INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
*Int. J. Numer. Meth. Fluids* 2012; **69**:842–858
Published online 24 June 2011 in Wiley Online Library (wileyonlinelibrary.com/journal/nmf). DOI: 10.1002/fld.2616

# Volume preserving immersed boundary methods for two-phase fluid flows

Yibao Li[1], Eunok Jung[2], Wanho Lee[2], Hyun Geun Lee[1] and Junseok Kim[1,*,†]

[1] *Department of Mathematics, Korea University, Seoul, Republic of Korea*
[2] *Department of Mathematics, Konkuk University, Seoul, Republic of Korea*

## SUMMARY

In this article, we propose a simple area-preserving correction scheme for two-phase immiscible incompressible flows with an immersed boundary method (IBM). The IBM was originally developed to model blood flow in the heart and has been widely applied to biofluid dynamics problems with complex geometries and immersed elastic membranes. The main idea of the IBM is to use a regular Eulerian computational grid for the fluid mechanics along with a Lagrangian representation of the immersed boundary. Using the discrete Dirac delta function and the indicator function, we can include the surface tension force, variable viscosity and mass density, and gravitational force effects. The principal advantage of the IBM for two-phase fluid flows is its inherent accuracy due in part to its ability to use a large number of interfacial marker points on the interface. However, because the interface between two fluids is moved in a discrete manner, this can result in a lack of volume conservation. The idea of an area preserving correction scheme is to correct the interface location normally to the interface so that the area remains constant. Various numerical experiments are presented to illustrate the efficiency and accuracy of the proposed conservative IBM for two-phase fluid flows. Copyright © 2011 John Wiley & Sons, Ltd.

Received 16 December 2010; Revised 17 March 2011; Accepted 29 April 2011

KEY WORDS: immersed boundary method; area preserving; two-phase fluid flow; multigrid method; indicator function; finite difference

## 1. INTRODUCTION

Many important industrial problems involve flows with multiple constitutive components. Because of their inherent nonlinearities and the complexity of dealing with unknown moving interfaces, multiphase flows are challenging. There are many ways to model moving interfaces. The two main approaches for simulating multiphase and multicomponent flows are interface tracking and interface capturing methods. In interface tracking methods (front-tracking [1, 2], immersed interface [3, 4], and immersed boundary [5–7]), Lagrangian particles are used to track the interfaces. In interface capturing methods, such as level-set [8, 9] and phase-field methods [10, 11], the interface is implicitly captured by a contour of a particular scalar function. The numerical method we will take is the immersed boundary method (IBM), which was originally developed by Peskin [5]. The IBM has been applied to two-phase fluid flows [12–14]. The motion of the fluid is influenced by the force generated by the interface and the interface moves with the local fluid velocity. The strength of this method is that it can accurately handle the complicated and time-dependent geometry of the interface.

*Correspondence to: Junseok Kim, Department of Mathematics, Korea University, Seoul 136-701, Republic of Korea.

†E-mail: cfdkim@korea.ac.kr

Contract/grant sponsor: National Research Foundation of Korea

Consider a viscous incompressible fluid that fills a rectangular domain $\Omega$ and an interface $\Gamma$ that is contained in the domain. The two-dimensional example setup with an immersed boundary curve is shown in Figure 1. We shall now consider the mathematical formulation of the equations of motion for the immiscible two-phase fluid. Let $\rho$ be the variable density and $\mu$ be the variable viscosity. The equations of motion of the mixture are then as follows:

$$\begin{aligned}
\rho(I)\left(\frac{\partial \mathbf{u}(\mathbf{x},t)}{\partial t} \right. &+ \left. \mathbf{u}(\mathbf{x},t)\cdot\nabla\mathbf{u}(\mathbf{x},t)\right) = -\nabla p(\mathbf{x},t) \\
&+ \nabla\cdot[\mu(I)(\nabla\mathbf{u}(\mathbf{x},t)+\nabla\mathbf{u}(\mathbf{x},t)^T)] + \mathbf{F}(\,\mathbf{x},t) + \rho(I)\mathbf{g}, \qquad (1)\\
\nabla\cdot\mathbf{u}(\mathbf{x},t) &= 0,\\
\mathbf{F}(\mathbf{x},t) &= \int_\Gamma \mathbf{f}(s,t)\delta^2(\mathbf{x}-\mathbf{X}(s,t))\mathrm{d}s,\\
\mathbf{f}(s,t) &= \sigma\frac{\partial^2\mathbf{X}(s,t)}{\partial s^2},\\
\frac{\partial\mathbf{X}(s,t)}{\partial t} &= \mathbf{U}(s,t), \qquad (2)\\
\mathbf{U}(s,t) &= \int_\Omega \mathbf{u}(\mathbf{x},t)\delta^2(\mathbf{x}-\mathbf{X}(s,t))\mathrm{d}\mathbf{x}. \qquad (3)
\end{aligned}$$

The fluid velocity $\mathbf{u}(\mathbf{x},t)$, fluid pressure $p(\mathbf{x},t)$, and singular surface tension force density $\mathbf{F}(\mathbf{x},t)$ are functions of $(\mathbf{x},t)$, where $\mathbf{x}=(x,y)$ are Cartesian coordinates. $t$ is the time and $\mathbf{g}$ is the gravity. The configuration of the immersed boundary is described by the function $\mathbf{X}(s,t)$, where $0\leqslant s\leqslant L$ and $L$ is the length of the boundary (see Figure 1). The boundary force density $\mathbf{f}(s,t)$ and the boundary velocity $\mathbf{U}(s,t)$ are also functions of $s$ and $t$. The core of the immersed boundary method is the delta function, which describes the interaction between the fluid and the immersed boundary. Discontinuous material properties can easily be accommodated through the numerical construction of an indicator function, $I(\mathbf{x},t)$. Let us define a gradient field,

$$\nabla I(\mathbf{x},t) = \int_\Gamma \mathbf{n}(\mathbf{X}(s,t))\delta^2(\mathbf{x}-\mathbf{X}(s,t))\mathrm{d}s, \qquad (4)$$

which is zero except near the interface. To find the indicator function, the Poisson's equation

$$\Delta I(\mathbf{x},t) = \nabla\cdot\int_\Gamma \mathbf{n}(\mathbf{X}(s,t))\delta^2(\mathbf{x}-\mathbf{X}(s,t))\mathrm{d}s$$

is solved. Then, the variable fluid properties, $\rho$ and $\mu$, can be represented by

$$\rho(I(\mathbf{x},t)) = \rho_1+(\rho_2-\rho_1)I(\mathbf{x},t) \text{ and } \mu(I(\mathbf{x},t)) = \mu_1+(\mu_2-\mu_1)I(\mathbf{x},t),$$

where $\rho_i$ and $\mu_i$ for $i=1,2$ are density and viscosity of fluid $i$, respectively. There is no guarantee that the advected immersed boundary preserves area over time. Area conservation is an important

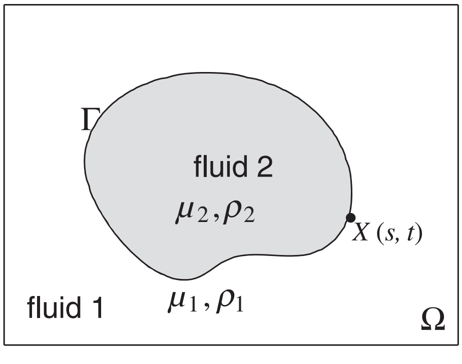


Figure 1. The immersed boundary curve $\Gamma$ on a rectangular domain $\Omega$.

issue in modeling free interface problems. If the area loss occurs, it could increase the local curvature of the interface and result in overestimating the surface tension force. An overestimated surface tension force induces an incorrect velocity field, which moves the interface to the wrong position.

When an exact projection method is used for solving the Navier–Stokes equations, the velocity field on the Eulerian grid will be discretely divergence free, but this does not guarantee that the interpolated velocity field through the delta function is continuously divergence free. This can result in volume loss [15]. This is particularly problematic for interfaces under tension, which have been shown to exhibit volume loss with the IBM [16–20]. A solution for fixing the volume loss problem is using the modified divergence stencils of Peskin and Printz [19]. The key idea is the introduction of a new finite difference divergence operator that is constructed in such a way that the interpolated velocity field in which the immersed boundary moves is more nearly divergence free.

