



# An interface capturing method with a continuous function: The THINC method with multi-dimensional reconstruction

Satoshi Ii <sup>a,\*</sup>, Kazuyasu Sugiyama <sup>a</sup>, Shintaro Takeuchi <sup>b</sup>, Shu Takagi <sup>a,c</sup>, Yoichiro Matsumoto <sup>a</sup>, Feng Xiao <sup>d</sup>

<sup>a</sup> Department of Mechanical Engineering, The University of Tokyo, 7-3-1 Hongo Bunkyo-ku, Tokyo 113-8656, Japan

<sup>b</sup> Department of Mechanical Engineering, Osaka University, 2-1 Yamada-oka Suita, Osaka 565-0871, Japan

<sup>c</sup> Computational Science Research Program, RIKEN, 2-1 Hirosawa Wako, Saitama 351-0198, Japan

<sup>d</sup> Department of Energy Sciences, Tokyo Institute of Technology, 4259 Nagatsuta Midori-ku, Yokohama 226-8502, Japan

## ARTICLE INFO

### Article history:

Received 26 May 2011

Received in revised form 16 November 2011

Accepted 24 November 2011

Available online 9 December 2011

### Keywords:

Interface capturing method

Volume-of-fluid (VOF)

Continuous function

Fixed Cartesian mesh

Incompressible immiscible fluids

## ABSTRACT

An interface capturing method with a continuous function is proposed within the framework of the volume-of-fluid (VOF) method. Being different from the traditional VOF methods that require a geometrical reconstruction and identify the interface by a discontinuous Heaviside function, the present method makes use of the hyperbolic tangent function (known as one of the sigmoid type functions) in the tangent of hyperbola interface capturing (THINC) method [F. Xiao, Y. Honma, K. Kono, A simple algebraic interface capturing scheme using hyperbolic tangent function, *Int. J. Numer. Methods Fluids* 48 (2005) 1023–1040] to retrieve the interface in an algebraic way from the volume-fraction data of multi-component materials. Instead of the 1D reconstruction in the original THINC method, a multi-dimensional hyperbolic tangent function is employed in the present new approach. The present scheme resolves moving interface with geometric faithfulness and compact thickness, and has at least the following advantages: (1) the geometric reconstruction is not required in constructing piecewise approximate functions; (2) besides a piecewise linear interface, curved (quadratic) surface can be easily constructed as well; and (3) the continuous multi-dimensional hyperbolic tangent function allows the direct calculations of derivatives and normal vectors. Numerical benchmark tests including transport of moving interface and incompressible interfacial flows are presented to validate the numerical accuracy for interface capturing and to show the capability for practical problems such as a stationary circular droplet, a drop oscillation, a shear-induced drop deformation and a rising bubble.

© 2011 Elsevier Inc. All rights reserved.

## 1. Introduction

Multi-phase flow analysis is widely required in the fields of science, engineering and medical applications. Many numerical approaches for the multi-component materials have been so far proposed in Lagrangian and/or Eulerian frame.

Among the Lagrangian type methods, the arbitrary Lagrangian Eulerian (ALE) approach [16,5,19,67] has gained a wide popularity because of the accurate treatment of the interface by using a body-fitted grid where the kinematic and dynamic boundary conditions on the interface can be explicitly specified or computed. Further advanced algorithms, such as

\* Corresponding author.

E-mail address: [sii@fel.t.u-tokyo.ac.jp](mailto:sii@fel.t.u-tokyo.ac.jp) (S. Ii).

deforming-spatial-domain/space-time (DSD/ST) method, were presented in [51,20] for the flows involving moving and deforming interfaces, where the physical fields are discretized at the space-time nodes to obtain stable and accurate solutions. However, all the methods using the body-fitted grid need a re-meshing process which takes extra time to construct a high quality mesh and is formidably difficult when the interfaces are largely deformed or topologically changed.

The above difficulty can be avoided in fixed-grid approaches in which the moving interface and flow fields are solved on a fixed Eulerian mesh, and thus re-meshing is not needed. However, another issues arising are how to represent and compute the moving interface and the boundary conditions upon it on a fixed mesh. The immersed boundary method [32,33] is one of the most successful methods for these types of problems. Under the framework of the immersed boundary method, the fluid equations are solved in an Eulerian frame, while the interface is tracked in a Lagrangian manner with a set of marker points. The force exerted by the interface on the Eulerian flow field is interpolated with the smoothed (or approximate) delta function. Many improvements and extensions have been proposed over the past decades. For example, a new version of the method was proposed in [23] that achieves a second-order accuracy for representing smooth solutions. The front tracking method [54,12,35,52,53] can be applied to multi-phase flow problems including the surface tension effect with different fluid properties. Moreover, the immersed interface method [24,26] provides a recipe for developing schemes for problems with piecewise smooth solutions, by introducing the modified Taylor expansion with the interfacial jump conditions. In these schemes, the interface can be accurately represented by the Lagrangian particles, which is of particular interests when the interface has structures that are under the resolution of the fixed Eulerian mesh. On the other hand, the particle-based methods do not automatically conserve the volume or mass enclosed by the surface reconstructed from the marker particles, and the numerical treatment of the topological change, such as coalescence or breakup, requires additional physical model [28,29,64].

For representing the interfaces, field variables or indicator functions defined on the Eulerian mesh are also widely used, rather than the Lagrangian particles, in the area of practical simulations involving resolved interfaces. Among the representative field functions used to compute the moving interface are the volume-of-fluid (VOF) function [17], level-set function (signed distance function) [31,45,47], density or color function [59,60,57] and the phase-field function (order parameter) [21,4,8]. For this Eulerian representation of the interface, large efforts have been paid for constructing/maintaining sharp interface. Using field function facilitates treatment of the coalescence or breakup of the free surface. More importantly, numerical conservation can be exactly guaranteed if the field function is updated by a finite volume scheme of flux form, which is naturally implemented in the VOF method.

In the present paper, we focus on the VOF approach where the volume fraction of a specific material (generally called VOF function of the material) is updated by solving the advection equation with a finite volume formulation on a fixed Eulerian mesh. In a conventional VOF method, the interface separating different fluids is piecewisely reconstructed for each mesh (or cell) by straight line segment before calculating the numerical fluxes to update the VOF function. This geometric reconstruction effectively eliminates the numerical diffusion that smears out the compactness of the transition layer of the interface. The accurate recovery to the exact surface geometry from the discretized VOF function, in other words the geometric faithfulness, is closely related to the accuracy of the method. Compared to the reconstruction by the simple line interface calculation (SLIC) method [27], the piecewise linear interface calculation (PLIC) method originally proposed by Youngs [65,66] gives numerical solutions of much superior quality. The successive progress of the PLIC type method can be found in Puckett et al. [36], Rider and Kothe [39], Harvie and Fletcher [15], Aulisa et al. [1] and Pilliod and Puckett [34]. In these PLIC algorithms, an interface is recovered from the volume fraction values by assembling the surface segments encompassing the cut volume, which usually involves the descriptions of planes cutting through the mesh cells. This kind of interface reconstruction is referred to as the geometrical reconstruction in our context.

Especially, Scardovelli and Zaleski [42] derived an elegant formulation to analytically determine the cutting interface from the volume fraction and vice versa, with the given normal direction. Their method substantially reduces the explicit geometrical computation in the traditional PLIC methods, and has been applied to a problem of transporting moving interface in [2].

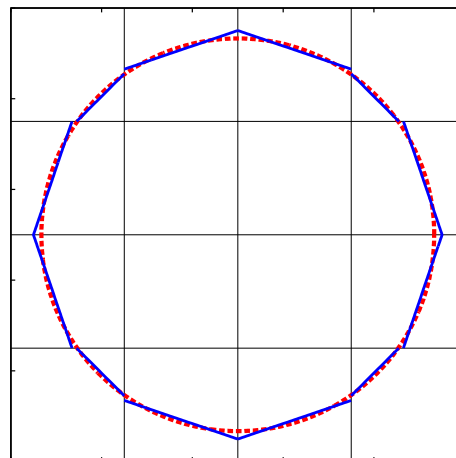
Further developments followed by taking the quadratic interface into account to improve the numerical accuracy in calculating the geometric properties such as the normal vector and the curvature [38,43] as well as in representing the curved interface so that the interface satisfies the  $C_1$  continuity across the computational mesh [9]. As a different philosophy to increase the accuracy of the curvature estimation, Sussman and Puckett [46] proposed the coupled level set and volume-of-fluid (CLSVOF) method, in which not only the VOF function but also the level-set function are employed to improve the geometric faithfulness of the computed interface.

One of the present authors has proposed another type of VOF method, the tangent of hyperbola for interface capturing (THINC) method [56], to avoid the explicit geometric reconstruction in the conventional VOF methods. In the THINC method, a continuous sigmoid function (1D hyperbolic tangent function) is employed instead of the Heaviside function which allows to represent the interface completely in an algebraic form, thus it enables computation of numerical flux without the geometric information. Based on the continuous sigmoid function, a transition layer across the interface is readily to be controlled as a step-like distribution, and thus can effectively remove the numerical diffusion. The original THINC method [56] employs the 1D reconstruction function and uses directional splitting for multi-dimensional cases. An improvement was proposed by combining the original THINC method and the first-order upwind method. The developed scheme is so-called THINC/WLIC (THINC/weighted linear interface capturing) method [61], in which the weighted average of the numerical fluxes (obtained from the first-order upwind method and the original THINC method) is determined by the respective

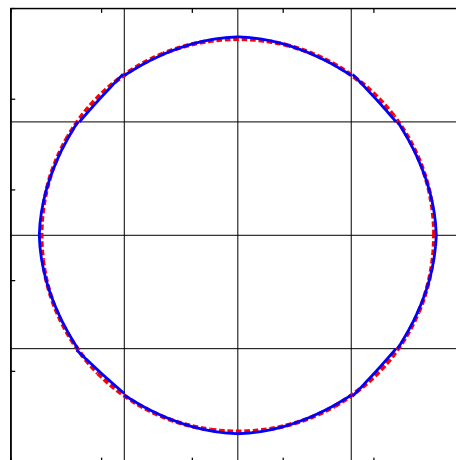
components of the unit normal vector. The THINC/WLIC scheme significantly improves the geometric faithfulness of the numerical solution and has been proved to be accurate and practical for real-case applications [62,63]. A more recent improvement has been devised from the observation that the thickness of the jump transition layer in the 1D THINC scheme can be effectively scaled to a desired width according on the orientation of the interface [58]. However, all these schemes use the 1D THINC scheme as the building block.

In the present paper, we further develop the THINC formulation and propose a new scheme, namely the multi-dimensional THINC (MTHINC) method by using the fully multi-dimensional hyperbolic tangent function for interface reconstruction. Given the orientation of an interface, a multi-dimensional hyperbolic tangent function with a smooth transition layer representing the plane surface cutting out a boundary cell (Fig. 1(a)) can be uniquely determined from the volume fraction for the corresponding cell. Furthermore, additionally provided with the local curvature, we can construct a multi-dimensional hyperbolic tangent function with the transition layer that represents a curved (quadratic) surface (Fig. 1(b)), thus a more accurate solution for the interface is obtained with a computational effort almost the same as in the plane surface case. The numerical fluxes can be directly evaluated by integrating the hyperbolic tangent function for either a plane or a curved surface. Similar to the 1D reconstruction, the multi-dimensional reconstruction also effectively prevents the numerical diffusion that smears out the interface transition layer. Another advantage of using the continuous function with a constant transition thickness is that the normal vector, curvature and approximate delta function can be directly obtained from the derivatives of the function. Smoothing or convolution, which is usually needed for conventional VOF methods, is not required.

The present paper is organized as follows. The numerical algorithm is described in detail in Section 2. The flow solver is introduced in Section 3. Numerical results to assess the present method are given in Section 4, where extensive numerical



(a) Linear reconstruction



(b) Quadratic reconstruction

**Fig. 1.** Comparison of the interface reconstruction on a mesh with  $N = 4$ . The dashed line is the exact circular interface and the solid line is the numerically-reconstructed interface.

tests for both advection equation and immiscible interfacial flows with surface tension are presented, followed by some concluding remarks in Section 5.

## 2. Numerical method

### 2.1. Basic definition

We introduce an indicator (or color) function  $H(\mathbf{x}, t)$  to identify a given fluid so that  $H(\mathbf{x}, t) = 1$  when  $\mathbf{x}$  falls in the region of the fluid or  $H(\mathbf{x}, t) = 0$  otherwise. Considering that the fluid is transported by the flow velocity, we update  $H(\mathbf{x}, t)$  in the Eulerian framework by the following advection equation:

$$\frac{DH}{Dt} \equiv \frac{\partial H}{\partial t} + \mathbf{u} \cdot \nabla H = 0, \quad (1)$$

where  $\mathbf{u}(\mathbf{x}) = (u(\mathbf{x}), v(\mathbf{x}), w(\mathbf{x}))$  is the velocity vector. The cell-averaged value of the indicator function  $H(\mathbf{x}, t)$  is defined as the volume fraction or volume-of-fluid (VOF) function within a finite control volume  $\delta V(\mathbf{x})$ ,

$$\phi(\mathbf{x}, t) \equiv \frac{1}{|\delta V(\mathbf{x})|} \int_{\delta V(\mathbf{x})} H(\mathbf{x}', t) dV'. \quad (2)$$

It is obvious that the VOF function  $\phi(\mathbf{x}, t)$  satisfies  $0 \leq \phi(\mathbf{x}, t) \leq 1$ , and the interface cuts out  $\delta V(\mathbf{x})$  when  $0 < \phi(\mathbf{x}, t) < 1$ .

From (1) and (2), we obtain the advection equation of VOF function in the divergence form:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}H) = \phi \nabla \cdot \mathbf{u}. \quad (3)$$

In a discretization context, the 3D computational domain  $(x, y, z) \in \mathbb{R}^3$  is divided into a Cartesian mesh with the mesh element (cell)  $\delta V_{ijk}$  as

$$\delta V_{ijk} \equiv [\delta x_i] \times [\delta y_j] \times [\delta z_k], \equiv \left[ x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}} \right] \times \left[ y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}} \right] \times \left[ z_{k-\frac{1}{2}}, z_{k+\frac{1}{2}} \right], \quad (i = 1, \dots, N_x, j = 1, \dots, N_y, k = 1, \dots, N_z), \quad (4)$$

where  $N_x, N_y, N_z$  are the total numbers of cells in  $x, y, z$  directions, respectively. The cell volume is

$$|\delta V_{ijk}| \equiv \Delta V_{ijk} = \Delta x_i \Delta y_j \Delta z_k,$$

where  $\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$ ,  $\Delta y_j = y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}}$  and  $\Delta z_k = z_{k+\frac{1}{2}} - z_{k-\frac{1}{2}}$ .

The VOF function on cell  $\delta V_{ijk}$  is then defined as

$$\phi_{ijk}(t) \equiv \frac{1}{\Delta x_i \Delta y_j \Delta z_k} \int_{\delta x_i} \int_{\delta y_j} \int_{\delta z_k} H(\mathbf{x}, t) dx dy dz. \quad (5)$$

### 2.2. Multi-dimensional THINC (MTHINC) reconstruction

The major task in VOF methods is to retrieve the indicator function  $H_{ijk}(\mathbf{x}, t)$  piecewisely on a cell  $\delta V_{ijk}$ . Instead of the Heaviside function in the conventional VOF methods, we make use of a hyperbolic tangent function as an approximate (indicator) function. We drop the cell indices  $ijk$  for simplicity unless it is necessary, and employ a local coordinate system  $\mathbf{X} = (X, Y, Z)$  defined by

$$(X, Y, Z) = \left( \frac{x - x_{i-\frac{1}{2}}}{\Delta x_i}, \frac{y - y_{j-\frac{1}{2}}}{\Delta y_j}, \frac{z - z_{k-\frac{1}{2}}}{\Delta z_k} \right). \quad (6)$$

Then the approximate function is given as

$$\hat{H}(\mathbf{X}) = \frac{1}{2} (1 + \tanh(\beta(P(\mathbf{X}) + d))), \quad X, Y, Z \in [0, 1], \quad (7)$$

where  $P(\mathbf{X})$  is a surface function in three dimensions, and the parameter  $\beta$  is for controlling the sharpness of the function  $\hat{H}(\mathbf{X})$ .

In this paper, practical choice of the surface function  $P(\mathbf{X})$  is of either linear (plane) or quadratic (curved surface). The linear surface function is given by

$$P(\mathbf{X}) = n_x X + n_y Y + n_z Z, \quad (8)$$

where  $n_a$  ( $a = X, Y, Z$ ) is the unit normal vector defined at the cell center,  $(X_c, Y_c, Z_c) = (0.5, 0.5, 0.5)$  in the local coordinate system. On the other hand, the quadratic surface function is specified according to not only the unit normal vector  $n_a$  ( $a = X, Y, Z$ ) but also the symmetric Cartesian curvature tensor. We represent the quadratic surface function  $P(\mathbf{X})$  as

$$P(\mathbf{X}) = c^{(X)} a_{200} X^2 + c^{(Y)} a_{020} Y^2 + c^{(Z)} a_{002} Z^2 + c^{(X)} c^{(Y)} a_{110} XY + c^{(Y)} c^{(Z)} a_{011} YZ + c^{(X)} c^{(Z)} a_{101} XZ + a_{100} X + a_{010} Y + a_{001} Z, \quad (9)$$

where  $c^{(\alpha)}$  ( $\alpha = X, Y, Z$ ) are the flag coefficients taking either 0 or 1 (described later), and  $a_{str}$  ( $s, t, r = 0, 1, 2, s + t + r \leq 2$ ) the coefficients uniquely determined from the following constraints:

$$\begin{cases} \frac{\partial P}{\partial X}(\mathbf{X}_c) = n_X, & \frac{\partial P}{\partial Y}(\mathbf{X}_c) = n_Y, & \frac{\partial P}{\partial Z}(\mathbf{X}_c) = n_Z, \\ \frac{\partial^2 P}{\partial X^2}(\mathbf{X}_c) = l_{XX}, & \frac{\partial^2 P}{\partial Y^2}(\mathbf{X}_c) = l_{YY}, & \frac{\partial^2 P}{\partial Z^2}(\mathbf{X}_c) = l_{ZZ}, \\ \frac{\partial^2 P}{\partial X \partial Y}(\mathbf{X}_c) = l_{XY}, & \frac{\partial^2 P}{\partial Y \partial Z}(\mathbf{X}_c) = l_{YZ}, & \frac{\partial^2 P}{\partial Z \partial X}(\mathbf{X}_c) = l_{XZ}. \end{cases} \quad (10)$$

Here,  $l_{ab}$  ( $a, b = X, Y, Z$ ) are the components of the symmetric Cartesian curvature tensor given by

$$l_{ab} = \frac{1}{2} \left( \frac{\partial n_a}{\partial b} + \frac{\partial n_b}{\partial a} \right), \quad (a, b = X, Y, Z). \quad (11)$$

By solving (10) together with (9), the coefficients  $a_{str}$  are determined as follows:

$$\begin{cases} a_{100} = n_X - \frac{1}{2} (c^{(X)} l_{XX} + c^{(X)} c^{(Y)} l_{XY} + c^{(X)} c^{(Z)} l_{XZ}), \\ a_{010} = n_Y - \frac{1}{2} (c^{(Y)} l_{YY} + c^{(X)} c^{(Y)} l_{XY} + c^{(Y)} c^{(Z)} l_{YZ}), \\ a_{001} = n_Z - \frac{1}{2} (c^{(Z)} l_{ZZ} + c^{(X)} c^{(Z)} l_{XZ} + c^{(Y)} c^{(Z)} l_{YZ}), \\ a_{110} = c^{(X)} c^{(Y)} l_{XY}, & a_{011} = c^{(Y)} c^{(Z)} l_{YZ}, & a_{101} = c^{(X)} c^{(Z)} l_{XZ}, \\ a_{200} = \frac{1}{2} c^{(X)} l_{XX}, & a_{020} = \frac{1}{2} c^{(Y)} l_{YY}, & a_{002} = \frac{1}{2} c^{(Z)} l_{ZZ}. \end{cases} \quad (12)$$

The unit normal vector and the curvature tensor are obtained from the VOF function  $\phi$  at each step, and the specific forms of them are given later.

