

Incompressible material point method for free surface flow [☆]



Fan Zhang ^a, Xiong Zhang ^{a,*}, Kam Yim Sze ^b, Yanping Lian ^{a,c}, Yan Liu ^a

^a School of Aerospace Engineering, Tsinghua University, Beijing 100084, China

^b Department of Mechanical Engineering, The University of Hong Kong, Pokfulam, Hong Kong, China

^c Department of Mechanical Engineering, Northwestern University, Evanston, IL 60208, USA

ARTICLE INFO

Article history:

Received 10 July 2015

Received in revised form 26 June 2016

Accepted 27 October 2016

Available online 11 November 2016

Keywords:

Free surface flow

Operator splitting

Material point method

Hourglass damping

Surface tension

ABSTRACT

To overcome the shortcomings of the weakly compressible material point method (WCMPM) for modeling the free surface flow problems, an incompressible material point method (iMPM) is proposed based on operator splitting technique which splits the solution of momentum equation into two steps. An intermediate velocity field is first obtained by solving the momentum equations ignoring the pressure gradient term, and then the intermediate velocity field is corrected by the pressure term to obtain a divergence-free velocity field. A level set function which represents the signed distance to free surface is used to track the free surface and apply the pressure boundary conditions. Moreover, an hourglass damping is introduced to suppress the spurious velocity modes which are caused by the discretization of the cell center velocity divergence from the grid vertexes velocities when solving pressure Poisson equations. Numerical examples including dam break, oscillation of a cubic liquid drop and a droplet impact into deep pool show that the proposed incompressible material point method is much more accurate and efficient than the weakly compressible material point method in solving free surface flow problems.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

Fluid flow with free surface has been playing an important role in ship hydrodynamics, aerospace, hydraulic engineering, mechanical engineering, petrochemical, civil engineering and so on. As it involves unsteady fragmentation and merging processes that pose some challenges for the existing numerical methods in terms of fluid model, accurate position description of interface, tracking of moving interface, coupling with structure and implement of boundary conditions, which are still focuses in computational fluid dynamics and hydrodynamics.

To address above mentioned issues, researchers had done a lot of work. The PIC (particle-in-cell) method [1] was one of the early attempts to use particles to define the moving boundaries and the finite difference method was used to solve the governing equations. Subsequently, an analogous method, MAC (marker-and-cell) [2] was proposed by Harlow. PIC and MAC have been widely applied in free surface fluid flows though two methods have an inherent defect of expensive computational cost especially for three-dimensional cases and fuzzy interface [3]. Some variants of MAC such as VOF (volume of fluid) [4], GENSMAC [5,6] have also been used widely to solve complex computational fluid dynamic problems including reactive flows, generalized Newtonian flows, multi-material flows, multi-phase flows.

[☆] Supported by the National Natural Science Foundation of China (11272180).

* Corresponding author.

E-mail address: xzhang@tsinghua.edu.cn (X. Zhang).

In the 1980s, a surface tracking technique named Level set method based on dynamic implicit surfaces became extremely popular [7]. The free surface is described by the field of scalar function like signed distance function. The greatest advantage of the level set method is its simplicity, especially for computing the curvature of a moving surface. On the other hand, its drawback is that it needs to solve an additional reinitialization equation besides the conservation equations on Eulerian grid.

Recently, growing interests have been focused on the so-called meshless methods as alternatives of traditional grid-based methods. The Fluid-implicit-particle (FLIP) method based on PIC was proposed by Brackbill [8] to simulate the fluid. Shortly afterwards this method was extended to incompressible fluid flow by Kothe and Brackbill [9]. Nowadays, a hybrid version PIC/FLIP is widely used in physically-based fluid animation [10–12]. The hybrid PIC/FLIP combines the PIC and FLIP to update the particle velocity, and is almost free of numerical dissipation and more stable, but its staggered grid needs complicated data structure and expensive computational cost. Smoothed particle hydrodynamics (SPH) is another popular particle method, whose major advantage over Eulerian methods is the ability to capture very complex interfaces without any special front tracking treatment. Several versions of SPH have been developed for free surface flow such as the weakly compressible form [13], the fully incompressible form based on solving pressure Poisson equations [14,15], and the fully incompressible form based on density prediction [16]. There are also some other meshfree methods applied in free surface flows, such as the moving particle semi-implicit (MPS) [17–19] and the finite point method (FPM) [20,21].

In this contribution, we focus on the material point method (MPM) proposed by Sulsky et al. [22] in 1994, which is an extension of FLIP method from fluid mechanics to solid mechanics. The material point method takes advantages of both Eulerian and Lagrangian methods, so it is promising in handling large deformation problems, such as impact/contact, penetration, cracks and fracture [23–36]. Furthermore, material interfaces can be tracked easily as in Lagrangian method. The MPM has also been applied in fluid mechanics to solve the compressible gas dynamics [37–43] and weakly compressible flow [44–46]. To model fluid-structure interaction (FSI) problems, the MPM has been coupled with other methods, such as finite-volume method [43,47], hybrid immersed boundary method (HIBM) [48], the finite difference method [49], and the finite element method for modeling fluid-structure interaction [45,28,26]. For handling a wider range of material behaviors, an augmented MPM [50] was also presented with dilational/deviatoric splitting of the constitutive model for heat transport, melting and solidifying materials.

For modeling the dynamic behavior of sloshing liquid in a moving container, Li et al. [44] first proposed a weakly compressible material point method (WCMPM) by employing a weakly compressible equation of state (EOS). Though the numerical results are in good agreement with experimental data, and it has also encountered the following issues. (1) The weakly compressible EOS relates pressure to density of the fluid by an artificial sound speed, which is normally taken as 10 times higher than the maximum fluid velocity in order to reduce the density fluctuation down to 1% [51]. Thus the critical time step size of the explicit time integration is very small and the computational cost is considerably high. (2) The material surface is not explicitly tracked in MPM, so it is difficult to accurately impose the pressure boundary conditions on the free surface. (3) The weakly compressible EOS leads to significant pressure oscillations of the fluid particles.

In order to fix the aforementioned issues associated with WCMPM, an incompressible material point method (iMPM) is proposed in this paper based on the operator splitting scheme. The solution of the momentum equation is split into two steps. In the first step, the pressure gradient term is ignored, and the intermediate velocities are obtained by solving the momentum equation whose critical time step size depends only on the speed of shear wave. Generally speaking, the speed of compression wave in a fluid is several orders of magnitude higher than that of the shear wave. Thus, this step can be solved efficiently by using an explicit integration scheme. In the second step, the pressure gradient term is used to correct the intermediate velocities, and the corrected velocities must be divergence free. Applying the divergence free condition on the corrected velocities results in a set of pressure Poisson equations. After solving the Poisson equations, the cell pressures are obtained, and then the grid nodal velocities are updated.

Furthermore, an implicit level set function is adopted to track the free surface, which is helpful to accurately impose pressure boundary conditions induced by surface tension or atmosphere by means of ghost fluid method. In order to suppress the spurious velocity modes associated with the semi-staggered grid employed in the iMPM, an hourglass damping is implemented. Additionally, a calculation scheme for the pressure gradient at grid nodes is presented to stabilize iMPM result.

The remaining part of the paper is organized as follows. The weakly compressible material point method is briefly reviewed in Section 2. Then the new fully incompressible material point method is presented in Section 3. The surface tracking method and surface tension model employed in the iMPM are described in Section 4. The pressure gradient discretization scheme, the spurious velocity modes suppressing scheme and particle velocity update scheme are developed in Section 5. The numerical implementation of iMPM is summarized in Section 6, and several numerical examples are shown in Section 7 to validate the proposed method. Conclusions are drawn in Section 8.

2. Weakly compressible material point method

MPM is a hybrid method with Eulerian–Lagrangian description in which a material domain is represented by a collection of Lagrangian particles moving through a Eulerian background grid, as shown in Fig. 1. The particles carry all state variables such as the position, the velocity, the acceleration, the stress, whereas the grid carries no permanent information. At each time step, the background grid is rigidly attached to the particles and deforms with them so that the grid can be viewed

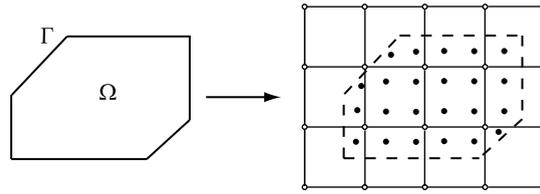


Fig. 1. MPM discrete configuration.

as a finite element discretization to the material domain. The momentum equations are solved on the grid, and the nodal solution is used to update the particles. At the end of each time step, the deformed background grid is reset to its initial state for the next time step.

In the updated Lagrangian framework, the governing equations for the material domain Ω as shown in Fig. 1 consist of

$$\dot{\rho} = 0 \tag{1}$$

$$\rho \ddot{u}_i - \sigma_{ij,j} = \rho b_i \tag{2}$$

$$\begin{cases} (n_j \sigma_{ij})|_{\Gamma_t} = \bar{t}_i \\ v_i|_{\Gamma_v} = \bar{v}_i \end{cases} \tag{3}$$

where ρ is the density of current state, t is the time, σ_{ij} is the Cauchy stress, b_i is the body force per unit mass, u_i is the displacement, the subscripts i and j denote the components of the spatial coordinates, and n_j is the unit outward normal of the boundary. Γ_t and Γ_v stand for the prescribed traction boundary and velocity boundary conditions of Ω , respectively. Taking the virtual velocity δv_i as a test function, the weak form of Eq. (2) reads

$$\int_{\Omega} \rho \ddot{u}_i \delta v_i d\Omega + \int_{\Omega} \sigma_{ij} \delta v_{i,j} d\Omega - \int_{\Omega} \rho b_i \delta v_i d\Omega - \int_{\Gamma_t} \bar{t}_i \delta v_i d\Gamma = 0 \tag{4}$$

In each time step, the background grid is attached rigidly to the particles and deforms with them. Consequently, the velocity v_i can be approximated through

$$v_i = \sum_{l=1}^n N_l v_{il} \tag{5}$$

where the subscript l denotes the background grid node, N_l is the corresponding grid shape function, n is the total number of grid nodes in a background cell, and v_{il} is the velocity of node l in i th component of velocity. Taking the eight-node brick element as an example, the grid shape functions are given by

$$N_l = \frac{1}{8} (1 + \xi \xi_l)(1 + \eta \eta_l)(1 + \zeta \zeta_l), \quad l = 1, 2, \dots, 8 \tag{6}$$

where (ξ_l, η_l, ζ_l) are the natural coordinates of element node l in the reference coordinate system ($\xi \in [-1, 1], \eta \in [-1, 1], \zeta \in [-1, 1]$).

