Comp. Part. Mech. (2015) 2:91–108
DOI 10.1007/s40571-015-0039-6

# Improved pressure calculation for the moving particle semi-implicit method

**Kazuya Shibata**[1] · **Issei Masaie**[2] · **Masahiro Kondo**[3] · **Kohei Murotani**[1] · **Seiichi Koshizuka**[1]

Received: 20 January 2015 / Revised: 13 April 2015 / Accepted: 24 April 2015 / Published online: 8 May 2015

**Abstract** We developed a practical technique for calculating pressures and pressure gradients for the moving particle semi-implicit (MPS) method. Specifically, a new free-surface boundary condition for the pressure Poisson equation was developed by assuming that there are virtual particles over the surface. We treat the pressure of the virtual particles as a known value. A single liquid-phase flow is simulated taking into account the pressure of these virtual particles. A technique for detecting surface particles also was developed and used for accurately imposing the free-surface boundary condition. The pressure gradient model was modified to mitigate particle clustering and a single-layer wall model was developed to reduce the number of wall particles. We applied our technique to several problems, verifying that virtual surface particles suppress pressure oscillations and particle clusterings. The technique enables reliable differences in free-surface pressures to be taken and simulates instances of lower fluid pressure than that of a free surface.

**Keywords** MPS method · SPH · Particle clustering · Pressure oscillation · Boundary condition · Detection of free-surface particles · Low pressure

✉ Kazuya Shibata
shibata.kazuya@sys.t.u-tokyo.ac.jp

[1] Department of Systems Innovation, School of Engineering, The University of Tokyo, 7-3-1, Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

[2] Prometech Software Inc., Hongo Dai-ichi Building, 8th Floor, 34-3, Hongo 3-chome, Bunkyo-ku, Tokyo 113-0033, Japan

[3] Department of Nuclear Engineering and Management, School of Engineering, The University of Tokyo, 7-3-1, Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

## Nomenclature

| | |
|---|---|
| $d$ | Number of space dimensions |
| $A$ | Glass width |
| $A_{illuminated}$ | Illuminated area of virtual screen |
| $A_{whole}$ | Whole area of virtual screen |
| $B$ | Tank width |
| $C_s$ | Parameter for upper limit of source term of Poisson equation |
| $C_\phi, C_\theta$ | Size of small segment of virtual screen |
| $f_k$ | One-dimensional array for detecting surface particles |
| $f_{k,l}$ | Two-dimensional array for detecting surface particles |
| $\vec{g}$ | Gravitational acceleration |
| $h$ | Water level or height of pull-up glass |
| $h'$ | Relative height of glass bottom from water level |
| $l_0$ | Initial spacing between particles |
| $N$ | Number of particles |
| $N_\Theta,\ N_\phi,\ N_\theta$ | Number of small segments of virtual screen |
| $n^*$ | Temporary particle number density after explicit phase |
| $n_i$ | Particle number density of $i$-th particle |
| $n^0$ | Constant of particle number density |
| $P$ | Pressure |
| $r$ | Distance between particles |
| $\vec{r}$ | Position vector of particle |
| $\vec{r}_i$ | Position vector of $i$-th particle |
| $\vec{r}^*$ | Temporary position vector of particle after explicit phase |
| $R_{illuminated}$ | Ratio of illuminated area |
| $r_e$ | Radius of interaction zone |
| $r_{e,gradient}$ | Radius of interaction zone for gradient model and particle number density |

| | |
|---|---|
| $r_{e,Laplacian}$ | Radius of interaction zone for Laplacian model |
| $S_{wall}$ | Set of wall particles |
| $S_{virtual}$ | Set of virtual particles |
| $t$ | Time |
| $T$ | Glass thickness |
| $\vec{u}$ | Flow velocity vector |
| $\vec{u}^*$ | Vector of temporary flow velocity after the explicit phase of the MPS method |
| $u$ | Flow velocity in x-direction |
| $v$ | Flow velocity in y-direction |
| $V_e$ | Volume of interaction zone |
| $w$ | Weight function |
| $\alpha$ | Parameter for detecting free-surface particles |
| $\beta$ | Free-surface parameter |
| $\theta$ | Direction of neighboring particle |
| $\Theta$ | Angle of illuminated area |
| $\phi$ | Arbitrary quantity or direction of neighboring particle |
| $\lambda^0$ | Parameter for Laplacian model |
| $\mu$ | Viscosity coefficient |
| $\nu$ | Kinematic viscosity coefficient |
| $\rho$ | Fluid density |

## 1 Introduction

Lagrangian methods, such as the moving particle semi-implicit (MPS) method [1], smoothed particle hydrodynamics (SPH) [2,3], and the particle finite element method [4,5] have been applied to various complex flows. For example, Shibata et al. simulated green water on a deck [6,7], ship motions in rough seas [8], and free-fall lifeboat [9] using the MPS method. Xiong et al. [10] simulated droplet impingement for erosion of steam generator tubes using the MPS-for-all-speed-flow method [11]. Tsukamoto et al. [12] used the MPS method to investigate the effects of an elastically-linked body on tank sloshing. Lee et al. [13] used the MPS method for simulating liquid-entry and slamming problems in two dimensions.

Various numerical models have also been developed for the MPS method. For example, Shibata et al. [14] developed a transparent boundary condition for open boundaries to simulate nonlinear water waves using the MPS method. Algorithms for suppressing the pressure oscillations in the MPS method were developed by Kondo and Koshizuka [15], Hibi and Yabushita [16], Tanaka and Masunaga [17], Khayyer and Gotoh [18–20], Lee et al. [21] and Tamai and Koshizuka [22]. Tanaka et al. [23] proposed a multi-resolution technique for the MPS method. Suzuki et al. [24] developed the Hamiltonian moving particle semi-implicit method, and showed excellent performance in conserving mechanical energy.

There has also been much progress with SPH and the particle finite element method. For example, Kajtar and Monaghan [25,26] simulated the motion of three linked ellipses through a viscous fluid in two dimensions using SPH. Khayyer et al. [27] proposed a corrected incompressible SPH for accurately tracking the water surface in breaking waves. They also performed enhanced wave impact simulations using the improved incompressible SPH [28]. Antuono et al. [29,30] simulated free-surface flows using SPH schemes with numerical diffusive terms. They showed that the diffusive terms enabled reducing the high-frequency numerical acoustic noise and smoothing of the pressure field. Cleary [31] extended SPH to fluid phenomena in low-pressure die casting, in which he simulated the solidification and freezing of an injected metal. Idelsohn et al. [32] simulated multi-fluid flows using the particle finite element method. They demonstrated the mixing of fluids with different densities. Lagrangian methods have therefore been studied actively.

In this study, we develop a practical technique for calculating pressure and pressure gradient for the MPS method. In the standard MPS method, Dirichlet boundary conditions are applied directly to free-surface particles. Therefore, particle interactions for the pressure gradient term do not work between free-surface particles, and particles cluster on the free surface. Our proposed technique overcomes this problem by assuming virtual particles over the free surface (Fig. 1). The boundary condition of the virtual particle is different from that of the ghost particle model by Schechter et al. [33] because the virtual particles are considered in the pressure Poisson equations without actually placing virtual particles in the simulation domain. We treat the pressures of the virtual particles as known values. A single liquid-phase flow is simulated considering the pressure of virtual particles. A new technique for detecting surface particles also was developed and used for accurately imposing the free-surface boundary condition. Moreover, the developed boundary condition for the pressure Poisson equation enables differences in free-surface pressures to be taken and instances of lower fluid pressures than that of a free surface to be simulated. Such low pressure needs to be considered in many fluid problems in industrial fields. For example, hydrofoil wings and blenders generate low pressure. However, it has been difficult for the MPS method to simulate pressures below that over a free surface because the simulations become unstable in some instances. Hence, such pressures are usually set to zero in MPS simulations. The boundary condition we developed overcomes this problem. A single-layer wall model also was developed to reduce the number of wall particles. We applied the technique to several problems.

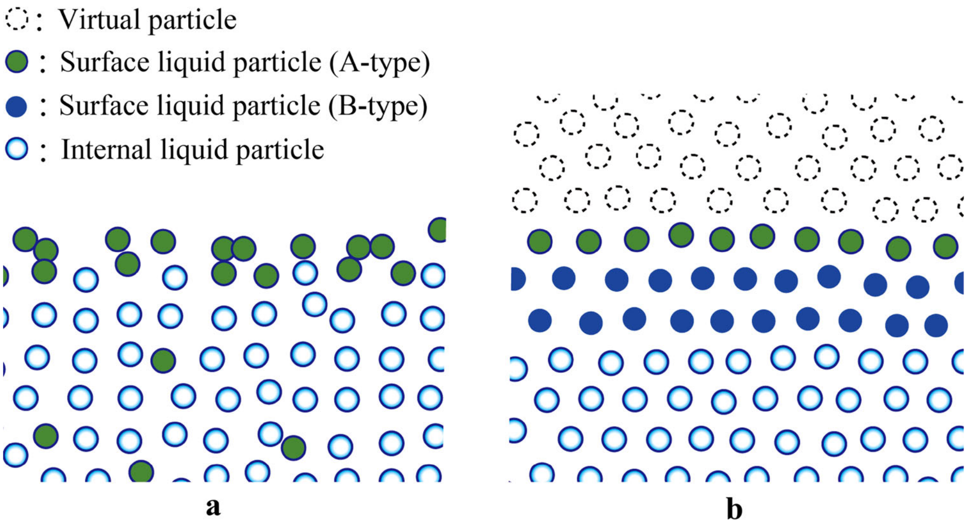

○ : Virtual particle
● : Surface liquid particle (A-type)
● : Surface liquid particle (B-type)
○ : Internal liquid particle
a
b

**Fig. 1** Schematic of free-surface boundary. **a** Standard pressure calculation; **b** improved pressure calculation

## 2 MPS method[1]

### 2.1 Governing equations

The governing equations of the MPS method are

$$\frac{D\rho}{Dt} + \rho\nabla\cdot\overrightarrow{u} = 0 \tag{1}$$

and

$$\rho\frac{D\overrightarrow{u}}{Dt} = -\nabla P + \mu\nabla^2\overrightarrow{u} + \rho\vec{g}. \tag{2}$$

Equations (1) and (2) are the conservation equations representing the mass and momentum for incompressible flows. Numerical diffusion arising from the discretization of the convection term does not occur because we use a fully Lagrangian description.

### 2.2 Particle approximations

In the MPS method, the governing equations are discretized by replacing the differential operators with the gradient and Laplacian models

$$\langle\nabla\phi\rangle_i = \frac{d}{n^0}\sum_{j\neq i}\left[\frac{\phi_j-\phi_i}{\left|\vec{r}_j-\vec{r}_i\right|^2}\left(\vec{r}_j-\vec{r}_i\right)w\left(\left|\vec{r}_j-\vec{r}_i\right|\right)\right] \tag{3}$$

and

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

where $w$ is a weight function given by

$$w(r) = \begin{cases} \left(\frac{r_e}{r}\right)-1 & (r < r_e) \\ 0 & (r \geq r_e) \end{cases}. \tag{5}$$

The particle number density is calculated with the above weight function as

$$n_i = \sum_{j\neq i} w\left(\left|\vec{r}_j-\vec{r}_i\right|\right). \tag{6}$$

The particle number density has two interpretations: one is the value proportional to the fluid density and the other is the normalization factor of the weighted average. The particle number density of internal fluid region should be constant to satisfy the mass conservation equation, Eq. (1). The parameter $n^0$ denotes the constant particle number density. The restraint condition for the particle number density is implicitly included in the pressure Poisson equation, to be discussed in the next section.

