

The Immersed Interface Method for Simulating Two-Fluid Flows

Miguel Uh¹ and Sheng Xu^{1,*}

¹ *Department of Mathematics, Southern Methodist University, Dallas, TX 75275-0156, USA.*

Received 15 October 2013; Accepted 25 February 2014

Available online 11 November 2014

Abstract. We develop the immersed interface method (IIM) to simulate a two-fluid flow of two immiscible fluids with different density and viscosity. Due to the surface tension and the discontinuous fluid properties, the two-fluid flow has nonsmooth velocity and discontinuous pressure across the moving sharp interface separating the two fluids. The IIM computes the flow on a fixed Cartesian grid by incorporating into numerical schemes the necessary jump conditions induced by the interface. We present how to compute these necessary jump conditions from the analytical principal jump conditions derived in [Xu, *DCDS*, Supplement 2009, pp. 838-845]. We test our method on some canonical two-fluid flows. The results demonstrate that the method can handle large density and viscosity ratios, is second-order accurate in the infinity norm, and conserves mass inside a closed interface.

AMS subject classifications: 76M20, 65M06, 35Q35

Key words: Immersed interface method, two-fluid flows, jump conditions, augmented variable approach, singular force, Cartesian grid methods.

1. Introduction

Many natural and industrial processes involve the flow of two immiscible fluids. Examples include rise of steam in boiler tubes, bubbles in oil wells, ocean waves, geysers and sprays. Direct numerical simulations can potentially increase the understanding of such flows. There are several difficulties in the direct numerical simulation of a two-fluid flow. The interface separating the two fluids is extremely thin, leading to the discontinuities of fluid density and viscosity in the flow field. The existence of surface tension would induce a pressure jump across the interface as well. Other factors such as high density and viscosity ratios, phase transition, topological changes, and a vast

*Corresponding author. *Email addresses:* muhzapata@gmail.com (M. Uh), sxu@smu.edu (S. Xu)

range of time and length scales make the development of a robust numerical method even more challenging [8].

In recent years, different numerical methods have been proposed to solve the governing Navier-Stokes equations for two-fluid flows, and each of them has its own strengths and weaknesses. These methods can be classified into two groups: Lagrangian methods that modify the grid to match the interface location, and Eulerian methods that extract the interface location from a fixed grid.

In a Lagrangian method, the computational mesh moves and distorts with an interface. A Lagrangian method permits an interface to be specifically delineated and precisely followed, and it allows interfacial conditions to be easily applied [6]. Lagrangian methods are successful for small interface deformations [30]. However, they have difficulties when interface undergoes large deformations to require re-meshing [27]. Some examples of Lagrangian methods can be found in [7, 21, 27].

In an Eulerian method, an interface moves through a fixed grid and its position is computed at each time step. The two main approaches to follow the interface motion are interface capturing and interface tracking. With interface capturing, the interface is implicitly captured by a contour of a scalar function. Some popular examples of this kind are the volume of fluid (VOF) method [9] and the level set method [23, 28]. In the VOF method, the location of an interface is determined by the volume fraction occupied by each fluid in each computational cell. In the level set method, an interface is represented as a zero set of an auxiliary scalar function (level set function). The signed distance function is commonly used as the scalar function. An interface tracking method uses a set of Lagrangian points to mark and track an interface. The interface is treated with either finite thickness or zero thickness. Examples of interface tracking methods include the front-tracking method [33, 34] and the ghost fluid method (GFM) [5, 15]. In the front-tracking method [33, 34], a two-fluid flow is treated and solved as one system with the delta function formulation, and the interface is smoothed by the discrete approximation of the delta function. The GFM [15] eliminates the numerical smearing prevalent in the delta function formulation and treats the interface in a sharp fashion. Its basic idea is to extrapolate the solution in each fluid onto fictitious ghost nodes located in the other fluid, and then solve the governing equations in both fluids separately [40].

In an attempt to overcome some of the limitations of the above methods, there has been some hybrid methods which exploit the best features of different methods. Some examples of hybrid methods include the level-set/volume-of-fluid methods [31], the particle level set method [4], the marker/volume-of-fluid methods [1], the level-contour front tracking methods [29], and the level-set/immersed boundary method [39].

The immersed interface method (IIM) [20] was initially proposed by LeVeque and Li [17] to improve the accuracy of Peskin's immersed boundary (IB) method [24, 25]. The IIM differs from the IB method in the treatment of the singular force appearing in the delta function formulation of an interface problem. The IIM can capture the jumps of a solution and its derivative by incorporating them directly into numerical schemes.

The IIM was initially developed for solving elliptic equations. Later it was extended to Stokes flows with singular forces [16] and the Navier-Stokes equations with singular forces [18]. An implementation of the IIM for solving incompressible flows of two fluids with different viscosity but the same density was proposed in [32].

In this paper we develop the IIM to simulate a two-fluid flow of two immiscible fluids with different density and different viscosity. To achieve second-order accuracy, jump conditions of all the first- and second-order Cartesian derivatives of the velocity and the pressure are needed. We present how to numerically compute these necessary jump conditions from the analytical principal jump conditions derived in [37].

The rest of the paper is organized as follows. In Section 2, the mathematical formulation and principal jump conditions for a two-fluid flow are given. In Section 3, an overview of the IIM is presented. In Section 4, the ideas on how to apply the IIM to a two-fluid flow are detailed. In Section 5, the numerical implementation of the IIM for a two-fluid flow is described. In Section 6, numerical tests to examine the accuracy, efficiency and robustness of the IIM are provided. Finally, in Section 7, conclusions and possible improvements are outlined.

2. The two-fluid flow

We consider an incompressible two-fluid flow of immiscible fluids 1 and 2 with different density and viscosity separated by an interface Γ in a fixed two-dimensional rectangular domain $\Omega = \Omega_1 \cup \Omega_2$, see Fig. 1.

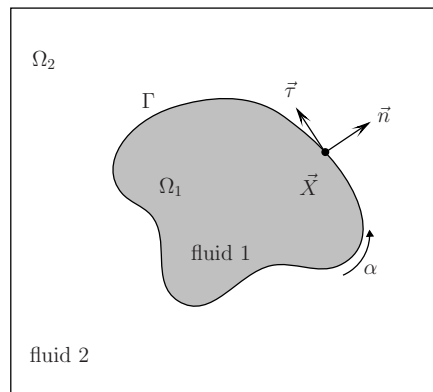


Figure 1: Schematic of a two-fluid system.

2.1. The formulation

With proper initial conditions and far-field boundary conditions, the two-fluid system is governed by a unified delta function formulation in the whole domain as [34]

$$\rho \left(\frac{\partial u_i}{\partial t} + \frac{\partial (u_i u_j)}{\partial x_j} \right) = \frac{\partial}{\partial x_j} \left(-p \delta_{ij} + \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right) + F_i + \rho G_i, \quad (2.1)$$

$$\frac{\partial u_i}{\partial x_i} = 0, \quad (2.2)$$

$$\frac{\partial X_i}{\partial t} = U_i, \quad (2.3)$$

where x_i ($i = 1, 2$ in 2D) is the Cartesian coordinates defining the domain, X_i is the Cartesian coordinates of a point on the interface, t is the time, u_i is the fluid velocity, U_i is the interface velocity, p is the fluid pressure, μ is the dynamic viscosity of the system, ρ is the density of the system, δ_{ij} is the Kronecker symbol, F_i is a singular force representing physical effects of the interface, and G_i is a finite smooth body force. For convenience, we hereafter use both a suffix notation and a vector notation to denote a vector. For example, the position vector of a point on an interface is denoted as X_i or \vec{X} .

The interface is parametrized with arc length $\alpha \in [0, l]$ as $X_i = X_i(\alpha, t)$. We assume that the interface is a smooth closed curve immersed in the fluid domain (i.e. $X_i(0, t) = X_i(l, t)$ and $X_i(\alpha, t) \in C^2_{[0, l]}$). The unit tangential vector $\vec{\tau}$ and the unit normal vector \vec{n} at the interface are

$$\vec{\tau} = (\tau_1, \tau_2) = \left(\frac{\partial X_1}{\partial \alpha}, \frac{\partial X_2}{\partial \alpha} \right), \quad \vec{n} = (n_1, n_2) = (\tau_2, -\tau_1).$$

We assume the density and viscosity are constant in each fluid, with values ρ_1 and μ_1 in the fluid 1 and ρ_2 and μ_2 in the fluid 2, respectively. The system density ρ and viscosity μ in Eq. (2.1) can be written as

$$\rho = \rho_1 H(\vec{x}, t) + \rho_2 (1 - H(\vec{x}, t)), \quad \mu = \mu_1 H(\vec{x}, t) + \mu_2 (1 - H(\vec{x}, t)),$$

where $H(\vec{x}, t)$ is defined by

$$H(\vec{x}, t) = \begin{cases} 1, & \vec{x} \in \Omega_1, \\ 0, & \vec{x} \in \Omega_2. \end{cases}$$

Note that Ω_1 and Ω_2 above are the regions occupied by fluid 1 and fluid 2 at time t , respectively, as shown in Fig. 1.

In general, the singular force F_i can be expressed as

$$F_i(\vec{x}, t) = \int_{\Gamma(\alpha, t)} f_i(\alpha, t) \delta(\vec{x} - \vec{X}(\alpha, t)) d\alpha, \quad (2.4)$$

where f_i is the force density. The force density in the two-fluid systems considered in this paper takes the following form [2]

$$f_i = \gamma \kappa n_i, \quad (2.5)$$

where γ is the surface tension, n_i denotes the normal to the interface, which points to fluid 2 as shown in Fig. 1, and κ is the local curvature of the interface, being reckoned as positive when the corresponding center of curvature lies on the side of the interface to which \vec{n} points.

2.2. Jump conditions

The IIM incorporates into numerical schemes the necessary jump conditions across the interface induced by the singular force and the discontinuous fluid density and viscosity. Define $f_n = \vec{f} \cdot \vec{n}$ and $f_\tau = \vec{f} \cdot \vec{\tau}$, which are the normal and tangential force density, respectively. Denote a jump condition across the interface as

$$[\cdot] := (\cdot)_{\text{fluid 2}} - (\cdot)_{\text{fluid 1}}.$$

The following principal jump conditions for a two-fluid system have been derived by Xu [37]

$$[u_i] = 0, \quad (2.6)$$

$$\left[\mu \frac{\partial u_i}{\partial \mathbf{n}} \right] = -f_\tau \tau_i - [\mu] \left(\left(\frac{\partial \vec{U}}{\partial \boldsymbol{\tau}} \cdot \vec{n} \right) \tau_i + \left(\frac{\partial \vec{U}}{\partial \boldsymbol{\tau}} \cdot \vec{\tau} \right) n_i \right), \quad (2.7)$$

$$[\Delta u_i] = \left[\frac{\partial^2 u_i}{\partial \mathbf{n}^2} \right] + \kappa \left[\frac{\partial u_i}{\partial \mathbf{n}} \right], \quad (2.8)$$

$$[p] = f_n - 2[\mu] \left(\frac{\partial \vec{U}}{\partial \boldsymbol{\tau}} \cdot \vec{\tau} \right), \quad (2.9)$$

$$\left[\frac{1}{\rho} \frac{\partial p}{\partial \mathbf{n}} \right] = \frac{\partial}{\partial \boldsymbol{\tau}} \left(-\vec{\tau} \cdot \left[\frac{\mu}{\rho} \frac{\partial \vec{u}}{\partial \mathbf{n}} \right] + \left[\frac{\mu}{\rho} \right] \left(\frac{\partial \vec{U}}{\partial \boldsymbol{\tau}} \cdot \mathbf{n} \right) \right), \quad (2.10)$$

$$[\Delta p] = \left[\rho \left(\frac{\partial G_i}{\partial x_i} - \frac{\partial u_j}{\partial x_i} \frac{\partial u_i}{\partial x_j} \right) \right], \quad (2.11)$$

where κ is the local curvature. These jump conditions are called *the principal jump conditions* since jump conditions of Cartesian derivatives can be derived from them.

