Politecnico di Torino

Master's Degree in Aerospace Engineering

Academic Year 2024/2025

July 2025

Numerical Reproduction of Regular and Irregular Free-Surface Waves using SPH: from Setup to Calibration

Supervisors:

Bruno Paduano
Oronzo Dell'Edera
Giovanni Bracco
Bonaventura Tagliafierro

Candidate:

Rita Nunziata

Abstract

Accurate modeling of free-surface waves is crucial for simulating realistic ocean conditions and for analyzing wave-structure interactions. However, replicating the wave profile at target locations, especially for irregular or steep wave groups, remains a complex and technically demanding task.

The current work focuses on the numerical reproduction of free-surface waves, both regular and irregular, using the Smoothed Particle Hydrodynamics method. This Lagrangian, meshless approach is implemented with the open-source solver DualSPHysics, widely used for free-surface flows in coastal and ocean engineering applications.

After a theoretical overview of the method and its properties, a two-dimensional numerical wave tank was generated and optimized for wave generation and absorption. A parametric study was conducted to investigate the influence of key simulation parameters - such as particle spacing, smoothing length, and artificial speed of sound - on numerical stability and fidelity.

To improve control over wave reproduction, an iterative calibration process was implemented. The simulated wave height and phase at a control location were compared with a theoretical target, and the input signal was adjusted accordingly until convergence was reached. The procedure was initially validated on regular waves and later extended to the reproduction of irregular waves with a predefined spectrum.

The proposed approach, inspired by grid-based CFD strategies, was successfully adapted to the Lagrangian nature of SPH. The results confirm the method capability to accurately reproduce wave dynamics, making it a reliable tool for wave generation studies in free-surface flow contexts.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Free surface waves are a type of wave generated at the interface between two media and governed by restoring forces, such as gravity and surface tension. This natural phenomenon is widely observed and of great interest in many engineering contexts and hydrodynamic problems: automotive, aerospace and aeronautics, hydraulics, industrial processes, fuel transport, nuclear engineering, coastal and marine engineering [1].

The latter specifically focuses on ocean waves, a type of free surface waves that occur at the interface between air and water. Their behaviour, primarily driven by wind, is subjected to a wide range of physical processes, such as refraction, diffraction, shoaling and breaking. As such, they represent a complex free-surface phenomenon governed by nonlinear hydrodynamic equations. Over the years, many methods have been proposed to address this problem. Among the earliest, in the 1980s and 1990s, were the Volume of Fluid and the Level Set Methods [2], used to track interfaces in multiphase flows using mesh-based solvers.

Traditional mesh-based methods, which are widely used in this context, often face challenges in handling large deformations and complex fluid–structure interactions. This issues can lead to mesh distortion and high computational costs due to the need for remeshing strategies [1]. For this reason, particle methods are becoming increasingly popular for simulating complex free surface flows, as they do not require a mesh connecting computational nodes. They are based on particles tracked in their motion, whose properties evolve according to the governing equations. This allows to manage complex free surfaces and large deformations [3]. Among these methods, the meshless Smoothed Particle Hydrodynamics stands out thanks to its Lagrangian nature, which has proven to be extremely effective in simulating complex hydrodynamic phenomena and describing wave motion [4].

1.1 Challenge

A proper definition of wave characteristics is essential in ocean engineering, especially for offshore structures, vessels and coastal installations, to ensure their safety and prevent accidents. Especially in advanced engineering scenarios, it is often necessary to generate a controlled wave event, that is a specific target wave shape occurring at a defined point in the computational domain [5].

However, studying wave propagation in water can be challenging, due to nonlinearities, surface deformations, wave breaking. Additionally, waves are subjected to viscous dissipation and energy loss, which cause amplitude decay along the domain [6]. Excessive numerical dissipation poses a significant issue in modeling free-surface waves, as it results in amplitude decay within just a few wavelengths from the point of generation, limiting physical reproducibility. In this regard, SPH is a promising method, since it does not require an explicit treatment of the free surface particles, avoiding to introduce numerical diffusion in the interface tracking. However, the dissipation mechanism of gravity waves must still be addressed for simulations involving viscous fluids [3].

Linear theory predicts that in an inviscid fluid, a wave should propagate indefinitely without amplitude change, so that total energy should be conserved. SPH, although wave generation is based on linear theory, does not preserve energy due to the presence of the numerical viscosity term in the momentum equation. As a result, wave amplitude decreases over time [3].

Many studies have focused on the dissipation and attenuation of free surface waves for SPH models, and various improvements and variants have been proposed, involving the study of parameters such as spatial resolution, the Reynolds number, the smoothing length and the choice of the viscous dissipation term in the SPH momentum equation [6, 3, 7].

Another important aspect in achieving accurate wave reproduction, which depends directly on the generation process and the proper definition of the input signal. In addition to the already mentioned linear theory, several approaches have been proposed in the literature based on higher-order corrections and nonlinearities of the wave group, as well as iterative techniques to compute both input phases and amplitudes [5, 8].

1.2 Contribution

In this study, the focus is on the accurate reproduction of a wave train with predefined characteristics, and its generation at a specific point in the computational domain. To address the issue of wave decay from the generation point to the point of interest, an iterative calibration method has been implemented. This process adjusts the input signal at the wave generator (both in phases and amplitudes) to produce an output that matches the target waveform as closely as possible. Unlike other approaches, this method uses the linearized spectrum of the target instead of the full target spectrum, intentionally excluding nonlinear components in order to test the method's ability to naturally generate them. This approach has already been validated in various CFD contexts [5] where Reynolds–averaged Navier–Stokes (RANS) solvers were successfully used to simulate focused waves.

In this work, the methodology is adapted and applied within the meshless Smoothed Particle Hydrodynamics framework. Specifically, the DualSPHysics code is employed, which represents one of the most advanced open-source environments for applying SPH to maritime engineering.

Following a brief overview of the main wave modeling approaches, Chapter 2 focuses on high-fidelity models, among which meshless methods stand out as the most versatile and suitable for representing free surface phenomena. In this study, the choice of the Smoothed Particle Hydrodynamics method was deemed optimal for its ability to accurately capture nonlinear behaviors, its flexibility in handling complex geometries, and its robustness in dynamic boundary conditions.

Chapter 3 explores the generation and calibration procedure for wave motion implemented in DualSPHysics, based on linear wave theory, through which it becomes possible to adjust the input signal at the wavemaker to obtain, at a fixed point in the domain, a wave with a spectrum that closely matches the desired target. Both regular and irregular wave cases are analyzed. Particular attention is given to the Biesel transfer function, through which free surface motion is converted into the motion of the wavemaker.

The numerical model built in DualSPHysics was presented in Chapter 4. First, the description of the numerical domain - a numerical wave tank - is introduced, parameterized according to the target wave dimensions. Following, a parametric study is conducted, assessing the model's performance and response to variations in key wave modeling parameters, in order to validate the model ability to accurately reproduce specific wave signals under different numerical conditions. Ultimately, the optimal values obtained from this study were selected for subsequent simulations.

Chapter 5 presents the results of the iterative calibration procedure, first applied to regular waves as a benchmark and then extended to irregular waves. In particular, the reproduction of an experimental signal is tested, starting solely from its linear components, assessing whether the full spectral content can emerge through the natural evolution of the SPH model.

Chapter 2

State of the Art

Due to this inherent complexity, free-surface wave modeling has become a multidisciplinary field. Over time, various approaches have been developed, each with specific strengths and limitations [9]. For instance, analytical and empirical models offer simplified representations of wave behavior. The former relies on simplified equations, often linear or weakly nonlinear, and are mainly used for theoretical insight or basic predictions. The latter derive from experimental data and provides practical formulas for engineering use, though they are only reliable within specific conditions. Physical models, on the other hand, involve scaled laboratory reproductions of real systems, offering high-fidelity but limited flexibility and relatively high cost. Numerical models overcome many of these limitations: they can simulate complex, nonlinear wave phenomena over arbitrary geometries and boundary conditions, making them the preferred approach for accurate and versatile wave modeling.

Before introducing high-fidelity schemes, it is useful to briefly recall the main categories of numerical wave models, each suited to different wave conditions [9]:

- Spectral Models (e.g., WAM, SWAN): efficient for large-scale forecasts, they resolve energy across frequencies and directions. However, since they do not operate in time domain, they discard phase information, limiting their accuracy on local effects such as wave diffraction or nearshore transformations [10].

- Boussinesq Models: based on depth-averaged equations, they include weak nonlinearity and dispersion, making them suitable for nearshore applications where shoaling, refraction, and wave breaking occur [11, 12].

- Potential Flow Models: solve the Laplace equation under inviscid, irrotational assumptions, and are often implemented using Boundary Element Methods (BEM) [13].

- Navier–Stokes Models: the most detailed, solving fully viscous flow with turbulence. They can potentially capture breaking, air entrainment, and structure interaction, at the cost of a high computational demand [9].

High-fidelity hydrodynamic models are generally based on the Navier–Stokes equations and are essential when nonlinear effects dominate and fine-scale accuracy is required.

2.1 High-Fidelity Numerical Models

A high-fidelity numerical model is characterized by the resolution of the fundamental dynamics equations, the use of fine spatial and temporal discretizations, the ability to include multiple and coupled phenomena, and the capacity to handle dynamic boundary conditions and complex domains.

In the context of hydrodynamic modeling, Computational Fluid Dynamics (CFD) consists of high-fidelity models based on the numerical resolution of the Navier–Stokes equations, widely used in engineering applications for their robustness and accuracy. Traditional CFD methods, such as Finite Volume Method, Finite Element Method and Finite Difference Method adopt an Eulerian approach [14], where the fluid domain is discretized on a mesh with fixed connectivity and flow variables are calculated at spatial locations that are stationary relative to the mesh. In this case, the validity of simulations strongly depends on the ability of the chosen method to capture localized phenomena and handle deformable or dynamic domains.

However, this approach can become unstable in the presence of large deformations, complex geometries or moving discontinuities. As a matter of fact, mesh-based methods face significant challenges when solid surfaces experience structural deformations or rigid bodies have relative motion with contacts. The required adaptation and remeshing techniques – such as dynamic mesh generation and deconstruction, penalty methods, augmented Lagrangian formulations, or immersed boundary conditions – are associated with the risk of element distortion and often come with higher computational cost [15, 16, 1]. In this context, meshless methods – particularly Smoothed Particle Hydrodynamics (SPH) – offer a promising alternative for building more flexible high-fidelity models [17, 18].

2.2 Meshless Numerical Methods

Meshless methods (or meshfree methods) have been developed precisely to overcome these obstacles. Instead of relying on a discretization based on elements with fixed connectivity, they build numerical approximations from a set of nodes distributed within the domain, without requiring an explicit mesh. The key concept is to represent physical fields (e.g., pressure or velocity) through weighted combinations of basis functions centered at the nodes, according to the general interpolation formula:

$$u(\mathbf{x}) \approx \sum_{i=1}^{N} \phi_i(\mathbf{x}) u_i, \tag{2.1}$$

where $\phi_i(\mathbf{x})$ are the shape functions associated with node i, N is the number of nodes and u_i represents the nodal value of the variable u. Unlike Finite Element Methods, in which ϕ_i depends on the geometry of the element, in meshless methods these functions are built based on the local distribution of nodes and supported weight functions [16].

Over the years, several meshless methods have been developed, each with specific characteristics:

- Smoothed Particle Hydrodynamics (SPH): a Lagrangian method originally developed for astrophysical problems, widely used in engineering and scientific applications. It uses kernel functions to interpolate field variables and is well-suited for problems involving large deformations, free surfaces, and fragmentation [1]

- Moving Particle Semi-implicit (MPS): uses simplified local averaging operators instead of kernel gradients. It has been widely adopted in hydraulics, coastal/ocean engineering, and nuclear safety applications [19].

- Element-Free Galerkin (EFG): a meshless extension of the Galerkin method that uses Moving Least Squares shape functions. It is particularly effective for problems requiring higher-order continuity and has shown robust results in fracture mechanics and large deformation simulations [20];

- Reproducing Kernel Particle Method (RKPM): a Galerkin meshless method that extends SPH by using corrected kernel functions to ensure consistency. It is mainly effective for problems involving fractures and large deformations[20];

- Meshless Local Petrov-Galerkin (MLPG): based on a local weak form applied over small subdomains surrounding each node. It offers great flexibility in choosing test functions enables easier treatment of boundary conditions. It has been applied successfully in problems ranging from elastostatics to free surface fluid dynamics [15].

These various meshless methods share the goal of overcoming the limitations of discretization, but they differ in numerical formulation, interpolation construction, and integration strategy.

For example, one of the most common approaches to constructing shape functions is the Moving Least Squares method [16], used in methods like EFG, RKPM and MLPG. The principle is to minimize a functional that measures the difference between nodal values and the field approximation:

$$J = \sum_{i=1}^{N} w_i(\mathbf{x}) \left[p^T(\mathbf{x})\mathbf{a}(\mathbf{x}) - u_i \right]^2 , \tag{2.2}$$

where $w_i(\mathbf{x})$ is a weight function centered at node i, $p^T(\mathbf{x})$ is a vector of monomials (e.g., $p^T = [1, x, y]$ for a linear basis in 2D), $\mathbf{a}(\mathbf{x})$ are the coefficients to be determined.

The optimal solution provides a continuous and differentiable approximation of the field, with consistency properties that can be controlled by choosing the polynomial basis.

In contrast, other formulations such as SPH use a kernel-based approximation that appears simpler and more intuitive, even if less accurate in terms of polynomial consistency. These differences reflect a trade-off between computational efficiency and implementation complexity [16].

Additionally, most meshless methods adopt a Lagrangian approach, in which the computational nodes (or particles) follow the motion of the fluid. This enables accurate representation of transport, deformation, and interfaces [21, 20].

From a computational point of view, the implementation of meshless methods does present some challenges [16]. In particular:

- the computation of shape functions and their derivatives requires operations on local support domains;

- numerical integration (especially in weak formulations) requires more advanced techniques than grid-based methods;

- the imposition of boundary conditions is not straightforward and requires special treatments, such as penalty methods or modified collocation.

Despite these difficulties, meshless methods have demonstrated excellent effectiveness in simulating free-surface flows, problems involving large deformations, and fluid–structure interactions. Their flexibility makes them ideal candidates for the numerical modeling of waves in coastal and offshore environments, where geometry is often irregular and constantly evolving [21, 19].

Among the most widely used meshless methods today, Smoothed Particle Hydrodynamics holds a leading position, thanks to its conceptual simplicity, purely Lagrangian formulation, and adaptability to highly nonlinear phenomena.

2.3 Smoothed Particle Hydrodynamics

The Smoothed Particle Hydrodynamics method, or SPH, is a meshfree numerical technique based on a Lagrangian formulation, in which the motion of fluid particles is tracked over time. Originally developed for astrophysical applications [22, 23], it has rapidly spread to various fields, including free-surface flow modeling, coastal fluid dynamics, wave-structure interaction, multiphase phenomena and many other engineering applications. The Lagrangian approach allows the motion of each fluid particle to be tracked over time, eliminating the need for a fixed spatial grid and enabling natural adaptation to deformable geometries and domains.

2.3.1 SPH Method Formulation

The central idea of the SPH method is to represent a continuous field as a weighted sum of discrete contributions from neighboring particles. Each particle carries physical information such as mass, velocity, pressure and density, and interacts with others within a defined influence radius.

The method starts from the integral representation of a generic function $f(\mathbf{x})$:

$$f(\mathbf{x}) = \int_{\Omega} f(\mathbf{x}')\delta(\mathbf{x} - \mathbf{x}')\, d\Omega, \tag{2.3}$$

where Ω is the domain in which $\mathbf{x}$ is defined, and $\delta(\mathbf{x} - \mathbf{x}')$ is the Dirac delta function:

$$\delta(\mathbf{x} - \mathbf{x}') = \begin{cases} 1, & \mathbf{x} = \mathbf{x}' \\ 0 & \mathbf{x} \neq \mathbf{x}' \end{cases} . \tag{2.4}$$

In SPH, the representation of a generic function $f(\mathbf{x})$ in a continuous domain is approximated by a convolution with a smoothing function (or *weighting function*) called a *kernel* W, defined as follows [17, 18]:

$$\langle f(\mathbf{x}) \rangle = \int_{\Omega} f(\mathbf{x}')W(\mathbf{x} - \mathbf{x}', h)\, d\mathbf{x}', \tag{2.5}$$

where:

- $W(\mathbf{x} - \mathbf{x}', h)$ is the kernel function, defined over a compact support controlled by the parameter h,
- h is the smoothing length, which controls the extent of the kernel support.

For numerical implementation, the integral is approximated by a discrete sum over neighboring particles [17]:

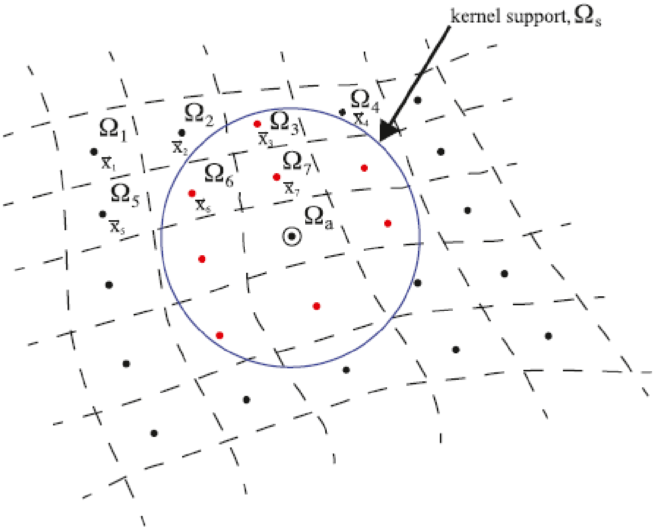


Figure 2.1: Schematic of the SPH discretization with a compactly supported kernel function [24].

$$f_a \approx \sum_b m_b \frac{f_b}{\rho_b} W(\mathbf{r}_a - \mathbf{r}_b, h), \tag{2.6}$$

where index a refers to the observation particle, and b ranges over particles within the kernel support. This approximation can be interpreted as a simple quadrature, where each particle represents an infinitesimal volume $V_b = m_b/\rho_b$, with the advantage of preserving the Lagrangian formulation [16].

Kernel Function Properties To be used effectively, the kernel W must satisfy several fundamental properties [16]:

- Non-negativity: $W(\mathbf{x} - \mathbf{x}', h) \geq 0$;
- Compact support: $W(\mathbf{x} - \mathbf{x}', h) = 0$ for $\|\mathbf{x} - \mathbf{x}'\| > kh$;
- Normalization: $\int_\Omega W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' = 1$;
- Symmetry: $W(\mathbf{x} - \mathbf{x}', h) = W(\mathbf{x}' - \mathbf{x}, h)$;
- Delta function limit: $\lim_{h\to 0} W(\mathbf{x} - \mathbf{x}', h) = \delta(\mathbf{x} - \mathbf{x}')$.