The parameter $\lambda^0$ in Eq. (4) is a constant given by

$$\lambda^0 = \frac{\sum_{j\neq\hat{i}}\left|\vec{r}_j^{\,0}-\vec{r}_{\hat{i}}^{\,0}\right|^2 w\left(\left|\vec{r}_j^{\,0}-\vec{r}_{\hat{i}}^{\,0}\right|\right)}{\sum_{j\neq i} w\left(\left|\vec{r}_j^{\,0}-\vec{r}_{\hat{i}}^{\,0}\right|\right)}, \tag{7}$$

where the index of $\hat{i}$ is the identification number of a particle located at internal region in fluid at the initial time step. This $\lambda^0$ is used as a common value among particles. This parameter corrects the variance increase of the Laplacian model so that the variance agrees with that of the analytical solution.

### 2.3 Algorithm for incompressible flow

The MPS method uses a semi-implicit algorithm. With the exception of the pressure gradient terms, the momentum conservation equations are explicitly (at time step $k$) solved in the first phase. The Poisson equation for pressure is then implicitly (at time step $k+1$) solved in the second phase. The following Poisson equation for pressure is deduced from the implicit mass conservation equation and the implicit pressure gradient term:

$$\left\langle \nabla^2 P \right\rangle_i^{k+1} = -\frac{\rho}{\Delta t^2} \frac{n_i^* - n^0}{n^0}, \tag{8}$$

where $n_i^*$ is the temporary particle number density after the explicit phase. The Laplacian operator on the left-hand side of Eq. (8) is discretized with the Laplacian model, Eq. (4). The right-hand side of Eq. (8) represents the source term, which is expressed as the deviation of the temporary particle number density from the constant particle number density, $n^0$; this is calculated in the initial particle arrangement using

$$n^0 = \sum_{j \neq \hat{i}} w\left(\left|\vec{r}_j^{\,0} - \vec{r}_{\hat{i}}^{\,0}\right|\right), \tag{9}$$

and is used as a common value among particles until the end of the simulation. Consequently, we obtain simultaneous linear equations from Eq. (8) and solve them using the conjugate gradient method.

This semi-implicit algorithm is similar to that of the finite volume method for incompressible flow. However, the right-hand side of the Poisson equation for pressure is different. We use the deviation of the particle number density in Eq. (8), whereas the velocity divergence is used in the finite volume method. For further details of the MPS method, see Khayyer and Gotoh [20], Koshizuka [34], Kondo and Koshizuka [15], Koshizuka and Oka [1], Lee et al. [13], and Tanaka and Masunaga [17].

### 2.4 Standard pressure calculation of the MPS method for free-surface flows

In the standard MPS simulation [1], there are no particles over the fluid surface. Therefore, the particle number density around a free surface is lower than the standard particle number density, $n^0$. Using this feature, particles whose particle number density is below $\alpha n^0$ are regarded as surface particles in the original free-surface boundary condition, where $\alpha$ is a constant scalar parameter ($0 < \alpha \leq 1$) and 0.97 is often used for $\alpha$. Dirichlet boundary conditions are directly applied to the surface particles to solve the pressure Poisson equation. Therefore, particle interactions for the pressure gradient term do not work between free-surface particles, and particles cluster on the free surface. Moreover, internal particles are sometimes mistakenly regarded as surface particles because of the method for detecting surface particles. As a result, the pressure distribution oscillates.

Pressures below that of a free surface are reset to zero to avoid unnatural low pressure around free surfaces. Because of this, it is impossible to simulate such low pressures using the original boundary condition.

## 3 Improved pressure calculation

We now improve the pressure calculation of the MPS method to mitigate pressure oscillation and particle clustering on a free surface, express differences between free-surface pressures, and simulate these low pressures below those of a free surface. Specifically, we modify the pressure Poisson equation and the pressure gradient term by assuming that there are virtual particles over the free surface (Fig 1). Although virtual particles are drawn in Fig. 1 to explain the concept, we do not need to actually locate, generate or move virtual particles because the effects of these particles are considered only mathematically by introducing some assumptions. This is an advantage of the virtual particles. Moreover, we develop a new method for accurately detecting A-type liquid surface particles on the free surface (see Fig. 1), which enables us to impose the free-surface boundary condition accurately. The A-type particle surface has a thickness of just one particle.

We apply our modified pressure Poisson equation to the surface liquid particles (A-type and B-type) shown in Fig. 1b, where B-type particles are particles whose $n^*$ is smaller than $n^0$, and the distance from the nearest A-type particles is shorter than $2.1 l_0$. The details are mentioned in the following sections.

### 3.1 Pressure Poisson equation

In the standard MPS method [1], the Poisson equation for pressure is given by Eq. (8). Using the Laplacian model of the MPS method given by Eq. (4), Eq. (8) is discretized as

$$\frac{2d}{\lambda^0 n^0} \sum_{j \neq i} \left(P_j^{k+1} - P_i^{k+1}\right) w\left(\left|\vec{r}_j^{\,*} - \vec{r}_i^{\,*}\right|\right) = -\gamma \frac{\rho^0}{(\Delta t)^2} \frac{\left(n_i^* - n^0\right)}{n^0}, \tag{10}$$

where $\gamma$ is a scalar parameter for stabilizing the simulation. The parameter $\rho^0$ is the initial fluid density, which is a common value of particles. When the $i$-th particle is an A- or B-type surface particle, we can assume that virtual particles are located over the free surface (Fig 1b). Therefore, in this case, we divide the left-hand side of Eq. (10) into two terms for the actual particles and virtual particles as follows:

$$\frac{2d}{\lambda^0 n^0}\sum_{j\neq i}\left(P_j^{k+1}-P_i^{k+1}\right)w\left(\left|\vec{r}_j^{\,*}-\vec{r}_i^{\,*}\right|\right)$$
$$+\frac{2d}{\lambda^0 n^0}\left(P_{virtual}-P_i^{k+1}\right)n_{i,virtual}$$
$$=-\gamma\frac{\rho^0}{(\Delta t)^2}\frac{\left(n_i^*-\tilde{n}_i^k\right)}{n^0}. \tag{11}$$

The first term on the left-hand side of Eq. (11) is the summation for actual neighboring particles around the $i$-th particle, while the second is the term due to the virtual neighboring particles. We assume that the pressure of the virtual particles is a constant, $P_{virtual}$. The parameter, $n_{i,virtual}$, is the particle number density for the virtual particles defined as

$$n_{i,virtual}\equiv\max\left\{n^0-n_i^*,0\right\}. \tag{12}$$

By using the assumption of Eq. (12), we do not need to actually locate or move virtual particles. To simply express equations, the term for compressibility of fluid that is used in the usual MPS simulations [1,17] is not written in Eqs. (10) and (11), although the term was added in the actual simulation of this study.

For the pressure Poisson equation, we apply Dirichlet boundary conditions only to the virtual particles. Therefore, the pressure of actual free-surface particles is not fixed to the gas-phase pressure. As a result, particle clustering is suppressed because of the particle interaction based on the pressure difference between free-surface particles. Moreover, compared with the standard pressure calculation, the second term on the left-hand side of Eq. (11) increases the free-surface particle pressure by about $\rho g\sqrt{\lambda^0}/2$. This pressure increment has the effect of sustaining free-surface particles at a certain water level using the modified pressure gradient model to be described below. When we need to compare pressures obtained from simulations with those of the analytical solution or experimental data, we need to subtract the simulated pressure by $\rho g\sqrt{\lambda^0}/2$. We use Eq. (11) for both A- and B-type surface liquid particles. For the other particles, we use Eq. (11) without the second term on the left-hand side.

Compared with Eq. (10), the right-hand side of Eq. (11) was modified by the following approximation based on Eq. (1):

$$\nabla\cdot\mathbf{u}_i^*=-\frac{1}{\rho^0}\frac{D\rho_i}{Dt}\cong-\frac{1}{\frac{n^0}{V_e}}\frac{\left(\frac{n_i^*}{V_e}-\frac{\tilde{n}_i^k}{V_e}\right)}{\Delta t}=-\frac{1}{\Delta t}\frac{\left(n_i^*-\tilde{n}_i^k\right)}{n^0}, \tag{13}$$

where $\tilde{n}_i^k$ is the standard number density of the $i$-th particle at the time step $k$. To produce stable simulations, we use instead of $n^0$ particle number density $\tilde{n}_i^k$,

$$\tilde{n}_i^k=\begin{cases}n^0 & if\ n_i^k\geq n^0\\ n_i^k & otherwise\end{cases}. \tag{14}$$

Hence, if $n_i^k$ is greater than $n^0$, $n^0$ is used for $\tilde{n}_i^k$ to prevent the fluid-volume compression, otherwise, $n_i^k$ is used. When we need to treat pressures below that for the free-surface particles, we use the following $\tilde{n}_i^k$:

$$\tilde{n}_i^k=\begin{cases}n^0 & if\ n_i^k\geq n^0\\ C_s n^0 & if\ n_i^k\leq n_i^{k-1}\ and\ n_i^k\leq\alpha n^0\\ n_i^k & otherwise\end{cases}, \tag{15}$$

where $C_s$ is a scalar parameter ($0<C_s\leq 1$); in this study, 1.0 was used for $C_s$. The density,$C_s n^0$, provides a lower limit in the particle number density and prevents occurrence of unnatural cavities in the fluid. Equation (15) is applied to internal particles, whereas Eq. (14) is applied to A- and B-type surface particles.

In the standard MPS method, 0.2 is often used for $\gamma$ in Eq. (10). In this study, 0.05 and 0.10 were used for $\gamma$ of fluid and wall particles, respectively, to suppress pressure oscillations. The values of $\gamma$ were empirically determined. Although it is not necessary to frequently change $\gamma$ depending on circumstances, the following ineq uality gives a guideline for setting $\gamma$:

$$1\geq\gamma\geq\left|\frac{n^{**}-n^0}{n^0}\right|^{-1}\frac{(\Delta t)^2 d}{\sqrt{\lambda^0}}\left|\frac{1}{\rho^0}\nabla P\right|_{\max}>0, \tag{16}$$

where $n^{**}$ is the permissible upper limit of temporal particle number density ($n^{**}>n^0$) after the explicit phase of the MPS method, and $\left|\frac{1}{\rho^0}\nabla P\right|_{\max}$ is the expected maximum acceleration due to pressure gradient term on wall boundaries.

