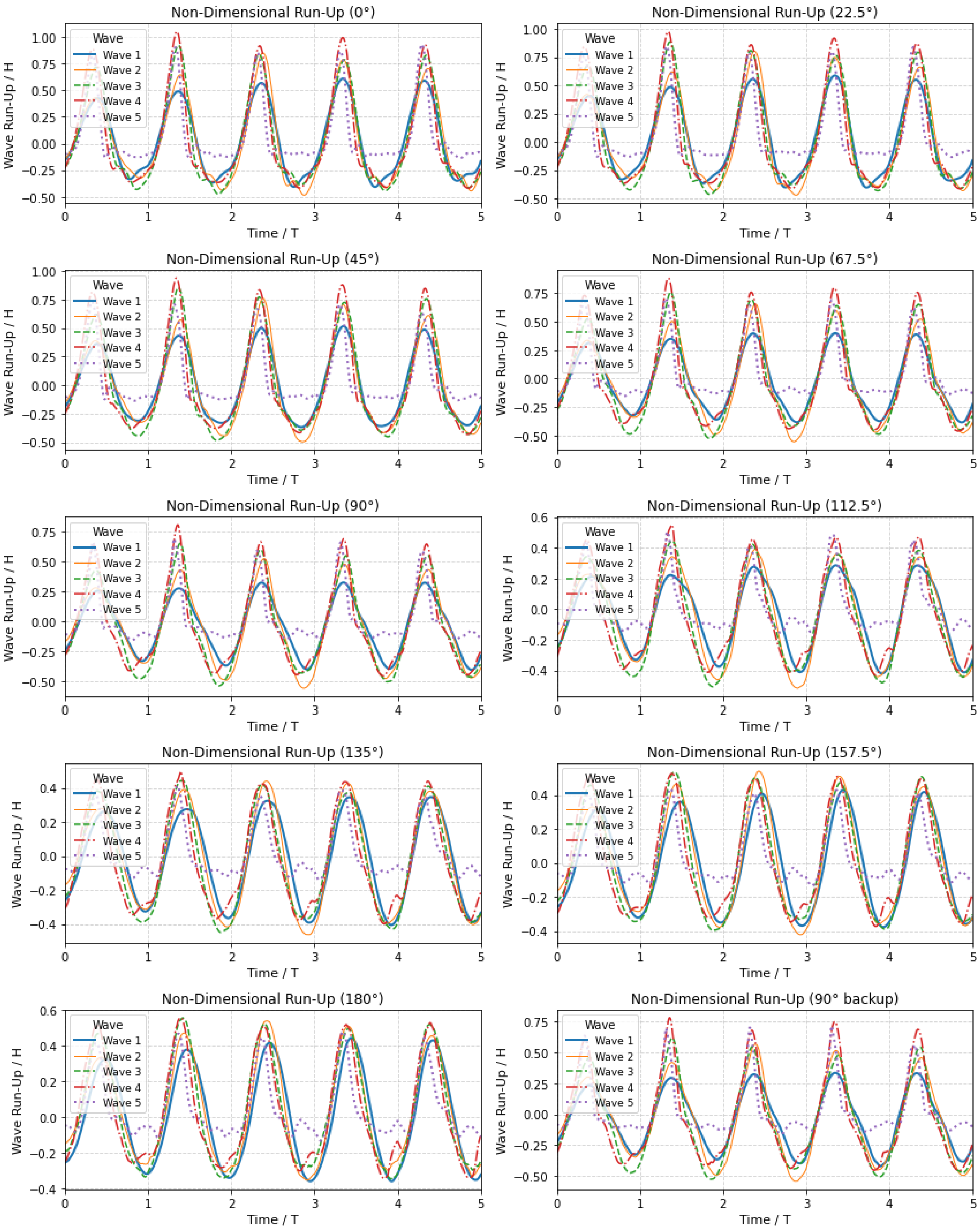


Figure B.4: Non-dimensional wave run-up for all angles on the monopile for different wave characteristics.

