

Article

Grid-Stamping on a Polygon Model for Implementing Arbitrary-Shaped Boundary Conditions in a Moving Particle Semi-Implicit Method

Hee-Sung Shin and Jong-Chun Park * 

Department of Naval Architecture and Ocean Engineering, Pusan National University,
Busan 46241, Republic of Korea; hs.shin@pusan.ac.kr

* Correspondence: jcpark@pnu.edu

Abstract: This study proposes a new wall boundary condition for the grid-stamping on a polygon (G-StoP) model, which enables a simpler and more efficient handling of boundary surfaces of arbitrarily complex-shaped bodies represented using polygons (or meshes). For example, computer-aided design surface data can be used to analyze flow using a particle-based fluid-solver moving particle semi-implicit method. For coupling simulations of fluid–multibody dynamics, the Pusan-National-University-modified MPS method is improved, and the coupling analysis is performed using RecurDyn, a commercial software package for multibody (or flexible multibody) dynamics. To confirm the applicability of the developed G-StoP model, hydrostatic pressure simulations are conducted in a rectangular tank at various corner angles. Then, the hydrostatic pressure results are compared with previously proposed polygonal wall boundary model results and theoretical solutions. That is, in the case with a corner angle of 30° , it was confirmed that the relative error to the experiment of the polygon model was 11.3%, while that of the G-StoP model was 1.3%. This demonstrates that the proposed G-StoP model is exceptional for numerical stability and robustness even when it is difficult to secure information on neighboring particles as the corner angle of the object becomes small. In addition, the G-StoP model was applied to dam breaking, subaerial landslide tsunami, and wine sloshing problems, and its accuracy and applicability were tested through comparison with experimental and other simulation results. As a result, it was shown that the present simulation results were much closer to the experiments than other simulations.

Keywords: explicitly represented polygon (ERP) wall boundary model; moving particle semi-implicit (MPS) method; multibody dynamics (MBD); RecurDyn; subaerial landslide tsunami generation



Citation: Shin, H.-S.; Park, J.-C. Grid-Stamping on a Polygon Model for Implementing Arbitrary-Shaped Boundary Conditions in a Moving Particle Semi-Implicit Method. *J. Mar. Sci. Eng.* **2023**, *11*, 742. <https://doi.org/10.3390/jmse11040742>

Academic Editors: Zhe Sun and Kamal Djidjeli

Received: 23 February 2023

Revised: 24 March 2023

Accepted: 28 March 2023

Published: 29 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Fluid systems used at industrial sites have extremely complex shapes, wherein single or multiple parts are interconnected (sometimes by joints) to perform various mechanical functions. Therefore, to analyze inner or outer flows that interact with these fluid systems, multibody dynamics (MBD) coupling simulations are required. Relevant interactive simulation technologies have become critical, given the extensive use of computer-aided engineering (CAE)-based designs in recent years.

However, two important concerns must be addressed to develop an interactive simulation technology [1]. The first is related to the description of the motion of fluids and solids. Typically, the motions of solids and fluids are described using the Lagrangian and Eulerian approaches, respectively, and discretized meshes (or control volumes) are used to numerically calculate their motions. However, in an interactive simulation that considers excessive displacement (deformation) or the rotational motion of a rigid (or flexible) body, the harmonization of motion-related descriptions is essential. Therefore, in coupling analyses, particularly for fluids, particle-based methods based on the Lagrangian approach are considered to avoid the problem of regenerating the control volume along

the boundary of a moving object. Representative particle methods include smoothed particle hydrodynamics (SPH) [2] and moving particle semi-implicit (MPS) methods [3]. The second is related to the interface between the fluid and solid domains. The interface is also a part of the fluid domain, and its position is irregular for relatively large motions. For a particle method, information on neighboring particles is required to calculate the partial differential operator of the governing equation expressed as a partial differential equation. Particularly, in the vicinity of the boundary, information regarding neighboring particles may be inadequate. Therefore, mechanical action is also required to satisfy the consistency of boundary conditions at the interface and calculate physical quantities with information on fluid particles and boundary surfaces. However, in a particle method, an extremely cautious approach is required to address solid interfaces with complex geometries because particles may penetrate the wall or the law of conservation of physical quantity may be inapplicable, considerably affecting the calculation results or stability of the entire flow field. Figure 1 illustrates a representative boundary treatment method proposed in some particle method studies. These methods possess the following characteristics:

- **Fixed dummy model:** This model places several dummy particles near the boundary and fixes their velocities at zero. It is relatively simple and has a low computational load. However, assigning accurate boundary conditions to the boundary is challenging. Additionally, the analytical results may have been inaccurate. In some cases, fluid particles penetrate the wall surface. In particular, this model has limitations when used for extremely thin objects. Koshizuka and Oka [3] adopted the fixed dummy model in the original MPS method. Marrone et al. [4] and Adami et al. [5] used a similar model in the SPH method.
- **Mirror model:** This model arranges virtual particles in symmetrical positions with respect to the positions of fluid particles near the boundary. The model satisfies the slip or no-slip condition of an object surface, prevents wall penetration, and enables a relatively accurate boundary surface placement. However, if the boundary curvature is discontinuous or complex, it can be difficult to place the corresponding virtual particles. Moreover, it has limitations in representing objects with extremely thin thicknesses. Akimoto [6] adopted the mirror model to predict the free surface of complex shapes in the MPS method, and Liu et al. [7] discussed it as a method for applying boundary conditions in the ISPH method.
- **Repulsive force model:** This model places wall particles on the boundary surface and imposes an artificial repulsive force on fluid particles approaching the wall particles according to their distance. It is relatively easy to apply and can forcibly prevent fluid particles from penetrating the wall surface. However, given that satisfying the slip or no-slip condition at the interface is challenging and the magnitude of the repulsive force must be determined using empirical constants, the reliability of the results can be rather low. The repulsive force model was proposed by Monaghan [8]. It has since been extended to arbitrary boundary shapes by Monaghan and Kajtar [9].

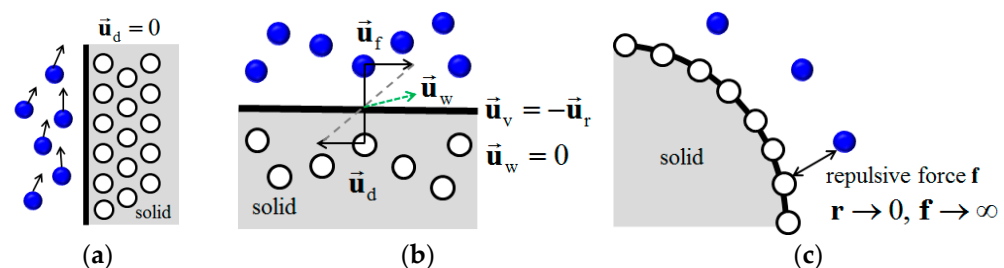


Figure 1. Wall boundary treatment models in particle method. (a) Fixed dummy, (b) Mirror and (c) Repulsive force.

Recently, Zhen et al. [10] carried out the mirror model of boundary conditions based on the approach of Liu et al. [7], and Antuono et al. [11] proposed one based on the

fixed dummy model but using the mirrored physical quantity of fluid particles. However, the application of the aforementioned models to three-dimensional (3D) problems may not be trivial. In other words, computer-aided design (CAD) data for complex shapes include complex and diverse cases related to surfaces with discontinuities or thin edges, as they comprise several polygons segmented from the object surface and their sizes are set appropriately. Therefore, various models have been proposed to address these limitations associated with particle methods. For example, in the MPS method, an explicitly represented polygon (ERP) wall boundary model based on a mirror particle arrangement approach was proposed [12]. This model can reduce the calculation time and memory consumption, as the partial differential operator model is calculated by applying the wall boundary condition only to fluid particles using the normal vector of each polygon. However, in the acute corner angle ($<90^\circ$) of a concave shape, inaccuracies may occur owing to inadequate information on fluid particles. Further, for SPH, a local uniform stencil (LUST) model [13] that extends the concept of an internal fluid stencil to 3Ds from a modified virtual boundary particle (MVBVP) model [14] based on a fixed dummy particle placement approach was proposed. At the beginning of the simulation, a unique stencil for the particle was created and stored in memory, particles inside the object boundary were excluded according to the boundary, and those outside the boundary were used for calculation. This is advantageous because it guarantees consistency even for complex boundary shapes. However, a ray-casting algorithm [15] must be used at every step to classify the boundary in the stencil. Unfortunately, this reduces computational speed.

This study proposes a new wall boundary condition model, the grid-stamping on a polygon (G-StoP) model, which is based on a fixed dummy particle arrangement at the interface where fluid and solid polygons are in contact. The model can compensate for the numerical instability and memory waste of traditional boundary condition models. It also can provide a simpler and more efficient way to handle arbitrarily complex-shaped bodies. The Pusan National University (PNU)-modified MPS (PNU-MPU) method [16] was improved to perform a coupling simulation of fluid–multibody dynamics using RecurDyn [17], commercial software for multibody dynamics (MBD) analyses. The interface for the coupling analysis was constructed using a method similar to that suggested by Yun et al. [18]. To verify the validity of the developed model, numerical simulations were performed for four problems: hydrostatic pressure testing in a tank with various corner edges, dam breaking, subaerial landslide tsunami generation, and wine sloshing. The simulation results were cross-checked with theoretical formulas, experimental values, and numerical analysis results reported in other studies. The accuracy and stability characteristics of this model are reviewed and discussed.

2. Enhanced PNU-MPS Method

Since the introduction of the MPS method by Koshizuka and Oka [3], several studies have proposed various models to stabilize the particle interaction model (described in a subsequent section), which approximates partial differential operators [19–23]. This study introduced enhanced particle interaction models proposed in other studies based on the PNU-MPS method [16], which significantly improved the instability of the pressure field. In addition, an artificial viscosity model [24] and a divergence-free model [25] were combined to improve the pressure oscillation and numerical instability that occur locally in various problems, such as object motions at boundaries and impact loads.

2.1. Governing Equations

The governing equations for incompressible viscous flow are the continuity and Navier–Stokes (N-S) equations.

$$\frac{D\rho}{Dt} = 0 \quad (1)$$

$$\frac{D\vec{u}}{Dt} = -\frac{1}{\rho}\nabla P + \nu\nabla^2\vec{u} + \vec{F} \quad (2)$$

where ρ is the density of the fluid, t is the time, \vec{u} is the velocity vector, P is the pressure, ν is the kinematic viscosity, and \vec{F} is the external force.

Equation (2) denotes the N-S equation based on the Lagrangian approach. The acceleration term on the left-hand side shows a differential form involving the advection of the Lagrangian computed by the direct motion of the fluid particles. The right-hand side consists of terms such as pressure gradient, viscous force, and external force. In the PNU-MPS method, to implement incompressible flows, terms including the partial differential operator of the governing equation were replaced with particle interaction models and then calculated.

2.2. Kernel Function

In the PNU-MPS method, the particle interaction model is constructed based on the kernel function (or weight function) used in Lee et al. [16].

$$w(r) = \begin{cases} \left(1 - \frac{r}{r_e}\right)^3 \left(1 + \frac{r}{r_e}\right)^3 & (0 \leq r < r_e) \\ 0 & (r_e \leq r) \end{cases} \quad (3)$$

where r is the distance between two particles, and r_e is the effective radius, indicating the maximum range of the interaction between the particles.

2.3. Gradient Model

For the gradient of the physical quantity ϕ between particles i and j , an improved gradient model using a correction matrix is used, as shown in Equation (4). Khayyer and Gotoh [20] argued that the gradient model of the original MPS did not exactly satisfy momentum conservation and introduced a corrective matrix to readjust the gradient operator to solve this problem.

$$\langle \nabla \phi \rangle_i = \frac{d}{n^0} \sum_{j \neq i} \frac{(\phi_j - \phi_i)}{|\vec{r}_j - \vec{r}_i|^2} (\vec{r}_j - \vec{r}_i) C_i w(|\vec{r}_j - \vec{r}_i|) \quad (4)$$

Here, d is the dimension of space, and n^0 is the initial particle number density fixed at the same value in the entire computational domain to satisfy the incompressibility condition. The particle number density at particle i is calculated using Equation (5).

$$n_i = \sum_{j \neq i} w(|\vec{r}_j - \vec{r}_i|) = \sum_{j \neq i} w_{ij} \quad (5)$$

C_i shown on the right-hand side of Equation (4) represents the correction matrix. It is obtained using Equation (6).

$$C_i = \begin{pmatrix} \sum_{j \neq i} V_{ij} \frac{w_{ij} x_{ij}^2}{r_{ij}^2} & \sum_{j \neq i} V_{ij} \frac{w_{ij} x_{ij} y_{ij}}{r_{ij}^2} \\ \sum_{j \neq i} V_{ij} \frac{w_{ij} x_{ij} y_{ij}}{r_{ij}^2} & \sum_{j \neq i} V_{ij} \frac{w_{ij} y_{ij}^2}{r_{ij}^2} \end{pmatrix}^{-1} \quad (6)$$

$$V_{ij} = \frac{d}{\sum_{j \neq i} w_{ij}} \quad (7)$$

Here, V_{ij} is the statistical volume of particle i .

