Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 393 (2022) 114809
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

An immersed finite element material point (IFEMP) method for free surface fluid–structure interaction problems

Ming-Jian Li[a], Yanping Lian[a,*], Xiong Zhang[b,*]

[a] *Institute of Advanced Structure Technology, Beijing Institute of Technology, Beijing 100081, China*
[b] *School of Aerospace Engineering, Tsinghua University, Beijing 100084, China*

Received 11 November 2021; received in revised form 22 February 2022; accepted 23 February 2022
Available online xxxx

Abstract

The inherent nonlinearity of free surface fluid–structure interaction (FSI) problems challenges numerical methods in terms of efficiency and fidelity. In this article, we propose an immersed finite element material point method for the water entry fluid–structure interaction problems. In this method, the fluid domain is discretized by an improved incompressible material point method (iMPM) using both Eulerian and Lagrangian descriptions, while the solid domain is solved by finite element method (FEM). The interaction between the iMPM and FEM is handled by a sharp immersed interface approach. Moreover, weighted tracing points are designed to track the fluid–structure interface with a low time complexity; a particle rearranging method is developed to eliminate the numerical cavities, which are non-physical voids caused by the highly disordered particle distribution, from which the original iMPM for FSI problems suffers. Various free surface FSI problems are presented to demonstrate the accuracy and effectiveness of the proposed method. The computational results are compared with analytical, experimental, and simulation data from the literature, with good agreement in cases where such data is available. The proposed method is expected to be a powerful tool for free surface FSI problems.

Keywords: Fluid–structure interaction; Multiphase flow; Immersed boundary; Water entry; Material point method

1. Introduction

Free surface fluid–structure interaction (FSI) problem is of great interest in energy, ocean, naval industries, and many other science and engineering fields. The applications, just to name a few, include water entry of spacecrafts and projectiles, and the coupled behaviors between wave and offshore oil platforms, ships and other floating bodies [1–5]. In order to solve such problems, numerical methods need to deal with not only the complex fluid–structure coupling issues, but also the capturing/tracking of violent free surface. The inherent nonlinearity involved in such problems makes it very challenging to develop numerical methods with high efficiency and fidelity.

Many efforts have been devoted to the development of the numerical methods for FSI problems. For the time being, there are many well-known numerical methods, such as ALE (arbitrary Lagrangian–Eulerian) methods, IB (immersed boundary) methods, among others. Although ALE methods [6–14] are attractive due to convenient

* Corresponding authors.
E-mail addresses: yanping.lian@bit.edu.cn (Y. Lian), xzhang@tsinghua.edu.cn (X. Zhang).

https://doi.org/10.1016/j.cma.2022.114809
0045-7825/

implementation of FSI boundary conditions, they have difficulties in dealing with large deformation and violent free surfaces. The IB methods [15–23], on the other hand, are suitable to solve FSI problems with large structural deformation. However, as reviewed by Huang and Tian [24], the challenges of IBM are manifolds, such as violation of conservation due to interpolation, modeling of turbulence boundary layers and multi-phase flow, and parallel efficiency. Among them, the air–water–structure interaction problem (e.g., the water entry problem in this study) is a very challenging topic with various numerical difficulties, e.g., simultaneously handling the fluid–solid and fluid–fluid interfaces, coupling of multi-physical fields. A feasible strategy to solve this kind of problem is to integrate IB methods with Eulerian interface capturing schemes, such as volume of fluid (VOF) [25], level-set [26], coupled VOF Level-set [27], or moment-of-fluid (MOF) [28] methods. Efforts in coupling IB methods with level-set methods can be found in [29,30]. However, interface capturing methods will increase considerable computational costs, since the free surface is either derived by solving certain functions, or iteratively reconstructed from background cell information. Moreover, the free surface often smears over computational cells, and the reconstruction of three-material interfaces is dependent on the looping sequence of the materials, if not handled properly [31]. All these difficulties call for more sophisticated interface reconstruction techniques, which are time consuming.

In order to directly track the free surface, meshless/particles methods can be employed, e.g., SPH (smoothed particle hydrodynamics) [32,33], MPS (moving particle semi-implicit) [34–38], and MPM (material point method) [39–44]. Although many works using the particles methods for FSI and free surface FSI problems [42,45–55] are presented, two major numerical issues are frequently encountered in these Lagrangian methods: (1) non-physical pressure oscillations; (2) numerical cavities (non-physical voids manifesting in highly disordered particles/material points distribution). For the first issue, the incompressible SPH has been developed [56–61]. Kularathna et al. [62] and Zhang et al. [63] have independently proposed the incompressible MPM (iMPM), where projection method is implemented to maintain the incompressibility. By using these techniques, non-physical pressure oscillations can be greatly suppressed. For the second issue, various remedies have been proposed, such as the density-variation constraint [64], particle shifting strategy [65,66], particle coalescing and splitting technique [67], and adaptive sampling scheme [68]. Additionally, in many particle-based methods, the neighboring information of particles needs to be stored and updated for every 1 or n time step(s). For particles in the number of n_p, the basic updating method of their neighboring information has a time complexity of $O(n_\mathrm{p}^2)$, and the tree search scheme can reduce it to $O(n_\mathrm{p} \log n_\mathrm{p})$. More advanced method [69] can achieve a time complexity of $O(n_\mathrm{p})$. While in MPM and iMPM, the neighboring information of particles is not required and thus they have a promising computational efficiency when dealing with extreme deformation and multi-phase solid–liquid problems [70,71]. However, iMPM suffers from numerical cavities when solving the fluid problems, and most of the above-mentioned techniques are not suitable for MPM, due to the lack of particle neighboring information.

In this article, we propose an IFEMP (immersed finite element material point) method, which couples FEM and improved iMPM with sharp immersed interface. In IFEMP, weighted tracing points, an efficient cell tagging strategy, and a particle rearranging method are developed to distinguish FSI interfaces with a time complexity of $O(n_\mathrm{p})$, and to eliminate numerical cavities. In addition, the original iMPM developed by Zhang et al. [63] is improved in pure fluid dynamics problems in terms of efficiency and eliminating numerical cavities. Various validation examples achieve good agreement with theoretical, numerical and experimental data in the literature. Finally, a practical free surface FSI problem encountered in aerospace engineering is presented.

In the remainder of this article, Section 2 lists the governing equations in the free surface FSI problem, Section 3 introduces the IFEMP method in detail, Section 4 is validation and application examples, and Section 5 presents conclusions.

2. Governing equations

The spatial domain Ω of the fluid–structure interaction problem consists of the fluid domain Ω_f and the solid domain Ω_s, as shown in Fig. 1. For more than one solid body, the solid domain is expressed as $\Omega_{\mathrm{s}1}$, $\Omega_{\mathrm{s}2}$, etc. The solid–solid contact interface is $\Gamma_\mathrm{contact} = \partial\Omega_{\mathrm{s}1} \cap \partial\Omega_{\mathrm{s}2}$, and the solid traction and displacement boundaries are expressed by Γ_st and Γ_sd, respectively. The fluid domain has a free surface Γ_free. The FSI interface is the common interface of fluid and solid domains, namely $\Gamma_\mathrm{FSI} = \partial\Omega_\mathrm{f} \cap \partial\Omega_\mathrm{s}$. The rest of domain Ω is the environmental gas with a referencing uniform pressure. The velocity field of the gas will not be solved to save the computational cost, since its effect on the solution accuracy is negligible for the problem considered in this study.

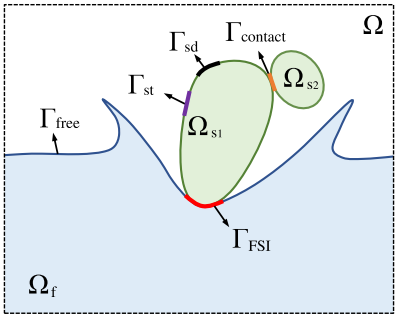


Fig. 1. The computational domain Ω consists of fluid domain Ω_{f}, solid domain Ω_{s}, solid–solid contact interface $\Gamma_{\mathrm{contact}}$, solid traction boundary Γ_{st}, solid displacement boundary Γ_{sd}, free surface Γ_{free} and FSI interface Γ_{FSI}.

2.1. Fluid domain

The general form of Navier–Stokes equation in the viscous incompressible fluid domain Ω_{f} is as follows:

$$\rho_{\mathrm{f}}\frac{\mathrm{D}\boldsymbol{u}_{\mathrm{f}}}{\mathrm{D}t} = -\nabla p + \mu\nabla^2\boldsymbol{u}_{\mathrm{f}} + \rho_{\mathrm{f}}\boldsymbol{g} + \boldsymbol{b}_{\mathrm{f}} \tag{1}$$

where ρ_{f} is the fluid density, $\boldsymbol{u}_{\mathrm{f}}$ is the fluid velocity vector, p is the fluid pressure, μ is the dynamic viscosity, $\boldsymbol{g}$ denotes gravitation, $\boldsymbol{b}_{\mathrm{f}}$ is fluid body force, and $\frac{\mathrm{D}}{\mathrm{D}t}$ denotes the material derivative with respect to time.

The continuity equation for incompressible fluid is:

$$\nabla\cdot\boldsymbol{u}_{\mathrm{f}} = 0 \tag{2}$$

On the free surface Γ_{free}, the Dirichlet pressure boundary condition is satisfied:

$$p|_{\Gamma_{\mathrm{free}}} = p_{\mathrm{atm}} \tag{3}$$

where p_{atm} is the gas pressure, and set to atmospheric pressure for water–air systems.

2.2. Structural domain

In the solid domain Ω_{s}, the momentum equation reads:

$$\rho_{\mathrm{s}}\frac{\partial^2\boldsymbol{d}}{\partial t^2} - \nabla\cdot\boldsymbol{\sigma}_{\mathrm{s}} = \boldsymbol{b}_{\mathrm{s}} \tag{4}$$

where ρ_{s} is the solid density, $\boldsymbol{d}$ is the solid displacement vector, $\boldsymbol{\sigma}_{\mathrm{s}}$ is the Cauchy stress, and $\boldsymbol{b}_{\mathrm{s}}$ denotes the solid body force.

The constitutive equation is given as:

$$\boldsymbol{\sigma}_{\mathrm{s}}^{\nabla} = \mathcal{S}(\dot{\boldsymbol{\epsilon}}, \boldsymbol{\sigma}_{\mathrm{s}}, \ldots) \tag{5}$$

where $\boldsymbol{\sigma}_{\mathrm{s}}^{\nabla}$ is the Jaumann stress rate, the function $\mathcal{S}$ is related to the specific constitutive model, and $\dot{\boldsymbol{\epsilon}}$ is the strain rate:

$$\dot{\boldsymbol{\epsilon}} = \frac{1}{2}\left[\nabla\boldsymbol{u}_{\mathrm{s}} + (\nabla\boldsymbol{u}_{\mathrm{s}})^{\mathrm{T}}\right] \tag{6}$$

with $\boldsymbol{u}_{\mathrm{s}}$ being the solid velocity.

On the traction boundary Γ_{st} and displacement boundary Γ_{su} of the solid domain, the following boundary conditions are imposed:

$$\boldsymbol{d}|_{\Gamma_{\mathrm{sd}}} = \bar{\boldsymbol{d}}_{\mathrm{sd}}, \quad (\boldsymbol{\sigma}_{\mathrm{s}}\cdot\boldsymbol{n}_{\mathrm{s}})|_{\Gamma_{\mathrm{st}}} = \bar{\boldsymbol{f}}_{\mathrm{st}} \tag{7}$$

where $\bar{\boldsymbol{d}}_{\mathrm{sd}}$ is the given boundary displacement, $\bar{\boldsymbol{f}}_{\mathrm{st}}$ is the traction, and $\boldsymbol{n}_{\mathrm{s}}$ is the normal vector of solid surface.

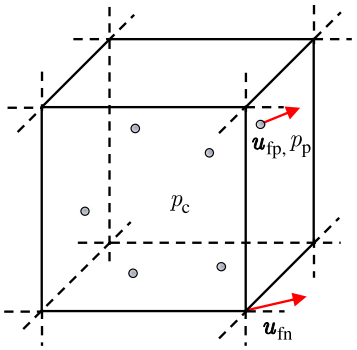


Fig. 2. Variables stores on cell centers, nodes and particles. The subscripts c, n, p represent the cell centers, nodes and particles, respectively.

For contact problems, the non-penetration condition needs to be satisfied at the solid–solid contact interface $\Gamma_{\mathrm{contact}}$:

$$[(\boldsymbol{u}_{\mathrm{s1}} - \boldsymbol{u}_{\mathrm{s2}}) \cdot \boldsymbol{n}_{\mathrm{s1}}]\,|_{\Gamma_{\mathrm{contact}}} \leqslant 0 \tag{8}$$

where $\boldsymbol{n}_{\mathrm{s1}}$ is the boundary normal vector of solid body 1, and $\boldsymbol{u}_{\mathrm{s1}}$ and $\boldsymbol{u}_{\mathrm{s2}}$ denote the velocity of solid body 1 and solid body 2, respectively.

2.3. Coupling conditions

At the fluid–structure interface Γ_{FSI}, the kinematic coupling condition is met:

$$\boldsymbol{u}_{\mathrm{f}}|_{\Gamma_{\mathrm{FSI}}} = \boldsymbol{u}_{\mathrm{s}}|_{\Gamma_{\mathrm{FSI}}} \tag{9}$$

where $\boldsymbol{u}_{\mathrm{f}}$ and $\boldsymbol{u}_{\mathrm{s}}$, respectively, denote velocities on the fluid and solid side.

The dynamic condition at the FSI interface is written as follows:

$$(\boldsymbol{\sigma}_{\mathrm{f}} \cdot \boldsymbol{n}_{\mathrm{f}})|_{\Gamma_{\mathrm{FSI}}} + (\boldsymbol{\sigma}_{\mathrm{s}} \cdot \boldsymbol{n}_{\mathrm{s}})|_{\Gamma_{\mathrm{FSI}}} = 0 \tag{10}$$

where $\boldsymbol{\sigma}_{\mathrm{f}}$ and $\boldsymbol{\sigma}_{\mathrm{s}}$ are fluid and solid stresses, $\boldsymbol{n}_{\mathrm{f}}$ and $\boldsymbol{n}_{\mathrm{s}}$ are normal vectors of fluid and solid boundaries.

3. Numerical methodology

This section introduces the IFEMP method, which belongs to the partitioned fluid–structure interaction formulation. The fluid solver is presented in Section 3.1, the solid solver in Section 3.2, and the FSI coupling algorithm in Section 3.3.

