Computers and Mathematics with Applications 81 (2021) 649–663

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Interaction between free surface flow and moving bodies with a dynamic mesh and interface geometric reconstruction approach

Ming-Jian Li

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

ARTICLE INFO

Article history:
Available online 17 February 2020

Keywords:
Fluid structure interaction
Multiphase flow
Dynamic mesh
VOF
PLIC

ABSTRACT

To investigate the fluid-rigid body interaction issues with free surfaces, a numerical approach has been developed. This algorithm is in an arbitrary Lagrangian–Eulerian description and Volume of Fluid (VOF) framework, using dynamic unstructured mesh to solve the coupled system. The fluid–solid interface uses partitioned Dirichlet–Neumann iterations with Aitken's relaxation. For the two-phase fluids part, an interface geometric reconstruction approach has been applied to accurately capture the free surfaces. This piecewise linear interface calculation (PLIC) based method uses Newton's iteration to efficiently reconstruct interfaces on an unstructured mesh, and applies an un-split scheme to transport variables. The algorithm has been successfully implemented in open source code OpenFOAM®, and was compared with the latter's built-in solver using interface compression method to deal with free surfaces. Numerical results suggest that our solver has better accuracy on multiphase flow problems, while the previous solver fails to obtain correct interfaces. Moreover, the capacity of accurately solving fluid-rigid body interaction problems with free surfaces has been achieved. Validation cases are provided for fluid–structure interaction problems with and without free surfaces, and results are in accordance with analytical and experimental data from the literature. The algorithm and solver in this paper, can be applied on fluid–structure interaction cases with free surfaces in the future, such as sloshing and water entry problems.

1. Introduction

Multiphase flow [1–3] and fluid–structure interaction (FSI) [4–6] are both challenging topics, which are extensively concerned in industrial and research areas, such as aeronautical engineering [7], biomechanics [5,6], heat transfer enhancement, bubble dynamics [8–10] and nanofluids [11–13]. The multiphase-FSI combined problems, considering fluid–structure interaction effects together with free surfaces evolutions, are of great interest and also complicated. Applications of such problems are numerous in both fundamental and applied research areas. To name a few, we have: the sloshing of liquid [14] and the damping of baffles in storage tanks [15]; the safety issues in spacecraft water landing [16]; the slamming forces on ships and other marine structures [17]; the ship-wave interaction problems [18]; the water entry of projectiles, and its under-water ballistic stability with cavitation [19].

On the numerical simulation of multiphase flow cases, the key issue is to accurately capture/track the free surfaces. The Volume of Fluid (VOF) method by Hirt et al. [20] is extensively applied in the CFD community. Other widely used

E-mail address: mingjian_li@163.com.

https://doi.org/10.1016/j.camwa.2020.01.020
0898-1221/

algorithms are level-set method by Osher et al. [21]; coupled level-set and volume of fluid (CLSVOF) method by Sussman et al. [22]; and front tracking method by Tryggvason et al. [23]. Traditional VOF method has difficulties in preserving sharp interfaces, while level-set method has the drawback of non-conservation. However, these problems have been overcome to a certain extent with the constant development of algorithms.

Numerical methods for FSI, according to spatial description of fluid domain, fall into three major categories: (1) the Eulerian framework, mainly the immersed boundary method [24–26] and the cut-cell method [27,28]; (2) the Lagrangian framework, usually mesh-free and particle based methods, e.g., the smoothed particle hydrodynamics (SPH) method [29–33], the moving particle semi-implicit (MPS) method [34–36], and the material point method (MPM) [37–39]; (3) the coupled framework, mainly the arbitrary Lagrangian–Eulerian (ALE) method [7,40–43], and the Over-set grid method [44].

For interaction problems between viscous, incompressible two-phase fluids and moving rigid bodies, which is concerned in this paper, a natural idea is to couple the VOF method and ALE description, since body-fit mesh in ALE method can explicitly distinguish the fluid–structure interface, thus avoids the complicated construction of three-phase interfaces in one computational cell. However, the inherent nonlinearity of PDEs in this problem still causes great difficulties for numerical simulations.

In this paper, we focused on the interaction issues between free surface flow and moving rigid bodies applying the coupled VOF-ALE framework. The numerical work was based on the open source code OpenFOAM® [45], which has been widely used due to its robustness and MPI feature. The VOF solver interFoam with interface compression method was built-in, however, we found it has great difficulties on free surface simulations of certain problems, other researchers also reported this problem [46]. We thus implemented our algorithm in VOF-PLIC (piecewise linear interface calculation) framework for unstructured grids to overcome this problem. The solver we developed was tested and compared with previous solver interFoam. Moreover, the VOF algorithm has been extended to an ALE framework, using dynamic mesh to simulate fluid-rigid body interaction problems with free surfaces. Numerical test cases have been presented using this solver, and suggest good accuracy and robustness.

The remainder of the paper is organized as follows. In Section 2, we present the numerical algorithm in detail, including the interface reconstruction method and fluid-rigid body interaction iterative scheme. The interface compression method is also introduced for comparison. In Section 3, validation cases are presented, including Zalesak's test and shear deformation test, to test the capability of our solver for two-phase interface capturing. A comparison has also been made to the previous solver in OpenFOAM®. Section 4 presents validation cases for both two-phase flow problem and fluid–structure interaction problem. Results suggest good accuracy of our solver, by comparing with those in the literature. In Section 5, we present application in fluid-rigid body interaction problem with free surfaces, mainly the water entry problem, to show the capability of our solver in complex coupled problems with unstructured mesh. Section 6 presents concluding remarks and discussion.

2. Numerical methodology

2.1. Mathematical formulation

The numerical methodology in this paper, is developed for fluid–structure interaction problems between two-phase fluids and moving rigid bodies. The mathematical formulations are as follows.

In the computational time domain $t \in [0, T]$, let Ω_f^t indicate the fluid spatial domain and Ω_s^t the solid region. Generally, the Lagrangian description is used for solid region and the Eulerian framework is suitable for fluid domain. However, to better describe the FSI problem in this paper's scope, the fluid computational domain is moving or deforming in accordance with the rigid body movement. Hence, an arbitrary Lagrangian–Eulerian (ALE) mapping function $\mathcal{A}^t$ is defined in the three dimensional fluid domain as

$$\mathcal{A}^t : \Omega_f^0 \to \Omega_f^t \subset \mathbb{R}^3, \quad \boldsymbol{x} = \mathcal{A}^t(\boldsymbol{X}) \in \Omega_f^t, \quad \boldsymbol{X} \in \Omega_f^0 \tag{1}$$

where Ω_f^0 and Ω_f^t respectively denote the initial and current spatial configurations, while $\boldsymbol{x}$ and $\boldsymbol{X}$ indicate coordinates in each of their coordinating systems.

Considering a viscous, incompressible, immiscible two-fluids system, with the ALE description above, the Navier–Stokes and the continuity equations are written as:

$$\rho \frac{\partial \boldsymbol{u}}{\partial t} + \rho \left(\boldsymbol{u} - \boldsymbol{u}_g\right) \cdot \nabla \boldsymbol{u} = -\nabla p + \mu \nabla^2 \boldsymbol{u} + \rho \boldsymbol{g} + \sigma \kappa \delta_s \boldsymbol{n} \quad \text{in} \quad \Omega_f^t \times [0, T] \tag{2}$$

$$\nabla \cdot \boldsymbol{u} = 0 \quad \text{in} \quad \Omega_f^t \times [0, T] \tag{3}$$

where ρ, μ, σ respectively denote the fluid density, dynamic viscosity, and surface tension coefficient. $\boldsymbol{u}$ describes the fluid velocity field, p denotes fluid pressure, and $\boldsymbol{g}$ is gravitational acceleration. In the surface tension term, $\boldsymbol{n}$ is the two-phase interface normal vector, κ is the interface curvature, and δ_s is Dirac delta function, to transform surface tension into volumetric force in a continuum surface force (CSF) model [47]. In the ALE framework, $\boldsymbol{u}_g$ is the velocity of the referential system. In fact, the referential system is usually attached to the fluid computational grids. Thus, $\boldsymbol{u}_g$ is also the grid velocity, and $\boldsymbol{u}_g = \partial \mathcal{A}^t(\boldsymbol{X})/\partial t$ is satisfied.