In this paper, we propose a simple area preserving correction scheme for two-phase immiscible incompressible flows with an IBM. The idea of an area preserving correction scheme is to correct the interface location normally to the interface so that the area remains constant. The rest of the paper is organized as follows. In Section 2, the numerical method will be introduced. The experimental results will be discussed in Section 3. Finally, some conclusions will be drawn in Section 4.

## 2. NUMERICAL METHOD

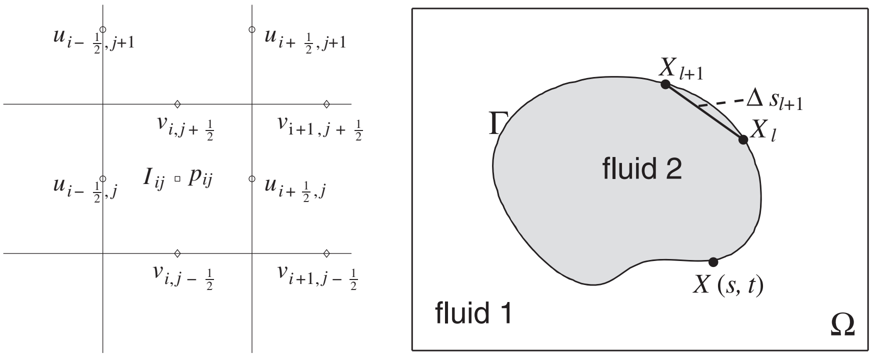
In this section, we present the numerical solution algorithm. A staggered marker-and-cell (MAC) mesh of Harlow and Welch [21] is used in which the pressure and indicator function are stored at cell centers and velocities at cell interfaces (see Figure 2).

Let a computational domain be partitioned in Cartesian geometry into a uniform mesh with mesh spacing $h$. The center of each cell, $\Omega_{ij}$, is located at $\mathbf{x}_{ij} = (x_i, y_j) = ((i-0.5)h, (j-0.5)h)$ for $i = 1, \cdots, N_x$ and $j = 1, \cdots, N_y$. $N_x$ and $N_y$ are the numbers of cells in the $x$ direction and $y$ direction, respectively. We use a set of $M$ Lagrangian points $\mathbf{X}_l^n = (x_l^n, y_l^n)$ for $l = 1, \ldots, M$ to discretize the immersed boundary with a representative line segment $\Delta s_{l+1} = \sqrt{(x_{l+1} - x_l)^2 + (y_{l+1} - y_l)^2}$.

At the beginning of each time step, given a divergence free velocity field $\mathbf{u}^n$ and a given boundary configuration $\mathbf{X}^n$, we want to find $\mathbf{u}^{n+1}$, $p^{n+1}$, and $\mathbf{X}^{n+1}$, which solve the following semi-implicit scheme:

$$\begin{aligned} \rho^n \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} &= -\rho^n \mathbf{u}^n \cdot \nabla_d \mathbf{u}^n - \nabla_d p^{n+1} + \mu \Delta_d \mathbf{u}^n + \mathbf{F}^n + \rho^n \mathbf{g}, \\ \nabla_d \cdot \mathbf{u}^{n+1} &= 0, \end{aligned}$$

where $\rho^n = \rho_1 + (\rho_2 - \rho_1) I^n$ and $\mathbf{g} = (0, -g)$.



(a) (b)

Figure 2. (a) Velocities are defined at cell boundaries and the pressure and indicator function are defined at the cell centers. (b) Immersed boundary points.

DOI: 10.1002/fld

*Step 1*. Find the force $\mathbf{f}^n$ on the immersed boundary from the given boundary configuration $\mathbf{X}^n$. For $l = 1, \dots, M$,

$$\mathbf{f}_l^n = \frac{\sigma}{\Delta s_{l+1/2}} \left( \frac{\mathbf{X}_{l+1}^n - \mathbf{X}_l^n}{\Delta s_{l+1}} - \frac{\mathbf{X}_l^n - \mathbf{X}_{l-1}^n}{\Delta s_l} \right), \tag{5}$$

where $\Delta s_{l+1/2} = (\Delta s_l + \Delta s_{l+1})/2$ and $\sigma$ is a surface tension coefficient. Note that the subscript arithmetic on $l$ in Equation (5) has to be interpreted in a periodic sense, because the boundary is closed: if $l = M$, then $l + 1 = 1$; if $l = 1$, then $l - 1 = M$.

*Step 2*. Spread the boundary force into the nearby lattice points of the fluid.

$$\mathbf{F}_{ij}^n = \sum_{l=1}^{M} \mathbf{f}_l^n \delta_h^2(\mathbf{x}_{ij} - \mathbf{X}_l^n) \Delta s_{l+1/2} \quad \text{for } i = 1, \dots, N_x \text{ and } j = 1, \dots, N_y,$$

where $\delta_h^2$ is a smoothed approximation to the two-dimensional Dirac delta function:

$$\delta_h^2(\mathbf{x}) = \frac{1}{h^2} \phi\left(\frac{x}{h}\right) \phi\left(\frac{y}{h}\right),$$

where

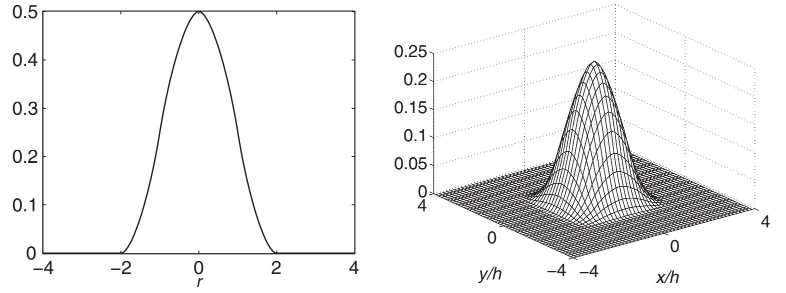
$$\phi(r) = \begin{cases} \frac{3-2|r|+\sqrt{1+4|r|-4r^2}}{8} & \text{if } |r| \leqslant 1, \\ \frac{5-2|r|-\sqrt{-7+12|r|-4r^2}}{8} & \text{if } 1 < |r| \leqslant 2, \\ 0 & \text{if } 2 < |r|. \end{cases}$$

One-dimensional and two-dimensional Dirac delta functions are shown in Figure 3. The motivation for this particular choice of $\phi(r)$ is given in [22].

*Step 3*. Solve the Navier–Stokes equations on the rectangular lattice to obtain $\mathbf{u}^{n+1}$ and $p^{n+1}$ from $\mathbf{u}^n$ and $\mathbf{X}^n$.

Solve an intermediate velocity field, $\tilde{\mathbf{u}}$, which generally does not satisfy the incompressible condition, without the pressure gradient term,

$$\frac{\tilde{\mathbf{u}} - \mathbf{u}^n}{\Delta t} + \mathbf{u}^n \cdot \nabla_d \mathbf{u}^n = \frac{\mu}{\rho^n} \Delta_d \mathbf{u}^n + \frac{1}{\rho^n} \mathbf{F}^n + \mathbf{g}.$$



(a) (b)

Figure 3. (a) One-dimensional and (b) two-dimensional Dirac delta functions.

The resulting finite difference equations are written out explicitly. They take the form

$$\begin{aligned}
\tilde{u}_{i+\frac{1}{2},j} &= u^n_{i+\frac{1}{2},j} - \Delta t(uu_x + vu_y)^n_{i+\frac{1}{2},j} + \frac{\Delta t}{\rho^n_{i+\frac{1}{2},j}} F^{x-\text{edge}}_{i+\frac{1}{2},j} \\
&\quad + \frac{\mu \Delta t}{h^2 \rho^n_{i+\frac{1}{2},j}} \left( u^n_{i+\frac{3}{2},j} + u^n_{i-\frac{1}{2},j} - 4u^n_{i+\frac{1}{2},j} + u^n_{i+\frac{1}{2},j+1} + u^n_{i+\frac{1}{2},j-1} \right), \\
\tilde{v}_{i,j+\frac{1}{2}} &= v^n_{i,j+\frac{1}{2}} - \Delta t(uv_x + vv_y)^n_{i,j+\frac{1}{2}} + \frac{\Delta t}{\rho^n_{i,j+\frac{1}{2}}} F^{y-\text{edge}}_{i,j+\frac{1}{2}} - g\Delta t \\
&\quad + \frac{\mu \Delta t}{h^2 \rho^n_{i,j+\frac{1}{2}}} \left( v^n_{i,j+\frac{3}{2}} + v^n_{i,j-\frac{1}{2}} - 4v^n_{i,j+\frac{1}{2}} + v^n_{i+1,j+\frac{1}{2}} + v^n_{i-1,j+\frac{1}{2}} \right),
\end{aligned}$$