3.1. Improved iMPM for fluid dynamics

3.1.1. Spatial discretization

The iMPM uses both Eulerian and Lagrangian descriptions. The spatial domain is discretized using the background grid, as well as a set of fluid particles, as shown in Fig. 2. The initial number of particles in each cell is $2 \times 2 \times 2$ for 3D cases. The background grid (i.e., structured mesh) is used to solve the momentum and pressure equations. To avoid the "checkboard" issue, the fluid velocities $\boldsymbol{u}_{\mathrm{f}}$ are stored on nodes, and pressures p on cell centers. The fluid particles, on the other hand, are employed to transport the physical quantities, and trace the free surface in a Lagrangian way.

At the initial of nth time step, we map the particles' velocity to the background grid as follows:

$$\boldsymbol{u}_{\mathrm{fn}}^{n} = \frac{\sum_{i=1}^{n_{\mathrm{pn}}} N_{\mathrm{np}i} \boldsymbol{u}_{\mathrm{fp}}^{n}}{\sum_{i=1}^{n_{\mathrm{pn}}} N_{\mathrm{np}i}} \tag{11}$$

where $N_{\mathrm{np}i}$ is the shape function of the grid node at the ith particle, and n_{pn} is the total number of particles located in the cells attached to the fluid node. In this work, since the background cells can be regarded as finite difference

cells, as well as FEM 8-node serendipity elements, the tri-linear shape function $N = \frac{1}{8}(1 \pm \xi)(1 \pm \eta)(1 \pm \zeta)$ is employed here.

3.1.2. Chorin's projection method

Chorin's projection method [72,73] is used to solve the coupled system of momentum equation (1) and continuity equation (2). It is an operator splitting method based on the Helmholtz–Hodge splitting principle. By introducing an intermediate velocity (or predicted velocity) $\boldsymbol{u}_{\mathrm{f}}^{*}$, Eq. (1) is split into the following two equations:

$$\rho_{\mathrm{f}}\frac{\boldsymbol{u}_{\mathrm{f}}^{*}-\boldsymbol{u}_{\mathrm{f}}^{n}}{\Delta t_{\mathrm{f}}}=\mu\nabla^{2}\boldsymbol{u}_{\mathrm{f}}^{n}+\rho_{\mathrm{f}}\boldsymbol{g}+\boldsymbol{b}_{\mathrm{f}},\quad \rho_{\mathrm{f}}\frac{\boldsymbol{u}_{\mathrm{f}}^{n+1}-\boldsymbol{u}_{\mathrm{f}}^{*}}{\Delta t_{\mathrm{f}}}=-\nabla p^{n+1} \tag{12}$$

where Δt_{f} is the time step size of the fluid solver.

Since the final velocity at next time step $\boldsymbol{u}_{\mathrm{f}}^{n+1}$ should satisfy the continuity equation (2), taking divergence of the second equation in Eq. (12) and combining Eq. (2) yield:

$$\nabla\cdot\boldsymbol{u}_{\mathrm{f}}^{*}=\frac{\Delta t_{\mathrm{f}}}{\rho_{\mathrm{f}}}\nabla^{2}p^{n+1} \tag{13}$$

Eqs. (1) and (2) can thus be solved through three steps: (1) predicting an intermediate velocity, (2) solving pressure, and (3) correcting the intermediate velocity to satisfy the continuity equation as follows.

The first step of the projection method is to obtain a predicted nodal velocity $\boldsymbol{u}_{\mathrm{fn}}^{*}$ by neglecting the pressure gradient term, namely:

$$\boldsymbol{u}_{\mathrm{fn}}^{*}=\boldsymbol{u}_{\mathrm{fn}}^{n}+\frac{\Delta t_{\mathrm{f}}}{\rho_{\mathrm{fn}}}\left(\mu\nabla^{2}\boldsymbol{u}_{\mathrm{fn}}^{n}+\rho_{\mathrm{fn}}\boldsymbol{g}+\boldsymbol{b}_{\mathrm{f}}\right) \tag{14}$$

where subscript n denotes the variables associated with background nodes.

The second step is to solve the cell center's pressure at the $(n+1)$th time step using the following pressure Poisson equation (PPE):

$$\nabla^{2}p_{\mathrm{c}}^{n+1}=\frac{\rho_{\mathrm{fc}}}{\Delta t_{\mathrm{f}}}\nabla\cdot\boldsymbol{u}_{\mathrm{fc}}^{*} \tag{15}$$

where subscript c denotes the variable on cell center. In 3D cases, the cell center's velocity divergence is obtained according to the divergence free condition in Eq. (2):

$$\nabla\cdot\boldsymbol{u}_{\mathrm{fc}}^{*}=\nabla\cdot\left[\sum_{i=1}^{8}\left(N_{\mathrm{cn}i}\boldsymbol{u}_{\mathrm{fn}i}^{*}\right)\right]=\sum_{i=1}^{8}\left(\boldsymbol{u}_{\mathrm{fn}i}^{*}\cdot\nabla N_{\mathrm{cn}i}\right) \tag{16}$$

where $N_{\mathrm{cn}i}$ is the shape function of the ith grid node at the cell center. By using Eq. (16), the velocities are no longer needed to be stored on cell centers.

Fig. 3 presents the finite difference scheme for solving Eq. (15). Seven stencils are used for the 3D cases. The Laplacian term of the cell center's pressure is discretized as:

$$\nabla^{2}p_{\mathrm{c}i,j,k}=\frac{p_{\mathrm{c}i-1,j,k}-2p_{\mathrm{c}i,j,k}+p_{\mathrm{c}i+1,j,k}}{\Delta L_{\mathrm{f}x}^{2}}+\frac{p_{\mathrm{c}i,j-1,k}-2p_{\mathrm{c}i,j,k}+p_{\mathrm{c}i,j+1,k}}{\Delta L_{\mathrm{f}y}^{2}}+\frac{p_{\mathrm{c}i,j,k-1}-2p_{\mathrm{c}i,j,k}+p_{\mathrm{c}i,j,k+1}}{\Delta L_{\mathrm{f}z}^{2}} \tag{17}$$

where i, j and k, respectively, denote cell indices in the x, y, z directions. $\Delta L_{\mathrm{f}x}$, $\Delta L_{\mathrm{f}y}$ and $\Delta L_{\mathrm{f}z}$ are cell sizes in the x, y, z directions. Fig. 4 presents the boundary condition treatments when solving the pressure. In Fig. 4-(a), the left edge is Dirichlet boundary and satisfies $p = p_{\mathrm{ref}}$, where p_{ref} is a given boundary pressure. In Fig. 4-(b), the lower edge is Neumann boundary and meets $\frac{\partial p}{\partial n} = c_N$, where c_N is a given pressure gradient, and usually set to $c_N = 0$ on fixed walls. In order to obtain $p_{\mathrm{c}i,j,k}$ at the corner, $p_{\mathrm{c}i-1,j,k}$ and $p_{\mathrm{c}i,j-1,k}$ are needed. Since they are out of the computational domain, an extrapolation method is applied as follows:

$$p_{\mathrm{c}i-1,j,k}=2p_{\mathrm{ref}}-p_{\mathrm{c}i,j,k},\quad p_{\mathrm{c}i,j-1,k}=p_{\mathrm{c}i,j,k}-c_N\Delta L_{\mathrm{f}y} \tag{18}$$

After the discretization above, the PPE is transformed into linear equations. We use a direct solving method for small scale PPE, and a preconditioned conjugate gradient (PCG) method for large scale problems.

The last step of the projection method is to correct the nodal velocity as follows:

$$\boldsymbol{u}_{\mathrm{fn}}^{n+1}=\boldsymbol{u}_{\mathrm{fn}}^{*}-\frac{\Delta t_{\mathrm{f}}}{\rho_{\mathrm{fn}}}\nabla p_{\mathrm{n}}^{n+1} \tag{19}$$

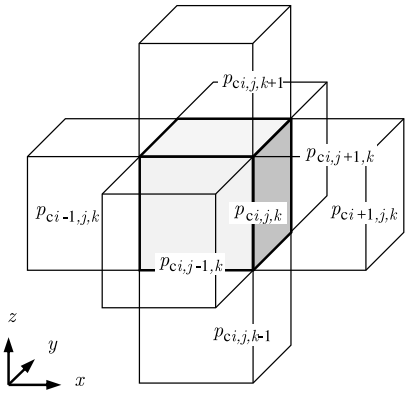


Fig. 3. Pressure storage scheme for solving the pressure Poisson equation.

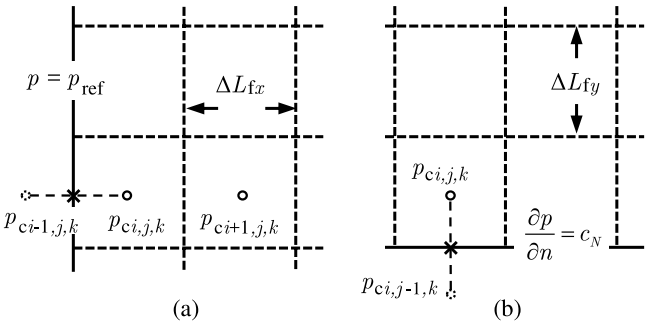


Fig. 4. Pressure boundary condition treatments: (a) Dirichlet boundary; (b) Neumann boundary.

The nodal velocity $\boldsymbol{u}_{\mathrm{fn}}^{n+1}$ at the $(n+1)$th time step is thus obtained. In order to get the nodal pressure gradient in Eq. (19), we reconstruct the nodal pressure from cell center pressure as follows:

$$p_{\mathrm{n}}^{n+1} = \frac{\sum_{i=1}^{8} N_{\mathrm{cn}i}\, p_{\mathrm{c}}^{n+1}}{\sum_{i=1}^{8} N_{\mathrm{cn}i}} \tag{20}$$

The nodal pressure gradient $\nabla p_{\mathrm{n}}^{n+1}$ is then calculated by finite difference method.

3.1.3. The particle updating scheme

The updated velocity and pressure obtained on the background grid are mapped back to the particles, and then transported by them. For particle's velocity mapping, the following scheme is used enlightened by Zhu et al. [74]:

$$\boldsymbol{u}_{\mathrm{fp}}^{n+1} = \gamma \sum_{i=1}^{8} N_{\mathrm{pn}i} \boldsymbol{u}_{\mathrm{fn}i}^{n+1} + (1-\gamma)\left(\boldsymbol{u}_{\mathrm{fp}}^{n} + \Delta t_{\mathrm{f}} \sum_{i=1}^{8} N_{\mathrm{pn}i} \frac{\partial \boldsymbol{u}_{\mathrm{fn}i}^{n}}{\partial t}\right) \tag{21}$$

where $N_{\mathrm{pn}i}$ is the shape function of the ith grid node at particle, γ is a user-set coefficient with the range from 0 to 0.3. Although this scheme is with small numerical viscosity, it is very robust according to our numerical tests. The nodal acceleration term is obtained by the following equation:

$$\frac{\partial \boldsymbol{u}_{\mathrm{fn}}^{n}}{\partial t} = \frac{1}{\rho_{\mathrm{fn}}}\left(-\nabla p_{\mathrm{n}}^{n} + \mu \nabla^2 \boldsymbol{u}_{\mathrm{fn}} + \rho_{\mathrm{fn}} \boldsymbol{g} + \boldsymbol{b}_{\mathrm{f}}\right) \tag{22}$$

The position of the particles is then be updated by using a third order TVD (total variation diminishing) Runge–Kutta scheme, and the physical information is transported by the updated particles.

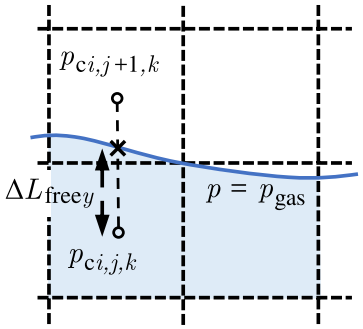


Fig. 5. Pressure boundary condition in free surface cell.

3.1.4. Free surface treatment

For a two-phase system, e.g., a water–gas system with water density ρ_{water} and gas density ρ_{gas}, the density of the cell ρ_{fc} is determined by calculating the density of the mixture. Therefore, the free surface cell has a density of $\alpha\rho_{water} + (1-\alpha)\rho_{gas}$, where α is the volume fraction of water and calculated as:

$$\alpha = \min\left\{\frac{1}{8}[\max_{i=1}^{n_{pic}}(\xi_{pi}) - \min_{i=1}^{n_{pic}}(\xi_{pi}) + \xi_{p0}]\cdot[\max_{i=1}^{n_{pic}}(\eta_{pi}) - \min_{i=1}^{n_{pic}}(\eta_{pi}) + \eta_{p0}]\cdot [\max_{i=1}^{n_{pic}}(\zeta_{pi}) - \min_{i=1}^{n_{pic}}(\zeta_{pi}) + \zeta_{p0}], 1\right\} \tag{23}$$

where $\xi_{pi}, \eta_{pi}, \zeta_{pi}$ are local coordinates of the ith particle in this cell, and n_{pic} is the number of particles in each cell. $\xi_{p0}, \eta_{p0}, \zeta_{p0}$ are initial spatial intervals between particles described by local coordinates. For a 3D cell with eight uniformly distributed particles, we have $\xi_{p0} = \eta_{p0} = \zeta_{p0} = 0.5$.

Fig. 5 presents the boundary condition treatment on the free surface when solving the PPE. The curve in the computational domain denotes the free surface (gas–liquid interface), and satisfies $p = p_{gas}$ with p_{gas} being the atmospheric pressure. Since pressure $p_{ci,j+1,k}$ is out of the computational domain, it is extrapolated as:

$$p_{ci,j+1,k} = p_{gas} - \frac{\Delta L_{fy} - \Delta L_{freey}}{\Delta L_{freey}}(p_{i,j,k} - p_{gas}) \tag{24}$$

and

$$\Delta L_{freey} = \frac{1}{2}\Delta L_{fy}\left(\left|\frac{1}{n_{pic}}\sum_{i=1}^{n_{pic}}\eta_{pi} + \frac{1}{2}\eta_{p0} - 1\right| + 1\right) \tag{25}$$

3.2. FEM for structural dynamics

The solid domain Ω_s is solved by FEM. By using virtual displacement $\delta\boldsymbol{d}$ as test function, the weak form of equilibrium equation (4) reads:

$$\int_{\Omega_s}\rho_s\ddot{\boldsymbol{d}}\cdot\delta\boldsymbol{d}\mathrm{d}\Omega + \int_{\Omega_s}\boldsymbol{\sigma}_s:\delta\nabla\boldsymbol{d}\mathrm{d}\Omega - \int_{\Omega_s}\boldsymbol{b}_s\cdot\delta\boldsymbol{d}\mathrm{d}\Omega - \int_{\Gamma_{st}}\bar{\boldsymbol{f}}_{st}\cdot\delta\boldsymbol{d}\mathrm{d}\Gamma - \int_{\Gamma_{FSI}}\bar{\boldsymbol{f}}_{sf}\cdot\delta\boldsymbol{d}\mathrm{d}\Gamma = 0 \tag{26}$$

where $\bar{\boldsymbol{f}}_{sf}$ is the fluid force at the FSI interface Γ_{FSI}, and the double superimposed dots denote second order time derivative. After FEM discretization, the displacement at any position $\boldsymbol{x}$ can by interpolated by the FEM nodal displacement $\boldsymbol{d}_k$:

$$\boldsymbol{d}(\boldsymbol{x},t) = \sum_{k=1}^{n_e}N_{ek}(\boldsymbol{x})\boldsymbol{d}_k(t) \tag{27}$$

where n_e is the number of nodes of the element, and N_{ek} is the shape function of the kth node at point $\boldsymbol{x}$.

