Title: Shared-Memory Parallelization of Consistent Particle Method for Violent Wave Impact Problems

Authors: M. Luo, C.G. Koh

PII: S0141-1187(17)30054-8
DOI: https://doi.org/10.1016/j.apor.2017.09.013
Reference: APOR 1441

To appear in:

Received date: 24-1-2017
Revised date: 24-8-2017
Accepted date: 30-9-2017

Shared-Memory Parallelization of Consistent Particle Method for Violent Wave Impact Problems

M. Luo[*] and C. G. Koh

Department of Civil and Environmental Engineering, National University of Singapore, Singapore 117576

[*] Corresponding author: Tel.: +65 6516 2163; fax: +65 6779 1635.

E-mail address: luomin@u.nus.edu, cveluom@163.com

Highlight

- A novel particle method is presented for violent wave impact problems
- Laplacian and gradient operators are computed based on Taylor series expansion
- Incompressible water and compressible air are simulated in a strongly-coupled way
- OpenMP parallelization is implemented to significantly improve computational efficiency
- Experimental studies of three-dimensional water sloshing and sloshing impact with entrapped air pocket are conducted on a rotational simulator for validation
- The natural frequency of an air pocket entrapped by wave impact is derived

Abstract

A shared-memory parallelization is implemented to the recently developed Consistent Particle Method (CPM) for violent wave impact problems. The advantages of this relatively new particle method lie in four key aspects: (1) accurate computation of Laplacian and gradient operators based on Taylor series expansion, alleviating spurious pressure fluctuation and being able to model two-phase flows characterized by large density difference, (2) a thermodynamics-based compressible solver for modelling compressible air that eliminates the need of determining artificial sound speed, (3) seamless coupling of the compressible air solver and incompressible water solver, and (4) parallelization of the numerical model based on Open Multi-Processing (OpenMP) and a parallel direct sparse solver (Pardiso) to significantly improve computational efficiency. Strong and weak scaling analyses of the parallelized CPM are conducted, showing an efficiency speedup of 100 times or more depending on the size of simulated problem. To demonstrate the accuracy of the developed numerical model, three numerical examples are studied including the benchmark study of wave impact on seawall, and our experimental studies of violent water sloshing under rotational excitations and sloshing impact with entrapped air pocket. CPM is shown to accurately capture highly deformed breaking waves and violent wave impact pressure including pressure oscillation induced by air cushion effect.

Keywords: particle method; wave impact; shared-memory; two-phase flow; air compressibility

1. Introduction

Modelling of wave impact on structures is of great practical interest in offshore and marine engineering, e.g. the design of seawalls against tsunami waves and the design of ocean platform structures against violent wave impact. With the rapid advances of computer power, many numerical methods have been developed to predict the wave profile and impact forces [1-5]. Most of these studies do not consider the presence of entrapped air pockets, or treat the air pockets as incompressible. While incompressibility is a reasonable assumption in some water-air flow scenarios, air entrapment or entrainment may be generated in some other problems such as violent wave impact on structures [6-8]. The compressibility of entrapped air pockets plays an significant role in influencing the pressure peak and impact duration in a wave impact process [9, 10]. Therefore, it is necessary to include air compressibility in the simulation of wave impact problems with entrapped air pocket.

The numerical difficulties to model violent wave impact problems with entrapped air pockets include the large and discontinuous deformation of fluid, the abrupt discontinuity of fluid properties (density, viscosity and compressibility) at the interface between water and air, and the integrated simulation of water and air which behave very differently (water incompressible and air compressible). To address these issues, the mesh-based methods have been developed such as Finite Difference Method [11, 12] and Finite Volume Method [13]. Because meshes are required, these methods may have difficulties to model highly deformed waves which involve fragmentation and coalescence. In addition, since these methods normally model the real fluid interface as a transition zone whose densities change gradually from the density of one fluid to another, they are difficult to reproduce a sharp fluid interface (termed as “dispersion of fluid interface” in the literature).

To better simulate large wave deformation, particle methods have been utilized and are becoming more and more popular such as Smoothed Particle Hydrodynamics (SPH) [14-16], Incompressible Smoothed Particle Hydrodynamics (ISPH) [17-19] and Moving Particle Semi-implicit (MPS) [4, 20]. For two-phase flow simulation, large numerical errors exist in the computation of spatial derivatives (a key step in particle method) because of the abrupt density change across fluid interface, which further make the numerical simulation very unstable. To address this issue, density or smearing schemes (similar to the transition zone in mesh-based methods) are adopted in some particle methods. Although improving numerical stability, these schemes produce unphysical dispersion of fluid interface.

To enhance the computation of gradient and Laplace operators, the Consistent Particle Method (CPM) was proposed by using Taylor series expansion to compute the spatial derivatives [21]. This method has explicit mathematical consistency and hence better accuracy compared to the kernel/weighting-based derivative computation schemes, particularly when particle distribution is very irregular (inevitable in violent situations). CPM was extended by Luo *et al.* [22] to deal with the abrupt change of fluid properties across fluid interface. Without using any smoothing or filtering scheme, the CPM can reproduce a sharp fluid interface such as water-air interface involving orders-of-magnitude difference in fluid properties. CPM with these two improvements is able to model incompressible free surface flows or two-phase flows characterized by large density difference. Moreover, a thermodynamically-consistent compressible solver is developed that can be integrated with the developed incompressible solver seamlessly [23]. This leads to a numerical model that can simulate incompressible-compressible two-phase flows in the same framework.

While having some advantages to model highly-deformed free-surface flows and two-phase flows, particle methods require significant computational resources [24], because (1) a fine resolution of particles is required to accurately capture the large wave deformation and (2) it is still difficult to implement the varying spatial discretization schemes [25]. Therefore, it is essential to enhance the computational efficiency of a particle method code by parallel computing.

In this context, CPM with OpenMP parallel computing is presented in this paper. After demonstrating the speedup and parallel efficiency of the parallelized code, three wave impact problems are studied: wave impact on sea wall, violent three-dimensional (3D) sloshing under rotational excitations, and sloshing impact with entrapped air pocket. In addition to comparing with published results, experimental studies are conducted for validation purpose.

2. Governing equations

The governing equations for viscous Newtonian fluids (both incompressible and compressible) in a two-fluid system are the Navier-Stokes equations as follows:

$$\frac{1}{\rho}\frac{D\rho}{Dt}+\nabla\cdot\mathbf{v}=0 \tag{1}$$

$$\frac{D\mathbf{v}}{Dt}=-\frac{1}{\rho}\nabla p+\frac{1}{\rho}\nabla\cdot\left[\mu\left(\nabla\mathbf{v}+\left(\nabla\mathbf{v}\right)^{T}\right)\right]+\mathbf{g} \tag{2}$$

where ρ is the density of fluid, $\mathbf{v}$ the particle velocity vector, p the fluid pressure, μ the dynamic viscosity of fluid and $\mathbf{g}$ the acceleration induced by body force. The water and air densities at the standard temperature and pressure condition, i.e. 1000 kg/m^3 and 1.204 kg/m^3, as well as the initial air pressure of 1.013×10^5 Pa are adopted. The real values of dynamic viscosities for water and air, i.e. 10^{-3} Pa·s and 1.983×10^{-5} Pa·s respectively, are used in the numerical simulations of this study.

3. CPM formulations

3.1. Two-step semi-implicit solution

For both incompressible and compressible fluids, the governing equations are solved by a predictor-corrector scheme [17, 20]. In the predictor step, the temporary particle velocities and positions are computed by neglecting the pressure gradient term as follows

$$\mathbf{v}^{*} = \mathbf{v}^{(k)} + \left\{ \frac{1}{\rho} \nabla \cdot \left[\mu \left(\nabla \mathbf{v}^{(k)} + \left(\nabla \mathbf{v}^{(k)} \right)^{T} \right) \right] + \mathbf{g} \right\} \Delta t \tag{3}$$

$$\mathbf{r}^{*} = \mathbf{r}^{(k)} + \mathbf{v}^{*} \Delta t \tag{4}$$

where $\Delta t = t^{(k+1)} - t^{(k)}$, $\mathbf{v}^{(k)} = \mathbf{v}(\mathbf{r}^{(k)},\ t^{(k)})$, and $\mathbf{v}^{(k)}$ and $\mathbf{r}^{(k)}$ are the particle velocity and position at time $t^{(k)}$.

In the corrector step, a pressure Poisson equation (PPE) can be derived as follows

$$\nabla \cdot \left(\frac{1}{\rho^{*}} \nabla p^{(k+1)} \right) = \frac{1}{\Delta t^{2}} \frac{\rho^{(k+1)} - \rho^{*}}{\rho^{(k+1)}} \tag{5}$$

For incompressible fluids, the incompressibility condition $\rho^{(k+1)} = \rho_0$, where ρ_0 is the initial fluid density, is imposed on the right-hand side of Equation (5). Within the influence radius r_e ($r_e = 2.1L_0$ is consistently adopted in this study according to [21], where L_0 is the initial particle spacing), the intermediate fluid density ρ^{*} for particle *i* is computed as [26]

$$\rho_i^{*} = m_i \sum_{j} w\left(r_{ij}\right) \tag{6}$$

where r_{ij} is the distance between the reference particle (*i*) and neighbor particle (*j*) based on the intermediate particle positions, and m_i the mass of particle *i* (fixed during simulation). In two-dimensional (2D) simulations, $w\left(r_{ij}\right)$ is a weighting function defined as follows [27]

$$w\left(r_{ij}\right) = \begin{cases} \dfrac{4\alpha_d}{r_e^2} \exp\left(-\left(\dfrac{2r_{ij}}{r_e} \right)^2 \dfrac{1}{\sigma} \right), & 0 < r_{ij} < r_c \\ \dfrac{1}{\pi r_e^2} \left(\dfrac{r_e}{r_{ij}} - 1 \right) & , \ r_c \le r_{ij} \le r_e \\ 0 & , \ r_{ij} > r_e \end{cases} \tag{7}$$

where r_c is the connection point of the spline and is selected to be $0.02r_e$ such that $w\left(r_{ij}\right)$ has a very large but finite value when two particles are very close. Based on the continuity and partition of unity conditions

[28], the coefficients α_d and σ are computed to be 27.4975 and 3.2×10^{-4} respectively. In 3D simulations, $w\left(r_{ij}\right)$ is defined as follows [29]

$$w\left(r_{ij}\right)=\begin{cases}\dfrac{6\alpha_d}{r_e^3}\exp\left(-\left(\dfrac{2r_{ij}}{r_e}\right)^2\dfrac{1}{\sigma}\right), & 0<r_{ij}<r_c\\[2ex] \dfrac{3}{2\pi r_e^3}\left(\dfrac{r_e}{r_{ij}}-1\right) & , \; r_c\le r_{ij}\le r_e\\[2ex] 0 & , \; r_{ij}>r_e\end{cases} \tag{8}$$

where $r_c=0.02r_e$, $\alpha_d=78.1336$ and $\sigma=5.34\times10^{-4}$. Letting the initial density of particle *i* to be ρ_{0i}, m_i can be evaluated based on the initial particle distribution as:

$$m_i=\rho_{0i}\Big/\sum_j w\left(r_{ij}\right) \tag{9}$$

For compressible fluids, although a similar approach is used to evaluate fluid density, a slow-slope weighting functions whose value at r = 0 is smaller is adopted to allow more compressibility of fluid as follows [23]

$$w_2\left(r_{ij}\right)=\begin{cases}1-6\left(\dfrac{r}{r_e}\right)^2+8\left(\dfrac{r}{r_e}\right)^3-3\left(\dfrac{r}{r_e}\right)^4 & , \; 0\le r\le r_e\\[2ex] 0 & , \; r_e<r\end{cases} \tag{10}$$

Another distinct feature in the simulation of compressible flows is that, without the incompressibility condition, the fluid density $\rho^{(k+1)}$ in Equation (5) should be treated as unknown (more details will be presented later).