### 3.2 Modified pressure gradient calculation

Based on Eq. (3), the pressure gradient term in the Navier-Stokes equation is discretized as follows:

$$\langle\nabla P\rangle_i=\frac{d}{n^0}\sum_{j\neq i}\frac{P_j-P_i}{\left|\vec{r}_j-\vec{r}_i\right|^2}\left(\vec{r}_j-\vec{r}_i\right)w\left(\left|\vec{r}_j-\vec{r}_i\right|\right). \tag{17}$$

To avoid instabilities in simulations arising from the attractive forces described in Eq. (17), the following pressure gradient is often used in the standard MPS method:

$$\langle\nabla P\rangle_i=\frac{d}{n^0}\sum_{j\neq i}\frac{P_j-\hat{P}_i}{\left|\vec{r}_j-\vec{r}_i\right|^2}\left(\vec{r}_j-\vec{r}_i\right)w\left(\left|\vec{r}_j-\vec{r}_i\right|\right), \tag{18}$$

where $\hat{P}_i$, the minimum pressure around the $i$-th particle, is defined as:

$$\hat{P}_i = \min\left\{ P_i, P_j \;\middle|\; \left|\vec{r}_j - \vec{r}_i\right| \le r_e \right\}. \tag{19}$$

Equation (18) is derived from Eq. (17) assuming Eq. (21):

$$\begin{aligned}\langle \nabla P \rangle_i &\cong \frac{d}{n^0} \sum_{j \ne i} \frac{P_j - P_i}{\left|\vec{r}_j - \vec{r}_i\right|^2} \left(\vec{r}_j - \vec{r}_i\right) w\left(\left|\vec{r}_j - \vec{r}_i\right|\right) \\ &\quad + \left(P_i - \phi_i\right) \frac{d}{n^0} \sum_{j \ne i} \frac{1}{\left|\vec{r}_j - \vec{r}_i\right|^2} \left(\vec{r}_j - \vec{r}_i\right) w\left(\left|\vec{r}_j - \vec{r}_i\right|\right) \\ &= \frac{d}{n^0} \sum_{j \ne i} \frac{\left\{P_j + \left(P_i - \phi_i\right)\right\} - P_i}{\left|\vec{r}_j - \vec{r}_i\right|^2} \left(\vec{r}_j - \vec{r}_i\right) w\left(\left|\vec{r}_j - \vec{r}_i\right|\right) \\ &= \frac{d}{n^0} \sum_{j \ne i} \frac{P_j - \phi_i}{\left|\vec{r}_j - \vec{r}_i\right|^2} \left(\vec{r}_j - \vec{r}_i\right) w\left(\left|\vec{r}_j - \vec{r}_i\right|\right), \end{aligned} \tag{20}$$

$$\frac{1}{n^0} \sum_{j \ne i} \frac{1}{\left|\vec{r}_j - \vec{r}_i\right|^2} \left(\vec{r}_j - \vec{r}_i\right) w\left(\left|\vec{r}_j - \vec{r}_i\right|\right) \cong \vec{0}, \tag{21}$$

where $\phi_i$ is an arbitrary scalar parameter of $i$-th particle and $\hat{P}_i$ is used for $\phi_i$ in the standard MPS method. In Eq. (21), it is assumed that neighboring particles of the $i$-th particle are arranged isotropically.

In this study, the following modified gradient model, which was derived by applying the $\tilde{P}_i$ for $\phi_i$ in Eq. (20), was used for all the particles.

$$\langle \nabla P \rangle_i = \frac{d}{n^0} \sum_{j \ne i} \frac{P_j - \tilde{P}_i}{\left|\vec{r}_j - \vec{r}_i\right|^2} \left(\vec{r}_j - \vec{r}_i\right) w\left(\left|\vec{r}_j - \vec{r}_i\right|\right), \tag{22}$$

$$\tilde{P}_i = \min\left\{ P_i, P_j, P_{virtual} \;\middle|\; \left|\vec{r}_j - \vec{r}_i\right| \le r_e \right\}. \tag{23}$$

Compared with Eq. (17) and Eq. (18), Eq. (22) maintains particle distance to the initial value because the repulsive force from the neighboring $j$-th particle is virtually increased by increasing its pressure by $P_i - \tilde{P}_i$ as shown in the second line of Eq. (20). If $P_{virtual} = 0\,\mathrm{Pa}$ and $P_j, P_i \ge 0$, Eq. (22) agrees with the following gradient model:

$$\langle \nabla P \rangle_i = \frac{d}{n^0} \sum_{j \ne i} \frac{P_j}{\left|\vec{r}_j - \vec{r}_i\right|^2} \left(\vec{r}_j - \vec{r}_i\right) w\left(\left|\vec{r}_j - \vec{r}_i\right|\right). \tag{24}$$

On free surfaces, presuming approximation Eq. (21), which was used for the derivation of Eq. (24), holds true, is not viable because there is no particle above the free surface. However, by using the concept of virtual particles, we can obtain the same equation as Eq. (24) for free-surface particles. This is discussed below. By assuming that there are virtual particles over the free surface and they are arranged isotropically, we assume the following approximation to express the pressure gradient of $i$-th particle [Eq. (17)], which is located around the free surface.

$$\begin{aligned}\langle \nabla P \rangle_i = &\frac{d}{n^0} \sum_{\substack{j \ne i \\ j \notin S_{virtual}}} \frac{\left(P_j - P_i\right)}{\left|\vec{r}_j - \vec{r}_i\right|^2} \left(\vec{r}_j - \vec{r}_i\right) w\left(\left|\vec{r}_j - \vec{r}_i\right|\right) \\ &+ \frac{d}{n^0} \sum_{\substack{j \ne i \\ j \in S_{virtual}}} \frac{\left(P_{virtual} - P_i\right)}{\left|\vec{r}_j - \vec{r}_i\right|^2} \left(\vec{r}_j - \vec{r}_i\right) w\left(\left|\vec{r}_j - \vec{r}_i\right|\right). \end{aligned} \tag{25}$$

We now suppose that neighboring particles including virtual and actual particles are arranged isotropically around the $i$-th particle. Then, we can assume the same approximation as for Eq. (21). Indeed, the left-hand side of Eq. (21) can be divided into two terms for actual particles and virtual particles as follows:

$$\begin{aligned}&\frac{1}{n^0} \sum_{\substack{j \ne i \\ j \notin S_{virtual}}} \frac{\left(\vec{r}_j - \vec{r}_i\right)}{\left|\vec{r}_j - \vec{r}_i\right|^2} w\left(\left|\vec{r}_j - \vec{r}_i\right|\right) \\ &+ \frac{1}{n^0} \sum_{\substack{j \ne i \\ j \in S_{virtual}}} \frac{\left(\vec{r}_j - \vec{r}_i\right)}{\left|\vec{r}_j - \vec{r}_i\right|^2} w\left(\left|\vec{r}_j - \vec{r}_i\right|\right) \cong \vec{0}. \end{aligned} \tag{26}$$

Therefore, the summation for the virtual particles is expressed by that for actual particles as follows:

$$\begin{aligned}&\frac{1}{n^0} \sum_{\substack{j \ne i \\ j \in S_{virtual}}} \frac{\left(\vec{r}_j - \vec{r}_i\right)}{\left|\vec{r}_j - \vec{r}_i\right|^2} w\left(\left|\vec{r}_j - \vec{r}_i\right|\right) \cong \\ &- \frac{1}{n^0} \sum_{\substack{j \ne i \\ j \notin S_{virtual}}} \frac{\left(\vec{r}_j - \vec{r}_i\right)}{\left|\vec{r}_j - \vec{r}_i\right|^2} w\left(\left|\vec{r}_j - \vec{r}_i\right|\right). \end{aligned} \tag{27}$$

Substituting Eq. (27) into the second term on the right-hand side of Eq. (25), we obtain the following equation:

$$\begin{aligned}\langle \nabla P \rangle_i = &\frac{d}{n^0} \sum_{\substack{j \ne i \\ j \notin S_{virtual}}} \frac{\left(P_j - P_i\right)}{\left|\vec{r}_j - \vec{r}_i\right|^2} \left(\vec{r}_j - \vec{r}_i\right) w\left(\left|\vec{r}_j - \vec{r}_i\right|\right) \\ &- \frac{d}{n^0} \sum_{\substack{j \ne i \\ j \notin S_{virtual}}} \frac{\left(P_{virtual} - P_i\right)}{\left|\vec{r}_j - \vec{r}_i\right|^2} \left(\vec{r}_j - \vec{r}_i\right) w\left(\left|\vec{r}_j - \vec{r}_i\right|\right). \end{aligned} \tag{28}$$

By combining the summations, we obtain the following modified gradient model for pressure:

$$\langle \nabla P \rangle_i = \frac{d}{n^0} \sum_{\substack{j \ne i \\ j \notin S_{virtual}}} \frac{\left(P_j - P_{virtual}\right)}{\left|\vec{r}_j - \vec{r}_i\right|^2} \left(\vec{r}_j - \vec{r}_i\right) w\left(\left|\vec{r}_j - \vec{r}_i\right|\right). \tag{29}$$

For $P_{virtual} = 0\,\mathrm{Pa}$, we obtain the following model, which agrees with Eq. (24),

$$\langle \nabla P \rangle_i = \frac{d}{n^0} \sum_{\substack{j \neq i \\ j \notin S_{virtual}}} \frac{P_j}{\left|\vec{r}_j - \vec{r}_i\right|^2} \left(\vec{r}_j - \vec{r}_i\right) w\left(\left|\vec{r}_j - \vec{r}_i\right|\right). \tag{30}$$

If particles are not arranged isotropically or located on the free surface, the gradient model of Eq. (24) for these particles does not have first-order accuracy because of the assumptions on particle arrangement that have been introduced. However, the benefits of this model are that it is simple, easy to implement in program code, and maintains particle spacing between particles.

### 3.3 Single-layer wall model

There are various ways to express wall boundaries for particle methods [21,35–39]. In usual MPS simulations, a wall boundary consists of a double layer of wall particles, with its associated pressure parameter, and a double layer of dummy wall particles, which have no pressure parameter and are used only for calculating particle number densities of wall particles. This increases the number of particles, the computational time and the needed size of memory. To solve this problem, we developed a technique which allows us to represent a wall boundary by a single layer of wall particles. The details are given in the following.

First, to find the contribution to the particle number density from outer dummy wall particles without actually locating the dummy wall particles, we add the following particle number density, $n_{i,correction}$, to those of all the wall particles:

$$n_{i,correction} \equiv \left(n^0 - n^0_{i,wall}\right) C_{shape}, \tag{31}$$

where $n^0_{i,wall}$ is the particle number density of $i$-th wall particle calculated at the initial time step considering only wall particles as follows:

$$n^0_{i,wall} = \sum_{\substack{j \neq i \\ j \in S_{wall}}} w\left(\left|\vec{r}^{\,0}_j - \vec{r}^{\,0}_i\right|\right). \tag{32}$$

The scalar parameter $C_{shape}$ characterizes the wall shape. In this study, we assumed that the wall can be locally approximated as a plane, using a value of 0.5 for $C_{shape}$ to simplify the problem.

Moreover, to prevent penetration of fluid particles through the walls, the following vector $\vec{b}_i$ was added to the pressure gradient term (Eq. (22)) of the $i$-th fluid particle located within $1.0 l_0$ from a wall:

$$\vec{b}_i = \frac{d}{n^0} \sum_{\substack{j \neq i \\ j \in S_{wall}}} \frac{P_j}{\left|\vec{r}_{j,virtual} - \vec{r}_i\right|^2} \left(\vec{r}_{j,virtual} - \vec{r}_i\right) w\left(\left|\vec{r}_{j,virtual} - \vec{r}_i\right|\right). \tag{33}$$

The vector, $\vec{r}_{j,virtual}$, in the above equation is the position of a virtual wall particle which is located behind $j$-th wall particle and was simply calculated from:

$$\vec{r}_{j,virtual} = \vec{r}_j - l_0 \vec{n}_{\perp j}. \tag{34}$$

where $\vec{n}_{\perp j}$ is the normal vector of the actual $j$-th wall particle. To simplify the problem in calculating Eq. (33), we assume that the virtual wall particle has the same pressure as that of the $j$-th wall particle, $P_j$. Therefore, it is not necessary to calculate or interpolate the pressure for the virtual wall particles. Moreover, it is not necessary to record the positions of these particles because they are calculated by Eq. (34). These make the difference from techniques of early studies [21,35–39] based on ghost particles or mirror particles. The virtual particle of the $j$-th wall particle is considered also in the calculation of the particle number density of fluid particles around the wall boundary.