The choice of the kernel significantly affects both accuracy and numerical stability; the accuracy of the SPH discretization also depends on the degree of consistency of the formulation. Some versions of the kernel and the discrete sum are specifically designed to ensure higher consistency, especially near boundaries or in the presence of strong gradients. Among the most commonly used examples are:

- Cubic spline:

$$W(q) = \frac{\sigma}{h^d} \begin{cases} 1 - \frac{3}{2}q^2 + \frac{3}{4}q^3, & 0 \leq q < 1 \\ \frac{1}{4}(2-q)^3, & 1 \leq q < 2 \\ 0, & q \geq 2 \end{cases},$$

- Wendland C2

$$W(q) = \frac{\sigma}{h^d}(1-q)^4(1+4q), \quad 0 \leq q \leq 1,$$

where $q = \|\mathbf{x}-\mathbf{x}'\|/h$, σ is a normalization coefficient, and d is the spatial dimension.

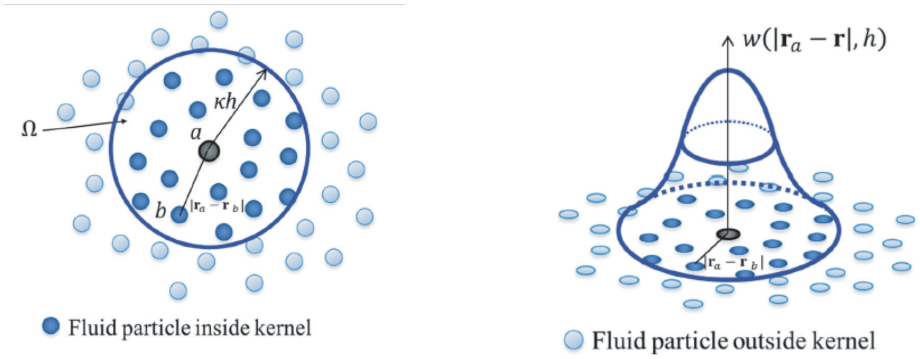

Ω
κh
a
b
|r_a − r_b|
w(|r_a − r|, h)
|r_a − r_b|
● Fluid particle inside kernel
○ Fluid particle outside kernel

Figure 2.2: Example of a 2-D particle distribution surrounding particle a. The radius of influence of the kernel is expressed as a multiple, k, of the smoothing length, h [17]

Kernel Approximation of Derivatives Starting from the continuous representation of a function $f(\mathbf{x})$ (2.3), one can derive a formulation for its gradient based on the kernel [18]. The integral identity is considered:

$$\nabla f(\mathbf{x}) = \int_\Omega f(\mathbf{x}')\nabla\delta(\mathbf{x}-\mathbf{x}')\,d\mathbf{x}', \tag{2.7}$$

which, like the function itself (2.5), is regularized by replacing the Dirac delta δ with a kernel function W:

$$\nabla f(\mathbf{x}) \approx \int_\Omega f(\mathbf{x}')\nabla W(\mathbf{x}-\mathbf{x}', h)\,d\mathbf{x}'. \tag{2.8}$$

This formulation highlights a fundamental aspect of the method: the differential operator acts on the kernel rather than on the function, simplifying the approximation over complex domains. By discretizing the integral as a sum over neighboring particles, an initial approximation is obtained:

$$\nabla f_a \approx \sum_b m_b \frac{f_b}{\rho_b} \nabla W_{ab}, \tag{2.9}$$

where index a identifies the observed particle, and b those within the kernel support.

However, this formulation can lead to errors in the conservation of momentum, as it is not symmetric [18]. To improve this property and ensure better conservation of physical quantities, a symmetric version of the gradient is preferred:

$$\nabla f_a = \sum_b m_b \left(\frac{f_b - f_a}{\rho_b} \right) \nabla W_{ab}. \tag{2.10}$$

This form ensures that the contribution from each particle pair is balanced, contributing to numerical stability and the correct conservation of fundamental physical laws [18].

2.3.2 Governing Equations in the SPH Formulation

The fundamental equations of fluid dynamics are here expressed in the Lagrangian form of the Navier–Stokes equations. The continuity equation (or mass conservation) is as follows:

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \mathbf{v}, \tag{2.11}$$

while the momentum equation is:

$$\frac{d\mathbf{v}}{dt} = -\frac{1}{\rho} \nabla P + \nu \nabla^2 \mathbf{v} + \mathbf{f}. \tag{2.12}$$

where ρ is the density, $\mathbf{v}$ is the velocity vector, P is the pressure and $\mathbf{f}$ is the acceleration due to external forces. Here, d refers to the material derivative.

In SPH, these equations are rewritten and discretized based on the particle representation [17], as shown below:

- Continuity equation:

$$\frac{d\rho_a}{dt} = \rho_a \sum_b \frac{m_b}{\rho_b} \mathbf{v}_{ab} \cdot \nabla W_{ab} + D_a. \tag{2.13}$$

- Momentum equation:

$$\frac{d\mathbf{v}_a}{dt} = -\sum_b m_b \left(\frac{P_a}{\rho_a^2} + \frac{P_b}{\rho_b^2} \right) \nabla W_{ab} + \Gamma_a + \mathbf{f}. \tag{2.14}$$

where $\mathbf{v}_{ab} = \mathbf{v}_a - \mathbf{v}_b$; P_k, ρ_k and $\mathbf{v}_k$ are respectively pressure, density and velocity evaluated for the particle k. In the momentum equation the term Γ_a represents the dissipation term, while in the continuity equation D_a is the numerical density diffusion term.

To account for viscous dissipation, a laminar viscosity model is commonly used for Γ_a [25]:

$$\nu_0 \nabla^2 \mathbf{v}_a = \sum_b m_b \frac{4\nu_0}{\rho_a + \rho_b} \frac{\mathbf{v}_{ab} \cdot \mathbf{r}_{ab}}{r_{ab}^2 + \eta^2} \nabla W_{ab}, \tag{2.15}$$

where $r_{ab} = r_a - r_b$,ν_0 is the kinematic viscosity (typically 10^{-6} m^2) and $\eta = \epsilon \cdot h^2$, with $\epsilon = 0.01$ being a small regularization parameter to avoid singularities in the formulation.

As an alternative – especially in the presence of discontinuities or high-velocity gradients – an artificial viscosity term Π_{ab} is added to the pressure term [26]. It does not represent a physical viscosity, but is included to prevent numerical instabilities.

$$\Pi_{ab} = \begin{cases} \frac{-\alpha c_{ab} \mu_{ab}}{\rho_{ab}}, & \text{if } \mathbf{v}_{ab} \cdot \mathbf{r}_{ab} < 0 \\ 0, & \text{if } \mathbf{v}_{ab} \cdot \mathbf{r}_{ab} \geq 0 \end{cases}, \tag{2.16}$$

with:

$$\mu_{ab} = \frac{h \mathbf{v}_{ab} \cdot \mathbf{r}_{ab}}{\mathbf{r}_{ab}^2 + \epsilon h^2}. \tag{2.17}$$

Here, α is an empirical parameter (typically between 0.01 and 0.1) to introduce proper dissipation and $c_{ab} = 0.5\,(c_a + c_b)$ is the mean speed sound between a and b.

To close the system, the pressure is calculated using a barotropic equation of state [26]:

$$P = \frac{c_s^2 \rho_0}{\gamma} \left[\left(\frac{\rho}{\rho_0} \right)^\gamma - 1 \right], \tag{2.18}$$

where $c_s = \sqrt{\frac{\partial P}{\partial \rho}}$ is the numerical speed of sound (typically at least 10 times the characteristic speed of the problem) and γ is the polytropic exponent, often assumed to be 7 for weakly compressible flows – such as water.

This formulation, known as *WCSPH* (Weakly Compressible SPH), avoids the direct solution of a Poisson equation for pressure as in incompressible methods, maintaining the computational simplicity of the method.

2.3.3 Consistency of the SPH Method

One of the fundamental issues of the SPH method concerns the loss of consistency, that is the method's ability to exactly reproduce polynomial functions up to a certain order. In other words, it is the local truncation error introduced by the numerical discretization of the governing equations [27]. This property is essential to ensure the accuracy and convergence of numerical solutions. Consistency involves not only scalar interpolation but also the correct approximation of derivatives (such as gradient and divergence), which can deteriorate significantly near free surfaces or under conditions of strong deformation [24].

Unlike grid-based methods, for SPH, two levels of consistency are distinguished [27]: kernel approximation and particle approximation. The first refers to the continuous integral approximation of the function via the kernel and is independent of particle distribution. The second originates from particle discretization and is highly dependent on the number of particle neighbors (thus on the smoothing length, h) and their spatial distribution (i.e., particle spacing, dp).

First-order consistency of the kernel is linked to the satisfaction of moments conditions [24]:

$$M_l = \int_\Omega (\mathbf{x}' - \mathbf{x})^l W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' = 0 \quad \text{for } l \geq 1. \tag{2.19}$$

However, this property is not guaranteed in the discrete formulation, especially near boundaries or in the presence of irregular particle distributions – for example, due to kernel support truncation, particle contributions may be unbalanced (a fluid particle near a boundary will have fewer neighbors than those in the interior), generating errors in the estimation of derivatives and physical quantities. This phenomenon is known as boundary deficiency [24].

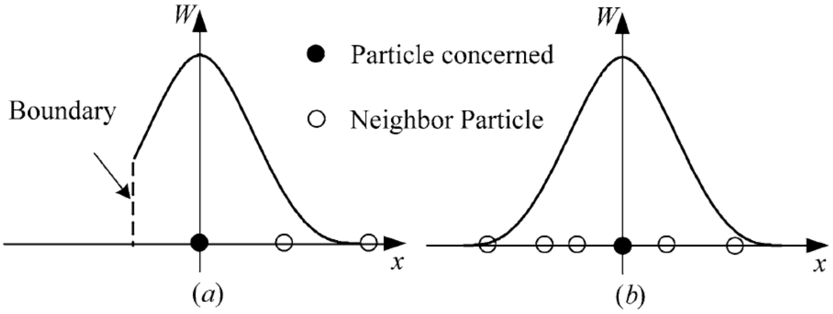


Figure 2.3: SPH particle approximations in one-dimensional cases [18]. (a) Particle approximation for a particle whose support domain is truncated by the boundary. (b) Particle approximation for a particle with irregular particle distribution in its support domain.

The loss of consistency due to particle discretization – known as particle inconsistency – arises from three main causes: truncation of the kernel support near physical boundaries, irregular particle distribution and variable smoothing length. This has a negative impact on the physical accuracy of the simulation, since it can compromise the conservation of mass, momentum, and energy [24]. However, Quinlan et al. [27] suggest that such errors decrease when h is reduced while keeping the ratio $\frac{dp}{h}$ constant, at a rate of h^2. Conversely, when $\frac{dp}{h}$ is decreased (i.e., the number of neighboring particles is increased) with h fixed, the error also decreases. Therefore, by adjusting these parameters, it is possible to restore particle consistency and ensure local conservation.

To mitigate consistency issues, several correction methods have been developed [24]. Among the most robust are those based on the Taylor series expansion of the kernel approximation and its spatial derivatives, such as the Corrective SPH [28].

It is important to note that consistency, while necessary to ensure the convergence of the numerical solution, is not sufficient by itself. Other factors, such as numerical stability and the treatment of boundary conditions, play equally crucial roles.

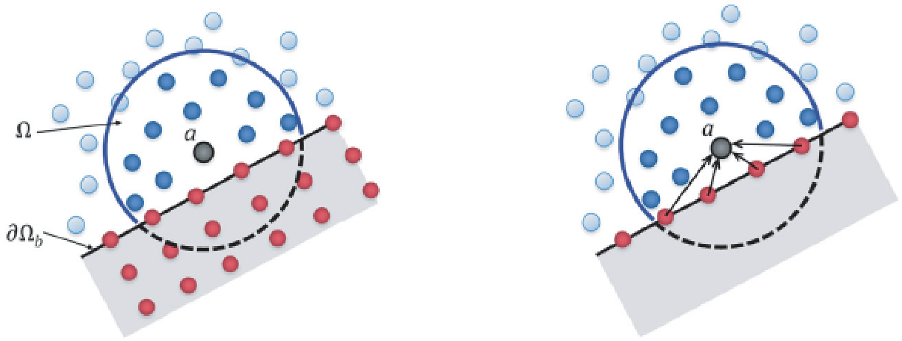
2.3.4 Boundary Conditions

Proper treatment of boundary conditions is fundamental to ensure numerical stability and simulation accuracy, especially in the presence of free surfaces or wave-structure interactions, as typically occurs in SPH simulations for coastal applications. Due to the absence of a fixed grid, the treatment of boundary conditions in the SPH method becomes extremely delicate. The fully Lagrangian formulation allows great geometric flexibility, but it also introduces boundary deficiency [24, 18].

Regarding free-surface treatment, SPH does not require explicit handling due to its Lagrangian formulation. However, the surface shape can only be resolved with a precision limited by the particle spacing [17].

For solid walls, the implementation of boundary conditions is carried out through various approaches. For instance, real boundary particles techniques [29] involve particles that are distributed along the edge of the domain. They may be static or dynamic (in case of moving walls) and interact directly with fluid particles. An alternative is given by the fictitious (or ghost) particles method [17], which generates artificial particles placed beyond the physical boundary, with properties defined to ensure symmetry (Figure 2.4a). They reflect or interpolate the properties of nearby fluid particles, contributing to proper kernel evaluation without altering the dynamics. Another example is the repulsive particles method [17]. This solution implements fixed particles that exert a central repulsive force to prevent fluid particles from penetrating the solid (Figure 2.4b). The force depends on distance and acts only within a certain radius, making it particularly useful in simulations of impact or wave-obstacle interaction.

Each approach has advantages and limitations: ghost particles are effective for rigid walls and symmetry conditions, while real boundary particles provide a more direct and physical description of boundaries [29]. The choice of the most appropriate technique depends on the type of problem, the geometry of the domain, and the presence of free surfaces or moving structures.



(a) Ghost particles.

(b) Repulsive particles.

Figure 2.4: SPH boundary treatments for rigid walls as particle a approaches the boundary $\partial\Omega_b$. The grey shaded area represents the solid boundary [17].

2.4 The DualSPHysics software

In recent years, the efficacy of the SPH formulation has been improved by the development of optimized numerical codes capable of handling realistic simulations. Among these, DualSPHysics [25] stands out as one of the most mature and widely used open-source implementations. Developed through collaboration between the University of Vigo, the University of Manchester, the Polytechnic University of Catalonia, the New Jersey Institute of Technology, The University of Edinburgh, the Instituto Superior Técnico in Lisbon, and the University of Parma, the code is designed to simulate large-scale fluid dynamics phenomena. Based on the WCSPH approach, it is optimized for GPU acceleration and enables efficient modeling of coastal dynamics.

In this regard, the main applications of the code include wave–structure interaction (e.g., dams, breakwaters, bridges), wave impact and overtopping phenomena, dynamics of floating bodies, wave propagation in channels or basins, analysis of wave energy conversion devices.

Architecture DualSPHysics features a modular architecture, structured into three main stages:

1. Pre-processing: domain setup and particle generation using the `GenCase` utility. Simulation parameters are defined through XML files, allowing detailed and reproducible control;

2. Core solver: execution of the SPH simulation, with support for multi-threaded CPU computation (OpenMP) and GPU (CUDA);

3. Post-processing: analysis and visualization of results using dedicated tools such as `MeasureTool`, and `FlowTool`.

The code is written in `C++` and highly parallelized and capable of managing simulations with tens of millions of particles in reasonable computational times.

2.4.1 Numerical Methods

In the WCSPH formulation, pressure is evaluated with an equation of state (2.18); viscous effects can be modeled using physical laminar viscosity (2.15) or alternatively Monaghan's artificial viscosity (2.16).

Different boundary conditions are implemented in DualSPHysics to manage fluid–solid interactions and domain behavior [30], namely Dynamic Boundary Condition (DBC), Modified Dynamic Boundary Condition (mDBC) and Open Boundary. In particular, mDBC [31] implements a set of boundary particles that satisfy the continuity equation just like the fluid particles, while their velocity is not computed using the momentum equation. Unlike the standard approach, however, the modified version includes an additional boundary interface between the fluid and the boundary particles, located at a distance of $\frac{dp}{2}$, where dp is the particle spacing, from the boundary itself. Across this interface, ghost particles are mirrored, as previously described in subsection 2.3.4.

Two explicit time integration schemes are available [25]. The velocity Verlet algorithm or, alternatively, the Symplectic Position Verlet scheme. The latter is implemented for improved stability and time-reversibility in conservative systems. It evolves the state of the particles as follows:

$$\mathbf{r}^{n+\frac{1}{2}} = \mathbf{r}^{n} + \frac{\Delta t}{2}\mathbf{v}^{n}, \tag{2.20}$$

$$\rho^{n+\frac{1}{2}} = \rho^{n} + \frac{\Delta t}{2}\frac{d\rho^{n}}{dt} \tag{2.21}$$

$$\mathbf{r}^{n+1} = \mathbf{r}^{n+\frac{1}{2}} + \frac{\Delta t}{2}\mathbf{v}^{n+1}, \tag{2.22}$$

$$\mathbf{v}^{n+1} = \mathbf{v}^{n+\frac{1}{2}} + \frac{\Delta t}{2}\frac{d\mathbf{v}^{n+\frac{1}{2}}}{dt}. \tag{2.23}$$

The time step Δt is computed according to the Courant–Friedrichs–Lewy (CFL) condition. In DualSPHysics, the condition is dynamically adjusted based on particle velocity, viscous effects, smoothing length h, and speed of sound c_s [30].

$$\Delta t = 0.25 \cdot \min(\Delta t_f, \Delta t_{cv}), \tag{2.24}$$

where:

$$\Delta t_f = \min_a \left(\sqrt{\frac{h}{|\mathbf{f}_a|}} \right) \quad \text{and} \quad \Delta t_{cv} = \min_a \left(\frac{h}{c_s + \max_b \left(\frac{4\nu_{ab}\tau_{ab}}{r_{ab}^2 + \eta^2} \right)} \right). \tag{2.25}$$

Here, $\mathbf{f}_a$ is the force per unit mass on particle a, and the second term in Δt_{cv} accounts for viscous diffusion between neighboring particles.