It should be remarked that, given a function  $P(\mathbf{X})$ ,  $d$  is determined by integrating the function  $\hat{H}(\mathbf{X})$  over the cubic cell in the local coordinate system  $[0, 1]^3$  using the volumetric constraint from (5):

$$\int_0^1 \int_0^1 \int_0^1 \hat{H}(\mathbf{X}) dX dY dZ = \phi. \quad (13)$$

Unfortunately, there is no general analytical expression within a class of elementary functions for multi-dimensional integral of the function  $\hat{H}(\mathbf{X})$ . In practice, we take a combined approach of employing a 1D analytical integration of the hyperbolic tangent function and the Gaussian quadrature for the multi-dimensional integral calculation. Here 1D analytical and numerical integration operators (along  $r$ -axis for  $r \in [0, 1]$ ),  $I_A^{(r)}$ : and  $I_N^{(r)}$ : , respectively, for function  $q(r) = (1 + \tanh(\beta(a_0 r + d)))/2$ , are given as

$$\begin{cases} I_A^{(r)} : q(r) = \frac{1}{2} \left( 1 + \frac{1}{\beta a_0} \ln \left( \frac{\cosh(\beta(a_0 + d))}{\cosh(\beta d)} \right) \right), & \text{for analytical,} \\ I_N^{(r)} : q(r) = \frac{1}{2} (q(r_p^-) + q(r_p^+)), & \text{for numerical,} \end{cases} \quad r \in [0, 1], \quad (14)$$

where the two-point Gaussian quadrature is applied with the Gaussian points  $r_p^\pm = (1 \pm 1/\sqrt{3})/2$ . A further consideration is given to express the integration of 1D hyperbolic tangent function with an analytical function in a specified direction, say  $X$  direction, by assuming that  $P(\mathbf{X})$  is a linear function of  $X$ . Then, its derivative  $\partial_X P(\mathbf{X}) = P_X(Y, Z)$  only depends on  $Y$  and  $Z$ . Therefore,  $P_X(Y, Z)$  should be a constant in the analytical expression of the integration of (7) in  $X$  direction,

$$\int_0^1 \hat{H}(\mathbf{X}) dX = \frac{1}{2} \left( 1 + \frac{1}{\beta P_X(Y, Z)} \ln \left( \frac{\cosh(\beta(P(1, Y, Z) + d))}{\cosh(\beta(P(0, Y, Z) + d))} \right) \right). \quad (15)$$

Given specific forms of  $P(X, Y, Z)$  and  $P_X(Y, Z)$ , the multi-dimensional integration is completed by applying the two-point Gaussian quadrature in both  $Y$  and  $Z$  directions, and thus  $d$  can be uniquely determined as follows:

$$I_N^{(r_N)} : \left( I_N^{(r_N)} : \left( I_A^{(r_A)} : \hat{H}(\mathbf{X}) \right) \right) = \phi \rightarrow d, \quad (16)$$

where  $r_{A, N} = X, Y, Z$  denote the respective spatial directions of the integration.

Taking above remarks into account, we consider the following three cases to determine the quadratic surface function  $P(\mathbf{X})$  and  $d$ .

- Case 1:  $|n_X| = \max(|n_X|, |n_Y|, |n_Z|)$
- $c^{(X)} = 0$ ,  $c^{(Y)} = c^{(Z)} = 1$  in (9) and (12).
  - $r_A = X$ ,  $r_N = Y, Z$  in (16).
- Case 2:  $|n_Y| = \max(|n_X|, |n_Y|, |n_Z|)$
- $c^{(Y)} = 0$ ,  $c^{(X)} = c^{(Z)} = 1$  in (9) and (12).
  - $r_A = Y$ ,  $r_N = X, Z$  in (16).
- Case 3:  $|n_Z| = \max(|n_X|, |n_Y|, |n_Z|)$
- $c^{(Z)} = 0$ ,  $c^{(X)} = c^{(Y)} = 1$  in (9) and (12).
  - $r_A = Z$ ,  $r_N = X, Y$  in (16).

The unit normal vector in the global coordinate system is defined as  $\mathbf{n} = \nabla \phi / |\nabla \phi|$ . Here we introduce  $\mathbf{m} = \nabla \phi$  as the gradient of the VOF function. We employ the Youngs approach [65,66] in which the normal vector is averaged and normalized by the values of  $\mathbf{m}$  at the surrounding cell corners. Here, we focus on the 2D case. First, the values of the derivative at the four cell corners indexed as  $i \pm \frac{1}{2}j \pm \frac{1}{2}$  are calculated by the VOF function on its surroundings, e.g.,

$$\begin{aligned} m_{xi+\frac{1}{2}j+\frac{1}{2}} &= \frac{\phi_{i+1j} + \phi_{i+1j+1} - \phi_{ij} - \phi_{ij+1}}{\Delta x_i + \Delta x_{i+1}}, \\ m_{yi+\frac{1}{2}j+\frac{1}{2}} &= \frac{\phi_{ij+1} + \phi_{i+1j+1} - \phi_{ij} - \phi_{i+1j}}{\Delta y_j + \Delta y_{j+1}}. \end{aligned} \quad (17)$$

By introducing the local coordinate system (6), the unit normal vector on a cell  $\delta V_{ij}$  is given by

$$(n_{xij}, n_{yij}) = \left( \frac{\Delta x_i m_{xij}}{\sqrt{(m_{xij})^2 + (m_{yij})^2 + \epsilon}}, \frac{\Delta y_j m_{yij}}{\sqrt{(m_{xij})^2 + (m_{yij})^2 + \epsilon}} \right), \quad (18)$$

where  $\epsilon$  is a small positive of about  $10^{-16}$  to avoid the zero division, and,

$$\begin{aligned} m_{xij} &= \frac{1}{4} (m_{xi-\frac{1}{2}j-\frac{1}{2}} + m_{xi+\frac{1}{2}j-\frac{1}{2}} + m_{xi-\frac{1}{2}j+\frac{1}{2}} + m_{xi+\frac{1}{2}j+\frac{1}{2}}), \\ m_{yij} &= \frac{1}{4} (m_{yi-\frac{1}{2}j-\frac{1}{2}} + m_{yi+\frac{1}{2}j-\frac{1}{2}} + m_{yi-\frac{1}{2}j+\frac{1}{2}} + m_{yi+\frac{1}{2}j+\frac{1}{2}}). \end{aligned} \quad (19)$$

The components of the symmetric Cartesian curvature tensor (11) at the cell centers are calculated from the surrounding  $\mathbf{n}_{i \pm \frac{1}{2}j \pm \frac{1}{2}} = (\mathbf{m} / \sqrt{\mathbf{m}^2 + \epsilon})_{i \pm \frac{1}{2}j \pm \frac{1}{2}}$  as:

$$\begin{aligned} l_{xxij} &= \Delta x_i \left( \frac{n_{xi+\frac{1}{2}j-\frac{1}{2}} + n_{xi+\frac{1}{2}j+\frac{1}{2}}}{2} - \frac{n_{xi-\frac{1}{2}j-\frac{1}{2}} + n_{xi-\frac{1}{2}j+\frac{1}{2}}}{2} \right), \\ l_{yyij} &= \Delta y_j \left( \frac{n_{yi-\frac{1}{2}j+\frac{1}{2}} + n_{yi+\frac{1}{2}j+\frac{1}{2}}}{2} - \frac{n_{yi-\frac{1}{2}j-\frac{1}{2}} + n_{yi+\frac{1}{2}j-\frac{1}{2}}}{2} \right), \\ l_{xyij} &= \frac{\Delta y_j}{2} \left( \frac{n_{yi+\frac{1}{2}j-\frac{1}{2}} + n_{yi+\frac{1}{2}j+\frac{1}{2}}}{2} - \frac{n_{yi-\frac{1}{2}j-\frac{1}{2}} + n_{yi-\frac{1}{2}j+\frac{1}{2}}}{2} \right) \\ &\quad + \frac{\Delta x_i}{2} \left( \frac{n_{xi-\frac{1}{2}j+\frac{1}{2}} + n_{xi+\frac{1}{2}j+\frac{1}{2}}}{2} - \frac{n_{xi-\frac{1}{2}j-\frac{1}{2}} + n_{xi+\frac{1}{2}j-\frac{1}{2}}}{2} \right). \end{aligned} \quad (20)$$

This procedure applies straightforwardly in 3D to obtain the normal vector and Cartesian curvature tensor. Note that  $l_{xy}$  may not be needed in 2D case since the cross term  $XY$  does not appear in  $P(X, Y)$  (see Appendix A.1), yet  $l_{xy}$ ,  $l_{yz}$  and  $l_{xz}$  are necessary to approximate the surface function  $P(\mathbf{X})$  in 3D and can be computed in the same manner shown above. The curvature  $\kappa_{ij}$  is then given by

$$\kappa_{ij} = -(\nabla \cdot \mathbf{n})_{ij} = -\left( \frac{l_{xxij}}{\Delta x_i^2} + \frac{l_{yyij}}{\Delta y_j^2} \right). \quad (21)$$

Since the two-point Gaussian quadratures are applied for two directions, a quartic equation is resulted for  $d$ . Fortunately, the volumetric constraint (13) leads to an analytical formula to uniquely determine  $d$ . The details are given in Appendices A and B.

The approximate function  $\hat{H}_{ijk}(\mathbf{X})$  has a fractional value only when the cell includes an interface, thus we assume the approximate function to be piecewise constant in a case when the VOF value is very close to 0 or 1, i.e.,

$$\hat{H}_{ijk}(\mathbf{X}) = \begin{cases} \hat{H}_{ijk}(\mathbf{X}), & \text{if } \phi_{\min} \leq \phi_{ijk} \leq 1 - \phi_{\min}, \\ \phi_{ijk}, & \text{otherwise.} \end{cases} \quad (22)$$

In the present paper, we employ  $\phi_{\min}$  to  $10^{-8}$ .

As long as the approximate function  $\hat{H}(\mathbf{X})$  is constructed, the numerical fluxes needed to update the VOF function can be directly computed. This will be discussed in Section 2.3.

### 2.3. Updating the VOF function

The multi-dimensional time evolution Eq. (3) for VOF function  $\phi$  can be calculated by either evaluating the numerical fluxes sequentially in each direction or evaluating all the numerical fluxes simultaneously at a single step. We refer to the first approach as directional splitting method and the second one as non-splitting method. Since the directional splitting method is robust and easy to compute numerical flux in each direction on the Cartesian coordinate mesh, we employ the directional splitting method in the present paper.

A staggered arrangement of the variables is employed; the velocity components are defined at the cell boundaries normal to the corresponding directions, i.e.  $u_{i\pm\frac{1}{2}jk}$ ,  $v_{ij\pm\frac{1}{2}k}$  and  $w_{ijk\pm\frac{1}{2}}$ , while the VOF function,  $\phi_{ijk}$ , is located at the cell center. Given the velocity field and the VOF function at the  $n$ th step ( $t = t^n$ ),  $\phi^n$ , the advection Eq. (3) is discretized in space and time in the directional splitting manner,

$$\begin{cases} \phi_{ijk}^* = \phi_{ijk}^n - \frac{1}{\Delta x_i} (\hat{f}_{i+\frac{1}{2}jk}^n - \hat{f}_{i-\frac{1}{2}jk}^n) + \frac{\Delta t}{\Delta x_i} \bar{\phi}_{ijk}^{(1)} (u_{i+\frac{1}{2}jk} - u_{i-\frac{1}{2}jk}), \\ \phi_{ijk}^{**} = \phi_{ijk}^* - \frac{1}{\Delta y_j} (\hat{g}_{ij+\frac{1}{2}k}^* - \hat{g}_{ij-\frac{1}{2}k}^*) + \frac{\Delta t}{\Delta y_j} \bar{\phi}_{ijk}^{(2)} (v_{ij+\frac{1}{2}k} - v_{ij-\frac{1}{2}k}), \\ \phi_{ijk}^{n+1} = \phi_{ijk}^{**} - \frac{1}{\Delta z_k} (\hat{h}_{ijk+\frac{1}{2}}^{**} - \hat{h}_{ijk-\frac{1}{2}}^{**}) + \frac{\Delta t}{\Delta z_k} \bar{\phi}_{ijk}^{(3)} (w_{ijk+\frac{1}{2}} - w_{ijk-\frac{1}{2}}), \end{cases} \quad (23)$$

where  $\Delta t = t^{n+1} - t^n$ , and  $\bar{\phi}_{ijk}^{(1)}$ ,  $\bar{\phi}_{ijk}^{(2)}$  and  $\bar{\phi}_{ijk}^{(3)}$  denote the time-averaged values for each sub-step. The numerical fluxes,  $\hat{f}^n$ ,  $\hat{g}^*$  and  $\hat{h}^{**}$ , are defined as

$$\begin{cases} \hat{f}_{i\pm\frac{1}{2}jk}^n \equiv \frac{1}{\Delta y_j \Delta z_k} \int_{\delta t^n} \int_{\delta y_j} \int_{\delta z_k} (uH(\mathbf{x}, t))_{i\pm\frac{1}{2}jk} dy dz dt, \\ \hat{g}_{ij\pm\frac{1}{2}k}^* \equiv \frac{1}{\Delta x_i \Delta z_k} \int_{\delta t^*} \int_{\delta x_i} \int_{\delta z_k} (vH(\mathbf{x}, t))_{ij\pm\frac{1}{2}k} dx dz dt, \\ \hat{h}_{ijk\pm\frac{1}{2}}^{**} \equiv \frac{1}{\Delta x_i \Delta y_j} \int_{\delta t^{**}} \int_{\delta x_i} \int_{\delta y_j} (wH(\mathbf{x}, t))_{ijk\pm\frac{1}{2}} dx dy dt. \end{cases} \quad (24)$$

Here, we introduce intermediate time regions as  $\delta t^n = [t^n, t^*]$ ,  $\delta t^* = [t^*, t^{**}]$  and  $\delta t^{**} = [t^{**}, t^{n+1}]$  of the same interval  $\Delta t = |\delta t^n| = |\delta t^*| = |\delta t^{**}|$ . The details for the evaluation of the numerical fluxes are described in Section 2.4.

Since the directional splitting approach separates the computation of divergence into a sequence of one dimensional operations, as shown in the last terms on the right hand sides of (23), a particular care must be paid to make the numerical solution consistent with the condition that the fully multi-dimensional operator satisfies. We employ the formulation in [36,1] to satisfy the divergence free condition in the directional splitting approach. Following [36,1],  $\bar{\phi}_{ijk}^{(1)}$ ,  $\bar{\phi}_{ijk}^{(2)}$  and  $\bar{\phi}_{ijk}^{(3)}$  in (23) are implicitly treated and the contribution in each direction is summed up for  $\phi^{n+1}$ , as summarized in the following:

$$\begin{cases} \phi_{ijk}^* = \phi_{ijk}^n - \frac{1}{\Delta x_i} (\hat{f}_{i+\frac{1}{2}jk}^n - \hat{f}_{i-\frac{1}{2}jk}^n) + \frac{\Delta t}{\Delta x_i} \phi_{ijk}^* (u_{i+\frac{1}{2}jk} - u_{i-\frac{1}{2}jk}), \\ \phi_{ijk}^{**} = \phi_{ijk}^* - \frac{1}{\Delta y_j} (\hat{g}_{ij+\frac{1}{2}k}^* - \hat{g}_{ij-\frac{1}{2}k}^*) + \frac{\Delta t}{\Delta y_j} \phi_{ijk}^{**} (v_{ij+\frac{1}{2}k} - v_{ij-\frac{1}{2}k}), \\ \phi_{ijk}^{***} = \phi_{ijk}^{**} - \frac{1}{\Delta z_k} (\hat{h}_{ijk+\frac{1}{2}}^{**} - \hat{h}_{ijk-\frac{1}{2}}^{**}) + \frac{\Delta t}{\Delta z_k} \phi_{ijk}^{***} (w_{ijk+\frac{1}{2}} - w_{ijk-\frac{1}{2}}), \\ \phi_{ijk}^{n+1} = \phi_{ijk}^{***} - \Delta t \left( \phi_{ijk}^* \frac{u_{i+\frac{1}{2}jk} - u_{i-\frac{1}{2}jk}}{\Delta x_i} + \phi_{ijk}^{**} \frac{v_{ij+\frac{1}{2}k} - v_{ij-\frac{1}{2}k}}{\Delta y_j} + \phi_{ijk}^{***} \frac{w_{ijk+\frac{1}{2}} - w_{ijk-\frac{1}{2}}}{\Delta z_k} \right). \end{cases} \quad (25)$$

For further improvement of the accuracy, the Strang splitting [44] can be easily implemented by interchanging the order of the fractional steps in this multi-dimensional calculation.