Applying the derivate approximation scheme presented in Section 3.2 to the left-hand sides of Equations (5) and (21), a system of linear equations with sparse and non-symmetric coefficients can be assembled and solved by parallel solvers, to obtain the fluid pressure in the entire computational domain. Using the fluid pressure, particle velocities and positions are updated as

$$\mathbf{v}^{(k+1)}=\mathbf{v}^{*}-\left(\frac{\nabla p}{\rho}\right)^{(k+1)}\Delta t \tag{11}$$

and

$$\mathbf{r}^{(k+1)}=\mathbf{r}^{(k)}+\mathbf{v}^{(k+1)}\Delta t \tag{12}$$

where the pressure gradient term is computed by Equation (18). Since acoustic wave is not important in the simulation of water wave impact, the Courant–Friedrichs–Lewy (CFL) condition for numerical study considers only the fluid particle velocity [30, 31]. Hence, the time step Δt is governed by

$$\frac{v_{\max}\Delta t}{L_0} \le C_{\max} \tag{13}$$

where $C_{\max}$ is selected to be 0.2, and $v_{\max}$ is the maximum particle velocity (of both water and air domains) at the previous step of computation.

3.2. Derivative computation based on Taylor series expansion

Taylor series expansion for a smooth function *f*(*x*) in the vicinity of a reference particle (x_0, y_0) can be expressed as

$$f(x,y) = f_0 + hf_{,x0} + kf_{,y0} + \frac{1}{2}h^2 f_{,xx0} + hkf_{,xy0} + \frac{1}{2}k^2 f_{,yy0} + O(r^3) \tag{14}$$

where $h = x - x_0$, $k = y - y_0$, $f_0 = f(x_0, y_0)$, $f_{,x0}$ is the first order derivative of function *f* with respect to *x* at (x_0, y_0) and $f_{,xy0}$ the second-order derivative of function *f* with respect to *x* and *y* at (x_0, y_0). Writing Equation (14) for each of the neighbor particles, the following equation system can be obtained

$$[\mathbf{A}]\{D\mathbf{f}\} - \{\mathbf{f}\} = 0 \tag{15}$$

where $[\mathbf{A}]$ is a function of relative particle positions (i.e. *h* and *k*), $\{\mathbf{f}\}$ a combination of the variable differences between the reference particle and its neighbor particles (i.e. $f - f_0$), and $\{D\mathbf{f}\}$ a vector including all the derivatives in Equation (14). Solving Equation (15) by a weighted-least-square approach, the first- and second-order derivatives can be directly obtained as [21]

$$\frac{\partial p_i}{\partial x} = \sum_{j \ne i}\left[w_j^2\left(a_1 h_j + a_2 k_j + 0.5a_3 h_j^2 + a_4 h_j k_j + 0.5a_5 k_j^2\right)\left(p_j - p_i^{'}\right)\right] = \sum_{j \ne i}\left[C_{1j}\left(p_j - p_i^{'}\right)\right] \tag{16}$$

and

$$\frac{\partial^2 p_i}{\partial x^2} = \sum_{j \ne i}\left[w_j^2\left(c_1 h_j + c_2 k_j + 0.5c_3 h_j^2 + c_4 h_j k_j + 0.5c_5 k_j^2\right)\left(p_j - p_i\right)\right] = \sum_{j \ne i}\left[C_{3j}\left(p_j - p_i\right)\right] \tag{17}$$

where w_j is the weighting function used in the weighted-least-square scheme to solve an over-determined equation system, p_i and p_j the pressures on particles *i* and *j* respectively, and *a* and *c* the coefficients generated by the weighted-least-square scheme (refer to Equation (21) in [21]). Computing derivatives using Taylor series expansion, this scheme has been demonstrated to model single-phase flows with good accuracy [21, 27]. It is noted that the weighting function in CPM formulations (i.e. w_j) comes from the weighted-least-square solving scheme and is essentially different from the kernel function in SPH and ISPH and the weighting function in MPS.

In the simulation of two-phase flows characterized by large density difference, the scheme presented in the previous section produces non-negligible errors at fluid interface because of the abrupt change of fluid density and hence fluid pressure [22]. Applying the generalized finite difference scheme to the normalized pressure gradient term (i.e. $\nabla p / \rho$), which is of the same order of magnitude in two fluids of a general dynamic problem, the formulation to compute the gradient and Laplacian operators with abrupt density discontinuity has been derived by Luo *et al.* [22]

$$\left(\frac{1}{\rho}\frac{\partial p}{\partial x}\right)_i = \sum_{j\neq i}\left[\frac{1}{0.5(\rho_i+\rho_j)}C_{1j}\left(p_j - p_i\right)\right] \tag{18}$$

and

$$\left(\frac{\partial}{\partial x}\left(\frac{1}{\rho}\frac{\partial p}{\partial x}\right)\right)_i = \sum_{j\neq i}\left[\frac{1}{0.5(\rho_i+\rho_j)}C_{3j}\left(p_j - p_i\right)\right] \tag{19}$$

The coefficients C_{1j} and C_{3j} are the same as those in Equations (16) and (17). Letting $\rho_i = \rho_j$, Equations (18) and (19) recover to Equations (16) and (17) for single fluid domain far away from fluid interface. Hence, the above reformulation is general which can simulate both single- and multi-phase flows. The accuracy of the enhanced derivative computation scheme has been demonstrated by Luo *et al.* [22]. Particularly, it enables the accurate simulation of sharp fluid interface (e.g. water and air whose density difference is about three orders of magnitude) because no density smoothing or smearing scheme is used.

3.3. Compressible solver based on thermodynamics

For compressible flows, $\rho^{(k+1)}$ in Equation (5) is unknown and hence a closure condition is needed to solve the PPE. The polytropic gas law is selected to be the closure relation as follows

$$\frac{p}{\rho^{\gamma}} = \text{constant} \tag{20}$$

where γ is the ratio of specific heats at constant pressure and constant volume. The theoretical value of γ for air is about 1.4. It has been conceptually explained (more than 98% of air being diatomic molecules) and numerically demonstrated that $\gamma = 1.4$ can model the entrapped air pocket in water wave impact problems with good accuracy [23]. Incorporating the closure condition of Equation (20) to Equation (5), the PPE accounting for fluid compressibility can be obtained as [23]

$$-\nabla\cdot\left(\frac{1}{\rho_i^*}\nabla p_i^{(k+1)}\right) + \frac{1}{\Delta t^2 \rho_i^*}\frac{\rho_{a0}}{p_{a0}}\frac{1}{\gamma}p_i^{(k+1)} = -\frac{1}{\Delta t^2}\frac{\rho_{a0}-\rho_i^*}{\rho_i^*} + \frac{1}{\Delta t^2\rho_i^*}\frac{\rho_{a0}}{\gamma} \tag{21}$$

Equation (21) does not require the input of speed of sound (c_s), which is dependent on the composition and temperature of a fluid. The implication of not using sound speed directly in the formulation is subtle but important. To simulate compressible flow, a commonly used equation of state involves sound speed which is given by $c_s = \sqrt{\partial p / \partial \rho}$ where p and ρ are the fluid pressure and density respectively. When modelling violent water-air flows, this approach has some limitations because c_s is dependent on the temperature of fluid (Wemmenhove 2008). Therefore, it is difficult to determine (or update) the correct value of sound speed in different scenarios (e.g. during and immediately after wave impact in a wave impact problem). In contrast, the polytropic formulation adopted in our method does not require the sound speed explicitly as an input but yet is able to account for the change in sound speed due to, for example, change of temperature. Furthermore, this compressible solver uses the predictor-corrector scheme to solve the governing equations for incompressible liquid and compressible gas. The difference is that the compressible solver for gas treats the fluid density in PPE in an implicit way. This enables a simultaneous simulation of two-phase flows in the same solution framework for incompressible and compressible fluids (e.g. wave impact problems with entrapped air pocket). In the following section, three wave impact examples will be studied using the developed numerical method.

3.4. Wall boundary conditions

The wall boundaries in this study are modelled by the fixed particle approach, i.e. one layer of particles placed on the physical wall position and two more layers of dummy particles outside the wall to facilitate the influence domain ($r_e = 2.1L_0$) of wall particles. The positions of wall and dummy particles corresponding to the moving walls are updated in accordance with the wall motion. The pressures on dummy particles are related to the pressure of the nearest wall particles by the hydrostatic relationship, satisfying the Neumann boundary condition $\partial p / \partial \mathbf{n} = -\rho \mathbf{g} \cdot \mathbf{n}$, where **n** is the outward unit vector on the solid boundary.

4. Enhancement of computational efficiency

4.1. Parallel direct sparse solver

In previous CPM studies, the sequential generalized minimal residual (GMRES) method was used to solve the pressure equation, costing 80% or more of the total computational time. Hence this is the main part for the enhancement of computational efficiency. The coefficient matrix of the PPE generated by CPM is sparse and non-symmetric. There are several suitable function solvers in the literature including the multi-frontal massively parallel solver (MUMPS) [32] and the parallel direct sparse solver (Pardiso) [33]. Among these two solvers, the Pardiso has been shown by other researchers to be faster and require less memory [34, 35], and is hence selected in this study (solid box in Figure 2). This solver utilizes the OpenMP directives for parallelization based shared memory multiprocessing systems. With the LU factorization (where "LU" stands for "lower and upper triangular matrices") as the mathematical basis, Pardiso contains three key steps, i.e. reordering, factorization and back substitution. The factorization is the most time-consuming step and is speeded up by an efficient parallel sparse LU factorization scheme [36]. Note that in solving the pressure equations by Pardiso, which takes up most of the computational time, the only variable (at every particle) is pressure and the system size does not change. The CPM code is written in Fortran and all variables are stored as array elements. The code is compiled and run using the 64-bit model of Intel Parallel Studio XE 2013 and a workstation with one Intel Xeon CPU E5-2680 v4 @ 2.40 GHz (14 cores, 28 threads).

The performance of Pardiso with one thread (without parallelism) is tested first. The execution time of GMRES and Pardiso solvers in seven task sizes (T_1 to T_7) are compared in Table 1, in which N_p is the total number of fluid and wall particles, t_{SG} and t_{SP1} are the times for solving the sparse equation system (PPE) by GMRES and Pardiso with one thread, respectively. The task size T_j (j =1, 2, …, 7) is defined as the ratio of N_p for T_j to N_p for T_1. As the equation system size increases from T_1 to T_7 (by 64 times), the PPE solver time by GMRES ($\propto N_p^{1.78}$) increases much more rapidly than that of Pardiso with one thread ($\propto N_p^{0.95}$). Specifically, t_{SG} is slightly larger than t_{SP1} in T_1, but t_{SG} / t_{SP1} increases to more than 33 in T_7. It means that the Pardiso (even using only one thread) is much more efficient than the sequential GMRES in solving large equation systems.

The Pardiso solver can be accelerated by using multiple threads as shown in Table 2, in which $t_{\mathrm{SP}i}$ is the function solver time by Pardiso with i threads and $t_{\mathrm{OS}i}$ the time for all other subroutines of CPM with i threads. Speedup is the ratio of the serial runtime to the time taken by the parallel code for the same work. Figure 3a shows that the speedup can reach more than 4 times (for the studied tasks) with larger systems having higher speedup. The speedup is effective when the number of used threads is less than 1/3 of the number of available threads, and the increase in speedup is less effective with further increase in the number of used threads. Particularly when the number of used threads is more than 2/3 of total threads, the computational efficiency decreases due to a relatively high overhead for data communication from the shared memory to the threads used. Figure 3b shows the strong scaling efficiency (ratio of speedup to the number of threads) of Pardiso. For big task size (i.e. larger equation system), the parallel efficiency is relatively high because the overhead for data communication is a small portion compared to the effort in solving the equation system.