Particle Shifting Algorithm As discussed in subsection 2.3.3, particle distribution can often be irregular, leading to discretization errors. To address this issue, DualSPHysics includes a shifting algorithm proposed by Lind et al. [32], based on Fick's first law of diffusion, which relates the diffusion flux J to the concentration C gradient:

$$J = -D_F \nabla C, \tag{2.26}$$

where D_F is the Fickian diffusion coefficient. The shifting distance [25] is computed as:

$$\delta r_a = -D_s \nabla C_a. \tag{2.27}$$

Here

$$D_s = A_s h \|\mathbf{v}\| \Delta t, \tag{2.28}$$

where A_s is a dimensionless constant, raging between 1 and 6, with 2 proposed as default [25]. The kernel gradient of the particle concentration can be expressed as:

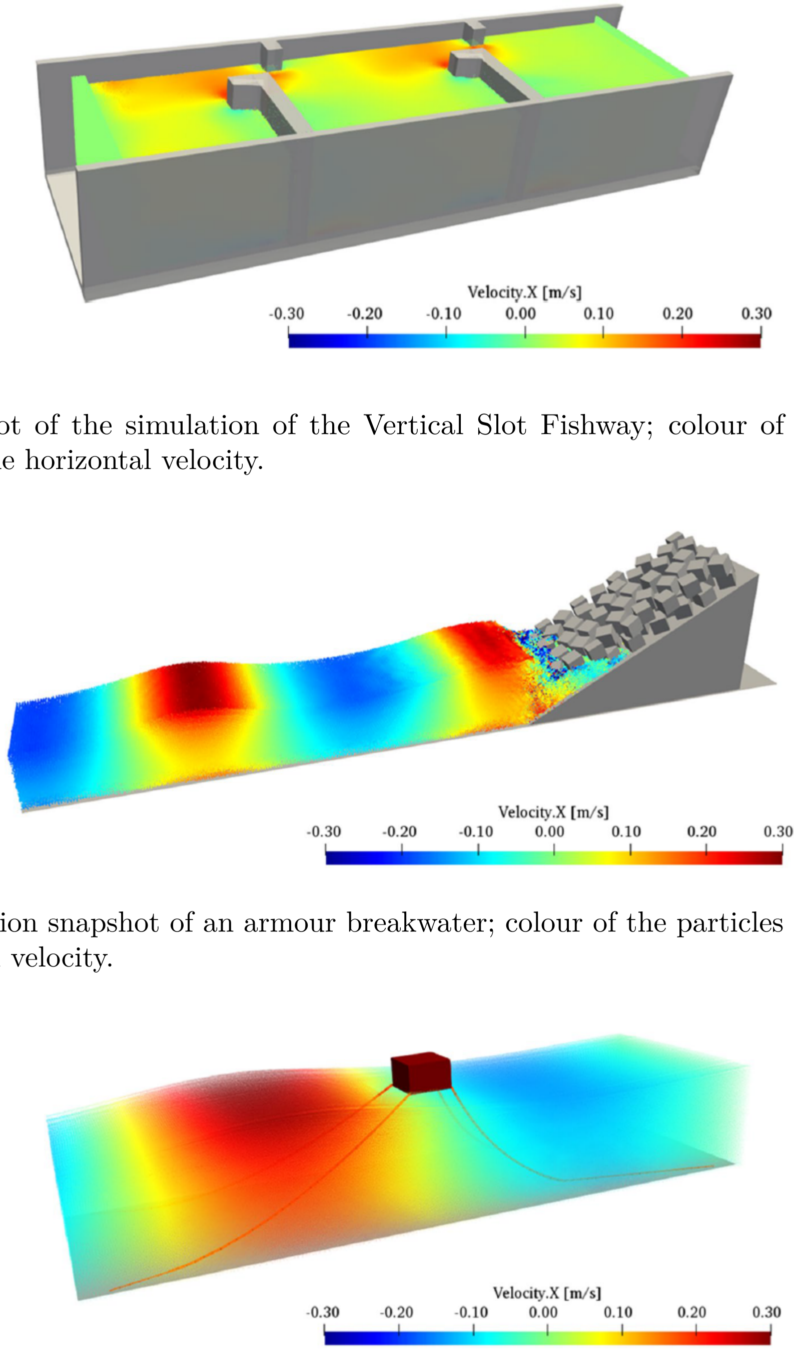
$$\nabla C_a = \sum_j \frac{m_j}{\rho_j} \nabla W_{aj} \tag{2.29}$$

2.4.2 Validation and Case Studies

The software DualSPHysics was validated by many reference works [30, 25], where the code was tested through numerous comparisons with physical experiments and other numerical models. Some of the most significant case studies include wave impact on rigid and moving walls, overtopping on coastal structures, oscillation of constrained floating bodies, tsunami propagation in laboratory tanks (Figure 2.5).

In all cases, DualSPHysics demonstrated good agreement with experimental data, both qualitatively (wave shape, flow behavior) and quantitatively (pressure, elevation, forces). Many of these tests are aligned with the SPHERIC benchmark suite, which provides standardized test problems for evaluating SPH solvers in terms of accuracy, stability, and computational performance [33].

The code, documentation, and test cases are available on the official website [34], which hosts an active international community of users and developers.



(a) Snapshot of the simulation of the Vertical Slot Fishway; colour of the particles represent the horizontal velocity.

(b) Simulation snapshot of an armour breakwater; colour of the particles represent the longitudinal velocity.

(c) Simulation snapshot of the moored box under regular waves; colour of the particles represent the horizontal velocity.

Figure 2.5: Significant case studies presented by Dominguez et al. [25].

Chapter 3

Wave Generation and Calibration in DualSPhysics

Accurate wave generation is essential for reproducing target conditions and ensuring that the simulated wave field matches theoretical or experimental expectations. This chapter focuses on the implementation and refinement of regular and irregular wave generation using a piston–type wavemaker in the DualSPHysics environment. The theoretical formulation of the target wave profile is first introduced, alongside the concepts of near field and far field, which are central to interpreting wave evolution along the domain.

However, despite the use of an idealized motion derived from linear theory, the actual wave profile generated by the piston may differ from the theoretical target due to a combination of numerical and physical effects, such as dissipation caused by numerical viscosity, partial reflections at the domain boundaries, limited spatial resolution. As a result, the wave height and phase observed in the far field may not match the prescribed values, even under nominally correct input conditions.

To address this issue, an iterative calibration procedure has been implemented in DualSPHysics, aimed at reproducing a wave shape that matches as close as possible the theoretical profile at a specific location within the numerical domain.

3.1 Wave Generation Theory

The definition of a target wave profile is a necessary prerequisite for assessing the accuracy of wave generation in numerical models. In this study, the reference signal is derived from linear wave theory, which provides an analytical description of wave propagation under simplifying assumptions.

3.1.1 Linear Wave Theory

Linear wave theory, also known as Airy wave theory [35], represents one of the earliest and most widely used analytical models for describing the two-dimensional propagation of surface gravity waves. This theory is based on the assumptions that the fluid is incompressible, inviscid and irrotational, and the wave height H is small relative to both the wavelength λ and the water depth h: $H/L \ll 1$, $H/h \ll 1$. Under these conditions, the wave free surface elevation is described by:

$$\eta(x,t) = a\cos(kx - \omega t), \tag{3.1}$$

where a is the wave amplitude, $\omega = \frac{2\pi}{T}$ is the angular frequency, $k = \frac{2\pi}{\lambda}$ is the wave number, T is the wave period, and λ is the wavelength.

The wave number k and angular frequency ω are linked by the linear dispersion relation:

$$\omega^2 = gk\tanh(kh), \tag{3.2}$$

which couples the wave propagation characteristics to the water depth h. In deep water conditions ($h \gg \lambda$), $\tanh(kh) \approx 1$, while in shallow water ($h \ll \lambda$), $\tanh(kh) \approx kh$. These assumptions yield the simplified dispersion relation valid under deep water conditions:

$$\lambda = \frac{gT^2}{2\pi}. \tag{3.3}$$

The speed at which the wave phase propagates, known as the phase velocity, is defined as:

$$c = \frac{\omega}{k} = \sqrt{\frac{g}{k}\tanh(kh)}. \tag{3.4}$$

Depending on the water depth, the expression simplifies to:

- Deep water: $c \approx \sqrt{\frac{g}{k}}$;
- Shallow water: $c \approx \sqrt{gh}$.

Fluid particles in linear wave theory follow elliptical trajectories that become increasingly flattened with depth. The horizontal and vertical components of particle velocity, u and w, are given by [35]:

$$u(x,z,t) = \frac{2\pi H}{T} \cdot \frac{\cosh(k(z+h))}{\sinh(kh)}\cos(kx - \omega t), \tag{3.5}$$

$$w(x,z,t) = \frac{2\pi H}{T} \cdot \frac{\sinh(k(z+h))}{\sinh(kh)}\sin(kx - \omega t), \tag{3.6}$$

where z is the vertical coordinate measured from the still water level (positive upwards), and h is the local water depth. These expressions describe how the particle motion decays exponentially with depth and are instrumental for understanding wave kinematics, particularly when implementing wave generation methods.

The accuracy of this wave description has been experimentally validated in large-scale wave tanks and serves as a standard reference for SPH validation campaigns [36].

3.1.2 Spectral Representation of Irregular Waves

In natural environments, wave motion is rarely regular. Instead, it results from the superposition of numerous harmonic components with varying frequencies, amplitudes, and directions (and phase). This complexity can be effectively captured through a spectral approach, where wave energy is distributed continuously over a range of frequencies [37].

In the linear spectral model, the free surface elevation is described as the sum of multiple sinusoidal components:

$$\eta(t) = \sum_{i=1}^{N} a_i \cos(\omega_i t + \phi_i), \tag{3.7}$$

where a_i is the amplitude, ω_i is the angular frequency, and ϕ_i is the phase of each component. The distribution of amplitudes a_i is governed by the wave energy spectrum $S(\omega)$, which represents the energy density per unit frequency:

$$a_i = \sqrt{2S(\omega_i)\Delta\omega}, \tag{3.8}$$

with $\Delta\omega$ representing the frequency discretisation interval.

Two commonly used spectra are the Pierson–Moskowitz and the JONSWAP spectra [37], reported here in their parametrized formulation as functions of the significant wave height, H_s, and the peak frequency, f_p:

1. Pierson–Moskowitz (PM) describes a fully developed sea:

$$S_\eta(f) = \frac{5}{16} H_s^2 f_p^4 f^{-5} \exp\left(-\frac{5}{4}\left(\frac{f_p}{f}\right)^4\right). \tag{3.9}$$

2. JONSWAP is suitable for young, developing seas with a sharper spectral peak enhanced by a factor $\gamma = 3.3$:

$$S_\eta(f) = \frac{1.4}{\gamma} \cdot \frac{5}{16} H_s^2 f_p^4 f^{-5} \gamma^\alpha \exp\left(-\frac{5}{4}\left(\frac{f_p}{f}\right)^4\right), \tag{3.10}$$

where:

$$\alpha = \exp\left(-\frac{(f - f_p)^2}{2\sigma_f^2 f_p^2}\right). \tag{3.11}$$

The spectral width σ_f is set to 0.10 when $f \leq f_p$, while it is set to 0.50 when $f > f_p$.

Each spectrum characterizes the sea state based on meteorological input, and the choice depends on whether the target sea is mature or still developing.

Validation of irregular wave simulations is typically based on spectral analysis of the simulated free surface elevation, which is then compared to the input wave spectrum. These aspects are critical in calibration procedures, especially when evaluating the fidelity of wave energy distribution in both time and frequency domains.

3.1.3 Near and Far Fields

The analytical description introduced earlier defines the target condition for the DualSPHysics simulations presented in this work. All input signals for wave generation, validation procedures, and error assessments are carried out with respect to this idealized form. However, due to both physical and computational (numerical dissipation) reasons, the wave profile generated near the boundary is not immediately identical to the theoretical solution. A transitional region, referred to as the near field [37], typically develops adjacent to the piston. In this region, the wave is still forming: it may exhibit non-periodic behavior, amplitude overshoots, or phase distortions caused by initial particle acceleration and local rearrangements.

As the wave propagates through the domain, it gradually stabilizes and converges toward the theoretical shape (Figure 3.1). This downstream area, where the wave exhibits periodicity and steady characteristics, is referred to as the far field [37]. It is in this region that comparisons with the theoretical target are most reliable, and where measurements of wave height and phase are typically carried out.

A more rigorous representation of the wave field near the wavemaker is given in Frigaard et al. [37], where the free surface elevation $\eta(x,t)$ is expressed as the sum of a propagating wave and a set of evanescent modes:

$$\eta(x,t) = A\cos(kx - \omega t) + \sum_{n=1}^{\infty} B_n e^{-\kappa_n x}\cos(\omega t), \tag{3.12}$$

where A is the amplitude of the primary wave, B_n are the modal coefficients of the evanescent components and κ_n are the spatial decay constants, increasing with n. The exponential terms $e^{-\kappa_n x}$ ensure rapid attenuation with distance from the generator.

This formulation shows that, very close to the wavemaker, the wave profile is significantly altered by non-propagating components that decay exponentially with x. The evanescent modes originate from the complex interaction between the moving boundary and the fluid column, and their effect becomes negligible approximately one wavelength (λ) away from the source.

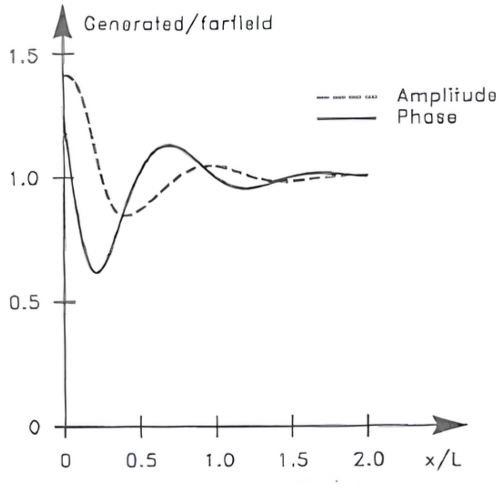


Figure 3.1: Wave amplitude and phase of the generated wave field relative to the far field solution by Frigaard et al. [37]. Water depth fixed at 0.7 m and wave period of 0.7 s.

Numerical and experimental evidence shows that a distance of 1-1.5 λ is typically required before the wave reaches a steady, periodic shape [38]. Beyond this point, the recorded wave signal corresponds predominantly to the propagating wave and aligns with theoretical expectations. In the present study, the far field is considered to begin at approximately $x = 1.5\lambda$ from the wave paddle, based on observations of waveform stabilization in the performed simulations (see subsection 4.2.1).

3.2 Implementation of Wave Generation in DualSPHysics

3.2.1 Overview of WaveMaker Types

In numerical wave tanks, wave generation is typically achieved by imposing a time-dependent boundary condition at one end of the domain. In the context of DualSPHysics, this is accomplished using a moving boundary element referred to as a wavemaker, which induces wave motion by displacing fluid particles.

Two common types of wavemakers are available [38]:

1. Piston-type wavemaker: horizontal translation of a vertical paddle, that displaces water uniformly along the vertical face;

2. Flap-type wavemaker: rotation of a paddle around a hinge located near the bed, simulating a vertical oscillation of the paddle tip.

The two types differ significantly in the resulting velocity field: the piston-type induces a nearly uniform horizontal velocity profile, while the flap-type leads to a more vertically varying field. Hence, the selection of the wavemaker type must consider the target wave conditions: flap-type systems better represent the vertical orbital motion typical of shallow water waves, as the induced particle velocity decreases with depth, while piston-type are more appropriate for deeper water due to their uniform velocity distribution.

An additional option is to impose the wave motion using the Relaxation Zone method (see subsection 4.3.1) [39].

3.2.2 Motion Signal Definition

In DualSPHyics, the approach used for defining the paddle motion is based on the Biesel transfer function, which relates the desired wave height and period to the displacement of the paddle through a geometry-dependent correction factor. According to Biesel's formulation [37], the displacement signal $s(t)$ of the wavemaker is expressed as:

$$s(t) = \frac{S_0}{2} \sin(\omega t + \phi), \tag{3.13}$$

specifically, the paddle stroke S_0 is obtained from

- Piston-type wavemaker (Figure 3.2):

$$\frac{H}{S_0} = \frac{2\sinh^2(kh)}{\sinh(kh)\cosh(kh) + kh}, \tag{3.14}$$

- Flap-type wavemaker (Figure 3.3):

$$\frac{H}{S_0} = \frac{2\sinh(kh)(1 - \cosh(kh) + kh\sinh(kh))}{kh(\sinh(kh)\cosh(kh) + kh)}. \tag{3.15}$$

For irregular waves, DualSPHysics generates the free surface elevation by discretizing the input energy spectrum $S(\omega)$ into a set of discrete harmonic components using the random-phase method. The continuous spectrum is sampled over a range of N angular frequencies ω_i, and each frequency is assigned an amplitude (according to equation 3.8) and a random phase.

The phases ϕ_i are uniformly distributed in $[0, 2\pi)$, allowing the reconstructed time series to reflect the irregular and random nature of real sea states. This approach follows the classical spectral reconstruction techniques [37], commonly adopted in both numerical and laboratory studies.

To ensure repeatability between simulations, the random number generator used for the phase assignment is initialized with a fixed seed.

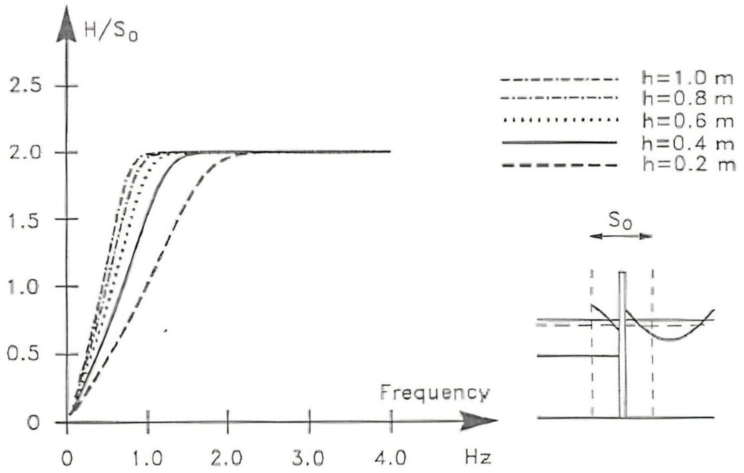


Figure 3.2: Biesel transfer function of piston-type wavemaker [37].

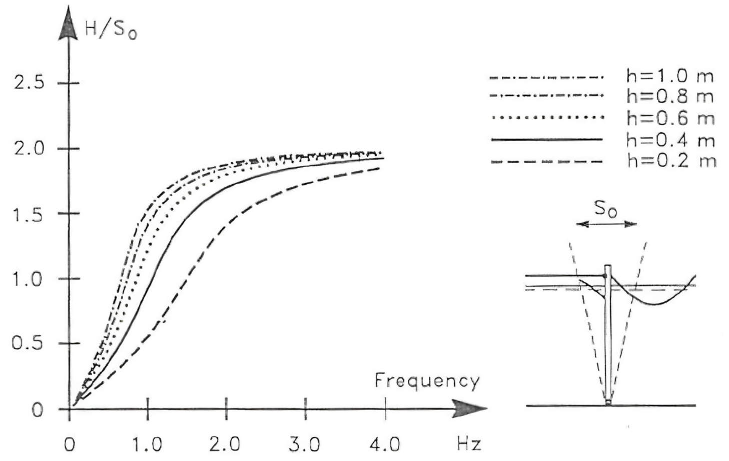


Figure 3.3: Biesel transfer function of flap-type wavemaker [37].

3.2.3 Signal Generation Options

DualSPHysics provides two main approaches for prescribing the motion of a wavemaker:

1. Internal signal generation using predefined wave parameters;
2. External input via a user-defined time series.

In the first case, the user specifies the target wave characteristics either through the graphical interface or directly in the XML configuration file:

- Regular waves: defined through parameters such as `<waveheight>`, `<period>`, `<phase>`;
- Irregular waves: spectrum–driven using `<spectrum="jonswap">` or `PM`, with input parameters like `<number of waves>` and `<seed>` for random phase reproducibility.