By substituting Eq. (27) into Eq. (26), the nodal momentum equation is obtained as:

$$M_k\ddot{\boldsymbol{d}}_k = \boldsymbol{f}_{sk} \tag{28}$$

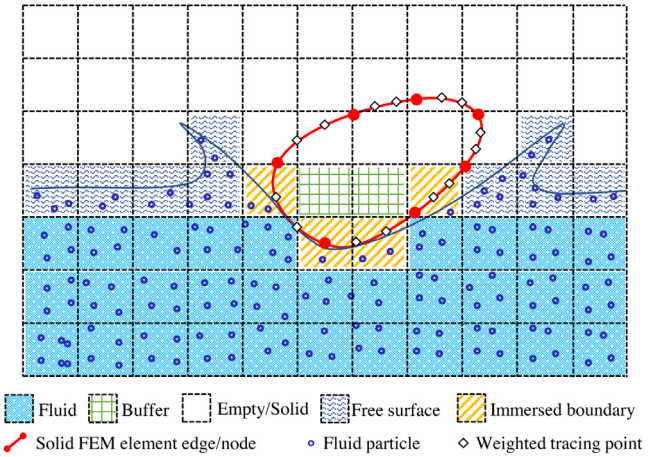


Fig. 6. Spatial discretization in the IFEMP (immersed finite element material point) method.

where the lumped mass of node k is:

$$M_k = \int_{\Omega_s} \rho_s N_{ek} \mathrm{d}\Omega \tag{29}$$

and the nodal force including external force $\boldsymbol{f}_{sk}^{\mathrm{ext}}$ and internal force $\boldsymbol{f}_{sk}^{\mathrm{int}}$ reads:

$$\begin{aligned} \boldsymbol{f}_{sk} &= \boldsymbol{f}_{sk}^{\mathrm{ext}} + \boldsymbol{f}_{sk}^{\mathrm{int}} \\ &= \int_{\Omega_s} N_{ek} \boldsymbol{b}_s \mathrm{d}\Omega + \int_{\Gamma_{st}} N_{ek} \bar{\boldsymbol{f}}_{st} \mathrm{d}\Gamma + \int_{\Gamma_{FSI}} N_{ek} \bar{\boldsymbol{f}}_{sf} \mathrm{d}\Gamma - \int_{\Omega_s} (\nabla N_{ek}) \cdot \boldsymbol{\sigma}_s \mathrm{d}\Omega \end{aligned} \tag{30}$$

where $\bar{\boldsymbol{f}}_{sf}$ is the fluid forces on solid surface and reads:

$$\bar{\boldsymbol{f}}_{sf} = -p\boldsymbol{n}_{FSI} + \mu \frac{\partial \boldsymbol{u}_f}{\partial \boldsymbol{n}_{FSI}} \tag{31}$$

with $\boldsymbol{n}_{FSI}$ being the normal vector on the FSI boundary pointing to the fluid side.

The velocity $\boldsymbol{u}_{sk}$ and displacement $\boldsymbol{d}_k$ on node k will be updated by the following explicit time integration scheme:

$$\boldsymbol{u}_{sk}^{n+1/2} = \boldsymbol{u}_{sk}^{n-1/2} + \Delta t_s^n \frac{\boldsymbol{f}_{sk}^n}{M_k^n}, \quad \boldsymbol{d}_k^{n+1} = \boldsymbol{d}_k^n + \Delta t_s^{n+1/2} \boldsymbol{u}_{sk}^{n+1/2} \tag{32}$$

where Δt_s is the time step size of the solid solver. In the FEM with explicit time integration scheme, one-point Gaussian quadrature is used, and the Flanagan/Belytschko damping is employed to suppress the hourglass modes.

To deal with the contact between different solid bodies, a point-to-surface contact method with a local search algorithm is used. The contact force is imposed by an iterative process, to simultaneously satisfy contact conditions for all contact pairs. For more details on the contact method, please refer to [75].

3.3. Fluid–structure interaction method

The IFEMP method uses a sharp immersed interface, to couple the iMPM and FEM with a Dirichlet–Neumann scheme. As shown in Fig. 6, the whole computational domain Ω is discretized into background grid, while the solid domain Ω_s is discretized into FEM elements and the fluid domain Ω_f into fluid particles. In order to accurately distinguish the FSI interface and free surface in the computational domain, background cells are tagged into five categories: fluid, immersed boundary, free surface, empty/solid, and fluid–solid buffer. At FSI interface Γ_{FSI}, the solid velocity is interpolated to the fluid side (Dirichlet boundary), and the fluid forces are applied on the solid surface (Neumann boundary).

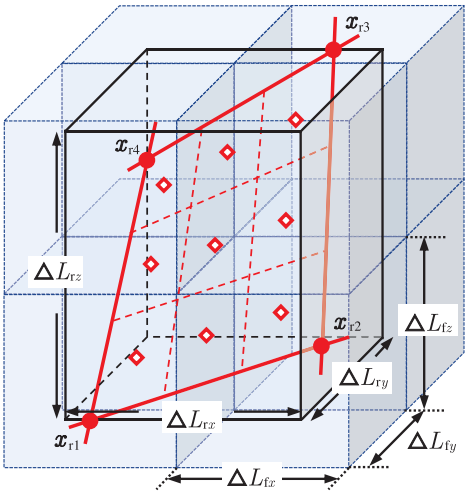


Fig. 7. Weighted tracing points (diamond shape dots) on the FEM element patches. The solid dots are FEM nodes, blocks with dotted lines are background cells, and the block with solid lines is the "bucket" occupied by the FEM patch.

It is definitely non-trivial to accurately identify the FSI interface and free surface. The separation and gathering of particles can cause numerical cavities, and bring difficulty to the calculation. In order to overcome these issues, we here introduce the concept of weighted tracing points, an efficient cell tagging strategy, and a particle rearranging method.

3.3.1. Weighted tracing points

In the strategy, the weighted tracing points are created on FEM patches of the solid surface. They will move together with the structural movement and deformation, and have twofold functions:

(1) Serving as markers to tag the cells and trace the FSI interface;
(2) Acting as a weight for FSI interface interpolation.

At the beginning of the FSI simulation, once the solid domain has been discretized into FEM elements, a group of weighted tracing points will be generated. As shown in Fig. 7, a quadrilateral FEM patch is located in the computational domain. The four FEM nodes on this patch have global coordinates $\boldsymbol{x}_{\mathrm{r}k}$ with $k = 1, \ldots, 4$, and two of the nodes will coincide if the patch degenerates into triangular patch. Considering the patch occupying a "bucket" in the computational domain (the block with solid lines in Fig. 7), the size of the bucket can be described with an array of three terms, $\Delta \boldsymbol{L}_{\mathrm{r}}(\Delta L_{\mathrm{r}x}, \Delta L_{\mathrm{r}y}, \Delta L_{\mathrm{r}z})$, which can be obtained by the maximum and minimum coordinates of the four nodes:

$$\Delta L_{\mathrm{r}i} = \max(\boldsymbol{x}_{\mathrm{r}1i}, \ldots, \boldsymbol{x}_{\mathrm{r}4i}) - \min(\boldsymbol{x}_{\mathrm{r}1i}, \ldots, \boldsymbol{x}_{\mathrm{r}4i}), \quad i = x, y, z \tag{33}$$

Since the background cell size $\Delta \boldsymbol{L}_{\mathrm{f}}(\Delta L_{\mathrm{f}x}, \Delta L_{\mathrm{f}y}, \Delta L_{\mathrm{f}z})$ is usually smaller than the bucket size $\Delta \boldsymbol{L}_{\mathrm{r}}$, we divide each edge of the FEM patch into n_{rd} segments, and thus divide the patch into n_{rd}^2 sub-patches. Therefore, n_{rd}^2 weighted tracing points are created at the centers of the sub-patches. The division number n_{rd} is determined by the following equation:

$$n_{\mathrm{rd}} = \max\left[\mathcal{R}\left(\frac{\Delta L_{\mathrm{r}x}}{\gamma_{\mathrm{rf}} \Delta L_{\mathrm{f}x}}\right), \mathcal{R}\left(\frac{\Delta L_{\mathrm{r}y}}{\gamma_{\mathrm{rf}} \Delta L_{\mathrm{f}y}}\right), \mathcal{R}\left(\frac{\Delta L_{\mathrm{r}z}}{\gamma_{\mathrm{rf}} \Delta L_{\mathrm{f}z}}\right), 1\right] \tag{34}$$

where $\gamma_{\mathrm{rf}} \in (0, 1)$ is a tunable coefficient, which means the ideal size ratio between the bucket and background cells, and $\mathcal{R}$ is the roundup function. Using a smaller γ will generate more points, which can improve the tagging accuracy at the expense of the computer memory. We find that $\gamma_{\mathrm{rf}} = 0.2$ can achieve acceptable accuracy, which usually generates 4 points for smaller patches and 9 points for larger patches.

The local coordinates $\boldsymbol{\chi}_{\mathrm{m}ij}$ of the weighted tracing point with local index (i, j) is obtained by:

$$\boldsymbol{\chi}_{\mathrm{m}ij}(\xi_{\mathrm{m}ij}, \eta_{\mathrm{m}ij}) = \left(\frac{2i-1}{n_{\mathrm{rd}}} - 1, \frac{2j-1}{n_{\mathrm{rd}}} - 1\right), \quad i = 1, \ldots, n_{\mathrm{rd}}, \quad j = 1, \ldots, n_{\mathrm{rd}} \tag{35}$$

The global coordinates of the (i, j)th weighted tracing point read:

$$\boldsymbol{x}_{\mathrm{m}ij} = \sum_{k=1}^{4} N_{\mathrm{r}ijk} \boldsymbol{x}_{\mathrm{r}k} \tag{36}$$

where $N_{\mathrm{r}ijk}$ is the shape function of kth node of the patch at (i, j)th weighted tracing point. The shape function has the same form with the linear quadrilateral element:

$$N_{\mathrm{r}ijk} = \frac{1}{4}(1 + \xi_{\mathrm{r}ij}\xi_{\mathrm{r}k})(1 + \eta_{\mathrm{r}ij}\eta_{\mathrm{r}k}), \quad k = 1, \ldots, 4 \tag{37}$$

The velocity of the (i, j)th weighted tracing point reads:

$$\boldsymbol{u}_{\mathrm{m}ij} = \sum_{k=1}^{4} N_{\mathrm{r}ijk} \boldsymbol{u}_{\mathrm{r}k} \tag{38}$$

where $\boldsymbol{u}_{\mathrm{r}k}$ is the velocity of the kth FEM node.

The area of the (i, j)th sub-patch is then assigned to the (i, j)th weighted tracing point, which is calculated by the cross-product of the sub-patch's edge vectors by the following equation:

$$\begin{aligned} A_{\mathrm{m}ij} = \frac{1}{2}[&|(\boldsymbol{x}_{\mathrm{m}ij2} - \boldsymbol{x}_{\mathrm{m}ij1}) \times (\boldsymbol{x}_{\mathrm{m}ij4} - \boldsymbol{x}_{\mathrm{m}ij3})| \\ &+ |(\boldsymbol{x}_{\mathrm{m}ij2} - \boldsymbol{x}_{\mathrm{m}ij3})| \times |(\boldsymbol{x}_{\mathrm{m}ij4} - \boldsymbol{x}_{\mathrm{m}ij3})|] \end{aligned} \tag{39}$$

with

$$\begin{cases} \boldsymbol{x}_{\mathrm{m}ij1} = \boldsymbol{x}_{\mathrm{m}}^{--} + \dfrac{(\boldsymbol{x}_{\mathrm{m}}^{+-} - \boldsymbol{x}_{\mathrm{m}}^{--})(i-1)}{n_{\mathrm{rd}}} \\ \boldsymbol{x}_{\mathrm{m}ij2} = \boldsymbol{x}_{\mathrm{m}}^{--} + \dfrac{(\boldsymbol{x}_{\mathrm{m}}^{+-} - \boldsymbol{x}_{\mathrm{m}}^{--})i}{n_{\mathrm{rd}}} \\ \boldsymbol{x}_{\mathrm{m}ij3} = \boldsymbol{x}_{\mathrm{m}}^{-+} + \dfrac{(\boldsymbol{x}_{\mathrm{m}}^{++} - \boldsymbol{x}_{\mathrm{m}}^{-+})(i-1)}{n_{\mathrm{rd}}} \\ \boldsymbol{x}_{\mathrm{m}ij4} = \boldsymbol{x}_{\mathrm{m}}^{-+} + \dfrac{(\boldsymbol{x}_{\mathrm{m}}^{++} - \boldsymbol{x}_{\mathrm{m}}^{-+})i}{n_{\mathrm{rd}}} \end{cases} \tag{40}$$

and

$$\begin{cases} \boldsymbol{x}_{\mathrm{m}}^{--} = \boldsymbol{x}_{\mathrm{r}1} + \dfrac{(\boldsymbol{x}_{\mathrm{r}4} - \boldsymbol{x}_{\mathrm{r}1})(j-1)}{n_{\mathrm{rd}}} \\ \boldsymbol{x}_{\mathrm{m}}^{+-} = \boldsymbol{x}_{\mathrm{r}2} + \dfrac{(\boldsymbol{x}_{\mathrm{r}3} - \boldsymbol{x}_{\mathrm{r}2})(j-1)}{n_{\mathrm{rd}}} \\ \boldsymbol{x}_{\mathrm{m}}^{++} = \boldsymbol{x}_{\mathrm{r}2} + \dfrac{(\boldsymbol{x}_{\mathrm{r}3} - \boldsymbol{x}_{\mathrm{r}2})j}{n_{\mathrm{rd}}} \\ \boldsymbol{x}_{\mathrm{m}}^{-+} = \boldsymbol{x}_{\mathrm{r}1} + \dfrac{(\boldsymbol{x}_{\mathrm{r}4} - \boldsymbol{x}_{\mathrm{r}1})j}{n_{\mathrm{rd}}} \end{cases} \tag{41}$$

To be noted, $A_{\mathrm{m}ij} = \dfrac{A_{\mathrm{r}}}{{n_{\mathrm{rd}}}^2}$ can be used as an approximation of the area, where A_{r} is the patch's area. This equation is more efficient, and is close to Eq. (39) when the elements are less distorted. The sub-patch area $A_{\mathrm{m}ij}$ is used as the weight for FSI interpolation. The generation process of weighted tracing points is detailed in Algorithm 1.