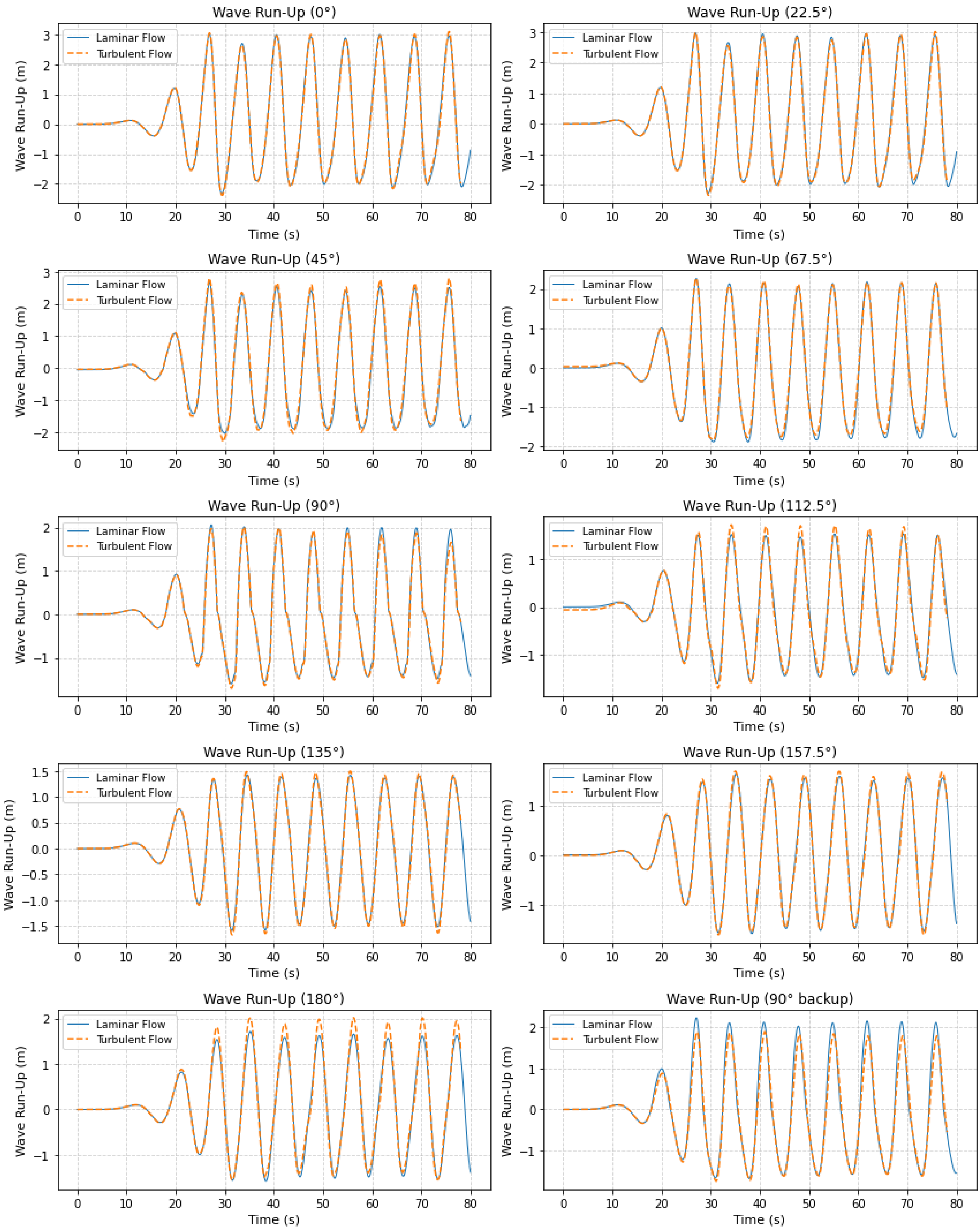


Figure B.5: Wave run-up on a monopile for a turbulent and laminar flow. All other parameters are the same. $H = 3.5m$, $T = 7s$, $d = 20m$, $S_o = 0.05$.