The unit step function $H(\vec{x}, t)$ can be solved from the Laplace equation $\Delta H = 0$ with the following principal jump conditions across the interface

$$[H] = -1, \quad \left[\frac{\partial H}{\partial \mathbf{n}} \right] = 0, \quad [\Delta H] = 0. \quad (2.12)$$

3. An overview of the IIM

The immersed interface method (IIM) preserves the nonsmoothness or discontinuities in a numerical solution (the velocity, the pressure or the unit step function) and achieves second-order or higher accuracy by incorporating necessary jump conditions into a standard finite difference scheme on a fixed Cartesian grid. A standard finite difference scheme has its usual form if its stencil does not cross the interface. Otherwise, it contains a jump contribution which is composed of necessary jump conditions of the solution. To determine the form of the jump contribution, we use the generalized Taylor expansion for a piecewise smooth function [35]. The following examples show

the central finite difference schemes for first- and second-order derivatives when the solution has discontinuities on the stencil.

Let $x_{i+1} - x_i = x_i - x_{i-1} = h > 0$ and $x_{i-1} < \xi < x_i \leq \eta < x_{i+1}$. Suppose that $\phi(x)$ is smooth except at discontinuous points of the first kind, ξ and η . Then

$$\begin{aligned} \frac{d\phi(x_i^-)}{dx} &= \frac{\phi(x_{i+1}^-) - \phi(x_{i-1}^+)}{2h} + \mathcal{O}(h^2) \\ &+ \frac{1}{2h} \left(\sum_{n=0}^2 \frac{-[\phi^{(n)}(\xi)]}{n!} (x_{i-1} - \xi)^n - \sum_{n=0}^2 \frac{[\phi^{(n)}(\eta)]}{n!} (x_{i+1} - \eta)^n \right), \end{aligned} \quad (3.1)$$

$$\begin{aligned} \frac{d^2\phi(x_i^-)}{dx^2} &= \frac{\phi(x_{i+1}^-) - 2\phi(x_i) + \phi(x_{i-1}^+)}{h^2} + \mathcal{O}(h^2) \\ &- \frac{1}{h^2} \left(\sum_{n=0}^3 \frac{-[\phi^{(n)}(\xi)]}{n!} (x_{i-1} - \xi)^n + \sum_{n=0}^3 \frac{[\phi^{(n)}(\eta)]}{n!} (x_{i+1} - \eta)^n \right). \end{aligned} \quad (3.2)$$

Notice that to obtain second-order accuracy in approximating the second-order derivative above, the jump condition of the third-order derivative $[\frac{\partial^3\phi}{\partial x^3}]$ is needed. However, this jump condition is difficult or costly to obtain. It was proved by Huang and Li [11] that even though the local truncation error is reduced from $\mathcal{O}(h^2)$ to $\mathcal{O}(h)$ near an interface, the IIM can still achieve second-order accuracy in the infinity norm in solving a Poisson equation. A second-order accuracy has also been numerically confirmed for the Navier-Stokes equations when only the jump conditions of the first- and second-order Cartesian derivatives are used.

Using the systematic approach in [35], we can derive the jump conditions for all first- and second-order Cartesian derivatives of ϕ if we know the following principal jump conditions

$$[\phi], \quad \left[\frac{\partial\phi}{\partial \mathbf{n}} \right], \quad [\Delta\phi].$$

In particular, the follows are spatial jump conditions in 2D obtained from the 3D results in [35]

$$\left[\frac{\partial\phi}{\partial x_1} \right] = \tau_1 \frac{\partial[\phi]}{\partial \alpha} + n_1 \left[\frac{\partial\phi}{\partial \mathbf{n}} \right], \quad (3.3)$$

$$\left[\frac{\partial\phi}{\partial x_2} \right] = \tau_2 \frac{\partial[\phi]}{\partial \alpha} + n_2 \left[\frac{\partial\phi}{\partial \mathbf{n}} \right], \quad (3.4)$$

$$\left[\frac{\partial^2\phi}{\partial x_1^2} \right] = r_1(\tau_1^2 - \tau_2^2) + r_2(2\tau_1\tau_2) + r_3(\tau_2^2), \quad (3.5)$$

$$\left[\frac{\partial^2\phi}{\partial x_1\partial x_2} \right] = r_1(2\tau_1\tau_2) + r_2(\tau_2^2 - \tau_1^2) + r_3(-\tau_1\tau_2), \quad (3.6)$$

$$\left[\frac{\partial^2\phi}{\partial x_2^2} \right] = r_1(\tau_2^2 - \tau_1^2) + r_2(-2\tau_1\tau_2) + r_3(\tau_1^2), \quad (3.7)$$

where r_1 , r_2 and r_3 are given by

$$\begin{aligned} r_1 &= \frac{\partial^2[\phi]}{\partial\alpha^2} - \left(\frac{\partial\tau_1}{\partial\alpha} \left[\frac{\partial\phi}{\partial x_1} \right] + \frac{\partial\tau_2}{\partial\alpha} \left[\frac{\partial\phi}{\partial x_2} \right] \right), \\ r_2 &= \frac{\partial}{\partial\alpha} \left[\frac{\partial\phi}{\partial\mathbf{n}} \right] - \left(\frac{\partial n_1}{\partial\alpha} \left[\frac{\partial\phi}{\partial x_1} \right] + \frac{\partial n_2}{\partial\alpha} \left[\frac{\partial\phi}{\partial x_2} \right] \right), \\ r_3 &= [\Delta\phi]. \end{aligned}$$

4. The IIM for the two-fluid flow

As shown in Section 3, computing the jump conditions of all the first- and second-order Cartesian derivatives of the velocity u_i and the pressure p requires the following principal jump conditions

$$\left[\frac{\partial u_i}{\partial\mathbf{n}} \right], \quad \left[\frac{\partial p}{\partial\mathbf{n}} \right].$$

For the two-fluid flow, the available principal jump conditions are

$$\left[\mu \frac{\partial u_i}{\partial\mathbf{n}} \right], \quad \left[\frac{1}{\rho} \frac{\partial p}{\partial\mathbf{n}} \right],$$

which are given in Eqs. (2.7) and (2.10) in Section 2.2. The available principal jump conditions are not in the desired form due to the discontinuous fluid properties. Furthermore, they involve the interface velocity \vec{U} , as indicated in Eqs. (2.7) and (2.10).

4.1. Interpolation of the interface velocity

The interface velocity is present in some principal jump conditions given by Eqs. (2.6)-(2.11), so it is necessary to interpolate it accurately. To obtain second-order accuracy, we incorporate the known jump condition $[\mu \frac{\partial u_i}{\partial\mathbf{n}}]$ in the interpolation.

To derive the interpolation formula, we use the following Taylor expansions in the normal direction of the interface

$$\begin{aligned} u_i^{+\delta n} &= U_i + \delta n \left. \frac{\partial u_i}{\partial\mathbf{n}} \right|_{\text{fluid2}} + \mathcal{O}(\delta n^2), \\ u_i^{-\delta n} &= U_i - \delta n \left. \frac{\partial u_i}{\partial\mathbf{n}} \right|_{\text{fluid1}} + \mathcal{O}(\delta n^2), \end{aligned}$$

where $u_i^{\pm\delta n}$ ($i = 1, 2$) is the velocity a distance of δn away from the interface along the normal direction on both sides of the interface, as illustrated in Fig. 2(a), and is interpolated from surrounding Cartesian grid points using standard second-order bilinear interpolation. Multiplying the first Taylor expansion by the viscosity μ_2 and the second by μ_1 and then adding the results, we obtain the following interpolation formula for the interface velocity

$$U_i = \frac{\mu_1 u_i^{-\delta n} + \mu_2 u_i^{+\delta n}}{\mu_1 + \mu_2} - \frac{1}{\mu_1 + \mu_2} \left[\mu \frac{\partial u_i}{\partial\mathbf{n}} \right] \delta n + \mathcal{O}(\delta n^2). \tag{4.1}$$

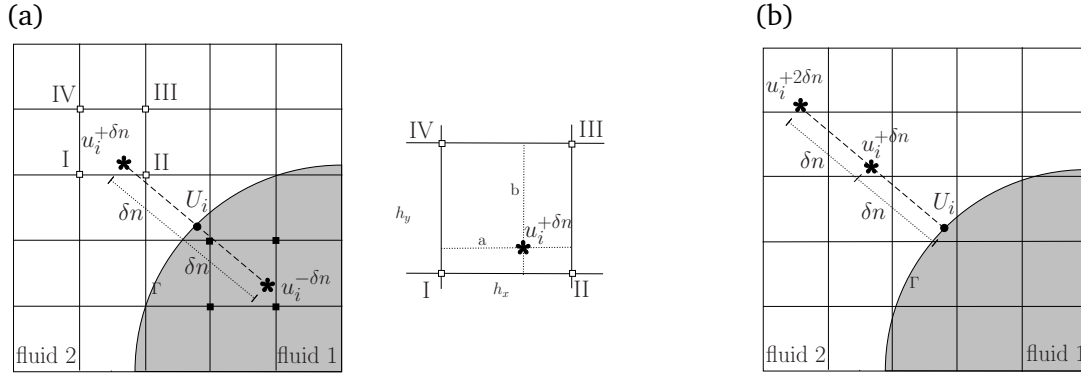


Figure 2: (a) Interpolation stencil; (b) one-sided finite difference stencil.

We use the fluid velocity at the current time step to find the current interface velocity. Since the jump condition $[\mu \frac{\partial u_i}{\partial \mathbf{n}}]$, given by Eq. (2.7), involves the interface velocity, the interpolation formula gives an implicit equation for U_i as

$$U_i = \frac{\mu_1 u_i^{-\delta n} + \mu_2 u_i^{+\delta n}}{\mu_1 + \mu_2} + \mathcal{O}(\delta n^2) - \frac{1}{\mu_1 + \mu_2} \left(-f_\tau \tau_i - [\mu] \left(\left(\frac{\partial \vec{U}}{\partial \boldsymbol{\tau}} \cdot \mathbf{n} \right) \tau_i + \left(\frac{\partial \vec{U}}{\partial \boldsymbol{\tau}} \cdot \boldsymbol{\tau} \right) n_i \right) \right) \delta n. \quad (4.2)$$

Please see the Appendix on how to treat this implicit equation numerically.