However, some splashed fluid particles can penetrate a wall in three-dimensional simulations. To prevent this, we used as an exceptional procedure a penalty force, which is often used in the discrete element method [40], for penetrating particles located within $0.30 l_0$ from a wall.

### 3.4 Detection of surface particles

Surface particles need to be detected to impose Dirichlet boundary conditions on the pressure Poisson equation, and is important in accurately imposing boundary effects. There are various ways to detect boundary particles for particle methods [1,5,17,21,41–43]. In the standard MPS method [1], particles whose particle number density is below $\alpha n^0$ are regarded as surface particles. However, in some cases, internal particles are mistakenly identified as surface particles using this method. Hence, we developed a new and accurate surface detection method using a virtual light source and virtual screen. We explain its details in the following.

#### *3.4.1 Two-dimensional simulation*

Figure 2 shows a schematic of the surface detection method we have developed for two-dimensional simulations. To judge whether the $i$-th particle is on the surface, we use a virtual light source and a virtual screen. The shape of this screen in two dimensions is a circle whose radius is $2.1 l_0$. The radius is empirically determined. The centers of the screen and source are located on the center of the $i$-th particle. If the $i$-th particle is on the surface, most of the area of the

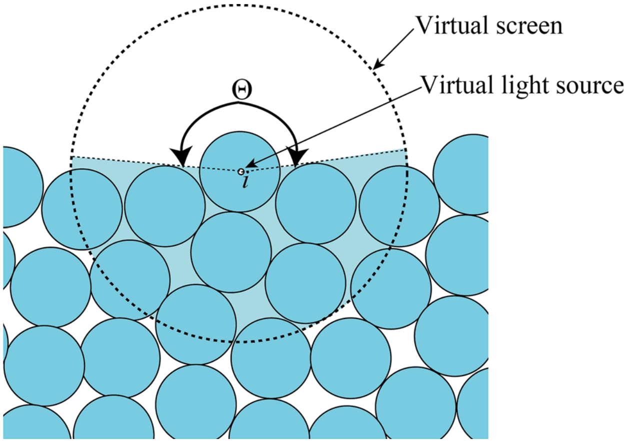


**Fig. 2** Our surface detection method in two dimensions. The *colored area* denotes a region under the free surface. (Color figure online)

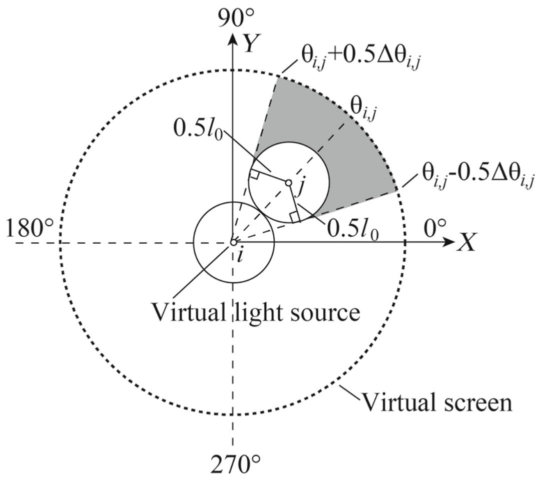


**Fig. 3** Two-dimensional surface detection algorithm using a virtual light source

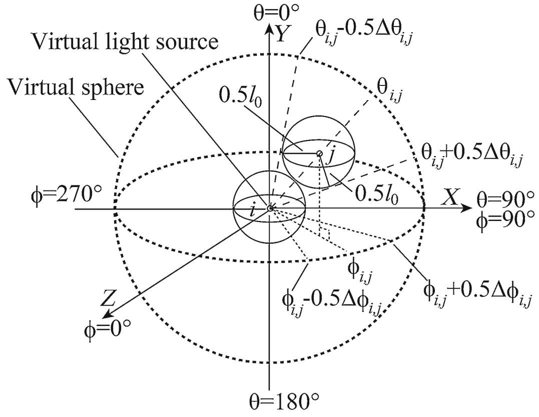


**Fig. 4** Three-dimensional surface detection algorithm using a virtual light source

screen is illuminated by the source, although some area is in the shadow of neighboring particles. The fraction of the area illuminated on the screen is denoted by $R_{illuminated}$. If $R_{illuminated}$ is over a threshold, the particle is regarded as a surface particle. If not, the particle is regarded as an internal particle. We have empirically determined the threshold to be 1/6. To exclude narrow gaps between particles, small illuminated areas whose subtended angle is smaller than 15° are regarded as shaded areas and are exempt in the calculation of $R_{illuminated}$. The shaded area of the $j$-th neighboring particle is calculated as

$$\theta_{i,j} = \left(\frac{180}{\pi}\right) \tan^{-1}\left(\frac{y_j - y_i}{x_j - x_i}\right) \tag{35}$$

and

$$\Delta\theta_{i,j} = \left(\frac{180}{\pi}\right) 2\tan^{-1}\left(\frac{0.5l_0}{\sqrt{\left|\vec{r}_j - \vec{r}_i\right|^2 - (0.5l_0)^2}}\right), \tag{36}$$

where $\theta_{i,j}$ is the relative angle between the $i$-th and $j$-th particles, and $\Delta\theta_{i,j}$ is the subtend angle of the shaded area of the $j$-th neighboring particle (Fig. 3). The coefficient, $\left(\frac{180}{\pi}\right)$, converts radians to degrees. When implementing the program, we divide the virtual screen into small segments of $360°/N_\Theta$,where $N_\Theta$ is the number of small segments, and use a one-dimensional array, $f_k$ $(k = 1, 2, 3, \ldots, N_\Theta)$, to calculate the illuminated area. In this study, $N_\Theta$ was 360. Therefore, each array index $k$ means the relative direction in degrees, and each array component $f_k$ has information on whether the segment of the virtual screen is illuminated in that direction. First, all the array values are initially set to 1, where 1 means the segment of the screen is illuminated by the virtual light source. If the $k$-th segment is in the shadow of the neighboring particles, $f_k$ is set to zero. Finally, the illuminated fraction of the area is calculated from:

$$R_{illuminated} = \frac{A_{illuminated}}{A_{whole}} = \frac{\Theta\, r_e}{2\pi r_e} = \frac{\Theta}{2\pi} \cong \frac{\sum_{k=1}^{N_\Theta} f_k}{N_\Theta}, \tag{37}$$

where $A_{whole}$, and $A_{illuminated}$ are the whole and illuminated areas of the virtual screen.

### *3.4.2 Three-dimensional simulation*

We extend the above surface detection method to three-dimensional simulations. The procedure is almost the same as the two-dimensional one. Figure 4 shows the surface detection algorithm in three dimensions. A virtual spherical screen whose radius is $2.1l_0$ is located on the $i$-th particle. The screen is divided into small segments in the longitudinal and latitudinal directions whose subtended angles are $C_\phi$ $\left(= 360°/N_\phi\right)$ and $C_\theta$ $(= 180°/N_\theta)$, respectively. In this study, $N_\phi$ was 36, and $N_\theta$ was 18. We have determined the segment size empirically by considering the computational cost. In implementing the program, we use a two-dimensional array, $f_{k,l}$ $\left(k = 1, 2, 3, \ldots, N_\phi;\ l = 1, 2, 3, \ldots, N_\theta\right)$, to calculate the shaded area, where $k$ and $l$ are the longitude and latitude divided by $C_\phi$ and $C_\theta$, respectively. In the same manner as in the two-dimensional simulation, $f_{k,l}$ is initialized and a flag is set so that it expresses whether each segment is illuminated; first, all the array values of $f_{k,l}$ are initially set to 1, where 1 indicates the segment of the screen is illuminated by the virtual light source. If the $k, l$-th segment is in the shade of neighboring particles, $f_{k,l}$ is set to zero. The shaded area of the $j$-th neighboring particle is calculated from

$$\theta_{i,j} = \left(\frac{180}{\pi}\right) \tan^{-1}\left(\frac{y_j - y_i}{\sqrt{\left(x_j - x_i\right)^2 + \left(z_j - z_i\right)^2}}\right), \tag{38}$$

$$\phi_{i,j} = \left(\frac{180}{\pi}\right) \tan^{-1}\left(\frac{x_j - x_i}{z_j - z_i}\right), \tag{39}$$

$$\Delta\theta_{i,j} = \left(\frac{180}{\pi}\right) 2\tan^{-1}\left(\frac{0.5l_0}{\sqrt{\left|\vec{r}_j - \vec{r}_i\right|^2 - (0.5l_0)^2}}\right), \tag{40}$$

and

$$\Delta\phi_{i,j} = \left(\frac{180}{\pi}\right) 2\tan^{-1}\left(\frac{0.5l_0}{\sqrt{\left|\vec{r}_j - \vec{r}_i\right|^2 - (0.5l_0)^2}}\right), \tag{41}$$

where $\phi_{i,j}$ and $\theta_{i,j}$ are the longitudinal and latitudinal directions of the shaded area of the $j$-th neighbor particle, respectively. The parameters $\Delta\phi_{i,j}$ and $\Delta\theta_{i,j}$ are the longitudinal and latitudinal angles of the shaded area of the neighbor particles as shown in Fig. 4. In the above calculation, each shaded area of a neighboring particle is approximated as a quadrilateral to simplify the problem, although its actual shadow is circular.

Finally, the illuminated fraction of the area,$R_{illuminated}$, is calculated using

$$R_{illuminated} = \frac{S_{illuminated}}{S_{whole}} \cong \frac{\sum_{l=1}^{N_\theta}\sum_{k=1}^{N_\phi}\left[\left\{r_e\left(\frac{\pi}{180}C_\theta\right)\right\}\left\{r_e\cos\left(\frac{\pi}{180}C_\theta l\right)\right\}\left(\frac{\pi}{180}C_\phi\right)f_{k,l}\right]}{\sum_{l=1}^{N_\theta}\sum_{k=1}^{N_\phi}\left[\left\{r_e\left(\frac{\pi}{180}C_\theta\right)\right\}\left\{r_e\cos\left(\frac{\pi}{180}C_\theta l\right)\right\}\left(\frac{\pi}{180}C_\phi\right)\right]}. \tag{42}$$

If $R_{illuminated}$ is over a preset threshold, the $i$-th particle is regarded as a surface particle; if not, the particle is regarded as an internal particle. We have empirically determined the threshold to be 1/5.

### 3.5 Low-pressure simulation

In the usual MPS simulations using the standard pressure calculation, pressures lower than that of the free-surface particles are reset to zero to avoid unnaturally low pressure around free surfaces and to produce stable simulations. The main reason for the unnaturally low pressure is that there are no particles over free surfaces, and the particle number density around a free surface becomes lower than the standard value, $n_0$. Therefore, it is not possible to simulate pressures below that for the free surfaces using the standard pressure calculation of the MPS method. The improved boundary condition developed in this study allows us to easily simulate such pressures because the right-hand side of pressure Poisson equation, Eq. (11), is improved using $\tilde{n}_i^k$, and the unnaturally low pressures around the free surfaces are suppressed. As a matter of course, we do not reset such pressures to zero in low-pressure simulations using our improved pressure calculation.