where the advection terms, $(uu_x + vu_y)^n_{i+\frac{1}{2},j}$ and $(uv_x + vv_y)^n_{i,j+\frac{1}{2}}$, are defined by

$$\begin{aligned}
(uu_x + vu_y)^n_{i+\frac{1}{2},j} &= u^n_{i+\frac{1}{2},j} \bar{u}^n_{x_{i+\frac{1}{2},j}} \\
&\quad + \frac{v^n_{i,j-\frac{1}{2}} + v^n_{i+1,j-\frac{1}{2}} + v^n_{i,j+\frac{1}{2}} + v^n_{i+1,j+\frac{1}{2}}}{4} \bar{u}^n_{y_{i+\frac{1}{2},j}}, \\
(uv_x + vv_y)^n_{i,j+\frac{1}{2}} &= v^n_{i,j+\frac{1}{2}} \bar{v}^n_{y_{i,j+\frac{1}{2}}} \\
&\quad + \frac{u^n_{i-\frac{1}{2},j} + u^n_{i-\frac{1}{2},j+1} + u^n_{i+\frac{1}{2},j} + u^n_{i+\frac{1}{2},j+1}}{4} \bar{v}^n_{x_{i,j+\frac{1}{2}}}.
\end{aligned}$$

The values $\bar{u}^n_{x_{i+\frac{1}{2},j}}$ and $\bar{u}^n_{y_{i+\frac{1}{2},j}}$ are computed using the upwind procedure. The procedure is

$$\bar{u}^n_{x_{i+\frac{1}{2},j}} = \begin{cases} \frac{u^n_{i+\frac{1}{2},j} - u^n_{i-\frac{1}{2},j}}{h} & \text{if } u^n_{i+\frac{1}{2},j} > 0 \\ \frac{u^n_{i+\frac{3}{2},j} - u^n_{i+\frac{1}{2},j}}{h} & \text{otherwise} \end{cases}$$

and

$$\bar{u}^n_{y_{i+\frac{1}{2},j}} = \begin{cases} \frac{u^n_{i+\frac{1}{2},j} - u^n_{i+\frac{1}{2},j-1}}{h} & \text{if } v^n_{i,j-\frac{1}{2}} + v^n_{i+1,j-\frac{1}{2}} + v^n_{i,j+\frac{1}{2}} + v^n_{i+1,j+\frac{1}{2}} > 0 \\ \frac{u^n_{i+\frac{1}{2},j+1} - u^n_{i+\frac{1}{2},j}}{h} & \text{otherwise.} \end{cases}$$

The quantities $\bar{v}^n_{x_{i,j+\frac{1}{2}}}$ and $\bar{v}^n_{y_{i,j+\frac{1}{2}}}$ are computed in a similar manner. Then, we solve the following equations for the advanced pressure field at $(n+1)$ time step.

$$\begin{aligned}
\frac{\mathbf{u}^{n+1} - \tilde{\mathbf{u}}}{\Delta t} &= -\frac{1}{\rho^n} \nabla_d p^{n+1}, && (6) \\
\nabla_d \cdot \mathbf{u}^{n+1} &= 0. && (7)
\end{aligned}$$

After applying the discrete divergence operator to Equation (6), we find the Poisson's equation for the pressure at the advanced time $(n+1)$

$$\nabla_d \cdot \left( \frac{1}{\rho^n} \nabla_d p^{n+1} \right) = \frac{1}{\Delta t} \nabla_d \cdot \tilde{\mathbf{u}}, \tag{8}$$

DOI: 10.1002/fld

where we have made use of Equation (7) and the terms are defined as follows:

$$\begin{aligned}\nabla_d\cdot\left(\frac{1}{\rho^n}\nabla_d p_{ij}^{n+1}\right) &= \frac{\frac{1}{\rho^n_{i+\frac{1}{2},j}}p^{n+1}_{i+1,j}+\frac{1}{\rho^n_{i-\frac{1}{2},j}}p^{n+1}_{i-1,j}+\frac{1}{\rho^n_{i,j+\frac{1}{2}}}p^{n+1}_{i,j+1}+\frac{1}{\rho^n_{i,j-\frac{1}{2}}}p^{n+1}_{i,j-1}}{h^2}\\ &\quad -\frac{\frac{1}{\rho^n_{i+\frac{1}{2},j}}+\frac{1}{\rho^n_{i-\frac{1}{2},j}}+\frac{1}{\rho^n_{i,j+\frac{1}{2}}}+\frac{1}{\rho^n_{i,j-\frac{1}{2}}}}{h^2}p^{n+1}_{ij},\\ \nabla_d\cdot\tilde{\mathbf{u}}_{ij} &= \frac{\tilde{u}_{i+\frac{1}{2},j}-\tilde{u}_{i-\frac{1}{2},j}}{h}+\frac{\tilde{v}_{i,j+\frac{1}{2}}-\tilde{v}_{i,j-\frac{1}{2}}}{h},\end{aligned}$$

where $\rho^n_{i+\frac{1}{2},j}=(\rho^n_{ij}+\rho^n_{i+1,j})/2$ and the other terms are similarly defined. The boundary condition for the pressure is

$$\mathbf{n}\cdot\nabla_d p^{n+1}=\mathbf{n}\cdot\left(-\rho^n\frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\Delta t}-\rho^n(\mathbf{u}\cdot\nabla_d\mathbf{u})^n+\mu\Delta_d\mathbf{u}^n+\mathbf{F}^n+\rho^n\mathbf{g}\right),$$

where **n** is the unit normal vector to the domain boundary.

The resulting linear system of Equation (8) is solved using a multigrid method [23], specifically, V-cycles with a Gauss–Seidel relaxation. Then the divergence-free normal velocities $u^{n+1}$ and $v^{n+1}$ are defined by

$$\begin{aligned}\mathbf{u}^{n+1} &= \tilde{\mathbf{u}}-\frac{\Delta t}{\rho^n}\nabla_d p^{n+1},\ \text{i.e.,}\\ u^{n+1}_{i+\frac{1}{2},j} &= \tilde{u}_{i+\frac{1}{2},j}-\frac{\Delta t}{\rho^n_{i+\frac{1}{2},j}h}(p_{i+1,j}-p_{ij}),\\ v^{n+1}_{i,j+\frac{1}{2}} &= \tilde{v}_{i,j+\frac{1}{2}}-\frac{\Delta t}{\rho^n_{i,j+\frac{1}{2}}h}(p_{i,j+1}-p_{ij}).\end{aligned}$$

*Step 4*. Once the updated fluid velocity, $\mathbf{u}^{n+1}$, has been determined, we can find the velocity, $\mathbf{U}^{n+1}$, and then the new position, $\mathbf{X}^{n+1}$, of the immersed boundary points. This is done using a discretization of Equations (2) and (3). That is, for $l=1,\dots,M$,

$$\begin{aligned}\mathbf{U}^{n+1}_l &= \sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\mathbf{u}^{n+1}_{ij}\delta^2_h(\mathbf{x}_{ij}-\mathbf{X}^n_l)h^2,\\ \mathbf{X}^{n+1}_l &= \mathbf{X}^n_l+\Delta t\mathbf{U}^{n+1}_l.\end{aligned}\tag{9}$$

This completes the description of the process (*Steps 1–4*, above) by which the quantities $\mathbf{u}^{n+1}$, $p^{n+1}$, and $\mathbf{X}^{n+1}$ are calculated.