4.2. Computing $\left[\frac{\partial u_i}{\partial \mathbf{n}} \right]$

To derive the jump conditions of first- and second- Cartesian derivatives of the velocity, the desired jump condition is $\left[\frac{\partial u_i}{\partial \mathbf{n}} \right]$, but the available jump condition is $[\mu \frac{\partial u_i}{\partial \mathbf{n}}]$. We can compute the desired jump condition using one-sided finite difference approximations. By the definition of a jump condition, it can be shown that

$$\left[\frac{\partial u_i}{\partial \mathbf{n}} \right] = \frac{1}{\mu_1} \left[\mu \frac{\partial u_i}{\partial \mathbf{n}} \right] - \frac{[\mu]}{\mu_1} \frac{\partial u_i}{\partial \mathbf{n}} \Big|_2, \quad (4.3)$$

$$\left[\frac{\partial u_i}{\partial \mathbf{n}} \right] = \frac{1}{\mu_2} \left[\mu \frac{\partial u_i}{\partial \mathbf{n}} \right] - \frac{[\mu]}{\mu_2} \frac{\partial u_i}{\partial \mathbf{n}} \Big|_1. \quad (4.4)$$

We can approximate the one-sided derivative $\frac{\partial u_i}{\partial \mathbf{n}} \Big|_{\text{fluid1}}$ or $\frac{\partial u_i}{\partial \mathbf{n}} \Big|_{\text{fluid2}}$ using a one-sided finite difference approximation with a stencil shown Fig. 2(b). Eqs. (4.3) and (4.4) imply that the error in the one-sided finite difference approximations is multiplied by a factor $\frac{[\mu]}{\mu_1}$ or $\frac{[\mu]}{\mu_2}$. We therefore choose the formula that has the smaller factor.

4.3. Computing $\left[\frac{\partial p}{\partial \mathbf{n}} \right]$

Taking the divergence of the momentum equation, Eq. (2.1), we have the Poisson equation for $p^* = p/\rho$ subject to the known principal jump conditions:

$$\Delta p^* = s_p, \tag{4.5}$$

$$[\rho p^*] = [p], \quad \left[\frac{\partial p^*}{\partial \mathbf{n}} \right] = \left[\frac{1}{\rho} \frac{\partial p}{\partial \mathbf{n}} \right], \quad [\Delta p^*] = [s_p], \tag{4.6}$$

The right hand side s_p is

$$s_p = -\frac{\partial D}{\partial t} - 2\nabla \cdot (\vec{u}D) + \frac{\mu}{\rho} \Delta D - 2 \left(\frac{\partial u_1}{\partial x_2} \frac{\partial u_2}{\partial x_1} - \frac{\partial u_1}{\partial x_1} \frac{\partial u_2}{\partial x_2} \right) + \nabla \cdot \vec{G}, \tag{4.7}$$

where $D = \nabla \cdot \vec{u}$ ($D = 0$ theoretically) is kept in s_p to better enforce the divergence-free condition numerically. We apply the augmented variable approach [19] to compute $p^* = p/\rho$ using the known principal jump conditions $[\rho p^*]$, $\left[\frac{\partial p^*}{\partial \mathbf{n}} \right]$, and $[\Delta p^*]$. We compute p^* instead of p , which turns out to make our algorithm more stable.

If we know $[p^*]$ in addition to $\left[\frac{\partial p^*}{\partial \mathbf{n}} \right]$ and $[\Delta p^*]$, we can derive all the jump conditions of the first- and second-order Cartesian derivatives of p^* (see Section 3) [35] and apply the IIM to compute p^* . In the augmented variable approach, the desired jump condition $[p^*]$ is set as an augmented variable and computed together with p^* in an iterative manner until the available jump condition $[\rho p^*] = [p]$ listed in Eq. (4.6) is satisfied. Below are the details.

Let η_e be the vector of the exact values of $[p^*]$ at the discrete points representing the interface. Let ψ_e be the vector of the available principal jump condition $[p] = [\rho p^*]$. Let η be an arbitrary guess of the exact jump condition η_e . Using the guess η , the discretized Poisson equation, Eq. (4.5), leads to the linear system

$$L\mathbf{p}^* + C\eta = \mathbf{b}, \tag{4.8}$$

where the matrix L corresponds to the discrete Laplacian, \mathbf{p}^* denotes the vector of the values of p^* on the Cartesian grid, the matrix C accounts for the jump contribution due to η , and the vector \mathbf{b} involves the discretized right-hand side s_p and the known jump conditions $\left[\frac{\partial p^*}{\partial \mathbf{n}} \right]$ and $[\Delta p^*]$. For the guess η , we can therefore solve Eq. (4.8) for p^* as

$$\mathbf{p}^* = L^{-1}(\mathbf{b} - C\eta). \tag{4.9}$$

Using the guess η and the computed \mathbf{p}^* , we can obtain the values of $[\rho p^*]$ at the discrete points on the interface, denoted as ψ , from one of the following formulas:

$$[\rho p^*] = \rho_2 [p^*] - [\rho] p^*|_1, \quad \text{if } \rho_2 > \rho_1, \tag{4.10}$$

$$[\rho p^*] = \rho_1 [p^*] - [\rho] p^*|_2, \quad \text{if } \rho_2 < \rho_1, \tag{4.11}$$

where $p^*|_1$ or $p^*|_2$ is extrapolated from the computed p^* . These formulas imply that ψ can be written as

$$\psi = B\boldsymbol{\eta} + E\mathbf{p}^*, \quad (4.12)$$

where $B = \rho_1 I$ or $B = \rho_2 I$ (I is the identity matrix), and E is the extrapolation matrix.

Combining Eqs. (4.9) and (4.12), we have

$$M\boldsymbol{\eta} = \psi - EL^{-1}\mathbf{b}, \quad (4.13)$$

where $M = B - EL^{-1}C$ is the Schur complement. Let ψ_0 correspond to the zero guess $\boldsymbol{\eta} = \mathbf{0}$. From Eq. (4.13), we then have

$$\psi_0 = BL^{-1}\mathbf{b},$$

and

$$M\boldsymbol{\eta} = \psi - \psi_0. \quad (4.14)$$

In particular the exact jump condition $\boldsymbol{\eta}_e$ can be found by solving the Schur complement linear system

$$M\boldsymbol{\eta}_e = \psi_e - \psi_0.$$

We use GMRES to solve the system because it does not need to form the matrix M explicitly. It only needs to compute the matrix vector product $M\boldsymbol{\eta}$ for a given guess $\boldsymbol{\eta}$. The result of the matrix product is just the right hand side of Eq. (4.14). In each GMRES iteration we need to solve the discrete Poisson equation. We use FFT to do so, which results in $\mathcal{O}(N \ln N)$ flops in each iteration, where N is the number of Cartesian grid points.

Note that we can also use the iterative two-fluid pressure solver proposed by Xu [38] to compute the pressure p^* using the available jump conditions.

5. Implementation

In this section, we describe the implementation of the IIM for solving the two-fluid flow. It is very similar to the implementation of the IIM for solving fluid-solid interaction in [36] (in fact, we modified our fluid-solid interaction code to produce our two-fluid flow code). Hereafter, we use the notations $\vec{u} = (u, v)$ and $\vec{x} = (x, y)$ instead of (u_1, u_2) and (x_1, x_2) to be consistent with [36].

5.1. Spatial discretization

We define the discrete flow variables on a staggered Marker-and-Cell (MAC) grid, as shown in Fig. 3(a). The pressure p is defined at the center of the cell (i, j) ($i \in \{1, 2, \dots, N_x\}$ and $j \in \{1, 2, \dots, N_y\}$). The velocity components u and v are defined at the edge centers (I, j) and (i, J) , respectively, where (I, J) corresponds to $(i + \frac{1}{2}, j + \frac{1}{2})$. The dimensions of the cell are Δx and Δy .

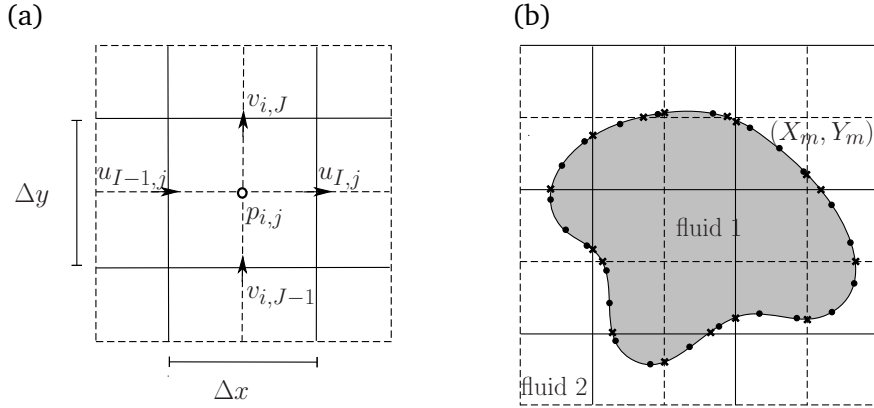


Figure 3: (a) MAC grid with staggered arrangement of flow variables; (b) Lagrangian markers (dots) and irregular points (x -symbols) on the interface.

We track the interface using N_s discrete Lagrangian markers (X_m, Y_m) which correspond to $m\Delta\alpha$ ($m \in \{0, 1, 2, \dots, N_s\}$), as shown in Fig. 3(b). We use periodic cubic splines to represent the closed interface. We calculate geometric quantities and jump conditions at the Lagrangian markers and interpolate the jump conditions at the intersections between grid lines and the interface (the x -symbols in Fig. 3(b)). We re-parametrize the interface every time step such that the Lagrangian markers are uniformly distributed along the interface.