#### 2.4. Computation of the numerical flux

The numerical fluxes defined in (24) are effectively obtained by replacing the temporal integration by the spatial integration along the upwind path on the velocity field. Given the velocity component in  $x$  direction,  $u_{i+\frac{1}{2}jk}$ , for example, the upstream path can be simply determined as  $\delta x_{up}^+ \equiv [x_{i+\frac{1}{2}}, -\Delta t u_{i+\frac{1}{2}jk}, x_{i+\frac{1}{2}}]$  for  $u_{i+\frac{1}{2}jk} \geq 0$  or  $\delta x_{up}^- \equiv [x_{i+\frac{1}{2}}, x_{i+\frac{1}{2}} - \Delta t u_{i+\frac{1}{2}jk}]$  for  $u_{i+\frac{1}{2}jk} < 0$ . Then, the flux component  $\hat{f}_{i\pm\frac{1}{2}jk}^n$  in  $x$  direction is obtained by integrating the indicator function over the upwind area based on either of the upstream path  $\delta x_{up}^+$  or  $\delta x_{up}^-$ :

$$\hat{f}_{i\pm\frac{1}{2}jk}^n = \begin{cases} \frac{1}{\Delta y_j \Delta z_k} \int_{\delta x_{up}^+} \int_{\delta y_j} \int_{\delta z_k} H(\mathbf{x}, t^n) dx dy dz, & \text{for } u_{i+\frac{1}{2}jk} \geq 0, \\ -\frac{1}{\Delta y_j \Delta z_k} \int_{\delta x_{up}^-} \int_{\delta y_j} \int_{\delta z_k} H(\mathbf{x}, t^n) dx dy dz, & \text{for } u_{i+\frac{1}{2}jk} < 0. \end{cases} \quad (26)$$

Here we substitute  $H(\mathbf{x}, t)$  with the approximate indicator function  $\hat{H}_{ijk}^n(\mathbf{X})$  which is piecewisely constructed approximate function (7) from the VOF function  $\phi^n$  at step  $n$ . In summary, the numerical flux  $\hat{f}_{i\pm\frac{1}{2}jk}^n$  is obtained as,

$$\hat{f}_{i\pm\frac{1}{2}jk}^n = \begin{cases} \Delta x_i \int_{\delta x_{up}^+} \int_{\delta y} \int_{\delta z} \hat{H}_{ijk}^n(\mathbf{X}) dX dY dZ, & \text{for } u_{i+\frac{1}{2}jk} \geq 0, \\ -\Delta x_{i+1} \int_{\delta x_{up}^-} \int_{\delta y} \int_{\delta z} \hat{H}_{i+1jk}^n(\mathbf{X}) dX dY dZ, & \text{for } u_{i+\frac{1}{2}jk} < 0, \end{cases} \quad (27)$$

where the integration regions are  $\delta x_{up}^+ = [1 - \frac{\Delta t}{\Delta x_i} u_{i+\frac{1}{2}jk}, 1]$ ,  $\delta x_{up}^- = [0, -\frac{\Delta t}{\Delta x_{i+1}} u_{i+\frac{1}{2}jk}]$  and  $\delta y = \delta z = [0, 1]$ .

In the similar way, we obtain the numerical flux in  $y$  and  $z$  directions after the approximate indicator function  $\hat{H}_{ijk}^*(\mathbf{X})$  and  $\hat{H}_{ijk}^{**}(\mathbf{X})$  are constructed by using the intermediate values of the VOF function,  $\phi^*$  and  $\phi^{**}$ , respectively, at the corresponding sub-steps as,



$$\hat{g}_{ij+\frac{1}{2}k}^* = \begin{cases} \Delta y_j \int_{\delta Y_{up}^+} \int_{\delta X} \int_{\delta Z} \hat{H}_{ijk}^*(\mathbf{X}) dX dY dZ, & \text{for } v_{ij+\frac{1}{2}k} \geq 0, \\ -\Delta y_{j+1} \int_{\delta Y_{up}^-} \int_{\delta X} \int_{\delta Z} \hat{H}_{ij+1k}^*(\mathbf{X}) dX dY dZ, & \text{for } v_{ij+\frac{1}{2}k} < 0, \end{cases} \quad (28)$$

with  $\delta Y_{up}^+ = \left[1 - \frac{\Delta t}{\Delta y_j} v_{ij+\frac{1}{2}k}, 1\right]$ ,  $\delta Y_{up}^- = \left[0, -\frac{\Delta t}{\Delta y_{j+1}} v_{ij+\frac{1}{2}k}\right]$  and  $\delta X = \delta Z = [0, 1]$ , and

$$\hat{h}_{ijk+\frac{1}{2}}^{**} = \begin{cases} \Delta z_k \int_{\delta Z_{up}^+} \int_{\delta X} \int_{\delta Y} \hat{H}_{ijk}^{**}(\mathbf{X}) dX dY dZ, & \text{for } w_{ijk+\frac{1}{2}} \geq 0, \\ -\Delta z_{k+1} \int_{\delta Z_{up}^-} \int_{\delta X} \int_{\delta Y} \hat{H}_{ijk+1}^{**}(\mathbf{X}) dX dY dZ, & \text{for } w_{ijk+\frac{1}{2}} < 0, \end{cases} \quad (29)$$

with  $\delta Z_{up}^+ = \left[1 - \frac{\Delta t}{\Delta z_k} w_{ijk+\frac{1}{2}}, 1\right]$ ,  $\delta Z_{up}^- = \left[0, -\frac{\Delta t}{\Delta z_{k+1}} w_{ijk+\frac{1}{2}}\right]$  and  $\delta X = \delta Y = [0, 1]$ .

As a summary of this section, we provide the following remarks as the distinguished points of our method.

- A piecewise multi-dimensional hyperbolic tangent function can be uniquely constructed from the given VOF distribution. The explicit manipulation of the geometric components, which is the essential part in a conventional PLIC algorithm, is not required here.
- Not only a plane surface as in most of the PLIC type methods, a curved surface (as a quadratic form in the present paper) can also be constructed in a straightforward manner.
- Using the hyperbolic tangent function (the sigmoid function) that has a transition zone with a controllable thickness, the present scheme essentially removes the numerical diffusion and smearing, and thus keeps the moving interface well regulated with a compact thickness.

### 3. Numerical solver for incompressible two-phase fluid flow with moving interface

The incompressible and immiscible two-phase fluid model with moving interface is governed by the following equations:

$$\begin{aligned} \frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}H) &= \phi \nabla \cdot \mathbf{u}, \\ \nabla \cdot \mathbf{u} &= 0, \\ \rho \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) &= -\nabla p + \nabla \cdot (\mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T)) + \sigma \kappa \mathbf{n} \delta - \rho g \mathbf{e}_z, \end{aligned} \quad (30)$$

where  $p$  is the pressure,  $\sigma$  the surface tension coefficient,  $\delta$  the 1D delta function,  $g$  the gravitational acceleration and  $\mathbf{e}_z = (0, 0, 1)$  denoting the unit vector in the vertical direction. The mixture density and dynamic viscosity are simply averaged from phase density  $\rho_{1,2}$  and viscosity  $\mu_{1,2}$  in terms of the VOF function,

$$\begin{cases} \rho = \phi \rho_1 + (1 - \phi) \rho_2, \\ \mu = \phi \mu_1 + (1 - \phi) \mu_2. \end{cases} \quad (31)$$

The surface tension force is computed by the continuum surface force (CSF) model [3]. In the present study, the VOF function across the interface smoothly distributes over several computational meshes, thus the 1D approximate delta function is directly approximated by  $\delta \approx |\nabla \phi|$ . Then the surface tension force is estimated by

$$\sigma \kappa \mathbf{n} \delta \approx \sigma \kappa \nabla \phi, \quad (32)$$

where the normal vector is given by  $\mathbf{n} = \nabla \phi / |\nabla \phi|$ . The parasitic current caused by the numerical errors in the curvature estimation has been effectively suppressed by the continuous VOF distribution with the curved surface reconstruction even without using the convolution technique in the CSF model [3].

The projection approach based on [14,22] is employed to enforce the divergence free condition of the velocity field. Hereafter, we use the superscript  $n$  to represent a value at the  $n$ th time level, and  $*$  denotes a value at an intermediate level. Note that these intermediate levels are different from those defined in (25).

The numerical procedure is summarized in the following. First, the VOF function is updated with a divergence free velocity using the numerical method described in Section 2,

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}^n H) = \phi \nabla \cdot \mathbf{u}^n \rightarrow \phi^{n+1}, \quad (33)$$

followed by the calculation of the mixture density and viscosity by (31).

Then, we obtain the predicted velocity  $\mathbf{u}^*$  by solving the equation of motion,

$$\rho^{n+1} \left( \frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + \frac{3}{2} \mathbf{u}^n \cdot \nabla \mathbf{u}^n - \frac{1}{2} \mathbf{u}^{n-1} \cdot \nabla \mathbf{u}^{n-1} \right) = -\nabla p^n + \frac{1}{2} \nabla \cdot (\boldsymbol{\tau}^n + \boldsymbol{\tau}^*) + \sigma \kappa^{n+1} \nabla \phi^{n+1} - \rho^{n+1} g \mathbf{e}_z, \quad (34)$$

where the viscous stress terms are given by

$$\begin{aligned} \boldsymbol{\tau}^n &= \mu^{n+1} (\nabla \mathbf{u}^n + (\nabla \mathbf{u}^n)^T) + \mathbf{T}^n, \\ \boldsymbol{\tau}^* &= \mu^{n+1} (\nabla \mathbf{u}^* + (\nabla \mathbf{u}^*)^T). \end{aligned} \quad (35)$$



Here,  $\mathbf{T}$  is a stress increment tensor which will be determined later. A second-order Adams–Bashforth method is applied for the advection term, and the Crank–Nicolson method is for the viscous stress term.

Subsequently, we solve a Poisson equation for the pressure correction  $\psi$  by using the intermediate velocity,

$$\nabla \cdot \left( \frac{\nabla \psi^{n+1}}{\rho^{n+1}} \right) = \frac{\nabla \cdot \mathbf{u}^*}{\Delta t}. \quad (36)$$

Then, the velocity and pressure are updated as follows,

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \frac{\nabla \psi^{n+1}}{\rho^{n+1}}, \quad (37)$$

$$p^{n+1} = p^n + \psi^{n+1}. \quad (38)$$

The time-increment of  $\mathbf{T}$  is completed by substituting  $\mathbf{u}^* = \mathbf{u}^{n+1} + \Delta t \psi^{n+1} / \rho^{n+1}$  into (34),

$$\mathbf{T}^{n+1} = \Delta t \mu^{n+1} \left( \nabla \left( \frac{\nabla \psi^{n+1}}{\rho^{n+1}} \right) + \nabla \left( \frac{\nabla \psi^{n+1}}{\rho^{n+1}} \right)^T \right). \quad (39)$$

The variables are assigned in a staggered arrangement, i.e. the pressure and VOF function are defined at cell centers, whereas the velocity components are defined on cell faces in the respective directions. For the tensorial components of  $\boldsymbol{\tau}$  (and  $\mathbf{T}$ ), the diagonal components  $\tau_{xx}$ ,  $\tau_{yy}$  and  $\tau_{zz}$  are computed at the cell centers, and other components  $\tau_{xy}$ ,  $\tau_{yz}$  and  $\tau_{xz}$  are computed at cell edges in parallel with  $z$ ,  $x$  and  $y$  axes, respectively. The second-order MUSCL (Monotone Upstream-centered Schemes for Conservation Laws) method [55] is employed for the convection term in the equation of motion. For the other terms, the second-order central difference scheme is employed.

#### 4. Numerical examples

Due to a requirement for the convergence of the approximate delta function in the global coordinate system, the interface width, typically characterized by a transition of the approximate indicator function between 0 and 1, becomes smaller with the mesh refinement. In the present formulation, since the parameter  $\beta$  in (7) determines the sharpness of the approximate function in the frame of local coordinate system, we specify a constant value to the parameter  $\beta$  depending on the problem. We confirmed that the interface is sharply captured within a range of  $\beta$  between 1 and 4 from our preliminary numerical experiment.

In the present approach of the continuous sigmoid function, a VOF profile has a continuous profile. Therefore, the initial VOF function  $\phi|_{t=t_0} = \phi^0$  on a cell  $\delta V_{ijk}$  is given by

$$\phi_{ijk}^0 = \frac{1}{\Delta x_i \Delta y_j \Delta z_k} \int_{\delta x_i} \int_{\delta y_j} \int_{\delta z_k} \tilde{H}^0(\mathbf{x}) dx dy dz, \quad (40)$$

with an initial smoothed indicator function:

$$\tilde{H}^0(\mathbf{x}) = \frac{1}{2} \left( 1 + \tanh \left( \frac{\beta}{h} \Psi(\mathbf{x}) \right) \right), \quad (41)$$

where  $\Psi(\mathbf{x})$  is a signed distance function to represent the interface, and  $h$  is a characteristic length which is related to the mesh size. In an isotropic uniform mesh system, for example,  $h = \Delta x = \Delta y = \Delta z$ .

##### 4.1. Accuracy and the geometric faithfulness of the interface reconstruction

First, we investigate the accuracy and geometric faithfulness of the interface reconstructions for advected interfaces in two dimensions. Numerical results with the plane (linear) surface and curved (quadratic) surface reconstructions are examined and compared.

In 2D, we define the linear interface by  $P(X, Y) = n_x X + n_y Y$ , and the quadratic interface by  $P(X, Y) = (l_{yy}/2)Y^2 + n_x X + (n_y - l_{yy}/2)Y$  for  $|n_x| \geq |n_y|$  or  $P(X, Y) = (l_{xx}/2)X^2 + (n_x - l_{xx}/2)X + n_y Y$  for  $|n_x| < |n_y|$ .

##### 4.1.1. Interface reconstructions for a circular shape

An interface of circular shape with a radius of  $R = 0.433$  is located on  $[0, 1]^2$  domain. The VOF function distributes according to (40) and (41) with  $\Psi(x, y) = R - \sqrt{(x - 0.5)^2 + (y - 0.5)^2}$ . A sharpness parameter  $\beta = 1$  is used.

The reconstructed interface and the exact circular interface are shown in Fig. 1 with both linear and quadratic surface approximations on a  $4 \times 4$  mesh. The linear approximation gives a result quite similar to the conventional PLIC method, whereas the quadratic approximation shows a remarkable improvement in capturing the geometry of the circular curve.

We investigate the numerical errors for the reconstructed interface in terms of the  $L_1$ ,  $L_2$  and  $L_\infty$  norms by sampling the numerical results at the points evenly distributed over the reconstructed interface (a total of 1000 surface points on each mesh). Note that these sample points are only used for visualization, more precisely these are not used for reconstructing the interface. As discussed in [34,7,11,10,30], the numerical accuracy of the normal vector and curvature (tensor) essentially affects the accuracy of the reconstructed interface. Using the Youngs approach [65,66] as discussed in Section 2.2 for obtaining the normal vector and curvature tensor from the VOF function, the convergence behavior through the mesh refinement test is shown in Fig. 2. The convergence rate is of a first-order accuracy in average for both linear and quadratic interfaces. To see the effect of approximate normal vector  $n_a$  and Cartesian curvature tensor  $l_{ab}$  (20) in the quadratic reconstruction, a test reconstruction is attempted with the exact  $n_a$  and  $l_{ab}$  rather than obtaining by the Youngs approach. The convergence rates are plotted in Fig. 3. In this case, the second-order accuracy is observed. Moreover, the error with the quadratic interface exhibits a reduced level than that of the linear interface, showing the superiority of quadratic reconstruction.

These results suggest that the accuracy of the normal vector and curvature is the key issue to get the overall accuracy for the interface. Some existing works on accurate retrieval of the normal vector and curvature can be found in [34,7,11,10,30]. However, since this is a topic beyond the scope of the present paper, we will use the simple approach of Youngs in the rest of the paper.

#### 4.1.2. Rotation of an asteroid-like body

To demonstrate the capability of the present method in capturing the moving interface with complex geometry, we carry out a revolution test of an asteroid-like body defined by  $r(\theta) = 0.3 \times (1 + \cos(10\theta)/2)$ , where  $\tan\theta = (y - 0.5)/(x - 0.5)$ , on  $[0, 1]^2$  domain. The rotational velocity  $\mathbf{u} = (0.5 - y, x - 0.5)$  is imposed, thus the body rotates in the counter-clockwise direction around its mass center. The number of cells along one end of the domain is doubly increased as  $N = 50, 100$  and  $200$ , and a mesh-dependent time increment  $\Delta t = 2\pi/628 \times 100/N$  is used. The sharpness parameter  $\beta = 1$  is employed.

The numerical results at  $t = 10\pi$  (after five revolutions) with the linear interface approximation are shown in Fig. 4 for the different cell numbers. An obvious lag in the phase speed and the asymmetrical error are observed in the numerical result on the  $N = 50$  mesh. The numerical solution can be improved with the refined meshes. These numerical errors are substantially improved when the quadratic approximation is used for the interface. As shown in Fig. 5, the interface is transported with less phase lag, and the asymmetry is removed even on the mesh of the lowest resolution.

#### 4.1.3. Rotation of a slotted-disk

A well known Zalesak's rotation problem [68] is attempted. A slotted disk with the inside indicator value unity is defined by

$$((x - 0.5)^2 + (y - 0.75)^2) \leq 0.15^2 \wedge (|x - 0.5| \geq 0.025 \vee y \geq 0.85). \quad (42)$$

The rotational velocity  $\mathbf{u} = (0.5 - y, x - 0.5)$  is imposed. We set the maximum CFL number to be 0.5 and the sharpness parameter  $\beta = 2$ . Numerical tests are carried out on a mesh employing  $N = 100$  on one side of the domain with both linear and quadratic interfaces. Fig. 6 shows the results at  $t = 2\pi$  (one revolution) and  $t = 10\pi$  (five revolutions).