### 2.1. Discretization of the indicator function

In this section, we will present the numerical method to calculate the indicator function. Let the discretization of the right hand side of Equation (4) be $\mathbf{G}^n_{ij}$:

$$\mathbf{G}^n_{ij}=\sum_{l=1}^{M}\mathbf{n}^n_l\delta^2_h(\mathbf{x}_{ij}-\mathbf{X}^n_l)\Delta s_{l+\frac{1}{2}}.$$

For a given interface position $\mathbf{X}_l$, the corresponding unit normal vector $\mathbf{n}_l=(m_l,n_l)$ can be calculated by using three points $\mathbf{X}_{l-1}=(x_{l-1},y_{l-1})$, $\mathbf{X}_l=(x_l,y_l)$, $\mathbf{X}_{l+1}=(x_{l+1},y_{l+1})$ with the quadratic polynomial approximation. Let the approximation be

$$x(t)=\alpha_1t^2+\beta_1t+\gamma_1\quad\text{and}\quad y(t)=\alpha_2t^2+\beta_2t+\gamma_2.$$

DOI: 10.1002/fld

And assume $\mathbf{X}_{l-1}=(x(0),y(0))$, $\mathbf{X}_l=(x(\Delta s_l),y(\Delta s_l))$, and $\mathbf{X}_{l+1}=(x(\Delta s_l+\Delta s_{l+1}),y(\Delta s_l+\Delta s_{l+1}))$, then the parameters $\alpha_1,\beta_1,\gamma_1,\alpha_2,\beta_2$, and $\gamma_2$ can be calculated by the following equations:

$$
\begin{aligned}
\begin{pmatrix}\alpha_1\\ \beta_1\\ \gamma_1\end{pmatrix} &= \begin{pmatrix}0 & 0 & 1\\ \Delta s_l^2 & \Delta s_l & 1\\ (\Delta s_l+\Delta s_{l+1})^2 & \Delta s_l+\Delta s_{l+1} & 1\end{pmatrix}^{-1}\begin{pmatrix}x(0)\\ x(\Delta s_l)\\ x(\Delta s_l+\Delta s_{l+1})\end{pmatrix},\\
\begin{pmatrix}\alpha_2\\ \beta_2\\ \gamma_2\end{pmatrix} &= \begin{pmatrix}0 & 0 & 1\\ \Delta s_l^2 & \Delta s_l & 1\\ (\Delta s_l+\Delta s_{l+1})^2 & \Delta s_l+\Delta s_{l+1} & 1\end{pmatrix}^{-1}\begin{pmatrix}y(0)\\ y(\Delta s_l)\\ y(\Delta s_l+\Delta s_{l+1})\end{pmatrix}.
\end{aligned}
$$

Now we get the unit normal vector as

$$
\mathbf{n}_l=(m_l,n_l)=\left(\frac{\frac{\mathrm{d}y(\Delta s_l)}{\mathrm{d}t}}{\sqrt{\left(\frac{\mathrm{d}x(\Delta s_l)}{\mathrm{d}t}\right)^2+\left(\frac{\mathrm{d}y(\Delta s_l)}{\mathrm{d}t}\right)^2}},\ \frac{-\frac{\mathrm{d}x(\Delta s_l)}{\mathrm{d}t}}{\sqrt{\left(\frac{\mathrm{d}x(\Delta s_l)}{\mathrm{d}t}\right)^2+\left(\frac{\mathrm{d}y(\Delta s_l)}{\mathrm{d}t}\right)^2}}\right).
$$

Then, we solve the following Poisson's equation using the multigrid method.

$$
\Delta_d I^n=\nabla_d\cdot\mathbf{G}^n.
$$

### 2.2. *The area correction algorithm*

To measure the volume enclosed by the immersed boundary points, we connect adjacent immersed boundary points with straight lines and then compute the area of the resulting polygon. For a given interface position $\mathbf{X}$, the polygonal area and the relative area error are defined as follows:

$$
A(\mathbf{X})=\frac{1}{2}\sum_{l=1}^{M}(X_lY_{l+1}-Y_lX_{l+1}),\ A_{\text{error}}(\mathbf{X})=\frac{|A(\mathbf{X}^0)-A(\mathbf{X})|}{A(\mathbf{X}^0)}.
$$

The outline of the area correction procedure is

*1) Update the interface $X_l^{n+1}$, $Y_l^{n+1}$ according to Equation (9)*

$$
(X_l^{n+1},Y_l^{n+1})=(X_l^n,Y_l^n)+\Delta t(U_l^{n+1},V_l^{n+1}).
$$

*2) Compute the polygonal area $A(\boldsymbol{X}^{n+1})$*

For a given tolerance, tol, check $A_{\text{error}}(\mathbf{X}^{n+1})<$ tol or not. If not, update the interface $\bar{X}_l^{n+1}$, $\bar{Y}_l^{n+1}$ with the following area correction algorithm by taking steps *3)* and *4)*.

$$
(\bar{X}_l^{n+1},\bar{Y}_l^{n+1})=(X_l^{n+1},Y_l^{n+1})-\epsilon(m_l,n_l).
$$

Here it should be noted that $(m_l,n_l)$ is the unit normal vector at the $l$-th interface node at time level $t^{n+1}$.

*3) Determine the parameter $\epsilon$*

$$
\epsilon=\frac{-\beta\pm\sqrt{\beta^2-4\alpha(A(\mathbf{X}^{n+1})-A(\mathbf{X}^0))}}{2\alpha}.
$$

The parameter $\epsilon$ is a root of a quadratic equation from the area correction.

$$
A(\mathbf{X}^0)=A(\mathbf{X}^{n+1})+\alpha\epsilon^2+\beta\epsilon,
$$

we get

$$
\begin{aligned}
\alpha &= \tfrac{1}{2}\sum_{l=1}^{M}(m_ln_{l+1}-n_lm_{l+1}),\\
\beta &= \tfrac{1}{2}\sum_{l=1}^{M}\left(m_lY_{l+1}^{n+1}+n_{l+1}X_l^{n+1}-(n_lX_{l+1}^{n+1}+m_{l+1}Y_l^{n+1})\right).
\end{aligned}
$$

*4) Choose parameter $\epsilon$*

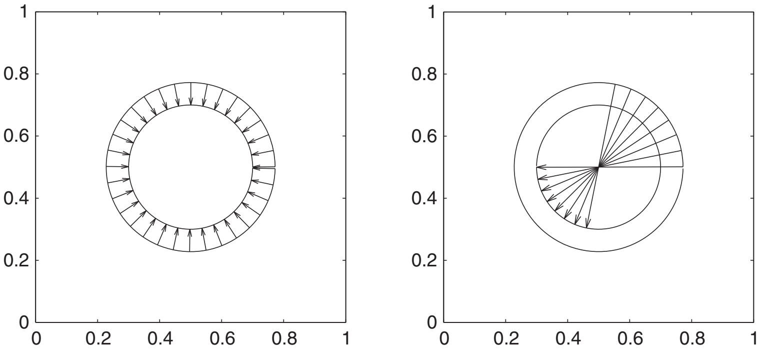
The following two case statements are directly obtainable from the above definitions. Let $\alpha > 0$ and

$$\begin{aligned}\epsilon_1 &= \frac{-\beta + \sqrt{\beta^2 - 4\alpha(A(\mathbf{X}^{n+1}) - A(\mathbf{X}^0))}}{2\alpha},\\ \epsilon_2 &= \frac{-\beta - \sqrt{\beta^2 - 4\alpha(A(\mathbf{X}^{n+1}) - A(\mathbf{X}^0))}}{2\alpha}.\end{aligned}$$

If $A(\mathbf{X}^{n+1}) > A(\mathbf{X}^0)$, then $\epsilon_2 < \epsilon_1 < 0$. Using any of $\epsilon_1$ and $\epsilon_2$, we can preserve the volume by moving the boundary position. However, in the case of $\epsilon_2$, the boundary position is moved far enough to entirely invert it (see Figure 4). To avoid this situation, we choose $\epsilon_1$. If $A(\mathbf{X}^{n+1}) < A(\mathbf{X}^0)$, then $\epsilon_2 < 0 < \epsilon_1$ and we choose $\epsilon_1$ for the same reason (see Figure 5). Therefore, we always choose $\epsilon_1$.

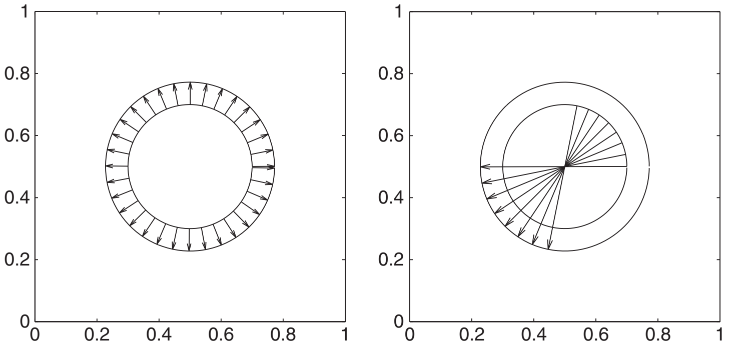
## 3. NUMERICAL EXAMPLES

It is well known that the immersed boundary method does not conserve the area enclosed by the immersed boundary although the velocity field on the Eulerian grid satisfies a discrete divergence free condition [15, 19, 24]. However, we overcome the area loss problem with the area correction algorithm that was described in Section 2.2. Our proposed scheme can be used in conjunction with



(a) (b)

Figure 4. $\epsilon_2 < \epsilon_1 < 0$. Area correction with (a) $\epsilon_1$ and with (b) $\epsilon_2$.