Based on Eqs. (3.1) and (3.2), we can discretize Eq. (4.5) as the following

$$\begin{aligned} & \frac{p_{i+1,j}^* - 2p_{i,j}^* + p_{i-1,j}^*}{\Delta x^2} + \frac{p_{i,j+1}^* - 2p_{i,j}^* + p_{i,j-1}^*}{\Delta y^2} \\ &= Z_{i,j} - 2 \left(\frac{u_{i,J} - u_{i,J-1}}{\Delta y} \frac{v_{I,j} - v_{I-1,j}}{\Delta x} - \frac{u_{I,j} - u_{I-1,j}}{\Delta x} \frac{v_{i,J} - v_{i,J-1}}{\Delta y} \right) \\ & \quad + \frac{G_{I,j}^x - G_{I-1,j}^x}{\Delta x} + \frac{G_{i,J}^y - G_{i,J-1}^y}{\Delta y} + C_{i,j}^p, \end{aligned} \quad (5.1)$$

where

$$\begin{aligned} Z_{i,j} &= \left(\frac{\partial D}{\partial t} \right)_{i,j} - 2 \left(\frac{u_{I,j} D_{I,j} - u_{I-1,j} D_{I-1,j}}{\Delta x} + \frac{v_{i,J} D_{i,J} - v_{i,J-1} D_{i,J-1}}{\Delta y} \right) \\ & \quad + \frac{\mu_{ij}}{\rho_{ij}} \left(\frac{D_{i+1,j} - 2D_{i,j} + D_{i-1,j}}{\Delta x^2} + \frac{D_{i,j+1} - 2D_{i,j} + D_{i,j-1}}{\Delta y^2} \right) \end{aligned}$$

and $C_{i,j}^p$ corresponds to the term due to the jump contribution from each finite difference approximation of the derivatives in Eq. (5.1). The time derivative $\frac{\partial D}{\partial t}$ is approximated by assuming divergence free condition at the next time level. The divergence $D_{i,j}$ is computed as

$$D_{i,j} = \frac{u_{I,j} - u_{I-1,j}}{\Delta x} + \frac{v_{i,J} - v_{i,J-1}}{\Delta y} + C_{i,j}^D,$$

where the jump contribution is denoted as $C_{i,j}^D$. The jump contributions $C_{i,i}^p$ and $C_{i,j}^D$ are nonzero only if the stencil of a finite difference approximation has irregular points.

Similarly, we can discretize Eq. (2.1) as

$$\begin{aligned} \left(\frac{\partial u}{\partial t}\right)_{I,j} = & C_{I,j}^u - \frac{p_{i+1,j}^* - p_{i,j}^*}{\Delta x} - \left(\frac{u_{i+1,j}^2 - u_{i,j}^2}{\Delta x} + \frac{(uv)_{I,J} - (uv)_{I,J-1}}{\Delta y} \right) \\ & + \frac{\mu_{I,j}}{\rho_{I,j}} \left(\frac{u_{I+1,j} - 2u_{I,j} + u_{I-1,j}}{\Delta x^2} + \frac{u_{I,j+1} - 2u_{I,j} + u_{I,j-1}}{\Delta y^2} \right), \end{aligned} \quad (5.2)$$

$$\begin{aligned} \left(\frac{\partial v}{\partial t}\right)_{i,J} = & C_{i,J}^v - \frac{p_{i,j+1}^* - p_{i,j}^*}{\Delta y} - \left(\frac{(uv)_{I,J} - (uv)_{I-1,J}}{\Delta x} + \frac{v_{i+1,j}^2 - v_{i,j}^2}{\Delta y} \right) \\ & + \frac{\mu_{i,J}}{\rho_{i,J}} \left(\frac{v_{i+1,J} - 2v_{i,J} + u_{i-1,J}}{\Delta x^2} + \frac{v_{i,J+1} - 2v_{i,J} + v_{i,J-1}}{\Delta y^2} \right), \end{aligned} \quad (5.3)$$

where $C_{I,j}^u$, $C_{i,J}^v$ are jump contributions.

5.2. Time integration

The discretized momentum equation and the kinematic equation for the interface can be written as

$$\frac{d\mathbf{u}}{dt} = \mathbf{R}(\mathbf{u}, \mathbf{X}, p), \quad (5.4)$$

$$\frac{d\mathbf{X}}{dt} = \mathbf{U}(\mathbf{u}, \mathbf{X}), \quad (5.5)$$

where \mathbf{u} is the velocity at all the grid points and \mathbf{X} is the coordinates of all the Lagrangian markers. We currently employ an explicit fourth-order Runge-Kutta scheme for the time integration of the above two equations. Explicit high-order time integration schemes are appropriate at moderate Reynolds numbers, as shown by Johnston and Liu [13, 14] and Liu [3]. We thus have

$$\begin{aligned} \mathbf{u}_1 &= \mathbf{u}^n, & \mathbf{X}_1 &= \mathbf{X}^n, \\ \mathbf{u}_2 &= \mathbf{u}^n + \frac{\Delta t}{2} \mathbf{R}(\mathbf{u}_1, \mathbf{X}_1, p_1), & \mathbf{X}_2 &= \mathbf{X}^n + \frac{\Delta t}{2} \mathbf{U}(\mathbf{u}_1, \mathbf{X}_1), \\ \mathbf{u}_3 &= \mathbf{u}^n + \frac{\Delta t}{2} \mathbf{R}(\mathbf{u}_2, \mathbf{X}_2, p_2), & \mathbf{X}_3 &= \mathbf{X}^n + \frac{\Delta t}{2} \mathbf{U}(\mathbf{u}_2, \mathbf{X}_2), \\ \mathbf{u}_4 &= \mathbf{u}^n + \Delta t \mathbf{R}(\mathbf{u}_3, \mathbf{X}_3, p_3), & \mathbf{X}_4 &= \mathbf{X}^n + \Delta t \mathbf{U}(\mathbf{u}_3, \mathbf{X}_3), \end{aligned}$$

and

$$\begin{aligned} \mathbf{u}^{n+1} &= \mathbf{u}^n + \frac{\Delta t}{6} (\mathbf{R}(\mathbf{u}_1, \mathbf{X}_1, p_1) + 2\mathbf{R}(\mathbf{u}_2, \mathbf{X}_2, p_2) + 2\mathbf{R}(\mathbf{u}_3, \mathbf{X}_3, p_3) + \mathbf{R}(\mathbf{u}_4, \mathbf{X}_4, p_4)), \\ \mathbf{X}^{n+1} &= \mathbf{X}^n + \frac{\Delta t}{6} (\mathbf{U}(\mathbf{u}_1, \mathbf{X}_1) + 2\mathbf{U}(\mathbf{u}_2, \mathbf{X}_2) + 2\mathbf{U}(\mathbf{u}_3, \mathbf{X}_3) + \mathbf{U}(\mathbf{u}_4, \mathbf{X}_4)). \end{aligned}$$

The pressure p_k ($k = 1, 2, 3, 4$) above is computed by the augmented variable approach described in Section 4.3 using the approximations \mathbf{u}_k and \mathbf{X}_k at the current time sub-step. Since the velocity update scheme is explicit, we enforce the following Courant-Friedrichs-Lewy (CFL) condition and the viscous time step restriction [15, 26]

$$\Delta t < \frac{1}{\frac{|u|_{\max}}{\Delta x} + \frac{|v|_{\max}}{\Delta y}} \quad \text{and} \quad \Delta t < \frac{1}{2 \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right)} \min \left\{ \frac{\rho_1}{\mu_1}, \frac{\rho_2}{\mu_2} \right\}. \quad (5.6)$$

5.3. Algorithm

We describe the algorithm in our method using the flow chart in Fig. 4.

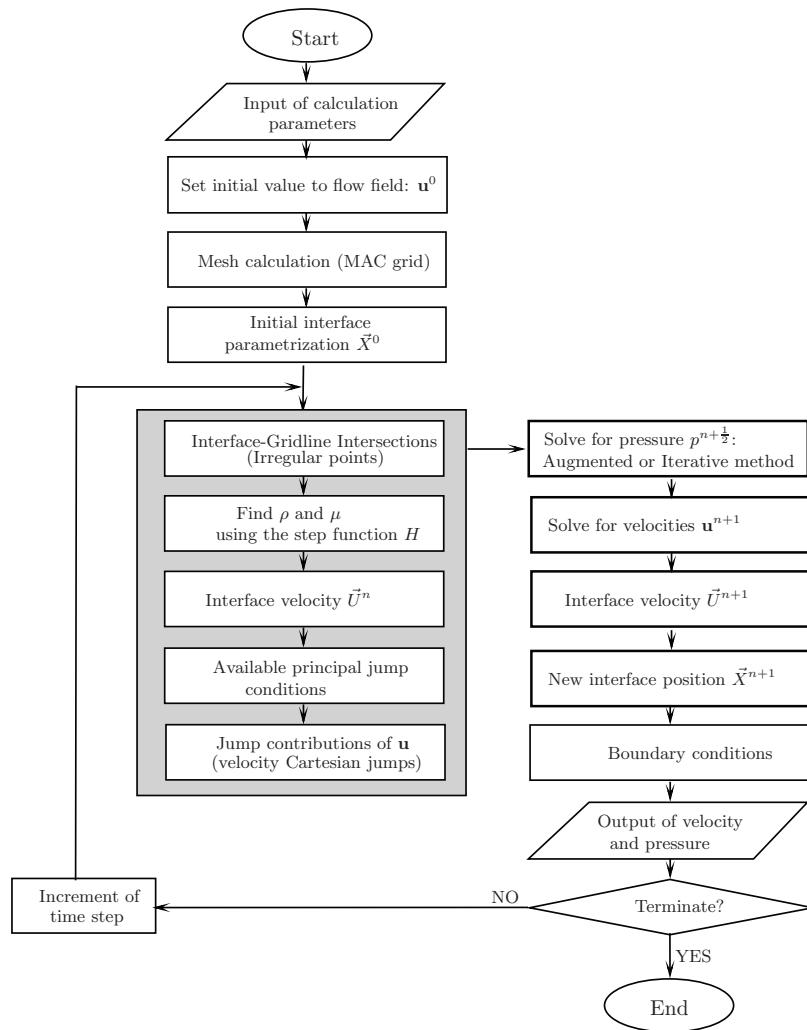


Figure 4: The flow chart of the algorithm.

6. Numerical results

In this section, we numerically test the accuracy and robustness of our method by simulating some canonical two-fluid flows. The non-dimensional parameters and results in each simulation are based on some reference density scale ρ_0 , viscosity scale μ_0 , length scale L , velocity scale U , and time scale L/U .

6.1. Example 1: Circular flow

In this example, we consider two fluids with different viscosity and density across a circular interface located at the radius $r = 0.5$ in the domain $\Omega = [-1, 1] \times [-1, 1]$, as shown in Fig. 5.

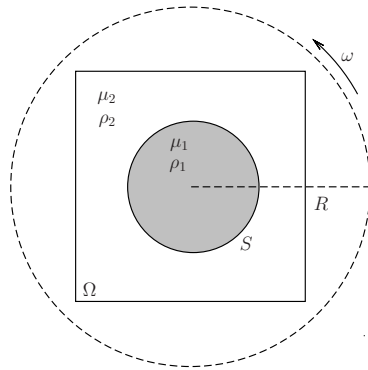


Figure 5: Geometry of a circular two-fluid flow.

We prescribe the angular velocity $\omega = 1$ at a circumference located at the radius $R = 3$ (outside the computational domain). The steady state of the flow is in rigid rotation:

$$\begin{aligned} u(x, y) &= -\omega y, \\ v(x, y) &= \omega x, \\ p(x, y) &= \frac{1}{2}\rho\omega^2(x^2 + y^2) + c, \end{aligned} \tag{6.1}$$

where c is an arbitrary constant. We set the Dirichlet boundary conditions for the velocity and the Neumann boundary conditions for the pressure at the domain boundaries according to the analytical steady state solution. The steady state solution is evolved out from the initial zero velocity and pressure fields. The jump conditions are calculated using the formulas given in Eqs. (2.6)-(2.11). We compare our numerical steady state with the analytical one.

