J. Indian Inst. Sci.
A Multidisciplinary Reviews Journal
ISSN: 0970-4140 Coden-JIISAD

REVIEW ARTICLE

Volume of Fluid Method: A Brief Review

*Ananthan Mohan and Gaurav Tomar**

Abstract | Understanding and predicting multiphase flows is of great relevance due to the ubiquitous nature of such flows in both nature and in many industrial applications. Rapid development of high speed computers and problem-specific algorithms in the last 2 decades has enabled the study of multiphase flows through numerical simulations. In this paper, we give a brief overview of different methods used in direct numerical simulations of two-phase flows. In particular, we focus on the volume of fluid (VOF) method used for locating and advecting the interface. VOF method is a mesh based interface capturing method in which a scalar function called void fraction field (which is the ratio of tracked fluid to the cell volume) is advected in order to track the interface position. A geometric VOF algorithm is detailed in this work. which strikes a balance between accuracy, ease of implementation and volume conservation on a structured grid. Another challenge in two-phase flow simulations is the inclusion of surface tension forces accurately. Here, we give a brief overview of Eulerian surface tension models and detail an approach balancing computational cost, curvature estimation and imposed timestep restriction. Finally, we discuss the most recent advances in VOF methods and outline the various numerical challenges we expect to encounter.

Keywords: *Multiphase flows, VOF, Simulations*

1 Introduction

Multiphase flows are ubiquitous in nature and in many industrial processes. Examples include but are not limited to bubbly flows[2,3], spray atomization[4], breaking waves[5,6], ink-jet printing[7] and rain formation[8]. Due to rapid development in high-performance computing and problem-specific algorithms, at present one can computationally study realistic two-phase flows[9]. Figure 1 gives an overview of different modeling approaches in numerical simulation of two-phase flows and popular methods in use in each approach.

Over the past 3 decades, several approaches have been developed for the numerical simulation of multiphase flows. Methods such as smooth particle hydrodynamics, two-fluid formulation, and one-fluid formulation are based on solving Navier–Stokes equations while lattice Boltzmann methods is based on solving mesoscopic kinetic equations on a lattice[10]. In two-fluid formulation in order to computationally simulate two-phase flows, we partition the domain into individual subdomains filled with distinct fluids. By solving Navier–Stokes equations in each subdomain and coupling the solutions at the interface using interface jump conditions, one can simulate the entire domain. The interface jump conditions are obtained using mass and momentum conservation principles. This approach was found to be extremely limiting for realistic flow simulations involving large deformations and topology changes[9]. A detailed review of two-fluid model is given by Ishii and Hibiki[11]. Lattice Boltzmann method[12] and smoothed particle hydrodynamics[13,14] are promising and rapidly

[1] *Department of Mechanical Engineering, Indian Institute of Science, Bangalore, India.*
**gtom@iisc.ac.in*

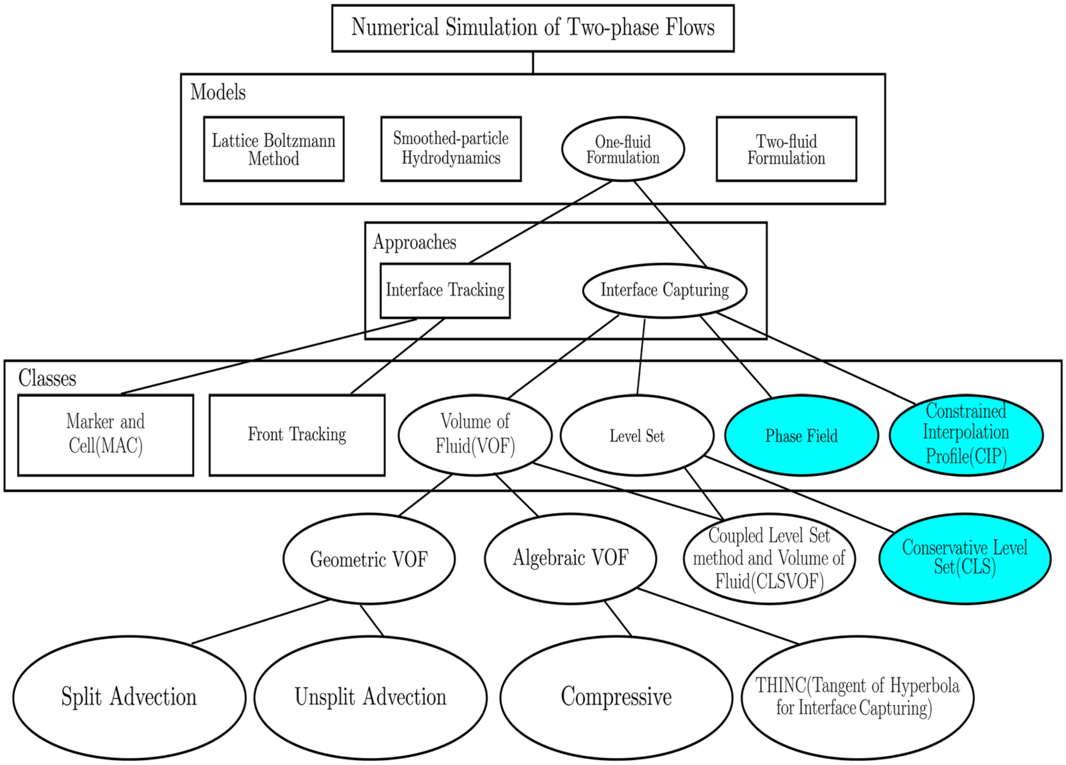


Figure 1: Modeling approaches for numerical simulation of two-phase flows[1] and popular methods used in each approach. Interface capturing methods are represented as ellipses and those with colored background use a diffused interface approach while white background implies a sharp interface method.

advancing modeling approaches for two-phase flow simulations.

Numerical methods for tracking interfaces in one-fluid approach can be broadly classified into interface tracking and interface capturing methods. The most popular methods in interface tracking approach are Front tracking[15,16] and Marker and Cell (MAC) methods. A detailed review of MAC method is given by Mangiavacchi et al.[17]. Popular interface capturing methods include volume of fluid (VOF), level-set approach, conservative level-set (CLS), phase field methods and constrained interpolation profile (CIP). An excellent review of these methods is given by Zaleski et al.[18]. A recent review of the state-of-the-art in one-fluid interface capturing methods is given by Dodd et al.[1]. A recent review of level-set methods is given by Osher et al.[19]. In this article, we discuss in detail various implementation aspects of only one class of interface capturing method, namely volume of fluid (VOF) method.

This review is organized as follows. Section 2 gives a brief overview on one-fluid formulation and discuss volume of fluid method in general. Section 3 gives a brief outline of the development of VOF methods over the years and explains in detail a few important landmarks. Section 4 explains in detail the first step in VOF method, i.e., initialization of void fraction. Subsequently, Sects. 5 and 6 detail the critical steps in VOF, i.e., reconstruction of the interface and advection of void fraction, respectively. Section 7 gives a brief introduction into the numerical approximation of surface tension forces in one-fluid approach. Finally, Sect. 8 gives a brief overview of recent advances in VOF method and outlines the challenges that need to be addressed in the near future.

2 One-Fluid Formulation

Consider a two-phase, incompressible fluid flow with a sharp (immiscible) interface between the fluid phases as shown in Fig. 2. Instead of solving the governing equations for each phase separately, one can use one set of governing equations for the whole domain with phase properties abruptly changing across the interface. Instead of using the jump conditions at the interface to conserve mass and momentum, a singular forcing term (Dirac-δ

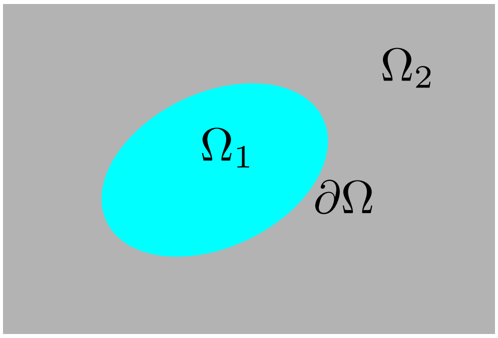
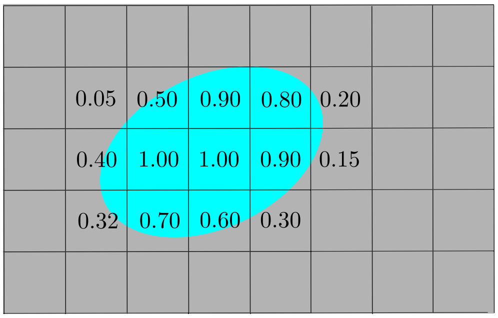


Figure 2: Schematic showing a two-phase fluid flow domain with phases Ω_1 and Ω_2 separated by a sharp interface $\partial\Omega$.



	0.05	0.50	0.90	0.80	0.20		
	0.40	1.00	1.00	0.90	0.15		
	0.32	0.70	0.60	0.30			

Figure 4: Schematic showing a highly discontinuous void fraction field with fluid 1 cells having $C = 1$ and fluid 2 having $C = 0$ and interfacial cells with $0 < C < 1$.

function) is added to the governing equations so as to capture the "extra" forces at the interface boundary. Thus, the use of single set of governing equations to describe the complete two-phase flow domain is called "one-fluid" formulation.

Using one-fluid formulation, the modified Navier–Stokes equation governing the motion of the two fluids separated by an interface can be written as[18]:

$$\rho(\phi)\left(\frac{\partial \mathbf{u}}{\partial t} + \nabla\cdot(\mathbf{uu})\right) = -\nabla p + \nabla\cdot\left[\mu(\phi)\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right)\right] + \rho(\phi)\mathbf{g} + \mathbf{f}_v^\sigma \tag{1}$$

where ϕ represents an indicator function which is an implicit representation of the interface, $\rho(\phi)$ and $\mu(\phi)$ are the density and viscosity functions, respectively, which gives us the fluid density and viscosity at any point in the domain based on the indicator function. The capillary forces which act only at the interface are made volumetric using smoothed surface Dirac delta function, and are denoted by $\mathbf{f}_v^\sigma$. The formulation of volumetric capillary forces and its numerical approximation will be discussed in detail in a subsequent section.

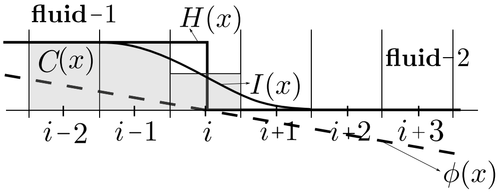


Figure 3: Interface representation using an indicator function $I(x)$, a Heaviside step function $H(x)$, a void fraction field $C(x)$ and a level-set function $\phi(x)$.