Let the time to complete a work unit (N_p particles in this study) using 1 thread be t_1 and the time to complete n work units ($n \times N_p$ particles) using n threads be t_n. The weak scaling in terms of normalized computational time (t_n/t_1) for Pardiso is shown in Figure 4. Ideally, the weak scaling is a horizontal straight line, meaning that the time to complete one work unit by one thread is the same as that to complete n work units by n threads. Due to communication overhead mainly, however, the normalized computational time increases as the task size increases. The slope of the weak-scaling curve for Pardiso is larger, implying a higher communication overhead than the other subroutines of CPM (which will be elaborated in Section 4.2). This is further demonstrated in Table 2 that the speedup of other subroutines of CPM (4.96 times) is slightly better than that of Pardiso (4.26 times).

4.2. Parallelization of all other CPM subroutines

While the PPE solver time is significantly reduced by using the direct sparse solver, the computational time for all other subroutines of the CPM code is comparable to or even more than (see t_{OS1} in Table 2) the

PPE solver time if not parallelized. To achieve the overall computational efficiency, it is necessary to parallel all other subroutines of CPM (shown in dashed boxes in Figure 2). The same instructions are performed in the subroutines for computing intermediate fluid velocity and position, searching for neighbor particles (by the link-list approach [37]) and computing intermediate density, computing the gradient and Laplacian operators, and updating fluid velocities and positions. The OpenMP directive "!$omp parallel do" is used to concurrently execute the repeating instructions (for all particles) on multithreads, thereby achieving the thread parallelism. Since the computational time for each loop is theoretically the same (in a single subroutine), static schedule is adopted without specifying the optional parameter *chunk* (distributing loops equally to available threads), which gives the least overhead incurred by the handling and distributing of different loops and hence achieves the best performance.

The speedup of other CPM subroutines (i.e. other than the function solver) is 4 to 5 times for the studied tasks with larger systems being accelerated slightly more as shown in Figure 5a. Similar to the function solver, the increase in speedup of other CPM subroutines is not significant after the number of used threads exceeds 2/3 of total threads due to communication overhead. As presented in Figure 5b, the strong scaling efficiency of all other CPM subroutines (~25%) is slightly better than that of Pardiso (~20%), although the overall trends are very similar. The weak scaling efficiency in terms of the normalized computational time (t_n/t_1) for other CPM subroutines is shown as the solid curve in Figure 4. The gentler slope compared to that of the dashed curve for function solver means that the communication overhead is lower for these subroutines and hence a higher efficiency enhancement. This is consistent with the strong scaling analysis and the data presented in Table 2.

The parallel direct sparse solver and OpenMP parallelization of other CPM subroutines lead to a shared-memory parallel version of CPM. Figure 6 shows the overall efficiency speedup of the parallelized CPM code compared to the original sequential CPM code with GMRES. Particularly, the overall efficiency enhancement of T_7 in this study is about 104 times. Of significant importance is that the efficiency enhancement varies almost linearly with the size of the computational system. The linear time $O(N_p)$ is certainly an advantageous feature for the parallelized CPM to simulate large-scale 3D problems.

5. Numerical examples

5.1. Wave impact on sea wall

Prediction of wave forces on sea wall is of great practical significance but very difficult because wave breaking strongly affects the wave characteristics near the structure [38]. CPM is used to study this problem by comparing with the experimental and SPH results in Didier *et al.* [38]. The dimensions of the problem are shown in Figure 7. The wave maker motion is governed by $x(t) = -A_b \sin \omega t$, where A_b is 0.05048 m and ω is $2\pi / 1.3$. In CPM simulation, an initial particle spacing of 0.005 m (56,224 particles in total) is adopted with a fixed time step of 0.001 s (satisfying Courant's condition).

Figure 8 shows the wave motion at typical time instants. The wave maker generates a wave propagating towards right and impacting on the sea wall. A large wave impact occurs at around t = 5.86 s (see Figure 8). The wave runs up along the wall (t = 6.06 s), turns over and falls back under gravity. The falling-back wave flows towards left (t = 6.26 s) and interacts with the next coming wave, generating a plunging wave (t = 6.8 s). It is then followed by the next wave impact. Specifically, the wave elevations at wave gauges G_1 and G_2 (2.643 m and 3.943 m from the initial position of the wave maker respectively) are presented in Figure 9. SPH slightly under predicts the wave elevation, whereas CPM reproduces the wave elevation peaks in better agreement with the experimental results. This is because CPM can accurately compute spatial derivatives and hence has less numerical dissipation. Figure 10 presents the wave impact pressures on the sea wall. As can been seen, the pressures at P_1 and P_2 predicted by SPH have large peaks, which, however, are not shown in the experimental results. And SPH slightly under predicts the pressure at P_3. In contrast, CPM accurately reproduces the real pressure peaks (essential in engineering design) and are in good agreement with the experimental results by Didier *et al.* [38].

5.2. Water sloshing under rotational excitation

For further validation, CPM is then used to model violent sloshing in a closed tank subjected to roll and pitch excitations on a rotational motion simulator as shown in Figure 11. The same rotational simulator as that in Luo *et al.* [29] is used. The way to measure the rotational angles of the rotational platform is schematically shown in Figure 12. The equation of the plane (plate) can be written as

$$ax + by + cz = 1 \tag{22}$$

The two displacement transducers (DT_1 and DT_2) are vertically installed at point 1 (x_1, y_1) and point 2 (x_2, y_2). The values of (x_1, y_1) and (x_2, y_2) are fixed and can be measured in the beginning of the experiment. The vertical positions of these two points, i.e. z_1 and z_2, are measured by the displacement transducers. Substituting (x_1, y_1, z_1) and (x_2, y_2, z_2) into the plane equation, two equations can be obtained. The third equation is that the distance from the pivot point (point O) of the rotational simulator to its projection point on the platform (point 3) is a constant value. With the three equations, the plane equation of the platform and hence the rotation angles along roll and pitch directions can be uniquely determined at every time step. The measured platform rotations are used as excitation input of numerical simulations.

The experimental set-up including the prismatic tank is the same as that used in Luo *et al.* [29] but with a different focus of study (3D swirling wave) in this paper. Based on our parametric studies, the filling depth of d/H = 0.3 generates large sloshing pressure and is adopted as the case study in this section. Estimated by linear wave theory, the fundamental natural frequencies for the sloshing system in the roll and pitch directions are 5.598 rad/s and 7.471 rad/s respectively. Dynamic pressures at P_1 and P_2 are measured by gauge pressure sensors (WIKA S10) and the sloshing wave motions are captured by a video camera.

Unlike in the study by Luo *et al.* [29], the present work focuses on the phenomenon of 3D swirling wave. According to Faltinsen *et al.* [39] and our preliminary studies, a relatively regular swirling wave can be generated by setting the frequencies of roll and pitch excitations to be the same value that is between the lowest natural frequencies of sloshing in the roll and pitch directions. Therefore, the excitation frequencies of roll and pitch rotations are both selected to be 5.969 rad/s. The measured pitch and roll angles are as shown in Figure 14. In the numerical simulation, an initial particle spacing of 0.008 m (99,789 particles in total) and a fixed time step of 0.0005 s are adopted.

The wave profiles at different time instants are presented in Figure 15. CPM accurately captures the highly deformed sloshing waves involving fluid fragmentation and coalescence. From t = 0.7s to 2.275s, a swirling wave is generated in the tank, the reason of which is the superposition of the wave components in the length and breadth directions of the tank. The swirling wave hits one corner of the tank at t = 1.625s and breaks at t = 2.275s. As the sloshing wave further develops, the breaking wave impinges on the side wall and the ceiling of the tank. The splashing water droplets fall back and impact on the main water body. Although breaking wave occurs, the sloshing wave behaves like a swirling wave in general and approaches the steady state after 3 seconds.

The violent waves generate large impact forces on the tank wall, as presented in Figure 16, which shows the sloshing pressures at P_1 and P_2 (indicated in Figure 13). CPM solutions are in fairly good agreement with experimental results both in amplitude and time phase. Of particular importance, the pressure peaks are successfully captured by CPM. The relative difference between the CPM and experimental results of the largest pressure peak is less than 2%. It also can be seen that the sloshing pressure near the tank corner (P_2) is generally larger than that in the middle (P_1). The practical implication of this phenomenon is that the structure components near the tank corner are more likely to be damaged and hence should be designed with a higher requirement.Sloshing impact with entrapped air pocket

In the previous two examples, although fluid motion is very violent, there is no entrapped air pocket for which accurate modelling of air compressibility is important. To study wave impact scenario with entrapped air pocket, a sloshing experiment is conducted in the same container with two connected tanks as that in Luo *et al.* [23]. Different than using the translational shake table in Luo *et al.* [23], the present experiment is conducted on the rotational simulator as presented in the previous section (Figure 17). In fact, sloshing due to rotational excitation is more severe than sloshing due to translational excitation. The dimension of the container is shown in Figure 18. Air pressure at the middle of the top wall of the right tank, i.e. P_{A1} in Figure 18, is measured by an absolute pressure sensor. Water pressures at 60 mm from the bottom on the right wall of the right tank (P_{W1}) and 30 mm from the bottom on the left wall of the left tank (P_{W3}) are measured by gauge pressure sensors.

Based on some preliminary studies, the filling depth is adopted to be 0.18 m (initial d_L and d_R in Figure 18) to have significant effect of the entrapped air pocket. The excitation frequency of $0.92\omega_0$ (= 3.6493 rad/s) is found to generate a relatively large variation of air pressure in the right tank, where ω_0 is the reference frequency which is the natural frequency of water in the left tank only (ignoring the right tank) with water depth (d_L) and length (L_L) based on linear wave theory. In numerical simulation, an initial particle distance of 0.005 m (11,124 particles in total) and fixed time step 0.0005 s are adopted to achieve a good tradeoff between accuracy and efficiency.

The numerical and experimental wave profiles at five time instants are compared in Figure 19, which shows generally good agreement. Since filling level is low, wave moves like a bore which develops over time (see snapshots at t = 2.00 s and 2.88 s in Figure 19). At t = 3.12 s, the sloshing wave severely impacts on the right wall of the left tank, applying a large force to the connecting channel. The force pushes the water in the connecting channel towards the right and compresses the enclosed air pocket in the right tank. As expected, a large peak of air pressure appears in the right tank, which is measured in the experiment and predicted well by CPM (Figure 20a). As water in the left tank runs up along the right wall (t = 3.20 s in Figure 19), the compression force continues to apply to the enclosed air pocket. This is why the large pressure in the air pocket lasts for some time as shown in Figure 20a. At t = 3.68 s, the run-up water falls back to the main water body in the left tank and begins to move towards the left. At this stage, the pressure in the air pocket is smaller than the atmospheric pressure. The air pressure oscillation also influences the water pressure near the air pocket (see water pressure at P_{w1} in Figure 20b).

It is noted that both the experimental and CPM results of air pressure in the air pocket show vibration and they are in good agreement. The pressure vibration in the air pocket is further investigated through a power spectral analysis using the Fast Fourier Transform (FFT). It is interesting to note that there is only one peak value, i.e. 6.120 Hz, in the frequency-power curve. This means that the air pressure vibrates with one distinctive frequency. To verify that this pressure vibration is real and not spurious due to the numerical algorithm, the natural frequency of the air tube (under compression of water) is derived. Following Ramkema [40] who addressed the problem of wave impact on coastal structures, the air-pocket-water system is represented by a mass-spring system as shown in Figure 1, in which the spring represents the air pocket compressibility and the mass is the water effectively contributing to the impinging force on the air pocket. The upper bound of the effective water mass is the water in the connecting channel and the right tank, while the lower bound is the upper bound excluding the water in the rectangular region at the right bottom corner of the container (the region within the dash-dot line in Figure 1). Since water at the right bottom corner (dark shaded region in Figure 1) is almost stationary relative to the tank (the right bottom corner of the container is theoretically a stagnation point), the effective mass of the present problem (light shaded region in Figure 1) can be approximated to be water in the connecting channel and the right tank excluding the right bottom corner.