Since the whole domain is discretized by n_p particles, the mass density can be approximated as follows

$$\rho(\mathbf{x}) = \sum_{p=1}^{n_p} m_p \frac{\chi(\mathbf{x} - \mathbf{x}_p)}{V_p} \tag{7}$$

where $\chi(\mathbf{x} - \mathbf{x}_p)$ is the particle characteristic function which defines the space occupied by the particle. According to discretization of weak form Eq. (4), a function \bar{S} used to map information between particles and background grid is given as

$$\bar{S}_{ip} = \frac{1}{V_p} \int_{\Omega_p \cap \Omega_i} \chi(\mathbf{x} - \mathbf{x}_p) N_i(\mathbf{x}) d\mathbf{x} \tag{8}$$

In the original MPM, the Dirac delta function represents the particle characteristic function as follows

$$\chi(\mathbf{x} - \mathbf{x}_p) = \delta(\mathbf{x} - \mathbf{x}_p) V_p \tag{9}$$

This version with Dirac delta function was called the standard MPM. Bardenhagen [52] presented an alternative scheme in which Heaviside function replaces the Dirac delta function

$$\chi(\mathbf{x} - \mathbf{x}_p) = H(\mathbf{x} - (\mathbf{x}_p - l_p)) - H(\mathbf{x} - (\mathbf{x}_p + l_p)) \tag{10}$$

where l_p is half of particle characteristic length associated with the size of sub-domain by the particle p , $H(x)$ stands for Heaviside function. This new scheme was named as the generalized interpolation material point method (GIMP).

Substituting Eq. (5) and Eq. (7) into Eq. (4) yields

$$\dot{p}_{il} = f_{il} \tag{11}$$

where $p_{il} = m_l v_{il}$ is the grid nodal momentum,

$$m_l = \sum_{p=1}^{n_p} \bar{S}_{lp} m_p \tag{12}$$

is the lumped masses,

$$f_{il} = f_{il}^{\text{ext}} + f_{il}^{\text{int}} \tag{13}$$

is the grid nodal force,

$$f_{il}^{\text{int}} = - \sum_{p=1}^{n_p} \bar{S}_{lp,j} \sigma_{ijp} \frac{m_p}{\rho_p} \tag{14}$$

is the internal grid nodal force, and

$$f_{il}^{\text{ext}} = \sum_{p=1}^{n_p} N_{lp} \bar{t}_{ip} h^{-1} \frac{m_p}{\rho_p} + \sum_{p=1}^{n_p} m_p \bar{S}_{lp} b_{ip} \tag{15}$$

is the external grid nodal force, h is the artificial thickness of the boundary layer to calculate the integral along traction boundary.

Eq. (11) can be integrated using a leapfrog scheme to update the velocity and the position of particles as

$$v_{ip}^{n+1/2} = v_{ip}^{n+1/2} + \Delta t^n \sum_{l=1}^{n_g} \frac{f_{il}^n}{m_l^n} \bar{S}_{pl}^n \tag{16}$$

$$x_{ip}^{n+1} = x_{ip}^n + \Delta t^{n+1/2} \sum_{l=1}^{n_g} \frac{p_{il}^{n+1/2}}{m_l^n} \bar{S}_{pl}^n \tag{17}$$

where $p_{il}^{n+1/2} = p_{il}^{n-1/2} + f_{il}^n \Delta t^n$, $t^{n+1} = t^n + \Delta t^{n+1/2}$, $t^{n+1/2} = t^n + \Delta t^{n+1/2}/2 = t^{n-1/2} + \Delta t^n$ and $\Delta t^n = (\Delta t^{n-1/2} + \Delta t^{n+1/2})/2$.

Based on the newly updated grid nodal velocity field, the strain increment of particle is calculated by

$$\Delta \varepsilon_{ijp} = \frac{\Delta t^{n+1/2}}{2} (\bar{S}_{pl,j}^n v_{il}^{n+1/2} + \bar{S}_{pl,i}^n v_{jl}^{n+1/2}) \tag{18}$$

and then a corresponding constitutive law is used to update the particle stress state. For weakly compressible fluid model, an artificial equation of state (EOS) is usually used to update pressure. Monaghan et al. [51] proposed the following EOS

$$p = p_0 ((\rho/\rho_0)^\gamma - 1) \tag{19}$$

where $p_0 = c^2 \rho_0 / \gamma$ with $\gamma = 7$ for water, c is the numerical sound speed which is much lower than the real sound speed of compressible wave to increase the critical time step. Another artificial EOS is proposed by Morris [53] as

$$p = c^2 \rho \tag{20}$$

In this paper, Morris's EOS is implemented in the weakly compressible material point method with the numerical sound speed 50 m/s.

At the end of the time step, the deformed background grid is discarded and a new regular background grid is used in the next time step. Readers may refer to Ma's paper [54] for more details.

3. Incompressible material point method

3.1. Governing equation

Decomposing the stress σ_{ij} into the sum of deviatoric stress s_{ij} and hydrostatic pressure p , i.e. $\sigma_{ij} = -p\delta_{ij} + s_{ij}$, Eq. (2) can be written as

$$\rho \ddot{u}_i = -p_{,i} + s_{ij,j} + \rho b_i \tag{21}$$

For fully incompressible fluid model, the velocity field must satisfy the divergence free condition

$$v_{i,i} = 0 \tag{22}$$

3.2. Operator splitting

Based on the operator splitting method, the computations of velocity and pressure fields can be decoupled. As a result, the integration of Eq. (21) consists of two steps [55,56]. In the first step, the pressure gradient term is ignored in Eq. (21) to compute an intermediate velocity field v_i^* as

$$v_i^* = v_i^n + \frac{\Delta t}{\rho} (s_{ij,j} + \rho b_i), \quad v_i^*|_{\Gamma_v} = \bar{v}_i \tag{23}$$

where v_i^n is the velocity at time t^n . For Newtonian fluid, the deviatoric stress is updated by

$$s_{ij} = 2\mu \dot{\epsilon}'_{ij} \tag{24}$$

where coefficient of dynamic viscosity μ is constant, and $\dot{\epsilon}'$ is the tensor of deviatoric strain rate.

In the second step, the intermediate velocity v_i^* is corrected by the pressure gradient term to yield the final solution v_i^{n+1} as

$$v_i^{n+1} = v_i^* - \frac{\Delta t}{\rho} p_{,i}^{n+1}, \quad v_i^{n+1}|_{\Gamma_v} = \bar{v}_i \tag{25}$$

which must satisfy the divergence free condition, i.e. Eq. (22).

Eq. (23) is almost identical to Eq. (2), so it can be solved via the explicit material point method as presented in Section 2. Thus, the intermediate grid nodal velocity v_{il}^* can be obtained as

$$v_{il}^* = v_{il}^n + \frac{\Delta t}{m_l} (f_{il}^{\text{ext}} + f_{il}^{\text{int},s}) \tag{26}$$

where the external grid nodal force f_{il}^{ext} is given in Eq. (15), and

$$f_{il}^{\text{int},s} = - \sum_{p=1}^{n_p} \bar{S}_{lp,j} s_{ijp} \frac{m_p}{\rho_p} \tag{27}$$

is the internal grid nodal force contributed by deviatoric stress only.

It should be noted that the critical time step size only depends on the speed of shear wave when integrating Eq. (23) explicitly. Since the speed of compression wave in a fluid is usually several orders of magnitude higher than that of the shear wave, the critical time step in iMPM can be much larger than that of WCMPM.

3.3. Pressure Poisson equations

Substituting Eq. (25) into Eq. (22) yields the pressure Poisson equation

$$v_{i,i}^* = \frac{\Delta t}{\rho} \nabla^2 p^{n+1} \tag{28}$$

In WCMPM, all state variables including the pressure are stored at particles. Because the number of particle is much greater than that of cells, it is much more efficient to store the pressure at cell centers rather than at particles, which is equivalent to assuming a piecewise constant pressure field in the grid.

Traditionally, the staggered grid [2] shown in Fig. 2(a) has been employed successfully to solve the Eq. (32), where the pressure p is stored at the cell centers, and the velocities are located at face centers to avoid even-odd pressure modes. However, we applied the semi-staggered grid as shown in Fig. 2(b), since it can be naturally incorporated into MPM. The

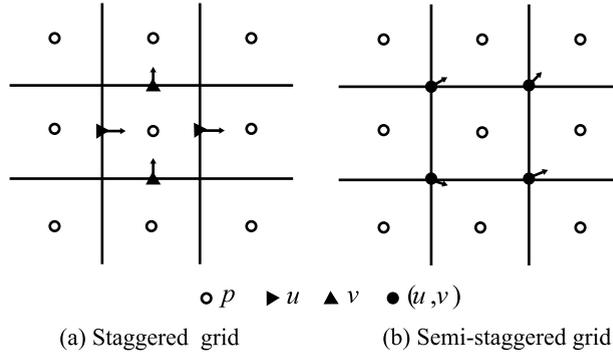


Fig. 2. The staggered grid and semi-staggered grid in two dimension.

pressure is stored at each cell center rather than at particle, and the grid nodal velocities are used to approximate the divergence of the intermediate velocity field v^* at cell center (i, j, k) by

$$v_{i,i}^*(\mathbf{x}_{i,j,k}) = \sum \frac{\partial N_I(\mathbf{x}_{i,j,k})}{\partial x_i} v_{il}^* \tag{29}$$

The second order derivative of pressure p with respect to x can be approximated at the cell center (i, j, k) using the central difference method as

$$\left(\frac{d^2 p}{dx^2}\right)_{i,j,k} = \frac{p_{i+1,j,k} + p_{i-1,j,k} - 2p_{i,j,k}}{\Delta x^2} \tag{30}$$

where Δx is the length of cell in the x -direction, and cubic cells are used with setting of $\Delta x = \Delta y = \Delta z = h$. The Laplacian $\nabla^2 p$ can be approximated at the cell center (i, j, k) using the seven-point stencil finite difference method

$$\left(\nabla^2 p\right)_{i,j,k} = \frac{p_{i+1,j,k} + p_{i-1,j,k} + p_{i,j+1,k} + p_{i,j-1,k} + p_{i,j,k+1} + p_{i,j,k-1} - 6p_{i,j,k}}{h^2} \tag{31}$$

Substituting Eq. (31) and Eq. (29) into Eq. (28) yields a system of linear equations as

$$\mathbf{A} \mathbf{p} = \mathbf{b} \tag{32}$$

where \mathbf{A} is the coefficient matrix, \mathbf{p} is a vector consisting of pressures at all cell centers, \mathbf{b} is a vector consisting of the negative divergence of the intermediate velocity at each cell center. This symmetric positive semi-definite linear systems can be solved efficiently by the preconditioned conjugate gradient (PCG) solver.

4. Surface tracking and surface tension model

4.1. Level set function

Another important issue is to track the topology changes such as the breaking waves, droplet or spray on the free surface. For free surface flow problems, the free boundary can not be determined in advance due to two types of boundary conditions – the kinematic boundary condition and dynamics boundary condition. These characteristics in free surface flow result in numerical difficulties.

In many Lagrangian particle methods, an isocontour of an implicit function is used to represent the fluid surface [57,58]. As shown in Fig. 3(a), each particle can be viewed as a sphere with the center of \mathbf{x}_p and radius of r which at least can cover over a cell. Then the implicit function centered at the particle \mathbf{x}_p is defined as [57]

$$\psi_p(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_p\| - r \tag{33}$$

which equals zero on surface of the sphere represented by particle \mathbf{x}_p . Consequently, the fluid interface is obtained as an assemble of spherical patches, which are shown as solid lines in Fig. 3(a).