2.4. Laplacian Model

The Laplacian model (or diffusion model) distributes some of the physical quantities of particle i in the direction of particle j . The refined Laplacian model [26], as shown in

Equation (8), is introduced to reduce errors that occur when the non-uniform arrangement is, particularly, due to particle motion.

$$\langle \nabla^2 \phi \rangle_i = \frac{2d}{\lambda n^0} \sum_{j \neq i} [(\phi_j - \phi_i) w(|\vec{r}_j - \vec{r}_i|)] - L_i \cdot \langle \nabla \phi \rangle_i \quad (8)$$

where λ is a coefficient for matching the degree of statistical diffusion with the analytic solution, as shown in Equation (9).

$$\lambda = \frac{\sum_{j \neq i} |\vec{r}_j - \vec{r}_i|^2 w(|\vec{r}_j - \vec{r}_i|)}{\sum_{j \neq i} w(|\vec{r}_j - \vec{r}_i|)} \quad (9)$$

The second term on the right-hand side of Equation (8) indicates the correction term introduced to eliminate numerical errors that can be generated from the first-order term in the diffusion model of the original MPS method. Moreover, it is used to obtain the dot product between the correction vector L_i and the gradient term. The correction vector L_i can be expressed by Equation (10).

$$L_i = \frac{2d}{\lambda n^0} \left(\sum_{j \neq i} x_{ij} w_{ij}, \sum_{j \neq i} y_{ij} w_{ij} \right) \quad (10)$$

2.5. Incompressible Model

The original MPS employs a semi-implicit scheme to solve incompressible flows. First, in the explicit stage, the temporal velocity \vec{u}_i^* is obtained via the viscous and external force terms of the N-S equation, assuming that the coordinates and velocity are given as \vec{r}_i^n and \vec{u}_i^n , respectively, at the n time step. Then, the temporal coordinate \vec{r}_i^* is calculated using Euler's explicit scheme in time, as shown in Equation (11).

$$\vec{r}_i^* = \vec{r}_i^n + \Delta t \vec{u}_i^* \quad (11)$$

In the intermediate stage, to solve the pressure Poisson equation (PPE), the source term on the right-hand side is used by blending the velocity divergence condition and the change in particle number density, as shown in Equation (12).

$$\langle \nabla^2 P \rangle_i^{n+1} = - \left[(1 - \gamma) \frac{\rho}{\Delta t} \nabla \cdot \vec{u}_i^* + \gamma \frac{\rho}{\Delta t^2} \frac{\vec{n}_i^* - n^0}{n^0} \right] \quad (12)$$

$$\langle \nabla \vec{u}_i \rangle = \frac{d}{n^0} \sum_{j \neq i} \frac{(\vec{u}_j - \vec{u}_i) \cdot (\vec{r}_j - \vec{r}_i)}{|\vec{r}_j - \vec{r}_i|^2} C_i w(|\vec{r}_j - \vec{r}_i|) \quad (13)$$

where the blending factor γ is determined by the influence of velocity divergence and change in particle number density. It was adopted from the recommended range of Lee et al. [16]: $0.01 < \gamma < 0.05$.

The PPE (12) can be calculated iteratively, and a conjugate gradient method [27] obtains an iterative solution.

After obtaining the pressure P_i^{n+1} at a time step $(n + 1)$, the pressure gradient is calculated using Equation (14).

$$\langle \nabla P \rangle_i^{n+1} = \frac{d}{n^0} \sum_{j \neq i} \frac{(P_j^{n+1} - P_i^{n+1})}{|\vec{r}_j - \vec{r}_i|^2} (\vec{r}_j - \vec{r}_i) C_i w(|\vec{r}_j - \vec{r}_i|) \quad (14)$$

Finally, the correction velocity \vec{u}_i^c is calculated in terms of P^{n+1} using Equation (15), and the velocity and coordinates at the $(n + 1)$ time step are calculated using Equations (16) and (17), respectively.

$$\vec{u}_i^c = -\frac{\Delta t}{\rho} \langle \nabla P \rangle_i^{n+1} \quad (15)$$

$$\vec{u}_i^{n+1} = \vec{u}_i^* + \vec{u}_i^c \quad (16)$$

$$\vec{r}_i^{n+1} = \vec{r}_i^* + \vec{r}_i^c \quad (17)$$

In determining the time interval Δt in the PNU-MPS method, a variable time step is used according to the maximum velocity at each time step of the particles in the flow field, as shown in Equation (18).

$$\Delta t = \frac{Cl_0}{\|v_{max}^n\|} \quad (18)$$

where C is the Courant number (herein, it is equal to 0.2), l_0 is the initial particle distance, and v_{max}^n is the maximum velocity of the fluid particles in the n -th step.

2.6. Free-Surface Boundary Condition

Particles located on the free surface should be searched accurately to impose the dynamic boundary condition that “the pressure on the free surface is approximated to atmospheric pressure” on particles located on the free surface. Koshizuka and Oka [3] determined free-surface particles when the particle number density was lower than the critical value. However, because this condition often fails to accurately search for free-surface particles, the PNU-MPS method is used in the following search condition.

$$n_i < \beta_1 n^0 \quad (19)$$

$$N_i < \beta_2 N^0 \quad (20)$$

where N_i represents the number of neighboring particles, and the coefficients, $\beta_1, \beta_2 \leq 1.0$, are used to set the critical value for the free-surface searching algorithm.

2.7. Collision Model

In the PNU-MPS method, particle redistribution is performed by applying a collision model to adjust particle position. The collision model comprises a repulsive force that moves the particles when they are close to each other. At this time, the particle is corrected using the rate of change in velocity derived from the relative velocity and the conservation of momentum.

$$\begin{aligned} \Delta \vec{u}_i &= -(1+b) \{ (\vec{u}_i - \vec{u}_{relative}) \cdot \vec{n}_r \} \vec{n}_r \\ \vec{u}_{relative} &= \frac{\rho_i \vec{u}_i + \rho_j \vec{u}_j}{\rho_i + \rho_j} \\ \vec{n}_r &= \frac{(\vec{r}_j - \vec{r}_i)}{|\vec{r}_j - \vec{r}_i|} \end{aligned} \quad (21)$$

Here, b is the reconstruction parameter; Lee et al. [16] confirmed that the error is minimized at $b \leq 0.2$. In this study, $b = 0.1$.

2.8. Artificial Viscosity Model

Neumann and Richtmyer [24] introduced the concept of artificial viscosity with improved pressure oscillation. This concept can be expressed as an additional term for pressure, assuming that the dissipation mechanism is negligible except in the vicinity of the impact. It can also be introduced and used in SPH [28,29]. To address the artificial viscosity in this study, a Q term was added to the N-S equation, as shown in Equation (22). The term

can be eliminated from the pressure term, and the gradient term of Q can be revised as in Equation (4) and expressed using Equation (23):

$$\frac{D\vec{u}}{Dt} = -\frac{1}{\rho}\nabla[P + Q] + \nu\nabla^2\vec{u} + \vec{F} \quad (22)$$

$$\begin{aligned} \langle Q \rangle_i &= \frac{d}{n^0} \sum_{j \neq i} \alpha \rho l_0 c \frac{(\vec{u}_j - \vec{u}_i) \cdot (\vec{r}_j - \vec{r}_i)}{|\vec{r}_j - \vec{r}_i|^2} (\vec{r}_j - \vec{r}_i) C_i w(|\vec{r}_j - \vec{r}_i|), \text{ if } \vec{u}_{ij} \cdot \vec{r}_{ij} < 0, \\ &= 0, \text{ if } \vec{u}_{ij} \cdot \vec{r}_{ij} > 0 \end{aligned} \quad (23)$$

$$c = 10\|v_{max}\| = 10\sqrt{gH}$$

Here, α is a coefficient that controls the intensity of the artificial viscosity, and was set to 0.01 in this study.

2.9. Divergence-Free Model

Jeong et al. [25] proposed a velocity divergence equation (24) based on the lattice approach to obtain a more stable pressure field than that derived in Equation (13) used in the existing PNU-MPS method.

This method, as shown in Equation (24), constructs the control volume based on the target particle i . Then, the representative velocities are obtained by interpolation from the velocities of neighboring particles at each plane. At this point, a certain smoothing (or filtering) effect is expected through the interpolation process.

$$\langle \nabla \cdot \vec{u} \rangle_i = \frac{u_{right} - u_{left}}{2l_0} + \frac{v_{top} - v_{bottom}}{2l_0} + \frac{w_{front} - w_{back}}{2l_0} \quad (24)$$

Here, l_0 is the particle size, and the representative velocities in each plane are as follows:

$$\begin{aligned} u_{right} &= \sum_{j \neq i} \frac{u_j w(|\vec{r}_j - \vec{r}_{right}|)}{w(|\vec{r}_j - \vec{r}_{right}|)}, & u_{left} &= \sum_{j \neq i} \frac{u_j w(|\vec{r}_j - \vec{r}_{left}|)}{w(|\vec{r}_j - \vec{r}_{left}|)} \\ v_{top} &= \sum_{j \neq i} \frac{v_j w(|\vec{r}_j - \vec{r}_{top}|)}{w(|\vec{r}_j - \vec{r}_{top}|)}, & v_{bottom} &= \sum_{j \neq i} \frac{v_j w(|\vec{r}_j - \vec{r}_{bottom}|)}{w(|\vec{r}_j - \vec{r}_{bottom}|)} \\ w_{front} &= \sum_{j \neq i} \frac{w_j w(|\vec{r}_j - \vec{r}_{front}|)}{w(|\vec{r}_j - \vec{r}_{front}|)}, & w_{back} &= \sum_{j \neq i} \frac{w_j w(|\vec{r}_j - \vec{r}_{back}|)}{w(|\vec{r}_j - \vec{r}_{back}|)} \end{aligned} \quad (25)$$

3. Wall Boundary Condition

This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation, as well as the experimental conclusions that can be drawn.

3.1. Establishment of Interface for Coupled Simulation with MBD Solver

When CAD data for the shape of a multibody are composed of polygons, an interface to perform coupling simulations with the flow field should be built. This study attempted to interoperate with the enhanced PNU-MPS method through the standard particle interface for co-simulation provided by RecurDyn [17], a commercial solver for MBD analysis. Because RecurDyn supports flexible multibody dynamic analyses, it can also analyze the deformations of objects owing to flow in the future [18].

Figure 2 shows the interface of the fluid–multibody dynamics. First, to analyze the fluid, the RecurDyn solver transfers the velocity, position, angular velocity, and quaternion information of the multibody to an enhanced PNU-MPS solver. The enhanced PNU-MPS solver integrates the pressure acting on the wall boundary particles and returns the force and torque acting on each element of the multibody to the RecurDyn solver. This process is

repeated for all time steps, and the numerical calculations for the fluid and the multibody are performed independently. The synchronization process in the time step between the two solvers is applied as reported by Yun et al. [18].

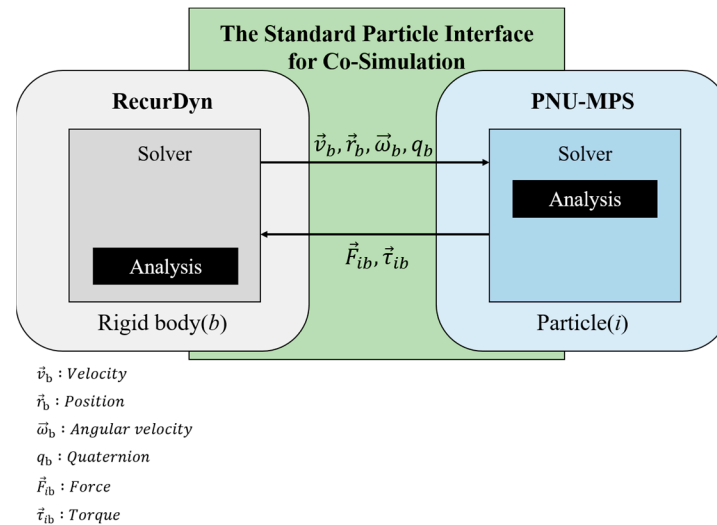


Figure 2. Diagram of handling interface for coupling simulation of fluid–multibody dynamics.