3.3.2. An efficient cell tagging strategy

After generating a list of weighted tracing points, the IB cells can now be tagged. However, difficulties are often encountered when tagging fluid cells due to numerical cavities. Here, we introduce an efficient cell tagging strategy,

Algorithm 1: Weighted tracing points generation

Input: FEM element and node list, fluid cell size $\Delta \boldsymbol{L}_{\mathrm{f}}$, tunable coefficient γ_{rf}
Output: Weighted tracing point list

foreach *FEM element* **do** Extract its exterior surface patches;
Form the FEM surface patches list;
foreach *FEM surface patch* **do**
  foreach *FEM node of this patch* **do** Get its global coordinates $\boldsymbol{x}_{rk}$;
  Get the bucket size of this patch $\Delta \boldsymbol{L}_{\mathrm{r}}(\boldsymbol{x}_{rk})$ by Eq. (33);
  Determine the number of divisions $n_{\mathrm{rd}}(\Delta \boldsymbol{L}_{\mathrm{r}}, \Delta \boldsymbol{L}_{\mathrm{f}}, \gamma_{\mathrm{rf}})$ by Eq. (34);
  for $i = 1$ **to** n_{rd} **do**
    for $j = 1$ **to** n_{rd} **do**
      Create the (i, j)th weighted tracing point;
      Assign its local coordinates $\boldsymbol{\chi}_{\mathrm{m}ij}(n_{\mathrm{rd}}, i, j)$ by Eq. (35);
      Obtain its global coordinates $\boldsymbol{x}_{\mathrm{m}ij}(\boldsymbol{\chi}_{\mathrm{m}ij}, \boldsymbol{x}_{\mathrm{r}k})$ by Eq. (36);
      Get its velocity $\boldsymbol{u}_{\mathrm{m}ij}\left(\boldsymbol{\chi}_{\mathrm{m}ij}, \boldsymbol{u}_{\mathrm{r}k}\right)$ by Eq. (38);
      Calculate its area $A_{\mathrm{m}ij}\left(\boldsymbol{x}_{\mathrm{r}k}, n_{\mathrm{rd}}, i, j\right)$ by Eq. (39);
    end
  end
end
Form the weighted tracing points list;

which is based on the neighboring information of background cells instead of particles. The major difference here is that the neighboring information of particles needs to be updated every time once the particles moved, while the neighboring information of background cells remain unchanged during the whole time.

The process of the efficient cell tagging strategy is presented in Algorithm 2. It does not solve any evolution equations, only takes a few times of traversals on cells and particles. The cell tagging strategy, integrated with the particle rearranging method, which will be introduced later in Section 3.3.3, can effectively and efficiently eliminate numerical cavities. The tagging strategy includes five major transversals, i.e. the five "foreach" in Algorithm 2, as shown in Fig. 8(a)–(e). The details of these steps are introduced as follows:

(1) The first step tags IB cells, and has a time complexity of $O(n_{\mathrm{c}})$ with n_{c} the background cell number. Each background cell has a dynamic array of pointers, which contains the addresses of tracing points inside this cell. Since structured grids are used, it is easy to know which cell a tracing point belongs to, simply by its global coordinates $\boldsymbol{x}_{\mathrm{m}}$. When updating position of tracing points at the end of every FSI simulating step, their addresses are incidentally pushed to the corresponding cells.
(2) This step is to eliminate numerical cavities. If an empty cell has no fluid particles, e.g., the cell with bold edges in Fig. 8-(b), it becomes a potential numerical cavity cell. If fluid particles exist in the vicinity of the cell boundary, the empty cell is tagged as a fluid cell. New particles are generated in this cell afterwards, and the numerical cavities will be effectively suppressed. The "vicinity" here is defined by "the absolute value of the natural coordinates of the particle exceeds 0.75", as the region occupied by each fluid particle is assumed to be one-eighth of the fluid cell volume. This loop has a time complexity of $O(n_{\mathrm{p}})$.
(3) This step is to only consider the influence of the "wet" part of the solid domain for improving the computational efficiency and robustness. In Fig. 8-(c), the two cells with bold edges are previous IB cells with no fluid particles inside, which indicates they may be away from the fluid domain. To further tag them correctly, the neighboring information is used. Each background cell has 6 neighboring cells (front, back, left, right, top and bottom) for 3D cases. Considering the loop of neighbors, this step has a time complexity between $O(n_{\mathrm{cIBe}})$ and $O(6n_{\mathrm{cIBe}})$, with n_{cIBe} being the number of IB cells without fluid particles inside, which is of course much smaller than the total cell number n_{c}. Actually, at most time it is not necessary to check all 6 neighbors to distinguish "dry" and "wet" cells. At average, this step has a time complexity of approximately $O(0.1n_{\mathrm{c}}) \sim O(0.3n_{\mathrm{c}})$. After this step, as shown in Fig. 8-(c), the true "dry" cell with bold edges is tagged as empty, and the "wet" cell with bold edges is still IB cell.

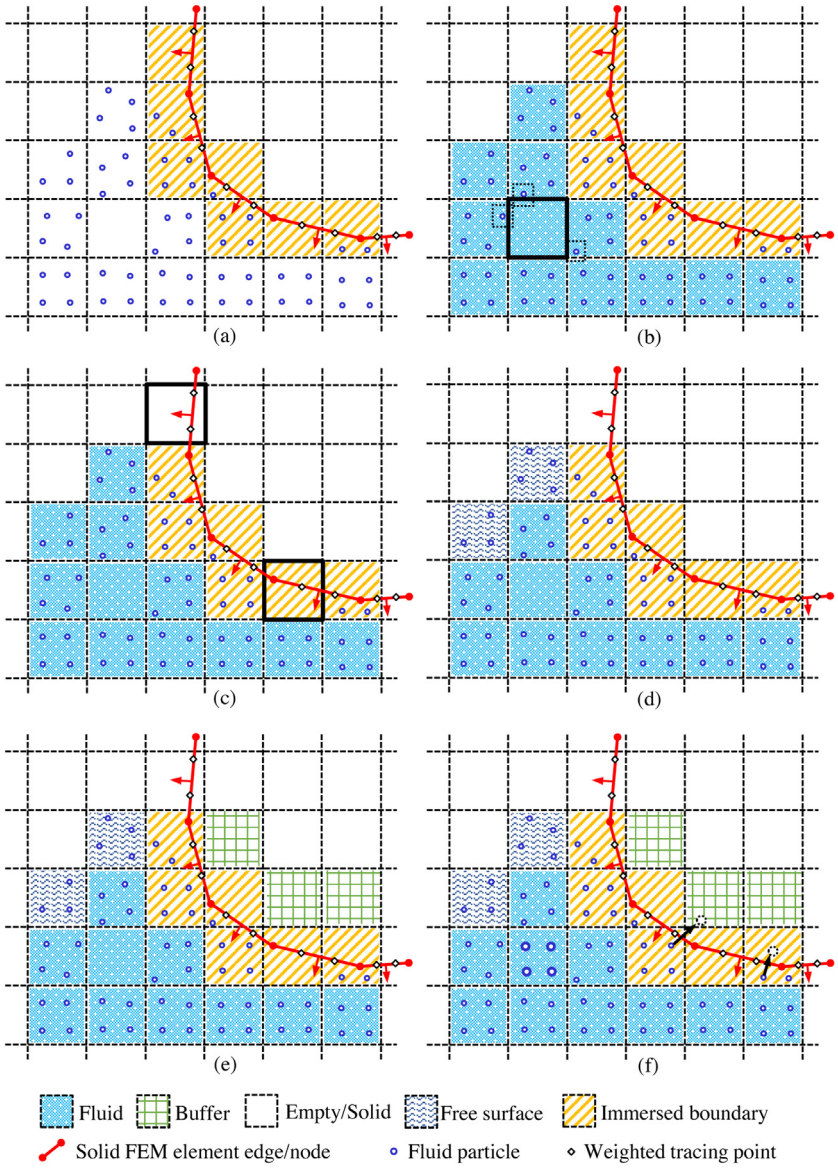


Fig. 8. Process of the efficient cell tagging strategy and particle rearranging method.

(4) This step is to tag free surface cells, see Fig. 8-(d). It has a time complexity between $O(n_{\mathrm{cf}})$ and $O(6n_{\mathrm{cf}})$, with n_{cf} the number of fluid cells. Usually most of the background cells are fluid cells, the time complexity is thus approximately $O(1.8n_{\mathrm{c}}) \sim O(2.4n_{\mathrm{c}})$.

(5) This step is to tag fluid–solid buffer cells, corresponding to Fig. 8-(e). The buffer cells help to absorb and delete particles that penetrate through the FEM patches. This can prevent from the effect of the penetrated particles on the tagging accuracy of next time step. The time complexity of this step is $O(n_{\mathrm{cIB}})$ with n_{cIB} the number of IB cells, which is approximately $O(0.1n_{\mathrm{c}}) \sim O(0.3n_{\mathrm{c}})$.

As stated above, the overall time complexity of the cell tagging method is approximately $O(n_{\mathrm{p}} + 3n_{\mathrm{c}}) \sim O(n_{\mathrm{p}} + 4n_{\mathrm{c}})$. Since a background cell usually contains 8 particles for 3D cases, the time complexity of the cell tagging method is about $O(n_{\mathrm{p}})$, which is efficient.

Algorithm 2: The efficient cell tagging strategy

Input: Background cells list, fluid particles list, weighted tracing points list
Output: Background cells list with updated tags

foreach *background cell* **do** (1)
  if *it contains tracing points* **then** tag as IB cell **else** tag as empty cell;
end
foreach *fluid particle* **do** (2)
  Get the cell contains this particle;
  if *the cell is* empty **then** tag as fluid cell;
  if *the particle is in the vicinity of an* empty *cell* **then**
    tag the empty neighbor as fluid cell
  end
end
foreach IB *cell without fluid particles inside* **do** (3)
  Check its neighboring information;
  if *it does not has any* fluid *neighbor(s)* **then** tag as empty cell;
end
foreach fluid *cell* **do** (4)
  Check its neighboring information;
  if *it has* empty *neighbor(s)* **then** tag as free surface;
end
foreach IB *cell* **do** (5)
  Get its average normal vector $\boldsymbol{n}_{\mathrm{IB}}$;
  Tag empty neighbors at the reverse directions of $\boldsymbol{n}_{\mathrm{IB}}$ as fluid–solid buffer cell;
end

3.3.3. Particle rearranging method

Particles will gather, separate, or penetrate the FEM surface patches after a certain time of movement. We here introduce a particle rearranging method, which deals with this problem. As can be seen in Fig. 8-(f), new particles (larger dots) are generated in the numerical cavity cell, and the particles are not allowed to penetrate the FSI interface. During the particle rearranging, the total number of particles can be changed. This will not lead to mass conservation problem, since the mass conservation is guaranteed by divergence free condition on background cells. The rearranging process of fluid particles is concluded in Algorithm 3.

As shown in Algorithm 3, there are three steps in sequence as follows:

(1) If the following equation is met, the particle is considered to penetrate the FEM surface patches:

$$\left(\boldsymbol{x}_{\mathrm{fp}}-\boldsymbol{x}_{\mathrm{cm}}\right)\cdot\boldsymbol{n}_{\mathrm{cm}}<0 \tag{42}$$

where $\boldsymbol{x}_{\mathrm{fp}}$ is the fluid particle's position, and $\boldsymbol{x}_{\mathrm{cm}}$ is the averaged FEM surface position and can be obtained by:

$$\boldsymbol{x}_{\mathrm{cm}}=\frac{\sum_{I=1}^{n_{\mathrm{mic}}}\boldsymbol{x}_{\mathrm{m}I}A_{\mathrm{m}I}}{\sum_{I=1}^{n_{\mathrm{mic}}}A_{\mathrm{m}I}} \tag{43}$$

where $\boldsymbol{x}_{\mathrm{m}I}$ is the weighted tracing point's position, and $A_{\mathrm{m}I}$ is the area carried by the weighted tracing point.

Algorithm 3: Particle rearranging method

Input: Particles list, background cells list, weighted tracing points list
Output: Rearranged particles list

```
foreach fluid–solid buffer cell do
    Remove all particles inside;
end
foreach IB cell do (1)
    Remove the particles penetrate FEM surface patches;
end
foreach fluid cell without particles do (2)
    Generate n_pic0 new particles;
end
foreach fluid cell with too many particles do
    Get its neighboring information;
    if it has neither IB nor free surface neighbors then
        Remove original particles and generate n_pic0 new particles ;
    end
end
```

Here, $\boldsymbol{n}_{\mathrm{cm}}$ is the averaged FEM surface normal vector calculated by:

$$\boldsymbol{n}_{\mathrm{cm}} = \frac{\sum_{I=1}^{n_{\mathrm{mic}}} \boldsymbol{n}_{\mathrm{m}I} A_{\mathrm{m}I}}{\sum_{I=1}^{n_{\mathrm{mic}}} A_{\mathrm{m}I}} \tag{44}$$

where $\boldsymbol{n}_{\mathrm{m}I}$ is the sub-patch's normal vector carried by the weighted tracing point, and n_{mic} is the number of weighted tracing points in the background cell.

(2) In this step, n_{pic0} is the initial number of particles in each cell, which is usually 8 for 3D cases. The pressure and velocity of new particles will be interpolated from the background nodes by shape functions.

3.3.4. Dirichlet-Neumann coupling

Since the background cells are already tagged, the solid velocity can be applied to fluid boundaries though background nodes. For background nodes belonging to IB cells, the nodal velocity is interpolated from the weighted tracing points:

$$\boldsymbol{u}_{\mathrm{fn}} = \frac{\sum_{I=1}^{n_{\mathrm{mn}}} \boldsymbol{u}_{\mathrm{m}I} A_{\mathrm{m}I}}{\sum_{I=1}^{n_{\mathrm{mn}}} A_{\mathrm{m}I}} \tag{45}$$

where n_{mn} is the number of weighted tracing points around the node, and $\boldsymbol{u}_{\mathrm{m}k}$ is velocity of the weighted tracing point given by Eq. (38).