The software then automatically generates the paddle motion using Biesel's equation, which accounts for the wavemaker geometry and tank depth. This method is fast and convenient and is especially useful for simple, idealized test cases.

Alternatively, DualSPHysics allows define the paddling trajectory explicitly by importing a displacement time series from a text file (`.txt` or `.csv`). This input file typically contains four columns: time and displacement in each direction (e.g., x, y, z). It is referenced in the XML file through the `<motion>` tag associated with the moving boundary object. This method offers greater flexibility and control over the imposed motion, especially when advanced tuning or calibration procedures are required.

3.2.4 Implementation in this Study

For the simulations presented in this work, the piston–type wavemaker was selected, due to its simplicity and compatibility with two-dimensional setups. This type is particularly effective when generating regular wave trains for calibration and benchmarking purposes.

The wavemaker motion was defined using a user-generated external file, containing the time history of the piston displacement. The signal was computed in `MATLAB` using Biesel's formulation for regular and irregular wave generation, given the target wave shape. This approach was preferred over the internal auto-generation feature, as it provides direct access to the input signal and simplifies the implementation of iterative calibration strategies described later in this work. Moreover, the external file can be updated independently of the main simulation setup, allowing fast testing of alternative waveforms.

3.3 Wave Calibration

For industrial applications, RANS solvers combined with free surface capturing algorithms – such as the Volume of Fluid method – represent the standard approach for reproducing wave dynamics using CFD tools [5]. Recent research has focused on the numerical modeling of wave groups; however, inconsistency with experimental results are frequently reported: a major limitation lies in the absence of a methodology to correct the input signal in CFD models [5].

In fact, the most challenging aspect in accurately generating such wave groups is the proper selection of the amplitudes and phases of the spectral components at the inlet. Linear wave theory and empirical iterative processes have been suggested over the years in order to achieve the required phase and amplitude focusing. However, the efficiency of these methods reduces as the non-linearity of the wave group increases: as a result, the focal point is shifted in both space and time and the quality of focusing reduces considerably [8].

The methodology proposed by Stagonas et al. [8] for highly accurate focused wave generation represents a significant improvement over previous techniques, as it effectively addresses these issues by implementing an iterative correction. The main innovation compared to previous approaches lies in the use of linearized target spectra as the initial condition at the wave maker, rather than the full target spectrum.

In reality, the full wave spectrum includes not only the primary linear components but also second-order effects arising from nonlinear interactions between linear modes, as described by second-order wave theory (e.g., Stokes theory) [35]. These interactions generate additional frequency components in the spectrum:

- Subharmonic components, resulting from the difference between pairs of linear frequencies $(f_i - f_j)$, which appear in the low-frequency region of the spectrum;

- Superharmonic components, generated by the sum of linear frequencies $(f_i + f_j)$, observed at higher frequencies beyond the main spectral band.

The objective of the methodology presented in this study is to accurately reproduce the dynamics of a real wave starting solely from its linear components. A controlled wave generation procedure is adopted, inspierd by the previous approach [5, 8] but implemented within the SPH framework. The iterative correction process acts on the wavemaker transfer function, with the aim of precisely focusing the wave group of a prescribed spectrum.

3.3.1 Main contribution of this Study

The methodology uses amplitudes and phases of linearized spectral components as input parameters and applies linear theory to generate a desired wave group at specified time and position within the numerical wave tank. The structure is presented as follows and illustrated in Figure 3.4:

1. The target spectrum is defined and his linear components are selected;

2. The corresponding motion signal for the wavemaker is generated and the simulation is performed;

3. The wave elevation is evaluated at the target location and the linear output spectrum is extracted;

4. The linearized output spectrum is compared with the target spectrum and a corrected input spectrum is calculated;

5. The procedure is applied iteratively until convergence is reached in both phase and amplitude.

The practical implementation of the methodology is presented in the following sections.

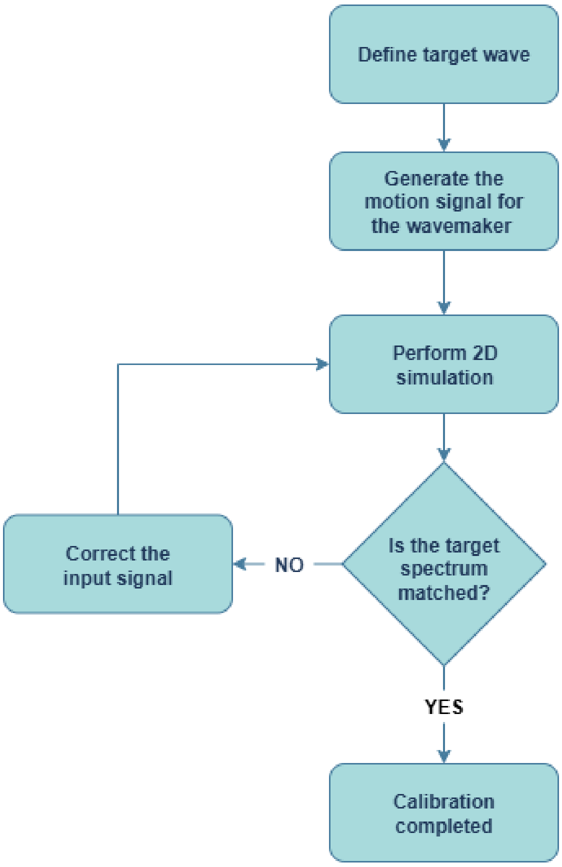


Figure 3.4: Interpolated free surface elevation signal used as input.

3.3.2 Calibration Procedure

The calibration procedure is based on the principle of feedback correction and consists of several stages: first, a free surface elevation signal is selected from experimental data. A Fast Fourier Transform (FFT) is then applied to obtain the complete wave spectrum, from which only the linear components – namely the first-order harmonic modes – are isolated. The corresponding amplitudes and phases are extracted, in order to test whether the model can naturally regenerate second-order effects during the simulation.

Next, a specific location within the numerical domain – referred to as the target point – is selected: this is the location where the desired wave shape is expected to be reproduced. The phases of the harmonic components are then propagated backward to the generator using the linear dispersion relation 3.3. The motion file for the wavemaker is constructed by applying the Biesel transfer function 3.13, ensuring a consistent relationship between the imposed motion and the desired free surface elevation.

After each simulation, the signal is sampled at the control point. The linear components are extracted again via FFT, and the resulting output amplitudes and phases are compared to the target values. Any discrepancies are corrected by applying the following correction algorithm, inspired by Vyzikas et al. [5] and Stagonas et al. [8].

The correction strategy consists of two components:

1. A scaling factor applied to the theoretical motion, to correct the amplitude of the generated wave:
$$\alpha_{\text{in}}^{(i+1)} = \alpha_{\text{in}}^{(i)} \cdot \frac{\alpha_{\text{tgt}}}{\alpha_{\text{out}}^{(i)}}. \tag{3.16}$$
2. A phase shift introduced in the input signal, to account for time delays in wave formation:
$$\phi_{\text{in}}^{(i+1)} = \phi_{\text{in}}^{(i)} + \left(\phi_{\text{tgt}} - \phi_{\text{out}}^{(i)}\right). \tag{3.17}$$

$\alpha_{\text{in}}^{(i)}$ and $\phi_{\text{in}}^{(i)}$ represent the input amplitudes and phases at iteration i, while $\alpha_{\text{out}}^{(i)}$ and $\phi_{\text{out}}^{(i)}$ are the values measured at the target location. α_{tgt} and ϕ_{tgt} denote the theoretical target values.

This procedure is repeated until the desired tolerance in amplitude and phase is met. Once convergence is achieved, the complete spectrum of the simulated wave is analyzed in order to assess the emergence of nonlinear components, particularly second-order harmonics (both super- and sub-harmonics). The results will be discussed in Chapter 5.

3.3.3 Process Automation

All stages of the procedure have been fully automated using a custom `MATLAB` script which, given a target signal, performs the following operations:

1. computes the physical parameters of the target wave and propagates them from the target point to the wavemaker using the linear dispersion relation;

2. generates the motion signal of the piston-type wavemaker using the Biesel transfer function;
3. updates the XML configuration file accordingly;
4. launches `GenCase` and `DualSPHysics` in sequence;
5. extracts the free surface elevation at the control point using `MeasureTool`;
6. performs FFT analysis to estimate α_{out} and ϕ_{out};
7. computes the correction factors and updates the input for the next iteration;
8. stores diagnostic plots, log files, and numerical summaries for each run.

Chapter 4

Numerical Model

Following the methodological framework outlined in the previous chapters, this section introduces the numerical setup adopted for the SPH simulations of wave generation in DualSPHysics. The domain configuration is defined based on the characteristics of the target wave, ensuring that wave propagation is properly handled within the numerical tank.

In parallel, a parametric study is carried out to explore how key numerical parameters – such as particle spacing, smoothing length, and artificial speed of sound – affect the performance of the SPH method. This analysis supports the identification of efficient parameter settings to be used in the subsequent stages of the work. In addition, the treatment of wave absorption at the downstream boundary is also addressed, presenting a comparative study between passive absorption methods.

4.1 Numerical Domain Setup

The geometry of the computational domain was defined parametrically through XML files, and then generated via `GenCase`, the pre-processing utility integrated within the DualSPHysics framework. The domain was constructed as a numerical three-dimensional rectangular tank with a single particle layer in the transverse direction (y), resulting in 2-D configuration. The principal dimensions of the domain are defined as functions of the wavelength, λ, and the wave height, H, of the input signal, as summarized in Table 4.1. Specifically:

- the initial water depth is fixed at $h = \lambda$;
- the wave paddle (piston) is positioned at $x = 0$;
- upstream, the bottom extends to $x = -10H$ to maintain continuity when the piston moves backward;

- the active area ranges from $x = 0$ to $x = 5\lambda$, including both a propagation and a damping zone.

Table 4.1: Parametric definition of the computational domain.

Label	**Expression**	**Description**
h	λ	Water depth
x_{min}	$-10H$	Left end of the bottom boundary
x_{paddle}	0	Paddle (wave generator) position
x_{damping}	3λ	Start of the damping zone
x_{max}	5λ	Right end of the domain

The domain length was chosen to include both the near field – where waves are still influenced by the generator motion – and the far field, where wave properties stabilize and reach a fully developed form suitable for quantitative evaluation. This distinction is relevant for assessing the accuracy of wave propagation over space, as already discussed in Chapter 3.

A depth of $h = \lambda$ was selected to fully meet deep-water criteria, according to which the water depth satisfies $h/\lambda \geq 0.5$. Under these conditions, the seabed has negligible influence on wave motion, and the velocity profile decays exponentially with depth.

To reduce wave reflection at the downstream boundary, a one-way numerical damping zone is implemented between $x = 3\lambda$ and $x = 5\lambda$, in agreement with the results from subsection 4.3.1. This region gradually reduces the particle velocity along the x-axis; the damping intensity is governed by the `redumax` parameter, here set to 5.

The domain, shown in Figure 4.1, was thus designed to provide a solid compromise between physical accuracy, numerical robustness, and computational performance, making it suitable for a systematic campaign of parametric simulations.

Waves are generated using a rigid piston whose motion is defined through an external input file to produce regular waves under linear conditions, as discussed in Chapter 3.

All numerical parameters and settings that were not analyzed in the parametric study were kept constant throughout all simulations, as listed in Table 4.2. The chosen configuration is consistent with those typically found in the literature for similar simulations [38, 39].

This layout allows the effects of the key investigated parameters – wave steepness, particle spacing dp, artificial speed of sound c_s, kernel smoothing length h – to be isolated and compared effectively.

Table 4.2: Constant numerical parameters and settings used in all simulations

Parameter	Value	Description
`Kernel`	Wendland [40]	Interpolation kernel function
`Boundary`	mDBC [31]	Boundary method (no-slip)
`ViscoTreatment`	Laminar [4]	Viscosity treatment
`Viscosity`	$1.0 \cdot 10^{-6}$ m^2/s	Kinematic viscosity
`Shifting`	Full (3) [32]	Shifting mode applied to all particles
`DensityDT`	Fourtakas [41]	Density correction model
`StepAlgorithm`	Symplectic [42]	Time integration scheme
`CFL`	0.2	Courant number
`Gamma`	7.0	Exponent in the equation of state

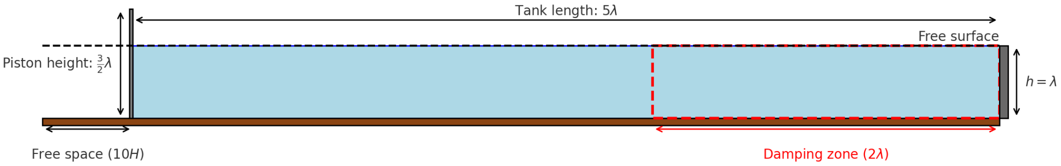


Figure 4.1: Schematic of the numerical wave tank, dimension in terms of λ and H.

4.2 Parametric Study

Once the computational domain and baseline numerical parameters are defined, a parametric study is conducted to validate the numerical model and analyze the behavior of the SPH method under such variations. This study is structured as follows:

1. Steepness variation: six distinct configurations were simulated, keeping the wave height constant while varying the wave period T from 0.80 s to 2.50 s. This setup resulted in wave steepness ranging approximately from 1:10 to 1:100, to analyze its impact on numerical accuracy and stability.

2. Spatial resolution analysis: for three of the previous cases ($T = 0.80$ s, 1.50 s, 2.50 s), the particle spacing *dp* was varied between 0.010 m, 0.015 m, and 0.020 m to assess its influence on the precision of the solution.

3. Variation of SPH numerical parameters: focusing on the intermediate case ($T = 1.50$ s, $dp = 0.015$ m), the effects of the following parameters were analyzed separately, in order to determine the sensitivity of the method to artificial compressibility and kernel support, respectively:
   - `coefsound`: 20, 25, 30 (with `coefh` fixed at 1.3);
   - `coefh`: 1.2, 1.3, 1.5 (with `coefsound` fixed at 20).

All tests were performed by varying only one parameter at a time, while keeping all other conditions unchanged as specified in Table 4.2, to isolate the specific influence of each variable.

For each configuration, the following aspects were evaluated: wave profile quality (e.g., height and phase error relative to the target signal), computational efficiency and agreement with theoretical expectations. The free surface elevation was analyzed using `MeasureTool`, the post-processing tool implemented in DualSPHysics. Measurements were taken at multiple locations along the domain, ranging from 0.5 to 3λ from the wave paddle. This range spans both the near field and the far field, to properly capture the wave reproduction.

4.2.1 Steepness Variation

Wave steepness is a dimensionless parameter that describes the sharpness of the wave profile and is defined as the ratio between wave height H and wavelength λ:

$$S = \frac{H}{\lambda}. \tag{4.1}$$

Steep waves ($S > 0.05$) involve sharper gradients of velocity and pressure, requiring increased numerical resolution, while long and flat waves (low steepness) are less demanding in terms of numerical stability. In fact, at low steepness waves tend to preserve a regular and predictable shape during propagation, with negligible nonlinear effects and more stable energy distribution. As steepness increases, nonlinear interactions between spectral components can lead to wave deformation, including energy transfer to higher frequencies, harmonic phase coupling, and eventually wave breaking [43, 44].

The goal of this analysis is to assess the model's ability to accurately simulate regular waves over a wide range of steepness values, while preserving physical consistency and numerical quality. To isolate the effect of steepness, the wave height was fixed at $H = 0.10$ m, while the wave period T was varied. In deep water, the wavelength λ is related to the period via the dispersion relation 3.3 [35]. As T increases, so does λ, resulting in a lower steepness. Six cases were analyzed, as summarized in Table 4.3:

Table 4.3: Test cases for steepness variation

Case	Period T [s]	Estimated steepness H/λ [-]
1	0.80	$\sim$ 1:10
2	1.00	$\sim$ 1:15
3	1.20	$\sim$ 1:25
4	1.50	$\sim$ 1:40
5	2.00	$\sim$ 1:75
6	2.50	$\sim$ 1:100

In all cases, the other numerical parameters were kept constant: $dp = 0.015$ m, `coefh` $= 1.3$, `coefsound` $= 20$. Simulations were carried out in the domain described in Section 4.1, without modifications to the geometry or boundary conditions. Each simulation was run for at least 15 wave periods to ensure stabilization of the wave train. The time step was automatically computed, based on the Symplectic time integration scheme and the CFL condition 2.24.

The free surface elevation was analyzed at multiple locations along the domain; at each location, the following quantities were assessed:

- simulated wave amplitude compared with the theoretical target value;
- phase shift relative to the theoretical wave profile.

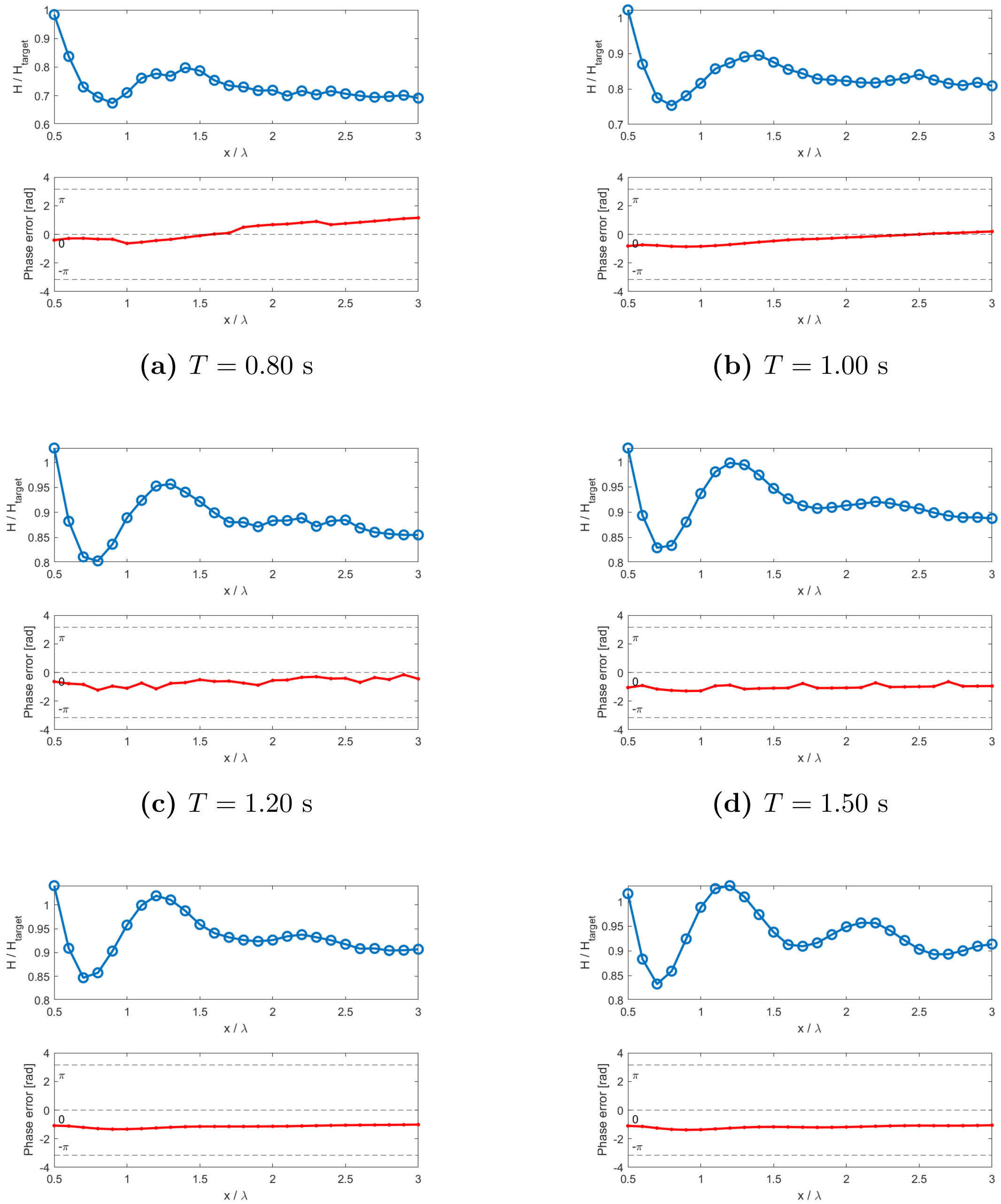
Computational efficiency was not considered at this stage, as the domain length varies with λ (as presented in Section 4.1), directly affecting the number of particles. This aspect will be addressed later, under conditions of fixed domain size.