The wall distance is calculated using the distance vector between the polygon vertex and the wall particle [30]. As shown in Figure 3, \vec{v}_p represents a vector from a vertex of the polygon to a particle. Additionally, \vec{v}_1 and \vec{v}_2 are the vectors from the first vertex to other vertices of the polygon. The value of \vec{v}_p can be calculated by the linear Equation (26). In case of the coefficient $\alpha_p + \beta_p \leq 1$, the particle can be considered to belong to the corresponding polygon. Additionally, in case of the coefficient $\gamma_p \leq 1.5l_0$, the particle is considered to be near the wall. The perpendicular vector between the polygon and particle can be expressed as $-\gamma_p \vec{v}_n$.

$$\begin{aligned}
 \{\alpha_p, \beta_p, \gamma_p\}^T [\{\vec{v}_1\}, \{\vec{v}_2\}, \{\vec{v}_n\}] &= \vec{v}_p \\
 \{\alpha_p, \beta_p, \gamma_p\}^T &= [\{\vec{v}_1\}, \{\vec{v}_2\}, \{\vec{v}_n\}]^{-1} \vec{v}_p \\
 \vec{n}_i &= -\gamma_p \vec{v}_n, \quad (\alpha_p + \beta_p \leq 1, \gamma_p < 1.5l_0)
 \end{aligned}
 \tag{26}$$

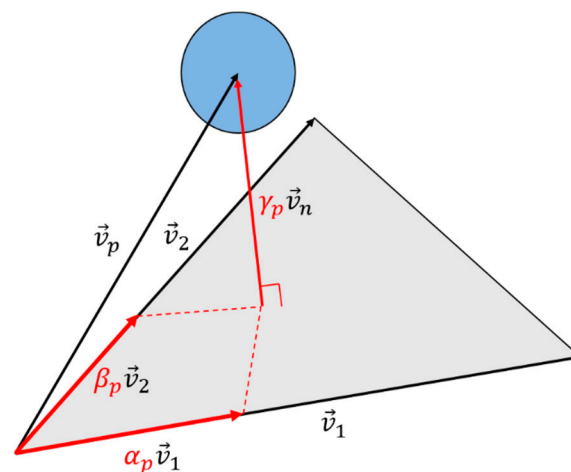


Figure 3. Definition of wall distance vector on polygon.

3.2. Explicitly Represented Polygon Wall Boundary Model

Unlike existing methods that use particles placed on the wall or inside the object, the ERP wall boundary model proposed by Mitsume et al. [12] is processed using multiple

polygon data representing the wall of the object. In other words, as shown in Figure 4, the wall boundary condition is applied by mirror particles temporarily generated inside the wall according to Equation (27) using the normal vector of the CAD data, which expresses the shape of an object as a polygon.

$$R^{ref}(\vec{n}_i) \equiv I - 2\vec{n}_i \otimes \vec{n}_i \quad (27)$$

where R^{ref} is the reflection transformation matrix, I is the identity matrix, and \vec{n}_i is the unit normal vector from particle i to the wall.

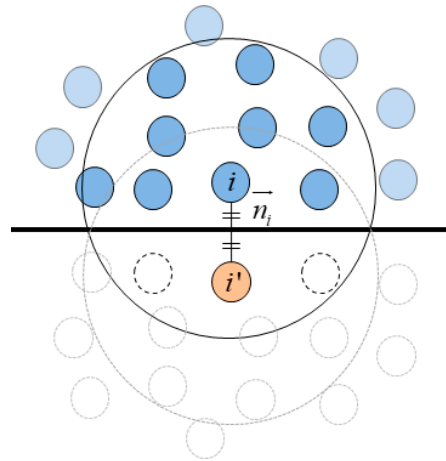


Figure 4. Concept of ERP wall boundary model.

In Equation (27), the vector \vec{n}_i can be obtained from polygon data, and the coordinate vector of i' , which is a virtual particle symmetrical to the wall, can be easily generated, as shown in Figure 4. In other words, the particles near i' can include i and its neighboring particles (\mathbb{P}), as shown in Equation (28).

$$\mathbb{P}_{i'} \subset \mathbb{P}_i + \{i\} \quad (28)$$

3.2.1. Pressure Gradient Term

In the ERP, the wall pressure boundary condition is applied along with the Neumann boundary condition. The gradient term for the wall particles is supplemented with an additional term, as shown in Equation (29).

$$\begin{aligned} \langle \nabla P \rangle_i &= \langle \nabla P \rangle_i^{\text{fluid}} + \langle \nabla P \rangle_{i'}^{\text{wall}} \\ &= \frac{d}{n^0} \left[\sum_{j \neq i} \frac{(P_j - P_i)}{|\vec{r}_j - \vec{r}_i|^2} (\vec{r}_j - \vec{r}_i) C_i w(|\vec{r}_j - \vec{r}_i|) \right. \\ &\quad \left. + R_i^{\text{ref}} \sum_{j \ni i} \frac{(P_j - P_i - \rho g(\vec{r}_j^{\text{wall}} \cdot \hat{k}))}{|\vec{r}_j - \vec{r}_{i'}|^2} (\vec{r}_j - \vec{r}_{i'}) C_i w(|\vec{r}_j - \vec{r}_{i'}|) \right] \end{aligned} \quad (29)$$

where \vec{r}_j^{wall} represents the distance vector between i and j , which is the mirror particle on the wall, and \hat{k} is the unit vector in the vertical direction.

The term $\rho g(\vec{r}_j^{\text{wall}} \cdot \hat{k})$ in Equation (29) was newly improved in this study to consider the difference in hydrostatic pressure of particles near the wall. Consequently, an accurate pressure can be obtained when calculating the gradient term near the boundary of the floor or ceiling wall.

3.2.2. Viscosity Term

The boundary condition for velocity is applied to the no-slip condition. This condition induces a viscous effect on the wall to prevent the transfer of physical quantities through the fluid wall. Therefore, when the velocity of the wall is \vec{u}_i^{wall} , the viscous term can be expressed using Equation (30).

$$\begin{aligned} \langle \nabla^2 u \rangle_i &= \langle \nabla^2 u \rangle_i^{fluid} + \langle \nabla^2 u \rangle_i^{wall} \\ &= \frac{2d}{\lambda n_0} \left[\sum_{j \neq i} (\vec{u}_j - \vec{u}_i) w(|\vec{r}_j - \vec{r}_i|) + \sum_{j \ni i} 2\vec{u}_i^{wall} - (\vec{u}_j - \vec{u}_{j'}) w(|\vec{r}_j - \vec{r}_{j'}|) \right] \end{aligned} \quad (30)$$

3.3. Grid-Stamping on Polygon (G-StoP) Model

3.3.1. Concept of G-StoP Model

The arrangement of several dummy particles on or near a boundary is well known. However, several challenges may arise. For example, dummy particles require considerable memory when applied to 3D complex shapes, must be rearranged whenever the boundary moves, or cannot easily address thin films (pellicles or thin walls). To overcome these disadvantages and secure the insufficient particle number density of the particle i near the wall, the newly proposed model in this study, termed the “G-StoP model” creates virtual wall particles temporally and automatically (without the need for memory storage) by locally interacting with polygonal walls.

The following steps are applied to create virtual wall particles in the G-StoP model.

1. In Figure 5a, the vectors \vec{n}_x , \vec{n}_y , and \vec{n}_z are the basis vectors in the Cartesian coordinate system. \vec{n}_i is the distance vector to the wall of particle i and is obtained from the polygonal CAD information corresponding to particle i .
2. To establish parallel vectors to generate temporarily local grids on polygons, \vec{n}_x is rotated along the axis of \vec{n}_\perp through an angle θ in the direction of \vec{n}_i . At this time, Rodrigues' rotation formula [31], which is expressed in the form of a rotational matrix, is used to rotate a vector by providing rotational angle and axis information. In other words, $\vec{n}_{x'}$ projected in the direction of \vec{n}_i as well as $\vec{n}_{y'}$ and $\vec{n}_{z'}$ in directions parallel to the polygon can be obtained using Equation (31), as shown in Figure 5b.
3. To generate grid vectors parallel to the wall with the generated polygon vectors $\vec{n}_{y'}$ and $\vec{n}_{z'}$, the coordinate information $P_k(x)$ and $P_k(y)$ of the 2D local grid system in Figure 5c is used. Therefore, the lattice vector can be expressed as $P_k(x)\vec{n}_{z'} + P_k(y)\vec{n}_{y'}$, and vectors parallel to the wall and directed to each lattice point can be obtained, as shown in Figure 5d. In this study, nine lattice points were used because the effective radius of the surrounding particles was set to 2.1 times the particle size. Depending on the nature of the problem, the number of grid points could be increased or decreased.
4. The final expression is simply derived in Equation (32). The distance information between the virtual and fluid particles can be obtained using the grid vectors $P_k(x)\vec{n}_{z'} + P_k(y)\vec{n}_{y'}$ and \vec{n}_i parallel to the wall, and \vec{r}_k^{wall} enables the assignment of the boundary conditions at the wall.

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} n_{\perp x}^2(1 - \cos \theta) + \cos \theta & n_{\perp x}n_{\perp y}(1 - \cos \theta) - n_{\perp z} \sin \theta & n_{\perp x}n_{\perp z}(1 - \cos \theta) - n_{\perp y} \sin \theta \\ n_{\perp y}n_{\perp x}(1 - \cos \theta) - n_{\perp z} \sin \theta & n_{\perp y}^2(1 - \cos \theta) + \cos \theta & n_{\perp y}n_{\perp z}(1 - \cos \theta) - n_{\perp x} \sin \theta \\ n_{\perp z}n_{\perp x}(1 - \cos \theta) - n_{\perp y} \sin \theta & n_{\perp z}n_{\perp y}(1 - \cos \theta) - n_{\perp x} \sin \theta & n_{\perp z}^2(1 - \cos \theta) + \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (31)$$

$$\vec{r}_k^{wall} = \vec{r}_k - \vec{r}_i = 2\vec{n}_i + l_0 \{ P_k(x)\vec{n}_{z'} + P_k(y)\vec{n}_{y'} \}; \text{ in this study } (1 \leq k \leq 9) \quad (32)$$

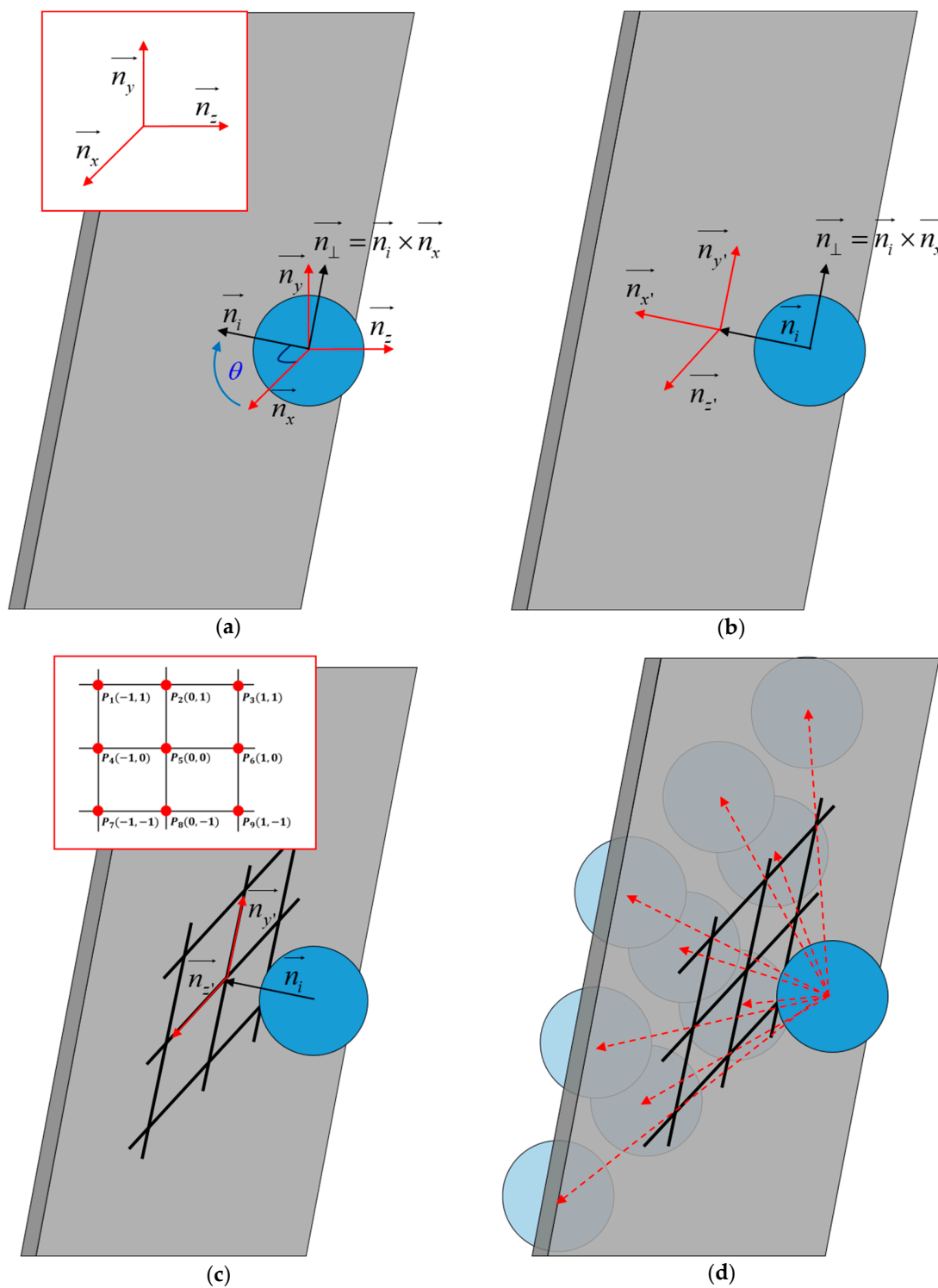


Figure 5. Concept of G-StoP model. (a) Axial vector rotation, (b) Polygon parallel vector generation, (c) Grid generation, (d) Particle arrangement.