To determine the fluid interface implicitly, a signed distance function $\varphi(\mathbf{x})$ is defined over the domain Ω , which is negative inside the fluid domain Ω^- and positive outside the fluid domain Ω^+ , as shown in Fig. 3(b). Then $\varphi(\mathbf{x}) = 0$ represents the fluid interface which is constructed based on the value of signed distance function at each cell center given by

$$\varphi_C = \min_p(\psi_p(\mathbf{x}_C)) \tag{34}$$

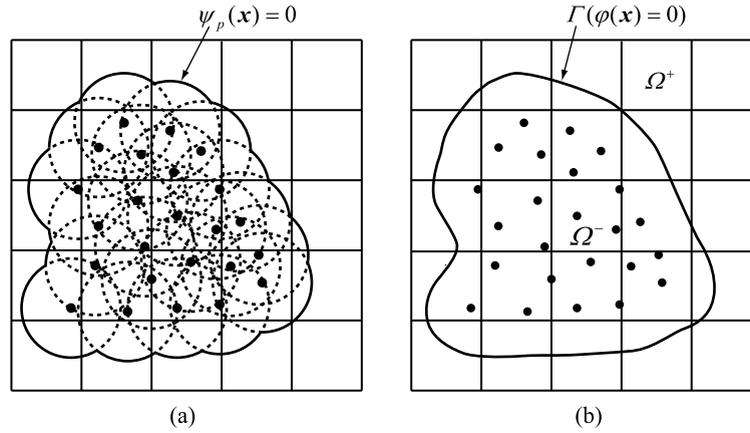


Fig. 3. Isocontour of an implicit function and signed distance function. Solid dots denote fluid particles, Γ denotes material interface, Ω^+ denotes non-fluid domain, and Ω^- denotes fluid domain.

The fluid interface $\varphi(\mathbf{x}) = 0$ constructed from φ_C is not a smooth surface, as shown in Fig. 3(a). To obtain a smooth interface as illustrated in Fig. 3(b), the signed distance function $\varphi(\mathbf{x})$ is smoothed using the fast sweeping method [59] from the initial guess of $\varphi(\mathbf{x})$ defined by Eq. (34).

The interface $\varphi(\mathbf{x}) = 0$ can be advected directly with the motion of fluid particles, in which a new signed distance function is reconstructed in the same way in next time step. This method is very simple, but it requires adequate particles to construct the surface during the computational process. Some non-physical interface may appear if too sparsely particles are used to track the fluid field [57].

The evolution of the free surface $\varphi(\mathbf{x}) = 0$ can be more accurately tracked by solving the level set equation

$$\dot{\varphi} = \varphi_{,t} + \mathbf{v} \cdot \nabla \varphi = 0 \quad (35)$$

Eq. (35) can be solved by using the third order WENO scheme in space and the explicit RK3-TVD in time. To make the values of φ to satisfy properties of the signed distance function, i.e. let $|\nabla \varphi| = 1$, a reinitialization must be conducted by solving

$$\begin{cases} \frac{\partial \varphi}{\partial \tau} = \text{sgn}(\varphi^0)(1 - |\nabla \varphi|) \\ \varphi(\mathbf{x}, 0) = \varphi^0(\mathbf{x}) \end{cases} \quad (36)$$

with forward Euler method in fictitious time τ . The first-order subcell fix scheme [60] is used to discretize the gradient $\nabla \varphi$ in Eq. (36).

4.2. Curvature of the interface

The unit outward normal of the interface $\varphi(\mathbf{x}) = 0$ is obtained from the gradient of the level set function $\varphi(\mathbf{x})$ as

$$\mathbf{n} = \frac{\nabla \varphi}{|\nabla \varphi|} \quad (37)$$

and the mean curvature is calculated by

$$\kappa = \nabla \cdot \mathbf{n} \quad (38)$$

The first-order and second-order derivatives of φ are required to calculate the curvature in Eq. (38). Due to the low accuracy of the signed distance function φ obtained from the low-order reinitialization scheme, the second-order derivatives of φ may be oscillating. To obtain an accurate curvature, an alternative quadratic polynomial function $\tilde{\varphi}(x, y, z)$ is constructed using the local least-squares fitting

$$\tilde{\varphi}(x, y, z) = a_0 + a_1x + a_2y + a_3z + a_4x^2 + a_5y^2 + a_6z^2 + a_7xy + a_8yz + a_9xz \quad (39)$$

A $3 \times 3 \times 3$ stencil with 27 sample points is used to determine the unknown coefficients a_1, a_2, \dots, a_9 . $\tilde{\varphi}$ is then used to calculate the mean curvature by Eq. (38).

4.3. Pressure boundary conditions

When solving the pressure Poisson equation Eq. (28), two kinds of boundary conditions regarding pressure need to be imposed. One is the prescribed pressure on the free surface, the other is normal pressure gradient imposed along the interface between the solid and fluid to guarantee the continuity of normal velocity.

The ghost fluid method [61–63] is used to impose the prescribed pressure boundary condition on the free surface. Take two dimensional pressure Poisson equation as an example. If the free surface is located between the centers of fluid cell (i, j) and air cell $(i + 1, j)$, a ghost pressure $p_{i+1,j}^{Gx}$ is defined at the center of cell $(i + 1, j)$ which leads to

$$\left(\frac{\partial^2 p}{\partial x^2}\right)_{i,j} = \frac{p_{i+1,j}^{Gx} + p_{i-1,j}^f - 2p_{i,j}^f}{\Delta x^2} \tag{40}$$

where

$$p_{i+1,j}^{Gx} = \frac{p^{fs} + (\theta - 1)p_{i,j}^f}{\theta}$$

is the ghost pressure of cell $(i + 1, j)$ in the x -direction, p^{fs} is the surface pressure, and $\theta = |(x_{fs} - x_i)/\Delta x|$. The surface pressure p^{fs} is equal to the air pressure p_{air} or $p_{air} + \sigma_{\kappa}^{fs}$ if the surface tension is taken into account. Substituting $p_{i+1,j}^{Gx}$ into Eq. (40) yields

$$\left(\frac{\partial^2 p}{\partial x^2}\right)_{i,j} = \frac{\frac{1}{\theta}p^{fs} + p_{i-1,j}^f - (1 + \frac{1}{\theta})p_{i,j}^f}{\Delta x^2} \tag{41}$$

4.4. Surface tension model

The surface tension will lead to a pressure jump σ_{κ}^{fs} between the two fluids proportional to the mean curvature, i.e.

$$\sigma_{\kappa}^{fs} = \gamma \kappa \tag{42}$$

where γ is the surface tension coefficient, which is 0.073 N/m for the interface between water and air at room temperature, κ is the mean curvature calculated by Eq. (38). In Eq. (23), the viscous force is added in the term of intermediate velocity, so the pressure boundary condition is modified as follows

$$p^{fs} = p_{air} + \sigma_{\kappa}^{fs} + s_{ij}n_i n_j \tag{43}$$

where $p_{air} = 0$ in this paper, and Eq. (32) will become complicated due to the contribution of deviatoric stress to the boundary pressure at free surface. A decoupled scheme [64] using the constant extrapolation of velocity nearby the free surface is applied in the iMPM code, so that the contribution of deviatoric stress s_{ij} to the surface pressure p^{fs} is eliminated and Eq. (43) can be simplified as $p^{fs} = \sigma_{\kappa}^{fs}$. The extrapolation of velocity benefits for decreasing the complexity of pressure boundary condition [65]. For the most of meshfree methods applied in incompressible fluid, the numerical instability will happen at the free surface, so the constant extrapolation of velocity not only gives a smoother motion of the implicit free surface, but is helpful to improve the numerical stability.

For the solid boundary condition, it is assumed that the solid is static and its boundary is in parallel with the grid line with no cell cutting. Hence, on the interface between solid and fluid, $\nabla p \cdot \mathbf{n} = 0$ is imposed when solving Eq. (28).

5. Velocity update

5.1. Pressure gradient calculation

It should be noted that the pressure gradient ∇p is required to update the velocity v_i^{n+1} using Eq. (25). In the MAC staggered grid, v_i^{n+1} is updated at face centers, so that the pressure gradient ∇p is also evaluated at face centers which can be obtained by the finite difference method as

$$\left(\frac{\partial p}{\partial x}\right)_{i+\frac{1}{2},j} = \frac{p_{i+1,j} - p_{i,j}}{\Delta x}, \quad \left(\frac{\partial p}{\partial y}\right)_{i,j+\frac{1}{2}} = \frac{p_{i,j+1} - p_{i,j}}{\Delta y} \tag{44}$$

At the free surface, $p_{i+1,j}$ and $p_{i,j+1}$ in Eq. (44) should be replaced with the ghost values $p_{i+1,j}^{Gx}$ and $p_{i,j+1}^{Gy}$ respectively to impose the pressure boundary condition using the ghost fluid method if both cells are outside the fluid domain.

Since the semi-staggered grid shown in Fig. 2(b) is used in iMPM, v_i^{n+1} is updated at grid node, so that the pressure gradient should also be evaluated at grid node. If all cells connected to a grid node are fluid cells as illustrated in Fig. 4(a), ∇p can be evaluated at the grid node $(i + \frac{1}{2}, j + \frac{1}{2})$ as

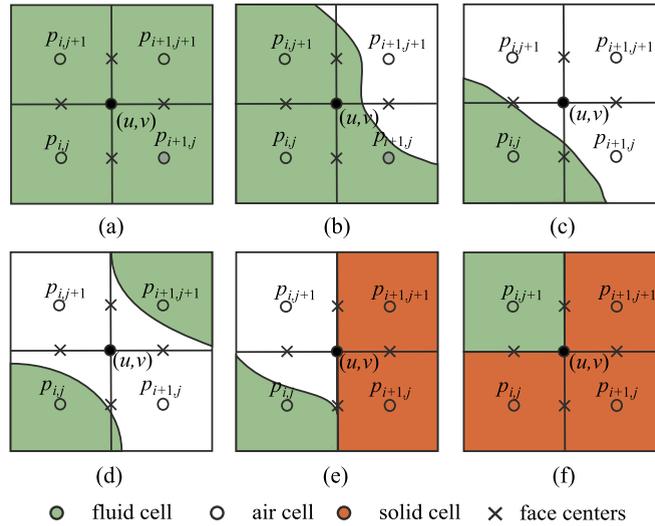


Fig. 4. Pressure gradient calculation.

$$\begin{aligned} \left(\frac{\partial p}{\partial x}\right)_{i+\frac{1}{2},j+\frac{1}{2}} &= \frac{1}{2} \left[\left(\frac{\partial p}{\partial x}\right)_{i+\frac{1}{2},j} + \left(\frac{\partial p}{\partial x}\right)_{i+\frac{1}{2},j+1} \right] \\ \left(\frac{\partial p}{\partial y}\right)_{i+\frac{1}{2},j+\frac{1}{2}} &= \frac{1}{2} \left[\left(\frac{\partial p}{\partial y}\right)_{i,j+\frac{1}{2}} + \left(\frac{\partial p}{\partial y}\right)_{i+1,j+\frac{1}{2}} \right] \end{aligned} \tag{45}$$

where

$$\begin{aligned} \left(\frac{\partial p}{\partial x}\right)_{i+\frac{1}{2},j} &= \frac{p_{i+1,j} - p_{i,j}}{\Delta x}, & \left(\frac{\partial p}{\partial x}\right)_{i+\frac{1}{2},j+1} &= \frac{p_{i+1,j+1} - p_{i,j+1}}{\Delta x} \\ \left(\frac{\partial p}{\partial y}\right)_{i,j+\frac{1}{2}} &= \frac{p_{i,j+1} - p_{i,j}}{\Delta y}, & \left(\frac{\partial p}{\partial y}\right)_{i+1,j+\frac{1}{2}} &= \frac{p_{i+1,j+1} - p_{i+1,j}}{\Delta y} \end{aligned} \tag{46}$$

are the pressure gradient components evaluated at face centers. Note that $(\partial p/\partial x)_{i+\frac{1}{2},j+\frac{1}{2}}$ is the average of $\partial p/\partial x$ at face centers $(i+1/2, j+1)$ and $(i+1/2, j)$, $(\partial p/\partial y)_{i+\frac{1}{2},j+\frac{1}{2}}$ is the average of $\partial p/\partial y$ at face centers $(i, j+1/2)$ and $(i+1, j+1/2)$.