(a) (b)

Figure 5. $\epsilon_2 < 0 < \epsilon_1$. Area correction with (a) $\epsilon_1$ and with (b) $\epsilon_2$.

the techniques of other methods [16–19]. In this section, we perform a number of numerical experiments to investigate the effect of our area correction algorithm for the immersed boundary problem. For all tests, we set the area correction tolerance to be tol $= 1e-4$.

### 3.1. *Convergence test*

To obtain an estimate of the convergence rate, we perform a number of simulations for a problem on a set of increasingly finer grids. We consider the passive advection of a disk by the background velocity field, such as

$$\mathbf{u} = (u, v) = (16\pi(y-0.5), -16\pi(x-0.5)). \tag{10}$$

The disk with a radius of 0.1 is centered at (0.5, 0.8) in the computational domain $\Omega = (0,1)\times(0,1)$. The numerical solutions are computed on the uniform grids, $h = 1/2^n$, and with corresponding time steps, $\Delta t = h/16$, for $n = 6$, 7, 8, 9, and 10. The calculations are run up to time $T = 1/32$. $\rho = 1$, $\mu = 0.01$, and $\sigma = 0$ are used. In this test, we let $A(n)$ be an initial polygonal area as shown in Figure 6(a). The error $E(n)$ is defined as the area differences between the calculated immersed boundary and the rotated disk with an area $A(n)$ at $T = 1/32$ (see Figure 6(b)). The rate of convergence is defined as: $\log_2(E(n)/E(n+1))$. The errors and rates of convergence are given in Table I. The results suggest that the scheme is indeed first order accurate.

### 3.2. *Relaxation to a disk*

The perturbed surface of the droplet is given in polar coordinates $(r, \theta)$ and the initial droplet boundary on the computational domain $\Omega = (0,1)\times(0,1)$ is

$$\mathbf{x} = (x, y) = (0.5 + r\cos(\theta), 0.5 + r\sin(\theta)), \qquad 0 \leqslant \theta < 2\pi,$$

where $r = 0.25 + 0.1\cos(n\theta)$ and $n$ is the oscillation mode. It is well known that the boundary will relax to a disk with the area unchanged. The computations are carried out for three different modes of $n = 3$, 5, and 8 (see the first row of Figure 7). We take $h = 1/128$, $\Delta t = h/64$, $\rho = 1$, $\mu = 0.01$, and $\sigma = 130$ as the parameters. The results with area correction are shown in the second, third, and fourth rows of Figure 7. As we expected, all three modes of the interfaces relaxed to a circle. We

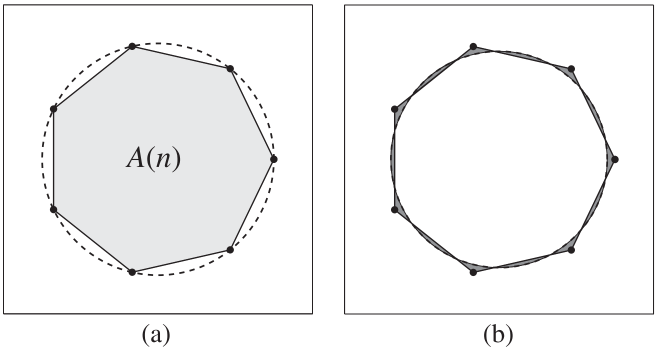


Figure 6. (a) Initial polygon and (b) schematic illustration of error $E(n)$ (an area of shaded regions).

Table I. Convergence result.

| Mesh size | Error | Rate |
|---|---|---|
| 64 × 64 | $4.7115e{-3}$ | |
| 128 × 128 | $2.3349e{-3}$ | 1.0128 |
| 256 × 256 | $1.1620e{-3}$ | 1.0068 |
| 512 × 512 | $5.7853e{-4}$ | 1.0061 |
| 1024 × 1024 | $2.8915e{-4}$ | 1.0006 |

DOI: 10.1002/fld

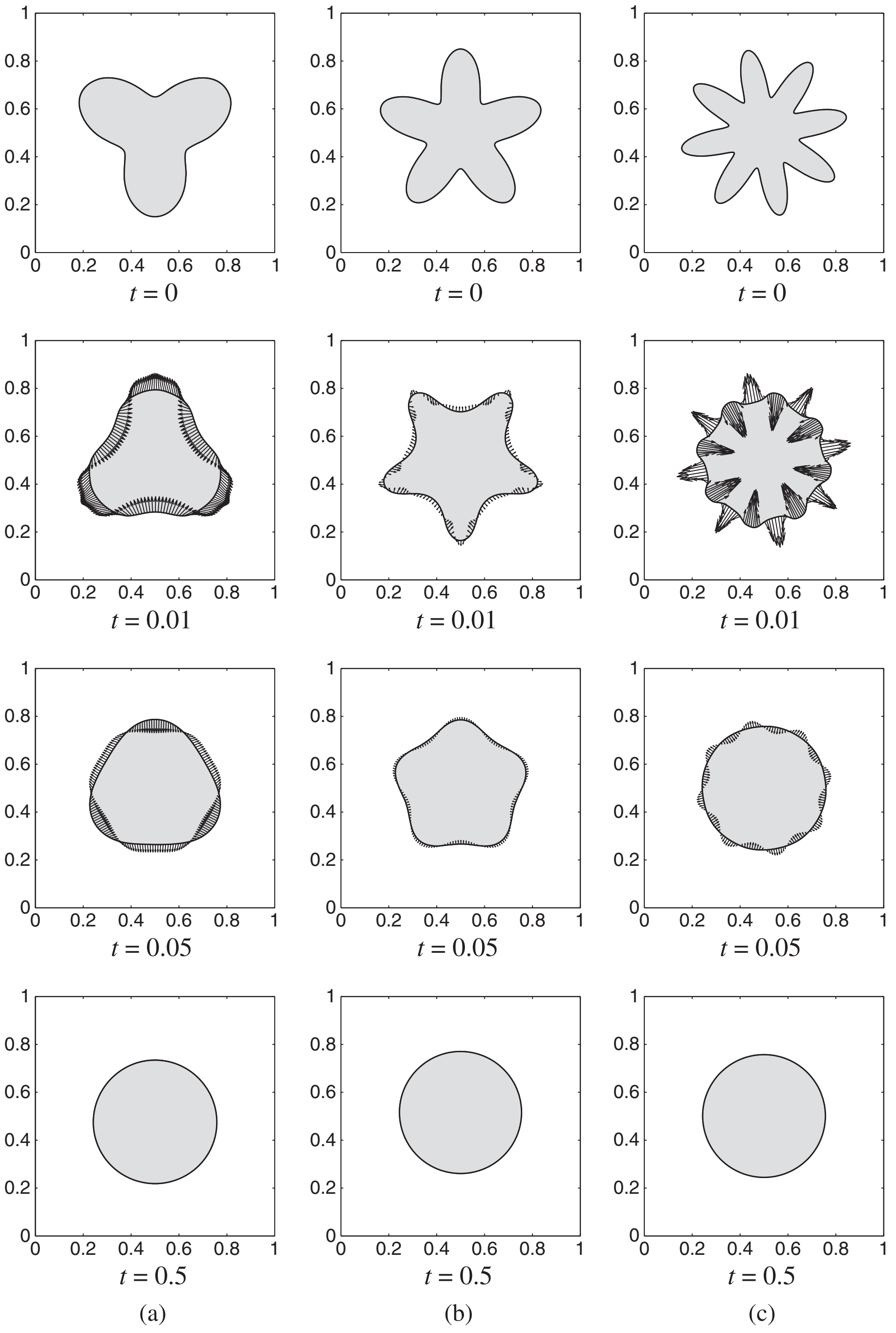


Figure 7. The evolution of interface with area correction for three different modes (a) $n = 3$, (b) $n = 5$, and (c) $n = 8$. The times are shown below each figure.

also see that the larger the mode is, the more quickly the interface relaxes to a circle because of the effect of the higher curvature of the interface.