3.3.2. Pressure Gradient Term

The pressure gradient term between the virtual wall particle k and the fluid particle i is imposed with the Neumann boundary condition and calculated separately for the fluid and wall particles, as shown in Equation (33). The difference in hydrostatic pressure between k and i is considered.

$$\begin{aligned}
\langle \nabla P \rangle_i &= \langle \nabla P \rangle_i^{\text{fluid}} + \langle \nabla P \rangle_k^{\text{wall}} \\
&= \frac{d}{n^0} \left[\sum_{j \neq i} \frac{P_j - P_i}{|\vec{r}_j - \vec{r}_i|^2} (\vec{r}_j - \vec{r}_i) C_i w(|\vec{r}_j - \vec{r}_i|) \right. \\
&\quad \left. + \sum_{k=1}^n \frac{-\rho g(\vec{r}_k^{\text{wall}} \cdot \hat{k})}{|\vec{r}_k - \vec{r}_i|^2} (\vec{r}_k - \vec{r}_i) C_i w(|\vec{r}_k - \vec{r}_i|) \right]
\end{aligned} \quad (33)$$

In Equation (33), the second term on the right-hand side compensates for the insufficient physical quantity of the fluid particle i near the wall.

3.3.3. Viscosity Term

The Laplacian model of the viscous term can be calculated using Equation (34) considering the no-slip condition of the wall.

$$\begin{aligned}
\langle \nabla^2 u \rangle_i &= \langle \nabla^2 u \rangle_i^{\text{fluid}} + \langle \nabla^2 u \rangle_k^{\text{wall}} \\
&= \frac{2d}{\lambda n^0} \sum_{j \neq i} \left[(\vec{u}_j - \vec{u}_i) w(|\vec{r}_j - \vec{r}_i|) + \sum_{k=1}^n (\vec{u}_k - \vec{u}_i) w(|\vec{r}_k - \vec{r}_i|) \right]
\end{aligned} \quad (34)$$

Here, \vec{u}_k is the velocity of the virtual particle k and should be extrapolated from the neighboring fluid particles to satisfy the no-slip boundary condition on the wall.

For the ERP, the velocity of the particles inside the boundary can be easily adapted to fulfill the no-slip condition on the wall from the corresponding fluid particles. However, for the G-StoP model, because fluid particles do not exist exactly in the symmetrical positions of the wall particles, the following steps are performed to extrapolate the velocity of the virtual particles: First, the symmetric position of k' in the fluid region can be found at the wall particle k using Equation (35). Then, as given in Equation (36), the weighted average velocity at the position k' is determined based on the velocity of the neighboring particles of i . At this point, insufficient information can be supplemented by applying it to the cloud model proposed by Akimoto [6], as shown in Figure 6. The obtained velocity of k' is delivered to the wall particle k considering the no-slip condition.

$$\vec{r}_{k'} = \vec{r}_k - 2\vec{n}_i \quad (35)$$

$$\vec{u}_k = - \sum_{j \ni i} \frac{\vec{u}_j w(|\vec{r}_j - \vec{r}_{k'}|)}{w(|\vec{r}_j - \vec{r}_{k'}|)} \quad (36)$$

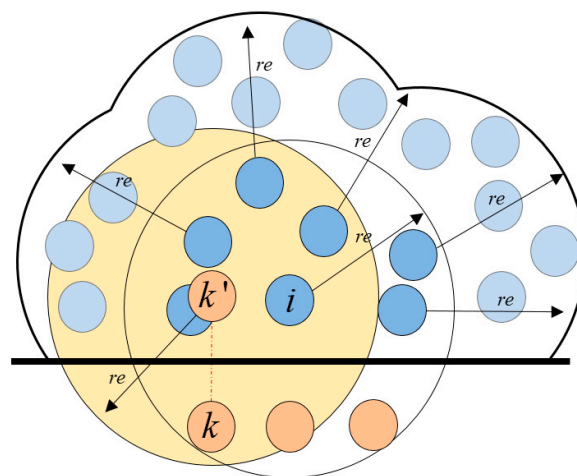


Figure 6. Selection of effective range for acquiring information of neighbor particles.

4. Verification and Validation (V&V)

4.1. Hydrostatic Pressure Problem with Various Corner Angles

In the first simulation, to crosscheck and compare the numerical stability depending on the boundary surface treatment of objects in the enhanced PNU-MPS method, to which the ERP and G-StoP models are applied, the hydrostatic pressure problem in a water tank with boundary shapes of various corner angles is simulated.

As illustrated in Figure 7, the water tank has a 3D shape and three corner angles (90° , 45° , and 30°). In all the scenarios, the water depth was set to a fixed value of 0.4 m. The 90° water tank is typically rectangular, with a length of 0.6 m at the base and a height of 0.8 m at the top. The 45° tank has a right-triangular shape with a floor length of 1.0 m and a maximum ceiling height of 0.5 m. The 30° tank has an obtuse angle at the top. The floor length is 1.6 m, and the ceiling height is 0.46 m. As the corner angle of the tank decreases, the number of particles used increases owing to the volume change inside the tank. Table 1 presents the details of the simulation conditions. The total calculation time is 5 s, and the pressure is measured at the midpoint of the bottom of each tank for comparison between the two boundary condition models.

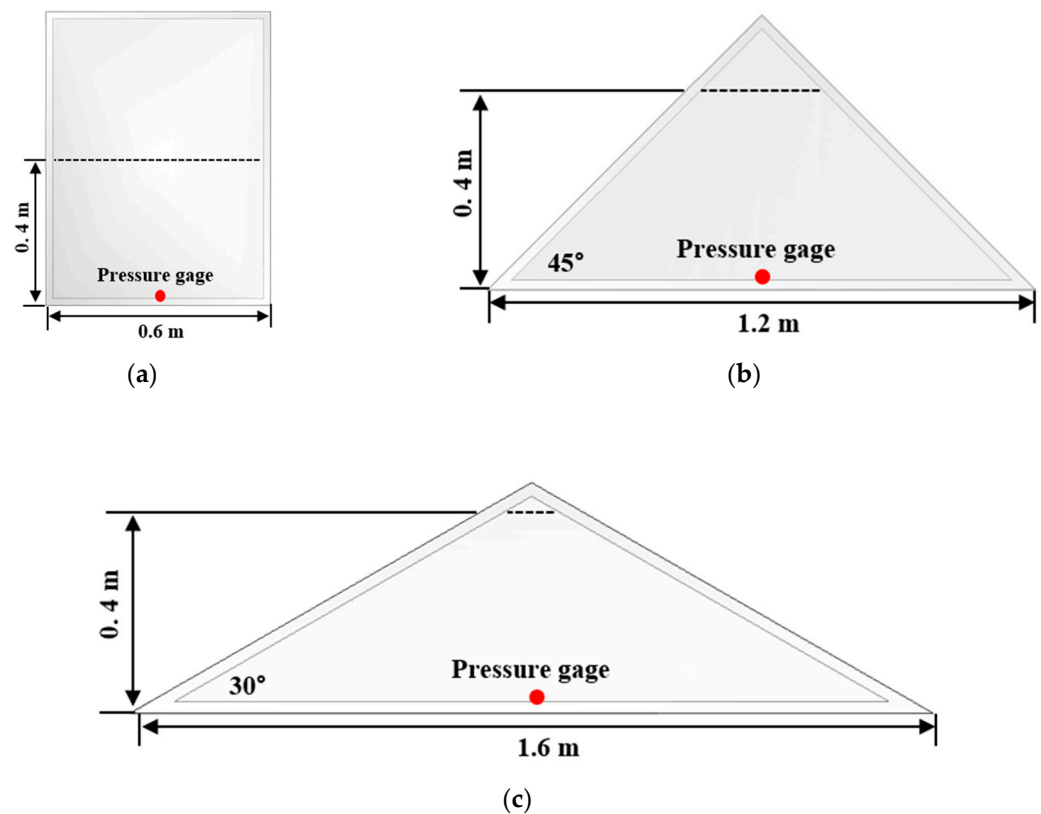
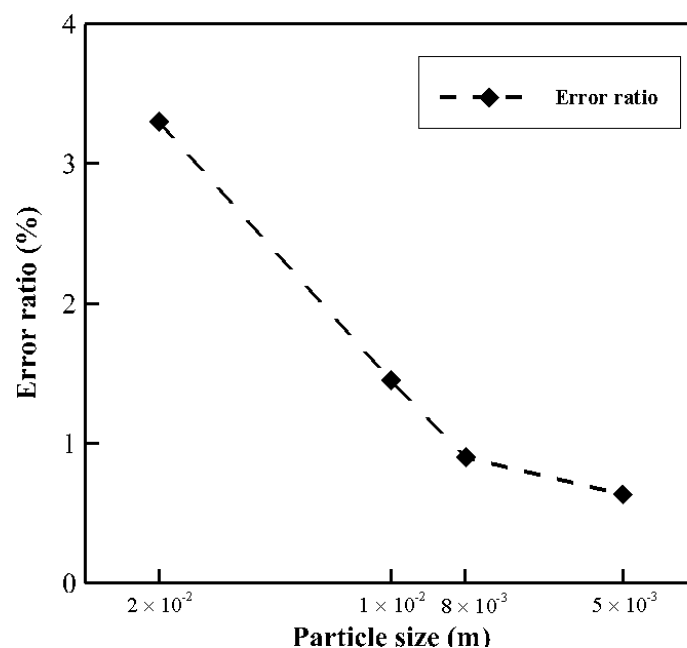


Figure 7. Side view of water tanks with various corner angles. (a) Corner angle = 90° , (b) Corner angle = 45° , (c) Corner angle = 30° .

Before verification, a convergence test of the particle size was performed. The test results were used to determine the appropriate particle size for additional validation simulations. The test was conducted using the G-StoP solver in a rectangular water tank with a corner angle of 90° . Four particle sizes were used in total within the range of $5 \times 10^{-3} - 2 \times 10^{-2}$. Figure 8 shows the relative error differences between the analytical solution and the present simulation according to the particle size at time $t = 5$ s. The maximum error decreased from 3% to 0.6% as the particle size decreased. In addition, within the error range of 1%, the error gradient decreased and tended to converge. Therefore, subsequent simulations used particle sizes within the range $l_0 \leq 5 \times 10^{-3}$.

Table 1. Condition of simulation for hydrostatic pressure problem.

Condition	Value		
Tilted angle (°)	90	45	30
Computational time (s)	5.0		
Number of particles	384,000	387,600	586,480
Particle length (m)	5.0×10^{-3}		
Density of fluid (kg/m ³)	1000.0		
Kinematic viscosity of fluid (m ² /s)	1.0×10^{-6}		

**Figure 8.** Relative errors of bottom pressure according to particle size.

Figures 9 and 10 show the spatial distribution of the pressure field and the time series results of the pressure measured at the center of the bottom of the tank, respectively. When the edge angle is 90°, both models exhibit stable particle placement and a stable pressure field, thus confirming that the measured values of the pressure time series are in reasonable agreement with the analytic solutions, except at extremely early times. The relative errors in the analytic solutions of the ERP and G-StoP models at the final time setting of $t = 5$ s are 0.6% and 0.7%, respectively. When the corner angle is 60°, the ERP model results show that the particles occupy all the free space near the wall and attempt to penetrate it, especially near the corner. Therefore, the pressure continuously decreases over time, even by a small proportion. When the corner angle is 30°, the particles of the ERP model pass through the walls successively and appear to disturb the entire pressure field, including the free surface, which causes a large range of oscillations in the pressure–time series results. In other words, because the ERP model is a mirror-based particle symmetry model, it is unstable near the boundary if the number of fluid particles is insufficient. Conversely, in the G-StoP model, the distance between the particle and wall is maintained at 30°, and the pressure value is consistent with the theoretical solution. The solutions of the ERP and G-StoP models at $t = 5$ s yield relative errors of approximately 4.5% and 1.8%, respectively. However, in the ERP model, an error of approximately 11.3% occurs instantaneously at $t = 4.6$ s when vibration occurs.