Other interfacial forces such as electrohydrodynamic forces in leaky dielectric fluids in the presence of electric field are also generally included in $\mathbf{f}_v^\sigma$.

Here, the interface position determines the value of the indicator function that we have used in the one-fluid formulation. Figure 3 represents the various ways in which we can numerically represent an interface between two fluids in one-dimensional flow. Here, the exact interface is represented by the Heaviside step function in the i^{th} grid cell. This function represents the separation of the domain with fluid 1 on one side and fluid 2 on the other. The various ways in which we can numerically represent this well-defined interface are shown in the schematic.

The void fraction function is defined as the ratio of the amount of fluid 1 inside each grid cell to the total volume of the grid cell. The void fraction function has a value zero in cells that are completely filled with fluid 2 and one in cells that are completely filled with fluid 1. For every cell which contains an interface, the void fraction value lies in between zero and one. Thus, one can use an indicator function (denoted by $I(x)$) to numerically represent the interface. The interface can also be represented implicitly as the zero level set of a level-set function (denoted by dash–dot line), where level-set function is taken as a signed distance function which has a positive value in one fluid and negative in the other.

If we use void fraction field to represent the interface, we get a strongly discontinuous distribution of void fraction values in cells across the interface as shown in Fig. 4. since we have the underlying velocity field which advects each fluid phase, one can solve Eq. 2 to get an updated void fraction field and thus track the interface implicitly:

$$\frac{\partial C}{\partial t} + \mathbf{u} \cdot \nabla C = 0 \tag{2}$$

If we use higher order advection schemes to solve Eq. 2, we will incur too much dispersion error, due to strong discontinuity of the void fraction field across the interface thus making the void fraction field unphysical. In order to avoid this, if we choose a higher order non-oscillatory scheme to advect the interface, we observe diffusion of void fraction field across multiple cells around the interface thus losing the ability of the method to maintain a sharp interface. Thus, one needs specific algorithms and schemes to accurately advect Eq. 2 without losing the sharp nature of the interface. Based on how one advects the void fraction field and solve Eq. 2, VOF methods can be broadly classified into algebraic VOF and geometric VOF methods.

In algebraic VOF method, as the name implies, the fluxing of void fraction field through the cell faces is performed through algebraic approximation of the indicator function. Based on the flux calculation, these methods are classified as compressive schemes and THINC (tangent of hyperbola for interface capturing) schemes. Compressive schemes use the interface normal (orientation with respect to cell face) to determine the void fraction flux scheme. Examples of compressive schemes include but are not limited to high-resolution interface capturing scheme (HRIC)[20], compressive interface capturing scheme for arbitrary meshes (CICSAM)[21], switching technique for advection and capturing of surfaces (STACS)[22], high-resolution artificial compressive formulation (HiRAC) and modified-CICSAM (M-CICSAM)[23]. Conventional algebraic VOF methods are generally not as accurate as geometric VOF methods and are subjected to stricter cell CFL number stability criteria. Recent progress in courant independent algebraic VOF schemes[24–27] looks very promising due to simplicity of implementation and computational cost. THINC schemes[28] are a class of algebraic VOF methods in which the fluxes through the cell faces are computed after approximating the indicator function with a hyperbolic tangent profile. THINC-based schemes have demonstrated comparable accuracy to certain geometric VOF methods without the computational cost[29,30] and has been validated for standard test cases including shear flows and vortex flows[31]. Rigorous testing and validation of THINC-based methods in realistic large-scale simulations are necessary to affirm these claims. Developing accurate algebraic VOF schemes is of great relevance due to the simplicity of implementation and computational cost and the authors feel significant progress will be made in the near future in this area.

Another way of solving Eq. 2 is to reconstruct the interface exactly from the underlying void fraction field and perform geometric flux calculation in order to update the void fraction field. This approach is often referred to as geometric VOF and in this review we will describe this interface tracking method in detail.

Geometric advection based VOF methods have become the standard method for the numerical simulation of multiphase flows. Due to rapid development in computing power and problem-specific algorithms, even complex multiphase flows with highly disparate time and length scales can be simulated[32,33]. For example, Fig. 5 shows the breakup of a liquid drop under a cross-flow, where the bag formation and subsequent atomization of droplet particles is captured numerically. Gerris[34], an open source multiphase solver which employs geometric VOF to capture the interface, was used to simulate the above.

In this review, we describe the geometric VOF algorithm in detail. All the equations and examples are demonstrated for two-dimensional square grid cells. Relevant references are provided for the extension of the algorithm to three-dimensional and axisymmetric coordinate systems.

3 Volume of Fluid Method: Historical Perspective

Noh and Woodward (1976)[36] were first to attempt advection of void fraction using a reconstructed interface. In this method, called simple line interface construction (SLIC), the void fraction is advected by time splitting, where C is advected in each direction sequentially. During advection in the x direction a mixed cell is divided by vertical lines separating full and empty parts, based on the void fraction values of the neighboring cells. Thus, each mixed cell would have fluid occupation numbers. Now, one can reconstruct interface geometry (a vertical line) according to 3 base cases, considering left, center, and right fluid occupation numbers. Once interface

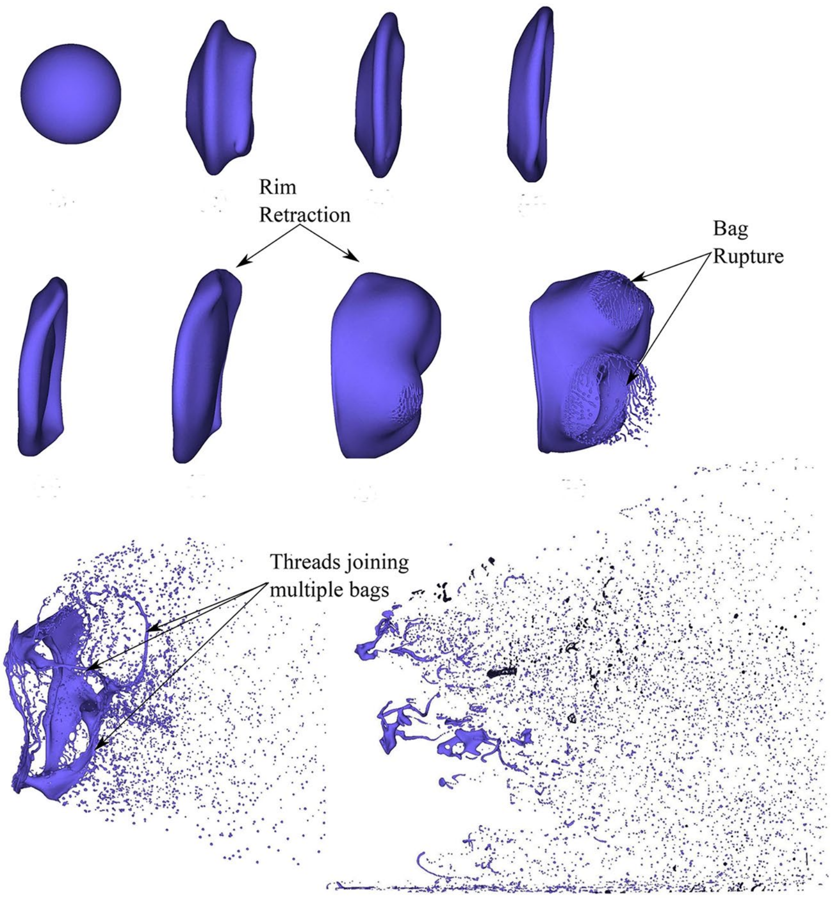


Figure 5: Numerical simulation of the disintegrating droplet under a cross-flow for Weber number, $We = 80$ at different times[35].

is determined, Eq. 2 is solved in x-direction by time integration of fluxes. We repeat the process for y-direction using horizontal lines for interface reconstruction.

Hirt and Nichols (1981)[37] modified this method (coining the term volume of fluid) where reconstructed interface is parallel to the coordinate axis but same orientation is used for advection in each direction. The orientation of the interface (whether it is horizontal or vertical) was determined by finding normal to the interface (using gradient of the void fraction field in the neighboring cells) and determining whether interface is aligned more towards horizontal or vertical coordinate axis. Tests by Rudman et al.[38] showed that neither of these methods are particularly accurate and resulted in islands of phase 1 generated numerically called “floatsam” or “jetsam”. Even for simple flows, these islands of fluid 1 which are numerically generated can make the flow completely unphysical (Fig. 6).

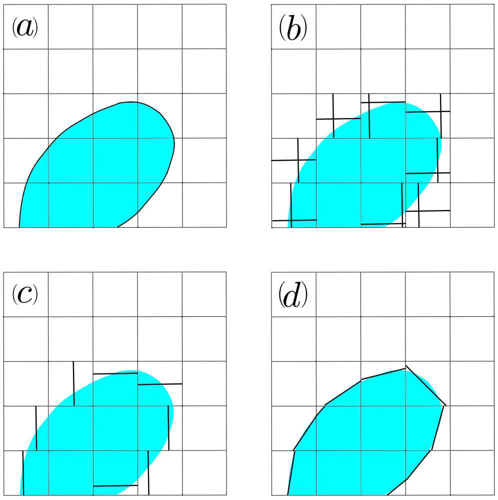


Figure 6: Reconstruction of the interface in VOF in two dimensions. **a** The original interface. **b** The original SLIC reconstruction. **c** Hirt and Nichols reconstruction. **d** The *PLIC* reconstruction.

Youngs (1982)[39] proposed the method of piecewise linear interface calculation(*PLIC*)[A] where the interface is represented as a line segment with arbitrary orientation. The orientation of the line segment is determined by the interface normal obtained by finding the gradient of the void fraction field. Once the interface is reconstructed, the void fraction is advected geometrically across the cell faces. Accuracy of *PLIC*-based methods depends on the computationally obtained interface normal and curvature accuracy.