If a grid node is nearby the free surface as illustrated in Fig. 4(b), the pressure $p_{i+1,j+1}$ in Eq. (45) is replaced with a ghost pressure to impose the pressure boundary condition on the free surface. Namely, $(\partial p/\partial x)_{i+1/2,j+1}$ is calculated from the pressure $p_{i,j+1}$ and ghost pressure $p_{i+1,j+1}^{Gx}$, and $(\partial p/\partial y)_{i+1,j+1/2}$ is calculated from the pressure $p_{i+1,j}$ and ghost pressure $p_{i+1,j+1}^{Gy}$. The ghost pressure $p_{i+1,j+1}^{Gx}$ is extrapolated from $p_{i,j+1}$ and the free surface pressure p_{j+1}^{fs} in the x -direction, while $p_{i+1,j+1}^{Gy}$ is extrapolated from $p_{i+1,j}$ and the free surface pressure p_{i+1}^{fs} in the y -direction.

The weights of pressure gradient at face centers in Eq. (45) are all equal to 1/2, which is only valid if at least one adjacent cell center of each face center is located within the fluid domain. Cases (a), (b) and (d) in Fig. 4 satisfy this requirement, but cases (c), (e) and (f) do not. For case (c), both adjacent cell centers of face center $i+1, j+1/2$ are not in the fluid domain, so the weight of $(\partial p/\partial y)_{i+1,j+1/2}$ should be zero. Similarly, the weight of $(\partial p/\partial x)_{i+1/2,j+1}$ should also be zero. Thus, Eq. (45) could be generalized as

$$\left(\frac{\partial p}{\partial x}\right)_{i+\frac{1}{2},j+\frac{1}{2}} = w_{i+\frac{1}{2},j} \left(\frac{\partial p}{\partial x}\right)_{i+\frac{1}{2},j} + w_{i+\frac{1}{2},j+1} \left(\frac{\partial p}{\partial x}\right)_{i+\frac{1}{2},j+1} \tag{47}$$

$$\left(\frac{\partial p}{\partial y}\right)_{i+\frac{1}{2},j+\frac{1}{2}} = w_{i,j+\frac{1}{2}} \left(\frac{\partial p}{\partial y}\right)_{i,j+\frac{1}{2}} + w_{i+1,j+\frac{1}{2}} \left(\frac{\partial p}{\partial y}\right)_{i+1,j+\frac{1}{2}} \tag{48}$$

where

$$\begin{cases} w_{i+\frac{1}{2},j} = \beta_{i+\frac{1}{2},j} / (\beta_{i+\frac{1}{2},j} + \beta_{i+\frac{1}{2},j+1}) \\ w_{i+\frac{1}{2},j+1} = \beta_{i+\frac{1}{2},j+1} / (\beta_{i+\frac{1}{2},j} + \beta_{i+\frac{1}{2},j+1}) \end{cases} \tag{49}$$

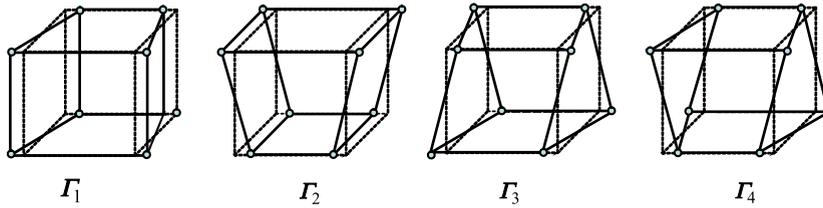


Fig. 5. Hourglass modes.

in which $\beta_{i+\frac{1}{2},j} = 1$ if $\varphi_{i,j} < 0$ or $\varphi_{i+1,j} < 0$, $\beta_{i+\frac{1}{2},j+1} = 1$ if $\varphi_{i,j+1} < 0$ or $\varphi_{i+1,j+1} < 0$. Otherwise, they equal zero. $w_{i+\frac{1}{2},j}$ and $w_{i+1,j+\frac{1}{2}}$ can be obtained in the same way. This technique can be readily extended to three dimensional problems. For example, $(\partial p / \partial x)_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}$ can be evaluated by

$$\begin{aligned} \left(\frac{\partial p}{\partial x}\right)_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} &= w_{i+\frac{1}{2},j+1,k} \left(\frac{\partial p}{\partial x}\right)_{i+\frac{1}{2},j+1,k} + w_{i+\frac{1}{2},j,k} \left(\frac{\partial p}{\partial x}\right)_{i+\frac{1}{2},j,k} \\ &+ w_{i+\frac{1}{2},j,k+1} \left(\frac{\partial p}{\partial x}\right)_{i+\frac{1}{2},j,k+1} + w_{i+\frac{1}{2},j+1,k+1} \left(\frac{\partial p}{\partial x}\right)_{i+\frac{1}{2},j+1,k+1} \end{aligned} \tag{50}$$

where

$$\begin{cases} w_{i+\frac{1}{2},j+1,k} = \beta_{i+\frac{1}{2},j+1,k} / (\beta_{i+\frac{1}{2},j+1,k} + \beta_{i+\frac{1}{2},j,k} + \beta_{i+\frac{1}{2},j,k+1} + \beta_{i+\frac{1}{2},j+1,k+1}) \\ w_{i+\frac{1}{2},j,k} = \beta_{i+\frac{1}{2},j+1,k} / (\beta_{i+\frac{1}{2},j+1,k} + \beta_{i+\frac{1}{2},j,k} + \beta_{i+\frac{1}{2},j,k+1} + \beta_{i+\frac{1}{2},j+1,k+1}) \\ w_{i+\frac{1}{2},j,k+1} = \beta_{i+\frac{1}{2},j+1,k} / (\beta_{i+\frac{1}{2},j+1,k} + \beta_{i+\frac{1}{2},j,k} + \beta_{i+\frac{1}{2},j,k+1} + \beta_{i+\frac{1}{2},j+1,k+1}) \\ w_{i+\frac{1}{2},j+1,k+1} = \beta_{i+\frac{1}{2},j+1,k} / (\beta_{i+\frac{1}{2},j+1,k} + \beta_{i+\frac{1}{2},j,k} + \beta_{i+\frac{1}{2},j,k+1} + \beta_{i+\frac{1}{2},j+1,k+1}) \end{cases} \tag{51}$$

This scheme for calculating pressure gradients at grid nodes is valid for all cases in Fig. 4, including those near the fixed solid boundary.

5.2. Spurious velocity modes

In iMPM, the divergence of velocity is calculated at cell centers when solving the pressure Poisson equations. As a result, the velocity modes illustrated in Fig. 5 that are divergence free at cell centers do not contribute to the pressure Poisson equations. That is to say, these spurious velocity modes are not resisted, and will lead to spurious oscillation in velocity field. These modes are the same as the hourglass modes in hexahedron element with single-point Gauss quadrature in FEM, an hourglass damping scheme is proposed to suppress the spurious velocity modes.

The velocity modes illustrated in Fig. 5 can be expressed as

$$\begin{cases} \Gamma_1 = [1, -1, 1, -1, 1, -1, 1, -1]^T \\ \Gamma_2 = [1, 1, -1, -1, -1, -1, 1, 1]^T \\ \Gamma_3 = [1, -1, -1, 1, 1, -1, -1, 1]^T \\ \Gamma_4 = [-1, 1, -1, 1, 1, -1, 1, -1]^T \end{cases} \tag{52}$$

In three dimensions, there are totally 12 spurious modes, with four modes listed in Eq. (52) for each direction. These spurious modes can be suppressed by introducing a hourglass damping, i.e.

$$\mathbf{v}_i = \mathbf{v}_i - \alpha_h \sum_{k=1}^4 \Gamma_k^T \mathbf{v}_i \Gamma_k \tag{53}$$

where \mathbf{v}_i is the velocity vector consisting of velocity v_i of grid nodes (m, n, p) , $(m + 1, n, p)$, $(m, n + 1, p)$, $(m + 1, n + 1, p)$, $(m, n, p + 1)$, $(m + 1, n, p + 1)$, $(m, n + 1, p + 1)$ and $(m + 1, n + 1, p + 1)$, and α_h is a coefficient, which is chosen as 0.05 in this paper if not specified.

5.3. Particle velocity update

In WCMPM, the particle velocity is updated in Eq. (16) based on the grid nodal acceleration, which is almost free of numerical dissipation. On the contrary, in PIC method, the particle velocity is updated by interpolating the grid nodal velocity, which causes excessive numerical dissipation. To introduce a little dissipation in iMPM scheme, the linear combination scheme [10] is employed, i.e.

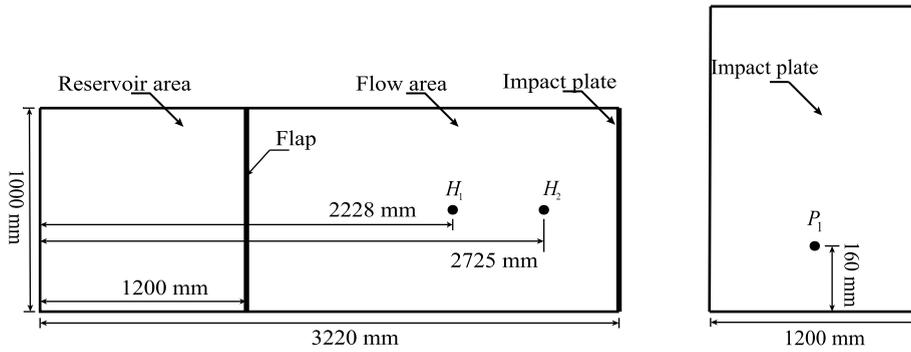


Fig. 6. 3D dam break schematic diagram.

$$v_{ip}^{n+1} = \chi \sum_{l=1}^{n_g} v_{il}^{n+1} \bar{S}_{pl} + (1.0 - \chi)(v_{ip}^n + \Delta v_{ip}) \quad (54)$$

where χ is a coefficient and $0 \leq \chi \leq 1$,

$$\Delta v_{ip} = \sum_{l=1}^{n_g} (v_{il}^{n+1} - v_{il}^n) \bar{S}_{pl} \quad (55)$$

The coefficient χ is taken as 0.03 in this paper.