Next, we compute the computational cost with different tolerance values when the area correction is applied. The problem is the same as above. When 4096 iterations have been executed, CPU times and numbers of the taken area correction algorithms are calculated. The results are in Table II and they indicate the following: (1) there is nearly no difference between CPU times with and without correction, and (2) varying the tolerance value does not impact the computational cost significantly. This shows that our proposed scheme requires only a slight additional computational cost to conserve the polygonal area and this is an improvement over previous attempts [16–19] that demanded more expensive computational costs.

### *3.3. Pressure difference of drop*

In this section, we theoretically and numerically calculate the pressure difference $[p]$ with different mesh sizes. In the absence of viscous, gravitational, or other external forces, surface tension causes a static drop to become spherical. Laplace's formula for an infinite cylinder surrounded by a background fluid at zero pressure, Equation (1), gives the internal drop pressure

$$p_{\text{drop}} = \frac{\sigma}{r},$$

where $r$ is the drop radius.

In this experiment, the droplet is placed at the center of the unit domain, and has a radius of 0.1. The pressure differences are computed on the uniform grids, $h = 1/2^n$, and with corresponding time steps, $\Delta t = h/64$, for $n = 5$, 6, 7, and 8. The calculations are run up to time $T = h/64$, and $\rho = 1$ and $\sigma = 20$ are employed. In this test, the exact pressure difference $[p]$ is 200. As shown in Table III, this sequence of events for the numerical pressure difference $[p]$ is qualitatively in agreement with the theoretical values by refining the mesh.

### *3.4. Area loss by spurious velocity fields induced by surface tension*

The circular drop is surrounded by a small amplitude velocity field because of the slight imbalance between the stresses at the sites in the interfacial region. Such unphysical flow is a spurious velocity. There are many studies for spurious velocities in incompressible flow problems [25–29]. In order to estimate the effect of a spurious velocity, we consider the circular drop with zero initial velocity field. The initial drop has a radius of 0.3 and is centered at the center of the unit domain. The other parameters we choose are $h = 1/128$, $\Delta t = h/64$, $\rho = 1$, $\mu = 0.01$, $\sigma = 130$, and $T = 3$. Figures 8(a), (b), and (c) show evolutions of the circular drop without area correction. The computational times are shown below each figure. Spurious velocities near the interface lead to a false appearance (the arrows represent amplitudes and directions of these velocities presented in Figure 8). As a result, we observe that the area is not conserved as predicted and the area loss is as large as 18.34% when $T = 3$. However, an exact mass conservation is achieved using the correction scheme. Figure 8(d) illustrates the area loss rates with and without correction.

Table II. CPU times with different tolerance values during 4096 iterations. The number in parentheses indicates the number of the taken area correction algorithms.

| | | With correction | | |
|---|---|---|---|---|
| Case | Without correction | tol $= 1e-3$ | tol $= 1e-4$ | tol $= 1e-5$ |
| $n = 3$ | 125.32$s$ | 124.90$s$ (52) | 124.87$s$ (483) | 125.48$s$ (4001) |
| $n = 5$ | 112.18$s$ | 112.59$s$ (47) | 112.54$s$ (432) | 113.12$s$ (3422) |
| $n = 8$ | 105.35$s$ | 105.28$s$ (38) | 105.40$s$ (357) | 106.17$s$ (4077) |

Table III. The pressure difference $[p]$ with $\sigma = 20$ and $r = 0.1$ for different mesh sizes.

| | Mesh sizes | | | |
|---|---|---|---|---|
| | 32 × 32 | 64 × 64 | 128 × 128 | 256 × 256 |
| $[p]$ | 201.147 | 200.673 | 200.379 | 200.312 |

DOI: 10.1002/fld

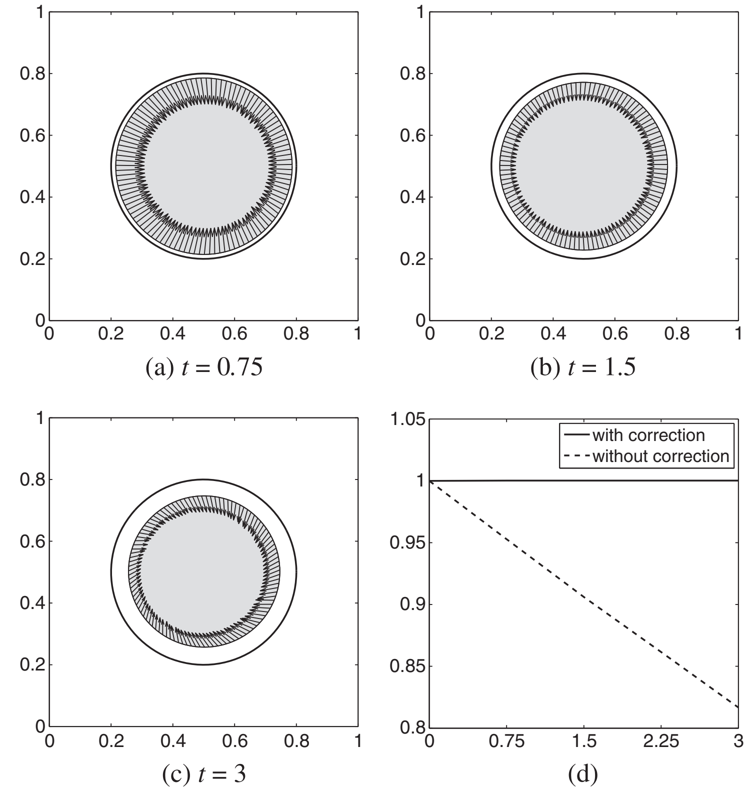


Figure 8. (a), (b), and (c) show evolutions of the circular drop by surface tension without area correction. (d) illustrates the area loss rates with and without correction.

### *3.5. Area change by the background fluid flows*

Even if a velocity field remains discretely divergence free on the Lagrangian grid, there is still a volume loss because we use a discrete time step. To confirm this, we consider the passive advection of a disk by the background velocity field such as in Equation (10). This velocity field is divergence free. The disk with a radius of 0.1 is centered at the center of the unit domain. One thousand immersed boundary points are on the circle. For each time step, the interface is advected for only one time step. Figure 9 shows the area errors versus time step size. From these results, when we discretely evolve Lagrangian particles, we see that the advected interface does not conserve the volume even though the velocity field is divergence free. However, using our proposed scheme, we completely eliminate the area errors when we are using a discrete time step. To show this, we consider the passive advection of a disk by the background velocity field, such as in Equation (10). The disk with a radius of 0.25 is placed at the center of the unit domain. We use the following simulation parameters: $h = 1/128$, $\rho = 1$, $\mu = 0.01$, $\sigma = 0$, $\Delta t = h/64$, and $T = 2$. Figures 10(a), (b), and (c) show evolutions of the disk without area correction. As can be seen, the area does not conserve as predicted. The larger the time step is, the more severe the area change becomes. We will discuss this in Section 3.6. Figure 10(d) illustrates the area gain rates with and without correction. If we solve for the velocity field without correction, then the area gains as much as 36.13%. However, the correction step completely eliminates the area errors.