However, we expect the kinematic energy of the fluid to increase in some low-pressure simulations because of particle oscillations caused by low pressure. This energy increase leads to unstable simulations. To solve this problem, we added an artificial viscosity based on particle collisions [21]. Moreover, if we do not need to simulate pressures below that of the virtual particles, we reset such pressure values to zero to suppress pressure oscillations.

## 4 Verification analysis

### 4.1 Hydrostatic pressure problem

#### *4.1.1 Simulation condition*

We verified the improved pressure calculation in a hydrostatic pressure problem. Figure 5 shows the simulation domain. The particle distribution and pressure distribution of the improved pressure calculation were compared with those of the standard pressure calculation. The simulated pressure at the bottom center $p_1$ also was compared with the analytical solution.

We set the kinematic viscosity coefficient to 1.00 × $10^{-6}\mathrm{m}^2/\mathrm{s}$, the fluid mass density to $1000\,\mathrm{kg/m}^3$, and the

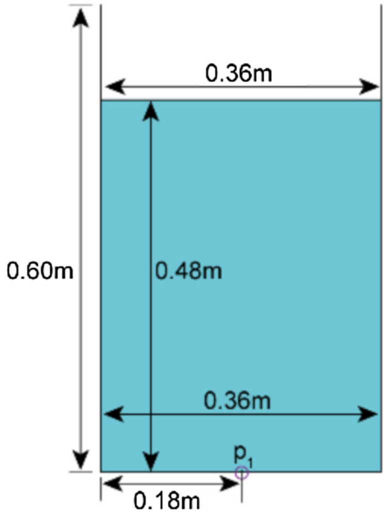


**Fig. 5** Simulation domain for verifying air pressure effect on particle distribution on free surface

acceleration of gravity to 10.0 m/s$^2$. The compressibility of fluid was 0.45 × $10^{-9}$ 1/Pa. The simulations were performed in two spatial resolutions, $l_0 = 0.012$ m and 0.006 m in two dimensions and three dimensions. Table 1 shows the number of particles. Compared with the standard pressure calculation, the number of particles for the improved pressure calculation was small because each wall boundary was expressed by a single layer of wall particles.

The gas-phase pressure, which means the pressure of virtual particles over free surface, was 0 Pa. The radius of the interaction zone for the gradient model, the Laplacian model and the particle number density was $2.0l_0$. The collision distance in the MPS method was $0.5l_0$. The coefficient of restitution was 0.2. The surface tension was neglected. We did not take pressures lower than the gas-phase pressure into account in this case because such pressures should not occur in this hydrostatic pressure problem; pressures below gas phase pressure were set to the gas phase pressure.

### *4.1.2 Simulation results*

The simulated fluid behavior and pressure distribution (Figs. 6, 7) indicate that the particle distribution of the improved pressure calculation (Fig. 7b) was more uniform than that of the standard pressure calculation (Fig. 6b), and the pressure distribution of the improved pressure calculation (Fig. 7c) had a more natural pressure gradient than that of the standard pressure calculation (Fig. 6c); for the improved pressure calculation, the pressure increased with water depth and was constant in the horizontal direction. This is due to the new surface detection method, the corrections for the vir-

**Table 1** Number of particles for the hydrostatic pressure problem

| Number of spatial dimensions<br>Spatial resolution ($l_0$) | 2<br>0.012 m | 2<br>0.006 m | 3<br>0.012 m | 3<br>0.006 m |
|---|---|---|---|---|
| Number of particles for the standard pressure calculation | 1768 | 5872 | 26,928 | 127,040 |
| Number of particles for the improved pressure calculation | 1332 | 5062 | 11,256 | 73,268 |

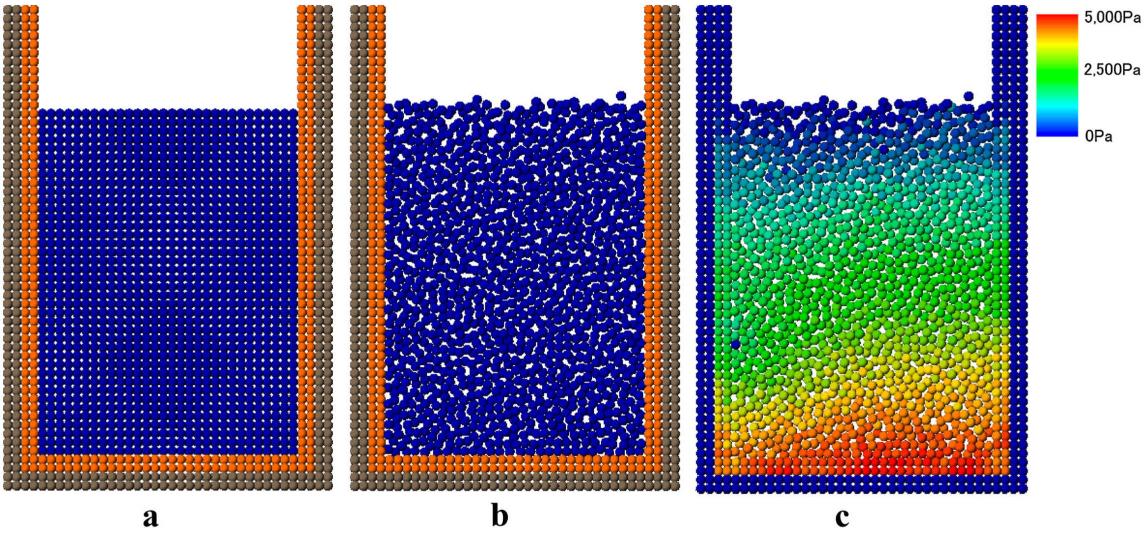


**Fig. 6** Fluid behavior and pressure distribution using the standard pressure calculation. **a** $t = 0.0$ s; **b** $t = 10.0$ s; **c** $t = 10.0$ s (pressure distribution)

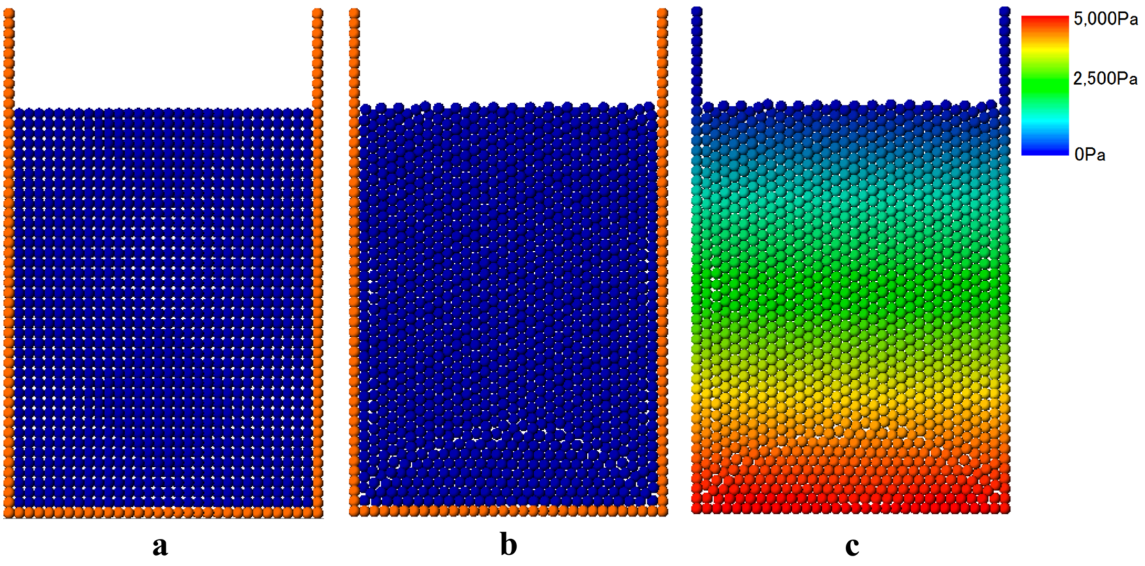


**Fig. 7** Fluid behavior and pressure distribution using the improved pressure calculation. **a** $t = 0.0$ s; **b** $t = 10.0$ s; **c** $t = 10.0$ s (pressure distribution)

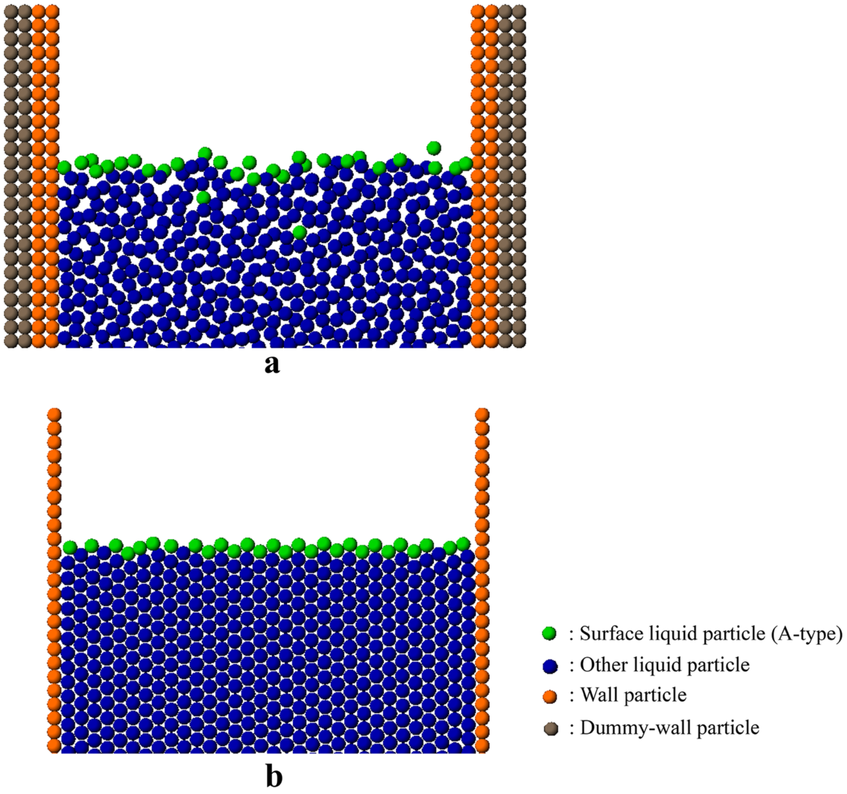

a
b
● : Surface liquid particle (A-type)
● : Other liquid particle
● : Wall particle
● : Dummy-wall particle

**Fig. 8** Particle distribution around surface ($l_0 = 0.012$ m). **a** Standard pressure calculation; **b** Improved pressure calculation

tual particles, and the improvement to the pressure gradient model.

An example of the particle distribution around the surface (Fig. 8) shows that particle clustering does not appear for the improved pressure calculation, whereas some particles on the free surface were too close to each other in the standard pressure calculation.

