Journal of
Marine Science and Engineering

MDPI

Article

Simulations of Wave–Structure Interactions in Incompressible SPH Using Modified Dynamic Boundary Conditions

Marco Simone [1,*], Giovanni Cannata [1] and Georgios Fourtakas [2]

[1] Department of Civil, Constructional and Environmental Engineering, Sapienza University of Rome, 00185 Roma, Italy; giovanni.cannata@uniroma1.it
[2] School of Engineering, University of Manchester, Manchester M13 9PL, UK; georgios.fourtakas@manchester.ac.uk
* Correspondence: marco.simone@uniroma1.it

Abstract

The simulation of free-surface flows in hydraulic engineering presents several challenges due to the intrinsic complexity of modeling a fluid that continuously deforms and evolves over time. In this context, the Smoothed Particle Hydrodynamics (SPH) method, a Lagrangian approach that represents the fluid as a set of moving particles, is better suited than traditional grid-based methods. However, compared to the latter, the SPH method also exhibits certain drawbacks, including increased difficulty in handling wall boundary conditions and a higher computational cost. This work proposes an original wall boundary treatment technique that, to the best of our knowledge, is applied in the Incompressible SPH (ISPH) approach for the first time. The proposed treatment relies on boundary particles external to the fluid and internal extrapolation points, where pressure is computed to enforce Neumann boundary conditions in a consistent manner. During the development of this technique, several intrinsic advantages over existing methods in the literature are identified. A series of numerical benchmarks are conducted to verify the validity of the proposed ISPH model. Numerical results show good agreement with experimental data reported in the literature, confirming the effectiveness of the proposed numerical model in reproducing free-surface flow hydraulic phenomena.

Keywords: boundary conditions; smoothed particle hydrodynamics; incompressible SPH; modified Dynamic Boundary Conditions; wave–structure interactions; free-surface flows

1. Introduction

Smoothed Particle Hydrodynamics (SPH) [1–3] is a meshless, Lagrangian numerical scheme applicable to fluid dynamics simulations. It is founded on two primary stages: a continuum approximation (kernel approximation), where a physical quantity is calculated via a domain of influence and a smoothing function, and a discrete approximation (particle approximation), in which the fluid volume is discretized into a set of nodes (or particles). The standard SPH formulation for hydrodynamics, known as Weakly Compressible SPH (WCSPH), employs an equation of state to explicitly calculate pressure as a function of density. While density is typically constrained to within 1% of a reference value, this approach often induces acoustic waves due to fluid compressibility, potentially leading to non-physical pressure fluctuations and numerical instabilities. Furthermore, WCSPH requires a small time step constrained by the numerical speed of sound, which must be set at least ten times higher than the maximum fluid velocity. An alternative approach

Academic Editor: Constantine Michailides

Received: 10 April 2026
Revised: 30 April 2026
Accepted: 2 May 2026
Published: 5 May 2026

aimed at overcoming the limitations of WCSPH is Incompressible SPH (ISPH). In this framework, density is no longer a variable of the problem and pressure is determined by solving a Pressure Poisson Equation (PPE). Many numerical studies suggest that ISPH yields noise-free pressure fields and accurate force predictions [4–7]. Additionally, because it lacks an explicit dependence on the speed of sound, larger time steps can be utilized. However, the primary drawback of this approach is the computational overhead associated with solving a large linear system, which, in the case of free-surface flows, also requires the explicit identification of the free surface at every time step.

Among the primary challenges associated with the SPH method is the treatment of boundary conditions [8]. Over the years, various techniques have been proposed to model solid boundaries, which are categorized here into four main groups. The repulsive forces technique was the first to be employed for simulating hydraulic problems within the SPH framework [9]. In this approach, boundaries are discretized using fixed particles that exert repulsive forces on the fluid particles. These forces are formulated to be inversely proportional to the distance between the interacting particles. Another widely adopted technique for solid wall treatment involves the use of so-called dummy particles (or ghost particles) [10–12]. The underlying concept is to represent the solid region through multiple layers of fixed particles, to which physical quantities such as velocity and pressure are assigned. This ensures that the support domain of fluid particles near the wall is adequately filled, preventing kernel truncation. A third technique for the treatment of solid boundaries is the semi-analytical formulation [13–16]. This approach involves decomposing the integral obtained from the kernel approximation into two distinct parts: one including the fluid domain and another covering the boundary region. The first integral is solved using the conventional particle approximation, while the second is evaluated directly without discretization. The final technique discussed herein is the mirror particles method [17]. For each fluid particle in the vicinity of the wall, a "mirror" particle is generated symmetrically with respect to the solid boundary. Velocity and pressure values are assigned to these particles to ensure the rigorous satisfaction of the prescribed boundary conditions. In addition to these techniques, there is the modified Dynamic Boundary Conditions (mDBC) proposed by English et al. [18], which originated within the WCSPH framework as an evolution of the dummy particle method, aiming to overcome its inherent limitations. In this work, this technique is extended to the ISPH context in an original manner for the first time. The proposed technique utilizes layers of fixed particles to discretize solid boundaries, introducing mirror extrapolation points within the fluid and interface points to ensure physical consistency. To implement this new boundary treatment, a computational code was developed in Fortran. To reduce the computational cost, the efficiency of this numerical code was enhanced through parallel computing across multiple processors, utilizing OpenMP directives and the PETSc library [19].

The manuscript is organized as follows: Section 2 recalls the governing equations and the time-stepping scheme; Section 3 details the proposed technique for the treatment of boundary conditions and illustrates how the Pressure Poisson Equation (PPE) system has been adapted; finally, Section 4 presents a series of test cases to validate the proposed numerical scheme.

2. Methodology

Momentum and Continuity Equations and Time-Marching Scheme

In the scheme presented below, the incompressible form of the Navier–Stokes equations is numerically solved,

$$\begin{aligned} \nabla \cdot \boldsymbol{v} &= 0 \\ \frac{D\boldsymbol{v}}{Dt} &= -\frac{1}{\rho}\nabla p + \nu \nabla^2 \boldsymbol{v} + \boldsymbol{f_{ext}} \end{aligned} \tag{1}$$

where $\boldsymbol{v}$ is the velocity, ρ is the density, p is the pressure, ν is the kinematic viscosity, and $\boldsymbol{f_{ext}}$ represents the body forces. Equation (1) is discretized using the Smoothed Particle Hydrodynamics (SPH) method. Specifically, reference is made to the consistent formulations presented by Schwaiger [20] for a generic field f and its first derivative,

$$f(\boldsymbol{r_i}) \simeq \frac{1}{V}\sum_j \frac{m}{\rho} f_j W_{ij} \tag{2}$$

$$\nabla_{\boldsymbol{r_i}} f(\boldsymbol{r_i}) \simeq \sum_j \frac{m}{\rho}(f_j - f_i)\nabla_{\boldsymbol{r_i}} {W_{ij}}^c \tag{3}$$

where the subscript i refers to the interpolating particle and j to the neighboring particles, $\boldsymbol{r}$ is the position, m is the mass, and W is the smoothing function; meanwhile, V denotes a normalization factor and $\nabla_{\boldsymbol{r_i}} {W_{ij}}^c$ represents the corrected gradient of the kernel function. In this work, the Wendland function [21] was selected as the smoothing kernel W. The Laplacian operator is expressed using the formulation proposed by Morris et al. [22],

$$\nabla^2_{r_i} f(r_i) = 2\sum_j \frac{m}{\rho}\frac{f_{ij}}{r_{ij}^2 + \eta^2}\boldsymbol{r}_{ij}\cdot\nabla_{\boldsymbol{r_i}} W_{ij} \tag{4}$$

where, for the sake of brevity, $f_{ij} = f_i - f_j$, $\boldsymbol{r}_{ij} = \boldsymbol{r}_i - \boldsymbol{r}_j$, $\eta = 0.1h$, in which h is the smoothing length.

Time integration is performed using a multi-step projection method [6,17,23]. The first step consists of calculating a predicted velocity $\boldsymbol{v}^*$ by solving the momentum balance equation, neglecting the contribution of the pressure gradient,

$$\boldsymbol{v}^* = \boldsymbol{v}^t + \Delta t\left[\nabla\cdot\left(\nu_E \nabla \boldsymbol{v}^t\right) + \boldsymbol{g}\right] \tag{5}$$

where the superscript t denotes the value at the current time step, $\nu_E = \nu + \nu_T$ is the effective viscosity, ν_T represents the turbulent viscosity computed via the $k-\varepsilon$ model [24], and $\boldsymbol{g}$ is the gravitational acceleration. The pressure is obtained by solving a Poisson equation of the form

$$\nabla^2 p^{t+\Delta t} = \frac{\rho}{\Delta t}\nabla\cdot\boldsymbol{v}^* \tag{6}$$

where the superscript $t+\Delta t$ denotes the value at the new time step. Finally, the velocity field is updated as follows:

$$\boldsymbol{v}^{t+\Delta t} = \boldsymbol{v}^* - \frac{\Delta t}{\rho}\nabla p^{t+\Delta t} \tag{7}$$

Once the velocity at the new time step is known, the particle positions can be advanced in time using a second-order integration scheme,

$$r^{t+\Delta t} = r^t + \Delta t\left(\frac{v^{t+\Delta t} + v^t}{2}\right) \tag{8}$$

An analysis conducted by Xu et al. [25] has shown that while this method yields accurate pressure fields, it tends to produce anisotropic particle distributions, which can lead to numerical instabilities as the simulation progresses. To overcome this issue, the technique proposed by Lind et al. [23] is adopted. At each time step, the particles are slightly shifted by a displacement $\delta\boldsymbol{r}_i$, defined as

$$\delta\boldsymbol{r}_i = -C_{shift}h^2\nabla C_i \tag{9}$$

where $C_{shift} \leq 0.5$ is a constant and ∇C_i represents the concentration gradient.

3. Numerical Modelling

3.1. Boundary Condition Treatment

To address solid boundary conditions within the WCSPH framework, English et al. [6] recently proposed the modified Dynamic Boundary Conditions (mDBC) technique. In the mDBC approach, solid boundaries are discretized using several layers of fixed particles, a procedure analogous to the dummy particle method. For each boundary particle, a mirrored extrapolation point (hereafter termed the mirror point) is defined along the normal direction relative to the fluid–solid interface. The density ρ_m and its gradient at these mirror points are computed using the first-order consistent SPH interpolation proposed by Liu and Liu [26], which involves solving a local linear system for each point. Subsequently, the density of the dummy particles is derived from their corresponding mirror points via a first-order truncated Taylor series expansion.

In the present work, the mDBC technique is implemented for the first time within an Incompressible SPH (ISPH) framework to evaluate the pressure and velocity fields required for the momentum balance equation on wall particles. To this end, a novel set of extrapolation points (hereafter referred to as edge points) has been introduced, situated precisely at the interface between the fluid and the solid wall. A schematic representation of this methodology is provided in Figure 1.