As shown in Fig. 9, the fluid forces are applied at the FSI interface Γ_{FSI}, which has been discretized into FEM sub-patches. The sub-patches are represented by weighted tracing points generated in Algorithm 1. The pressure on one patch of the solid element p_{patch} can thus be given by:

$$p_{\mathrm{patch}} = \frac{\sum_{I=1}^{n_{\mathrm{rd}}^2} p_{\mathrm{m}I} A_{\mathrm{m}I}}{\sum_{I=1}^{n_{\mathrm{rd}}^2} A_{\mathrm{m}I}} \tag{46}$$

where n_{rd}^2 is the number of sub-patches, as mentioned in Eq. (34), and $p_{\mathrm{m}I}$ is the pressure on the Ith weighted tracing point given by the cell pressure. For example, the following equations are met in Fig. 9:

$$p_{\mathrm{m1}} = p_{\mathrm{m2}} = p_{ci-1,j,k}, \quad p_{\mathrm{m3}} = p_{ci,j,k}, \quad p_{\mathrm{m4}} = p_{\mathrm{m5}} = p_{ci,j-1,k} \tag{47}$$

The solid equivalent nodal forces contributed by the fluid effect will then be obtained by using standard FEM procedures.

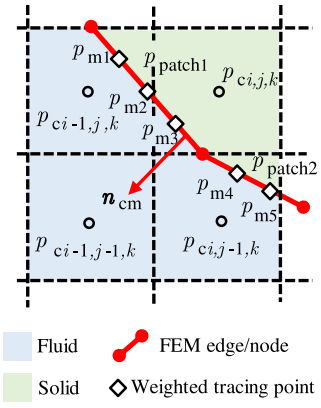


Fig. 9. The pressure implementation on FSI interface.

3.3.5. Time integration

The time integration schemes of the fluid domain and solid domain have been respectively introduced in Sections 3.1 and 3.2. The updating scheme of position of fluid particles is as follows:

$$\begin{cases} \boldsymbol{x}_{\mathrm{fp}}^{n+1/3} = \boldsymbol{x}_{\mathrm{fp}}^{n} + \Delta t_{\mathrm{f}} \boldsymbol{u}_{\mathrm{fp}}^{n} \\ \boldsymbol{x}_{\mathrm{fp}}^{n+2/3} = \dfrac{3}{4}\boldsymbol{x}_{\mathrm{fp}}^{n} + \dfrac{1}{4}\boldsymbol{x}_{\mathrm{fp}}^{n+1/3} + \dfrac{1}{4}\Delta t_{\mathrm{f}} \boldsymbol{u}_{\mathrm{fp}}^{n+1/3} \\ \boldsymbol{x}_{\mathrm{fp}}^{n+1} = \dfrac{1}{3}\boldsymbol{x}_{\mathrm{fp}}^{n} + \dfrac{2}{3}\boldsymbol{x}_{\mathrm{fp}}^{n+2/3} + \dfrac{2}{3}\Delta t_{\mathrm{f}} \boldsymbol{u}_{\mathrm{fp}}^{n+2/3} \end{cases} \tag{48}$$

The position updating of weighted tracing points uses the FEM nodal position:

$$\boldsymbol{x}_{\mathrm{m}} = \sum_{k=1}^{4} N_{\mathrm{r}k} \boldsymbol{x}_{\mathrm{r}k} \tag{49}$$

During the FSI calculation, the fluid time step Δt_{f} is determined by the CFL condition:

$$\Delta t_{\mathrm{f}} \leqslant \min\left(\frac{\Delta L_{\mathrm{f}}}{|\boldsymbol{u}_{\mathrm{f}}|}\right) \tag{50}$$

where the characterized velocity $\boldsymbol{u}_{\mathrm{f}}$ is often controlled by the velocity of IB interface. Since the FEM uses the explicit central difference scheme, the solid time step size Δt_{s} is determined by the solid sound speed:

$$\Delta t_{\mathrm{s}} \leqslant \min\left(\frac{\Delta L_{\mathrm{e}}}{\sqrt{E_{\mathrm{s}}/\rho_{\mathrm{s}}}}\right) \tag{51}$$

where ΔL_{e} is the element size, and E_{s} is the Young's modulus. The solid time step size (determined by the sound speed) is often much smaller than the fluid time step size, i.e. $\Delta t_{\mathrm{s}} \ll \Delta t_{\mathrm{f}}$, due to an explicit integration scheme used in FEM. Therefore, after one fluid solving step, the solid domain will solve for several steps, until the physical time synchronized for the fluid domain and solid domain, and the fluid and solid domains are therefore loosely coupled.

3.4. The solving procedures

The flowchart of the IFEMP (immersed finite element material point) method is shown in Fig. 10 and details for one time step are as follows.

1. Reconstruct background nodal velocity $\boldsymbol{u}_{\mathrm{fn}}$ by Eq. (11).
2. Identify fluid, solid, free surface and FSI domains by tagging background cells in Algorithm 2;
3. Apply solid–fluid Dirichlet boundary condition by interpolating velocities from weighted tracing points $\boldsymbol{u}_{\mathrm{m}}$ to background nodes $\boldsymbol{u}_{\mathrm{fn}}$ in Eq. (45);

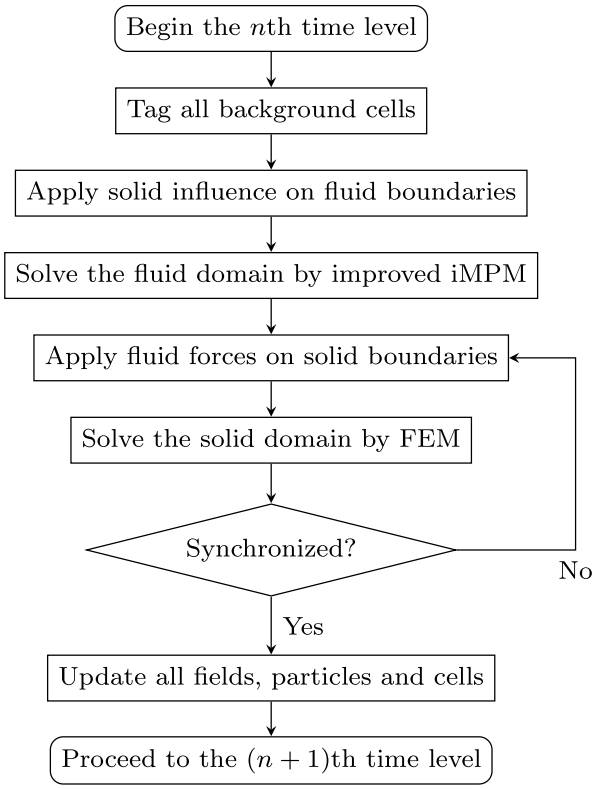


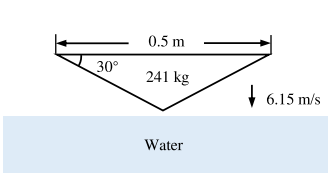
Fig. 10. Solving process of incompressible immersed boundary material point method.

4. Solve fluid fields, obtain fluid nodal velocity $\boldsymbol{u}_{\mathrm{fn}}$ and cell center pressure p_{c} by conducting the three-step projection method in Eqs. (14), (15) and (19);
5. Apply fluid–solid Neumann boundary condition by employing FSI force $\bar{\boldsymbol{f}}_{\mathrm{sf}}$ in Eq. (31);
6. Solve the structural fields by FEM in Eqs. (26)–(32);
7. Obtain velocities of fluid particles $\boldsymbol{u}_{\mathrm{fp}}$ by Eq. (21), rearrange and update positions of fluid particles by Algorithm 3 and (48);

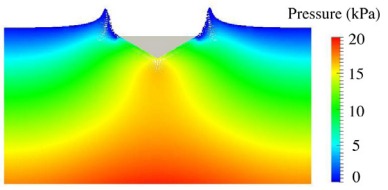
4. Numerical examples

4.1. Validation of solid velocity and fluid pressure during water entry

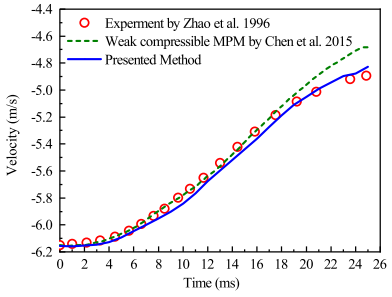
First, the water entry process of a wedge is presented. As shown in Fig. 11-(a), the wedge is 1 m long, 0.5 m wide and has a mass of 241 kg. The initial velocity at the moment it touches the water surface is 6.15 m/s. The water domain has a width of 2 m and a depth of 1 m. The wedge is discretized by hexahedral elements and the water domain by 3D background grids, with $2 \times 2 \times 2$ initial particles in each cell. Symmetric boundary conditions were applied on both fluid and solid grids to simulate the 2D case. The pressure field with the distribution of particles at 0.04 s is plotted in Fig. 11-(b). As can be seen, the pressure contour is smooth without obvious non-physical oscillations. Time history of the velocity of the wedge has been presented in Fig. 11-(c), where the experimental result by Zhao et al. [76] and the simulated result using weakly compressible MPM [77] are provided for comparison. Furthermore, Fig. 11-(d) compares the pressure distribution along the wedge at 0.0202 s between the analytical result [78], the experimental data [76], and the simulated results by the proposed method and SPH [45], where the definitions of the non-dimensional z^* and p^* can be found in [45]. It is clear that the predicted velocity and pressure results are in line with those in the literature.



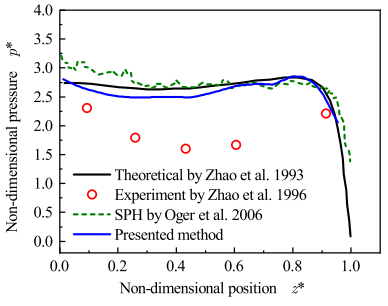
(a) Geometry of the wedge.



(b) Pressure and particles distribution at 0.04 s.



(c) Velocity of the wedge comparing with experimental data [76] and simulated results by weakly compressible MPM [77]



(d) Pressure distribution along the wedge at 0.0202 s comparing with theoretical solution [78], experimental data [76] and simulated results by SPH [45].

Fig. 11. Numerical results of the water entry problem of a wedge.

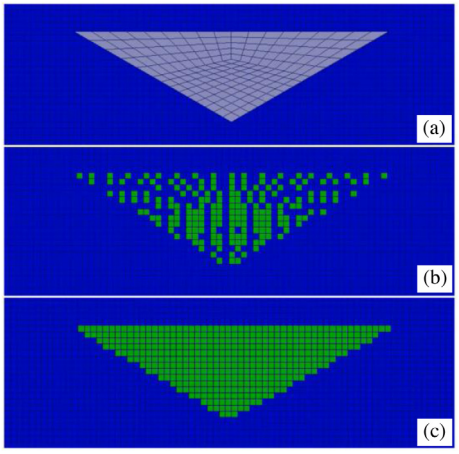


Fig. 12. (a) Front face of FEM hexahedral elements in the background cells; (b) Cell tagging without weighted tracing points; (c) Cell tagging with weighted tracing points.

Fig. 12 demonstrates that weighted tracing points work well in the water entry problem, where the FEM elements are larger than the background cells. If we tag the cells by searching FEM nodes, we will get IB cells as shown in Fig. 12-(b). From the observation, one can find that many cells are wrongly tagged and consequently jeopardize the calculation. By using weighted tracing points, all cells are correctly tagged, as shown in Fig. 12-(c).

The second case is the water entry of an oblique cylinder [47,79]. The cylinder has a density of 900 kg/m^3, a diameter of 0.05 m, and a length of 0.2 m. The initial vertical velocity is 6.11 m/s, with an inclination angle of 56.5°. The weighted tracing points on the FEM surface patches are plotted in Fig. 13, where 4 tracing points are created on smaller patches and 9 tracing points on larger patches. The evolution of the interaction between the

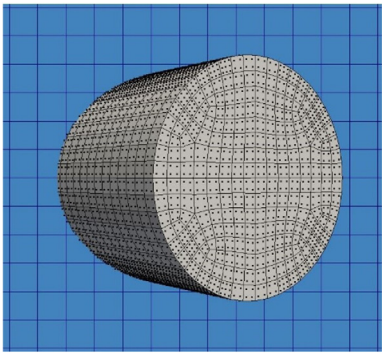


Fig. 13. Weighted tracing points on the FEM surface patches of the cylinder.

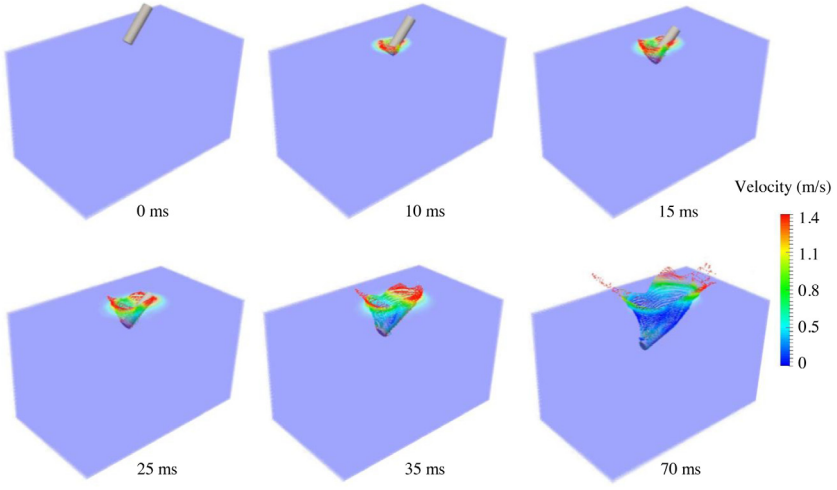


Fig. 14. The fluid particles' distribution and their velocities at different time instances.

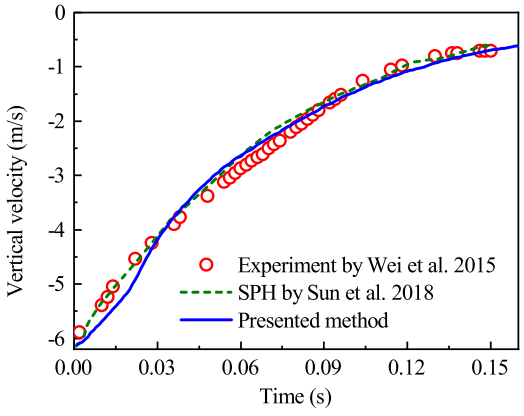


Fig. 15. Vertical velocity of the oblique cylinder, compared with experimental data [79] and simulated results by SPH [47].

oblique cylinder and the water is plotted in Fig. 14 by snapshots at increasing time. Furthermore, a quantitative comparison of the vertical velocity profile of the cylinder between the experimental data [79] and numerical results from the proposed method and SPH [47] is presented in Fig. 15, which demonstrates the accuracy of the proposed method.

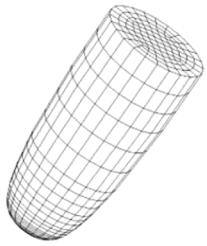


Fig. 16. FEM model of the bullet.