We first take a look at solutions obtained with $\rho_1 = 1$, $\rho_2 = 2$, $\mu_1 = 1$, $\mu_2 = 2$ at time $t = 2$ using $N_x = N_y = N$, $N_s = 2N$, where N can take different values for different spatial resolutions. The convergence test using $N = 32, 64, 128, 256$ is given in Table 1, indicating about second order accuracy in both the velocity and the pressure.

Table 1: Convergence test with the augmented variable approach for p^* : $\rho_1 = 1$, $\rho_2 = 2$, $\mu_1 = 1$, $\mu_2 = 2$.

N	$\ u - u_e\ _\infty$	Order	$\ v - v_e\ _\infty$	Order	$\ p^* - p_e^*\ _\infty$	Order
32	3.46×10^{-4}	-	3.46×10^{-4}	-	3.96×10^{-3}	-
64	8.82×10^{-5}	1.97	8.82×10^{-5}	1.97	1.10×10^{-3}	1.85
128	2.20×10^{-5}	1.99	2.20×10^{-5}	1.99	4.11×10^{-4}	1.42
256	5.30×10^{-6}	2.01	5.30×10^{-6}	2.01	8.63×10^{-5}	2.25

Table 2: Convergence test for the circular flow using different densities ratios ρ_2/ρ_1 and same Reynolds number $Re_1 = 1$, $Re_2 = 2$.

N	$\rho_2/\rho_1 = 10$			$\rho_2/\rho_1 = 100$			$\rho_2/\rho_1 = 1000$		
	$\ u - u_e\ _\infty$	n_i	Order	$\ u - u_e\ _\infty$	n_i	Order	$\ u - u_e\ _\infty$	n_i	Order
32	3.84×10^{-3}	2	-	6.24×10^{-3}	4	-	6.54×10^{-3}	5	-
64	9.61×10^{-3}	2	2.00	1.54×10^{-3}	4	2.02	1.61×10^{-3}	5	2.02
128	2.39×10^{-5}	2	2.01	3.81×10^{-4}	3	2.02	3.99×10^{-4}	5	2.02

For small density ratios ($1 \leq \rho_2/\rho_1 < 10$), computing either p or p^* by the augmented variable approach does not affect the stability of the method, but this is not the case if density ratios are large. The method using p becomes unstable, while the method using p^* can still maintain stability. The improvement in the stability may be attributed to the more accurate calculation of the pressure gradient $\frac{1}{\rho} \frac{\partial p}{\partial x_i} = \frac{\partial p^*}{\partial x_i}$ in the Navier-Stokes equations with the method using p^* , even though the error in the pressure $p = \rho p^*$ can be larger than the method using p . Table 2 compares the errors in u and the number n_i of GMRES iterations at different densities ratios for the method using p^* . In general, the Poisson solver requires more GMRES iterations for large density ratios, and the method is still second order.

6.2. Example 2: Static bubble

In the second test, we consider a circular bubble in static equilibrium without the gravity. This example is used to verify the calculation of surface tension and the possible presence of parasitic currents. The surface tension is introduced in the simulations by the singular force $f_i = \gamma \kappa n_i$, where γ is the surface tension coefficient. Theoretically, in the absence of external forces and initial velocity, the velocity field should remain zero throughout the domain and the pressure rises from a constant value outside the bubble to a different constant value inside the bubble with the jump $[p] = \gamma \kappa$. In the simulation, the computational domain is the square $[-1, 1] \times [-1, 1]$; and the bubble has the center $(0, 0)$ and the radius $r = 0.5$. The no-slip condition is imposed on all the domain boundaries.

Fig. 6 shows the velocity component u and the pressure p on a 64×64 grid at time $t = 1$ with $\rho_2/\rho_1 = 1000/1$, $\mu_2/\mu_1 = 100/1$ and $\gamma = 0.05$, indicating negligible parasitic currents. We also compare the computed pressure jumps $[p]$ with the analytic value $[p]_e = 2\gamma$ at $t = 1$ for various parameter combinations. The relative errors in the pressure jumps in all cases are less than 0.005%.

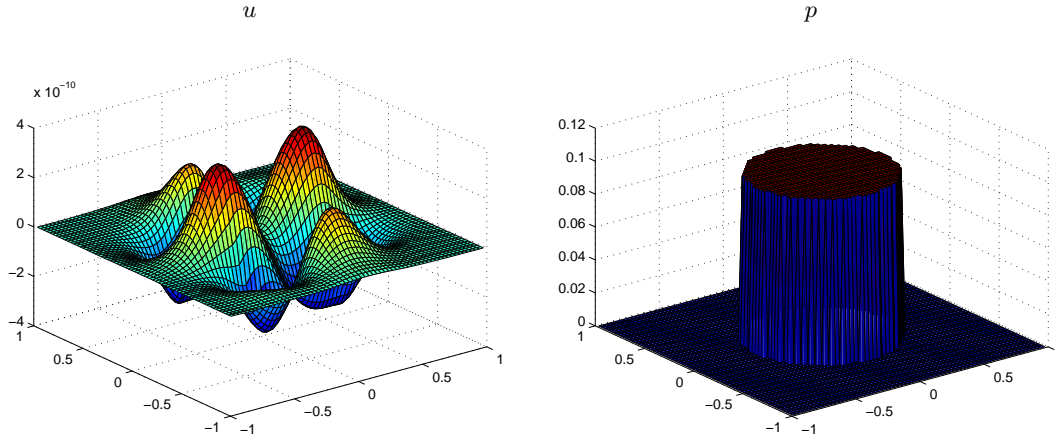


Figure 6: Velocity component u and pressure p for a static bubble in a fluid with $\rho_2/\rho_1 = 1000$, $\mu_2/\mu_1 = 100$, $\gamma = 0.05$.

6.3. Example 3: Oscillating bubble

In the third example, we consider the relaxation of a non-spherical bubble toward its static equilibrium shape to test the conservation of the bubble area. The relaxation is driven by the surface tension. The initial bubble shape is a rounded plate as shown in Fig. 7, and the fluid is initially at rest. The computational domain is $\Omega = [-1, 1] \times [-1, 1]$. No-slip conditions are applied at all the domain boundaries. The gravitational and other external forces are absent. The dimensionless parameters used in the simulation were set to $Re = 50$, $Bo = 10$, $\rho_1 = 10$, $\rho_2 = 1$ and $\mu_1 = 10$, $\mu_2 = 1$, where $Re = \frac{\rho_0 L U}{\mu_0}$, and $Bo = \frac{\rho_0 U^2 L}{\gamma}$ (the reference velocity U is determined based on the Bond number Bo and the surface tension coefficient γ).

Fig. 7 shows the bubble shape and the velocity field at different time. At time $t = 3$, the bubble is almost circular. The decay of the bubble oscillation is apparent in the variation of the total fluid kinetic energy $K = \frac{1}{2} \int_{\Omega} \rho \vec{u} \cdot \vec{u} dA$ shown in Fig. 8(a). The relative change of bubble area versus time is plot in Fig. 8(b). The maximum relative error in the bubble area on the considered non-dimensional time interval $[0, 4]$ is about 1%.

We estimate the convergence rate by comparing the maximum difference between numerical solutions at successive grid sizes. We compute the solutions at four different resolutions with $N_i \times N_i$ ($i = 1, 2, 3, 4$) Cartesian grid points and $N_{si} = 2N_i$ Lagrangian markers, and let $N_1 = 32$, $N_2 = 64$, $N_3 = 128$, $N_4 = 256$. The order of accuracy can be estimated from

$$\text{order} = -\frac{\log(E_{i+1}(q)/E_i(q))}{\log(N_{i+1}/N_i)}, \quad (6.2)$$

where $E_i(q) = \|q_{N_i} - q_{N_{i+1}}\|_{\infty}$ denotes the max norm of the error in the variable q at the grid level N_i . We set $Re = 50$, $Bo = 1$, $\rho_2/\rho_1 = 10$, $\mu_2/\mu_1 = 10$ and calculate the

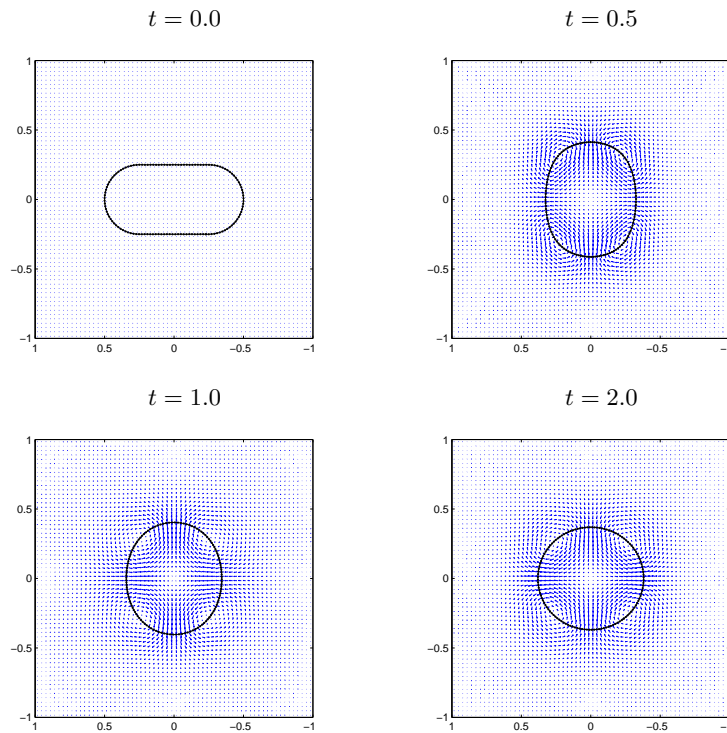


Figure 7: Bubble shape and velocity field at different time as a bubble is relaxing to its equilibrium with $Re = 50$, $Bo = 10$, $\rho_2/\rho_1 = 10$, $\mu_2/\mu_1 = 10$.