In a typical VOF approach, a volume fraction function $f(\boldsymbol{x}, t)$ is defined to distinguish two fluids in the mixture system. In each discretized control volume, $f \in [0, 1]$ is satisfied, and $f = 1$ means the control volume is full of phase-1 fluid. The density ρ and viscosity μ of the mixture are written as

$$\rho = f\rho_1 + (1-f)\rho_2 \tag{4}$$

$$\mu = f\mu_1 + (1-f)\mu_2 \tag{5}$$

where subscripts 1,2 indicate each phase of the fluids. The fluid property can be treated as an average of the two fluids. Without phase change, the volume fraction f is transported with the following equation in an ALE framework:

$$\frac{\partial f}{\partial t} + (\boldsymbol{u} - \boldsymbol{u}_g) \cdot \nabla f = 0. \tag{6}$$

This equation is different from a typical volume fraction advection equation $\frac{\partial f}{\partial t} + \boldsymbol{u} \cdot \nabla f = 0$ in VOF, since ALE description is used, and the velocity of referential system is taken into account.

For the solid region Ω_s^t, the Lagrangian description is applied, and the motion equation is given as

$$m_s \frac{\partial \boldsymbol{u}_s}{\partial t} + c_s \boldsymbol{u}_s = \boldsymbol{F}_f + m_s \boldsymbol{g} \quad \text{in} \quad \Omega_s^t \times [0, T] \tag{7}$$

where m_s, $\boldsymbol{u}_s$, $\boldsymbol{F}_f$ are respectively the general mass, general velocity vector and fluid general force vector. c_s is damping coefficient if needed, but is usually zero. The wording "general" means, in a 6 degrees of freedom problem, m_s will include mass and moment of inertia, $\boldsymbol{u}_s$ includes velocity and angular velocity, and $\boldsymbol{F}_f$ includes force and moment. We here focus on the rigid bodies with only translational degrees of freedom in three directions. If an extension to problems with 6 degrees of freedom is needed, either a global/local remeshing process, or a rotating spherical mesh with sliding interface is needed. Otherwise, the fluid grids will easily be tangled and distorted with a large solid rotating angle, thus jeopardize the whole simulation.

On the fluid–structure interface $\Gamma_{\text{FSI}} = \partial\Omega_f^t \cap \partial\Omega_s^t$, the velocity should be continuous, i.e. the kinematic condition is

$$\boldsymbol{u} = \boldsymbol{u}_s \quad \text{on} \quad \Gamma_{\text{FSI}}. \tag{8}$$

where $\boldsymbol{u}$ is the fluid mixture velocity vector and $\boldsymbol{u}_s$ is the solid velocity. Additionally, the forces on the fluid side and the solid side need to be balanced, i.e. the dynamic condition is

$$\boldsymbol{F}_f = \int_{\Gamma_{\text{FSI}}} \boldsymbol{\sigma}_f \cdot \boldsymbol{n} \mathrm{d}S \tag{9}$$

where the fluid total force $\boldsymbol{F}_f$ is integrated over the fluid–structure interface Γ_{FSI} with a normal vector $\boldsymbol{n}$. The fluid stress tensor is expressed as $\boldsymbol{\sigma}_f = -p\boldsymbol{I} + \mu(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T)$, with $\boldsymbol{u}$, p, μ the velocity, pressure and viscosity of the two-fluids mixture, and $\boldsymbol{I}$ the identity tensor. By applying the virtual work principle, the energy conservation can be met on the fluid–structure interface.

2.2. Algorithm for fluid, solid and their coupling equations

To solve the PDEs above, the fluid domain was solved considering solid influence as velocity boundary, and the solid domain was solved considering fluid effects as external forces. Iterations are needed in each time step, and the following numerical algorithm has been designed. The fluid part uses the Finite Volume Method (FVM) based on unstructured collocated grids. The pressure implicit split operator (PISO) algorithm by Issa et al. [48] is used to solve the velocity and pressure, with the Euler backward scheme. Applying the Gauss theorem, the Navier–Stokes equation is discretized as

$$\begin{aligned} &\frac{1}{\Delta t}\left(1.5\boldsymbol{u}_c - 2\boldsymbol{u}_c^n + 0.5\boldsymbol{u}_c^{n-1}\right) V_c + \sum_{a=1}^{na} \left(\boldsymbol{u}^n - \boldsymbol{u}_g\right)_a \cdot S_a \boldsymbol{u}_a \\ &= -\frac{1}{\rho}\left(\nabla p^n\right)_c V_c + \frac{1}{\rho}\sum_{a=1}^{na} S_a \cdot (\mu\nabla\boldsymbol{u})_a + \frac{1}{\rho}\boldsymbol{b}_c V_c \end{aligned} \tag{10}$$

where superscripts n and $n-1$ denote latest two time steps, subscripts a and c indicate cell face and center. S_a is cell face area and V_c denotes the control volume. $\boldsymbol{b}$ includes gravity and surface tension. The following equation is consequently achieved as

$$A_c^u \boldsymbol{u}_c' = \mathbf{H}\left(\boldsymbol{u}'\right) - \left(\nabla\frac{p^n}{\rho}\right)_c \tag{11}$$

where A_c^u and $\mathbf{H}\left(\boldsymbol{u}'\right)$ denote coefficients obtained after temporal and spatial discretization. The following algebraic equations are formed for velocity $\boldsymbol{u}_c'$ with 1st prediction:

$$[A]\{\boldsymbol{u}_c'\} = \{R\}. \tag{12}$$

Here, $[A]$ is a sparse matrix, $\{\boldsymbol{u}_c'\}$ is the array of velocity vectors to be solved, and $\{R\}$ denotes source term. Using the solved $\boldsymbol{u}_c'$, unsolved pressure p^* with 1st prediction, we have the following equation for velocity $\boldsymbol{u}''$ with 2nd prediction:

$$A_c^u \boldsymbol{u}_c'' = \mathbf{H}\left(\boldsymbol{u}'\right) - \left(\nabla \frac{p^*}{\rho}\right)_c. \tag{13}$$

The pressure Poisson's equation is therefore obtained as

$$\nabla \cdot \left(A_p^u \boldsymbol{u}_c'\right) = \nabla^2 \left(\frac{p^* - p^n}{\rho}\right)_c. \tag{14}$$

By solving the above equation, pressure p^* with 1st prediction is obtained, and velocity $\boldsymbol{u}''$ with 2nd prediction is then solved by Eq. (13). Similarly, we have

$$\nabla \cdot \left(A_p^u \boldsymbol{u}_c''\right) = \nabla^2 \left(\frac{p^{**} - p^*}{\rho}\right)_c. \tag{15}$$

The pressure with 2nd prediction p^{**} is solved by the above equation, and velocity with correction $\boldsymbol{u}'''$ can be get. Usually, these two corrections are enough, however, additional corrections can still be made if necessary.

To solve the solid motion equation, the temporal central differencing scheme is used. The acceleration and velocity at time level n are given as

$$\frac{\partial^2 \boldsymbol{d}_s^n}{\partial t^2} = \frac{1}{(\Delta t)^2}\left(\boldsymbol{d}_s^{n-1} - 2\boldsymbol{d}_s^n + \boldsymbol{d}_s^{n+1}\right), \quad \frac{\partial \boldsymbol{d}_s^n}{\partial t} = \frac{1}{2\Delta t}\left(-\boldsymbol{d}_s^{n-1} + \boldsymbol{d}_s^{n+1}\right) \tag{16}$$

where $\boldsymbol{d}_s$ is the solid displacement. The displacement at next time level $\boldsymbol{d}_s^{n+1}$ can be explicitly obtained by the following equation with a second-order accuracy:

$$\left[\frac{1}{(\Delta t)^2} m_s + \frac{1}{2\Delta t} c_s\right] \boldsymbol{d}_s^{n+1} = \boldsymbol{F}_f^n + m_s \boldsymbol{g} + \frac{2m_s}{(\Delta t)^2} \boldsymbol{d}_s^n - \left[\frac{1}{(\Delta t)^2} m_s - \frac{1}{2\Delta t} c_s\right] \boldsymbol{d}_s^{n-1}. \tag{17}$$

The fluid velocity $\boldsymbol{u}$ and solid velocity $\partial \boldsymbol{d}_s / \partial t$ on the FSI boundary are actually not matched, but staggered during time advancing. This may cause severe numerical instability [49,50], and put great challenges to FSI algorithms. To guarantee the kinematic condition (8) in each time level, iterative methods can be used, one promising way is the quasi-Newton technique [51,52], which has been shown with high efficiency. In this paper, an implicit scheme with Aitken's dynamic relaxation [53] is applied. At time level n, several sub-steps are taken, and fluid boundary displacement (the same as solid displacement) at kth sub-step is

$$\boldsymbol{d}_s^{n+1,k} = \boldsymbol{d}_s^{n+1,k-1} + \omega^{n,k-1} \boldsymbol{r}^{n,k-1}. \tag{18}$$

where ω is a dynamic factor in the following form:

$$\omega_\alpha^{n,k} = -\omega_\alpha^{n,k-1} \frac{\left(\boldsymbol{r}^{n,k-1}\right)^T \left(\boldsymbol{r}^{n,k} - \boldsymbol{r}^{n,k-1}\right)}{\left(\boldsymbol{r}^{n,k} - \boldsymbol{r}^{n,k-1}\right)^T \left(\boldsymbol{r}^{n,k} - \boldsymbol{r}^{n,k-1}\right)} \tag{19}$$

and the residual is $\boldsymbol{r}^{n,k} = \boldsymbol{d}_s^{n+1,k+1} - \boldsymbol{d}_s^{n+1,k}$.