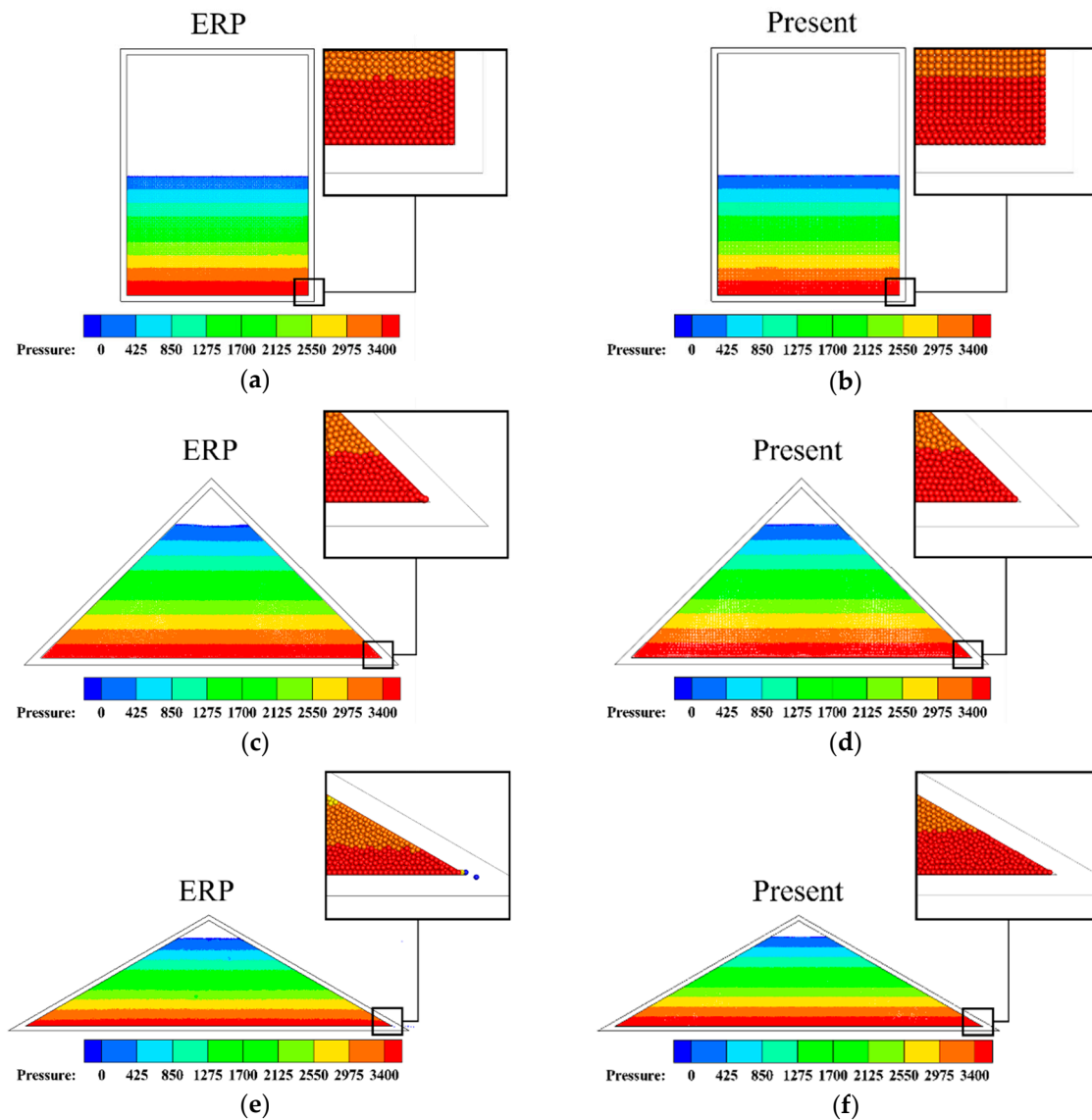
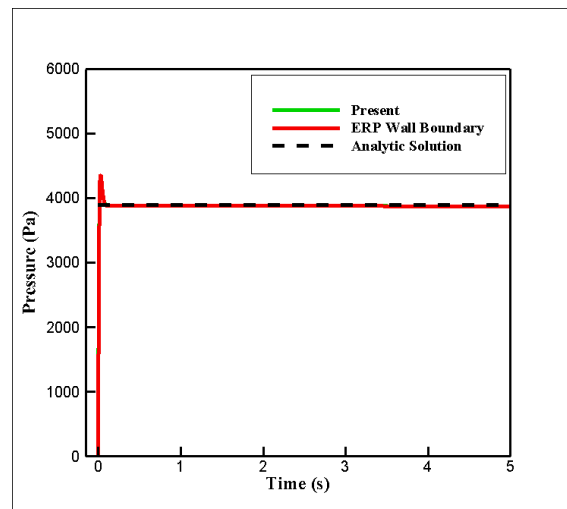
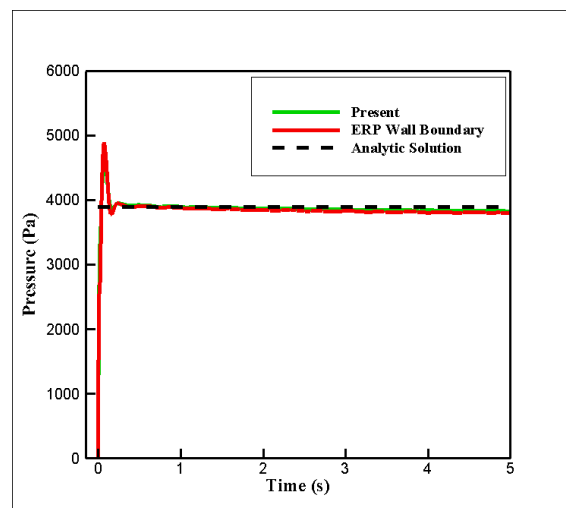


Figure 9. Distribution of pressure at $t = 5$ s. (a) ERP at corner angle = 90° , (b) G-StoP at corner angle = 90° , (c) ERP at corner angle = 45° , (d) G-StoP at corner angle = 45° , (e) ERP at corner angle = 30° , (f) G-StoP at corner angle = 30° .

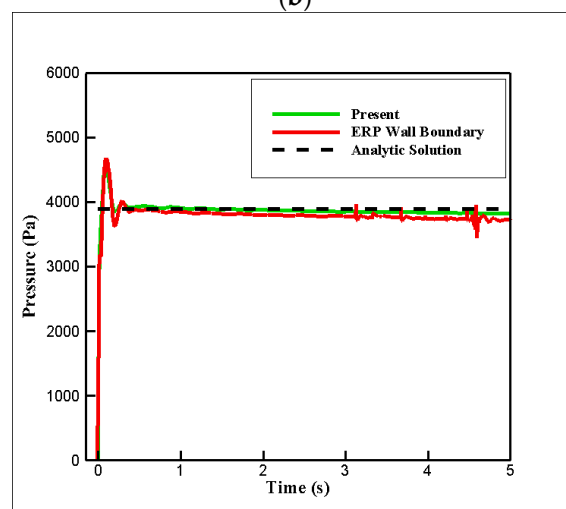
Figure 11 shows a plot of the pressure along the water depth for all particles used in the simulation when $t = 5$ s. The top of the water (depth = 0 m) represented a free surface, and the pressure increased as the depth increased because of the hydrostatic pressure effect. The ERP model results at 30° were noticeable. Overall, the measured pressure differed considerably from the analytical value. In particular, at the lower water depth, the pressure oscillated from zero to within the entire range of the analytical hydrostatic pressure, thus exhibiting an unstable appearance. The zero pressure at the bottom of the tank (i.e., atmospheric pressure) implied that the same pressure was applied to the free-surface particles. Therefore, the corresponding particle could be determined as a free-surface particle using the free-surface particle search described in Section 2.6. This could further disrupt the stability of the flow field. Conversely, in the G-StoP model, the vertical pressure distribution did not exhibit significant differences from the analytical values even when the corner angle was changed.



(a)



(b)



(c)

Figure 10. Time history of bottom pressure. (a) Corner angle = 90° , (b) Corner angle = 45° , (c) Corner angle = 30° .

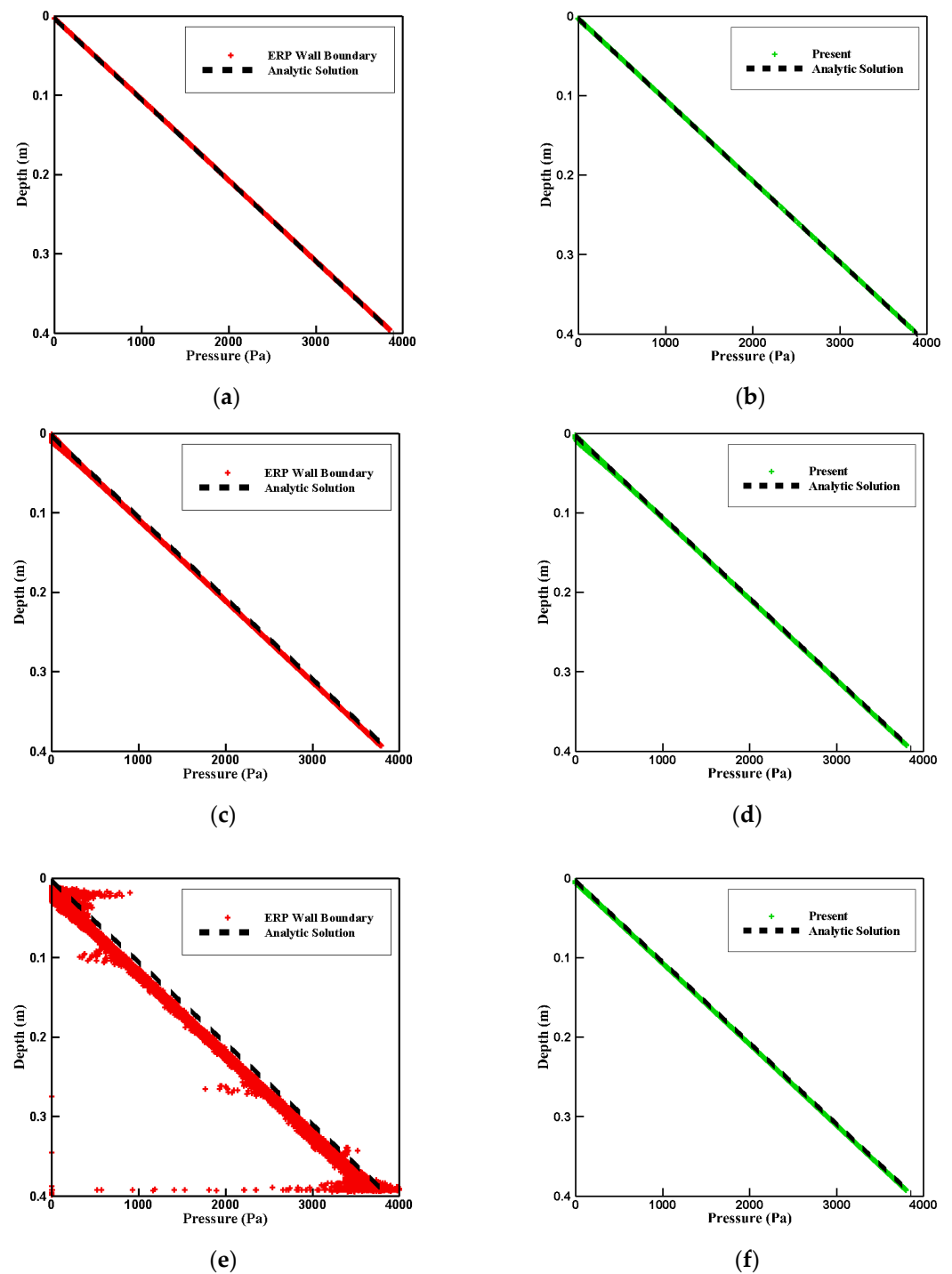


Figure 11. Depth-distributed pressure profiles of all particles. (a) ERP at corner angle = 90° , (b) G-StoP at corner angle = 90° , (c) ERP at corner angle = 45° , (d) G-StoP at corner angle = 45° , (e) ERP at corner angle = 30° , (f) G-StoP at corner angle = 30° .

4.2. Dam Breaking Problem

To verify the enhanced PNU-MPS method, the second simulation is performed on a 3D dam breaking problem owing to a collision with a cuboid obstacle. Although the obstacle may be considered to be simple as it contains only a 90° corner, it seems to be a good example for dealing with 3D interfaces with co-existing 90° concave and 135° convex angles from the viewpoint of fluid particles. The experimental data for comparison were obtained from Kleefsman et al. [32].

Table 2. Condition of simulation for dam breaking problem with obstacle.

Condition	Value
Computational time (s)	4.0
Number of particles	5,461,200 ($246 \times 111 \times 200$)
Particle length (m)	5.0×10^{-3}
Density of fluid (kg/m ³)	1000.0
Kinematic viscosity of fluid (m ² /s)	1.0×10^{-6}

Figure 13 shows the flow pattern over time by comparing it with the snapshot of the experimental results. To check the pressure field inside the fluid, only half of the area is shown by cutting off the center plane in the z-direction in the simulation. The water column initially confined within the dam collapses immediately (owing to gravity and removal of the sluice gate) and flows toward the obstacle. At $t = 0.4$ s, the wavefront of the collapsed water column reaches the anterior part of the obstacle. After colliding with the obstacle at $t = 0.6$ s, the fluid flow breaks significantly and rises vertically, and the remaining fluid moves rapidly in the horizontal direction along the narrow space on the left and right sides of the obstacle. At $t = 0.84$ s, the fluid collides with the downstream sidewalls of the tank, and a high-pressure region is generated at the corner. Notably, the evolution process of the simulated flow is expressed qualitatively and reasonably compared with that of the experiment.