Assuming the water level in the right tank to be horizontal and giving it a small perturbation z, the force (per unit width) applied on the effective water mass is as follows

$$
\begin{aligned}
F &= (p_{a0} - p)L_{\mathrm{R}} = \left[1 - \left(1 + \frac{z}{H_{\mathrm{a}0} - z}\right)^{\gamma}\right] p_{a0} L_{\mathrm{R}} \\
&\approx \left[1 - \left(1 + \gamma \frac{z}{H_{\mathrm{a}0} - z}\right)\right] p_{a0} L_{\mathrm{R}} = -\frac{\gamma p_{a0} L_{\mathrm{R}}}{H_{\mathrm{a}0} - z} z \approx -\frac{\gamma p_{a0} L_{\mathrm{R}}}{H_{\mathrm{a}0}} z
\end{aligned}
\tag{23}
$$

where p_{a0} is the initial air pressure in the right tank, L_{R} the length of the right tank and H_{a0} the initial height of the air tube. Ignoring the friction forces from the tank walls, the dynamic equation for the effective water mass is as follows

$$
M_{\mathrm{w}} \frac{d^2 z}{dt^2} + \frac{\gamma p_{a0} L_{\mathrm{R}}}{H_{\mathrm{a}0}} z = 0 \tag{24}
$$

where M_{w} is the effective water mass (per unit width). Then the natural frequency of the dynamic system can be obtained as

$$
f = \frac{1}{2\pi} \sqrt{\frac{\gamma p_{a0} L_{\mathrm{R}}}{H_{\mathrm{a}0} M_{\mathrm{w}}}} \tag{25}
$$

Equation (25) is similar to that derived by Cuomo *et al.* [41] who analytically studied wave impingement entrapping an air pocket against vertical wall. Substituting the upper and lower bounds of M_{w} into Equation (25), the lower and upper bounds of the natural frequency of the entrapped air pocket can be obtained to be 5.668 Hz and 6.507 Hz, whereas the natural frequency corresponding to the adopted value of M_{w} is 6.296 Hz. Compared to the observed frequency of pressure vibration (i.e. 6.120 Hz) in the experimental result, the relative differences are only 7.3 %, 6.3 % and 2.8 %, respectively, for the lower and upper bounds and the adopted value of M_{w}. Therefore, the study on the natural frequency of the air pocket substantiates that the pressure oscillations observed in the experiment and CPM simulation are real and due to the natural vibration of the entrapped air pocket (air cushion effect).

6. Conclusions

In this paper, the novel CPM is presented with several features: (1) Accurate computation of first- and second-order derivatives in a way consistent with Taylor series expansion even in two-phase cases with abrupt density change to about 1000; (2) A thermodynamically-consistent compressible solver by employing the polytropic gas law; (3) Seamless integration of the incompressible and compressible solvers such that wave impact problems with entrapped air pocket can be simulated in a simultaneous way; and (4) shared-memory parallelization of the code to achieve significant speedup of computational efficiency.

The parallelization includes two parts: a parallel direct sparse solver – Pardiso (based on OpenMP) to solve the PPE and the parallelization of CPM subroutines by OpenMP directives. Through serial computations (using only one thread) of seven different tasks, Pardiso is shown to be 33 times (or more for larger systems) faster than the sequential GMRES which was previously used in CPM. Strong and weak scaling analyses are conducted. The speedups of Pardiso function solver and other CPM subroutines reach 4.5 and 5 times respectively (with larger systems being accelerated slightly more) on a workstation equipped with a 14-core CPU. The parallel efficiencies of Pardiso and other CPM subroutines are about 20 % and 25 % respectively when the maximum speedup is achieved. The weak scaling of computational time shows that the communication overhead in the function solver is slightly more than that in the other CPM subroutines which involve near-neighbor communication. Compared to the original serial code with GMRES, the parallel CPM achieves 100 times or more efficiency speedup. The efficiency enhancement varies almost linearly with the size of the computational system.

Three numerical examples are studied to demonstrate the accuracy of CPM. The first case is concerned with wave impact on seawall, which is an important problem in coastal engineering. The highly nonlinear plunging wave is successfully reproduced and the predicted wave elevation and impact pressure are in good agreement with published experimental results. The second example is the violent sloshing under rotational excitations. The complicated 3D sloshing waves (e.g. swirling wave) are accurately captured and the violent impact pressures are in good agreement with our experimental results.

Finally, the sloshing impact with entrapped air pocket is experimentally studied in a container with two connected tanks under rotational excitation. CPM solutions including wave profiles, wave impact pressures and particularly the pressure vibration in the air pocket agree well with the experimental results. The natural frequency derived for the entrapped air pocket is close to the vibration frequencies of both the experimental and numerical air pocket pressure. The significant pressure oscillation in an entrapped air pocket during a violent wave impact process is thus caused by natural vibration of the compressible air pocket and is accurately modelled by CPM.

7. Acknowledgement

The authors appreciate the research grant provided by the Singapore Maritime Institute (Project SMI-2014-OF-02) as well as the funding and technical support of Sembcorp Marine Technology Pte Ltd.

References

[1] Liu D, Lin P. A numerical study of three-dimensional liquid sloshing in tanks. Journal of Computational Physics. 2008;227:3921-39.

[2] Wang Y, Shu C, Huang HB, Teo CJ. Multiphase lattice Boltzmann flux solver for incompressible multiphase flows with large density ratio. Journal of Computational Physics. 2015;280:404-23.

[3] Rogers BD, Dalrymple RA, Stansby PK. Simulation of caisson breakwater movement using 2-D SPH. Journal of Hydraulic Research. 2010;48:135-41.

[4] Khayyer A, Gotoh H. Modified Moving Particle Semi-implicit methods for the prediction of 2D wave impact pressure. Coastal Engineering. 2009;56:419-40.

[5] Skillen A, Lind S, Stansby PK, Rogers BD. Incompressible smoothed particle hydrodynamics (SPH) with reduced temporal noise and generalised Fickian smoothing applied to body–water slam and efficient wave–body interaction. Computer Methods in Applied Mechanics and Engineering. 2013;265:163-73.

[6] Faltinsen OM, Landrini M, Greco M. Slamming in marine applications. Journal of Engineering Mathematics. 2004;48:187-217.

[7] Mokos A, Rogers BD, Stansby PK. A multi-phase particle shifting algorithm for SPH simulations of violent hydrodynamics with a large number of particles. Journal of Hydraulic Research. 2016;55:143-62.

[8] Rogers BD, Stansby P, Lind SJ. Slam Modelling with SPH: The Importance of Air. the 8th International SPHERIC Workshop. Trondheim2013. p. 282-8.

[9] Peregrine DH, Thais L. The effect of entrained air in violent water wave impacts. Journal of Fluid Mechanics. 2006;325:377- 97.

[10] Lind SJ, Stansby PK, Rogers BD, Lloyd PM. Numerical predictions of water–air wave slam using incompressible–compressible smoothed particle hydrodynamics. Applied Ocean Research. 2015;49:57-71.

[11] Xue MA, Lin P. Numerical study of ring baffle effects on reducing violent liquid sloshing. Computers & Fluids. 2011;52:116-29.

[12] Archer PJ, Bai W. A new non-overlapping concept to improve the Hybrid Particle Level Set method in multi-phase fluid flows. Journal of Computational Physics. 2015;282:317-33.

[13] Wang Z, Zou Q, Reeve D. Simulation of spilling breaking waves using a two phase flow CFD model. Computers & Fluids. 2009;38:1995-2005.

[14] Monaghan JJ. Smoothed Particle Hydrodynamics and Its Diverse Applications. Annual Review of Fluid Mechanics. 2012;44:323-46.

[15] Liu GR, Liu MB. Smoothed Particle Hydrodynamics: A meshfree particle method. Singapore: World Scientific; 2003.

[16] Lee ES, Moulinec C, Xu R, Violeau D, Laurence D, Stansby P. Comparisons of weakly compressible and truly incompressible algorithms for the SPH mesh free particle method. Journal of Computational Physics. 2008;227:8417-36.

[17] Shao S, Lo EYM. Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface. Advances in Water Resources. 2003;26:787-800.

[18] Liu X, Lin P, Shao S. An ISPH simulation of coupled structure interaction with free surface flows. Journal of Fluids and Structures. 2014;48:46-61.

[19] Lind SJ, Xu R, Stansby PK, Rogers BD. Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves. Journal of Computational Physics. 2012;231:1499-523.

[20] Koshizuka S, Nobe A, Oka Y. Numerical analysis of breaking waves using the moving particle semi-implicit method. International Journal for Numerical Methods in Fluids. 1998;26:751-69.

[21] Koh CG, Gao M, Luo C. A new particle method for simulation of incompressible free surface flow problems. International Journal for Numerical Methods in Engineering. 2012;89:1582–604.

[22] Luo M, Koh CG, Gao M, Bai W. A particle method for two-phase flows with large density difference. International Journal for Numerical Methods in Engineering. 2015;103:235-55.

[23] Luo M, Koh CG, Bai W, Gao M. A particle method for two-phase flows with compressible air pocket. International Journal for Numerical Methods in Engineering. 2016;108:695-721.

[24] Ferrari A, Dumbser M, Toro EF, Armanini A. A new 3D parallel SPH scheme for free surface flows. Computers & Fluids. 2009;38:1203-17.

[25] Marrone S, Bouscasse B, Colagrossi A, Antuono M. Study of ship wave breaking patterns using 3D parallel SPH simulations. Computers & Fluids. 2012;69:54-66.

[26] Hu XY, Adams NA. A multi-phase SPH method for macroscopic and mesoscopic flows. Journal of Computational Physics. 2006;213:844-61.

[27] Koh CG, Luo M, Gao M, Bai W. Modelling of liquid sloshing with constrained floating baffle. Computers & Structures. 2013;122:270-9.

[28] Liu MB, Liu GR. Smoothed Particle Hydrodynamics (SPH): An overview and recent developments. Archives of Computational Methods in Engineering. 2010;17:25-76.

[29] Luo M, Koh CG, Bai W. A three-dimensional particle method for violent sloshing under regular and irregular excitations. Ocean Engineering. 2016;120:52-63.

[30] Kadioglu SY, Sussman M. Adaptive solution techniques for simulating underwater explosions and implosions. Journal of Computational Physics. 2008;227:2083-104.

[31] Kwatra N, Su J, Grétarsson JT, Fedkiw R. A method for avoiding the acoustic time step restriction in compressible flow. Journal of Computational Physics. 2009;228:4146-61.

[32] Amestoy PR, Duff IS, L'Excellent J-Y. MUMPS multifrontal massively parallel solver version 2.0. 1998.

[33] Schenk O, Gärtner K. Solving unsymmetric sparse systems of linear equations with PARDISO. Future Generation Computer Systems. 2004;20:475-87.

[34] Solovev SA, Pudov SG. PARallel DIrect SOlver to solve SLAE on shared memory architectures. 2008.

[35] Raju MP, Khaitan SK. High Performance Computing of Three-Dimensional Finite Element Codes on a 64-bit Machine. Journal of Applied Fluid Mechanics. 2012;5:123-32.

[36] Schenk O, Gärtner K. Two-level scheduling in PARDISO-Improved scalability on shared memory multiprocessing systems. Parallel Computing. 2002;28:187-97.

[37] Gesteira MG, Rogers BD, Dalrymple RA, Crespo AJC, Narayanaswamy M. User guide for the SPHysics code. 2010.

[38] Didier E, Neves DRCB, Martins R, Neves MG. Wave interaction with a vertical wall: SPH numerical and experimental modeling. Ocean Engineering. 2014;88:330-41.