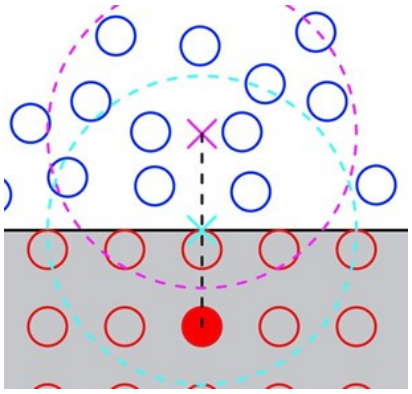

Figure 1. Schematic representation of the proposed technique for the treatment of wall boundary conditions. Blue circles represent fluid particles, red circles represent wall particles, the magenta cross denotes the mirror point, the light-blue cross denotes the edge point, the magenta and light-blue dotted circles represent the support domain radii of magenta and light-blue crosses, respectively.

For the assignment of velocities on dummy particles, a Dirichlet boundary condition is applied. In the simulations performed, two different conditions were used:

- Adherence, in which the wall velocities are opposite to their corresponding fluid counterparts,

$$\boldsymbol{v}_d = -\boldsymbol{v}_m \tag{10}$$

- Free-slip, in which the tangential and normal velocity components at the wall are respectively equal and opposite to those of the corresponding fluid counterparts,

$$\begin{aligned} v_d^t &= v_m^t \\ v_d^n &= -v_m^n \end{aligned} \tag{11}$$

Here, the superscripts t and n denote the tangential and normal components of velocity, while the subscripts d and m refer to dummy particles and mirror points, respectively. The velocities at the mirror points are extrapolated using

$$\boldsymbol{v}_m = \frac{1}{V_m^f} \sum_j^{f \in D_m} \frac{m}{\rho} \boldsymbol{v}_j W_{mj} \tag{12}$$

where the summation extends only over the fluid particles within the support domain D_m of the considered mirror point m, and $V_m^f = \sum_j^{f \in D_m} \frac{m}{\rho} W_{mj}$ is the renormalization factor that accounts solely for the fluid particles in the neighborhood of m. Extending the extrapolation to dummy particles is avoided, as it would necessitate an iterative solution of a linear system, thereby increasing computational costs. The velocities obtained in this way are then used for the computation of the diffusive term in the momentum balance equation. For the assignment of pressure on the dummy particles, a Neumann boundary condition is applied,

$$p_d = p_m - \rho \frac{d_{md} \left(\boldsymbol{v}_w^* - \boldsymbol{v}_w^{t+\Delta t} \right)}{\Delta t} \cdot \boldsymbol{n} \tag{13}$$

where the pressure p_m at the mirror point is obtained from Equation (2), taking into account both the fluid and dummy particles for the extrapolation. This approach was necessary to ensure the correct behavior of the hydrostatic pressure. As shown in the next section, Equation (13) is used as an additional equation of the linear system obtained by discretizing the Poisson equation for the fluid particles. This linear system for pressure is solved by an iterative method.

The proposed technique offers several intrinsic advantages over those previously reported in the literature:

- Compared to repulsive force and dummy particle techniques, a physically consistent imposition of boundary conditions is ensured;
- Compared to semi-analytical techniques, this approach offers greater ease of implementation and can guarantee consistency up to the second order;
- Compared to the mirror particle technique, this approach reduces the difficulties associated with complex geometries and sharp corners; furthermore, the interaction between fluid and solid phases is represented by a heterogeneous area rather than a single particle.

3.2. Additional Measures Adopted for mDBC in ISPH

During the development of the mDBC technique within the ISPH framework, some numerical issues were encountered. These challenges are described below, along with the methodologies employed to resolve them. To calculate the predicted wall velocity $\boldsymbol{v}_w^*$, for use in Equation (13), a multiplicative coefficient, C_{cut}, was introduced. Consequently, Equation (2), when applied to a generic edge point located on the boundary, is reformulated as follows:

$$\boldsymbol{v}_e^* = \frac{1}{V_e} \sum_j \frac{m}{\rho} \boldsymbol{v}_j^* W_{ej} C_{cut} \tag{14}$$

where the subscript e refers to the edge points. The coefficient C_{cut} is defined as

$$C_{cut} = \begin{cases} e^{\frac{1}{\beta dx}\left(\frac{dx}{2} - d_w\right)} \; if \; free - surface \; particle \\ 1 \; if \; internal \; particle \end{cases} \tag{15}$$

where dx denotes the particle diameter, β is an empirical coefficient that is recommended to be set to 0.1 and d_w represents the minimum distance between the different walls. This modification was introduced to prevent anomalous behavior near the boundaries.

Consider, for example, the case of a single particle falling under gravity, as shown in Figure 2. When this particle approaches the bottom boundary, it "sees" (i.e., it is located at a distance from the first dummy particle smaller than the smoothing length) the wall and remains suspended in mid-air. The reason for this behavior lies in the fact that Equation (2), which utilizes the renormalization factor, immediately extrapolated a velocity equal to $g\Delta t$ (where g is the gravitational acceleration constant) onto the edge point at the wall. Consequently, the pressure calculated on the dummy particle through Equation (13) would generate a pressure gradient that counterbalances the fluid particle, preventing it from falling (Figure 2b).

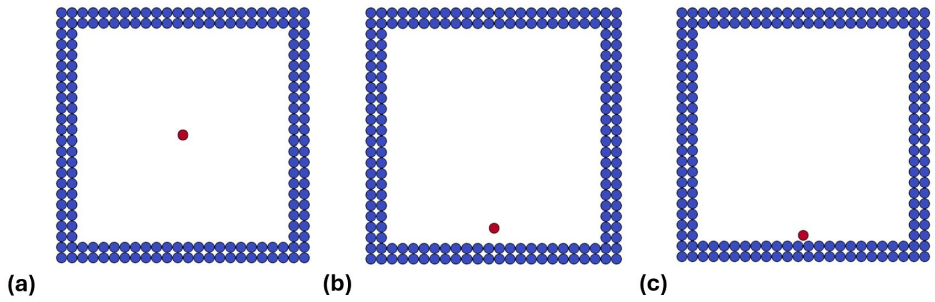

Figure 2. A single particle released under the action of gravity: (**a**) initial position, (**b**) position after 1 s without using C_{cut}, (**c**) position after 1 s with the use of C_{cut}.

It must be noted that, by definition of the smoothing function, the interaction between boundary particles and the single fluid particle begins when the distance between the latter and its final rest position is equal to dx. This implies that setting $\beta = 0.1$ ensures that, for any particle diameter used in the simulation, the value of coefficient C_{cut} at the beginning of the interaction with the single fluid particle is approximately 10^{-5}. Starting from this distance, the exponent of Equation (15) tends to zero continuously within the space of one dx, and coefficient C_{cut} tends to 1. Values of β greater than 0.2 activate the braking effect due to the presence of the wall too early, while values lower than 0.01 could, in theory, excessively reduce the braking effect and lead to error when the particle velocity is very high. It should be noted that the Courant–Friedrichs–Lewy condition used to limit the computational time step ($dt = 0.1dx/v_{max}$, where v_{max} is the maximum particle velocity) guarantees the activation of the boundary effect, even for high particle velocity, as it ensures that the particle travels a distance equal to dx in more than one time step. This implies that, for the method to work, it is sufficient for the beta values to be less than 0.2.

It should be noted, however, that numerical errors due to isolated particles are generally common in the SPH method, which performs robustly when a particle's support domain is full. Furthermore, it must be borne in mind that particles do not physically exist; they are merely computational nodes representing a macroscopic portion of fluid, which vary according to the degree of accuracy.

Another issue introduced by the use of renormalized SPH interpolation (Equation (2)) concerns the extrapolation of pressures along vertical walls. Dummy particles positioned above the free surface are assigned pressures that do not comply with the hydrostatic profile. To prevent this behavior, a variable pointer C_{van} has been introduced for the dummy particles, defined as follows:

- $C_{van} = 1$ when the lower portion of a dummy particle i lies below the centroid of at least one neighboring fluid particle j, specifically when $y_i - 0.5dx < y_j$, where dx denotes the particle diameter and y denotes the vertical coordinate, or when the dummy particle belongs to a horizontal wall;

- $C_{van} = 0$ when the lower portion of a dummy particle i does not lie below any fluid particle j in its neighborhood, specifically when $y_i - 0.5dx > y_j$, and the dummy particle belongs to a vertical wall.

A dummy particle assigned with $C_{van} = 0$ is excluded from the computation of time-updated variables. For a better understanding of how this pointer is utilized, reference should be made to Figure 3, which illustrates the case of a stationary volume of water within a rectangular domain. It can be observed that, when the pointer C_{van} is not employed, the water volume fails to remain in equilibrium.

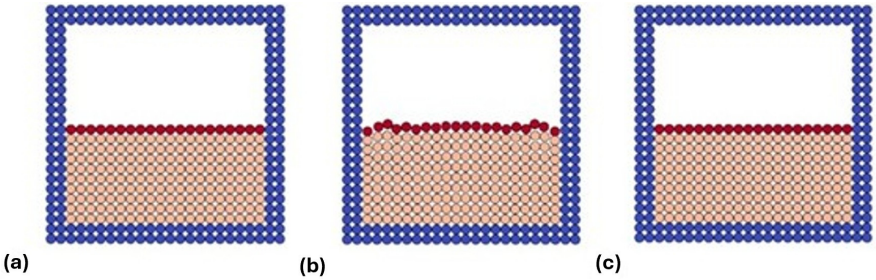

Figure 3. Simulation of a stationary volume of water in a rectangular domain: (**a**) initial position, (**b**) position after 5 s without using C_{van}, (**c**) position after 5 s with the use of C_{van}.

3.3. Generation of Mirror Points

The method used to determine the positioning of the mirror extrapolation points in the presence of complex geometries is discussed below. In the figures presented below, the walls are represented by the black segments r and s, while dummy particles and mirror points are represented by grey circles and crosses, respectively. $\boldsymbol{\alpha}$ is the angle between the wall segments, spanning the portion of the plane exterior to the fluid domain. Following [22], three distinct cases are identified based on the angle formed with the solid walls:

1. The first case (Figure 4) occurs when the angle formed by the wall is $\alpha \geq \frac{3}{2}\pi$. In this situation, three different sub-cases can be distinguished, depending on the region in which the dummy particles are positioned:
   - When the dummy particles are located within the region between s and r′ (or between r and s′), where r′ is the segment r mirrored with respect to s, the extrapolation point is identified along the normal direction relative to segment s;
   - When the dummy particles are positioned in the region bounded by r′ and the extension of r (or between s′ and the extension of s), the extrapolation point is obtained by mirroring along the normal direction to s, starting from the intersection point between r′ and the line parallel to the segment s passing through the particle center (red segment). The same extrapolation point can also be obtained by mirroring, with respect to the vertex, the intersection point between said line and the extension of r;
   - When the dummy particles are located in the region confined between the extensions of r and s, their extrapolation point is obtained by mirroring them with respect to the vertex.
2. The second case (Figure 5) corresponds to an angle $\pi \leq \alpha < \frac{3}{2}\pi$. For dummy particles positioned in the region between the extension of r and r′ (or between the extension of s and s′), the mirror extrapolation points could potentially be located in two ways: either along the normal direction to segment s (or r) or with respect to the vertex. In the present work, the mirror points for dummy particles located in this region are determined by mirroring with respect to the vertex. For dummy particles located in the region between r and the extension of s, their extrapolation points are identified

by mirroring their positions with respect to r. Finally, for dummy particles located in the region between r′ and s′, their positions are mirrored with respect to the vertex.