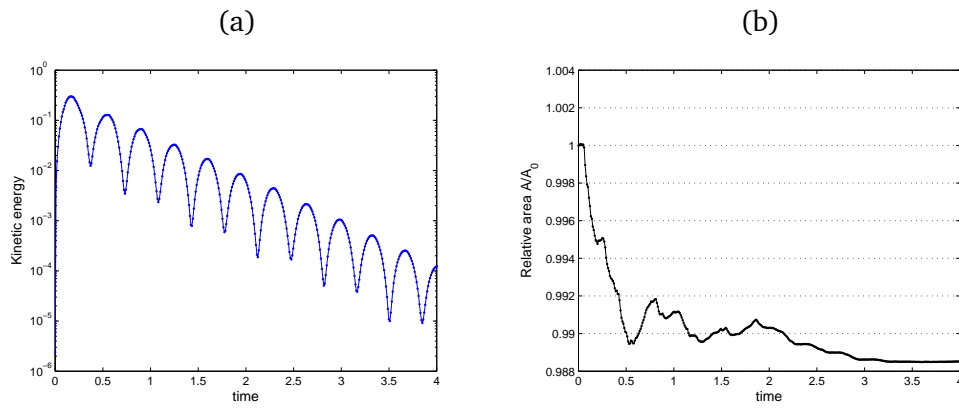


Figure 8: A relaxing bubble with $Re = 50$, $Bo = 10$, $\rho_2/\rho_1 = 10$, $\mu_2/\mu_1 = 10$: (a) Fluid kinetic energy versus time and (b) relative change of bubble area versus time.

errors at $t = 1$. The results are displayed in Table 3, indicating also near second order accuracy.

To validate our method, we also compare our results with those by Olsson et al. [22], which were obtained by a finite element discretization and a conservative level

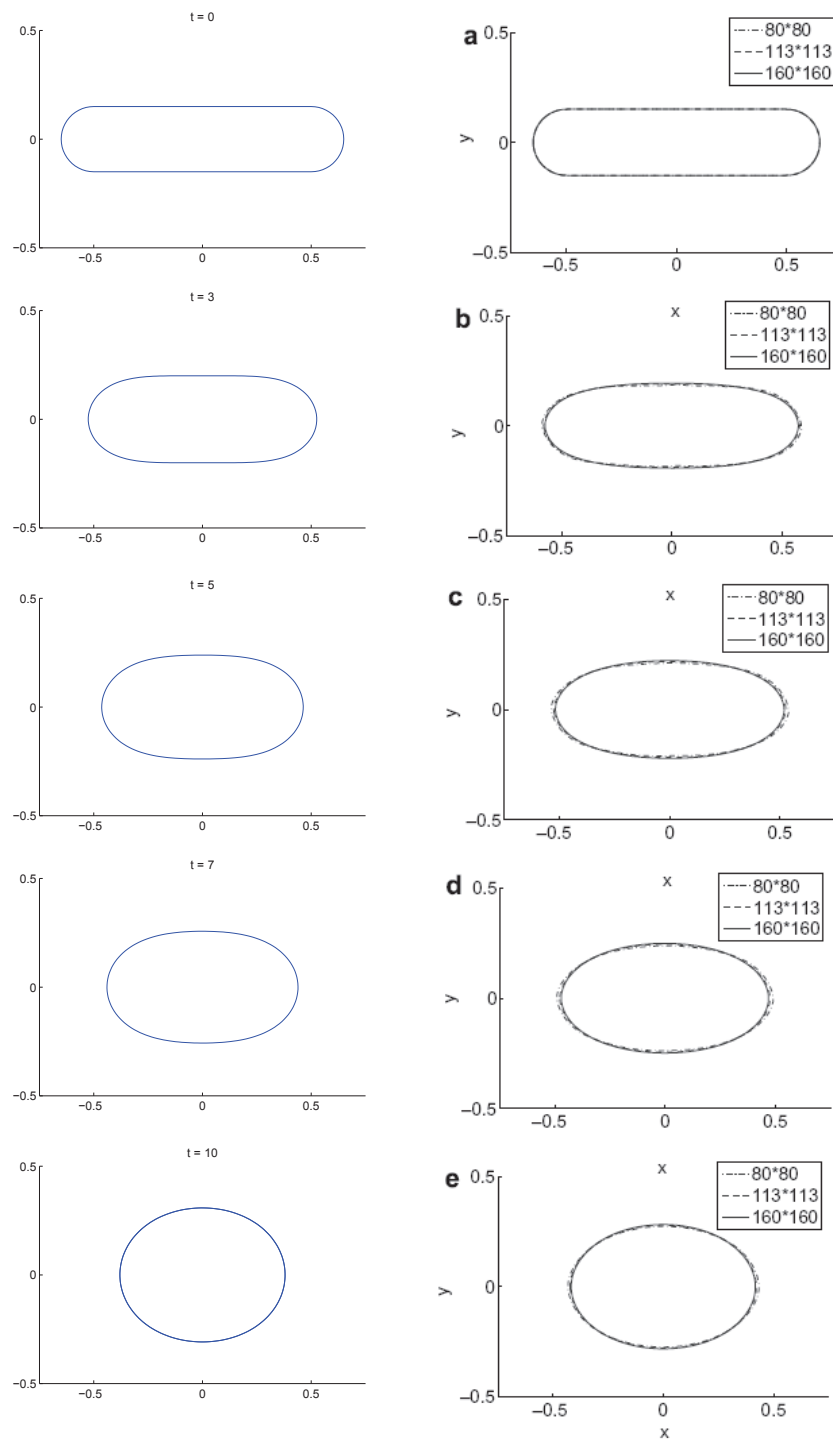


Figure 9: Comparison of the bubble shape at (a) $t = 0$, (b) $t = 3$, (c) $t = 5$, (d) $t = 7$, (e) $t = 10$ for an oscillating bubble with $Re = 1$, $Bo = 1$, $\rho_1 = 10$, $\rho_2 = 1$, $\mu_1 = 10$, $\mu_2 = 1$. Left: Our method, the IIM; Right: the Finite Element Method [22].

Table 3: Convergence analysis for a relaxing bubble with $Re = 50$, $Bo = 10$, $\rho_2/\rho_1 = 10$, $\mu_2/\mu_1 = 10$.

	u	order	v	order
E_1	4.78×10^{-3}	-	3.89×10^{-3}	-
E_2	1.42×10^{-3}	1.75	1.37×10^{-3}	1.50
E_3	4.29×10^{-4}	1.73	3.85×10^{-4}	1.84

set method. The test case has the non-dimensional parameters $Re = 1$, $Bo = 1$, $\rho_1 = 10$, $\rho_2 = 1$ and $\mu_1 = 10$, $\mu_2 = 1$. The domain is $[-1, 1] \times [-1, 1]$; and the grid size is 80×80 . The initial shape of the bubble was composed by a 1×0.3 rectangle and two half circles of diameter 0.3. Fig. 9 compares the bubble evolution between our simulation and the simulation by Olsson et al. [22]. The comparison of the bubble area variation is given in Fig. 10. The area loss is less than 0.16% in our simulation in the time interval $[0, 10]$.

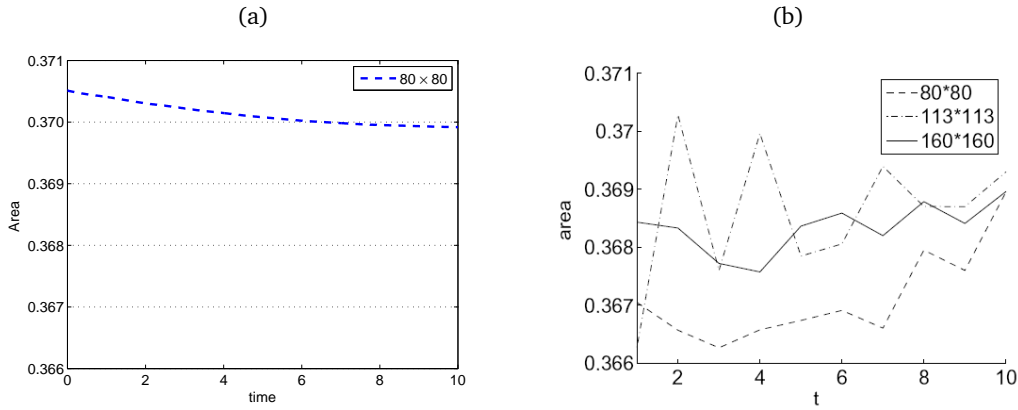


Figure 10: Comparison of the area conservation: (a) our method, the IIM; (b) the Finite Element Method [22].

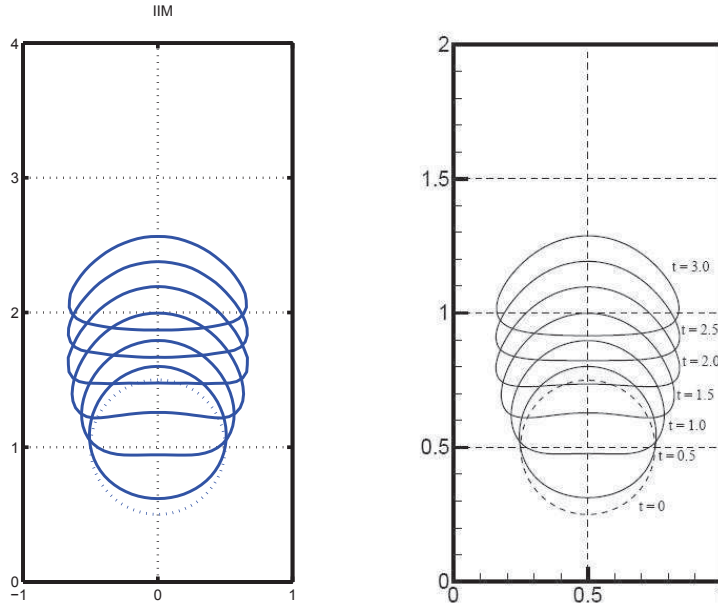
6.4. Example 4: Rising bubble

In the final test, we simulate a bubble rising in an initially quiescent fluid under buoyancy. The buoyant force is given as $\vec{G} = \rho \vec{g}$ in Eq. (2.1), where ρ is the piecewise constant density of the system, and \vec{g} is the gravitational acceleration. We chose the reference scales for this example as $L = d$, $U = \sqrt{gd}$, $\rho_0 = \rho_2$, $\mu_0 = \mu_2$, where d is the effective diameter of a bubble, and $g = |\vec{g}|$. The two-fluid flow due to a rising bubble can be characterized by four non-dimensional parameters, namely the density ratio $r_\rho = \rho_2/\rho_1$, the viscosity ratio $r_\mu = \mu_2/\mu_1$, the Reynolds number $Re = \frac{\rho_0 \sqrt{gd}^{3/2}}{\mu}$, and the Bond number $Bo = \frac{\rho g d^2}{\gamma}$.

Since an analytic solution is not available for this example, we estimate the order of accuracy using a grid refinement study as the example of an oscillating bubble in

Table 4: Convergence test for a rising bubble at $t = 1$ with $Bo = 1$, $Re = 10$, $r_\rho = 10$, $r_\mu = 10$.

	u	order	v	order
E_1	5.35×10^{-2}	-	3.89×10^{-2}	-
E_2	1.50×10^{-2}	1.83	1.34×10^{-2}	1.53
E_3	4.30×10^{-3}	1.81	3.20×10^{-3}	2.07

Figure 11: Comparison of the bubble evolution at $Bo = 10$, $Re_2 = 35$, $\rho_2/\rho_1 = 10$, $\mu_2/\mu_1 = 10$. (a) Our method, the IIM; (b) the Finite Element/Level Set Method [12].