The similar observation to the above test is obtained; numerical errors, such as the phase lag and the asymmetry, are remarkable in the linear reconstruction in the regions where the interface has large or discontinuous curvature. On the other hand, the quadratic surface reconstruction preserves the symmetry of the initial geometry without the phase lag even after

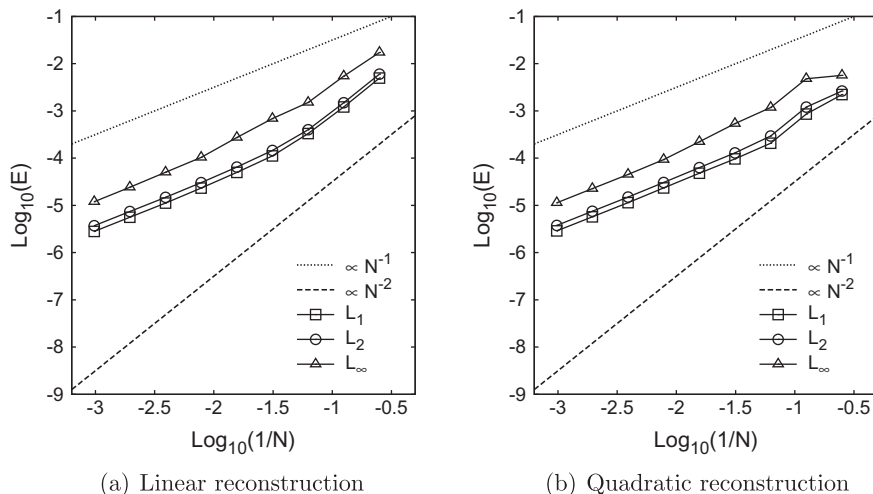
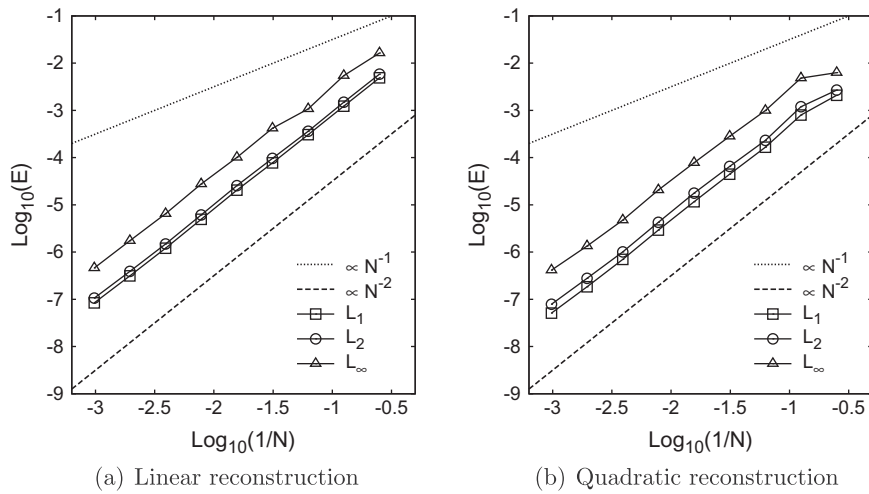
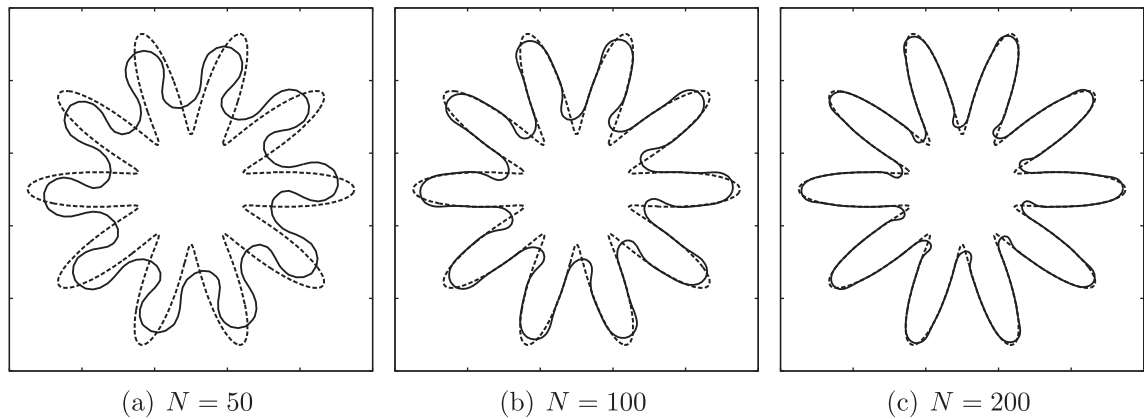


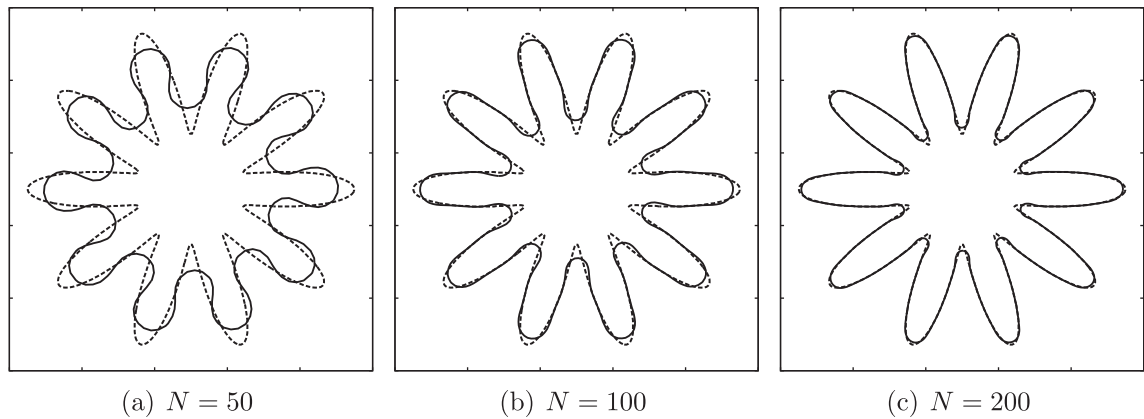
Fig. 2. Reconstruction errors for a circular interface by using the numerically approximated normal vector and curvature (in the quadratic reconstruction).



**Fig. 3.** Same as Fig. 2, but by using the exact normal vector and curvature.

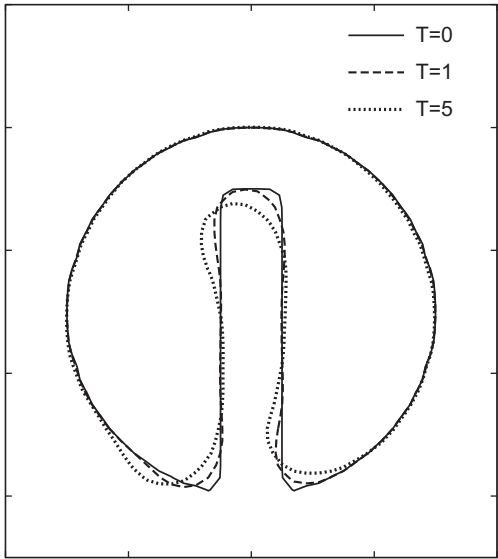


**Fig. 4.** Profiles with the linear reconstruction for different resolutions after five revolutions. Dashed line is the exact solution and the solid line is the numerical solution.

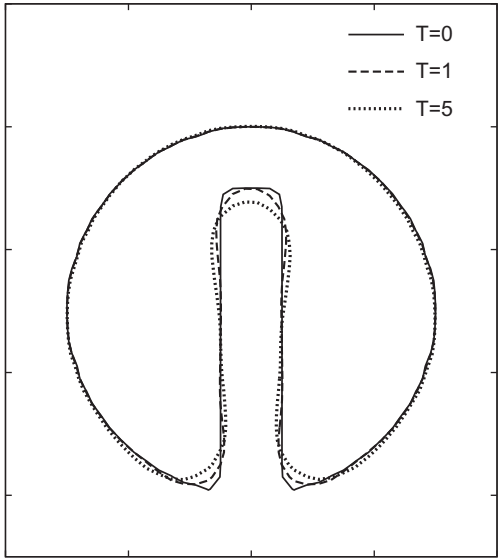


**Fig. 5.** Same as Fig. 4, but with the quadratic reconstruction.

five revolutions. This, again, demonstrates the advantage of the quadratic interface reconstruction over the linear one. Table 1 shows the numerical errors measured by



(a) Linear reconstruction



(b) Quadratic reconstruction

**Fig. 6.** The solutions in the Zalesak's rotation problem on  $N = 100$  mesh. The initial profile and the numerical results after one and five revolutions are shown with the iso-contours of  $\phi = 0.5$ .

**Table 1**  
Numerical errors and the respective convergence rates in the revolution test of a slotted-disk, with linear and quadratic reconstructions.

Revolution	$N$	Linear		Quadratic	
		Error	Order	Error	Order
One	50	3.38E-2	–	2.93E-2	–
	100	1.75E-2	0.95	1.61E-2	0.86
	200	8.73E-3	1.00	7.91E-3	1.03
Five	50	8.73E-2	–	6.59E-2	–
	100	3.34E-2	1.39	2.57E-2	1.36
	200	1.60E-2	1.06	1.32E-2	0.96

$$E_\phi = \frac{\sum_{ij} \Delta x_i \Delta y_j |\phi_{ij}^n - \phi_{ij}^0|}{\sum_{ij} \Delta x_i \Delta y_j |\phi_{ij}^0|}, \quad (43)$$

on  $N = 50, 100$  and  $200$  meshes. Here  $\phi^0$  and  $\phi^n$  are the initial solution (the exact solution) and the numerical solution, respectively. Due to the numerical approximations in computing normal vector and curvature, the convergence rates of the volume error by both linear and quadratic interface reconstructions are nearly first-order. Yet, according to the error levels of the two approximations, the quadratic approximation gives more improved numerical solution.

As shown in the comparison test for the recovery of the exact (circular) geometry, the quadratic reconstruction showed better results because a higher-order approximate surface function facilitates to replicate complex geometries. Moreover, it is confirmed that this geometric faithfulness improves the results in the advection problem. In the present method, the quadratic surface can be reconstructed with a negligible increase in computational effort. Thus, we use the quadratic interface throughout the following tests.

#### 4.2. The influence of the sharpness parameter $\beta$

##### 4.2.1. The translation of a unit circle

A unit circle is initially located at the center of a computational domain  $[0, 4]^2$ . The circle is translated at a diagonal velocity  $\mathbf{u} = (1, 1)$  with periodic boundary conditions in both directions, and it comes back to the initial position at  $t = 4$ .

First we check our code with a CFL number 1, in which the calculations of the numerical fluxes in (27) and (28) are the same as those in the reconstruction of the indicator function (13), and thus the solution should be exactly retrieved. We specify the parameter  $\beta = 1$ . At  $t = 4$  (one period), the numerical errors measured by (43) are given in Table 2. It is found that the errors are within a level of the round-off error (machine epsilon).

Next, we carry out a computation with a CFL number 0.25, and investigate the influence of the different values of parameter  $\beta$ . Here we focus on two particular cases of  $\beta = 1$  and  $2$ . The numerical results at  $t = 4$  and  $t = 20$  (five periods) on  $N = 128$  mesh are shown in Figs. 7 and 8, respectively. In both results, the interface width is maintained without the numerical diffusion or smearing even after five periods. As expected, the transition of  $\phi$  is controllable by adjusting the parameter  $\beta$ , i.e.  $\beta = 2$  exhibits a sharper interface. However, the circular shape with  $\beta = 2$  after five periods is slightly deformed. The result means that a less steep transition of  $\phi$  is much preferred for capturing the geometry of the interface. This tendency might be somewhat case-dependent, which will be mentioned later. In this example, the solution with  $\beta = 1$  preserves the circular geometry well even after five periods. We also examine the convergence rate for the spatial resolution with the doubly-increased number of cells  $N = 32, 64, 128$  and  $256$  at  $t = 4$  and  $t = 20$ . In Table 3, we show the numerical errors for the VOF function measured by (43). The VOF function converges with approximately the first-order accuracy, and  $\beta = 1$  gives more reduced errors in all runs.

We further investigate the deformation modes of the interface. According to [48], the modal amplitudes on the Eulerian coordinate system are estimated by the volume integration with the approximate delta function  $|\nabla \phi|$ :

$$\begin{aligned} |R_0| &= \frac{1}{2\pi} \sum_{ij} \Delta x_i \Delta y_j |\nabla \phi_{ij}|, \\ |R_n| &= \sqrt{\left( \frac{1}{\pi} \sum_{ij} \Delta x_i \Delta y_j |\nabla \phi_{ij}| \cos(n\theta_{ij}) \right)^2 + \left( \frac{1}{\pi} \sum_{ij} \Delta x_i \Delta y_j |\nabla \phi_{ij}| \sin(n\theta_{ij}) \right)^2} \\ (n &= 1, 2, \dots), \\ \tan \theta_{ij} &= \frac{y_j - y_c}{x_i - x_c}, \end{aligned} \quad (44)$$

where  $\mathbf{x}_c = (x_c, y_c)$  is the circular centroid approximately given by

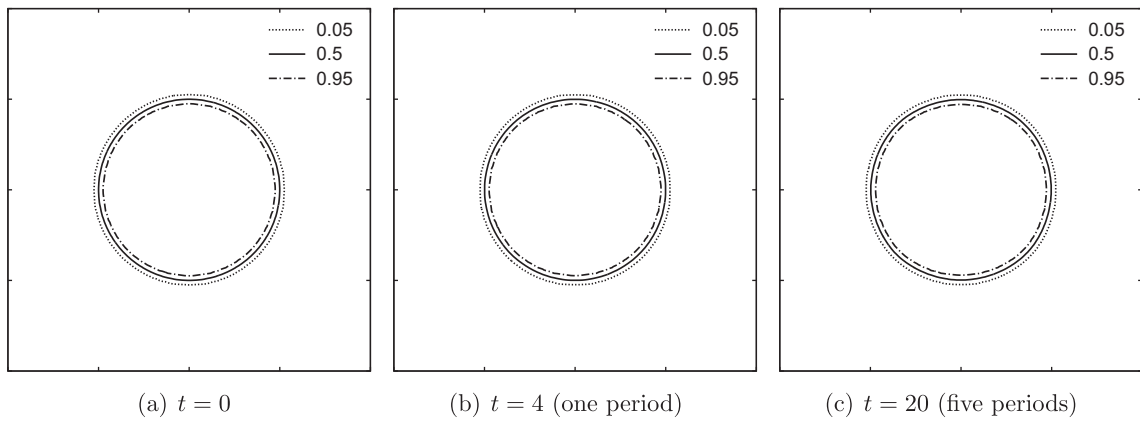
$$\mathbf{x}_c \approx \frac{\sum_{ij} \mathbf{x}_{ij} \phi_{ij}}{\sum_{ij} \phi_{ij}}. \quad (45)$$

Figs. 9 and 10 show convergence behaviors of the computed centroid and several modal amplitudes ( $n = 0, 1, 2, 3$ ) at  $t = 4$  and  $t = 20$  for  $\beta = 1$  and  $2$ , respectively. Interestingly, the centroid shown by filled circle almost converges with a second-order accuracy in both cases of  $\beta = 1$  and  $2$ . However, in the solution with  $\beta = 2$ , the convergence of the modal amplitudes degrades to a first-order accuracy, and the numerical solutions for the first and second-order modes do not show a uniform

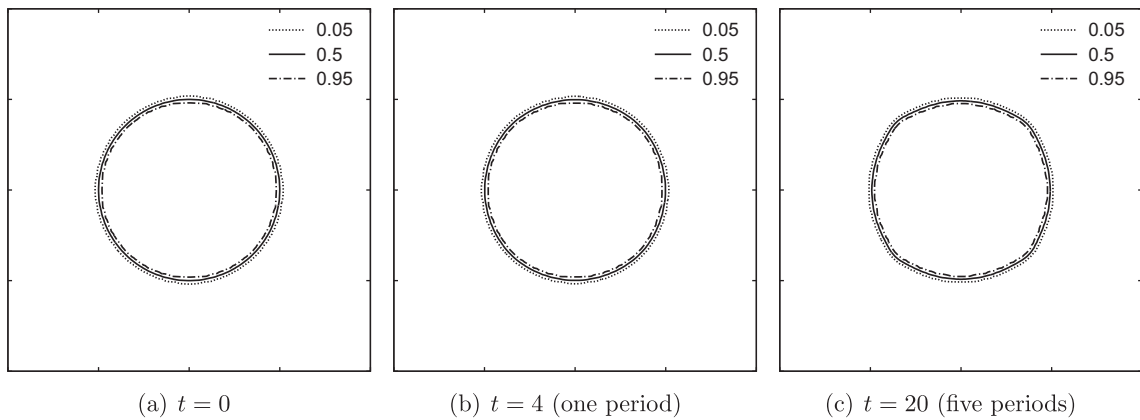
**Table 2**

Numerical errors in the translation in  $x$  direction of a unit circle at  $t = 4$ . The CFL number is 1.

$N$	32	64	128
Error	4.07E–15	2.52E–15	1.66E–15



**Fig. 7.** The VOF profiles before and after the translation of a unit circle at a diagonal velocity  $\mathbf{u} = (1, 1)$ . Three different iso-contours are shown at different periods of time. The number of cells  $N = 128$  and the sharpness parameter  $\beta = 1$  are used.



**Fig. 8.** Same as Fig. 7, but with sharpness parameter  $\beta = 2$ .

**Table 3**

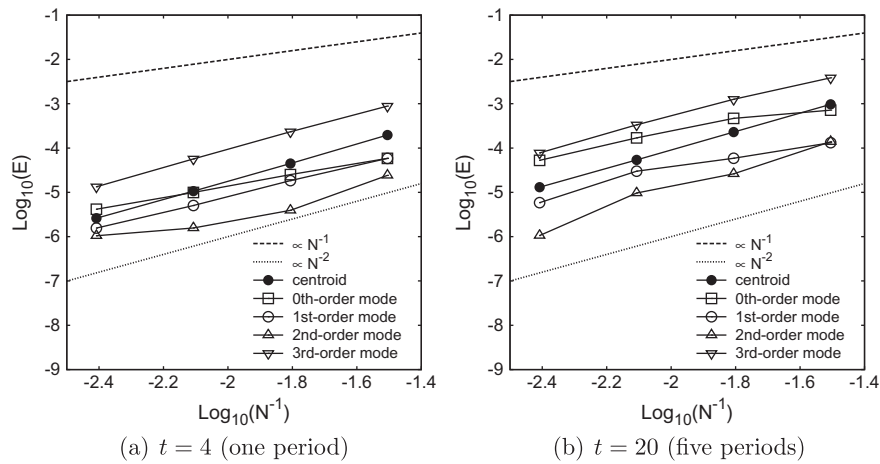
Numerical errors and the respective convergence rates in the diagonal translation of a unit circle at  $t = 4$  (one period) and  $t = 20$  (five periods). The CFL number is 0.25.

	$N$	$(\beta = 1)$		$(\beta = 2)$	
		Error	Order	Error	Order
$t = 4$	32	4.68E-3	–	1.71E-2	–
	64	2.77E-3	0.76	1.02E-2	0.75
	128	1.43E-3	0.95	5.22E-3	0.97
	256	7.18E-4	0.99	2.63E-3	0.99
$t = 20$	32	1.73E-2	–	5.82E-2	–
	64	1.32E-2	0.39	4.78E-2	0.28
	128	7.10E-3	0.89	2.63E-2	0.86
	256	3.59E-3	0.98	1.31E-2	1.01

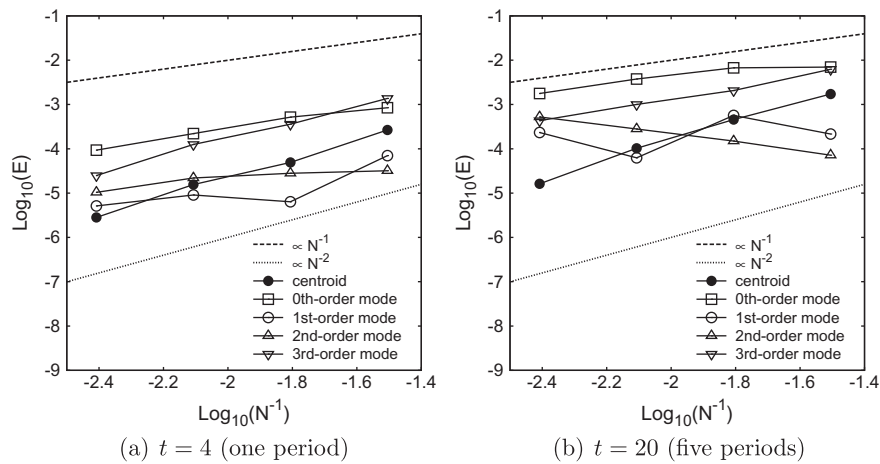
convergence with respect to  $N$ . On the other hand, the deformation modes under  $\beta = 1$  exhibit almost a second-order accuracy even at  $t = 20$ . Again, a smoothed transition of  $\phi$  gives a better geometric representation for curved interfaces.