3. The third case (Figure 6) occurs when the angle formed by the wall is $\alpha < \pi$. In this case, for dummy particles located along the angle bisector formed by the two walls, their mirror extrapolation points are identified by mirroring their positions with respect to the vertex. For dummy particles that do not lie along this bisector, the mirror extrapolation points are determined by mirroring their positions with respect to the nearest segment.

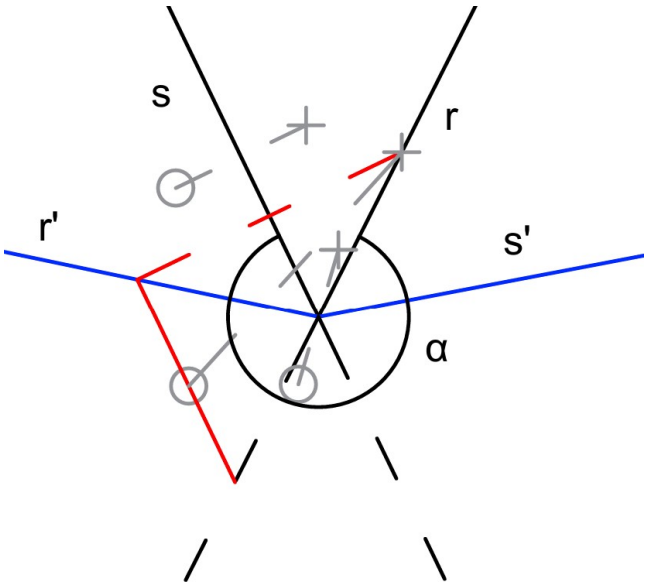

Figure 4. Positioning of mirror extrapolation points for a wall angle $\alpha \geq \frac{3}{2}\pi$.

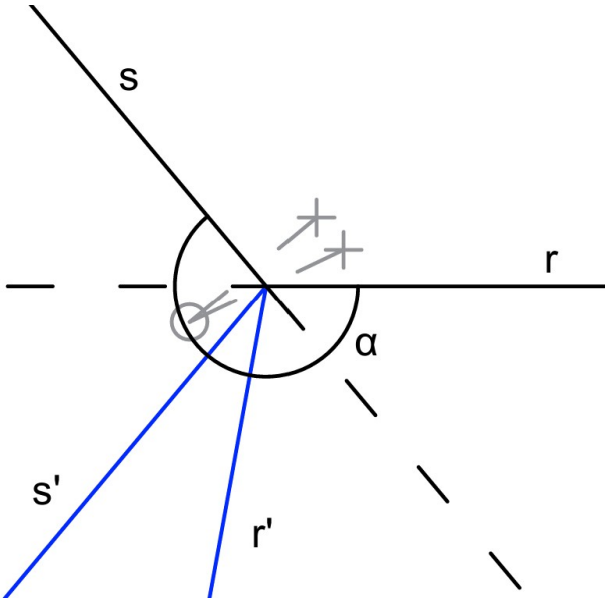

Figure 5. Positioning of mirror extrapolation points for a wall angle $\pi \leq \alpha < \frac{3}{2}\pi$.

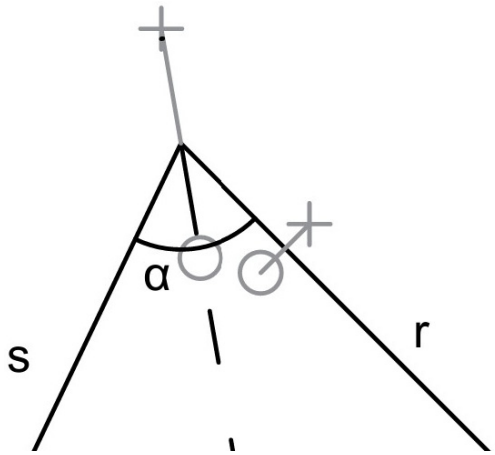

Figure 6. Positioning of mirror extrapolation points for a wall angle $\alpha < \pi$.

3.4. Poisson Solver

The numerical solution of the Poisson Pressure Equation (PPE) system represents one of the key aspects of ISPH schemes. Compared to mesh-based methods, the additional complexity in solving the PPE using the SPH method arises from the fact that the nonzero coefficients of the sparse matrix change at every time step. The number of equations to be solved in this system is $N = n_f + n_d$, corresponding to the sum of the number of fluid particles n_f and dummy particles n_d. These equations can be written in matrix form as a linear system of the type $A\boldsymbol{x} = \boldsymbol{b}$, where A is the $N \times N$ coefficient matrix, $\boldsymbol{x}$ is the vector of unknowns (in this case, the pressures p), and b is the right-hand-side vector. For the fluid particles, the equation to be discretized is Equation (6), which reads as

$$\sum_j 2\frac{m}{\rho}\frac{(\boldsymbol{r}_i - \boldsymbol{r}_j)\cdot\nabla W_{ij}^c}{r_{ij}^2}(p_i - p_j) = -\frac{\rho}{\Delta t}\sum_j \frac{m}{\rho}\left(\boldsymbol{v}_i^* - \boldsymbol{v}_j^*\right)\cdot\nabla W_{ij}^c \tag{16}$$

From Equation (16), it can be observed that for the fluid particles, the coefficients of the matrix A are given by

$$\begin{aligned} A_{ii} &= \sum_j 2\tfrac{m}{\rho}\tfrac{(\boldsymbol{r}_i - \boldsymbol{r}_j)\cdot\nabla W_{ij}^c}{r_{ij}^2} \\ A_{ij} &= -2\tfrac{m}{\rho}\tfrac{(\boldsymbol{r}_i - \boldsymbol{r}_j)\cdot\nabla W_{ij}^c}{r_{ij}^2} \end{aligned} \tag{17}$$

while the components of the right-hand-side vector $\boldsymbol{b}$ are given by

$$b_i = -\frac{\rho}{\Delta t}\sum_j \frac{m}{\rho}\left(\boldsymbol{v}_i^* - \boldsymbol{v}_j^*\right)\cdot\nabla W_{ij}^c \tag{18}$$

To obtain the pressure on the dummy particles, Equation (13) must be solved, where the pressure at the mirror points must in turn be computed using Equation (2). Substituting the latter into Equation (13) and discretizing yields

$$p_d - \frac{1}{V_m}\sum_j \frac{m}{\rho} p_j W_{mj} = -\rho\frac{d_{dm}\left(\boldsymbol{v}_w^* - \boldsymbol{v}_w^{t+\Delta t}\right)}{\Delta t}\cdot\boldsymbol{n} \tag{19}$$

where subscript m refers to the mirror extrapolation point corresponding to the dummy particle d. It can therefore be observed that, for the dummy particles, the coefficients of the matrix A are given by

$$\begin{aligned} A_{dd} &= 1 - \tfrac{1}{V_m}\tfrac{m}{\rho}W_{md} \\ A_{dj} &= -\tfrac{1}{V_m}\tfrac{m}{\rho}W_{mj} \end{aligned} \tag{20}$$

whereas the components of the right-hand-side vector $\boldsymbol{b}$ for these particles are

$$b_d = -\rho \frac{d_{dm}\left(\boldsymbol{v}_w^* - \boldsymbol{v}_w^{t+\Delta t}\right)}{\Delta t} \cdot \boldsymbol{n} \tag{21}$$

It should be noted that, in the case where a fluid particle is identified as belonging to the free surface, its equation is modified to impose the Dirichlet boundary condition $p = 0$. In this case, the coefficients of the matrix A become

$$\begin{aligned} A_{ii} &= 1 \\ A_{ij} &= 0 \end{aligned} \tag{22}$$

while the components of the right-hand-side vector $\boldsymbol{b}$ are

$$b_i = 0 \tag{23}$$

A schematic representation of the coefficient matrix A for the PPE system is shown in Figure 7.

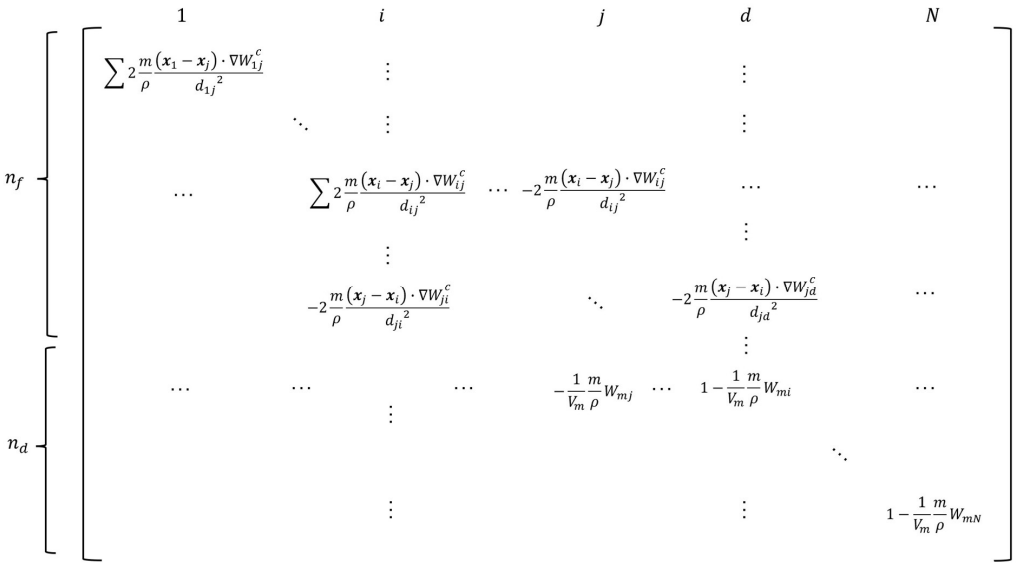

Figure 7. Schematic representation of the coefficient matrix A for the PPE system.