6. Implementation of incompressible material point method

The proposed iMPM scheme has been implemented in our three-dimensional explicit material point method code MPM3D[®] [66] as a solver, which is suitable for low speed flow problems with free surface. The implementation of iMPM in a time step is briefly summarized as follows.

1. Map the masses and momenta of particles to background grid to initialize the grid nodal masses and momenta.
2. Calculate the intermediate velocities v_i^* using Eq. (23), and impose the velocity boundary conditions.
3. Initialize the signed distance function and identify the fluid cell, the air cell and the free surface either based on the isocontour of particle spherical function or based on a user defined initial grid level set function.
4. Assemble the pressure Poisson equations (32) and solve it via a PCG solver.
5. Update the corrected velocities v_i^{n+1} via Eq. (25).
6. Suppress the hourglass modes of velocities v_i^{n+1} via Eq. (53).
7. Update the particle velocities via Eq. (54).
8. Update the particle positions using RK3-TVD method with the updated grid velocities v_i^{n+1} , and next time step continues from step 1 to step 8 again.

Compared to the WCMPM, the fully incompressible condition is explicitly imposed in the iMPM so that the pressure is solved from the pressure Poisson equations instead of from the local particle strain increment, which avoids the pressure oscillation inherent in WCMPM.

7. Numerical examples

Three numerical examples are studied in this section to validate the proposed incompressible material point method, including the dam break, the oscillation of a cubic liquid drop under surface tension and a water droplet impact into a deep pool. In the following part, we denote WCMPM with Heaviside function as density interpolation by WCGIMP, and iMPM with Heaviside function by iGIMP.

7.1. Dam break

The dam break experiment performed by Zhou et al. [67] and Lobovsky et al. [68] is studied to evaluate the performance of iMPM. A schematic of the test is shown in Fig. 6. At the beginning of the test, water was located in the reservoir area on the left side of the flap with a depth of 0.60 m. After the flap was lifted quickly, the water crashes into the flow area due to the gravity. The tank has a depth h_0 of 0.60 m, width w_0 of 1.0 m and length l_0 of 3.22 m. The density ρ , viscosity μ and gravity g are chosen as 1.0×10^3 kg/m³, 1.01×10^{-3} Pa s and 9.8 m/s², respectively. The water domain is discretized by 5760000 particles, and a background grid consisting of $161 \times 125 \times 50$ cells with cell size of 0.02 m is used.

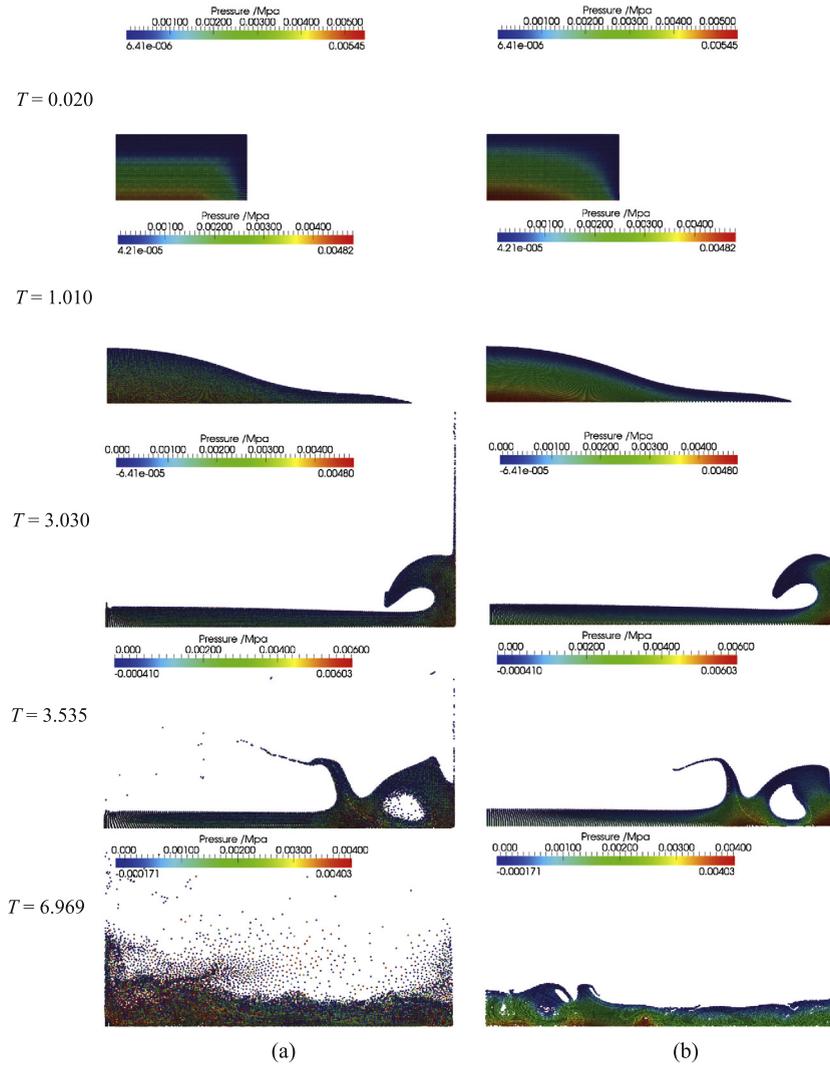


Fig. 7. Dam break configurations obtained by (a) WCMPM and (b) iMPM. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

In this problem, the viscosity is very low, and the free-slip boundary condition is imposed on all solid walls, and surface tension is ignored. The interface is advected directly with the motion of fluid particles, as stated in Section 4.1. For comparison, this example is also simulated by using the WCMPM/WCGIMP with the artificial equation of state Eq. (20).

The following non-dimensional parameters are applied to evaluate the results

$$T = t \sqrt{\frac{h_0 g}{l_0^2}}, \quad L(T) = \frac{l(T)}{l_0}, \quad H(T) = \frac{h(T)}{h_0}, \quad P(T) = \frac{p(T)}{\rho g h_0}$$

where t is the time measured since the start of the gate’s vertical motion, l is the location of the wave front, h is the water level at a specified location, and p is the pressure.

A sequence of snapshots obtained by using the WCMPM and iMPM are compared in Fig. 7. Although the free surface profiles obtained by both methods are very similar to each other before the second plunging wave impacts on the right wall, there is a significant difference in terms of the pressure distribution over the fluid domain. At the very beginning, hydrostatic pressure obtained by both methods is close to the linear distribution along the vertical direction. However, due to the weakly compressible equation of state and crossing-cell noise, the pressure obtained by WCMPM soon shows high-frequency oscillations when water flow develops along the deck. Eventually, there are lots of non-physical sprays and splashes in WCMPM results at $T = 6.969$. Although, the WCGIMP can indeed improve the accuracy of solving the particle stress in a limited extent, it has little contribution to eliminate the high-frequency pressure oscillation and non-physical spray over time. Conversely, the pressure distribution obtained by iMPM/iGIMP is smooth and stable during the whole process and the free surface profiles are reasonable.

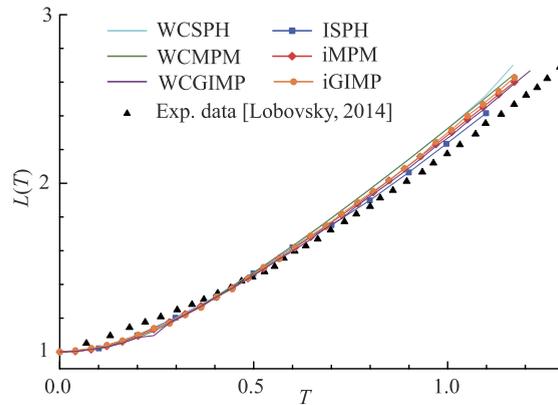


Fig. 8. Time evolution of the water-front toe. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

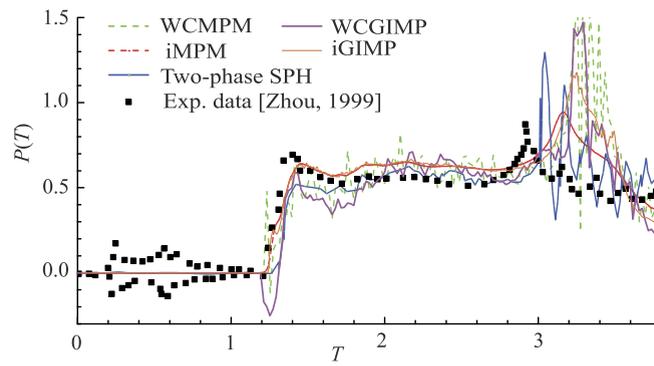


Fig. 9. Time evolution of the pressure at location P_1 . (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

In Fig. 8, the non-dimensional displacements of the wave front from the gate along the dry horizontal bed obtained by WCPM/WCGIMP, iMPM/iGIMP are compared with the experimental data of Lobovsky [68]. Since the hydrostatic pressure plays a dominant role in the low-viscosity flow, the numerical result can be improved by using of the generalized interpolation which benefits in solving of EOS for WCGIMP or divergence calculation in the fluid cell for iGIMP, the results by WCGIMP and iGIMP are a little better than that by WCPM and iMPM, respectively. The difference between experimental data and numerical results may be possibly due to several factors in the experiment, such as the velocity of flap removal and development of turbulence near the water front which causes the delay of moving of downstream. Meanwhile, the numerical results obtained by ISPH [69] and WSPH are also listed in Fig. 8 as reference.

The pressures at P_1 obtained by iMPM/iGIMP, WCPM/WCGIMP and two-phase SPH [70] are compared with the experimental data by Zhou [67] in Fig. 9. The results in iMPM/iGIMP are in good agreement with the experimental data, whereas the WCPM result shows severe oscillation. When the WCGIMP was used to simulate this problem, the high-frequency component of pressure is effectively suppressed, the result from the curve of pressure time-history with WCGIMP is superior to that with WCPM. For iGIMP, its curve of pressure time-history almost coincides with iMPM's curve except in the second peak. The delay of the second pressure peak in the numerical methods is due to the air in the cavity which is not taken into account in the simulations.