[39] Faltinsen OM, Rognebakke OF, Timokha AN. Classification of three-dimensional nonlinear sloshing in a square-base tank with finite depth. Journal of Fluids and Structures. 2005;20:81-103.

[40] Ramkema C. A model law for wave impacts on coastal structures. Coastal Engineering Proceedings1978.

[41] Cuomo G, Allsop W, Takahashi S. Scaling wave impact pressures on vertical walls. Coastal Engineering. 2010;57:604-9.

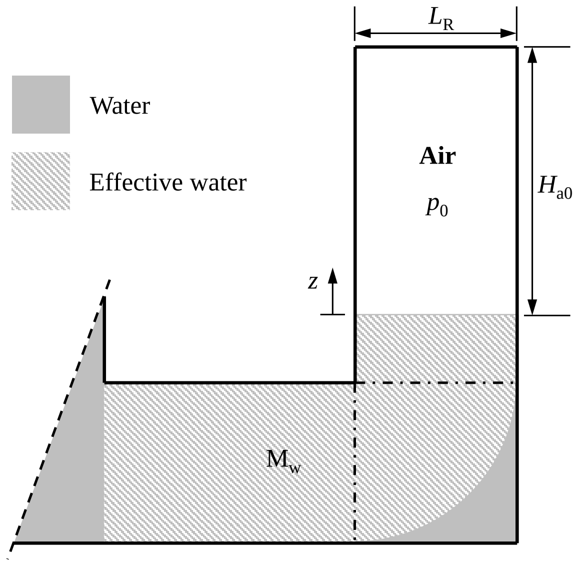


Figure 1. Schematic view of water impact on an air pocket (not to scale)

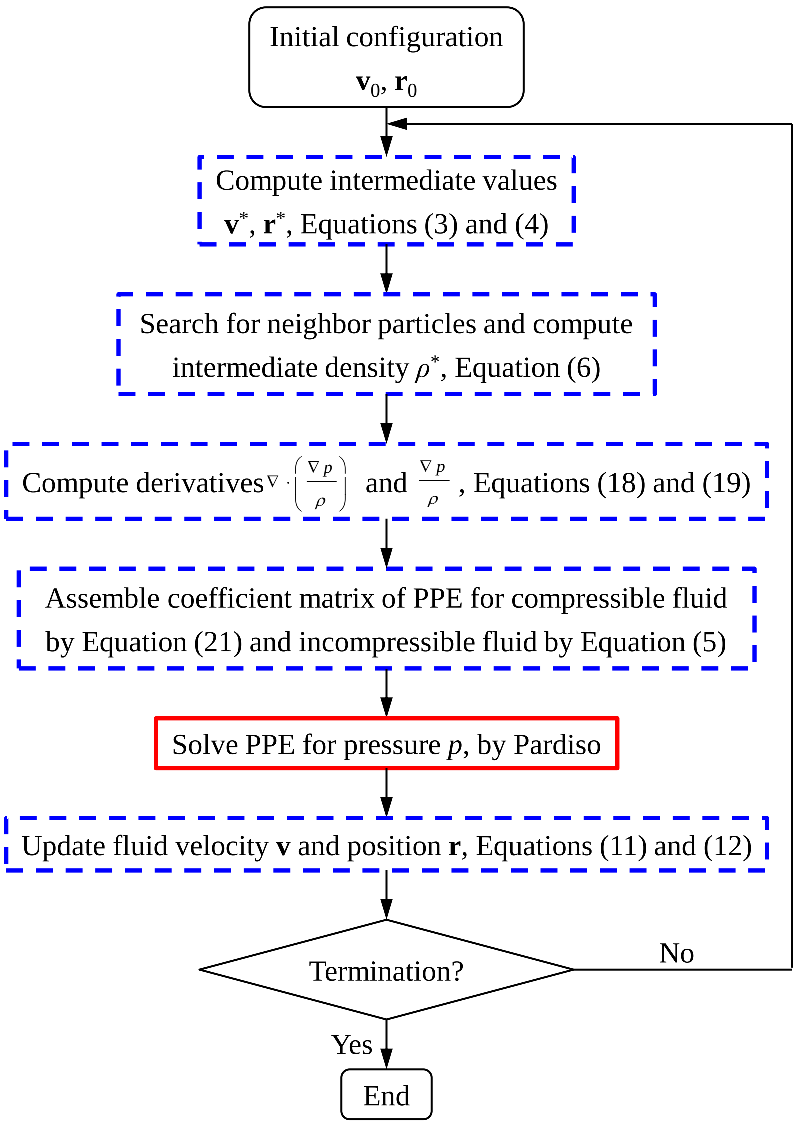

Initial configuration
$\mathbf{v}_0$, $\mathbf{r}_0$
Compute intermediate values
$\mathbf{v}^*$, $\mathbf{r}^*$, Equations (3) and (4)
Search for neighbor particles and compute
intermediate density ρ^*, Equation (6)
Compute derivatives $\nabla\cdot\left(\frac{\nabla p}{\rho}\right)$ and $\frac{\nabla p}{\rho}$, Equations (18) and (19)
Assemble coefficient matrix of PPE for compressible fluid
by Equation (21) and incompressible fluid by Equation (5)
Solve PPE for pressure p, by Pardiso
Update fluid velocity $\mathbf{v}$ and position $\mathbf{r}$, Equations (11) and (12)
Termination?
No
Yes
End

Figure 2. Flow chart of parallel Consistent Particle Method: solid box is function solver (Pardiso); dashed boxes are all other CPM subroutines parallelized by OpenMP directives

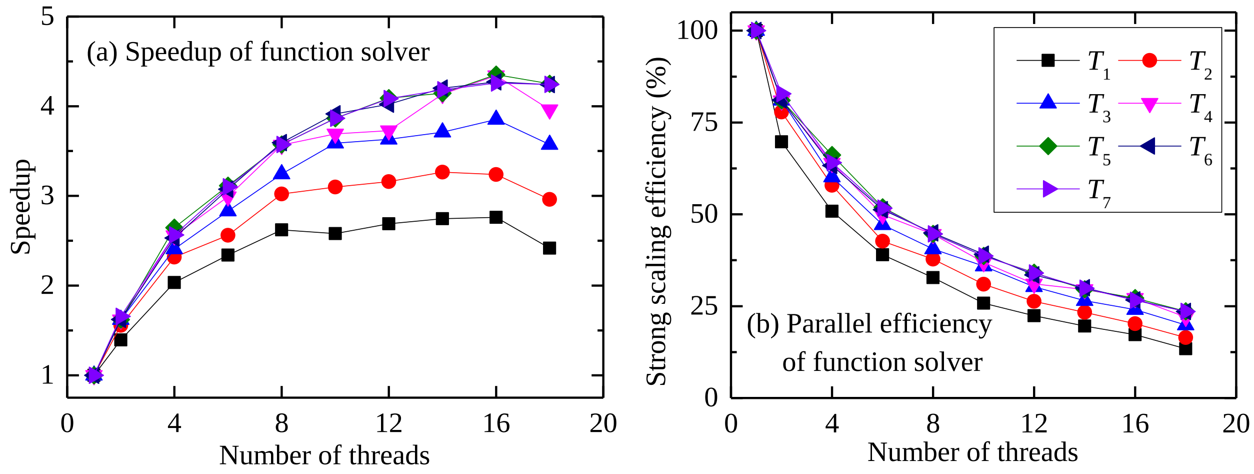


Figure 3. Speedup and strong scaling efficiency of parallel direct sparse solver – Pardiso

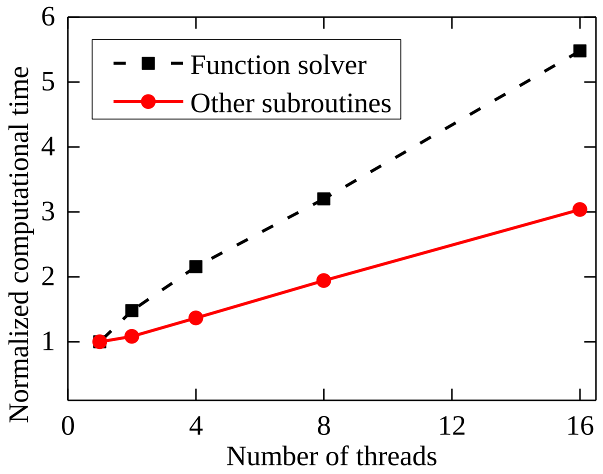


Figure 4. Normalized computational time for parallel direct sparse solver (black square) and all other subroutines of CPM (red circle)

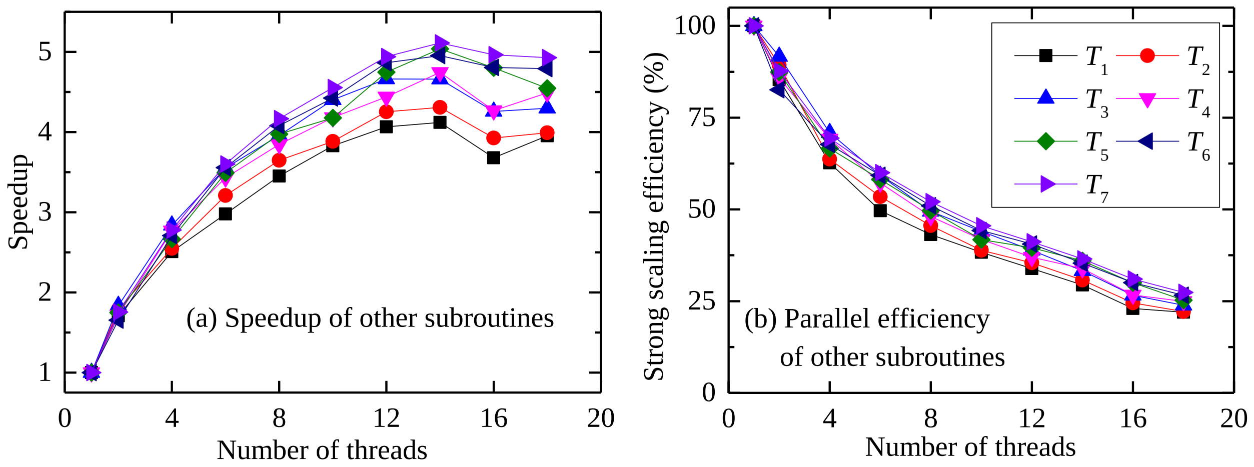


Figure 5. Speedup and strong scaling efficiency of all other CPM subroutines (i.e. other than function solver) parallelized by OpenMP

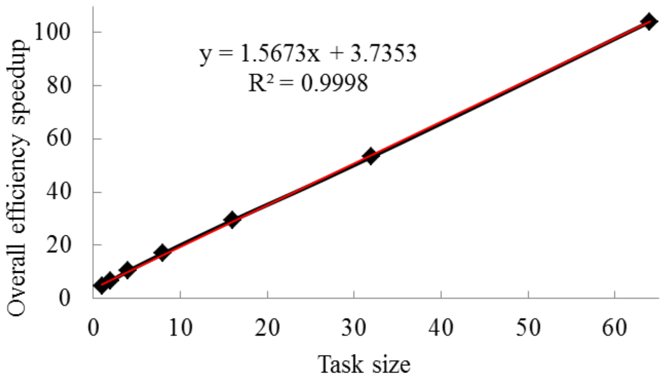


Figure 6. Overall speedup of computational efficiency with task size

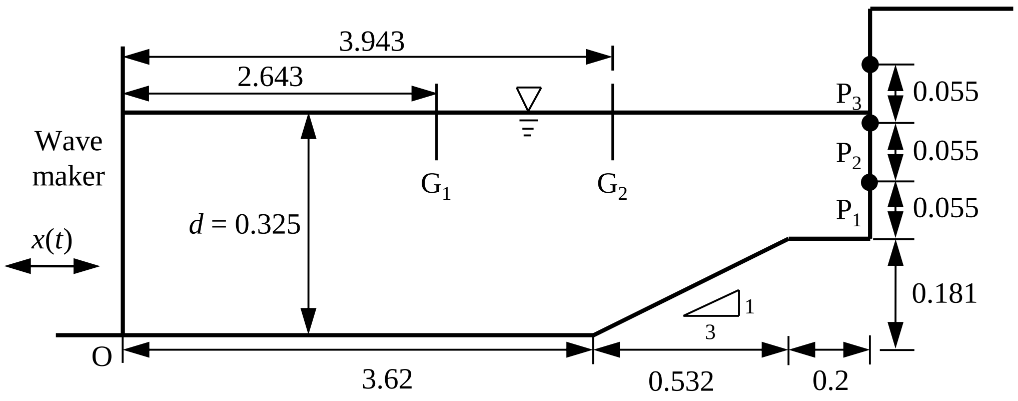


Figure 7. Schematic view of wave impact on sea wall (Unit: m)