More accurate methods for interface reconstruction like PROST (parabolic reconstruction of surface tension)[40], where interface is assumed to be a parabola and geometrical area preserving VOF method[41], have been successfully used for interface reconstruction. Although more accurate, these methods have a significantly high computational cost compared to *PLIC* method. Hybrid methods have been proposed to combine the advantages of the VOF method with other interface tracking methods, such as coupled level-set volume of fluid method (CLSVOF)[42] and phase field coupled with VOF[43]. Among these hybrid methods, CLSVOF is the most widely adopted as it significantly improves the accuracy of the normal and curvature calculation[44]. Although the accuracy of CLSVOF is higher than conventional VOF, the computational cost and bottlenecks in achieving parallel scalability are issues yet to be resolved. Considering the ease of implementation, computational cost and high degree of accuracy, the authors recommend using a *PLIC*-based VOF method[44] with geometric advection.

4 Initialization of Void Fraction

The first step in VOF method is the initialization of volume fraction in each grid cell in the computational domain. It is trivial to initialize the volume fraction field for grid cells which are lying completely inside either fluid 1 (where $C = 1$) or completely inside fluid 2 (where $C = 0$). For interfacial cells which contain the interface between fluid 1 and fluid 2, an accurate initialization of volume fraction is very important since initial loss of volume of a fluid can result in erroneous simulations as the flow progresses. For accurate determination of volume fraction in mixed cells, we need to compute the volume cut by the interface and the cell boundary.

Table 1: Results for relative error in volume for a circle of radius 0.25 for different grid sizes.

Relative error in volume		
Grid	VOFI Library	Gerris Solver
16 × 16	4.7×10^{-16}	5.5×10^{-3}
32 × 32	2.8×10^{-16}	6.4×10^{-3}
64 × 64	4.3×10^{-16}	3.1×10^{-4}
128 × 128	7.1×10^{-16}	5.1×10^{-4}

Given an implicit equation of an interface, a direct method for initializing volume fraction in an interfacial cell is to distribute internal points and then volume fraction is the ratio of number of points inside the fluid 1 with total number of distributed points inside the cell. Monte Carlo methods based sampling techniques can be used to increase the accuracy of such methods[45]. Another strategy is to recursively refine each grid cell locally and to use a linear line segment as an approximation of the interface in the most refined sub-cell[46]. These methods while relatively easy to implement are computationally very expensive. The accuracy of these methods are directly proportional to the computational cost.

A more robust method for determining the volume fraction field is by direct integration of the implicit equation of the interface. VOFI[47] is an open source library which initializes void fraction field upto machine accuracy given an implicit interface equation in square grid cells in Cartesian coordinate system using direct integration methods. For a given mixed cell, using the implicit interface equation, the local height function (see Fig. 14) is integrated with a variable number of nodes using double Gauss–Legendre quadrature rule. Table 1 shows the relative error in initialized volume fraction field between VOFI library and Gerris flow solver[34]. We initialize a circle of radius $r = 0.25$ in the center of a computational domain of size 1 × 1. The area of the circle is given by $V_c = \pi r^2$. The accuracy of initialization can be determined by calculating the relative error in the volumes given by

$$E = \frac{|V - V_c|}{V_c} \tag{3}$$

where V is the computed volume given by $V = \Sigma \Delta x \Delta y C$. Thus, it is quite clear that direct integration methods can initialize volume fraction fields up to machine accuracy. Ananthan and

[A] PLIC: piecewise linear interface calculation is a numerical representation of the interface using a line segment with an arbitrary orientation,

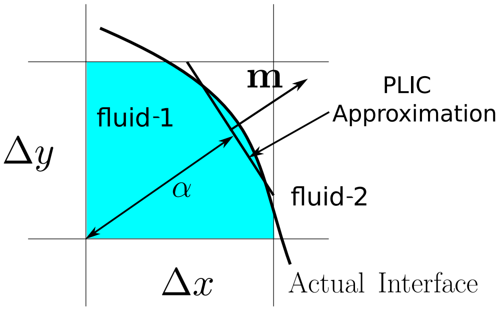


Figure 7: Reconstructed *PLIC*-based line segment representing the interface separating fluid 1 and fluid 2.

Tomar[48] proposed an algorithm using which one can extend this method to axisymmetric coordinate systems as well. An optimized version of VOFI library was recently released by Zaleski et al.[49].

5 Reconstruction

Given a volume fraction field in each grid cell in the computational domain, the first step in VOF method is the reconstruction of the *PLIC*-based line segment representing the interface in each mixed cell. Assume the equation of this line segment for a two-dimensional case given by

$$\mathbf{m}.\mathbf{x} = m_x x + m_y y = \alpha. \tag{4}$$

Here, m_x and m_y are the components of the normal to the line and α is the line constant (Fig. 7).

Since the void fraction within the cell does not provide any indication about the direction of the interface, the normals must be explicitly calculated. There exist many different methods to determine the normals, simplest approximation being the Youngs method[39]. Here, the normal vector is given by the gradient of the volume fraction field:

$$\mathbf{m} = -\nabla_h C. \tag{5}$$

Green–Gauss gradient is used which employs a 3 × 3 block of cells to determine the normal components at the central cell (i, j) (see Fig. 8a) as

$$\begin{aligned} m^x_{i,j} = -&\frac{1}{\Delta x} \\ &(C_{i+1,j+1} + 2C_{i+1,j} + C_{i+1,j-1} \\ &- C_{i-1,j+1} - 2C_{i-1,j} - C_{i-1,j-1}) \end{aligned} \tag{6}$$

$$\begin{aligned} m^y_{i,j} = -&\frac{1}{\Delta y} \\ &(C_{i+1,j+1} + 2C_{i,j+1} + C_{i-1,j+1} \\ &- C_{i+1,j-1} - 2C_{i,j-1} - C_{i-1,j-1}). \end{aligned} \tag{7}$$

This method is first-order accurate and can lead to drastic errors in estimation of normals.

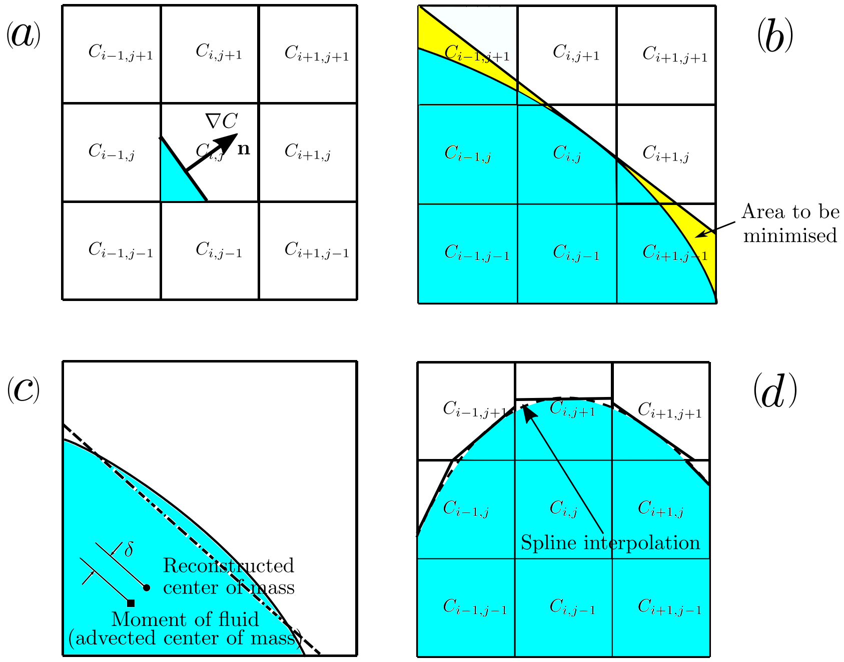


Figure 8: Various methods in determining interface normal in *PLIC*-based reconstruction. **a** Youngs gradient method. **b** Least square minimization of area. **c** Moment of fluid method. **d** Spline interpolation method. Figure adapted from Comminal et al.[53].

Scardovelli and Zaleski[50] introduced centered columns approach where in the same neighborhood block of cells, volume fractions can be added in a vertical/horizontal direction to determine a height/width function, respectively (see Fig. 14). Using this function, one can compute the interface normal with a centered scheme. Aulisa et al.[51] introduced a method which combines Youngs method and centered columns approach, which is shown to perform accurately in both high and low resolutions. Due to low computational cost and high accuracy, this method, namely mixed Youngs centered (MYC), is widely adopted in several open source codes[34,52].

Puckett et al.[54] proposed a least square VOF interface reconstruction algorithm (LVIRA) in which one minimizes the area between the extended interface into the nearby cells and the actual volume fractions in these cells (see Fig. 8b). Since the minimization requires solution of a non-linear problem, an iterative optimization algorithm is used in this method. Pilliod and Puckett[55] introduced a more efficient variation of this method called ELVIRA (Efficient LVIRA). Both LVIRA and ELVIRA are second-order accurate but computational cost is high compared to other methods especially for three-dimensional problems. Algorithm 1 discusses an efficient implementation[56] of ELVIRA algorithm of two-dimensional problems.

In moment of fluid interface reconstruction algorithm[57], one advects the center of mass of the liquid phase in each mixed cell. The algorithm consists of minimizing the distance δ between the center of mass of reconstructed interface and the advected center of mass, see Fig. 8c. Since a line can be determined by only two parameters (an intercept and a slope), the linear interface in a cell is actually over-determined by specifying the volume fraction and centroid; thus, an exact reconstruction of the linear interface can be computed[58]. Another advantage of this method is that this algorithm does not have a dependency on the neighborhood volume fraction field. Recent studies[58–60] present moment of fluid method as a viable alternative for standard void fraction-based reconstruction algorithms. Alternatively, one can reconstruct the interface by spline interpolation of the slopes along the interface[61,62], see Fig. 8d. Using this algorithm, one can estimate the surface curvature more accurately which would be useful in modeling the surface tension forces.