The grid velocity on FSI boundary takes the form

$$\boldsymbol{u}_g^{n+1,k} = \left(\boldsymbol{d}_s^{n+1,k} - \boldsymbol{d}_s^n\right)/\Delta t, \quad \text{on} \quad \Gamma_{\text{FSI}}. \tag{20}$$

For FSI problems with far-field boundaries, $\boldsymbol{u}_g$ can be directly applied to Eq. (2), which means the whole fluid computational domain will be moved with the referential coordinating system. This approach avoids mesh distortion and becomes very robust and efficient if the near-structure fluid field is the main concern. For other problems whose boundary effects must be taken into account, the boundary should remain still, and the computational domain must absorb the deformation caused by solid motion. A Laplacian equation $\nabla \cdot \left(\varsigma \nabla \boldsymbol{u}_g\right) = 0$ can be solved to spread the grid velocity into the whole fluid region, with a tunable coefficient ς to help control the quality of deformed grids. After determining $\boldsymbol{u}_g$ in the whole domain, an ALE mapping is carried out as

$$\mathcal{A}^t(\boldsymbol{X}) = \boldsymbol{X} + \boldsymbol{u}_g \Delta t. \tag{21}$$

In the following parts, we use both methods mentioned above. As for cases e.g. water entry problems, the 1st kind of mesh motion is used, since the water can be seen as a semi-infinite domain and only fluid fields in the vicinity of solid need to be considered. However, for problems such as the vibrating cylinder in a fluid container, the 2nd method must be applied, since the boundary effects are crucial in this case.

2.3. Interface compression method implemented in OpenFOAM®

In this part, to be compared with our approach, we first introduce the interface compression method [54,55], which has been adopted in OpenFOAM® [45]. In a two-phase flow problem, instead of constructing a sharp interface between the two fluids, the interface compression method smears out the interface. However, this effect is limited to a certain region, say, a few computational cells, which means the interface is "compressed" in a sense.

In one computational cell, let $\boldsymbol{u}_i$ $(i = 1, 2)$ indicate velocity of each fluid and $f \in [0, 1]$ indicate the volume fraction of the first fluid, the mixture average velocity $\boldsymbol{u}$ is then defined as

$$\boldsymbol{u} = f\boldsymbol{u}_1 + (1-f)\boldsymbol{u}_2. \tag{22}$$

Additionally, a relative velocity $\boldsymbol{u}_r$ is defined as the velocity difference between the two fluids:

$$\boldsymbol{u}_r = \boldsymbol{u}_1 - \frac{1}{1-f}(\boldsymbol{u} - f\boldsymbol{u}_1). \tag{23}$$

The volume fraction transportation equation for the first fluid in a general form $\frac{\partial f}{\partial t} + \nabla \cdot (f\boldsymbol{u}_1) = 0$ can then be written as

$$\frac{\partial f}{\partial t} + \boldsymbol{u} \cdot \nabla f + \nabla \cdot [\boldsymbol{u}_r f(1-f)] = 0. \tag{24}$$

This equation is the same with the standard volume fraction transportation equation (6), except for the addition of an interface compression term $\nabla \cdot [\boldsymbol{u}_r f(1-f)]$. To be noted, this additional term is only valid when $f \in (0, 1)$, which is in the vicinity of the two-phase interface. This prevents excessive diffusion of the interface. Eq. (24) can be solved without explicit expression of each fluid's velocity $\boldsymbol{u}_i$, but only with the mixture velocity $\boldsymbol{u}$, if the relative velocity $\boldsymbol{u}_r$ is numerically given in the form

$$(\boldsymbol{u}_r)_f = \boldsymbol{n}_f \min\left[c_\gamma \left| \frac{\phi}{|\boldsymbol{S}|} \right|, \left| \frac{\phi}{|\boldsymbol{S}|} \right|_{\max} \right] \tag{25}$$

where ϕ is the velocity flux, $\boldsymbol{S}$ is the cell face area with orientation, and c_γ is a predefined coefficient. The interface normal vector $\boldsymbol{n}_f$ is calculated by

$$\boldsymbol{n}_f = \frac{(\nabla f)_f}{\left|(\nabla f)_f + \delta\right|} \tag{26}$$

where δ is a small coefficient preventing float number overflow. Eq. (24) is solved by a multidimensional universal limiter solver (MULES), during which process, coefficient c_γ tunes the interface diffusion/compression effect. A smaller c_γ leads to good numerical robustness, however, the interface will be greatly diffused. Contrarily, a large c_γ helps keep a sharp interface, but numerical instability occurs from time to time. Generally, $c_\gamma = 1$ is recommended in the OpenFOAM® [45] official documents, and it will be adopted in the following part, to compare the interface compression method with our interface geometric reconstruction approach.

2.4. Advection algorithm with interface geometric reconstruction

The accurate simulation of two-phase interface evolution is always a great challenge to numerical algorithms. For a typical Eulerian approach, the calculation of free surfaces usually includes two steps: (1) to reconstruct an accurate interface, and to determine its position and orientation; (2) to correctly calculate the flux across cells, with the two-phase interface taken into account. The piecewise linear interface calculation (PLIC) method is a promising approach, which generally uses an oriented plane to simulate interface in one cell, and then takes several dimensional split advection steps.

Let us consider a control volume V_c cut by a plane representing the two-phase interface (Fig. 1), the plane can be expressed as

$$\boldsymbol{n}_f \cdot \boldsymbol{x} + q = 0 \tag{27}$$

where $\boldsymbol{n}_f$ is the normal vector of the plane, and q is a constant to be determined. At the beginning of current time level, the volume fraction is already known. The volume of phase-1 fluid formed by Eq. (27) can therefore be written as

$$V(q) = V_c f^n \tag{28}$$

Since the normal vector has already be obtained by the volume fraction gradient Eq. (26), the problem of constructing an interface within a cell becomes the solving of q. Scardovelli et al. [56] analytically derived the plane factor q with given f in a cube, and it was then extended to parallelepiped. However, for general hexahedron, or polyhedron, which is commonly seen in unstructured mesh, analytical solution is not available.

For a general convex polyhedron, Eq. (28) is a monotonous function of q. We can thus use iterative method to obtain q. Generally, bisection method [57] is used. Here, we use Newton's iteration to get better convergence. Considering the

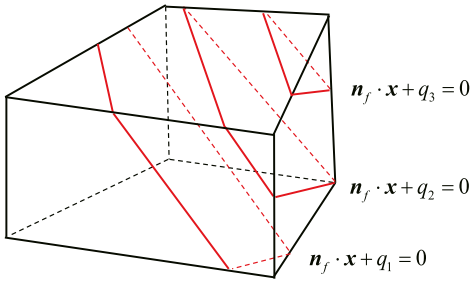

Fig. 1. A computational cell cut by a group of trial planes.