#### 4.2.2. Deformed interface in a shearing flow

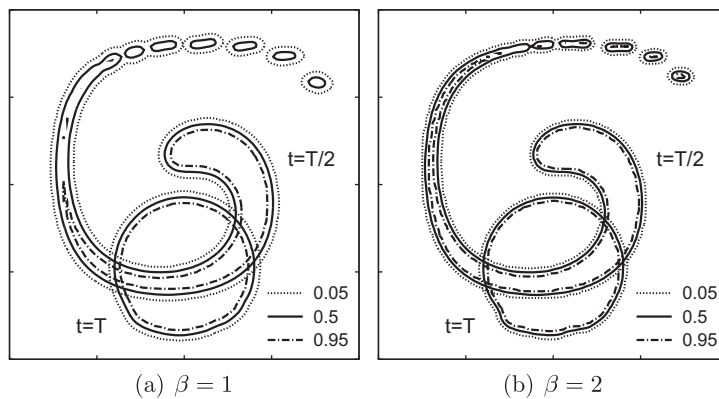
In order to test the present method in capturing a heavily deformed and stretched interface, we attempt the same benchmark test introduced in [40]. An initial circle with a radius of  $0.2\pi$  is located at  $(0.5\pi, 0.2(\pi + 1))$  on  $[0, \pi]^2$  plane. It is transported by a shearing velocity field defined by  $(u, v) = (\sin(x)\cos(y), -\cos(x)\sin(y))$ . The initial circle will be stretched into a spiral with a very thin tail which may be under the resolvable scale on a fixed Eulerian mesh for long term computation.



**Fig. 9.** The convergence rate for the centroid and the 0th, 1st, 2nd and 3rd-order modal amplitudes. The sharpness parameter  $\beta = 1$  is used.



**Fig. 10.** Same as Fig. 9, but the sharpness parameter  $\beta = 2$  is used.



**Fig. 11.** Comparison of VOF profiles at  $t = T/2$  and  $t = T$  in a periodic shearing flow with  $T = 10\pi$ . The number of cells  $N = 100$  is used.

At a certain time  $t = T/2$  when the interface exhibit a good distortion, the velocity field is inverted. The deformed interface is supposed to resume its initial circular shape in another  $T/2$  (at time  $t = T$ ).



We fix the maximum CFL number to 0.25 and the period to  $T = 10\pi$ . When a mesh of  $N = 100$  on one side of the domain is used, the time increment is set to  $\Delta t = 0.0025\pi$ .

The numerical results with different  $\beta$  are shown in Fig. 11. Similar to the other VOF methods, the interface is largely distorted at  $t = T/2$ , and the thin tail under the mesh resolution is broken into small droplets. Nevertheless, the interface is transported back to the original position after one period ( $t = T$ ) with reasonable accuracy compared to the existing other VOF methods [40,61]. In Fig. 12, the interface shape at  $t = T$  is shown together with the initial interface (the exact solution). The result with  $\beta = 1$  is more circular than that with  $\beta = 2$ . The convergence rate shown in Table 4 manifests a second-order accuracy in space for both  $\beta = 1$  and 2, and the error level is comparable to those in [40,61]. Numerical experiments are also carried out on the mesh of  $N = 100$  for different periods varied in the following range:  $T = 1.25\pi, 2.5\pi, 5\pi$  and  $10\pi$ . In Table 5, we compare the numerical errors measured by (43) with those of the original THINC method [58], the THINC/WLIC method [61] and the Youngs PLIC method reported in [40]. It is found that the present results show a consistent tendency with the PLIC method in terms of the error growth as the period  $T$  (and, in turn, deformation) increases.

We also test the time-dependent single vortex flow introduced in [6,39]. The velocity field is given by the following time-dependent stream function:

$$\frac{1}{\pi} \sin^2(\pi x) \cos^2(\pi y) \cos\left(\frac{\pi t}{T}\right), \quad x, y \in [0, 1]. \quad (46)$$

A circle is initially located at  $(0.5, 0.75)$  with a radius of 0.15. The computation is run up to  $t = T = 2$ , and in the meantime, the maximum CFL number is set to 1. On  $N = 32, 64$  and 128 meshes, we evaluate the error in a form of  $E_\phi = \sum_{i,j} \Delta x_i \Delta y_j |\phi_{ij} - \phi_{ij}^0|$ , and the accuracy is compared with the existing VOF methods [39,43] in Table 6. In this test, the interface largely deforms and its tail becomes a thin film smaller than the mesh width. In this case, the sharper reconstruction with  $\beta = 2$  shows a better accuracy. The present results exhibit a nearly second-order accuracy, which is comparable to the existing VOF methods. Fig. 13 shows the initial interfacial shape (the exact solution) and the numerical results after one period on the meshes of  $N = 32$  and 128. Compared with the solution on the  $N = 32$  mesh, the recovery of the interface on the finer mesh ( $N = 128$ ) is very much improved, showing almost the initial profile, for both  $\beta = 1$  and 2.

#### 4.2.3. Rotation of a slotted-sphere

Two dimensional Zalesak's problem is extended to a three dimensional case. A slotted sphere (with the inner value of the indicator function 1) is set as:

$$((x - 0.5)^2 + (y - 0.75)^2 + (z - 0.5)^2) \leq 0.15^2 \wedge (|x - 0.5| \geq 0.025 \vee y \geq 0.725).$$

We impose the rotational velocity  $\mathbf{u} = (0.5 - y, x - 0.5, 0)$ , and set the maximum CFL number and sharpness parameter  $\beta$  to 0.5 and 2, respectively.

The numerical results after one revolution ( $t = 2\pi$ ) are shown in Figs. 14 and 15, respectively, for the meshes employing  $N = 100$  and 200. The solutions still keep the initial slot and the symmetric shape, which is probably due to the employment of the quadratic interface reconstruction. Table 7 shows the numerical errors measured by

$$E_\phi = \sum_{i,j,k} \Delta x_i \Delta y_j \Delta z_k |\phi_{ijk}^n - \phi_{ijk}^0|. \quad (47)$$

The numerical solution converges with a first-order accuracy in space.

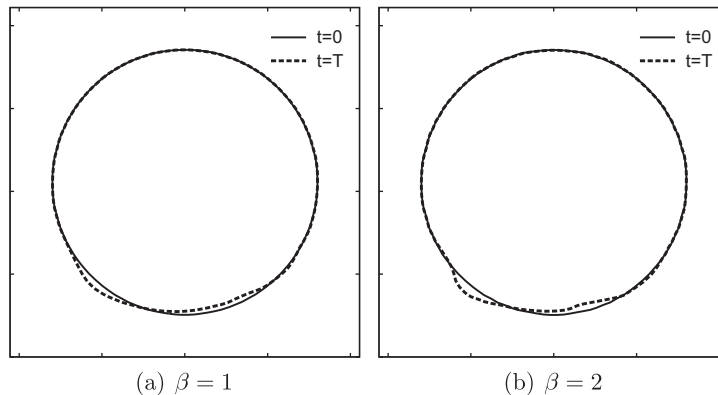


Fig. 12. Comparison of the interface after one full inversion of a circle in a time-dependent single vortex flow on  $N = 100$  mesh.

**Table 4**

Numerical errors and the respective convergence rates for the time-dependent single vortex flow with the period of  $T = 10\pi$ .

$N$	$(\beta = 1)$		$(\beta = 2)$	
	Error	Order	Error	Order
50	8.81E–2	–	5.54E–2	–
100	1.76E–2	2.32	1.81E–2	1.61
200	4.99E–3	1.82	5.47E–3	1.73

**Table 5**

Numerical errors in a periodic shearing flow for different periods  $T = 1.25\pi, 2.5\pi, 5\pi, 10\pi$ . Tabulated are the results with  $\beta = 1$  and 2, as well as those from the original THINC method [58], the THINC/WLIC method [61] and the Youngs PLIC method reported in [40]. The number of cells and the CFL number are fixed to  $N = 100$  and 0.25, respectively.

$T$	1.25 $\pi$	2.5 $\pi$	5 $\pi$	10 $\pi$
MTHINC ( $\beta = 1$ )	2.35E–3	4.34E–3	7.02E–3	1.76E–2
MTHINC ( $\beta = 2$ )	2.26E–3	4.25E–3	8.28E–3	1.81E–2
THINC [58]	3.21E–2	3.53E–2	3.84E–2	6.81E–2
THINC/WLIC [61]	1.11E–2	1.21E–2	1.62E–2	4.02E–2
Youngs-PLIC [40]	2.61E–3	5.12E–3	8.60E–3	3.85E–2

**Table 6**

Numerical errors and the respective convergence rates in the time-dependent single vortex flow with a period  $T = 2$ . The errors of the present method with  $\beta = 1, 2$  and the PLIC method [39,43] are shown.

$N$	$(\beta = 1)$		$(\beta = 2)$		[39]		[43]	
	Error	Order	Error	Order	Error	Order	Error	Order
32	9.07E–3	–	3.70E–3	–	2.36E–3	–	1.88E–3	–
64	2.80E–3	1.70	1.00E–3	1.89	5.85E–4	2.01	4.42E–4	2.08
128	7.17E–4	1.97	2.97E–4	1.75	1.31E–4	2.16	9.36E–5	2.24

#### 4.2.4. A sphere in a time-dependent vortex flow

According to [25], a sphere (radius 0.15) initially centered at (0.35, 0.35, 0.35) is stretched by a three dimensional velocity field:

$$\begin{cases} u(x, y, z) = 2 \sin^2(\pi x) \sin(2\pi y) \sin(2\pi z) \cos(\pi t/T), \\ v(x, y, z) = -\sin(2\pi x) \sin^2(\pi y) \sin(2\pi z) \cos(\pi t/T), \\ w(x, y, z) = -\sin(2\pi x) \sin(2\pi y) \sin^2(\pi z) \cos(\pi t/T), \end{cases} \quad (48)$$

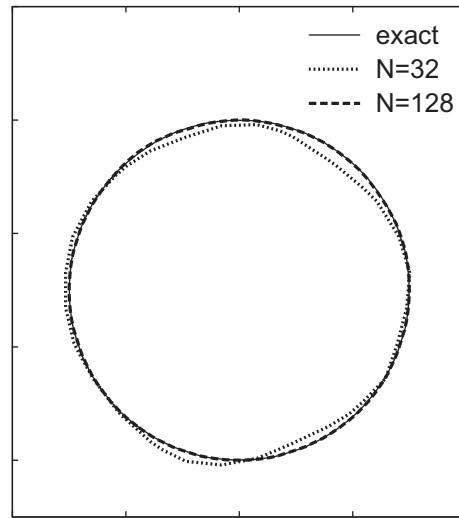
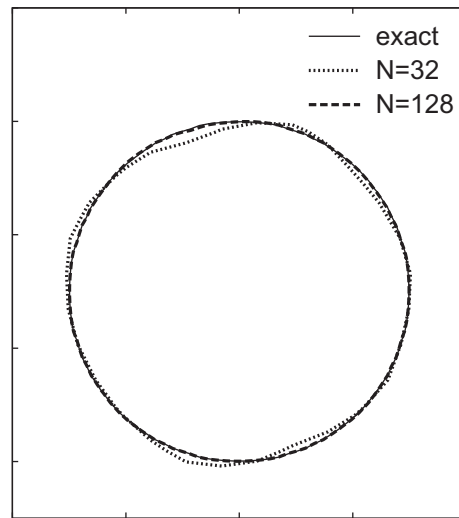
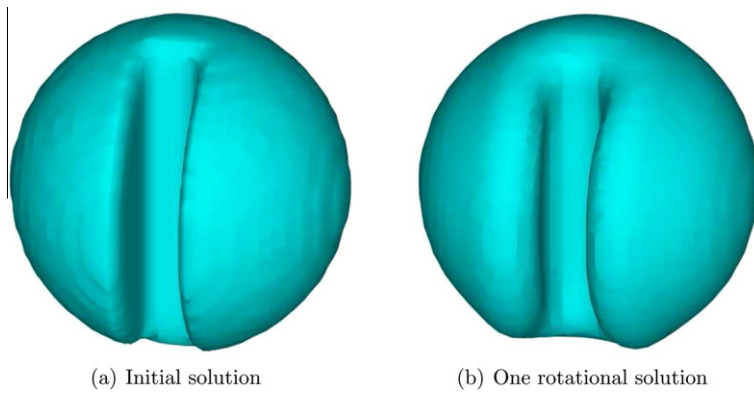
$x, y, z \in [0, 1]$ .

The period  $T$  is set to 3, and the maximum CFL number is set to 0.2.

Fig. 16 shows the results with the sharpness parameter  $\beta = 2$  on  $N = 100$  and 200 meshes. For the low resolution case, the thin interface of the most deformed structure at  $t = T/2$  are not adequately captured. The increase in the mesh resolution improves the interface quality of the solution, and the initial spherical shape is well reproduced after one period of computation. However, there still exist a few droplets left out to the big spherical body at the final moment ( $t = T$ ) even on the mesh of  $N = 200$ . A computation with a larger sharpness parameter  $\beta = 3.5$  is attempted for comparison, and the result is shown in Fig. 17. The iso-contours at  $t = T/2$  suggest an improved capability of capturing the thin interface by employing a larger sharpness parameter. Consequently, the final solution at  $t = T$  is much improved without broken-droplet left out of the main droplet. The corresponding numerical errors are summarized in Table 8 where we observe a second-order accuracy. Those results indicate that a larger  $\beta$  is desirable for computing an interface with thin structures.

#### 4.3. Multi-fluid dynamics with moving interface

All the runs presented in this subsection are carried out under a convergence criterion of (36) with  $L_2$  residual of  $10^{-5}$  (if not mentioned otherwise), and our parametric study shows that the numerical results are insensitive to the above threshold, as will be discussed later.

(a)  $\beta = 1$ (b)  $\beta = 2$ **Fig. 13.** The interfacial profiles on  $N = 32$  and  $N = 128$  meshes in the single vortex flow test.

(a) Initial solution

(b) One rotational solution

**Fig. 14.** The VOF profiles visualized by the iso-contour of 0.5 in the rotation test of a slotted-sphere on the  $N = 100$  mesh.

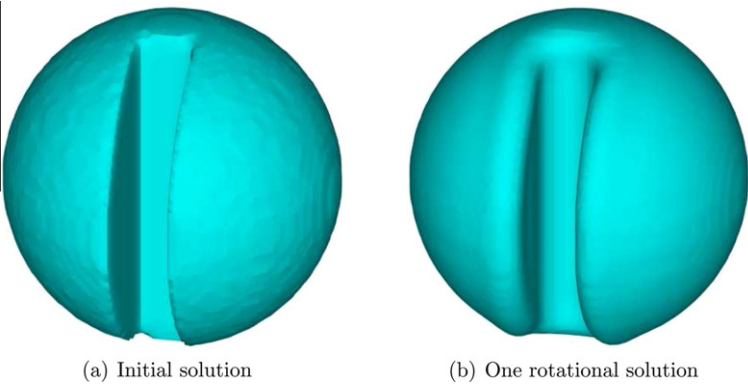


Fig. 15. Same as Fig. 14, but the number of cells  $N = 200$  is used.

**Table 7**  
Numerical errors and the respective convergence rates in the 3D revolution of a slotted-sphere.

$N$	50	100	200
Error	1.19E-3	5.60E-4	2.97E-4
Order	–	1.09	0.91

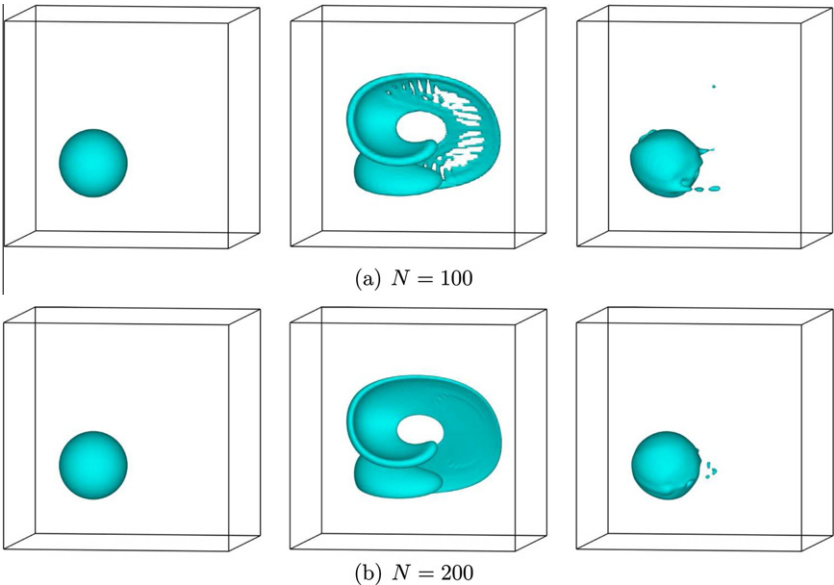


Fig. 16. Isosurfaces in a 3D time-dependent vortex flow (with a period  $T = 3$ ) at  $t = 0$  (left),  $t = T/2$  (middle) and  $t = T$  (right). The sharpness parameter  $\beta = 2$  is used.

4.3.1. A circular droplet in static field

In a static field involving multiple immiscible fluids under no effect of gravity, the whole field should remain still with a certain pressure jump across the interface balancing with the surface tension. However, at a discretized level, keeping the balance between the surface tension and pressure jump is not a trivial task, and, as a consequence, spurious current arises. As widely discussed in the literature, some VOF methods with PLIC reconstructions usually generate “parasite current” or “spurious current” and do not converge to steady state [35,52].

A 2D circular droplet with surface tension in a stationary fluid without gravity is considered. The droplet with a radius of  $R = 0.25$  is centered at  $(0.5, 0.5)$  in a domain  $[0, 1]^2$ . The fluids inside and outside of the droplet have the same density 4 and viscosity 1 according to [11]. The surface tension coefficient  $\sigma$  is set to 0.357. We run a computation with periodic boundary condition until the solution reaches steady state. We employ  $N = 32, 64, 128$  meshes, and the mesh-dependent time

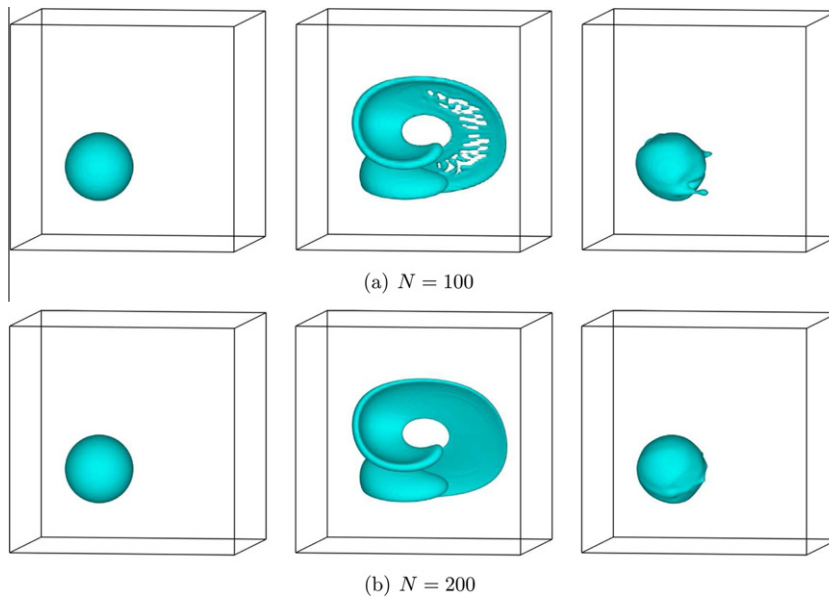


Fig. 17. Same as Fig. 16, but  $\beta = 3.5$  is used.