Once the interface normal is obtained, the next step in *PLIC* interface reconstruction is the determination of the line constant α. Given void fraction field and interface normals, one can enforce conservation of volume to determine the line constant geometrically. For two-dimensional square grid cell with unitary

Algorithm 1 2D ELVIRA interface reconstruction algorithm[56]

```
For each cell:
  Skip if void fraction is 0 or 1.
  Gather VOF data for neighboring $3 \times 3$ block into working array.
  Compute 6 ELVIRA candidate normals (using height function), and append
  computational update of previous normal vector as $7th$ candidate.
  for each candidate normal do
    Compute correct location of interface to match center cell volume fraction.
    for each of the cell in the $3 \times 3$ block do
      Extend the interface from the center cell and compute the implied volume
      fraction.
      Accumulate the square of the difference between the actual volume fraction
      and the implied volume fraction.
    end for
    If the error for this normal vector is less than the minimum so far, Note this as
    minimizing index.
  end for
  Set the interface normal to the minimizing value out of the candidates, and record
  the interface state.
```

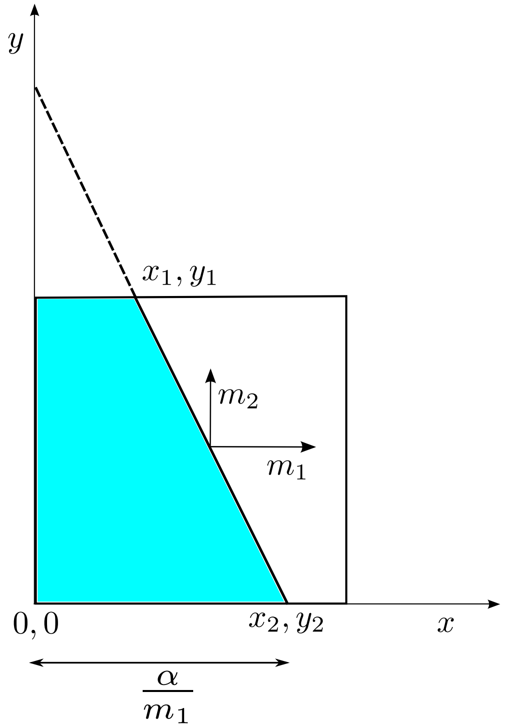


Figure 9: The standard case considered for interface reconstruction in which both $m_1, m_2 \geq 0$ and fluid 1 is occupying the bottom left corner of the cell.

normal vector $|\mathbf{n}| = 1$, for the case $m_x, m_y \geq 0$, the volume/area enclosed by the *PLIC* interface is given by

$$V = \frac{\alpha^2}{2m_x m_y}, \tag{8}$$

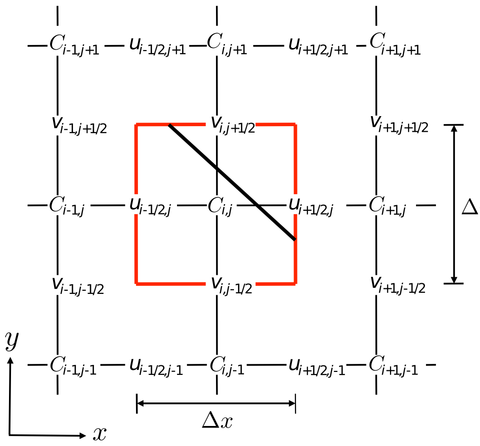


Figure 10: The control volume for a mixed cell where the void fraction values are stored in the cell center and velocities are stored in staggered positions in the cell faces.

provided both the x and y intercepts of the *PLIC* line lie within the grid cell, that is,

$$\frac{\alpha}{m_{x,y}} \leq \Delta x. \tag{9}$$

If this condition is not met (see Fig. 9), relevant triangular areas are subtracted from right hand side of Eq. 8 to obtain the enclosed volume. Since we know that enclosed volume is a function of the void fraction, one can analytically solve this equation to obtain a relationship between the void fraction, line constant and interface normal. Zaleski et al.[63] derived these relations for rectangular grids in Cartesian coordinates for a standard case ($m_x, m_y \geq 0$).

For a two-dimensional square grid cell, taking the positive normalized values of m_x and m_y and using the relation below, one can get the line constant α. Let $m_1 + m_2 = 1, (m_1 \leq m_2)$ and $A_1 = m_1/(2m_2)$, then,

$$\alpha = \sqrt{2m_1 m_2 A} \qquad for: 0 \leq A < A_1 \tag{10a}$$

$$\alpha = m_2 A + \frac{m_1}{2} \qquad for: A_1 \leq A \leq \frac{1}{2}. \tag{10b}$$

In practice since the normals can be negative, the geometry is mapped in to the standard case using linear transformations. Once the line constant is obtained, it is mapped back into the original configuration. Ananthan and Tomar[48] have extended this approach for axisymmetric coordinate system. Zaleski et al.[63] derived these relations for three-dimensional problems and Lehmann and Gekle[64] extended these relations to capture all the edge cases.

6 Advection of Void Fraction Field

Once the interface is reconstructed, the second recurring step in VOF method is the advection of void fraction. Given locally divergence-free velocity field at the cell faces of a computational grid cell of void fraction field (see Fig. 10), we advect the void fraction field geometrically using the advection equation:

$$\frac{\partial C}{\partial t} + \mathbf{u} \cdot \nabla C = 0. \tag{11}$$

Equation 11 can be written as

$$\frac{\partial C}{\partial t} + \nabla \cdot (\mathbf{u}C) = C(\boldsymbol{\nabla} \cdot \mathbf{u}). \tag{12}$$

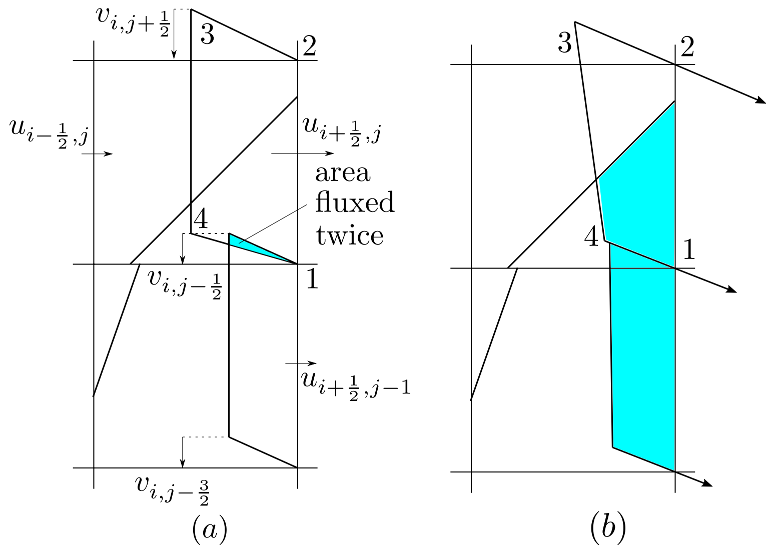


Figure 11: Unsplit advection methods with different flux polygon constructions. **a** Rider and Kothe[65] used staggered velocities to construct the trapezoid. **b** Lopez et al.[62] introduced a method where velocities interpolated to cell vertices were used to construct the trapezoid. Figure adapted from Tryggvasson et al.[18].

Thus, one can time integrate Eq. 12 by finding the void fraction fluxes across the cell faces accurately. Based on the calculation of the fluxes of void fraction field, geometric advection methods can be classified into (a) unsplit methods[65] and (b) split methods[54]. Split methods use operator splitting to advect the void fraction field in each spatial direction sequentially, while the unsplit methods advect in a single step.

In unsplit method, the right hand side term in Eq. 12 goes to zero since the advection takes just a single step. Thus, in an unsplit method, the void fraction flux calculation across the cell faces involves calculation of volume/area of complex polyhedra/polygon cut by the interface. These calculations are computationally expensive and difficult to implement accurately, but the advantage being only a single advection and reconstruction step is required per timestep. Rider and Kothe[65] introduced one of the first unsplit algorithms in *PLIC*-based reconstruction. In this approach, face-centered velocities were used to construct trapezoids (see Fig. 11a), which were used to determine the fluxing volumes/areas. This leads to regions of overlap resulting in overshoots or undershoots and the void fraction field was not locally conserved. Lopez et al.[62] introduced an unsplit method which uses cell vertex velocities to construct the advected polygon in two dimensions. This approach ensures no overlap between the fluxed volumes, see Fig. 11b. Owkes and Desjardins[66] extended this approach for three dimensions thus detailing a locally conservative and bounded unsplit algorithm. Interface reconstruction library is an open source code by Desjardins et al.[67] which can be used for accurate computation of the volume of complex polyhedra generated during unsplit advection.

In split method, we use operator splitting to advect the void fraction field in each spatial direction sequentially, thus the RHS term (called dilatation or divergence correction term) in Eq. 12 is needed since the velocity field is not divergence free in each direction independently. Split methods can be broadly divided into Eulerian and Lagrangian methods[18]. In Lagrangian methods the end points of the reconstructed interface are advected by the flow in each direction sequentially. Here, the local velocity maybe interpolated between the cell faces. A detailed overview of Lagrangian advection method is given by Tryggvasson et al.[18]. In Eulerian advection methods, the flux through each cell face is computed sequentially. We will be discussing Eulerian-based advection methods in detail here.

Given a volume fraction ($C^n_{i,j}$) and velocity field ($u^n_{i+1/2,j}, v^n_{i,j+1/2}$) at the *nth* timestep, the discretized Eq. 12 is given by

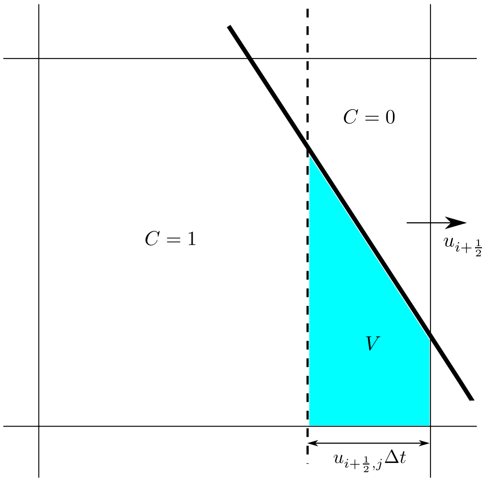


Figure 12: The fluxed volume through the right face of the cell when $u_{i+1/2,j}$ is positive.

$$\begin{aligned}C_{i,j}^{n+1} &= C_{i,j}^{n} + \frac{\Delta t}{\Delta x}\left(\delta V_{i-1/2,j} - \delta V_{i+1/2,j}\right)\\ &+ \frac{\Delta t}{\Delta y}\left(\delta V_{i,j-1/2} - \delta V_{i,j+1/2}\right)\\ &+ C_{i,j}^{n}\left(\frac{\Delta t}{\Delta x}\left(u_{i+1/2,j}^{n} - u_{i-1/2,j}^{n}\right)\right.\\ &\left.+\frac{\Delta t}{\Delta y}\left(v_{i,j+1/2}^{n} - v_{i,j-1/2}^{n}\right)\right)\end{aligned} \tag{13}$$

where $\delta V_{i+1/2,j} = (uC)_{i+1/2,j}^{n}$ is the amount of volume fraction fluxed through the right cell face. Similarly, fluxes $\delta V_{i-1/2,j}, \delta V_{i,j+1/2}$ and $\delta V_{i,j-1/2}$ can be computed for other cell faces.