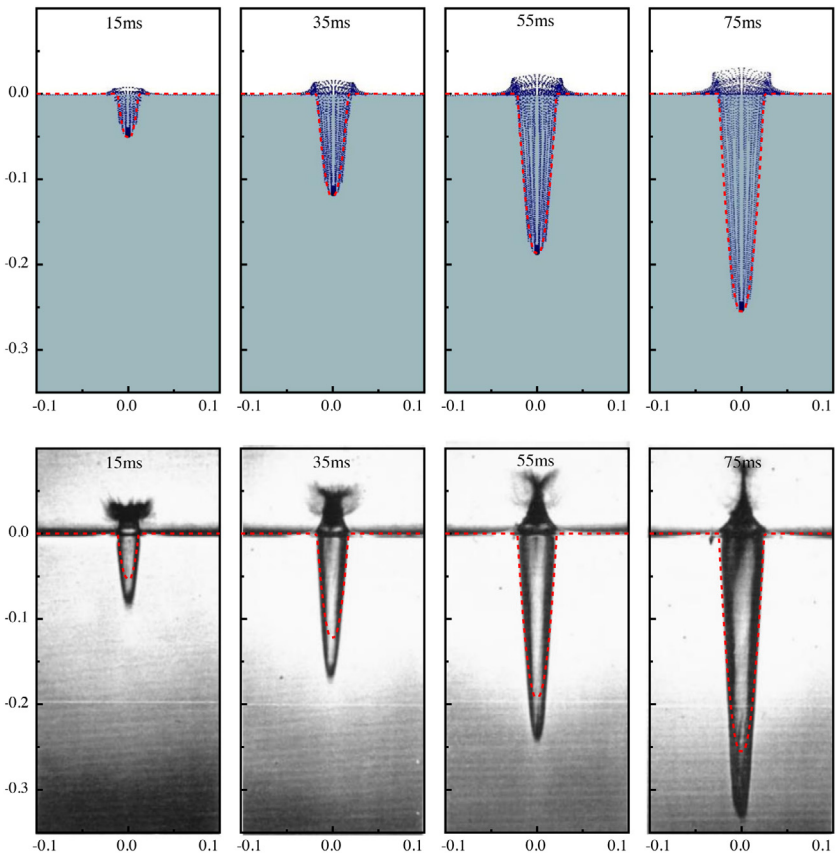


Fig. 17. Cavity shape formed by a vertically water-entry bullet: simulated results by the algorithm in this article (shown by the particles); the Logvinovich's principle (shown by the dashed lines); and the experimental images [80].

4.2. Validation of cavity shape evolution

This section presents the cavity shape evolution during the water entry of a bullet. The profile of the bullet is a least square curve given in experimental work by Shi et al. [80], and was modeled by FEM, as shown in Fig. 16. The cavity formed by the bullet was recorded by the high-speed camera in their experiments. The lead bullet has a mass of 2.67 g, a density of 11,400 kg/m^3, a length of 12.3 mm and a diameter of 5.7 mm. The initial velocity is 342 m/s with $\pm$5 m/s error in the experiments, and the simulated initial velocity is 340 m/s. The water tank is 60 cm in length and width, and 80 cm in depth. Fig. 17 compares the cavity shape from our numerical result, semi-theoretical result with Logvinovich's principle (given in the Appendix), and the experimental result. Note that

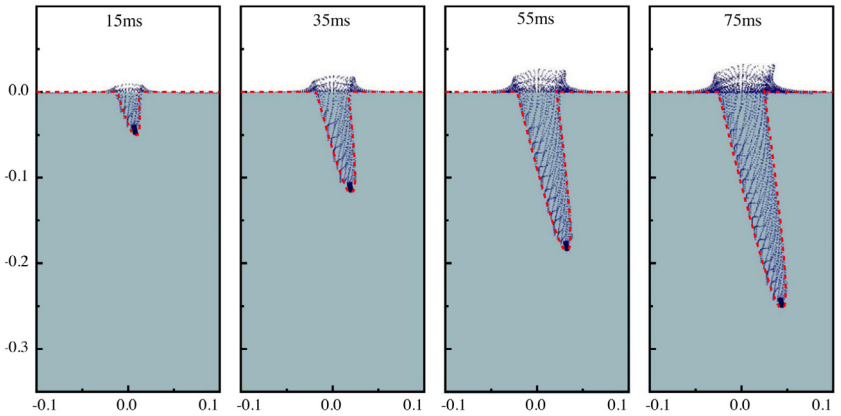


Fig. 18. Cavity shape formed by a obliquely water-entry bullet: simulated result by the algorithm in this article comparing with the Logvinovich's principle.

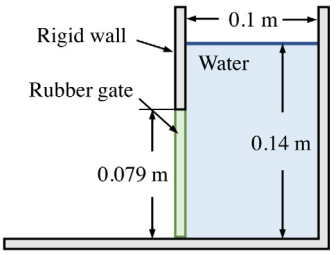


Fig. 19. Geometry of the dam break flow through a rubber gate.

the outermost particles in free surface cells are used to illustrate the cavity shape. As can be seen, the simulated and semi-theoretical cavity shapes are close. However, considerable difference exists between the experimental images and the theoretical/simulated results. Since only images of cavity shape are available from [80], the error of image extraction is a possible reason for the difference.

Fig. 18 shows the water-entry of the bullet with an inclination angle of $\theta_v = 80°$. As can be seen, the simulated cavity shape is in line with the Logvinovich's principle.

4.3. Validation of dam break problems with large structural deformations

This section presents two free surface FSI problems with large structural deformations: (1) dam-break flow through a hyperelastic gate and (2) dam-break flow onto an elastic baffle. The two cases are conducted using 3D models with symmetric boundary conditions.

For the first case, the experiment of the dam-break flow through a rubber gate conducted by Antoci et al. [81] is reproduced. The initial configuration of this case is illustrated in Fig. 19. The water column is with width of 0.1 m, height of 0.14 m, density of 1000 kg/m^3, and dynamic viscosity of 1×10^{-3} Pa s. The rubber gate is with thickness of 0.005 m, height of 0.079 m, and fixed at its upper end. The rubber was approximated by Antoci et al. [81] as a linear elastic model with Young's modulus of 12 MPa and Poisson's ratio of 0.4. However, a hyperelastic model will be more appropriate to describe mechanical behavior of the rubber. Lobovsky et al. [82] and Yang et al. [83] used hyperelastic models, which fitted the tensile test data by Antoci et al. [84], to simulate the rubber gate in this case, and achieved better agreement with experimental results. The results by Yang et al. [83] suggested that by using a linear elastic model, the stiffness of rubber gate is over-estimated, and the maximum horizontal displacement of the gate is about 30% lower than that simulated by a hyperelastic model (e.g., Mooney–Rivlin model).

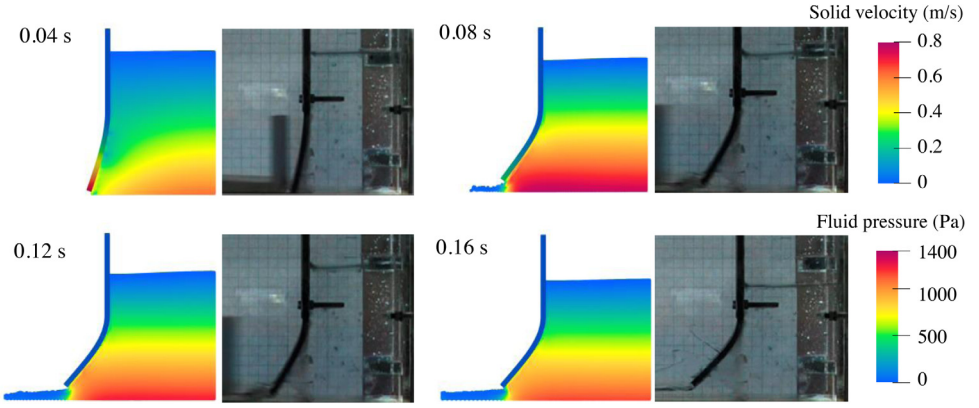


Fig. 20. Fluid pressure and solid velocity at different time instances: dam-break flow through a rubber gate. The experimental results are from Antoci et al. [81].

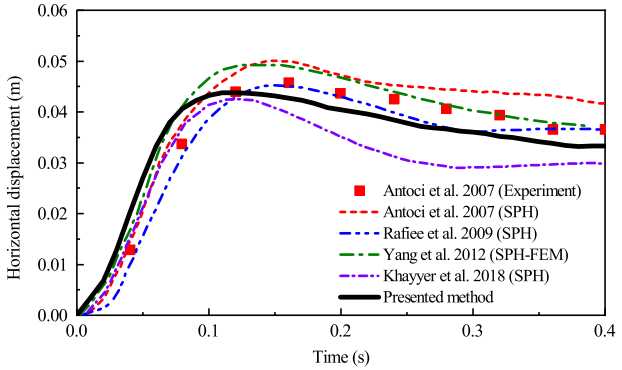


Fig. 21. Horizontal displacement of the free end of rubber gate, compared with experimental data by Antoci et al. [81], and simulated results by Antoci et al. [81], Rafiee et al. [85], Yang et al. [83] and Khayyer et al. [86].

In this study, we apply an incompressible hyperelastic Mooney–Rivlin model to simulate the rubber gate. In this model, the strain energy function W reads:

$$W = C_{10}\,(I_1 - 3) + C_{01}\,(I_2 - 3) \tag{52}$$

where I_1 and I_2 are the invariants of Cauchy–Green deformation tensor, C_{10} and C_{01} are material constants with the relationship of $2(C_{10}+C_{01}) = G$, and G is the shear modulus. In this case, $C_{10} = -0.277$ MPa and $C_{01} = 1.1$ MPa according to the tensile test data by Antoci et al. [84]. The density of rubber is 1100 kg/m^3. The water domain is discretized into 56 000 particles and the rubber gate into 80 elements. Fig. 20 presents fluid pressure distribution and solid velocity distribution at the time instance of 0.04, 0.08, 0.12 and 0.16 s.

Fig. 21 shows the profile of horizontal displacement at the free end of the rubber gate, where experimental data by Antoci et al. [81], and numerical results by Antoci et al. [81], Rafiee et al. [85], Yang et al. [83] and Khayyer et al. [86] are presented for comparison. A good agreement with the results from the literature demonstrates the accuracy and capability of the proposed method, in terms of structural deformation prediction in this free surface FSI problem. In addition, we also conduct the simulation with a linear elastic model for the rubber gate. The maximum horizontal displacement achieved is 0.034 m, which agrees well with the result by Yang et al. [83] using linear elastic model (0.035 m).

For the second case, the initial configuration is shown in Fig. 22. The water column is initially placed in a water tank. The tank is open at the top and has rigid walls in other boundaries. An elastic baffle is fixed at the bottom

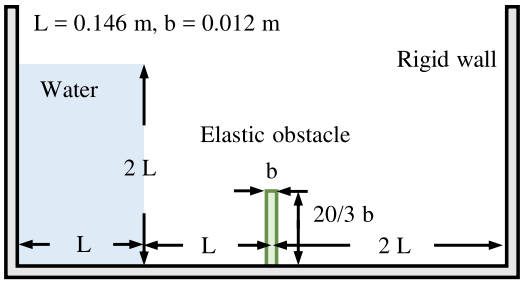


Fig. 22. Geometry of the dam break flow on an elastic baffle.

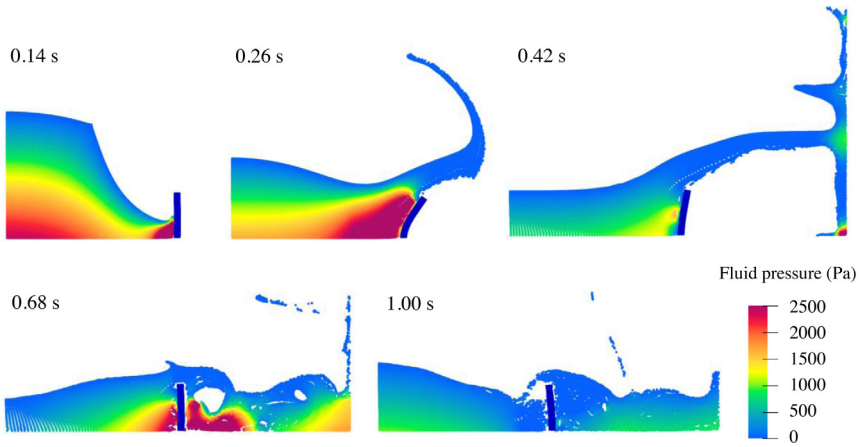


Fig. 23. Fluid pressure at different time instances: dam-break flow onto an elastic baffle.

wall of the water tank. The water has density of 1000 kg/m^3 and dynamic viscosity of 1×10^{-3} Pa s. The material law used for the baffle is a linear elastic model with Young's modulus of 1 MPa, Poisson's ratio of 0, and density of 2500 kg/m^3. The water domain is discretized into 42 632 particles and the baffle into 60 elements.

Fig. 23 presents the wave front, fluid pressure distribution, and solid deformation with snapshots at increasing time. Fig. 24 presents the time history of deflection of the baffle tip. Since there is no available experiment data of this case, the predicted results from the proposed method are compared with other available numerical results in the literature [42,85,87,88].

4.4. Application in water entry of a structure with airbags

This section presents a practical application of the proposed method in aerospace engineering. It is the water entry process of a structure with airbags as shown in Fig. 25. The box consists of a cubic shell and internal components, which are discretized using the shell–solid mixed elements. The four airbags are discretized by membrane elements, and the ropes are discretized by truss elements. The material laws used for the structure in this case are linear elastic constitutive laws. The box has Young's modulus of 70 GPa, Poisson's ratio of 0.3 and density of 2700 kg/m^3, the rope has Young's modulus of 12 GPa, Poisson's ratio of 0.2 and density of 750 kg/m^3, and the membrane has Young's modulus of 1 GPa, Poisson's ratio of 0.2 and density of 800 kg/m^3. The airbags are inflated with an initial pressure p_0, and the pressure in the airbag p_{bag} is calculated by the equation of state as follows:

$$p_{\text{bag}} \Omega_{\text{bag}}^{\gamma_{\text{bag}}} = p_0 \Omega_0^{\gamma_{\text{bag}}} \tag{53}$$

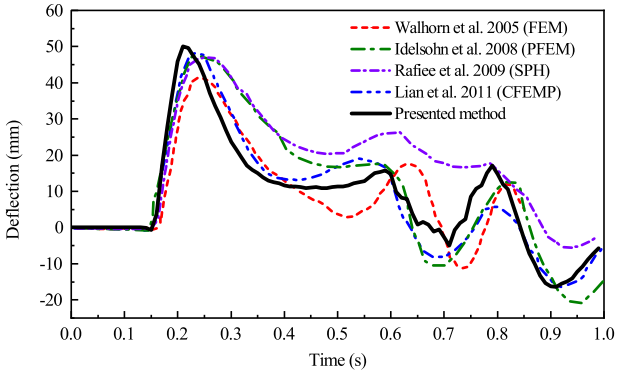


Fig. 24. Deflection of the elastic baffle tip, compared with other numerical results by FEM [87], PFEM [88], SPH [85] and CFEMP [42].