Table 8

Numerical errors and the respective convergence rates in the 3D time-dependent vortex flow with a period  $T = 3$ .

$N$	$(\beta = 2)$		$(\beta = 3.5)$	
	Error	Order	Error	Order
50	5.59E-3	–	4.99E-3	–
100	1.74E-3	1.68	1.60E-3	1.64
200	3.70E-4	2.23	3.81E-4	2.07

increment is set to be  $\Delta t = 0.0001 \times 32/N$ . The sharpness parameter is set to  $\beta = 1$ . The deformation modes given by (44) are investigated to see if the numerical solution converges to steady state. Fig. 18 shows the time history for the modal amplitudes of  $n = 0, 1, 2$  and 3. The four modes converge to respective constant values within a short range of time. In this case, although the numerical spurious current actually arises, it does not affect the interfacial shape significantly.

We also measure the maximum norms of the errors in velocity and pressure jump,

$$L_{\infty}(\mathbf{u}) = \max_{ij}(\max(|u_{ij}|), \max(|v_{ij}|)),$$

$$E(\Delta p) = \left| \frac{\Delta p^n - \Delta p^e}{\Delta p^e} \right|, \quad \Delta p^n = \frac{\sum_{ij} \phi_{ij}^n p_{ij}^n}{\sum_{ij} \phi_{ij}^n} - \frac{\sum_{ij} (1 - \phi_{ij}^n) p_{ij}^n}{\sum_{ij} (1 - \phi_{ij}^n)}, \quad \Delta p^e = \frac{\sigma}{R}, \quad (49)$$

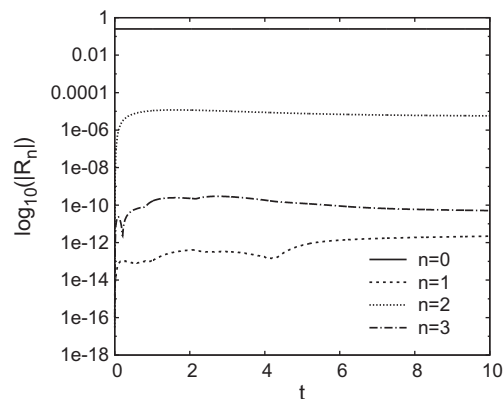


Fig. 18. The time history of the modal amplitudes for  $n = 0, 1, 2$  and 3 in the test of a circular drop in a stationary fluid.

where  $\Delta p^e$  and  $\Delta p^n$  are the exact and the numerical pressure jumps, respectively. The exact pressure jump is obtained from the Young–Laplace equation. In Fig. 19, the time histories for the numerical errors with doubly-increased meshes from  $N = 32$  to 128 are shown. All the errors uniformly decrease to respective small values over time, and the errors become smaller with the mesh refinement. The convergence rate plotted in Fig. 20 shows nearly a first-order accuracy. In this test, since both density and viscosity are the same for the fluids inside and outside of the droplet, we infer that the first-order accuracy comes from curvature estimation and the approximate delta function in the CSF model.

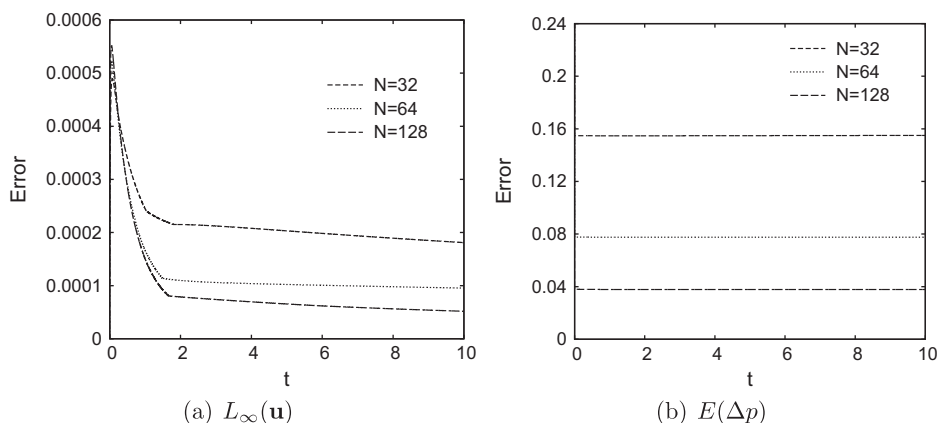
It should be noted that the “parasitic current” is caused by incorrect curvature estimation in the discretized governing equation. As further developments to reduce/eliminate the “parasitic current”, CLSVOF method [46], height function approaches [10,30] and balanced-force approach [11] have been devised.

#### 4.3.2. Drop oscillation with the surface tension force

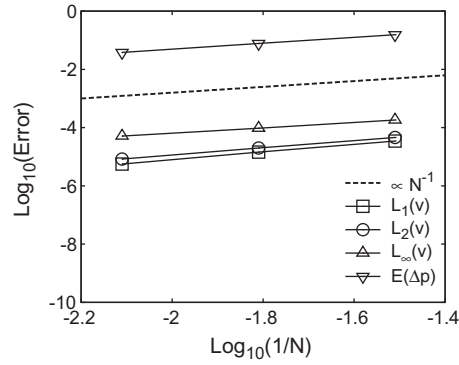
To see the dynamic interaction of two (immiscible) fluids of different flow properties, oscillating behavior of a drop with no gravity effect is studied. The drop, initially set to be elliptic, experiences axisymmetric oscillation due to the surface tension and it eventually attains a static circular shape under the contribution of surface tension and viscosity. The same computational conditions as [52] are employed, i.e. the drop density and viscosity are 1 and 0.01, and the surrounding gas is specified by density of 0.01 and viscosity of 0.00005, respectively. The surface tension coefficient  $\sigma$  is 1. The initial elliptic shape is given by  $(x - 10)^2/a^2 + (y - 10)^2/b^2 = 1$  ( $a = 3$ ,  $b = 2$ ) on a computational domain of  $[0, 20]^2$ . The number of cells  $N$  is varied in a range of 64, 128 and 256, and the time increment depends on the mesh and specified as  $\Delta t = 0.05 \times 64/N$ . The sharpness parameter  $\beta$  is set to 1.

The evolution of the kinetic energy, computed by  $\frac{1}{2} \sum_{i,j} \rho_{ij}^n |\mathbf{u}_{ij}^n|^2 \Delta x_i \Delta y_j$ , is plotted in Fig. 21. The oscillation pattern, such as the amplitude and frequency, converges as mesh resolution increases (Fig. 21(a)), and also the numerical solution converges uniformly to steady state. We can confirm that the present result agrees well with the result computed by an existing VOF method [11] and a front-tracking method [52]. Fig. 22 shows the instantaneous VOF distributions for different  $N$  at  $t = 500$  when steady state is attained. The continuous VOF function is specified with respect to the mesh size in the reconstruction procedure (see the approximate function (7)), therefore, the interface width looks spreading on  $N = 64$  mesh. As mentioned before, the interface width is readily to be controlled and kept within a few cells. Thus, the interface could become more compact with the mesh refinement. A snapshot of the pressure profile during the oscillation is shown in Fig. 23(a), and the steady distribution of pressure is shown in Fig. 23(b) on  $N = 256$  mesh. It is found that the pressure has a smooth transition region across the interface due to the continuous reconstruction. A higher pressure jump is observed near the ellipsoidal edges due to the large curvature and stronger surface tension during the oscillation, and eventually the pressure distribution converges to a flat plateau within the circle in steady state, where the analytical steady pressure jump is determined by the Laplace pressure  $\Delta p^e = \sigma/R$  with the geometric mean  $R = \sqrt{ab} = \sqrt{6}$ . Fig. 24 plots the pressure profiles for both analytical and numerical solutions on the axis of  $y = 10$ . We again confirm that the pressure jump is captured with several cells, and it converges gradually to the analytical solution as the number of cells increases. As summarized in Table 9, it is found again that the pressure jump error measured by (49) converges with a first-order accuracy in space.

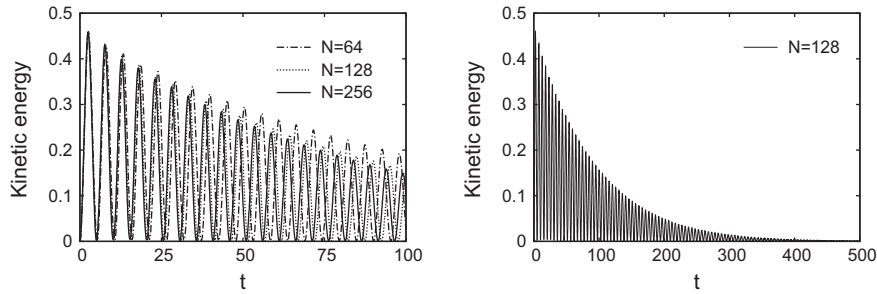
It should be noted that the present scheme focuses on mass conservation rather than volume conservation, and we found mass is exactly conserved for every case we tested. On the other hand, the way to determine the interface location from a distribution having a finite thickness for the jump transition remains an open problem with some arbitrariness. As discussed in [58], a reasonable choice is to take the  $\phi = 0.5$  surface as the interface, but this does not necessarily guarantee the conservation of the volume encompassed by the  $\phi = 0.5$  surface, even the total mass is rigorously conserved. Nevertheless, since the



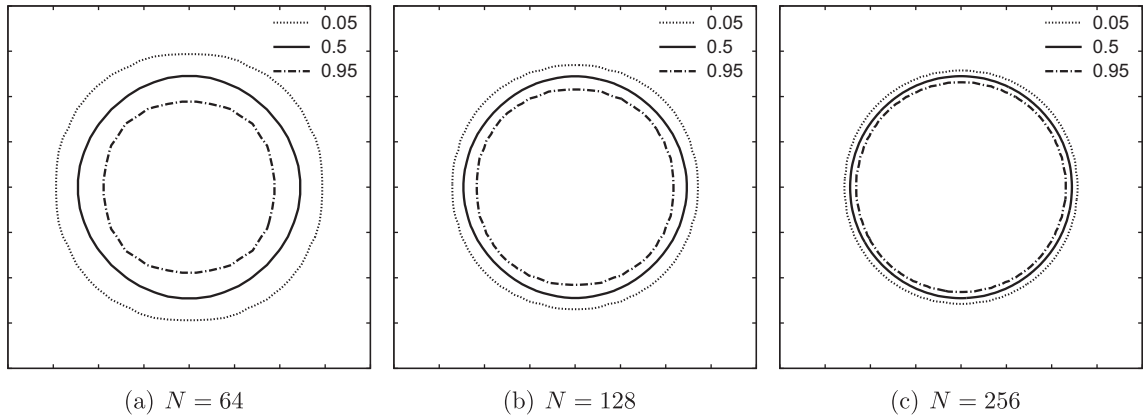
**Fig. 19.** The convergence processes of the errors in velocity and pressure represented by (49) (for the flows involving a circular drop) with time; suggesting effective suppress of parasitic current.



**Fig. 20.** The convergence rate for the velocity and the pressure jump in terms of the mesh refinement.



**Fig. 21.** The evolution of the total kinetic energy in the drop oscillation problem. The left figure shows the results on  $N = 64, 128$  and  $256$  meshes for  $0 \leq t \leq 100$ , and the right figure shows the convergence behavior on  $N = 128$  mesh for a long term computation.



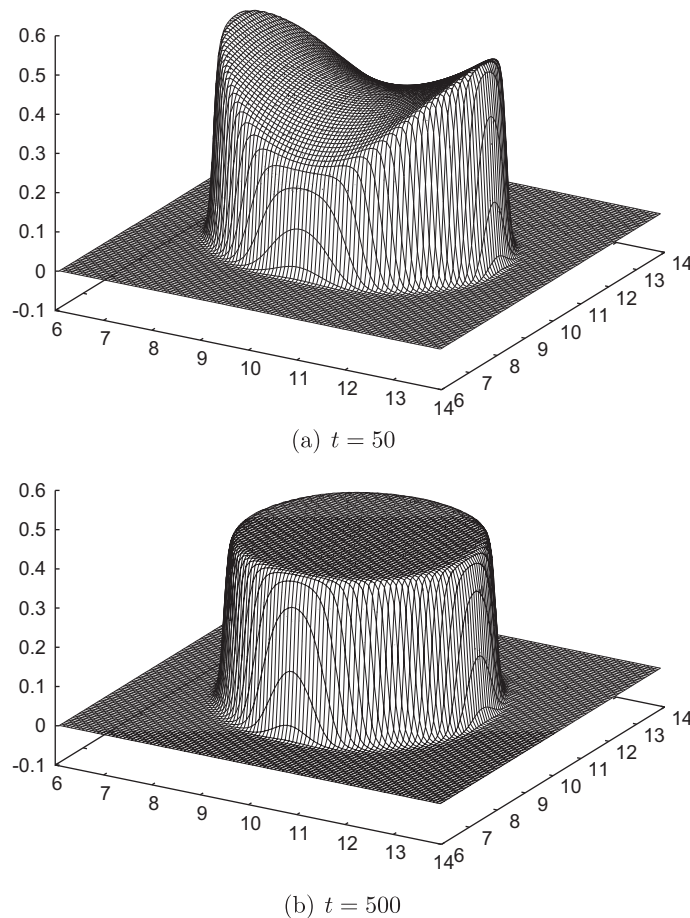
**Fig. 22.** The VOF profiles at steady state ( $t = 500$ ).

thickness of the interface transition zone remains compact in respect to the size of grid element, the computed volume converges when the grid is refined. This can be verified from the numerical tests presented in this paper.

#### 4.3.3. A deformable drop in an unbounded shear flow

In a computational domain  $x \in [-4, 4]$ ,  $y \in [-2, 2]$ ,  $z \in [-4, 4]$ , a unit spherical drop is located at the center without the effect of gravity. The opposite velocities  $\pm U$  that result in a shear rate  $\dot{\gamma} = 2U/H_z = U/4$  are imposed on the top and bottom walls. A schematic of the flow field is shown in Fig. 25. The periodic boundary condition is imposed in  $x$  and  $y$  directions. The same density and viscosity are specified for both sphere and surrounding fluid. The viscosity is given by Reynolds number  $Re = \rho \dot{\gamma} R^2 / \mu$ . In this test,  $\rho = 1$ ,  $\dot{\gamma} = 1$  and  $Re = 0.1$  are used. The capillary number  $Ca = \mu \dot{\gamma} R / \sigma$  is varied from 0.1 to 0.4 at the intervals of 0.1. Note that the drop becomes more flexible with higher  $Ca$  for a fixed ambient viscosity. The number of cells is  $(N_x, N_y, N_z) = (128, 64, 128)$  and the time increment is  $\Delta t = 10^{-4}$ . We use  $\beta = 2$  in this test.



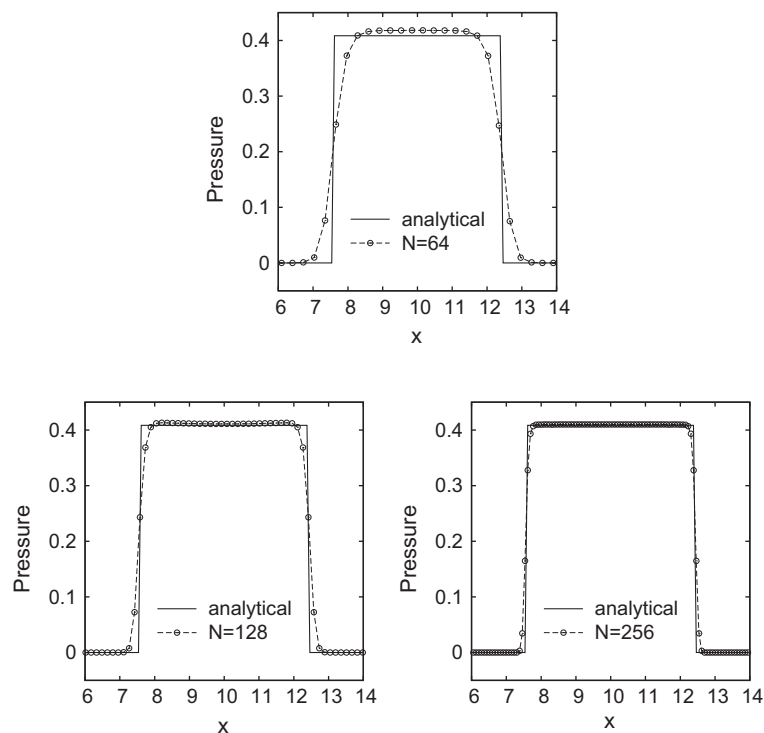


**Fig. 23.** The instantaneous pressure profiles on  $N = 256$  mesh.

The isosurfaces of the VOF function at steady state are illustrated in Fig. 26 together with the velocity vectors, and the interfaces cutting through a cross-section  $y = 0$  are overlaid in Fig. 27. The interface becomes elliptic in steady state, and its deformation becomes larger as the capillary number increases. A theoretical solution derived on the assumptions of the Stokes flow and small deformation [49] shows that the drop is distorted into an ellipse and the deformation parameter given by  $D = (l - s)/(l + s)$  is linearly changed with regard to the capillary number. Here,  $l$  and  $s$  denote the semi-major and semi-minor axes of the ellipse, respectively. In order to retrieve the computed ellipse, we sampled the coordinates of the interfacial points from the 0.5 contour line of the VOF function. Then,  $l$  and  $s$  are identified by fitting the interfacial points to an ellipse by a least-square approach. The correlation between the capillary number and the deformation parameter is plotted in Fig. 28. The shape in steady state agrees well with the theoretical expression [49] in the range of small capillary number. In the range of large capillary number, on the other hand, the theory underestimates the deformation parameter due to the geometric nonlinearity neglected therein, as is well known in [37].

#### 4.3.4. A rising bubble in a viscous fluid

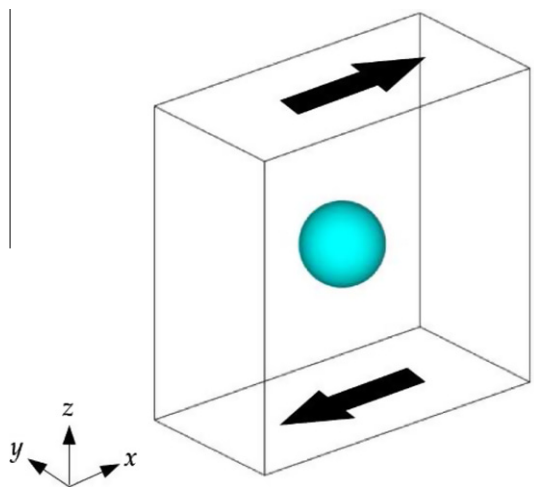
Finally, we present the numerical results for a single bubble rising in a viscous fluid. Hnat and Buckmaster [18] experimentally studied the detailed behaviors of rising bubbles in various cases. Following one of the cases in their experiments, the physical parameters are given as  $\rho = 1$  and  $\mu = 0.001$  for gas inside the bubble,  $\rho = 875.5$  and  $\mu = 0.118$  for the outer liquid, and  $\sigma = 0.0322$  and  $g = 9.8$ . Under this condition, the terminal rise velocity  $V_\infty$  of the bubble was found to be 0.215, and the corresponding bubble Reynolds and Weber numbers based on the terminal velocity are estimated to be 9.8, 0.76 and 7.6, respectively [18]. To get rid of the boundary effect of the computational domain, we use a sufficiently large domain  $[0, 6D] \times [0, 6D] \times [0, 12D]$  compared with the bubble size, and the slip boundary conditions are imposed on the surrounding boundaries. A spherical bubble with a radius of  $R = D/2 = 0.00608$  is initially located at  $(3D, 3D, 2.5D)$ . In the above, all the properties are indicated in SI units. The bubble deforms and rises with time, and the shape is deformed into a spherical cap shape and it reaches steady state.