The resulting coefficient matrix is sparse, since it contains zero elements corresponding to particles that do not fall within the influence domain D_i. Moreover, it is non-symmetric because the off-diagonal coefficients differ when the interacting particles are of different types (for example, a fluid and a dummy particle, or an internal fluid particle and a free-surface fluid particle) and because different mirror points are used in Equation (20). Finally, the coefficient matrix is diagonally dominant, as the diagonal terms in Equation (17) correspond to the sum of all the off-diagonal terms. Equations (17), (18) and (20)–(23) characterize the linear system $A\boldsymbol{x} = \boldsymbol{b}$, specifying the governing equations for the three different particle types: Equations (17) and (18) for fluid particles; Equations (20) and (21) for boundary particles; and Equations (22) and (23) for fluid particles belonging to the free surface. This linear system is solved using the PETSc library [19], which provides a suite of solvers for linear equations and is compatible with Fortran-based codes. For the specific simulations presented in this work, the Bi-Conjugate Gradient method was employed, paired with an LU preconditioner. Furthermore, to optimize memory usage and ensure efficient access to coefficients during matrix–vector multiplications, the Compressed Row Storage (CRS) format was adopted.

4. Results

4.1. Dambreak

The dam-break problem is a classic validation benchmark for CFD codes designed to simulate free-surface flows, such as SPH methods [23,27–29]. It involves simulating the collapse of a water column onto a flat bottom, with the objective of evaluating the free-surface evolution and the pressure distribution on the wall opposite the initial water volume. Although the problem is inherently three-dimensional, simulations are often simplified to two dimensions to reduce computational cost while maintaining a meaningful comparison with experimental data. In this work, reference is made to the experimental tests conducted by Lobovsky et al. [30]. The setup consists of a tank with a length of 1610 mm and a height of 600 mm, with a gate positioned at a distance of 600 mm. Four wave gauges for measuring free-surface elevation were placed at distances of 300 mm, 865 mm, 1114 mm, and 1362.5 mm from the initial reservoir wall. To evaluate the pressure, four sensors were considered at heights of 3 mm, 15 mm, 30 mm, and 80 mm from the bottom. A schematic representation of the experimental setup is shown in Figure 8.

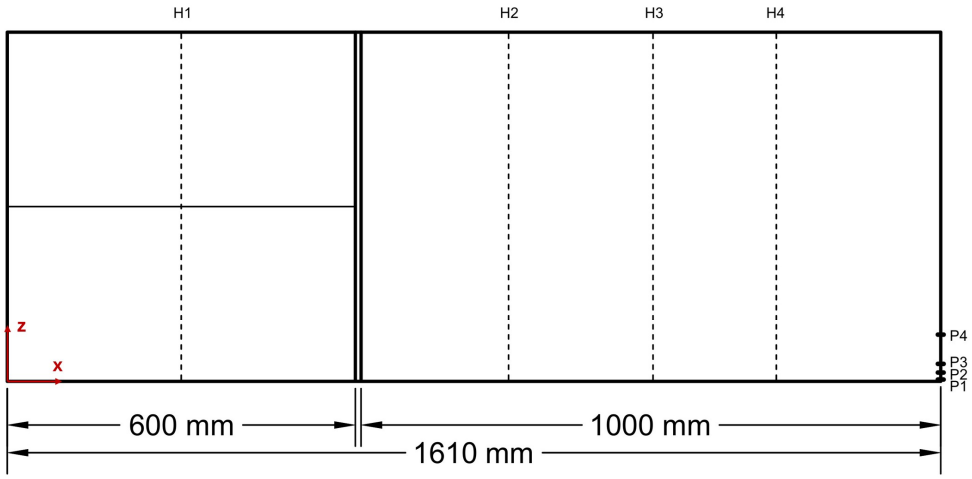

Figure 8. Scheme of the experiments carried out by Lobovsky et al. [20].

The particle diameter was set to $dx = 0.005$ m, corresponding to a total number of 13,512 particles. The total simulation time was 3 s, with a time step of $\Delta t = 5{\cdot}10^{-5}$ s. The simulations were performed on an AMD Ryzen 9 5950X using 8 cores, with a wall-clock time of approximately 6 h. The test develops through four distinct phases, illustrated in Figure 9: the first phase corresponds to the propagation of a dam-break wave front (Figure 9a), followed by the impact and run-up of a portion of water (Figure 9b); subsequently, this fluid volume falls back (dam-break splash-down) (Figure 9c), eventually leading to the formation of two splashes and the propagation of a secondary wave in the direction opposite to the initial one (Figure 9d).

The propagation of the first wave front following the collapse of the water column is reported in Figure 10. The values in the plot have been non-dimensionalized: on the abscissa, the time has been multiplied by $\sqrt{gd}$, where g denotes the gravitational acceleration constant and $d = 0.3$ m is the initial undisturbed water depth; on the ordinate, the front advancement x has been divided by d. Excellent agreement was observed between the numerical simulation results and the experimental data, with a normalized Root Mean Square Error (nRMSE) of 0.0331, confirming the accurate reproduction of the first phase of the phenomenon by the numerical model.

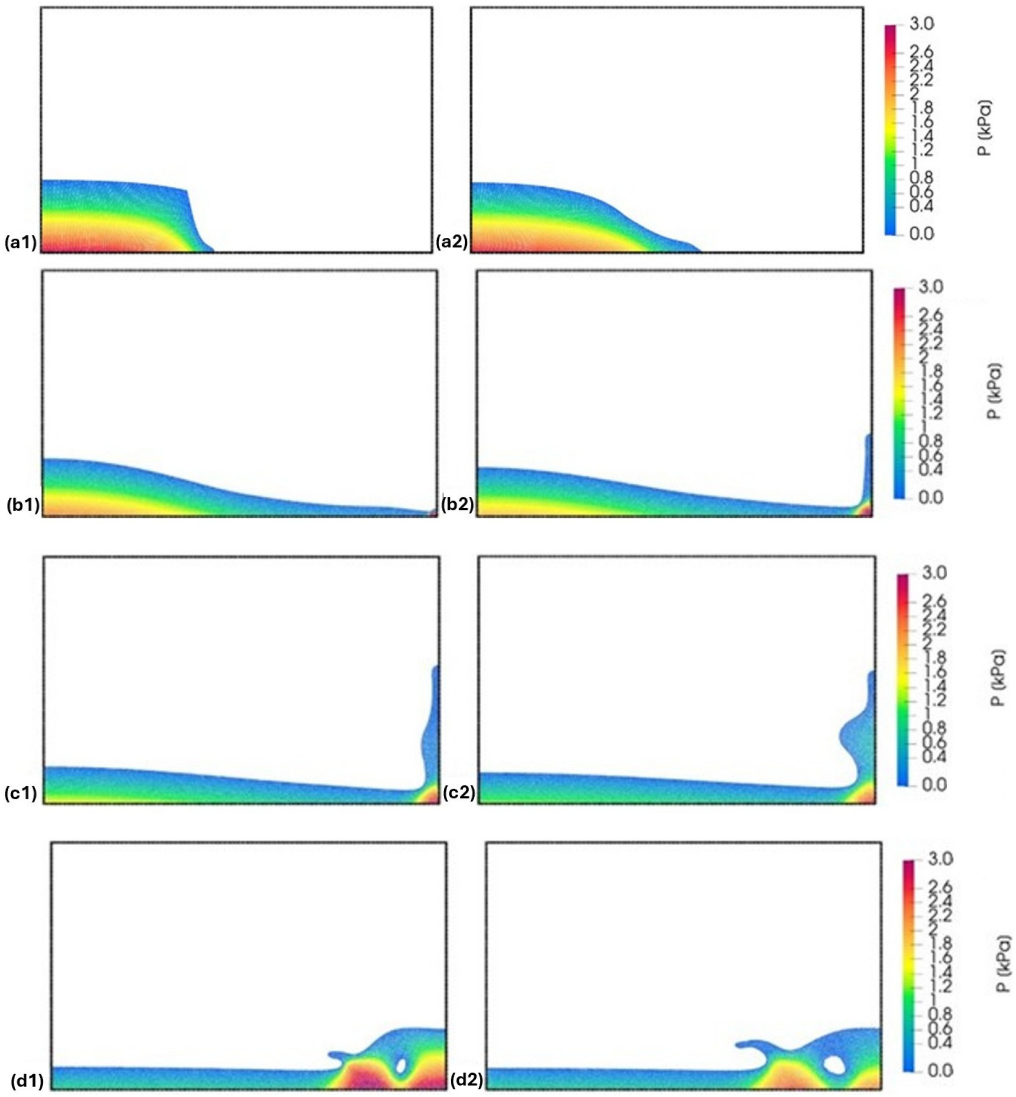

Figure 9. Snapshots at different instants during the simulation of the test: (**a**) propagation of the surge wave generated by the dam break ((**a1**) t = 0.1 s; (**a2**) t = 0.2 s); (**b**) impact and upward motion of a portion of water ((**b1**) t = 0.43 s; (**b2**) t = 0.55 s; (**c**) fall back of the fluid portion ((**c1**) t = 0.73 s; (**c2**) t = 0.85 s); (**d**) generation of splashes and propagation of the secondary wave ((**d1**) t = 1.1 s; (**d2**) t = 1.15 s).

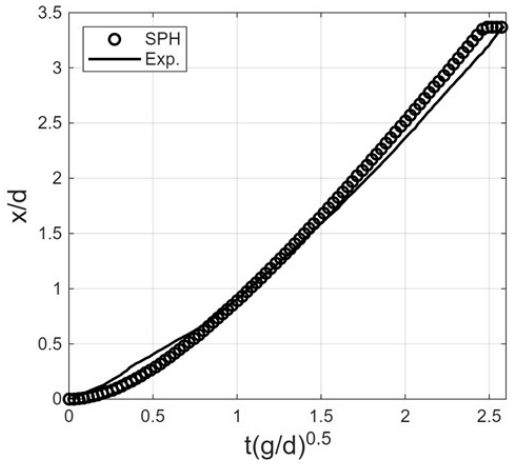

Figure 10. Comparison between numerical and experimental results for the propagation of the wave front generated by the dam break.

Figure 11 illustrates the time evolution of the free-surface elevation η at the four gauge locations. Once again, the quantities have been non-dimensionalized.