Using operator splitting, we can split the above equation as following:

$$\begin{aligned}C_{i,j}^{*} &= C_{i,j}^{n} + \frac{\Delta t}{\Delta x}\\ &\left(\delta V_{i-1/2,j} - \delta V_{i+1/2,j}\right)\\ &+ C_{i,j}^{c}\left(\frac{\Delta t}{\Delta x}\left(u_{i+1/2,j}^{n} - u_{i-1/2,j}^{n}\right)\right)\end{aligned} \tag{14}$$

$$\begin{aligned}C_{i,j}^{n+1} &= C_{i,j}^{*} + \frac{\Delta t}{\Delta y}\\ &\left(\delta V_{i,j-1/2} - \delta V_{i,j+1/2}\right)\\ &+ C_{i,j}^{c}\left(\frac{\Delta t}{\Delta y}\left(v_{i,j+1/2}^{n} - v_{i,j-1/2}^{n}\right)\right)\end{aligned} \tag{15}$$

where $C_{i,j}^{*}$ is the intermediate value of the volume fraction. Here, $C_{i,j}^{c}$ is a function based on which we can get different operator splitting approaches. If $C_{i,j}^{c} = C_{i,j}^{*}$, then we get Puckett's operator split approach[54]. An implicit scheme is used in the first direction and an explicit scheme in the second direction to maintain the conservation of volume fraction[54]. The order of sweep direction is alternated every timestep[68] ("Strang splitting") to achieve second-order accuracy in time. If we use,

$$C_{i,j}^{c} = \begin{cases} 1 & C_{i,j}^{n} \geq 0.5 \\ 0 & C_{i,j}^{n} < 0.5 \end{cases} \tag{16}$$

where $C_{i,j}^{c}$ depends only on the previous timestep value of volume fraction, we get Weymouth and Yue's operator split approach[69]. This method is explicit in both the sweep directions.

For an operator split method to be strictly conservative, it has to fulfill (following Weymouth and Yue[69]) a set of concurrent requirements:

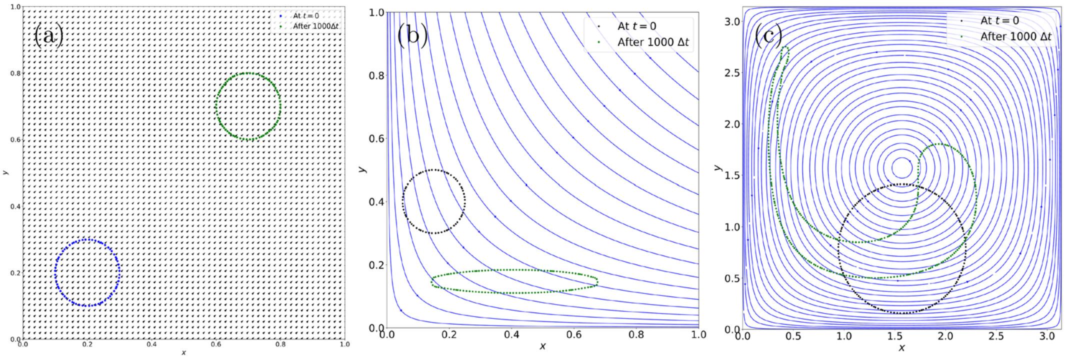


Figure 13: A few standard test cases for the validation of reconstruction and advection of void fraction.

Table 2 Results for relative error in change in volume for different test cases for $\Delta x = 0.25$ and $CFL = 0.25$

Relative error in volume					
Case	Domain	u velocity	v velocity	Puckett's method	Weymouth and Yue
(a)	1×1	0.1	0.1	3.1×10^{-7}	ϵ
(b)	1×1	x	$-y$	5.1×10^{-6}	ϵ
(c)	$\pi \times \pi$	$\sin x \cos y$	$-\cos x \sin y$	1.5×10^{-7}	ϵ

Using Weymouth and Yue's[69] method one can discretely conserve mass upto machine precision ϵ

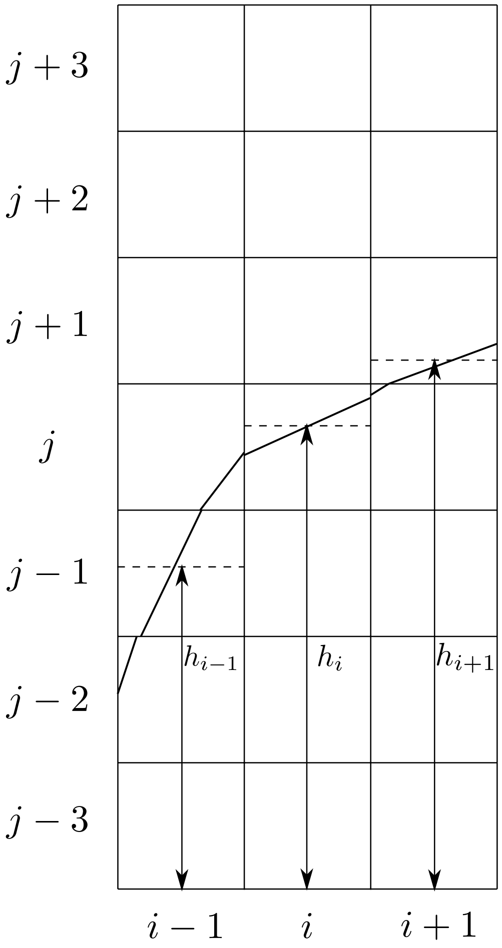


Figure 14: Schematic showing the height function calculation for the case $|n_y| > |n_x|$ in the cell (i, j). Adapted from Afkhami et al.[80].

1. The volume flux terms are conservative, and
2. the dilatation term sums to zero, and
3. no clipping or filling of a cell is needed to impose $0 \leq C \leq 1$.

Using geometric advection for volume fluxing and a divergence-free velocity field, both Puckett's and Weymouth and Yue's methods satisfy the first two requirements. Puckett's method does require clipping or filling of a cell to impose $0 \leq C \leq 1$ and, therefore, is not strictly conservative. However, the mass conservation errors introduced in this method are small and thus it has been widely adopted previously. Weymouth and Yue[69] proved that, by imposing a grid Courant number restriction of,

$$\Delta t \sum_{d=1}^{N} |\frac{u_d}{\Delta x_d}| < \frac{1}{2} \qquad (17)$$

one can satisfy the third requirement for the operator split approach to be conservative. Here, N is the number of sweep directions, and u_d and Δx_d are the maximum velocity and grid size in each of those directions. Thus, using Weymouth and Yue's operator splitting approach, we can preserve the mass exactly when advecting the volume fraction field. Note that this method can be easily extended to three-dimensional calculations.

The volume flux through cell faces, $\delta V_{cell-face}$, is computed geometrically. Consider the schematic in Fig. 12, where the shaded region shows the volume of fluid-1 in the cell to be fluxed through the right face ($\delta V_{i+1/2,j}$). Considering the face velocity ($u_{i+1/2,j}$) to be positive, the flux can be computed as

$$\delta V_{i+\frac{1}{2},j} = \frac{u_{i+1/2,j} V}{u_{i+1/2,j} \Delta t \Delta y} = \frac{V}{\Delta t \Delta y} \qquad (18)$$

where V is the volume of fluid 1 fluxed through the right face (shown as the shaded region in Fig. 12). Since we have the equation of the *PLIC* line and the line of total fluxing (dashed line in Fig. 12), one can easily calculate the fluxed volume V through each face. Ananthan and Tomar[48] extended Weymouth and Yue's approach for axisymmetric coordinate system.

Figure 13 shows a few standard test cases for the validation of reconstruction and advection of the void fraction field. Void fraction field is initialized in a circle and subjected to underlying velocity fields and this velocity field is reversed after a certain number of timesteps for the same duration and final and initial geometries are then compared. Case (a) is a translating circle, case (b) is circle in a corner flow[69] and case (c) is circle in a vortex[65,70]. Table 2 gives the relative change in volume for each of these cases for both Puckett's method and Weymouth and Yue's approach. Other standard test cases include Zalesak's disk[71] and three-dimensional deformation case proposed by Leveque[72].

Note that geometric advection of void fraction conserves mass upto machine precision if the underlying discretized velocity field is divergence free. Ananthan and Tomar[48] found that as the divergence-free nature of the velocity deteriorates so does the accuracy of the geometric VOF. In practice, accuracy of the velocity field is determined by the error tolerance of the pressure Poisson equation. Thus, multiphase flow simulations require use of efficient and highly accurate pressure Poisson solvers[73].

7 Numerical Calculation of Surface Tension Forces

Consider one-fluid formulation of Navier–Stokes equation:

$$\begin{aligned}\rho(\phi)\left(\frac{\partial \mathbf{u}}{\partial t}+\boldsymbol{\nabla}\cdot(\mathbf{u}\mathbf{u})\right) &= -\nabla p \\ &+ \boldsymbol{\nabla}\cdot\left[\mu(\phi)\left(\nabla\mathbf{u}+\nabla\mathbf{u}^T\right)\right] \\ &+ \rho(\phi)\mathbf{g}+\mathbf{f}^\sigma.\end{aligned} \tag{19}$$

Here, $\mathbf{f}^\sigma$ is the surface tension force per unit volume. Numerical approximation of the surface tension forces in two-phase flow simulations is an area which has seen considerable progress in the last 2 decades. An excellent review of these different methods is given by Popinet[74]. Here, we will look into the numerical approximation of surface tension forces through volumetric formulation (body forces) since this approach ensures discrete balance of surface tension and pressure gradient terms[75].

In an elementary volume Ω intersected by the curve at two points A and B, the total surface tension force is given by $\int_\Omega \mathbf{f}^\sigma = \oint_A^B \sigma d\mathbf{t}$ where σ is the surface tension and $\mathbf{t}$ is the unit tangent vector. Using the first Frenet formula for parametric curves $d\mathbf{t} = \kappa\mathbf{n}ds$ where κ is the curvature, $\mathbf{n}$ is the unit normal and s the curvilinear coordinate, we get

$$\begin{aligned}\int_\Omega \mathbf{f}^\sigma &= \oint_A^B \sigma\kappa\mathbf{n}ds \\ &= \int_\Omega \sigma\kappa\mathbf{n}\delta_s\end{aligned} \tag{20}$$

where δ_s is the surface Dirac δ-function which is non-zero only at the interface[18]. For variable surface tension, tangential (Marangoni) stresses have to be included in this formulation. Afkhami et al.[76] extended this formulation for variable surface tension forces.