Results and Discussion The analysis of free surface elevation along the domain shows consistent behavior with physical and numerical expectations. As the wave period increases (and steepness decreases), wave height attenuation diminishes and the waveform is better preserved, as shown in Figure 4.2. In high steepness cases (e.g., $H/\lambda \approx 1:10$), wave elevation drops significantly within the first wavelengths, indicating stronger dissipation, likely due to insufficient spatial resolution to capture steep gradients.

In all cases, a distinct oscillatory behavior in the free surface elevation is observed near the piston, due to the presence of evanescent modes and initial wave development. As stated before in Chapter 3, this transient region – known as the near field – typically extends over the first wavelength and gradually stabilizes at approximately 1–1.5 wavelengths from the wavemaker. For this reason, in the present study, the far field is operationally defined to begin at $x = 1.5\lambda$.

The phase analysis shows a general mismatch between the simulated and theoretical wave profiles in all cases. However, the phase error behaves differently depending on steepness. In high steepness cases, the phase error initially decreases along the domain, reaches zero, and subsequently increases again, indicating a temporary wave front alignment that later diverges (Figure 4.3). In low steepness cases, the phase shift remains approximately constant over time, even if slightly oscillating in some cases. This behavior is likely due to the more regular and linear nature of the wave, which reduces numerical distortion.

Overall, the results confirm that low to moderate steepness waves can be reliably simulated using the parameter set adopted in this SPH configuration.



(a) $T = 0.80$ s

(b) $T = 1.00$ s

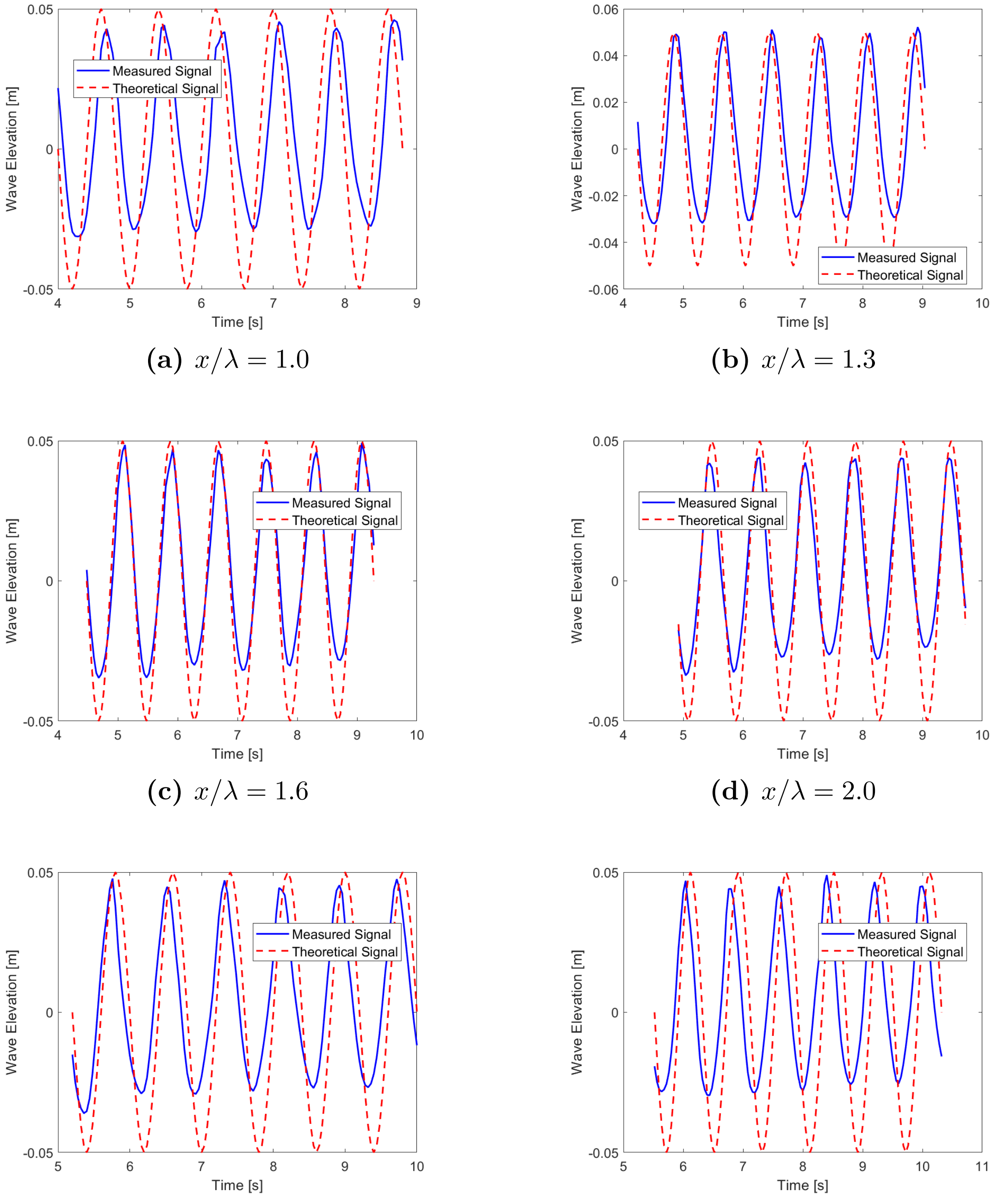
(c) $T = 1.20$ s

(d) $T = 1.50$ s

(e) $T = 2.00$ s

(f) $T = 2.50$ s

Figure 4.2: Wave elevation and phase error relative to the target solution.



(a) $x/\lambda = 1.0$

(b) $x/\lambda = 1.3$

(c) $x/\lambda = 1.6$

(d) $x/\lambda = 2.0$

(e) $x/\lambda = 2.5$

(f) $x/\lambda = 3.0$

Figure 4.3: Detail of the phase shift for high steepness condition ($T = 0.80$ s)

4.2.2 Spatial Resolution Variation

In the SPH method, spatial resolution is determined by the initial spacing between fluid particles, denoted as *dp*. This parameter directly affects the model's ability to accurately represent continuous fields such as pressure, velocity, and surface elevation. A finer resolution (smaller *dp*) improves the capacity to capture local gradients and reduces physical distortions, especially for steeper waves, but at the cost of a significantly higher particle count and computational effort [6].

In this study, three representative wave steepness conditions were selected:

- High steepness: $T = 0.80$ s, $H/\lambda \approx 1 : 10$;
- Moderate steepness: $T = 1.50$ s, $H/\lambda \approx 1 : 40$;
- Low steepness: $T = 2.50$ s, $H/\lambda \approx 1 : 100$.

Since the wavelength, λ, varies across the selected cases, the computational domain also scales accordingly, and so does the total particle count. Therefore, it is important to assess the effect of *dp* not only in absolute terms, but also in terms of its ratio to wave height and wavelength.

For each case, the particle spacing was varied among 0.010 m, 0.015 m, and 0.02 m, in order to meet at least 50-100 particles per wavelength, following the recommended resolution range for regular wave propagation in SPH simulations [25] . Moreover, since the input wave height remained fixed at $H = 0.10$ m, the chosen *dp* values result in 10, 6.67, and 5 particles per wave height, respectively – a key indicator of vertical resolution (Table 4.4).

Table 4.4: Resolution and number of particles per wave height ($H = 0.10$ m)

dp [m]	Particles per wave height
0.010	10.00
0.015	6.67
0.020	5.00

All other numerical and geometrical parameters were kept constant (`coefh` = 1.3, `coefsound` = 20), and the domain geometry and boundary conditions were left unchanged. Simulations were run for a duration of at least 15 wave cycles. Free surface elevation was analyzed at different measurement points from the wave paddle, covering both near and far field regions. The evaluation criteria included:

- Simulated wave height compared to the theoretical target value;
- Phase error as the temporal shift between theoretical and simulated peaks;

- Total number of particles;
- Wall-clock simulation time.

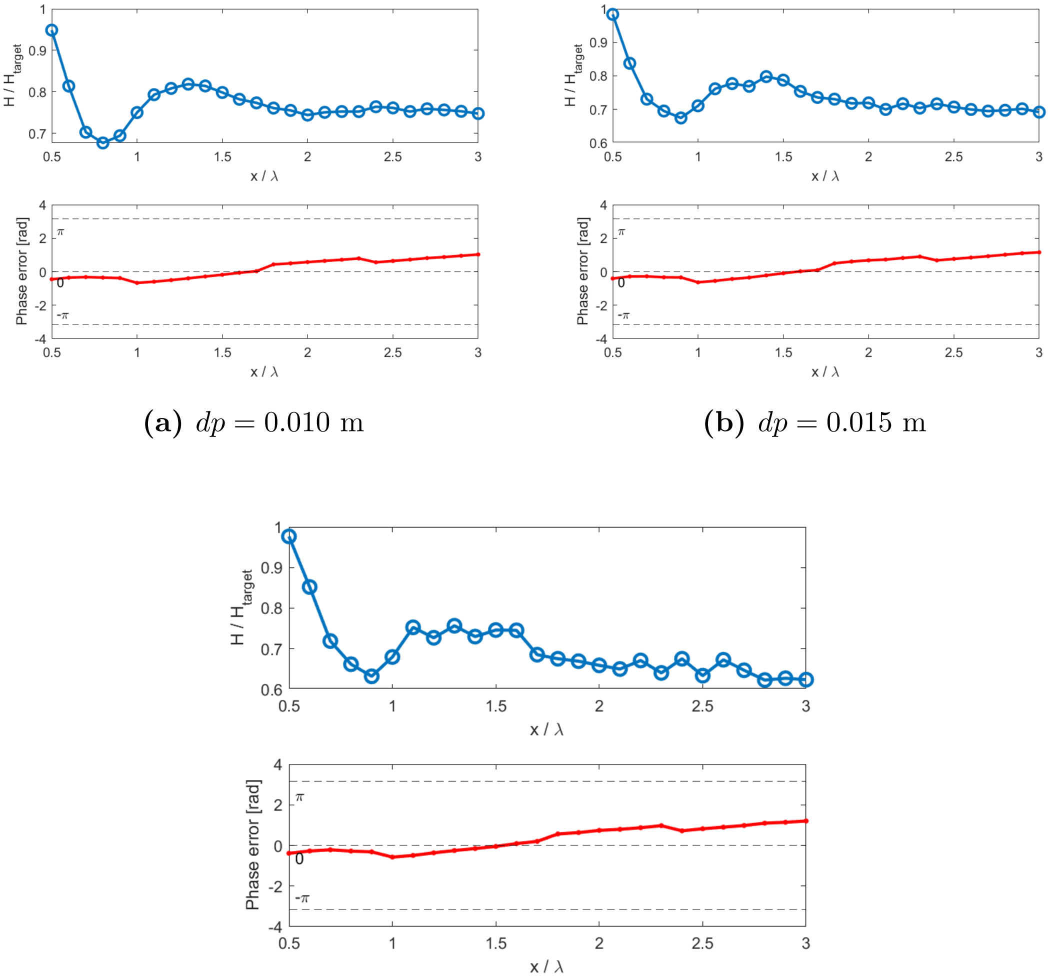
Results and Discussion

Case $T = 0.80$ s – High Steepness This scenario, with steepness around 1:10, showed strong sensitivity to resolution (Figure 4.4). The finest spacing ($dp = 0.010$ m) maintained wave amplitude close to theoretical expectations throughout the domain. Coarser settings led to increased attenuation: at $dp = 0.020$ m, normalized surface elevation dropped below 70% downstream, suggesting numerical dissipation due to insufficient resolution. However, phase accuracy showed only minor variations across resolutions, and all cases exhibited comparable phase shifts.

Computationally, higher resolution significantly increased particle count and runtime, as shown in Table 4.5. Reducing the spacing from $dp = 0.020$ m to $dp = 0.010$ m more than doubled the cost. The intermediate value ($dp = 0.015$ m) provided a fair balance between accuracy and efficiency.

Table 4.5: Number of particles and simulation times for $T = 0.80$ s

dp [m]	Particles	Simulation Time [s]	Δ Time
0.010	54,020	515.6	–
0.015	25,001	256.2	-50%
0.020	14,520	188.0	-64%



(a) $dp = 0.010$ m

(b) $dp = 0.015$ m

(c) $dp = 0.020$ m

Figure 4.4: Comparison of different spacial resolution for high steepness

Case $T = 1.50$ s − Moderate Steepness For moderate steepness, results were less resolution-dependent. All cases yielded wave profiles close to theoretical predictions (Figure 4.5), although the finest still performed best in amplitude preservation. Phase errors remained stable across all resolutions as well, even at $dp = 0.020$ m.

As expected, computational demand increased with resolution (as shown in Table 4.6); yet, for this wave regime, $dp = 0.015$ m again proved optimal, ensuring accurate results at substantially lower computational cost than $dp = 0.010$ m.

Table 4.6: Number of particles and simulation times for $T = 1.50$ s

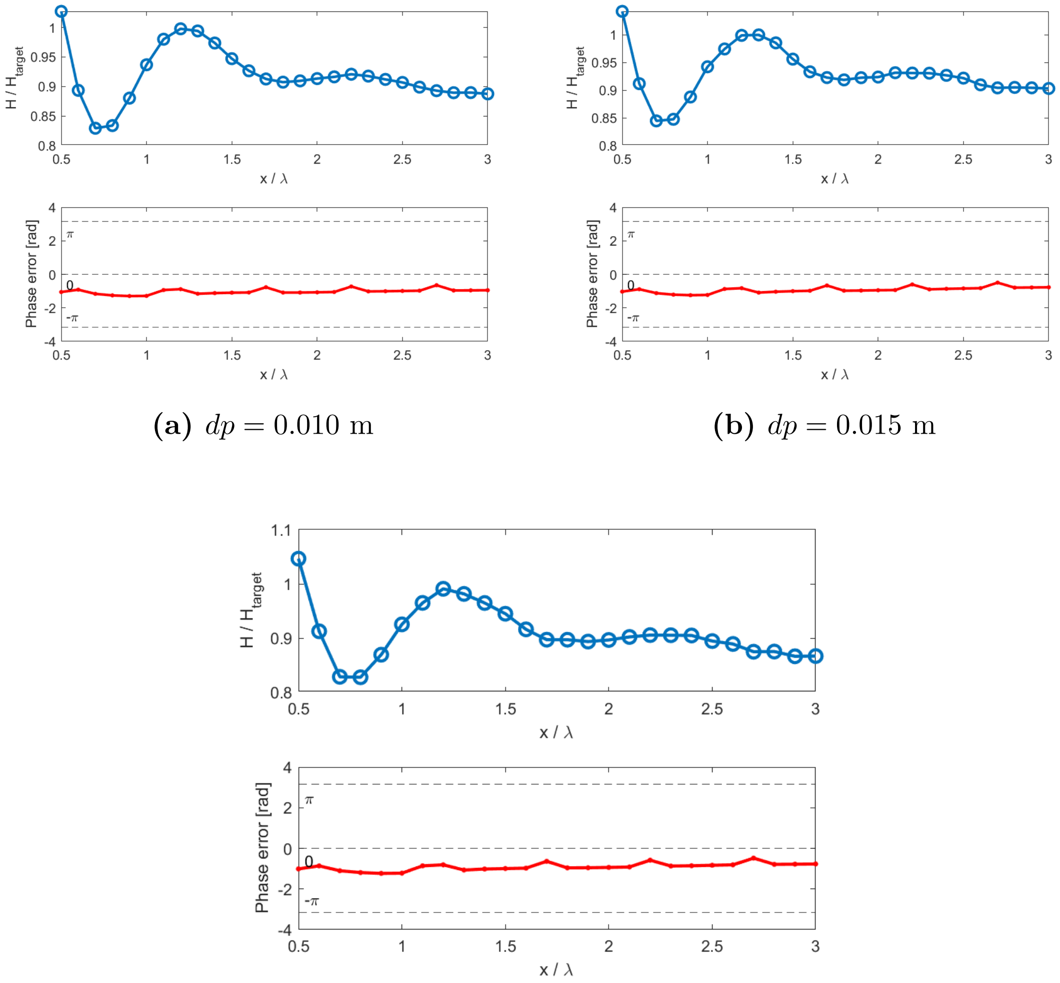
dp [m]	Particles	Simulation Time [s]	Δ Time
0.010	629,021	2,058.3	–
0.015	282,225	1,057.6	-49%
0.020	160,698	647.4	-69%

Case $T = 2.50$ s - Low Steepness For the low steepness wave, the influence of resolution variation was minimal (Figure 4.6). Wave propagation remained coherent and stable across all the cases. Even the coarsest resolution proved sufficient, making it possible to significantly reduce computational cost (Table 4.7) without compromising result quality.

Table 4.7: Number of particles and simulation times for $T = 2.50$ s

dp [m]	Particles	Simulation Time [s]	Δ Time
0.010	4,794,828	28,840.5	–
0.015	2,139,660	9,305.4	-68%
0.020	1,207,194	4,466.9	-85%

This analysis confirms that spatial resolution has a significant impact on SPH simulation quality. For steep waves, at least 7-10 particles per wave height are required to maintain accuracy and stability. For longer and flatter waves, 5-7 particles are usually adequate to ensure correct propagation. The intermediate value $dp = 0.015$ m, corresponding to 6.67 particles per wave height, proved to be a suitable compromise between computational efficiency and numerical fidelity.