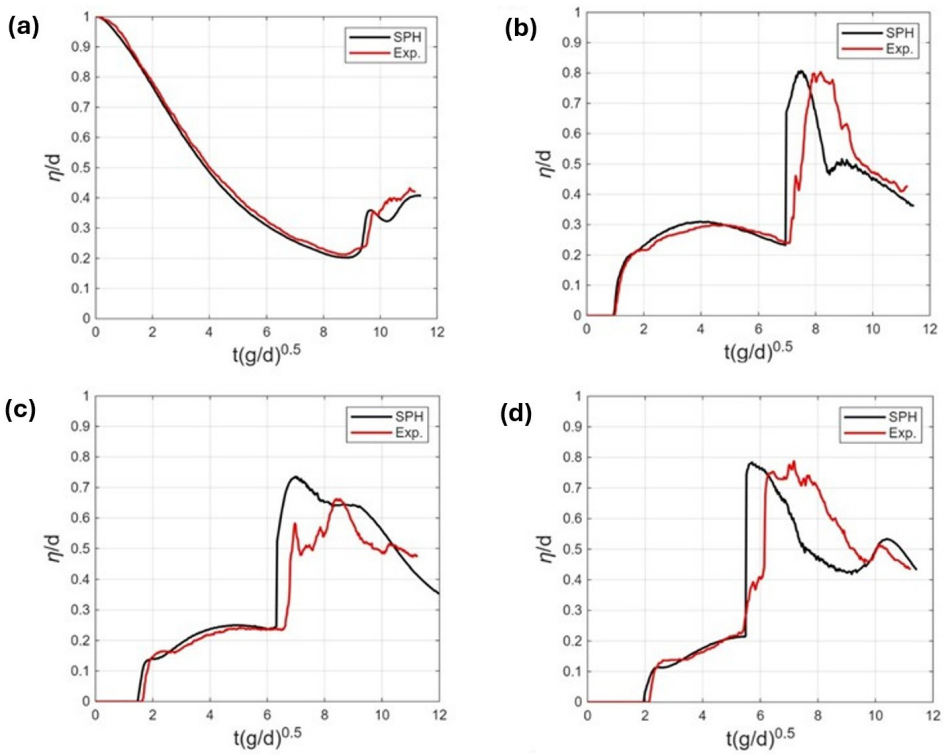

Figure 11. Temporal evolution of free-surface elevation at different locations within the domain: (**a**) $H_1 = 300$ mm; (**b**) $H_2 = 865$ mm; (**c**) $H_3 = 1114$ mm; (**d**) $H_4 = 1365.5$ mm.

The overall behavior is accurately captured; however, while the first stage, corresponding to the initial wave propagation, shows near-perfect agreement, with an nRMSE of 0.0263, 0.0551, 0.085, and 0.074 for H_1, H_2, H_3 and H_4, respectively, during the phase associated with the secondary wave propagation, the numerical results exhibit a slightly earlier response compared to the experimental data, with an nRMSE of 0.0288, 0.1455, 0.1415, and 0.161 for H_1, H_2, H_3 and H_4. The main difficulty in numerically reproducing the free-surface elevation and position associated with this secondary wave is due to its extreme irregularity and low repeatability. Indeed, as explicitly stated by the authors of the experimental measurements, during the laboratory test campaign, for each trial of the same dam-break test, the impact of the fluid with the wall generated a secondary wave characterized by significant differences in both the shape and celerity of the wave front. This makes it particularly challenging to numerically replicate the free-surface elevation trend associated with this wave. This difficulty is highlighted in the plots in Figure 12, which show a comparison between the experimental data in [30] produced by the secondary wave and those obtained by various SPH models presented in the literature. As seen in the figure, the results obtained with the proposed model better agree with the experimental data than those obtained by [31–33], except for a short interval in the plot relating to sensor H4 (Figure 12d), where the proposed model differs more significantly from the experimental results.

Figure 13 shows the time history of the pressures at the four sensor locations. Again, the quantities are presented in non-dimensional form. No filtering has been applied to the pressure measurements.

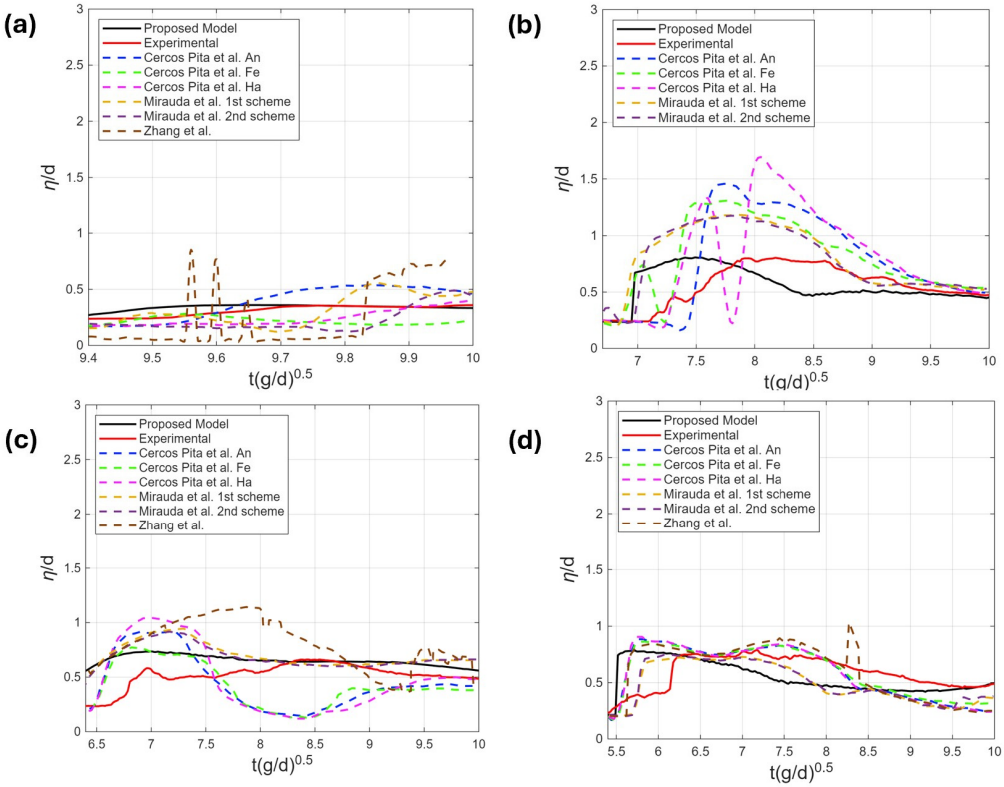

Figure 12. Comparison between experimental data and different numerical results for the temporal evolution of the free-surface elevation at different locations within the domain: (**a**) $H_1 = 300$ mm; (**b**) $H_2 = 865$ mm; (**c**) $H_3 = 1114$ mm; (**d**) $H_4 = 1365.5$ mm.

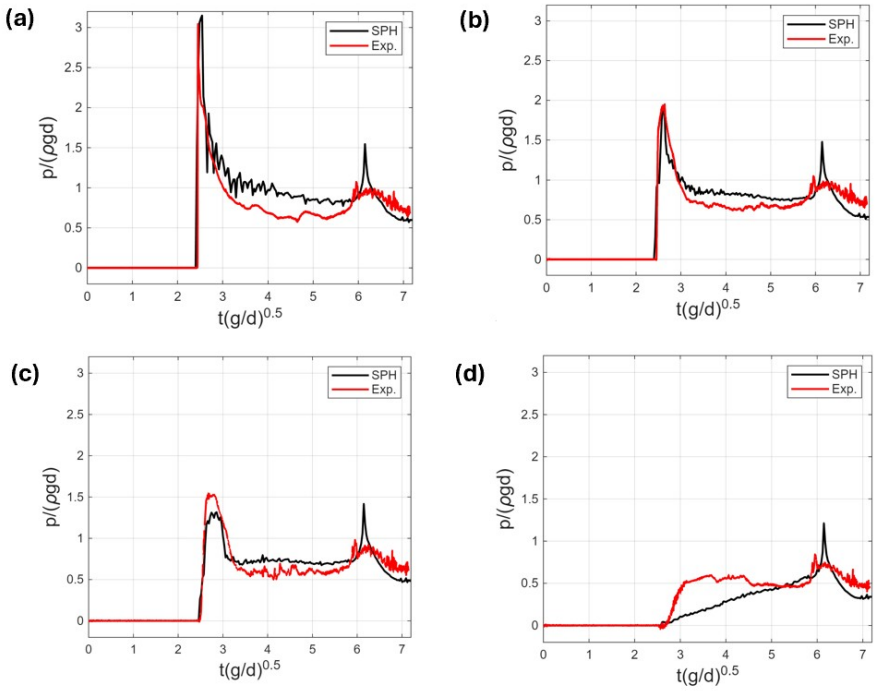

Figure 13. Temporal evolution of pressure for different sensors: (**a**) $P_1 = 3$ mm, (**b**) $P_2 = 15$ mm, (**c**) $P_3 = 30$ mm, (**d**) $P_4 = 80$ mm.

The trends obtained from the numerical simulations show good agreement with the experimental results, with an nRMSE of 0.0732, 0.0765, 0.0865, and 0.14 for P1, P2, P3, and P4, respectively, confirming the code's capability to accurately predict the pressure time-history. The pressure peaks associated with the impact are well-captured by the first three

sensors; however, for the uppermost sensor, the impact effect appears less pronounced, resulting in a quasi-linear pressure increase. It is worth noting that for all four sensors, an instantaneous peak was observed at $t\sqrt{g/d} = 6.14$ (corresponding to approximately 1.07 s). In the numerical simulations, this corresponds to the formation of the splash near the wall (the rightmost one in Figure 9d).

4.2. Wave Train Generation

To verify the computational code's capability to generate regular wave trains, reference was made to the work of Lind et al. [23], in which the results obtained from an ISPH code are compared with those produced by the SAWW (Steady Arbitrary Water Waves) code. A rectangular channel with a length of $L = 18$ m, equipped with a wave maker at the right end, contains a water volume with an undisturbed depth of $d = 0.5$ m. The geometry of the simulated setup is shown in Figure 14.

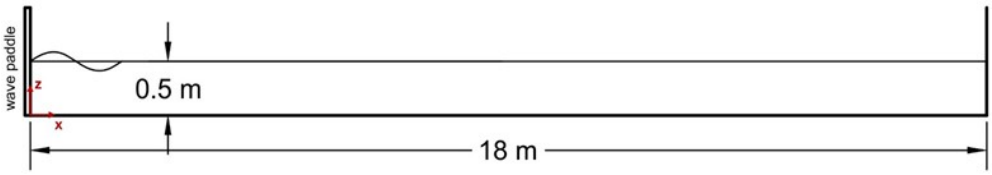

Figure 14. Schematic representation of the test performed by Lind et al. [18].

Two different cases were considered: the first simulates wave trains with a wave height $H_W = 0.05$ m and a wavelength $L_W = 1.5$ m; the second involves a wave height $H_W = 0.1$ m and a wavelength $L_W = 3.0$ m. These parameters result in wave periods T_W of 0.9951 s and 1.5688 s, respectively. Figure 15 shows two snapshots of the examined cases, employing a distorted scale (amplified by a factor of 10) in the vertical direction to emphasize the oscillation induced by the wave propagation. It can be observed that the wave appears slightly irregular near the generation zone; however, progressive regularization is noted as it propagates.