Section 6.3. We use four grid resolutions corresponding to $N_1 = 32$, $N_2 = 64$, $N_3 = 128$, $N_4 = 256$. In the present convergence test, the domain is $[-1, 1] \times [0, 4]$ and the parameters are $Bo = 1$, $Re_2 = 10$, $r_\rho = 10$, $r_\mu = 10$. The results are displayed in Table 4, indicating near second order accuracy again.

To validate our method, we compare our simulation results with those by Hysing [12], which were obtained by combining the finite element method and the level set method. The test case has non-dimensional parameters $r_\rho = 10$, $r_\mu = 10$, $Re = 35$, $Bo = 10$, and dimensional parameters $\rho_0 = \rho_2 = 1000[\text{kg}/\text{m}^3]$, $\mu_0 = \mu_2 = 10[\text{kg}/(\text{m}\cdot\text{s})]$, $g = 0.98[\text{m}/\text{s}^2]$, $\gamma = 24.5[\text{kg}/\text{s}^2]$ and $d = 0.5[\text{m}]$. The rectangular domain is $[-1, 1] \times [0, 4]$, and the initial interface is circular with the radius $d/2$. Besides the visual comparison of the bubble shape, we also use the centroid, circularity and rise velocity of the bubble for quantitative comparison. The results are given in Fig. 12 and they agree very well with the ones by Hysing [12].

The simulated terminal shapes of a single rising bubble at a range of Reynolds and Bond numbers are shown in Table 5. The density and viscosity ratios are kept to be

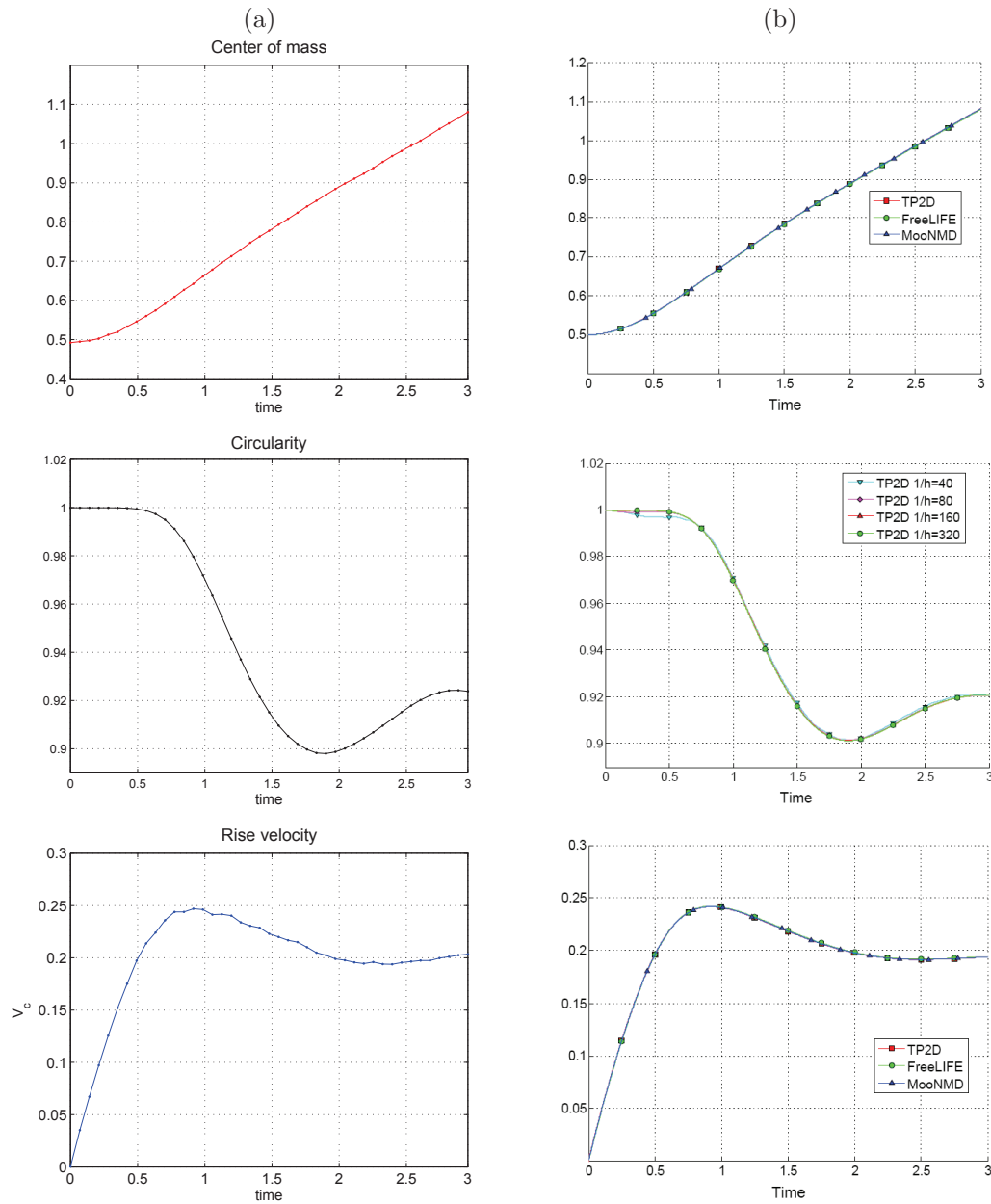












































Figure 12: Comparison of the center of mass, circularity and rise velocity of a rising bubble: (a) our method, the IIM; (b) the Finite Element/Level Set Method [12].

$r_\rho = 1000$ and $r_\mu = 100$ in these simulations. The results indicate that larger Reynolds numbers and Bond numbers cause larger deformation of the bubble [10]. We note that the method has numerical instability with high Reynolds numbers or Bond numbers in the table. This is the subject of future research.

Table 5: Bubble shapes at time $t = 5$ at different Reynolds and Bond numbers for $r_\rho = 1000$ and $r_\mu = 100$. x-mark: The corresponding simulation is unstable.

		Bo						
		0.5	1	5	10	20	50	100
Re	1							
	5							 x
	10							 x
	20							 x
	50							 x
	100				 x	 x	 x	 x

7. Conclusions

In this paper, we develop the immersed interface method (IIM) for simulating a two-fluid flow. The IIM uses a fixed Cartesian grid to solve the flow. The effect of the two-fluid interface is captured as jump contributions in a numerical scheme. The jump contributions involve the jump conditions of all the first-order and second-order Cartesian derivatives of the velocity and pressure. These Cartesian jump conditions can be computed from the analytical principal jump conditions. In particular, we have employed an interpolation formula to interpolate the interface velocity, one-sided finite difference approximations to compute the Cartesian jump conditions for the velocity, and the augmented variable approach to compute the Cartesian jump conditions for the pressure. We implement the IIM with the MAC scheme and explicitly integrate the discrete momentum equations in time using a Runge-Kutta method. We track the interface using Lagrangian markers that are connected by periodic cubic splines.

We have conducted several numerical experiments to investigate the performance of our method. The results have demonstrated that the method: (1) is of second-order accuracy in the infinity norm; (2) generates very few parasitic spurious currents in a flow; (3) conserves mass in a non-penetration closed interface; (4) works well for large ranges of density ratio (up to 1000) and viscosity ratio (up to 1000) in the parameter region $1 < Re < 50$ and $0.5 < Bo < 50$; and (5) suffers from numerical instability for a relatively large Reynolds number ($Re > 100$) or Bond number ($Bo > 50$) in the dynamic problem of a rising bubble. In the future, we will improve the method to handle large deformation and topological change of an interface, which may relate to the observed numerical instability. We are currently working on the change of the

interface tracking to the interface capturing by the level set, and the employment of implicit schemes to improve numerical stability.

Acknowledgments S. Xu thanks the support of this work by the NSF grant DMS 0915237.

Appendix: Approximation of the interface velocity

The interpolation formula for U_i gives an implicit equation for U_i as

$$U_i = \frac{\mu_1 u_i^{-\delta n} + \mu_2 u_i^{+\delta n}}{\mu_1 + \mu_2} + \mathcal{O}(\delta n^2) - \frac{1}{\mu_1 + \mu_2} \left(-f_\tau \tau_i - [\mu] \left(\left(\frac{\partial \vec{U}}{\partial \boldsymbol{\tau}} \cdot \mathbf{n} \right) \tau_i + \left(\frac{\partial \vec{U}}{\partial \boldsymbol{\tau}} \cdot \boldsymbol{\tau} \right) n_i \right) \right) \delta n. \quad (\text{A.1})$$

Let $\vec{U}^m = (U^m, V^m)$ be interface velocity at the Lagrangian marker m . Assuming $f_\tau = 0$, the above equation can be broken into

$$U^m = U_{initial}^m + \frac{[\mu]}{\mu_1 + \mu_2} \left(\left(\frac{\partial U^m}{\partial \boldsymbol{\tau}} n_1 + \frac{\partial V^m}{\partial \boldsymbol{\tau}} n_2 \right) \tau_1 + \left(\frac{\partial U^m}{\partial \boldsymbol{\tau}} \tau_1 + \frac{\partial V^m}{\partial \boldsymbol{\tau}} \tau_2 \right) n_1 \right),$$

$$V^m = V_{initial}^m + \frac{[\mu]}{\mu_1 + \mu_2} \left(\left(\frac{\partial U^m}{\partial \boldsymbol{\tau}} n_1 + \frac{\partial V^m}{\partial \boldsymbol{\tau}} n_2 \right) \tau_2 + \left(\frac{\partial U^m}{\partial \boldsymbol{\tau}} \tau_1 + \frac{\partial V^m}{\partial \boldsymbol{\tau}} \tau_2 \right) n_2 \right),$$

where

$$U_{initial}^m = \frac{\mu_1 (u_1^{-\delta n})^m + \mu_2 (u_1^{+\delta n})^m}{\mu_1 + \mu_2},$$

$$V_{initial}^m = \frac{\mu_1 (u_2^{-\delta n})^m + \mu_2 (u_2^{+\delta n})^m}{\mu_1 + \mu_2}.$$