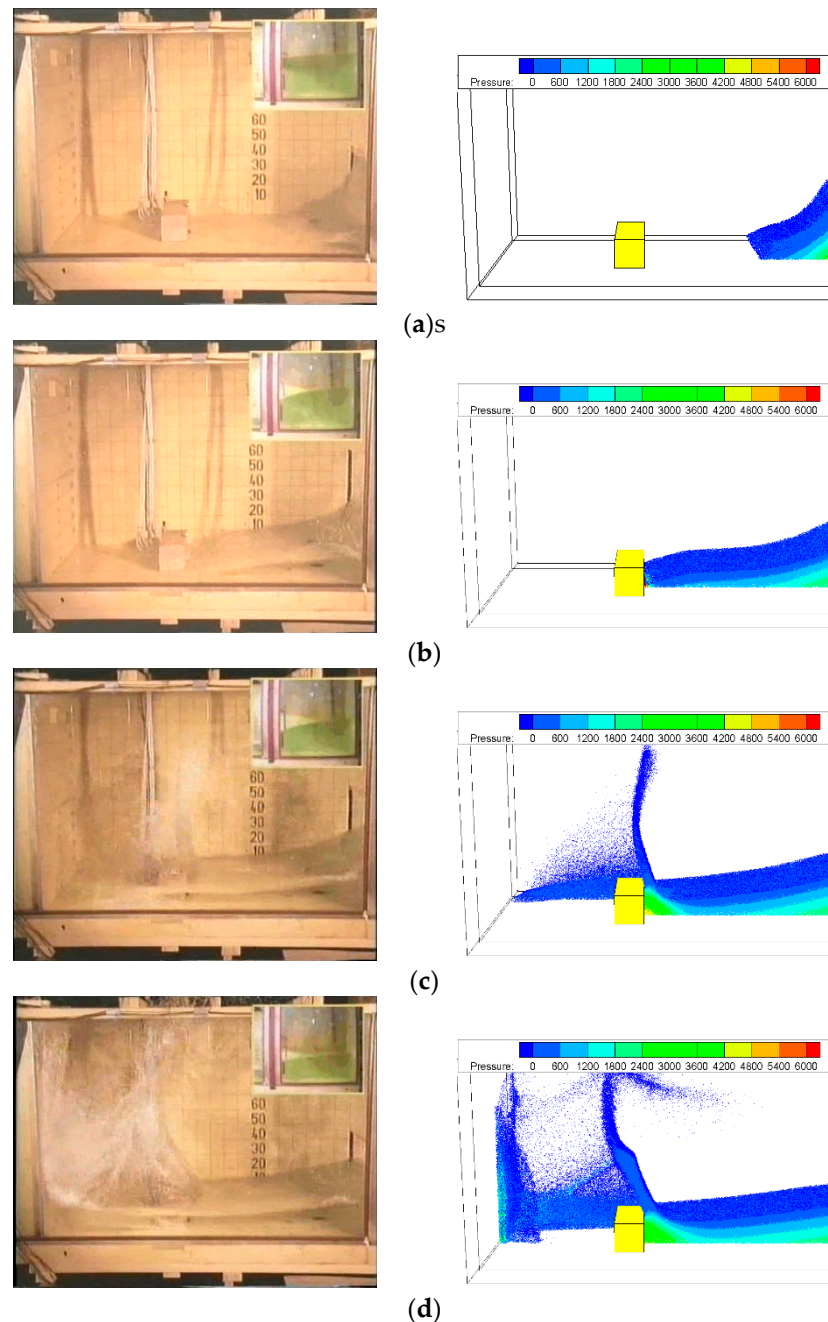


Figure 13. Comparison of water surfaces interacting with obstacle. (a) $t = 0.2$ s, (b) $t = 0.4$ s, (c) $t = 0.6$ s, (d) $t = 0.84$ s.

For a more quantitative comparison, the time series of the pressure at points P1 and P3 in front of the obstacle are shown in Figure 14, along with the experimental and simulation results of other studies [32,33]. The rise, peak value, and decay trend of the impact pressure at P1 in this simulation are relatively more consistent with the experimental results than the results of other simulations. However, the pressure peak value at P3 is slightly smaller than that in the experiment, but similar patterns are observed in other simulation results based on the volume of fluid and SPH. The results show that the trend of the overall pressure change in time is simulated accurately.

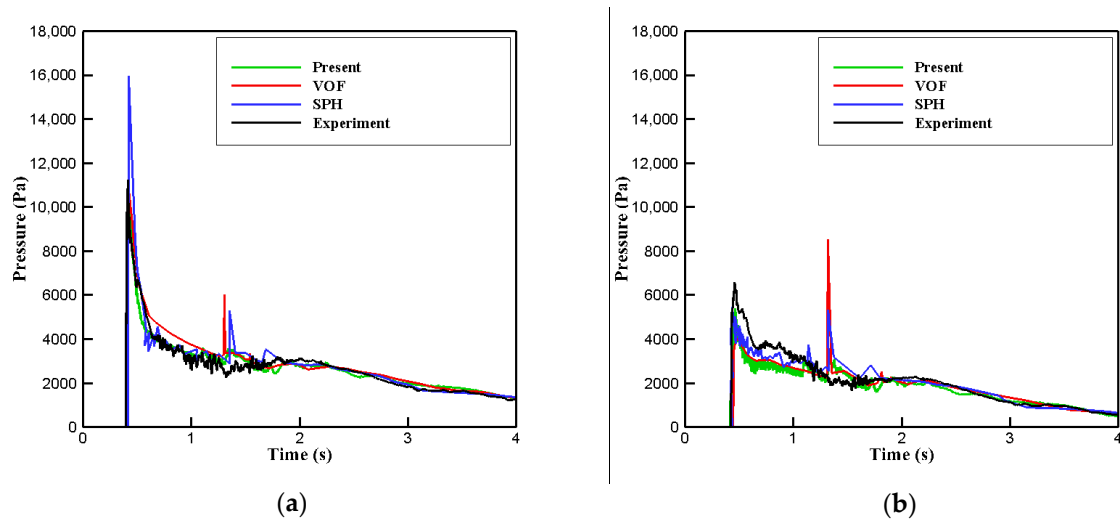


Figure 14. Comparison of pressure in time series with experiment and other simulations. (a) at P1, VOF (Kleefsman et al. [32]) and SPH (Chiron et al. [33]), (b) at P2, VOF (Kleefsman et al. [32]) and SPH (Chiron et al. [33]).

4.3. Subaerial Landslide Tsunami Generation Problem

The third validation was the subaerial landslide tsunami generation problem, which has been independently tested by Heller et al. [34]. In this example, the enhanced PNU-MPS method was interconnected with the multibody dynamics analysis software RecurDyn [17] and the two-way coupling simulation described in Section 3.1.

Figure 15 shows the initial setup of the experiment. The tank with an open top is 21 m long, 0.6 m wide, and 0.24 m deep, and includes a 45° beach slope with a side length of 1.63 m on the left downstream side. A curved shape with a radius of 0.6 m is formed between the slope and bottom of the tank. The slide is 0.599 m long, 0.12 m high, and 0.577 m deep, with a 45° frontal angle. The distance that the slide covers toward the free surface is 0.55 m. The slide is assumed to be a rigid body, and only the heave, surge, and pitch motion are considered. The friction coefficients between the rigid body of the slide and the inclined wall are set to 0.23 and 0.21 above and below the free surface, respectively. The total simulation time is 4 s and approximately 3.73 million particles are used. Table 3 summarizes the simulation conditions. The pressure measurement points on the right side of the front of the slide are P3 and P4, as in the experiment, and these two points are located 0.029 and 0.093 m from the bottom of the slide, respectively. Wave heights are measured at two points, H1 and H2, located at 1.2 and 1.8 m away from the point at which the slope and water surface meet, respectively.

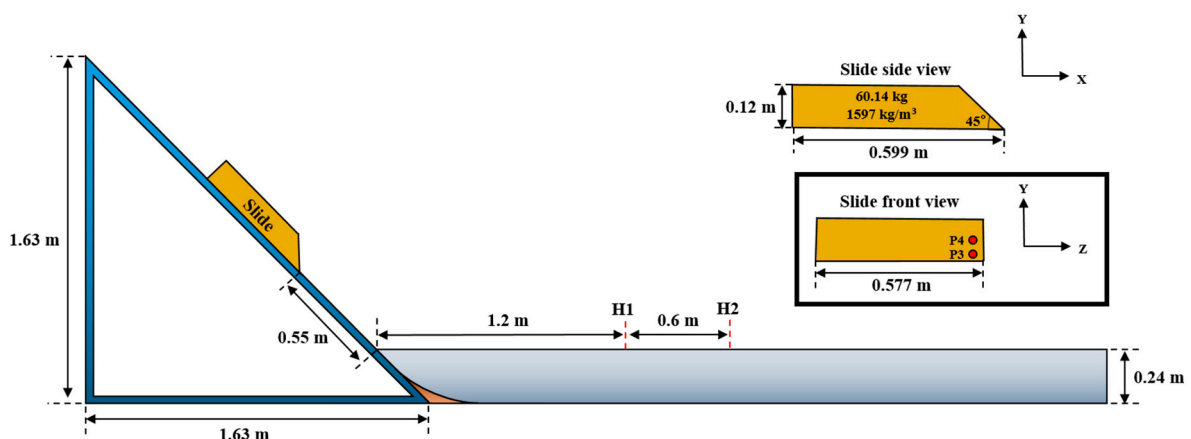


Figure 15. Schematic view of the subaerial landslide experiment.

Table 3. Condition of simulation for land-sliding problem.

Condition		Value		
Computational time (s)		4.0		
Number of particles		3,733,956		
Particle length (m)		5.0×10^{-3}		
Density of fluid (kg/m^3)		1000.0		
Kinematic viscosity of fluid (m^2/s)		1.0×10^{-6}		
Slide mass (kg)		60.14		
Slide volume (m^3)		0.038		
Inertia tensor	Ixx	2.493	Ixy	0.72
	Iyy	2.422	Iyz	−1.314
	Izz	1.583	Izx	−2.685

Figure 16 compares the velocity and position of the slide in the proposed model with the experiments and other particle-based simulations [35,36]. The present simulation results agree with the experimental results. However, a slightly different trend from the experiment can be observed when $0.5 \text{ s} < t < 0.9 \text{ s}$, as shown in Figure 16a. This is because a stronger repulsive force from the fluid acts on the slide in this simulation than in the experiment during a collision with the bottom. In addition, the present result is closer to the experimental results than other numerical results because different friction coefficients are applied to the upper and lower parts of the free surface, in consideration of the “friction reduction effect on a submerged slope” pointed out by Zhang et al. [35]. Furthermore, the slide in the experiment exhibits a backward motion for an extremely short time after it collides with the fluid at approximately $t = 0.87 \text{ s}$. Our results confirm that a slight backward motion occurs in the same way as in the experiment. Meanwhile, the position profile in Figure 16b maintains the same shape as that obtained in the experiment until $t = 0.7 \text{ s}$. However, a local difference is observed at the position where the slide finally settles. This error is approximately less than 4% and appears to be due to the above-mentioned friction-reduction effect.

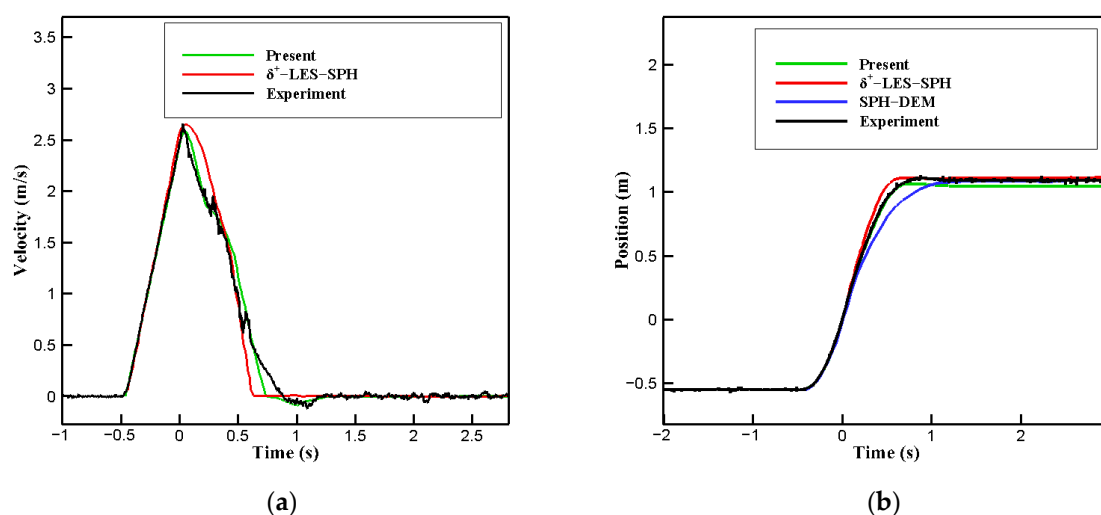


Figure 16. Comparison of velocity and position of slide with experiment and other simulations. (a) velocity, δ^+ -LES-SPH (Zhang et al. [35]), (b) position δ^+ -LES-SPH (Zhang et al. [35]), SPH-DEM (Tan et al. [36]).