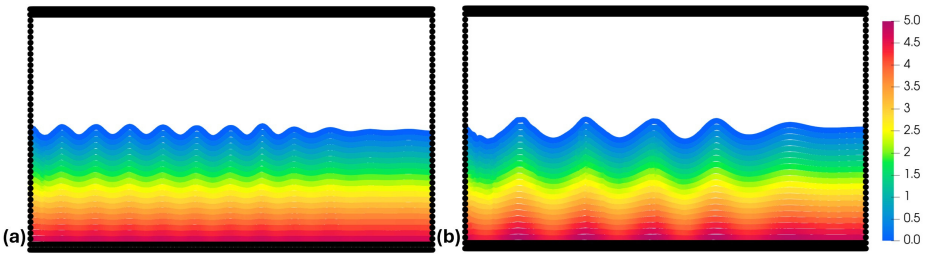

Figure 15. Snapshots of propagating wave trains (vertical direction magnified by a factor of 10): (**a**) case with a wave height H_W0.05 m (a $t = 15$ s); (**b**) case with a wave height $H_W = 0.1$ m (a $t = 9.5$ s).

The motion of the wave maker follows the linearized wave-maker theory [34]. The velocity of the paddle, v_{pad}, which dictates the movement of the wave maker, is given by

$$v_{pad} = 0.5 S_{pad} \omega_W [sin(\omega_W t)] \tag{24}$$

where S_{pad} denotes the paddle stroke, and ω_W is the angular frequency of the wave maker. It should be noted that this velocity must be imposed as a Dirichlet boundary condition in Equation (19), replacing $v_W^{t+\Delta t}$. The paddle stroke S_{pad} is given by

$$S_{pad} = \frac{H_W[sinh(2k_W d) + 2kd]}{2cosh(2k_W d) - 1} \tag{25}$$

where $k_W = \frac{2\pi}{L_W}$ is the wave number. The angular frequency of the wave is determined by the dispersion relation

$$\omega_W^2 = g k_W \cdot tanh(k_W d) \tag{26}$$

The two aforementioned cases were numerically reproduced using the model proposed in this work. The simulations were carried out for a total duration of 20 s, with a time step of $dt = 10^{-4}$ s. The particle diameter dx was set to 0.025 m, resulting in a total of 22,036 particles; this configuration ensures an accurate representation of the wave-induced oscillation without significantly increasing the wall-clock time. The simulations were performed on an AMD Ryzen 9 5950X using 8 cores, with a wall-clock time of approximately 52 h for both cases. To prevent wave reflections, a damping zone (or sponge layer) was introduced at the left boundary of the channel, spanning a distance equal to one wavelength (1.5 m for the first case and 3.0 m for the second case). Within this region, the particle velocities are reduced through a damping function f_{damp}, defined as

$$f_{damp} = 1 - e^{\alpha_{damp}(x - x_{max})} \tag{27}$$

where α_{damp} is a parameter set to 2 in these simulations, x denotes the horizontal coordinates of the particle, and $x_{max} = 18$ m corresponds to the channel end. Figures 16 and 17 present the comparisons between the horizontal velocity v_x, the total pressure p_{tot}, and the dynamic pressure p_{dyn} obtained from the SAWW code and those computed by the numerical model developed in this work, for both the wave crest (in red) and the wave trough (in blue). Specifically, for the SPH results, the third crest and the third trough were considered for both the $H_W = 0.05$ m and $H_W = 0.1$ m cases at simulation times of approximately 15 s and 10 s, respectively. Each point on the plot corresponds to a particle's value along the vertical direction in the SPH code. The horizontal velocity v_x is updated at each iteration, while the total pressure p_{tot} is calculated by solving the system of Poisson equations (see the pressure calculation in Section 3.2). The dynamic pressure p_{dyn} was evaluated exclusively for this comparison using the following expression:

$$p_{dyn} = p_{tot} - (\eta - z)\rho g \tag{28}$$

where η denotes the free-surface elevation and z represents the vertical coordinates of the considered particles.

Figures 16 and 17 demonstrate excellent agreement between the values obtained in all cases for each analyzed variable, indicating that the computational code correctly reproduces the wave trains. Specifically, the Root Mean Square Error (RMSE) values calculated between the SAWW and SPH codes are: 0.036 (16a trough), 0.0655 (16a crest), 0.0111 (16b trough), 0.0114 (16b crest), 0.0159 (16c trough), 0.034 (16c crest), 0.0135 (17a trough), 0.062 (17a crest), 0.0118 (17b trough), 0.0054 (17b crest), 0.026 (17c trough), and 0.0515 (17c crest).

4.3. Evaluation of Pressure on a Vertical Breakwater

In this final test, the numerical code was used to evaluate the pressures acting on a breakwater. The numerical simulation results were validated by comparison with experimental tests conducted by Didier et al. [35], in which the pressures exerted on a breakwater by a regular wave train generated by a wave maker were measured. The tests were carried out in a wave flume with a length of 49.4 m, a width of 1.6 m, and a height of 1.2 m. The structure subjected to the incident waves consists of a breakwater composed of a berm 0.181 m high with a 1:3 slope, and a vertical wall 0.385 m high, positioned 0.2 m from the crest of the slope. The distance between the toe of the berm and the wave maker is 3.62 m. A schematic representation of the experimental setup geometry is shown in Figure 18.

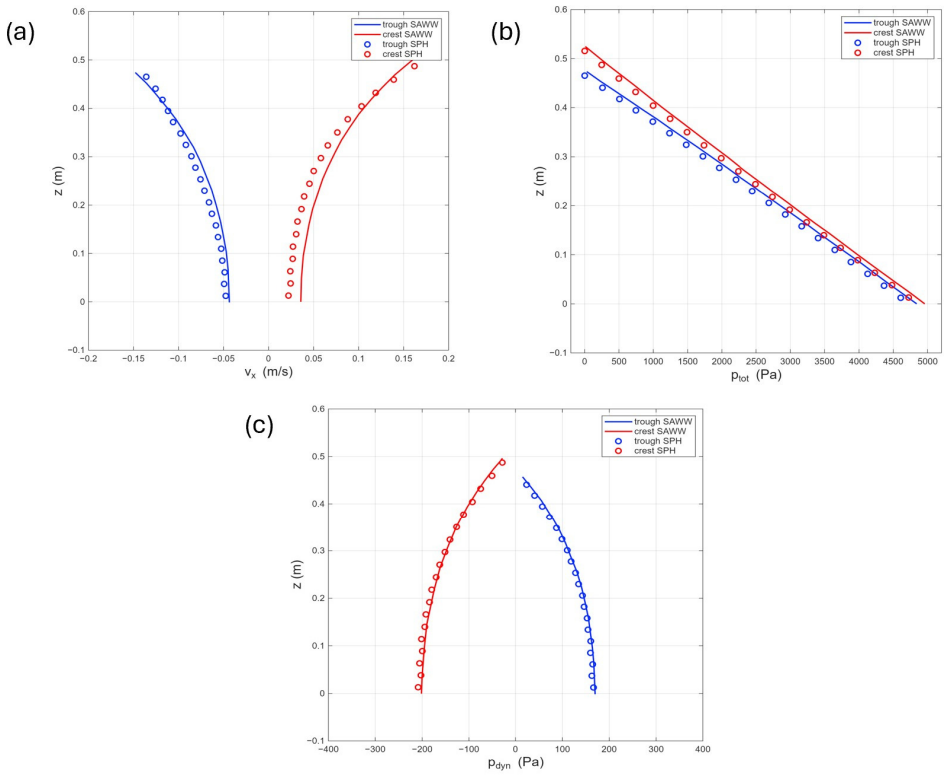

Figure 16. Comparison between the result obtained with SAWW code and the SPH code for the case with $H_W = 0.05$ m, showing different variables as function of depth z: (**a**) horizontal velocity v_x, (**b**) total pressure p_{tot}, (**c**) dynamic pressure p_{dyn}.

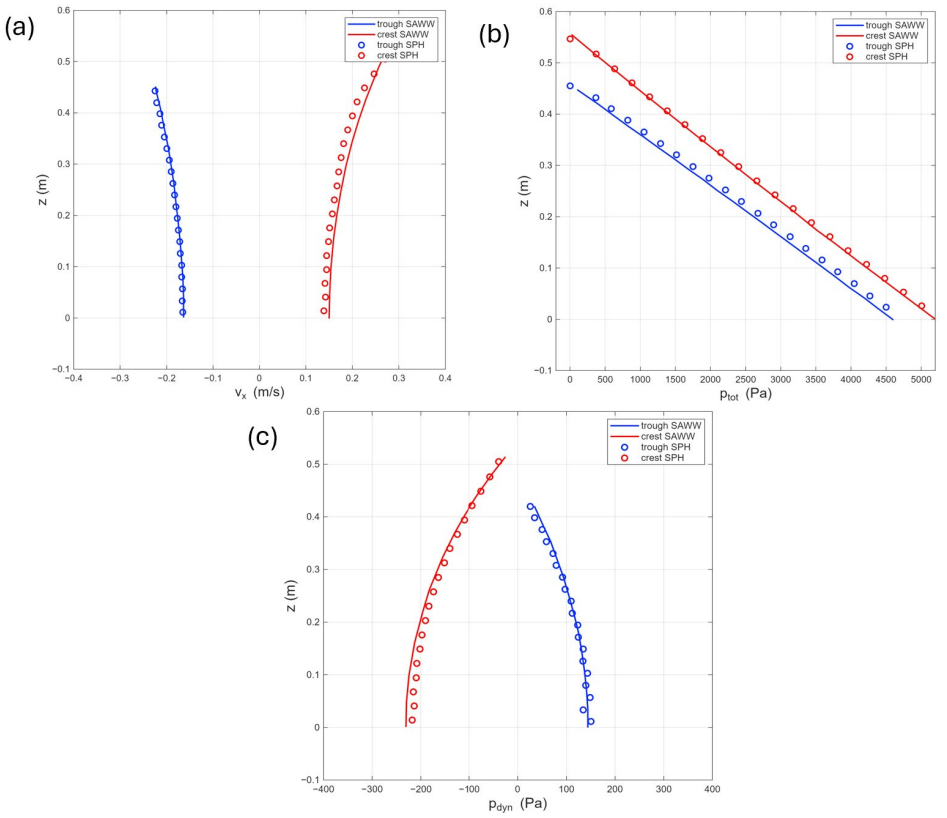

Figure 17. Comparison between the result obtained with SAWW code and the SPH code for the case with $H_W = 0.1$ m, showing different variables as function of depth z: (**a**) horizontal velocity v_x, (**b**) total pressure p_{tot}, (**c**) dynamic pressure p_{dyn}.

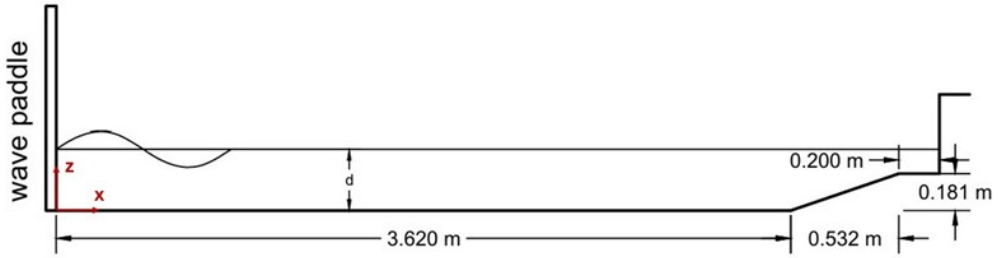

Figure 18. Schematic representation of the experimental tests conducted by Didier et al. [5].