B.1. Non-zero initial pressure force (z-direction)

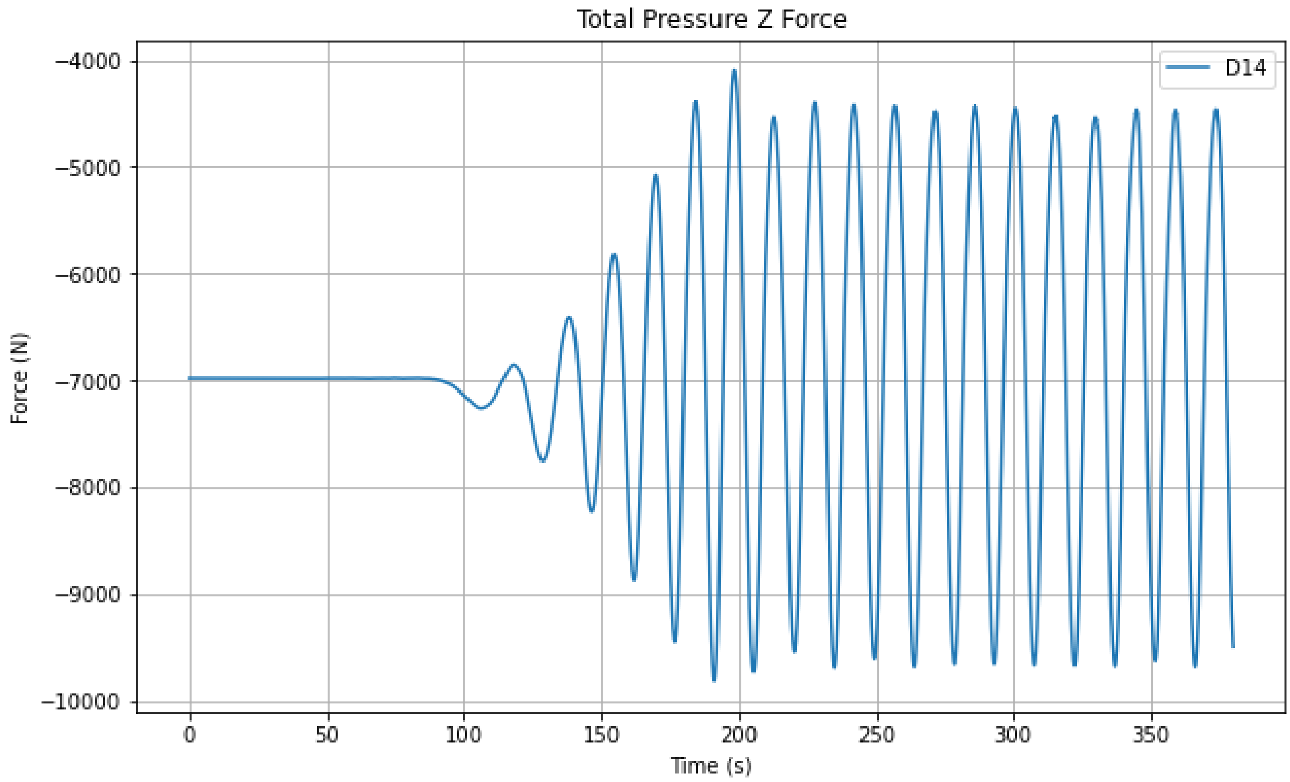


Figure B.6: Total vertical (z-direction) pressure force on the monopile at $t = 0$ for the D14 case, showing an initial force caused by non-zero surface normals at the interface between the refinement region and the surrounding mesh.