Table 2 shows the averaged minimum distance between particles in the hydrostatic pressure problem. In the table, "internal-internal" signifies the average distance between internal particles, whereas "surface-all" signifies the distance between a particle on the free surface and all the other particles. The averaged minimum distance of the improved pressure calculation was longer than that of the standard

**Table 2** Averaged minimum distance between particles

| Spatial resolution ($l_0$)<br>Type of distance | 0.012 m<br>Surface-all | 0.012 m<br>Internal-internal | 0.006 m<br>Surface-all | 0.006 m<br>Internal-internal |
|---|---|---|---|---|
| (a) Two-dimensional hydrostatic pressure problem | | | | |
| Standard pressure calculation | $0.74\,l_0$ | $0.88\,l_0$ | $0.76\,l_0$ | $0.88\,l_0$ |
| Improved pressure calculation | $1.00\,l_0$ | $1.05\,l_0$ | $0.96\,l_0$ | $1.06\,l_0$ |
| (b) Three-dimensional hydrostatic pressure problem | | | | |
| Standard pressure calculation | $0.71\,l_0$ | $0.86\,l_0$ | $0.72\,l_0$ | $0.85\,l_0$ |
| Improved pressure calculation | $0.97\,l_0$ | $1.04\,l_0$ | $0.96\,l_0$ | $1.07\,l_0$ |

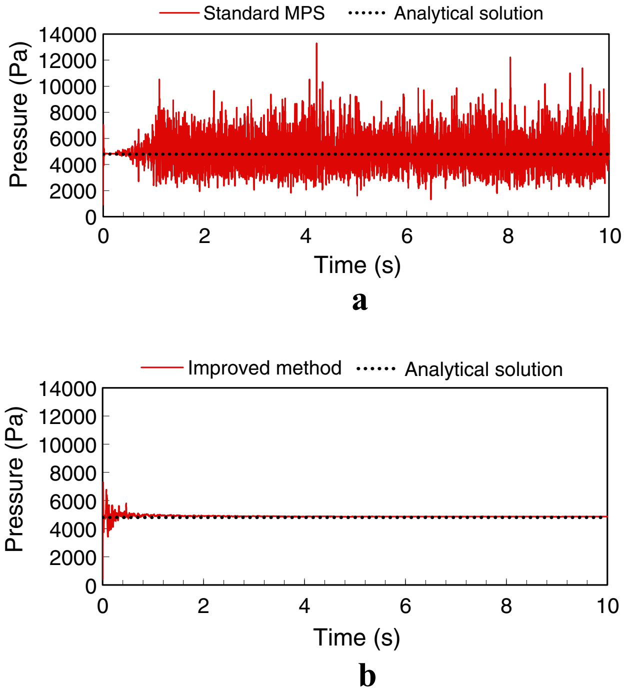


**Fig. 9** Time history of pressure acting on the center of the tank bottom. **a** Standard pressure calculation ($l_0 = 0.012$ m); **b** Improved pressure calculation ($l_0 = 0.012$ m)

pressure calculation and it was close to the initial spacing between particles $l_0$. The table quantitatively confirms that the improved pressure calculation suppresses particle clustering around free surfaces.

The time history of the simulated pressure (Fig. 9) at the center of the tank bottom, $p_1$ in Fig. 5, shows that the pressure fluctuation is suppressed by the improved pressure calculation. Table 3 shows the error of the time-averaged pressure at $p_1$. The error is the difference between the simulation results and the analytical solution

$$P = \rho g h. \tag{43}$$

These data were obtained by time averaging from 8 s to 10 s. From the table, we find that the error in the result using the improved pressure calculation was smaller than that for the result using the standard pressure calculation.

Table 4 shows the computation time. The computer used in this study was equipped with the CPU: Intel Core i7-3770 3.4GHz and the main memory:32 GB. A single core was used without parallel computing. From the table, it is found that the computation time of the improved pressure was almost the same as that of the standard pressure calculation if the same radius for the interaction zone was used. This is because the single layer wall model used in the improved pressure calculation reduces the number of wall particles while the new surface detection method slightly increases the computation time. In total, there was no large difference in computation times. Compared with the standard pressure calculation, the improved pressure calculation appears to enable use of a small radius for the interaction zone because pressure oscillations are suppressed. In cases needing use of a large radius for the interaction zone, computation times using the improved pressure technique were shorter than those using standard pressure calculations.

**Table 3** Error for time-averaged pressure at the tank bottom for the hydrostatic pressure problem

| Number of spatial dimensions<br>Spatial resolution ($l_0$) | 2<br>0.012 m | 2<br>0.006 m | 3<br>0.012 m | 3<br>0.006 m |
|---|---|---|---|---|
| Error for the standard pressure calculation (%) | 3.9 | 4.0 | 2.9 | 2.5 |
| Error for the improved pressure calculation (%) | 1.3 | 1.6 | 1.9 | 1.2 |

**Table 4** Computation times for the hydrostatic pressure problem

| Number of spatial dimensions | 2 | 2 | 3 | 3 |
|---|---|---|---|---|
| Spatial resolution ($l_0$) | 0.012 m | 0.006 m | 0.012 m | 0.006 m |
| The number of time steps | 10,000 | 20,000 | 10,000 | 20,000 |
| Computation time for the standard pressure calculation ($r_{e,gradient} = 2.1l_0$ and $r_{e,Laplacian} = 3.1l_0$) | 221 s | 2,305 s | 14,033 s | 233,868 s |
| Computation time for the standard pressure calculation ($r_{e,gradient} = 2.0l_0$ and $r_{e,Laplacian} = 2.0l_0$) | 133 s | 1,389 s | 5,311 s | 97,767 s |
| Computation time for the improved pressure calculation ($r_{e,gradient} = 2.0l_0$ and $r_{e,Laplacian} = 2.0l_0$) | 134 s | 1,547 s | 4,861 s | 96,676 s |

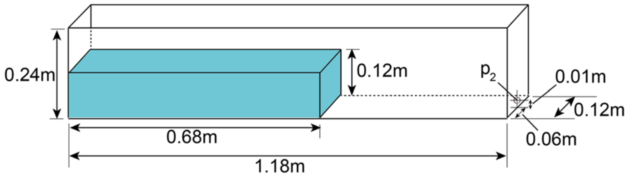


**Fig. 10** Initial water profile for the dam breaking problem

## 4.2 Dam breaking problem

### *4.2.1 Simulation settings*

The stability of the improved pressure calculation was verified in a three-dimensional dam breaking problem. Figure 10 shows the simulation domain. A water column was set in the initial state. The depth from front to back was 0.12 m. We evaluated the simulated pressure at $p_2$. Parameter settings include the initial spacing between particles of $5.0 \times 10^{-3}$ m, the total number of particles of 109,676, a kinematic viscosity coefficient of $1.0 \times 10^{-6}$ $\text{m}^2/\text{s}$ and the acceleration of gravity was $9.8\text{m/s}^2$. The other settings were the same as in Sect. 4.1.1.

### *4.2.2 Simulation results*

Figure 11 shows the simulated fluid behaviors. For visualization, the front wall boundary is not drawn in the figure. We see that the breaking water column and water splashes are well simulated with good numerical stability using the improved pressure calculation. Overall, the fluid behavior of the improved pressure calculation agreed with that of the standard pressure calculation. The detection result of surface particles (Fig. 12) show that the developed technique can detect free-surface particles appropriately. Moreover, the time history of the simulated pressure (Fig. 13) acting on $p_2$ in Fig. 10 confirms that the pressure oscillation are suppressed.

## 4.3 U-shaped tube

### *4.3.1 Simulation condition*

We simulated fluid behavior in a U-shaped tube for further verification of the improved pressure calculation. The simulation domain of the U-shaped tube (Fig. 14) contains.a viscous fluid of mass density 1000 $\text{kg/m}^3$ and kinematic viscosity coefficient $1.00 \times 10^{-4}$ $\text{m}^2/\text{s}$. Uniform virtual particles' pressures, $P_{left}$ and $P_{right}$ were imposed on the left and right fluid surfaces, respectively. Pressure $P_{left}$ was varied from 1000 to 4000 Pa, while $P_{right}$ remained at zero. The initial water levels $h_{left}$ and $h_{right}$ were 0.600 m. The initial spacing between particles, $l_0$, was 0.005 m. The total number of particles was 15,440 and the acceleration of gravity was $9.81\,\text{m/s}^2$. Other settings were the same as in Sect. 4.1.1.

### *4.3.2 Simulation results*

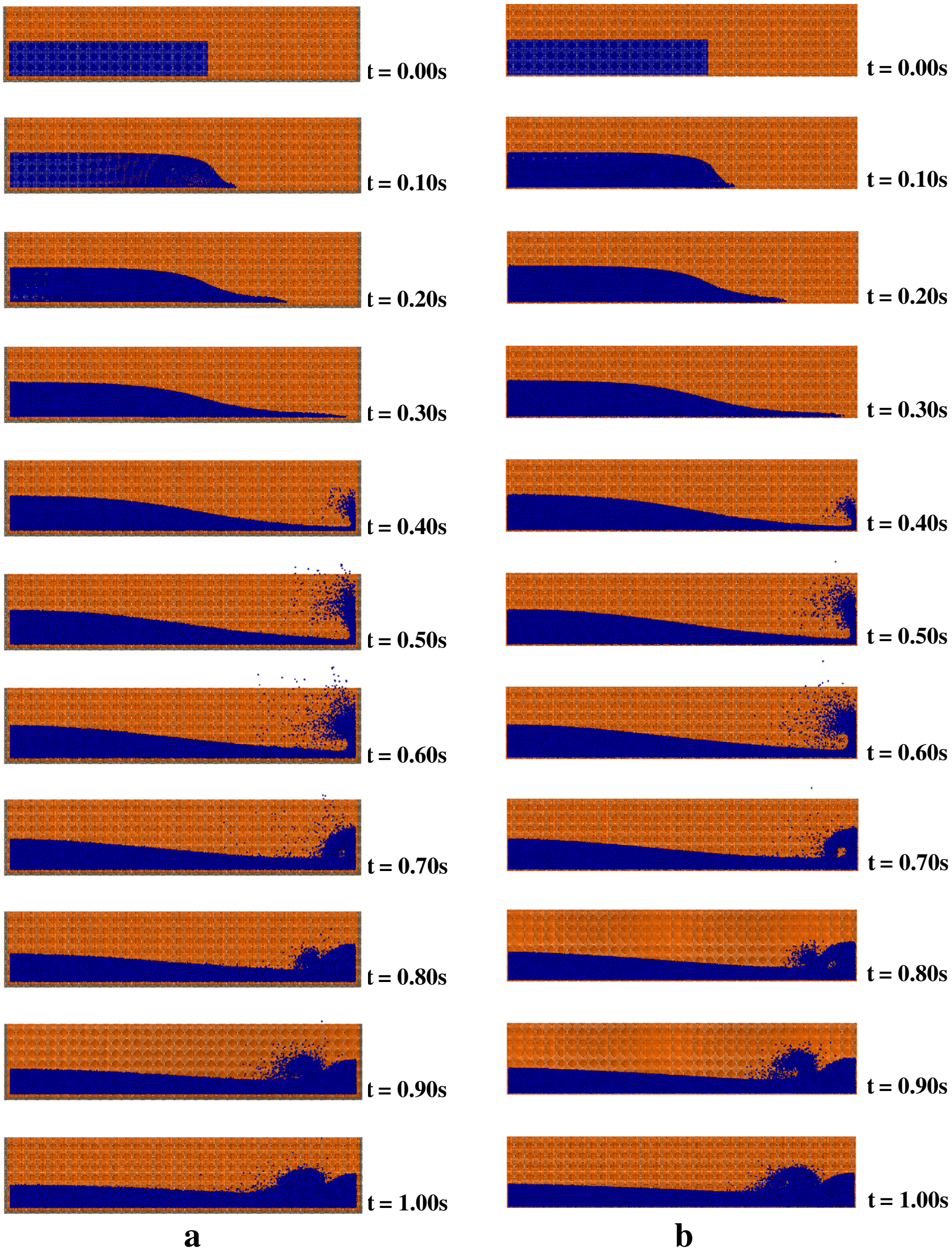
Figure 15 depicts an example of the steady-state fluid behavior where $P_{left}$ and $P_{right}$ were 4000 and 0 Pa, respectively. The water level on the right has risen, while that on the left has fallen because of the difference in gas-phase pressure. The color scale represents the fluid pressure and indicates that the left surface of fluid has a higher pressure than the right.