Moreover, the water levels at points H_1 and H_2 obtained by WCPM/WCGIMP, iMPM/iGIMP and two-phase SPH are compared in Fig. 10. In the initial stage before the rebounding wave reaches H_1 or H_2 , the numerical results obtained by all these methods agree well with the experimental results. However, at the second stage when a steep increase in height is observed due to the overturning wave, the water levels obtained by iMPM/iGIMP and WCPM/WCGIMP are higher than the experimental results. Although the two-phase SPH had taken the air-cushion effect into account, it still can not give reasonable results in the second stage.

In this example, iMPM not only gives better results, especially in terms of the pressure distribution, than WCPM/WCGIMP, but also is 3.1 times faster than WCPM and is 1.81 times faster than iGIMP in computational efficiency.

7.2. Oscillation of a cubic liquid drop

The formation of a spherical liquid drop from a $0.01 \text{ m} \times 0.01 \text{ m} \times 0.01 \text{ m}$ cubic liquid drop due to surface tension ignoring the gravity is simulated. The density ρ , viscosity μ and surface tension coefficient γ are chosen as $1.0 \times 10^3 \text{ kg/m}^3$,

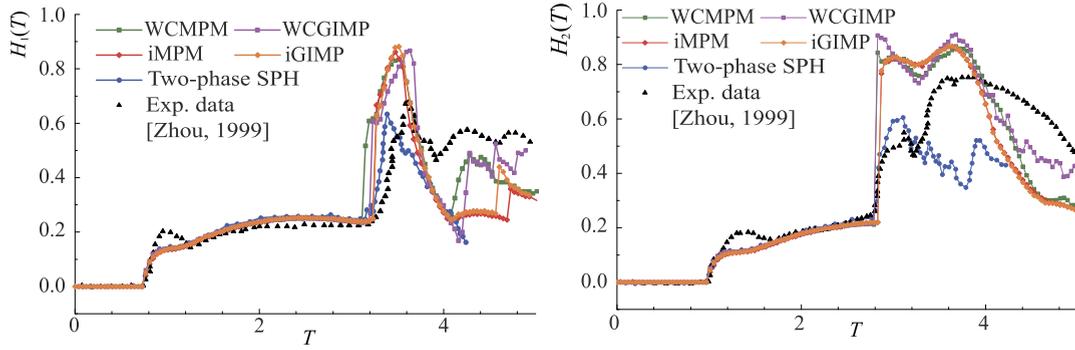


Fig. 10. Time history of the wave heights measured at location H_1 and H_2 . (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

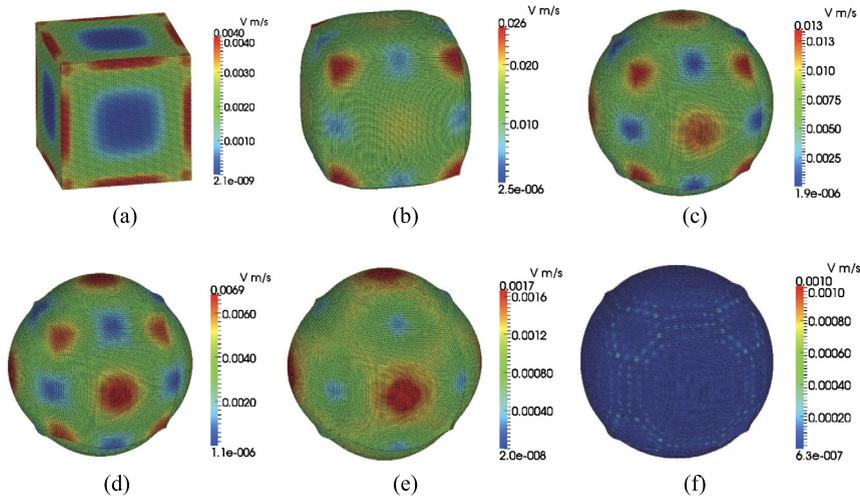


Fig. 11. Configurations of the liquid drop and velocity distribution at time (a) $t = 0.02$ s; (b) $t = 0.06$ s; (c) $t = 0.10$ s; (d) $t = 0.12$ s; (e) $t = 0.16$ s; (f) $t = 1.00$ s. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

0.05 Pa s and 2.4×10^{-3} N/m, respectively. With the given parameters, the drop will become a sphere of radius $R_0 = 6.2035$ mm (or a mean curvature of $\kappa = 2/R_0$) with pressure $p_{\text{drop}} = 0.773756$ Pa at equilibrium.

This test has been investigated by other numerical methods including CSF (continuum surface force) [71], SPH [72], GIMP-CSF [73]. In GIMP-CSF [73], the surface tension is modeled approximately as a body force within an assumed transition zone at the interface. GIMP-CSF is a kind of weakly compressible MPM, and it employed GIMP interpolation scheme to suppress the crossing-cell noise.

Fig. 11 illustrates the snapshots with velocity distribution at different times obtained by iGIMP with cell size of 0.5 mm and $4 \times 4 \times 4$ particles per cell (PPC).

Initially, the cubic droplet is driven by surface tension concentrated at the corners where the mean curvatures are the largest, as shown in Fig. 11(a). After that, the eight round corners of liquid drop continue to contract and non-equilibrium sphere at time $t = 0.1$ s, as shown in Fig. 11(c). Because of liquid viscosity, the droplet shape tends to be sphere at equilibrium state due to the dissipation of kinetic energy, as shown in Fig. 11(f). As illustrated in Fig. 11, several characteristic shapes of the liquid drop during the formation, such as dumbbell, non-equilibrium sphere, diamond, sphere at equilibrium in our method can be observed that are consistent with Zhang’s analysis [72].

Furthermore, six cases, including two different cell sizes 1.0 mm and 0.5 mm with $1 \times 1 \times 1$, $2 \times 2 \times 2$ and $4 \times 4 \times 4$ PPC for each cell size, have been studied by using both iMPM and iGIMP. To investigate the accuracy of iMPM and iGIMP, the non-dimensional average pressure p^* and the root-mean-square relative error L_2 [71,73]

$$p^* = \frac{\frac{1}{N_d} \sum_{i,j,k} p_{i,j,k}}{p_{\text{drop}}}, \quad L_2 = \left[\frac{\sum_{i,j,k} (p_{i,j,k} - p_{\text{drop}})^2}{N_d p_{\text{drop}}^2} \right]^{1/2} \quad (56)$$

are calculated when the droplet reached an equilibrium state at $t = 0.25$ s [73], where N_d is the number of grid cells occupied by the liquid. Table 1 compares p^* and L_2 obtained by iMPM and iGIMP for different cases. The accuracy of both

Table 1
Accuracy comparison between iMPM and iGIMP.

Cell size	PPC	Method	p^*	L_2
1.0	$1 \times 1 \times 1$	iMPM	1.0201	2.54×10^{-2}
		iGIMP	1.0090	1.13×10^{-2}
	$2 \times 2 \times 2$	iMPM	1.0091	1.06×10^{-2}
		iGIMP	1.0012	3.55×10^{-3}
	$4 \times 4 \times 4$	iMPM	1.0037	5.78×10^{-3}
		iGIMP	0.9999	3.20×10^{-3}
0.5	$1 \times 1 \times 1$	iMPM	1.0083	9.51×10^{-3}
		iGIMP	1.0045	5.08×10^{-3}
	$2 \times 2 \times 2$	iMPM	1.0047	5.23×10^{-3}
		iGIMP	1.0023	3.39×10^{-3}
	$4 \times 4 \times 4$	iMPM	1.0032	3.77×10^{-3}
		iGIMP	1.0017	3.21×10^{-3}

Table 2
Parameters used in the droplet impact simulation.

Density ρ (kg/m)	1.1×10^3
Dynamic viscosity μ (Pa s)	3.5×10^{-3}
Surface tension coefficient γ (N/m)	0.067
Droplet diameter R_0 (mm)	4
Impact velocity U_0 (m/s)	2.42
Pool depth h (mm)	10
Pool length and width $L_x \times L_y$ (mm)	40×40
Corresponding free falling height (mm)	300
Weber number We	385
Reynolds number Re	3042
Hourglass damping coefficient α_h	0.01

methods increases with the number of PPC increasing and the cell size decreasing, moreover, the increasing rate is faster from PPC = 1 to PPC = 2 than from PPC = 2 to PPC = 4. The accuracy of iMPM is lower than iGIMP owing to smoother interpolation function used in iGIMP method. Compared with CSF-GIMP [73], iMPM with coarse grid $20 \times 20 \times 20$ gives better results than CSF-GIMP with refined grid $30 \times 30 \times 30$. This shows that the accuracy of iMPM in modeling 3D liquid drop with surface tension is higher than CSF-GIMP. iMPM is also more efficient than CSF-GIMP because the time step is limited by the numerical sound speed in CSF-GIMP but by fluid particle velocity in iMPM.

7.3. Droplet impact into a deep pool

The whole process of a liquid droplet impact onto a liquid surface up to the formation of the Worthington jet is simulated by iMPM/iGIMP with surface tension model. The parameters are chosen based on the experiment by Nishio et al. [74], as listed in Table 2. The fluid used in the experiment was a mixture of water and glycerin.

The background grid used in this simulation is a $40 \text{ mm} \times 40 \text{ mm} \times 40 \text{ mm}$ regular grid with a cell size of 0.4 mm and $2 \times 2 \times 2$ PPC. The bottom of the liquid pool is aligned with the bottom of the grid, on which the non-slip condition is imposed. The top boundary of the grid is unrestricted so that any particles beyond it are removed from the simulation. The droplet and liquid pool are discretized by 2003912 fluid particles.

The experimental snapshots obtained by Nishio et al. [74] are shown in Fig. 12. It is clear that the droplet coalesced into the liquid surface with small splashing spray and a hemispherical crater was growing with a crown-like rim, as shown in Fig. 12(b) and 12(c). The crater depth reached up to the maximum value at about $t = 20$ ms while the crater further extended along the lateral direction, as shown in Fig. 12(d). Owing to the rebound motion concentrated toward the crater center, the crater turned into an inverted conical at $t = 30$ ms, as shown in Fig. 12(e) and continued to rise up to form a central column called 'Worthington jet' as illustrated in Fig. 12(f).

The numerical results obtained by iMPM are shown in Fig. 13. The process of crater and crown formations can be observed in the simulation and experiment results. The numerical crater rebound takes place at $t = 27$ ms, which occurs a little later than that in experiment result. When the Worthington jet reaches to the peak, a new smaller droplet begins to detach from the Worthington jet as shown in Fig. 13(g) and Fig. 13(h) (no experiment data available after $t = 60$ ms). Second droplet impact continues with a lower velocity until the pool water calms.