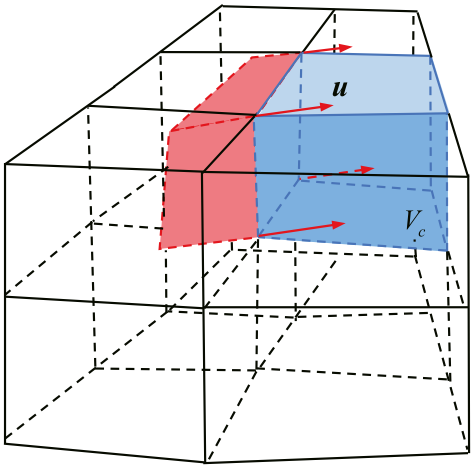

Fig. 2. Lagrangian backtracking method to obtain flux cross cells. Blue polyhedron is the control volume to be solved, while red polyhedron is created by backtracking using velocity. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

volume of phase-1 fluid is the volumetric integration of the sub-cell cut by the plane, its derivative is thus the area of the section cut by the interface. The Newton's iteration is thus expressed as

$$q^{k+1} = q^k - \frac{V_c(q^k) - V_c f^n}{S_e(q^k)} \tag{29}$$

where k denotes the number of sub-iterations, and S_e is the area of section.

After determining the interface position, flux need to be calculated. Dimension split method is usually applied due to its concise form, e.g. Cifani et al. [46]. However, this method is not strictly conservative and has difficulties when extending to unstructured mesh. Here, a Lagrangian backtracking method with un-split scheme is applied. As in Fig. 2, the blue cell is control volume which needs to be solved. By using its nodal velocities and time step, a Lagrangian backtracking polyhedron is formed. For cases with CFL number < 1, it can be guaranteed that the polyhedron will not cover areas beyond the neighbor cells. In this case, for 2D problems, 8 neighbor cells are needed as stencils, and for 3D problems, 26 neighbor cells are necessary. The polyhedron is then decomposed into simplices, with its face triangulated. This process can use a classical clipping and capping method [58], as well as improved methods with better efficiency [59,60]. Here, we use the clipping and capping framework to obtain flux considering two-phase interfaces. Results are presented in the following parts.

3. Advection validation cases

3.1. Rotating disk with a slot

Zalesak's test [61,62] is one of the most classical validation cases for two-phase problems. The case considers two immiscible fluids, one of which has the shape of a slotted disk, and the other is a square. Under a predefined velocity field, the slotted disk consequently rotates for a certain time, and will hopefully return to its original position, maintaining its shape unchanged. However, capturing the two-phase interface accurately is definitely non-trivial, especially in Eulerian

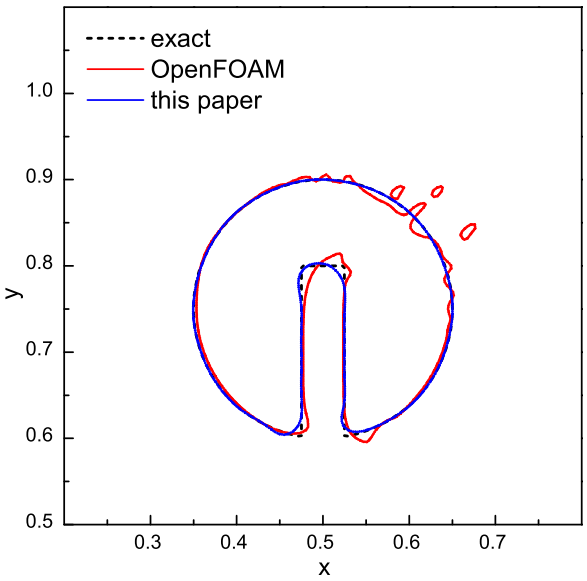

Fig. 3. Zalesak's test: comparison between the final shape solved by OpenFOAM®, and the shape calculated by our solver. The profile lines were extracted by a volume fraction of 0.5.

Table 1
Zalesak's test: comparison of accumulated error.

	Error by Eq. (31)
Rudman et al. [62]	1.09×10^{-2}
Scardovelli et al. [63]	1.00×10^{-2}
Cifani et al. [46]	1.36×10^{-2}
OpenFOAM®. [45]	7.03×10^{-2}
Solver in this paper	1.92×10^{-2}

frameworks. This case therefore becomes very suitable to test the capability of a two-phase solver, since it excludes the error from Navier–Stokes equation, and focuses on the solution of advection equation (6).

In this case, the spatial computational domain is $[0, 1] \times [0, 1]$, and the disk has a radius of 0.15, with its center $(0.5, 0.75)$. The slot has a dimension of $[0.475, 0.525] \times [0.6, 0.8]$, and the predefined velocity field is

$$\begin{aligned} u &= -\omega_c (y - y_c) \\ v &= \omega_c (x - x_c) \end{aligned} \tag{30}$$

where ω_c is the constant angular velocity, and (x_c, y_c) is the center of the fluid domain. u, v are fluid velocities in x and y directions. The spatial domain was discretized into 200×200 cells. Various cases were calculated with a maximum Courant number (based on the maximum velocity magnitude) from 0.4 to 0.8, and the temporal convergence has been achieved. For the following comparison, 16,000 time steps were used to test the accumulated error, which is defined as

$$E_{\text{Zalesak}} = \frac{\sum_i^{N_{\text{cell}}} \left| f_i - \bar{f}_i \right|}{\sum_i^{N_{\text{cell}}} \bar{f}_i} \tag{31}$$

where f_i is the final volume fraction in the ith cell, $\bar{f}_i$ is the initial volume fraction in each cell, and N_{cell} is the total cells number.

In Fig. 3, the final shapes calculated by the "interFoam" solver in OpenFOAM® and by our solver were presented. It is suggested that the original solver in OpenFOAM® has failed to maintain an accurate shape after certain advection steps, moreover, to increase temporal and spatial resolution will not help. On the other hand, despite a little smoothing in the sharp corners, the solver we developed can accurately preserve the initial configuration.

In Table 1, the accumulated errors calculated by Eq. (31) are presented. The results were from our solver, the interFoam solver and the algorithms in the literature. It is suggested that our method has similar accuracy with those in the literature, and the error of OpenFOAM® is the largest among these results.

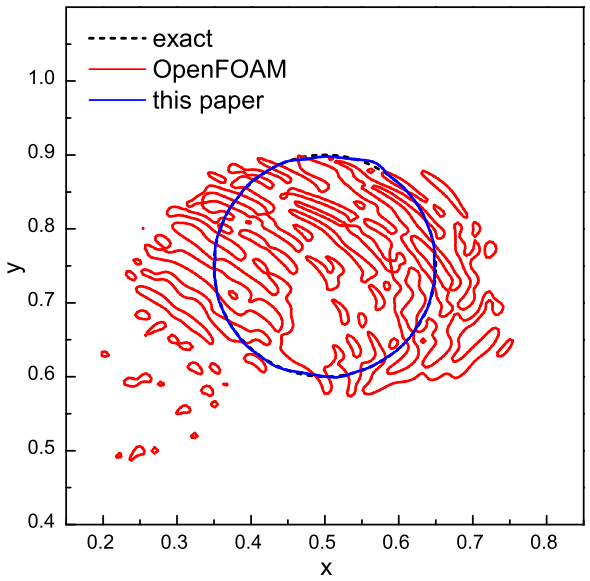

Fig. 4. Shear deformation test: comparison of the final shapes at $t = T$, calculated by OpenFOAM and by our solver. The contour lines were extracted by a volume fraction of 0.5.

3.2. Shear deformation of two-phase fluids

The following test of a fluid disk with shear deformation was first proposed by Leveque et al. [64], and then applied by many researchers [64–66], to test the interface capturing/tracking capability.

In the spatial fluid domain $[0, 1] \times [0, 1]$, a fluid disk in another phase is located, with a radius of 0.15 and its center $(0.5, 0.75)$. A predefined velocity field is applied to the two-phase system as

$$\begin{aligned} u &= \sin(2\pi y)\sin^2(\pi x)\cos(\pi t/T) \\ v &= -\sin(2\pi x)\sin^2(\pi y)\cos(\pi t/T) \end{aligned} \tag{32}$$

where u, v are fluid velocities in x and y directions. T is the period of the velocity field, and $T = 8$ is used here. Comparing with Zalesak's test, in this case, the fluid disk will not maintain its shape at all time, but will be greatly stretched in $t \in (0, T/2)$. When $t \in [T/2, T)$, the velocity field reverses and the fluid disk will gather again and finally return to its initial shape at $t = T$.