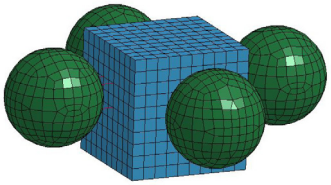


Fig. 25. FEM mesh of the structure with airbags: the spherical airbags are attached to the cube with four ropes.

where Ω_0 and Ω_{bag} are the initial and current volume of the airbag, respectively, γ_{bag} is the coefficient of the gas state equation. To calculate the volume of the airbag, the surface patches on the airbag are connected to the centroid of the airbag to form tetrahedrons and pyramids. The volume of the airbag is thus the sum of all these tetrahedrons and pyramids.

Two cases are simulated. In both cases, the structure falls into the water with an initial speed of $\sim$50 m/s, and an inclination angle of 90° (for Case I) and 45° (for Case II).

Figs. 26 and 27 present the water entry processes of the structure with airbags for Case I and II, respectively, where the cavity interface is illustrated by the contour of cell tags. As shown in these figures, the box decelerates since it touches the water surface. Meanwhile, airbags move upwards rapidly thanks to the fluid force, collide with the box, and then drag the box upward through the ropes between them. As a result, a large cavity is gradually formed, and the structure decelerates greatly until the closure of the cavity. It is recognized that for water entry case with 45° inclination angle, the structure rotates and finally sinks vertically, which is different from Case I.

For both cases, 8 million fluid particles are used for the fluid domain, and 2030 elements in total for the structure. The simulations above were run in a personal computer with an Intel i7-9900 processor with eight cores. It took about 15 h for a physical time of 1.5 s, consisting of fluid time steps of 15 000 and the structural time steps of about ten times the former.

5. Conclusions

In this article, an IFEMP (immersed finite element material point) method is proposed. It is a partitioned fluid–structure interaction coupling method, where an improved incompressible material point method is used for fluid domain and FEM for structure domain, and the sharp interface immersed boundary method for the representation of the fluid–structure interface. Moreover, weighted tracing points, an efficient cell tagging strategy, and particle rearranging method are developed. These techniques can efficiently and correctly tag the background cells, and effectively eliminate numerical cavities. Various numerical examples, including water entry problems, dam-break flow through a rubber gate, onto an elastic baffle, and water entry of a structure with airbags, are presented with their predicted results in line with theoretical, experimental and other numerical results in the literature. It is demonstrated that IFEMP is a powerful tool for solving free surface FSI problems.

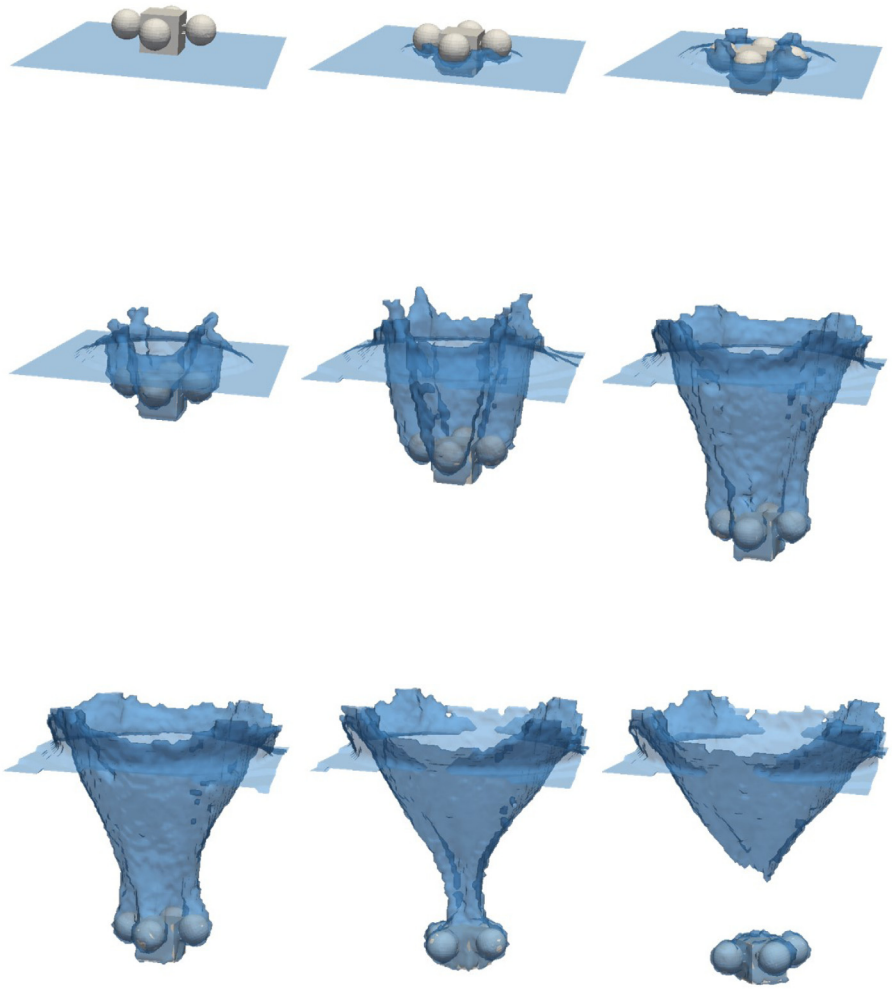


Fig. 26. Vertical water entry of the structure with airbags.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors acknowledge the support from the National Natural Science Foundation of China, under Grant Nos. 52105287, 11972086 and 12172192.

Appendix. Logvinovich's solution to cavity shape evolution

For water entry of high speed objects, Logvinovich [89] proposed the principle of independence of the cavity sections expansion in the 1960s. According to this principle, in the cavity formed by a high speed object, along the trajectory center line of the object, each cross section of the cavity expands independently, and its expansion depends only on the state of the object at the moment when it passes through the cross section (including object size, velocity, drag, pressure difference, etc.), and is not affected by the movement of the object at other moments.

The Logvinovich's principle was verified by many experiments afterwards. For instance, Semenenko et al. [90] used a projectile to penetrate a group of steel plates in a water tank. The results showed that in the fluid regions

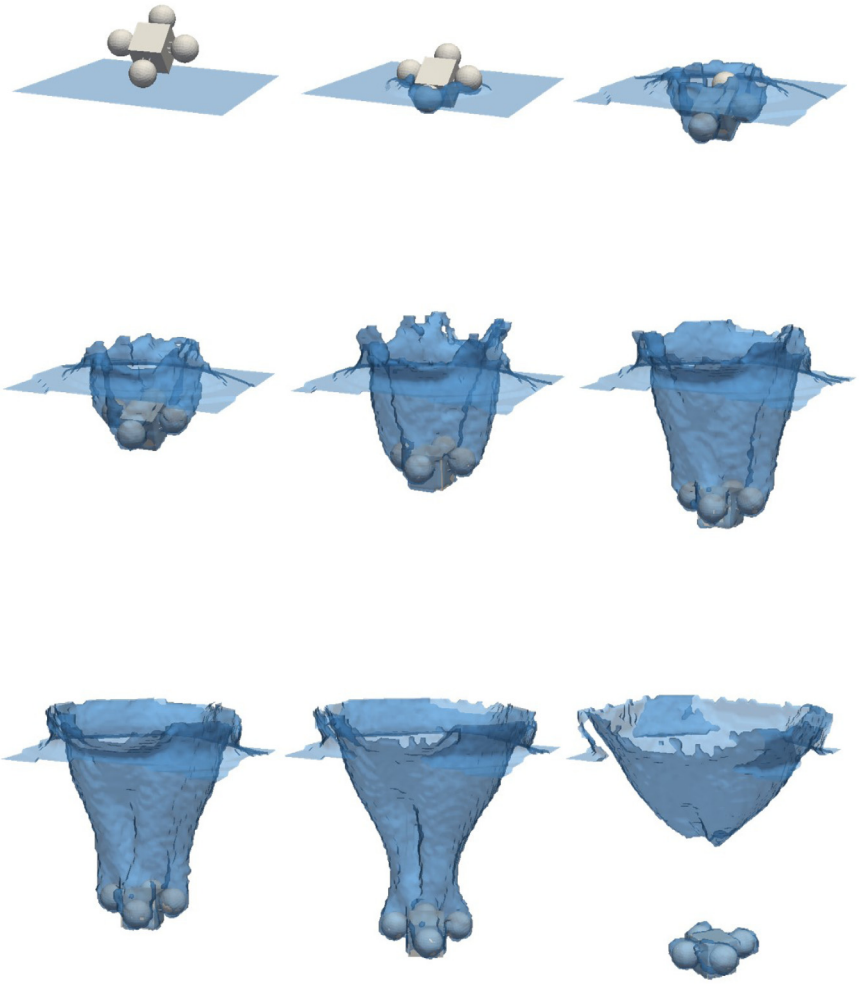


Fig. 27. Water entry of the structure with airbags with a 45° inclination angle.

separated by the steel plate, the cavity developed independently, which proved that the development of cavity section was only related to the initial state. According to the Logvinovich's principle, we can obtain the following semi-theoretical solution of the cavity shape.

Let us use $S(\iota, t)$ to denote the section area of the cavity formed after water entry of the object, where ι is the coordinate along the trajectory, and satisfies $\iota = 0$ at the centroid of the object. $t = 0$ is for the moment the object touches the water surface. $\dot{S}$ and $\ddot{S}$ are the speed and acceleration of the expansion of cavity section, and satisfy:

$$\ddot{S} = -\frac{\beta \Delta p}{\rho_{\mathrm{f}}}, \quad \beta = \frac{4\pi C_{\mathrm{v}}}{a^2} \tag{A.1}$$

where a is a coefficient determined by experiments, which is usually $2 \sim 2.5$. Δp is the pressure difference, ρ_{f} is the fluid density, C_{v} is the drag coefficient of the object, which is a function of the cavitation number σ_{v}. Here we use the following relation:

$$C_{\mathrm{v}} = C_{\mathrm{v0}}(1 + \sigma_{\mathrm{v}}), \quad C_{\mathrm{v0}} = 0.83 \tag{A.2}$$

The expansion speed of the cavity section $\dot{S}$ is:

$$\dot{S} = \dot{S}_0 - \beta \int_0^t \frac{\Delta p}{\rho_{\mathrm{f}}} \mathrm{d}t, \quad \dot{S}_0 = \frac{2\pi C_x R_0 u_0}{a} \tag{A.3}$$

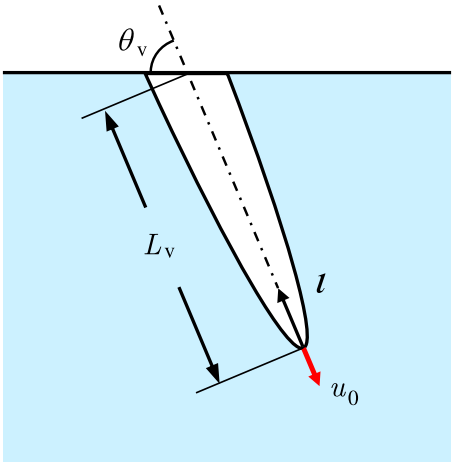


Fig. A.28. Cavity formed by the bullet.

where S_0 is the initial section area, R_0 is the initial section radius, u_0 is the initial velocity, the cavity section area S is:

$$S = S_0 + \dot{S}_0 t - \beta \int_0^t \int_0^t \frac{\Delta p}{\rho} \mathrm{d}t \mathrm{d}t \tag{A.4}$$

Considering a constant u_0 and an inclination angle θ_v shown in Fig. A.28, and using L_v to denote the total length of the trajectory, the area of the cavity section is then:

$$S(\iota, t) = S_0 + \frac{\iota}{u_0}\left[\dot{S}_0 - \beta g(L_v - \iota)\sin\theta_v\, t\right] - \frac{\beta g L \sin\theta_v}{2u_0^2}\iota^2 + \frac{\beta g \sin\theta_v}{2u_0^2}\iota^3 \tag{A.5}$$

where g is the gravitation acceleration, when $\theta_v = 90°$, the equation becomes the vertical water entry problem.

References

[1] T.T. Truscott, B.P. Epps, J. Belden, Water entry of projectiles, Annu. Rev. Fluid Mech. 46 (2014) 355–378.
[2] K.G. Bodily, S.J. Carlson, T.T. Truscott, The water entry of slender axisymmetric bodies, Phys. Fluids 26 (7) (2014) 072108.
[3] J. Jiao, S. Huang, T. Tezdogan, M. Terziev, C.G. Soares, Slamming and green water loads on a ship sailing in regular waves predicted by a coupled CFD–FEA approach, Ocean Eng. 241 (2021) 110107.
[4] S.A.M. Gavgani, H.H. Jalali, M.F. Farzam, Semi-active control of jacket platforms under wave loads considering fluid-structure interaction, Appl. Ocean Res. 117 (2021) 102939.
[5] Y. Zhou, D. Ning, D. Liang, S. Cai, Nonlinear hydrodynamic analysis of an offshore oscillating water column wave energy converter, Renew. Sustain. Energy Rev. 145 (2021) 111086.
[6] W.F. Noh, CEL: A time-dependent, two-space-dependent, coupled Euler-Lagrange code, Methods Comput. Phys. 3 (1964) 117–179.
[7] C.W. Hirt, A.A. Amsden, J.L. Cook, An arbitrary Lagrangian-Eulerian computing method for all flow speeds, J. Comput. Phys. 135 (2) (1997) 203–216.
[8] T.J.R. Hughes, W.K. Liu, T. Zimmermann, Lagrangian-Eulerian finite element formulation for incompressible viscous flows, Comput. Methods Appl. Mech. Engrg. 29 (3) (1981) 329–349.
[9] J. Donea, S. Giuliani, J.P. Halleux, An arbitrary Lagrangian-Eulerian finite element method for transient dynamic fluid-structure interactions, Comput. Methods Appl. Mech. Engrg. 33 (1982) 689–723.
[10] L.G. Margolin, M. Shashkov, Second-order sign-preserving conservative interpolation (remapping) on general grids, J. Comput. Phys. 184 (1) (2003) 266–298.
[11] J. Degroote, J. Vierendeels, Multi-solver algorithms for the partitioned simulation of fluid–structure interaction, Comput. Methods Appl. Mech. Engrg. 200 (25) (2011) 2195–2210.
[12] J.-L. Pfister, O. Marquet, M. Carini, Linear stability analysis of strongly coupled fluid–structure problems with the arbitrary-Lagrangian–Eulerian method, Comput. Methods Appl. Mech. Engrg. 355 (2019) 663–689.
[13] B. Augier, J. Yan, A. Korobenko, J. Czarnowski, G. Ketterman, Y. Bazilevs, Experimental and numerical FSI study of compliant hydrofoils, Comput. Mech. 55 (6) (2015) 1079–1090.
[14] M. Balmus, A. Massing, J. Hoffman, R. Razavi, D.A. Nordsletten, A partition of unity approach to fluid mechanics and fluid–structure interaction, Comput. Methods Appl. Mech. Engrg. 362 (2020) 112842.