Based on the computation of curvature κ, the normal $\mathbf{n}$ and the Dirac δ-function, different numerical approximation for volumetric surface tension forces are obtained such as continuous surface force (*CSF*)[77], ghost-fluid method (GFM)[78] and smoothed Heaviside method[79]. We discuss here the continuous surface force model of Brackbill et al.[77]. The relation[74]

$$\sigma\kappa\mathbf{n}\delta_s = \sigma\kappa\nabla H(\mathbf{x}-\mathbf{x_s}) \tag{21}$$

where H is the Heaviside function and $\mathbf{x_s}$ is the position of the interface gives a smoothed Dirac delta representation of the surface tension force. In *CSF*[B] model, we choose void fraction field, $H_\epsilon = C$ to represent the Heaviside function where $\epsilon = \Delta$, the grid size such that

$$\lim_{\epsilon\to 0} H_\epsilon = H. \tag{22}$$

Depending upon the choice of the numerical method, one can choose different approximations for the Heaviside function. The next step is to accurately estimate the interface curvature. One can use the relation $\mathbf{n} = -\frac{\nabla C}{|\nabla C|}$ and $\kappa = -\nabla.\mathbf{n}$. Even if we use a smoothed void fraction field $\tilde{C}$ as originally proposed by Brackbill et al., this approach does not result in a numerically consistent curvature estimation. A widely adopted and accurate method to estimate the curvature is to use the height function approach.

The height function approach is based on the simple idea that any interface can be described as a graph of a function defined in a local coordinate system. One can thus determine the derivatives of this function to estimate the curvature. Sussman[81] introduced the height function approach for curvature calculation. Cummins

[B] CSF: continuous surface force model is a numerical representation of surface tension forces where the volumetric surface tension forces are approximated using void fraction field as the Heaviside function.

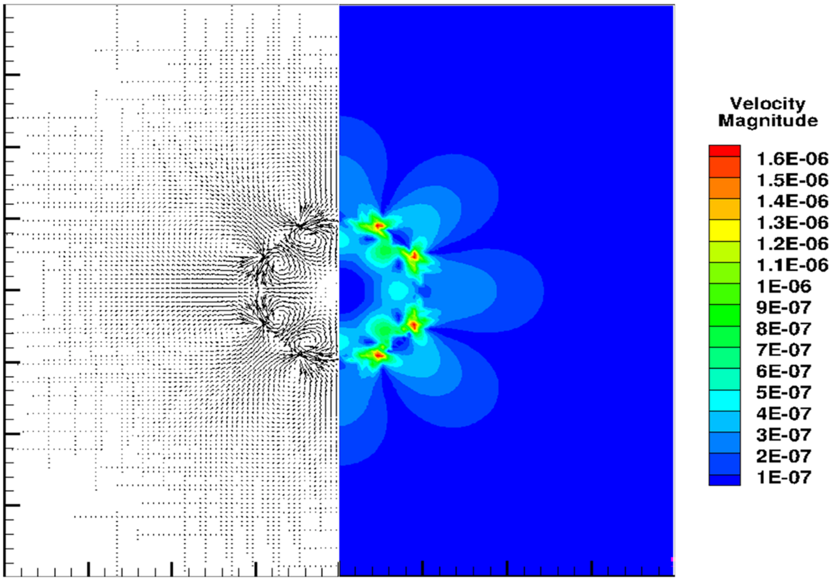


Figure 15: The static drop test case in which a drop of diameter, $D = 1$ is initialized into a domain of size 4×4. The test parameters are Laplace number, $La = \frac{\rho_{liquid} D\sigma}{\mu^2_{liquid}} = 0.120$, the density and viscosity ratios are unity and $\frac{D}{\Delta x} = 30$.

et al.[82] and Popinet[83] have analyzed and extended this approach for robust calculation of the curvature on a structured mesh. Estimation of the curvature using height function involves the following steps:

1. Determine the local coordinate orientation using the maximal component of the normal **n**.
2. Estimate the height function in this orientation by summing over the void fraction values.
3. Compute the curvature using central differencing.

In Fig. 14 for cell (i, j), we have $|n_y| > |n_x|$; thus, we use a 3×7 stencil to compute the vertical height function $y = h(x)$. In order to compute the discrete height function, we use

$$h_i = \Delta x \sum_{k=-3}^{3} C_{i,j+k}. \tag{23}$$

Thus, we can compute the curvature as

$$\kappa = -\frac{h_{xx}}{(1 + h_x^2)^{3/2}} \tag{24}$$

where the derivatives h_x and h_{xx} are computed using central differencing. Similarly, we can choose a 7×3 stencil for the horizontal case. It is found that the height function approach is second-order accurate and fairly easy to implement in comparison to other methods of comparable accuracy. In order to reduce the computational expense and increase the accuracy of estimation of the curvature, one can adopt a variable stencil height function approach as described by Popinet[83] and Lopez et al.[84]. Accuracy of curvature estimation using Height function approach deteriorates rapidly when the interface is not well resolved. Popinet[74] reported that when $\kappa \Delta x > 1/5$, it is extremely difficult to obtain consistent height functions. One has to use alternative techniques like parabolic reconstruction of interface (PROST)[40] in these scenarios. Desjardins et al.[85] introduced a mesh decoupled height function approach which is found to improve the curvature calculation for under-resolved interfaces thus alleviating this issue.

Since volumetric formulation of surface tension forces are neither well balanced nor momentum conserving[74], we see spurious velocity currents generated at the interface. These numerically generated velocity currents can lead to artificial generation of kinetic energy and heat transfer[86,87]. Dodd et al.[88] found that only if the root mean square velocity of the spurious currents is negligible in comparison to velocity fluctuations due to turbulence, one can resolve a turbulent flow fully. The standard benchmark test to access the accuracy of numerical implementation of surface tension forces is the static drop test case[89].

Figure 15 shows the static drop test case where we see spurious velocity currents in the long time limit. In this test case, the timescale required to reach equilibrium depends upon the oscillation timescale, $T_\sigma = \sqrt{\frac{\rho D^3}{\sigma}}$ and the viscous dissipation timescale $T_\mu = \frac{\rho D^2}{\mu}$. In order for the simulations to correspond to the equilibrium state, one needs to run the simulation longer than either of these timescales[74]. As Magnini et al.[90] reported, it would be more insightful if one were to look at the time averaged norm of the spurious current than the spurious current magnitude at an instant. Detailed parameters and test case setup can be found in Lafaurie et al.[91] and Popinet and Zaleski[92]. One can argue that if the current implementation of surface tension force limits the maximum magnitude of spurious current in the long time limit then it can be used accurately for simulation of multiphase flow. Other than the static drop test case one needs to look at the translated droplet test case[83]. This test case is more relevant for practical applications and an interesting comparative study was presented in Abadie et al.[93]. Other standard test cases include capillary oscillations of a drop/bubble in the limit of vanishing amplitude and viscosity both for planar and circular/spherical interfaces[94]. Other more challenging variants can be found in Torres and Brackbill[95], Herrmann[96] and Fuster et al.[33]. Another standard test case is that of the analytical solution of exponential damping of oscillatory modes of a capillary wave when viscosity is introduced. Prosperetti[97,98] derived closed-form solutions for this case and details for the test case are given in Popinet and Zaleski[92].

8 Looking Ahead

Using VOF methods, many areas of practical interest other than two-phase flows have been extensively investigated including but not limited to heat transfer[99,100], thermocapillary motion[101,102], mass transfer[103,104], insoluble and soluble surfactants[105,106], electrohydrodynamics[107,108], boiling and evaporation[109–111], solidification and dendrite formation[112–114] and particle-laden flows[115]. In most of these cases, one needs to modify VOF or couple it with additional governing equations so as to capture the physics of the phenomenon under study. The authors anticipate significant advancements in these domains in the near future, both in terms of development of computational algorithms and in study of highly specialized areas using the VOF method.

Multiphase flows typically involve highly disparate time and length scales; thus, for numerical simulation of any realistic flow, one needs to use either unstructured grid or adaptive mesh refinement (AMR). Although VOF algorithm described here is for structured mesh, it is straight forward to adapt them for AMR grid structures. In both cell-based[34,83,116] and block-based[117] AMR grids, geometric VOF methods have been implemented successfully. For unstructured grids, algebraic VOF is the preferred method due to inherent computational cost and difficulty of implementation of geometric VOF. In the last decade, we have seen open source tools and libraries like VOFTools[118,119] being used for geometric VOF in unstructured grids and we expect to see the community adopting these resources.

As in every other field of science and technology, machine learning (ML), Artificial intelligence (AI), and data science are rapidly adopted and used in fluid dynamics research also[120]. In VOF methods, machine learning has been successfully used for the estimation of curvature[121–123], reconstruction of interface[124] and design of accurate schemes for advection of void fraction[125]. Machine learning techniques are also being used for developing new robust predictive models for multiphase flows and reducing the overall computational effort and time[126]. It is the firm belief of the authors that this is the beginning of a promising transformative paradigm and we are at its infancy.

In summary, in this review, we have given an overview of geometric VOF method and described in detail the methods and algorithms that maintain a balance between the ease of implementation, computational cost and accuracy of the solution. Using the framework of VOF method, one can tackle several interesting and relevant multiphase flow problems which in turn are expected to lead to creative extensions of these existing methods and development of entirely problem specific novel algorithms.

Publisher's Note

Data Availability

Data will be made available on request.

Declarations

Conflict of Interest

On behalf of all the authors, the corresponding author states that there is no conflict of interest.