Fig. 4 provides the final shapes calculated by our solver, and by OpenFOAM®. The exact shape is also presented for comparison. This case was solved on 256×256 grids. As can be seen, our solver preserved the interface shape, while OpenFOAM failed to maintain the interfacial continuity. Fig. 5 further provides the profiles at time $t = T/2$, when the fluid disk suffers the largest deformation. Results suggest that our solver can obtain good interfacial accuracy, while OpenFOAM presents broken liquid drops. The failure of interFoam is possibly caused by the lack of geometric reconstruction of interface, and the inaccurate transport of volume fraction during free surface evolution. The algorithm in this work, on the other hand, dealing with the interface reconstruction carefully in each time step, can thus obtain accurate results.

4. Violent free surface flow and fluid–structure interaction cases

In this section, we will present more validation cases, including pure free surface flow problem, and pure fluid-rigid body interaction problem, to test the algorithm and solver in this paper.

4.1. Dam break problem

Dam break problems were studied by many researchers [38,67–69], by both numerical and experimental approaches. This problem has the feature of violent free surfaces, therefore is appropriate to test multiphase solvers. Here, we compared our solver and OpenFOAM®, with the experimental results reported by Lobovsky et al. [67]. To reprise this problem, dimensions are in accordance with the experimental setup. As in Fig. 6, the domain is $[0, 1.61] \times [0, 0.6]$, and an initially quiescent water block has the dimensions $[1.01, 1.61] \times [0, 0.3]$. Besides, a pressure sensor is located at $[0, 0.003]$ (units in meter). Material properties of water are $\rho_w = 1 \times 10^3$ kg m^{-3} and $\mu_w = 1.33 \times 10^{-3}$ Pa s. The air has $\rho_a = 1$ kg m^{-3} and $\mu_a = 1.8 \times 10^{-5}$ Pa s. Surface tension coefficient between air and water is 7.2×10^{-2} N m^{-1}.

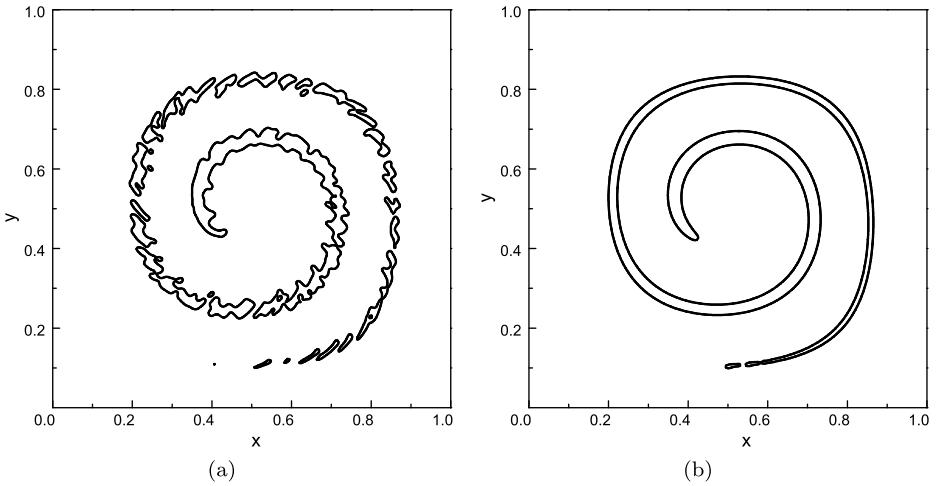

Fig. 5. Profiles of the two-phase interface at time $t = T/2$ in the shear deformation test case: (a) calculated by interFoam solver in OpenFOAM®; (b) calculated by solver in this paper. Computations were carried out on 256 × 256 cells, and the profiles were extracted by a volume fraction of 0.5.

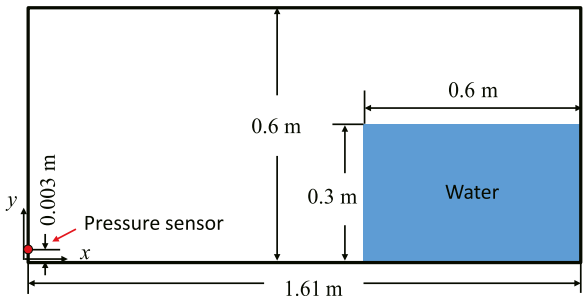

Fig. 6. Dimensions of the dam break problem. The water block is initially still and will collapse as the simulation starts. Lobovsky et al. [67] have reported experimental pressure data by a sensor on the left wall. The computational dimensions are in accordance with their model for comparison.

Fig. 7 shows the evolution of the free surface in detail. The profile of the two-phase interface at time 0.16, 0.28, 0.37, 0.45, 0.57, 0.86, 1.02 (units in second) are presented. Meanwhile, the experimental pictures by Lobovsky et al. are provided for comparison. Results suggest that the free surface solved by our solver is similar with the experimental results, especially at time 0.57, 0.86, and 1.02 s.

In Fig. 8, a quantitative comparison has been made. Lobovsky et al. carried out 100 tests and concluded a range of the measured pressure on the sensor. We provides pressure results both by our solver and by interFoam. It can be seen, that our pressure is in good agreement with the experiments, however, OpenFOAM failed to get accurate pressure, since the free surfaces are not reliable in the first place.

4.2. Fluid-rigid body interaction validation

To validate the fluid-rigid body interaction capability of the solver, we present a case considering a vibrating cylinder in a fluid container. This problem has been studied in [70–72] by different numerical schemes, and becomes a benchmark case. Chen et al. [73] presented the analytical solution of this problem and Li et al. [74] made correction and extension to this analytical solution. We here calculated this case by our solver with dynamic mesh and ALE method mentioned above. Following the parameters in the literature, the cylinder mass is $m_s = 3.408 \times 10^{-1}$ kg m^{-1}, the stiffness of spring attached to the cylinder is $k_s = 3461.13$ kg m^{-1} s^{-2}. The cylinder has a radius of $R_1 = 6.35 \times 10^{-3}$ m with an initial displacement $0.02R_1$, the fluid container has a radius of $5R_1$. The fluid is mineral oil and has $\rho_o = 0.935 \times 10^3$ kg m^{-3} and $\mu_o = 4.1 \times 10^{-2}$ Pa s. The calculation was carried out on 12 × 12 cells and 200 × 200 cells (in radial and circumferential directions). Fig. 9 presents the displacement time-history by these two sets of meshes. It is suggested that 200 × 200 grids can achieve accurate results in good agreement with the analytical solution [74].

Fig. 10 shows the pressure field distribution, as well as the streamlines at time $t = 0.6$ s. Compared with results in the literature [70–72,74], it is suggested that the pressure and streamlines are accurate.