**Fig. 24.** The pressure profiles at the cross-section of  $y = 10$  for steady state ( $t = 500$ ) with the number of cells  $N = 64, 128$  and  $256$ . The solid line indicates the exact Laplace pressure.

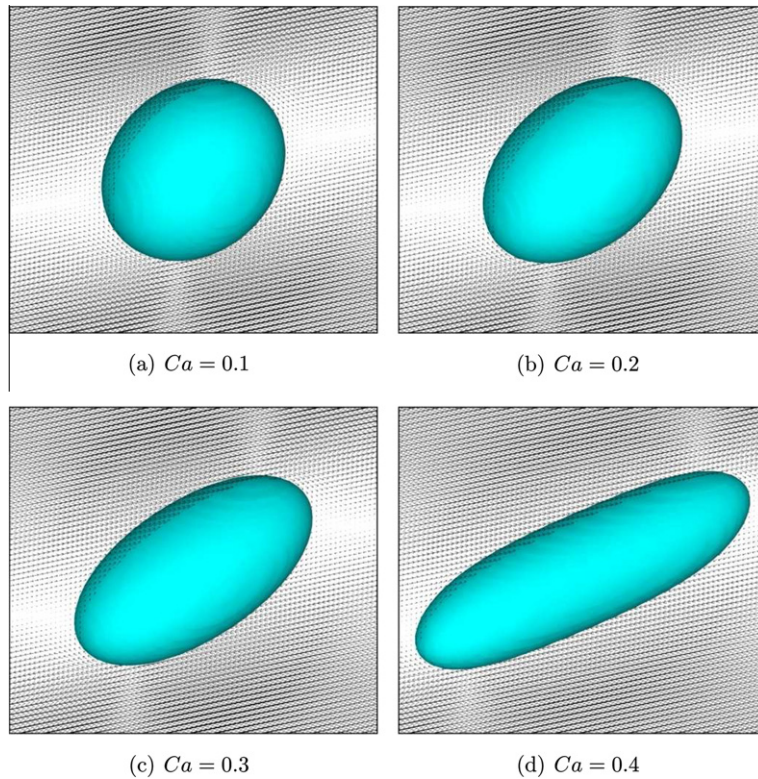
**Table 9**  
Numerical errors and the respective convergence rates for the pressure jump in steady state.

$N$	64	128	256
Error	$1.38\text{E}-1$	$7.03\text{E}-2$	$3.40\text{E}-2$
Order	–	0.97	1.05

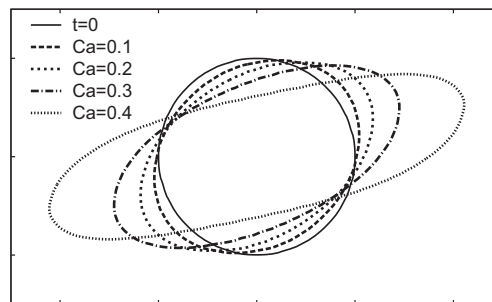


**Fig. 25.** An initial shape of the deformed drop in a shear flow.

At the first setout, we check the convergence behavior of the bubble shape and streamline (based on the bubble rise velocity) in a cross-section of  $y = 3D$ , with respect to the spatial resolutions of  $\Delta x = D/10, D/20$  and  $D/30$ . The corresponding time increments are set to  $\Delta t = 0.0008, 0.0004$  and  $0.0002$ , respectively. The numerical results at non-dimensional time



**Fig. 26.** The fully-developed deformed surfaces subjected to a linear shear flow of different capillary numbers  $Ca$ . The velocity vectors are also shown in an  $x$ - $z$  cross-section passing through  $y = 0$ .

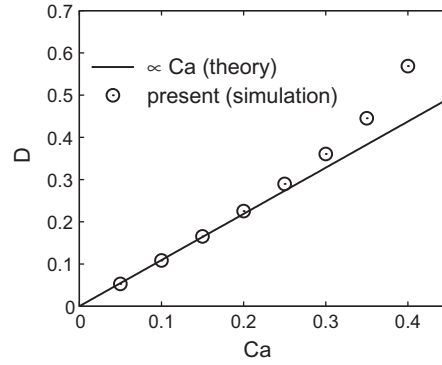


**Fig. 27.** The interfaces on a cross-section ( $x$ - $z$ ) for various capillary numbers  $Ca$ .

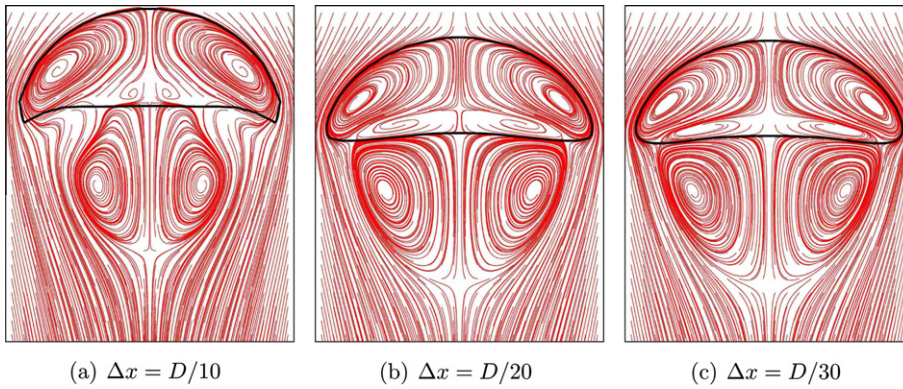
$t^* = tV_\infty/R = 15.6$  are shown in Fig. 29 for the different spatial resolutions. The results are found that the bubble turns into a spherical cap shape and wake structure behind the bubble converges as the mesh resolution increases. Moreover, an axisymmetric flow is well reproduced in our 3D code. From this first trial, we use the mesh size  $\Delta x = D/30$  hereafter.

Next, we investigate the bubble rise velocity given as  $V_{\text{rise}}(t) = (z_c(t) - z_c(t - \Delta t))/\Delta t$ , where  $z_c$  represents the centroid of the bubble obtained from (45). Time history of bubble rise velocity is plotted in Fig. 30, and it is found that the bubble rise velocity attains a constant when all the exerting forces on the bubble are well balanced. The present result converges to the experimental data [18] over time, which is also plotted in Fig. 30.

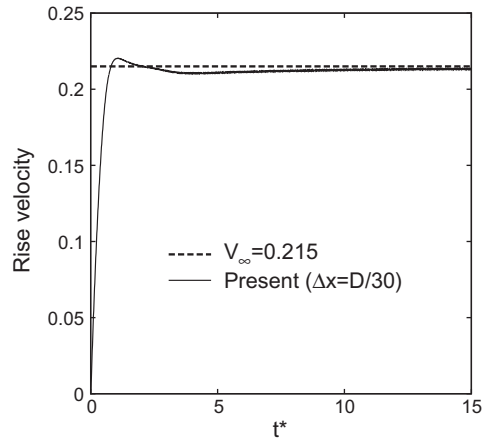
In Fig. 31, we compare the bubble shapes and wake structures of the present result with the experiment [18] and other well-validated simulation results including the body-fitted approaches [41,50], unstructured ALE approach [67], VOF approach [13] and level-set approach [47]. It should be mentioned that the original printed literature [18] with the clear experimental photograph was not available, thus we referred to [67]. The bubble shape and wake structure based on the present method reproduces the experimental result well. Other simulation results slightly vary due to the different numerical methodology, nevertheless, the present result also shows good agreement with other numerical results.



**Fig. 28.** Correlation between capillary number  $Ca$  and Taylor deformation parameter  $D$ . The solid line is given by a theoretical analysis by Taylor [49] on the assumptions of small deformation and Stokes flow.



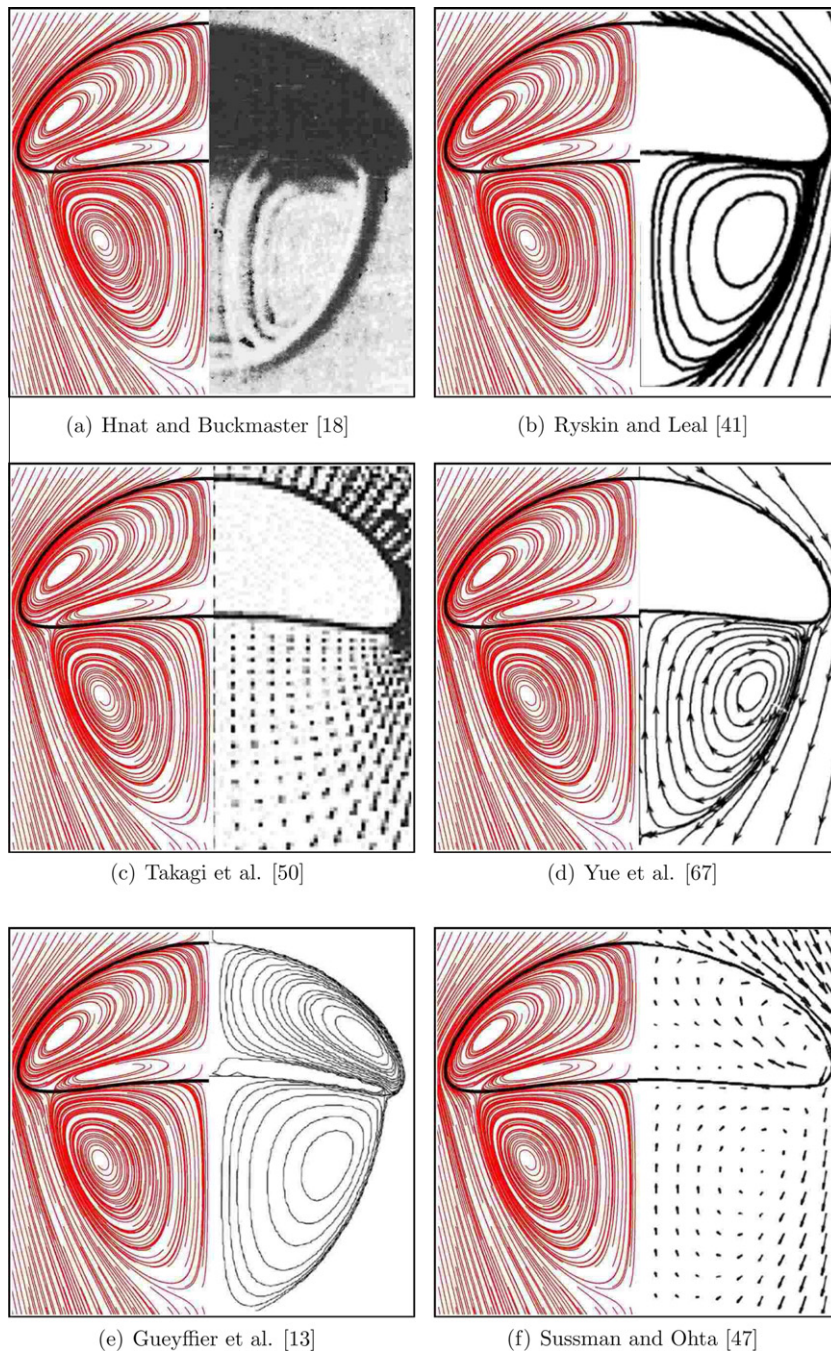
**Fig. 29.** The bubble shapes and streamlines based on the bubble rise velocities at non-dimensional time  $t^* = tV_\infty/R = 15.6$  with different mesh resolutions.



**Fig. 30.** The bubble rise velocity  $V_{\text{rise}}$  (solid line) plotted against non-dimensional time  $t^*$  with the mesh resolution of  $\Delta x = D/30$ . The reference solution  $V_\infty = 0.215$  (dashed line) indicates the terminal velocity experimentally obtained by Hnat and Buckmaster [18].

To check the influence of the convergence criteria specified for the iteration solution of (36) on the numerical results, we employ three different convergence criteria, i.e.,  $E_c = 10^{-4}$ ,  $10^{-5}$  and  $10^{-7}$ , which results in the maximum divergence errors of approximately  $10^{-3}$ ,  $10^{-4}$  and  $10^{-6}$ , respectively. Regardless of the different criteria, it is observed that the bubble shapes are not visibly distinguished, and the bubble rise velocities are also found to be insensitive to the convergence criteria tested. The deviations of the terminal rise velocities with  $E_c = 10^{-4}$  and  $10^{-5}$  from that with  $E_c = 10^{-7}$  are less than 0.001% and 0.0001%, respectively. It indicates to some extent that the present model is robust even for the problems with large density ratios.





**Fig. 31.** Comparisons of the bubble shape and streamline based on the bubble rise velocity. The left side and right side of each panel from (a) to (f) indicate the present result and experiment/other simulation results.

## 5. Concluding remarks

We have proposed a novel interface capturing method with a continuous function in the VOF framework. The multi-dimensional hyperbolic tangent function was used for reconstructing the approximate indicator function, and it was also used for obtaining the numerical fluxes, therefore, the singularity associated with the discontinuous Heaviside function in the conventional VOF methods has been removed. This approach enables reconstruction of interface by a linear (plane) surface as well as a quadratic (curved) surface with a negligible increase in the computational effort, which has been a formidable difficult task for the conventional VOF methods employing a strategy of geometric reconstruction. The numerical

results show that the present method maintains the smoothed VOF profile which does not suffer from the numerical diffusion nor smear-out over time.

The numerical results also show the applicability of the present method in solving two-phase flow problems with surface tension force. The parasitic current caused by the numerical errors in the curvature estimation has been effectively suppressed by the continuous VOF distribution with the curved surface reconstruction even without using the convolution technique in the CSF model. The 3D numerical tests have proved that the present method is versatile and potent, offering sufficiently high accuracy for practical applications.

The present method can be easily extended to unstructured meshes such as the triangular mesh or tetrahedral mesh without essential changes in its algorithm. Application to unstructured mesh is the subject of ongoing research and future paper of the present authors.

## Acknowledgments

The authors thank Prof. Huaxiong Huang, Prof. Stéphane Zaleski, Prof. Shigenobu Okazawa and Mr. Shigeo Noda for fruitful discussions. This research was supported by Research and Development of the Next-Generation Integrated Simulation of Living Matter, a part of the Development and Use of the Next-Generation Supercomputer Project of the Ministry of Education, Culture, Sports, Science and Technology (MEXT). Sugiyama was supported in part from MEXT (Grant-in-Aid for Young Scientist (B), No. 21760120), and Takeuchi gratefully acknowledges the financial support of MEXT (Grant-in-Aid for Young Scientists No. 23686030).

## Appendix A. Determination of $d$

### A.1. 2D case

The 2D approximate function is given by

$$\hat{H}(X, Y) = \frac{1}{2}(1 + \tanh(\beta(P(X, Y) + d))), \quad X, Y \in [0, 1]. \quad (\text{A.1})$$

We consider **Case 1**:  $|n_X| = \max(|n_X|, |n_Y|)$ ,

$$\begin{aligned} P(X, Y) &= a_{02}Y^2 + a_{10}X + a_{01}Y, \\ a_{10} &= n_X, \quad a_{01} = n_Y - \frac{1}{2}l_{YY}, \quad a_{02} = \frac{1}{2}l_{YY}. \end{aligned} \quad (\text{A.2})$$

From the constraint, we obtain a following equation:

$$\begin{aligned} I_N^{(Y)} : \left( I_A^{(X)} : \hat{H}(X, Y) \right) \\ &= \frac{1}{2} \left[ \frac{1}{2} \left( x + \frac{1}{\beta a_{10}} \ln \left( \cosh(\beta(P(X, r_p^-) + d)) \right) \right) \right]_0^1 + \frac{1}{2} \left[ \frac{1}{2} \left( x + \frac{1}{\beta a_{10}} \ln \left( \cosh(\beta(P(X, r_p^+) + d)) \right) \right) \right]_0^1 \\ &= \frac{1}{2} + \frac{1}{4\beta a_{10}} \ln \left( \frac{\cosh(\beta(P(1, r_p^-) + d)) \cosh(\beta(P(1, r_p^+) + d))}{\cosh(\beta(P(0, r_p^-) + d)) \cosh(\beta(P(0, r_p^+) + d))} \right) = \frac{1}{2} + \frac{1}{4\beta a_{10}} \ln \left( \frac{(AB^-D + 1)(AB^+D + 1)}{A(B^-D + 1)(B^+D + 1)} \right) = \phi, \end{aligned} \quad (\text{A.3})$$

where  $r_p^\pm$  are the Gaussian quadrature points and we use a relationship  $\cosh(a) = (\exp(a) + \exp(-a))/2$ . Finally, a quadratic equation for  $D = \exp(2\beta d)$  is obtained as

$$AB^-B^+(A - Q)D^2 + A(B^- + B^+)(1 - Q)D + 1 - AQ = 0, \quad (\text{A.4})$$

where,

$$\begin{cases} A = \exp(2\beta a_{10}), \\ B^\pm = \exp(2\beta(a_{02}(r_p^\pm)^2 + a_{01}r_p^\pm)), \\ Q = \exp(2\beta a_{10}(2\phi - 1)). \end{cases} \quad (\text{A.5})$$

Although there are two solutions  $D^\pm$  in (A.4),  $d$  can be uniquely determined by choosing the positive value of  $D^\pm$  due to a definition of  $d = \ln(D)/(2\beta)$ .

We now consider **Case 2**:  $|n_Y| = \max(|n_X|, |n_Y|)$ ,

$$\begin{aligned} P(X, Y) &= a_{20}X^2 + a_{10}X + a_{01}Y, \\ a_{10} &= n_X - \frac{1}{2}l_{XX}, \quad a_{01} = n_Y, \quad a_{20} = \frac{1}{2}l_{XX}. \end{aligned} \quad (\text{A.6})$$

The unknown  $d$  is determined from  $D$  by solving a similar quadratic equation (A.4) with the coefficients:

$$\begin{cases} A = \exp(2\beta a_{01}), \\ B^\pm = \exp(2\beta(a_{20}(r_p^\pm)^2 + a_{10}r_p^\pm)), \\ Q = \exp(2\beta a_{01}(2\phi - 1)), \end{cases} \quad (\text{A.7})$$

## A.2. 3D case

In a similar way, the unknown  $d$  is determined from the constraint of the VOF function. However, as compared with the 2D case, a quartic equation for  $D$  must be solved.