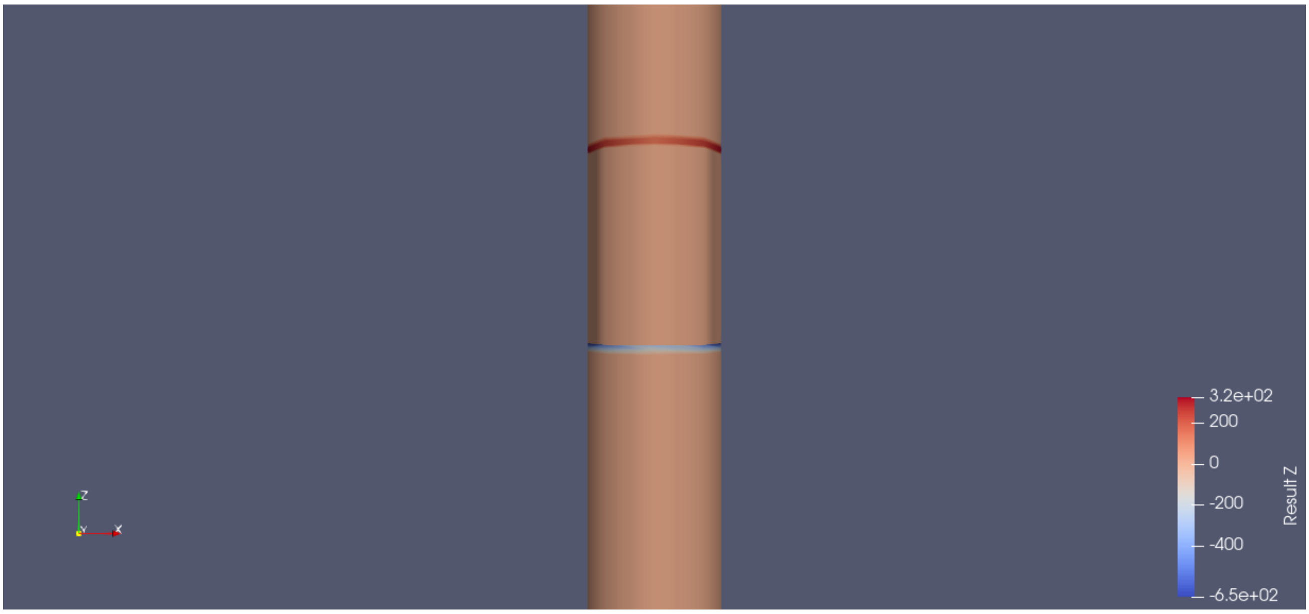


Figure B.7: Visualization of the z-component of the pressure force per unit area at $t = 0$ for the D14 case, computed as $-p_{\text{rgh}} \cdot \vec{n}$. This represents the distributed vertical pressure load on the surface. By integrating this field over the monopile surface area, the total pressure z force is obtained.

C
Mansard and Funke Reflection method

To analyze wave reflections in the numerical wave tank, a Fourier Transform is applied to the time series data of the free surface elevation at three different gauges. This allows the extraction of the dominant wave frequency, amplitude, and phase, which are then used in the reflection analysis.

The Discrete Fourier Transform (DFT) of a discrete time series $\eta(t)$ with N samples is given by:

$$\hat{\eta}(f_k) = \sum_{n=0}^{N-1} \eta_n e^{-i2\pi kn/N} \tag{C.1}$$

where:

- $\hat{\eta}(f_k)$ is the complex Fourier coefficient at frequency f_k,
- η_n is the free surface elevation at time step n,
- N is the total number of time steps,
- k represents the discrete frequency index.

In the numerical implementation, this is computed using the Fast Fourier Transform (FFT), which efficiently computes the DFT using an optimized algorithm. The discrete frequency components corresponding to the FFT output are:

$$f_k = \frac{k}{N\Delta t}, \quad k = 0, 1, 2, \dots, \frac{N}{2} \tag{C.2}$$

where Δt is the time step between measurements. Only the positive frequency components are considered in the spectral analysis.

The Power Spectral Density (PSD), which represents the energy distribution over frequencies, is given by:

$$S(f_k) = \frac{|\hat{\eta}(f_k)|^2}{N} \tag{C.3}$$

where $|\hat{\eta}(f_k)|$ is the magnitude of the FFT output. The dominant frequency is then equal to the highest peak in the spectrum:

$$f_{\text{dom}} = \arg\max S(f) \tag{C.4}$$

The corresponding normalized amplitude A of the dominant wave frequency f_{dom} is then given by:

$$a = \frac{2|\hat{\eta}(f_{\text{dom}})|}{N} \tag{C.5}$$

And the corresponding phase θ is computed as:

$$\theta = \arg(\hat{\eta}(f_{\text{dom}})) \tag{C.6}$$

Now that the amplitude and phase of the dominant frequencies at each gauge are determined. The incident wave amplitude Z_i and reflected wave amplitude Z_r can be computed using the known wave number k, gauge amplitudes a_1, a_2, a_3, and the known distances dx_{12} and dx_{13} between the measurement points following the method of Mansard and Funke (1980):