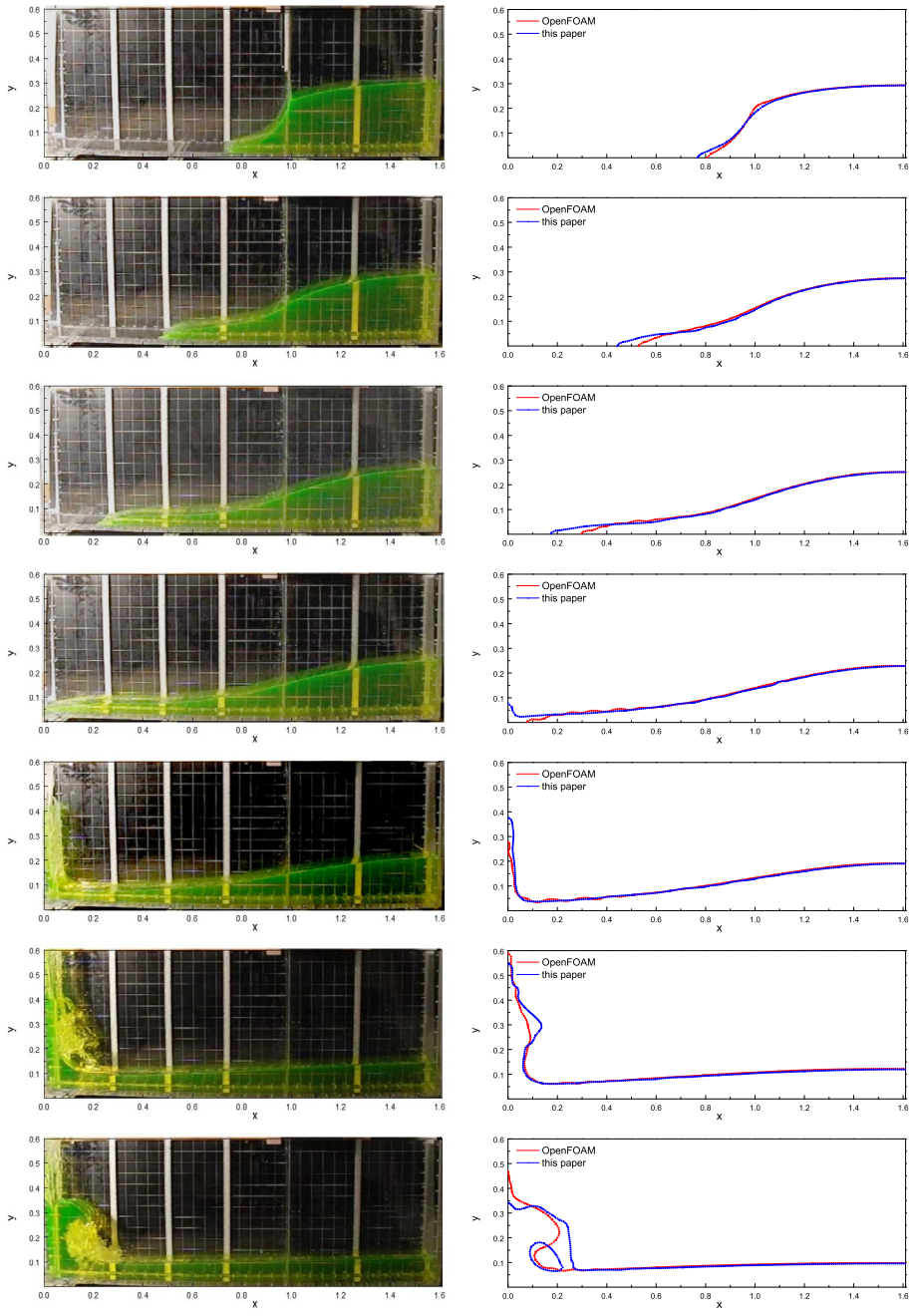

Fig. 7. The evolution of free surface profiles in a dam break problem. Results by OpenFOAM® and by our solver are presented at times 0.16, 0.28, 0.37, 0.45, 0.57, 0.86, 1.02 (units in second), and are compared with experimental results.
Source: Experimental pictures are from [67].

5. Application in water entry problem

Water entry problems combine violent free surface flow and fluid-rigid body interaction issues. The water entry of a wedge has been investigated experimentally and numerically [75,76] by researchers. Here, this case was calculated by our solver. Using body-fit unstructured dynamic mesh, the fluid domain is described in an ALE framework, and the free surfaces were reconstructed by the geometric un-spit scheme introduced above.

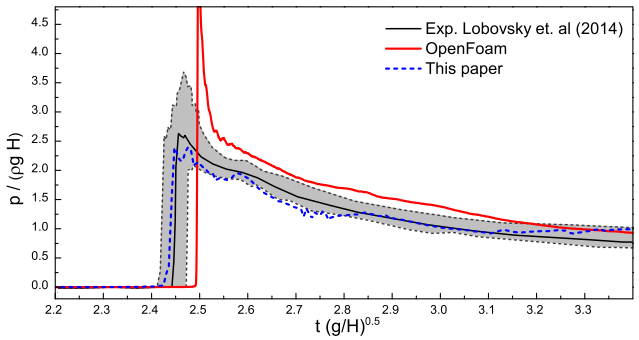

Fig. 8. Comparison of pressure time histories: the central black line indicates the median of 100 tests by Lobovsky et al. [67], and the gray area presents the lower and upper bounds of 2.5% and 97.5% estimation in the their experiments. The red line was solved by OpenFOAM®, and the blue dashed line was calculated by the solver developed in this paper. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

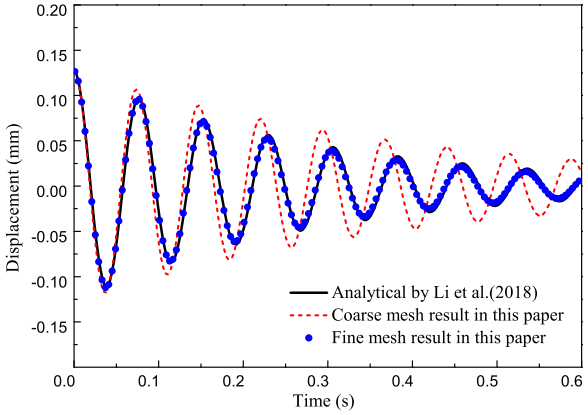

Fig. 9. Displacement comparison between our solver and analytical solution by Li et al. [74].

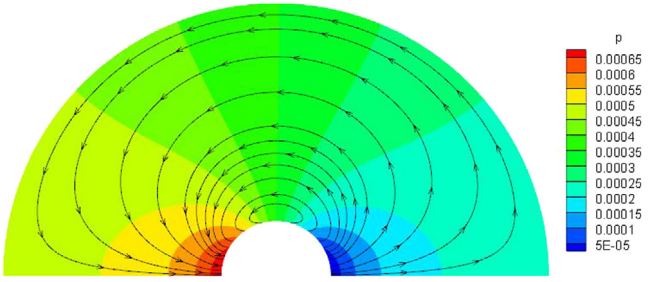

Fig. 10. The pressure field and streamlines at time $t = 0.6$ s, considering a vibrating rigid body in mineral oil. 1/2 model is presented here due to symmetry.

Following Zhao et al.'s experimental work, the computational model has been established with dimensions in Fig. 11. The width of the wedge is 0.5 m and the total mass is 241 kg. Material properties of water are $\rho_w = 1 \times 10^3$ kg m^{-3} and $\mu_w = 1.33 \times 10^{-3}$ Pa s. The air has $\rho_a = 1$ kg m^{-3} and $\mu_a = 1.8 \times 10^{-5}$ Pa s. Surface tension coefficient between air and water is 7.2×10^{-2} N m^{-1}.

Fig. 13 shows pressure field and free surface evolution at time 0.005 s, 0.016 s, and 0.02 s. By comparing with the SPH simulation results by Oger et al. [76] and MPM simulation by Chen et al. [37], it can be found that our solver can obtain more smooth pressure spatial fields (Pictures of their pressure results can be found in the cited literature). This is due to the non-physical oscillation mainly caused by the weakly compressible assumption taken by their investigations. In

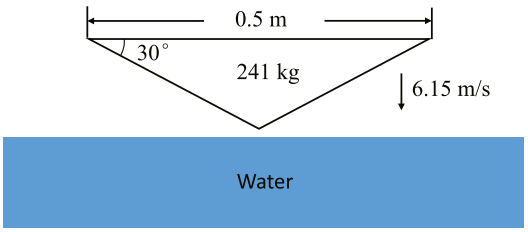

Fig. 11. The dimensions of the water entry problem of a wedge.

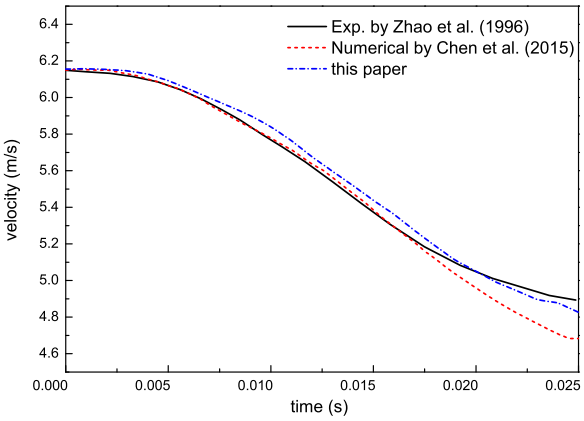

Fig. 12. Comparison of wedge velocities during water entry.

Fig. 12, we compared the wedge velocity calculated by our solver, by Chen et al.'s numerical simulation and by Zhao et al.'s experimental work. Results suggest good agreement between our results and the experimental data in the literature.

6. Concluding remarks

For fluid-rigid body interaction problems with free surfaces, an ALE-VOF based algorithm has been developed and successfully implemented in the open source code OpenFOAM®.

The PLIC based method using Newton's iteration can efficiently reconstruct interfaces in unstructured mesh. Together with the conservative un-split geometric flux calculation scheme, the free surfaces can be accurately captured and evolved. The ALE method was coupled with the VOF-PLIC solver, with an iterative scheme in dynamic mesh. Numerical results suggest good accuracy and robustness of this solver in problems with free surfaces and fluid-rigid body interaction.