In order to evaluate the accuracy of iMPM/iGIMP quantitatively, the time history of the crater depth obtained by iMPM/iGIMP is compared with those obtained by SPH and experiment in Fig. 14, where the depth D_C is defined as the depth from the initial liquid surface to the deepest point of the crater, thus it takes a negative value during the crater formation. As shown in Fig. 14, the crater depth obtained by iMPM/iGIMP agrees very well with the experimental data before $t = 15$ ms, but it has stagnated for a longer time when it reached the bottom. The rebound velocity obtained by iMPM/iGIMP

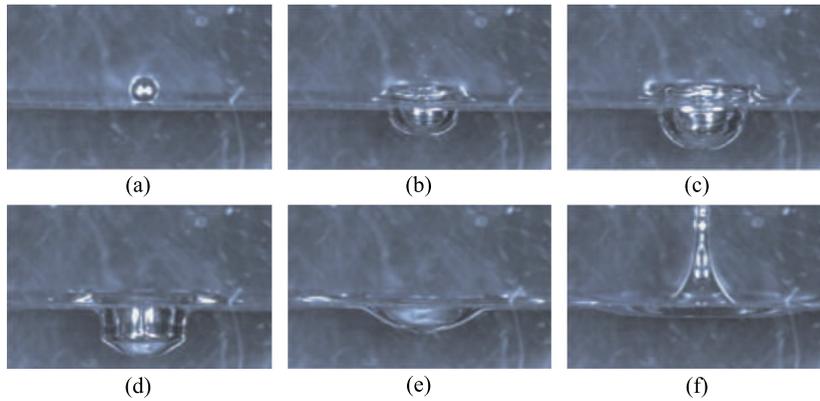


Fig. 12. Experimental snapshots of the droplet impact [74] at time (a) $t = 0$ ms; (b) $t = 5$ ms; (c) $t = 10$ ms; (d) $t = 20$ ms; (e) $t = 30$ ms; (f) $t = 60$ ms.

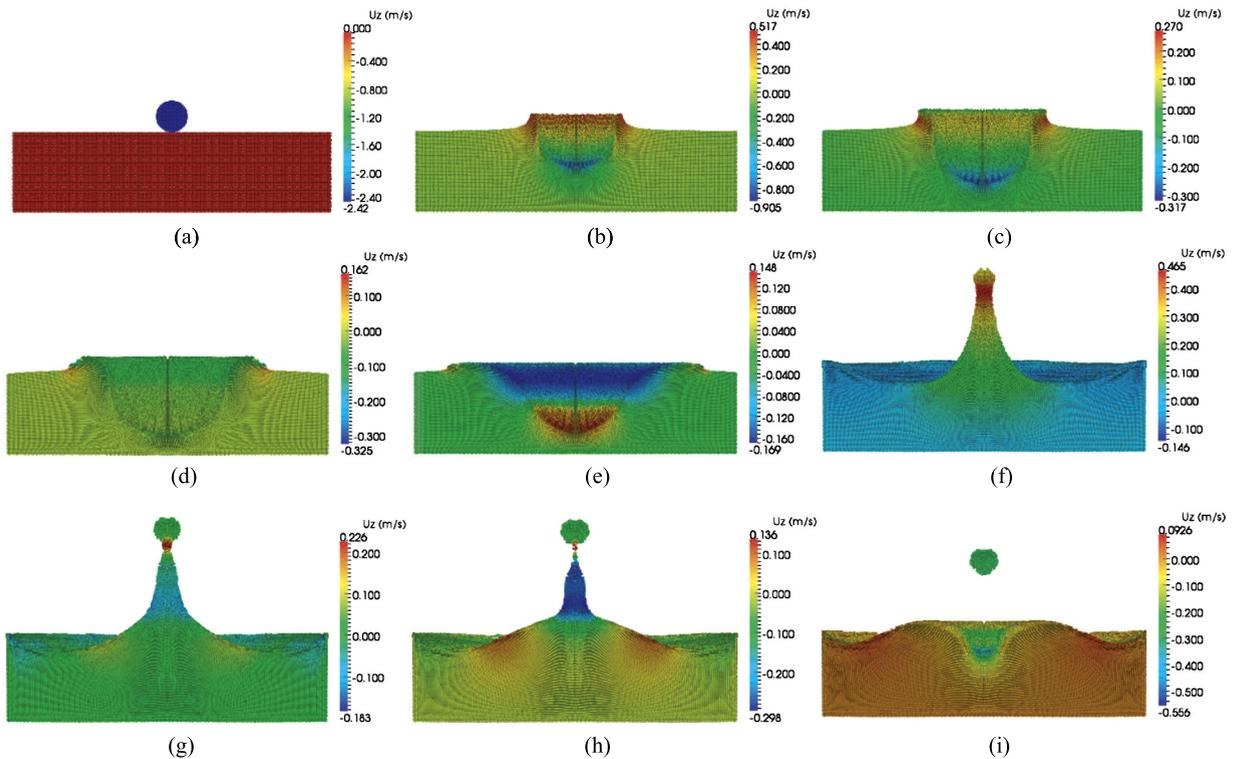


Fig. 13. iMPM results of the droplet impact: (a) $t = 0$ ms; (b) $t = 5$ ms; (c) $t = 10$ ms; (d) $t = 20$ ms; (e) $t = 30$ ms; (f) $t = 60$ ms; (g) $t = 80$ ms; (h) $t = 90$ ms; (i) $t = 110$ ms. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

starts to increase rapidly after $t = 30$ ms, which is faster than that in the experiment. A possible reason is that the initial shape of the droplet in the experiment falling from the height of h_0 to the pool surface is non-spherical, and other factors such as cavitation under the crater is not taken into account in iMPM simulation. Compared to curve from iGIMP as shown in Fig. 14, due to generalized interpolation which extends the support domain of particle in iGIMP, the contraction of crater from iGIMP is ahead of that from iMPM after $t = 25$ ms. Meanwhile, the numerical result by SPH is also shown in Fig. 14 for comparison. By the overall comparison, iMPM/iGIMP can reproduce the complex process of Worthington jet.

8. Discussions and conclusion

In this paper, an incompressible material point method is proposed to model the free surface flow. It distinguishes from the weakly compressible material point method in following aspects: (1) The momentum equation is solved by using the operator splitting technique, in which the pressure is obtained from the pressure Poisson equations that satisfy the incompressible condition. This key improvement avoids numerical oscillation that appeared in WCMPM. Moreover, the critical

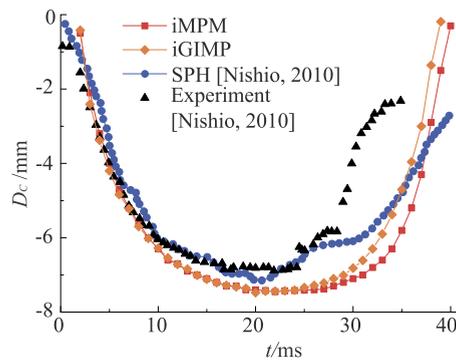


Fig. 14. Comparison of crater depths between simulation and experiment.

time step size in iMPM can be much larger than that in WCMPM method; (2) Level set function is used to track the topology changes of free surface. The pressure boundary condition is imposed more accurately using the ghost fluid method. In addition to that, the surface tension can be included; (3) When updating the grid nodal velocity, a pressure gradient calculation scheme is presented to calculate the pressure gradient at face centers, which is suitable for all cases including inside the fluid, nearby the interface between fluid and air, and interface between fluid and solid; (4) A hourglass damping is employed to suppress the spurious velocity modes appeared in the semi-staggered grid.

Numerical results demonstrate that the proposed incompressible material point method can well simulate the free surface fluid flow, including the dam breaking, the sloshing, the droplet oscillation and the Worthington jet.

References

- [1] F.H. Harlow, The particle-in-cell computing method for fluid dynamics, *Methods Comput. Phys.* 3 (1964) 319–343.
- [2] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface, *Phys. Fluids* 8 (1965) 2182–2189.
- [3] S. McKee, M.F. Tomé, V.G. Ferreira, J.A. Cuminato, A. Castelo, F.S. Sousa, N. Mangiavacchi, The MAC method, *Comput. Fluids* 37 (8) (2008) 907–930.
- [4] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, *J. Comput. Phys.* 39 (1) (1981) 201–225.
- [5] M.F. Tome, S. McKee, GENSMAC: a computational marker and cell method for free surface flows in general domains, *J. Comput. Phys.* 110 (1) (1994) 171–186.
- [6] M.F. Tome, S. McKee, Numerical simulation of viscous flow: buckling of planar jets, *Int. J. Numer. Methods Fluids* 29 (6) (1999) 705–718.
- [7] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, *J. Comput. Phys.* 79 (1) (1988) 12–49.
- [8] J.U. Brackbill, H.M. Ruppel, FLIP: a method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions, *J. Comput. Phys.* 65 (1986) 314–343.
- [9] D. Kothe, J. Brackbill, Flip-inc: a particle-in-cell method for incompressible flows, Unpublished manuscript.
- [10] Y.N. Zhu, R. Bridson, Animating sand as a fluid, *ACM Trans. Graph. (TOG)* 24 (3) (2005) 965–972.
- [11] W. Zheng, B. Zhu, B. Kim, R. Fedkiw, A new incompressibility discretization for a hybrid particle MAC grid representation with surface tension, *J. Comput. Phys.* 280 (2015) 96–142.
- [12] J. Zhu, Y.Q. Liu, Y.Z. Chang, E.H. Wu, Animating turbulent water by vortex shedding in PIC/FLIP, *Sci. China Inf. Sci.* 56 (3) (2013) 1–11.
- [13] E.S. Lee, C. Moulinec, R. Xu, D. Violeau, D. Laurence, P. Stansby, Comparisons of weakly compressible and truly incompressible algorithms for the SPH mesh free particle method, *J. Comput. Phys.* 227 (18) (2008) 8417–8436.
- [14] S.J. Cummins, M. Rudman, An SPH projection method, *J. Comput. Phys.* 152 (2) (1999) 584–607.
- [15] Z. Chen, Z. Zong, M.B. Liu, H.T. Li, A comparative study of truly incompressible and weakly compressible SPH methods for free surface incompressible flows, *Int. J. Numer. Methods Fluids* 73 (9) (2013) 813–829.
- [16] M. Ihmsen, J. Cornelis, B. Solenthaler, C. Horvath, M. Teschner, Implicit incompressible SPH, *IEEE Trans. Vis. Comput. Graph.* 20 (3) (2014) 426–435.
- [17] S. Koshizuka, Y. Oka, Moving-particle semi-implicit method for fragmentation of incompressible fluid, *Nucl. Sci. Eng.* 123 (3) (1996) 421–434.
- [18] S. Koshizuka, A. Nobe, Y. Oka, Numerical analysis of breaking waves using the moving particle semi-implicit method, *Int. J. Numer. Methods Fluids* 26 (7) (1998) 751–769.
- [19] X.J. Pan, H.X. Zhang, Y.T. Lu, Numerical simulation of viscous liquid sloshing by moving-particle semi-implicit method, *J. Mar. Sci. Appl.* 7 (3) (2008) 184–189.
- [20] E. Onate, A finite point method in computational mechanics, *Int. J. Numer. Methods Eng.* 39 (1996) 3839–3866.
- [21] M.B. Liu, W.P. Xie, G.R. Liu, Modeling incompressible flows using a finite particle method, *Appl. Math. Model.* 29 (12) (2005) 1252–1270.
- [22] D. Sulsky, Z. Chen, H.L. Schreyer, A particle method for history-dependent materials, *Comput. Methods Appl. Mech. Eng.* 118 (1–2) (1994) 179–196.
- [23] J. Burghardt, B. Leavy, J. Guilkey, Z. Xue, R. Brannon, Application of Uintah-MPM to shaped charge jet penetration of aluminum, in: *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 10, IOP Publishing, 2010.
- [24] P. Huang, X. Zhang, S. Ma, X. Huang, Contact algorithms for the material point method in impact and penetration simulation, *Int. J. Numer. Methods Eng.* 85 (4) (2011) 498–517.
- [25] V. Lemiale, J. Nairn, A. Hurmane, Material point method simulation of equal channel angular pressing involving large plastic strain and contact through sharp corners, *Comput. Model. Eng. Sci.* 70 (1) (2010) 41.
- [26] Z.P. Chen, X.M. Qiu, X. Zhang, Y.P. Lian, Improved coupling of finite element method with material point method based on a particle-to-surface contact algorithm, *Comput. Methods Appl. Mech. Eng.* 293 (15) (2015) 1–19.
- [27] Y.P. Lian, X. Zhang, F. Zhang, X.X. Cui, Tied interface grid material point method for problems with localized extreme deformation, *Int. J. Impact Eng.* 70 (2014) 50–61.
- [28] Y.P. Lian, Y. Liu, X. Zhang, Coupling of membrane element with material point method for fluid–membrane interaction problems, *Int. J. Mech. Mater. Des.* 10 (2) (2014) 199–211.