[15] C.S. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (2) (1972) 252–271.
[16] J. Yang, F. Stern, Sharp interface immersed-boundary/level-set method for wave–body interactions, J. Comput. Phys. 228 (17) (2009) 6590–6616.
[17] F.-B. Tian, H. Dai, H. Luo, J.F. Doyle, B. Rousseau, Fluid–structure interaction involving large deformations: 3D simulations and applications to biological systems, J. Comput. Phys. 258 (2014) 451–469.
[18] F. Örley, V. Pasquariello, S. Hickel, N.A. Adams, Cut-element based immersed boundary method for moving geometries in compressible liquid flows with cavitation, J. Comput. Phys. 283 (2015) 1–22.
[19] A. Nitti, J. Kiendl, A. Reali, M.D. de Tullio, An immersed-boundary/isogeometric method for fluid–structure interaction involving thin shells, Comput. Methods Appl. Mech. Engrg. 364 (2020) 112977.
[20] J. Boustani, M.F. Barad, C.C. Kiris, C. Brehm, An immersed boundary fluid–structure interaction method for thin, highly compliant shell structures, J. Comput. Phys. 438 (2021) 110369.
[21] S. Tschisgale, J. Fröhlich, An immersed boundary method for the fluid-structure interaction of slender flexible structures in viscous fluid, J. Comput. Phys. 423 (2020) 109801.
[22] J. Ma, Z. Wang, J. Young, J.C. Lai, Y. Sui, F.-B. Tian, An immersed boundary-lattice Boltzmann method for fluid-structure interaction problems involving viscoelastic fluids and complex geometries, J. Comput. Phys. 415 (2020) 109487.
[23] F. Sotiropoulos, X. Yang, Immersed boundary methods for simulating fluid–structure interaction, Prog. Aerosp. Sci. 65 (2014) 1–21.
[24] W.-X. Huang, F.-B. Tian, Recent trends and progress in the immersed boundary method, Proc. Inst. Mech. Eng. C 233 (23) (2019) 7617–7636.
[25] 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.
[26] 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.
[27] M. Sussman, E.G. Puckett, A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows, J. Comput. Phys. 162 (2) (2000) 301–337.
[28] V. Dyadechko, M. Shashkov, Moment-of-Fluid Interface Reconstruction, Los Alamos Report la-UR-05-7571, 2005.
[29] A. Calderer, S. Kang, F. Sotiropoulos, Level set immersed boundary method for coupled simulation of air/water interaction with complex floating structures, J. Comput. Phys. 277 (2014) 201–227.
[30] L. Wang, G.M. Currao, F. Han, A.J. Neely, J. Young, F.-B. Tian, An immersed boundary method for fluid–structure interaction with compressible multiphase flows, J. Comput. Phys. 346 (2017) 131–151.
[31] V. Dyadechko, M. Shashkov, Reconstruction of multi-material interfaces from moment data, J. Comput. Phys. 227 (11) (2008) 5361–5384.
[32] L.B. Lucy, A numerical approach to the testing of the fission hypothesis, Astrophys. J. 8 (12) (1977) 1013–1024.
[33] J.J. Monaghan, Simulating free surface flows with SPH, J. Comput. Phys. 110 (2) (1994) 399–406.
[34] S. Koshizuka, Y. Oka, Moving-particle semi-implicit method for fragmentation of incompressible fluid, Nucl. Sci. Eng. 123 (3) (1996) 421–434.
[35] X. Pan, H. Zhang, Y. Lu, Numerical simulation of viscous liquid sloshing by moving-particle semi-implicit method, J. Mar. Sci. Appl. 7 (3) (2008) 184–189.
[36] M. Tanaka, T. Masunaga, Stabilization and smoothing of pressure in MPS method by quasi-compressibility, J. Comput. Phys. 229 (11) (2010) 4279–4290.
[37] A. Khayyer, H. Gotoh, Enhancement of stability and accuracy of the moving particle semi-implicit method, J. Comput. Phys. 230 (8) (2011) 3093–3118.
[38] Y. Sun, G. Xi, Z. Sun, A generic smoothed wall boundary in multi-resolution particle method for fluid–structure interaction problem, Comput. Methods Appl. Mech. Engrg. 378 (2021) 113726.
[39] D. Sulsky, S. Zhou, H.L. Schreyer, Application of a particle-in-cell method to solid mechanics, Comput. Phys. Comm. 87 (1995) 236–252.
[40] D. Sulsky, Z. Chen, H.L. Schreyer, A particle method for history-dependent materials, Comput. Methods Appl. Mech. Engrg. 118 (1994) 179–196.
[41] X. Zhang, Z. Chen, Y. Liu, The Material Point Method: A Continuum-Based Particle Method for Extreme Loading Cases, Cambridge Academic Press, 2016.
[42] Y. Lian, X. Zhang, Y. Liu, Coupling of finite element method with material point method by local multi-mesh contact method, Comput. Methods Appl. Mech. Engrg. 200 (47) (2011) 3482–3494.
[43] Y. Lian, X. Zhang, Y. Liu, An adaptive finite element material point method and its application in extreme deformation problems, Comput. Methods Appl. Mech. Engrg. 241–244 (2012) 275–285.
[44] C.M. Mast, P. Mackenziehelnwein, P. Arduino, G.R. Miller, W. Shin, Mitigating kinematic locking in the material point method, J. Comput. Phys. 231 (16) (2012) 5351–5373.
[45] G. Oger, M. Doring, B. Alessandrini, P. Ferrant, Two-dimensional SPH simulations of wedge water entries, J. Comput. Phys. 213 (2) (2006) 803–822.
[46] M. Liu, J. Shao, H. Li, An SPH model for free surface flows with moving rigid objects, Internat. J. Numer. Methods Fluids 74 (9) (2014) 684–697.
[47] P. Sun, A.-M. Zhang, S. Marrone, F. Ming, An accurate and efficient SPH modeling of the water entry of circular cylinders, Appl. Ocean Res. 72 (2018) 60–75.
[48] Y.-X. Zhang, D.-C. Wan, T. Hino, Comparative study of MPS method and level-set method for sloshing flows, J. Hydrodyn. 26 (4) (2014) 577–585.

[49] Z.-P. Chen, X. Zhang, X.-M. Qiu, Y. Liu, A frictional contact algorithm for implicit material point method, Comput. Methods Appl. Mech. Engrg. 321 (2017) 124–144.
[50] J. Guilkey, T. Harman, B. Banerjee, An Eulerian-Lagrangian approach for simulating explosions of energetic devices, Comput. Struct. 85 (11–14) (2007) 660–674.
[51] X.X. Cui, X. Zhang, X. Zhou, Y. Liu, F. Zhang, A coupled finite difference material point method and its application in explosion simulation, CMES Comput. Model. Eng. Sci. 98 (6) (2014) 565–599.
[52] Z. Zhang, T. Long, J. Chang, M. Liu, A smoothed particle element method (SPEM) for modeling fluid–structure interaction problems with large fluid deformations, Comput. Methods Appl. Mech. Engrg. 356 (2019) 261–293.
[53] S.L. Fuchs, C. Meier, W.A. Wall, C.J. Cyron, A novel smoothed particle hydrodynamics and finite element coupling scheme for fluid–structure interaction: The sliding boundary particle approach, Comput. Methods Appl. Mech. Engrg. 383 (2021) 113922.
[54] W.-K. Sun, L.-W. Zhang, K. Liew, A smoothed particle hydrodynamics–peridynamics coupling strategy for modeling fluid–structure interaction problems, Comput. Methods Appl. Mech. Engrg. 371 (2020) 113298.
[55] P. Zhang, S. Sun, Y. Chen, S. Galindo-Torres, W. Cui, Coupled material point Lattice Boltzmann method for modeling fluid–structure interactions with large deformations, Comput. Methods Appl. Mech. Engrg. 385 (2021) 114040.
[56] S.J. Cummins, M. Rudman, An SPH projection method, J. Comput. Phys. 152 (2) (1999) 584–607.
[57] F. Colin, R. Egli, F.Y. Lin, Computing a null divergence velocity field using smoothed particle hydrodynamics, J. Comput. Phys. 217 (2) (2006) 680–692.
[58] 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, Internat. J. Numer. Methods Fluids 73 (9) (2013) 813–829.
[59] S. Shao, E.Y.M. Lo, Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface, Adv. Water Resour. 26 (7) (2003) 787–800.
[60] C. Tsurudome, D. Liang, Y. Shimizu, A. Khayyer, H. Gotoh, Study of beach permeability's influence on solitary wave runup with ISPH method, Appl. Ocean Res. 117 (2021) 102957.
[61] A. Khayyer, Y. Shimizu, H. Gotoh, S. Hattori, Multi-resolution ISPH-SPH for accurate and efficient simulation of hydroelastic fluid-structure interactions in ocean engineering, Ocean Eng. 226 (2021) 108652.
[62] S. Kularathna, K. Soga, Implicit formulation of material point method for analysis of incompressible materials, Comput. Methods Appl. Mech. Engrg. 313 (2017) 673–686.
[63] F. Zhang, X. Zhang, K.Y. Sze, Y. Lian, Y. Liu, Incompressible material point method for free surface flow, J. Comput. Phys. 330 (2017) 92–110.
[64] X. Hu, N. Adams, An incompressible multi-phase SPH method, J. Comput. Phys. 227 (1) (2007) 264–278.
[65] R. Xu, P. Stansby, D. Laurence, Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach, J. Comput. Phys. 228 (18) (2009) 6703–6725.
[66] S. Lind, R. Xu, P. Stansby, B. Rogers, Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves, J. Comput. Phys. 231 (4) (2012) 1499–1523.
[67] R. Vacondio, B. Rogers, P. Stansby, P. Mignosa, J. Feldman, Variable resolution for SPH: A dynamic particle coalescing and splitting scheme, Comput. Methods Appl. Mech. Engrg. 256 (2013) 132–148.
[68] R. Ando, N. Thurey, R. Tsuruno, Preserving fluid sheets with adaptively sampled anisotropic particles, IEEE Trans. Vis. Comput. Graph. 18 (8) (2012) 1202–1214.
[69] S. Plimpton, Fast parallel algorithms for short-range molecular dynamics, J. Comput. Phys. 117 (1) (1995) 1–19.
[70] A. Yerro, A. Rohe, K. Soga, Modelling internal erosion with the material point method, Procedia Eng. 175 (2017) 365–372, Proceedings of the 1st International Conference on the Material Point Method (MPM 2017).
[71] F. Ceccato, A. Yerro, V. Girardi, P. Simonini, Two-phase dynamic MPM formulation for unsaturated soil, Comput. Geotech. 129 (2021) 103876.
[72] A.J. Chorin, Numerical solution of the Navier-Stokes equations, Math. Comp. 22 (104) (1968) 745–762.
[73] J. Guermond, P. Minev, J. Shen, An overview of projection methods for incompressible flows, Comput. Methods Appl. Mech. Engrg. 195 (44) (2006) 6011–6045.
[74] Y.N. Zhu, R. Bridson, Animating sand as a fluid, ACM Trans. Graph. 24 (3) (2005) 965–972.
[75] Y. Song, Y. Liu, X. Zhang, A non-penetration FEM-MPM contact algorithm for complex fluid-structure interaction problems, Comput. & Fluids 213 (2020) 104749.
[76] R. Zhao, O. Faltinsen, J. Aarsnes, Water entry of arbitrary two-dimensional sections with and without flow separation, in: Proceedings of the 21st Symposium on Naval Hydrodynamics, National Academy Press, Trondheim, Norway, Washington, DC, USA, 1996, pp. 408–423.
[77] Z. Chen, X. Qiu, X. Zhang, Y. Lian, Improved coupling of finite element method with material point method based on a particle-to-surface contact algorithm, Comput. Methods Appl. Mech. Engrg. 293 (2015) 1–19.
[78] R. Zhao, O. Faltinsen, Water entry of two-dimensional bodies, J. Fluid Mech. 246 (1993) 593–612.
[79] Z. Wei, C. Hu, Experimental study on water entry of circular cylinders with inclined angles, J. Mar. Sci. Technol. 20 (4) (2015) 722–738.
[80] H.-H. Shi, M. Itoh, T. Takami, Optical observation of the supercavitation induced by high-speed water entry, J. Fluids Eng. 122 (4) (2000) 806–810.
[81] C. Antoci, M. Gallati, S. Sibilla, Numerical simulation of fluid–structure interaction by SPH, Comput. Struct. 85 (11) (2007) 879–890.
[82] L. Lobovsky, P. Groenenboom, Remarks on FSI simulations using SPH, in: 4th ERCOFTAC SPHERIC Workshop on SPH Applications, 2009, pp. 378–383.

[83] Q. Yang, V. Jones, L. McCue, Free-surface flow interactions with deformable structures using an SPH–FEM model, Ocean Eng. 55 (2012) 136–147.
[84] C. Antoci, Simulazione Numerica Dell'interazione Fluido-Struttura Con La Tecnica SPH (Ph.D. thesis, Ph.D. thesis), Universita di Pavia. 100, 2006.
[85] A. Rafiee, K.P. Thiagarajan, An SPH projection method for simulating fluid-hypoelastic structure interaction, Comput. Methods Appl. Mech. Engrg. 198 (33) (2009) 2785–2795.
[86] A. Khayyer, H. Gotoh, H. Falahaty, Y. Shimizu, An enhanced ISPH–SPH coupled method for simulation of incompressible fluid–elastic structure interactions, Comput. Phys. Comm. 232 (2018) 139–164.
[87] E. Walhorn, A. Kölke, B. Hübner, D. Dinkler, Fluid–structure coupling within a monolithic model involving free surface flows, Comput. Struct. 83 (25) (2005) 2100–2111.
[88] S. Idelsohn, J. Marti, A. Limache, E. Oñate, Unified Lagrangian formulation for elastic solids and incompressible fluids: Application to fluid–structure interaction problems via the PFEM, Comput. Methods Appl. Mech. Engrg. 197 (19) (2008) 1762–1776.
[89] G. Logvinovich, Hydrodynamics of Flows with Free Boundaries, Naukova Dumka, Kiev, 1969.
[90] V. Semenenko, Artificial Supercavitation. Physics and Calculation, in: Lecture Notes from the RTO AVT/VKI Special Course on Supercavitating Flows, 2001, p. 33.