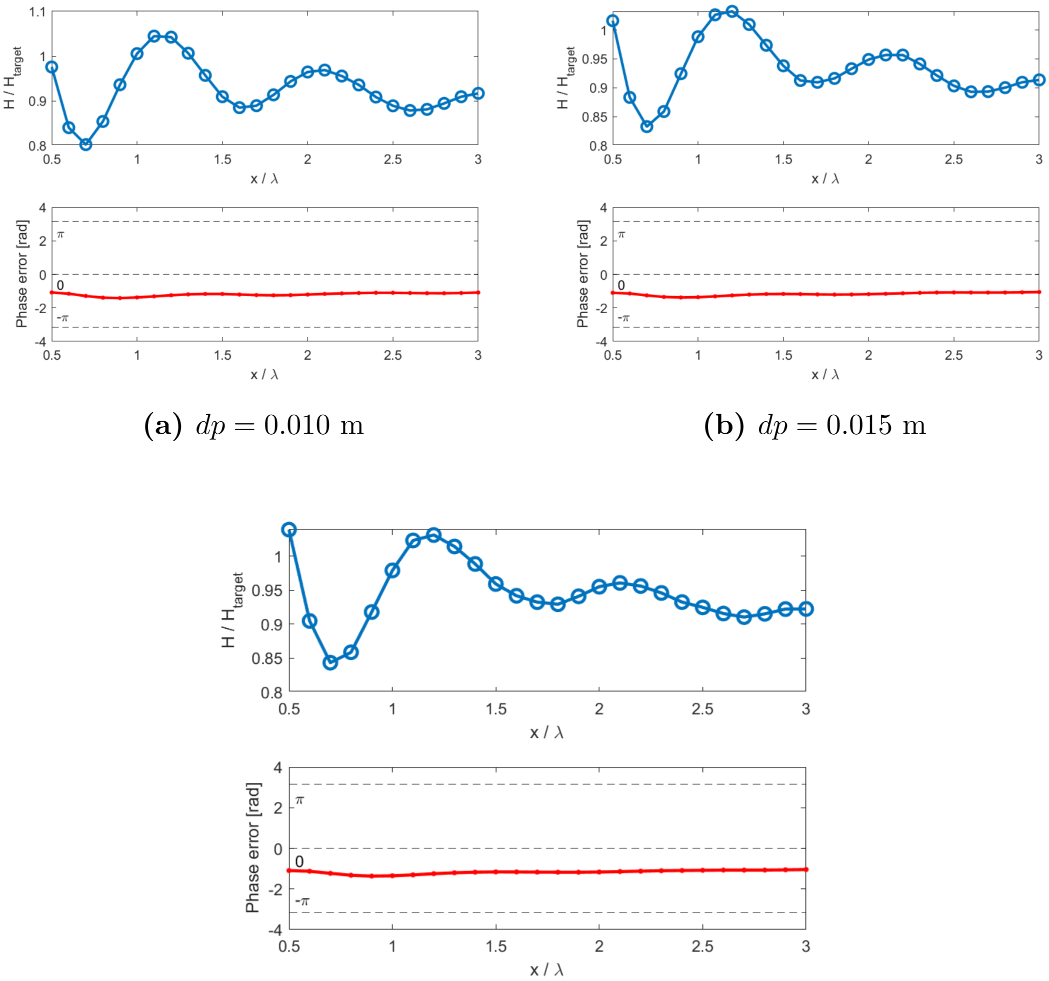


(a) $dp = 0.010$ m

(b) $dp = 0.015$ m

(c) $dp = 0.020$ m

Figure 4.5: Comparison of different spacial resolution for moderate steepness



(a) $dp = 0.010$ m

(b) $dp = 0.015$ m

(c) $dp = 0.020$ m

Figure 4.6: Comparison of different spacial resolution for low steepness

4.2.3 Numerical Speed of Sound Variation

In the SPH method, fluid pressure is computed using an artificial equation of state that relates pressure to density, assuming a pseudo-compressible behavior, as mentioned in subsection 2.3.2. The typical relation adopted in DualSPHysics is the equation of state (2.18), used in the WCSPH formulation. Here, c_s is the imposed numerical speed of sound, which governs the fluid's pressure response to density variations. The latter is computed as:

$$c_s = \texttt{coefsound} \cdot u_{ref}, \tag{4.2}$$

where u_{ref} is a characteristic velocity, typically the dambreak velocity.

Higher values of `coefsound` lead to a less compressible response, improving the fidelity of the pressure field and numerical stability. However, this comes at the cost of reduced time step size, due to the CFL condition [1]; as a result, increasing values of `coefsound` extends the total simulation time. Conversely, excessively low values may lead to poor pressure representation, numerical instabilities, or spurious oscillations.

In practice, `coefsound` is a key tuning parameter that balances physical accuracy and computational cost. To ensure weak compressibility, in this study the artificial speed of sound coefficient was set to satisfy the condition $c_s \geq 10 \cdot u_{max}$, as recommended in the DualSPHysics formulation [30, 25].

All tests were performed with a wave period $T = 1.50$ s, selected for its numerical stability; wave height $H = 0.10$ m, particle spacing $dp = 0.015$ m and smoothing coefficient `coefh` $= 1.3$. The tested values for `coefsound` were:

$$\texttt{coefsound} = \{20,\ 25,\ 30\}$$

All simulations used the same parametric domain, with consistent setup and duration (covering at least 15 wave cycles). Free surface elevation and phase error were measured at multiple positions along the domain and wall-clock time was recorded as well.

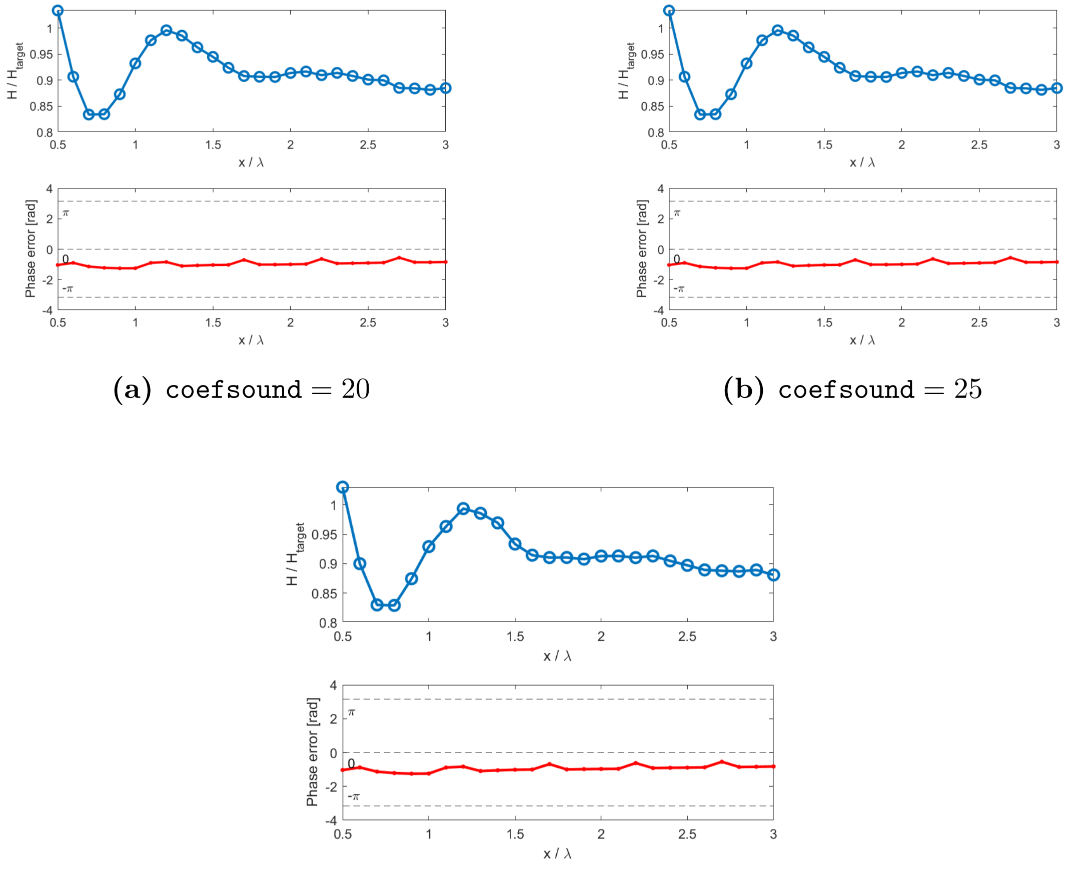
Results and Discussion The results in Figure 4.7 show that changing the `coefsound` parameter under moderate steepness conditions has a limited impact on simulation quality. Across all test cases, the simulated free surface elevation remained in close agreement, with only minor differences observed. Even at the lowest tested value (`coefsound` $= 20$), both amplitude and phase remained within acceptable limits.

From a computational standpoint, higher `coefsound` values significantly increased simulation time, as expected (Table 4.8).

Table 4.8: Simulation time as a function of `coefsound` (case $T = 1.50$ s)

`coefsound`	**Simulation Time** [s]	**Change vs.** `coefsound`**=20**
20	1,057.6	–
25	1,305.8	+23%
30	1,568.9	+48%

Although higher `coefsound` values may offer marginal improvements in wave accuracy, the increase in computational cost is considerable. Given the absence of instabilities or significant errors, the default setting of `coefsound` = 20 used in this study is considered appropriate.



(a) `coefsound` = 20

(b) `coefsound` = 25

(c) `coefsound` = 30

Figure 4.7: Comparison of different coefsound values for moderate steepness

4.2.4 Smoothing Length Variation

The extent of the interpolation kernel support, that is the maximum distance within which a particle influences its neighbors during smoothing operations, is a crucial quantity in the SPH formulation, as stated in subsection 2.3.1. This distance, known as the smoothing length h, is computed as:

$$h = \texttt{coefh} \cdot dp, \tag{4.3}$$

The value of `coefh` determines the number of neighboring particles involved in the interpolation and directly affects the quality of the approximated physical fields [6]. Higher values of `coefh` produce a wider kernel support, including more neighbors and resulting in smoother and more stable fields. However, this may lead to excessive diffusion, reduced resolution, and signal attenuation. Conversely, very low values of `coefh` decrease the influence radius, increasing sensitivity to noise and the risk of numerical instabilities [26, 6].

The goal of this analysis is to evaluate how `coefh` affects the propagation of regular waves, to identify a value that ensures a good compromise between numerical and physical accuracy and computational cost.

The `coefh` variation was tested using the case with $T = 1.50$ s, previously selected as the reference. The numerical and geometric parameters were kept constant: $dp = 0.015$ m, `coefsound` $= 20$, wave height $H = 0.10$ m.

The computational domain, initial conditions, and boundary conditions remained unchanged. As in previous analyses, the simulation time was set to cover at least 15 wave cycles to ensure full stabilization of the wave train.

The tested values for the `coefh` parameter were chosen in order to guarantee sufficient neighbour interaction [25]:

$$\texttt{coefh} = \{1.2,\ 1.3,\ 1.5\}$$

For each case, wave elevation, phase error (with respect to the theoretical signal), and computational performance were evaluated.

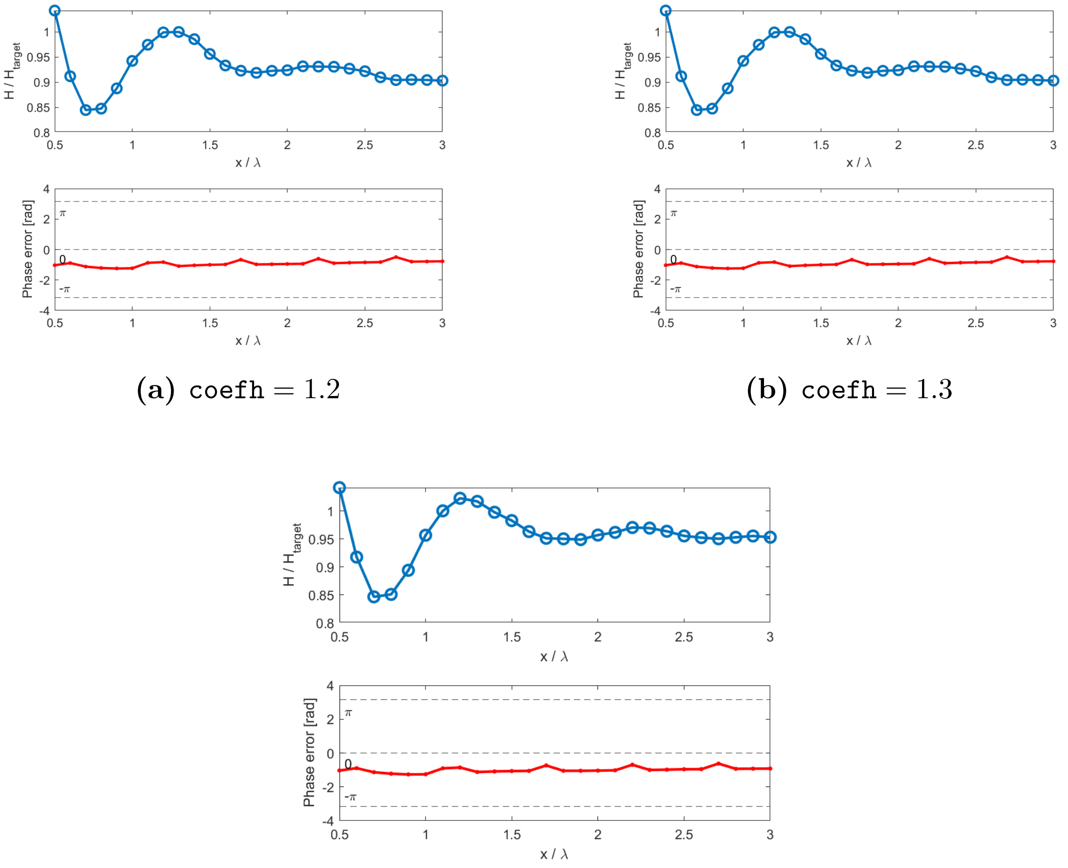
Results and discussion The results in Figure 4.8 show that the kernel smoothing length, controlled by `coefh`, moderately influences the numerical behavior of wave propagation.

In all configurations, the wave elevation is reasonably well preserved along the domain. The case with `coefh` $= 1.5$ shows slightly higher and more uniform wave amplitudes both in near and far field areas, suggesting reduced numerical dissipation. In contrast, lower values (`coefh` $= 1.2$ and 1.3) lead to wave elevations that converge at slightly reduced levels, but still acceptable. The phase error remains low and relatively constant in all cases, with minimal variations across different values of `coefh`.

From a computational perspective, increasing `coefh` leads to a higher number of neighbors to process, thus raising the computational load. The simulation time grows accordingly, as summarized in Table 4.9.

Table 4.9: Simulation time as a function of `coefh` (case $T = 1.50$ s)

`coefh`	**Simulation time [s]**	**Δ Time relative to 1.2**
1.2	939.1	–
1.3	1,057.6	+13%
1.5	1,103.7	+17.5%



(a) `coefh` $= 1.2$

(b) `coefh` $= 1.3$

(c) `coefh` $= 1.5$

Figure 4.8: Comparison of different smoothing length values for moderate steepness

The default value `coefh` = 1.3 chosen for this work proves to be a well-balanced choice, providing numerical stability without imposing excessive computational cost.

4.3 Wave Absorption Strategy and Comparative Evaluation

A critical aspect in the numerical generation of waves is the effective treatment of their downstream propagation, particularly to prevent unphysical reflections at the boundary. Reflected waves can interfere destructively or constructively with the incident wave train, introducing errors in the solution, distorting the free surface elevation in the far field, and compromising the accuracy of wave calibration and validation [38].

4.3.1 Absorption Methods in DualSPHysics

The `DualSPHysics` framework provides different options to mitigate wave reflections, which can be classified in three main categories:

1. Passive absorption: These methods rely on geometric or artificial damping mechanisms to dissipate wave energy as it approaches the downstream boundary.

    - Sloped beach: A physical absorption technique that mimics natural dissipation by introducing a sloped bed at the end of the tank. As waves break on the slope, their energy is dissipated through bottom friction and turbulence. While simple and effective for long waves in shallow water, this approach is spatially expensive and less efficient for short waves or deep water conditions [36].
    - Damping zone: A numerically implemented region near the downstream boundary, where particle velocities are gradually reduced in time. DualSPHysics reduces particle velocities after each time step using a quadratic decay function:

    $$\mathbf{v} = \mathbf{v_0} \cdot \left(1 - \beta \cdot \Delta t \cdot \left(\frac{x - x_0}{x_1 - x_0}\right)^2\right), \tag{4.4}$$

    where $\mathbf{v}$ is the particle velocity, $\mathbf{v_0}$ is the initial velocity at x_0, β is a damping coefficient (typically set to 10), Δt is the time step, x_0 and x_1 define the start and end of the damping zone.

    This method offers good efficiency in compact domains and can be tuned to different wave conditions with minimal setup effort [39].

2. Relaxation Zone: An increasingly adopted approach is the use of relaxation zones, where the computed velocity and surface elevation are blended with a predefined target state via a spatial weighting function $C(x)$ [39].

   The blended horizontal and vertical velocities are computed respectively as:

   $$u = C\,u_{cl} + (1 - C)\,u_{pt}, \tag{4.5}$$

   $$w = C\,w_{cl} + (1 - C)\,w_{pt}. \tag{4.6}$$

   where u_{cl} and w_{cl} are the theoretical velocities and u_{pt} and w_{pt} are the ones computed in the numerical model.

   The relaxation zone offers high absorption efficiency with minimal reflected energy, provided that the zone length and weighting function are tuned correctly. It is also compatible with wave generation, allowing its use at both ends of the domain [38, 39].

4.3.2 Evaluation of Sloped Beach and Damping Zone Performance

In this study, only passive wave absorption strategies were adopted – namely, the sloped beach and the numerical damping zone –, since these are among the most commonly adopted techniques in DualSPHysics due to their geometric simplicity and independence from real–time feedback mechanisms [38].

To assess their relative performance, a comparative test was carried out using a regular wave case characterized by a wave height $H = 0.1$ m and period $T = 1.50$ s. The SPH parameters were set with particle spacing $dp = 0.015$ m, `coefsound`$= 20$ and `coefh`$= 1.3$, while all other values were kept at their default settings as defined in the reference Table 4.2. This wave profile corresponds to the standard configuration identified in the parametric study.