The pressures exerted by the incident waves on the breakwater are measured using six sensors, uniformly spaced by 0.055 m. The tests were conducted with a regular wave train characterized by a wave period $T_W = 1.3$ s and a wave height $H_W = 0.1$ m. The undisturbed water depth was varied to reproduce different conditions. The duration of each test was set to 8.5 s to prevent interference between the waves reflected by the breakwater and the wave generation process by the wave maker.

The experimental tests described above were numerically reproduced using the computational code developed in this work. Figure 19 shows snapshots of a simulation at a specific instant when a wave impacts the breakwater.

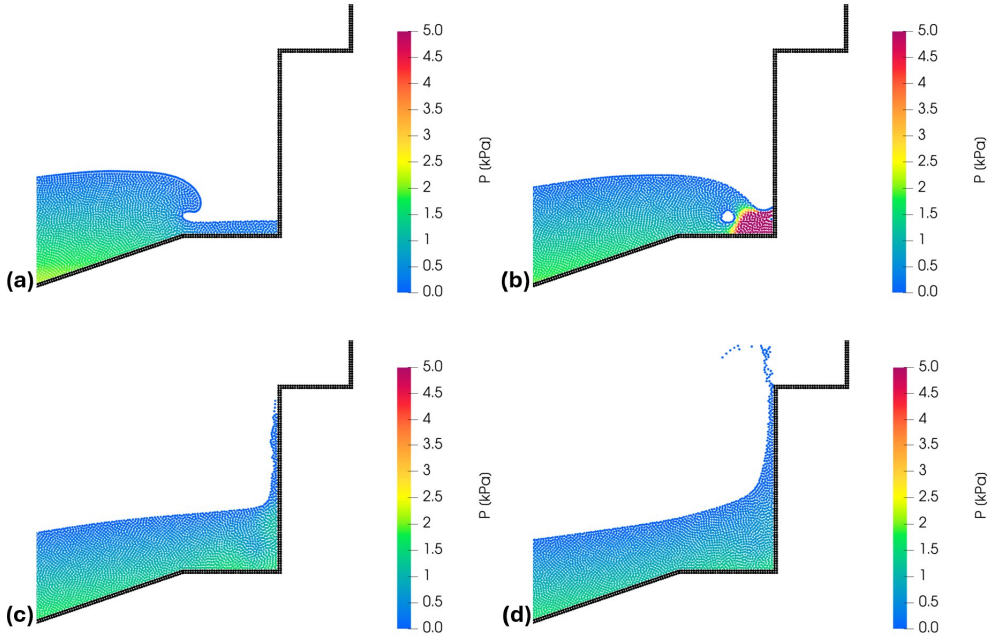

Figure 19. Snapshots of a wave impacting the vertical breakwater: (**a**) t = 5.6 s; (**b**) t = 5.7 s; (**c**) t = 5.8 s; (**d**) t = 5.9 s.

Two different cases were analyzed: the first corresponding to an undisturbed free-surface height $d = 0.266$ m, and the second with an undisturbed height $d = 0.300$ m. Under these conditions, waves with a period $T_W = 1.3$ s and a height $H_W = 0.1$ m yield wavelengths L_W of 1.879 m and 1.965 m, respectively. The wave train was generated by a wave maker whose motion follows the law described in Section 4.2. A particle diameter $dx = 0.005$ m was chosen, resulting in 42,762 particles for the first case and 44,852 for the second. The number of dummy particles was 2548, while the number of mirror and edge extrapolation points was 2548 and 1272, respectively. The time step was set to $\Delta t = 10^{-4}$ s. The simulations were performed on an AMD Ryzen 9 5950X using 8 cores, with a wall-clock time of approximately 149 h for the first case and 207 h for the second.

Figures 20 and 21 show a comparison between the pressures obtained during the experimental tests and those from the numerical simulations. As observed in the plots, four

waves impact the breakwater within the selected simulation window. The pressure trends clearly indicate that the impacts exhibit high-intensity peaks over short durations (approximately $0.01T_W$), followed by more persistent quasi-static loads lasting about $0.2 - 0.5T_W$. Figure 20 illustrates the evolution of the pressures acting on the breakwater for the case with a water depth $d = 0.266$ m. The experimental data (in red) and numerical results (in black) are compared only for the first three pressure sensors (P1, P2, and P3), as the remaining sensors are located at higher elevations and are reached by the water within durations too short to yield significant results. Excellent agreement is observed for the pressures following the initial peak, whereas larger discrepancies occur at the peak itself. The resulting nRMSE values are 0.0607 for P1, 0.1224 for P2, and 0.1248 for P3, showing a decrease in accuracy for the higher sensors. Figure 21 displays the comparison for the case with an undisturbed water depth $d = 0.300$ m. In this instance, a good correlation is also observed, with nRMSE values of 0.0848 for P1, 0.0715 for P2, and 0.0947 for P3. Once again, a slight overestimation of the impact pressure peaks is detected. As stated by Didier et al. [35], these discrepancies may be attributed to an insufficient sampling frequency of the sensors, which might fail to fully capture values occurring over such extremely short intervals (approximately $0.01T_w$). Another possible explanation lies in the fact that the numerical model does not account for air-entrainment effects, which significantly influence such measurements. Regardless, this remains a highly complex phenomenon that exhibited significant variability even between different experimental runs.

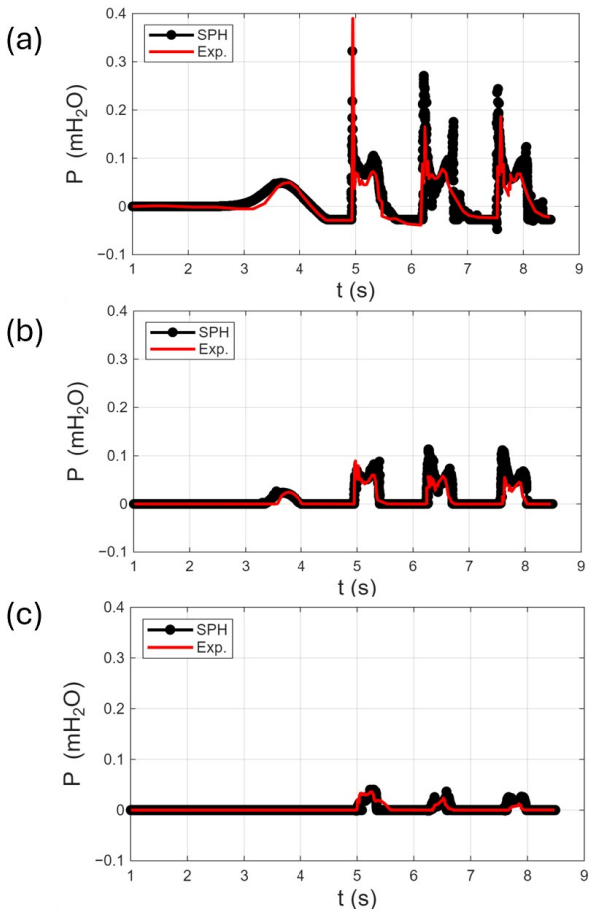

Figure 20. Pressure evolution in time for d = 0.266 m. Comparison between the numerical model (black line) and the experimental data (red line) for sensor P1 (**a**), P2 (**b**) and P3 (**c**).

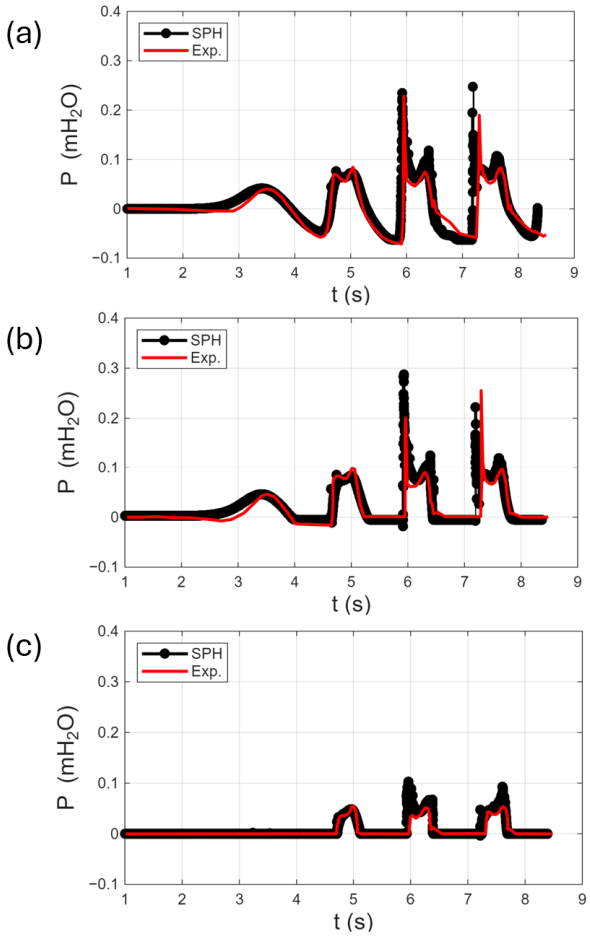

Figure 21. Pressure evolution in time for d = 0.3 m. Comparison between the numerical model (black line) and the experimental data (red line) for sensor P1 (**a**), P2 (**b**) and P3 (**c**).

Figure 22 shows the speedup of the numerical simulation as a function of the number of cores used for each of the presented validation tests.

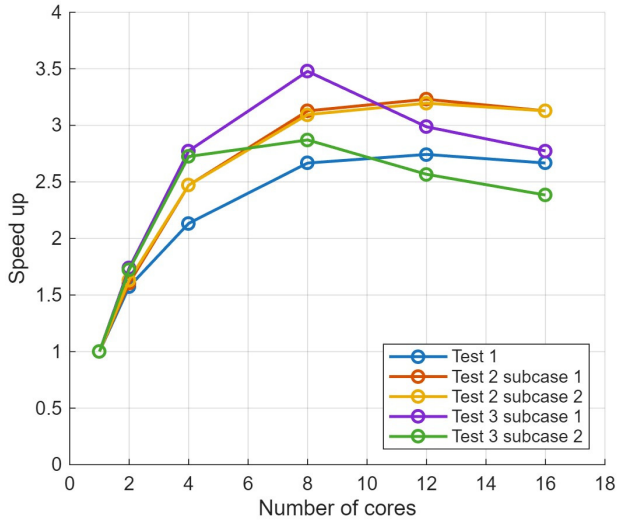

Figure 22. Speedup versus the number of cores employed: blue line (Test 1), dambreak; orange line (Test 2 subcase 1), wave train generation with H_W = 0.05 m; yellow line (Test 2 subcase 2), wave train generation with H_W = 0.1 m; purple line (Test 3 subcase 1), evaluation of pressure on a vertical breakwater with d = 0.266 m; green line (Test 3 subcase 2), evaluation of pressure on a vertical breakwater with d = 0.3 m.