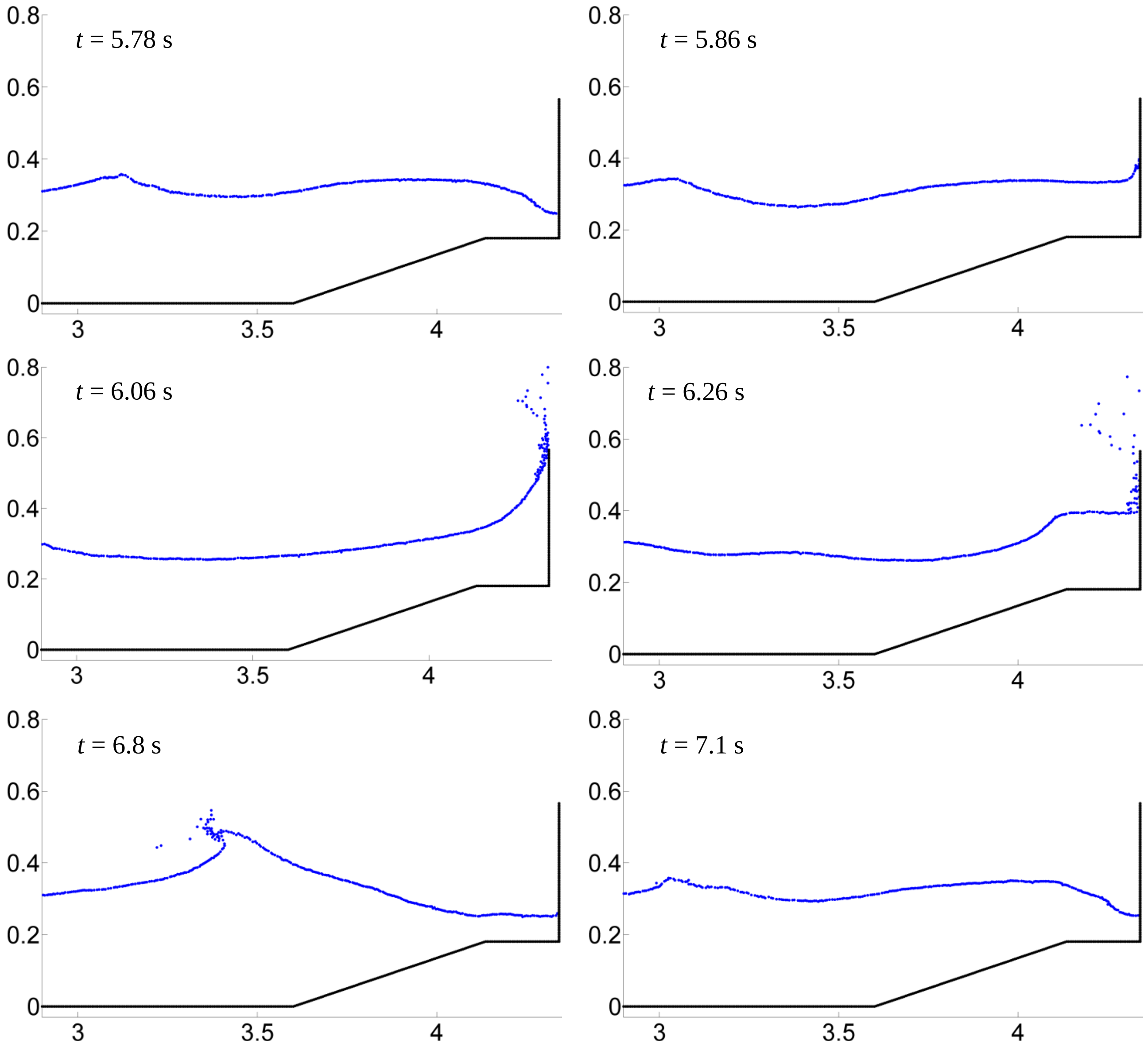


Figure 8. Snapshots of wave motion at typical time instants

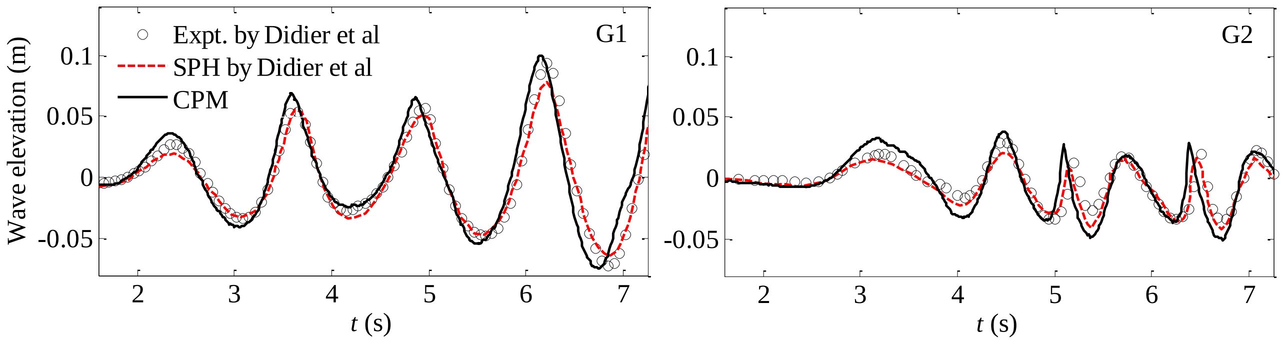


Figure 9. Wave elevation at G1 and G2: experimental and SPH results by Didier *et al.* [38] and CPM result

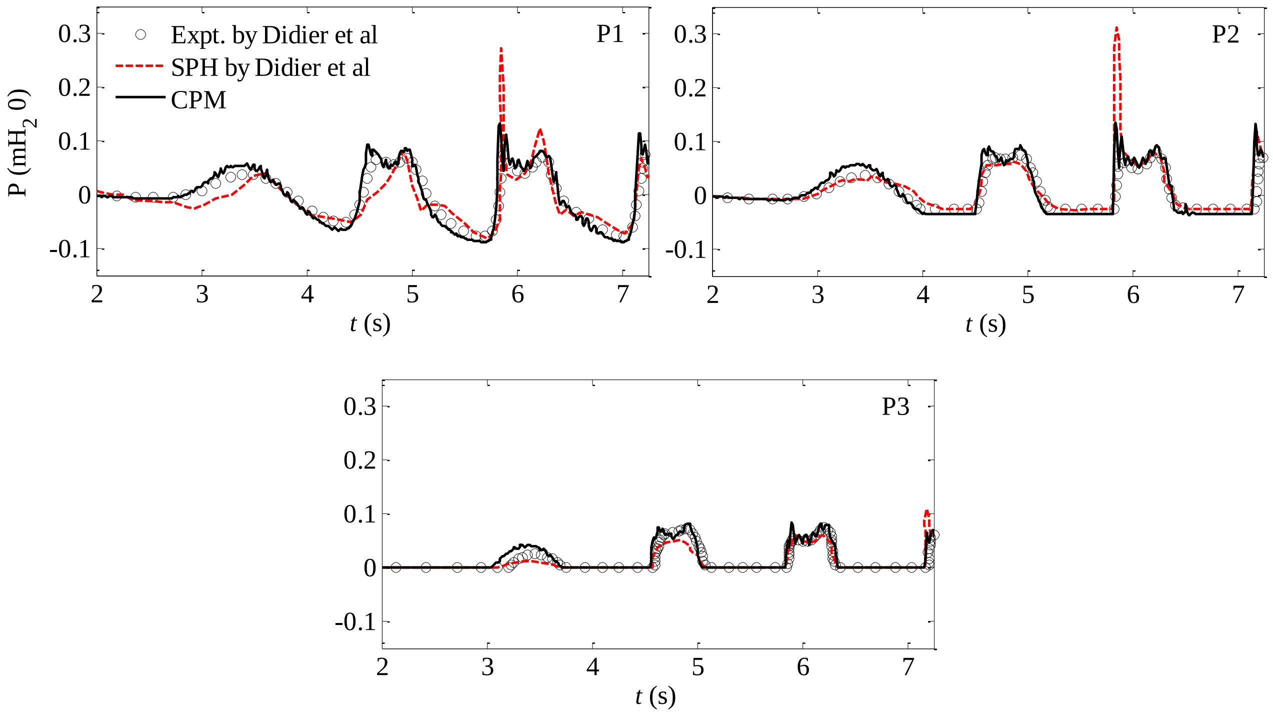


Figure 10. Pressure at P_1, P_2 and P_3: experimental and SPH results by Didier *et al.* [38] and CPM result