Figure 17 shows the time series results of the pressure measured at points P3 and P4 on the right side of the front of the slide. The peak value generated at the time of collision

appears to be slightly larger than that in the experiment, which may be related to the weak compressibility employed to mitigate the vibration and aggregation phenomena in a particle method, similar to the results of a previous study [35]. The overall trend appears to be reasonably consistent with the experimental results. Meanwhile, in Figure 17b at $t = 0.5$ s, the pressure appears to be suddenly increased compared to that in experiment. It seems to be because a single-phase flow was applied in this simulation, not a multiphase flow that can reproduce the air cavity effect caused by the air entrainment into the slide top. In particular, in the SPH results, the vibration between $t = 0.4$ s and $t = 0.7$ s is considerable, whereas the results of this study agree well with the experimental results.

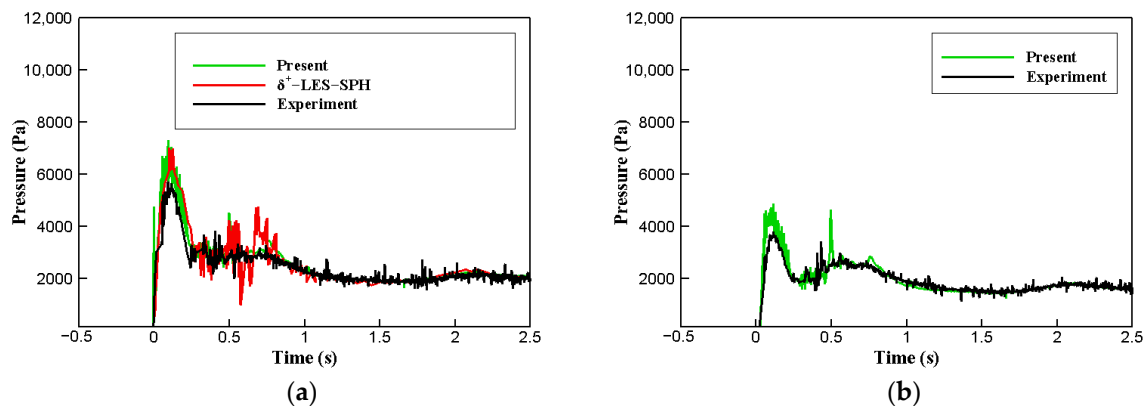


Figure 17. Comparison of pressure in time series with experiment and other simulation. (a) at P3, δ^+ -LES-SPH (Zhang et al. [35]), (b) at P4.

Figure 18 shows the wave heights measured at H1 and H2. The wave heights at the two points exhibit extremely similar tendencies to the experimental trend.

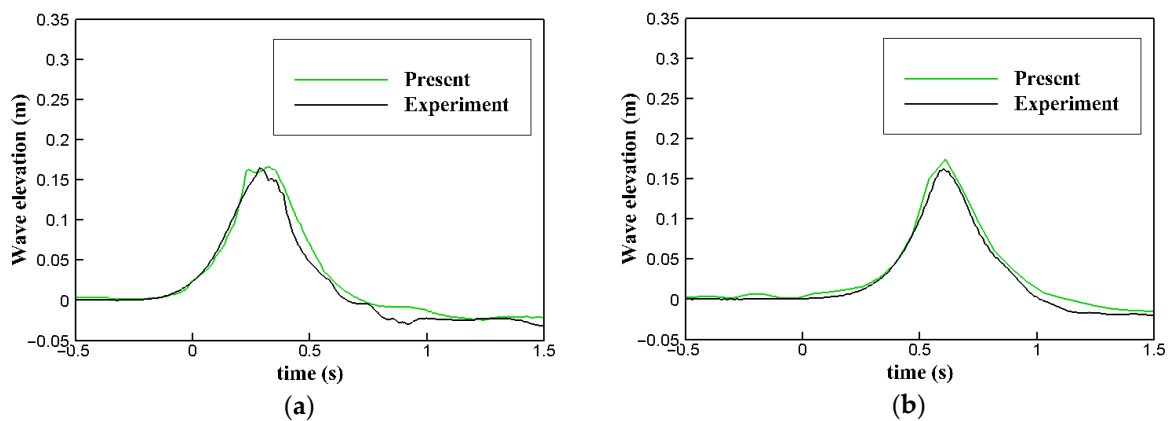


Figure 18. Comparison of wave elevation with experiment. (a) at H1, (b) at H2.

Finally, Figure 19 shows snapshots of the flow field used for comparison. At $t = 0.4$ s, the fluid particles colliding with the front wedge surface of the slide appear to rise vertically, as shown in Figure 19b. Subsequently, the ascending fluid particles descend. As shown in Figure 19c, at $t = 0.6$ s, most of them form waves by impinging on the upper part of the front part of the slide, and some of them on the inclined surface of the rear part of the slide. Overall, the simulation accurately simulated the experiment. However, the cavitation associated with air entrainment at the top and rear of the slide cannot be simulated in the present model. To simulate cavitation, a numerical algorithm for multiphase flows should be introduced in the future.

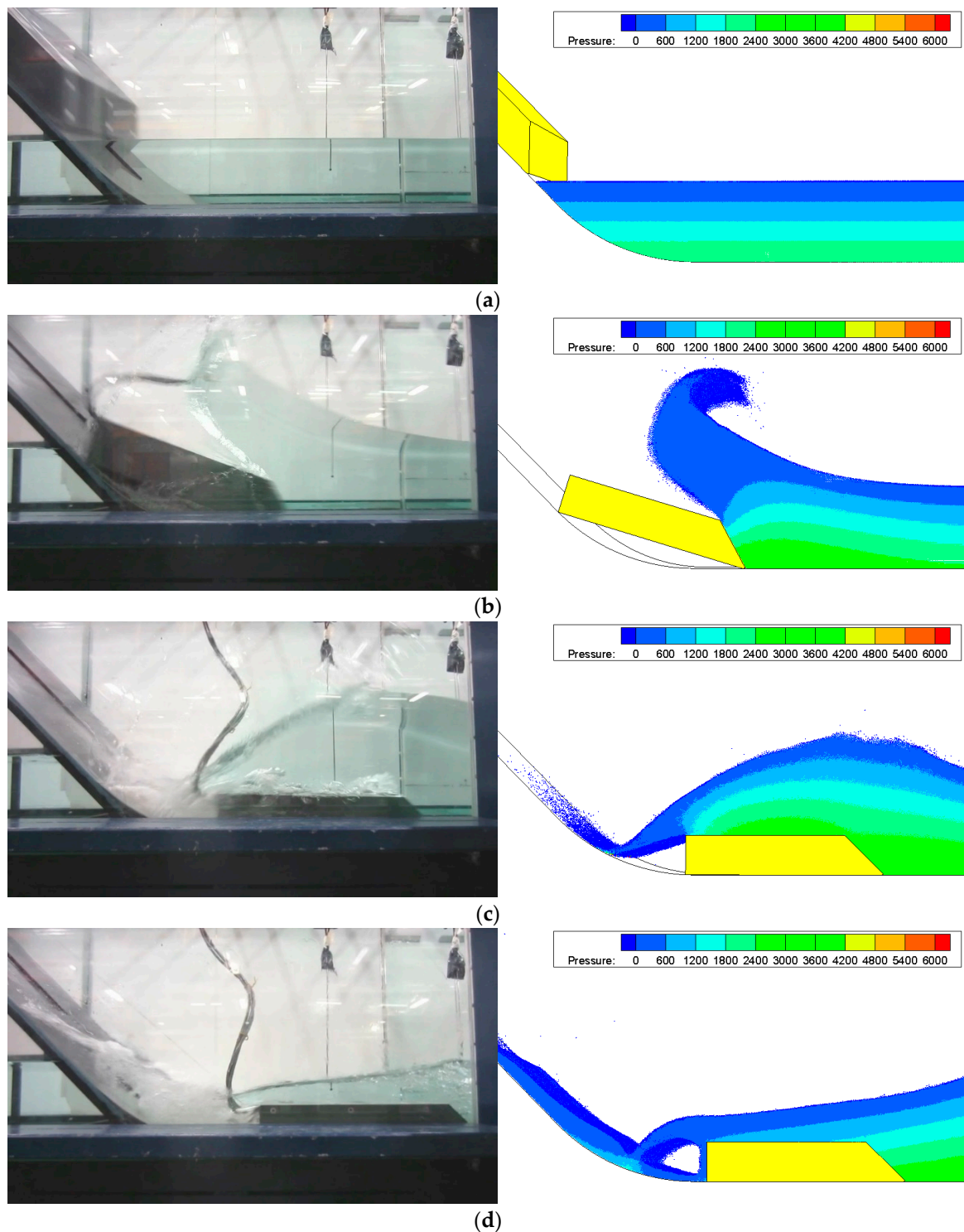


Figure 19. Comparison of water surfaces with experiment and pressure distributions. (a) $t = 0$ s, (b) $t = 0.4$ s, (c) $t = 0.6$ s, (d) $t = 1.6$ s.

4.4. Wine Sloshing Problem

Thus far, the handling of the boundary surface of an object shape with a straight surface has been verified. In this section, a boundary surface comprising curved surfaces is verified. As shown in Figure 20, let us assume a sloshing motion inside a wine glass with a periodic in-line motion with an amplitude of $A = 0.01$ m and a period of $T = 0.45$ s.

To validate the simulation, a simple visualization experiment is performed in which the sloshing behavior of wine inside the glass is filmed using a high-speed camera.



Figure 20. Schematic view of the wine glass.

The cup used in the experiment is commercially available, and the computational simulation model has the same shape, except for the handle. The polygonal mesh of the CAD shape comprises 18,432 elements. Nemea wine from Greece was used in this experiment. The average value [37] was applied to the material properties (Table 4). The filling ratio of the inner wine was set to 25% of the total height standard, the total simulation time was 3.5 s, and the number of particles used in the simulation was approximately equal to 2 million (Table 5).

Table 4. Average values for wine studied at 20 °C.

NEMEA Wine (Greece)	
Dynamic viscosity	1.75 (20 °C) mPa · s
Density	0.992 ± 0.001 (g/mL)
Alcohol content	12.4 ± 0.1 (% <i>, v/v</i>)

Table 5. Condition of simulation for wine sloshing problem.

Condition	Value
Computational time	3.5 (s)
Number of particles	2,099,658
Particle length	3.0×10^{-4} (m)
Density of fluid	992.0 (kg/m ³)
Kinematic viscosity of fluid	1.75×10^{-6}
Amplitude	0.01 m
Oscillation period	0.45 (s)

Figure 21 shows the time evolution of wine sloshing. The flow characteristics in the simulation are qualitatively comparable to those in the experiment. Although conducting a direct and detailed comparison was challenging because of the small bit of three-dimensionality, we observed a high degree of similarity between the motional patterns of the free surface in the experiment and simulation. In particular, the flow going up

the left wall at $t = 0.73$ s and the wave overturning by the turbulent bore at $t = 2.42$ s were extremely similar in the simulations and experiments.

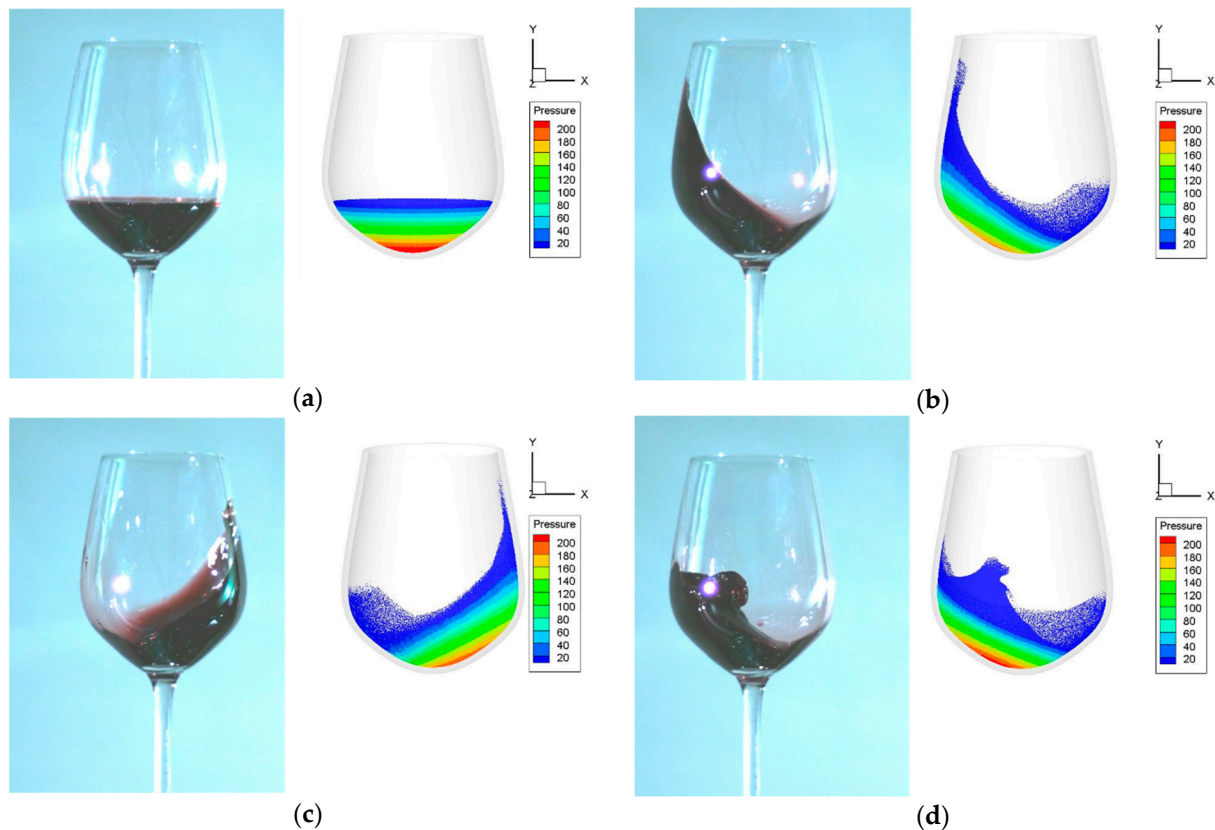


Figure 21. Comparison of experimental wine sloshing surfaces and numerical pressure distributions. (a) $t = 0$ s, (b) $t = 0.73$ s, (c) $t = 0.94$ s, (d) $t = 2.42$ s.