- [29] Y.P. Lian, X. Zhang, X. Zhou, Z.T. Ma, A FEMP method and its application in modeling dynamic response of reinforced concrete subjected to impact loading, *Comput. Methods Appl. Mech. Eng.* 200 (17–20) (2011) 1659–1670.
- [30] Y. Liu, H.K. Wang, X. Zhang, A multiscale framework for high-velocity impact process with combined material point method and molecular dynamics, *Int. J. Mech. Mater. Des.* 9 (2) (2013) 127–139.
- [31] P. Liu, Y. Liu, X. Zhang, Simulation of hyper-velocity impact on double honeycomb sandwich panel and its staggered improvement with internal-structure model, *Int. J. Mech. Mater. Des.* (2015) 1–14.
- [32] H.L. Tan, J.A. Nairn, Hierarchical, adaptive, material point method for dynamic energy release rate calculations, *Comput. Methods Appl. Mech. Eng.* 191 (19–20) (2002) 2095–2109.
- [33] Y. Guo, J.A. Nairn, Calculation of J-integration and stress intensity factors using the material point method, *Comput. Model. Eng. Sci.* 6 (2004) 295–308.
- [34] J.A. Nairn, Numerical implementation of imperfect interfaces, *Comput. Mater. Sci.* 40 (2007) 525–536.
- [35] F.A. Gilibert, V.C. Cantavella, E. Sanchez, G. Mallol, Modelling fracture process in ceramic materials using the Material Point Method, *Europhys. Lett.* 96 (2011) 24002.
- [36] Z. Chen, W. Hu, L.M. Shen, X. Xin, R. Brannon, An evaluation of the MPM for simulating dynamic failure with damage diffusion, *Eng. Fract. Mech.* 69 (2002) 1873–1890.
- [37] L.T. Tran, J. Kim, M. Berzins, Solving time-dependent PDEs using the material point method, a case study from gas dynamics, *Int. J. Numer. Methods Fluids* 62 (7) (2010) 709–732.
- [38] A.R. York, D. Sulsky, H.L. Schreyer, Fluid-membrane interaction based on the material point method, *Int. J. Numer. Methods Eng.* 48 (6) (2000) 901–924.
- [39] W.Q. Hu, Z. Chen, Model-based simulation of the synergistic effects of blast and fragmentation on a concrete wall using the MPM, *Int. J. Impact Eng.* 32 (12) (2006) 2066–2096.
- [40] S. Ma, X. Zhang, Y.P. Lian, X. Zhou, Simulation of high explosive explosion using adaptive material point method, *Comput. Model. Eng. Sci.* 39 (2) (2009) 101–123.
- [41] P. Hu, L. Xue, K. Qu, K. Ni, M. Brenner, Unified solver for modeling and simulation of nonlinear aeroelasticity and fluid-structure interactions, in: *AIAA Atmospheric Flight Mechanics Conference*, AIAA, vol. 6148, 2009, pp. 10–13.
- [42] P. Hu, L. Xue, S. Mao, et al., Material point method applied to fluid-structure interaction (FSI)/aeroelasticity problems, in: *Proceedings of 8th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, 2010.
- [43] J.E. Guillkey, T.B. Harman, B. Banerjee, An Eulerian–Lagrangian approach for simulating explosions of energetic devices, *Comput. Struct.* 85 (2007) 660–674.
- [44] J.G. Li, Y. Hamamoto, Y. Liu, X. Zhang, Sloshing impact simulation with material point method and its experimental validations, *Comput. Fluids* 103 (2014) 86–99.
- [45] Y.P. Lian, X. Zhang, Y. Liu, Coupling of finite element method with material point method by local multi-mesh contact method, *Comput. Methods Appl. Mech. Eng.* 200 (2011) 3482–3494.
- [46] C.M. Mast, P. Mackenzie-Helnwein, P. Arduino, G.R. Miller, W. Shin, Mitigating kinematic locking in the material point method, *J. Comput. Phys.* 231 (16) (2012) 5351–5373.
- [47] L. Sun, S.R. Mathur, J.Y. Murthy, An unstructured finite-volume method for incompressible flows with complex immersed boundaries, *Numer. Heat Transf., Part B, Fundam.* 58 (2010) 217–241.
- [48] A. Gilmanov, S. Acharya, A hybrid immersed boundary and material point method for simulating 3D fluid–structure interaction problems, *Int. J. Numer. Methods Fluids* 56 (12) (2008) 2151–2177.
- [49] X.X. Cui, X. Zhang, X. Zhou, Y. Liu, F. Zhang, A coupled finite difference material point method and its application in explosion simulation, *Comput. Model. Eng. Sci.* 98 (6) (2014) 565–599.
- [50] A. Stomakhin, C. Schroeder, C. Jiang, L. Chai, J. Teran, A. Selle, Augmented mpm for phase-change and varied materials, *ACM Trans. Graph. (TOG)* 33 (4) (2014) 138.
- [51] J.J. Monaghan, Simulating free surface flows with SPH, *J. Comput. Phys.* 110 (2) (1994) 399–406.
- [52] S.G. Bardenhagen, E.M. Kober, The generalized interpolation material point method, *Comput. Model. Eng. Sci.* 5 (6) (2004) 477–495.
- [53] J.P. Morris, P.J. Fox, Y. Zhu, Modeling low Reynolds number incompressible flows using SPH, *J. Comput. Phys.* 136 (1) (1997) 214–226.
- [54] S. Ma, X. Zhang, X.M. Qiu, Comparison study of MPM and SPH in modeling hypervelocity impact problems, *Int. J. Impact Eng.* 36 (2009) 272–282.
- [55] A.J. Chorin, Numerical solution of the Navier–Stokes equations, *Math. Comput.* 22 (104) (1968) 745–762.
- [56] E.Y.M. Lo, S.D. Shao, Simulation of near-shore solitary wave mechanics by an incompressible SPH method, *Appl. Ocean Res.* 24 (5) (2002) 275–286.
- [57] N. Foster, R. Fedkiw, Practical animation of liquids, in: *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, 2001, pp. 23–30.
- [58] J. Bloomenthal, C. Bajaj, J. Blinn, M.-P. Cani-Gascual, A. Rockwood, B. Wyvill, G. Wyvill, *Introduction to Implicit Surfaces*, Morgan Kaufmann Publishers Inc., San Francisco, 1997.
- [59] H.K. Zhao, A fast sweeping method for Eikonal equations, *Math. Comput.* 74 (250) (2005) 603–627.
- [60] G. Russo, P. Smereka, A remark on computing distance functions, *J. Comput. Phys.* 163 (1) (2000) 51–67.
- [61] R.P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), *J. Comput. Phys.* 152 (2) (1999) 457–492.
- [62] F. Gibou, R.P. Fedkiw, L.T. Cheng, M. Kang, A second-order-accurate symmetric discretization of the Poisson equation on irregular domains, *J. Comput. Phys.* 176 (1) (2002) 205–227.
- [63] D. Enright, D. Nguyen, F. Gibou, R. Fedkiw, Using the particle level set method and a second order accurate pressure boundary condition for free surface flows, in: *ASME/JSME 2003 4th Joint Fluids Summer Engineering Conference*, American Society of Mechanical Engineers, 2003, pp. 337–342.
- [64] D. Enright, S. Marschner, R. Fedkiw, Animation and rendering of complex water surfaces, *ACM Trans. Graph. (TOG)* 21 (3) (2002) 736–744.
- [65] C. Batty, R. Bridson, Accurate viscous free surfaces for buckling, coiling, and rotating liquids, in: *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, 2008, pp. 219–228.
- [66] Z. Ma, X. Zhang, P. Huang, An object-oriented MPM framework for simulation of large deformation and contact of numerous grains, *Comput. Model. Eng. Sci.* 55 (1) (2010) 61–87.
- [67] Z.Q. Zhou, J.O. De Kat, B. Buchner, A nonlinear 3-D approach to simulate green water dynamics on deck, in: *Proc. 7th International Symposium on Numerical Ship Hydrodynamics*, 1999, Report, no. 82000-NSH.
- [68] L. Lobovský, E. Botía-Vera, F. Castellana, J. Mas-Soler, A. Souto-Iglesias, Experimental investigation of dynamic pressure loads during dam break, *J. Fluids Struct.* 48 (2014) 407–434.
- [69] J.P. Hughes, D.I. Graham, Comparison of incompressible and weakly-compressible sph models for free-surface water flows, *J. Hydraul. Res.* 48 (S1) (2010) 105–117.
- [70] A. Colagrossi, M. Landrini, Numerical simulation of interfacial flows by smoothed particle hydrodynamics, *J. Comput. Phys.* 191 (2) (2003) 448–475.
- [71] J.U. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modeling surface tension, *J. Comput. Phys.* 100 (2) (1992) 335–354.
- [72] M.Y. Zhang, Simulation of surface tension in 2D and 3D with smoothed particle hydrodynamics method, *J. Comput. Phys.* 229 (19) (2010) 7238–7259.

- [73] L. Chen, J.H. Lee, C.F. Chen, On the modeling of surface tension and its applications by the generalized interpolation material point method, *Comput. Model. Eng. Sci.* 86 (3) (2012) 199–224.
- [74] N. Nishio, K. Yamana, Y. Yamaguchi, T. Inaba, K. Kuroda, T. Nakajima, K. Ohno, H. Fujimura, Large-scale SPH simulations of droplet impact onto a liquid surface up to the consequent formation of Worthington jet, *Int. J. Numer. Methods Fluids* 63 (12) (2010) 1435–1447.