As can be noticed, the maximum speedup is achieved when using 8 cores; beyond this point, no further performance improvements are observed. In the case of Test 3, which is the one involving the largest number of particles, performance degradation is even evident.

5. Conclusions

In this work, a novel Incompressible Smoothed Particle Hydrodynamics (ISPH) numerical model was developed for the simulation of free-surface flows. Within this framework, solid boundaries are modeled using the modified Dynamic Boundary Conditions (mDBC) technique, which is applied here for the first time in this specific context. The implementation of this technique offers several advantages over existing approaches, including the consistent enforcement of the Neumann boundary condition for pressure through fixed particles whose positions can be defined at the initialization stage without requiring subsequent updating. Several comparisons between numerical simulations of the proposed model and benchmark data available in the literature were performed, focusing on free-surface elevation, velocity, and pressure fields. Initially, a dam-break test, a standard benchmark for SPH model validation, was conducted. The results confirmed the efficacy of the computational code in capturing the dynamics of chaotic phenomena. The model's ability to simulate regular wave trains was verified by comparing simulation results with those of Lind et al. [23], validating the pressure and velocity fields across two distinct cases. Finally, wave impact pressures on a vertical breakwater were evaluated and compared with experimental tests conducted by Didier et al. [35] under two different conditions. In all cases, despite the limited number of particles used to discretize the fluid domain, the results showed excellent agreement with literature data, confirming the reliability and accuracy of the developed model in reproducing the investigated phenomena.

Future developments of this work may address several key areas. A significant initial step could involve the parallelization of the code on GPUs to enhance computational efficiency. Particular attention should be directed toward the pressure-solving procedure, which, despite improvements achieved through multi-processor execution, still accounts for over 90% of the computational time per iteration. To enable the simulation of more complex phenomena, a three-dimensional version of the code should be developed. To achieve second-order consistency in the SPH approximation, the formulation proposed by Fatehi and Manzari [36] could be integrated into the numerical scheme; however, it must be noted that this would entail additional computational costs. Finally, further applications and benchmarks involving both experimental data and comparisons with other numerical methods could be conducted to further verify the accuracy and robustness of the proposed model.

Author Contributions: Conceptualization, methodology, validation and investigation M.S.; conceptualization, review and supervision G.C.; review and supervision G.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Koschier, D.; Bender, J.; Solenthaler, B.; Teschner, M. Smoothed particle hydrodynamics techniques for the physics based simulation of fluids and solids. *arXiv* **2019**, arXiv:2009.06944.
2. Le Touzé, D.; Rogers, B.D. Smoothed particle hydrodynamics for free-surface and multiphase flows: A review. *J. Fluids Struct.* **2023**, *118*, 103859. [CrossRef] [PubMed]
3. Monaghan, J.J. Smoothed particle hydrodynamics. *Rep. Prog. Phys.* **2005**, *68*, 1703–1759. [CrossRef]
4. Gotoh, H.; Khayyer, A. On the state-of-the-art of particle methods for coastal and ocean engineering. *Coast. Eng. J.* **2018**, *60*, 79–103.
5. Hughes, J.P.; Graham, D.I. Comparison of incompressible and weakly-compressible SPH models for free-surface water flows. *J. Hydraul. Res.* **2010**, *48*, 105–117. [CrossRef]
6. Lee, E.-S.; Moulinec, C.; Xu, R.; Violeau, D.; Laurence, D.; Stansby, P. Comparisons of weakly compressible and truly incompressible algorithms for the SPH. *J. Comput. Phys.* **2008**, *227*, 8417–8436. [CrossRef]
7. Raizah, Z.; El-Sapa, S.; Aly, A.M. ISPH simulations of thermosolutal convection in an annulus amongst an inner prismatic shape and outer cavity including three circular cylinders. *Case Stud. Therm. Eng.* **2022**, *30*, 101736. [CrossRef]
8. Vacondio, R.; Altomare, C.; De Leffe, M.; Hu, X.; Le Touzé, D.; Lind, S.; Marongiu, J.-C.; Marrone, S.; Rogers, B.D.; Souto-Iglesias, A. Grand challenges for smoothed particle hydrodynamics numerical schemes. *Comput. Part. Mech.* **2021**, *8*, 575–588. [CrossRef]
9. Monaghan, J.J. Simulating free surface flows with SPH. *J. Comput. Phys.* **1994**, *110*, 399–406. [CrossRef]
10. Issa, R.; Lee, E.S.; Violeau, D.; Laurence, D.R. Modelling a plunging breaking solitary wave with eddy-viscosity turbulent SPH models. *J. Hydraul. Res.* **2005**, *43*, 610–624.
11. Randles, P.W.; Libersky, L.D. Smoothed particle hydrodynamics: Some recent improvements and applications. *Comput. Methods Appl. Mech. Eng.* **1996**, *139*, 375–408. [CrossRef]
12. Shao, S.; Lo, E.Y. Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface. *Adv. Water Resour.* **2003**, *26*, 787–800. [CrossRef]
13. Di Monaco, P.; Guelfi, R.; Marrone, S.; Bertazzi, F. SPH modeling of solid boundaries through a semi-analytic approach. *Eng. Appl. Comput. Fluid Mech.* **2011**, *5*, 148–155. [CrossRef]
14. Ferrand, M.; Laurence, D.R.; Rogers, B.D.; Violeau, D.; Kassiotis, C. Unified semi-analytical wall boundary conditions for inviscid, laminar or turbulent flows in the meshless SPH method. *Int. J. Numer. Methods Fluids* **2013**, *71*, 446–472. [CrossRef]
15. Kulasegaram, S.; Bonet, J.; Lewis, R.W.; Profit, M. A variational formulation based contact algorithm for rigid boundaries in 2D SPH applications. *Comput. Mech.* **2004**, *33*, 316–325. [CrossRef]
16. Manenti, S.; Sibilla, S.; Gallati, M.; Agate, G.; Guagnini, R. 3D WCSPH modelling of landslide-water dynamics during 1963 (16)Vajont disaster. *Coast. Eng.* **2016**, *116*, 210–228.
17. Napoli, E.; De Marchis, M.; Gianguzza, G. PANORMUS-SPH: A new smoothed particle hydrodynamics solver for incompressible flows. *Comput. Fluids* **2015**, *109*, 114–126. [CrossRef]
18. English, A.; Domínguez, J.M.; Vacondio, R.; Crespo, A.J.C.; Stansby, P.K.; Gómez-Gesteira, M. Modified dynamic boundary conditions (mDBC) for general-purpose smoothed particle hydrodynamics (SPH): Application to tank sloshing, dam break and fish pass problems. *Comput. Methods Appl. Mech. Eng.* **2019**, *355*, 22–40. [CrossRef]
19. Balay, S.; Abhyankar, S.; Adams, M.F.; Brown, J.; Brune, P.; Buschelman, K.; Dalcin, L.; Eijkhout, V.; Gropp, W.D.; Kaushik, D.; et al. *PETSc Users Manual (No. ANL-21/39-Rev 3.15)*; Argonne National Laboratory: Lemont, IL, USA, 2021.
20. Schwaiger, H.F. An implicit corrected SPH formulation for thermal diffusion with linear free surface boundary conditions. *Int. J. Numer. Methods Eng.* **2008**, *75*, 647–671. [CrossRef]
21. Wendland, H. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Adv. Comput. Math.* **1995**, *4*, 389–396. [CrossRef]
22. Morris, J.P.; Fox, P.J.; Zhu, Y. Modeling low Reynolds number incompressible flows using SPH. *J. Comput. Phys.* **1997**, *136*, 214–226. [CrossRef]
23. Lind, S.J.; Xu, R.; Stansby, P.K.; Rogers, B.D. Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves. *J. Comput. Phys.* **2012**, *231*, 1499–1523. [CrossRef]
24. Launder, B.E.; Spalding, D.B. The numerical computation of turbulent flows. *Comput. Methods Appl. Mech. Eng.* **1974**, *3*, 269–289. [CrossRef]
25. Xu, R.; Stansby, P.; Laurence, D. Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach. *J. Comput. Phys.* **2009**, *228*, 6703–6725. [CrossRef]
26. Liu, M.B.; Liu, G.R. Restoring particle consistency in smoothed particle hydrodynamics. *Appl. Numer. Math.* **2006**, *56*, 19–36. [CrossRef]
27. Marrone, S.; Antuono, M.; Colagrossi, A.; Colicchio, G.; Le Touzé, D.; Graziani, G. δ-SPH model for simulating violent impact flows. *Comput. Methods Appl. Mech. Eng.* **2011**, *200*, 1526–1542. [CrossRef]

28. Jančík, L.; Gombotz, M. Simulation of a 2D dam break problem using SPH method. *Eng. Mech.* **2014**, *21*, 161–171.
29. Salis, M.; Marrone, S.; Antuono, M.; Di Monaco, P.; Fabbri, L. Wave generation and wave–structure impact modelling with WCSPH. *Appl. Ocean. Res.* **2019**, *92*, 101905. [CrossRef]
30. Lobovský, L.; Hartl, E.; Vimmr, J.; Hlaváč, Z. Experimental investigation of dynamic pressure loads during dam break. *J. Fluids Struct.* **2014**, *48*, 407–434. [CrossRef]
31. Cercos-Pita, J.L.; Dalrymple, R.A.; Herault, A. Diffusive terms for the conservation of mass equation in SPH. *Appl. Math. Model.* **2016**, *40*, 8722–8736. [CrossRef]
32. Mirauda, D.; Albano, R.; Sole, A.; Adamowski, J. Smoothed Particle Hydrodynamics Modeling with Advanced Boundary Conditions for Two-Dimensional Dam-Break Floods. *Water* **2020**, *12*, 1142.
33. Zhang, C.; Hu, X.Y.; Adams, N.A. A weakly compressible SPH method based on a low-dissipation Riemann solver. *J. Comput. Phys.* **2017**, *335*, 605–620. [CrossRef]
34. Dean, R.G.; Dalrymple, R.A. *Water Wave Mechanics for Engineers and Scientists*; World Scientific Publishing Company: Singapore, 1991.
35. Didier, E.; Neves, M.G.; Martins, R.; Guanche, R. Wave interaction with a vertical wall: SPH numerical and experimental modeling. *Ocean Eng.* **2014**, *88*, 330–341. [CrossRef]
36. Fatehi, R.; Manzari, M.T. Error estimation in smoothed particle hydrodynamics and a new scheme for second derivatives. *Comput. Math. Appl.* **2011**, *61*, 482–498. [CrossRef]