Using the fact that

$$\frac{\partial U^m}{\partial \boldsymbol{\tau}} = \frac{\partial U^m}{\partial \alpha},$$

where α is the arc length parameter. We obtain the following system of ODEs

$$\begin{pmatrix} U^m \\ V^m \end{pmatrix} + \begin{pmatrix} c_1^m & c_2^m \\ c_2^m & c_3^m \end{pmatrix} \frac{\partial}{\partial \alpha} \begin{pmatrix} U^m \\ V^m \end{pmatrix} = \begin{pmatrix} U_{initial}^m \\ V_{initial}^m \end{pmatrix}, \quad (\text{A.2})$$

where

$$c_1^m = -\frac{[\mu]}{\mu_1 + \mu_2} (2n_1 \tau_1) \delta n,$$

$$c_2^m = -\frac{[\mu]}{\mu_1 + \mu_2} (n_2 \tau_1 + n_1 \tau_2) \delta n,$$

$$c_3^m = -\frac{[\mu]}{\mu_1 + \mu_2} (2n_2 \tau_2) \delta n.$$

Using central finite differences to approximate the derivatives,

$$\begin{aligned} \frac{\partial U^m}{\partial \alpha} &= \frac{1}{2\Delta\alpha} (U^{m+1} - U^{m-1}) + \mathcal{O}(\Delta\alpha^2), \\ \frac{\partial V^m}{\partial \alpha} &= \frac{1}{2\Delta\alpha} (V^{m+1} - V^{m-1}) + \mathcal{O}(\Delta\alpha^2), \end{aligned}$$

we obtain

$$\begin{aligned} (-k_1^m U^{m-1} + U^m + k_1^m U^{m+1}) + (-k_2^m V^{m-1} + k_2^m V^{m+1}) &= U_{initial}^m, \\ (-k_2^m U^{m-1} + k_2^m U^{m+1}) + (-k_3^m V^{m-1} + V^m + k_3^m V^{m+1}) &= V_{initial}^m, \end{aligned}$$

where

$$k_1^m = \frac{c_1^m}{2\Delta\alpha}, \quad k_2^m = \frac{c_2^m}{2\Delta\alpha}, \quad k_3^m = \frac{c_3^m}{2\Delta\alpha}.$$

Let N_s be the number of Lagrangian markers and $m = 0, \dots, N_s - 1$. Define

$$\begin{aligned} U &= [U^0, U^1, U^2, \dots, U^{N_s-1}]^T, \\ V &= [V^0, V^1, V^2, \dots, V^{N_s-1}]^T. \end{aligned}$$

Then we have the following linear system for the interface velocity

$$A \begin{pmatrix} U \\ V \end{pmatrix} = \begin{pmatrix} U_{initial} \\ V_{initial} \end{pmatrix},$$

where

$$A = \begin{pmatrix} \begin{array}{cccccc|cccccc} 1 & k_1^0 & & & & -k_1^0 & & & & & & \\ -k_1^1 & 1 & k_1^1 & & & & & & & & & -k_2^0 \\ & -k_1^2 & 1 & k_1^2 & & & & & & & & \\ & & \ddots & \ddots & \ddots & & & & & & & \\ & & & -k_1^{N_s-2} & 1 & k_1^{N_s-2} & & & & & & \\ k_1^{N_s-1} & & & & k_1^{N_s-1} & 1 & & & & & & \\ \hline 0 & k_2^0 & & & & -k_2^0 & & & & & & \\ -k_2^1 & 0 & k_2^1 & & & & & & & & & \\ & -k_2^2 & 0 & k_2^2 & & & & & & & & \\ & & \ddots & \ddots & \ddots & & & & & & & \\ & & & -k_2^{N_s-2} & 0 & k_2^{N_s-2} & & & & & & \\ k_2^{N_s-1} & & & & k_2^{N_s-1} & 0 & & & & & & \\ \hline & & & & & & k_3^0 & & & & & -k_3^0 \\ & & & & & & 1 & k_3^1 & & & & \\ -k_3^1 & & & & & & -k_3^1 & 1 & k_3^1 & & & \\ & & & & & & -k_3^2 & & 1 & k_3^2 & & \\ & & & & & & & \ddots & \ddots & \ddots & & \\ & & & & & & & -k_3^{N_s-2} & & 1 & k_3^{N_s-2} & \\ k_3^{N_s-1} & & & & & & k_3^{N_s-1} & & & k_3^{N_s-1} & 1 & \end{array} \end{pmatrix}.$$

References

- [1] E. Aulisa, S. Manservigi and R. Scardovelli, *A mixed marker and volume of fluid method for the reconstruction and advection of interfaces in two-phase and free-boundary flows*, J. Comput. Phys., 188, 2003.
- [2] G. K. Batchelor, *An Introduction to Fluid Dynamics*, Cambridge University Press, 2000.
- [3] W. E and J.-G. Liu, *Vorticity boundary conditions and related issues for finite differences schemes*, J. Compt. Phy., 124, pp. 368-382, 1996.

- [4] D. Enright, R. Fedkiw, J. Ferziger and I. Mitchell, *A hybrid particle level set method for improved interface capturing*, J. Comput. Phys. 183, pp. 83-116, 2002.
- [5] C. Farhat, A. Rallu and S. Shankaran, *A higher-order generalized ghost fluid method for the poor for the three-dimensional two-phase flow computation of underwater implosions*, J. Comput. Phys., 227, pp. 7674-7700, 2008.
- [6] J. M. Floryan and H. Rasmussen, *Numerical methods for viscous flows with moving boundaries*, Appl. Mech. Rev. 42(12), 323, 1989.
- [7] M. Hayashi, K. Hatanaka and M. Kawahara, *Lagrangian finite element method for free surface Navier-Stokes flows using fractional step methods.*, Int. J. Num. Meth. Fluids, 13, pp. 805-840, 1991.
- [8] Marcus Herrmann, *Two-phase flow tutorial*, Center for Turbulence Research, Stanford University, 2006.
- [9] C. W. Hirt and B. D. Nichols, *Volume of Fluid (VOF) method for the dynamics of free boundaries*, J. Comput. Phys. 39, pp. 201-225, 1981.
- [10] J. Hua and J. Lou, *Numerical simulation of bubble rising in viscous liquid*, J. Comput. Phys., 222, pp.769-795, 2007.
- [11] H. Huang and Z. Li, *Convergence analysis of the immersed interface method*, Journal of Numerical Analysis, 19, pp.583-608, 1999
- [12] S. Hysing, *Numerical Simulation of Immiscible Fluids with FEM Level Set Techniques*, Doctoral dissertation, Dem Fachbereich Mathematik der Universitat Dortmund, 2007.
- [13] H. Johnston and J.-G. Liu, *Finite differences schemes for incompressible flow based on local pressure boundary conditions*, J. Comput. Phys., 180, pp. 120-154, 2002.
- [14] H. Johnston and J.-G. Liu, *Accurate, stable and efficient Navier-Stokes solvers based on explicit treatment of the pressure term*, J. Comput. Phys., 199, pp. 221-259, 2004.
- [15] M. Kang, R. P. Fedkiw and X.-D. Liu, *A boundary condition capturing method for multi-phase incompressible flow*, J. of Scient. Comput., 15, pp 323-360, 2000.
- [16] Ming-Chih Lai and Hsiao-Chieh Tseng *A simple implementation of the immersed interface methods for Stokes flows with singular forces*, Computer and fluids, 37, pp.99-106, 2008.
- [17] R. LeVeque and Z. Li, *The immersed interface method for elliptic equations with discontinuous coefficients and singular forces*, SIAM J. Numer. Anal., 31, pp 1019-1044, 1994.
- [18] Z. Li and M.-C. Lai, *The immersed interface method for the Navier-Stokes equations with singular forces*, J. Comput. Phys., 171, 822, 2001.
- [19] Zhilin Li, and K. Ito, *An augmented approach for the pressure boundary condition in a Stokes flow*, Comm. Comput. Phys., vol 1, pp 874-885, 2006.
- [20] Zhilin Li and Kazufumi Ito, *The Immersed Interface Method – Numerical Solutions of PDEs Involving Interfaces and Irregular Domains*, SIAM Frontiers in Applied mathematics, 33, ISBN: 0-89971-609-8, 2006.
- [21] T. Okamoto and M. Kawahara, *Two-dimensional sloshing analysis by Lagrangian finite element method*, Int. J. Num. Meth. Fluids, 11, pp. 453-477, 1990.
- [22] E. Olsson, G. Kreiss and S. Zahedi, *A conservative level set method for two phase flow II*, J. Comput. Phys., 225, pp.785-807, 2007.
- [23] S. Osher and R. P. Fedkiw, *Level set method: an overview and some recent results*, J. Comput. Phys., 169, pp. 463-502, 2001.
- [24] C. S. Peskin, *Flow patterns around heart valves: a numerical method*, J. Comput. Phys., 10, pp. 252-271, 1972.
- [25] C. S. Peskin, *The immersed boundary method*, Acta Numerica., 11, pp. 479-517, 2002.
- [26] A. Prosperetti and G. Tryggvason, *Computational methods for multiphase flow*, Cambridge, 2007.

- [27] R. Scardovelli and S. Zaleski, *Direct numerical simulation of free-surface and interfacial flow*, *Annu. Rev. Fluid Mech.*, 31, pp. 567-603, 1999.
- [28] J. A. Sethian and P. Smereka, *Level set methods for fluids interfaces*, *Annu. Rev. Fluid Mech.*, 35, pp. 341-372, 2003.
- [29] S. Shin and D. Juric, *Modeling three dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity*, *J. Comput. Phys.*, 180, 2002.
- [30] A. Smolianski, *Numerical modeling of two-fluid interfacial flows*, Doctoral dissertation, University of Jyvaskyla, 2001.
- [31] M. Sussmann, *A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles*, *J. Comput. Phys.*, 187, pp. 110-136, 2003.
- [32] Z. Tan, D.V. Le, Z. Li, K.M. Lim and B.C. Khoo, *An immersed interface method for solving incompressible viscous flows with piecewise constant viscosity across a moving elastic membrane*, *J. Comput. Phys.*, 227, pp. 9955-9983, 2008.
- [33] G. Tryggvason et al, *A front-tracking method for computations of multiphase flows*, *J. Comput. Phys.*, 169, pp. 708-759, 2001.
- [34] S. O. Unverdi and G. Tryggvason, *A front-tracking method for viscous, incompressible, multi-fluid flows*. *J. Comput. Phys.*, 100:25-37, 1992.
- [35] Sheng Xu and Z. Jane Wang, *Systematic Derivation of Jump Conditions for the Immersed Interface Method in Three-Dimensional Flow Simulation*, *SIAM J. Sci. Comput.*, vol 27, no 6, pp. 1948-1980, 2006.
- [36] Sheng Xu and Z. Jane Wang, *An immersed interface method for simulating the interaction of a fluid with moving boundaries*, *J. Comput. Phys.*, vol 216, no 2, pp.454-493, 2006.
- [37] Sheng Xu, *Derivation of principal jump conditions for the immersed interface method in two-fluid flow simulation*, *Discrete and Continuous Dynamical Systems-Supplement*, pp. 838-845, 2009.
- [38] Sheng Xu, *An iterative two-fluid pressure solver based on the immersed interface method*, *Commun. Comput. Phys.*, Vol. 12, No. 2, pp. 528-543, 2011.
- [39] Y. Zhang, Q. Zou, D. Greaves, D. Reeve, A. Hunt-Raby, D. Graham, P. James and X. Lv, *A level set immersed boundary method for water entry and exit*, *Commun. Comput. Phys.*, 2010
- [40] Y. C. Zhou, J. Liu and D. L. Harry *A matched interface and boundary method for solving multi-flow Navier-Stokes equations with applications to geodynamics*, *J. Comput. Phys.*, 231, pp. 223-242, 2012.