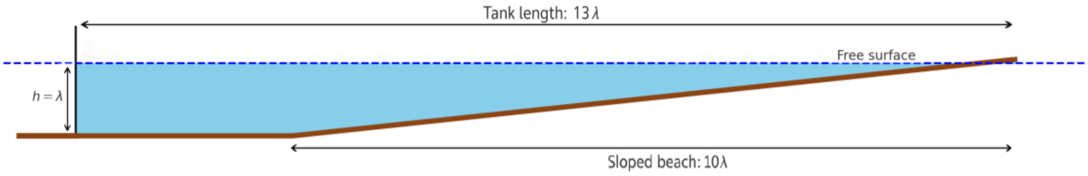
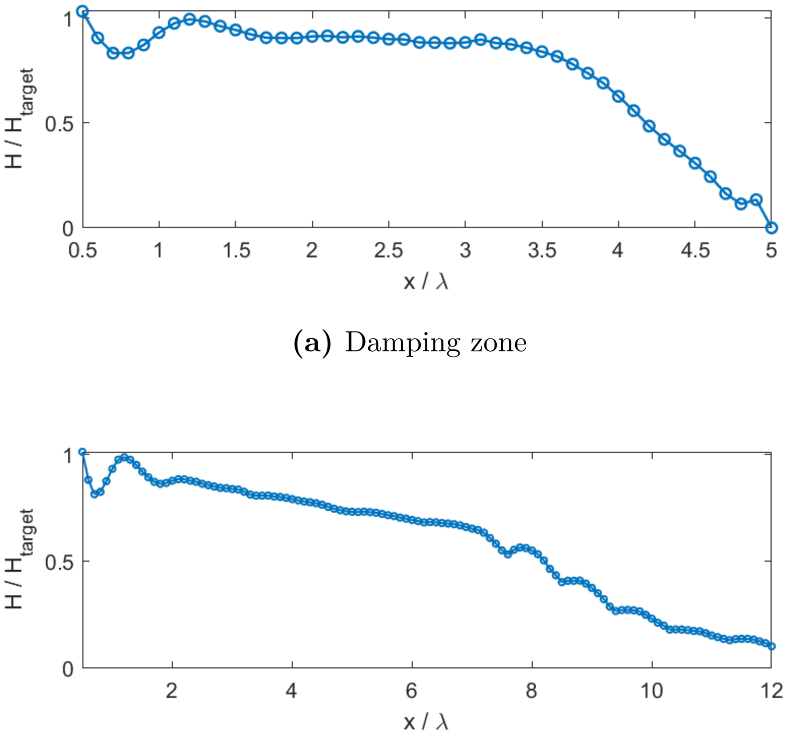


Figure 4.9: Schematic of the numerical wave tank with a sloped beach

In both simulations, the main computational domain consisted of a flat–bottom propagation zone of length 3λ, with a constant water depth equal to λ to ensure deep water condition, in consistency with all test cases. The only variation concerned the downstream absorption region, configured as follows:

- Sloped beach: a 1:10 slope starting at the end of the flat region and extending over a total length of approximately 10λ, gradually reaching the still water level (Figure 4.9).
- Damping zone: a region of length 2λ, in which particle velocities were progressively reduced following the formulation (4.4) (Figure 4.1). A damping coefficient $\beta = 5$ was adopted to account for the extended damping region, set to twice the standard length (2λ instead of λ, for which $\beta = 10$ is typically used [25]).

Performance evaluation focused on wave attenuation, evaluated by analyzing the evolution of the wave along the domain, with particular attention to the decay in the absorption region, and computational efficiency, measured by the total simulation time and the number of fluid particles required.



(a) Damping zone

(b) Sloped beach

Figure 4.10: Wave height evolution along the domain.

Results and Discussion The results of the comparison are shown in Figure 4.10, summarized in Table 4.10 and discussed below.

Table 4.10: Comparison between sloped beach and damping zone in terms of absorption efficiency and computational performance.

Parameter	**Sloped Beach**	**Damping Zone**
Implementation	Geometric modeling	Velocity scaling
Absorption effectiveness	Moderate	High
Propagation zone length	3λ	3λ
Absorption region length	10λ	2λ
Total domain length	13λ	5λ
Number of particles	569,097	282,225
Simulation runtime	2,117 s	1,058 s

The numerical damping zone led to a rapid and complete attenuation of wave amplitude within the absorption region. The decay was smooth and progressive, with the wave height reaching zero at the end of the damping zone. This behavior indicates effective energy dissipation and a high absorption performance, minimizing unwanted disturbances in the domain and preserving both the target wave height and the waveform shape.

Conversely, the sloped beach exhibited a slower and less pronounced amplitude decay. Although the wave height decreased significantly along the slope, it did not vanish entirely, remaining at low but non negligible levels by the end of the domain. This residual energy may potentially result in minor distortions of the waveform within the propagation zone, especially in long term or multi-wave scenarios.

Beyond its physical performance, the damping zone offers clear advantages in terms of computational efficiency. The simulation with the sloped beach required more than twice the runtime and double the number of particles , primarily due to the larger domain needed to accommodate the 10λ slope. By contrast, the damping zone achieved better absorption results using a compact 2λ region attached to the propagation zone, significantly reducing both computational cost and memory usage.

The use of the damping coefficient $\beta = 5$, as previously defined, proved to be an effective compromise between computational cost and wave absorption performance. As a result, this damping strategy was adopted in all simulations.

Chapter 5

Results

The following sections outline the calibration process described in Chapter 3 and present the results of the controlled wave generation approach based exclusively on the linear components of a target wave. The procedure is first applied to regular waves, used here as a reference to validate the method, and then is extended to irregular wave groups with the necessary adaptations.

5.1 Regular Wave Calibration

As a preliminary verification case, the procedure was initially tested on a regular monochromatic wave. In this simplified scenario, the calibration focused exclusively on correcting the amplitude of the linear spectrum, neglecting both the phase and the higher-order harmonic components. This verification will be addressed more thoroughly in the context of irregular waves, where spectral phase coherence plays a crucial role.

A regular target wave was selected by specifying its period and height, along with a fixed point in the domain where the wave signal was to be accurately reproduced. In this study, the control point is located at a distance of $x = 1{,}5\lambda$ from the generator, corresponding to the beginning of the far field, where the wave is fully developed and no longer affected by transients or evanescent modes.

Starting from this target condition, the signal was prescribed to the piston-type wavemaker using the Biesel transfer function (3.13). At the end of each simulation, the free surface elevation signal was extracted at the control point using the `MeasureTool` utility and analyzed in post-processing via FFT. To avoid distortions due to the initial transient, the analysis was restricted to a stationary time window. The FFT yielded the spectral amplitude a associated with the wave period, from which the measured wave height is computed as $H_{\text{out}} = 2a$.

The comparison between H_{out} and the target value H_{target} allowed updating the input H_{in} for the next iteration using:

$$H_{\text{in}}^{i+1} = H_{\text{in}}^{i} \times \frac{H_{\text{target}}}{H_{\text{out}}^{i}}.$$

This relation progressively reduces the discrepancy between the simulated and target wave heights; the process is repeated until the relative error falls below a predefined tolerance threshold:

$$\left| \frac{H_{\text{target}}}{H_{\text{out}}^{i}} - 1 \right| < \varepsilon.$$

In this study, a tolerance of $\varepsilon = 5\%$ was adopted, set with a maximum of 20 iterations.

5.1.1 Amplitude Calibration Convergence

The calibration procedure was tested on three cases characterized by different steepness levels, obtained by varying the wave period while keeping the height fixed at $H = 0.10$ m. The selected cases – $T = 0.80$ s, $T = 1.50$ s, and $T = 2.50$ s – correspond to steepness values of approximately 1:10, 1:40, and 1:100, representative of high, moderate, and low wave steepness as defined in earlier tests. All other simulation parameters (such as, computational domain, boundary conditions, spatial resolution, total simulation time) were kept constant and matched the standard configuration defined in Chapter 4.

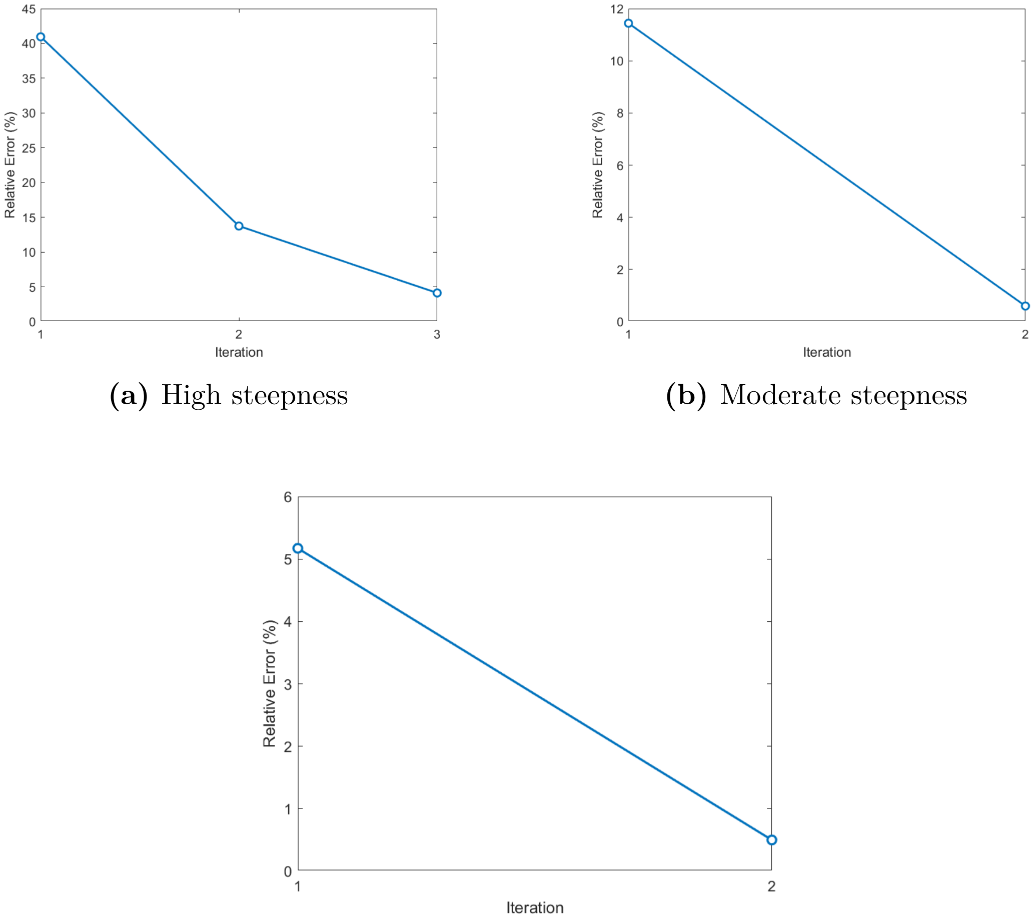
The results, presented in the three subplots of Figure 5.1, show that the case with the highest steepness ($T = 0.80$ s) required three iterations to reach the target wave height. In contrast, for the less steep cases ($T = 1.50$ s and $T = 2.50$ s), convergence was achieved in only two iterations.

In all cases, the first attempt underestimated the target wave height; however, the error was significantly larger in the steepest case and only minor for the less steep conditions. The ratio $H_{\text{target}}/H_{\text{out}}$ quickly approached unity in the following iterations, confirming the effectiveness of the method.

This behavior aligns with expectations: steeper waves are more sensitive to numerical dissipation and nonlinear deformations, which reduce the predictability of the system's response and may require additional iterations to correctly calibrate the input.

5.2 Irregular Wave Calibration

Following the successful validation of the calibration procedure on regular monochromatic waves, the methodology was extended to the case of irregular wave groups



(a) High steepness

(b) Moderate steepness

(c) Low steepness

Figure 5.1: Convergence of the amplitude calibration procedure for three steepness levels.

characterized by a predefined linear spectrum. This more complex scenario allows to evaluate the robustness of the approach under realistic conditions and to investigate the model's ability to spontaneously reproduce nonlinear interactions.

The irregular wave signal used in this study was extracted from an experimental campaign conducted at Oregon State University's wave tank facility, as described in [45]. The tests were part of the preliminary assessment of the SWINGO wave energy converter and aimed to characterize the system's response under realistic sea-state conditions. The experiments were carried out in a rectangular wave tank measuring 48.8 m in length, 26.5 m in width, and 2.1 m in depth. Waves were generated by piston-type wavemakers; the surface elevation was recorded along the flume to capture both near field and far field wave behavior.

The selected experimental signal corresponds to a wave group with a notable steepness and a clear focusing behavior: an initial low-amplitude oscillation gradually builds up to a well-defined peak around $t = 21\,\mathrm{s}$ (Figure 5.2).

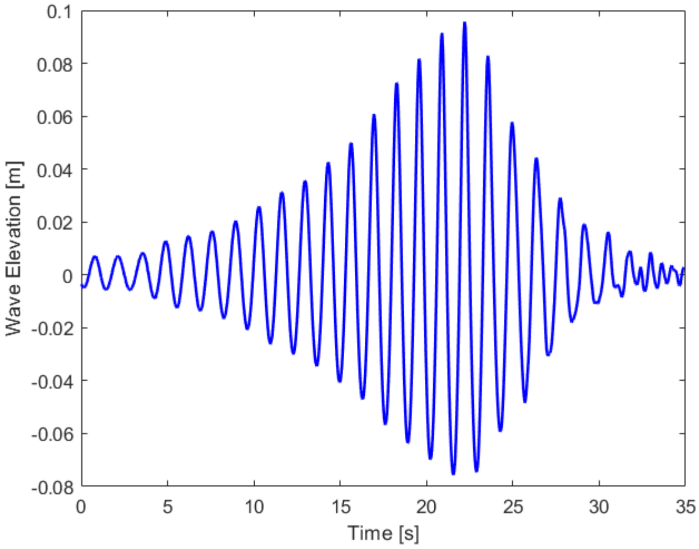
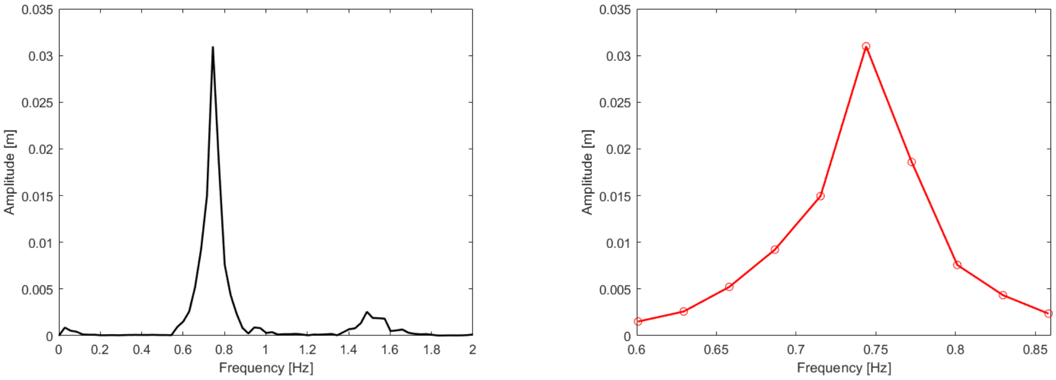


Figure 5.2: Experimental free surface elevation signal, measured in the far field of the domain under deep water conditions.

The full FFT spectrum of the signal (Figure 5.3a) shows that the energy is predominantly concentrated in a narrow band around $f_p = 0.75\,\mathrm{Hz}$, with minor contributions from higher or lower frequencies. To filter the linear spectrum, the range 0.60–0.90 Hz was selected, effectively capturing the first-order harmonics while excluding second-ordercontributions, namely subharmonic and superharmonic components (Figure 5.3b). However, the definition of an appropriate frequency window is not always trivial; in this case, the spectrum was sufficiently narrow-banded to allow a clear separation between linear and higher-order components.



(a) Full FFT spectrum of the original signal.

(b) Filtered spectrum in the range 0.60–0.90 Hz.

Figure 5.3: Full FFT spectrum and the filtered band used for signal reconstruction.

5.2.1 Numerical Setup

In the DualSPHysics numerical model, the free surface elevation was evaluated at a distance of $1.5\,\lambda_p$ from the virtual wavemaker, where λ_p is the peak wavelength associated with the target spectrum. This placement ensures that the observed signal is not affected by transient components or near field disturbances. The overall dimensions of the numerical wave tank were also defined as a function of λ_p, following the setup described in Chapter 4.

The bidimensional numerical domain was modeled using a uniform particle spacing of $dp = 0.015$ m, with a smoothing length ratio of `coefh` $= h/dp = 1.3$, and an artificial speed of sound coefficient set to `coefsound` $= 20$, consistent with the settings adopted in Chapter 4. The simulation duration was set to 50 s, exceeding the original experimental window of 35 s, to ensure the complete development of the wave group at the control point. At each iteration the free-surface elevation was sampled using `MeasureTool` and analyzed via FFT to assess the convergence of the input calibration and the emergence of nonlinear components.

5.2.2 Spectral Tuning Procedure

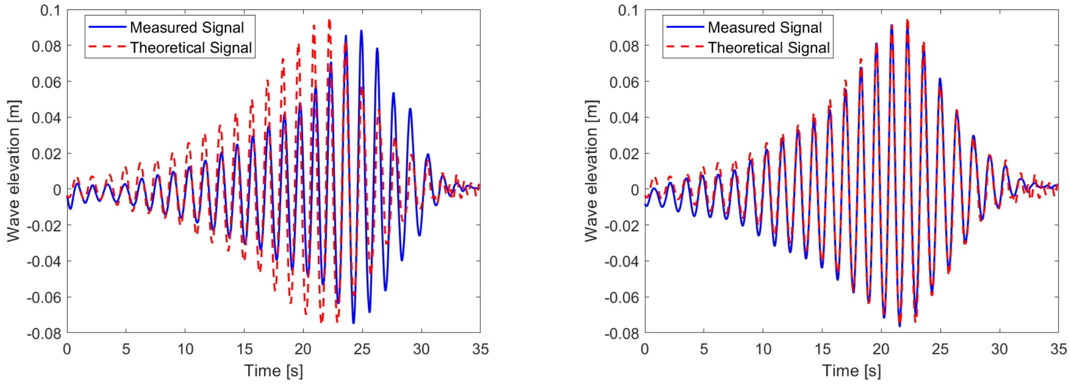
The filtered spectrum, described in the previous section, was used to reconstruct the linear input signal by extracting the amplitudes and phases of the first-order harmonics. These components were then propagated backward from the control point to the wavemaker using the linear dispersion relation, and the piston motion was computed via the Biesel transfer function. The calibration procedure was then carried out iteratively, following the strategy outlined in subsection 3.3.3.

Convergence was considered achieved once the relative amplitude error fell below 5% and the phase error remained within $\pi/18$ radians for the dominant spectral components in the target band. In the presented case, convergence was reached within six iterations, confirming the robustness of the method even under broadband and steep wave conditions.

The final calibrated output signal, along with the results of each iterations, are analyzed in the next section to evaluate the effectiveness of the calibration procedure, with an additional focus on the nonlinear spectral evolution and the potential emergence of higher-order components: their presence in the simulation would demonstrate the SPH model ability to naturally reproduce second-order interactions.

5.2.3 Convergence Analysis and Spectral Validation

The final simulation outputs are shown in Figure 5.4a and Figure 5.4b, which compare the time histories of the numerical and target signals at the first and sixth iteration, respectively. The initial result shows noticeable deviations in both phase and amplitude, especially near the focal region. These discrepancies are progressively corrected through the iterative procedure, and by the sixth iteration, the numerical signal closely matches the theoretical one, especially around the main crest. The overall envelope of the wave group is well captured; minor residual mismatches remain in the trailing portion of the signal, likely due to numerical dispersion after the focusing point.



(a) First iteration

(b) Sixth iteration

Figure 5.4: Comparison between theoretical and measured wave elevation at the control point. The agreement improves significantly after six iterations.