Case 1:  $P(\mathbf{X}) = a_{020}Y^2 + a_{002}Z^2 + a_{011}YZ + a_{100}X + a_{010}Y + a_{001}Z$ .

$$\begin{aligned} I_N^{(Z)} : (I_A^{(Y)} : (I_A^{(X)} : \hat{H}(X, Y, Z))) \\ = \frac{1}{4} \left[ \frac{1}{2} \left( x + \frac{1}{\beta a_{100}} \ln \left( \cosh(\beta(P(X, r_p^-, r_p^-) + d)) \right) \right) \right]_0^1 + \frac{1}{4} \left[ \frac{1}{2} \left( x + \frac{1}{\beta a_{100}} \ln \left( \cosh(\beta(P(X, r_p^+, r_p^-) + d)) \right) \right) \right]_0^1 \\ + \frac{1}{4} \left[ \frac{1}{2} \left( x + \frac{1}{\beta a_{100}} \ln \left( \cosh(\beta(P(X, r_p^-, r_p^+) + d)) \right) \right) \right]_0^1 + \frac{1}{4} \left[ \frac{1}{2} \left( x + \frac{1}{\beta a_{100}} \ln \left( \cosh(\beta(P(X, r_p^+, r_p^+) + d)) \right) \right) \right]_0^1 \\ = \frac{1}{2} + \frac{1}{8\beta a_{100}} \ln \left( \frac{(AB^-D + 1)(AB^+D + 1)(AC^-D + 1)(AC^+D + 1)}{A^2(B^-D + 1)(B^+D + 1)(C^-D + 1)(C^+D + 1)} \right) = \phi. \end{aligned} \quad (\text{A.8})$$

A quartic equation for  $D = \exp(2\beta d)$  is

$$\begin{aligned} A^2 B^- B^+ C^- C^+ (A^2 - Q) D^4 + A^2 (B^- B^+ (C^- + C^+) + (B^- + B^+) C^- C^+) (A - Q) D^3 + A^2 ((B^- + B^+) (C^- + C^+) + B^- B^+ \\ + C^- C^+) (1 - Q) D^2 + A (B^- + B^+ + C^- + C^+) (1 - AQ) D + 1 - A^2 Q, \end{aligned} \quad (\text{A.9})$$

where,

$$\begin{cases} A = \exp(2\beta a_{100}), \\ B^\pm = \exp(2\beta(a_{020}(r_p^\pm)^2 + a_{002}(r_p^-)^2 + a_{011}r_p^\pm r_p^- + a_{010}r_p^\pm + a_{001}r_p^-)), \\ C^\pm = \exp(2\beta(a_{020}(r_p^-)^2 + a_{002}(r_p^\pm)^2 + a_{011}r_p^- r_p^\pm + a_{010}r_p^- + a_{001}r_p^\pm)), \\ Q = \exp(4\beta a_{100}(2\phi - 1)). \end{cases} \quad (\text{A.10})$$

Analogously, in **Case 2**, the quartic equation (A.9) is given with

$$\begin{cases} A = \exp(2\beta a_{010}), \\ B^\pm = \exp(2\beta(a_{200}(r_p^\pm)^2 + a_{002}(r_p^-)^2 + a_{101}r_p^\pm r_p^- + a_{100}r_p^\pm + a_{001}r_p^-)), \\ C^\pm = \exp(2\beta(a_{200}(r_p^-)^2 + a_{002}(r_p^\pm)^2 + a_{101}r_p^- r_p^\pm + a_{100}r_p^- + a_{001}r_p^\pm)), \\ Q = \exp(4\beta a_{010}(2\phi - 1)), \end{cases} \quad (\text{A.11})$$

and in **Case 3**, the quartic equation (A.9) is given with

$$\begin{cases} A = \exp(2\beta a_{001}), \\ B^\pm = \exp(2\beta(a_{200}(r_p^\pm)^2 + a_{020}(r_p^-)^2 + a_{110}r_p^\pm r_p^- + a_{100}r_p^\pm + a_{010}r_p^-)), \\ C^\pm = \exp(2\beta(a_{200}(r_p^-)^2 + a_{020}(r_p^\pm)^2 + a_{110}r_p^- r_p^\pm + a_{100}r_p^- + a_{010}r_p^\pm)), \\ Q = \exp(4\beta a_{001}(2\phi - 1)). \end{cases} \quad (\text{A.12})$$

## Appendix B. Analytical expression for a quartic equation in the present method

A quartic equation is defined as

$$a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0 = 0. \quad (\text{B.1})$$

We replace (B.1) by

$$x^4 + b_3 x^3 + b_2 x^2 + b_1 x + b_0 = 0, \quad (\text{B.2})$$



where  $b_3 = a_3/a_4$ ,  $b_2 = a_2/a_4$ ,  $b_1 = a_1/a_4$  and  $b_0 = a_0/a_4$ . (B.2) can be further expressed as the difference of two squared terms:

$$(x^2 + Ax + B)^2 - (Cx + D)^2 = 0. \quad (\text{B.3})$$

Here the following relationship is obtained from identity (B.2) and (B.3).

$$\begin{cases} 2A = b_3, \\ A^2 + 2B - C^2 = b_2, \\ 2AB - 2CD = b_1, \\ B^2 - D^2 = b_0. \end{cases} \quad (\text{B.4})$$

Then, by eliminating  $A$ ,  $C$  and  $D$ , and supposing  $z = 2B$ , a cubic equation for  $z$  is given by

$$z^3 + c_2 z^2 + c_1 z + c_0 = 0, \quad (\text{B.5})$$

$$\begin{cases} c_2 = -b_2, \\ c_1 = b_1 b_2 - 4b_0, \\ c_0 = b_0(4b_2 - b_3^2) - b_1^2. \end{cases} \quad (\text{B.6})$$

According to Cardano's formula, one real solution  $z_1$  to the cubic equation (B.5) is obtained as

$$z_1 = \begin{cases} \sqrt[3]{\frac{-b + \sqrt{b^2 + 4a^3}}{2}} - \sqrt[3]{\frac{b + \sqrt{b^2 + 4a^3}}{2}} - \frac{c_2}{3}, & \text{for } b^2 + 4a^3 \geq 0, \\ 2\sqrt{-b} \cos\left(\frac{1}{3} \tan^{-1}\left(\frac{\sqrt{-(b^2 + 4a^3)}}{-b}\right)\right) - \frac{c_2}{3}, & \text{for } b^2 + 4a^3 < 0, \end{cases} \quad (\text{B.7})$$

where,

$$a = -\frac{c_2^2}{9} + \frac{c_1}{3}, \quad b = \frac{2c_2^3}{27} - \frac{c_1 c_2}{3} + c_0. \quad (\text{B.8})$$

According to Brown's formula, when one real number solution  $z_1$  to solution (B.5) is known, the coefficients  $A$ ,  $B$ ,  $C$  and  $D$  in (B.4) are determined as follows:

$$\begin{cases} A = \frac{b_3}{2}, \\ B = \frac{z_1}{2}, \\ D = \sqrt{B^2 - b_0}, \\ C = \frac{-b_1/2 + AB}{D}. \end{cases} \quad (\text{B.9})$$

It remains possible that  $D$  becomes imaginary when  $B^2 - b_0$  becomes negative in (B.8). Fortunately for the present method, it never becomes negative from our numerical experiments. Therefore, the above relations hold. Finally, four solutions to (B.1) are obtained by solving the two quadratic equations from (B.3) as follows,

$$\begin{aligned} x^2 + (A - C)x + (B - D) &= 0, \\ x^2 + (A + C)x + (B + D) &= 0. \end{aligned} \quad (\text{B.10})$$

In practice, we choose the solution given by

$$x = \frac{-(A - C) + \sqrt{(A - C)^2 + 4(B - D)}}{2} \quad (\text{B.11})$$

from the physical requirements, and it is easily found that (B.10) truly works as the solution to (A.9) in all cases.

## References

- [1] E. Aulisa, S. Manservigi, R. Scardovelli, S. Zaleski, A geometrical area-preserving volume-of-fluid advection method, *J. Comput. Phys.* 192 (2003) 355–364.
- [2] E. Aulisa, S. Manservigi, R. Scardovelli, S. Zaleski, Interface reconstruction with least-squares fit and split advection in three-dimensional Cartesian geometry, *J. Comput. Phys.* 225 (2007) 2301–2319.
- [3] J.U. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modeling surface tension, *J. Comput. Phys.* 100 (1992) 335–354.
- [4] V.E. Badalassi, H.D. Cenicerros, S. Banerjee, Computation of multiphase systems with phase field models, *J. Comput. Phys.* 190 (2003) 371–397.
- [5] T. Belytschko, Fluid–structure interaction, *Comput. Struct.* 12 (1980) 459–469.
- [6] J.B. Bell, P. Colella, H.M. Glaz, A second-order projection method for the incompressible Navier–Stokes equations, *J. Comput. Phys.* 85 (1989) 257–283.
- [7] S.J. Cummins, M.M. Francois, D.B. Kothe, Estimating curvature from volume fractions, *Comput. Struct.* 83 (2005) 425–434.
- [8] H. Ding, P.D.M. Spelt, C. Shu, Diffuse interface model for incompressible two-phase flows with large density ratios, *J. Comput. Phys.* 226 (2007) 2078–2095.
- [9] S.V. Diwakar, S.K. Das, T. Sundararajan, A quadratic spline based interface (QUASI) reconstruction algorithm for accurate tracking of two-phase flows, *J. Comput. Phys.* 228 (2009) 9107–9130.

- [10] P.A. Ferdowsi, M. Bussmann, Second-order accurate normals from height functions, *J. Comput. Phys.* 227 (2008) 9293–9302.
- [11] M.M. Francois, S.J. Cummins, E.D. Dendy, D.B. Kothe, J.M. Sicilian, M.W. Williams, A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework, *J. Comput. Phys.* 213 (2006) 141–173.
- [12] J. Glimm, J.W. Grove, X.L. Li, K.M. Shyue, Y.N. Zeng, Q. Zhang, Three-dimensional front tracking, *SIAM J. Sci. Comput.* 19 (1998) 703–727.
- [13] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, S. Zaleski, Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows, *J. Comput. Phys.* 152 (1999) 423–456.
- [14] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, *Phys. Fluids* 8 (1965) 2182–2189.
- [15] D.J.E. Harvie, D.F. Fletcher, A new volume of fluid advection algorithm: the stream scheme, *J. Comput. Phys.* 162 (2000) 1–32.
- [16] C.W. Hirt, A.A. Amsden, J.L. Cook, An arbitrary Lagrangian–Eulerian computing method for all flow speeds, *J. Comput. Phys.* 14 (1974) 227–253.
- [17] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, *J. Comput. Phys.* 39 (1981) 201–225.
- [18] J.G. Hnat, J.D. Buckmaster, Spherical cap bubbles and skirt formation, *Phys. Fluids* 19 (1976) 182–194.
- [19] T.J.R. Hughes, W.K. Liu, T.K. Zimmermann, Lagrangian–Eulerian finite element formulation for incompressible viscous flows, *Comput. Methods Appl. Mech. Eng.* 29 (1981) 329–349.
- [20] T.J.R. Hughes, J.R. Stewart, A space-time formulation for multiscale phenomena, *J. Comput. Appl. Math.* 74 (1996) 217–229.
- [21] D. Jacqmin, Calculation of two-phase Navier–Stokes flows using phase-field modeling, *J. Comput. Phys.* 1455 (1999) 96–127.
- [22] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier–Stokes equations, *J. Comput. Phys.* 59 (1985) 308–323.
- [23] M.C. Lai, C.S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, *J. Comput. Phys.* 160 (2000) 705–719.
- [24] R.J. LeVeque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, *SIAM J. Numer. Anal.* 31 (1994) 1019–1044.
- [25] R. LeVeque, High-resolution conservative algorithms for advection in incompressible flow, *SIAM J. Numer. Anal.* 33 (1996) 627–665.
- [26] Z. Li, M.C. Lai, The immersed interface method for the Navier–Stokes equations with singular forces, *J. Comput. Phys.* 171 (2001) 822–842.
- [27] W.F. Noh, P. Woodward, SLIC (simple line interface method), *Lect. Notes Phys.* 24 (1976) 330–340.
- [28] M.R. Nobari, Y.-J. Jan, G. Tryggvason, Head-on collision of drops – a numerical investigation, *Phys. Fluids* 8 (1996) 29–42.
- [29] M.R.H. Nobari, G. Tryggvason, Numerical simulations of three-dimensional drop collisions, *AIAA J.* 34 (1996) 750–755.
- [30] J. López, J. Hernández, On reducing interface curvature computation errors in the height function technique, *J. Comput. Phys.* 229 (2010) 4855–4868.
- [31] S.J. Osher, J.A. Sethian, Fronts propagating with curvature dependent speed. Algorithms based on Hamilton–Jacobi formulations, *J. Comput. Phys.* 79 (1988) 12–49.
- [32] C.S. Peskin, Flow patterns around heart valves: a numerical method, *J. Comput. Phys.* 10 (1972) 252–271.
- [33] C.S. Peskin, The immersed boundary method, *Acta Numer.* 11 (2002) 479–517.
- [34] J.E. Pilliod Jr., E.G. Puckett, Second-order accurate volume-of-fluid algorithms for tracking material interfaces, *J. Comput. Phys.* 199 (2004) 465–502.
- [35] S. Popinet, S. Zaleski, A front-tracking algorithm for accurate representation of surface tension, *Int. J. Numer. Methods Fluids* 30 (1999) 775–793.
- [36] E.G. Puckett, A.S. Almgren, J.B. Bell, D.L. Marcus, W.J. Rider, A high-order projection method for tracking fluid interfaces in variable density incompressible flows, *J. Comput. Phys.* 130 (1997) 269–282.
- [37] J.M. Rallison, The deformation of small viscous drops and bubbles in shear flows, *Annu. Rev. Fluid Mech.* 16 (1984) 45–66.
- [38] Y. Renardy, M. Renardy, PROST: a parabolic reconstruction of surface tension for the volume-of-fluid method, *J. Comput. Phys.* 183 (2002) 400–421.
- [39] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, *J. Comput. Phys.* 141 (1998) 112–152.
- [40] M. Rudman, Volume-tracking methods for interfacial flow calculations, *Int. J. Numer. Methods Fluids* 24 (1997) 671–691.
- [41] G. Ryskin, L.G. Leal, Numerical solution of free-boundary problems in fluid mechanics. Part 2. Buoyancy-driven motion of a gas bubble through a quiescent liquid, *J. Fluid Mech.* 148 (1984) 19–35.
- [42] R. Scardovelli, S. Zaleski, Analytical relations connecting linear interfaces and volume fractions in rectangular grids, *J. Comput. Phys.* 164 (2000) 228–237.
- [43] R. Scardovelli, S. Zaleski, Interface reconstruction with least-square fit and split Eulerian Lagrangian advection, *Int. J. Numer. Methods Fluids* 41 (2003) 251–274.
- [44] G. Strang, On the construction and comparison of difference schemes, *SIAM J. Numer. Anal.* 5 (1968) 506–517.
- [45] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flows, *J. Comput. Phys.* 114 (1994) 146–159.
- [46] M. Sussman, E.G. Puckett, A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows, *J. Comput. Phys.* 162 (2000) 301–337.
- [47] M. Sussman, M. Ohta, A stable and efficient method for treating surface tension in incompressible two-phase flow, *SIAM J. Sci. Comput.* 31 (2009) 2447–2471.
- [48] K. Sugiyama, S. li, S. Takeuchi, S. Takagi, Y. Matsumoto, A full Eulerian finite difference approach for solving fluid–structure coupling problems, *J. Comput. Phys.* 230 (2011) 596–627.
- [49] G.I. Taylor, The deformation of emulsions in definable fields of flows, *Proc. R. Soc. Lond. A* 146 (1934) 501–523.
- [50] S. Takagi, Y. Matsumoto, H. Huang, Numerical analysis of a single rising bubble using boundary-fitted coordinate system, *JSME Int. J. Ser. B Fluids Therm. Eng.* 40 (1997) 42–50.
- [51] T.E. Tezduyar, M. Behr, J. Liou, A new strategy for finite element computations involving moving boundaries and interfaces – the deforming-spatial-domain/space-time procedure: I. The concept and the preliminary numerical tests, *Comput. Methods Appl. Mech. Eng.* 94 (1992) 339–351.
- [52] D.J. Torres, J.U. Brackbill, The point-set method: front-tracking without connectivity, *J. Comput. Phys.* 165 (2000) 620–644.
- [53] G. Tryggvason, B.B. Bunner, A. Esmaeili, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.J. Jan, A front-tracking method for the computations of multiphase flow, *J. Comput. Phys.* 169 (2001) 708–759.
- [54] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, *J. Comput. Phys.* 100 (1992) 25–37.
- [55] B. Van Leer, Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov’s method, *J. Comput. Phys.* 46 (1979) 101–136.
- [56] F. Xiao, Y. Honma, K. Kono, A simple algebraic interface capturing scheme using hyperbolic tangent function, *Int. J. Numer. Methods Fluids* 48 (2005) 1023–1040.
- [57] F. Xiao, A. Ikebata, An efficient method for capturing free boundary in multi-fluid simulations, *Int. J. Numer. Methods Fluids* 42 (2003) 187–210.
- [58] F. Xiao, S. li, C.G. Chen, Revisit to the THINC scheme: a simple algebraic VOF algorithm, *J. Comput. Phys.* 230 (2011) 7086–7092.
- [59] T. Yabe, F. Xiao, Description of complex and sharp interface during shock wave interaction with liquid drop, *J. Phys. Soc. Jpn.* 62 (1993) 2537–2540.
- [60] T. Yabe, F. Xiao, T. Utsumi, The constrained interpolation profile method for multiphase analysis, *J. Comput. Phys.* 169 (2001) 556–593.
- [61] K. Yokoi, Efficient implementation of THINC scheme: a simple and practical smoothed VOF algorithm, *J. Comput. Phys.* 226 (2007) 1985–2002.
- [62] K. Yokoi, A numerical method for free-surface flows and its application to droplet impact on a thin liquid layer, *J. Sci. Comput.* 35 (2008) 372–396.
- [63] K. Yokoi, D. Vadiello, J. Hinch, I. Hutchings, Numerical studies of the influence of the dynamic contact angle on a droplet impacting on a dry surface, *Phys. Fluids* 21 (2010) 072102.
- [64] Y. Yonemoto, T. Kunugi, Multi-scale modeling of the gas–liquid interface based on mathematical and thermodynamic approaches, *Open Transport Phenom. J.* 2 (2010) 69–79.
- [65] D.L. Youngs, Time-dependent multi-material flow with large fluid distortion, in: K.W. Morton, M.J. Baines (Eds.), *Numerical Methods for Fluid Dynamics*, Academic, New York, 1982, pp. 273–285.
- [66] D.L. Youngs, An Interface Tracking Method for a 3D Eulerian Hydrodynamics Code, Technical Report 44/92/35, AWRE, 1984.
- [67] P. Yue, J.J. Feng, C.A. Bertelo, H.H. Hu, An arbitrary Lagrangian–Eulerian method for simulating bubble growth in polymer foaming, *J. Comput. Phys.* 226 (2007) 2229–2249.
- [68] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, *J. Comput. Phys.* 31 (1979) 335–362.