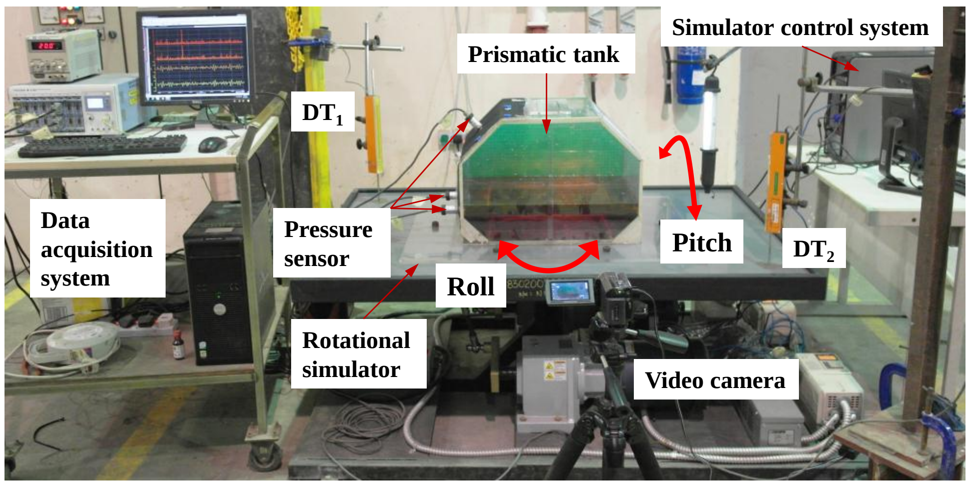


Figure 11. Water sloshing on rotational simulator

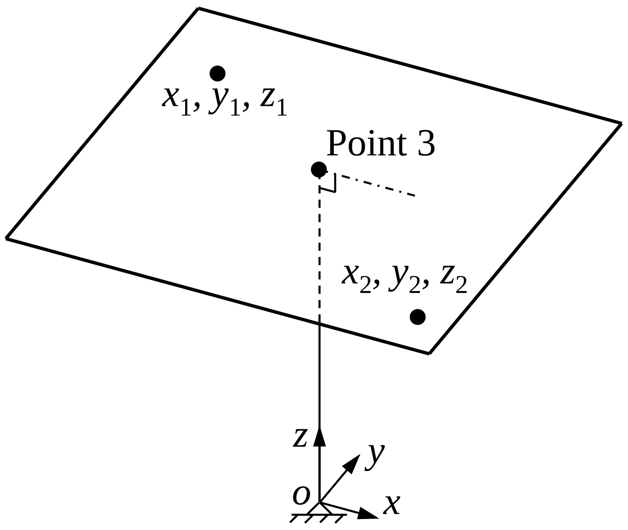
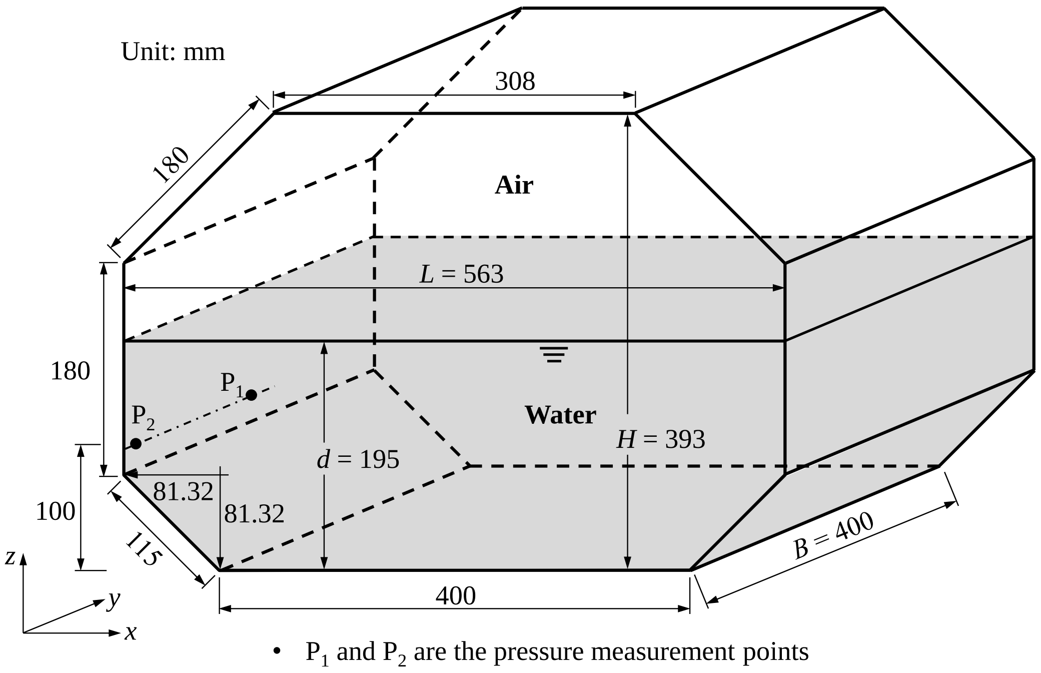


Figure 12. Schematic view of how to measure the rotational angles of a rotational platform



- P$_1$ and P$_2$ are the pressure measurement points

Figure 13. Geometric dimensions of water tank used in sloshing experiment

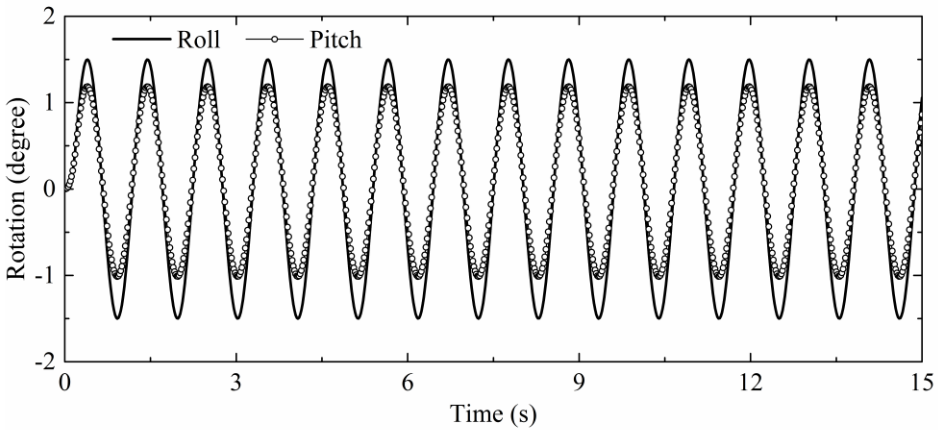


Figure 14. Water sloshing under rotational excitation: roll and pitch angles

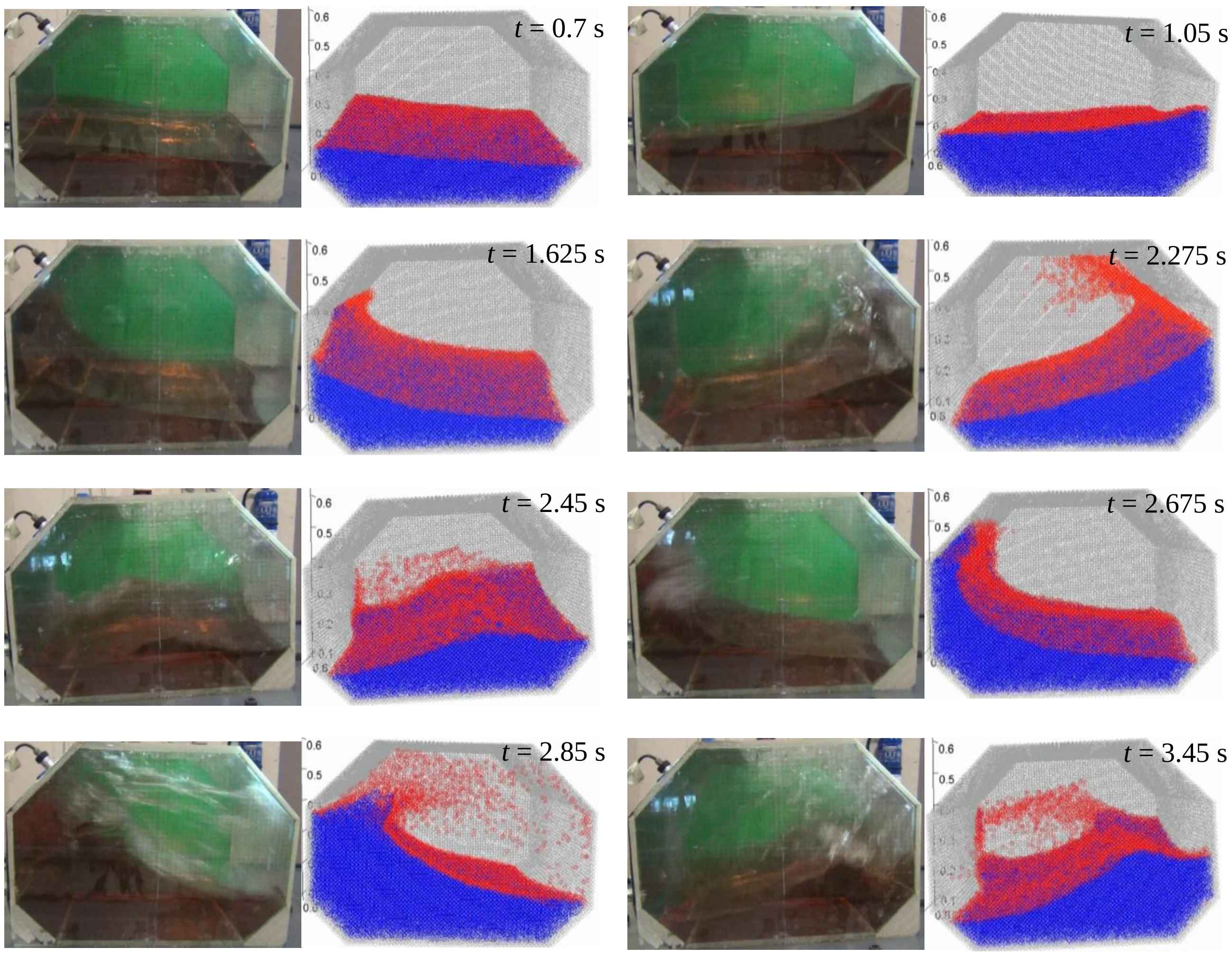


Figure 15. Water sloshing under rotational excitation: wave profiles at typical time instants

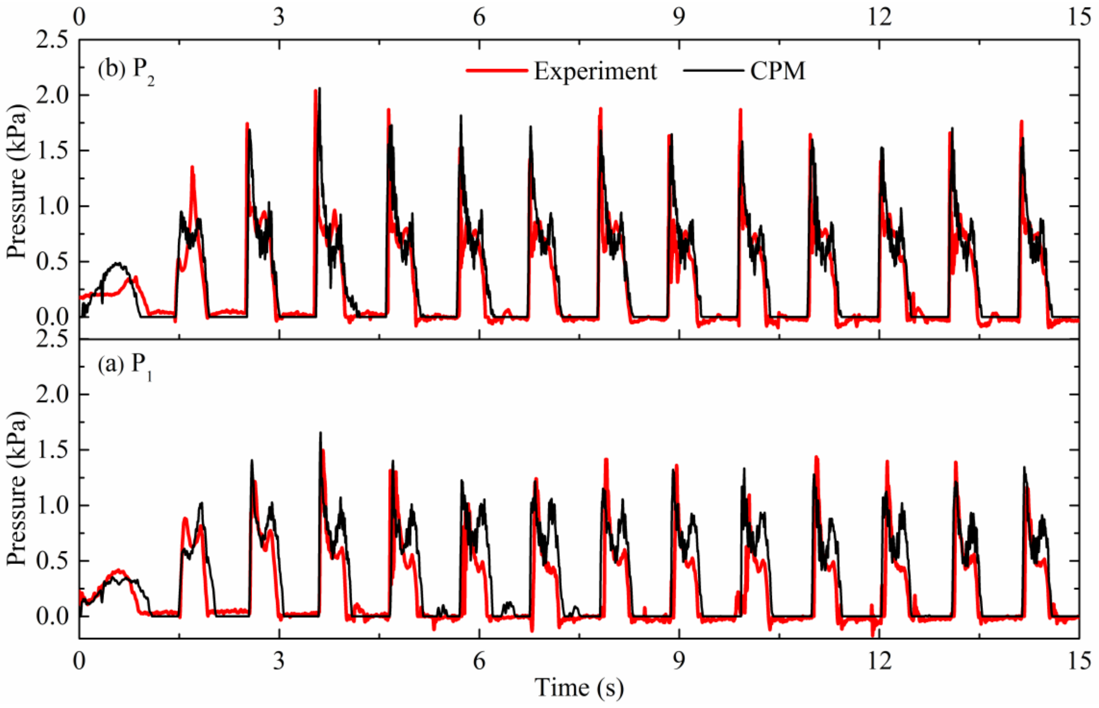


Figure 16. Comparison of pressure histories at P_1 and P_2

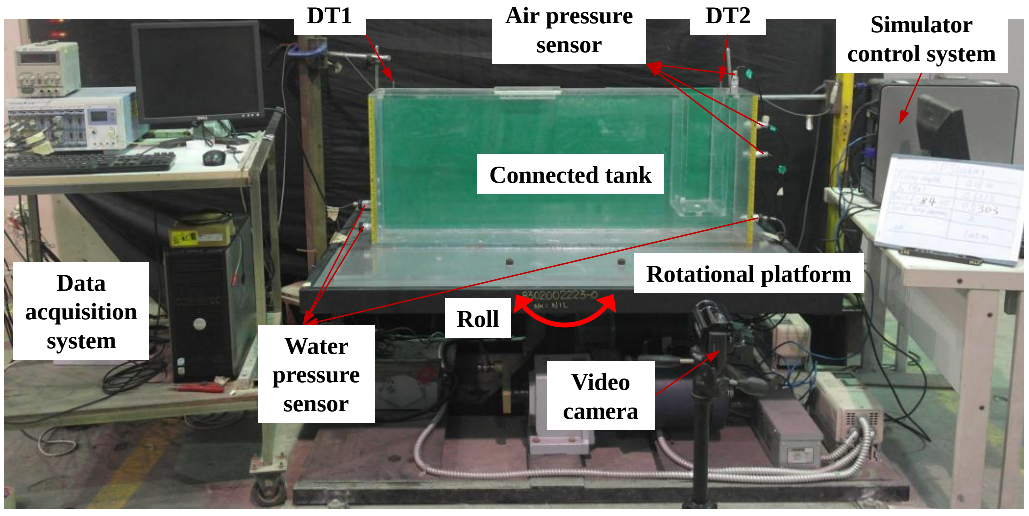


Figure 17. Water-air sloshing in a connected container under rotational excitation