### *3.6. Rotated disk by background fluid flows*

We consider the passive advection of a disk by the background velocity field such as in Equation (10). The disk with a radius of 0.1 is centered at $(0.5, 0.8)$ in the unit domain. We take $h = 1/128$, $\rho = 1$, $\mu = 0.01$, $\sigma = 0$, and $T = 1/8$ with different time steps $\Delta t = h/128$ and $\Delta t = h/16$. For

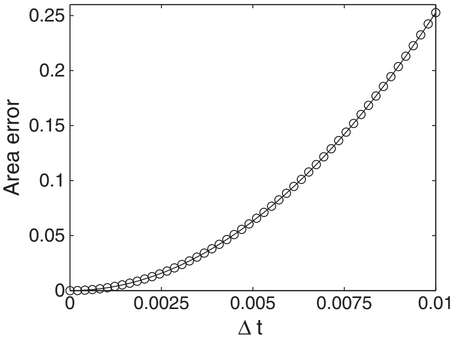
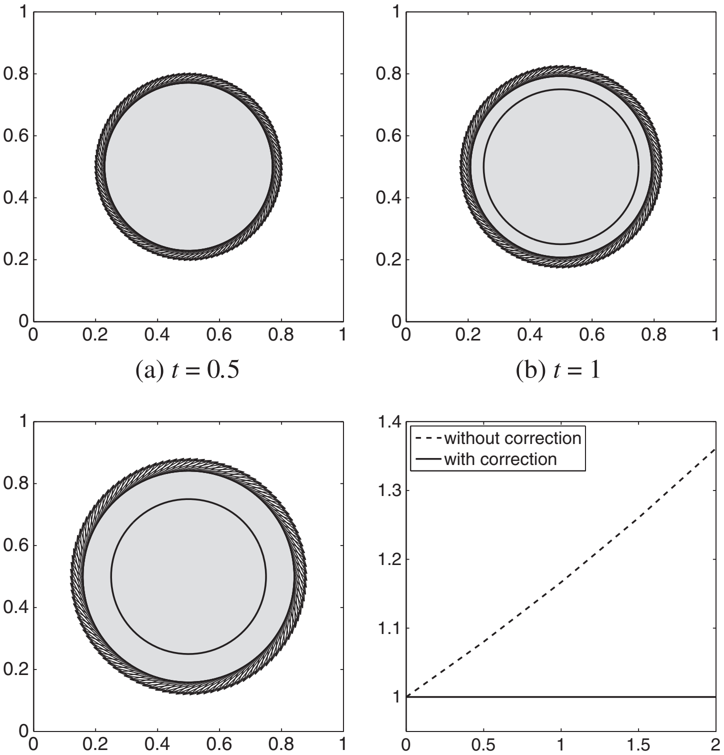


Figure 9. The area errors versus time-step size without correction.



(a) $t = 0.5$

(b) $t = 1$

(c) $t = 2$

(d)

Figure 10. (a), (b), and (c) show evolutions of the disk by rotation without area correction. (d) illustrates the area change rates with and without correction.

each time step, evolutions are shown in Figure 11. As can be seen from the area error in Table IV, the area is more conservative when the time step is smaller. CPU times with and without correction are shown in Table V for each time step.

### 3.7. *Deformation of a drop under shear flow*

Next, we investigate the deformation of a drop under shear flow. A simulation of this problem has previously been performed by Chinyoka *et al*. [30]. The initial drop is circular with a radius of $0.5H$ centered at $(H, H)$ in the computational domain $\Omega = (0, 2H) \times (0, 2H)$. The top moves to the right with velocity $U$ and the bottom moves to the left with velocity $-U$. Figure 12 (a) shows a schematic illustration of the initial condition. To compare the result of Chinyoka *et al*. [30], we begin with a

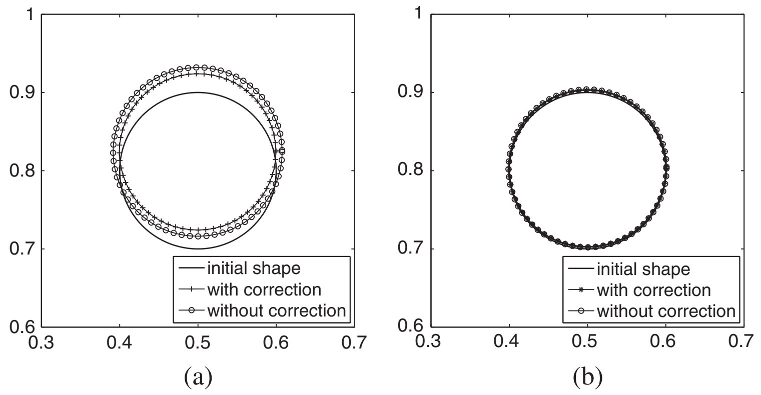


Figure 11. A rotated disk with a time step (a) $\Delta t = h/16$ and (b) $\Delta t = h/128$.

Table IV. Area error without correction.

| | Time step | |
|---|---|---|
| | $\Delta t = h/128$ | $\Delta t = h/16$ |
| $A_{\text{error}}(\mathbf{X})$ | 1.94% | 16.61% |

Table V. CPU times for each time step with and without correction. In the number $A/B$ in parentheses, $A$ indicates the number of the taken area correction algorithms and $B$ indicates the number of total iterations.

| Case | Without correction | With correction |
|---|---|---|
| $\Delta t = h/16$ | 1.266$s$ | 1.265$s$ (256/256) |
| $\Delta t = h/128$ | 5.687$s$ | 5.688$s$ (186/2048) |

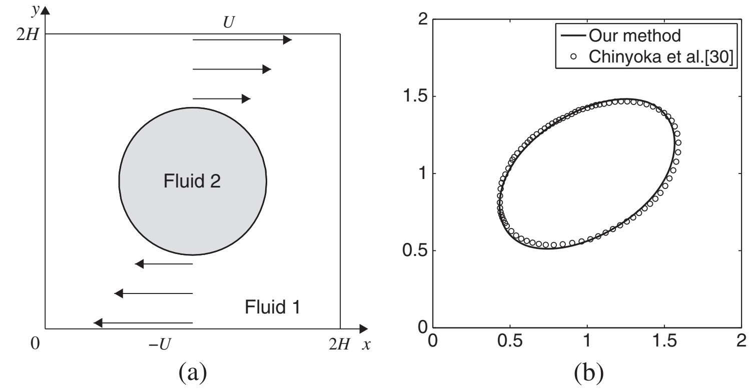


Figure 12. (a) Schematic illustration of the initial condition. (b) The deformation of a drop under shear flow: the solid line and the open circle represent the results using our proposed method and that of Chinyoka *et al.* [30], respectively.

drop of radius 0.5 positioned at the center of the computational domain $\Omega = (0,2) \times (0,2)$. In this simulation we take the following parameters: $\mu = 1$, $\sigma = 0.8$, and $U = 1$. A mesh size of $128 \times 128$ and a time step of $\Delta t = h^2/16$ are used. The calculation is run up to time $T = 1.5$. Figure 12(b) shows that our result and the previous result of Chinyoka *et al.* [30] are in good qualitative agreement.

DOI: 10.1002/fld

### *3.8. Buoyancy-driven flow*

In this section, we consider the buoyancy-driven flow. The initial bubble with radius $r = 0.1$ is positioned at $(0.5, 0.5)$ in the computational domain $\Omega = (0,1) \times (0,2)$ with $128 \times 256$ mesh grids. The densities are $\rho_1 = 1000$ and $\rho_2 = 500$ ($\rho_1$ and $\rho_2$ are the densities outside and inside the bubble, respectively). Other parameters are defined as follows: $\mu = 0.01, \sigma = 0.1, g = 9.81e{-}3, h = 1/128$, and $\Delta t = 0.01$. To solve the buoyancy-driven flow, we use a periodic boundary condition to vertical boundaries and no-slip boundary condition to the top and bottom domains. Therefore,

$$\mathbf{n} \cdot \nabla_d p^{n+1} = \mathbf{n} \cdot \rho^n \mathbf{g}, \quad \text{i.e.,} \quad p_y = -\rho^n g \quad \text{at } y = 0 \text{ and } y = 2.$$

Figure 13 shows that the bubble starts to rise because of the effect of buoyancy in the cylinder and it eventually deforms to a steady-state shape. We obtained similar results as the results in [31].

### *3.9. The effect of the correction algorithm on capillary forces*

Because the new interface position is updated by $\bar{\mathbf{X}}^{n+1} = \mathbf{X}^{n+1} - \epsilon \mathbf{n}$, we have $\bar{\mathbf{X}}_{ss}^{n+1} = \mathbf{X}_{ss}^{n+1} + \epsilon(\kappa_s \mathbf{X}_s^{n+1} + \kappa \mathbf{X}_{ss}^{n+1})$ by using the identity $\mathbf{n}_s = -\kappa \mathbf{X}_s$. Here, $\kappa$ is the curvature of the interface. Therefore, it is possible that if either the curvature or its derivative is large then the original force $\mathbf{f}(s,t) = \sigma \mathbf{X}_{ss}$ may change significantly because of this area correction. However, the results from

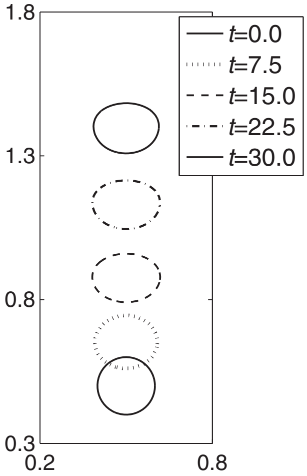
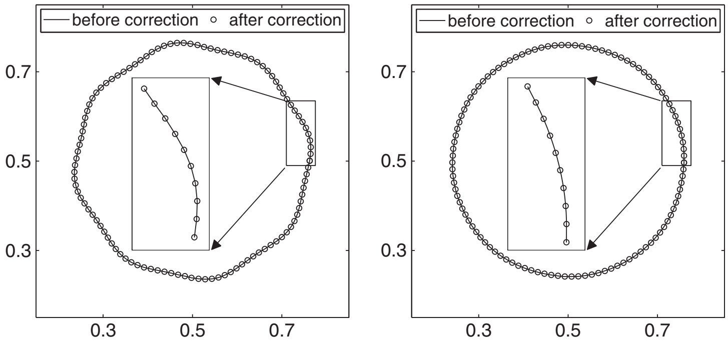


Figure 13. Instantaneous bubble shapes at different time.