Received: 5 January 2024 Accepted: 26 February 2024
Published online: 16 April 2024

References

1. Mirjalili S, Jain SS, Dodd M (2017) Interface-capturing methods for two-phase flows: an overview and recent developments. Center Turbul Res Annu Res Briefs 2017(117–135):13
2. Sato Y, Sadatomi M, Sekoguchi K (1981) Momentum and heat transfer in two-phase bubble flow-i: theory. Int J Multiph Flow 7(2):167–177
3. Clift R, Grace JR, Weber ME (2005) Bubbles drops, and particles. Courier Corporation
4. Sirignano WA (2010) Fluid dynamics and transport of droplets and sprays. Cambridge University Press
5. Melville WK (1996) The role of surface-wave breaking in air-sea interaction. Annu Rev Fluid Mech 28(1):279–321
6. Deike L (2022) Mass transfer at the ocean-atmosphere interface: the role of wave breaking, droplets, and bubbles. Annu Rev Fluid Mech 54:191–224
7. Lohse D (2022) Fundamental fluid dynamics challenges in inkjet printing. Annu Rev Fluid Mech 54:349–382
8. Shaw RA (2003) Particle-turbulence interactions in atmospheric clouds. Annu Rev Fluid Mech 35(1):183–227
9. Prosperetti A, Tryggvason G (2009) Computational methods for multiphase flow. Cambridge University Press
10. Chen S, Doolen GD (1998) Lattice Boltzmann method for fluid flows. Ann Rev Fluid Mech 30(1):329–364
11. Ishii M, Hibiki T (2010) Thermo-fluid dynamics of two-phase flow. Springer
12. Inamuro T, Ogata T, Tajima S, Konishi N (2004) A lattice Boltzmann method for incompressible two-phase flows with large density differences. J of Comput Phys 198(2):628–644
13. Colagrossi A, Landrini M (2003) Numerical simulation of interfacial flows by smoothed particle hydrodynamics. J Comput Phys 191(2):448–475
14. Violeau D, Rogers BD (2016) Smoothed particle hydrodynamics (sph) for free-surface flows: past, present and future. J Hydraul Res 54(1):1–26
15. Tryggvason G, Bunner B, Esmaeeli A, Juric D, Al-Rawahi N, Tauber W, Han J, Nas S, Jan Y-J (2001) A front-tracking method for the computations of multiphase flow. J Comput Phys 169(2):708–759
16. Unverdi SO, Tryggvason G (1992) A front-tracking method for viscous, incompressible, multi-fluid flows. J Comput Phys 100(1):25–37
17. McKee S, Tomé MF, Ferreira VG, Cuminato JA, Castelo A, Sousa F, Mangiavacchi N (2008) The mac method. Comput Fluids 37(8):907–930
18. Tryggvason G, Scardovelli R, Zaleski S (2011) Direct numerical simulations of gas-liquid multiphase flows. Cambridge University Press
19. Gibou F, Fedkiw R, Osher S (2018) A review of level-set methods and some recent applications. J Comput Phys 353:82–109
20. Muzaferija S (1998) Computation of free surface flows using interface-tracking and interface-capturing methods. Nonlinear water-wave interaction. Computational Mechanics, Southampton
21. Ubbink O, Issa R (1999) A method for capturing sharp fluid interfaces on arbitrary meshes. J Comput Phys 153(1):26–50
22. Darwish M, Moukalled F (2006) Convective schemes for capturing interfaces of free-surface flows on unstructured grids. Numer Heat Transf, Part B: Fundam 49(1):19–42
23. Zhang D, Jiang C, Liang D, Chen Z, Yang Y, Shi Y (2014) A refined volume-of-fluid algorithm for capturing sharp fluid interfaces on arbitrary meshes. J Comput Phys 274:709–736
24. He Z, Ruan Y, Yu Y, Tian B, Xiao F (2022) Self-adjusting steepness-based schemes that preserve discontinuous structures in compressible flows. J Comput Phys 463:111268
25. Chakraborty B, Banerjee J (2016) A sharpness preserving scheme for interfacial flows. Appl Math Model 40(21–22):9398–9426
26. Arote A, Bade M, Banerjee J (2020) An improved compressive volume of fluid scheme for capturing sharp interfaces using hybridization. Numer Heat Transf, Part B: Fundam 79(1):29–53
27. Anghan C, Bade MH, Banerjee J (2021) A modified switching technique for advection and capturing of surfaces. Appl Mathl Model 92:349–379

28. Xiao F, Honma Y, Kono T (2005) A simple algebraic interface capturing scheme using hyperbolic tangent function. Int J Numer Methods Fluids 48(9):1023–1040
29. Yokoi K (2007) Efficient implementation of thinc scheme: a simple and practical smoothed vof algorithm. J Comput Phys 226(2):1985–2002
30. Xiao F, Ii S, Chen C (2011) Revisit to the thinc scheme: a simple algebraic vof algorithm. J Comput Phys 230(19):7086–7092
31. Qian L, Wei Y (2019) Improved thinc/sw scheme for computing incompressible two-phase flows. Int J Numer Methods Fluids 89(6):216–234
32. Scardovelli R, Zaleski S (1999) Direct numerical simulation of free-surface and interfacial flow. Annu Rev Fluid Mech 31(1):567–603
33. Fuster D, Agbaglah G, Josserand C, Popinet S, Zaleski S (2009) Numerical simulation of droplets, bubbles and waves: state of the art. Fluid Dyn Res 41(6):065001
34. Popinet S (2003) Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries. J Comput Phys 190(2):572–600
35. Suryaprakash R, Tomar G (2019) Secondary breakup of drops. J Indian Inst Sci 99(1):77–91
36. Noh WF, Woodward P (1976) Slic (simple line interface calculation). In: Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics June 28–July 2, 1976 Twente University, Enschede. Springer. pp 330–340
37. Hirt CW, Nichols BD (1981) Volume of fluid (vof) method for the dynamics of free boundaries. J Comput Phys 39(1):201–225
38. Rudman M (1997) Volume-tracking methods for interfacial flow calculations. Int J Numer Methods Fluids 24(7):671–691
39. Youngs DL (1982) Time-dependent multi-material flow with large fluid distortion. Numerical methods for fluid dynamics. Academic Press, pp 275–285
40. Renardy Y, Renardy M (2002) Prost: a parabolic reconstruction of surface tension for the volume-of-fluid method. J Comput Phys 183(2):400–421
41. Aulisa E, Manservisi S, Scardovelli R, Zaleski S (2003) A geometrical area-preserving volume-of-fluid advection method. J Comput Phys 192(1):355–364
42. Sussman M, Puckett EG (2000) A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows. J Comput Phys 162(2):301–337
43. Liu Y, Yu X (2016) A coupled phase-field and volume-of-fluid method for accurate representation of limiting water wave deformation. J Comput Phys 321:459–475
44. Gerlach D, Tomar G, Biswas G, Durst F (2006) Comparison of volume-of-fluid methods for surface tension-dominant two-phase flows. Int J Heat Mass Transf 49(3–4):740–754
45. Evans M, Swartz T (2000) Approximating integrals via monte Carlo and deterministic methods. OUP Oxford
46. Cummins SJ, Francois MM, Kothe DB (2005) Estimating curvature from volume fractions. Comput Struct 83(6–7):425–434
47. Bná S, Manservisi S, Scardovelli R, Yecko P, Zaleski S (2016) Vofi-a library to initialize the volume fraction scalar field. Comput Phys Commun 200:291–299
48. Mohan A, Tomar G (2021) Interface reconstruction and advection schemes for volume of fluid method in axisymmetric coordinates. J Comput Phys 446:110663
49. Chierici A, Chirco L, Le Chenadec V, Scardovelli R, Yecko P, Zaleski S (2022) An optimized vofi library to initialize the volume fraction field. Comput Phys Commun 281:108506
50. Scardovelli R, Zaleski S (2003) Interface reconstruction with least-square fit and split Eulerian-Lagrangian advection. Int J Numer Methods Fluids 41(3):251–274
51. Aulisa E, Manservisi S, Scardovelli R, Zaleski S (2007) Interface reconstruction with least-squares fit and split advection in three-dimensional cartesian geometry. J Comput Phys 225(2):2301–2319
52. Arrufat T, Dabiri S, Fuster D, Ling Y, Malan L, Scardovelli R, Tryggvason G, Yecko P, Zaleski S (2014) The PARIS-Simulator code
53. Comminal R, Spangenberg J, Hattel JH (2015) Cellwise conservative unsplit advection for the volume of fluid method. J Comput Phys 283:582–608
54. Puckett EG, Almgren AS, Bell JB, Marcus DL, Rider WJ (1997) A high-order projection method for tracking fluid interfaces in variable density incompressible flows. J Comput Phys 130(2):269–282
55. Pilliod JE Jr, Puckett EG (2004) Second-order accurate volume-of-fluid algorithms for tracking material interfaces. J Comput Phys 199(2):465–502
56. Robey JM (2019) On the design, implementation, and use of a volume-of-fluid interface tracking algorithm for modeling convection and other processes in the earth mantle. University of California, Davis
57. Dyadechko V, Shashkov M (2005) Moment-of-fluid interface reconstruction. Los Alamos Report LA-UR-05-7571, 49
58. Kucharik M, Garimella RV, Schofield SP, Shashkov MJ (2010) A comparative study of interface reconstruction methods for multi-material ale simulations. J Comput Phys 229(7):2432–2452
59. Zinjala H, Banerjee J (2016) A lagrangian-eulerian advection scheme with moment-of-fluid interface reconstruction. Numer Heat Transf, Part B: Fundam 69(6):563–574
60. Zinjala HK, Banerjee J (2018) A consistent balanced force refined moment-of-fluid method for surface tension dominant two-phase flows. Numer Heat Transf, Part B: Fundam 74(1):432–449
61. Ginzburg I, Wittum G (2001) Two-phase flows on interface refined grids modeled with vof, staggered