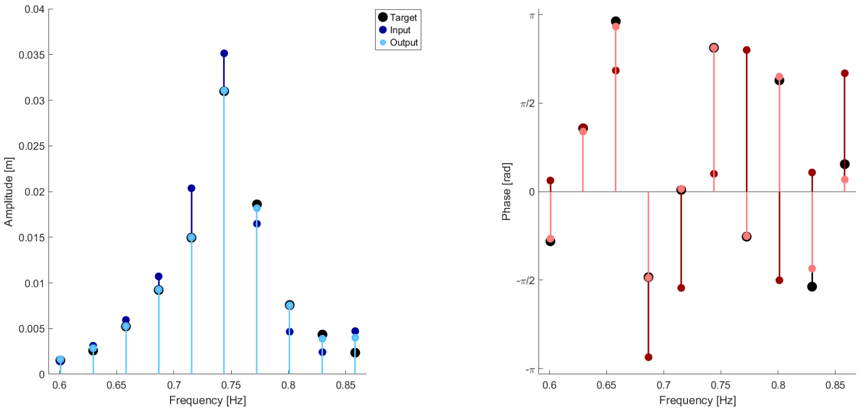


Figure 5.5: Amplitude and phase comparison between input, output, and target at the last iteration for the main frequency components.

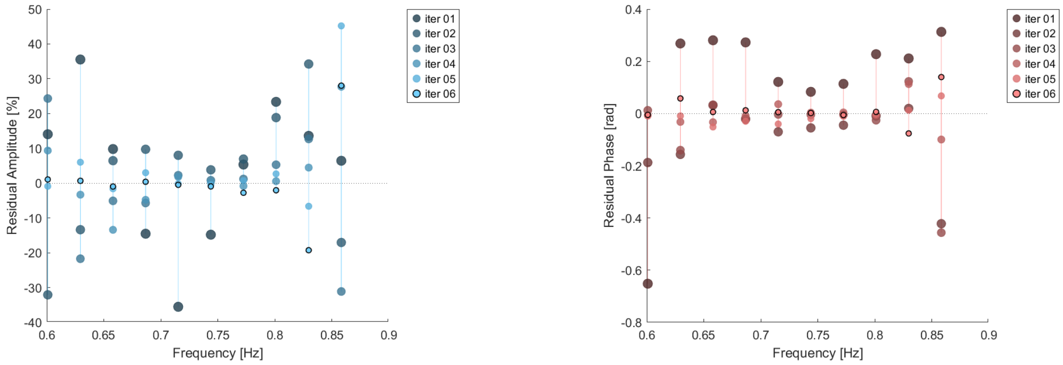
In the frequency domain, the calibration proves effective in reconstructing the desired linear spectrum. As shown in Figure 5.5, the numerical and target spectra agree well around the peak frequency ($f_p \approx 0.75\,\text{Hz}$), confirming that the iterative process successfully tuned both amplitude and phase of the first-order components.

Figure 5.6a and Figure 5.6b display the evolution of amplitude and phase residuals during the iterative process. The amplitude error gradually decreases over the iterations and falls below 3% for all relevant components by the sixth iteration. Similarly, phase errors converge to near-zero values, confirming validity of the spectral correction algorithm. A slightly slower convergence was observed for higher-frequency components, likely due to the method greater sensitivity to low amplitudes.

Beyond the linear spectrum, the SPH simulation also reproduced visible second-order components. These include both superharmonic peaks near 1.5 Hz and low-frequency subharmonics below 0.5 Hz. Their appearance is a strong evidence of the SPH model ability to capture nonlinear energy transfers intrinsically, without requiring them to be prescribed in the input signal.

The comparison in Figure 5.7 shows qualitative agreement with the experimental signal, although some differences remain. In particular, the amplitudes in the subharmonic region are slightly overestimated in the simulation. This may be attributed to the high sensitivity of the model to components with very low amplitude, which could lead to an overestimation of low-frequency energy. Similar effects have been reported in previous studies [5], and could also be influenced by the spatial resolution or the choice of smoothing length.

Overall, the results demonstrate that the SPH model, combined with the proposed calibration strategy, can accurately reproduce both linear and nonlinear



(a) Residual amplitude error. **(b)** Residual phase error.

Figure 5.6: Residual amplitude and phase errors over successive iterations of the calibration process.

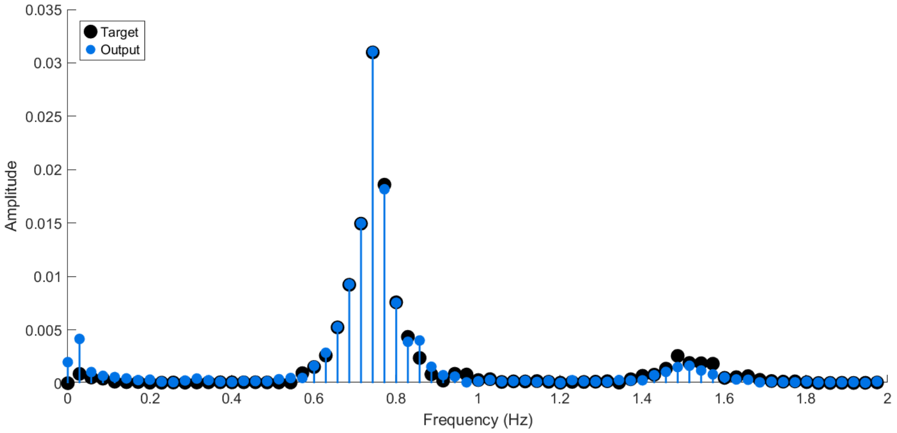


Figure 5.7: Comparison between target and simulated FFT spectra at the sixth iteration.

wave characteristics. The method proved effective in reconstructing a focused wave profile with high fidelity in both time and frequency domains. Moreover, the spontaneous emergence of second-order components confirms the physical realism of the numerical approach.

Chapter 6

Conclusions

The present study aimed to model free-surface water waves in a numerical tank, with a specific focus on reproducing a target wave with given characteristics at a selected point within the domain, starting solely from its linearized spectrum definition. As stated at the beginning of this work, wave amplitude decays between the generation and control points, due to both physical and numerical dissipation [6, 7, 3]. The analysis therefore began by evaluating this phenomenon under baseline conditions, conducting a parametric study on key SPH parameters in order to achieve more accurate reproduction.

In particular, the results concerning spatial resolution dp and kernel smoothing length h, which have been identified as key parameters influencing numerical dissipation [6, 3], were consistent with recommendations reported in literature [25, 30, 6, 26]. The final choice of the parameters adopted in the subsequent simulations – $dp = 0.015$ m, `coefsound`= $\frac{c_s}{u_{ref}} = 20$, `coefh`= $\frac{h}{dp} = 1.3$, laminar viscosity treatment – represented a trade–off between results fidelity and computational cost.

After tuning key parameters to manage numerical dissipation, the focus shifted to adjusting the input signal for accurate wave generation. The implemented calibration procedure, inspired by the work of [5], was first tested on regular waves, using amplitude as the only control variable. This yielded promising results, successfully reproducing the target surface elevation at the control point in no more than two iterations, with a relative error of 5%, or even lower for less steep waves.

Given the successful application of the technique on regular waves, it was then extended to irregular waves, including phase control as well. The test, performed on a wave measured under experimental conditions [45], confirmed that the model is capable of reproducing the full wave spectrum even when initialized with only linear components, including second–order effects such as subharmonics and superharmonics.

The calibration procedure enabled high accuracy reconstruction of the linearized spectrum in both phase and amplitude, especially for the dominant components, which where reproduced with amplitude errors below 3% and phase errors within $\frac{\pi}{18}$ radians in fewer that six iterations.

However, some limitations were observed, particularly in reproducing low-frequency harmonic components and components with very low amplitudes. This may be attributed to the model's limited sensitivity to weak signals and could potentially be improved by implementing active control of second-order effects – a feature that is, in fact, available within the DualSPHysics framework [38].

Bibliography

[1] David Le Touzé and Andrea Colagrossi. «Smoothed Particle Hydrodynamics for free-surface and multiphase flows: a review». In: *Reports on Progress in Physics* (2025) (cit. on pp. 1, 5, 6, 46).

[2] James A Sethian and Peter Smereka. «Level set methods for fluid interfaces». In: *Annual review of fluid mechanics* 35.1 (2003), pp. 341–372 (cit. on p. 1).

[3] Matteo Antuono, Andrea Colagrossi, Salvatore Marrone, and Claudio Lugni. «Propagation of gravity waves through an SPH scheme with numerical diffusive terms». In: *Computer Physics Communications* 182.4 (2011), pp. 866–877 (cit. on pp. 1, 2, 62).

[4] Robert A Dalrymple and Benedict D Rogers. «Numerical modeling of water waves with the SPH method». In: *Coastal engineering* 53.2-3 (2006), pp. 141–147 (cit. on pp. 1, 34).

[5] Thomas Vyzikas, Dimitris Stagonas, Eugeny Buldakov, and Deborah Greaves. «The evolution of free and bound waves during dispersive focusing in a numerical and physical flume». In: *Coastal Engineering* 132 (2018), pp. 95–109 (cit. on pp. 2, 3, 27, 28, 30, 60, 62).

[6] Andrea Colagrossi, Antonio Souto-Iglesias, Matteo Antuono, and Salvatore Marrone. «Smoothed-particle-hydrodynamics modeling of dissipation mechanisms in gravity waves». In: *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* 87.2 (2013), p. 023302 (cit. on pp. 2, 40, 48, 62).

[7] V Zago, LJ Schulze, G Bilotta, N Almashan, and RA Dalrymple. «Overcoming excessive numerical dissipation in SPH modeling of water waves». In: *Coastal Engineering* 170 (2021), p. 104018 (cit. on pp. 2, 62).

[8] Dimitris Stagonas, Eugeny Buldakov, and RR Simons. «Focusing unidirectional wave groups on finite water depth with and without currents». In: International Conference on Coastal Engineering (ICCE). 2014 (cit. on pp. 2, 28, 30).

[9] Pengzhi Lin. *Numerical modeling of water waves.* CRC Press, 2008 (cit. on pp. 4, 5).

[10] Stanislaw R Massel. «Extended refraction-diffraction equation for surface waves». In: *Coastal Engineering* 19.1-2 (1993), pp. 97–126 (cit. on p. 4).

[11] Ge Wei, James T Kirby, Stephan T Grilli, and Ravishankar Subramanya. «A fully nonlinear Boussinesq model for surface waves. Part 1. Highly nonlinear unsteady waves». In: *Journal of fluid mechanics* 294 (1995), pp. 71–92 (cit. on p. 4).

[12] Patrick J Lynett. «Nearshore wave modeling with high-order Boussinesq-type equations». In: *Journal of waterway, port, coastal, and ocean engineering* 132.5 (2006), pp. 348–357 (cit. on p. 4).

[13] DUAN Wenyang. «Taylor expansion boundary element method for floating body hydrodynamics». In: *Proceedings of the 27th international workshop on water waves and floating bodies. Copenhagen, Denmark.* 2012 (cit. on p. 4).

[14] Digambar Patil and Sachin Kadam. «Basics of computational fluid dynamics: An overview». In: *IOP conference series: Earth and environmental science*. Vol. 1130. 1. IOP Publishing. 2023, p. 012042 (cit. on p. 5).

[15] Q.W. Ma. «Meshless local Petrov–Galerkin method for two-dimensional nonlinear water wave problems». In: *Journal of Computational Physics* 205.2 (2005), pp. 611–625 (cit. on pp. 5, 6).

[16] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. «Meshless methods: An overview and recent developments». In: *Computer methods in applied mechanics and engineering* 139 (1996), pp. 3–47 (cit. on pp. 5–7, 9).

[17] Damien Violeau and Benedict D Rogers. «Smoothed particle hydrodynamics (SPH) for free-surface flows: past, present and future». In: *Journal of Hydraulic Research* 54.1 (2016), pp. 1–26 (cit. on pp. 5, 8, 10, 11, 14, 15).

[18] MB Liu and GR2593940 Liu. «Smoothed particle hydrodynamics (SPH): an overview and recent developments». In: *Archives of computational methods in engineering* 17 (2010), pp. 25–76 (cit. on pp. 5, 8, 10, 11, 13, 14).

[19] Qinghang Cai, Ronghua Chen, Kailun Guo, Wenxi Tian, Suizheng Qiu, and GH Su. «An enhanced moving particle semi-implicit method for simulation of incompressible fluid flow and fluid-structure interaction». In: *Computers & Mathematics with Applications* 145 (2023), pp. 41–57 (cit. on pp. 6, 7).

[20] Shaofan Li and Wing Kam Liu. «Meshfree and particle methods and their applications». In: *Appl. Mech. Rev.* 55.1 (2002), pp. 1–34 (cit. on pp. 6, 7).

[21] Nan-Jing Wu, Ting-Kuei Tsay, and DL Young. «Meshless numerical simulation for fully nonlinear water waves». In: *International journal for numerical methods in fluids* 50.2 (2006), pp. 219–234 (cit. on p. 7).

[22] Leon B Lucy. «A numerical approach to the testing of the fission hypothesis». In: *Astronomical Journal, vol. 82, Dec. 1977, p. 1013-1024.* 82 (1977), pp. 1013–1024 (cit. on p. 8).

[23] Robert A Gingold and Joseph J Monaghan. «Smoothed particle hydrodynamics: theory and application to non-spherical stars». In: *Monthly notices of the royal astronomical society* 181.3 (1977), pp. 375–389 (cit. on p. 8).

[24] Leonardo Di G Sigalotti, Jaime Klapp, and Moncho Gómez Gesteira. «The mathematics of smoothed particle hydrodynamics (SPH) consistency». In: *Frontiers in Applied Mathematics and Statistics* 7 (2021), p. 797455 (cit. on pp. 9, 13, 14).

[25] Jose M Domínguez et al. «DualSPHysics: from fluid dynamics to multiphysics problems». In: *Computational Particle Mechanics* 9.5 (2022), pp. 867–895 (cit. on pp. 12, 15–17, 19, 40, 46, 48, 52, 62).

[26] Joe J Monaghan. «Simulating free surface flows with SPH». In: *Journal of computational physics* 110.2 (1994), pp. 399–406 (cit. on pp. 12, 48, 62).

[27] Nathan J Quinlan, Mihai Basa, and Martin Lastiwka. «Truncation error in mesh-free particle methods». In: *International Journal for Numerical Methods in Engineering* 66.13 (2006), pp. 2064–2085 (cit. on pp. 13, 14).

[28] JK Chen, JE Beraun, and CJ Jih. «An improvement for tensile instability in smoothed particle hydrodynamics». In: *Computational Mechanics* 23.4 (1999), pp. 279–287 (cit. on p. 14).

[29] Damien Violeau and Reza Issa. «Numerical modelling of complex turbulent free-surface flows with the SPH method: an overview». In: *International Journal for Numerical Methods in Fluids* 53.2 (2007), pp. 277–304 (cit. on pp. 14, 15).

[30] Alejandro J.C. Crespo, José M. Domínguez, Benedict D. Rogers, Moncho Gómez-Gesteira, Sam Longshaw, Ricardo Canelas, Renato Vacondio, Alejandro Barreiro, and Oscar García-Feal. «DualSPHysics: Open-source parallel CFD solver based on Smoothed Particle Hydrodynamics (SPH)». In: *Computer Physics Communications* 187 (2015), pp. 204–216 (cit. on pp. 16, 17, 46, 62).

[31] Aaron English, JM Domínguez, Renato Vacondio, AJC Crespo, PK Stansby, SJ Lind, Luca Chiapponi, and M Gómez-Gesteira. «Modified dynamic boundary conditions (mDBC) for general-purpose smoothed particle hydrodynamics (SPH): Application to tank sloshing, dam break and fish pass problems». In: *Computational Particle Mechanics* 9.5 (2022), pp. 1–15 (cit. on pp. 16, 34).

[32] Steven J Lind, Rui Xu, Peter K Stansby, and Benedict D Rogers. «Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves». In: *Journal of Computational Physics* 231.4 (2012), pp. 1499–1523 (cit. on pp. 17, 34).

[33] *SPHERIC Benchmark Test Cases*. `https://spheric-sph.org/benchmarks`. Accessed: June 2025 (cit. on p. 18).

[34] DualSPHysics Team. *DualSPHysics Project*. `https://dual.sphysics.org/`. Accessed: June 2025. 2025 (cit. on p. 18).

[35] Charles L Mader. *Numerical modeling of water waves*. CRC press, 2004 (cit. on pp. 21, 28, 35).

[36] Andi Trimulyono and Hirotada Hashimoto. «Experimental validation of smoothed particle hydrodynamics on generation and propagation of water waves». In: *Journal of Marine Science and Engineering* 7.1 (2019), p. 17 (cit. on pp. 22, 50).

[37] Peter Frigaard, Michael Høgedal, and Morten Christensen. «Wave generation theory». In: (1993) (cit. on pp. 22–26).

[38] Corrado Altomare, Jose M Domínguez, AJC Crespo, J González-Cao, T Suzuki, M Gómez-Gesteira, and P Troch. «Long-crested wave generation and absorption for SPH-based DualSPHysics model». In: *Coastal Engineering* 127 (2017), pp. 37–54 (cit. on pp. 24, 33, 50, 51, 63).

[39] Corrado Altomare, Bonaventura Tagliafierro, Jose M Dominguez, Tomohiro Suzuki, and Giacomo Viccione. «Improved relaxation zone method in SPH-based model for coastal engineering applications». In: *Applied Ocean Research* 81 (2018), pp. 15–33 (cit. on pp. 25, 33, 50, 51).

[40] Holger Wendland. «Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree». In: *Advances in computational Mathematics* 4.1 (1995), pp. 389–396 (cit. on p. 34).

[41] Georgios Fourtakas, Jose M Dominguez, Renato Vacondio, and Benedict D Rogers. «Local uniform stencil (LUST) boundary condition for arbitrary 3-D boundaries in parallel smoothed particle hydrodynamics (SPH) models». In: *Computers & Fluids* 190 (2019), pp. 346–361 (cit. on p. 34).

[42] Benedict Leimkuhler. «Molecular dynamics». In: *Encyclopedia of Applied and Computational Mathematics*. Springer, 2015, pp. 931–940 (cit. on p. 34).

[43] Wojciech Sulisz and Maciej Paprota. «Modeling of the propagation of transient waves of moderate steepness». In: *Applied Ocean Research* 26.3-4 (2004), pp. 137–146 (cit. on p. 35).

[44] Iskander Abroug, Nizar Abcha, Armelle Jarno, and François Marin. «Laboratory study of non-linear wave–wave interactions of extreme focused waves in the nearshore zone». In: *Natural Hazards and Earth System Sciences* 20.12 (2020), pp. 3279–3291 (cit. on p. 35).

[45] E Pasta, F Carapellese, B Paduano, Mattia Glorioso, G Papini, N Faedo, G Mattiazzo, and P Lomonaco. «Preliminary results of the experimental assessment of the SWINGO wave energy system». In: *Innovations in Renewable Energies Offshore*. CRC Press, 2024, pp. 183–191 (cit. on pp. 57, 62).