One possible improvement of this solver, lies in the time efficiency. Currently, a typical FSI case with free surface needs 10–12 iterations per time step in average. To obtain accurate free surface, high spatial resolution is needed, which will in turn increase the time consumption. More efficient iterative schemes are under consideration such as the quasi-Newton technique [51,52]. Also, a Lagrangian way to trace the free surface, e.g. the SPH method, may increase efficiency. Moreover, to validate the numerical simulation with more details, comparisons with experiments considering breaking waves [77] are in progress. Again, the current work focuses on an accurate Eulerian approach to simulate FSI cases with free surface. The algorithm and solver, are already applied to engineering problems, and are expected to be used in more areas considering the coupled effects of FSI and free surfaces.

CRediT authorship contribution statement

Ming-Jian Li: Conceptualization, Investigation, Methodology, Software, Writing - original draft, Writing - review & editing.

Acknowledgment

The authors acknowledge the support from China Postdoctoral Science Foundation, under Grant No. 2019M650646.

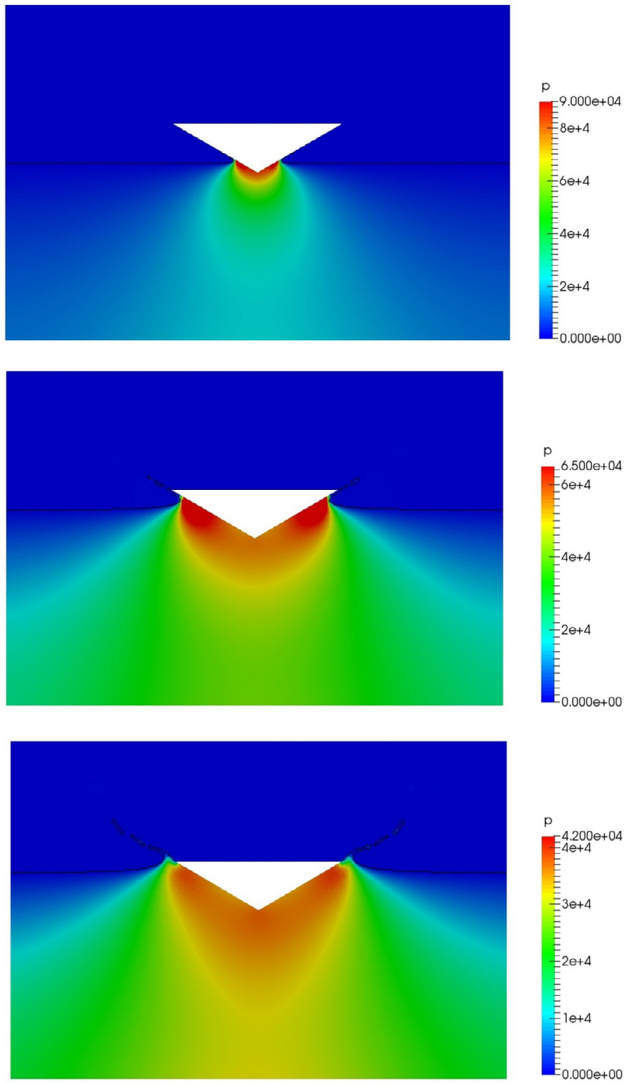

Fig. 13. Pressure distribution and free surface profiles at time 0.005, 0.016, 0.02 (units in second) in the water entry problem.

References

[1] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annu. Rev. Fluid Mech. 31 (1) (1999) 567–603.
[2] S. Balachandar, J.K. Eaton, Turbulent dispersed multiphase flow, Annu. Rev. Fluid Mech. 42 (2010) 111–133.
[3] M. Sussman, A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles, J. Comput. Phys. 187 (1) (2003) 110–136.
[4] M.J. Shelley, J. Zhang, Flapping and bending bodies interacting with fluid flows, Annu. Rev. Fluid Mech. 43 (2011) 449–465.
[5] R. Mittal, J.H. Seo, V. Vedula, Y.J. Choi, H. Liu, H.H. Huang, S. Jain, L. Younes, T. Abraham, R.T. George, Computational modeling of cardiac hemodynamics: current status and future outlook, J. Comput. Phys. 305 (2016) 1065–1082.
[6] T.E. Tezduyar, M. Schwaab, S. Sathe, Sequentially-coupled arterial fluid–structure interaction (SCAFSI) technique, Comput. Methods Appl. Mech. Engrg. 198 (45) (2009) 3524–3533.
[7] C. Farhat, V.K. Lakshminarayan, An ALE formulation of embedded boundary methods for tracking boundary layers in turbulent fluid–structure interaction problems, J. Comput. Phys. 263 (2014) 53–70.
[8] S. Shu, N. Yang, Direct numerical simulation of bubble dynamics using phase-field model and lattice Boltzmann method, Ind. Eng. Chem. Res. 52 (33) (2013) 11391–11403.
[9] S. Shu, D. Vidal, F. Bertrand, J. Chaouki, Multiscale multiphase phenomena in bubble column reactors: A review, Renew. Energy 141 (2019) 613–631.
[10] S. Shu, J. Zhang, N. Yang, Gpu-accelerated transient lattice Boltzmann simulation of bubble column reactors, Chem. Eng. Sci. (2019) 115436.
[11] Q. Xiong, S.-C. Kong, High-resolution particle-scale simulation of biomass pyrolysis, ACS Sustain. Chem. Eng. 4 (10) (2016) 5456–5461.
[12] A. Izadi, M. Siavashi, Q. Xiong, Impingement jet hydrogen, air and cuh2o nanofluid cooling of a hot surface covered by porous media with non-uniform input jet velocity, Int. J. Hydrogen Energy 44 (30) (2019) 15933–15948.