finite volumes, and spline interpolants. J Comput Phys 166(2):302–335

62. López J, Hernández J, Gómez P, Faura F (2004) A volume of fluid method based on multidimensional advection and spline interface reconstruction. J Comput Phys 195(2):718–742
63. Scardovelli R, Zaleski S (2000) Analytical relations connecting linear interfaces and volume fractions in rectangular grids. J Comput Phys 164(1):228–237
64. Lehmann M, Gekle S (2022) Analytic solution to the piecewise linear interface construction problem and its application in curvature calculation for volume-of-fluid simulation codes. Computation 10(2):21
65. Rider WJ, Kothe DB (1998) Reconstructing volume tracking. J Comput Phys 141(2):112–152
66. Owkes M, Desjardins O (2014) A computational framework for conservative, three-dimensional, unsplit, geometric transport with application to the volume-of-fluid (vof) method. J Comput Phys 270:587–612
67. Chiodi R, Desjardins O (2022) General, robust, and efficient polyhedron intersection in the interface reconstruction library. J Comput Phys 449:110787
68. Strang G (1968) On the construction and comparison of difference schemes. SIAM J Numer Anal 5(3):506–517
69. Weymouth GD, Yue DK-P (2010) Conservative volume-of-fluid method for free-surface simulations on cartesian-grids. J Comput Phys 229(8):2853–2865
70. Bell JB, Colella P, Glaz HM (1989) A second-order projection method for the incompressible Navier–Stokes equations. J Comput Phys 85(2):257–283
71. Zalesak ST (1979) Fully multidimensional flux-corrected transport algorithms for fluids. J Comput Phys 31(3):335–362
72. Leveque RJ (1996) High-resolution conservative algorithms for advection in incompressible flow. SIAM J Numer Anal 33(2):627–665
73. Dodd MS, Ferrante A (2014) A fast pressure-correction method for incompressible two-fluid flows. J Comput Phys 273:416–434
74. Popinet S (2018) Numerical models of surface tension. Annu Rev Fluid Mech 50:49–75
75. Francois MM, Cummins SJ, Dendy ED, Kothe DB, Sicilian JM, Williams MW (2006) A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. J Comput Phys 213(1):141–173
76. Seric I, Afkhami S, Kondic L (2018) Direct numerical simulation of variable surface tension flows using a volume-of-fluid method. J Comput Phys 352:615–636
77. Brackbill JU, Kothe DB, Zemach C (1992) A continuum method for modeling surface tension. J Comput Phys 100(2):335–354
78. Fedkiw RP, Aslam T, Merriman B, Osher S (1999) A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). J Comput Phys 152(2):457–492
79. Sussman M, Smereka P, Osher S (1994) A level set approach for computing solutions to incompressible two-phase flow. J Comput Phys 114(1):146–159
80. Afkhami S, Bussmann M (2008) Height functions for applying contact angles to 2d vof simulations. Int J Numer Methods Fluids 57(4):453–472
81. Sussman M (2003) A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. J Comput Phys 187(1):110–136
82. Cummins SJ, Francois MM, Kothe DB (2005) Estimating curvature from volume fractions. Comput Struct 83(6–7):425–434
83. Popinet S (2009) An accurate adaptive solver for surface-tension-driven interfacial flows. J Comput Phys 228(16):5838–5866
84. Lopez J, Zanzi C, Gomez P, Zamora R, Faura F, Hernandez J (2009) An improved height function technique for computing interface curvature from volume fractions. Comput Methods Appl Mech Eng 198(33–36):2555–2564
85. Owkes M, Desjardins O (2015) A mesh-decoupled height function method for computing interface curvature. J Comput Phys 281:285–300
86. Hardt S, Wondra F (2008) Evaporation model for interfacial flows based on a continuum-field representation of the source terms. J Comput Phys 227(11):5871–5895
87. Gupta R, Fletcher DF, Haynes BS (2009) On the cfd modelling of Taylor flow in microchannels. Chem Eng Sci 64(12):2941–2950
88. Dodd MS, Ferrante A (2016) On the interaction of Taylor length scale size droplets and isotropic turbulence. J Fluid Mech 806:356–412
89. Gunstensen AK (1992) Lattice-Boltzmann studies of multiphase flow through porous media. PhD thesis, Massachusetts Institute of Technology
90. Magnini M, Pulvirenti B, Thome J (2016) Characterization of the velocity fields generated by flow initialization in the cfd simulation of multiphase flows. Appl Math Model 40(15–16):6811–6830
91. Lafaurie B, Nardone C, Scardovelli R, Zaleski S, Zanetti G (1994) Modelling merging and fragmentation in multiphase flows with surfer. J Comput Phys 113(1):134–147
92. Popinet S, Zaleski S (1999) A front-tracking algorithm for accurate representation of surface tension. Int J Numer Methods Fluids 30(6):775–793
93. Abadie T, Aubin J, Legendre D (2015) On the combined effects of surface tension force calculation and interface advection on spurious currents within volume of fluid and level set frameworks. J Comput Phys 297:611–636
94. Fyfe DE, Oran ES, Fritts M (1988) Surface tension and viscosity with Lagrangian hydrodynamics on a triangular mesh. J Comput Phys 76(2):349–384

95. Torres D, Brackbill J (2000) The point-set method: front-tracking without connectivity. J Comput Phys 165(2):620–644
96. Herrmann M (2008) A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids. J Comput Phys 227(4):2674–2706
97. Prosperetti A (1980) Free oscillations of drops and bubbles: the initial-value problem. J Fluid Mech 100(2):333–347
98. Prosperetti A (1981) Motion of two superposed viscous fluids. Phys Fluids 24(7):1217–1223
99. Ganapathy H, Shooshtari A, Choo K, Dessiatoun S, Alshehhi M, Ohadi M (2013) Volume of fluid-based numerical modeling of condensation heat transfer and fluid flow characteristics in microchannels. Int J Heat Mass Transf 65:62–72
100. Davidson MR, Rudman M (2002) Volume-of-fluid calculation of heat or mass transfer across deforming interfaces in two-fluid flow. Numer Heat Transf: Part B Fundam 41(3–4):291–308
101. Ma C, Bothe D (2011) Direct numerical simulation of thermocapillary flow based on the volume of fluid method. Int J Multiph Flow 37(9):1045–1058
102. Samareh B, Mostaghimi J, Moreau C (2014) Thermocapillary migration of a deformable droplet. Int J Heat Mass Transf 73:616–626
103. Bothe D, Fleckenstein S (2013) A volume-of-fluid-based method for mass transfer processes at fluid particles. Chem Eng Sci 101:283–302
104. Haroun Y, Legendre D, Raynal L (2010) Volume of fluid method for interfacial reactive mass transfer: application to stable liquid film. Chem Eng Sci 65(10):2896–2909
105. Renardy YY, Renardy M, Cristini V (2002) A new volume-of-fluid formulation for surfactants and simulations of drop deformation under shear at a low viscosity ratio. Eur J Mech-B/Fluids 21(1):49–59
106. James AJ, Lowengrub J (2004) A surfactant-conserving volume-of-fluid method for interfacial flows with insoluble surfactant. J Comput Phys 201(2):685–722
107. Tomar G, Gerlach D, Biswas G, Alleborn N, Sharma A, Durst F, Welch SW, Delgado A (2007) Two-phase electrohydrodynamic simulations using a volume-of-fluid approach. J Comput Phys 227(2):1267–1285
108. López-Herrera J, Popinet S, Herrada M (2011) A charge-conservative approach for simulating electrohydrodynamic two-phase flows using volume-of-fluid. J Comput Phys 230(5):1939–1955
109. Tomar G, Biswas G, Sharma A, Agrawal A (2005) Numerical simulation of bubble growth in film boiling using a coupled level-set and volume-of-fluid method. Phys Fluids 17(11):112103
110. Guion A, Afkhami S, Zaleski S, Buongiorno J (2018) Simulations of microlayer formation in nucleate boiling. Int J Heat Mass Transf 127:1271–1284
111. Palmore J Jr, Desjardins O (2019) A volume of fluid framework for interface-resolved simulations of vaporizing liquid-gas flows. J Comput Phys 399:108954
112. López J, Gómez P, Hernández J, Faura F (2013) A two-grid adaptive volume of fluid approach for dendritic solidification. Comput Fluids 86:326–342
113. Reitzle M, Kieffer-Roth C, Garcke H, Weigand B (2017) A volume-of-fluid method for three-dimensional hexagonal solidification processes. J Comput Phys 339:356–369
114. Karagadde S, Bhattacharya A, Tomar G, Dutta P (2012) A coupled vof-ibm-enthalpy approach for modeling motion and growth of equiaxed dendrites in a solidifying melt. J Comput Phys 231(10):3987–4000
115. Vincent S, De Motta JCB, Sarthou A, Estivalezes J-L, Simonin O, Climent E (2014) A Lagrangian vof tensorial penalty method for the dns of resolved particle-laden flows. J Comput Phys 256:582–614
116. Popinet S (2014) Basilisk. URl: http://basilisk.fr. Accessed 21 Oct 2019
117. Natarajan M, Chiodi R, Kuhn M, Desjardins O (2019) An all-Mach multiphase flow solver using block-structured amr. In: ILASS-Americas 30th Annual Conference on Liquid Atomization and Spray Systems, Tempe, Az
118. López J, Hernández J, Gómez P, Faura F (2018) Voftools- a software package of calculation tools for volume of fluid methods using general convex grids. Comput Phys Commun 223:45–54
119. López J, Hernandez J, Gomez P, Zanzi C, Zamora R (2020) Voftools 5: an extension to non-convex geometries of calculation tools for volume of fluid methods. Comput Phys Commun 252:107277
120. Brunton SL, Noack BR, Koumoutsakos P (2020) Machine learning for fluid mechanics. Annu Rev Fluid Mech 52:477–508
121. Qi Y, Lu J, Scardovelli R, Zaleski S, Tryggvason G (2019) Computing curvature for volume of fluid methods using machine learning. J Comput Phys 377:155–161
122. Patel H, Panda A, Kuipers J, Peters E (2019) Computing interface curvature from volume fractions: a machine learning approach. Comput Fluids 193:104263
123. Önder A, Liu PLF (2022) Deep learning of interfacial curvature: a symmetry-preserving approach for the volume of fluid method. arXiv preprint arXiv:2206.06041
124. Ataei M, Bussmann M, Shaayegan V, Costa F, Han S, Park CB (2021) Nplic: a machine learning approach to piecewise linear interface construction. Comput Fluids 223:104950
125. Després B, Jourdren H (2020) Machine learning design of volume of fluid schemes for compressible flows. J Comput Phys 408:109275
126. Zhu L-T, Chen X-Z, Ouyang B, Yan W-C, Lei H, Chen Z, Luo Z-H (2022) Review of machine learning for hydrodynamics, transport, and reactions in multiphase flows and reactors. Ind Eng Chem Res 61(28):9901–9949

Ananthan Mohan Ananthan is currently pursuing PhD in the department of Mechanical Engineering, Indian Institute of Science. He completed his BTech and Masters from the department of Mechanical Engineering, NSS college of Engineering and from the department of Mechanical Engineering, Indian Institute of Science in 2014 and 2017, respectively. His areas of research interest are numerical simulations of multiphase flows and thin fluid films and experimental studies involving bubble surface interactions.

Gaurav Tomar Dr. Gaurav Tomar is a Professor in the Department of Mechanical engineering at Indian Institute of Science, Bangalore. He completed his Bachelor's and PhD from the Indian Institute of Technology, Kanpur, in 2003 and 2008, respectively. Before joining the Indian Institute of Science in 2010 as an Assistant Professor, he worked as a postdoctoral fellow in the University of Pierre et Marie Curie, Paris, and University of California Santa Barbara, USA. His research interests are in the area of numerical simulations of multiphase phenomena, interfacial instabilities, multiscale simulations, electrohydrodynamics of bubbles and drops and complex fluid flows.