From the simulations, the relation between the difference in gas-phase pressure and the difference in water level $h$ $(= h_{right} - h_{left})$ at the steady state (Fig. 16) were compared with the analytical solution (dashed line)

$$h = \frac{P_{left} - P_{right}}{\rho g}. \tag{44}$$

An agreement between the two is evident.

**Fig. 11** Fluid behavior for three-dimensional dam breaking ($l_0 = 5.0 \times 10^{-3}$m). **a** Standard pressure calculation; **b** improved pressure calculation



## 4.4 Low-pressure simulation

### *4.4.1 Simulation condition*

For a simple low-pressure problem, we then applied the improved pressure calculation to the two-dimensional glass-raising problem where there is a lower pressure than that of the free surface. Figure 17 shows the simulation domain in the initial state. An upturned glass filled with water was set in a water tank and raised to some height $h$ above the surface. The glass dimensions were 0.050 m in width $A$, 0.0175 m in thickness $T$ and 0.10 m in height. The water tank dimensions were 0.50 m in width $B$ and 0.20 m in height. The simulation time was 3 s.

If the glass is raised, the water in the glass also rises because of a lower pressure than the free-surface pressure. The simulated pressure $P$ at the center of the glass bottom, $\mathrm{p}_3$, was compared with the analytical solution

$$P = -\rho g h' \tag{45}$$

and

$$h' = \frac{hB}{B - A - 2T}, \tag{46}$$

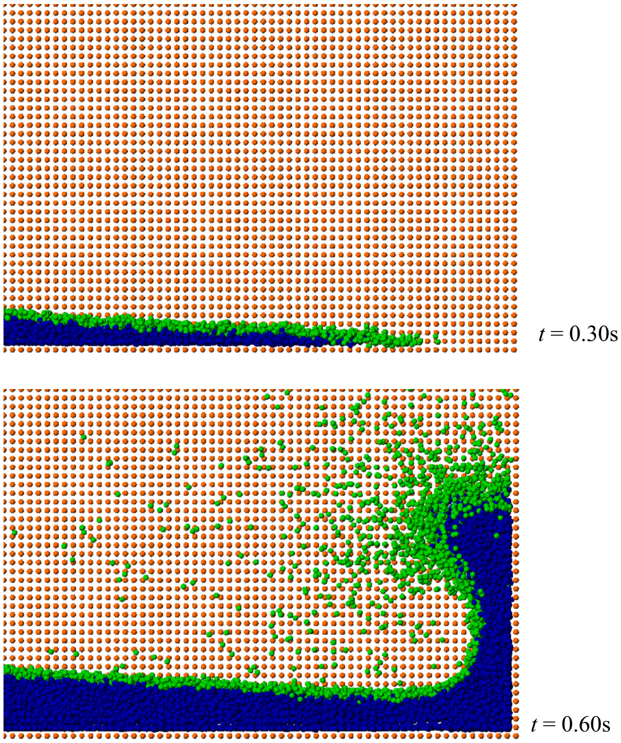


Fig. 12 Detection result of surface particles

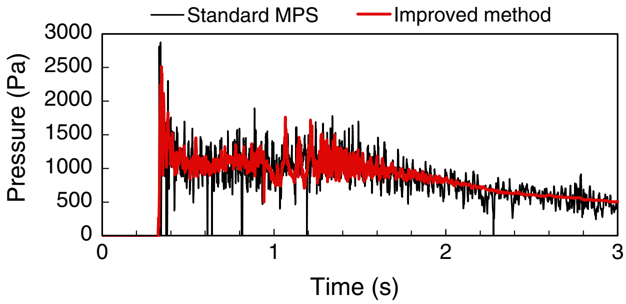


Fig. 13 Time history of pressure in a three-dimensional dam breaking problem

where $h'$ is the relative height of the glass bottom from the water level. The simulations were performed for five raising heights: $h = 0.01, 0.02, 0.03, 0.04$, and $0.05$ m. The glass was raised from its initial position to height $h$ in 1 s at the velocity expressed by a time history of sin curve. The parameter settings include a spatial resolution of 0.0025 m, kinematic viscosity coefficient of $1.0 \times 10^{-6}$ m$^2$/s, acceleration due to gravity of 9.8 m/s$^2$, and total number of particles of 16,502. We took account of pressures below the virtual-particle pressure. The collision distance for the artificial viscosity of the MPS method was $0.85l_0$. The other conditions were the same as in Sect. 4.1.1.

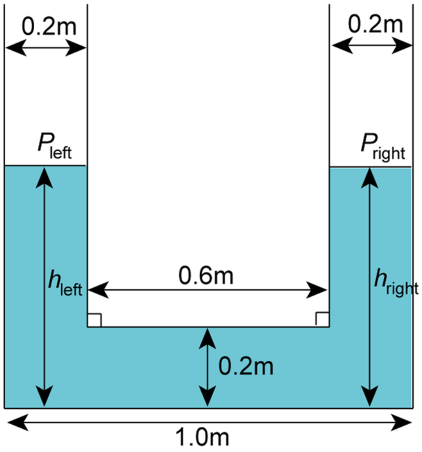


Fig. 14 Simulation domain for verifying air pressure acting on fluid in a U-shaped tube

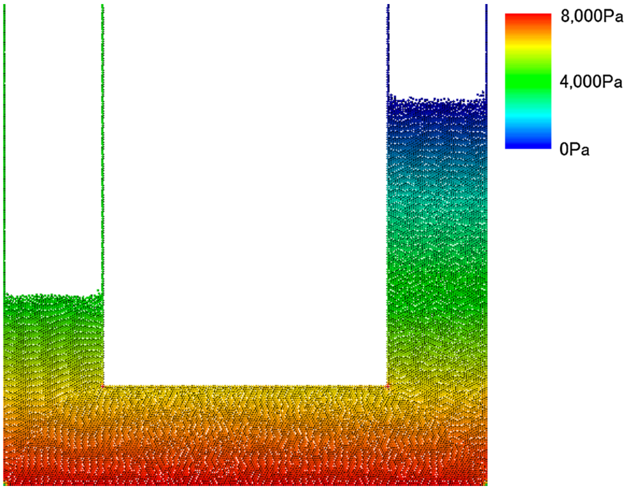


Fig. 15 Simulated fluid shape under steady state in a U-tube ($t$ = 10.0 s). The *left* pressure is 4000 Pa and the *right* pressure 0 Pa. *Color* shows the pressure distribution. (Color figure online)

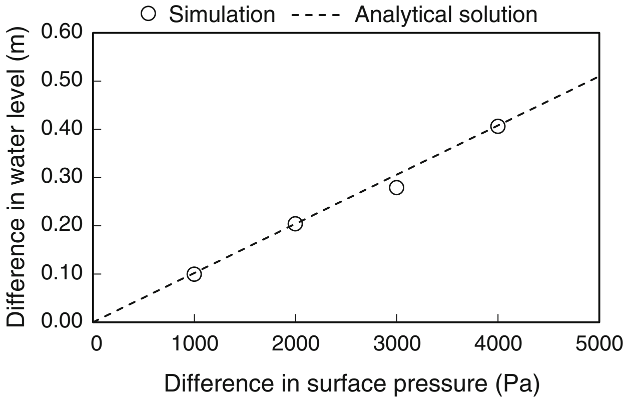


Fig. 16 Relation between pressure difference on free surfaces and water level difference

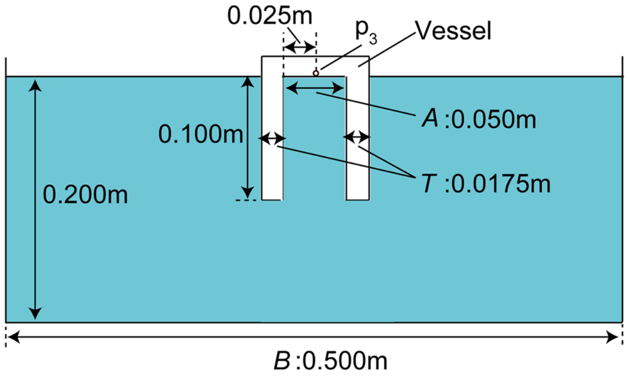


**Fig. 17** Schematic of simulation domain for verifying low pressure

### 4.4.2 Simulation results

Figure 18 shows an example of the simulated fluid behavior and pressure distribution for $h = 0.05$ m. We depict the time-averaged pressure over the last 0.5 s for the particle arrangement at 3 s. The figure shows that low glass pressure was successfully simulated; the minimum pressure occurs at the glass bottom.

Figure 19 shows the relationship between the raising height and the time-averaged pressure obtained at the center of the glass bottom, $p_3$. The pressure was averaged over the last 1.5 s, representing a steady state. It is found that the results of the MPS method using the improved pressure calculation agreed well with the analytical solution.

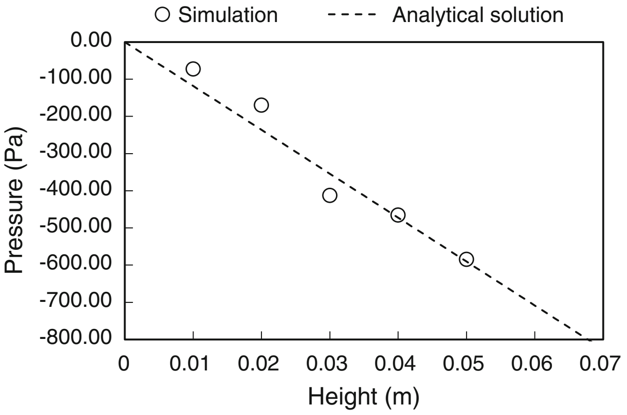


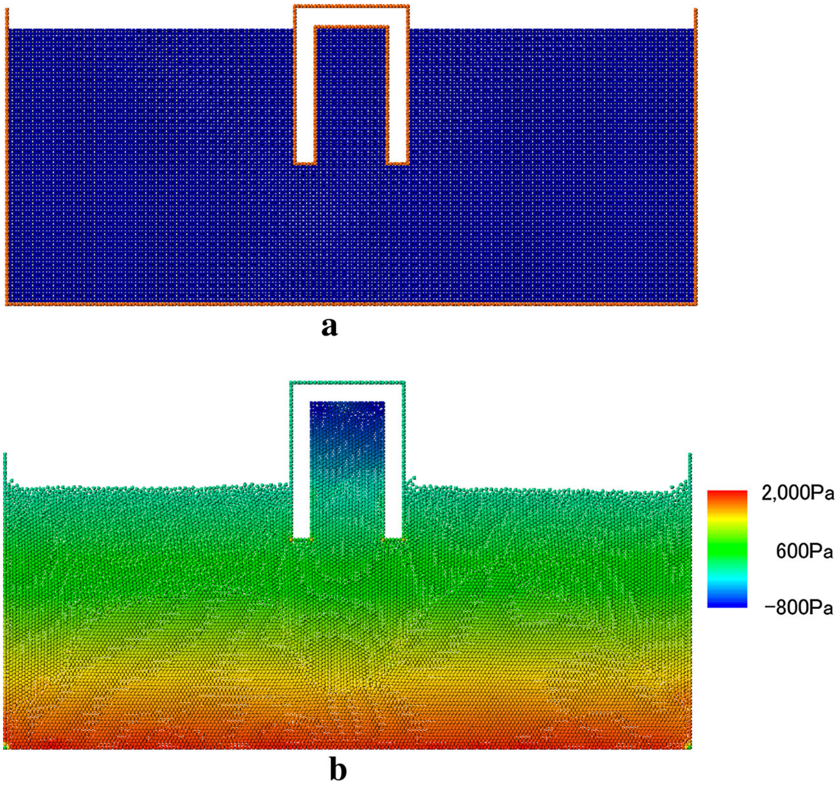
**Fig. 19** Relation between raising height and pressure

## 5 Conclusion

We have improved the pressure calculation of the MPS method by assuming that there are virtual particles over the free surfaces. We derived a modified pressure Poisson equation and a modified approximation model for the pressure gradient. A technique for detecting free-surface particles was also developed for which a virtual light source and screen are used to accurately impose the boundary condition.