[13] M.V. Bozorg, M.H. Doranehgard, K. Hong, Q. Xiong, Cfd study of heat transfer and fluid flow in a parabolic trough solar receiver with internal annular porous structure and synthetic oil–al2o3 nanofluid, Renew. Energy 145 (2020) 2598–2614.
[14] J. Cho, H. Lee, Numerical study on liquid sloshing in baffled tank by nonlinear finite element method, Comput. Methods Appl. Mech. Engrg. 193 (23) (2004) 2581–2598.
[15] Z. Hu, W. Tang, H. Xue, X. Zhang, A SIMPLE-based monolithic implicit method for strong-coupled fluid–structure interaction problems with free surfaces, Comput. Methods Appl. Mech. Engrg. 299 (2016) 90–115.
[16] C. Seddon, M. Moatamedi, Review of water entry with applications to aerospace structures, Int. J. Impact Eng. 32 (7) (2006) 1045–1067.
[17] F. Dias, J.-M. Ghidaglia, Slamming: Recent progress in the evaluation of impact pressures, Annu. Rev. Fluid Mech. 50 (2018) 243–273.
[18] K. Shibata, S. Koshizuka, M. Sakai, K. Tanizawa, Lagrangian simulations of ship-wave interactions in rough seas, Ocean Eng. 42 (2012) 13–25.
[19] T.T. Truscott, B.P. Epps, J. Belden, Water entry of projectiles, Annu. Rev. Fluid Mech. 46 (2014) 355–378.
[20] 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.
[21] 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.
[22] 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.
[23] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, J. Comput. Phys. 100 (1) (1992) 25–37.
[24] C.S. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (2) (1972) 252–271.
[25] R. Mittal, G. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid Mech. 37 (2005) 239–261.
[26] 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.
[27] D.K. Clarke, H. Hassan, M. Salas, Euler calculations for multielement airfoils using cartesian grids, AIAA J. 24 (3) (1986) 353–358.
[28] V. Pasquariello, G. Hammerl, F. Örley, S. Hickel, C. Danowski, A. Popp, W.A. Wall, N.A. Adams, A cut-cell finite volume–finite element coupling approach for fluid–structure interaction in compressible flow, J. Comput. Phys. 307 (2016) 670–695.
[29] R.A. Dalrymple, B. Rogers, Numerical modeling of water waves with the SPH method, Coast. Eng. 53 (2–3) (2006) 141–147.
[30] M. Gomez-Gesteira, B.D. Rogers, R.A. Dalrymple, A.J. Crespo, State-of-the-art of classical SPH for free-surface flows, J. Hydraul. Res. 48 (S1) (2010) 6–27.
[31] T. Long, D. Hu, D. Wan, C. Zhuang, G. Yang, An arbitrary boundary with ghost particles incorporated in coupled fem–sph model for fsi problems, J. Comput. Phys. 350 (2017) 166–183.
[32] C. Antoci, M. Gallati, S. Sibilla, Numerical simulation of fluid–structure interaction by sph, Comput. Struct. 85 (11–14) (2007) 879–890.
[33] 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.
[34] M. Tanaka, T. Masunaga, Stabilization and smoothing of pressure in MPS method by quasi-compressibility, J. Comput. Phys. 229 (11) (2010) 4279–4290.
[35] S.-C. Hwang, A. Khayyer, H. Gotoh, J.-C. Park, Development of a fully lagrangian mps-based coupled method for simulation of fluid–structure interaction problems, J. Fluids Struct. 50 (2014) 497–511.
[36] Y.-x. Zhang, D.-c. Wan, H. Takanori, Comparative study of mps method and level-set method for sloshing flows, J. Hydrodyn. B 26 (4) (2014) 577–585.
[37] 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.
[38] 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.
[39] Y. Liang, X. Zhang, Y. Liu, An efficient staggered grid material point method, Comput. Methods Appl. Mech. Engrg. 352 (2019) 85–109.
[40] M. Bukač, S. Čanić, B. Muha, A partitioned scheme for fluid–composite structure interaction problems, J. Comput. Phys. 281 (2015) 493–517.
[41] Q. Zhang, T. Hisada, Studies of the strong coupling and weak coupling methods in FSI analysis, Internat. J. Numer. Methods Engrg. 60 (12) (2004) 2013–2029.
[42] K.-J. Bathe, G.A. Ledezma, Benchmark problems for incompressible fluid flows with structural interactions, Comput. Struct. 85 (11) (2007) 628–644.
[43] J.W. Banks, W.D. Henshaw, D.W. Schwendeman, An analysis of a new stable partitioned algorithm for FSI problems. Part I: Incompressible flow and elastic solids, J. Comput. Phys. 269 (2014) 108–137.
[44] J.L. Steger, J.A. Benek, On the use of composite grid schemes in computational aerodynamics, Comput. Methods Appl. Mech. Engrg. 64 (1–3) (1987) 301–320.
[45] OpenFOAM: The open source CFD toolbox, http://www.openfoam.com.
[46] P. Cifani, W. Michalek, G. Priems, J.G. Kuerten, C. van der Geld, B.J. Geurts, A comparison between the surface compression method and an interface reconstruction method for the vof approach, Comput. & Fluids 136 (2016) 421–435.
[47] J.U. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modeling surface tension, J. Comput. Phys. 100 (2) (1992) 335–354.
[48] R.I. Issa, Solution of the implicitly discretised fluid flow equations by operator-splitting, J. Comput. Phys. 93 (2) (1991) 388–410.
[49] C. Michler, E. Van Brummelen, S. Hulshoff, R. De Borst, The relevance of conservation for stability and accuracy of numerical methods for fluid–structure interaction, Comput. Methods Appl. Mech. Engrg. 192 (37) (2003) 4195–4215.
[50] J. Degroote, J. Vierendeels, Multi-level quasi-Newton coupling algorithms for the partitioned simulation of fluid–structure interaction, Comput. Methods Appl. Mech. Engrg. 225 (2012) 14–27.
[51] A.E. Bogaers, S. Kok, B.D. Reddy, T. Franz, An evaluation of quasi-newton methods for application to fsi problems involving free surface flow and solid body contact, Comput. Struct. 173 (2016) 71–83.
[52] J. Degroote, A. Souto-Iglesias, W. Van Paepegem, S. Annerel, P. Bruggeman, J. Vierendeels, Partitioned simulation of the interaction between an elastic structure and free surface flow, Comput. Methods Appl. Mech. Engrg. 199 (33–36) (2010) 2085–2098.
[53] U. Küttler, W.A. Wall, Fixed-point fluid–structure interaction solvers with dynamic relaxation, Comput. Mech. 43 (1) (2008) 61–72.
[54] H. Rusche, Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions (Ph.D. thesis), Imperial College London (University of London), 2003.
[55] E. Berberović, N.P. van Hinsberg, S. Jakirlić, I.V. Roisman, C. Tropea, Drop impact onto a liquid layer of finite thickness: Dynamics of the cavity evolution, Phys. Rev. E 79 (3) (2009) 036306.
[56] R. Scardovelli, S. Zaleski, Analytical relations connecting linear interfaces and volume fractions in rectangular grids, J. Comput. Phys. 164 (1) (2000) 228–237.
[57] H.T. Ahn, M. Shashkov, Multi-material interface reconstruction on generalized polyhedral meshes, J. Comput. Phys. 226 (2) (2007) 2096–2132.
[58] M.B. Stephenson, H.N. Christiansen, A polyhedron clipping and capping algorithm and a display system for three dimensional finite element models, ACM SIGGRAPH Comput. Graph. 9 (3) (1975) 1–16.
[59] M. Owkes, O. Desjardins, A computational framework for conservative, three-dimensional, unsplit, geometric transport with application to the volume-of-fluid (vof) method, J. Comput. Phys. 270 (2014) 587–612.

[60] X. Chen, X. Zhang, Z. Jia, A robust and efficient polyhedron subdivision and intersection algorithm for three-dimensional mmale remapping, J. Comput. Phys. 338 (2017) 1–17.
[61] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, J. Comput. Phys. 31 (3) (1979) 335–362.
[62] M. Rudman, Volume-tracking methods for interfacial flow calculations, Internat. J. Numer. Methods Fluids 24 (7) (1997) 671–691.
[63] R. Scardovelli, S. Zaleski, Interface reconstruction with least-square fit and split eulerian–lagrangian advection, Internat. J. Numer. Methods Fluids 41 (3) (2003) 251–274.
[64] R.J. LeVeque, High-resolution conservative algorithms for advection in incompressible flow, SIAM J. Numer. Anal. 33 (2) (1996) 627–665.
[65] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, J. Comput. Phys. 141 (2) (1998) 112–152.
[66] D. Enright, R. Fedkiw, J. Ferziger, I. Mitchell, A hybrid particle level set method for improved interface capturing, J. Comput. Phys. 183 (1) (2002) 83–116.
[67] L. Lobovskỳ, E. Botia-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.
[68] Z. Zhou, J. De Kat, B. Buchner, A nonlinear 3D approach to simulate green water dynamics on deck, in: Proceedings of the Seventh International Conference on Numerical Ship Hydrodynamics, Nantes, France, 1999, pp. 1–15.
[69] T.-h. Lee, Z. Zhou, Y. Cao, Numerical simulations of hydraulic jumps in water sloshing and water impacting, J. Fluids Eng. 124 (1) (2002) 215–226.
[70] T. Nomura, T.J. Hughes, An arbitrary lagrangian-eulerian finite element method for interaction of fluid and a rigid body, Comput. Methods Appl. Mech. Engrg. 95 (1) (1992) 115–138.
[71] D. Ishihara, S. Yoshimura, A monolithic approach for interaction of incompressible viscous fluid and an elastic body based on fluid pressure Poisson equation, Internat. J. Numer. Methods Engrg. 64 (2) (2005) 167–203.
[72] S. Wang, B.C. Khoo, G. Liu, G. Xu, L. Chen, Coupling GSM/ALE with ES-FEM-T3 for fluid–deformable structure interactions, J. Comput. Phys. 276 (2014) 315–340.
[73] S. Chen, M. Wambsganss, J. Jendrzejczyk, Added mass and damping of a vibrating rod in confined viscous fluids, J. Appl. Mech. 43 (2) (1976) 325–329.
[74] M.-J. Li, N.-M. Zhang, M.-J. Ni, Magneto-fluid-structure interaction issues for vibrating rigid bodies in conducting fluids: The numerical and the analytical approaches, Comput. Struct. 210 (2018) 41–57.
[75] 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, Trondheim, Norway, National Academy Press, Washington, DC, USA, 1996, pp. 408–423.
[76] G. Oger, M. Doring, B. Alessandrini, P. Ferrant, Two-dimensional sph simulations of wedge water entries, J. Comput. Phys. 213 (2) (2006) 803–822.
[77] Z. Wei, C. Li, R.A. Dalrymple, M. Derakhti, J. Katz, Chaos in breaking waves, Coast. Eng. 140 (2018) 272–291.