Thus, the proposed G-StoP model for treating object boundaries can be simply and robustly applied to curved body shapes and corners of various angles.

5. Conclusions

In this study, a new wall boundary condition, the grid-stamping on polygon (G-StoP) model, has been proposed. The model can conveniently and efficiently treat the boundary of an arbitrarily complex-shaped body by analyzing the flow using the PNU-MPS method of a particle-based fluid solver. To confirm the stability and robustness of the proposed model, numerical simulations were performed for four problems: hydrostatic pressure testing in a tank with various corner edges, dam breaking, subaerial landslide tsunami generation, and wine sloshing. The results obtained for each problem are summarized as follows:

- Hydrostatic pressure simulations were conducted in a rectangular tank with various corner angles. Although the conventional ERP model exhibited a large numerical fluctuation and was unstable when the corner angle was reduced, the proposed model exhibited excellent robustness and safety without any special treatment. For a more quantitative comparison, the hydrostatic pressure was compared with the results of the ERP model and the analytic solution. The result shows that, when the corner angle was 30° , the maximum errors of the ERP and G-StoP models were 11.3% and 1.8%, respectively.
- In the dam breaking simulations, the time series of the pressure in front of the obstacle was compared with other simulations and experimental results. The rise time, peak value, and decay trend of the impact pressure matched the results of the other sim-

ulations relatively well. In addition, the flow development process was expressed qualitatively and compared reasonably with experimental results.

- In the subaerial landslide tsunami generation, the slide velocity, position, pressure, and wave height results were consistent with the experimental data. However, relative errors of approximately 4% in the experiment occurred at the position at which the slide finally settled. This may be attributed to the repulsive force from the fluid acting on the slide being stronger in this simulation than in the experiment during a collision with the bottom. Meanwhile, currently, the effect of cavitation associated with air bubbles at the top and back of the slide was not simulated. Moreover, a model that can consider this type of multiphase phenomenon should be adopted in the future.
- Finally, for the wine sloshing problem, a simple visualization experiment was performed on an in-line periodic motion with a wine glass partially filled with wine. Notably, the simulation represented qualitatively similar behavior to the free surface of wine in a glass with a curved boundary compared with the experiment.

The findings of this study are potential tools for solving various problems in complex fluid systems at future industrial sites. In addition, the proposed method is expected to be widely used in the field of applied multiphysics analysis involving frequent interaction for complex shapes or multiple rigid bodies.

Author Contributions: Conceptualization, H.-S.S. and J.-C.P.; methodology, H.-S.S. and J.-C.P.; validation, H.-S.S.; investigation, H.-S.S.; data curation, H.-S.S.; writing—original draft preparation, H.-S.S.; writing—review and editing, J.-C.P.; supervision, J.-C.P. 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: Not applicable.

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

References

1. Sarrate, J.; Huerta, A.; Donea, J. Arbitrary Lagrangian-Eulerian formulation for fluid-rigid body interaction. *Comput. Methods Appl. Mech. Eng.* **2001**, *190*, 3171–3188. [\[CrossRef\]](#)
2. Monaghan, J.J. An Introduction to Sph. *Comput. Phys. Commun.* **1988**, *48*, 89–96. [\[CrossRef\]](#)
3. Koshizuka, S.; Oka, Y. Moving-particle semi-implicit method for fragmentation of incompressible fluid. *Nucl. Sci. Eng.* **1996**, *123*, 421–434. [\[CrossRef\]](#)
4. Marrone, S.; Antuono, M.; Colagrossi, A.; Colicchio, G.; Le Touze, D.; Graziani, G. δ -SPH model for simulating violent impact flows. *Comput. Methods Appl. Mech. Eng.* **2011**, *200*, 1526–1542. [\[CrossRef\]](#)
5. Adami, S.; Hu, X.Y.; Adams, N.A. A generalized wall boundary condition for smoothed particle hydrodynamics. *J. Comput. Phys.* **2012**, *231*, 7057–7075. [\[CrossRef\]](#)
6. Akimoto, H. Numerical simulation of the flow around a planing body by MPS method. *Ocean Eng.* **2013**, *64*, 72–79. [\[CrossRef\]](#)
7. Liu, X.; Lin, P.; Shao, S. An ISPH simulation of coupled structure interaction with free surface flows. *J. Fluids Struct.* **2014**, *48*, 46–61. [\[CrossRef\]](#)
8. Monaghan, J.J. Simulating Free-Surface Flows with Sph. *J. Comput. Phys.* **1994**, *110*, 399–406. [\[CrossRef\]](#)
9. Monaghan, J.J.; Kajtar, J.B. SPH particle boundary forces for arbitrary boundaries. *Comput. Phys. Commun.* **2009**, *180*, 1811–1820. [\[CrossRef\]](#)
10. Zheng, X.; Lv, X.; Ma, Q.; Duan, W.; Khayyer, A.; Shao, S. An improved solid boundary treatment for wave–float interactions using ISPH method. *Int. J. Nav. Archit. Ocean Eng.* **2018**, *10*, 329–347. [\[CrossRef\]](#)
11. Antuono, M.; Pilloton, C.; Colagrossi, A.; Durante, D. Clone particles: A simplified technique to enforce solid boundary conditions in SPH. *Comput. Methods Appl. Mech. Eng.* **2023**, *409*, 115973. [\[CrossRef\]](#)
12. Mitsume, N.; Yoshimura, S.; Murotani, K.; Yamada, T. Explicitly represented polygon wall boundary model for the explicit MPS method. *Comput. Part. Mech.* **2015**, *2*, 73–89. [\[CrossRef\]](#)
13. Fourtakas, G.; Jose, M.D.; Vacondio, R.; Rogers, B.D. Local uniform stencil (LUST) boundary condition for arbitrary 3-D boundaries in parallel smoothed particle hydrodynamics (SPH) models. *Comput. Fluids* **2019**, *190*, 346–361. [\[CrossRef\]](#)

14. Vacondio, R.; Rogers, B.D.; Stansby, P.K. Smoothed Particle Hydrodynamics: Approximate zero-consistent 2-D boundary conditions and still shallow-water tests. *Int. J. Numer. Methods Fluids* **2012**, *69*, 226–253. [CrossRef]
15. Roth, S.D. Ray Casting for Modeling Solids. *Comput. Vis. Graph* **1982**, *18*, 109–144. [CrossRef]
16. Lee, B.H.; Park, J.C.; Kim, M.H.; Hwang, S.C. Step-by-step improvement of MPS method in simulating violent free-surface motions and impact-loads. *Comput. Methods Appl. Mech. Eng.* **2011**, *200*, 1113–1125. [CrossRef]
17. RecurDyn Manual, Version V9R4. Available online: <https://functionbay.com/documentation/onlinehelp/default.htm#!Documents/introduction.htm> (accessed on 22 February 2023).
18. Yun, S.M.; Kim, S.P.; Chung, S.M.; Shin, W.J.; Cho, D.S.; Park, J.C. Structural Safety Assessment of Connection between Sloshing Tank and 6-DOF Platform Using Co-Simulation of Fluid and Multi-Flexible-Body Dynamics. *Water* **2020**, *12*, 2108. [CrossRef]
19. Khayyer, A.; Gotoh, H. A higher order Laplacian model for enhancement and stabilization of pressure calculation by the MPS method. *Appl. Ocean Res.* **2010**, *32*, 124–131. [CrossRef]
20. Khayyer, A.; Gotoh, H. Enhancement of stability and accuracy of the moving particle semi-implicit method. *J. Comput. Phys.* **2011**, *230*, 3093–3118. [CrossRef]
21. Kondo, M.; Koshizuka, S. Improvement of stability in moving particle semi-implicit method. *Int. J. Numer. Methods Fluids* **2011**, *65*, 638–654. [CrossRef]
22. Shibata, K.; Masaie, I.; Kondo, M.; Murotani, K.; Koshizuka, S. Improved pressure calculation for the moving particle semi-implicit method. *Comput. Part. Mech.* **2015**, *2*, 91–108. [CrossRef]
23. Wang, L.Z.; Jiang, Q.; Zhang, C.K. Improvement of moving particle semi-implicit method for simulation of progressive water waves. *Int. J. Numer. Methods Fluids* **2017**, *85*, 69–89. [CrossRef]
24. Vonneumann, J.; Richtmyer, R.D. A Method for the Numerical Calculation of Hydrodynamic Shocks. *J. Appl. Phys.* **1950**, *21*, 232–243. [CrossRef]
25. Jeong, S.M.; Park, J.I.; Park, J.C. Numerical Simulation of 2-D Solitary Wave Run-Up over Various Slopes Using a Particle-Based Method. *Water* **2019**, *11*, 462. [CrossRef]
26. Duan, G.T.; Koshizuka, S.; Yamaji, A.; Chen, B.; Li, X.; Tamai, T. An accurate and stable multiphase moving particle semi-implicit method based on a corrective matrix for all particle interaction models. *Int. J. Numer. Methods Eng.* **2018**, *115*, 1287–1314. [CrossRef]
27. Hestenes, M.R.; Stiefel, E. Methods of Conjugate Gradients for Solving Linear Systems. *J. Res. Natl. Bur. Stand.* **1952**, *49*, 409–436. [CrossRef]
28. Monaghan, J.J.; Gingold, R.A. Shock Simulation by the Particle Method Sph. *J. Comput. Phys.* **1983**, *52*, 374–389. [CrossRef]
29. Oger, G.; Doring, M.; Alessandrini, B.; Ferrant, P. An improved SPH method: Towards higher order convergence. *J. Comput. Phys.* **2007**, *225*, 1472–1492. [CrossRef]
30. Ren, D. Development of Numerical Analysis System for Predicting Ice-Breaking Resistance Based on Lagrangian Meshless Method. Ph.D. Thesis, Pusan National University, Busan, Republic of Korea, 2020.
31. Rodrigues, O. Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendants des causes qui peuvent les produire. *J. Math. Pures Appl.* **1840**, *5*, 380–440.
32. Kleefsman, K.M.T.; Fekken, G.; Veldman, A.E.P.; Iwanowski, B.; Buchner, B. A Volume-of-Fluid based simulation method for wave impact problems. *J. Comput. Phys.* **2005**, *206*, 363–393. [CrossRef]
33. Chiron, L.; de Lefte, M.; Oger, G.; Le Touze, D. Fast and accurate SPH modelling of 3D complex wall boundaries in viscous and non viscous flows. *Comput. Phys. Commun.* **2019**, *234*, 93–111. [CrossRef]
34. Heller, V.; Bruggemann, M.; Spinneken, J.; Rogers, B.D. Composite modelling of subaerial landslide-tsunamis in different water body geometries and novel insight into slide and wave kinematics. *Coast. Eng.* **2016**, *109*, 20–41. [CrossRef]
35. Zhang, G.B.; Chen, J.Y.; Qi, Y.T.; Li, J.; Xu, Q. Numerical simulation of landslide generated impulse waves using a δ^+ -LES-SPH model. *Adv. Water Resour.* **2021**, *151*, 103890. [CrossRef]
36. Tan, H.; Xu, Q.; Chen, S.H. Subaerial rigid landslide-tsunamis: Insights from a block DEM-SPH model. *Eng. Anal. Bound. Elem.* **2018**, *95*, 297–314. [CrossRef]
37. Yanniotis, S.; Kotseridis, G.; Orfanidou, A.; Petraki, A. Effect of ethanol, dry extract and glycerol on the viscosity of wine. *J. Food Eng.* **2007**, *81*, 399–403. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.