We simulated a hydrostatic pressure problem to verify that the improved pressure calculation suppresses particle clustering and pressure oscillation. The stability of the devel-

**Fig. 18** Fluid behavior of low-pressure simulation ($h = 0.05$ m). **a** Initial configuration ($t = 0.0$ s); **b** Pressure distribution ($t = 3.0$ s)



oped technique was verified in a dam breaking simulation. Also the analysis of a U-shaped tube demonstrated that the improved pressure calculation is capable in evaluating the difference in free-surface pressures. Moreover, analyses for low pressure show that the improved pressure calculation is capable in simulating lower pressures than that for the free surface.

**Acknowledgments** This work was supported by the Japan Society for the Promotion of Science (Grants-in-Aid for Scientific Research, Grant No. 25420860).

## References

1. Koshizuka S, Oka Y (1996) Moving-particle semi-implicit method for fragmentation of incompressible fluid. Nucl Sci Eng 123(3):421–434
2. Lucy LB (1977) Numerical approach to testing of fission hypothesis. Astron J 82(12):1013–1024. doi:10.1086/112164
3. Gingold RA, Monaghan JJ (1977) Smoothed particle hydrodynamics—theory and application to non-spherical stars. Mon Not R Astron Soc 181(2):375–389
4. Idelsohn SR, Onate E, Del Pin F (2004) The particle finite element method: a powerful tool to solve incompressible flows with free-surfaces and breaking waves. Int J Numer Methods Eng 61(7):964–989. doi:10.1002/Nme.1096
5. Onate E, Idelsohn SR, Del Pin F, Aubry R (2004) The particle finite element method—an overview. Int J Comput Methods Sing 1(2):267–307. doi:10.1142/S0219876204000204
6. Shibata K, Koshizuka S (2007) Numerical analysis of shipping water impact on a deck using a particle method. Ocean Eng 34(3–4):585–593. doi:10.1016/j.oceaneng.2005.12.012
7. Shibata K, Koshizuka S, Tanizawa K (2009) Three-dimensional numerical analysis of shipping water onto a moving ship using a particle method. J Mar Sci Technol 14(2):214–227. doi:10.1007/s00773-009-0052-7
8. Shibata K, Koshizuka S, Sakai M, Tanizawa K (2012) Lagrangian simulations of ship-wave interactions in rough seas. Ocean Eng 42:13–25. doi:10.1016/j.oceaneng.2012.01.016
9. Shibata K, Koshizuka S, Sakai M, Tanizawa K, Ota S (2013) Numerical Analysis of acceleration of a free-fall lifeboat Using the MPS method. Int J offshore Polar 23(4):279–285
10. Xiong JB, Koshizuka S, Sakai M, Ohshima H (2012) Investigation on droplet impingement erosion during steam generator tube failure accident. Nucl Eng Des 249:132–139. doi:10.1016/j.nucengdes.2011.08.048
11. Arai J, Koshizuka S (2009) Numerical analysis of droplet impingement on pipe inner surface using a particle method. J Power Energy Syst 3(1):228–236. doi:10.1299/jpes.3.228
12. Tsukamoto MM, Cheng LY, Nishimoto K (2011) Analytical and numerical study of the effects of an elastically-linked body on sloshing. Comput Fluids 49(1):1–21. doi:10.1016/j.compfluid.2011.04.008
13. Lee BH, Park JC, Kim MH, Jung SJ, Ryu MC, Kim YS (2010) Numerical simulation of impact loads using a particle method. Ocean Eng 37(2–3):164–173. doi:10.1016/j.oceaneng.2009.12.003
14. Shibata K, Koshizuka S, Sakai M, Tanizawa K (2011) Transparent boundary condition for simulating nonlinear water waves by a particle method. Ocean Eng 38(16):1839–1848. doi:10.1016/j.oceaneng.2011.09.012
15. Kondo M, Koshizuka S (2011) Improvement of stability in moving particle semi-implicit method. Int J Numer Methods Fluids 65(6):638–654. doi:10.1002/Fld.2207
16. Hibi S, Yabushita K (2004) A study on reduction of unusual pressure fluctuation of MPS method. J Kansai Soc Naval archit Jpn 241:125–131
17. Tanaka M, Masunaga T (2010) Stabilization and smoothing of pressure in MPS method by quasi-compressibility. J Comput Phys 229(11):4279–4290. doi:10.1016/j.jcp.2010.02.011
18. Khayyer A, Gotoh H (2009) Modified moving particle semi-implicit methods for the prediction of 2D wave impact pressure. Coast Eng 56(4):419–440. doi:10.1016/j.coastaleng.2008.10.004
19. Khayyer A, Gotoh H (2010) A higher order Laplacian model for enhancement and stabilization of pressure calculation by the MPS method. Appl Ocean Res 32(1):124–131. doi:10.1016/j.apor.2010.01.001
20. Khayyer A, Gotoh H (2011) Enhancement of stability and accuracy of the moving particle semi-implicit method. J Comput Phys 230(8):3093–3118. doi:10.1016/j.jcp.2011.01.009
21. Lee BH, Park JC, Kim MH, Hwang SC (2011) Step-by-step improvement of MPS method in simulating violent free-surface motions and impact-loads. Comput Method Appl Mech Eng 200(9–12):1113–1125. doi:10.1016/j.cma.2010.12.001
22. Tamai T, Koshizuka S (2014) Least squares moving particle semi-implicit method. Comp Part Mech 1(3):277–305. doi:10.1007/s40571-014-0027-2
23. Tanaka M, Masunaga T, Nakagawa Y (2009) Multi-resolution MPS method. Trans Jpn Soc Comput Eng Sci 2009:20090001
24. Suzuki Y, Koshizuka S, Oka Y (2007) Hamiltonian moving-particle semi-implicit (HMPS) method for incompressible fluid flows. Comput Method Appl Mech Eng 196(29–30):2876–2894. doi:10.1016/j.cma.2006.12.006
25. Kajtar J, Monaghan JJ (2008) SPH simulations of swimming linked bodies. J Comput Phys 227(19):8568–8587. doi:10.1016/j.jcp.2008.06.004
26. Kajtar JB, Monaghan JJ (2010) On the dynamics of swimming linked bodies. Eur J Mech B 29(5):377–386. doi:10.1016/j.euromechflu.2010.05.003
27. Khayyer A, Gotoh H, Shao SD (2008) Corrected Incompressible SPH method for accurate water-surface tracking in breaking waves. Coast Eng 55(3):236–250. doi:10.1016/j.coastateng.2007.10.001
28. Khayyer A, Gotoh H, Shao SD (2009) Enhanced predictions of wave impact pressure by improved incompressible SPH methods. Appl Ocean Res 31(2):111–131. doi:10.1016/j.apor.2009.06.003
29. Antuono M, Colagrossi A, Marrone S, Molteni D (2010) Free-surface flows solved by means of SPH schemes with numerical diffusive terms. Comput Phys Commun 181(3):532–549. doi:10.1016/j.cpc.2009.11.002
30. Antuono M, Colagrossi A, Marrone S, Lugni C (2011) Propagation of gravity waves through an SPH scheme with numerical diffusive terms. Comput Phys Commun 182(4):866–877. doi:10.1016/j.cpc.2010.12.012
31. Cleary PW (2010) Extension of SPH to predict feeding, freezing and defect creation in low pressure die casting. Appl Math Model 34(11):3189–3201. doi:10.1016/j.apm.2010.02.012
32. Idelsohn S, Mier-Torrecilla M, Onate E (2009) Multi-fluid flows with the particle finite element method. Comput Method Appl Mech Eng 198(33–36):2750–2767. doi:10.1016/j.cma.2009.04.002
33. Schechter H, Bridson R (2012) Ghost SPH for animating water. Acm Trans Gr 31(4): 61. doi:10.1145/2185520.2185557
34. Koshizuka S (2011) Current achievements and future perspectives on particle simulation technologies for fluid dynamics and heat transfer. J Nucl Sci Technol 48(2):155–168. doi:10.3327/Jnst.48.155

35. Yildiz M, Rook RA, Suleman A (2009) SPH with the multiple boundary tangent method. Int J Numer Meth Eng 77(10):1416–1438. doi:10.1002/Nme.2458
36. Marrone S, Antuono M, Colagrossi A, Colicchio G, Le Touze D, Graziani G (2011) delta-SPH model for simulating violent impact flows. Comput Method Appl Mech Eng 200(13–16):1526–1542. doi:10.1016/j.cma.2010.12.016
37. Akimoto H (2013) Numerical simulation of the flow around a planing body by MPS method. Ocean Eng 64:72–79. doi:10.1016/j.oceaneng.2013.02.015
38. Asai M, Fujimoto K, Tanabe S, Beppu M (2013) Slip and no-slip boundary treatment for particle simulation model with incompatible step-shaped boundaries by using a virtual maker. Trans Jpn Soc Comput Eng Sci 2013:20130011–20130011. doi:10.11421/jsces.2013.20130011
39. Marrone S, Colagrossi A, Antuono M, Colicchio G, Graziani G (2013) An accurate SPH modeling of viscous flows around bodies at low and moderate Reynolds numbers. J Comput Phys 245:456–475. doi:10.1016/j.jcp.2013.03.011
40. Cundall PA, Strack ODL (1979) Discrete numerical-model for granular assemblies. Geotechnique 29(1):47–65
41. Dilts GA (2000) Moving least-squares particle hydrodynamics II: conservation and boundaries. Int J Numer Methods Eng 48(10):1503. doi:10.1002/1097-0207(20000810)48:10<1503:Aid-Nme832>3.0.Co;2-D
42. Idelsohn SR, Onate E, Calvo N, Del Pin F (2003) The meshless finite element method. Int J Numer Methods Eng 58(6):893–912. doi:10.1002/Nme.798
43. Marrone S, Colagrossi A, Le Touze D, Graziani G (2010) Fast free-surface detection and level-set function definition in SPH solvers. J Comput Phys 229(10):3652–3663. doi:10.1016/j.jcp.2010.01.019