(a) $t = 0.04$ (b) $t = 0.1$

Figure 14. Comparison of results by two types of interface force: $\mathbf{f}_1 = \sigma \mathbf{X}_{ss}^{n+1}$ (solid line) and $\mathbf{f}_2 = \sigma \bar{\mathbf{X}}_{ss}^{n+1}$ (circle).

the comparison with and without area correction algorithm show that there is no significant difference. This is partly because when $\sigma$ is large, the surface curvature is not large. To confirm this, we set the initial shape as shown in Section 3.2 with $n = 8$. The simulation parameters are $h = 1/128$, $\Delta t = 2h^2$, $\rho = 1$, $\mu = 0.005$, $\sigma = 300$, and $T = 0.1$. In Figure 14, the line and circle are the numerical solutions with $\mathbf{f}_1 = \sigma \mathbf{X}_{ss}^{n+1}$ and $\mathbf{f}_2 = \sigma \bar{\mathbf{X}}_{ss}^{n+1}$, respectively. Which means that we define the interfacial force before and after area correction. As can be seen in Figure 14, results from the two types of interface force are almost identical.

## 4. CONCLUSIONS

We proposed a simple area preserving correction scheme for two-phase immiscible incompressible flows with an IBM. The idea is to correct the interface location normal to the interface so that the area remains constant. Various numerical tests were presented to illustrate the efficiency and accuracy of our proposed scheme for two-phase fluid flows.

### ACKNOWLEDGEMENTS

E. Jung was supported by a Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2009-0077511). J. S. Kim was supported by a Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2010-0003989). The authors thank the reviewers for their constructive and helpful reviews, comments, and suggestions on the revision of this article.

### REFERENCES

1. Galaktionov OS, Anderson PD, Peters GWM, Van de Vosse FN. An adaptive front tracking technique for three-dimensional transient flows. *International Journal for Numerical Methods in Fluids* 2000; **32**:201–217.
2. Shin S, Juric D. A hybrid interface method for three-dimensional multiphase flows based on front tracking and level set techniques. *International Journal for Numerical Methods in Fluids* 2009; **60**:753–778.
3. Rutka V, Li Z. An explicit jump immersed interface method for two-phase Navier–Stokes equations with interfaces. *Computer Methods in Applied Mechanics and Engineering* 2008; **197**:2317–2328.
4. Tan Z, Lim KM, Khoo BC. A level set-based immersed interface method for solving incompressible viscous flows with the prescribed velocity at the boundary. *International Journal for Numerical Methods in Fluids* 2010; **62**:267–290.
5. Peskin C. Numerical analysis of blood flow in the heart. *Journal of Computational Physics* 1977; **25**:220–252.
6. Deng J, Shao X-M, Ren A-L. A new modification of the immersed-boundary method for simulating flows with complex moving boundaries. *International Journal for Numerical Methods in Fluids* 2006; **52**:1195–1213.
7. Sheu TWH, Ting HF, Lin RK. An immersed boundary method for the incompressible Navier–Stokes equations in complex geometry. *International Journal for Numerical Methods in Fluids* 2008; **56**:877–898.
8. Chang Y, Hou T, Merriman B, Osher S. A level set formulation of Eulerian interface capturing methods for incompressible fluid flows. *Journal of Computational Physics* 1996; **124**:449–464.
9. Grooss J, Hesthaven J. A level set discontinuous Galerkin method for free surface flows. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:3406–3429.
10. Kim J. A continuous surface tension force formulation for diffuse-interface models. *Journal of Computational Physics* 2005; **204**:784–804.
11. Kim J. A generalized continuous surface tension force formulation for phase-field models for immiscible multi-component fluid flows. *Computer Methods in Applied Mechanics and Engineering* 2009; **198**:3105–3112.
12. Kim Y, Peskin C. Numerical study of incompressible fluid dynamics with nonuniform density by the immersed boundary method. *Physics of Fluids* 2008; **20**:062101-1–062101-10.
13. Francois M, Shyy W. Computations of drop dynamics with the immersed boundary method, Part 1: Numerical algorithm and buoyancy-induced effect. *Numerical Heat Transfer B* 2003; **44**:101–118.
14. Francois M, Uzgoren E, Jackson J, Shyy W. Multigrid computations with the immersed boundary technique for multiphase flows. *International Journal of Numerical Methods for Heat and Fluid Flow* 2004; **14**:98–115.
15. Newren E, Fogelson A, Guy R, Kirby R. Unconditionally stable discretizations of the immersed boundary equations. *Journal of Computational Physics* 2007; **222**:702–719.
16. Cortez R, Minion M. The blob projection method for immersed boundary problems. *Journal of Computational Physics* 2000; **161**:428–453.
17. Leveque R, Li Z. Immersed interface methods for Stokes flow with elastic boundaries or surface tension. *SIAM Journal on Scientific Computing* 1997; **18**:709–735.

DOI: 10.1002/fld

18. Peskin C. The immersed boundary method. *Acta Numerica* 2002; **11**:1–39.
19. Peskin C, Printz B. Improved volume conservation in the computation of flows with immersed elastic boundaries. *Journal of Computational Physics* 1993; **105**:33–46.
20. Newren EP. Enhancing the immersed boundary method: stability, volume conservation, and implicit solvers. *PhD thesis*, University of Utah, 2007.
21. Harlow F, Welch J. Numerical calculations of time dependent viscous incompressible flow with free surface. *Physics of Fluids* 1965; **8**:2182–2189.
22. Peskin C, Mcqueen D. A general method for the computer simulation of biological systems interacting with fluids. *Symposia of the Society for Experimental Biology* 1995; **49**:265–276.
23. Trottenberg U, Oosterlee C, Schüller A. *Multigrid*. Academic Press: London, 2001.
24. Lai M-C, Tseng Y-H, Huang H. An immersed boundary method for interfacial flows with insoluble surfactant. *Journal of Computational Physics* 2008; **227**:7279–7293.
25. Ganesan S, Matthies G, Tobiska L. On spurious velocities in incompressible flow problems with interfaces. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**:1193–1202.
26. Gerbeau J-F, Le Bris C, Bercovier M. Spurious velocities in the steady flow of an incompressible fluid subjected to external forces. *International Journal for Numerical Methods in Fluids* 1997; **25**:679–695.
27. Gresho P, Lee R, Chan S, Leone J. A new finite element for incompressible or Boussinesq fluids. *Proceedings of the Third International Conference on Finite Elements in Flow Problems*, Banff, Canada, 1980.
28. Lafaurie B, Nardone C, Scardovelli R, Zaleski S, Zanetti G. Modelling merging and fragmentation in multiphase flows with SURFER. *Journal of Computational Physics* 1994; **113**:134–147.
29. Pelletier D, Fortin A, Camarero R, Are FEM. solutions of incompressible flows really incompressible (Or how simple flows can cause headaches). *International Journal for Numerical Methods in Fluids* 1989; **9**:99–112.
30. Chinyoka T, Renardy YY, Renardy M, Khismatullin DB. Two-dimensional study of drop deformation under simple shear for Oldroyd-B liquids. *Journal of Non-Newtonian Fluid Mechanics* 2005; **130**:45–56.
31. Raessi M, Bussmann M, Mostaghimi J. A semi-implicit finite volume implementation of the CSF method for treating surface tension in interfacial flows. *International Journal for Numerical Methods in Fluids* 2009; **59**:1093–1110.

DOI: 10.1002/fld