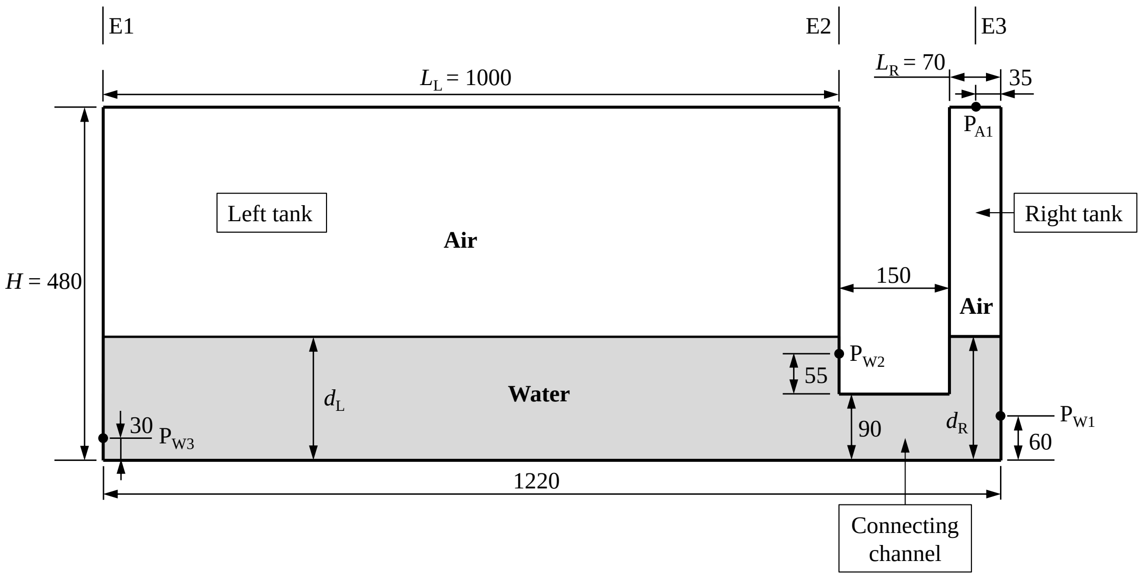
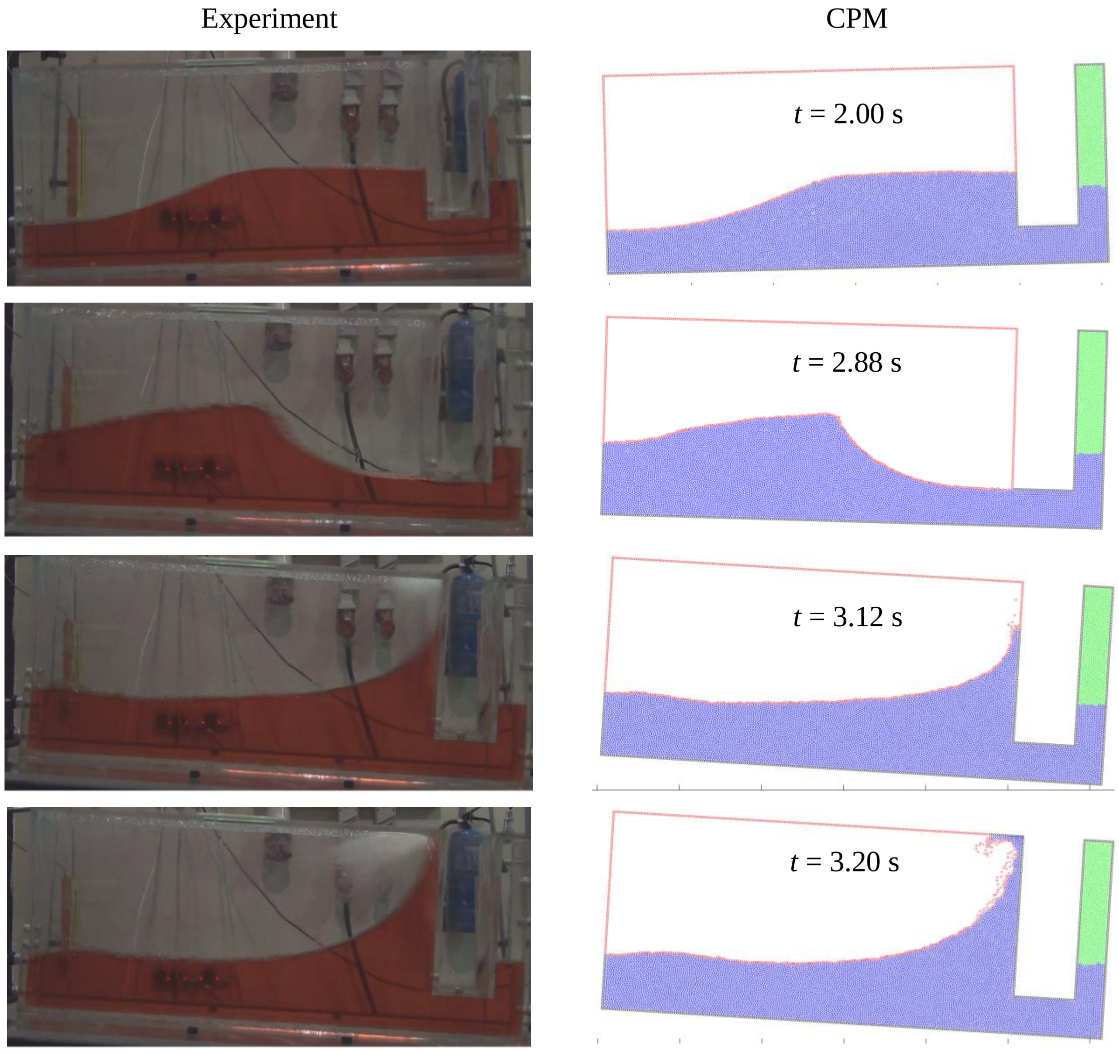


Figure 18. Geometric dimensions of the connected container used in experiment (Unit: mm)



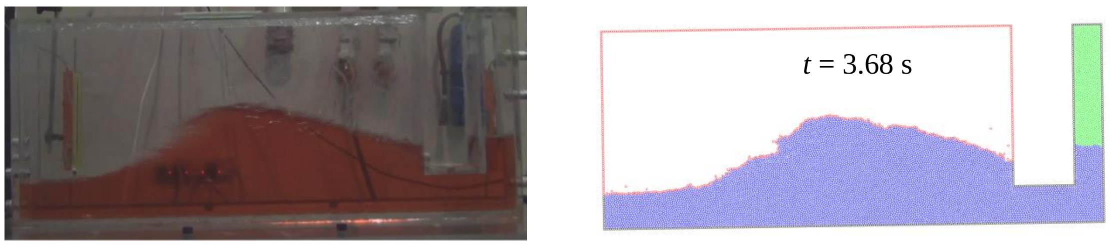


Figure 19. Wave profiles of sloshing with closed air pocket under rotational excitation: experimental result and CPM simulation

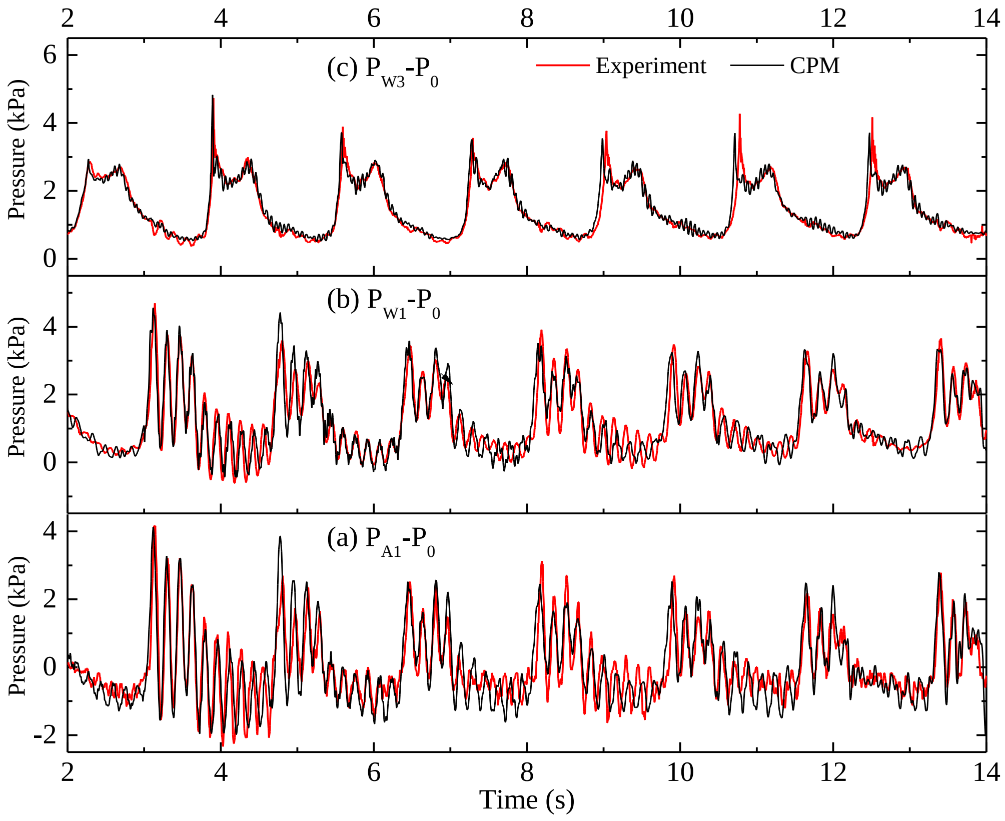


Figure 20. Simulated air pressure at Point P_{A1} and water pressures at Point P_{W1} and P_{W3} in comparison with experimental results

Table 1. Execution times of sequential GMRES and 1-thread Pardiso

Task No.	Task size	N_p (10^3)	Function solver using 1 thread		
			t_{SG} (s)	t_{SP1} (s)	t_{SG} / t_{SP1}
T_1	1	25.6	4.26	2.24	1.90
T_2	2	51.2	12.49	5.16	2.42
T_3	4	102.4	39.83	11.63	3.43
T_4	8	204.8	134.10	25.57	5.24
T_5	16	409.6	477.80	53.40	8.95

T_6	32	819.2	1802.32	106.51	16.92
T_7	64	1638.4	7153.56	214.33	33.38

Table 2. Execution times of Pardiso function solver and all other subroutines with different thread numbers

Task No.	Pardiso function solver			All other subroutines		
	t_{SP1} (s)	t_{SP16} (s)	t_{SP1} / t_{SP16}	t_{OS1} (s)	t_{OS16} (s)	t_{OS1} / t_{OS16}
T_1	2.24	0.81	2.76	1.58	0.43	3.68
T_2	5.16	1.59	3.24	3.06	0.78	3.93
T_3	11.63	3.02	3.85	6.14	1.44	4.26
T_4	25.57	5.89	4.34	11.84	2.77	4.27
T_5	53.40	12.27	4.35	23.09	4.81	4.80
T_6	106.51	24.94	4.27	46.74	9.73	4.81
T_7	214.33	50.35	4.26	95.72	19.28	4.96