The auxiliary variables β and γ are defined as:

$$\beta = dx_{12} \cdot k, \quad \gamma = dx_{13} \cdot k \tag{C.7}$$

Using these, the denominator term D_k is computed as:

$$D_k = 2\left(\sin^2\beta + \sin^2\gamma + \sin^2(\gamma - \beta)\right) \tag{C.8}$$

The reflection-related terms are:

$$R_{1k} = \sin^2\beta + \sin^2\gamma, \quad R_{2k} = \sin\gamma\sin(\gamma - \beta), \quad R_{3k} = -\sin\beta\sin(\gamma - \beta) \tag{C.9}$$

The quadrature terms for phase adjustments are:

$$Q_{1k} = \sin\beta\cos\beta + \sin\gamma\cos\gamma \tag{C.10}$$

$$Q_{2k} = \sin\gamma\cos(\gamma - \beta) - 2\sin\beta \tag{C.11}$$

$$Q_{3k} = \sin\beta\cos(\gamma - \beta) - 2\sin\gamma \tag{C.12}$$

Using these relationships, the incident wave amplitude Z_i and reflected wave amplitude Z_r are computed as:

$$Z_i = \frac{a_1(R_{1k} + iQ_{1k}) + a_2(R_{2k} + iQ_{2k}) + a_3(R_{3k} + iQ_{3k})}{D_k} \tag{C.13}$$

$$Z_r = \frac{a_1(R_{1k} - iQ_{1k}) + a_2(R_{2k} - iQ_{2k}) + a_3(R_{3k} - iQ_{3k})}{D_k} \tag{C.14}$$

These can than be used to calculate the reflection coefficient as shown in Formula 3.7.

D
Formulas

D.1. Wavelength and Wavespeed

The wavelength (λ) and wavespeed (c) are found using the dispersion relation:

$$\omega^2 = gk \tanh(kh) \tag{D.1}$$

This equation must be solved iteratively for k . Once k is found, the wavelength and wavespeed are respectively:

$$\lambda = \frac{2\pi}{k} \tag{D.2}$$

$$c = \frac{\lambda}{T} \tag{D.3}$$

where:

- $\omega = \frac{2\pi}{T}$ is the angular frequency [rad/s],
- $k = \frac{2\pi}{\lambda}$ is the wavenumber,
- h is the water depth [m]
- T is the wave period [s]

D.2. Statistical Indicators

To evaluate the accuracy and reliability of the numerical model, a set of statistical indicators is used to compare the simulated results against reference or experimental data. These indicators provide insight into the model's systematic error, agreement, correlation, and overall predictive skill.

- **Bias**. Measures the average difference between predicted and observed values, indicating whether the model systematically overpredicts or underpredicts:

$$\text{Bias} = \overline{y} - \overline{x} \tag{D.4}$$

- **Index of Agreement (I_a)**. Quantifies how well the predicted results match the observed data. It ranges from 0 (no agreement) to 1 (perfect agreement):

$$I_a = 1 - \frac{\sum_{i=1}^{N}(x_i - y_i)^2}{\sum_{i=1}^{N}(|x_i - \overline{x}|+|y_i - \overline{y}|)^2} \tag{D.5}$$

- **Coefficient of Determination (R^2)**. Measures the strength of the linear correlation between predicted and observed values. It ranges from 0 (no correlation) to 1 (perfect linear relationship):

$$R^2 = \frac{\sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y})^2}{\sum_{i=1}^{N}(x_i - \overline{x})^2 \sum_{i=1}^{N}(y_i - \overline{y})^2} \tag{D.6}$$

- **Scatter Index (S_I).** Provides a normalized measure of the root mean square error, useful for comparing model performance across datasets with different magnitudes:

$$S_I = \frac{1}{\sqrt{N \sum_{i=1}^{N} (y_i - x_i)^2}} \tag{D.7}$$

Where:

- x_i, y_i = predicted and observed (or simulated) values (m)
- $\overline{x}, \overline{y}$ = average of predicted and observed values (m)
- N = number of data points